

## TEMA 2.2. ACCESIBILIDAD

La digitalización de la información es una forma de acceder simple y rápidamente a información sobre una asociación, empresa, producto, etc. En una generación conectada, los resultados de una búsqueda deben proporcionar datos verdaderos e informativos. Retener a los visitantes es un factor clave para transmitir la información deseada y que nuestra aplicación tenga éxito. No obstante, tener una interfaz visualmente atractiva, exitosa y transformativa no garantiza que todas las personas la puedan usar. Por ejemplo, sólo en Europa, se estima que hay más de 30 millones de personas ciegas o con discapacidad visual (datos de la Comisión Europea). Es imperativo que en un mundo cada vez más progresista e inclusivo todas las personas tengan los mismos derechos y, por lo tanto, el mismo acceso a los datos.

Hay varios aspectos que deben tenerse en cuenta al implementar aplicaciones e interfaces accesibles. Debido al aspecto interactivo de éstas, se deben implementar varios procedimientos para que, sin importar qué persona acceda, la experiencia sea la misma. Este tipo de linealidad es difícil de lograr. Si, por ejemplo, una interfaz está constituida principalmente por imágenes o vídeos, es muy difícil brindar una experiencia completa a una persona ciega. Si tiene podcasts, una persona sorda no puede beneficiarse de ellos. Si tiene muchos enlaces es difícil para una persona con discapacidad motora interactuar. Otros ejemplos de barreras digitales son fáciles de encontrar, pero difíciles de resolver.

### Inclusividad Digital

La inclusividad digital significa que todas las personas sean capaces de acceder y comprender la información que se les brinda en las posibles aplicaciones. No obstante, es necesario entender a qué tipos de problemas se enfrentan las personas con discapacidad. Con tal fin, la siguiente tabla presenta de una forma simplificada (y no exhaustiva) del modo de abordar las distintas discapacidades:

Categoría	Resumen	Discapacidad Asociada
Visual	Puede tratarse de usuarios no videntes, con poca capacidad visual, con visión obstruida o incluso ancianos.	Miopía
		Daltonismo
		Glaucoma
		Albinismo
Auditiva	La interfaz es un medio visual, pero los subtítulos y alternativas para los medios auditivos deben considerarse para los usuarios con discapacidad auditiva.	Presbiacusia
		Trauma acústico
		Trastorno del procesamiento auditivo

		Otosclerosis
Motora	Las personas con impedimentos motores generalmente usan una amplia gama de tecnología de asistencia, desde teclados especializados o rastreadores oculares, hasta el uso de botones individuales.	Parálisis cerebral
		Parkinson
		Distrofia muscular
		RSI
Cognitiva	Problemas con el procesamiento de información compleja.	Autismo
		Retraso global del desarrollo
		Dislexia
		Síndrome de Down

Con esto es posible acotar los distintos problemas de una posible implementación a nivel de diseño o planificación de una interfaz o aplicación.

## Reglas para la implementación

### WCAG 1.0

La idea de la inclusión en el desarrollo de aplicaciones a menudo se olvida debido a varios factores como una mala planificación, falta de conocimiento, etc. Este problema se agrava con el hecho de que no existe una guía definitiva para desarrollar una interfaz de usuario inclusiva. El W3C ha propuesto las Pautas de Accesibilidad al Contenido en la Web (WCAG) 1.0, en 1997, como una propuesta inicial de pautas de desarrollo y métodos de garantía. Aunque fue un esfuerzo valioso, tuvo poco impacto y fue criticado por las asociaciones de personas con discapacidad porque contenía errores de suposiciones y recomendaciones.

### WCAG 2.1

La versión 2.1 era, al mismo tiempo, una mejora de la versión anterior y una simplificación. La mejora incluye las quejas y sugerencias de la comunidad de personas con discapacidad y sus asociaciones. Además, con la evolución de la tecnología llegan nuevos procedimientos y características, que también se incluyeron en esta versión. El proceso de simplificación se realizó para atender a un espectro más amplio de discapacidades. Este proceso fue el resultado directo del método anterior, eliminando reglas obsoletas y consolidando las existentes.

La versión 2.1 establece directamente cuatro principios: inteligible, interactivo, comprensible y robusto. Estos cuatro principios establecen las reglas básicas para un desarrollo accesible. A continuación se incluye una breve explicación de cada uno de los principios:

**Inteligible:** la información y los componentes de la interfaz de usuario deben presentarse para que cualquier usuario pueda conocerlos;

**Interactivo:** los componentes de la interfaz de usuario deben ser interactivos;

**Comprensible:** la información presentada debe ser comprendida por los usuarios;

**Robusto:** el contenido debe ser lo suficientemente robusto como para ser interpretado directamente de manera confiable por los usuarios y por los técnicos.

Aparte de estos principios hay reglas más específicas que establecen guías específicas que deben seguirse para implementar interfaces accesibles e inclusivas, aunque eso queda fuera de este curso.

## ARIA

El contenido digital debe desarrollarse cuidadosamente para transmitir el mensaje incluso si el que está leyendo lo está haciendo en condiciones menos óptimas. Los elementos de la interfaz de usuario que no son textuales, deben tener información adicional sobre lo que están presentando porque habrá personas que no podrán verlos ni escucharlos. El Assistive Rich Internet Applications (ARIA) es la especificación relativa a este campo que utiliza campos HTML para proporcionar información adicional a los usuarios.

Específicamente, el ARIA establece los atributos que la mayoría de los lectores de pantalla pueden leer. Por ejemplo, los botones y las barras de navegación son difíciles de interpretar sin un guía visual y los lectores de pantalla a veces tienen dificultades para traducirlos en elementos navegables. ARIA establece reglas que pueden usarse para mejorar el conocimiento de los lectores de pantalla sobre lo que están leyendo.

## El proyecto A11Y

El proyecto A11Y está desarrollado por la comunidad de desarrolladores y diseñadores web y presenta pautas para desarrollar interfaces accesibles. La diferencia entre el WGAC y este proyecto es que el A11Y se enfoca en mostrar, recorriendo a demostraciones, cómo se pueden implementar ciertas reglas presentes en el WGAC. Si bien no establece reglas y estándares, este proyecto es muy útil para los desarrolladores, ya que aprenden a interpretar la web accesible.

## Guías de desarrollo

### Generadores Automáticos

Una recomendación compartida por el A11Y y el WGAC es que sistemas de administración de contenido (CMS) como Wordpress, Dupral y otros, deben evitarse si es posible. Esto es porque son muy cerrados (en términos de código fuente y edición) y se basan en temas (tramas visuales que dictan el aspecto visual de la interfaz). Son muy pocos los temas que se preocupan por los estándares de accesibilidad, no garantizando que la accesibilidad se

mantenga cuando se hacen cambios al tema o contenido. Además, dependen, en gran medida, de complementos (plug-ins) para implementar algunas de las características.

Estos complementos tampoco están libres de sus propios problemas:

- Son de código fuente cerrado o son difíciles de editar correctamente y de manera oportuna;
- Los que tienen más funcionalidades son caros o tienen una suscripción periódica;
- Raramente se actualizan;
- Pueden depender de otros complementos o ser incompatibles con otros complementos;
- Pueden dejar de funcionar cuando se actualiza el CMS.

Estos problemas implican un gran problema en términos de control y costes. Si bien para el editor o el propietario del contenido es muy fácil publicar contenido nuevo, éstos tienen poco control sobre cómo se presenta. Además, si el coste es una variable que debe considerarse, los desarrolladores deben investigar las características que necesitan y el beneficio de los complementos correspondientes.

## Desarrollo Manual

Se deben fomentar las herramientas de código abierto, ya que se sabe que son más seguras y se actualizan con frecuencia, por ejemplo, que el servidor web sea de código abierto como Apache, Nginx, Node, XAMP, lighttpd, etc.

Con estos servidores apenas es necesario programar estáticamente una interfaz de usuario (usando HTML y CSS). Esto abre la posibilidad al desarrollador de controlar todos los aspectos de las aplicaciones desarrolladas. El despliegue de esos archivos es sencillo: cuando el archivo se carga en el servidor, está inmediatamente disponible como una interfaz con el formato visual apropiado. La navegación de las páginas se realiza comúnmente a través de una estructura de carpetas. Por lo tanto, para el proveedor de contenido, la tarea de escribir contenido es como escribir un archivo de texto para que pueda enfocarse en el contenido y su estructura.

## Estructura de contenido

En cuanto a la estructura de la página, el contenido debe desarrollarse cuidadosamente para transmitir el mensaje para todas las personas. Los elementos de la página no textuales deben tener información adicional sobre lo que están presentando porque habrá personas que no podrán verlos ni escucharlos.

El HTML debe usarse en todo su potencial, recurriendo a los campos especiales para proporcionar información adicional que puede no mostrarse visualmente/auditivamente, pero cuando un lector de pantalla los interpreta, representará esa información para la persona que lo utiliza. Esto, por ejemplo, se puede hacer con los siguientes campos:

**Propiedad HTML Alt:** fue diseñado para presentar texto alternativo a una imagen. Por lo tanto, el Alt debe usarse de manera adecuada y en favor de las personas que no pueden ver esa imagen.

**Propiedad HTML *initial-scale*:** indica la escala inicial a que deben ser representados los elementos de una aplicación. Esto influye en la adaptabilidad de contenido y debe ser usada para indicar si elementos visuales pueden ser expandidos o cambiados de forma elegante (para el caso de lectores de pantalla).

**Propiedad CSS *outlines*:** se debe evitar usar reglas que eliminen los indicadores visuales de los enlaces o controlen el estado enfocable de los mismos. Además, se deben evitar métodos que oculten o cambien la apariencia significativamente (como el desenfoque) ya que los usuarios que sólo usan teclado no podrán interactuar con el enlace o el control.

**Propiedad HTML *track*:** el HTML5 ha introducido la capacidad de incluir subtítulos y transcripciones al vídeo. Al incluir subtítulos, el vídeo es más accesible para aquellos con problemas de audición y tener una transcripción en la interfaz puede proporcionar la capacidad de traducir ese vídeo / audio a otros idiomas.

Si bien parece obvio, planificar a conciencia es muy importante en las primeras etapas del desarrollo. Teniendo en cuenta que se utilizarán interfaces estáticas, funciones de JavaScript y aspectos visuales personalizados, es fundamental planificar qué propiedades y efectos se utilizarán. Además, los desarrolladores deben anticipar futuras revisiones y crear una lista comprensiva de propiedades y ejemplos de su uso.

## Estructura visual y aspecto visual

La estructura visual y el aspecto visual deben considerarse cuidadosamente. La estructura visual es cómo la aplicación estructura su información, mientras que el aspecto visual es el resultado de esa representación. Las aplicaciones deben ser atractivas y al mismo tiempo funcionales. Los rasgos visuales deben considerarse cuidadosamente, así como los medios detrás de ellos.

El contenido debe mostrarse cuidadosamente teniendo en cuenta la tabla presentada anteriormente. Nunca es suficiente enfatizar que la presentación es crítica para transmitir el mensaje presentado por el contenido. La aplicación debe ser atractiva y no estar *abarrotada*, transmitir el mayor contenido posible sin ser agotadora y, lo más importante, captar la atención del usuario.

Sin embargo, el contenido de terceros y las funcionalidades extravagantes deben elegirse bien y con cuidado. Un carrusel de imágenes es una adición bonita y ayuda a capturar el ojo del usuario, pero es muy complicado para los usuarios que no usan el ratón y puede provocar ataques si las imágenes se cambian demasiado rápido. Esto no significa que una interfaz de usuario deba ser despojada de toda interacción, sino que los desarrolladores deben probar todos los componentes y verificar que funcionan en todas las situaciones posibles.

Por ejemplo, algo a tener en cuenta es la reproducción automática de vídeos con música, ya que distraen mucho y pueden afectar a personas con discapacidades cognitivas.

Otro ejemplo es el contraste de color. El WCAG 2.1 define cuatro niveles de relaciones de contraste que funcionan en términos de redacción. Los aceptables son:

- **AA:** criterio de éxito de 1.4.3 (contraste mínimo) para fuentes de menos de 18 puntos.
- **AAA:** criterio de éxito de 1.4.6 (contraste mejorado) para fuentes de menos de 18 puntos.

A estos niveles se debe agregar lo siguiente:

- **Colores:** según el brillo y la diferencia de color. Un uso correcto significa que la página web es totalmente compatible con colores no agresivos.

Node	Foreground	Background	Sample	Contrast Ratio	Brightness difference	Color difference
BODY class='div'	#000000	#FFFFFF	Sample Text	21:1 AAA		
... DIV id='cc-inner' class='cc-content-parent'	#000000	#FFFFFF	Sample Text	21:1 AAA		
... ... DIV id='cc-tp-container' class='cc-content-parent'	#000000	#FFFFFF	Sample Text	21:1 AAA		
... ... ... DIV id='cc-tp-nav'	#000000	#FFFFFF	Sample Text	21:1 AAA		
... ... ... ... DIV class='cc-tp-gutter'	#000000	#FFFFFF	Sample Text	21:1 AAA		
... ... ... ... ... DIV	#000000	#FFFFFF	Sample Text	21:1 AAA		
... ... ... ... ... ... DIV class='l-				21:1		

- **Diferencia de color:** cuán legible es el texto para personas con daltonismo.

## Pruebas

Finalmente, para probar estas reglas hay varias herramientas y formas que se pueden usar. Las más simples son las siguientes:

### Herramienta 1: desconecta el ratón

La primera prueba es muy simple. Al no usar el ratón y solamente el teclado, se prueba la experiencia de navegación que tienen las personas que no pueden usar el ratón. Esta prueba probablemente mostrará los errores más comunes y peores cometidos.

### Herramienta 2: desconecta el teclado

Al igual que la herramienta anterior, este procedimiento mostrará cómo las personas que no pueden usar el teclado pueden navegar por la interfaz. Además, al usar el teclado virtual que usa la mayoría de estas personas, los desarrolladores pueden ver si una función funciona correctamente. Algunos procedimientos de interacción están vinculados a la activación de teclas o teclas de activación, que a veces no funcionan sin un teclado físico.