



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Accesibilidad Web y Experiencia de Usuario en los Parlamentos Autonómicos

*Web Accessibility and User Experience in Autonomous
Parlaments*

Elena Rijo García

La Laguna, 13 de junio de 2022

D. **María Elena Sánchez Nielsen**, con N.I.F. 42.848.599-J profesora Titular de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

C E R T I F I C A (N)

Que la presente memoria titulada:

"Accesibilidad Web y Experiencia de Usuario en los Parlamentos Autonómicos"

ha sido realizada bajo su dirección por D. **Elena Rijo García**, con N.I.F. 78.766.873-S.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de junio de 2022

Agradecimientos

En primer lugar, quiero agradecer a todos los profesores que he tenido a lo largo de esta etapa universitaria, por haber fomentado el desarrollo de mi curiosidad por este ámbito.

En especial agradecer a mis padres por animarme y apoyarme incondicionalmente para seguir adelante, aún estando lejos de casa. Por darme la oportunidad de realizar estos estudios. Por ser el motor que me impulsa a superarme y por confiar en mí más que yo misma.

A mi hermana, por ser esa compañera de vida que no ha dejado que me rinda en ningún instante. Gracias por enseñarme tantas cosas, por ser mi guía y por demostrarme que con esfuerzo se consigue lo que se quiere.

Agradecer a mi pareja por haber estado a lo largo de mi carrera apoyándome en todo momento. Gracias por la paciencia, por haber estado ahí para escucharme y por los abrazos.

Y a mis amigos y amigas por las vivencias compartidas y todo lo aprendido. Gracias por animarme en todo momento.

Licencia

* Si NO quiere permitir que se compartan las adaptaciones de tu obra y NO quieres permitir usos comerciales de tu obra indica:



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional.

Resumen

El trabajo de fin de grado elaborado consta de la investigación y el análisis de la accesibilidad y la experiencia de usuario sobre un caso de estudio, en concreto: los parlamentos autonómicos, del Congreso de Diputados y el Senado de España, en el ámbito de las Tecnologías de la Información y la Comunicación.

El caso de estudio de accesibilidad se lleva a cabo a través de la metodología para el Seguimiento Simplificado UNE-EN 301549:2019 y de herramientas automáticas online y de código abierto, que ayudan a inspeccionar el cumplimiento de los criterios establecidos por la WCAG 2.1. Así mismo, se generan gráficas y estadísticas con las que podamos verificar el nivel de ayuda y fiabilidad que aportan dichas herramientas.

En cuanto al estudio de experiencia de usuario, se utilizan las métricas Core Web Vitals que ofrece Google. Se comprueba la UX de los casos de estudio a través de la herramienta PageSpeed Insight, la cual implementa el motor de análisis mencionado anteriormente. De esta forma, se recogen periódicamente los datos sobre el rendimiento y el tiempo de carga de página, con los que gracias a recomendaciones se podrán mejorar.

Se desarrolla un aplicativo web que contiene las soluciones propuestas para el caso de estudio. De esta manera, a través de un modelo de APIs se comprueba, por un lado la validación de los criterios de accesibilidad disponibles, y por otro lado el rango en el que se encuentran las métricas analizadas para la experiencia de usuario. La información obtenida será procesada y posteriormente maquetada para su visualización. Además, el aplicativo contiene las gráficas dinámicas que se generan en el estudio previo de las herramientas automáticas de accesibilidad.

Palabras clave: accesibilidad, usabilidad, experiencia de usuario, herramientas automáticas, metodologías, casos de estudio, TIC, criterios, Core Web Vitals, aplicativo web, Angular, API.

Abstract

The project consists of research and analysis of the accessibility and user experience of the autonomous parliaments, the Congress of Deputies and the Senate of Spain, in the field of Information and Communication Technologies.

The accessibility case study is carried out through the methodology for Simplified Monitoring UNE-EN 301549:2019 and automatic online tools and open sources, which help to inspect compliance with the criteria established by the WCAG 2.1. In this way, graphs and statistics are generated with which we can verify the level of help and reliability provided by these tools.

As for the user experience study, the Core Web Vitals metrics offered by Google are used. The UX of the case studies is checked through the PageSpeed Insight tool, which implements the aforementioned analysis engine. In this way, data on performance and page load time is collected on a regular basis, with which recommendations can be made for improvement.

A web application containing the solutions proposed for the case study is created. In this way, through a model of APIs, the validation of the available accessibility criteria is checked, on the one hand, and on the other hand, the range in which the metrics analysed for the user experience are found. The information obtained will be processed, and subsequently laid out for visualisation. In addition, the application contains the dynamic graphs generated in the previous study of the automatic accessibility tools.

Keywords: Accessibility, Usability, User Experience, Automatic Tools, Methodologies, Case Studies, ICT, Criteria, Core Web Vitals, Web Application, Angular, APIs.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Objetivos	2
1.3. Planificación del proyecto	2
2. Estado del Arte	3
2.1. Accesibilidad	3
2.1.1. Legislación	3
2.1.2. Metodología existentes	4
2.1.3. Herramientas existentes	6
2.2. Usabilidad	8
2.3. Experiencia de usuario	9
3. Accesibilidad	10
3.1. Caso de Estudio	10
3.2. Metodología empleada	10
3.3. Herramientas empleadas	11
3.4. Aspectos generales	11
3.5. Resultado del Caso de Estudio de Accesibilidad	13
3.5.1. Parlamento de Canarias	14
3.5.2. Parlamento de Andalucía	15
3.5.3. Parlamento de la Comunidad Valenciana	16
3.5.4. Asamblea de Madrid	16
3.5.5. Parlamento de Cataluña	17
3.5.6. Parlamento de Extremadura	17
3.5.7. Parlamento de Galicia	18
3.5.8. Parlamento del Principado de Asturias	19
3.5.9. Congreso de Diputados	20
3.5.10 Senado de España	20
4. Usabilidad	22
4.1. Análisis heurístico	22
5. Experiencia de Usuario	26
5.1. Core Web Vitals	26
5.2. PageSpeed Insight	28
5.3. Resultado del Caso de Estudio de UX	28
5.3.1. Parlamento de Canarias	29

5.3.2. Parlamento de Andalucía	30
5.3.3. Parlamento de la Comunidad Valenciana	30
5.3.4. Asamblea de Madrid	31
5.3.5. Parlamento de Cataluña	32
5.3.6. Parlamento de Extremadura	32
5.3.7. Parlamento de Galicia	33
5.3.8. Parlamento del Principado de Asturias	34
5.3.9. Congreso de Diputados	35
5.3.10.Senado de España	36
6. Aplicativo Web	37
6.1. Tecnologías y herramientas	37
6.2. Diseño del Front-End	39
6.3. Desarrollo de accesibilidad	42
6.3.1. Estructura del aplicativo	42
6.3.2. Funcionamiento	45
6.3.3. Estadísticas	46
6.4. Desarrollo de experiencia de usuario	48
6.4.1. Estructura del aplicativo	49
6.4.2. Funcionamiento	51
7. Conclusiones y Lineas Futuras	53
8. Conclusions and Future Lines of Work	55
9. Presupuesto	57
A. Acrónimos	58

Índice de Figuras

1.1. Diagrama de Gantt de la planificación del proyecto	2
3.1. Segunda tabla creada.	13
3.2. Diagrama de accesibilidad del Parlamento de Canarias.	15
3.3. Diagrama de accesibilidad del Parlamento de Andalucía.	15
3.4. Diagrama de accesibilidad del Parlamento de Valencia.	16
3.5. Diagrama de accesibilidad de la Asamblea de Madrid.	17
3.6. Diagrama de accesibilidad del Parlamento de Cataluña.	17
3.7. Diagrama de accesibilidad del Parlamento de Extremadura.	18
3.8. Diagrama de accesibilidad del Parlamento de Galicia.	19
3.9. Diagrama de accesibilidad del Parlamento de Asturias.	19
3.10 Diagrama de accesibilidad del Congreso de Diputados.	20
3.11 Diagrama de accesibilidad del Senado de España.	21
5.1. Configuración perteneciente al entorno de trabajo..	29
5.2. Rendimiento de carga del Parlamento de Canarias a día 21 de mayo de 2022 a las 11:53	29
5.3. Rendimiento de carga del Parlamento de Andalucía a día 21 de mayo de 2022 a las 11:12	30
5.4. Rendimiento de carga de las Cortes Valencianas a día 21 de mayo de 2022 a las 11:54	31
5.5. Rendimiento de carga de la Asamblea de Madrid a día 21 de mayo de 2022 a las 12:16	31
5.6. Rendimiento de carga del Parlamento de Cataluña día 21 de mayo de 2022 a las 12:53	32
5.7. Rendimiento de carga del Parlamento de Extremadura día 21 de mayo de 2022 a las 13:04	33
5.8. Rendimiento de carga del Parlamento de Galicia día 21 de mayo de 2022 a las 13:35	33
5.9. Rendimiento de carga del Parlamento del Principado de Asturias día 21 de mayo de 2022 a las 13:52	34
5.10 Rendimiento de carga del Congreso de Diputados día 21 de mayo de 2022 a las 14:28	35
5.11 Rendimiento de carga del Senado de España día 21 de mayo de 2022 a las 14:44	36
6.1. Estructura componentes de solución.	37
6.2. Página de inicio.	40
6.3. Página de accesibilidad en el que se muestra el contenido del apartado legislación.	41

6.4. Página de usabilidad en el que se muestra el contenido de los 10 principios de Jakob Nielsen.	41
6.5. Método para llamada a la API Wave.	42
6.6. Interfaces de accesibilidad.	43
6.7. Implementación principal del análisis de accesibilidad.	44
6.8. Modelado de datos para la representación gráfica.	44
6.9. Página inicial del apartado validador.	45
6.10 Analizar accesibilidad de un caso concreto.	45
6.11 Criterios del análisis de accesibilidad.	46
6.12 Contenido asociado a un criterio.	46
6.13 Estructura de componentes estadísticas.	47
6.14 Página principal del apartado estadísticas.	47
6.15 Gráfica del estudio de accesibilidad del Parlamento de Canarias.	48
6.16 Gráfica de porcentajes y por nivel de adecuación del estudio de accesibilidad del Parlamento de Canarias.	48
6.17 Método para la llamada a la API de UX.	49
6.18 Estructura de datos UX.	50
6.19 Estructura de datos UX.	50
6.20 Llamada al servicio y modelado de datos.	51
6.21 Asignación de color y categoría por métrica.	51
6.22 Página principal del apartado UX.	52
6.23 Análisis de UX del Parlamento de Canarias.	52

Índice de Tablas

9.1. Presupuesto estimado del proyecto. 57

Capítulo 1

Introducción

1.1. Introducción

En un mundo digital es cada vez más importante que los sitios web estén correctamente diseñados, desarrollados, sean fáciles de usar y proporcionen una buena experiencia de usuario. El poder de la web está en su universalidad y por consiguiente, tanto la accesibilidad web, como la experiencia de usuario son dos aspectos esenciales para llevarlo a cabo. Actualmente, la legislación en España y Europa recogen diferentes consideraciones que deben ser tenidas en cuenta, así como diferentes guías y métricas para llevar a cabo las evaluaciones en empresas y organismos públicos.

Por ese motivo este Trabajo de Fin de Grado (TFG) tiene como objetivo principal la investigación y el análisis de la accesibilidad y experiencia de usuario sobre los parlamentos autonómicos de España, así como sobre el Congreso de Diputados y el Senado.

El ámbito de estudio trata sobre las Tecnologías de la Información y la Comunicación (TIC), ya que interesa saber cuánta ayuda y fiabilidad aportan las herramientas y soluciones disponibles. Por ello, se han seleccionado tres herramientas automáticas, así como una metodología de estudio concreta para llevar a cabo la investigación. Cabe la posibilidad de que las herramientas online y de código abierto puedan proporcionar un análisis erróneo, por lo que los datos que se obtienen en el estudio no son concluyentes ya que no todos los criterios se encuentran consensuados en el análisis de accesibilidad. El mayor inconveniente es que se pueden obtener falsos positivos o negativos en el análisis. Por ende, a día de hoy se hace imprescindible la auditoría de un experto humano que compruebe estos resultados obtenidos. Es por ello que se ha desarrollado un aplicativo web basado en un modelo flexible de APIs, el cual permite una validación de todos los criterios disponibles. Además, este aplicativo contendrá el estudio previo realizado a través de gráficas dinámicas en la que se podrán visualizar los resultados obtenidos por las herramientas automáticas seleccionadas.

Por otro lado, el estudio sobre la experiencia de usuario (UX) se realiza sobre las métricas Core Web Vitals [1], una iniciativa de Google para incluir en una guía aquellos indicadores de calidad para proporcionar una buena experiencia de usuario. Por ese motivo, para la investigación se hace uso del motor de análisis más importante y conocido en la actualidad, *PageSpeed Insight*. Lo que se consigue con esta herramienta es que se obtenga una serie de valores en función de ciertos umbrales, que proporcionan

diversas recomendaciones para mejorar el tiempo de carga y el rendimiento de los casos de uso sobre los que se hace el análisis. Además, para completar la implementación anteriormente mencionada, se incluye en el aplicativo web un apartado para UX, en el cual se implementa la llamada a la API de *PageSpeed Insight*. Estos datos son recogidos de los últimos 28 días y permitirán comprobar de manera más sencilla el tiempo de carga de las páginas y la categorización que reciben por ello.

1.2. Objetivos

El objetivo de este proyecto es evaluar la accesibilidad web y experiencia de usuario en el contexto de los diferentes parlamentos autonómicos. Por ello, las diferentes actividades estarán focalizadas en tres aspectos esenciales: el estudio previo de la legislación vigente, guías y métricas; evaluación de los diferentes parlamentos autonómicos en función de los indicadores necesarios; resultados concluyentes de dicha evaluación basadas como por ejemplo en rankings, calidad de datos, etc; y propuestas de solución en el ámbito de las TIC en base a los resultados obtenidos.

1.3. Planificación del proyecto

El proyecto se ha dividido en diversas tareas para una planificación coordinada de la realización de estas. A continuación, se presenta en un diagrama de Gantt la planificación seguida:



Figura 1.1: Diagrama de Gantt de la planificación del proyecto

Capítulo 2

Estado del Arte

El objetivo de este capítulo es hablar sobre el estado del arte del trabajo de fin de grado. En él se tratan los aspectos fundamentales de los diferentes objetivos establecidos.

2.1. Accesibilidad

La accesibilidad web tiene como objetivo lograr que cualquier tipo de persona, ya sea una persona con alguna discapacidad o no, o independientemente de los conocimientos que tenga y de las características del equipo utilizado, pueda lograr hacer uso de los sitios web, es decir, conseguir webs universales.

2.1.1. Legislación

La "*Directiva (UE) 2016/2102 del Parlamento Europeo y del Consejo sobre la accesibilidad de los sitios web y aplicaciones para dispositivos móviles de los organismos del sector público*"[2], traspuesta a la legislación española mediante el "*Real Decreto 1112/2018 sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público*"[3], obliga a dichos portales y aplicaciones del sector público a ser accesibles y a llevar un seguimiento periódico del cumplimiento del mismo. El Real Decreto (**RD**) **1112/2018** establece que el Ministerio de Política Territorial y Función Pública sea el encargado de realizar el seguimiento y de presentar los informes a la Comisión Europea.

Por otro lado, el Observatorio de Accesibilidad Web (**OAW**) [4] nace en 2010 con el objetivo de ayudar a mejorar el grado de cumplimiento en materia de accesibilidad de los sitios web y aplicaciones para dispositivos móviles de las administraciones públicas (**AAPP**).

La "*Decisión de Ejecución (UE) 2018/1524 de la Comisión Europea, de 11 de octubre de 2018, por la que se establecen una metodología de seguimiento y las disposiciones para la presentación de informes por parte de los Estados miembros de conformidad con la Directiva (UE) 2016/2102*" [5] especifican cómo deben los estados realizar el seguimiento de accesibilidad de los portales, con qué métodos hacer el seguimiento para la verificación, cuáles son los métodos de seguimiento simplificado con herramientas automatizadas y cómo se deben presentar los informes. Además, se indica la muestra de sitios a analizar, así como las páginas que se seleccionan teniendo en cuenta la población del país, que sea una muestra representativa y los ratios de rotación en los sitios web analizados.

La Comisión Europea ha establecido dos métodos de seguimiento que deberán aplicar los Estados miembros: el método de seguimiento simplificado y el método de seguimiento en profundidad. Por lo que, la unión de ambas es necesaria para poder reportar el grado de cumplimiento de los sitios web del sector público español con respecto a los requisitos de accesibilidad. Estas normas están a su vez alineadas con las WCAG 2.1 del World Wide Web Consortium (**W3C**) [6].

2.1.2. Metodología existentes

La OAW aprobó en noviembre de 2020 y marzo de 2021 dos metodologías de estudio para los sitios web. Estas son *La Metodología para el Seguimiento Simplificado UNE-EN 301549:2019* [7] y *La Metodología en Profundidad para Sitios Web UNE-EN 301549:2019* [8], respectivamente.

Además, se debe tener en cuenta que la validación se puede realizar en la fase de desarrollo del proyecto web y/o en la fase de producción del sitio web, una vez finalizado el proyecto.

Metodología para el Seguimiento Simplificado UNE-EN 301549:2019

Se trata de una metodología que se aplica para la revisión del cumplimiento de la accesibilidad de los sitios web en España. La estimación del número de comprobaciones se realiza de forma automática a través de distintos algoritmos y métricas. A continuación se detallan cuatro grandes apartados que tienen que estar presente en todos los análisis realizados con esta metodología:

Selección de la muestra de sitios web. La muestra seleccionada debe dar cumplimiento a la *Decisión de ejecución (UE) 2018/1524*, siendo representativa en el ámbito gubernamental, la temática y la distribución geográfica. Además, para la muestra se tendrá en cuenta las aportaciones que realicen las asociaciones de personas con discapacidad.

Muestra de páginas. De manera automática, mediante un proceso aleatorio, se selecciona el número de páginas que se van a analizar, teniendo en cuenta el tamaño y la complejidad del sitio web, así como los distintos niveles en la estructura de navegación. Los niveles de complejidad son:

- Baja. Muestra de 17 páginas con profundidad 4 y amplitud 4.
- Media. Muestra de 33 páginas con profundidad 4 y amplitud 8.
- Alta. Muestra de 51 páginas con profundidad 5 y amplitud 10.

Además, la muestra siempre contiene la página principal del sitio web.

Por otro lado se analiza la profundidad, siendo esta el número de clics necesarios para llegar hasta el recurso solicitado y la amplitud, que corresponde al número de páginas que se han analizado en cada nivel de profundidad.

Lista de verificaciones. Las verificaciones van a estar definidas dependiendo del nivel de adecuación. El nivel A es el mínimo incluido en las pautas WCAG 2.1 [9]; mientras que

el nivel AA es que se establece en la norma UNE-EN 301549:2019. También se puede categorizar como no válido.

Resultados obtenidos. Se generarán gráficas y valores estadísticos que ayudan a reflejar los resultados obtenidos. Existen otros indicadores como pueden ser las puntuaciones medias para evaluar el estado de los sitios web. Estas son tres: *Puntuación Media de Página (PMP)*, *Puntuación Media del Sitio Web (PMSW)* y *Puntuación Media de Verificación (PMV)*.

Se debe recalcar que es necesario una revisión manual complementando todos los requisitos de UNE-EN 301 549:2019 para determinar el nivel de adecuación real del sitio web.

Metodología en Profundidad para Sitios Web UNE-EN 301549:2019

Esta metodología definirá los requisitos mínimos para que se cumpla la muestra de sitios web a analizar y las métricas a nivel de un sitio web, así como a nivel global. Estas últimas mencionadas no serán de estudio ya que se debe recopilar una cantidad de información bastante significativa para poder hacer el análisis.

Selección de la muestra de sitios web. La sección se divide en subcategorías para que la muestra sea exhaustiva. Se deberán tener en cuenta:

- Que el tamaño de la muestra sea representativo en cuanto al número de habitantes que tenga el estado miembro, así como dependiendo de la complejidad con la que se quiere hacer el análisis del sitio web y el tamaño.
- Se seleccionan aquellos sitios web que conforman el seguimiento simplificado pero que por sus características no se ha podido incluir una muestra en él. Además, se tendrá en cuenta la puntuación obtenida en el seguimiento simplificado, para así seleccionar las de mayor puntuación y realizar un análisis más exhaustivo que pueda corroborar los resultados o determinar problemas no encontrados. El seguimiento en profundidad se hará de manera manual y analizando todos los criterios.
- La selección realizada debe corresponder a una distribución diversa, representativa y equilibrada geográficamente. Además, los análisis que se realizan a las AAPP son en cuatro ámbitos de actuación diferenciados, en este caso son: estatal, regional, local y otros.

Análisis y métricas de un sitio web. Se deben cumplir las siguientes pautas para que el análisis se catalogue como completo:

- Que se cumplan los criterios de conformidad en base a los criterios de cada página, calificándose como conforme, no conforme y no aplica. La calificación no aplica hace referencia a un criterio que no se ha podido evaluar, ya que no existen elementos con las características necesarias para realizar las comprobaciones.
- Que se determine el cumplimiento del sitio web (plenamente conforme, parcialmente conforme y no conforme) en base a los criterios de conformidad de cada página.

- La puntuación media y el porcentaje que obtiene el sitio web después de analizar los dos ítem anteriores.

Una vez realizado los análisis y métricas de los sitios web se recogen los resultados en un informe de revisión de accesibilidad.

2.1.3. Herramientas existentes

Las herramientas automáticas son de gran utilidad en un periodo de tiempo corto para identificar o analizar los problemas de accesibilidad que se presentan en una web. Pero estas no son del todo eficientes y no detectan todos los problemas. Por eso, se recomienda que para completar el proceso de validación se utilice, además de las automáticas, otras herramientas manuales o semiautomáticas.

Además, se debe llevar una revisión periódica que garantice el mantenimiento del sitio web, independientemente de si se añade o modifica contenido. El periodo de estas no pueden superar los tres años.

Herramientas de evaluación automática

Con las herramientas de evaluación automática se detectan numerosos problemas que de otra manera sería muy difícil y supondría demasiado tiempo. Pero no todas las herramientas detectan los mismos problemas e incluso pueden llegar a dar falsos positivos. Cabe destacar que la mayoría de estas herramientas tienen versión online y gratuitas, pero validan según las WCAG 2.0, y algunas en proceso de adaptación para las WCAG 2.1. Existen diversas herramientas, pero en esta ocasión se tratan las más conocidas:

- Servicio de diagnóstico en línea del Observatorio de Accesibilidad Web [10]. Es una herramienta proporcionada de manera gratuita a la administración pública por el OAW. Ya que el estudio se basa en los documentos aportados por el Observatorio de Accesibilidad Web, hubiera sido interesante poder hacer uso de esta herramienta, pero no ha sido posible su utilización porque se debe ser miembro de la administración pública, además de ser parte de la red SARA para poder acceder y hacer uso de ella.
- TAW [11]. Se trata de un servicio online gratuito desarrollado por la Fundación CTIC, que permite hacer un análisis de accesibilidad web sobre los criterios WCAG 2.0, mediante la aportación de la URL y pudiendo seleccionar el nivel de complejidad. Además, envía al correo electrónico un documento más explícito del análisis realizado.
- eXaminator [12]. Es un servicio online gratuito que realiza la evaluación de accesibilidad automática en español mediante los criterios WCAG 2.0. Esta herramienta dejó de estar disponible en el año 2021.
- AChecker [13]. Se trata de un validador gratuito en inglés, que hace su validación igual que la herramienta TAW. El informe se organiza en pestañas diferenciando entre los problemas conocidos, posibles problemas que requieren de una revisión

manual posterior, problemas potenciales que no se han verificado automáticamente y que tienen que ser revisados manualmente, así como los problemas relativos a la validación del código HTML y CSS. Para cada error se indica el número de línea, una descripción del error, un extracto del código y un breve comentario describiendo cómo se puede solucionar.

- SortSite [14]. Es una herramienta online que permite verificar la accesibilidad de una página web tanto por las WCAG 1.0 y 2.0, pero con una muestra de diez páginas. Comprueba accesibilidad, usabilidad, SEO, compatibilidad de navegadores, errores de código, privacidad, validación de estándares, entre otros.
- Wave [15]. Es una herramienta gratuita basada en las WCAG 2.1. La validación se ejecuta directamente dentro del navegador sin ser necesaria la comunicación con el servicio online. Es una herramienta de gran ayuda a la hora del desarrollo del proyecto y la comprobación del mismo.

Herramientas de ayuda en la evaluación manual

La mayoría de estas herramientas son plugins o extensiones para el navegador. Muchas de dichas herramientas se emplean a lo largo del desarrollo del sitio web o se necesita el código fuente para poder analizarlo.

- Web developer toolbar [16]. Extensión gratuita para Chrome, de utilidad para los desarrolladores. Ofrece diferentes funciones como: deshabilitar ciertas características, definir una hoja de estilo de usuario, etc.
- Colour contrast analyser [17]. Contrasta los colores del primer plano y los del fondo.
- Contrast-A [18]. Aplicación web para la combinación de colores.
- WCAG Contrast Checker [19]. Extensión para Firefox y Chrome, para el contraste de color.
- HeadingsMap [20]. Extensión para Firefox y Chrome que genera un índice o mapa de cualquier documento web estructurado.

2.2. Usabilidad

A lo largo de la historia se han definido diferentes descripciones sobre la usabilidad, pero se enfocan de la misma manera. El término usabilidad hace referencia a la capacidad de uso de un software por determinados usuarios de manera sencilla y funcional, es decir, de manera intuitiva, rápida y segura el usuario puede interactuar con el software. Aunque se esté hablando exclusivamente de software esta terminología se aplica a cualquier otro ámbito. Se incluye a esta terminología en la categoría de requerimiento no funcional, ya que debe ser cuantificada mediante medidas indirectas o atributos.

Jakob Nielsen, considerado el padre de la usabilidad, define usabilidad como atributo de calidad que evalúa cuán fácil son de usar las interfaces de usuario. Existen otras definiciones del término usabilidad:

- Calidad de experiencia de un usuario cuando interactúa con un producto o sistema, la cual es medida a través de la relación entre las herramientas y quienes las utilizan. De esta manera se sabe la eficiencia en el uso de los elementos ofrecidos y la efectividad del cumplimiento de las tareas que se pueden realizar.
- La Organización Internacional para la Estandarización "International Organization for Standardization"(ISO) en su norma ISO/IEC 9126 [21] define la usabilidad como "la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso"
- La Organización Internacional para la Estandarización (ISO) en su norma ISO/IEC 9241 [22] define usabilidad como "la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico"

La usabilidad proporciona un factor de gran importancia como es el posicionamiento "Search Engine Optimization" (SEO) [23]. El posicionamiento SEO es un "conjunto de técnicas que se aplican en una página web con el objetivo de mejorar su posición y su visibilidad en los motores de búsqueda de los principales navegadores".

Por otro lado, la usabilidad mide tres factores principales: la eficiencia, que trata de cubrir las necesidades del fin del usuario, la eficacia que procura cubrir en tiempo breve el objetivo del usuario, y la satisfacción que pretende que la navegación por el sitio web del usuario sea sencilla, rápida y eficiente.

Una buena usabilidad repercute tanto en los usuarios como en los propietarios de la web, por ello se consideran diferentes objetivos:

- Diseño claro, sencillos, intuitivos que ayuden al usuario y que se adapten a diferentes dispositivos.
- Tiempos de carga cortos para disminuir la tasa de abandono de los usuarios y la tasa de rebote.
- Información relevante, consistente y fácilmente localizada sobre los productos y servicios ofrecidos.

- Adaptar la tecnología a las diversas necesidades de los usuarios.
- Tener en cuenta la duración del promedio de las sesiones en el sitio web, para conseguir un aumento del tiempo de visitas.
- Mejorar la experiencia y satisfacción de los usuarios.

La importancia de la usabilidad también recae en conseguir un máximo aprovechamiento de los recursos y en no perder usuarios potenciales.

2.3. Experiencia de usuario

La experiencia de usuario (**UX**) se define como un conjunto de elementos relacionados con la interacción del usuario frente a un producto o servicio. Existen factores que influyen en la manera en la que se percibe e interactúa con el servicio. Se debe tener en cuenta que no se trata de algo universal, ya que cada usuario interactúa o reacciona de diferente manera en cada situación que se le presente, por lo que conocer el objetivo del aplicativo y al público al que va dirigido es de gran importancia para diseñar la estructura. Otros factores a tener en cuenta son los test de usabilidad, que se realizan a una serie de usuarios antes de que el aplicativo salga al mercado así como el feedback que genera el cliente final.

El objetivo principal de UX es que el usuario tenga la apreciación de control y manejo del aplicativo. Además, la experiencia de usuario se compone por tres elementos clave, de los cuales dos de ellos ya se han mencionado en apartados anteriores, estos elementos son la accesibilidad, la usabilidad y el diseño. El diseño, no solo se refiere al aspecto visual, sino que incluye también la arquitectura de la información, el diseño adaptable (*responsive design*), la facilidad de navegación y la velocidad de carga del aplicativo.

Capítulo 3

Accesibilidad

En el capítulo 2, se definió la accesibilidad como aquella práctica que debe garantizar que un sitio web sea inclusivo y universal. En este capítulo, se realiza un estudio y evaluación exhaustiva sobre la accesibilidad de diversos sitios web, previamente seleccionados. Además, se elige una metodología y unas herramientas específicas para el mismo. Como resultado se desea obtener un análisis propio sobre cada caso de uso y así poder definir una solución a la problemática encontrada.

3.1. Caso de Estudio

Para este estudio se escogen los enlaces de referencia de sitio web de los parlamentos autonómicos de España, así como de la web de El Congreso de Diputados y El Senado de España. Han sido seleccionadas aleatoriamente teniendo en cuenta el tamaño de la población y la inversión económica que ha hecho cada parlamento en transformación digital, estos son: Parlamento de Canarias [24], Parlamento de Extremadura [25], Parlamento de Valencia [26], Parlamento de Andalucía [27], Asamblea de Madrid [28], Parlamento de Cataluña [29], Parlamento de Galicia [30] y Parlamento de Asturias [31].

Los casos de estudio se clasifican en dos ámbitos: ámbito estatal que incluye el Congreso de Diputados [32] y El Senado de España[33] y ámbito local que hace referencia a los parlamentos autonómicos seleccionados.

3.2. Metodología empleada

Como se menciona anteriormente existen dos metodologías, pero el estudio se centra en *La Metodología para el Seguimiento Simplificado UNE-EN 301549:2019* para el análisis de los casos de estudio. Esta se basa en realizar unas comprobaciones automáticas con la que se verifique, dependiendo del nivel de adecuación y de la muestra de páginas seleccionadas, los criterios que se establecen en las WCAG 2.1.

Cabe destacar que la muestra de páginas se escoge de forma automática mediante un proceso aleatorio, a través de algoritmos, en donde se seleccionan distintos niveles en la estructura de navegación según el grado de complejidad que se desee.

3.3. Herramientas empleadas

Se hace uso de herramientas automáticas para realizar el estudio. Estas son herramientas online y de código abierto, por lo que se comprueba cuánta ayuda y fiabilidad pueden aportar. La metodología empleada se enfoca en las pautas WCAG 2.1, pero estas herramientas y la mayoría con estas características, validan sobre las pautas WCAG 2.0. Esto se debe a que algunas están en proceso de actualización o, si se quiere hacer la comprobación sobre las pautas WCAG 2.1, se debe pagar por usarlas.

Por ello se han seleccionado tres herramientas que se ajustan bastante a la evaluación de los criterios y que han sido mencionadas y reconocidas en la guía de accesibilidad del Observatorio de Accesibilidad Web de España. Estas son: TAW web, AChecker y Sortsite, las cuales sus características fueron explicadas en la sección de Estado del Arte de este documento. Cabe resaltar nuevamente que no se utilizarán herramientas manuales, ya que para poder hacer uso de ellas se necesita ciertos conocimientos avanzados o a un profesional experto en ese ámbito.

3.4. Aspectos generales

El estudio llevado a cabo en este TFG consiste en la creación y cumplimentación de dos tablas.

La primera tabla comprende el título del nombre de cada parlamento seguido de la terminación WCAG 2.1. En ella se completa cada criterio, correspondiente a un principio determinado, dependiendo del valor que haya sido resultante al generar el análisis en las herramientas nombradas.

Existen cuatro principios. El principio *perceptible* trata de mostrar al usuario la información de forma que lo pueda entender. Por otro lado, el principio *operable* se ajusta a que los componentes puedan ser manejados por los usuarios. Mientras que el principio *comprensible* debe mostrar información inteligible para el usuario. Y por último el principio *robusto* trata que el contenido sea estable para que pueda ser interpretado por diferentes tecnologías de asistencia y agentes de usuario.

Esta tabla comprende otras columnas a completar:

- **Nivel de análisis.** El nivel de análisis puede ser A o AA, dependiendo del criterio.
- **Pautas que cumple.** En cada celda definida por un criterio se seleccionará si se cumple el criterio (tendrá un valor de 1), si no cumple el criterio (tendrá un valor de 0), si no se aplica (no se le dará valor) o celdas vacías que serán completadas en las siguientes dos columnas.
- **Posibles pautas que se cumplen pero deben de comprobarse manualmente.** Esta columna hace referencia a criterios los cuales las herramientas automáticas marcan como posible error pero que se deben corroborar de manera manual para comprobar que no se trate de un falso positivo o negativo. Por ello, al no tener una valoración clara sobre estos, se puntúan con un 0,5 , y así no descartar un posible error.

- **No se pueden comprobar automáticamente.** Esta columna marca los criterios por los cuales las herramientas no pueden hacer el análisis. No tendrán valoración.

Por otro lado, al final de cada principio se realiza una media de este, en donde se tendrá en cuenta exclusivamente aquellos criterios que se han podido analizar y han obtenido una puntuación. Además, al finalizar esta columna se encuentra la valoración media global del sitio web. Esta media dará un resultado entre cero y diez. La fórmula utilizada es:

$$\text{Media por Principio} = \frac{\sum \text{del valor obtenido de cada criterio}}{\text{Número de criterios con valor}} * 10$$

$$\text{Media del Sitio Web} = \frac{\sum \text{del valor obtenido de cada 'Media por Principio'}}{\text{Número de principios}}$$

Nota: el 'Número de principios' siempre será 4

De las valoraciones obtenidas de cada tabla, se realizan gráficas y estadísticas para su posterior análisis. De esta manera, los resultados obtenidos ayudan a visualizar cuantos criterios se pueden comprobar gracias a las herramientas automáticas y cuanta fiabilidad dan. Por tanto en este TFG, no se va a aplicar el valor de la media obtenida por ambas fórmulas, ya que estos resultados no son representativos y no se utilizarán para realizar comparaciones entre los casos de estudio.

La segunda tabla comprende el título del nombre de cada parlamento. Esta tabla se ha desarrollado como mera tabla informativa y no contendrá valoración. Ha sido diseñada gracias a la metodología para el seguimiento simplificado publicado por el OAW. En ella se hace la unión de la información contenida en las tablas detalladas por cada nivel junto con la correspondencia con WCAG 2.1 y UNE-EN 301549:2019 para que el resultado sea una tabla genérica lo más completa posible. Las columnas que comprende son:

- **Identificador.** Corresponde a un número
- **Nombre.** Hace referencia al nombre del identificador.
- **Criterio WCAG 2.1.** Cada identificador va a cumplir uno o varios de los criterios WCAG 2.1.
- **Pregunta.** La cuestión que deben responder cada vez que se compruebe el identificador.
- **Pautas que se incumplen.** Aquellos criterios correspondiente a WCAG 2.1 que se incumplen en cada identificador.
- **Posibles pautas que se incumplen pero deben de comprobarse manualmente.** Estas hacen referencia a criterios WCAG 2.1 que las herramientas automáticas marcan como posible error pero que se deben corroborar de manera manual para comprobar que no se trate de un falso positivo o negativo.

Tabla de Análisis de la web del Parlamento de X sobre el marco del Observatorio de Accesibilidad Web						
Identificador	Nombre	Criterio WCAG 2.1	Requisito UNE-EN 301549:2019	Pregunta	Pautas que incumple	Posibles pautas que se incumple pero debe de comportarse manualmente
1.1	Existencia de alternativas textuales	WCAG 1.1.1	9.1.1.1	¿Los elementos no textuales tienen texto alternativo y éste es correcto?		
1.2	Uso de encabezados	WCAG 1.3.1	9.1.3.1	¿Se usan encabezados para reflejar la estructura del documento?		
1.3	Uso de listas	WCAG 1.3.1	9.1.3.1	¿Las listas están correctamente marcadas?		
1.4	Tablas de datos	WCAG 1.3.1	9.1.3.1	¿Las tablas de datos tienen encabezados, información de resumen adecuada, asociación de celdas en caso de ser complejas y sus celdas de datos tienen contenido?		
1.5	Agrupación estructural	WCAG 1.3.1	9.1.3.1	¿Se realiza un marcado adecuado de los párrafos de texto?		
1.6	Separación de contenido y presentación	WCAG 1.3.1	9.1.3.1	¿Se separa adecuadamente el contenido de la presentación, sin usar estilos para transmitir información ni elementos estructurales únicamente con fines presentacionales?		
1.7	Identificación del idioma principal	WCAG 3.1.1	9.3.1.1	¿Se identifica el idioma principal correctamente?		
1.8	Navegación con Javascript accesible y Control de Usuario	WCAG 2.1.1	9.2.1.1	¿Se usa Javascript de forma independiente de dispositivo? Y ¿Puede el usuario controlar los parpadeos, las redirecciones y actualizaciones correctamente?		
		WCAG 4.1.2	9.4.1.2			
1.9	Formularios y etiquetas	WCAG 2.2.1	9.2.2.1	¿Todos los controles de formulario se etiquetan correctamente y se identifican los campos obligatorios en los formularios?		
		WCAG 2.3.1	9.2.3.1			
1.10	Formularios y estructura	WCAG 1.3.1	9.1.3.1	¿Se agrupan estructuralmente los controles y otros elementos de formulario relacionados entre sí?		
		WCAG 3.3.2	9.3.3.2			
1.11	Título de página y de marcos	WCAG 4.1.2	9.4.1.2	¿La página y los marcos tienen un título significativo que identifica su contenido?		
		WCAG 2.4.1	9.2.4.1			
1.12	Enlaces descriptivos	WCAG 2.4.4	9.2.4.4	¿Los enlaces tienen un texto adecuado?		
1.13	Cambios de contexto	WCAG 3.2.2	9.3.2.2	¿Los cambios de contexto se realizan adecuadamente?		
1.14	Compatibilidad	WCAG 4.1.1	9.4.1.1	¿El código es procesable?		
2.1	Identificación de los cambios de idioma	WCAG 3.1.2	9.3.1.2	¿Se identifican los cambios de idioma correctamente?		
2.2	Legibilidad y Contraste	WCAG 1.4.3	9.1.4.3	¿El contraste entre el color del texto y el color de fondo es el suficiente con una utilización correcta de los estilos en línea para las propiedades de espaciado?		
2.3	Maquetación adaptable	WCAG 1.4.10	9.1.4.10			
2.4	Múltiples vías de navegación	WCAG 2.4.5	9.2.4.5	¿La maquetación del sitio web se adapta correctamente a diferentes tamaños de ventana o zoom?		
2.5	Independencia de dispositivo	WCAG 1.3.4	9.1.3.4	¿Se respeta la visibilidad y el orden del foco del teclado, la orientación del dispositivo y los valores correctos de autocompletado en formularios?		
		WCAG 2.4.3	9.2.4.3			
2.6	Navegación consistente	WCAG 1.3.5	9.1.3.5	¿El uso de los enlaces es consistente y el esperado por los usuarios?		

Figura 3.1: Segunda tabla creada.

Basándose en la primera tabla, después de realizar los análisis y ya teniendo datos con los que hacer comparativas, se comprueba que las herramientas usadas y otras con las mismas características, realizan el estudio con una muestra de una única página, en este caso la página inicial de cada web seleccionada. Esto ha generado un resultado poco satisfactorio, ya que no se tiene un conjunto mínimo de páginas que se ajuste a uno de los tres niveles de complejidad para realizar un análisis más exhaustivo.

3.5. Resultado del Caso de Estudio de Accesibilidad

El número total de criterios a analizar es de 49 criterios de nivel A y AA de adecuación. Al realizar los análisis para cada sitio web seleccionado, se percibe que los criterios a evaluar por herramientas automáticas son escasos y siempre son los mismos dentro de cada principio. Además, se distinguen criterios que las herramientas directamente no pueden comprobar, por lo que para conocer cuáles son estos criterios se realiza una recogida de datos en la que se comprueba cuáles son los que no se han podido evaluar y por tanto se deben evaluar manualmente a través de un experto o herramientas que lo permitan, estos son: 1.2.1 Solo audio y video (pregrabado), 1.2.2 Subtítulos (pregrabado), 1.2.3 Descripción de audio o alternativa de medios (pregrabado), 1.2.4 Subtítulos (pregrabado), 1.2.5 Descripción de audio (pregrabado), 1.3.4 Orientación, 1.3.5 Identificar el propósito de entrada, 1.4.2 Control de sonido, 1.4.10 Reflujo, 1.4.11 Contraste sin texto, 1.4.12 Espaciado de texto, 2.1.2 Sin trampa de teclado, 2.1.4 Atajos de teclas de caracteres, 2.2.1 Tiempo ajustable, 2.2.2 Pausar, detener y ocultar, 2.4.5 Múltiples formas, 2.5.1 Gestos de puntero, 2.5.2 Cancelación de puntero, 2.5.3 Etiqueta en nombre, 2.5.4 Actuación de movimiento, 3.2.1 Enfocado, 3.2.2 En la entrada, 3.2.4 Identificación consistente y 4.1.3 Mensajes de estado.

Por otro lado, las herramientas automáticas online y de código abierto comprueban el cumplimiento de un número bajo de criterios, que se van a categorizar en: se cumplen,

no se cumplen y se deben comprobar manualmente. Sobre estos últimos las herramientas hacen una pequeña comprobación pero se requiere de validación de manera manual para no confundirlo con un falso positivo o negativo. Estos criterios, en total 25, de los cuales se obtiene un resultado son: 1.1.1 Contenido sin texto, 1.3.1 Información y Relaciones, 1.3.2 Secuencia significativa, 1.3.3 Características sensoriales, 1.4.1 Uso del color, 1.4.3 Contraste, 1.4.4 Cambiar el tamaño del texto, 1.4.5 Imágenes de texto, 2.1.1 Teclado, 2.3.1 Tres destellos o por debajo del umbral, 2.4.1 Bloques de derivación, 2.4.2 Página titulada, 2.4.3 Orden de enfoque, 2.4.4 Propósito del enlace, 2.4.6 Encabezados y etiquetas, 2.4.7 Enfoque Visible, 3.1.1 Idioma de la página, 3.1.2 Idioma de las partes, 3.2.3 Navegación consistente, 3.3.1 Error de identificación, 3.3.2 Etiquetas o instrucciones, 3.3.3 Sugerencia de error, 3.3.4 Prevención de errores, 4.1.1 Análisis y 4.1.2 Nombre, Rol, Valor. En el listado se recogen los criterios tanto del nivel de adecuación A, siendo estos 16, como del AA, siendo estos 9, ya que son los dos niveles que se van a analizar.

Por el gran número de criterios no evaluados, las notas medias por principio y la nota media total de cada uno de los casos de estudio han sido bajas, lo que son insatisfactorios de acuerdo a las métricas establecidas y por tanto no se pueden concretar como datos concluyentes. Por consiguiente, se ha decidido no tener en cuenta estas notas, sino que se tendrá en cuenta el conteo de aquellos criterios que se clasifiquen dependiendo del nivel de adecuación. Estos son: si se cumplen, si no se cumplen, si se deben comprobar manualmente, si no aplican y si no han podido ser comprobados por la herramienta. De esta manera, se consigue un estudio en el que se verifica cuánta ayuda puede proporcionar aquellas herramientas automáticas online y de código abierto que están destinadas a corroborar la accesibilidad del sitio web que se desea analizar. Asimismo, cabe destacar que se hace imprescindible de un experto humano para verificar los resultados obtenidos.

3.5.1. Parlamento de Canarias

El análisis del sitio web del Parlamento de Canarias comprende un total de 5 criterios de nivel A y 1 de nivel AA que se incumplen, sobre todos aquellos mencionados que se han podido analizar. Estos son: el contenido que no es texto; la información, la estructura y las relaciones transmitidas a través de la presentación; el propósito del enlace; los encabezados y etiquetas que describen el tema o el propósito; las etiquetas o instrucciones de asistencia de entrada; y el criterio de nombre, rol y valor.

Se comprueban otros 8 criterios de nivel A y 6 de nivel AA han dado un posible error pero estos potencialmente se deben volver a analizar de manera manual. Además, cabe destacar que los criterios de idiomas y poder cambiar el tamaño del texto se cumplen. Los demás criterios analizados quedan divididos entre "no aplica" y "la herramienta no los ha podido comprobar".

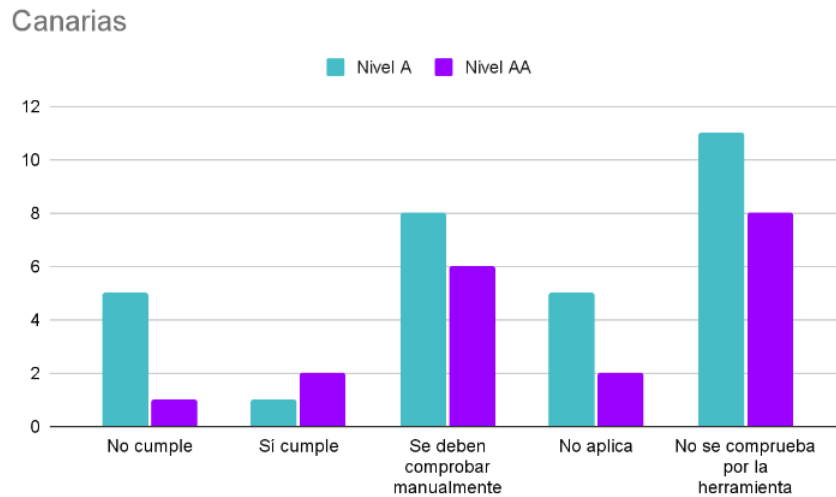


Figura 3.2: Diagrama de accesibilidad del Parlamento de Canarias.

3.5.2. Parlamento de Andalucía

Esta web incumple un total de 4 criterios de nivel A siendo estos: el análisis del principio robusto; el propósito del enlace; el uso del color; y la información y relaciones. De manera sorprendente solo se cumple el criterio de idioma de la página, siendo también de nivel A. Destacar que las herramientas comprueban el cumplimiento de los criterios de nivel AA, pero estos se encuentran categorizados en las otras secciones.

Por ende, los demás criterios de ambos niveles se encuentran en los rangos en los que "no se han podido comprobar", "si se han comprobado pero necesitan que se corrobore manualmente los "no aplica".

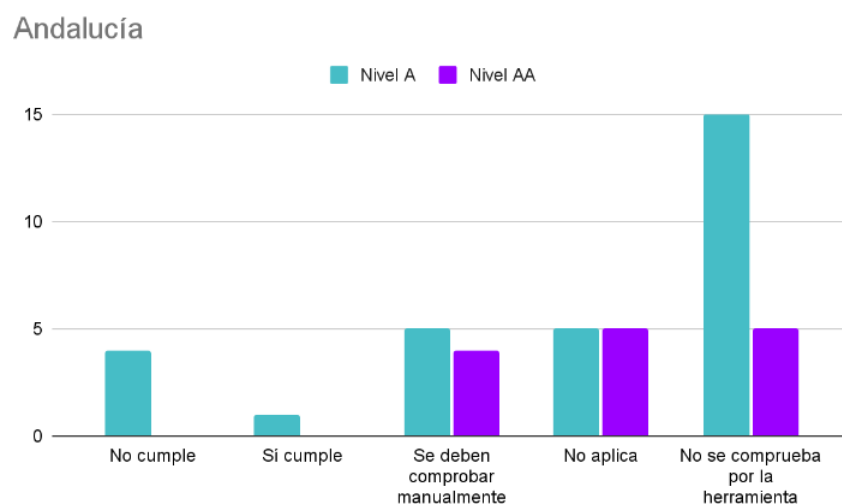


Figura 3.3: Diagrama de accesibilidad del Parlamento de Andalucía.

3.5.3. Parlamento de la Comunidad Valenciana

Al realizar el análisis se comprueba que solo se cumplen 2 criterios, en concreto de nivel A, de entre todos los que se han analizado. Estos son: idioma de la página y etiquetas o instrucciones. Por ende, los demás criterios se encuentran categorizados en criterios que no se cumplen, criterios que no aplican, criterios que no se han podido comprobar o criterios que sí se han comprobado pero necesitan que se corrobore manualmente. La gran mayoría de estos están repartidos entre los dos últimos nombrados. Reinciden aquellos criterios que se incumplen con los mencionados en apartados anteriores. Estos son: el contenido que no es texto como alternativa de texto; el propósito de enlace; el análisis compatible del principio robusto; y de este mismo principio, el nombre, rol y valor.

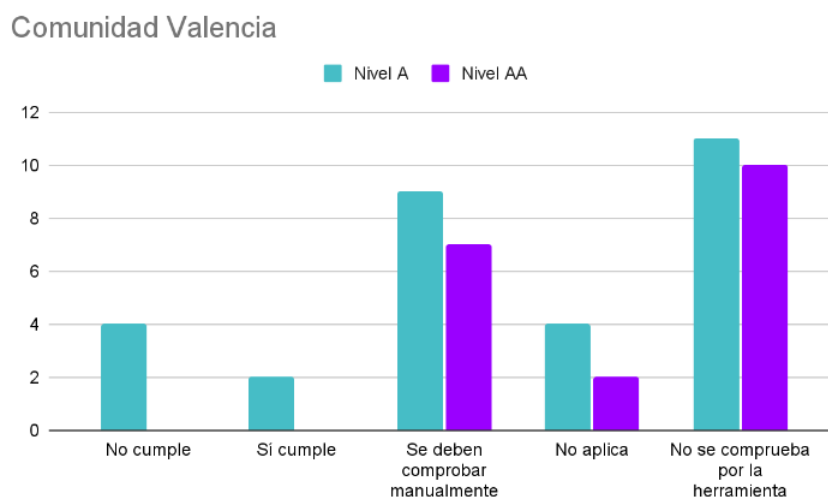


Figura 3.4: Diagrama de accesibilidad del Parlamento de Valencia.

3.5.4. Asamblea de Madrid

Esta web incumple un total de 3 criterios, dos de nivel A y uno de nivel AA. Estos son: información y relaciones del principio perceptible, análisis del principio robusto, y cambiar el tamaño del texto del principio perceptible. Además, se cumplen también un total de 3 criterios con la misma disposición que la anterior. Son: los dos criterios correspondientes a los idiomas y el criterio de etiquetas o instrucciones del principio comprensible.

Por otro lado, los demás criterios de estudio se vuelven a dividir en las otras tres categorías que han sido comentadas en apartados anteriores. Se observa una similitud en el número de criterios que no se pueden comprobar por las herramientas y el número de las que se comprueban pero deben de corroborarse manualmente. Estas últimas nombradas comprenden los siguientes criterios: contenido sin texto, secuencia significativa, uso del color, imágenes de texto, teclado, tres destellos o por debajo del umbral, página titulada, orden de enfoque, propósito del enlace, encabezados y etiquetas, navegación consistente, error de identificación, sugerencia de error, prevención de errores y nombre, rol, valor.

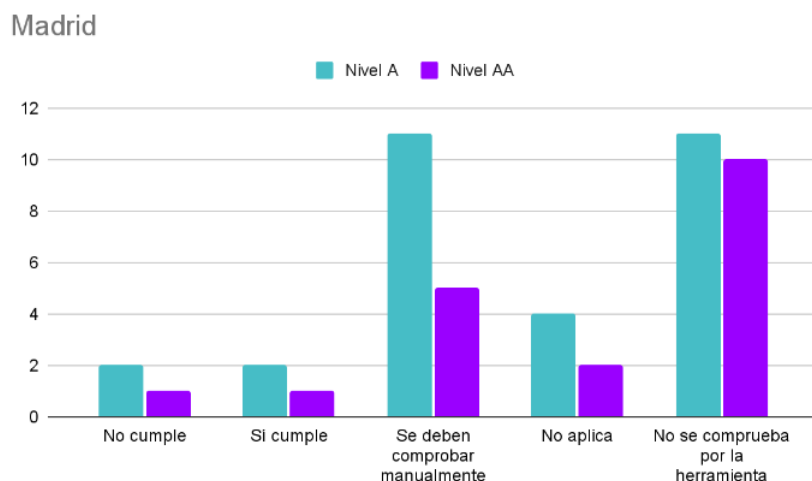


Figura 3.5: Diagrama de accesibilidad de la Asamblea de Madrid.

3.5.5. Parlamento de Cataluña

En este estudio se observa que el mayor número de criterios se encuentran concentrados en la columna identificada como "no se comprueba por la herramienta", por lo que las herramientas nuevamente vuelven a comprobar un número muy bajo de criterios que si se cumplen o no se cumplen. En concreto, se cumplen 2 criterios, siendo estos el idioma de la página y las etiquetas o instrucciones; y se incumplen un total de 6, siendo estos: contenido sin texto, información y relaciones, cambiar el tamaño del texto, propósito de enlace, análisis del principio robusto y nombre, rol y valor.

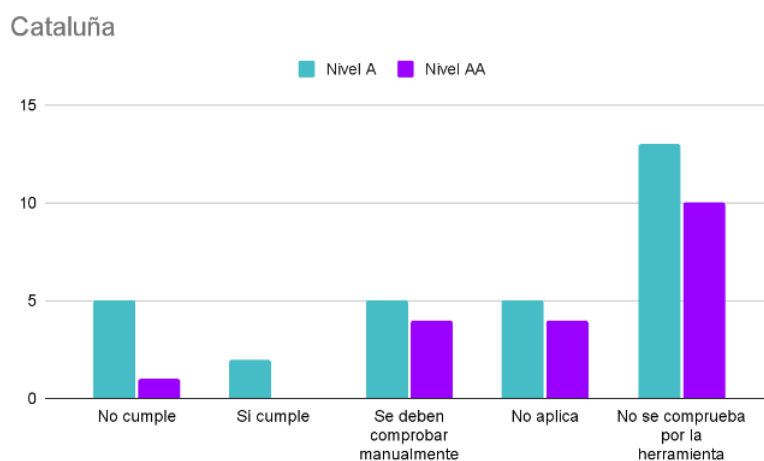


Figura 3.6: Diagrama de accesibilidad del Parlamento de Cataluña.

3.5.6. Parlamento de Extremadura

En este caso, sorprendentemente no se incumplen ningún criterio de entre los que se están analizando, pero exclusivamente se cumplen 3. Por lo que las herramientas con las que se ha realizado el análisis vuelven a catalogar los demás criterios analizados dentro de las categorías de "no se comprueban por la herramienta", "no aplica" y "se deben

comprobar manualmente”. El listado de criterios que se debe comprobar manualmente es: contenido sin texto, características sensoriales, uso del color, cambiar tamaño del texto, página titulada, bloques de derivación, propósito de enlace, encabezados y etiquetas, idioma por partes, error de identificación, sugerencia de error, prevención de errores, análisis y mensajes de estado.

Se debe aclarar que los criterios que se cumplen reinciden en los mismos nombrados en los apartados anteriores.

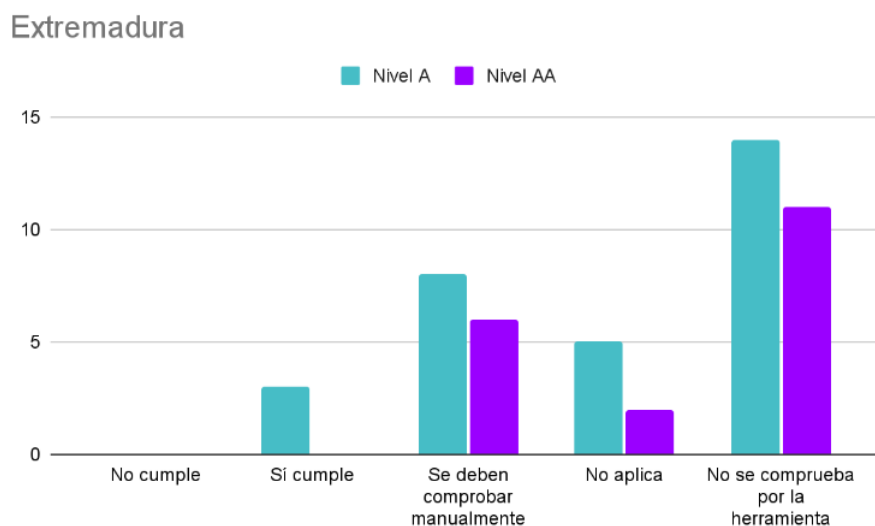


Figura 3.7: Diagrama de accesibilidad del Parlamento de Extremadura.

3.5.7. Parlamento de Galicia

Esta web incumple y cumple un total de 4 criterios por cada uno de estas categorías. Los criterios que se incumplen son: contenido sin texto, propósito de enlace, etiqueta en nombre y análisis del principio robusto. Y los criterios que se cumplen de nivel A son: etiquetas o instrucciones, información y relaciones, y secuencia significativa; y el que se cumple de nivel AA es cambiar el tamaño del texto.

Por ende, el otro grupo de criterios que quedan por analizar, vuelven a encontrarse categorizados dentro de “no se comprueba por la herramienta”.

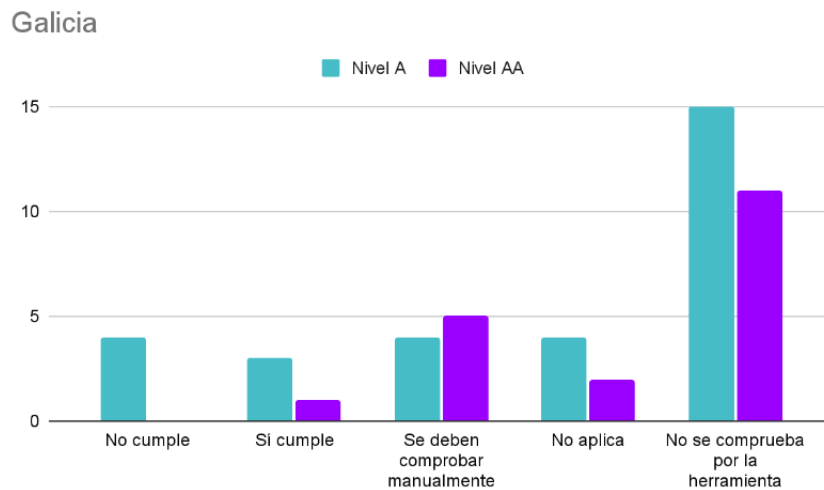


Figura 3.8: Diagrama de accesibilidad del Parlamento de Galicia.

3.5.8. Parlamento del Principado de Asturias

El análisis del sitio web del Parlamento del Principado de Asturias, incumple un total de 6 criterios, en concreto de nivel A, sobre todos aquellos mencionados que se han podido analizar. Estos son: el contenido que no es texto; la información y las relaciones transmitidas a través de la presentación; el propósito del enlace; las instrucciones y etiquetas que describen el tema o el propósito; el criterio de análisis y el de nombre, rol y valor.

Se comprueban otros 7 criterios de nivel A y 6 de nivel AA han dado un posible error pero estos potencialmente se deben volver a analizar de manera manual. Además, destacar que nuevamente el criterio de idioma de la página se cumple. Los demás criterios analizados de los dos niveles de adecuación quedan divididos entre "no aplica" y los que "la herramienta no ha podido comprobar".

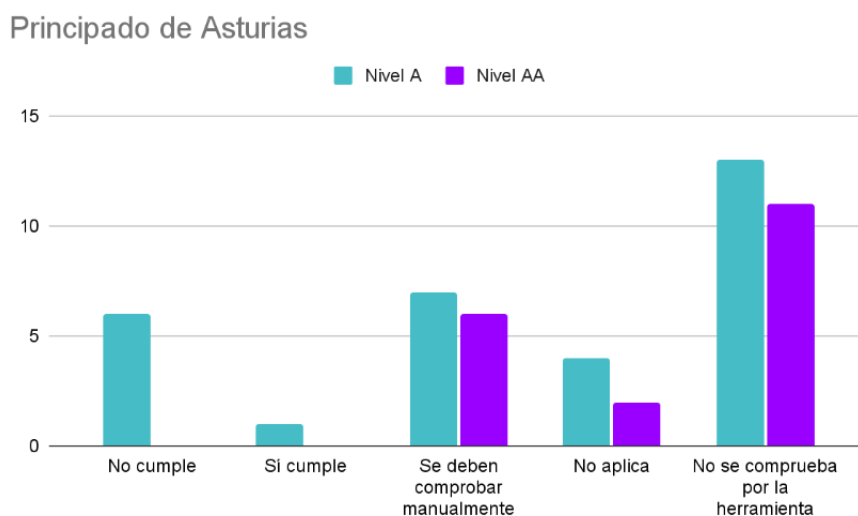


Figura 3.9: Diagrama de accesibilidad del Parlamento de Asturias.

3.5.9. Congreso de Diputados

Esta web incumple un total de 5 criterios de nivel A y 2 de nivel AA. Estos son: información y relaciones, cambiar el tamaño del texto e imágenes de texto del principio perceptible, análisis y nombre rol, valor del principio robusto, propósito del enlace del principio operable y etiquetas o instrucciones del principio comprensible. Además, se cumplen los dos criterios referentes al idioma.

Por otro lado, los demás criterios de estudio se vuelven a dividir en las otras tres categorías que han sido comentadas en apartados anteriores. Se observa una similitud en el número de criterios que no se pueden comprobar por las herramientas y el número de las que se comprueban pero deben de corroborarse manualmente. Esta última clasificación nombrada comprende los siguientes criterios: contenido sin texto, secuencia significativa, características sensoriales, uso del color, teclado, bloques derivación, orden de enfoque, propósito del enlace, encabezados y etiquetas, etiqueta en nombre, navegación consistente, error de identificación, sugerencia de error y prevención de errores.

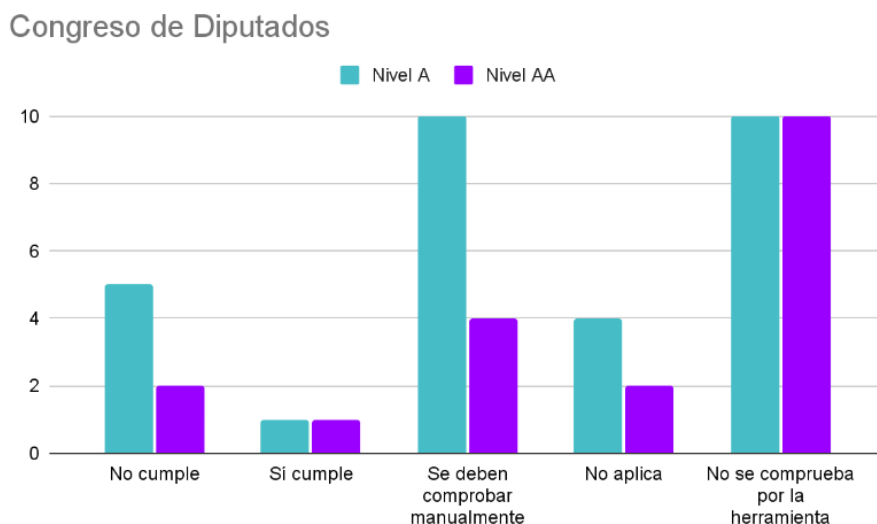


Figura 3.10: Diagrama de accesibilidad del Congreso de Diputados.

3.5.10. Senado de España

En este último caso de análisis, se comprueba que nuevamente el total de los criterios que se cumple o se incumplen no supera a los que no se pueden evaluar por la herramienta. Por ende, como se dijo en la introducción del estudio, no podemos concretar estos datos de estudio como datos concluyentes.

Se cumplen, de manera reincidente, los dos mismos criterios nombrados anteriormente; y se incumplen un total de 7 criterios. Estos últimos son: contenido sin texto, información y relaciones, cambiar el tamaño del texto, propósito del enlace, etiquetas o instrucciones, análisis del principio robusto y nombre, rol, valor clasificado dentro del mismo principio.

Por otro lado, se genera un gran listado de criterios que se deben comprobar manualmente, con un total de 15 criterios. Son: solo audio y video, secuencia significativa,

características sensoriales, cambiar tamaño del texto, imágenes de texto, tres destellos o por debajo del umbral, bloques de derivación, página titulada, orden de enfoque, encabezados y etiquetas, predecible en la entrada, navegación consistente, error de identificación, sugerencia de error y prevención de errores. Además, de existir un número significativo de criterios que "no puede comprobar la herramienta" otro tanto clasificado como "no aplica".

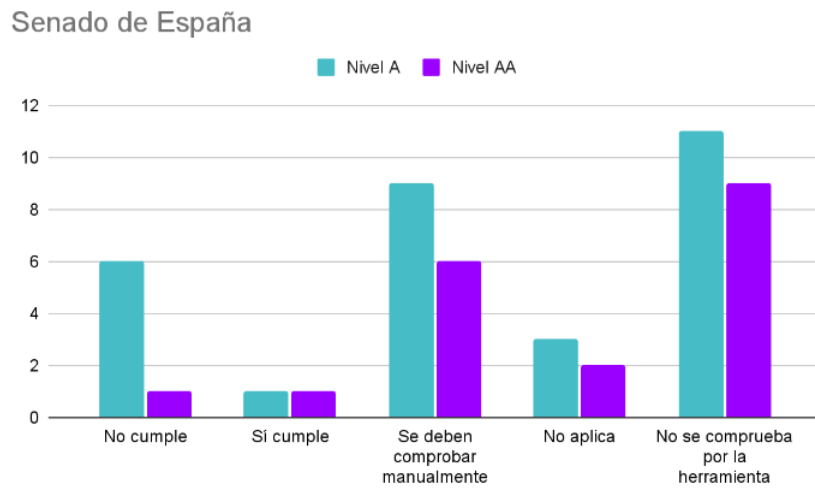


Figura 3.11: Diagrama de accesibilidad del Senado de España.

Capítulo 4

Usabilidad

En el capítulo 2.2, se define la usabilidad como la capacidad de uso del software por usuarios de manera sencilla y funcional. En este capítulo, se presentan los aspectos relacionados y los resultados concretos que se han obtenido.

4.1. Análisis heurístico

La evaluación heurística se trata de un método para determinar la usabilidad de un sistema de interfaces sin usuarios reales, la cual es realizada por una serie de expertos a partir de unos principios de usabilidad establecidos. Este método consiste en: identificar el problema, definir el plan para solucionarlo, realizar dicho plan y analizar el resultado que se obtiene.

El principal objetivo es medir, a través de una serie de comprobaciones, la calidad de la interfaz de cualquier aplicativo en función de la facilidad en la que los usuarios la usen por primera vez, cómo se manejan con ella y cómo la han aprendido. De esta manera obtenemos una serie de resultados que contienen los problemas potenciales del aplicativo a nivel de usabilidad, con la intención de inspeccionarlos, predecir errores futuros y hacer propuestas de mejora.

Existen multitud de listas creadas por expertos del sector que contienen diferentes principios heurísticos [34] para valorar la usabilidad, dependiendo de la interfaz, el dispositivo y el contexto de uso a analizar. Alguno de los principios más conocidos son:

- Las 8 reglas de oro de Ben Schneiderman, (1987).
- Primera versión de los conocidos principios heurísticos por Jakob Nielsen y Rolf Molich, (1990).
- Los 10 principios heurísticos de Jakob Nielsen, (1994).
- La lista de comprobaciones de ítems de Deniese Pierotti, (2004). En esta se añaden tres principios heurísticos más a los ya pautados por Nielsen.

Existen otros principios heurísticos publicados en diferentes libros por D.Mayhew (1992), Larry Constantin (1994) y Keith Instone (1996). Sin embargo, son los principios publicados por Jakob Nielsen y Rolf Molich y posteriormente revisados y resumidos por Jakob Nielsen los que destacan como los 10 principios heurísticos más reconocidos en el mundo del diseño web. En este desarrollo de proyecto de fin de carrera se habla sobre los principios

de Jakob Nielsen y las buenas y malas prácticas que nos encontramos en cada uno de ellos.

Jakob Nielsen *“es una de las personas más respetadas en el ámbito mundial sobre usabilidad en la web”*. Fue considerado el padre de la usabilidad y la definió como *“atributo de calidad que mide lo fáciles de usar que son las interfaces web”*. Las buenas prácticas en el diseño de interfaz web se apoyan en la obra más famosa de este autor, esta obra es *“10 Usability Heuristics for User Interface Design”* [35]. Nielsen menciona que los usuarios prefieren pasar más tiempo en sitios webs que ya conocen y saben que funciona, que en sitios nuevos. Por ello, se decide seguir las convenciones de diseño para que el usuario se centre en el contenido más que en el diseño.

A continuación, se explica cada uno de los principios heurísticos de Nielsen, junto con ejemplos de buenas y malas prácticas.

1. **Visibilidad del estatus del sistema.** Se debe mantener a los usuarios informados de lo que está sucediendo a través del sistema mediante retroalimentación apropiada en un tiempo razonable. Las barras de carga, los indicadores de los procesos de compra, los indicadores de estado o tiempo, son ejemplos de buenas prácticas en las que se mantiene informado a los usuarios en todo momento sobre las acciones que están realizando. Un ejemplo de mala práctica es cuando queremos descargar un elemento de algún sitio web y no te indica el tiempo que tarda en descargarse o por el contrario ha finalizado la descarga del elemento.
2. **Relación entre el sistema y el mundo real.** Se trata de mantener un lenguaje acorde al usuario, es decir, mediante palabras, frases o expresiones familiares en la que la información sea fácil y lógica para los usuarios, además de aparecer en un orden natural y coherente. El ejemplo más claro como buena práctica, es hacer uso del icono de la papelera como símbolo para borrar/eliminar. Este símbolo es mundialmente conocido y se asocia de una manera muy fácil. Si hablamos de malas prácticas un claro ejemplo es cuando se produce un error 404 y la web muestra el mensaje tal cual, sin apenas dar información al usuario de lo que está ocurriendo. En este caso no utilizamos palabras familiares para la gran mayoría de la sociedad, además de no dar explicación de lo que está sucediendo.
3. **Control y libertad del usuario.** Este principio hace referencia a cuando se comete un error o equivocación al realizar una acción, por lo que se le autoriza al usuario a tener la libertad para poder corregirlo. Un ejemplo de buenas prácticas es tener implementado en la web un botón de deshacer, rehacer o cancelar, asegurándose de que esté bien desarrollado. Otro ejemplo es dar la posibilidad de editar el perfil personal después de haberlo creado. Como mala práctica en el aplicativo de Youtube, cuando sale un anuncio en algún video que se quiere reproducir, no te deja interactuar con la barra de control del video, limitando la posibilidad de adelantarlo.
4. **Consistencia y estándares.** Los usuarios no deben preguntarse si existe el mismo significado entre palabras o acciones diferentes. En este caso es mejor seguir los convenios establecidos para el diseño de tu aplicativo. Por ejemplo al intercambiar los colores de los botones de aceptar y rechazar se provoca una gran confusión a los usuarios que hagan uso de estos, ya que al estar tan estandarizado que el color verde se asocia con aceptar y el rojo con rechazar el cambio produce la insatisfacción y el

abandono del usuario en el aplicativo por haberse equivocado. Esto sería un ejemplo de mala práctica en la consistencia y estándares a la hora de diseñar. Por ende, una buena práctica se categoriza por el correcto uso de los estándares que ya están creados o la coherencia a la hora de crear estándares nuevos.

5. **Prevención de errores.** Se trata de anticiparse a que ocurra un error mediante diseños cuidadosos, de esta manera al ponerse en la piel del usuarios evitamos que puedan cometer acciones en las que se vean afectados o comprometidos. Por ejemplo al comprar un billete en la compañía Binter, después de realizar y aceptar la compra, aparece un mensaje en el que el usuario tiene unos minutos para, mediante un aplicativo bancario aceptar o rechazar la compra, de esta manera se realiza una buena práctica previniendo posibles errores. Un ejemplo típico de mala práctica es no indicar qué características o qué longitud debe tener la contraseña del formulario que estás rellenando.
6. **Reconocimiento en lugar de recuerdo.** Se trata de que el usuario no tenga que recordar información de la interfaz, de esta manera se debe hacer visible aquellos elementos, acciones u opciones que sean más necesarias para la interacción del usuario. Por ejemplo, una mala práctica es cuando el usuario se encuentra comprando en una tienda online y en el carrito de compras no se indica el número de artículos que lleva añadidos en la cesta y de la misma manera cuando hace clic en este carrito redirige automáticamente a otra página en la que se muestran los artículos, perdiendo la página que se estaba consultando. Por otro lado, una buena práctica sería el uso de breadcrumbs o migas de pan en un aplicativo web. Los breadcrumbs son enlaces de texto de un tamaño más reducido que el resto que se utilizan para orientar al usuario dentro del sitio web en el que se encuentra.
7. **Flexibilidad y eficiencia de uso.** Este principio hace referencia a tener un aplicativo preparado para cualquier tipo de usuario, refiriéndose a usuarios experimentados o novatos. De esta manera, al incluir “aceleradores” ayudas a los usuarios experimentados a una interacción más rápida y eficiente. Un ejemplo claro de “acelerador” y por ende de buena práctica se trata de un buscador con ventana desplegable en la que se pueda escribir la búsqueda deseada, en lugar de ir mirando una por una todas las opciones del desplegable. De esta manera al ir introduciendo lo que se desea se van reduciendo los campos y de manera más eficiente se va a encontrar lo que se busca. En este caso si hablamos de malas prácticas podemos poner como ejemplo, cuando a la hora de rellenar y validar un formulario existe algún error, todos los campos que se han rellenado en el formulario, estos se eliminaran. De esta manera no se consigue que sea eficiente y rápido de cara al usuario.
8. **Diseño estético y minimalista.** Se debe incluir información relevante en las interfaces. De esta manera se consigue mantener el contenido y el diseño esencial que sea objeto principal del usuario. Por lo que todo el contenido innecesario se debe eliminar para una apariencia limpia y una carga rápida. Por ello, aquellos sitios webs que no diferencian su contenido y además presentan una sobrecarga de información y de elementos de interacción están dentro de malas prácticas. Mientras que un sitio web que solo muestre la funcionalidad o funcionalidades principales que pueda realizar el usuario está dentro de buenas prácticas, como por ejemplo, la funcionalidad de validar la accesibilidad del aplicativo web que se menciona en este proyecto.

9. **Ayudar a reconocer, diagnosticar y recuperarse de errores.** Los mensajes de errores deben estar escritos en un lenguaje claro en el que los usuarios puedan entender qué está pasando, sin utilizar tecnicismos y sin indicar el código de error. También se debe sugerir una solución o “camino de salida”. Si hablamos de buena práctica un ejemplo claro es cuando se rellena un formulario y a la hora de introducir el nombre de usuario y/o contraseña al lado derecho aparece un mensaje de error en el que se comentan los fallos que se comenten al rellenarlo u ofrecen una pista de cómo se debería hacer. Por el contrario, un ejemplo de mala práctica sería la situación en la que accedes a un sitio web, en donde se encuentra un mensaje de error sin filtrar y no se sabe cómo actuar, porque además no queda a la vista ni el menú del sitio web para poder redirigirse a otra página de la web. Es un ejemplo muy simple pero que sucede más veces de lo que se espera.

10. **Ayuda y documentación.** Este principio trata de que un sistema ideal es aquel que no necesita una guía de ayuda o documentación. Pero en el caso de que el usuario necesite ayuda o documentación para realizar algunas tareas del sitio web, esta se debe proporcionar de manera clara y encontrarse en un sitio visible en el que se pueda acceder siempre que se necesite. Un ejemplo de buena práctica es cuando se hace uso por primera vez de un aplicativo que tenga una funcionalidad específica, como por ejemplo un editor web de fotografías o videos y al entrar aparece un mensaje emergente sobre si se desea recibir información de ayuda sobre el uso de este; o a medida que se va navegando por el aplicativo se encuentran iconos de ayuda con una breve descripción de lo que hace. Por ende, si se busca un ejemplo de mala práctica, es todo aquel sitio web en el que la información o acciones que se pueden realizar se presente de manera confusa y no exista ningún tipo de ayuda disponible en la web, ya sea documentación, contacto o servicio de ayuda en línea.

Capítulo 5

Experiencia de Usuario

En el Capítulo 2.3, se definió la experiencia de usuario como un conjunto de elementos relacionados con la interacción del usuario. En este capítulo, se lleva a cabo el análisis, estudio y la presentación de los resultados concretos realizados manualmente sobre experiencia de usuario de los casos de uso previamente mencionados.

5.1. Core Web Vitals

Se realiza un estudio de la experiencia de usuario a los casos de uso definidos previamente a través de Core Web Vitals.

Web Vitals nace de una iniciativa de Google por incluir en una guía aquellos indicadores de calidad para proporcionar una buena experiencia de usuario. A raíz de querer simplificar y centrarse en las métricas más importantes nace un subconjunto de Web Vitals llamadas Core Web Vitals, las cuales son métricas de rendimiento centradas en el usuario. Estas métricas evolucionan con el tiempo, pero el conjunto actual desde 2020 se centra en la velocidad de carga, la interactividad y la escalabilidad visual. Estas son:

- *Largest Contentful Paint (LCP)* [36]. Se traduce como “despliegue del contenido más extenso” y reporta el tiempo que transcurre desde que la página empieza a cargarse hasta que la imagen o el bloque de texto más grande visible dentro de la ventana de visualización se representa en la pantalla. Si hablamos de los umbrales de esta métrica, para permanecer dentro del buen umbral se debe obtener 2,5 segundos o menos. Si se encuentra dentro de 2,5 y 4 segundos se trata de un umbral mejorable, en donde se han detectado diferentes inconvenientes que van a dificultar la experiencia del usuario. Ya pasado los 4 segundos la velocidad de carga es muy lenta y mala.

Para mejorar esta métrica se ven afectados cuatro factores que en la medida de lo posible se deberían optimizar. Estos son: los tiempos de respuesta lentos del servidor, los tiempos de carga de recursos, el bloqueo que produce JavaScript y CSS en la renderización y la renderización que se produce del lado del cliente.

- *First Input Delay (FID)* [37]. Se traduce como tiempo de “demora de la primera entrada” y mide el tiempo que tarda la página desde que el usuario interactúa por primera vez con algún elemento de esta hasta que el navegador procesa los controladores de eventos como respuesta a la interacción. Se centra en eventos de entrada para acciones como hacer clic, pulsar y presionar teclas.

La demora de entrada se produce la mayoría de veces porque el subproceso principal del navegador está realizando otras acciones y no puede responder al usuario en ese momento. Se debe tener en cuenta que esta métrica mide los casos en los que no se detectó un evento, ya que algunas interacciones del usuario no requieren de una acción de eventos.

Para encontrarse dentro de una buena puntuación de FID y que por ende la experiencia de usuario sea satisfactoria, los sitios deben tardar 100 milisegundos o menos. Si el umbral se encuentra entre los 100 y 300 milisegundos se trata de un tiempo que necesita mejoras. Por tanto, si pasa los 300 ms el tiempo es muy lento y se obtiene como resultado una insatisfacción por parte del usuario.

En el caso de que se quiera optimizar la demora de la primera entrada se debe tener en cuenta los siguientes aspectos: dividir las tareas largas, optimizar la parte de interacción de su página y reducir el tiempo de ejecución de JavaScript. Estos últimos se encuentran definidos por W3C como “script JavaScript ejecutado desde una página HTML que se ejecuta en segundo plano, independientemente de los scripts que también pueden haber sido ejecutados desde la misma página HTML”.

- *Cumulative Layout Shift (CLS)* [38]. Se traduce como “cambio acumulativo del diseño” y con ella se mide la estabilidad visual y se cuantifica la frecuencia en la que los usuarios experimentan cambios inesperados de diseño durante toda la vida útil de la página. Si se habla de los umbrales de esta métrica, se considera una buena puntuación cuando el rango se encuentra entre 0,1 segundos o menos. Pasado los 0,1 segundos y hasta los 0,25 segundos se encuentra en un rango en el que se puede realizar mejoras en la experiencia de usuario. Ya pasados esos 0,25 segundos el rango pasa a estar en una medida muy deficiente.

Para mejorar y evitar los cambios de diseño inesperados podemos seguir tres principios: incluir atributos de tamaño en videos e imágenes o reservar el espacio requerido, nunca insertar contenido por encima del contenido actual (exceptuando si es la respuesta de la interacción del usuario) y animar las transiciones de manera que se proporcione contexto y continuidad de un estado a otro.

Se hace uso del percentil 75 para medir el umbral de cargas de páginas para cada una de estas métricas, de esta manera se garantiza que se cumple el objetivo para la mayoría de usuarios.

5.2. PageSpeed Insight

Para realizar el análisis de la experiencia de usuario de los ocho parlamentos autonómicos, el Congreso de Diputados y el Senado de España, se hace uso de la herramienta PageSpeed Insights [39] la cual es compatible con los Core Web Vitals.

PageSpeed Insights (PSI) es una herramienta creada por Google y se utiliza para comprobar el rendimiento real agregado a nivel de página y a nivel de origen tanto para dispositivos móviles como para ordenadores en los últimos 28 días. Además, proporciona consejos sobre cómo mejorar el rendimiento de estas e informes detallados de los errores detectados. La herramienta está disponible tanto en aplicativo web como en API.

Para el estudio previo se emplea la web de PageSpeed Insight para realizar los análisis y posteriormente en el aplicativo web que se ha desarrollado se hace uso de la API [40] mediante llamadas.

5.3. Resultado del Caso de Estudio de UX

Cada vez que se realiza el análisis para uno de los casos de uso, lo primero que aporta la herramienta es una evaluación de las métricas principales de la web con un periodo de recogida de los últimos 28 días. Después aparece un diagnóstico de problemas de rendimiento, que contiene una valoración sobre 0 a 100 del rendimiento del aplicativo. Estas puntuaciones son valores estimados y pueden variar a lo largo del tiempo. Seguidamente aparecen algunas sugerencias que pueden ayudar a mejorar el tiempo de carga de tu página, y otras información de mejoras. Se realizará el análisis mediante PSI sobre la experiencia de usuario de cada uno de los casos de uso.

Este estudio se realiza a nivel de página de origen sobre los sitios webs que se despliegan en ordenadores. Por un lado, la evaluación de las métricas web principales que está dentro de lo que experimentan los usuarios reales contempla un periodo de recogida de datos de los últimos 28 días. Se realiza desde diferentes ordenadores con diferentes conexiones de red y en diferentes versiones de Chrome. Por lo que gracias a esto se recopila una gran muestra que se recoge en los informes de experiencia de usuario de Chrome.

Por otro lado, el diagnóstico de problemas de rendimiento son valores estimados y pueden variar. La puntuación puede fluctuar debido a cambios en las condiciones subyacentes. Alguno de los problemas que se describen a continuación en la página de PageSpeed son:

- Pruebas A/B o cambios en los anuncios que se publican.
- Cambios en el enrutamiento del tráfico de Internet.
- Pruebas en diferentes dispositivos, como por ejemplo: un ordenador de escritorio de alto rendimiento y un portátil de bajo rendimiento. En este caso en concreto, se hace uso de un ordenador portátil para realizar el análisis.
- Extensiones de navegador que inyectan JavaScript y agregan/modifican solicitudes de red.

- Software antivirus.



Figura 5.1: Configuración perteneciente al entorno de trabajo..

5.3.1. Parlamento de Canarias

La evaluación de las métricas web principales recogidas en los últimos 28 días ha sido favorable. Las métricas LCP, FID y CLS se encuentran dentro de un buen umbral, siendo sus valores 1,5 segundos, 2 milisegundos y 0,01 centisegundos respectivamente. Por otro lado, el diagnóstico de problemas de rendimiento comprueba que no existen errores potenciales que afecten directamente al rendimiento y tiempo de carga del sitio web. Este es calificado con un 92 % sobre 100 % de rendimiento, por lo que se trata de una buena puntuación de rendimiento.

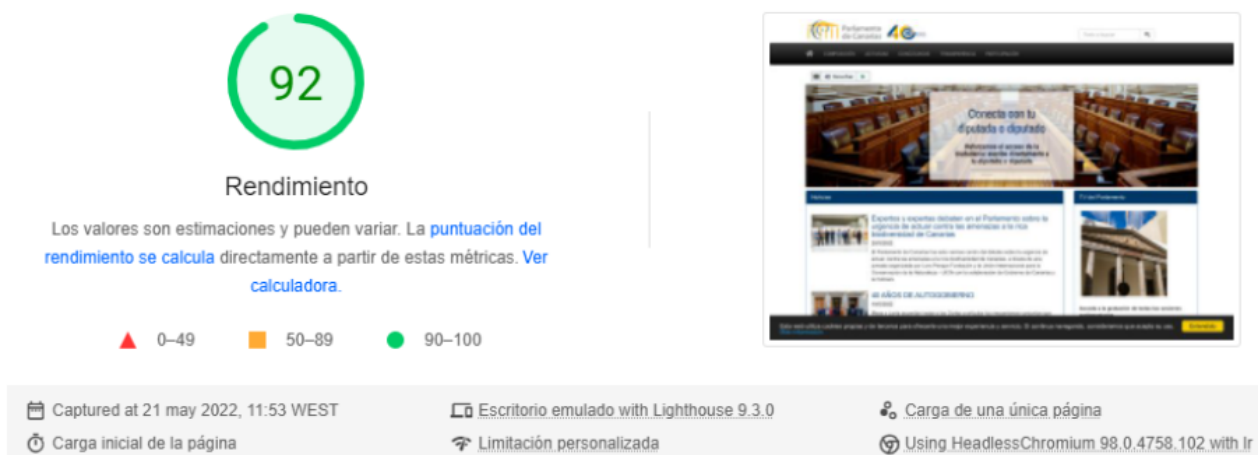


Figura 5.2: Rendimiento de carga del Parlamento de Canarias a día 21 de mayo de 2022 a las 11:53

5.3.2. Parlamento de Andalucía

La evaluación de las métricas principales recogida en los últimos 28 días no es superada. Esto se debe a que el CLS se encuentra en un rango mejorable con un valor de 0,11 centisegundos de tiempo de respuesta que han experimentado usuarios reales. Destacar que las métricas LCP y FID se encuentran con una buena puntuación dentro del umbral.

Por otro lado, el renderizado del mayor elemento del contenido tiene un valor de 2,7 segundos, lo cual es excesivamente alto y se diagnostica como un problema de rendimiento junto con el cambio acumulativo del diseño. Por lo que la valoración total del rendimiento se encuentra dentro de un rango mejorable.

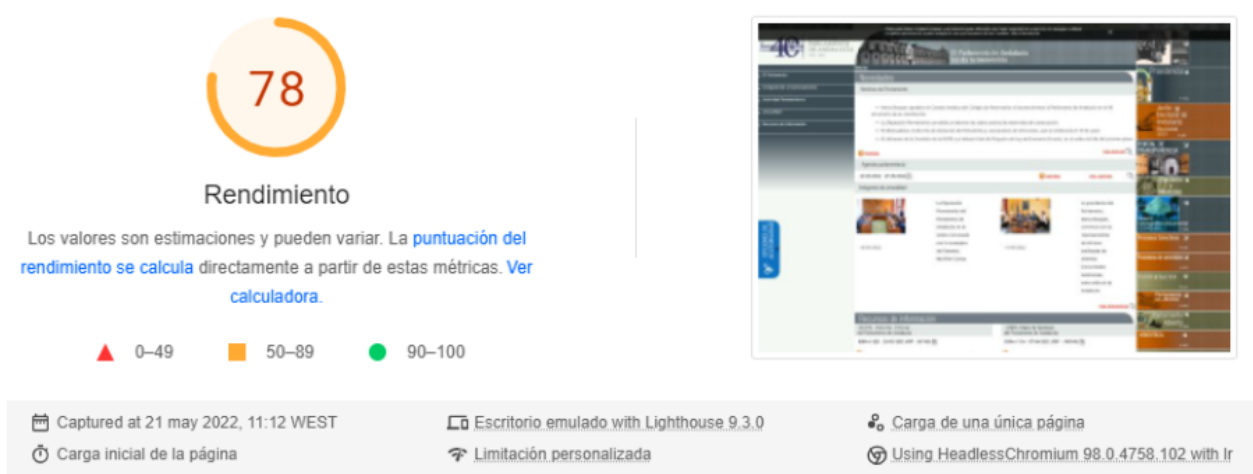


Figura 5.3: Rendimiento de carga del Parlamento de Andalucía a día 21 de mayo de 2022 a las 11:12

5.3.3. Parlamento de la Comunidad Valenciana

En cuanto a la experiencia de usuarios reales, la evaluación de las métricas principales es superada, con valores de 1,6 segundos para LCP, 2 milisegundos para FID y 0,04 centisegundos para CLS.

Si se observa el diagnóstico de problemas de rendimiento la puntuación es de 53 %, por lo que se encuentra dentro de un rango en el que se necesitan mejoras. El renderizado del mayor elemento con contenido es la métrica con peor valoración, en concreto con 6,5 segundos, por lo que se considera un tiempo de carga excesivamente lento. Algunas oportunidades de mejora que recomienda PSI son: publicar imágenes con formatos WebP o AVIF y usar carga diferida con imágenes ocultas y que no aparecen en pantalla, así una vez que todos los recursos críticos hayan terminado de cargarse se reduce el tiempo hasta que la página es interactiva.



Figura 5.4: Rendimiento de carga de las Cortes Valencianas a día 21 de mayo de 2022 a las 11:54

5.3.4. Asamblea de Madrid

La evaluación de de las métricas principales dentro de lo que experimentan los usuarios reales ha sido superada. Se observa que tanto la métrica LCP como la CLS, se encuentran en el límite del umbral para seguir considerándose dentro de un rango bueno, por lo que se debe llevar un control para que no superen la velocidad óptima. En cuanto a la FID tarda 3 milisegundos en procesar los controladores de eventos como respuesta a la interacción, por lo que está dentro de un rango bueno. El rendimiento de la página se encuentra en un 49 % por lo que se trata de una velocidad de carga lenta y la cual debe mejorarse. Algunas de las mejoras son: eliminar los recursos que bloqueen el renderizado; habilitar la compresión de texto, ya que estos se deberían publicar comprimidos para minimizar el total de bytes de la red; y reducir el tiempo de respuesta inicial del servidor. De esta manera, se mejoraría considerablemente la métrica LCP.



Figura 5.5: Rendimiento de carga de la Asamblea de Madrid a día 21 de mayo de 2022 a las 12:16

5.3.5. Parlamento de Cataluña

La valoración de la experiencia en los últimos 28 días de los usuarios reales dentro de la evaluación de las métricas principales es superada, por lo que las métricas LCP, FID Y CLS se encuentran dentro de un umbral favorable con tiempos de 2,1 segundos, 4 milisegundos y 0,06 centisegundos respectivamente.

La puntuación del rendimiento es de 65 %, por lo que se encuentra dentro de un rango de mejoras. Se observa que nuevamente el renderizado del mayor elemento con contenido (LCP) se encuentra con un valor elevado de 3,1 segundos que supera el umbral de conformidad. Las principales mejoras recomendadas por PSI son: publicar imágenes con formatos como WebP y AVIF para reducir el tiempo de descarga, y eliminar recursos que bloqueen el renderizado. También se ofrece otra información relevante para el rendimiento del sitio web cómo: publicar recursos estáticos con política de caché eficaz y evitar un tamaño excesivo de DOM.



Figura 5.6: Rendimiento de carga del Parlamento de Cataluña día 21 de mayo de 2022 a las 12:53

5.3.6. Parlamento de Extremadura

La evaluación de las métricas principales dentro de la que experimentan los usuarios reales se valora positivamente, ya que las métricas de estudio se encuentran dentro del umbral favorable. El tiempo de carga para cada una de las métricas es 1,5 segundos para LCP, 2 milisegundos para FID y 0,02 centisegundos para CLS. Se debe recordar que esta valoración ha sido recopilada en los últimos 28 días.

Por otra parte, el diagnóstico de rendimiento realizado proporciona una valoración del 75 %. Se trata de una puntuación alta dentro de un rango mejorable, por lo que una propuesta de mejora es reducir el tiempo de respuesta inicial del servidor, el cual ayudará a conseguir un rendimiento óptimo.

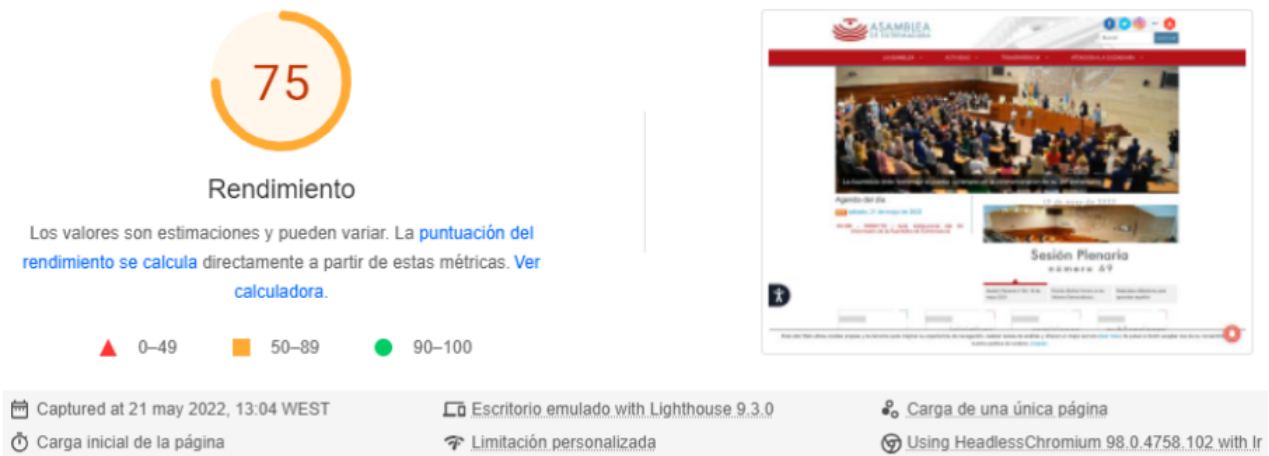


Figura 5.7: Rendimiento de carga del Parlamento de Extremadura día 21 de mayo de 2022 a las 13:04

5.3.7. Parlamento de Galicia

La evaluación de las métricas principales es favorable. Los tiempos de carga de página para cada una de las métricas evaluadas son de 1,4 segundos para LCP, de 2 milisegundos para FID y 0,06 centisegundos para CLS. Todas se encuentran en un umbral bueno de carga.

Por otro lado, la valoración del rendimiento es del 77%, que se encuentra en un umbral mejorable cerca de obtener un rendimiento óptimo. Las mejoras más considerables para obtenerlo tienen que ver con cómo se publican las imágenes, el tamaño de estas y la codificación. Recordar que esta valoración fluctúa debido a limitaciones personales del entorno en el que se hace el estudio como se menciona en la introducción.

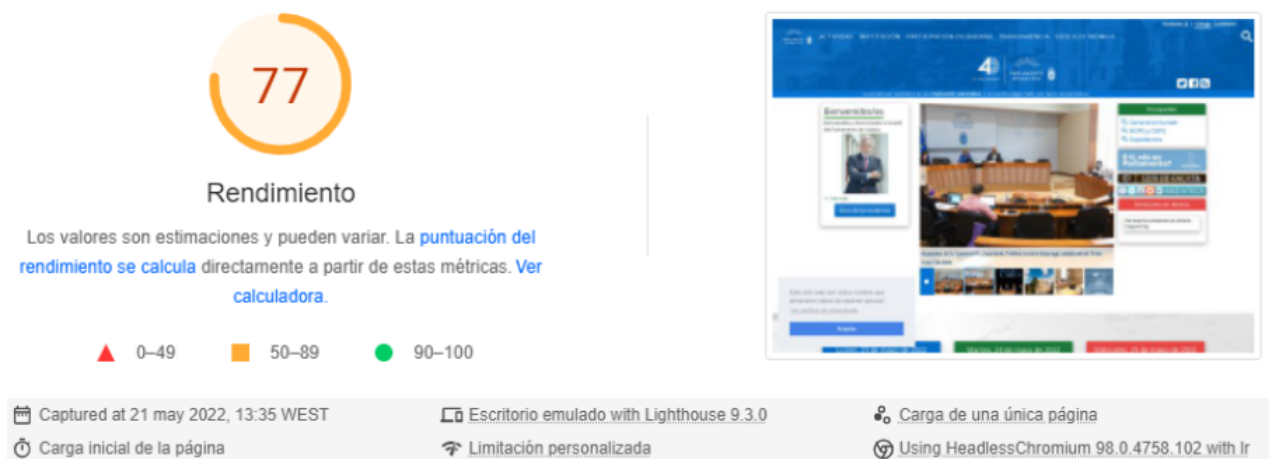


Figura 5.8: Rendimiento de carga del Parlamento de Galicia día 21 de mayo de 2022 a las 13:35

5.3.8. Parlamento del Principado de Asturias

En cuanto a este caso de uso concreto, se comprueba que la evaluación de las métricas principales de la experiencia de los usuarios reales no ha sido superada debido al cambio acumulativo del diseño. Esta métrica se encuentra en un umbral muy desfavorable debido al tiempo de carga que ha obtenido, el cual es de 0,87 centisegundos. Además, la métrica LCP se encuentra en los límites del umbral mejorable a un paso de considerarla también desfavorable, ya que ha obtenido un tiempo de carga de 3,9 segundos. Como punto positivo la métrica FID obtiene una muy buena puntuación dentro del umbral positivo, de unos 2 milisegundos en tiempo de carga.

Las mismas métricas desfavorables que se mencionan anteriormente son las que afectan a la puntuación que recibe el rendimiento del sitio web, el cual recibe una calificación estimada del 58%. Se recomienda eliminar los recursos que bloqueen el renderizado de la página, a través de los elementos de JavaScript y CSS críticos insertados y posponer aquello que no son esenciales. También se recomienda reducir el tiempo de respuesta inicial del servidor, publicar recursos estáticos con una política de caché eficaz y evitar el tamaño excesivo de DOM. Esto ayudará a mejorar el rendimiento de la página web.



Figura 5.9: Rendimiento de carga del Parlamento del Principado de Asturias día 21 de mayo de 2022 a las 13:52

5.3.9. Congreso de Diputados

La evaluación de las métricas principales resulta no superada. Esto se debe principalmente a que la métrica de despliegue del contenido más extenso se encuentra dentro del umbral mejorable cerca del límite, con un tiempo de carga de 3,5 segundos. También se ve afectado por la métrica CLS, ya que esta se encuentra al límite entre el umbral bueno y el mejorable, con un tiempo de carga de 0,095 centisegundos. En cuanto a la métrica FID, se encuentra dentro del umbral favorable.

Por otro lado, el diagnóstico de rendimiento califica este con un 77 %, incidiendo en el renderizado del mayor elemento con contenido como la métrica que más afecta a la calificación del rendimiento. Como punto positivo, el valor de carga de la métrica CLS dentro de este diagnóstico consigue un tiempo de carga de 0,03 centisegundos, encontrándose así dentro de un umbral favorable.

Algunas de las mejoras sugeridas para los elementos más críticos son: eliminar los recursos que bloquean el primer renderizado de la página y las mejoras destinadas a las imágenes, como el formato en que son publicadas, la codificación de estas y la prioridad de carga para aquellas que aparecen en pantalla.



Figura 5.10: Rendimiento de carga del Congreso de Diputados día 21 de mayo de 2022 a las 14:28

5.3.10. Senado de España

La evaluación de las métricas web principales recogidas en los últimos 28 días ha sido favorable, consiguiendo tiempos que se encuentran dentro de un buen rango de carga de la página. Los tiempos obtenidos han sido 1,3 segundos para LCP, 2 milisegundos para FID y 0,07 centisegundos para CLS.

Además, se refleja un rendimiento óptimo, consiguiendo un 90 % de valoración. Esto supone que no se encuentren mejoras críticas que afecten a la puntuación del rendimiento dentro del diagnóstico de problemas.

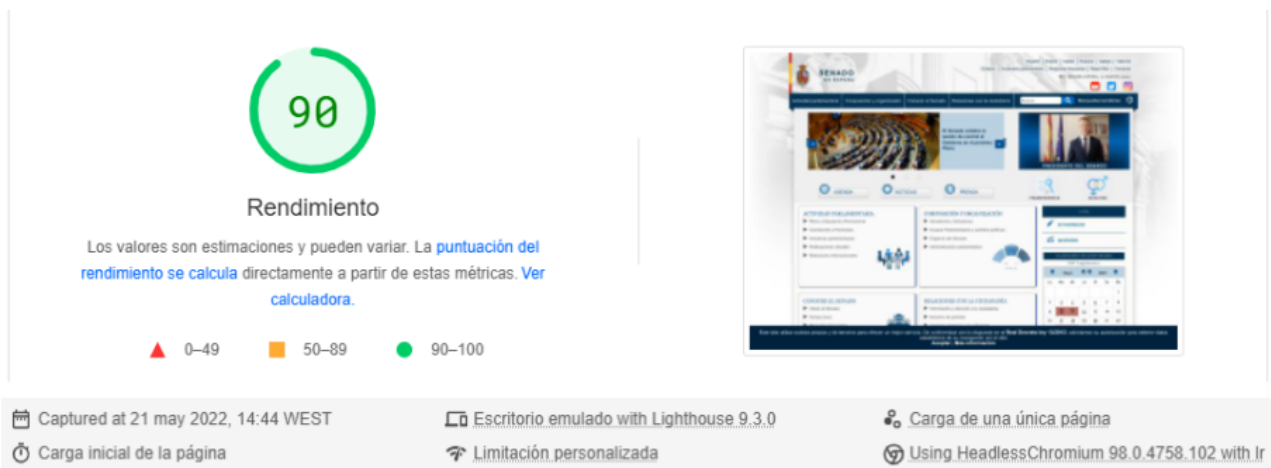


Figura 5.11: Rendimiento de carga del Senado de España día 21 de mayo de 2022 a las 14:44

Capítulo 6

Aplicativo Web

En el Capítulo 3 y en el Capítulo 5 se realizó el análisis y estudio sobre la accesibilidad y la experiencia de usuario para el caso de estudio previamente definido. Este capítulo trata el desarrollo de la solución propuesta para ambos capítulos, incluyendo qué tecnologías y herramientas se han utilizado, así como el desarrollo y diseño del aplicativo web. A continuación, se muestra una gráfica de la estructura de los componentes de la solución:

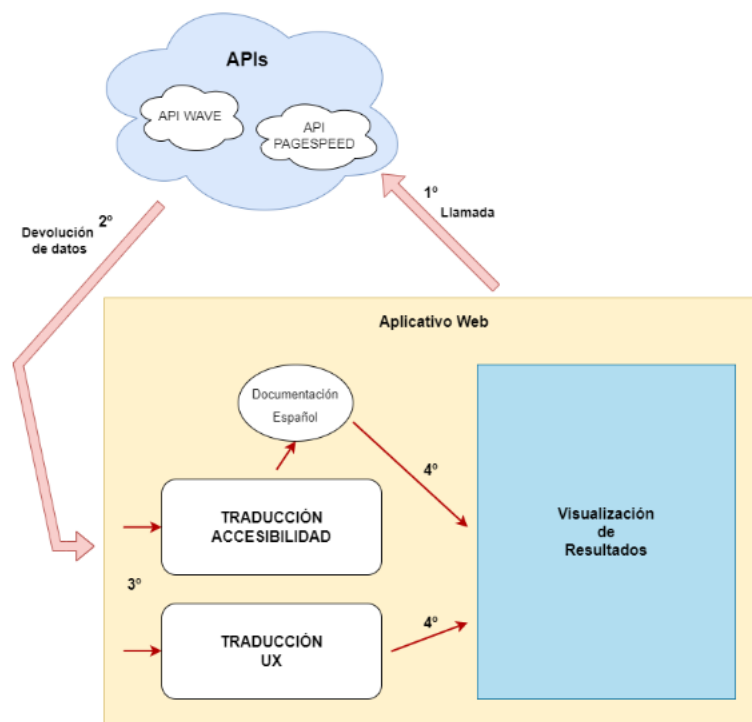


Figura 6.1: Estructura componentes de solución.

6.1. Tecnologías y herramientas

Se hace uso de diversas herramientas y tecnologías para el desarrollo del aplicativo web. A continuación se mencionan dichas herramientas utilizadas:

- **Visual Studio Code** [41]. Se ha decidido utilizar el editor de código Visual Studio Code para el desarrollo del aplicativo. Fue desarrollado por Microsoft para diferentes

sistemas operativos, poniéndolo bajo licencia gratuita y de código abierto. Además, incluye soporte para la depuración, resaltado de sintaxis, finalización inteligente de código, refactorización de código y control integrado de Git. Este último ha sido de gran utilidad, ya que se ha creado y sincronizado el proyecto en GitHub.

- **GitHub** [42]. Se trata de una plataforma que, mediante el sistema de control de versiones de Git, puede alojar proyectos desarrollados por cualquier persona. Su finalidad es almacenar todos los cambios realizados en los diferentes ficheros del proyecto que se esté desarrollando. En concreto, se utilizará esta herramienta para alojar el aplicativo que se ha desarrollado.
- **NPM** [43]. Se trata de un gestor de paquetes el cual puede instalar, actualizar y desinstalar librerías que se especifiquen. Se utiliza para agregar dependencias necesarias para el desarrollo del aplicativo web.
- **Angular** [44]. Angular es el framework que se utiliza para el desarrollo del proyecto. Está desarrollado en TypeScript, es de código abierto y se utiliza para crear y mantener aplicaciones web de una sola página. Este framework se basa en “componentes”, los cuales se usan para hacer el binding de los datos. Además, se hace uso de la herramienta de línea de comandos Angular CLI, la cual nos permite añadir componentes y previsualizar de forma instantánea la aplicación. Se hace uso de la última versión estable de Angular, en concreto la versión 13

Los lenguajes de programación utilizados para este proyecto son:

- TypeScript. Se trata de un lenguaje de programación fuertemente tipado que se basa en JavaScript, que ejecuta las aplicaciones en el lado del cliente o del servidor. En nuestro caso, se trata de la parte cliente.
 - HTML. Se trata del lenguaje de marcado más conocido y utilizado para elaborar páginas web.
 - CSS. Se trata de un lenguaje de diseño gráfico para crear y definir la apariencia o estilo de un documento estructurado escrito en un lenguaje de marcado.
-
- **Librería: Angular Material** [45]. Se trata de una librería de estilos desarrollada por el equipo de Angular, que está basada en la guía de diseño de Material Design. Material Design es un conjunto de especificaciones sobre un estilo y diseño creado por Google para ayudar a los equipos de desarrollo a crear experiencias digitales de alta calidad.
 - **Librería: ngx-charts** [46]. Se trata de una librería para representar gráficas dinámicas. En concreto, se hace uso de la gráfica de barra para mostrar el número de criterios incumplidos por los diferentes rangos de análisis sobre accesibilidad sobre una URL que se introduzca.

- **Librería: RxJS** [47]. Se trata de una librería para componer programas asíncronos y que se basan en eventos mediante el uso de secuencias observables. Combina diferentes patrones de diseño como el patrón observable y el patrón iterador, además de utilizar conceptos de programación funcional para modelar la estructura que maneja la secuencia de eventos.
- **API de WAVE**. WAVE es un conjunto de herramientas de evaluación que ayuda a hacer que el contenido web sea más accesible. Esta herramienta identifica un gran número de errores de accesibilidad y de las Directrices de Accesibilidad al Contenido Web (WCAG). La API se pone a disposición por la WebAIM. La llamada a la API permite el análisis de accesibilidad automatizado y remoto de páginas web utilizando el motor de procesamiento WAVE.

Para hacer uso de ella se debe realizar un registro para obtener la clave API. Posteriormente, se realiza una solicitud GET que contiene dicha clave, la URL que se quiera pasar y otros parámetros de configuración adicionales disponibles, los cuales nos permiten personalizar la solicitud y los datos que se recibe.

- **API de PageSpeed Insight**. Se trata de una API que informa sobre “el rendimiento de las páginas tanto en dispositivos móviles como en ordenadores y ofrece sugerencias para mejorarlas” [48]. La API facilita datos de experimentos, que son datos recogidos en un entorno controlado para depurar problemas de rendimiento; y datos de campo, que son aquellos relacionados con la experiencia de usuarios reales. Ambos datos recogidos sobre las páginas que deseamos analizar.

En el proyecto se utiliza para medir el rendimiento de cada caso de uso de manera dinámica, además de obtener sugerencias sobre cómo mejorar algunos aspectos de cada uno. Para ello se debe adquirir la clave de API y posteriormente, se realiza la petición GET que contiene la URL que se desea analizar, la clave obtenida y diversos parámetros de configuración opcionales.

- **Firestore** [49]. Es una plataforma de Google en la nube para el desarrollo de aplicaciones web y móviles. Consta de diversas herramientas divididas en cuatro grupos principales: desarrollo, crecimiento, monetización y análisis. En concreto, se hace uso de la herramienta hosting que se encuentra en el grupo de desarrollo. Firestore ofrece un servidor para alojar las aplicaciones de manera segura y estática. Proporciona certificados de seguridad SSL y HTTP2 de forma automática y gratuita para cada dominio.

6.2. Diseño del Front-End

Antes de empezar a maquetar la información en el aplicativo web, se realiza con la herramienta Figma [50] un wireframe como guía visual para representar la estructura del sitio web. Se hace uso de la herramienta Color Hunt [51] para decidir una paleta de colores que será representativa para el aplicativo web. En cuanto a la maquetación del sitio web se hace uso de los elementos de la librería de estilos Angular Material,

basada en la guía de diseño de Material Design. A continuación, se explica cada una de las páginas que componen el sitio web.

- La página de inicio contiene los botones principales que redirigen a los demás apartados de estudio en el proyecto.

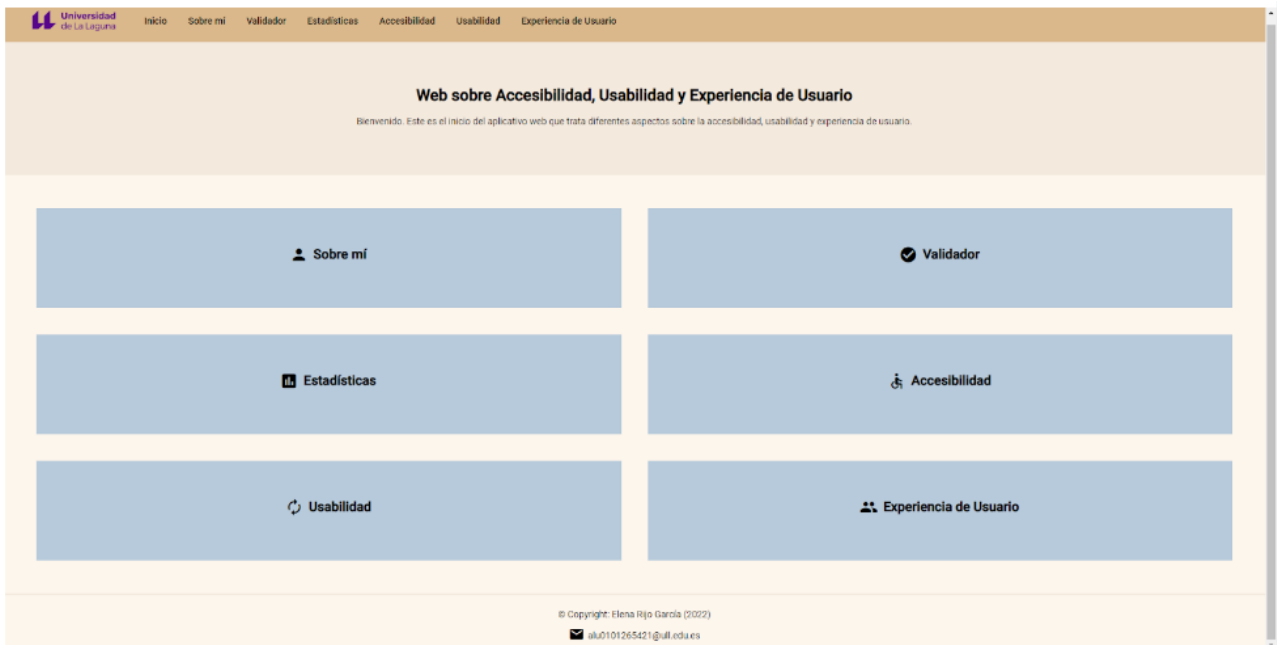


Figura 6.2: Página de inicio.

- En el apartado “Sobre mí” se expone la presentación de la autora del desarrollo.
- El diseño de los apartados “Validador” y “Estadísticas” se muestra en los capítulos “6.3 Desarrollo de accesibilidad” y “6.3.3 Estadísticas” respectivamente. A través del “Validador” se va a comprobar la accesibilidad de un sitio web; y las “Estadísticas” muestran de manera dinámica el estudio previo que se realizó.
- El apartado “Accesibilidad” contiene los aspectos relevantes tratados en el estudio realizado, es decir, contiene la legislación vigente, las metodologías existentes y cuál es la que se utiliza, y las herramientas actuales de las que se hace uso.

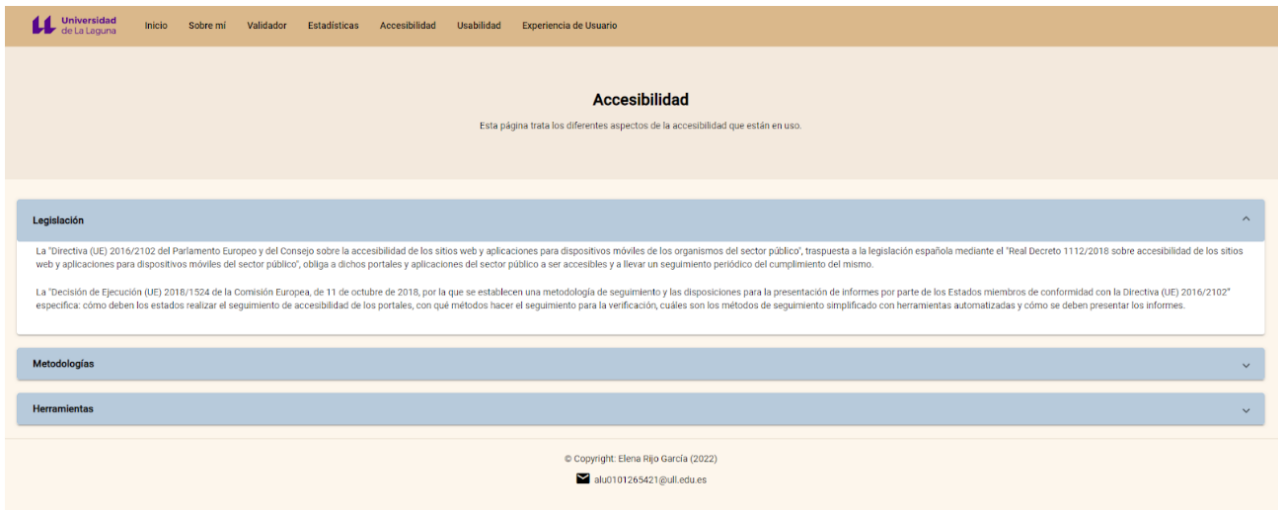


Figura 6.3: Página de accesibilidad en el que se muestra el contenido del apartado legislación.

- El apartado “Usabilidad” contiene el estudio previo realizado en este proyecto, en el cual se tratan los temas de “¿qué es la usabilidad?”, “análisis heurístico” y “los 10 principios de Jakob Nielsen”.

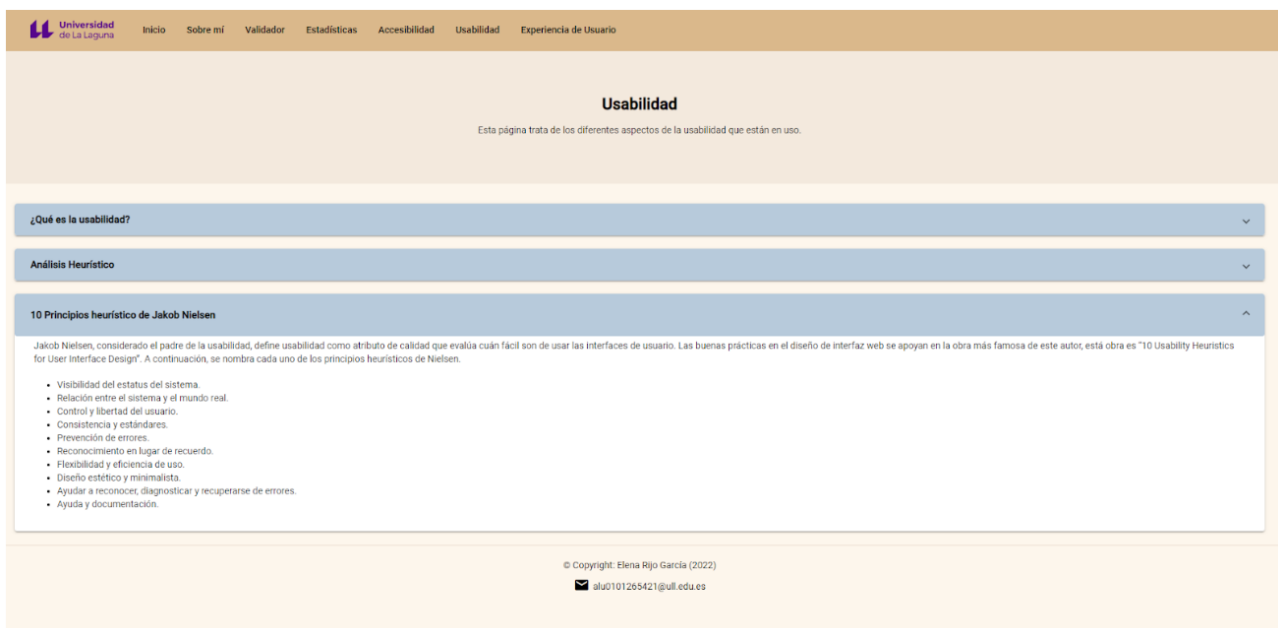


Figura 6.4: Página de usabilidad en el que se muestra el contenido de los 10 principios de Jakob Nielsen.

- El diseño del apartado “Experiencia de Usuario” se muestra en el capítulo “6.4 Desarrollo de experiencia de usuario”. De manera dinámica se muestran los resultados obtenidos por el análisis de UX a través de la API de PageSpeed Insight.

6.3. Desarrollo de accesibilidad

En este apartado se explica la solución de desarrollo propuesta para el estudio de accesibilidad de los casos de uso seleccionados en el Capítulo 3. Se puede encontrar el código fuente alojado en un repositorio de GitHub [52].

6.3.1. Estructura del aplicativo

A partir del estudio previo realizado en el Capítulo 3, se propone el desarrollo de un aplicativo web basado en un modelo de APIs, donde en concreto se implementa la llamada a la API de WAVE, la cual permite una validación de todos los criterios disponibles.

Lo primero que se hace es crear un servicio propio el cuál se va a encargar de traer los datos de análisis y acceder a ellos para poder tratarlos. Primero se obtiene la clave y la URL base de la API para poder acceder. Dentro de la clase, se crea un método al cual se le debe pasar una URL determinada y a través de una petición HTTP, en concreto el método de solicitud GET, se hace una llamada a la API. Esta llamada está compuesta por la URL base de la API, la clave obtenida, la URL que se desee analizar y el nivel de reporte que se quiera obtener de los datos devueltos. En concreto, se le asigna un nivel 2 ya que permite estudiar con más detalle la información obtenida de los criterios analizados.

```
analizarWeb(url: string): Promise<IAnalisisResult> | null {
  try {
    return lastValueFrom(
      this.http
        .get<IAnalisisResult>(
          `${this.URL_BASE}/request?key=${this.API_KEY}&reporttype=2&url=${url}`
        )
        .pipe(take(1))
    );
  } catch (error) {
    console.error(error);
    return null;
  }
}
```

Figura 6.5: Método para llamada a la API Wave.

Los datos devueltos por la API son en formato JSON. Por tanto, en el mismo servicio se crean las estructuras que van a almacenar dichos datos. Estas estructuras van a permitir posteriormente modelar la información que se requiera y para ello 6 interfaces diferentes como estructuras. La primera interfaz contiene la distribución principal del JSON, que se divide en categorías y estadísticas. En nuestro caso, se analizan las categorías por lo que se crea otra interfaz que contenga la estructura de estas. A continuación, se explican cada una de las interfaces utilizadas:

- **IAnalisisResult:** contiene la distribución principal del JSON, que se divide en categorías y estadística.
- **ICategory:** contiene la estructura interna de las categorías.

- **IDocResult**: esta interfaz contiene un título y un conjunto de guidelines. El título corresponde a una breve descripción por la que se adjuntan los criterios y los guidelines almacenan las pautas o criterios correspondiente a ese título. Esto ayuda a clasificar cuáles son los criterios que componen un título que se ven afectados.
- **IGuideline**: guarda el nombre que recibe el criterio.
- **ICategoria**: contiene el tipo de categoría que correspondiente y las pautas que se incumplen.
- **ICriterio**: esta interfaz almacena la información necesaria para cada criterio. Cabe destacar que se almacena la descripción, la cual brevemente explica el significado del criterio. Esta descripción se ha añadido manualmente para poder comprender mejor el criterio que se ve afectado en el análisis de accesibilidad.

```

export interface IAnalysisResult {
  categories: {
    error: ICategory;
    alert: ICategory;
    feature: ICategory;
    structure: ICategory;
    aria: ICategory;
  };
  statistics: {
    pagetitle: string;
    pageurl: string;
    time: number;
    creditsremaining: number;
  };
}
export interface ICategory {
  description: string;
  count: number;
  items: {
    [key: string]: {
      id: string;
      description: string;
      count: number;
    };
  };
}
export interface IDocResult {
  guidelines: IGuideline[];
  title: string;
}
export interface IGuideline {
  name: string;
}
export interface ICategoria {
  tipo: string;
  pautas: {
    titulo: string;
    criterios: ICriterio[];
  }[];
}
export interface ICriterio {
  id: string;
  title: string;
  descripcion: string;
  link?: string;
}

```

Figura 6.6: Interfaces de accesibilidad.

Una vez se tenga la estructura de los datos, se comienza a modelar la información obtenida. Para ello se crea un componente llamado api-componente, que en primer lugar contiene el método para analizar la web. Este método debe ser asíncrono, por lo que hacemos uso de las funciones que contienen las promesas, ya que ayudan a representar resultados de una operación asíncronica como si fueran síncronos. En él se implementa la llamada al servicio y una vez devuelta la estructura de datos se pasa a modelar la información.

Los resultados que se devuelven a nuestro aplicativo son en inglés, por lo que se decide desarrollar la función de traducción. Previamente se ha creado la documentación necesaria en español, para completar la función antes mencionada.

```
async analizarWeb() {  
  
  this.resultadosAnalisisWeb = await this.apiService.analizarWeb(this.direccion);  
  if (!this.resultadosAnalisisWeb?.categories) return;  
  
  const categories = Object.values(this.resultadosAnalisisWeb.categories);  
  
  this.resultadosTraducidos = categories.map((category) => {  
    const tipo = category.description;  
  
    const items = Object.values(category.items);  
  
    const pautas = items.map((item) => {  
      const id = item.id;  
      const criterios = this.apiService.obtenerPautasDeID(id);  
  
      return { id: id, criterios: criterios, titulo: item.description };  
    });  
    return { tipo, pautas };  
  });  
};
```

Figura 6.7: Implementación principal del análisis de accesibilidad.

Además, se añade una funcionalidad extra y esta es modelar los datos necesarios para representarlos en una gráfica dinámica. Para la representación de estos valores almacenados se hace uso de la librería ngx-charts.

```
this.grafica = this.resultadosTraducidos.map((rt) => {  
  const valorGrafica: IGrafica = {  
    name: rt.tipo,  
    value: rt.pautas.length,  
  };  
  return valorGrafica;  
});
```

Figura 6.8: Modelado de datos para la representación gráfica.

6.3.2. Funcionamiento

El planteamiento de la estructura de la propuesta de solución ha sido de la manera antes mencionada, por ende se puede introducir una URL de cualquier sitio web del cual se quiera analizar la accesibilidad. El aplicativo al estar compuesto por una estructura modular va a permitir mostrar los datos del análisis, así como una gráfica en la que se determina el número de factores afectados de una manera más visual. La página inicial de este apartado se ve de la siguiente forma:

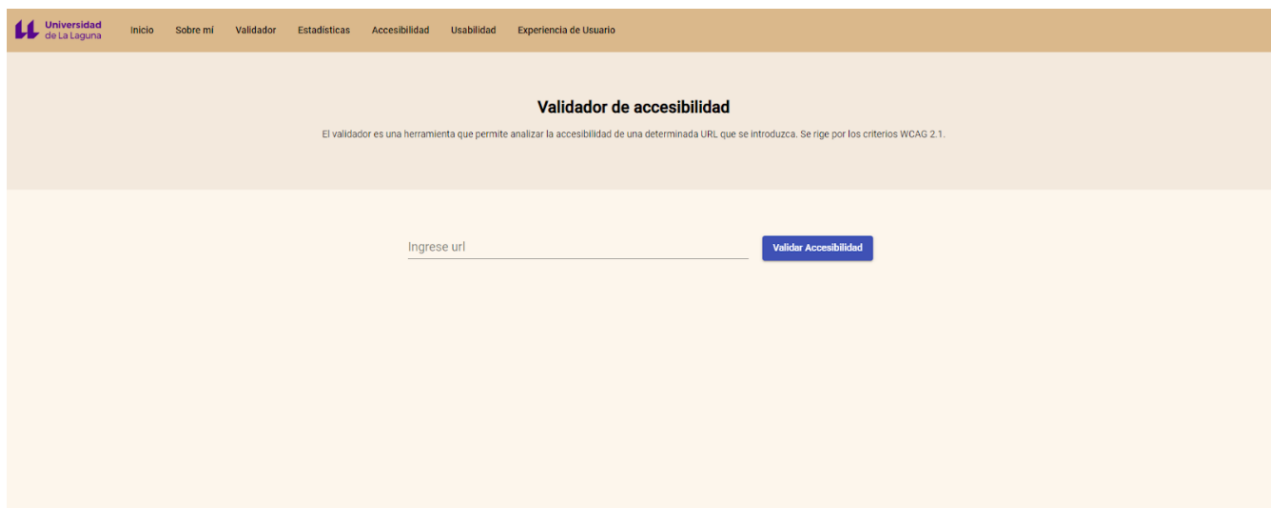


Figura 6.9: Página inicial del apartado validador.

Se introduce la URL que se quiera analizar y se espera a que la petición a la API devuelva los datos. Lo primero que se visualiza es la gráfica que se mencionaba anteriormente.

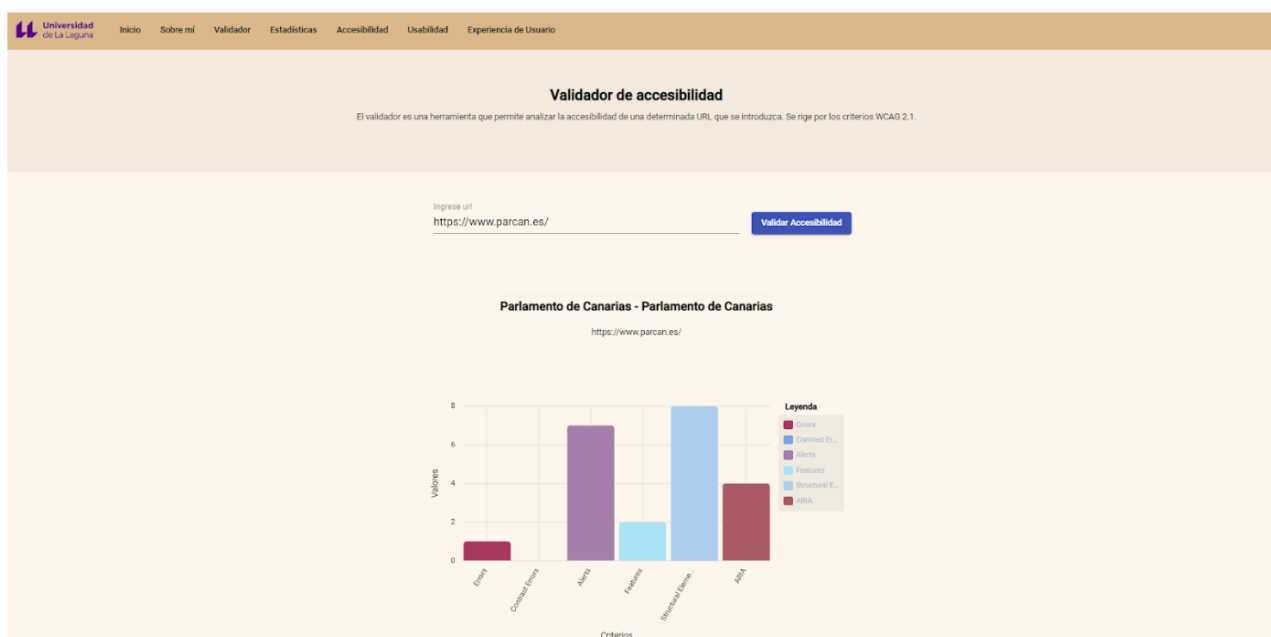


Figura 6.10: Analizar accesibilidad de un caso concreto.

A continuación de las gráficas, aparecen cada una de las categorías que contienen criterios que se ven afectados en la accesibilidad.



Figura 6.11: Criterios del análisis de accesibilidad.

Si se despliega alguna de estas pautas se observan los criterios que están asociados a cada una. Además, cada criterio contiene una breve descripción sobre el significado de cada uno, así como un botón para poder consultar la información oficial de W3C.



Figura 6.12: Contenido asociado a un criterio.

6.3.3. Estadísticas

Se incluye en el aplicativo el estudio realizado en el Capítulo 3, en concreto el Apartado 3.5. De esta forma, mediante gráficas dinámicas se muestran los datos obtenidos por el estudio realizado para cada caso de uso, en el cual se muestra una comparativa entre los dos niveles de adecuación analizados. Cabe recordar que las herramientas con las que se realizó el estudio fueron TAW y Achecker, por lo que los datos que se representan no son concluyentes ya que no todos los criterios están consensuados. Además, se hace imprescindible la revisión manual de un experto humano que compruebe estos resultados obtenidos.

En el aplicativo web se crea un componente principal llamado estadísticas y dentro de este hay tantos como casos de uso haya. De esta manera, dividimos por páginas los componentes del estudio previo realizado de accesibilidad.

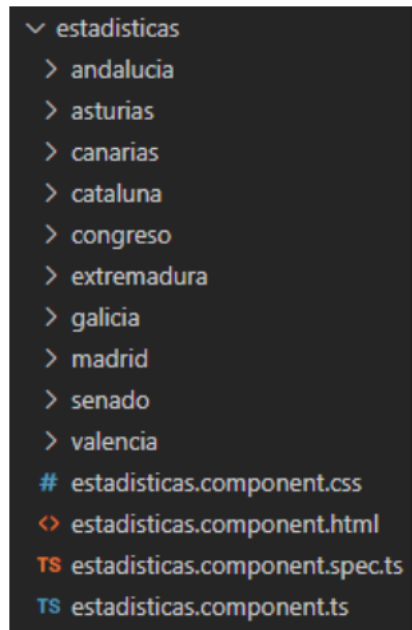


Figura 6.13: Estructura de componentes estadísticas.

Cada componente sigue la misma estructura, en la cual se hace referencia a cada una de las gráficas correspondiente a un caso de uso determinado del estudio realizado. La página principal de este apartado se vería del siguiente modo:



Figura 6.14: Página principal del apartado estadísticas.

Para visualizar el estudio de alguno de los casos de uso, únicamente se debe hacer clic sobre el que se quiera y se redirigirá a él. Al acceder se debe esperar a que los datos para las gráficas se carguen. Una vez cargados, aparece primero una gráfica comparativa de los criterios evaluados para los dos niveles de adecuación. Y posteriormente, se separan en gráficas de porcentajes cada uno de estos niveles para una mejor comprensión.



Figura 6.15: Gráfica del estudio de accesibilidad del Parlamento de Canarias.

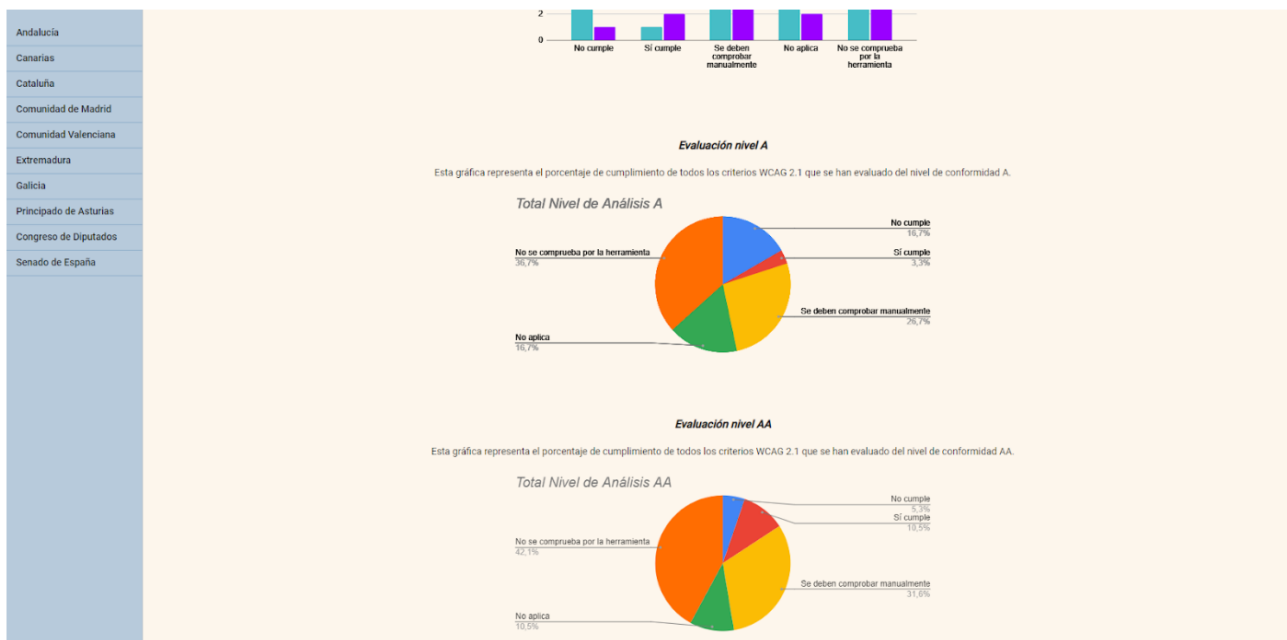


Figura 6.16: Gráfica de porcentajes y por nivel de adecuación del estudio de accesibilidad del Parlamento de Canarias.

6.4. Desarrollo de experiencia de usuario

En este apartado se explica la solución de desarrollo propuesta para el estudio de la experiencia de usuario de los casos de uso seleccionados en un principio.

6.4.1. Estructura del aplicativo

Como se menciona en el apartado 7.1, se hace uso de la API de PageSpeed Insight como motor de análisis de la experiencia de usuario. Una vez obtenida la clave API, se crea un servicio propio que se va a encargar de traer los datos a la aplicación.

Este servicio se encuentra en la clase “services”, que contiene otros servicios del aplicativo. Dentro de la clase, se crea un método al cual se le debe pasar una URL determinada y a través de una petición HTTP, en concreto el método de solicitud GET, se hace una llamada a la API. Esta llamada está compuesta por la concatenación de la propia URL de la API, seguido de la URL que se desee analizar y la clave API obtenida.

```
analizarUX(url: string) {
  try {
    return lastValueFrom(
      this.http
        .get<IUxResult>(
          `${this.URL_BASE}url=${url}&key=AIzaSyAVyAt1dQvCvHF1_T1AB1o9tS-297m4vec`
        )
        .pipe(take(1))
    );
  } catch (error) {
    console.error(error);
    return null;
  }
}
```

Figura 6.17: Método para la llamada a la API de UX.

Los datos devueltos por la API son en formato JSON. Por tanto, en el mismo servicio se crean las estructuras que van a almacenar dichos datos. Estas estructuras nos van a permitir posteriormente modelar la información que se requiera y para ello 4 interfaces diferentes como estructuras. La primera interfaz contiene la distribución principal del JSON, pero es en concreto los datos de la experiencia de origen los que se van a tratar. Por tanto, se crea una segunda interfaz que contenga la estructura de los resultados aportados en el objeto *origenLoadingExperience*. Concretamente de esta segunda interfaz, es el objeto *metrics* el cual va a contener las métricas que se desean estudiar, por lo que se crea una tercera interfaz que las contiene. La cuarta interfaz contiene la estructura de las métricas.

```

export interface IUxResult {
  analysisUTCTimestamp: string;
  captchaResult: string;
  id: string;
  kind: string;
  lighthouseResult: any[];
  loadingExperience: any[];
  originLoadingExperience: IOrigenResult;
}
export interface IOrigenResult {
  id: string;
  initial_url: string;
  metrics: IMetricas;
  overall_category: string;
}
export interface IMetricas {
  CUMULATIVE_LAYOUT_SHIFT_SCORE: IEstructuraUX;
  EXPERIMENTAL_INTERACTION_TO_NEXT_PAINT: IEstructuraUX;
  EXPERIMENTAL_TIME_TO_FIRST_BYTE: IEstructuraUX;
  FIRST_CONTENTFUL_PAINT_MS: IEstructuraUX;
  FIRST_INPUT_DELAY_MS: IEstructuraUX;
  LARGEST_CONTENTFUL_PAINT_MS: IEstructuraUX;
}
export interface IEstructuraUX {
  percentile: number;
  distributions: any[];
  category: string;
}

```

Figura 6.18: Estructura de datos UX.

Una vez se tenga la estructura de los datos, se comienza a modelar la información obtenida. Para ello se crea un componente principal llamado UX y dentro de este hay tantos como casos de uso haya. De esta manera, dividimos por páginas los componentes de análisis de experiencia de usuario.

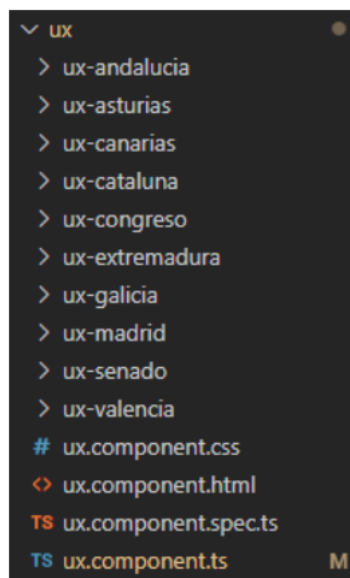


Figura 6.19: Estructura de datos UX.

Cada componente sigue la misma estructura. Angular tiene un método propio que se ejecuta después del constructor, este es el método *ngOnInit* y permite inicializar el componente una vez ha recibido las propiedades de entrada. Se decide utilizar este método para implementar la llamada al servicio y una vez devuelta la estructura de datos

modelar la información. En concreto, se muestran las métricas definidas en las Core Web Vitals, de las cuales se ha hablado en el Capítulo 6.

Cabe destacar que se hace uso de promesas [53], ya que se pueden representar resultados de una operación asíncronica como si fueran síncronas. En concreto se utiliza `async` para definir la función `ngOnInit` como asíncrona.

```
async ngOnInit(): Promise<void> {  
  
  this.resultadosUX = await this.uxService.analizarUX('https://www.parcen.es/');  
  if (!this.resultadosUX?.originLoadingExperience) return;  
  
  this.inicial_url = this.resultadosUX.originLoadingExperience.initial_url;  
  this.origenResult = this.resultadosUX.originLoadingExperience;  
  
  this.LCP = this.origenResult.metrics.LARGEST_CONTENTFUL_PAINT_MS;  
  this.FID = this.origenResult.metrics.FIRST_INPUT_DELAY_MS;  
  this.CLS = this.origenResult.metrics.CUMULATIVE_LAYOUT_SHIFT_SCORE;  
  
  this.asignarColores();  
}
```

Figura 6.20: Llamada al servicio y modelado de datos.

Además, se desarrolla un método el cual proporciona color y una categoría dependiendo del umbral en el que se encuentre la métrica evaluada.

```
case 'FAST':  
  this.LCPColor = 'green';  
  metrics.LARGEST_CONTENTFUL_PAINT_MS.category = 'RÁPIDO';  
  break;  
  
case 'AVERAGE':  
  this.LCPColor = 'orange';  
  metrics.LARGEST_CONTENTFUL_PAINT_MS.category = 'MEDIO';  
  break;  
  
case 'SLOW':  
  this.LCPColor = 'red';  
  metrics.LARGEST_CONTENTFUL_PAINT_MS.category = 'LENTO';  
  break;
```

Figura 6.21: Asignación de color y categoría por métrica.

6.4.2. Funcionamiento

El planteamiento de la estructura de la propuesta de solución ha sido de la manera antes mencionada ya que exclusivamente se quiere hacer el estudio sobre el caso de estudio propuesto. Por ende, si en un futuro se quisiera analizar cualquier sitio web, el aplicativo consta de una estructura modular la cual va a permitir introducir cualquier URL y mostrar los datos del análisis. La página inicial de este apartado se ve de la siguiente manera:



Figura 6.22: Página principal del apartado UX.

Para comprobar la experiencia de usuario sobre las Core Web Vitals de alguno de los casos de uso, únicamente se debe hacer clic sobre el que se quiera y se redirigirá a él. Al acceder los datos no se encuentran cargados, porque se debe esperar a que la petición a la API los devuelva. Una vez devueltos se representan indicando el tiempo de carga de cada métrica evaluada y el umbral en el que se encuentran. Destacar nuevamente, que estos datos están recogidos en los últimos 28 días.

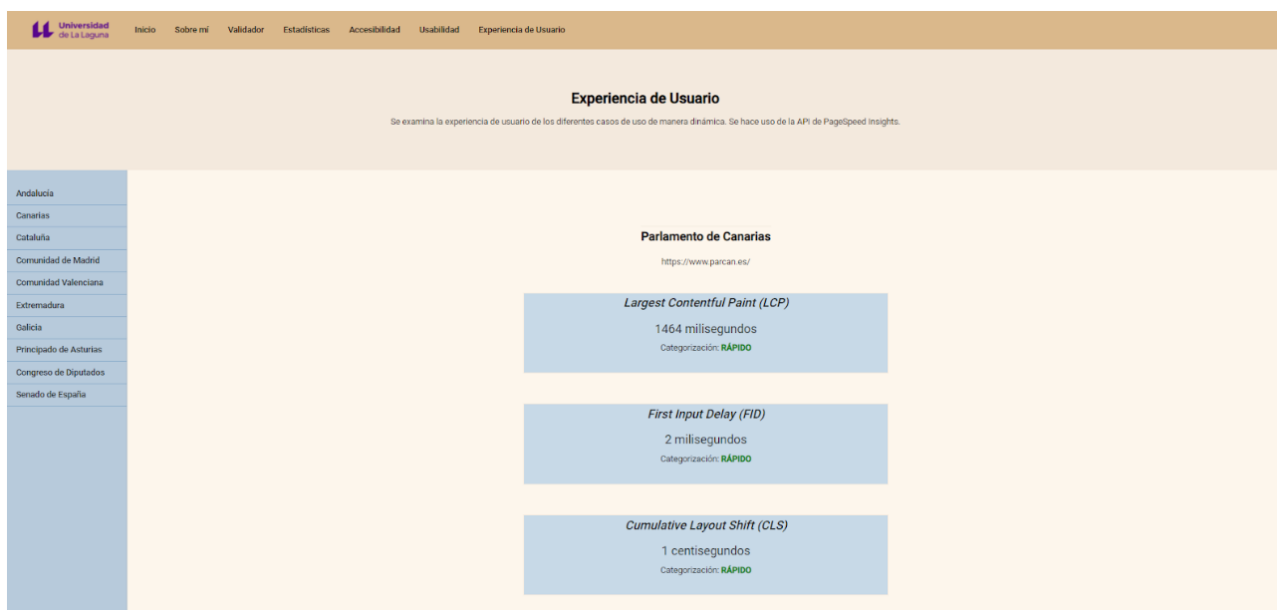


Figura 6.23: Análisis de UX del Parlamento de Canarias.

Capítulo 7

Conclusiones y Lineas Futuras

Tal y como se ha podido comprobar, el proyecto realiza una investigación y un análisis tanto de accesibilidad como de experiencia de usuario para un caso de estudio basado en un conjunto de casos seleccionados. Por una parte se comprueba cuanta fiabilidad y ayuda aportan al análisis de accesibilidad las herramientas automatizadas que son online y de código abierto. Y por otra parte, cómo de útil es la herramienta de Google Core Web Vitals.

Tras el estudio de accesibilidad, se corrobora que los resultados obtenidos son insatisfactorios de acuerdo a las métricas y al nivel de adecuación establecido, por lo que esto deriva a obtener datos no concluyentes. Además de que las herramientas que se han elegido como estudio no comprueban todos los criterios pautados por las WCAG 2.1 y el mayor inconveniente es que pueden darse falsos positivos o negativos. Esto ocurre porque dichas herramientas tienen limitaciones para verificar los criterios existentes, ya sea porque para poder acceder a un análisis más exhaustivo se debe pagar por una licencia o simplemente porque en el motor de análisis que tienen desarrollado no se implementa la comprobación de todos los criterios. Por lo que se hace imprescindible que un experto humano cerciore el estudio que se realiza con herramientas automáticas.

Por otro lado, la investigación realizada sobre UX resulta satisfactoria, ya que los umbrales que se establecen en las métricas Core Web Vitals ayudan a comprobar en qué rango se encuentra el tiempo de carga del despliegue del contenido, de la escalabilidad visual y de la interactividad. Existen diversas herramientas con las que realizar este análisis, pero principalmente se utiliza PageSpeed Insight que, además de ser una herramienta creada por Google, es bastante completa. Los resultados de las métricas Core Web Vitals se recogen de la experiencia de los usuarios reales vividos en los últimos 28 días, también se aporta un diagnóstico en tiempo real del rendimiento de la URL que se esté estudiando y diversas recomendaciones de mejora para el sitio web. Hay que destacar que la puntuación obtenida en el diagnóstico de rendimiento puede fluctuar debido a las condiciones en las que se esté realizando.

Respecto a la implementación del aplicativo web basado en un modelo de APIs, se ha hecho uso de dos, una para validar la accesibilidad de cualquier sitio web, y otra para comprobar de manera dinámica los umbrales obtenidos por las métricas Core Web Vitals en los últimos 28 días de los casos de uso seleccionados. Los datos que se obtienen han sido tratados y maquetados. En este aplicativo se ha incluido el estudio realizado de manera manual tanto de accesibilidad como de usabilidad. Para ello se hace uso de

herramientas software como: GitHub, Latex, Firebase, Angular y diversas librerías para la implementación.

Como posibles líneas de trabajo futuro, se podría realizar un desarrollo propio de una API, en la cual se implementara la comprobación de ciertos criterios que no se hayan podido comprobar con otras herramientas ya existentes. De esta manera, se introduciría un enlace de un sitio web que se quiera evaluar y se devolverá de forma automatizada los resultados de la valoración de las pautas correspondientes, de la misma forma que lo haría una herramienta de validación automática. Otra posible línea de trabajo futuro, podría ser que los resultados que se han obtenido en el análisis tanto de accesibilidad como de UX se pudieran guardar en una base de datos. De este modo si se quisiera consultar el análisis que se realizó dos meses atrás se tuviera un historial de los resultados, e incluso unas estadísticas para comprobar la evolución que ha ido teniendo.

En cuanto al diseño de la interfaz de usuario, se podrían realizar un conjunto de test de usuario a un gran número de personas distintas, con la intención de recibir retroalimentación para adaptar y mejorar el diseño visual y la facilidad de uso del aplicativo.

Capítulo 8

Conclusions and Future Lines of Work

As has been shown, the project carries out a study and analysis of both accessibility and user experience for a set of selected cases. In this way, on the one hand, it is verified how reliable and helpful the automated tools that are online and open source are for accessibility analysis, and on the other hand, how useful the Google Core Web Vitals tool is.

After the accessibility study, it is corroborated that the results obtained are unsatisfactory according to the metrics and the established level of suitability, which leads to inconclusive data. In addition to the fact that the tools chosen for the study do not check all the criteria set out in the WCAG 2.1, the main drawback is that false positives or negatives can occur. This happens because these tools have limitations to check the existing criteria, either because in order to have access to a more exhaustive analysis a licence fee must be paid, or simply because the analysis engine they have developed does not implement the checking of all the criteria. It is therefore essential that a human expert verifies the study that is carried out with automatic tools.

On the other hand, the research carried out on UX is satisfactory, as the thresholds established in the Core Web Vitals metrics help to check in what range is the loading time of content display, visual scalability and interactivity. There are various tools with which to perform this analysis, but PageSpeed Insight is used, which, in addition to being a tool created by Google, is quite comprehensive. The results of the Core Web Vitals metrics are gathered from the experience of real users over the last 28 days, and also provide a real-time diagnosis of the performance of the URL under study, as well as various recommendations for improvement of the website. It should be noted that the score obtained in the performance diagnosis may fluctuate due to the conditions in which it is being carried out.

Regarding the implementation of the web application based on an ecosystem of APIs, two have been used, one to validate the accessibility of any website, and another to dynamically check the thresholds obtained by the Core Web Vitals metrics in the last 28 days of the selected use cases. The data obtained have been processed and mapped. This application includes a manual study of both accessibility and usability. For this purpose, software tools such as GitHub, Latex, Firebase, Angular and various libraries were used for the implementation.

As possible lines of work, an API could be developed, in which the verification of certain criteria that could not be checked with other existing tools could be implemented. In this

way, a link to a website to be evaluated would be entered and the results of the evaluation of the corresponding guidelines would be returned automatically, in the same way as an automatic validation tool would do.

Another possible line of work could be that the results obtained in the analysis of both accessibility and UX could be saved in a database. In this way, if you wanted to consult the analysis that was carried out two months ago, you would have a history of the results, and even statistics to check the evolution of the results.

In terms of user interface design, a set of user tests could be carried out on a large number of different people, with the intention of receiving feedback to adapt and improve the visual design and user-friendliness of the application.

Capítulo 9

Presupuesto

En este capítulo se recoge el presupuesto estimado para el desarrollo del proyecto, el cual ha tenido una duración de 5 meses. En él, se recogen los gastos de recursos materiales y humanos. Estos son:

- En un principio el uso de la API de WAVE no tiene costes, ya que no se realizan peticiones de manera masiva para obtener diversos datos, sino que simplemente se hace una consulta normal. Esta consulta es básica, por lo que tampoco genera un coste adicional. En el caso de que se quisieran hacer muchas peticiones y el grado de análisis fuera más complejo, se debe pagar por los créditos que hacen la llamada a la API.
- La API de PSI no tiene costes por uso, exclusivamente se tiene que estar registrado en ella.
- El hosting del aplicativo web en Firebase Hosting proporciona almacenamiento, transferencia de datos, dominio personalizado y SSL. En este caso, se hace uso de esta herramienta para obtener el certificado SSL y un dominio personalizado, los cuales no conllevan costes.
- Un personal técnico para la realización del proyecto. Además del estudio y análisis que se realiza, el proyecto contiene el desarrollo de un aplicativo web, el cual también será tarea del individuo.

En la siguiente tabla se recogen los costes de los recursos necesarios correspondientes a la duración del desarrollo del proyecto.

Tipos	Cantidad	Coste Unidad	Coste Total
Personal técnico	5 meses	1568,15 €	7840,75€
Firebase	12 meses	0 €	0 €
API de WAVE	100 créditos *	0 €	0 €
API de PSI	12 meses	0 €	0 €
Total			7840,75€

Tabla 9.1: Presupuesto estimado del proyecto.

* Si se agotan los 100 créditos, habrá que comprar créditos adicionales que sí tienen un costo de 0,037 € el crédito.

Apéndice A

Acrónimos

- **TIC:** Tecnologías de la Información y la Comunicación.
- **UX:** experiencia de usuario.
- **API:** Interfaz de programación de aplicaciones.
- **UE:** Unión Europea.
- **RD:** Real Decreto.
- **OAW:** Observatorio de Accesibilidad Web.
- **W3C:** World Wide Web Consortium.
- **WCAG:** Directrices de Accesibilidad al Contenido Web
- **PMP:** Puntuación Media de Página.
- **PMSW:** Puntuación Media del Sitio Web.
- **PMV:** Puntuación Media de Verificación.
- **ISO:** International Organization for Standardization.
- **SEO:** Search Engine Optimization.
- **LCP:** Largest Contentful Paint.
- **FID:** First Input Delay.
- **CLS:** Cumulative Layout Shift.
- **PSI:** PageSpeed Insights.
- **AAPP:** Administraciones Públicas.
- **TFG:** Trabajo de Fin de Grado.

Bibliografía

- [1] Web vitals. <https://web.dev/i18n/es/vitals/>.
- [2] EUR-Lex 32016L2102 EN EUR-Lex. Eur-lex — access to european union law. <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX:32016L2102>.
- [3] BOE.es BOE-A-2018-12699 Real Decreto 1112/2018 de 7 de septiembre sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público. Boe.es - agencia estatal boletín oficial del estado. <https://www.boe.es/eli/es/rd/2018/09/07/1112>.
- [4] Observatorio de Accesibilidad. Pae. https://administracionelectronica.gob.es/pae_Home/pae_Estrategias/pae_Accesibilidad/pae_Observatorio_de_Accesibilidad.html.
- [5] EUR-Lex 32018D1524 EN EUR-Lex. Eur-lex — access to european union law. https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=uriserv:OJ.L_.2018.256.01.0108.01.SPA&toc=OJ:L:2018:256:FULLL.
- [6] World Wide Web Consortium (W3C). World wide web consortium (w3c). <https://www.w3.org/>.
- [7] Metodología para el seguimiento simplificado UNE-EN 301549:2019. Cpage. <https://cpage.mpr.gob.es/producto/metodologia-para-el-seguimiento-simplificado-une-en-3015492019/>.
- [8] Metodología para el seguimiento en profundidad para sitios web UNE-EN 301549:2019. Cpage. <https://cpage.mpr.gob.es/producto/metodologia-para-el-seguimiento-en-profundidad-para-sitios-web-une-en-3015492019/>.
- [9] W. W. A. I. Wcag 2.1 de un vistazo". web accessibility initiative (wai). <https://www.w3.org/WAI/standards-guidelines/wcag/glance/es>.
- [10] Servicio de diagnóstico en línea del Observatorio de Accesibilidad Web. Pae. https://administracionelectronica.gob.es/ctt/oaw#.Yo_NiXbMI2w.
- [11] Taw | web accessibility and w3c standardization services. <https://www.tawdis.net/>.
- [12] examiner. <http://examinator.net/>.
- [13] Web accessibility checker for wcag 2.0 aoda compliance monitoring. <https://achecker.achecks.ca/checker/index.php>.

- [14] Website Error Checker: Accessibility and Link Checker SortSite. Powermapper.com - website testing and site mapping tools. <https://www.powermapper.com/products/sortsite/>.
- [15] Wave web accessibility evaluation tool. <https://wave.webaim.org/>.
- [16] Web Developer. Google chrome - download the fast, secure browser from google. <https://chrome.google.com/webstore/detail/web-developer/bfbameneiokkghdmiekhjnmfkcldhnm?hl=es>.
- [17] Colour Contrast Analyser | Vision Australia. Blindness and low vision services. Vision australia. blindness and low vision services. <https://www.visionaustralia.org/services/digital-access/resources/colour-contrast-analyser>.
- [18] Contrast-A: Find Accessible Color Combinations. Das plankton | ui/ux design, branding and illustration. <http://dasplankton.de/ContrastA/>.
- [19] WCAG Color contrast checker. Google chrome - download the fast, secure browser from google. <https://chrome.google.com/webstore/detail/wcag-color-contrast-check/plnahcmalebfffmaghcpcmpaciebdhgdf?hl=es>.
- [20] HeadingsMap. Google chrome - download the fast, secure browser from google. <https://chrome.google.com/webstore/detail/headingsmap/flbjommegcjonpdmenkdiocclhjcmibi>.
- [21] Todo sobre la ISO/IEC 9126: 2001 y su importancia en las empresas | Verity. Verity. <https://www.verity.cl/blog/que-es-norma-iso-iec-9126-2001>.
- [22] ISO 9241:2010 estándar internacional para la industria digital. Blog ida chile | estrategia para el éxito de tu negocio. <https://blog.ida.cl/estrategia-digital/iso-92412010-estandar-internacional-industria-digital/>.
- [23] ¿Qué es el posicionamiento SEO y por qué es tan importante?. Garage. <https://es.godaddy.com/blog/que-es-posicionamiento-seo/>.
- [24] Web del parlamento de canarias. <https://www.parcn.es/>.
- [25] Web de asamblea de extremadura. <https://www.asambleaex.es/>.
- [26] Web corts valencianes. <https://www.cortsvalencianes.es/>.
- [27] Web del parlamento de andalucia. <http://www.parlamentodeandalucia.es/>.
- [28] Web asamblea de madrid. <https://www.asambleamadrid.es/>.
- [29] Web de parlament de catalunya. <https://www.parlament.cat/web/index.html>.
- [30] Web parlamento de galicia. <http://www.parlamento.gal/>.
- [31] Web del parlamento del principado de asturias. <https://www.jgpa.es/>.
- [32] Congreso de los diputados. <https://www.congreso.es/>.
- [33] Senado de españa. <https://www.senado.es/web/index.html>.

- [34] Evaluación Heurística (PARTE I) Interactius. Interactius. <https://interactius.com/evaluacion-heuristica-parte-i/>.
- [35] 10 Usability Heuristics for User Interface Design. Nielsen norman group. <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- [36] Largest Contentful Paint (LCP). web.dev. <https://web.dev/lcp/>.
- [37] First Input Delay (FID). web.dev. <https://web.dev/fid/>.
- [38] Cumulative Layout Shift (CLS). web.dev. <https://web.dev/cls/>.
- [39] Pagespeed insights. <https://pagespeed.web.dev/>.
- [40] About PageSpeed Insights | Google Developers. Google developers. <https://developers.google.com/speed/docs/insights/about>.
- [41] Visual Studio Code Code Editing. Redefined. Microsoft. <https://code.visualstudio.com/>.
- [42] GitHub: Where the world builds software. Github. <https://github.com/>.
- [43] npm. <https://www.npmjs.com/>.
- [44] Angular. <https://angular.io/>.
- [45] Angular material. <https://material.angular.io/>.
- [46] Introduction - ngx-charts. <https://swimlane.gitbook.io/ngx-charts/>.
- [47] Rxjs. <https://rxjs.dev/>.
- [48] Acerca de la API de PageSpeed Insights | Google Developers. (s. f. b). Google developers. <https://developers.google.com/speed/docs/insights/v5/about?hl=es>.
- [49] Firebase. <https://firebase.google.com/?hl=es>.
- [50] Figma: the collaborative interface design tool. Figma. <https://www.figma.com/>.
- [51] Color palettes for designers and artists - color hunt. <https://colorhunt.co/>.
- [52] Repositorio del proyecto Trabajo de Fin de Grado. Elena rijo garcía. <https://github.com/alu0101265421/proyecto-TFG>.
- [53] Usar promesas JavaScript | MDN. Mdn web docs. https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Using_promises.