



IAAP

International Association
of Accessibility Professionals



**ESPECIALISTA IN ACCESSIBILIDAD
WEB (WAS) DE LA IAAP**

**Guía de Estudio para la
certificación WAS**

Octubre 2020

Unidos en la Accesibilidad

www.accessibilityassociation.org

Guía de Estudio para la certificación WAS
ESPECIALISTA EN ACCESIBILIDAD WEB
DE LA IAAP

Octubre 2020

Unidos en la Accesibilidad www.accessibilityassociation.org (en inglés)

A. El Propósito de este Documento

Esta Guía de Estudio para la certificación WAS describe las habilidades que se esperan de las personas que buscan obtener la certificación de Especialista en Accesibilidad Web (WAS por sus siglas en inglés). El examen de Especialista en Accesibilidad Web (WAS) de la IAAP es un examen de conocimientos técnicos.

El conocimiento de un Especialista en Accesibilidad Web incluye la comprensión de la teoría, los principios y la información fundamental de la accesibilidad web adecuada para un nivel intermedio de habilidades y experiencia. La certificación IAAP WAS requiere de tres a cinco años de experiencia laboral práctica en o con un equipo de accesibilidad web.

Específicamente, los tres propósitos principales de este documento son los siguientes:

1. Enumerar las temáticas cubiertas en el examen.
2. Presentar información general sobre cada temática.
3. Enumerar recursos adicionales para ayudar a las personas a prepararse para el examen WAS.

La Guía de Estudio está diseñada para ser un punto de partida al estudiar para el examen WAS. No pretende ser una explicación exhaustiva de cada concepto o pregunta del examen. Tenga en cuenta que el uso de esta guía no garantiza la finalización exitosa del examen.

A partir de la actualización de la Guía de Estudio para la certificación WAS de junio de 2020, esta cubre conceptos de WCAG 1, WCAG 2.0 y WCAG 2.1. A menos que se discuta información específica sobre una determinada versión, se utilizará "WCAG".

Si no logra ingresar a alguno de los enlaces, comuníquese a: certification@accessibilityassociation.org

B. Recursos de preparación para el examen de la IAAP

Quienes apliquen al examen, pueden estudiar desde cualquier recurso disponible para prepararse para el examen. La IAAP enumera una serie de recursos para la preparación del examen WAS, algunos son gratuitos y otros se pueden comprar. Puede encontrar estos recursos en la [página web de la IAAP Preparación para el examen WAS \(en inglés\)](#).

Quienes quieran tomar el examen, deben revisar cada sección del [esquema de contenido de WAS \(en inglés\)](#) para determinar donde tienen más experiencia, donde tienen algún conocimiento, e identificar secciones que son menos familiares, donde pasarán la mayor parte de su tiempo estudiando para prepararse para el examen WAS. Quienes estén interesados en tomar el examen también pueden utilizar esta Guía de Estudio para obtener información más detallada. Todos los elementos del examen WAS están escritos a partir del contenido incluido en esta Guía de Estudio para la certificación WAS.

C. Acerca de la certificación WAS

La certificación de Especialista en Accesibilidad Web (WAS) de la IAAP está destinada a profesionales de accesibilidad, de quienes se espera puedan realizar una evaluación de accesibilidad sobre contenido existente, de acuerdo a los estándares y lineamientos técnicos y que puedan, además, proporcionar información detallada sobre recomendaciones de remediación.

El examen WAS permite a las personas certificar sus habilidades profesionales en la disciplina especializada de accesibilidad web. Las personas que aprueben el Certificado de Profesional en Competencias Básicas de Accesibilidad (CPACC) y el Especialista en Accesibilidad Web (WAS) obtendrán la certificación de nivel superior llamada Profesional Certificado en Accesibilidad Web (CPWA).

La certificación WAS acredita la competencia técnica de nivel intermedio de experiencia en diseño, desarrollo, implementación y evaluación de contenido, proyectos y servicios accesibles basados en la web. Este examen no está destinado a principiantes o aquellos sin experiencia en remediación o identificación de errores de accesibilidad en el código. El examen WAS no pretende ilustrar o evaluar la capacidad de escribir código. El conocimiento de programación HTML por sí solo no proporciona los antecedentes necesarios para lograr la certificación WAS con éxito. Se requiere experiencia práctica y conocimiento de elementos de código programático, estándares WCAG 2.1 y de las implicaciones contextuales para las personas usuarias finales de la tecnología de apoyo.

Se espera que los Especialistas en Accesibilidad Web conozcan y utilicen las tecnologías relevantes, no solo sean conscientes de ellas. Los campos relevantes para la designación como Especialista en Accesibilidad Web incluyen:

- Crear contenido web accesible,
- identificar errores/problemas de accesibilidad,
- remediar (arreglar) errores de accesibilidad.

La accesibilidad web se refiere a la práctica inclusiva de hacer que la web sea utilizable por personas con y sin discapacidad.

Resultados esperados del examen WAS:

En el examen se demostrarán los siguientes conocimientos:

- Comprensión de los principios generales de accesibilidad de las Tecnologías de la Información y las Comunicaciones (TIC), basados en las necesidades de las personas con discapacidad y cómo usan Internet, las herramientas que pueden usar y la creación o apoyo de un ecosistema TIC accesible.
- Cómo crear contenido digital accesible.
- Principios generales sobre cómo implementar actividades/proyectos de accesibilidad TIC para personas con discapacidad.
- Cómo contribuir a la creación de una sociedad inclusiva.

Información adicional

- [Página de inicio de la IAAP \(en Inglés\).](#)
- [Información general sobre la certificación IAAP \(en Inglés\).](#)

- [Resumen del contenido del examen WAS \(en Inglés\).](#)
- [Preguntas frecuentes sobre WAS \(en Inglés\).](#)
- [Recursos de preparación WAS \(en Inglés\).](#)

D. Resumen del contenido del examen WAS

Creación de soluciones web accesibles (40% del examen)

1. Pautas, principios y técnicas para cumplir con los criterios de éxito (incluidas las WCAG, WAI-ARIA, ATAG, conceptos básicos, limitaciones de las pautas específicas, principios, y técnicas, qué es normativo vs. no normativo; lo que se incluye en los diferentes niveles (A, AA, AAA))
2. Conocimientos básicos de programación (a nivel conceptual; principios y conceptos relacionados con la programación; el impacto de prácticas de codificación específicas en soluciones web vs. escribir código específico)
3. Garantía de la calidad de la accesibilidad (es decir, asegurar la calidad de la accesibilidad en todo el ciclo de vida del desarrollo, la diferencia y la superposición entre la experiencia de la persona usuaria y accesibilidad)
4. Tecnologías compatibles con la accesibilidad (incluidas las tecnologías de apoyo y funciones de accesibilidad; una combinación de tecnologías de apoyo y agente de personas usuarias; decisiones de diseño en la elección de tecnologías que soporten accesibilidad. Por ejemplo, no elegir Flash cuando otra opción tiene un mejor soporte de accesibilidad, las diferencias de soporte y comportamientos de las tecnologías de apoyo, diferencias en el soporte táctil cuando un el lector de pantalla está activado o desactivado)
5. Controles estándar versus controles personalizados (por ejemplo, usar los estándares cuando sea posible, y si se usan los personalizados, utilizar las mejores prácticas de WAI-ARIA)
6. Aplicaciones de página única (por ejemplo, control de enfoque, compatibilidad en retrasos para el lector de pantalla y AJAX, anuncios en vivo)
7. Estrategias de las personas con discapacidad en el uso de soluciones web (por ejemplo, navegación de personas usuarias de lectores de pantalla, encabezados y puntos de referencia, estrategias de manejo, métodos de preferencia de las personas usuarias versus métodos específicos del sitio web, usar teclado versus ratón)

Identificar problemas de accesibilidad en soluciones web (40% del examen)

1. Problemas de interoperabilidad y compatibilidad (por ejemplo, funciona con JAWS, Chrome, Safari, etc.).
2. Identificar lineamientos y principios relacionados a los errores de accesibilidad (incluyendo WCAG, WAI ARIA, ATAG, conceptos básicos, limitaciones de las pautas, principios y técnicas, qué es normativo vs. no normativo; lo que se incluye en los diferentes niveles (A, AA, AAA))
3. Pruebas con tecnologías de apoyo (por ejemplo, navegación con lectores de pantalla, encabezados y puntos de referencia, magnificador de pantalla, alto contraste, uso del teclado frente al uso del ratón)
4. Pruebas con personas usuarias finales (por ejemplo, baja visión, cognitiva, móvil/táctil)
5. Herramientas de prueba para la web (herramientas tanto automáticas como manuales, es decir, qué son y cuáles son sus limitaciones; por ejemplo, pruebas unitarias, herramientas basadas en

navegador, indexadores web, bookmarklet, herramientas automatizadas utilizadas para monitorear el sitio frente a herramientas externas)

Corrección de errores en soluciones web (20% del examen)

1. Nivel de severidad y priorización de los errores (por ejemplo, costo-beneficio, riesgo legal, impacto en personas usuarias, cuál es el problema, en qué enfocarse primero)
2. Recomendar estrategias y/o técnicas para solucionar errores (es decir, la mejor solución, una solución útil, factibilidad de la solución, reparación vs. rediseño, cómo arreglarlo)

1. Creación de contenido web accesible

1.1. Comprender e interpretar la accesibilidad. Especificaciones y técnicas.

1.1.1. Resumen

Esta competencia se centra en diseñar y crear contenidos web de acuerdo con las siguientes especificaciones de accesibilidad de la W3C:

- Pautas de Accesibilidad del Contenido Web (WCAG) 2.1:
 - [WCAG 2.1 \(normativa\) \(en inglés\)](#).
 - [Comprender las WCAG 2.1 \(no normativas\) \(en inglés\)](#).
 - [Técnicas para WCAG 2.1 \(no normativas\) \(en inglés\)](#).
 - [Cómo cumplir con WCAG 2.1 \(no normativa\) \(en inglés\)](#).
- Aplicaciones de Internet Enriquecidas Accesibles (WAI-ARIA) 1.1:
 - [WAI-ARIA 1.1 \(normativa\) \(en inglés\)](#).
 - [WAI-ARIA 1.1 El Modelo de Roles \(no normativa\) \(en inglés\)](#).
 - [Prácticas de autor de WAI-ARIA \(no normativa\) \(en inglés\)](#).
 - [Nombre accesible y cálculo de descripción 1.1 \(no normativo\) \(en inglés\)](#).
- Pautas de accesibilidad para las herramientas de creación de contenido (ATAG) 2.0:
 - [ATAG 2.0 \(normativa\) \(en inglés\)](#).
 - [Implementando ATAG 2.0 \(no normativa\) \(en inglés\)](#).

1.1.2. El W3C

[El W3C \(en inglés\)](#) es un organismo de estándares web reconocido internacionalmente que identifica sus estándares de especificación técnica aprobados como "Recomendaciones W3C" (como HTML, CSS, etc.). El consorcio tiene varias especificaciones de accesibilidad que han alcanzado el status de Recomendación del W3C, incluidas WCAG, ATAG y WAI-ARIA. Otras recomendaciones del W3C relacionadas con la accesibilidad, como las pautas de Accesibilidad del Agente de Usuario (UAAG), están fuera del alcance de la certificación IAAP WAS.

1.1.3. Pautas de accesibilidad para el contenido web (WCAG) 2.1

El estándar internacional, WCAG 2.1, cubre una amplia gama de recomendaciones para hacer que el contenido web sea más accesible. Seguir estas pautas hará que el contenido sea más accesible para una gama más amplia de personas con discapacidades, incluidas las adaptaciones para ceguera y baja visión, sordera y distintos grados de pérdida auditiva, movimiento limitado, discapacidades del habla, fotosensibilidad y combinaciones de estas, y algunas adaptaciones para personas que enfrentan barreras en el aprendizaje y discapacidades cognitivas; pero no abordará absolutamente todas las necesidades de los usuarios con estas discapacidades. Las pautas WCAG son independientes del dispositivo, abordan la accesibilidad del contenido web en computadoras de escritorio, portátiles, tablets y dispositivos móviles. Seguir estas pautas también suele hacer que el contenido web sea más utilizable para las personas usuarias en general.

Las Pautas de Accesibilidad al Contenido Web (WCAG) se desarrollan a través del proceso dispuesto por el W3C en cooperación con personas y organizaciones de todo el mundo, compartiendo el objetivo de proporcionar un único estándar compartido para la accesibilidad del contenido web que satisfaga las necesidades de personas, organizaciones y gobiernos a nivel internacional. Los documentos WCAG explican cómo hacer que el contenido web sea más accesible para las personas con discapacidad.

1.1.4. Resumen del W3C sobre las Pautas de accesibilidad para el contenido web (WCAG)

El documento WCAG está organizado en principios, pautas y criterios de éxito. Los cuatro principios POUR (por sus siglas en inglés) son:

- **Perceptible:** La información y los componentes de la interfaz de usuario deben ser presentados en formas que las personas con discapacidad pueden percibirlos (incluyendo ceguera, baja visión, sordera y distintos grados de pérdida auditiva, movimiento limitado y discapacidades cognitivas).
- **Operable:** Los componentes de la interfaz de usuario y la navegación deben ser operables de distintas maneras (por ejemplo, tener funcionalidad desde el teclado).
- **Comprensible:** La información y el funcionamiento de la interfaz de usuario deben ser comprensibles.
- **Robusto:** El contenido debe ser lo suficientemente robusto como para que pueda ser interpretado de manera confiable por una amplia variedad de agentes de usuario, incluidas las tecnologías de apoyo.

WCAG 2.1 tiene 13 pautas dentro de estos principios, mientras que WCAG 2.0 tiene 12. Las Modalidades de Entrada es la nueva pauta añadida a WCAG 2.1. Esta adición ayuda a dejar en claro que los sitios que se ajustan a las WCAG 2.1 también se ajustan a las WCAG 2.0.

Cada pauta tiene al menos uno o más criterios de éxito. Cada criterio de éxito tiene asignado un nivel: A, AA o AAA. Un sitio que cumple con todos los criterios de éxito en el nivel A se dice que "se ajusta al" nivel A. Un sitio que cumple con todos los criterios de éxito tanto del nivel A y el nivel AA se ajusta al nivel AA. Un sitio que cumple con todos los criterios de éxito en todos los niveles se ajusta al nivel AAA.

La mayoría de las leyes y políticas que hacen referencia a las WCAG 2.0 o WCAG 2.1 se centran en la conformidad del nivel AA (que incluye criterios de éxito tanto del nivel A como del AA).

El W3C ha publicado técnicas que orientan en el cumplimiento de las WCAG 2.0 y WCAG 2.1. Éstas pueden ser:

- **Técnicas suficientes:** Si el contenido web cumple con las técnicas suficientes, cumple correctamente los criterios de éxito.
- **Técnicas recomendables:** Son opcionales o condicionales. Pueden representar mejores prácticas de accesibilidad u otras formas de cumplir con los criterios de éxito.
- **Técnicas de fallo** Si el contenido de la web falla en cualquier de estos, no cumple con los criterios de éxito.

Las técnicas publicadas por el W3C no son normativas, pero han sido examinadas por un grupo de profesionales de accesibilidad. Las técnicas pueden cambiar con el tiempo debido a tecnologías en evolución, mejoras en el soporte de la accesibilidad, mejores ideas, la aparición de nuevas especificaciones de accesibilidad, etc.

Para más información, ir a:

[Comprendiendo las técnicas para los criterios de éxito de las WCAG \(en inglés\).](#)

1.1.5. Versión de WCAG

Las personas que deseen realizar el examen deben comprender el historial de publicaciones de WCAG desde WCAG 2.0 en 2008, a WCAG 2.1 en 2018, y demostrar una comprensión de los cambios introducidos en WCAG 2.1.

WCAG 2.1 agregó 17 criterios de éxito a WCAG 2.0. WCAG 2.1 no cambió los 63 criterios de éxito existentes en WCAG 2.0. La última actualización de WCAG incluye cómo el contenido para nuevas tecnologías, como teléfonos móviles o tablets, debe desarrollarse o remediarse para que las personas con discapacidad puedan utilizarlos. WCAG 2.1 también incluye más criterios que se centran en personas con baja visión o con discapacidades cognitivas, que no se abordaron en WCAG 2.0.

Para más información, ir a: [Lo nuevo en WCAG 2.1 \(en inglés\)](#)

1.1.6. Aplicaciones de Internet Enriquecidas Accesibles (WAI-ARIA) 1.1

La Iniciativa de Accesibilidad Web (WAI) del W3C creó WAI-ARIA para mejorar la accesibilidad del contenido, en particular contenido dinámico, para usuarios de tecnologías de apoyo, por ejemplo, usuarios de lectores de pantalla. WAI-ARIA define atributos que se pueden agregar al HTML estándar para definir el nombre, el rol y los valores (propiedades y estados) de elementos, especialmente para widgets personalizados. Uno de los objetivos de WAI-ARIA es hacer que las aplicaciones web se comporten como componentes software. Los widgets WAI-ARIA interactúan con la API de accesibilidad del sistema operativo, proporcionando a las tecnologías de apoyo con la semántica y actualizaciones en vivo necesarias para una accesibilidad total.

ARIA permite a quienes desarrollan marcar contenido dinámico, incluidos los controles personalizados creados con AJAX, HTML y JavaScript, para mejorar su accesibilidad. Las prácticas para el autor de WAI-ARIA definen las mejores prácticas para la estructura de los widgets, los comportamientos del teclado, y el contenido dinámico. También promueven la interoperabilidad.

Algunos ejemplos de ARIA:

- aria-label
 - Invisible,

- define un nuevo nombre para el elemento, que generalmente proviene de su contenido,
 - no permite hacer clic en el texto,
 - contiene una cadena de caracteres.
- aria-labelledby
 - Se relaciona con el ID de otro elemento,
 - puede relacionar múltiples ID de diferentes elementos,
 - los textos relacionados son normalmente visibles en la página,
 - reemplaza el nombre del elemento presentado a las tecnologías de apoyo que generalmente proviene de su contenido.
 - aria-describedby
 - Relaciona el ID de otro elemento,
 - proporciona información adicional al elemento.

Para más información ir a: [Introducción a WAI-ARIA del W3C \(en Inglés\)](#).

1.1.7 Pautas de Accesibilidad para las Herramientas de Creación de Contenido. (ATAG) 2.0

Las pautas ATAG por sus siglas en inglés, requieren que las herramientas de creación de contenido (editores de HTML, sistemas de gestión de contenidos (CMS) y sitios web que permiten a las personas usuarias añadir contenido tales como blogs y redes sociales):

- 1) Tengan una interfaz de usuario accesible.
- 2) Permitan la producción de contenido accesible.

Es decir hacer que las herramientas de creación de contenido sean accesibles en sí mismas, de forma que las personas con discapacidad puedan crear contenido web, y ayudar a los autores a crear contenido web más accesible.

Para más información ir a: [Resumen de las pautas de accesibilidad para las herramientas de creación de contenido \(ATAG\) 2.0. \(en inglés\)](#)

1.1.8 Documentos normativo versus no normativos:

Los documentos "normativos" definen las prácticas de accesibilidad requeridas para cumplir con un criterio de conformidad (a una especificación).

Los "no normativos" proporcionan orientación y técnicas para interpretar y cumplir con los requisitos normativos, pero las técnicas no normativas no son obligatorias para lograr la conformidad.

Los documentos no normativos proporcionan información sobre las diferentes formas en que las tecnologías web necesitan trabajar con las herramientas de creación de contenido, los agentes de usuario y las tecnologías de apoyo.

Los documentos no normativos pueden cambiar con más frecuencia que los documentos normativos, para adaptarse a las tecnologías cambiantes y las mejores prácticas actuales.

Estudiar temas relacionados con especificaciones y técnicas: Temas que deben estudiarse y dominarse para permitirle aprobar los componentes del WCAG 2.1 del examen WAS.

- Comprender e interpretar WCAG 2.1
 - Comprender la relación entre principios, pautas y criterios de conformidad.
 - Comprender la intención, requerimiento e impacto de cada principio, pauta y criterio de conformidad.
 - Estar familiarizado con las técnicas suficientes, técnicas recomendables y errores comunes que se utilizan para definir si se cumple o no con un criterio de conformidad.
 - Comprender los 3 niveles de conformidad (A, AA y AAA)
 - Identificar el nivel de conformidad de cada criterio.
 - Comprender los tres tipos de técnicas y su proceso de comprobación por la W3C.

Para más información ir a: [Introducción a las WCAG \(en inglés\)](#)

- Comprender e interpretar WAI-ARIA 1.1
 - Comprender el propósito y el impacto de WAI-ARIA 1.1.
 - Comprender el modelo de WAI-ARIA 1.1 de nombres, roles, y valores.
 - Saber cuándo y por qué utilizar WAI-ARIA 1.1 y cuándo utilizar HTML estándar.
 - Estar familiarizado con las prácticas de creación de widgets personalizados, incluida la estructura semántica, el comportamiento del teclado, etc.

Para más información ir a:

[Clarificando WAI-ARIA de CanAdapt \(en inglés\)](#)

[WAI ARIA sitio de la Universidad de Alicante](#)

- Comprender e interpretar ATAG 2
 - Comprender cómo ATAG 2 se implementa para las herramientas de creación de contenido.
 - Comprender el significado y la intención de las dos secciones principales de ATAG 2.
 - Comprender la intención, los requisitos y el impacto de cada principio, pauta, y criterio de conformidad.
 - Distinguir entre las buenas prácticas automatizadas en las herramientas de autoría y las buenas prácticas que requieren acción por parte del autor/usuario.
 - Comprender el poder y las limitaciones de sus funcionalidades automatizadas de accesibilidad.

Para más información, ir a: [Guía del W3C para comprender e implementar las pautas de accesibilidad de herramientas de autor 2.0. \(en inglés\)](#)

- Comprender la diferencia entre documentos normativos y no normativos, información, y poder identificar cuales documentos son normativos.

1.2. Crear contenido interactivo accesible con JavaScript y AJAX

El examen WAS no contempla los detalles de la sintaxis de programación de JavaScript (una persona puede aprobar el examen sin ser un programador profesional de JavaScript). Sin embargo, quienes diseñan y desarrollan web deben conocer cómo el contenido interactivo en JavaScript y AJAX afecta a la accesibilidad. Deben poder identificar los conceptos, principios y estrategias de diseño de interacción accesible con JavaScript.

1.2.1. Compatibilidad con JavaScript en APIs de accesibilidad y tecnologías de apoyo

Los lectores de pantalla modernos y otras tecnologías de apoyo pueden procesar los resultados de los procesos de JavaScript siempre que el código JavaScript esté escrito teniendo en cuenta la accesibilidad.

No existen barreras inherentes a las propias tecnologías de apoyo que hagan que el código JavaScript sea inaccesible.

1.2.2. Estudiar temas relacionados con contenido interactivo en JavaScript y AJAX

Algunos de los aspectos más destacados del diseño de interacción accesible se muestran en la lista a continuación:

- **Definir el foco:** Cuando JavaScript cambia el foco visual (por ejemplo, al activarse un cuadro de diálogo), se debe utilizar JavaScript para gestionar el foco del teclado y que siga el foco visual.
- **Utilizar HTML semántico:** HTML define un conjunto de elementos, atributos y valores para los atributos. Estas características tienen significados semánticos específicos que los agentes de usuario intentan procesar de diferentes maneras.
- **Tener en cuenta el orden del DOM y la probable ubicación del foco al agregar o actualizar el contenido de forma dinámica:** Cuando se agrega o modifica contenido en una página, se realiza generalmente después del punto de foco actual porque es menos probable que los usuarios de lectores de pantalla naveguen hacia atrás en el DOM que hacia adelante, lo que hace que se pierdan la mayoría de los cambios en posiciones anteriores en el DOM.

- **Crear controladores de eventos independientes del dispositivo:** Los controladores de eventos de JavaScript (event handlers) deben ser independientes del dispositivo. La funcionalidad debe estar disponible con el teclado, ratón, pantallas táctiles, voz, etc.
- **Crear widgets de JavaScript accesibles:** Diseñar patrones y ejemplos de widgets comunes.
- **Simplificar los eventos:** Los botones y otros elementos interactivos generalmente tienen solo un tipo de evento asociado. Los elementos de eventos múltiples son más difíciles de hacer accesibles y de comprender para los usuarios. Por ejemplo, un elemento de un menú debe actuar como un enlace o como un botón que expande un submenú. Codificar un elemento de un menú para expandir el menú al pasar el ratón y activar un enlace al hacer clic es problemático para los usuarios de teclado, usuarios de pantallas táctiles y usuarios de lectores de pantalla móviles basados en gestos.

Para más información, ir a: [Buenas prácticas de accesibilidad CSS y JavaScript](#)

1.3. Integrar la accesibilidad en el proceso de control de calidad.

Planear - Crear - Probar (PCT). Estas son las tres tareas a realizar durante todo el proceso de desarrollo de un nuevo sitio web o su diseño, una nueva característica o la remediación del mismo. Los expertos en accesibilidad y las personas con discapacidad deben formar parte de todas estas etapas para asegurar la calidad y usabilidad del producto.

La accesibilidad debe integrarse en todo el ciclo de vida del producto. Los detalles pueden variar de un equipo de desarrollo web a otro, pero el ciclo de vida del producto debe incluir concepto, requisitos, diseño, creación de prototipos, desarrollo, pruebas de calidad, pruebas de usuario, soporte y pruebas de regresión. El rol de cada persona en el ciclo de vida del producto debe incluir algún aspecto de accesibilidad.

Tres tareas a realizar durante el proceso de desarrollo de un nuevo sitio web o su diseño:

Las fases de planificación y diseño incluyen:

- Investigación,
- requerimientos,
- diseño de arquitectura de la información (IA, por sus siglas en inglés) y experiencia de usuario (UX, por sus siglas en inglés).

La fase de creación de contenido y componentes incluye:

- creación de un maquetado o interfaz de usuario (frontend) y programación,
- creación de contenido en formato texto,
- creación de contenido en formato multimedia.

La fase de prueba de contenido y componentes incluye:

- revisión de la programación y la interfaz de usuario,
- revisión del contenido de texto,
- revisión del contenido multimedia.

1.3.1 Estudiar temas relacionados al aseguramiento de la calidad de la accesibilidad

- Identificar las características de las disciplinas Ágiles y de Cascada como metodologías para la administración de proyectos, así como sus diferencias y compare cómo ambas tecnologías abordan el control de la calidad de la accesibilidad. La metodología Ágil implica realizar pruebas en fases mientras que la metodología de Cascada propone que el software se pruebe una sola vez.
- Comprender los beneficios de diseñar contenido digital teniendo en cuenta la accesibilidad en lugar de corregirlo después durante una remediación.
- Distinguir y diferenciar entre las disciplinas de Accesibilidad y Diseño de Experiencia de Usuario y comparar las teorías de cada disciplina.
- Demostrar un entendimiento del proceso de pruebas de usuario y compararlo a las pruebas de verificación de accesibilidad (AVT).
- Comprender que la accesibilidad necesita estar integrada dentro del ciclo de vida de un producto, incluyendo concepción, requerimientos, diseño, prototipado, desarrollo, control de calidad (QA), pruebas de aceptación de usuario (UAT), soporte y pruebas de regresión.
- Identificar los roles que participan en el ciclo de vida de un producto y de qué maneras cada uno de ellos puede contribuir con la accesibilidad.

Para más información ir a: [Integrando la accesibilidad en el ciclo de vida del desarrollo en procesos ágiles y de cascada, por Irfan Ali \(en inglés\)](#)

1.4 Elegir técnicas de accesibilidad que tengan un soporte amplio y reconocido

La mayoría de los lectores de pantalla modernos tienen un adecuado soporte para las características y técnicas de accesibilidad más relevantes. Sin embargo frente a nuevas características, como algunos tipos de controles ARIA, puede ser incompleto, inconsistente o defectuoso.

Es importante diseñar considerando estándares de accesibilidad en lugar de diferencias o errores que existen entre los diferentes lectores de pantalla. En algunos casos puede ser apropiado implementar alternativas, prescindir de alguna funcionalidad o aplicar [polyfills \(en inglés\)](#). Además, reconocer que el comportamiento de los lectores de pantalla puede cambiar en

cualquier momento, pero que en contraste los lineamientos y recomendaciones son más estables en el tiempo y usualmente representan la mejor opción.

1.4.1 Estudiar temas relacionados al soporte de la accesibilidad

- Comprender la importancia de estándares de codificación, en lugar de ajustes especiales o uso de características que aplican solamente a un grupo de tecnologías.
- Comprender el concepto de mejora progresiva, que permite que la interacción básica sea posible utilizando tecnologías desactualizadas, mientras que las características más modernas de un sitio web pueden ser aprovechadas por tecnología moderna.
- Comprender la importancia de validar la accesibilidad del diseño web a través de una variedad de plataformas, navegadores y tecnologías de apoyo, no simplemente asumir que funcionarán incluso si tecnológicamente cumplen con especificaciones de accesibilidad publicadas.
- Conocer qué combinaciones de tecnología de apoyo y navegadores funcionan mejor a efectos de las validaciones y pruebas.
- Aprender a reconocer cuando una falla en accesibilidad es el resultado de un diseño deficiente versus un problema de soporte de la tecnología.
- Aprender a reconocer la diferencia entre contenido no accesible versus técnicas incompletas o defectuosas de pruebas de accesibilidad.
- Evitar tecnologías o técnicas de diseño que solamente funcionan en ciertas plataformas.
- Conocer cuándo puede ser apropiado escribir código que suplemente, corrija o reemplace errores en navegadores o tecnologías de apoyo (¡Hacer esto con mucha precaución!).

Para más información ir a: [Técnicas de la WCAG 2.1 \(en inglés\)](#)

1.5. Crear controles/widgets interactivos (estándar o personalizados) basados en las mejores prácticas de accesibilidad

A los diseñadores web a menudo les gusta superar los límites del diseño web convencional mediante la creación de controles o widgets interactivos, utilizando diseños personalizados que superan o exceden las capacidades del HTML estándar. Pero estos controles interactivos no tienen por qué convertirse en barreras, siempre y cuando se diseñen teniendo en cuenta la accesibilidad. Las técnicas de accesibilidad para controles y widgets personalizados

generalmente requieren atributos y patrones ARIA, lo que a veces puede complicar su desarrollo exitoso.

1.5.1. Estudiar temas relacionados con controles/widgets interactivos

Algunos aspectos destacados de los conceptos clave a tener en cuenta con los controles/widgets interactivos incluyen los siguientes:

- Familiarizarse con el modelo de interacción del teclado para los widgets personalizados con ARIA. Existen patrones de teclado generales, además de patrones de teclado específicos para distintos tipos de widgets.
- En muchos de los widgets, el modelo de interacción del teclado es ingresar al widget como un todo, o al elemento activo/seleccionado dentro del widget usando la tecla Tab y luego usar las teclas de flecha para navegar dentro del widget. La tecla de Tab generalmente no se usa para navegar dentro del widget. Se pueden recomendar otras combinaciones de teclas para ciertos widgets.
- Siempre que sea posible, se deben usar widgets HTML nativos en lugar de widgets personalizados con WAI-ARIA, debido a las funciones de accesibilidad integradas de los widgets HTML nativos. La implementación de widgets personalizados requiere una mayor atención a los detalles en las técnicas y patrones de codificación, y la compatibilidad con los widgets personalizados puede variar, especialmente para los widgets menos comunes.
- Al crear widgets ARIA, se debe prestar atención a la estructura semántica de los roles. Algunos tienen roles primarios o secundarios requeridos, y otros tienen atributos requeridos.
- Cuando se asigna un rol personalizado a un elemento, este anula por completo el rol nativo. Por ejemplo, `<li role= "button">` será tratado como un botón por la API de accesibilidad, no como un elemento de lista.
- Roles para describir el tipo de widget presentado, como "menu"(un menú), "treeitem" (elementos en un árbol), "slider" (control deslizante), y "progressbar" (barra de progreso).
- Roles para describir la estructura de la página web que incluye headings (encabezados), regions (regiones), y tables (tablas).
- Propiedades para describir los estados de los widgets tales como checked (verificada) en el caso de un checkbox (casilla de verificación) o expanded (expandido) en el caso de un menú desplegable.
- Propiedades para definir las regiones modificables de una página que muy probablemente tengan actualizaciones tales como las cotizaciones de las bolsas de valores, así como alguna política de interrupción para dichas actualizaciones. Un

ejemplo de ello sería que las actualizaciones críticas se presenten en un cuadro de diálogo de alerta y las actualizaciones incidentales, dentro de la página.

- En el caso de las aplicaciones web complejas o de contenido dinámico, los roles, estados y propiedades ARIA se pueden utilizar de manera que comuniquen al lector de pantalla lo que está ocurriendo en la interfaz.
- El role="application" se debe usar en contadas ocasiones ya que sobrescribe muchos de los atajos de teclado que utilizan las tecnologías de apoyo, tales como los que le permiten a los usuarios de lectores de pantalla navegar por los elementos semánticos como encabezados, puntos de referencia, tablas, etc.

Para más información ir a:

[Ejemplos de la Biblioteca de Códigos de Accesibilidad de la Universidad Deque: ARIA Widgets \(en inglés\)](#)

[Prácticas de autor WAI-ARIA del W3C 1.1 \(en inglés\)](#)

1.6. Crear aplicaciones de una sola página accesibles o SPAs (single page applications)

Los diseños de una sola página (diseños que insertan nuevo contenido, con frecuencia mediante procesos de AJAX, sin volver a cargar la página) presentan una serie de retos de accesibilidad muy particulares.

Los lectores de pantalla casi siempre reaccionan a un evento normal de carga de la página leyendo el título de la página, y en el caso de algunos lectores, leyendo un resumen de los elementos semánticos disponibles. Como ejemplo, el lector de pantalla lee: la página tiene 7 encabezados, 3 puntos de referencia y 27 enlaces. Las aplicaciones SPA no disparan eventos normales de carga. Las SPAs usan AJAX para incorporar contenido en la URL de la página actual en vez de cargar una nueva página o URL, usualmente con la intención de mejorar el desempeño y agilizar la experiencia que el usuario tiene en la aplicación web. Los lectores de pantalla usualmente no emiten anuncios cuando el contenido se carga mediante AJAX, así que los usuarios no se dan cuenta que se ha cargado nuevo contenido.

1.6.1 Estudiar temas relacionados con las SPAs.

Si el contenido AJAX se carga como resultado directo de una acción del usuario, por ejemplo, activar un botón, el usuario debe recibir una notificación de que se ha cargado nuevo contenido.

Algunos métodos que se pueden utilizar para notificar a los lectores de pantalla que se ha cargado nuevo contenido incluyen:

- Mover el foco al nuevo contenido.
- Utilizar aria-live para anunciar el contenido sin que el foco se mueva.
- Si el contenido AJAX se carga pasivamente (es decir, no como resultado directo de una acción de la persona usuaria, como activar un botón), puede no ser necesario notificar a las personas usuarias que se ha cargado nuevo contenido. Esto dependerá de la importancia y la urgencia del contenido nuevo, y de si el contenido se ha insertado arriba de la posición actual de la persona usuaria, o no.

Para más información, ir a: [Actualizaciones dinámicas, AJAX, y aplicaciones de página única, en a11yportal \(Portal de Accesibilidad\) \(en inglés\)](#)

1.7. Crear contenido web compatible con las estrategias utilizadas por personas con discapacidad para acceder al contenido web.

Las personas con discapacidad suelen usar métodos diferentes para acceder al contenido web. Pueden usar solamente el teclado, comandos de voz, u otros dispositivos de entrada alternativos. Muchas personas con discapacidad usan tecnologías de apoyo. Hay muchos tipos de tecnologías de apoyo, diseñadas para muchos tipos diferentes de discapacidades. Muchas personas ciegas, por ejemplo, usan lectores de pantalla para escuchar el contenido web, y para navegar por el contenido web siguiendo la estructura semántica (por ejemplo: etiquetas, encabezados, tablas, etc.). Crear contenido accesible requiere saber sobre diferentes estrategias, y cómo diferentes conceptos, diseños e implementaciones afectan esas estrategias.

A continuación, se presenta un resumen de algunos puntos a destacar de las estrategias y tecnologías utilizadas por personas con varios tipos de discapacidad. La lista no es exhaustiva.

1.7.1. Visión

Dependiendo de si una persona usuaria tiene algo de visión o nada, las estrategias serán bastante diferentes. Las personas con visión limitada, a menudo usan la visión que sí tienen para ayudarse a navegar y encontrar su camino en una interfaz, mientras que las personas ciegas necesitan otras estrategias para navegar. También hay diferencias entre usar o no usar interfaces interactivas táctiles.

1.7.2. Personas ciegas usuarias de lectores de pantalla

Las personas ciegas necesitan lectores de pantallas para traducir lo que se muestra visualmente en la pantalla a voz o, menos comúnmente, a braille. Dado que ellas no pueden ver la interfaz, el lector de pantalla necesita brindar mecanismos que les ayuden a comprender la estructura de la interfaz y navegarla.

Un enfoque común en lectores de pantalla es generar listas de encabezados, links, controles para formularios, etc. De esta manera, las personas usuarias pueden moverse rápidamente entre esos elementos en una página. Para que esto funcione, es importante usar el marcado semántico correcto. El soporte para elementos HTML estándar es, a menudo, bastante bueno; y la mayoría de los lectores de pantalla, además, soportan partes significativas de WAI-ARIA. Pero es necesario saber cómo soportan los lectores de pantalla y navegadores comunes el marcado que se está planeando usar.

Entre las personas que utilizan lectores de pantalla, los conocimientos y estrategias varían considerablemente. Algunas personas solo utilizan algunas teclas, como las teclas de flechas, para navegar de elemento a elemento en la página, y la tecla de Tab para ir a elementos enfocables.

En caso de los mensajes de texto, por ejemplo: en el cuestionario, una persona ciega que navega con un lector de pantalla utiliza la tecla de Tab para moverse de campo a campo. Si las instrucciones se encuentran en una etiqueta <p>, , o <div>, estas no recibirán el foco predeterminado, y el usuario probablemente no los encuentre. La mejor forma de resolver esto es usando aria-describedby para que los lectores de pantalla lean las consideraciones cuando los usuarios navegan entre los campos de formulario.

1.7.3. Personas ciegas usuarias de pantalla táctil

El modelo de interacción para usuarios sin discapacidad visual en pantallas táctiles es bastante diferente al modelo de interacción para personas ciegas en pantallas táctiles. Las personas sin discapacidad visual deslizan y activan elementos basados en su posición en la pantalla al tocarlos directamente. Las personas ciegas, por otro lado, tienen un conjunto completamente diferente de gestos disponibles para ellos cuando el lector de pantalla se activa en un dispositivo táctil. El lector de pantalla anula el método visual para interactuar y lo reemplaza por un sistema basado en gestos. Existen gestos para lectores de pantalla para avanzar a través del contenido (deslizar a la derecha), ir atrás (deslizar a la izquierda), activar el botón actual o un enlace (al tocar dos veces desde cualquier lugar de la pantalla), y así sucesivamente. Los gestos específicos varían de una marca de lector de pantalla a otra.

Se debería estar familiarizado con los lectores de pantalla más comunes, incluyendo cómo trabajan y qué partes del HTML5 y WAI-ARIA son compatibles con ellos. Algunos ejemplos de lectores de pantalla más comunes incluyen:

- [JAWS para Windows por Freedom Scientific \(en inglés\).](#)
- [NVDA por NV Access \(en inglés\).](#)
- [Narrator para Windows por Microsoft \(en inglés\).](#)
- [VoiceOver para MacOS por Apple \(en inglés\).](#)
- [ChromeVox \(en Chromebooks\).](#)
- [TalkBack \(Android\) \(en inglés\).](#)

1.7.4. Combinaciones de lectores de pantalla y navegadores

No todas las combinaciones de lectores de pantalla, navegadores y sistemas operativos trabajan de la misma manera en términos de soporte de accesibilidad. Es recomendable ejecutar pruebas con al menos dos lectores de pantalla. Entre los más utilizados se encuentran:

- JAWS,
- NVDA,
- VoiceOver.

1.7.5. Combinaciones recomendadas para garantizar una mejor compatibilidad:

- Windows:
 - JAWS y Google Chrome (y en algunos casos IE 11),
 - NVDA y Firefox,
 - Narrator y Edge.
- Mac OS:
 - VoiceOver y Safari.
- IOS:
 - VoiceOver y Safari.
- Android:
 - TalkBack y Chrome.

1.7.6. Consideraciones en el uso del lector de pantalla

- La diferencia entre el orden visual del contenido y el orden estructural del contenido presentado al lector de pantalla.
- Cómo trabajan el atributo de rol en WAI-ARIA y los elementos semánticos en HTML para crear la estructura de la página.
- Cómo utilizar en los lectores de pantalla listas de diferentes objetos (por ejemplo, tablas, listas, imágenes, etc.) para encontrar información importante en la pantalla.
- Cómo navegar con el teclado usando un lector de pantalla.
- Cómo ingresar información en un formulario.

Mientras navega con teclado y lector de pantalla, el analista de calidad debe verificar varios puntos:

- Los elementos deben recibir foco (enlaces, campos de entrada de datos o botones).
- La ubicación del foco es visible en la página web para las personas con visión que utilizan el teclado.
- El lector de pantalla anuncia etiquetas adecuadas o textos alternativos.
- Las funciones de los componentes, como menú y submenú, acordeón, panel de pestañas (tabpanel), carrusel, etc., se pueden navegar o ejecutar correctamente. En dichos componentes, se debe verificar la notificación de roles y estado.

Nota: debido a los distintos tipos de navegadores y lectores de pantalla (y diferentes versiones de estos), los resultados de las pruebas pueden variar.

Para más información, consultar los siguientes recursos:

[Documentación JAWS 2020 \(en inglés\).](#)

[Guía del usuario de NVDA \(en inglés\).](#)

[Guía del usuario de VoiceOver para Mac OS \(en inglés\).](#)

[Guía del usuario de ChromeVox \(en inglés\).](#)

[Guía del usuario de Android Talkback \(en inglés\).](#)

[Guía del usuario de VoiceOver de iOS \(en inglés\).](#)

1.7.7. Personas con baja visión

Las personas con baja visión también pueden usar un lector de pantalla para leer el texto en voz alta, pero a menudo usan la visión que tienen para orientarse en la interfaz. Hay diferentes estrategias para personas con baja visión:

- Uso de magnificador de texto y zoom en el navegador.
- Cambio de colores en el navegador o sistema operativo.
- Uso de herramientas de magnificación.

Quienes quieran tomar el examen deben tener una comprensión básica de las herramientas de magnificación comunes como:

- [ZoomText por Freedom Scientific \(en inglés\).](#)
- [SuperNova por Dolphin \(en inglés\).](#)

También deben comprender cómo darle soporte al aumento de tamaño del texto en celulares inteligentes y cómo operan las funcionalidades de zoom:

- [Funciones de Accesibilidad Visual en iPhone \(en inglés\).](#)
- [Magnificación en Android.](#)
- [Tamaño de fuente y pantalla en Android.](#)

Algunos conceptos importantes de comprender sobre las personas usuarias son:

- Es bastante común usar un ratón para navegar pero también usar atajos de teclado para acelerar la interacción.
- Necesitan bordes bien definidos para poder ver dónde comienzan y terminan las diferentes áreas.
- Pueden alterar los colores, el contraste y la fuente.
- Suelen combinar el zoom y el lector de pantalla.

- Existe una gran brecha entre quienes solamente necesitan el zoom integrado en el navegador, y quienes magnifican la interfaz 32 veces y utilizan los lectores de pantalla como técnica complementaria. Sus estrategias varían respectivamente.
- Puede que quienes usan lectores de pantalla también utilicen para navegar herramientas de entrada de voz, ya que muchos tienen o desarrollan debilidades físicas asociadas a actividades repetitivas, como por ejemplo el síndrome del túnel carpiano y otras limitaciones.

1.7.8 Lectura

Las personas que tienen dificultades en la lectura pueden o no utilizar tecnologías de apoyo como Kurzweil o texto a voz proporcionado por la plataforma. Otras prefieren consultar información que no sea texto y en lugar de buscar en Google, buscan en YouTube. Si existe una alternativa al texto, ésta puede ser más conveniente.

Al leer texto, es posible realizar cambios en la forma en que se presenta en el navegador. Debe conocerse cómo afecta esto a la interfaz y qué desafíos pueden surgir al realizar dichos cambios. Algunos de las modificaciones que pueden realizarse son:

- Tamaño del texto,
- color de texto y color del fondo,
- tipo de fuente.

Las personas pueden incluso definir una hoja de estilo personalizada.

Existen tecnologías de apoyo disponibles para este grupo de personas usuarias. Se basan en lectores de pantalla o en la modificación de la forma en la que el texto se presenta en el navegador.

Los lectores de pantalla para personas con dificultades en la lectura leen el texto que la persona selecciona. Este tipo de lectores de pantalla no ofrecen ayuda para la navegación, ya que la mayoría de las personas de este grupo no lo requieren. Algunos de ellos son:

- [read&write por Texthelp \(en Inglés\).](#)
- [BrowseAloud por Texthelp \(en Inglés\).](#)
- [NaturalReader por NaturalSoft \(en Inglés\).](#)
- [Kurzweil \(en Inglés\).](#)

Para aquellas tecnologías de apoyo que alteran la forma en la que se presenta el texto en el navegador, es de vital importancia que se ofrezca como texto real y no como imágenes de texto. De lo contrario, se vuelve difícil o imposible para la tecnología de apoyo modificarlo.

Algunos conceptos importantes de comprender son:

- Diferencias entre lectores de pantalla para este grupo y lectores de pantalla para usuarios ciegos.
- La importancia de usar texto real en lugar de imágenes de texto.
- Tipo de letra, color, espaciado y más: las fuentes legibles incluyen sugerencias de tipo de letra, así como el tamaño de fuente y el ancho del espaciado entre letras y palabras.

1.7.9. Cognitiva

Este grupo también es muy diverso. Incluye personas con dificultades de concentración, personas con discapacidades neuropsicológicas o con discapacidades intelectuales, entre otras.

Algunas personas con discapacidades intelectuales no pueden utilizar la interfaz incluso con tecnologías de apoyo. Sin embargo, la gran mayoría puede utilizar la interfaz aunque requieran algunas alteraciones o tecnologías de apoyo.

Las tecnologías de apoyo utilizadas, suelen ser principalmente las mismas que usan las personas con dificultades de lectura. Además, también pueden utilizar bloqueadores de anuncios, screen masking y otras herramientas similares para ayudar a mantener la concentración en la tarea actual.

También existen tecnologías de apoyo que ayudan a las personas con discapacidades cognitivas a añadir y editar datos en las páginas web, como por ejemplo, rellenar los campos de un formulario.

Algunos conceptos importantes son:

- La importancia de un diseño y una presentación limpios y sencillos: Proporcionar el control de tantos aspectos del sitio web como sea posible. Puede utilizarse CSS para darle a la persona el control de cómo es presentada la información.
- El lenguaje sencillo es clave para este grupo.
- Alternativas de contenido: Utilice también sonidos para presentar la información. Las personas difieren en la manera de procesar la información en diferentes formatos (texto, imagen, audio). Por lo tanto, ofrecer más de un formato alcanza a más personas.
- Siempre que sea posible, utilizar imágenes y contenido multimedia para complementar el texto. El uso de gráficos apropiados y claros puede ayudar a mejorar la comprensión de los materiales, pero no hay que utilizarlos en exceso y es mejor evitar los gráficos animados, ya que pueden distraer y aumentar la carga cognitiva.
- Las personas con discapacidades cognitivas suelen utilizar tecnologías de apoyo para leer y escribir.

Para más información ir a:

- [Accesibilidad cognitiva en W3C \(en inglés\).](#)

- [Hacer que el contenido sea utilizable para personas con discapacidades cognitivas y de aprendizaje del W3C \(en Inglés\).](#)

1.7.10. Motriz

Las personas que tienen discapacidades motoras son un grupo amplio. Algunas tienen parálisis completas o parciales, otras tienen dificultades con las acciones motrices finas, tal vez debido a temblores o reumatismo. Otro ejemplo son quienes necesitan controlar la interfaz con un teclado. Estas personas a menudo usan la tecla Tab para saltar entre objetos interactivos en la interfaz, como botones, campos de formulario y enlaces.

Algunas personas utilizan una solución basada en apuntar y hacer clic; que podría ser:

- Ratón común.
- Alfombrilla de ratón con filtrado de temblores.
- [SteadyMouse](#) (tecnología de apoyo con filtrado de temblores).
- Seguimiento ocular.
 - [EyeGaze](#) (tecnología de apoyo con seguimiento ocular).
 - [Tobii](#) (tecnología de apoyo con seguimiento ocular).
- Escaneo de puntos con control de tipo interruptor.
 - [Switch Control o Control por botón en iPhone, iPad y iPod Touch.](#)
 - [Switch Access o Accesibilidad con interruptores para Android.](#)
- [Head Pointer o Función de puntero de seguimiento de cabeza de Mac OS.](#)

Algunas personas usan el teclado, como quienes utilizan lector de pantalla, pero con algunas diferencias importantes:

- Las personas deben poder ver dónde está el foco; qué enlace, control de formulario o botón es enfocado mientras navegan por la página con teclado, ya que no usan un lector de pantalla que pueda anunciarles esto.
- No pueden utilizar las listas de encabezados, enlaces y otros elementos para navegar por bloques de contenido, como con lector de pantalla.
- Cuando un sitio tiene una estructura ordenada e intuitiva y es navegable con teclado (generalmente con Tab, Enter y teclas de flecha), grupos importantes de personas son beneficiados: como las que no pueden usar un ratón o aquellas con discapacidad motriz.

Además, estructurar el sitio para que sea operable a través del teclado facilita la navegación con tecnologías de apoyo (como interruptores), que simulan el movimiento con las teclas Tab o Enter.

Algunas personas utilizan control por voz. Las personas candidatas deben tener conocimientos básicos de cómo funciona.

Algunos ejemplos incluyen:

- [Voice Control o Control de voz para iOS.](#)
- [Voice Control o Control de voz para Mac OS.](#)
- [Voice Access o Acceso por voz para Android.](#)
- [Speech Recognition o Reconocimiento de voz para Windows.](#)

Algunos conceptos que es importante comprender incluyen:

- Las personas tienen diferentes dispositivos de entrada, por lo que nuestra solución web debe posibilitar su uso con diferentes métodos de entrada como:
 - Ratón,
 - teclado,
 - voz,
 - toques/gestos.
- Tener áreas cliqueables grandes, claramente visibles y fáciles de detectar es de gran importancia.
- Es clave ayudar a los usuarios a ver qué objeto está actualmente enfocado.
- La predicción y el autocompletado son muy útiles, ya que reducen el número de pulsaciones de teclas necesarias al escribir.
- Nota: Al crear páginas web que tengan diseño, apariencia, navegación y funcionamiento predecible y consistente, se evitan confusiones. Por esta razón, no deben crearse contenidos inesperados. Esto da mayor claridad a personas con discapacidades visuales, cognitivas o motoras.

Para más información ir a:

- [Discapacidades Motrices por la Universidad Deque \(en inglés\)](#)
- [Video por Level Access: Accesibilidad Digital Impacto en Usuarios: Discapacidades Motoras \(en inglés\)](#)

1.7.11. Audición

Las personas que nacen sordas suelen tener la lengua de señas como primera lengua. Esto significa que la información de texto que perciben está escrita en lo que es a menudo un segundo o tercer idioma.

Existen algunos intentos de crear tecnologías de apoyo que traducen el texto escrito a lengua de señas. Aunque esto todavía no es una práctica común, es preferible utilizar un lenguaje fácil para ayudar a este tipo de tecnologías de apoyo y a aquellas personas que no utilizan ninguna. Los íconos, las ilustraciones y las imágenes también deben usarse para reforzar la información.

Un subtítulo preciso es muy importante para el contenido de video. El uso de herramientas automáticas y aquellas que se basan en inteligencia artificial tienden a crear una traducción incorrecta del contenido hablado, por lo que no se aconseja utilizarlas como única fuente.

Algunos conceptos importantes son:

- Proporcionar alternativas, como subtítulos o transcripciones, para los medios basados en el tiempo.
- Para solo audio pregrabado, proporcionar una alternativa que presente información equivalente, como una transcripción.
- Brindar transcripción beneficia a las personas con discapacidad auditiva. Ésta puede estar disponible en la misma pantalla donde se encuentra el audio o a través de un enlace.

Para más información ir a:

[Subtítulos de video por la W3C \(en inglés\).](#)

2. Identificar errores o problemas de accesibilidad

2.1 Identificar problemas de interoperabilidad y compatibilidad

Al evaluar el contenido web para detectar errores de accesibilidad, es importante identificar aspectos de diseño que puedan interferir con los métodos, tecnologías o estrategias utilizadas por las personas con discapacidad.

2.1.1 Estudiar temas relacionados a las estrategias utilizadas por las personas con discapacidad:

Algunos de los temas a buscar incluyen:

2.1.2 Accesibilidad mediante el teclado

- Los elementos accionables (enlaces, botones, controles, etc.) reciben foco con el teclado.
- Todos los elementos enfocables deben tener un indicador visible.
- Orden lógico de navegación.
- No debe haber trampas de teclado.

Mientras se navega con teclado y lector de pantalla, verificar:

- Los elementos deben recibir foco (enlaces, entradas de datos y botones)
- Se visualiza la ubicación del foco.
- El lector de pantalla anuncia etiquetas adecuadas o textos alternativos.
- Las funciones de los componentes, como menú y submenú, acordeón, panel de pestañas (Tabpanel), carrusel, etc., se pueden navegar o activar correctamente.

2.1.3 Accesibilidad en dispositivos táctiles

- Tamaño suficientemente grande del objetivo táctil.
- Diseño y construcción para cualquier orientación de pantalla.
- Los gestos personalizados y complejos (por ejemplo, con varios dedos) deben tener un método alternativo de activación (por ejemplo, activando un botón).
- Los eventos activados por movimiento (por ejemplo, sacudir el dispositivo) deben tener un método alternativo de activación (por ejemplo, hacer clic en un botón). La persona usuaria debe poder desactivar la función de movimiento para evitar la activación accidental debido a temblores o espasmos.

Para más información, ir a: [CSS, HTML, ARIA, navegadores, tecnología de asistencia e interoperabilidad por el Grupo Paciello \(en inglés\)](#)

2.2. Determinar la conformidad a las especificaciones de accesibilidad en función de los errores encontrados.

En primer lugar, es importante comprender el estándar que se está aplicando (por ejemplo, WCAG 2.0 o WCAG 2.1 nivel AA), y distinguir entre errores que incumplen este estándar, frente a otros que son decisiones de diseño incorrectas (incluso si son errores en un estándar diferente o en un nivel diferente dentro del estándar)

2.2.1. Identificar el criterio de éxito para cada error de accesibilidad.

A menudo, el objetivo principal de una evaluación de accesibilidad es determinar si el diseño se ajusta a una especificación particular o no. El estándar en muchos entornos es WCAG 2.0 o WCAG 2.1 nivel AA. Si este es el caso, es importante poder asociar cada error de accesibilidad a uno de los criterios de éxito dentro de esa especificación. Una imagen que carece de texto alternativo, por ejemplo, se asignaría al criterio de éxito 1.1.1.

2.2.2. Diferenciar entre WCAG 2.0, WCAG 2.1

Como se mencionó anteriormente, WCAG 2.1 tiene 13 pautas y sigue una metodología de diseño centrada en el usuario y en la investigación para producir el resultado más efectivo y flexible, incluyendo las funciones de creación de contenido, soporte de agente de usuario y soporte de herramientas de creación. WCAG 2.1 tiene 12 criterios de éxito más que WCAG 2.0 en el Nivel A y el Nivel AA. Se relacionan con las discapacidades cognitivas, los dispositivos de interfaz táctil y la orientación de la pantalla. Mientras que los ejecutivos necesitan conocer WCAG 2.0 como estándares de política normativa, quienes desarrollan software tienen como objetivo implementar las técnicas de criterios de éxito de WCAG 2.1.

2.2.3. Distinguir entre errores (de criterios de éxito) y mejores prácticas opcionales.

Si el objetivo es la conformidad, no incluya todas las mejores prácticas de accesibilidad posibles en una auditoría. Si lo hace, será más difícil alcanzar el objetivo de conformidad rápidamente. Cualquier problema que no esté directamente relacionado con el objetivo de conformidad debe documentarse por separado o indicarse específicamente como no requerido para la conformidad (incluso si son buenas ideas que impactan positivamente en la accesibilidad).

2.2.4. Estudiar temas relacionados a como determinar el nivel de conformidad de las especificaciones de accesibilidad.

- Familiarícese con las especificaciones (WCAG, WAI-ARIA, y ATAG) y conozca que criterios de éxito aplican y qué nivel de conformidad.
- Prepárese para distinguir entre errores de criterios de accesibilidad versus incorrectas prácticas de accesibilidad que no están referenciadas en las especificaciones.

Para más información, ir a: [Novedades en WCAG 2.1 de W3C](#)

2.3 Probar con tecnologías de apoyo

Probar con tecnologías de apoyo (AT) permite al analista de calidad verificar problemas desde la perspectiva de personas con discapacidades que utilizan éstas para acceder a páginas web, aplicaciones web y aplicaciones de software. Esas tecnologías se categorizan en: lectores de pantalla, magnificadores de pantalla, reconocimiento de voz e incluso teclados. Es importante notar que un analista de calidad que simula el uso de AT es muy diferente de alguien que utiliza esas tecnologías todo el tiempo.

Probar con un lector de pantalla o con reconocimiento de voz, por ejemplo, permite verificar si una página es navegable a través de sus encabezados, listas o puntos de referencia. Se puede verificar si es posible acceder a campos de sus formularios o navegar a través de sus tablas así como probar si los controles de la interfaz de usuario pueden ser accedidos y activados utilizando los atajos o mecanismos que provea la tecnología de apoyo. Probar con un magnificador de pantalla como ZoomText, permite verificar si el contenido puede ser visto con diferentes niveles de contraste. Estos magnificadores también deben utilizarse para confirmar si estos atajos son funcionales cuando se navegan páginas web.

A pesar de que el teclado podría no considerarse como una AT, es esencial para usuarios de lectores de pantalla y personas que no pueden utilizar un ratón. Cuando realice pruebas de una página web con un teclado, asegúrese de que puede navegarla utilizando la tecla Tab. Acceder a los controles de tipo radio button (botón de tipo radio) o tree (árboles) requiere el uso de las teclas de flechas.

Las AT deben ser testeadas con navegadores compatibles. Hasta junio de 2020, JAWS funciona mejor con IE 11 o Chrome, NVDA funciona mejor con Firefox y VoiceOver funciona mejor con Safari. Incluso si una AT funciona con una combinación de navegadores, algunos errores podrían aparecer solamente en un navegador. Por ejemplo, JAWS no puede leer los encabezados en ciertas versiones de Internet Explorer incluso cuando dichas páginas web contienen una estructura de encabezados.

Usar diferentes navegadores es recomendable: (Internet Explorer, Chrome, Firefox, Safari, Opera) y sus lectores de pantalla compatibles (Ver "Consideraciones acerca de lectores de pantallas y navegadores").

2.3.1. Estudiar temas relacionados con las pruebas mediante tecnologías de apoyo

- Saber cómo usar lectores de pantalla para navegar por elementos tales como puntos de referencia (landmarks), encabezados (headings), tablas, formularios, etc.
- Saber cómo avanzar y retroceder a través del contenido enfocable de los lectores de pantalla (por ejemplo, usando la tecla de Tab o shift+Tab en los navegadores de escritorio).
- Saber cómo avanzar y retroceder en todo el contenido de los lectores de pantalla (por ejemplo, usando las teclas de flecha hacia arriba o hacia abajo en la mayoría de estos).
- Conocer las consecuencias de usar modos de mejora de contraste como el que provee Windows.

- Saber cómo usar el teclado solo para navegar por formularios.
- Saber cómo accionar combinaciones de teclado para navegar por varios tipos de elementos con un lector de pantalla.
- Saber cómo usar el teclado y los lectores de pantalla para navegar por los widgets personalizados de ARIA.
- Conocer sus limitaciones en el uso de la tecnología de apoyo. Si no tiene experiencia, no asuma que algo es un error si es posible que no sepa cómo usar la tecnología asistiva correctamente.
- Saber por qué es importante obtener las opiniones y comentarios de los usuarios con discapacidades que utilizan sus respectivas tecnologías de apoyo.
- Saber qué combinaciones de tecnologías de apoyo funcionan mejor y con cuales navegadores. No pruebe en combinaciones que no se recomiendan o que no son totalmente compatibles con la accesibilidad.

2.4. Impacto de la Prueba de usuario final

Algunos diseños web pueden ser difíciles de utilizar o poco intuitivos para las personas con discapacidad, incluso si el diseño cumple técnicamente las pautas de accesibilidad.

Las pautas no son exhaustivas, lo que significa que hay ciertas buenas técnicas de accesibilidad que no están representadas en las guías. Algunas razones por las cuales éstas no se encuentran listadas son:

- Es difícil verificar objetivamente el cumplimiento de la técnica.
- Los redactores de las pautas no reconocieron la necesidad de la técnica al escribirlas.
- La técnica no era necesaria (o al menos no prevista) en el momento en que las pautas fueron escritas porque las tecnologías o circunstancias que requieren la técnica son más recientes que estas.

Incluso si se puede acceder a una página web con tecnologías de apoyo y cumple con los estándares de accesibilidad, puede no ser útil para usuarios finales con discapacidades. Por ejemplo, aunque el enlace del chat es accesible, es el último enlace de una página web. Otro ejemplo podría ser si la página web está sobrecargada con imágenes que tienen atributos alt, la página aún puede ser accesible pero no muy fácil de comprender.

Accesibilidad para personas con discapacidades cognitivas significa hacer un sitio web o programa lo más claro y conciso posible. Este tipo de accesibilidad implica una coherencia de diseño de página en todo el sitio web, enlaces visibles en la misma posición y párrafos cortos. Estas técnicas no solo facilitan la accesibilidad a las personas con discapacidades cognitivas; también mejora la usabilidad para todas las personas. Si el contenido de un sitio web no se puede comprender fácilmente entonces nadie puede usarlo. Sin embargo, algunas de estas técnicas no son pautas oficiales, como la longitud de un párrafo para que sea breve.

Con avance diario de la tecnología, las pautas se vuelven obsoletas rápidamente. Por ejemplo, cuando se escribieron las WCAG 1.0, no se desarrollaron criterios o técnicas de éxito para ARIA o Silverlight, por no hablar de los dispositivos móviles. Desde entonces, se han creado las WCAG 2.1, Mejores Prácticas de Mobile Web 1.0.

2.4.1. Estudiar temas relacionados con el impacto de las pruebas de usuario final

- Comprender el valor de las pruebas de usuario realizadas por personas con diferentes tipos de discapacidad.
- Pensar en las consecuencias de ciertos tipos de errores de accesibilidad. Algunos defectos son más perjudiciales que otros.
- Considerar la usabilidad del diseño, no solo la accesibilidad o conformidad con las especificaciones.

Para más información ir a: [Involucrar usuarios en la evaluación de accesibilidad web de W3C \(en inglés\)](#)

2.5. Usar las herramientas de prueba de accesibilidad de manera efectiva.

Ninguna herramienta de software de accesibilidad puede encontrar todos los problemas de accesibilidad en un sitio web. Sin embargo, estas pueden acelerar la búsqueda de errores de accesibilidad y aumentar la precisión general cuando se complementan con una evaluación manual habilidosa del mismo contenido. Por ejemplo, una herramienta automática puede filtrar todos los textos alternativos de las imágenes de la página (para comprobar si todas tienen textos alternativos). Aun así, solo un ser humano puede determinar si los textos son adecuados para las imágenes.

Algunos aspectos claves a considerar durante un proceso de evaluación manual:

- El entorno de pruebas: evaluar individualmente las páginas elegidas (ej.: página de inicio).
- Verificar la correcta navegación y funcionalidad usando el ratón y el teclado:
 - Los elementos interactivos deben recibir el foco, como por ejemplo: enlaces, entrada de datos (inputs) o botones.
 - La ubicación del foco debe visualizarse.
 - El lector de pantalla debe anunciar etiquetas adecuadas o textos alternativos.
 - Las funciones de los componentes, como menú y submenú, acordeón, panel de pestañas (tabpanel), carrusel, etc., se pueden navegar o ejecutar correctamente. En dichos componentes debe verificarse que los roles y estados sean notificados.
- Una persona analista de calidad (tester) debería identificar dos puntos importantes en los resultados:
 - Errores de accesibilidad.
 - Resultados accesibles óptimos.

Existen muchas herramientas y marcas disponibles, algunas de las cuales se enumeran a continuación.

Nota: La siguiente lista no implica un respaldo por parte de la IAAP de ninguno de los productos o de las empresas que los producen. Estos son simplemente ejemplos de varios tipos

de software para pruebas de accesibilidad. Para obtener más opciones, consulte la Lista de Herramientas de Evaluación de Accesibilidad Web del W3C (el contenido o la precisión de esa lista no es responsabilidad de la IAAP).

2.5.1. Pruebas automatizadas

- Revisión e informes de sitio completo
 - AMP (por Level Access/SSB Bart)
 - WAVE API (por WebAIM)
 - Tenon.io
 - WorldSpace Comply (por Deque Systems)

- Análisis de páginas basadas en servidor
 - wave.webaim.org (por WebAIM)
 - Cynthia Says (por Cryptzone)
 - SiteImprove Accessibility (por SiteImprove)

- **Herramientas de desarrollo/control de calidad basadas en el navegador (una página a la vez)**
 - aXe browser add-on (por Deque)
 - Google accessibility developer tools
 - WAVE browser add-on (por WebAIM)
 - Bookmarklets de JavaScript desarrollados para probar criterios de accesibilidad específicos o exponer ciertos tipos de marcado (por varias empresas e individuos)
 - AInspector para Firefox (por University of Illinois at Urbana-Champaign)
 - ANDI
 - ARC Toolkit por TPG
 - MS Accessibility Insights
 - Google Chrome Lighthouse

- **Pruebas unitarias durante el desarrollo**
 - aXe API (con integración a Selenium, etc., por Deque)
 - Tenon API
 - WorldSpace Attest (por Deque)

- **Pruebas de integración antes del despliegue en entornos de prueba o productivos (deploy)**
 - aXe API (con integración a Selenium, etc., por Deque)
 - Tenon API

2.5.2 Herramientas de pruebas manuales

- **Pruebas guiadas basadas en las heurísticas**
 - WorldSpace Assure (por Deque)
 - JAWS Access
- **Herramientas de desarrollo y complementos (add-ons) del navegador**
 - La función de “Inspeccionar” en las Herramientas de desarrollo en los navegadores
 - Web Developer Toolbar - Barra de herramientas para desarrolladores web - (por Chris Pederick)
- **Lectores de la API de accesibilidad**
 - Accessibility Viewer - Visor de accesibilidad - (para Windows, por Paciello Group)
 - XCode Accessibility Inspector - Inspector de accesibilidad de XCode - (MacOS, de Apple)

- **Simuladores**
 - Color Oracle (para simulación de daltonismo, por Bernard Jenny)
 - Simulador de visión No Coffee (baja visión y otras condiciones, por Aaron Leventhal)

- **Herramientas de accesibilidad multipropósito**
 - Web Accessibility Toolbar - Barra de herramientas de accesibilidad web - (por Paciello Group)

- **Herramientas de un solo propósito**
 - Headings Map - Mapa de Encabezados - (por Jorge Rumoroso)
 - PEAT (Photosensitive Epilepsy Analysis Tool) - herramienta de análisis de epilepsia fotosensible (por Trace R&D Center)
 - Contrast Checker - Comprobador de contraste - (por WebAim)
 - Color Contrast Analyser - Analizador de contraste de color - (del Grupo Paciello)

2.5.3. Estudiar temas relacionados con las herramientas de pruebas de accesibilidad:

- Conocer las fortalezas y limitaciones de las herramientas de prueba automatizadas.
- Diferenciar entre los tipos de errores de accesibilidad que se pueden encontrar con herramientas automatizadas y aquellos que requieren pruebas manuales.
- Comprender cómo se pueden utilizar las herramientas de software de accesibilidad en varias etapas del proceso de desarrollo web (por ejemplo, diseño/desarrollo/prueba).
- Estar familiarizado con los tipos de herramientas de software disponibles (escaneo de todo el sitio, análisis basado en el servidor, pruebas unitarias, pruebas de integración, herramientas de desarrollo de navegador, complementos del navegador, simuladores, pruebas manuales guiadas, etc.).
- Saber generar y traducir resultados en análisis utilizables, priorización, y categorización, junto con cómo diferenciar los informes técnicos de informes comerciales.

3. Remediar (arreglar) problemas de accesibilidad

3.1. Priorizar los problemas de accesibilidad en función del nivel de severidad.

Al administrar proyectos de accesibilidad, aborde primero los errores de accesibilidad más importantes. Comience con las funciones principales del sitio web. Si hubiera un carrito de compras, este sería considerado una función central. Cada paso de la experiencia de compra debe ser accesible: por ejemplo, buscar productos, leer acerca de los productos, agregar el producto al carrito, revisar el carrito, ingresar información de la cuenta, ingresar la información del pago y recibir la confirmación de la compra. Las funciones no principales del sitio pueden esperar hasta que se haya abordado la accesibilidad de las funciones principales.

3.1.1. Estudiar temas relacionados con la priorización de errores de accesibilidad en función del nivel de severidad

- Identificar el error: identificar el error de accesibilidad ya sea, en el estilo, marcado o funcionalidad del contenido.
- Identificar el impacto al usuario: asociar el error con el impacto sobre la accesibilidad para la persona usuaria. Diferenciar entre errores menores que se pueden solucionar a través de métodos avanzados de accesibilidad y errores completamente bloqueantes.
- Identificar el Riesgo Legal y el Costo-Beneficio: Determinar si el error identificado es un riesgo legal o una potencial mejora de usabilidad. Considerar factores como la visibilidad y la frecuencia. Determinar el costo-beneficio de la remediación, de acuerdo a la posibilidad o imposibilidad de las personas usuarias para completar el flujo en cuestión.
- Determinar el Nivel de Esfuerzo asociado con la remediación de errores: diferenciar entre cambios de estilo, marcado y funcionalidad.
- Priorizar: utilice el impacto al usuario, el riesgo legal, el costo-beneficio y el nivel de esfuerzo para determinar la severidad, identificar las fallas más expuestas y priorizar todos los errores.

Para más información, ir a: [Priorización del contenido web de MSU para la revisión y corrección de la accesibilidad \(en inglés\)](#)

3.2. Recomendar estrategias y/o técnicas para solucionar problemas de accesibilidad.

Recomendar estrategias de remediación requiere el conocimiento de cómo crear contenido accesible (competencia 1), la capacidad de identificar errores de accesibilidad (competencia 2), y la sabiduría para elegir una técnica de remediación apropiada para las circunstancias, teniendo en cuenta todas las limitaciones prácticas de las situaciones del mundo real.

La tarea de hacer accesible un sitio web puede ser simple o compleja. Hacerlo accesible depende de muchos factores, como el tipo de contenido, el tamaño y complejidad del sitio, las herramientas de desarrollo y su entorno. Recomendar las mejores estrategias y técnicas que reflejen condiciones particulares (tiempo, dinero, el estado actual del desarrollo, Sistemas de gestión de contenidos -CMS por sus siglas en inglés- elegido, etc.) puede ayudar tanto a las personas usuarias como a las que desarrollan a encontrar la forma más eficiente de hacer que el sitio web sea accesible.

3.2.1. Estudiar temas relacionados a la recomendación de estrategias y/o técnicas para solucionar errores de accesibilidad

- Caracterizar y diferenciar entre la solución ideal/mejor y una solución “suficientemente buena”, respetando el proyecto particular, su entorno, grupos a los que está dirigido y recursos.
- Demostrar el entendimiento entre la solución del error en particular y un rediseño completo de la página web.
- Demostrar la habilidad de distinguir la viabilidad de una solución particular en diferentes contextos.
- Demostrar el conocimiento de consejos prácticos y sencillos, que conducen a una mejor accesibilidad web.
- Comunicar el propósito, el abordaje y la estrategia de remediación.
- Asegurar que las partes interesadas estén incluidas, informadas y capacitadas sobre las recomendaciones de implementación.
- Considerar el uso de modelos de madurez y herramientas para ilustrar el progreso y sostenibilidad.

Para más información, ir a: [Planificación y gestión de la accesibilidad web del W3C \(en inglés\)](#)

- Esta Guía de estudio del especialista en accesibilidad web (WAS) de la IAAP es provista como información general sobre el estado actual y el futuro de las mejores prácticas de accesibilidad web. Este documento pretende identificar niveles intermedios de conocimiento para aquellos que trabajan en accesibilidad web, como una parte de la profesión de accesibilidad.
- El Equipo de Certificación de IAAP puede ser contactado por correo electrónico en certification@accessibilityassociation.org.