

# *Accesibilidad*

## Accesibilidad Web

Tras la lectura de la presentación de esta Unidad, responde a las siguientes cuestiones sobre Accesibilidad Web.

1. Desarrolla en qué consisten los 4 pasos para la estandarización de una especificación del W3C.
  - Working Draft (WD):
  - Candidate Recommendation (CR):
  - Proposed Recommendation (PR):
  - W3C Recommendation (REC):
2. Busca el estado en el que se encuentra la especificación oficial de la pseudoclase :has() en la web del W3C. También, desde caniuse.com verifica el nivel de soporte que tienen los navegadores de esta pseudoclase actualmente.
3. Lee la guía de introducción a la Accesibilidad Web (en castellano):  
<https://www.w3.org/WAI/fundamentals/accessibility-intro/es> y visualiza el video de introducción.  
Después responde a las siguientes cuestiones:
  1. ¿A quién beneficia hacer la Web accesible?
  2. ¿Qué significa Accesibilidad Web?
  3. Visualiza el video:

Escribe todos los casos de técnicas de accesibilidad y usabilidad que se explican (10 en total).

4. Además de a personas con algún tipo de discapacidad, la accesibilidad también es útil para personas sin discapacidad. Pon varios ejemplos de este beneficio.
  5. También, la accesibilidad da soporte a la inclusión social, ¿a quién?
  6. ¿De qué se encarga WAI?
  7. ¿Desde qué punto del desarrollo de un proyecto es conveniente incorporar la accesibilidad?
  8. ¿Se puede evaluar la accesibilidad? ¿Cómo?
- 
4. Consultando la guía de referencia de las técnicas para satisfacer los requisitos definidos en las WCAG (<https://www.w3.org/WAI/WCAG21/Techniques/>) responde a las siguientes cuestiones:
    1. En las técnicas generales, localiza cuáles son las recomendaciones con respecto a: pausar contenido, alineación de texto, contraste de colores, localización en el sitio.
    2. Técnicas HTML: orden del foco de los elementos, atributos de las imágenes, botones en formularios.
    3. Técnicas CSS: interlineado, unidades de medida para los contenedores principales, uso de unidades para fuentes.

4. Errores comunes: fallos en alineación del texto, fallos en enlaces, fallos en imágenes.
  
5. Realiza un análisis de accesibilidad a las siguientes webs y analiza los resultados:
  1. <https://juntadeandalucia.es/>
  2. <https://www.upo.es/>
  3. <http://www.as.com/>

## WAI-ARIA

Lee el apartado "ARIA, el aliado (casi) desconocido", página 197, del libro "Accesibilidad Web - WCAG 2.1 de forma sencilla" y responde a las siguientes cuestiones:

1. ¿Cuál es la idea principal por la que el W3C desarrolló WAI-ARIA?
2. ¿Cuáles son los tres elementos fundamentales de los que se compone ARIA?
3. Analiza el Ejemplo 1: "Cambiar el funcionamiento de un objeto: una capa que se comporta como un botón".
4. ¿En qué casos se recomienda utilizar ARIA en lugar de elementos HTML nativos?
5. ¿Qué es un rol en ARIA?
6. ¿Qué es un landmark role"?
7. ¿Qué es una "live region"?
8. ¿Qué diferencia hay entre los estados y las propiedades ARIA?
9. ¿Qué usos tiene el atributo aria-label?

10. ¿Y aria-labelledby? ¿Y aria-describedby?
11. Analiza el Ejemplo 2: "Mejorar el uso accesible: navegación en pestañas".
12. Revisa la lista de Buenas prácticas.
13. Analiza el Ejemplo 3: "Validar un campo obligatorio".
14. Revisa las 21 técnicas específicas de ARIA.

## Accesibilidad Web

### **1. Desarrolla en qué consisten los 4 pasos para la estandarización de una especificación del W3C.**

- **Working Draft (WD):** Primer borrador público de la especificación. Se publica para recibir comentarios de la comunidad.
- **Candidate Recommendation (CR):** La especificación está lista para ser implementada y probada en el mundo real.
- **Proposed Recommendation (PR):** Se somete a revisión final antes de su adopción oficial.
- **W3C Recommendation (REC):** Se convierte en un estándar oficial del W3C.

### **2. Estado de la especificación oficial de la pseudoclase :has( ) en el W3C y soporte en navegadores.**

- Estado en W3C: Consultar <https://www.w3.org/TR/selectors-4/>.
- Soporte en navegadores: Consultar <https://caniuse.com/css-has>.

### **3. Introducción a la Accesibilidad Web**

#### **1. ¿A quién beneficia hacer la Web accesible?**

- Personas con discapacidad visual, auditiva, motriz o cognitiva.
- Personas sin discapacidad en situaciones temporales o contextuales.
- Usuarios con dispositivos de acceso limitados.

#### **2. ¿Qué significa Accesibilidad Web?**

- Características que permiten que cualquier persona pueda acceder, comprender y operar una web.

### 3. Técnicas de accesibilidad y usabilidad (10 ejemplos):

- Texto alternativo en imágenes.
- Subtítulos en videos.
- Alto contraste de colores.
- Compatibilidad con lectores de pantalla.
- Navegación mediante teclado.
- Etiquetas claras en formularios.
- HTML semántico.
- Descripciones adecuadas en enlaces.
- Evitar animaciones molestas.
- Diseño flexible y adaptable.

### 4. Beneficios para personas sin discapacidad:

- Un conductor usando navegación por voz.
- Un usuario con conexión lenta beneficiándose de un diseño optimizado.
- Un adulto mayor con dificultades motrices.

### 5. Apoyo a la inclusión social:

- Personas mayores, personas en zonas con baja conectividad, hablantes de otros idiomas.

### 6. ¿De qué se encarga WAI?

- Desarrollar estándares y pautas para la accesibilidad web.

### 7. ¿Cuándo se debe incorporar la accesibilidad?

- Desde el inicio del desarrollo del proyecto.



## 8. ¿Se puede evaluar la accesibilidad? ¿Cómo?

- Sí, con herramientas automáticas (WAVE, Lighthouse) y pruebas con usuarios reales.

## 4. Recomendaciones de las WCAG

### 1. Técnicas generales:

- **Pausar contenido:** Debe poder detenerse o esconderse.
- **Alineación de texto:** Preferencia por alineación izquierda.
- **Contraste de colores:** Relación mínima de 4.5:1.
- **Localización en el sitio:** Incluir mapas del sitio y navegación clara.

### 2. Técnicas HTML:

- **Orden del foco:** Uso de tabindex correctamente.
- **Atributos en imágenes:** alt obligatorio.
- **Botones en formularios:** Uso de etiquetas label.

### 3. Técnicas CSS:

- **Interlineado:** Mínimo 1.5 veces el tamaño del texto.
- **Unidades de medida:** Uso de em o rem en lugar de píxeles.
- **Fuentes:** Evitar tamaños de fuente muy pequeños.

### 4. Errores comunes:

- **Alineación incorrecta del texto.**
- **Enlaces sin descripciones claras.**
- **Uso inadecuado de imágenes sin alt.**

## 5. Análisis de accesibilidad de sitios web

- Evaluar usando herramientas como WAVE y Lighthouse.

## WAI-ARIA

### 1. ¿Por qué se desarrolló WAI-ARIA?

- Para mejorar la accesibilidad de interfaces interactivas.

### 2. Tres elementos fundamentales de ARIA:

- **Roles:** Definen la función de los elementos.
- **Estados:** Indican cambios dinámicos.
- **Propiedades:** Proveen más información.

### 3. Ejemplo 1: Capa como botón

- Se debe asignar `role="button"` y manejar eventos de teclado.

### 4. Cuándo usar ARIA en lugar de HTML:

- Cuando HTML nativo no ofrece suficiente semántica.

### 5. Definiciones clave:

- **Rol:** Define la función de un elemento.
- **Landmark Role:** Define la estructura de una página.
- **Live Region:** Áreas con contenido dinámico.
- **Estados vs. Propiedades:** Estados cambian dinámicamente, propiedades son estáticas.

### 6. Usos de atributos ARIA:

- **aria-label:** Proporciona una etiqueta accesible.
- **aria-labelledby:** Relaciona un elemento con otro.
- **aria-describedby:** Proporciona información adicional.

## 7. Ejemplos de ARIA:

- **Navegación en pestañas:** Gestionar foco y estados.
- **Validación de campos:** `aria-invalid`, `aria-required`.

## 8. Revisión de técnicas específicas de ARIA.

- Implementaciones detalladas en los documentos proporcionados.

## 9. Ejemplo 2: Mejorar el uso accesible: navegación en pestañas

- Se utilizan roles y estados ARIA para mejorar la accesibilidad de los tabuladores.
- `role="tablist"` para el contenedor de pestañas.
- `role="tab"` para cada pestaña individual con `aria-selected="true"` o `false`.

## 10. Lista de Buenas Prácticas en ARIA:

- Usar roles y propiedades ARIA solo cuando sea necesario.
- No reemplazar elementos HTML nativos con ARIA si ya son accesibles.
- Garantizar que los estados ARIA sean actualizados dinámicamente.

## 11. Ejemplo 3: Validar un campo obligatorio

- Se usa `aria-required="true"` para indicar que un campo es obligatorio.
- `aria-invalid="true"` cuando hay un error en el campo.
- Mensajes de error accesibles con `aria-describedby`.

## 12. Revisión de las 21 técnicas específicas de ARIA

- Técnicas para mejorar accesibilidad en formularios, menús, navegación y controles interactivos.