

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN PROTOTIPO WEB DE UN PLANO INTERIOR ACCESIBLE PARA PERSONAS NO VIDENTES. CASO DE ESTUDIO: SEGUNDO PISO DE LA FACULTAD DE INGENIERÍA DE SISTEMAS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

CRISTINA GABRIELA RIVERA PASTRANO

cristina.rivera@epn.edu.ec

Director: Ing. Tania Elizabeth Calle Jiménez

tania.calle@epn.edu.ec

Quito, agosto 2016

DECLARACIÓN

Yo, Cristina Gabriela Rivera Pastrano declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Cristina Gabriela Rivera Pastrano

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Cristina Gabriela Rivera Pastrano, bajo mi supervisión.

Ing. Tania Calle

DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Quiero agradecer a mis padres, especialmente a mi madre por brindarme el apoyo necesario en mis momentos de dificultad. Por darme las fuerzas necesarias y alentarme a continuar con mis estudios y cumplir mis metas. Por darme un ejemplo de perseverancia, para seguir adelante a pesar de mis tropiezos.

Agradezco a mis hermanos, que fueron mi ejemplo para convertirme en profesional y supieron aclarar mis dudas con sus conocimientos en los momentos más duros de mi vida estudiantil.

A mis amigos, con quienes pasé los momentos más alegres de mi carrera. Los cuales no solo fueron compañeros de clase, sino de vida, cuyas experiencias me acompañarán para siempre.

A mi novio, que fue mi apoyo incondicional durante toda mi carrera universitaria. Ha sido mi mejor amigo y mi pilar, cuyas enseñanzas y consejos me han permitido convertirme en la persona que soy ahora.

A mi tutora Tania Calle y a la ingeniera Sandra Sánchez, que me brindaron su carisma y sus conocimientos profesionales y humanos, que me permitieron desarrollar mi proyecto de titulación y a la vez ser mejor profesional.

A mis maestros, que formaron parte de mi desarrollo profesional.

A todas las personas que alguna vez formaron parte de esta etapa de mi vida y que de una u otra manera estuvieron presentes para alcanzar este triunfo académico.

DEDICATORIA

*Dedico esta tesis a mi familia y amigos,
Si no hubiera sido por ustedes no lo hubiera logrado.*

CONTENIDO

PRESENTACIÓN.....	1
RESUMEN.....	2
CAPÍTULO 1. DEFINICIÓN DEL PROBLEMA Y SU ENTORNO DE SOLUCIÓN.....	3
1.1. ANÁLISIS DE ESPECIFICACIONES DE SITIOS WEB ACCESIBLES PARA PERSONAS NO VIDENTES.....	6
1.1.1. <i>Société de transport de Montréal</i>	7
1.1.2. <i>Prototipo accesible con Google Maps API V3</i>	11
1.1.3. <i>London Sound Survey</i>	13
1.1.4. <i>Secretaría técnica para la gestión Inclusiva en Discapacidades</i>	18
1.2. ANÁLISIS DE PAUTAS DE ACCESIBILIDAD WCAG 2.0	22
1.2.1. <i>WCAG 2.0</i>	22
1.3. JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO.....	23
1.3.1. <i>Metodología de desarrollo Scrum</i>	26
1.4. JUSTIFICACIÓN DE LAS HERRAMIENTAS DE DESARROLLO	29
1.4.1. <i>HTML 5</i>	29
1.4.2. <i>CSS3</i>	30
1.4.3. <i>JavaScript</i>	31
1.4.4. <i>SVG</i>	31
1.4.5. <i>Sails.js</i>	33
CAPÍTULO 2. ANALISIS, DISEÑO Y ELABORACIÓN DEL PROTOTIPO.....	34
2.1. ESPECIFICACIÓN DE REQUERIMIENTOS.....	34
2.1.1. <i>Historias de usuario y criterios de aceptación</i>	35
2.1.2. <i>Historias de usuario técnicas</i>	40
2.2. ANÁLISIS DEL PROTOTIPO.....	51
2.2.1. <i>Lista inicial del product backlog</i>	51
2.2.2. <i>Definición de Sprints</i>	52
2.3. DISEÑO DEL PROTOTIPO	62
2.3.1. <i>Diseño del Primer Sprint</i>	62
2.4. CONSTRUCCIÓN DEL PROTOTIPO.	80
2.4.1. <i>Diseño del Segundo Sprint</i>	80
2.4.2. <i>Diseño del Tercer Sprint</i>	86

CAPÍTULO 3. PRUEBAS DEL PROTOTIPO EN EL CASO DE ESTUDIO	87
3.1. PRUEBAS DEL PROTOTIPO CON HERRAMIENTAS VALIDADORAS.	87
3.1.1. <i>Examinator</i>	87
3.1.2. <i>TAW</i>	89
3.2. PRUEBAS DEL PROTOTIPO CON HERRAMIENTAS SIMULADORAS.....	92
3.2.1. <i>No coffee</i>	92
3.2.2. <i>Impairment Simulator</i>	94
3.2.3. <i>Spectrum</i>	97
3.3. PRUEBAS DEL PROTOTIPO CON PERSONAS NO VIDENTES.	100
3.3.1. <i>Datos del primer usuario</i>	100
3.3.2. <i>Datos de la primera evaluación.</i>	100
3.3.3. <i>Análisis de resultados de la primera evaluación</i>	102
3.3.4. <i>Datos del segundo usuario</i>	103
3.3.5. <i>Datos de la segunda evaluación.</i>	103
3.3.6. <i>Análisis de resultados de la segunda evaluación</i>	105
CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES	107
4.1. CONCLUSIONES.....	107
4.2. RECOMENDACIONES.....	109
BIBLIOGRAFÍA.....	111
GLOSARIO	115
ANEXOS.....	117

LISTA DE TABLAS

Tabla 1.1: Resumen de pautas de accesibilidad WCAG2.0.	6
Tabla 1.2: Resumen de análisis de accesibilidad en sitios web con mapas accesibles ...	22
Tabla 1.3: Tabla comparativa entre las metodologías ágiles XP, Scrum e Iconix.	25
Tabla 1.4: Descripción de las reuniones de Scrum.	28
Tabla 1.5: Descripción de los roles de Scrum.	28
Tabla 1.6: Tabla comparativa entre XHTML, HTML y HTML5.	30
Tabla 1.7: Tabla comparativa entre Canvas y SVG.	32
Tabla 2.1: Roles de Scrum.	34
Tabla 2.2: Elementos y descripciones para historias de usuario y criterios de aceptación.	36
Tabla 2.3: Historias de usuario y criterios de aceptación [28].	37
Tabla 2.4: Historia de usuario - Diseñar arquitectura del prototipo.HU01	40
Tabla 2.5: Historia de usuario - Diseñar e implementar plano interior accesible.HU02. ...	41
Tabla 2.6: Historia de usuario - Diseñar e implementar interfaz principal del sitio web.HU03	42
Tabla 2.7: Historia de usuario - Diseñar e implementar menú de accesibilidad.HU04.	43
Tabla 2.8: Historia de usuario - Implementar modo simulación.HU05.	44
Tabla 2.9: Historia de usuario - Implementar navegabilidad en el sitio web.HU06.	45
Tabla 2.10: Historia de usuario - Implementar navegabilidad en el plano interior.HU07.	46
Tabla 2.11: Historia de usuario - Diseñar Menú de ayuda.HU08.	47
Tabla 2.12: Historia de usuario - Diseñar y ejecutar pruebas con validadores.HU09.	48
Tabla 2.13: Historia de usuario - Diseñar y ejecutar pruebas con simuladores.HU10.	49
Tabla 2.14: Historia de usuario - Diseñar pruebas con usuarios.HU11.	50
Tabla 2.15: Product Backlog Inicial.	52
Tabla 2.16: Tareas por historias de usuario del Primer sprint.	54
Tabla 2.17: Tareas por historias de usuario del Segundo sprint.	57
Tabla 2.18: Tareas por historias de usuario del Tercer sprint.	60
Tabla 3.1: Pruebas de accesibilidad realizadas con Usuario 1.	101
Tabla 3.2: Pruebas de accesibilidad realizadas con Usuario 2.	104

LISTA DE ILUSTRACIONES

Ilustración 1.1: Página web del Metro de Montreal	7
Ilustración 1.2: Menú principal accesible por teclado.	7
Ilustración 1.3: Mapa de metro de Montreal y descripción de las paradas.	8
Ilustración 1.4: Despliegue de la página web en un dispositivo móvil.	8
Ilustración 1.5: Conformidad de accesibilidad de la página web.	9
Ilustración 1.6: Tecnologías de verificación de conformidad de accesibilidad de la página web.	9
Ilustración 1.7: Atajos de teclado para cambiar el tamaño de fuente.	10
Ilustración 1.8: Atajos de teclado para manejo de fechas y formularios.	10
Ilustración 1.9: Página principal del prototipo.	11
Ilustración 1.10: Menú de rutas.	12
Ilustración 1.11: Sección de servicios de rutas del prototipo.	12
Ilustración 1.12: Nivel de conformación de accesibilidad de la página web.	13
Ilustración 1.13: Vista principal de la página "London Sound Survey".	13
Ilustración 1.14: Ejemplo de mapa sonoro de la página "London Sound Survey".	14
Ilustración 1.15: Despliegue de la página "London Sound Survey" en un dispositivo móvil.	15
Ilustración 1.16: Vista de la página "London Sound Survey" en blanco y negro.	15
Ilustración 1.17: Mapa "London Sound Survey" accesible por teclado.	16
Ilustración 1.18: Menú de navegación de la página "London Sound Survey".	16
Ilustración 1.19: Mapa por regiones de la página "London Sound Survey".	17
Ilustración 1.20: Vista textual de la página "London Sound Survey".	17
Ilustración 1.21: Vista principal de la página web de SETEDIS.	18
Ilustración 1.22: Opción de lenguaje de señas para presentación de información.	19
Ilustración 1.23: Lector de pantalla propio de la página web de SETEDIS.	19
Ilustración 1.24: Visualización de la página web de SETEDIS en un dispositivo móvil. ...	20
Ilustración 1.25: Menú de accesibilidad de la página web de SETEDIS.	20
Ilustración 1.26: Opción de Zoom de la página web de SETEDIS.	21
Ilustración 1.27: Product Backlog y Sprint Backlog.	27
Ilustración 2.1: Product Release Plan.	53
Ilustración 2.2: Product Backlog.	53
Ilustración 2.3: Sprint Backlog Template para el primer sprint.	56
Ilustración 2.4: Gráfico de esfuerzo vs Número de Horas para el primer sprint.	56
Ilustración 2.5: Sprint backlog template para el segundo sprint.	59

Ilustración 2.6: Gráfico de esfuerzo vs Número de horas para el Segundo sprint.	59
Ilustración 2.7: Sprint backlog template para Tercer sprint.	61
Ilustración 2.8: Gráfico de esfuerzo vs Número de horas para el Tercer sprint.	62
Ilustración 2.9: Modelo de arquitectura usada en el prototipo.....	64
Ilustración 2.10: Diagrama conceptual de datos.....	64
Ilustración 2.11: Diagrama físico de datos.....	65
Ilustración 2.12: Captura de Vista Layout.	65
Ilustración 2.13: Captura de Vista informacionMapa.....	66
Ilustración 2.14: Captura de Vista mapaPlano.	66
Ilustración 2.15: Captura de Vista mapaPlanoColor.....	67
Ilustración 2.16: Captura de Vista mapaPlanoNegro.	67
Ilustración 2.17: Captura de vista menuPlano.	68
Ilustración 2.18: Captura de vista Plano.	68
Ilustración 2.19: Modelo implementado en el framework Sails.	69
Ilustración 2.20: Script de coordenadas en formato JSON.	70
Ilustración 2.21: Representación de plano a escala del segundo piso de la facultad de sistemas.....	71
Ilustración 2.22: Atributos WAI-ARIA en código SVG del plano interior accesible.	72
Ilustración 2.23: Estilos en plano modelado en SVG.	73
Ilustración 2.24: Demostración de svg.pan-zoom.js añadido al plano.....	73
Ilustración 2.25: Esquema principal del prototipo con vistas parciales.	74
Ilustración 2.26: Diseño de interfaz principal del prototipo web.	75
Ilustración 2.27: Menú collapse para menú Oficinas de Docentes.	76
Ilustración 2.28: Vista principal de Menú de Accesibilidad.....	76
Ilustración 2.29: Vista principal de opciones de estilos para el prototipo.....	77
Ilustración 2.30: Vista de función tamaño de letra en el prototipo.	77
Ilustración 2.31: Vista de función interespaciado en el prototipo.	78
Ilustración 2.32: Vista de función tipo de letra en el prototipo.....	78
Ilustración 2.33: Opciones avanzadas en menú accesibilidad.....	79
Ilustración 2.34: Patrones aplicados en plano accesible.....	79
Ilustración 2.35: Opciones de filtros aplicados sobre el prototipo web.	80
Ilustración 2.36: Archivo JSON de rutas preestablecidas mediante coordenadas.	81
Ilustración 2.37: Función para graficar las rutas del ORM en formato SVG.	82
Ilustración 2.38: Función de reproducción de audio embebido en las rutas del modo simulación.....	82
Ilustración 2.39: Resultado de la ejecución del modo simulación.....	83

Ilustración 2.40: Orden secuencial de navegación de los elementos del prototipo.	83
Ilustración 2.41: Función para navegación con cursores en el elemento div del menú accesibilidad.	84
Ilustración 2.42: Funciones para atajos por teclado.	84
Ilustración 2.43: Esquema de navegación secuencial del plano interior.	85
Ilustración 2.44: Función para navegación con cursores aplicada al plano interior.	85
Ilustración 2.45: Descripción en menú de ayuda.	86
Ilustración 2.46: Indicaciones en menú de ayuda.	86
Ilustración 3.1: Examinador con el prototipo accesible.	88
Ilustración 3.2: Captura de las pruebas realizadas sobre el prototipo con el puntaje correspondiente.	88
Ilustración 3.3: Pruebas simuladas por personas.	89
Ilustración 3.4: Pruebas de accesibilidad realizadas con TAW.	90
Ilustración 3.5: Principio Perceptible evaluado por TAW.	90
Ilustración 3.6: Principio Operable evaluado por TAW.	91
Ilustración 3.7: Principio Comprensible evaluado por TAW.	91
Ilustración 3.8: Principio robusto evaluado por TAW.	91
Ilustración 3.9: Simulación de visión borrosa de No Coffee.	93
Ilustración 3.10: Filtro de efecto fantasma de No Coffee.	93
Ilustración 3.11: Filtro de degeneración macular de No Coffee.	93
Ilustración 3.12: Filtro de retinitis pigmentosa de No Coffee.	94
Ilustración 3.13: Filtro de retinopatía diabética de No Coffee.	94
Ilustración 3.14: Captura de la pantalla principal de Impairment Simulator.	95
Ilustración 3.15: Filtro de Glaucoma de Impairment Simulator.	95
Ilustración 3.16: Filtro de Retinopatía diabética de Impairment Simulator.	96
Ilustración 3.17: Filtro de baja agudeza visual de Impairment Simulator.	96
Ilustración 3.18: Filtro de Cataratas de Impairment Simulator.	97
Ilustración 3.19: Filtro de Protanopia de Spectrum.	98
Ilustración 3.20: Filtro de Tritanopia de Spectrum.	98
Ilustración 3.21: Filtro de acromatopsia de Spectrum.	99
Ilustración 3.22: Filtro de Deuteranopia de Spectrum.	99
Ilustración 3.23: Evidencia de usuario realizando las pruebas sobre el prototipo.	103
Ilustración 3.24: Evidencia de usuario realizando las pruebas sobre el prototipo.	106

PRESENTACIÓN

La accesibilidad web surge ante la necesidad de crear contenidos web, que puedan ser utilizados en condiciones equitativas por cualquier usuario. De tal manera, que las discapacidades no sean una limitante al momento de interactuar con una interfaz.

La importancia de la accesibilidad web radica en asegurar que los contenidos web estén al alcance de las personas sin importar su condición económica, social, geográfica o de salud.

En base a esto, surge la World Wide Web Consortium (W3C) [1], que es la entidad internacional instaurada para fomentar el uso de estándares para el desarrollo de la World Wide Web.

En Ecuador se ha implementado la norma denominada "Tecnología de la información - Directrices de accesibilidad para el contenido web del W3C - Web Content Accessibility 2.0 (WCAG 2.0) (ISO/IEC 40500:2012, IDT)" [2]. Por lo cual, en el país se están tomando en cuenta las directrices de accesibilidad para elaborar sitios web accesibles.

Uno de los temas que representa mayor dificultad para las personas no videntes, es la representación de planos y mapas geográficos en sitios web. Esto se debe a que la mayor parte de información se representa visualmente y no incluye funcionalidades que permitan navegar utilizando tecnologías de asistencia y lectores de pantalla. Esto representa un problema para las personas no videntes al tratar de ubicar elementos dentro de un mapa visual.

Para solucionar el problema escogido, se plantea el desarrollo de un prototipo web de un plano interior accesible para personas no videntes. Lo que busca es solucionar el problema de la ubicación de personas no videntes en espacios desconocidos mediante la creación de un sitio web elaborado en base a pautas de accesibilidad para permitir su operación sin requerir del uso de la visión.

RESUMEN

El presente proyecto de titulación tiene como objetivo, el desarrollo de un prototipo web de un plano interior accesible para personas no videntes. Cuyo caso de estudio es el segundo piso de la Facultad de Ingeniería de Sistemas. Para ello, se elabora un análisis de accesibilidad en sitios web que contienen mapas, para comprender cómo las pautas de accesibilidad WCAG 2.0 se aplican en el desarrollo de sitios web que contienen este tipo de información visual.

A continuación, se elabora una descripción de las pautas de accesibilidad WCAG 2.0 en las que se centró el desarrollo del prototipo. Además, se realiza un análisis de las herramientas que se usaron para construir el prototipo, con su respectiva justificación

El desarrollo del prototipo se guía en la metodología SCRUM, que se utiliza para construir un prototipo accesible mediante herramientas de desarrollo centradas en los principios de accesibilidad. El desarrollo del prototipo se subdivide en tres iteraciones principales, cada una de ellas integrada por tareas que se han estimado en base al riesgo y la prioridad en el prototipo.

Posteriormente, se evalúa el grado de accesibilidad del prototipo mediante pruebas con validadores para verificar el grado de cumplimiento de las pautas de accesibilidad, simuladores para observar como ciertas discapacidades visuales afectan el despliegue del prototipo y con personas no videntes para entender como el prototipo e se muestra en un entorno real.

En el capítulo final se presentan las conclusiones y recomendaciones, en donde se demuestra que se cumplieron los objetivos iniciales, logrando alcanzar un nivel de conformidad AA de acuerdo a las pautas WCAG 2.0.

CAPÍTULO 1. DEFINICIÓN DEL PROBLEMA Y SU ENTORNO DE SOLUCIÓN.

Se estima que alrededor de 1 billón de personas posee alguna discapacidad, esto es equivalente al 15% de la población mundial según The World Bank [3]; De este porcentaje, 285 millones de personas poseen algún tipo de discapacidad visual, de los cuales 39 millones son no videntes y 246 poseen baja visión según la Organización Mundial de la Salud [4]. Estas personas se encuentran excluidos de procesos educativos, salud, empleo, sistemas sociales y jurídicos.

Por ésta razón nace la Reunión de Alto Nivel de la Asamblea General sobre la Discapacidad y el Desarrollo (HLDM), que busca promover a representantes y organizaciones gubernamentales y no gubernamentales, organizaciones civiles y de discapacidad a unirse al cumplimiento de procesos de desarrollo de personas discapacitadas para lograr una participación equitativa en la sociedad [5].

En Ecuador se han establecido entidades para promover la inclusión de personas con discapacidad. Según el principio de Buen Vivir del Ministerio de Inclusión Económica y Social (MIES) que hace referencia a los artículos 35 a 55, se enfoca en garantizar los derechos tanto en formulación, definición, cumplimiento y ejecución de políticas tanto para la sociedad, familia y condiciones de discapacidad. Esto involucra también a buscar la eliminación de limitantes que influyan en ejercer derechos, involucración de aspectos tecnológicos y atención a personas con derechos vulnerados [6].

La ceguera es una de las deficiencias visuales que más limitantes involucra en varios ámbitos. Un estudio realizado en el 2013 indica que alrededor de 100 mil personas poseen ceguera en Ecuador, esto representa el 0.8% de la población total del país. Esto dificulta el libre desarrollo en áreas cognitivas, comunicativas, sensopercepción, así como aspectos tecnológicos, de comunicación e información [7].

Por ésta razón, se ha buscado impulsar el desarrollo de contenidos web en el Ecuador, con el objetivo de ser utilizados por personas con discapacidad con el fin de incluirse en las áreas de la sociedad como el acceso a la tecnología.

En base a esto, el 28 de enero de 2014, se publicó en el Registro Oficial n° 171, la aprobación de la norma denominada NTE INEN-ISO/IEC 40500 "Tecnología de la información - Directrices de accesibilidad para el contenido web del W3C (WCAG) 2.0 (ISO/IEC 40500:2012, IDT)" [2].

Esta norma busca impulsar la implementación de accesibilidad web en Ecuador para minimizar las barreras tecnológicas de acceso, a personas con discapacidades visuales.

La accesibilidad web se refiere al uso universal de la web para personas con discapacidad, es decir que puedan percibir, navegar, entender e interactuar con elementos web, además de integrarse a la contribución de la misma. Se enfoca en que todas las personas sean capaces de interactuar con la web, independientemente de los dispositivos usados, región, capacidades y recursos. La importancia de la accesibilidad web radica en permitir una inclusión de las personas discapacitadas en ámbitos como educación, trabajo, comercio, salud, entretenimiento, entre otras. Además, se sustenta en un conjunto de estándares formulados por la W3C – Web Access Initiative (WAI).

Estos estándares permiten a personas con discapacidades visuales a acceder a sitios web, que por lo general tienen dificultades en hallar la información y elementos que se transmiten por el canal visual. A través de estos lineamientos de accesibilidad se busca desarrollar sitios web accesibles, con implementaciones que faciliten el acceso y retroalimentación a personas no videntes.

Otro aspecto dentro de los sitios web son los mapas y planos interiores, en donde la mayor parte de información se transmite por el canal visual. Esto es un problema para las personas no videntes que requieren de estos medios, para navegar y ubicarse en espacios físicos desconocidos. Si dichos medios no poseen las alternativas adecuadas, por lo cual la información visual no puede ser expresada a totalidad para una persona no vidente, y esta información no puede ser interpretada.

Por éste motivo, el siguiente proyecto busca mitigar la barrera de accesibilidad web referente a planos interiores para personas no videntes. Las principales barreras en planos interiores, radican en la falta de implementaciones para navegación entre los elementos del plano y la falta de alternativas textuales que le indiquen al usuario mediante un lector de pantalla, el tipo de elemento y la información que este alberga.

Con este proyecto se busca mejorar la percepción espacial que conforma un caso de estudio práctico el cual incluye el segundo piso de la Facultad de Ingeniería en Sistemas de la Escuela Politécnica Nacional.

Con este prototipo se busca reducir los problemas con los que se enfrentan las personas no videntes al interactuar en espacios desconocidos, al ofrecer una herramienta que proporcione una retroalimentación anticipada de un lugar en donde el usuario se va a dirigir.

1.1. ANÁLISIS DE ESPECIFICACIONES DE SITIOS WEB ACCESIBLES PARA PERSONAS NO VIDENTES.

Las pautas de accesibilidad proporcionan guías de diseño web con puntos de verificación para reducir los problemas de acceso a la información. Ofrecen la flexibilidad necesaria para proporcionar accesibilidad a la información en distintos escenarios con el fin de crear páginas legibles y útiles [1].

Para analizar el grado de accesibilidad de los ejemplos de sitios web, la Tabla 1.1 ha sido elaborada en base a las pautas de accesibilidad WCAG 2.0. Contiene un resumen de las pautas que se han creado para cada principio de accesibilidad según la WCAG 2.0 [8].

Tabla 1.1: Resumen de pautas de accesibilidad WCAG2.0.

	Título	Pautas
1	Perceptible: La información y los componentes de la interfaz de usuario deben ser presentados de modo puedan ser percibidos.	1.1 Alternativas textuales. 1.2: Medios tempodependientes. 1.3 Adaptable. 1.4 Distinguible.
2	Operable: Los componentes de la interfaz de usuario y la navegación deben ser operables.	2.1 Accesible por teclado. 2.2 Tiempo suficiente. 2.3 Convulsiones. 2.4 Navegable.
3	Comprensible: La información y el manejo de la interfaz de usuario deben ser comprensibles.	3.1 Legible. 3.2 Predecible. 3.3 Entrada de datos asistida
4	Robusto: El sitio web debe poder ser interpretado de manera confiable por varias aplicaciones de usuario en donde se incluyen las tecnologías de asistencia.	4.1 Compatible

1.1.1. SOCIÉTÉ DE TRANSPORT DE MONTRÉAL

El sitio web pertenece a la agencia de transporte público del metro de Montreal. La página detalla el mapa principal de las rutas del metro con la descripción de cada una de sus paradas y sus horarios de apertura. Además, incluye un menú principal para acceder a otros servicios que proporciona el sitio web [9].

La Ilustración 1.1 representa la vista principal del sitio web “Société de transport de Montréal”.



Ilustración 1.1: Página web del Metro de Montreal

1.1.1.1. Características de accesibilidad

- El menú principal es accesible por teclado con la tecla TAB, en donde las opciones de menú se abren automáticamente para ver sus opciones cuando estas reciben el foco con la tecla TAB.

La Ilustración 1.2 presenta el menu principal del sitio web “Société de transport de Montréal” y su contenido de manera general.

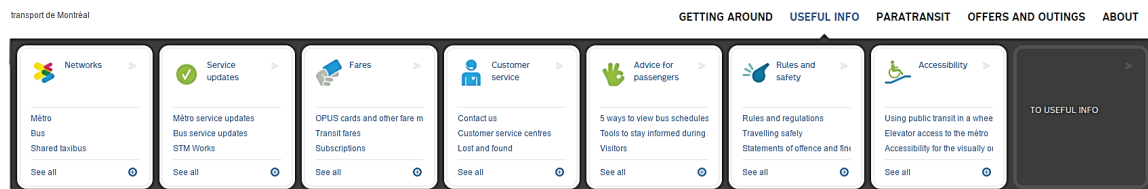


Ilustración 1.2: Menú principal accesible por teclado.

- Las paradas y elementos del mapa contienen descripciones textuales y elementos “title”. Además, cada elemento del mapa contiene información sobre el horario, descripción, entradas y otros servicios que proporcionan las paradas. La Ilustración 1.3 presenta una captura del metro de Montreal con la descripción de sus paradas.

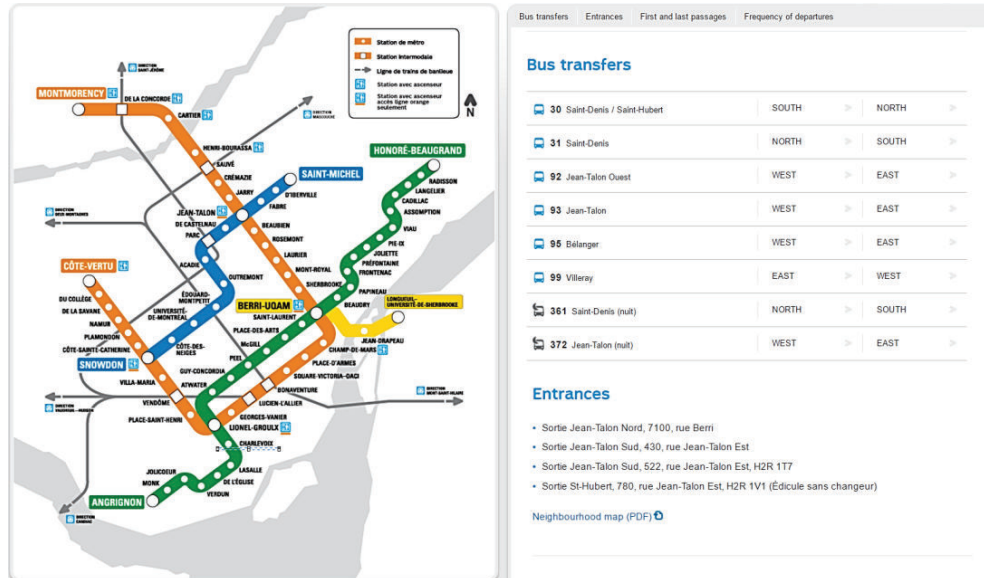


Ilustración 1.3: Mapa de metro de Montreal y descripción de las paradas.

- La página se adapta a la resolución del dispositivo en donde se despliega, en caso de un Smartphone, el contenido se redistribuye de tal manera que los elementos no pierden información como se observa en la Ilustración 1.4.

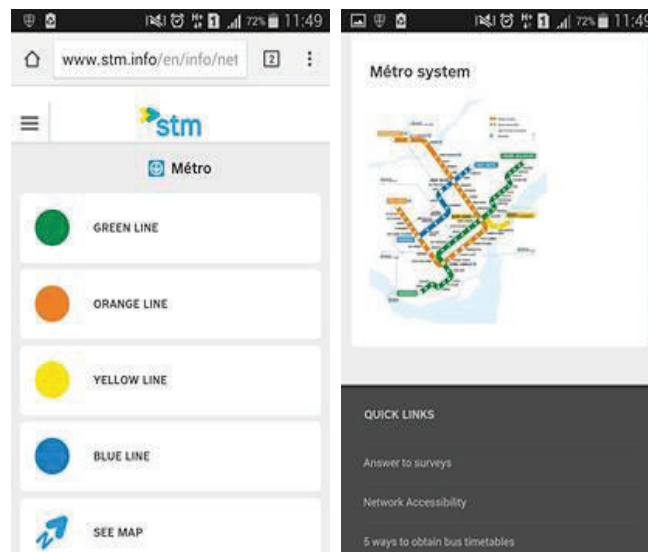


Ilustración 1.4: Despliegue de la página web en un dispositivo móvil.

- La página web posee un nivel AA de conformidad según el estándar WCAG2.0. La Ilustración 1.5 presenta la sección de accesibilidad del sitio web.

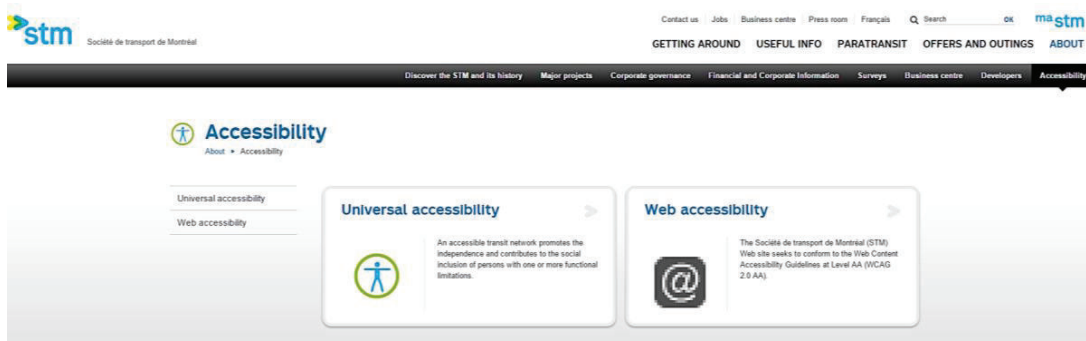


Ilustración 1.5: Conformidad de accesibilidad de la página web.

- Posee una especificación respecto a las tecnologías usadas para verificar el nivel de accesibilidad de la página. Las herramientas se centran en sistemas operativos, navegadores y lectores de pantalla. La ilustración 1.6 presenta las tecnologías de adaptación utilizadas para verificar el grado de conformancia.

Web accessibility

Web Accessibility

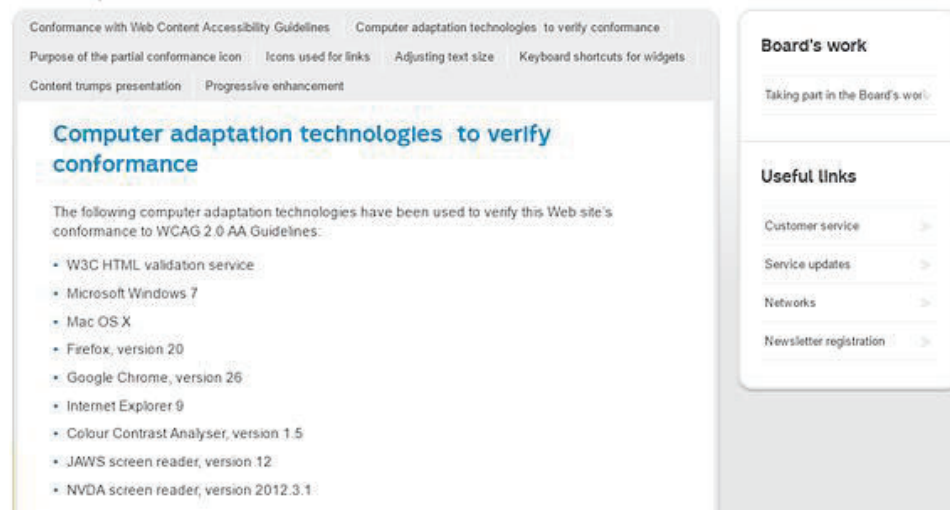


Ilustración 1.6: Tecnologías de verificación de conformidad de accesibilidad de la página web.

- Posee opciones para cambio de tamaño de fuente mediante atajos por teclado. La ilustración 1.7 muestra las opciones de atajos de teclado que se pueden usar para Microsoft Windows, Mac OS X y GNU/Linux.

Adjusting text size

On this Web site, text size can be adjusted via your user agent options [☞](#). The procedure varies depending on the user agent you use to browse the Web. Refer to the following tables for the proper procedure for the computer workstation you are using.

The following table shows how to increase character size using the keyboard shortcuts of your user agent:

Increase text size via your user agent's keyboard shortcuts.

Microsoft Windows	Mac OS X	GNU/Linux
Ctrl + +	Command + +	Ctrl + +

The following table indicates how to decrease character size using the keyboard shortcuts of your user agent:

Decrease text size using your user agent's keyboard shortcuts.

Microsoft Windows	Mac OS X	GNU/Linux
Ctrl + -	Command + -	Ctrl + -

Ilustración 1.7: Atajos de teclado para cambiar el tamaño de fuente.

- La página web contiene atajos por teclado para completar campos de entrada de texto, como son horarios, secciones de mapas y fechas. Las fechas se pueden escoger mediante un calendario que puede ser operado mediante teclado. La Ilustración 1.8 representa las ayudas de teclado para navegar por el sitio web, autocompletar campos y escoger otras funcionalidades como fechas.

Keyboard shortcuts for widgets

The Getting around page contains many interactions involving forms. Some interactions using the keyboard are unique and call for specific shortcuts.

Tabs

When the focus is on a tab, the Up and Down keys are used to move between tabs. Tabs are used in three cases:

- 1• To select the type of trip search the user wishes to do: Your Trip, Schedules, Maps, Fares
- 2• To switch from Bus view to Métro view in the Schedules and Maps sections
- 3• To choose a fare type in the Fares section

Autocomplete fields

When searching for an address, line number, or bus stop number, a list of choices will be displayed as soon as a few characters are typed in the search field. Use the Up and Down keys to move between choices.

Choosing the date

For fields requiring a date, a drop-down calendar opens when the focus is on them. Dates can be selected with the help of a number of keyboard shortcuts:

- **PAGE UP** : go to previous month
- **PAGE DOWN** : go to next month
- **CTRL** + **PAGE UP** : go to previous year
- **CTRL** + **PAGE DOWN** : go to next year
- **CTRL** + **HOME** : go to current month; open the calendar if it is closed
- **CTRL** + **GAUCHE** : go to previous day
- **CTRL** + **DROITE** : go to next day
- **CTRL** + **HAUT** : go to previous week
- **CTRL** + **BAS** : go to next week
- **ENTER** : confirm the selected date
- **CTRL** + **END** : close the calendar and erase the date
- **ESCAPE** : close the calendar and cancel the date change

Ilustración 1.8: Atajos de teclado para manejo de fechas y formularios.

1.1.2. PROTOTIPO ACCESIBLE CON GOOGLE MAPS API V3

La página corresponde a un prototipo desarrollado con la API de google MAPS. Consta de 5 secciones en donde se muestra un buscador de rutas y lugares, el mapa digital y los detalles de rutas y ubicaciones. El prototipo tiene la función de realizar búsquedas de lugares que aparecerán en el mapa, además permite establecer rutas con los pasos a seguir para llegar a un destino seleccionado [10]. La Ilustración 1.9 muestra la vista principal del prototipo accesible elaborado con Google Maps API V3.

Prototipo accesible con Google Maps API V3

Busca en Google Places

Buscador

Busca una dirección, lugar o establecimiento

BUSCAR

Puedes buscar tipos de establecimientos o el nombre exacto de un comercio.
[ve a los resultados](#)

Servicio de rutas

Buscador de direcciones y puntos de interés

Origen

Usar tu localización

Pulsando sobre este botón la aplicación utilizará tu posición actual como punto de origen para el servicio de rutas. También puedes entrarla manualmente a través del siguiente formulario.

P. Moncayo, Quito 170136, Ecuador

Destino

Centro Cultural Ilchimbía, Quito, Pichincha, Ecuador

Medio de transporte

En transporte público

Borrar Cómo llegar




Ilustración 1.9: Página principal del prototipo.

1.1.2.1. Características de accesibilidad

- La página web tiene la opción de navegar mediante la tecla Tabulador, entre elementos como campos de formulario, botones y bloques de información.
- Posee opciones para saltar contenido e ir directamente a las rutas y puntos de interés generados.
- El prototipo posee varias opciones de ruta que pueden ser escogidas con las indicaciones necesarias para llegar el destino. La Ilustración 1.10 muestra un menú de rutas con las instrucciones para llegar a la dirección de destino.

Ruta recomendada

Rutas sugeridas:

Iquique y Jose Maria Aguirre 1,5 km. 5 min aproximadamente

Jose Maria Aguirre 1,8 km. 6 min aproximadamente

Jose Maria Aguirre 1,7 km. 8 min aproximadamente

A P.Moncayo, Quito 170136, Ecuador

1,7 km. 8 min aproximadamente

1.	Dirígete hacia el suroeste en P.Moncayo hacia Yaguachi	56 m
2.	Gira a la derecha hacia Yaguachi	0,2 km
3.	Gira a la izquierda hacia Calle de Ote	0,2 km
4.	Gira ligeramente a la izquierda hacia Av. Gran Colombia	0,2 km
5.	Gira a la izquierda hacia Ramon Egas	0,1 km
6.	Gira a la derecha hacia Los Rios	0,1 km
7.	Gira a la izquierda hacia Julio Castro	0,2 km
8.	Gira a la derecha hacia Iquique	0,2 km
9.	Continúa por Jose Maria Aguirre. El destino está a la derecha.	0,5 km

B Jose Maria Aguirre, Quito 170136, Ecuador


Ilustración 1.10: Menú de rutas.

- La página posee instrucciones sobre el uso de algunas opciones como el uso de localización en tiempo real y el tipo de medio de transporte. En la Ilustración 1.11 se muestra el menú de servicio de ruta que permite utilizar la ubicación automáticamente mediante GPS.

Servicio de rutas

Buscador de direcciones y puntos de interés

Origen

 Usar tu localización

Pulsando sobre este botón la aplicación utilizará tu posición actual como punto de origen para el servicio de rutas. También puedes entrarla manualmente a través del siguiente formulario.

Introduce la dirección de origen

Destino

Introduce la dirección de destino

Medio de transporte

En transporte público ▼

Borrar

Cómo llegar

Ilustración 1.11: Sección de servicios de rutas del prototipo.

- La página está desarrollada en base a pautas de accesibilidad, además de hacer uso de WAI-ARIA como aria-live, aria-atomic y aria-relevant. [11]. La Ilustración 1.12 muestra el nivel de conformidad de la página según las pautas W3C aplicadas a HTML5, CSS, WAI-AA y WCAG 2.0.



Ilustración 1.12: Nivel de conformidad de accesibilidad de la página web.

1.1.3. LONDON SOUND SURVEY

Esta página muestra un mapa con el sistema fluvial de Londres, con sus puntos de referencia principales. Dentro de cada punto se encuentra un audio pregrabado con sonido ambiental que describe el entorno del punto seleccionado. La página también contiene mapas con las regiones de Londres que reproducen sonidos de la naturaleza para identificar cada lugar. La Ilustración 1.13 representa la vista principal de la sección "London's waterways" del sitio web "London Sound Survey".

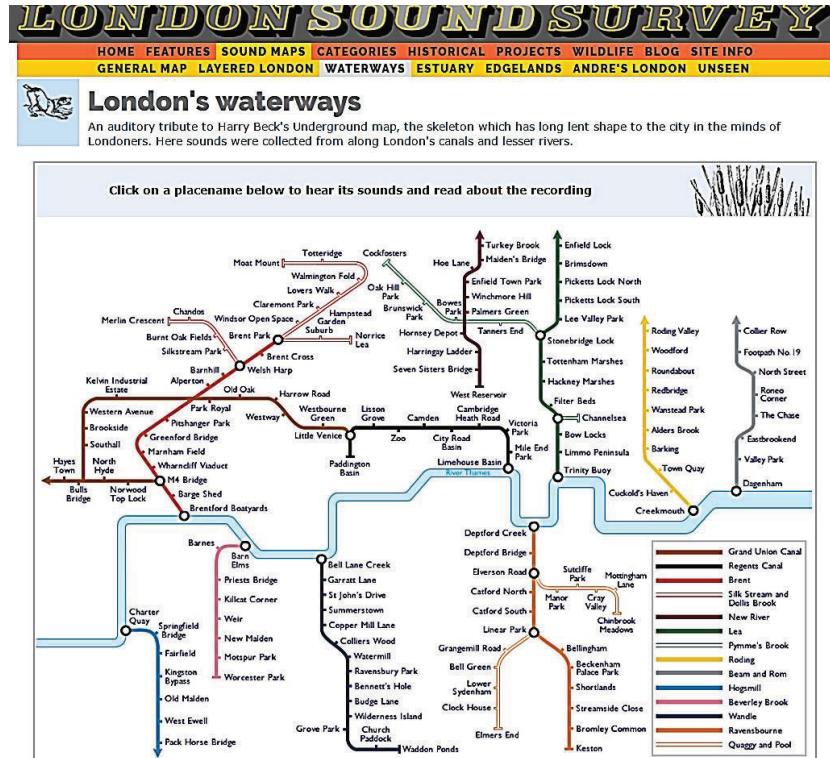


Ilustración 1.13: Vista principal de la página "London Sound Survey".

1.1.3.1. Características de accesibilidad

- La página web consiste de varios mapas sonoros, en donde se puede reproducir desde sonidos ambientales de los elementos del mapa, hasta descripciones textuales sobre el contenido de los elementos. La Ilustración 1.14 muestra un mapa con diferentes sonidos y descripciones de las regiones de Londres.

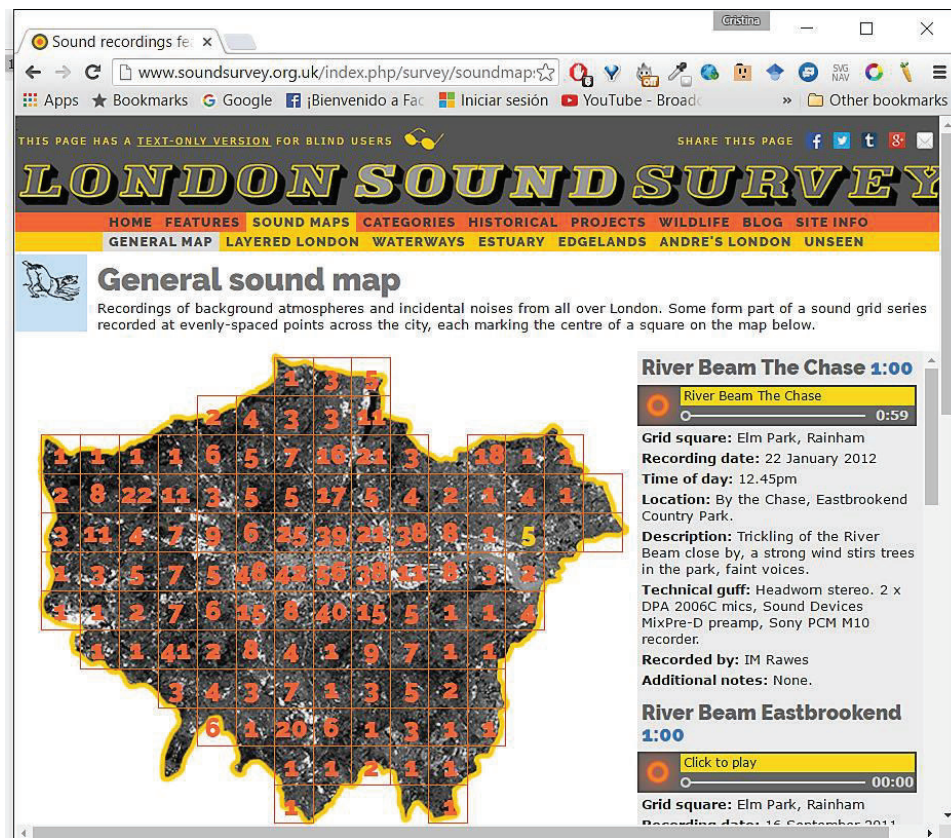


Ilustración 1.14: Ejemplo de mapa sonoro de la página "London Sound Survey".

- La página no pierde su calidad ni la distribución de su contenido al desplegarse en un dispositivo con otro tipo de resolución, como un Smartphone. En la Ilustración 1.15 se muestra al sitio web "London Sound Survey" desplegado en un dispositivo móvil adaptándose a la resolución de la pantalla.



Ilustración 1.15: Despliegue de la página "London Sound Survey" en un dispositivo móvil.

- A pesar de que la página utiliza colores para distinguir sus rutas, se puede apreciar que el contenido no es dependiente del color. La Ilustración 1.16 muestra la vista del mapa "London's waterways" en blanco y negro sin perder la calidad de la información transmitida.

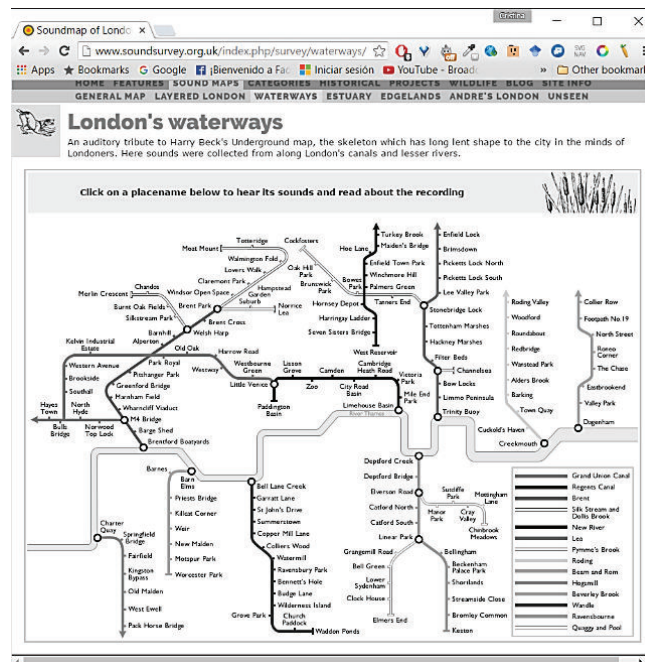


Ilustración 1.16: Vista de la página "London Sound Survey" en blanco y negro.

- Las distintas regiones del mapa pueden ser accedidas mediante la tecla Tabulador, que facilita la navegación y reproduce el audio respectivo a cada punto con la tecla Enter. La ilustración 1.17 representa una sección del mapa “London Sound Survey”, cuyos elementos son accesibles y navegables por teclado.



Ilustración 1.17: Mapa "London Sound Survey" accesible por teclado.

- La página posee un menú principal que puede ser accedido por teclado para dirigirse al contenido principal de la página o a los mapas. La ilustración 1.18 representa el menú de navegación con las diferentes secciones del sitio web.



Ilustración 1.18: Menú de navegación de la página "London Sound Survey".

- Las secciones de la página poseen un encabezado claro que describe el contenido de la página y los enlaces a los que dirige los elementos del mapa. La Ilustración 1.19 representa una sección del sitio web que representa el mapa de Londres subdividido en secciones.

LONDON SOUND SURVEY

HOME FEATURES SOUND MAPS CATEGORIES HISTORICAL PROJECTS WILDLIFE BLOG SITE INFO

GENERAL MAP LAYERED LONDON WATERWAYS ESTUARY EDGELANDS ANDRE'S LONDON UNSEEN

Layered London sound map

This sound map combines some recordings from the General Sound Map and the Sound Actions section into a single interface. Different historical map layers can be selected in turn as backgrounds to the modern-day sounds of the city.

LAYERS ON THE LONDON MAP

OPEN STREET MAP
Present-day map tiles derived from the Mapnik layer of the OpenStreetMap project, an open-source collaborative world map.

LAND UTILISATION 1930s
London part of a nationwide survey of land use intended as a 'modern-day Domesday book'. Much of the work was done by schoolchildren.

BOOTH LONDON POVERTY 1898
A colour-coded map of London depicting levels of poverty and wealth. Commissioned by the social reformer Charles Booth.

ORDNANCE SURVEY FIRST SERIES
London map derived from four separate sheets of the Ordnance Survey First Series, published from 1805 onwards.

Select a map square below and click to explore

Palmers Green, Edmonton, Tottenham

Ilustración 1.19: Mapa por regiones de la página "London Sound Survey".

- Proporciona una versión de solo texto, compatible con lectores de pantalla en donde se integra una descripción de todos los audios y una explicación de cada tipo de mapa. En la Ilustración 1.20 se muestra a versión de solo texto de la página web "London Sound Survey".

Andre's London sou: x

www.soundsurvey.org.uk/index.php/text/andre

Apps ★ Bookmarks Google f Bienvenido a F... Iniciar sesión YouTube - Bro... LoginsAE

[Jump to main content](#)

THE LONDON SOUND SURVEY: TEXT ONLY

MAIN SITE SECTION LINKS: TEXT VERSIONS

[Home page](#), [Sound actions](#), [Sound maps](#), [Wildlife](#), [About and contact](#).

MAIN SITE SECTION LINKS: GRAPHICS VERSIONS

[Home page](#), [London map](#), [Sound actions](#), [Sound maps](#), [London wildlife](#), [Historical](#), [Blog](#), [About](#), [Say hello](#).

ANDRE'S LONDON SOUND MAP RECORDINGS

INTRODUCTION BY ANDRE LOUIS: RECORDING IS MORE THAN HALF THE STORY

As a child I ran around the house with a cheap tape recorder, spools spinning, capturing anything and everything around me. I no longer have those tapes but I wish I did. Family friends you don't see any more. Family members that have passed away. How you sounded as a child. All gone. Sighted and blind people too have a tendency to forget people's voices after a while because they don't have recordings to refer back to. But because I've lived with blindness all my life and, because audio is what my life is about, I record all kinds of things and keep them for posterity. Even if I never play them again I have them.

LIST OF RECORDINGS

Ilustración 1.20: Vista textual de la página "London Sound Survey".

1.1.4. SECRETARÍA TÉCNICA PARA LA GESTIÓN INCLUSIVA EN DISCAPACIDADES

La siguiente página pertenece a la Secretaría Técnica para la Gestión Inclusiva en Discapacidades, se ha escogido esta página para el análisis ya que en Ecuador no se han desarrollado sitios web accesibles que vinculen mapas o planos. Además, que en esta página se han incluido varias opciones de accesibilidad y está desarrollada en base a WCAG 2.0.

La página web ha sido desarrollada para ser operada de forma universal, para ello ha implementado un lector de pantalla que lee las opciones y contenido de la página web, además ha incluido una ventana en donde se muestra una descripción de la sección en lenguaje de señas, posee un menú para cambiar los colores y el contraste, el tamaño de letra y una lupa para ampliar la visibilidad de secciones de la página [12].

En la Ilustración 1.21 se muestra la vista principal del sitio web de la Secretaría Técnica para la Gestión Inclusiva en Discapacidades. En la imagen se puede apreciar el menú principal, de accesibilidad y de contenido.



Ilustración 1.21: Vista principal de la página web de SETEDIS.

1.1.4.1. Características de accesibilidad

- La página web puede ser presentada en formato textual, auditivo y en lenguaje de señas. Además, posee un modo exclusivo para personas con discapacidad visual. En la Ilustración 1.22 se puede apreciar la declaración de accesibilidad del sitio web y su tabla de contenido.



Ilustración 1.22: Opción de lenguaje de señas para presentación de información.

- El texto de la página web puede ser leído con facilidad mediante una herramienta online que lee dinámicamente el contenido de la página web. En la Ilustración 1.23 se aprecia el lector de pantalla que se puede utilizar para describir el contenido del sitio web.

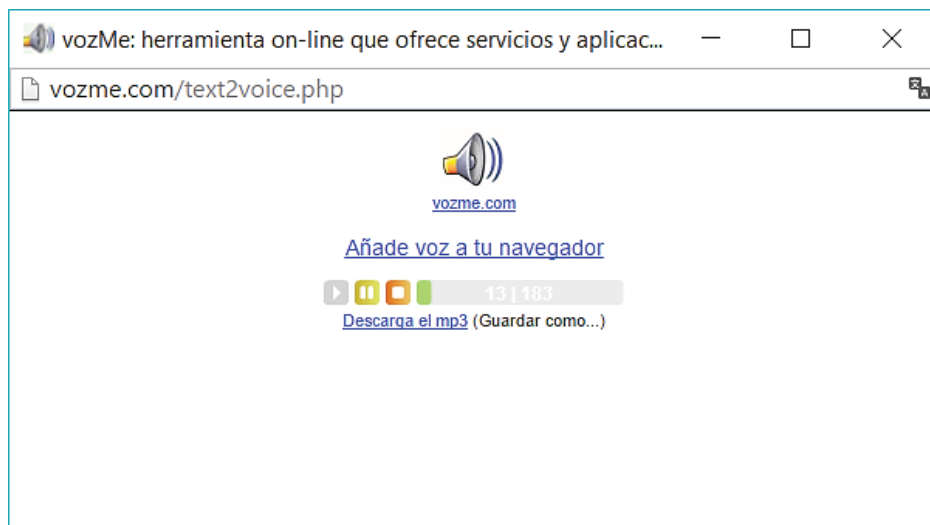


Ilustración 1.23: Lector de pantalla propio de la página web de SETEDIS

- La página puede ser desplegada en cualquier dispositivo sin perder la legibilidad de su contenido. En la Ilustración 1.24 se muestra el sitio web del SETEDIS desplegada en un dispositivo móvil. Se puede apreciar como el contenido se adapta a la resolución de pantalla.



Centros de Atención Integral

Ilustración 1.24: Visualización de la página web de SETEDIS en un dispositivo móvil.

- La página web integra varias opciones de cambio de color del estilo de la página para adaptarse a combinaciones de color que sean legibles para cualquier tipo de discapacidad visual. En la Ilustración 1.25 se aprecia el menú de accesibilidad integrado a la página web con opciones de cambio de estilo, tamaño de letra y tipo de fuente.



Ilustración 1.25: Menú de accesibilidad de la página web de SETEDIS.

- Los elementos de la página web pueden ser accedidos mediante teclado, como la navegación que se puede realizar con la tecla TAB.
- El contenido de la página es legible y la información se encuentra ubicada de manera ordenada para mejorar la navegabilidad.
- El menú de accesibilidad permite adaptar el contenido de la página a las necesidades del usuario, en caso de ser una persona con discapacidades visuales o auditivas. En la Ilustración 1.26 se puede ver la opción de lupa para ver el texto de la página web con incremento.



Ilustración 1.26: Opción de Zoom de la página web de SETEDIS.

A continuación, la tabla 1.2 muestra un resumen de las pautas de accesibilidad que se han analizado en los sitios web que contienen mapas accesibles. En el análisis se ha comprobado que los sitios web “Société de transport de Montréal” y “London Sound Survey” cumplen con todos los principios analizados. El principio Robusto no se ha tomado en cuenta para el análisis.

Tabla 1.2: Resumen de análisis de accesibilidad en sitios web con mapas accesibles

Sitio web / Pautas de accesibilidad	Perceptible	Operable	Comprensible
Société de transport de Montréal	✓	✓	✓
Prototipo accesible con Google Maps API V3	✓		✓
London Sound Survey	✓	✓	✓
Secretaría técnica para la gestión inclusiva en discapacidades	✓	✓	

1.2. ANÁLISIS DE PAUTAS DE ACCESIBILIDAD WCAG 2.0

Para garantizar el uso universal de páginas web, se han elaborado estándares y pautas de accesibilidad que guían y regulan el desarrollo de páginas web para garantizar un uso equitativo.

1.2.1. WCAG 2.0

Los documentos WCAG detallan cómo hacer páginas web accesibles a las personas con discapacidad, estas directrices están dirigidas a desarrolladores de contenido web, con el objetivo de hacer el contenido accesible. Permite herramientas de evaluación para identificar los problemas y las barreras a la accesibilidad en el contenido de la web.

WCAG 1.0 fue publicada como una Recomendación del W3C. Además, los documentos WCAG2.0 se han desarrollado sobre la base de tecnologías avanzadas fáciles de entender con mayor precisión a prueba.

WCAG se basa en 14 directrices de diseño, que incluyen medidas basadas en ejemplos cotidianos que explican cómo el diseño puede influir o limitar la universalidad información también incluye algunos puntos de control para verificar algunos errores probables.

Los puestos de control se asignan niveles de prioridad son las siguientes.

Prioridad 1: Son directrices que un desarrollador Web debe cumplir con el fin de que ciertos grupos de usuarios pueden tener acceso a los contenidos de un sitio web.

Prioridad 2: Estas son pautas que un desarrollador Web debe cumplir con el fin de que ciertos grupos de usuarios no tengan muchas dificultades para acceder a los contenidos de un sitio web.

Prioridad 3: Son directrices que un desarrollador Web debe cumplir con el fin de que ciertos grupos de usuarios no tengan serias dificultades para acceder a los contenidos de un sitio web [13].

Sobre la base de estas prioridades, se han establecido los siguientes niveles de cumplimiento:

Nivel de Conformidad "A": se cumplen todos los puntos de verificación de prioridad 1.

Nivel de Conformidad "AA": se cumplen todos los puntos de verificación de prioridad 1 y 2.

Nivel de Conformidad "AAA": Todos los puntos de verificación de prioridad 1, 2 y 3 se cumplen [13].

Estos lineamientos establecidos pueden ser considerados para los sitios web accesibles, por lo que se ofrece el acceso a páginas legibles y útiles para cualquier individuo.

1.3. JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO

Una metodología de desarrollo consiste en un grupo de procesos, herramientas y técnicas basadas en procedimientos documentales para guiar el desarrollo de software. Se fundamenta en un ciclo de vida que muestra las varias etapas que muestran cual debe ser el resultado final de cada una de ellas [14].

Para este proyecto se ha considerado el uso de metodologías ágiles para desarrollar un prototipo cuyos requerimientos se basan en pautas de accesibilidad, que por lo general se tiene cambios constantes en los

requerimientos según las pautas que se van integrando. Esto implica la necesidad de usar una metodología flexible a requerimientos. Además, para obtener un prototipo que cumpla con los requisitos del usuario, se debe verificar que su desarrollo se apegue a las pautas de accesibilidad lo que implica elaborar varias pruebas a lo largo del ciclo de vida, lo que garantiza la calidad de cada módulo antes de continuar con la siguiente iteración, de esta manera se evita realizar una sola prueba al final que pondría en riesgo al prototipo.

Además, para cumplir con los objetivos de este estudio, se desarrolló un producto funcional, cuyo desarrollo debe ser rápido sin enfocarse principalmente en la documentación, es decir que se debe dar prioridad a la funcionalidad y calidad de desarrollo [15].

Las metodologías escogidas para el análisis son las siguientes:

Extreme Programming (XP), es una metodología ágil usada en desarrollo de software cuando los requerimientos son variantes o ambiguos. Se basa en adaptabilidad y flexibilidad a cambios y está centrada en valores como simplicidad, retroalimentación, coraje y comunicación [16].

Scrum es un marco de desarrollo de software basado en prácticas, roles y artefactos para entregar productos de valor y calidad. Su estrategia se basa en el desarrollo incremental. La calidad del producto se basa en calidad del equipo de trabajo más que en calidad de los procesos.

Seguimiento de las fases simultáneamente [14].

Iconix Es una metodología de desarrollo basada en Extreme programming y Rational Unified Process (RUP). Es un proceso ágil para obtener un sistema informático. Se dedica a la construcción de sistemas de gestión de mediana y pequeña complejidad con la participación de los usuarios finales [17].

El siguiente cuadro muestra una comparación entre las metodologías de desarrollo ágiles más comunes.

Tabla 1.3: Tabla comparativa entre las metodologías ágiles XP, Scrum e Iconix.

CARACTERÍSTICAS	XP	SCRUM	Iconix	Observaciones
Adaptable a todo tamaño de proyecto	X	✓	X	Iconix y XP se usan para proyectos pequeños y medianos.
Definición de requerimientos mediante historias de usuario.	✓	✓	X	Iconix utiliza casos de uso.
Tamaño del equipo de trabajo menor a 10 personas.	✓	✓	✓	
Establece reuniones diarias	✓	✓	X	Solo se establecen reuniones iniciales para definir requerimientos.
Flexibilidad de cambios	✓	✓	X	
Participación del cliente en el desarrollo del producto.	✓	✓	X	El cliente solo proporciona los requerimientos iniciales
Experiencia previa	✓	✓	X	

Con este análisis se ha determinado que la metodología ágil que se adapta de mejor manera al desarrollo del prototipo es Scrum.

Esto se debe al tipo de proyecto y el tiempo en el que se desarrollará el prototipo, por lo que se requiere que la metodología sea flexible y se puedan evidenciar entregables en un menor plazo de tiempo.

Además, se requiere de iteraciones que permitan conocer y evaluar el avance del desarrollo a medida que se prueba y se otorga valor al mismo. Ya que la metodología cuenta con pocos roles principales, estos pueden ser cubiertos con un equipo de trabajo pequeño como el que se cuenta para desarrollar este proyecto.

1.3.1. METODOLOGÍA DE DESARROLLO SCRUM

1.3.1.1. Artefactos de Scrum

1.3.1.1.1. *Product Backlog*

El product backlog corresponde a un listado de las funcionalidades que serán implementadas en el desarrollo del producto y se considera la fuente inicial de requisitos de usuario; además evoluciona en base al desarrollo del producto, en donde se pueden agregar nuevas características, correcciones y mejoras que entregarán valor al producto final.

El product Owner se encarga de gestionar el listado, mantener los elementos ordenados según la prioridad. Ésta se determina en base a los elementos que otorgarán mayor valor al producto y que por lo tanto determinan las tareas que se desarrollarán de manera inmediata. Los elementos que tienen una prioridad mayor deben ser comprendidos y detallados.

Los elementos que conforman el product backlog poseen atributos como, un identificador único, prioridad, una descripción del requisito, el campo y una estimación de esfuerzo [14].

1.3.1.1.2. *Sprint Backlog*

El Sprint backlog es un listado que integra las desintegra las funcionalidades del product backlog en las actividades que serán desarrolladas en el sprint. Se elabora en la reunión de planificación del sprint en donde se asigna las tareas a los miembros del equipo de Scrum e indicando el tiempo y esfuerzo que requiere cada tarea.

Este artefacto descompone las tareas en unidades más pequeñas para que puedan ser monitoreadas diariamente e identificar con mayor facilidad riesgos y problemas según el desarrollo. Se consideran a las tareas grandes las que requieren de más de un día para ser elaboradas.

El sprint backlog únicamente puede ser modificado por el equipo de trabajo, éste debe cubrir las tareas identificadas para cumplir con el sprint. En la Ilustración 1.27 se puede ver un ejemplo de la estructura del product backlog y como se relaciona con el sprint backlog [14].

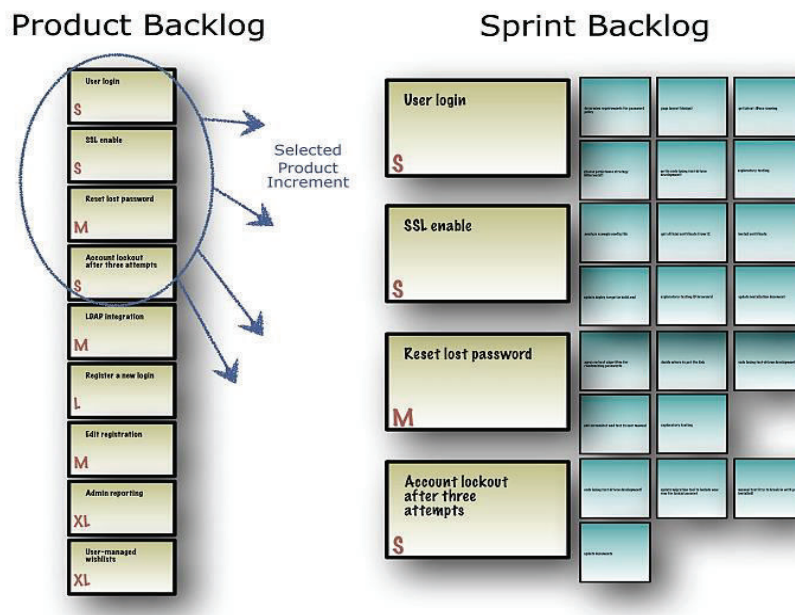


Ilustración 1.27: Product Backlog y Sprint Backlog.

1.3.1.1.3. *Sprint*

El ciclo de vida de Scrum se subdivide en Sprints que corresponde a iteraciones en donde se diseña, codifica y prueba el producto. Por lo general dura entre 2 a 4 semanas.

Los Sprints se dividen en tareas que se encargan de especificar el desarrollo de cada elemento del product backlog. Cada tarea es desarrollada por un responsable y pueden ser reestimadas en tiempo, esfuerzo y razones de demora. Las tareas no solo incluyen actividades de desarrollo, sino también pruebas, actualización de software, documentación, etc.

1.3.1.1.4. *Incremento*

El incremento es el resultado al final de cada Sprint, su principal característica es que debe estar completamente operativo, probado y en condiciones de entregarse al cliente. Los entregables se refieren a funcionalidades operativas no a trabajos de diseño interno.

1.3.1.1.5. *Reuniones Scrum*

En la Tabla 1.4 se puede ver una descripción de las reuniones que existen en Scrum con su respectiva descripción.

Tabla 1.4: Descripción de las reuniones de Scrum.

Reuniones	Descripción
Reunión de planificación del Sprint	Es una reunión que se realiza al inicio de cada sprint en donde se determina el objetivo y las tareas para desarrollar.
Reunión diaria	Es una reunión breve donde se discuten los siguientes temas: El trabajo que se realizó el día anterior. El trabajo por realizar. Los impedimentos y obstáculos que deben eliminarse.
Reunión de revisión del Sprint	Se elabora un análisis y observación el incremento desarrollado y como el product backlog se debe adaptar.
Reunión de retrospectiva del Sprint	Se revisa los acontecimientos dentro del sprint y se elabora un plan de mejoras para implementar en el siguiente sprint.

1.3.1.1.6. Roles de Scrum

En la Tabla 1.5 se puede ver una descripción de los roles de Scrum con su respectiva descripción.

Tabla 1.5: Descripción de los roles de Scrum.

Roles	Descripción
Product Owner	Toma las decisiones sobre el producto y proporciona valor al mismo. Se encarga de gestionar el product backlog, es decir que tiene el poder de aprobar o rechazar los cambios e incrementos y designar prioridades a las tareas.
Scrum Master	Es la persona responsable de guiar el proceso de desarrollo en base a las reglas de Scrum. Proporciona asesoramiento y formación al equipo de desarrollo y al product owner, resolviendo impedimentos en el ambiente de trabajo, promoviendo prácticas de desarrollo de software para elaborar el producto final.
Equipo de desarrollo	Lo conforman las personas responsables de elaborar el incremento en cada sprint. El equipo de desarrollo debe constar de no más de 9 personas ya que dificulta la comunicación y toma de decisiones. Las tareas a realizar se designan a los diferentes miembros del equipo, pero trabajan por un objetivo común. El equipo de trabajo está liderado por una entidad principal que vela por el cumplimiento de las tareas de cada individuo.

1.4. JUSTIFICACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

1.4.1. HTML 5

Hyper Text Markup Language es un lenguaje utilizado para mostrar contenido web. Funciona como una referencia estándar definido por el W3C para el desarrollo de páginas web. HTML contiene una estructura basada en código para mostrar contenido web que puede ser texto, imágenes, datos, audio, video, multimedia, tablas de datos, etc.

Para construir un sitio web, debe estar compuesto de HTML, hojas de estilo o CSS y scripts que describen el comportamiento de la página.

Un documento HTML se define con la extensión .html que permite que el navegador para identificar cómo se debe estructurar la información contenida en el documento.

Código básico se definen mediante la apertura y cierre de etiquetas, que se estructuran por paréntesis angulares, donde se definen los atributos y elementos de contenido.

La estructura básica de un documento HTML es parte de la definición del documento en la primera línea, una etiqueta HTML que indica apertura y cierre de la parte superior de la página, el contenido interno se divide en cabeza y el cuerpo. Dentro de la etiqueta de la cabeza tiene la página de información general, mientras que el cuerpo tiene el contenido que se mostrará en el navegador.

Una estructura del documento HTML se compone de definición de documento que se realiza en la primera línea, también contiene una etiqueta HTML que indica apertura y cierre, la parte superior de la página y el contenido interno se divide en cabeza y cuerpo. Dentro de la etiqueta de la cabeza tiene la página de información general, mientras que el cuerpo tiene el contenido que se mostrará en el navegador [18].

La Tabla 1.6 elabora un análisis entre XHTML [19], HTML [19] y HTML5 [20].

Tabla 1.6: Tabla comparativa entre XHTML, HTML y HTML5.

Características	XHTML	HTML	HTML5	Observaciones
Aplicación basada en Lenguaje de Marcado Generalizado Estándar (SGML)	X	✓	✓	
Permite minimización de atributos y etiquetas.	X	✓	✓	XHTML es un formato estricto que produce documentos bien formados.
Separa contenido de forma.	✓	X	X	En HTML y HTML 5 se puede incluir atributos de estilo en las etiquetas.
Posee atributos para definir formatos y estilos.	X	✓	✓	
Flexible respecto al cierre de etiquetas y uso de atributos.	X	✓	✓	
Implementa etiquetas para uso de elementos multimedia y nuevas API's.	X	X	✓	HTML5 al ser una nueva versión incluye etiquetas para creación de estilos y multimedia.

HTML5 es la herramienta que más se ajusta a las necesidades del prototipo, debido a la necesidad de incluir elementos multimedia y API's que ayudan a la visualización de gráficos y animaciones en SVG. Además, es un lenguaje flexible en el uso de atributos y etiquetas para manejo de estilos y programación dentro del código.

1.4.2. CSS3

CSS fue creado con el fin de gestionar un formato estilizado de las páginas web, separando el contenido HTML del diseño. Esta separación entre la estructura y

el estilo de la página proporciona beneficios en el mantenimiento, el rendimiento, la producción y la mejora de la calidad de estilos [21].

CSS 3 a diferencia de CSS 2 incluye nuevas mejoras como la velocidad de carga de estilos y visualización. Además, proporciona separación por módulos, es decir varios documentos que incluyen nuevas funcionalidades sin afectar al resto de la estructura, entre los módulos publicados se encuentran Media Queries, Namespaces, Selectors y Color [22].

Al mismo tiempo, CSS 3 incluye nuevos elementos como pseudoclasas que se relaciona a la posición de los elementos en el documento, animaciones, bordes, gradientes, efectos de texto y divisiones en base a las columnas del texto [23].

1.4.3. JAVASCRIPT

JavaScript es un lenguaje de programación, el código se desarrolla a través de secuencias de comandos que se pueden incrustar en el código HTML o en referencia al documento HTML. Es un idioma que se utiliza para dar el comportamiento de los elementos de una página web como botones, formularios, enlaces, etc.

JavaScript fue desarrollado a partir de Java, pero no es lo mismo que Java es un lenguaje completo, mientras JavaScript, sólo funciona en un entorno de páginas web [24].

El principal motivo para usar JavaScript se debe a que es un lenguaje sencillo y netamente orientado al desarrollo web, lo que lo hace compatible con la mayoría de navegadores.

Además, JavaScript permite desarrollar aplicaciones multicapa, es decir desde la interfaz de usuario hasta la conexión de la base de datos. JavaScript es compatible con varias librerías que permiten crear aplicaciones dinámicas en donde se puede controlar la presentación visual, el comportamiento y los datos de una manera ordenada y simple.

1.4.4. SVG

Scalable Vector Graphics es un lenguaje vectorial utilizado para crear gráficos vectoriales, de trama y texto en formato XML. La versión 1.1.2 de SVG es la versión recomendada por la W3C ya que es soportado por la mayoría de navegadores web [25]. La tabla 1.7 muestra una comparación entre SVG y Canvas.

Tabla 1.7: Tabla comparativa entre Canvas y SVG.

Características	SVG	Canvas	Observaciones
Estándar de W3C	✓	X	SVG es un estándar recomendado por la W3C para creación de imágenes vectoriales.
Creación de gráficos, imágenes y formas en 2D.	✓	✓	
No se pierde la calidad de la imagen al cambiar de resolución.	✓	X	SVG no pierde la calidad al desplegarse en otra resolución ya que se basa en gráficos vectoriales, mientras que Canvas es un formato en base a mapa de pixeles.
Compatible con lenguajes como HTML, CSS y JavaScript.	✓	X	
Flexible a implementación de estándares de accesibilidad	✓	X	SVG al ser una recomendación de la W3C, es compatible con el uso de estándares de accesibilidad.

SVG tiene la facilidad para ser integrado directamente al código HTML CSS y el uso de estilos, posee un formato escalable que permite una mayor definición y se puede adaptar de acuerdo con la resolución o la especificación de cualquier dispositivo.

SVG se ha escogido debido a la facilidad para ser integrado directamente al código HTML, en donde sus elementos pueden ser manipulados independientemente al resto del código en formato de etiquetas en base a estándares de accesibilidad.

SVG es compatible con CSS y el uso de estilos, posee un formato escalable que permite una mayor definición y se puede adaptar de acuerdo con la resolución o la especificación de cualquier dispositivo.

1.4.5. SAILS.JS

Sails.js es un marco de trabajo que permite construir aplicaciones web basadas en Node.js. Su diseño se basa en la arquitectura Modelo Vista Controlador (MVC) que se utiliza para separar los datos de la aplicación de la lógica del negocio.

Sails soporta cualquier motor de base de datos y trabaja de forma transparente con su Object Relational Model (ORM) Waterline. Un ORM es un modelo de programación que utiliza entidades como objetos para acceder a los datos, con la facilidad de manejar operaciones create, read, update y delete (CRUD) en un lenguaje orientado a objetos. Las aplicaciones desarrolladas en Sails se construyen completamente en JavaScript [26].

Se ha escogido sails como marco de trabajo debido al tipo de arquitectura MVC, ante la necesidad de crear una aplicación web robusta y escalable. Este esquema de programación basado en un modelo de capas, se ha utilizado para separar la lógica del negocio, lógica de usuario y lógica de datos, proporcionando a la vez un acoplamiento entre estos elementos. Además proporciona compatibilidad con librerías JavaScript para presentación visual y librerías usadas en animación que son compatibles con SVG.

CAPÍTULO 2. ANALISIS, DISEÑO Y ELABORACIÓN DEL PROTOTIPO

La aplicación tiene como objetivo desplegar un plano interior accesible que proporcione una ayuda a las personas no videntes a localizarse e identificar elementos en ambientes desconocidos. La aplicación contará con funciones de accesibilidad que faciliten la interacción con el prototipo y tomará en cuenta pautas de accesibilidad WCAG 2.0.

En este capítulo se procede a describir el proceso de desarrollo de la aplicación, por lo cual se usa SCRUM como metodología de desarrollo en base al análisis desarrollado en el capítulo 1.

2.1. ESPECIFICACIÓN DE REQUERIMIENTOS.

Para establecer los requerimientos iniciales, se ha elaborado la Tabla 2.1 que define los roles SCRUM del equipo de trabajo.

Tabla 2.1: Roles de Scrum.

Rol	Persona encargada	Descripción
Product Owner	Tania Calle	Negocia decisiones sobre el producto, con el cliente.
Scrum Master	Sandra Sánchez.	Guía el equipo en la metodología Scrum.
Equipo de desarrollo	Cristina Rivera.	Encargada de diseñar y desarrollar la aplicación.

Los roles para el equipo de Scrum se han escogido en base a las capacidades de cada individuo. Tania Calle se ha escogido como Product Owner debido a su conocimiento en el campo de accesibilidad y comprensión de los requerimientos de usuarios no videntes, es la persona intermediaria entre el equipo de trabajo y el cliente.

Como Scrum Master se ha escogido a Sandra Sánchez debido a su especialización en la metodología Scrum, por lo cual dirige al equipo de desarrollo en uso de la metodología.

Como equipo de desarrollo se tiene a Cristina Rivera como encargada del diseño y desarrollo de la aplicación, así como la realización y ejecución de las pruebas para verificar los resultados del prototipo.

2.1.1. HISTORIAS DE USUARIO Y CRITERIOS DE ACEPTACIÓN

Las historias de usuario definen de manera simple las funcionalidades del sistema y permiten la creación del product backlog. Consiste en describir en lenguaje común del usuario las especificaciones que serán incorporadas al sistema, para ello estas deben ser acordadas entre el cliente y el equipo de desarrollo.

Las historias de usuario deben recoger una cantidad mínima de información que sea relevante para el proyecto, es decir que deben ser concretas y específicas para que puedan ser estimadas y verificables [27].

2.1.1.1. Perfiles de la aplicación

El prototipo no requiere de perfiles de usuario, por lo cual se ha establecido un usuario general como perfil único. Este usuario puede o no tener discapacidades visuales como ceguera, baja visión o daltonismo.

Las preferencias de cada usuario se escogen en la misma aplicación según su discapacidad sin recurrir al uso de perfiles.

2.1.1.2. Definición de historias de usuario

Para definir los parámetros que se utilizarán en las historias de usuario, se ha utilizado una plantilla llamada “Historias de usuario y criterios de aceptación” que puede ser usada en un entorno de trabajo con SCRUM e incluye la documentación referente a las descripciones de los parámetros de las historias y criterios de aceptación [28].

La tabla 2.2 muestra los elementos que se han escogido para elaborar las historias de usuario con su respectiva descripción.

Mientras que la tabla 2.3 incluye las historias de usuario que se han elaborado para la construcción del prototipo y los criterios de aceptación.

Tabla 2.2: Elementos y descripciones para historias de usuario y criterios de aceptación.

Elemento	Descripción
Id de la historia	Es el código que identifica la historia de usuario. Su formato se establece por el equipo de desarrollo.
Rol Cambiar por perfil	El rol indica la función que está ejerciendo una persona en el equipo de trabajo.
Característica/Funcionalidad	Relata las funciones que el rol desea ejecutar en el sistema
Razón / Resultado	Muestra los resultados que el rol obtiene al realizar una acción.
Número de Escenario	Identifica un valor de la historia de usuario asociado a un escenario.
Criterio de Aceptación (Título)	Descripción del escenario que es descrito en contexto.
Evento	Es la acción que el usuario realiza.
Resultado / Comportamiento esperado	Es la consecuencia al ejecutar una acción por parte del usuario.

Tabla 2.3: Historias de usuario y criterios de aceptación [28].

Enunciado de la historia				Criterios de aceptación			
ID de la historia	Rol	Característica/ Funcionalidad	Razón/ Resultado	No de Escenario	Criterio de Aceptación (Título)	Evento	Resultado/ Comportamiento esperado
P01AT	Como usuario general	Necesito acceder a la página utilizando un lector de pantalla.	Con el objetivo de escuchar las descripciones de los elementos.	1	Imágenes	Colocar el foco del lector de pantalla sobre el elemento.	Lectura del atributo "alt", "title" y descripción de la imagen.
				2	Botones	Colocar el foco del lector de pantalla sobre el elemento.	Lectura del nombre del botón, el tipo de elemento y su estado (desplegado o no desplegado)
				3	Bloques de información	Colocar el foco del lector de pantalla sobre el elemento.	Lectura del texto dentro del bloque de información.
P02MT	Usuario general	Deseo acceder a medios auditivos.	Con el fin de conocer la información.	1	Descripciones auditivas	Colocar el foco en cualquier elemento.	Lectura del nombre y descripción, texto que se encuentre en el elemento.
				2	Audio pregrabado	Dar click al botón de lectura de ruta de información.	Reproducción del audio pregrabado.
P03A	Usuario general	Deseo acceder al contenido de forma ordenada	Con el objetivo de entender la información desplegada.	1	Información	Colocar el foco del lector de pantalla sobre el elemento.	Lectura del contenido.
				2	Lectura de contenido	Colocar el foco del lector de pantalla sobre el elemento.	Lectura del contenido.
				3	Distribución del contenido	Colocar el foco del lector de pantalla sobre el elemento.	Lectura del contenido.

P04D	Usuario general	Deseo interactuar con el contenido de la página y sus elementos.	Con el fin de conocer la información.	1	Uso de color	Dar click a las barras de cambio de combinación de color.	Los colores de la página y el plano se adaptarán según los cambios que realice el usuario.
				2	Uso de estilos (texto y contraste)	Dar click a los botones de cambio de estilo, tipo y tamaño de letra.	Los estilos de la página cambiarán de color, el tamaño de letra y el tipo de letra cambiarán.
				3	Control de audio	Seleccionar un elemento de la página.	El usuario escuchará la descripción del elemento.
				4	Gráficos	Seleccionar un elemento del plano.	El usuario escuchará la descripción del elemento.

O01AT	Usuario general	Deseo navegar por la página mediante el teclado	Con el objetivo de no utilizar el canal visual para ubicar elementos.	1	Acceso por teclado secuencial.	Presionar la tecla tabulador para navegar.	El foco se moverá en orden secuencial, de izquierda a derecha.
				2	Acceso por teclado espacial.	Presionar los cursores para navegar.	El foco se moverá según el cursor presionado, es decir, arriba, abajo, izquierda, derecha.
				3	Atajos de teclado	Presionar la tecla Alt combinada de otras teclas.	El elemento asociado a los atajos de teclado recibirá el foco.

O04N	Usuario general	Deseo navegar con facilidad por la página web.	Con el objetivo de conocer la información	1	Títulos encabezados	Seleccionar un elemento con ratón o teclado.	El lector de pantalla indicará al usuario si ha seleccionado un título.
-------------	-----------------	--	---	---	---------------------	--	---

				de la página sin recurrir al canal visual.	2	Formas de navegación	Colocar el foco sobre un elemento.	El lector de pantalla indicará el tipo de elemento seleccionado.
					3	Orden y visibilidad del foco	Colocar el foco sobre un elemento.	El foco se moverá al elemento seleccionado.

C01L	Usuario general	Deseo comprender la información transmitida en la página.	Con el objetivo de aprender sobre	1	Idioma	Seleccionar un bloque de texto.	El lector de pantalla leerá el texto según el idioma con el que fue configurado.
				2	Pronunciación	Seleccionar un bloque de texto.	El lector de pantalla leerá el texto según el idioma con el que fue configurado.
				3	Lectura	Seleccionar un bloque de texto.	La lectura será en base a la pronunciación del lector de pantalla.

C02P	Usuario general	Deseo realizar acciones en la página y obtener un resultado esperado.		1	Movimiento del foco	Seleccionar un elemento de la página.	El foco se colocará sobre el elemento seleccionado.
				2	Cambio de petición	Seleccionar un elemento de la página.	El foco se colocará sobre el elemento seleccionado.
				3	Navegación coherente	Presionar tecla Tabulador para navegar.	El foco se colocará sobre el elemento ubicado a la derecha del elemento actual.

2.1.2. HISTORIAS DE USUARIO TÉCNICAS

A continuación, se desarrollan las historias de usuario técnicas con el objetivo de determinar las historias de usuario relevantes para la construcción del prototipo. Para elaborar las historias de usuario, se ha tomado como referencia las siguientes plantillas para historias de usuario [28].

2.1.2.1. Definición de historias de usuario.

2.1.2.1.1. Definición de arquitectura del prototipo.

La tabla 2.4 describe la historia de usuario 1 que consiste en la definición de la arquitectura del prototipo.

Tabla 2.4: Historia de usuario - Diseñar arquitectura del prototipo.HU01

Historia de Usuario	
Número: 1	Usuario: Usuario general
Nombre historia: Definir arquitectura del prototipo.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Iteración asignada: 1	Responsable: Cristina Rivera
<p>Descripción:</p> <p>El usuario hace uso de la aplicación, cuya arquitectura se basa en el esquema Modelo Vista Controlador. Este esquema de programación basado en un modelo de capas, se usará para separar la lógica del negocio, lógica de usuario y lógica de datos, proporcionando a la vez un acoplamiento entre estos elementos.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Diseñar arquitectura de la aplicación. 2. Diseño de la base de datos de coordenadas y el modelo del ORM. 3. Definición de controladores y funciones. 4. Diseño de los elementos de la vista principal del prototipo. 	
<p>Observaciones:</p> <p>La arquitectura de la aplicación será modelada previamente antes de implementarse en Sails.js.</p>	

2.1.2.1.2. Plano interior accesible.

La tabla 2.5 describe la historia de usuario 2 que consiste en diseñar e implementar plano interior accesible.

Tabla 2.5: Historia de usuario - Diseñar e implementar plano interior accesible.HU02.

Historia de Usuario	
Número: 2	Usuario: Usuario General
Nombre historia: Diseñar e implementar plano interior accesible	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Iteración asignada: 1	Responsable: Cristina Rivera
<p>Descripción:</p> <p>El usuario accede a la aplicación para interactuar con los componentes del plano interior accesible. El usuario podrá escoger el elemento deseado, ya sea con ratón o teclado con el fin de conocer la información del mismo.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Diseño del plano interior accesible a escala en base a medias reales. 2. Incluir información correspondiente a cada elemento. 3. Incluir atributos de accesibilidad como "title", "alt" y "description". 4. Incluir estilos y formatos con CSS. 	
<p>Observaciones:</p> <p>El plano será construido en formato SVG.</p>	

2.1.2.1.3. Diseñar Interfaz principal de la aplicación.

La tabla 2.6 describe la historia de usuario 3 que consiste en diseñar e implementar la interfaz principal del sitio web.

Tabla 2.6: Historia de usuario - Diseñar e implementar interfaz principal del sitio web.HU03

Historia de Usuario	
Número: 3	Usuario: Usuario General
Nombre historia: Diseño e implementar interfaz principal del sitio web.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Iteración asignada: 1	Responsable: Cristina Rivera
<p>Descripción: El usuario accede a la aplicación para interactuar con los bloques de información y módulos correspondientes al prototipo. En la parte central se coloca el plano interior accesible para poder navegar por sus elementos y desplegar la información en el panel derecho. En la parte superior se tiene una barra de menú.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Diseño de la interfaz principal de acceso al prototipo. 2. Debe contener un menú al lado izquierdo y un panel donde se desplegará la información al lado derecho. 3. Incluir atributos de accesibilidad para ser compatible con lectores de pantalla. 4. Los colores se establecerán en base a pautas de accesibilidad. 	
<p>Observaciones: El diseño del prototipo se establecerá en base a pautas de diseño establecidas por la WCAG2.0.</p>	

2.1.2.1.4. Diseñar menú de accesibilidad.

La tabla 2.7 describe la historia de usuario 4 que consiste en diseñar e implementar el menú de accesibilidad.

Tabla 2.7: Historia de usuario - Diseñar e implementar menú de accesibilidad.HU04.

Historia de Usuario	
Número: 4	Usuario: Usuario General
Nombre historia: Diseñar e implementar menú de accesibilidad	
Prioridad en el negocio: Media	Riesgo en desarrollo: Alto
Iteración asignada: 1	Responsable: Cristina Rivera
<p>Descripción:</p> <p>El usuario accede a la aplicación para interactuar con los componentes del plano interior accesible. Para adaptar el contenido de la página a sus necesidades visuales, accederá al menú de accesibilidad para cambiar el estilo de la página, el tamaño, tipo y espaciado de letra. Además, podrá cambiar el estilo del plano, agregar patrones para diferenciar regiones y cambiar las tonalidades del plano.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Diseño de la barra de menú de accesibilidad. 2. Incluir opciones para cambiar el aspecto visual de la página, como combinaciones de color de alto y bajo contraste, y uso de colores para personas con discapacidad del color. 3. Incluir opciones de cambio de tamaño y tipo de fuente para personas de baja visión. 4. Incluir varias opciones de cambio de tonalidades tanto en el plano como en la página para personas con discapacidad al color. 5. Incluir patrones en el estilo del mapa para distinguir las regiones para personas ciegas al color. 	
<p>Observaciones:</p> <p>Incluir atajos de teclado para acceder al menú de accesibilidad.</p>	

2.1.2.1.5. Implementar modo simulación

La tabla 2.8 describe la historia de usuario 5 que consiste en la implementación del modo simulación.

Tabla 2.8: Historia de usuario - Implementar modo simulación.HU05.

Historia de Usuario	
Número: 5	Usuario: Usuario General
Nombre historia: Implementar modo simulación	
Prioridad en el negocio: Media	Riesgo en desarrollo: Alto
Iteración asignada: 2	Responsable: Cristina Rivera
<p>Descripción:</p> <p>El usuario accede al modo simulación con el fin de conocer las rutas preestablecidas para ubicar elementos en el plano. Cuando el usuario selecciona la ruta deseada, esta se grafica en el plano interior y al mismo tiempo se reproduce un audio que ofrece indicaciones de cómo llegar el elemento en base a los elementos contiguos.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Diseño del menú de rutas y la vista preliminar de la graficación. 2. Ingresar coordenadas para graficación de rutas en el ORM. 3. Graficar las coordenadas dinámicamente en el plano utilizando SVG. 4. Incluir audio pregrabado para describir las indicaciones para llegar el elemento seleccionado. 	
<p>Observaciones:</p> <p>El audio se reproduce a medida que la ruta es graficada dinámicamente.</p>	

2.1.2.1.6. Implementar navegabilidad en la página

La tabla 2.9 describe la historia de usuario 6 que consiste en implementar la navegabilidad en el sitio web.

Tabla 2.9: Historia de usuario - Implementar navegabilidad en el sitio web.HU06.

Historia de Usuario	
Número: 6	Usuario: Usuario General
Nombre historia: Implementar navegabilidad en el sitio web.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Alto
Iteración asignada: 2	Responsable: Cristina Rivera
<p>Descripción: El usuario puede interactuar con los elementos de la página mediante el teclado, esto lo puede realizar de manera secuencial o mediante los cursores, arriba, abajo, izquierda o derecha. Esto facilita el uso del prototipo a personas no videntes.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Implementar navegación con "tabindex". 2. Implementar navegación con cursores. 	
<p>Observaciones: La navegación con cursores se implementará con funciones en JavaScript.</p>	

2.1.2.1.7. Implementar navegabilidad en el plano interior

La tabla 2.10 describe la historia de usuario 7 que consiste en implementar la navegabilidad en el plano interior.

Tabla 2.10: Historia de usuario - Implementar navegabilidad en el plano interior.HU07.

Historia de Usuario	
Número: 7	Usuario: Usuario General
Nombre historia: Implementar navegabilidad en el plano interior	
Prioridad en el negocio: Media	Riesgo en desarrollo: Alto
Iteración asignada: 2	Responsable: Cristina Rivera
<p>Descripción: El usuario accede a la aplicación para interactuar con los elementos del plano interior. Para ello puede utilizar tanto navegación secuencial como por cursores con el fin de moverse entre los elementos y oficinas del plano interior. Esto facilita la usabilidad a personas no videntes.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Implementar navegación con "tabindex". 2. Implementar navegación con cursores. 	
<p>Observaciones: La navegación con cursores se implementará con funciones en JavaScript.</p>	

2.1.2.1.8. Diseñar menú de ayuda

La tabla 2.11 describe la historia de usuario 8 que consiste en el diseño e implementación del menú de ayuda.

Tabla 2.11: Historia de usuario - Diseñar Menú de ayuda.HU08.

Historia de Usuario	
Número: 8	Usuario: Usuario General
Nombre historia: Diseñar e implementar menú de ayuda	
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Iteración asignada: 3	Responsable: Cristina Rivera
<p>Descripción: El usuario accederá al menú de ayuda para conocer la estructura principal del prototipo, las indicaciones para su navegación y los diferentes modos en los que opera. La información será proporcionada textualmente.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Diseño del plano interior accesible a escala en base a medias reales. 2. Incluir una descripción detallada de la estructura del prototipo y los elementos que lo componen. 3. Incluir una descripción de las indicaciones para navegar por el plano mediante el teclado y los atajos que se proporcionan. 4. Incluir una descripción de los modos en los que opera el prototipo y como acceder a ellos. 	
<p>Observaciones: Incluir atajos de teclado para acceder al menú de ayuda.</p>	

2.1.2.1.9. Diseñar pruebas con validadores

La tabla 2.12 describe la historia de usuario 9 que consiste en implementar el diseño y ejecución de pruebas con validadores.

Tabla 2.12: Historia de usuario - Diseñar y ejecutar pruebas con validadores.HU09.

Historia de Usuario	
Número: 9.	Usuario: Usuario General.
Nombre historia: Diseñar y ejecutar pruebas con validadores.	
Prioridad en el negocio: Media.	Riesgo en desarrollo: Medio.
Iteración asignada: 3.	Responsable: Cristina Rivera.
<p>Descripción: El usuario accede a la aplicación accesible para interactuar con sus elementos. Por esta razón, la accesibilidad de la página debe ser evaluada y probada con validadores que indiquen el grado de implementación de pautas de accesibilidad en el prototipo.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Evaluar el grado de accesibilidad de la página con Examiner. 2. Evaluar el grado de accesibilidad con TAW. 	
<p>Observaciones: Se realizarán validaciones de accesibilidad y diseño.</p>	

2.1.2.1.10. Pruebas con simuladores

La tabla 2.13 describe la historia de usuario 10 que consiste en implementar el diseño y ejecución de pruebas con simuladores.

Tabla 2.13: Historia de usuario - Diseñar y ejecutar pruebas con simuladores.HU10.

Historia de Usuario	
Número: 10.	Usuario: Usuario General.
Nombre historia: Diseñar y ejecutar pruebas con simuladores.	
Prioridad en el negocio: Media.	Riesgo en desarrollo: Medio.
Iteración asignada: 3.	Responsable: Cristina Rivera.
<p>Descripción: El usuario accede a la aplicación accesible para interactuar con sus elementos. Por esta razón, la accesibilidad de la página debe ser evaluada y probada con validadores que indiquen el grado de implementación de pautas de accesibilidad en el prototipo.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Evaluar el uso de colores con Spectrum. 2. Evaluar el grado de accesibilidad con NoCoffee e Impairment Simulator. 	
<p>Observaciones: Se realizarán validaciones de uso de colores y usabilidad según el tipo de discapacidad visual.</p>	

2.1.2.1.11. Diseñar pruebas con usuarios

La tabla 2.14 describe la historia de usuario 11 que consiste en implementar el diseño y ejecución de pruebas con usuarios.

Tabla 2.14: Historia de usuario - Diseñar pruebas con usuarios.HU11.

Historia de Usuario	
Número: 11.	Usuario: Usuario General.
Nombre historia: Diseñar y ejecutar pruebas con usuarios.	
Prioridad en el negocio: Media.	Riesgo en desarrollo: Medio.
Iteración asignada: 3.	Responsable: Cristina Rivera.
<p>Descripción: Las pruebas de usabilidad del prototipo son realizadas en base a las funciones principales que se han implementado en el prototipo. Las pruebas se realizan a personas no videntes con la ayuda de un lector de pantalla.</p> <p>Descripción técnica:</p> <ol style="list-style-type: none"> 1. Acceder al menú ayuda, descripción e indicaciones. 2. Acceder a una oficina mediante el menú principal y mediante el plano. 3. Comprender la información que se despliega de los elementos. 4. Retornar a las secciones del prototipo mediante los botones de acceso rápido. 5. Utilizar atajos de teclado para acceder a diferentes secciones. 6. Utilizar el modo simulación para ubicar un elemento en el plano. 	
<p>Observaciones: Las pruebas se realizarán con personas no videntes que hayan interactuado con tecnologías como lectores de pantalla y navegadores web.</p>	

2.2. ANÁLISIS DEL PROTOTIPO.

En el análisis del prototipo se define el product backlog como la lista de funcionalidades del prototipo. Estas funcionalidades se agruparán en base a la prioridad de las historias de usuario clasificadas según la cuantificación y en base a la clasificación de requerimientos se establecerán los Sprints para el desarrollo del prototipo.

2.2.1. LISTA INICIAL DEL PRODUCT BACKLOG

En la siguiente tabla se han agrupado las historias de usuario en base al riesgo y prioridad. La primera historia con prioridad alta es el diseño de la arquitectura de la aplicación; en base a esta historia se establecerá el modelo principal sobre el cual se desarrollarán el resto de funcionalidades.

El resto de historias de usuario con prioridad alta son el diseño del plano interior accesible debido a que, sobre este, se define la mayoría de funcionalidad de la página y el diseño de la interfaz principal, que permitirá acceder al resto de módulos que se implementarán.

La navegabilidad tanto en el plano como en la página también posee una prioridad alta. Sus implementaciones se podrán utilizar en el prototipo mediante teclado, cumpliendo las especificaciones de accesibilidad para personas no videntes.

Los diseños del menú de accesibilidad y ayuda poseen una prioridad media a causa de que su desarrollo no afecta el desempeño del resto del prototipo. Estas funcionalidades son una agregación de accesibilidad para garantizar un uso universal.

La implementación del modo simulación implica una prioridad media. Este módulo es una agregación del prototipo que mejora la percepción de los elementos del plano y cuya ejecución no es obligatoria para conocer los otros elementos implementados.

Por último, se tiene las pruebas con prioridad media, debido a que las pruebas se realizan al final del desarrollo del prototipo para verificar el grado de concordancia de accesibilidad en base a los parámetros establecidos anteriormente. En la tabla 2.15 se tiene descrita la lista inicial del product backlog, en donde se describen número, id de la historia, el enunciado de la historia, el riesgo, prioridad e iteración a la que pertenece la historia.

Tabla 2.15: Product Backlog Inicial.

No	ID de la historia	Enunciado de la historia	Riesgo	Prioridad	Iteración
1	HU01	Definir arquitectura del prototipo.	Alto	Alto	1
2	HU02	Diseñar e implementar plano interior accesible	Alto	Alto	1
3	HU03	Diseño e implementar interfaz principal del sitio web.	Alto	Alto	1
4	HU04	Diseñar e implementar menú de accesibilidad	Alto	Media	1
5	HU05	Implementar modo simulación	Alto	Media	2
6	HU06	Implementar navegabilidad en el sitio web	Alto	Media	2
7	HU07	Implementar navegabilidad en el plano interior	Alto	Media	2
8	HU08	Diseñar e implementar menú de ayuda	Medio	Media	3
9	HU09	Diseñar e implementar pruebas con validadores	Medio	Media	3
10	HU10	Diseñar e implementar pruebas con simuladores	Medio	Media	3
11	HU101	Diseñar e implementar pruebas con usuarios	Medio	Media	3

2.2.2. DEFINICIÓN DE SPRINTS

Para definir los Sprints se ha tomado como fecha de inicio de proyecto el 30 de marzo de 2016 con una fecha de fin el 31 de mayo de 2016, dando como resultado un proyecto de duración de 63 días.

Para definir los Sprints, se han estimado el número de horas por tarea de cada historia de usuario, considerando un ritmo de trabajo de 4 horas diarias aproximadamente.

A continuación, se muestra una ilustración de la plantilla para el Product Release Plan elaborado para el proyecto, conjuntamente con la plantilla de Product Backlog con las historias de usuario definidas en el Product Backlog Inicial.

2.2.2.1. Product Release Plan

La Ilustración 2.1 muestra el Product release plan, que integra el nombre del equipo de trabajo, el número de horas de trabajo por Sprint y el número de días por cada Sprint con las fechas correspondientes.

Product Release Plan		Inicio del proyecto	30-Mar-2016	Fin del proyecto	31-May-2016	Número de días	63	Días restantes	-6	Daily SCRUM Meeting (minutes)	15 minutos
Sprint Plan											
Sprint	Inicio	Días	Fin	Número de horas	Promedio de horas por	Tamaño del equipo	Estado	Objetivo			
1	30-Mar-2016	27	25-Abr-2016	105	3,89	1	Done	Arquitectura y diseño de la aplicación.			
2	26-Abr-2016	22	17-May-2016	85	3,86	1	Done	Prototipo funcional.			
3	18-May-2016	14	31-May-2016	56	4,00	1	Done	Mejoras y pruebas finales.			
Equipo											
Nombre	Horas por Sprint	Horas por Sprint 2	Horas por Sprint 3								
Cristina Rivera	100	78	80	Equipo							
Tania Calle	3	5	4	Product Owner							
Sandra Sánchez	2	2	2	Scrum Master							

Ilustración 2.1: Product Release Plan.

2.2.2.2. Product Backlog

La Ilustración 2.2 muestra el listado del product backlog con las historias de usuario elaboradas en las tablas anteriores. Contiene el estado de la historia de usuario, el sprint al que pertenecen, la prioridad y la fecha de aprobación de la historia.

Product BackLog Plano interior accesible						
Inicio del proyecto		30-Mar-2016	Fin del proyecto		31-May-2016	
		Ongoing	Planned		High	
		Removed	Done		Low	
		Done			Medium	
ID de la Historia	Nombre de la historia	Estado	Sprint	Prioridad	Aprobación	Comentarios
S1	Historia 1 Diseñar arquitectura del prototipo.	Done	1	High	7/3/2016	
S2	Historia 2 Diseñar plano interior accesible	Done	1	High	7/3/2016	
S3	Historia 3 Diseño interfaz principal del sitio web.	Done	1	High	7/3/2016	
S4	Historia 4 Diseñar menú de accesibilidad	Done	1	Medium	7/3/2016	
S6	Historia 5 Implementar modo simulación	Done	2	Medium	14/3/2016	
S5	Historia 6 Implementar navegabilidad en el sitio web.	Done	2	Medium	14/3/2016	
S7	Historia 7 Implementar navegabilidad en el plano interior	Done	2	Medium	14/3/2016	
S8	Historia 8 Diseñar menú de ayuda	Done	3	Medium	14/3/2016	
S9	Historia 9 Diseñar e implementar pruebas con validadores	Done	3	Medium	27/3/2016	
S10	Historia 10 Diseñar e implementar pruebas con simuladores	Done	3	Medium	27/3/2016	
S11	Historia 11 Diseñar e implementar pruebas de usuarios	Done	3	Medium	27/3/2016	

Ilustración 2.2: Product Backlog.

2.2.2.3. Primer Sprint

En primer Sprint se conforma por las cuatro primeras tareas definidas en el Product Backlog Inicial. El inicio de proyecto corresponde al 30 de marzo y el fin de proyecto se estima para el 31 de mayo. Por lo cual este Sprint iniciará el 7 de marzo y finalizará el 25 de abril, tiene como duración 27 días. Las reuniones se las realiza diariamente a las 9 am con una duración máxima de 15 minutos.

El objetivo de este primer Sprint es desarrollar las tareas especificadas en la siguiente tabla, en donde se desglosan actividades relacionadas con el desarrollo de la arquitectura de la aplicación, diseño del plano interior e interfaz principal del prototipo y el diseño del menú de accesibilidad.

La tabla 2.16 muestra un desglose de las tareas por cada historia de usuario por tarea, en donde se contabiliza el número de horas por tarea y horas totales por historia de usuario.

Tabla 2.16: Tareas por historias de usuario del Primer sprint.

ID	Requerimiento	Tareas	Estimación(horas)
HU01	Definir arquitectura del prototipo	Definir modelo del ORM	4
		Definir vistas mediante layout.	4
		Definir controladores - parte 1.	4
		Definir controladores - parte 2.	4
		Establecer conexión con la base de datos.	4
		Generar script de coordenadas.	4
		Revisión del diseño de la arquitectura.	4
		Total	28
HU02	Diseñar plano interior accesible	Tomar medidas del entorno real.	3
		Modelar plano en herramienta de edición SVG a escala - parte 1.	4
		Modelar plano en herramienta de edición SVG a escala - parte 2.	3
		Generar documento SVG.	2
		Limpiar código SVG.	4
		Incluir atributos de accesibilidad.	4
		Incluir estilos con CSS.	4
		Incluir funciones con JavaScript.	4
		Total	28
HU03	Diseño interfaz principal del sitio web.	Definir estructura de la página con HTML.	4
		Incluir estilos en la página con CSS.	4
		Establecer bloques de información.	4
		Implementar menús.	4
		Incluir atributos de accesibilidad a etiquetas - parte 1.	4
		Incluir atributos de accesibilidad a etiquetas - parte 2.	4
		Implementar estilos CSS en base a pautas de accesibilidad.	4
		Retroalimentación y mejoras.	1
		Total	29

HU04	Diseñar menú de accesibilidad	Definir estructura del menú con HTML.	3
		Incluir estilos en el menú con CSS.	3
		Incluir botones para funciones de estilo y fuente.	4
		Establecer funciones para control de estilos en el plano.	4
		Establecer función para cambio de color en el plano.	3
		Revisión de diseño del prototipo	1
		Retrospectiva del Sprint	2
		Total	20

Total Primer Sprint: 105 horas

2.2.2.3.1. *Sprint Backlog*

En el sprint backlog se muestran las tareas desglosadas por historia de usuario y el número de horas estimadas por tarea. Además, se tiene el responsable por cada tarea y el número de horas de trabajo de cada miembro del equipo, lo que da como resultado un total de 105 horas trabajadas en el primer sprint.

En la ilustración 2.3 se muestra las tareas elaboradas en el primer sprint. La plantilla incluye es estado de la tarea, el número de horas y el responsable por tarea. Al final se obtiene un total del número de horas que cada miembro del equipo ha trabajado por Sprint.

Sprint 1					
	Nombre	Horas a trabajar	Horas trabajadas reales	Horas reales trabajadas por persona	
Fecha de inicio	Cristina Rivera	100	100	Horas de sprint que deberían realizarse	
30-Mar-2016	Tania Calle	3	3	Horas de sprint reales	
Fecha de fin	Sandra Sánchez	2	2	Horas restantes por Sprint	
25-Abr-2016	None	0	0	Horas trabajadas por equipo	
Días restantes por Sprint	None	0	0	Fecha	
0	None	0	0	Horas trabajadas por fecha	
ID Tarea	Descripción de la tarea	Estado	Horas estimadas	Responsable/ Horas totales trabajadas por equipo según fecha	Total de horas trabajadas por tarea
T1-H1-S1	Definir modelo del ORM	Done	4	Cristina Rivera	4
T2-H1-S1	Definir vistas mediante layout.	Done	4	Cristina Rivera	4
T3-H1-S1	Definir controladores.	Done	4	Cristina Rivera	4
T4-H1-S1	Definir controladores2.	Done	4	Cristina Rivera	4
T5-H1-S1	Establecer conexión con la base de datos.	Done	4	Cristina Rivera	4
T6-H1-S1	Generar script de coordenadas.	Done	4	Cristina Rivera	4
T7-H1-S1	Revisión del diseño de la arquitectura.	Done	1	Tania Calle	1
T7-H1-S1	Revisión del diseño de la arquitectura.	Done	3	Cristina Rivera	3
T8-H2-S1	Tomar medidas del entorno real.	Done	3	Cristina Rivera	3
T9-H2-S1	Modelar plano en herramienta de edición	Done	4	Cristina Rivera	4
T10-H2-S1	Modelar plano en herramienta de edición	Done	3	Cristina Rivera	3
T11-H2-S1	Generar documento SVG.	Done	2	Cristina Rivera	2
T12-H2-S1	Limpiar código SVG.	Done	4	Cristina Rivera	4
T13-H2-S1	Incluir atributos de accesibilidad.	Done	4	Cristina Rivera	5
T14-H2-S1	Incluir estilos con CSS.	Done	4	Cristina Rivera	1
T15-H2-S1	Incluir funciones con JavaScript.	Done	4	Cristina Rivera	4
T16-H3-S1	Definir estructura de la página con HTML.	Done	4	Cristina Rivera	4
T17-H3-S1	Incluir estilos en la página con CSS.	Done	4	Cristina Rivera	4
T18-H3-S1	Establecer bloques de información.	Done	4	Cristina Rivera	3
T19-H3-S1	Implementar menús.	Done	4	Cristina Rivera	3
T20-H3-S1	Incluir atributos de accesibilidad a etiquetas.	Done	4	Cristina Rivera	4
T21-H3-S1	Incluir atributos de accesibilidad a etiquetas.	Done	4	Cristina Rivera	4
T22-H3-S1	Implementar estilos CSS en base a pautas	Done	4	Cristina Rivera	3
T23-H3-S1	Retroalimentación y mejoras.	Done	1	Tania Calle	2
T24-H4-S1	Definir estructura del menú con HTML.	Done	3	Cristina Rivera	3
T25-H4-S1	Incluir estilos en el menú con CSS.	Done	3	Cristina Rivera	3
T26-H4-S1	Incluir botones para funciones de estilo y	Done	4	Cristina Rivera	4
T27-H4-S1	Establecer funciones para control de estilos	Done	4	Cristina Rivera	4
T28-H4-S1	Establecer función para cambio de color en	Done	3	Cristina Rivera	7
T29-H4-S1	Revisión de diseño del prototipo	Done	1	Tania Calle	1
T30-H4-S1	Retrospectiva del Sprint	Done	2	Sandra Sánchez	2

Ilustración 2.3: Sprint Backlog Template para el primer sprint.

2.2.2.3.2. Burn-down Chart

La Ilustración 2.4 se denomina gráfico de esfuerzo vs Número de horas o Burn-down Chart, muestra el esfuerzo realizado y el número de horas promedio trabajadas por día. El gráfico indica que el número de horas trabajadas a diario han sido constantes por lo que se obtiene un esfuerzo parejo, dando como resultado un gráfico uniforme.

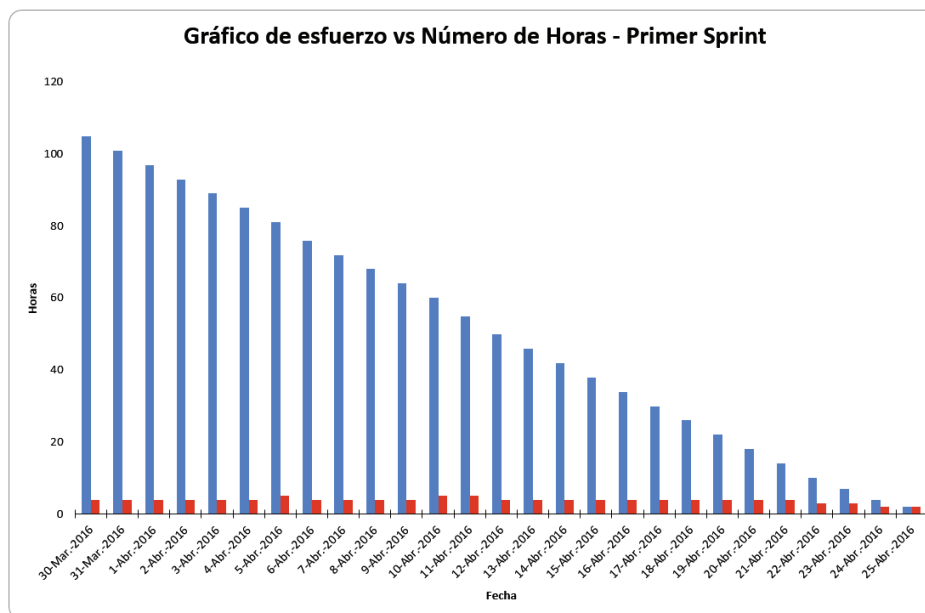


Ilustración 2.4: Gráfico de esfuerzo vs Número de Horas para el primer sprint.

2.2.2.4. Segundo Sprint

El segundo Sprint se conforma por las tareas 5,6 y 7 del product backlog.

Este Sprint inicia el 26 de abril y finaliza el 17 de mayo, tiene como duración 22 días, a un ritmo de trabajo de 4 horas diarias aproximadamente. Las reuniones se las realiza diariamente a las 9 am con una duración máxima de 15 minutos.

El objetivo del segundo sprint es implementar funcionalidades a los diseños elaborados anteriormente. Por lo cual se han desglosado las actividades de cada requerimiento como se muestra a continuación.

La tabla 2.17 muestra una estimación de horas por tarea implementada en cada historia de usuario, al final se contabiliza el número total de horas para completar la historia de usuario.

Tabla 2.17: Tareas por historias de usuario del Segundo sprint.

ID	Requerimiento	Tareas	Estimación(horas)
HU05	Implementar modo simulación	Definir rutas mediante coordenadas.	4
		Generar script de coordenadas.	4
		Implementar función para graficar rutas en SVG.	4
		Leer coordenadas desde ORM.	3
		Graficar coordenadas en el plano interior.	2
		Describir instrucciones de rutas.	2
		Grabar instrucciones y descripciones.	3
		Embeber audio y definir menú de rutas en la interfaz principal.	4
		Revisión de modo simulación.	1
		Total	27
HU06	Implementar navegabilidad en la página	Determinar elementos para navegación de la página.	4
		Definir orden de navegación.	4
		Incluir atributo "tabindex" en elementos de la página.	3
		Revisión de navegabilidad con tecla TAB	1
		Definir mediante funciones las teclas para navegación por cursores.	3
		Implementar funciones para navegación con cursores - parte 1.	4

		Implementar funciones para navegación con cursores - parte 2.	4
		Implementar atajos de teclado en elementos principales.	2
		Revisión de navegabilidad en la página.	2
		Total	27

HU07	Implementar navegabilidad en el plano interior	Determinar elementos para navegación del plano.	4
		Definir orden de navegación.	4
		Incluir atributo "tabindex" en elementos del plano.	3
		Revisión de navegabilidad con tecla TAB	1
		Definir mediante funciones las teclas para navegación por cursores.	4
		Implementar funciones para navegación con cursores - parte 1.	4
		Implementar funciones para navegación con cursores - parte 2.	4
		Implementar atajos de teclado en elementos principales.	3
		Revisión de funcionalidad del prototipo	1
		Retrospectiva del Segundo sprint	2
		Total	31

Total Segundo sprint: 85 horas

2.2.2.4.1. *Sprint Backlog*

En el segundo Sprint se tiene un total de 85 horas trabajadas en equipo. La Ilustración 2.5 muestra detalladamente las tareas que se han realizado en el segundo Sprint. Además, se aprecia estado de la tarea, el número de horas estimadas, el responsable por tarea y el número total de horas trabajadas por tarea.

Sprint 2		Nombre	Horas a trabajar	Horas trabajadas reales	Horas reales trabajadas por persona realizarse
Fecha de inicio	26-Abr-2016	Cristina Rivera	78	78	Horas de sprint reales
Fecha de fin	17-May-2016	Tania Calle	5	5	Horas restantes por Sprint
Días restantes por Sprint	0	Sandra Sánchez	2	2	Horas trabajadas por equipo
			0	0	Fecha
			0	0	Horas trabajadas por fecha

ID Tarea	Descripción de la tarea	Estado	Horas estimadas	Responsable/ Horas totales trabajadas por equipo según fecha	Total de horas trabajadas por tarea
T1-H5-S2	Definir rutas mediante coordenadas.	Done	4	Cristina Rivera	4
T2-H5-S2	Generar script de coordenadas.	Done	4	Cristina Rivera	4
T3-H5-S2	Implementar función para graficar rutas en	Done	4	Cristina Rivera	4
T4-H5-S2	Leer coordenadas desde OFIM.	Done	3	Cristina Rivera	3
T5-H5-S2	Graficar coordenadas en el plano interior.	Done	2	Cristina Rivera	2
T6-H5-S2	Describir instrucciones de rutas.	Done	2	Cristina Rivera	2
T7-H5-S2	Grabar instrucciones y descripciones.	Done	3	Cristina Rivera	3
T8-H5-S2	Embeber audio y definir menú de rutas en la	Done	4	Cristina Rivera	4
T9-H5-S2	Revisión de modo simulación.	Done	1	Tania Calle	1
T10-H6-S2	Determinar elementos para navegación de la	Done	4	Cristina Rivera	4
T11-H6-S2	Definir orden de navegación.	Done	4	Cristina Rivera	4
T12-H6-S2	Incluir atributo tabindex en elementos de la	Done	3	Cristina Rivera	3
T13-H6-S2	Revisión de navegabilidad con tecla TAB	Done	1	Tania Calle	1
T14-H6-S2	Definir mediante funciones las teclas para n	Done	3	Cristina Rivera	3
T15-H6-S2	Implementar funciones para navegación con	Done	4	Cristina Rivera	4
T16-H6-S2	Implementar funciones para navegación con	Done	4	Cristina Rivera	4
T17-H6-S2	Implementar atajos de teclado en elementos	Done	3	Cristina Rivera	2
T18-H6-S2	Revisión de navegabilidad en la página.	Done	1	Tania Calle	2
T19-H7-S2	Determinar elementos para navegación del p	Done	4	Cristina Rivera	4
T20-H7-S2	Definir orden de navegación.	Done	4	Cristina Rivera	4
T21-H7-S2	Incluir atributo tabindex en elementos del pl	Done	3	Cristina Rivera	3
T22-H7-S2	Revisión de navegabilidad con tecla TAB	Done	1	Tania Calle	1
T23-H7-S2	Definir mediante funciones las teclas para n	Done	4	Cristina Rivera	4
T24-H7-S2	Implementar funciones para navegación con	Done	4	Cristina Rivera	4
T25-H7-S2	Implementar funciones para navegación con	Done	4	Cristina Rivera	4
T26-H7-S2	Implementar atajos de teclado en elementos	Done	4	Cristina Rivera	3
T27-H7-S2	Revisión de funcionalidad del prototipo	Done	1	Tania Calle	2
T28-H7-S2	Retrospectiva del Sprint2	Done	2	Sandra Sánchez	2

Ilustración 2.5: Sprint backlog template para el segundo sprint.

2.2.2.4.2. Burn-down Chart

La Ilustración 2.6 se denomina gráfico de esfuerzo vs Número de horas o Burn-down Chart, muestra el esfuerzo realizado y el número de horas promedio trabajadas por día. El número de horas trabajadas fue constante por lo que se obtuvo un gráfico equilibrado, en los últimos días se observa un trabajo de pocas horas debido a la finalización de las tareas.

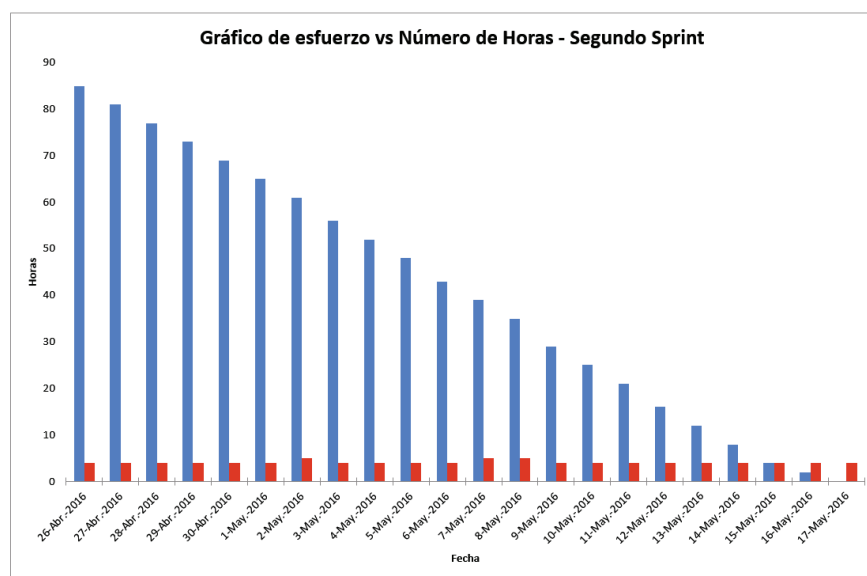


Ilustración 2.6: Gráfico de esfuerzo vs Número de horas para el Segundo sprint.

2.2.2.5. Tercer Sprint

El tercer sprint se conforma por las tareas 8,9,10 y 11 del product backlog.

Este Sprint iniciará el 18 de mayo y finalizará el 31 de mayo, tiene como duración 14 días a un ritmo de trabajo de 4 horas diarias aproximadamente. Las reuniones se las realiza diariamente a las 9 am con una duración máxima de 15 minutos.

El objetivo tercer Sprint es desarrollar pruebas sobre el prototipo y obtener retroalimentaciones para implementar mejoras. A continuación, se muestra las tareas por historia de usuario.

En la Ilustración 2.18 se pueden ver las tareas que se realizan por cada historia de usuario del tercer Sprint. Cada tarea tiene un estimado de duración en horas y por cada historia de usuario se contabiliza el número de horas totales.

Tabla 2.18: Tareas por historias de usuario del Tercer sprint.

ID	Requerimiento	Tareas	Estimación(horas)
HU08		Definir estructura del menú con HTML.	3
		Incluir menú y botones para opciones de ayuda.	4
		Implementar atributos de accesibilidad.	4
		Incluir estilos en el menú con CSS.	3
		Incluir instrucciones y descripciones.	2
		Total	16
HU09	Diseñar y ejecutar pruebas con validadores.	Realizar pruebas con eXaminator.	2
		Realizar pruebas con TAW	2
		Realizar pruebas con CSS-validator	2
		Tabular resultados de las pruebas.	3
		Retroalimentación y resultados.	1
		Total	10
HU10	Diseñar y ejecutar pruebas con simuladores	Realizar pruebas con Spectrum.	2
		Realizar pruebas con Coblis.	2
		Realizar pruebas con Visual Impairment Simulator	3
		Realizar pruebas con Disability Impairment Approximation Simulator (DIAS).	3
		Tabular resultados de las pruebas.	3

	Retroalimentación y resultados	1
	Total	14

HU11	Diseñar y ejecutar pruebas con usuarios	Realizar pruebas con usuario 1	4
		Retroalimentación y resultados	2
		Realizar pruebas con usuario 2	4
		Retroalimentación y resultados	2
		Análisis de resultados de pruebas.	2
		Retrospectiva del Tercer sprint	2
		Total	16

Total Tercer sprint: 56 horas

2.2.2.5.1. Sprint Backlog

En la Ilustración 2.7 se puede apreciar que en el segundo Sprint se tiene un total de 50 horas trabajadas en equipo. Además, se puede evidenciar todas las tareas realizadas por Sprint son sus respectivos responsables.

Sprint 3		Nombre	Horas a trabajar	Horas trabajadas reales	Horas reales trabajadas por persona
Fecha de inicio	18_May_2016	Cristina Rivera	50	50	realizarse
Fecha de fin	31_May_2016	Tania Calle	4	4	Horas de sprint reales
Días restantes por Sprint	0	Sandra Sánchez	2	2	Horas restantes por Sprint
			0	0	Horas trabajadas por equipo
			0	0	Fecha
			0	0	Horas trabajadas por fecha
ID Tarea	Descripción de la tarea	Estado	Horas estimadas	Responsable/ Horas totales trabajadas por equipo según fecha	Total de horas trabajadas por tarea
T1-H8-S3	Definir estructura del menú con HTML.	Done		3 Cristina Rivera	3
T2-H8-S3	Incluir menú y botones para opciones de ayuda.	Done		4 Cristina Rivera	4
T3-H8-S3	Implementar atributos de accesibilidad.	Done		4 Cristina Rivera	4
T4-H8-S3	Incluir estilos en el menú con CSS.	Done		3 Cristina Rivera	3
T5-H8-S3	Incluir instrucciones y descripciones.	Done		2 Cristina Rivera	2
T6-H9-S3	Realizar pruebas con eXaminator.	Done		3 Cristina Rivera	3
T7-H9-S3	Realizar pruebas con TAW	Done		3 Cristina Rivera	3
T9-H9-S3	Tabular resultados de las pruebas con validadores.	Done		3 Cristina Rivera	3
T10-H9-S3	Retroalimentación y resultados con validadores.	Done		1 Tania Calle	1
T11-H10-S3	Realizar pruebas con Spectrum.	Done		2 Cristina Rivera	2
T12-H10-S3	Realizar pruebas con Coblis.	Done		2 Cristina Rivera	2
T13-H10-S3	Realizar pruebas con Visual Impairment Simulator	Done		3 Cristina Rivera	3
T14-H10-S3	Realizar pruebas con Disability Impairment Approximation	Done		3 Cristina Rivera	3
T15-H10-S3	Tabular resultados de las pruebas con simuladores.	Done		3 Cristina Rivera	3
T16-H10-S3	Retroalimentación y resultados con simuladores.	Done		1 Tania Calle	1
T17-H11-S3	Realizar pruebas con usuario 1	Done		4 Cristina Rivera	4
T18-H11-S3	Retroalimentación y resultados	Done		2 Cristina Rivera	2
T19-H11-S3	Realizar pruebas con usuario 2	Done		4 Cristina Rivera	4
T20-H11-S3	Retroalimentación y resultados	Done		2 Cristina Rivera	2
T21-H11-S3	Análisis de resultados de pruebas.	Done		2 Tania Calle	2
T22-H7-S2	Retrospectiva del Sprint3	Done		2 Sandra Sánchez	2

Ilustración 2.7: Sprint backlog template para Tercer sprint.

2.2.2.5.2. Burn-down Chart

La Ilustración 2.8 se denomina gráfico de esfuerzo vs Número de horas o Burn-down Chart, muestra el esfuerzo realizado y el número de horas promedio

trabajadas por día. En el gráfico se aprecia que el número de horas trabajadas fue constante, por lo que las tareas terminaron un día antes de la planificación.

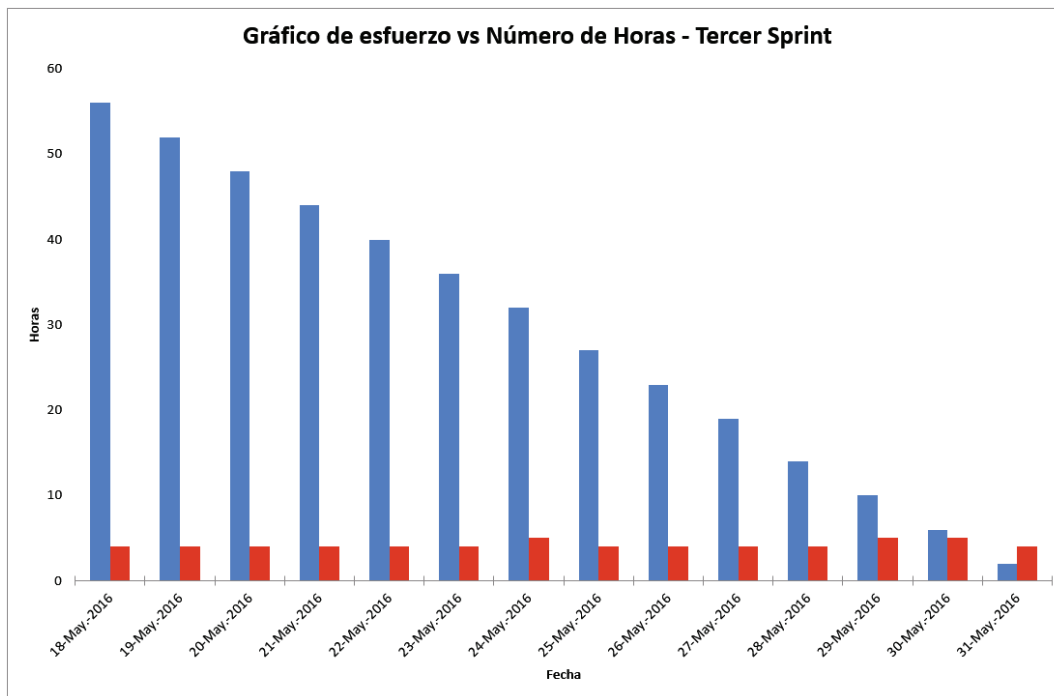


Ilustración 2.8: Gráfico de esfuerzo vs Número de horas para el Tercer sprint.

2.3. DISEÑO DEL PROTOTIPO

2.3.1. DISEÑO DEL PRIMER SPRINT

El objetivo principal del primer Sprint es obtener la arquitectura y el diseño de la aplicación.

La arquitectura de la aplicación se ha planteado tomando en cuenta el patrón Modelo Vista Controlador.

El diseño del plano como de la página se elaboran tomando en cuenta las pautas de accesibilidad que indican como debe ser el diseño de las interfaces para ser utilizado por personas con discapacidad visual severa como la ceguera.

Las historias de usuario que se elaboran en el primer Sprint son:

1. Definir arquitectura del prototipo.
2. Diseñar e implementar plano interior accesible
3. Diseño e implementar interfaz principal del sitio web.
4. Diseñar e implementar menú de accesibilidad

2.3.1.1. Definición de la arquitectura del prototipo

El esquema utilizado para desarrollar el prototipo del plano interior accesible es Modelo-Vista-Controlador(MVC), esto surge ante la necesidad de crear una aplicación web robusta y escalable. Este esquema de programación basado en un modelo de capas, se ha utilizado para separar la lógica del negocio, lógica de usuario y lógica de datos, proporcionando a la vez un acoplamiento entre estos elementos [29].

El modelo corresponde a la capa que figura la información con la cual opera la aplicación. El modelo se encarga de gestionar las operaciones sobre dicha información y actualizar la información que se presenta en la Vista. Las peticiones de información que se solicitan al modelo, se las realizan por medio del controlador. En este caso práctico, se ha utilizado un Object relational mapping (ORM) como motor de persistencia. Dentro de este ORM se han almacenado coordenadas tomadas de un entorno real, estas coordenadas simulan la ruta que sería utilizada para ubicar un elemento del plano en un ambiente práctico.

La vista es la capa que contiene el código que será mostrado en forma de interfaces al usuario. La vista se renderiza en base a la información que proporciona el modelo. En el caso del prototipo, la vista alberga código Scalable Vector Graphics (SVG) con el cual se contruyó el plano interior a escala. SVG interactúa directamente con HTML y CSS para su presentación, y JQUERY para manejar los eventos que se han implementado en el plano. El plano en formato HTML se complementa con código JavaScript para construir la lógica detrás de la aplicación web y definir el comportamiento que tiene el plano.

El controlador corresponde a la capa responsable de responder y verificar las peticiones proporcionadas por el usuario, realizar llamadas sobre el Modelo, seleccionar los datos que serán presentados en la capa de Vista. Dentro del controlador se tienen funciones, procesamiento de datos y respuesta a eventos.

Finalmente se presenta la aplicación por medio de la vista al usuario, que es el encargado de elaborar solicitudes por medio de la vista y recibir la información mediante la capa de vista. Por medio del cliente se puede analizar la aplicación en validadores de accesibilidad como son Examiner y TAW por medio de una URL.

Además se puede integrar a un lector de pantalla como ChromeVox, Narrator o NDVA para transformar el contenido textual a un formato auditivo. En la Ilustración 2.9 se muestra una esquematización del modelo de arquitectura propuesto para el desarrollo del prototipo.

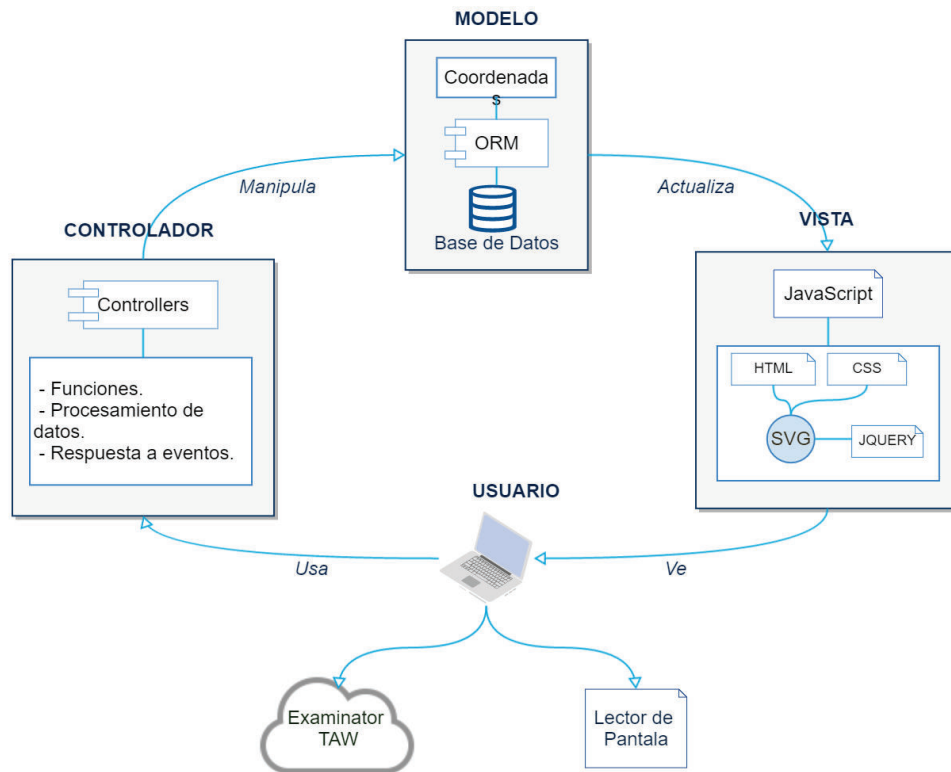


Ilustración 2.9: Modelo de arquitectura usada en el prototipo.

2.3.1.1.1. Diagrama conceptual de datos

Las tablas que existen en el modelo conceptual de datos son Rutas y Coordenadas. La tabla de Coordenadas contiene la posición tanto en X como en Y, y hace referencia a idRutas como clave foránea de la tabla rutas. La tabla rutas contiene el nombre para la identificación de cada ruta elegible en el prototipo. La relación de las tablas es de uno a muchos. En la Ilustración 2.10 se muestra un esquema en donde se define el diagrama conceptual de datos.

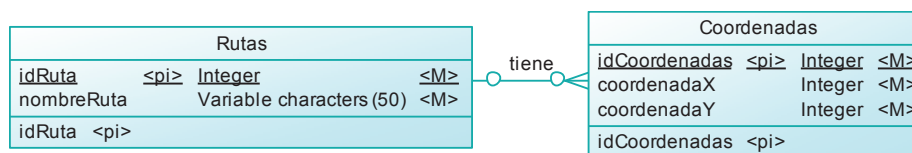


Ilustración 2.10: Diagrama conceptual de datos.

2.3.1.1.2. Diagrama físico de datos

En el modelo físico al igual que el conceptual se tiene las tablas rutas y coordenadas con sus respectivas claves foráneas y claves principales. En la ilustración 2.11 se muestra la esquematización del diagrama físico de datos.

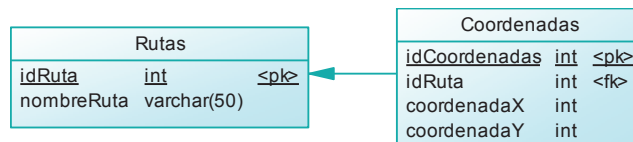


Ilustración 2.11: Diagrama físico de datos.

2.3.1.1.3. Definición de vistas

Layout: Es la vista principal que alberga el resto de vistas. Contiene los diseños principales como cabecera, que alberga e menú de accesibilidad y de ayuda, y el pie de página. La ilustración 2.12 muestra la captura de pantalla de la vista Layout.

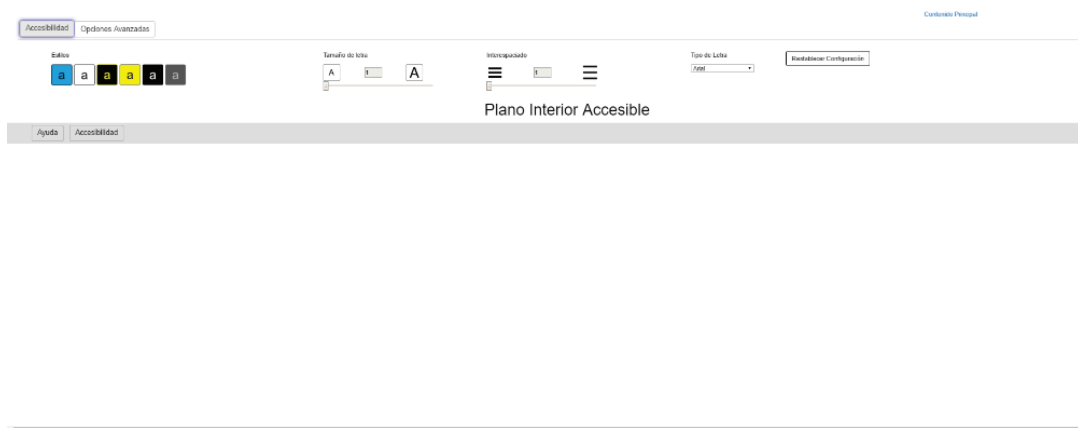


Ilustración 2.12: Captura de Vista Layout.

InformaciónMapa: Es la vista que se encarga de mostrar las descripciones de los elementos del plano. La información que se muestra es el nombre del elemento, información como carga horaria y descripciones, y los elementos aledaños en el plano. La ilustración 2.13 muestra la captura de la vista informaciónMapa con la descripción de la oficina subdecanato.

Información Principal:
Subdecanato.

Materias e información:
Es la segunda autoridad ejecutiva de la Facultad. Sus funciones son, Colaborar con el Decano en el cumplimiento de sus funciones y en el ejercicio de sus atribuciones, Coordinar la revisión curricular de las carreras de la Facultad La subdecana es la Dra. Jenny Torres.

Información del elemento:
Arriba: Menú accesibilidad.
Abajo: Baño del rectorado.
Derecha: Sala de reuniones.
Izquierda: Botón oficinas de docentes.

Volver al mapa

Volver al menú principal

Inicio

Ilustración 2.13: Captura de Vista informacionMapa.

mapaPlano: Esta vista corresponde al plano original con todos sus elementos. Esta vista llama al plano elaborado en SVG y embebido en un archivo HTML, es la vista principal del plano que contiene el prototipo. En la Ilustración 2.14 se muestra la captura del plano interior correspondiente a la vista mapaPlano.



Ilustración 2.14: Captura de Vista mapaPlano.

mapaPlanocolor: Esta vista, al igual que el plano original contiene al plano cuyos colores se han elaborado con patrones, esto sirve para distinguir las regiones del mapa a personas con alguna discapacidad al color. En la Ilustración 2.16 se muestra las vistas mapaPlanocolor con los patrones y colores escogidos para diferenciar las regiones del plano.

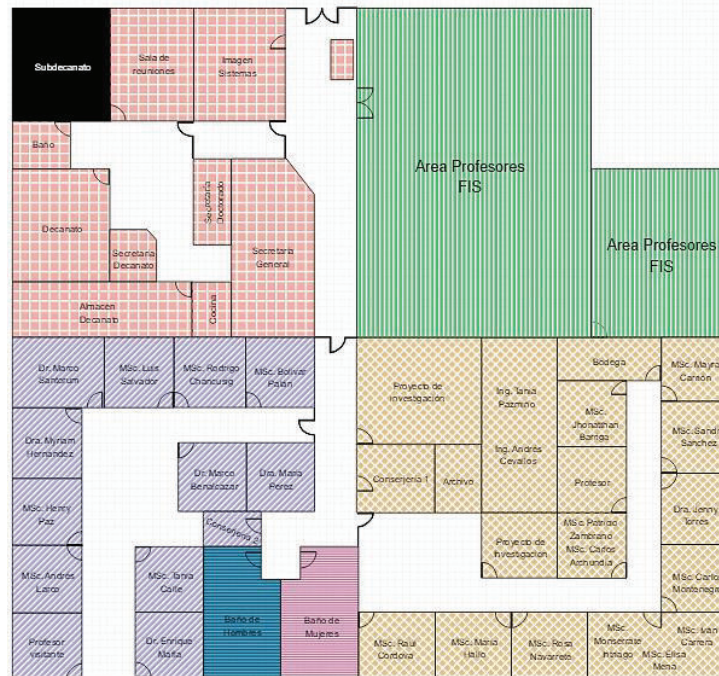


Ilustración 2.15: Captura de Vista mapaPlanoColor.

mapaPlanonegro: Esta vista corresponde al diseño del plano creado en patrón a blanco y negro, este diseño sirve para que las personas con ceguera al color para que puedan distinguir las regiones del plano. En la ilustración 2.16 se muestra la captura de la vista mapaPlanonegro con patrones a blanco y negro para distinguir las regiones principales del plano.



Ilustración 2.16: Captura de Vista mapaPlanoNegro.

menuPlano: Esta vista corresponde al menú de elementos del plano. Además, contiene un listado de las rutas que pueden ser graficadas en el modo simulación. En la ilustración 2.17 se aprecia el menú de oficinas de docentes ordenado alfabéticamente y el menú simulación de rutas.



Ilustración 2.17: Captura de vista menuPlano.

Plano: Esta vista corresponde a la parte central del prototipo. Alberga las vistas de menú, plano e información. La ilustración 2.18 muestra la captura de la vista Plano que contiene a las vistas descritas anteriormente.



Ilustración 2.18: Captura de vista Plano.

2.3.1.1.4. Controladores

1. **MapaController:**

- 1.1. Index:** Este es el primer controlador que se llama al iniciar la petición cliente servidor; se encarga de inicializar las cookies principales y redireccionar a la URL /Mapa/plano.
- 1.2. Plano:** Este controlador se encarga de la carga de la vista del plano verificando previamente si no existe una configuración previamente establecida por medio de cookies.
- 1.3. informacionOficina:** Este controlador se encarga de cargar la información correspondiente a cada elemento del plano seleccionado y cargándolo en la vista informacionMapa.
- 1.4. tipoPlano:** Tiene la función de renderizar el plano dependiendo de la selección de estilo, es decir, que carga el plano original, o el plano con patrón y color o el plano con patrón a blanco y negro.
- 1.5. Coordenadas:** Es el controlador encargado de la lectura de coordenadas desde el ORM y mandarlo a la vista correspondiente para su graficación en SVG.
- 1.6. cookieStart:** Inicializa las cookies por defecto o lee las cookies almacenadas anteriormente en el cliente.
- 1.7. cookieRead:** Este controlador se encarga de la lectura de cookies almacenadas en el cliente para ser aplicadas a las distintas vistas del prototipo.
- 1.8. cookieMapa:** Este controlador carga la cookie encargada de renderizar el plano escogido.

2.3.1.1.5. Modelos

En la ilustración 2.19 se puede apreciar el esquema del modelo final para su implementación en el framework Sails.js.



Ilustración 2.19: Modelo implementado en el framework Sails.

Partiendo desde el modelo físico de datos, se traslada al framework Sails.js para ser parte del ORM y con esto poderlo utilizar dentro de los controladores descritos anteriormente.

2.3.1.1.6. *Script de coordenadas*

El Script de coordenadas se ha generado desde una base de datos con coordenadas simuladas en un ambiente real. El script de coordenadas se ha generado en formato JSON para que sea compatible con el ORM del framework Sails.js. En la ilustración 2.20 se puede mostrar el script de rutas con sus coordenadas en X y Y, y el identificador de la ruta a la que pertenecen las coordenadas,

```

6 // Database 'posicion'
7
8 // posicion.coordenadas
9
10 [{"idcoordenadas":"1","coordenadax":"465","coordenaday":"100","idruta":"1"},
{"idcoordenadas":"2","coordenadax":"465","coordenaday":"125","idruta":"1"},
{"idcoordenadas":"3","coordenadax":"465","coordenaday":"150","idruta":"1"},
{"idcoordenadas":"4","coordenadax":"465","coordenaday":"175","idruta":"1"},
{"idcoordenadas":"5","coordenadax":"465","coordenaday":"200","idruta":"1"},
{"idcoordenadas":"6","coordenadax":"465","coordenaday":"225","idruta":"1"},
{"idcoordenadas":"7","coordenadax":"440","coordenaday":"235","idruta":"1"},
{"idcoordenadas":"8","coordenadax":"415","coordenaday":"235","idruta":"1"},
{"idcoordenadas":"9","coordenadax":"390","coordenaday":"235","idruta":"1"},
{"idcoordenadas":"10","coordenadax":"365","coordenaday":"235","idruta":"1"},
{"idcoordenadas":"11","coordenadax":"340","coordenaday":"235","idruta":"1"},

```

Ilustración 2.20: Script de coordenadas en formato JSON.

2.3.1.2. **Diseñar e implementar plano interior accesible**

2.3.1.2.1. *Medidas y escala*

Para diseñar el plano interior correspondiente al segundo piso de la facultad de sistemas, se debe establecer la distribución espacial de los elementos del piso, tal como oficinas, salas de reuniones, baños, bodegas, etc.

Posteriormente se graficará dichos elementos en base a las medidas aproximadas que se han tomado del piso como se muestra a continuación.

En la imagen 2.21 se aprecia la representación del plano elaborada a escala y con medidas expresadas en metros.

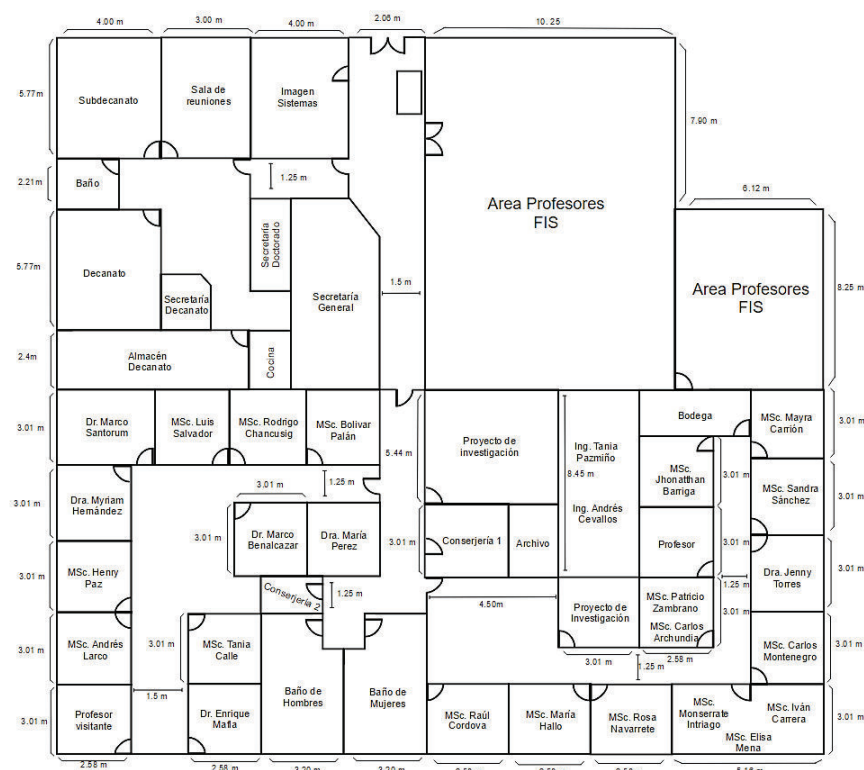


Ilustración 2.21: Representación de plano a escala del segundo piso de la facultad de sistemas.

2.3.1.2.2. Modelamiento del plano

Para dibujar el plano, se debe tomar como base, las medidas tomadas en el ambiente real. Para ello, es necesario establecer una escala métrica. En este caso se ha hecho una equivalencia al número de pixeles utilizados al graficar el plano en la herramienta; la escala utilizada equivale a 30 pixeles por metro real. La herramienta a utilizar es Vector Paint, es una herramienta online que permite exportar imágenes en formato SVG, de tal manera que el código es generado sin dependencias para manipular directamente sus etiquetas y atributos [30].

2.3.1.2.3. Limpieza y atributos de accesibilidad en el código SVG generado.

El código SVG que genera la herramienta, debe ser revisado para establecer una estructura ordenada, de tal manera que permita agregar atributos de accesibilidad sin inconvenientes.

Además, se deben eliminar las etiquetas y atributos innecesarios para facilitar el acoplamiento de estilos y funciones.

Los atributos de accesibilidad incluidos en el código SVG se basan en el estándar WAI-ARIA, que establece una forma de elaborar contenido web accesible para

personas discapacitadas, se enfoca en tecnologías de contenido dinámico e interfaces de usuario como JavaScript, HTML y tecnologías semejantes [31]. Además, se agregaron otros atributos de accesibilidad como “tabindex”, para definir un orden de navegación secuencial entre los elementos del plano. “title” para asociar un nombre al elemento definido y addinfo para incluir información extra sobre el elemento. En la ilustración 2.22 se puede apreciar los atributos WAI-ARIA en un fragmento de código HTML del prototipo.

```

1 <svg width="1058.823533371658" height="992.9411822784323" aria-labelledby="text" aria-
describedby="desc" style="display: inline; width: inherit; min-width: inherit; max-width:
inherit; height: inherit; min-height: inherit; max-height: inherit;" id="plano" fill="none"
class="svg">
2   <defs>
3     <pattern id="grid20" width="10" height="10" patternUnits="userSpaceOnUse">
4       <path d="M 100 0 L 0 0 0 100" stroke="#e0e0e0" stroke-width="1" />
5     </pattern>
6   </defs>
7
8   <rect fill="url(#grid20)" width="1058.823533371658" height="992.9411822784323" y="0"></rect>
9
10  <title>Mapa accesible facultad de ingenieria de sistemas</title>
11
12  <g title="profesor" class="currentLayer">
13
14
15    <g id="svg_2">
16      <a onclick="cargarInformacionOficina('99')">
17        <title>Bodega</title>
18
19        <path title="Bodega" class="<%= idOficina == '99' ? 'svgSeleccionado' :
'oficinasA' %>" d="M912.68,491 C912.68,491 912.67,548.35 912.67,548.35 C912.67,548.35
774.69,548.37 774.69,548.37 C774.69,548.37 774.74,491.02 774.74,491.02 C774.74,491.02 912.68,491
912.68,491 z" addinfo="Bodega2" stroke="#000000" stroke-opacity="1" id="Bodega2" tabindex="99" />
20
21      </a>
22    </g>

```

Ilustración 2.22: Atributos WAI-ARIA en código SVG del plano interior accesible.

2.3.1.2.4. Generación de estilos y funciones

Los estilos se agregaron al plano utilizando una hoja de estilos externa. Los colores utilizados en el plano se basaron en tonos escogidos mediante las regiones que contiene el plano, es decir, decanato, área de profesores y las dos áreas de oficinas, de tal manera que puedan ser distinguibles y no alteren la visualización del resto de elementos.

Los colores designados para elementos seleccionados y en los que se ubica el foco, se escogen de acuerdo a la distinción con el resto de elementos. El color negro se ha escogido para los elementos seleccionados ya que puede ser percibido con cualquier persona, con o sin discapacidad al color, mientras que el color designado para el foco se ha escogido tonos más brillantes y un borde azul que no interfiere con la identificación del resto de elementos. La ilustración 2.23

muestra la selección de la oficina de Área de Profesores mediante una diferencia de color en el fondo y borde del elemento.

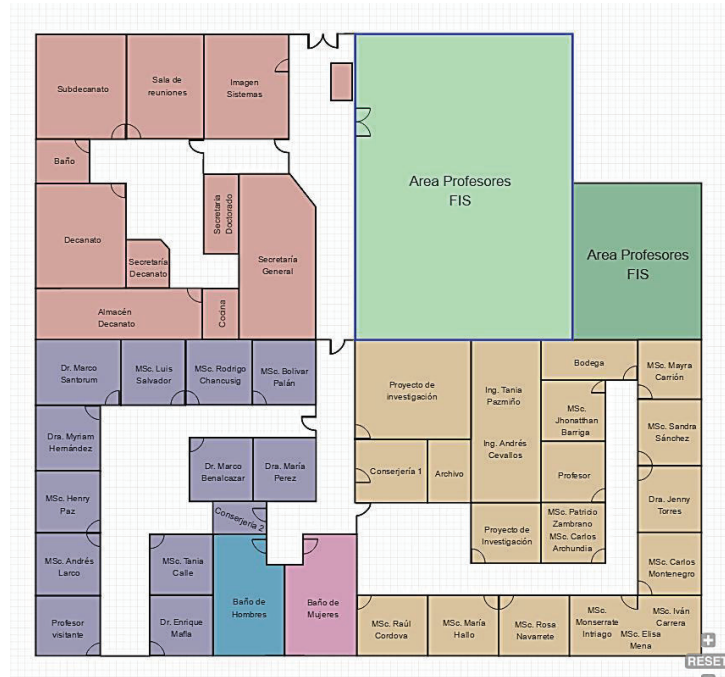


Ilustración 2.23: Estilos en plano modelado en SVG.

Las funciones agregadas en las etiquetas del documento SVG, consisten en funciones que llaman mediante un identificador único a la información cargada en el controlador correspondiente para cargarlo en la vista de Información.

Para implementar una barra de zoom en el plano, se ha utilizado una librería en JavaScript denominada `svg.pan-zoom.js`. La librería sirve para centrar, incrementar y disminuir el tamaño en elementos SVG [32].

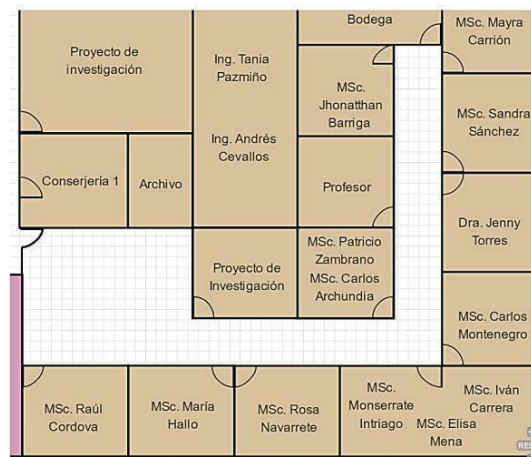


Ilustración 2.24: Demostración de `svg.pan-zoom.js` añadido al plano.

La ilustración 2.24 muestra una captura de la herramienta `svg.pan-zoom.js` al realizar zoom sobre el plano elaborado en SVG.

2.3.1.3. Diseñar e implementar interfaz principal del sitio web

2.3.1.3.1. Definición de estructura de la página con HTML y CSS.

Para definir la estructura HTML y CSS en la interfaz principal del prototipo, se ha utilizado Bootstrap como herramienta de maquetación.

Bootstrap es un framework que permite implementar sitios web con JavaScript y CSS. Los sitios web desarrollados con Bootstrap se adaptan a la resolución del dispositivo en donde se despliegan, son de diseño simple e intuitivo.

La estructura principal de la página que alberga el menú de oficinas, el plano y el bloque de información se elaboran mediante una estructura Grid de columnas para alinear ordenadamente los elementos de la página.

La Ilustración 2.25 muestra una captura de la vista principal del prototipo con las vistas parciales que la componen.

```

1 <link rel="stylesheet" href="/styles/css/mapas/main/plano2.css">
2
3 <script src="/js/dependencias/Jquery/jquery-1.11.3.js"></script>
4 <script src="/js/mapa/svgPanZoom.js"></script>
5 <script src="/js/mapa/svgbarra.js"></script>
6 <script src="/js/dependencias/Jquery/d3.js"></script>
7 <script src="/js/dependencias/Jquery/d3.min.js"></script>
8
9
10 <div class="col-md-3 offset3 span6 colordescrpcionmapa">
11   <%- partial ('../Mapa/menuPlano.ejs') %>
12 </div>
13
14 <div class="col-md-6 estilomapa divtamaniomap">
15   <div id="contenedorPlanoOficinas" class="tamaniomapa container-fluid">
16     <%- partial ('../Mapa/'+mapacookie+'.ejs', {idOficina: idOficina}) %>
17   </div>
18 </div>
19
20
21 <div class="col-md-3 offset3 span6 colordescrpcionmapa">
22   <%- partial ('../Mapa/informacionMapa.ejs') %>
23 </div>

```

Ilustración 2.25: Esquema principal del prototipo con vistas parciales.

Para establecer un diseño intuitivo, se estableció que tanto el menú de accesibilidad como de ayuda se ubique en la parte superior; el menú de oficinas y rutas de simulación de ubica al lado izquierdo, cuyos elementos se despliegan en un menú collapse. La parte central corresponde al plano elaborado en SVG y el bloque derecho se carga dinámicamente con la información de cualquier elemento del plano que sea seleccionado. La Ilustración 2.26 muestra el diseño de la interfaz principal prototipo.

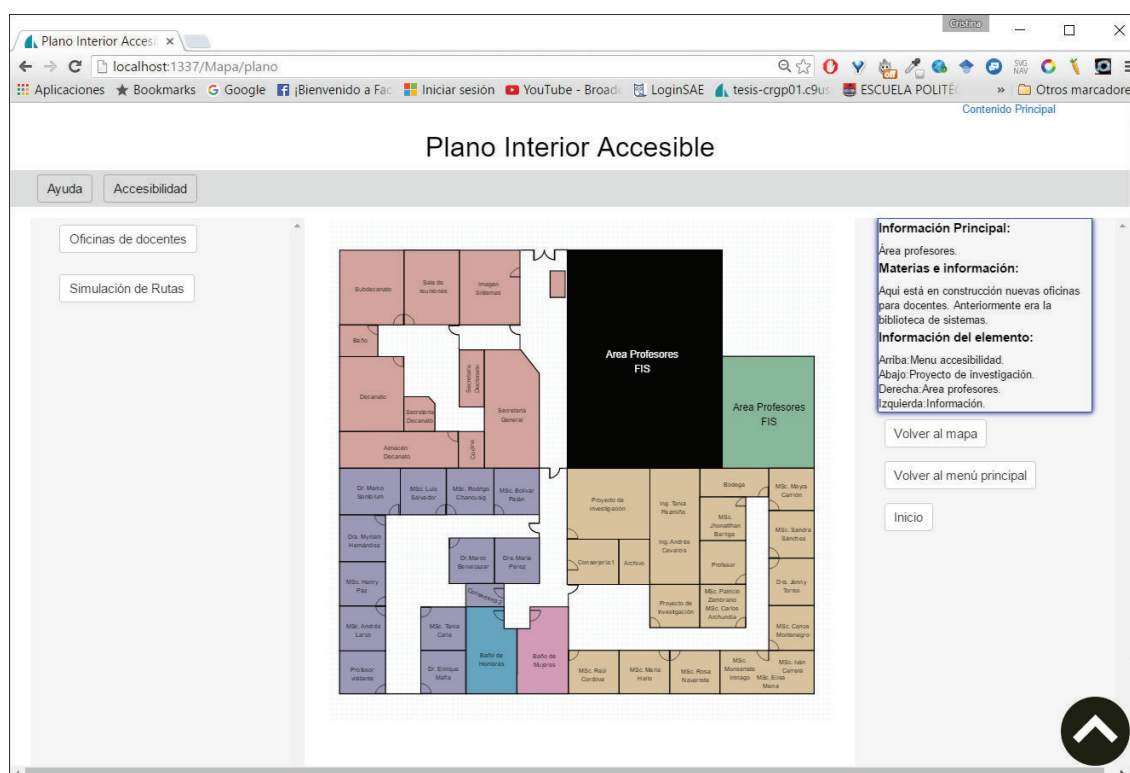


Ilustración 2.26: Diseño de interfaz principal del prototipo web.

Los atributos de accesibilidad agregados son “title” para desplegar un tooltip de información sobre el elemento seleccionado, alt en imágenes para describir al elemento cuando se utiliza un lector de pantalla, respeta la implementación de estilos de manera independiente del código y el uso de enlaces externos,

2.3.1.3.2. Bloques de información e implementación menús.

Los menús se encuentran en la vista menuPlano, y se han creado mediante botones de tipo collapse de Bootstrap como se muestra en la Ilustración 2.27, que puede ser desplegado o contraído para no interferir con la navegación en el prototipo. Los elementos del menú contienen atributos como “alt”, “title” y “tabindex” para agregar información adicional sobre el elemento y permitir una navegación secuencial.

Además, se han implementado funciones para desplegar información del elemento y seleccionar la ubicación del elemento en el plano, de tal manera que se tiene dos maneras de acceder a un elemento, mediante el menú o el plano.

```

1 <div class="container">
2   <div>
3     <div>
4       <p>
5         <button data-toggle="collapse" class="btn btn-default elementomenu subtítulos "
          alt="Oficinas de docentes" title="Oficinas de docentes" id="ofidocentes" aria-expanded="false"
          tabindex="15" href="#collapse1">Oficinas de docentes</button>
6         <!-- <h3 class="text-center">Oficinas de Docentes</h3-->
7       </p>
8     </div>
9     <div title="" id="collapse1" class="panel-collapse collapse">
10      <ul class="nav nav-tabs">
11        <li class="active">
12          <a data-toggle="tab" href="#ac" title="a-c" id="taba-c">
13            <input class="alfabeto" type="image" src="../../images/a-f.png" tabindex="17"
              title="A-C" id="a-c" alt="rango a-c">
14          </a>
15        </li>
16        <li>
17          <a data-toggle="tab" href="#dl" title="d-l" id="tabd-l">
18            <input class="alfabeto" type="image" src="../../images/g-l.png" tabindex="17"
              title="D-L" id="d-l" alt="rango d-l">
19          </a>
20        </li>
21        <li>
22          <a data-toggle="tab" href="#menu2" title="m-p" id="tabm-p">
23            <input class="alfabeto" type="image" src="../../images/m-s.png" tabindex="17"
              title="M-P" id="m-p" alt="rango m-p">
24          </a>
25        </li>

```

Ilustración 2.27: Menú collapse para menú Oficinas de Docentes.

El bloque de información corresponde a la vista informacionMapa. Se ha estructurado de tal manera que permita la carga dinámica de la información proveniente del controlador informacionOficina. Adicionalmente se agregan tres botones como ayudas de navegación, volver al mapa que redirige el foco al primer elemento del plano, volver al menú principal que redirige el foco al botón oficinas de docentes y salir que redirige el foco al título del sitio web.

2.3.1.4. Diseñar e implementar menú de accesibilidad

2.3.1.4.1. Estructura con HTML y CSS

El menú de accesibilidad se compone de dos secciones principales. La primera que corresponde a cambios de estilo en la página y el menú avanzado que corresponde a cambios de estilo sobre el plano.

La sección del menú accesibilidad se compone de cuatro funciones principales, cambios de estilo, tamaño de letra, interespaciado y tipo de letra como se muestra en la Ilustración 2.27.



Ilustración 2.28: Vista principal de Menú de Accesibilidad.

2.3.1.4.2. Funciones de control de estilo y fuente

La primera que corresponde a estilos, contiene seis estilos que modifican la vista principal del prototipo, los estilos son estilo por defecto, negro sobre blanco, blanco sobre negro, amarillo sobre negro, negro sobre amarillo y tonos grises. Estas preferencias se han creado con el fin de proporcionar al usuario otras formas de visualización del prototipo en caso de que posea alguna discapacidad relacionada al color. En la Ilustración 2.29 se puede apreciar las combinaciones de estilo aplicadas sobre el prototipo.

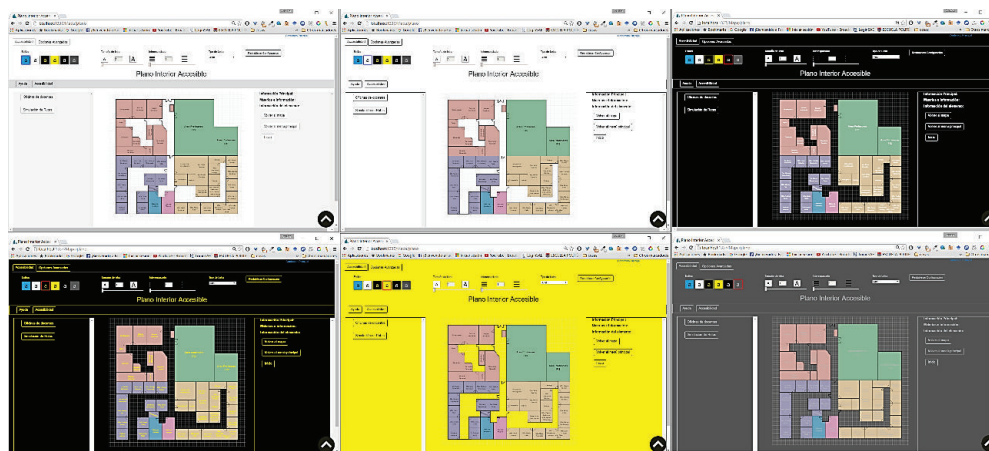


Ilustración 2.29: Vista principal de opciones de estilos para el prototipo.

La segunda parte, corresponde al tamaño de letra que puede seleccionarse mediante una barra que indica el número de veces en el que se incrementa el tamaño de letra. Esta barra ayuda a la visualización del texto a personas que poseen baja visión como se muestra en la Ilustración 2.30.

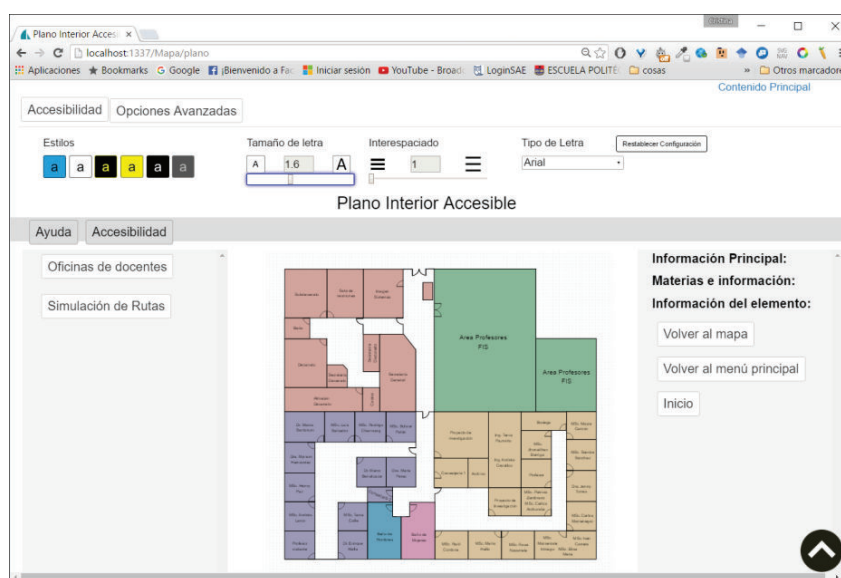


Ilustración 2.30: Vista de función tamaño de letra en el prototipo.

La opción de interespaciado permite incrementar la separación del texto para mejorar su legibilidad como se muestra en la Ilustración 2.31. La opción de interespaciado mejora la visualización del texto y es útil en caso de dislexia.

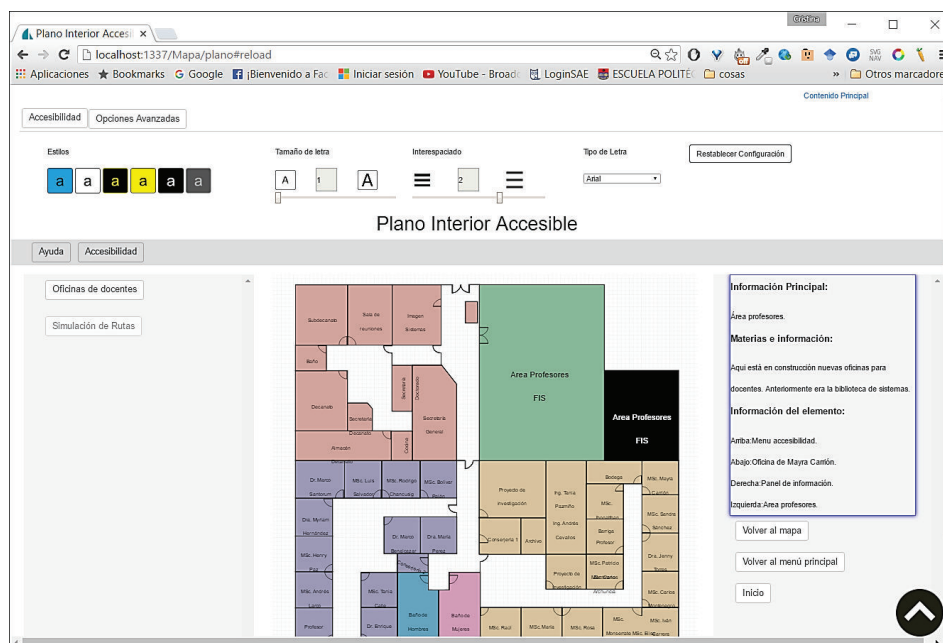


Ilustración 2.31: Vista de función interespaciado en el prototipo.

La opción de tipo de letra permite intercambiar la fuente según la preferencia del usuario en letras tipo Sans serif y Serif, esto influye en la distinción de los caracteres del texto. En la Ilustración 2.32 se puede apreciar al prototipo con el tipo de fuente Times New Roman.

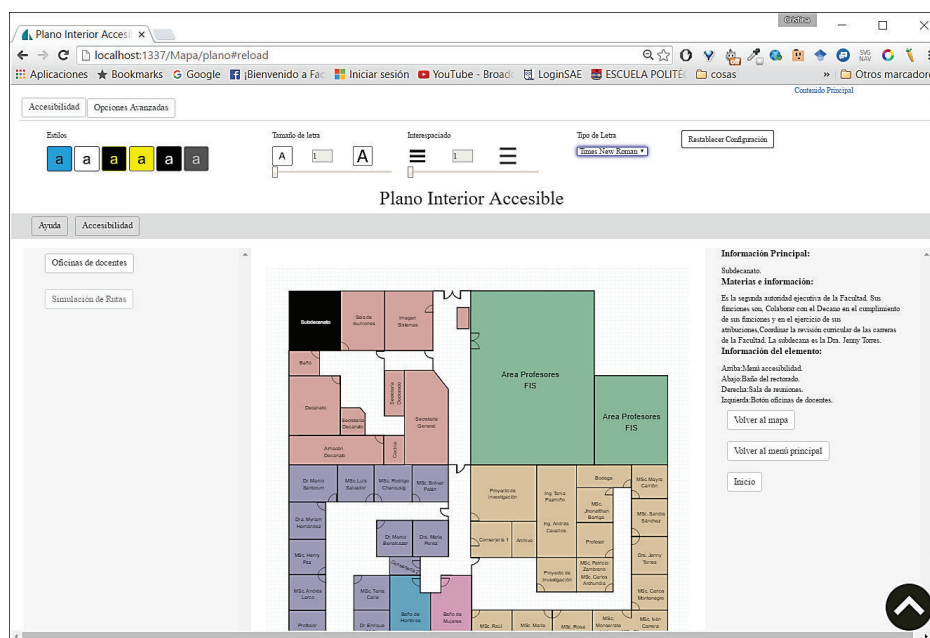


Ilustración 2.32: Vista de función tipo de letra en el prototipo.

La estructura del código incluye atributos de accesibilidad para incluir información extra sobre cada elemento y “tabindex” para navegación secuencia.

2.3.1.4.3. Funciones de estilo en el plano

La segunda parte del menú de accesibilidad corresponde a las opciones avanzadas. En estas opciones se podrá cambiar el estilo y formato del plano. En la Ilustración 2.33 se puede apreciar el menú de opciones avanzadas con sus dos secciones, patrones y filtros.



Ilustración 2.33: Opciones avanzadas en menú accesibilidad.

La primera opción corresponde a patrones, en esta opción se puede seleccionar los dos formatos de plano, con patrón a color o a blanco y negro como se aprecia en la Ilustración 2.34. Esta opción permite visualizar las regiones del plano sin depender de la distinción del color.

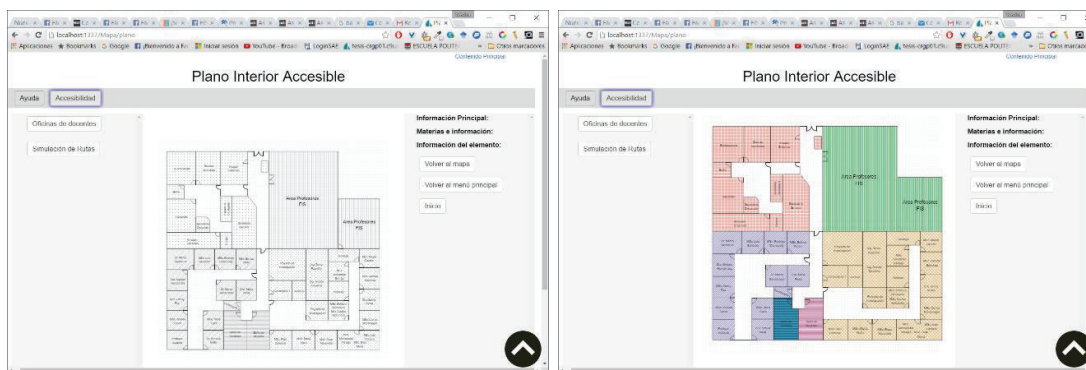


Ilustración 2.34: Patrones aplicados en plano accesible.

La segunda opción corresponde a los filtros que se puede aplicar sobre el plano y la página. Los filtros son saturación, tono, brillo, sepia, grises, invertir, opacidad y contraste como se visualiza en la Ilustración 2.35. El usuario puede modificar estas preferencias para que la visualización del prototipo se ajuste a sus necesidades.

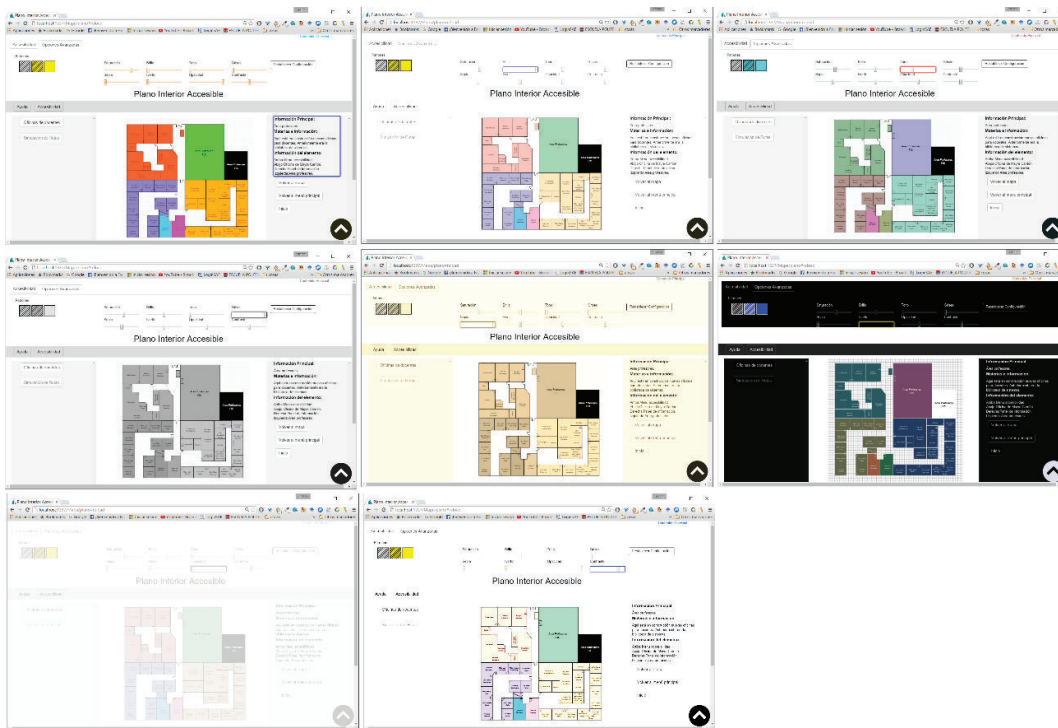


Ilustración 2.35: Opciones de filtros aplicados sobre el prototipo web.

2.4. CONSTRUCCIÓN DEL PROTOTIPO.

2.4.1. DISEÑO DEL SEGUNDO SPRINT

La etapa de construcción del prototipo corresponde al segundo sprint, cuyo entregable es el prototipo funcional.

En este sprint se desarrollan tareas finales como la implementación del modo simulación y la navegabilidad en todo el prototipo.

En este sprint se desarrollarán las siguientes historias de usuario:

1. Implementar modo simulación
2. Implementar navegabilidad en el sitio web.
3. Implementar navegabilidad en el plano interior.

2.4.1.1. Implementación del modo simulación

El modo simulación toma las coordenadas en x y en Y ingresadas en el ORM y las grafica en el plano SVG dinámicamente. Simultáneamente a la graficación, se reproduce un audio que indica los elementos cercanos a la posición actual para que el usuario reconozca el entorno y pueda ubicar un elemento en el plano.

2.4.1.1.1. Rutas mediante coordenadas.

Mediante las rutas ingresadas en la base de datos, se hizo una clasificación y ordenamiento de las coordenadas para agruparlas en cuatro rutas preestablecidas. Estas rutas se ingresaron en el ORM mediante un archivo JSON, las cuales por medio de una petición en AJAX se genera una animación dinámica que simula el movimiento mediante pasos. En la Ilustración 2.36 se puede apreciar el archivo JSON que alberga las coordenadas en el ORM.

```

1  {
2    "data": {
3      "informacion": [],
4      "inicio": [],
5      "coordenadas": [
6        {
7          "idcoordenadas": 1,
8          "coordenadax": 455,
9          "coordenaday": 80,
10         "owner": 1,
11         "id": 1
12       },
13       {
14         "idcoordenadas": 2,
15         "coordenadax": 455,
16         "coordenaday": 100,
17         "owner": 1,
18         "id": 2
19       },
20       {
21         "idcoordenadas": 3,
22         "coordenadax": 455,
23         "coordenaday": 120,
24         "owner": 1,
25         "id": 3
26       },

```

Ilustración 2.36: Archivo JSON de rutas preestablecidas mediante coordenadas.

2.4.1.1.2. Función para graficar rutas en SVG.

La representación gráfica de las coordenadas en SVG se ha realizado mediante el controlador coordenadas. Este controlador alberga una función que se puede apreciar en la Ilustración 2.37, en donde toma las coordenadas y las grafica utilizando una librería de animación para SVG llamada d3. Es una librería JavaScript usada para crear documentos basados en datos, es decir que permite manipular estos datos utilizando HTML, CSS y SVG [33]. Las coordenadas se han adaptado a la resolución del plano y se grafican mediante un circulo en SVG, la función define un tiempo entre la graficación de cada coordenada que simula los pasos de una persona promedio.

```

4360 function mostrarRuta(tabindex, idRuta) {
4361     $.post(
4362         '/Mapa/coordenadas', {
4363             idRuta: idRuta
4364         },
4365         function (data) {
4366             //renderiar mapa
4367             var datauser = data;
4368             renderizarMapa(datauser['mapa'], tabindex);
4369
4370             var aux = 1;
4371             var total = Object.keys(datauser['user']).length;
4372
4373             $.each(datauser['user'], function (i, item) {
4374
4375                 setTimeout(function () {
4376                     if (aux == total) {
4377                         d3.select(".currentLayer").append("circle").attr("cx", item["coordenadax"]).attr("cy",
4378                         item["coordenaday"]).attr("r", 4).style("fill", "yellow");
4379                     } else {
4380                         d3.select(".currentLayer").append("circle").attr("cx", item["coordenadax"]).attr("cy",
4381                         item["coordenaday"]).attr("r", 4).style("fill", "red");
4382                     }
4383                     aux++;
4384                 }, 1500 * i);
4385             });
4386             console.log("Error: en ajax plano");
4387         });

```

Ilustración 2.37: Función para graficar las rutas del ORM en formato SVG.

2.4.1.1.3. Instrucciones y grabación de audio de indicaciones.

Las instrucciones se grabaron tomando en cuenta el número aproximado de pasos que tiene que seguir el usuario para llegar al elemento. Las instrucciones se proporcionan referenciando elementos como puertas y oficinas

Las instrucciones se han grabado utilizando la herramienta online text to speech [34], para controlar la velocidad de lectura de las instrucciones y generar un archivo mp3 que será embebido al código HTML. En la Ilustración 2.38 se observa la función para reproducción de audio en el modo simulación.

```

4389 function playaudio(audio1) {
4390     var audio = document.getElementById(audio1);
4391     audio.play();
4392 }
4393 }

```

Ilustración 2.38: Función de reproducción de audio embebido en las rutas del modo simulación.

2.4.1.1.4. Audio y definición de menú de rutas en la interfaz principal.

El menú de simulación se ha creado mediante una lista en donde cada elemento corresponde a una ruta que llama a la función mostrarRutas y playAudio para graficar las coordenadas en SVG y simultáneamente reproducir el audio. En la ilustración 2.39 se muestra el modo simulación en ejecución.

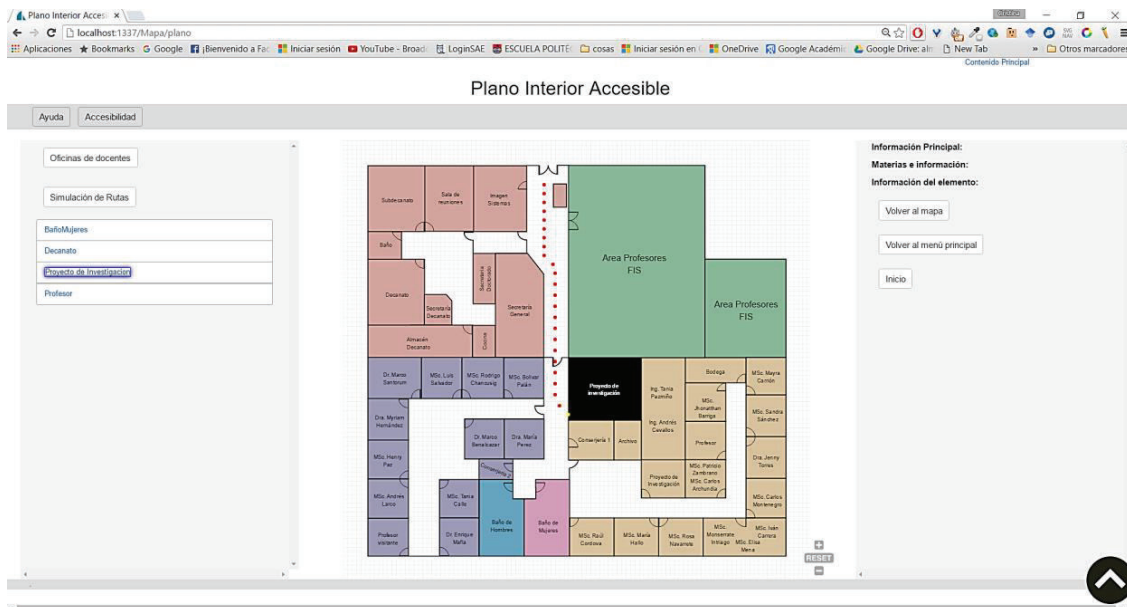


Ilustración 2.39: Resultado de la ejecución del modo simulación.

2.4.1.2. Implementación de navegabilidad en el sitio web.

2.4.1.2.1. Elementos y orden de navegación.

Los elementos en donde se han agregado atributos de navegación son elementos que incluyen funcionalidades al mapa, es decir botones que otorgan alguna funcionalidad al prototipo, títulos principales y bloques de información que serán leídos por el lector de pantalla al recibir el foco.

A continuación, se muestra el orden de navegación secuencial, establecido mediante el atributo “tabindex”.

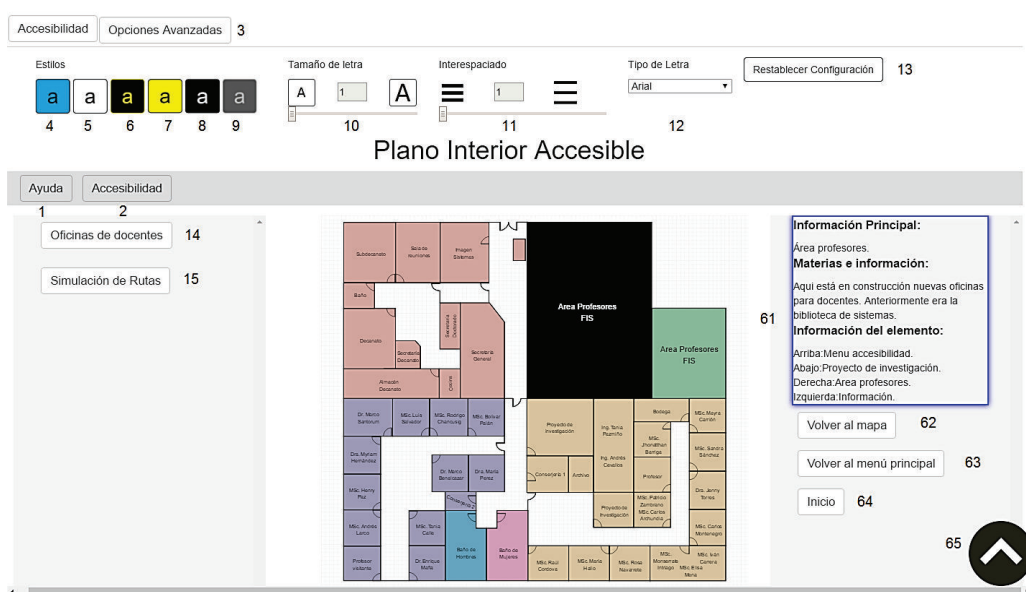


Ilustración 2.40: Orden secuencial de navegación de los elementos del prototipo.

2.4.1.2.2. Funciones para navegación con cursores

A más de la navegación secuencial, es necesario añadir navegación mediante cursores para que el prototipo sea totalmente operable por teclado. Para ello se definieron funciones para asociar los cursores a las posiciones contiguas de cada elemento, arriba, abajo, izquierda y derecha.

Posteriormente se crearon funciones con JQuery como las mostradas en la Ilustración 2.41, que están asociadas al id de cada tecla. Esto se lo realiza con el fin de colocar el foco en los elementos de todo el prototipo según la selección del usuario. La programación se basó en casos, tomando en cuenta la posición de los componentes alrededor del elemento que posee el foco.

```

}
//div accesibilidad
if (e.which == 37 || e.which == 39) {
  switch (focused.attr("id")) {
    case 'menuaccesibilidad2':
      if (e.which == 37) {
        $("#select[tabindex='1']").focus();
      } else {
        $("#a[tabindex = '1']").focus();
      }
      break;
    }
  }
}

if (e.which == 38 || e.which == 40) {
  switch (focused.attr("id")) {
    case 'menuaccesibilidad2':
      if (e.which == 38) {
        if ($("#button[id='1']").attr('aria-expanded') == 'false') {
          $("#button[id='2']").focus();
        } else {
          $("#button[id='1']").focus();
        }
      }
      } else {
        $("#button[tabindex='1']").focus();
      }
      break;
    }
  }
}
}

```

Ilustración 2.41: Función para navegación con cursores en el elemento div del menú accesibilidad.

Además, se crearon atajos de teclado para los elementos principales del prototipo, como el primer elemento del plano, menú de accesibilidad y ayuda, botón de simulación, etc. Estos atajos de teclado permiten activar de manera automática varias funcionalidades del prototipo.

```

//Creación de accesskey
document.getElementById("ofidocentes").accessKey = "o";
document.getElementById("volvermapa").accessKey = "p";
document.getElementById("Guardia").accessKey = "i";
document.getElementById("botondeaccesibilidad").accessKey = "c";
document.getElementById("botondeayuda").accessKey = "h";
document.getElementById("a-c").accessKey = "a";
document.getElementById("d-l").accessKey = "l";
document.getElementById("m-p").accessKey = "m";
document.getElementById("q-z").accessKey = "q";
document.getElementById("simurutas").accessKey = "s";

```

Ilustración 2.42: Funciones para atajos por teclado.

2.4.1.3. Implementación de la navegabilidad en el plano interior.

2.4.1.3.1. Elementos y orden de navegación.

El esquema de navegación secuencial para el plano se ha establecido en la Ilustración 2.43. La navegación secuencial se realiza con la tecla Tabulador para colocar el foco sobre el elemento y seleccionarlo con la tecla Enter para desplegar su información.

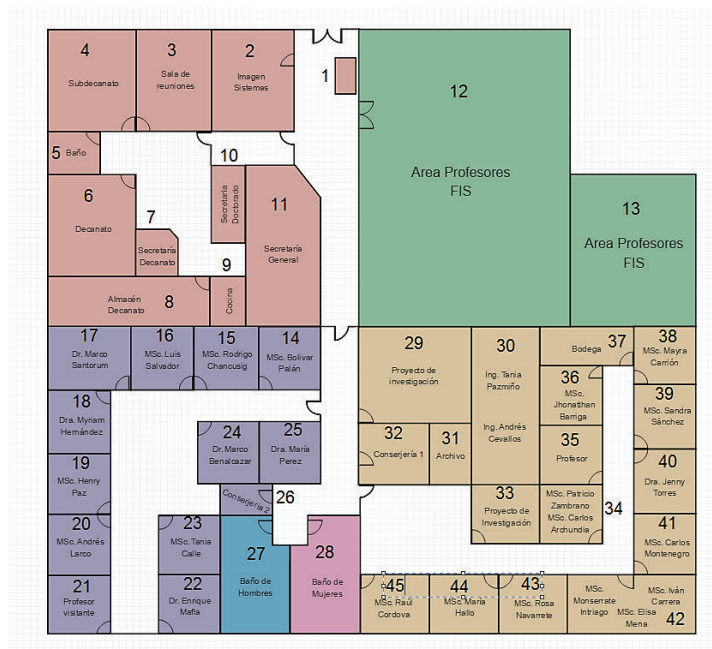


Ilustración 2.43: Esquema de navegación secuencial del plano interior.

2.4.1.3.2. Funciones para navegación con cursores

La implementación de navegación mediante cursores es similar en todo el prototipo, en el caso del plano se ha seleccionado un rango de valores de "tabindex" para aplicar la selección de los elementos mediante la tecla Enter tal y como se muestra en la Ilustración 2.44.

```
//Keypress event with enter key
$("body").keypress(function (e) {
  if (e.which == 13) {
    var focused = $(':focus');
    var focusedParent = focused.parent();
    if (focused.attr('tabindex') >= 29 && focused.attr('tabindex') <= 115) {
      focusedParent.trigger("click");
    } else {
      focused.trigger("click");
    }
  }
});
```

Ilustración 2.44: Función para navegación con cursores aplicada al plano interior.

2.4.2. DISEÑO DEL TERCER SPRINT

2.4.2.1. Diseñar menú de ayuda

2.4.2.1.1. Estructura con HTML y CSS

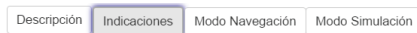
El menú de ayuda se compone de 4 secciones. Descripción, Indicaciones, Modo Navegación y Modo Simulación. Se ha agregado el atributo “tabindex” para establecer un orden de navegación entre las opciones del menú. Cada uno de los botones del menú posee el atributo “alt” que alberga la descripción del elemento. La Ilustración 2.45 muestra la estructura HTML del menú de ayuda.

```
<div id="menuayuda" class="panel-collapse collapse" data-collapse-group="divayuda">
  <ul class="nav nav-tabs">
    <li class="active">
      <button data-toggle="tab" class="btn btn-default elementomenu subtitulos" href="#descripcion" alt="Descripción"
id="botondescrpcion" tabindex="11" onclick="menuayuda()">Descripción</button>
    </li>
    <li>
      <button data-toggle="tab" class="btn btn-default elementomenu subtitulos" href="#indicaciones" alt="Indicaciones"
id="botonindicaciones" tabindex="12" onclick="menuayuda2()">Indicaciones</button>
    </li>
    <li>
      <button data-toggle="tab" class="btn btn-default elementomenu subtitulos" href="#navegacion" alt="Modo Navegación"
id="botonnavegacion" tabindex="13" onclick="menuayuda3()">Modo Navegación</button>
    </li>
    <li>
      <button data-toggle="tab" class="btn btn-default elementomenu subtitulos" href="#simulacion" alt="Modo Simulación"
id="botonsimulacion" tabindex="14" onclick="menuayuda4()">Modo Simulación</button>
    </li>
  </ul>
```

Ilustración 2.45: Descripción en menú de ayuda.

2.4.2.1.2. Instrucciones y descripciones

La descripción alberga información que detalla al usuario la estructura del prototipo y la distribución de sus elementos. Las indicaciones le dan al usuario instrucciones de uso del prototipo; contiene opciones de navegación y atajos de teclado para utilizar las opciones más importantes del prototipo. Los botones de navegación y simulación dan una descripción de ambos modos y su forma de usar. La ilustración 2.46 muestra el botón de indicaciones dentro del menú de ayuda.



Indicaciones.

- Para navegar por el plano, puede utilizar la tecla Tabulador para moverse en orden secuencial, o las flechas arriba, abajo, izquierda y derecha para navegar en cualquier dirección.
 - Puede utilizar los siguientes atajos de teclado para ir a los siguientes elementos:
 - Menú oficinas de docentes: Alt + o.
 - Menú simulacion de rutas: Alt + s.
 - Ultimo elemento del plano: Alt + p.
 - Menú ayuda: Alt + h.
 - Menú accesibilidad: Alt + c.
 - Rango A - C: Alt + a.
 - Rango D - L: Alt + l.
 - Rango M - P: Alt + m.
 - Rango Q - Z: Alt + q.
 - Inicio: Escape.
- La navegación se la realiza ubicando el elemento deseado ya sea con los cursores, tecla tab o ratón. Una vez seleccionado el elemento en el plano interior, el foco se colocará automáticamente en la descripción del elemento. Si se desea retornar al plano, se debe escoger el botón inferior "Volver al mapa", con la tecla tab o con la flecha inferior. Si se desea retornar al menú izquierdo de búsqueda, se debe seleccionar el botón "Volver al menú principal". Si desea salir y retornar al inicio, se debe escoger el boton salir.

Ilustración 2.46: Indicaciones en menú de ayuda.

CAPÍTULO 3. PRUEBAS DEL PROTOTIPO EN EL CASO DE ESTUDIO

Para verificar el grado de accesibilidad del prototipo, es necesario realizar una serie de pruebas para comprobar el grado de concordancia con las pautas de accesibilidad WGAG 2.0. Las pruebas que se realizan a continuación, tienen el objetivo de identificar que el desarrollo del prototipo web cumpla con los estándares y principios de accesibilidad necesarios para llegar a un nivel de conformidad AA.

En este capítulo se describen las pruebas realizadas con herramientas validadoras, herramientas simuladoras y usuarios finales no videntes; con el fin de evaluar el despliegue del prototipo en un ambiente real.

3.1. PRUEBAS DEL PROTOTIPO CON HERRAMIENTAS VALIDADORAS.

Las herramientas validadoras se encargan de verificar el cumplimiento de las pautas de accesibilidad WCAG 2.0 mediante una serie de pruebas aplicadas en el estilo y estructura de la página web.

3.1.1. EXAMINATOR

Examinator analiza la estructura del sitio web tomando como principio las pautas de accesibilidad WGAC 2.0, otorgando una calificación sobre 10 puntos al sitio web. Cada prueba aplicada al sitio web posee un ponderado, el cual se determina en base a la importancia de la prueba aplicada sobre el análisis.

En este caso, se ha evaluado el uso de estilos, jerarquía de encabezados, alternativas textuales en imágenes, enlaces para saltar contenido e idioma de la página como se muestra en la Ilustración 3.1.

Ilustración 3.1: Examinator con el prototipo accesible.

En la Ilustración 3.2 se puede observar que la herramienta ha elaborado 10 pruebas, las cuales se tienen dos operadores, el primer operador es un ponderado entre 5 y 10, y el segundo operador es un ponderado entre 1 y 3.5 según la relevancia de la pauta evaluada. El resultado final se obtiene de la multiplicación entre ambos operadores por cada prueba y el promedio entre la suma de todas ellas.

El ponderado final se obtiene dividiendo la suma del segundo denominador por el promedio final.

#	Situación	N	P	N/P
1	Hay 63 casos de reglas CSS que no especifican los colores de primer plano y fondo a la vez	5	1.08	5.4
2	Se usan 5 elementos de encabezado	10	3.42	34.2
3	No se usan atributos para controlar la presentación visual	10	3.2	32
4	Todas las imágenes tienen una alternativa textual	10	2.7	27
5	El primer enlace de la página lleva al contenido principal de la página	10	2.24	22.4
6	Hay 5 enlaces que permiten saltar bloques de contenido	10	2.24	22.4
7	No se usan elementos para controlar la presentación visual	10	2.2	22
8	Todos los botones gráficos tienen una alternativa textual	10	1.98	19.8
9	Se identifica el idioma principal de la página con el código "es"	10	1.8	18
10	La página tiene un elemento title	10	1.62	16.2
			2.248	21.940
Score ponderado = round(21.940 / 2.248) = 9.8				

Ilustración 3.2: Captura de las pruebas realizadas sobre el prototipo con el puntaje correspondiente.

En los resultados por personas se toma en cuenta, limitaciones totales para ver, limitaciones graves para ver, limitaciones de miembros superiores, limitaciones de comprensión y derivadas con la edad. En la Ilustración 3.3 se muestran las pruebas realizadas por personas y el puntaje obtenido.

Resultados por personas

- Limitación total para ver: Score **10** (9 pruebas)
- Limitación grave para ver: Score **9.6** (10 pruebas)
- Limitación de los miembros superiores: Score **10** (7 pruebas)
- Limitación para comprender: Score **10** (5 pruebas)
- Limitaciones derivadas de la edad: Score **9.3** (9 pruebas)

Ilustración 3.3: Pruebas simuladas por personas.

3.1.2. TAW

TAW analiza un sitio web en base al URL. El análisis se basa en las Pautas de Accesibilidad WCAG 2.0 y genera un informe de resumen basado en los hallazgos.

El resultado incluye los problemas encontrados, las advertencias, y puntos no verificados, que se clasifican en base a cada principio de accesibilidad, perceptible, operable, comprensible y robusto.

El informe incluye tres vistas principales:

Vista marcada: Muestra las incidencias detectadas.

Detalle: Muestra la línea de código en donde indiquen los hallazgos y las técnicas relacionados con los mismos.

Listado: Corresponde a un resumen agrupado en los cuatro principios, a nivel de normativa y pauta [35].

En la Ilustración 3.4 se muestra el resultado del análisis de accesibilidad realizado con TAW. En el resultado se aprecia los problemas encontrados, las advertencias y los puntos de verificados.

Resumen de resultados

Información del análisis

Recurso: <https://tesis-crgp01.c9users.io/>
 Fecha: 27/06/2016 22:00
 Pautas WCAG 2.0
 Nivel del análisis: AA
 Tecnologías: HTML, CSS

1 Problemas en 1 criterios de éxito
 Son necesarias correcciones
 Perceptible 0
 Operable 0
 Comprensible 0
 Robusto 1

292 Advertencias en 3 criterios de éxito
 Es necesario revisar manualmente
 Perceptible 0
 Operable 3
 Comprensible 0
 Robusto 289

19 No verificados en 19 criterios de éxito
 Comprobación completamente manual
 Perceptible 4
 Operable 9
 Comprensible 5
 Robusto 1

Acceda al [Informe detallado](#) para obtener más información sobre las incidencias detectadas.

Ilustración 3.4: Pruebas de accesibilidad realizadas con TAW.

En el análisis del principio Perceptible se toman en cuenta características sensoriales, información mediante color, contraste y texto en las imágenes. En la Ilustración 3.5 se puede ver que los resultados constan como no verificados, lo que no influye en el análisis final.

Perceptible	Operable	Comprensible	Robusto		
La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que puedan percibirlos.					
Tipología	Comprobación	Técnicas	Resultado	Incidencias	Números de Líneas
1.3.3 - Características sensoriales					
Presentación	Características sensoriales	[G96]		1	
1.4.1 - Uso del color					
Presentación	Información mediante color	[G14 G122 G182 G183]		1	
1.4.3 - Contraste (Mínimo)					
Presentación	Contraste	[G18 G148 G174]		1	
	Contraste para fuentes grandes	[G145 G148 G174]		1	
1.4.5 - Imágenes de texto					
Imágenes	Imágenes susceptibles de ser sustituidas por marcado	[C22 C30 G140]		1	

Ilustración 3.5: Principio Perceptible evaluado por TAW.

En el principio operable, se evalúan aspectos como movimiento del foco, movimiento mediante teclado, contenido parpadeante o repetitivo, orden de navegación, encabezados y etiquetas. En la Ilustración 3.6 se muestra los resultados del análisis del principio Operable en la herramienta TAW.

Perceptible	Operable	Comprendible	Robusto		
Los componentes de la interfaz de usuario y la navegación deben ser operables.					
Tipología	Comprobación	Técnicas	Resultado	Incidencias	Números de Líneas
2.1.1 - Teclado					
Scripts	Movimiento automático del foco ?	[G98]	?	1	
2.1.2 - Sin bloques de teclado					
Página web	Movimiento del foco mediante teclado ?	[G21]	?	1	
2.2.1 - Tiempo ajustable					
Página web	Limite de tiempo de sesión ?	[G133 G198]	?	1	
	Limite de tiempo controlado mediante un script ?	[G198 G180 SCR16]	?	1	
	Lectura de textos en movimiento ?	[G4 G198 SCR33 SCR36]	?	1	
2.2.2 - Pausar, detener, ocultar					
Página web	Contenido en movimiento o parpadeante ?	[G4 SCR33 G187 G192 SCR22 G186 G191]	?	1	
2.3.1 - Umbral de tres destellos o menos					
Presentación	Destellos por debajo del umbral límite ?	[G19 G126 G15]	?	1	
2.4.1 - Evitar bloques					
Navegación	Saltar bloques de contenido repetidos ?	[G1 G123 G124]	?	1	
	Bloques de contenido ?	[H50 H70 SCR28]	?	1	
2.4.2 - Páginas tituladas					
Página web	Página con título descriptivo ?	[G88]	!	1	6
2.4.3 - Orden del foco					
Navegación	Orden lógico de navegación ?	[G59 H4 SCR26 SCR37 SCR27]	?	1	
2.4.5 - Múltiples vías					
Sitio web	Múltiples medios de localización ?	[G125 G64 G63 G161 G126 G185]	?	1	
2.4.6 - Encabezados y etiquetas					
Estructura y semántica	Contenido adecuado de encabezados y etiquetas ?	[G130 G131]	!	2	44, 57
2.4.7 - Foco visible					
Scripts	Cambio de foco con el evento 'onfocus' ?	[F55]	?	1	

Ilustración 3.6: Principio Operable evaluado por TAW.

En la Ilustración 3.7 se puede ver el resultado del análisis del principio Comprendible, en donde se evalúan aspectos como el idioma del sitio web, cambios de contenido del foco, navegación y consistencia.

Perceptible	Operable	Comprendible	Robusto		
La información y el manejo de la interfaz de usuario debe ser comprensible.					
Tipología	Comprobación	Técnicas	Resultado	Incidencias	Números de Líneas
3.1.2 - Idioma de las partes					
Página web	Cambios en el idioma ?	[H58]	?	1	
3.2.1 - Al recibir el foco					
Scripts	Cambio de contenidos con el evento 'onfocus' ?	[G107]	?	1	
	Cambios inesperados del foco en el evento 'onfocus' ?	[F55]	?	1	
	Apertura de ventana al cambiar el foco ?	[G107]	?	1	
	Apertura de ventana con el evento 'onload' ?	[F52]	?	1	
Página web	Cambio de contenidos con el evento 'onfocus' ?	[G107]	?	1	
3.2.2 - Al introducir datos					
Formularios	Cambios causados por el evento 'onChange' en un selector ?	[H84]	?	1	
3.2.3 - Navegación consistente					
Sitio web	Navegación consistente ?	[G61]	?	1	
3.2.4 - Identificación consistente					
Sitio web	Denominación consistente ?	[G197]	?	1	

Ilustración 3.7: Principio Comprendible evaluado por TAW.

En robustez, se evalúa que el documento del sitio web este bien formado y la validación de hojas de estilo, como se puede apreciar en la Ilustración 3.8.

Perceptible	Operable	Comprendible	Robusto		
El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de agentes de usuario, incluyendo las ayudas técnicas.					
Tipología	Comprobación	Técnicas	Resultado	Incidencias	Números de Líneas
4.1.1 - Procesamiento					
Página web	Página 'bien formada' ?	[G134]	✘	1	103
	Validación de las hojas de estilo ? (https://geovisualizadoraccesible-crgp01.cifusers.io/styles/css/general/bootstrap.min.css)	[G134]	!	136	69, 169, 170, 187, 188, 199, 200, 220, ...*
	Validación de las hojas de estilo ? (https://geovisualizadoraccesible-crgp01.cifusers.io/styles/css/general/bootstrap.css)	[G134]	!	118	169, 170, 187, 188, 199, 200, 1426, 1463, ...*
	Validación de las hojas de estilo ? (https://geovisualizadoraccesible-crgp01.cifusers.io/styles/css/general/index.css)	[G134]	!	22	12, 13, 14, 15, 16, 17, 18, 18, ...*
	Validación de las hojas de estilo ? (https://geovisualizadoraccesible-crgp01.cifusers.io/styles/css/general/estilo.css)	[G134]	!	13	24, 25, 26, 27, 32, 33, 34, 35, ...*
4.1.2 - Nombre, función, valor					
Página web	Nombre, rol y valor ?	[G108 SCR21 G135 G10]	?	1	

Ilustración 3.8: Principio robusto evaluado por TAW.

3.2. PRUEBAS DEL PROTOTIPO CON HERRAMIENTAS SIMULADORAS.

Las herramientas simuladoras permiten emular diferentes defectos visuales sobre una interfaz. De esta manera se puede visualizar cómo diferentes discapacidades visuales influyen en la percepción de los elementos de un sitio web. Esto permite mejorar el diseño de las pantallas principales para que puedan ser usadas indistintamente de la discapacidad del usuario.

3.2.1. NO COFFEE

NoCoffee es un simulador visual que permite entender los problemas a los que se enfrentan las personas con discapacidad visual al interactuar con una interfaz. Entre las discapacidades que simula NoCoffee están:

Baja visión: esto se relaciona al tamaño disminuido de fuente.

Bajo contraste: Este tipo de discapacidad impide diferenciar texto en un contraste muy bajo.

Daltonismo: Es un tipo de discapacidad asociada al color que impide diferenciar tonalidades.

Cataratas, efecto fantasma y deslumbramiento: son discapacidades que interfieren con la visualización completa de una interfaz.

Nistagmo: Movimiento rápido e involuntario de los ojos.

También alberga otras discapacidades como obstrucción del campo visual, flotadores, visión central obstruida o irregular, campos visuales parciales, etc.

El prototipo se ha evaluado con varias discapacidades para medir el grado de usabilidad que se tiene. Debido al uso combinado de lector de pantalla con teclado, las dificultades de uso debido a las discapacidades visuales, se observan disminuidas [36].

Visión borrosa

En la Ilustración 3.9 se puede apreciar el filtro de visión borrosa aplicado al prototipo. Se puede ver cómo afecta mayormente a la legibilidad del texto.

Retinitis pigmentosa

La Ilustración 3.12 muestra el filtro de retinitis pigmentosa, este tipo de discapacidad permite ver únicamente los elementos que se encuentran en el foco central, en el caso del prototipo, únicamente se puede ver el plano interior mas no los elementos como menú y descripciones.

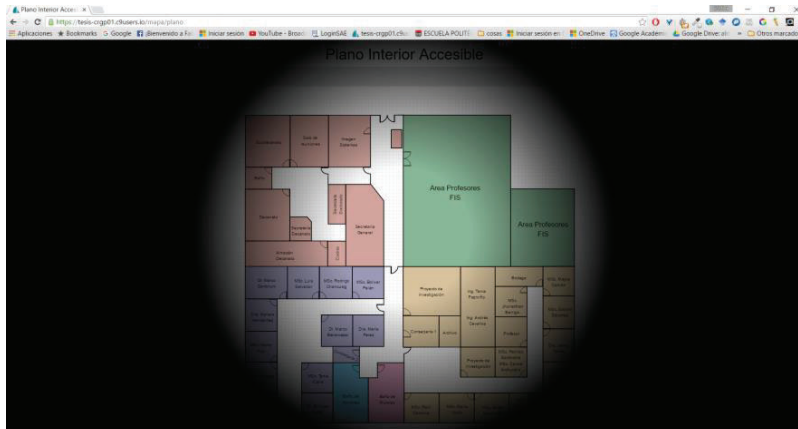


Ilustración 3.12: Filtro de retinitis pigmentosa de No Coffee.

Retinopatía diabética

En la Ilustración 3.13 se aprecia como el filtro de retinopatía diabética afecta la mayor parte de visibilidad del sitio web, dificultando su usabilidad.

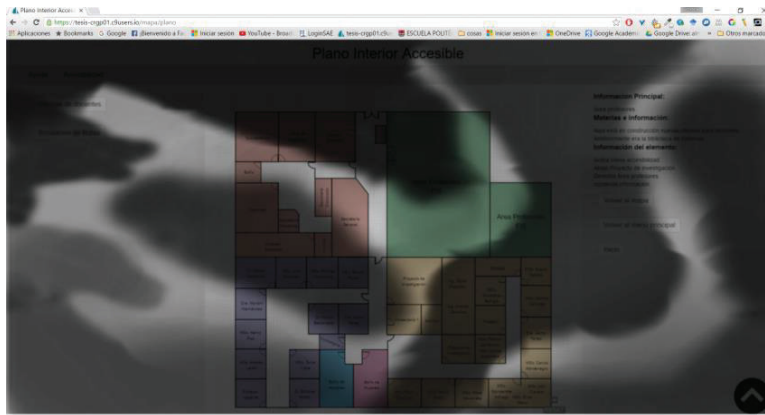


Ilustración 3.13: Filtro de retinopatía diabética de No Coffee.

3.2.2. IMPAIRMENT SIMULATOR

Impairment simulator emula varios defectos de visión y las deficiencias auditivas en imágenes y archivos de sonido. Da una indicación de cómo esos archivos pueden ser percibido en diferentes condiciones de discapacidad. El propósito de este simulador es ayudar a entender como diferentes tipos de discapacidades afectan a la interacción con diferentes sistemas [37]. En la Ilustración 3.14 se puede ver la pantalla principal de Impairment Simulator con su descripción.

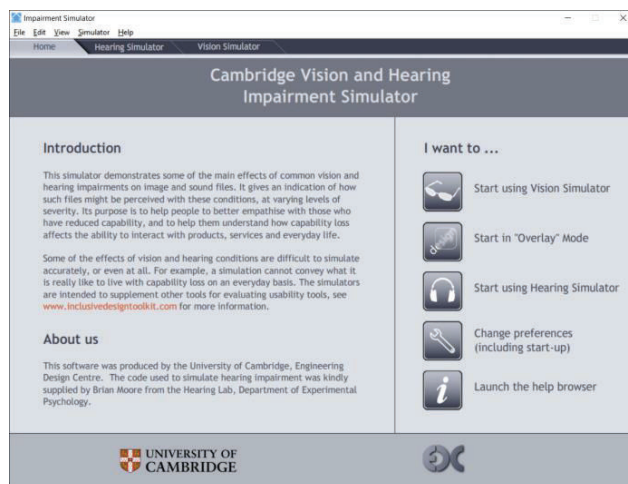


Ilustración 3.14: Captura de la pantalla principal de Impairment Simulator.

Glaucoma

En la Ilustración 3.15 se muestra como el filtro de glaucoma afecta la visualización del prototipo, oscureciendo las partes de alrededor del centro de la imagen.

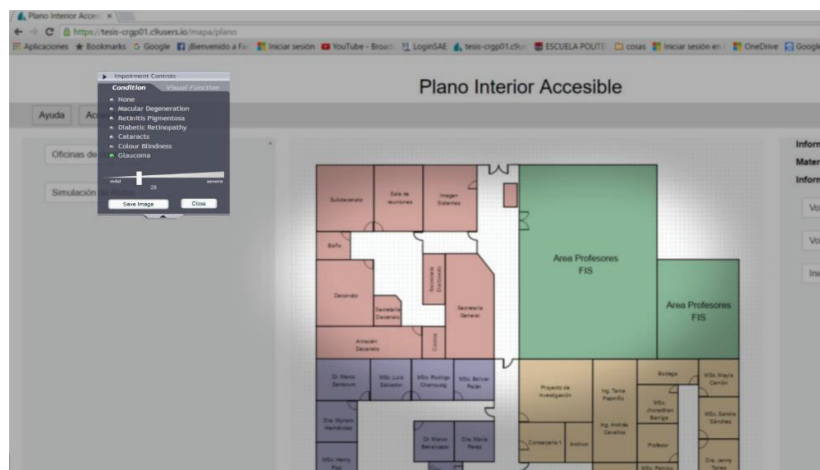


Ilustración 3.15: Filtro de Glaucoma de Impairment Simulator.

Retinopatía diabética

En la Ilustración 3.16 se aprecia como el filtro de retinopatía diabética impide visualizar claramente el prototipo al colocar varias manchas sobre el campo visual.

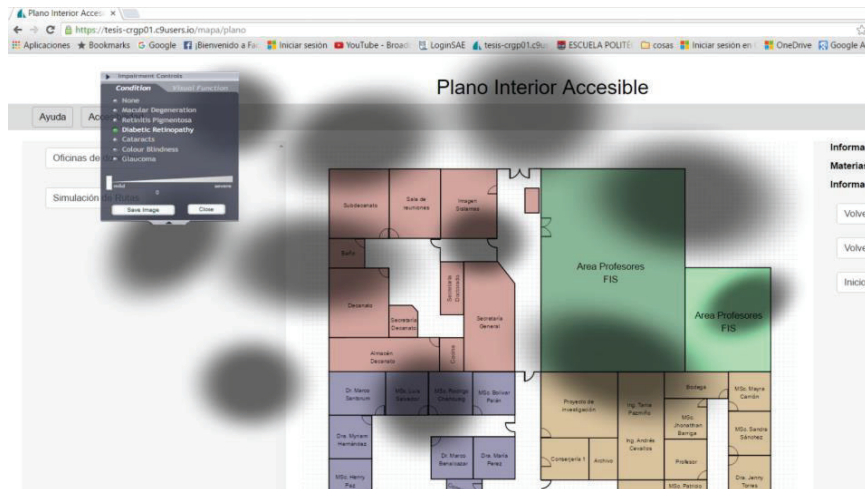


Ilustración 3.16: Filtro de Retinopatía diabética de Impairment Simulator.

Baja agudeza visual

En la Ilustración 3.17 se muestra como el filtro de baja agudeza visual impide ver y utilizar el prototipo al opacar el texto y las imágenes.

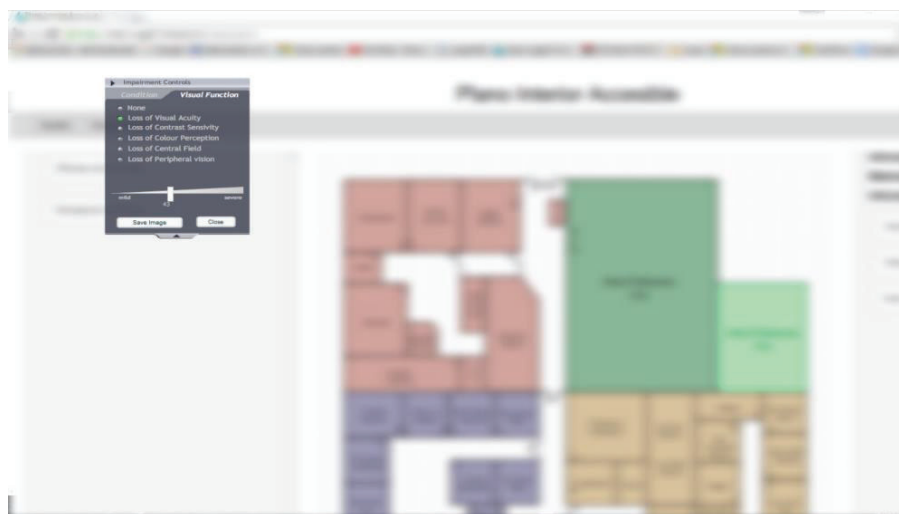


Ilustración 3.17: Filtro de baja agudeza visual de Impairment Simulator.

Cataratas

La Ilustración 3.18 muestra el filtro de cataratas. Se puede apreciar que este filtro afecta principalmente en el reconocimiento de colores y en la nitidez de la imagen.

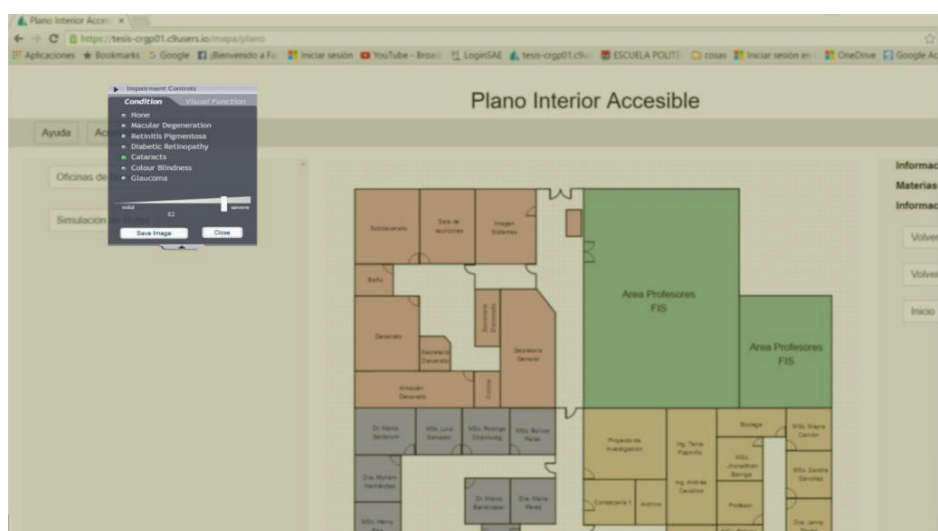


Ilustración 3.18: Filtro de Cataratas de Impairment Simulator.

Mediante las herramientas simuladoras se puede entender como varias discapacidades visuales afectan el campo de visión del prototipo.

Mediante las funcionalidades de navegabilidad, ayudas de teclado y atributos de accesibilidad, se puede disminuir los efectos de las deficiencias visuales sobre el prototipo, de tal manera que puede ser usado sin requerir del uso visual.

3.2.3. SPECTRUM

Esta extensión le ayuda a probar las páginas web para las personas con diferentes tipos de enfermedades visuales asociadas al color. Es particularmente útil para sitios web con visualizaciones de datos, debido a que algunos colores pueden no ser distinguible de otros colores en las tablas.

A continuación, se muestra algunos ejemplos de visualización del prototipo con varios tipos de discapacidades asociadas al color.

Protanopia

La protanopia impide ver los tonos rojos. En la Ilustración 3.19 se aprecia que en el plano interior únicamente se observan tonos grises, marrón y azul.

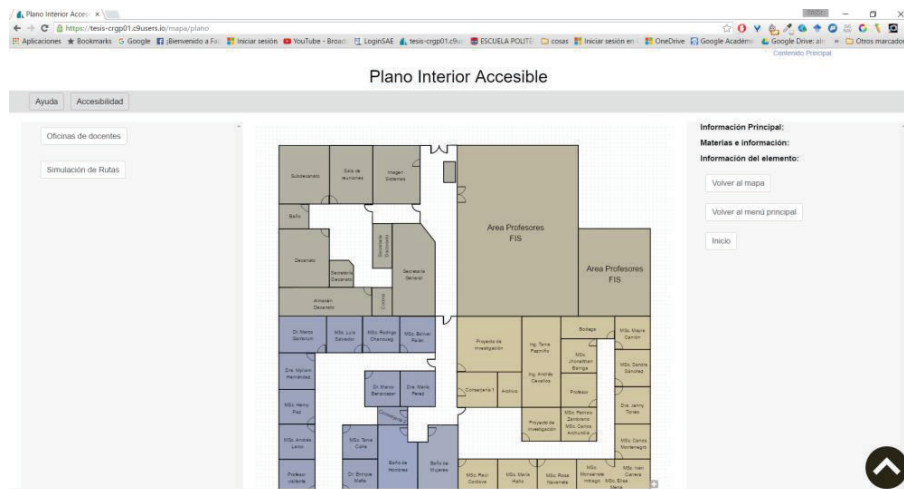


Ilustración 3.19: Filtro de Protanopia de Spectrum.

Tritanopia

En la tritanopia se tiene una carencia en reconocer el color azul. En la Ilustración 3.10 se muestra como la imagen se muestra en tonos verdes y rojos.

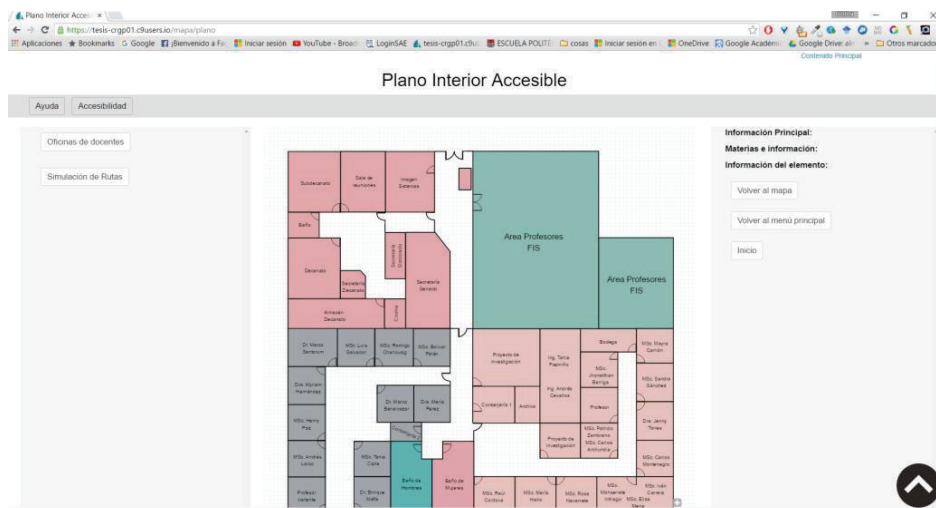


Ilustración 3.20: Filtro de Tritanopia de Spectrum.

Acromatopsia

En el filtro de acromatopsia se aprecia que hay ausencia de color como se muestra en la Ilustración 3.21.

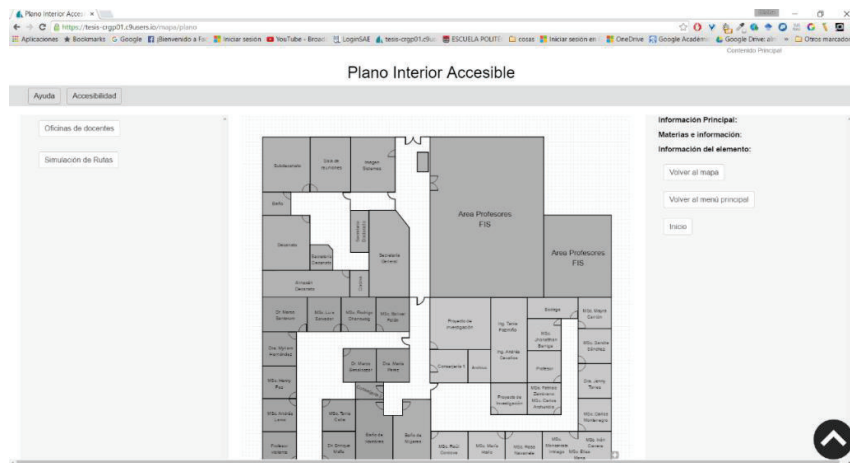


Ilustración 3.21: Filtro de acromatopsia de Spectrum.

Deuteranopia

En la Ilustración 3.22 se puede apreciar como se observa el prototipo cuando no se distinguen los tonos rojos de los verdes.

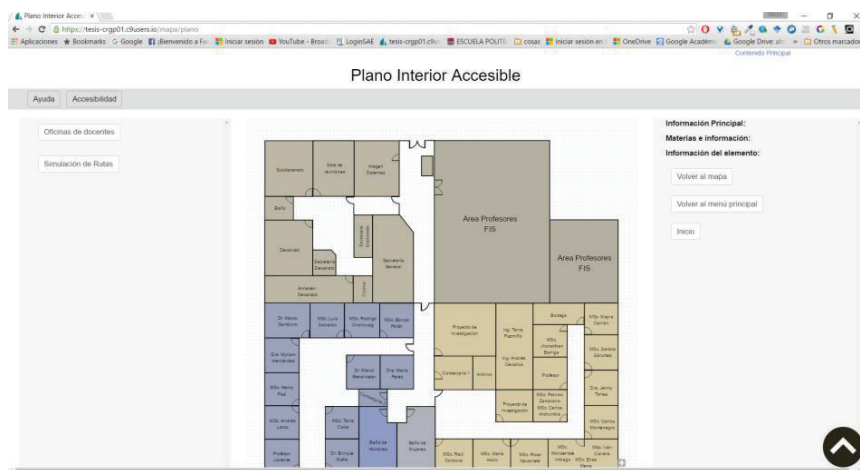


Ilustración 3.22: Filtro de Deuteranopia de Spectrum.

Las discapacidades visuales asociadas a la percepción del color se pueden corregir mediante el uso del filtro de tono y los estilos del prototipo, dentro de las opciones de accesibilidad. Mediante estas ayudas de cambio de estilo, se puede escoger la combinación de color que mejor se ajuste a la percepción del usuario para distinguir los elementos del prototipo.

3.3. PRUEBAS DEL PROTOTIPO CON PERSONAS NO VIDENTES.

Para realizar las pruebas de accesibilidad en un entorno real se ha contado con la participación de dos usuarios con pérdida total de visión. Los usuarios son profesionales y trabajan en entidades públicas, los cuales han adquirido experiencia en el uso de tecnología como computadoras y lectores de pantalla.

3.3.1. DATOS DEL PRIMER USUARIO

Nombre de usuario: Franklin Yaulu.

Edad: 36 años.

Profesión: Profesor de enseñanza inicial en Universidad Tecnológica Equinoccial.

3.3.2. DATOS DE LA PRIMERA EVALUACIÓN.

Fecha: 03/05/2016.

Hora de inicio: 10:00.

Hora de fin: 11:50.

Línea base:

- *Modelo del equipo:* Laptop Lenovo Z50-70.
- *Sistema operativo:* Windows 10.
- *Navegador:* Google Chrome v 51.0.
- *Lector de Pantalla:* Chrome Vox.
- *Tecnologías de asistencia:* Teclado y parlantes.

En la tabla 3.1 se puede observar el formato utilizado para las pruebas. Como elementos contiene el número de la tarea, descripción de la tarea, el grado de dificultad, si la tarea se ejecutó con ayuda y las observaciones.

Tabla 3.1: Pruebas de accesibilidad realizadas con Usuario 1.

No	Tarea	Grado de dificultad			¿Se ejecutó la tarea con ayuda?		Observaciones
		Fácil	Medio	Difícil	SI	NO	
1	Acceder al menú ayuda.	X			X		El teclado es desconocido para el usuario.
2	Acceder al menú descripción.		X			X	Problemas al reconocer teclas izquierda y derecha.
3	Acceder al menú indicaciones.		X			X	
4	Acceder a la oficina decanato mediante menú de búsqueda.	X				X	
5	Entender la información desplegada del elemento decanato.	X				X	
6	Volver al plano, mediante el botón "Volver al mapa".			X	X		Problemas para localizar los cursores en el teclado.
7	Acceder a la oficina del Ing. Andrés Larco mediante búsqueda en el plano.		X	X	X		Es necesario establecer un punto de referencia para localizar un elemento.
8	Entender la información desplegada del elemento Andrés Larco.	X				X	
9	Retornar al menú de búsqueda mediante el botón "Volver al menú principal"		X			X	
10	Utilizar el atajo en teclado para acceder al menú de búsqueda.	X				X	Es necesario familiarizarse con los atajos de teclado.
11	Activar el atajo en teclado para salir.	X				X	
12	Acceder al menú accesibilidad.	X				X	
13	Activar la opción de simulación.			X	X		Dificultad para encontrar el modo simulación y navegación.
14	Activar la ruta "Proyecto de investigación".	X			X		
15	Escuchar la información y llegar al elemento seleccionado.						El ritmo de paso de la persona influye en las indicaciones.

3.3.3. ANÁLISIS DE RESULTADOS DE LA PRIMERA EVALUACIÓN

La secuencia de rutas dentro del modo simulación no se pudo probar debido a que la prueba se la realizó fuera del segundo piso de la Facultad de Sistemas. Sin embargo, se comprobó que las instrucciones auditivas otorgadas según la descripción, van acorde a los requerimientos del usuario. El usuario nos supo manifestar que, en su percepción, se guía mejor mediante pasos que en metros. Aunque manifestó que la cantidad de información que se trasmite es demasiada como para poderla memorizar.

Se comprobó que tanto el modo simulación como navegación, deben estar disponibles en todo momento, debido a la dificultad que se presenta en cambiar de un modo a otro al confundir con el resto de opciones.

El usuario nos supo manifestar que durante la navegación hubo confusiones al no establecer un esquema de filas, es decir, que, al terminar de recorrer los elementos en una sola dirección, la navegación se detiene, en vez de continuar con el primer elemento de la siguiente fila.

Hubo confusión en algunas descripciones del lector de pantalla, sobre todo en los menús de tipo collapse. El usuario supo manifestar que se obtiene una mejor percepción cuando se tiene una lista desplegada de todos los elementos del prototipo.

Finalmente, el usuario dijo que el nivel de accesibilidad y usabilidad fue satisfactorio, el lector de pantalla le resultó novedoso y se mostró motivado a seguirlo utilizando para su trabajo. Recomendó que el único requerimiento para no obtener dificultades en la navegación, es familiarizarse con el prototipo.

En la Ilustración 3.23 se muestra una fotografía del usuario no vidente, Franklin Yaulu realizando las pruebas de accesibilidad sobre el prototipo.



Ilustración 3.23: Evidencia de usuario realizando las pruebas sobre el prototipo.

3.3.4. DATOS DEL SEGUNDO USUARIO.

Nombre de usuario: Nicholas Hoekstra.

Edad: 31 años.

Profesión: Asesor sobre Educación Inclusiva en el Ministerio Coordinador de Conocimiento y Talento Humano.

3.3.5. DATOS DE LA SEGUNDA EVALUACIÓN.

Fecha: 05/04/2016.

Hora de inicio: 9:30.

Hora de fin: 11:30.

Línea base:

- *Modelo del equipo:* Dell XPS One 27 Touch All-in-One PC.
- *Sistema operativo:* Windows 8.
- *Navegador:* Google Chrome v 51.0.
- *Lector de Pantalla:* Chrome Vox.
- *Tecnologías de asistencia:* Teclado y parlantes.

En la tabla 3.2 se puede observar el formato utilizado para las pruebas. Como elementos contiene el número de la tarea, descripción de la tarea, el grado de dificultad, si la tarea se ejecutó con ayuda y las observaciones.

Tabla 3.2: Pruebas de accesibilidad realizadas con Usuario 2.

No	Tarea	Grado de facilidad				¿Se ejecutó la tarea con ayuda?		Observaciones
		Fácil	Medio	Difícil	SI	NO		
1	Acceder al menú ayuda.		X			X	Hubo confusión con la sección de descripción del sitio web.	
2	Acceder al menú descripción.	X				X		
3	Acceder al menú indicaciones.		X			X	Se debe escuchar toda la información para entender las indicaciones	
4	Acceder a la oficina decanato mediante menú de búsqueda.		X		X		Se requiere de práctica para entender el funcionamiento del prototipo.	
5	Entender la información desplegada del elemento decanato.	X				X	La información es comprensible.	
6	Volver al plano, mediante el botón "Volver al mapa".		X			X		
7	Acceder a la oficina del Ing. Andrés Larco mediante búsqueda en el plano.		X			X		
8	Entender la información desplegada del elemento Andrés Larco.	X				X		
9	Retornar al menú de búsqueda mediante el botón "Volver al menú principal"	X				X		
10	Utilizar el atajo en teclado para acceder al menú de búsqueda.		X			X	Los atajos simplifican la navegación.	
11	Activar el atajo en teclado para salir.	X				X		
12	Acceder al menú accesibilidad.	X				X		
13	Activar la opción de simulación.		X		X			
14	Activar la ruta "Proyecto de investigación".	X			X			
15	Escuchar la información y llegar al elemento seleccionado.	X					El ritmo de paso de la persona influye en las indicaciones.	

3.3.6. ANÁLISIS DE RESULTADOS DE LA SEGUNDA EVALUACIÓN

En esta prueba se logró evaluar todas las rutas del modo simulación de manera satisfactoria. El usuario supo manifestar que era útil escuchar primero la ruta un par de veces y luego proceder a seguir las indicaciones. Además, se mostró satisfecho al haber incluido elementos fáciles de localizar como puntos de referencia, tales como puertas, pasillos y entradas principales.

El usuario indicó que las instrucciones deberían describirse en metros y no en pasos, debido a que es una medida estándar que la mayoría de personas pueden interpretar, sin embargo, mencionó que los pasos no tienen un rango de error muy alto y que podrían utilizarse sin problema. Además, mostró satisfacción al utilizar grados para indicar los giros que debe realizar para llegar a un elemento. En el prototipo, el usuario mencionó que sería de utilidad colocar un punto de referencia, ya que es difícil ubicar donde está el norte o el sur.

Indicó que las indicaciones de uso del prototipo albergan demasiada información textual, que no se puede interpretar ni recordar con facilidad. Sin embargo, se mostró satisfecho con los atajos de teclado al facilitar el acceso a varias opciones. Para ello, también dio a conocer que sería de utilidad configurar los atajos de teclado propios del lector de pantalla para simplificar aún más la navegación.

Recomendó implementar un esquema de filas, es decir, que cuando la navegación llega al límite de la pantalla, continuar con el elemento de la siguiente fila o columna.

El usuario quedó satisfecho con el resultado, incluso recomendó implementar el prototipo con otros tipos de plano, que serían de utilidad al moverse en entornos que requieren de familiarización.

En la Ilustración 3.24 se muestra una fotografía del usuario no vidente, Nicholas Hoekstra realizando las pruebas de accesibilidad sobre el prototipo.

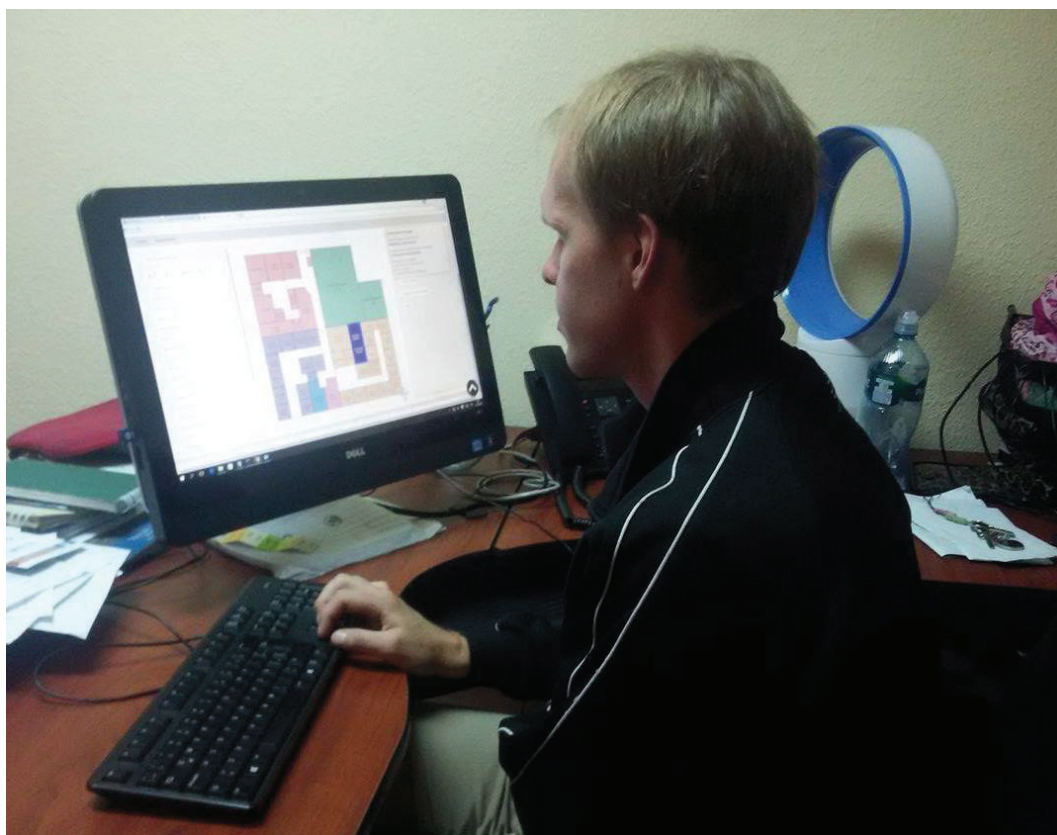


Ilustración 3.24: Evidencia de usuario realizando las pruebas sobre el prototipo.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Mediante el análisis realizado al sitio web de SETEDIS se pudo comprobar que algunas entidades gubernamentales de Ecuador están realizando implementaciones de accesibilidad tanto para discapacidades visuales, auditivas y motrices. De tal manera que se evidencia que el país está tomando la iniciativa de hacer cumplir la norma denominada "Tecnología de la información - Directrices de accesibilidad para el contenido web del W3C (WCAG) 2.0 (ISO/IEC 40500:2012, IDT)".
- El desarrollo del prototipo web de un plano interior accesible para personas no videntes, ha permitido conocer los requisitos de accesibilidad de las personas con discapacidad visual. De esta manera, se logró aplicar de manera exitosa las especificaciones de accesibilidad establecidas por las pautas WCAG 2.0, alcanzando un nivel de conformidad AA.
- El prototipo alcanzó el nivel de conformidad AA mediante el seguimiento de los estándares de accesibilidad W3C, que indican el uso correcto de HTML5, CSS3, JavaScript y SVG. De esta manera que se creó un código bien formado, estructurado y optimizado, para que cualquier tecnología de asistencia pueda interpretar la información transmitida y entregársela al usuario final.
- La metodología SCRUM permitió organizar al equipo de desarrollo en roles principales. Además, permitió definir de manera clara los requisitos del sistema que posteriormente se subdividieron en tareas agrupadas en tres iteraciones principales. Esto permitió llevar a cabo un desarrollo organizado y óptimo en donde se alcanzó de manera satisfactoria el producto final.

- Mediante la metodología SCRUM se pudo realizar un acoplamiento entre los principios de accesibilidad y requisitos de usuario, mediante historias de usuario y criterios de aceptación, para orientar el desarrollo de las tareas del prototipo a las recomendaciones de accesibilidad.
- Se han elaborado pruebas de accesibilidad utilizando herramientas validadoras para ver el grado de cumplimiento de las pautas WCAG 2.0 en el desarrollo del prototipo. De esta manera se pudo obtener una retroalimentación del cumplimiento de las pautas de accesibilidad que se aplicaron en el desarrollo del prototipo.
- Las herramientas simuladoras se utilizaron para verificar el despliegue del prototipo en caso de interactuar con un usuario con alguna discapacidad visual, de esta manera se puede entender como los elementos y la información se trasmite al usuario. Esto permite corregir el diseño del prototipo para mejorar la visualización del prototipo frente a personas con alguna discapacidad visual o asociada al color.
- Las pruebas realizadas con usuarios permitieron conocer el desempeño del prototipo en un ambiente real. Esto permitió comprobar como los atributos de accesibilidad, el correcto diseño del sitio web y las funciones, que, al ser utilizadas con lectores de pantalla y teclado, permiten al usuario interactuar con el prototipo, sin requerir de la vista para obtener una retroalimentación de la información transmitida.
- El modo simulación permitió conocer como el usuario no vidente interactúa con los elementos de un entorno desconocido y como se pueden integrar mejoras para ayudar al usuario a ubicar elementos de manera más sencilla. Las instrucciones dadas en el modo simulación se dictan en lenguaje entendible para el usuario, en este caso se utilizaron pasos y elementos de referencia para ubicar elementos, lo que fue de utilidad para el usuario que interactúa con un entorno real.

4.2. RECOMENDACIONES

- Para trabajos futuros se pueden utilizar medidas del sistema internacional para las medidas de distancia en el modo simulación. Al ser utilizadas en un ambiente interior, el rango de error no influye en la ubicación de elementos, sin embargo, las medidas internacionales mejoran el entendimiento de la distribución de los elementos del plano.
- Para futuras versiones, se puede implementar una solución en donde el prototipo se adapte a cualquier tipo de plano desarrollado según el ambiente en donde se opere. Es decir que no se requiera desarrollar un prototipo adaptado a un solo entorno, sino que este pueda adaptarse al medio en donde se encuentre el usuario.
- Para versiones posteriores se deben implementar mejoras en la adaptabilidad del prototipo, para que pueda ser desplegado en cualquier tipo de navegador. Además, el prototipo debe ser usado con cualquier lector de pantalla sin que esto afecte el desempeño del prototipo, para ello se debe utilizar versiones compatibles con formatos SVG y WAI-ARIA.
- Se deben realizar las pruebas con otros tipos de discapacidad visual, como retinopatía, cataratas, baja visión o daltonismo. Esto con el objetivo de incluir adaptaciones al prototipo para que este pueda adaptarse a cualquier tipo de discapacidad visual.
- En versiones posteriores se deben realizar pruebas con más personas no videntes, para obtener una retroalimentación más amplia y un resultado más preciso. De esta manera se puede comprender de mejor manera los problemas más comunes que enfrentan los usuarios no videntes al utilizar sitios web. De esta manera se puede añadir las implementaciones necesarias para lograr que el prototipo pueda ser utilizado por cualquier tipo de usuario.

- Para futuras versiones se considera implementar funciones en el prototipo que complementen otro tipo de discapacidades como auditivas o motrices. Para ello se debe implementar ayudas de navegación, simplificar la información transmitida e incluir una versión textual para las instrucciones del modo simulación.
- Se debe mejorar la navegación del prototipo para evitar elementos confusos y redundantes. Además, se debe implementar otros atajos de teclado vinculados a las funciones más relevantes del prototipo, para operar completamente el prototipo mediante el teclado.
- La información presentada en el prototipo debe ser simplificada para mejorar el entendimiento de las indicaciones de uso y las instrucciones de operación. Además, se debe incluir solo la información más relevante de cada elemento, para simplificar el aprendizaje del plano interior y no se requiera escucharla nuevamente.

BIBLIOGRAFÍA

- [1] W3C, «Guía Breve de Accesibilidad Web,» 2008. [En línea]. Available: <http://w3c.es/Divulgacion/GuiasBreves/Accesibilidad>. [Último acceso: 05 2016].
- [2] S. Luján Mora, «Ecuador ya tiene una norma sobre accesibilidad web,» 08 02 2014. [En línea]. Available: <http://accesibilidadenlaweb.blogspot.com/2014/02/ecuador-ya-tiene-una-norma-sobre.html>. [Último acceso: 05 05 2016].
- [3] The World Bank, «Disability Overview,» 04 04 2016. [En línea]. Available: <http://www.worldbank.org/en/topic/disability/overview>. [Último acceso: 05 05 2016].
- [4] World Health Organization, «Visual impairment and blindness,» 08 2014. [En línea]. Available: <http://www.who.int/mediacentre/factsheets/fs282/en/>. [Último acceso: 05 05 2016].
- [5] World Health Organization, «High-level Meeting on Disability and Development,» 23 11 2015. [En línea]. Available: <http://www.who.int/disabilities/hlm/en/>. [Último acceso: 05 2016].
- [6] Ministerio de Inclusión Económica y Social, «OBJETIVO ESTRATÉGICO,» [En línea]. Available: <http://www.inclusion.gob.ec/objetivo-estrategico/>. [Último acceso: 05 2016].
- [7] E. G. Pinos Velez, «METODO DE DETECCIÓN POR TONOMETRÍA PARA EL DIAGNÓSTICO TEMPRANO DEL GLAUCOMA A TRAVÉS DE HERRAMIENTAS DE SIMULACIÓN.,» de *Tecnologías de Inclusión*, Cuenca, 2013, p. 2.
- [8] Universidad de Alicante, «Accesibilidad Web,» [En línea]. Available: <http://accesibilidadweb.dlsi.ua.es/>. [Último acceso: 05 05 2016].
- [9] Société de transport de Montréal, «Metro Montreal,» [En línea]. Available: <http://www.stm.info/en/info/networks/metro>. [Último acceso: 05 2016].
- [10] R. Alcazar, «Prototipo accesible con Google Maps API V3,» [En línea]. Available: <http://www.rubenalcaraz.es/tfm/prototipo/>. [Último acceso: 05 2016].
- [11] R. Alcaraz Martínez y M. Ribera Turró, «Mapas digitales y aplicaciones basadas en la localización: mejoras en su accesibilidad para las personas ciegas,» [En línea]. Available:

- http://www.nosolousabilidad.com/articulos/mapas_digitales.htm. [Último acceso: 05 05 2016].
- [12] SETEDIS, «Secretaría Técnica para la Gestión Inclusiva en Discapacidades,» Gobierno Nacional de la Republica del Ecuador, [En línea]. Available: <http://www.setedis.gob.ec/>. [Último acceso: 05 05 2016].
- [13] W3C, «Web Content Accessibility Guidelines (WCAG) 2.0,» 11 12 2008. [En línea]. Available: <https://www.w3.org/TR/WCAG20/>. [Último acceso: 05 05 2016].
- [14] . M. James, «Scrum Reference Card,» de *Scrum Artifacts*, 2012, pp. 3-5.
- [15] J. Palacio, «Agilidad,» de *Gestión de proyectos Scrum Manager*, Rights info, 2014, pp. 14-16.
- [16] J. C. Cortizo Pérez, D. Expósito Gil y M. Ruiz Leyva, *eXtreme Programming*, 2013.
- [17] L. O. Amavizca Valdez, A. C. García Ruíz, E. Jiménez López, G. L. Duarte Guerrero y J. C. Vázquez Brindis, Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software: implementación y publicación de un sitio WEB para una empresa SPIN - OFF en el Sur de Sonora, México., Guayaquil, Ecuador: Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014), 2014.
- [18] T. Powell, «Core Markup,» de *HTML & CSS, The complete Reference*, McGrawHill, 2010, pp. 55-60.
- [19] W3Schools, «HTML and XHTML,» [En línea]. Available: http://www.w3schools.com/html/html_xhtml.asp. [Último acceso: 21 05 2016].
- [20] W3Schools, «HTML5 New Elements,» [En línea]. Available: http://www.w3schools.com/html/html5_new_elements.asp. [Último acceso: 21 05 2016].
- [21] J. Duckett, de *HTML & CSS : Design and Build Websites*, Indianapolis, John Wiley & Sons Inc, pp. 226-242.
- [22] W3Schools, «CSS3 Introduction,» [En línea]. Available: http://www.w3schools.com/css/css3_intro.asp. [Último acceso: 21 05 2016].
- [23] experto, «Introducción a CSS3: Nuevas Características, Capacidades y Recursos,» 25 11 2014. [En línea]. Available: <http://www.exp3rto.com/introduccion-a-css3-nuevas-caracteristicas-capacidades-y-recursos/>. [Último acceso: 21 05 2016].
- [24] M. Haverbeke, «Eloquent JavaScript,» de *A Modern Introduction to Programming*, 2014, pp. 6-8.

- [25] W3C, «What is SVG?,» 2013. [En línea]. Available: <https://www.w3.org/Graphics/SVG/>. [Último acceso: 05 05 2016].
- [26] Sails.js, «What Is Sails.js?,» [En línea]. Available: <http://sailsjs.org/>. [Último acceso: 05 05 2016].
- [27] K. V. Suaza, «Historias de usuario,» de *Definición de equivalencias entre historias de usuario y especificaciones en UN-LENCEP para el desarrollo ágil de software*, 2013.
- [28] PMOinformatica.com, «Plantillas Scrum: historias de usuario y criterios de aceptación,» 01 10 2012. [En línea]. Available: <http://www.pmoinformatica.com/2012/10/plantillas-scrum-historias-de-usuario.html>. [Último acceso: 05 16 2016].
- [29] A. Sharp, «Model View Controller,» de *Small Talk by example*, Berne, 2005, pp. 1-6.
- [30] B. McDonald, «Vector Paint,» 20 04 2015. [En línea]. Available: <http://vectorpaint.yaks.co.nz/>. [Último acceso: 2016 06 09].
- [31] W3C, «WAI-ARIA Overview,» 15 01 2016. [En línea]. Available: <https://www.w3.org/WAI/intro/aria.php>. [Último acceso: 2016 06 09].
- [32] jillix, «svg.pan-zoom.js,» 04 02 2016. [En línea]. Available: <http://jillix.github.io/svg.pan-zoom.js/>. [Último acceso: 10 06 2016].
- [33] M. Bostock, «Data-Driven Documents,» 2015. [En línea]. Available: <https://d3js.org/>. [Último acceso: 06 2016].
- [34] «Text to Speech,» 2015. [En línea]. Available: <http://www.fromtexttospeech.com/>. [Último acceso: 06 2016].
- [35] Centro Tecnológico de la Información y la Comunicación, «TAW,» [En línea]. Available: <http://www.tawdis.net/>. [Último acceso: 07 2016].
- [36] A. Leventhal, «NoCoffee – Vision Simulator for Chrome,» 09 02 2013. [En línea]. Available: <https://accessgarage.wordpress.com/>. [Último acceso: 07 2016].
- [37] University of Cambridge, «Impairment Simulator,» [En línea]. Available: <http://www.inclusivedesign toolkit.com/betterdesign2/simsoftware/simsoftware.html>. [Último acceso: 07 2016].
- [38] W3C, «Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0,» 11 12 2008. [En línea]. Available: <http://www.sidar.org/traduccion/wcag20/es/>. [Último acceso: 05 05 2016].

- [39] W3C, «World Wide Web Consortium (W3C),» [En línea]. Available: <https://www.w3.org/>. [Último acceso: 05 05 2016].
- [40] Scrum Manager, Safe Creative, 2011.
- [41] Government of Sweden, «Government Offices of Sweden,» [En línea]. Available: <http://www.government.se/>. [Último acceso: 5 05 2015].

GLOSARIO

Accesibilidad web: El objetivo de la accesibilidad web es permitir que sitios web sean utilizados de manera universal, independientemente de las capacidades o conocimientos de los usuarios.

AJAX: Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una forma de desarrollar contenido web para elaborar aplicaciones dinámicas.

Alt: Es un atributo de HTML usado para agregar un texto alternativo a un gráfico.

Bootstrap: Es un framework usado para maquetar sitios web responsivos mediante CSS, HTML y JavaScript.

Examinator: Es una herramienta on-line para verificar el nivel de accesibilidad de un sitio web en base a las pautas de accesibilidad WCAG 2.0.

JavaScript: Es un lenguaje de programación interpretado en el lado del cliente para elaboración de sitios web dinámicos.

JQuery: Es una librería de JavaScript que se usa para interactuar con archivos HTML y elaborar animaciones y páginas web interactivas.

JSON: JavaScript Object Notation (Notación de objetos JavaScript), es un formato de texto utilizado para intercambiar datos.

Product backlog: Es un listado de las funcionalidades que serán implementadas en el desarrollo del producto.

Sprint: Ciclo o iteración en donde se diseña, codifica y prueba el producto.

Sprint backlog: Es un listado que integra las desintegra las funcionalidades del product backlog en las actividades que serán desarrolladas en el sprint.

SVG: Scalable Vector Graphics (Gráficos de vectores escalables), es un lenguaje vectorial en dos dimensiones para crear gráficos vectoriales, gráficos de trama y el texto en formato XML.

TabIndex: Es un atributo utilizado para definir un orden de navegación entre elementos HTML.

TAW: Es un analizador de sitios web en donde mide el grado de accesibilidad en base a las pautas de Accesibilidad, generando un informe de los resultados.

Title: Es un atributo HTML para añadir información complementaria o un tooltip a un elemento.

W3C: World Wide Web Consortium (Consortio de la red mundial), es la entidad internacional instaurada para fomentar el uso de estándares para el desarrollo de la World Wide Web.

WAI-ARIA: Accessible Rich Internet Applications (Aplicaciones de Internet accesibles y dinámicas), es una iniciativa de la W3C que define una forma de elaborar contenido web accesible enfocado en tecnologías de contenido dinámico e interfaces de usuario como JavaScript y HTML.

WCAG 2.0: Web Content Accessibility Guidelines (Directrices de accesibilidad de contenido web), detallan cómo hacer páginas web accesibles para personas con discapacidad, éstas directrices están dirigidas a desarrolladores web, con el objetivo de hacer el contenido accesible.

ANEXOS

Anexo I. Plantillas de Scrum (formato Excel).