



Universidad
Zaragoza

Trabajo Fin de Grado

Evaluación de accesibilidad web en Drupal CMS con
aportaciones técnicas a la comunidad Drupal

Web accessibility evaluation in Drupal CMS with
technical contributions to the Drupal community

Autora

Nerea Salamero Labara

Directores

Ricardo Julio Rodríguez Fernández

Iván Pacheco Encuentra

Grado en Ingeniería Informática
ESCUELA DE INGENIERÍA Y ARQUITECTURA
2026

AGRADECIMIENTOS

A mis padres, mi hermano y mi abuela. Por su apoyo constante, por creer en mí durante todos estos años aún cuando ni yo me creía capaz y por todas las velas encendidas en días de examen o entregas, gracias de corazón.

A mis amigos, por escucharme cada vez que lo he necesitado (aunque no entendieran nada) y por hacerme muchísimo más amena esta etapa universitaria.

A Iván, Virginia, José y todo el departamento de Drupal de NTT DATA Zaragoza, por su consejo y acompañamiento a lo largo del desarrollo de mi Trabajo de Fin de Grado.

RESUMEN

La accesibilidad web ha evolucionado a una velocidad vertiginosa a lo largo de las últimas décadas, impulsada en gran medida por el aumento del uso de Internet en la vida diaria, además de la digitalización de la mayoría de servicios, tanto públicos como privados. En un inicio, la accesibilidad web se centraba únicamente en implementar soluciones técnicas aisladas, pero actualmente ha adquirido un carácter fundamental en el diseño y desarrollo de software. Las directrices del *World Wide Web Consortium* son el principal referente en la estandarización de la accesibilidad web. Los criterios de accesibilidad se han visto ampliados gracias a las versiones 2.0, 2.1 y 2.2 de las recomendaciones WCAG (del inglés *Web Content Accessibility Guidelines*), incorporando progresivamente aspectos relacionados con dispositivos móviles y experiencia de usuario.

En el presente Trabajo de Fin de Grado se evalúa la accesibilidad web de un sitio creado con Drupal CMS y el tema Mercury, analizando el cumplimiento de las pautas WCAG en el nivel AA mediante una combinación de auditorías automáticas y manuales. El trabajo abarca la configuración de un entorno de prueba completo con componentes habituales en aplicaciones web, la realización de auditorías automáticas con las herramientas especializadas Lighthouse, axe DevTools y WAVE y de auditorías manuales mediante navegación por teclado y lector de pantalla. Una vez obtenidos los resultados de la auditoría, se desarrollan diversas aportaciones técnicas a la comunidad de código abierto de Drupal, que se publican en la página oficial de la comunidad.

ABSTRACT

Web accessibility has evolved at a dizzying pace over the last few decades, especially due to the increased use of the Internet in daily life, as well as the digitalisation of most services, both public and private. Initially, web accessibility focused solely on implementing isolated technical solutions, but it has now become fundamental to software design and development. The World Wide Web Consortium (W3C) guidelines are the main reference point for web accessibility standardisation. Accessibility criteria have been expanded thanks to versions 2.0, 2.1, and 2.2 of the WCAG (Web Content Accessibility Guidelines) recommendations, incorporating aspects related to mobile devices and user experience.

This Bachelor's Thesis evaluates the web accessibility of a site created with Drupal CMS and the Mercury theme, analysing compliance with the WCAG AA guidelines through a combination of automated and manual audits. The work encompasses setting up a complete test environment with common web application components, performing automated audits using the specialised tools Lighthouse, axe DevTools, and WAVE, and conducting manual audits using keyboard navigation and a screen reader. Once the audit results are obtained, various technical contributions are developed for the Drupal open-source community, which are published on the official community website.

Índice

Lista de Figuras	VII
-------------------------	------------

Lista de Tablas	VIII
------------------------	-------------

1. Introducción y objetivos	1
1.1. Motivación	1
1.2. Objetivo y alcance del proyecto	1
1.3. Estructura del documento	2
2. Marco teórico	3
2.1. Accesibilidad web	3
2.1.1. Tipos de discapacidad	4
2.1.2. Tecnologías de asistencia	4
2.2. Web Content Accessibility Guidelines	5
2.2.1. Principios fundamentales	5
2.2.2. Versiones	6
2.2.3. Niveles de cumplimiento	6
2.3. ISO	6
2.4. Auditoría de accesibilidad	7
2.5. Drupal CMS	7
3. Auditoría de accesibilidad web	9
3.1. Auditoría automática de accesibilidad	10
3.2. Auditoría manual de accesibilidad	13

3.3. Conclusiones	18
4. Aportación técnica a la comunidad Drupal	19
4.1. Aportación 1. Texto alternativo en el enlace del logotipo	20
4.2. Aportación 2. Contraste insuficiente en campos de formulario	21
4.3. Aportación 3. Elementos de navegación sin aria único	22
5. Conclusiones y trabajo futuro	24
Bibliografía	26
Anexos	30
A. Organización del proyecto	31

Lista de Figuras

3.1. Navegación por teclado de la página <i>Home</i>	14
3.2. Navegación por teclado de la página <i>Contacto</i>	14
3.3. Navegación por teclado de la página <i>Service</i>	15
3.4. Navegación por teclado de la página <i>Recursos</i>	15
3.5. Navegación por teclado de la página <i>Base Canvas</i>	16
4.1. Contraste insuficiente en el campo “Mensaje” del formulario antes del <i>fix</i>	22
4.2. Contraste corregido en el campo “Mensaje” del formulario tras el <i>fix</i> .	22
A.1. Diagrama de Gantt del proyecto	31

Lista de Tablas

3.1. Clasificación de incidencias detectadas durante la auditoría automática	10
3.2. Puntuaciones obtenidas con Lighthouse	10
3.3. Incidencias detectadas con axe DevTools	11
3.4. Incidencias detectadas con WAVE	12
3.5. Incidencias comunes detectadas por las tres herramientas automáticas .	13
4.1. Opciones de aportaciones técnicas identificadas a partir de las auditorías y de las ya reportadas en drupal.org	19
A.1. Esfuerzo invertido en las diferentes etapas del proyecto	32

Capítulo 1

Introducción y objetivos

1.1. Motivación

El acceso a la información y a los servicios digitales se ha convertido en un elemento esencial del día a día de la sociedad. La digitalización de servicios públicos, educativos y comerciales ha transformado internet en un elemento del que depende una gran parte de la población. Sin embargo, este proceso no siempre se ha visto adaptado, ya que muchos sitios web siguen presentando barreras que impiden o dificultan su uso a personas con algún tipo de discapacidad. Para este colectivo, un sitio web inaccesible supone una barrera real de integración en la vida diaria. Debido a esto, la accesibilidad web debe dejar de ser algo opcional y convertirse en un requisito de inclusión.

En este contexto, los sistemas de gestión de contenidos (del inglés *Content Management System*, CMS) tienen un papel relevante ya que son la base sobre la que se construyen muchos sitios web. Drupal es uno de los CMS de código abierto más utilizados y, a pesar de tener la accesibilidad web como uno de sus principios, el lanzamiento de Drupal CMS y de su sistema de componentes visuales Drupal Canvas abre la cuestión de si este nuevo entorno cumple los requisitos de accesibilidad web básicos.

1.2. Objetivo y alcance del proyecto

El presente Trabajo de Fin de Grado tiene como objetivo principal evaluar la accesibilidad web [7, 22] en entornos desarrollados con Drupal CMS [12], analizando el cumplimiento de las pautas WCAG (del inglés *Web Content Accessibility Guidelines*) [40] y detectando posibles barreras de accesibilidad presentes en diversos componentes de interfaz y elementos interactivos.

Para llevar a cabo este análisis, se ha desarrollado un sitio web completo utilizando

Drupal CMS y su sistema de componentes visuales, Drupal Canvas [11], incorporando diferentes elementos comunes en aplicaciones web, como formularios, tablas, galerías multimedia o acordeones. Sobre este entorno se han realizado pruebas de accesibilidad automáticas y manuales, utilizando herramientas especializadas y técnicas de evaluación basadas en las recomendaciones del *World Wide Web Consortium* (W3C).

Además del análisis de accesibilidad, el proyecto incluye aportaciones a la comunidad de código abierto de Drupal. Estas consisten en la documentación y corrección de incidencias de accesibilidad detectadas durante las auditorías, así como en el desarrollo o revisión de incidencias ya reportadas en la comunidad Drupal.

1.3. Estructura del documento

La presente memoria se divide como se explica a continuación. El Capítulo 2 presenta el marco teórico del proyecto, incluyendo los conceptos de accesibilidad web, las pautas WCAG y las características de Drupal CMS. En el Capítulo 3 se describe el proceso de configuración del sitio web y se detallan las pruebas de accesibilidad realizadas, tanto automáticas como manuales, así como los resultados obtenidos y un análisis de las principales incidencias encontradas. El Capítulo 4 expone las aportaciones técnicas desarrolladas para la comunidad Drupal. Por último, el Capítulo 5 presenta las conclusiones del trabajo, así como las posibles líneas de trabajo futuro.

Por último, en el Anexo A se muestra la organización del proyecto, así como su planificación en forma de diagrama de Gantt [18] y una tabla con las horas invertidas en este trabajo.

Capítulo 2

Marco teórico

Este capítulo presenta los conceptos clave en los que se basa el trabajo. Para comenzar, se revisan los conceptos de accesibilidad web, así como los tipos de discapacidad a tener en cuenta en la accesibilidad web y las diferentes tecnologías de asistencia que facilitan la navegación web. A continuación, se describen las WCAG, sus principios, versiones y niveles; las normas ISO; y los tipos de auditorías de accesibilidad web. Por último, se realiza una introducción a Drupal y a su gestor de contenido, Drupal CMS.

2.1. Accesibilidad web

La accesibilidad web hace referencia al diseño y desarrollo de sitios, contenidos y aplicaciones digitales de manera que puedan ser utilizados por todas las personas, incluyendo a aquellas con algún tipo de discapacidad. Su objetivo principal es garantizar el acceso equitativo a la información y a los servicios disponibles en internet, eliminando aquellas barreras que dificulten la interacción de los usuarios con los contenidos.

Actualmente, la accesibilidad web representa un aspecto fundamental a la hora de desarrollar software y diseñar interfaces. La importancia de la accesibilidad web se ve reflejada en el número de personas potencialmente afectadas. Según la Organización Mundial de la Salud, aproximadamente el 16 % de la población mundial presenta algún tipo de discapacidad, alcanzando un total aproximado de 1300 millones de personas [36]. Además, existen otras situaciones que pueden derivar en que otras muchas personas experimenten limitaciones temporales o situacionales, como lesiones, envejecimiento o dificultades derivadas del entorno de uso [7].

En Europa, la Ley Europea de Accesibilidad (Directiva UE 2019/882) obliga a que todos los servicios digitales sean accesibles para todas las personas, forzando el cumplimiento de estándares como WCAG 2.1. En España, esta ley se refuerza con

el Real Decreto 1112/2018, que incluye al sector privado en la obligatoriedad de la accesibilidad web [2, 3].

2.1.1. Tipos de discapacidad

Existen diferentes tipos de discapacidad que deben tenerse en cuenta a la hora de trabajar la accesibilidad de un sitio web [22].

Discapacidades visuales. Incluyen la ceguera total, la baja visión y el daltonismo. Las personas con discapacidad visual pueden tener dificultades para percibir imágenes, colores, textos pequeños o elementos gráficos. Para facilitar su uso, se utilizan alternativas textuales, contrastes adecuados, compatibilidad con lectores de pantalla y posibilidad de hacer zoom en el contenido.

Discapacidades auditivas. Estas discapacidades afectan a personas con pérdida parcial o total de la audición. Estas personas pueden encontrar dificultades para acceder a contenidos multimedia basados en sonido, como pueden ser los archivos de vídeo o audio. Para solucionarlo, se hace uso de subtítulos, transcripciones y lenguaje de signos en los contenidos audiovisuales.

Discapacidades motoras. Comprenden aquellas limitaciones físicas que dificultan el uso preciso del teclado, ratón o de dispositivos táctiles. Para facilitar la navegación, estos usuarios pueden utilizar teclados adaptados, pulsadores o sistemas de navegación alternativa, por lo que es importante que los sitios web puedan ser completamente navegados mediante teclado y que dispongan de áreas de interacción suficientemente amplias.

Discapacidades cognitivas. Incluyen dificultades relacionadas con la memoria, el aprendizaje, la concentración o la comprensión. Para mejorar la accesibilidad cognitiva es importante utilizar un lenguaje claro, estructuras simples, una navegación coherente, así como evitar elementos que puedan generar confusión o una sobrecarga de información.

2.1.2. Tecnologías de asistencia

Las tecnologías de asistencia son herramientas que permiten a las personas con discapacidad interactuar con sistemas digitales y acceder a los contenidos web. Entre

las principales tecnologías de asistencia se encuentran los lectores de pantalla, los magnificadores de pantalla, los sistemas de reconocimiento de voz, los dispositivos de entrada alternativos y las líneas de braille electrónicas.

Uno de los principales objetivos de la accesibilidad web es la compatibilidad con estas tecnologías, de modo que todos los usuarios puedan utilizar y navegar por los sitios web independientemente de las herramientas que empleen.

2.2. Web Content Accessibility Guidelines

Las WCAG son un conjunto de recomendaciones desarrolladas por el W3C con el objetivo de mejorar la accesibilidad de los contenidos web, facilitando su acceso para las personas con discapacidad [40]. Actualmente, las WCAG son el estándar de referencia más utilizado en accesibilidad web, constituyendo la base para normativas y legislaciones de diversos países.

2.2.1. Principios fundamentales

Las pautas WCAG se organizan en torno a cuatro principios esenciales [30].

1. **Perceptible.** La información y los componentes de la interfaz deben presentarse de manera que puedan ser percibidos por todos los usuarios. Esto implica, entre otras cosas, proporcionar textos alternativos para imágenes, garantizar un contraste suficiente entre el texto y el fondo y ofrecer subtítulos en contenido multimedia.
2. **Operable.** Los componentes de interfaz y navegación deben poder usarse mediante distintos métodos de interacción, como la navegación por teclado, el control de foco, etc.
3. **Comprensible.** La información y el funcionamiento de la interfaz deben resultar comprensibles para los usuarios. Este principio incluye aspectos como el uso de un lenguaje claro, formularios correctamente etiquetados, mensajes de error comprensibles y navegación coherente.
4. **Robusto.** El contenido debe ser lo suficientemente robusto para ser interpretado correctamente por distintos navegadores, dispositivos y tecnologías de asistencia. Esto requiere utilizar un código semántico válido y componentes compatibles con lectores de pantalla y otras tecnologías de apoyo.

2.2.2. Versiones

La versión WCAG 2.1 [37] fue publicada en 2018 como una ampliación de WCAG 2.0, incorporando nuevos criterios relacionados con los dispositivos móviles, la accesibilidad cognitiva y los usuarios con baja visión. Posteriormente, en 2023 se publicó WCAG 2.2 [39], que añade nuevos criterios enfocados a mejorar la experiencia de aquellos usuarios con discapacidades cognitivas y motoras. Entre las mejoras implementadas se encuentran requisitos relacionados con el tamaño de elementos táctiles, la visibilidad del foco del teclado y la simplificación de procesos de autenticación. WCAG 2.2 incluye todos los criterios de WCAG 2.1 y versiones anteriores. Actualmente se está comenzando a trabajar en las WCAG 3.0 y se prevé que se empiecen a implementar en unos años. Entre los objetivos principales de esta nueva versión se incluye la facilidad de entendimiento, la intención de cubrir más necesidades del usuario y la flexibilidad que ofrecerá [38].

2.2.3. Niveles de cumplimiento

Las WCAG establecen además tres niveles de cumplimiento o conformidad [32]:

1. **Nivel A.** Representa el nivel mínimo de accesibilidad. Incluye requisitos básicos indispensables para que algunos usuarios puedan acceder al contenido.
2. **Nivel AA.** Es el nivel recomendado, así como el más utilizado en normativas y proyectos profesionales. Este grado garantiza un nivel adecuado de accesibilidad para la mayor parte de usuarios, y suele ser el estándar exigido legalmente.
3. **Nivel AAA.** Corresponde al máximo nivel de accesibilidad, incluyendo requisitos avanzados que mejoran de forma significativa la experiencia de usuario. No obstante, no siempre es posible cumplir todos estos requisitos en todos los tipos de contenido.

2.3. ISO

Las normas ISO (del inglés *International Organization for Standardization*) [31] proporcionan estándares internacionales que ayudan a garantizar la calidad, usabilidad y accesibilidad de los sistemas digitales. Una de las normas más importantes en accesibilidad web es la ISO 40500:2012, que establece WCAG 2.0 como estándar internacional y define criterios técnicos para el desarrollo de contenido web accesible, además de servir como referencia en certificaciones y auditorías de sitios web.

La aplicación de estas normas contribuye a garantizar la calidad de los sistemas web, aumentando la inclusividad de las plataformas [25].

2.4. Auditoría de accesibilidad

La realización de una auditoría de accesibilidad web [24] consiste en la evaluación de un sitio web con el objetivo de identificar posibles barreras de acceso y verificar el cumplimiento de estándares como las WCAG. La combinación de auditorías automáticas y manuales ofrece una evaluación completa y precisa de la accesibilidad de un sitio web.

La auditoría automática tiene como objetivo detectar problemas de accesibilidad técnicos mediante el uso de herramientas especializadas que analizan el código y la estructura del sitio web. Estas herramientas permiten identificar errores frecuentes, como la ausencia de texto alternativo en imágenes, problemas de contraste o etiquetas HTML incorrectas. Entre las herramientas más utilizadas destacan Lighthouse [4], axe DevTools [6] y WAVE Web Accessibility Evaluation Tool [33].

Aunque las herramientas automáticas agilizan mucho el proceso de análisis, no son capaces de detectar todos los problemas de accesibilidad, especialmente aquellos relacionados con la comprensión del contenido o la experiencia real del usuario.

Por otro lado, la auditoría manual complementa el análisis automático realizado previamente mediante pruebas llevadas a cabo por personas. Esta auditoría incluye la navegación mediante teclado, el uso de lectores de pantalla y la evaluación de aspectos de usabilidad y comprensión. Este tipo de auditoría permite detectar problemas que las herramientas automáticas no son capaces de identificar, como pueden ser estructuras de navegación confusas, mensajes ambiguos o dificultades a la hora de interactuar con la interfaz.

2.5. Drupal CMS

Drupal es un sistema de gestión de contenidos de código abierto desarrollado principalmente en PHP y creado en 2001 por Dries Buytaert. Desde entonces, ha evolucionado hasta convertirse en una de las plataformas más utilizadas para el desarrollo de sitios web complejos, escalables y personalizados. Se caracteriza por su flexibilidad y capacidad de adaptación a diferentes tipos de proyectos, destacando los entornos institucionales, educativos y corporativos. A lo largo de los años, Drupal ha

incorporado mejoras relacionadas con el rendimiento, experiencia de usuario, seguridad y accesibilidad [23, 29].

Utiliza una arquitectura modular basada en un núcleo que proporciona las funcionalidades básicas del sistema. Este núcleo puede ampliarse mediante el uso de módulos adicionales, desarrollados tanto por la comunidad como por desarrolladores externos a ella. Esta modularidad permite adaptar el CMS a diferentes necesidades sin modificar directamente el núcleo, facilitando las actualizaciones y el mantenimiento del sistema [8].

Además del núcleo, entre los principales componentes de Drupal destacan los módulos, que añaden características específicas como formularios avanzados o sistemas de autenticación; los temas, que controlan la apariencia de los sitios web; las entidades y taxonomías, que permiten al usuario organizar el contenido; y los roles y permisos, utilizados para gestionar el acceso y las acciones de los usuarios dependiendo de sus roles.

En lo relacionado con la accesibilidad web, Drupal incorpora diversas funcionalidades orientadas a facilitar el desarrollo de sitios accesibles y compatibles con estándares internacionales como las WCAG. Promueve el uso de buenas prácticas de desarrollo accesible mediante la generación de código HTML semántico, la compatibilidad con tecnologías de asistencia, la navegación mediante teclado, el uso de atributos ARIA (del inglés *Accessible Rich Internet Applications*), formularios accesibles y soporte para obtener un contraste y estructura adecuados del contenido [9].

Drupal CMS es una distribución basada en Drupal diseñada para facilitar la creación y administración de sitios web. A diferencia de la instalación tradicional de Drupal, Drupal CMS incorpora configuraciones, módulos y funcionalidades ya instaladas con el objetivo de acelerar el desarrollo y mejorar la experiencia del usuario. Se desarrolló con el objetivo de ofrecer una plataforma más intuitiva, pero manteniendo la flexibilidad característica de Drupal [12].

Capítulo 3

Auditoría de accesibilidad web

En este trabajo, las auditorías de accesibilidad se han realizado sobre un sitio web desarrollado con Drupal CMS [12] y el tema Mercury [15], configurado en un entorno local mediante el uso de DDEV [5] sobre Windows, utilizando Google Chrome como navegador principal para la realización de la auditoría y Drupal Canvas [11] como sistema de componentes visuales. La instalación de Drupal CMS se realizó siguiendo el procedimiento oficial de la comunidad Drupal, definiendo durante el proceso los parámetros iniciales del sitio (nombre, idioma principal y credenciales del usuario administrador) [16].

Para simular un entorno real de uso, se han creado cinco páginas de prueba que incorporan componentes habituales en aplicaciones web. La página *Home* incluye una cabecera con botones, tarjetas de contenido con imagen y texto, así como un bloque de cita. La página *Contacto* contiene un formulario desarrollado con el módulo Webform, con campos de texto, selector desplegable, casillas de verificación y botones de tipo radio. La página *Service* incorpora acordeones, un elemento de vídeo, tarjetas de precio con botones de selección y un bloque de cita. La página *Recursos* contiene encabezados, párrafos, listas ordenadas y desordenadas, una imagen con texto alternativo y una tabla. Por último, la página *Base Canvas* reúne la mayor parte de los componentes disponibles en Drupal Canvas entre los que se encuentran tarjetas, secciones *hero*, iconos y componentes de precios.

Una vez configurado el sitio web, se ha realizado una auditoría de accesibilidad con el objetivo de evaluar el grado de cumplimiento de los estándares WCAG. En este proyecto se ha aplicado una metodología que combina las auditorías automáticas y manuales. Las herramientas automáticas son capaces de detectar errores técnicos rápidamente, mientras que las pruebas manuales permiten el análisis de los aspectos relacionados con la experiencia de uso de un usuario real. Por tanto, este enfoque permite obtener

una evaluación completa del sitio web.

3.1. Auditoría automática de accesibilidad

Para la auditoría automática se han utilizado herramientas ampliamente empleadas en el ámbito de la accesibilidad web. En concreto, se han utilizado Lighthouse, axe DevTools y WAVE. Este análisis se ha aplicado a las principales páginas del sitio web (*Home*, *Contacto*, *Service*, *Recursos* y *Base Canvas*), con el objetivo de detectar errores técnicos y problemas de estructura. Las incidencias detectadas durante la auditoría han sido clasificadas en cuatro niveles, tal y como se muestra en la Tabla 3.1.

Nivel	Descripción
Crítica	Problemas que impiden completamente el acceso a determinados contenidos o funcionalidades
Grave	Barreras importantes que dificultan significativamente la navegación o comprensión del sitio
Moderada	Problemas que afectan parcialmente a la experiencia de usuario o incumplen recomendaciones WCAG
Leve	Incidencias menores relacionadas con buenas prácticas o mejoras recomendadas

Tabla 3.1: Clasificación de incidencias detectadas durante la auditoría automática

A continuación se detallan los resultados obtenidos en las auditorías automáticas, divididas por herramienta. Lighthouse [4] es una herramienta integrada en el navegador que permite analizar el rendimiento, la accesibilidad, las buenas prácticas y el SEO de un sitio web. Los resultados obtenidos se pueden consultar en la Tabla 3.2.

Página	Dispositivo	Rendimiento	Accesibilidad	Buenas prácticas	SEO
/home	Ordenador	61	94	100	92
	Móvil	71	94	100	92
/contact	Ordenador	72	96	100	92
	Móvil	72	96	100	92
/service	Ordenador	100	96	100	92
	Móvil	96	96	100	92
/recursos	Ordenador	100	96	100	92
	Móvil	91	96	100	92
/base-canvas	Ordenador	100	96	100	92
	Móvil	99	96	100	92

Tabla 3.2: Puntuaciones obtenidas con Lighthouse

Las puntuaciones obtenidas en la sección de accesibilidad oscilan entre 94 y 96 sobre 100, lo que indica un buen nivel general de accesibilidad. Sin embargo, el rendimiento

de la página */home* en escritorio presenta un valor de 61 sobre 100, el más bajo de las páginas analizadas. Esto posiblemente se debe a la existencia de una mayor cantidad de elementos multimedia presentes en ella.

En cuanto a las incidencias de accesibilidad detectadas, hay dos incidencias importantes identificadas por Lighthouse. La primera, *Links do not have a discernible name*, está presente en las cinco páginas: el enlace del logotipo de Mercury no dispone de texto visible, atributo `aria-label` ni atributo `title`, lo que impide que los lectores de pantalla puedan comunicar su propósito al usuario. La segunda incidencia, *Heading elements are not in a sequentially-descending order*, se detecta en la página *Home*, en la cual las tarjetas de contenido utilizan un elemento `h3` sin que exista un `h2` previo en esa sección, rompiendo la jerarquía semántica de encabezados recomendada por el nivel A del WCAG.

axe DevTools [6] es otra extensión de navegador especializada en la detección de incidencias WCAG y problemas de accesibilidad en componentes interactivos. El análisis con esta herramienta se ha realizado en las mismas cinco páginas, obteniendo las incidencias recogidas en la Tabla 3.3.

Página	Nº incidencias	Incidencias
<i>/home</i>	1	Moderado: Los niveles de encabezado solo deben aumentar de uno en uno.
<i>/contact</i>	2	Grave: Los enlaces deben tener texto reconocible. Moderado: La página debe contener un <code>h1</code> .
<i>/service</i>	2	Grave: Los enlaces deben tener texto reconocible. Moderado: La página debe contener un <code>h1</code> .
<i>/recursos</i>	2	Grave: Los enlaces deben tener texto reconocible. (x2)
<i>/base-canvas</i>	5	Grave: Los enlaces deben tener texto reconocible. (x2) Moderado: Los landmarks deben tener una combinación única de rol y etiqueta. (x2) Moderado: La página debe contener un <code>h1</code> .

Tabla 3.3: Incidencias detectadas con axe DevTools

Las incidencias más importantes detectadas con esta herramienta son las siguientes. La primera incidencia coincide con la ya identificada por la herramienta Lighthouse (*Links must have discernible text*): el enlace del logotipo carece de nombre accesible en las páginas analizadas, incumpliendo las recomendaciones del nivel A de WCAG. La siguiente incidencia, *Page should contain a level-one heading*, señala que las páginas *Contacto*, *Service* y *Base Canvas* no contienen ningún elemento `h1`, dificultando la navegación para los usuarios de lectores de pantalla que utilizan los encabezados como puntos de referencia. Por último, *Landmarks should have a unique role/label* detecta

que en la página *Base Canvas* existen dos elementos `<nav>` con el atributo `aria-label` vacío, entorpeciendo la tarea de las tecnologías de asistencia.

Otra de las herramientas utilizadas para la auditoría automática es WAVE (*Web Accessibility Evaluation Tool*) [33], que facilita la identificación de errores semánticos, problemas de contraste y elementos con etiquetas incorrectas. Los resultados obtenidos se muestran en la Tabla 3.4.

Página	AIM [35]	Nº Incidencias	Incidencias detectadas
/home	9.4/10	4	Imagen enlazada sin texto alternativo. (x2) Nivel de encabezado omitido. Enlace redundante.
/contact	9.1/10	4	Imagen enlazada sin texto alternativo. (x2) Ausencia de encabezado h1. Enlace redundante.
/service	5.4/10	5	Imagen enlazada sin texto alternativo. (x2) Ausencia de encabezado h1. Enlace redundante. Elemento de vídeo o audio en HTML5 sin subtítulos.
/recursos	8.2/10	5	Imagen enlazada sin texto alternativo. Enlace vacío. Referencia ARIA rota. Posible encabezado no marcado. Enlace redundante.
/base-canvas	5.9/10	6	Imagen enlazada sin texto alternativo. Enlace vacío. Referencia CSS rota. (x2) Ausencia de encabezado h1. Nivel de encabezado omitido.

Tabla 3.4: Incidencias detectadas con WAVE

Las puntuaciones más bajas de AIM (del inglés *Accessibility Impact*), de valores 5.4 y 5.9 sobre 10, corresponden a las páginas con mayor número y variedad de componentes, es decir, las páginas *Service* y *Base Canvas*. Entre las incidencias más relevantes detectadas destacan las siguientes.

Linked image missing alternative text está presente en todas las páginas analizadas, y hace referencia al logotipo de Mercury, que carece del atributo `alt`, incumpliendo las recomendaciones del nivel A de WCAG. *Missing first level heading* se detecta en las páginas *Contacto*, *Service* y *Base Canvas*, y marca la ausencia de encabezado h1, tal y como detectó la herramienta axe DevTools. *HTML5 video or audio*, detectado en la página *Service*, indica la presencia de un elemento multimedia sin subtítulos ni transcripción asociada, lo que supone una barrera para los usuarios con discapacidad

auditiva, así como un incumplimiento del nivel A de WCAG. Por último, *Redundant link* está presente en cuatro de las cinco páginas del sitio web, y detecta múltiples enlaces que apuntan al mismo destino, pudiendo confundir la navegación mediante lectores de pantalla.

La Tabla 3.5 recoge las incidencias detectadas por más de una herramienta, lo que ha permitido identificar los errores más consistentes del sitio web.

Incidencia	Lighthouse	axe	WAVE	Severidad	WCAG
Logo sin texto alternativo / enlace sin nombre accesible	✓	✓	✓	Grave	A
Ausencia de encabezado h1	–	✓	✓	Moderada	A/AA
Jerarquía de encabezados incorrecta	✓	✓	✓	Moderada	A

Tabla 3.5: Incidencias comunes detectadas por las tres herramientas automáticas

Este análisis comparativo muestra que la incidencia más grave del sitio es la ausencia de texto alternativo en el logotipo, detectada por las tres herramientas en todas las páginas analizadas. Esta incidencia pertenece al tema Mercury y, debido a su gravedad, es la principal candidata a convertirse en aportación técnica.

3.2. Auditoría manual de accesibilidad

La auditoría manual se ha realizado con el objetivo de analizar aspectos no detectables por herramientas automáticas. La auditoría manual realizada en este proyecto se ha centrado en dos aspectos principales: la navegación por teclado y la evaluación con lector de pantalla.

Para comenzar la auditoría manual, se ha realizado un análisis completo de la navegación mediante teclado en las páginas del sitio web. Para la navegación mediante teclado se ha utilizado la tecla **Tab** para avanzar entre elementos interactivos y la combinación **Shift**+**Tab** para retroceder. El objetivo de esta prueba ha sido verificar el cumplimiento de los criterios WCAG relacionados con la operabilidad del sitio web mediante el uso del teclado. Las figuras 3.1 a 3.5 muestran el recorrido del foco en las diferentes páginas analizadas.

En la página *Home* (Figura 3.1) se ha detectado una diferencia muy leve entre los elementos seleccionados y los no seleccionados, especialmente en los botones.



Figura 3.1: Navegación por teclado de la página *Home*.

En la página *Contacto* (Figura 3.2), el foco sigue un orden lógico. Sin embargo, se ha detectado un error en lo que respecta al contraste de los campos al recibir el foco, dificultando la identificación visual del texto del campo en cuestión.

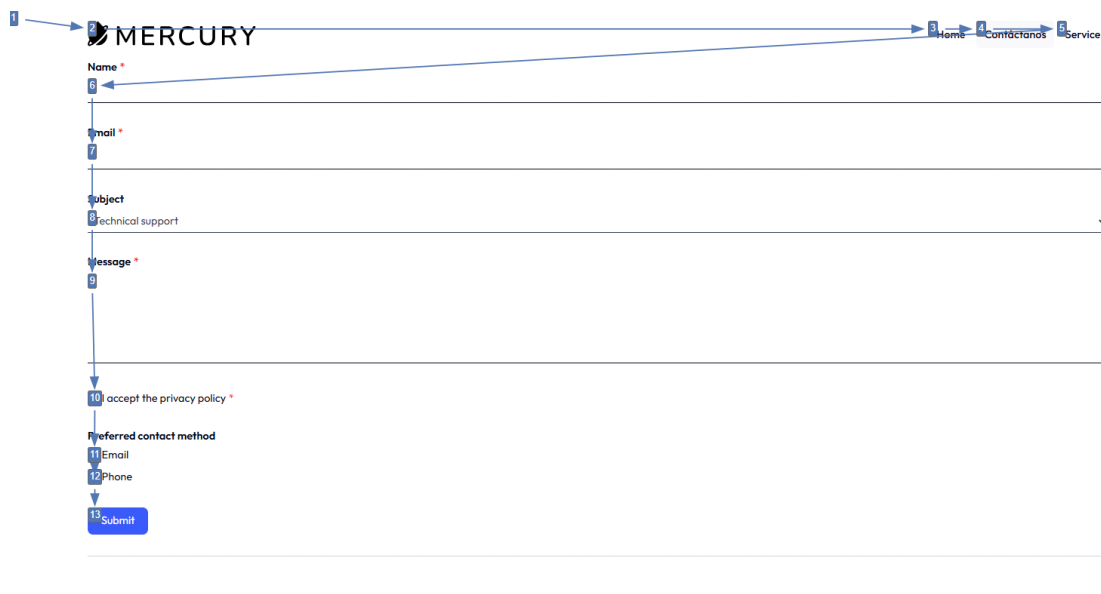


Figura 3.2: Navegación por teclado de la página *Contacto*.

En la página *Service* (Figura 3.3) se detectó que los botones, tal y como sucedía en la página *Home*, no muestran casi diferencias al estar seleccionados. Por lo demás, el foco del teclado sigue un orden lógico, permitiendo extender y comprimir los desplegables

con la tecla **Enter**.

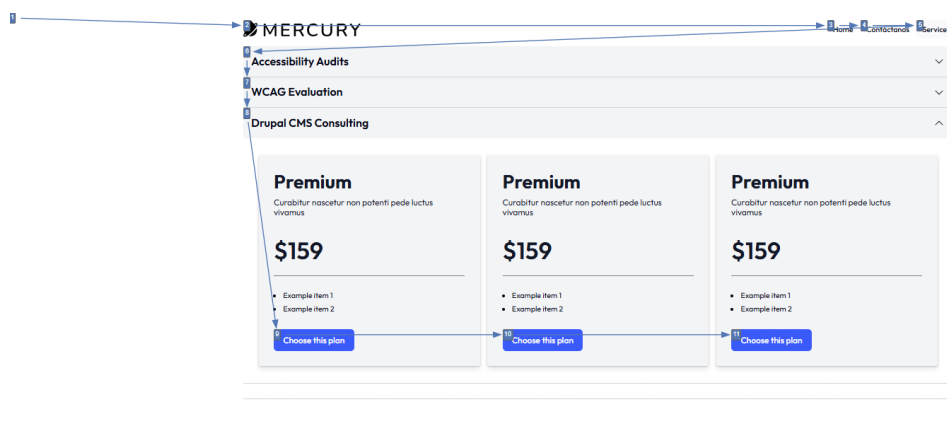


Figura 3.3: Navegación por teclado de la página *Service*.

En la página *Recursos* (Figura 3.4) el foco del teclado sigue un orden lógico, pudiendo navegar verticalmente por la página gracias a las flechas **↑** y **↓**.



Figura 3.4: Navegación por teclado de la página *Recursos*.

Por último, en la página *Base Canvas* (Figura 3.5) tampoco se han encontrado errores importantes que no se hubieran mencionado anteriormente. Como sucede en las páginas

Home y *Service*, el contraste de los botones al ser seleccionados no es demasiado evidente.

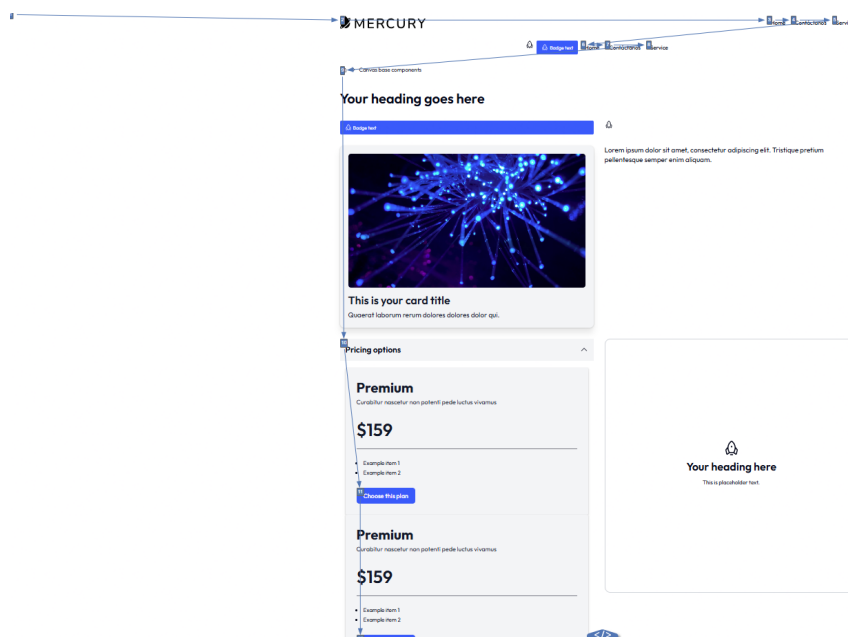


Figura 3.5: Navegación por teclado de la página *Base Canvas*.

En resumen, el sitio es mayormente operable mediante teclado, con un orden de foco generalmente lógico. No obstante, destacan las siguientes incidencias. En primer lugar, la visibilidad del indicador de foco es insuficiente, ya que en varias secciones del sitio, especialmente en los botones, el indicador visual de foco presenta un contraste reducido frente al fondo, dificultando su localización. En segundo lugar, existe un bajo contraste en los campos del formulario al recibir el foco, dificultando la visualización del contenido del campo seleccionado.

Después, la auditoría manual ha seguido con la evaluación con lector de pantalla, realizada con la herramienta NVDA (*NonVisual Desktop Access*) [28] y combinada con el navegador de Google Chrome. NVDA es uno de los lectores de pantalla más utilizados por usuarios con discapacidad visual, lo que permite evaluar la accesibilidad de un sitio web desde una perspectiva real. Las pruebas realizadas han incluido la navegación por encabezados, por enlaces y por elementos de formulario, así como la navegación por la estructura general mediante el diálogo de elementos.

A continuación se describen los resultados obtenidos en esta auditoría, organizados por página.

Página */home*. Durante la navegación por la página principal se han descubierto dos problemas de concatenación de texto. En primer lugar, al llegar a las tarjetas de

contenido, NVDA lee el enlace a la imagen y el título concatenadamente, sin realizar ninguna separación entre ellos, pudiendo así generar confusión. En segundo lugar, NVDA no incluye separación alguna entre frases no separadas por un punto, pese a tratarse de elementos HTML distintos.

En lo referente a la estructura del documento, la auditoría por NVDA muestra lo mismo que las anteriores auditorías realizadas: un salto directo de un **h1** a un **h3**, sin ningún **h2** intermedio. Además, se han detectado dos regiones de contenido vacías, lo que puede generar confusión al usuario.

Página /contact. La página de contacto ha sido la que más incidencias ha revelado. Al navegar por los campos del formulario con la tecla **F**, se ha detectado que los campos de tipo casilla de verificación y botón de tipo radio carecen de etiqueta accesible, lo que provoca que NVDA los anuncie sin un texto descriptivo asociado. Además, NVDA anuncia la casilla de verificación de aceptación de la política de privacidad como “*entrada inválida*”, mostrando un mensaje de error que puede confundir al usuario.

Página /service. En la página Service se han encontrado otros dos problemas. En el primero de ellos se ha detectado que los botones de selección de plan de precios están incluidos como campos de formulario, lo que puede confundir al usuario al no dejar clara la forma de interactuar con ellos.

Por otro lado, la jerarquía de encabezados carece de estructura lógica, ya que tanto los títulos de los acordeones como los de las tarjetas de precios son **h3**, y los precios de las tarjetas son **h4**. Esto puede confundir al usuario al navegar por la página con la tecla **H**.

Página /recursos. En la página Recursos no se han encontrado casi incidencias. Las imágenes disponen de texto alternativo, por lo que son leídas correctamente. Además, las listas ordenadas y desordenadas son identificadas correctamente y NVDA lee sus elementos con su correspondiente número.

Página /base-canvas. Esta página recoge la jerarquía de encabezados más problemática del sitio web. La secuencia detectada es: **h2** (navegación principal), **h2** (breadcrumb), **h2** (*Your heading goes here*), **h3** (*This is your card title*), **h3** (*Pricing options*), **h3** (*Premium*), **h4** (\$159), **h3** (*Premium*), **h4** (\$159), **h3** (*Your heading here*), **h2** (*Your heading goes here*). Esta estructura, como se puede observar, carece de

encabezados `h1`. Además, los botones de *“Choose this plan”* están implementados como campos de formulario, tal y como sucedía en la página *Service*.

Los resultados de la auditoría manual complementan los obtenidos en la auditoría automática. La navegación por teclado tiene un orden de foco generalmente lógico, y se han detectado problemas de visibilidad del foco y de contraste en formularios.

Por otra parte, la evaluación con NVDA revela incidencias no detectables automáticamente, como son la concatenación de texto entre elementos o una jerarquía de encabezados caótica.

3.3. Conclusiones

Del análisis conjunto de las dos auditorías se pueden extraer las siguientes conclusiones.

El sitio web presenta un nivel de accesibilidad generalmente aceptable, con puntuaciones Lighthouse bastante elevadas. Sin embargo, este valor no refleja la realidad del sitio debido a la existencia de incidencias de nivel A que se escapan a la detección automática.

Se han identificado dos incidencias críticas presentes en todas las páginas: la ausencia de texto alternativo en el logotipo y los problemas en la jerarquía de encabezados, relacionadas con el tema Mercury y con Drupal Canvas respectivamente. Por otra parte, la auditoría manual con NVDA ha revelado incidencias relevantes que tampoco son detectables automáticamente: concatenación de texto entre elementos, ausencia de etiquetas en formularios y la implementación semánticamente incorrecta de botones.

Capítulo 4

Aportación técnica a la comunidad Drupal

A partir de los resultados obtenidos en las auditorías de accesibilidad web descritas en el Capítulo 3 se realizó un análisis de las posibles aportaciones técnicas a la comunidad Drupal, así como de las incidencias previamente reportadas por los desarrolladores de la comunidad de Drupal [14].

Se considera aportación técnica a la comunidad Drupal cualquiera de las siguientes acciones: reportar un nuevo error documentándolo con evidencias, desarrollar la solución a un error ya reportado o revisar y verificar la solución propuesta por otro colaborador. En este proyecto se han desarrollado aportaciones de los dos primeros tipos. Para ello, se ha seguido la guía de contribución proporcionada por la comunidad Drupal [10], así como los consejos ofrecidos por expertos en la tecnología.

La Tabla 4.1 recoge el conjunto de posibles aportaciones identificadas durante las auditorías realizadas, así como de incidencias recogidas en la comunidad de Drupal mencionada anteriormente. La columna *Referencia* indica el código identificativo de la incidencia ya reportada en `drupal.org`. Cuando una incidencia aún no ha sido reportada a la comunidad, la columna aparece vacía.

Tabla 4.1: Opciones de aportaciones técnicas identificadas a partir de las auditorías y de las ya reportadas en `drupal.org`

Incidencia	Componente	Impacto	WCAG	Referencia
Añadir <code>aria-label</code> al enlace del logotipo	Mercury	Alto	A	#3591885
Contraste insuficiente en campos de formulario al recibir foco	Mercury	Alto	AA	-

Continúa en la página siguiente

Incidencia	Componente	Impacto	WCAG	Referencia
<code>aria-label</code> vacío en elementos <code><nav></code>	Mercury	Medio	AAA	-
Jerarquía de encabezados en tarjetas incorrecta (<code>h3</code> sin <code>h2</code>)	Canvas	Alto	A	-
Ausencia de <code><h1></code> en páginas interiores	Mercury	Medio	AA	-
Ausencia de etiquetas accesibles en casillas de verificación y botones de tipo radio	Webform	Medio	A	-
Botones y enlaces de la barra de herramientas sin nombre accesible	Gin	Alto	A	#3539420
Mejora de la jerarquía de encabezados en el tema Gin	Gin	Alto	A	#3492096
Mejora de la accesibilidad del módulo <i>autosave_form</i>	Autosave Form	Medio	AA	#3502158
Orden de tabulación incorrecto en la cabecera de la tabla	Gin	Medio	A	#3506338

Del conjunto de aportaciones identificadas, se han seleccionado tres aportaciones para su desarrollo en este proyecto, que se corresponden con las principales incidencias detectadas durante las auditorías automáticas y manuales.

4.1. Aportación 1. Texto alternativo en el enlace del logotipo

Como se detectó en las auditorías automáticas, el logotipo del tema Mercury no dispone de un texto alternativo ni del atributo `aria-label` por lo que, además, los lectores de pantalla lo anuncian como un enlace vacío. Tras analizar el problema, se detectó que es debido a que el template de bloque del logotipo de este tema envuelve la imagen en un enlace `<a>` que carece de nombre accesible. La solución consiste en añadir un atributo `aria-label` al enlace del logotipo, de manera que las herramientas lectoras de pantalla anuncien su propósito. Además, se ha modificado también el texto alternativo de la imagen para que no esté vacío.

Tras aplicar la solución, se realizó de nuevo la auditoría automática, que confirmó la desaparición del error *Links must have a discernible text*. La aportación se ha publicado en el repositorio oficial del tema Mercury en drupal.org como solución a una incidencia previamente reportada. La incidencia puede consultarse en <https://git.drupalcode.org/project/mercury/-/issues/3539420>.

drupalcode.org/project/mercury/-/work_items/3591885 y la petición de fusión con la solución desarrollada en https://git.drupalcode.org/project/mercury/-/merge_requests/207.

4.2. Aportación 2. Contraste insuficiente en campos de formulario

Al recibir el foco, los campos del formulario presentan problemas de contraste. Por un lado, el borde inferior del campo que recibe el foco utiliza la variable CSS `--accent` del tema Mercury, que tiene un contraste muy por debajo del mínimo exigido por el Nivel AA. Por otro lado, la etiqueta del campo cambia al mismo color `--accent` al recibir el foco, lo que genera el mismo problema de contraste insuficiente a la hora de identificar visualmente los detalles del campo. Esta incidencia fue detectada durante la auditoría por teclado realizada sobre la página *Contacto*, al observar que el contenido y la etiqueta del campo resultaban prácticamente imperceptibles al recibir el foco.

La solución encontrada consiste en modificar el archivo `src/main.css` del tema Mercury para sustituir el uso de `--accent` en los estados de foco de los campos de formulario por `--foreground`, el color de texto principal del tema, que dispone de contraste suficiente sobre fondos claros para cumplir el Nivel AA del WCAG. Se han modificado dos reglas CSS: la primera afecta al borde inferior del campo al recibir el foco, mientras que la segunda afecta a la etiqueta del campo cuando está activo.

Tras aplicar el *fix* y recompilar el CSS con el proceso de compilación de Tailwind del tema, el borde inferior y la etiqueta del campo activo presentan un contraste suficiente frente al fondo de la página, superando el ratio mínimo exigido por el Nivel AA de WCAG, tal y como se puede observar en el ejemplo de las Figuras 4.1 y 4.2.

La incidencia se ha reportado en el repositorio del tema Mercury en drupal.org y se puede consultar en https://git.drupalcode.org/project/mercury/-/work_items/3592153, mientras que el código aportado se encuentra en <https://git.drupalcode.org/issue/mercury-3592153/-/tree/3592153-insufficient-focus-contrast-form-fields>.

Name *

Email *

Subject
- None -

Message *

☐ I accept the privacy policy *

Preferred contact method

☐ Email

☐ Phone

Submit

Figura 4.1: Contraste insuficiente en el campo “Mensaje” del formulario antes del *fix*

Name *

Email *

Subject
- None -

Message *

☐ I accept the privacy policy *

Preferred contact method

☐ Email

☐ Phone

Submit

Figura 4.2: Contraste corregido en el campo “Mensaje” del formulario tras el *fix*

4.3. Aportación 3. Elementos de navegación sin aria único

La plantilla del menú principal del tema Mercury genera un elemento `<nav>` con un atributo `aria-label="{{ menu_label }}"`. Sin embargo, hay veces en las que la variable `menu_label` llega vacía, lo que significa que el atributo `aria-label` resulta vacío. Esta incidencia se ha detectado en la página *Base Canvas*, en la que los dos elementos de navegación son anunciados por NVDA de forma idéntica, “*Navegación, 3 elementos*”, sin ninguna etiqueta que los diferencie. Además, esta incidencia también fue detectada por la herramienta automática axe DevTools, que la identificó como *Landmarks should have a unique role/label*.

La solución encontrada para este problema consiste en añadir una regla que, en el caso en el que `menu_label` llegue vacío, el atributo `aria-label` reciba un valor por

defecto, siendo este valor traducible gracias al filtro `|t`. Tras aplicar los cambios y limpiar la caché de Drupal, axe DevTools confirmó la desaparición del error previamente mencionado.

El parche desarrollado para esta aportación se ha generado en formato `.patch` y está disponible en el repositorio del proyecto¹. Su reporte y publicación en el repositorio oficial del tema Mercury en `drupal.org` quedan pendientes como una futura línea de trabajo.

¹Véase https://github.com/nereasalamero/tfg_accesibilidad

Capítulo 5

Conclusiones y trabajo futuro

Actualmente, la accesibilidad web representa un requisito en el desarrollo de software, desde una perspectiva tanto técnica como legal. Sin embargo, su evaluación sigue siendo una tarea compleja que requiere la combinación de herramientas automáticas y manuales. Este proyecto ha permitido comprobar de primera mano esta realidad, evaluando la accesibilidad de un sitio desarrollado con Drupal CMS y el tema Mercury.

El trabajo ha conseguido alcanzar los objetivos planteados al inicio del proyecto. Se ha configurado un entorno de prueba completo, se han realizado auditorías automáticas con tres herramientas especializadas, y se ha llevado a cabo una evaluación manual mediante navegación por teclado y lector de pantalla. Los resultados muestran que el sitio presenta un nivel de accesibilidad generalmente aceptable, pero revelan incidencias que afectan a todos aquellos sitios desarrollados con el tema Mercury.

Una de las conclusiones más relevantes del proyecto es que las herramientas automáticas, aunque resultan muy útiles y agilizan mucho el proceso de evaluación del sitio web, no son suficientes para realizar una evaluación completa. Algunas incidencias, como el contraste insuficiente en los campos del formulario al recibir el foco o la concatenación de texto entre elementos consecutivos detectada por NVDA , no fueron identificadas por ninguna de las herramientas automáticas empleadas. Esto confirma, una vez más, el valor de la auditoría manual como complemento del proceso de evaluación de accesibilidad.

Otra de las conclusiones más significativas es que las incidencias más graves se han detectado en el tema Mercury o en el sistema de componentes Canvas y no en el contenido específico del sitio. Esto implica que cualquier sitio desarrollado con estas herramientas presenta los mismos problemas, demostrando así la importancia de las aportaciones desarrolladas en este proyecto. Las correcciones implementadas y publicadas en el repositorio oficial de Mercury en drupal.org contribuyen a mejorar

la accesibilidad web de todos los sitios que utilicen este tema.

A partir de los resultados obtenidos y de las limitaciones del proyecto, se proponen las siguientes posibles líneas de trabajo futuro. En primer lugar, la publicación de la tercera aportación técnica desarrollada en este proyecto, que queda pendiente de ser reportada y publicada en el repositorio oficial de Mercury en drupal.org, así como realizar un seguimiento de las dos aportaciones ya publicadas, que están pendientes de ser revisadas por parte de los mantenedores del proyecto.

En segundo lugar, sería interesante ampliar la evaluación manual a otros lectores de pantalla. Este trabajo se ha centrado en la evaluación mediante el uso de NVDA en Google Chrome, pero se podría extender al uso de JAWS [34], uno de los lectores de pantalla más utilizados en entornos profesionales, o de VoiceOver [1] en macOS e iOS, lo que permitiría obtener una visión más completa de la experiencia de los usuarios con discapacidad visual.

En tercer lugar, la evaluación de accesibilidad web y desarrollo de mejoras del panel de administración. Este trabajo se ha centrado en la evaluación del sitio que ve el usuario final, pero el panel de administración de Drupal CMS, basado generalmente en el tema Gin, presenta sus propias incidencias de accesibilidad, varias de las cuales ya han sido detectadas y reportadas por la comunidad Drupal. Su evaluación y posterior desarrollo de mejoras sería una línea futura de trabajo lógica de este trabajo.

Otra línea futura de trabajo lógica es participar activamente en el reporte y solución de incidencias reportadas en drupal.org, ampliando el impacto de este proyecto y contribuyendo a lo largo del tiempo a la mejora de la accesibilidad en la comunidad Drupal.

Por último, la evaluación con usuarios reales con discapacidad aportaría una perspectiva muy interesante y de gran valor a este trabajo. Las auditorías realizadas han sido llevadas a cabo con un simulador del uso de tecnologías de asistencia, así que su participación sería de gran ayuda para la identificación de problemas no detectados a lo largo de las auditorías ya finalizadas y para la confirmación de los problemas identificados y reportados.

Bibliografía

- [1] APPLE, *Introducción a VoiceOver*. https://www.apple.com/es/voiceover/info/guide/_1121.html. Consultado en junio de 2026.
- [2] BOLETÍN OFICIAL DEL ESTADO, *Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público*. https://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-12699, 2018. Consultado en mayo de 2026.
- [3] —, *Directiva (UE) 2019/882 del Parlamento Europeo y del Consejo, de 17 de abril de 2019, sobre los requisitos de accesibilidad de los productos y servicios*. <https://www.boe.es/buscar/doc.php?id=DOUE-L-2019-80999>, 2019. Consultado en mayo de 2026.
- [4] CHROME DEVELOPERS, *Lighthouse*. <https://developer.chrome.com/docs/lighthouse>, 2026. Consultado en abril de 2026.
- [5] DDEV FOUNDATION, *DDEV*. <https://ddev.com/>, 2026. Consultado en febrero de 2026.
- [6] DEQUE, *axe DevTools*. <https://www.deque.com/axe/devtools/extension/>, 2026. Consultado en abril de 2026.
- [7] F. J. DÍAZ, I. HARARI, AND A. P. AMADEO, *Accesibilidad web*, Libros de Cátedra, (2020).
- [8] DRUPAL, *Drupal como Sistema de Gestión de Contenidos*. https://www.drupal.org/es/docs/user_guide/es/understanding-drupal.html, 2025. Consultado en mayo de 2026.
- [9] —, *Accessibility*. <https://www.drupal.org/about/features/accessibility>, 2026. Consultado en abril de 2026.
- [10] —, *Contributor guide*. <https://www.drupal.org/community/contributor-guide>, 2026. Consultado en junio de 2026.

- [11] —, *Drupal Canvas*. <https://www.drupal.org/project/canvas>, 2026. Consultado en enero de 2026.
- [12] —, *Drupal CMS*. <https://new.drupal.org/drupal-cms>, 2026. Consultado en enero de 2026.
- [13] —, *Drupal.org*. <https://new.drupal.org/home>, 2026. Consultado en febrero de 2026.
- [14] —, *Issues for all projects*. <https://www.drupal.org/project/issues>, 2026. Consultado en mayo de 2026.
- [15] —, *Mercury Theme*. <https://www.drupal.org/project/mercury>, 2026. Consultado en marzo de 2026.
- [16] DRUPAL ASSOCIATION, *Install with DDEV*. https://project.pages.drupalcode.org/drupal_cms/get-started/ddev/, 2026. Consultado en marzo de 2026.
- [17] DRUSH, *Drush*. <https://www.drush.org/13.x/>, 2025. Consultado en febrero de 2026.
- [18] GANTT CHART SOFTWARE, INFORMATION, AND SOFTWARE, *What is a Gantt Chart?* <https://www.gantt.com/>, 2026. Consultado en febrero de 2026.
- [19] GEEKS FOR GEEKS, *Agile Methodology*. <https://www.geeksforgeeks.org/software-testing/what-is-agile-methodology/>, 2026. Consultado en marzo de 2026.
- [20] GET COMPOSER, *Composer*. <https://getcomposer.org/>, 2026. Consultado en febrero de 2026.
- [21] GITHUB, INC., *GitHub*. <https://github.com/>, 2026. Consultado en febrero de 2026.
- [22] Y. HASSAN MONTERO AND F. J. MARTÍN FERNÁNDEZ, *Qué es la accesibilidad web*, No solo usabilidad, (2003).
- [23] HIBERUS, *Seguridad en Drupal: Razones por las que Drupal es el CMS más seguro*. <https://www.hiberus.com/crecemos-contigo/drupal-cms-mas-seguro/>, 2025. Consultado en junio de 2026.

- [24] ILUNION, *Guía para realizar una auditoría de accesibilidad web*. <https://www.ilunion.com/es/blog-puntoilunion/guia-auditoria-de-accesibilidad-web>, 2026. Consultado en junio de 2026.
- [25] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO), *Accesibilidad*. <https://www.iso.org/es/sectores/diversidad-inclusion/accesibilidad>, 2018. Consultado en mayo de 2026.
- [26] KANBAN TOOL, *Metodología Kanban*. <https://kanbantool.com/es/metodologia-kanban>, 2026. Consultado en febrero de 2026.
- [27] NOTION LABS, INC., *Notion*. <https://www.notion.so>, 2026. Consultado en febrero de 2026.
- [28] NV ACCESS, *NVDA (Non Visual Desktop Access)*. <https://nvda.es/>, 2026. Consultado en mayo de 2026.
- [29] PROMETEO INNOVATIONS, *Drupal refuerza su posición de liderazgo*. <https://www.prometeoinnovations.com/drupal-refuerza-su-posicion-de-liderazgo/>, 2026. Consultado en junio de 2026.
- [30] M. RIBERA, *La nueva normativa de accesibilidad WCAG 2.0 y los documentos en Internet*, Hipertext. net, 7 (2009).
- [31] UNIVERSIDAD DE ALICANTE, *International Organization for Standardization*. <https://www.iso.org/home.html>, 2026. Consultado en abril de 2026.
- [32] —, *Significado de los niveles A, AA y AAA de accesibilidad web*. <https://accesibilidadweb.dlsi.ua.es/?menu=que-significan-niveles-a-aa-aaa-accesibilidad-web>, 2026. Consultado en abril de 2026.
- [33] UTAH STATE UNIVERSITY, *WAVE Web Accessibility Evaluation Tools*. <https://wave.webaim.org/>, 2026. Consultado en abril de 2026.
- [34] VISPERO, *JAWS Screen Reader*. <https://vispero.com/jaws-screen-reader-software/>, 2026. Consultado en junio de 2026.
- [35] WAVE - WEB ACCESSIBILITY EVALUATION TOOL, *WAVE AIM Score*. <https://wave.webaim.org/aimscore>, 2026. Consultado en mayo de 2026.

- [36] WORLD HEALTH ORGANIZATION, *Disability and health*. <https://www.who.int/news-room/fact-sheets/detail/disability-and-health>, 2023. Consultado en mayo de 2026.
- [37] WORLD WIDE WEB CONSORTIUM, *Web content accessibility guidelines (WCAG) 2.1*. <https://www.w3.org/TR/WCAG21/>, 2023. Consultado en mayo de 2026.
- [38] —, *WCAG 3 Introduction*. <https://www.w3.org/WAI/wcag3>, 2025. Consultado en junio de 2026.
- [39] —, *Web Content Accessibility Guidelines (WCAG) 2.2*. <https://www.w3.org/TR/WCAG22/>, 2025. Consultado en mayo de 2026.
- [40] —, *Web Content Accessibility Guidelines (WCAG)*. <https://www.w3.org/WAI/standards-guidelines/wcag/>, 2026. Consultado en abril de 2026.

Anexos

Anexo A

Organización del proyecto

El proyecto se organizó siguiendo prácticas de gestión ágil [19]. Para comenzar, se mantuvieron reuniones bisemanales con un equipo formado por profesionales en accesibilidad y Drupal. Además, se utilizó un repositorio en GitHub [21], donde se gestionó el control de versiones y del código. Se creó una página en la herramienta Notion [27] para llevar un seguimiento de las tareas en formato Kanban [26], organizándolas en seis columnas (no empezada, en proceso, bloqueada, en revisión, completada, descartada).

La figura A.1 muestra la planificación seguida a lo largo del desarrollo del proyecto, que ha ido variando a lo largo del tiempo para ajustarse al tiempo disponible y a la redimensión de las tareas. Además, en la Tabla A.1 se puede observar el total de horas dedicadas al proyecto.

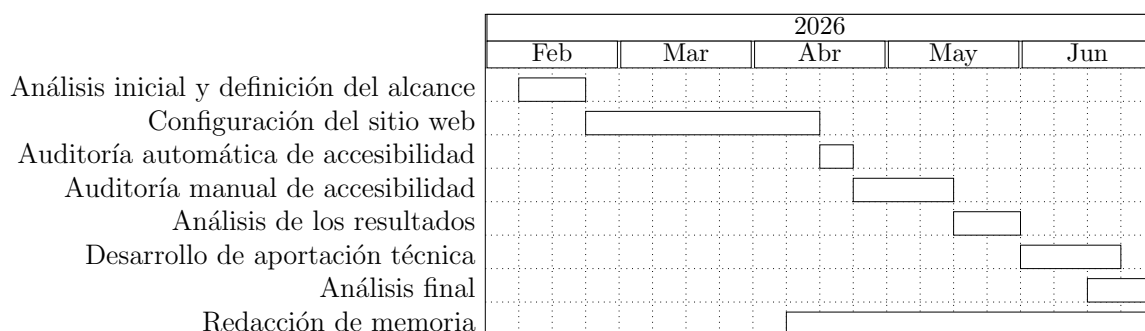


Figura A.1: Diagrama de Gantt del proyecto

Tarea	Dedicación (h)
Análisis inicial y definición del alcance	20
Configuración del sitio web	40
Auditoría automática de accesibilidad	20
Auditoría manual de accesibilidad	32
Análisis de los resultados	30
Desarrollo de aportación técnica	38
Análisis final	10
Redacción de memoria	110
Reuniones de seguimiento	12
Total	312

Tabla A.1: Esfuerzo invertido en las diferentes etapas del proyecto