



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
LICENCIATURA EN CIENCIAS COMPUTACIONALES

TESIS

**Desarrollo de una extensión para navegador web
como apoyo a personas con daltonismo: DaltonAssist**

Para obtener el título de
Licenciado en Ciencias Computacionales

PRESENTA
PDLCC. Diego López Niembro

Directora
MID. Norma Laura Salazar Viveros

Comité tutorial
Dra. Verónica Martínez Lazcano
MID. Norma Laura Salazar Viveros
Mtra. María del Carmen Vera Carranza
Dr. Manuel Alejandro Ojeda Misses

Pachuca de Soto, Hgo., México, enero, 2026



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences
Dirección
Office of the Director

Mineral de la Reforma, Hgo., a 28 de abril de 2026

Número de control: ICBI-D/865/2026

Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio, le comunico que el Jurado asignado al egresado de la Licenciatura en Ciencias Computacionales **Diego López Niembro**, quien presenta el trabajo de titulación **"Desarrollo de una extensión para navegador web como apoyo a personas con daltonismo: DaltonAssist"**, ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: Dra. Verónica Martínez Lazcano

Secretario: M.T.E. María del Carmen Vera Carranza

Vocal: MID. Norma Laura Salazar Viveros

Suplente: Dr. Manuel Alejandro Ojeda Misses

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
"Amor, Orden y Progreso"

Dr. Gabriel Vergara Rodríguez
Director del ICBI

GVR/MMM

"Amor, Orden y Progreso"



Ciudad del Conocimiento, Carretera Pachuca-Tulancingo
Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo,
México. C.P 42184

Teléfono: 52 (771) 71 720 00 Ext. 40001

direccion_icbi@uaeh.edu.mx, vergara@uaeh.edu.mx

uaeh.edu.mx

Agradecimientos

A mi mamá, por no solo el amor que me ha demostrado en todo este tiempo y por las veces que creí que no me podía levantar ella estuvo ahí, por todo lo que tuvo que sacrificar y soportar por mí, te amo jefa. Gracias por siempre ser esa estrella que me guía en las noches oscuras.

A mi hermana, por todo lo que me hizo aprender y me apoyo para las cosas que desconocía y que me resultaban complicadas y siempre darme un consejo y amor, te amo carnala.

Las amo con todo mi corazón y estoy agradecido de poder contar con personas tan increíbles y amorosas, este logro no solo es mío, es de todos. Gracias por recorrer este camino conmigo.

A Lexi, por todo el amor, paciencia y oportunidades de ser mejor que me has brindado mi amor. Gracias por ser mi confidente y mi empujón en todo lo que me propongo te amo.

A mis amigos de años Alonso, Abraham, Rodrigo, Daniel, Adrián y Mau por ser siempre una salida de la realidad mientras platicábamos y decíamos cualquier cosa para distraernos. Gracias por siempre ser el oído y consejo que necesito los quiero.

A mis amigos de la universidad Diego, Ian, Daniel, Francisco, Ale y Ana, gracias por todo lo que compartimos en este proceso tan complicado y corto, en el que nos reímos, lloramos, nos escuchamos y salimos adelante ante cualquier situación.

A mi asesora, Norma, por ser la enseñanza, paciencia y esfuerzo que ocupo en mí. Gracias por guiarme por encima de los obstáculos que se encontraron y siempre estar presente ante cualquier duda, comentario o situación personal en la que la necesitaba, la quiero porfa.

Por último, a todas aquellas personas que de forma directa o indirecta participaron en este camino, sin ustedes este logro sería imposible, gracias por el apoyo y aprecio en este viaje.

Resumen

El presente proyecto “Desarrollo de una extensión para navegador web como apoyo a personas con daltonismo: DaltonAssist” tiene como objetivo diseñar, desarrollar e implementar una extensión de navegador, enfocada en solucionar la pérdida de experiencia e información que presentan los usuarios con daltonismo al navegar en la *web*.

La extensión va enfocada a personas que tienen daltonismo, que experimenten una pérdida de información o experiencia visual al momento de navegar dentro de un sitio *web*.

Para el desarrollo se hizo uso de la metodología *Kanban*, por medio de ocho *sprints* haciendo uso de las fases de planificación, limitación del trabajo en proceso, ciclo de desarrollo y mejora continua. Se empleó el uso de tecnologías como *Astro*, *JS*, *HTML*, *CSS* y *Tailwind* para la creación de interfaces y el funcionamiento de la extensión.

Los resultados muestran que la extensión es útil para los usuarios con daltonismo que buscan una alternativa a la navegación *web*. Finalmente, DaltonAssist es una herramienta efectiva y accesible, que apoya a las personas con daltonismo y da opciones a una navegación más accesible.

Abstract

The present project “Development of a web browser extension to support people with color blindness: DaltonAssist” aims to design, develop, and implement a browser extension, focused on solving the loss of experience and information presented by users with color blindness when navigating the web.

The extension is focused on people who have color blindness, who experience a loss of information or visual experience when navigating within a website.

For the development, use was made of the Kanban methodology, through eight sprints using the phases of planning, limitation of work in process, development cycle, and continuous improvement. The use of technologies such as Astro, JS, HTML, CSS, and Tailwind was employed for the creation of interfaces and the operation of the extension.

The results show that the extension is useful for users with color blindness who are looking for an alternative to web navigation. Finally, DaltonAssist is an effective and accessible tool, which supports people with color blindness and gives options for a more accessible navigation.

Contenido

Introducción	1
Antecedentes	2
Planteamiento del problema	3
Propuesta de solución	3
Justificación	4
Objetivo general	5
Objetivos específicos	5
Alcance	5
Limitaciones	6
Capítulo 1. Fundamentación teórica	7
1.1 Marco Teórico	7
1.1.1 WCAG	7
1.2 Marco conceptual	10
1.2.1 Daltonismo	10
1.2.2 Internet	13
1.2.3 Páginas <i>web</i>	16
1.2.4 Diseño <i>web</i>	18
1.3 Marco tecnológico	19
1.3.1 Chrome Extensions	19
1.3.2 Figma	20
1.3.3 Firebase	21
1.4 Marco metodológico	22
1.4.1 <i>Scrum</i>	23
1.4.2 <i>Kanban</i>	25
1.4.3 Desarrollo guiado por pruebas (<i>TDD</i>)	29
1.5 Estado del arte	30
1.5.1 Dalton (<i>Colorblind Extension</i>)	30
1.5.2 <i>SightGlow</i>	31
1.5.3 <i>NoCoffee – Vision Simulator</i>	32
1.5.4 <i>Colorblindly</i>	33
Capítulo 2. Desarrollo	37
2.1 Planificación	37
2.1.1 Creación de las historias de usuario	37

2.1.2 Definición de las tareas del proyecto	39
2.1.3 Creación del tablero <i>Kanban</i>	39
2.2 Limitaciones del proceso (<i>WIP</i>).....	43
2.3 Ciclo de desarrollo.....	43
2.4 Mejora continua	66
Capítulo 3: Resultados del test de Ishihara	68
3.1 Fundamentos del cuestionario	68
3.2 Resultados del test de Ishihara	68
Capítulo 4: Validación	79
4.1 Objetivo de la validación	79
4.2 Justificación del instrumento.....	79
4.3 Procedimiento de aplicación.....	80
4.4 Resultados obtenidos.....	80
4.5 Análisis crítico	83
Conclusiones	83
Trabajos futuros.....	84
Apéndice 1. Preguntas del test Ishihara	85
Anexo 1: Tabla de los 78 criterios de WCAG 2.2.....	88
Anexo 2: Test Ishihara	91
Referencias.....	97

Índice de figuras

Capítulo 1

Figura 1.1 Ratio de proporción.....	8
Figura 1.2 Mensaje de error poco específico	9
Figura 1.3 Mensaje de error.....	10
Figura 1.4 Imágenes de muestra de visión con tritanopia.....	12
Figura 1.5 Imágenes de muestra de visión con Deuteranopia	12
Figura 1.6 Imágenes de muestra de visión con protanopia.....	13
Figura 1.7 Tablero Kanban por Sonya Siderova.....	26
Figura 1.8 Interfaz de Colorblind.....	31
Figura 1.9 Interfaz de SightGlow	32
Figura 1.10 Vista previa del filtro de NoCoffee	33
Figura 1.11 Interfaz de la extensión Colorblindly.....	34

Capítulo 2

Figura 2.1 Backlog del proyecto DaltonAssist.....	39
Figura 2.2 Pestaña de creación de tableros en Jira	40
Figura 2.3 Pestaña de selección de plantillas de Jira	40
Figura 2.4 Pestaña de selección de tablero Kanban.....	41
Figura 2.5 Pestaña de elección de tipo de espació.....	41
Figura 2.6 Pestaña de Creación de Proyecto.....	42
Figura 2.7 Pestaña de Añadir Participantes.....	42
Figura 2.8 Tablero de DaltonAssist en Jira.....	43
Figura 2.9 Flujo de datos del sprint-1.....	44
Figura 2.10 Tarea desplegada.....	44
Figura 2.11 Flujo de datos del sprint-2.....	45
Figura 2.12 Página de designación de nombre al proyecto.....	45
Figura 2.13 Página de activación de Google Analytics.....	46
Figura 2.14 Página de proyectos creados en Firebase.....	47
Figura 2.15 Layout básico del proyecto en Firebase.....	47
Figura 2.16 Interfaz de elección de tipo de aplicación.....	48
Figura 2.17 Página de asignación de nombre a la aplicación web.....	48
Figura 2.18 Ejemplo de configuración de las llaves de Firebase.....	49
Figura 2.19 Flujo de datos del sprint-3.....	50
Figura 2.20 Diseño de la interfaz de la extensión.....	51
Figura 2.21 Archivo JSON.....	51
Figura 2.22 Fragmento de código del archivo HTML.....	52
Figura 2.23 Flujo de trabajo Sprint-4.....	52
Figura 2.24 Ejemplo de testeo y corrección de errores.....	53
Figura 2.25 Flujo de trabajo Sprint-5.....	53
Figura 2.26 Función de aplicar filtro de Deuteranopía.....	54
Figura 2.27 Flujo de trabajo Sprint-6.....	54
Figura 2.28 Pestaña de gestión de extensiones.....	55
Figura 2.29 Interfaz de la extensión en desplegada.....	55
Figura 2.30 Primer flujo de trabajo Sprint-7.....	56

Figura 2.31 Diseño en Figma del test.....	57
Figura 2.32 Fragmento de código del archivo Index.	57
Figura 2.33 Fragmento de código del archivo test.js.....	58
Figura 2.34 Fragmento de código del archivo configuración.js.	58
Figura 2.35 Dashboard de EmailJS.....	59
Figura 2.36 Ventana modal de selección de servicio y configuración del servicio Gmail.	59
Figura 2.37 Pestaña de configuración de las platillas.	60
Figura 2.38 Fragmento de código del archivo configuración.js.	60
Figura 2.39 Segundo flujo de trabajo Sprint-7.	61
Figura 2.40 Icono de manejo de versiones de git en VSC.....	61
Figura 2.41 Pestaña de Source Control dentro de VSC.	62
Figura 2.42 Tercer flujo de trabajo Sprint-7.	62
Figura 2.43 Línea de consola con el comando de ejecución de del entorno de pruebas.	63
Figura 2.44 Primer flujo de trabajo Sprint-8.	63
Figura 2.45 Dashboard de administración de hosting de Firebase.....	64
Figura 2.46 Ejemplo de consola de ejecución de firebase init.....	64
Figura 2.47 Fragmento de código del archivo firebase.json.	65
Figura 2.48 Fragmento de consola de ejecución del comando firebase deploy.....	65
Figura 2.49 Segundo flujo de trabajo Sprint-8.	66

Capítulo 3

Figura 3.1 Gráfica de resultados de la lámina 1 del test de Ishihara.	68
Figura 3.2 Gráfica de resultados de la lámina 2 del test de Ishihara.	69
Figura 3.3 Gráfica de resultados de la lámina 3 del test de Ishihara.	69
Figura 3.4 Gráfica de resultados de la lámina 4 del test de Ishihara.	70
Figura 3.5 Gráfica de resultados de la lámina 5 del test de Ishihara.	70
Figura 3.6 Gráfica de resultados de la lámina 6 del test de Ishihara.	70
Figura 3.7 Gráfica de resultados de la lámina 7 del test de Ishihara.	71
Figura 3.8 Gráfica de resultados de la lámina 8 del test de Ishihara.	71
Figura 3.9 Gráfica de resultados de la lámina 9 del test de Ishihara.	72
Figura 3.10 Gráfica de resultados de la lámina 10 del test de Ishihara.	72
Figura 3.11 Gráfica de resultados de la lámina 11 del test de Ishihara.	72
Figura 3.12 Gráfica de resultados de la lámina 12 del test de Ishihara.	73
Figura 3.13 Gráfica de resultados de la lámina 13 del test de Ishihara.	73
Figura 3.14 Gráfica de resultados de la lámina 14 del test de Ishihara.	74
Figura 3.15 Gráfica de resultados de la lámina 15 del test de Ishihara.	74
Figura 3.16 Gráfica de resultados de la lámina 16 del test de Ishihara.	74
Figura 3.17 Gráfica de resultados de la lámina 17 del test de Ishihara.	75
Figura 3.18 Gráfica de resultados de la lámina 18 del test de Ishihara.	75
Figura 3.19 Gráfica de resultados de la lámina 20 del test de Ishihara.	76
Figura 3.20 Gráfica de resultados de la lámina 20 del test de Ishihara.	76
Figura 3.21 Gráfica de resultados de la lámina 21 del test de Ishihara.	76
Figura 3.22 Gráfica de resultados de la lámina 22 del test de Ishihara.	77
Figura 3.23 Gráfica de resultados de la lámina 23 del test de Ishihara.	77

Figura 3.24 Gráfica de resultados de la lámina 24 del test de Ishihara.	78
--	----

Capítulo 4

Figura 4.1 Ejemplo de escala diferencial para UEQ.....	79
Figura 4.2 Grafica de la tabla promedios por cada dimisión de UEQ.	82

Apéndice 1

Figura A1.1 Cards del Test Ishihara de la lámina 1 a la 4.	85
Figura A1.2 Cards del Test Ishihara de la lámina 5 a la 8.	86
Figura A1.3 Cards del test Ishihara de la lámina 9 a la 12.	86
Figura A1.4 Cards del Test Ishihara de la lámina 13 a la 16.	86
Figura A1.5 Cards del Test Ishihara de la lámina 17 a la 20.	87
Figura A1.6 Cards del Test Ishihara de la lámina 21 a la 24.	87

Índice de Tablas

Capítulo 1

Tabla 1.1 Cuadro comparativo de metodologías	30
--	----

Capítulo 2

Tabla 2.1 Tabla de historias de usuario	37
---	----

Capítulo 4

Tabla 4.1 Promedios por cada una de las preguntas.....	81
Tabla 4.2 Promedios por cada dimisión de UEQ.....	82

Anexo 1

Tabla A-nexo 1.1 Tabla de criterios WCAG 2.2.....	88
---	----

Anexo 2

Tabla A-nexo 2.1 Tabla de Laminas del test Ishihara.	91
---	----

Introducción

Lo que el ser humano percibe con la vista suele ser muy subjetivo dado que no todos aprecian el mundo de la misma manera o matices refiriéndose a los colores, tomando esto en cuenta, existe una afección visual conocida como daltonismo que implica la forma en la que se observan los colores.

El daltonismo se expresa como una discapacidad cromática en la visión dando paso a una discapacidad en lo que a la vista se refiere, ésta puede representar alto o bajo riesgo dependiendo en qué ambiente se aplique (Awareness, About Colour Blindness, 2022).

Desde el principio de la historia de la humanidad ha existido el daltonismo, solo que no fue hasta finales del siglo XVIII que un químico llamado John Dalton el cual, mientras realizaba una actividad no pudo diferenciar entre un tono de color y otro lo que fue el origen de su investigación, llevando a describir lo que se conoce hoy en día como Daltonismo (Lanthony, 2013).

Si bien existen diversas herramientas, las cuales ofrecen una mejor noción acerca de los colores; como el nombre del color en las etiquetas del producto que se adquiere o un cuentagotas que indique el color, aun con todo esto, dentro de la visualización del contenido dentro de un navegador, no existe ninguna herramienta que tome en cuenta a los usuarios que padecen daltonismo.

Haciendo referencia a lo mencionado con anterioridad, es importante atender a los usuarios con esta afección que realizan una búsqueda en internet y no son considerados al momento del desarrollo de los sitios *web*, debido a que los implicados en el desarrollo tienen un desconocimiento o una falta de formación en accesibilidad, cuando se diseña el sitio *web* (Rodriguez, 2024).

Debido a todo ello, se ha tomado la decisión de desarrollar una herramienta para ayudar a los usuarios que padecen daltonismo y permitirles personalizar el navegador *web* por medio de un módulo de software habitualmente conocido como *extensión de navegador*.

Dicho módulo aplicará un filtro a los sitios *web* para que el contenido y, por consiguiente, la experiencia que éstos proporcionen, no sea poco apreciada por los usuarios con esta afección y se les conceda un ambiente más acorde a sus necesidades.

La extensión es compatible con los navegadores más populares del mercado, permite al usuario activarla según sus necesidades. Debido a su bajo impacto en el sistema, su ejecución no resulta invasiva, evitando así comprometer el rendimiento de los sitios *web* consultados, mejorando la experiencia visual del usuario.

Antecedentes

Los estudios acerca de cómo las personas interpretan los colores que ven, han creado diversas posturas acerca de cómo éstos se aprecian. Una de ellas es el estudio de la discromatopsia, mayormente conocido como Daltonismo, este no había sido diagnosticado debido a lo subjetivo que resulta la percepción del color de cada persona (Avanzada, 2024). El daltonismo tuvo sus primeras investigaciones relevantes a finales del siglo XVIII, por lo que en épocas anteriores no se cuenta con los antecedentes de esta afección.

El estudio del daltonismo dio el paso crucial en 1793, cuando el químico inglés John Dalton, mientras realizaba un encargo dado por su madre, el cual constaba en comprar unas calcetas en un tono de color rojo, Dalton llevó una tonalidad diferente, solo que él no notaba la diferencia. Esto lo llevo a preguntarse ¿por qué él los veía de igual tono y su madre no? y lo motivó a realizar la investigación de la discromatopsia (Lanthony, 2013).

John Dalton no tenía experiencia en este tema, pero aun así tras muchos estudios, comprobaciones e investigaciones descubrió que hay personas que no pueden ver los colores de la misma forma que los demás; ya que, dentro de la pupila hay diferentes células, llamadas conos y bastones, que permiten percibir los colores dependiendo de la cantidad de luz que rebote en el objeto, por lo tanto, hay ciertos colores o pigmentaciones que las personas con daltonismo no pueden ver (Lanthony, 2013).

Sin tener conocimiento de la investigación de Dalton, John Herschel un químico inglés, se involucró en el estudio de la discromatopsia conocida normalmente como daltonismo, el cual consistía en realizar pruebas de luz, así como de identificación de colores con dos sujetos, uno con visión normal y el otro con daltonismo. Este estudio, lo llevó a la conclusión de que las personas que padecen de esta afección poseen visión dicrómata refiriéndose a que los ven dos colores cuando las personas que no lo poseen normalmente apreciarían tres (Lanthony, 2013).

Como lo afirma el novelista alemán Johann Wolfgang Von Goethe, el cual, en colaboración con Johann Christoph Friedrich Schiller, se involucraron en el estudio del daltonismo al incluir a dos de sus estudiantes en un test, que prueba igualar los colores que se encontraban pintados en una superficie, descubriendo que uno de ellos no podía reproducir con normalidad los tonos azules (Lanthony, 2013).

En la investigación titulada *Usabilidad web para usuarios daltónicos*, Quispe (2024) presentadora en el V Congreso Iberoamericano Soporte del Conocimiento con la Tecnología (SOCOTE), analizó la usabilidad *web* para daltónicos y determinó que se deberían de realizar correcciones al sitio *web* haciendo uso de las reglas que propone la W3C para diseñar *Webs* accesibles a personas con deficiencias a la percepción del color, debido a que las personas con daltonismo llegan a presentar dificultades en su navegación en *webs* que no toman en cuenta estas reglas.

En la actualidad, debido a que la población se vuelve más y más incluyente, el entorno de las tecnologías para la integración de las personas con alguna discapacidad se encuentra en su apogeo.

Debido a este apogeo, nacen organizaciones sin fines de lucro, las cuales crean conciencia acerca de las alteraciones en la población. En el caso del daltonismo existe una organización llamada *Colour Blind Awareness*, la cual proporciona los datos de que 1 de cada 12 hombres (8%) y 1 de cada 200 mujeres (0.005%) tienen esta afección visual, y a nivel mundial, se estima que cerca de 300 millones de personas tienen daltonismo (Awareness, About Colour Blindness, 2022).

Planteamiento del problema

Si bien existen muchos problemas asociados a las deficiencias visuales que puede traer consigo el daltonismo, como lo es el simple uso de las señales de tránsito o de los semáforos. La dificultad de poder identificar cuando es momento de pasar debido a que las personas con los tipos de daltonismo más comunes como lo son la ceguera al rojo (Deuteranipia) o la ceguera al verde (Protanopia) pueden llegar a confundirse ya que el semáforo usa esos colores para determinar una indicación de “alto” y “siga”, su confusión puede derivar en situaciones de alto riesgo y accidentes viales de gravedad.

El daltonismo representa un desafío en el desarrollo tecnológico, ya que los desarrolladores deben integrar mecanismos de accesibilidad para garantizar una experiencia de usuario completa. La navegación *web* puede presentar barreras para el usuario debido al uso inadecuado de contrastes y paletas de colores en los sitios puede provocar una pérdida de información o una experiencia de usuario deficiente durante proceso de navegación.

Por lo tanto, se determina que el daltonismo constituye una barrera significativa en la navegación digital. Dependiendo del diseño y la estructura visual del sitio *web*, el usuario puede tener problemas para ver ciertos contenidos, lo que genera confusión y pérdida de información, haciendo que el sitio sea poco accesible para una población ignorada como lo son las personas con daltonismo.

Propuesta de solución

Para dar solución a la problemática presentada anteriormente, se propone un sistema integral que comienza con la aplicación de un test mediante el cual es posible determinar el tipo de daltonismo que presenta el usuario. Tras recopilar el resultado es enviado automáticamente al correo electrónico del interesado para su consulta y registro personal.

Una vez determinado el tipo de daltonismo, con el fin de mitigar las barreras de navegación mencionadas en el apartado anterior, se plantea el presente proyecto que se avoca al desarrollo de una extensión para navegadores *web* cuyo propósito es intervenir dinámicamente la página en la que el usuario se encuentra navegando, ajustando los elementos visuales para asegurar que la información sea accesible y clara.

Para tener un registro de esto, se implementa una base de datos que permite almacenar la información de los usuarios, con dos propósitos, siendo el primero gestionar el envío de los resultados por correo electrónico y el segundo posteriormente, servir como fuente para realizar análisis estadísticos, generando graficas para el análisis de los datos.

Por otra parte, el sistema está diseñado para cambiar el aspecto visual de las páginas de forma automática, de manera similar a como funciona un “modo oscuro”. Esto permite ajustar las tonalidades del sitio hacia gamas de colores que sean fáciles de distinguir para personas con daltonismo, asegurando que el contenido sea siempre legible y evitando que la estructura del sitio sea un obstáculo para el usuario.

Justificación

Para los ojos de una persona con daltonismo es difícil reconocer que tipo de color realmente se aprecia, no es hasta que otra persona debate con ella y pregunta “¿Qué color es?”, que se empieza a plantear el color que se visualiza.

El usuario, al realizar una búsqueda en un navegador y dentro de los sitios *web*, no tiene a la otra persona que le plantea el color que ve, por lo tanto, esto causa un problema de accesibilidad en los sitios *web*, llegando a tener una pérdida de información o de la experiencia que ofrece el sitio al usuario.

El propósito de la extensión es que los individuos que presenten esta afección visual no tengan complicaciones al navegar por internet, por lo que los beneficios para este sector de población concretamente serían grandiosos, ya que implementar el cambio de colores sería más útil a la vista de los usuarios y, gracias a este módulo de software, no se perderían de la experiencia que puede ofrecer el sitio *web*, así como de toda la información contenida en él, al igual que la comprensión de ésta,

El tomar en cuenta esta problemática de visión de las personas con la afectación, permitiría que los desarrolladores volteen a ver a los usuarios con esta discapacidad y de esta manera modificar el nivel de diseño de los sitios *web*. El problema también se resolvería llevando a cabo mejoras en la accesibilidad al crear sitios *web* que consideren a ese sector de la población, si bien las desarrolladoras, así como las empresas, incluso el mismo usuario al usar ya sea lentes para daltonismo o implementando todo el filtro dentro de la computadora podría corregir el problema.

Por otra parte, sería muy útil que los diseñadores *web* creen una paleta de colores de prueba al implementar el sitio *web*, atendiendo el espectro de daltonismo para aquellos usuarios que tienen problema en distinguir un solo color, así como para los usuarios que se enfrentan a no distinguir diferentes colores.

Se puede decir que las extensiones que implementan el forzado de estilos también implementen uno a fin de cada uno de los tres tipos de daltonismo que predominan, y de esta manera volverse más incluyentes con este sector de población.

La razón para la realización del módulo de software es que las personas que se encuentran dentro del rango de población que llegan a tener daltonismo ya sea en menor o mayor grado, puedan identificar la información, así como la experiencia, sin llegar a perder detalle dentro del sitio *web* y así identificar todos los componentes dentro de éste.

Objetivo general

Desarrollar una extensión para navegador *web*, haciendo uso de la API de Chrome Extensions, con la finalidad de que los usuarios con daltonismo apliquen el filtro más acorde a sus necesidades, lo cual les permitirá mejorar su experiencia al navegar en la *web*.

Objetivos específicos

Para alcanzar el objetivo general propuesto y asegurar la viabilidad técnica y funcional de la solución, se han definido los siguientes objetivos específicos, los cuales detallan las fases de desarrollo, implementación y validación del sistema.

1. Desarrollar el test *Ishihara* de forma digital dentro de un sitio *web*, por medio del *framework Astro*, permitiendo que el usuario pueda saber qué tipo de daltonismo tiene.
2. Diseñar y programar un sitio *web* en el cual se pueda alojar el formulario de recolección de datos del usuario y el test de daltonismo donde ambos estén relacionados con una base de datos y el envío a correo electrónico.
3. Construir la extensión de navegador utilizando la *API* de *Chrome Extensions*, permitiendo que el usuario elija el filtro mejor enfocado a sus necesidades.
4. Realizar pruebas de usabilidad con usuarios daltónicos para evaluar la efectividad de los filtros aplicados y ajustar el algoritmo según los resultados.

Alcance

La creación de la extensión propone que sin importar el dispositivo en el que se esté usando, se pueda navegar en la *web* sin mayores complicaciones para los usuarios daltónicos, así mismo en un futuro pueda ser usado por los desarrolladores al momento de elegir paletas de colores para este sector de la población.

De igual manera, dar la disponibilidad a la experimentación de las paletas de colores sobre los filtros, para un desarrollo de un sitio *web* incluyente con las y los usuarios con daltonismo y tomando en cuenta el tipo de afectación de cada uno de ellos, viendo a futuro el desarrollo de software incluyente y accesible.

Dar apertura a que las personas sean más conscientes de los problemas que puede llegar a tener estos individuos, así del modo en el cual observan su entorno y el cómo llegar a apoyarlos y tomarlos en cuenta, inclusive con la finalidad de no discriminarlos.

La extensión realiza un cambio de estilo en los colores de la página por medio de una inyección de *CSS*, permite al usuario elegir entre tres tipos diferentes de inyección uno por cada uno de los tipos de daltonismo, así mismo al descargarla proporciona un test de daltonismo para determinar en dónde se encuentra el usuario el cuestionario hace uso de dos *Apis*, una para enviar los resultados por correo electrónico y la otra para guardar los resultados en una base de datos.

Limitaciones

La construcción de la extensión se encuentra condicionada por diversos factores críticos, asociados principalmente a las necesidades específicas del público objetivo y a las particularidades del ecosistema donde opera.

A pesar de que la *API* de *Chrome Extensions* ofrece diferentes opciones de compatibilidad, navegadores como Safari y *Mozilla Firefox* no pueden aprovechar plenamente sus beneficios debido a que emplean sistemas de extensiones distintos. Esta diferencia limita su compatibilidad y en consecuencia, los usuarios de estos navegadores se ven privados de funcionalidades avanzadas que ya están disponibles en Chrome.

El uso de la extensión de cambio de estilo puede afectar el rendimiento de páginas con muchas animaciones, así como de los videos a menos que ellos lo patenten, por ejemplo, *YouTube* no permite el intercambio de estilos sobre el video ya que disminuye la velocidad de reproducción de video y su calidad, tomando en cuenta las restricciones que llegan a imponer los sitios *web* a las extensiones, por políticas del sitio.

Capítulo 1. Fundamentación teórica

En este capítulo se presentan los términos de mayor importancia dentro del proyecto, así como la manera en que el daltonismo se involucra en la sociedad y la tecnología como punto relevante para esta, se tomarán en cuenta.

1.1 Marco Teórico

Este apartado examina los principios y criterios de las *WCAG*, estableciendo los estándares de accesibilidad que rigen el diseño de interfaces inclusivas para usuarios con deficiencias visuales.

1.1.1 WCAG

El tema principal por discutir es el uso de las diversas pautas en internet en las que no se toma en cuenta a las personas con daltonismo, a las cuales va dirigida la extensión que se plantea en este proyecto.

En el ámbito del desarrollo, existe información acerca de cómo se debería de ver o parametrizar las páginas *web* para que se vean más estéticas o llamen más la atención sin embargo la *W3C* creó las *WCAG* (*Web Content Accessibility Guidelines*) en español el modelo de accesibilidad *web*, el cual establece si un sitio *web* es accesible o no, tomando cuatro principios:

- Perceptible
- Operable
- Comprensible
- Robusta.

Cada uno de los principios mencionados anteriormente se dividen entre 13 pautas generales y 78 criterios de conformidad específicos, así, estableciendo 3 niveles de conformidad A, AA, AAA. De los 78 criterios 30 son de nivel A, 20 de nivel AA y 28 de nivel AAA, a continuación, se explican los 3 niveles de conformidad existentes:

- Nivel A: Este envuelve a los criterios que son indispensables cuando se solicita acceso.
- Nivel AA: Se incluyen los criterios que tratan la legislación de los sitios.
- Nivel AAA: En este se involucran los criterios cuya finalidad es mejorar la usabilidad y el aumentar la cantidad de usuarios que acceden al sitio *web*.

Es importante mencionar que si no se cumple con los criterios anteriores no es posible subir de nivel, es decir, que para un nivel A se deben de cumplir los 30 criterios del nivel A y para ser nivel AA se deben cumplir los 30 criterios del nivel A y los 20 del nivel AA, del mismo modo para el nivel AAA (WAI, 2023).

Tomando en cuenta los tres niveles mencionados se identifican en dos campos, donde el primero se describe como la información se muestra legible y el segundo detalla la armonía que tenga un color correcto, al segundo se le dará más peso, debido a que influye en primero:

- Información Legible

Aquí se hace uso del primer principio el cual es: perceptible, basándose en que los elementos sean distinguibles del primer plano con respecto del fondo.

El ratio de contraste se enfoca en cómo los colores quedan unos con otros, respecto al fondo en contraste con el color del elemento, principalmente si es el mismo color, pero con un brillo diferente, llegando a una buena luminiscencia, esto varía entre 1:1 que es ningún contraste y 21:1 esta escala se contrasta haciendo uso de la fórmula de ratio de luminancia relativa, que compara la luminosidad de ambos colores. La fórmula es:

$$\text{Ratio de contraste} = \frac{L_1 + 0.05}{L_2 + 0.05}$$

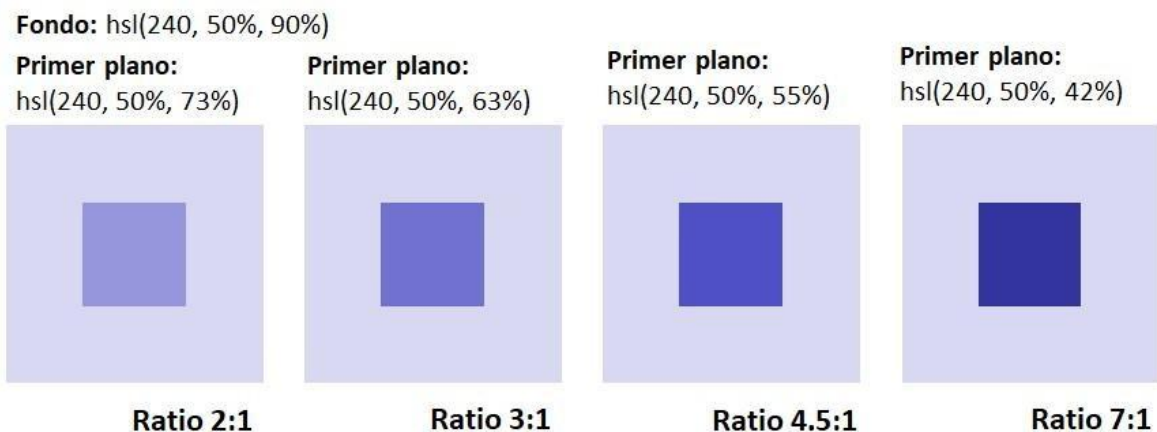
Donde:

- L_1 : es la luminancia relativa del color más claro.
- L_2 : es la luminancia relativa del color más oscuro.
- 0.05: es una constante que compensa la luz ambiental y el ruido visual.

A esto se le puede dar un claro ejemplo con el modelo *HSL* el cual es el mejor para obtener una mejor ratio de contraste, ya que solo cuando se cambia el último de los parámetros modifica la luminosidad del color que se coloque (W3C, 2018).

La Figura 1.1 muestra el ratio de aspecto con su plano y su fondo.

Figura 1.1
Ratio de proporción



Los criterios que abarcan este concepto son los siguientes 1.3.3, 1.4.1, 1.4.3, 1.4.6, 1.4.11. Todos los conceptos se encuentran en el Anexo I.

- Uso del Color

El uso del color se enfoca principalmente en cómo un único medio visual transmite la información indicada de una acción haciendo referencia a que no sólo el color tiene que ser el adecuado, sino que se puede hacer uso de figuras o formas que, junto con el color, ofrezcan una metáfora de cómo se hace el uso de la información.

Si se toma en cuenta un semáforo vehicular, en este sólo se proyecta el color verde indica avanzar/continuar, amarillo indica reducción de velocidad y rojo señala detenerse, esto es un claro ejemplo de cómo se transmite información acerca del tráfico únicamente con el uso de un color, lo cual podría implicar que las personas que tienen daltonismo pueden presentar dificultades, ya que no siempre cuentan con una orientación de cómo están colocados los colores en el semáforo lo cual puede llegar a provocar un accidente vial.

En contraparte, el semáforo peatonal indica cuándo se detiene por medio de la figura de una persona que se encuentra de pie sin hacer nada y avanzar cuando la figura de una persona está caminando, esto da información más clara que un color al momento de indicar si se trata de seguir o detenerse.

Ciertamente, en muchos sitios *web* se usa el color rojo, verde, azul o amarillo como indicador de estados en los que se encuentra el sistema: rojo si algo está saliendo mal, verde como confirmación, azul como información y amarillo como advertencia, lo cual suele ser reflejado con el simple color, pero, como se vio en el ejemplo anterior, debe comunicarse con mayor claridad a la hora de comunicar la información al usuario dentro del sistema *web*.

Se puede ejemplificar que, dentro de la *web*, si algún elemento presenta el color rojo y lleva una X, es más claro el mensaje de error, si es color verde y lleva una marca de verificación es más claro el mensaje de que es correcto, si es color amarillo y lleva un signo de admiración es muy probable que algo no se encuentre bien; y de esta manera el usuario con daltonismo comprenda la información de mejor manera y verifique si su proceso fue realizado correctamente o no.

Como se puede observarse en la Figura 1.2, solo se presenta texto en un cuadro y ambos son de color rojo.

Figura 1.2
Mensaje de error poco específico



Como se puede apreciar en la Figura 1.3 tanto el texto como la caja son de color rojo dando a entender que la acción no fue realizada o hubo algún problema. Se puede ver que tiene los dos elementos de la imagen anterior, pero ahora cuenta con una X,

marcando que algo estuvo mal y la palabra error en mayúsculas las cuales pueden ayudar a tener una mejor percepción de la información.

Figura 1.3
Mensaje de error



Cuando no se proporciona información clara acerca de las condiciones en las que se realizó la subida del archivo, por ejemplo, si existe un límite de tiempo, existe el riesgo de que un usuario con dificultad para ver el color rojo llegue a interpretar de forma errónea el resultado, por lo que si se utiliza iconos u otros elementos visuales que indiquen si la subida fue exitosa o fallo, puede deducirse con mayor facilidad el resultado.

Esta problemática se extiende a múltiples escenarios, comenzando con la interpretación de gráficas hasta la organización visual del contenido, donde un diseño accesible facilita la identificación de la información crítica independientemente si tiene la discapacidad o no, previniendo su pérdida y mejorando la comprensión general del sitio *web*.

1.2 Marco conceptual

En este apartado se describen los conceptos fundamentales sobre el daltonismo y la percepción visual necesarios para el desarrollo de esta tesis, detallando las terminologías específicas que permiten comprender cómo se identifica y organiza en sus distintos tipos.

1.2.1 Daltonismo

El daltonismo o ceguera de color se define como la de eficiencia a una visión con color o *CVD* (*Color Vision Deficiency*, por sus siglas en inglés). El daltonismo se encuentra en 1 de cada 12 hombres es decir el 8% de éstos y una de cada 200 mujeres el 0.005%. Tomando en cuenta las cifras dadas por la organización *Colour Blind Awareness* del Reino Unido, hay aproximadamente 3 millones de personas con daltonismo en ese país, presentándose más en hombres que en mujeres. Alrededor del mundo se estima que hay 300 millones de personas con esta afección, las cuales no en su totalidad saben que lo padecen (Awareness, About Colour Blindness, 2022).

Se tiene conocimiento de distintas causas que pueden provocar el daltonismo, la causa más común es la herencia, transmitiéndose principalmente a través del cromosoma X de la madre al hijo, aunque sea muy inusual proviene de la madre. Por otra parte, se desarrolla como consecuencia de enfermedades tales como diabetes o esclerosis múltiple; o por el uso prolongado de ciertos medicamentos y sustancias químicas.

Actualmente no existe tratamiento o cura para el daltonismo de origen hereditario por lo que la mayoría de las personas que cuentan con esta discapacidad se tienen que adaptar

al ambiente en el cual viven. Los niños requieren especial atención, particularmente en su entorno educativo. En el caso de los adultos esta afección implica ajustes tanto al estilo de vida como el ambiente laboral debido a que la falta de accesibilidad hace que el manejo del daltonismo cause complicaciones cuando se adaptan a su discapacidad (Awareness, 2022).

El daltonismo causa problemas en las tareas diarias, tales como la combinación de colores al vestirse, la identificación de señales de tránsito, el trabajo con cableado o componentes eléctricos o la interpretación de diagramas escolares o laborales, una forma de dar apoyo a las personas que sufren daltonismo, es por medio de anteojos, estos ayudan a las personas daltónicas a diferenciar los colores actuando como un filtro de color que se pone enfrente a sus ojos para ayudar a identificar ciertos colores de mejor manera y aumentar el contraste entre colores.

Otra forma de apoyo consiste en ayudas visuales mediante aplicaciones para dispositivos móviles o computadoras, las cuales favorecen la percepción de los colores de manera diferente. Un método rápido sería tomar fotografías con el teléfono o tableta y, posteriormente, se utiliza la herramienta de cuentagotas para que el dispositivo identifique los colores.

La identificación del daltonismo suele ocurrir mediante la observación contextual, ya que no existe una percepción objetiva de los colores que se alteran en esta condición, el diagnóstico normalmente surge a partir de actividades cotidianas donde aparecen dificultades en la distinción de tonalidades particularmente cuando se plantean preguntas directas como “¿Qué color es este?”. Desde una perspectiva científica los principales indicadores consisten en: la incapacidad de diferenciar ciertos colores, variaciones en la percepción del brillo y dificultad para distinguir matices específicos dentro de la misma gama cromática (Institute, Daltonismo).

Esta capacidad se divide en tres tipos: Tritanopia, Deuteranopia y Protanopia, siendo estos complicados ya que existen las *normalias*, qué es la dificultad de distinguir entre dos colores o más, a diferencia de las *nopias*, que hacen que sea imposible la distinción entre dos colores o más.

- Tritanopia

La tritanopia es el tipo de daltonismo que causa la dificultad de distinguir entre varias combinaciones de colores principalmente azul y amarillo, lo que hace que no se perciben diferencias entre el azul y el verde, entre el morado y el rojo, y entre el amarillo y el rosado. Otro efecto es que hace que los colores se vean menos brillantes, siendo este tipo de daltonismo el menos común (Awareness, 2022).

La tritanomalía, hace que solo sea difícil la distinción entre azul y verde, y amarillo y rojo. en la Figura 1.4 se muestra la diferencia de cómo vería la imagen una persona sin daltonismo del lado izquierdo y del lado derecho se muestra cómo la vería una persona con tritanopia.

Figura 1.4

Imágenes de muestra de visión con tritanopia <https://www.colourblindawareness.org/colour-blindness/>



- **Deuteranopia**

La deuteranopia es un tipo de daltonismo que afecta de forma específica a la percepción de los colores rojo y verde, impidiendo la distinción entre estos tonos. Este tipo de daltonismo afecta a aproximadamente a tres de cada cuatro personas con el padecimiento, siendo éste el más común de los tres tipos.

La Figura 1.5 muestra del lado izquierdo cómo ve los colores una persona normal y del lado derecho muestra como lo ve una persona con deuteranopía, la diferencia resulta ser perceptible y esta, a veces interfiere en la vida de la persona.

Figura 1.5

Imágenes de muestra de visión con Deuteranopia <https://www.colourblindawareness.org/colour-blindness/>



En cuanto a la deuteranomalía, muchas tonalidades verdes pueden llegarse a percibirse más rojas haciendo difícil distinguir entre rojo y verde. Este tipo por lo general no interfiere normalmente en las actividades cotidianas, por lo que es difícil que la persona se dé cuenta de las diferencias que hay entre los colores (Awareness, 2022).

- **Protanopía**

La protanopía es el tipo de daltonismo que presenta características similares a la deuteranopía en cuanto a la dificultad para distinguir los colores rojo y verde; sin embargo,

lo que la distingue es que las tonalidades rojas se perciben con un matiz más verdoso y con menor intensidad lumínica.

En la Figura 1.6 del lado izquierdo se aprecia la como vería los colores una persona sin daltonismo y del lado derecho como lo vería una persona con protanopia, se aprecia que los lápices de colores de tonalidades rojas son perceptibles como tonalidades de color verde (Awareness, 2022).

Figura 1.6

Imágenes de muestra de visión con protanopia <https://www.colourblindawareness.org/colour-blindness/>



1.2.2 Internet

El internet constituye uno de los avances tecnológicos más significativos de la historia de la humanidad, se establece una red de comunicación global accesible desde cualquier punto del planeta, siempre y cuando se cuente con conexión. Esta capacidad de interconexión mundial representó un paradigma revolucionario que superó cualquier expectativa tecnológica previa a su creación.

- Historia del Internet

Fue en agosto de 1962 cuando se redactó una serie de memorándums escritos por JCR Licklider proveniente del *MIT* en dónde se escribe el primer concepto de internet denominado galáctica, en esta se imaginó un conjunto de computadoras, que se interconectarán entre sí, de forma rápida y desde cualquier punto del planeta, Licklider era director del programa de investigación informática *DARPA*.

Durante su período como director de la oficina de técnicas de procesamiento de información (IPTO) en *DARPA* en 1962, el científico J.C.R. Licklider mantuvo conversaciones clave con 3 personas influyentes en el desarrollo del internet: Ivan Sutherland, pionero en gráficos por computadora; Bob Taylor, quien posteriormente dirigiría el IPTO y promovería *ARPANET*; y Lawrence G. Roberts, principal diseñador técnico de *ARPANET*.

En Julio de 1961 el ingeniero eléctrico y científico Leonard Kleinrock marcó un hito fundamental en el desarrollo del internet, al publicar el trabajo *Information Flow in Large Communication Nets* en el *MIT*. Este el documento se presentó por primera vez los

principios matemáticos de la conmutación de paquetes sentando bases teóricas para redes digitales. Tres años más tarde Kleinrock consolidó estos avances en otro libro llamado *Communication Nets*, demostrando la viabilidad técnica de lo que posteriormente se convertiría en internet.

En 1965, se realizó una prueba de envío de información de una computadora TX-2 en Massachusetts, a una computadora Q-32 en California, mediante una línea telefónica conmutada de poca velocidad, dando el primer paso a lo que se conoce hoy en día como internet.

A finales de 1966, se formuló el plan principal para *ARPANET*, cuyo diseño se presentó públicamente en 1967 durante el congreso *ACM* sobre principios de sistemas operativos, donde hubo una discusión acerca de tecnología de redes de conmutación de paquetes. Este congreso sirvió como catalizador para que instituciones académicas y gubernamentales se coordinaran para la estandarización de los protocolos de red.

El primer enlace experimental de *ARPANET* se estableció en 1969, entre cuatro nodos: la universidad de California en Los Ángeles (*UCLA*), el Stanford Research Institute, la Universidad de California en Santa Bárbara y la universidad de Utah. El primer mensaje exitoso entre host se transmitió el 9 de octubre de 1969 desde *UCLA* al *Stanford Research Institute*.

En 1970, el *Network Working Group (NWG)* desarrolló el *Network Control Protocol (NCP*, por sus siglas en ingles), el primer protocolo estándar de comunicación para *ARPANET*. Este hito tecnológico sentó las bases para el desarrollo de aplicaciones de red.

En 1972, durante la *International Conference on Computer Communications (ICCC*, por sus siglas en ingles), Se realizó la primera demostración pública exitosa de una conexión a un host remoto a través de *ARPANET*. Este hito tecnológico marcó el origen del envío de mensajes electrónicos mediante la aplicación “*SNDMSG*”, desarrollada por Ray Tomlinson, considerado el primer sistema de correo electrónico funcional (Segaller, 1999).

- Bases para un mundo conectado.

La evolución de *ARPANET* hacia el internet requirió un complejo desarrollo tecnológico que trascendió sus orígenes iniciales. Este proceso implicó la integración de múltiples sistemas de redes: conmutación de paquetes terrestres, redes satelitales y redes de radio por paquetes, entre otras. La arquitectura de red que se conoce hoy en día surge precisamente de la convergencia tecnológica dónde cada tipo de red contribuye con sus avances particulares lo que hizo posible la internet actual.

Como antecedente al *NCP*, se desarrolló el Protocolo de Control de Transmisión/Protocolo de Internet (*TCP/IP*), un sistema de comunicación que funciona como el “lenguaje universal” De internet al dividir la información en paquetes, enrutador los eficientemente y reensamblándolos en su destino. Para su construcción se establecieron cuatro principios fundamentales siendo el primero que la red debía ser

autónoma en su mantenimiento evitando modificaciones innecesarias en las redes interconectadas cuando no fueran estrictamente requeridas.

El segundo principio fundamental establece que el sistema operaría bajo un mecanismo de “mejor esfuerzo” (*best-effort delivery*), donde la red priorizaría la entrega eficiente de paquetes, pero en caso de que fallara la transmisión, el protocolo implementaría automáticamente un proceso de retransmisión desde el nodo de origen. Este enfoque garantizaba la robustez sin requerir infraestructura compleja de almacenamiento inmediato.

El tercer principio fundamental establecía la implementación de dispositivos de interconexión denominados *puertas de enlace*. Estos dispositivos fueron diseñados con una arquitectura simple, opera sin almacenamiento de estados individuales del flujo de los paquetes cumpliendo con dos objetivos clave: minimizar la complejidad, evitando el rastreo de conexiones específicas y maximizar la robustez, facilitando la adaptación a diferentes tecnologías de red y mecanismos de recuperación de errores.

El cuarto y último principio, garantiza que ninguna entidad o gobierno tuviera autoridad global sobre la red permitiendo:

- La libertad de conexión: cualquier red quedaría integrada sin requerir permisos centrales.
 - La descentralización: operación distribuida geopolíticamente.
 - Resistencia: capacidad de sobrevivir a fallos o restricciones locales.
-
- Principales problemáticas en la creación del internet.

Los investigadores del proyecto *ARPANET*, liderados por Vint Cerf y Bob Kahn, enfrentaron desafíos técnicos fundamentales durante el desarrollo de los protocolos de red. Estos problemas fueron clave en el establecimiento del principio de conexión de extremo a extremo mejormente conocido como *end-to-end*, el cual permitió garantizar la entrega completa de los paquetes de datos a través de la red, sentando así las bases técnicas para el funcionamiento integral del internet.

Los desafíos técnicos incluyeron el desarrollo de algoritmos especializados cuyo objetivo era prevenir que la pérdida ocasional de paquetes interrumpiera permanentemente la comunicación y la implementación de mecanismos de retransmisión confiables, que garantizaran el reenvío de cualquier paquete de forma precisa hasta su nodo de origen.

Otro desafío consistió en la implementación la segmentación efectiva de hosts que evitara el envío indiscriminado de paquetes, lo que causaba congestión en redes intermedias y posibles confusiones en los nodos destino. Paralelamente, se requería que las puertas de enlace interpretaran correctamente los encabezados IP para cuatro acciones: realizar el enrutamiento preciso de los paquetes, gestionar adecuadamente las interfaces de red, fragmentar los paquetes cuando fuera necesario y separar eficientemente las cabeceras de los datos.

El proceso por el cual un paquete se ensambla y se desensambla al momento de llegar al destino, se volverá a ensamblar. Evitar que la información se duplicara fue uno de los problemas más críticos del internet. Era necesario que la red tuviera alcance global, que incluyera técnicas de control de flujo para gestionar el envío y recepción de información, que fuera compatible con distintos sistemas operativos y que ofreciera tanto buen rendimiento como eficiencia en su implementación, teniendo como objetivo construir una infraestructura de comunicación robusta, interoperable y escalable, capaz de conectar computadoras en todo el mundo de manera confiable (Tanenbaum, 1996) .

1.2.3 Páginas *web*

Una página *web* es un documento digital que se visualiza a través de un navegador, puede contener texto, imágenes, enlaces, videos y otros recursos interactivos que permiten que el usuario acceda a información o realizar acciones específicas.

Las páginas *web* se construyen por medio de programación haciendo uso de *HTML*, *CSS* y *JavaScript*, lo que permite la presentación a la información, así de la organización del contenido de la página y las interacciones del usuario con la página.

- Propuesta de Tim Berners-Lee

Tim Berners-Lee es reconocido como el creador de la *World Wide Web (WWW)*. Berners-Lee se encontraba en la búsqueda de una forma en la cual compartiera documentos científicos entre sus colegas investigadores y académicos, los cuales se encontraban en diferentes puntos geográficos.

Berners-Lee desarrolló un sistema de hipertexto que permitía la comunicación para enlazar documentos entre sí.

En marzo de 1989 Berners-Lee presentó una propuesta en la *CERN (Conseil Européen pour la Recherche Nucléaire)*, que en español significa Organización Europea para la Investigación Nuclear por sus siglas en francés), con el objetivo de mejorar la transmisión y la accesibilidad de la información a nivel global, la propuesta describía lo que se conoce hoy en día como sistema de hipertexto, el cual permitiría el enlace de diferentes documentos que se encontraran almacenados en diferentes dispositivos y así crear una red de información más eficiente y de forma distribuida (Berners-Lee & Fischetti, 1999).

El principal desafío radicaba en la ausencia de precedentes tecnológicos para un sistema de esta escala, lo que planteaba interrogantes fundamentales sobre los mecanismos de acceso distribuido que permitirían a múltiples usuarios consultar documentos almacenados en diferentes nodos de red de manera simultánea y coherente.

Los componentes de la propuesta "*Information Management: A Proposal*" dada por Berners-Lee, presenta cuatro componentes importantes dentro de la propuesta los cuales son:

- *Hipertexto*: hace referencia al sistema de organización de la información en que se realiza la conexión de documentos mediante *hipervínculos*, mayormente conocidos como *enlaces*, estos permiten al usuario la navegación de un documento a otro sin que sea de una forma lineal, permitiendo un acceso más flexible y dinámico en la navegación. La importancia de ese sistema fue tan grande que se usa en todos los sitios *web* formando una red de navegación global.
- *Sistema Distribuido*: el sistema no debía de depender de un servidor centralizado ni de una base de datos única, refiriéndose a que los datos no podían existir en un solo lugar, sino que éste se encontraría en diferentes servidores, pero debería permanecer accesible desde cualquier parte, por medio de la red. Considerando los puntos anteriores, se entiende que el sistema debería ser escalable y resistente, lo que daría paso a un crecimiento de la *web* sin sobrecargar un solo servidor para que la información estuviera disponible en todo momento, sin importar donde esté almacenada.
- *Acceso Uniforme*: este componente se refiere a que, sin importar el sistema operativo, el dispositivo o plataforma en la cual se ingrese, debería de poderse acceder a la *web*, sin ninguna complicación, permitiendo que se pueda acceder a un sitio *web* desde una computadora, dispositivo móvil u otros dispositivos, todos usando un navegador que interprete el mismo lenguaje *HTML*.
- *La colaboración entre computadora y personas*: en este último componente, la *web* no se planteaba solo como un sistema para acceder a la información sino también como una plataforma para la colaboración ya que la *web* debía facilitar el crecimiento y la modificación de todo aquel contenido que proviniera de los usuarios, con la finalidad de poder colaborar en tiempo real, compartir ideas y con ello construir conocimiento de una manera colaborativa. La importancia que tuvo este, es tal que se puede considerar uno de los pilares de la *web* moderna en donde múltiples usuarios pueden colaborar juntos, y compartir información en tiempo real, transformando industrias, revolucionando la ciencia, innovando la educación y redefiniendo los modelos de negocio.

Los conceptos fundamentales para el internet moderno son *URL*, *HTTP* y *HTML*, los cuales se describen a continuación:

- *URL*: el *Uniform Resource Locator* (*URL*, por sus siglas en inglés), es la dirección específica en la cual los recursos de la *web* y el cómo se acceden a él tienen forma son esenciales para localizar y acceder a cualquier recurso en internet desde distintas páginas *web* o incluso hasta imágenes también permitiendo la descarga o su vida de archivos sin estas la navegación sería muy difícil o imposible.

- *HTTP*: el *Hypertext Transfer Protocol* (*HTTP*, por sus siglas en inglés) es el protocolo que establece los estándares para la transmisión y formato de mensajes entre clientes y servidores *web*. Define cómo los navegadores envían solicitudes (como la carga de archivos o envío de datos de formularios) y cómo los servidores responden a estas peticiones. Considerado la columna vertebral de la comunicación *web*, *HTTP* permite a los usuarios navegar y recibir el contenido solicitado a través de sus navegadores. Una variante más segura, el protocolo *HTTPS* (*HyperText Transfer Protocol Secure*, por sus siglas en inglés), añade cifrado mediante SSL/TLS (Secure Sockets Layer / Transport Layer Security), protocolos que protegen la integridad y confidencialidad de los datos transmitidos.
- *HTML*: El *Hypertext Markup Language* (*HTML*, por sus siglas en inglés) es el lenguaje de marcado estándar utilizado para la creación y la estructuración páginas *web*, mediante un sistema de etiquetas, definiendo elementos fundamentales como títulos, encabezados, párrafos, enlaces, imágenes, tablas y formularios. Estas etiquetas proporcionan instrucciones a los navegadores *web* para interpretar y mostrar el contenido con un formato visual organizado y legible para los usuarios.

1.2.4 Diseño *web*

El diseño *web* ha experimentado una transformación profunda desde sus primeras manifestaciones en la década de 1990. En sus inicios, los sitios *web* eran estructuras estáticas construidas con *HTML* básico, donde el contenido se organizaba mediante tablas y los elementos visuales eran mínimos debido a las limitaciones tecnológicas de la época. La estética era secundaria frente a la funcionalidad, y la experiencia del usuario se limitaba a la navegación entre enlaces subrayados y bloques de texto plano.

Con la introducción de las hojas de estilo en cascada (*CSS*) a mediados de los años noventa, se abrió la posibilidad de separar el contenido de la presentación visual. Esta innovación permitió controlar aspectos como colores, tipografías y disposición de elementos, marcando el inicio de una preocupación más formal por la estética y la usabilidad. Posteriormente, el uso de *JavaScript* amplió las capacidades interactivas de los sitios, dando paso a interfaces más dinámicas y receptivas.

Durante los años 2000, el diseño *web* comenzó a adoptar principios del diseño gráfico tradicional, incorporando gradientes, sombras, animaciones y una mayor atención al branding. El auge de Flash permitió experiencias multimedia avanzadas, aunque su uso fue eventualmente desplazado por estándares abiertos como *HTML5* y *CSS3*, que ofrecían mayor compatibilidad y accesibilidad (Eric Meyer, 2023).

La llegada de los dispositivos móviles transformó radicalmente el enfoque del diseño *web*. El diseño responsivo se convirtió en una práctica esencial, garantizando que los sitios se

adaptaran a diferentes tamaños de pantalla y resoluciones. Paralelamente, se consolidaron metodologías centradas en el usuario, como el diseño centrado en la experiencia (*UX*) y la interfaz de usuario (*UI*), que priorizan la accesibilidad, la eficiencia y la satisfacción del usuario final.

En la actualidad, el diseño *web* se apoya en un ecosistema de herramientas sofisticadas que permiten prototipado, colaboración y desarrollo ágil. Plataformas como Figma y Adobe XD facilitan el diseño colaborativo en tiempo real, mientras que *frameworks* como *Tailwind CSS*, *Bootstrap* y Material UI ofrecen componentes reutilizables que aceleran la implementación visual. Por otro lado, tecnologías como *React*, *Astro* y *GSAP* permiten integrar diseño y funcionalidad de forma fluida, favoreciendo experiencias interactivas y accesibles. Estas herramientas no solo optimizan el flujo de trabajo, sino que también promueven prácticas inclusivas, como la corrección de contrastes y la adaptación para usuarios con discapacidades visuales (Kholmatova, 2021).

1.3 Marco tecnológico

El internet constituye uno de los avances tecnológicos más significativos de la historia de la humanidad, se establece una red de comunicación global accesible desde cualquier punto del planeta, siempre y cuando se cuente con conexión. Esta capacidad de interconexión mundial representó un paradigma revolucionario que superó cualquier expectativa tecnológica previa a su creación.

1.3.1 Chrome Extensions

Si bien existen diferentes herramientas destinadas al ajuste del nivel cromático del sistema operativo con filtros integrados o aplicaciones de terceros, las modificaciones afectan de forma global todo el contenido visual del dispositivo por lo que una extensión de navegador ofrece una solución menos invasiva debido a que se limita a dicho navegador mediante la inyección dinámica de código *CSS* para modificaciones en tiempo real y de la misma forma facilita su activación y desactivación inmediata según las necesidades del usuario.

Es importante añadir que una extensión permite el uso de filtros dinámicos para cambio de color, así como lo hace un filtro de fotografía o de cualquier aplicación de diseño en el cual los usuarios con daltonismo al momento de usarlos les permite la identificación de los colores.

Además, no es nada complicado utilizar la *API* que proporciona Google para poder crear todo este tipo de extensiones, permitiendo personalizar la interfaz del usuario, cómo opera el navegador y, por medio de un editor de código, compilar la extensión y subirla al mismo Google haciendo uso de *HTML*, *CCS* y *JavaScript*, y si así lo requiere el desarrollador un *framework* que se base en estos tres, permitiendo flexibilidad y fácil uso de la *API*.

Es importante recalcar que la *API* permite el trabajo en un medio local, publicando la extensión dentro del navegador de forma local mandando llamar a la extensión desde un directorio dentro de la computadora, así como los archivos necesarios para poder aplicar los filtros de color de forma adecuada.

También permite su publicación gratuita en la *store* del mismo Chrome, facilitando el acceso para la descarga y una vez alojada dentro del navegador no resulta necesaria la conexión para usarla, lo que permitirá que aun así se sigan viendo los cambios de color en páginas descargadas previamente debido a que se guardan en el almacenamiento local del navegador, otro ejemplo caso son los archivos *PDF*, que utilicen el navegador como interfaz para que el usuario pueda interactuar con ellos.

En resumen, Chrome Extensions es la mejor herramienta para desarrollar esta aplicación no solo por su ecosistema sino por su compatibilidad con los navegadores más comerciales del mercado, tales como Google Chrome, Mozilla Firefox y Microsoft Edge (Google, 2026).

1.3.2 Figma

Figma surgió como una respuesta a las limitaciones del diseño tradicional de escritorio, con la visión de llevar el diseño colaborativo directamente al navegador y hacerlo accesible para equipos distribuidos. Fundada en 2012 por Dylan Field y Evan Wallace, la empresa apostó por un modelo basado en la nube que permitiría edición simultánea, comentarios en contexto y un historial de versiones centralizado, sentando las bases de una nueva cultura de diseño compartido. Su enfoque en la colaboración en tiempo real transformó flujos de trabajo y aceleró la toma de decisiones, consolidando su reputación como una herramienta disruptiva en el ecosistema del diseño de interfaces.

El lanzamiento público de Figma en 2016 marcó un hito al ofrecer un editor vectorial y de prototipado principalmente *web*, complementado por aplicaciones de escritorio para uso sin conexión y aplicaciones móviles para visualizar prototipos en dispositivos iOS y Android. Esta arquitectura permitió trabajar sin fricciones entre plataformas y estandarizó la colaboración entre diseñadores, *product* managers y desarrolladores en un mismo espacio de trabajo, con gran foco en *UI* y *UX* y soporte de formatos abiertos como *SVG* y *PDF*. La propuesta *web-first* redujo barreras de instalación, facilitó la adopción y posicionó a Figma frente a herramientas de escritorio al priorizar accesibilidad y trabajo simultáneo (Figma, s.f.).

La evolución funcional de Figma se ha caracterizado por integrar prototipado interactivo, componentes reutilizables, variantes, *auto-layout* y librerías compartidas, reforzando la coherencia de diseño y escalabilidad a nivel de sistemas de diseño. En paralelo, la comunidad impulsó la creación y difusión de *plugins*, archivos-plantilla y prácticas de diseño colaborativo que enriquecen el ecosistema y aceleran la producción de interfaces y productos digitales. Este tejido comunitario y la capacidad de co-edición en tiempo real se volvieron diferenciadores clave, alineando la herramienta con metodologías ágiles y ciclos de entrega continuos (Frost, 2016).

En términos de impacto, Figma consolidó una nueva manera de trabajar donde el diseño deja de ser un silo y se integra al ciclo de producto con transparencia y participación transversal. La edición simultánea, los comentarios en vivo y la compartición de librerías favorecen la gobernanza del diseño y la consistencia entre equipos, al tiempo que disminuyen el “*handoff*” friccionado y los desajustes entre intención y ejecución. Esta convergencia funcional y cultural reconfiguró el rol del diseño como práctica colaborativa y medible, y expandió su alcance más allá del equipo de diseño.

Hoy, Figma se posiciona como una plataforma integral que combina diseño, prototipado, especificación y colaboración en un entorno unificado, con soporte multiplataforma y un ecosistema de extensiones y comunidades activas. Su adopción generalizada responde a la capacidad de conectar a equipos dispersos, mantener sistemas de diseño vivos y sostener un flujo continuo entre idea, validación y entrega, manteniendo el foco en accesibilidad, consistencia y velocidad (Jeff Gothelf, 2016).

1.3.3 Firebase

Firebase fue fundado en 2011 por James Tamplin y Andrew Lee como plataforma tecnológica orientada a ofrecer soluciones de *backend* como servicio (*Backend as a Service*, por sus siglas en inglés *BaaS*). Su innovación principal consistió en proporcionar una herramienta llamada Firestore la cual es un gestor de base de datos en tiempo real, permitiendo a los desarrolladores implementar funcionalidades *backend* sin necesidad de gestionar infraestructura propia. Este modelo revolucionario abstrae la complejidad del desarrollo *backend*, permitiendo a los equipos concentrarse en la lógica de negocio y la experiencia de usuario.

En 2014 Google adquirió Firebase integrándolo a su ecosistema y convirtiéndolo en una plataforma completa de desarrollo para aplicaciones *web* y móviles. Tras su adquisición por Google, Firebase evolucionó significativamente, transformándose de una simple solución *BaaS* a una plataforma de desarrollo integral. Esta evolución incorporó:

- Un conjunto completo de herramientas para el ciclo de desarrollo completo
- Funcionalidades avanzadas de administración y monitoreo
- Capacidades extendidas para el análisis de datos
- Soporte para integraciones con extensiones de navegador

Esta expansión permitió a los desarrolladores no solo implementar *backend*, sino también gestionar, analizar y optimizar sus aplicaciones de manera holística.

- Funcionalidades de Firebase

Las principales funciones y herramientas que tiene Firebase se encuentran en la amplia gama de productos agrupados en las siguientes áreas:

- *Desarrollo de la aplicación*

Al ser una plataforma de desarrollo integral que opera en tiempo real, almacena los datos en un formato denominado *JSON*, y no en *SQL* como se realiza comúnmente. Esta arquitectura permite la sincronización inmediata entre múltiples usuarios y dispositivos conectados, facilitando el desarrollo de aplicaciones y ecosistemas *web* con capacidades de actualización instantánea. Estas características son ideales para implementar funcionalidades como visualizaciones de datos en tiempo real (gráficas dinámicas), aplicaciones colaborativas multiusuario o extensiones interactivas.

Dentro de su arquitectura en la nube, Firebase permite a los desarrolladores ejecutar código *backend* en respuesta a eventos, lo que posibilita realizar cambios en la base de datos, gestionar la autenticación de usuarios e interactuar con otros servicios tanto de Firebase como externos. A este servicio se le denomina computación *serverless* o sin servidor, eliminando la necesidad de gestionar infraestructura manualmente.

Firebase incluye un servicio de hosting diseñado para implementar aplicaciones *web* de forma rápida y segura. Este servicio soporta tanto contenido estático como dinámico, integrando funcionalidades como *Cloud Functions* para operaciones del lado del servidor. Entre sus características destacadas se encuentran: la configuración automática de certificados *Secure Sockets Layer (SSL)*, la posibilidad de usar dominios personalizados y la distribución global de contenido a través de la red de entrega de Google (*CDN*, por sus siglas en inglés *Content Delivery Network*), que consiste en una red de servidores distribuidos geográficamente para acelerar la entrega de archivos al usuario final, reduciendo la latencia y mejorando el rendimiento.

- *Calidad de la Aplicación*

La calidad es un criterio fundamental al seleccionar servicios tecnológicos. *Firestore*, como parte del ecosistema Firebase, incluye herramientas avanzadas para garantizar este aspecto, específicamente, ofrece *Firebase Crashlytics*, un sistema de monitoreo en tiempo real que permite a los desarrolladores:

- Detectar y diagnosticar fallos inmediatamente
- Obtener informes detallados con contexto completo de cada error
- Priorizar problemas según su impacto en la estabilidad de la aplicación

Esta solución integral para gestión de errores se ha convertido en estándar para el mantenimiento de aplicaciones críticas, permitiendo correcciones proactivas antes de que afecten a los usuarios finales (Digital55, 2020).

1.4 Marco metodológico

En este apartado se realiza la descripción de tres metodologías ágiles, mencionando su nacimiento y describiendo el proceso que utilizan para llevar a cabo el desarrollo en el cual las empleen.

1.4.1 Scrum

Scrum se define como un marco de gestión de proyectos ágil, la palabra *scrum* procede del *rugby* y hace referencia al modo en el que el equipo trabaja para poder mover el balón por la cancha (Sutherland & Schwaber, 2020).

La metodología *scrum* trabaja por medio de *sprints*. Un *sprint* se define como un periodo breve de tiempo (*timebox*), en el cual un equipo completa una cantidad de trabajo establecida. Un *sprint* suele durar de una a cuatro semanas. Los *sprints* completan las historias de usuario preestablecidas que forman parte de una lista priorizada de tareas (*Product Backlog*), en la que se recopila la funcionalidad a trabajar, el valor para el cliente y la priorización u orden en el que se trabajará.

Los sprints se planifican mediante una serie de 4 de pasos:

- *Paso uno*: es la preparación de la reunión de planificación de *sprints*. En dicha reunión deberá estar presente el propietario del producto (*Product Owner*), el equipo de desarrollo y el encargado de que se cumplan los sprints establecidos (*Scrum Master*) principalmente, si se requiere la presencia de alguna persona la cual pueda aportar a la reunión ya sea empleado, experto en el tema, etc, estas personas son denominadas *stakeholders*. Durante la reunión se dividirá en dos partes las cuales serán el qué se hará en cada sprint y el cómo se llevará a cabo:
 - *Primera parte*: se realiza la revisión del *Product Backlog* en la que el propietario del producto presenta a lo que el desea llegar en ese sprint (*Sprint Goal*) y el equipo de desarrollo cuestiona acerca del alcance y los criterios de aceptación. El equipo de desarrollo seleccionará las funcionalidades a las que se comprometen a completar en el tiempo establecido, el número de las funcionalidades lo delimita la velocidad del equipo.
 - *Segunda parte*: cada una de las funcionalidades del *Product Backlog* que fueron seleccionadas por el equipo de desarrollo, se desglosan en tareas concretas, así mismo se estima el esfuerzo que llevará completarla, posteriormente se hace la de terminación (*Definition of Done*, *DoD* por sus siglas en inglés) donde se acuerdan las condiciones que determinan que la tarea esta completada y finalmente el equipo acepta el compromiso de culminación de la tarea.
- *Paso dos*: es la fijación de límites de tiempo para la planificación de los *sprints*. Este tiempo no podrá exceder las dos horas por cada semana de duración del sprint, por ejemplo, de si el sprint dura tres semanas la reunión no deberá durar

más de seis horas y si el equipo de desarrollo está satisfecho con el resultado antes del tiempo límite se podrá concluir la reunión. Hay que resaltar que el *Scrum Master* es el encargado de que este tiempo límite sea cumplido.

- *Paso tres*: es la definición de los objetivos del *sprint*. Debido a que durante la planificación puede haber poca información disponible o esta puede basarse en suposiciones, tomando por ejemplo cuando el cliente no sabe en su totalidad que es lo que requiere, por ello las historias de usuario son útiles para complementar el nivel de la información.
- *Paso cuatro*: es la estimación del esfuerzo del *sprint*. En este, el equipo de desarrollo se centra en ver qué es capaz de completar durante el tiempo establecido para el *sprint*, tomando en cuenta el conocimiento que tiene el propio equipo de desarrollo.

Una vez que se ha establecido el *sprint*, posteriormente se lleva a cabo el desarrollo del *sprint* en el cual se realizan reuniones diarias durante la duración del mismo. A estas reuniones se les denomina reuniones rápidas (*Daily Scrum*). En estas reuniones se trata el progreso del *sprint*, así como también se identifican los impedimentos que lleguen a existir. Durante la reunión se plantean tres preguntas para generar una estructura, las cuales son:

- ¿Qué hice ayer?
- ¿Qué haré hoy?
- ¿Qué obstáculos tengo?

La reunión no deberá tener una duración mayor a 15 minutos. Durante la reunión deberán de estar presentes: el *Scrum Master*, el equipo de desarrollo y en caso de haber necesidad, los *Stakeholders* implicados.

Una vez que el *sprint* haya finalizado, se realiza la revisión de este (*Sprint Review*), la cual tiene una duración de 1 hora, en la revisión se tratan los siguientes puntos:

- *Demostración del trabajo realizado*: el equipo de desarrollo muestra las funciones que fueron completadas durante el *sprint* a los *Stakeholders* implicados y al *Product Owner*.
- *Recopilación de comentarios*: el *Product owner* y los *Stakeholders*, comentan acerca de lo que le mostro el equipo de desarrollo. Los comentarios le permiten al equipo de desarrollo ajustar el proyecto y realizar las modificaciones de forma más eficaz.

- *Adecuación a la visión del producto*: aquí se realiza un ajuste de visión, así como de los objetivos generales del producto validando si los esfuerzos son suficientes y los objetivos del *sprint* avanzan de forma adecuada.
- *Celebración de logros*: da la oportunidad de reconocer el trabajo del equipo de desarrollo y su esfuerzo implicado en el *sprint*.
- *Identificación de mejoras*: el propósito de este punto es identificar áreas de mejora dentro del proceso de desarrollo y aplicarlas mejoras en futuros *sprints*.

El último paso a tomar en cuenta es la retrospectiva (*Sprint Retrospective*), la cual es una reunión donde el equipo de desarrollo debe reflexionar acerca de sus procesos, tomando en cuenta: la eficacia de los procesos y las herramientas utilizadas, la clasificación de recursos en aquellos que funcionan bien y los que no y, finalmente, la creación de un plan de mejora en la forma de trabajo. Esta retrospectiva no debe de tomar más de cuarenta y cinco minutos por cada semana del *sprint*, por ejemplo, si el *sprint* tiene dos semanas el tiempo máximo será de una hora y media (Atlassian, Scrum, 2025).

1.4.2 Kanban

Kanban es un marco de trabajo para el desarrollo ágil, en el cual los elementos se representan visualmente en un tablero, lo que permite a los miembros del equipo ver el estado del proyecto, así como de cada una de las tareas en cualquier momento.

La metodología *Kanban* se desarrolló a finales de los años cuarenta en Toyota por el ingeniero Taichi Ohno. Su inspiración provino de los mercados estadounidenses, donde se utilizaba un modelo de abastecimiento basado en almacenar únicamente lo necesario para satisfacer la demanda del cliente. Esta idea respondía a la necesidad de la empresa debido a que se dejaba de sobre producir y se evitaba el almacenamiento innecesario.

Ohno diseñó una producción *Just-in-Time* (*JIT*, por sus siglas en inglés), donde se requería de un control visual del flujo de materiales usando tarjetas físicas para autorizar la producción reduciendo costos en inventario, de ahí el nombre de *Kanban* que en Japones significa “tarjeta visual” (Kyōkai, 1989).

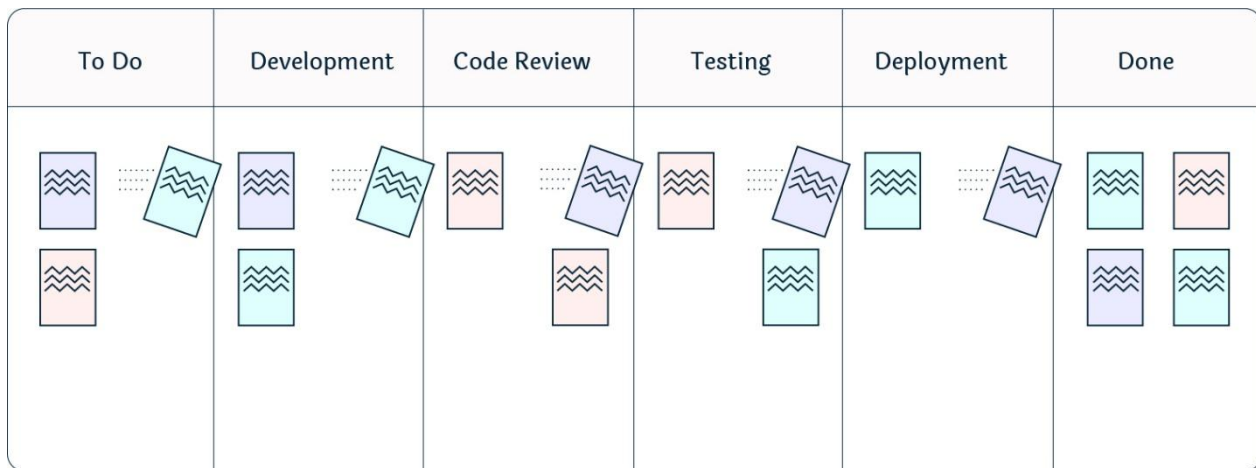
Lo que creo Ohno dio como resultado el Sistema de Producción Toyota (*Toyota Product System*, *TPS*, por sus siglas en inglés), su objetivo era el control de inventario para mejorar la eficiencia en producción mediante un sistema *pull* (como el de los mercados estadounidenses) en lugar de uno *push*. En el sistema *pull*, la producción se activa únicamente cuando hay una demanda real del cliente, evitando acumulaciones innecesarias. En cambio, el sistema *push* produce en función de previsiones, empujando productos hacia el mercado sin certeza de que serán requeridos, lo que puede generar sobreproducción y exceso de inventario.

La herramienta más importante dentro de la metodología es el Tablero *Kanban*, el cual se define como una herramienta de gestión en proyectos, diseñada para dar apoyo en la visualización del trabajo, así como limitar el trabajo en curso y maximizar la eficiencia haciendo uso de tarjetas y columnas para su manejo.

En la Figura 1.7 se muestra como es un tablero Kanban, así como las tareas se encuentran en proceso y movimiento en las diferentes partes del tablero.

Figura 1.7

Tablero Kanban por Sonya Siderova (<https://www.pexels.com/es-es/foto/oficina-negocio-espacio-de-trabajo-empresa-emergente-6804093/>)



Los tableros *Kanban* según David Anderson se dividen en cinco componentes: señales visuales, columnas, límites del trabajo en curso, punto de compromiso y un punto de entrega (Anderson, 2010), a continuación, se definen los componentes mencionados anteriormente:

- **Señales visuales:** estas son las tarjetas se pueden usar ya sea adhesivos, *tickets*, *post-its*, etc. Los equipos que utilizan *Kanban* escriben todos sus proyectos y elementos de trabajo en tarjetas, limitados a uno por tarjeta.
- **Columnas:** la columna representa una actividad específica dentro del proyecto. Estas actividades en conjunto crean un flujo de trabajo, las tarjetas se mueven dentro del flujo de trabajo, este flujo puede ser sencillo como por hacer (*To do*), en curso (*Doing*) y terminado (*Done*).
- **Límites del trabajo en curso:** los límites son el número máximo de tarjetas que puede haber en una columna en un momento del flujo. Se estipula cuántas tareas puede haber en cierta columna, por ejemplo, si se estipula que no puede haber más de tres tareas en la columna de en curso entonces cuando llegue un momento

que haya tres todo el equipo de trabajo deberá de enfocarse en terminar esas tres en lugar de colocar otra tarjeta.

- *Punto de compromiso*: es cuando el equipo estipula una idea y se empieza con el trabajo del proyecto, esto con ayuda de una lista de pendientes (*Backlog*) para el tablero que no han entrado en el flujo de trabajo.
- *Punto de entrega*: es el final del flujo de trabajo para el equipo, es decir cuando el producto final ya se encuentra en las manos del cliente, haciendo que todas las tarjetas se ubiquen en la última columna del flujo.

Kanban se basa en seis prácticas clave para su flujo de trabajo, garantizando la fluidez de procesos y la correcta gestión del flujo de trabajo, estas prácticas son:

- *Visualizar el flujo de trabajo*: es importante tener una visión clara acerca del flujo de trabajo, debido a que sobre este flujo se centrará el proyecto entero, asegurándose de que el flujo sea el correcto para el proyecto que se va a llevar.
- *Estandarizar el flujo de trabajo*: aquí, se estandarizan y definen las fases del flujo de trabajo según la cantidad de pasos, lo cual da paso a las columnas del tablero, normalmente se hace uso de las más comunes, pero se puede personalizar según el flujo de trabajo.
- *Identificar los bloqueadores y las dependencias*: hay que asegurarse que en el tablero se encuentren indicaciones de bloqueo o dependencia, permitiendo que todo el equipo tenga transparencia al realizar las acciones.
- *Fijar límites del trabajo en curso (Work In Progress, WIP, por sus siglas en inglés)*: los límites de trabajo son vitales para mantener la estabilidad en el flujo de trabajo, haciendo que aumente la productividad y optimiza los recursos de trabajo.
- *Fomentar la colaboración*: para que el proyecto avance sin problemas, se debe de tener una buena colaboración en equipo, en la cual, los miembros superen problemas y complicaciones que se lleguen a presentar en el flujo de trabajo, garantizando eficiencia y agilidad al finalizar tareas.
- *Utilizar tarjetas de Kanban*: cada una de las tareas debe de ser representada con una tarjeta, dentro de las tarjetas se dan detalles importantes como: la descripción de la tarea, la persona asignada y el tiempo estimado de finalización. Haciendo un proceso eficiente y transparente.

El proceso de desarrollo de la metodología *Kanban* se divide en cuatro fases:

- *Planificación*: en esta primera fase se identifican, organizan y priorizan los requerimientos del sistema, realizando tres pasos:
 - *Creación de historias de usuario*: las historias de usuario tienen como objetivo dar contexto al equipo para tener un mayor entendimiento de las tareas y el valor que cada una de ellas aporta al proyecto. Las historias de usuario se componen de tres elementos, quien solicita la tarea (puede ser una persona o un departamento completo), que tarea se solicita (esta petición puede ser general o específica), con qué fin se solicita la tarea (lo que se espera lograr con esta tarea), por ejemplo, “Como [el departamento de finanzas], quiere un botón, para descargar las gráficas”.
 - *Definición de tareas del proyecto*: una vez que se cuenta con las historias de usuario se realiza el establecimiento de tareas que pertenecerán al backlog, estas tareas se dividen en dos tareas técnicas y no técnicas. Las tareas técnicas son todas aquellas que están relacionadas directamente con la programación y el desarrollo del proyecto al contrario de las no técnicas que éstas están relacionadas con la investigación, *testing* y el diseño. Cada una de las tareas debe contener Nombre de la actividad, descripción de la tarea, prioridad dentro del *backlog*, responsable de la tarea, estado en el que se encuentra.
 - *Creación de tablero Kanban*: el tablero *Kanban* se modifica acorde de las necesidades del equipo partiendo de cuatro columnas. La primera es el *backlog* en la que se encuentran todas las tareas del proyecto, la segunda es por hacer (*To Do*) en la cual se colocan las tareas en espera a ser iniciadas, la tercera es en proceso (*Doing*) en esta son las tareas que se encuentran en proceso de elaboración y la última columna es terminado (*Done*), en la cual se encuentran las tareas que ya fueron aprobadas, Estas columnas pueden aumentar dependiendo de las necesidades del proyecto.
- *Limitaciones del proceso (WIP)*: la segunda fase se establece el límite de tareas que debe de haber en cada una de las columnas evitando la saturación de trabajo y facilitando el conocimiento del estado las tareas.
- *Ciclo de desarrollo*: en esta fase se establece el ciclo que va a tomar cada una de las tareas, el cual se repetirá hasta completar todas ellas, por ejemplo, planeación de que tareas se deberán completar, proceso de desarrollo de la tarea, *testing* y validación de la tarea, finalización de la tarea.

- *Mejora continua*: en esta última fase se realiza una revisión de las conferencias y los procesos que se realizaron durante el desarrollo, con la finalidad de evaluar el rendimiento, identificar problemas y proponer estrategias de mejoras en el flujo del trabajo (Atlassian, Kanban, 2025).

1.4.3 Desarrollo guiado por pruebas (*TDD*)

El Desarrollo Guiado por Pruebas (*Test-Driven Development*, *TDD*, por sus siglas en inglés), es una metodología de desarrollo de software que se enfoca en la escritura de pruebas automatizadas antes de la implementación de código funcional, dándole mayor prioridad a las pruebas. Surge de la idea de Kent Beck busca cuando mejorar la calidad del software reducir defectos y facilitar el diseño modular (Beck, 2022).

La Metodología *TDD* surge a partir de *XP*, por Kent Beck a finales de los años noventa por medio de su libro "*Test-Driven Development: By Example*", tomando para su desarrollo el modelo en espiral y las pruebas unitarias automatizadas.

TDD al ser una metodología que prioriza las pruebas automatizadas antes que el código, propone plantear una hipótesis sobre cómo se comporta el sistema que se quiere desarrollar. Posteriormente se corrobora si esa hipótesis o comportamiento con el código que se escribe en ese momento, tomando en cuenta un punto importante, que el código sea el mínimo posible para validarla.

Por lo tanto, *TDD* consiste en iteraciones sobre un ciclo de tres etapas conocido como *Red-Green-Refactor*, donde en cada una de las iteraciones se plantea una hipótesis nueva acerca del comportamiento que se desea para el sistema. A continuación, se describen las tres etapas:

- *Rojo (Red)*: en esta etapa se escribe un test automatizado para realizar una prueba de funcionalidad, la cual falla inicialmente, debido a que la funcionalidad aún no se ha implementado, validando que el test sea válido, en otras palabras, que siempre va a fallar hasta que se le da una función.
- *Verde (Green)*: en esta etapa se escribe el código mínimo necesario para que el test pase, centrándose en que el test pase lo más rápido posible, sin detenerse a perfeccionar código.
- *Refactorización (Refactor)*: una vez ya se ha pasado el test, al código se le realizan ajustes con el propósito de mejorar la estructura y calidad, mientras el test siga funcionando, eliminando redundancias y mejorando la legibilidad.

En la Tabla 1.1 se presenta una comparativa entre las metodologías descritas para la gestión y ejecución del proyecto: *Scrum*, *Kanban* y *TDD*, se detallan sus características principales con el fin de justificar la selección de Kanban como la metodología óptima para el desarrollo de esta tesis, debido a su flexibilidad y enfoque en el flujo continuo de trabajo.

Tabla 1.1
Cuadro comparativo de metodologías

Criterio de Evaluación	Scrum	TDD (Test Driven Development)	Kanban (Seleccionada)
Enfoque Principal	Gestión de proyectos por iteraciones fijas.	Calidad de software mediante pruebas unitarias previas.	Gestión del flujo de trabajo y entrega continua.
Unidad de Tiempo	Sprints (normalmente de 2 a 4 semanas).	Ciclo de desarrollo (Rojo -> Verde -> Refactor).	Flujo continuo (sin tiempos de ciclo fijos).
Visualización	Burndown charts y Sprint Backlog.	Cobertura de pruebas (Test Coverage).	Tablero Kanban con columnas de estado.
Manejo de Cambios	No se permiten cambios durante el Sprint.	Los cambios requieren reescribir las pruebas.	Alta flexibilidad; los cambios se añaden al flujo.
Roles y Equipo	Scrum Master, Product Owner, Dev Team.	No define roles, es una técnica de ingeniería.	No requiere roles; fomenta la autogestión.
Limitación de Carga	Basada en la velocidad del equipo por Sprint.	Basada en la complejidad de la función.	Limitación estricta del WIP (Work In Progress).
Objetivo de Calidad	Definición de Terminado (DoD).	Cero errores en fase de codificación.	Mejora continua y eliminación de cuellos de botella.

1.5 Estado del arte

En este apartado se dará una descripción de los proyectos utilizados para la evaluación de *DaltonAssist* en contraste con proyectos similares, mencionando las capacidades de las otras extensiones, con el objetivo de determinar las ventajas de *DaltonAssist* frente a otras soluciones.

1.5.1 Dalton (*Colorblind Extension*)

Dalton es una extensión gratuita para navegadores basados en *Chromium*, como Google Chrome, diseñada para mejorar la experiencia visual de personas con distintos tipos de daltonismo. Su objetivo es permitir que los usuarios vean colores que normalmente no pueden distinguir, ajustando dinámicamente la paleta cromática del navegador.

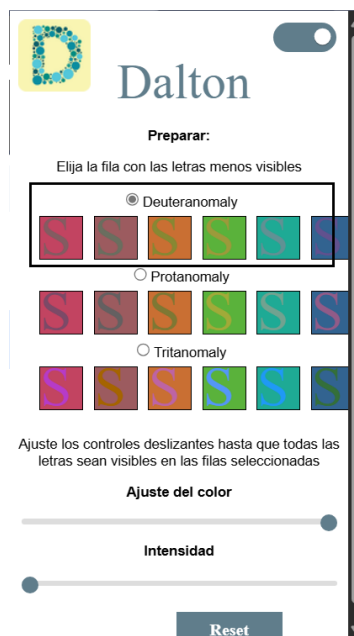
Características principales:

- Simulación y corrección para tres tipos de daltonismo: protanopia (rojo), deuteranopia (verde) y tritanopia (azul).
- Interfaz sencilla con controles deslizantes para el ajuste de intensidad de los filtros.
- Activación rápida mediante un ícono en la barra de herramientas del navegador.
- Compatible con múltiples idiomas.
- No requiere configuración avanzada ni conocimientos técnicos.
- Gratuita y de fácil instalación desde la *Chrome Web Store*.

En la Figura 1.8 se muestra la interfaz de la extensión de *Colorblind* una vez se encuentra instalada en el navegador.

Figura 1.8

Interfaz de *Colorblind* extension://afcafnelafcgjinkaeohkalmfececool/src/popup.html



1.5.2 *SightGlow*

SightGlow es una extensión web centrada en la mejora de la accesibilidad digital para personas con daltonismo, especialmente aquellas con deficiencia rojo-verde. Su enfoque está en transformar la experiencia de navegación mediante ajustes visuales personalizados que facilitan la distinción de colores y la lectura de contenido.

Características principales:

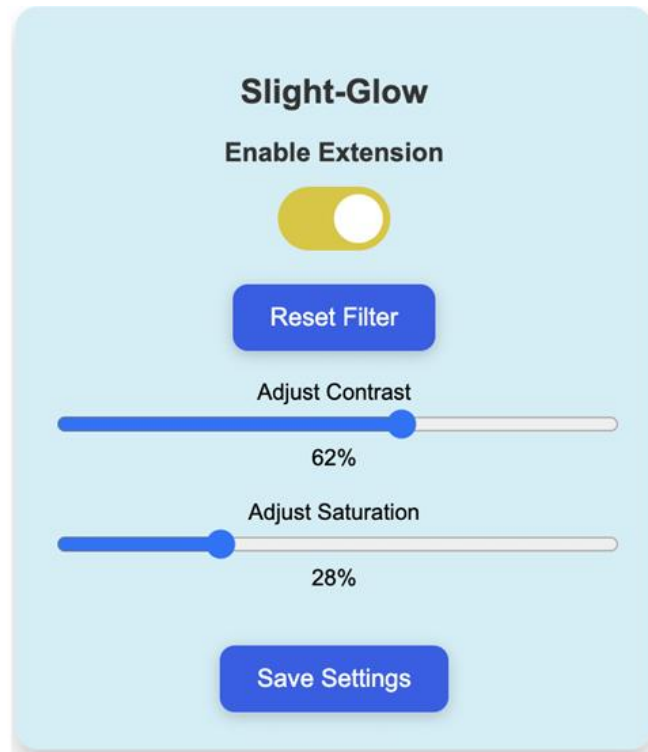
- Filtros de color personalizados que permiten la modificación de los colores en pantalla según las necesidades del usuario.
- *Zoom* selectivo para ampliación de secciones específicas del texto.
- Interfaz accesible con botones grandes y diseño intuitivo.

- Desarrollo colaborativo con retroalimentación de personas con daltonismo.
- También útil para personas con pérdida de visión relacionada con la edad.
- Mejora la experiencia en redes sociales, sitios *web* y plataformas digitales.

La Figura 1.9 se aprecia la interfaz de la extensión de *SightGlow*, en la cual se aprecia el ajuste del contraste, el reinicio de del contraste y la saturación.

Figura 1.9

Interfaz de *SightGlow* <https://arxiv.org/pdf/2412.11004>



1.5.3 NoCoffee – Vision Simulator

NoCoffee es una extensión para navegadores que simula diversas discapacidades visuales, incluyendo varios tipos de daltonismo. Fue creada por Aaron Leventhal y está pensada principalmente para desarrolladores web que desean evaluar la accesibilidad de sus sitios.

Características principales:

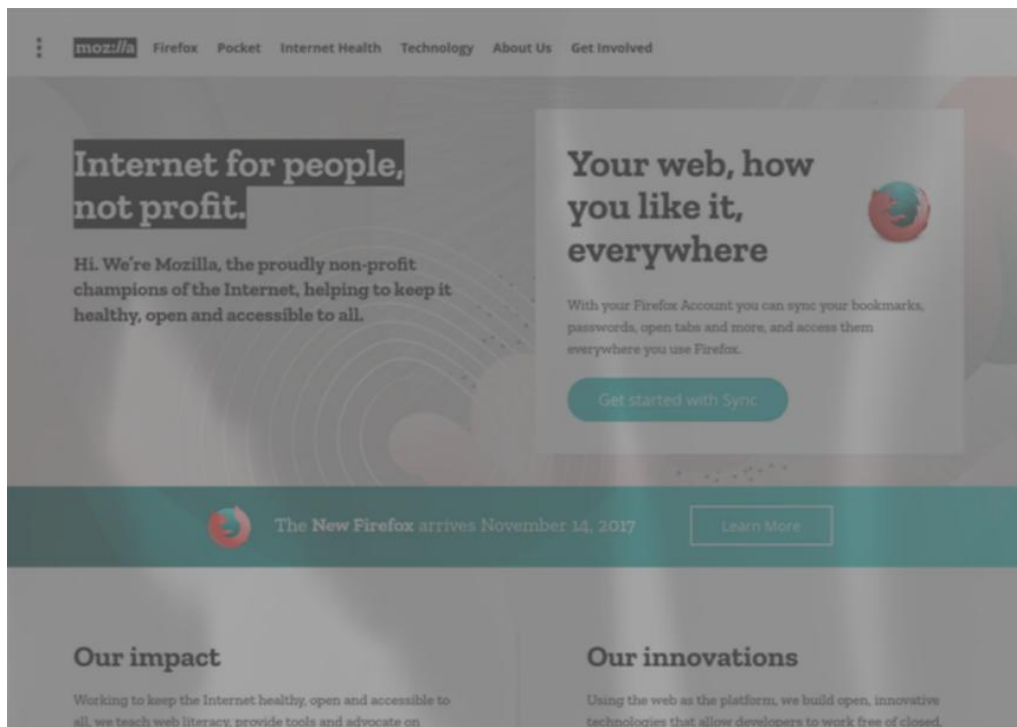
- Simulación de múltiples condiciones visuales: daltonismo, baja visión, visión borrosa, pérdida de contraste, entre otras.
- Compatible con Chrome y Firefox.

- Herramienta educativa para diseñadores y desarrolladores interesados en accesibilidad.
- Permite la visualización de cómo una persona con discapacidad visual percibe un sitio *web*.
- Código abierto disponible en GitHub.
- Ayuda a identificar problemas de diseño que afectan la inclusión digital.

En la Figura 1.10 se visualiza una vista previa del filtro de NoCoffee, haciendo uso del filtro de visión borrosa.

Figura 1.10

Vista previa del filtro de NoCoffee <https://accessgarage.wordpress.com/2013/02/09/458/>



1.5.4 Colorblindly

Colorblindly es una extensión para Chrome que permite simular cómo se ve una página web desde la perspectiva de una persona con daltonismo. Está orientada a desarrolladores y diseñadores que buscan crear interfaces más inclusivas.

Características principales:

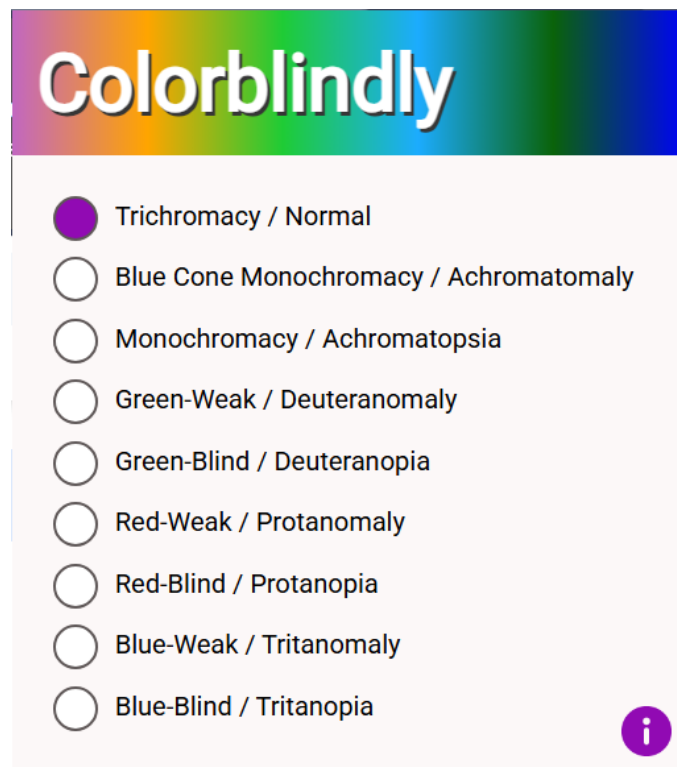
- Simulación de ocho tipos de daltonismo, incluyendo protanopia, deuteranopia, tritanopia, monocromacia y sus variantes débiles.
- Basada en el modelo de simulación de *Brettel*, reconocido por su precisión.

- Fácil de usar: se activa en cualquier sitio *web* (excepto páginas internas de Chrome).
- Herramienta útil para pruebas de accesibilidad visual.
- Gratuita y de código abierto.
- Permite evaluar y ajustar esquemas de color en tiempo real.

En la Figura 1.11 se muestra la interfaz de la extensión *Colorblindly* en la cual se muestran las diferentes opciones en la cuales se aplica el filtro de daltonismo.

Figura 1.11

Interfaz de la extensión *Colorblindly* extension://floniaahm ccleoclineebhbmjgdfjgg/popup/popup.html



En la Tabla 1.2 se comparan diferentes puntos de cada uno de los proyectos existentes evaluando: tipos de daltonismo simulados, método de simulación, plataforma, precisión en la simulación, interacción con el usuario, accesibilidad, compatibilidad con navegadores, código abierto, resultados del test, generación de reportes y análisis de datos.

Tabla 1.2
Cuadro comparativo de extensiones

Característica	Dalton (Colorblind Extension)	SightGlow	NoCoffee – Vision Simulator	Colorblindly	DaltonAssist
Tipos de Daltonismo Simulados	Protanopia, Deuteranopia, Tritanopia	Principalmente rojo-verde	Deuteranopia, Protanopia, Tritanopia	Deuteranopia, Protanopia, Tritanopia	Deuteranopia, Protanopia, Tritanopia
Método de Simulación	Filtros de corrección cromática	Ajuste de colores personalizados	Simulación visual directa sobre el sitio web	Modelo de simulación de Brettel	Inyecciones CSS por medio de JavaScript
Plataforma	Extensión para Chrome	Extensión web	Extensión web	Extensión para Chrome	Extensión de navegador
Precisión en la Simulación	Media	Media	Alta	Alta	Alta
Interacción con el Usuario	Controles deslizantes simples	Botones grandes, zoom selectivo	Activación directa, sin interacción compleja	Activación rápida, selección de tipo de daltonismo	Formulario inicial para datos del usuario y test de Ishihara
Accesibilidad	Enfocada en usuarios con daltonismo	También útil para adultos mayores	Enfocada en desarrolladores y diseñadores	Enfocada en desarrolladores y diseñadores	No diseñado para otras alteraciones visuales
Compatibilidad con Navegadores	Chrome, Edge	Chrome	Chrome, Firefox	Chrome	Probado en Edge y Chrome, esperado en Mozilla y Brave
Código Abierto	No	No	Sí	No	No
Resultados del Test	N/A	N/A	N/A	N/A	Test de Ishihara con 24 imágenes, 5 resultados posibles
Generación de Reportes	N/A	N/A	N/A	N/A	Envío de resultados por correo y generación de PDF
Análisis de Datos	N/A	N/A	N/A	N/A	Análíticas mediante Google Charts

La Tabla 1.2 permite identificar que *DaltonAssist* supera las capacidades de las otras extensiones con las cuales se compara, al ofrecer funciones de diagnóstico y gestión de datos inexistentes en estas. Su ventaja principal radica en que no solo es una herramienta de accesibilidad sino también un sistema de evaluación, permitiendo tener un seguimiento de los resultados a través de reportes y gráficas de análisis.

Capítulo 2. Desarrollo

En este capítulo se describe el proceso de desarrollo del proyecto *DaltonAssist*, una extensión para navegadores, que proporciona una solución a los problemas que presentan las personas con daltonismo durante la navegación en la *web*. Se aplicará la metodología *Kanban* como se mencionó, anteriormente en el marco metodológico en el marco metodológico.

2.1 Planificación

Dentro de esta fase se definen las bases para el desarrollo de *DaltonAssist*, estableciendo las historias de usuario, las tareas y la creación del tablero.

Esta fase es esencial para garantizar que los requerimientos del sistema, optimizando tiempos y asegurando que el proyecto sea exitoso.

2.1.1 Creación de las historias de usuario

En este apartado se definirán las historias de usuario tomando en cuenta cuatro roles el de usuario, investigador, desarrollador y tester.

En la Tabla 2.1 contine todas las historias de usuario presentadas para la primera fase del proyecto, haciendo uso de cuatro roles siendo estos Investigador, Tester, Desarrollador y Usuario.

Tabla 2.1
Tabla de historias de usuario

Tipo de usuario	Historia de usuario
Investigador	Como investigador, quiero acceder a estudios previos sobre daltonismo digital para fundamentar el desarrollo de soluciones accesibles.
Investigador	Como investigador, quiero buscar bases de datos con perfiles de usuarios con daltonismo para entender mejor sus necesidades visuales.
Investigador	Como investigador, quiero explorar cómo otras extensiones o herramientas abordan la accesibilidad para personas con daltonismo, para identificar buenas prácticas.
Investigador	Como investigador, quiero consultar artículos científicos sobre percepción del color en personas con distintos tipos de daltonismo, para enriquecer el diseño de filtros.
Investigador	Como investigador, quiero recopilar información sobre estándares internacionales de accesibilidad visual, para asegurar que la extensión cumpla con ellos.
Investigador	Como investigador, quiero investigar qué navegadores ofrecen soporte nativo para corrección de color, para orientar el desarrollo hacia plataformas más abiertas.
Investigador	Como investigador, quiero buscar testimonios de usuarios

Tipo de usuario	Historia de usuario
	con daltonismo sobre su experiencia en la web, para identificar puntos de dolor comunes.

Tabla 2.1
(Continuación)

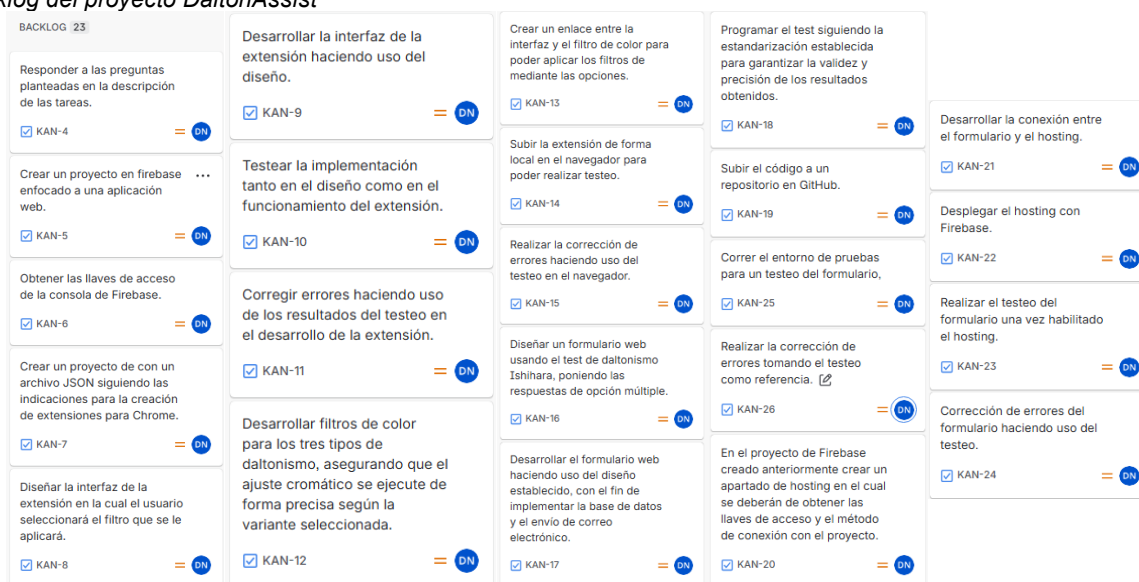
Tipo de usuario	Historia de usuario
Investigador	Como investigador, quiero revisar documentación técnica de APIs relacionadas con accesibilidad visual, para proponer integraciones útiles.
Tester	Como tester, quiero probar la extensión en diferentes dispositivos para asegurar su compatibilidad.
Tester	Como tester, quiero verificar que los filtros no afecten negativamente la legibilidad de textos.
Tester	Como tester, quiero validar que la extensión no interfiera con otras funcionalidades del navegador.
Tester	Como tester, quiero simular diferentes tipos de daltonismo para comprobar la efectividad de cada filtro.
Desarrollador	Como desarrollador, quiero implementar filtros de corrección de color para cada tipo de daltonismo.
Desarrollador	Como desarrollador, quiero que la extensión sea compatible con los principales navegadores.
Desarrollador	Como desarrollador, quiero recibir reportes de errores para poder corregirlos rápidamente.
Desarrollador	Como desarrollador, quiero que la extensión tenga un diseño accesible y fácil de usar.
Desarrollador	Como desarrollador, quiero automatizar pruebas de funcionalidad para asegurar que los cambios no rompan el sistema.
Desarrollador	Como desarrollador, quiero documentar el código para facilitar el mantenimiento y colaboración.
Usuario	Como usuario, quiero que los colores se ajusten automáticamente para poder distinguirlos mejor.
Usuario	Como usuario, quiero que los tonos rojos se modifiquen para que no se confundan con el verde.
Usuario	Como usuario, quiero que los verdes se transformen en tonos más visibles para mí.
Usuario	Como usuario, quiero que los azules se adapten para que no se confundan con el amarillo.
Usuario	Como usuario, quiero seleccionar el tipo de daltonismo que tengo para que la extensión se adapte a mi visión.
Usuario	Como usuario, quiero ajustar la intensidad del filtro para encontrar el nivel que más me ayuda.

2.1.2 Definición de las tareas del proyecto

En este apartado se muestra cómo se realiza una tarea, para este proyecto se usarán tres tipos de tareas, una para la investigación, la segunda para desarrollo y finalmente la tercera para la implementación.

En la Figura 2.1 se muestra el *backlog* de las tareas, estas ya se encuentran acomodadas por prioridad empezando por la de investigación y terminando por la de implementación.

Figura 2.1
Backlog del proyecto DaltonAssist



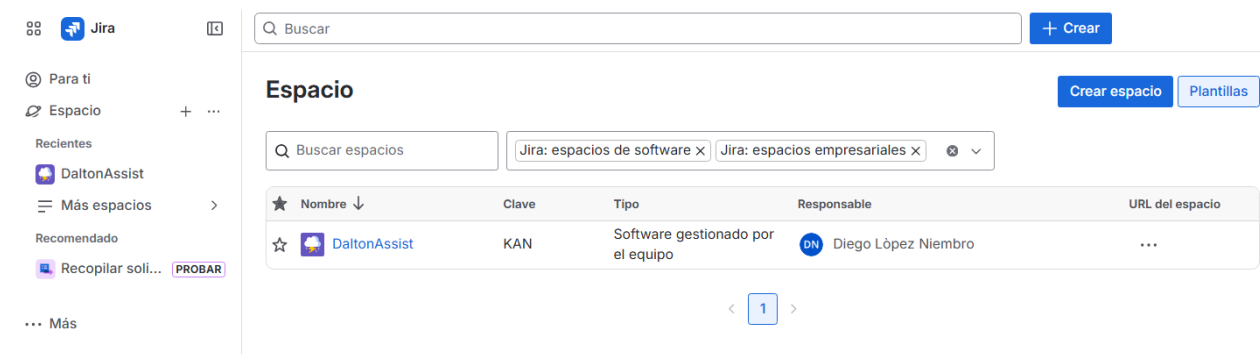
2.1.3 Creación del tablero Kanban

Para la gestión visual de las tareas y el control del flujo de trabajo, se creó un tablero *Kanban* utilizando la herramienta *Jira Software*, desarrollada por *Atlassian*. *Jira* facilita la aplicación de metodologías ágiles como *Kanban* y *Scrum*, ofreciendo un entorno colaborativo en el que es posible la creación, asignación y seguimiento de tareas de manera organizada.

El tablero se estructuró con columnas que representan las distintas fases del ciclo de desarrollo, estas columnas son: *backlog*, pendiente (*To Do*), en curso (*Doing*), terminado (*Done*).

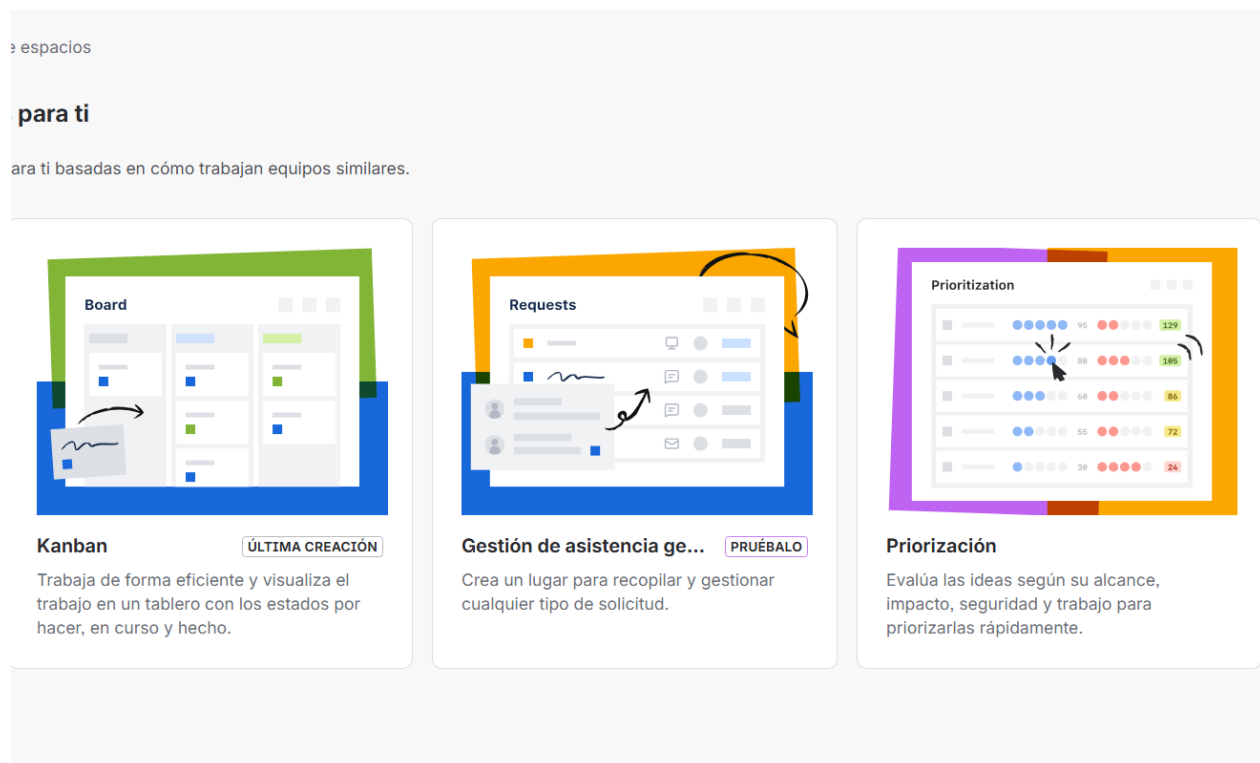
En la Figura 2.2 se ve la pestaña de creación de espacios en *Jira*.

Figura 2.2
Pestaña de creación de tableros en Jira.



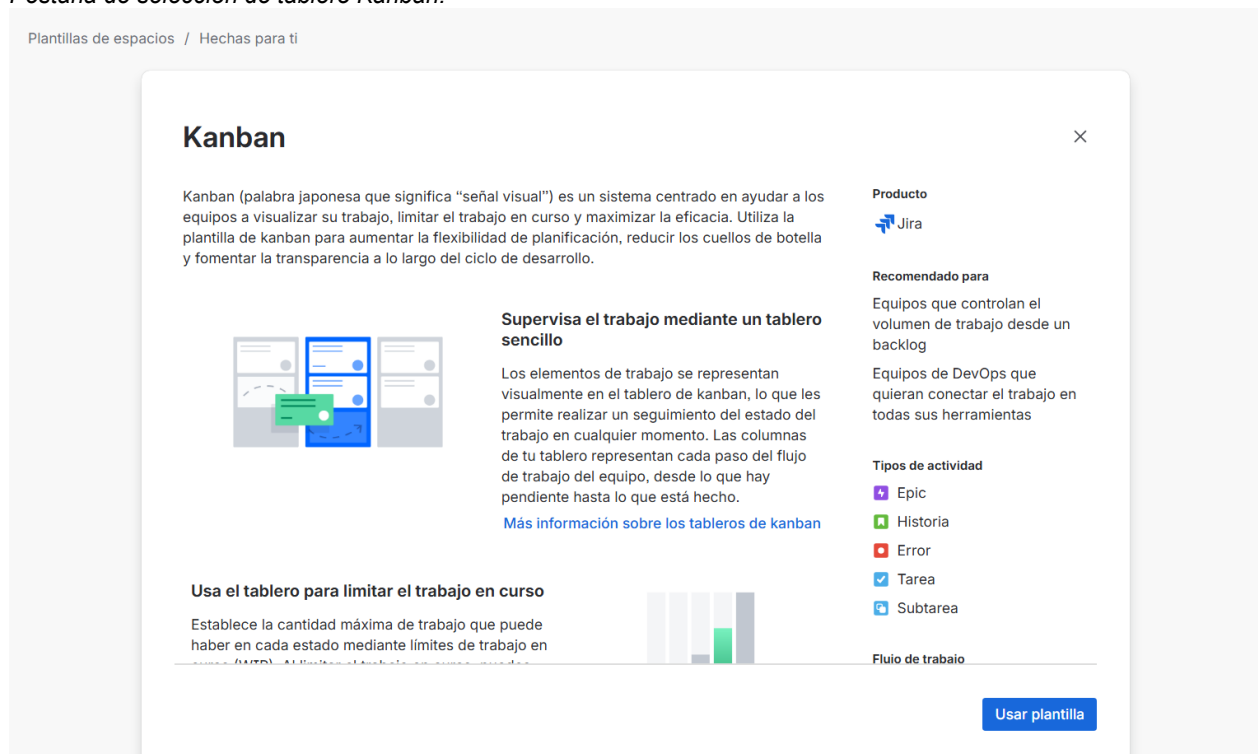
La Figura 2.3 la selección de plantillas que proporciona *Jira* para sus tableros.

Figura 2.3
Pestaña de selección de plantillas de Jira.



En la Figura 2.4 se aprecia lo que proporciona *Kanban*, así como un ejemplo de cómo se usa y las ventajas que tiene.

Figura 2.4
Pestaña de selección de tablero Kanban.



Una vez que se halla dado clic en el botón de “Usar plantilla” se abrirá la pestaña de selección de tipo de espacio de trabajo con la plantilla de *Kanban* tal como lo muestra la Figura 2.5.

Figura 2.5
Pestaña de elección de tipo de espacio.



Se seleccionará la opción de gestionarlo por equipo lo que llevará a una pestaña de creación de proyecto donde se asignará el nombre y permite cambiar los aspectos anteriores tal como lo muestra la Figura 2.6.

Figura 2.6

Pestaña de Creación de Proyecto.

← Volver a las plantillas de proyectos

Crear proyecto

Los campos obligatorios están marcados con un asterisco *

Nombre *

Prueba con el nombre del equipo, la meta del proyecto, el l

Tipo de espacio

Personas Gestionados por el equipo

Permisos de proyectos

Elige quién puede ver, editar y comentar

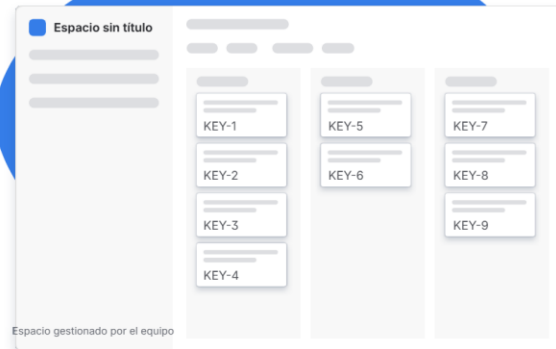
Con los permisos del proyecto, controla qué personas pueden acceder y lo que pueden hacer.

[Pruébalo gratis durante 14 días](#)

Plantilla

Cambiar plantilla

**Kanban**
 **Jira**
Visualiza tu espacio y hazlo avanzar utilizando actividades en un tablero lleno de posibilidades.



Una vez asignado el nombre te dará la opción de añadir más participantes como se muestra en la Figura 2.7.

Figura 2.7

Pestaña de Añadir Participantes.

Tráete al equipo

Añade a las personas con las que ya hayas trabajado en Jira, o invita a alguien nuevo.

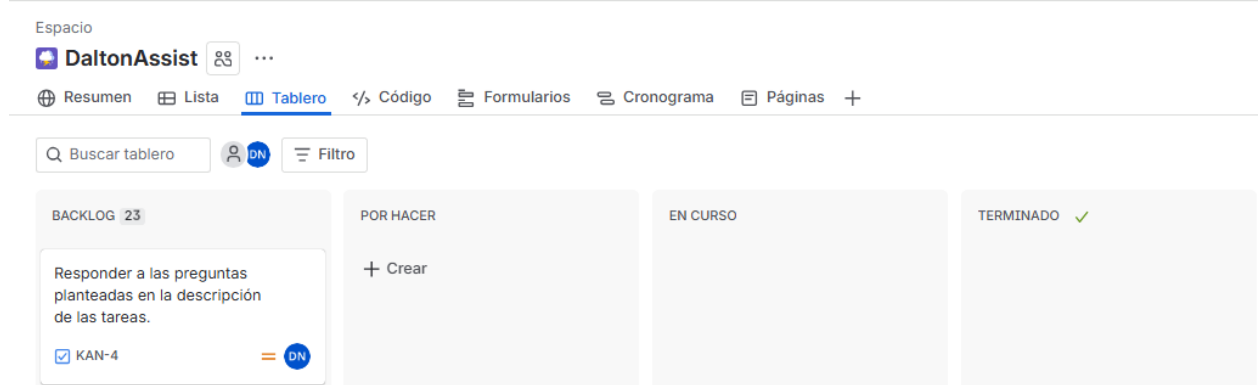
Introducir nombres o correos electrónicos

Rol

Administrador

Se añadirán los participantes, en caso contrario, se dará lugar a omitir y se mostrará el *dashbord* de trabajo del tablero como en la Figura 2.8.

Figura 2.8
Tablero de DaltonAssist en Jira.



En este tablero se creará las columnas correspondientes para su flujo de trabajo asignado en cada caso.

2.2 Limitaciones del proceso (*WIP*)

En este apartado se explicará cómo es que se realizó el manejo de las tareas y como se establecieron los límites del trabajo.

Los objetivos de las limitaciones son:

- Reducir el tiempo de entrega por tarea.
- Evitar la acumulación de tareas en la fase de desarrollo.
- Mejorar la calidad del producto evitando sobrecarga de trabajo.
- Facilitar el movimiento y la revisión de las tareas.

Con la finalidad de establecer un flujo de trabajo eficiente y evitar la sobrecarga de tareas el límite de trabajo se estableció un límite de tres tareas en el apartado de “en curso”, permitiendo el enfoque de tareas y en caso de que se presentara problema alguno introducir una tarea con la finalidad de solucionar el problema que se presente al momento del *testing*.

2.3 Ciclo de desarrollo

En este apartado se definirán los ciclos de trabajo (*Sprints*) que se realizaron en la elaboración de cada una de las partes del proyecto, comenzando con la investigación y finalizando con el *hosting*.

Las tareas se desplegarán todas las del *sprint*, una por una, en las tareas de desarrollo se mostrará un fragmento de código.

Sprint-1 Investigación

En este primer *sprint* se realizará la investigación de todas las preguntas que se plantearon para poder llevar a cabo el proyecto, la Figura 2.9 se muestra el flujo de trabajo en el tablero.

Figura 2.9

Flujo de datos del sprint-1.



En la Figura 2.10 se ve la tarea desplegada, la cual contiene toda la información acerca de la tarea.

Figura 2.10

Tarea desplegada

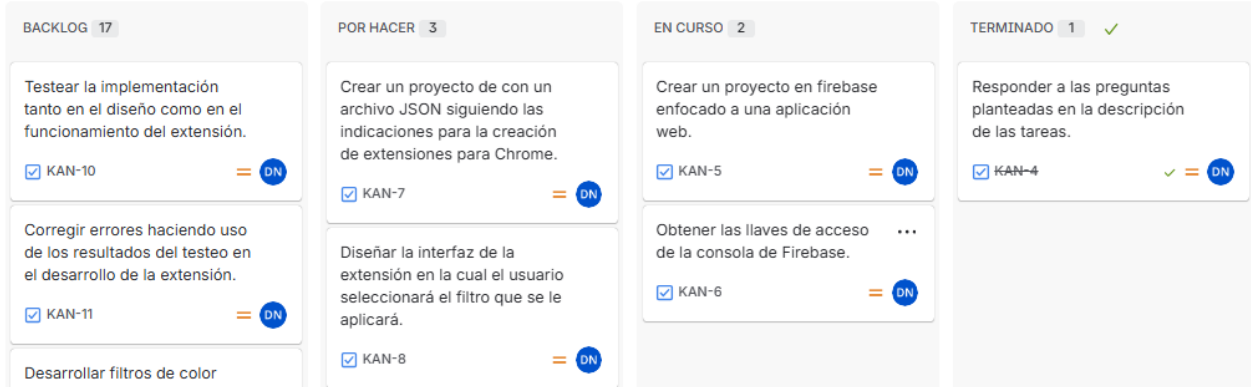
Sprint-2 Creación del Proyecto en Firebase

En este *sprint* se empezará con la parte de desarrollo en la cual se creará un proyecto de *Firebase* utilizando su formato de aplicación *web*. Lo que nos lleva a dos tareas la creación del propio proyecto y después la obtención de las llaves para su implementación en el proyecto.

En la Figura 2.11 se muestra el flujo de trabajo de el *sprint-2* donde se hace la creación y configuración del proyecto de *Firebase*, así mismo se realiza la obtención de llaves de acceso a la conexión.

Figura 2.11

Flujo de datos del sprint-2.



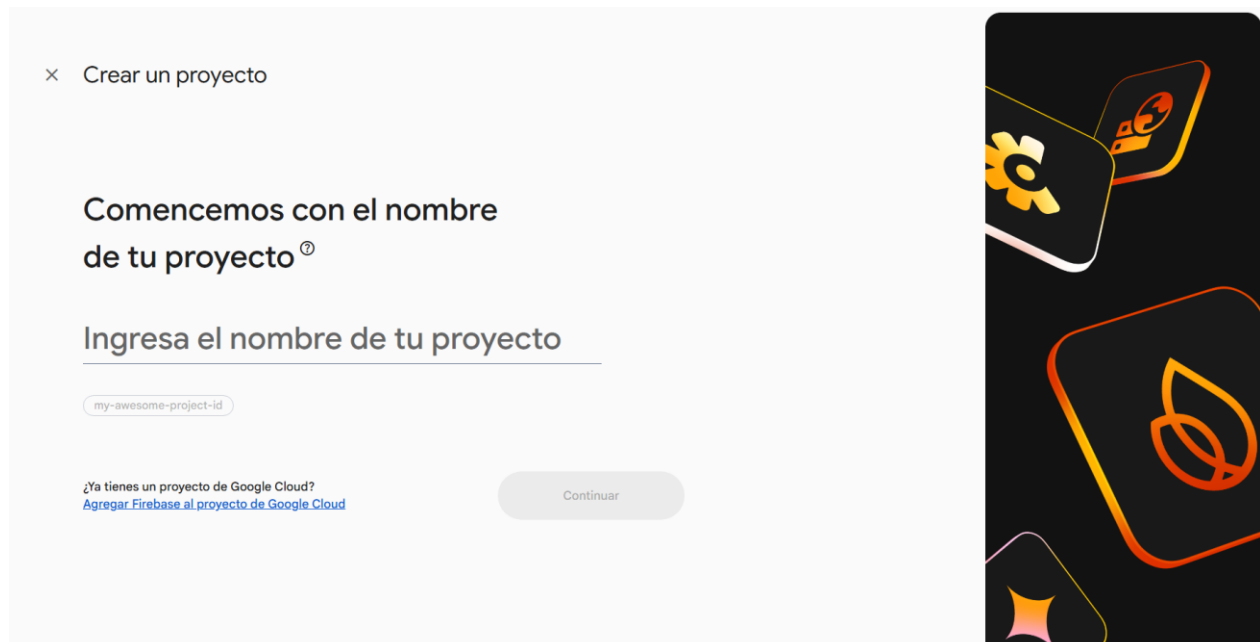
- Crear Proyecto en Firebase

El primer paso consiste en configurar un proyecto dentro de la consola de *Firebase*. Una vez creada la cuenta, se debe acceder a *Firebase Console* y hacer clic en el botón “Crear proyecto”.

A continuación, se mostrará una ventana como la ilustrada en la Figura 2.12, donde se debe ingresar el nombre del proyecto. Se recomienda que este nombre esté relacionado con el objetivo de la base de datos para facilitar su identificación en etapas posteriores.

Figura 2.12

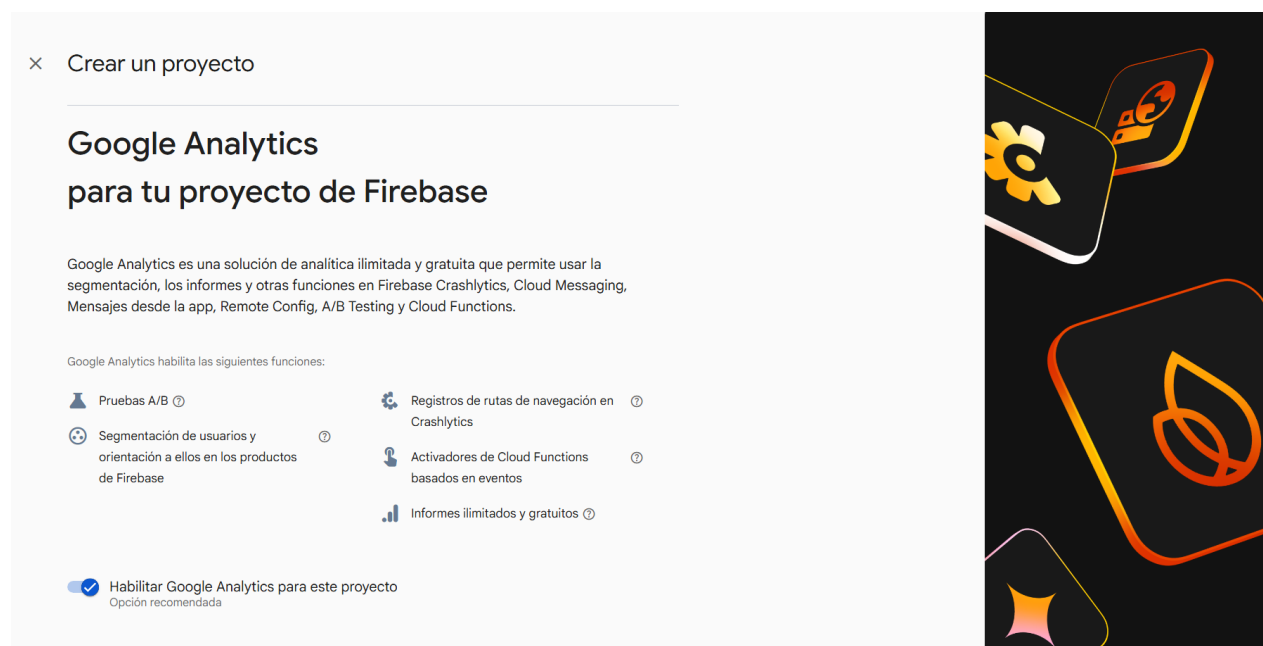
Página de designación de nombre al proyecto.



Si la cuenta utilizada para crear el proyecto no tiene el dominio “@gmail.com”, será necesario seleccionar un recurso superior. Para ello, se debe hacer clic en el botón “Seleccionar recurso superior”, lo que desplegará una ventana modal con el dominio del correo electrónico. Se selecciona el dominio correspondiente y se confirma con el botón “Listo”. Luego, se continúa con el proceso haciendo clic en “Continuar”.

Posteriormente, se mostrará una ventana como la Figura 2.13, donde se ofrece la opción de activar *Google Analytics*. Esta funcionalidad puede habilitarse en este momento o más adelante, según las necesidades del proyecto.

Figura 2.13
Página de activación de Google Analytics.



Una vez tomada la decisión, se activa o desactiva la opción y se hace clic en “Continuar”. *Firebase* procederá a crear el proyecto, lo cual puede tardar unos segundos. Al finalizar, se redirigirá al panel principal donde se listan los proyectos creados.

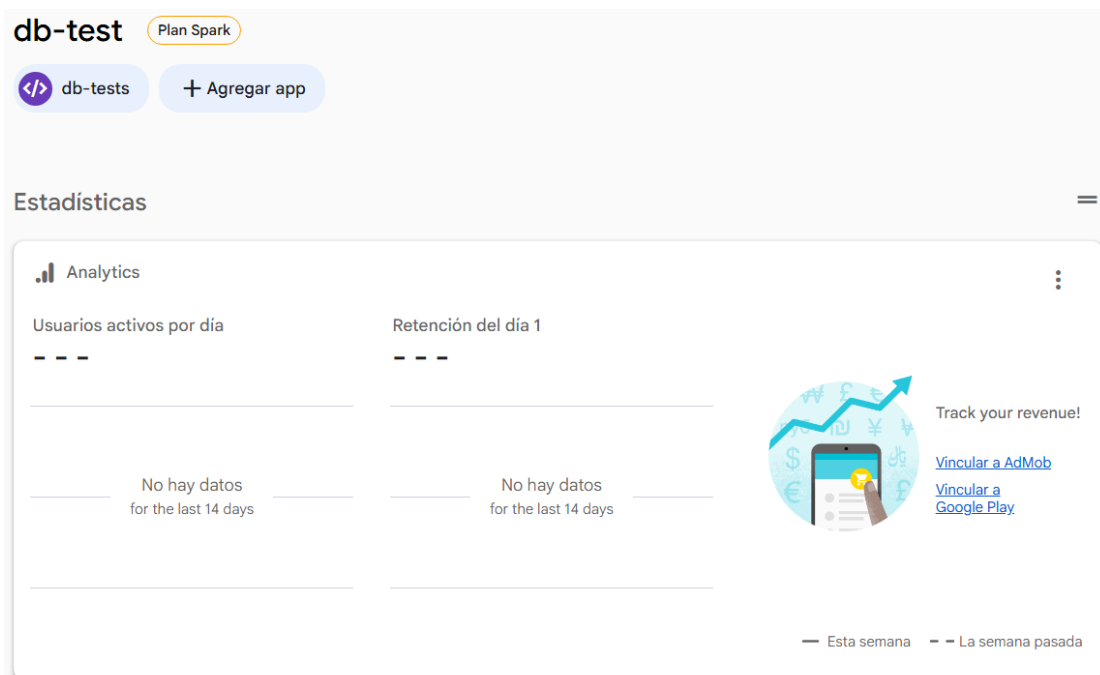
La Figura 2.14 muestra cómo se visualiza la pestaña “Tus proyectos”. Se debe seleccionar el proyecto recién creado para acceder a su consola principal.

Figura 2.14
Página de proyectos creados en Firebase.



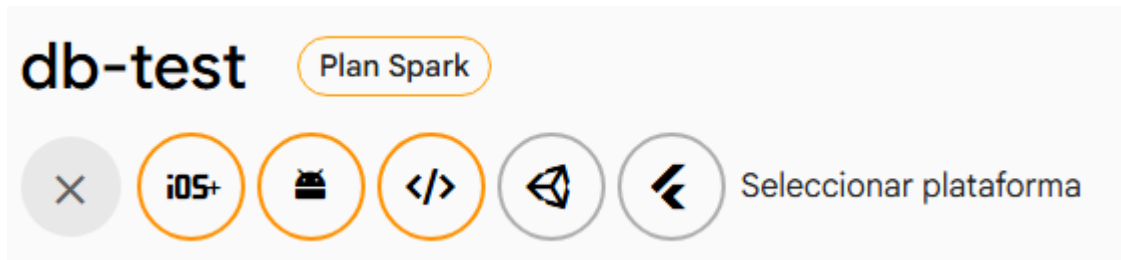
Al ingresar al proyecto, se cargará una interfaz como la de la Figura 2.15. Esta es la consola principal desde donde se pueden administrar los servicios asociados al proyecto.

Figura 2.15
Layout básico del proyecto en Firebase.



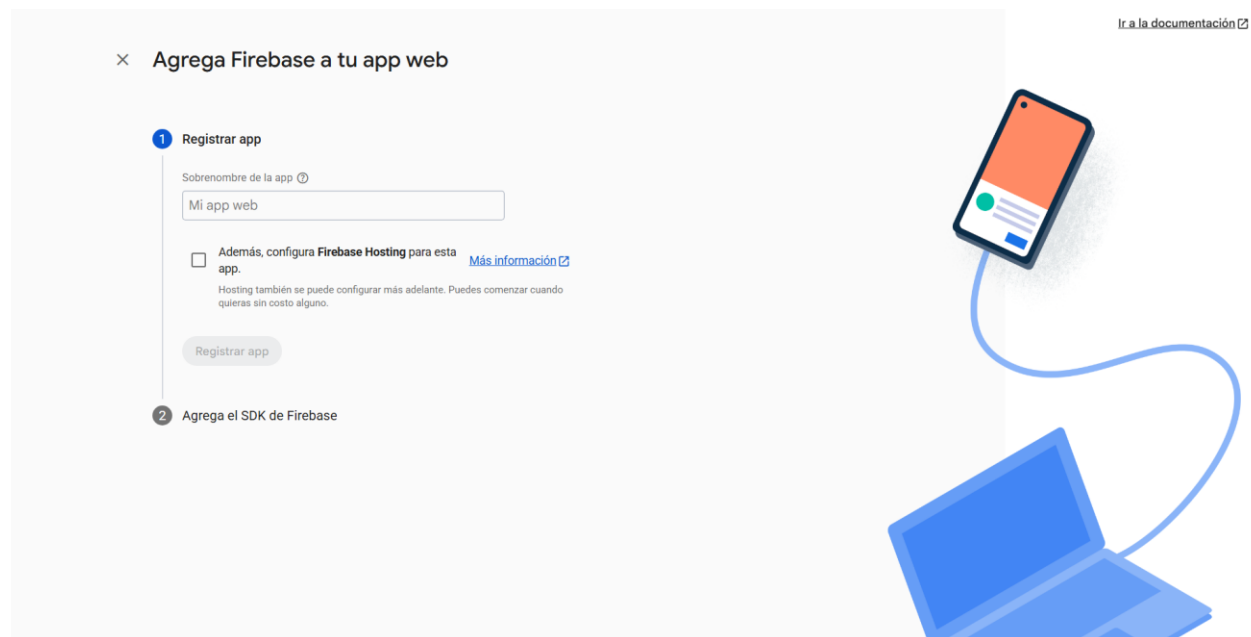
Para vincular *Firebase* con una aplicación *web*, se debe hacer clic en el apartado “Comienza por agregar *Firebase* a tu app” y seleccionar la opción correspondiente a aplicaciones *web*, como se muestra en la Figura 2.16.

Figura 2.16
Interfaz de elección de tipo de aplicación.



Se abrirá una nueva pestaña donde se debe ingresar el nombre de la aplicación. También se ofrece la opción de configurar *Firebase Hosting*, aunque esta puede activarse posteriormente. La interfaz se muestra en la Figura 2.17.

Figura 2.17
Página de asignación de nombre a la aplicación web.

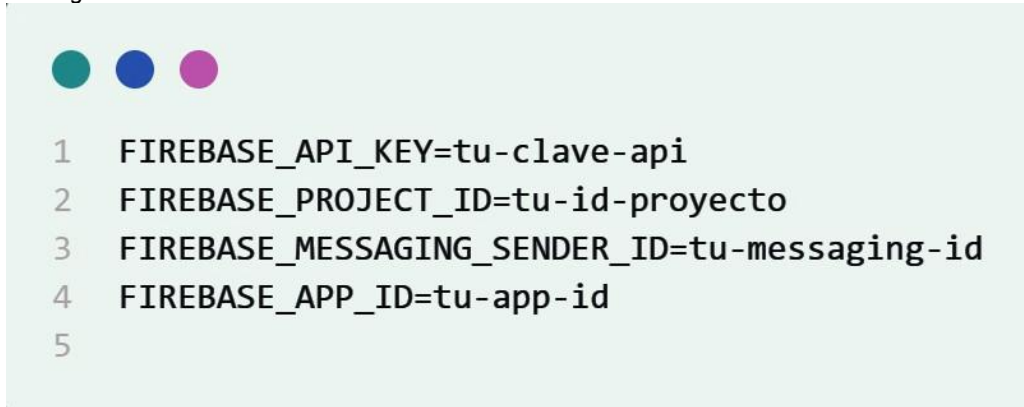


Al registrar la aplicación, *Firebase* proporcionará las credenciales necesarias para la integración. Estas deben almacenarse en un archivo `.env` dentro del proyecto local, el cual no debe compartirse públicamente, ya que contiene llaves de acceso sensibles.

Ejemplo de configuración en el archivo `.env` se muestra en la Figura 2.18.

Figura 2.18

Ejemplo de configuración de las llaves de Firebase.

A screenshot of a code editor with a light blue background. At the top left, there are three colored circles: teal, blue, and pink. Below them, there is a list of five lines of code, each preceded by a number from 1 to 5. The code is in a monospaced font and represents Firebase configuration variables for an environment file.

```
1  FIREBASE_API_KEY=tu-clave-api
2  FIREBASE_PROJECT_ID=tu-id-proyecto
3  FIREBASE_MESSAGING_SENDER_ID=tu-messaging-id
4  FIREBASE_APP_ID=tu-app-id
5
```

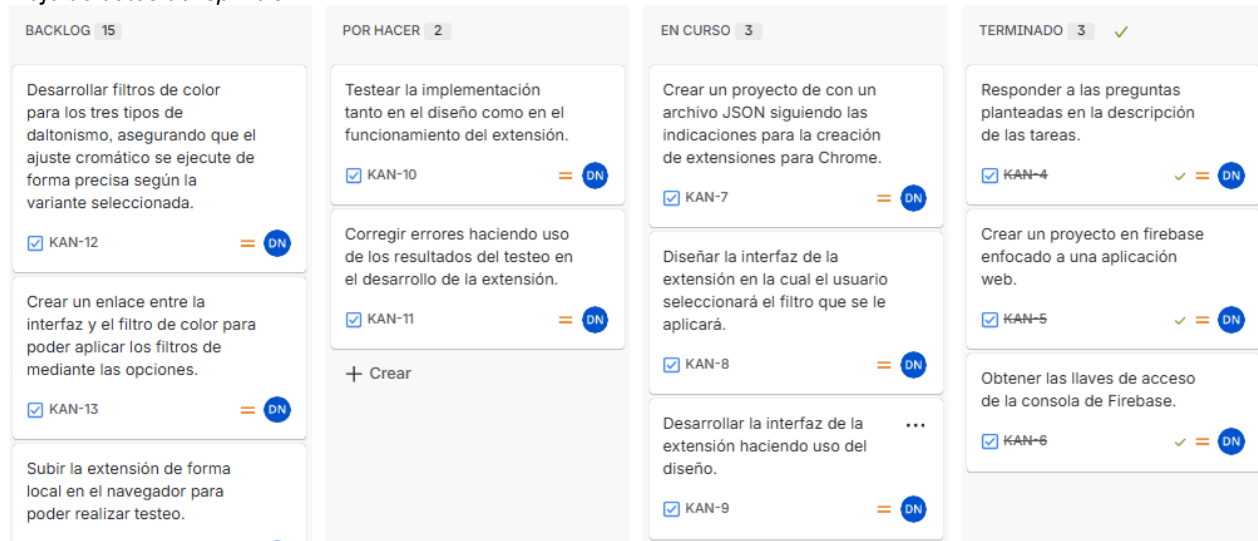
Para habilitar la base de datos, se debe acceder al apartado “*Firestore Database*” dentro del panel de compilación del proyecto. Al hacer clic, se abrirá una nueva pestaña donde se debe seleccionar “Crear base de datos”.

Se mostrará una ventana modal solicitando la elección del servidor más cercano a la ubicación actual. Luego, se debe seleccionar el modo de operación: prueba o producción, según convenga al entorno de desarrollo. Finalmente, se creará la base de datos, la cual estará lista para almacenar y consultar información en tiempo real.

Sprint-3 Desarrollo de la Extensión

En el *sprint* actual se empieza con la escritura de código por medio de tres tareas siendo estas la configuración del archivo *JSON* el cual dictará cómo se comporta la extensión en el navegador y de donde trae los demás archivos, el diseño de la interfaz con la cual interactuará el usuario y el desarrollo de la misma interfaz, tal y como se muestra en la Figura 2.19.

Figura 2.19
Flujo de datos del sprint-3.



Para la parte de diseño se buscó referencias de una interfaz sencilla la cual permitiera un fácil manejo de la extensión, buscando algo sin muchos colores y directo al propósito de la extensión.

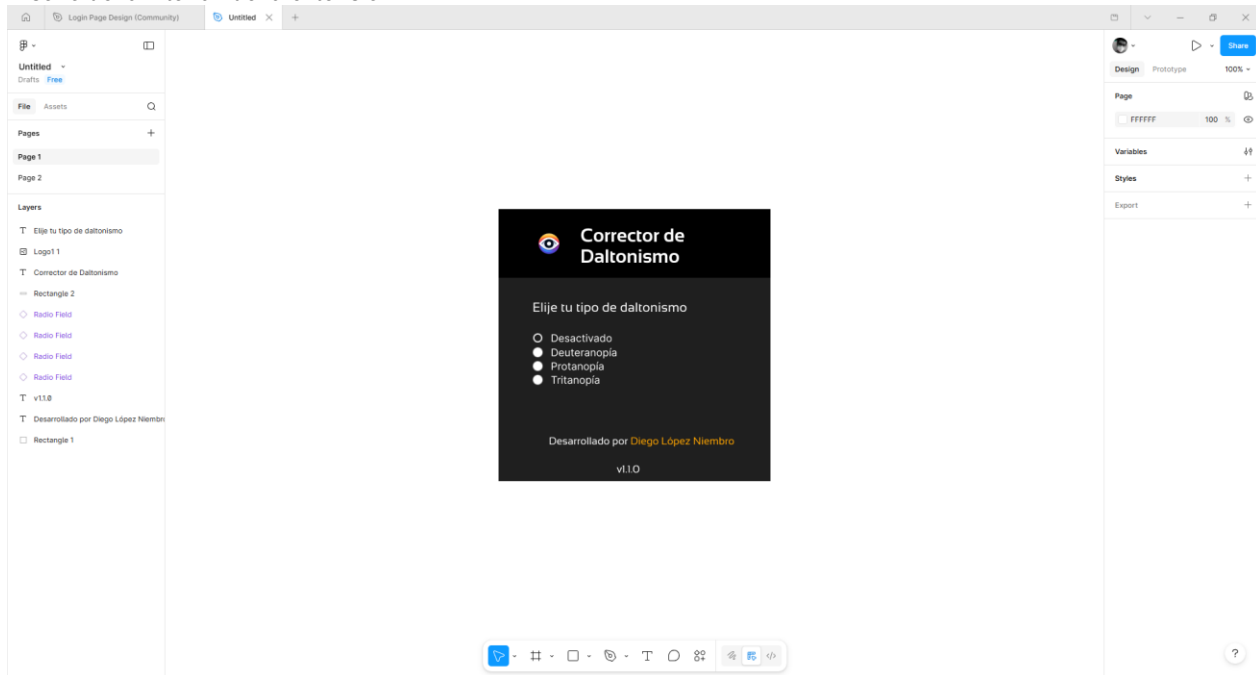
Se hizo uso de *Figma* para el desarrollo de las interfaces para tener una flexibilidad y visión al momento de realizar el diseño.

Se utilizó un marco (*frame*) para el diseño, se utilizaron dos *frames* para el esqueleto de la interfaz uno que contenía el título y el segundo para el cuerpo. Dentro del primer *frame* se acomodó el título y el logo de la extensión, en el segundo se colocó un *header* instructivo, cuatro inputs para las opciones, dos párrafos para el creador y otro para la versión de la extensión.

En la figura 2.20 se muestra el diseño del interfaz conseguido.

Figura 2.20

Diseño de la interfaz de la extensión.



Una vez realizado el diseño de la extensión, se inicia el desarrollo de la misma con la configuración del archivo *JSON*, el cual será utilizado por el navegador para poder reconocer la extensión.

El archivo *JSON* se construye, dependiendo de la versión que se utiliza, empezando por el nombre, una descripción, el número de versión del proyecto, el autor, las restricciones locales y del *host*, y finalmente, los *scripts* y los archivos que conforman la extensión.

En la Figura 2.21 se muestra un fragmento de código del archivo *JSON*.

Figura 2.21

Archivo JSON.

```
1  {
2    "manifest_version": 3,
3    "name": "Daltonismo Extencion",
4    "version": "1",
5    "description": "Extencion utilizada para poder
6    "author": "Diego López Niembro",
7    "permissions": ["activeTab", "storage"],
8    "host_permissions": ["*://*/*"],
9    "action": {
10     "default_popup": "menu/index.html",
11     "default_icon": {
12       "128": "imagenes/Logo1.png"
13     }
14   }
```

Lo siguiente por hacer es definir una carpeta en la que se guardarán la interfaz y otra en la que se guardarán las imágenes que se empleen en la extensión.

Se hará uso de un archivo *HTML* y otro *CSS* para el desarrollo. El archivo *HTML* incluye la estructura de la página en la cual se usan inputs para seleccionar los filtros y el archivo *CSS* se usará para modificar el archivo *HTML* conforme al diseño realizado.

En la Figura 2.22 se muestra un fragmento de código del archivo *HTML*.

Figura 2.22

Fragmento de código del archivo *HTML*.

```
1 <main>
2   <div class="instructions">Elije tu tipo de daltonismo</div>
3   <div class="options">
4     <div class="mode-option">
5       <input type="radio" id="modeDisabled" name="displayMode" value="0" checked />
6       <label for="modeDisabled">Desactivado</label>
7     </div>
8     <div class="mode-option">
9       <input type="radio" id="modeDeuteranopia" name="displayMode" value="1" />
10      <label for="modeGreen">Deuteranopia</label>
11    </div>
12    <div class="mode-option">
13      <input type="radio" id="modeProtanopia" name="displayMode" value="2" />
14      <label for="modeRed">Protanopia</label>
15    </div>
16    <div class="mode-option">
17      <input type="radio" id="modeTritanopia" name="displayMode" value="3" />
18      <label for="modeBlue">Tritanopia</label>
19    </div>
20  </div>
21 </main>
```

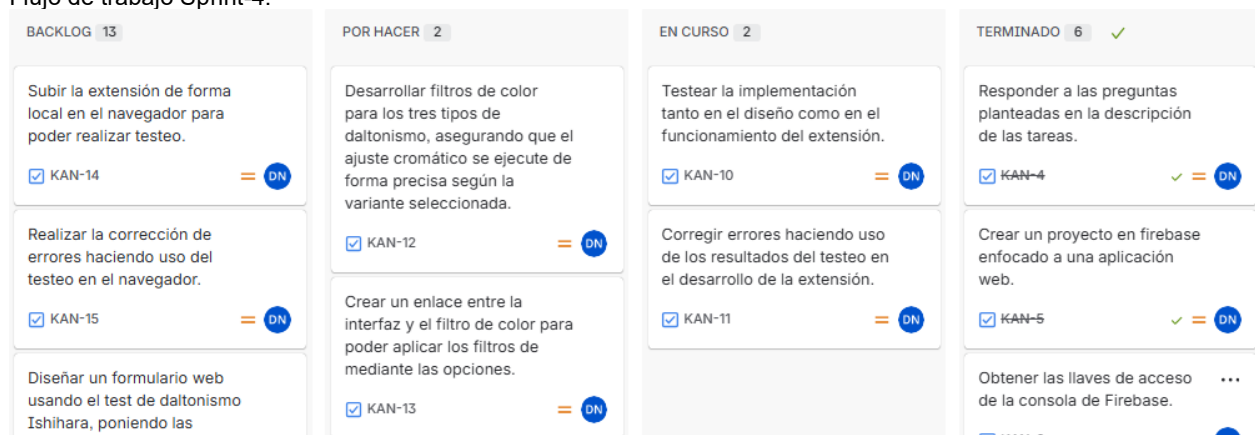
Sprint-4 Implementación Extensión

En este *sprint* se realiza el testeo de la extensión y la corrección de errores que se presenten, por medio del entorno de desarrollo.

En la Figura 2.23 se muestra el flujo de trabajo del *sprint* donde se encuentran dos tareas la de testeo y la de corrección de errores.

Figura 2.23

Flujo de trabajo Sprint-4.



En la Figura 2.24 se muestra cómo se visualizaron las pruebas para el desarrollo junto con un fragmento del código.

Figura 2.24

Ejemplo de testeo y corrección de errores.



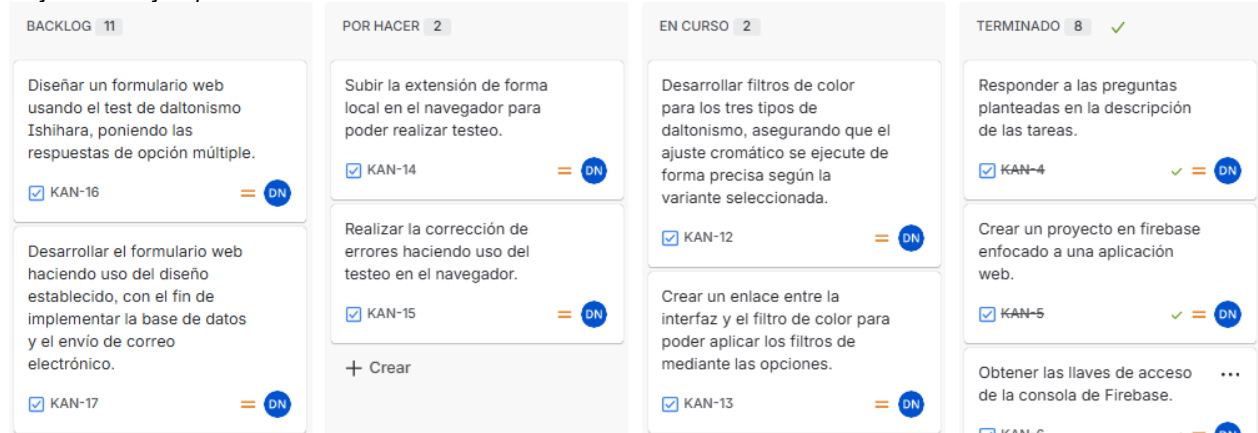
Sprint-5 Filtro de color

En este *sprint* se desarrolla el filtro de color, así como su vinculación con la interfaz para que estos puedan ser aplicados.

En la Figura 2.25 se puede ver el flujo de datos del *sprint-5*, trabajando en dos tareas.

Figura 2.25

Flujo de trabajo Sprint-5.



Para desarrollar los filtros de corrección de color para daltonismo, primero se recorren todos los elementos de la página *web* para identificar y guardar los colores utilizados. Estos se convierten del modelo *HSL* (basado en tono, saturación y luminosidad, más cercano a la percepción humana) a *RGB* (basado en la mezcla de aditivos de rojo, verde y azul) para analizar su componente cromático predominante.

Según el tipo de daltonismo a corregir, se ajustan los valores *RGB* aplicando una transformación que mejora el contraste y la diferenciación de colores. Finalmente, los nuevos estilos se inyectan dinámicamente en el *CSS* de la página, permitiendo una visualización accesible sin alterar el código original.

En la Figura 2.26 se aprecia un fragmento de código de la función que aplica los filtros de colores.

Figura 2.26

Función de aplicar filtro de Deuteranopía.

```
function applyDeuteranopiaCorrection() {
  removeCustomCSS();
  restoreOriginalColors();
  const allElements = document.querySelectorAll('*');
  allElements.forEach(el => {
    // Corrige color de fondo
    const bg = getComputedStyle(el).backgroundColor;
    if (bg && bg !== 'rgba(0, 0, 0, 0)' && bg !== 'transparent') {
      if (!el.hasAttribute('data-original-bg')) {
        el.setAttribute('data-original-bg', el.style.backgroundColor || '');
      }
      el.style.backgroundColor = getDeuteranopiaSafeColor(bg);
    }
    // Corrige color de texto
    const color = getComputedStyle(el).color;
    if (color) {
      if (!el.hasAttribute('data-original-color')) {
        el.setAttribute('data-original-color', el.style.color || '');
      }
      el.style.color = getDeuteranopiaSafeColor(color);
    }
    // Corrige color de borde
    const borderColor = getComputedStyle(el).borderColor;
    if (borderColor && borderColor !== 'rgba(0, 0, 0, 0)' && borderColor !== 'transparent') {
      if (!el.hasAttribute('data-original-border')) {
        el.setAttribute('data-original-border', el.style.borderColor || '');
      }
      el.style.borderColor = getDeuteranopiaSafeColor(borderColor);
    }
  });
}
```

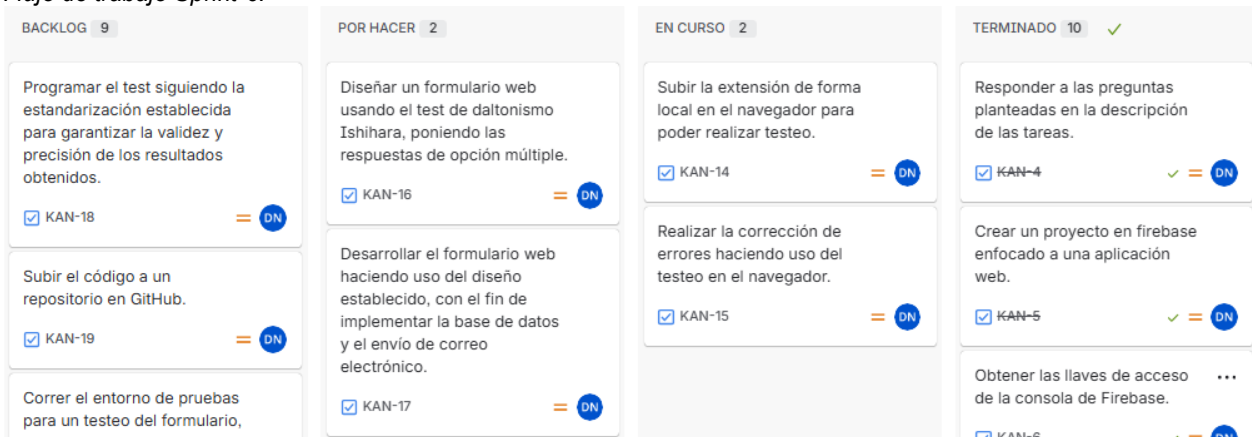
Sprint-6 Implementación de la extensión

En este *sprint* se realiza la implementación de la extensión en el navegador haciendo uso del desempaquetamiento del archivo *JSON* y subiéndolo al apartado de extensiones, para realizar testeo y corrección de errores.

En la Figura 2.27 se muestra el flujo de trabajo en el *sprint-6*, donde se trabajan las tareas de implementación.

Figura 2.27

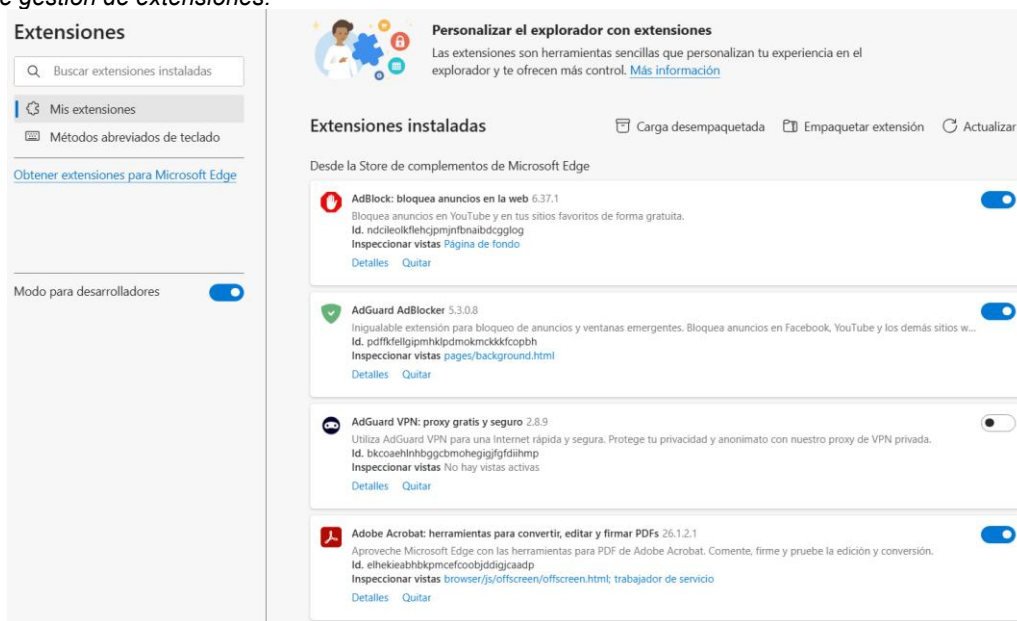
Flujo de trabajo Sprint-6.



El proceso de incorporación de una extensión al navegador mediante desempaquetamiento local consiste en preparar la carpeta del proyecto que contiene el archivo *JSON* junto con los recursos necesarios, como *scripts*, hojas de estilo e íconos. Posteriormente, se accede al panel de gestión de extensiones del navegador y se habilita el modo desarrollador o equivalente. Desde dicho panel, se selecciona la opción para cargar extensiones sin empaquetar, lo que permite seleccionar directamente la carpeta raíz del proyecto desde el sistema de archivos.

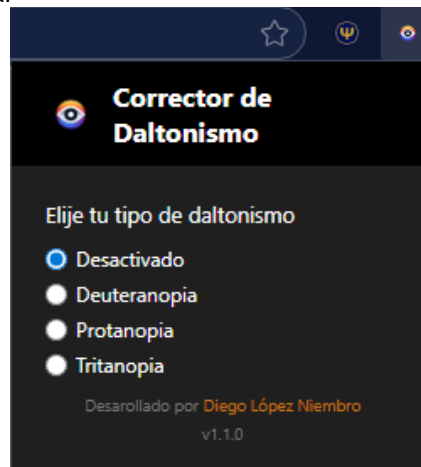
En la Figura 2.28 se muestra la pestaña de gestión de extensiones en el navegador de *Microsoft Edge*.

Figura 2.28
Pestaña de gestión de extensiones.



En la Figura 2.29 se muestra cómo se vería la extensión en un medio local ya en el navegador.

Figura 2.29
Interfaz de la extensión en desplegada.



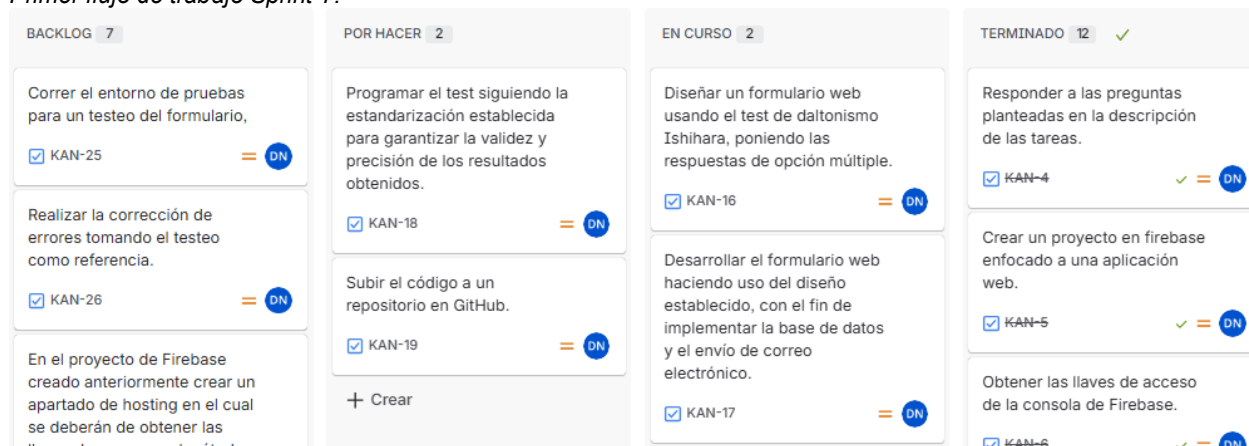
Sprint-7 Formulario

En este *sprint* se realiza el diseño del formulario *web* tomando como base el test *Ishihara* para daltonismo así mismo se desarrolla el formulario buscando la implementación de la base de datos, así como del envío de correos electrónicos.

En la Figura 2.30 se muestra el primer flujo de trabajo en el *sprint-7*, donde se trabajan las tareas de implementación.

Figura 2.30

Primer flujo de trabajo Sprint-7.

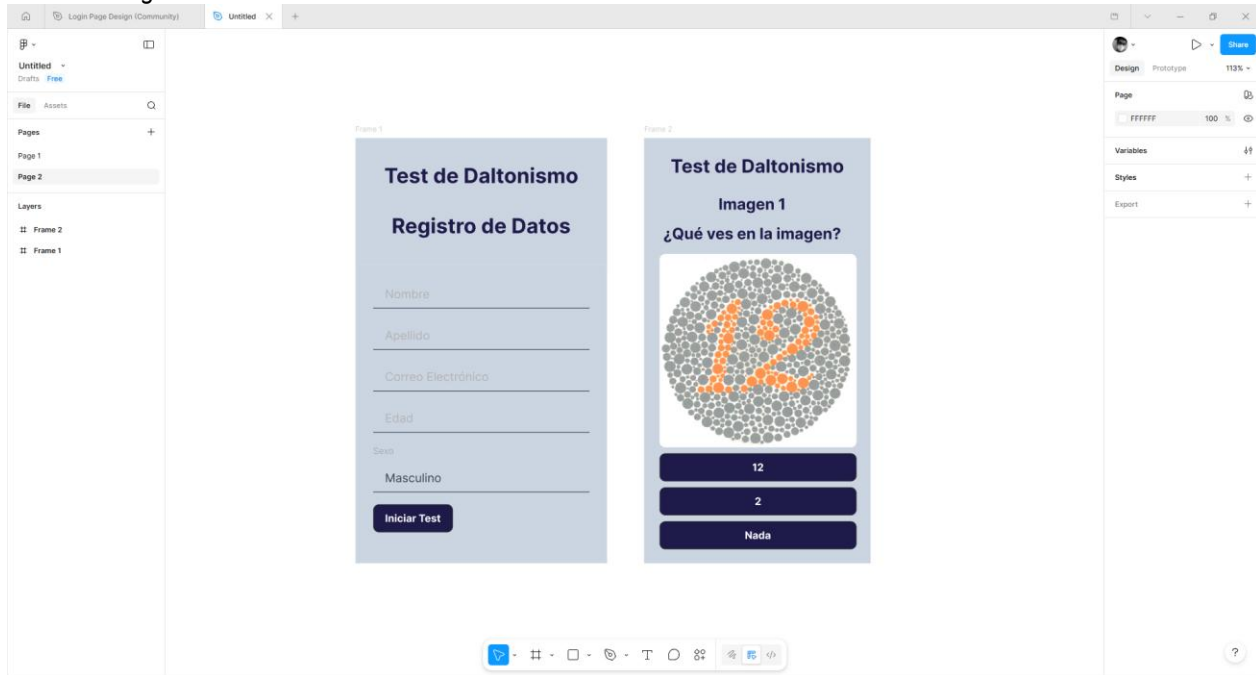


Para el diseño se utilizó como referencia un formulario sencillo con el objetivo de no desconcertar al usuario utilizando colores que, teniendo o no daltonismo, puedan ser distinguidos sin que haya pérdida de información. En el caso de las preguntas, al tratarse del test de *Ishihara* para daltonismo, cada lámina se presenta con la instrucción “¿Qué ves en la imagen?” y se ofrecen tres posibles opciones de respuesta para que el usuario seleccione la que corresponda a su percepción.

Antes de iniciar el test al usuario se le solicitan datos para tener información acerca de quién lo contesta ya que es muy importante conocer su sexo y su edad. Para el diseño del test se usaron dos *frames*: uno con la parte de formulario para recopilar la información y el otro para la parte donde se verán las imágenes del tests y las posibles respuestas. El primer *frame* está compuesto de un *header* y un cuerpo, el *header* contiene dos títulos y el cuerpo contiene cuatro *inputs*: tres de escritura, uno de menú desplegable y un botón.

El segundo *frame* está dividido en un *header* y un cuerpo, el *header* contiene dos títulos y el cuerpo contiene un título, una imagen y tres botones. En la Figura 2.31 se puede ver el diseño que se tomó para el registro de datos y el formato que toman las preguntas para hacer respondidas por el usuario.

Figura 2.31
Diseño en Figma del test.



Una vez que se cuenta con el diseño, se comienza con el desarrollo del formulario haciendo uso de *HTML* en conjunto con *CSS* para realizar el desarrollo de la interfaz, para los procesos, la vinculación con la interfaz se usa *JS* y finalmente con los servicios como el envío de correos y la base de datos se usa *JS* en conjunto con las *API's*.

En la Figura 2.32 se puede ver un fragmento de código *HTML* en la interfaz del primer formulario el cual recopila los datos del usuario.

Figura 2.32
Fragmento de código del archivo Index.

```
1 <div class="relative z-0 w-full mb-5 group">
2   <input type="text" id="nombre" name="nombre" class="block py-2.5 px-0 w-full text-
3   <label for="nombre" class="peer-focus:font-medium absolute text-sm text-gray-400
4 </div>
5
6 <div class="relative z-0 w-full mb-5 group">
7   <input type="text" id="apellido" name="apellido" class="block py-2.5 px-0 w-full
8   <label for="apellido" class="peer-focus:font-medium absolute text-sm text-gray-400
9 </div>
10
11 <div class="relative z-0 w-full mb-5 group">
12   <input type="email" id="correo" name="correo" class="block py-2.5 px-0 w-full text-
13   <label for="correo" class="peer-focus:font-medium absolute text-sm text-gray-400
14 </div>
```

Para el proceso en el cual el usuario resuelve el formulario, se hace uso de una función la cual intercala la etiqueta de "*hidden*" para poder intercalar con las tarjetas del formulario, en la Figura 2.33 se muestra un fragmento de código del archivo que se encarga del funcionamiento del formulario.

Figura 2.33

Fragmento de código del archivo *test.js*.

```
1 // Función para mostrar la siguiente pregunta
2 function mostrarPregunta(preguntaActual, preguntaSiguiente) {
3   document.getElementById(`pregunta${preguntaActual}`).classList.add('hidden');
4   if (preguntaSiguiente <= 24) {
5     document.getElementById(`pregunta${preguntaSiguiente}`).classList.remove('hidden');
6   } else {
7     // Mostrar el resultado si es la última pregunta
8     mostrarResultado();
9   }
10 }
```

Una vez que se comprueba que el formulario avanza de forma correcta, el siguiente paso es guardar las respuestas y estructurar un camino de preguntas para que el test pueda arrojar un resultado.

El siguiente paso es la vinculación de la base de datos y el servicio de correo electrónico por medio de las *API's*, para lograr esta conexión se necesita hacer uso de las llaves recolectadas, en el caso de la base de datos se usará *Firestore* para poder vincularla. Para el envío del correo electrónico se hará uso de *EmailJS*.

En el caso de *Firestore* se utiliza las llaves mediante un formato específico en la Figura 2.34 se muestra un fragmento de código del archivo de vínculo entre el proyecto y *Firebase*.

Figura 2.34

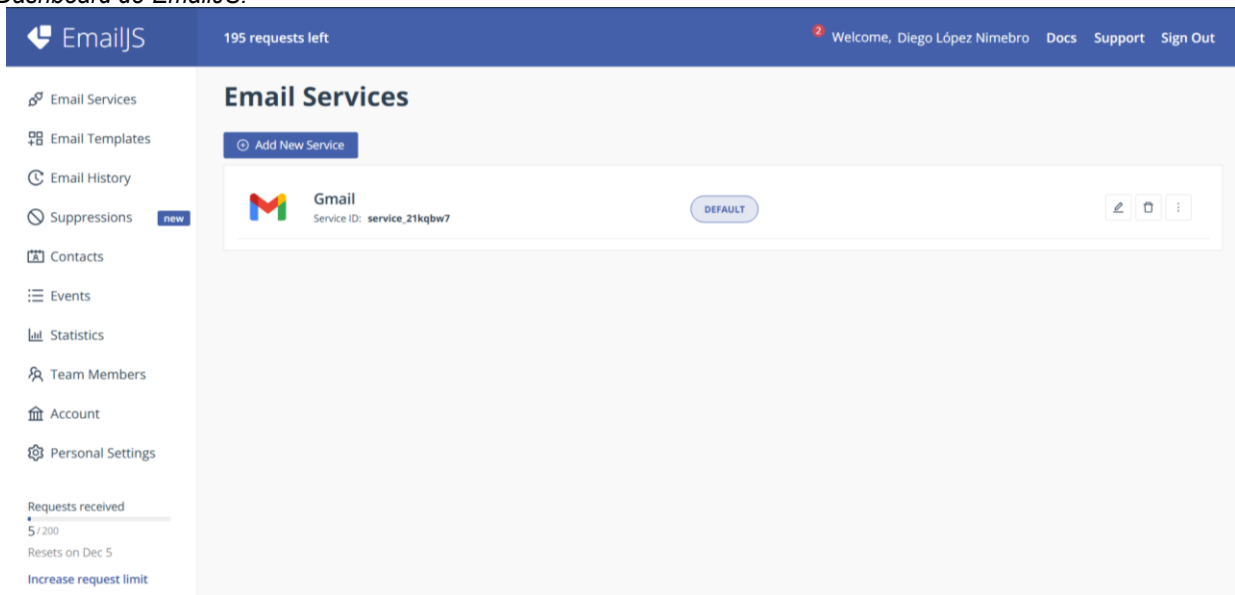
Fragmento de código del archivo *configuración.js*.

```
1 firebase.initializeApp(firebaseConfig);
2
3 const formularioDB = firebase.database().ref('formulario');
4
5 document.getElementById('formulario').addEventListener('submit', submitForm);
6
7 function submitForm(e){
8   e.preventDefault();
9
10   var nombre = getElementVal('nombre');
11   var apellido = getElementVal('apellido');
12   var correo = getElementVal('correo');
13   var edad = getElementVal('edad');
14   var sexo = getElementVal('sexo');
15
16   saveMessages(nombre, apellido, correo, edad, sexo);
17 }
```

En el caso de la implementación de *EmailJS* se hace un proceso similar al de *Firebase*. En este caso, el primer paso es crear la cuenta de usuario de *EmailJS*, la cual te permitirá tener acceso al *dashboard*. Desde allí se puede gestionar los servicios de correo, las plantillas de mensaje y las credenciales necesarias para conectar el proyecto con la plataforma.

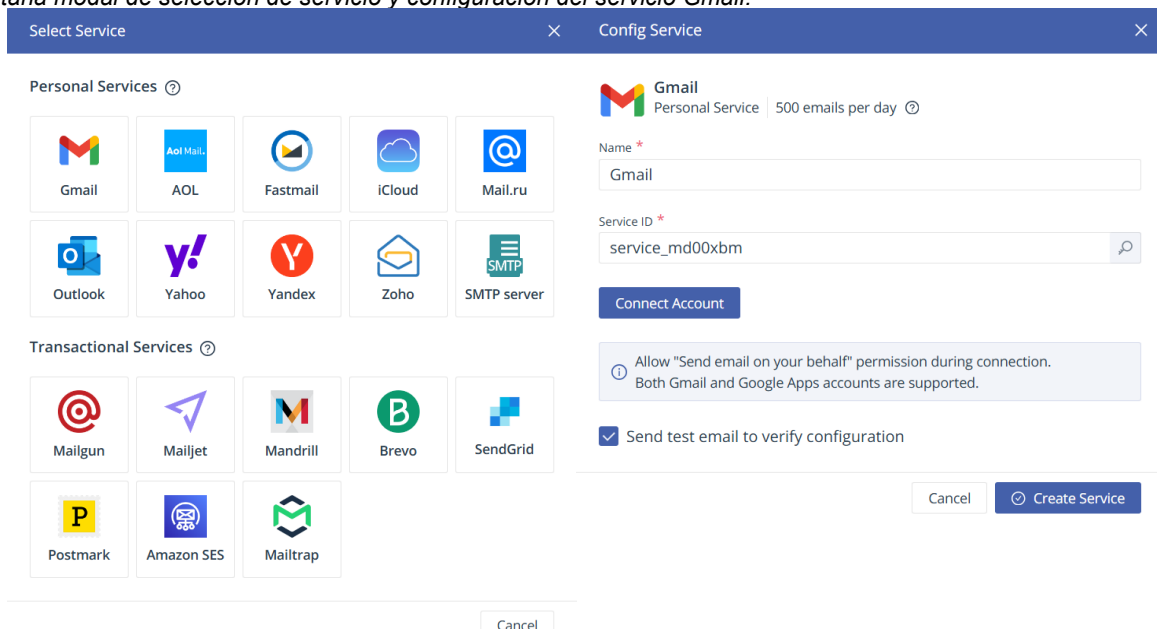
Una vez dentro del *dashboard*, lo primero que se debe hacer es crear un servicio de correo. Esto implica vincular la cuenta de correo electrónico (como *Gmail*, *Outlook* o cualquier otro proveedor compatible) con *EmailJS*. Este servicio será el canal por el cual se enviarán los correos desde la aplicación. En la Figura 2.35 Se muestra el *dashbord* de *EmailJS*.

Figura 2.35
Dashboard de EmailJS.



En la Figura 2.36 se muestra la interfaz de selección de servicio de correos así de como un ejemplo de cómo funciona la incorporación del servicio de *Gmail*.

Figura 2.36
Ventana modal de selección de servicio y configuración del servicio Gmail.



Después de configurar el servicio de correo, el siguiente paso es crear una plantilla de correo. En esta plantilla se define el contenido del mensaje que se enviará, incluyendo el asunto, el cuerpo del correo y las variables dinámicas que se reemplazarán con los datos del usuario, como nombre, correo o mensaje. En la Figura 2.37 Se muestra la interfaz de creación de las plantillas.

Figura 2.37

Pestaña de configuración de las platillas.

Con el servicio y la plantilla listos, se debe integrar *EmailJS*. En la Figura 2.3 se muestra un fragmento de código de la implementación de *EmailJS* con el proyecto.

Figura 2.38

Fragmento de código del archivo configuración.js.

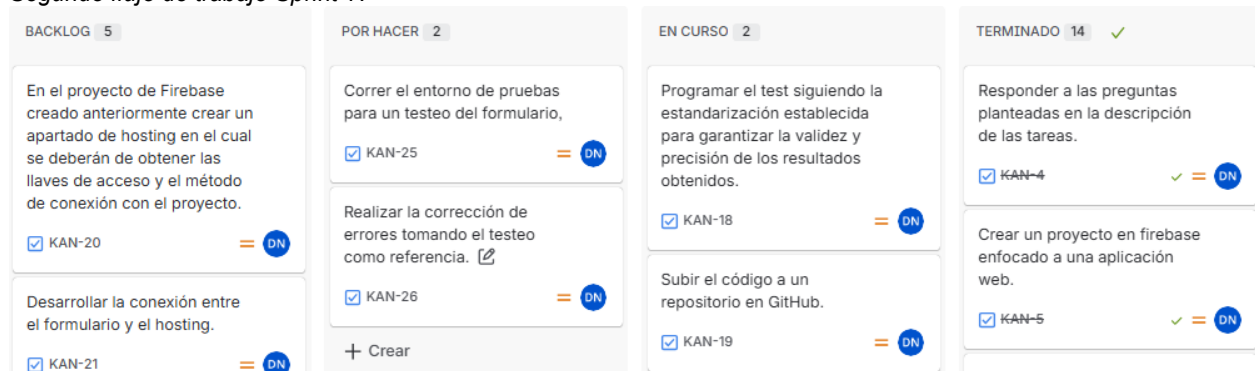
```

1  // Función para enviar el correo
2  function enviarCorreo() {
3      let parametros = {
4          nombre: document.getElementById("nombre").value,
5          apellido: document.getElementById("apellido").value,
6          correo: document.getElementById("correo").value,
7          edad: document.getElementById("edad").value,
8          sexo: document.getElementById("sexo").value,
9          resultadoTest: document.getElementById('resultadoTexto').innerText
10     };
11
12
13     // Enviar el correo electrónico
14     emailjs.send('service_21kqb7', 'template_7xp6n8', {
15         ...parametros,
16         // Agregar aquí Los parámetros que se usarán en el template
17     }).then(function(response) {
18         console.log('Correo enviado exitosamente', response.status, response.text);
19     }, function(error) {
20         console.log('Error al enviar el correo', error);
21     });
22 }

```

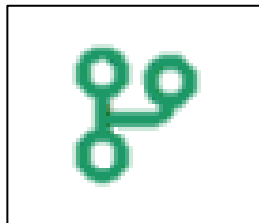
Una vez realizado el desarrollo se realiza un respaldo del proyecto dentro de un repositorio dentro de *GitHub* para evitar la pérdida de las versiones en la fase de corrección de errores. En la Figura 2.39 se muestra el segundo flujo del trabajo el cual involucra: que el resultado sea el correcto, las respuestas y la subida a *GitHub*.

Figura 2.39
Segundo flujo de trabajo Sprint-7.



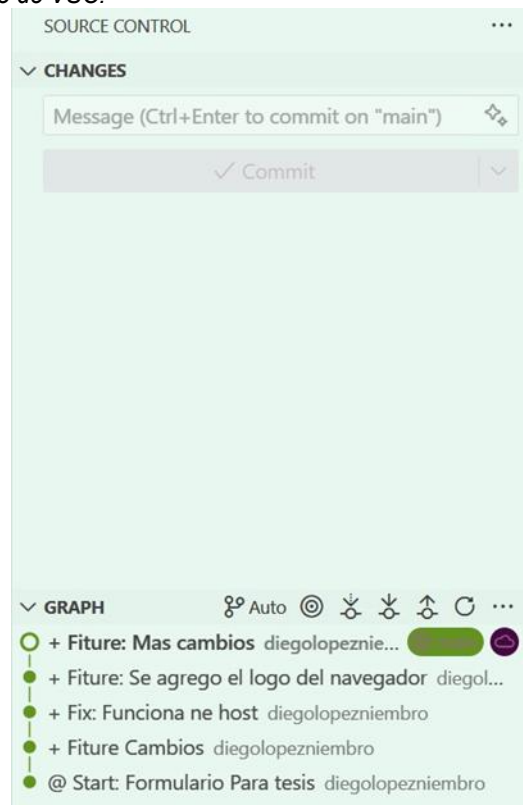
Para realizar la vinculación de un proyecto local con un repositorio remoto utilizando la interfaz gráfica de *Visual Studio Code* (VSC, por sus siglas en inglés), se debe iniciar abriendo la carpeta del proyecto dentro del entorno de desarrollo. Al seleccionar la opción “*Initialize Repository*”, se crea un repositorio local que permite comenzar a gestionar el historial de versiones del proyecto. En la Figura 2.40 se muestra el icono de manejo de versiones dentro de la barra lateral de VSC.

Figura 2.40
Icono de manejo de versiones de git en VSC.



Una vez inicializado el repositorio, los archivos modificados aparecerán listados en la interfaz de control de versiones. Para registrar un primer cambio, se debe redactar un mensaje descriptivo en el campo destinado a los *commits* y confirmar la acción mediante el ícono de verificación. La Figura 2.41 muestra la pestaña de *Source Control* donde se muestra el mensaje y las modificaciones propuestas y los gráficos que son las distintas versiones del sistema.

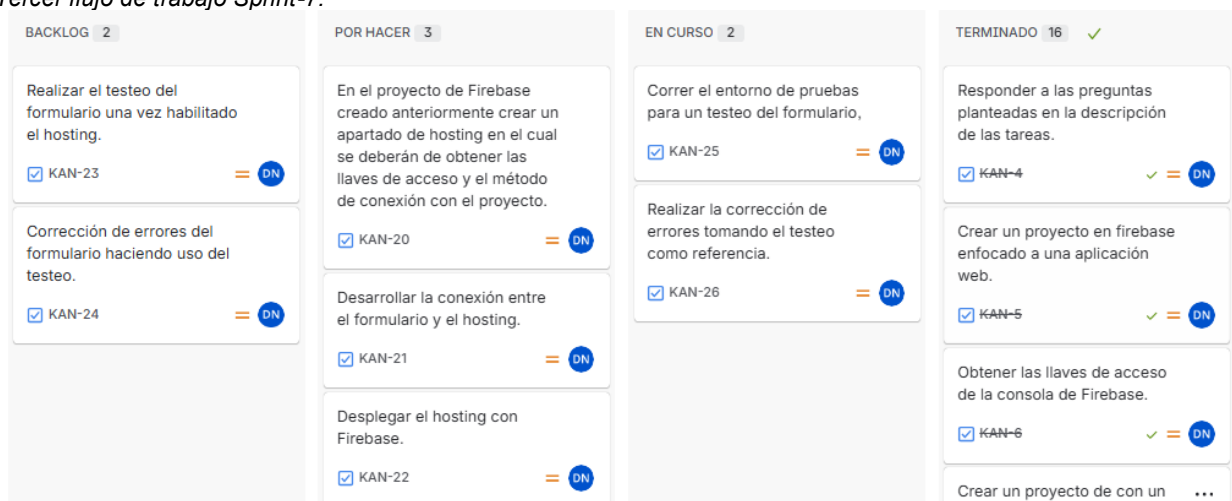
Figura 2.41
Pestaña de Source Control dentro de VSC.



Ya que se ha subido el proyecto *git* se procede a la implementación del formulario en un entorno de pruebas dentro del navegador para su funcionamiento y se realiza la corrección de errores correspondiente.

En la Figura 2.42 se muestra el flujo de trabajo del *sprint* donde se encuentran dos tareas la de testeo en navegador por medio de un entorno de pruebas y la de corrección de errores.

Figura 2.42
Tercer flujo de trabajo Sprint-7.



En la Figura 2.43 se muestra el comando a introducir en la línea de consola para correr el entorno de pruebas.

Figura 2.43

Línea de consola con el comando de ejecución de del entorno de pruebas.

```
PS L:\Tesis\Formulario> npm run dev
```

Una vez que se establece el entorno de pruebas se realiza el testeo en conjunto con la corrección de errores.

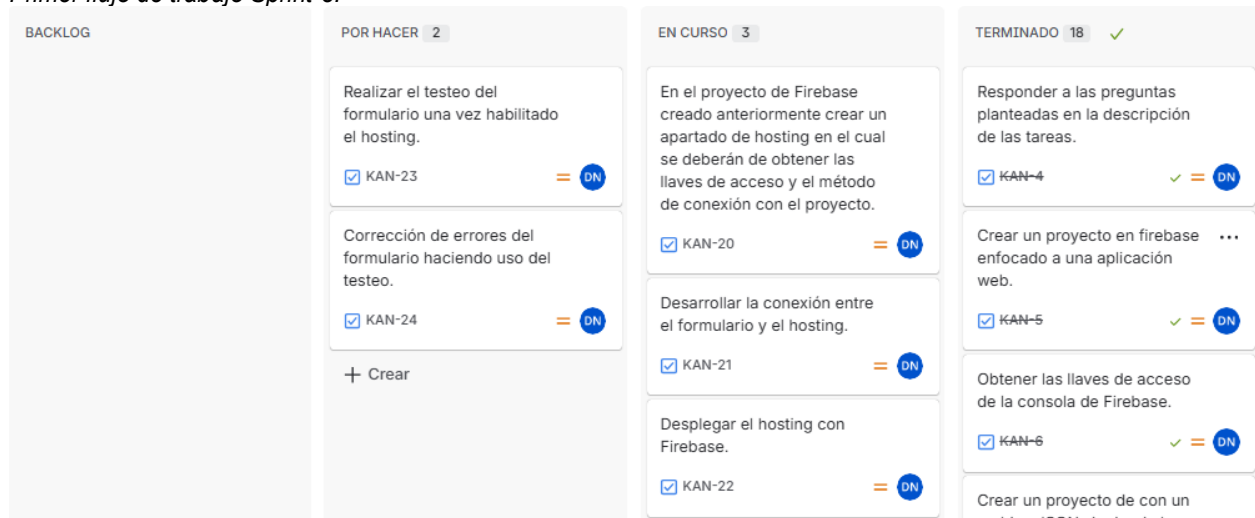
Sprint-8 Hosting

En este *sprint* se realiza la sincronización con el *hosting* de *Firebase* obteniendo las llaves de acceso y el método de conexión con el proyecto, así como su testeo y corrección de errores.

En la Figura 2.44 se muestra el primer flujo de trabajo en el *sprint-8*, donde se trabajan las tareas de obtención de llaves, el desarrollo de la conexión y el despliegue.

Figura 2.44

Primer flujo de trabajo Sprint-8.

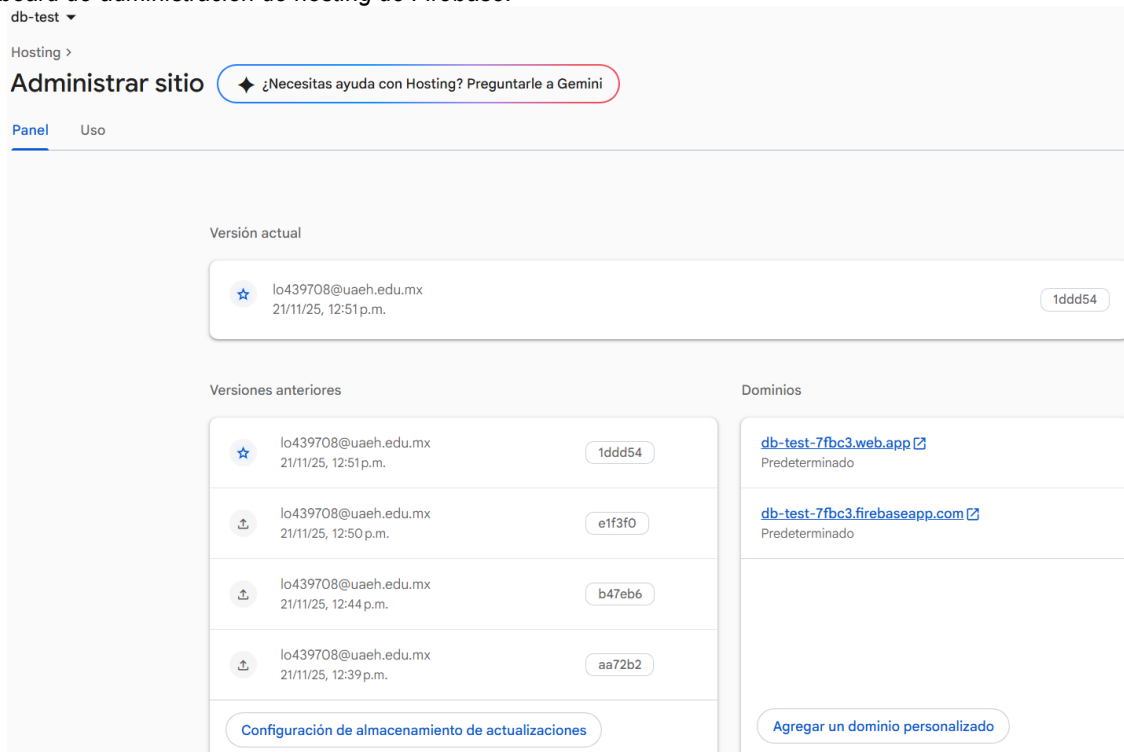


La publicación del proyecto en *Firebase hosting* requiere, como primer paso, la configuración inicial del servicio desde la consola *web* de *Firebase*. Para ello, en el proyecto ya creado con anterioridad se habilita el *hosting*. Siguiendo el asistente de configuración. Este proceso permite registrar el dominio de despliegue y establecer las condiciones básicas de alojamiento, incluyendo la carpeta raíz del proyecto y las reglas de acceso.

La Figura 2.45 muestra el *dashboard* de administración del *hosting*, así como el dominio establecido para el proyecto.

Figura 2.45

Dashboard de administración de hosting de Firebase.



Con el servicio de *hosting* ya habilitado en la consola, se procede a vincular el entorno local mediante la autenticación del usuario. Para ello, se utiliza el comando “*firebase login*”, el cual permite establecer una sesión segura entre la terminal del sistema y la cuenta de *Firebase* previamente configurada.

A continuación, se ejecuta el comando “*firebase init*”, que permite configurar los servicios específicos que se utilizarán en el proyecto. En este caso, se selecciona únicamente la opción “*Hosting*”, y se especifica la carpeta de distribución que contiene los archivos compilados del proyecto el cual será la carpeta de “*dist*”.

En la Figura 2.46 se muestra un ejemplo de cómo es ejecutar el comando de “*firebase init*”.

Figura 2.46

Ejemplo de consola de ejecución de firebase init.



Una vez completada la configuración, *Firebase* genera los archivos *firebase.json* y *.firebaserc*, que contienen las directrices de despliegue y la referencia al proyecto remoto. Con esta estructura establecida, se compila el proyecto local mediante el comando “*npm run build*”, y se ejecuta el comando “*firebase deploy*” para publicar los archivos en el servidor. Al finalizar el proceso, *Firebase* proporciona una URL pública desde la cual se puede acceder a la aplicación desplegada.

En la Figura 2.47 se muestra un fragmento de código del archivo generado de *firebase.json*.

Figura 2.47

Fragmento de código del archivo firebase.json.

```
1  {
2    "hosting": {
3      "public": "dist",
4      "ignore": [
5        "firebase.json",
6        "**/.*",
7        "**/node_modules/**"
8      ],
9      "rewrites": [
10     {
11       "source": "**",
12       "destination": "/index.html"
13     }
14   ]
15 }
16 }
```

En la Figura 2.48 se muestra un ejemplo de consola de la ejecución del comando “*firebase deploy*”.

Figura 2.48

Fragmento de consola de ejecución del comando firebase deploy.

```
PS L:\Tesis\Formulario> firebase deploy
(node:35932) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please
  use `node --trace-deprecation ...` to show where the warning was created)

=== Deploying to 'db-test-7fbc3'...

i  deploying hosting
i  hosting[db-test-7fbc3]: beginning deploy...
i  hosting[db-test-7fbc3]: found 33 files in dist
+  hosting[db-test-7fbc3]: file upload complete
i  hosting[db-test-7fbc3]: finalizing version...
+  hosting[db-test-7fbc3]: version finalized
i  hosting[db-test-7fbc3]: releasing new version...
+  hosting[db-test-7fbc3]: release complete

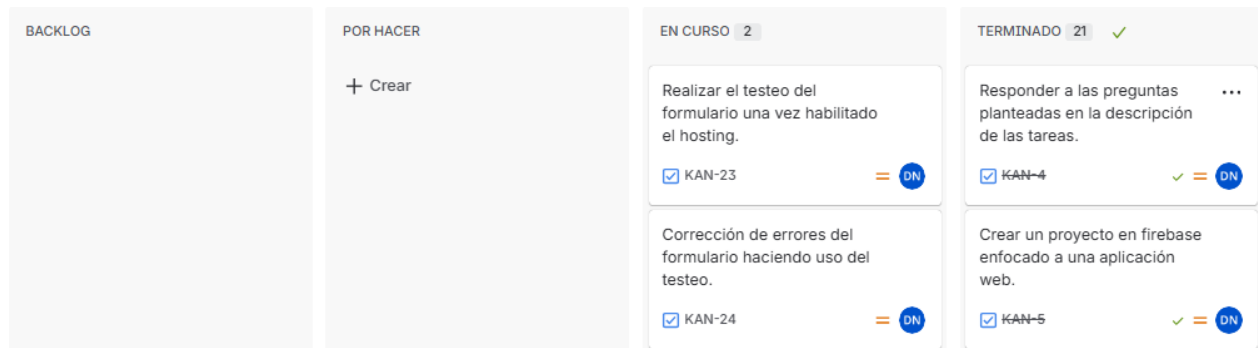
+  Deploy complete!

Project Console: https://console.firebase.google.com/project/db-test-7fbc3/overview
Hosting URL: https://db-test-7fbc3.web.app
```

Una vez que el proyecto ya se encuentra desplegado en la *web* se procede a hacer el testeo y la corrección de errores correspondiente pensando que es un entorno de producción ya no local.

En la Figura 2.49 se muestra el segundo flujo de trabajo del *sprint-8* En el cual se encuentran la tarea de testeo y la corrección de errores.

Figura 2.49
Segundo flujo de trabajo Sprint-8.



Los aspectos relevantes que considerar durante el proceso de testeo y corrección de errores una vez desplegado el proyecto en la *web* son: los componentes reaccionan igual debido a que su comportamiento es diferente en un navegador que en un entorno local y el acomodo de los directorios del proyecto ya que el renombramiento puede modificarse.

2.4 Mejora continua

En este apartado se explicarán los procesos de optimización en el desarrollo, así como la eficiencia de estos. El proceso de mejora se llevó a cabo por medio de los siguientes puntos.

- Revisión del flujo de trabajo: el tablero *Kanban* implementado tiene como objetivo de evitar cuellos de botella, sobrecarga de trabajo y proponer límites en el proceso de trabajo.
- Retroalimentación del proyecto: después de cada finalización de ciclo se realiza un análisis del proyecto, con la finalidad de detección de problemas, escritura de comentarios, mejoras en la lectura del código y mejoras en la velocidad del proceso.
- Actualización continua del tablero: conforme al avance del proyecto, las mejoras y problemas a solucionar fueron tratados mediante modificaciones a las tareas, en el caso de mejoras y adición de tareas nuevas en el caso de problemas, con el propósito de adaptar el proyecto a las necesidades.

- Implementación de buenas prácticas y documentación: al final de proyecto se realiza un escaneo de código buscando malas prácticas y que exista una documentación correcta del proyecto tanto en las llaves de acceso, como en las funciones.

Capítulo 3: Resultados del test de Ishihara

En este capítulo se presentan los resultados obtenidos al aplicar el test de *Ishihara* a los participantes de la investigación. El objetivo es identificar la presencia de daltonismo dentro de la muestra y comprobar la hipótesis planteada. Los datos se muestran mediante gráficas y se acompañan de una interpretación que permite entender cómo las respuestas reflejan las dificultades en la percepción cromática.

3.1 Fundamentos del cuestionario

El test de *Ishihara* fue desarrollado por *Shinobu Ishihara* en 1917 y se ha consolidado como una de las pruebas más confiables para la detección de daltonismo. Consiste en una serie de láminas compuestas por puntos de colores que forman números o figuras.

- *Principio de funcionamiento*: las personas con visión cromática normal identifican los números sin dificultad, mientras que quienes presentan deficiencias en la percepción rojo-verde tienden a confundirlos o no reconocerlos.
- *Aplicación en este estudio*: cada participante observó las láminas en condiciones controladas y registró la respuesta percibida.
- *Relevancia*: los resultados permiten clasificar a los participantes en dos grupos: aquellos con visión cromática normal y aquellos con deficiencia, lo que constituye la base para validar la pertinencia de la extensión desarrollada.

3.2 Resultados del test de Ishihara

A continuación, se muestran las gráficas del test. Cabe mencionar que la totalidad de las láminas se encuentran en el Anexo II. Los incisos en conjunto con la lámina están en el Apéndice II; para dar sustento a los datos, el instrumento fue aplicado a una muestra de ochenta hombres.

Lámina 1. De los participantes el 96% seleccionó la opción 1, en tanto que el 4% seleccionó la opción 3, lo que significa que la mayoría de los participantes identificaron correctamente el número “12”. Lo anterior se muestra de forma gráfica en la Figura 3.1.

Figura 3.1
Gráfica de resultados de la lámina 1 del test de Ishihara.
Pregunta 1

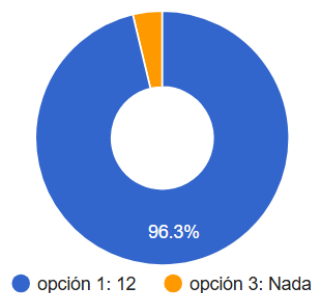


Lámina 2. De los participantes, el 82% seleccionó la opción 1, en tanto que el 10% seleccionó la opción 2 y el 8% seleccionó la opción 3, lo que significa que gran parte de los participantes identificaron correctamente el número “8” en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.2.

Figura 3.2

Gráfica de resultados de la lámina 2 del test de Ishihara.

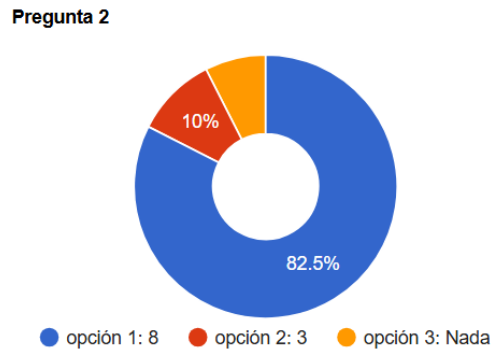


Lámina 3. De los participantes el 5% seleccionó la opción 1, el 89% seleccionó la opción 2 y el 6% eligió la opción 3, lo que significa que la mayoría de los participantes identificaron correctamente el número “29”. Lo anterior se muestra de forma gráfica en la Figura 3.3.

Figura 3.3

Gráfica de resultados de la lámina 3 del test de Ishihara.

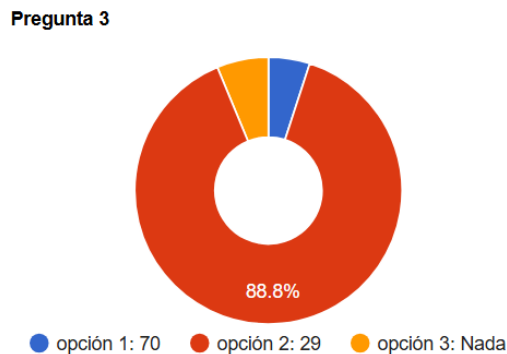


Lámina 4. De los participantes el 5% seleccionó la opción 1, el 86% seleccionó la opción 2 y el 9% eligió la opción 3, lo que significa que la gran parte de los participantes identificaron correctamente el número “5”. Lo anterior se muestra de forma gráfica en la Figura 3.4.

Figura 3.4

Gráfica de resultados de la lámina 4 del test de Ishihara.

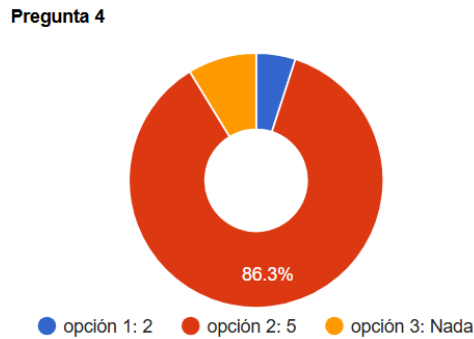


Lámina 5. De los participantes el 5% seleccionó la opción 1, el 86% seleccionó la opción 2 y el 9% eligió la opción 3, lo que significa que la mayor parte de los participantes identificaron correctamente el número “3”. Lo anterior se muestra de forma gráfica en la Figura 3.5.

Figura 3.5

Gráfica de resultados de la lámina 5 del test de Ishihara.

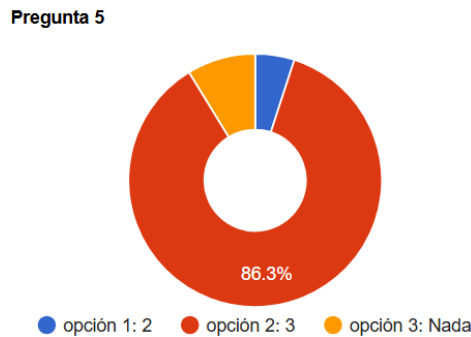


Lámina 6. De los participantes el 85% seleccionó la opción 1, el 6% seleccionó la opción 2 y el 9% eligió la opción 3, lo que representa que una proporción significativa de los participantes identificaron correctamente el número “15”. Lo anterior se muestra de forma gráfica en la Figura 3.6.

Figura 3.6

Gráfica de resultados de la lámina 6 del test de Ishihara.

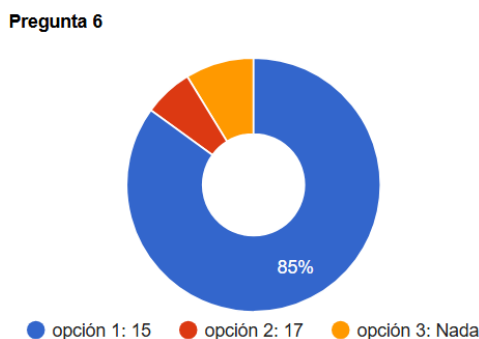


Lámina 7. De los participantes el 82% seleccionó la opción 1, el 9% seleccionó la opción 2 y el 9% eligió la opción 3, lo que significa que la amplia mayoría de los participantes identificaron correctamente el número “74”. Lo anterior se muestra de forma gráfica en la Figura 3.7.

Figura 3.7

Gráfica de resultados de la lámina 7 del test de Ishihara.

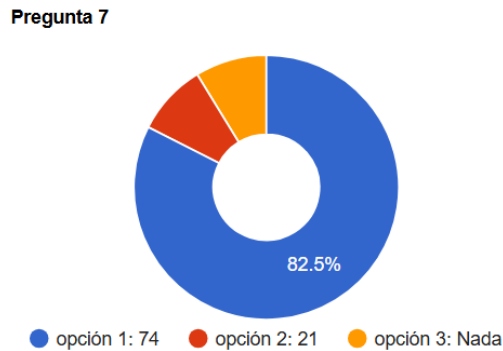


Lámina 8. De los participantes el 84% seleccionó la opción 1, el 2% seleccionó la opción 2 y el 14% eligió la opción 3, lo que significa que la amplia proporción de los participantes identificaron correctamente el número “6”. Lo anterior se muestra de forma gráfica en la Figura 3.8.

Figura 3.8

Gráfica de resultados de la lámina 8 del test de Ishihara.

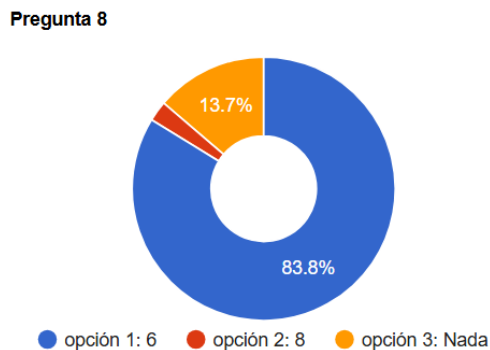


Lámina 9. De los participantes el 87% seleccionó la opción 2 y el 13% eligió la opción 3, lo que significa que la mayor parte de los participantes identificaron correctamente el número “45”. Lo anterior se muestra de forma gráfica en la Figura 3.9.

Figura 3.9

Gráfica de resultados de la lámina 9 del test de Ishihara.

Pregunta 9

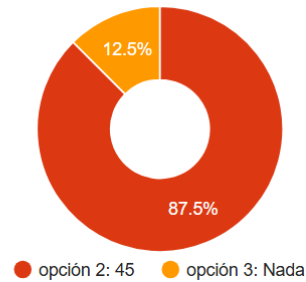


Lámina 10. De los participantes el 84% seleccionó la opción 1, el 5% seleccionó la opción 2 y el 11% eligió la opción 3, lo que significa que una amplia mayoría de los participantes identificaron correctamente el número “5”. Lo anterior se muestra de forma gráfica en la Figura 3.10.

Figura 3.10

Gráfica de resultados de la lámina 10 del test de Ishihara.

Pregunta 10

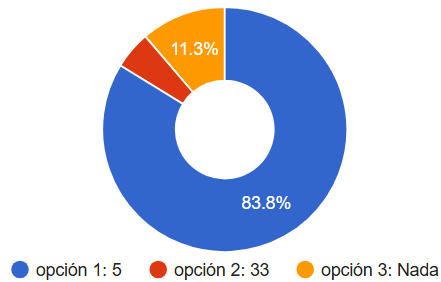


Lámina 11. De los participantes el 4% seleccionó la opción 1, el 85% seleccionó la opción 2 y el 11% eligió la opción 3, lo que significa que una gran parte de los participantes identificaron correctamente el número “7”. Lo anterior se muestra de forma gráfica en la Figura 3.11.

Figura 3.11

Gráfica de resultados de la lámina 11 del test de Ishihara.

Pregunta 11

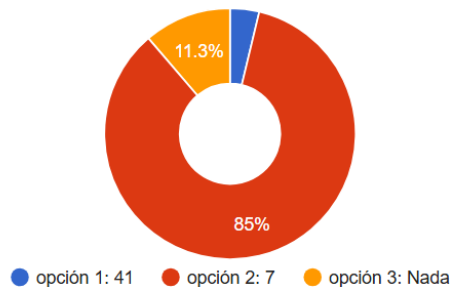


Lámina 12. De los participantes el 1% seleccionó la opción 1, el 88% seleccionó la opción 2 y el 11% eligió la opción 3, lo que significa que la amplia proporción de los participantes identificaron correctamente el número “16”. Lo anterior se muestra de forma gráfica en la Figura 3.12.

Figura 3.12

Gráfica de resultados de la lámina 12 del test de Ishihara.

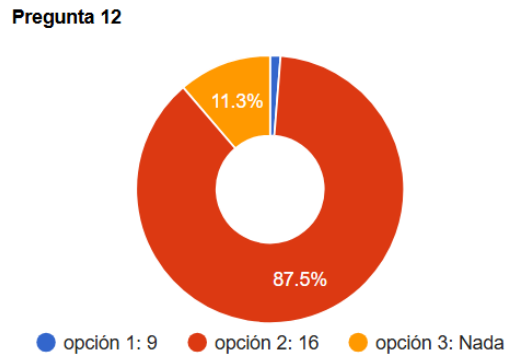


Lámina 13. De los participantes el 90% seleccionó la opción 1 y el 10% eligió la opción 3, lo que significa que la mayoría de los participantes identificaron correctamente el número “73”. Lo anterior se muestra de forma gráfica en la Figura 3.13.

Figura 3.13

Gráfica de resultados de la lámina 13 del test de Ishihara.

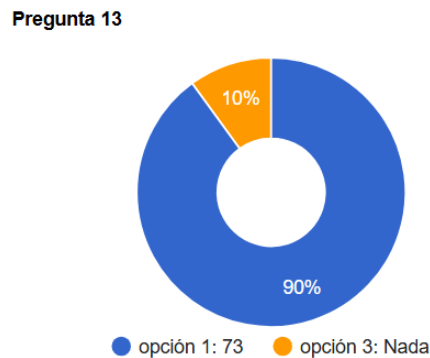


Lámina 14. De los participantes el 89% seleccionó la opción 1, el 6% seleccionó la opción 2 y el 5% eligió la opción 3, lo que significa que una proporción considerable de los participantes identificaron correctamente la opción “nada”. Lo anterior se muestra de forma gráfica en la Figura 3.14.

Figura 3.14

Gráfica de resultados de la lámina 14 del test de Ishihara.

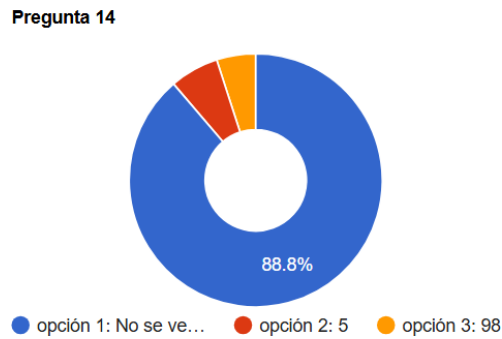


Lámina 15. De los participantes el 5% seleccionó la opción 1, el 92% seleccionó la opción 2 y el 3% eligió la opción 3, lo que significa que la gran mayoría de los participantes identificaron correctamente la opción “nada”. Lo anterior se muestra de forma gráfica en la Figura 3.15, la opción uno es “No se ve ningún número”.

Figura 3.15

Gráfica de resultados de la lámina 15 del test de Ishihara.

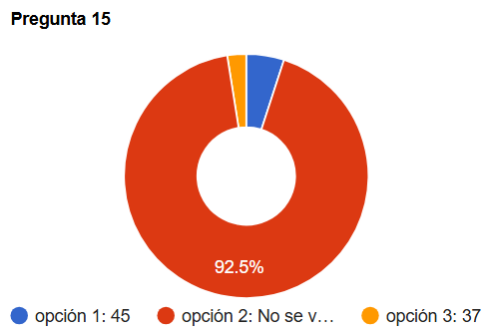


Lámina 16. De los participantes el 2% seleccionó la opción 1, el 8% seleccionó la opción 2 y el 90% eligió la opción 3, lo que significa que la mayor parte de los participantes identificaron correctamente el número “26”. Lo anterior se muestra de forma gráfica en la Figura 3.16, la opción dos es “No se ve ningún número”.

Figura 3.16

Gráfica de resultados de la lámina 16 del test de Ishihara.

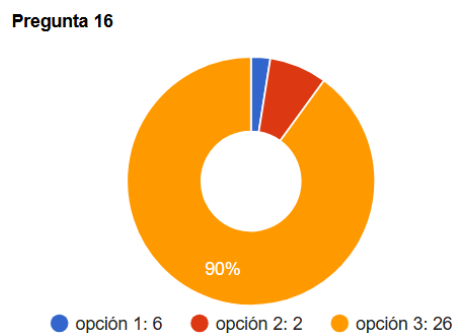


Lámina 17. De los participantes el 2% seleccionó la opción 1, el 94% seleccionó la opción 2 y el 4% eligió la opción 3, lo que significa que una amplia proporción de los participantes identificaron correctamente el número “42”. Lo anterior se muestra de forma gráfica en la Figura 3.17.

Figura 3.17

Gráfica de resultados de la lámina 17 del test de Ishihara.

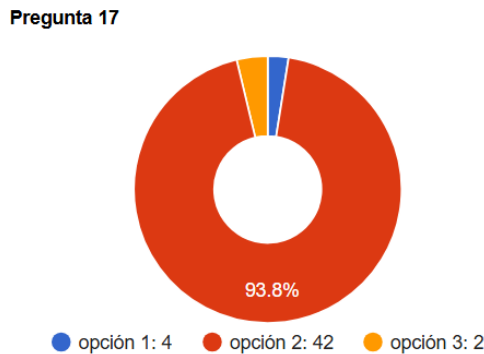


Lámina 18. De los participantes el 8% seleccionó la opción 1, el 5% seleccionó la opción 2 y el 87% eligió la opción 3, lo que significa que una gran mayoría de los participantes identificaron correctamente los dos caminos en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.18, donde la primera opción corresponde a “1 camino arriba”, la segunda a “1 camino abajo” y la tercera a “2 caminos”.

Figura 3.18

Gráfica de resultados de la lámina 18 del test de Ishihara.

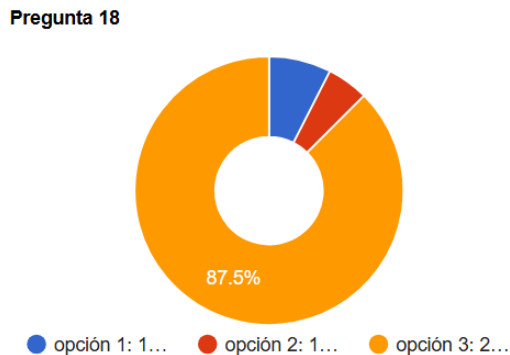


Lámina 19. De los participantes el 23% seleccionó la opción 1, el 41% seleccionó la opción 2 y el 36% eligió la opción 3, lo que representa que una parte significativa de los participantes identificaron un escorpión en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.19, donde la primera opción corresponde a “Un círculo del mismo color”, la segunda a “Un escorpión” y la tercera a “Una línea”.

Figura 3.19

Gráfica de resultados de la lámina 20 del test de Ishihara.

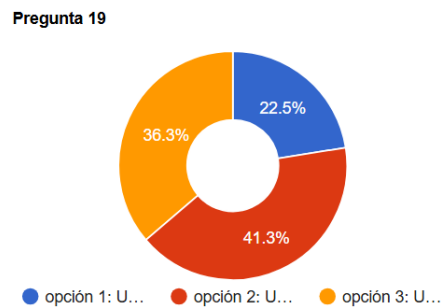


Lámina 20. De los participantes el 65% seleccionó la opción 1, el 9% seleccionó la opción 2 y el 26% eligió la opción 3, lo que significa que una parte considerable de los participantes identificaron correctamente un camino en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.20, donde la primera opción corresponde a “1 camino”, la segunda a “1 camino sin terminar” y la tercera a “Nada”.

Figura 3.20

Gráfica de resultados de la lámina 20 del test de Ishihara.

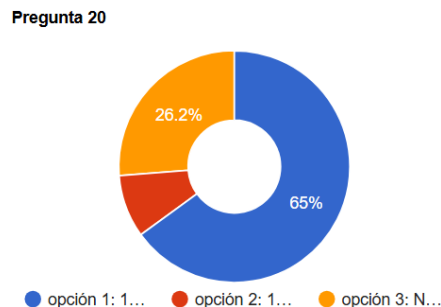


Lámina 21. De los participantes el 66% seleccionó la opción 1, el 11% seleccionó la opción 2 y el 23% eligió la opción 3, lo que significa que una mayoría de los participantes identificaron correctamente un camino en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.21, donde la primera opción corresponde a “1 camino”, la segunda a “1 camino sin terminar” y la tercera a “Nada”.

Figura 3.21

Gráfica de resultados de la lámina 21 del test de Ishihara.

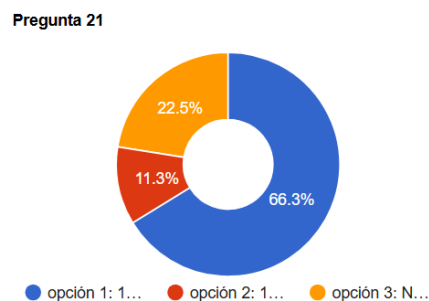


Lámina 22. De los participantes el 69% seleccionó la opción 1, el 12% seleccionó la opción 2 y el 19% eligió la opción 3, lo que representa que una proporción significativa de los participantes identificaron correctamente un camino de dos colores verde-azul y amarillo-verde en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.22, donde la primera opción corresponde a “1 camino de 2 colores verde-azul y amarillo-verde”, la segunda a “1 camino de 2 colores rojo y naranja” y la tercera a “Nada”.

Figura 3.22

Gráfica de resultados de la lámina 22 del test de Ishihara.

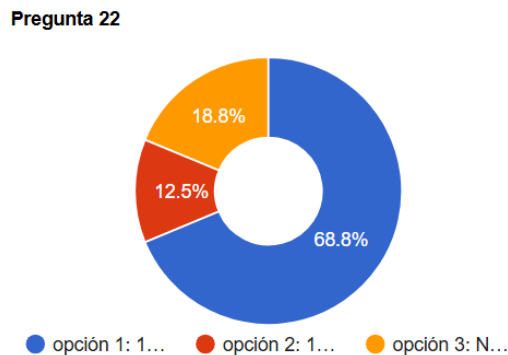


Lámina 23. De los participantes el 85% seleccionó la opción 1, el 8% seleccionó la opción 2 y el 6% eligió la opción 3, lo que significa que una amplia mayoría de los participantes identificaron correctamente un camino de dos colores rojo y naranja en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.23, donde la primera opción corresponde a “1 camino de 2 colores rojo y naranja”, la segunda a “1 camino de 2 colores rojo y verde-azul” y la tercera a “Nada”.

Figura 3.23

Gráfica de resultados de la lámina 23 del test de Ishihara.

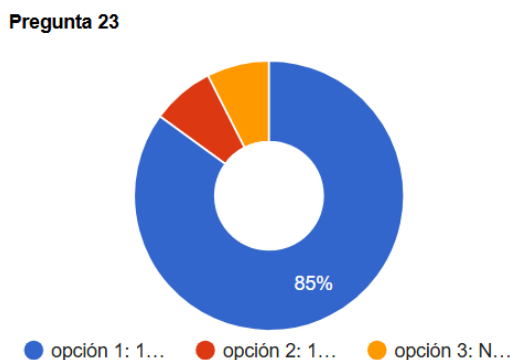
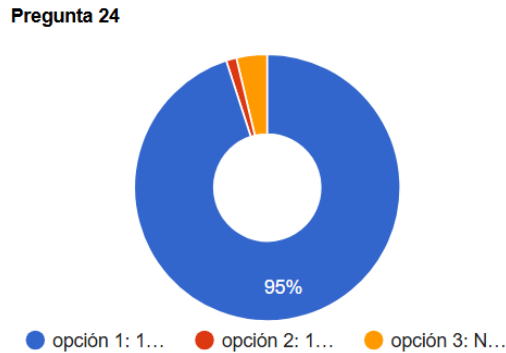


Lámina 24. De los participantes el 95% seleccionó la opción 1, el 1% seleccionó la opción 2 y el 4% eligió la opción 3, lo que significa que la gran mayoría de los participantes identificaron correctamente un camino en la lámina. Lo anterior se muestra de forma gráfica en la Figura 3.24, donde la primera opción corresponde a “1 camino”, la segunda a “1 sonrisa” y la tercera a “Nada”.

Figura 3.24

Gráfica de resultados de la lámina 24 del test de Ishihara.



De acuerdo con los resultados mostrados en las gráficas anteriores, se comprueba que la estadística presentada por organismos especializados, tales como la OMS (Organización Mundial de la Salud), es correcta, ya que los valores obtenidos coinciden con las cifras internacionales que señalan que aproximadamente el 8% de los hombres presentan algún grado de daltonismo, según la *American Academy of Ophthalmology* y el *National Eye Institute*. Esta concordancia entre los datos locales y las estimaciones globales respalda la validez del test de Ishihara como herramienta diagnóstica y confirma que los hallazgos forman parte de un patrón mundial ampliamente documentado.

Capítulo 4: Validación

En este capítulo se abarca la validación de la extensión *DaltonAssist* por medio del cuestionario de experiencia de usuario o *UEQ* (User Experience Questionnaire, por sus siglas en inglés), una metodología estandarizada que permite medir percepciones subjetivas en torno a productos interactivos.

4.1 Objetivo de la validación

El objetivo principal de esta validación es determinar en qué medida la extensión propuesta cumple con criterios de calidad, tales como facilidad de uso, eficiencia, atractivo visual y capacidad de generar una experiencia positiva. Esta evaluación no solo aporta evidencia empírica sobre el desempeño del producto, sino que también orienta futuras mejoras desde una perspectiva centrada en el usuario.

Con el propósito de que la validación fuera más consistente con la información, se buscó un mayor número de participantes con daltonismo. En el proceso de validación, se aplicó la prueba a 22 usuarios más con daltonismo, dando como resultado 32 usuarios que realizaron la evaluación de la extensión respondiendo al formulario diseñado para tal efecto.

4.2 Justificación del instrumento

El *UEQ* fue seleccionado como instrumento principal de validación por su robustez metodológica y amplia adopción en estudios académicos. Desarrollado por Thomaschewski, Rauschenberger y Schrepp, el *UEQ* permite evaluar seis dimensiones clave de la experiencia de usuario:

- Atractivo (*Attractiveness*): impresión general del producto, combinando todos los aspectos evaluados.
- Claridad (*Perspicuity*): facilidad para comprender y aprender a usar el sistema.
- Eficiencia (*Efficiency*): rapidez y esfuerzo necesario para completar tareas.
- Fiabilidad (*Dependability*): sensación de control, seguridad y predictibilidad.
- Estimulación (*Stimulation*): grado de motivación e interés generado por el producto.
- Novedad (*Novelty*): percepción de innovación y creatividad.

Estas dimensiones se miden mediante pares de adjetivos opuestos en una escala de diferencial semántico de siete puntos, lo que permite capturar matices en la percepción del usuario. Tal como se muestra en la Figura 4.1.

Figura 4.1
Ejemplo de escala diferencial para *UEQ*.

atractivo	☐	☒	☐	☐	☐	☐	☐	no atractivo
-----------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	--------------

La elección del *UEQ* se fundamenta en su capacidad para ofrecer resultados cuantitativos comparables con los puntos de vista (*benchmarks*) internacionales, así como en su facilidad de implementación y análisis mediante herramientas automatizadas.

Además, se utilizó el *UEQ Handbook* como guía metodológica para la correcta aplicación e interpretación de los resultados, y se empleó la versión oficial en español del cuestionario para asegurar la comprensión por parte de los participantes.

4.3 Procedimiento de aplicación

La aplicación del cuestionario *UEQ* se realizó posterior a la interacción de los usuarios con la extensión desarrollada. Se seleccionó una muestra de participantes representativa del público objetivo, quienes utilizaron la herramienta en condiciones reales de uso. Una vez finalizada la experiencia, se les solicitó completar el cuestionario *UEQ* en su versión en español.

El cuestionario fue distribuido en formato digital, permitiendo una recolección eficiente de los datos. Cada participante evaluó 26 ítems agrupados en las seis dimensiones mencionadas, utilizando una escala de 7 puntos que va de “muy negativo” a “muy positivo”.

Los datos recolectados fueron procesados mediante la hoja de cálculo oficial proporcionada por el equipo *UEQ*, la cual permite calcular:

- Media y desviación estándar por ítem y por dimensión.
- Comparación con valores de referencia.
- Visualización gráfica de los resultados.

Este procedimiento garantiza la objetividad del análisis y permite identificar fortalezas y áreas de mejora en la experiencia de usuario ofrecida por la extensión. Asimismo, se respetaron los principios éticos de participación voluntaria, anonimato y consentimiento informado.

4.4 Resultados obtenidos

Los resultados obtenidos mediante la aplicación del cuestionario *UEQ* reflejan una experiencia de usuario altamente positiva en todas las dimensiones evaluadas. A continuación, se presentan los valores medios por dimensión:

En la tabla 4.1 se muestra la tabla de cada una de las escalas diferenciales con su debida escala. En esta tabla se calcularon los promedios tomando en cuenta los valores de la escala de 1 a 7 de los 33 participantes.

Tabla 4.1*Promedios por cada una de las preguntas.*

Nº	Escala	Valor
1	Atracción	5,9375
2	Claridad	5,677419355
3	Estímulo	2,4375
4	Claridad	2,15625
5	Atracción	2,46875
6	Atracción	5,5
7	Eficiencia	6,09375
8	Estímulo	4,75
9	Eficiencia	2,34375
10	Claridad	2,375
11	Fiabilidad	5,5
12	Atracción	2,03125
13	Atracción	5,78125
14	Fiabilidad	5,65625
15	Estímulo	5,3125
16	Claridad	5,75
17	Fiabilidad	2,1875
18	Eficiencia	2,25
19	Originalidad	2,3125
20	Originalidad	5,75
21	Atracción	2,21875
22	Estímulo	5,161290323
23	Originalidad	2,46875
24	Originalidad	2,28125
25	Atracción	2,09375
26	Originalidad	4,875

En la tabla 4.2 se muestran los promedios de la tabla 4.1 tomando en cuenta solo los valores correspondientes a cada una de las escalas de *UEQ* y se separó por dimensión y valor promedio. Aunque en *UEQ* se utiliza una tabla cuyos valores van de -3 a 3, se utilizó la escala de 1 a 7 debido a que esto ofrece una representación más intuitiva y familiar tanto para los usuarios como para los lectores. Debido a ello los promedios obtenidos no presentan números negativos.

Tabla 4.2

Promedios por cada dimensión de UEQ.

Dimensión	Valor promedio
Atracción	3.72
Claridad	3.99
Eficiencia	3.56
Fiabilidad	4.45
Estímulo	4.42
Originalidad	3.54

Como puede observarse, todos los valores se encuentran por encima de 3.5, lo que indica una percepción muy positiva por parte de los usuarios. Según los rangos establecidos en el *UEQ Handbook*, estos resultados se consideran excelentes, ya que superan ampliamente el umbral de 1.5 que define una experiencia destacada.

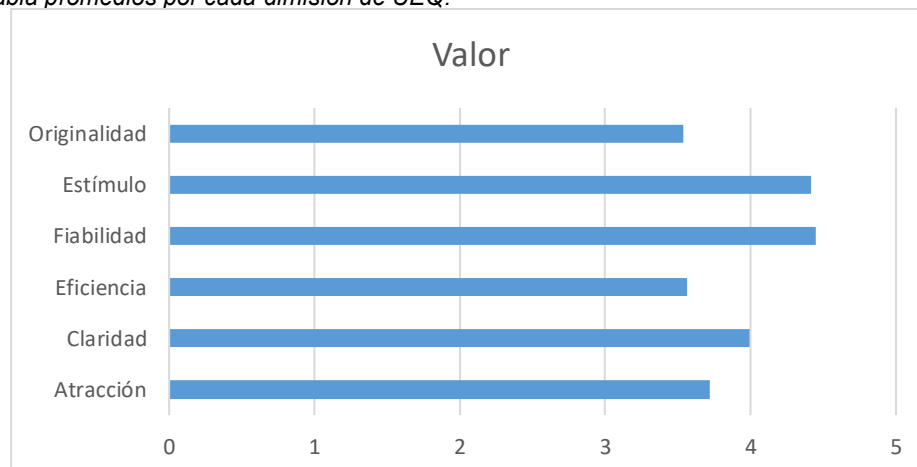
A continuación, se detalla la interpretación de cada una de las dimensiones *UEQ* obtenidas en la evaluación de la extensión hecha por los encuestados:

- *Claridad* (3.99) y *Fiabilidad* (4.45) fueron las dimensiones mejor valoradas, lo que sugiere que los usuarios encontraron la herramienta fácil de entender, confiable y segura en su funcionamiento.
- *Estímulo* (4.42) también obtuvo una puntuación alta, indicando que la extensión genera interés y motivación durante su uso.
- *Originalidad* (3.54) y *Atracción* (3.72) muestran que el diseño y propuesta de la herramienta fueron bien recibidos, aunque podrían explorarse mejoras visuales o funcionales para reforzar la percepción de innovación.

En la Figura 4.2 se ve representado el gráfico de la tabla de promedios de cada dimensión, la escala esta puesta de 0 a 5 según los promedios.

Figura 4.2

Gráfica de la tabla promedios por cada dimensión de UEQ.



4.5 Análisis crítico

El uso del *UEQ* en la evaluación de la extensión aporta un respaldo empírico sólido a la calidad de la experiencia de usuario ofrecida. Esta herramienta, ampliamente validada en la literatura, permite medir la UX en múltiples dimensiones que abarcan desde lo pragmático hasta lo hedónico, y desde lo cognitivo hasta lo emocional. Su aplicación, breve y sencilla, facilita la obtención de resultados confiables y comparables con una amplia base de referencia de productos en distintos dominios.

En el caso de la extensión analizada, los resultados del *UEQ* confirman que no solo cumple con criterios funcionales, sino que también genera una interacción clara, agradable y estimulante. Este hallazgo refuerza la pertinencia de la solución como alternativa accesible para personas con daltonismo, al demostrar que la experiencia de uso no se limita a la eficacia técnica, sino que también incorpora aspectos emocionales y de satisfacción.

Es importante señalar que, aunque el *UEQ* tradicionalmente utiliza una escala bipolar de -3 a +3, en algunos reportes los resultados se presentan en una escala de 1 a 7. Esta transformación, como se mencionó anteriormente, responde a la necesidad de ofrecer una representación más intuitiva y familiar para los usuarios y lectores, evitando valores negativos y alineándose con otros cuestionarios de satisfacción. No obstante, se trata de una re-escalación lineal que no altera la validez estadística ni la comparabilidad de los datos, manteniendo intacta la capacidad del *UEQ* para reflejar la percepción y expectativas de los usuarios.

Conclusiones

El desarrollo y validación de la extensión permitió evidenciar varios aspectos relevantes sobre la percepción del daltonismo en la sociedad y en el ámbito tecnológico. En primer lugar, se constató que muchas personas desconocen su condición de daltónicas hasta que realizan pruebas específicas, como lo es el test de *Ishihara*. Esta situación refleja la falta de conciencia general sobre la existencia y las implicaciones de esta discapacidad visual.

Asimismo, se identificó que las empresas desarrolladoras de software suelen dejar de lado a este sector de la población, lo que genera barreras de accesibilidad y limita la experiencia de los usuarios con daltonismo. En consecuencia, estos usuarios frecuentemente se sienten excluidos, hasta que encuentran herramientas o apoyos que apoyan su situación.

La extensión desarrollada demostró ser funcional y útil, ofreciendo una alternativa concreta para mejorar la interacción de las personas con daltonismo en entornos digitales. El proceso representó un reto personal y académico, ya que implicó enfrentar dificultades técnicas y metodológicas, como la comprensión de los archivos de la extensión y la metodología que lleva detrás un realizar una prueba de daltonismo, así

mismo permitió un aprendizaje significativo en torno al diseño inclusivo y la importancia de la accesibilidad digital.

Finalmente, la gratificación obtenida al observar el impacto positivo que la herramienta generó en los usuarios confirma la relevancia de este tipo de proyectos. Más allá de los resultados técnicos, la experiencia reafirma que la tecnología puede y debe ser un medio para promover la inclusión y mejorar la calidad de vida de quienes enfrentan condiciones poco visibilizadas.

Trabajos futuros

El proyecto desarrollado abre la posibilidad de realizar diversas mejoras que incrementen su alcance y utilidad.

La primera línea de trabajo consiste en actualizar la extensión para que se adapte con precisión a los distintos tipos de daltonismo y las gamas de colores asociadas. Esto permitirá ofrecer una experiencia más personalizada y adaptada a las necesidades específicas de cada usuario, fortaleciendo la accesibilidad y la inclusión digital.

La segunda línea de trabajo es la adaptación de la extensión a otros navegadores, permitiendo una diversidad de accesibilidad entre los distintos tipos de sistemas operativos tales como MacOS y Linux.

La tercera línea de trabajo es la migración de la extensión hacia dispositivos móviles. Dado que gran parte de la interacción tecnológica actual ocurre en teléfonos inteligentes y tabletas, contar con una versión móvil garantizará que más personas puedan beneficiarse de la herramienta en cualquier momento y lugar, ampliando significativamente su impacto.

La cuarta línea de trabajo sería, la promoción del test de Ishihara en entornos empresariales. Al implementar este tipo de pruebas en las organizaciones, se puede generar mayor conciencia sobre la existencia del daltonismo y fomentar prácticas inclusivas que consideren las necesidades de los empleados que presentan esta condición. De esta manera, se contribuye a la construcción de ambientes laborales más equitativos y sensibles a la diversidad.

La última línea de trabajo planteada es la difusión en redes sociales representa una estrategia clave para llegar a un público más amplio. Al compartir información sobre el daltonismo y la existencia de herramientas de apoyo, se puede brindar acompañamiento a usuarios que se sienten incomprendidos o que desconocen su condición. Fortaleciendo la visibilidad del tema y ayuda a crear comunidades de apoyo, recordando que existen soluciones tecnológicas pensadas para ellas.

Apéndice 1. Preguntas del test Ishihara

En este apartado se presentan todas las láminas utilizadas para la aplicación del test de *Ishihara* en formato para *web*. Cada sección incluye las diferentes láminas correspondientes junto con sus posibles respuestas, tal como fueron mostradas a los participantes durante la evaluación.

En este apartado se muestran las *cards* del del diseño del cuestionario de Ishihara que se aplicó a la población encuestada. Las *cards* fueron diseñadas de acuerdo a la página oficial del test de *Ishihara*.

Se decidió aplicar la versión de 24 láminas del test de *Ishihara* en lugar de la versión extensa de 80, debido a consideraciones metodológicas y prácticas. La versión abreviada mantiene la validez diagnóstica necesaria para identificar los principales tipos de deficiencia cromática (protanopia y deuteranopia), pero reduce significativamente el tiempo de aplicación y la fatiga de los participantes, lo que garantiza respuestas más confiables y una mejor experiencia durante el cuestionario. Además, al integrarse en un entorno *web* con otros instrumentos de evaluación, la versión de 24 láminas resulta más adecuada para estudios con muestras amplias, ya que permite obtener resultados precisos sin sobrecargar el proceso de recolección de datos.

El diseño del cuestionario se favorece la concentración del participante al mostrar una sola lámina por vez, reduce la sobrecarga cognitiva y permite adaptar el test a dispositivos móviles sin comprometer la calidad diagnóstica.

Como resultado de lo anteriormente descrito, el cuestionario del presente trabajo se presenta a los usuarios como se muestra en las Figuras A2.1 a la A2.6

Figura A1.1
Cards del Test Ishihara de la lámina 1 a la 4.

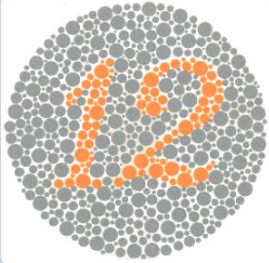
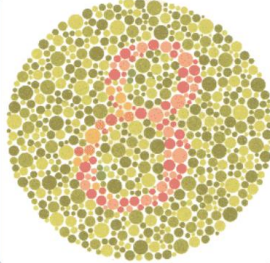
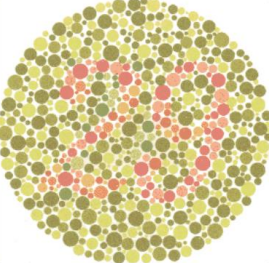
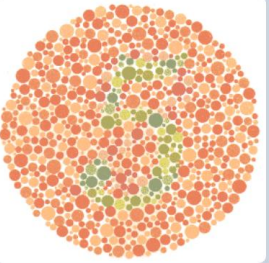
Test de Daltonismo	Test de Daltonismo	Test de Daltonismo	Test de Daltonismo
Imagen 1 ¿Qué ves en la imagen?	Imagen 2 ¿Qué ves en la imagen?	Imagen 3 ¿Qué ves en la imagen?	Imagen 4 ¿Qué ves en la imagen?
			
12	8	70	2
2	3	29	5
Nada	Nada	Nada	Nada

Figura A1.2
Cards del Test Ishihara de la lámina 5 a la 8.

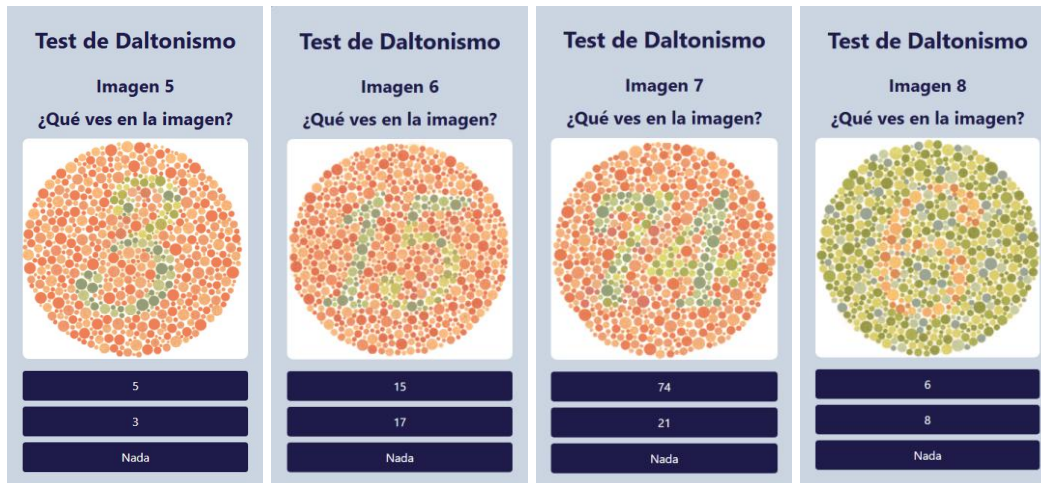


Figura A1.3
Cards del test Ishihara de la lámina 9 a la 12.

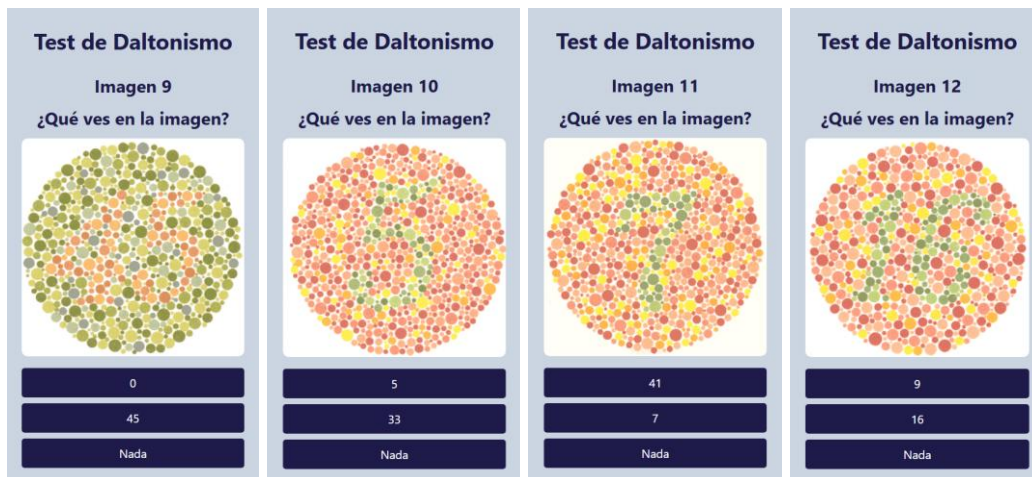


Figura A1.4
Cards del Test Ishihara de la lámina 13 a la 16.

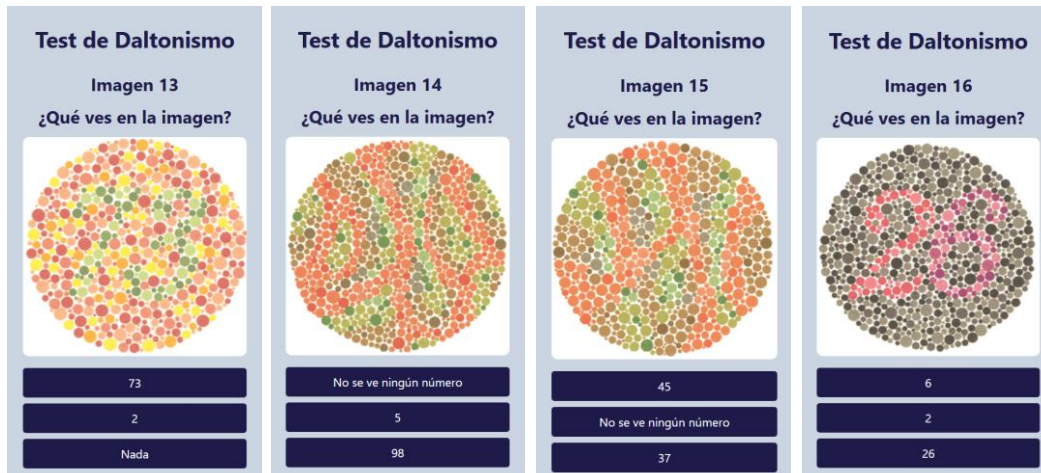


Figura A1.5
Cards del Test Ishihara de la lámina 17 a la 20.

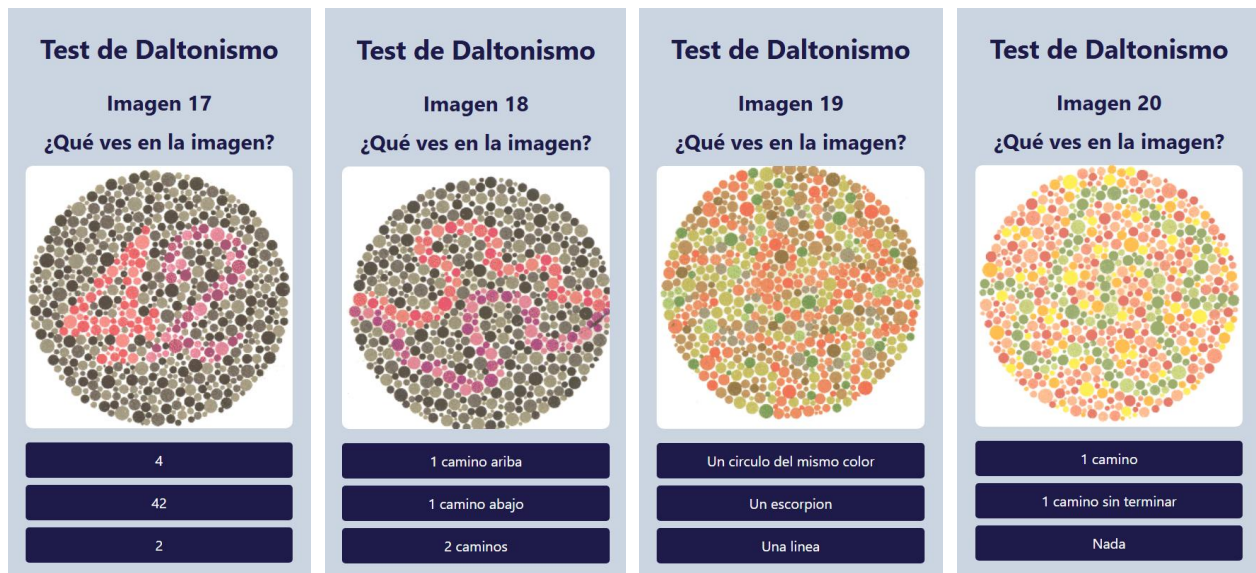
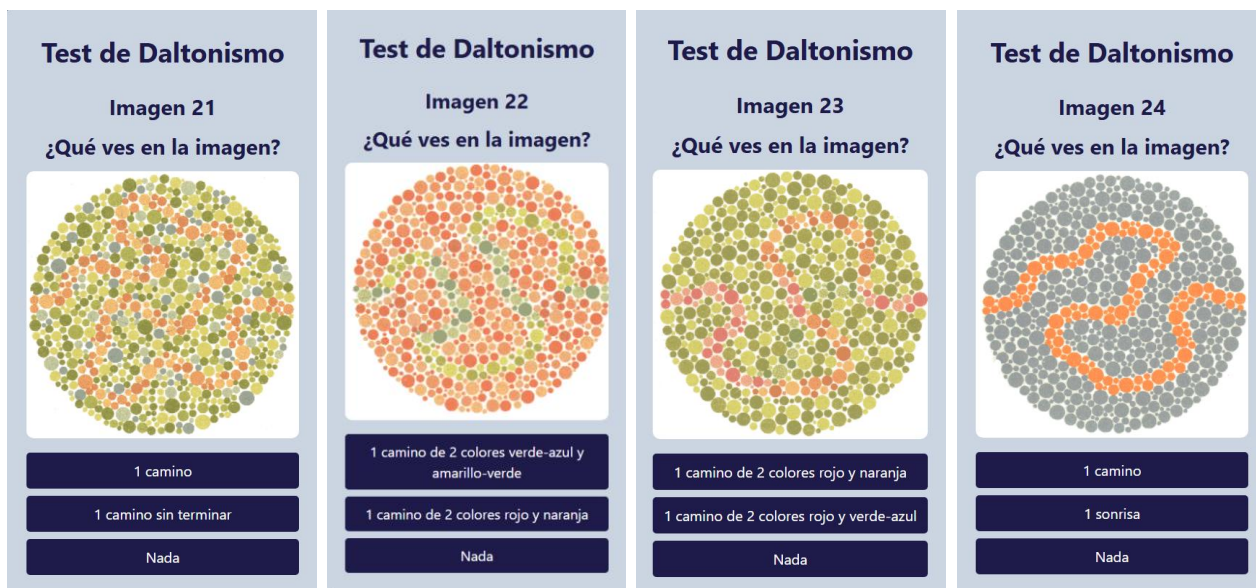


Figura A1.6
Cards del Test Ishihara de la lámina 21 a la 24.



Anexo 1: Tabla de los 78 criterios de WCAG 2.2

Las pautas de accesibilidad para el contenido *web* (WCAG 2.2) constituyen el estándar internacional más reconocido para garantizar que los sitios *web* y aplicaciones digitales sean accesibles para todas las personas, incluidas aquellas con discapacidades visuales, auditivas, motrices y cognitivas.

En esta versión, se definen 78 criterios de éxito heredados de WCAG 2.1, organizados bajo los cuatro principios fundamentales de la accesibilidad: perceptible, operable, comprensible y robusto (*POUR*). Cada criterio establece una regla verificable que permite evaluar si el contenido cumple con los niveles de conformidad A, AA o AAA, que representan desde los requisitos mínimos hasta los más avanzados.

La importancia de estos criterios radica en que ofrecen un marco común para diseñar experiencias digitales inclusivas, mejorar la usabilidad general y cumplir con normativas legales en distintos países. Además, WCAG 2.2 incorpora nuevos criterios que refuerzan la accesibilidad en contextos móviles y cognitivos, ampliando el alcance de las pautas y consolidando su papel como referencia esencial en el desarrollo *web* responsable.

Tabla A-nexo 1.1

Tabla de criterios WCAG 2.2.

Nº	Criterio	Nivel	Descripción breve
1.1.1	Texto alternativo	A	Proporcionar texto alternativo para imágenes y elementos no textuales.
1.2.1	Solo audio/vídeo (prerregistro)	A	Ofrecer alternativa para contenido solo audio o vídeo grabado.
1.2.2	Subtítulos (prerregistro)	A	Incluir subtítulos en vídeos grabados.
1.2.3	Audiodescripción/alternativa multimedia	A	Proporcionar audiodescripción o alternativa textual para vídeos.
1.2.4	Subtítulos (en directo)	AA	Subtítulos en transmisiones en vivo.
1.2.5	Audiodescripción (prerregistro)	AA	Añadir audiodescripción en vídeos grabados.
1.2.6	Lengua de señas (prerregistro)	AAA	Proporcionar interpretación en lengua de señas.
1.2.7	Audiodescripción ampliada	AAA	Versión extendida con más detalles narrados.
1.2.8	Alternativa multimedia	AAA	Texto completo que describe el contenido multimedia.
1.2.9	Solo audio (en directo)	AAA	Alternativa para transmisiones solo audio en vivo.
1.3.1	Información y relaciones	A	Mantener relaciones semánticas en el contenido.
1.3.2	Secuencia significativa	A	El orden de lectura debe ser lógico.
1.3.3	Características sensoriales	A	No depender solo de color, forma o sonido.
1.3.4	Orientación	AA	Contenido usable en cualquier orientación (vertical/horizontal).
1.3.5	Propósito de entrada	AA	Identificar propósito de campos de formulario.
1.3.6	Identificación del propósito	AAA	Identificar propósito de componentes de interfaz.
1.3.7	Accesibilidad de contenido visible	AA	Asegurar que el contenido visible sea accesible.
1.4.1	Uso del color	A	No usar solo color para transmitir información.
1.4.2	Control de audio	A	Permitir detener o ajustar audio automático.
1.4.3	Contraste mínimo	AA	Texto debe tener contraste suficiente con fondo.
1.4.4	Redimensionar texto	AA	Texto ampliable sin pérdida de funcionalidad.
1.4.5	Imágenes de texto	AA	Evitar imágenes de texto salvo necesidad.

Tabla A-nexo 1.1
(Continuación).

Nº	Criterio	Nivel	Descripción breve
1.4.6	Contraste mejorado	AAA	Contraste más alto para texto.
1.4.7	Sonido de fondo bajo	AAA	Reducir sonido de fondo en audio.
1.4.8	Presentación visual	AAA	Ofrecer opciones de personalización visual.
1.4.9	Imágenes de texto (sin excepción)	AAA	No usar imágenes de texto en absoluto.
1.4.10	Reflujo	AA	Contenido debe adaptarse sin desplazamiento horizontal.
1.4.11	Contraste de elementos no textuales	AA	Controles e iconos con contraste suficiente.
1.4.12	Espaciado del texto	AA	Permitir ajuste de espaciado sin pérdida.
1.4.13	Contenido en hover/focus	AA	Contenido emergente debe ser accesible.
2.1.1	Teclado	A	Todo debe ser operable con teclado.
2.1.2	Sin trampas de teclado	A	No bloquear al usuario en un área con teclado.
2.1.3	Teclado (sin excepción)	AAA	Todo operable sin ratón.
2.1.4	Atajos de teclado	A	Permitir desactivar o modificar atajos.
2.1.5	Acceso rápido	AA	Proporcionar métodos rápidos de acceso.
2.1.6	Atajos personalizables	AAA	Atajos deben ser configurables.
2.2.1	Ajuste de tiempo	A	Dar opción de ampliar límites de tiempo.
2.2.2	Pausar, detener, ocultar	A	Control sobre contenido en movimiento.
2.2.3	Sin tiempo	AAA	No imponer límites de tiempo.
2.2.4	Interrupciones	AAA	Permitir posponer interrupciones.
2.2.5	Re autenticación	AAA	Mantener datos al volver a autenticarse.
2.3.1	Umbral de parpadeo	A	Evitar destellos que causen convulsiones.
2.3.2	Umbral mejorado	AAA	Límites más estrictos para destellos.
2.4.1	Evitar bloques	A	Saltar bloques repetitivos.
2.4.2	Título de página	A	Páginas deben tener título descriptivo.
2.4.3	Orden de foco	A	Orden lógico al navegar con teclado.
2.4.4	Propósito de enlaces (en contexto)	A	Enlaces deben ser comprensibles.
2.4.5	Múltiples vías	AA	Varias formas de encontrar contenido.
2.4.6	Encabezados y etiquetas	AA	Usar encabezados claros y etiquetas.
2.4.7	Foco visible	AA	El foco debe ser claramente visible.
2.4.8	Ubicación	AAA	Indicar ubicación del usuario en el sitio.
2.4.9	Propósito de enlaces (solo enlace)	AAA	Enlaces claros sin contexto adicional.
2.4.10	Encabezados de sección	AAA	Encabezados deben organizar contenido.
2.5.1	Gestos con puntero	A	Evitar gestos complejos.
2.5.2	Cancelación de puntero	A	Permitir cancelar acciones con puntero.
2.5.3	Nombre en etiquetas	A	Etiquetas deben coincidir con nombre visible.
2.5.4	Movimiento	A	Funcionalidad no debe depender solo de movimiento.
2.5.5	Tamaño del objetivo	AA	Áreas táctiles suficientemente grandes.
3.1.1	Idioma de la página	A	Definir idioma principal.
3.1.2	Idioma de las partes	AA	Indicar idioma de fragmentos distintos.
3.2.1	Al recibir foco	A	No cambiar contexto automáticamente.
3.2.2	Al recibir entrada	A	No cambiar contexto inesperadamente.
3.2.3	Navegación consistente	AA	Menús y navegación coherentes.
3.2.4	Identificación consistente	AA	Etiquetas y nombres consistentes.
3.2.5	Cambio a petición	AAA	Cambios solo si el usuario lo solicita.

Tabla A-nexo 1.1
(Continuación).

Nº	Criterio	Nivel	Descripción breve
3.3.1	Identificación de errores	A	Informar claramente errores en formularios.
3.3.2	Etiquetas o instrucciones	A	Proporcionar instrucciones claras.
3.3.3	Sugerencias de errores	AA	Ofrecer sugerencias para corregir errores.
3.3.4	Prevención de errores (legal, financiero, datos)	AA	Confirmar antes de enviar datos críticos.
3.3.5	Ayuda	AAA	Proporcionar ayuda contextual.
3.3.6	Prevención de errores (todos)	AAA	Confirmación para cualquier envío.
3.3.7	Redundancia de entrada	AAA	Evitar pedir la misma información repetida.
4.1.1	Procesamiento	A	Código debe ser válido y procesable.
4.1.2	Nombre, rol, valor	A	Elementos deben tener nombre, rol y valor definidos.
4.1.3	Mensajes de estado	AA	Mensajes deben ser anunciados por tecnologías de asistencia.
4.1.4	Anotaciones	AAA	Proporcionar anotaciones semánticas adicionales.

Anexo 2: Test Ishihara

El test de *Ishihara* es una de las pruebas más utilizadas a nivel mundial para la detección de alteraciones en la visión cromática, especialmente el daltonismo. Fue desarrollado en 1917 por el médico japonés Shinobu Ishihara, y desde entonces se ha convertido en un estándar clínico por su sencillez, fiabilidad y rápida aplicación.

La prueba consiste en una serie de láminas que contienen círculos formados por puntos de diferentes colores y tamaños. Dentro de cada círculo se encuentra un número o figura que solo puede ser identificado correctamente si la persona posee una percepción normal de los colores. Aquellos con deficiencias en la visión cromática (como dificultad para distinguir entre tonos rojos y verdes) presentan errores característicos al intentar reconocer las figuras.

El objetivo principal del test es evaluar la capacidad de discriminación cromática y detectar de manera temprana anomalías que pueden afectar la vida cotidiana, el desempeño académico o profesional, y la seguridad en actividades que requieren una percepción precisa de los colores.

Su aplicación es rápida, no invasiva y ampliamente aceptada en ámbitos médicos, laborales y educativos, lo que lo convierte en una herramienta esencial dentro de los protocolos de evaluación oftalmológica.

Tabla A-nexo 2.1

Tabla de Láminas del test Ishihara.

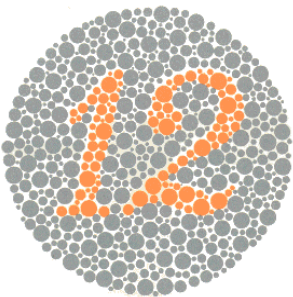
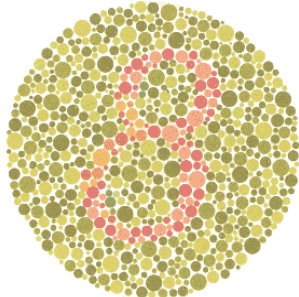
Nº de lámina	Lamina	Descripción de lo que evalúa
1		Todas las personas (con o sin daltonismo) deben ver el número 12.
2		Evalúa la capacidad de distinguir números bajo contraste rojo-verde.

Tabla A-nexo 2.1
(Continuación)

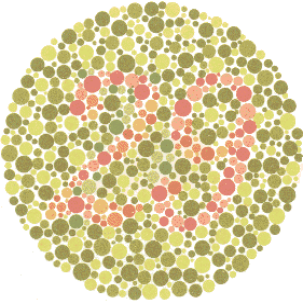
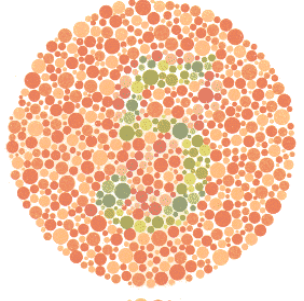
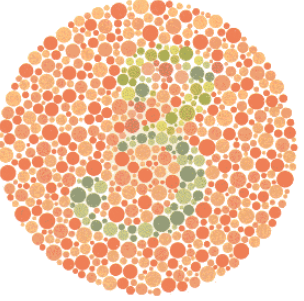
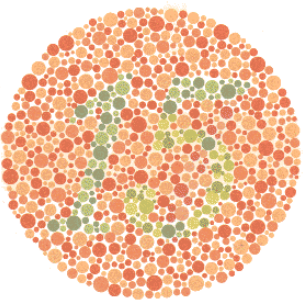
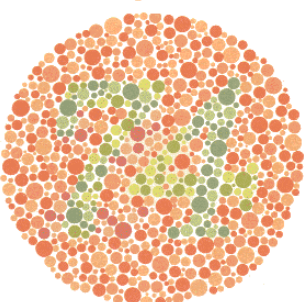
Nº de lámina	Lamina	Descripción de lo que evalúa
3		Explora la percepción de números con variaciones de color, útil para detectar protanopía y deuteranopía.
4		Evalúa la discriminación cromática en figuras numéricas con tonos cercanos.
5		Similar a la anterior, busca respuestas frente a números ocultos en patrones.
6		Evalúa la identificación de números con contraste reducido.
7		Explora la percepción de números en tonos verdes y rojos.

Tabla A-nexo 2.1
(Continuación)

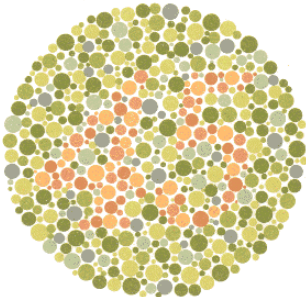
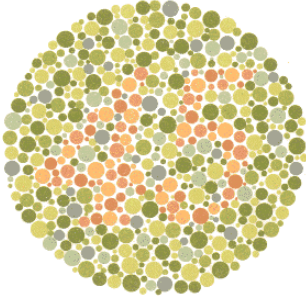
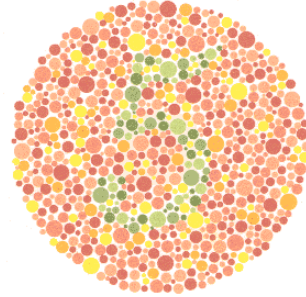
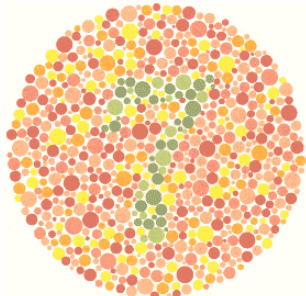
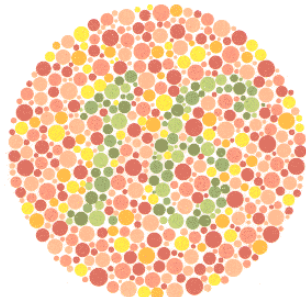
Nº de lámina	Lamina	Descripción de lo que evalúa
8		Evalúa la capacidad de distinguir números con mezcla de colores.
9		Evalúa la percepción de números en condiciones de ambigüedad cromática.
10		Evalúa la discriminación rojo-verde en números con contraste medio.
11		Evalúa la percepción de números en tonos cercanos, útil para diferenciar protanopes y deuteranopes.
12		Evalúa la identificación de números con mezcla de colores, midiendo sensibilidad cromática.

Tabla A-nexo 2.1
(Continuación)

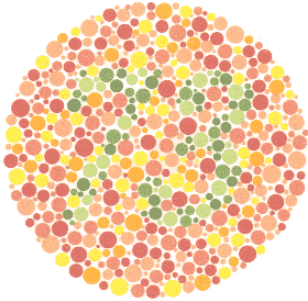
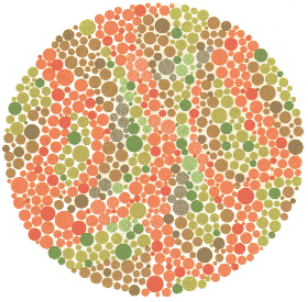
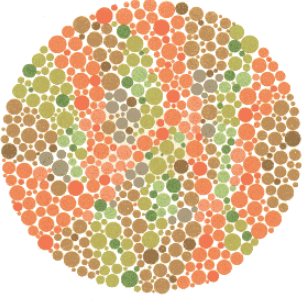
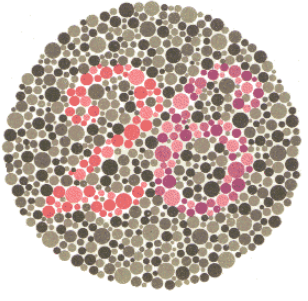
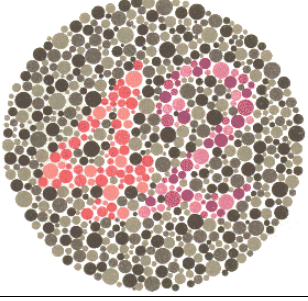
Nº de lámina	Lamina	Descripción de lo que evalúa
13		Evalúa la percepción de números en condiciones de ambigüedad cromática más compleja.
14		Control intermedio: personas con visión normal no ven nada; quienes tienen daltonismo rojo-verde ven un "5".
15		Personas con daltonismo rojo-verde ven un "45".
16		Protanopes ven un "6"; deuteranopes ven un "2".
17		Protanopes ven un "2"; deuteranopes ven un "4".

Tabla A-nexo 2.1
(Continuación)

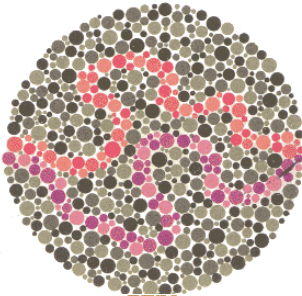
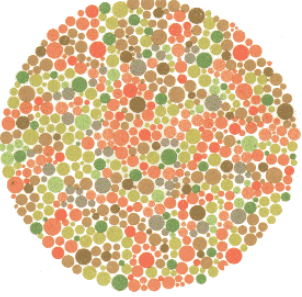
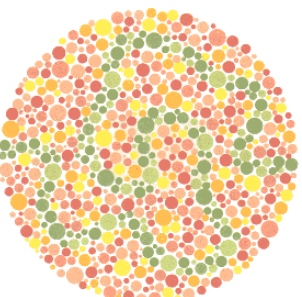
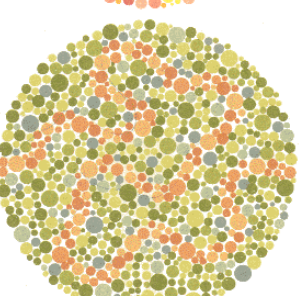
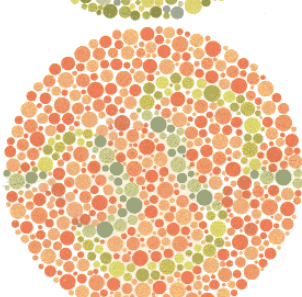
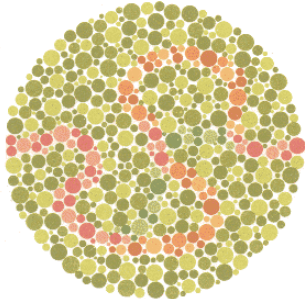
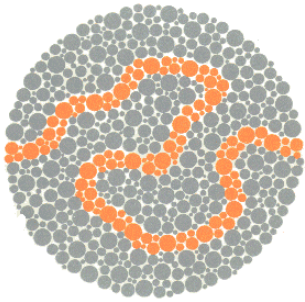
Nº de lámina	Lamina	Descripción de lo que evalúa
18		Visión normal traza dos líneas (roja y púrpura). Protanopes solo la púrpura; deuteranopes solo la roja.
19		Personas con daltonismo rojo-verde pueden trazar la línea dependiendo de la severidad.
20		Visión normal: línea verde.
21		Visión normal: línea naranja.
22		Visión normal: línea azul-verde/amarillo-verde.

Tabla A-nexo 2.1
(Continuación)

Nº de lámina	Lamina	Descripción de lo que evalúa
23		Visión normal: líneas roja y naranja. Daltonismo rojo-verde: roja y azul-verde.
24		Todas las personas deben poder trazar la línea. Sirve para verificar consistencia.

Referencias

- ¿Qué es tritanopsia? *Diccionario Médico*. (s.f.). Clínica U. Navarra.: <https://www.cun.es/diccionario-medico/terminos/tritanopsia>
- Albornoz, M. C., Berón, M., & Montejano, G. (28 de agosto de 2024). *UNLP*. https://sedici.unlp.edu.ar/bitstream/handle/10915/62078/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Alvites, M. A. (2015). Daltonismo y uso del Computador en Educación a Distancia. *Hamut'ay*, 32-48.
- Amazon Web Services, I. (28 de Agosto de 2024). ¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS. <https://aws.amazon.com/es/what-is/api/>
- Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Alemania: Blue Hole Press.
- Atlassian. (s.f.).
- Atlassian. (24 de abril de 2025). *Kanban*. Atlassian: <https://www.atlassian.com/es/agile/kanban>
- Atlassian. (25 de Abril de 2025). *Scrum*. Atlassian: <https://www.atlassian.com/es/agile/scrum/>
- Avanzada, A. O. (28 de Agosto de 2024). *Area Oftalmologica*. <https://areaoftalmologica.com/neurooftalmologia/discromatopsia/>
- Awareness, C. B. (27 de Abril de 2022). *About Colour Blindness*. Colour Blind Awareness: <https://www.colourblindawareness.org/colour-blindness/>
- Awareness, C. B. (27 de Abril de 2022). *Causes of Colour Blindness*. Colour Blind Awareness: <https://www.colourblindawareness.org/colour-blindness/causes-of-colour-blindness/>
- Beck, K. (2022). *Test Driven Development: By Example*. Reino Unido: Pearson Education.
- Berners-Lee, T., & Fischetti, M. (1999). *Weaving the Web The Past, Present and Future of the World Wide Web by Its Inventor*. Orion Business.
- Computación, D. d. (s.f.). *Introducción a la Operación de Computadoras Personales*. Argentina: Universidad Nacional del Sur.
- Crovi, D., Aguirre, D., Apodaca, J., & Camacho, O. (2002). Página Web. Una propuesta para su análisis. *Revista Mexicana de Ciencias Políticas y Sociales*, 167185.
- de las Heras del Dedo, R., Álvarez García, A., & Lasa Gómez, C. (2018). *Métodos Ágiles. Scrum, Kanban, Lean*. España: ANAYA MULTIMEDIA.
- Design, U. C. (s.f.). *Crux Collaborative*. <https://cruxcollaborative.com/insights/understanding-color-blindness-guide-to-accessible-design>
- Digital55. (17 de Mayo de 2020). *Digital55*. Firebase: qué es, para qué sirve, funcionalidades y ventajas: <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones>
- Eric Meyer, E. W. (2023). *CSS: The Definitive Guide*. O'Reilly Media.
- Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. 2024: <https://dle.rae.es/software>
- Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. <https://dle.rae.es/navegador>

Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. <https://dle.rae.es/filtro>

Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. <https://dle.rae.es/daltonismo>

Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. <https://dle.rae.es/interfaz>

Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. <https://dle.rae.es/sitio>

Española, R. A. (28 de Agosto de 2024). *Real Academia Española*. <https://dle.rae.es/p%C3%A1gina>

Figma. (s.f.). *Figma*. Herramienta de diseño UX y software UX gratuitos en línea: <https://www.figma.com/es-la/herramienta-diseno-ux/>

Frost, B. (2016). *Atomic design*. Brad Frost Web.

Furtado, R. (28 de Agosto de 2024). *Vodien*. <https://www.vodien.com/learn/what-are-domain-extensions-2/>

Google. (28 de Agosto de 2024). *Chrome for Developers*. <https://developer.chrome.com/docs/extensions/develop/concepts/content-scripts?hl=es-419>

Google. (28 de Agosto de 2024). *Chrome for Developers*. <https://developer.chrome.com/docs/extensions/how-to?hl=es-419>

Google. (28 de Agosto de 2024). *Chrome for Developers*. <https://developer.chrome.com/docs/extensions/develop?hl=es-419>

Google. (2026). *Chrome Extensions Documentation: Architecture overview*. Chrome for Developers: <https://developer.chrome.com/docs/extensions/develop?hl=es-419>

How to determine color blindness: A Nurse's guide. (1 de Septiembre de 2023). Maryville Nursing: <https://nursing.maryville.edu/blog/nurses-guide-to-assessing-color-blindness.html>

Inga, G. F. (2022). *Desarrollo e implementación de una extensión para el navegador web Google Chrome para análisis y evaluación de accesibilidad de una página web como complemento de herramienta de análisis automático de accesibilidad web de la Universidad Politécnica salesi*. Ecuador: Universidad Politecnica Salesiana.

Initiative. (s.f.). *W.A.A.* Resumen de los estándares de accesibilidad de W3C: <https://www.w3.org/WAI/standards-guidelines/es>

Institute, N. E. (s.f.). *Daltonismo*. National Eye Institute : <https://www.nei.nih.gov/espanol/aprenda-sobre-la-salud-ocular/enfermedades-y-afecciones-de-los-ojos/daltonismo>

Institute, N. E. (s.f.). *Tipos de Daltonismo*. National Eye Institute : <https://www.nei.nih.gov/espanol/aprenda-sobre-la-salud-ocular/enfermedades-y-afecciones-de-los-ojos/daltonismo/tipos-de-daltonismo>

Ishihara test for color blindness. (2 de Junio de 2021). <https://www.colour-blindness.com/colour-blindness-tests/ishihara-colour-test-plates/>

Jeff Gothelf, J. S. (2016). *Lean UX Designing Great Products with Agile Teams*. O'Reilly Media.

JMJ INFORMÁTICO. (28 de Agosto de 2024). <https://jmjinformatico.es/que-es-un-modulo-en-programacion/>

Kholmatova, A. (2021). *Design Systems: Building digital products with consistency and efficiency*. Smashing Magazine.

Kyōkai, N. N. (1989). *Kanban Just-in Time at Toyota: Management Begins at the Workplace*. Reino Unido: Taylor & Francis.

- Lanthony, P. (2013). *The History of Color Blindness*. Paraguay: Wayenborgh Publishing.
- LAREDO, L. M. (2005). *FILTRO DE PÁGINAS WEB*. Barcelona: Universidad Abierta de Cataluña.
- Lillo Jover, J., & Moreira Villegas, H. (2013). *Percepción del color y daltonismos: Descripción, diagnóstico e intervención*. España: Ediciones Pirámide.
- Lillo, J., Moreira, H., & Melnikova, A. (2019). *Daltonismos, vidas de distinto color*. Mente y Cerebro.
- Mimenza, O. C. (28 de Agosto de 2024). *Psicología y Mente*. <https://psicologiaymente.com/clinica/daltonismo>
- Rehkopf, M. (2025 de abril de 2025). *Scrum*. Atlassian: <https://www.atlassian.com/es/agile/scrum>
- Rodriguez, A. Q. (28 de Agosto de 2024). *USMP*. https://www.usmp.edu.pe/pdf/Paper_SOCOTE%202013_Usabilidad_Web_para_usuarios_daltonicos.pdf
- SALAZAR ZÚÑIGA, J. E., & PUENTES ANDRADE, O. (2009). *MODULO DISEÑO DE SITIOS WEB*. Mexico: UNAD.
- Sarmiento Curillo, C. A., Escudero Sánchez, G. S., Robles Reyes, V., & Isuiza Valdera, I. D. (2023). Daltonismo y visión de color: explorando las causas y soluciones potenciales. *recimundo*, 59-65.
- Segaller, S. (1999). *Nerds 2.0.1: A Brief History of the Internet*. TV Books.
- Shtherland, J. (2015). *Scrum: El arte de hacer el doble de trabajo en la mitad de tiempo*. D.F.: Oceano.
- Silver, A. (21 de Junio de 2016). *Improving The Color Accessibility For Color-Blind Users*. Smashing Magazine: <https://www.smashingmagazine.com/2016/06/improving-color-accessibility-for-color-blind-users/>
- Sutherland, J., & Schwaber, K. (2020). *La Guía Definitiva de Scrum: Las Reglas del Juego*. Creative Commons.
- Tanenbaum, A. (1996). *Computer networks*. Prentice Hall PTR.
- Therapy, G. (s.f.). *Neitz Vision*. Gene Therapy: <http://www.neitzvision.com/research/gene-therapy/#crackingthecode>
- Turbert, D. (28 de Agosto de 2024). *American Academy of Ophtalmology*. <https://www.aaof.org/salud-ocular/enfermedades/daltonismo>
- UPANA. (28 de Agosto de 2024). *UPANA*. <https://ingenieria.upana.edu.gt/wp-content/uploads/2020/07/Edutek-dia-5-RESTful-api-RESTCONF2.pdf>
- Valencia, U. d. (28 de agosto de 2024). *Universidad de Valencia*. <https://www.uv.es/fragar/html/pdf/html13.pdf>
- W3C. (2018). *"Relative Luminance" and "Contrast Ratio" definitions*. Web Content Accessibility Guidelines (WCAG) 2.1: <https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html>
- WAI, W. (2023). *WCAG 2 de un vistazo*. Web Accessibility Initiative (WAI): <https://www.w3.org/WAI/standards-guidelines/wcag/glance/es>