



**UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



**DISEÑO Y DESARROLLO DE APLICACIÓN PARA APOYO EN LA
CONFIRMACIÓN DEL ALTA DE PACIENTES HOSPITALIZADOS EN EL
SERVICIO DE MEDICINA INTERNA DEL HOSPITAL LAS HIGUERAS**

POR

Fernanda Arlette Henríquez Parra

Informe Final Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de
Concepción para optar al grado académico de Ingeniera Civil Biomédica

Profesores Guía
Rosa Liliana Figueroa Iturrieta
Hernán Carrillo Bestagno

Comisión
Mario Medina Carrasco
Esteban Pino Quiroga

Agosto
Concepción
(Chile)

© 2024 Fernanda Arlette Henríquez Parra

© 2024 Fernanda Arlette Henríquez Parra

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.



Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido de alguna manera a este logro.

En primer lugar, quiero agradecer a mi familia por ser mi fuente de inspiración. A mis queridos abuelos, les agradezco por siempre estar a mi lado, brindándome su amor incondicional y constante apoyo. Su presencia, cada palabra de aliento y cada abrazo lleno de cariño ha sido fundamental en mi vida.

A mi mamita preciosa, darle un profundo agradecimiento por su apoyo inquebrantable y su presencia incondicional. Gracias por creer en mí siempre y ser mi mayor defensora, sin duda este logro es tanto suyo como mío.

Quiero agradecer a mi novio Diego, quien ha sido fundamental en todo momento, dándome su apoyo y confiando en mis capacidades siempre.

A mis profesores, agradecer por su dedicación y su compromiso con mi crecimiento académico. En especial, quiero reconocer el cariño de mi profesora Rosa Figueroa, quien no solo me brindó orientación académica, sino también su apoyo en momentos difíciles. Su comprensión y aliento fueron fundamentales para mí, estaré eternamente agradecida.

A mis queridos compañeros y amigos, les agradezco por compartir momentos que siempre recordaré y sin duda han hecho que esta etapa sea mucho más enriquecedora e inolvidable. Gracias por su amistad, compañerismo y apoyo constante.

Finalmente, a todos aquellos que han contribuido a mi crecimiento y desarrollo, les doy las gracias de todo corazón. Su generosidad, amistad y apoyo fueron indispensables en este paso por la universidad.

Resumen

Esta memoria de título se enfoca en el diseño y desarrollo de una aplicación web para optimizar el proceso de gestión de altas de pacientes hospitalizados en el Servicio de Medicina Interna del Hospital Las Higueras de Talcahuano. Actualmente, la gestión de información se realiza mediante planillas de registro digitales, las cuales presentan importantes limitaciones en términos de eficiencia, precisión y seguridad de los datos. El objetivo del proyecto es desarrollar una aplicación especializada que supere estas limitaciones, mejorando la calidad de la atención médica y facilitando la comunicación entre el personal médico y de enfermería.

La aplicación propuesta está diseñada para reducir los tiempos de estancia hospitalaria, asegurar la confidencialidad y privacidad de la información del paciente y disminuir el tiempo dedicado a tareas administrativas. Además, pretende mejorar la coordinación y minimizar errores, proporcionando una herramienta intuitiva y fácil de usar para el personal hospitalario.

El desarrollo de la aplicación se llevó a cabo utilizando la metodología ágil Scrum, con la participación de los profesionales clínicos como usuarios finales. Estos profesionales colaboraron en el diseño y desarrollo, aportando información sobre los requerimientos y validando las entregas del proyecto para asegurar la pertinencia y efectividad de la aplicación en el contexto hospitalario.

Los principales logros del proyecto incluyen la implementación de un sistema de autenticación seguro, el desarrollo de interfaces intuitivas para el ingreso y visualización de datos, y la integración con Firebase para el almacenamiento y sincronización de datos en tiempo real. Las pruebas realizadas con datos anonimizados de 100 pacientes confirmaron que la aplicación cumple con los requisitos funcionales y es fácil de usar.

En conclusión, la aplicación desarrollada mejora significativamente la eficiencia, precisión y capacidad de gestión en el proceso de altas hospitalarias. La participación de los profesionales clínicos en el proceso de diseño y desarrollo garantizó que la aplicación satisficiera sus necesidades, proporcionando una solución tecnológica que fortalece la seguridad de los datos del paciente y optimiza los procesos diarios del personal médico. Esta herramienta no solo apoya la gestión de altas, sino que también podría ser utilizada para la entrega oficial de turnos, un aspecto crucial para la calidad de la atención en el hospital.

Abstract

This thesis focuses on the design and development of a web application aimed at optimizing the patient discharge management process for hospitalized patients in the Internal Medicine Service of Hospital Las Higueras in Talcahuano. Currently, information management is conducted through digital record sheets, which present significant limitations in terms of efficiency, accuracy, and data security. The goal of this project is to develop a specialized application that overcomes these limitations, improving the quality of medical care and facilitating communication between medical and nursing staff.

The proposed application is designed to reduce hospital stay times, ensure the confidentiality and privacy of patient information, and decrease the time dedicated to administrative tasks. Additionally, it aims to improve coordination and minimize errors, providing an intuitive and easy-to-use tool for hospital staff.

The application was developed using the agile Scrum methodology, with the active participation of clinical professionals as end-users. These professionals collaborated in the design and development process, providing information on requirements and validating project deliverables to ensure the application's relevance and effectiveness in the hospital context.

The main achievements of the project include the implementation of a secure authentication system, the development of intuitive interfaces for data entry and visualization, and the integration with Firebase for real-time data storage and synchronization. Tests conducted with anonymized data from 100 patients confirmed that the application meets functional requirements and is easy to use.

In conclusion, the developed application significantly improves the efficiency, accuracy, and management capacity of the hospital discharge process. The involvement of clinical professionals in the design and development process ensured that the application met their needs, providing a technological solution that strengthens patient data security and optimizes the daily processes of medical staff. This tool not only supports discharge management but could also be used for the official handover of shifts, a crucial aspect of quality care in the hospital.

Tabla de Contenidos

CAPÍTULO 1. INTRODUCCIÓN	1
1.1. INTRODUCCIÓN GENERAL	1
1.2. OBJETIVOS	2
1.2.1 <i>Objetivo General</i>	2
1.2.2 <i>Objetivos Específicos</i>	2
1.3. ALCANCES Y LIMITACIONES	2
1.4. METODOLOGÍA	2
1.5. TEMARIO	3
CAPÍTULO 2. REVISIÓN BIBLIOGRÁFICA	5
2.1. INTRODUCCIÓN	5
2.2. MEDICINA HOSPITALARIA Y ROL DE LOS HOSPITALISTAS	5
2.2.1 <i>Adopción de la medicina hospitalaria en Chile</i>	5
2.3. MEDIDAS Y DESAFÍOS EN EL SISTEMA DE SALUD DE CHILE	6
2.3.1 <i>Sistema GRD</i>	6
2.3.2 <i>Duración de la estancia hospitalaria</i>	6
2.4. ESTRATEGIAS HACIA LA EFICIENCIA CLÍNICA	7
2.5. ESTANDARIZACIÓN Y DIGITALIZACIÓN DE VISITAS TÉCNICAS	7
2.6. TRABAJOS PREVIOS	8
2.6.1 <i>SIGHFA: Aplicación web con módulo de gestión hospitalaria</i>	8
2.6.2 <i>Aplicación para la atención y seguimiento de pacientes hospitalizados</i>	9
2.6.3 <i>EDUS: software con sistemas de información integrados</i>	10
2.7. DISCUSIÓN	11
CAPÍTULO 3. MARCO TEÓRICO	12
3.1. INTRODUCCIÓN	12
3.2. APLICACIONES WEB	12
3.3. BENEFICIOS DE LA DIGITALIZACIÓN EN LA SALUD	12
3.4. SEGURIDAD EN APLICACIONES WEB	13
3.5. USABILIDAD EN EL DESARROLLO DE SOFTWARE	13
3.5.1 <i>Visual Studio Code</i>	13
3.5.2 <i>Flutter</i>	13
3.5.3 <i>Firebase</i>	14
3.6. DISCUSIÓN	14
CAPÍTULO 4. METODOLOGÍA	15
4.1. DESCRIPCIÓN GENERAL	15
4.2. APROBACIÓN COMITÉ DE ÉTICA-CIENTÍFICO Y CONSENTIMIENTO INFORMADO	16
4.2.1 <i>Reuniones y observaciones</i>	16
4.2.2 <i>Levantamiento de requerimientos</i>	16
4.2.3 <i>Desarrollo</i>	17
4.3. OBTENCIÓN Y FILTRADO DE DATOS GRD	17
4.4. USO DE DATOS ANONIMIZADOS	18
4.4.1 <i>Obtención de resultados</i>	18
CAPÍTULO 5. DESARROLLO	19
5.1. INTRODUCCIÓN	19
5.2. REQUISITOS FUNCIONALES Y NO FUNCIONALES	19
5.3. ENTORNO DE DESARROLLO Y ARQUITECTURA DE LA APLICACIÓN	19
5.4. ESTRUCTURA DE LA BASE DE DATOS	20
5.5. DISEÑO DE LA APLICACIÓN	21
5.6. CREACIÓN Y CONFIGURACIÓN DEL PROYECTO	22
5.7. IMPLEMENTACIÓN DE PANTALLAS Y FUNCIONES	23
CAPÍTULO 6. RESULTADOS	36

6.1.	INTRODUCCIÓN	36
6.2.	RESULTADOS DE LAS ENTREVISTAS Y FOCUS GROUPS.....	36
6.2.1	<i>Proceso actual de visitas clínicas y sus limitaciones</i>	36
6.2.2	<i>Hallazgos de los Focus groups y entrevistas</i>	36
6.2.3	<i>Desarrollo de sprints</i>	37
6.3.	RESULTADOS DE LA APLICACIÓN WEB	37
6.3.1	<i>Pantalla de inicio de sesión</i>	37
6.3.2	<i>Pantalla de crear cuenta</i>	38
6.3.3	<i>Pantalla de recuperar contraseña</i>	39
6.3.4	<i>Pantalla principal</i>	40
6.3.5	<i>Pantalla de ingresar visitas</i>	41
6.3.6	<i>Resumen diario</i>	42
6.3.7	<i>Pantalla Archivos de visitas</i>	44
6.3.8	<i>Pantalla GRD</i>	45
6.3.9	<i>Historial de salas</i>	46
6.3.10	<i>Administración de usuarios</i>	47
6.4.	PRUEBAS DE FUNCIONALIDAD.....	47
6.4.1	<i>Resultados de checklist de verificación de requerimientos</i>	48
CAPÍTULO 7. CONCLUSIONES		49
7.1.	DISCUSIÓN	49
7.2.	CONCLUSIONES.....	50
7.3.	TRABAJO FUTURO.....	50
CAPÍTULO 8. GLOSARIO.....		52
CAPÍTULO 9. REFERENCIAS.....		53
ANEXO A. PROCEDIMIENTO GENERAL SIGUIENDO UNA METODOLOGÍA SCRUM.....		56
ANEXO B. MÉTODO DE ESTRATEGIAS HACIA LA EFICIENCIA CLÍNICA SUGERIDO.....		57
ANEXO C. PARTICIPANTES DE FOCUS GROUPS Y ENTREVISTAS SEMIESTRUCTURADAS		58
ANEXO D. RESULTADOS DE HISTORIAS DE USUARIO Y PRODUCT BACKLOG		59
ANEXO E. DESARROLLO DE SPRINTS		60
ANEXO F. BASE DE DATOS		61
ANEXO G. RESULTADOS ADICIONALES DE LA APLICACIÓN.....		63
ANEXO H. CHECKLIST DE VERIFICACIÓN DE REQUERIMIENTOS		66

Lista de Figuras

Figura 2.1: Vista de página de inicio de sesión de la aplicación SIGHFA [16].....	8
Figura 2.2: Vista de pantalla del módulo de hospitalización de la página web SIGHFA [16].	9
Figura 2.3: Pantalla que muestra a pacientes manejados por el hospital, con opciones de internar, editar y detalle [17].....	9
Figura 2.4: Pantalla de registro para la admisión de un paciente a hospitalización [17].	10
Figura 2.5 Visualización de pantalla del sistema de hospitalización del Software EDUS [18].....	10
Figura 2.6 Visualización dashboards: caracterización de hospitalizaciones en el contexto COVID-19, Costa Rica [18].....	11
Figura 4.1 Comparación entre metodología ágil y tradicional [29].	15
Figura 4.2 Flujo de trabajo de equipo Scrum	16
Figura 5.1 Requisitos funcionales y no funcionales de la aplicación.....	19
Figura 5.2 Diseño de mapa de navegación de aplicación	20
Figura 5.3 Bocetos referenciales para el diseño de la interfaz de la aplicación.	21
Figura 5.4 Código con parte de las dependencias agregadas al archivo <i>'pubspec.yaml'</i> para inicialización de Firebase del proyecto en el archivo <i>'main.dart'</i>	22
Figura 5.5 Verificación de Nivel de Usuario y Redirección en Firebase.....	24
Figura 5.6 Método <i>'_initializeOptionsAndItems'</i> que inicializa las opciones de la barra de navegación.	25
Figura 5.7 Función <i>'_loadSalas'</i> que carga las salas y camas en la planilla de ingreso de visitas de EH1.....	27
Figura 5.8 Extracto de código de la función <i>'_loadVisitas'</i> que permite cargar y guardar los datos.	28
Figura 5.9 Función <i>'_copyRowData'</i> que permite copiar los datos de una fila.	28
Figura 5.10 Código de colores según el tipo de salida seleccionado y construcción de gráfico indicador de progreso de visita.....	28
Figura 5.11 Función <i>'getDailySummaryStream'</i> que agrupa los datos de visitas por equipo.	29
Figura 5.12 Integración de EmailJS en la aplicación de Flutter.....	30
Figura 5.13 Código utilizado para crear una tabla con el resumen diario.....	31
Figura 5.14 Función <i>'_fetchHistorialSalas'</i> que obtiene datos de Firestore.....	32
Figura 5.15 Diagnósticos con mayor frecuencia en el SMI (datos obtenidos desde GRD del HLH).	32

Figura 5.16 Resultados de egresos por diagnóstico de diagnóstico N39.0 de pacientes con severidad moderada.	33
Figura 5.17 Función ‘ <i>filterDiagnosticos</i> ’ que filtra los diagnósticos según el texto ingresado.....	33
Figura 5.18 Parte del código que construye una tabla con los datos del diagnóstico GRD.....	34
Figura 5.19 Código para guardar y cargar Información de usuarios en Firebase Firestore.	34
Figura 6.1 Pantalla de inicio de sesión.....	38
Figura 6.2 Pantalla de crear cuenta.	38
Figura 6.3 Pantalla de recuperar contraseña.....	39
Figura 6.4 Pantalla principal.	40
Figura 6.5 Pantalla de ingreso a planillas de registro de visitas.....	41
Figura 6.6 Pantalla de resumen diario.	43
Figura 6.7 Pantalla de historial con archivos de visitas	44
Figura 6.8 Pantalla con información GRD.....	45
Figura 6.9 Pantalla de historial de salas.....	46
Figura 6.10 Pantalla de administración de usuarios.....	47
Figura A.1 Procedimiento general para el desarrollo de la aplicación siguiendo metodología Scrum.	56
Figura B.1 Métodos de estrategias hacia la eficiencia clínica sugeridas en [6].....	57
Figura C.1 Gráfica con cantidad de participantes (usuarios finales) por nivel de acceso permitido en la aplicación.....	58
Figura D.1 Resultados de historias de usuario.	59
Figura D.2 Product Backlog con requerimientos para la aplicación y su nivel de prioridad.....	59
Figura E.1 Desarrollo de sprints.....	60
Figura F.1 Diagrama de base de datos	61
Figura F.2 Base de datos en Cloud Firestore.	62
Figura G.1 Pantalla de ingreso a la aplicación en distintos navegadores (Chrome y Safari).....	63
Figura G.2 Funcionalidades (1. Selección de opciones y código de colores según tipo de salida, 2. Ingreso de episodio de paciente, 3. Cierre de episodio y 4. Copia de datos del paciente en forma de lista).....	64
Figura G.3 Funcionalidades (1. Visualización de estadías en MIH y HLH, con cambio de color, 2. Calendario para seleccionar fecha de ingreso en HLH y MIH, 3. Mover de sala/cama al paciente).	65

Capítulo 1. Introducción

1.1. Introducción General

En el entorno de la atención médica, la gestión eficiente de pacientes hospitalizados se ha convertido en un pilar fundamental en el Servicio de Medicina Interna del Hospital Las Higueras desde el año 2018, cuando los médicos hospitalistas asumieron esta responsabilidad. Con la llegada de la pandemia de COVID-19, se implementó el uso de planillas de registro como principal herramienta de registro para el seguimiento de pacientes hospitalizados, una práctica que ha perdurado hasta la actualidad. Sin embargo, el registro manual de información clínica presenta limitaciones significativas en términos de seguridad y almacenamiento de datos.

La manipulación y análisis de grandes volúmenes de datos mediante planillas es complejo y limitado, siendo susceptibles a la pérdida y alteración de datos, careciendo de medidas de seguridad y respaldo adecuadas [1]. En cambio, el uso de herramientas especializadas permite una recopilación de información más precisa, estructurada y segura, facilitando su análisis y uso para decisiones e investigaciones clínicas [2]. Por otra parte, el registro manual es propenso a errores, lo que dificulta la estandarización, el análisis de la información y generación de estadísticas. Asimismo, la falta de una base de datos centralizada y herramientas de análisis limita la identificación de patrones y tendencias, desaprovechando el potencial para la investigación y mejora de procesos [3]. Debido a esto, los sistemas digitales son esenciales pues revelan resultados y áreas de mejora [4].

Una encuesta de la Asociación Médica Estadounidense mostró que el porcentaje de médicos que consideran las herramientas de salud digitales como una ventaja aumentó del 85% en 2016 al 93% en 2022 [5]. En línea con estas tendencias, a inicios del 2024 el Ministerio de Salud de Chile (MINSAL) ha propuesto estrategias para mejorar la eficiencia tales como: la digitalización y uso de indicadores GRD, el uso de checklist y otros procesos operativos para una gestión más ágil y eficiente [6]. Sin embargo, estas estrategias son insuficientes para enfrentar los retos actuales y se enfatiza la necesidad de adoptar herramientas más avanzadas que optimicen los recursos y mejoren la gestión del servicio, beneficiando tanto a pacientes como a profesionales de la salud.

Para abordar estas limitaciones y mejorar la eficiencia clínica, este proyecto busca implementar un sistema avanzado para la búsqueda, monitoreo, almacenamiento y recuperación de los datos recopilados en las visitas clínicas. Esta solución permitirá mejorar la toma de decisiones clínicas, optimizar los recursos hospitalarios y promover la investigación científica en medicina interna, contribuyendo a la mejora del proceso de gestión de altas hospitalarias.

1.2. Objetivos

1.2.1 Objetivo General

Desarrollar una aplicación que optimice el proceso de gestión de altas de las visitas técnicas a pacientes hospitalizados en el Servicio de Medicina Interna del Hospital Las Higueras.

1.2.2 Objetivos Específicos

- Levantar y documentar el proceso que se lleva a cabo en las visitas técnicas del Servicio de Medicina Interna, identificando los pasos clave e información relevante que se recopilan durante el proceso.
- Desarrollar una aplicación web que permita la recopilación, registro y visualización de datos durante las visitas técnicas en el Servicio de Medicina Interna.
- Evaluar funcionalidad y usabilidad de la aplicación web a nivel de laboratorio con usuarios expertos.

1.3. Alcances y Limitaciones

- i. Este proyecto, se centrará en desarrollar una aplicación web para facilitar la gestión de altas hospitalarias. La aplicación para desarrollar solo contempla una prueba de concepto con un desarrollo tecnológico TRL3 (pruebas analíticas y cualitativas de laboratorio para validar físicamente las funcionalidades y usabilidad de la tecnología).
- ii. La aplicación web incluye un módulo de análisis de datos y tendencias que se podría ver limitado por desafíos técnicos y de recursos, como la disponibilidad de servidores para su implementación final y la disponibilidad de datos históricos suficientes para el entrenamiento de los modelos.
- iii. Este proyecto cuenta con la aprobación del comité ético científico del Servicio de Salud Talcahuano, Acta N°29, fecha 08 mayo 2024. Dado lo anterior, las actividades a realizar se enmarcan de acuerdo con lo autorizado por el comité.

1.4. Metodología

La implementación de la aplicación sigue una metodología Scrum, conocida por su enfoque flexible y colaborativo. Este proyecto involucra a profesionales clínicos del Servicio de Medicina Interna (SMI) del Hospital Las Higueras (HLH) como usuarios finales, utilizando entrevistas

semiestructuradas y grupos focales para recopilar información. En la investigación preliminar, se revisan estudios sobre la medicina hospitalaria, desafíos del sistema de salud y desarrollo de aplicaciones web. En la fase de diseño y desarrollo, se definen los requisitos funcionales y no funcionales respecto a la información recopilada. De esta forma, se obtienen los bocetos de la aplicación y se desarrollan las funcionalidades requeridas realizando constantes entrevistas con usuarios para correcciones y mejoras. En la etapa resultados, los usuarios finales completan un checklist de evaluación de requerimientos para evaluar la funcionalidad de la aplicación. De esta forma, se obtienen las conclusiones del trabajo realizado y se proponen futuras mejoras.

1.5. Temario

- Capítulo 1. Introducción: Este capítulo proporciona una visión general del trabajo, delineando los objetivos y alcances del proyecto, además de presentar la metodología de desarrollo empleada.
- Capítulo 2. Revisión bibliográfica: En este capítulo se examinan estudios previos relacionados con el tema del proyecto. Se analizan prácticas y tecnologías empleadas en proyectos similares, estableciendo un marco de referencia y justificando la elección de enfoques y herramientas para el trabajo actual.
- Capítulo 3. Marco teórico: Se proporciona información fundamental que respalda el desarrollo del proyecto. Incluye una revisión sobre el rol de la medicina hospitalista, los sistemas de salud utilizados y las herramientas para el desarrollo de aplicaciones web, ofreciendo un sustento teórico para la implementación del prototipo.
- Capítulo 4. Desarrollo: Este capítulo describe el contexto actual del proceso de visita técnica en el SMI del HLH, los participantes involucrados y las herramientas utilizadas para el registro de información. Se detallan los requisitos funcionales y no funcionales identificados, la arquitectura de la aplicación y las decisiones de diseño. Además, se explica la creación y configuración del proyecto y la base de datos, así como la implementación de las diferentes pantallas y funciones de la aplicación.
- Capítulo 6. Resultados: Se presentan los resultados obtenidos de las pruebas de funcionalidad, usabilidad y visualización de datos. También se incluyen los resultados del checklist realizado por los usuarios finales, proporcionando una evaluación integral de la aplicación.

- Capítulo 7. Conclusiones: En este capítulo se presentan las conclusiones del proyecto, destacando los logros y las áreas de mejora. También se discute el trabajo futuro que podría realizarse para continuar perfeccionando la aplicación y ampliar su alcance.



Capítulo 2. Revisión bibliográfica

2.1. Introducción

La medicina hospitalaria liderada por los médicos hospitalistas, desempeña un papel fundamental en la atención de pacientes hospitalizados y en la gestión eficiente de los recursos sanitarios. En este capítulo, se explorará en profundidad el impacto de los médicos hospitalistas en la estadía hospitalaria, así como los desafíos que enfrenta el sistema de salud en relación con la duración de la hospitalización. Además, se analizará el Sistema de Grupos Relacionados por el Diagnóstico (GRD) y su influencia en la gestión de recursos y calidad de la atención. Por último, se revisarán trabajos previos sobre el desarrollo de aplicaciones web para el registro y visualización de procesos médicos, proporcionando una base sólida para el desarrollo de estrategias futuras en este ámbito.

2.2. Medicina hospitalaria y rol de los hospitalistas

La Medicina Hospitalaria o Modelo Hospitalista es una disciplina dedicada a proporcionar cuidados médicos integrales a pacientes hospitalizados, además de participar en la enseñanza clínica, investigación y liderazgo en el campo de la medicina interna [7]. Los hospitalistas son médicos que aplican este modelo y tienen formación de medicina general, con especialidad internista y formación subespecializada [8]. A diferencia de los internistas generales, los hospitalistas son miembros de equipos multidisciplinarios que ofrecen una atención completa al paciente, desempeñando un papel crucial en la coordinación de cuidados médicos durante la hospitalización [7]. Trabajan estrechamente con enfermeras y otros equipos de apoyo, coordinando y evaluando intervenciones y exámenes para ajustar el plan de tratamiento según sea necesario a lo largo del día. Esta colaboración permite optimizar el tiempo de hospitalización, agilizar la obtención de resultados y permite una planificación eficiente de altas, manteniendo la calidad de la atención y previniendo reingresos [9]. Su enfoque en la optimización del tiempo de estadía y la planificación del alta contribuye a mejorar los flujos de pacientes dentro del hospital, aumentando la disponibilidad de camas y garantizando una transición fluida hacia la atención ambulatoria [10].

2.2.1 Adopción de la medicina hospitalaria en Chile

La presencia de médicos hospitalistas ha sido parte de un esfuerzo por mejorar la calidad de la atención hospitalaria en Chile, siendo fundamental en la implementación de estrategias orientadas a mejorar los resultados clínicos y la experiencia del paciente durante su estancia en el hospital. Sin

embargo, se requiere un mayor apoyo institucional y gubernamental para su implementación y expansión en el país.

Al evaluar ciertos estándares básicos en numerosos hospitales, se revela una carencia de cultura de seguridad y eficiencia en los procesos, evidenciando que el progreso en salud no reside en la mera expansión hospitalaria, sino en una reformulación y reingeniería de los procesos internos [7].

2.3. Medidas y desafíos en el sistema de salud de Chile

En los últimos años, se han aplicado diversas medidas para asignar más recursos al sistema de salud, con el fin de hacer frente al creciente costo de la atención médica, así como para mejorar la gestión y eficiencia del sistema. La hospitalización representa el principal gasto relacionado con el cuidado de los pacientes y es un área clave en la gestión de la salud.

2.3.1 Sistema GRD

En 2020, se implementó una modificación en el sistema de financiamiento de la red hospitalaria en Chile. Este cambio consistió en la transición de un modelo de financiamiento basado en el pago de insumos productivos a otro basado en el pago por actividad o resultados, medido por el sistema de GRD para la atención cerrada o con hospitalización [11].

La implementación de los GRD en hospitales de alta complejidad en Chile ha estandarizado la actividad hospitalaria y facilitado el benchmarking entre instituciones. Esta herramienta permite clasificar a los pacientes de manera uniforme, mejorar la documentación de diagnósticos y facilitar el análisis de datos clínicos [11]. Al estandarizar los diagnósticos con el sistema GRD, se establece un marco común que mejora la precisión y la coherencia en la documentación clínica, lo que contribuye a una mejor comprensión casuística hospitalaria y promueve la toma de decisiones clínicas informadas [12].

2.3.2 Duración de la estancia hospitalaria

La duración de la estancia hospitalaria, medida a través del Length of Stay (LOS), es crucial en la gestión y planificación de prestaciones hospitalarias, ya que los días de estadía prolongados afectan la producción interna y la seguridad del paciente. Un estudio realizado en hospitales públicos de Chile que operan bajo el sistema de financiamiento por GRD reveló que factores como el sexo, la edad, el ingreso por servicio de urgencias, el reingreso, la gravedad del caso y la complejidad del tratamiento afectan la duración de la estadía. Además, se encontraron diferencias en la duración de la

estadía según el día de la semana de ingreso y la categoría de ingreso de Fonasa (Fondo Nacional de Salud de Chile) [13].

2.4. Estrategias hacia la eficiencia clínica

El documento "Estrategias hacia la eficiencia clínica hospitalaria desde la Unidad de Emergencias" [6] del MINSAL de Chile propone medidas para mejorar la atención hospitalaria. Presenta una metodología detallada que incluye la identificación de áreas críticas y la disponibilidad de camas, resaltando la importancia de analizar los indicadores GRD. Estos indicadores, como los egresos anuales, la ocupación de camas y la mortalidad hospitalaria, son fundamentales para evaluar el rendimiento del hospital y encontrar áreas de mejora. Además, se menciona que la digitalización de estos indicadores y procesos operativos puede tener un impacto significativo en la eficiencia y calidad de la atención, ofreciendo una gestión más ágil y transparente. Se proponen también métodos tradicionales como checklist y planillas de relleno, los cuales se pueden ver en el Anexo B, Figura B.1. Sin embargo, estas estrategias pueden ser insuficientes para los desafíos actuales. Por ello, se destaca la importancia de adoptar una tecnología digital hospitalaria para mejorar la calidad de atención y optimizar los recursos disponibles, beneficiando tanto a pacientes como a profesionales de la salud y modernizando el sistema de salud.

2.5. Estandarización y digitalización de visitas técnicas

La visita clínica propuesta como estrategia para la eficiencia clínico-hospitalaria es una valiosa fuente de generación de datos y su estandarización permitiría una recolección precisa y segura, proporcionando un sólido respaldo a las decisiones del equipo clínico en la gestión del alta de pacientes y la continuidad de su cuidado [6]. Por ello, contar con los requisitos y el diseño básico de un sistema para digitalizar, almacenar, analizar y aprovechar los datos recopilados de las visitas técnicas es fundamental para implementar una estrategia efectiva de gestión del alta del paciente [14] [15]. Para asegurar un levantamiento de requisitos efectivo que conduzca a un diseño de software adecuado, es esencial emplear metodologías flexibles, adaptativas y que promuevan la interacción y colaboración directa con los usuarios de la solución. Las metodologías ágiles en la ingeniería de requerimientos ofrecen un enfoque iterativo e incremental que facilita la captura de requisitos comprensibles para los usuarios y valiosos para los desarrolladores mediante la creación de prototipos de interfaz en cada ciclo. Los requerimientos se pueden representar mediante mockups generados a partir de historias de usuarios, que contienen funcionalidades y elementos de arquitectura. Este

método se destaca en el diseño de sistemas de apoyo a la toma de decisiones clínicas, al no imponer una solución definitiva desde el principio [16].

2.6. Trabajos previos

Esta revisión muestra proyectos de aplicaciones similares que han abordado la gestión hospitalaria, el registro de altas médicas y la coordinación de cuidados en entornos clínicos. Esto permitirá identificar elementos que contribuyan a enriquecer el diseño y la funcionalidad de la aplicación propuesta.

2.6.1 SIGHFA: Aplicación web con módulo de gestión hospitalaria

En el Hospital de Especialidades N°1 de las Fuerzas Armadas de Ecuador, se desarrolló una aplicación web utilizando JAVA y siguiendo la metodología ágil Scrum. El objetivo fue mejorar la gestión de áreas críticas como emergencia, consulta externa, hospitalización y enfermería, creando un sistema que agilizará dichos procesos. Para el almacenamiento de datos, utilizó base de datos relacional y fue diseñada para adaptarse a las necesidades específicas del hospital, permitiendo su implementación en diversos entornos. Esta aplicación web facilita la realización de tareas diarias del personal de salud y el registro detallado de las acciones que realizan en los distintos módulos [16]. En la Figura 2.1 se puede ver la página de ingreso de la página web SIGHFA y en la Figura 2.2 se puede ver el módulo de hospitalización.

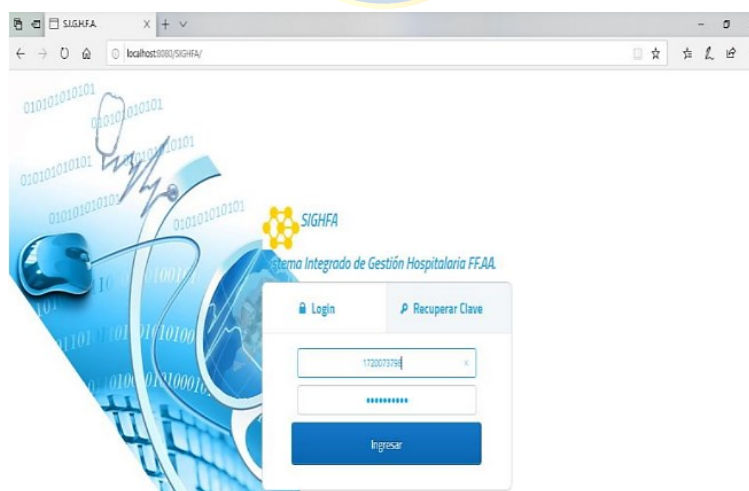


Figura 2.1: Vista de página de inicio de sesión de la aplicación SIGHFA [16].

The screenshot shows the 'Sistema Integrado de Gestión Hospitalaria FFAA' interface. On the left is a dark sidebar menu with options like 'MODULO MEDICO', 'Admision', 'Agenda Médica', 'Emergencia', 'Pacientes en Emergencia', 'Triage - Rack pacientes', 'Hospitalización', 'Consulta Externa', 'Odontología', 'Solicitudes', 'Quirofanos', 'Labores de Enfermería', and 'Certificados Médicos'. The main area displays a table of patients under the 'Pacientes Atendidos' tab. The table has columns for 'Ord.', 'Fecha', 'Hora', 'Nro. Turno', 'Nombre Paciente', 'No.HC', 'Edad', 'Situación', 'Llamar a Paciente', 'Atender a Paciente', and 'Registrar Abandono Paciente'. There are 5 rows of patient data.

Ord.	Fecha	Hora	Nro. Turno	Nombre Paciente	No.HC	Edad	Situación	Llamar a Paciente	Atender a Paciente	Registrar Abandono Paciente
1	15/05/2019	13:34:22	2	TORRES BARRIONUEVO LUIS FERNANDO	5156	54	HUJO SP			
2	03/03/2020	15:12:40	2	COYACHAMIN VELAZQUE JOSE ANDRES	0503102634	33	MILITAR ACTIVO			
3	04/03/2020	08:51:30	1	CHANALLUSA COYACHAMIN ANTHONY SNEYDER	423733		MILITAR PASIVO			
4	13/03/2020	10:00:01	1	COYACHAMIN VELAZQUE SEGUNDO GERARDO	0502918238	36	MILITAR ACTIVO			
5	20/03/2020	11:20:13	1	RODRIGUEZ BENAVIDES JOSE LUIS		34	HUJO SA			

Figura 2.2: Vista de pantalla del módulo de hospitalización de la página web SIGHFA [16].

2.6.2 Aplicación para la atención y seguimiento de pacientes hospitalizados

En el Hospital P.N.P del Perú, se desarrolló un sistema [17] para abordar la problemática de la deficiente atención en el proceso de administración de medicamentos y seguimiento del estado del paciente hospitalizado. Este sistema fue desarrollado en el lenguaje C# y utilizando SQL Server 2014 como base de datos, siguiendo la metodología Rational Unified Process (RUP). El proyecto incluyó la elaboración de tres módulos principales: módulo de atenciones médicas, seguimiento de pacientes y el módulo de gestión de enfermeras. El sistema inicia con el ingreso de usuario, que al autenticarse accede a la pantalla de consulta de pacientes donde se puede registrar la admisión de un paciente a hospitalización como se puede ver en la Figura 2.3. Para ello, se completa el formulario que se observa en la Figura 2.4, registrando la información: médico, sala, síntomas, medicamentos y observaciones.

The screenshot shows the 'SB ADMIN 2' web application interface. The main area is titled 'Consulta de pacientes' and displays a table of patients. The table has columns for 'Nombre', 'N° Documento', 'Estado', and 'Acciones'. There are 4 rows of patient data. The 'Acciones' column contains buttons for 'Internar', 'Editar', and 'Detalle'.

Nombre	N° Documento	Estado	Acciones
IRENDA VENERIO	70334436	EN SALA	
FRED RAMOS	88754612	EN SALA	
JUAN P. OHRES	70334434	EN SALA	
MIGUEL SALAZAR	123456781	EN SALA	

Figura 2.3: Pantalla que muestra a pacientes manejados por el hospital, con opciones de internar, editar y detalle [17].

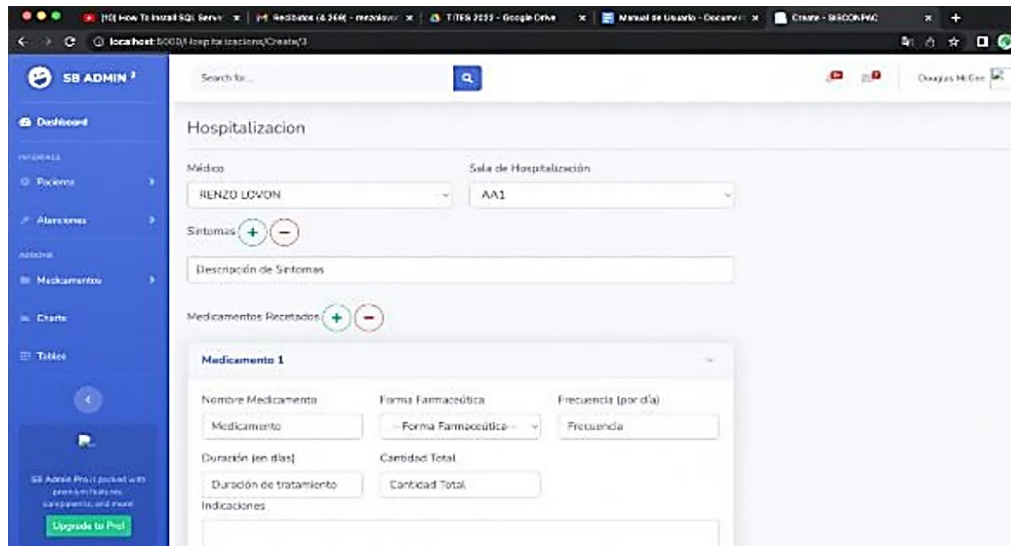


Figura 2.4: Pantalla de registro para la admisión de un paciente a hospitalización [17].

2.6.3 EDUS: software con sistemas de información integrados

El Expediente Digital Único en Salud (EDUS) [18], desarrollado en Costa Rica, ha sido reconocido como un avance significativo en sistemas de salud digital. Este software está compuesto por varios sistemas diseñados para diferentes necesidades y usuarios, entre ellos el sistema SIES y ARCA, que se enfocan en la gestión del ingreso y egreso hospitalario. Las funciones incluyen el registro detallado de datos de ingreso, seguimientos de la estancia hospitalaria y la gestión de altas, como se puede ver en la Figura 2.5. Además, EDUS cuenta con un sistema de reportes con dashboards, que facilita la visualización y exportación de datos estadísticos, siendo utilizado en contexto de pandemia, como se puede ver en la Figura 2.6, logrando visualizar las estadísticas de hospitalizaciones por COVID-19.

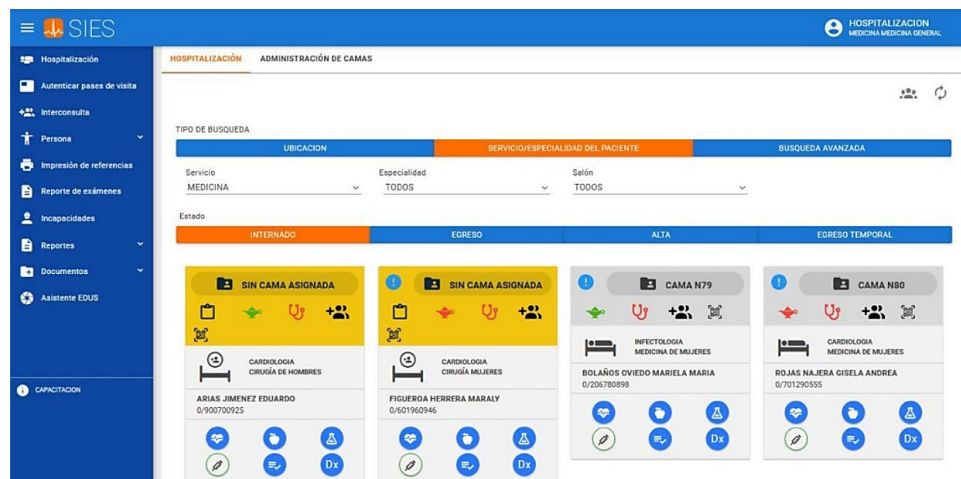


Figura 2.5 Visualización de pantalla del sistema de hospitalización del Software EDUS [18].



Figura 2.6 Visualización dashboards: caracterización de hospitalizaciones en el contexto COVID-19, Costa Rica [18].

Para el desarrollo del software se utilizó el lenguaje de programación JAVA y PostgreSQL como base de datos, incorporando medidas de seguridad como la autenticación electrónica e implementando una estrategia de seguridad de datos desde el inicio del proceso de creación, destacando la importancia de utilizar contraseñas robustas y un manejo adecuado de estas.

2.7. Discusión

La revisión bibliográfica destaca el papel crucial de los médicos hospitalistas en la atención y gestión hospitalaria, así como los desafíos que enfrenta el sistema de salud en términos de eficiencia clínica. A pesar de las estrategias propuestas por el MINSAL, como el uso de checklist o planillas de relleno, estos métodos podrían ser insuficientes ante los retos actuales.

Por otra parte, la adopción de tecnología digital para la recolección y almacenamiento de datos, junto con la estandarización de la duración de la estancia hospitalaria mediante el sistema GRD, ofrece mayor claridad en las estadísticas hospitalarias. A pesar de la posible resistencia al cambio, esta transición hacia la estandarización mejorará la eficiencia y calidad de la atención médica, estableciendo un modelo ejemplar para otros servicios hospitalarios. Tras esto, la implementación de una aplicación web para la gestión del alta de pacientes hospitalizados representa un avance significativo, agilizando los procesos administrativos y mejorando la transparencia en la gestión de datos clínicos. De esta forma, los resultados obtenidos respaldan la efectividad y subrayan la necesidad de seguir avanzando en este ámbito, donde el desarrollo de una aplicación web emerge como una solución efectiva para optimizar la gestión del alta de pacientes hospitalizados y contribuir a una atención hospitalaria de mayor calidad y seguridad.

Capítulo 3. Marco teórico

3.1. Introducción

En este capítulo, se exploran los elementos fundamentales para comprender el desarrollo de aplicaciones web, desde la definición y el funcionamiento de una aplicación web hasta la diferenciación entre el frontend y el backend en su desarrollo. Por otra parte, se indagan los beneficios de la digitalización y aspectos cruciales como la seguridad y usabilidad. Asimismo, se presentan las herramientas clave utilizadas en el desarrollo de aplicaciones, como Visual Studio Code, Flutter y Firebase, resaltando la importancia en la creación de aplicaciones web eficientes y funcionales.

3.2. Aplicaciones web

Una aplicación web es un tipo de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (servidor web) y el protocolo de transferencia de hipertexto (HTTP) están estandarizados y no necesitan ser creados por el programador de aplicaciones [19]. A diferencia de las aplicaciones de escritorio, las aplicaciones web no requieren instalación local, ya que se accede a través de un navegador y se ejecutan en un servidor remoto, permitiendo un acceso rápido desde cualquier dispositivo con conexión a Internet.

Las aplicaciones web, destacan su compatibilidad multiplataforma, pudiendo ejecutarse en diferentes sistemas operativos. Además, requieren menos recursos de hardware en comparación con las aplicaciones de escritorio y ofrecen un entorno más protegido para los datos de los usuarios al almacenarse en servidores confiables, protegiéndolos contra virus y fallos de disco duro [20].

El modelo cliente-servidor es un diseño arquitectónico donde las tareas se reparten entre proveedores de recursos o servicios, llamados servidores, y los solicitantes de servicios, denominados clientes. Este modelo permite que las aplicaciones web funcionen eficientemente distribuyendo las cargas de trabajo. En este contexto, el frontend actúa como el cliente que solicita datos y servicios, mientras que el backend, alojado en un servidor, procesa estas solicitudes y maneja la lógica de negocio y almacenamiento de datos [20].

3.3. Beneficios de la digitalización en la salud

La digitalización en el ámbito de la salud ofrece numerosos beneficios, incluyendo la mejora en la precisión y disponibilidad de datos, reducción de errores humanos, y optimización de recursos. Los sistemas digitales permiten una mejor coordinación del cuidado del paciente, facilitando el acceso

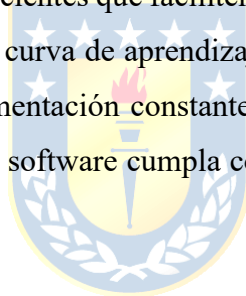
a la información clínica en tiempo real y mejorando la toma de decisiones médicas. Además, la digitalización permite la realización de análisis avanzados y estudios epidemiológicos, contribuyendo a la investigación médica y a la mejora continua de los servicios de salud [21].

3.4. Seguridad en aplicaciones web

La seguridad es un aspecto crucial en el desarrollo de aplicaciones web, especialmente en el sector de la salud, donde se maneja información sensible. Las medidas de seguridad incluyen la autenticación y autorización de usuarios, encriptación de datos, implementación de firewalls y prácticas de desarrollo seguro como la validación de entradas y la protección contra ataques [22].

3.5. Usabilidad en el desarrollo de software

La usabilidad es un aspecto clave en el desarrollo de software, especialmente en aplicaciones destinadas a ser utilizadas por profesionales de la salud. La usabilidad en el desarrollo de software se enfoca en crear interfaces intuitivas y eficientes que faciliten la interacción del usuario con el sistema. Este enfoque es esencial para reducir la curva de aprendizaje y aumentar la satisfacción del usuario. Las pruebas de usabilidad y la retroalimentación constante son cruciales para identificar y corregir problemas de diseño, asegurando que el software cumpla con las necesidades de los usuarios finales y mejore su experiencia general [23].



3.5.1 Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta popular para el desarrollo de software debido a su flexibilidad, eficiencia y amplia gama de extensiones. Ha sido diseñado para ser ligero pero potente, ofrece características avanzadas para la edición de código, depuración, control de versiones y la integración con herramientas y servicios populares. Su soporte para múltiples lenguajes de programación y su comunidad activa de desarrolladores han contribuido a su popularidad en el desarrollo de aplicaciones web, móviles y de escritorio [24].

3.5.2 Flutter

Flutter es un SDK (Software Development Kit) de código abierto desarrollado por Google para la creación de aplicaciones móviles nativas para Android e iOS a partir de un código base. Utilizando el lenguaje de programación Dart, Flutter permite a los desarrolladores construir interfaces

de usuario atractivas y de alto rendimiento con widgets personalizables y una arquitectura reactiva. Se ha convertido en una opción popular para el desarrollo de aplicaciones móviles multiplataforma, por su enfoque en la rapidez de desarrollo, la consistencia visual y la experiencia de usuario. En el contexto del desarrollo de aplicaciones web, Flutter también se utiliza para construir interfaces de usuario web, lo que permite a los desarrolladores compartir una gran parte del código entre las aplicaciones móviles y web, proporcionando una experiencia coherente en todas las plataformas [25].

3.5.3 Firebase

Firestore es una plataforma de desarrollo de aplicaciones móviles y web, con estructura NoSQL y desarrollada por Google, que ofrece una amplia gama de servicios backend para ayudar a los desarrolladores a crear aplicaciones de alta calidad de manera rápida y eficiente. Entre sus servicios principales se encuentra una base de datos alojada en la nube que permite la autenticación de usuarios, almacenamiento de archivos, hosting web y mensajería en la nube, entre otros. Estos servicios, escalables y seguros, permiten a los desarrolladores enfocarse en la creación de experiencias de usuario innovadoras sin preocuparse por la infraestructura backend. Además, Firestore proporciona una solución completa para las necesidades de backend, permitiendo almacenar datos, autenticar usuarios y enviar notificaciones a través de una API simple e intuitiva, siendo su base de datos especialmente útil para aplicaciones que requieren funcionalidades en tiempo real [26].

3.6. Discusión

En el marco teórico se destacan componentes clave para el desarrollo de aplicaciones web en salud, como la arquitectura cliente-servidor, herramientas robustas (Visual Studio Code, Flutter y Firestore), digitalización, seguridad y usabilidad. La arquitectura cliente-servidor optimiza el rendimiento y la escalabilidad, crucial para manejar grandes volúmenes de datos. Además, la digitalización mejora la precisión de los datos y la eficiencia operativa, mientras que la seguridad es vital para proteger la información sensible. Por último, la usabilidad garantiza una experiencia de usuario intuitiva, facilitando la adopción de la aplicación. Esta combinación de elementos es especialmente relevante en aplicaciones que registran visitas clínicas en hospitales, donde la precisión y la eficiencia son esenciales.

Capítulo 4. Metodología

4.1. Descripción general

En este proyecto se utilizó la metodología ágil Scrum debido a su enfoque flexible y colaborativo, que permite adaptarse a cambios y mejorar continuamente el producto. A diferencia de las metodologías tradicionales de desarrollo de software, como se muestra en la Figura 4.1, Scrum comienza definiendo las funcionalidades básicas del proyecto en el Product Backlog, mejorándolo continuamente a través de iteraciones llamadas “sprints”. Cada sprint divide el proyecto en tareas pequeñas y concretas que se desarrollan en períodos breves. Durante cada sprint, el equipo completa tareas definidas en colaboración con los usuarios finales, quienes también establecen prioridades. Al finalizar el sprint, se realiza una revisión del trabajo y una retrospectiva para identificar áreas de mejora [27].

Los roles principales en Scrum incluyen al “Product Owner”, que gestiona el Product Backlog, crea historias de usuario y evalúa los resultados de cada sprint; el “Scrum Master”, que facilita el proceso y elimina obstáculos; y el “Development Team”, que ejecuta las tareas. Los usuarios finales colaboran en la creación de historias de usuario y validan el desarrollo del producto [28]. Este ciclo iterativo y colaborativo permite la entrega continua de productos de alta calidad, alcanzando los objetivos del proyecto y obteniendo el producto final deseado. Los documentos clave en Scrum son el Product Backlog, el Sprint Backlog y las historias de usuario [29].

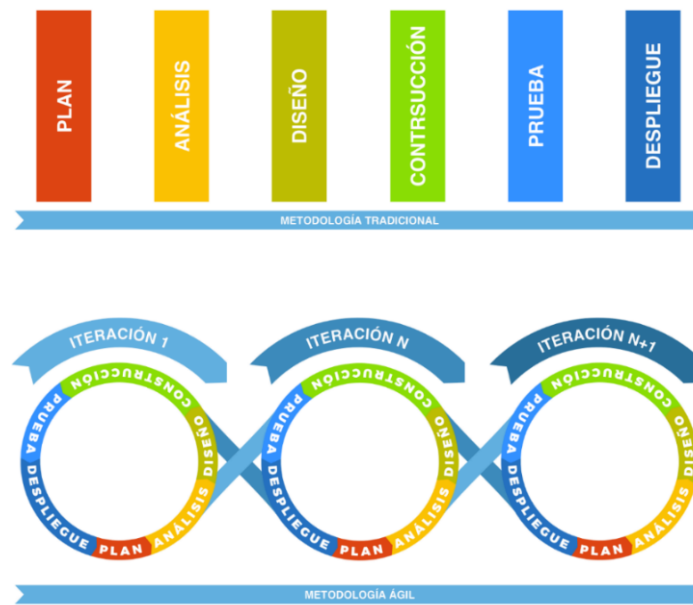


Figura 4.1 Comparación entre metodología ágil y tradicional [29].

4.2. Aprobación Comité de ética-científico y consentimiento informado

El proyecto fue aprobado por el Comité de Ética Científico (CEC) del Servicio de Salud de Talcahuano (SST), asegurando que todas las actividades de investigación cumplieran con los estándares éticos y legales aplicables. Los participantes del proyecto, incluyendo médicos, enfermeras y administradores del SMI del HLH, fueron informados sobre los objetivos y procedimientos del estudio. Se les proporcionó un formulario de consentimiento informado, el cual debieron firmar para aceptar su participación voluntaria. Este consentimiento garantizó que los participantes estuvieran plenamente informados y de acuerdo con su participación en el proyecto.

4.2.1 Reuniones y observaciones

Para entender mejor el contexto del proyecto, se llevaron a cabo reuniones con profesionales del SMI del HLH. Además, se realizaron visitas técnicas de observación para analizar el proceso actual de registro de visitas clínicas y recopilar información valiosa directamente de los usuarios finales.

4.2.2 Levantamiento de requerimientos

El product backlog se desarrolló en colaboración con los clientes a través de entrevistas y sesiones de trabajo. Durante estas sesiones, se identificaron y priorizaron las funcionalidades necesarias para la aplicación, asegurando que todas las necesidades clave fueran abordadas. Para el desarrollo de historias de usuario, se organizaron grupos focales con los usuarios finales para comprender mejor cómo interactuarían con la aplicación y qué tareas realizarían. En la Figura 4.2 se puede observar el flujo de trabajo Scrum utilizado.

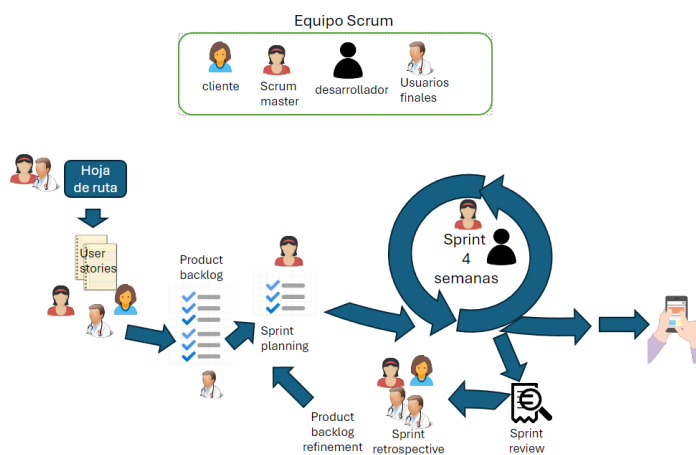


Figura 4.2 Flujo de trabajo de equipo Scrum

Este proyecto contó con la participación de profesionales clínicos del SMI del HLH, quienes aceptaron colaborar en el diseño y desarrollo de la aplicación como usuarios finales. Para ello, se emplearon técnicas conversacionales, como entrevistas semiestructuradas individuales y grupos focales, para recopilar información detallada y validar el diseño y desarrollo de la aplicación. Las entrevistas grupales reunieron a expertos que compartían una experiencia común, con el objetivo de discutir y elaborar temas relevantes desde su perspectiva personal. Esta técnica reactiva y semidirigida favoreció la integración de perspectivas individuales y la comparación de diferencias [30]. Los guiones de las entrevistas se elaboraron antes de cada sesión, ajustándose al progreso de cada etapa de la metodología, con el fin de ofrecer una solución intuitiva y adaptable a las necesidades específicas del SMI del HLH, mejorando así la eficiencia y la calidad de la atención médica a los pacientes hospitalizados.

A partir de estas discusiones y colaboraciones, se crearon historias de usuario detalladas, que sirvieron como base para el desarrollo de la aplicación. En la Figura A.1 del Anexo A, se puede observar el procedimiento general que se siguió para el desarrollo de aplicación utilizando la metodología Scrum.

4.2.3 Desarrollo

Utilizando los requisitos recopilados, se crearon mockups y mapas de navegación para la interfaz de usuario. Siguiendo los requisitos funcionales y no funcionales, y basándose en los diseños preliminares, se implementaron las funcionalidades de la aplicación. Este proceso se realizó en sprints iterativos, con revisiones constantes y ajustes basados en la retroalimentación de los usuarios, recogiendo sus opiniones y sugerencias, las cuales fueron incorporadas en las iteraciones posteriores del desarrollo.

4.3. Obtención y filtrado de datos GRD

Para proporcionar a los médicos una referencia precisa sobre la cantidad de días de estadía de los pacientes según el diagnóstico, se llevó a cabo un trabajo de recopilación y filtrado de datos del sistema GRD del hospital. Los pasos seguidos fueron los siguientes:

1. **Filtrado de Diagnósticos:** Para asegurar la relevancia y utilidad de la información, se realizó un filtrado de los diagnósticos para identificar aquellos con mayor frecuencia de egresos. Este proceso permitió concentrar el análisis en los diagnósticos más comunes y relevantes del SMI del HLH.

Los datos se extrajeron para los meses comprendidos entre enero de 2023 y diciembre de 2024. Además, se seleccionaron pacientes mayores de 15 años con un nivel de severidad menor, moderada y mayor, cuyo motivo de egreso fue el retorno al domicilio. Este enfoque permitió centrarse en las estadias hospitalarias que culminaron en una recuperación, ofreciendo así una perspectiva clara sobre los diagnósticos y las estadias medianas de los pacientes que se recuperaron satisfactoriamente.

2. **Recopilación de datos:** Los datos recopilados se estructuraron en tablas donde se reflejan los diagnósticos más frecuentes. Luego se aplican los filtros y se obtienen las estancias correspondientes.
3. **Integración en la Aplicación:** Los datos filtrados y las estadias medianas calculadas se integraron en la aplicación para su visualización, mostrando los diagnósticos junto con la estadia mediana desglosada por rango de edad y nivel de severidad.

4.4. Uso de datos anonimizados

Para probar la funcionalidad y rendimiento del sistema y asegurar su fiabilidad, se utilizaron 100 registros de pacientes anonimizados. Estos registros cubrieron el ciclo de atención, desde el ingreso del paciente hasta su alta, permitiendo evaluar el desempeño del sistema.

La anonimización de los datos garantizó la protección de la privacidad y confidencialidad de los pacientes, cumpliendo con las normativas vigentes en materia de protección de datos.

4.4.1 Obtención de resultados

Al finalizar el desarrollo, se llevaron a cabo entrevistas semiestructuradas y grupos focales con los usuarios finales para evaluar la funcionalidad y usabilidad de la aplicación. Esta retroalimentación fue crucial para realizar los ajustes y mejoras finales.

Se aplicó un checklist con pruebas predefinidas para verificar las principales funcionalidades de la aplicación. Este checklist fue evaluado por usuarios expertos, quienes revisaron el cumplimiento de cada ítem, asegurando que la aplicación cumpliera con los requisitos y expectativas de los usuarios finales.

Capítulo 5. Desarrollo

5.1. Introducción

En este capítulo se detallan los aspectos relacionados con el diseño y desarrollo de la aplicación. Se presentan los requisitos funcionales y no funcionales identificados durante el proceso de análisis, así como la arquitectura general de la aplicación, las decisiones de diseño tomadas y las funcionalidades implementadas para cumplir con los objetivos del proyecto.

5.2. Requisitos funcionales y no funcionales

En la

Requisitos funcionales	
RF1	Registro de visitas clínicas que incluya campos como nombre del paciente, diagnóstico, tratamiento y observaciones.
RF2	Acceso por nivel de usuario, asegurando que cada uno acceda solo a la información y funcionalidades pertinentes a su rol.
RF3	Generación automática de informes y estadísticas que permitan un análisis rápido y preciso de datos.
RF4	Exportación de planillas en formato PDF para facilitar la distribución y archivo de la información.
RF5	Asociar los diagnósticos con los días de estancia hospitalaria basados en el sistema GRD.
RF6	Proporcionar herramientas de búsqueda y filtrado para facilitar el seguimiento de pacientes por parte del personal médico.
Requisitos no funcionales	
RNF1	Implementación de autenticación de usuarios mediante contraseñas y medidas adicionales de seguridad para proteger los datos.
RNF2	Desarrollo de una interfaz de usuario intuitiva y fácil de usar, que minimice la necesidad de formación y soporte.
RNF3	Diseñar el sistema para que pueda manejar un aumento en el volumen de datos y usuarios sin comprometer el rendimiento.
RNF4	Asegurar que la aplicación sea compatible con una variedad de dispositivos y navegadores webs utilizados por el personal médico.
RNF5	Garantizar la disponibilidad del sistema las 24 horas del día, los 7 días de la semana.

Figura 5.1 Requisitos funcionales y no funcionales de la aplicación.

se presentan los requisitos funcionales y no funcionales de la aplicación, según lo recopilado en las entrevistas y focus groups.

5.3. Entorno de desarrollo y arquitectura de la aplicación

Las decisiones de diseño tomadas durante el desarrollo del proyecto se fundamentaron en la necesidad de flexibilidad, eficiencia y usabilidad. Se optó por una arquitectura de tres capas, dividiéndose en presentación, lógica de negocio y acceso a datos, con la finalidad de facilitar el mantenimiento y la expansión futura de la aplicación.

Para el desarrollo del frontend, se seleccionó Flutter debido a su capacidad de crear interfaces de usuario interactivas y su soporte para desarrollo multiplataforma, lo que permite una implementación fluida en diferentes dispositivos. La capa de presentación consiste en interfaces de

usuario desarrolladas en Flutter, proporcionando una experiencia intuitiva y eficiente. La lógica de negocio, también desarrollada en Flutter, se encarga de las operaciones y la gestión de datos.

Firestore fue elegido para la base de datos y servicios de autenticación gracias a su integración sencilla con Flutter y sus capacidades de gestión de datos en tiempo real y seguridad. La capa de acceso a datos utiliza Firestore para almacenar y recuperar información, garantizando la integridad y seguridad de los datos.

Se empleó Visual Studio Code como entorno de desarrollo principal debido a sus características avanzadas como el completado de código y la depuración integrada, lo cual facilita el proceso de desarrollo. Para gestionar las dependencias del proyecto, se utilizó el gestor de paquetes de Dart, permitiendo integrar bibliotecas externas de manera sencilla.

Adicionalmente, se ha integrado el servicio de mensajería de Email JS para enviar correos electrónicos, complementando las funcionalidades de notificación y comunicación dentro de la aplicación.

A continuación, en la Figura 5.2 se incluye el diseño del mapa de navegación de la aplicación.

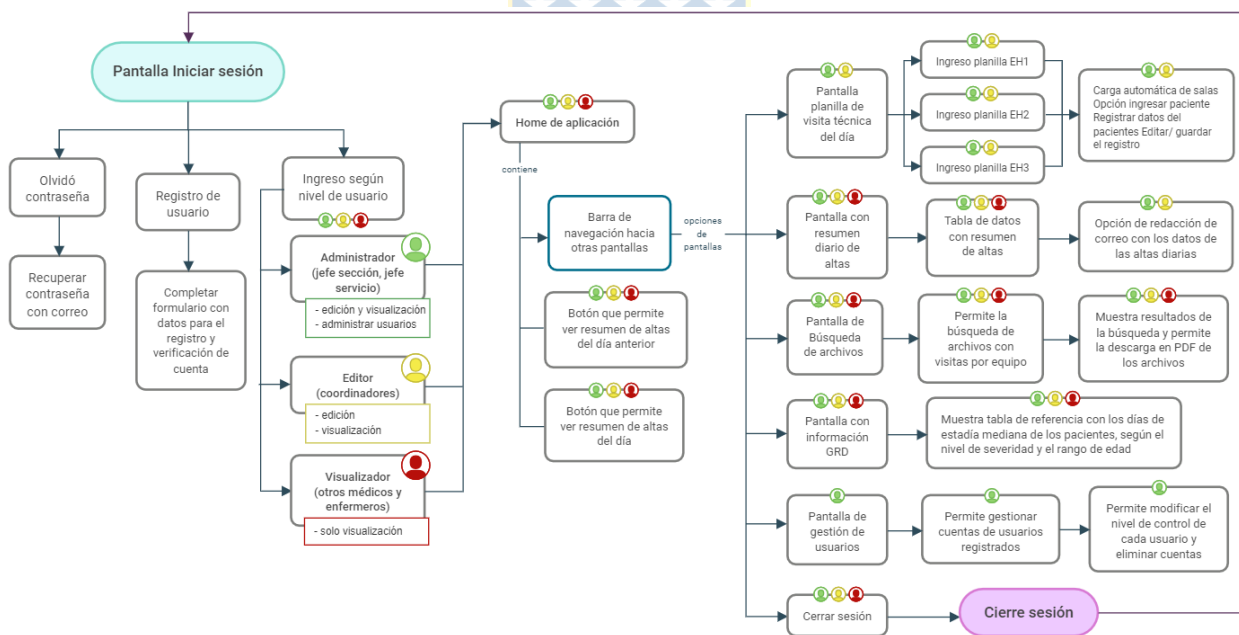


Figura 5.2 Diseño de mapa de navegación de aplicación.

5.4. Estructura de la base de datos

La base de datos de la aplicación se estructura en torno a varias colecciones principales. La colección "usuarios" almacena información como nombre, tipo de usuario, RUT, y nivel de acceso. La colección "pacientes" contiene datos sobre los pacientes, incluyendo ID, nombre, apellidos y RUT.

La colección "episodios" registra los episodios de hospitalización de cada paciente. La colección "visitas" detalla cada visita técnica realizada, con información sobre sala, condición, diagnósticos, equipo, fechas de ingreso y alta, y tipo de salida. La colección "salas" mantiene un historial de ocupación de las salas. La colección "GRD" almacena códigos de diagnósticos con su severidad y descripción. La colección "severidad" asocia la severidad de los diagnósticos con rangos de edad. La colección "resumen_diario" registra datos agregados diarios sobre altas, pacientes en distintos estados y traslados. Finalmente, la colección "estadísticas" almacena datos como fechas, altas y porcentajes de altas. En la Figura F.1 del Anexo F, se presenta el diagrama de la base de datos, donde se detalla la estructura de cada colección y los datos que almacena.

5.5. Diseño de la aplicación

En esta sección se detalla el proceso de diseño de la interfaz de usuario de la aplicación. Se incluyen los bocetos (mockups) creados para visualizar la apariencia y funcionalidad de las diferentes pantallas de la aplicación antes de proceder con la implementación. Estos bocetos fueron creados utilizando la plataforma www.app.mogups y posteriormente fueron presentados a los participantes del proyecto para obtener retroalimentación y asegurar que el diseño cumpliera con sus expectativas y necesidades. El contenido de cada uno de los bocetos se presenta en la Figura 5.3.

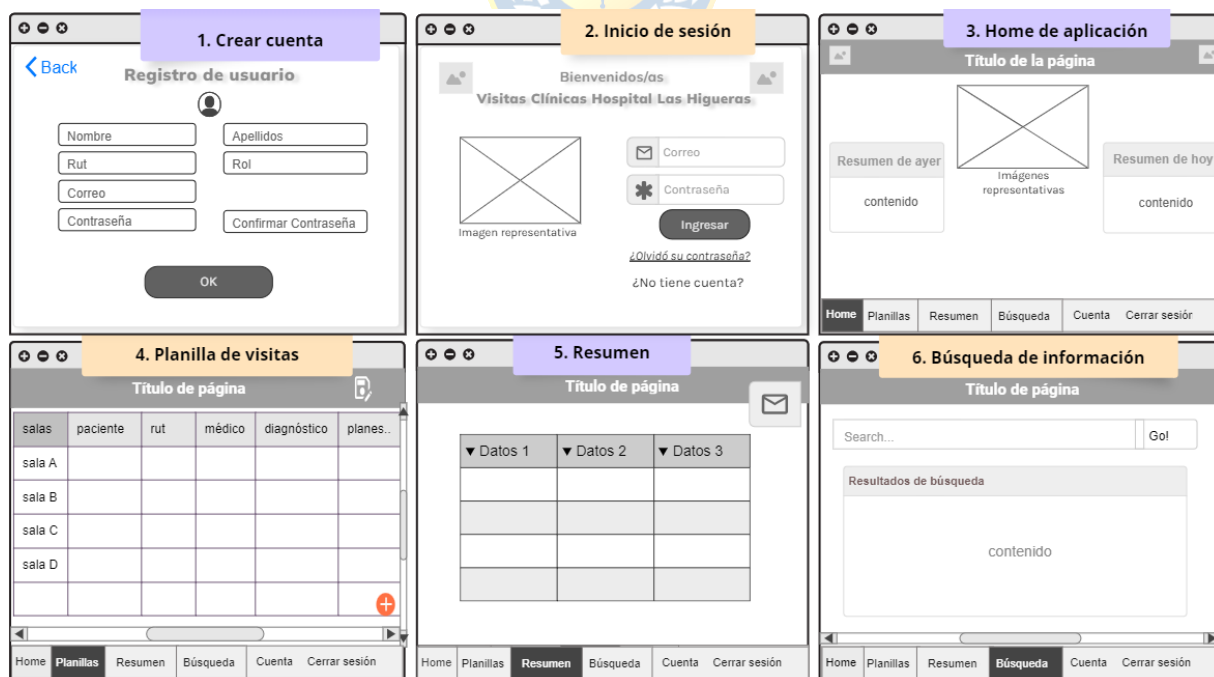


Figura 5.3 Bocetos referenciales para el diseño de la interfaz de la aplicación.

5.6. Creación y configuración del proyecto

1. Principales carpetas del proyecto

Los principales archivos del proyecto se muestran en la Figura 5.4, donde:

- **/lib:** es la carpeta que contiene todas las pantallas de la aplicación. Contiene archivos como *'main.dart'*, *'inicio_sesion.dart'*, *'home_app.dart'*, *'grd.dart'*, *'planilla_diaria_eh1'*.
- **/assets:** es la carpeta que contiene las imágenes utilizadas en la aplicación.
- **/web:** contiene el archivo *'index.html'* correspondiente a la página principal para el hosting web con Firebase.
- **Dependencias:** *'pubspec.yaml'* es el archivo de configuración de Flutter que incluye las dependencias y otros parámetros del proyecto, *'README.md'* documentación del proyecto.
- **Firebase:** *'firebase'* contiene archivos de configuración de Firebase, *'firebaserc'* que contiene el archivo de configuración del proyecto Firebase, *'firebase.json'* es el archivo de configuración para Firebase Hosting.

2. Configuración de Firebase

Se configuró Firebase para la autenticación y la base de datos en tiempo real habilitando los servicios de Firestore Database y Firebase Authentication, estableciendo reglas de seguridad para asegurar la integridad de los datos. Para su integración con Flutter se agregaron las dependencias necesarias en el archivo *'pubspec.yaml'* del proyecto en Flutter, donde además se visualiza la inicialización de Firebase.

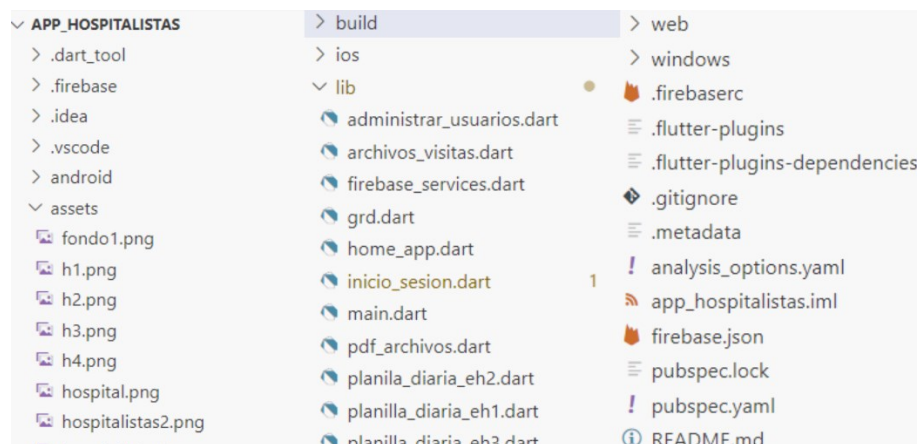


Figura 5.4 Principales archivos del proyecto.

5.7. Implementación de pantallas y funciones

La implementación de las pantallas y funciones de la aplicación se llevó a cabo utilizando el framework Flutter, el cual proporciona una interfaz de usuario flexible y moderna. Para la autenticación y manejo de datos, se integró Firebase para proporcionar servicios en la nube, como la autenticación de usuarios y el almacenamiento de datos. Esto se logró mediante el paquete *'firebase_core'*, que inicializa Firebase con las credenciales correspondientes. A continuación, se describen las principales pantallas con sus principales funcionalidades y widgets incorporados, destacando los paquetes y dependencias utilizadas, así como fragmentos de código relevantes.

1. Pantalla de Inicio de sesión

La pantalla de inicio debe permitir el registro de usuario, la recuperación de contraseña y el ingreso a la aplicación.

- Estructura principal:** Para ello, se utilizó el widget *'Scaffold'* del paquete *'material.dart'* de Flutter para proporcionar una estructura básica consistente con las directrices de Material Design. Este widget incluye una *'AppBar'* que muestra el título "Inicio de Sesión", lo que facilita la identificación de la funcionalidad de la pantalla por parte del usuario.
- Formulario de inicio de sesión:** El núcleo de la pantalla es un formulario compuesto por varios *'TextFormField'* widgets, donde los usuarios ingresan su correo electrónico y contraseña. Estos campos están decorados con *'InputDecoration'* para proporcionar pistas visuales y etiquetas, guiando al usuario sobre qué información debe ingresar. Cada campo tiene validadores asociados para asegurar que los datos ingresados cumplan con los formatos esperados. Se utiliza un *'ElevatedButton'* para enviar el formulario, estilizado para destacarse como el principal punto de interacción en la pantalla. La propiedad *'onPressed'* del botón está vinculada a una función que maneja la lógica de autenticación. Además, se incluye un *'TextButton'* para opciones secundarias como "Olvidé mi contraseña". La función *'_submitForm'* gestiona el estado del formulario, validando cada campo y llamando a *'_signIn'* para autenticar al usuario con Firebase Authentication, como muestra la Figura 5.5. En caso de errores, se utiliza *'_showErrorDialog'* para mostrar mensajes amigables al usuario.
- Seguridad y buenas prácticas:** Se siguen buenas prácticas de seguridad, como ocultar las contraseñas con la propiedad *'obscureText'* y utilizar validaciones tanto en el cliente como en el

servidor para asegurar que solo usuarios autorizados puedan acceder a la aplicación. La integración con Firebase Authentication se realiza mediante el paquete *firebase_auth* de Flutter, permitiendo gestionar de forma segura el registro, inicio de sesión y sesiones de usuarios. Esto simplifica gran parte del desarrollo al delegar la gestión de la autenticación a un servicio confiable y bien mantenido.

```
User? user = await _authService.signInWithEmailAndPassword(email, password);
if (user != null) {
  if (user.emailVerified) {
    //se obtiene el nivel de usuario desde Firebase
    DocumentSnapshot<Map<String, dynamic>> userDoc = await FirebaseFirestore.instance
      .collection('usuarios')
      .doc(user.uid)
      .get();

    String? nivelUsuario = userDoc.data()?['nivel_usuario'];

    if (nivelUsuario == 'Por definir') {
      setState(() {
        _errorMessage = 'Solicite acceso a su administrador.';
      });
    } else if (nivelUsuario == 'Administrar' ||
      nivelUsuario == 'Editar' ||
      nivelUsuario == 'Visualizar') {
      Navigator.push(
        context, MaterialPageRoute(builder: (context) => HomeApp(nivelUsuario: nivelUsuario!)).

```

Figura 5.5 Verificación de nivel de usuario y redirección en Firebase.

2. Pantalla home de la aplicación

La pantalla principal actúa como el punto de inicio desde donde los usuarios pueden navegar a otras partes de la aplicación. Esta pantalla utiliza un *BottomNavigationBar* para la navegación entre las diferentes secciones de la aplicación, como el resumen diario, la administración de usuarios y el cierre de sesión. Similar a la pantalla de inicio de sesión, se utiliza *Scaffold* para proporcionar la estructura básica de la pantalla, incluyendo una barra de navegación inferior.

- **Configuración de acceso:** Para controlar el acceso a las diferentes pantallas y funciones de la aplicación se utiliza el método de la Figura 5.6 *initializeOptionsAndItems*.

```

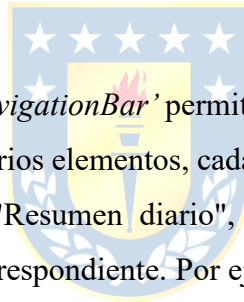
void _initializeOptionsAndItems() {
  _opcionesWidget = [
    _InicioWidget(),
    if (widget.nivelUsuario != 'Visualizar') _IngresarVisitaWidget(),
    ResumenDiarioWidget(nivelUsuario: widget.nivelUsuario),
    const ArchivosVisitas(),
    const GrdPage(),
    const HistorialSalas(), // Añadir la nueva página del historial de salas
    if (widget.nivelUsuario == 'Administrar') const AdministrarUsuariosPage(),
    const Text('Cerrar Sesión', style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),
  ];

  _items = [
    const BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Inicio'),
    if (widget.nivelUsuario != 'Visualizar') const BottomNavigationBarItem(icon: Icon(Icons.assignment), label: 'Ingresar visita'),
    const BottomNavigationBarItem(icon: Icon(Icons.ballot), label: 'Resumen diario'),
    const BottomNavigationBarItem(icon: Icon(Icons.manage_search_outlined), label: 'Archivos Visitas'),
    const BottomNavigationBarItem(icon: Icon(Icons.local_fire_department), label: 'GRD'),
    const BottomNavigationBarItem(icon: Icon(Icons.bed), label: 'Historial'), // Añadir ícono de cama para historial
    if (widget.nivelUsuario == 'Administrar') const BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Administración'),
    const BottomNavigationBarItem(icon: Icon(Icons.exit_to_app), label: 'Cerrar Sesión'),
  ];
}

```

Figura 5.6 Método `'_initializeOptionsAndItems'` que inicializa las opciones de la barra de navegación.

Este método configura las opciones de widgets y los ítems de la barra de navegación inferior basándose en el nivel de usuario proporcionado al widget `'HomeApp'`. Esto permite controlar el acceso a diferentes partes de la aplicación según los permisos del usuario otorgados por el usuario administrador.



- Estructura principal:** El `'BottomNavigationBar'` permite la navegación entre diferentes secciones de la aplicación, configurado con varios elementos, cada uno representando una sección diferente como "Inicio", "Ingresar visita", "Resumen diario", etc. Para cada sección de la barra de navegación, se define un widget correspondiente. Por ejemplo, `'_InicioWidget'` para la página de inicio, `'_IngresarVisitaWidget'` para ingresar visitas, `'ResumenDiarioWidget'`, `'ArchivosVisitas'`, `'GrdPage'`, `'HistorialSalas'` y `'AdministrarUsuariosPage'` para la gestión de usuarios.
- Principales widgets y funciones:** El `'BottomNavigationBar'` permite la navegación entre diferentes secciones de la aplicación, configurado con varios elementos (`'BottomNavigationBarItem'`), cada uno representando una sección diferente como "Inicio", "Ingresar visita", "Resumen diario", etc. Para cada sección de la barra de navegación, se define un widget correspondiente. Por ejemplo, `'_InicioWidget'`, `'_IngresarVisitaWidget'` para ingresar visitas, `'ResumenDiarioWidget'`, `'ArchivosVisitas'`, `'GrdPage'`, `'HistorialSalas'` y `'AdministrarUsuariosPage'` para la gestión de usuarios.

3. Pantallas planillas diarias

La pantalla de planillas diarias debe permitir la selección de la planilla para el ingreso del registro de visitas según el equipo de médicos hospitalistas y solo debe estar disponible para usuarios editores y administradores. Para manejar el ingreso a esta pantalla se añade una condición tal que solo estará visible si el usuario es igual a *'editor || administrador'*. Cada opción contiene el archivo correspondiente a una planilla diaria para un equipo específico (EH1, EH2, EH3). Al presionar alguna opción, se dirige a la opción de plantilla de visita donde la pantalla de ingreso de visitas permite a los usuarios ingresar detalles específicos de visitas diarias. En cada planilla diaria se utiliza una combinación de widgets de entrada de datos (*'TextField'*, *'DropDownButton'*), botones de envío (*'ElevatedButton'*), y conectividad a Firestore para guardar los datos ingresados por el usuario.

- **Estructura principal:** Se utiliza *'Scaffold'* para la estructura básica de la pantalla con una barra de aplicaciones. Cada planilla diaria está envuelta en un *'Form'* para facilitar la validación y gestión de los campos de entrada. Utiliza una clave global (*'_formKey'*) para identificar el formulario y validar sus campos. El *'TextField'*, *'DropDownButton'*, *'DateField'* y *'EditableCell'* se utilizan para capturar varios tipos de datos, como el nombre del paciente, su RUT, sala, diagnóstico, e incluyen validación básica para asegurarse de que los campos no estén vacíos.
- **Construcción de planilla:** Para construir la planilla se utilizó *'DataTable'* el cual permite mostrar datos en forma tabular con *'TableRow'* para definir las filas y *'TableCell'* para definir las celdas, y cada celda puede contener texto o widgets personalizados. Para visualizar los datos en una tabla desplazable tanto horizontal como verticalmente, se utiliza una combinación de *'SingleChildScrollView'* y *'DataTable'*. Luego, con la función de la Figura 5.7 *'_loadSalas'* se cargan las salas y camas en la planilla.

```

Future<void> _loadSalas() async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  List<String>? savedSalas = prefs.getStringList('eh1_salaOptions');

  if (savedSalas != null && savedSalas.isNotEmpty) {
    setState(() {
      _salaOptions = savedSalas;
      _initializeAndLoadData();
    });
  } else {
    setState(() {
      _isLoading = true;
    });
    try {
      List<String> salaOptions = [];
      var salas = ['salasUGA', 'salasAISLA', 'sala3', 'sala4'];
      for (var sala in salas) {
        QuerySnapshot snapshot = await _firestore.collection('salas').doc('eh1salas').collection(sala).get();
        for (var doc in snapshot.docs) {
          salaOptions.add(doc.id);
        }
      }
    } catch (e) {}
  }
}

```

Figura 5.7 Función ‘_loadSalas’ que carga las salas y camas en la planilla de ingreso de visitas de EHI.

La función ‘_loadVisitas’ carga los datos de las visitas que se van registrando a diario como se muestra en la Figura 5.8, donde se llama a la función ‘_saveVisita’ para guardar los datos ingresados en la base de datos de Firestore, asegurando que toda la información se almacene de manera segura y esté disponible. Además, se utiliza el selector de fechas del widget ‘Date and Time Picker’ que permite ingresar las fechas desde un calendario desplegable. Luego, para la funcionalidad del botón de copia, se añadió la función ‘_copyRowData’ de la Figura 5.9, que utiliza ‘ClipboardData’ la cual permite copiar la fila de datos de cada paciente en formato de lista como se puede apreciar en la Figura 5.10. Luego, para permitir el cambio de sala se utilizó el despliegue de un ‘alertDialog’ donde se puede visualizar y seleccionar una sala disponible para el cambio de sala.

```

var rowDataIndex = initialData.indexWhere((element) => element['sala'] == visitaData['sala']);
if (rowDataIndex != -1) {
  initialData[rowDataIndex] = {
    ...initialData[rowDataIndex],
    'rut': visitaData['rut'] ?? '',
    'nombre': visitaData['nombre'] ?? '',
    'apellidos': visitaData['apellidos'] ?? '',
    'edad': visitaData['edad'] ?? '',
    'medico': visitaData['medico'] ?? '',
    'comuna': visitaData['comuna'] ?? '',
    'diagnosticos': visitaData['diagnosticos'] ?? '',
    'planes': visitaData['planes'] ?? '',
    'gestiones_pendientes': visitaData['gestiones_pendientes'] ?? '',
    'fecha_ingreso_med': visitaData['fecha_ingreso_med'] ?? '',
    'fecha_ingreso_hlh': visitaData['fecha_ingreso_hlh'] ?? '',
    'estadia_hlh': visitaData['estadia_hlh'] ?? '',
    'estadia_medicina': visitaData['estadia_medicina'] ?? '',
    'control_residente_turno': visitaData['control_residente_turno'] ?? '',
    'tiempo_probable_alta': visitaData['tiempo_probable_alta'] ?? '',
    'tipo_salida': visitaData['tipo_salida'] ?? '',
    'net': visitaData['net'] ?? '',
    'condicion': visitaData['condicion'] ?? '',
    'episodio': visitaData['episodio'] ?? '',
    'estado_sala': 'Ocupada',
    'equipo': 'eh1',
  };
  // Guardar la nueva visita para hoy
  await _saveVisita(initialData[rowDataIndex]);
}

```

Figura 5.8 Extracto de código de la función ‘*_loadVisitas*’ que permite cargar y guardar los datos.

```

void _copyRowData(Map<String, dynamic> rowData) {
  String rowDataString = rowData.entries.map((entry) => '${entry.key}: ${entry.value}').join('\n');
  Clipboard.setData(ClipboardData(text: rowDataString));
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Datos copiados al portapapeles')),
  );
}

```

Figura 5.9 Función ‘*_copyRowData*’ que permite copiar los datos de una fila.

Por otra parte, se implementó el código de colores según la selección del campo ‘*tipo_salida*’ donde dependiendo de la selección, la fila del paciente cambia de color, como se muestra en la Figura 5.10. En ella también se puede visualizar la construcción del indicador de progreso de la visita, en donde se utilizó la dependencia ‘*percent_indicator*’ para crear el gráfico.

```

color _getColorForTipoSalida(String tipoSalida) {
  switch (tipoSalida) {
    case 'Domicilio':
      return Color.fromARGB(255, 134, 255, 243);
    case 'Hodom':
      return Color.fromARGB(255, 158, 247, 161);
    case 'Hodom Privado':
      return Color.fromARGB(255, 187, 247, 189);
    case 'Otro Servicio HLH':
      return Color.fromARGB(255, 120, 192, 255);
    case 'Rescate a su centro':
      return Color.fromARGB(255, 192, 190, 255);
    case 'Traslado en red':
      return Color.fromARGB(255, 255, 178, 204);
    case 'Traslado fuera de red':
      return Color.fromARGB(255, 255, 176, 241);
    case 'IPC':

```

```

//indicador progreso de visita
Widget _buildCompletionIndicator(Map<String, dynamic> rowData) {
  if (rowData['estado_sala'] == 'Ocupada') {
    double completionPercentage = _calculateCompletionPercentage(rowData);
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 0.0),
      child: CircularPercentIndicator(
        radius: 20.0,
        lineWidth: 4.0,
        percent: completionPercentage,
        center: Text(
          "${(completionPercentage * 100).toInt()}%",
          style: GoogleFonts.poppins(fontSize: 9, fontWeight: FontWeight.bold),
        ), // Text
        progressColor: completionPercentage == 1.0 ?
          Color.fromARGB(255, 17, 255, 0) : Color.fromARGB(255, 72, 0, 255)
      );
    }
}

```

Figura 5.10 Código de colores según el tipo de salida seleccionado y construcción de gráfico indicador de progreso de visita.

- **Validación y limpieza de datos:** Para asegurar la integridad y precisión de los registros en la aplicación hospitalaria, se implementaron diversas estrategias de limpieza de datos. Antes de enviar los datos a la base de datos, se valida la entrada del usuario con los validadores de *'TextFormField'* y *'DropDownButtonFormField'*, asegurando que los campos obligatorios no estén vacíos y sigan el formato correcto. Por otro lado, se utilizan funciones para automatizar la limpieza y normalización de datos, y funciones programadas para ejecutar tareas de actualización de datos diariamente. Estos procesos recopilan y procesan la información del día anterior y preparan los datos para el día siguiente. Los datos se sincronizan automáticamente entre la aplicación y Firestore utilizando *'streams'* y *'listeners'*, reflejando instantáneamente cualquier cambio. Al guardar los datos en Firestore, se verifican nuevamente los campos obligatorios y se eliminan los registros duplicados mediante reglas de seguridad y validaciones en Firestore para asegurar que los datos cumplan con los requisitos esperados.

4. Resumen diario

- **Estructura principal:** La pantalla se define en la clase *'ResumenDiarioWidget'*, que extiende *'StatelessWidget'*. Utiliza un *'Scaffold'* para proporcionar la estructura básica, incluyendo una *'AppBar'* que muestra el título "Resumen Diario". La *'AppBar'* también contiene un botón para enviar el resumen diario por correo electrónico.
- **Stream de datos en tiempo real:** Para mantener los datos actualizados en tiempo real, se utiliza un *'StreamBuilder'* que escucha cambios en la base de datos de Firestore. El Stream se define en la función *'getDailySummaryStream'* que se muestra en la Figura 5.11, la cual obtiene y agrupa los datos de las visitas diarias por equipo (EH1, EH2, EH3). Los datos se estructuran en un mapa que contiene el total de pacientes y los tipos de salida para cada equipo.

```
// carga de pacientes por equipo al RESUMEN DIARIO
Stream<Map<String, Map<String, dynamic>>> getDailySummaryStream() {
  final FirebaseFirestore firestore = FirebaseFirestore.instance;
  DateTime currentDate = DateTime.now();
  String formattedDate = DateFormat('dd-MM-yy').format(currentDate);

  return firestore.collection('pacientes').snapshots().asyncMap((snapshot) async {
    Map<String, Map<String, dynamic>> summary = {
      'eh1': {'Total pacientes': 0, 'pacientes': <String, List<Map<String, String>>>{}},
      'eh2': {'Total pacientes': 0, 'pacientes': <String, List<Map<String, String>>>{}},
      'eh3': {'Total pacientes': 0, 'pacientes': <String, List<Map<String, String>>>{}},
    };
  });
}
```

Figura 5.11 Función *'getDailySummaryStream'* que agrupa los datos de visitas por equipo.

- Construcción de la tabla de resumen:** La función `'_buildResumenDiarioTable'` construye una tabla para mostrar el resumen diario. Esta tabla utiliza `'TableRow'` y `'TableCell'` para definir las filas y celdas, respectivamente. Cada fila representa una categoría de tipo de salida (como "Domicilio", "Hodom", "Fallecido", etc.) y muestra el número de pacientes para cada equipo. La tabla también incluye filas adicionales para mostrar el total de pacientes y las altas por equipo. La tabla se presenta dentro de un `'SingleChildScrollView'` anidado, permitiendo desplazamiento tanto vertical como horizontal para facilitar la navegación y visualización de los datos en dispositivos de diferentes tamaños.
- Envío de email:** Para implementar la funcionalidad de envío de correos electrónicos se utiliza el servicio gratuito de EmailJS [32], un servicio que permite enviar correos electrónicos directamente desde la aplicación sin la necesidad de configurar un servidor de correo propio. Primero, se creó una cuenta en EmailJS y se obtuvieron las credenciales necesarias (`'service_id'`, `'template_id'` y `'user_id'`). En el editor de EmailJS, se diseñó una plantilla de correo con campos personalizables (`{{'to_name'}}`, `{{'subject'}}`, `{{'message'}}`, `{{'from_name'}}`) como se muestra en la Figura 5.12. En la aplicación, se implementó una función `'sendEmail()'` en Dart que utiliza el paquete `'http'` para enviar una solicitud POST a la API de EmailJS, incluyendo un cuerpo JSON con los parámetros de la plantilla y las credenciales. Esta función recibe parámetros como `'toEmail'`, `'toName'`, `'subject'`, `'message'` y `'fromName'`, los cuales personalizan el correo. De esta forma, los administradores podrán enviar un reporte del resumen diario de altas mediante un botón que abre un cuadro de diálogo donde se ingresan los detalles del correo y se carga automáticamente el resumen de las altas, incluyendo los pacientes dados de alta. Este enfoque permite enviar correos de manera eficiente y segura, mejorando la funcionalidad y la experiencia del usuario.

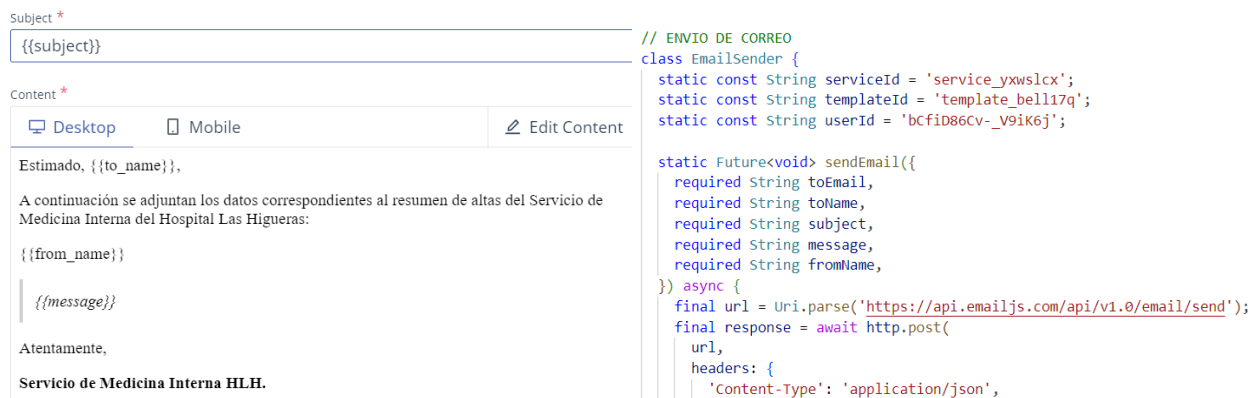
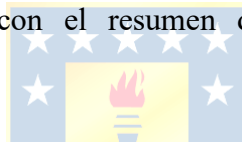


Figura 5.12 Integración de EmailJS en la aplicación de Flutter.

5. Pantalla de archivos de visitas

- **Estructura principal:** La estructura principal de la pantalla se define en la clase *'ArchivosVisitas'*, un *'StatefulWidget'* que permite mantener y actualizar el estado del widget de manera dinámica. La clase *'_ArchivosVisitasState'* contiene el estado del widget, que incluye la variable para almacenar el valor de búsqueda y una instancia de *Firestore* para interactuar con la base de datos *Firestore*.
- **Principales widgets y métodos:** El método de la Figura 5.13 *'_buildDataTable'*, crea una tabla utilizando el widget *'DataTable'* permitiendo la visualización de datos en una estructura de filas y columnas, y proporcionando funcionalidades de desplazamiento horizontal y vertical. Para la exportación de documentos PDF, se utiliza el archivo *'pdf_archivos.dart'*, que contiene las funciones *'generatePdf'* y *'generateResumenDiarioPdf'* para crear los documentos PDF correspondientes a las visitas y resúmenes diarios. El método *'_showResumenDiarioDialog'* se utiliza para mostrar un diálogo con el resumen diario de visitas, utilizando el widget *'ResumenDiarioTable'*.



```
Widget _buildDataTable(Map<String, Map<String, List<Map<String, dynamic>>>> visits) {
  List<String> dates = visits.keys.toList()..sort((a, b) => a.compareTo(b));
  return SingleChildScrollView(
    scrollDirection: Axis.vertical,
    child: SingleChildScrollView(
      scrollDirection: Axis.horizontal,
      child: DataTable(
        columns: const [
          DataColumn(label: Text('Fecha visita')),
          DataColumn(label: Text('EH1')),
          DataColumn(label: Text('EH2')),
          DataColumn(label: Text('EH3')),
          DataColumn(label: Text('PDF Visitas')),
          DataColumn(label: Text('Resumen Diario')),
          DataColumn(label: Text('PDF Resumen Diario')),
        ],
      ),
    ),
  );
}
```

Figura 5.13 Código utilizado para crear una tabla con el resumen diario.

6. Pantalla de historial pacientes por salas

La pantalla "Historial de Salas" proporciona a los usuarios una visión clara y detallada del historial de ocupación y cambios en las salas del servicio. Esta funcionalidad es esencial para la gestión eficiente de recursos y la planificación operativa de las salas que ocupa el SMI. La implementación se realiza en la clase *'HistorialSalasPage'*, un *'StatefulWidget'* que maneja el estado y los datos de la pantalla.

- **Principales métodos:** La clase ‘*HistorialSalasPageState*’ gestiona el estado de la página y los datos, llamando a la función que se muestra en la Figura 5.14, ‘*fetchHistorialSalas*’ para obtener datos de Firestore y agregando un ‘*listener*’ al controlador de búsqueda. La función ‘*fetchHistorialSalas*’ obtiene los datos de la colección ‘*historial_salas*’ de Firestore y los asigna a las listas ‘*historialSalas*’ y ‘*filteredHistorialSalas*’. La interfaz de usuario se construye utilizando ‘*Scaffold*’, incluyendo un ‘*AppBar*’, un ‘*TextField*’ para la búsqueda de salas, y un ‘*ListView.builder*’ que muestra los elementos filtrados.

```
Future<void> _fetchHistorialSalas() async {
  try {
    final snapshot = await FirebaseFirestore.instance.collection('historial_salas').get();
    setState(() {
      historialSalas = snapshot.docs.map((doc) => doc.data() as Map<String, dynamic>).toList();
      filteredHistorialSalas = historialSalas;
    });
  }
}
```

Figura 5.14 Función ‘*fetchHistorialSalas*’ que obtiene datos de Firestore.

7. Pantalla GRD

La pantalla de GRD muestra los diagnósticos más frecuentes del SMI y la estancia mediana de los pacientes, según el nivel de severidad y el rango de edad. Esta información sirve como referencia para que los médicos puedan considerar un tiempo probable de alta sugerido, alineado con el funcionamiento del hospital. Como se puede ver en la Figura 5.15 se consideraron los principales diagnósticos, los cuales comprenden el 21% del total de egresos, y cada uno de ellos superaba los 100 egresos. Inicialmente, se trabajó con el diagnóstico que tenía el mayor número de egresos correspondiente al diagnóstico N39.0 – Infección de vías urinarias, sitio no especificado.

Servicio Egreso	Diag 01 Principal (cod+des)	Egresos	% Egresos	Estancia Mediana	Estancia Mínima	Estancia Máxima		
20-110 - Medicina	N39.0 - Infección de vías urinarias, sitio no especificado	329	6.3%	5.0	0	89	21%	MAYORES DE 100 EGRESOS (25%)
	I50.0 - Insuficiencia cardíaca congestiva	299	5.7%	6.0	0	76		
	J18.8 - Otras neumonías, de microorganismo no especificado	178	3.4%	5.0	0	83		
	K92.2 - Hemorragia gastrointestinal, no especificada	151	2.9%	5.0	0	81		
	N17.8 - Otras insuficiencias renales agudas	146	2.8%	5.0	0	85		
	Z51.1 - Sesión de quimioterapia por tumor	118	2.2%	4.0	0	21		
	N18.5 - Enfermedad renal crónica, estadio 5	115	2.2%	9.0	1	51		
	N10 - Nefritis tubulointersticial aguda	99	1.9%	5.0	0	42		

Figura 5.15 Diagnósticos con mayor frecuencia en el SMI (datos obtenidos desde GRD del HLH).

A partir de este, se obtuvieron los datos de estancia mediana. Para realizar el análisis, se aplicaron los siguientes filtros: tipo de actividad es igual a hospitalización; hospital es igual a HLH de Talcahuano; servicio de egreso es igual a Medicina; versión CIE es igual a CIE-v2013; y el nivel de

severidad se consideró en tres categorías: menor, moderada y mayor. Estos filtros permitieron un análisis detallado de los datos, enfocándose en los egresos con mayor relevancia y proporcionando información precisa sobre las estancias medianas según los diagnósticos seleccionados. Los datos se extrajeron para los meses comprendidos entre enero de 2023 y diciembre de 2024. Además, se seleccionaron pacientes mayores de 15 años con un nivel de severidad menor, moderada y mayor, cuyo motivo de egreso fue el retorno al domicilio. Este enfoque permitió centrarse en las estancias hospitalarias que culminaron en una recuperación, ofreciendo así una perspectiva clara sobre los diagnósticos y las estancias medianas de los pacientes que se recuperaron satisfactoriamente. En la Figura 5.16 se muestra el resultado tras aplicar los filtros indicados para un nivel de severidad moderada.

Nivel de severidad	Servicio Egreso	Diag 01 Principal (cod+des)	Grupo Edad	Estancia Mediana	Estancia Mínima	Estancia Máxima
Moderada	20-110 - Medicina	N39.0 - Infección de vías urinarias, sitio no especificado	(1) > 75 a.	5.0	0	25
			(2) 65 - 74 a.	5.5	1	28
			(3) 45 - 64 a.	5.0	2	33
			(4) 15 - 44 a.	8.0	1	14

Figura 5.16 Resultados de egresos por diagnóstico de diagnóstico N39.0 de pacientes con severidad moderada.

- **Estructura principal:** La estructura principal de la pantalla se define en la clase `'GrdPage'`, que es un `'StatelessWidget'`. La clase `'_GrdPageState'` maneja el estado de la pantalla, incluyendo variables para almacenar los diagnósticos, los diagnósticos filtrados, el diagnóstico seleccionado y las severidades asociadas. Para obtener los diagnósticos desde Firestore y almacenarlos en el estado se utiliza el método `'_fetchDiagnosticos'` el cual filtra los diagnósticos en función del texto ingresado en el campo de búsqueda como se muestra en la Figura 5.17.

```
Future<void> _fetchDiagnosticos() async {
  try {
    final snapshot = await FirebaseFirestore.instance.collection('GRD').get();
    setState(() {
      diagnosticos = snapshot.docs.map((doc) => {
        'codigo': doc.id,
        'descripcion': doc.get('descripcion')
      }).toList();
      filteredDiagnosticos = diagnosticos;
    });
  }
}
```

Figura 5.17 Función `'_filterDiagnosticos'` que filtra los diagnósticos según el texto ingresado.

- **Principales widgets y métodos:** El método `'_fetchSeveridades'` obtiene los niveles de severidad para un diagnóstico seleccionado desde Firestore y los almacena en el estado, mostrando un

diálogo con los detalles de severidad. En la Figura 5.18 se puede ver el método ‘*_buildSeveridadesTable*’ que construye una tabla que muestra los niveles de severidad y la estadía mediana según el rango de edad, desplegándose en un cuadro de diálogo al seleccionar un diagnóstico.

```
Widget _buildSeveridadesTable() {
  if (severidades.isEmpty) {
    return const Center(child: Text('Selecciona un diagnóstico para ver detalles'));
  }

  List<String> ordenSeveridades = ['menor', 'moderada', 'mayor'];
  List<String> ordenGruposEdad = ['15-44', '45-64', '65-74', '>75'];
  List<TableRow> rows = [];

  for (var severidad in ordenSeveridades) {
    if (severidades.containsKey(severidad)) {
      Map<String, dynamic> data = severidades[severidad];
```

Figura 5.18 Parte del código que construye una tabla con los datos del diagnóstico GRD.

8. Pantalla de administración

La pantalla incluye un cuadro de búsqueda y se permite editar el nivel de permiso de los usuarios registrados, actualizar el acceso y/o eliminar un usuario. La implementación se realiza en la clase ‘*AdministrarUsuariosPage*’, un ‘*StatefulWidget*’ para manejar el estado y la dinámica de los datos en tiempo real.

- **Estructura principal:** En el método ‘*initState*’, se inicializa el estado, se llama a ‘*_loadUserData*’ para obtener datos de Firestore y la función *_saveUserData* guarda la información del usuario editado y se agrega un ‘*listener*’ al controlador de búsqueda. La función ‘*_loadUserData*’ obtiene los datos de la colección usuarios de Firestore y los asigna a las listas ‘*_usersData*’ y ‘*_filteredUsersData*’ donde se llama a la función ‘*_filterUsers*’ que filtra la lista de usuarios según el campo de búsqueda ingresado. Como se puede ver en la Figura 5.19.

```
// guarda los datos en el usuario que se edito informacion
void _saveUserData(int index) async {
  final userId = _filteredUsersData[index]['id'];

  if (userId != null) {
    await FirebaseFirestore.instance.collection('usuarios').doc(userId).update({
      'nivel_usuario': _filteredUsersData[index]['nivel_usuario'],
    });
  } else {
```

```
Future<void> _loadUserData() async {
  QuerySnapshot<Map<String, dynamic>> snapshot =
    await FirebaseFirestore.instance.collection('usuarios').get();

  List<Map<String, dynamic>> usersData = snapshot.docs.map((doc) {
    Map<String, dynamic> userData = doc.data();
    userData['id'] = doc.id;
    return userData;
  }).toList();
```

Figura 5.19 Código para guardar y cargar información de usuarios en Firebase Firestore.

- **Principales métodos utilizados:** La interfaz de usuario se construye utilizando ‘*Scaffold*’, incluyendo un ‘*AppBar*’, un ‘*TextField*’ para la búsqueda de usuarios, y un ‘*ListView.builder*’ que

muestra los elementos filtrados. El widget *'HorizontalDataTable'* se utiliza para mostrar una tabla de datos desplazable tanto horizontal como verticalmente. Luego, los títulos de las columnas se crean con *'_getTitleItemWidget'*, y las filas de la tabla se generan dinámicamente con *'_generateFirstColumnRow'* y *'_generateRightHandSideColumnRow'*, mostrando los datos de los usuarios y permitiendo la edición y eliminación. La funcionalidad para editar y eliminar usuarios se implementa con un diálogo de confirmación para la eliminación, asegurando que los datos del usuario se actualicen y/o eliminen tanto de Firebase Authentication como de Firestore.

9. Botón de cerrar sesión

En el método *'initState'* que se muestra en la Figura 5.6 se inicializan las opciones del menú y los elementos de la barra de navegación, incluyendo el botón de “Cerrar Sesión”. Luego, si el índice seleccionado corresponde a “Cerrar Sesión”, se realiza el cierre de sesión utilizando *'FirebaseAuth.instance.signOut()'* y es redirigido a la pantalla de inicio de sesión, asegurando que no puedan acceder a áreas restringidas sin autenticarse nuevamente.



Capítulo 6. Resultados

6.1. Introducción

En este capítulo se presentan los resultados obtenidos durante el desarrollo y la implementación de la aplicación web para la gestión de visitas clínicas y la administración de pacientes en el entorno hospitalario del SMI. Se detallan los hallazgos relevantes a partir de las entrevistas semiestructuradas y focus groups, la implementación técnica, las pruebas de la aplicación, y la retroalimentación recibida de los usuarios finales.

6.2. Resultados de las entrevistas y Focus groups

Para comprender las necesidades y expectativas de los usuarios, se realizaron entrevistas semiestructuradas y focus groups con diferentes profesionales involucrados en el proceso de visitas técnicas y gestión de pacientes. Las preguntas se centraron en los procesos actuales, los requisitos para la nueva aplicación, la seguridad de los datos, y las expectativas de resultados. En total participaron 29 profesionales, como se aprecia en la Figura C.1 del Anexo C.

6.2.1 Proceso actual de visitas clínicas y sus limitaciones

El proceso actual de visitas clínicas en el SMI del HLH se basa en hojas de cálculo para registrar datos clínicos. Cada equipo territorial, liderado por un coordinador, utiliza un portátil o tablet personal para registrar información durante las visitas. Al finalizar el día, el coordinador jefe descarga la planilla en PDF y envía un resumen de altas. Este método manual presenta limitaciones significativas en seguridad, estandarización y análisis de datos, dificultando la identificación de patrones y mejoras continuas. La generación manual de resúmenes y la conversión de planillas a PDF consume tiempo y es propensa a errores, y el sistema actual es inseguro, susceptible a la pérdida de datos y carece de medidas de seguridad robustas.

6.2.2 Hallazgos en focus groups y entrevistas

Los resultados revelaron varios puntos críticos en el proceso actual de registro de visitas clínicas: el acceso está limitado solo a los médicos hospitalistas coordinadores y a dos administradores, lo que restringe la visibilidad y participación del equipo; funciones críticas como la descarga de la planilla en PDF y el envío del reporte diario son frecuentemente olvidadas, interrumpiendo el registro de visitas diarias; la variable 'probable alta' se estima intuitivamente, lo que lleva a una falta de

estandarización en los tiempos de alta; la búsqueda de pacientes y el historial de una sala es complicada y los datos se copian manualmente, lo que debería ser automatizado; el uso inconsistente de códigos de colores para el tipo de salida del paciente puede causar confusiones; la planilla actual no permite identificar reingresos y no existe una trazabilidad clara de los cambios de sala; las visitas no siempre se realizan en orden y la actualización de cambios de sala no es simultánea, resultando en registros duplicados; además, el sistema actual carece de medidas de seguridad avanzadas, representando un riesgo significativo de acceso no autorizado o pérdida de datos. Estos puntos fueron cruciales para definir los requisitos funcionales y no funcionales de la nueva aplicación, priorizando las funcionalidades a implementar para abordar las limitaciones del proceso actual. pues se realizaron ajustes en la interfaz y en la funcionalidad conforme al feedback recibido.

6.2.3 Desarrollo de sprints

El desarrollo de la aplicación se organizó en sprints, cada uno con objetivos y entregables. Este se puede ver en la Figura E.1 del Anexo E, donde se detallan los objetivos, tareas y resultados de cada sprint.

6.3. Resultados de la aplicación web

En esta sección se presenta la versión inicial de la aplicación desarrollada, incluyendo capturas de pantalla que ilustran su interfaz y funcionalidades clave. Esta versión inicial representa el inicio de un proyecto más amplio. El nombre de la aplicación “VISIMED” es provisional y solo una idea.

6.3.1 Pantalla de inicio de sesión

En la Figura 6.1 se puede ver la pantalla de inicio de sesión, la cual permite a los usuarios ingresar al sistema de visitas clínicas del SMI del HLH. Esta incluye los siguientes elementos:

1. **Correo electrónico:** Campo para ingresar el correo electrónico del usuario, con validación de campo y mensajes de error en caso de fallos.
2. **Contraseña:** Campo para ingresar la contraseña del usuario con una opción para mostrar u ocultar la contraseña, también incluye validación de campo y mensajes de error.
3. **Iniciar sesión:** Botón para enviar las credenciales y acceder a la aplicación.
4. **Crear cuenta:** Enlace para registrar una nueva cuenta.
5. **Recuperar contraseña:** Enlace para recuperar la contraseña en caso de olvido.



Figura 6.1 Pantalla de inicio de sesión.

6.3.2 Pantalla de crear cuenta

La pantalla de creación de cuenta de la Figura 6.2 permite a los nuevos usuarios registrarse en el sistema. En esta se presenta campos para completar con sus datos e incluye validaciones para asegurar la correcta entrada de datos. Esta contiene:

← Crear cuenta

Nombre completo 1

Rut 2 Ya existe una cuenta con este RUT.

Correo Institucional 3 Ya existe una cuenta con este correo electrónico.

Contraseña 4

La contraseña debe tener al menos:

- ✗ 12 caracteres
- ✗ 1 letra mayúscula
- ✗ 1 letra minúscula
- ✗ 2 números
- ✗ 2 símbolos

Confirmar contraseña 5

Crear cuenta

Contraseña

La contraseña debe tener al menos:

- ✗ 12 caracteres
- ✓ 1 letra mayúscula
- ✓ 1 letra minúscula
- ✓ 2 números
- ✓ 2 símbolos

Confirmar contraseña

Las contraseñas no coinciden

mailply@base1-987be.firebaseio.com
para el +

Hola:

Visita este vínculo para verificar tu dirección de correo electrónico.

https://base1-987be.firebaseio.com/_/mailply@base1-987be.firebaseio.com/confirm-email?email=mailply@base1-987be.firebaseio.com&token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTk4ODQ4aWp0eS119

Si no solicitaste la verificación de esta dirección, ignora este correo electrónico.

Gracias.

El equipo de base1-987be

Se verificó tu correo electrónico

Ahora puedes acceder con tu cuenta nueva

Solicite acceso a su administrador.

Iniciar sesión

Figura 6.2 Pantalla de crear cuenta.

1. **Nombre completo:** Campo para ingresar el nombre completo del usuario, con validación y mensajes de error.
2. **RUT:** Campo para ingresar el RUT del usuario, con verificación de la existencia de cuentas con el mismo RUT.
3. **Correo Institucional:** Campo para ingresar el correo electrónico institucional del usuario, también con verificación de la existencia de cuentas con el mismo correo electrónico.
4. **Contraseña:** Campo para crear una contraseña, con requisitos de seguridad claramente indicados y validación.
5. **Confirmar contraseña:** Campo para confirmar la contraseña creada, con validación de correo electrónico para asegurar coincidencia.

Además, se proporciona retroalimentación inmediata sobre la fortaleza de la contraseña y la existencia de cuentas duplicadas. Una vez que la cuenta ha sido creada y el correo electrónico verificado, se debe esperar a que el administrador otorgue acceso a la aplicación según el nivel de acceso del usuario.

6.3.3 Pantalla de recuperar contraseña

La pantalla de recuperación de contraseña de la Figura 6.3, permite a los usuarios restablecer su contraseña a través de un correo de verificación y contiene:

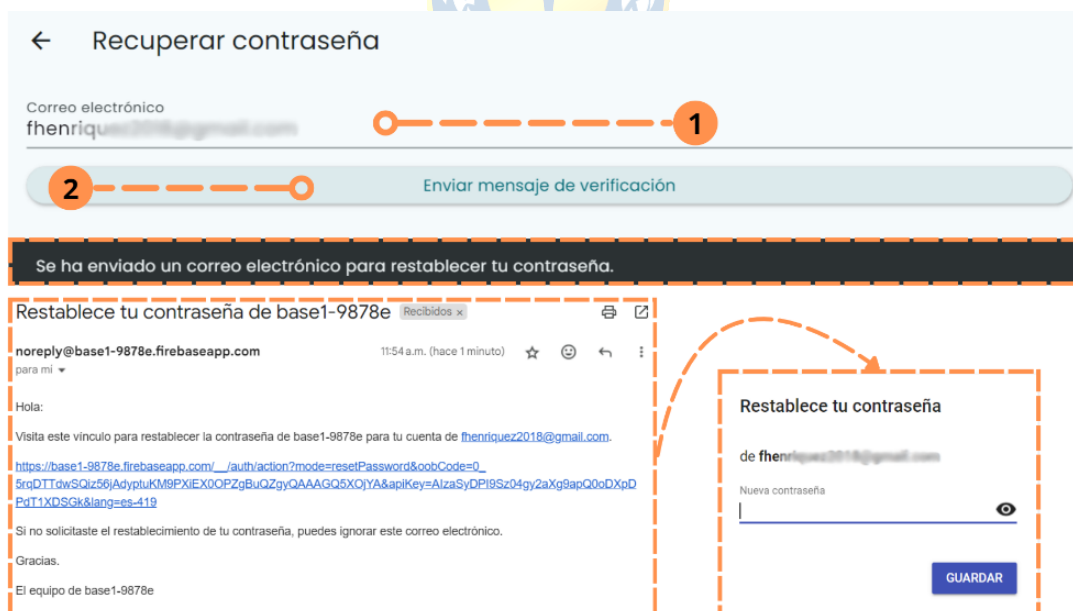


Figura 6.3 Pantalla de recuperar contraseña.

1. **Correo electrónico:** Campo para ingresar el correo electrónico del usuario, con validación y mensajes de error.

2. **Enviar mensaje de verificación:** Botón para enviar un correo electrónico con un enlace de restablecimiento.

Una vez enviado el correo, el usuario recibirá un enlace para restablecer su contraseña de manera segura.

6.3.4 Pantalla principal

La pantalla principal de la aplicación proporciona una interfaz intuitiva para la navegación y acceso a diferentes funcionalidades. Esta pantalla incluye dos botones principales como se muestra en la Figura 6.4:

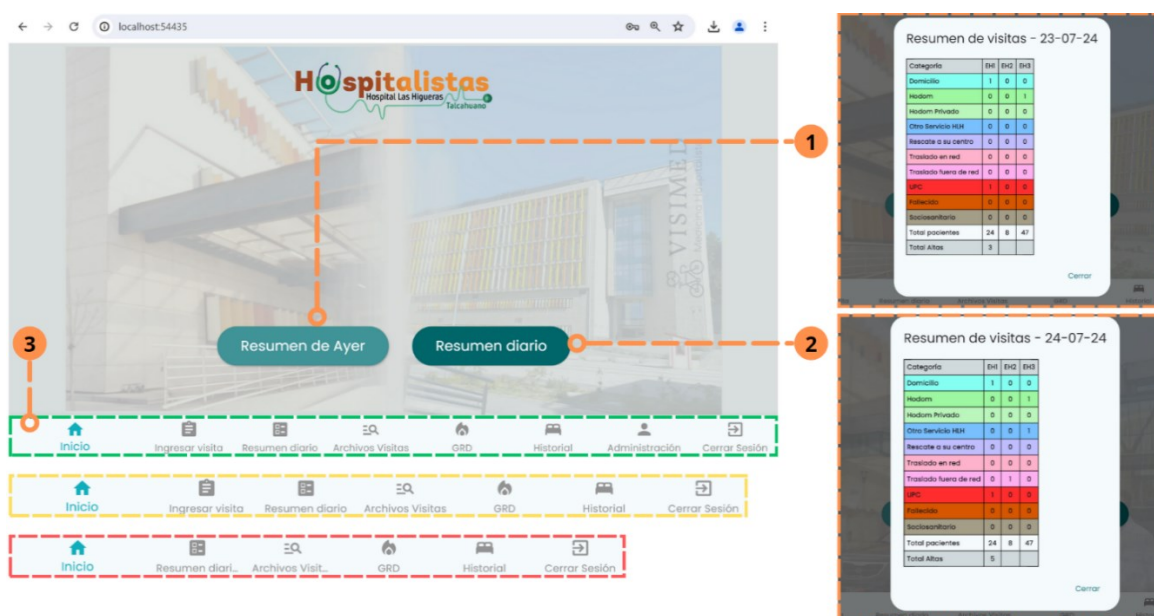


Figura 6.4 Pantalla principal.

1. **Resumen de Ayer:** Permite a los usuarios ver un resumen de las visitas realizadas el día anterior.
2. **Resumen Diario:** Muestra el resumen de las visitas del día actual.
3. **Barra de navegación:** La barra de navegación en la parte inferior permite a los usuarios navegar entre las diferentes secciones de la aplicación. La barra de navegación varía según el tipo de usuario:
 - **Verde (Administradores):** Disponible solo para administradores, incluye botones para 'Inicio', 'Ingresar visita', 'Resumen diario', 'Archivos Visitas', 'GRD', 'Historial', 'Administración' y 'Cerrar Sesión'.
 - **Amarillo (Editores):** Disponible para usuarios editores, incluye botones para 'Inicio', 'Ingresar visita', 'Resumen diario', 'Archivos Visitas', 'GRD', 'Historial' y 'Cerrar Sesión'.

- **Rojo (Visualizadores):** Disponible para usuarios visualizadores, incluye botones para 'Inicio', 'Resumen diario', 'Archivos Visitas', 'GRD', 'Historial' y 'Cerrar Sesión'.

6.3.5 Pantalla de ingresar visitas

La pantalla de ingreso a planillas de registro de visitas está diseñada para que los usuarios puedan registrar y visualizar las visitas diarias de pacientes según el equipo hospitalista correspondiente (EH1, EH2, EH3), como se muestra en la Figura 6.5. Estas se pueden desplazar vertical y horizontalmente para revisar y gestionar toda la información.

The screenshot displays the 'Hospitalistas' system interface for recording daily technical visits. The main menu includes buttons for 'Ingresar visita EH1', 'Ingresar visita EH2', and 'Ingresar visita EH3'. Three data tables are shown, each representing a different hospitalist team (EH1, EH2, EH3). Each table lists patient details such as RUT, Name, Surname, Age, Doctor, and Municipality. The 'Visitas EH1' table has a red header and one row highlighted in red. The 'Visitas EH2' table has a blue header and one row highlighted in green. The 'Visitas EH3' table has a pink header and one row highlighted in pink. Numbered callouts (1-7) point to various UI elements like the menu, buttons, and table headers.

Visitas EH1							
	RUT	Nombre	Apellidos	Edad	Médico	Comuna	Diag
UGA 1-1							
UGA 1-2	8.2	Marta	Me	82		San Pedro de la Paz	
UGA 1-3	8.0	Olga	Sa	84		Coronel	
UGA 1-4	8.	Rigoberto	Te	84		Talcahuano	
UGA 1-5	4.6	Ana	M.	85		Concepción	
UGA 1-6	6.	Juan	M.	85		Hualpén	
UGA 2-1	5.2	Rodelindo	C.	86		Penco	
UGA 2-2	9.	Olivia	M.	79		Concepción	
UGA 2-3	8.4	Matias	A.	75		Hualpén	
UGA 2-4	6.0	Ignacio	Lé	76		Talcahuano	

Visitas EH2							
	RUT	Nombre	Apellidos	Edad	Médico	Comuna	Diag
5-1	12	Rodrigo	Ti	51	D. Parra	Talcahuano	
5-2	13.2	Julio	Ig	49	D. Parra	Hualpén	
5-3	13.7	Rafael	Ni	50	D. Parra	Dichato	
5-4	14	Emily	C.	44			
5-5	16	Jennifer	Li	41	D. Parra	Talcahuano	
5-6	10	Camila	Ca	68	D. Parra	Talcahuano	
6-1	22	Linda	Co	15		Talcahuano	
6-2	12	Antonio	Bi	48		Talcahuano	
6-3	11	Alejandro	Fe	62		Talcahuano	

Visitas EH3						
	RUT	Nombre	Apellidos	Edad	Médico	Comuna
cirugia 401-1	6.89E	Ricardo	Si	83		Concepción
cirugia 401-2	8.2	Graciela	Lu	75		Concepción
cirugia 401-3	16	Alicia	Me	42		Talcahuano
cirugia 401-4	9.	Éster	NU	73		Hualpén
cirugia 401-5	13.5	Floribella	Re	47		Talcahuano
cirugia 402-1	13.82	Tomas	Go	48		Concepción
cirugia 402-2	9.06	Fernando	Pa	47		Concepción
cirugia 402-3	14.5	Tomasa	Alc	51		Talcahuano
cirugia 402-4	19.9	Victor	Ca	35		Talcahuano
cirugia 402-5	8.6	Julia	Fl	83		Hualpén

Figura 6.5 Pantalla de ingreso a planillas de registro de visitas.

La pantalla principal presenta botones claramente diferenciados para ingresar visitas de los tres equipos hospitalistas: EH1, EH2, y EH3. Cada equipo tiene su propia tabla de registros de visitas.

1. **Botón ingreso a planilla EH1:** Permite a los usuarios registrar las visitas diarias del equipo hospitalista EH1. Cada fila en la planilla muestra detalles importantes del paciente, como RUT, nombre, apellidos, edad, médico encargado, comuna y diagnóstico.
2. **Botón ingreso a planilla EH2:** Similar a la anterior, esta planilla está destinada al equipo hospitalista EH2, con la misma disposición y campos de información.
3. **Botón ingreso a planilla EH3:** Finalmente, esta planilla está destinada al equipo hospitalista EH3.

4. **Botón de regreso:** Botón que permite regresar a la pantalla de ingreso a planillas por equipo.
5. **Indicador de sala:** Indicador circular que cambia de estado a verde cuando la cama está disponible y a rojo cuando está ocupada
6. **Gráfico de progreso:** Gráfico que indica el progreso de la visita. A medida que se completan los datos de la visita, el gráfico avanza hasta llegar al 100% al finalizar todos los datos.
7. **Botón de actualización:** Este botón permite manualmente actualizar la planilla de ser necesario.

Las planillas (ver el Anexo G. Resultados adicionales de la aplicación) cargan automáticamente las salas/camas de cada planilla. Al día siguiente los datos se cargan automáticamente con los pacientes hospitalizados para facilitar el seguimiento y solo se limpian los campos que cambian continuamente y se vuelven a ingresar en cada visita. La columna acciones de cada planilla permite realizar varias acciones importantes, como ingresar un nuevo episodio de hospitalización, validar el RUT del paciente, ingresar nombre y apellido, cerrar el episodio una vez que el paciente se dé de alta, copiar los datos del paciente en formato de lista y mover al paciente a otra sala seleccionando y confirmando la sala de traslado. Los usuarios pueden ingresar a un paciente, siempre y cuando la sala esté desocupada y se ingrese correctamente el RUT del paciente y otros campos obligatorios como el nombre, apellidos, fecha ingreso HLH y fecha ingreso MIH. Luego, para ingresar al paciente se debe seleccionar el botón de ingresar nuevo episodio y una vez sea dado de alta, se debe seleccionar el botón que permite cerrar el episodio de hospitalización. Las filas contienen campos que permiten el ingreso de texto libre (diagnósticos, planes, gestiones pendientes), mientras que otras tienen opciones predefinidas de selección (condición, tipo de salida, probable alta). Al seleccionar un tipo de salida, la fila del paciente se colorea de acuerdo con la opción seleccionada. Luego, para ingresar las fechas de ingreso al seleccionar la fecha, se despliega un calendario que permite el ingreso de la fecha. Al ingresar la fecha, se calculan automáticamente los días de estadía de MIH y HLH, donde se colorea el número de días de acuerdo con la estancia (estancias mayores a 7 días se colorean en rojo, mientras que las menores a 7 están en verde). Finalmente, para guardar los datos se debe presionar el botón guardar.

6.3.6 Resumen diario

La pantalla de resumen diario proporciona una visión consolidada de las visitas clínicas realizadas durante el día. Como se puede ver en la Figura 6.6, incluye una tabla que representa los datos de

manera clara y concisa, facilitando la revisión y el análisis de los datos. Presenta los datos agrupados por categorías, mostrando los datos correspondientes a los pacientes según los equipos hospitalistas (EH1, EH2, EH3) y el total. Las categorías incluyen los tipos de salida, se muestra la cantidad total de pacientes por equipo y en total. También se indican las altas de cada equipo y el total. Se calcula el porcentaje de altas según el total de pacientes hospitalizados y los que tienen un tipo de salida. Este porcentaje se presenta claramente para una rápida visión del desempeño del día.

The image displays three overlapping screenshots of a mobile application interface for a daily summary report.

Top Left Screenshot: Shows a summary table with the following data:

Categoría	EH1	EH2	EH3	Total
Domicilio	1	1	0	2
Hodam	0	2	0	2
Hodam Privado	0	0	1	1
Otra Servicio HLH	0	2	0	2
Rescate a su centro	0	0	1	1
Trasladado en red	0	0	1	1
Trasladado fuera de red	0	1	1	2
UPC	1	0	0	1
Salvada	0	0	0	0
Sociosanitario	0	2	1	3
Total pacientes	27	11	40	78
Altas por equipo	2	8	5	15
% altas	3%	10%	6%	19%

Top Right Screenshot: Shows a form to send the report via email. A confirmation message at the bottom reads: "Envío de correo realizado".

Bottom Left Screenshot: Shows a "Detalles de Pacientes" table with the following data:

Nombre	Apellidos	RUT	Salida
Alejandro	Fe	11.	6-3
Rodrigo	Tc	12.3.	5-1

Bottom Right Screenshot: Shows an email received from "fhenriq...@gmail.com" with the subject "Reporte de resumen diario de altas del Servicio de Medicina Interna". The email body includes a summary of the report and a detailed list of patients:

Detalles de pacientes:		RUT: 9.168
Tipo de Salida: Domicilio	Nombre: Rosa	Tipo de Salida: Rescate a su centro
RUT: 13.6	Nombre: Ester N	RUT: 9.50
Tipo de Salida: UPC	Nombre: Olga	
RUT: 8.5		
Tipo de Salida: Sociosanitario		

Figura 6.6 Pantalla de resumen diario.

1. **Envío de resumen diario por correo:** La aplicación permite enviar el resumen diario por correo electrónico, funcionalidad disponible para el usuario administrador. Los usuarios pueden especificar la dirección de correo electrónico y el nombre del destinatario en un formulario dedicado. Al enviar el correo, se muestra una confirmación de envío exitoso.
2. **Tabla con detalle:** Al presionar sobre el número en la tabla, se despliega una tabla con el detalle de los pacientes que tienen dicho tipo de salida, incluyendo nombre, apellidos, RUT y sala.

4. **Tabla con resumen diario:** Los usuarios pueden ver el resumen diario directamente en la aplicación antes de proceder a la exportación, asegurándose de que todos los datos están correctamente presentados.
5. **Resumen diario en PDF:** Además de las visitas individuales, los usuarios pueden exportar un resumen diario consolidado en formato PDF. Este incluye un desglose por categorías, equipos y mostrando el número de altas.

6.3.8 Pantalla GRD

La pantalla GRD de la Figura 6.8, muestra información con los diagnósticos y estadías medianas de pacientes utilizando información obtenida de la unidad de GRD del hospital.

The screenshot shows the GRD application interface. At the top, there is a search bar labeled 'Buscar Diagnóstico' with the text 'infe' entered. Below the search bar is a table of diagnostic codes and descriptions. The table has two columns: 'Código diagnóstico' and 'Descripción'. The visible rows are:

Código diagnóstico	Descripción
I50.0	Insuficiencia cardíaca congestiva
J18.8	Otras neumonías, de microorganismo no especificado
K92.2	Hemorragia gastrointestinal, no especificada
N17.8	Otras insuficiencias renales agudas
N18.5	Enfermedad renal crónica, estadio 5
N39.0	Infección de vías urinarias, sitio no especificado
Z51.1	Sesión de quimioterapia por tumor

Callout 1 points to the search bar. Callout 2 points to a detailed table for diagnosis N39.0. The detailed table is titled 'Detalles para N39.0' and has three columns: 'Severidad', 'Rangos de edad (años)', and 'Estancia mediana (días)'. The data is as follows:

Severidad	Rangos de edad (años)	Estancia mediana (días)
menor	15-44	2.5
	45-64	3.0
	65-74	3.5
	>75	3.0
moderado	15-44	8.0
	45-64	5.0
	65-74	5.5
	>75	5.0
mayor	15-44	12.0
	45-64	6.0
	65-74	5.0
	>75	5.0

The bottom of the screen shows a navigation bar with icons for Inicio, Ingresar visita, Resumen día..., Archivos Visit..., GRD, Historial, Administraci..., and Cerrar Sesión.

Figura 6.8 Pantalla con información GRD.

1. **Búsqueda de diagnósticos:** Los usuarios pueden buscar diagnósticos específicos utilizando el campo de búsqueda. Al ingresar un término en el campo de búsqueda, se filtran los diagnósticos que coinciden con el término ingresado, mostrando tanto el código del diagnóstico como su descripción. Por ejemplo, al buscar "infe", se muestra "N39.0 Infección de vías urinarias, sitio no especificado".
2. **Tabla de detalles:** Al seleccionar un diagnóstico específico de la lista, se despliega una tabla con los detalles asociados a ese diagnóstico. La tabla incluye información sobre la severidad,

rangos de edad, y la estancia mediana en días, utilizando un código de colores para indicar la severidad (menor, moderado, mayor), facilitando la interpretación rápida de la información.

6.3.9 Historial de salas

La pantalla historial de salas muestra el historial de ocupación de las salas del hospital, permitiendo a los usuarios ver cambios en la ocupación por fecha y paciente, como se puede ver en la Figura 6.9.

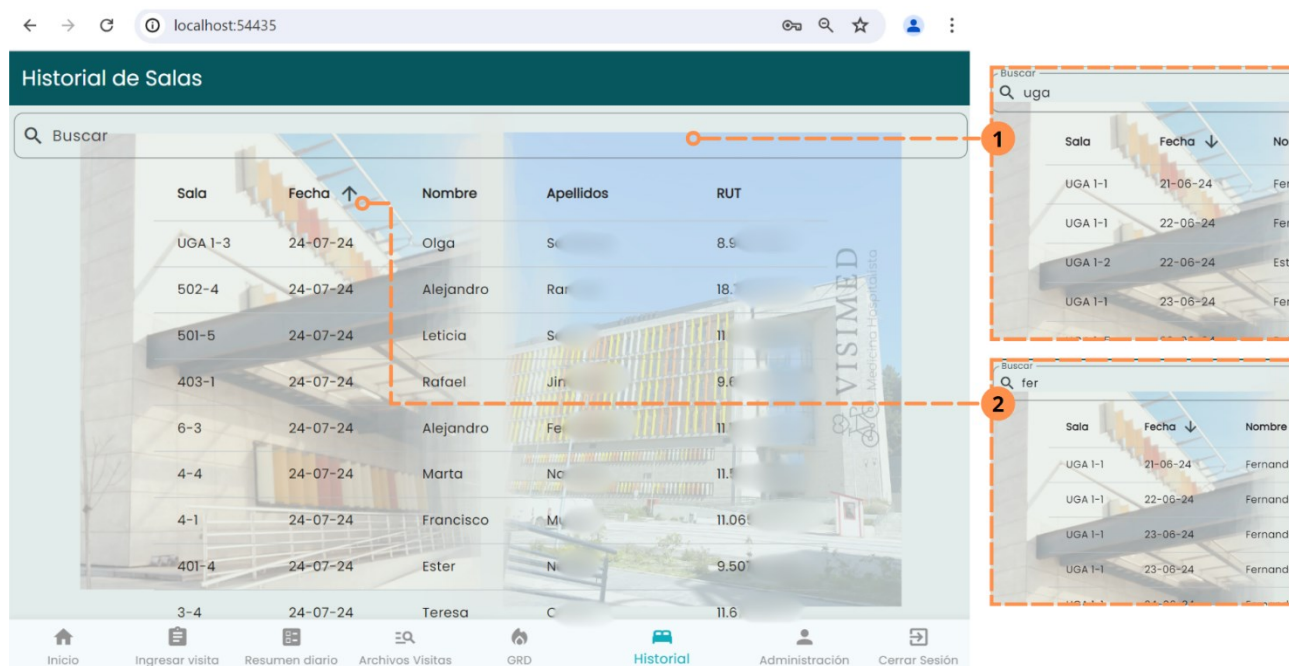


Figura 6.9 Pantalla de historial de salas.

1. **Campo de búsqueda:** Permite buscar información específica sobre la ocupación de las salas. Actualiza resultados en tiempo real mientras se ingresa el término de búsqueda. Por ejemplo, al buscar "uga", se muestran todas las entradas correspondientes a las salas UGA.
2. **Tabla de historial:** Muestra información sobre sala, fecha, nombre y apellido del paciente, y su RUT. Permite ordenar la información por fecha, proporcionando una visión clara y cronológica del historial de ocupación.
3. **Detalles de la sala:** Se proporciona información detallada de cada paciente en una sala específica. Permite a los usuarios ver rápidamente los datos relevantes, facilitando la gestión de recursos.

4. **Filtrado de resultados:** Filtra la información basada en la búsqueda por sala o nombre del paciente. Mejora la eficiencia en la identificación de ocupaciones específicas y la planificación de recursos hospitalarios.

6.3.10 Administración de usuarios

La pantalla de administración de usuarios de la Figura 6.10, permite gestionar los perfiles de usuario y sus permisos, donde se incluye opciones para agregar, editar y eliminar usuarios. En ella se muestra una tabla con información sobre nombre, correo electrónico y permisos de cada usuario.



Figura 6.10 Pantalla de administración de usuarios.

1. **Campo de Búsqueda:** Permite buscar usuarios por nombre. Filtra y actualiza los resultados en tiempo real mientras se ingresa el término de búsqueda.
2. **Columna de acciones:** Contiene los botones editar o eliminar usuarios.
3. **Editar permisos:** Permite cambiar los permisos de un usuario entre las opciones de Visualizar, Editar y Administrador.
4. **Eliminación de usuario:** Se permite eliminar un usuario y se muestra un cuadro de confirmación para confirmar si está seguro de eliminar o no al usuario, lo cual lo protege contra eliminaciones accidentales.

6.4. Pruebas de funcionalidad

Para asegurar la funcionalidad de la aplicación, se realizaron pruebas con datos anonimizados de 100 pacientes. Estas pruebas incluyeron el ingreso de visitas diarias para cada paciente, la

generación y visualización de resúmenes diarios, y la exportación de datos en formato PDF. También se probó la búsqueda y visualización de archivos de visitas, la visualización de diagnósticos y estadías medianas, así como la gestión de usuarios y permisos. Se llevaron a cabo pruebas para verificar la operatividad de todas las funcionalidades de la aplicación. Los datos se ingresaron y visualizaron adecuadamente, y la exportación a PDF funcionó sin problemas. La búsqueda y visualización de archivos de visitas fue eficiente y rápida al inicio, pero a medida que se fueron cargando más y más datos, la aplicación comenzó a ralentizarse, afectando el rendimiento. La información de diagnósticos y estadías medianas se presentó de manera clara y útil para los usuarios. Además, la gestión de usuarios permitió agregar, editar y eliminar perfiles de manera correcta.

6.4.1 Resultados de checklist de verificación de requerimientos

Para validar la aplicación, se distribuyó un checklist de verificación de requerimientos (ver Anexo G) a los usuarios finales, entre los cuales se incluyeron usuarios editores, administradores y visualizadores. A pesar de que inicialmente participaron 29 usuarios en los focus groups y entrevistas semiestructuradas, solo 19 completaron el checklist final. Durante una demostración del funcionamiento de la aplicación, se aplicó este checklist junto con preguntas clave sobre la integración de funcionalidades, la utilidad en visitas técnicas, la complejidad de uso, la necesidad de apoyo técnico y sugerencias de mejoras para futuras versiones.

El checklist abarcó aspectos como la gestión de perfiles, gestión de pacientes, registro y consulta de visitas, registro de hospitalización, seguridad y privacidad, interfaz de usuario, y soporte y funcionalidades adicionales. Los resultados mostraron un cumplimiento total del 100%, indicando que todos los usuarios que completaron el checklist consideraron que se cumplían todos los requerimientos funcionales, seleccionando "SI" en cada ítem.

Capítulo 7. Conclusiones

7.1. Discusión

El desarrollo de la aplicación para la gestión de visitas técnicas a pacientes en el SMI del HLH ha sido un proceso detallado y multifacético. Este proyecto fue aprobado por la CEC del SST, garantizando que todas las actividades se realizaron cumpliendo con los estándares éticos y legales aplicables. La evaluación exhaustiva del proceso actual y sus limitaciones, junto con la implementación de una solución tecnológica pueden mejorar significativamente la eficiencia y seguridad del registro de datos clínicos.

Para comprender completamente el proceso de visitas clínicas, se realizó una inmersión en las actividades diarias del personal del SMI, acompañando a los profesionales durante las visitas técnicas. Esto permitió obtener una visión integral de las dificultades y necesidades que enfrentan los médicos en su trabajo cotidiano. El proceso actual de visitas clínicas en el SMI del HLH se basa en el uso de hojas de cálculo para registrar datos clínicos, un método que presenta numerosas limitaciones. La falta de estandarización, la propensión a errores manuales, y la inseguridad en el manejo de datos son algunos de los problemas identificados. Además, el proceso manual de generación y conversión de resúmenes a PDF es ineficiente y consume tiempo. La falta de visibilidad y participación del equipo en el acceso a la planilla, la estimación intuitiva de la “probable alta”, y la dificultad en la búsqueda y gestión de pacientes y salas complican aún más la gestión diaria de visitas clínicas.

El desarrollo de la aplicación se organizó en sprints, lo que permitió una entrega continua de mejoras y funcionalidades basadas en la retroalimentación directa de los usuarios finales. El enfoque ágil de Scrum facilitó la adaptación a los cambios y la mejora constante del producto, asegurando que la aplicación respondiera a las necesidades reales de los usuarios. Por otra parte, la integración con Firebase garantiza la autenticación y sincronización de datos en tiempo real, asegurando la disponibilidad y consistencia de la información.

Las pruebas funcionales y de usabilidad, realizadas con datos anonimizados de 100 pacientes y mediante la aplicación de un checklist, confirmaron que la aplicación cumple con todos los requisitos funcionales y es fácil de usar. Sin embargo, se observó que el rendimiento de la aplicación disminuía a medida que aumentaba la cantidad de datos, lo que indica la necesidad de optimización adicional. Pese a lo anterior, los usuarios finales valoraron positivamente la aplicación, destacando su potencial para mejorar no solo la gestión de altas, sino también la entrega de turno del servicio, un aspecto crítico en la calidad de los servicios hospitalarios. Este último es un indicador de calidad y

uno de los elementos medibles en la pauta de cotejo utilizada para la evaluación del proceso de acreditación de instituciones de atención cerrada.

7.2. Conclusiones

El desarrollo de la aplicación utilizando la metodología Scrum logró cumplir con los objetivos iniciales, proporcionando una herramienta adaptable y eficiente para la gestión hospitalaria. La colaboración constante con el personal del hospital y la implementación de sus comentarios fueron claves para el éxito del proyecto. Los principales logros incluyen la simplificación del proceso de envío de correos electrónicos mediante EmailJS, la implementación de un sistema de autenticación seguro, el desarrollo de interfaces intuitivas para el ingreso y visualización de datos, y la integración efectiva con Firebase para la autenticación y almacenamiento en tiempo real. La aplicación ha demostrado ser funcional y fácil de usar, y ha sido bien valorada por los usuarios finales, quienes destacan su potencial para mejorar significativamente la gestión de altas y la entrega de turno oficial. Sin embargo, aunque la aplicación cumple con todos los requerimientos funcionales, uno de los requisitos importantes es el rendimiento. A medida que se cargaban más datos, la aplicación comenzó a ralentizarse, indicando la necesidad de optimización adicional. Esto se debe en gran medida al procesamiento de datos. Algunas operaciones toman mucho tiempo o consumen muchos recursos, lo que puede ralentizar todo el sistema. Además, las consultas a la base de datos no están optimizadas, por lo que es esencial mejorar estas consultas para garantizar una experiencia de usuario fluida y eficiente. Por lo tanto, es crucial considerar mejoras en el rendimiento.

7.3. Trabajo Futuro

Para continuar mejorando la aplicación, se recomienda desarrollar funcionalidades que permitan a los usuarios personalizar sus informes y resúmenes según sus necesidades específicas. Además, es crucial implementar mejoras para optimizar la velocidad de carga de datos y la eficiencia general de la aplicación, especialmente a medida que aumenta la cantidad de datos manejados. También es importante incorporar más diagnósticos en el sistema para ampliar su utilidad y contribuir a la estandarización de diagnósticos, mejorando la calidad y coherencia de la gestión hospitalaria.

Otra área clave de mejora es la actualización y guardado de datos. Se sugiere implementar un sistema más robusto para la sincronización de datos en tiempo real, asegurando que los cambios realizados por los usuarios se guarden y reflejen inmediatamente en la base de datos. Esta mejora no

solo garantizará la integridad y consistencia de la información, sino que también optimizará el flujo de trabajo de los usuarios, permitiendo una experiencia más fluida y eficiente

Mantener la metodología Scrum para futuros desarrollos es esencial ya que permite el desarrollo iterativo donde se debe tener una comunicación constante con los usuarios finales para garantizar que la aplicación evolucione conforme a sus necesidades y feedback. Por otra parte, la incorporación de registros ordenados y seguros facilita la investigación, permitiendo realizar estudios que generen resultados significativos en el campo de la medicina interna y para el hospital. También se podría considerar la posibilidad de conectar la aplicación con el servicio de TrakCare, el sistema de registro clínico electrónico utilizado en el hospital, con el fin de mejorar la integración y flujo de información entre ambos sistemas y así potenciar aún más la utilidad de la aplicación.

Finalmente, sería importante a futuro añadir una sección educativa a la aplicación, permitiendo a los médicos internos y estudiantes absorber conocimientos a través de esta plataforma, acercándolos a los casos de los pacientes y a la realidad del hospital.



Capítulo 8. Glosario

HLH	: Hospital Las Higueras.
SMI	: Servicio de Medicina Interna.
MINSAL	: Ministerio de Salud.
GRD	: Grupo de Relacionados por el Diagnóstico.
CIE-10	: Clasificación Estadística Internacional de enfermedades.
MIH	: Medicina Interna Hospitalista.
EH1	: Equipo hospitalista 1.
EH2	: Equipo hospitalista 2.
EH3	: Equipo hospitalista 3.



Capítulo 9. Referencias

- [1] “Especificación y límites del modelo de datos - Soporte técnico de Microsoft”. Microsoft Support. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://support.microsoft.com/es-es/office/especificación-y-límites-del-modelo-de-datos-19aa79f8-e6e8-45a8-9be2-b58778fd68ef>.
- [2] “Sistemas de información para la salud”. PAHO/WHO | Pan American Health Organization. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://www.paho.org/es/temas/sistemas-informacion-para-salud>.
- [3] J. M. Rosa y E. L. Frutos, “Ciencia de datos en salud: desafíos y oportunidades en América Latina”, *Rev. Medica Clin. Condes*, vol. 33, n.º 6, pp. 591–597, noviembre de 2022. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.rmclc.2022.09.007>.
- [4] J. E. Guevara Camarena, “Importancia de los registros, la estadística y los sistemas de información para la gerencia de los servicios de salud”, *RS*, n.º 4, pp. 10–30, febrero de 2022.
- [5] “AMA digital health care 2022 study findings”. American Medical Association. Accedido el 21 de abril de 2024. [En línea]. Disponible: <https://www.ama-assn.org/about/research/ama-digital-health-care-2022-study-findings>.
- [6] Subsecretaria de Redes Asistenciales, “ORD N°432 Estrategias hacia eficiencia clínica hospitalaria desde la Unidad de Emergencias”, Santiago, 2024.
- [7] G. Eymin y A. K. Jaffer, “Medicina hospitalaria a 15 años de su implementación: ¿Cuáles son los potenciales beneficios en nuestro medio?”, *Rev. Medica Chile*, vol. 141, n.º 3, pp. 353–360, marzo de 2013. Accedido el 2 de marzo de 2024. [En línea]. Disponible: <https://doi.org/10.4067/s0034-98872013000300011>.
- [8] J. A. Norambuena Pape. “Evaluación del modelo hospitalista en una clínica de Chile mediante simulación - ProQuest”. ProQuest | Better research, better learning, better insights. Accedido el 15 de marzo de 2024. [En línea]. Disponible: <https://www.proquest.com/docview/2552114595?%20Theses&fromopenview=true&pq-origsite=gscholar&sourcetype=Dissertations%20>.
- [9] “El rol del Hospitalista en el Sistema de Salud”. Escuela de Medicina. Accedido el 20 de marzo de 2024. [En línea]. Disponible: <https://medicina.uc.cl/publicacion/rol-del-hospitalista-sistema-salud/>.
- [10] “Medicina Interna Hospitalaria”. Escuela de Medicina. Accedido el 26 de marzo de 2024. [En línea]. Disponible: <https://medicina.uc.cl/divisiones/medicina/medicina-interna-2/medicina-interna-hospitalaria/>.
- [11] M. Z. M., “Importancia del sistema grd para alcanzar la eficiencia hospitalaria”, *Rev. Medica Clin. Condes*, vol. 29, n.º 3, pp. 347–352, mayo de 2018. Accedido el 1 de abril de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.rmclc.2018.04.010>.
- [12] “Bases grd”. Fonasa Chile 2024. Accedido el 2 de abril de 2024. [En línea]. Disponible: <https://www.fonasa.cl/sites/fonasa/datos-abiertos/bases-grd>.
- [13] D. Sugg Herrera. “Factores que inciden en la duración de la estancia hospitalaria en Chile”. e-Repository UPF. Accedido el 23 de abril de 2024. [En línea]. Disponible: https://repositori.upf.edu/bitstream/handle/10230/55092/Daniela%20Sugg_TFM.pdf?sequence=1&isAllowed=y.

- [14] M. Emes, S. Smith, S. Ward y A. Smith, "Improving the patient discharge process: implementing actions derived from a soft systems methodology study", *Health Syst.*, vol. 8, n.º 2, pp. 117–133, octubre de 2018. Accedido el 18 de febrero de 2024. [En línea]. Disponible: <https://doi.org/10.1080/20476965.2018.1524405>.
- [15] A. Tyler, A. Boyer, S. Martin, J. Neiman, L. A. Bakel y M. Brittan, "Development of a discharge readiness report within the electronic health record-A discharge planning tool", *J. Hospital Medicine*, vol. 9, n.º 8, pp. 533–539, mayo de 2014. Accedido el 20 de marzo de 2024. [En línea]. Disponible: <https://doi.org/10.1002/jhm.2212>.
- [16] C. M. Villa Chicaiza, "Diseño de una red de datos inalámbrica para la mejora de la atención médica en el Hospital Básico de Chone," Tesis de Grado, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador, 2021. Accedido el 18 de marzo de 2024. [En línea]. Disponible: <https://repositorio.espe.edu.ec/bitstream/21000/24950/1/M-ESPEL-sit-0095.pdf>.
- [17] R. A. Lovon Leng y M. A. Salazar Morales, "Sistema multiplataforma para la atención de los pacientes hospitalizados en el Hospital P.N.P.," Tesis de Grado, Universidad Ricardo Palma, Lima, Perú, 2022. Accedido el 12 de marzo de 2024. [En línea]. Disponible: <https://hdl.handle.net/20.500.14138/6138>.
- [18] L. Ochoa Chaves, O. Jiménez Alvarado, y F. Martínez de Lemos, "Expediente digital único en salud (EDUS) de Costa Rica: buenas prácticas, historia e implementación". Accedido el 12 de marzo de 2024. [En línea]. Disponible : https://www.academia.edu/109396085/Expediente_digital_%C3%BAnico_en_salud_EDUS_de_Costa_Rica_buenas_pr%C3%A1cticas_historia_e_implementaci%C3%B3n?uc-sb-sw=44961669.
- [19] S. Luján Mora, "Programación de aplicaciones web: historia, principios básicos y clientes web," Universidad de Alicante, Alicante, España, 2002. Accedido el 29 de marzo de 2024. [En línea]. Disponible: https://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf.
- [20] Y. L. Guerra Machado, F. G. Fabregat-Sanjuan, y M. J. García Gordo, "Modelo predictivo de eficiencia energética en instalaciones de bombeo de acueductos," *Revista Universidad y Sociedad*, vol. 14, no. 3, pp. 487-492, 2022. Accedido el 17 de marzo de 2024. [En línea]. Disponible: <https://rus.ucf.edu.cu/index.php/rus/article/view/2889/2842>.
- [21] Comisión Nacional de Productividad, "Calidad de los datos públicos". Accedido el 20 de marzo de 2024. [En línea]. Disponible: <https://cnep.cl/wp-content/uploads/2023/10/Calidad-datos-publicos-v-2.pdf>.
- [22] J. E. Moros Ruiz, "Implementación de un sistema de gestión documental basado en BPM para mejorar los procesos en una empresa de servicios," Tesis de Grado, Universidad del Norte, Barranquilla, Colombia, 2015. Accedido el 15 de marzo de 2024. [En línea]. Disponible: <http://hdl.handle.net/11371/1163>.
- [23] E. X. Méndez Balcázar, "Modelo de simulación de la red vial rural de Colombia utilizando lógica difusa," Universidad de los Andes, Bogotá, Colombia, 2013. Accedido el 19 de marzo de 2024. [En línea]. Disponible: <https://core.ac.uk/download/pdf/47264961.pdf>.
- [24] Microsoft, "Visual Studio". Accedido el 27 de marzo de 2024. [En línea]. Disponible: <https://visualstudio.microsoft.com/es/>.
- [25] J. Singh, S. Srivastva, D. Raj, S. Singh, y M. J. Rasool, "Flutter and Firebase: Making Cross-Platform Application," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 4, pp. 1819-1827, Apr. 2022. Accedido el 26 de marzo de

2024. [En línea]. Disponible: https://www.irjmet.com/uploadedfiles/paper/issue_4_april_2022/21281/final/fin_irjmet1651217656.pdf.
- [26] Google, "Firebase Realtime Database". Accedido el 16 de abril de 2024. [En línea]. Disponible: <https://firebase.google.com/products/realtime-database?hl=es419#:~:text=Firebase%20Realtime%20Database%20es%20una,tus%20usuarios%20en%20tiempo%20real>.
- [27] S. M. Velásquez, J. D. Vahos Montoya, M. E. Gómez Adasme, E. J. Restrepo Zapata, A. A. Pino, y S. Londoño Marín, "Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software," *Revista Cintex*, vol. 24, no. 2, pp. 13-23, Dic. 2019. Accedido el 24 de abril de 2024. [En línea]. Disponible: <https://revistas.pascualbravo.edu.co/index.php/cintex/article/view/334>.
- [28] A. Srivastava, S. Bhardwaj, y S. Saraswat, "Scrum Model for Agile Methodology," en 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 864-869. Accedido el 17 de abril de 2024. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8229928>.
- [29] Eniun, "Metodología Scrum: Desarrollo de Software Ágil". Accedido el 26 de febrero de 2024 [En línea]. Disponible: <https://www.eniun.com/metodologia-scrum-desarrollo-software-agil/>.
- [30] R. K. Merton, "The Focussed Interview and Focus Groups: Continuities and Discontinuities," *The Public Opinion Quarterly*, vol. 51, no. 4, pp. 550-566, 1987. Accedido el 25 de febrero de 2024. [En línea]. Disponible: <http://www.jstor.org/stable/2749327>.
- [31] EmailJS Pte Ltd, "EmailJS". Accedido el 5 de junio de 2024. [En línea]. Disponible: <https://www.emailjs.com/>.
- [32] Médicos Hospitalistas Servicio de Medicina Interna, Hospital Las Higueras, «Planilla Excel Visita Técnica Equipo Hospitalista 1-2,» Talcahuano, 2023.

Anexo A. Procedimiento general siguiendo una metodología Scrum

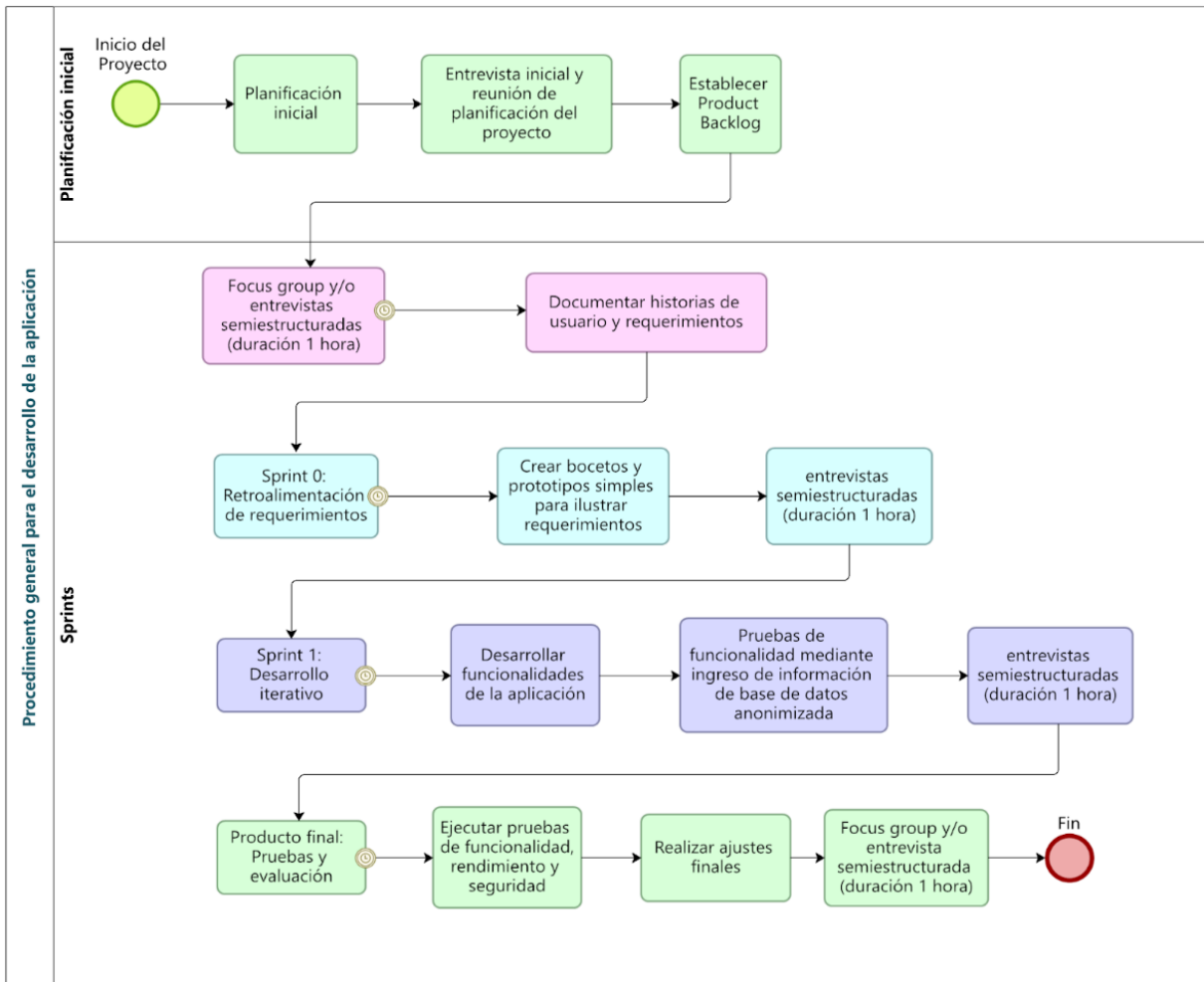


Figura A.1 Procedimiento general para el desarrollo de la aplicación siguiendo metodología Scrum.

Anexo B. Método de estrategias hacia la eficiencia clínica sugerido

A continuación, se presenta parte de los métodos de estrategias sugeridos en documento ORD N°432 “Estrategias hacia eficiencia clínica hospitalaria desde la Unidad de Emergencias” [6].

6.4 Indicadores GRD especialidad.

6.4.1 Indicadores GRD Bloque médico (incorporar valores agrupados de especialidades médicas tales como, medicina interna, neurología, gastroenterología, nefrología, oncología, hematología, broncopulmonar, etc.).

(especialidad)	2019	2020	2021	2022	2023
N° Egresos					
IEMA Inlier					
Estancia Media					
Peso Medio GRD					
% Egresos Outliers superiores					
Peso Medio Egresos Outliers superiores					
Estancias Evitables					

Anexo 3. Aspectos para considerar para realizar la visita clínica

1. Nombre

2. Edad

3. Fecha de ingreso Establecimiento _____ Unidad _____

4. Motivo de hospitalización/diagnostico

Ejemplo: SCA para terapia de reperfusión, sepsis foco urinario para manejo antibiótico y soporte, sepsis foco respiratorio para oxigenoterapia, soporte y manejo antibiótico, colecistitis aguda para resolución quirúrgica, retención urinaria aguda para resolución obstrucción (RAO) urinaria aguda estudio y manejo de la causa.

5. Manejo realizado hasta la fecha actual:

Ejemplo: SCASSDST terapia de antiagregación dual, estatinas, NTG

1. Evolución del paciente/respuesta a terapia:

Estacionario ___ Favorable ___ Desfavorable ___ Causas _____

Complicaciones: NO ___ SI ___ Cuales _____

7. Pendientes: Estudio/Interconsultas/exámenes/cirugía, con la justificación de su pertinencia:

Ejemplo:

SCA pendiente evaluación por cardiología para definir estudio con PTCA, Eco cardio 2DTT para evaluar función sistólica, complicaciones.

8. Tiempo estimado de estada para ejecutar plan de manejo y estudio con fecha probable de alta.

Figura B.1 Métodos de estrategias hacia la eficiencia clínica sugeridas en [6].

Anexo C. Participantes de Focus groups y entrevistas semiestructuradas

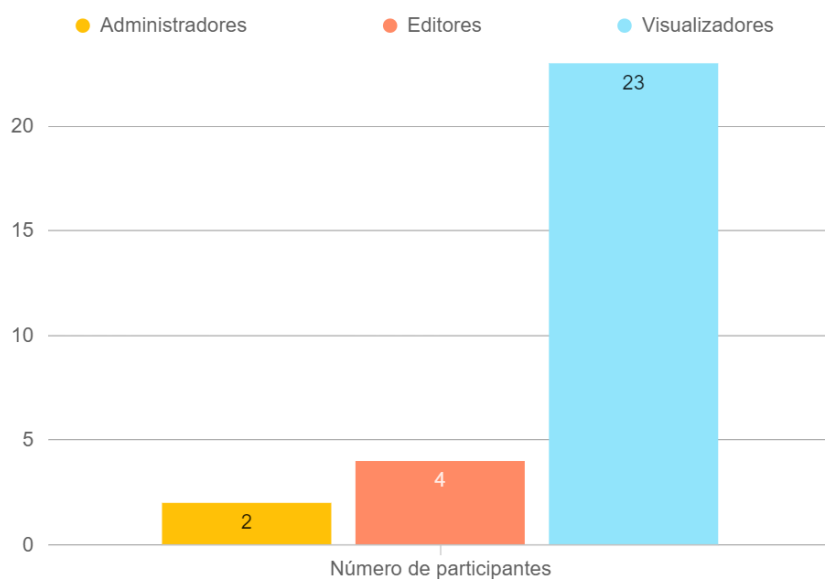


Figura C.1 Gráfica con cantidad de participantes (usuarios finales) por nivel de acceso permitido en la aplicación.



Anexo D. Resultados de historias de usuario y product backlog

Como <tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <el logro algún objetivo>
Usuario	crear mi cuenta, recuperar y cambiar mi contraseña.	tener el acceso y para mantener la seguridad de mi cuenta.
Administrador	gestionar roles y permisos de usuarios, incluyendo la capacidad de eliminar perfiles	para asegurar que solo el personal autorizado tenga acceso, mantener el control y la seguridad de la información
	quiero tener acceso total a la plataforma	para reflejar correctamente su estado en el sistema
	copiar los datos de la visita de un paciente en forma de lista	facilitar la transferencia de información
	acceso a información de estadia por diagnóstico, según datos recogidos del sistema GRD del hospital	una mejor toma de decisiones y comprensión de la situación clínica de los pacientes
	generar la redacción de un correo para enviar a superiores con el resumen de altas diarias	para optimizar tiempo en la redacción de correo de informe diario
	solo el personal con permisos según rol pueda acceder a ciertos datos	asegurar la privacidad y seguridad de la información
Editor	registrar nuevos pacientes y actualizar los datos en cada visita	mantener un registro actualizado de todos los pacientes
	ingresar, actualizar cada visita del episodio de hospitalización y cerrar el episodio de hospitalización de cada paciente al ser dado de alta	reflejar correctamente su estado en el sistema, llevar un seguimiento detallado, revisar antecedentes y tomar decisiones informadas
	ver el historial de visitas y resumen de altas diarias con la opción de búsqueda además de poder exportar información como PDF	encontrar fácilmente la información necesaria
	un indicador de progreso de la visita de cada paciente	saber cuánto falta para completar la información
	usa una planilla de registro de visita diaria, una por cada equipo	organizar mejor la distribución de pacientes
	los datos del día siguiente se carguen automáticamente, limpiando ciertos campos y recargando otros	optimizar tiempo y facilitar la actualización diaria de información
	ver un resumen de altas diario	tener un panorama claro de las altas diarias y estar al tanto de actividades recientes
	copiar los datos de la visita de un paciente en forma de lista	facilitar la transferencia de información
	ver el historial de pacientes por sala con una sección de búsqueda por sala, fecha de visita, paciente y rut	tener un registro detallado y encontrar fácilmente la información necesaria
	un código de colores según el tipo de salida del paciente	identificar rápidamente el estado de cada paciente y confirmar dicha información
	cambiar de sala al paciente	organizar y optimizar el uso de camas y recursos hospitalarios
Visualizador	ver el historial de visitas de cada paciente y buscar visitas por fecha	revisar antecedentes y encontrar fácilmente la información necesaria
	ver resúmenes diarios por equipo y ver el historial de pacientes por sala	estar al tanto de las actividades recientes

Figura D.1 Resultados de historias de usuario.


Requerimientos	Prioridad
Creación y gestión de perfiles por parte de usuarios y administradores, incluyendo la recuperación y cambio de contraseñas	
Registro, actualización y cierre de episodios de hospitalización de pacientes	
Registro de nuevas visitas clínicas, consulta del historial de visitas y búsqueda de visitas por fecha	
Registro diario de visitas por equipo, cambio de sala de pacientes, modificación y cierre de episodios, acceso a resúmenes de altas y exportación de historial en PDF	
Gestión de roles y permisos para asegurar que solo el personal autorizado pueda visualizar datos sensibles y acceder a la interfaz de la aplicación	
Una interfaz intuitiva y compatible con todos los dispositivos en plataforma web	
Visualización del historial de pacientes por sala, búsqueda avanzada, código de colores según el tipo de salida del paciente e indicadores de progreso de las visitas	

Figura D.2 Product Backlog con requerimientos para la aplicación y su nivel de prioridad

Anexo E. Desarrollo de sprints

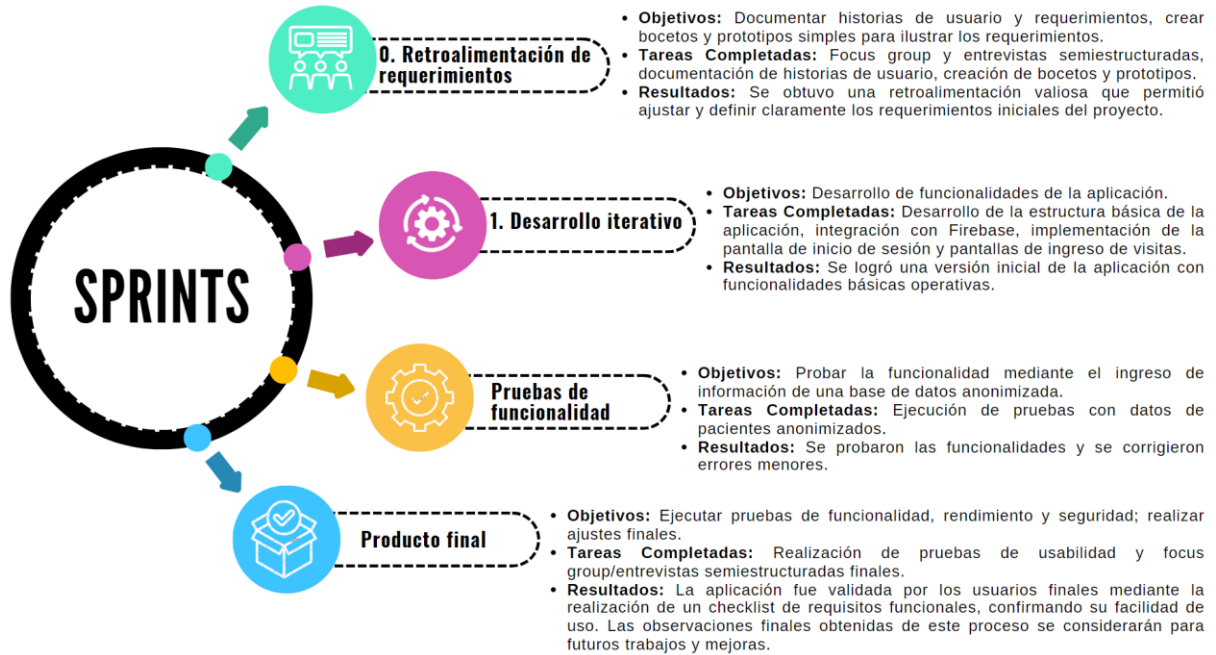


Figura E.1 Desarrollo de sprints.



Anexo F. Base de datos

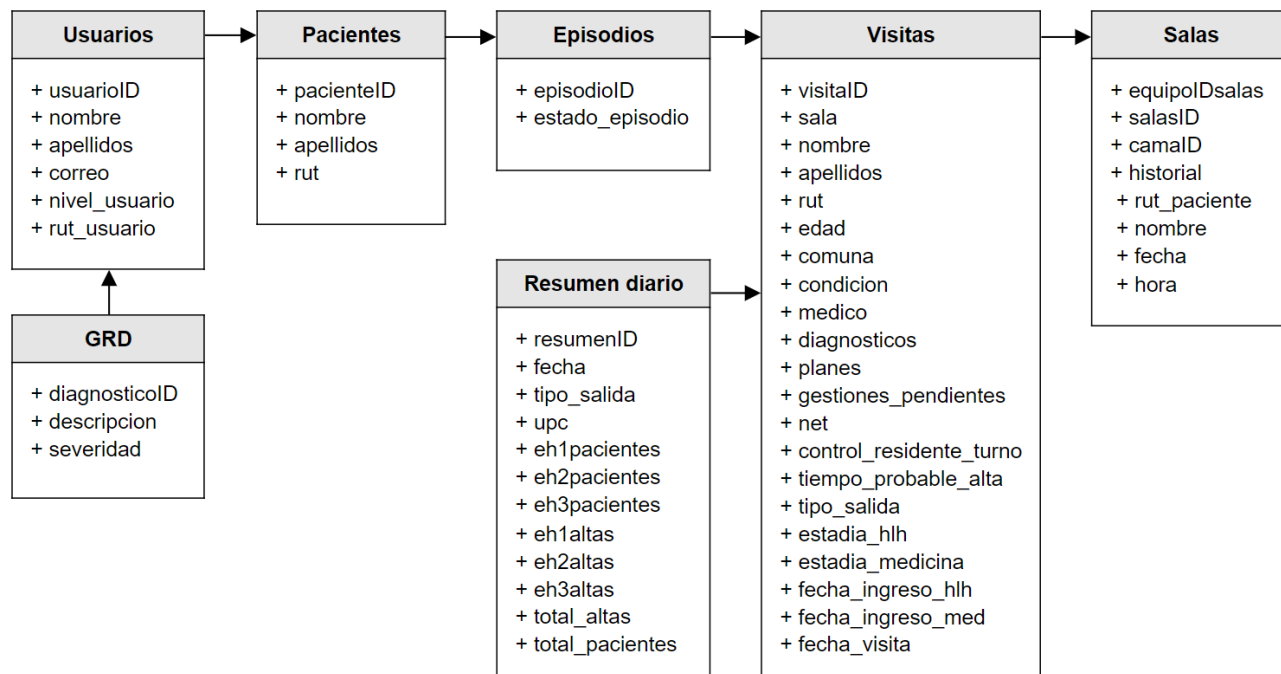


Figura F.1 Diagrama de base de datos

🏠 > GRD > I50.0 ☁ Más función		
(default)	GRD	I50.0
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
GRD >	I50.0 >	severidad
pacientes	J18.8	
resumen_diario	K92.2	
salas	N17.8	
usuarios	N18.5	
	N39.0	
	Z51.1	
		+ Agregar campo
		descripcion: "Insuficiencia cardíaca congestiva"

🏠 > GRD > N39.0 > severidad		
GRD	N39.0	severidad
+ Agregar documento	+ Iniciar colección	+ Agregar documento
I50.0	severidad >	mayor
J18.8		menor
K92.2		moderada
N17.8		
N18.5		
N39.0 >	+ Agregar campo	
Z51.1	descripcion: "Infección de vías urinarias, sitio no especificado"	

Figura F.2 Base de datos en Cloud Firestore.

Anexo G. Resultados adicionales de la aplicación

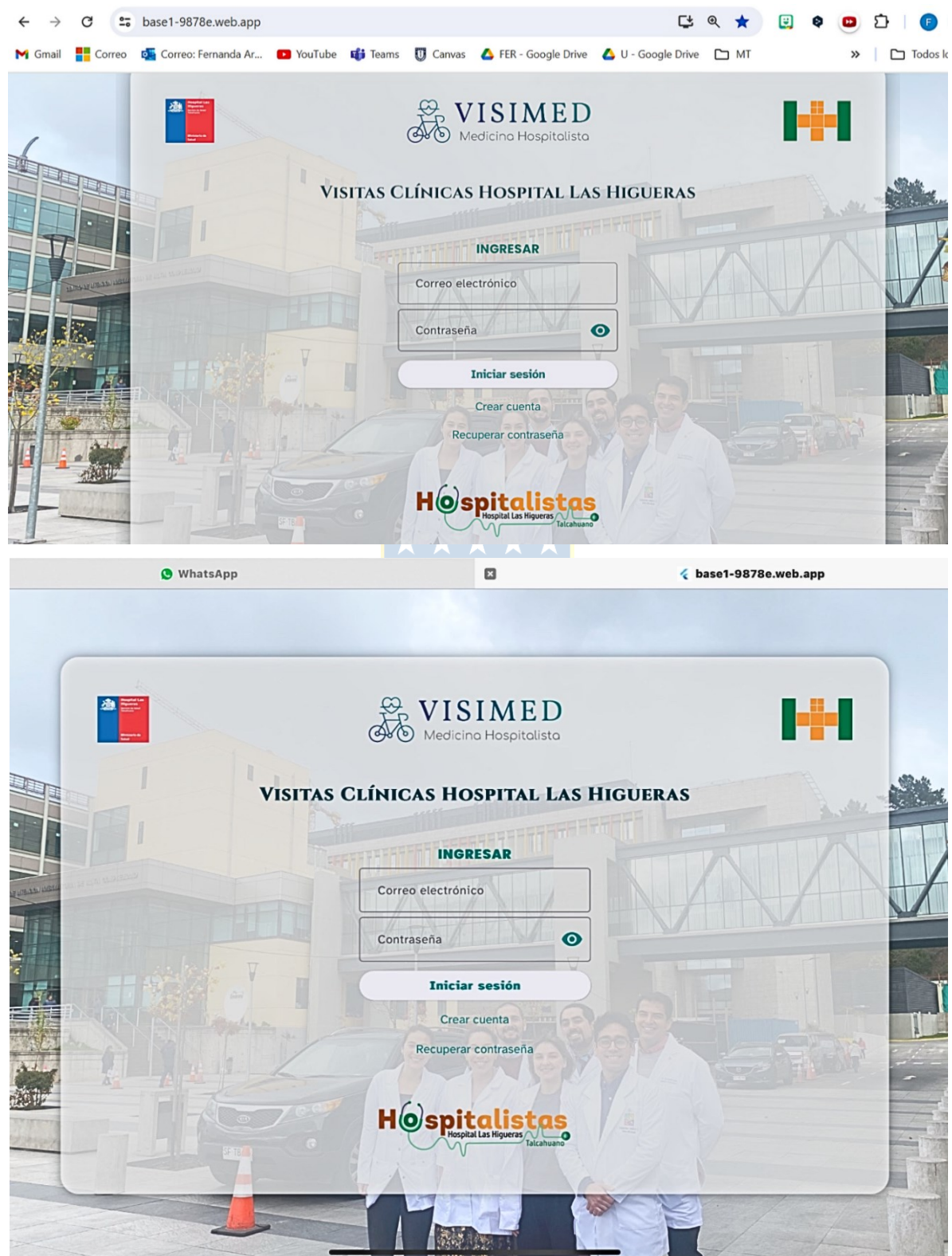


Figura G.1 Pantalla de ingreso a la aplicación en distintos navegadores (Chrome y Safari).

The image displays a mobile application interface for managing patient visits. It features a list of visits under the heading "Visitas EHI". Each visit entry includes a status indicator (e.g., 85%, 80%, 75%), a UGA code (e.g., UGA 1-2, UGA 1-3), RUT, Name, Surname, and Age. The interface is annotated with numbered callouts (1-4) and action panels.

Callout 1: Points to a detailed view of a patient's data, including "Condición" (Estable, De cuidado, Grave), "Tipo Salida" (Domicilio, UPC), and "Domicilio" (Hodom, Hodom Privado, Otro Servicio HLH, Rescate a su centro, Traslado en red, Traslado fuera de red).

Callout 2: Points to the "Abrir nuevo episodio" (Open new episode) action in the "Acciones" menu.

Callout 3: Points to the "Cerrar episodio" (Close episode) action in the "Acciones" menu.

Callout 4: Points to the "Copiar datos" (Copy data) action in the "Acciones" menu, which results in a toast message "Datos copiados al portapapeles" (Data copied to clipboard) and a text box containing patient information: sala: UGA 1-1, rut: 13.68, nombre: Rosa, apellidos: Esp, edad: 48, medico: comuna: Talcahuano, diagnosticos:.

Condición	Tipo Salida	Domicilio
Estable	UPC	Hodom
De cuidado		Hodom Privado
Grave		Otro Servicio HLH
		Rescate a su centro
		Traslado en red
		Traslado fuera de red

Figura G.2 Funcionalidades (1. Selección de opciones y código de colores según tipo de salida, 2. Ingreso de episodio de paciente, 3. Cierre de episodio y 4. Copia de datos del paciente en forma de lista).

Estadía MIH	Estadía HLH	Fecha Ingreso HLH	Fecha Ingreso MIH	Fecha Visita	Acciones
3	3	21-07-24	21-07-24	24-07-24	[Iconos de acciones]
2	2	22-07-24	22-07-24	24-07-24	[Iconos de acciones]
5	9	15-07-24	19-07-24	24-07-24	[Iconos de acciones]
2	2	22-07-24	22-07-24	24-07-24	[Iconos de acciones]
3	8	16-07-24	21-07-24	24-07-24	[Iconos de acciones]
2	2	22-07-24	22-07-24	24-07-24	[Iconos de acciones]
6	10	14-07-24	18-07-24	24-07-24	[Iconos de acciones]

Selección de fecha en calendario para el ingreso HLH.

Formulario de selección de fecha con campo de texto y botones de acción.

Modal de selección de sala con lista desplegable y botones de confirmación.

Lista de visitas EHI con columnas de RUT, Nombre y estado.

Figura G.3 Funcionalidades (1. Visualización de estadías en MIH y HLH, con cambio de color, 2. Calendario para seleccionar fecha de ingreso en HLH y MIH, 3. Mover de sala/cama al paciente).



Anexo H. Checklist de verificación de requerimientos



Checklist de verificación de requerimientos

Nombre del proyecto:	Diseño y desarrollo de aplicación para apoyo en la confirmación del alta de pacientes hospitalizados en el Servicio de Medicina Interna del Hospital Las Higueras
Nombre de la plataforma:	VisiMed
Fecha:	

CRITERIO DE EVALUACIÓN	SI	NO	Observación
1. Gestión de perfiles			
1.1. Los funcionarios pueden crear y gestionar sus perfiles			
1.2. El administrador puede gestionar perfiles de usuarios visualizadores y editores (incluye eliminar perfil)			
1.3. El perfil administrador puede gestionar los perfiles de todos los usuarios			
1.4. Los usuarios pueden recuperar y cambiar sus contraseñas			
2. Gestión de pacientes			
2.1. Los usuarios editores pueden registrar nuevos pacientes			
2.2. Los usuarios editores pueden actualizar la información de los pacientes			
2.3. Los usuarios editores y administradores pueden cerrar el registro de episodio de hospitalización de un paciente			
3. Registro y consulta de visitas			
3.1. Los usuarios administradores y editores pueden registrar nuevas visitas clínicas			
3.2. Los usuarios pueden ver el historial de visitas de cada paciente			
3.3. Los usuarios pueden filtrar y buscar visitas por fecha			
4. Gestión de registro de hospitalización			
4.1. Existen 3 planillas de registro de visita diaria, una por cada equipo			
4.2. Los usuarios administradores y editores pueden ingresar a un nuevo paciente a una sala disponible			
4.3. Los usuarios administradores y editores pueden cambiar de sala al paciente			
4.4. Los usuarios administradores y editores pueden modificar y cerrar el episodio de hospitalización al dar de alta al paciente			
4.5. Los usuarios editores y visualizadores pueden ver un resumen con la cantidad de altas según el tipo de salida del paciente			
4.6. Tanto los usuarios editores y administradores pueden copiar los datos de la visita de un paciente en forma de lista			
4.7. Se muestra el resumen del día anterior por equipo y el resumen del día actual en el inicio de página			
4.8. Los usuarios tienen acceso a información de días de estadía por diagnóstico, según datos recogidos del sistema GRD del hospital			
4.9. Los usuarios pueden exportar en formato PDF el historial de visitas y resúmenes diarios			
5. Seguridad y Privacidad			
5.1. Solo los usuarios con permisos según rol otorgado por el administrador puede visualizar datos sensibles			
5.2. Los usuarios registrados solo podrán acceder y visualizar correctamente la interfaz de la aplicación después de que el administrador les haya asignado el rol correspondiente			
5.3. Los roles y permisos están gestionados para que los usuarios solo accedan a la información pertinente a sus funciones			
6. Interfaz de usuario			
6.1. La interfaz es intuitiva y fácil de usar			
6.2. La aplicación funciona para todos los dispositivos en plataforma web			
7. Soporte y funcionalidades adicionales			
7.1. La aplicación permite a los usuarios ver el historial de pacientes por sala			
7.2. La aplicación tiene una sección de búsqueda por sala, fecha de visita, paciente y rut			
7.3. Existe un código de colores según el tipo de salida del paciente			
7.4. Existe un indicador gráfico que muestra el progreso de la visita de cada paciente en la planilla			

UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA

RESUMEN DE MEMORIA DE TITULO

Departamento	: Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería Civil Biomédica
Nombre del memorista	: Fernanda Arlette Henríquez Parra
Título de la memoria	: Diseño y desarrollo de aplicación para apoyo en la confirmación del alta de pacientes hospitalizados en el Servicio de medicina interna del Hospital Las Higueras.
Fecha de la presentación oral	: 26 de agosto del 2024
Profesor(es) Guía	: Rosa Figueroa Iturrieta, Hernán Carrillo Bestagno
Profesor(es) Revisor(es)	: Mario Medina Carrasco, Esteban Pino Quiroga
Concepto	:
Calificación	:

Resumen

Esta memoria de título se enfoca en el diseño y desarrollo de una aplicación web para optimizar el proceso de gestión de altas de pacientes hospitalizados en el Servicio de Medicina Interna del Hospital Las Higueras de Talcahuano. Actualmente, la gestión de información se realiza mediante planillas de registro digitales, las cuales presentan importantes limitaciones en términos de eficiencia, precisión y seguridad de los datos. El objetivo del proyecto es desarrollar una aplicación especializada que supere estas limitaciones, mejorando la calidad de la atención médica y facilitando la comunicación entre el personal médico y de enfermería.

La aplicación propuesta está diseñada para reducir los tiempos de estancia hospitalaria, asegurar la confidencialidad y privacidad de la información del paciente y disminuir el tiempo dedicado a tareas administrativas. Además, pretende mejorar la coordinación y minimizar errores, proporcionando una herramienta intuitiva y fácil de usar para el personal hospitalario.

El desarrollo de la aplicación se llevó a cabo utilizando la metodología ágil Scrum, con la participación de los profesionales clínicos como usuarios finales. Estos profesionales colaboraron en el diseño y desarrollo, aportando información sobre los requerimientos y validando las entregas del proyecto para asegurar la pertinencia y efectividad de la aplicación en el contexto hospitalario. Los principales logros del proyecto incluyen la implementación de un sistema de autenticación seguro, el desarrollo de interfaces intuitivas para el ingreso y visualización de datos, y la integración con Firebase para el almacenamiento y sincronización de datos en tiempo real. Las pruebas realizadas con datos anonimizados de 100 pacientes confirmaron que la aplicación cumple con los requisitos funcionales y es fácil de usar.

En conclusión, la aplicación desarrollada mejora significativamente la eficiencia, precisión y capacidad de gestión en el proceso de altas hospitalarias. La participación de los profesionales clínicos en el proceso de diseño y desarrollo garantizó que la aplicación satisficiera sus necesidades, proporcionando una solución tecnológica que fortalece la seguridad de los datos del paciente y optimiza los procesos diarios del personal médico. Esta herramienta no solo apoya la gestión de altas, sino que también podría ser utilizada para la entrega oficial de turnos, un aspecto crucial para la calidad de la atención en el hospital.