



UNIVERSIDAD
NACIONAL
AUTÓNOMA DE
NICARAGUA,
MANAGUA
UNAN - MANAGUA

Facultad Regional Multidisciplinaria de Chontales

“Cornelio Silva Arguello”

Departamento de Ciencia, Tecnología y Salud.

Proyecto de Graduación para optar al título de:

Ingeniero en Sistemas de Información

TÍTULO:

Desarrollo de una aplicación web usando tecnología React JS, para la gestión y control de la medicación de pacientes con diabetes tipo 1 y mellitus tipo 2, en la ciudad de Juigalpa-Chontales durante el segundo Semestre del año 2023.

CARRERA:

Ingeniería en Sistemas de Información.

AUTORES:

Br. Javiera Nazaret Leiva García

Br. Donald Octavio Alvarez Bendaña

Br. Brando José Estrada Zambrana

TUTOR:

- M. Sc. Miriam Patricia Téllez Marín
- M. Sc. Saira María Urbina Cienfuegos

Juigalpa Chontales, diciembre 2023

¡A la Libertad por la Universidad!

Autores:

Javiera Nazaret Leiva García

Correo electrónico: javieraLeiva026@gmail.com

Donald Octavio Alvarez Bendaña

Correo electrónico: donaldoctavio0210@gmail.com

Brando José Estrada Zambrana

Correo electrónico: brandoestrada7@gmail.com

Tutores:

M. Sc. Miriam Patricia Téllez Marín

Correo: miriam.tellez@unan.edu.ni

M. Sc. Saira María Urbina Cienfuegos

Correo: surbina@unan.edu.ni

Valoración Docente

La tecnología es un recurso importante en la vida diaria de los seres humanos. Con el uso de aplicaciones web o móviles adecuadas es posible comunicarse con facilidad, realizar transacciones y gestionar servicios, esto agiliza procesos complejos y reduce costos significativos. Son incontables las aplicaciones que ofrecen soluciones innovadoras que incluyen herramientas relevantes con Inteligencia Artificial para dar respuesta a las necesidades de los usuarios.

Uno de los problemas actuales que existe es el control de la medicación, alimentación y actividad física para pacientes con diabetes, quienes deben cumplir con rutinas de ejercicios, dietas saludables y fármacos según indicaciones médicas, con base en este contexto se ha desarrollado una aplicación web que contribuye a mejorar la calidad de vida de las personas con este padecimiento.

En la documentación presentada del proyecto, es posible observar la aplicación de metodologías y estándares en la búsqueda de un trabajo con un grado de calidad pertinente, además de, la implementación de tecnologías innovadoras en un proceso tradicional, estas evidencias se corresponden con los requisitos establecidos por la UNAN Managua para aceptar este proyecto como defensa de grado, que demuestra el cumplimiento de un aprendizaje integral por parte de los estudiantes.

Por lo antes expuesto, se considera que los estudiantes **Javiera Nazaret Leiva García, Donald Octavio Alvarez Bendaña y Brando José Estrada Zambrana**, reúnen los requisitos para la evaluación del jurado examinador, en la defensa de trabajo de grado para optar al título de Ingeniero en Sistemas de Información.

M.Sc. Saira María Urbina Cienfuegos

M.Sc. Miriam Patricia Téllez Marín

Dedicatoria

Con profunda gratitud, dedicamos este logro en la carrera de Ingeniería de Información a Dios quien ha sido nuestro guía supremo, seguidamente a nuestros queridos padres, cuyo amor inquebrantable y sacrificio han sido la base sobre la cual construimos nuestros sueños, a nuestros apreciados familiares por su apoyo constante, y a nuestros tutores de carrera por su invaluable orientación cuya sabiduría nos ha llevado más allá de lo que jamás imaginamos. Este éxito no es solo nuestro, sino de todos ustedes que han iluminado nuestro camino. Con el corazón rebosante de gratitud, compartimos este logro con la esperanza de que también sea un reflejo de su influencia en nuestras vidas.

Br. Javiera Nazaret Leiva García.

Br. Donald Octavio Alvarez Bendaña.

Br. Brando José Estrada Zambrana.

Agradecimientos

Con profundo agradecimiento, deseo expresar mi reconocimiento a Dios, cuya presencia en mi vida ha sido mi fuente de fuerza y guía constante. A mi amada madre, cuyo amor incondicional y apoyo incansable han sido un faro en mi camino. A mi valiente padre, cuya sabiduría y ejemplo han moldeado mi carácter y ambiciones. A mis queridos hermanos, quienes han compartido risas, desafíos y triunfos, creando recuerdos invaluableles. Su influencia y amor han enriquecido mi vida de formas innumerables, y por ello, les agradezco desde lo más profundo de mi corazón.

Br. Javiera Nazaret Leiva García.

Con profunda gratitud, quiero expresar mi agradecimiento a Dios, quien ha sido la constante guía y fuente inquebrantable de fortaleza en mi vida. A mi querida madre, cuyo amor incondicional y apoyo sólido han sido el cimiento de mi existencia. A mi querido padre, cuya sabiduría y ejemplaridad han moldeado mi carácter y mis aspiraciones. Cada uno de ellos han sido parte de mi camino de manera única y significativa, y les estoy eternamente agradecido por las innumerables bendiciones que han traído a mi vida.

Br. Donald Octavio Alvarez Bendaña.

En este momento crucial de mi vida académica al concluir mi proyecto de graduación, quiero expresar mi profundo agradecimiento a quienes fueron fundamentales en este viaje. Agradezco a Dios por su guía y fortaleza. A mi amada esposa, por ser mi apoyo inquebrantable. A mis padres, por su amor y sacrificio. A mis compañeros de clase, por su amistad y colaboración. A mis maestros, por su sabiduría. Cada uno de ustedes fue esencial en este logro. Mi corazón rebosa de gratitud.

Br. Brando José Estrada Zambrana

Resumen

La falta de control y gestión de la diabetes es común debido a la falta de horarios regulares para la medicación, cambios en el estilo de vida, lo cual puede generar depresión y dificultades en el proceso de la enfermedad. Para ello se desarrolla la aplicación móvil Control Health para ayudar a las personas con diabetes. Ofreciendo registro de medicación con recordatorios, información actualizada sobre la enfermedad, dietas nutricionales, rutinas de ejercicio y agendamiento de citas médicas, incluyendo un ChatBot de la mano de un prediagnóstico que redirige a la cita. El desarrollo sigue la metodología SCRUM, enfocándose en historias de usuarios para desarrollo, diseño, análisis y entregables. Se utilizan tecnologías como React Js, Redux y Firebase para crear una aplicación de calidad, escalable y siempre accesible, con interfaces, control de estado y almacenamiento en la nube. La aplicación ofrece acceso desde cualquier dispositivo móvil/web para un control y gestión efectiva de la diabetes, utilizando funcionalidades específicas para resolver las dificultades diarias.

Palabras Clave: Control, gestión, diabetes, registro, medicación, dietas, rutinas, citas, chatbot, React JS, Redux, Firebase.

Índice

I.	Solución.....	1
II.	Diseño o Modelo	3
2.1	Diseño de funcionamiento.....	3
2.2	Diseño de navegación	5
2.3	Diseño arquitectónico.....	9
2.3.1	<i>Diagrama de Componente Nivel 0.....</i>	9
2.3.2	<i>Diagrama de Componente Nivel 2.....</i>	11
III.	Justificación De La Tecnología	12
3.1	Tecnología de desarrollo	12
3.2	Gestores de código	15
3.3	Tecnología de diseño.....	18
3.4	Tecnologías para prototipado.....	22
3.5	Base de Datos	26
IV.	Justificación de la Metodología.....	29
V.	Oportunidad de Mercado	32
VI.	Propuesta de Valor	34
VII.	Ciclo del proyecto.....	35
7.1	Historias de Usuario.....	35
7.1.1	<i>Historias de Usuario Paciente.....</i>	36
7.1.2	<i>Historias de Usuario Médico.....</i>	39
7.1.3	<i>Historias de Usuario Administrador</i>	41
7.2	Product Backlog	42
7.3	Product Backlog Organizado por Sprint	43
7.4	Sprint Backlog	43

7.4.1 <i>Sprint 1</i>	43
7.4.1.1 Entregables Sprint 1.	46
7.4.1.2 Prueba de software del Sprint 1.	49
7.4.2 <i>Sprint 2</i>	50
7.4.2.1 Entregables Sprint 2.	52
7.4.2.2 Pruebas de software Sprint 2.	56
7.4.3 <i>Sprint 3</i>	57
7.4.3.1 Entregables Sprint 3.	59
7.4.3.2 Pruebas de software Sprint 3.	61
7.4.4 <i>Sprint 4</i>	62
7.4.4.1 Entregables del Sprint 4.	63
7.4.4.2 Pruebas de software Sprint 4.	64
VIII. Presupuesto	65
IX. Marco Lógico.....	67
9.1 Árbol del problema	68
9.2 Árbol de objetivo	69
9.3 Árbol de Soluciones.....	70
X. Cronograma de Actividades	78
XI. Referencias.....	81
XII. ANEXOS	83
XIII. Plan de Pruebas	92
XIV. Manual de Usuario.....	146
XV. Manual Técnico.....	172

Índice de Figuras

Figura 1 Diseño o Modelo.....	4
Figura 2 Mapa de Navegación web	5
Figura 3 Diseño de Navegación del Paciente	6
Figura 4 Diseño de Navegación del Médico	7
Figura 5 Diseño de Navegación del Administrador	8
Figura 6 Diagrama de componente nivel 0	9
Figura 7 Diagrama de componente nivel 1	10
Figura 8 Diagrama de componente nivel 2	11
Figura 9 Informe de trabajo completado Sprint 1.....	45
Figura 10 HUP04-Espacios Nutricionales	46
Figura 11 HUP05-Espacios de Actividad Física	47
Figura 12 HUP06-registar Usuario Paciente	48
Figura 13 Prueba de Software del Sprint 1	49
Figura 14 Informe de trabajo completado Sprint 2.....	51
Figura 15 HUP01-Control de la medicación.....	52
Figura 16 HUP07-Creación de citas	53
Figura 17 HUP03-Espacios informativo de Diabetes	54
Figura 18 HUM04-Agendar Cita.....	54
Figura 19 HUP08-ChatBot.....	55
Figura 20 HUM01 Registrar usuario médico	55
Figura 21 Prueba de Software Sprint 2	56
Figura 22 Informe de trabajo completado Sprint 3.....	58
Figura 23 HUM03-Espacio del médico	59
Figura 24 HUM05-Creación del calendario	59
Figura 25 HU02-Notificaciones.....	60
Figura 26 Pruebas de software Sprint 3	61
Figura 27 HUA02- Estadísticas.....	63
Figura 28 HUA01-Crud Administrativo	63

Figura 29 Pruebas de Software Sprint 4.....	64
Figura 30 Árbol del Problema	68
Figura 31 Árbol de Objetivos.....	69
Figura 32 Árbol de Soluciones	70
Figura 33 Diagrama de GANTT.....	80

Índice de tablas

Tabla 1 Selección de la Tecnología de Desarrollo	12
Tabla 2 Selección del Gestor de Código	15
Tabla 3 Selección de la Tecnología de Diseño	18
Tabla 4 Selección de la Tecnología para Prototipado	22
Tabla 5 Selección de la Base de Datos	26
Tabla 6 Justificación de la Metodología	29
Tabla 7 Historia de Usuario Paciente 01	36
Tabla 8 Historia de Usuario Paciente 02	36
Tabla 9 Historia de Usuario Paciente 03	37
Tabla 10 Historia de Usuario Paciente 04	37
Tabla 11 Historia de Usuario Paciente 05	38
Tabla 12 Historia de Usuario Paciente 06	38
Tabla 13 Historia de Usuario Paciente 07	39
Tabla 14 Historia de Usuario Médico 01	39
Tabla 15 Historia de Usuario Médico 02	40
Tabla 16 Historia de Usuario Médico 03	40
Tabla 17 Historia de Usuario Médico 04	41
Tabla 18 Historia de Usuario Administrador 01	41
Tabla 19 Historia de Usuario Administrador 02	42
Tabla 20 Product Backlog	42
Tabla 21 Product Backlog organizado por Sprint	43
Tabla 22 Sprint 1	44
Tabla 23 Sprint 2	50
Tabla 24 Sprint 3	57
Tabla 25 Sprint 4	62
Tabla 26 Presupuesto	65
Tabla 27 Marco Lógico	71
Tabla 28 Cronograma de Actividades	78

Anexos

Anexo A. Modelo de Negocio CANVAS	84
Anexo B Encuesta para obtención de requerimientos de usuarios	85
Anexo C Aplicación web responsiva a móvil	86
Anexo D Experiencia vivida en Hackathon Nicaragua IA 2023	90

I. Solución

Control Health:

Una solución móvil integral para el manejo de la diabetes, Control Health es una innovadora aplicación móvil diseñada específicamente para ayudar a las personas con diabetes a controlar y manejar de manera efectiva su enfermedad. Esta aplicación multifuncional ofrece una amplia gama de características y herramientas que permiten a los usuarios llevar un estilo de vida saludable y mantener un control óptimo de su diabetes.

Control Health ofrece:

Notificaciones personalizadas: Control Health cuenta con un sistema de notificaciones personalizadas que recuerdan a los usuarios sobre la toma de medicación, estas notificaciones ayudan a mantener una rutina constante y garantizan que los usuarios no se olviden de ninguna dosis diaria para su salud.

Dietas nutricionales: La aplicación incluye un apartado dedicado a proporcionar dietas nutricionales adaptadas a las necesidades de cada usuario. Con un amplio catálogo de opciones, los usuarios pueden acceder a planes de alimentación saludables y equilibrados que les ayuden a controlar los niveles de glucosa en sangre y mantener un peso adecuado.

Información y educación: Control Health brinda a los usuarios acceso a una amplia base de datos con información actualizada sobre la diabetes, sus tipos, síntomas, complicaciones y consejos sobre cómo sobrellevar la enfermedad. Esta sección educativa les permite comprender mejor su condición y tomar decisiones informadas en cuanto a su cuidado y tratamiento.

Ejercicio físico: Reconociendo la importancia del ejercicio físico en el manejo de la diabetes, Control Health ofrece un apartado con rutinas de ejercicio diseñadas especialmente para las personas con esta enfermedad. Los usuarios pueden acceder a videos de ejercicio adaptados a sus necesidades y capacidades, lo que les ayuda a mantener un estilo de vida activo y saludable.

Registro de medicación y citas médicas: La aplicación permite a los usuarios llevar un registro detallado de su medicación, incluyendo dosis y horarios, lo que facilita el seguimiento y evita olvidos.

Además, pueden programar y gestionar citas médicas con facilidad, teniendo acceso a un calendario que les ayuda a organizar sus consultas y tratamientos.

Chatbot: La aplicación permite al usuario, en calidad de paciente, realizar un prediagnóstico mediante su algoritmo, el cual calcula el porcentaje de probabilidad de padecer diabetes. Esto facilita la interacción con el proceso de programación de citas con un especialista.

Perfil médico: Control Health también ofrece a los médicos la posibilidad de crear un perfil dentro de la aplicación donde los médicos pueden agendar citas y enviar respuesta a los usuarios, lo que facilita la coordinación de la atención médica y mejora la adherencia al tratamiento.

Control Health es una aplicación web integral que proporciona a las personas con diabetes una solución completa para el manejo efectivo de su enfermedad. Desde notificaciones personalizadas y dietas nutricionales hasta información educativa, ejercicios físicos, registro de medicación y citas médicas, esta aplicación se convierte en un aliado indispensable para el control de la diabetes. Con Control Health, los usuarios pueden mejorar su estilo de vida, manteniendo su salud en óptimos niveles de glucosa que mejora su condición de manera significativa, mientras los médicos pueden ofrecer sus servicios de manera eficiente y brindar seguimiento personalizado a sus pacientes.

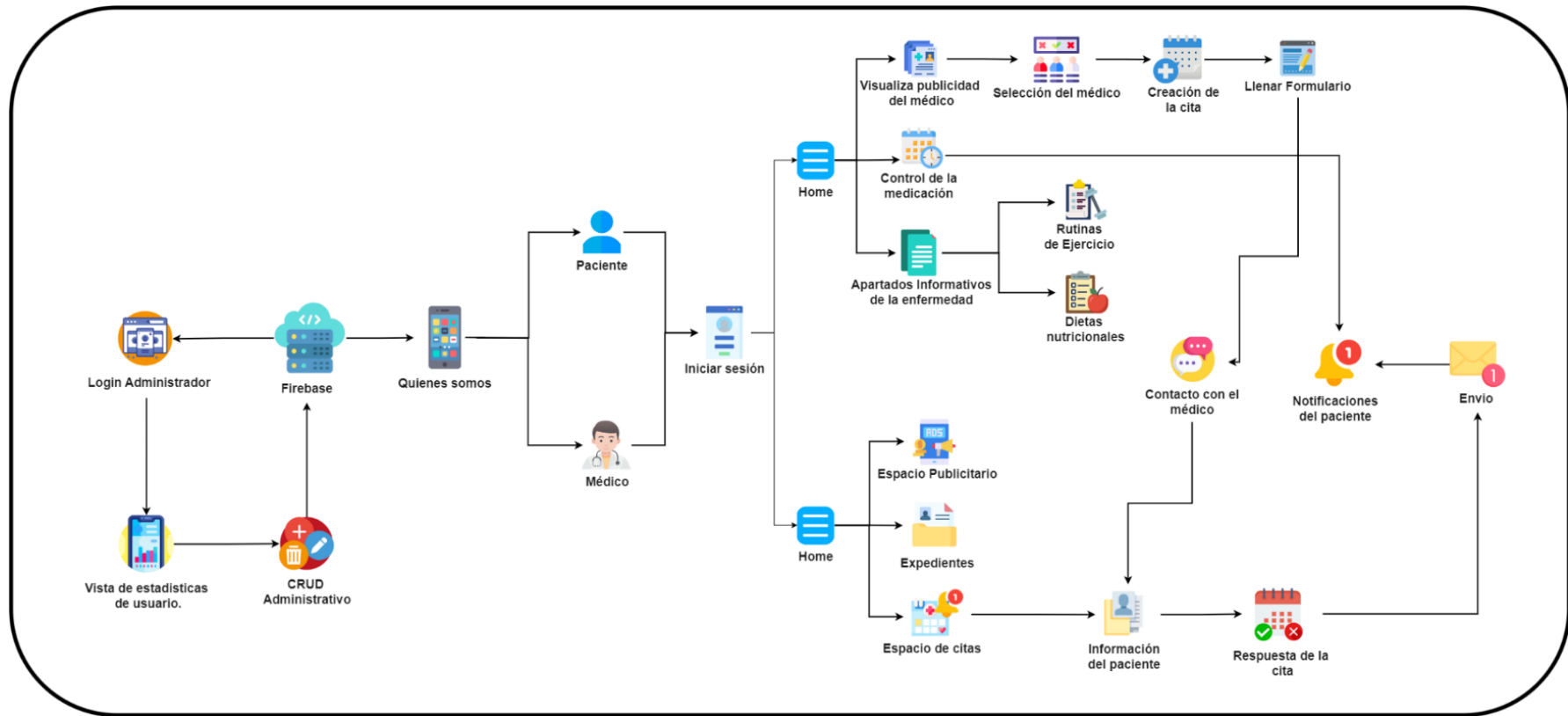
II. Diseño o Modelo

A continuación, se presenta de forma gráfica el funcionamiento de la aplicación web/móvil Control Health. Esta plataforma gestionará tres tipos de usuarios, cada uno con funciones específicas que reflejan las funciones principales de la aplicación. Tres tipos de usuarios: Pacientes acceden a módulos como Control de Medicación, Información de enfermedad, Nutrición, Ejercicio y Creación de citas; Administradores visualizan estadísticas y gestionan usuarios, con capacidad para agregar, editar, eliminar y actualizar perfiles; Médicos promocionan servicios y gestionan citas, aceptando o rechazando solicitudes en la aplicación.

2.1 Diseño de funcionamiento

La representación gráfica que sigue proporciona una visión clara del funcionamiento de la aplicación, detallando las opciones disponibles para los tres tipos de usuarios: Paciente, Médico y Administrador. En cuanto al usuario Paciente, se ilustra el proceso que le permitirá agregar medicamentos, así como acceder a los módulos de nutrición, ejercicio e información relevante sobre la enfermedad. Por otro lado, se describen las funciones del Administrador, que se centran en la gestión de los usuarios de la aplicación. Además, se presenta cómo el Médico puede promocionar sus servicios, acceder a los expedientes médicos y gestionar las solicitudes de citas médicas.

Figura 1

Diseño o Modelo

Nota: El diseño de Control Health asegura una experiencia fluida con una interfaz intuitiva y eficiente para la gestión de información de salud.

2.2 Diseño de navegación

A continuación, se presenta la estructura visual y organizativa de Control Health en la interfaz web y móvil, lo que facilita la comprensión de la navegación para los usuarios.

Figura 2

Mapa de Navegación web

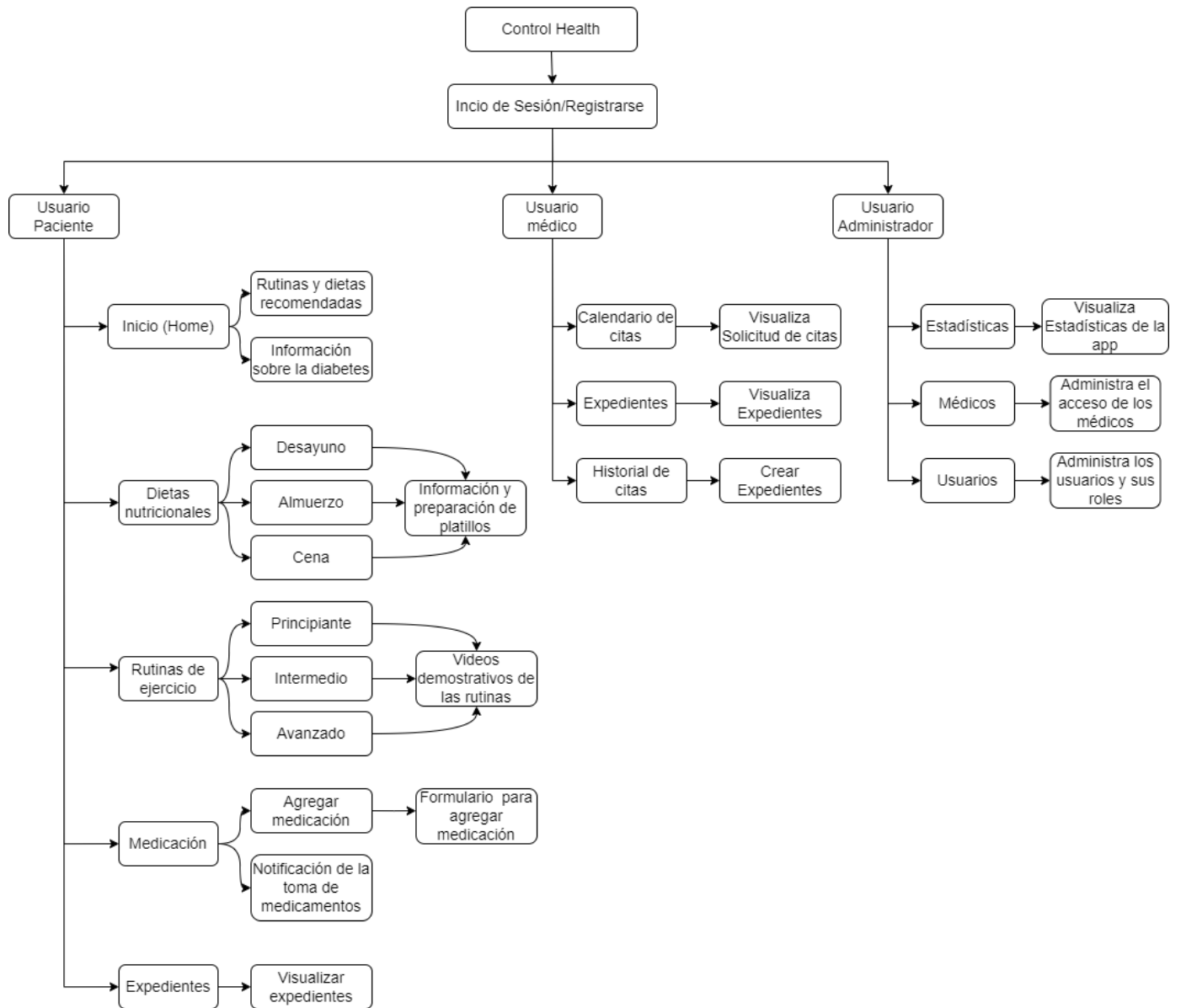
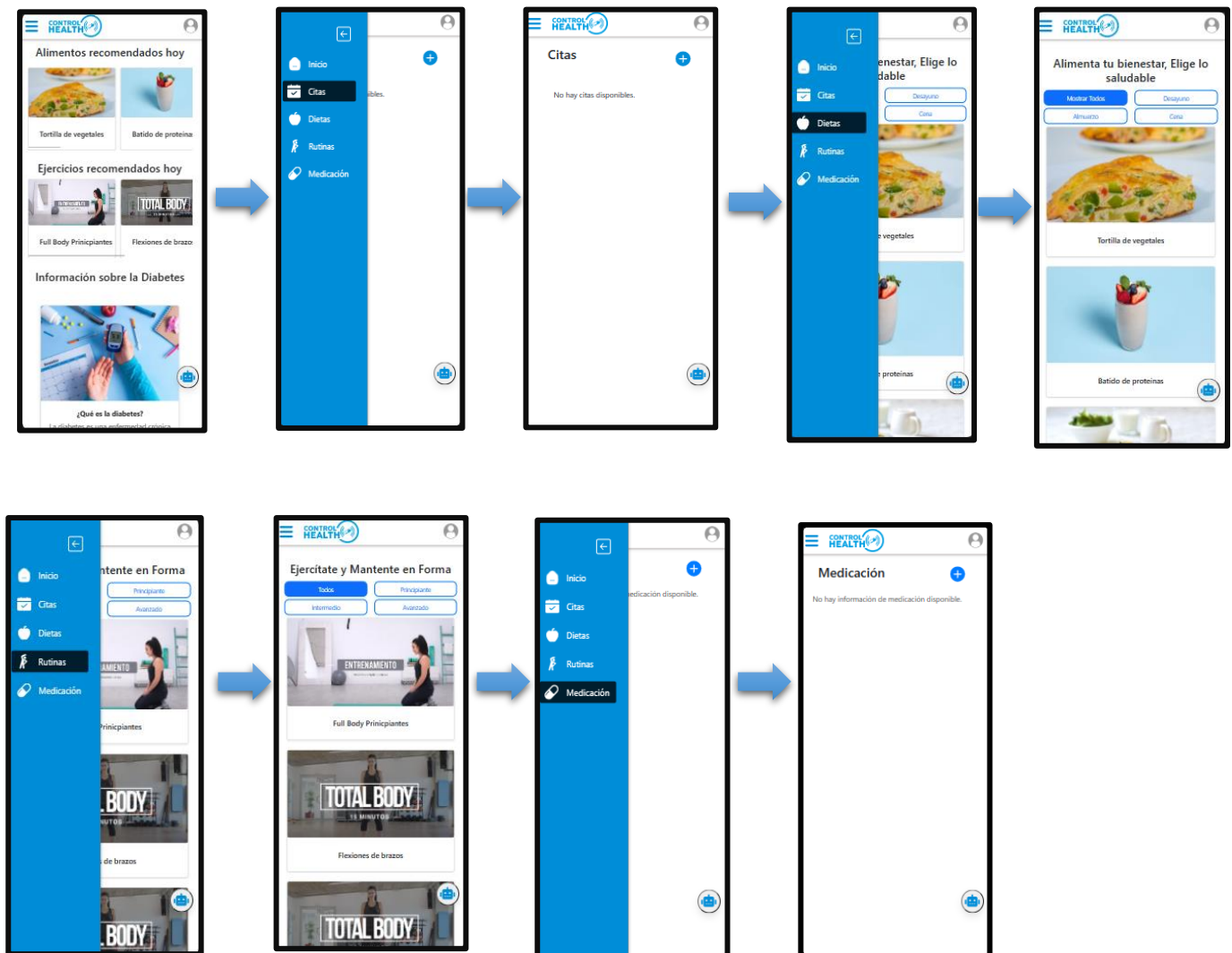
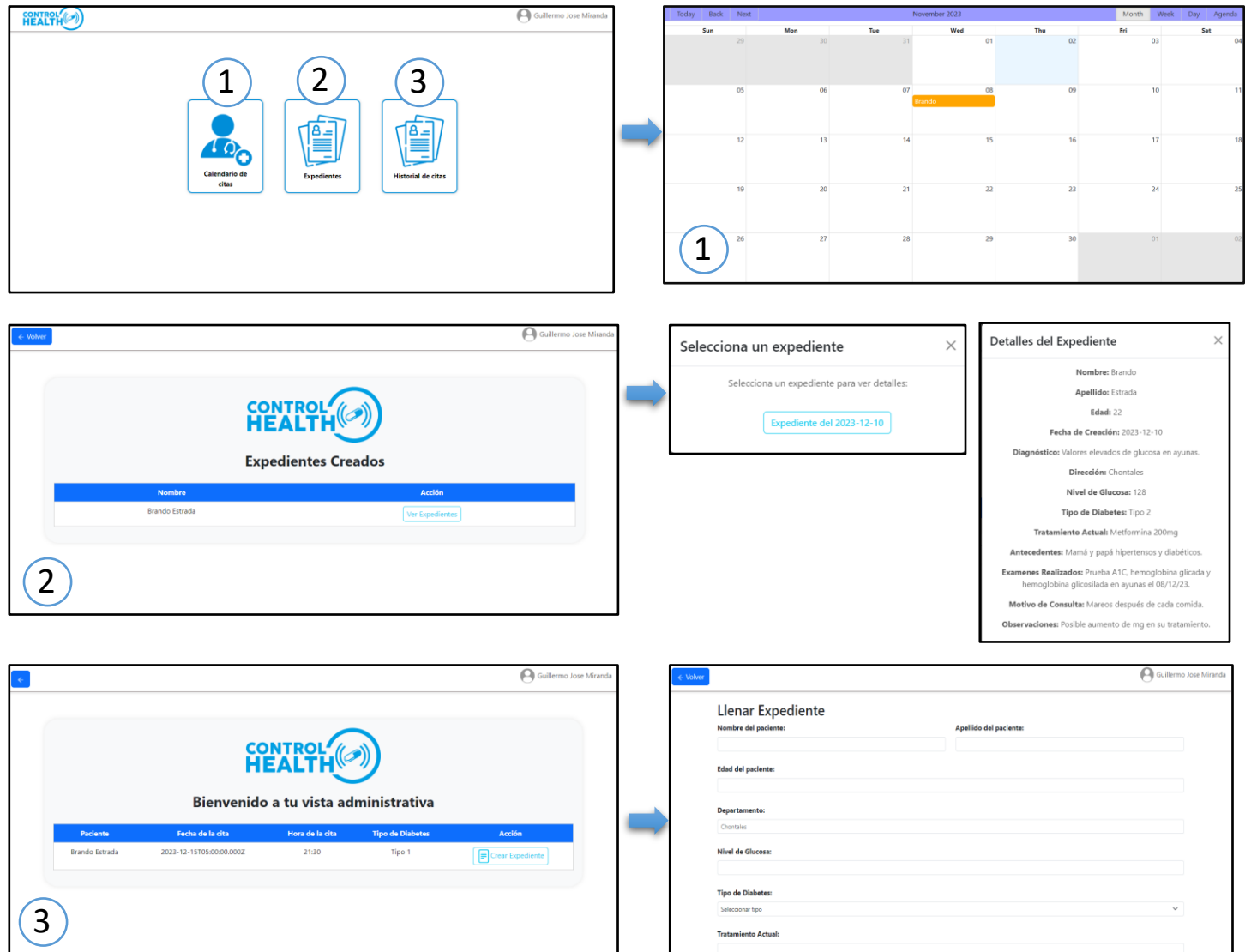


Figura 3*Diseño de Navegación del Paciente*

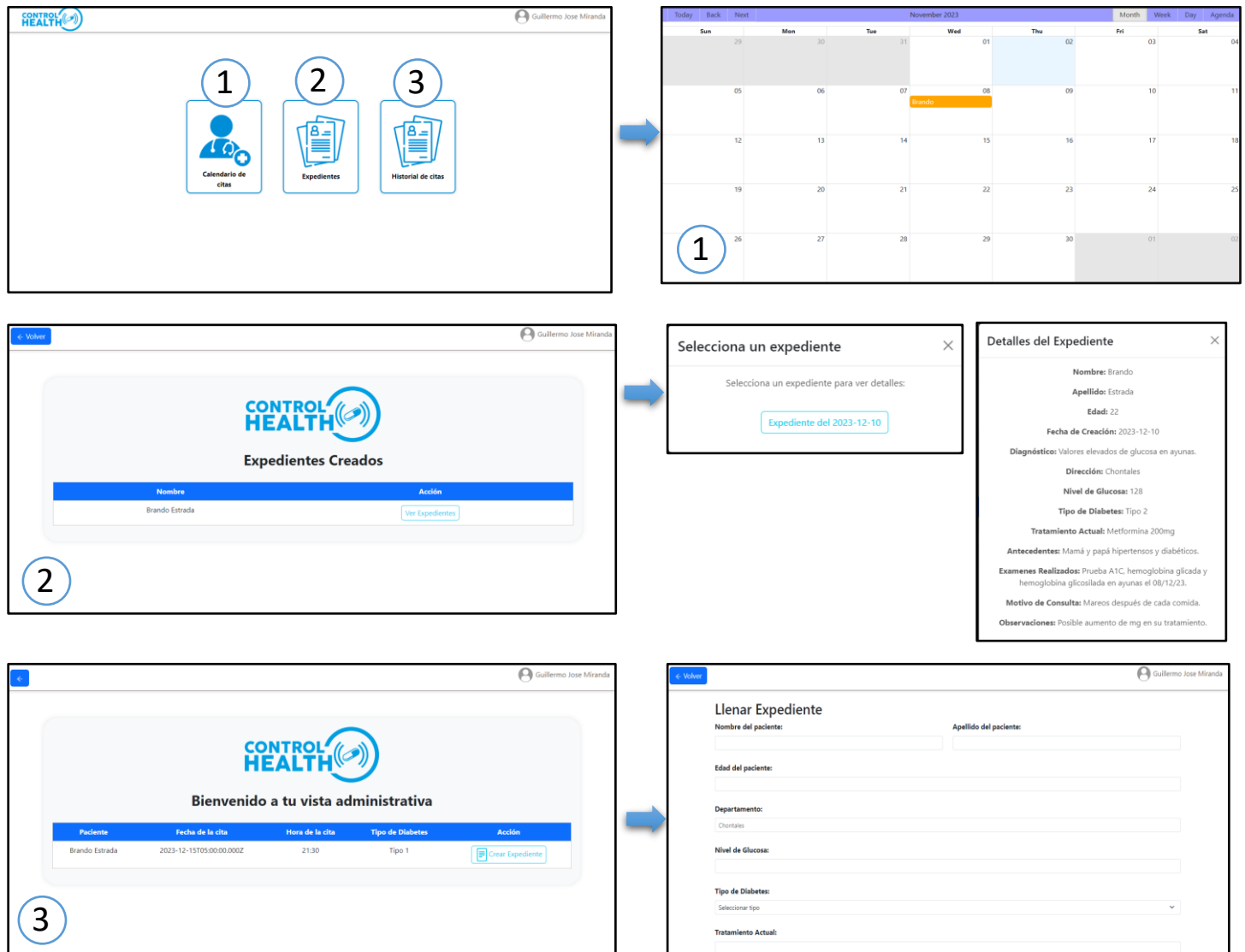
Nota: La representación visual del diseño de navegación en Control Health destaca su estructura para guiar a los usuarios con claridad y accesibilidad, buscando una experiencia intuitiva.

Figura 4

Diseño de Navegación del Médico

Nota: El diseño de navegación en Control Health está visualmente centrado en proporcionar una experiencia eficiente y específica para los profesionales médicos.

Figura 5

Diseño de Navegación del Administrador

Nota: Se presenta el diseño de navegación desde la Vista Administrativa, destacando la navegación entre sus diversas funcionalidades.

2.3 Diseño arquitectónico

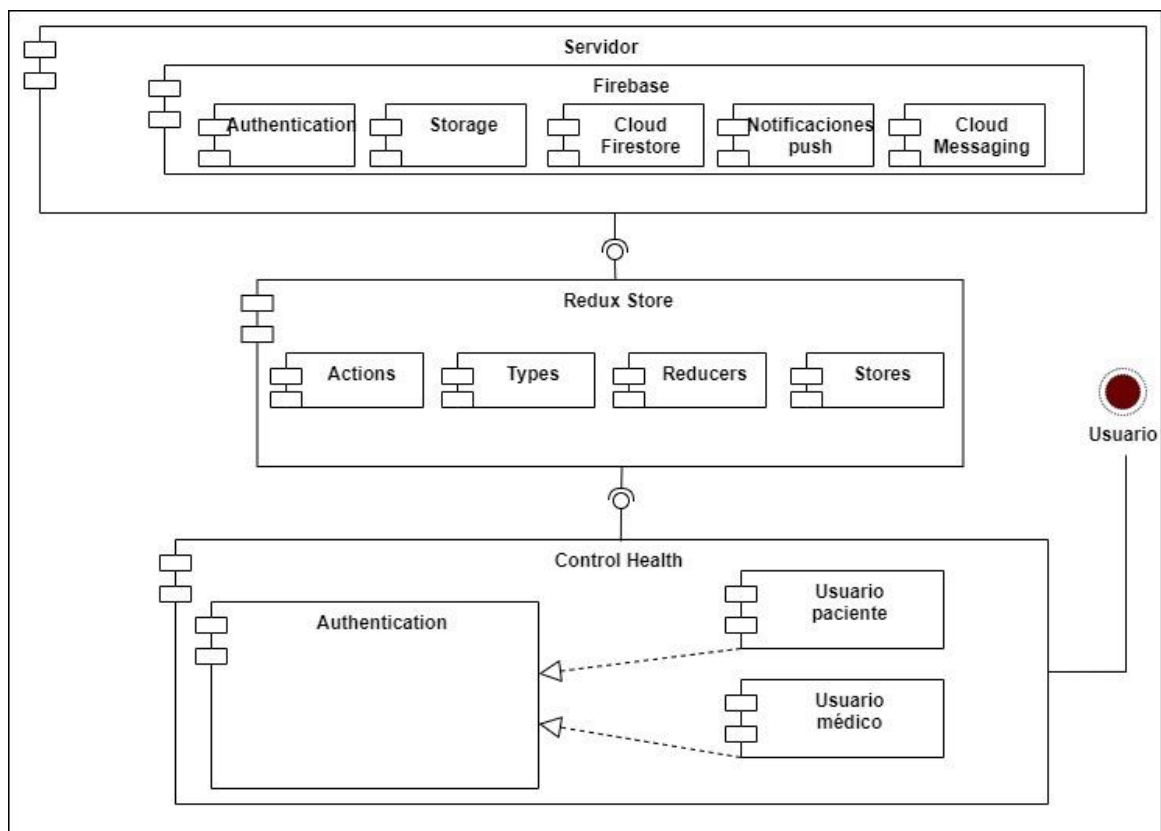
A continuación se muestran los diagramas categorizados en 3 niveles los cuales demuestran la arquitectura de la aplicación y el funcionamiento que esta tiene de una manera gráfica.

2.3.1 Diagrama de Componente Nivel 0

Este diagrama ilustra los principales componentes de la aplicación, los cuales se dividen en: el servidor proveniente de Firebase, el Store que almacena el estado global de la aplicación y la parte del Front-End donde se establecen los distintos tipos de usuarios para la interacción entre los usuarios pacientes y los usuarios médicos.

Figura 6

Diagrama de componente nivel 0



2.3.2 Diagrama de componente nivel 1

El diagrama muestra los componentes y las relaciones entre los diferentes módulos de la aplicación. Se definen los apartados de autenticación para pacientes y médicos, así como los siguientes módulos específicos:

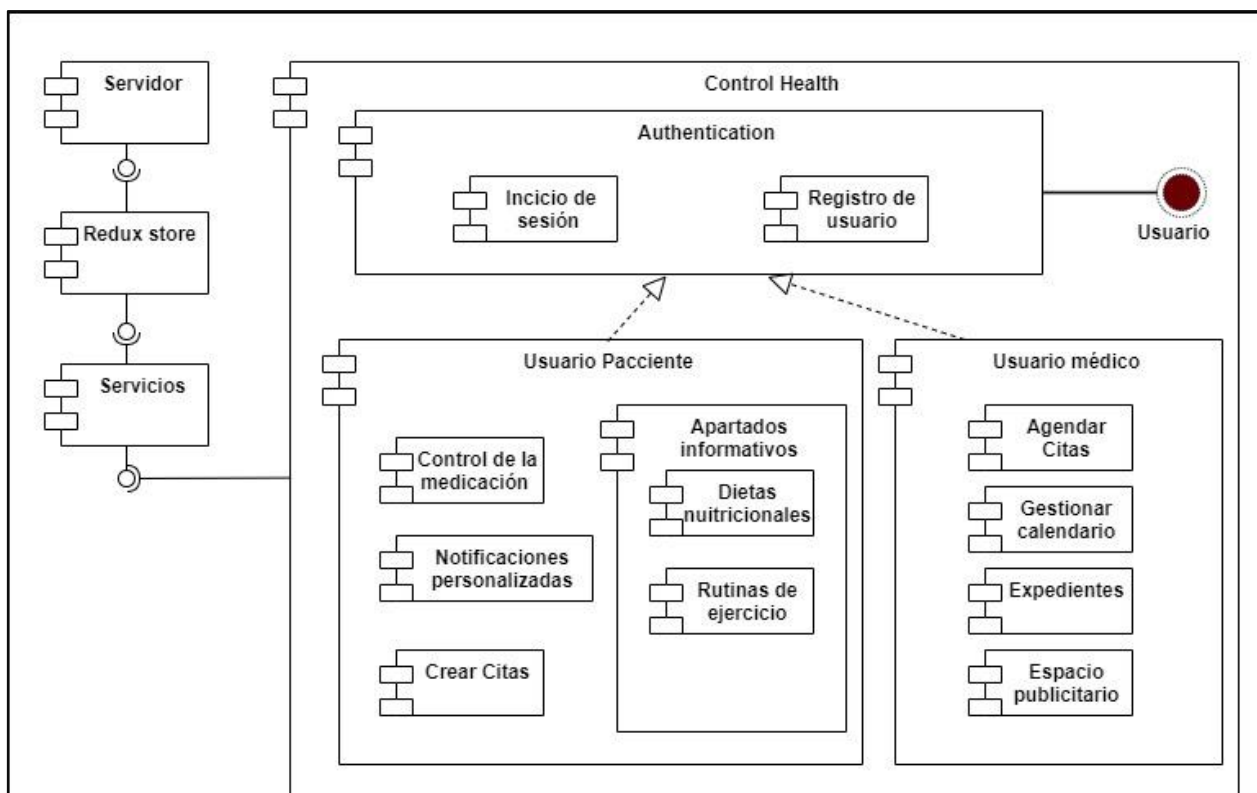
- Control de la medicación.
- Notificaciones personalizadas.
- Creación de citas.

Apartados informativos, que incluyen dietas nutricionales y rutinas de ejercicio. Estos apartados están disponibles para los usuarios pacientes. En el caso de los médicos, se incluyen los siguientes módulos:

- Agendar citas.
- Gestionar calendario.

Figura 7

Diagrama de componente nivel 1

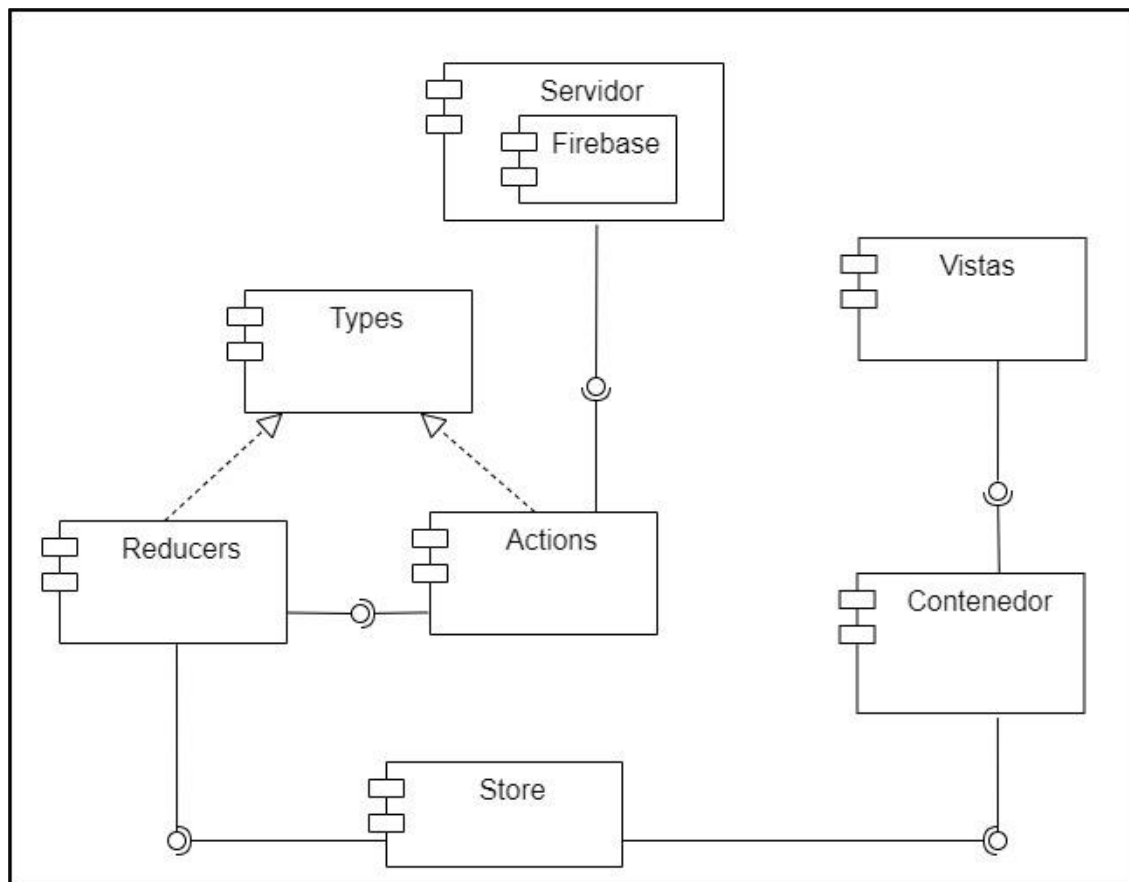


2.3.2 Diagrama de Componente Nivel 2

La arquitectura cliente-servidor es un modelo ampliamente utilizado en el desarrollo de sistemas y aplicaciones distribuidas. Se basa en la división de roles y responsabilidades entre dos componentes principales (cliente y servidor). En este modelo, los clientes solicitan y consumen los servicios proporcionados por los servidores, y los servidores procesan y responden a esas solicitudes. En esta ilustración se puede observar como el diagrama de componentes con Firebase y Redux demuestra cómo se establece una comunicación fluida entre el cliente y el servidor para gestionar el estado de la aplicación. Las acciones del usuario son procesadas en el servidor, donde se lleva a cabo la lógica de negocio. Posteriormente, los reductores en el cliente actualizan el estado de la aplicación, reflejando estos cambios en las vistas disponibles para el usuario.

Figura 8

Diagrama de componente nivel 2



III. Justificación De La Tecnología

3.1 Tecnología de desarrollo

Tabla de tecnologías para el desarrollo de la aplicación Control Health, elaborada en función de parámetros clave ofreciendo una base sólida en la elección de herramientas tecnológicas.

Tabla 1

Selección de la Tecnología de Desarrollo

Tecnología de Desarrollo	Python	Kotlin	React Js
Documentación	La documentación de Python es extensa y está respaldada por una sólida comunidad de desarrolladores. Tanto la documentación oficial como los recursos adicionales en línea, libros y la comunidad Python ofrecen una amplia gama de información y soporte para programadores de todos los niveles. (Amazon, 2023)	La documentación de Kotlin es amplia y abarca tanto los aspectos fundamentales como las características avanzadas del lenguaje. Tanto la documentación oficial como los recursos de la comunidad proporcionan una base sólida para aprender y aprovechar al máximo Kotlin en diferentes contextos de desarrollo, ya sea en el backend, el desarrollo de aplicaciones de escritorio o el desarrollo de aplicaciones móviles en Android.	ReactJS tiene una documentación exhaustiva y bien organizada que facilita la comprensión y el uso de la biblioteca. Además, la comunidad de React es activa y contribuye con tutoriales, ejemplos y recursos adicionales en línea. (Kinsta, 2022)
Escalabilidad	En términos de escalabilidad, Python ha demostrado ser efectivo en una amplia gama de aplicaciones y escenarios. Es utilizado en pequeños scripts y prototipos, así como en aplicaciones web y de gran envergadura. Python ofrece una gran cantidad de bibliotecas y frameworks que facilitan el desarrollo de	En términos de escalabilidad, Kotlin es muy adecuado para una amplia gama de aplicaciones. Al ser compatible con la plataforma Java y con acceso completo a las bibliotecas y frameworks de Java, Kotlin puede aprovechar la amplia gama de herramientas y recursos disponibles en el ecosistema Java. Esto	ReactJS es altamente escalable. La estructura basada en componentes permite dividir la aplicación en partes reutilizables, lo que facilita el manejo de proyectos de cualquier tamaño. Además, herramientas como Redux y Context API ayudan a gestionar el estado de manera eficiente en

aplicaciones escalables, como Django y Flask para desarrollo web. (Grupo Digital SDi, 2018)

incluye opciones de escalabilidad como el uso de contenedores Java, balanceo de carga, clustering y soluciones de alta disponibilidad. (AppMaster, 2023)

aplicaciones más grandes. (Zamudio, 2020)

Usabilidad

Es conocido por su alta usabilidad. Se destaca por su sintaxis clara y legible, que se asemeja al lenguaje humano y facilita la comprensión del código. Python se enorgullece de su filosofía "Zen de Python, que enfatiza la legibilidad y la simplicidad. Python ofrece una gran cantidad de bibliotecas y módulos estándar, lo que facilita el desarrollo rápido de aplicaciones y la resolución de problemas complejos con poco esfuerzo. La facilidad de uso de Python ha llevado a su adopción en una amplia gama de dominios, como desarrollo web, ciencia de datos, automatización de tareas y scripting. " (The Data Schools, 2023)

Kotlin se ha ganado una gran reputación por su usabilidad. Fue diseñado para ser conciso, expresivo y fácil de leer, lo que lo convierte en un lenguaje muy legible y amigable para los desarrolladores. Kotlin incorpora características modernas y poderosas que mejoran la productividad del desarrollador, como la inferencia de tipos, la seguridad nula, las extensiones de funciones y la programación orientada a objetos. Además, Kotlin es compatible con el entorno de desarrollo de Java y puede interoperar fácilmente con código Java existente, lo que facilita la transición de proyectos Java a Kotlin (Caronea, 2022)

ReactJS se enfoca en la creación de interfaces de usuario (UI) declarativas y componentes reutilizables, lo que simplifica el desarrollo y la mantenibilidad del código. También es compatible con JSX, lo que permite escribir componentes de manera más legible y similar a HTML. (Kinsta, 2022)

Costos

Python es un lenguaje de programación de código abierto y gratuito.

Kotlin es un lenguaje de programación de código abierto y gratuito.

ReactJS es de código abierto y gratuito, lo que reduce los costos de licencia. Además, su

eficiencia y escalabilidad pueden ayudar a reducir los costos de desarrollo y mantenimiento a largo plazo. (Emprendy, 2023)

Compatibilidad	Python no es nativamente compatible para el desarrollo móvil, pero existen enfoques alternativos y herramientas disponibles que permiten utilizar Python en el desarrollo de aplicaciones móviles a través de frameworks híbridos. (CompuHoy, 2020)	Desde 2017, Google ha respaldado oficialmente Kotlin como un lenguaje de programación de primera clase para el desarrollo de aplicaciones Android. (González, 2021)	ReactJS es compatible con una amplia gama de navegadores web y puede integrarse fácilmente con otras bibliotecas o frameworks. También es utilizado en aplicaciones móviles a través de React Native, lo que facilita la creación de aplicaciones multiplataforma. (React, 2023)
Curva de aprendizaje	50%	70%	10%
Valoración	1	3	5

3.2 Gestores de código

Selección de tecnologías para el gestor de código utilizado en la aplicación Control Health, fundamentada en parámetros clave.

Tabla 2

Selección del Gestor de Código

Gestores de código	Sublime Text	Atom	Visual Studio Code
Documentación	Sublime Text es conocido por su simplicidad y facilidad de uso, y su documentación refleja esto. Si bien la documentación oficial de Sublime Text no es extensa en términos de cantidad de contenido, proporciona una visión general de las características y funcionalidades principales del editor. Incluye información sobre la configuración, personalización, atajos de teclado, manejo de proyectos y complementos disponibles.	Atom, desarrollado por GitHub, cuenta con una documentación más extensa en comparación con Sublime Text. La documentación oficial de Atom abarca una amplia gama de temas, desde la instalación y configuración inicial hasta el uso avanzado del editor. Incluye información sobre la interfaz de usuario, el sistema de paquetes y temas, la personalización, la depuración y las características avanzadas del editor.	La documentación oficial de VS Code cubre todos los aspectos del editor, desde los conceptos básicos hasta las características avanzadas. Proporciona instrucciones detalladas sobre la configuración y personalización, la gestión de proyectos, el uso de extensiones, la depuración, el control de versiones y más. La documentación también incluye guías específicas para diferentes lenguajes de programación y frameworks populares, lo que ayuda a los desarrolladores a utilizar VS Code de manera efectiva en sus proyectos. Además, la comunidad de VS Code es muy activa y contribuye con una gran cantidad de recursos adicionales en línea, como tutoriales, videos y ejemplos de código.
Escalabilidad	Sublime Text es conocido por su rendimiento rápido y ligero, lo que lo hace adecuado para proyectos de diferentes tamaños. Su enfoque	Fue diseñado específicamente para ser altamente escalable y personalizable. Atom ofrece una amplia gama de funciones y características que se pueden extender	Desarrollado por Microsoft, también es conocido por su escalabilidad y capacidad de adaptación a diferentes proyectos y flujos de trabajo. VS Code ofrece una amplia gama de

minimalista y su capacidad para administrar múltiples archivos y proyectos de manera eficiente lo convierten en una opción escalable. Sin embargo, en términos de capacidad de extensión, Sublime Text puede ser limitado en comparación con Atom y Visual Studio Code.

mediante paquetes y temas. Esto hace que Atom sea altamente escalable en términos de funcionalidad y personalización, lo que lo convierte en una opción atractiva para proyectos de diversos tamaños y requisitos.

características y funcionalidades integradas, y también es altamente extensible a través de su sistema de extensiones. La comunidad de VS Code ha desarrollado una amplia variedad de extensiones que cubren diferentes lenguajes de programación, frameworks y herramientas. Esto proporciona una gran flexibilidad y escalabilidad a los desarrolladores para adaptar y ampliar VS Code según sus necesidades específicas.

Usabilidad

La usabilidad de Sublime Text se destaca por su simplicidad y capacidad para realizar tareas comunes de edición de texto de manera rápida y sin complicaciones. Sublime Text ofrece una navegación rápida, resaltado de sintaxis personalizable, atajos de teclado intuitivos y una respuesta ágil.

La usabilidad de Atom se basa en una interfaz de usuario moderna y amigable que es fácil de navegar y comprender. Atom ofrece una amplia gama de opciones de personalización y temas, lo que permite a los usuarios adaptar el editor a sus preferencias individuales. Además, Atom es altamente extensible mediante la instalación de paquetes que agregan funcionalidades adicionales.

La usabilidad de VS Code se basa en su interfaz de usuario intuitiva y bien diseñada, que proporciona una experiencia de edición de texto fluida. VS Code ofrece una amplia gama de características útiles, como el resaltado de sintaxis avanzado, la finalización de código, la depuración integrada y la integración con control de versiones. Además, VS Code es altamente extensible a través de su sistema de extensiones, lo que permite a los usuarios agregar funcionalidades adicionales según sea necesario.

Costos

Sublime Text se puede descargar y probar de forma gratuita, pero después de un período de evaluación, se espera

Atom es un editor de texto de código abierto y se puede descargar y utilizar de forma gratuita. No hay costos directos asociados con el uso de Atom en

Visual Studio Code (VS Code) es un editor de texto gratuito y de código abierto desarrollado por Microsoft. Puedes descargar y utilizar VS Code de forma gratuita sin ningún costo inicial. La

que los usuarios compren una licencia para su uso continuado. La licencia de Sublime Text tiene un costo único y permite el uso ilimitado en todas las plataformas compatibles (Windows, macOS y Linux).

términos de adquisición de licencias o derechos de uso. Sin embargo, dado que Atom es altamente personalizable y extensible, es posible que se encuentren complementos y temas pagos en el mercado que puedas optar por comprar según tus necesidades.

ventaja de utilizar VS Code es que cuenta con una amplia gama de extensiones y complementos gratuitos disponibles en Marketplace, que pueden mejorar aún más la funcionalidad del editor según tus necesidades específicas.

Compatibilidad

Sublime Text no está específicamente diseñado para el desarrollo móvil, pero es un editor de texto altamente personalizable y extensible. Puedes utilizar Sublime Text para escribir y editar el código fuente de tus aplicaciones móviles, pero es posible que necesites configurar e instalar complementos y herramientas adicionales para aprovechar características específicas del desarrollo móvil, como el resaltado de sintaxis para lenguajes móviles, la integración con herramientas de compilación o la depuración de aplicaciones móviles.

Atom ofrece una mayor compatibilidad con el desarrollo móvil en comparación con Sublime Text. Atom cuenta con una amplia gama de complementos y paquetes disponibles en el Marketplace, algunos de los cuales están diseñados específicamente para el desarrollo de aplicaciones móviles. Orientadas al desarrollo móvil.

Visual Studio Code (VS Code) es especialmente compatible con el desarrollo móvil, ya que ofrece una amplia gama de características y extensiones diseñadas específicamente para este propósito. VS Code cuenta con complementos y extensiones que admiten lenguajes y frameworks móviles populares, como React Native, Flutter y Xamarin. Estos complementos proporcionan resaltado de sintaxis avanzado, sugerencias de código, plantillas, emuladores y herramientas de depuración para facilitar el desarrollo de aplicaciones móviles.

Curva de aprendizaje

50%

60%

10%

Valoración	2	3	5
-------------------	---	---	---

3.3 Tecnología de diseño

Selección de tecnología de diseño en la aplicación Control Health, fundamentada en parámetros clave.

Tabla 3

Selección de la Tecnología de Diseño

Tecnologías de Diseño	Corel Draw	Adobe Photoshop	Adobe Illustrator
Documentación	CorelDRAW cuenta con una amplia documentación que abarca todos los aspectos del programa, desde los conceptos básicos hasta técnicas avanzadas de diseño. La documentación oficial de CorelDRAW incluye manuales de usuario, guías de inicio rápido, tutoriales paso a paso, videos instructivos y una extensa base de conocimientos en línea. Además, CorelDRAW tiene una comunidad de usuarios activa que comparte recursos, tutoriales y consejos adicionales. archivos y más. (Rondón, 2006)	Photoshop incluye guías de usuario, tutoriales, recursos en línea y una amplia gama de materiales de aprendizaje. Cubre todos los aspectos del programa, desde las herramientas y los ajustes básicos hasta técnicas avanzadas de retoque fotográfico, composición, manipulación de capas, uso de filtros, trabajos con texto y mucho más. (Adobe, 2023)	La documentación oficial de Adobe Illustrator incluye guías de usuario detalladas, tutoriales paso a paso, recursos en línea y una amplia variedad de materiales de aprendizaje. Cubre una amplia gama de temas, desde las herramientas y los conceptos básicos de diseño vectorial hasta técnicas avanzadas como la manipulación de trazados, el trabajo con capas, la creación de ilustraciones complejas, la tipografía y la exportación de archivos en diferentes formatos. Al igual que con Photoshop, Adobe proporciona una amplia comunidad en línea donde los usuarios pueden compartir conocimientos, trucos y recursos adicionales. (Adobe, 2023)
Escalabilidad	Es conocido por ser un programa de diseño gráfico altamente	En términos de escalabilidad, puede encontrar ciertas	Adobe Illustrator es conocido por su escalabilidad y

escalable. Puede manejar proyectos de diversos tamaños y complejidades, desde ilustraciones y logotipos simples hasta diseños complejos y proyectos de impresión de gran escala. CorelDRAW está diseñado para ofrecer un rendimiento óptimo incluso en proyectos más grandes, y su capacidad de trabajar con vectores y mapas de bits le brinda flexibilidad en la creación de diseños detallados y escalables. (CORELDRAW, 2021)

limitaciones. Aunque Photoshop es capaz de manejar proyectos de diferentes tamaños y niveles de complejidad, el rendimiento puede verse afectado en proyectos extremadamente grandes con múltiples capas y elementos. Sin embargo, Adobe ha implementado mejoras en cada versión para mejorar la escalabilidad y el rendimiento. (Adobe, 2023)

capacidad para trabajar con gráficos vectoriales en proyectos de diferentes tamaños. Como software basado en vectores, Illustrator puede escalar fácilmente diseños sin perder calidad y puede manejar proyectos complejos con múltiples elementos y capas. Illustrator tiene un rendimiento sólido y es adecuado para proyectos grandes, como ilustraciones detalladas, diseños de logotipos, maquetas de diseño y más. La herramienta de rendimiento de Illustrator permite ajustar las configuraciones para optimizar el rendimiento y la escalabilidad según las necesidades del proyecto. (Adobe, 2023)

Usabilidad

La usabilidad de CorelDRAW se destaca por su enfoque en el diseño gráfico y la creación de ilustraciones. La interfaz de usuario está diseñada para facilitar la creación de diseños, con herramientas y paneles organizados de manera lógica. CorelDRAW ofrece una amplia gama de funciones y herramientas de diseño, lo que permite a los usuarios crear ilustraciones, logotipos, gráficos y diseños de

La usabilidad de Photoshop se destaca por su potencia y versatilidad en el manejo de tareas relacionadas con la edición de imágenes. Aunque puede tener una curva de aprendizaje más pronunciada debido a su amplia gama de herramientas y funciones.

La usabilidad de Illustrator se destaca por su enfoque en la creación de gráficos vectoriales precisos y profesionales. La interfaz de usuario de Illustrator se asemeja a otros programas de Adobe, lo que facilita la transición para aquellos que ya están familiarizados con su suite de productos. Illustrator ofrece herramientas y paneles intuitivos para crear y editar formas, trazados, ilustraciones y diseños

manera eficiente.
(CorelDraw, 2023)

complejos. También cuenta con opciones de personalización y configuración para adaptarse a las preferencias de los usuarios. La capacidad de trabajar con vectores permite una mayor flexibilidad en el diseño y la escalabilidad de los proyectos. (Alba, 2019)

Costos

CorelDRAW ofrece diferentes opciones de licencia y suscripción. Puedes adquirir una licencia perpetua para CorelDRAW Graphics Suite, que es una compra única con un costo que varía según la versión y las opciones adicionales incluidas. Además, CorelDRAW también ofrece opciones de suscripción mensual o anual a través de su modelo de CorelDRAW Subscription.

Adobe Photoshop se ofrece a través de Adobe Creative Cloud, que es un servicio de suscripción mensual o anual. Adobe Creative Cloud incluye acceso a Photoshop y a otros programas de Adobe, como Illustrator, InDesign y más. Los planes de suscripción de Creative Cloud varían según las necesidades y pueden incluir opciones para individuos, estudiantes, equipos y empresas.

Al igual que Photoshop, Adobe Illustrator también está disponible a través de Adobe Creative Cloud. Los planes de suscripción de Creative Cloud incluyen acceso a Illustrator y otros programas de Adobe. Los costos de los planes de Creative Cloud varían según las opciones de suscripción y los requisitos específicos.

Compatibilidad

CorelDRAW es un software de diseño gráfico vectorial que cuenta con una amplia gama de herramientas y funcionalidades para la creación de logotipos. CorelDRAW es compatible con formatos de archivo vectorial como AI (Adobe Illustrator), EPS y SVG, lo que permite importar y exportar logotipos de manera fluida entre

Aunque Photoshop no es un software de diseño vectorial en sí mismo, es compatible con la creación de logotipos utilizando elementos y técnicas de diseño gráfico. Con Photoshop, se puede trabajar con imágenes y elementos rasterizados para crear logotipos que combinen elementos gráficos y texto.

Con Illustrator, se pueden crear logotipos de forma precisa y escalable utilizando herramientas de dibujo vectorial, manipulación de formas, opciones de texto, efectos y mucho más. Illustrator ofrece una amplia gama de características y funcionalidades diseñadas específicamente para la creación de logotipos,

	diferentes programas de diseño.		como la capacidad de crear trazados limpios, utilizar colores Pantone y trabajar con formatos de exportación específicos para logotipos.
Curva de aprendizaje	90%	50%	20%
Valoración	1	3	5

3.4 Tecnologías para prototipado

Selección de tecnología para prototipado en la aplicación Control Health, fundamentada en parámetros clave.

Tabla 4

Selección de la Tecnología para Prototipado

Tecnologías para prototipado	InVision	Figma	Adobe XD
Documentación	La documentación oficial de InVision incluye guías de inicio rápido, tutoriales paso a paso, documentación detallada sobre el uso de sus funciones y herramientas, y una base de conocimientos en línea. También ofrecen cursos y recursos de aprendizaje en línea adicionales. La documentación de InVision cubre una amplia gama de temas, desde la creación de prototipos interactivos y el diseño de interfaces de usuario hasta la colaboración en equipo y la integración con otras herramientas y servicios.	La documentación oficial de Figma incluye guías detalladas sobre la interfaz de usuario, las funciones y herramientas clave, la colaboración en equipo y la creación de prototipos. También ofrecen tutoriales y recursos adicionales para ayudar a los usuarios a sacar el máximo provecho de la plataforma. Figma cuenta con una comunidad activa de usuarios que comparten recursos y consejos en línea, lo que amplía aún más la información disponible para los usuarios.	La documentación oficial de Adobe XD incluye guías de usuario, tutoriales, documentación detallada sobre las funciones y características del programa, y una base de conocimientos en línea. También ofrecen recursos de aprendizaje adicionales, como cursos en línea y videos instructivos. Adobe XD se integra con el ecosistema de Adobe Creative Cloud, lo que permite a los usuarios acceder a recursos y tutoriales adicionales disponibles para otros productos de Adobe. (Adobe, 2023)
Escalabilidad	Permite la creación de prototipos interactivos y el diseño de interfaces de usuario para proyectos de diferentes tamaños y complejidades. InVision está diseñado para facilitar la colaboración en equipo, lo que permite a los miembros del equipo	Figma permite a los equipos de diseño colaborar en tiempo real y trabajar en proyectos de diferentes tamaños sin problemas. La funcionalidad basada en la nube de Figma permite a los miembros del equipo acceder y editar diseños	Ofrece escalabilidad en términos de diseño y colaboración en equipos de diferentes tamaños. Adobe XD permite crear diseños y prototipos interactivos para proyectos pequeños y grandes. También ofrece

trabajar juntos en proyectos simultáneamente. Además, InVision proporciona características adicionales para la gestión de proyectos, como la capacidad de organizar y compartir activos de diseño, comentar y revisar diseños, y realizar pruebas de usabilidad. Estas características hacen que InVision sea una opción escalable para proyectos de diseño colaborativo.

desde cualquier ubicación, lo que facilita la colaboración remota. Figma también ofrece funciones de gestión de componentes y estilos para garantizar la consistencia en los diseños y facilitar la escalabilidad en proyectos grandes.

características de colaboración que permiten a los equipos trabajar juntos en proyectos de forma simultánea. Además, Adobe XD se integra con el ecosistema de Adobe Creative Cloud, lo que permite a los usuarios aprovechar otros recursos y herramientas de Adobe para una mayor escalabilidad en sus flujos de trabajo.

Usabilidad

InVision ofrece una amplia gama de características y herramientas que facilitan la creación y presentación de diseños, como la capacidad de compartir y comentar prototipos, realizar pruebas de usabilidad y colaborar en tiempo real con otros miembros del equipo. La usabilidad de InVision se ve reforzada por su enfoque en la colaboración, lo que facilita la comunicación y la iteración entre diseñadores, desarrolladores y stakeholders.

Figma ofrece características avanzadas como el diseño de componentes reutilizables y la posibilidad de trabajar en varios proyectos simultáneamente. La facilidad de uso y la colaboración en tiempo real hacen que Figma sea una opción popular para equipos de diseño distribuidos.

Adobe XD se destaca por su familiaridad y su integración con el ecosistema de Adobe Creative Cloud. La interfaz de usuario de Adobe XD es intuitiva y sigue los estándares de diseño de otros productos de Adobe. Adobe XD ofrece características y herramientas potentes para el diseño de interfaces de usuario y la creación de prototipos interactivos. Además, Adobe XD permite una integración fluida con otros productos de Adobe, como Photoshop e Illustrator, lo que facilita el flujo de trabajo para aquellos que ya están familiarizados con el

			ecosistema de Adobe. (Adobe, 2023)
Costos	InVision ofrece diferentes opciones de planes de suscripción para adaptarse a las necesidades de los usuarios. Tienen planes tanto para individuos como para equipos. Los precios varían según el plan seleccionado y la cantidad de usuarios que se incluyan	Figma también ofrece diferentes planes de suscripción para individuos y equipos. Tienen un plan gratuito que permite a los usuarios crear y colaborar en proyectos básicos. Además, tienen planes de pago que ofrecen características y funcionalidades adicionales, como la colaboración en tiempo real y la administración de equipos. Los precios varían según el plan seleccionado y la cantidad de usuarios.	Adobe XD se ofrece a través de Adobe Creative Cloud, que es un servicio de suscripción mensual o anual que incluye acceso a varias aplicaciones de Adobe, incluyendo Adobe XD. Los planes de suscripción de Creative Cloud varían según las necesidades y pueden incluir opciones para individuos, estudiantes, equipos y empresas. Los precios de los planes de Creative Cloud pueden variar según la región y los requisitos específicos.
Compatibilidad	InVision permite importar y trabajar con archivos vectoriales en sus proyectos, lo que te permite crear y diseñar elementos vectoriales para tus prototipos móviles. Además, InVision ofrece funcionalidades de prototipado específicas para dispositivos móviles, como la capacidad de simular gestos táctiles, transiciones de pantalla y animaciones.	Figma es una herramienta que permite crear y editar gráficos vectoriales directamente en la herramienta, lo que facilita la creación de elementos vectoriales para prototipos móviles. Figma ofrece funcionalidades de diseño y prototipado específicas para dispositivos móviles, como la capacidad de diseñar interfaces de usuario responsivas y crear prototipos con transiciones y animaciones adaptadas a dispositivos móviles.	Adobe XD también ofrece compatibilidad con la creación de vectores y prototipos móviles. Puedes crear gráficos vectoriales directamente en la herramienta o importar archivos vectoriales desde otras aplicaciones de Adobe. Adobe XD proporciona funcionalidades específicas para dispositivos móviles, como la capacidad de diseñar interfaces de usuario adaptadas a diferentes tamaños de pantalla y la creación de prototipos interactivos

con transiciones y animaciones específicas para aplicaciones móviles.

Curva de aprendizaje	60%	40%	20%
Valoración	1	3	5

3.5 Base de Datos

Selección de base de datos en la aplicación Control Health, fundamentada en parámetros clave.

Tabla 5

Selección de la Base de Datos

Bases de datos	SQLite	Mongo DB	Firebase
Documentación	<p>La documentación oficial, hay numerosos recursos en línea, tutoriales y libros disponibles que cubren temas específicos y ofrecen ejemplos de uso de SQLite en diferentes lenguajes de programación.</p> <p>(SQLite, 2023)</p>	<p>MongoDB cuenta con una documentación extensa y completa disponible en su sitio web oficial. La documentación incluye guías de instalación, tutoriales, referencia de comandos, ejemplos de código y preguntas frecuentes. Además, MongoDB ofrece recursos adicionales como webinars, blogs y una comunidad activa donde los usuarios pueden intercambiar conocimientos y obtener ayuda.</p> <p>(Mongo DB, 2023)</p>	<p>La documentación incluye guías de inicio rápido, tutoriales detallados, referencias de API, ejemplos de código y preguntas frecuentes. Firebase también ofrece soporte a través de su comunidad en línea, donde los usuarios pueden hacer preguntas y obtener respuestas de expertos.</p> <p>(Firebase, 2023)</p>
Escalabilidad	<p>Es adecuada para proyectos más pequeños y aplicaciones con requisitos de transacciones y concurrencia moderados. No se recomienda para aplicaciones que necesiten escalar horizontalmente en múltiples servidores, ya que funciona como una biblioteca embebida en lugar de un servidor independiente. Sin embargo, SQLite es</p>	<p>MongoDB es altamente escalable y está diseñado para gestionar grandes volúmenes de datos y cargas de trabajo intensivas. Ofrece capacidades de escalabilidad horizontal mediante la distribución de datos en múltiples servidores, lo que permite aumentar la capacidad de almacenamiento y el rendimiento de manera flexible.</p>	<p>Firebase está diseñado para escalar automáticamente según la demanda de la aplicación. Puede manejar cargas de trabajo variables y picos de tráfico sin requerir una configuración manual adicional. Firebase se ejecuta en la infraestructura de Google Cloud Platform, lo que brinda escalabilidad y rendimiento confiables. Además, Firebase ofrece opciones para expandir la</p>

escalable en términos de capacidad de almacenamiento y rendimiento dentro de un solo dispositivo o servidor.

capacidad de almacenamiento y el rendimiento a medida que la aplicación crece.

Usabilidad

Su usabilidad radica en su simplicidad y ligereza, lo que facilita su implementación y administración. SQLite es ideal para aplicaciones móviles que requieren un almacenamiento local y un acceso rápido a los datos, especialmente en casos en los que la aplicación no necesita una gran cantidad de datos o una escalabilidad significativa. SQLite es fácil de usar y ofrece un rendimiento rápido para operaciones de lectura y escritura en un entorno de dispositivo móvil. (Avantgeo, 2023)

Su usabilidad radica en su capacidad para manejar grandes volúmenes de datos no estructurados y adaptarse a los cambios en el esquema de datos de la aplicación. MongoDB es una opción adecuada para aplicaciones móviles que requieren almacenar y gestionar datos complejos o cambiantes. Su modelo de documento facilita la integración de datos y permite una mayor agilidad en el desarrollo de aplicaciones.

(MongoDB, 2023)

Su usabilidad radica en su facilidad de integración y su conjunto de características completas. Firebase proporciona una API intuitiva y herramientas de desarrollo que simplifican la implementación de funciones comunes en aplicaciones móviles, como la sincronización en tiempo real y la autenticación de usuarios. Su enfoque en la nube facilita la escalabilidad y el acceso a los datos desde diferentes dispositivos. (Firebase, 2023)

Costos

SQLite es una base de datos de código abierto y gratuita que no tiene costos de licencia asociados.

La versión de comunidad de MongoDB es adecuada para proyectos más pequeños o aplicaciones no comerciales, y no tiene costos de licencia. Sin embargo, la versión empresarial de MongoDB tiene costos asociados, y los precios varían según el tamaño del clúster, el nivel de soporte y otros servicios adicionales. (MongoDB, 2023)

Firebase ofrece un modelo de precios basado en el uso y tiene diferentes planes disponibles. Firebase ofrece un plan gratuito con ciertos límites de uso, como almacenamiento, ancho de banda y número de usuarios simultáneos. También ofrecen planes de pago que varían según los recursos y las características incluidas, como almacenamiento adicional, notificaciones push, funciones

avanzadas de análisis y más. Los precios de Firebase están sujetos a cambios.
(Firebase, 2023)

Compatibilidad	SQLite es compatible con las aplicaciones móviles y se utiliza comúnmente como una base de datos local en dispositivos móviles. Es compatible con una variedad de plataformas móviles, como Android, iOS y Windows Mobile.	MongoDB es compatible con las aplicaciones móviles a través de sus controladores y APIs disponibles para diversas plataformas móviles. MongoDB cuenta con controladores y bibliotecas oficiales para lenguajes de programación móvil populares, como Java (para Android) y Swift/Objective-C (para iOS).	Firestore está diseñado específicamente para el desarrollo de aplicaciones móviles y es altamente compatible con plataformas móviles como Android y iOS. Firestore ofrece una amplia gama de servicios y características, como bases de datos en tiempo real, autenticación de usuarios, notificaciones push, almacenamiento en la nube y más, todos ellos diseñados para ser utilizados directamente en aplicaciones móviles
Curva de aprendizaje	90%	60%	20%
Valoración	1	3	5

IV. Justificación de la Metodología

Selección de metodología ágil para implementar durante el desarrollo de la aplicación Control Health, fundamentada en parámetros clave.

Tabla 6

Justificación de la Metodología

Metodologías ágiles	Design Sprint (Metodología de Google)	Kanban	SCRUM
Documentación	<p>El libro "Sprint: Cómo resolver grandes problemas y testear nuevas ideas en solo cinco días" escrito por Jake Knapp, John Zeratsky y Braden Kowitz, quienes fueron miembros del equipo de diseño de Google Ventures. El libro proporciona una descripción detallada de la metodología y ejemplos prácticos de cómo aplicarla.</p> <p>El sitio web oficial de Google Design Sprint que ofrece una introducción a la metodología, guías paso a paso, recursos adicionales y ejemplos de casos de estudio.</p> <p>Hay numerosos artículos, blogs y videos en línea que exploran la Design Sprint y brindan información detallada sobre su aplicación y beneficios.</p>	<p>El libro "Kanban: Successful Evolutionary Change for Your Technology Business" escrito por David J. Anderson, quien es considerado uno de los principales expertos en Kanban. El libro ofrece una guía completa sobre cómo implementar Kanban y sus principios fundamentales.</p> <p>Sitios web especializados, blogs y comunidades en línea donde los profesionales de Kanban comparten sus experiencias, consejos y mejores prácticas.</p> <p>Documentación oficial de herramientas de gestión de proyectos y colaboración que ofrecen características de Kanban, como Trello y Jira. Estos recursos proporcionan información sobre cómo utilizar Kanban en un entorno digital.</p>	<p>La "Guía SCRUM" escrita por Ken Schwaber y Jeff Sutherland, los cocreadores de SCRUM. La guía proporciona una descripción completa de la metodología, los roles, los artefactos y los eventos de SCRUM.</p> <p>Organizaciones y certificaciones reconocidas en SCRUM, como SCRUM Alliance y SCRUM.org, que ofrecen recursos de aprendizaje, cursos y exámenes para obtener certificaciones SCRUM.</p> <p>Sitios web especializados, foros y comunidades en línea donde profesionales y expertos en SCRUM comparten información, consejos y casos de estudio.</p>
Escalabilidad	La Design Sprint es una metodología que se enfoca en abordar desafíos de	Kanban es una metodología ágil de gestión visual del flujo de	SCRUM es una metodología ágil para la gestión de proyectos

diseño y desarrollo de productos de manera rápida y efectiva. Aunque la Design Sprint se basa en un proceso de cinco días, se puede adaptar y escalar según las necesidades del proyecto y el equipo. Puede aplicarse a proyectos pequeños o grandes, y es flexible en términos de personalización y adaptación a diferentes contextos. Si bien la metodología original está diseñada para un equipo pequeño y multidisciplinario, se pueden realizar ajustes para involucrar a equipos más grandes y gestionar proyectos más complejos.

trabajo que permite una escalabilidad natural. Se basa en la visualización de tareas y la limitación del trabajo en progreso (WIP) para optimizar la eficiencia y la productividad. Kanban se puede aplicar tanto a proyectos pequeños como a proyectos a gran escala, y se adapta a medida que los equipos y los requisitos cambian. Los tableros Kanban pueden expandirse y personalizarse según las necesidades del proyecto y el equipo, lo que facilita la escalabilidad.

que se centra en la colaboración, la adaptabilidad y la entrega incremental de resultados. SCRUM es altamente escalable y se puede aplicar a proyectos de diferentes tamaños y complejidades. La metodología se basa en equipos autónomos y multifuncionales que trabajan en eSprints, ciclos de desarrollo iterativos y enfocados. Los equipos SCRUM pueden escalar utilizando enfoques como SCRUM de SCRUMs o SCRUM a nivel de programa, que permiten coordinar múltiples equipos y gestionar proyectos más grandes.

Usabilidad

La usabilidad de la Design Sprint radica en su enfoque claro y estructurado, que guía a los equipos a través de pasos específicos para lograr resultados rápidos y obtener retroalimentación temprana de los usuarios. La Design Sprint es particularmente efectiva para proyectos que requieren una toma de decisiones rápida y una validación temprana de conceptos.

La usabilidad de Kanban radica en su enfoque simple y visual, que facilita la comprensión y seguimiento del progreso del trabajo. Los tableros Kanban permiten a los equipos tener una visibilidad clara de las tareas y los estados de trabajo, lo que ayuda a optimizar la colaboración y la gestión del flujo de trabajo. Kanban es efectivo para equipos que buscan mejorar la eficiencia y la productividad al

La usabilidad de SCRUM radica en su enfoque estructurado y colaborativo, que permite a los equipos entregar valor de manera incremental y adaptativa. SCRUM se basa en roles claros, eventos regulares y artefactos definidos, lo que facilita la planificación, el seguimiento y la colaboración efectiva. SCRUM es efectivo para proyectos complejos y cambiantes, ya que

		visualizar y limitar la carga de trabajo.	fomenta la transparencia, la inspección y la adaptación continua.
Costos	No tienen costos inherentes, ya que son marcos de trabajo o enfoques utilizados para gestionar proyectos y procesos.		
Compatibilidad	Puede ser utilizada para acelerar el proceso de diseño y desarrollo de una aplicación web responsiva a móvil en un corto período de tiempo. Durante una semana, se pueden realizar actividades como la identificación de problemas, la generación de ideas, la creación de prototipos y las pruebas de usuario para iterar rápidamente y obtener resultados concretos. La Design Sprint es especialmente útil al inicio de un proyecto para alinear al equipo, definir los objetivos y validar conceptos antes de pasar a una fase de desarrollo más amplia.	Con Kanban, puedes dividir el trabajo en tarjetas o elementos individuales y visualizar su progreso en un tablero Kanban. Puedes establecer columnas que representen las etapas del flujo de trabajo, como "Por hacer", "En progreso" y "Terminado". A medida que avanzas en el desarrollo de la aplicación web responsiva a móvil, puedes mover las tarjetas a través de las columnas para reflejar su estado actual. Kanban te permite tener una visibilidad clara del trabajo en curso, priorizar las tareas y adaptarte a medida que surjan nuevos requisitos o cambios.	Puede ser utilizado para desarrollar una aplicación web responsiva a móvil en un semestre dividiendo el proyecto en Sprints de duración fija, generalmente de 1 a 4 semanas. Cada Sprint se inicia con una planificación en la que se seleccionan las tareas a realizar y se establece un objetivo claro. Durante el Sprint, el equipo trabaja en colaboración para completar las tareas y lograr el objetivo establecido. Al final de cada Sprint, se realiza una revisión para demostrar el progreso y recopilar retroalimentación. SCRUM te permite realizar ajustes continuos a medida que avanzas en el desarrollo de la aplicación web responsiva a móvil y priorizar las funcionalidades más importantes para su entrega.
Curva de aprendizaje	80%	50%	10%
Valoración	2	3	5

V. Oportunidad de Mercado

En el año 2023, el mundo ha experimentado una revolución en la atención médica gracias a la pandemia de COVID-19, lo cual ha acelerado el desarrollo de la telemedicina. Este avance ha mejorado significativamente la eficiencia, accesibilidad y seguridad médica a nivel global.

En Nicaragua, la diabetes es una enfermedad crónica que afecta a una considerable porción de la población. Según el Censo de crónicos y sistema de atenciones ambulatorias, se estima que aproximadamente el 10% de la población nicaragüense padece diabetes, lo que equivale a alrededor de 700,000 personas (MINSA, 2022). La falta de control diario en la medicación precisa según su condición ha agravado esta situación, lo que ha llevado al alarmante aumento de las muertes relacionadas con la diabetes en el país.

Ante esta situación, se determina la necesidad de desarrollar una herramienta integral que ofrezca servicios médicos especializados para el control y gestión de la diabetes. Con este fin, se ha creado un software específico que utiliza tecnologías de calidad, garantizando escalabilidad y eficiencia. Este sistema integral y personalizado permitirá a las personas que viven con diabetes tener un mejor control de su enfermedad, proporcionando información precisa y en tiempo real, facilitando la toma de decisiones y mejorando su calidad de vida.

Para asegurar la fiabilidad de esta plataforma de salud, se ha realizado una exhaustiva encuesta a médicos diabetólogos y a 50 personas con Diabetes tipo 1 y 2. También se han recibido muy buenas opiniones por parte de entrenadores físicos y nutricionistas, quienes forman parte de la aplicación. Esto garantiza que la información y el contenido exclusivo, como videos de ejercicios y recetas diseñadas especialmente para personas diabéticas, sean confiables y de alta calidad.

Sin embargo, es importante tener en cuenta que existen riesgos potenciales al implementar esta solución, como desafíos legales, financieros y técnicos, que deben ser abordados adecuadamente para garantizar su éxito a largo plazo. A pesar de estos desafíos, esta innovación no solo satisfará las necesidades médicas, sino que también podría convertirse en un catalizador de cambio positivo en la salud de la población nicaragüense en general.

Es relevante señalar que en Nicaragua la mayoría de la población tiene acceso a dispositivos móviles e Internet, lo que crea una oportunidad excepcional para utilizar una plataforma de atención

médica con una amplia gama de servicios. Estos van desde el monitoreo de los niveles de glucosa en sangre y la gestión de la dieta y el ejercicio, hasta la programación de recordatorios para la toma de medicamentos y consultas médicas virtuales con especialistas en diabetes. Además, esta innovadora solución puede convertirse en una fuente valiosa de educación y apoyo, proporcionando información actualizada sobre la enfermedad, consejos de autocuidado y recursos para adoptar un estilo de vida saludable.

VI. Propuesta de Valor

"Control Health - Transformando la Gestión de la Diabetes en Nicaragua"

Control Health, está comprometido a mejorar la calidad de vida de las personas que viven con diabetes en Nicaragua, abordando las preocupaciones fundamentales que enfrentan diariamente. Esta plataforma integral y especializada en diabetes está diseñada para ofrecer una solución completa y personalizada a través de servicios médicos de vanguardia, educación actualizada y apoyo continuo.

¿Qué ofrece Control Health?

Control Médico Personalizado: Brinda acceso a médicos diabetólogos altamente calificados y consultas médicas virtuales para un seguimiento médico constante y una atención individualizada.

Educación Integral: Proporciona información actualizada sobre la enfermedad, consejos de autocuidado y recursos para ayudarte a adoptar un estilo de vida más saludable.

¿Por qué Control Health?

Necesidad Respaldada: Se ha realizado un estudio de mercado que incluye la opinión de médicos diabetólogos del país y la voz de 50 personas que padecen Diabetes tipo 1 y 2, respaldando la urgente necesidad de nuestros servicios. Es importante subrayar la destacada participación y comunicación activa de los nutricionistas y entrenadores físicos nicaragüenses, quienes desempeñarán un papel fundamental en la exitosa implementación y prestación de alta calidad de los módulos disponibles en la plataforma.

Tecnología Accesible: La creciente disponibilidad de tecnología en Nicaragua, asegurando que esta plataforma sea accesible para la mayoría de la población.

Transformación de la Atención Médica: Inspirados por la revolución global de la telemedicina, se mejora la eficiencia, accesibilidad y seguridad de la atención médica en Nicaragua.

Agente de Cambio Positivo: No solo se aborda las necesidades médicas, sino que también se trabaja para aumentar la conciencia sobre la diabetes y promover hábitos de vida saludables en la población en general.

La misión de Control Health es proporcionar a las personas con diabetes mediante una aplicación web que les permita gestionar su salud desde casa, ofreciendo apoyo personalizado, conectándolos con expertos y herramientas para controlar la medicación, alimentación, actividad física y niveles de azúcar en la sangre con el objetivo de mejorar su calidad de vida.

VII. Ciclo del proyecto

El proyecto Control Health se gestionará siguiendo la metodología SCRUM para desarrollo de software.

7.1 Historias de Usuario

A continuación, se presentan los requisitos clave organizados en historias de usuario siguiendo la metodología establecida. Estos requisitos Han sido cuidadosamente definidos teniendo en cuenta tres roles fundamentales en la aplicación:

Paciente: Este tipo de usuario utilizará la aplicación para acceder a las funcionalidades propuestas y llevar un control de su medicación, retroalimentarse con los apartadores informativos de la diabetes, dietas nutricionales y rutinas de ejercicio.

Médico: Los profesionales de la salud usarán la aplicación para un control de expedientes y agendamiento de citas médicas.

Administrador: Se encargará de supervisar y gestionar la plataforma visualizando la cantidad de usuarios que se han registrado y utilidad que le dan a la aplicación además de tener un control en forma de CRUD que permita administrar la aplicación de forma correcta.

Además de definir los roles de usuario, en las historias de usuario se incluyen detalles esenciales que determinan las necesidades a satisfacer, la prioridad de desarrollo, el impacto en la experiencia del usuario, los criterios de validación y estimaciones de tiempo y recursos para implementar las funcionalidades necesarias. Esto garantizará que la aplicación cumpla con las expectativas y requerimientos de manera eficiente y efectiva.

7.1.1 Historias de Usuario Paciente

Tabla 7

Historia de Usuario Paciente 01

HUP01-Control de la medicación.			
Como Paciente			
Quiero llevar un control de mi medicación			
Para tener orden de los medicamentos previamente recetados			
Validación	La aplicación debe permitir al paciente registrar la información de los medicamentos (nombre, dosis y hora). La aplicación debe ser capaz de enviar notificaciones de la toma del medicamento.	Impacto	9
		Prioridad	10
		Estimación	32
			horas

Tabla 8

Historia de Usuario Paciente 02

HUP02-Notificaciones.			
Como Paciente			
Quiero recibir notificaciones personalizables que me notifiquen la toma del medicamento y respuestas de las citas			
Para toma de medicamentos en tiempo y forma de manera adecuada, y también para mis citas médicas futuras			
Validación	Verificar la personalización de notificaciones para pacientes. Verificar el uso de notificaciones para asegurar entregas apropiadas y una comunicación efectiva entre pacientes y médicos. Verificar que pacientes y médicos reciban notificaciones detalladas de citas confirmadas, incluyendo fecha, hora y ubicación. Verificar la confirmación de recepción de notificaciones por parte de los usuarios para garantizar una comunicación efectiva.	Impacto	8
		Prioridad	9
		Estimación	36
			horas

Tabla 9

Historia de Usuario Paciente 03

HUP03-Espacios informativos de la diabetes.			
Como Paciente			
Quiero visualizar información relevante de la enfermedad, información resumida de todos los módulos y servicios de la app			
Para obtener retroalimentación sobre la enfermedad y las funcionalidades			
Validación	Verificar que la aplicación proporcione información actualizada sobre la diabetes.	Impacto	6
	Asegurar que la información proporcionada sea clara y comprensible para los pacientes.	Prioridad	7
	Verificar que los pacientes puedan acceder fácilmente a la información de la enfermedad.	Estimación	24 horas

Tabla 10

Historia de Usuario Paciente 04

HUP04-Espacios nutricionales.			
Como Paciente			
Quiero visualizar dietas nutricionales con su procedimiento de preparación e ingredientes correspondientes			
Para llevar una buena alimentación y un estilo de vida saludable			
Validación	Verificar que las dietas sean elaboradas por profesionales de la salud.	Impacto	7
	Asegurar que las dietas se ajusten a las necesidades de los pacientes.	Prioridad	7
	Verificar que el procedimiento e ingredientes estén claros y comprensibles.	Estimación	24 horas
	Asegurar que la interfaz sea visualmente atractiva utilizando imágenes de alimentos o platos terminados.		

Tabla 11

Historia de Usuario Paciente 05

HUP05-Espacio de actividad física.			
Como Paciente			
Quiero visualizar imágenes y videos con su descripción			
Para realizar actividad física desde la comodidad de mi hogar y evitar complicaciones en mi salud			
Validación	Verificar que la aplicación permita visualizar imágenes y video con su descripción.	Impacto	5
	Evaluar la calidad de las descripciones proporcionadas para las imágenes y videos.	Prioridad	6
	Verificar que los pacientes puedan acceder fácilmente a las imágenes y videos.	Estimación	24 horas
	Verificar que las rutinas sean hechas por un profesional en el tema.		

Tabla 12

Historia de Usuario Paciente 06

HUP06-Registrar paciente usuarios.			
Como Paciente			
Quiero registrar mis datos personales e información			
Para tener acceso a las funcionalidades de la aplicación			
Validación	Validar que la aplicación permita al paciente registrarse de manera exitosa.	Impacto	3
	Verificar que una vez registrado el paciente puede iniciar sesión.	Prioridad	5
	Validar que la aplicación implemente medidas de seguridad para proteger credenciales del paciente.	Estimación	24 horas

Tabla 13

Historia de Usuario Paciente 07

HUP07-Creación de citas.			
Como Paciente			
Quiero crear una cita con el médico de mi preferencia			
Para solicitar el servicio de un médico diabetólogo disponible en la app			
Validación	Validar que la aplicación cuente con una lista de médicos diabetólogos.	Impacto	8
		Prioridad	8
	Verificar que el paciente pueda seleccionar al médico de su preferencia. Validar que el paciente pueda crear una cita con el médico seleccionado. Verificar que se recibió una confirmación de la cita creada. Verificar que se llene el formulario para la gestión de la cita.	Estimación	36 horas

7.1.2 Historias de Usuario Médico

Tabla 14

Historia de Usuario Médico 01

HUM01-Registrar usuario médico			
Como médico			
Quiero registrarme en la aplicación con un correo y una contraseña			
Para tener acceso a las funcionalidades de la aplicación			
Validación	Validar que la aplicación permita al médico registrarse en la app usando correo y contraseña.	Impacto	3
		Prioridad	5
	Verificar que el médico pueda iniciar sesión usando sus credenciales. Validar que las aplicaciones implemente medidas de seguridad adecuada para las credenciales del médico. Validar que el médico tenga accesos a las funcionalidades adecuadas.	Estimación	24 horas

Tabla 15

Historia de Usuario Médico 02

HUM02-Espacio del médico			
Como médico			
Quiero tener un espacio en la aplicación			
Para poder gestionar las citas de mis pacientes asignados			
Validación	Validar que el médico pueda acceder a su espacio designado para gestionar los expedientes.	Impacto	7
	Verificar que el médico pueda comunicarse con sus pacientes de manera segura.	Prioridad	7
	Validar que la aplicación aplique medidas de seguridad para proteger la privacidad de los pacientes.	Estimación	36 horas

Tabla 16

Historia de Usuario Médico 03

HUM03-Agendar citas			
Como médico			
Quiero agendar las citas que gestionen los usuarios			
Para brindar mis servicios como médico diabetólogo			
Validación:	Validar que el médico tenga acceso al agendamiento de citas.	Impacto:	7
	Validar que el médico reciba las solicitudes de las citas.	Prioridad:	7
	Verificar que el médico pueda programar citas realizadas por los usuarios.	Estimación:	32 horas
	Validar que el médico pueda confirmar o rechazar las citas programadas.		

Tabla 17

Historia de Usuario Médico 04

HUM04-Creación de calendario			
Como médico			
Quiero gestionar un calendario			
Para para establecer horarios de consulta, fechas y ver la disponibilidad según sea necesario			
Validación:	Validar que el médico tenga acceso a una interfaz de	Impacto	6
	gestión de calendario de la aplicación.	Prioridad	7
	Verificar que el médico pueda establecer horarios de	Estimación	32 horas
	consulta de acuerdo con la disponibilidad.		
	Validar que el médico pueda administrar fechas y citas en		
	su calendario.		
	Verificar que el médico pueda ver de manera clara y		
	precisa su disponibilidad en el calendario.		

7.1.3 Historias de Usuario Administrador

Tabla 18

Historia de Usuario Administrador 01

HUA01-Administración			
Como Administrador			
Quiero Visualizar todos los usuarios en la aplicación			
Para validar y eliminar usuarios			
Validación:	Mostrar lista de todos los usuarios en orden alfabético.	Impacto	8
		Prioridad	8
		Estimación	24 horas

Tabla 19

Historia de Usuario Administrador 02

HUA02-Estadísticas de la aplicación			
Como Administrador			
Quiero Visualizar estadísticas de la funcionalidad de citas y la predominancia de la diabetes			
Para Obtener información valiosa e importante del uso frecuente de las funcionalidades y usuarios			
Validación:	Visualizar Gráficos estadísticos que permitan ver el nivel de aceptación de las funcionalidades.	Impacto	9
		Prioridad	9
		Estimación	24 horas

7.2 Product Backlog

Tabla que describe las historias de usuario junto con su prioridad, importancia, impacto y tiempo estimado. Se asignado una puntuación de importancia del 1 al 30, una puntuación del 1 al 10 en cuanto a la prioridad y del 1 al 10 en cuanto a su impacto para reflejar el valor que cada historia aporta al proyecto.

Tabla 20

Product Backlog

ID	Historia	Prioridad	Impacto	Importancia	Tiempo estimado
HUP01	Control de la medicación	10	9	38	32 horas
HUP02	Notificaciones personalizadas	9	8	35	36 horas
HUP03	Espacio informativo de la diabetes	7	6	25	24 horas
HUP04	Espacios nutricionales	7	7	38	24 horas
HUP05	Espacios de actividad física	6	5	20	24 horas
HUP06	Registrar usuario paciente	5	3	20	24 horas
HUP07	Creación de citas	8	8	38	36 horas
HUP08	ChatBot	8	8	20	24 horas
HUM01	Registro de usuario médico	5	3	15	24 horas
HUM02	Espacio del médico	7	7	20	36 horas
HUM03	Agendar citas	7	7	25	32 horas
HUM04	Creación de calendario	7	6	30	32 horas
HUP01	Crud Administrativo	8	8	20	24 horas
HUA02	Estadísticas de la aplicación	9	9	25	24 horas

7.3 Product Backlog Organizado por Sprint

A continuación se presenta el Product Backlog Organizado por Sprint caracterizada por ser una lista de tareas y funcionalidades de Control health. Se gestiona en ciclos llamados Sprints, donde se seleccionan y abordan elementos específicos del backlog.

Tabla 21

Product Backlog organizado por Sprint

ID	Historia	Importancia	Sprint	Fecha Prevista
HUP04	Espacios nutricionales	38	1	04/09/2023
HUP05	Espacios de actividad física	20	1	09/09/23
HUP06	Registrar usuario paciente	20	1	
HUP01	Control de la medicación	38	2	20/09/2023
HUP07	Creación de citas	38	2	20/10/2023
HUM03	Agendar Citas	25	2	
HUP03	Espacios informativos de la diabetes	25	2	
HUM01	Registro de usuario médico	15	2	
HUP08	ChatBot	20	2	
HUM02	Espacio del médico	20	3	24/10/2023
HUM04	Creación de calendario	30	3	29/10/2023
HUP02	Notificaciones personalizadas	35	3	
HUA01	Crud Administrativo	20	4	29/10/2023
HUA02	Estadísticas de la aplicación	25	4	30/10/2023

Nota: Historias de usuario en orden descendente según su importancia.

7.4 Sprint Backlog

Representa un conjunto de tareas que fueron seleccionadas del product backlog durante la planeación para el actual Sprint tomando en cuenta las tareas necesarias para realizar un incremento de producto.

7.4.1 Sprint 1

Se muestra el inicio del primer Sprint, programado para una semana de duración. En este Sprint, se planifica la inclusión de tres historias de usuario junto con la asignación de un total de diecinueve tareas que deberán ser completadas dentro de ese período.

Tabla 22

Sprint 1

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUP04	Espacios nutricionales.	Realizar Entrevista a Nutricionista.	2
		Diseñar interfaz para el espacio y estilos css.	2
		Agregar información relevante y verificada de dietas nutricionales en la diabetes.	5
		Crear un data para almacenar los datos de ese platillo.	5
		Crear CARD que permita visualizar el platillo.	5
		Crear Modal para ver los detalles de ese platillo.	5
HUP05	Espacios de actividad física.	Realizar Entrevista entrenador físico.	4
		Diseñar interfaz para el espacio.	4
		Agregar información relevante y verificada de la actividad física en la diabetes.	4
		Crear una data para almacenar los datos de las rutinas.	4
		Crear CARD que permite visualizar las rutinas.	4
		Crear Modal para ver la rutina con su detalle y video.	4
HUP06	Registrar usuarios Paciente.	Configurar Firebase Authentication.	3
		Crear colección para almacenamiento de usuarios en Firestore.	3
		Configurar redux para la Autenticación de usuarios.	3
		Codificar funcionalidad de Registro de usuario.	3
		Diseño de la vista de inicio y registro de sesión.	6
		Validar la entrada del Usuario.	6
		Almacenar información adicional del usuario.	6

Figura 9*Informe de trabajo completado Sprint 1*

Nota: Se muestra un gráfico del progreso de incidencias de trabajo realizado en el Sprint 1.

7.4.1.1 Entregables Sprint 1.

Figura 10

HUP04-Espacios Nutricionales

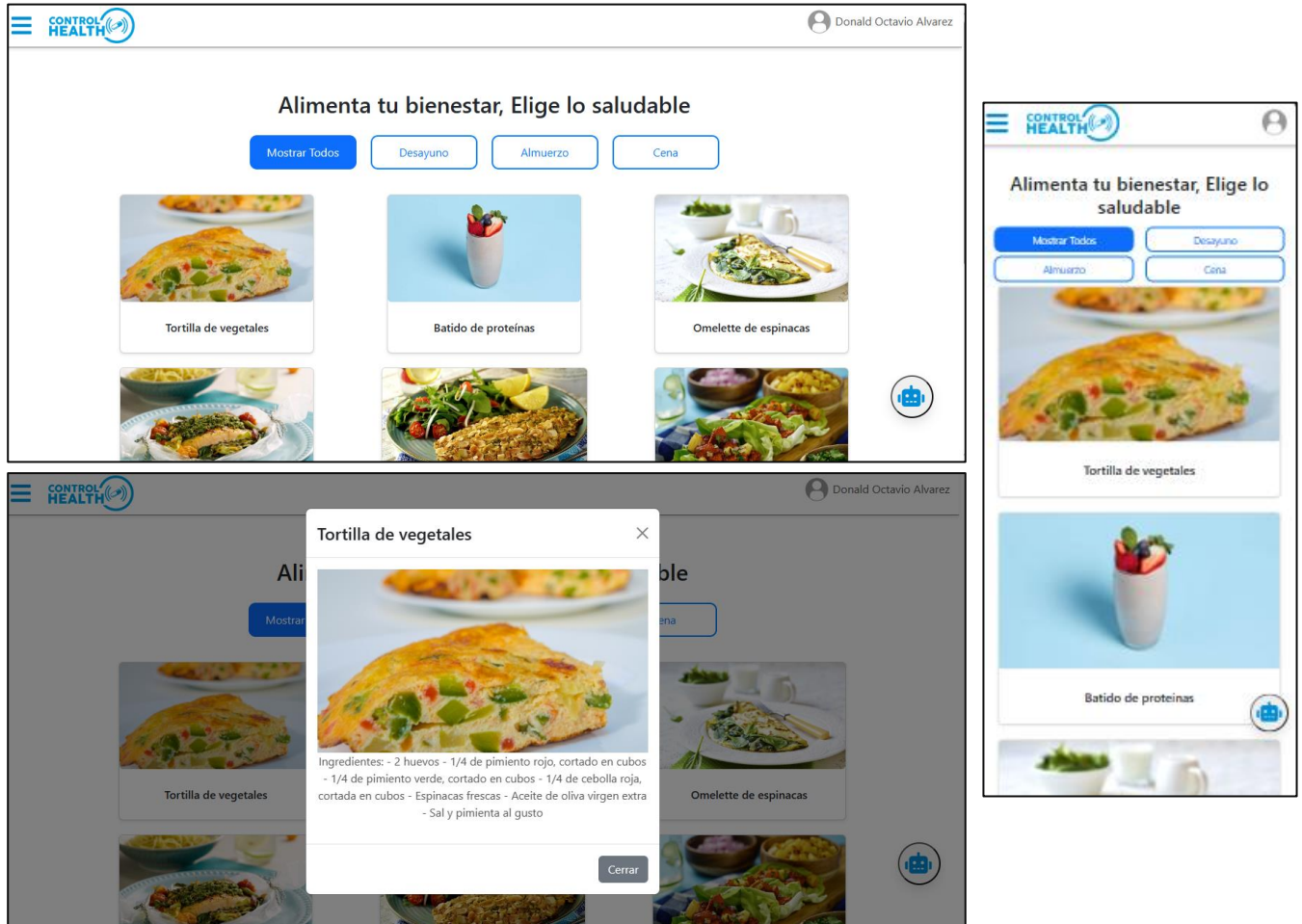


Figura 11

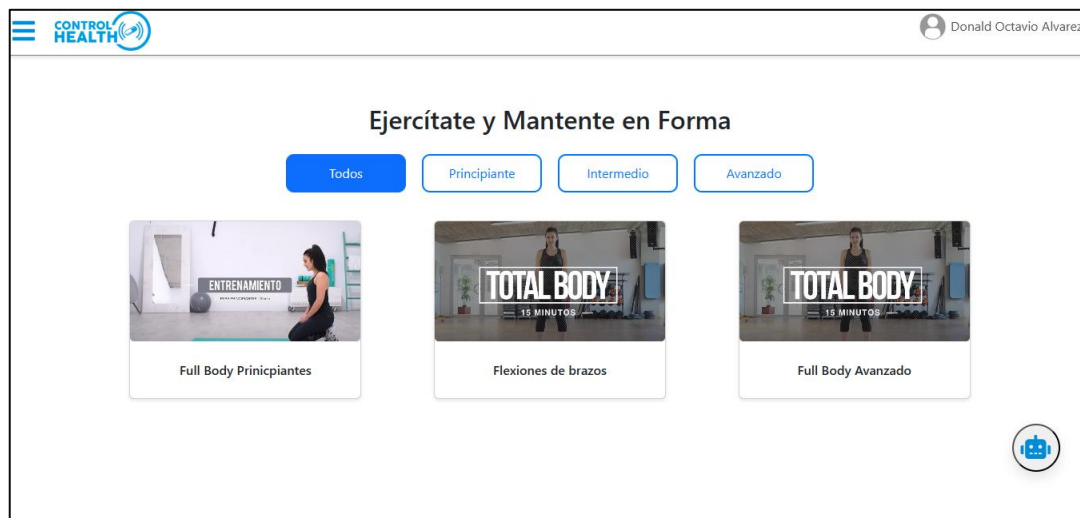
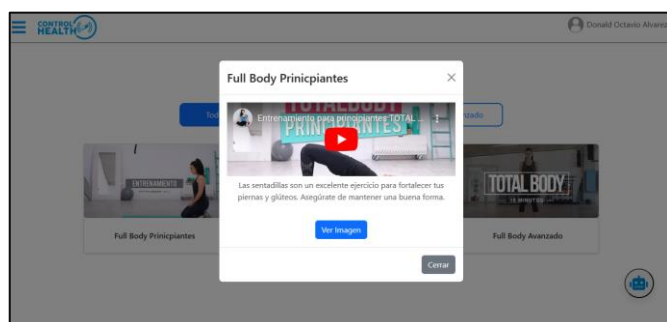
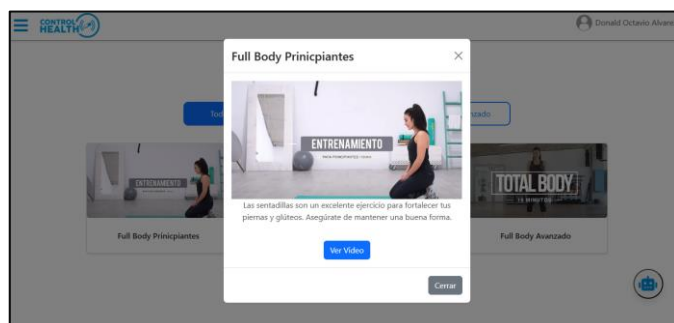
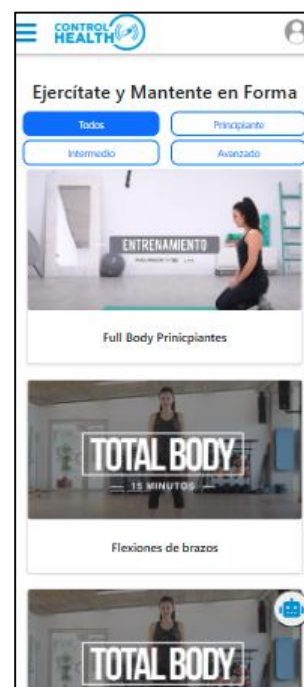
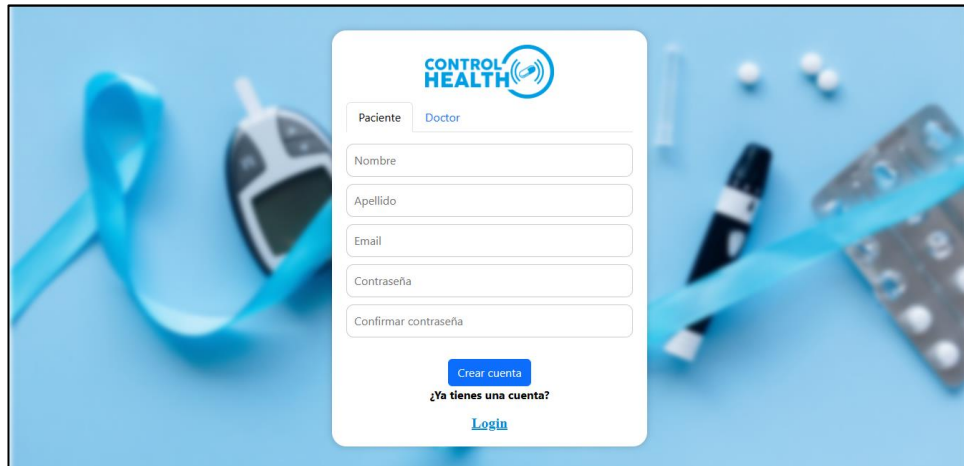
HUP05-Espacios de Actividad Física

Figura 12*HUP06-registar Usuario Paciente*

CONTROL HEALTH

Paciente ☐ Doctor ☐

Nombre

Apellido

Email

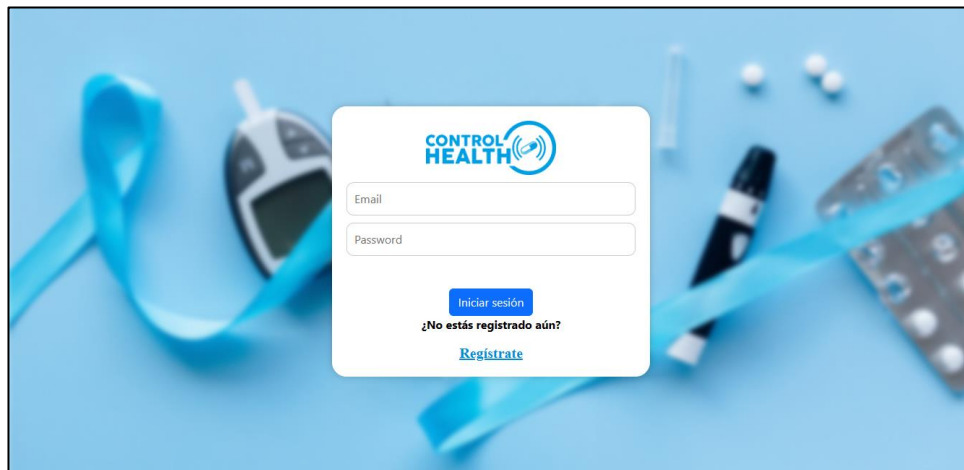
Contraseña

Confirmar contraseña

[Crear cuenta](#)

¿Ya tienes una cuenta?

[Login](#)



CONTROL HEALTH

Email

Password

[Iniciar sesión](#)

¿No estás registrado aún?

[Regístrate](#)



CONTROL HEALTH

Paciente ☐ Doctor ☐

Nombre

Apellido

Email

Contraseña

Confirmar contraseña

[Crear cuenta](#)

¿Ya tienes una cuenta?

[Login](#)



CONTROL HEALTH

Email

Password

[Iniciar sesión](#)

¿No estás registrado aún?

[Regístrate](#)

7.4.1.2 Prueba de software del Sprint 1.

Figura 13

Prueba de Software del Sprint 1

```

PASS src/Pruebas/dietprueba.test.js
Diet Component
  ✓ renders without crashing (49 ms)
  ✓ displays the category buttons (25 ms)
  ✓ displays item cards (107 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.862 s, estimated 3 s
Ran all test suites matching /dietprueba.test.js/i.

PASS src/Pruebas/register.test.js
  ✓ Renderiza el componente UserRegister sin errores (56 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.675 s
Ran all test suites matching /register.test.js/i.

PASS src/Pruebas/filtrardiet.test.js
  ✓ Filtrar elementos por categoría en el componente Diet (95 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.983 s, estimated 3 s
Ran all test suites matching /filtrardiet.test.js/i.

```

Nota: Se muestra el resultado de la ejecución de la prueba donde se observa que los componentes Diet y Register se comportan como se espera.

7.4.2 Sprint 2

Inicio del segundo Sprint, programado para tres semanas de duración. En este Sprint se ha planificado la inclusión de cinco historias de usuario junto a la asignación de un total de 25 tareas que deberán ser completadas dentro de este periodo.

Tabla 23

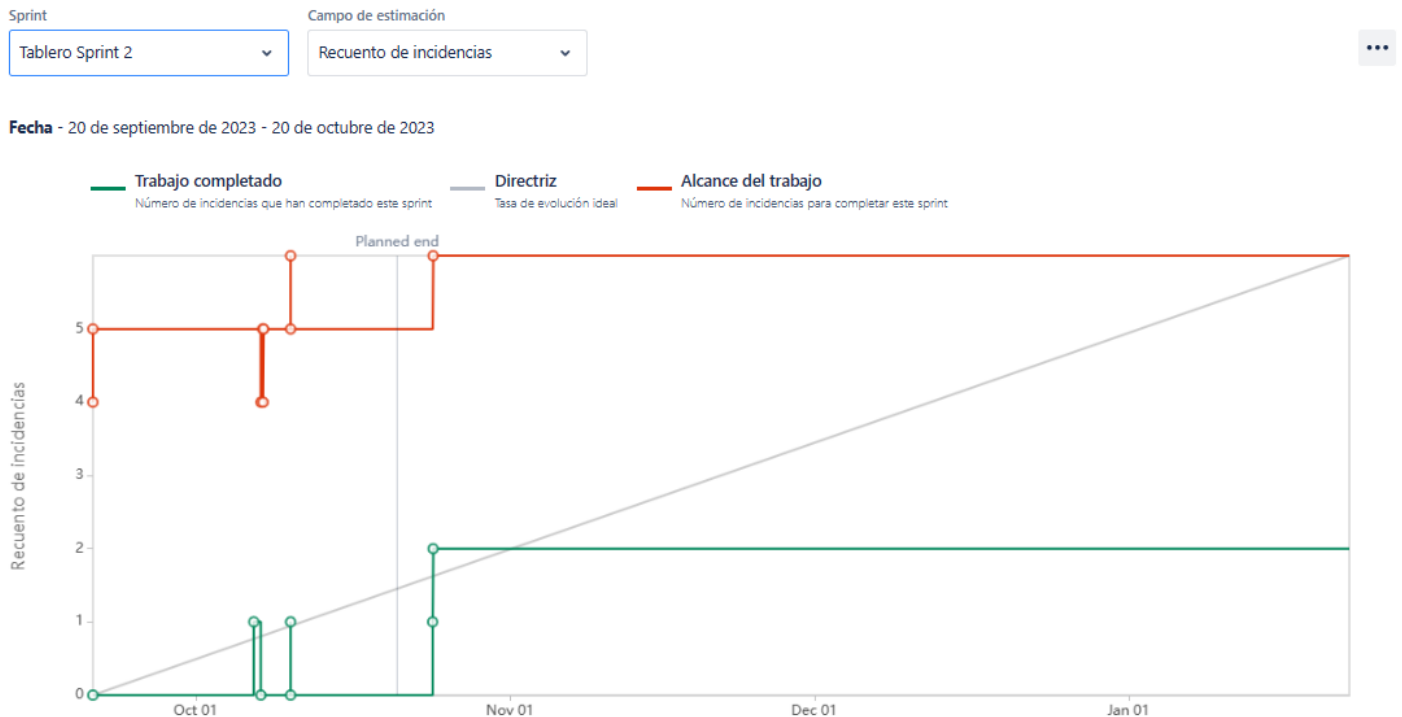
Sprint 2

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUP08	ChatBot	Creación de formulario para el ChatBot.	8
		Creación y estilos de botón flotante.	8
		Importación en vistas correspondientes.	8
HUP01	Control de la medicación.	Análisis de formularios para la creación de la vista de control de la medicación.	6
		Configuración de Firebase-Firestore para almacenar los datos de los medicamentos.	6
		Creación de formularios para el control de la medicación.	4
		Creación de modal para visualizar datos del formulario.	4
		Diseñar interfaz para el control de la medicación.	4
		Creación de CARD para visualizar medicamentos agregados.	4
		Creación de modal para visualizar y editar los datos.	4
HUP07	Creación de citas.	Análisis de formulario para la creación de citas.	10
		Configuración de Firebase-Firestore para almacenar los datos de las citas.	8
		Crear formularios para la solicitud de la cita.	6
		Crear modal para visualizar formulario.	6
		Crear CARD para mostrar la solicitud de la cita	6
HUM03	Agendar citas.	Consumir los datos del Firestore recibiendo la cita.	8
		Diseñar espacio del médico para guardar las citas.	8
		Crear función que permita aceptar o rechazar cita.	8

		Crear función que permita editar campos permitidos de la cita como el día y la hora.	8
HUP03	Espacio informativo sobre la diabetes.	Realizar entrevista a médico diabetólogo.	8
		Diseñar interfaz del espacio.	8
		Agregar información relevante y verificada de la diabetes.	8
HUM01	Registro de usuario médico	Combinar registro de usuarios con médicos y pacientes.	3
		Modificar Firebase Authentication.	3
		Modificar redux para la Autenticación de usuarios paciente y médico.	3
		Modificar interfaz agregando registro del médico.	3

Figura 14

Informe de trabajo completado Sprint 2



Nota: Progreso de incidencias de trabajo realizado en el Sprint 2.

7.4.2.1 Entregables Sprint 2.

Figura 15

HUP01-Control de la medicación

The figure displays two mobile application screens for medication control, both featuring the 'CONTROL HEALTH' logo and a user profile icon for Donald Octavio Alvarez.

Left Screen (Medication List):

- Header:** Medicación
- Table:**

	Nombre del Medicamento:	Dosificación:	Unidades:	Tomas al Día:	Fecha de Inicio:	Hora:	
	Metformina	mg	500	1	2023-11-03	2:42 a. m.	

- Bottom Right:** A circular icon with a robot head.

Right Screen (Medication Detail):

- Header:** Medicación
- Form:**

	Nombre del Medicamento:	Dosificación:	
	Metformina	mg	
Unidades:	Tomas al Día:	Fecha de Inicio:	Hora:
500	1	2023-11-03	2:42 a. m.

- Bottom Left:**
- Bottom Right:** A circular icon with a robot head.

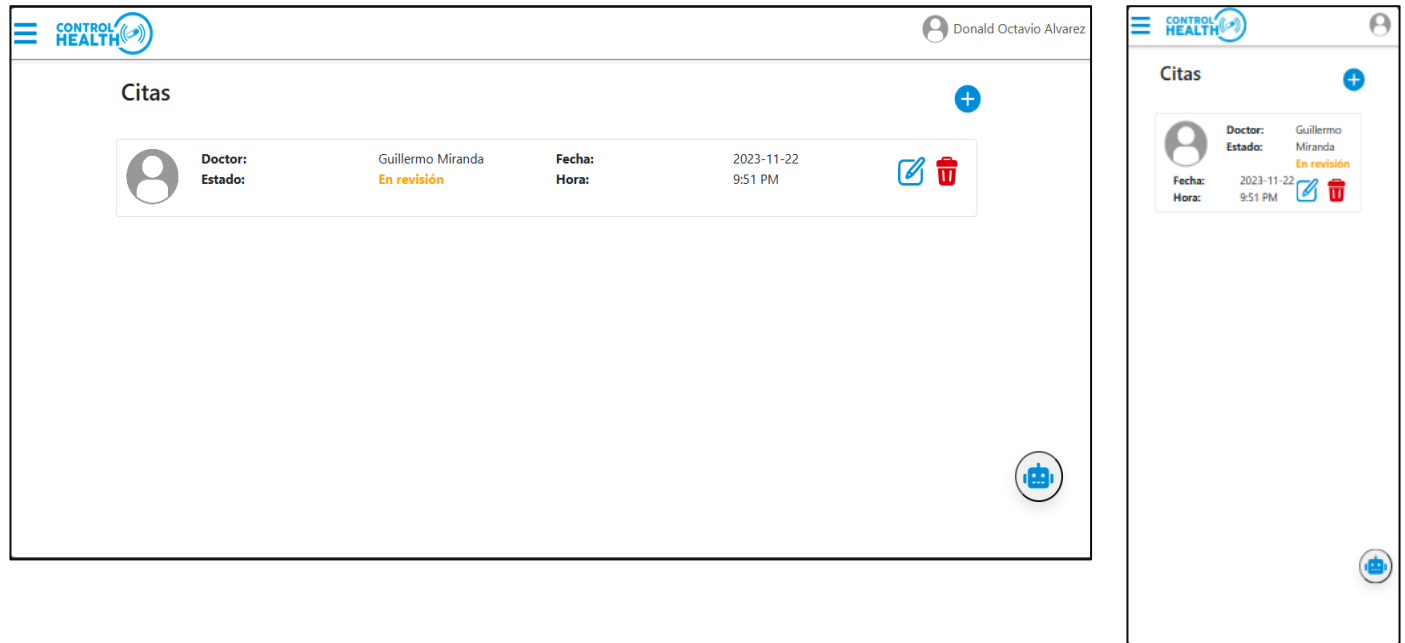
Figura 16*HUP07-Creación de citas*

Figura 17

HUP03-Espacios informativo de Diabetes

Figura 18

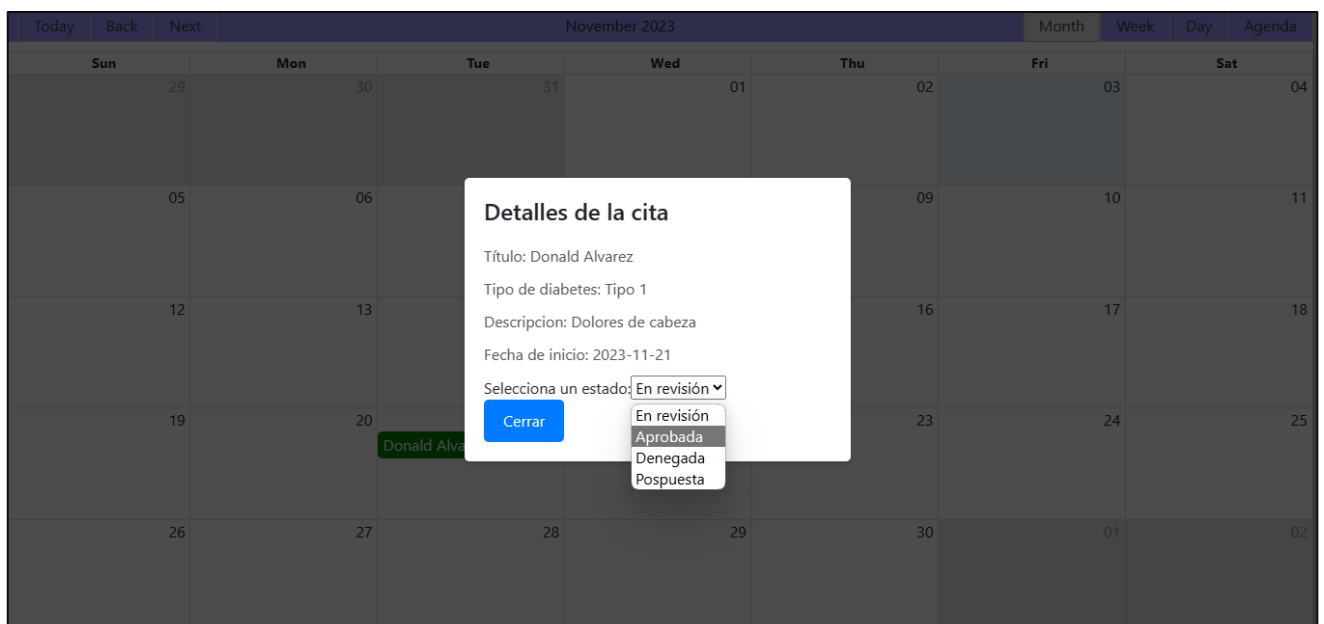
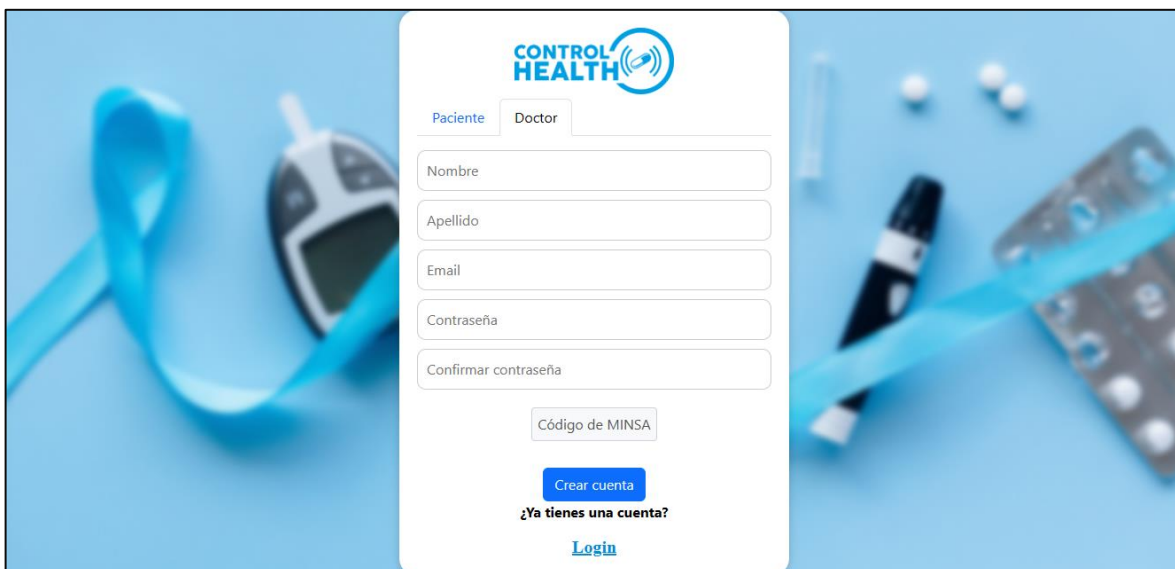
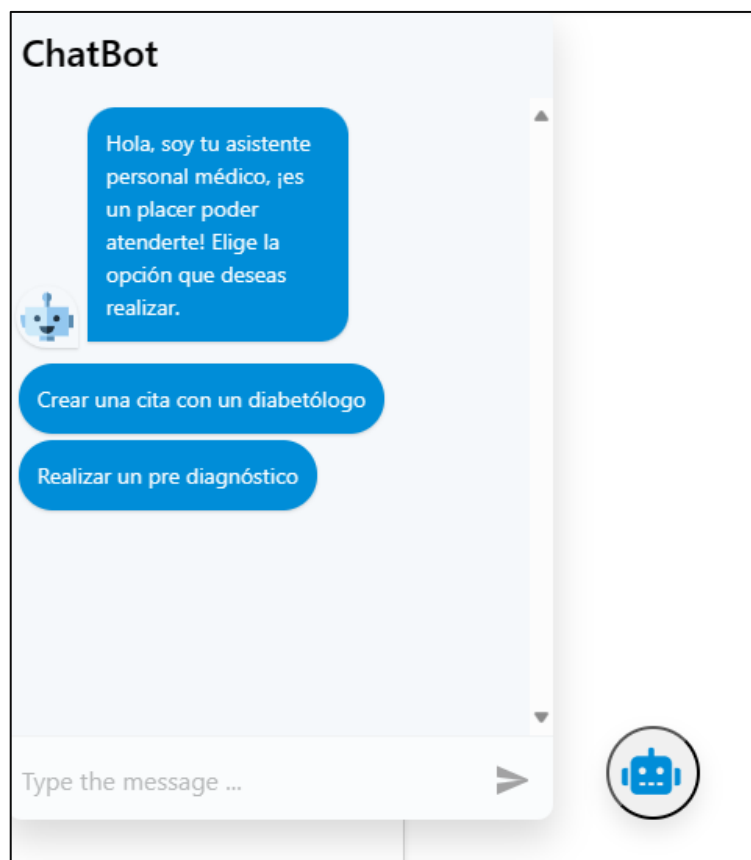
HUM04-Agendar Cita

Figura 20*HUM01 Registrar usuario médico*

The screenshot shows a registration form for 'CONTROL HEALTH'. The form is centered on a light blue background with medical-themed images (a blue ribbon, a glucose meter, and a blister pack of pills). The form has a white background and includes the following elements:

- Logo:** 'CONTROL HEALTH' with a blue circular icon containing a white signal wave.
- Role Selection:** Two tabs, 'Paciente' (selected) and 'Doctor'.
- Input Fields:** Five white rectangular input fields for 'Nombre', 'Apellido', 'Email', 'Contraseña', and 'Confirmar contraseña'.
- MINSA Code:** A small grey box labeled 'Código de MINSA'.
- Buttons:** A blue 'Crear cuenta' button and a blue 'Login' link.
- Text:** A line of text asking '¿Ya tienes una cuenta?'.

Figura 19*HUP08-ChatBot*

The screenshot shows a chatbot interface titled 'ChatBot'. The chat area has a light blue background and contains the following elements:

- Greeting:** A blue speech bubble with a white robot icon on the left, containing the text: 'Hola, soy tu asistente personal médico, ¿es un placer poder atenderte! Elige la opción que deseas realizar.'
- Options:** Two blue rounded rectangular buttons with white text: 'Crear una cita con un diabetólogo' and 'Realizar un pre diagnóstico'.
- Input Area:** A white input field with the placeholder text 'Type the message ...' and a blue send button (a white triangle inside a blue circle).
- Bot Icon:** A blue robot icon inside a white circle, located at the bottom right of the chat area.

7.4.2.2 Pruebas de software Sprint 2.

Ejecución de la prueba control de la medicación del Sprint 2 destacando el enfoque de la experiencia donde se muestra el medicamento agregado con éxito, La creación de citas, Agendar Citas médicas y registro de usuario médico.

Figura 21

Prueba de Software Sprint 2

```

PASS src/Pruebas/medication.test.js
  ✓ Renderiza el componente Medication con datos en la pantalla (75 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        3.448 s, estimated 5 s
Ran all test suites matching /medication.test.js/i.

PASS src/Pruebas/quotesolicitud.test.js
  ✓ Renderiza el componente Quote con datos en las cards (33 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.652 s, estimated 4 s
Ran all test suites matching /quotesolicitud.test.js/i.

PASS src/Pruebas/agendarcitas.test.js
  ✓ Renderiza el calendario de CitasDr y verifica una cita agendada (27 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.498 s
Ran all test suites matching /agendarcitas.test.js/i.

PASS src/Pruebas/registermedico.test.js
  ✓ Renderiza el formulario de registro de usuario médico con campo de carga de código minsa (27 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.479 s
Ran all test suites matching /registermedico.test.js/i.

PASS src/Pruebas/chatbotprueba.test.js
  ✓ Muestra y oculta el chatbot al hacer clic en el botón (54 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.476 s
Ran all test suites matching /chatbotprueba.test.js/i.

```

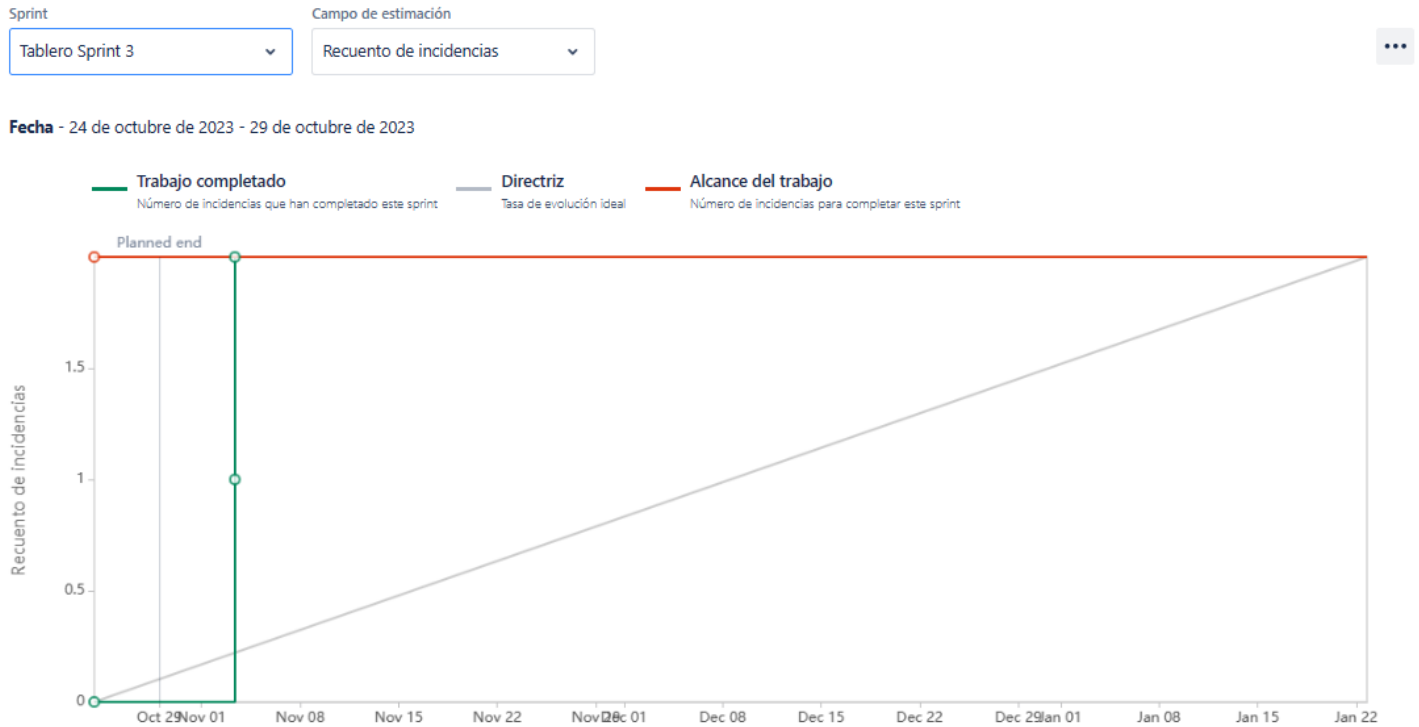
7.4.3 Sprint 3

Inicio del tercer Sprint, programado para una semana de duración. En este Sprint, hemos planificado la inclusión de tres historias de usuario junto con la asignación de un total de 15 tareas que deberán ser completadas dentro de ese período.

Tabla 24

Sprint 3

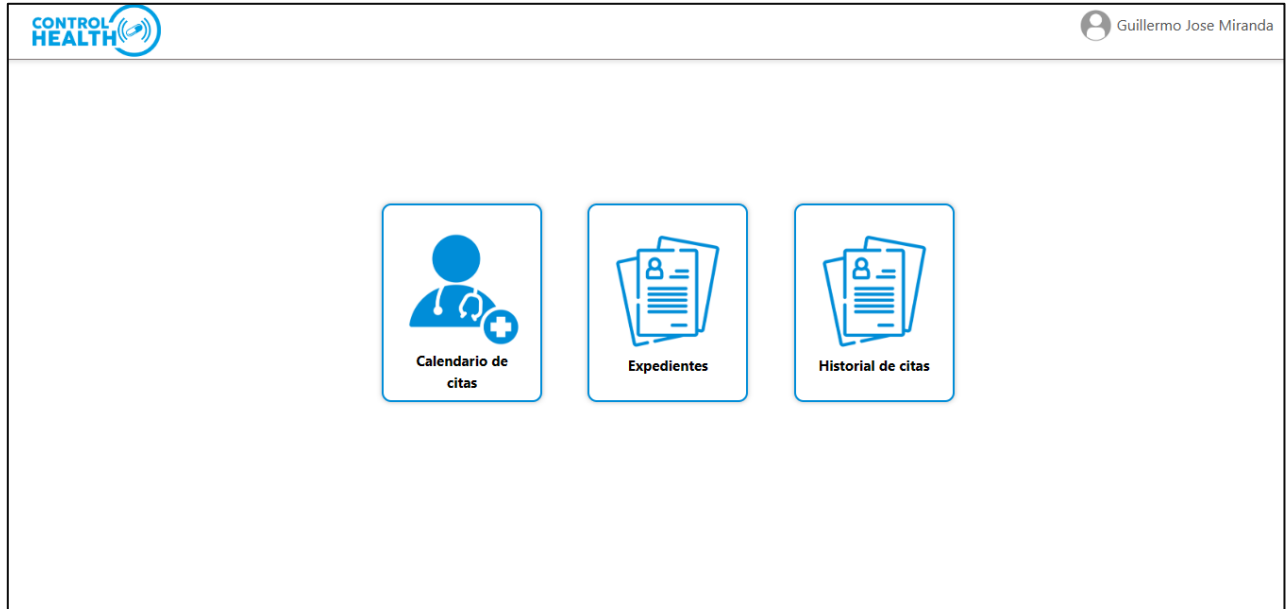
ID	Historia de usuario	Tareas	Tiempo (Horas)
HUM03	Espacio del médico.	Consumir datos de Firestore.	5
		Configurar redux para recibir los datos.	5
		Configurar notificaciones del médico.	5
		Diseñar interfaz para el espacio del médico.	4
		Diseñar módulos del médico.	5
HUM05	Creación de calendario.	Configurar Firebase-Firestore para almacenar los datos.	5
		Diseñar interfaz del calendario.	5
		Programar calendario para poder interactuar agregando cita o anotaciones.	8
		Programar funciones de actualización y eliminación de eventos.	8
		Gestionar la visualización y navegación del Calendario.	6
HU02	Notificaciones.	Configurar Notificaciones.	18
		Programar funcionalidades de las notificaciones.	18

Figura 22*Informe de trabajo completado Sprint 3*

7.4.3.1 Entregables Sprint 3.

Figura 23

HUM03-Espacio del médico



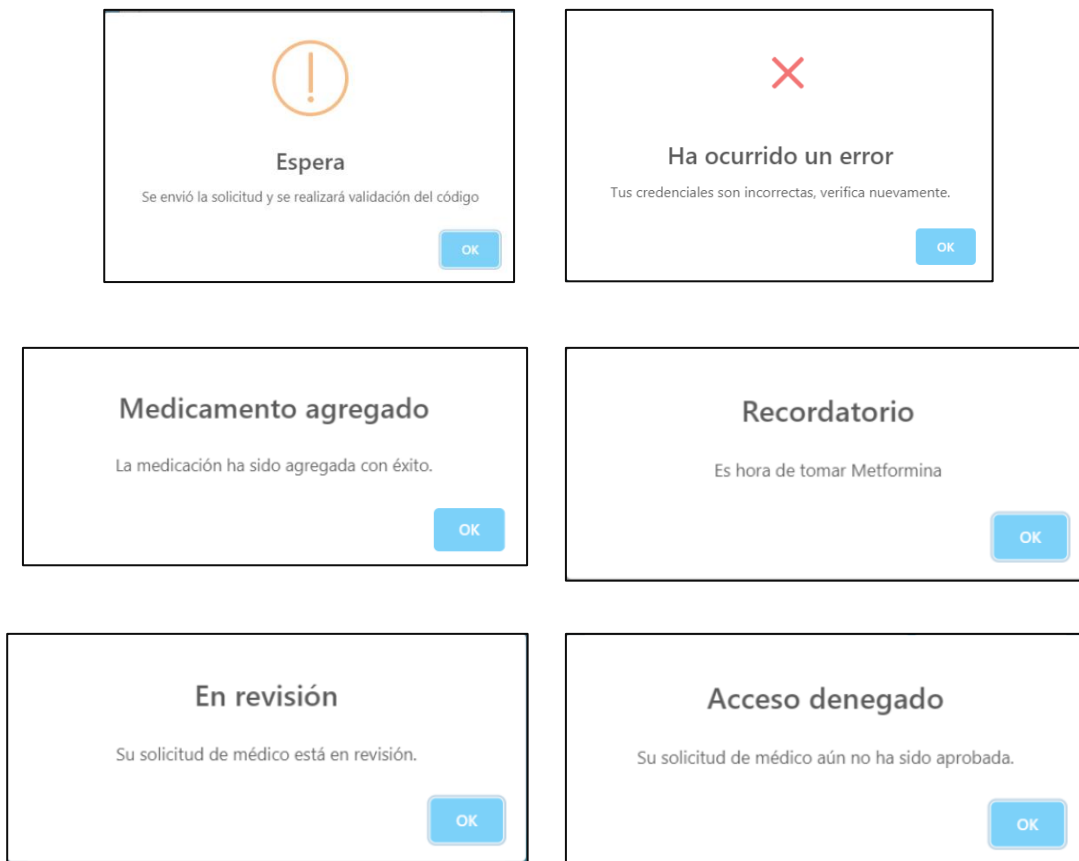
Nota: Resultado de la vista realizada del espacio del médico.

Figura 24

HUM05-Creación del calendario

Today	Back	Next	November 2023				Month	Week	Day	Agenda
Sun	Mon	Tue	Wed	Thu	Fri	Sat				
29	30	31	01	02	03	04				
05	06	07	08 Brando	09	10	11				
12	13	14	15	16	17	18				
19	20	21 Donald Alvarez	22	23	24	25				
26	27	28	29	30	01	02				

Nota: Vista realizada del calendario del médico.

Figura 25*HU02-Notificaciones*

Nota: Notificaciones de la aplicación Control Health.

7.4.3.2 Pruebas de software Sprint 3.

Ejecución de la prueba de control de medicación correspondiente al Sprint 3. Estas imágenes ponen de manifiesto el enfoque centrado en la experiencia del usuario al mostrar tanto el test de la medicación como el panel principal del médico en su entorno de trabajo.

Figura 26

Pruebas de software Sprint 3

```

PASS src/Pruebas/medication.test.js
  ✓ Renderiza el componente Medication y muestra la alerta de toma (38 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.385 s, estimated 3 s
Ran all test suites matching /medication.test.js/i.

PASS src/Pruebas/calendariocitas.test.js
  ✓ Renderiza el calendario de CitasDr y verifica una cita agendada (29 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.582 s
Ran all test suites matching /calendariocitas.test.js/i.

PASS src/Pruebas/homedrprueba.test.js
  ✓ Renderiza el componente HomeDr con datos simulados (57 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.508 s
Ran all test suites matching /homedrprueba.test.js/i.

```

7.4.4 Sprint 4

Inicio del cuarto Sprint, programado para 3 días de duración. En este Sprint, hemos planificado la inclusión de 2 historias de usuario junto con la asignación de un total de nueve tareas que deberán ser completadas dentro de ese período.

Tabla 25

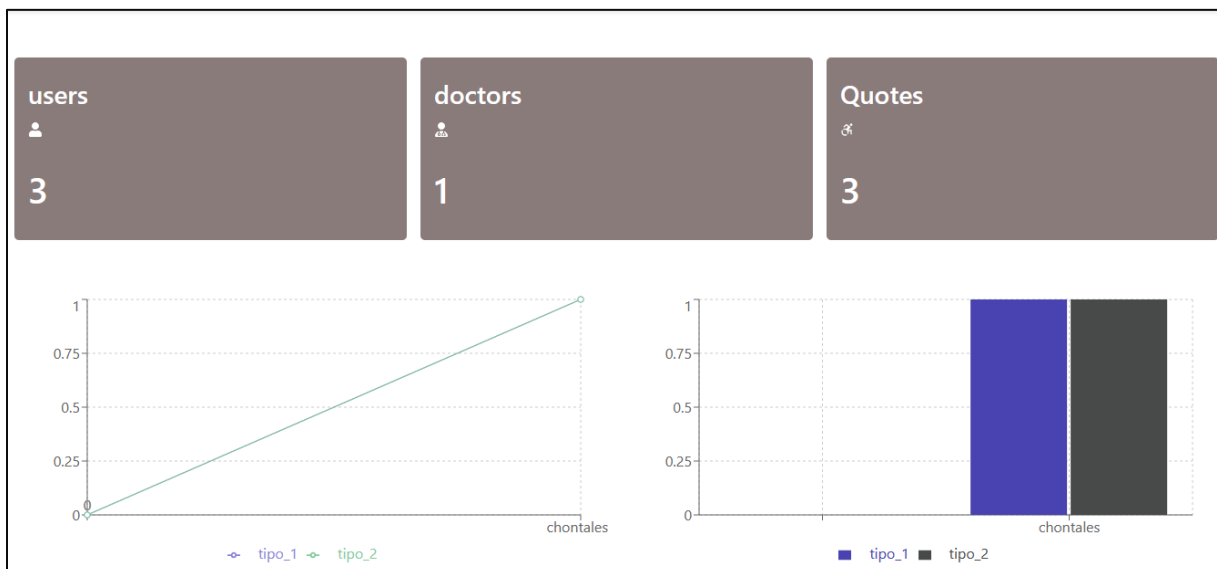
Sprint 4

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUA01	CRUD Administrativo.	Configurar Firestore para obtener los datos.	5
		Diseñar interfaz de la vista administrativa.	5
		Programar funcionalidad del CRUD.	5
		Crear tabla para mostrar los datos de los usuarios.	5
		Configurar botones para poder interactuar el CRUD.	4
HUA02	Estadísticas de la aplicación.	Configurar Firestore para obtener las analíticas.	6
		Diseñar interfaz para mostrar las gráficas.	6
		Codificar lógica para que las gráficas sean coherentes.	6
		Analizar los datos y funcionalidades que más utilizan.	6

7.4.4.1 Entregables del Sprint 4.

Figura 27

HUA02- Estadísticas



Nota: Vista administrativa donde se visualizan las estadísticas de usuarios y su tipo de diabetes y el uso que se le da a la funcionalidad de citas.

Figura 28

HUA01-Crud Administrativo

The administrative view features the 'CONTROL HEALTH' logo and a welcome message: 'Bienvenido a tu vista administrativa'. Below this is a 'Verificación MINSa' button. A table lists doctors with their names, emails, and approval status, along with action dropdowns.

Doctor	Email	Estado	Acciones
Javiera Leiva García	javiera.leiva@gmail.com	Denegada	Denegada
Guillermo Miranda	guillermomiranda@gmail.com	Aprobada	Aprobada

Nota: Vista administrativa en la que se pueden validar a los médicos para así darles permisos de entrar a la aplicación.

7.4.4.2 Pruebas de software Sprint 4.

Figura 29

Pruebas de Software Sprint 4

```

PASS src/Pruebas/admindoctors.test.js
  Admindoctors
    ✓ debería permitir cambiar el estado de los médicos (265 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        5.242 s
Ran all test suites matching /admindoctors.test.js/i.

PASS src/Pruebas/admin.test.js
  ✓ Renderiza y visualiza estadísticas correctamente (62 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        3.617 s
Ran all test suites matching /admin.test.js/i.

```

Nota: Resultados detallados de las pruebas de administración del Sprint 4, resaltando el enfoque centrado en la experiencia del usuario al presentar estadísticas de la aplicación y validación médica.

VIII. Presupuesto

Tabla 26

Presupuesto

Presupuesto que abarca los servicios, la contratación de personal, la adquisición de software, mobiliario y equipo necesarios para llevar a cabo el proyecto. También se contempla el presupuesto para las pruebas de software que se llevarán a cabo en el marco de este proyecto.

ID	Elementos	Tipo de recursos	Tipo de Unidad	Unidad	Gasto Total
1	Etapas de Análisis				
	Analista	Gastos	Días	7	C\$793.33
	Programador	Gastos	Días	7	C\$793.33
	Computadora Analista	Hardware	Cantidad	1	C\$120.00
	Computadora Programador	Hardware	Cantidad	1	C\$500.00
	Celular Analista	Hardware	Cantidad	1	C\$174.00
	Celular Programador	Hardware	Cantidad	1	C\$120.00
Total					C\$2'500.67
2	Diseño				
	Diseñador	Gastos	Días	6	C\$460.00
	Programador	Gastos	Días	3	C\$350.00
	Computadora Diseñador	Hardware	Cantidad	1	C\$220.00
	Computadora Programador	Hardware	Cantidad	1	C\$500.00
	Celular Diseñador	Hardware	Cantidad	1	C\$100.00
	Celular Programador	Hardware	Cantidad	1	C\$120.00
	Adobe Illustrator	Software	Cantidad	1	C\$755.64
	Adobe xD	Software	Cantidad	1	C\$359.64
	Draw.io	Software	Cantidad	1	C\$0.00
Total					C\$2'865.28
3	Desarrollo				
	Programador	Gastos	Días	24	C\$2'720.00
	Computadora Programador	Hardware	Cantidad	1	C\$500.00

	Celular Programador	Hardware	Cantidad	1	C\$120.00
	Firebase	Software	Cantidad	1	C\$1'800.00
	Android Studio	Software	Cantidad	1	C\$0.00
	Visual Studio Code	Software	Cantidad	1	C\$0.00
Total					C\$5'140.00
4	Implementación				
	Programador	Gastos	Días	3	C\$350.00
	Diseñador	Gastos	Días	3	C\$230.00
	Analista	Gastos	Días	3	C\$340.00
	Celular Programador	Hardware	Cantidad	1	C\$120.00
	Celular Analista	Hardware	Cantidad	1	C\$174.00
	Celular Diseñador	Hardware	Cantidad	1	C\$100.00
Total					C\$1'314.00
5	Reporte de pruebas				
	Programador	Gastos	Días	5	C\$566.67
	Computadora Programador	Hardware	Cantidad	1	C\$500.00
	Android Studio	Software	Cantidad	1	C\$0.00
	Visual Studio Code	Software	Cantidad	1	C\$0.00
Total					C\$1'066.67
6	Marketing				
	Diseñador	Gastos	Días	3	C\$230.00
	Computadora Diseñador	Hardware	Cantidad	1	C\$100.00
	Adobe Illustrator	Software	Cantidad	1	C\$755.64
	Papel	Físicos	Cantidad	1	C\$190.00
	Lápiz	Físicos	Cantidad	1	C\$120.00
	Cuaderno	Físicos	Cantidad	1	C\$80.00
	Impresora	Hardware	Cantidad	1	C\$4'000.00
Total					C\$5'475.64
Total					C\$18'362.25

IX. Marco Lógico

Este segmento aborda la conceptualización, diseño, ejecución y evaluación de un proyecto utilizando la metodología del marco lógico. A través de esta metodología analítica, se crea la Matriz, que condensa los objetivos y estrategias del proyecto, los supuestos clave, así como el sistema de monitoreo y evaluación de los insumos y productos del proyecto.

De acuerdo con la CEPAL en su MANUAL N° 42 (2005) [16], esta metodología se divide en dos etapas que se desarrollan de manera secuencial durante las fases de identificación y diseño del ciclo de vida del proyecto:

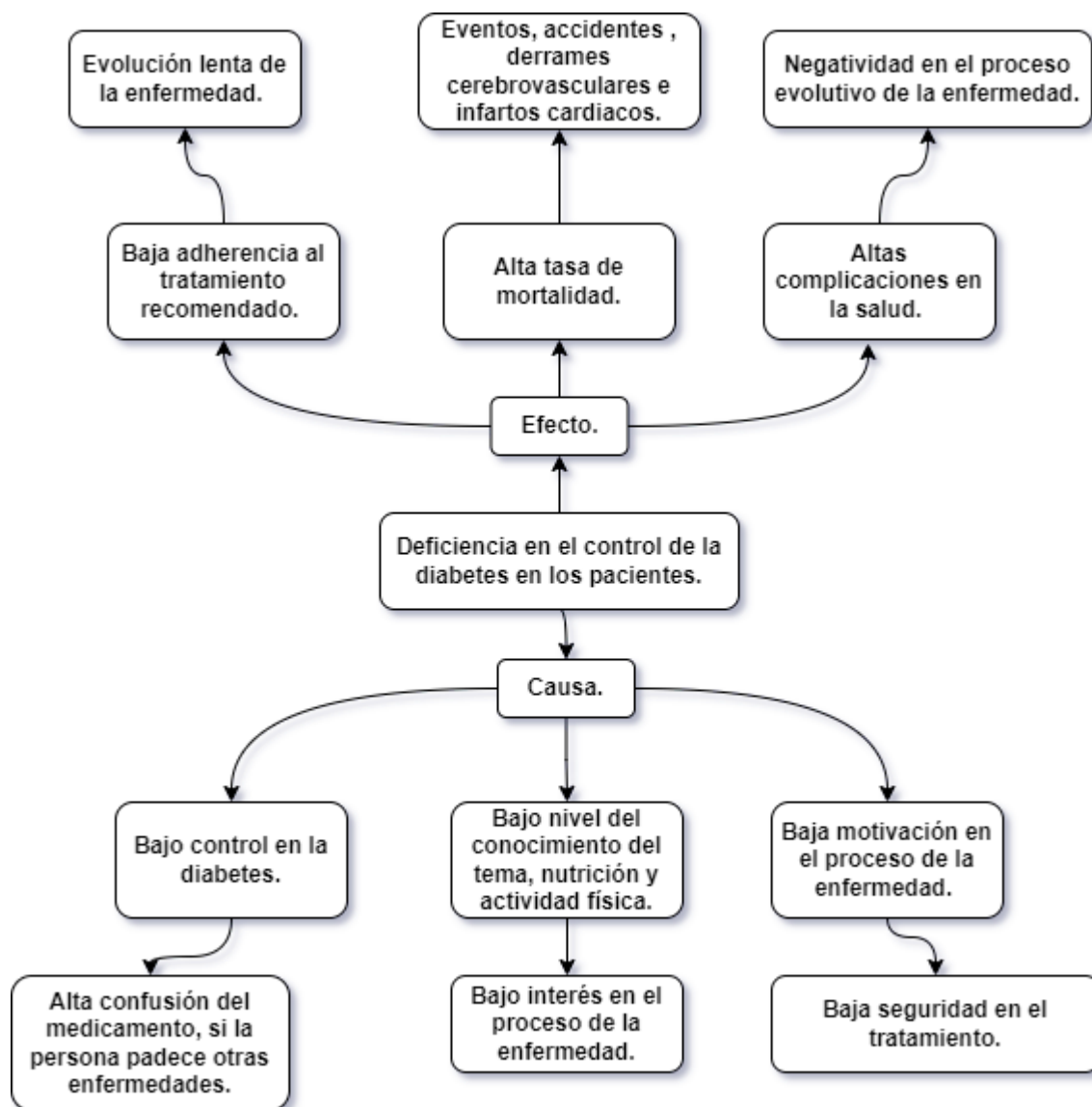
Identificación del problema y opciones de solución: En esta etapa se analiza la situación actual para establecer una visión del estado deseado y seleccionar las estrategias necesarias para alcanzarlo. Se realizan cuatro tipos de análisis: involucrados, problemas, objetivos y estrategias, los cuales se plasman en el árbol del problema, el árbol de objetivos y el árbol de la solución.

Etapas de planificación: Aquí, la concepción del proyecto se traduce en un plan operativo concreto para su implementación. La Matriz de marco lógico se desarrolla en esta fase, donde se definen y programan las actividades y recursos necesarios en un marco temporal específico.

9.1 Árbol del problema

Figura 30

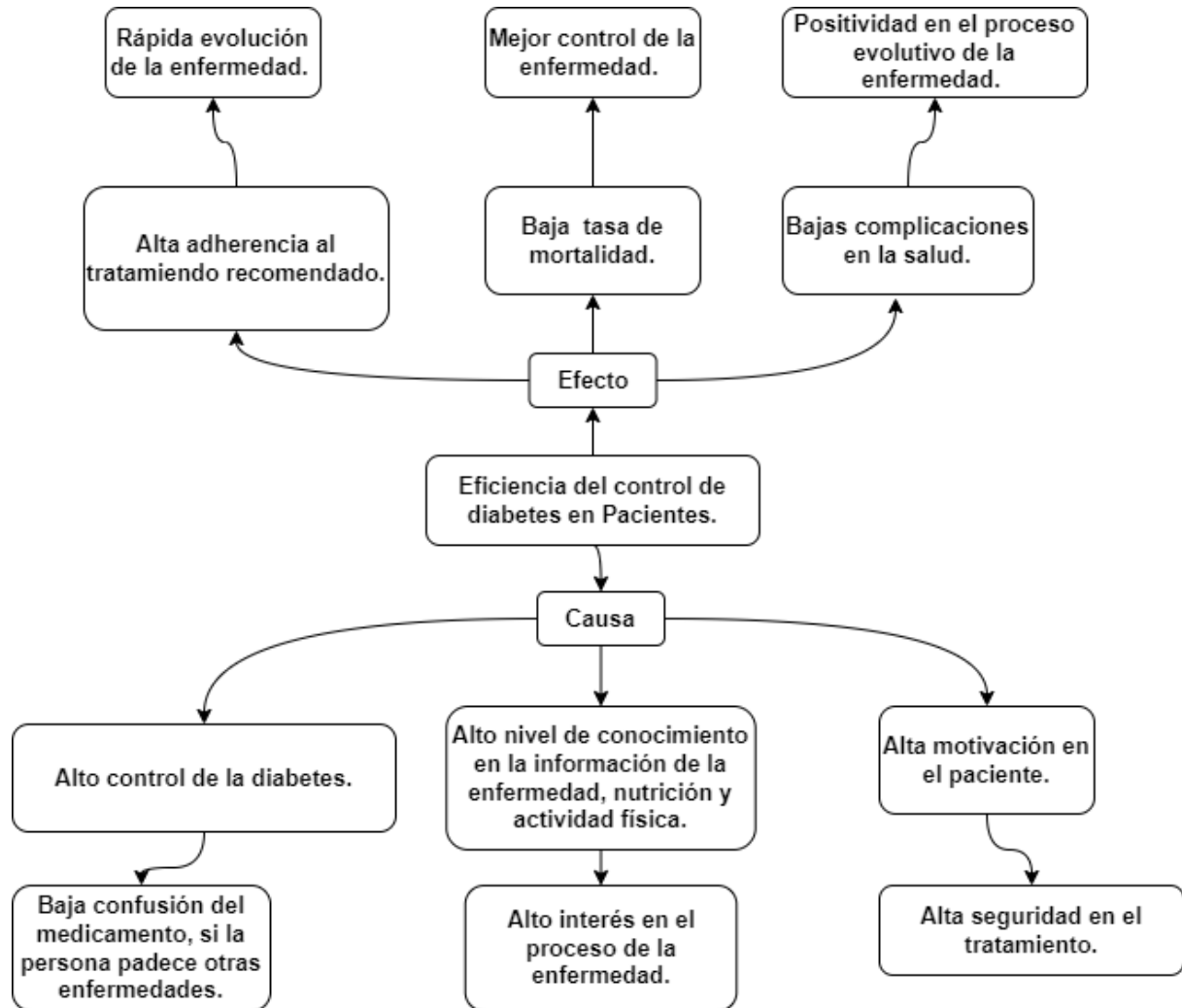
Árbol del Problema



9.2 Árbol de objetivo

Figura 31

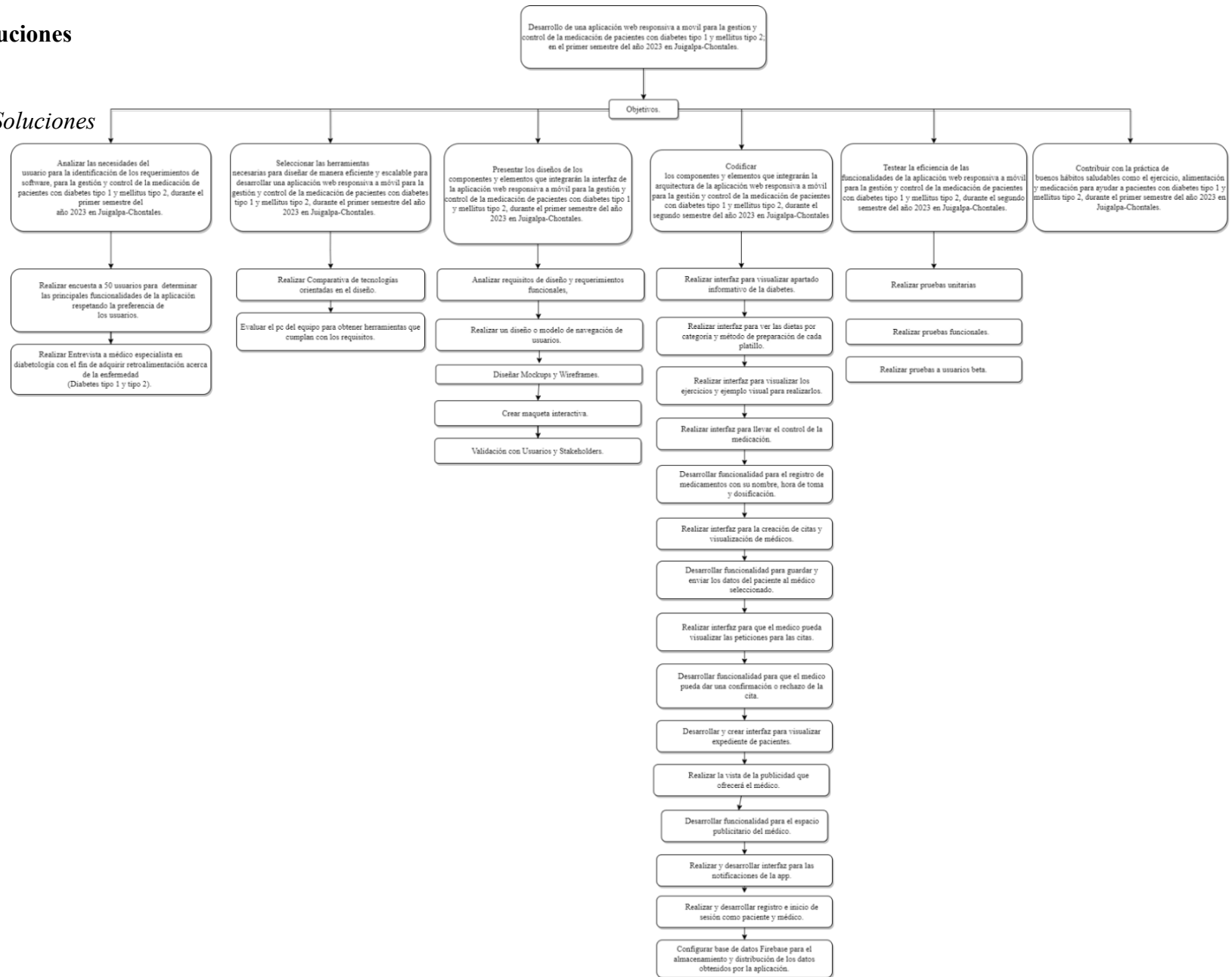
Árbol de Objetivos



9.3 Árbol de Soluciones

Figura 32

Árbol de Soluciones



9.4 Marco Lógico

Tabla 27

Marco Lógico

Resumen Narrativo de Objetivos	Descripción	Indicadores	Medios de Verificación	Supuestos
Fin	Garantizar un control efectivo y una evolución positiva de la enfermedad a través de una estricta adherencia al tratamiento.	Incrementar en un 85% la tasa de pacientes que siguen el tratamiento recomendado de acuerdo con las pautas médicas.	Estadísticas que Firebase facilita.	Acceso a la tecnología. Cumplimiento y compromiso en la adherencia al tratamiento.
Propósito	Desarrollar una aplicación web responsiva a móvil para la gestión y control de la medicación de pacientes con diabetes tipo 1 y mellitus tipo 2.	Desarrollar el 100% de la aplicación web responsiva a móvil durante el segundo semestre del año 2023. Un 80% de Comentarios positivos por parte de los usuarios en la eficiencia de la aplicación web responsiva a móvil debido a las funcionalidades existentes en esta.	Control de Versiones de GitHub. Tablero de Jira con tareas y Sprint realizados. Tablero de Trello llamado “Health Innovision” con las tareas realizadas exitosamente. Encuesta para los usuarios beta encargados de testear el software.	Acceso a dispositivos móviles. Conformidad en la aplicación por parte de los usuarios. Acceso a Internet.
Componentes	Analizar las necesidades del usuario para la identificación de los requerimientos de software.	Satisfacción del usuario final en un 90%. Eficiencia en el proceso de desarrollo en un 100%	Encuestas completadas por los usuarios para recopilar información sobre sus necesidades y expectativas. Informe de análisis de requerimientos terminado.	Acceso y Disposición de los Usuarios. Conciencia de la Satisfacción del Usuario. Participación de Usuarios
	Seleccionar las herramientas necesarias para	Disminución en un 50% la cantidad de errores y	Maqueta interactiva de Adobe XD bien	Respuesta Positiva de Usuarios Finales.

diseñar de manera eficiente y escalable la aplicación web responsiva a móvil.	reelaboración de diseños. Facilidad de la curva de aprendizaje de un 20% de la herramienta. Eficiencia en la selección de herramientas en un 90%.	definida en su entorno de desarrollo.	Adecuación de herramientas a futuros requisitos.
Presentar los diseños de los componentes y elementos que integrarán la interfaz de la aplicación web responsiva a móvil. Codificar los componentes y elementos que integrarán la arquitectura de la aplicación web responsiva a móvil.	Aprobación de los usuarios en un 90%. Realismo y fidelidad en 100%. Complejidad baja de un 20% al momento de codificar los componentes que integraran la aplicación. El 50% de los componentes a utilizar sean reutilizables. Satisfacción en un 95% por parte del equipo de desarrollo de software.	Prototipos Visuales. Mockups y Wireframes. Maquetas Interactivas. Control de versiones de GitHub. Módulos Funcionales codificados. Evaluación del rendimiento. Tablero de Jira.	Colaboración con usuarios finales. Compatibilidad con servicios externos. Eficiencia en el Uso de Recursos
Testear la eficiencia de las funcionalidades de la aplicación web responsiva a móvil.	Evaluación del grado de optimización del código para mejorar la eficiencia, velocidad y utilización de recursos de la aplicación, logrando un avance del 85% en este aspecto. Evaluación del nivel de seguridad	Resultados en pruebas de rendimiento. Pruebas en diferentes Dispositivos de usuarios Beta.	Disponibilidad de Usuarios Beta.

integrado en la codificación para salvaguardar los datos y la privacidad de los usuarios, alcanzando un nivel de 90% en términos de protección. Esperar una experiencia satisfactoria por parte de los usuarios de más del 85%.

Actividades	Componente 1			
	Componente 2			
	Realizar encuesta a 50 usuarios para determinar las principales funcionalidades de la aplicación respetando la preferencia de los usuarios. Realizar Entrevista a médico especialista en diabetología con el fin de adquirir retroalimentación acerca de la enfermedad (Diabetes tipo 1 y tipo 2).	Participación de los 50 usuarios encuestados en un 100%. Participación del médico en un 100%. Retroalimentación de la información obtenida en un 100%.	Resultados de la Encuesta. Resultado de la Entrevista. Informe y gráficos representativos en relación con los resultados.	Transparencia por parte de los encuestados. Experiencia amplia por parte del Diabetólogo en el ámbito de “Entrevistas”.
	Realizar Comparativa de tecnologías orientadas en el diseño. Evaluar el pc del equipo para obtener herramientas que cumplan con los requisitos.	Curva de aprendizaje valorada en un 20% Valoración positiva de equipo en un 100%	Informe de comparativas de tecnologías. Documentación de tecnologías seleccionadas.	Cambios en las características de las tecnologías. Rendimiento de Hardware.

Componente 3

Analizar requisitos de diseño y requerimientos funcionales.
Realizar un diseño o modelo de navegación de usuarios.
Diseñar Mockups y Wireframes.
Crear maqueta interactiva.
Validación con Usuarios y Stakeholders.

Coincidencia en un 100% respecto a los prototipos y los requisitos visuales.
Aprobación en un 100% por parte de los stakeholders de los mockups y wireframes presentados.
Funcionamiento en un 100% en la interacción y navegación logrado en la maqueta interactiva.

Informe de Requerimientos funcionales y no funcionales.
Diagrama de flujos de navegación.
Documentación de diseños de la aplicación.

Aceptación de los usuarios.

Componente 4

Realizar interfaz para visualizar apartado informativo de la diabetes.
Realizar interfaz para ver las dietas por categoría y método de preparación de cada platillo.
Realizar interfaz para visualizar los ejercicios y ejemplo visual para realizarlos.
Realizar interfaz para llevar el control de la medicación.
Desarrollar funcionalidad para el registro de medicamentos con su nombre, hora de toma y dosificación.

Cumplimiento y funcionalidad de la interfaz informativa de la diabetes realizada en un 100% al culminar el segundo Sprint.
Cumplimiento y funcionalidad de la interfaz de dietas realizada en un 100% al culminar el segundo Sprint.
Cumplimiento y funcionalidad de la interfaz para visualizar ejercicios en un 100% al culminar el segundo Sprint.
Cumplimiento y funcionalidad de la interfaz para el control de la medicación en un 100% al culminar el primer Sprint.

Interfaz que incluya información relevante, clara y precisa sobre la Diabetes tipo 1 y Mellitus tipo 2.
Interfaz que permita a los usuarios acceder a distintas categorías de dietas relacionadas con la enfermedad y que proporcione información detallada a cerca del platillo correspondiente y de preferencia del usuario.
Interfaz que muestre una variedad de ejercicios adecuados para personas con diabetes, junto con ejemplos visuales claros de cómo realizar cada ejercicio correctamente.

Disponibilidad de conexión a Internet para usuarios.
Colaboración con socios externos.
Cambio en las tendencias tecnológicas.
Cambios en las plataformas móviles.

Realizar interfaz para la creación de citas y visualización de médicos.	Funcionalidad del registro de medicamentos en un 100% al culminar el primer Sprint.	Interfaz completamente funcional respecto al registro, permitiendo también llevar un seguimiento correcto de los medicamentos, su dosificación y hora de la administración
Desarrollar funcionalidad para guardar y enviar los datos del paciente al médico seleccionado.	Cumplimiento y funcionalidad para la interfaz de creación de citas en un 100% al culminar el primer Sprint.	Pruebas para asegurar el funcionamiento de registro de los datos de la medicación.
Realizar interfaz para que el medico pueda visualizar las peticiones para las citas.	Funcionalidad para el guardado y envío de los formularios de manera exitosa alcanzando en un 100% al culminar el primer Sprint.	Interfaz que permita a los usuarios programar citas médicas y acceder a la información de los médicos disponibles.
Desarrollar funcionalidad para que el medico pueda dar una confirmación o rechazo de la cita.	Cumplimiento y funcionalidad de interfaz de peticiones de citas para el medico realizada en un 100% al culminar el primer Sprint.	Pruebas para asegurarse de que los datos del paciente se guarden correctamente y se envíen de manera segura al médico seleccionado.
Desarrollar y crear interfaz para visualizar expediente de pacientes.	Funcionamiento del 100% de la interfaz de aceptación o rechazo del médico al culminar en el primer Sprint.	Interfaz que permita a los médicos ver las solicitudes de citas pendientes por confirmar o rechazar.
Realizar la vista de la publicidad que ofrecerá el médico.	Funcionamiento del 100% de interfaz de visualización de expediente al culminar el tercer Sprint.	Pruebas para asegurarse que los médicos puedan confirmar o rechazar citas de manera efectiva.
Desarrollar interfaz para las notificaciones de la app.	Funcionamiento del 100% de la vista publicitaria del médico al culminar el segundo Sprint.	Asegurarse de que la interfaz de expediente muestre de manera organizada y detallada la información relevante de cada paciente.
Desarrollar registro e inicio de sesión como paciente y médico.	Funcionamiento correcto en un 100% sobre el espacio	
Configurar base de datos Firebase para el almacenamiento		

y distribución de los datos obtenidos por la aplicación.	<p>publicitario del médico al culminar el segundo Sprint.</p> <p>Funcionamiento del 100% de la interfaz de notificaciones de la app al culminar el primer Sprint.</p> <p>Registro e Inicio de sesión funcional en un 100% al culminar el cuarto Sprint.</p> <p>Configuración y adaptación de Firebase en la aplicación web responsiva a móvil en un 100% al culminar todos los Sprint.</p>	<p>Interfaz que muestre la publicidad del médico de manera adecuada y en un formato que sea comprensible para los usuarios.</p> <p>Pruebas para asegurarse de que la funcionalidad de espacio publicitario del médico funcione correctamente, mostrando la publicidad de manera correcta y apropiada.</p> <p>Interfaz de notificaciones que muestre alertas importantes de manera clara y oportuna.</p> <p>Pruebas exhaustivas para asegurar que tanto los pacientes como los médicos puedan registrarse e iniciar sesión de manera segura y sin problemas.</p> <p>Base de datos Firebase configurada con datos almacenados y distribuidos de manera correcta.</p>
--	--	--

Componente 5

Realizar pruebas unitarias.	Resultado con éxito en su totalidad con un 100% al culminar el quinto Sprint.	Registro de resultados de pruebas.	Calidad de red de los usuarios.
Realizar pruebas funcionales.	Funcionalidades testeadas con éxito en un 100% al culminar el quinto Sprint.	Registro de tiempo y respuesta de las funcionalidades.	Dispositivos y configuración de los usuarios.
Realizar pruebas a usuarios beta.		Retroalimentación y experiencia de los usuarios.	Influencia de entorno (factores ambientales o

Funcionamiento y
aprobación de los
usuarios en un 100%
al culminar el quinto
Sprint.

condiciones en la
señal).

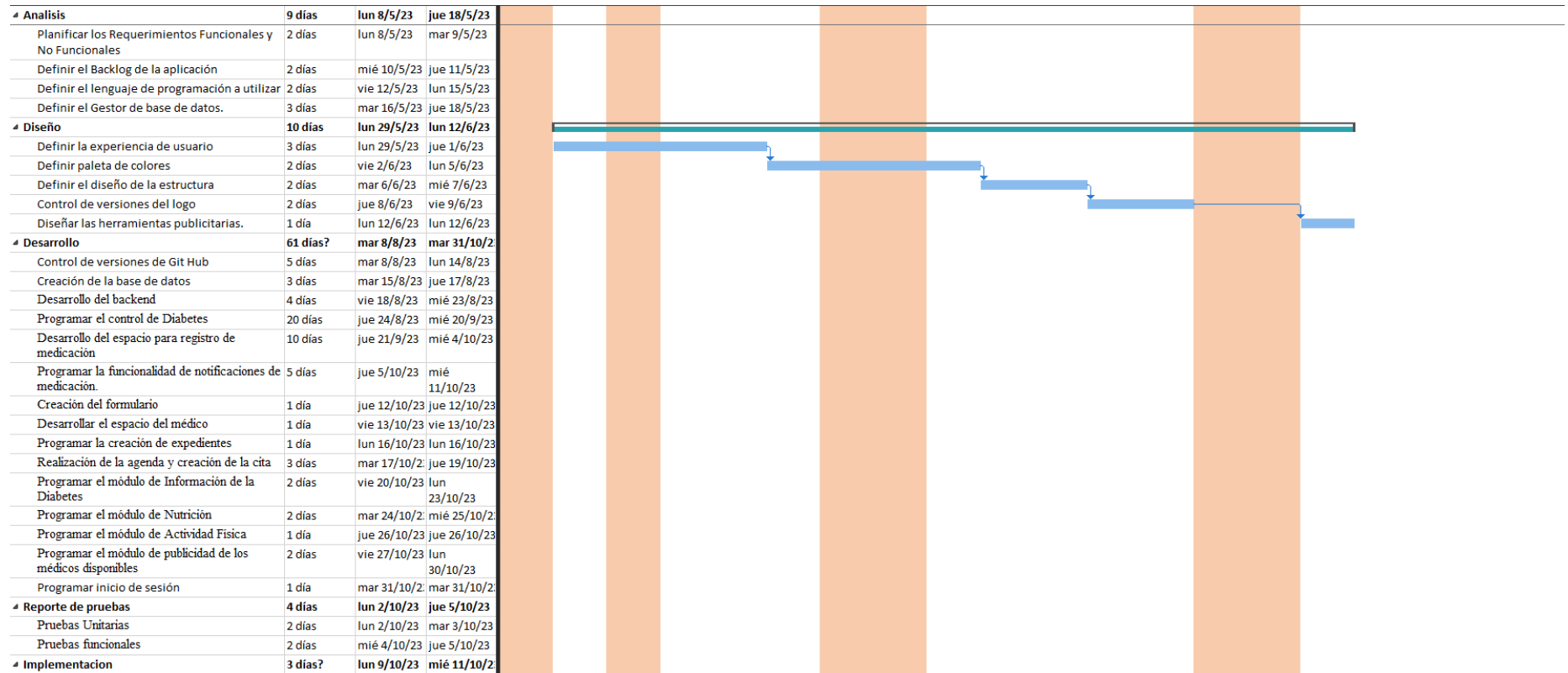
X. Cronograma de Actividades

Tabla 28

Cronograma de Actividades

Nombre de tarea	Duración	Comienzo	Fin
Análisis	9 días	lun 8/5/23	jue 18/5/23
Planificar los Requerimientos Funcionales y No Funcionales	2 días	lun 8/5/23	mar 9/5/23
Definir el Backlog de la aplicación	2 días	mié 10/5/23	jue 11/5/23
Definir el lenguaje de programación a utilizar	2 días	vie 12/5/23	lun 15/5/23
Definir el Gestor de base de datos.	3 días	mar 16/5/23	jue 18/5/23
Diseño	10 días	lun 29/5/23	lun 12/6/23
Definir la experiencia de usuario	3 días	lun 29/5/23	jue 1/6/23
Definir paleta de colores	2 días	vie 2/6/23	lun 5/6/23
Definir el diseño de la estructura	2 días	mar 6/6/23	mié 7/6/23
Control de versiones del logo	2 días	jue 8/6/23	vie 9/6/23
Diseñar las herramientas publicitarias.	1 día	lun 12/6/23	lun 12/6/23
Desarrollo	61 días	mar 8/8/23	mar 31/10/23
Control de versiones de Git Hub	5 días	mar 8/8/23	lun 14/8/23
Creación de la base de datos	3 días	mar 15/8/23	jue 17/8/23
Desarrollo del backend	4 días	vie 18/8/23	mié 23/8/23
Programar el control de Diabetes	20 días	jue 24/8/23	mié 20/9/23
Desarrollo del espacio para registro de medicación	10 días	jue 21/9/23	mié 4/10/23
Programar la funcionalidad de notificaciones de medicación.	5 días	jue 5/10/23	mié 11/10/23
Creación del formulario	1 día	jue 12/10/23	jue 12/10/23
Desarrollar el espacio del médico	1 día	vie 13/10/23	vie 13/10/23
Programar la creación de expedientes	1 día	lun 16/10/23	lun 16/10/23
Realización de la agenda y creación de la cita	3 días	mar 17/10/23	jue 19/10/23
Programar el módulo de Información de la Diabetes	2 días	vie 20/10/23	lun 23/10/23
Programar el módulo de Nutrición	2 días	mar 24/10/23	mié 25/10/23
Programar el módulo de Actividad Física	1 día	jue 26/10/23	jue 26/10/23
Programar el módulo de publicidad de los médicos disponibles	2 días	vie 27/10/23	lun 30/10/23
Programar inicio de sesión	1 día	mar 31/10/23	mar 31/10/23
Reporte de pruebas	4 días	lun 2/10/23	jue 5/10/23
Pruebas funcionales	2 días	mié 4/10/23	jue 5/10/23
Implementación	3 días	lun 9/10/23	mié 11/10/23
Capacitación a los usuarios en temas relacionados con la diabetes.	2 días	lun 9/10/23	mar 10/10/23

Lanzar al mercado la aplicación con una estrategia enfocada.	1 día	mié 11/10/23	mié 11/10/23
--	-------	--------------	--------------

Figura 33*Diagrama de GANTT*

Nota: Se presenta el cronograma detallado de las actividades ejecutadas para el desarrollo de la aplicación, junto con su correspondiente Diagrama de Gantt.

XI. Referencias

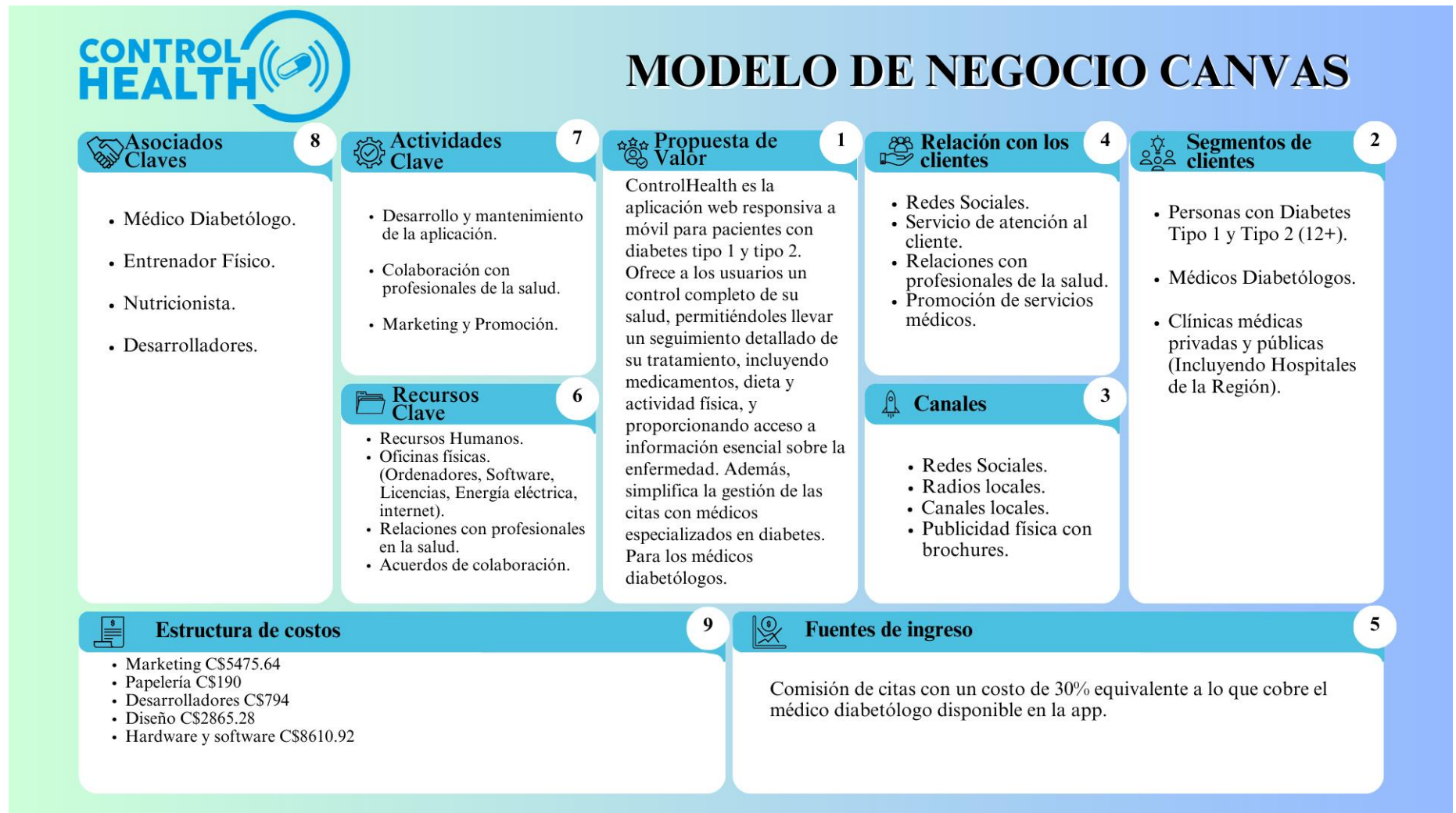
- Adobe. (24 de Enero de 2023). *Descubra Adobe Illustrator*. <https://acortar.link/wecO05>
- Adobe. (03 de Marzo de 2023). *Guía del usuario de Photoshop Elements*. <https://acortar.link/ysiw0r>
- Adobe. (17 de Enero de 2023). *Mejorar el rendimiento de Illustrator*. <https://acortar.link/rppCUv>
- Adobe. (24 de Mayo de 2023). *Sugerencias y técnicas para ejecutar Photoshop*. <https://acortar.link/1KqNFQ>
- Alba. (29 de Octubre de 2019). *Evolución de Illustrator*. <https://acortar.link/pupS4m>
- Amazon. (1 de Diciembre de 2023). *Python y sus beneficios*. <https://acortar.link/YHrB6F>
- AppMaster. (26 de Abril de 2023). *Kotlin: La alternativa a Java*. <https://acortar.link/CB2ArH>
- Avantgeo. (8 de Enero de 2023). *SQLite. Base de datos para movilidad*. <https://acortar.link/BP2KwG>
- Bolufér Gil, M. (s.f.). *Análisis de las Ventajas y Desventajas de Adobe Photoshop*. <https://acortar.link/A0c7VJ>
- Caronea, E. (30 de Marzo de 2022). *¿Qué es Kotlin y para qué sirve?*. <https://acortar.link/5wNf8Y>
- CompuHoy. (10 de Enero de 2020). *¿Se puede utilizar Python para aplicaciones de Android?*. <https://acortar.link/ZlkslK>
- CORELDRAW. (2021). *GUÍA DEL USUARIO*. <https://acortar.link/N7HKQe>
- CorelDraw. (2023). *CorelDRAW Graphics Suite para instituciones educativas*. <https://acortar.link/tiwR6a>
- Emprendy. (2023). *React desde cero*. <https://acortar.link/q4BMk2>
- Firebase. (2023). *Documentación y contenido de aprendizaje*. <https://acortar.link/wPKHnJ>
- Firebase. (2023). *Planes de precios*. <https://acortar.link/yelObI>
- González, R. (05 de Marzo de 2021). *¿Qué es Kotlin? El nuevo lenguaje de programación de Android que necesitas conocer*. <https://acortar.link/2sIQJO>
- Grupo Digital SDi. (26 de Julio de 2018). *Python, un lenguaje de programación con mucho presente y más futuro*. <https://acortar.link/SgzRUG>
- Kinsta. (19 de Diciembre de 2022). *¿Qué es React.js? Un Vistazo a la Popular Biblioteca de JavaScript*. <https://acortar.link/KoSaME>
- Kinsta. (19 de Diciembre de 2022). *Un Vistazo a la Popular Biblioteca de JavaScript*. <https://acortar.link/KoSaME>
- MINSA. (2022). *Mapa Nacional de la Salud en Nicaragua*. <https://acortar.link/eX9rKm>

- Mongo DB. (2023). *Desarrolle más rápidamente, desarrolle de forma más inteligente*. <https://acortar.link/0HRSU6>
- MongoDB. (2023). *¿Qué es MongoDB?* <https://acortar.link/Ksi7A3>
- MongoDB. (2023). *MongoDB Pricing*. <https://acortar.link/mr418F>
- React. (2023). *Requerimientos del entorno de JavaScript*. <https://acortar.link/n5rXuq>
- Rondón, M. D. (2006). *Potencia y versatilidad de Photoshop*. <https://acortar.link/TZjM00>
- SQLite. (16 de Mayo de 2023). *Documentacion de SQLite*. <https://acortar.link/NBkW4D>
- The Data Schools. (2023). *Python, lenguaje de programación interpretado, de alto nivel y multipropósito*. <https://acortar.link/noMM6T>
- Zamudio, F. (13 de Noviembre de 2020). *ReactJs/Redux: Una estructura para mejorar la escalabilidad*. <https://acortar.link/7JtXPC>

XII. ANEXOS

Anexo A.

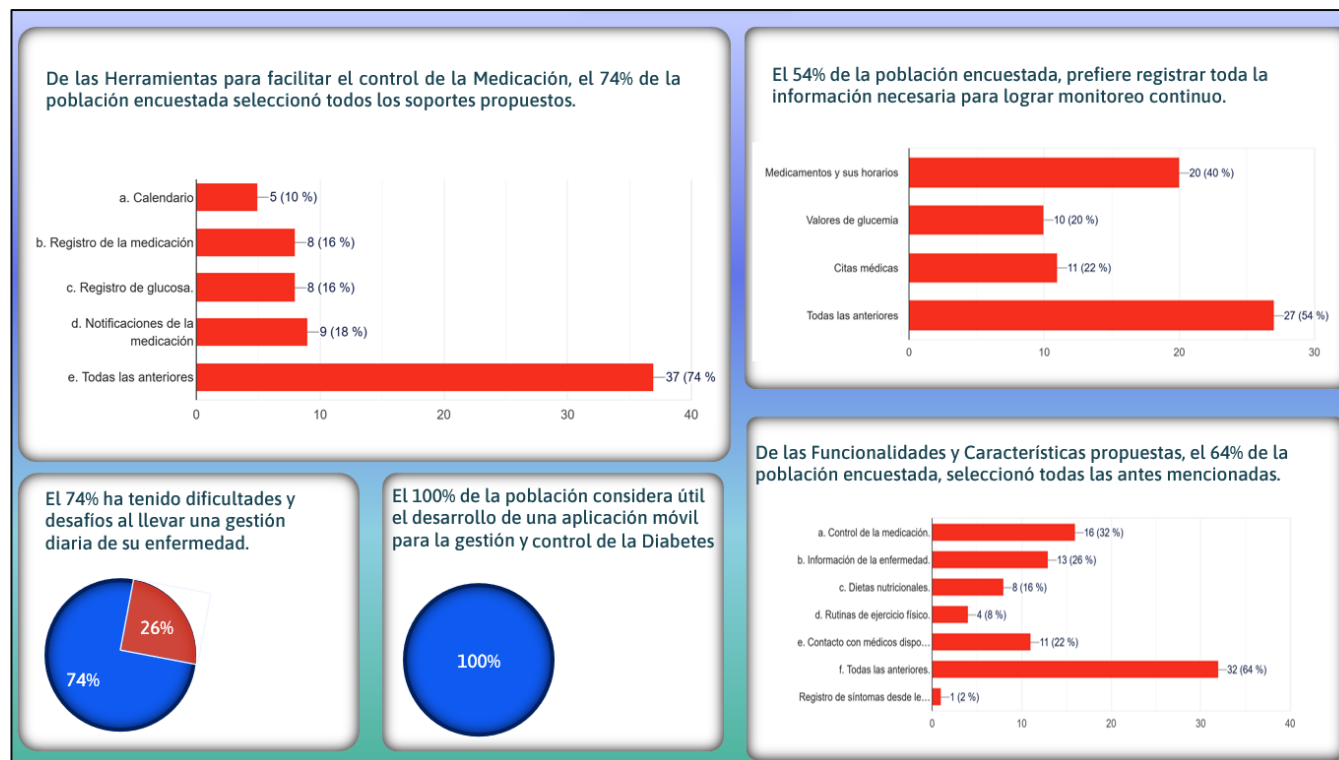
Modelo de Negocio CANVAS



Nota: Elaboración propia.

Anexo B

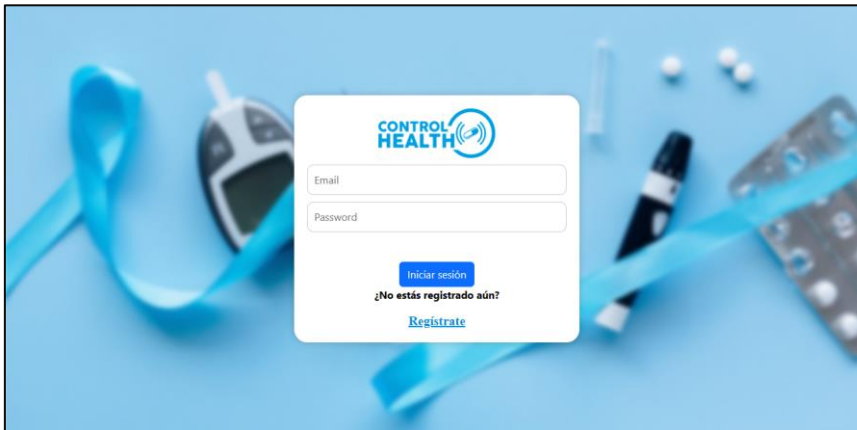
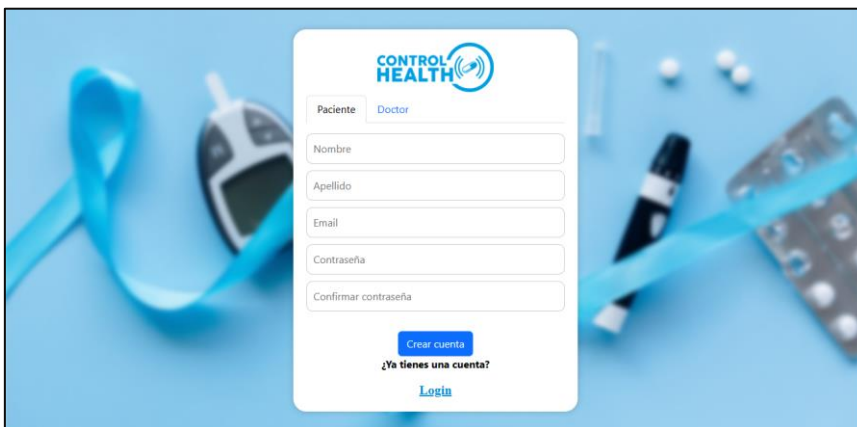
Encuesta para obtención de requerimientos de usuarios



Nota: Elaboración propia.

Anexo C

Aplicación web responsiva a móvil



Nota: Aplicación “Control Health” en la vista web y móvil, presentado primeramente la vista de Registro e Inicio de sesión.

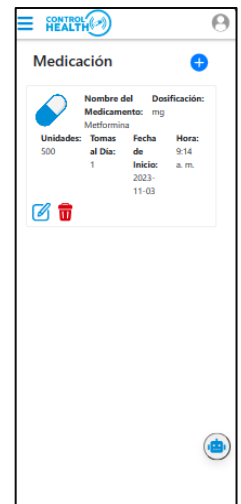


Nota: Vista principal de la aplicación donde el usuario puede encontrar los módulos interactivos e informativos de Alimentación, Ejercicio Físico e Información general de la Diabetes.



Medicación

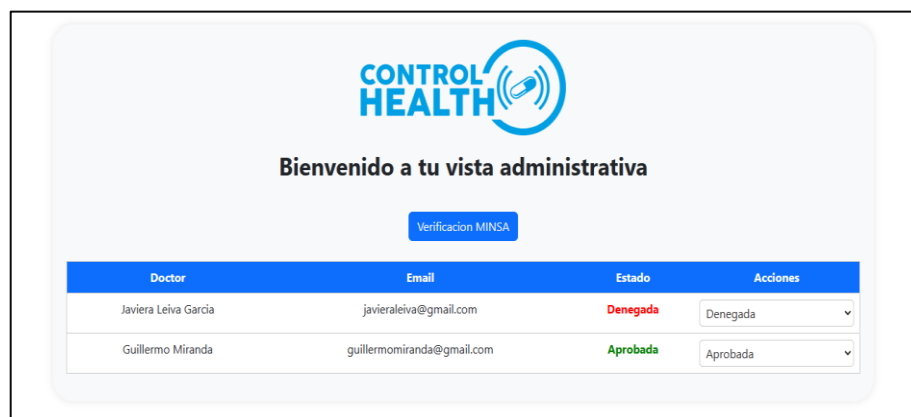
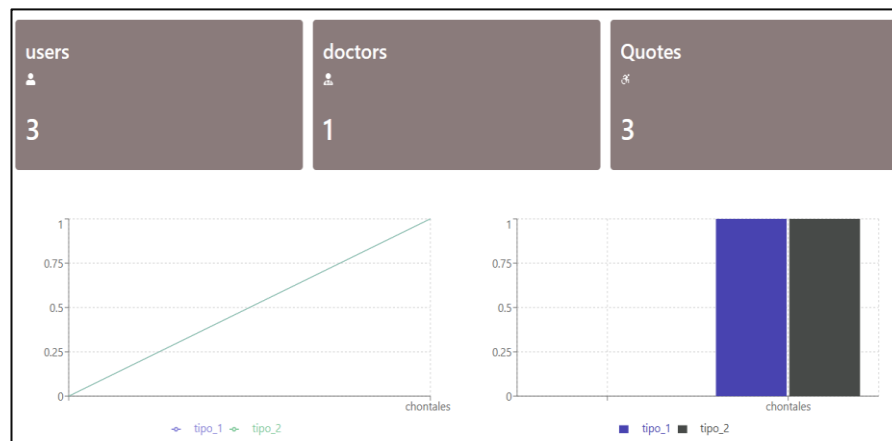
Nombre del Medicamento:	Dosificación:	Unidades:	Tomas al Día:	Fecha de Inicio:	Hora:	
Metformina	mg	500	1	2023-11-03	9:14 a. m.	 



Medicación

Nombre del Medicamento:	Dosificación:	Unidades:	Tomas al Día:	Fecha de Inicio:	Hora:
Metformina	mg	500	1	2023-11-03	9:14 a. m.

Nota: Módulo de “Medicación” donde el usuario “Paciente” guardara toda su medicación, incluyendo la de otros padecimientos para recibir sus notificaciones.



Nota: La vista administrativa permite al administrador acceder a las estadísticas de usuarios, gestionar perfiles de pacientes con capacidad de edición y eliminación, y aprobar/rechazar solicitudes de médicos, incluyendo verificación del código MINSA.

Anexo D

Experiencia vivida en Hackathon Nicaragua IA 2023

Como equipo "Health Innovision", deseamos expresar nuestro más sincero agradecimiento. Nuestra gratitud se extiende principalmente a Dios, a nuestra familia y a cada uno de nuestros respetados profesores de la UNAN Managua FAREM-Chontales. Recordaremos a estos grandes educadores con profundo respeto y admiración, ya que no solo compartieron conocimientos con nosotros, sino que también se convirtieron en grandes amigos y ejemplos a seguir que han dejado una huella significativa en nuestras vidas como futuros profesionales. ¡Este logro ha sido posible gracias a nuestro esfuerzo conjunto y nos enorgullece formar parte de la Carrera de Ingeniería en Sistemas de Información!







PLAN DE PRUEBAS

APLICACIÓN PARA CONTROL Y GESTIÓN DE DIABETES.

Juigalpa-Chontales, diciembre 2023

Bitácora de cambios

Fecha	Modificado por	Versión	Descripción
25/10/2023	Br. Donald Alvarez	1.0	Definición del plan de pruebas
25/10/2023	Br. Javiera Leiva	1.1	Características para probar
25/10/2023	Br. Brando Estrada	1.2	Informe de pruebas del Sprint 1
26/10/2023	Br. Donald Alvarez	1.3	Informe de pruebas del Sprint 2
26/10/2023	Br. Javiera Leiva	1.4	Casos de prueba Sprint 3
26/10/2023	Br. Brando Estrada	1.5	Informe de pruebas del Sprint 3
26/10/2023	Br. Donald Alvarez	1.6	Casos de prueba del Sprint 4
26/10/2023	Br. Javiera Leiva	1.7	Informe de pruebas del Sprint 4
27/10/2023	Br. Brando Estrada	1.8	Casos de prueba del Sprint 5
27/10/2023	Br. Donald Alvarez	1.9	Informe de pruebas del Sprint 5

Formato preliminar del documento

Título:	PLAN DE PRUEBAS APLICACIÓN PARA CONTROL Y GESTIÓN DE LA DIABETES		
ID:	PLAN_001_Control Health		
Fecha elaboración:	25/10/2023		
Sumario	El propósito de este documento es verificar las funciones y procesos de los diversos módulos y componentes de la aplicación, identificar y corregir cualquier fallo o error que pueda surgir durante las pruebas, y validar si el sistema cumple con los requisitos que definen su funcionamiento completo.		
Palabras Claves:	Pruebas, fallos, componentes, aplicación, requerimientos, funcionalidad.		
Formato:	.pdf	Lenguaje:	Español
Dependencia:	Health Innovision		
Versión:	1.0	Estado:	Aprobado
Autor (es):	Br. Javiera Leiva. Br. Donald Alvarez. Br. Brando Estrada.		
Revisó:	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.	Firmas:	
Aprobó:	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.		
Información Adicional:	Los componentes y las relaciones de cada módulo de la aplicación definen los principales apartados de Dietas nutricionales, Rutinas de ejercicio, Control de la medicación, espacio del médico, ChatBot y autenticación.		

Tabla de contenido

CONTENIDO

I. Introducción	1
1.1 Objetivo general	1
1.2 Estrategia de pruebas.....	1
1.3 Alcance	1
1.4 Propósito	1
II. Entregables.....	2
2.1 Documentación a entregar.....	2
III. Características a ser probadas	3
3.1 Características a probar del Sprint 1	3
3.2 Características a probar del Sprint 2	4
3.3 Características a ser probadas del Sprint 3	5
3.4 Características a ser probadas del Sprint 4	5
IV. Características a no ser probadas	6
4.1 Características a no ser probadas del Sprint 1	6
4.2 Características a no ser probadas del Sprint 2	6
V. Criterios de aprobación y fallo	7
VI. Criterios de suspensión y reanudación	8
VII. Tareas de las pruebas	9
VIII. Necesidades ambientales.....	10
8.1 Hardware	10
8.2 Planeación de costos.....	11
8.3 SUT (Sistema Bajo Pruebas)	12
IX. Informe de pruebas	13

9.1	Prueba Sprint 1.....	13
9.2	Pruebas Sprint 2.....	17
9.3	Prueba Sprint 3.....	23
9.4	Prueba Sprint 4.....	27
X.	Planificación SCRUM	30
10.1	Product Backlog	30
10.2	Product Backlog organizado por Sprint	31
10.3	Sprint Backlog	32
10.3.1	Sprint 1.....	32
10.3.2	Casos de prueba del Sprint 1	33
10.3.3	Sprint 2.....	34
10.3.4	Casos de prueba del Sprint 2	36
10.3.5	Sprint 3.....	41
10.3.6	Casos de prueba Sprint 3.....	42
10.3.7	Sprint 4.....	45
10.3.8	Casos de prueba Sprint 4	46

Índice de tablas

Tabla 1 Documentación a entregar.....	2
Tabla 2 Características a ser probadas del sprint 1	3
Tabla 3 Características a ser probadas del Sprint 2	4
Tabla 4 Características a ser probadas del Sprint 3	5
Tabla 5 Características a ser probadas del Sprint 4	5
Tabla 6 Características a no ser probadas del Sprint 1	6
Tabla 7 Características a no ser probadas del Sprint 2	6
Tabla 8 Criterios de aprobación y fallo.....	7
Tabla 9 Criterios de suspensión y reanudación	8
Tabla 10 Tareas de las Pruebas	9
Tabla 11 Necesidades ambientales de Hardware.....	10
Tabla 12 Planeación de costos	11
Tabla 13 Sistema bajo Pruebas	12
Tabla 14 Product Backlog	30
Tabla 15 Product Backlog organizado por Sprint	31
Tabla 16 Sprint 1	32
Tabla 17 Sprint 2	35
Tabla 18 Sprint 3	41
Tabla 19 Sprint 4.....	45

Índice de figuras

Figura 1 Script register.test.js	13
Figura 2 Script dietprueba.test.js	14
Figura 3 Script filtardiet.test.js.....	15
Figura 4 Resultado de las pruebas Sprint 1	16
Figura 5 Script medication.test.js.....	17
Figura 6 Script quotesolicitud.test.js.....	18
Figura 7 Script agendarctias.test.js	19
Figura 8 Script registermedico.test.js	20
Figura 9 Script chatbotprueba.test.js.....	21
Figura 10 Resultado de las pruebas Sprint 2	22
Figura 11 Script Medication.test.js.....	23
Figura 12 Script-calendariocitas.test.js.....	24
Figura 13 Script homedrprueba.test.js	25
Figura 14 Resultado de la prueba Sprint 3.....	26
Figura 15 Script-admindoctors.test.js.....	27
Figura 16 Script-admin.test.js	28
Figura 17 Resultado de la prueba Sprint 4.....	29

I. Introducción

El siguiente documento detalla el Plan de Pruebas para la aplicación de Control y Gestión de la Diabetes, denominada Control Health. Este plan abarca los aspectos relacionados con los entregables, documentación, características sujetas a pruebas, exclusiones de pruebas, criterios de aprobación y rechazo, así como los criterios de suspensión y reanudación. Además, se describen las tareas relacionadas con las pruebas y las necesidades ambientales asociadas a este proceso.

1.1 Objetivo general

Elaborar un Plan de Pruebas exhaustivo y minucioso con el propósito de validar la correcta operación de los módulos y componentes de la aplicación Control Health, destinada al control y gestión de la Diabetes.

1.2 Estrategia de pruebas

Realizar pruebas de funcionalidad en los componentes de la aplicación con el fin de identificar posibles errores y defectos que requieran corrección.

1.3 Alcance

Este plan de pruebas se transforma en un manual que orienta de manera estructurada todas las actividades llevadas a cabo en el proceso de pruebas a lo largo del desarrollo del proyecto de la aplicación de control y gestión de Diabetes.

1.4 Propósito

Este documento tiene como finalidad establecer las técnicas, herramientas y actividades necesarias para llevar a cabo la ejecución y validación del plan de pruebas. Incluye la asignación de responsabilidades para cada tarea, la asignación de recursos pertinentes, y los requisitos esenciales que deben ser considerados a fin de garantizar el cumplimiento de los requerimientos en el contexto del desarrollo del proyecto de la aplicación de control y gestión de diabetes.

II. Entregables

2.1 Documentación a entregar

Tabla 1

Documentación a entregar

Documento	Persona que entrega	Persona que recibe	Fecha planeada	Fecha de entrega
Planificación Scrum	Br. Donald Alvarez	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.	04/09/2023	30/10/2023
Casos de Pruebas y resultado del Sprint 1	Br. Brando Estrada	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.	25/10/2023	25/10/2023
Casos de Pruebas y resultado del Sprint 2	Br. Donald Alvarez	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.	26/10/2023	26/10/2023
Casos de Pruebas y resultado del Sprint 3	Br. Javiera Leiva	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.	26/10/2023	26/10/2023
Casos de Pruebas y resultado del Sprint 4	Health Innovision	M.Sc. Saira Urbina Cienfuegos. M.Sc Miriam Téllez. M.Sc Carmen Téllez.	27/10/2023	27/10/2023

III. Características a ser probadas

3.1 Características a probar del Sprint 1

Tabla 2

Características a ser probadas del sprint 1

Característica	Descripción	Módulo	Código de prueba
Funcionalidad de Registro del usuario paciente.	Característica la aplicación que permite a los individuos crear cuentas o perfiles personales dentro de Control Health diseñada para la gestión de información médica y atención de salud.	-Componente de Registro de Usuario. -Componente Redux UserAction y reducer User.	PM01
Funcionalidad para visualizar una Dieta Nutricional.	Se mostrará un modal en el cual se vera la información detallada del platillo.	Componente de Dietas.	PP04
Funcionalidad para filtrar dietas.	Se mostrarán tres botones llamados, Desayunos, Almuerzos y Cenas que filtrarán los alimentos.	Componente de Dietas.	PP04.1

3.2 Características a probar del Sprint 2

Tabla 3

Características a ser probadas del Sprint 2

Característica	Descripción	Módulo	Código de prueba
Funcionalidad del Control de la medicación.	Se mostrará los medicamentos agregados en una CARD al ser agregados en un formulario que se encuentra dentro de un modal.	Componente de Medicación.	PP01
Funcionalidad de Creación de citas.	Se mostrará una solicitud de cita gestionada por el usuario.	Componente de Citas.	PP07
Funcionalidad de Agendar Citas.	Se enviará la solicitud de la cita, mostrándole al médico la opción de aceptar o rechazar cita.	Componente de Citas Médicas.	PM04
Funcionalidad de registro de usuario médico.	Característica de la aplicación que permite registrar médicos con su código Minsa.	-Componente de Registro de Usuario.	PM06
Funcionalidad de ChatBot.	Característica que permite un flujo de interacción para el usuario de mano de la IA.	-Componente Redux DoctorAction y reducer Doctor. Componente de ChatBot.	PP08

3.3 Características a ser probadas del Sprint 3

Tabla 4

Características a ser probadas del Sprint 3

Característica	Descripción	Módulo	Código de prueba
Funcionalidad que permita Notificaciones dentro de la app	Se mostrarán notificaciones en la toma de los medicamentos.	-Componente de Medicación.	PU02
Funcionalidad del Calendario	Se mostrará un calendario con las citas agendadas.	Componente de Citas Médicas.	PP07
Funcionalidad del Espacio del médico.	Se mostrará una vista de los módulos establecidos del médico con sus rutas funcionales.	Componente constants.	PM03

3.4 Características a ser probadas del Sprint 4

Tabla 5

Características a ser probadas del Sprint 4

Característica	Descripción	Módulo	Código de prueba
Funcionalidad del CRUD Administrativo.	Se mostrará una vista administrativa con las opciones de validar a los médicos.	-Componente CRUD -Componente Administrador.	PA01
Funcionalidad de Estadísticas de los módulos.	Se mostrarán gráficas de las estadísticas de los usuarios registrados y funcionalidades utilizadas dentro de la app.	Componente Estadísticas.	PA02

IV. Características a no ser probadas

4.1 Características a no ser probadas del Sprint

Tabla 6

Características a no ser probadas del Sprint 1

Característica	Descripción	Justificación	Riesgos
Espacios de Actividad Física	Mostrar rutinas de ejercicio categorizadas en Principiante, Intermedio y Avanzado.	Este componente cuenta con una dependencia y estructura similar al de Dietas Nutricionales.	Impedir las funcionalidades de los modales de manera correcta y no poder visualizar detalles del ejercicio.

4.2 Características a no ser probadas del Sprint 2

Tabla 7

Características a no ser probadas del Sprint 2

Característica	Descripción	Justificación	Riesgos
Espacio Informativo sobre la Diabetes	Mostrar información relevante sobre la enfermedad dentro de distintas CARDS en el Home del paciente.	Este componente dependerá de una DATA y de datos traídos de Firestore.	Impedir que la DATA y Firestore no cargue la información.

V. Criterios de aprobación y fallo

Tabla 8

Criterios de aprobación y fallo

Criterio	Descripción	Aprobación	Fallo
P01	Evaluar el estado del store de la aplicación	El estado retornado debe ser un objeto o un arreglo iterable después de haber llamado al action y reducir del store	El estado retornado es un objeto o un arreglo vacío
P02	Verificar que un medicamento pueda ser añadido de manera exitosa.	El estado retornado debe ser un objeto que contenga los elementos de la medicación añadidos de manera exitosa.	El estado retornado es un objeto o un arreglo vacío.
P03	Validar que las dietas puedan ser visualizadas.	El modal debe mostrar la información que almacenan las dietas	El modal no se abra o no muestre ninguna información.
P04	Validar que un usuario puede registrarse e iniciar sesión de manera correcta.	El estado retornado debe ser un objeto que contenga los datos del usuario agregados a la colección.	El estado muestre un error de conexión y no se muestre un objeto con los datos.
P05	Verificar que las citas puedan ser realizadas y agendadas de manera correcta.	El estado retornado debe ser un objeto que contenga los datos de la cita añadidos de manera exitosa.	El estado retornado es un objeto o un arreglo vacío.
P06	Validar que El ChatBot siga su flujo adecuado a la hora de una consulta.	El algoritmo del flujo debe responder de una manera lógica.	El algoritmo falle y el ChatBot de respuestas incoherentes.

VI. Criterios de suspensión y reanudación

Tabla 9

Criterios de suspensión y reanudación

Criterio de Suspensión	Criterio de Reanudación
Cuando el 80% de las pruebas no cumplen con los criterios de aceptación (fallo).	Detectar errores en la ejecución de las pruebas.
Después de la instalación y configuración del sistema, se evidencia que el ambiente de pruebas no es lo suficientemente estable para la ejecución de las pruebas.	Volver a configurar el ambiente de pruebas y comprobar que este es óptimo.
Los resultados generados por las pruebas de automatización no contengan las mínimas variables que permitan un análisis acertado de cada una de las pruebas realizadas.	Cambiar el método con el que se están realizando las pruebas.
La aplicación no cumple con los requerimientos funcionales especificados.	Concluir el desarrollo de las funcionalidades.

VII. Tareas de las pruebas

7.1 Tareas de las pruebas

Tabla 10

Tareas de las Pruebas

Tarea	Descripción	Fecha Inicio	Fecha fin	Duración (hrs)	Responsable	Rol
Definir casos de prueba	Crear los casos de prueba de caja negra y caja blanca siguiendo la plantilla elegida.	25/10/2023	25/10/2023	24horas	Donald Alvarez	Desarrollador
					Javiera Leiva	Analista
					Brando Estrada	Diseñador
Prueba de componentes	Verificar la renderización de los componentes de la aplicación.	25/10/2023	25/10/2023	24horas	Brando Estrada	Diseñador
					Javiera Leiva	Analista
Prueba de funciones	Examinar la ejecución de las funciones críticas de los módulos y componentes.	25/10/2023	25/10/2023	24horas	Brando Estrada	Diseñador
Preparar el entorno de ejecución de pruebas	Realizar la instalación y configuración de las herramientas elegidas para llevar a cabo las pruebas.	25/10/2023	25/10/2023	24horas	Donald Alvarez	Desarrollador
Definir las suites de prueba	Elaborar la estructura organizativa de los scripts de prueba.	25/10/2023	25/10/2023	24horas	Brando Estrada	Diseñador
					Javiera Leiva	Analista

Codificar los scripts de prueba	Crear pruebas utilizando el framework de JavaScript llamado Jets y Frontend terminado,	26/10/2023	26/10/2023	24horas	Donald Alvarez	Desarrollador
Ejecución de los scripts de prueba	Ejecutar los comandos de prueba en la terminal.	26/10/2023	26/10/2023	24horas	Donald Alvarez	Desarrollador
Corrección de fallas encontradas	Corregir errores encontrados en las funcionalidades	26/10/2023	26/10/2023	24horas	Donald Alvarez	Desarrollador

VIII. Necesidades ambientales

8.1 Hardware

Tabla 11

Necesidades ambientales de Hardware

Dispositivo	Marca	Características	¿tenemos el equipo?
Computador portátil	HP-Laptop14-cf2xxx	Pantalla 14 pulgadas 16GB RAM Intel(R) Core (TM) i5-10210U	Si
Teléfono móvil	Samsung Note20	Pantalla dinámica AMOLED de 6.9 pulgadas 12GB RAM Resolución Quad HD+	Si
Teléfono móvil	Samsung A31	Pantalla Super AMOLED de 6.4 pulgadas 4GB RAM Resolución Full HD+.	Si

8.2 Planeación de costos.

Tabla 12

Planeación de costos

Id	Tipo de recurso	Forma de adquisición	cantidad	Costo de Recurso unidad	Día(s)	Total
01	Tester	Contratación del servicio	-	\$ 10.13	18	\$ 182.42
02	Internet	Contratación del servicio	-	\$ 40.99	30	\$ 40.99
03	Energía eléctrica	Contratación del servicio	-	\$ 36.61	30	\$ 36.61
04	Computadora	Compra	1	\$ 300.00	-	\$ 300.00
05	Mobiliario de oficina	Compra	-	\$ 130.40	-	\$ 130.40
06	Papelería	Compra	-	\$ 10.05	-	\$ 10.05
Total						\$700.47

8.3 SUT (Sistema Bajo Pruebas)

Tabla 13

Sistema bajo Pruebas

Modulo	¿dependencia entre módulos?	Requerimientos
Redux store	Si	Incluye el árbol completo del estado de la aplicación y devuelve un objeto con propiedades y métodos.
Autenticación	Si	Contiene la instancia del usuario y devuelve el perfil del usuario en forma de objeto.
Pacientes	Si	Devuelve los componentes relacionados con el perfil del paciente, que incluyen: Dietas nutricionales, Rutinas, Creación de citas y Control de la medicación.
Médicos	Si	Devuelve los componentes relacionados con el perfil del médico, que son: Expedientes y Citas.
Administrador	Si	Devuelve los componentes relacionados con el perfil de usuario del administrador, que incluyen: CRUD de usuarios registrados y Estadísticas del uso de la aplicación.

IX. Informe de pruebas

9.1 Prueba Sprint 1.

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo UserRegister de la aplicación **Según el caso de prueba PM01.**

Figura 1

Script register.test.js

```
import React from 'react';
import { render } from '@testing-library/react';
import UserRegister from '../Components/UserRegister/UserRegister';

test('Renderiza el componente UserRegister sin errores', () => {
  const formik = {
    handleSubmit: jest.fn(),
    handleChange: jest.fn(),
    handleBlur: jest.fn(),
    values: {},
    touched: {},
    errors: {},
  };

  const { getByText } = render(<UserRegister formik={formik} />);
  const createAccountButton = getByText('Crear cuenta');
  expect(createAccountButton).toBeInTheDocument();
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo Diet de la aplicación **Según el caso de prueba PP04.**

Figura 2

Script dietprueba.test.js

```
import React from 'react';
import { render } from '@testing-library/react';
import Diet from '../Views/Diet/Diet';

describe('Diet Component', () => {
  it('renders without crashing', () => {
    render(<Diet />);
  });

  it('displays the category buttons', () => {
    const { getByText } = render(<Diet />);

    expect(getByText('Mostrar Todos')).toBeInTheDocument();
  });

  it('displays item cards', () => {
    const { getAllByRole } = render(<Diet />);

    const itemCards = getAllByRole('button');
    expect(itemCards.length).toBeGreaterThan(0);
  });
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre la funcionalidad de filtrar dietas nutricionales de la aplicación **Según el caso de prueba PP04.1.**

Figura 3

Script filtardiet.test.js

```
import React from 'react';
import { render, fireEvent } from '@testing-library/react';
import Diet from '../Views/Diet/Diet';
import { DietData } from '../Utils/DietData';

test('Filtrar elementos por categoría en el componente Diet', () => {
  const { getByText, getByAltText, queryByText } = render(<Diet />);

  DietData.forEach((category) => {
    category.items.forEach((item) => {
      expect(getByAltText(item.name)).toBeInTheDocument();
    });
  });

  DietData.forEach((category) => {
    const categoriaButton = getByText(category.title);
    fireEvent.click(categoriaButton);
    DietData
      .filter((cat) => cat.id === category.id)
      .forEach((cat) => {
        cat.items.forEach((item) => {
          expect(getByAltText(item.name)).toBeInTheDocument();
        });
      });
    DietData
      .filter((cat) => cat.id !== category.id)
      .forEach((cat) => {
        cat.items.forEach((item) => {
          expect(queryByText(item.name)).toBeNull();
        });
      });
  });
});
```

En la siguiente imagen se muestra el resultado de la ejecución de pruebas donde se puede observar que hubo un resultado con un 100% en cuanto a los componentes probados.

Figura 4

Resultado de las pruebas Sprint 1

```
PASS src/Pruebas/dietprueba.test.js
  Diet Component
    ✓ renders without crashing (49 ms)
    ✓ displays the category buttons (25 ms)
    ✓ displays item cards (107 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.862 s, estimated 3 s
Ran all test suites matching /dietprueba.test.js/i.

PASS src/Pruebas/register.test.js
  ✓ Renderiza el componente UserRegister sin errores (56 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.675 s
Ran all test suites matching /register.test.js/i.

PASS src/Pruebas/filtrardiet.test.js
  ✓ Filtrar elementos por categoría en el componente Diet (95 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.983 s, estimated 3 s
Ran all test suites matching /filtrardiet.test.js/i.
```

9.2 Pruebas Sprint 2

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo Medications en donde se muestran los medicamentos agregados en una CARD Según el caso de prueba PP01.

Figura 5

Script medication.test.js

```
import React from 'react';
import { render, screen } from '@testing-library/react';
test('Renderiza el componente Medication con datos en la pantalla', () => {
  const medications = [
    {
      id: 'medication1',
      data: {
        data: {
          medication: {
            nombreMedicamento: 'Medicamento 1',
            dosificacion: '10mg',
            unidades: 30,
            tomasDelDia: 2,
            fechadeinicio: '2023-11-03',
            hora: '08:00:00',
          },
        },
      },
    },
  ],
  render(<Medication medications={medications} />);
  expect(screen.getByText('Nombre del Medicamento: Medicamento 1')).toBeInTheDocument();
  expect(screen.getByText('Dosificación: 10mg')).toBeInTheDocument();
  expect(screen.getByText('Unidades: 30')).toBeInTheDocument();
  expect(screen.getByText('Tomas al Día: 2')).toBeInTheDocument();
  expect(screen.getByText('Fecha de Inicio: 2023-11-03')).toBeInTheDocument();
  expect(screen.getByText('Hora: 08:00:00')).toBeInTheDocument();
  expect(screen.getByAltText('icon-delete')).toBeInTheDocument();
  expect(screen.queryByText('No hay información de medicación disponible.')).toBeNull();
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo Quotes en donde se ve que si renderiza el componente con los datos agregados en las CARDS Según el caso de prueba PP01.

Figura 6

Script quotesolicitud.test.js

```
import React from 'react';
import { render, screen } from '@testing-library/react';
import Quote from '../Components/Quotes/Quote';

test('Renderiza el componente Quote con datos en las cards', () => {
  const citas = [
    {
      id: 1,
      data: {
        data: {
          cita: {
            doctor: {
              userName: 'Dr. Smith',
            },
            status: 'approved',
            date: '2023-11-05',
            time: '03:30 PM',
          },
        },
      },
    },
  ],
  render(<Quote />);

  setTimeout(() => {
    citas.forEach((cita) => {
      const doctorName = screen.getByText(`Doctor: ${cita.data.data.cita.doctor.userName}`);
      const status = screen.getByText(`Estado: ${cita.data.data.cita.status}`);
      const date = screen.getByText(`Fecha: ${cita.data.data.cita.date}`);
      const time = screen.getByText(`Hora: ${cita.data.data.cita.time}`);
      expect(doctorName).toBeInTheDocument();
      expect(status).toBeInTheDocument();
      expect(date).toBeInTheDocument();
      expect(time).toBeInTheDocument();
    });
  }, 1000);
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre la funcionalidad de Agendar citas **Según el caso de prueba PM03**

Figura 7

Script agendarctias.test.js

```
import React from 'react';
import { render } from '@testing-library/react';

test('Renderiza el calendario de CitasDr y verifica una cita agendada', () => {
  const { container } = render(
    <div>
      <div className="rbc-calendar">Calendario Aquí</div>
      <div className="cita-agendada">Cita Agendada</div>
    </div>
  );

  // Asegúrate de que el calendario esté presente
  const calendar = container.querySelector('.rbc-calendar');
  expect(calendar).toBeInTheDocument();

  // Verifica que la cita esté agendada
  const citaAgendadaElement = container.querySelector('.cita-agendada');
  expect(citaAgendadaElement).toBeInTheDocument();
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre la funcionalidad de Registrar usuario médico **Según el caso de prueba PM01**

Figura 8

Script registermedico.test.js

```
import React from 'react';
import { render } from '@testing-library/react';
test('Renderiza el formulario de registro de usuario médico con campo de carga de código minsa', () => {
  const { getByText, getByPlaceholderText, getByLabelText } = render(
    <Form>
      <label htmlFor="username">Nombre de usuario</label>
      <input
        type="text"
        id="username"
        name="username"
        placeholder="Nombre de usuario"
      />
    </>
    <label htmlFor="email">Correo electrónico</label>
    <input
      type="email"
      id="email"
      name="email"
      placeholder="Correo electrónico"
    />
    </>
    <label htmlFor="password">Contraseña</label>
    <input
      type="password"
      id="password"
      name="password"
      placeholder="Contraseña"
    />
    </>
    <label htmlFor="confirmPassword">Confirmar contraseña</label>
    <input
      type="password"
      id="confirmPassword"
      name="confirmPassword"
      placeholder="Confirmar contraseña"
    />
    </>
    <label htmlFor="codeMinsa">Cargar código minsa</label>
    <input
      type="File"
      id="codeMinsa"
      name="codeMinsa"
    />
    </>
    <button type="submit">Crear cuenta</button>
  </form>
);
const createAccountButton = getByText('Crear cuenta');
const usernameInput = getByPlaceholderText('Nombre de usuario');
const emailInput = getByPlaceholderText('Correo electrónico');
const passwordInput = getByPlaceholderText('Contraseña');
const confirmPasswordInput = getByPlaceholderText('Confirmar contraseña');
const codeMinsaInput = getByLabelText('Cargar código minsa');
expect(createAccountButton).toBeInTheDocument();
expect(usernameInput).toBeInTheDocument();
expect(emailInput).toBeInTheDocument();
expect(passwordInput).toBeInTheDocument();
expect(confirmPasswordInput).toBeInTheDocument();
expect(codeMinsaInput).toBeInTheDocument();
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre la funcionalidad de ChatBot **Según el caso de prueba PP08**

Figura 9

Script chatbotprueba.test.js

```
import React, { useState } from 'react';
import { render, fireEvent, screen } from '@testing-library/react';
test('Muestra y oculta el chatbot al hacer clic en el botón', () => {
  const Chatbot = () => {
    const [visible, setVisible] = useState(false);
    return (
      <>
        <button onClick={() => setVisible(!visible)}>Botón del chatbot</button>
        {visible && <div data-testid="chatbot">Chatbot Aquí</div>}
      </>
    );
  };
  render(<Chatbot />);
  expect(screen.queryById('chatbot')).toBeNull();
  const botButton = screen.getByText('Botón del chatbot');
  fireEvent.click(botButton);
  expect(screen.getByTestId('chatbot')).toBeInTheDocument();
  fireEvent.click(botButton);
  expect(screen.queryById('chatbot')).toBeNull();
});
```

En la siguiente imagen se muestra el resultado de la ejecución de pruebas donde se puede observar que hubo un resultado con un 100% en cuanto a los componentes probados.

Figura 10

Resultado de las pruebas Sprint 2

```
Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       3.448 s, estimated 5 s
Ran all test suites matching /medication.test.js/i.

PASS src/Pruebas/quotesolicitud.test.js
  ✓ Renderiza el componente Quote con datos en las cards (33 ms)

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       2.652 s, estimated 4 s
Ran all test suites matching /quotesolicitud.test.js/i.

PASS src/Pruebas/agendarcitas.test.js
  ✓ Renderiza el calendario de CitasDr y verifica una cita agendada (27 ms)

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       2.498 s
Ran all test suites matching /agendarcitas.test.js/i.

PASS src/Pruebas/registermedico.test.js
  ✓ Renderiza el formulario de registro de usuario médico con campo de carga de código minsa (

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       2.479 s
Ran all test suites matching /registermedico.test.js/i.

PASS src/Pruebas/chatbotprueba.test.js
  ✓ Muestra y oculta el chatbot al hacer clic en el botón (54 ms)

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       2.476 s
Ran all test suites matching /chatbotprueba.test.js/i.
```

9.3 Prueba Sprint 3

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo Medications en donde se muestran las notificaciones de los medicamentos **Según el caso de prueba PU02**.

Figura 11

Script Medication.test.js

```
import React from 'react';
import { render, screen, fireEvent } from '@testing-library/react';
const Medication = ({ medications, handleDelete }) => {
  return (
    <div>
      {medications.map((medication, index) => (
        <div key={index}>
          <p>Nombre del Medicamento: {medication.data.data.medication.nombreMedicamento}</p>
          <button onClick={() => handleDelete(medication.id)}>Eliminar</button>
        </div>
      ))}
    </div>
  );
};

const showTomaAlert = jest.fn();
test('Renderiza el componente Medication y muestra la alerta de toma', () => {
  const medications = [
    {
      id: 'medication1',
      data: {
        data: {
          medication: {
            nombreMedicamento: 'Medicamento 1',
          },
        },
      },
    },
  ];
  const handleDelete = (id) => {
    showTomaAlert();
  };
  render(<Medication medications={medications} handleDelete={handleDelete} />);
  expect(screen.getByText('Nombre del Medicamento: Medicamento 1')).toBeInTheDocument();
  fireEvent.click(screen.getByText('Eliminar'));
  expect(showTomaAlert).toHaveBeenCalled();
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo Calendario en donde se muestran las citas agregadas **Según el caso de prueba PP04.**

Figura 12

Script-calendariocitas.test.js

```
import React from 'react';
import { render } from '@testing-library/react';

test('Renderiza el calendario de CitasDr y verifica una cita agendada', () => {
  const { container } = render(
    <div>
      <div className="rbc-calendar">Calendario Aquí</div>
      <div className="cita-agendada">Cita Agendada</div>
    </div>
  );

  // Asegúrate de que el calendario esté presente
  const calendar = container.querySelector('.rbc-calendar');
  expect(calendar).toBeInTheDocument();

  // Verifica que la cita esté agendada
  const citaAgendadaElement = container.querySelector('.cita-agendada');
  expect(citaAgendadaElement).toBeInTheDocument();
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo HomeDr en donde se renderiza la vista del médico **Según el caso de prueba PM02.**

Figura 13

Script homedrprueba.test.js

```
import React from 'react';
import { render, screen } from '@testing-library/react';

test('Renderiza el componente HomeDr con datos simulados', () => {
  render(
    <div>
      <span>Citas</span>
      <span>Notificaciones</span>
      <span>Expedientes</span>
      <span>Cuenta</span>
      
    </div>
  );

  // Verificamos que el componente HomeDr se renderice
  expect(screen.getByText('Citas')).toBeInTheDocument();
  expect(screen.getByText('Notificaciones')).toBeInTheDocument();
  expect(screen.getByText('Expedientes')).toBeInTheDocument();
  expect(screen.getByText('Cuenta')).toBeInTheDocument();
  expect(screen.getByAltText('Logo')).toBeInTheDocument();
});
```

En la siguiente imagen se muestra el resultado de la ejecución de pruebas donde se puede observar que hubo un resultado con un 100% en cuanto a los componentes probados.

Figura 14

Resultado de la prueba Sprint 3

```
PASS src/Pruebas/medication.test.js
  ✓ Renderiza el componente Medication y muestra la alerta de toma (38 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.385 s, estimated 3 s
Ran all test suites matching /medication.test.js/i.

PASS src/Pruebas/calendariocitas.test.js
  ✓ Renderiza el calendario de CitasDr y verifica una cita agendada (29 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.582 s
Ran all test suites matching /calendariocitas.test.js/i.

PASS src/Pruebas/homedrprueba.test.js
  ✓ Renderiza el componente HomeDr con datos simulados (57 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.508 s
Ran all test suites matching /homedrprueba.test.js/i.
```

9.4 Prueba Sprint 4

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo AdminDoctors en donde se prueba la validación de los médicos **Según el caso de prueba PA01.**

Figura 15

Script-adminDoctors.test.js

```
import React from 'react';
import { render, screen, fireEvent } from '@testing-library/react';
describe('AdminDoctors', () => {
  it('debería permitir cambiar el estado de los médicos', () => {
    render(
      <div>
        <table>
          <thead>
            <tr>
              <th>Doctor</th>
              <th>Email</th>
              <th>Estado</th>
              <th>Acciones</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>Doctor Uno</td>
              <td>doctor1@example.com</td>
              <td>
                <select>
                  <option value="enRevisión">enRevisión</option>
                  <option value="aprobado">aprobado</option>
                </select>
              </td>
            </tr>
          </tbody>
        </table>
      </div>
    );
    const selectElement = screen.getByRole('combobox');
    expect(selectElement).toHaveValue('enRevisión');
    fireEvent.change(selectElement, { target: { value: 'aprobado' } });
    expect(selectElement).toHaveValue('aprobado');
  });
});
```

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo Admin en donde se prueba la visualización de estadísticas **Según el caso de prueba PA02.**

Figura 16

Script-admin.test.js

```
import React from 'react';
import { render, screen } from '@testing-library/react';
test('Renderiza y visualiza estadísticas correctamente', () => {
  const Data = {
    usersCount: 3,
    doctorsCount: 1,
    quotesCount: 3,
  };
  render(
    <main>
      <div>
        <h3>DASHBOARD</h3>
      </div>
      <div>
        <div>
          <h3>users</h3>
          <h1>{Data.usersCount}</h1>
        </div>
        <div>
          <h3>doctors</h3>
          <h1>{Data.doctorsCount}</h1>
        </div>
        <div>
          <h3>Quotes</h3>
          <h1>{Data.quotesCount}</h1>
        </div>
      </div>

      <div data-testid="line-chart"></div>
      <div data-testid="bar-chart"></div>
    </main>
  );
  expect(screen.getAllByText('users').length > 0).toBe(true);
  expect(screen.getAllByText('doctors').length > 0).toBe(true);
  expect(screen.getAllByText('Quotes').length > 0).toBe(true);
  expect(screen.getAllByText(String(Data.usersCount)).length > 0).toBe(true);
  expect(screen.getAllByText(String(Data.doctorsCount)).length > 0).toBe(true);
  expect(screen.getAllByText(String(Data.quotesCount)).length > 0).toBe(true);
  expect(screen.getByTestId('line-chart')).toBeInTheDocument();
  expect(screen.getByTestId('bar-chart')).toBeInTheDocument();
});
```

En la siguiente imagen se muestra el resultado de la ejecución de pruebas donde se puede observar que hubo un resultado con un 100% en cuanto a los componentes probados.

Figura 17

Resultado de la prueba Sprint 4

```
PASS src/Pruebas/admindoctors.test.js
  Admindocors
    ✓ debería permitir cambiar el estado de los médicos (265 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        5.242 s
Ran all test suites matching /admindocors.test.js/i.

PASS src/Pruebas/admin.test.js
  ✓ Renderiza y visualiza estadísticas correctamente (62 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        3.617 s
Ran all test suites matching /admin.test.js/i.
```

X. Planificación SCRUM

10.1 Product Backlog

A continuación, se presenta una tabla que describe las historias de usuario junto con su importancia y tiempo estimado. Se ha asignado una puntuación de importancia del 1 al 30, una puntuación del 1 al 10 en cuanto a la prioridad y del 1 al 10 en cuanto a su impacto para reflejar el valor que cada historia aporta al proyecto.

Tabla 14

Product Backlog

ID	Historia	Prioridad	Impacto	Importancia	Tiempo estimado
HUP01	Control de la medicación	10	9	38	32 horas
HUP02	Notificaciones personalizadas	9	8	35	36 horas
HUP03	Espacio informativo de la diabetes	7	6	25	24 horas
HUP04	Espacios nutricionales	7	7	38	24 horas
HUP05	Espacios de actividad física	6	5	20	24 horas
HUP06	Registrar usuario paciente	5	3	20	24 horas
HUP07	Creación de citas	8	8	38	36 horas
HUP08	ChatBot	8	8	20	24 horas
HUM01	Registro de usuario médico	5	3	15	24 horas
HUM02	Espacio del médico	7	7	20	36 horas
HUM03	Agendar citas	7	7	25	32 horas
HUM04	Creación de calendario	7	6	30	32 horas
HUP01	Crud Administrativo	8	8	20	24 horas
HUA02	Estadísticas de la aplicación	9	9	25	24 horas

10.2 Product Backlog organizado por Sprint

A continuación, se muestran las historias de usuario en orden descendente según su importancia, lo que a su vez permite separarlas por Sprint

Tabla 15

Product Backlog organizado por Sprint

ID	Historia	Importancia	Sprint	Fecha Prevista
HUP04	Espacios nutricionales	38	1	04/09/2023-
HUP05	Espacios de actividad física	20	1	09/09/23
HUP06	Registrar usuario paciente	20	1	
HUP01	Control de la medicación	38	2	20/09/2023-
HUP07	Creación de citas	38	2	20/10/2023
HUM03	Agendar Citas	25	2	
HUP03	Espacios informativos de la diabetes	25	2	
HUM01	Registro de usuario médico	15	2	
HUP08	ChatBot	20	2	
HUM02	Espacio del médico	20	3	24/10/2023-
HUM04	Creación de calendario	30	3	29/10/2023
HUP02	Notificaciones personalizadas	35	3	
HUA01	Crud Administrativo	20	4	29/10/2023-
HUA02	Estadísticas de la aplicación	25	4	30/10/2023

10.3 Sprint Backlog

10.3.1 Sprint 1

A continuación, se muestra el inicio del primer Sprint, programado para una semana de duración. En este Sprint, hemos planificado la inclusión de tres historias de usuario junto con la asignación de un total de diecinueve tareas que deberán ser completadas dentro de ese período.

Tabla 16

Sprint 1

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUP04	Espacios nutricionales.	Realizar Entrevista a Nutricionista.	2
		Diseñar interfaz para el espacio y estilos css.	2
		Agregar información relevante y verificada de dietas nutricionales en la diabetes.	5
		Crear un data para almacenar los datos de ese platillo	5
		Crear CARD que permita visualizar el platillo	5
		Crear Modal para ver los detalles de ese platillo	5
HUP05	Espacios de actividad física.	Realizar Entrevista entrenador físico.	4
		Diseñar interfaz para el espacio.	4
		Agregar información relevante y verificada de la de la actividad física en la diabetes.	4
		Crear un data para almacenar los datos de las rutinas.	4
		Crear CARD que permite visualizar las rutinas	4
		Crear Modal para ver la rutina con su detalle y video.	4
HUP06	Registrar usuarios Paciente.	Configurar Firebase Authentication.	3
		Crear colección para almacenamiento de usuarios en Firestore.	3
		Configurar redux para la Autenticación de usuarios.	3
		Codificar funcionalidad de Registro de usuario.	3
		Diseño de la vista de inicio y registro de sesión.	6
		Validar la entrada del Usuario.	6
		Almacenar información adicional del usuario.	6

10.3.2 Casos de prueba del Sprint 1

Test Case ID	PM01	Test Case Description	Prueba de Registro de sesión de un usuario paciente (Registro de usuario paciente)
Test type	Caja Negra	Testing Method	Escenario
Created By	Donald Alvarez	Reviewed By	Version 1.0

QA Tester's Log

Tester's Name	Javiera Leiva	Date Tested	25/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerrequisitos:	S#	Test Data		
1	Cargar la aplicación	1	Jeyson Gonzales		
2	Interfaz elaborada	2	12345678		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Dar clic en la opción regístrate.	Carga la vista de registro de usuarios.	Vista de registro de usuarios.	Pass
2	Llena formulario con sus datos correspondientes.	Datos aceptados de manera correcta.	Formulario llenado de manera correcta.	Pass
3	Dar clic en crear cuenta.	Spinner de carga para redirigir a la vista principal (Home).	Vista de carga ejecutada con éxito.	Pass
4	Esperar que cargue a la vista principal.	Carga la vista de Home con el usuario que acaba de registrarse.	Vista del Home con su usuario correspondiente.	Pass

Test Case ID	PP04.1	Test Case Description	Filtro de dietas nutricionales (Funcionalidad de filtro de dietas nutricionales.)	
Test type	Caja Negra	Testing Method	Escenario	
Created By	Donald Alvarez	Reviewed By	Version	1.0

QA Tester's Log

Tester's Name	Brando Estrada	Date Tested	25/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	----------------	--------------------	------------	---	------

S #	Prerrequisitos:
1	Cargar la aplicación
2	Interfaz elaborada e inicio de sesión previo

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /Suspended
1	Agregar sus datos y dar clic en iniciar sesión.	Inicio de sesión Exitoso.	Vista de Home.	Pass
2	Dar clic al menú desplegable e irse a dietas.	Visualizar Menú desplegable.	Menú de navegación.	Pass
3	Visualizar dietas nutricionales.	Carga de la vista de dietas nutricionales.	Dietas cargadas con éxito.	Pass
4	Dar clic en los botones De Desayuno, Almuerzo y Cena.	Cargar alimentos correspondientes a su filtro.	Alimentos cargados con éxito.	Pass

10.3.3 Sprint 2

A continuación, se muestra el inicio del segundo Sprint, programado para tres semanas de duración. En este Sprint, hemos planificado la inclusión de cinco historias de usuario junto con la asignación de un total de veinticinco tareas que deberán ser completadas dentro de ese período.

Tabla 17
Sprint 2

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUP01	Control de la medicación.	Análisis de formularios para la creación de la vista de control de la medicación.	6
		Configuración de Firebase- Firestore para almacenar los datos de los medicamentos.	6
		Creación de formularios para el control de la medicación.	4
		Creación de modal para visualizar datos del formulario.	4
		Diseñar interfaz para el control de la medicación.	4
		Creación de CARD para visualizar medicamentos agregados.	4
		Creación de modal para visualizar y editar los datos.	4
HUP07	Creación de citas.	Análisis de formulario para la creación de citas.	10
		Configuración de Firebase- Firestore para almacenar los datos de las citas.	8
		Crear formularios para la solicitud de la cita.	6
		Crear modal para visualizar formulario.	6
		Crear CARD para mostrar la solicitud de la cita	6
HUM03	Agendar citas.	Consumir los datos del Firestore recibiendo la cita.	8
		Diseñar espacio del médico para guardar las citas.	8
		Crear función que permita aceptar o rechazar cita.	8
		Crear función que permita editar campos permitidos de la cita como el día y la hora.	8
HUP03	Espacio informativo sobre la diabetes.	Realizar entrevista a médico diabetólogo.	8
		Diseñar interfaz del espacio.	8
		Agregar información relevante y verificada de la diabetes.	8
HUM01	Registro de usuario médico	Combinar registro de usuarios con médicos y pacientes.	3
		Modificar Firebase Authentication.	3
		Modificar redux para la Autenticación de usuarios paciente y médico.	3
		Modificar interfaz agregando registro del médico.	3

10.3.4 Casos de prueba del Sprint 2

Test Case ID	PP01	Test Case Description	Mostrar los medicamentos en una CARD al ser agregados en un formulario que se encuentra dentro de un Modal.	
Test type	Caja Negra	Testing Method	Escenario	
Created By	Donald Alvarez	Reviewed By	Version	1.3

Tester's Name	Javiera Leiva	Date Tested	26/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	---	------

QA Tester's Log

S #	Prerrequisitos:
1	Cargar la aplicación
2	Interfaz elaborada e inicio de sesión previo.
3	Tener acceso al formulario que contiene el campo para agregar medicamentos.
4	Tener acceso al Modal que muestra la CARD con los medicamentos agregados.
5	Tener una lista de medicamentos para agregar al formulario.
6	Contar con un entorno de prueba adecuado.

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /Suspended
1	Agregar sus datos y dar clic en iniciar sesión.	Inicio de sesión exitoso.	Vista de Home.	Pass
2	Dar clic al menú desplegable e irse a dietas.	Visualizar Menú desplegable.	Menú de Medicación.	Pass
3	Visualizar la vista de medicación.	Carga de la vista de dietas Medicación.	Medicación cargada con éxito.	Pass
4	Dar clic en los botones de agregar medicamentos.	Cargar el modal de manera exitosa.	Medicamentos guardados de manera exitosa.	Pass

Test Case ID	PP07	Test Case Description	Prueba de mostrar una solicitud de cita gestionada por el usuario paciente.
Test type	Caja Negra	Testing Metod	Escenario
Created By	Javiera Leiva	Reviewed By	Version 1.3

QA Tester's Log

Tester's Name	Donald Alvarez	Date Tested	26/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	----------------	--------------------	------------	---	------

S Prerrequisitos:

- #
- 1 Cargar la aplicación
- 2 Interfaz elaborada
- 3 Contar con un usuario válido y activo de tipo paciente en el sistema
- 4 Disponer de un navegador web compatible con la aplicación.
- 5 Asegurarse de que la aplicación tenga la funcionalidad de mostrar la solicitud de cita correctamente implementada.
- 6 Asegurarse de que los datos de la solicitud de cita, como la fecha, hora, etc., estén ingresados correctamente en el sistema.

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /Suspended
1	Dar clic en el botón añadir.	La carga del MODAL.	Vista del MODAL de manera exitosa.	Pass
2	Llenar los datos del formulario.	Datos aceptados de manera correcta.	Formulario llenado de manera correcta.	Pass
3	Dar clic en el botón crear cita.	La carga del MODAL de Creación de cita.	Vista de carga ejecutada con éxito.	Pass
4	Visualizar cita en la CARD.	Carga la vista de la Cita en proceso.	Vista de Cita aceptada o rechazada con éxito.	Pass

Test Case ID	PM04	Test Case Description	Mostrar opción de aceptar o rechazar cita.	
Test type	Caja Negra	Testing Method	Escenario	
Created By	Brando Estrada	Reviewed By	Version	1.3

Tester's Name	Donald Alvarez	Date Tested	26/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	-----------------------	--------------------	-------------------	---	-------------

S #	Prerequisites:
1	Cargar la aplicación
2	Interfaz elaborada e inicio de sesión previo.
3	Implementar la lógica de programación que permita al usuario seleccionar la opción de aceptar o rechazar la cita
4	Tener acceso al Calendario
5	Tener una lista de medicamentos para agregar al formulario.

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar sesión.	Inicio de sesión exitoso.	Vista de Home.	Pass
2	Ir a la vista del Calendario.	Visualizar calendario.	Calendario abierto de manera exitosa.	Pass
3	Abrir el MODAL de opciones.	Carga del MODAL de opciones.	MODAL Abierto de manera exitosa.	Pass
4	Dar clic en los botones de aceptar, rechazar o posponer cita.	Clic de decisión de manera exitosa.	Cita aceptada, rechazada o pospuesta con éxito.	Pass

Test Case ID	PM06	Test Case Description	Prueba de Registro de sesión de un usuario Medico (Registro de usuario Medico).	
Test type	Caja Negra	Testing Method	Escenario	
Created By	Donald Alvarez	Reviewed By	Version	1.3

QA Tester's Log

Tester's Name	Javiera Leiva	Date Tested	26/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerrequisitos:	S #	Test Data		
1	Cargar la aplicación	1	Guillermo Miranda		
2	Interfaz elaborada	2	12345678		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /Suspended
1	Dar clic en la opción regístrate.	Carga la vista de registro de usuarios.	Vista de registro de usuarios.	Pass
2	Llena formulario con sus datos correspondientes.	Datos aceptados de manera correcta.	Formulario llenado de manera correcta.	Pass
3	Selecciona tu carné MINSA en la Galería de Fotos.	Selección de foto de manera correcta.	Muestra de selección de la foto de manera exitosa.	Pass
4	Dar clic en crear cuenta.	Spinner de carga para redirigir a la vista principal (Home).	Vista de carga ejecutada con éxito.	Pass
5	Esperar que cargue a la vista principal.	Carga la vista de Home con el usuario que acaba de registrarse.	Vista del Home con su usuario correspondiente.	Pass

Test Case ID	PP08	Test Case Description	Prueba que permite un flujo de interacción para el usuario Paciente de la mano de IA.	
Test type	Caja Negra	Testing Method	Escenario	
Created By	Brando Estrada	Reviewed By	Version	1.3

QA Tester's Log

Tester's Name	Donald Alvarez	Date Tested	26/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	----------------	--------------------	------------	---	------

S #	Prerequisites:	S #	Test Data
1	Cargar la aplicación	1	Alba García
2	Interfaz elaborada		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /Suspended
1	Dar clic en el Chatbot.	Carga la vista del chatbot.	Vista del Chatbot.	Pass
2	Responde las preguntas de manera intuitiva.	Datos respondidos de manera correcta.	Respuestas correctas.	Pass
3	Espera tu Pre-Diagnostico.	Pre- Diagnostico mostrado al Usuario.	Muestra de Pre-Diagnostico de manera correcta.	Pass
4	Dar Opción de Crear Cita.	Dirigir al usuario a crear cita.	Vista de creación de citan redirigida de manera exitosa.	Pass

10.3.5 Sprint 3

A continuación, se muestra el inicio del tercer Sprint, programado para una semana de duración. En este Sprint, hemos planificado la inclusión de tres historias de usuario junto con la asignación de un total de 15 tareas que deberán ser completadas dentro de ese período.

Tabla 18

Sprint 3

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUM03	Espacio del médico.	Consumir datos de Firestore	5
		Configurar redux para recibir los datos	5
		Configurar notificaciones del médico	5
		Diseñar interfaz para el espacio del médico	4
		Diseñar módulos del médico	5
HUM02	Espacio publicitario del médico.	Diseñar interfaz del espacio publicitario	8
		Establecer medio de contacto con diseñador de la aplicación	8
		Programar medio de contacto entre el diseñador y el médico	8
HUM05	Creación de calendario.	Configurar Firebase-Firestore para almacenar los datos	5
		Diseñar interfaz del calendario	5
		Programar calendario para poder interactuar agregando cita o anotaciones	8
		Programar funciones de actualización y eliminación de eventos	8
		Gestionar la visualización y navegación del Calendario	6
HU02	Notificaciones	Configurar Notificaciones	18
		Programar funcionalidades de las notificaciones	18

10.3.6 Casos de prueba Sprint 3

Test Case ID	PU02	Test Case Description	Mostrar notificaciones en la toma de medicamentos.	
Test type	Caja negra	Testing Method	Escenario	
Created By	Javiera Leiva	Reviewed By	Version	1.4

QA Tester's Log

Tester's Name	Brando Estrada	Date Tested	26/10/2020	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:				
1	Cargar la aplicación.				<div>Test Scenario</div>
2	El usuario ha iniciado sesión con un tipo de cuenta paciente.				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Establecer horario	Establecimiento correcto	Establecimiento exitoso	Pass
2	Guardar medicamento	Guardado correctamente	Guardado exitosamente	Pass
3	Esperar la notificación	La notificación llegue con sonido.	La notificación llega con sonido correctamente	Pass
4	Dar Clic en “Aceptar” en la alerta.	El botón funcione.	Clic correcto y sin errores.	Pass

Test Case ID	PP07	Test Case Description	Mostrar calendario con citas agregadas.	
Test type	Caja negra	Testing Method	Escenario	
Created By	Donald Alvarez	Reviewed By	Version	1.4

QA Tester's Log

Tester's Name	Javiera Leiva	Date Tested	26/10/2020	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	---	------

S #	Prerrequisitos:
1	Cargar la aplicación.
2	El usuario ha iniciado sesión con un tipo de cuenta médico.

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar sesión como medico	Iniciar sesión de manera correcta	Inicio de sesión exitoso	Pass
2	Dar Clic en el botón de citas	Botón en funcionamiento correcto.	Ingreso a citas de manera exitosa.	Pass

Test Case ID	PM03	Test Case Description	Mostrar la vista de los módulos establecidos en el espacio del médico.	
Test type	Caja negra	Testing Method	Escenario	
Created By	Brando Estrada	Reviewed By	Version	1.4

QA Tester's Log

Tester's Name	Donald Alvarez	Date Tested	26/10/2020	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerrequisitos:				
1	Cargar la aplicación.				
2	El usuario ha iniciado sesión con un tipo de cuenta médico.				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar sesión como medico	Iniciar sesión de manera correcta	Inicio de sesión exitoso	Pass

10.3.7 Sprint 4

A continuación, se muestra el inicio del cuarto Sprint, programado para 3 días de duración. En este Sprint, hemos planificado la inclusión de 2 historias de usuario junto con la asignación de un total de nueve tareas que deberán ser completadas dentro de ese período.

Tabla 19

Sprint 4

ID	Historia de usuario	Tareas	Tiempo (Horas)
HUA01	CRUD Administrativo.	Configurar Firestore para obtener los datos	5
		Diseñar interfaz de la vista administrativa	5
		Programar funcionalidad del CRUD	5
		Crear tabla para mostrar los datos de los usuarios	5
		Configurar botones para poder interactuar el CRUD	4
HUA02	Estadísticas de la aplicación.	Configurar Firestore para obtener las analíticas	6
		Diseñar interfaz para mostrar las gráficas	6
		Codificar lógica para que las gráficas sean coherentes	6
		Analizar los datos y funcionalidades que más utilizan	6

10.3.8 Casos de prueba Sprint 4

Test Case ID	PA01	Test Case Description	Mostrar la vista administrativa para validar médicos y visualizar estadísticas.	
Test type	Caja negra	Testing Method	Escenario	
Created By	Javiera Leiva	Reviewed By	Version	1.6

QA Tester's Log

Tester's Name	Donald Alvarez	Date Tested	26/10/2020	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	----------------	--------------------	------------	---	------

S #	Prerrequisitos:
1	Cargar la aplicación.
2	El usuario ha iniciado sesión con un tipo de cuenta Administrativa

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar sesión como Administración	Iniciar sesión de manera correcta	Inicio de sesión exitoso	Pass

Test Case ID	PA02	Test Case Description	Mostrar estadísticas del uso de la aplicación	
Test type	Caja negra	Testing Method	Escenario	
Created By	Donald Alvarez	Reviewed By	Version	1.6

QA Tester's Log

Tester's Name	Brando Estrada	Date Tested	26/10/2020	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	----------------	--------------------	------------	---	------

S #	Prerrequisitos:
1	Cargar la aplicación.
2	El usuario ha iniciado sesión con un tipo de cuenta Administrativa

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar sesión como Administración.	Iniciar sesión de manera correcta.	Inicio de sesión exitoso.	Pass
2	Dar Clic en el botón de Estadísticas.	Redirección correcta.	Visualización exitosa.	Pass



Manual de Usuario Aplicación Control Health

Para el control y gestión de la mediación de la diabetes.

Versión 1

Tabla de Contenido

I.	Introducción	1
II.	Requisitos para el uso.....	2
III.	Usuario Paciente	3
3.1	Vista de inicio de sesión usuario Paciente	3
3.2	Vista de registro de usuario Paciente.....	3
3.3	Vista de inicio usuario Paciente	4
3.4	Vista de citas médicas usuario Paciente	5
3.5	Vista de dietas usuario Paciente.....	6
3.6	Vista de rutinas de ejercicio usuario Paciente	8
3.7	Vista de control de la medicación usuario Paciente	10
3.8	Vista de chatbot usuario Paciente	10
IV.	Usuario Médico	12
4.1	Vista de inicio de sesión usuario Médico	12
4.2	Vista de registro de usuario Médico	13
4.3	Vista de inicio Médico	14
4.4	Vista de citas de usuario Médico.....	14
4.5	Vista de expedientes usuario Médico.....	15
4.6	Vista de historial de citas usuario Médico.....	17
V.	Usuario Administrador	19
5.1	Vista menú (Administrador)	19
5.2	Vista de estadísticas (Administrador).....	20
5.3	Vista de acceso a médicos (Administrador)	21
5.4	Vista de roles de usuarios (Administrador)	22

Índice de Figuras

Figura 1 Vista de inicio de sesión (usuario Paciente)	3
Figura 2 Vista de registro (Usuario Paciente)	4
Figura 3 Vista de inicio (Usuario Paciente)	5
Figura 4 Vista de citas médicas (Usuario Paciente)	6
Figura 5 Vista de dietas (Usuario Paciente)	7
Figura 6 Ingredientes de la receta (Usuario Paciente)	7
Figura 7 Vista de rutinas de ejercicio (Usuario Paciente)	8
Figura 8 Vista de indicaciones de la rutina (Usuario Paciente)	9
Figura 9 Vista de reproducción de la rutina (Usuario Paciente)	9
Figura 10 Vista de control de la medicación (Usuario Paciente)	10
Figura 11 Vista de chatbot (Usuario Paciente)	11
Figura 12 Vista del chatbot habilitado (Usuario Paciente)	11
Figura 13 Vista de inicio de sesión (Usuario Médico)	12
Figura 14 Vista de registro (Usuario Médico)	13
Figura 15 Vista de inicio (Usuario Médico)	14
Figura 16 Vista de citas (Usuario Médico)	15
Figura 17 Vista de expedientes (Usuario Médico)	16
Figura 18 Vista de selección de expedientes (Usuario Médico)	16
Figura 19 Vista de historial de citas (Usuario Médico)	17
Figura 20 Vista de formulario de expedientes (Usuario Médico)	18
Figura 21 Vista de menú (Administrador)	19
Figura 22 Vista de estadísticas (Administrador)	20
Figura 23 Vista de acceso a médicos (Administrador)	21
Figura 24 Vista de roles (Administrador)	22

I. Introducción

Control Health es una aplicación diseñada para brindarte un control efectivo sobre la Diabetes. Sus funciones abarcan desde el seguimiento de tu medicación hasta la educación en Diabetes, nutrición y ejercicio. Además, conecta al paciente con médicos diabetólogos a través de un práctico chatbot para programar citas médicas con facilidad.

Aunque Control Health se destaca por su simplicidad de uso, es importante mencionar que se espera que los usuarios tengan cierta familiaridad con los conceptos y términos relacionados con la Diabetes. Este Manual de Usuario de Control Health se convierte en una guía esencial, proporcionando una visión completa de la aplicación y brindando instrucciones detalladas, paso a paso, para aprovechar al máximo todas las funcionalidades.

Recomendamos encarecidamente estudiar detenidamente este manual antes de aventurarse en el uso del software. De esta manera, se estará equipado con el conocimiento necesario para sacar el máximo provecho de Control Health y tomar el control de la salud de manera efectiva.

II. Requisitos para el uso

Para el correcto funcionamiento de la aplicación es necesario tener cubiertos una serie de requisitos, tanto de hardware como software, mencionados a continuación:

Requisitos mínimos para los diferentes dispositivos	
Requisitos mínimos PC	Requisitos mínimos Android
<ul style="list-style-type: none"> • SO: Windows 7 • Procesador: 1.0 GHz • Memoria: 256 MB de RAM • Gráficos: Tarjeta compatible con DirectX y con 64 MB* • DirectX®: 8.1 o superior • Disco Duro: 600 MB de espacio libre • Sonido: Compatible con DirectX • Otros: Estos requisitos son los mínimos se puede usar en equipos más actuales. 	<ul style="list-style-type: none"> • SO: Android 4 • Procesador: Octa-core 1.2 GHz • Memoria: 2 MB de RAM y 50 MB ROM • Otros: Estos requisitos son los mínimos se puede usar en equipos más actuales.

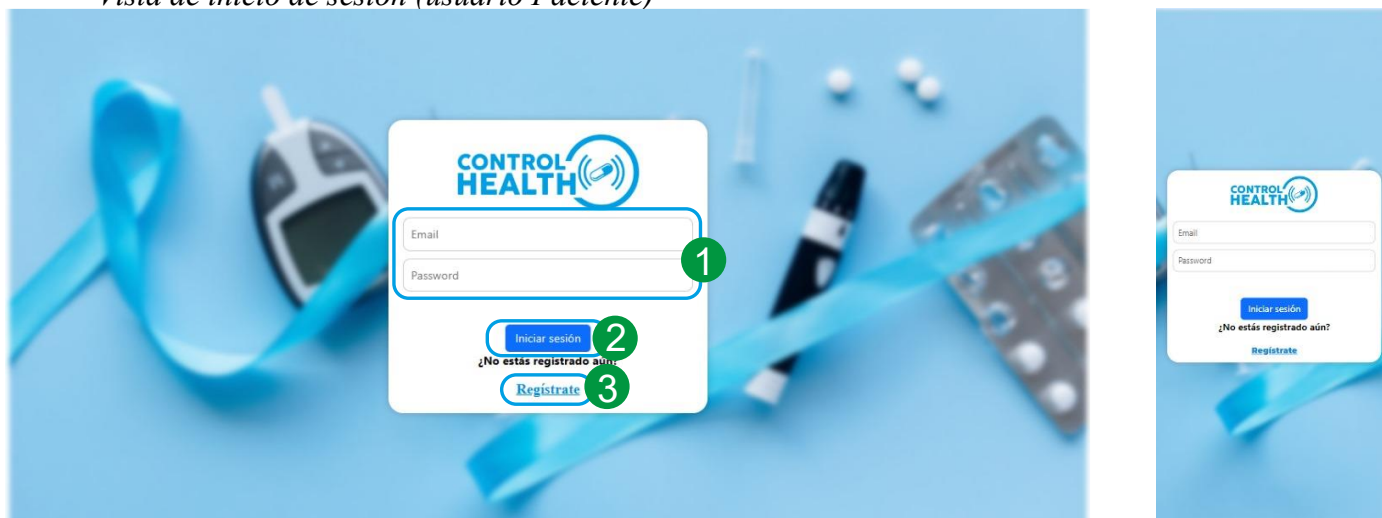
III. Usuario Paciente

3.1 Vista de inicio de sesión usuario Paciente

Para acceder a la página, la persona debe dirigirse al enlace www.controlhealth.com/login. Una vez en la página, se encontrará con la opción de ingresar los datos en los campos designados para el correo electrónico y la contraseña (1). Después de introducir la información requerida, deberá hacer clic en el botón "Iniciar sesión" (2). En caso de no contar con una cuenta, se le indicará que haga clic en "Registrarse" ubicado debajo de la opción de iniciar sesión (3). Posteriormente, al completar el proceso de registro, la persona deberá seleccionar su rol, ya sea como paciente o médico.

Figura 1

Vista de inicio de sesión (usuario Paciente)



Nota: Esta ilustración muestra la vista el inicio de sesión los usuarios de la página web y la perspectiva de la aplicación móvil.

3.2 Vista de registro de usuario Paciente

Una vez que el usuario ha seleccionado su rol, en este caso, sería paciente (1), procede a completar los datos solicitados (2) asegurándose de que no haya ningún dato erróneo. Al finalizar este proceso, deberá presionar el botón "Crear cuenta" (3) para activar su cuenta y así poder utilizar la plataforma web. En caso de que ya disponga de una cuenta, puede hacer clic en "Login" (4) para regresar al apartado de inicio de sesión.

Figura 2*Vista de registro (Usuario Paciente)*

CONTROL HEALTH

1 Paciente Doctor 2

Nombre

Apellido

Email

Contraseña

Confirmar contraseña

3 Crear cuenta

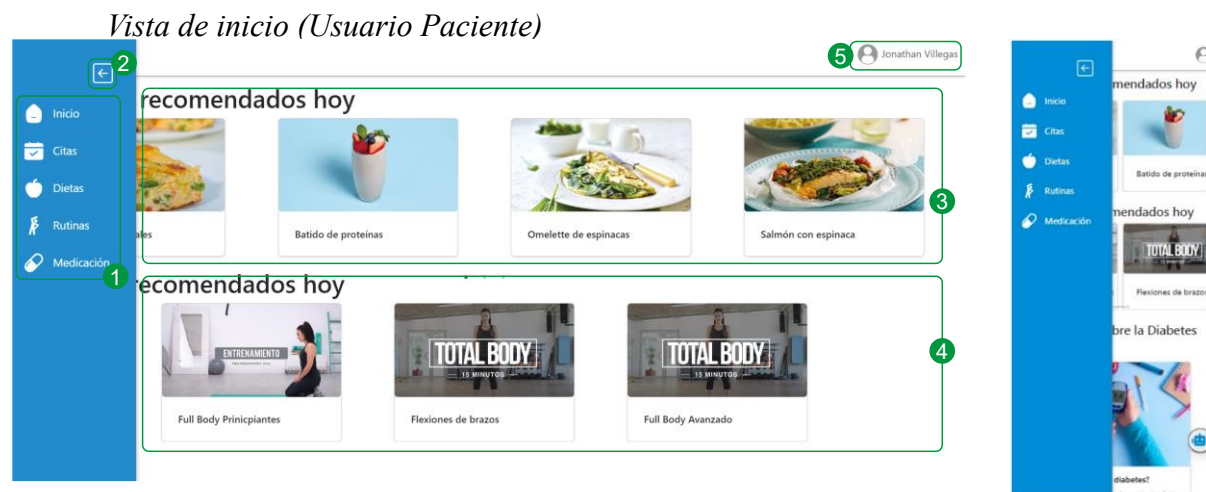
¿Ya tienes una cuenta?

4 Login

Nota: Esta figura muestra la ventana de creación de nuevos usuarios pacientes.

3.3 Vista de inicio usuario Paciente

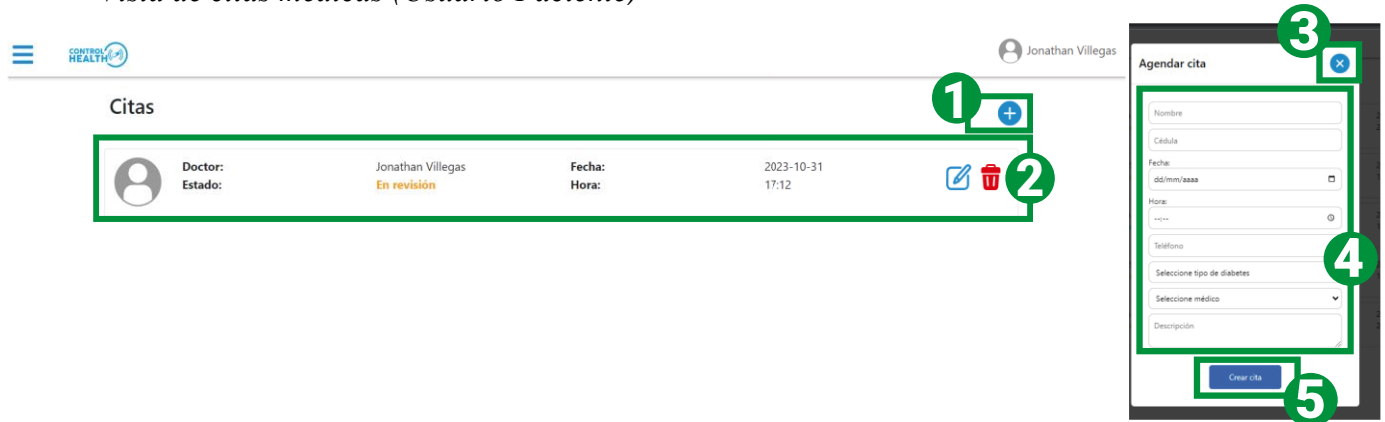
Al iniciar sesión, el usuario se encontrará con una vista que contiene un menú, el cual se despliega al hacer clic (1), revelando las diversas funcionalidades del software. Si el usuario desea ocultar el menú, simplemente debe hacer clic (2), y este se ocultará automáticamente. En la interfaz, se visualizan las recetas recomendadas (3) y las rutinas recomendadas (4), simplificando así la búsqueda para el usuario. En la parte superior (5), se muestra el nombre del usuario correspondiente a la cuenta con la que está navegando en la plataforma web.

Figura 3

Nota: Esta figura muestra la vista del home al iniciar sesión desde el usuario paciente de la página web y la perspectiva de la aplicación móvil.

3.4 Vista de citas médicas usuario Paciente

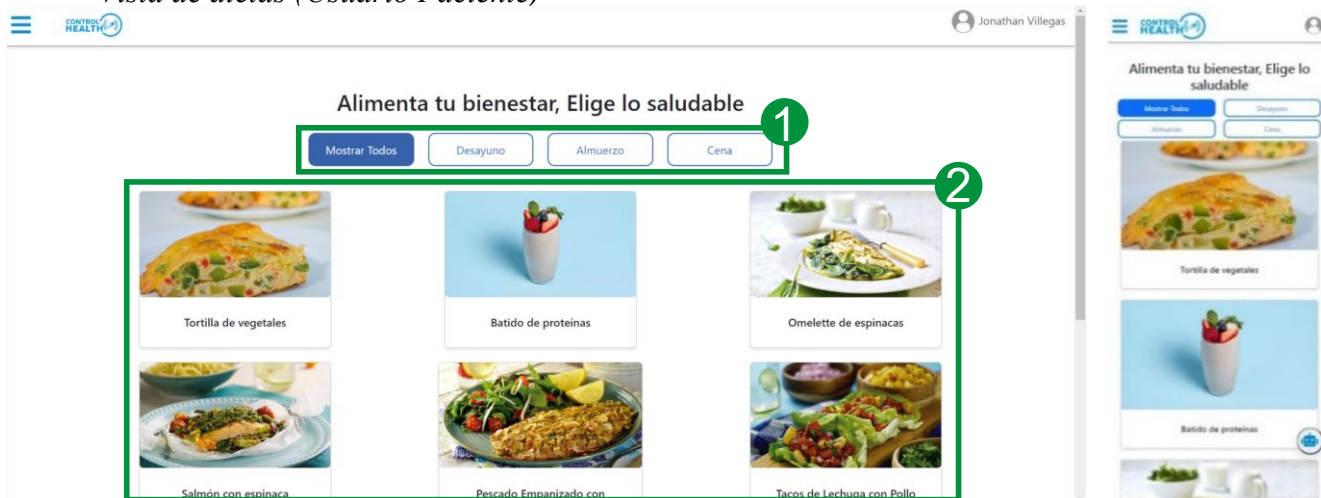
Una vez que el usuario selecciona una opción Citas del menú, se le presentará una ventana en la cual puede agregar una nueva cita al hacer clic en el botón correspondiente (1). También tiene la opción de visualizar las citas en proceso, donde puede eliminar o editar la información de una cita específica. Al hacer clic en "Agregar cita" (1), se abrirá una ventana adicional que contiene un botón (3) para salir de dicha ventana. Sin embargo, si el usuario desea agendar la cita, deberá completar toda la información requerida, como se muestra en (4). Una vez que toda la información está completa, solo es necesario hacer clic en (5) para enviar la solicitud de la cita.

Figura 4*Vista de citas médicas (Usuario Paciente)*

Nota: Esta ilustración muestra la vista del home al iniciar sesión desde el usuario paciente de la página web y la perspectiva de la aplicación móvil.

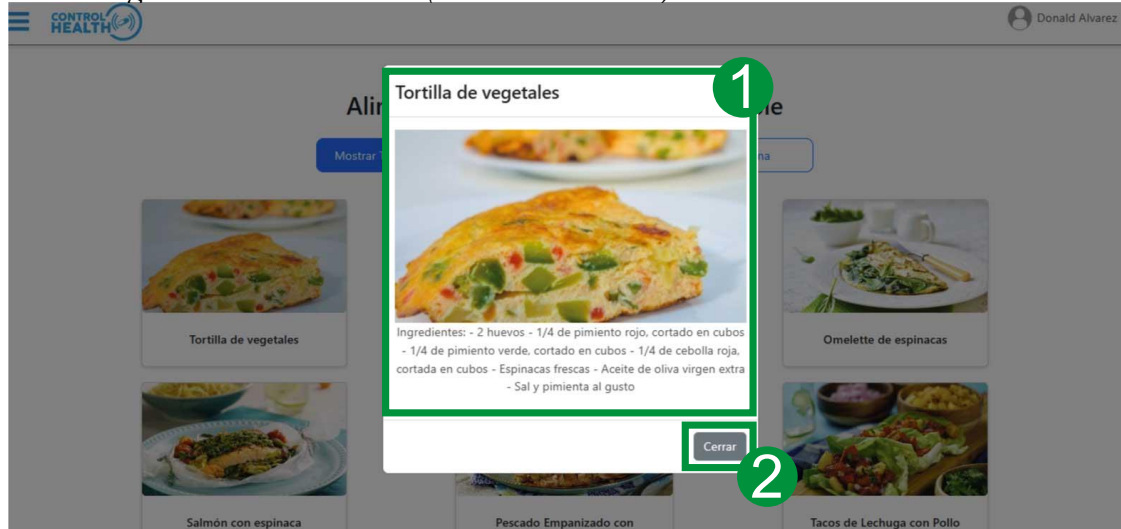
3.5 Vista de dietas usuario Paciente

Al seleccionar la opción "Dietas" en el menú, al usuario se le presentará una ventana en la que se pueden visualizar las recetas recomendadas (2). Además, en la interfaz se exhiben los filtros (1) de recetas, organizados según las categorías de Desayuno, Almuerzo o Cena, con el propósito de facilitar la navegación tanto en la vista web como en la vista móvil.

Figura 5*Vista de dietas (Usuario Paciente)*

Nota: Esta ilustración muestra la vista del home al iniciar sesión desde el usuario paciente de la página web y la perspectiva de la aplicación móvil.

Al seleccionar una de las recetas, el usuario será dirigido a una ventana (1) que proporciona información detallada, incluyendo los ingredientes de la receta y las instrucciones para la preparación. En la parte inferior de la ventana, se encuentra el botón "Cerrar" (2), el cual permite al usuario regresar a la sección de recetas y explorar otros contenidos disponibles.

Figura 6*Ingredientes de la receta (Usuario Paciente)*

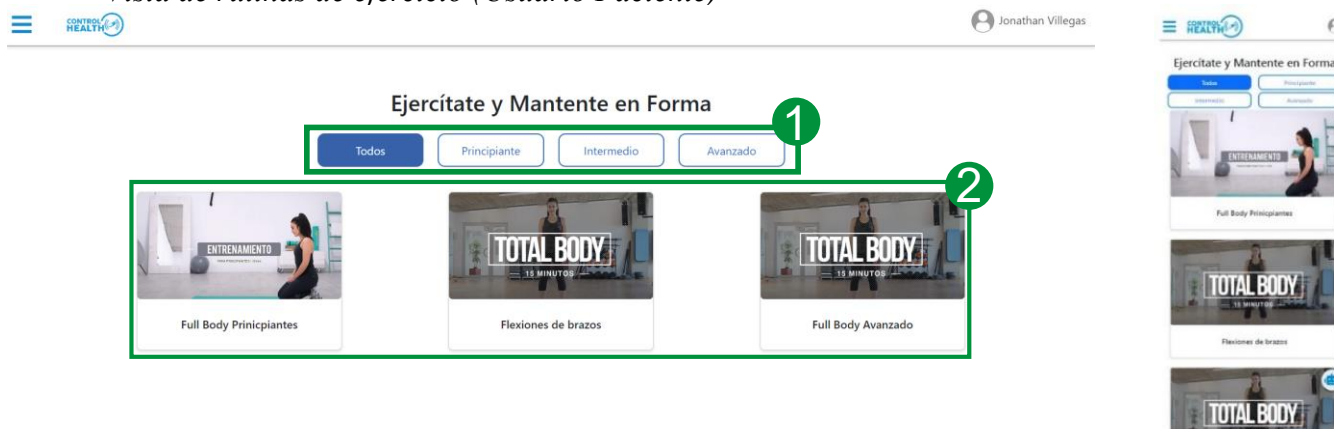
Nota: Esta ilustración muestra la ventana de los ingredientes y preparación de la receta.

3.6 Vista de rutinas de ejercicio usuario Paciente

Al seleccionar la opción "Rutinas" en el menú, se abrirá una ventana que presenta las rutinas recomendadas (2) y la posibilidad de buscar rutinas a través de los filtros (1). Estos filtros están divididos en categorías de Principiantes, Intermedio o Avanzado, lo que simplifica la utilización del software. Esta estructura de filtros se muestra de manera consistente tanto en la vista web como en la vista móvil para una experiencia de usuario coherente.

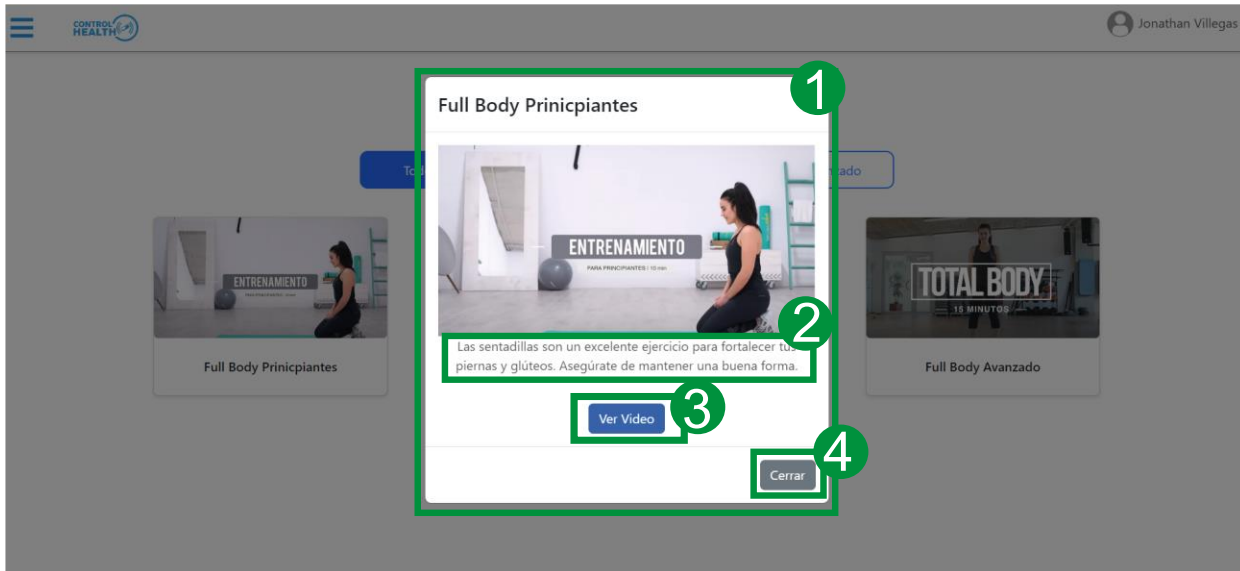
Figura 7

Vista de rutinas de ejercicio (Usuario Paciente)



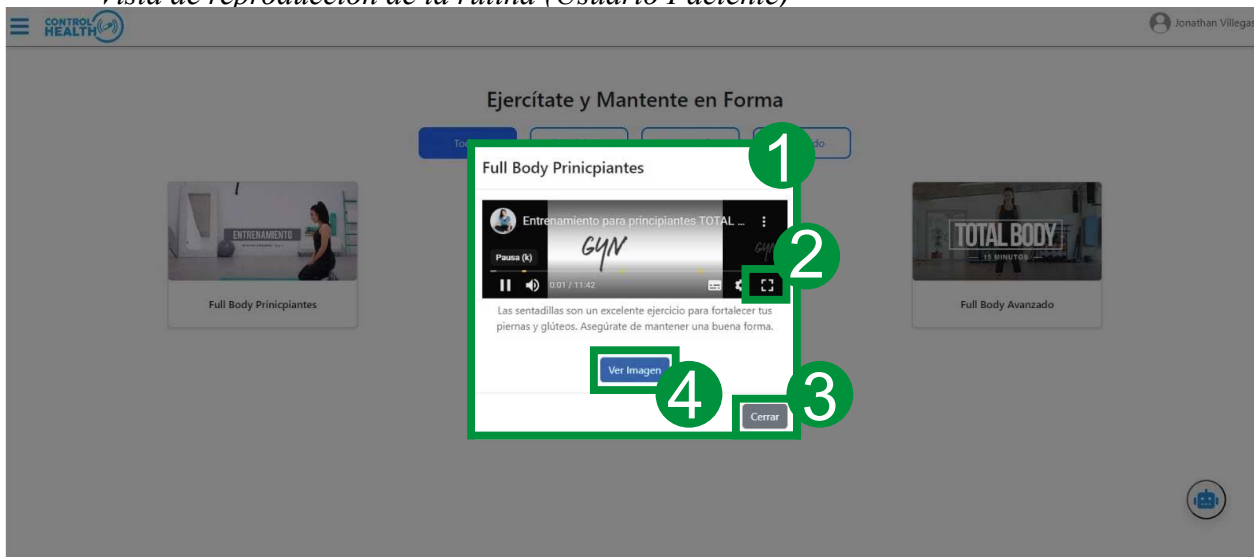
Nota: Esta ilustración muestra la vista de las rutinas físicas para los pacientes y una vista previa de cada ejercicio de la página web y la perspectiva de la aplicación móvil.

Al seleccionar una rutina, se desplegará una vista (1) que presenta información crucial, como las indicaciones previas al inicio de la rutina (2). Después de leer estas indicaciones, solo es necesario hacer clic en el botón "Ver Video" (3) para comenzar a reproducir el video correspondiente a la rutina seleccionada.

Figura 8*Vista de indicaciones de la rutina (Usuario Paciente)*

Nota: Esta ilustración muestra la vista de las rutinas físicas para los pacientes y una vista previa de cada ejercicio de la página web y la perspectiva de la aplicación móvil.

Una vez dado clic en ver video se muestra la ventana del reproductor (1) al dar clic (2) se puede reproducir el video en pantalla completa para mejor comodidad, también está el botón cerrar (3) para salir de ese apartado y el botón (4) para regresar a las indicaciones.

Figura 9*Vista de reproducción de la rutina (Usuario Paciente)*

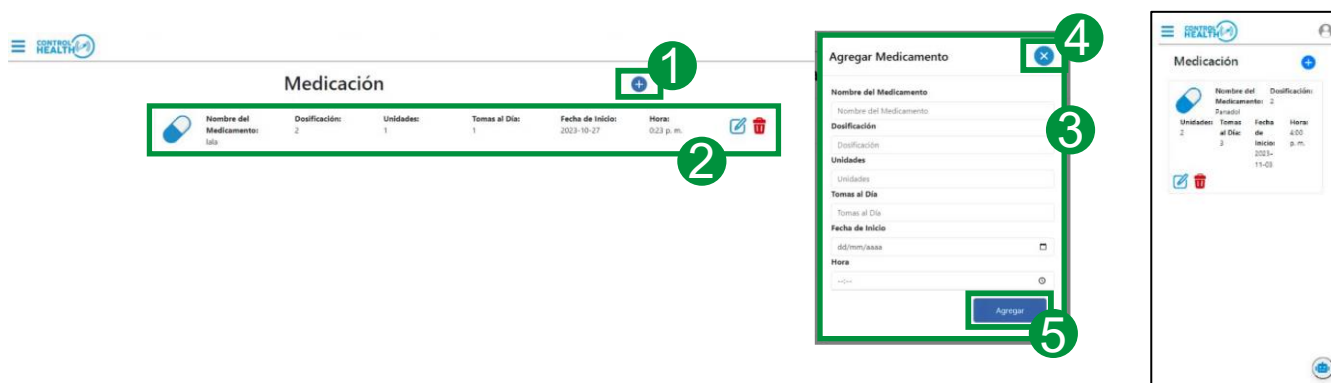
Nota: Esta ilustración muestra la ventana del reproductor del video de las rutinas físicas.

3.7 Vista de control de la medicación usuario Paciente

Cuando se selecciona la opción "Medicación", se despliega la interfaz para agregar un nuevo medicamento (1) y se presentan las alarmas programadas (2), tanto en la versión web como en la versión móvil. Al hacer clic en (1) se abrirá la ventana "Agregar una nueva alarma", donde se deben ingresar los datos necesarios (3). Posteriormente, al hacer clic en "Agregar" (5), la alarma quedará registrada y se mostrará en la sección de alarmas programadas (2). En caso de no desear agregar una nueva alarma, basta con hacer clic en (4) para salir de la ventana correspondiente, asegurando una experiencia consistente tanto en la versión web como en la móvil.

Figura 10

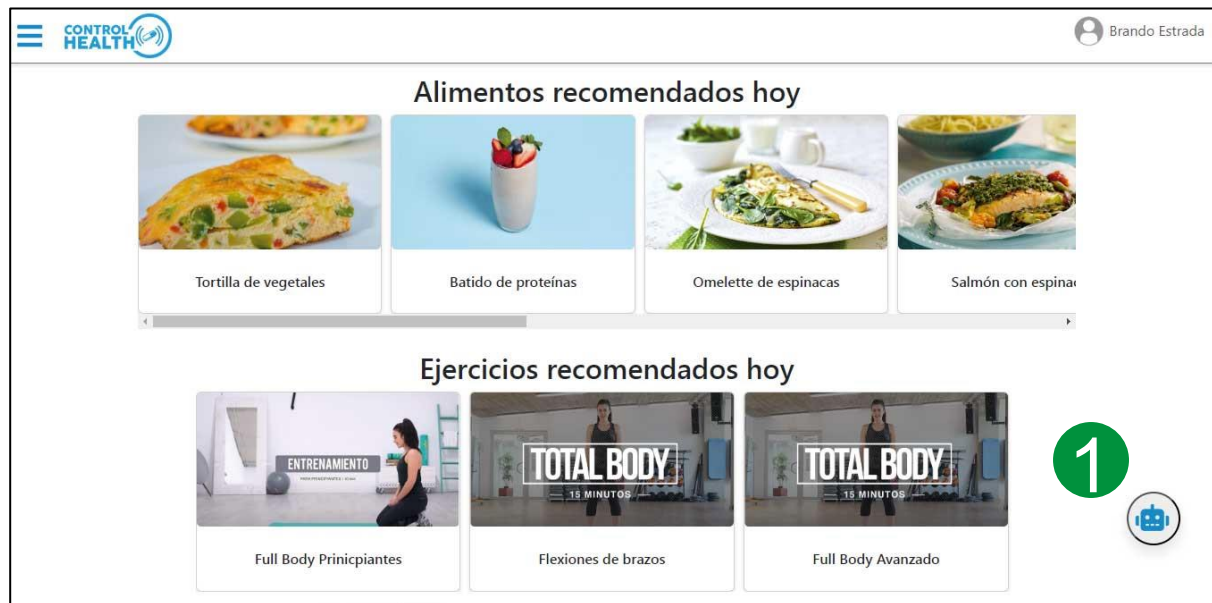
Vista de control de la medicación (Usuario Paciente)



Nota: Esta ilustración muestra la vista del apartado de Medicación para los pacientes de la página web y la perspectiva de la aplicación móvil.

3.8 Vista de chatbot usuario Paciente

Una de las funciones principales del servicio es el chatbot (1), un asistente virtual que ofrece la posibilidad de realizar un prediagnóstico para proporcionar al usuario una idea inicial de sus resultados. No obstante, se recomienda encarecidamente que el usuario consulte con uno de nuestros médicos para obtener una evaluación más precisa y completa.

Figura 11*Vista de chatbot (Usuario Paciente)*

Nota: Esta ilustración muestra la ubicación del ícono del ChatBot en las diferentes vistas.

Después de hacer clic en el chatbot, se iniciará una conversación en la que se te realizarán una serie de preguntas programadas mediante un algoritmo. Al final de esta interacción, el chatbot proporcionará un resultado indicando el porcentaje de probabilidad de tener diabetes. Además, se recomendará agendar una cita (1) con uno de nuestros médicos para obtener una evaluación más precisa y detallada.

Figura 12*Vista del chatbot habilitado (Usuario Paciente)*

Nota: Esta ilustración muestra cómo se realiza una conversación con el ChatBot.

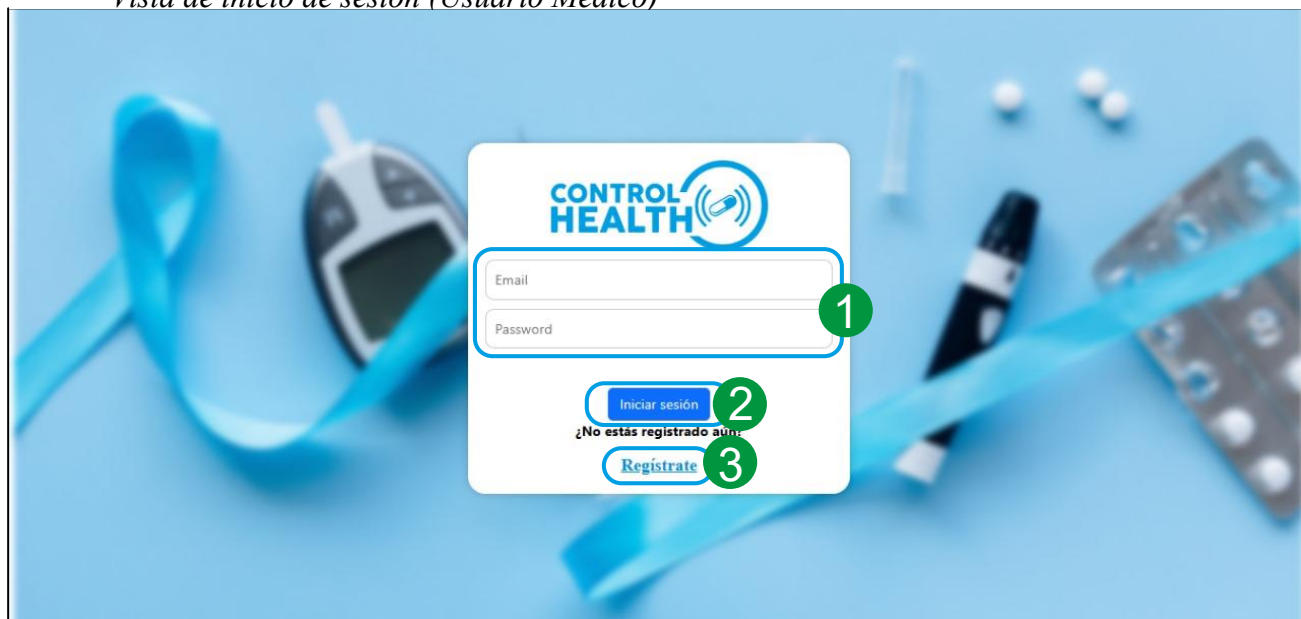
IV. Usuario Médico

4.1 Vista de inicio de sesión usuario Médico

Para acceder a la página, es necesario dirigirse al siguiente enlace: www.controlhealth.com/login. Una vez en la página, se encuentra la opción de ingresar los datos en los campos proporcionados para correo y contraseña (1). Después de introducir la información correspondiente, se debe hacer clic en "Iniciar sesión" (2). En caso de no poseer una cuenta, se puede seleccionar la opción de registrarse ubicada debajo del botón de iniciar sesión (3). Al acceder al apartado de registro, se solicitará seleccionar el rol, ya sea paciente o médico, para completar el proceso de registro.

Figura 13

Vista de inicio de sesión (Usuario Médico)



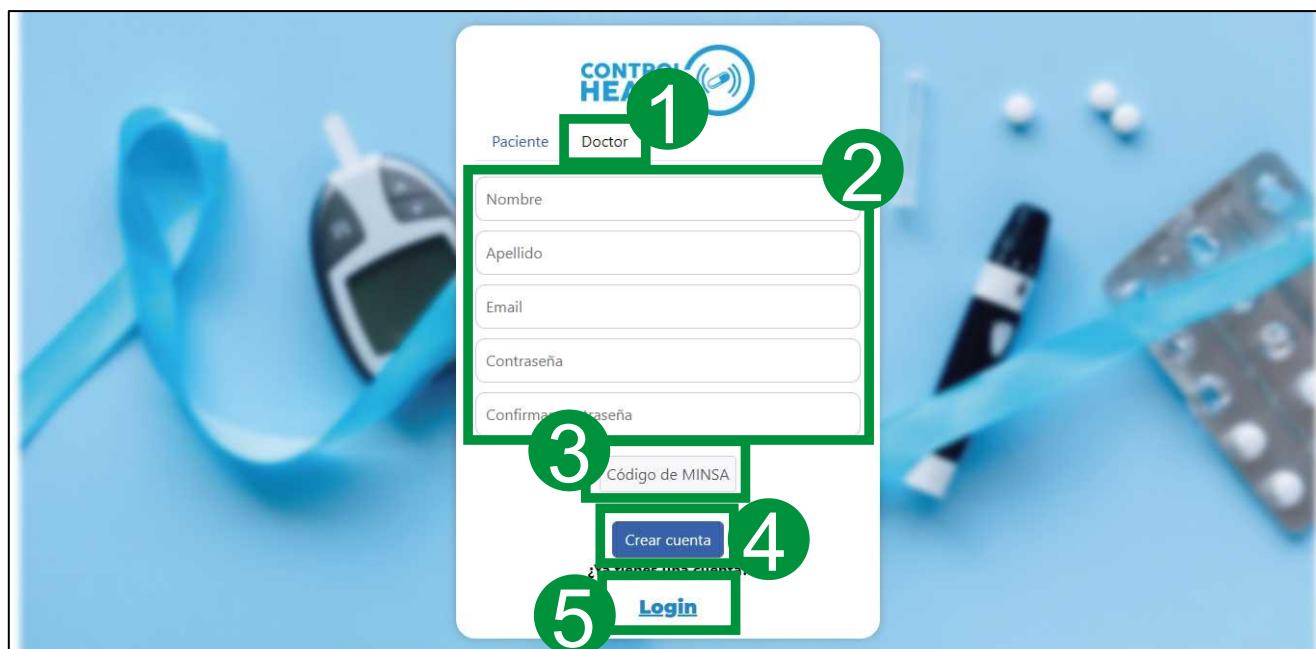
Nota: Esta ilustración muestra la vista el inicio de sesión los usuarios de la página web y la perspectiva de la aplicación móvil.

4.2 Vista de registro de usuario Médico

Una vez que hayas seleccionado tu área correspondiente, en este caso, sería Médico (1), procedes a completar los datos solicitados (2). Sin embargo, para ingresar esta información, se requiere introducir un código único (3) proporcionado por el Minsa. Después de ingresar el código único, solo queda hacer clic en "Crear Cuenta" (4). Posteriormente, solo debes esperar a que tu solicitud de cuenta sea aceptada. En caso de ya disponer de una cuenta, simplemente haz clic en "Login" (5) para acceder con tus datos existentes.

Figura 14

Vista de registro (Usuario Médico)



The image shows a registration form for a medical user, overlaid on a background of medical supplies. The form is titled "CONTROL HE" and has a radio button to select between "Paciente" and "Doctor". The "Doctor" option is selected and marked with a green circle containing the number 1. Below this, there are five input fields: "Nombre", "Apellido", "Email", "Contraseña", and "Confirmar contraseña", all of which are grouped by a green bracket and marked with a green circle containing the number 2. Below these fields is a field for "Código de MINSA", marked with a green circle containing the number 3. At the bottom of the form, there are two buttons: "Crear cuenta" (marked with a green circle containing the number 4) and "Login" (marked with a green circle containing the number 5).

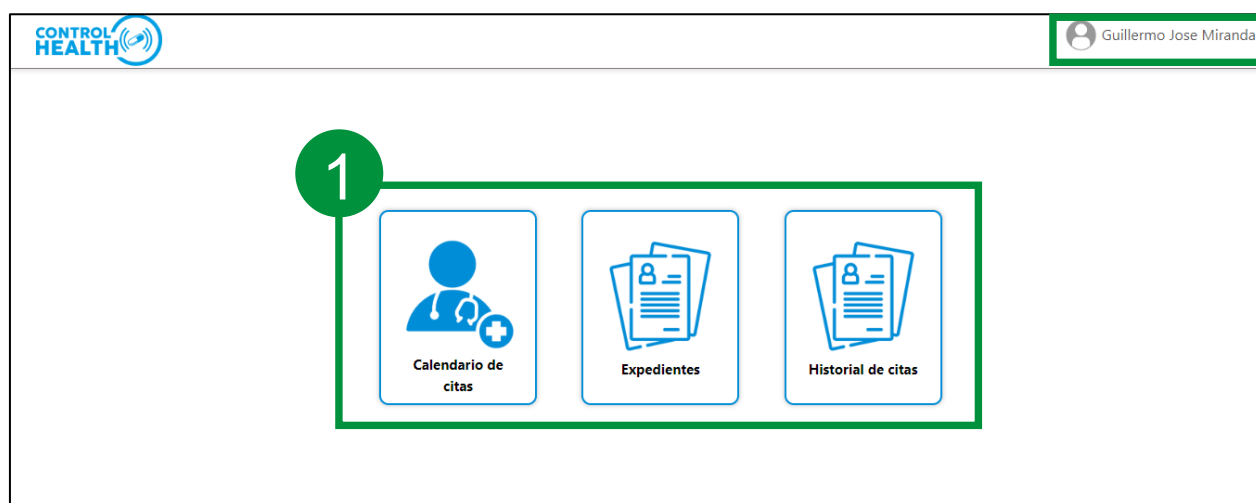
Nota: Esta ilustración muestra la ventana de creación de nuevos usuarios médicos.

4.3 Vista de inicio Médico

Una vez que ha iniciado sesión en el apartado de médico, se accede a la pantalla de inicio que muestra las funciones principales del software (1). En la parte superior de la pantalla (2), se exhibe el nombre del usuario al que pertenece la cuenta utilizada para la navegación en la plataforma web. Esta información proporciona una identificación clara del usuario en la sesión.

Figura 15

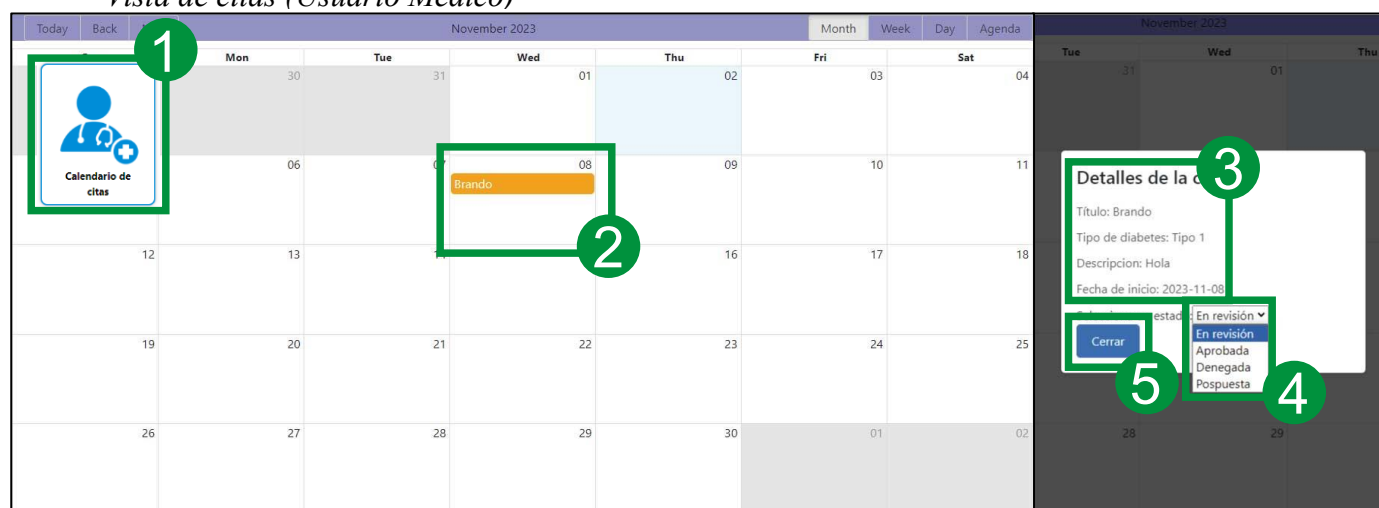
Vista de inicio (Usuario Médico)



Nota: Esta ilustración muestra la vista del menú con el que cuenta el médico en la web.

4.4 Vista de citas de usuario Médico

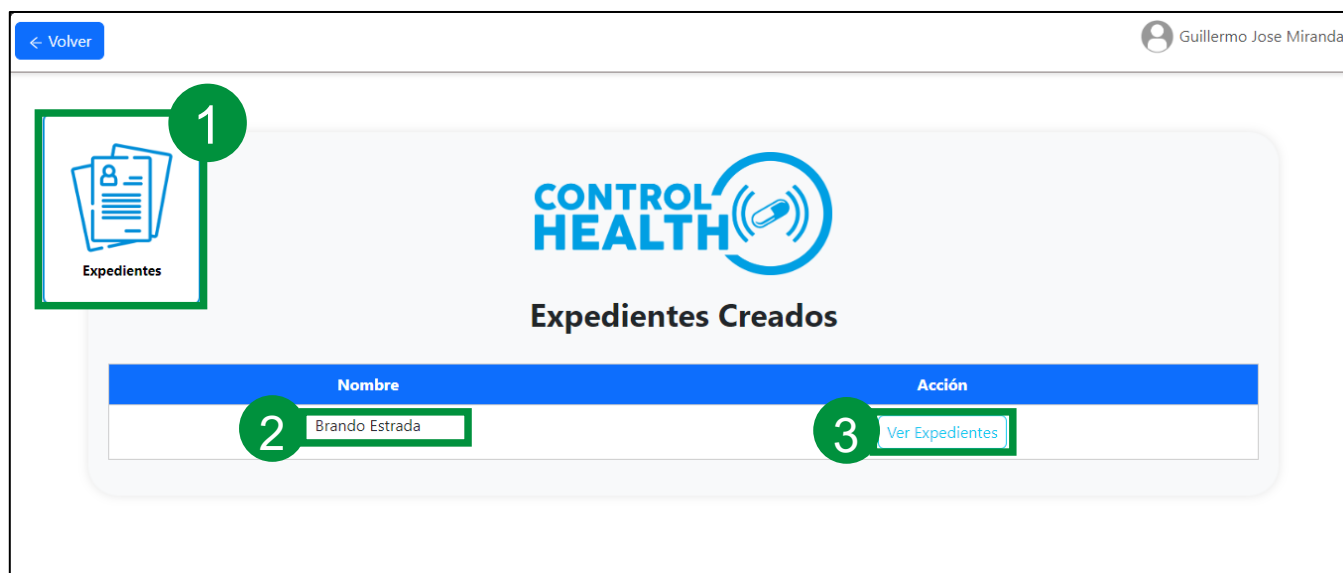
Cuando se selecciona la opción de "Citas" del bloque (1), se accede a un apartado que presenta un calendario con las citas agendadas (2). Este calendario proporciona una visión organizada de las citas programadas, permitiendo al médico llevar un control eficiente. Al hacer clic en la fecha específica o en la sección designada (2), se obtendrán más detalles en la sección (3) sobre las citas agendadas para ese día o periodo de tiempo determinado, se selecciona el estado de la cita en la sección (4). Una vez seleccionado el estado de la cita se le da clic en la sección cerrar (5).

Figura 16*Vista de citas (Usuario Médico)*

Nota: Esta ilustración muestra la ventana del calendario del médico donde puede revisar sus citas próximas, futuras o agregar una nueva cita.

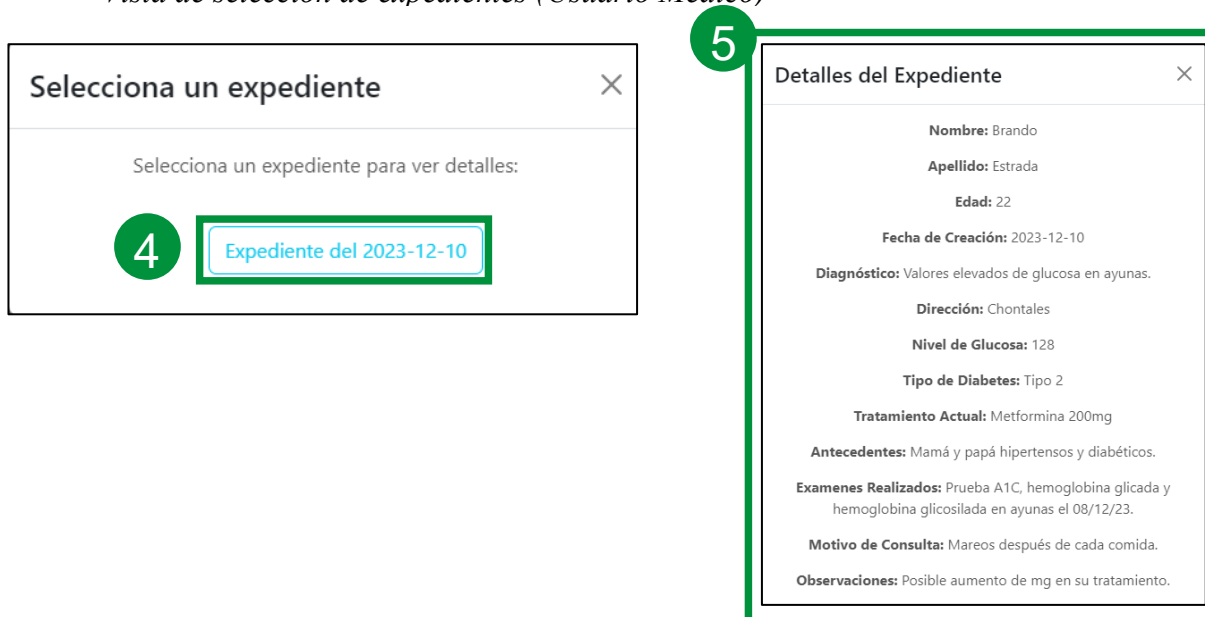
4.5 Vista de expedientes usuario Médico

Cuando se selecciona la opción “Expedientes” del [Ver Expedientes](#) bloque (1), Se mostrar una vista con el nombre del paciente que ya tiene un expediente creado en la sección (2). Luego se da clic en el botón para visualizar los expedientes creados del paciente en la sección (3).

Figura 17*Vista de expedientes (Usuario Médico)*

Nota: Esta ilustración muestra los expedientes que el doctor tiene a su cargo.

Al hacer clic en la sección (3) se abrirá una ventana en la que se encuentran los expedientes creados del paciente en la sección (4) si le damos clic a ese expediente creado, se mostrara los detalles del expediente en la sección (5).

Figura 18*Vista de selección de expedientes (Usuario Médico)*

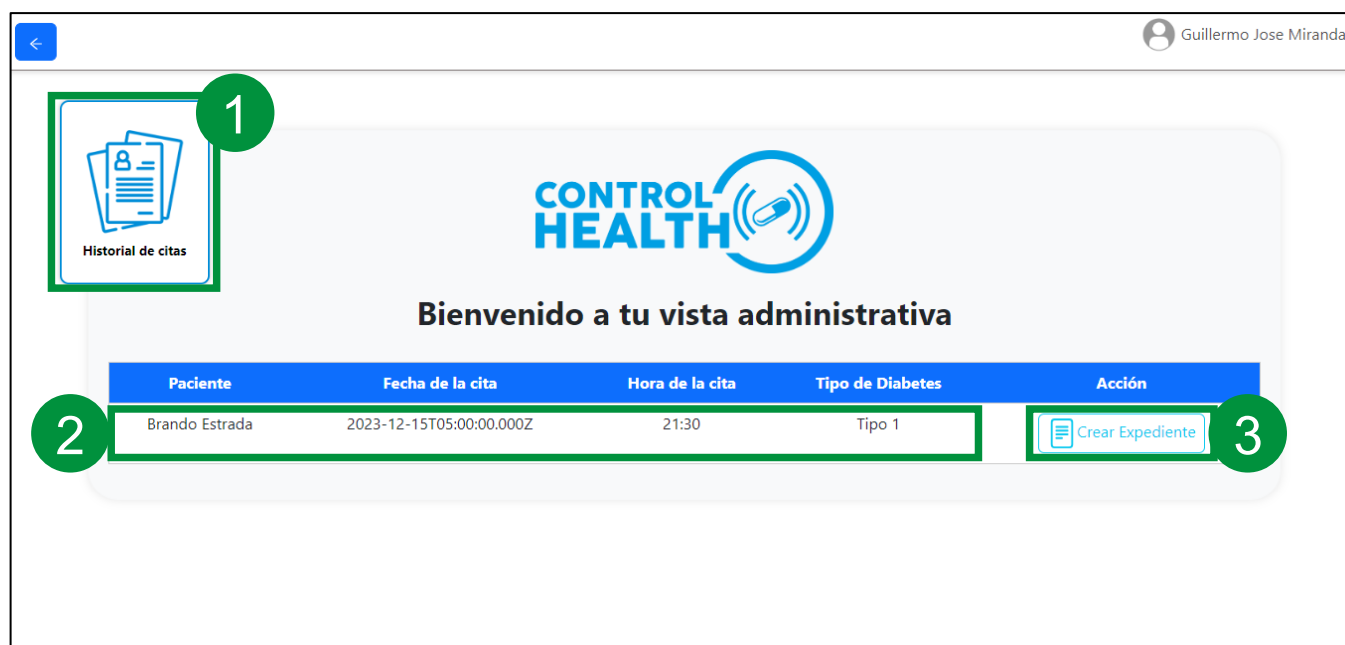
Nota: Esta ilustración detalla la información que se mostrara si se le da clic a la sección (3) en la que se muestra los expedientes creados del paciente y la información que almacena cada expediente.

4.6 Vista de historial de citas usuario Médico

Cuando se selecciona la opción Historial de Citas del bloque (1), se mostrará una tabla con todos los pacientes que han solicitado citas en la sección (2). En esta tabla se permite la creación de expediente en la sección (3).

Figura 19

Vista de historial de citas (Usuario Médico)



Nota: Esta ilustración muestra el historial de citas que el médico actualmente activo tiene a su cargo, además de la creación de expediente para los pacientes.

Si damos clic en “Crear expedientes” en la sección (3) se mostrará un formulario en la sección (4), en el cual se deberá de llenar la información del paciente al que se le está creando el expediente, cuando todos los campos del formulario estén llenos se procede a dar clic en el botón “Enviar Expediente” en la sección (5)

Figura 20

Vista de formulario de expedientes (Usuario Médico)

4 Llenar Expediente

Nombre del paciente: Apellido del paciente:

Edad del paciente:

Departamento:

Nivel de Glucosa:

Tipo de Diabetes:

Tratamiento Actual:

Diagnóstico:

Motivo de consulta:

Antecedentes:

Exámenes Realizados:

Observaciones:

5

Nota: Esta ilustración muestra el formulario que se debe de llenar para crear el expediente de un paciente.

V. Usuario Administrador

5.1 Vista menú (Administrador)

Dentro de este menú, se encuentran varias secciones importantes. En primer lugar, está la opción de obtener datos estadísticos (1), que proporciona información sobre la participación de los usuarios en la web. Luego, está la sección de "Médico" (2), donde se puede aprobar o rechazar la solicitud de un nuevo médico. La sección de "Usuarios" (3) permite visualizar los nuevos registros de usuarios. Finalmente, en la sección (4), se encuentra la opción para cerrar sesión en la cuenta de administrador. Estas funciones son cruciales para la gestión eficiente y la toma de decisiones en la plataforma.

Figura 21

Vista de menú (Administrador)

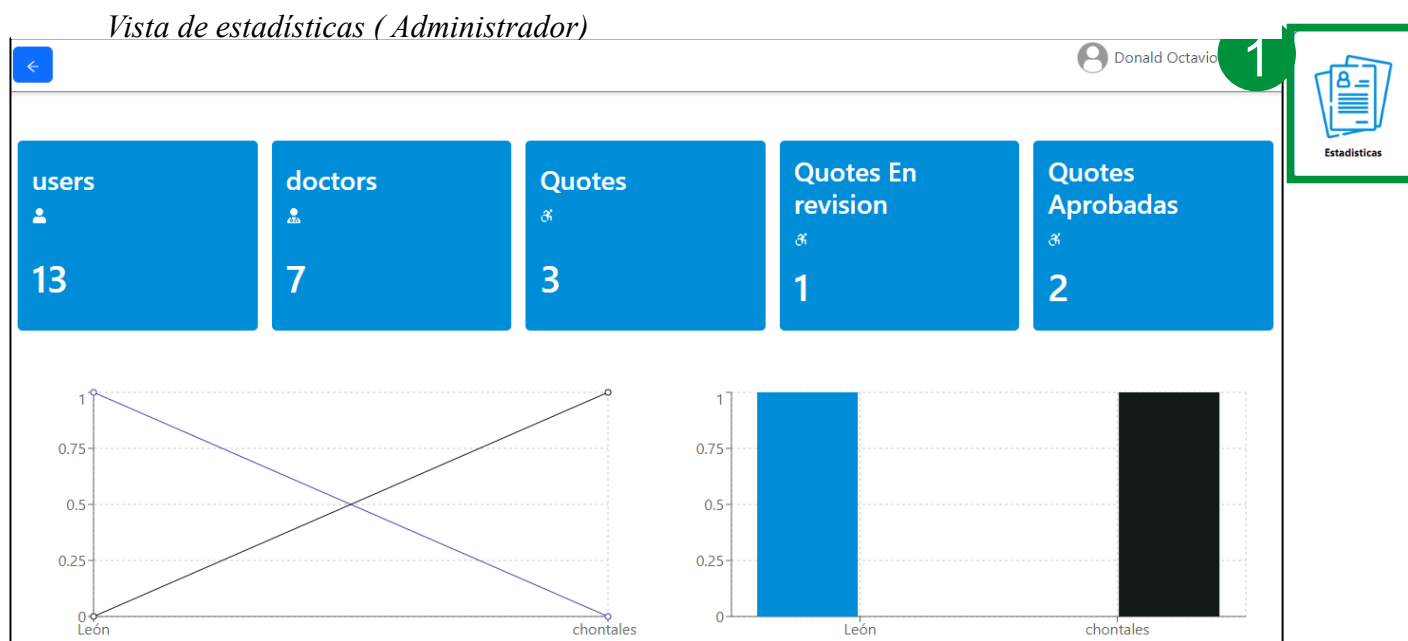


Nota: Esta ilustración muestra las opciones principales en las que el administrador puede llevar el control de la aplicación web.

5.2 Vista de estadísticas (Administrador)

En este apartado se presentan los registros de usuarios, incluyendo pacientes, doctores y citas. Además, se proporciona información sobre los usuarios actualmente activos en la web. Esta sección ofrece una visión general de la base de usuarios, permitiendo un seguimiento efectivo de la participación y presencia en la plataforma.

Figura 22



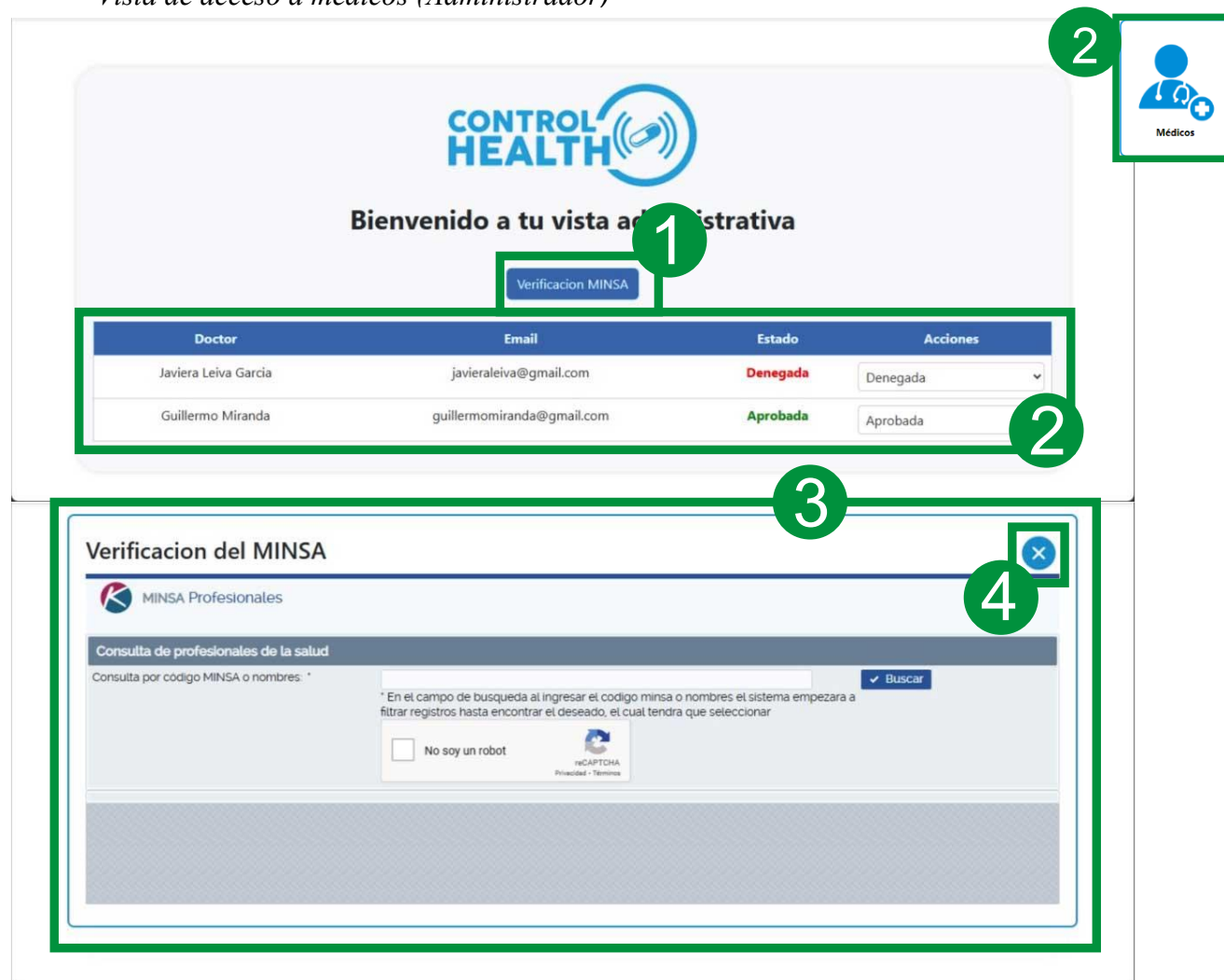
Nota: Esta ilustración muestra la ventana de las estadísticas obtenidas en la aplicación web.

5.3 Vista de acceso a médicos (Administrador)

Al acceder al apartado médico, se despliega una ventana que presenta a los médicos (2) que están solicitando unirse a la plataforma. En el apartado (1), hay un botón que muestra los códigos solicitados por los médicos, los cuales sirven para verificar (3) la autenticidad del médico. Finalmente, se encuentra el botón "Cerrar" (4), que permite salir de este apartado. Estas funciones son esenciales para gestionar las solicitudes de médicos y garantizar la autenticidad de su participación en la plataforma.

Figura 23

Vista de acceso a médicos (Administrador)



Nota: Esta ilustración muestra la ventana para aceptar, rechazar y la manera de verificar al médico.

5.4 Vista de roles de usuarios (Administrador)

Al acceder al apartado Usuarios (3), se muestra una tabla en la que se puede asignar roles por medio del cuadro de selección en la sección (1).

Figura 24

Vista de roles (Administrador)

3

Donald Octavio Alvarez

CONTROL HEALTH

Bienvenido a tu vista administrativa

Usuario	Email	Estado	Acciones
Jonathan Villegas	doc1@gmail.com	admin	admin
Donald Octavio Alvarez	donaldoctavio0210@gmail.com	admin	admin
Brando Estrada	brandoelIngeniero@nic.es	patient	patient
Jonathan Villegas	usuario3@gmail.com	patient	patient

1

Nota: Esta ilustración muestra como el administrador asigna los roles de los usuarios es decir si es paciente o se le otorgan permisos de administrador.



Manual Técnico Aplicación Control Health

Para el control y gestión de la medicación de la diabetes

Tabla de contenido

I.	Introducción	1
II.	Requerimientos Técnicos.....	1
2.1	Requisitos de Instalación del Sistema	1
2.1.1	<i>Requisitos de software compatibles con navegadores de la web</i>	<i>1</i>
2.1.2	<i>Requisitos mínimos de hardware desde una PC compatible con navegadores en la Web</i>	<i>2</i>
III.	Frameworks y Estándares utilizados	2
IV.	Estructura de la aplicación	4
4.1	Componentes	5
4.2	Views	6
4.3	Módulo Assets.....	8
V.	Bases de datos.....	10
5.1	Estructura de la Base de Datos	11
VI.	Arquitectura Utilizada.....	14
VII.	Dependencias utilizadas.....	15
VIII.	Referencias	16

Índice de Figuras

Figura 1 Estructura de la aplicación	4
Figura 2 Carpeta Components	5
Figura 3 Views	6
Figura 4 Módulo Assets	8
Figura 5 Modulo Store	9
Figura 6 Firebase Servicios	10
Figura 7 JSON User	11
Figura 8 JSON Quotes	11
Figura 9 JSON Medication	12
Figura 10 JSON Diet	12
Figura 11 JSON Routines	13
Figura 12 Arquitectura Cliente-Servidor	14
Figura 13 Dependencias utilizadas	15

I. Introducción

Control Health es una aplicación web responsiva móvil diseñada para brindar un control efectivo sobre la Diabetes. Sus funciones abarcan desde el seguimiento de tu medicación hasta la educación en Diabetes, nutrición y ejercicio. Además, conecta al paciente con médicos diabetólogos a través de un práctico chatbot para programar citas médicas con facilidad.

Para alcanzar estos propósitos se ha hecho uso del framework de JavaScript llamado React JS, el cual se adecúa a las nuevas necesidades de las aplicaciones web.

En cuanto a este manual se ha considerado incluir todos los aspectos técnicos necesarios para el manejo y control de la web.

II. Requerimientos Técnicos

2.1 Requisitos de Instalación del Sistema

Para el correcto funcionamiento de la aplicación es necesario tener cubiertos una serie de requisitos, tanto de hardware como software, mencionados a continuación:

2.1.1 Requisitos de software compatibles con navegadores de la web

- **SO:** Android 4
- **Procesador:** Octa-core 1.2 GHz
- **Memoria:** 2 MB de RAM y 50 MB ROM
- **Otros:** Estos requisitos son los mínimos se puede usar en equipos actuales.

2.1.2 Requisitos mínimos de hardware desde una PC compatible con navegadores en la Web

- **SO:** Windows 7
- **Procesador:** 1.0 GHz
- **Memoria:** 256 MB de RAM
- **Gráficos:** Tarjeta compatible con DirectX y con 64 MB*
- **DirectX®:** 8.1 o superior
- **Disco Duro:** 600 MB de espacio libre
- **Sonido:** Compatible con DirectX
- **Otros:** Estos requisitos son los mínimos se puede usar en equipos actuales.

III. Frameworks y Estándares utilizados



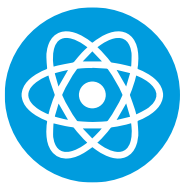
Firebase es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. Ofrece una variedad de servicios y herramientas que facilitan el desarrollo, la gestión y el despliegue de aplicaciones. (Mora, 2020)



Node.js es un entorno de tiempo de ejecución (runtime) de código abierto que permite a los desarrolladores construir aplicaciones de servidor y aplicaciones web en el lado del servidor utilizando JavaScript. A diferencia de JavaScript en el navegador, que se ejecuta en el lado del cliente, Node.js permite ejecutar código JavaScript en el lado del servidor, lo que lo convierte en una tecnología versátil para el desarrollo web y aplicaciones de red. (Simões, 2021)



Redux es una biblioteca de gestión de estado de código abierto para aplicaciones JavaScript, en particular para aplicaciones basadas en React. Fue desarrollada originalmente por Dan Abramov y Andrew Clark. Redux proporciona un patrón de arquitectura de datos predecible y un flujo unidireccional de datos, lo que facilita la administración del estado de una aplicación de manera más organizada y escalable. (Arias, 2023)



React también conocido como React.js, es una biblioteca de JavaScript de código abierto desarrollada por Facebook que se utiliza para crear interfaces de usuario (UI) interactivas y componentes reutilizables en aplicaciones web. React se centra en la creación eficiente y la actualización de la interfaz de usuario, lo que lo hace ideal para aplicaciones de una sola página (SPA) y aplicaciones web que requieren una experiencia de usuario rápida y dinámica. (A., 2023)

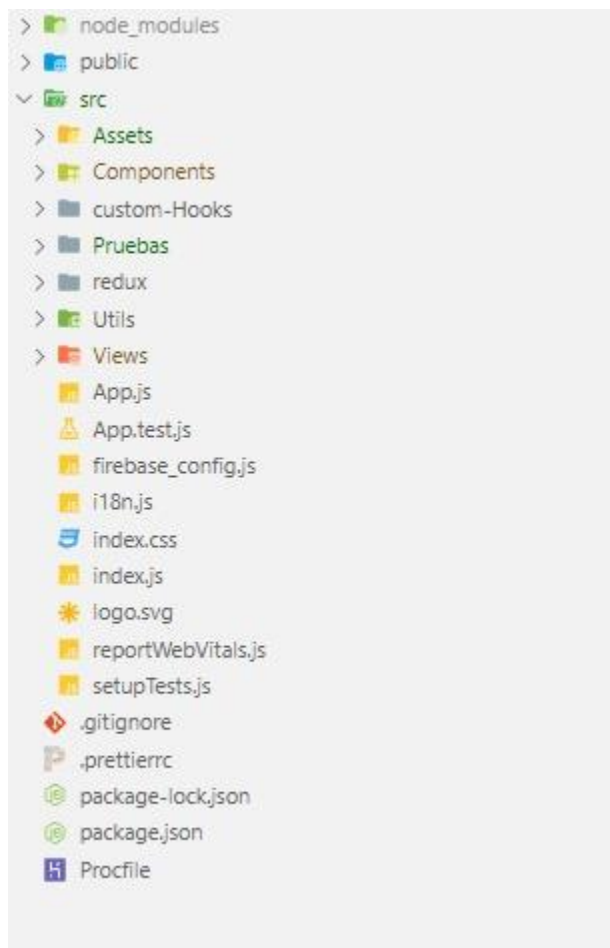


React Bootstrap es un marco de diseño de código abierto (framework) desarrollado por Twitter, y ahora es mantenido por la comunidad de desarrollo de código abierto. Se utiliza para facilitar y acelerar el proceso de diseño y desarrollo de sitios web y aplicaciones web. Bootstrap proporciona una colección de componentes, estilos, y herramientas predefinidas que permiten crear interfaces de usuario atractivas y receptivas de manera más eficiente. (Desarrollador Fantasma, 2023)

IV. Estructura de la aplicación

Figura 1

Estructura de la aplicación



La carpeta "src" enriquece la aplicación con una estructura organizada y funcional. En su interior, aloja los siguientes módulos fundamentales:

Assets: Este módulo es un tesoro de recursos visuales, que incluye íconos e imágenes que dan vida a la riqueza multimedia de la aplicación.

Components: Aquí reside la colección de elementos de código reutilizable, diseñados para optimizar la eficiencia en el desarrollo y la coherencia en la interfaz de usuario.

Custom-Hooks: Este módulo representa la conexión vital con Firebase en el proceso de autenticación de usuarios, lo que brinda un sólido cimiento para la seguridad y la gestión de usuarios.

Redux: En este espacio, las variables de estado encuentran su hogar. Aquí se ejerce el control maestro

sobre la gestión del estado de la aplicación, asegurando cohesión y previsibilidad.

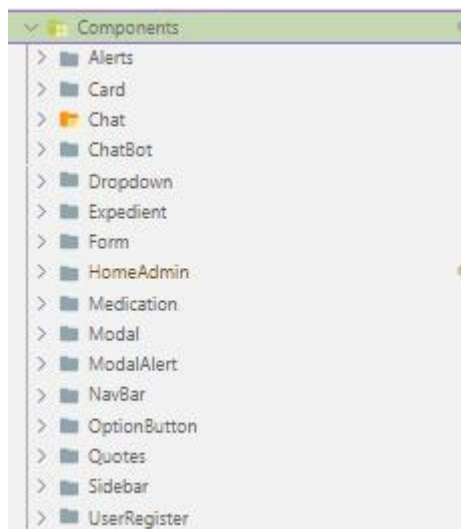
Utils: El módulo "Utils" se encarga de albergar datos cruciales, rutas de navegación y estados de citas que alimentan la aplicación, un recurso fundamental para el funcionamiento integral.

Views: Este módulo cumple una función esencial al rastrear y gestionar las vistas a medida que se desenvuelven en la aplicación. Además, controla el aspecto visual de cada una de estas vistas mediante la administración de sus estilos CSS correspondientes.

4.1 Componentes

Figura 2

Carpeta Components



Esta sección es un tesoro de componentes que desempeñan un papel esencial en cada rincón de la aplicación. Aquí, una amplia gama de elementos comparte un espacio dedicado:

Alerts: En este componente, las alertas toman el centro del escenario, brindando una experiencia de usuario informada y eficaz.

Card: Un versátil componente que se presta tanto para la presentación de medicamentos como para exhibir las citas disponibles en la aplicación.

Chatbot: La magia del Chatbot se manifiesta aquí, a través del componente "bot", donde se guarda la codificación que impulsa esta funcionalidad.

Dropdown: La visualización del perfil y la opción de cierre de sesión encuentran su hogar en este componente, que simplifica la navegación del usuario.

HomeAdmin: En este robusto componente, se revela la vista administrativa, destinada al administrador de la aplicación. Aquí, los usuarios y gráficos estadísticos se despliegan con claridad.

Medication: En esta zona, las CARDS que almacenan información sobre medicamentos ocupan un lugar destacado, proporcionando datos cruciales.

Modal: Un componente polivalente que simplifica la presentación de contenido adicional o interacciones sin necesidad de cambiar de pantalla, mejorando la experiencia del usuario.

ModalAlert: Este componente se reserva para mensajes informativos y notificaciones críticas que demandan la atención del usuario, asegurando una comunicación efectiva.

NavBar: La "NavBar" o barra de navegación se erige como un componente esencial para facilitar la navegación y el acceso a diferentes secciones de la aplicación.

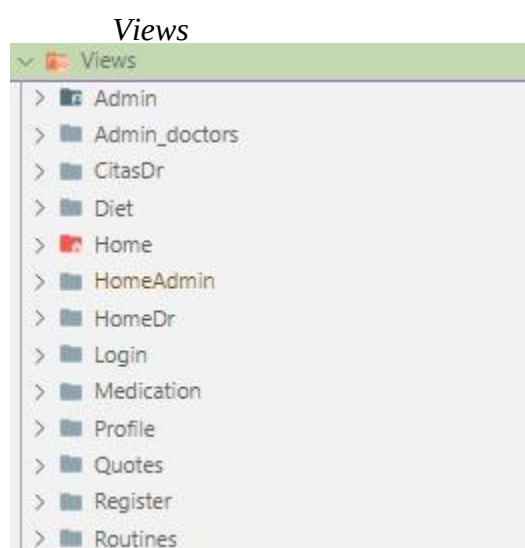
OptionButton: Un componente estratégico que allana el camino en la vista del médico, permitiendo rutas y acciones específicas.

Quotes: En este componente, se gestiona con precisión la programación y administración de las citas para los usuarios pacientes.

UserRegister: Aquí se orquesta el proceso mediante el cual un usuario da vida a su cuenta o perfil en la aplicación, habilitando un acceso personalizado y funcional.

4.2 Views

Figura 3



Admin: Dentro de este módulo, se alojan las vistas dedicadas a los usuarios con privilegios administrativos.

Incluye "**Admin_users**," que proporciona herramientas para la gestión de usuarios, y "**Home Admin**," que presenta gráficos y datos esenciales.

Admin_doctors: En esta vista, el administrador realiza un importante proceso de verificación. Se asegura de la existencia de los médicos en la página del MINSA, garantizando la autenticidad de los usuarios doctores.

CitasDr: Los médicos encuentran en esta vista la capacidad de gestionar las citas. Pueden programar y organizar las citas de manera eficiente a través de un calendario incorporado.

Diet: Esta vista proporciona información detallada sobre dietas, incluyendo ingredientes y métodos de preparación. Los usuarios pueden acceder a recursos valiosos para mantener un estilo de vida saludable.

Home: Se trata de la vista principal y punto de partida para todos los usuarios. Ofrece acceso central a las características esenciales de la aplicación.

HomeAdmin: Diseñada específicamente para los administradores, esta vista se convierte en un centro de control. Permite la gestión de usuarios y proporciona visualizaciones claras de datos estadísticos.

Login: La vista de inicio de sesión actúa como la puerta de entrada común para todos los usuarios. Aquí, se inicia la experiencia de usuario y se accede a las cuentas personales.

Medication: En esta vista, los usuarios exploran tarjetas informativas ("CARDS") que detallan la medicación disponible. Los pacientes pueden acceder a información relevante sobre medicamentos proporcionados por otros usuarios.

Profile: Los perfiles de usuario se presentan en esta vista, permitiendo a los usuarios ver y editar su información personal. Es un componente clave para la personalización de la experiencia del usuario.

Quotes: En esta vista, los detalles de las citas solicitadas se muestran de manera organizada en tarjetas ("CARDS"). Los usuarios pueden acceder a información crítica sobre sus citas.

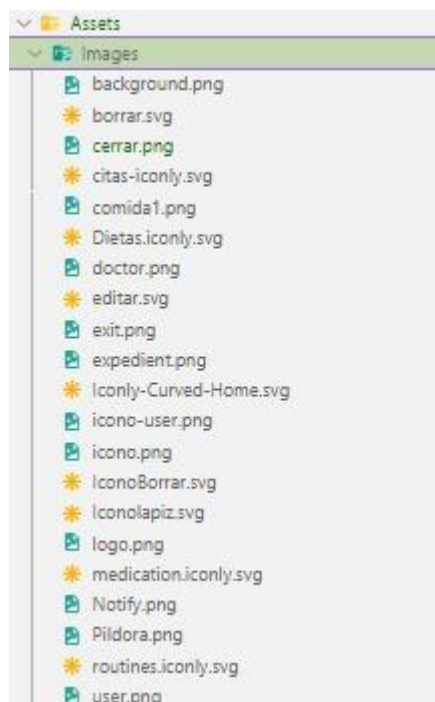
Register: En esta vista, se presentan los campos necesarios para completar el proceso de registro de usuario. Los nuevos registros permiten a los usuarios crear perfiles personalizados.

Routines: Los usuarios pueden explorar rutinas categorizadas y acceder a videos tutoriales demostrativos en esta vista. Esto fomenta la incorporación de hábitos saludables y la adopción de un estilo de vida más equilibrado.

4.3 Módulo Assets

Figura 4

Módulo Assets



Dentro de este módulo, se establece un repositorio integral de elementos multimedia que desempeñan un papel fundamental en la funcionalidad y estética de la aplicación. Este espacio se caracteriza por su organización cuidadosa en dos secciones distintas:

La primera sección se dedica a los íconos y las imágenes que nutren y enriquecen el carrusel visual de la aplicación, agregando un toque de elegancia y dinamismo a la experiencia del usuario.

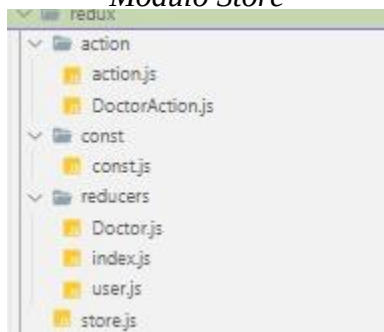
La segunda sección, por su parte, se reserva para otros archivos de carácter público, accesibles y compartidos, que desempeñan un papel crucial en la interacción y funcionalidad general de la aplicación.

Este enfoque en la accesibilidad y la versatilidad asegura que los recursos multimedia estén disponibles cuando y donde se necesiten en la aplicación, facilitando una experiencia integral y rica para los usuarios.

4.4 Modulo Store

Figura 5

Modulo Store



La función esencial del "store" en el contexto de la aplicación basada en Redux es la gestión integral del estado global de la aplicación. Actúa como un núcleo de control, desempeñando un papel central en dos aspectos cruciales:

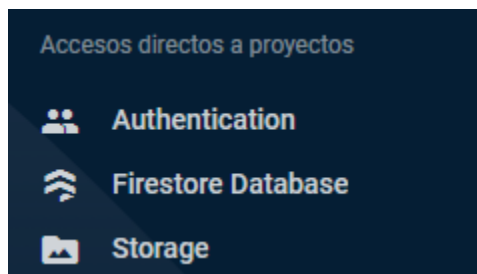
Almacenamiento y Acceso del Estado Global: El "store" es el custodio del estado de la aplicación en su totalidad. Almacena datos, configuraciones y variables críticas, y al mismo tiempo, proporciona métodos para acceder y consultar este estado. Esto garantiza que los componentes de la aplicación puedan acceder a la información necesaria para funcionar y mostrar datos relevantes.

Coordinación de Cambios de Estado: Cuando las acciones se desencadenan en la aplicación, el "store" juega un papel clave en la orquestación de cambios en el estado en respuesta a esas acciones. Los reductores, que son funciones puras asociadas al "store," determinan cómo se debe actualizar el estado en función de las acciones. Esto asegura que el estado de la aplicación cambie de manera coherente y predecible, lo que simplifica la depuración y el mantenimiento.

V. Bases de datos

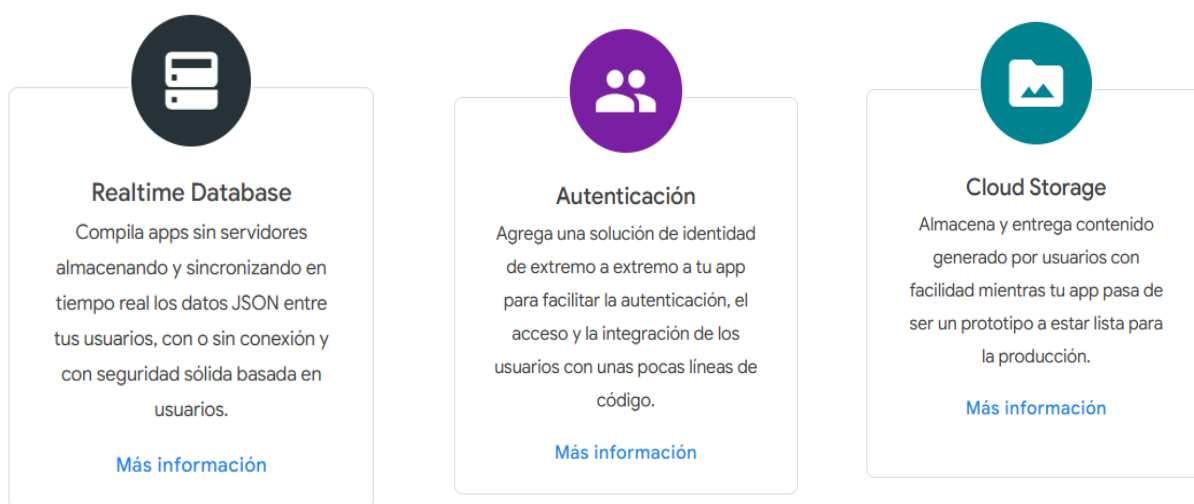
Figura 6

Firestore Servicios



Los servicios proporcionados por Firebase desempeñan un papel crucial en el funcionamiento integral de la aplicación, destacándose en tres áreas clave. En primer lugar, la autenticación de usuarios se erige como la piedra angular de la seguridad y el acceso controlado a la plataforma. Seguidamente, Firestore se

consagra como la robusta base de datos que gestiona y organiza los datos de la aplicación de manera eficiente. Por último, Storage se encarga de proporcionar un espacio fiable y escalable para el alojamiento de imágenes, garantizando una experiencia fluida y rápida en la visualización de contenido visual. Estos tres pilares combinados se unen para ofrecer una experiencia de usuario excepcional y una gestión de datos eficaz en la aplicación.



5.1 Estructura de la Base de Datos

Figura 7

JSON User



```
"User": [
  {
    "departamento": "",
    "edad": "",
    "email": "",
    "photoURL": "",
    "tel": "",
    "typeUser": "",
    "uid": "",
    "userName": ""
  }
],
```

User proporcionado representa la estructura de datos utilizada en la aplicación "Control Health" para gestionar la información de los usuarios. Contiene campos como "departamento" (ubicación), "edad", "email" (correo electrónico), "photoURL" (foto de perfil), "tel" (número de teléfono), "typeUser" (tipo de usuario), "uid" (identificador único), y "userName" (nombre de usuario). El campo "typeUser" se utiliza para validar el rol del usuario, que puede ser "patient" (paciente), "doctor" (médico) o "admin" (administrador), lo que determina sus permisos y acceso en la aplicación.

Figura 8

JSON Quotes



```
"Quotes": [
  {
    "date": "",
    "description": "",
    "dni": "",
    "doctor": "",
    "email": "",
    "fileUrl": "",
    "id": "",
    "typeUser": "",
    "uid": "",
    "userName": "",
    "validated": "",
    "name": "",
    "numberPhone": "",
    "typeDiabetes": "",
    "status": "",
    "uid": ""
  }
],
```

Quotes se utiliza en la aplicación "Control Health" para almacenar información sobre citas médicas y registros de pacientes. Contiene campos que incluyen la fecha, descripción, identificación del paciente, médico, correo electrónico, archivos relacionados, identificador único y más.

Figura 9

JSON Medication

```
"Medication": [
  {
    "data": "",
    "medication": "",
    "dosificacion": "",
    "fechadeinicio": "",
    "hora": "",
    "nombreMedicamento": "",
    "tomasDelDia": "",
    "unidades": "",
    "uid": ""
  }
],
```

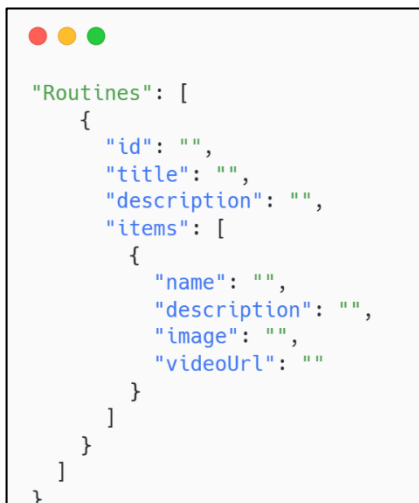
Medication se utiliza en la aplicación "Control Health" para almacenar información sobre la medicación de los pacientes. Incluye campos como la fecha de inicio, la dosificación, el nombre del medicamento, la hora de administración, la cantidad de tomas diarias, y un identificador único (uid).

Figura 10

JSON Diet

```
"Diet": [
  {
    "id": "",
    "title": "",
    "description": "",
    "items": [
      {
        "name": "",
        "preparation": "",
        "Ingredientes": "",
        "image": ""
      }
    ]
  }
],
```

Diet se utiliza en la aplicación "Control Health" para almacenar información sobre las dietas nutricionales. Contiene campos como "id" (identificador), "title" (título), "description" (descripción), y "items" (elementos de la dieta) que incluyen el nombre del alimento, la preparación, los ingredientes y la URL de la imagen relacionada.

Figura 11*JSON Routines*

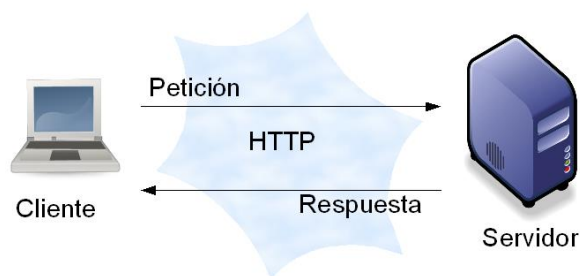
```
{
  "Routines": [
    {
      "id": "",
      "title": "",
      "description": "",
      "items": [
        {
          "name": "",
          "description": "",
          "image": "",
          "videoUrl": ""
        }
      ]
    }
  ]
}
```

Routines se utiliza en la aplicación "Control Health" para gestionar información sobre rutinas de actividad física. Contiene campos como "id" (identificador), "title" (título), "description" (descripción) y "items" (elementos de la rutina), que incluyen el nombre de la actividad, una descripción, una URL de imagen y una URL de video relacionada.

VI. Arquitectura Utilizada

Figura 12

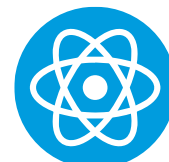
Arquitectura Cliente-Servidor



La arquitectura cliente-servidor utilizada en la aplicación "Control Health" se basa en una estructura que separa claramente las responsabilidades entre el cliente (la parte de la aplicación que se ejecuta en el dispositivo del usuario) y el servidor (el componente remoto que almacena y administra los datos y la lógica de la aplicación). Esta arquitectura permite una

comunicación eficiente y segura entre ambas partes, lo que es esencial para el funcionamiento de la aplicación. (edgarbc.wordpress, 2014)

En el lado del cliente, se utiliza React.js, un framework de desarrollo de interfaz de usuario que permite la creación de componentes interactivos y altamente responsivos.



Además, se implementa Redux, una biblioteca de gestión de estado, que asegura que los datos se mantengan de manera consistente en toda la aplicación, lo que es fundamental para la sincronización de la información entre el cliente y el servidor.

Por otro lado, en el lado del servidor, se aprovecha Firebase, una plataforma de desarrollo que proporciona una infraestructura escalable y servicios en la nube. Firebase se encarga de gestionar la base de datos en tiempo real, la autenticación de usuarios y otras funcionalidades de backend, lo que permite el almacenamiento y recuperación eficientes de datos, así como la autenticación segura de los usuarios.

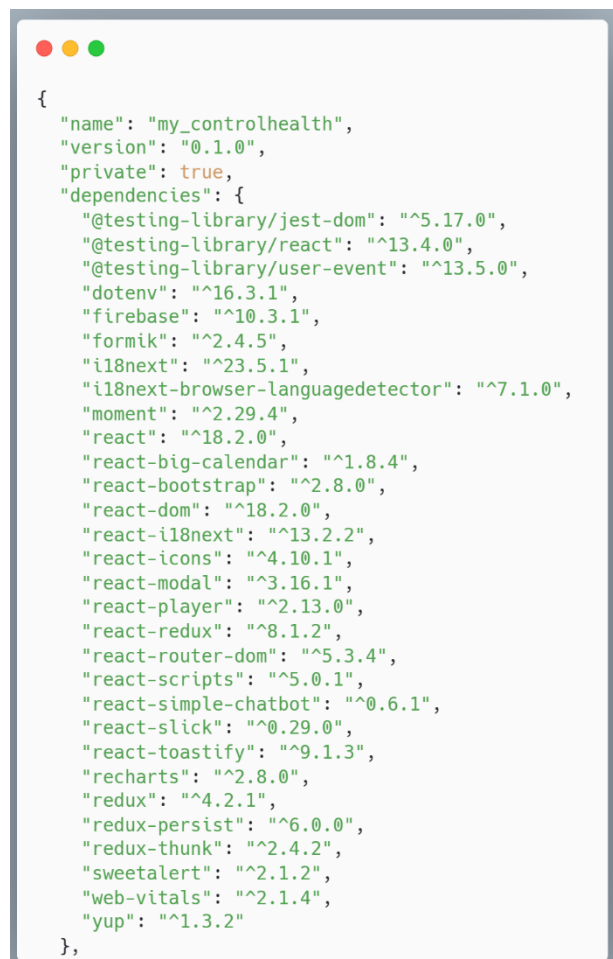


La comunicación entre el cliente y el servidor se realiza a través de protocolos seguros, como HTTPS, lo que garantiza la privacidad y la integridad de los datos transmitidos. Esta arquitectura cliente-servidor permite a los usuarios de "Control Health" acceder y actualizar sus datos de manera confiable, al tiempo que asegura que la aplicación funcione de manera rápida y efectiva, brindando una experiencia de usuario óptima.

VII. Dependencias utilizadas

Figura 13

Dependencias utilizadas



```
{
  "name": "my_controlhealth",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "dotenv": "^16.3.1",
    "firebase": "^10.3.1",
    "formik": "^2.4.5",
    "i18next": "^23.5.1",
    "i18next-browser-languagedetector": "^7.1.0",
    "moment": "^2.29.4",
    "react": "^18.2.0",
    "react-big-calendar": "^1.8.4",
    "react-bootstrap": "^2.8.0",
    "react-dom": "^18.2.0",
    "react-i18next": "^13.2.2",
    "react-icons": "^4.10.1",
    "react-modal": "^3.16.1",
    "react-player": "^2.13.0",
    "react-redux": "^8.1.2",
    "react-router-dom": "^5.3.4",
    "react-scripts": "^5.0.1",
    "react-simple-chatbot": "^0.6.1",
    "react-slick": "^0.29.0",
    "react-toastify": "^9.1.3",
    "recharts": "^2.8.0",
    "redux": "^4.2.1",
    "redux-persist": "^6.0.0",
    "redux-thunk": "^2.4.2",
    "sweetalert": "^2.1.2",
    "web-vitals": "^2.1.4",
    "yup": "^1.3.2"
  },
}
```

Las dependencias listadas en el archivo son fundamentales en la aplicación "**Control Health**".

Firebase permite la gestión de la base de datos y la autenticación, mientras que **Redux** y **React Redux** son esenciales para el control del estado de la aplicación.

React Router facilita la navegación entre páginas, y

React Bootstrap proporciona componentes visuales coherentes. **Formik** y **Yup** se utilizan para la gestión de formularios, y **i18next** posibilita la

internacionalización. Además, otras dependencias como recharts, **react-toastify**, y **react-player** añaden funcionalidades específicas, mejorando la experiencia del usuario en la aplicación.

VIII. Referencias

A., D. (29 de 06 de 2023). *Qué es React: definición, características y funcionamiento*. Obtenido de <https://acortar.link/k2aZvP>

Arias, J. (08 de 04 de 2023). *Qué es Redux.js y como funciona?* Obtenido de <https://acortar.link/fJ8w8k>

Desarrollador Fantasma. (15 de 06 de 2023). *¿Qué es React-bootstrap? ¿Vale la pena?* Obtenido de <https://acortar.link/65UgMt>

edgarbc.wordpress. (Febrero de 2014). *Arquitectura Cliente/Servidor (Dos capas)*. Obtenido de <https://acortar.link/O75yZK>

Mora, S. L. (17 de 05 de 2020). *Firebase: qué es, para qué sirve, funcionalidades y ventajas*. Obtenido de <https://acortar.link/hSF8in>

Simões, C. (27 de 07 de 2021). *¿Qué es Node.js, y para qué sirve?* Obtenido de <https://acortar.link/5fEcG>