Oficina de Accesibilidad Universal

DIRECTRICES WCAG, PRINCIPIO 4: ROBUSTEZ

Accesibilidad DIGITAL

aspaym castilla y león

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

Servicios Sociales de Castilla y León

Junta de Castilla y León

ÍNDICE

PAUTA 13: COMPATIBLE

El objetivo de esta pauta es maximizar la compatibilidad con los agentes de usuario actuales y futuros, incluidas las tecnologías de asistencia.

Esto se consigue:

- ✓ Asegurándose de que los autores/as no hagan cosas que romperían o eludirían las tecnologías de asistencia.
  Por ejemplo, usar marcado o código no convencional.
- ✓ Poniendo la información de formas estándares que las tecnologías de asistencia puedan reconocer (e interactuar con ellas).

  - **Nivel A**

❖ **Criterio de conformidad: análisis**

En el **contenido** implementado utilizando **lenguajes de marcado**:

- ✓ Los elementos deben tener etiquetas de inicio y finalización completas.
- ✓ Las etiquetas de inicio y finalización a las que les falta un carácter crítico (por ejemplo, un paréntesis o una comilla) no están completas.
- ✓ Los elementos se deben anidar, según sus especificaciones.
- ✓ Los elementos no deben contener atributos duplicados.
- ✓ Cualquier Identificador de Usuario debe ser único.

El **objetivo** es que los **agentes de usuario** (incluidas las tecnologías de asistencia) puedan **interpretar y analizar** el contenido con precisión.

Si el **contenido no se puede analizar** en una estructura de datos, algunos agentes de usuario pueden presentarlo de manera diferente o ser incapaces de analizarlo.

Algunos agentes de usuario usan **“técnicas de reparación”** para representar un contenido mal codificado.

Estas técnicas **varían entre los agentes** de usuario, por lo que los/as creadores deben generar contenido siguiendo las reglas definidas en la gramática formal para esa tecnología.

Técnicas suficientes

- ✓ G 134, validar las páginas web.
- ✓ G 192, totalmente conforme a las especificaciones.
- ✓ H 88, usar HTML siguiendo las especificaciones.
- ✓ Garantizar que las páginas web se puedan analizar mediante una de estas técnicas:
  - H 74, garantizar que las etiquetas de apertura y cierre se usan según la especificación; H 93: garantizar que los atributos de identificación sean únicos en una página web; y H 94: garantizar que los elementos no tengan atributos duplicados.
  - H 75, asegurarse de que las páginas web están bien formadas.
- ✓ SL 33, usar XAML bien formado para definir una interfaz de usuario de Silverlight.

❖ **Criterio de conformidad: nombre, función, valor**

Para todos los **componentes de la interfaz de usuario** (incluidos elementos de formularios, enlaces, etc.):

- El **nombre y función** pueden ser determinados programáticamente.
- Los **estados, propiedades y valores** que la persona pueda establecer pueden ser programados.
- La **notificación de cambios** en estos artículos está **disponible** para los agentes de usuario, incluidas las tecnologías de asistencia.

Así, se garantiza que las tecnologías de asistencia puedan **recopilar información, activar** (o establecer) y **mantenerse al día** sobre el estado de los controles de la interfaz de usuario en el contenido.

Ofrecer **información** de la función, estado y valor de todos los componentes de la interfaz de usuario permite la compatibilidad con las tecnologías de asistencia. Por ejemplo, lectores de pantalla, ampliadores de pantalla, etc.

Cuando se usan **controles estándar** de tecnologías accesibles, este proceso es sencillo.

Si se crean **controles personalizados** o se programan **elementos** de la interfaz para que tengan un **rol y/o función diferente** a la habitual, se deben tomar **medidas adicionales** para que los controles brinden información importante a las tecnologías de asistencia y para que estas puedan controlarlos.

Técnicas suficientes

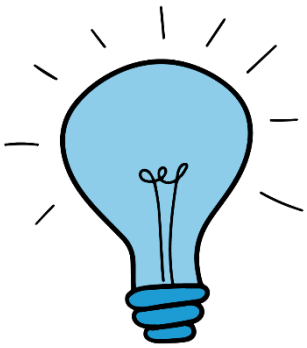
Si se usa un **componente de interfaz de usuario** estándar en un lenguaje de marcas (por ejemplo, HTML), se puede:

- ✓ ARIA 14, usar aria-label para ofrecer una etiqueta invisible donde no se pueda usar una visible.
- ✓ ARIA 16, usar aria-labelledby para dar un nombre a los controles de la interfaz de usuario.
- ✓ G 108, usar funciones de marcado para exponer el nombre y el rol, permitir que las propiedades configurables por la persona usuaria se establezcan directamente y ofrecer notificaciones de cambios, usando estas técnicas:
  - H 91, usar enlaces y controles de formulario.
  - H 44, usar elementos de etiqueta para asociar etiquetas de texto con controles de formulario.
  - H 64, usar el atributo de título del marco y los elementos iframe.

- H 65, usar el atributo para identificar controles de formulario cuando no se pueda usar el elemento de etiqueta.
- H 88, usar HTML según las especificaciones.

Si se usa **script** o código para reutilizar un componente de interfaz de usuario estándar en un lenguaje de marcado, se pueden: **exponer los nombres y roles**; permitir que las **propiedades configurables** por la persona usuaria se establezcan directamente; y ofrecer notificaciones de cambios.

Si se usa un componente de interfaz de usuario en una **tecnología de programación**, se pueden: usar las características de la API de accesibilidad de una tecnología para **exponer nombres y funciones**; permitir que las propiedades configurables por la persona se establezcan directamente; y ofrecer notificaciones de cambio. Para ello, se usan las técnicas siguientes:



- ✓ FLASH 32, usar el etiquetado automático para asociar etiquetas de texto con controles de formulario.
- ✓ Configurar la propiedad de etiqueta para componentes de formulario (FLASH 29) o especificar nombres accesibles para botones de imagen (FLASH 30).
- ✓ Ofrecer etiquetas para controles de formulario interactivos en documentos PDF (PDF 10) o información de nombre, función y valor para campos de formularios en documentos PDF (PDF 12).
- ✓ Usar Labeledby para asociar etiquetas y objetivos en Silverlight (SL 26) o elementos de texto de Silverlight para un rol de accesibilidad adecuado (SL 32).

Si se crea un componente de interfaz de usuario propio en un lenguaje de programación, se pueden: crear componentes usando una tecnología que admita las características de la **API de accesibilidad** de las plataformas donde se ejecutan los agentes de usuario para exponer nombres y roles; permitir que las propiedades configurables por la persona usuaria se establezcan directamente; y ofrecer notificaciones de cambio.

Para ello, se usan las técnicas siguientes:

- ✓ ARIA 4, usar un rol WAI-ARIA para exponer el rol de un componente de interfaz de usuario.
- ✓ ARIA 5, usar atributos de propiedad y estado WAI-ARIA para exponer el estado de un componente de interfaz de usuario.
- ✓ ARIA 16, usar aria-labelledby para dar un nombre a los controles de la interfaz de usuario.
- ✓ SL 6, definir un UI de automatización de la interfaz de usuario para un control Silverlight personalizado.
- ✓ SL 18, ofrecer un texto equivalente para controles Silverlight sin texto con Automationproperties.Name.
- ✓ SL 20, confiar en Silverlight Automation Peer Behaviour para establecer Automationproperties.Name.

- ✓ SL 30: usar Silverlight Control Compositing y Automationproperties.Name.

Técnicas recomendables

Ofrecer etiquetas para todos los controles de formulario que no tengan etiquetas implícitas.

- **Nivel AA**

❖ **Criterio de conformidad: mensajes de estado**

Este criterio se dirige, sobre todo, a las personas ciegas o con baja visión, usuarias de tecnologías de asistencia.

Las **personas** deben ser **conscientes** de los **cambios** importantes en el **contenido** a los que no se presta atención, de una forma que no interrumpa su trabajo.

Este criterio es específico para los cambios en el contenido que implican mensajes de estado. Estos **mensajes de estado**:

- ✓ Informan a la persona de los resultados de una acción, estado de espera de una aplicación, progreso de un proceso o existencia de errores.
- ✓ No se entregan mediante un cambio en el contexto.

Se puede añadir información a las páginas que no cumplan esta definición de mensajes de estado.

Un **ejemplo** de este criterio es cuando una persona pulsa el botón “Buscar”: el contenido de la página se actualiza para incluir los resultados de la búsqueda, que aparecen debajo de dicho botón. El cambio de contenido incluye el mensaje "Se encontraron 5 resultados”.

Técnicas suficientes

Si un mensaje de estado informa sobre los resultados de una acción o el estado de una aplicación, se puede:

- ✓ Usar role=status para presentar dicho mensaje
- ✓ G 199, ofrecer comentarios de éxito cuando los datos se envían correctamente

Si un mensaje de estado informa de una **sugerencia** o una advertencia sobre un error, se puede usar ARIA role=alert o Live Regions para identificar errores, además de:

- ✓ G 83, ofrecer descripciones de texto para identificar los campos obligatorios que no se completaron.
- ✓ Ofrecer una descripción de texto, si la persona da información que no está en la lista de valores permitidos (G 84) o si está fuera del formato o los valores requeridos (G 85).
- ✓ Ofrecer texto de corrección sugerido (G 177) o revisión ortográfica y sugerencias para el ingreso de texto (G 194).

Si un mensaje de estado informa del **progreso** de un proceso, se puede usar role=log para identificar actualizaciones de información secuencial (ARIA 23) o role=status para presentar mensajes de estado (ARIA 22), con ayuda de un asistente en la página web (G 193).

NORMATIVA

W3C Accessibility Guidelines (Web Content Accessibility Guidelines) 3.0.