



UNIVERSIDAD
NACIONAL
AUTÓNOMA DE
NICARAGUA,
MANAGUA
UNAN-MANAGUA

Facultad Regional Multidisciplinaria de Chontales

Departamento de Ciencias, Tecnología y Salud

Proyecto de Graduación para optar al título de:

Ingeniero en Sistemas de Información

TÍTULO DEL PROYECTO:

Desarrollo de una aplicación web de gestión de citas médicas y asistencia previa a pacientes llamada "Medicall" en clínicas locales de la ciudad de Juigalpa-Chontales utilizando la biblioteca React, durante el I semestre del año 2022.

CARRERA:

Ingeniería en Sistemas de Información

AUTOR:

Br. Carlos Guthier Duarte Rocha

TUTORA:

Ing. Isamar Del Socorro Bonilla Martínez

Juigalpa-Chontales, noviembre de 2022

¡A la libertad por la Universidad!

Información General

Autor

Carlos Guthier Duarte Rocha

Correo: cgdr010601@gmail.com

Tutora

Isamar Del Socorro Bonilla Martínez

Correo: isa3006bonilla@gmail.com

Palabras claves:

Desarrollo web, desarrollo ágil, Front-End, Back-End, automedicación, consultas en línea, citas médicas, prescripción médica, automedicación, medicamentos.

Carta aval del docente

El uso de las tecnologías de información y comunicación TIC's han beneficiado al sector de la salud mejorando la calidad en la atención a los pacientes, el sector sanitario ha experimentado una gran transformación en cuanto a la digitalización, reduciendo costos administrativos y descentralizando la atención médica con la implementación de nuevos modelos sanitarios. La crisis de la pandemia de COVID-19 ha acelerado este proceso, la inserción de las TIC's es imprescindible en cualquier sistema de salud moderno.

En Nicaragua se utiliza un sistema de salud con asistencia primaria, que presenta algunos inconvenientes con respecto a tiempos prolongados de espera para recibir atención médica y hacinamiento de pacientes, con base en este contexto nace la oportunidad de desarrollar "MEDICALL", una aplicación web que permite realizar consultas médicas en línea para agilizar la preatención, que contribuye a reducir la automedicación en los pacientes y mejorar el acceso a los servicios médicos.

En este documento se evidencia la utilización de metodologías, tecnologías y estándares de calidad utilizados durante el proceso de desarrollo de software, que garantiza un producto con características aceptables, cumpliendo con la misión de la UNAN Managua que se corresponde con los objetivos del Plan Nacional De Lucha Contra La Pobreza Y Para El Desarrollo Humano. El cumplimiento de estos requisitos permite aceptar este proyecto como defensa de grado y demuestra el aprendizaje integral del alumno con dominio de las competencias científico-técnicas que aportan al desarrollo económico del país.

Por lo antes expuesto, se considera que el estudiante: Br. Carlos Guther Duarte Rocha, reúne 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.

Ing. Isamar Bonilla Martínez

Dedicatoria y agradecimiento

A Dios por darme la vida, la inteligencia y la oportunidad de continuar creciendo personal y profesionalmente. A mis padres, que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. A mi pareja, agradezco por tenerme tanta paciencia, estar a mi lado en todo momento y motivarme a cumplir todo lo que me proponga. Son muchos los docentes que han sido parte de mi camino universitario, y a todos ellos les quiero agradecer por transmitirme los conocimientos necesarios para culminar este logro en la carrera que me apasiona, sin ustedes los conceptos serían solo palabras.

Br. Carlos Guther Duarte Rocha

Resumen ejecutivo

La desinformación acerca de las enfermedades comunes es notable en la sociedad, el ciudadano desconoce cuál especialidad y medicamento atiende la enfermedad que padece, además, existen personas a las cuales se les complica trasladarse a recibir atención médica, ya sea por dificultades físicas que le impidan el desplazamiento o por evitar aglomeraciones que puedan finalizar con posibles contagios de enfermedades.

Para ello, se ideó una aplicación web llamada Medicall, que posee funcionalidades como: reserva de citas tanto en consultorio médico como a domicilio y que permite realizar consultas médicas en línea para agilizar la preatención. Esta aplicación se desarrolló utilizando la metodología SCRUM obteniendo un desarrollo ágil y colaborativo, controlando cada una de las etapas: desarrollo, diseño, estructuras, arquitectura, y los entregables de la aplicación.

Para llevar a cabo este desarrollo, se seleccionaron distintas tecnologías: React; para creación de interfaces de usuario interactivas por medio de componentes, Redux; para control y manejo del estado de la aplicación y Firebase; para la gestión de datos con el objetivo de realizar un producto de calidad, escalable y que brinde una posible solución a la problemática que se presenta en la población.

Es posible acceder básicamente desde cualquier lugar con conexión a internet mediante un navegador de preferencia y reservar cita con médicos de distintas especialidades, dándole tu enfoque personalizado al momento de elegir tus propios horarios, por ende, eliminando gastos de transporte o complicaciones en el tráfico además de ahorro de tiempo.

Índice

I.	Solución.....	1
II.	Diseño o modelo.....	2
2.1.	Diseño de funcionamiento	2
2.2.	Diseño de navegación	5
2.3.	Diseño arquitectónico	6
2.3.1.	Diagrama Nivel 0.....	6
2.3.2.	Diagrama Nivel 1.....	7
2.3.3.	Diagrama Nivel 2.....	8
III.	Justificación de la tecnología.....	9
1.1.	Selección del framework de diseño.....	11
1.2.	Selección del gestor de base de datos.....	13
IV.	Justificación de la metodología	17
V.	Oportunidad de mercado	20
VI.	Propuesta de valor	21
VII.	Ciclo del proyecto.....	22
7.1.	Historias de usuario	22
7.2.	Product Backlog	31
7.3.	Product Backlog organizado por Sprints.....	32
7.4.	Sprint Backlog.....	33
7.4.1.	Sprint 1.....	34
7.4.2.	Sprint 2.....	37
7.4.3.	Sprint 3.....	40
7.4.4.	Sprint 4.....	46
VIII.	Presupuesto.....	49
IX.	Marco lógico	51
9.1.	Árbol del problema	52
9.2.	Árbol de objetivos	53
9.3.	Árbol de la solución	53
9.4.	Matriz de marco lógico.....	55
X.	Cronograma de actividades.....	58
10.1.	Tabla.....	58

10.2.	Diagrama de Gantt	60
XI.	Referencias	61
XII.	Anexos	63
12.1.	Modelo CANVAS	63
12.2.	Base de datos	64
XIII.	Manual de usuario	1
1.	Resumen	1
2.	Paciente	1
2.1.	Crear cuenta	1
2.2.	Iniciar sesión	3
2.3.	Buscar médico	4
2.4.	Agendar cita médica	9
2.5.	Realizar consulta médica	12
2.6.	Ver listado de citas agendadas	14
2.7.	Ver listado de consultas solicitadas	14
2.8.	Cerrar sesión	15
3.	Médico	15
3.1.	Crear cuenta	15
3.2.	Iniciar sesión	16
3.3.	Ver listado de citas recibidas	17
3.4.	Ver listados de consultas recibidas	17
XIV.	Manual técnico	1
1.	Resumen	1
2.	Tecnologías utilizadas	2
3.	Requerimientos de hardware	5
3.1.	Computadora	6
3.2.	Dispositivo móvil	6
4.	Arquitectura Front-End	7
5.	Estructura del Front-End	8
XV.	Plan de Pruebas	1
1.	Características a ser probadas	1
1.1.	Características a ser probadas Sprint 1	1
1.2.	Características a ser probadas Sprint 2	1

1.3.	Características a ser probadas Sprint 3.....	2
1.4.	Características a ser probadas Sprint 4.....	2
2.	Características a no ser probadas.....	3
2.1.	Características a no ser probadas Sprint 1.....	3
2.2.	Características a no ser probadas Sprint 2.....	3
2.3.	Características a no ser probadas Sprint 3.....	4
3.	Informe de pruebas.....	5
3.1.	Pruebas Sprint 1.....	5
3.2.	Pruebas Sprint 2.....	7
3.3.	Pruebas Sprint 3.....	9
3.4.	Pruebas Sprint 4.....	11
3.5.	Casos de prueba Sprint 1.....	12
3.6.	Casos de prueba Sprint 2.....	14
3.7.	Casos de prueba Sprint 3.....	16
3.8.	Casos de prueba Sprint 4.....	18

Índice de ilustraciones

Ilustración 1. Diagrama de funcionamiento de paciente.....	3
Ilustración 2. Diagrama de funcionamiento de médico.....	4
Ilustración 3. Diagrama de navegación	5
Ilustración 4. Diagrama Nivel 0.....	6
Ilustración 5. Diagrama Nivel 1.....	7
Ilustración 6. Diagrama Nivel 2.....	9
Ilustración 7. H1 - Agendar cita médica.....	35
Ilustración 8. H3 - Gestión de citas	35
Ilustración 9. H4 - Administrar citas	36
Ilustración 10. H2 - Consulta médica	38
Ilustración 11. H15 - Gestión de consultas	38
Ilustración 12. H16 - Administrar consultas	39
Ilustración 13. H5 - Selección de médico (filtros).....	42
Ilustración 14. H5 - Selección de médico (especialidades)	42
Ilustración 15. H5 - Selección de médico (establecimientos)	43
Ilustración 16. H5 - Selección de médico (listado)	43
Ilustración 17. H5 - Selección de médico (perfil).....	44
Ilustración 18. H8 - Registro de paciente.....	44
Ilustración 19. H13 - Inicio de sesión	48
Ilustración 20. Árbol del problema	52
Ilustración 21. Árbol de objetivos.....	53
Ilustración 22. Árbol de la solución	54
Ilustración 23. Modelo CANVAS	63
Ilustración 24. Base de datos en Cloud Firestore	64

Índice de tablas

Tabla 1. Comparación de tecnologías de desarrollo web	10
Tabla 2. Comparación de frameworks de diseño	12
Tabla 3. Comparación de gestores de base de datos	14
Tabla 4. Comparación de metodologías de desarrollo de software ágiles	18
Tabla 5. Historia de usuario 1	23
Tabla 6. Historia de usuario 2	23
Tabla 7. Historia de usuario 3	24
Tabla 8. Historia de usuario 4	24
Tabla 9. Historia de usuario 5	25
Tabla 10. Historia de usuario 6	25
Tabla 11. Historia de usuario 7	26
Tabla 12. Historia de usuario 8	26
Tabla 13. Historia de usuario 9	27
Tabla 14. Historia de usuario 10	27
Tabla 15. Historia de usuario 11	28
Tabla 16. Historia de usuario 12	28
Tabla 17. Historia de usuario 13	29
Tabla 18. Historia de usuario 14	29
Tabla 19. Historia de usuario 15	30
Tabla 20. Historia de usuario 16	30
Tabla 21. Product Backlog	32
Tabla 22. Product Backlog organizado por Sprints	33
Tabla 23. Sprint 1	34
Tabla 24. Sprint 2	37
Tabla 25. Sprint 3	41
Tabla 26. Sprint 4	47
Tabla 27. Presupuesto	50
Tabla 28. Matriz de marco lógico	57

I. Solución

La automedicación es la utilización de medicamentos por iniciativa propia sin ninguna intervención por parte del médico (ni en el diagnóstico de la enfermedad, ni en la prescripción o supervisión del tratamiento). La automedicación es un hábito común en nuestra sociedad y no está exento de riesgos: se utilizan medicamentos para el dolor de cabeza, para problemas gástricos, para relajarse, para la tos, para la alergia, etc. La automedicación responsable puede ser conveniente si se utiliza para tratar síntomas menores como el dolor, la fiebre, la acidez de estómago, el resfriado, etc. y durante un tiempo limitado. De hecho, existen medicamentos que no necesitan receta médica, denominados Especialidades Farmacéuticas Publicitarias o EFP. Aun así, que un medicamento sea una EFP y que se dispense sin receta no quiere decir que sea inocuo y no pueda resultar perjudicial en determinadas situaciones ya que no deja de ser un medicamento.

Recientemente, la falta de información y las pocas medidas para responder oportunamente a la pandemia de COVID-19 en Nicaragua provocaron que la población aumentara de manera masiva la automedicación para contrarrestar los síntomas de esta enfermedad, poniendo en grave riesgo su salud y provocando mayores complicaciones.

Para evitar la automedicación, se ideó “Medicall”, una aplicación web que estará disponible 24/7 en atención a pacientes.

Reúne un sinnúmero de médicos clasificados, brinda información acerca de las especialidades médicas, permite reservar citas tanto en consultorio médico como a domicilio y realizar consultas médicas en línea para agilizar la preatención. El paciente elige quién lo atenderá, a través de filtros de especialidades y establecimientos médicos basándose en el perfil de los profesionales y en las puntuaciones que hayan recibido sobre los servicios brindados.

Se puede acceder básicamente desde cualquier lugar con conexión a internet y reservar cita con médicos de distintas especialidades, dándole un enfoque personalizado al momento de elegir tus propios horarios, por ende, eliminando gastos de transporte o complicaciones en el tráfico además del ahorro de tiempo.

II. Diseño o modelo

Se muestra una representación gráfica del funcionamiento que tendrá la aplicación web que se ha desarrollado. Medicall maneja dos tipos de usuarios donde cada uno dispone de una serie de opciones que representarán las funcionalidades principales de la aplicación. Ambos tipos de usuario, médico y paciente, pueden realizar actividades del sistema posterior al inicio de sesión, los pacientes podrán realizar consultas médicas en línea y agendar citas médicas tanto en consultorio como a domicilio, por su parte, los médicos podrán gestionar las citas y consultas recibidas.

2.1. Diseño de funcionamiento

Las siguientes imágenes definen de manera gráfica el funcionamiento de la aplicación, detallando las opciones disponibles para los dos tipos de usuarios: médico y paciente. En cuanto al usuario paciente se muestra el proceso que le permitirá conocer los distintos médicos que brindan servicios en la aplicación, agendar una cita en consultorio o a domicilio y solicitar una consulta médica en línea. De la misma forma, se definen las funciones del usuario médico que serán gestionar citas y consultas médicas recibidas por parte del usuario paciente.

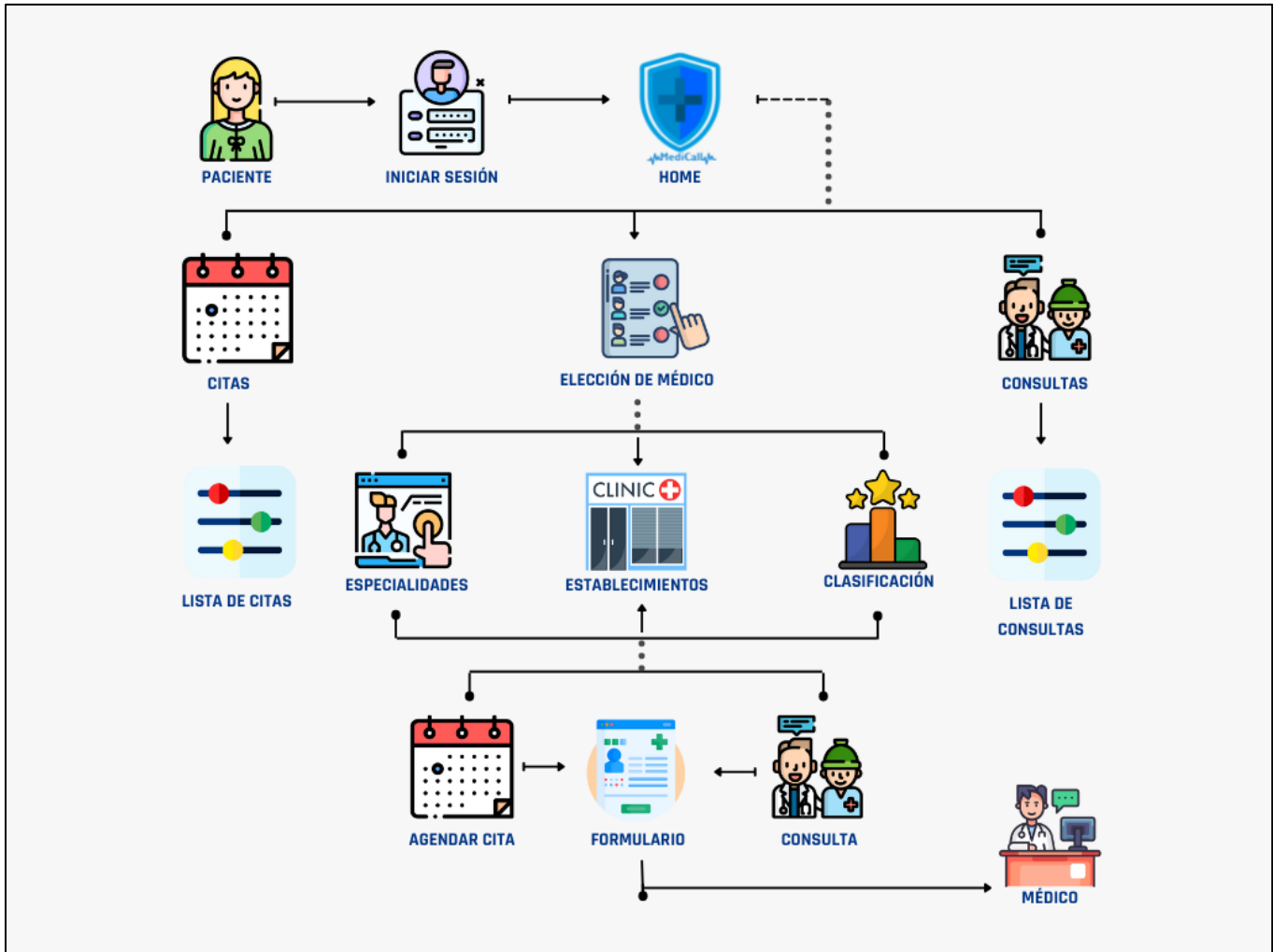


Ilustración 1. Diagrama de funcionamiento de paciente

Fuente: Elaboración propia

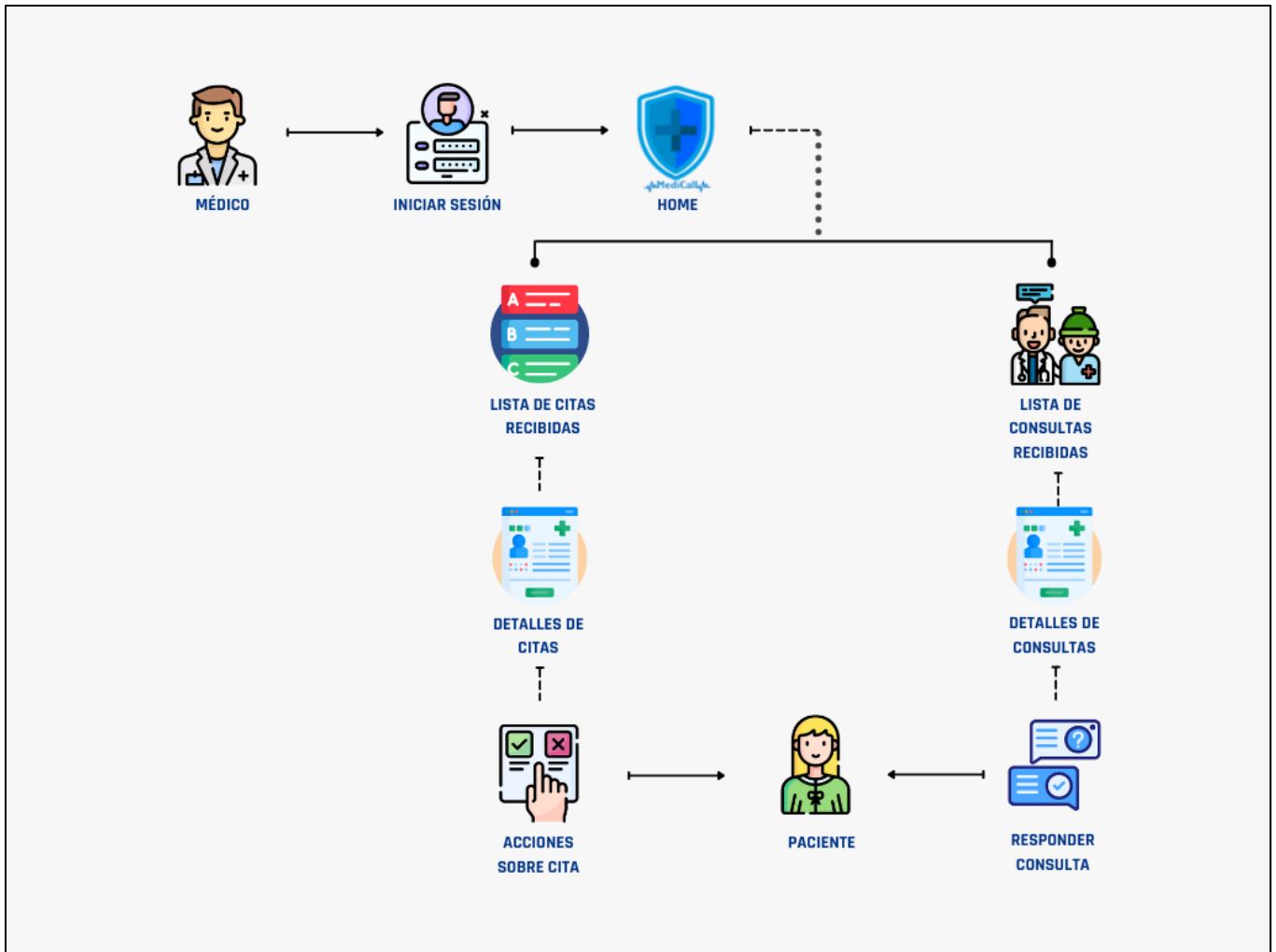


Ilustración 2. Diagrama de funcionamiento de médico

Fuente: Elaboración propia

2.2. Diseño de navegación

El mapa de navegación es un esquema, que se puede pensar como un árbol jerárquico, que representa la arquitectura de las páginas de un sitio web. Esta representación puede ser, como un mapa conceptual, de forma gráfica para visualizar y entender sobre un mapa el recorrido de visitante, es decir su navegación y cómo este llega a lo que está buscando. También, en este se organiza la distribución y la jerarquía del contenido.

Así, entre las ventajas de contar con un mapa de navegación está la diferenciación en un sitio amigable y que entiende al usuario, un fácil acceso al contenido, y con ello, la generación de más conversiones y de satisfacción con el servicio que se provee en un sitio web, como lo puede ser en algunos casos el informativo. (Fetecua, 2022)

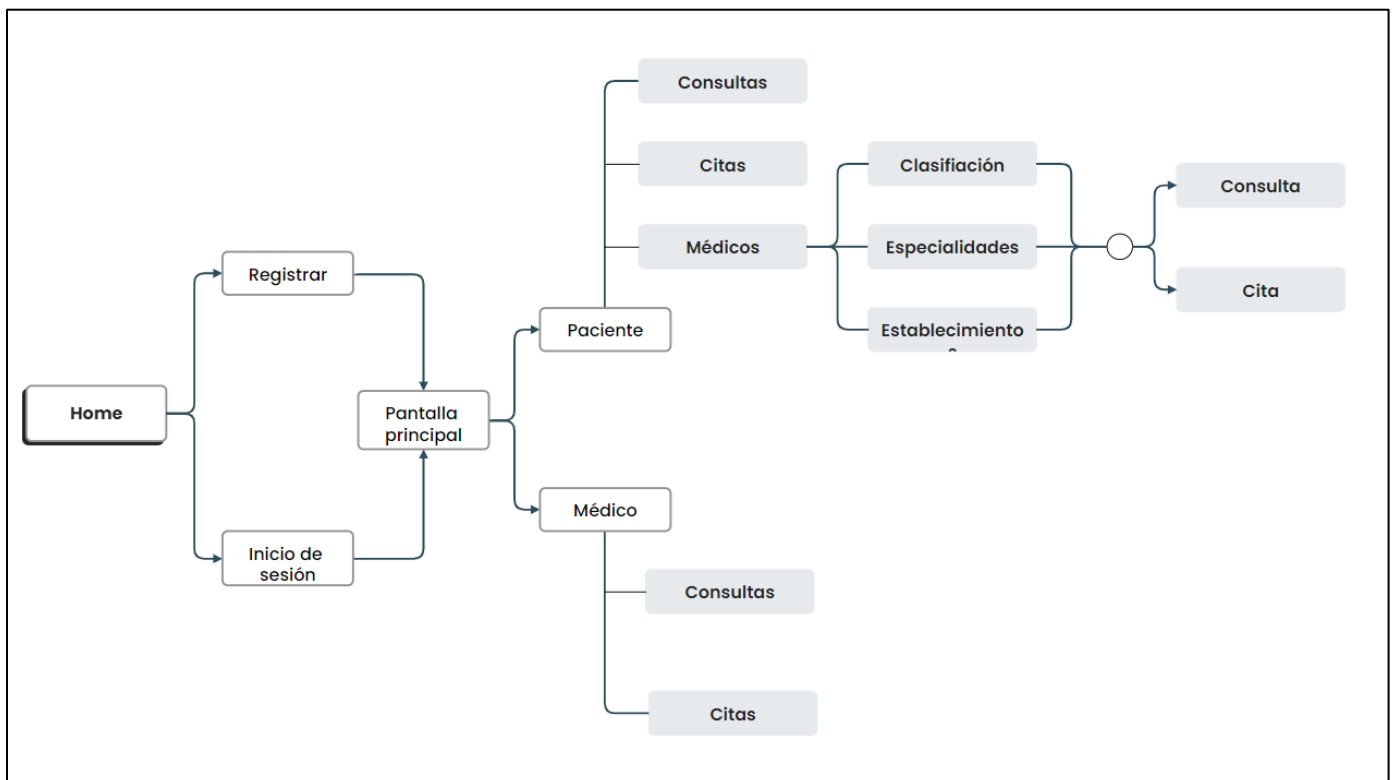


Ilustración 3. Diagrama de navegación

Fuente: Elaboración propia

2.3. Diseño arquitectónico

A continuación, se define la estructura y las responsabilidades de los componentes que conforman la arquitectura del proyecto de software.

Flux es la arquitectura manejada por la biblioteca de JavaScript llamada React y es bajo la cual se desarrolló la aplicación. Esta es similar a la arquitectura Modelo-Vista-Controlador porque también contiene su modelo, vista y controladores, pero está pensada en un flujo de datos unidireccional. Los datos viajan desde la vista por medio de acciones y llegan a un store desde el cual se actualizará la vista nuevamente.

2.3.1. Diagrama Nivel 0

Este diagrama representa los componentes principales en los que está dividida la aplicación, los cuales son: el servidor que es proveído por Firebase, el store que contiene el estado global de la aplicación y la parte del Front-End donde se definen todos los tipos de usuarios disponibles para la interacción del usuario como paciente y médico.

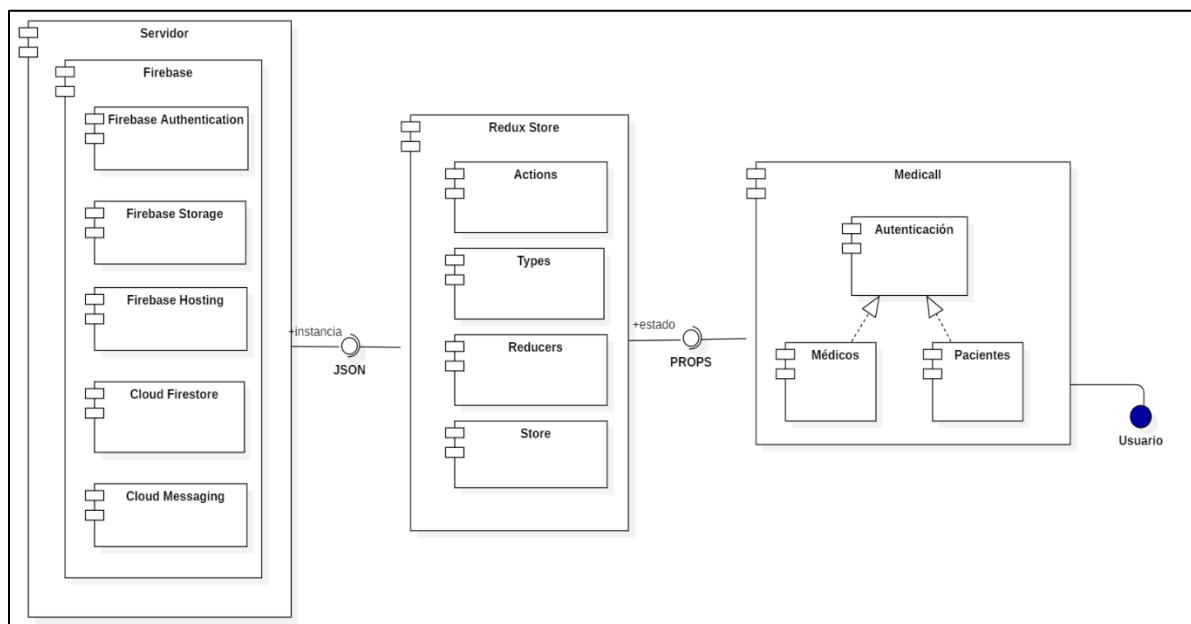


Ilustración 4. Diagrama Nivel 0

Fuente: Elaboración propia

2.3.2. Diagrama Nivel 1

El siguiente diagrama refleja los componentes y las relaciones de cada módulo de la aplicación, definiendo los apartados de autenticación, paciente, médico, citas y consultas.

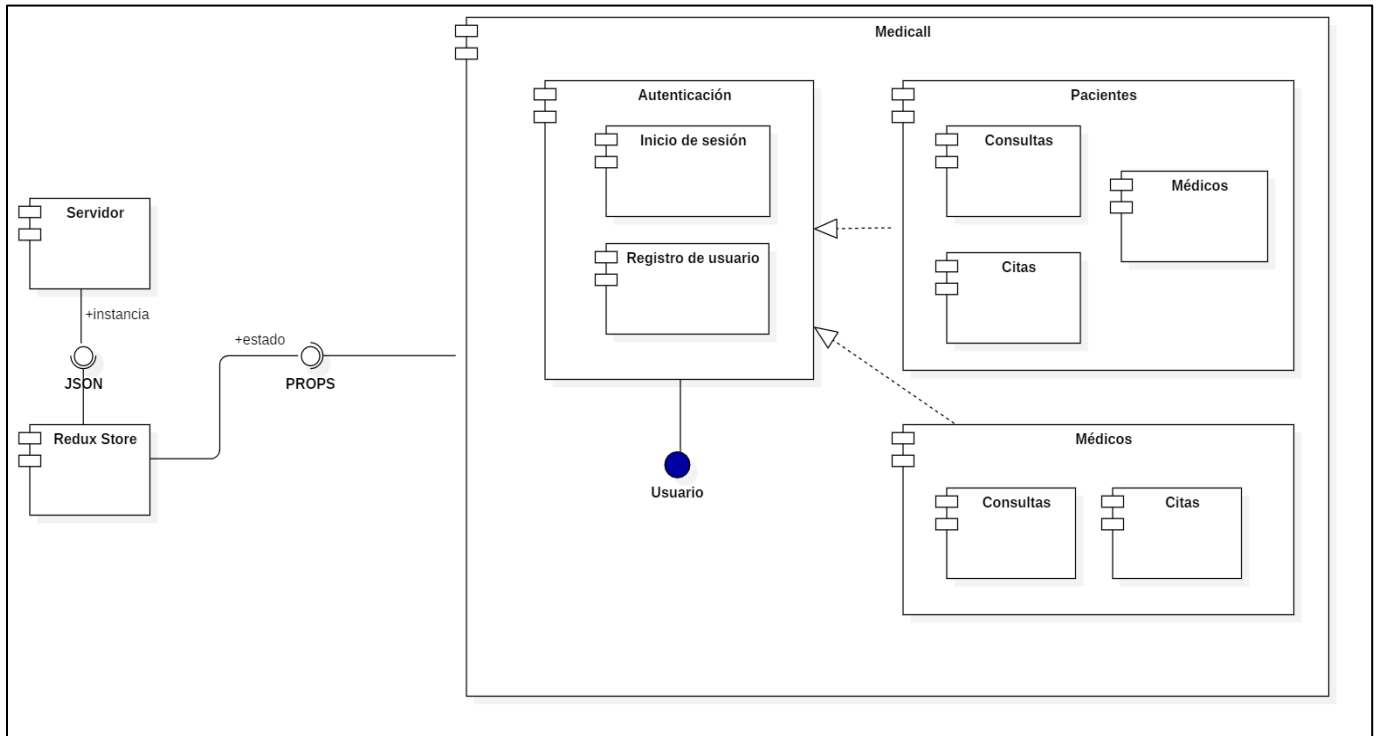


Ilustración 5. Diagrama Nivel 1

Fuente: Elaboración propia

2.3.3. Diagrama Nivel 2

Flux es la arquitectura de aplicaciones que utiliza Facebook para crear aplicaciones web del lado del cliente. Complementa los componentes de vista componibles de React utilizando un flujo de datos unidireccional. (Tay, 2022)

Flux nació desde Facebook por un problema que se les presentaba al tener una comunicación bidireccional entre los modelos y los controladores, haciéndoles muy difícil poder depurar y rastrear errores, propone una arquitectura en la que el flujo de datos es unidireccional. Los datos viajan desde la vista por medio de acciones y llegan a un store desde el cual se actualizará la vista de nuevo, teniendo un único camino, y un sitio donde se almacena el estado de la aplicación, es más sencillo depurar errores y saber que está pasando en cada momento. (Azaustre, 2018)

A continuación, se representa la interacción de la aplicación y su store, según la arquitectura Flux. Se detalla el flujo de datos que es unidireccional y cómo se implementan los llamados reducers, que se encargan de procesar las acciones y generar un nuevo estado en el store.

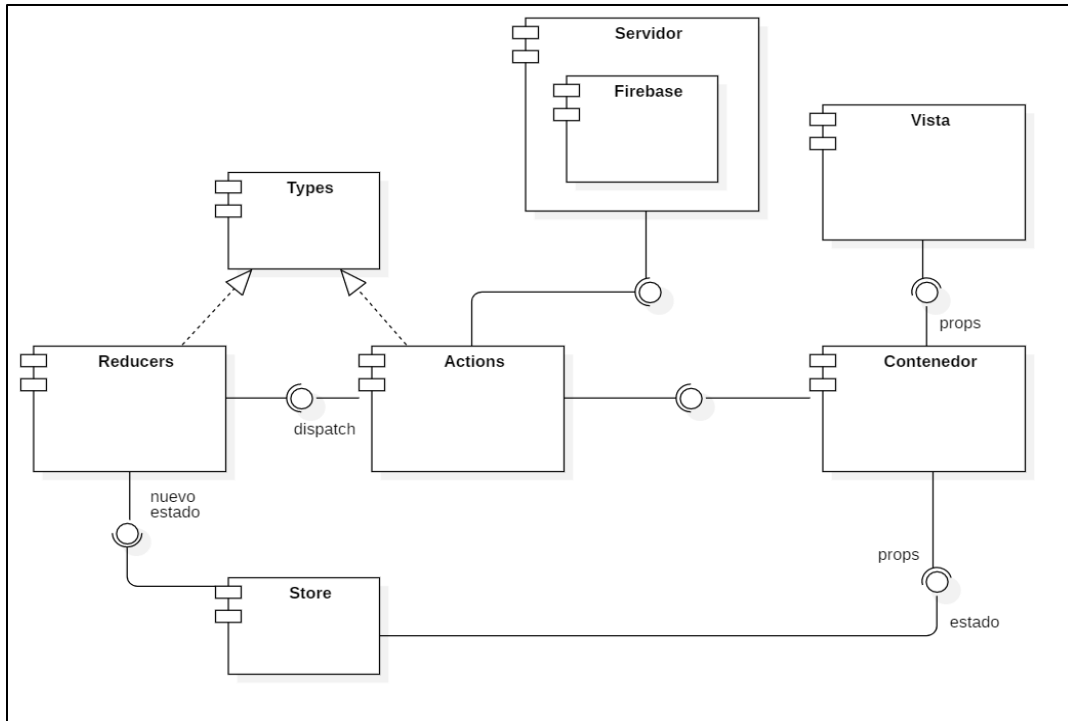


Ilustración 6. Diagrama Nivel 2

Fuente: Elaboración propia

III. Justificación de la tecnología

A continuación, se presenta un cuadro comparativo correspondiente a las tecnologías más populares en el desarrollo web, tomando en cuenta la experiencia, curva de aprendizaje y las ventajas como parámetros principales para elegir la opción más adecuada.

Tecnologías para el desarrollo web						
Nombre	Licencia	Lenguaje	Experiencia	Curva de Aprendizaje	Descripción	Ventajas
Angular	MIT License	TypeScript	50%	50%	Angular es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de una sola página (Angular, s.f.)	<ul style="list-style-type: none"> • Un marco basado en componentes. • Colección de bibliotecas bien integradas • conjunto de herramientas de desarrollo. (Angular, s.f.)
JQuery	MIT License	JavaScript	80%	20%	jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. (jQuery, s.f.)	<ul style="list-style-type: none"> • flexible y rápido para el desarrollo web • Plugins • Excelente integración con AJAX (Programación - Diseño - Web, 2013)
React	MIT License	JavaScript	80%	20%	React es una biblioteca de JavaScript para crear interfaces de usuario. (React, s.f.)	<ul style="list-style-type: none"> • Declarativo • Basado en componentes (React, s.f.)

Tabla 1. Comparación de tecnologías de desarrollo web

Fuente: Elaboración propia

Posterior a la evaluación de las tecnologías, se ha seleccionado React para el desarrollo web debido a que el equipo adquirió experiencia con el desarrollo de proyectos anteriores, haciendo que la curva de aprendizaje sea relativamente corta y el proceso de desarrollo sea más eficiente.

1.1. Selección del framework de diseño

A continuación, se presenta un cuadro comparativo correspondiente a las tecnologías más populares en el diseño web, tomando en cuenta las ventajas, desventajas y la compatibilidad con la tecnología seleccionada para el desarrollo como parámetros principales para elegir la opción más adecuada.

Framework	Descripción	Ventajas	Desventajas
MUI	MUI es un conjunto integral de herramientas para construir interfaces de usuario para React. (MUI, s.f.)	<ul style="list-style-type: none"> • Animaciones incorporadas • Experiencia unificada para cualquier dispositivo 	<ul style="list-style-type: none"> • Intrínsecamente ligado a Google • Directrices marcadas
Bootstrap	Framework front-end para desarrollar aplicaciones web con layout adaptativo. (Bootstrap, s.f.)	<ul style="list-style-type: none"> • Mobile first • Biblioteca de componentes (Rock Content, 2020) 	<ul style="list-style-type: none"> • Optimización • Código HTML Complejo • Similitud de diseños. (Zúñiga, 2015)
Tailwind CSS	Marco de CSS altamente personalizable basado en clases aplicables al HTML. (Rodríguez, 2021)	<ul style="list-style-type: none"> • Personalización • Patrones de utilidad • Interacción fluida 	<ul style="list-style-type: none"> • Mercado de hipertexto desorganizado • Tiempos de desarrollo. (Abba, 2022)

Tabla 2. Comparación de frameworks de diseño

Fuente: Elaboración propia

Finalmente, después de comparar los frameworks de diseño se ha decidido utilizar Material UI, conocido como MUI, ya que brinda mayor compatibilidad y ventajas al ser utilizado con la tecnología de React.

1.2. Selección del gestor de base de datos

Para la selección del gestor de base de datos se ha creado un cuadro comparativo para evaluar los aspectos de cada uno de ellos, tomando en cuenta la curva de aprendizaje, el tipo de dato con que trabaja, el tipo de alojamiento y la disponibilidad de forma gratuita.

Base de datos	Tipo de datos	Licencia	Soporte	Curva de aprendizaje	Alojamiento
MySQL	Relacional	Licencia pública general/Licencia comercial	Apps móviles, web, Windows, Linux, MacOS	40%	Local y en la nube
MongoDB	Orientada a documentos	Server-Side Public License (SSPL)	Android, iOS, web	50%	Local y en la nube
PostgreSQL	Combina relacional y orientado a objetos	Licencia PostgreSQL	Android, iOS, web, Windows, Linux, MacOS	90%	Local y en la nube
SQLite	Relacional	Dominio público	Apps móviles y web	30%	Local y en la nube
Cloud Firestore	Orientada a documentos	Creative Commons Attribution 4.0 License	Android, iOS, web	10%	En la nube

Tabla 3. Comparación de gestores de base de datos

Fuente: Elaboración propia

El gestor de base de datos elegido fue Cloud Firestore por su alojamiento gratuito en la nube y debido a que el equipo adquirió experiencia con el desarrollo de proyectos anteriores, haciendo que la curva de aprendizaje sea relativamente corta.

Cabe señalar que Cloud firestore está integrado dentro de la plataforma de Backend as a Service (BaaS) desarrollada por Google llamada Firebase.

Según (Edwards, 2020) Firebase comenzó como una arquitectura de mensajería móvil que fue adquirida por Google. Desde entonces, ha evolucionado hasta convertirse en un conjunto de más de 25 componentes que interoperan con GoogleCloud Platform.

Firebase consta de Kits de Desarrollo de Software (SDK), que permiten a los desarrolladores web y móviles acceder a la funcionalidad de la nube de forma sencilla, segura y fiable. Compensan automáticamente la mala conectividad de la red. Hay una consola web de Firebase para habilitar, administrar y proteger componentes. También hay herramientas de línea de comandos y API REST para un uso más profundo.

Firebase ha evolucionado para convertirse en una plataforma que permite a los desarrolladores de aplicaciones móviles y web desarrollar aplicaciones completas sin la necesidad de servidores de fondo. Las mejoras recientes han facilitado enormemente las soluciones sin servidor que brindan una alternativa viable, escalable y rentable a las soluciones de servidor de máquinas virtuales en la nube.

Las funciones que ofrecen son:

- Autenticación

Firebase Authentication se encarga de que sus usuarios inicien sesión y se identifiquen. Este producto es esencial para configurar correctamente algunas de las otras funcionalidades, especialmente si necesita restringir el acceso a los datos por usuario, lo que en casi todas las aplicaciones se debe hacer.

La autenticación de Firebase es quizás el componente de Firebase más utilizado. Los usuarios pueden seleccionar uno o más de varios mecanismos de autenticación. Estos son la

dirección de correo electrónico y la contraseña, los números de teléfono y los proveedores de identidad federados Google, Facebook, Twitter y GitHub.

- Cloud Firestore

Se considera el reemplazo de Realtime Database. Amplía la funcionalidad de Realtime Database. Los datos no están en un árbol JavaScript Object Notation (JSON) sino en una colección jerárquica de documentos. Cada documento consta de un conjunto de pares clave-valor y subdocumentos opcionales. Las consultas permiten un filtrado y una clasificación más complejos y solo devuelven documentos completos. Las consultas no devuelven subdocumentos.

Cloud Firestore pronto reemplazará a Cloud Datastore. Se puede ejecutar en modo Datastore o en modo nativo. Todas las aplicaciones de Datastore se migrarán automáticamente a Cloud Firestore.

- Cloud Storage

Firebase cuenta con un sistema de almacenamiento, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones y sincronizarlos. Al igual que la mayoría de herramientas de Firebase, es personalizable mediante determinadas reglas.

Este almacenamiento es de gran ayuda para tratar archivos de los usuarios (por ejemplo, fotografías que hayan subido), que se pueden servir de forma más rápida y fácil. También hace la descarga de referencias a ficheros más segura.

- Cloud functions

Las cloud functions son una tecnología importante para crear aplicaciones sin servidor. Una cloud function se puede escribir en JavaScript, TypeScript, Python o Go, que se implementa directamente en la nube de Google. Una función se activa mediante una solicitud HTTP o un evento en la nube, como escribir en Cloud Storage.

Una función en la nube solo puede manejar una solicitud a la vez, pero la nube escala automáticamente la función al replicarla. Una cloud function escrita en Python usa la biblioteca Flask para manejar solicitudes HTTP. La función toma un objeto de solicitud como parámetro y devuelve el cuerpo de la respuesta.

- Firebase Hosting

El alojamiento de Firebase permite que el contenido web estático, que incluye JavaScript, se aloje en la nube sin la necesidad de un servidor web. El contenido se almacena en caché en los bordes de una red de entrega de contenido (CDN) global. Esto permite un acceso rápido al contenido desde cualquier parte del mundo.

IV. Justificación de la metodología

Para la selección de la metodología se realizó un cuadro comparativo correspondiente a las metodologías de desarrollo de software ágiles con el objetivo de analizar las principales características, proceso, roles y herramientas para elegir la que mejor se adapte al tipo de proyecto.

Metodología	Características	Proceso de desarrollo	Herramientas
SCRUM	<p>marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos.</p> <p>Muestra la eficacia relativa de la gestión de producto y las prácticas de desarrollo.</p> <p>transparencia, inspección y adaptación. (Guerrero, 2018)</p>	<p>Se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas).</p> <p>Cada iteración tiene que proporcionar un incremento de producto final.</p>	<p>Product Backlog lista ordenada de necesidades en el producto.</p> <p>Sprint Backlog conjunto de elementos de la lista de producto seleccionados para el Sprint.</p> <p>Burndown chart gráfica de trabajo pendiente</p>
KANBAN	<p>Proporciona un flujo de trabajo para dividir el proceso en fases.</p> <p>Los 4 principios básicos son:</p> <ul style="list-style-type: none"> • Acepta el cambio. • Respeta el proceso actual, roles y responsabilidades. • Liderazgo en todos los niveles. (K. Schwaber, 2013) 	<p>Kanban permite que el software se desarrolle en un solo ciclo de desarrollo. A pesar de esto, Kanban es un ejemplo de una metodología ágil porque, aunque no es iterativo, es incremental.</p>	<p>Tablero Kanban: Se usa para organizar las tareas del sprint en curso, se puede adaptar a las necesidades, se van moviendo las tarjetas por las diferentes columnas para tener una visión global del estado actual del sprint.</p>

Tabla 4. Comparación de metodologías de desarrollo de software ágiles

Fuente: Elaboración propia

Se seleccionó la metodología SCRUM que es un marco de desarrollo ágil de software, ya que este proceso permite aplicar un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible.

Según (Redacción APD, 2022) esta metodología cuenta con las siguientes fases que hacen posible que se lleve a cabo con éxito:

- Planificación: **Product Backlog**

El Product Backlog es la fase en la que se establecen las tareas prioritarias y donde se obtiene información breve y detallada sobre el proyecto que se va a desarrollar.

Con el método SCRUM no es necesario definir todos los objetivos al comienzo del proyecto. El Product Owner, de forma conjunta con el equipo de trabajo comienzan a listar lo más importante para el Product Backlog.

El Product Backlog es necesario para poder arrancar con el primer sprint, tiene permitido cambiar y crecer tantas veces como sea necesario en función del aprendizaje adquirido en el desarrollo del producto.

- Ejecución: **Sprint**

Dentro del método SCRUM, el Sprint es el corazón, un intervalo de tiempo que como máximo tiene una duración de un mes y en donde se produce el desarrollo de un producto que es entregable potencialmente.

Para entenderlo mejor, si el Product Owner solicita el producto se requiere un mínimo esfuerzo para su entrega al cliente.

También se puede definir el Sprint como un mini proyecto en donde el equipo de trabajo se focaliza en el desarrollo de tareas para alcanzar el objetivo que se ha definido previamente en el Sprint planning.

- Control: **Burn Down**

El Burn Down es la fase en la que se mide el progreso de un determinado proyecto SCRUM. En ella, el Scrum Master será el encargado de actualizar los gráficos cuando se finalice cada uno de los Sprint.

A modo de resumen, la metodología SCRUM se centra en el trabajo en equipo, siguiendo una serie de criterios establecidos para obtener los mejores resultados en un proyecto.

Debido al corto tiempo con el que se cuenta para el desarrollo de la app, SCRUM es la metodología ideal para el equipo.

v. Oportunidad de mercado

La aplicación web para la gestión de citas médicas llamada Medicall, pretende incursionar en un mercado inexistente en su categoría, es decir, no tiene competencias. El objetivo de mercado lo componen los distintos clientes (pacientes) que buscan atención médica en múltiples especialidades (odontología, pediatría, dermatología, entre otras) y los más de 30 mil médicos existentes en el país, según la Asociación Médica Nicaragüense (AMN).

Actualmente, Nicaragua atraviesa un proceso de avance tecnológico, pero en el país aún no existe una aplicación que permita acordar citas médicas y/o realizar consultas médicas en línea, déficit que aprovecha Medicall, conteniendo estas y más funcionalidades que brindan personalización al momento de recibir atención médica, brindando un servicio a bajo costo y que conforme avanza el desarrollo tecnológico del país se vuelve una necesidad, combinación que ascienden las posibilidades de éxito.

VI. Propuesta de valor

Una propuesta de valor es un recurso para transmitir de manera objetiva y directa las ventajas que una empresa puede aportar a sus clientes. (Higuerey, 2019)

Es más que una simple descripción del producto o servicio: es la solución específica que tu empresa brinda a un cliente, y que un competidor no puede ofrecer. (HubSpot, 2022)

- Educación médica

La aplicación cuenta con distintas especialidades médicas (anestesiología, cardiología, dermatología, endocrinología, ginecología, odontología, oftalmología, ortopedia, pediatría, psiquiatría, radiología, urología) para las cuáles habrá un apartado informativo, con el cuál, los pacientes pueden informarse acerca de estas.

- Pre atención médica en línea

Capacidad de realizar consultas médicas en línea, en los horarios establecidos previamente por cada médico, evitando el traslado a establecimientos médicos.

- Registro de Citas

Posibilidad de acordar una cita en línea evitando el traslado al establecimiento y, por ende, evitando incidentes en el tráfico, exposición a posibles contagios de enfermedades, robos y ahorrando tiempo además de gastos de transporte.

- Personalización

El paciente podrá dar su enfoque personalizado al elegir sus propios horarios y qué médico lo atenderá, por medio de filtros de búsqueda (establecimiento y especialidad).

- Referencias

El paciente podrá visualizar, en el perfil de cada médico, comentarios y puntuaciones otorgadas por pacientes, esto, con la finalidad de tener referencias acerca de la calidad del servicio que ofrece dicho médico.

- Variedad

La aplicación tendrá integrada un sinnúmero de médicos profesionales dispuestos a suplir las necesidades de la sociedad, estos contarán con un perfil en el cual brindarán información tanto personal como profesional, donde los pacientes puedan calificar al médico basándose en servicios médicos recibidos.

VII. Ciclo del proyecto

El ciclo del proyecto, entendido como una secuencia de actividades relacionadas entre sí destinadas a lograr un objetivo, en un tiempo determinado, y contando con un conjunto definido de recursos, desde su inicio, tienen un proceso determinado.

Al trabajar con la metodología SCRUM, se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de retroalimentación de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

7.1. Historias de usuario

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente. (Rehkopf, 2019)

H1. AGENDAR CITA MÉDICA			
Como	Paciente		
Quiero	Solicitar cita con médico por especialidades		
Para	Recibir atención médica presencial a domicilio o en consultorio		
Validación	<ul style="list-style-type: none"> Llenar formulario de citas 	Impacto (1-10)	10
		Prioridad (1-N)	1
	<ul style="list-style-type: none"> Guardar cita en BD La nueva cita debe aparecer en el apartado “Mis citas” 		
		Estimación (h)	10

Tabla 5. Historia de usuario 1

Fuente: Elaboración propia

G

H2. CONSULTA MÉDICA			
Como	Médico		
Quiero	Realizar consulta médica en línea		
Para	Evitar desplazarse al consultorio.		
Validación	<ul style="list-style-type: none"> Llenar formulario de consulta incluido historial clínico. 	Impacto (1-10)	10
		Prioridad (1-N)	2
	<ul style="list-style-type: none"> Guardar consulta en BD y notificar al médico. 	Estimación (h)	10

Tabla 6. Historia de usuario 2

Fuente: Elaboración propia

H3. GESTIÓN DE CITAS			
Como	Médico		
Quiero	Ver el listado de mis citas agendadas		
Para	Gestionar y responder a cada cita recibida.		
Validación	<ul style="list-style-type: none"> Visualizar listado de citas recibidas. 	Impacto (1-10)	9
		Prioridad (1-N)	3
	<ul style="list-style-type: none"> Acciones para cada cita. 	Estimación (h)	6

Tabla 7. Historia de usuario 3

Fuente: Elaboración propia

H4. ADMINISTRAR CITAS			
Como	Paciente		
Quiero	Visualizar listado de mis citas		
Para	Gestionar el control detallado de cada cita.		
Validación	<ul style="list-style-type: none"> Visualizar listado de mis citas programadas 	Impacto (1-10)	8
		Prioridad (1-N)	4
	<ul style="list-style-type: none"> Acciones para cada cita 	Estimación (h)	4

Tabla 8. Historia de usuario 4

Fuente: Elaboración propia

H5. SELECCIÓN DE MÉDICO			
Como	Paciente		
Quiero	Seleccionar médico por especialidad		
Para	Revisar la información contenida en su perfil.		
Validación	<ul style="list-style-type: none"> Visualizar médicos disponibles mediante los distintos filtros (búsqueda directa, especialidades, establecimientos). Mostrar información del médico cuando este sea seleccionado. 	Impacto (1-10)	9
		Prioridad (1-N)	5
		Estimación (h)	23

Tabla 9. Historia de usuario 5

Fuente: Elaboración propia

H6. CANCELAR CITA MÉDICA			
Como	Paciente		
Quiero	Cancelar una cita programada		
Para	Eliminar la cita para que no sea visualizada por el médico, debido a que por algún incidente no la realizaré		
Validación	<ul style="list-style-type: none"> En el apartado de “Mis Citas”, cada cita contará con la opción de cancelar cita. Notificar al médico de la cancelación de la cita. 	Impacto (1-10)	8
		Prioridad (1-N)	6
		Estimación (h)	3

Tabla 10. Historia de usuario 6

Fuente: Elaboración propia

H7. REPORTE			
Como	Médico		
Quiero	Reportar a un paciente que no acudió a una cita programada		
Para	Aumentar el compromiso con la plataforma.		
Validación	<ul style="list-style-type: none"> Reportar paciente 	Impacto (1-10)	8
		Prioridad (1-N)	7
	<ul style="list-style-type: none"> Aumentar contador de reportes a paciente 	Estimación (h)	3

Tabla 11. Historia de usuario 7

Fuente: Elaboración propia

H8. REGISTRO DE PACIENTE			
Como	Paciente		
Quiero	Crear mi cuenta en la aplicación		
Para	Poder hacer uso de la aplicación		
Validación	<ul style="list-style-type: none"> Guardar información del usuario en la BD 	Impacto (1-10)	7
		Prioridad (1-N)	8
	<ul style="list-style-type: none"> Iniciar sesión con las credenciales proporcionadas. 	Estimación (h)	4

Tabla 12. Historia de usuario 8

Fuente: Elaboración propia

H9. REGISTRO DE MÉDICO			
Como	Médico		
Quiero	Crear mi cuenta en la aplicación		
Para	Poder hacer uso de la aplicación		
Validación	<ul style="list-style-type: none"> • Guardar información del usuario en la BD • Iniciar sesión con las credenciales proporcionadas. 	Impacto (1-10)	7
		Prioridad (1-N)	9
		Estimación (h)	4

Tabla 13. Historia de usuario 9

Fuente: Elaboración propia

H10. NOTIFICACIÓN RECORDATORIO			
Como	Paciente		
Quiero	Recibir una notificación de recordatorio para cita agendada		
Para	Evitar posibles olvidos de citas programadas		
Validación	<ul style="list-style-type: none"> • Recibir notificación de cita programada (15 min anticipación) que se aproxima. 	Impacto (1-10)	8
		Prioridad (1-N)	10
		Estimación (h)	11

Tabla 14. Historia de usuario 10

Fuente: Elaboración propia

H11. COMENTARIOS Y PUNTUACIONES			
Como	Paciente		
Quiero	Brindar mi opinión acerca del servicio médico recibido mediante comentarios y puntuaciones		
Para	Que otros pacientes tengan referencia de la calidad del servicio que brinda dicho doctor.		
Validación	<ul style="list-style-type: none"> Redactar comentario y enviar a BD 	Impacto (1-10)	9
		Prioridad (1-N)	11
	<ul style="list-style-type: none"> Otros pacientes pueden ver mis comentarios 	Estimación (h)	12

Tabla 15. Historia de usuario 11

Fuente: Elaboración propia

H12. CITAS FUERA DE LA APP			
Como	Médico		
Quiero	Agregar citas que no son programadas a través de la aplicación.		
Para	Mantener actualizado el horario disponible para citas dentro de la aplicación.		
Validación	<ul style="list-style-type: none"> Agregar nueva cita 	Impacto (1-10)	9
		Prioridad (1-N)	12
	<ul style="list-style-type: none"> Actualizar horario de citas disponible 	Estimación (h)	4

Tabla 16. Historia de usuario 12

Fuente: Elaboración propia

H13. INICIO DE SESIÓN			
Como	Paciente o Médico		
Quiero	Iniciar sesión		
Para	Hacer uso de la aplicación y poder realizar citas y consultas en línea		
Validación	<ul style="list-style-type: none"> Validar información ingresada en los campos (correo y contraseña). Detectar el rol de usuario que intenta acceder. Ingresar a la aplicación según rol detectado. 	Impacto (1-10)	8
		Prioridad (1-N)	13
		Estimación (h)	5

Tabla 17. Historia de usuario 13

Fuente: Elaboración propia

H14. NOTIFICACIÓN DE NUEVA CITA			
Como	Médico		
Quiero	Recibir notificación de nueva cita o de que se acerca la fecha de una cita programada		
Para	Brindar una respuesta rápida al paciente (aceptar o rechazar la cita solicitada).		
Validación	<ul style="list-style-type: none"> Recibir notificación cuando un paciente solicita una cita. 	Impacto (1-10)	8
		Prioridad (1-N)	14
		Estimación (h)	6

Tabla 18. Historia de usuario 14

Fuente: Elaboración propia

H15. GESTIÓN DE CONSULTAS			
Como	Médico		
Quiero	Ver listado de mis consultas agendadas		
Para	Gestionar y responder a cada consulta recibida.		
Validación	<ul style="list-style-type: none"> Visualizar listado de consultas recibidas. 	Impacto (1-10)	9
		Prioridad (1-N)	3
	<ul style="list-style-type: none"> Acciones (responder) para cada consulta. 	Estimación (h)	6

Tabla 19. Historia de usuario 15

Fuente: Elaboración propia

H16. ADMINISTRAR CONSULTAS			
Como	Paciente		
Quiero	Visualizar listado de mis consultas médicas		
Para	Manejar un control detallado de cada consulta.		
Validación	<ul style="list-style-type: none"> Visualizar listado de mis consultas programadas 	Impacto (1-10)	8
		Prioridad (1-N)	4
	<ul style="list-style-type: none"> Acciones (cancelar, editar, posponer) para cada consulta 	Estimación (h)	5

Tabla 20. Historia de usuario 16

Fuente: Elaboración propia

7.2. Product Backlog

El Product Backlog o pila de producto en un proyecto que sigue la metodología SCRUM consiste en una lista con todos los requerimientos iniciales del producto que se va a desarrollar. Se trata de una lista dinámica, que irá evolucionando a medida que lo hace el producto y el entorno del proyecto. La finalidad de crear esta lista no es otra que identificar las necesidades del producto para lograr su máxima utilidad.

Esta lista de Product Backlog contiene la descripción de las tareas que se van a realizar para la ejecución de cada requisito. Tareas que se organizarán en función de sus prioridades. Además, la pila de producto también indica una estimación del tiempo en la que cada tarea se va a desarrollar y el valor que cada una le da al producto. (EALDE, s.f.)

ID	Historia de usuario	Importancia	Tiempo estimado (h)
H1	Agendar cita médica	10	10
H3	Gestión de citas	9	6
H4	Administrar citas	9	3
H15	Gestión de consultas	9	6
H16	Administrar consultas	9	5
H2	Consulta médica	10	10
H5	Selección de médico	8	23
H6	Cancelar cita médica	7	3
H7	Reporte	7	3
H8	Registro de paciente	8	4
H9	Registro de médico	8	2
H10	Notificación recordatoria	8	11
H11	Comentarios y puntuaciones	9	12
H12	Citas fuera de la app	10	4
H13	Inicio de sesión	8	5
H14	Notificación de nueva cita	10	4

Tabla 21. Product Backlog

Fuente: Elaboración propia

7.3. Product Backlog organizado por Sprints

La siguiente tabla detalla las historias de usuario, calculando la importancia de cada una y su tiempo estimado. Para esto se ha designado una puntuación de 1 a 10 en cuanto al valor que aportan al proyecto.

ID	Historia de usuario	Estimación del valor (1-10)	Sprint programado	Inicio - fin
H1	Agendar cita médica	10	1	10/10/2022
H3	Gestión de citas	9	1	-
H4	Administrar citas	9	1	14/10/2022
H15	Gestión de consultas	9	2	17/10/2022
H16	Administrar consultas	9	2	-
H2	Consulta médica	10	2	21/10/2022
H5	Selección de médico	8	3	
H6	Cancelar cita médica	7	3	24/10/2022
H7	Reporte	7	3	-
H8	Registro de paciente	8	3	28/10/2022
H9	Registro de médico	8	3	
H10	Notificación recordatoria	8	4	
H11	Comentarios y puntuaciones	9	4	31/10/2022
H12	Citas fuera de la app	10	4	-
H13	Inicio de sesión	8	4	04/11/2022
H14	Notificación de nueva cita	10	4	

Tabla 22. Product Backlog organizado por Sprints

Fuente: Elaboración propia

7.4. Sprint Backlog

Sprint es el nombre que va a recibir cada uno de los ciclos o iteraciones que se tendrán dentro de un proyecto SCRUM. En cada Sprint o cada ciclo de trabajo lo que se va conseguir es lo que se denomina un entregable o incremento del producto, que aporte valor al cliente. (Mesa, 2018)

El Sprint Backlog es la suma del objetivo del Sprint, los elementos del Product Backlog elegidos para el Sprint, más un plan de acción de cómo crear el incremento de Producto. (García, 2020)

7.4.1. Sprint 1

ID	Historia de usuario	Tareas	Tiempo (h)
H1	Agendar cita médica	Diseñar y programar botón para citas en perfil del médico que redirigirá al formulario.	2
		Diseñar y validar formulario para agendar citas.	5
		Programar conexión a la BD para añadir la nueva cita.	3
H3	Gestión de citas	Diseñar y maquetar interfaz para mostrar lista de citas programadas.	2
		Programar conexión a la BD para obtener todas las citas programadas.	1
		Programar a cada cita, opciones de gestión como: cancelar, posponer, marcar como finalizada.	3
H4	Administrar citas	Diseñar y maquetar interfaz para mostrar lista de citas programadas.	2
		Programar conexión a la BD para obtener todas las citas programadas.	1
	Pruebas	Verificar que se cumple con ciertas funcionalidades	

Tabla 23. Sprint 1

Fuente: Elaboración propia

7.4.1.1. Entregables Sprint 1

Citas | Registro

Búsqueda directa

Dermatología **Diuris Jeyling**

Dirección: REMASA 2C al Oeste

Fecha: 11/15/2022

Hora: 01:00 PM

Tipo: En el consultorio A domicilio

REGISTRAR CITA

Ilustración 7. H1 - Agendar cita médica

Fuente: Elaboración propia

Citas

Búsqueda directa

Tipo de cita: Todas

Médico	Hora	Fecha	
Diuris Jeyling Lazo Maldana	1:00 PM	15/11/2022	
Mario Gazir Figueroa Nuñez	2:12 PM	18/10/2022	
Rodrigo Fabricio Fonseca Amador	3:05 PM	18/10/2022	

Ilustración 8. H3 - Gestión de citas

Fuente: Elaboración propia

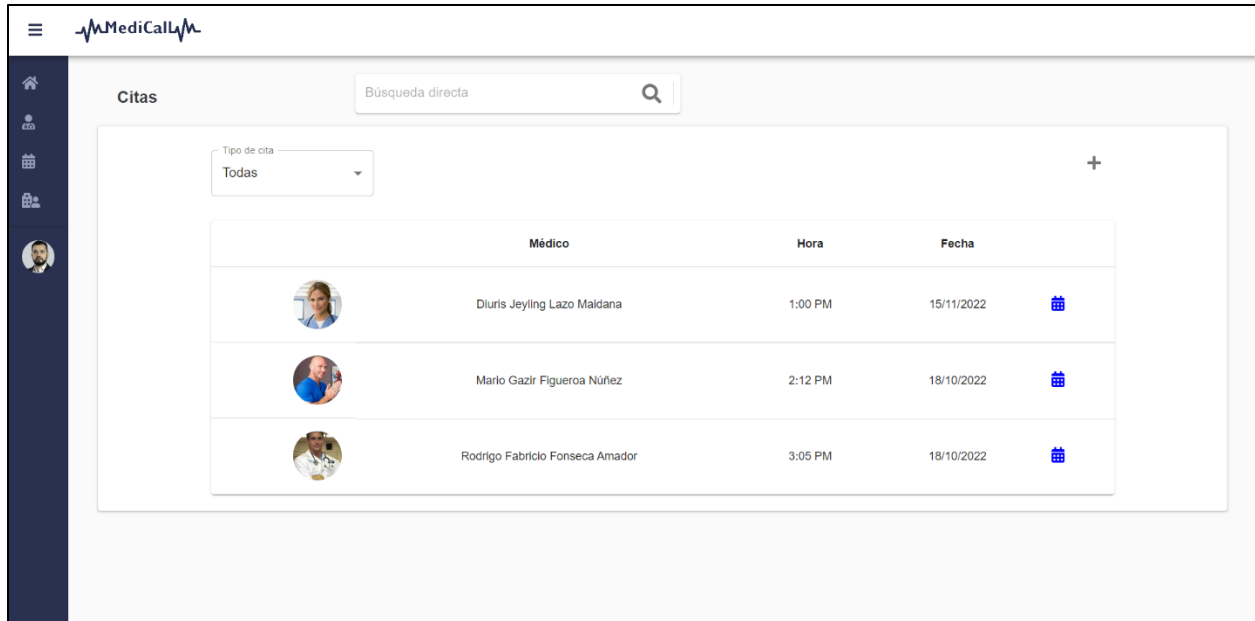


Ilustración 9. H4 - Administrar citas

Fuente: Elaboración propia

7.4.1.2. Pruebas de software Sprint 1

```

RUNS src/___tests___/register-meeting.test.js
PASS src/___tests___/register-meeting.test.js
  PRUEBAS SPRINT 1
    ✓ debería retorna listado de citas (1ms)
    ✓ debería retornar el listado de citas con la nueva cita agregada (1174ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        3.406s
Ran all test suites related to changed files.

```

Ilustración 10. Pruebas de software Sprint 1

Fuente: Elaboración propia

7.4.2. Sprint 2

Id	Historia de usuario	Tareas	Tiempo (h)
H2	Consulta médica	Diseñar y programar botón para consultas en perfil del médico que redirigirá al formulario.	2
		Diseñar y validar formulario para consultas.	5
		Programar conexión a la BD para añadir la nueva consulta.	3
H15	Gestión de consultas	Diseñar y maquetar interfaz para mostrar lista de consultas recibidas.	2
		Programar conexión a la BD para obtener todas las consultas recibidas.	1
		Programar a cada consulta, opciones para contestar.	3
H16	Administrar consultas	Diseñar y maquetar interfaz para mostrar lista de consultas.	2
		Programar conexión a la BD para obtener todas las consultas.	1
		Programar a cada consulta, opción de visualizar la respuesta del médico.	2
	Pruebas	Verificar que se cumple con ciertas funcionalidades	

Tabla 24. Sprint 2

Fuente: Elaboración propia

7.4.2.1. Entregables Sprint 2

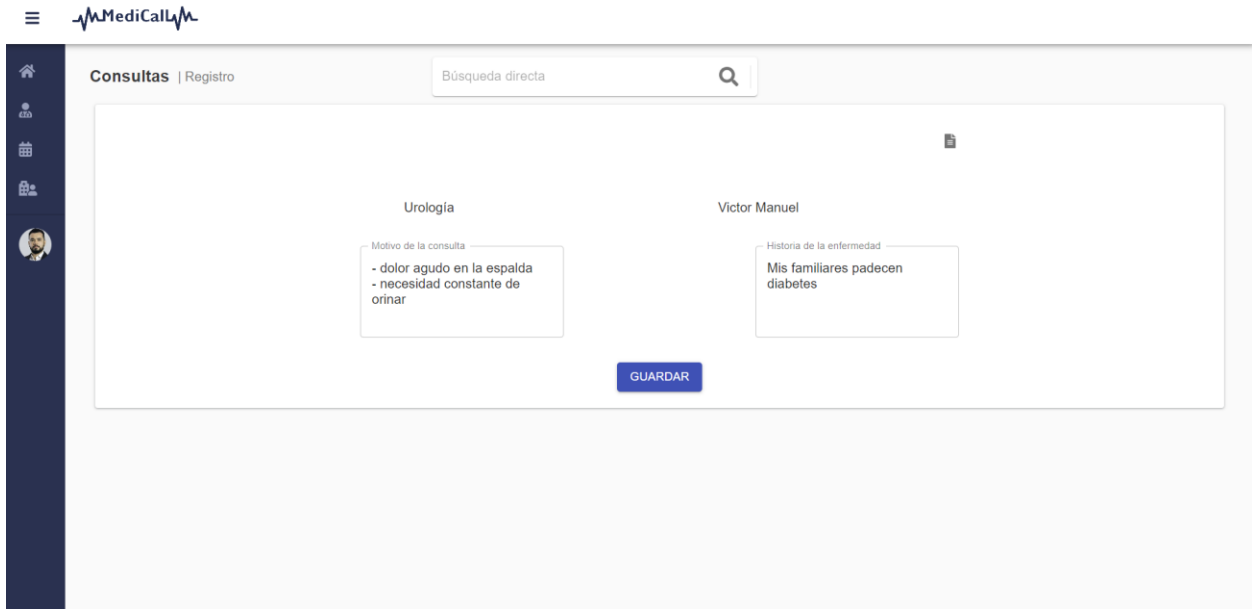


Ilustración 10. H2 - Consulta médica

Fuente: Elaboración propia

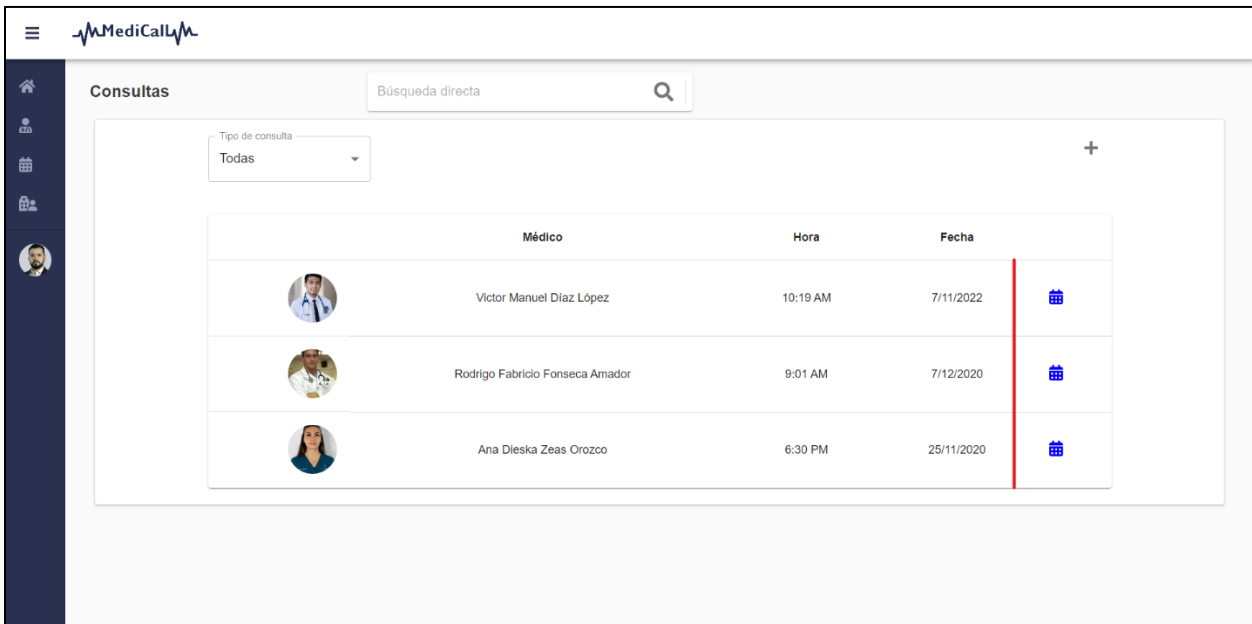


Ilustración 11. H15 - Gestión de consultas

Fuente: Elaboración propia

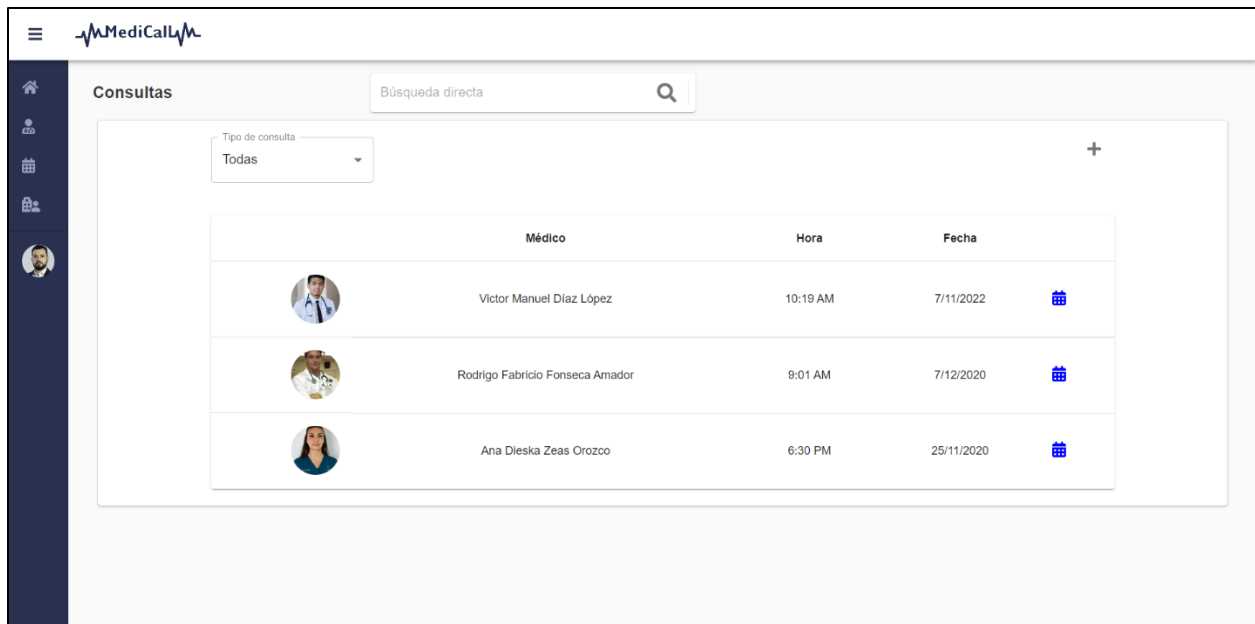


Ilustración 12. H16 - Administrar consultas

Fuente: Elaboración propia

7.4.2.2. Pruebas de software Sprint 2

```

PASS src/___tests___/register-consultation.test.js
PRUEBAS SPRINT 2
  ✓ debería retorna listado de consultas (1ms)
  ✓ debería retornar el listado de consultas con la nueva consulta agregada

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.846s, estimated 2s
Ran all test suites matching /register-consultation.test.js/i.

```

Ilustración 13. Pruebas de software Sprint 2

Fuente: Elaboración propia

7.4.3. Sprint 3

Id	Historia de usuario	Tareas	Tiempo
H5	Selección de médico	Diseñar y maquetar interfaz de filtro de médicos (búsqueda directa, especialidades, establecimientos, mejores puntuaciones).	2
		Programar conexión a la BD para la búsqueda directa de un médico.	3
		Diseñar y maquetar interfaz de lista de especialidades.	2
		Programar conexión a la BD para listar especialidades registradas.	2
		Diseñar y maquetar interfaz de lista de establecimientos.	2
		Programar conexión a la BD para listar establecimientos registrados.	2
		Diseñar y maquetar interfaz para listar médicos según filtro utilizado	2
		Programar conexión a la BD para obtener médicos según filtro utilizado.	3
		Diseñar y maquetar interfaz para mostrar información del perfil del médico seleccionado.	3
		Programar conexión a la BD para obtener información del médico seleccionado.	2
H6	Cancelar cita médica	Diseñar y maquetar botón a cada cita dentro del listado para cancelar cita.	1

		Programar conexión a la BD para cambiar el estado de la cita a cancelada.	2
H7	Reporte	Diseñar y maquetar botón a cada cita dentro del listado para reportar paciente	1
		Programar conexión para añadir reporte a paciente que no asistió ni canceló la cita	2
H8	Registro de paciente	Diseñar y maquetar interfaz de formulario de registro de pacientes	2
		Validar formulario antes de capturar información.	1
		Programar conexión a la BD para almacenar información capturada.	1
H9	Registro de médico	Diseñar y maquetar interfaz de formulario de registro de pacientes	2
		Validar formulario antes de capturar información.	1
		Validar formulario antes de capturar información.	1
	PRUEBAS	Verificar que se cumple con ciertas funcionalidades	

Tabla 25. Sprint 3

Fuente: Elaboración propia

7.4.3.1. Entregables Sprint 3

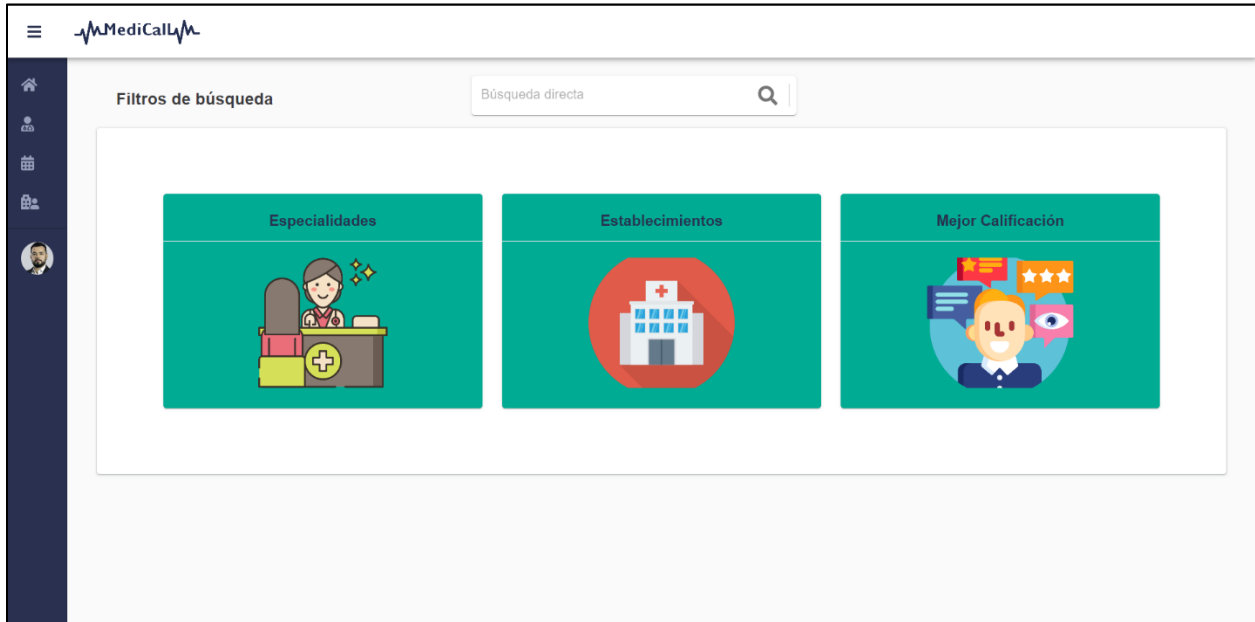


Ilustración 13. H5 - Selección de médico (filtros)

Fuente: Elaboración propia

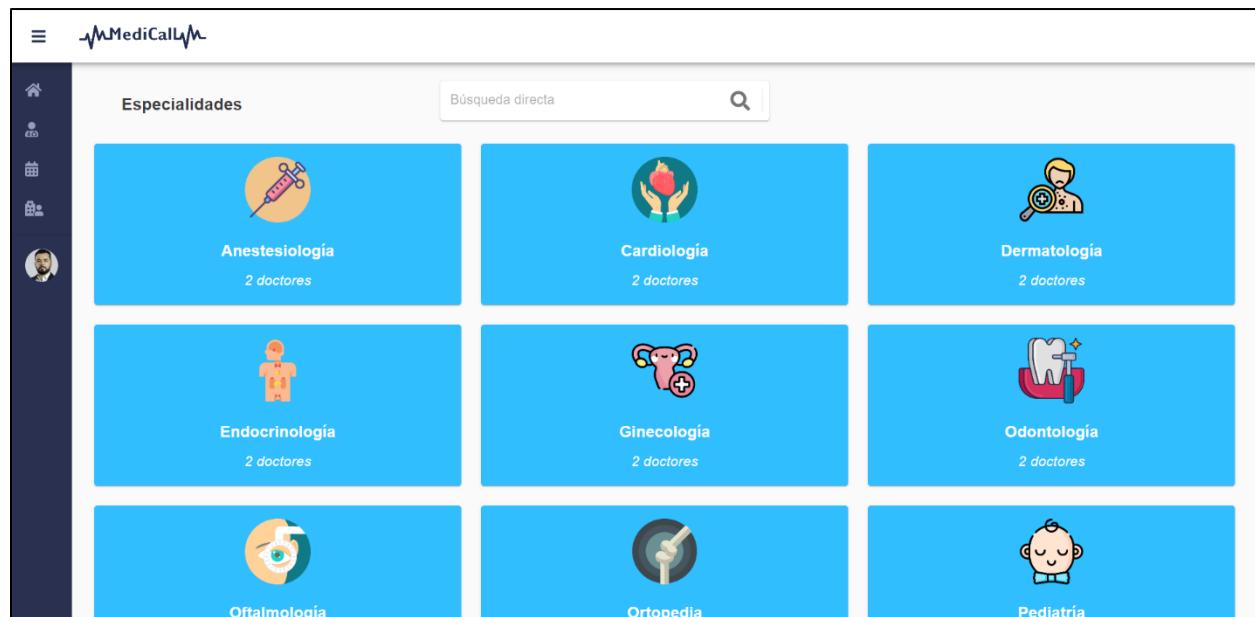


Ilustración 14. H5 - Selección de médico (especialidades)

Fuente: Elaboración propia

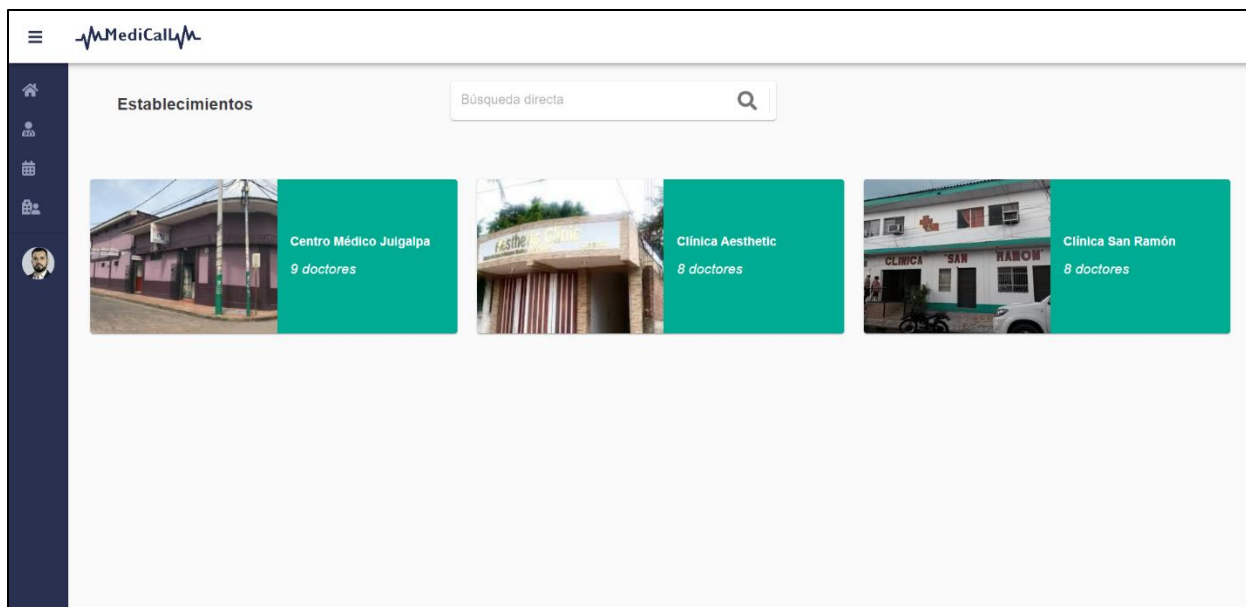


Ilustración 15. H5 - Selección de médico (establecimientos)

Fuente: Elaboración propia

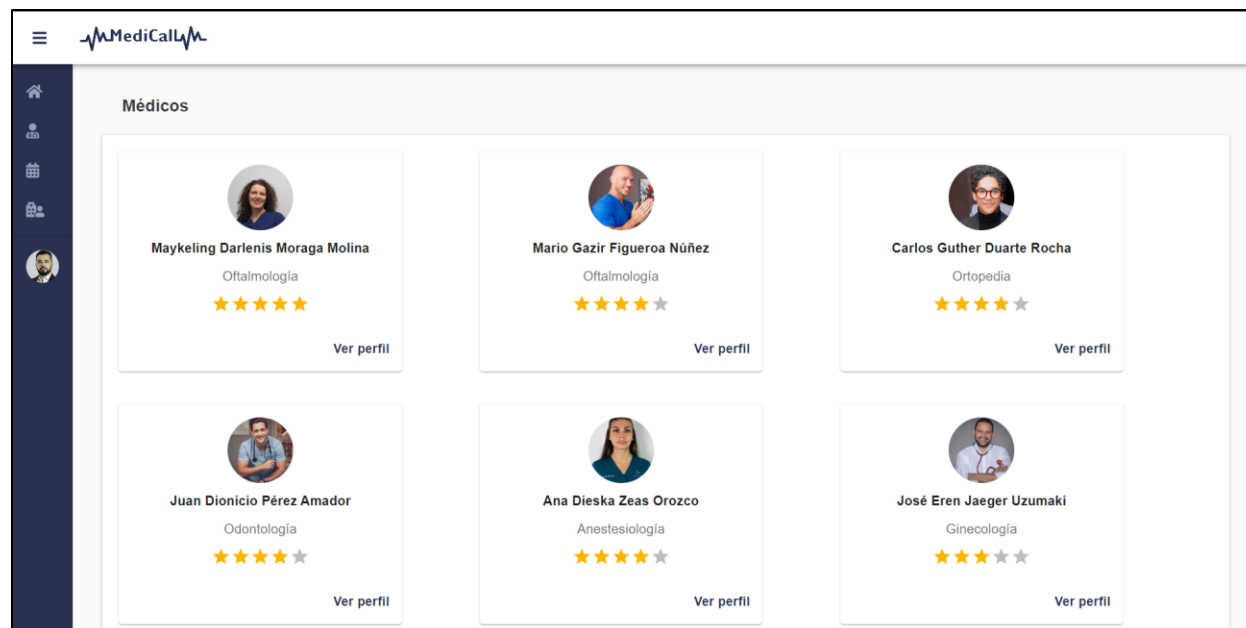


Ilustración 16. H5 - Selección de médico (listado)

Fuente: Elaboración propia

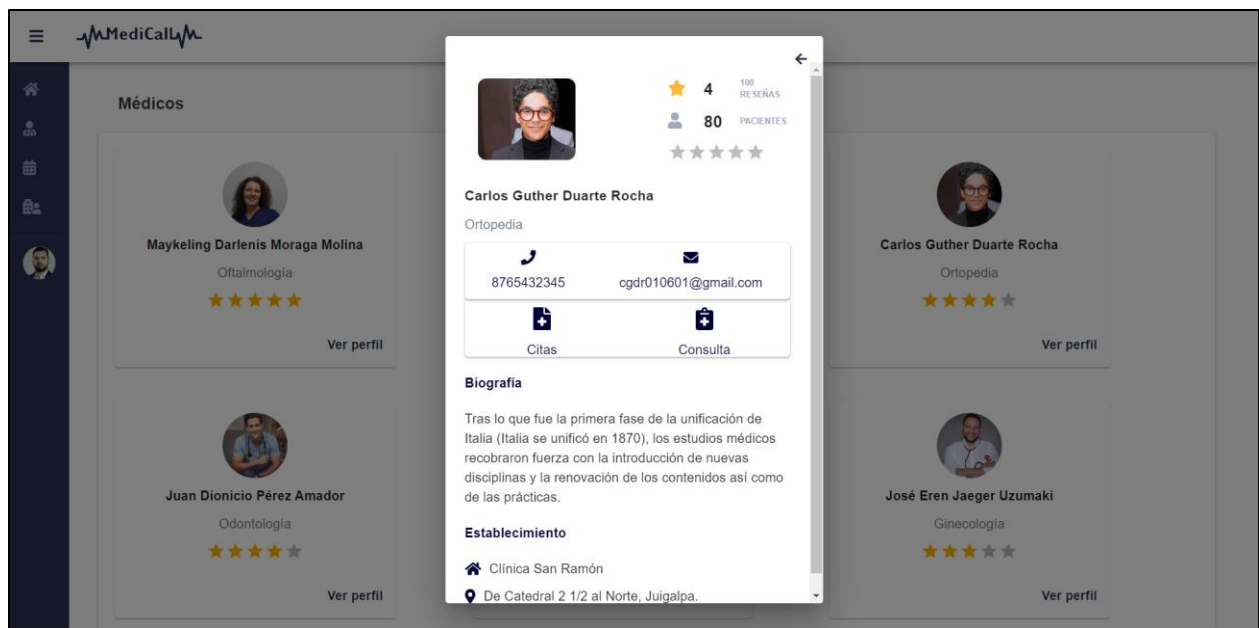


Ilustración 17. H5 - Selección de médico (perfil)

Fuente: Elaboración propia

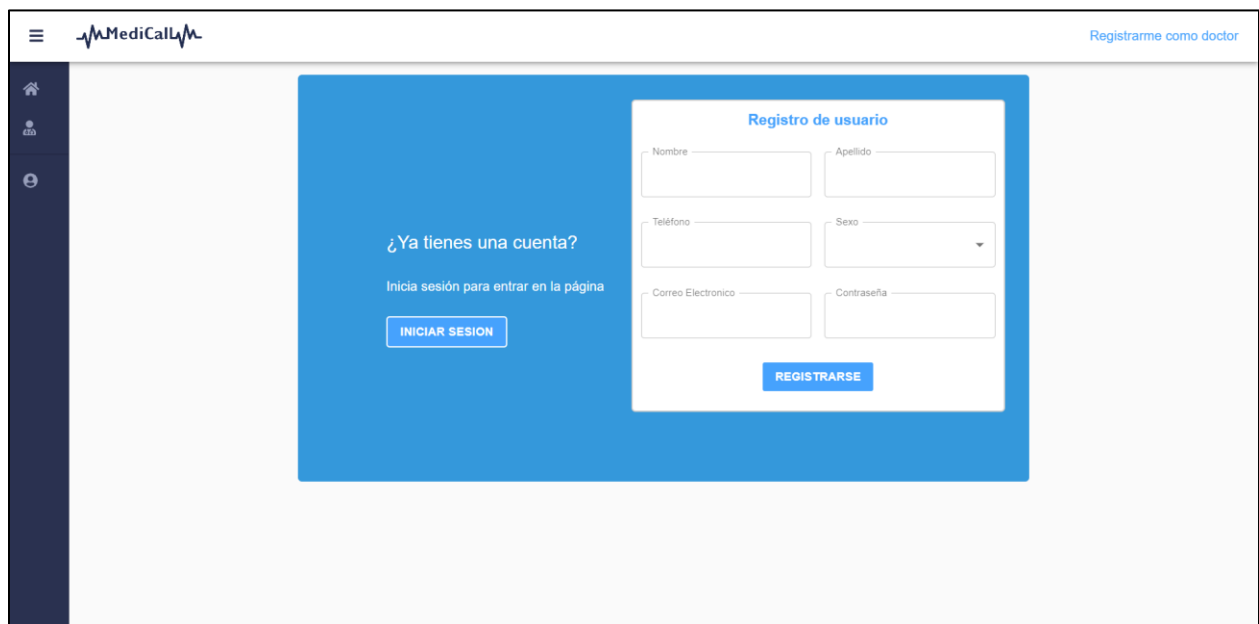


Ilustración 18. H8 - Registro de paciente

Fuente: Elaboración propia

7.4.3.2. Pruebas de software Sprint 3

```
PASS src/__tests__/register-user.test.js
PRUEBAS SPRINT 3
  ✓ Validar datos ingresados (1ms)
  ✓ debería retornar el listado de usuarios con el nuevo usuario agregado

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.834s, estimated 2s
Ran all test suites matching /register-user.test.js/i.
```

Ilustración 19. Pruebas de software Sprint 3

Fuente: Elaboración propia

7.4.4. Sprint 4

ID	Historia de usuario	Tareas	Tiempo
H10	Notificación recordatoria	Programar notificación para que el paciente otorgue permisos para recibir notificaciones	3
		Diseñar y maquetar input en cada cita para que el paciente indique la hora en que quiere recibir la notificación recordatorio.	2
		Programar conexión a la BD para almacenar la hora proporcionada	2
		Programar servicio de Firebase para enviar notificación en la hora indicada.	4
H11	Comentarios y puntuaciones	Diseñar y maquetar botón (estrellas) para indicar la puntuación enviada.	2
		Programar conexión a la BD para almacenar la puntuación indicada.	2
		Diseñar y maquetar input para realizar comentarios del servicio médico	2
		Programar conexión a la BD para almacenar mensaje recibido.	2
		Diseñar y maquetar interfaz donde se mostrarán los mensajes recibidos	2
		Programar conexión a la BD para obtener los mensajes recibidos.	2
H12	Citas fuera de la app	Diseñar y maquetar botón para añadir cita dentro del listado de citas	2
		Redirigir a la interfaz de añadir nueva cita	2

H13	Inicio de sesión	Diseñar y maquetar interfaz de formulario de inicio de sesión	2
		Validar formulario antes de capturar credenciales.	1
		Programar conexión a la BD para validar si existe usuario con las credenciales capturadas.	1
		Mostrar vista de la aplicación según rol de usuario.	1
H14	Notificación de nueva cita	Programar notificación para que el médico otorgue permisos para recibir notificaciones	2
		Programar servicio de Firebase para enviar notificación al médico cuando se solicita nueva cita.	4
	PRUEBAS	Verificar que se cumple con ciertas funcionalidades	

Tabla 26. Sprint 4

Fuente: Elaboración propia

7.4.4.1. Entregables Sprint 4

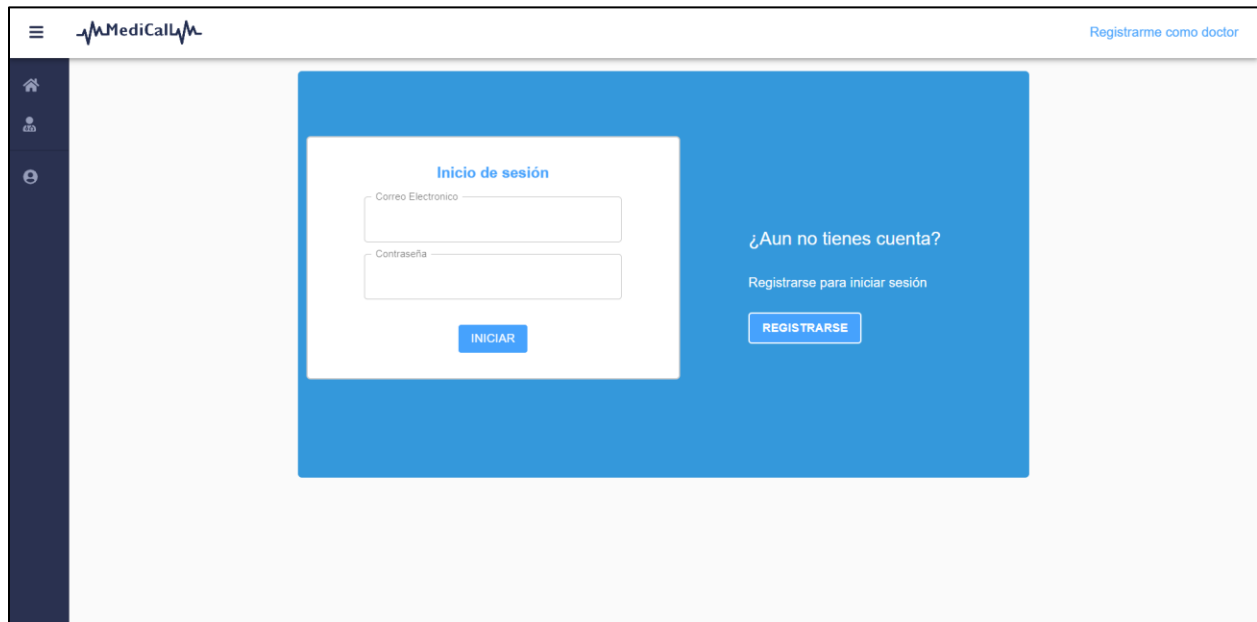


Ilustración 19. H13 - Inicio de sesión

Fuente: Elaboración propia

7.4.4.2. Pruebas de software Sprint 4

```
PASS src/___tests___/login.test.js
PRUEBAS SPRINT 4
  ✓ validar campos: correo y contraseña (1ms)
  ✓ debería retornar true si se encuentra un usuario con el correo y contraseña ingresados

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.832s, estimated 2s
Ran all test suites matching /login.test.js/i.
```

Ilustración 20. Pruebas de software Sprint 4

Fuente: Elaboración propia

VIII. Presupuesto

El presupuesto intenta determinar cuánto dinero, esfuerzo, recursos y tiempo supondrá constituir un sistema o producto específico de software; es importante porque nos permite estimar el costo de un software antes de empezar a construirlo. (Ing Software UMG)

El presupuesto proviene de la estimación de los distintos gastos involucrados en el desarrollo de la aplicación en cada una de sus etapas, cabe recalcar que la información referente a precios o salarios mínimos provienen de la página web (Trabajo y pago, 2022). Dichos precios o salarios se convirtieron a la moneda dólar con el tipo de cambio actual que consta de C\$ 35.88 por \$1 en la fecha 7 de noviembre de 2022.

	Tasa/Mes	Meses	Total
Recursos humanos			
Médico general (asesoría)	\$258.69	0.5	\$129.345
Desarrollador	\$215	3	\$645
Diseñador	\$171.37	3	\$514.11
Pruebas (tester)	\$134.71	0.3	\$44.9
Servicios			
Energía eléctrica	\$14	3	\$42
Internet	\$23	3	\$69
Gestor de base de datos	\$25	3	\$75
Hosting y Dominio (plan premium en Hostinger)	\$2.49	3	\$7.47
Agua	\$2.79	3	\$8.37
Software			
Software diseño	\$19	3	\$57
Software utilitario	\$8	3	\$24
	Valor unitario	Cantidad	Total
Mobiliario y equipo			
Computadora	\$950	1	\$950
Silla	\$40	1	\$40
Mesa	\$35	1	\$35
Total			\$2 641.195

Tabla 27. Presupuesto

Fuente: Elaboración propia

IX. Marco lógico

Según la CEPAL en su SERIE manuales N° 42, (Edgar, 2015) la metodología contempla dos etapas, que se desarrollan paso a paso en las fases de identificación y de diseño del ciclo de vida del proyecto:

- **Identificación del problema y alternativas de solución:** Se analiza la situación existente para crear una visión de la situación deseada y seleccionar las estrategias que se aplicarán para conseguirla. Existen cuatro tipos de análisis para realizar: el análisis de involucrados, el análisis de problemas, el análisis de objetivos y el análisis de estrategias. En esta etapa se realiza el árbol del problema, el árbol de objetivos y el árbol de la solución.
- **La etapa de planificación,** en la que la idea del proyecto se convierte en un plan operativo práctico para la ejecución. En esta etapa se elabora la matriz de marco lógico. Las actividades y los recursos son definidos en cierto tiempo.

9.1. Árbol del problema

El árbol del problema es una técnica participativa que ayuda a definir problemas, causas y efectos de manera organizada. Genera un modelo de relaciones causales entorno a un problema.

Se origina con la identificación del problema más importante, que constituirá el tronco del árbol; luego, se identifica sus causas – raíces del tronco -, y consecuencias, los frutos. En las raíces se expresan las causas y orígenes del problema central; pueden jerarquizarse, ordenándose en causas principales y secundarias. El mismo proceso con las consecuencias. (CISE, s.f.)

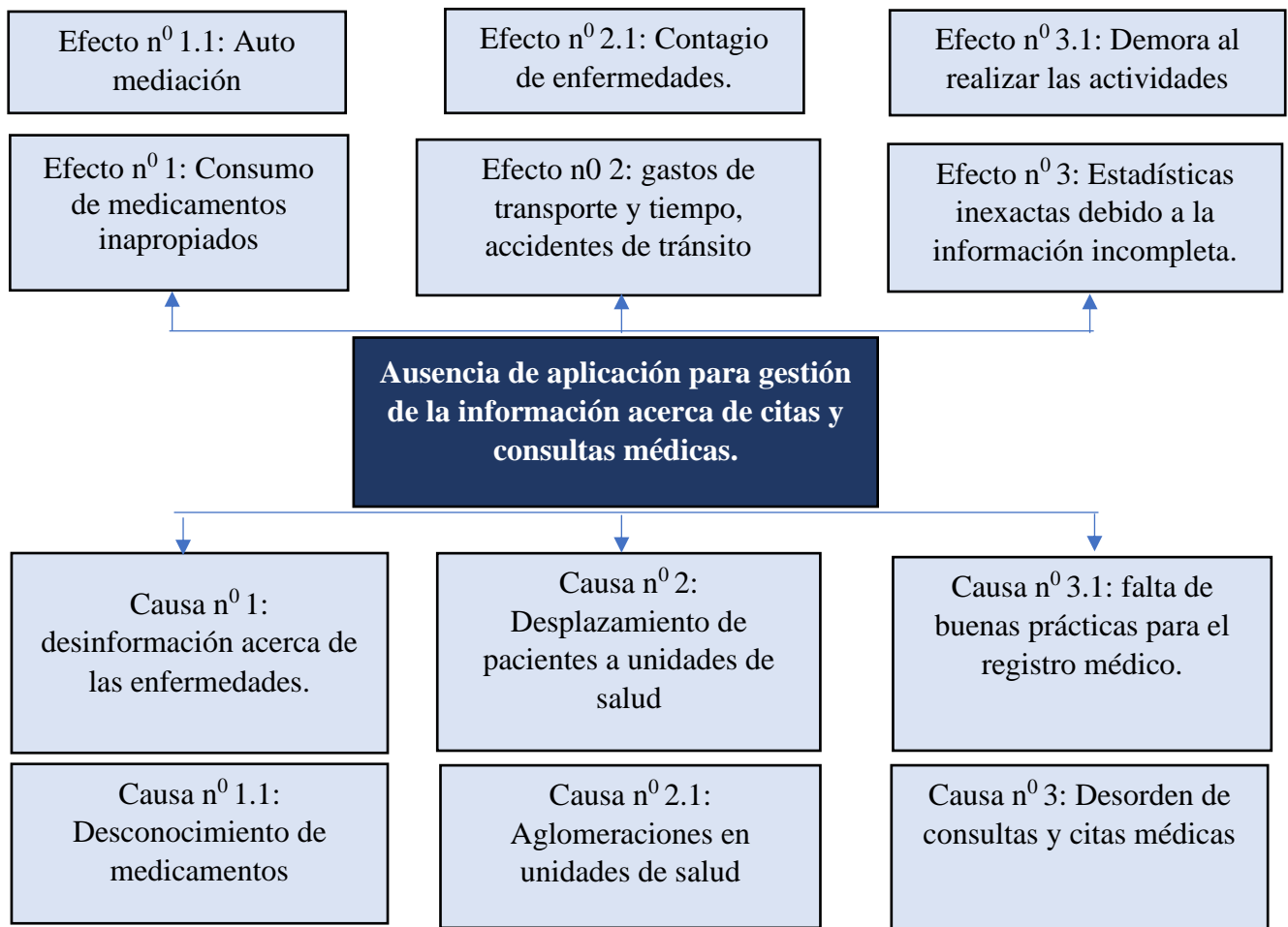


Ilustración 20. Árbol del problema

Fuente: Elaboración propia

9.2. Árbol de objetivos

El árbol de objetivos reúne los medios y alternativas para solucionar el problema principal. Gracias a ello, se logra una visión positiva de las situaciones negativas que aparecían en el árbol anterior, aunque utilice la misma estructura. Así, se busca ir resolviendo el problema paso a paso. (PROED, s.f.)

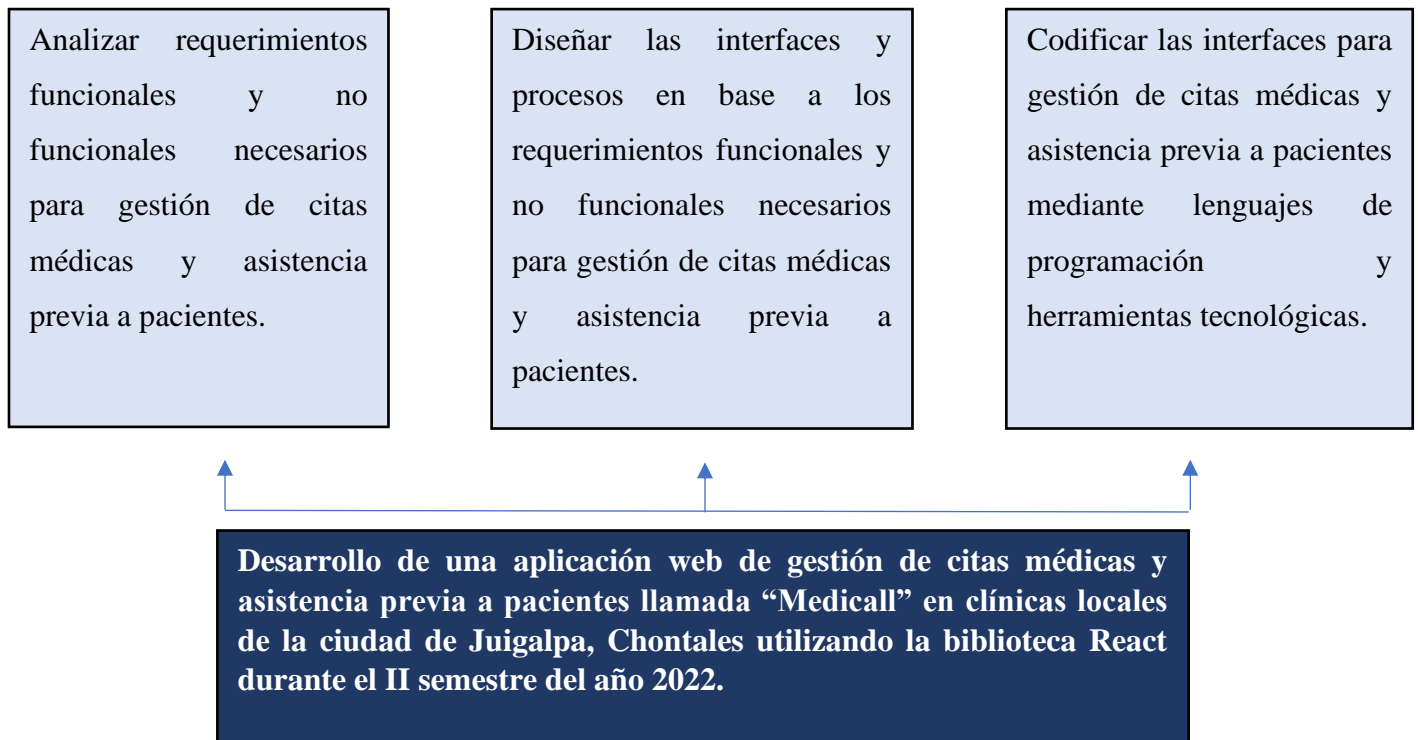


Ilustración 21. Árbol de objetivos

Fuente: Elaboración propia

9.3. Árbol de la solución

El árbol de soluciones es una herramienta similar y complementaria al árbol de problemas, que pretende facilitar la comprensión amplia de la solución y su formulación en términos operativos. El objetivo de este análisis es identificar los cambios requeridos para que el problema sea resuelto. La construcción del árbol de resultados esperados se realiza con base en el

árbol de problemas realizado anteriormente, transformando las situaciones desfavorables en las situaciones contrarias que pudieran resolver el malestar generado por la situación problema, y ajustando estos planteamientos a los cambios que estén al alcance del analista. (Junta de Andalucía, 2011)

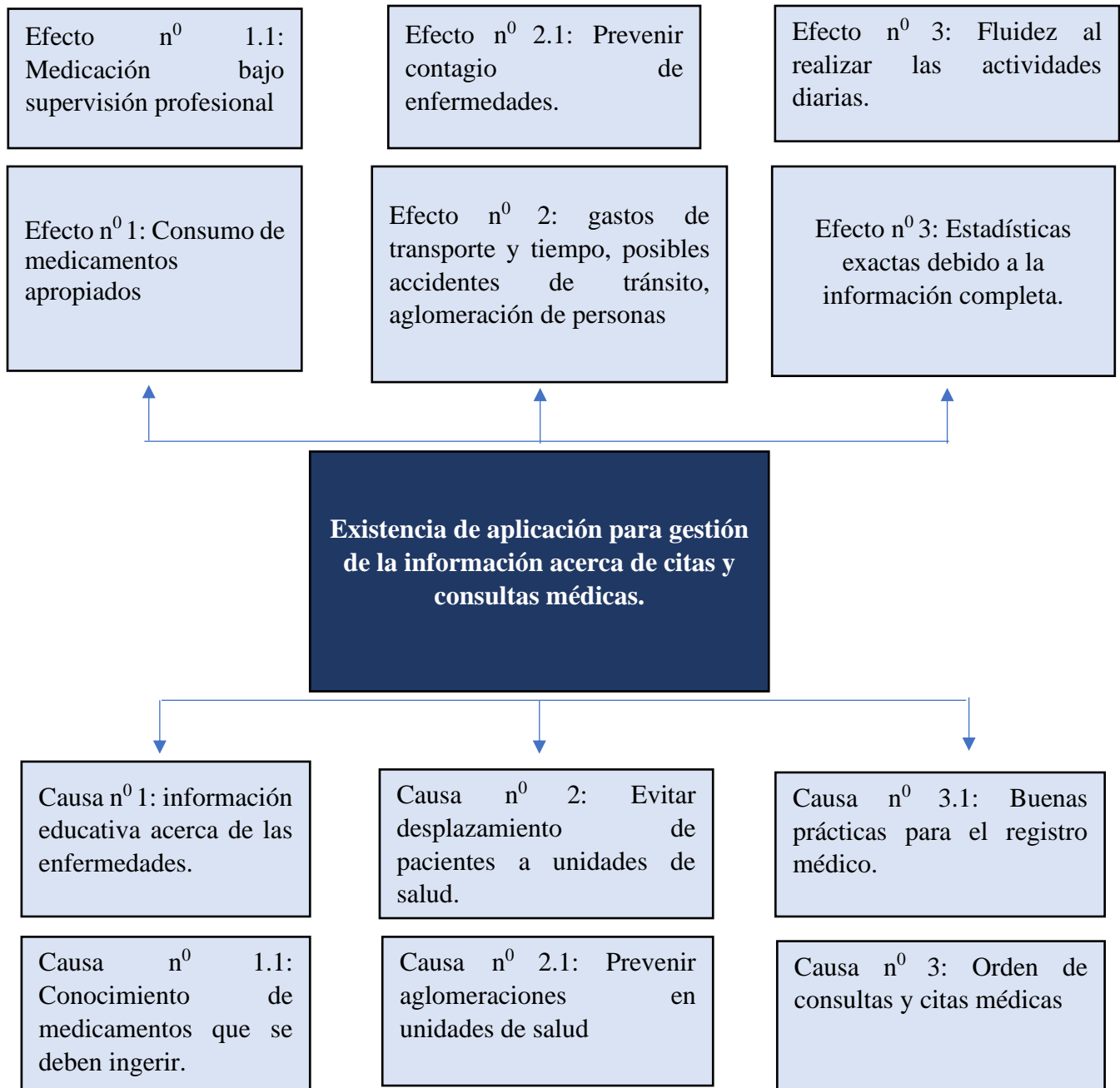


Ilustración 22. Árbol de la solución

Fuente: Elaboración propia

9.4. Matriz de marco lógico

La Matriz de Marco Lógico es un resumen narrativo que sintetiza las actividades del proyecto, los productos que se entregarán, y los resultados de corto, mediano y largo plazo que se esperan lograr en la población objetivo. (Legalidad por México, s.f.)

	Descripción	Indicador	Medios de Verificación	Supuestos
F I N	Disminuir la automedicación y largos plazos para recibir atención médica.	Aumentar el uso de prescripción médica. Disminuir los plazos para llevar a cabo una cita o consulta médica.	Historial de citas y consultas realizadas	Aceptación por parte de médicos y pacientes.
P R O P Ó S I T O	Crear una aplicación web para gestión de citas médicas y asistencia previa a pacientes.	Cumplimiento de cronograma de actividades Desarrollo de una aplicación web mínimo 90% durante el 2do semestre del 2022.	Cantidad de interfaces y funcionalidades terminadas durante el 2do semestre del 2022.	Los usuarios de la aplicación no realizan consultas en línea y no agendan citas en línea.
C O M P O N E N T E S	Incorporar funcionalidad para que los médicos y pacientes puedan registrarse en la aplicación Añadir funcionalidad para que los pacientes puedan realizar consultas en línea Agregar funcionalidad para que los pacientes puedan agendar cita médica por medio de la aplicación.	Nivel de satisfacción con el sitio web Eficacia para adquirir atención médica Grado de utilidad del manual de usuario para médicos y pacientes en el uso de la aplicación web.	Registro de médicos Registro de pacientes Historial de citas y consultas realizadas.	Uso de la aplicación web por parte de los médicos y pacientes.

	Descripción	Indicador	Medios de Verificación	Supuestos
A C T I V I D A D E S	Componente 1			
	Desarrollar interfaz para que los médicos puedan afiliarse	Funcionamiento del 100% de la interfaz al culminar el tercer sprint.	Interfaz de registro de médicos	Cambios en la disponibilidad de las tecnologías de desarrollo utilizadas.
	Desarrollar interfaz para que los pacientes puedan registrarse	Funcionamiento del 100% de la interfaz al culminar el tercer sprint.	Interfaz de registro de pacientes	
	Componente 2			Modificaciones en las tecnologías de desarrollo.
	Desarrollar interfaz para que los pacientes puedan seleccionar un médico	Funcionamiento del 100% de la interfaz al culminar el tercer sprint	Filtros de médicos	Modificaciones en las tecnologías de desarrollo.
	Desarrollar interfaz para que los pacientes puedan realizar una consulta en línea	Funcionamiento del 100% de la interfaz al culminar el segundo sprint	Interfaz de creación de nueva consulta	
Componente 3			Obsolescencia de las tecnologías de desarrollo utilizadas.	
Desarrollar interfaz para que los pacientes puedan agendar una cita médica	Funcionamiento del 100% de la interfaz al culminar el primer sprint	Interfaz de creación de nueva cita	Obsolescencia de las tecnologías de desarrollo utilizadas.	

Tabla 28. Matriz de marco lógico

Fuente: Elaboración propia

X. Cronograma de actividades

El cronograma es una herramienta esencial para elaborar calendarios de trabajo o actividades. Un documento en el que se establece la duración de un proyecto, la fecha de inicio y final de cada tarea; es decir, una manera sencilla de organizar el trabajo. (Bizneo, 2020)

La gestión del proyecto para el desarrollo de la aplicación web Medicall, se ha previsto para realizarse en un total de 14 semanas.

10.1. Tabla

	Nombre de tarea	Duración	Comienzo	Fin
	Desarrollo de aplicación web para gestión de citas médicas y asistencia previa a pacientes PJ1	69 días	lun 8/8/22	mié 11/16/22
	Análisis PJ1.1	3 días	lun 8/8/22	mié 8/10/22
	Marco lógico PJ1.1.1	3 días	lun 8/8/22	mié 8/10/22
	Árbol del problema PJ1.1.1.1	1 día	lun 8/8/22	lun 8/8/22
	Árbol de objetivos PJ1.1.1.2	1 día	mar 8/9/22	mar 8/9/22
	Árbol de la solución PJ1.1.1.3	1 día	mié 8/10/22	mié 8/10/22
	Diseño PJ1.2	10 días	jue 8/11/22	mié 8/24/22
	Diseño de funcionamiento PJ1.2.1	2 días	jue 8/11/22	lun 8/22/22
	Diseño de navegación PJ1.2.2	1 día	mar 8/23/22	mar 8/23/22
	Diseño arquitectónico PJ1.2.3	1 día	mié 8/24/22	mié 8/24/22
	Justificación de la tecnología PJ1.3	1 día	jue 8/25/22	jue 8/25/22
	Justificación de la metodología PJ1.4	1 día	vie 8/26/22	vie 8/26/22
	Elaboración del modelo CANVAS PJ1.5	2 días	lun 8/29/22	mar 8/30/22
	Elaboración de propuesta de valor PJ1.6	2 días	mié 8/31/22	jue 9/1/22
	Elaboración de oportunidad de mercado PJ1.7	1 día	vie 9/2/22	vie 9/2/22
	Definición del ciclo del proyecto PJ1.8	10 días	lun 9/5/22	mar 9/20/22
	Creación de historias de usuario PJ1.8.1	5 días	lun 9/5/22	vie 9/9/22
	Definición de actividades PJ1.8.2	3 días	lun 9/12/22	vie 9/16/22
	Definición de Sprints PJ1.8.3	2 días	lun 9/19/22	mar 9/20/22
	Desarrollo PJ1.9	22 días	mié 9/21/22	vie 10/21/22
	Sprint 1 PJ1.9.1	5 días	mié 9/21/22	mar 9/27/22
	Agendar cita médica PJ1.9.1.1	2 días	mié 9/21/22	jue 9/22/22
	Gestión de citas PJ1.9.1.2	2 días	vie 9/23/22	lun 9/26/22
	Administrar citas P.11 9 1 3	1 día	mar 9/27/22	mar 9/27/22

Ilustración 23. Tabla cronograma de actividades parte 1




	Nombre de tarea	Duración	Comienzo	Fin
	♣ Sprint 2 PJ1.9.2	5 días	mié 9/28/22	mar 10/4/22
	Gestión de consultas PJ1.9.2.1	2 días	mié 9/28/22	jue 9/29/22
	Administrar consultas PJ1.9.2.2	1 día	vie 9/30/22	vie 9/30/22
	Consulta médica PJ1.9.2.3	2 días	lun 10/3/22	mar 10/4/22
	♣ Sprint 3 PJ1.9.3	6 días	mié 10/5/22	jue 10/13/22
	Selección de médico PJ1.9.3.1	3 días	mié 10/5/22	vie 10/7/22
	Cancelar cita médica PJ1.9.3.2	1 día	lun 10/10/22	lun 10/10/22
	Reporte PJ1.9.3.3	1 día	mar 10/11/22	mar 10/11/22
	Registro de paciente PJ1.9.3.4	0.5 días	jue 10/13/22	jue 10/13/22
	Registro médico PJ1.9.3.5	0.5 días	jue 10/13/22	jue 10/13/22
	♣ Sprint 4 PJ1.9.4	6 días	vie 10/14/22	vie 10/21/22
	Notificación recordatorio PJ1.9.4.1	1 día	vie 10/14/22	vie 10/14/22
	Comentarios y puntuaciones PJ1.9.4.2	2 días	lun 10/17/22	mar 10/18/22
	Citas fuera de la app PJ1.9.4.3	1 día	mié 10/19/22	mié 10/19/22
	Inicio de sesión PJ1.9.4.4	1 día	jue 10/20/22	jue 10/20/22
	Notificación de nueva cita PJ1.9.4.5	1 día	vie 10/21/22	vie 10/21/22
	Pruebas de software PJ1.10	10 días	lun 10/24/22	lun 11/7/22
	Manual de usuario PJ1.11	3 días	mar 11/8/22	jue 11/10/22
	Manual técnico PJ1.12	4 días	vie 11/11/22	mié 11/16/22
	Entrega del Documento Final PJ1.13	1 día?	mié 2/15/23	mié 2/15/23
	Presentación del Proyecto PJ1.13	1 día?	mié 2/22/23	mié 2/22/23

Ilustración 24. Tabla cronograma de actividades parte 2

10.2. Diagrama de Gantt

Un diagrama de Gantt es una herramienta de gestión de proyectos que se emplea para visualizar todas las tareas desde el inicio de un proyecto hasta su finalización.

Ya que tiene la estructura similar a una línea de tiempo, el diagrama de Gantt presenta cada tarea y sus dependencias, la titularidad por las tareas, la duración de la tarea y la duración prevista para todo el proyecto. (Lucidchart, 2022)

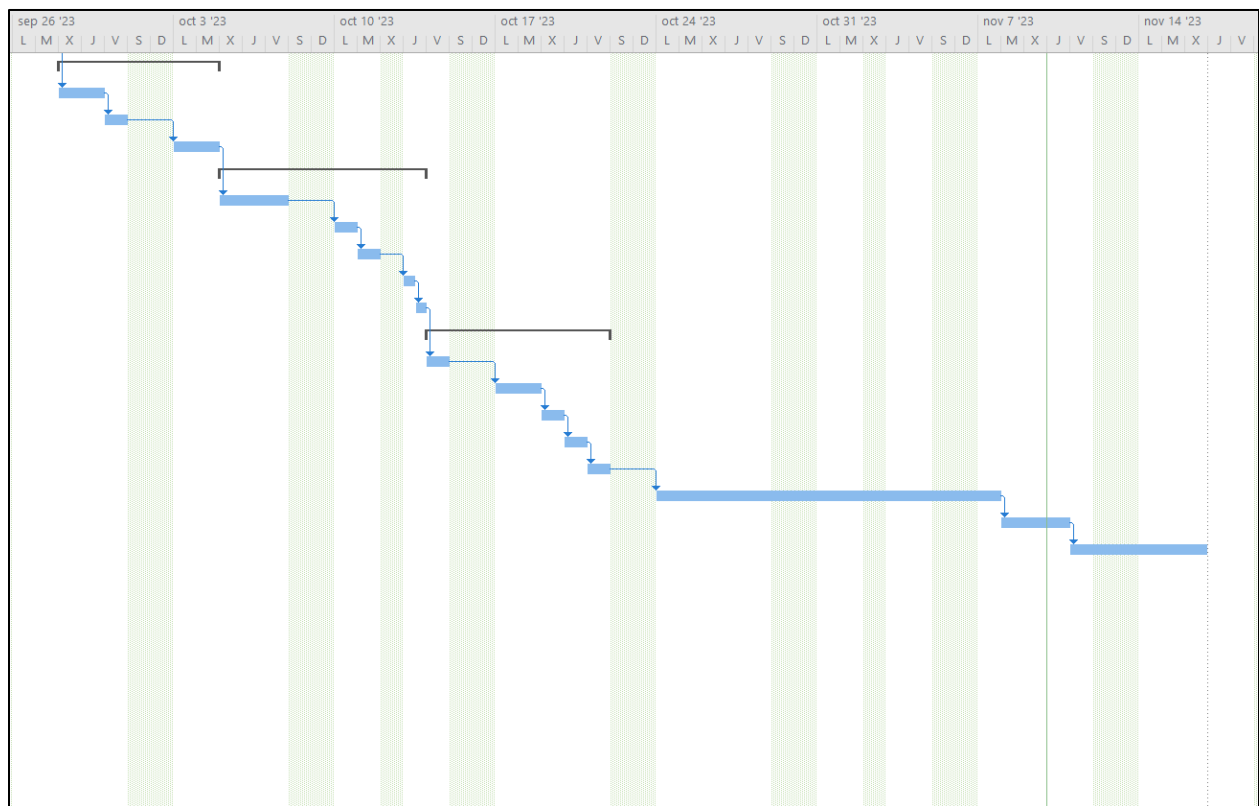


Ilustración 25. Diagrama de Gantt

Fuente: Generado por Microsoft Project

XI. Referencias

- Abba, I. (5 de Julio de 2022). *Kinsta*.
- Angular. (s.f.). *Angular*. Obtenido de <https://angular.io/docs>
- Angular. (s.f.). *Angular*. Obtenido de <https://angular.io/guide/what-is-angular>
- Azaustre, C. (25 de Mayo de 2018). *Carlos Azaustre*. Obtenido de <https://carlosazaustre.es/como-funciona-flux>
- Bizneo. (18 de Noviembre de 2020). *Bizneo*. Obtenido de <https://www.bizneo.com/blog/cronograma/#:~:text=El%20cronograma%20es%20una%20herramienta,sencilla%20de%20organizar%20el%20trabajo>.
- Bootstrap*. (s.f.). Obtenido de <https://getbootstrap.com/>
- CISE. (s.f.). *Centro de Investigaciones y Servicios Educativos*.
- EALDE. (s.f.). *EALDE*. Obtenido de <https://www.ealde.es/product-backlog-sprint-backlog/>
- Edgar, O. J. (2015). *CEPAL*. Obtenido de CEPAL: https://repositorio.cepal.org/bitstream/handle/11362/5607/S057518_es.pdf
- Edwards, P. (25 de Septiembre de 2020). *Toptal*. Obtenido de Toptal: <https://www.toptal.com/google/firebase-serverless-mobile-and-web-apps>
- Fetecua, A. (7 de Octubre de 2022). *Design Plus*. Obtenido de <https://designplus.co/blog/disenio-web/que-es-un-mapa-de-navegacion-web/>
- Firestore. (s.f.). Obtenido de <https://firebase.google.com/>
- Flux. (s.f.). Obtenido de <https://facebook.github.io/flux>
- García, M. (17 de Julio de 2020). *IT tude*. Obtenido de <https://ittude.com.ar/b/scrum/que-es-el-sprint-backlog/>
- Guerrero, N. (25 de Mayo de 2018). *Programa en línea*. Obtenido de programaenlinea.net
- Higuerey, E. (7 de Mayo de 2019). *rockcontent*. Obtenido de <https://rockcontent.com/es/blog/propuesta-de-valor/>
- HubSpot*. (1 de Febrero de 2022). Obtenido de <https://blog.hubspot.es/marketing/crear-propuesta-de-valor>
- Ing Software UMG*. (s.f.). Obtenido de <https://ingsoftwareumg.files.wordpress.com/2012/05/presupuestos-2.pdf>
- JQuery. (s.f.). Obtenido de JQuery: <https://jquery.com/>

Junta de Andalucía. (17 de Noviembre de 2011). Obtenido de <https://www.buenastareas.com/ensayos/Arbol-De-Soluciones/3129356.html>

K. Schwaber, J. S. (2013). *La Guía Definitiva de SCRUM: Las Reglas del Juego.* Obtenido de <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Legalidad por México. (s.f.). Obtenido de <https://legalidadpormexico.org/mcl/modulo1/matrizdeml.html>

Lucidchart. (2022). *Lucidchart.* Obtenido de <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-gantt>

Material UI. (s.f.). Obtenido de <https://mui.com/>

Mesa, A. R. (19 de Diciembre de 2018). *OpenWebinars.* Obtenido de <https://openwebinars.net/blog/que-es-un-sprint-scrum/>

MUI. (s.f.). Obtenido de <https://mui.com/>

PROED. (s.f.). *Tecnicatura en Gestión Universitaria.* Obtenido de <https://patgu.eco.catedras.unc.edu.ar/unidad-3/herramientas/arbol-de-objetivos-y-tormenta-de-ideas/>

Programación - Diseño - Web. (11 de Junio de 2013). Obtenido de <https://carrillosystem.wordpress.com/2013/06/11/jquery-que-es-origenes-ventajas-y-desventajas/>

React. (2022). *React.* Obtenido de <https://es.reactjs.org/>

React. (s.f.). *React.* Obtenido de <https://reactjs.org/>

React. (s.f.). *React JS.* Obtenido de <https://es.reactjs.org/>

Redacción APD. (13 de Enero de 2022). Obtenido de <https://www.apd.es/metodologia-scrum-que-es/#:~:text=Planificaci%C3%B3n%3A%20Product%20Backlog,objetivos%20al%20comienzo%20del%20proyecto.>

Redux. (s.f.). Obtenido de <https://redux.js.org/>

Rehkopf, M. (2019). *Atlassian.* Obtenido de <https://www.atlassian.com/es/agile/project-management/user-stories>

Rock Content. (12 de Abril de 2020). Obtenido de <https://rockcontent.com/es/blog/bootstrap/>

Rodríguez, J. L. (26 de Enero de 2021). *atSistemas.* Obtenido de <https://www.atsistemas.com/es/blog/que-es-tailwind>

Tay, Y. (6 de Marzo de 2022). *Flux.* Obtenido de <https://facebook.github.io/flux/docs/in-depth-overview.html#content>

Trabajo y pago. (2022). Obtenido de <https://tusalario.org/nicaragua/tu-carrera-profesional/nicaragua-trabajo-y-pago>

UNID. (25 de Agosto de 2015). *Universidad Interamericana para el Desarrollo*. Obtenido de <https://www.unid.edu.mx/>

Zúñiga, F. G. (10 de Agosto de 2015). *Arsys - Programación y BBDD*. Obtenido de <https://www.arsys.es/blog/bootstrap-responsive>

XII. Anexos

12.1. Modelo CANVAS



Ilustración 26. Modelo CANVAS

Fuente: Elaboración propia

12.2. Base de datos

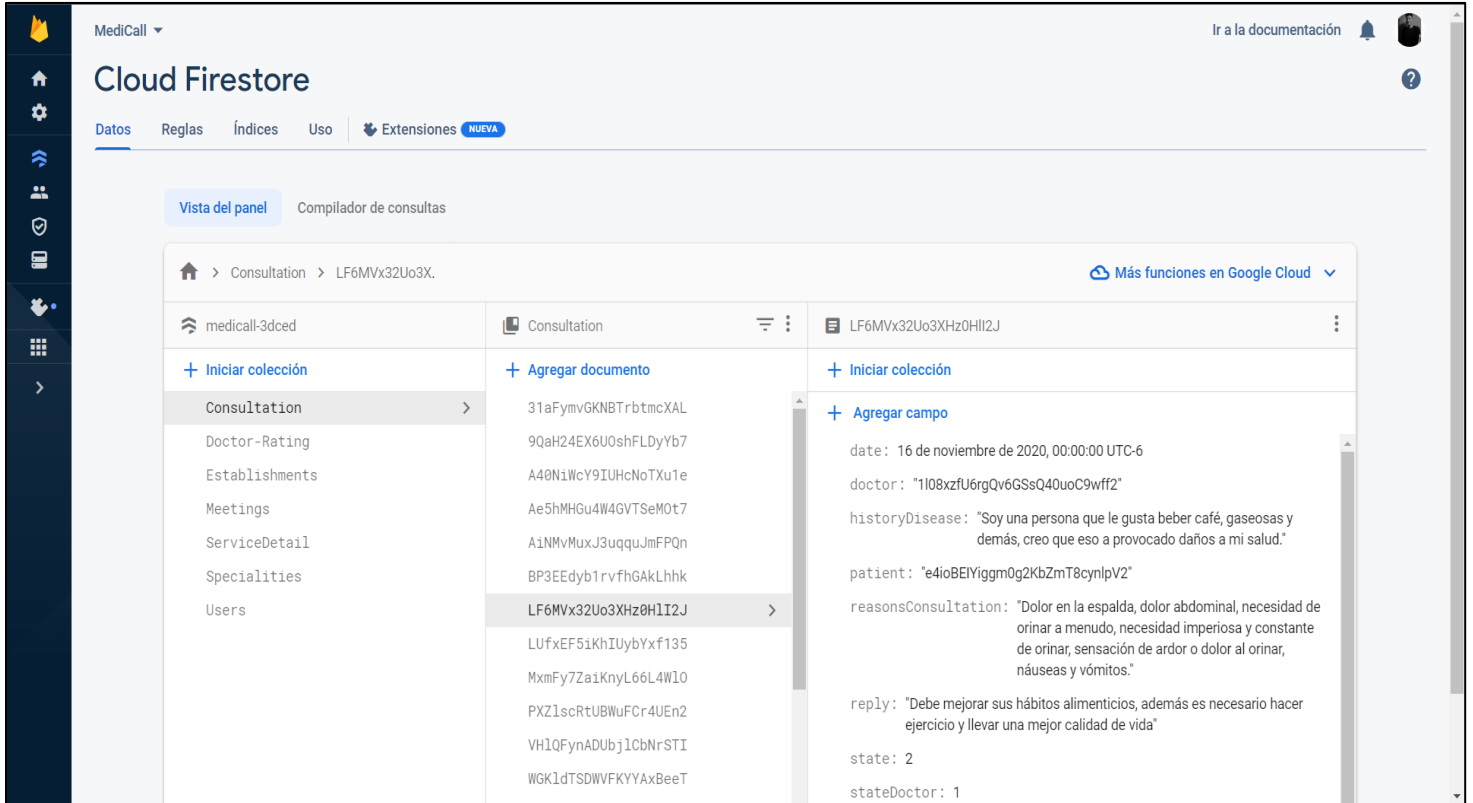


Ilustración 27. Base de datos en Cloud Firestore

Fuente: Elaboración propia

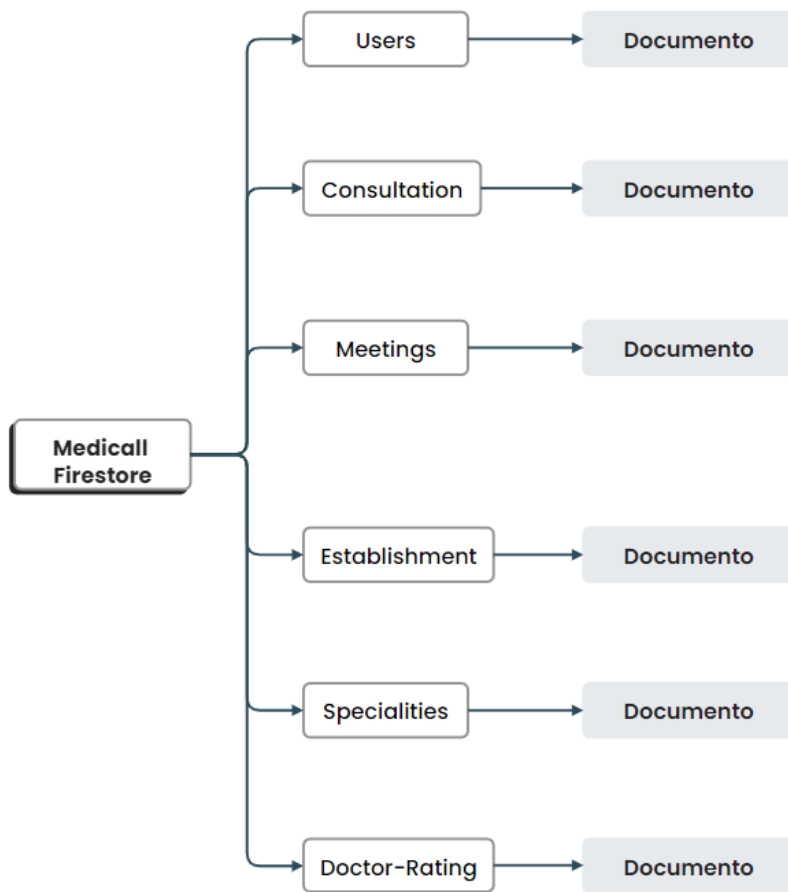


Ilustración 28. Estructura de datos en Firestore

Fuente: Elaboración propia

XIII. Manual de usuario



Manual de Usuario

Medicall

Índice del manual de usuario

I.	Resumen	1
II.	Paciente	1
2.1.	Crear cuenta	1
2.2.	Iniciar sesión	3
2.3.	Buscar médico	4
2.4.	Agendar cita médica	9
2.5.	Realizar consulta médica	12
2.6.	Ver listado de citas agendadas	14
2.7.	Ver listado de consultas solicitadas	14
2.8.	Cerrar sesión	15
III.	Médico	15
3.1.	Crear cuenta	15
3.2.	Iniciar sesión	16
3.3.	Ver listado de citas recibidas	17
3.4.	Ver listados de consultas recibidas	17

Ilustraciones del manual de usuario

Ilustración 1. Home	1
Ilustración 2. Home con panel abierto	2
Ilustración 3. Registro	3
Ilustración 4. Iniciar sesión	4
Ilustración 5. Médicos	5
Ilustración 6. Especialidades	6
Ilustración 7. Establecimientos	7
Ilustración 8. Mejor calificación	8
Ilustración 9. Búsqueda directa	9
Ilustración 10. Listado de médicos	10
Ilustración 11. Perfil del médico	11
Ilustración 12. Agendar cita	11
Ilustración 13. Listado de médicos	12
Ilustración 14. Realizar consulta	13
Ilustración 15. Listado de citas	14
Ilustración 16. Listado de consultas	15
Ilustración 17. Cerrar sesión	15
Ilustración 18. Registro médico	16
Ilustración 19. Iniciar sesión	16
Ilustración 20. Listado de citas como médico	17

1. Resumen

Medicall es una aplicación web intuitiva que cuenta con dos tipos de usuarios: paciente y médico.

En este manual se explica de manera sencilla los pasos que debe hacer un usuario para la consecución de las funcionalidades desarrolladas para cada tipo de usuario.

2. Paciente

2.1. Crear cuenta

Estando en el home de la aplicación, dar click en el menú hamburguesa señalado para abrir el panel de opciones.

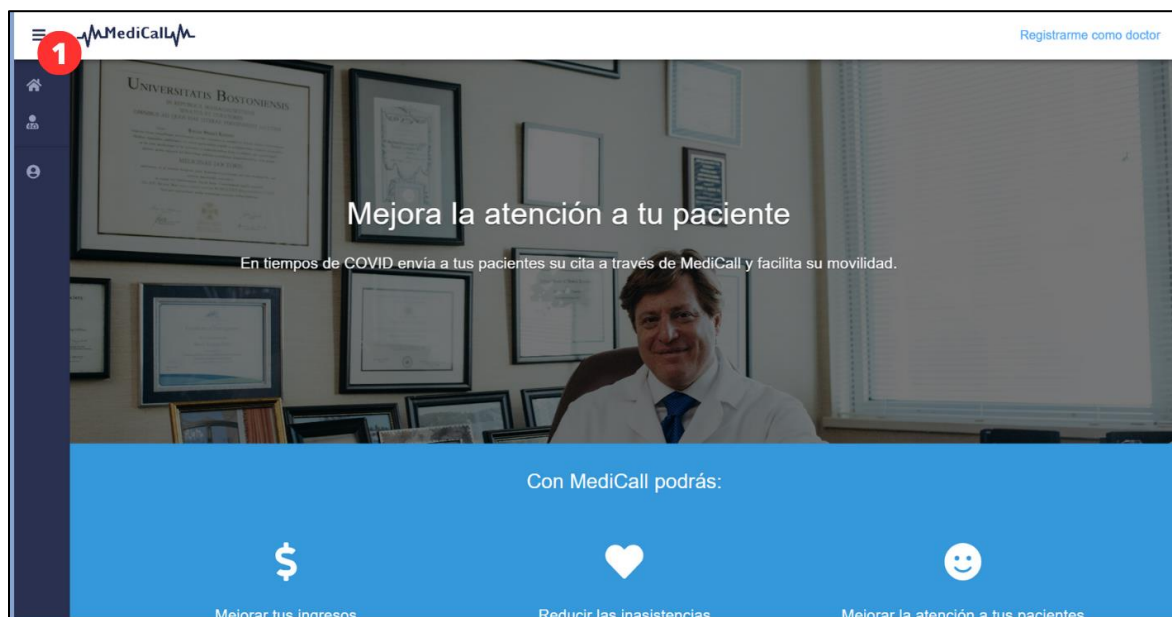


Ilustración 29. Home

Una vez abierto el panel, expandir el apartado cuenta y dar click en Registrarse, esto lleva a la página de registros de usuario como Paciente.



Ilustración 30. Home con panel abierto

Ingresar información necesaria y registrar cuenta.

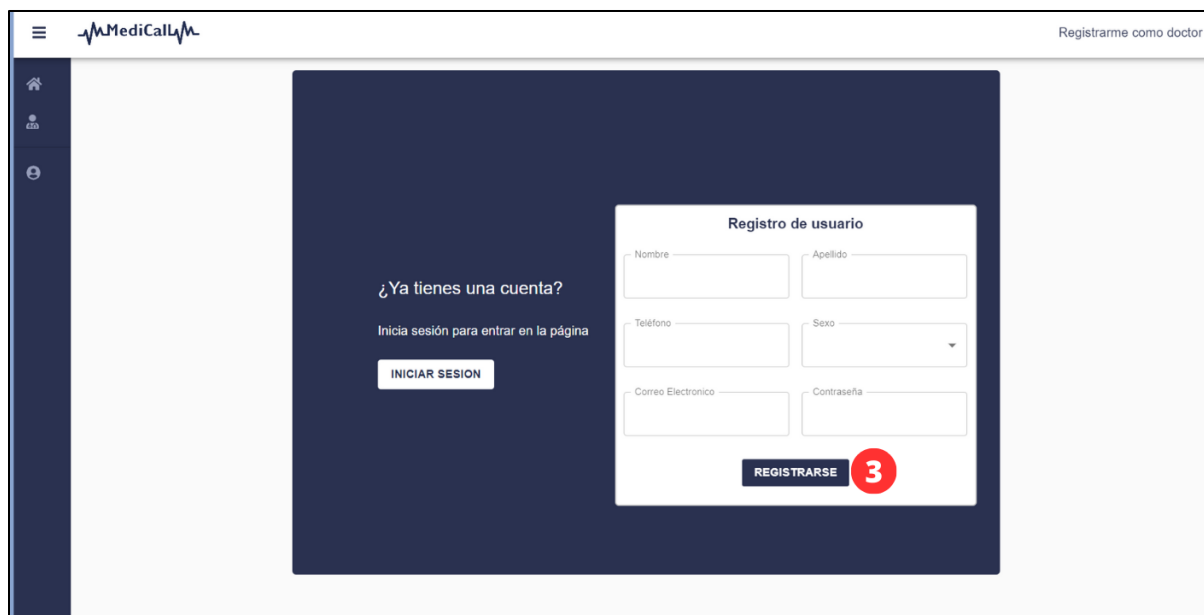


Ilustración 31. Registro

2.2. Iniciar sesión

Para iniciar sesión, abrir el panel de opciones, expandir apartado cuenta y dar click en iniciar sesión, completar información requerida e iniciar sesión.

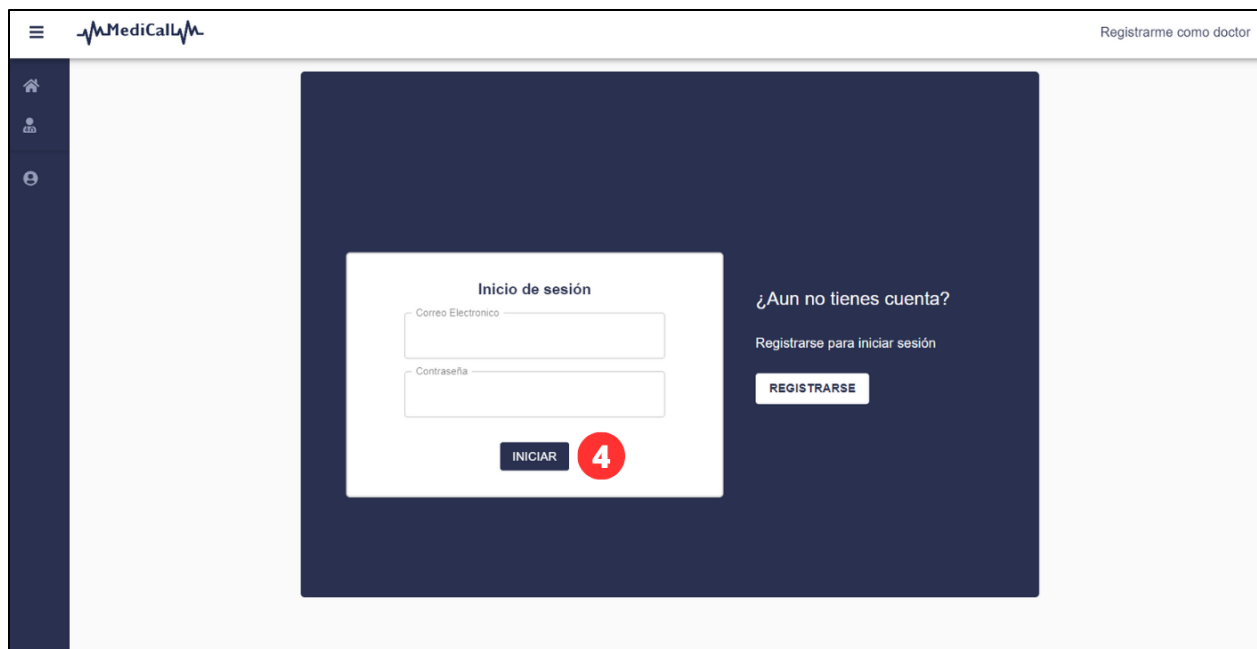


Ilustración 32. Iniciar sesión

2.3. Buscar médico

Con la sesión iniciada, abrir panel de opciones y dar click en el apartado Médicos, esto lleva a la vista de filtros de médicos, seleccionar un filtro disponible.

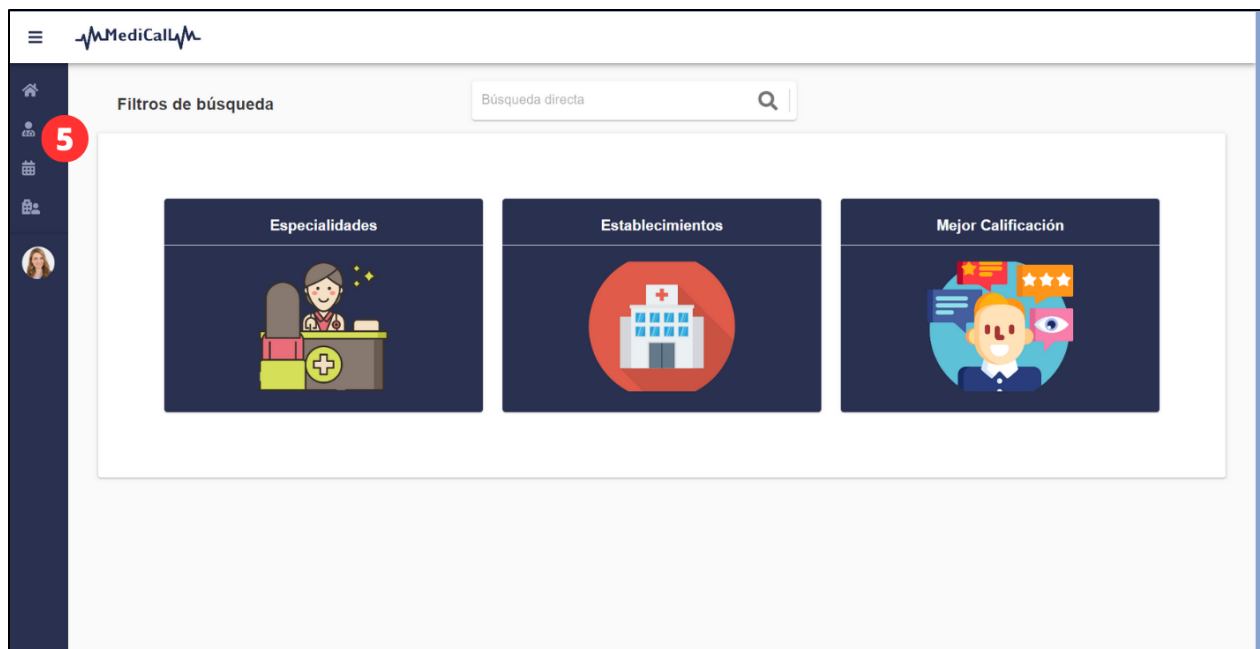


Ilustración 33. Médicos

Al seleccionar el filtro de especialidades, se obtendrá el listado de especialidades disponibles dentro de la app, seleccionar una.

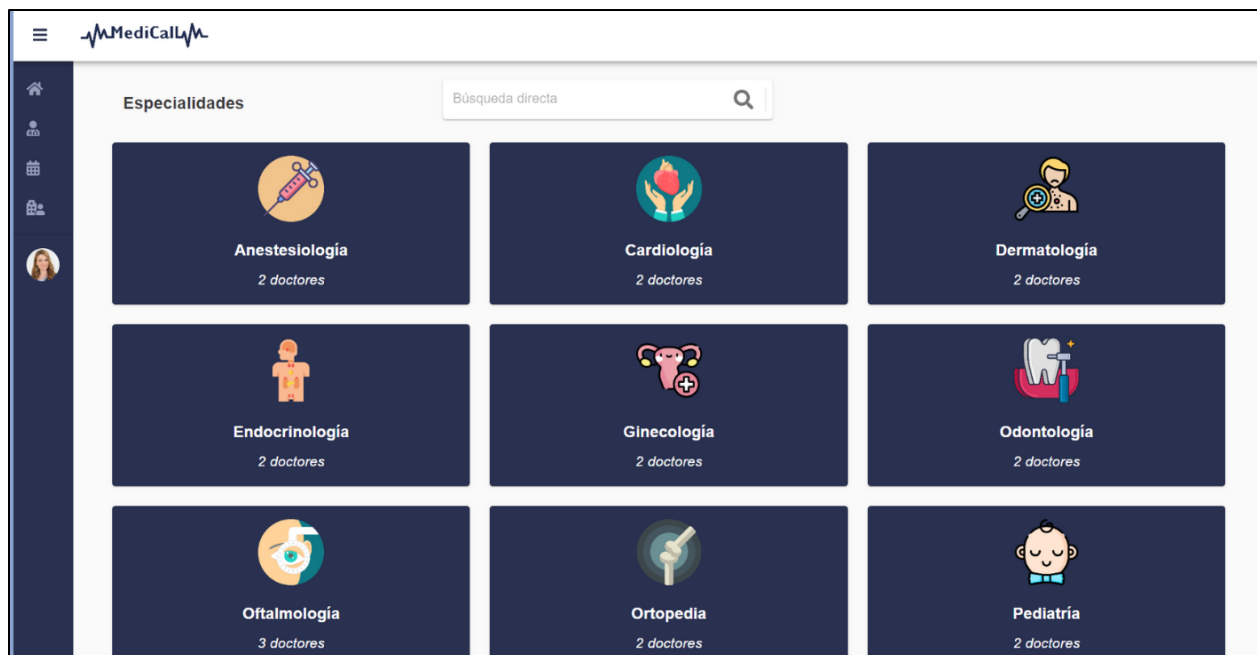


Ilustración 34. Especialidades

Al seleccionar el filtro de establecimientos, se obtendrá el listado de establecimientos disponibles dentro de la app, seleccionar uno.

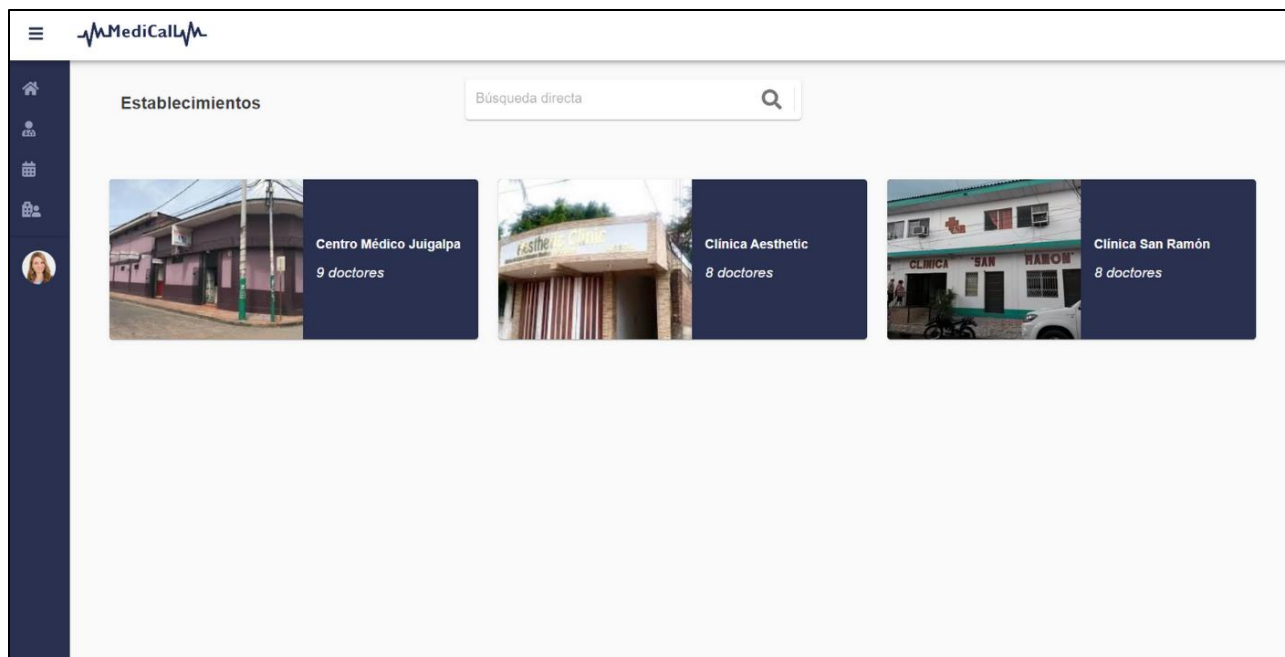


Ilustración 35. Establecimientos

Al seleccionar el filtro de mejor calificación, se obtendrá el listado de médicos con la mejor calificación recibida por parte de otros pacientes que hayan recibido servicios de dicho médico dentro de la app.

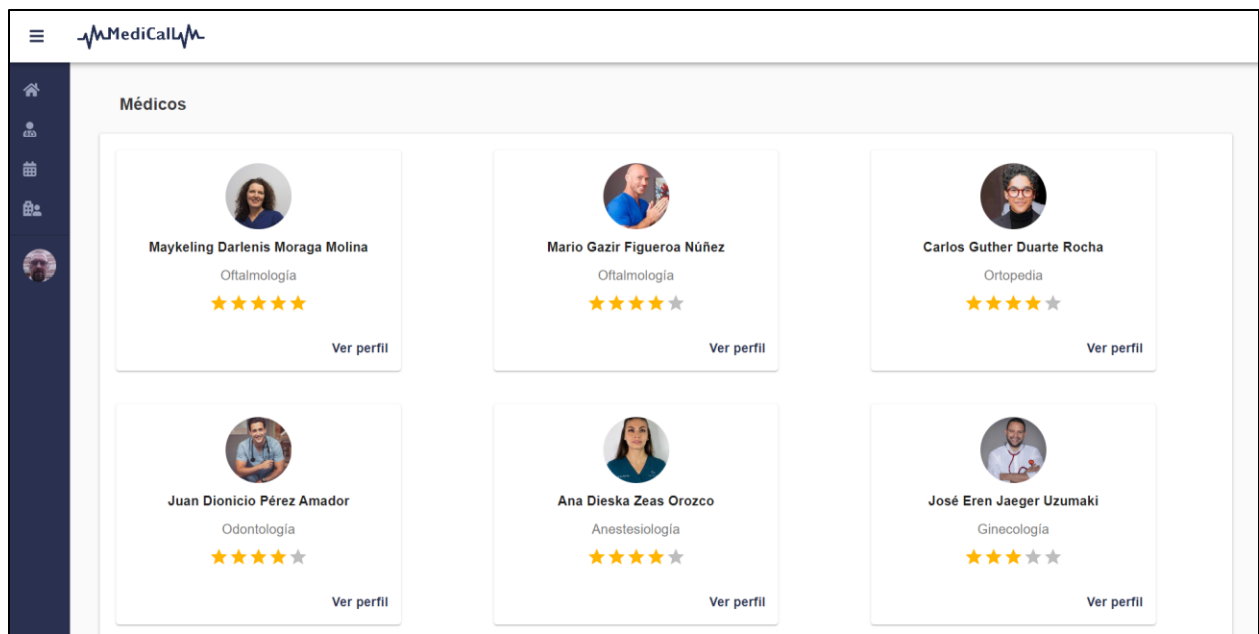


Ilustración 36. Mejor calificación

Al seleccionar el filtro de búsqueda directa, se obtendrá el listado de médicos disponibles dentro de la app filtrados por los términos ingresados.

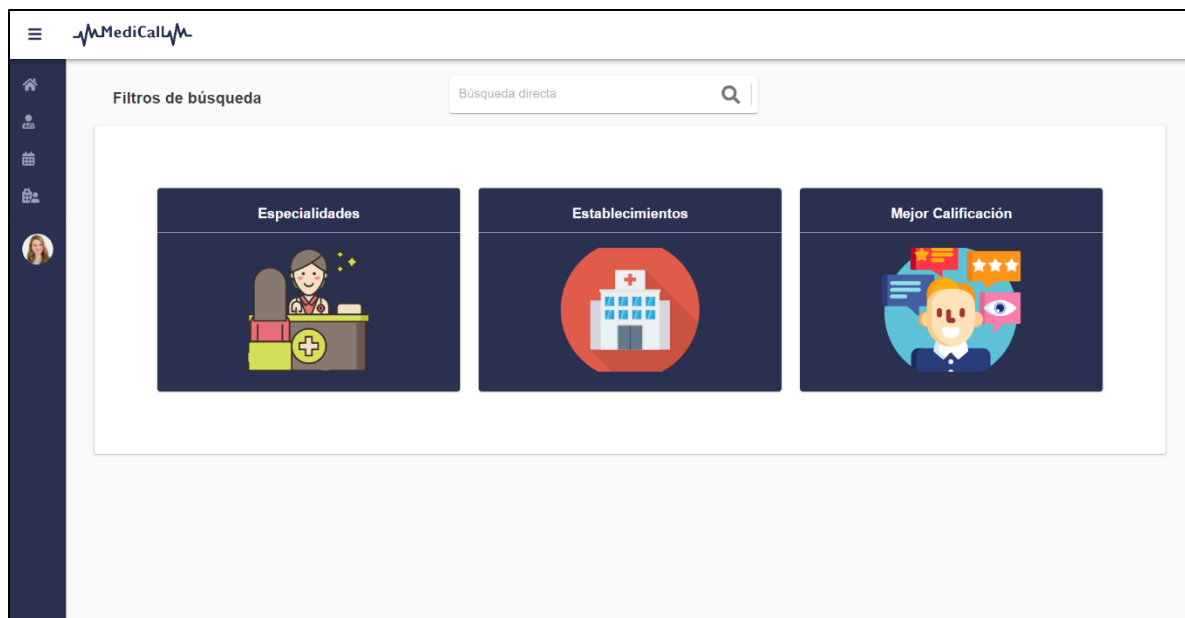


Ilustración 37. Búsqueda directa

2.4. Agendar cita médica

Para agendar una cita, se debe seleccionar un médico mediante cualquier filtro brindado por la aplicación hasta llegar al listado de médicos y dar click en ver perfil al médico de preferencia.

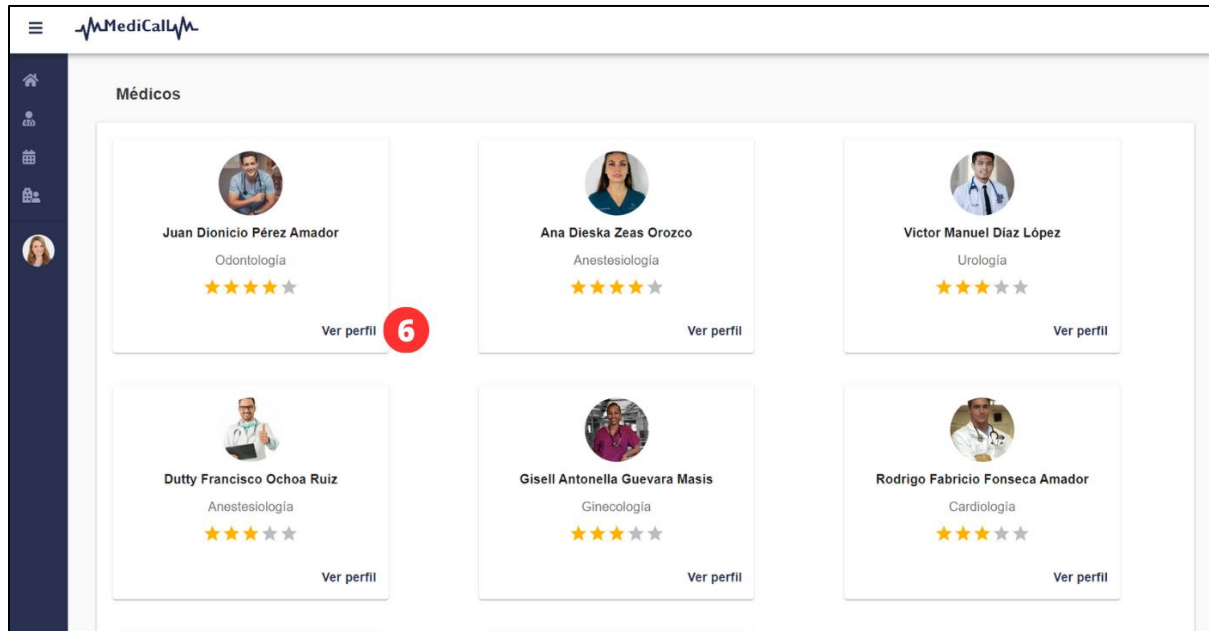


Ilustración 38. Listado de médicos

Se despliega una ventana emergente con información acerca del médico y una opción de agendar cita.

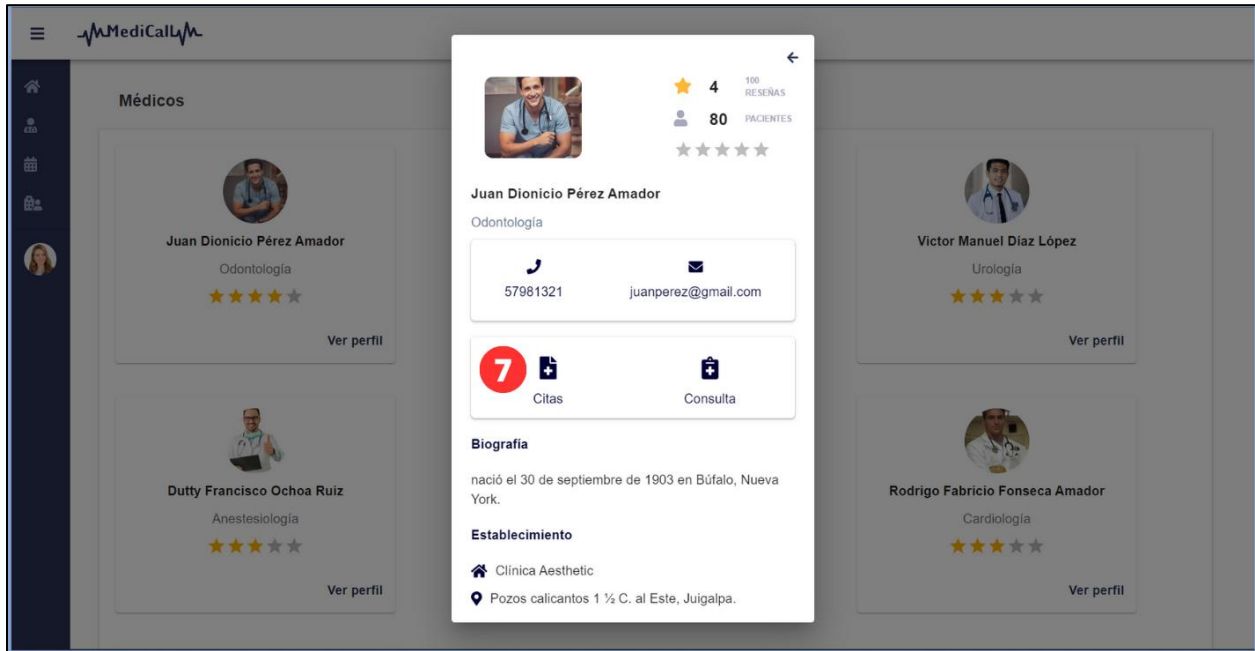


Ilustración 39. Perfil del médico

Esto lleva a la vista para agendar cita, completar campos y agendar.

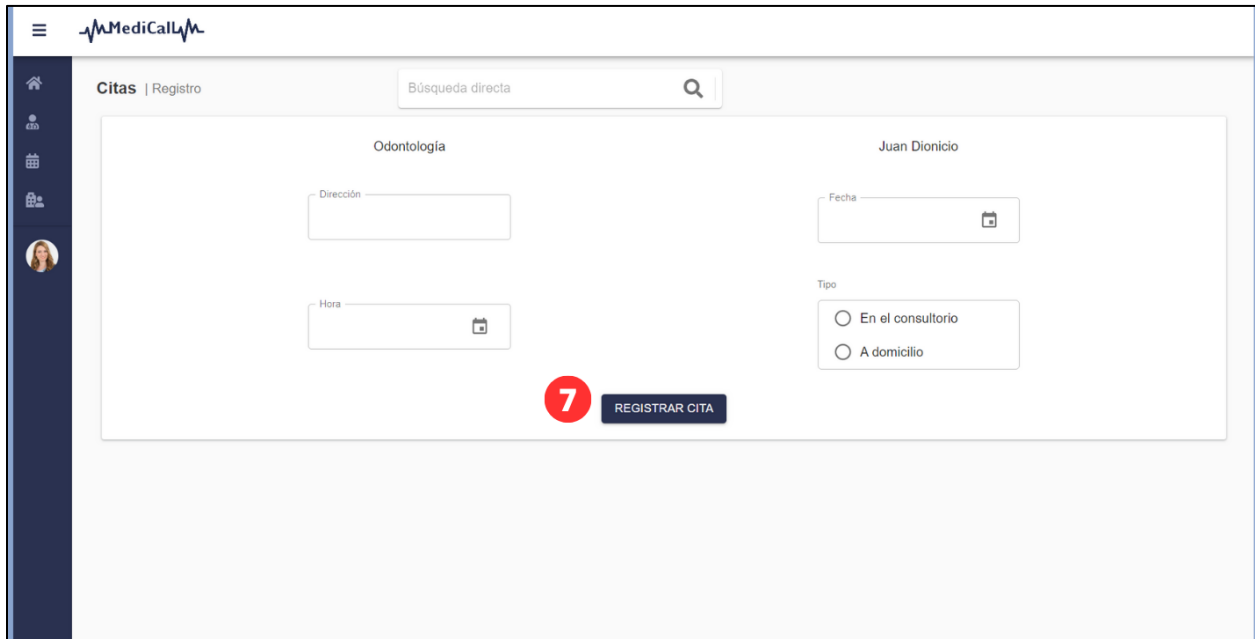


Ilustración 40. Agendar cita

2.5. Realizar consulta médica

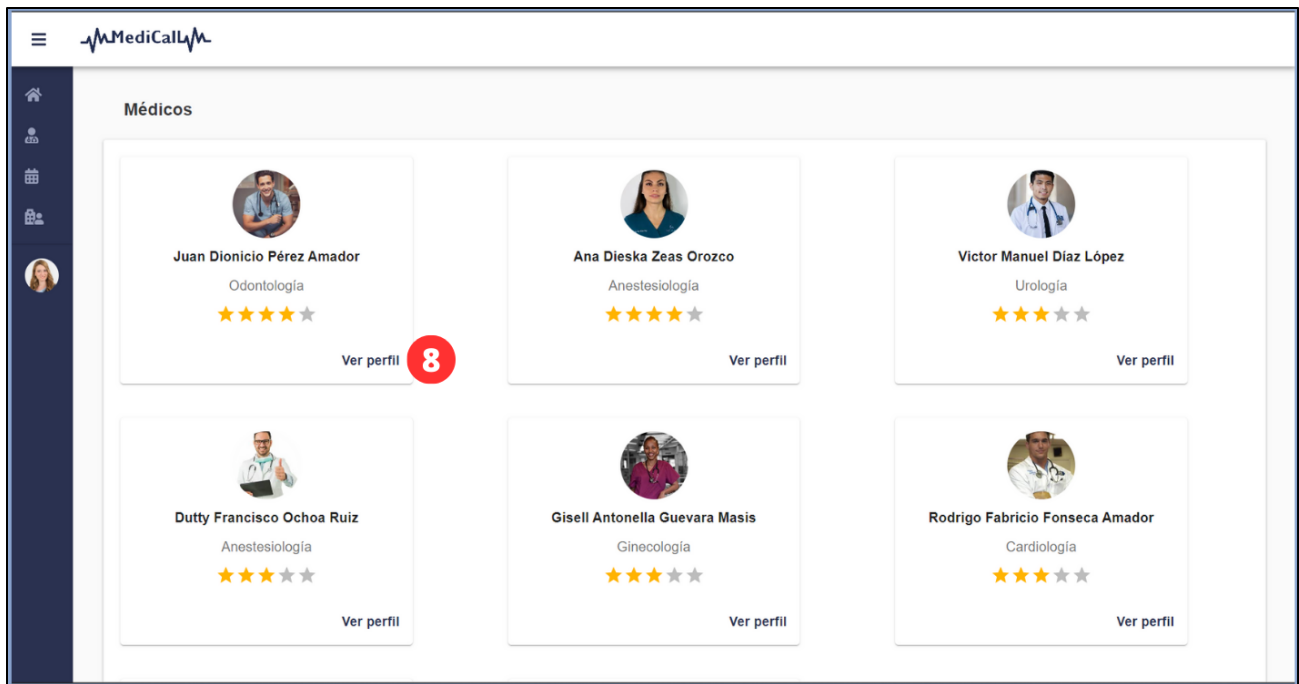


Ilustración 41. Listado de médicos

Para realizar una consulta, se debe seleccionar un médico mediante cualquier filtro brindado por la aplicación hasta llegar al listado de médicos y dar click en ver perfil al médico de preferencia.

Se despliega una ventana emergente con información acerca del médico y una opción de agendar cita.

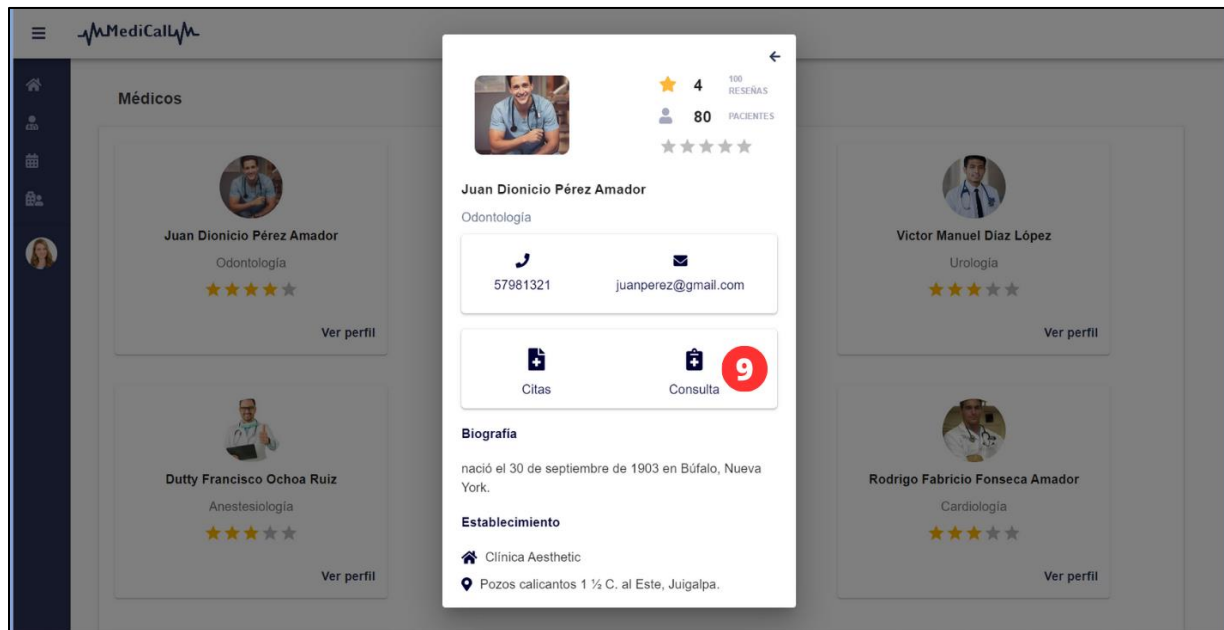


Ilustración 14. Perfil del médico

Esto lleva a la vista para solicitar una consulta, completar campos y solicitar.

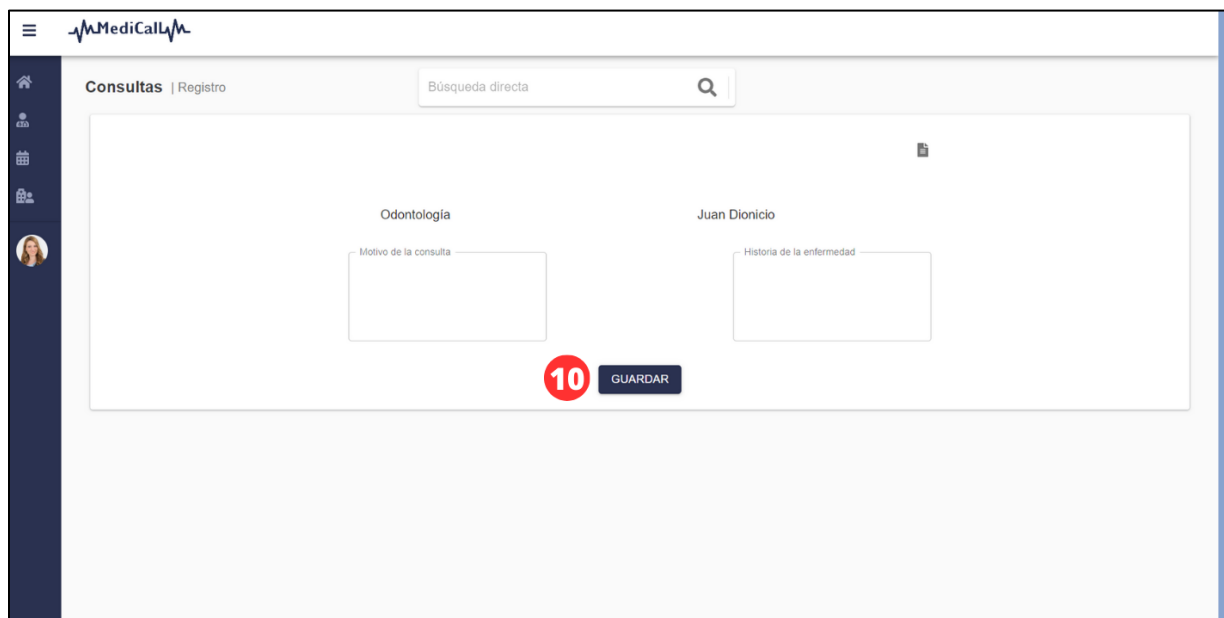


Ilustración 42. Realizar consulta

2.6. Ver listado de citas agendadas

Abrir panel de opciones y dar click en el apartado de citas para ver el listado de citas agendadas.

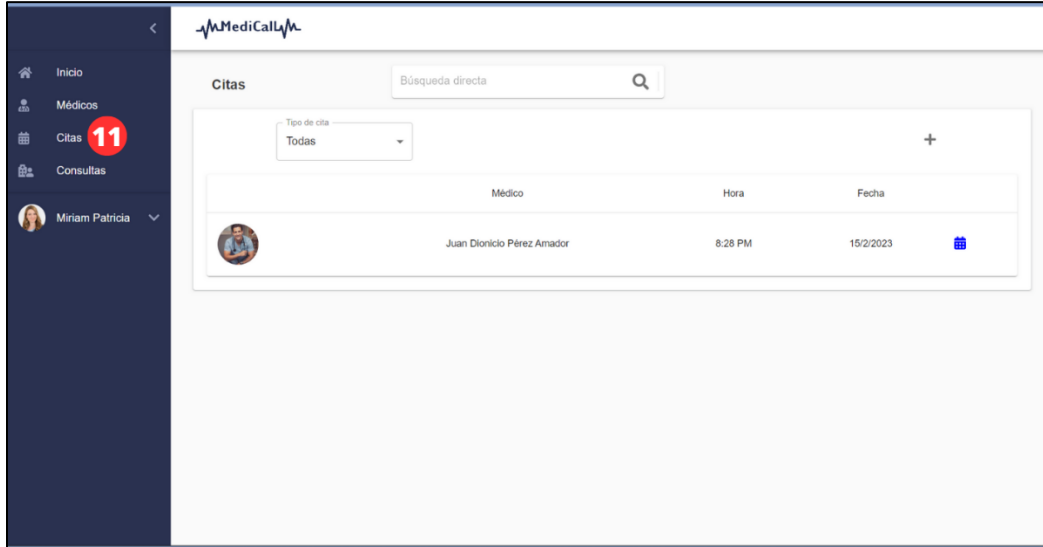


Ilustración 43. Listado de citas

2.7. Ver listado de consultas solicitadas

Abrir panel de opciones y dar click en el apartado de consultas para ver el listado de consultas realizadas.

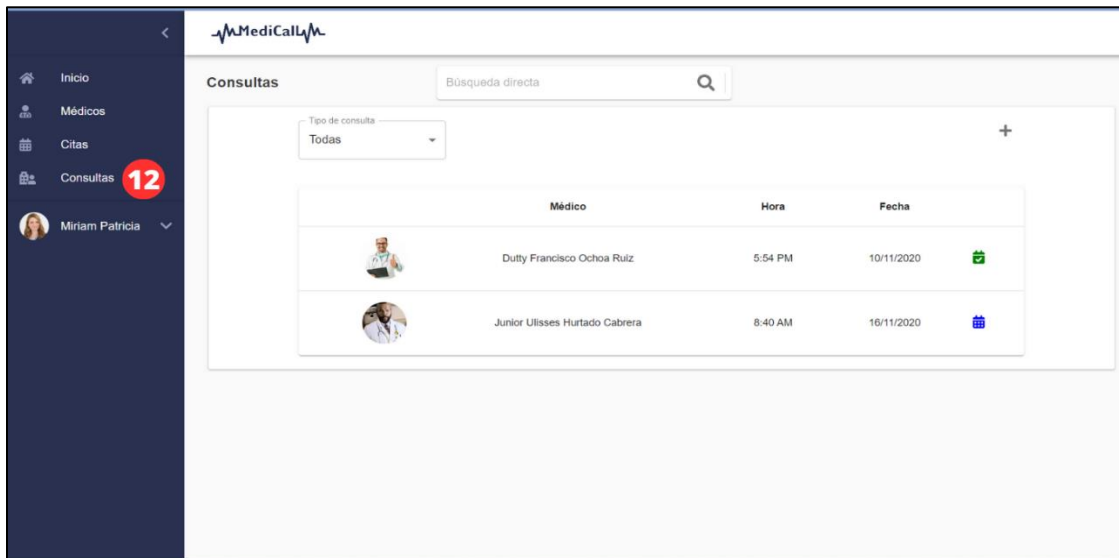


Ilustración 44. Listado de consultas

2.8. Cerrar sesión

Abrir panel de opciones y dar click en el apartado en perfil para expandir opciones de cuenta y dar click en cerrar sesión.

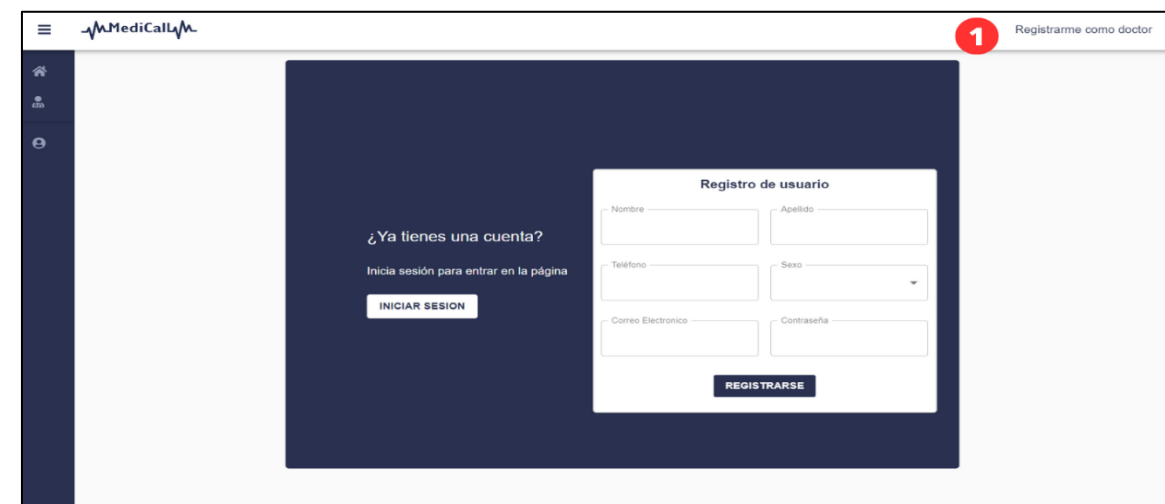


Ilustración 45. Cerrar sesión

3. Médico

3.1. Crear cuenta

Para crear cuenta cómo médico, dar click en botón ubicado en la parte superior derecha, completa campos requeridos y registrarse.



3.2. Iniciar sesión

Para iniciar sesión, abrir el panel de opciones, expandir apartado cuenta y dar click en iniciar sesión, completar información requerida e iniciar sesión.

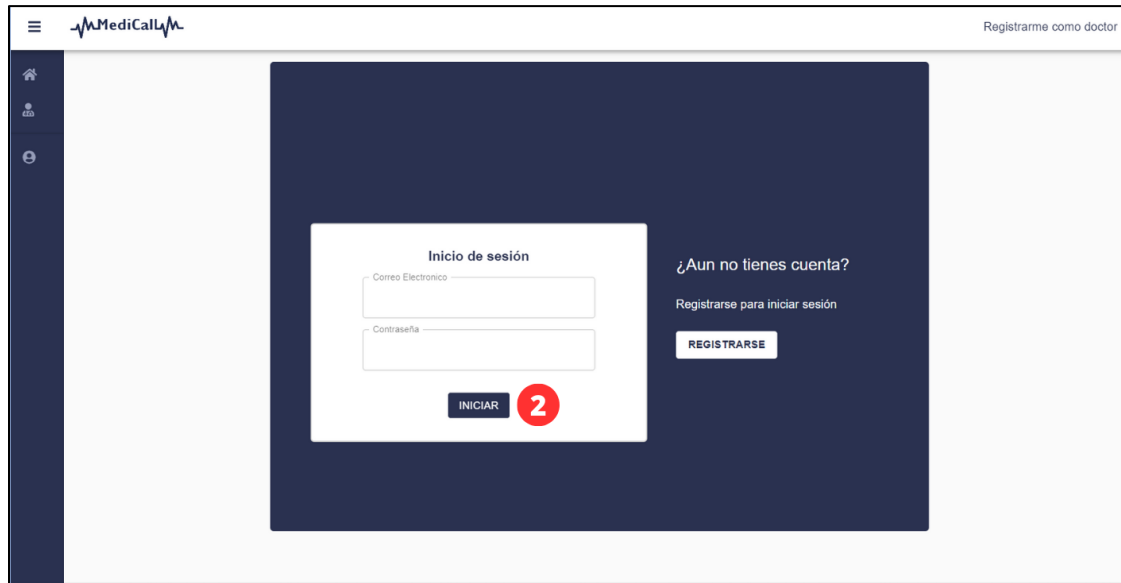


Ilustración 47. Iniciar sesión

3.3. Ver listado de citas recibidas

Para ver las citas recibidas, abrir panel de opciones y dar click en apartado de citas, verá el listado de las citas que ha recibido

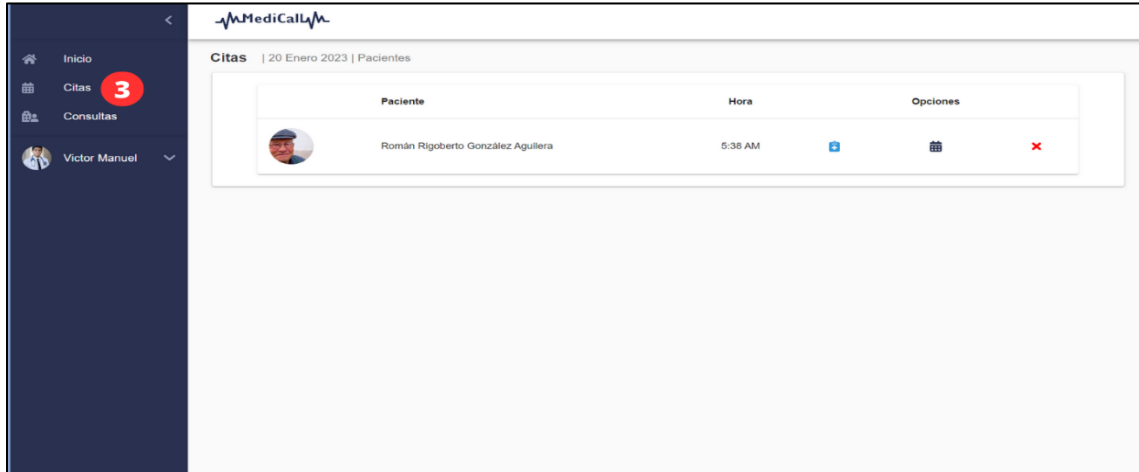


Ilustración 48. Listado de citas como médico

3.4. Ver listados de consultas recibidas

Para ver las consultas recibidas, abrir panel de opciones y dar click en apartado de consultas, verá el listado de las consultas que ha recibido

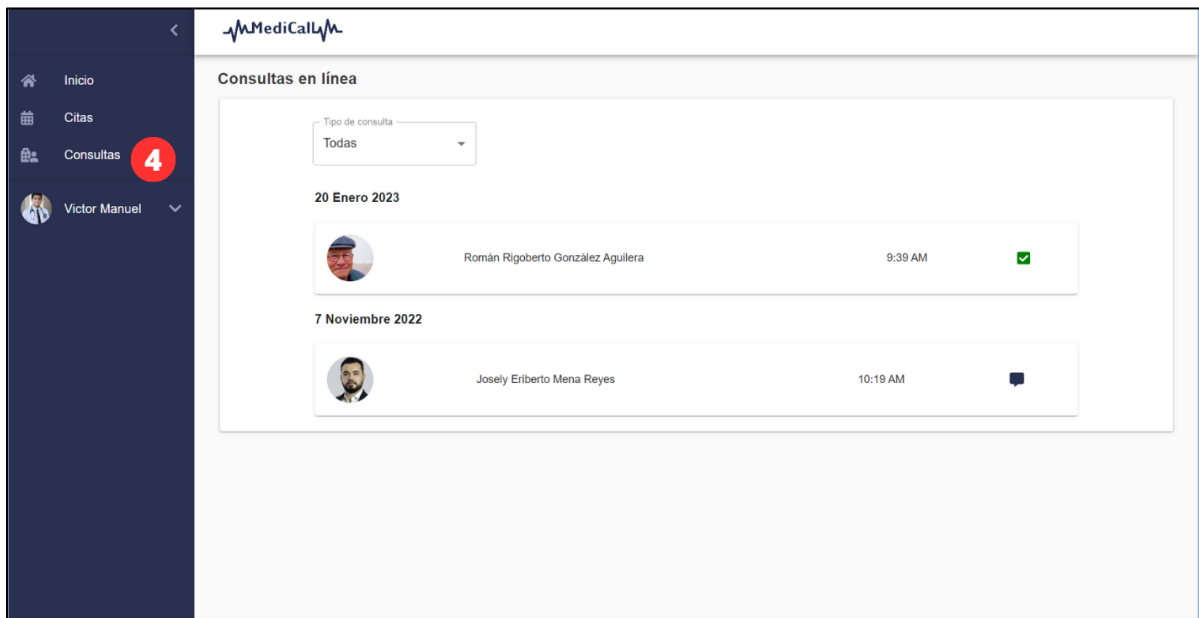


Ilustración 21. Listado de consultas como médico

XIV. Manual técnico



Manual Técnico

Medicall

Tabla de contenido

IV. Resumen.....	1
V. Tecnologías utilizadas.....	2
VI. Requerimientos de hardware.....	5
3.1. Computadora.....	6
3.2. Dispositivo móvil.....	6
VII. Arquitectura Front-End.....	7
VIII. Estructura del Front-End.....	8

Ilustraciones del manual técnico

Ilustración 1. Logo de React.....	2
Ilustración 2. Logo de MUI.....	3
Ilustración 3. Logo de Redux.....	4
Ilustración 4. Logo de Firebase	5
Ilustración 5. Logo de Flux	7
Ilustración 6. Estructura del Front-End.....	8

Tablas del manual técnico

Tabla 1. React	2
Tabla 2. MUI	3
Tabla 3. Redux	4
Tabla 4. Firebase	5
Tabla 5. Requerimientos de hardware Computadora	6
Tabla 6. Requerimientos de hardware Dispositivo móvil	6

1. Resumen

Medicall está diseñada para ayudar a gestionar los procesos de citas médicas y asistencia previa a pacientes en línea, permitiendo automatizar los procedimientos tradicionales que se utilizan en la actualidad.

Medicall está desarrollada utilizando la biblioteca de JavaScript llamada React para la construcción de las interfaces de usuario acompañado del framework MUI (antes Material UI) para las hojas de estilo, incluyendo la estructura de HTML5 y CSS3. También, esta utiliza la arquitectura Flux para el manejo y flujo de datos mediante Redux, tiene conexión a una base de datos en Firebase.

En este manual se explica de manera sencilla en que tecnologías (lenguajes de programación, framework, base de datos) está construido el software, que arquitectura utiliza y cómo es su lógica interna.

2. Tecnologías utilizadas

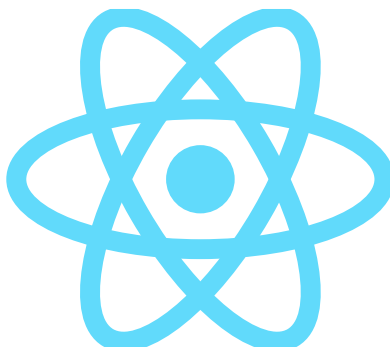


Ilustración 49. Logo de React

Fuente: por <https://es.reactjs.org/>

“React (también llamada React.js o ReactJS) es una biblioteca JavaScript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página”. (React, 2022)

Descripción	Versión	Licencia
Para el Front-End se utilizó React para el desarrollo de las interfaces de usuario, facilitando el uso de datos que cambian constantemente, ya que sus componentes responden ante la variación de esos datos.	V17.0.1	MIT

Tabla 29. React

Fuente: Elaboración propia

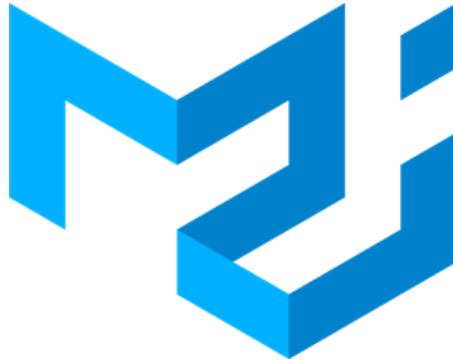


Ilustración 50. Logo de MUI

Fuente: por <https://mui.com/>

MUI es un framework de diseño desarrollado para React y que ofrece un conjunto integral de herramientas de interfaz de usuario que ayudan a enviar funciones con mayor rapidez.

Descripción	Versión	Licencia
Para el Front-End se utilizó MUI para el diseño de las interfaces de usuario, facilitando el uso de herramientas altamente compatibles con React.	V4.11	MIT

Tabla 30. MUI

Fuente: Elaboración propia



Ilustración 51. Logo de Redux

Fuente: por <https://redux.js.org/>

Redux es una librería JavaScript de código abierto para el manejo del estado de las aplicaciones. Es comúnmente usada con otras tecnologías como React o Angular para la construcción de interfaces de usuario.

Descripción	Versión	Licencia
Para el manejo del estado de la aplicación se utilizó Redux,	V4.0.5	MIT

Tabla 31. Redux



Firebase

Ilustración 52. Logo de Firebase

Fuente: por <https://firebase.google.com/>

Firebase es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad.

Descripción	Versión	Licencia
Para base de datos se utilizó Firebase, específicamente el servicio Cloud Firestore que brinda múltiples beneficios, entre los que destaca el almacenamiento en la nube.	V8.1.1	MIT

Tabla 32. Firebase

3. Requerimientos de hardware

En la siguiente tabla de enumeran los requisitos de hardware mínimos y recomendados para hacer uso de la aplicación web.

3.1. Computadora

Componente	Mínimo	Recomendado
Procesador	Procesador de x86 o x64 bits de doble núcleo de 1,9 gigahercios (GHz) o más, con el conjunto de instrucciones SSE2	Procesador de x64 bits de doble núcleo de 3,3 gigahercios (GHz) o más, con el conjunto de instrucciones SSE2
Memoria RAM	2 GB	4GB o más

Tabla 33. Requerimientos de hardware Computadora

Fuente: Elaboración propia

3.2. Dispositivo móvil

Componente	Mínimo	Recomendado
Procesador	Snapdragon QSD8650	Snapdragon 410
Memoria RAM	2 GB	4GB o más

Tabla 34. Requerimientos de hardware Dispositivo móvil

Fuente: Elaboración propia

4. Arquitectura Front-End



Ilustración 53. Logo de Flux

Fuente: por <https://facebook.github.io/flux>

Flux es la arquitectura de aplicaciones que utiliza Facebook para crear aplicaciones web del lado del cliente. Complementa los componentes de vista componible de React al utilizar un flujo de datos unidireccional.

5. Estructura del Front-End

A continuación, se presenta la estructura del Front-End implementando la arquitectura

Flux:

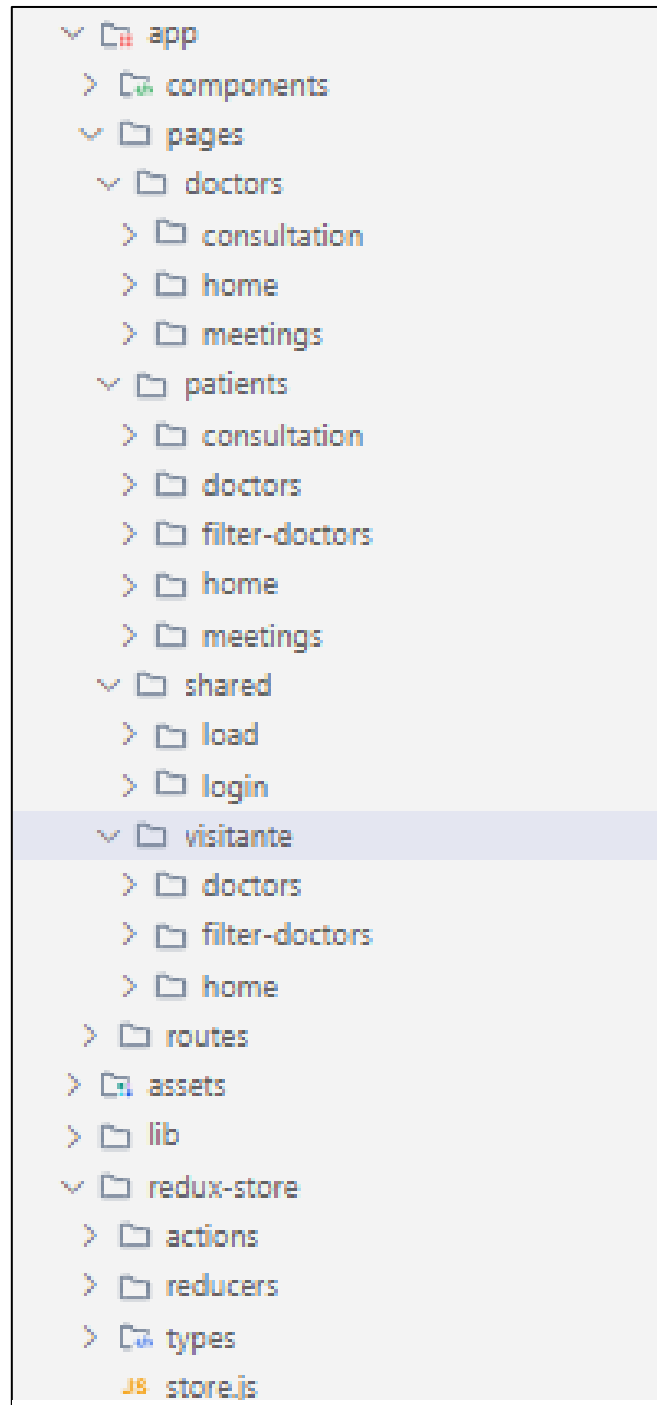


Ilustración 54. Estructura del Front-End

A continuación, se explica el componente de Especialidades debido a la importancia que este tiene dentro de la aplicación. Cabe recalcar que cada componente cuenta con tres archivos para realizar la segregación de lógica, estilos y vista:

La lógica de este componente está determinada por la petición a Firebase para obtener la información acerca de las especialidades registradas en la aplicación.

```
1 import React from 'react';
2 import { withRouter } from 'react-router-dom';
3 import SpecialitiesView from './specialities.view';
4 import { getSpecialities } from './../../../../lib/data/speciality.data';
5
6
7 class Specialities extends React.Component {
8   constructor(props) {
9     super(props);
10    this.state = {
11      data: []
12    };
13  }
14  goDoctors = (id) => {
15    const { history } = this.props;
16    history.push({pathname: "/doctors", state: {id: id}});
17  }
18
19  render() {
20    const { data } = this.state;
21    return (
22      <SpecialitiesView
23        goDoctors={this.goDoctors}
24        data={data}
25      />
26    );
27  }
28  async componentDidMount() {
29    const specialities = await getSpecialities();
30    this.setState({data: specialities});
31  }
32 }
33
34 export default withRouter(Specialities);
```

Ilustración 55. Lógica del componente Specialities

Los estilos se aplican de la mano de Material UI utilizando una técnica llamada Style Components, se decidió separar dichos estilos de la vista para mejorar la lectura de la vista.

```
1 import { makeStyles } from '@material-ui/core/styles';
2
3 const specialitiesStyles = makeStyles(theme => ({
4   container: {
5     display: "flex",
6     flexDirection: "column",
7     alignItems: "center",
8     justifyContent: "right",
9     marginBotton: 20
10  },
11  cardContainer: {
12    width: "95%",
13    display: "flex",
14    flexDirection: "row",
15    justifyContent: "center",
16    backgroundColor: "white",
17    boxShadow: "rgba(60, 64, 67, 0.3) 0px 1px 2px 0px, rgba(60, 64, 67, 0.15) 0px 1px 3px 1px",
18    borderRadius: 3,
19  },
20  title: {
21    fontSize: 20,
22    marginLeft: 30,
23    marginTop: 20,
24    marginBottom: 10,
25    fontWeight: "bold",
26    color: "„39393b",
27  },
28  titleContainer: {
29    display: 'flex',
30    alignItems: 'center',
31    marginLeft: 30,
32    marginTop: 80,
33    marginBottom: 10
34  },
35  centerCard: {
36    width: "90%",
37    marginTop: 30,
38    marginBottom: 30
39  },
40  specialityCard: {
41    backgroundColor: '#31beff',
42  },
43  cardAction: {
44    display: 'flex'
45  },
46  image: {
47    width: 80,
48    height: 80,
49    marginBottom: 20
50  },
51  cardContent: {
52    display: 'flex',
53    flexDirection: 'column',
54    alignItems: 'center',
55  },
56  quantity: {
57    marginTop: 10,
58    fontStyle: 'italic',
59    fontSize: 17,
60    color: 'white'
61  },
62  name: {
63    color: 'white',
64    fontWeight: 'bold',
65  },
66  },
67  },
68  },
69  },
70  },
71  },
72  },
73  },
74  },
75  },
76  },
77  },
78  },
79  },
80  },
81  },
82  },
83  },
84  },
85  },
86  },
87  },
88  },
89  },
90  },
91  },
92  },
93  },
94  },
95  },
96  },
97  },
98  },
99  },
100 },
101 });
```

Ilustración 56. Estilos del componente Specialities

La vista está construida utilizando componentes de la librería de Material UI y se encarga de utilizar la información recibida por la lógica.

```
1 import React from 'react';
2 import Layout from '../shared/theme/layout';
3 import Search from '../search/search'
4 import {
5   Grid,
6   Typography,
7   Avatar,
8   Card,
9   CardActionArea,
10  CardContent,
11 } from '@material-ui/core';
12 import { specialitiesStyles } from './specialities.styles';
13
14 const SpecialitiesView = (props) => {
15   const classes = specialitiesStyles ();
16   const { goDoctors, data } = props;
17
18   return (
19     <Layout>
20       <div>
21         <div className={classes.titleContainer}>
22           <Typography className={classes.title}>Especialidades</Typography>
23           <Search/>
24         </div>
25
26         <Grid spacing={3} container >
27           {
28             data && data.map(item => (
29               <Grid item lg = {4}>
30                 <Card className={classes.specialityCard}>
31                   <CardActionArea onClick={() => goDoctors(item)}>
32                     <CardContent className={classes.cardContent}>
33                       <Avatar className={classes.image} variant="square" src={item.image}/>
34                       <Typography className={classes.name}>{item.name}</Typography>
35                       <Typography className={classes.quantity}>
36                         {`${item.doctorsQuantity} doctores`}
37                       </Typography>
38                     </CardContent>
39                   </CardActionArea>
40                 </Card>
41               </Grid>
42             ))
43           }
44         </Grid>
45       </div>
46     </Layout>
47   );
48 }
49
50 export default SpecialitiesView;
51
```

Ilustración 57. Vista del componente *Specialities*

xv. Plan de Pruebas



PLAN DE PRUEBAS APLICACIÓN PARA CITAS Y CONSULTAS MÉDICAS EN LÍNEA

Juigalpa, Chontales, noviembre de 2022

1. Características a ser probadas

CARACTERÍSTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Funcionalidad para agendar cita médica	Se gestiona el árbol de estado de la aplicación, que permite agregar nueva cita médica.	<ul style="list-style-type: none">• Componente Redux store• Servicio Firebase• Componente Citas	P01
Funcionalidad para administrar citas	Se obtendrán las citas correspondientes al usuario actual.	<ul style="list-style-type: none">• Componente Redux store• Componente ListaCitas	P02

1.1. Características a ser probadas Sprint 1

1.2. Características a ser probadas Sprint 2

1.3. Características a ser probadas Sprint 3

CARACTERÍSTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Funcionalidad para registrar un paciente	Se gestiona el árbol de estado de la aplicación, que permite agregar nuevo usuario de la aplicación	<ul style="list-style-type: none"> • Componente Redux store • Servicio de Firebase • Componente Register 	P05

1.4. Características a ser probadas Sprint 4

CARACTERÍSTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Funcionalidad para iniciar sesión	Se gestionan los datos ingresados para validar la presencia de un usuario registrado con dichos datos.	<ul style="list-style-type: none"> • Componente Redux store • Servicio Firebase • Componente Login 	P05
Funcionalidad para realizar una consulta médica	Se gestiona el árbol de estado de la aplicación, que permite agregar nueva consulta médica.	<ul style="list-style-type: none"> • Componente Reduxstore • Servicio Firebase • Componente Consultas 	P03
Funcionalidad para administrar consultas	Se mostrarán las consultas correspondientes al usuario actual.	<ul style="list-style-type: none"> • Componente Redux store • Componente ListaCitas 	P04

CARACTERISTICA	DESCRIPCIÓN	JUSTIFICACION	RIESGO
Gestión de citas	Permite realizar acciones sobre cada cita como: reprogramar o cancelar	Esta funcionalidad tiene relación de dependencia con el servicio Cloud Firestore de Firebase	Impedir realizar acciones sobre cada cita

CARACTERISTICA	DESCRIPCIÓN	JUSTIFICACION	RIESGO
Gestión de consultas	Permite realizar acciones sobre cada consulta como: cancelar	Esta funcionalidad tiene relación de dependencia con el servicio Cloud Firestore de Firebase	Impedir realizar acciones sobre cada consulta

2. Características a no ser probadas

2.1. Características a no ser probadas Sprint 1

2.2. Características a no ser probadas Sprint 2

CARACTERISTICA	DESCRIPCIÓN	JUSTIFICACION	RIESGO
Selección de médico	Permite seleccionar que médico lo atenderá.	Esta funcionalidad tiene relación de dependencia con el servicio Cloud Firestore de Firebase	Impedir la visualización de especialidades, establecimientos y médicos registrados
Registro de médico	Permite registrar nuevo usuario de tipo médico	Esta funcionalidad tiene similitud con el registro de pacientes.	Impedir el registro de médicos.

2.3. Características a no ser probadas Sprint 3

CARACTERISTICA	DESCRIPCIÓN	JUSTIFICACION	RIESGO
Notificación recordatoria	Permite enviar notificación recordatoria a la aproximación de una cita agendada.	Esta funcionalidad tiene relación de dependencia con el servicio Cloud Messaging de Firebase	Impedir la visualización de especialidades, establecimientos y médicos registrados

2.4. Características a no ser probadas Sprint 4

3. Informe de pruebas

3.1. Pruebas Sprint 1

A continuación, se muestran los scripts correspondientes a las pruebas unitarias realizadas sobre la aplicación según el **caso de prueba P01 y P02**.

```
1 import { createMeeting, getMeetingsUser } from "../lib/data/meetings.data";
2
3 describe("PRUEBAS SPRINT 1", () => {
4   test("debería retorna listado de citas", () => {});
5   test("debería retornar el listado de citas con la nueva cita agregada", async () => {
6     const newMeeting = {
7       date: new Date(
8         "Thu Nov 24 2022 08:21:00 GMT-0600 (hora estándar central)"
9       ),
10      direction: "REMASA 2C AL OESTE",
11      doctor: "1l08xzfU6rgQv6GSsQ40uoC9wff2",
12      patient: "phtsagoSetMFP8jAkjRQHLx60jB3",
13      speciality: "Js3iWAzL8UQLS7mhLwFd",
14      state: 2,
15      time: new Date(
16        "Tue Nov 22 2022 13:00:41 GMT-0600 (hora estándar central)"
17      ),
18      type: 2,
19    };
20
21    await createMeeting(newMeeting);
22    const obtainedMeetings = await getMeetingsUser(
23      "phtsagoSetMFP8jAkjRQHLx60jB3"
24    );
25
26    const expectedMeetings = {
27      date: "11/24/2022",
28      doctor: "Junior Ulisses Hurtado Cabrera",
29      id: obtainedMeetings[obtainedMeetings.length - 1].id,
30      image: obtainedMeetings[obtainedMeetings.length - 1].image,
31      patient: "phtsagoSetMFP8jAkjRQHLx60jB3",
32      state: 2,
33      time: "1:00 PM",
34      type: 2,
35    };
36
37    expect(obtainedMeetings[obtainedMeetings.length - 1]).toEqual(
38      expectedMeetings
39    );
40  });
41 });
42
```

Ilustración 58. Script prueba de sprint 1

```
RUNS src/___tests___/register-meeting.test.js
PASS src/___tests___/register-meeting.test.js
PRUEBAS SPRINT 1
  ✓ debería retorna listado de citas (1ms)
  ✓ debería retornar el listado de citas con la nueva cita agregada (1174ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        3.406s
Ran all test suites related to changed files.

Watch Usage
  > Press a to run all tests.
  > Press f to run only failed tests.
  > Press q to quit watch mode.
  > Press t to filter by a test name regex pattern.
  > Press p to filter by a filename regex pattern.
  > Press Enter to trigger a test run.
```

Ilustración 59. Resultados de pruebas Sprint 1

3.2. Pruebas Sprint 2

A continuación, se muestran los scripts correspondientes a las pruebas unitarias realizadas sobre la aplicación según el caso de prueba P03 y P04.

```
1 import {
2   createConsulta,
3   getConsultationsUser,
4 } from "../lib/data/consultation.data";
5
6 describe("PRUEBAS SPRINT 2", () => {
7   test("debería retorna listado de consultas", () => {});
8   test("debería retornar el listado de consultas con la nueva consulta agregada", async () => {
9     const newConsultation = {
10      date: new Date(
11        "Tue Nov 22 2022 11:07:31 GMT-0600 (hora estándar central)"
12      ),
13      doctor: "KmtgLfhFHpejUEHXj6fjDRvF8jA2",
14      historyDisease: "NO HAY",
15      patient: "phtsagoSetMFP8jAkjRQHLx60jB3",
16      reasonsConsultation: "ORINAR CONSTANTEMENTE",
17      state: 2,
18      stateDoctor: 1,
19      time: new Date(
20        "Tue Nov 22 2022 11:07:31 GMT-0600 (hora estándar central)"
21      ),
22      type: 1,
23    };
24
25    await createConsulta(newConsultation);
26    const obtainedConsultations = await getConsultationsUser(
27      "phtsagoSetMFP8jAkjRQHLx60jB3"
28    );
29
30    const expectedConsultation = {
31      birthdate: "10/17/1987",
32      date: "11/22/2022",
33      doctor: "Charlie Mike Argueta Figueroa ",
34      doctorImage:
35        obtainedConsultations[obtainedConsultations.length - 1].doctorImage,
36      email: "mauricio@gmail.com",
37      genderPatient: "Masculino",
38      id: obtainedConsultations[obtainedConsultations.length - 1].id,
39      patient: "Mauricio Eli Díaz Mena",
40      patientImage:
41        obtainedConsultations[obtainedConsultations.length - 1].patientImage,
42      phone: 87676765,
43      state: 2,
44      time: obtainedConsultations[obtainedConsultations.length - 1].time,
45      type: 1,
46    };
47
48    expect(obtainedConsultations[obtainedConsultations.length - 1]).toEqual(
49      expectedConsultation
50    );
51  });
52 });
53
```

Ilustración 60. Script prueba de sprint 2

```
PASS src/__tests__/register-consultation.test.js
PRUEBAS SPRINT 2
  ✓ debería retorna listado de consultas (1ms)
  ✓ debería retornar el listado de consultas con la nueva consulta agregada

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.846s, estimated 2s
Ran all test suites matching /register-consultation.test.js/i.

Active Filters: filename /register-consultation.test.js/
  > Press c to clear filters.

Watch Usage
  > Press a to run all tests.
  > Press f to run only failed tests.
  > Press o to only run tests related to changed files.
  > Press q to quit watch mode.
  > Press t to filter by a test name regex pattern.
  > Press p to filter by a filename regex pattern.
  > Press Enter to trigger a test run.
```

Ilustración 61. Resultados de pruebas Sprint 2

3.3. Pruebas Sprint 3

A continuación, se muestran los scripts correspondientes a las pruebas unitarias realizadas sobre la aplicación según el **caso de prueba P05**.

```
1 import { getPatients } from "../lib/data/patients.data";
2 import { createUser } from "../lib/data/user.data";
3
4 describe("PRUEBAS SPRINT 3", () => {
5   test("Validar datos ingresados", () => {});
6   test("debería retornar el listado de usuarios con el nuevo usuario agregado", async () => {
7     let newUser = {
8       firstName: "Saira",
9       lastName: "Cienfuegos",
10      phone: 889966655,
11      gender: 1,
12      email: "sairacienfuegosv3@gmail.com",
13      password: "cienfuegos",
14      type: 1,
15    };
16
17    await createUser(newUser);
18
19    const patients = await getPatients();
20
21    let userExist = patients.find((patient) => {
22      if (patient.email === newUser.email) {
23        return true;
24      }
25    });
26
27    expect(userExist).toBeTruthy();
28  });
29 });
30
```

Ilustración 62. Script prueba de sprint 3

```
PASS src/__tests__/register-user.test.js
PRUEBAS SPRINT 3
  ✓ Validar datos ingresados (1ms)
  ✓ debería retornar el listado de usuarios con el nuevo usuario agregado

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.834s, estimated 2s
Ran all test suites matching /register-user.test.js/i.

Active Filters: filename /register-user.test.js/
  > Press c to clear filters.

Watch Usage
  > Press a to run all tests.
  > Press f to run only failed tests.
  > Press o to only run tests related to changed files.
  > Press q to quit watch mode.
  > Press t to filter by a test name regex pattern.
  > Press p to filter by a filename regex pattern.
  > Press Enter to trigger a test run.
```

Ilustración 63. Resultados de pruebas Sprint 3

3.4. Pruebas Sprint 4

A continuación, se muestran los scripts correspondientes a las pruebas unitarias realizadas sobre la aplicación según el caso de prueba P06.

```
1 import { getPatients } from "../lib/data/patients.data";
2
3 describe("PRUEBAS SPRINT 4", () => {
4   test("validar campos: correo y contraseña", () => {});
5   test("debería retornar true si se encuentra un usuario con el correo y contraseña ingresados", async () => {
6     let email = "miriam@gmail.com";
7     let password = "miriam";
8
9     const patients = await getPatients();
10
11    let userExist = patients.find((patient) => {
12      if (patient.email === email && patient.password === password) {
13        return true;
14      }
15    });
16
17    expect(userExist).toBeTruthy();
18  });
19 });
20
```

Ilustración 64. Script prueba de sprint 4

```
PASS src/__tests__/login.test.js
PRUEBAS SPRINT 4
  ✓ validar campos: correo y contraseña (1ms)
  ✓ debería retornar true si se encuentra un usuario con el correo y contraseña ingresados

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 1.832s, estimated 2s
Ran all test suites matching /login.test.js/i.

Active Filters: filename /login.test.js/
  > Press c to clear filters.

Watch Usage
  > Press a to run all tests.
  > Press f to run only failed tests.
  > Press o to only run tests related to changed files.
  > Press q to quit watch mode.
  > Press t to filter by a test name regex pattern.
  > Press p to filter by a filename regex pattern.
  > Press Enter to trigger a test run.
```

Ilustración 65. Resultados de pruebas Sprint 4

3.5. Casos de prueba Sprint 1

Test Case ID	P01 Y P02	Test Case Description	Agregar cita con los datos ingresados y cargar lista de citas agendadas del usuario actual.		
Created By	Carlos Duarte	Reviewed By	Carlos Duarte	Version	1.3

QA Tester's Log

Tester's Name	Carlos Duarte	Date Tested	22/11/2022	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	-------------------------------------------	------

S #	Prerequisites:
1	Crear conexión a Firebase
2	Crear funcionalidad que añada documento a la colección de citas
3	Crear funcionalidad que lista los documentos de la colección de citas
4	

S #	Test Data
1	newMeeting = {Object}
2	
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Crear objeto newMeeting con las propiedades esperadas por la función para añadir citas: fecha, dirección,	Objeto creado	Objeto creado	Pass

	doctor, paciente, especialidad, estado, hora, tipo			
2	Ejecutar funcionalidad para añadir citas pasando como parámetro el objeto creado anteriormente	Función ejecutada creando nueva cita	Nueva cita creada	Pass
3	Ejecutar función que retorna la lista de documentos ubicados en la colección de citas filtrando por el usuario actual	Lista de citas del usuario actual obtenidas	Lista de citas obtenida	Pass
4	Validar si la cita agregada anteriormente se encuentra en la lista de citas	Cita agregada se encuentra en la lista obtenida	Cita encontrada	Pass

3.6. Casos de prueba Sprint 2

Test Case ID	P03 Y P04	Test Case Description	Agregar consulta con los datos ingresados y cargar lista de consultas realizadas del usuario actual.		
Created By	Carlos Duarte	Reviewed By	Carlos Duarte	Version	1.0

QA Tester's Log

Tester's Name	Carlos Duarte	Date Tested	22/11/2022	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	-------------------------------------------	------

#	Prerequisites:
1	Crear conexión a Firebase
2	Crear funcionalidad que añada documento a la colección de consultas
3	Crear funcionalidad que lista los documentos de la colección de consultas
4	

S #	Test Data
1	newConsultation = {Object}
2	
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Crear objeto newConsultation con las propiedades esperadas por la función para añadir consultas:	Objeto creado	Objeto creado	Pass

	fecha, doctor, historial de la enfermedad, paciente, razón de la consulta, estado, estado del doctor, hora, tipo			
2	Ejecutar funcionalidad para añadir consulta pasando como parámetro el objeto creado anteriormente	Función ejecutada creando nueva consulta	Nueva consulta creada	Pass
3	Ejecutar función que retorna la lista de documentos ubicados en la colección de consulta filtrando por el usuario actual	Lista de consultas del usuario actual obtenidas	Lista de consultas obtenida	Pass
4	Validar si la consulta agregada anteriormente se encuentra en la lista de consultas	Consulta agregada se encuentra en la lista obtenida	Consulta encontrada	Pass

3.7. Casos de prueba Sprint 3

Test Case ID	P05	Test Case Description	Agregar nuevo usuario con los datos ingresados		
Created By	Carlos Duarte	Reviewed By	Carlos Duarte	Version	1.0

QA Tester's Log

Tester's Name	Carlos Duarte	Date Tested	22/11/2022	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	-------------------------------------------	------

S #	Prerequisites:
1	Crear conexión a Firebase
2	Crear funcionalidad que añada documento a la colección de users
3	Crear funcionalidad que lista los documentos de la colección de users
4	

S #	Test Data
1	newUser = {Object}
2	
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Crear objeto newUser con las propiedades esperadas por la función para añadir consultas:	Objeto creado	Objeto creado	Pass

	firstName, lastName, phone, gender, email, password, type			
2	Ejecutar funcionalidad para añadir nuevo usuario pasando como parámetro el objeto creado anteriormente	Función ejecutada creando nuevo usuario	Nuevo usuario creado	Pass
3	Ejecutar función que retorna la lista de documentos ubicados en el usuario de consulta filtrando por el usuario actual	Lista de usuarios	Lista de usuarios obtenida	Pass
4	Validar si el usuario agregado anteriormente se encuentra en la lista de usuarios	usuario agregado se encuentra en la lista obtenida	usuario encontrado	Pass

3.8. Casos de prueba Sprint 4

Test Case ID	P06	Test Case Description	Inicio de sesión		
Created By	Carlos Duarte	Reviewed By	Carlos Duarte	Version	1.0

QA Tester's Log

Tester's Name	Carlos Duarte	Date Tested	22/11/2022	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	-------------------------------------------	------

S #	Prerequisites:
1	Crear conexión a Firebase
2	Crear funcionalidad que lista los documentos de la colección de users
3	
4	

S #	Test Data
1	Let email = ""
2	Let password = ""
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Crear variables para que almacenen las credenciales del usuario que intenta iniciar sesión: email y password	Variables creadas	Variables creadas	Pass

2	Ejecutar funcionalidad para listar documentos de la colección users	Lista de usuario obtenida	Lista de usuarios obtenida	Pass
3	Validar si las credenciales ingresadas se encuentran dentro de la lista de usuarios	Credenciales encontradas	Credenciales encontradas	Pass