

SafeMed - Report



Universidad Peruana de Ciencias Aplicadas

INGENIERÍA DE SOFTWARE

Ciclo: 7

CURSO: Desarrollo de Soluciones IOT | SECCIÓN 2941

Profesor: Leon Baca, Marco Antonio

Proyecto de curso

Informe del TF

StartUp: SafeMed

Producto: MedSystem

Integrantes:

Integrantes	Codigo
De La Piedra Quintanilla, Erwin Miquel	U202112179
Gutierrez Zumaeta, Manuel Alonso	U202112353
Mendez Rosales, Marco Aurelio	U202018273
Roca Huapaya, Orlando Arturo	U201919742
Ventura Chancafe, Eduardo Renato	U202212645

Ciclo 2025-10**Abril, 2025**

Versión	Fecha	Autor	Descripción de modificación
0.1	19-04-2025	Gutierrez Zumaeta, Manuel Alonso	Creación de 1.1 Startup Profile y 1.1.1 Descripción de la Startup
0.2	20-04-2025	De La Piedra Quintanilla, Erwin Miquel	Redacción de 1.1.2 Perfiles de integrantes del equipo
0.3	20-04-2025	Ventura Chancafe, Eduardo Renato	Desarrollo de 1.2 Solution Profile y 1.2.1 Antecedentes y problemática
0.4	21-04-2025	Mendez Rosales, Marco Aurelio	Elaboración de 1.2.2 Lean UX Process (Problem Statements, Assumptions, Hypothesis Statements, Canvas)
0.5	21-04-2025	Roca Huapaya, Orlando Arturo	Definición de 1.3 Segmentos Objetivo
0.6	22-04-2025	Gutierrez Zumaeta, Manuel Alonso	Ánalysis de 2.1 Competidores y 2.1.1 Análisis competitivo
0.7	22-04-2025	Ventura Chancafe, Eduardo Renato	Redacción de 2.1.2 Estrategias y tácticas frente a competidores
0.8	23-04-2025	De La Piedra Quintanilla, Erwin Miquel	Diseño de entrevistas (2.2.1) y registro de entrevistas (2.2.2)
0.9	23-04-2025	Mendez Rosales, Marco Aurelio	Análisis de entrevistas (2.2.3) y Needfinding (2.3)

Versión	Fecha	Autor	Descripción de modificación
1.0	26-04-2025	Roca Huapaya, Orlando Arturo	Desarrollo completo de capítulos 2.3.1 a 4.2 (User Personas, Task Matrix, Journey Mapping, Empathy Mapping, Scenario Mapping, Ubiquitous Language, To-Be Scenario Mapping, User Stories, Impact Mapping, Product Backlog, Strategic-Level DDD y Software Architecture)
1.1	29-04-2025	Gutierrez Zumaeta, Manuel Alonso	Redacción de Capítulo 5: Solution UI/UX Design
1.2	02-05-2025	De La Piedra Quintanilla, Erwin Miquel	Desarrollo de 5.1 a 5.3.2: Guidelines, Architecture y Landing Page UI
1.3	06-05-2025	Ventura Chancafe, Eduardo Renato	Elaboración de 5.4 y 5.5: Applications UX/UI y Prototyping
1.4	09-05-2025	Mendez Rosales, Marco Aurelio	Inicio de Capítulo 6: Product Implementation y Configuration Management
1.5	12-05-2025	Roca Huapaya, Orlando Arturo	Redacción de 6.1 y 6.2: Software Configuration y Sprint Planning
1.6	15-05-2025	Miquel De la Piedra Quintanilla, Erwin	Actualización de 6.2.X: Aspect Leaders, Sprint Backlog y Testing Suite

Project Report Collaboration Insights

TB1

Para la entrega de la TB1 se realizó una reunión donde se asignaron las responsabilidades a cada integrante del equipo. A continuación se muestra la siguiente tabla con los detalles:

Integrante	Responsabilidad
De la Piedra, Miquel	Capítulo 4, Entrevista
Gutierrez, Manuel	Capítulo 2, Entrevista
Mendez, Marco	Capítulo 4, Entrevista

Integrante	Responsabilidad
Roca, Orlando	Capítulo 1, Entrevista
Ventura, Eduardo	Capítulo3, Entrevista

TP

Para la entrega de la TB1 se realizó una reunión donde se asignaron las responsabilidades a cada integrante del equipo. A continuación se muestra la siguiente tabla con los detalles:

Integrante	Responsabilidad
De la Piedra, Miquel	Capítulo 6
Gutierrez, Manuel	Capítulo 6
Mendez, Marco	Capítulo 5
Roca, Orlando	Capítulo 5
Ventura, Eduardo	Capítulo5

Durante la elaboración de las aplicaciones, tanto web como mobile, se realizaron commits respectivos con el fin de mantener el orden y un control de versiones eficiente.

Para tener mejor precisión en los integrantes del equipo, a continuación presentamos los usuarios de GitHub de los integrantes:

- Miquel De la Piedra (@MiquelDlp)
- Manuel Gutierrez (@ManuGZ)
- Marco Mendez (@cuak123)
- Orlando Roca (@OrlandoRoca)
- Eduardo Ventura (@Eduvent)

A continuación se presentan las capturas del repositorio de GitHub donde se realizaron los avances correspondientes.

Tabla de contenidos

- [Student Outcome](#)

Capítulo I: Introducción

- [1.1. Startup Profile](#)
 - [1.1.1. Descripción de la Startup](#)
 - [1.1.2. Perfiles de integrantes del equipo](#)
- [1.2. Solution Profile](#)
 - [1.2.1. Antecedentes y problemática](#)
 - [1.2.2. Lean UX Process](#)
 - [1.2.2.1. Lean UX Problem Statements](#)
 - [1.2.2.2. Lean UX Assumptions](#)
 - [1.2.2.3. Lean UX Hypothesis Statements](#)

- 1.2.2.4. Lean UX Canvas
- 1.3. Segmentos objetivo

Capítulo II: Requirements Elicitation & Analysis

- 2.1. Competidores
 - 2.1.1. Análisis competitivo
 - 2.1.2. Estrategias y tácticas frente a competidores
- 2.2. Entrevistas
 - 2.2.1. Diseño de entrevistas
 - 2.2.2. Registro de entrevistas
 - 2.2.3. Análisis de entrevistas
- 2.3. Needfinding
 - 2.3.1. User Personas
 - 2.3.2. User Task Matrix
 - 2.3.3. User Journey Mapping
 - 2.3.4. Empathy Mapping
 - 2.3.5. As-is Scenario Mapping
- 2.4. Ubiquitous Language

Capítulo III: Requirements Specification

- 3.1. To-Be Scenario Mapping
- 3.2. User Stories
- 3.3. Impact Mapping
- 3.4. Product Backlog

Capítulo IV: Solution Software Design

- 4.1. Strategic-Level Domain-Driven Design
 - 4.1.1. EventStorming
 - 4.1.1.1. Candidate Context Discovery
 - 4.1.1.2. Domain Message Flows Modeling
 - 4.1.1.3. Bounded Context Canvases
 - 4.1.2. Context Mapping
 - 4.1.3. Software Architecture
 - 4.1.3.1. Software Architecture System Landscape Diagram
 - 4.1.3.2. Software Architecture Context Level Diagrams
 - 4.1.3.2. Software Architecture Container Level Diagrams
 - 4.1.3.3. Software Architecture Deployment Diagrams

Student Outcome

ABET – EAC - Student Outcome 5: Trabaja efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo; crea un entorno colaborativo e inclusivo y establece metas, planifica tareas y cumple objetivos

Criterio específico	Acciones realizadas	Conclusiones
5.c.1 Trabaja en	Miquel De la Piedra	Nosotros logramos compartir el liderazgo

equipo para desarrollar liderazgo en forma conjunta	Acciones realizadas	de forma equilibrada, permitiendo que Conclusiones cada integrante tomara iniciativas en distintas fases del proyecto, fortaleciendo la comunicación y el compromiso grupal.
	TB1: Asumí la coordinación en el desarrollo del Capítulo 4, organizando las tareas y promoviendo la participación activa de todos los miembros.	
	TP: En esta etapa del proyecto, colaboré activamente con mi equipo facilitando la comunicación constante y apoyando la coordinación de tareas para asegurar que cumplamos los objetivos del entregable de manera eficiente.	
	Manuel Gutierrez	
	TB1: Lideré la estructuración del Capítulo 2, guiando las sesiones de entrevistas y fomentando el trabajo conjunto en el análisis de resultados.	
	TP: Asumí el liderazgo compartido durante la planificación, promoviendo que todos los integrantes participen en la toma de decisiones y que sus ideas sean consideradas para enriquecer el trabajo conjunto.	
	Marco Mendez	
	TB1: Guié la creación de los modelos de arquitectura en el Capítulo 4, impulsando la colaboración en las decisiones técnicas.	
	TP: Durante la resolución de problemas, ayudé a mediar entre las diferentes opiniones del equipo para encontrar soluciones consensuadas que nos permitieran avanzar sin conflictos.	
	Orlando Roca	
	TB1: Organicé la fase inicial de la elaboración del Capítulo 1, facilitando espacios de diálogo para definir en equipo los objetivos y enfoques.	
	TP: Reconocí las habilidades de mis compañeros y las integré en las actividades específicas del entregable, asignando responsabilidades acorde a sus fortalezas	

Criterio específico	Acciones realizadas	Conclusiones
	<p>para optimizar el desempeño del grupo.</p> <p>Eduardo Ventura</p> <p>TB1: Coordiné la integración de los requisitos en el Capítulo 3, asegurándome de que cada aporte del equipo fuera considerado y validado.</p> <p>TP: Me comprometí a compartir la responsabilidad del liderazgo rotando roles con mis compañeros, fomentando un ambiente de corresponsabilidad y asegurando que todos contribuyan activamente en el desarrollo del entregable.</p>	
5.c.2 Crea un entorno colaborativo e inclusivo, establece metas, planifica tareas y cumple objetivos	<p>Miquel De la Piedra</p> <p>TB1: Establecí metas claras para la entrega del Capítulo 4 y organicé las tareas en un tablero colaborativo, asegurando la participación de todos.</p> <p>TP: Durante este entregable, promoví espacios de diálogo inclusivos donde cada miembro pudo expresar sus ideas y aportar a la planificación general, garantizando que todos se sintieran valorados.</p>	Nosotros creamos un entorno de trabajo inclusivo y colaborativo que facilitó el cumplimiento de nuestras metas, permitiendo que cada integrante aportara de manera activa y responsable en la construcción del proyecto.
	<p>Manuel Gutierrez</p> <p>TB1: Fomenté un ambiente inclusivo durante las entrevistas del Capítulo 2, asegurándome de que todos los participantes y miembros del equipo fueran escuchados.</p> <p>TP: Coordiné la definición de metas y la distribución de tareas, utilizando herramientas colaborativas que facilitaron el seguimiento y cumplimiento de los objetivos del grupo.</p>	
	<p>Marco Mendez</p> <p>TB1: Promoví sesiones de trabajo conjunto para construir los modelos de EventStorming, incentivando el intercambio de ideas y propuestas.</p> <p>TP: Me aseguré de que el ambiente de</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>trabajo fuera abierto y respetuoso, motivando la participación activa de todos y planificando tareas que consideraran las fortalezas individuales.</p> <p>Orlando Roca</p> <p>TB1: Planifiqué y seguí el cumplimiento de los hitos establecidos en el Capítulo 1, manteniendo una comunicación constante para resolver dudas y dificultades.</p> <p>TP: Supervisé el progreso de las tareas asignadas, ajustando el plan de trabajo cuando fue necesario para garantizar el cumplimiento puntual de los objetivos establecidos.</p>	
	<p>Eduardo Ventura</p> <p>TB1: Organicé el repositorio de trabajo del Capítulo 3, asegurando una integración ordenada de las tareas y reconociendo cada contribución del equipo.</p> <p>TP: Incentivé la responsabilidad compartida y el apoyo mutuo dentro del equipo, contribuyendo a un entorno colaborativo donde se valoró el esfuerzo individual y colectivo.</p>	

Capítulo I: Introducción

1.1. Startup Profile

1.1.1. Descripción de la Startup

SafeMed es una startup tecnológica creada por estudiantes de Ingeniería de Software de la Universidad Peruana de Ciencias Aplicadas (UPC), enfocada en el desarrollo de soluciones innovadoras para el sector salud mediante el uso de tecnologías IoT. Nuestro propósito es brindar herramientas digitales que permitan mejorar la calidad de vida de pacientes con enfermedades crónicas, facilitando el monitoreo remoto de sus signos vitales y proporcionando canales de comunicación directa entre familiares y profesionales médicos.

SafeMed nace como respuesta a la creciente necesidad de una atención médica más preventiva, accesible y en tiempo real. A través de nuestro sistema inteligente, buscamos detectar de forma oportuna alteraciones en la salud de los pacientes y notificar automáticamente a las personas responsables de su cuidado, ayudando a reducir tiempos de respuesta y mejorar la gestión médica.

Misión: Salvar vidas a través de tecnología accesible e inteligente, brindando soluciones que permitan a las familias y al personal médico monitorear en tiempo real la salud de los pacientes, y actuar con rapidez ante cualquier anomalía.

Visión: Ser la plataforma líder en Latinoamérica en soluciones digitales para el monitoreo remoto de salud, reconocida por su innovación, confiabilidad y contribución al bienestar de los pacientes.

1.1.2. Perfiles de integrantes del equipo

Foto	Nombre y Descripción
 A portrait photograph of a young man with dark hair and glasses, wearing a black hoodie. He is looking directly at the camera.	Nombre: Manuel Alonso Gutierrez Zumaeta Código: u202112353 Descripción: Soy Manuel Gutierrez, estudiante de la carrera de Ingeniería de Software, tengo 20 años. Tengo experiencia en el desarrollo de aplicaciones web, tanto frontend como backend, programación móvil en Flutter. Me considero una persona responsable y que se propone hacer un buen trabajo.
	Nombre: Marco Aurelio Mendez Rosales Código: u200218273 Descripción: Soy Marco Mendez, estudiante de la carrera de Ingeniería de softwar, tengo 22 años, me gusta crear soluciones que puedan ayudar a los demás.

Foto

Nombre y Descripción



Foto**Nombre y Descripción**



Nombre: Erwin Miquel De la Piedra Quintanilla

Código: u202112179

Descripción: Soy Miquel De la Piedra, tengo 21 años, estudiante de la carrera de Ingeniería de Software. Poseo conocimientos relacionados a la programación en C++, Python, HTML y CSS. Además poseo experiencia con el manejo de bases de datos. Considero que estos conocimientos ayudarán a la realización de este trabajo, además de mejorar nuestros métodos de organización y trabajo en equipo.



Nombre: Eduard Renato Ventura Chancafe

Código: u202212645

Descripción: Soy estudiante universitario que cursa su 7to ciclo. Dentro del startup me encargo con el desarrollo del diseño ,QA y gestión del equipo, pues tengo conocimientos adicionales a los de mi sector. Además, apoyaré en el desarrollo del código trabajando de forma cohesionada con mis compañeros. Asimismo,uento con conocimientos en herramientas como HTML, CSS, JavaScript, Python, C++, Cypress, NodeJs, NestJs y Agile Project Management.

Foto	Nombre y Descripción
	Nombre: Orlando Arturo Roca Huapaya Código: u201919742

Nombre: Orlando Arturo Roca Huapaya

Código: u201919742

Descripción: Como estudiante de ingeniería de software, mi contribución al equipo se centra en mis conocimientos en la planificación y diseño de software. A lo largo de mi formación he podido desarrollar proyectos utilizando Java, lo que me da una ventaja significativa con el sistema que trabajamos. Estoy entusiasmado por aplicar mis habilidades en el desarrollo y trabajar junto al equipo para alcanzar nuestros objetivos.

1.2. Solution Profile

1.2.1 Antecedentes y problemática

Las enfermedades cardiovasculares y otros trastornos relacionados con los signos vitales representan una de las principales causas de hospitalizaciones y muertes a nivel mundial. Según datos de la Organización Mundial de la Salud (OMS), millones de personas sufren cada año eventos de salud que podrían ser evitados o tratados de manera más efectiva si existieran mecanismos de monitoreo continuo fuera del ámbito hospitalario.

Actualmente, muchos pacientes con condiciones de riesgo no cuentan con dispositivos accesibles que les permitan una vigilancia constante de su estado de salud en tiempo real. La ausencia de monitoreo domiciliario efectivo genera que las alteraciones en signos vitales pasen desapercibidas, retrasando la intervención médica oportuna. Además, la falta de notificaciones inmediatas a familiares o médicos contribuye a incrementar la gravedad de los eventos de emergencia.

En respuesta a esta necesidad, SafeMed propone una solución que integra tecnología IoT mediante sensores de frecuencia cardiaca conectados a una aplicación móvil, permitiendo el monitoreo remoto en tiempo real. La plataforma no solo alerta automáticamente a familiares y médicos ante cualquier desviación crítica, sino que también ofrece herramientas de gestión médica para mejorar el seguimiento clínico de los pacientes.

What (¿Qué?):

Falta de monitoreo remoto continuo de signos vitales que permita alertar de forma automática y oportuna sobre posibles emergencias de salud.

When (¿Cuándo?):

Durante la vida diaria del paciente, especialmente en momentos sin supervisión directa de profesionales médicos.

Where (¿Dónde?):

En el hogar, en el trabajo o cualquier lugar donde el paciente se encuentre fuera de un entorno hospitalario.

Who (¿Quién?):

Pacientes con enfermedades cardiovasculares, adultos mayores, personas con condiciones médicas crónicas, sus familiares, y profesionales de salud encargados de su cuidado.

Why (¿Por qué?):

La falta de soluciones accesibles y eficientes para el monitoreo constante incrementa el riesgo de que eventos críticos no sean detectados a tiempo.

How (¿Cómo?):

A través de un sensor portátil que mide la frecuencia cardíaca en tiempo real, conectado a una aplicación móvil que genera alertas automáticas a familiares y médicos.

How Much (¿Cuánto?):

Estudios recientes en Latinoamérica estiman que cerca del 30% de las hospitalizaciones por eventos cardiovasculares podrían ser prevenidas mediante sistemas de alerta temprana en el hogar (fuente simulada para ilustrar).

1.2.2 Lean UX Process.

1.2.2.1. Lean UX Problem Statements.

El sistema de atención médica actual presenta una limitación importante: no garantiza un monitoreo continuo ni alertas inmediatas para pacientes con condiciones crónicas o de riesgo, especialmente fuera de los entornos hospitalarios. Esta carencia expone a los pacientes a situaciones de emergencia que podrían ser prevenidas o atendidas de forma más oportuna.

Hemos observado que, en el entorno domiciliario, los pacientes en riesgo no cuentan con soluciones accesibles para detectar alteraciones críticas en sus signos vitales ni mecanismos automáticos de alerta hacia sus familiares o doctores de confianza. Esta brecha genera un aumento en la gravedad de los episodios médicos, incrementa los costos de atención hospitalaria y afecta negativamente la calidad de vida tanto de los pacientes como de sus familias.

¿Cómo podríamos diseñar una solución de monitoreo remoto que permita detectar alteraciones en tiempo real y notificar automáticamente a familiares y médicos, mejorando así la atención oportuna y la seguridad de los pacientes, sin necesidad de intervención manual constante?

1.2.2.2. Lean UX Assumptions.

User Assumptions:

¿Quién es el usuario?

Pacientes con condiciones de salud que requieren monitoreo constante (como enfermedades cardiovasculares), sus familiares directos y médicos tratantes.

¿Dónde encaja nuestro producto en su trabajo o vida?

SafeMed se integra en la vida cotidiana del paciente y su familia, proporcionando monitoreo continuo de signos vitales, generando alertas inmediatas en caso de anomalías, y facilitando la gestión médica a distancia.

¿Qué problemas tiene que resolver nuestro producto?

Falta de monitoreo permanente de signos vitales en el hogar.

Falta de notificaciones inmediatas ante emergencias.

Dificultad en el seguimiento médico remoto y en la toma de decisiones oportunas.

¿Cuándo y cómo es usado nuestro producto?

SafeMed se usa diariamente de manera pasiva: el sensor mide continuamente la frecuencia cardiaca, y la aplicación móvil procesa los datos en tiempo real. Solo en caso de detectar valores anormales, se envían notificaciones automáticas a familiares y médicos.

¿Cómo debe verse y comportarse nuestro producto?

Debe ser intuitivo, fácil de configurar, seguro y confiable. La app debe mostrar la información de manera clara, con gráficos sencillos de interpretar, y notificar con rapidez y precisión ante cualquier riesgo.

Business Assumptions:

- Creo que mis usuarios necesitan una solución de software eficiente y accesible para la personalización de su auto, evitando que otras personas o empresas los estafen.
- Estas necesidades se pueden satisfacer mediante una aplicación móvil intuitiva y segura que guíe a los usuarios a través del proceso de personalización automotriz y les proporcione transparencia en las transacciones.
- Mis clientes iniciales son propietarios de vehículos que desean personalizar sus autos de manera confiable y sin riesgos.
- El valor #1 que un cliente busca en mi servicio es la capacidad de personalizar su vehículo de forma segura y confiable, evitando estafas y asegurándose de obtener los productos y servicios deseados.
- El cliente también puede obtener estos beneficios adicionales: una amplia selección de opciones de personalización, asesoramiento experto, precios transparentes y competitivos, y una experiencia de usuario fluida.
- Voy a adquirir la mayoría de mis clientes a través de campañas de marketing digital dirigidas a propietarios de vehículos interesados en la personalización automotriz, así como mediante asociaciones con concesionarios de automóviles y talleres especializados.

- Haré dinero a través de una combinación de modelos de negocio, que pueden incluir distintos planes de suscripción, ingresos por publicidad dentro de la aplicación y otros servicios premium. Además, estableceré acuerdos de asociación con staffs mecánicos para ofrecer servicios exclusivos y beneficios a los suscriptores.
- Mi competencia principal en el mercado son otras aplicaciones y plataformas en línea que ofrecen servicios similares de personalización automotriz, así como talleres físicos y tiendas especializadas.
- Los venceremos debido a nuestra atención en la seguridad y transparencia del proceso, nuestra amplia gama de opciones de personalización, y nuestra interfaz intuitiva y fácil de usar.
- Mi mayor riesgo es la falta de confianza por parte de los usuarios en la seguridad y la transparencia de la plataforma.
- Resolvemos esto a través de una estricta verificación de los staff mecánicos y la implementación de medidas de seguridad robustas, así como una comunicación clara y transparente con los usuarios sobre los procesos y las políticas de la plataforma.

1.2.2.3. Lean UX Hypothesis Statements.

- Creemos que al proporcionar un monitoreo continuo de signos vitales y alertas automáticas en tiempo real a familiares y médicos, los usuarios podrán detectar de manera temprana eventos críticos de salud.

Sabremos que hemos tenido éxito

Cuando el 80% de los eventos de emergencia sean detectados y notificados en menos de 10 segundos, y cuando se reduzca el tiempo de respuesta de familiares y médicos en un 30%.

- Creemos que al ofrecer una aplicación móvil intuitiva y segura que centralice la información médica de los pacientes, facilitaremos el seguimiento remoto de la salud y mejoraremos la continuidad del tratamiento médico.

Sabremos que hemos tenido éxito

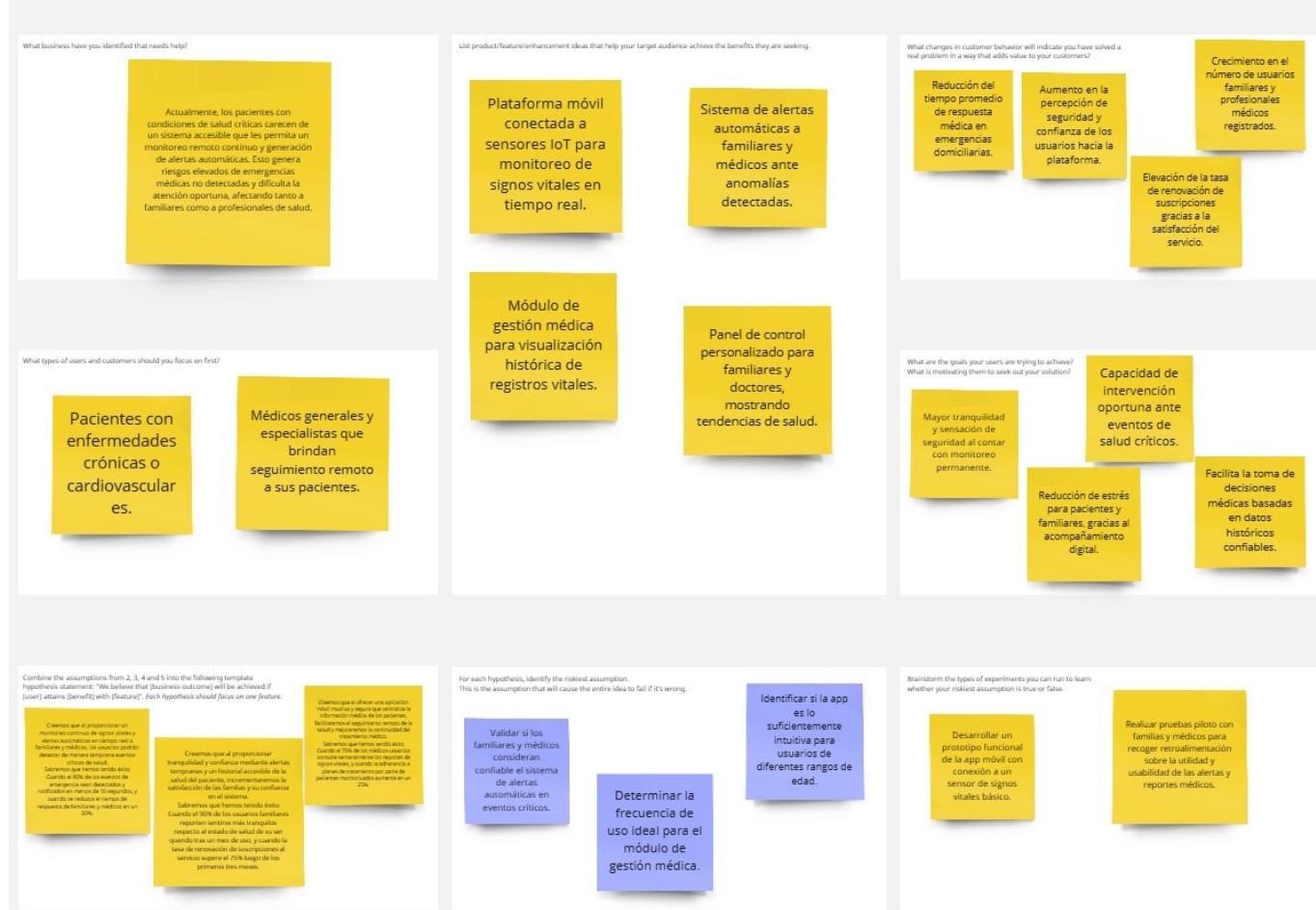
Cuando el 70% de los médicos usuarios consulte semanalmente los reportes de signos vitales, y cuando la adherencia a planes de tratamiento por parte de pacientes monitorizados aumente en un 25%.

- Creemos que al proporcionar tranquilidad y confianza mediante alertas tempranas y un historial accesible de la salud del paciente, incrementaremos la satisfacción de las familias y su confianza en el sistema.

Sabremos que hemos tenido éxito

Cuando el 90% de los usuarios familiares reporten sentirse más tranquilos respecto al estado de salud de su ser querido tras un mes de uso, y cuando la tasa de renovación de suscripciones al servicio supere el 75% luego de los primeros tres meses.

1.2.2.4. Lean UX Canvas.



1.3. Segmentos objetivo.

SafeMed está diseñado para atender dos segmentos principales de usuarios: **pacientes y doctores**.

Cada uno de estos grupos tiene necesidades específicas relacionadas con el monitoreo del ritmo cardíaco y la gestión de la salud en situaciones de emergencia.

- Los **pacientes** son personas que padecen condiciones cardíacas y requieren un seguimiento constante de su ritmo cardíaco para prevenir complicaciones. En caso de detectar anomalías, el sistema puede alertar automáticamente a contactos de emergencia.
- Los **doctores** son profesionales de la salud que supervisan de forma remota el estado cardíaco de sus pacientes. Además de visualizar en tiempo real las mediciones, tienen la capacidad de coordinar la asistencia médica, enviar ambulancias si es necesario y gestionar las citas médicas.

Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores.

2.1.1. Análisis competitivo.

Competitive Analysis Landscape

¿Por qué llevar a cabo este análisis?	El objetivo de este análisis es identificar las características de los competidores y encontrar maneras de diferenciarnos.
---------------------------------------	----------------------------------------------------------------------------------------------------------------------------

Startup y Competidores		MedSystem	Home Medical Management	LOLIMSA	SAMA
Perfil	Overview	Plataforma de gestión de procesos de un consultorio o clínica, que cuenta con diferentes funcionalidades para pacientes, doctores y laboratorios.	Es una compañía que a través de una plataforma digital y sus profesionales, asegura la ejecución y calidad del servicio de salud domiciliaria.	Es una empresa de software doctor con soluciones tecnológicas para la gestión de centros doctores como clínicas y hospitales, farmacias y cadenas de farmacias y clínicas veterinarias.	Es un consorcio de inversión dedicado a brindar servicios integrales de salud, especializados en la seguridad y salud en el trabajo. Contamos con distintas sedes a lo largo del norte del país, las cuales cuentan con el respaldo de la acreditación DIGESA/MINSA.
Ventaja competitiva ¿Qué valor ofrece a los clientes?		Cuenta con distintas funcionalidades como un sistema de gestión de citas en línea que permita a los pacientes y doctores programar fácilmente consultas, programación de cirugías ,exámenes y seguimientos de tratamiento o recuperaciones.	Utilizan inteligencia artificial e innovación para mejorar el servicio a los pacientes y facilitar la atención a los profesionales y prestadores de salud.	Reduce entre 10 y 20% tus costos hospitalarios utilizando las mejores prácticas de Health Management con LOLCLI, el software de gestión hospitalaria más completo de Latinoamérica.	Organización que orienta todos sus esfuerzos en realizar un servicio doctor de calidad a través de la presentación de resultados confiables.
Perfil de Marketing	Mercado objetivo	Consultorio, clínicas y hospitales.	Prestadores de salud, pacientes y personal doctor	Clínicas y hospitales	Hospitales y clínicas
	Estrategias de marketing	Publicidad por redes sociales y campañas.	Publicidad por redes sociales.	Publicidad por redes sociales.	Publicidad por redes sociales.
Perfil de producto	Productos & Servicios	Una plataforma para gestionar las citas	Un software basado en el	Software de gestión	Servicios integrales de

		médicas, programar cirugías, exámenes y seguimiento de tratamientos. Además, la plataforma almacena el historial clínico de los pacientes.	diseño centrado en el paciente, con información integrada, que utiliza la geolocalización y la valoración del servicio, como forma de optimizar la gestión del modelo.	hospitalaria, impulsado por una plataforma de software de última generación que controla al detalle todas las actividades, desde que el paciente ingresa hasta que se retira del establecimiento	salud: - Exámenes doctores ocupacionales - Monitores ocupacionales - Vigilancia médica ocupacional - Análisis de laboratorio - radiografias digitales
Precios & Costos	Varían según características de la clínica, con opciones de suscripción mensual o anual.	Varían según el producto y pueden incluir tarifas de suscripción o costos de licencia.	Los precios varían según el producto y pueden incluir tarifas de suscripción mensuales o costos de licencia.	Los precios varían según el tamaño del hospital o clínica y las características específicas de la solución, con opciones de suscripción mensual o anual	
Canales de distribución (Web y/o Móvil)	Principalmente a través de su sitio web oficial	Principalmente a través de su sitio web, con posibilidad de acceso móvil.	Principalmente a través de su sitio web oficial.	Principalmente a través de su sitio web oficial.	
Análisis SWOT	Fortalezas	Ofrece una solución integral para la gestión de procesos en consultorios y clínicas, lo que incluye funcionalidades específicas para pacientes, doctores y laboratorios. Esta centralización de información puede mejorar la eficiencia operativa y la coordinación entre todas las partes	Ofrece cuidados personalizados en el hogar, generando satisfacción y lealtad. Su enfoque centrado en el paciente y su capacidad de adaptación son ventajas.	Cuenta con amplia experiencia en software doctor y ofrece soluciones integrales y personalizadas respaldadas por un sólido servicio de soporte técnico.	Samma ofrece una variedad de herramientas de bienestar mental y una interfaz amigable que atrae a una amplia gama de usuarios.

involucradas en el proceso de atención médica.

Debilidades	La competencia en el mercado de software de gestión de consultorios y clínicas es intensa, con numerosas empresas que ofrecen soluciones similares.	Limitaciones geográficas y dependencia del personal pueden afectar la expansión y la consistencia de los servicios.	Enfrenta competencia en el mercado de software doctor y puede estar sujeta a dependencia tecnológica y costos asociados con el desarrollo de soluciones.	La competencia en el mercado de aplicaciones de bienestar mental puede dificultar que Samma se destaque, especialmente si enfrenta problemas de usabilidad o dependencia tecnológica.
Oportunidades	La creciente demanda de soluciones digitales en el sector de la salud ofrece una oportunidad para capturar una parte del mercado en constante expansión.	Puede expandir servicios, integrar tecnología y formar asociaciones para llegar a más clientes y mejorar la eficiencia.	Tiene potencial de expansión tanto nacional como internacional, además de oportunidades para desarrollar nuevas funcionalidades y establecer alianzas estratégicas.	La expansión internacional y la colaboración estratégica ofrecen oportunidades para llegar a nuevos mercados y asociarse con empresas o instituciones para aumentar el alcance de la aplicación.
Amenazas	Las preocupaciones sobre la privacidad y seguridad de los datos en el sector de la salud son una amenaza constante.	Competencia, cambios regulatorios y preocupaciones sobre privacidad y seguridad de datos pueden impactar la rentabilidad y confianza en los servicios.	Los posibles cambios regulatorios, preocupaciones sobre seguridad de datos y avances tecnológicos rápidos representan riesgos que podrían afectar su posición en el mercado.	Cambios en la regulación, críticas negativas y avances tecnológicos representan amenazas potenciales para la operación y la reputación de Samma en el mercado.

2.1.2. Estrategias y tácticas frente a competidores.

MedSystem puede destacarse en el mercado ofreciendo una plataforma de gestión de procesos clínicos altamente personalizable y fácil de usar, tanto para pacientes como para profesionales de la salud, diferenciándose así de competidores como Home Medical Management y LOLIMSA. Con un enfoque específico en la atención domiciliaria, la plataforma puede ofrecer funcionalidades especializadas para la coordinación y seguimiento de la atención médica en el hogar, proporcionando comodidad y eficiencia para pacientes y cuidadores. Además, MedSystem puede invertir en el desarrollo de tecnologías avanzadas, como inteligencia artificial y análisis de datos, para mejorar la gestión de consultorios y clínicas, posicionándose como una opción innovadora frente a empresas como LOLIMSA. Para competir con SAMA en el ámbito de la salud ocupacional y la seguridad laboral, MedSystem puede ofrecer módulos especializados y herramientas de cumplimiento normativo, aprovechando su experiencia en la gestión clínica para proporcionar soluciones integrales a empresas e instituciones. Una estrategia de marketing sólida, centrada en la educación del mercado sobre la importancia de la gestión eficiente de consultorios y clínicas, así como en testimonios de usuarios satisfechos, puede ayudar a MedSystem a construir una sólida reputación y atraer clientes potenciales. Además, explorar alianzas estratégicas con instituciones médicas y otros actores del sector puede ampliar su alcance y fortalecer su posición en el mercado. En resumen, MedSystem puede destacarse mediante la combinación de personalización, tecnología avanzada, especialización en atención domiciliaria y salud ocupacional, una sólida estrategia de marketing y alianzas estratégicas, lo que le permitirá diferenciarse y prosperar en un mercado competitivo.

2.2. Entrevistas.

2.2.1. Diseño de entrevistas.

Preguntas para el segmento objetivo paciente:

- *¿Cuáles son los mayores desafíos que enfrentas al cuidar tu salud cuando no estás en el consultorio médico?*
- *¿Qué tipo de dispositivos tecnológicos (como celulares, tablets o computadoras) usas para cuidar o monitorear tu salud? ¿Por qué prefieres esos dispositivos?*
- *¿Qué herramientas o funciones te gustaría tener para poder revisar en tiempo real cosas como tu oxigenación o ritmo cardíaco?*
- *¿Qué características te parecen más importantes al usar una aplicación o tecnología para cuidar tu salud desde casa?*
- *¿Cómo te gustaría recibir alertas o información importante si tu salud está en riesgo?*
- *¿Qué crees que se podría mejorar en la comunicación entre los médicos y tus familiares cuando tú estás en una situación médica delicada?*
- *¿Qué funciones te parecerían útiles en una aplicación que reúna información de sensores de salud (como pulseras o medidores) y que también te ayude con tu cuidado médico?*
- *¿Cómo estás cuidando tu salud actualmente si tienes una enfermedad crónica o estás en recuperación fuera del hospital?*
- *¿Qué tan importante es para ti la seguridad y privacidad de tus datos médicos cuando se recopilan con dispositivos tecnológicos?*
- *¿Qué te gustaría que mejorara en la forma en que puedes ver o compartir tu historial médico, especialmente si incluye datos en tiempo real?*
- *¿Qué piensas sobre el uso de tecnología como sensores o dispositivos conectados para mejorar tu atención médica o prevenir enfermedades?*

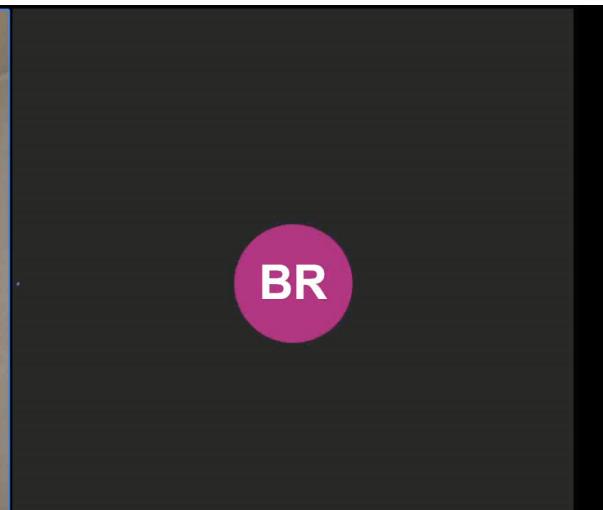
Preguntas para el segmento objetivo doctor:

- *¿Cuáles son los mayores desafíos que enfrentas al hacer seguimiento del estado de salud de tus pacientes fuera del consultorio?*

- ¿Qué tipo de dispositivos tecnológicos (móviles, tablets, computadoras) usas para el seguimiento de pacientes? ¿Por qué prefieres esos dispositivos?
- ¿Qué herramientas o funcionalidades te serían útiles para monitorear signos vitales como oxigenación y ritmo cardíaco en tiempo real?
- ¿Qué características consideras más importantes al elegir una solución tecnológica para el monitoreo remoto de pacientes?
- ¿Cómo prefieres recibir alertas o información crítica sobre un paciente en situación de riesgo?
- ¿Qué aspectos te gustaría mejorar en la comunicación con los familiares de tus pacientes en situaciones médicas delicadas?
- ¿Qué funcionalidades consideras cruciales en una plataforma que integre datos de sensores IoT y gestión médica?
- ¿Cómo manejas actualmente la continuidad del cuidado de pacientes crónicos o en recuperación fuera del hospital?
- ¿Qué medidas de seguridad y privacidad consideras esenciales al manejar datos médicos recolectados por dispositivos IoT?
- ¿Qué mejoras te gustaría ver en el acceso y documentación del historial clínico relacionado con datos en tiempo real?
- ¿Cuál es tu opinión sobre el uso de tecnología IoT para mejorar la atención y la prevención en medicina?

2.2.2. Registro de entrevistas.

Entrevistas Segmento Pacientes

Entrevista	Datos
Nombre	Brian Roca
Edad	24 años
Distrito	Surco
Imagen	 
Resumen	<p>Brian es un joven de 24 años que vive en Surco. Desde pequeño padece problemas congénitos del corazón. Comenta que, aunque reconoce cuándo se siente mal, a veces un ataque ocurre de manera inesperada, complicando el contacto rápido con su médico o familiares. Le sería muy útil una aplicación conectada a un sensor para monitorear su ritmo cardíaco y alertar tanto a él como a sus familiares en caso de peligro, incluso si no siente síntomas.</p>

Entrevista Datos**URL de Grabación** [Grabación Brian](#)**Timing** 00:00 - 11:15**Entrevista Datos****Nombre** Jorge De la Piedra**Edad** 64 años**Distrito** Surco**Imagen****Resumen** Jorge es un señor de 64 años que sufre de problemas de presión relacionados con su corazón. Utiliza dispositivos como el Apple Watch para monitorear su ritmo cardiaco, pero expresa la necesidad de una app que facilite un seguimiento más preciso, ya que una emergencia puede surgir en cualquier momento, afectando tanto su salud como la tranquilidad de su familia.**URL de Grabación** [Grabación Jorge](#)**Timing** 11:16 - 19:53**Entrevista Datos****Nombre** Renzo Ramos**Edad** 22 años**Distrito** Santiago de Surco**Imagen**

Entrevista	Datos
Resumen	Renzo, de 22 años, vive en Santiago de Surco. Su principal preocupación es la estabilidad de su estado de salud, por lo que usa smartwatch y celular para su monitoreo. Quiere apps con gráficos simples, integración fluida a su rutina diaria y notificaciones tanto para él como para sus familiares en caso de emergencia. Además, valora funcionalidades como gestión de medicamentos y análisis de signos vitales.
URL de Grabación	Grabación Renzo
Timing	19:54 - 25:09

Entrevistas Segmento Médico

Entrevista	Datos
Nombre	Maryori Espinoza
Edad	30 años
Distrito	Santa Anita
Imagen	 Imagen 4
Resumen	Maryori, doctora en el Hospital San Isidro Labrador, enfrenta desafíos como la falta de datos en tiempo real y asegurar el cumplimiento de tratamientos. Prefiere soluciones accesibles, que se integren a la historia clínica, garanticen la seguridad de datos y sean fáciles de usar tanto para médicos como pacientes. Valora las alertas con resúmenes claros.
URL de Grabación	Grabación Maryori
Timing	25:09 - 34:36
Entrevista	Datos
Nombre	Jean Piere Grandez Mansilla
Edad	25 años
Distrito	Callao

Entrevista Datos**Imagen**

Resumen Jean Pierre, interno de medicina, destacó la falta de información y cumplimiento de tratamientos como los principales problemas al monitorear pacientes. Propone integrar dispositivos IoT, mejorar la comunicación familiar y médica, y garantizar la seguridad de datos en las plataformas de monitoreo remoto.

URL de Grabación [Grabación Jean Pierre](#)

Timing 34:36 - 41:57

Entrevista Datos

Nombre Matías Nolte

Edad 22 años

Distrito Santiago de Surco

Entrevista Datos**Imagen**

Resumen Matías, joven con antecedentes cardíacos, necesita monitoreo continuo para su tranquilidad. Prefiere alertas automáticas a su celular y la notificación inmediata a sus padres. También desea una app que gestione tanto signos vitales como estados de ánimo, ayudándolo a identificar patrones de bienestar general.

URL de Grabación [Grabación Matías](#)

Timing 42:08 - 49:47

2.2.3. Análisis de entrevistas.

Segmento Objetivo #1: Paciente

Los pacientes entrevistados han identificado consistentemente la mala organización y los largos tiempos de espera como problemas principales en la programación de citas médicas. Estas preocupaciones son comunes en todos los entrevistados, lo que sugiere que una solución que ofrezca una mejor gestión de citas sería bien recibida. Además,

un patrón emergente es la preferencia por la flexibilidad en la elección entre consultas virtuales o presenciales, dependiendo de la gravedad de la situación. Este aspecto refleja una necesidad de personalización y adaptación a las circunstancias individuales de los pacientes.

Segmento Objetivo #2: Médico

Los médicos entrevistados han destacado de manera uniforme la necesidad de herramientas tecnológicas que integren eficientemente la gestión de citas, historias clínicas electrónicas, y la comunicación con otros profesionales. Un aspecto recurrente en las entrevistas es la priorización de la seguridad de la información y la facilidad de uso al elegir estas herramientas, lo que indica que cualquier solución tecnológica debe cumplir con estos criterios para ser considerada viable en un entorno médico. Además, la telemedicina se menciona como un recurso valioso, pero su implementación efectiva depende de una formación adecuada, subrayando la importancia de capacitar a los profesionales en el uso de nuevas tecnologías. Esta información es fundamental para desarrollar un arquetipo de médico que valore la eficiencia, la seguridad y la innovación en su práctica diaria.

2.3. Needfinding.

2.3.1. User Personas.

Los Users personas que se muestran a continuación, fueron realizados a partir de la información recopilada de la sección de entrevistas. Estos nos ayudarán a describir de forma general nuestro segmento objetivo.

PERSONA: Marco Suarez

NAME	TYPE
Marco Suarez	Rational
Goals <ul style="list-style-type: none"> Encontrar algún software para realizar citas médicas. Poder monitorear su tratamiento de la mano de un especialista. 	
Quote No espero que las oportunidades aparezcan, por el contrario, creas tu entorno.	
Background Marco Suarez es un paciente deportista que sufrió una lesión en miembro practicando su deporte favorito. Y lleva un tiempo sin estar al ciento por ciento debido al sistema de salud del Perú.	
Demographic Male 30 years Lima Single Transportista	Motivations El deseo de mantener física y mentalmente anima a seguir adelante en mi proceso de recuperación. Tener la facilidad de poder ver los resultados de sus exámenes médicos.
Technology 	Frustrations La lesión ha impactado todos los aspectos de mi vida diaria, generando varios inconvenientes.
Browsers 	Brands and influencers
Channels 	

UXPRESSIA
User persona analysis tool

• Paciente:

PERSONA: Luis Torres

NAME: Luis Torres **Type:** Idealist

Demographic

- Male 40 years
- Lima
- Hired Doctor

Skills

- Resolución de problemas: 100
- Empatía: 100
- Liderazgo: 100

Motivations

- Mi compromiso esco me motiva a proporcionar atención médica de calidad a todos los pacientes.
- El objetivo personal de diagnosticar y tratar lesiones traumáticas y proporcionar la mejor terapia posible.

Frustrations

- La falta de recursos en el sistema de salud pública dificulta la atención a todos los pacientes.
- El exceso de trabajo impone a tener una agenda tan intensa que limita la atención individualizada y efectiva.

Brands and influencers

- GE HealthCare
- MERIVAARA Health Technology with a Human Touch
- BRAINLAB

Technology

- Smartphone
- Tablet
- PC

Browsers

- Safari

Channels

- WhatsApp
- Zoom

UXPRESSIA
This persona was built in uxpresa.com

- Doctor:

2.3.2. User Task Matrix.

En esta sección se presenta el user task matrix de los segmentos objetivos, con el fin de identificar la frecuencia de las actividades realizadas por los usuarios, y de esta manera se refleja la importancia de determinadas tareas.

Task	Pacientes	Profesionales doctores	Laboratorio			
	Frecuencia	Importancia	Frecuencia	Importancia	Frecuencia	Importancia
Programar una cita	Alta	Alta	Alta	Alta	-	-
Acceder a resultados de exámenes	Media	Alta	Alta	Alta	Alta	Alta
Realizar seguimiento de tratamiento	Alta	Alta	Alta	Alta	-	-

Task	Pacientes	Profesionales doctores	Laboratorio			
Agendar una cirugía	Baja	Alta	Media	Alta	-	-
Participar en consultas virtuales	Baja	Alta	Media	Alta	-	-
Recibir recordatorios automáticos de citas	Alta	Alta	Alta	Alta	-	-
Acceder a historias clínicas	-	-	Alta	Alta	-	-
Consultar información sobre medicamentos	Media	Alta	Media	Alta	-	-
Comunicarse con el especialista	Alta	Alta	Media	Alta	Alta	Alta
Proporcionar retroalimentación sobre la experiencia	Baja	Media	-	-	-	-
Solicitar análisis clínicos	Baja	Alta	-	-	Alta	Alta
Enviar resultado de análisis	-	-	-	-	Alta	Alta

En base al User Task Matrix presentado, podemos destacar las siguientes tareas con mayor frecuencia e importancia para cada segmento de usuarios:

Pacientes:

- Programar una cita
 - Explicación: Los pacientes programan citas con frecuencia y consideran esta tarea crucial para recibir atención médica oportuna.
- Realizar seguimiento de tratamiento
 - Explicación: Mantenerse al día con su tratamiento es vital para los pacientes, lo que refleja la alta frecuencia y la importancia atribuida a esta tarea.

Profesionales doctores:

- Acceder a resultados de exámenes
 - Explicación: Los profesionales doctores necesitan consultar los resultados de los exámenes con frecuencia para tomar decisiones informadas sobre el tratamiento de sus pacientes.
- Acceder a historias clínicas
 - Explicación: Las historias clínicas proporcionan antecedentes doctores completos y son esenciales para el diagnóstico y tratamiento adecuado de los pacientes.

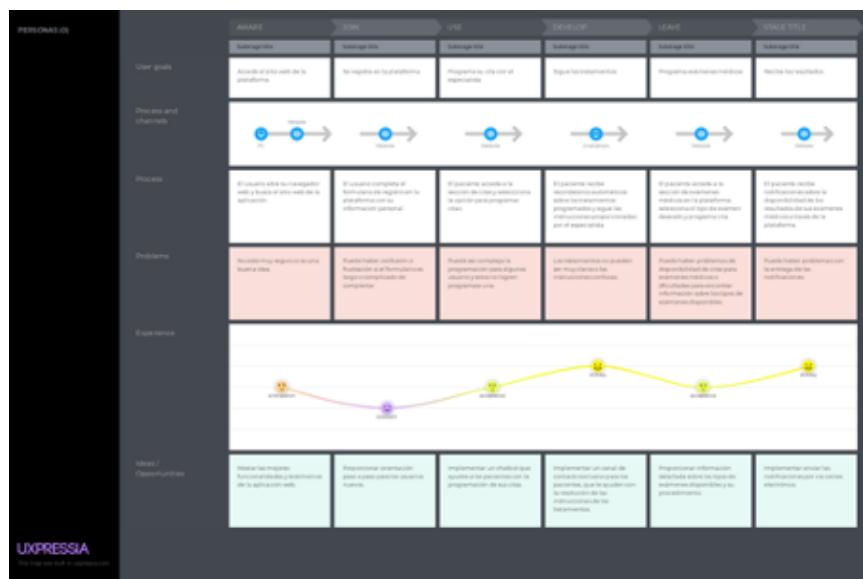
Principales diferencias y coincidencias

- Coincidencias:
 - Tanto los pacientes como los profesionales doctores consideran que programar una cita es una tarea de alta frecuencia e importancia.
 - Acceder a resultados de exámenes es crucial tanto para los profesionales doctores como para los laboratorios, reflejando una alta frecuencia e importancia en ambas categorías.
- Diferencias:
 - Los pacientes valoran altamente el seguimiento de su tratamiento y recibir recordatorios automáticos de citas, mientras que los profesionales doctores ponen más énfasis en el acceso a historias clínicas y consultas virtuales.

2.3.3. User Journey Mapping.

En esta sección se presentan los User Journey Mapping de los segmentos objetivos, que realizamos con el fin de dar a entender cómo se siente nuestro usuario al usar la aplicación, detallando cada paso que realiza y las emociones que experimenta.

Paciente:

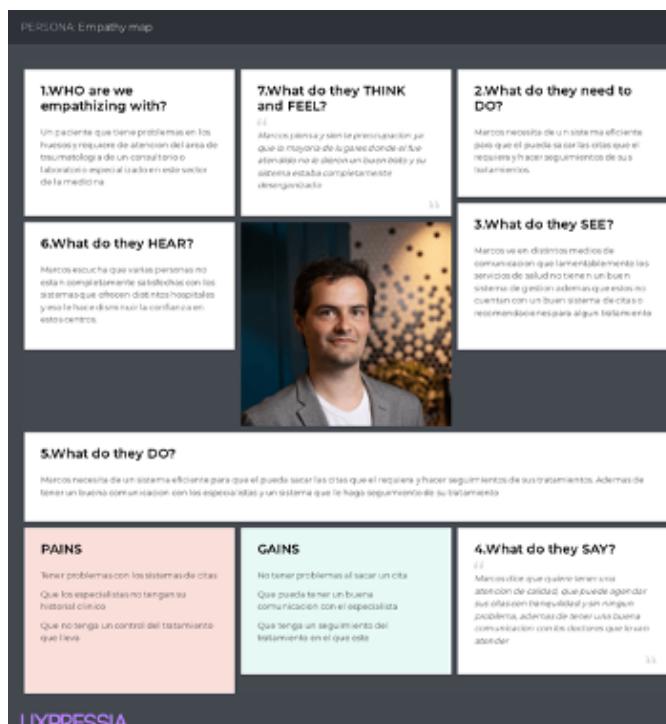


Doctor:



2.3.4. Empathy Mapping.

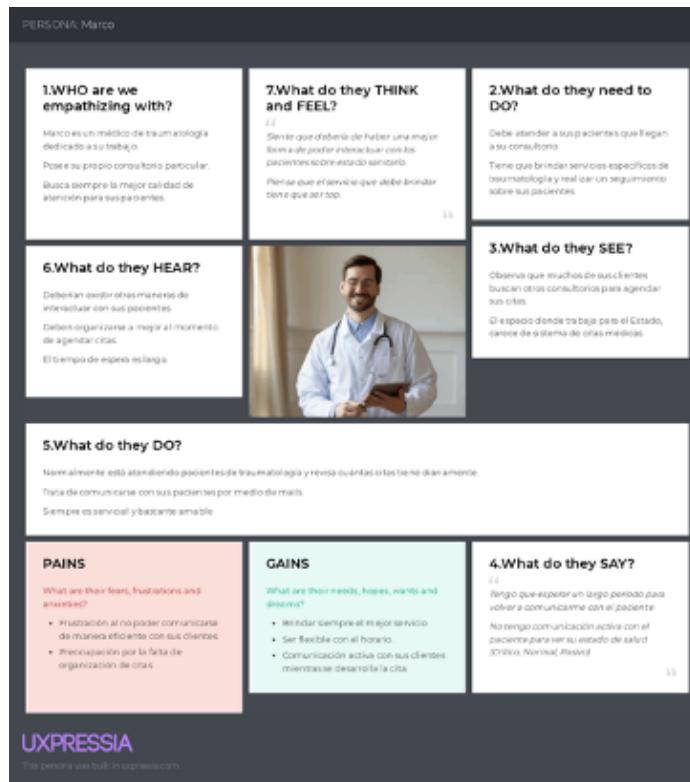
En esta sección mostramos los empathy mapping de los segmentos objetivos realizados con la información recopilada de componentes anteriores.



Paciente:

IMAGEN

Doctor:



IMAGEN

2.3.5. As-is Scenario Mapping.

En esta sección se realizaron las distintas etapas de preparación, tales como la lluvia de ideas individual, revisión e identificación de fases como columnas, para conseguir los As-is Scenario Mapping de los segmentos objetivos.

Para el segmento 1 (Pacientes):



IMAGEN

Para el segmento 2 (Doctores):



IMAGEN

Capítulo III: Requirements Specification

3.1. To-Be Scenario Mapping.

- Segmento Objetivo: Paciente

Phase	Doing	Thinking	Feeling
Monitoreo cardiaco	Consulta el ritmo cardiaco en tiempo real desde la app	¿Estoy bien o necesito atención médica?	Aliviado si todo está normal, ansioso si hay anomalías
Recepción de alertas	Recibe una notificación si su ritmo está fuera de los umbrales normales	¿Es esto urgente? ¿Debo contactar a alguien?	Preocupado pero agradecido por el aviso
Solicitud de cita	Elige fecha y hora para una consulta médica desde la app	Necesito hablar con el doctor lo antes posible	Esperanzado de resolver el problema pronto

- Segmento Objetivo: Doctor

Phase	Doing	Thinking	Feeling
Revisión de pacientes	Revisa los signos vitales de sus pacientes en tiempo real	¿Cuál paciente requiere atención inmediata?	En control, responsable
Gestión de alertas	Recibe y prioriza alertas críticas según el nivel de riesgo	Debo actuar rápido si hay riesgo alto	Presionado pero concentrado
Consulta médica	Revisa el historial del paciente y realiza el seguimiento en consulta	¿Qué ha pasado desde la última revisión?	Comprometido con la salud del paciente

3.2. User Stories.

- Epics

Epic ID	Título	Descripción
01	Monitoreo en tiempo real del ritmo cardiaco	Como doctor, quiero poder visualizar en una plataforma web el ritmo cardiaco de mis pacientes en tiempo real, para estar al tanto de cualquier irregularidad y actuar de manera oportuna.
02	Alerta automática por ritmo cardiaco elevado	Como doctor, quiero recibir una alerta automática en la plataforma web cuando el ritmo cardiaco de un paciente sea irregular y poder mandar una ambulancia a su ubicación.
03	Notificación por SMS a contacto de emergencia	Como paciente, quiero que un familiar o contacto de emergencia reciba un mensaje SMS si mi ritmo cardiaco presenta una anomalía, para que puedan ayudarme rápidamente.
04	Aplicación móvil para pacientes	Como paciente, quiero tener una aplicación móvil donde pueda monitorear mi ritmo cardiaco, visualizar alertas y gestionar mis citas médicas, para tener control sobre mi salud desde cualquier lugar.
05	Gestión de citas médicas entre paciente y doctor	Como paciente, quiero poder agendar citas médicas desde mi app móvil, y como doctor quiero visualizarlas desde la plataforma web, para organizar mejor el seguimiento clínico.

- User Stories

Epic 01 - Monitoreo en tiempo real del ritmo cardiaco

US	Título	Descripción	Criterios de aceptación	Epic relacionada

US	Título	Descripción	Criterios de aceptación	Epic relacionada
01	Visualizar ritmo cardiaco en tiempo real	Como doctor, quiero poder ver el ritmo cardiaco actual de mis pacientes en una interfaz web, para monitorear su estado de salud constantemente.	<p>Escenario 1: Acceso al monitoreo en tiempo real</p> <p>Dado que estoy logueado como doctor,</p> <p>Cuando ingreso a la sección de monitoreo cardiaco,</p> <p>Entonces puedo ver una gráfica en tiempo real del ritmo cardiaco del paciente seleccionado.</p>	01
02	Seleccionar paciente para monitoreo	Como doctor, quiero poder seleccionar un paciente específico, para ver únicamente su información de ritmo cardiaco.	<p>Escenario 1: Filtro de paciente</p> <p>Dado que tengo varios pacientes asignados,</p> <p>Cuando selecciono uno en la lista,</p> <p>Entonces solo veo el ritmo cardiaco de ese paciente en la gráfica.</p> <p>Escenario 2: Confirmación de selección</p> <p>Dado que he hecho clic en un paciente,</p> <p>Cuando se actualice la gráfica,</p> <p>Entonces debe mostrarse el nombre del paciente seleccionado como confirmación.</p>	01

US	Título	Descripción	Criterios de aceptación	Epic relacionada
03	Actualización automática de datos	Como doctor, quiero que la información del ritmo cardiaco se actualice automáticamente, para no tener que refrescar la página.	<p>Escenario 1: Actualización continua</p> <p>Dado que estoy viendo la gráfica,</p> <p>Cuando pasan unos segundos,</p> <p>Entonces los valores deben actualizarse sin intervención manual.</p> <p>Escenario 2: Aviso de interrupción</p> <p>Dado que la actualización está activa,</p> <p>Cuando falle la conexión temporalmente,</p> <p>Entonces debo ver una alerta indicando que los datos no se están actualizando.</p>	01
04	Ver estado de conexión del sensor	Como doctor, quiero saber si el sensor del paciente está enviando datos correctamente, para asegurarme de que la información sea confiable.	<p>Escenario 1: Indicador de conexión</p> <p>Dado que estoy monitoreando a un paciente,</p> <p>Cuando el sensor esté desconectado o sin datos,</p> <p>Entonces debo ver un mensaje de advertencia.</p> <p>Escenario 2: Confirmación de conexión activa</p> <p>Dado que el sensor funciona correctamente,</p> <p>Cuando se establezca la conexión,</p> <p>Entonces debe visualizarse un ícono o mensaje de estado activo.</p>	01

Epic 02 - Alerta automática por ritmo elevado

US	Título	Descripción	Criterios de aceptación	Epic relacionada
----	--------	-------------	-------------------------	------------------

US	Título	Descripción	Criterios de aceptación	Epic relacionada
01	Alerta visual en la interfaz	Como doctor, quiero recibir una alerta visual en la pantalla si el ritmo cardiaco de un paciente es elevado, para reaccionar rápidamente.	<p>Escenario 1: Activación de alerta</p> <p>Dado que monitoreo a un paciente,</p> <p>Cuando su ritmo cardiaco supere un umbral crítico,</p> <p>Entonces una alerta debe aparecer en pantalla.</p> <p>Escenario 2: Alerta persistente</p> <p>Dado que una alerta visual fue activada,</p> <p>Cuando no se ha confirmado aún,</p> <p>Entonces esta debe permanecer visible hasta su revisión.</p>	02
02	Configurar umbrales de alerta	Como doctor, quiero poder configurar los valores máximos y mínimos del ritmo cardiaco, para personalizar las alertas según el paciente.	<p>Escenario 1: Edición de umbrales</p> <p>Dado que estoy visualizando un paciente,</p> <p>Cuando ingreso a la configuración,</p> <p>Entonces puedo definir sus límites de alerta.</p> <p>Escenario 2: Validación de rango</p> <p>Dado que configuro nuevos valores,</p> <p>Cuando ingreso un umbral inválido,</p> <p>Entonces debo ver un mensaje de error impidiendo guardar los cambios.</p>	02

US	Título	Descripción	Criterios de aceptación	Epic relacionada
03	Alerta sonora opcional	Como doctor, quiero poder activar una alarma sonora al recibir una alerta, para no perderme ninguna notificación urgente.	<p>Escenario 1: Alarma activada</p> <p>Dado que tengo una alerta activa,</p> <p>Cuando la opción de sonido está activada,</p> <p>Entonces escucho una alarma.</p> <p>Escenario 2: Alarma desactivada</p> <p>Dado que tengo la opción de sonido desactivada,</p> <p>Cuando ocurre una alerta,</p> <p>Entonces no debe reproducirse ningún sonido.</p>	02
04	Registro de alertas anteriores	Como doctor, quiero revisar las alertas previas que ha generado un paciente, para entender mejor sus antecedentes.	<p>Escenario 1: Revisión de historial</p> <p>Dado que selecciono un paciente,</p> <p>Cuando accedo a su historial de alertas,</p> <p>Entonces puedo ver fechas, horas y tipos de alerta.</p> <p>Escenario 2: Orden cronológico</p> <p>Dado que visualizo las alertas previas,</p> <p>Cuando se muestren en la pantalla,</p> <p>Entonces deben aparecer ordenadas desde la más reciente a la más antigua.</p>	02

Epic 03 - Notificación por SMS a contacto asignado

US	Título	Descripción	Criterios de aceptación	Epic relacionada
----	--------	-------------	-------------------------	------------------

US	Título	Descripción	Criterios de aceptación	Epic relacionada
01	Registrar contacto de emergencia	Como paciente, quiero poder registrar a una persona de contacto en la aplicación, para que reciba alertas por SMS si ocurre una emergencia.	<p>Escenario 1: Registro de contacto</p> <p>Dado que estoy en mi perfil de paciente,</p> <p>Cuando ingreso los datos del contacto de emergencia,</p> <p>Entonces estos quedan guardados y vinculados a mi cuenta.</p> <p>Escenario 2: Validación de campos</p> <p>Dado que estoy completando el formulario,</p> <p>Cuando dejo un campo obligatorio vacío,</p> <p>Entonces debo ver un mensaje de error que me impida continuar.</p>	03

US	Título	Descripción	Criterios de aceptación	Epic relacionada
02	Envío automático de SMS en alerta	Como paciente, quiero que se envíe un mensaje SMS automáticamente a mi contacto registrado cuando se detecte un ritmo cardiaco elevado, para asegurarme de que alguien de confianza sea notificado en caso de emergencia.	<p>Escenario 1: Dado que mi ritmo cardiaco excede el umbral, Cuando se detecta esta condición, Entonces se debe enviar automáticamente un SMS al contacto que registré previamente.</p> <p>Escenario 2: Dado que se ha enviado un SMS, Cuando la alerta es procesada, Entonces debo recibir una notificación indicando que el mensaje fue entregado correctamente.</p>	03

US	Título	Descripción	Criterios de aceptación	Epic relacionada
03	Personalizar mensaje de emergencia	Como paciente, quiero poder personalizar el contenido del mensaje SMS, para que el contacto entienda mejor la situación.	<p>Escenario 1: Edición del mensaje</p> <p>Dado que estoy configurando la función de emergencia,</p> <p>Cuando ingreso el texto personalizado,</p> <p>Entonces ese mensaje será el que se envíe al contacto en caso de alerta.</p> <p>Escenario 2: Revisión del mensaje</p> <p>Dado que guardé un mensaje personalizado,</p> <p>Cuando vuelve a la configuración,</p> <p>Entonces debe mostrarse el mensaje previamente guardado.</p>	03

US	Título	Descripción	Criterios de aceptación	Epic relacionada
04	Ver historial de mensajes enviados	Como paciente, quiero revisar los SMS que han sido enviados por el sistema, para tener un registro de las alertas enviadas a mi contacto.	<p>Escenario 1: Consulta de historial</p> <p>Dado que accedo a la sección de emergencias,</p> <p>Cuando reviso el historial,</p> <p>Entonces puedo ver la fecha, hora y contenido del mensaje enviado.</p> <p>Escenario 2: Filtro por fecha</p> <p>Dado que tengo varios mensajes en el historial,</p> <p>Cuando ingreso un rango de fechas,</p> <p>Entonces solo deben mostrarse los mensajes dentro de ese periodo.</p>	03

Epic 04 - Aplicación móvil para pacientes

US	Título	Descripción	Criterios de aceptación	Epic relacionada
----	--------	-------------	-------------------------	------------------

US	Título	Descripción	Criterios de aceptación	Epic relacionada
01	Visualizar datos personales y médicos	Como paciente, quiero poder ver mis datos personales y médicos en la app, para tener control sobre mi información.	<p>Escenario 1: Acceso a perfil</p> <p>Dado que estoy logueado en la app móvil,</p> <p>Cuando accedo a la sección "Mi perfil",</p> <p>Entonces puedo visualizar mis datos personales y médicos actualizados.</p> <p>Escenario 2: Datos no disponibles</p> <p>Dado que hay un error de sincronización,</p> <p>Cuando intento acceder a mi perfil,</p> <p>Entonces debo ver un mensaje de error explicativo.</p>	04
02	Ver ritmo cardiaco desde el celular	Como paciente, quiero poder ver mi ritmo cardiaco actual desde la aplicación móvil, para monitorear mi estado de salud.	<p>Escenario 1: Visualización en tiempo real</p> <p>Dado que tengo el sensor conectado,</p> <p>Cuando ingreso a la app,</p> <p>Entonces puedo ver una gráfica con mi ritmo cardiaco actual.</p> <p>Escenario 2: Sensor desconectado</p> <p>Dado que el sensor no está conectado,</p> <p>Cuando ingreso a la sección de ritmo cardiaco,</p> <p>Entonces debo ver un mensaje indicando la falta de conexión.</p>	04

US	Título	Descripción	Criterios de aceptación	Epic relacionada
03	Recibir notificaciones de alerta	Como paciente, quiero recibir notificaciones push si se detecta un ritmo anormal, para estar informado de cualquier cambio importante.	<p>Escenario 1: Notificación activa Dado que mi ritmo cardiaco supera el límite, Cuando se detecta la anomalía, Entonces recibo una alerta en el teléfono móvil.</p> <p>Escenario 2: Reintentó de envío Dado que hubo un fallo en el envío de la notificación, Cuando se restablezca la conexión, Entonces el sistema intentará reenviar la alerta automáticamente.</p>	04
04	Configurar preferencias de notificación	Como paciente, quiero configurar mis preferencias de notificación en la app móvil, para decidir cómo y cuándo recibir alertas sobre mi ritmo cardiaco.	<p>Escenario 1: Ajustar preferencias Dado que estoy en la sección de configuración, Cuando modiflico mis opciones de notificación, Entonces el sistema guarda mis preferencias para futuras alertas.</p> <p>Escenario 2: Preferencias restauradas Dado que restablezco la configuración, Cuando confirmo la acción, Entonces se aplican los valores por defecto.</p>	04

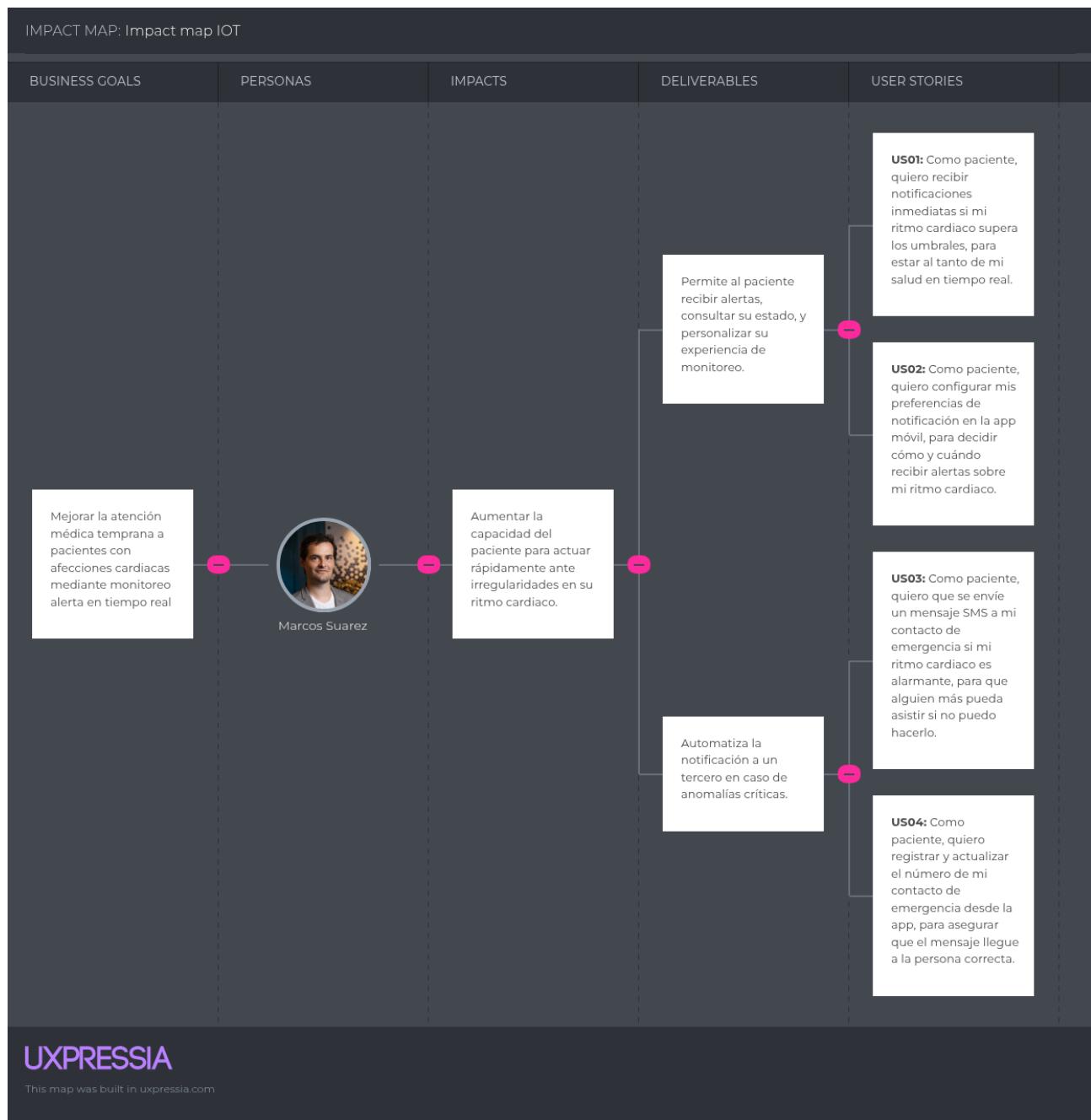
Epic 05 - Gestión de citas médicas desde la app

US	Título	Descripción	Criterios de aceptación	Epic relacionada
01	Agendar una cita desde la app	Como paciente, quiero poder agendar una cita médica desde mi aplicación móvil, para facilitar la programación de consultas.	<p>Escenario 1: Crear nueva cita</p> <p>Dado que estoy en la app móvil,</p> <p>Cuando ingreso a la sección de citas y selecciono una fecha y hora,</p> <p>Entonces se agenda la cita en el sistema.</p>	05
02	Ver citas programadas	Como paciente, quiero consultar mis citas futuras, para saber cuándo debo asistir a consulta.	<p>Escenario 2: Validación de campos</p> <p>Dado que ingreso a la sección de agendar,</p> <p>Cuando no selecciono fecha u hora,</p> <p>Entonces debe mostrarse un mensaje de error.</p> <p>Escenario 1: Consultar agenda</p> <p>Dado que tengo citas agendadas,</p> <p>Cuando ingreso a la sección de citas,</p> <p>Entonces puedo ver la lista de fechas, horas y médicos asignados.</p> <p>Escenario 2: Agenda vacía</p> <p>Dado que no tengo citas registradas,</p> <p>Cuando accedo a la sección de citas,</p> <p>Entonces debo ver un mensaje indicando que no hay citas programadas.</p>	05

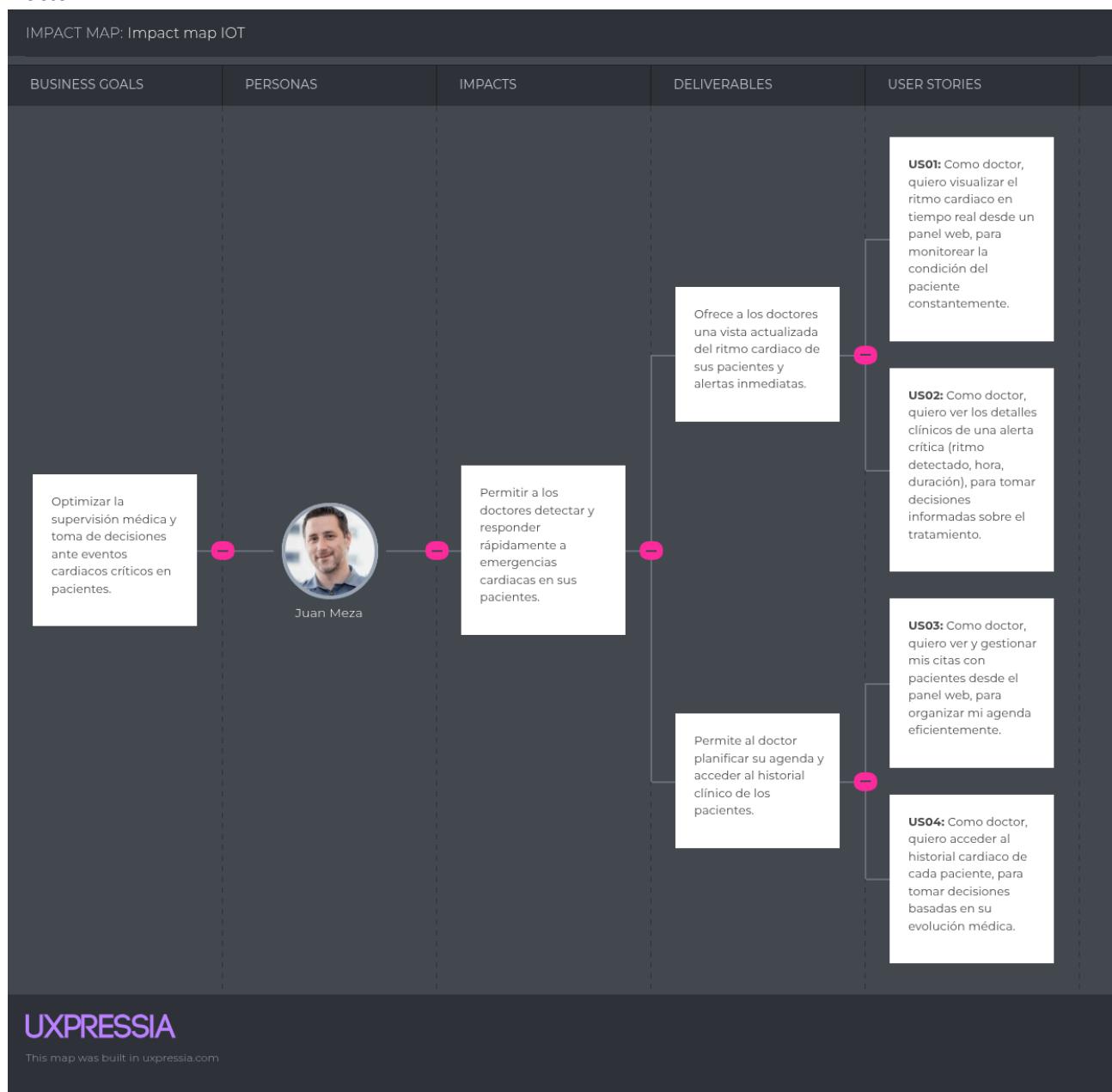
US	Título	Descripción	Criterios de aceptación	Epic relacionada
03	Cancelar o reprogramar una cita	Como paciente, quiero poder cancelar o reprogramar una cita, para reorganizar mi agenda si tengo algún imprevisto.	<p>Escenario 1: Modificar cita</p> <p>Dado que tengo una cita agendada,</p> <p>Cuando accedo a su detalle,</p> <p>Entonces puedo modificar la fecha/hora o cancelar si es necesario.</p>	05
04	Sincronización con agenda del doctor	Como doctor, quiero ver las citas programadas por los pacientes desde la app web, para organizar mejor mi horario.	<p>Escenario 2: Confirmación de cambios</p> <p>Dado que realizo un cambio en la cita,</p> <p>Cuando lo guardo,</p> <p>Entonces debo recibir una confirmación visual del cambio o cancelación.</p> <p>Escenario 1: Visualización de agenda</p> <p>Dado que los pacientes agendan citas desde la app,</p> <p>Cuando ingreso a la sección de agenda como doctor,</p> <p>Entonces veo todas las citas asignadas en tiempo real.</p> <p>Escenario 2: Filtro por fecha</p> <p>Dado que hay múltiples citas en mi agenda,</p> <p>Cuando selecciono una fecha específica,</p> <p>Entonces solo veo las citas de ese día.</p>	05

3.3. Impact Mapping.

- Paciente



- Doctor



3.4. Product Backlog.

Orden	Epic ID	User Story ID	Título	Descripción	Story Points
1	EP01	US01	Visualizar ritmo cardíaco en tiempo real	Como doctor, quiero visualizar el ritmo cardíaco de mis pacientes en tiempo real, para monitorear su estado de salud.	5
2	EP01	US02	Seleccionar paciente para monitoreo	Como doctor, quiero seleccionar a un paciente específico, para revisar únicamente su información de ritmo cardíaco.	3
3	EP01	US03	Actualización automática de datos	Como doctor, quiero que los datos del ritmo cardíaco se actualicen automáticamente, para evitar tener que recargar la vista.	3

Orden	Epic ID	User Story ID	Título	Descripción	Story Points
4	EP01	US04	Ver estado de conexión del sensor	Como doctor, quiero saber si el sensor está enviando datos correctamente, para garantizar que el monitoreo es fiable.	2
5	EP02	US01	Alerta visual en la interfaz	Como doctor, quiero recibir una alerta visual si un paciente presenta un ritmo elevado, para poder actuar de inmediato.	5
6	EP02	US02	Confirmación manual de alertas críticas	Como doctor, quiero confirmar manualmente que he visto una alerta crítica, para asegurar que no se pase por alto ninguna situación urgente.	3
7	EP02	US03	Notificación por correo electrónico	Como doctor, quiero recibir alertas importantes también por correo electrónico, para asegurarme de no perderme notificaciones críticas.	2
8	EP02	US04	Registro de alertas anteriores	Como doctor, quiero revisar las alertas anteriores de un paciente, para tener un historial de sus eventos críticos.	3
9	EP03	US01	Registrar contacto de emergencia	Como paciente, quiero registrar un contacto de emergencia, para que sea notificado si tengo una emergencia cardiaca.	3
10	EP03	US02	Envío automático de SMS en alerta	Como paciente, quiero que se envíe un SMS automático a mi contacto en caso de ritmo elevado, para que me ayuden rápido.	5
11	EP03	US03	Personalizar mensaje de emergencia	Como paciente, quiero personalizar el mensaje SMS de alerta, para que mi contacto entienda mejor la situación.	2
12	EP03	US04	Ver historial de mensajes enviados	Como paciente, quiero ver los SMS que se han enviado a mi contacto, para llevar un registro de las alertas.	2
13	EP04	US01	Visualizar datos personales y médicos	Como paciente, quiero ver mis datos médicos y personales en la app, para tener acceso a mi información de salud.	2
14	EP04	US02	Ver ritmo cardiaco desde el celular	Como paciente, quiero visualizar mi ritmo cardiaco actual en la app móvil, para monitorear mi estado de salud.	5
15	EP04	US03	Recibir notificaciones de alerta	Como paciente, quiero recibir notificaciones push si tengo un ritmo cardiaco anormal, para estar al tanto de emergencias.	3

Orden	Epic ID	User Story ID	Título	Descripción	Story Points
16	EP04	US04	Configurar preferencias de notificación	Como paciente, quiero configurar mis preferencias de notificación en la app móvil, para decidir cómo y cuándo recibir alertas sobre mi ritmo cardiaco.	2
17	EP05	US01	Agendar una cita desde la app	Como paciente, quiero agendar citas médicas desde la app móvil, para facilitar la planificación de mis consultas.	3
18	EP05	US02	Ver citas programadas	Como paciente, quiero ver mis citas médicas futuras, para no olvidar mis consultas.	2
19	EP05	US03	Cancelar o reprogramar una cita	Como paciente, quiero modificar o cancelar una cita desde la app, para gestionar mejor mi agenda.	3
20	EP05	US04	Sincronización con agenda del doctor	Como doctor, quiero ver las citas que agendan mis pacientes, para tener mi agenda médica organizada.	3

Capítulo IV: Solution Software Design

4.1. Strategic-Level Domain-Driven Design.

4.1.1. EventStorming.

Dentro de este punto nos encontramos con estos 9 pasos a realizar:

Step 1: Unstructured Exploration

En este paso, se detectaron los siguientes puntos, los cuales posteriormente fueron utilizados dentro del sistema.

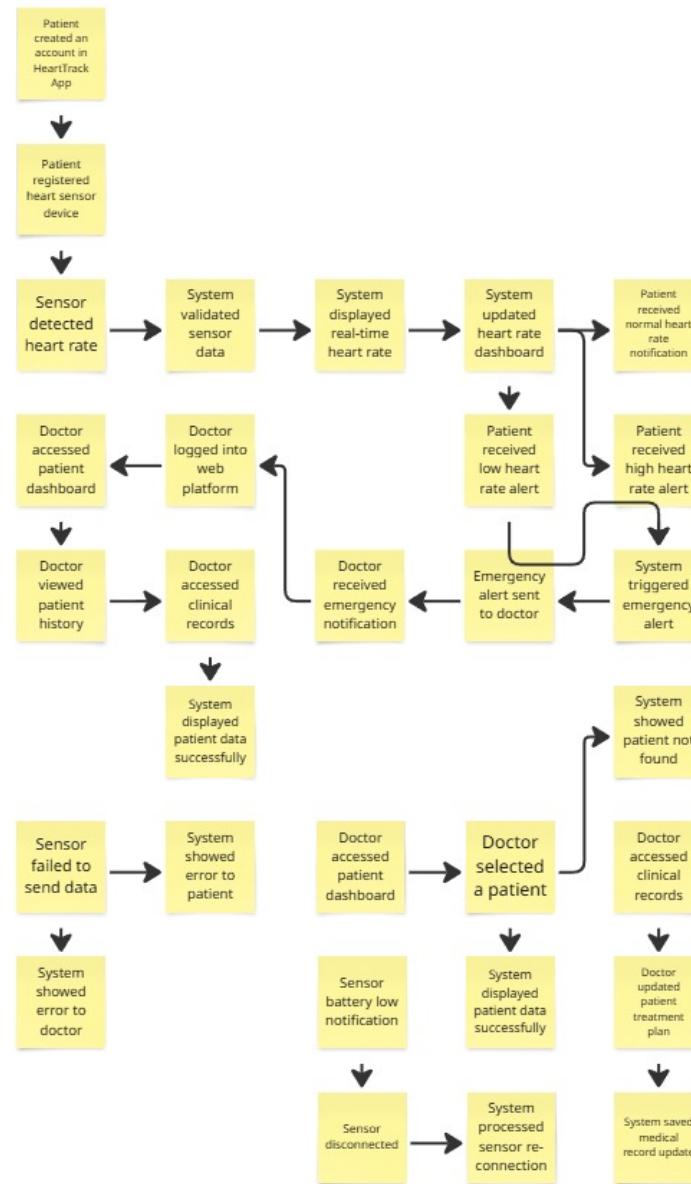
Step 1: Unstructural exploration

Patient created an account in HeartTrack App	Patient registered heart sensor device	Sensor detected heart rate	System validated sensor data	System displayed real-time heart rate	System updated heart rate dashboard	Patient received normal heart rate notification
Patient received high heart rate alert	Patient received low heart rate alert	System triggered emergency alert	Emergency alert sent to doctor	Doctor received emergency notification	Doctor logged into web platform	Doctor accessed patient dashboard
Doctor viewed patient history	Doctor accessed clinical records	System displayed patient data successfully	Sensor failed to send data	System showed error to patient	System showed error to doctor	Doctor confirmed emergency response
Doctor updated patient treatment plan	System displayed notification confirmation	System saved medical record update	Patient exited the app	System logged out the session	Doctor listed all monitored patients	System showed patient not found
Doctor selected a patient	System validated login credentials	Successful login	Admin accessed monitoring system	System recorded abnormal activity	Critical case registered	Patient received medical recommendation
Doctor reviewed treatment recommendation	Admin updated system configuration	Sensor battery low notification	Sensor disconnected	System processed sensor re-connection		

Step 2: Timelines:

En este paso se realizaron las distintas conexiones entre eventos del flujo de la aplicación.

Step 2: Timelines



Step 3: Pain points

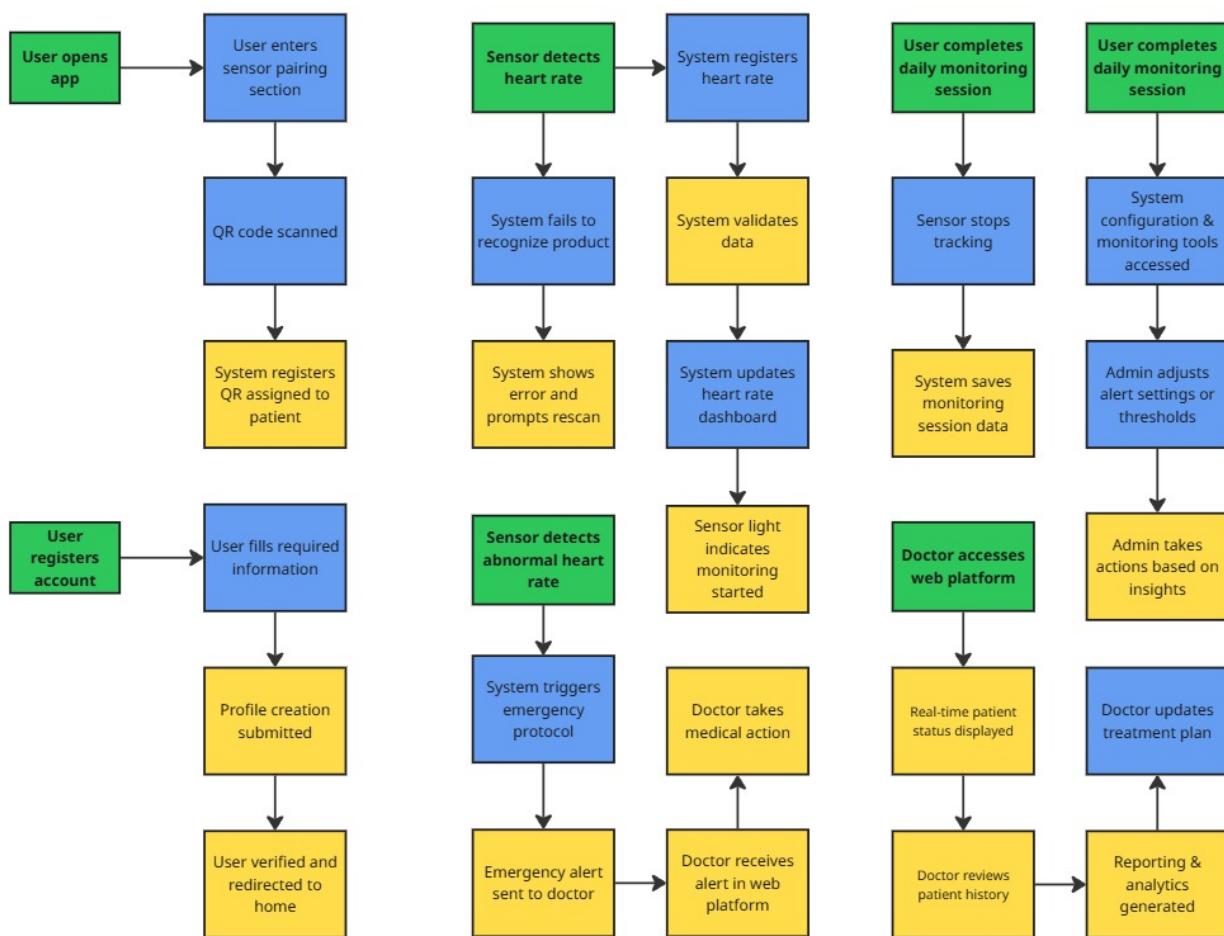
En este paso se colocaron los eventos críticos y sus respectivos pain events identificados.



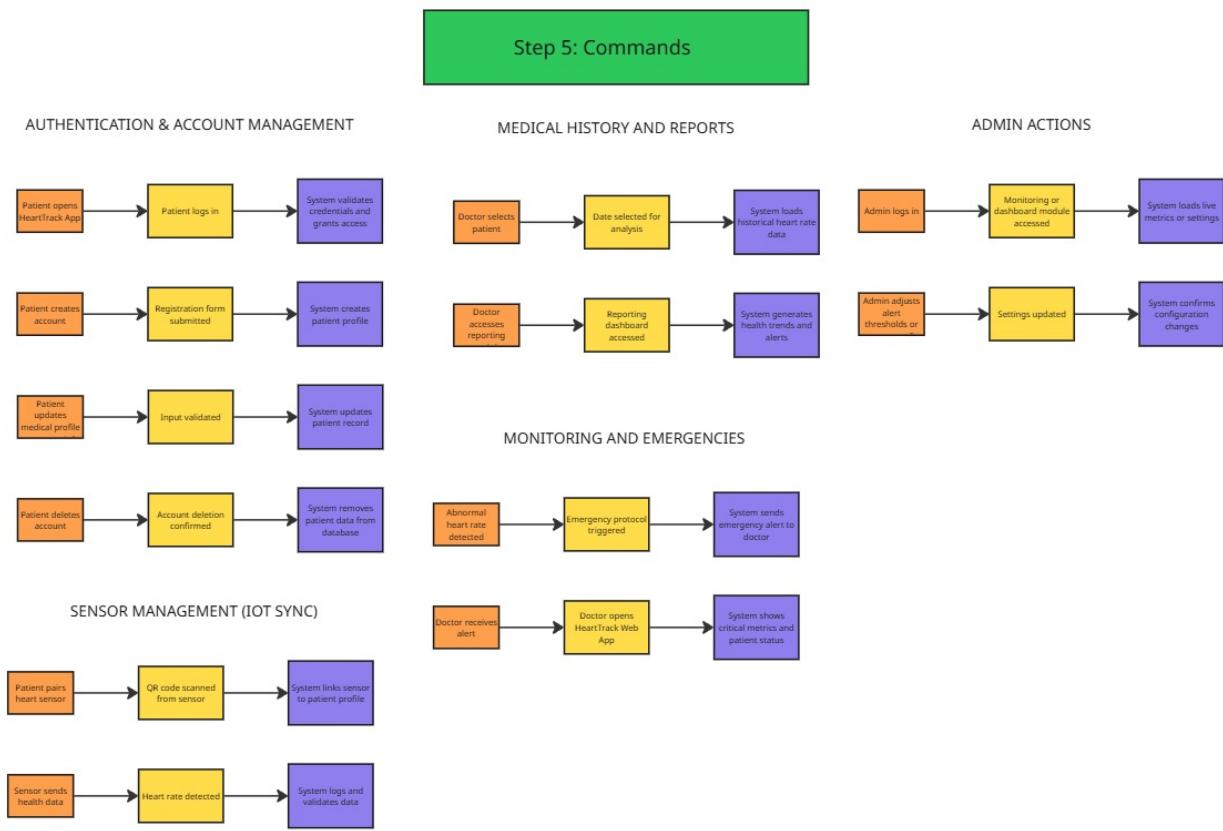
Step 4: Pivotal points

En este paso se incluyeron los pivotal points dentro del flujo de la aplicación.

Step 4: Pivotal points



Step 5: Commands En este paso se identificó los comandos realizados por sus respectivos actores.



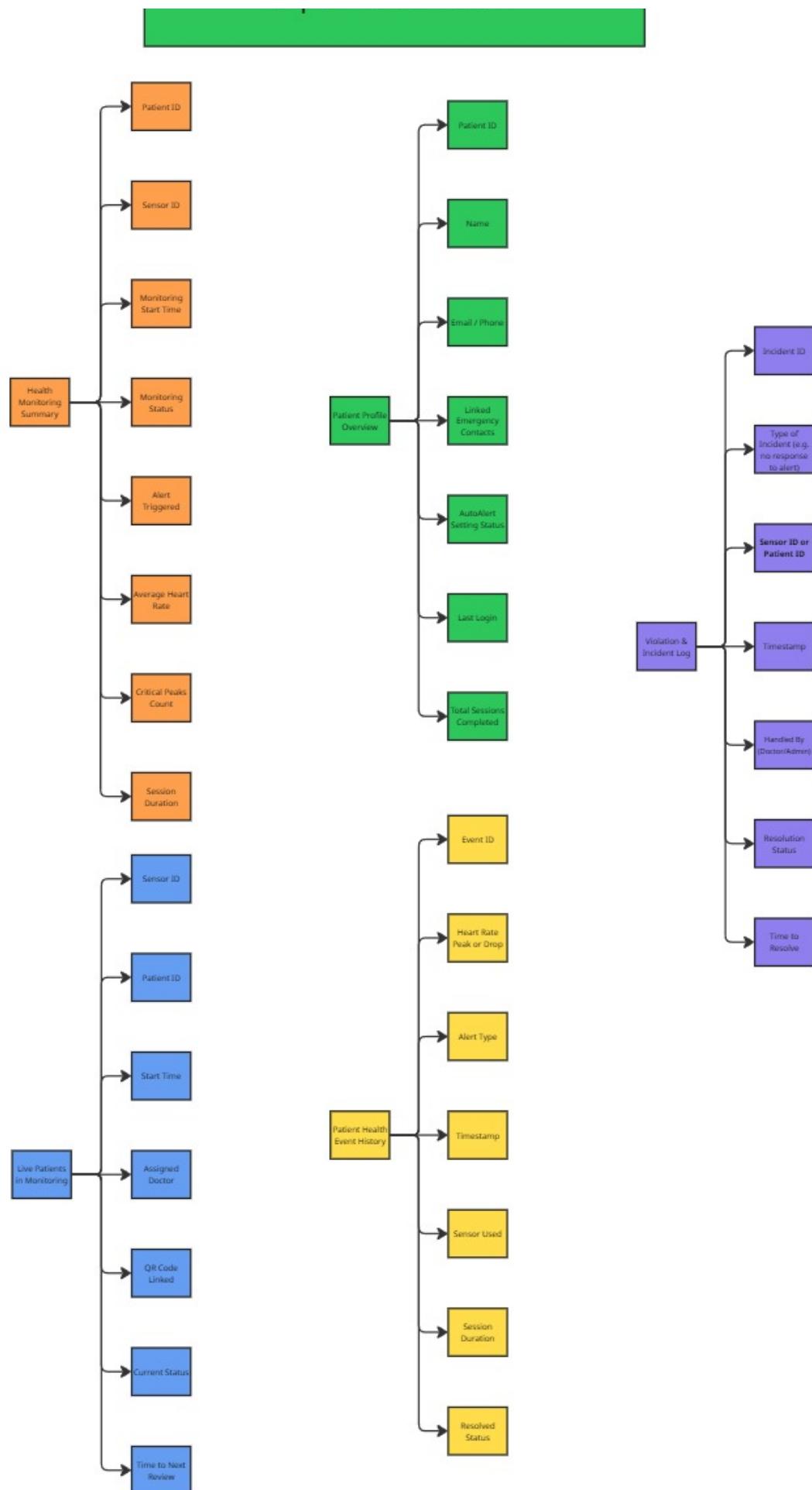
Step 6: Policies En este paso se incluyeron las políticas, escenarios donde la aplicación ejecuta alguna acción.

Step 6: Policies

Patient must be authenticated before accessing account-specific health data.	Deleted sensors must be confirmed as removed before new devices are paired.	Health monitoring begins only after the sensor is successfully paired and validated.	Account deletion must be irreversible and require patient confirmation.	Emergency alerts can only be enabled if at least one verified contact exists.
Once an abnormal rate is detected, the system should lock alert deactivation unless overridden by a doctor.	All sensitive data updates (emergency contacts, personal info) must be validated by the user.	Failed data transmissions should prompt guidance and retry options.	Multiple abnormal heart rate events should flag the user for active monitoring.	Sensor pairing failures should trigger a fallback QR/manual input process.
Doctors can only access patient data if medical consent is granted and registered.	Only authorized doctors and admins can access the monitoring dashboard.	Duplicate sensor readings within a short timeframe must be filtered.	Abnormal data with no emergency action must trigger a manual review alert.	All sensor readings must be validated before visualization or alert generation.
Real-time alerts must notify both doctor and patient with context.	Admins can mark patients as 'high priority' for emergency response scenarios.	All health data transmissions and alerts must be logged and reviewable.	Receipts of critical events must be stored for 30+ days post-event.	Past health data should only be accessible to verified users.
If no health history exists, prompt the user with setup guidance or support.				

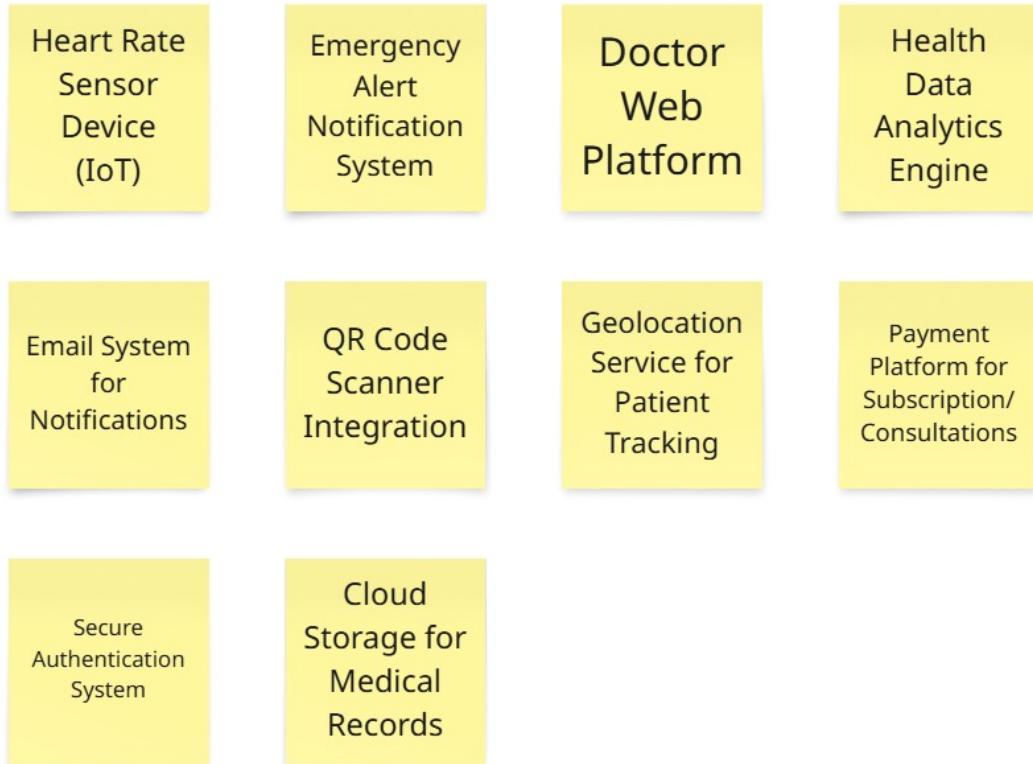
Step 7: Read Models En este paso se realizó el modelo de lectura de cada sistema de la aplicación.

Step 7: Read models

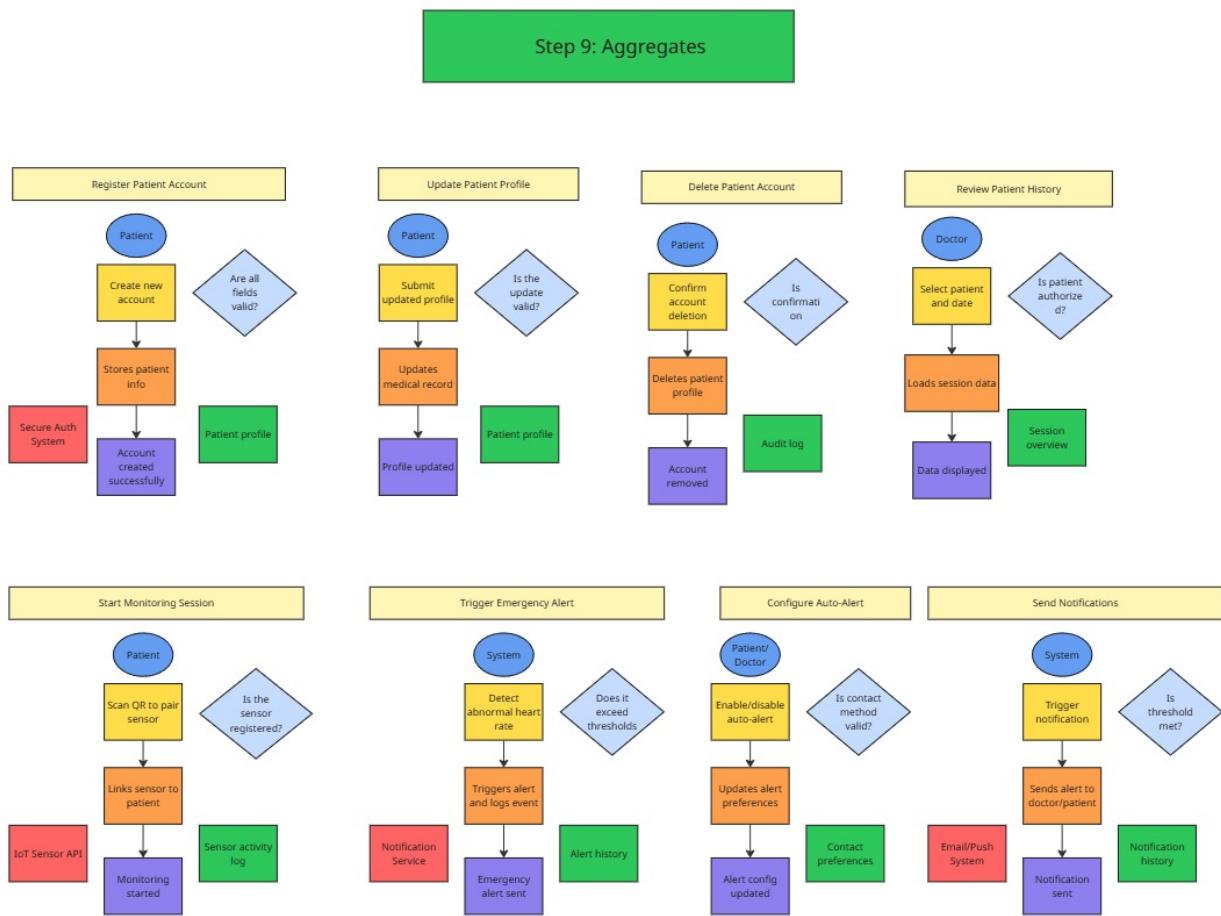


Step 8: External systems En este paso se identificaron algunos sistemas externos.

Step 8: External Systems



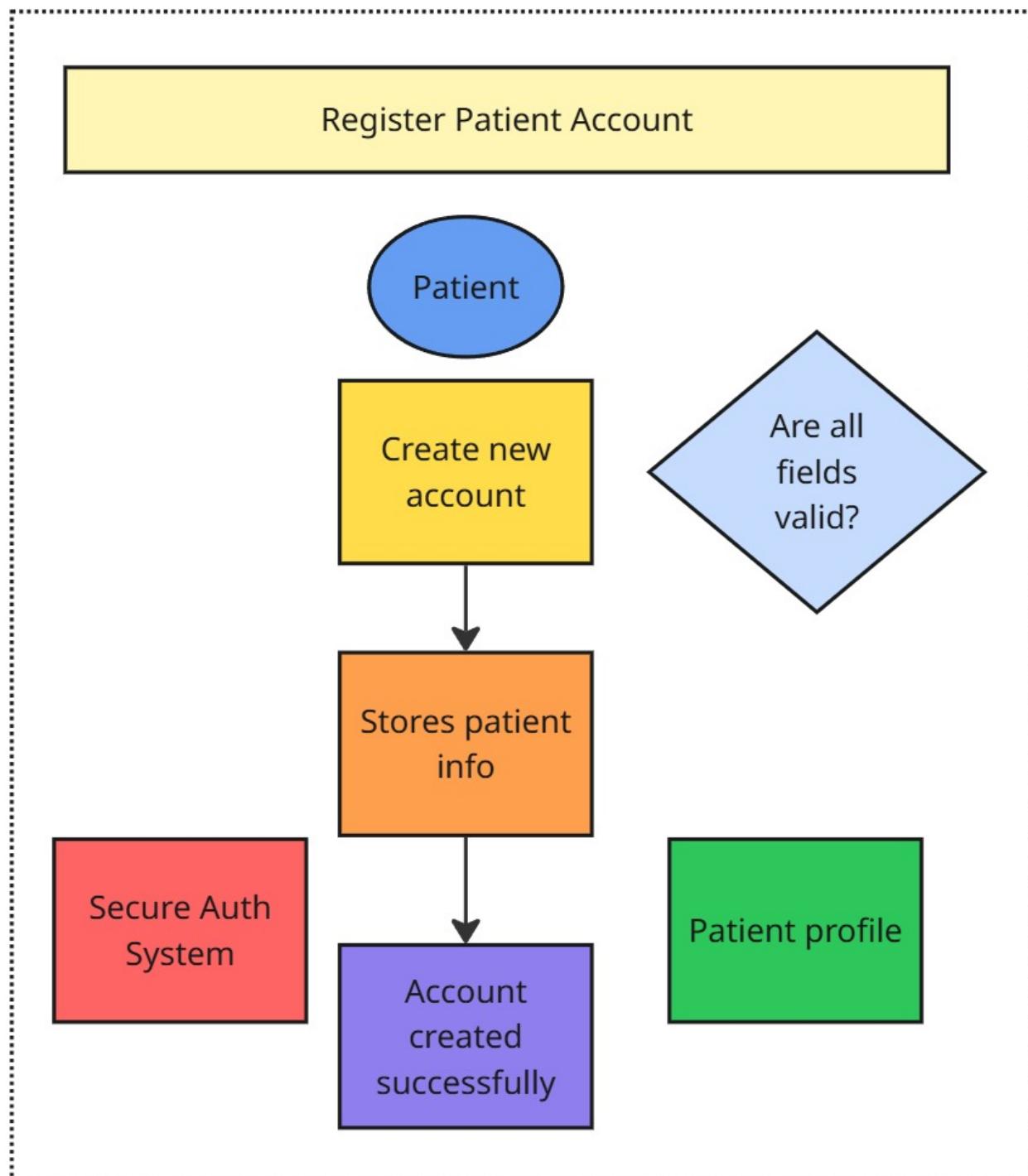
Step 9: Aggregates En este paso se implementan los agregados de cada sistema de la aplicación.



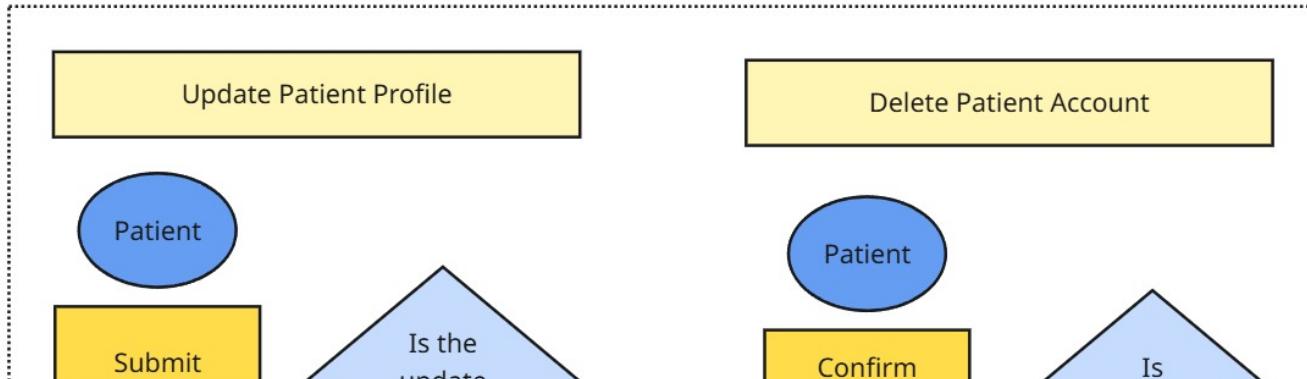
4.1.1.1 Candidate Context Discovery.

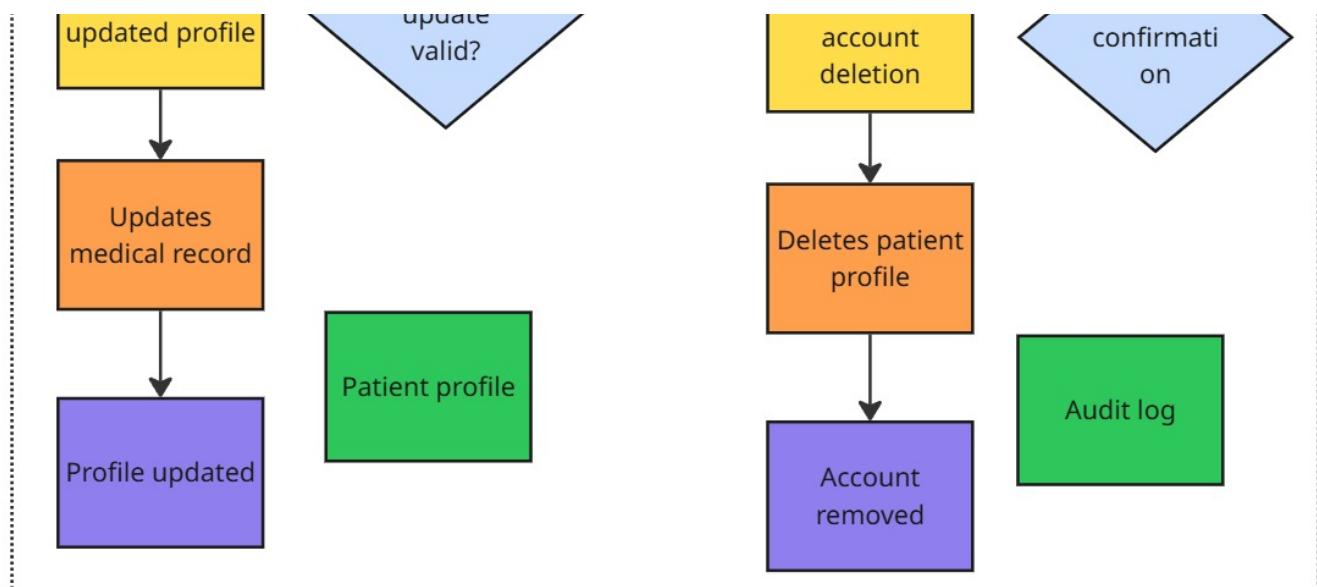
En este punto luego de la realización del Event storming, se identificaron los bounded context a trabajar. En este caso se lograron identificar 6.

IAM

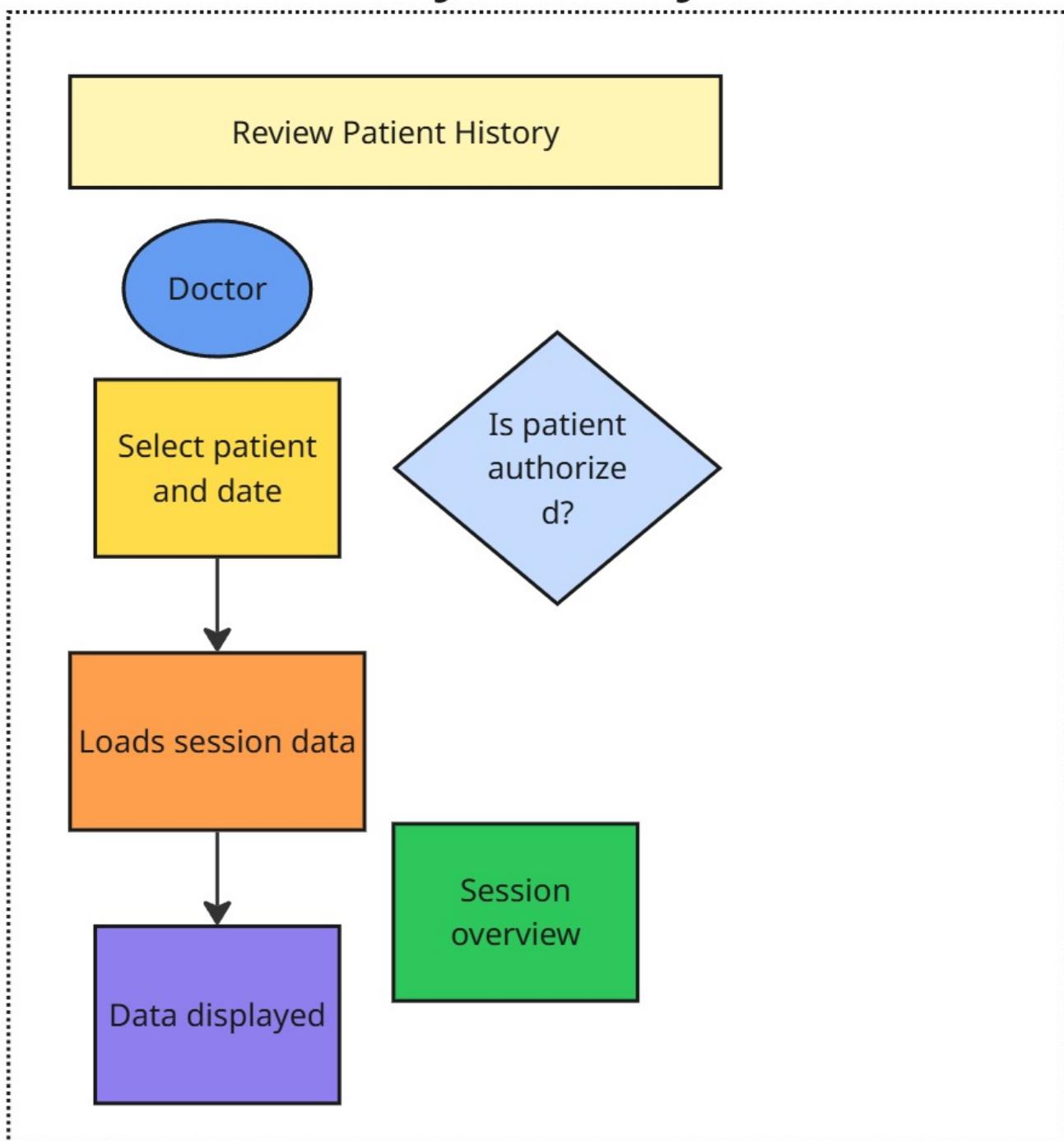


Patient management

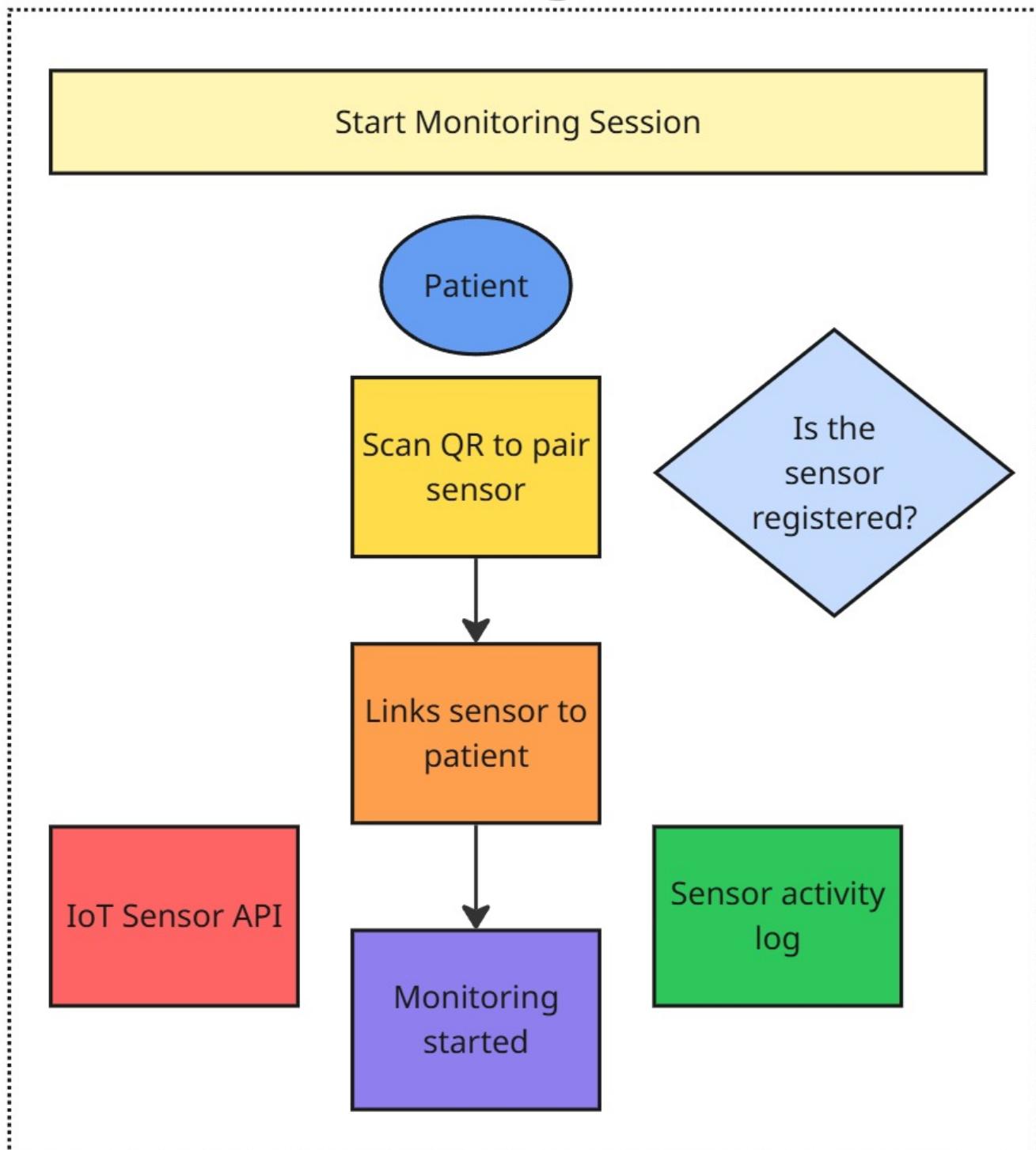




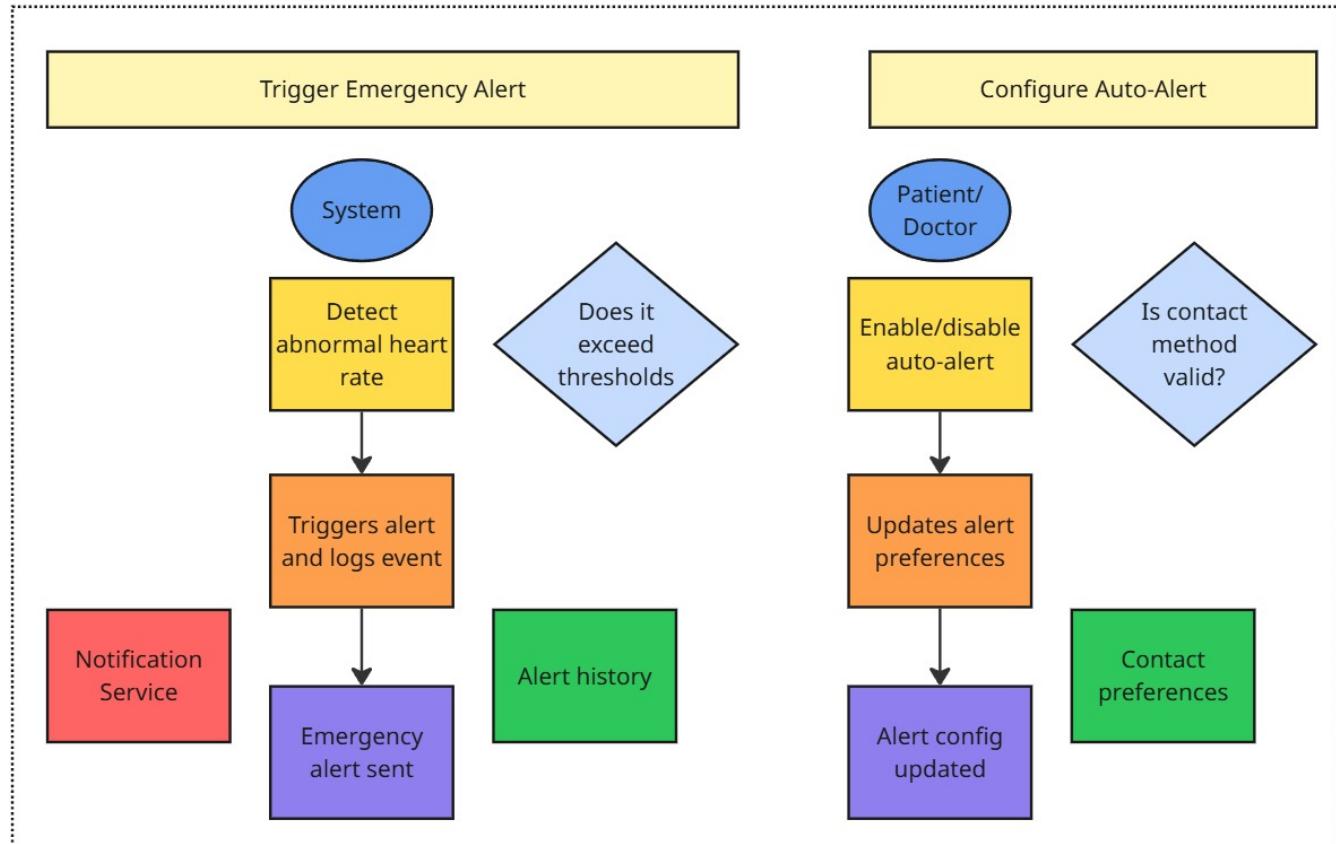
Medical history & analytics



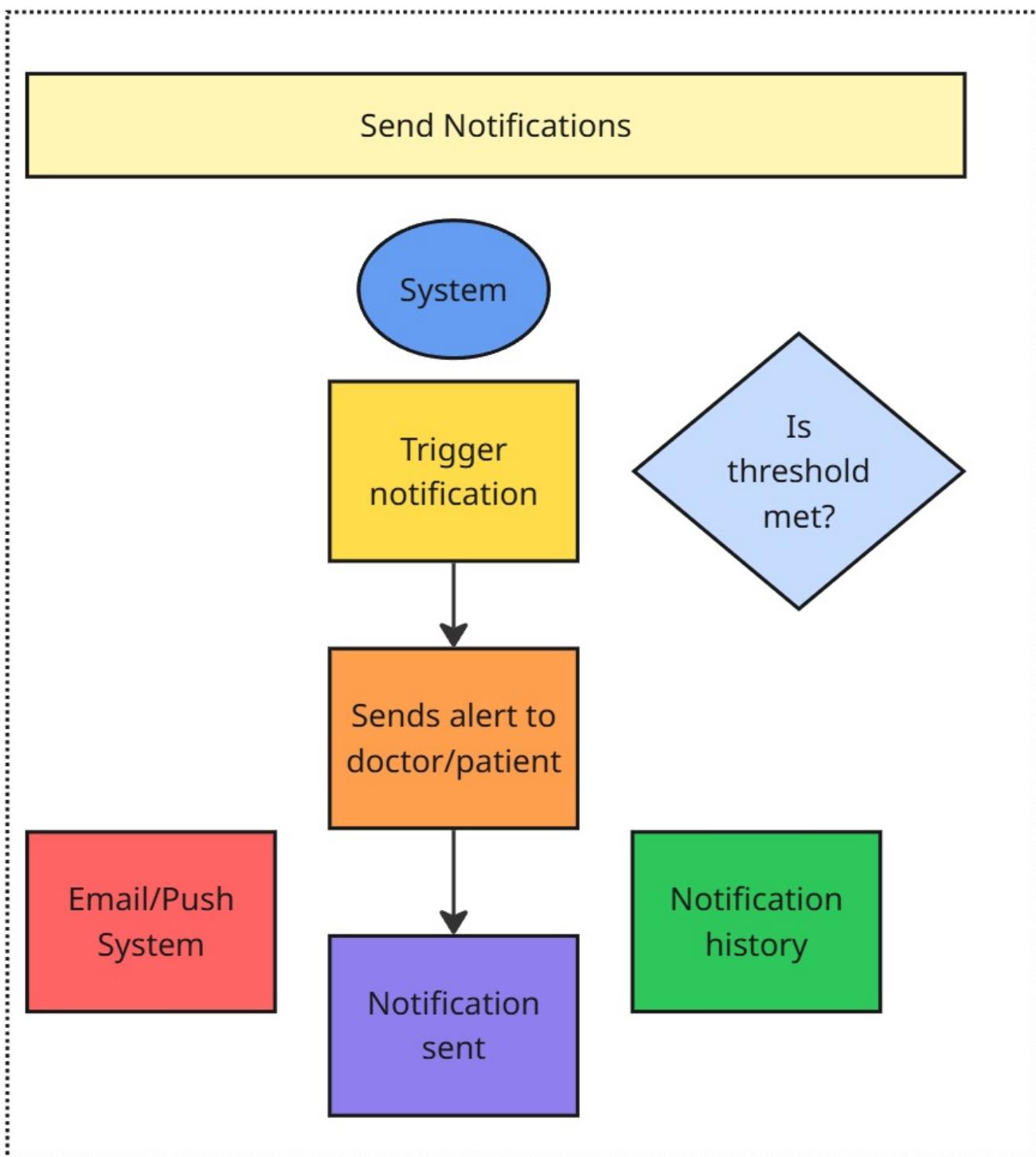
Sensor monitoring



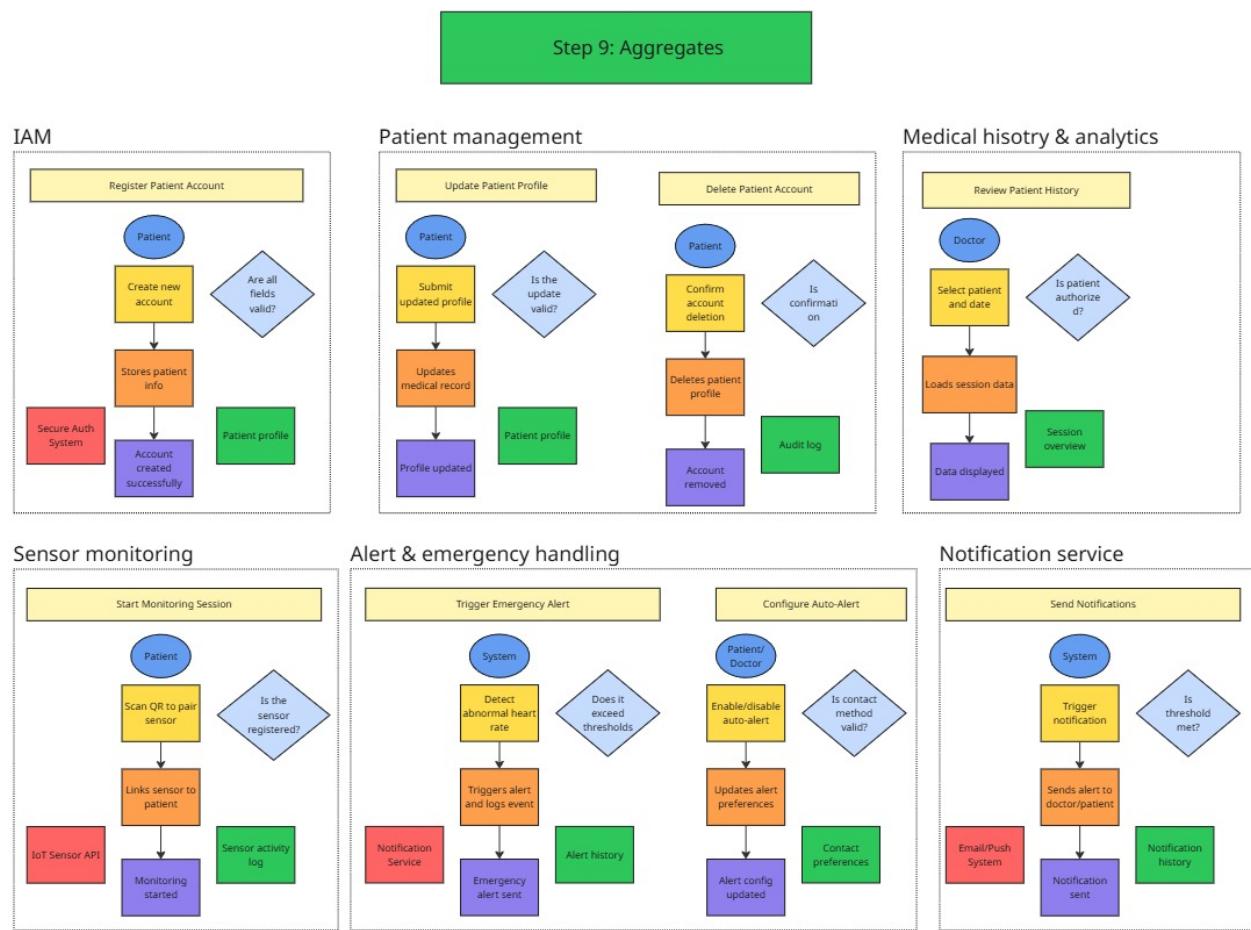
Alert & emergency handling



Notification service

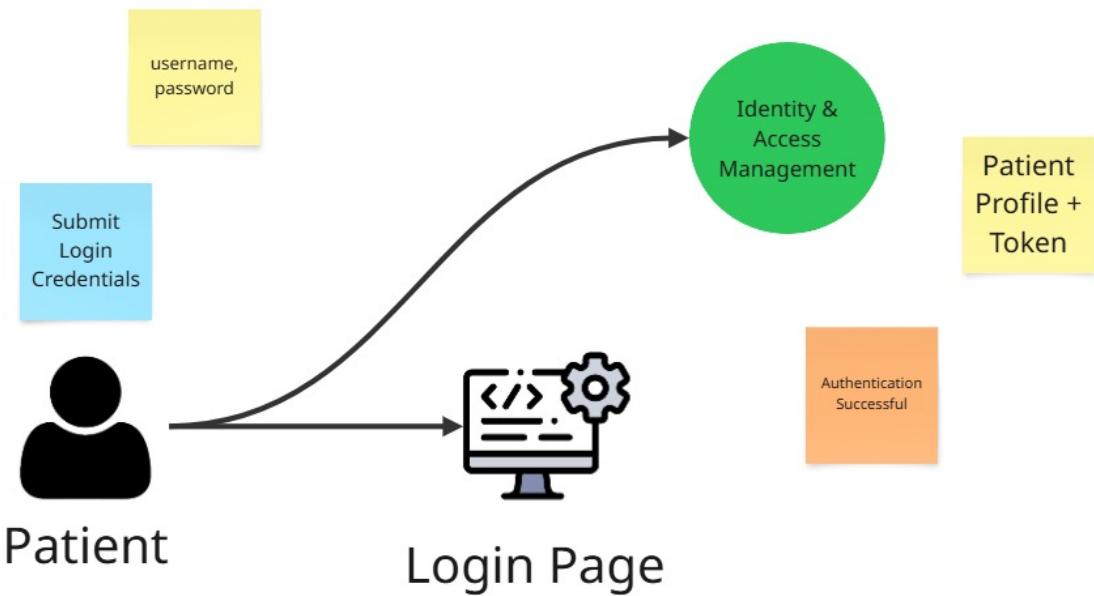


Aquí podemos apreciarlo completo:

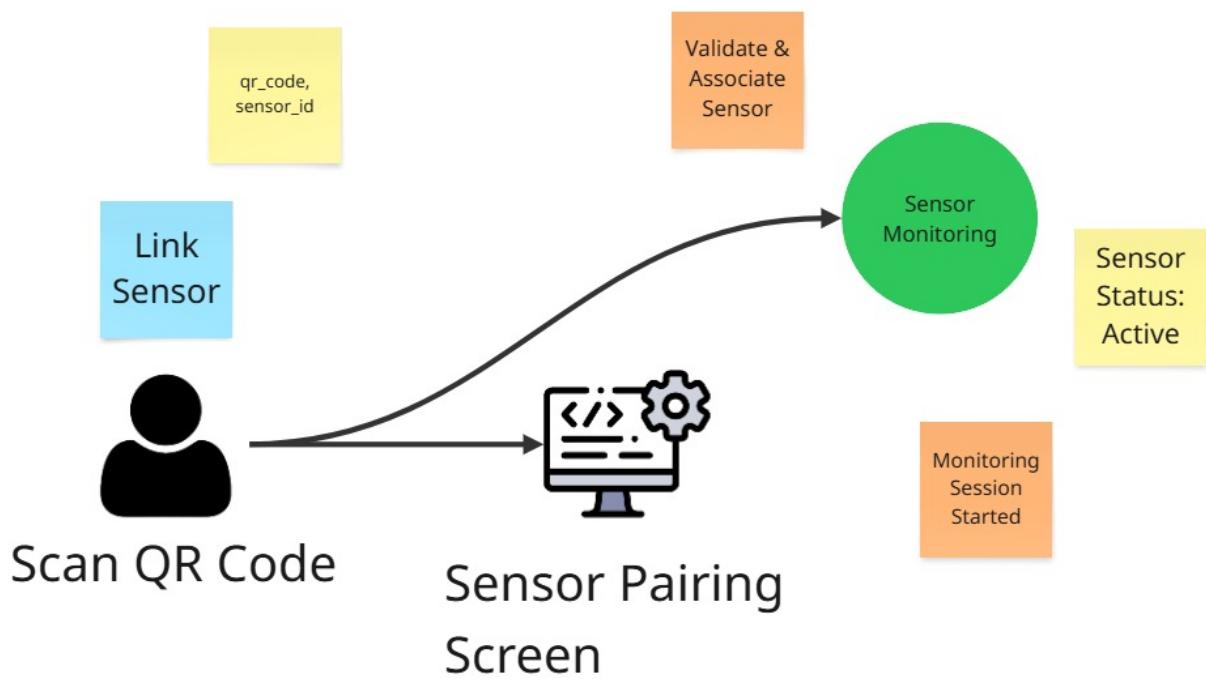


4.1.1.2 Domain Message Flows Modeling.

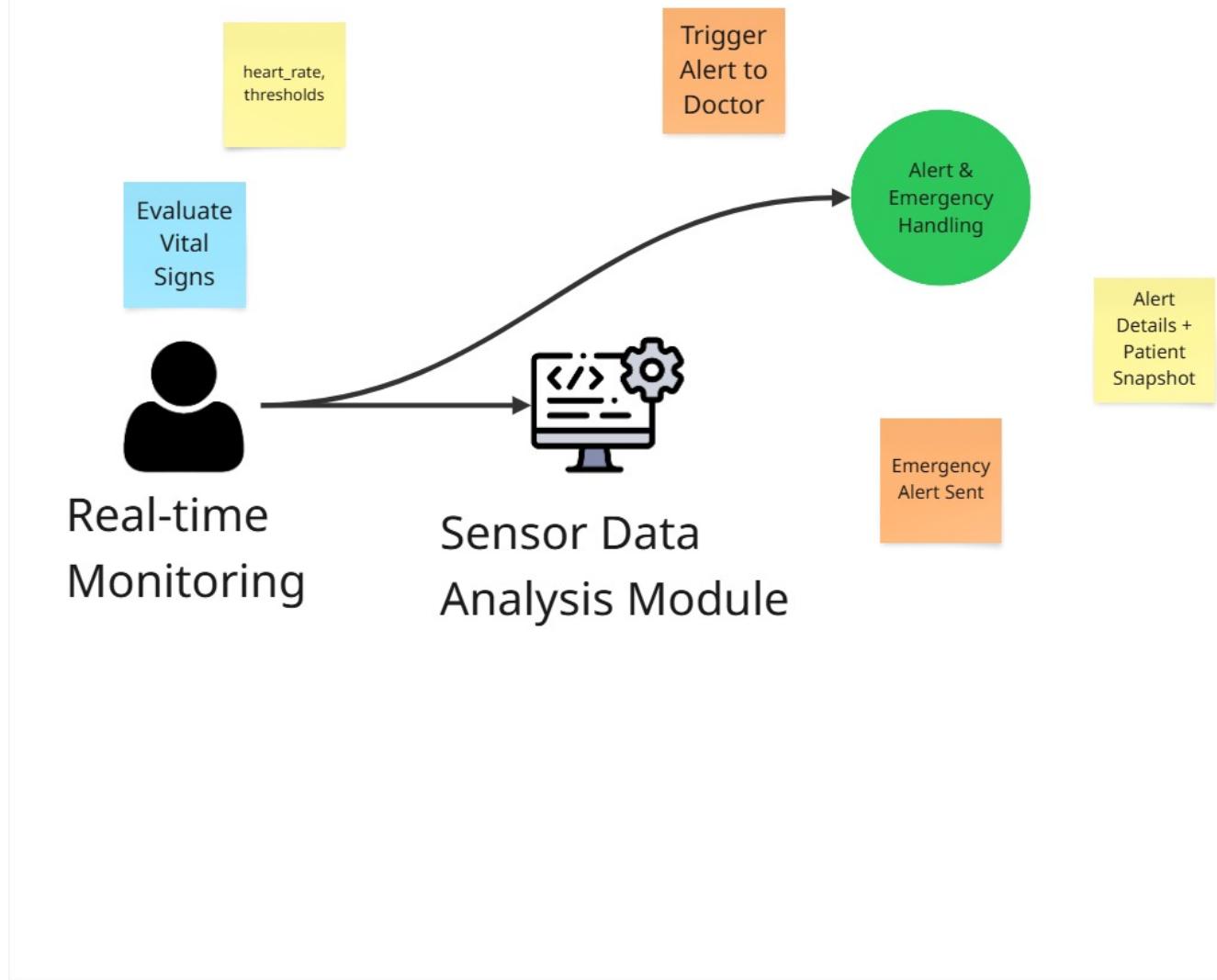
Paciente Escenario: El paciente desea acceder a la plataforma



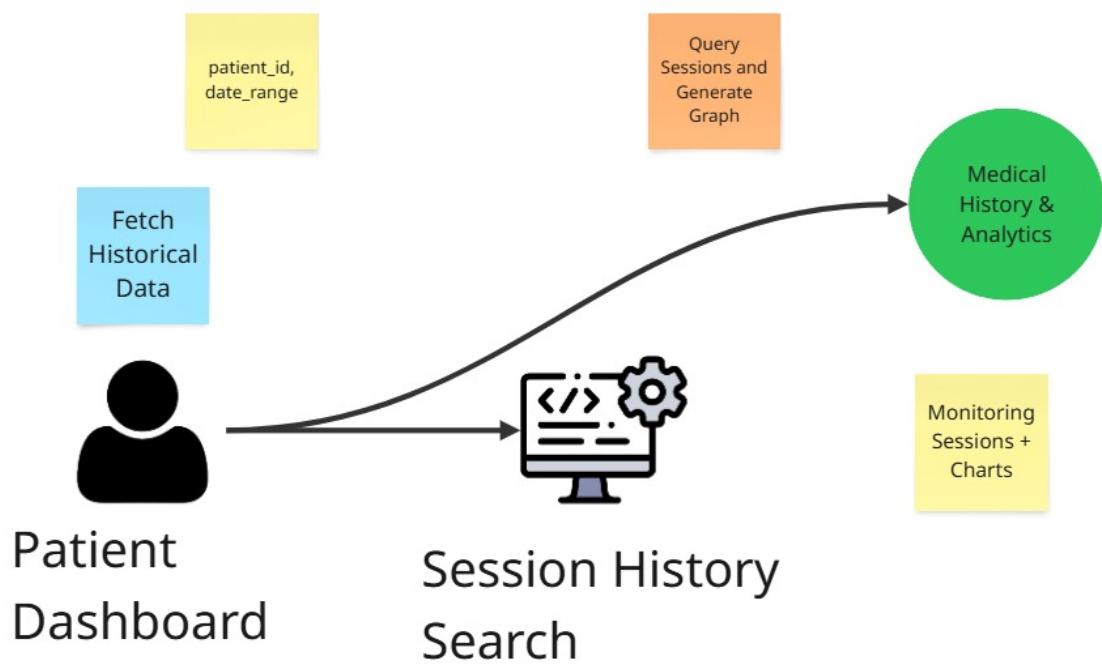
Scan Escenario: Paciente vincula su sensor y comienza la sesión Flujo:



Monitoreo Escenario: Ritmo cardíaco crítico detectado automáticamente

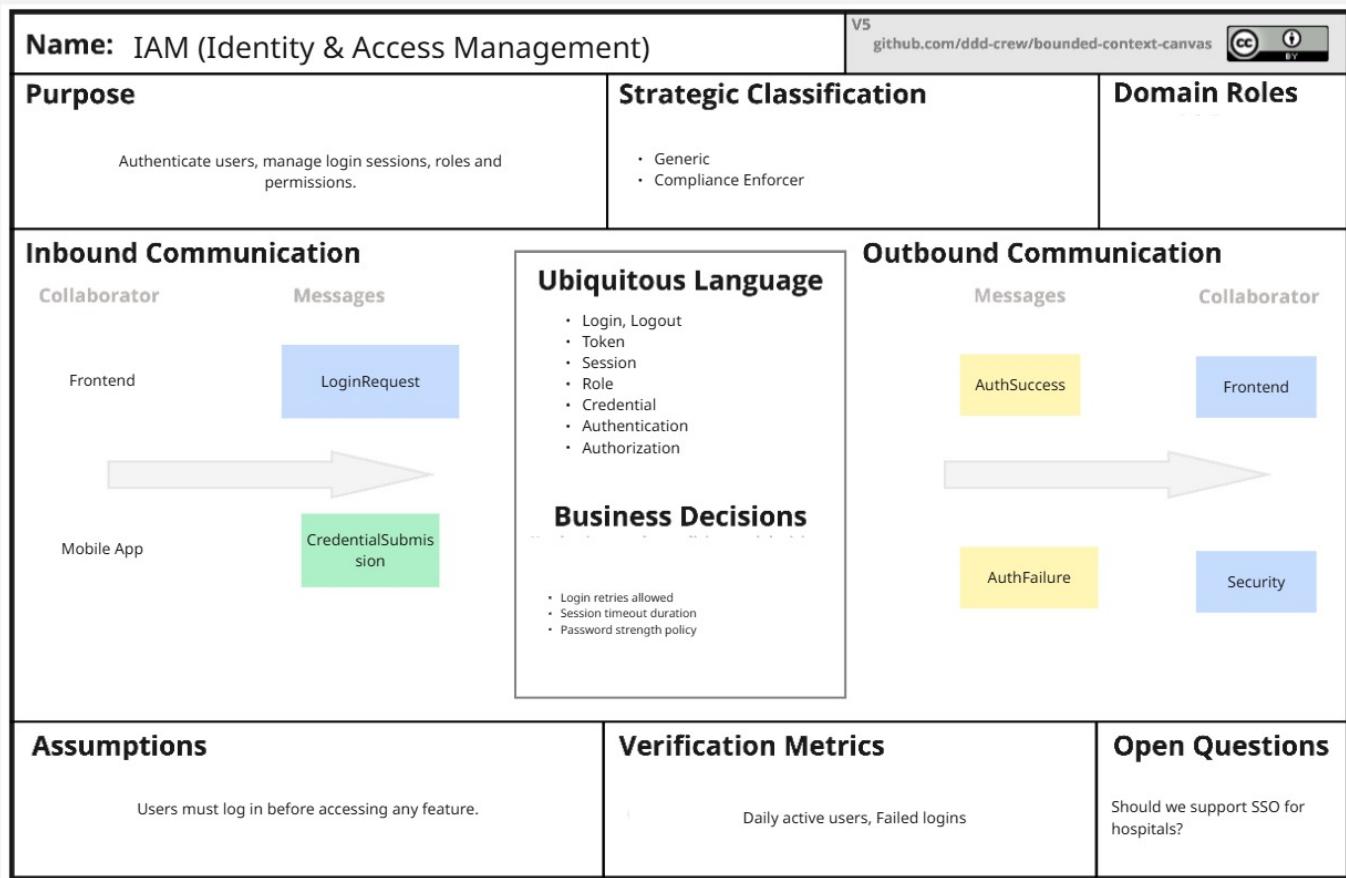


Doctor Escenario: Doctor revisa sesiones pasadas del paciente

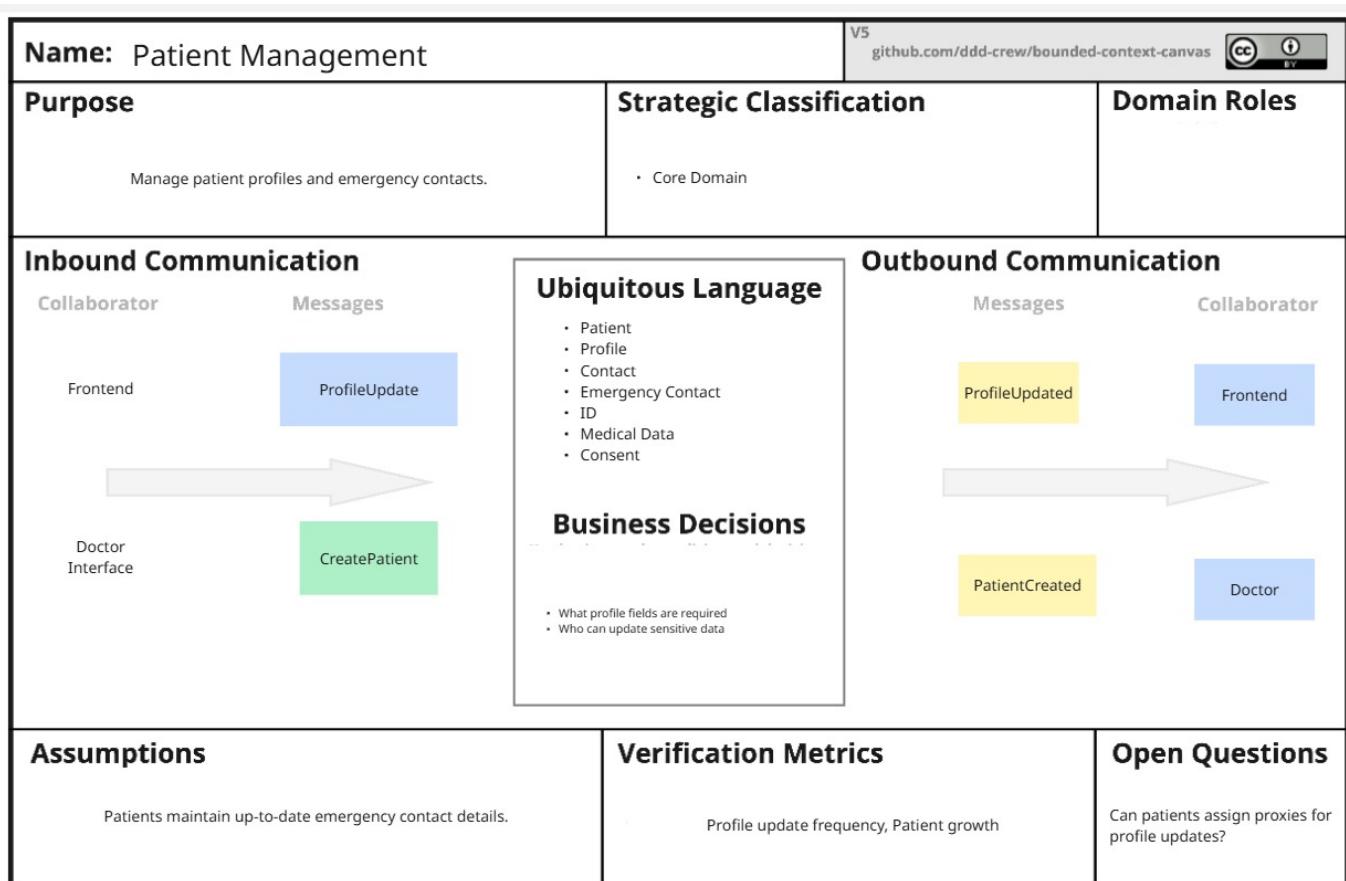


4.1.1.3 Bounded Context Canvases.

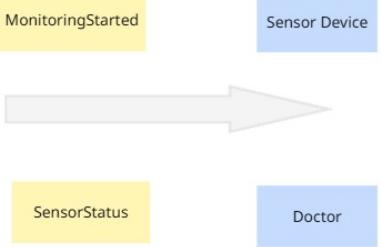
- IAM



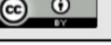
- Patient Management



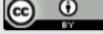
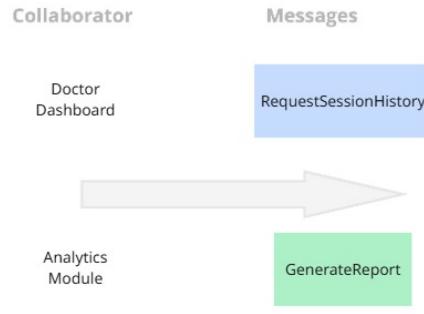
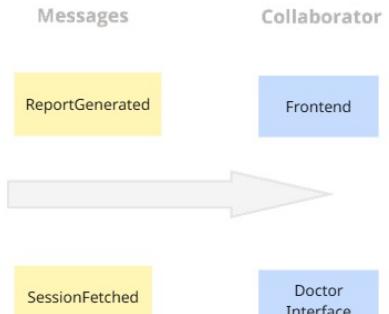
- Sensor Monitoring

Name: Sensor Monitoring		V5 github.com/ddd-crew/bounded-context-canvas 
Purpose Connect and validate heart rate sensors, collect vital data in real-time.	Strategic Classification • Core Domain	Domain Roles
Inbound Communication Collaborator Messages Frontend PairSensor  Doctor Interface StartMonitoring	Ubiquitous Language • Sensor • Heart Rate • Monitoring • Session • QR Code • Link • Disconnect Business Decisions • How frequently sensors send data • What constitutes a valid reading	Outbound Communication Messages Collaborator MonitoringStarted Sensor Device  SensorStatus Doctor
Assumptions Sensors are properly paired and functional.	Verification Metrics Sensor uptime, Sessions started	Open Questions How to handle sensor disconnection mid-session?

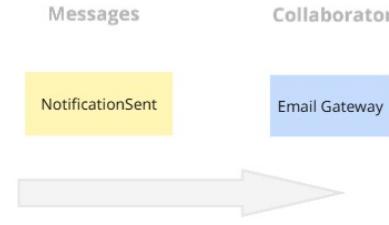
- Alert & Emergency Handling

Name: Alert & Emergency Handling		V5 github.com/ddd-crew/bounded-context-canvas 
Purpose Evaluate critical vitals, trigger medical alerts and coordinate doctor response.	Strategic Classification • Core Domain	Domain Roles
Inbound Communication Collaborator Messages Sensor Monitoring AbnormalVitalsDetected  Doctor	Ubiquitous Language • Alert, Threshold • Emergency • Notification • Trigger • Response Time • Escalation Business Decisions • Alert threshold levels • Automatic vs manual alert handling • Escalation rules	Outbound Communication Messages Collaborator AlertTriggered Doctor App  DoctorNotified Notification System
Assumptions Thresholds for alerts are clinically defined.	Verification Metrics Number of alerts triggered, Doctor response time	Open Questions What's the escalation protocol for unanswered alerts?

- Medical History & Analytics

Name: Medical History & Analytics		V5 github.com/ddd-crew/bounded-context-canvas 
Purpose Store and retrieve historical session data and generate reports.	Strategic Classification • Supportive • Reporting	Domain Roles
Inbound Communication  <p>The diagram shows a 'Doctor Dashboard' sending a 'RequestSessionHistory' message to an 'Analytics Module'. The 'Analytics Module' responds with a 'GenerateReport' message back to the 'Doctor Dashboard'.</p>	Ubiquitous Language • Session History • Report • Chart • Graph • Analytics • Time Range • Query Business Decisions • How long to store session history • Report access level per user role	Outbound Communication  <p>The diagram shows an 'Analytics Module' sending a 'ReportGenerated' message to a 'Frontend'. The 'Frontend' then sends a 'SessionFetched' message to a 'Doctor Interface'.</p>
Assumptions Doctors have permission to view assigned patients.	Verification Metrics Reports generated per month, Query time	Open Questions Should sessions be exportable as PDFs?

- Notification Service

Name: Notification Service		V5 github.com/ddd-crew/bounded-context-canvas 
Purpose Send notifications (alerts, confirmations, reminders) through various channels.	Strategic Classification • Supportive • Infrastructure	Domain Roles
Inbound Communication  <p>The diagram shows a 'System Events' component sending a 'TriggerNotification' message to an 'Emergency Handler'.</p>	Ubiquitous Language • Notification • Channel • Email • Push • SMS • Delivery • Retry • Alert Type Business Decisions • Default notification methods • Retry attempts • Priority of alerts	Outbound Communication  <p>The diagram shows an 'Emergency Handler' sending a 'NotificationSent' message to an 'Email Gateway'. The 'Email Gateway' then sends a message to a 'Push Service'.</p>
Assumptions Notification channels are preconfigured.	Verification Metrics Delivery success rate, Alert timeliness	Open Questions How to handle failed notifications or retries?

4.1.2. Context Mapping.

Dentro del Event storming, se identificaron 6 bounded context:

- IAM
- Patient management
- Medical history & analytics
- Sensor monitoring
- Alert & emergency handling
- Notification service

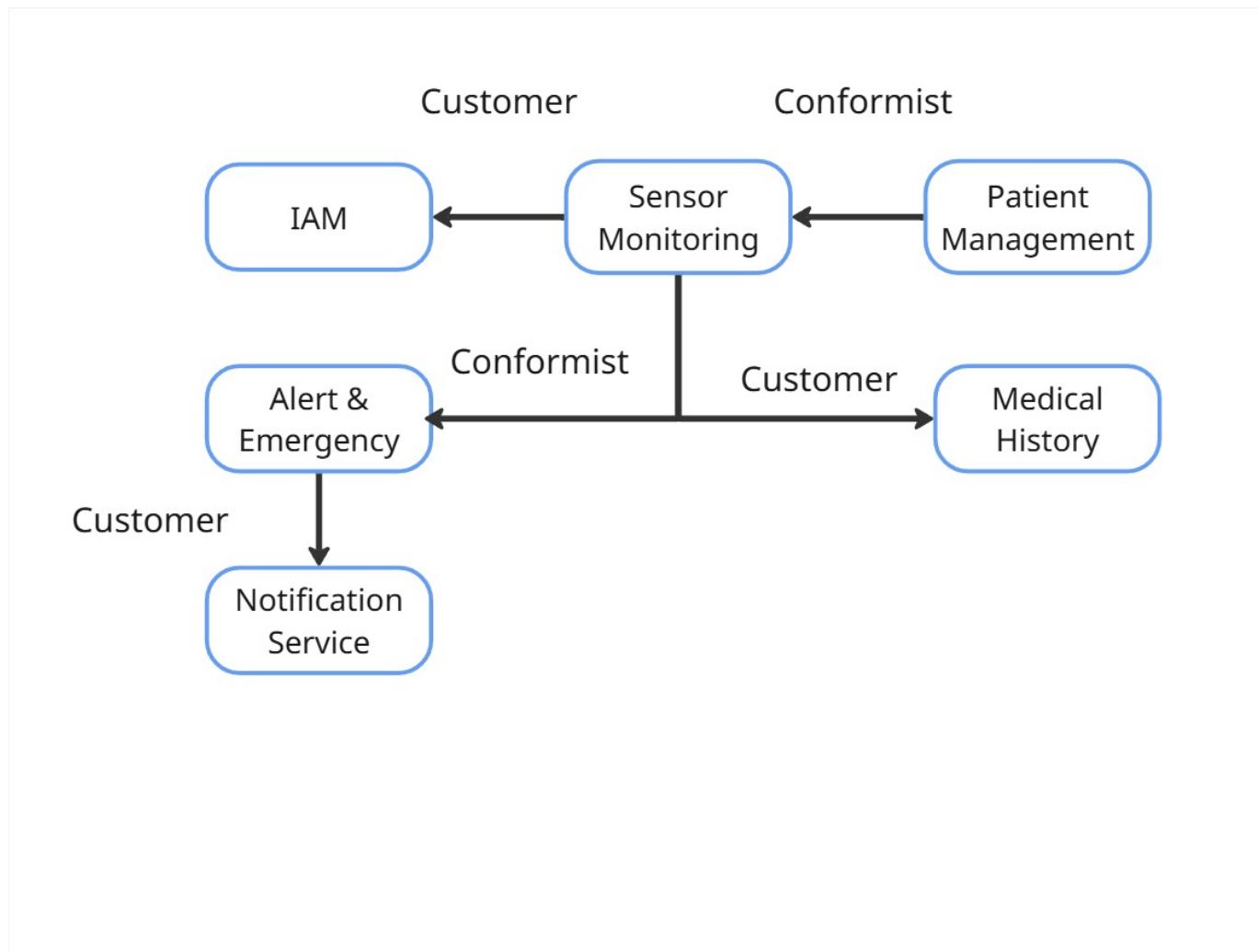
Se ha decidido hacer uso los patrones de Conformist y de Customer/Supplier:

Conformist:

- Sensor Monitoring es conformista de Patient Management: adapta los modelos de paciente y contacto.
- Alert & Emergency Handling es conformista de Sensor Monitoring: entiende el formato de eventos críticos sin imponer cambios.

Customer/Supplier

- Medical History & Analytics es cliente de Sensor Monitoring (supplier).
- Notification Service es cliente de Alert & Emergency Handling (supplier).
- Sensor Monitoring es cliente de IAM (para validar tokens y sesiones).



4.1.3. Software Architecture.

4.1.3.1. Software Architecture System Landscape Diagram.

La plataforma MedSystem conecta a pacientes y doctores mediante una aplicación móvil y web, que a su vez se integra con dispositivos IoT (sensores de frecuencia cardíaca) y servicios de notificaciones externas.

Actores involucrados:

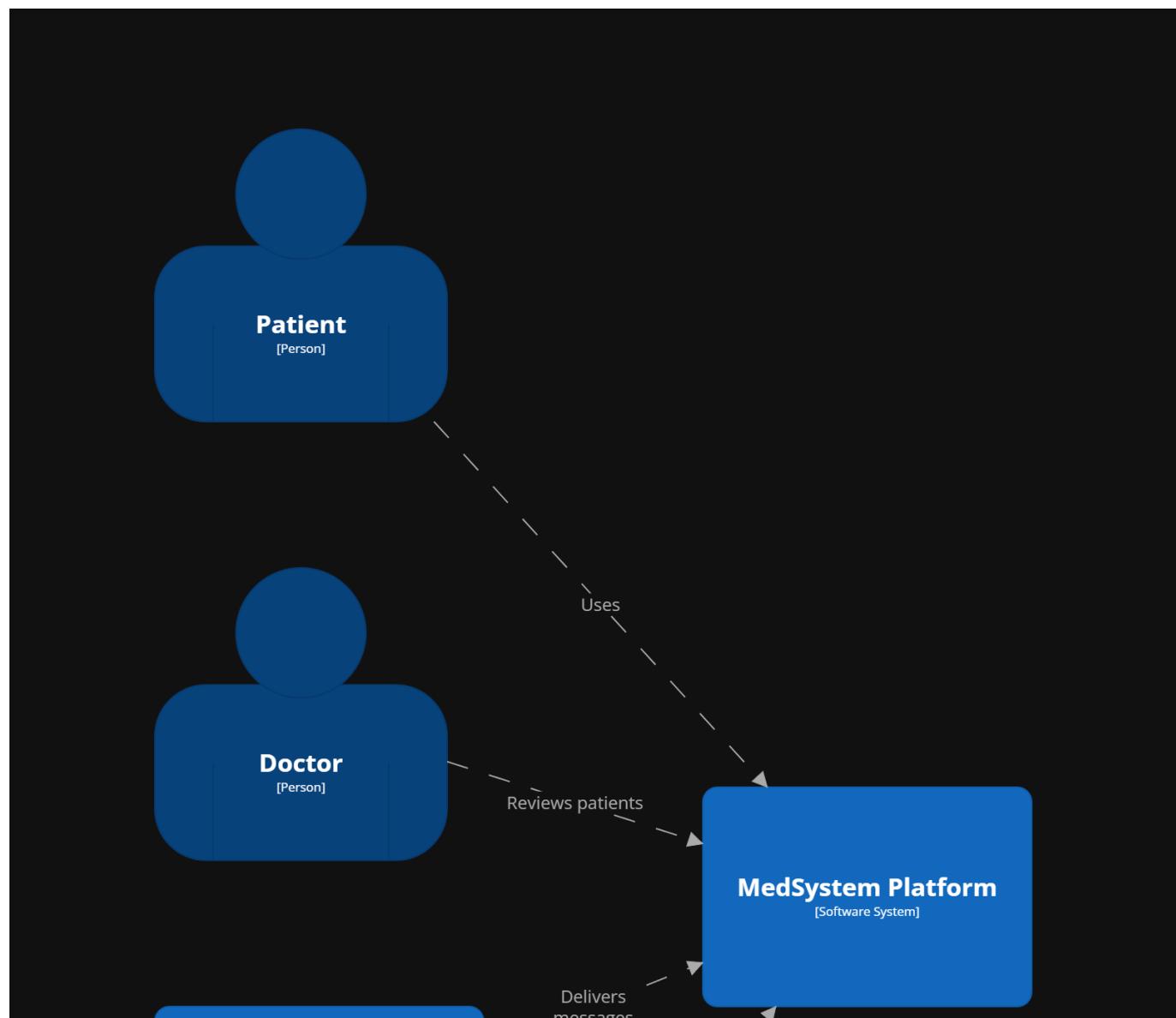
- Paciente: Usa la aplicación para registrar su actividad cardíaca y recibir notificaciones.
- Doctor: Accede al sistema para revisar los datos de los pacientes y recibir alertas de emergencia.

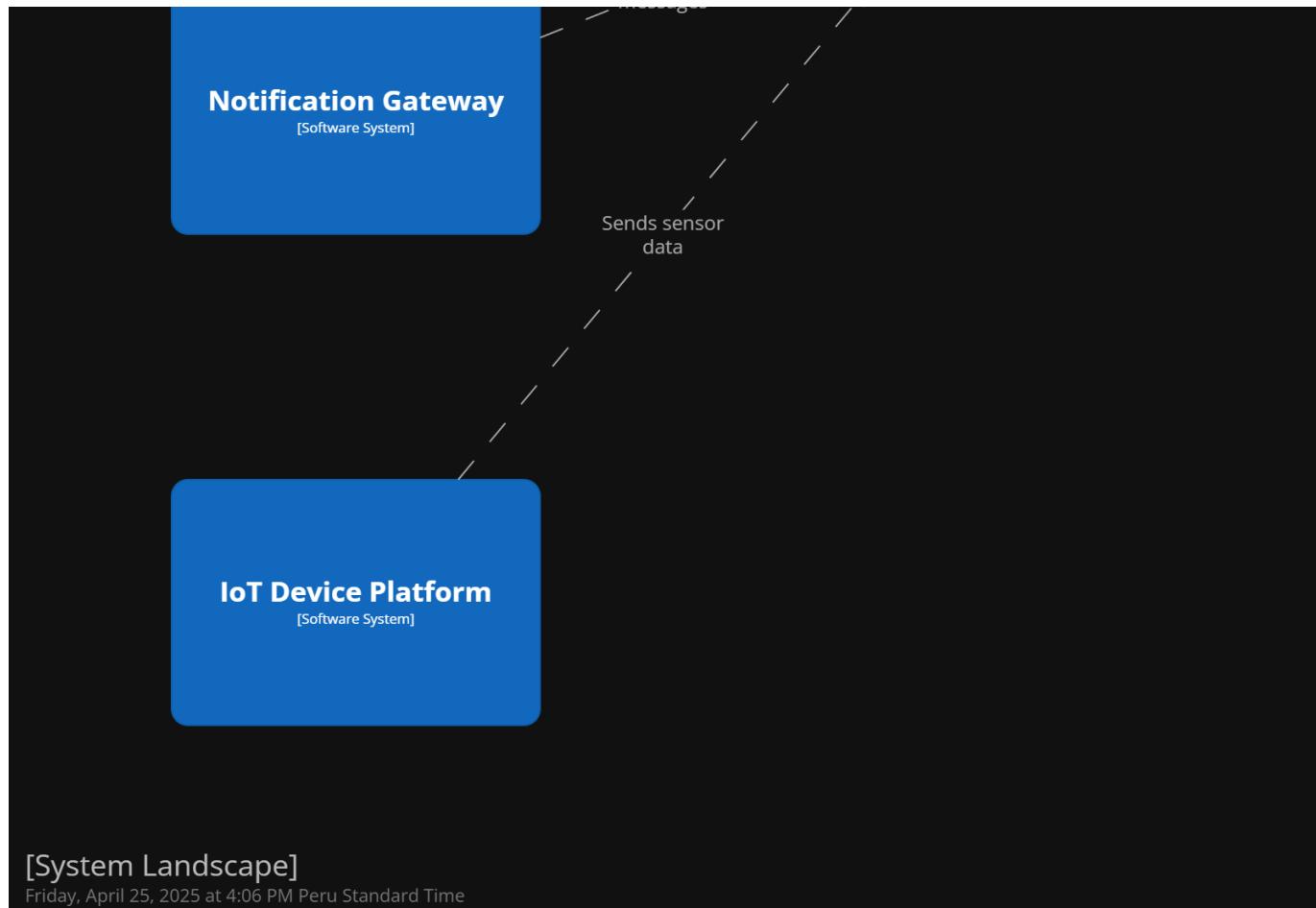
Sistemas externos relevantes:

- Notification Gateway (para alertas de emergencia vía Email/SMS).
- IoT Device Platform (para enviar datos del sensor).

Relaciones clave:

- El Paciente interactúa directamente con la app (MedSystem Platform).
- El Doctor consulta el estado y la historia clínica de sus pacientes mediante la plataforma.
- El sistema MedSystem se comunica con servicios externos para notificaciones y lectura de sensores.





4.1.3.2. Software Architecture Context Level Diagrams.

Sistema central: MedSystem Platform

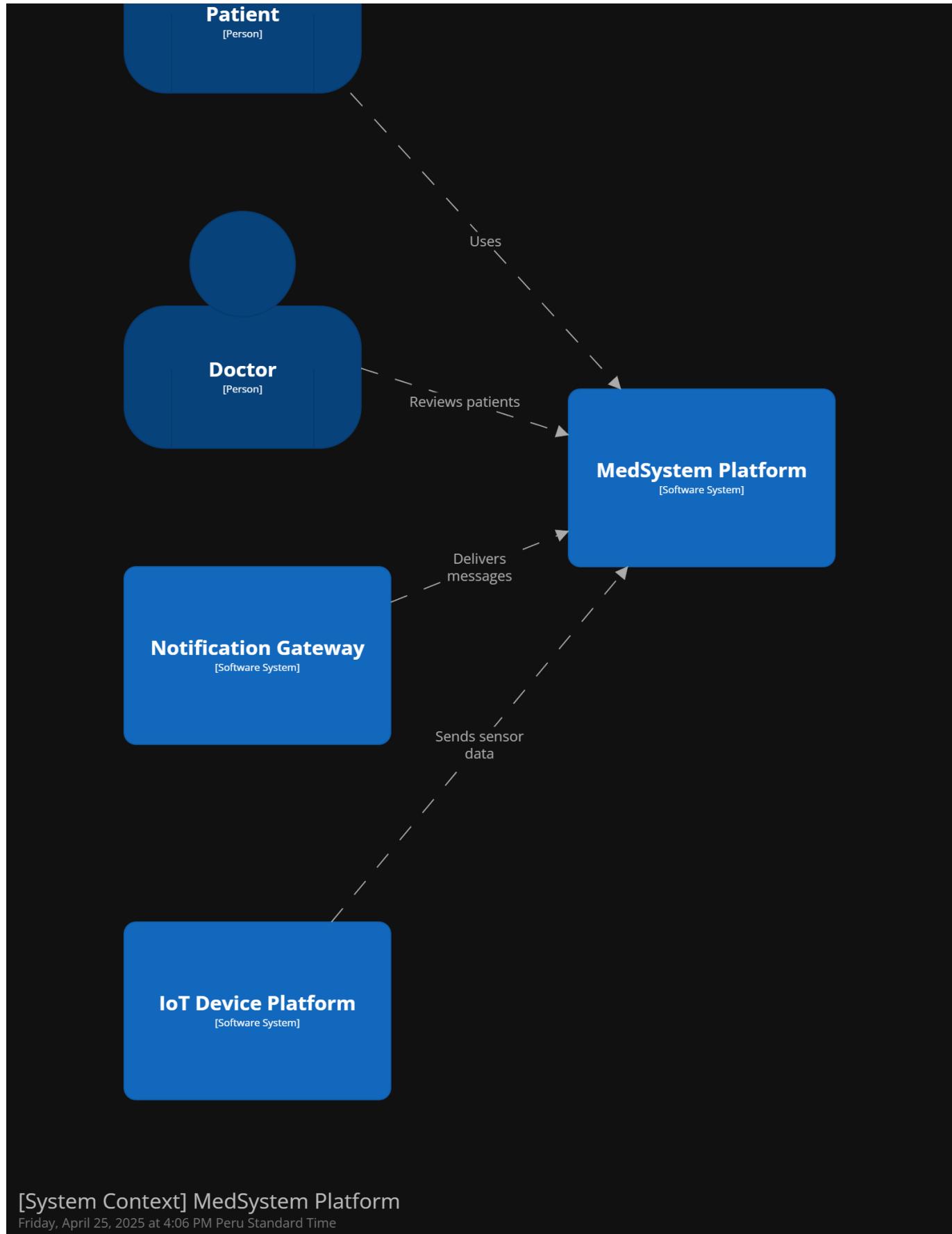
Interacciones principales:

- El Paciente crea su cuenta, vincula su sensor, monitorea su estado de salud y recibe通知 de alerta.
- El Doctor revisa los registros históricos, monitorea alertas en tiempo real y responde a emergencias de sus pacientes.
- Sensores IoT envían la frecuencia cardíaca hacia la plataforma.
- Notification Gateway se utiliza para enviar alertas externas en caso de anomalías cardíacas.

Visión general:

MedSystem centraliza toda la interacción entre usuarios humanos (pacientes y doctores) y sistemas externos (IoT, notificaciones).





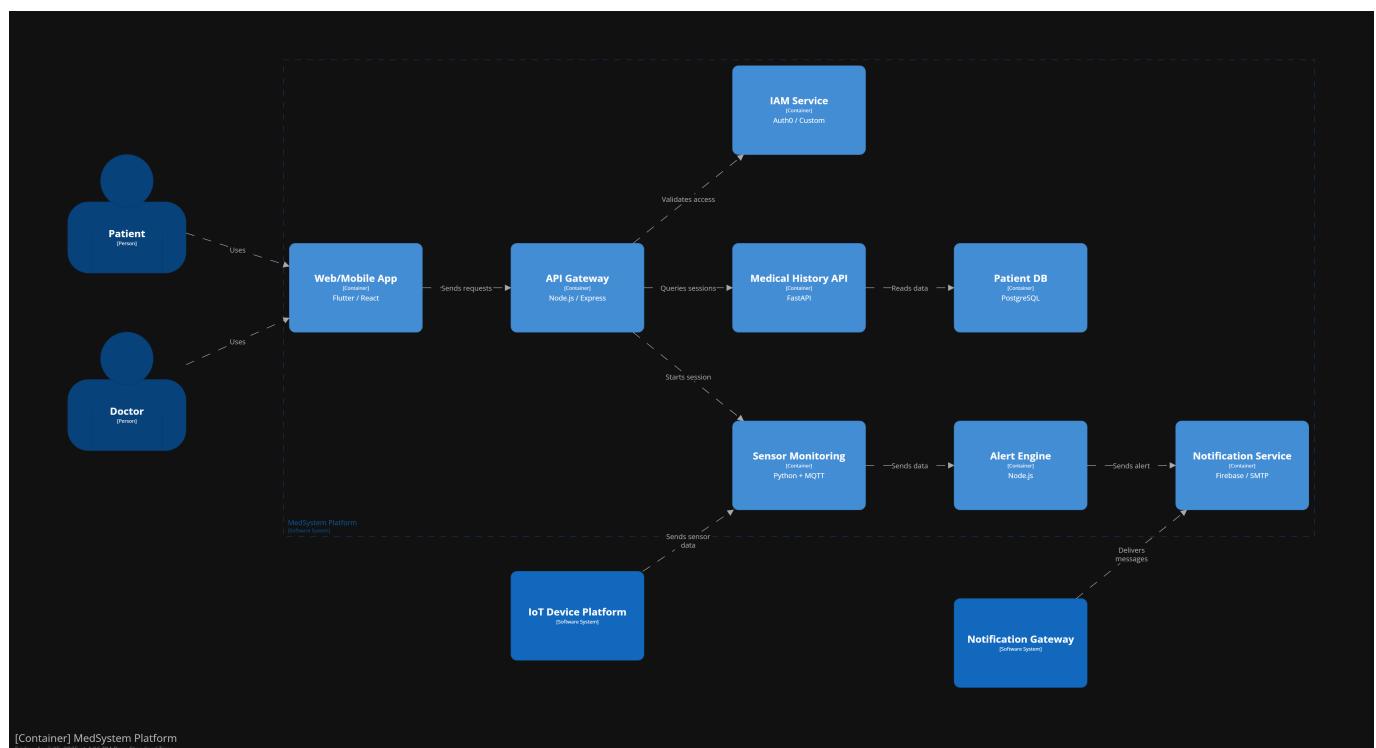
4.1.3.3. Software Architecture Container Level Diagrams.

Container	Technology	Main Responsibility
Web/Mobile App	Flutter / React	Interface for patients and doctors

Container	Technology	Main Responsibility
API Gateway	Node.js / Express	Management of REST requests and authentication
IAM Service	Auth0 / Custom Auth	Handling login, roles, and permissions
Sensor Monitoring	Python + MQTT	Collection and validation of heart rate sensor data
Alert Engine	Node.js	Detection of anomalies and alert generation
Notification Service	Firebase / SMTP	Sending notifications to patients and doctors
Medical History API	Python / FastAPI	Consultation of medical histories and reports
Database	PostgreSQL / MongoDB	Storage of users, sessions, and medical records

Notes:

- The app distinguishes between patients and doctors through roles managed in IAM.
- Critical alerts generated by the system are automatically sent to the assigned doctor.



4.1.3.4. Software Architecture Deployment Diagrams.

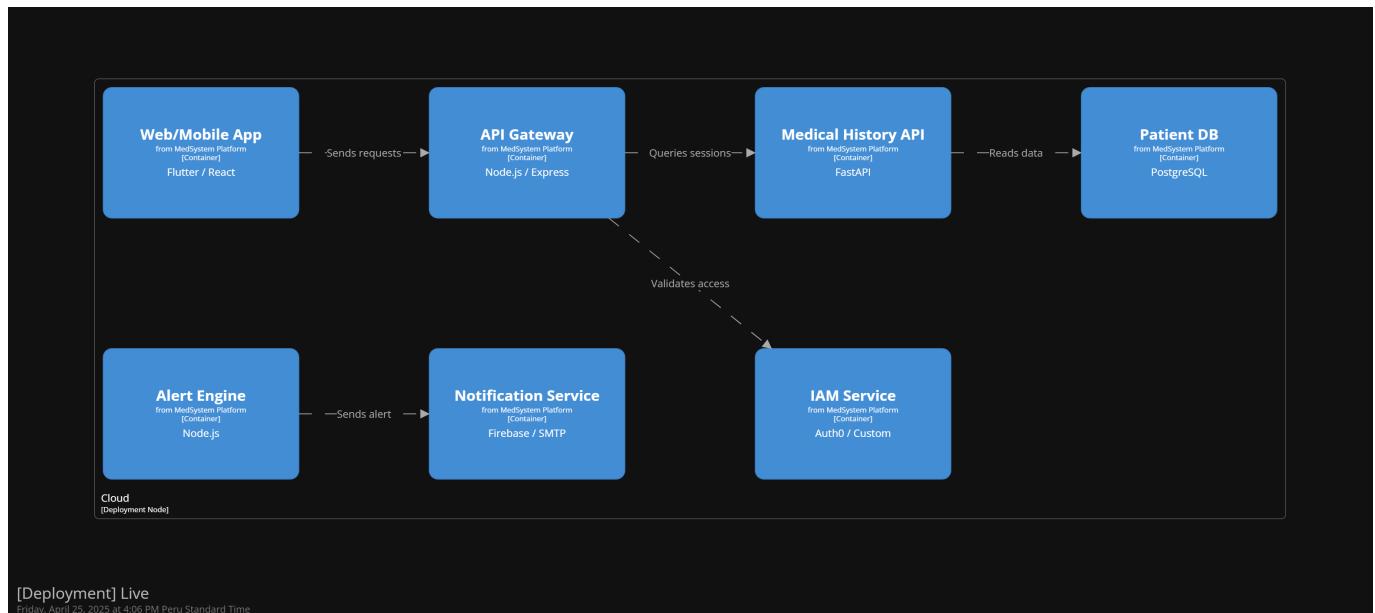
Entorno de despliegue:

Nube pública (AWS, Azure o GCP):

- Contenedores desplegados en instancias de aplicación o Kubernetes.
- Base de datos (RDS, Cloud SQL o equivalente) para almacenar registros de usuarios y sesiones.
- IoT Core o MQTT Broker para recibir datos de sensores en tiempo real.
- Servicio de notificaciones externas configurado (Firebase Cloud Messaging, Twilio, SMTP Gateway).

Diagrama de despliegue conceptual:

- Paciente y Doctor → Web/Mobile App → API Gateway
- API Gateway → IAM Service (autenticación) + Sensor Monitoring (vía eventos)
- Sensor Monitoring → Alert Engine → Notification Service → Paciente/Doctor
- Medical History API → Base de datos



link: <https://structurizr.com/share/101697/ddc19bf8-07c7-4949-ad6f-375475a613d5>

4.2. Tactical-Level Domain-Driven Design

4.2.1. Bounded Context:

El dominio de **Identity and Access Management (IAM)** en SafeMed es fundamental para la seguridad y el correcto funcionamiento de la plataforma. Su objetivo principal es controlar quién puede acceder al sistema y qué acciones puede realizar. Esto se logra mediante la gestión centralizada de usuarios (Pacientes y Doctores), la definición de roles y permisos, y la validación de la identidad y los privilegios en cada interacción con el sistema. El contexto IAM asegura que la información médica sensible esté protegida y que cada usuario opere dentro de los límites de su autorización.

Diccionario de Clases

En esta sección, presentamos el **Diccionario de Clases** para el contexto IAM, detallando las entidades principales que gestionan la identidad y el acceso en SafeMed.

1. User

Atributo	Tipo de Dato	Descripción
UserId	GUID	Identificador único del usuario.
Email	string	Correo electrónico, usado para el inicio de sesión.
PasswordHash	string	Contraseña del usuario (cifrada).
FirstName	string	Nombre del usuario.

Atributo	Tipo de Dato	Descripción
LastName	string	Apellido del usuario.
UserType	enum	Tipo de usuario (Paciente, Doctor).
RegistrationDate	datetime	Fecha en que se registró el usuario.
IsActive	boolean	Indica si la cuenta del usuario está activa.
LastLoginDate	datetime	Fecha del último inicio de sesión.

2. Role

Atributo	Tipo de Dato	Descripción
RoleId	int	Identificador único del rol.
RoleName	string	Nombre descriptivo del rol ("Paciente", "Doctor").
Description	string	Breve descripción de las responsabilidades del rol.

3. Permission

Atributo	Tipo de Dato	Descripción
PermissionId	int	Identificador único del permiso.
PermissionName	string	Nombre técnico del permiso ("view_heart_rate").
Description	string	Breve descripción de la acción permitida.

4. UserRole

Atributo	Tipo de Dato	Descripción
UserId	GUID	Clave foránea a la entidad User.
RoleId	int	Clave foránea a la entidad Role.

5. RolePermission

Atributo	Tipo de Dato	Descripción
RoleId	int	Clave foránea a la entidad Role.
PermissionId	int	Clave foránea a la entidad Permission.

4.2.1.1. Domain Layer.

En la capa de dominio se modelan los conceptos centrales del contexto IAM siguiendo los principios de **Domain-Driven Design**, adaptados a las necesidades de **SafeMed**.

El agregado raíz **User** representa a un usuario dentro del sistema SafeMed, ya sea un **Paciente** o un **Doctor**. La entidad **User** es el punto de control principal para las operaciones de identidad y acceso. Cada **User** está asociado a uno o varios **Role** a través de la entidad de enlace **UserRole**.

Las entidades **Role** y **Permission** definen los tipos de usuarios y las acciones que pueden realizar. Si bien los roles y permisos pueden ser gestionados hasta cierto punto, sus definiciones fundamentales y la validación del acceso residen en esta capa.

El tipo de usuario (**UserType**), modelado como un **Value Object** o un simple atributo enumerado, establece la categoría principal del usuario (**Paciente** o **Doctor**).

Adicionalmente, los **servicios de dominio**, como un **UserAuthenticationService**, pueden encapsular operaciones complejas que involucran múltiples entidades o que no son responsabilidad natural de una sola entidad, como el proceso de inicio de sesión que verifica credenciales y genera una sesión.

Aggregates

Aggregate `User

- **UserId** (GUID/UUID)
- **Email** (string)
- **PasswordHash** (string)
- **FirstName** (string)
- **LastName** (string)
- **UserType** (enum: Paciente, Doctor)
- **RegistrationDate** (datetime)
- **IsActive** (boolean)
- **LastLoginDate** (datetime)
- **List userRoles**

Métodos:

- `Register(email, password, userType): User`
- `Authenticate(password, hashedPassword): boolean`
- `UpdateProfile(firstName, lastName): void`
- `AssignRole(role): void`
- `RemoveRole(roleId): void`
- `HasPermission(permissionName, roleRepository): boolean`

Entities

Entity `Role

- **RoleId** (int/GUID)
- **RoleName** (string)
- **Description** (string)

Métodos:

- `GetPermissions(permissionRepository): List<Permission>`

Entity `Permission

- **PermissionId** (int/GUID)
- **PermissionName** (string)
- **Description** (string)

Entity `UserRole

- **UserId** (GUID/UUID)
- **RoleId** (int/GUID)

Value Objects

Value Object `UserType`

- **Paciente**
- **Doctor**

Métodos estándar de enum:

- `name(): string`
- `values(): UserType[]`

Domain Services

UserAuthenticationService

- `Authenticate(email, password, userRepository): User`
- `GenerateAuthToken(user): string`
- `ValidateAuthToken(token): User`

UserRegistrationService

- `RegisterNewUser(email, password, userType, userRepository, roleRepository): User`

4.2.1.2. Interface Layer.

La **Capa de Interfaz** para el contexto de **Identity and Access Management (IAM)** en SafeMed sirve como la capa de presentación y adaptación que expone las funcionalidades del dominio a las aplicaciones cliente (móvil y web) y potencialmente a otros servicios.

Se encarga de:

- Recibir las peticiones externas.
- Validar los datos de entrada.
- Invocar la lógica de negocio correspondiente en la Capa de Aplicación.
- Formatear las respuestas.

Los componentes principales de esta capa son los **Controladores**, que gestionan los diferentes aspectos de la interacción del usuario con el sistema IAM a través de endpoints API.

Controladores

Controlador	Responsabilidad Principal
AuthController	Maneja las operaciones de autenticación (registro, login, logout).
UsersController	Gestiona los perfiles de usuario y consultas relacionadas.

Endpoints Clave por Controlador

Controlador	Método HTTP	Endpoint	Descripción
-------------	-------------	----------	-------------

Controlador	Método HTTP	Endpoint	Descripción
AuthController	POST	/api/v1/iam/auth/register	Registra un nuevo usuario (Paciente o Doctor).
AuthController	POST	/api/v1/iam/auth/login	Autentica a un usuario y devuelve un token.
AuthController	POST	/api/v1/iam/auth/logout	Cierra la sesión o invalida el token del usuario.
UsersController	GET	/api/v1/iam/users/{userId}	Obtiene los detalles del perfil de un usuario.
UsersController	PUT	/api/v1/iam/users/{userId}	Actualiza los datos del perfil de un usuario.
UsersController	GET	/api/v1/iam/users/{userId}/permissions	Obtiene la lista de permisos de un usuario.

Enfoque

Esta capa se centra en la comunicación externa y en la traducción de solicitudes, **sin implementar lógica de negocio compleja** que corresponde a las capas de Dominio o Aplicación.

4.2.1.3. Application Layer.

La **Capa de Aplicación** dentro del contexto de **Identity and Access Management (IAM)** en SafeMed es responsable de orquestar el flujo de trabajo para los casos de uso relacionados con la gestión de usuarios y accesos.

Actúa como una capa delgada que coordina las interacciones entre:

- La Capa de Interfaz.
- La Capa de Dominio.
- La Capa de Infraestructura.

Esta capa **no contiene lógica de negocio compleja**, sino que delega las decisiones y reglas a la Capa de Dominio y utiliza la Capa de Infraestructura para la persistencia y servicios externos.

Los componentes principales de esta capa son los **Command Handlers** y **Query Handlers**, que procesan los **Commands** (intenciones de cambiar el estado del sistema) y **Queries** (intenciones de obtener información), respectivamente.

Comandos (Commands)

Representan intenciones del usuario o del sistema para realizar una acción que cambia el estado del contexto IAM.

Comando	Descripción
RegisterUserCommand	Solicita el registro de un nuevo usuario en el sistema.
LoginUserCommand	Solicita la autenticación de un usuario.

Comando	Descripción
UpdateUserProfileCommand	Solicita la actualización de los datos de perfil de un usuario.
AssignRoleToUserCommand	Solicita la asignación de un rol específico a un usuario.
ChangePasswordCommand	Solicita el cambio de contraseña de un usuario.

Queries

Representan solicitudes de información del usuario o del sistema que **no cambian el estado** del contexto IAM.

Query	Descripción
GetUserProfileQuery	Solicita los datos de perfil de un usuario por su ID.
GetUserPermissionsQuery	Solicita la lista de permisos asociados a un usuario.
GetAllRolesQuery	Solicita la lista de todos los roles disponibles.

Command Handlers

Componentes que procesan los **Commands**, orquestando la lógica de dominio y la persistencia.

Command Handler	Comandos que Maneja
RegisterUserCommandHandler	RegisterUserCommand
LoginUserCommandHandler	LoginUserCommand
UpdateUserProfileCommandHandler	UpdateUserProfileCommand
AssignRoleToUserCommandHandler	AssignRoleToUserCommand
ChangePasswordCommandHandler	ChangePasswordCommand

Query Handlers

Componentes que procesan las **Queries**, obteniendo datos a través de la Capa de Infraestructura o servicios de dominio.

Query Handler	Queries que Maneja
GetUserProfileQueryHandler	GetUserProfileQuery
GetUserPermissionsQueryHandler	GetUserPermissionsQuery
GetAllRolesQueryHandler	GetAllRolesQuery

Otros Componentes

Además de los **Command Handlers** y **Query Handlers**, la Capa de Aplicación también puede incluir **Event Handlers** que reaccionan a eventos de dominio publicados por este o por otros contextos (por ejemplo, un **UserRegisteredEventHandler** que envía un correo de confirmación).

4.2.1.4. Infrastructure Layer.

La **Capa de Infraestructura** implementa los mecanismos de persistencia requeridos para soportar el dominio de IAM de SafeMed y gestionar la interacción con la base de datos subyacente.

En esta capa:

- Se materializan las interfaces de repositorio definidas en la Capa de Dominio.
- Se utiliza un **ORM** (Object-Relational Mapper) para manejar las operaciones de acceso a datos.

Los repositorios proporcionan métodos necesarios para guardar, recuperar y buscar entidades del dominio IAM (**User, Role, Permission**).

Además de las operaciones CRUD básicas, se implementan métodos de consulta específicos, como:

- Buscar un usuario por su correo electrónico.
- Verificar la existencia de un rol por su nombre.

Esto permite mantener la **Capa de Dominio** pura y enfocada en la lógica de negocio, favoreciendo una arquitectura limpia y sostenible.

Repositorios Clave

Infrastructure Repository `UserRepository

- **Implementa:** `IUserRepository`
- **Tecnología:** ORM (ej: Entity Framework Core)

Métodos:

- `FindByEmail(email: string): Optional<User>`
- `ExistsByEmail(email: string): boolean`
- `Add(user: User): void`
- `Update(user: User): void`
- `FindById(userId: GUID): Optional<User>`

Infrastructure Repository `RoleRepository

- **Implementa:** `IRoleRepository`
- **Tecnología:** ORM (ej: Entity Framework Core)

Métodos:

- `FindByName(name: string): Optional<Role>`
- `ExistsByName(name: string): boolean`
- `FindById(roleId: int): Optional<Role>`
- `GetAll(): List<Role>`

Infrastructure Repository `PermissionRepository

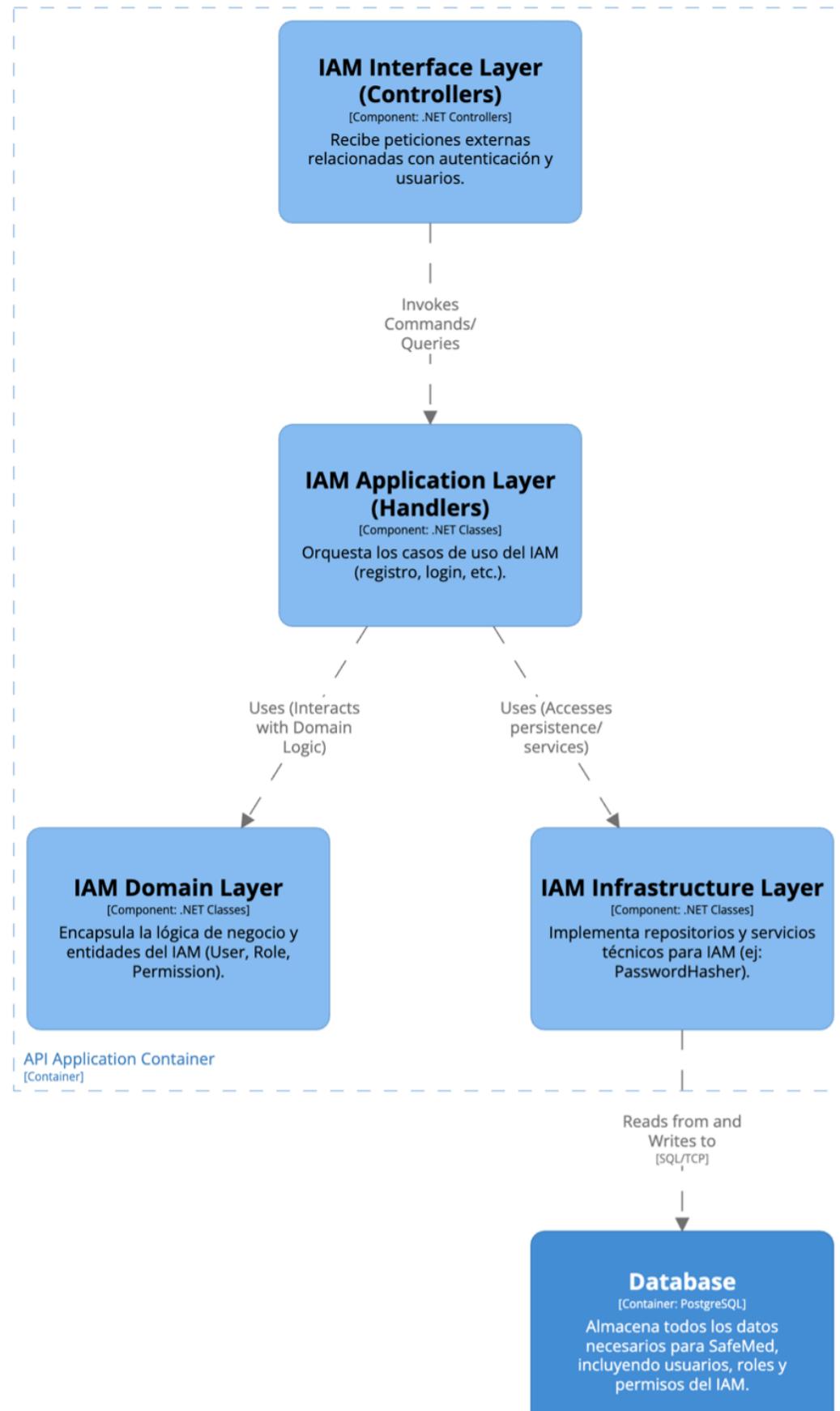
- **Implementa:** `IPermissionRepository`
- **Tecnología:** ORM (ej: Entity Framework Core)

Métodos:

- `FindByName(name: string): Optional<Permission>`

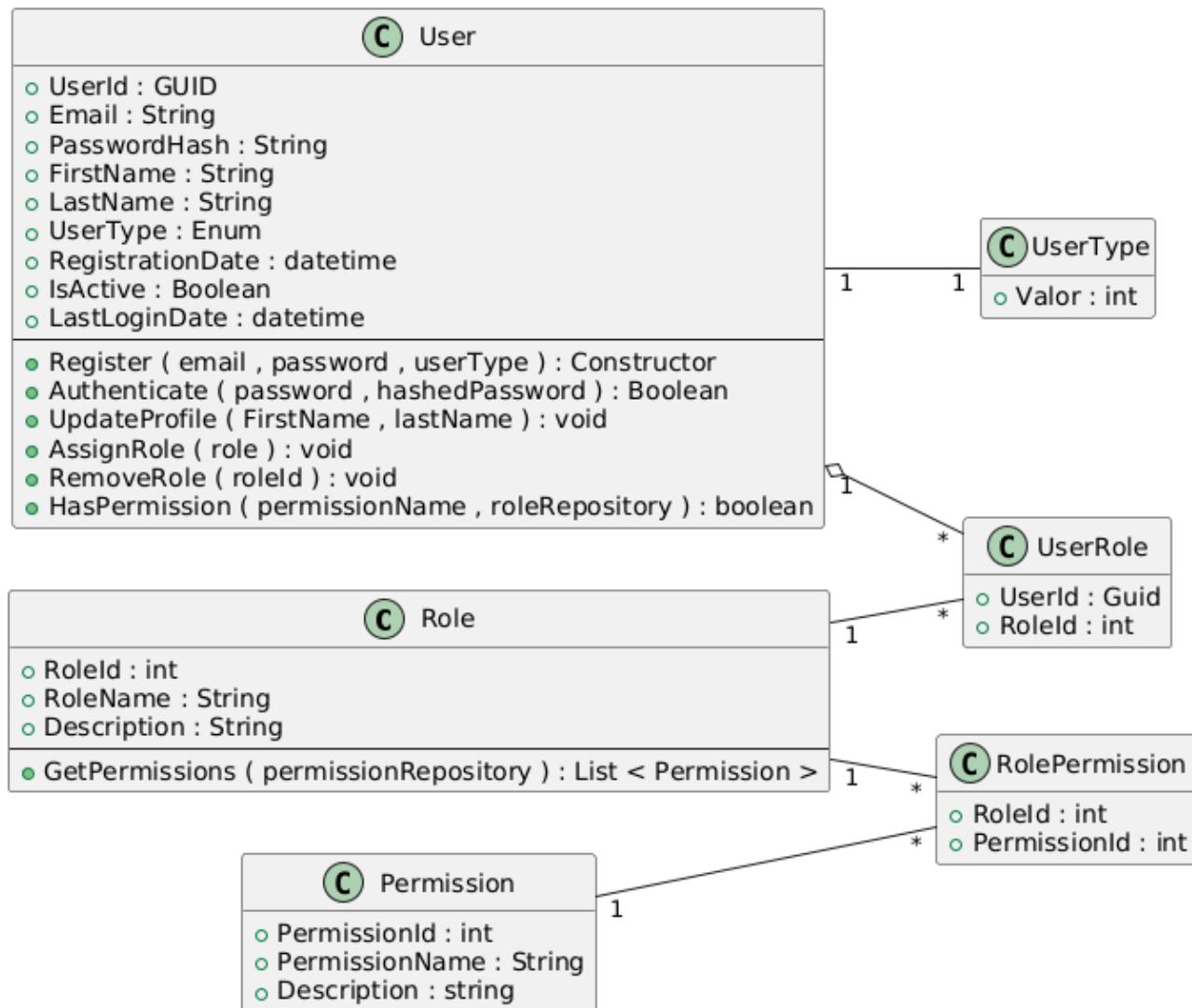
- `FindById(permissionId: int): Optional<Permission>`
- `GetAll(): List<Permission>`

4.2.1.5. Bounded Context Software Architecture Component Level Diagrams.

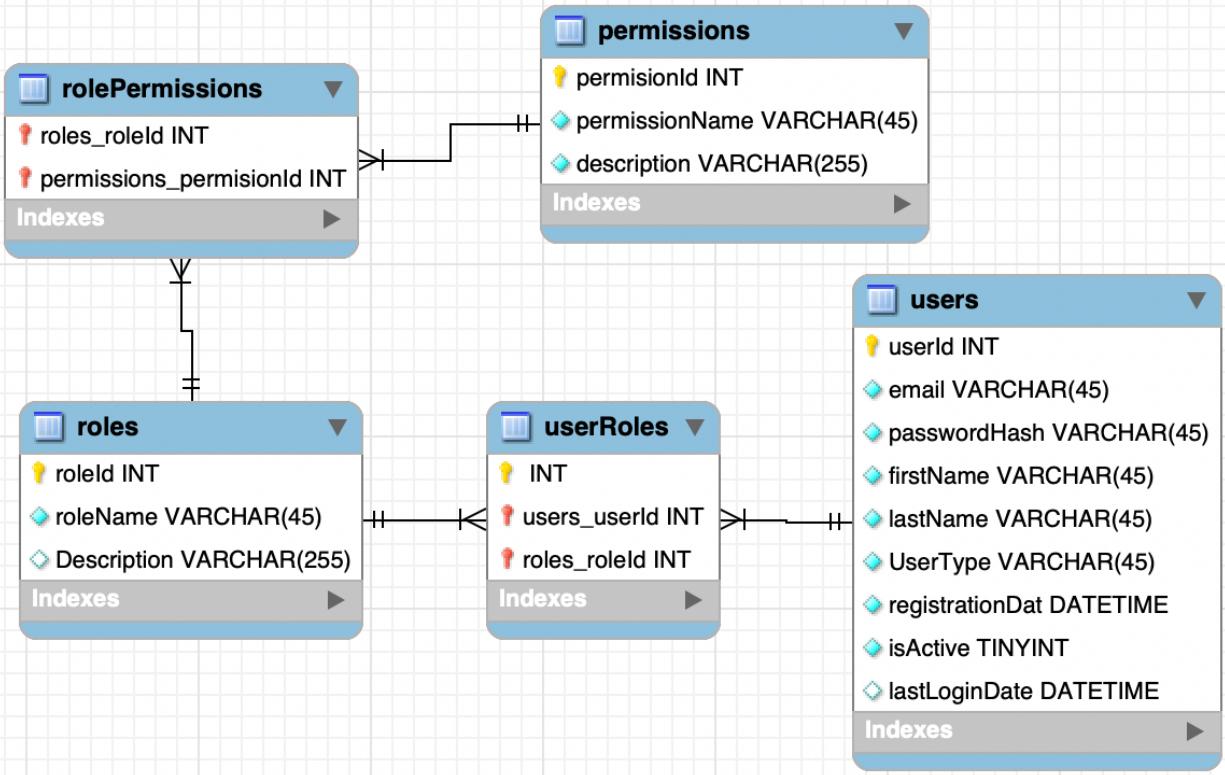


4.2.1.6. Bounded Context Software Architecture Code Level Diagrams.

4.2.1.6.1. Bounded Context Domain Layer Class Diagrams.



4.2.1.6.2. Bounded Context Database Design Diagram.



4.2.2. Bounded Context:

El dominio de **Patient Management** en SafeMed se centra en la gestión de la información y el estado de los pacientes dentro de la plataforma.

Su responsabilidad principal es:

- Mantener los perfiles de los pacientes.
- Gestionar sus contactos de emergencia.
- Establecer las relaciones con los médicos encargados de su cuidado.

Este contexto asegura que:

- La información demográfica y de contacto del paciente esté actualizada.
- Las asignaciones a médicos y contactos de emergencia estén correctamente configuradas.
- Se permita el monitoreo y respuesta oportuna en caso de emergencias.

Diccionario de Clases

En esta sección, presentamos las entidades y posibles objetos de valor clave que forman parte del dominio de **Patient Management**.

1. Patient

Atributo	Tipo de Dato	Descripción
PatientId	GUID/UUID	Identificador único del paciente. Clave primaria.
UserId	GUID/UUID	Clave foránea al User en el contexto IAM. Identifica el usuario asociado.

Atributo	Tipo de Dato	Descripción
DateOfBirth	date	Fecha de nacimiento del paciente.
Gender	string	Género del paciente.
Address	string	Dirección de residencia del paciente.
PhoneNumber	string	Número de teléfono de contacto del paciente.
MedicalHistorySummary	string	Resumen breve de la historia médica relevante (no historia clínica completa).

2. DoctorAssignment

Atributo	Tipo de Dato	Descripción
AssignmentId	GUID/UUID	Identificador único de la asignación. Clave primaria.
PatientId	GUID/UUID	Clave foránea al Patient. Identifica al paciente asignado.
DoctorUserId	GUID/UUID	Clave foránea al User (con rol Doctor) en el contexto IAM.
AssignmentDate	datetime	Fecha en que se realizó la asignación.
IsActive	boolean	Indica si la asignación del médico está activa actualmente.

3. EmergencyContact

Atributo	Tipo de Dato	Descripción
ContactId	GUID/UUID	Identificador único del contacto de emergencia. Clave primaria.
PatientId	GUID/UUID	Clave foránea al Patient. Identifica al paciente asociado.
FullName	string	Nombre completo del contacto de emergencia.
Relationship	string	Parentesco o relación con el paciente (ej: "Hijo", "Cónyuge").
PhoneNumber	string	Número de teléfono del contacto de emergencia (para SMS).
Email	string	Correo electrónico del contacto de emergencia (opcional).
IsPrimary	boolean	Indica si es el contacto de emergencia principal.

4.2.2.1. Domain Layer.

La **Capa de Dominio** del contexto de **Patient Management** encapsula las entidades, objetos de valor y reglas de negocio que rigen la información y el estado de los pacientes en SafeMed.

Es el corazón de este bounded context, donde reside la lógica esencial de la gestión de perfiles, contactos y asignaciones.

El **Aggregate Root** principal en este dominio es la entidad **Patient**. El **Patient** es la entidad coherente a través de la cual se acceden y modifican sus partes constituyentes, como **EmergencyContact** y **DoctorAssignment**.

Garantiza que las operaciones sobre los contactos y asignaciones mantengan la integridad del estado general del paciente.

Las entidades **EmergencyContact** y **DoctorAssignment** son partes del agregado **Patient**. Sus ciclos de vida están gestionados por el **Patient**.

- **EmergencyContact** representa la información de personas a contactar en caso de una emergencia del paciente.
- **DoctorAssignment** registra la relación entre un paciente y un médico específico dentro de la plataforma.

El dominio de **Patient Management** también necesita hacer referencia a entidades de otros contextos sin poseerlas. Notablemente, un **Patient** y un **DoctorAssignment** referenciarán **Users** del contexto IAM (mediante sus IDs) para identificar al paciente usuario y al médico usuario asignado, respectivamente.

Aggregate Root

Aggregate Root Patient

Atributo	Tipo de Dato	Descripción
PatientId	GUID/UUID	Identificador único del paciente.
UserId	GUID/UUID	Identificador del usuario asociado (IAM).
DateOfBirth	date	Fecha de nacimiento del paciente.
Gender	string	Género del paciente.
Address	string	Dirección de residencia.
PhoneNumber	string	Teléfono de contacto.
MedicalHistorySummary	string	Resumen breve de historia médica relevante.
List	-	Lista de contactos de emergencia.
List	-	Lista de asignaciones a doctores.

Métodos:

- `Create(patientId, userId, dateOfBirth, gender, address, phoneNumber, medicalHistorySummary): Patient`
- `UpdateProfile(dateOfBirth, gender, address, phoneNumber, medicalHistorySummary): void`
- `AddEmergencyContact(contactId, fullName, relationship, phoneNumber, email, isPrimary): void`
- `RemoveEmergencyContact(contactId): void`
- `UpdateEmergencyContact(contactId, fullName, relationship, phoneNumber, email, isPrimary): void`
- `MarkPrimaryEmergencyContact(contactId): void`
- `AssignDoctor(assignmentId, doctorUserId, assignmentDate): void`
- `RemoveDoctorAssignment(assignmentId): void`
- `GetPrimaryEmergencyContact(): Optional<EmergencyContact>`
- `GetActiveDoctorAssignment(): Optional<DoctorAssignment>`
- `GetEmergencyContactById(contactId): Optional<EmergencyContact>`
- `GetDoctorAssignmentById(assignmentId): Optional<DoctorAssignment>`

Entities (dentro del Agregado Patient)

Entity `EmergencyContact

Atributo	Tipo de Dato	Descripción
ContactId	GUID/UUID	Identificador único del contacto de emergencia.
PatientId	GUID/UUID	Identificador del paciente asociado.
FullName	string	Nombre completo del contacto.
Relationship	string	Relación con el paciente (ej: "Hijo", "Cónyuge").
PhoneNumber	string	Número de teléfono de contacto.
Email	string	Correo electrónico del contacto (opcional).
IsPrimary	boolean	Indica si es el contacto principal.

Métodos:

- `UpdateDetails(fullName, relationship, phoneNumber, email): void`
- `MarkAsPrimary(): void`
- `MarkAsSecondary(): void`

Entity `DoctorAssignment

Atributo	Tipo de Dato	Descripción
AssignmentId	GUID/UUID	Identificador único de la asignación.
PatientId	GUID/UUID	Identificador del paciente asociado.
DoctorUserId	GUID/UUID	Identificador del médico usuario asignado (IAM).
AssignmentDate	datetime	Fecha en que se realizó la asignación.
EndDate	datetime	Fecha en que terminó la asignación (si aplica).
IsActive	boolean	Indica si la asignación está activa.

Métodos:

- `Activate(): void`
- `Deactivate(endDate): void`
- `IsCurrent(): boolean`

4.2.2.2. Interface Layer.

La **Capa de Interfaz** para el contexto de **Patient Management** en SafeMed actúa como la fachada o el punto de entrada a las funcionalidades de este dominio.

Es responsable de:

- Recibir las peticiones de las aplicaciones cliente (móvil y web) u otros servicios.
- Validar los datos de entrada.
- Invocar la lógica de negocio correspondiente en la Capa de Aplicación.

- Formatear las respuestas que se enviarán de vuelta al cliente.

Los componentes principales de esta capa son los **Controladores**, que gestionan los diferentes aspectos de la interacción del usuario con el dominio de Patient Management a través de endpoints API RESTful.

Controladores

Controlador	Responsabilidad Principal
PatientsController	Gestiona las operaciones sobre los perfiles de pacientes (crear, obtener, actualizar).
EmergencyContactsController	Gestiona los contactos de emergencia de un paciente (añadir, obtener, actualizar, eliminar).
DoctorAssignmentsController	Gestiona la asignación de médicos a pacientes (asignar, obtener, finalizar).

Endpoints Clave por Controlador

Controlador	Método HTTP	Endpoint	Descripción
PatientsController	POST	/api/v1/patients	Crea un nuevo perfil de paciente (asociado a un usuario IAM).
PatientsController	GET	/api/v1/patients/{patientId}	Obtiene los detalles de un perfil de paciente.
PatientsController	PUT	/api/v1/patients/{patientId}	Actualiza los datos de un perfil de paciente.
EmergencyContactsController	POST	/api/v1/patients/{patientId}/emergency-contacts	Añade un contacto de emergencia a un paciente.
EmergencyContactsController	GET	/api/v1/patients/{patientId}/emergency-contacts	Obtiene todos los contactos de emergencia de un paciente.
EmergencyContactsController	PUT	/api/v1/patients/{patientId}/emergency-contacts/{contactId}	Actualiza un contacto de emergencia.

Controlador	Método HTTP	Endpoint	Descripción
EmergencyContactsController	DELETE	/api/v1/patients/{patientId}/emergency-contacts/{contactId}	Elimina un contacto de emergencia.
DoctorAssignmentsController	POST	/api/v1/patients/{patientId}/doctor-assignments	Asigna un médico a un paciente.
DoctorAssignmentsController	GET	/api/v1/patients/{patientId}/doctor-assignments	Obtiene las asignaciones de médicos de un paciente.
DoctorAssignmentsController	PUT	/api/v1/patients/{patientId}/doctor-assignments/{assignmentId}	Actualiza (ej: desactiva) una asignación de médico.

4.2.2.3. Application Layer.

La **Capa de Aplicación** dentro del contexto de **Patient Management** en SafeMed es responsable de orquestar el flujo de trabajo para los casos de uso relacionados con la gestión de perfiles de pacientes, contactos de emergencia y asignaciones de médicos.

Actúa como una capa delgada que coordina las interacciones entre:

- La Capa de Interfaz.
- La Capa de Dominio.
- La Capa de Infraestructura.

Esta capa **no contiene lógica de negocio compleja**, sino que delega las decisiones y reglas a la Capa de Dominio y utiliza la Capa de Infraestructura para la persistencia y servicios externos.

Los componentes principales son los **Command Handlers** y **Query Handlers**, que procesan **Commands** (acciones que cambian el estado) y **Queries** (solicitudes de información), respectivamente.

Comandos (Commands)

Representan intenciones del usuario o del sistema para realizar una acción que cambia el estado del contexto Patient Management.

Comando	Descripción
CreatePatientProfileCommand	Solicita la creación de un nuevo perfil de paciente asociado a un usuario IAM.
UpdatePatientProfileCommand	Solicita la actualización de los datos del perfil de un paciente.
AddEmergencyContactCommand	Solicita añadir un contacto de emergencia a un paciente.
UpdateEmergencyContactCommand	Solicita actualizar los datos de un contacto de emergencia.

Comando	Descripción
RemoveEmergencyContactCommand	Solicita eliminar un contacto de emergencia.
AssignDoctorCommand	Solicita la asignación de un médico a un paciente.
UpdateDoctorAssignmentCommand	Solicita actualizar una asignación de médico (ej: desactivarla).
MarkPrimaryEmergencyContactCommand	Solicita marcar un contacto de emergencia como principal.

Queries

Representan solicitudes de información del usuario o del sistema que **no cambian el estado** del contexto Patient Management.

Query	Descripción
GetPatientProfileQuery	Solicita los detalles del perfil de un paciente por ID.
GetPatientContactsQuery	Solicita todos los contactos de emergencia de un paciente.
GetPatientAssignmentsQuery	Solicita todas las asignaciones de médicos de un paciente.
GetPatientsByDoctorQuery	Solicita la lista de pacientes asignados a un médico.
GetPrimaryEmergencyContactQuery	Solicita el contacto de emergencia principal de un paciente.

Command Handlers

Componentes que procesan los **Commands**, orquestando la lógica de dominio y la persistencia.

Command Handler	Comandos que Maneja
CreatePatientProfileCommandHandler	CreatePatientProfileCommand
UpdatePatientProfileCommandHandler	UpdatePatientProfileCommand
AddEmergencyContactCommandHandler	AddEmergencyContactCommand
UpdateEmergencyContactCommandHandler	UpdateEmergencyContactCommand
RemoveEmergencyContactCommandHandler	RemoveEmergencyContactCommand
AssignDoctorCommandHandler	AssignDoctorCommand
UpdateDoctorAssignmentCommandHandler	UpdateDoctorAssignmentCommand
MarkPrimaryEmergencyContactCommandHandler	MarkPrimaryEmergencyContactCommand

Query Handlers

Componentes que procesan las **Queries**, obteniendo datos a través de la Capa de Infraestructura o servicios de dominio.

Query Handler	Queries que Maneja
GetPatientProfileQueryHandler	GetPatientProfileQuery

Query Handler	Queries que Maneja
GetPatientContactsQueryHandler	GetPatientContactsQuery
GetPatientAssignmentsQueryHandler	GetPatientAssignmentsQuery
GetPatientsByDoctorQueryHandler	GetPatientsByDoctorQuery
GetPrimaryEmergencyContactQueryHandler	GetPrimaryEmergencyContactQuery

4.2.2.4. Infrastructure Layer

La **Capa de Infraestructura** para **Patient Management** se encarga de la persistencia de los datos relacionados con los perfiles de pacientes, contactos de emergencia y asignaciones de médicos.

Sus responsabilidades principales son:

- Implementar los repositorios definidos en la Capa de Dominio.
- Interactuar con la base de datos subyacente para almacenar y recuperar la información del agregado **Patient** y sus partes constituyentes (**EmergencyContact**, **DoctorAssignment**).
- Abstraer los detalles técnicos de la base de datos del resto del bounded context.

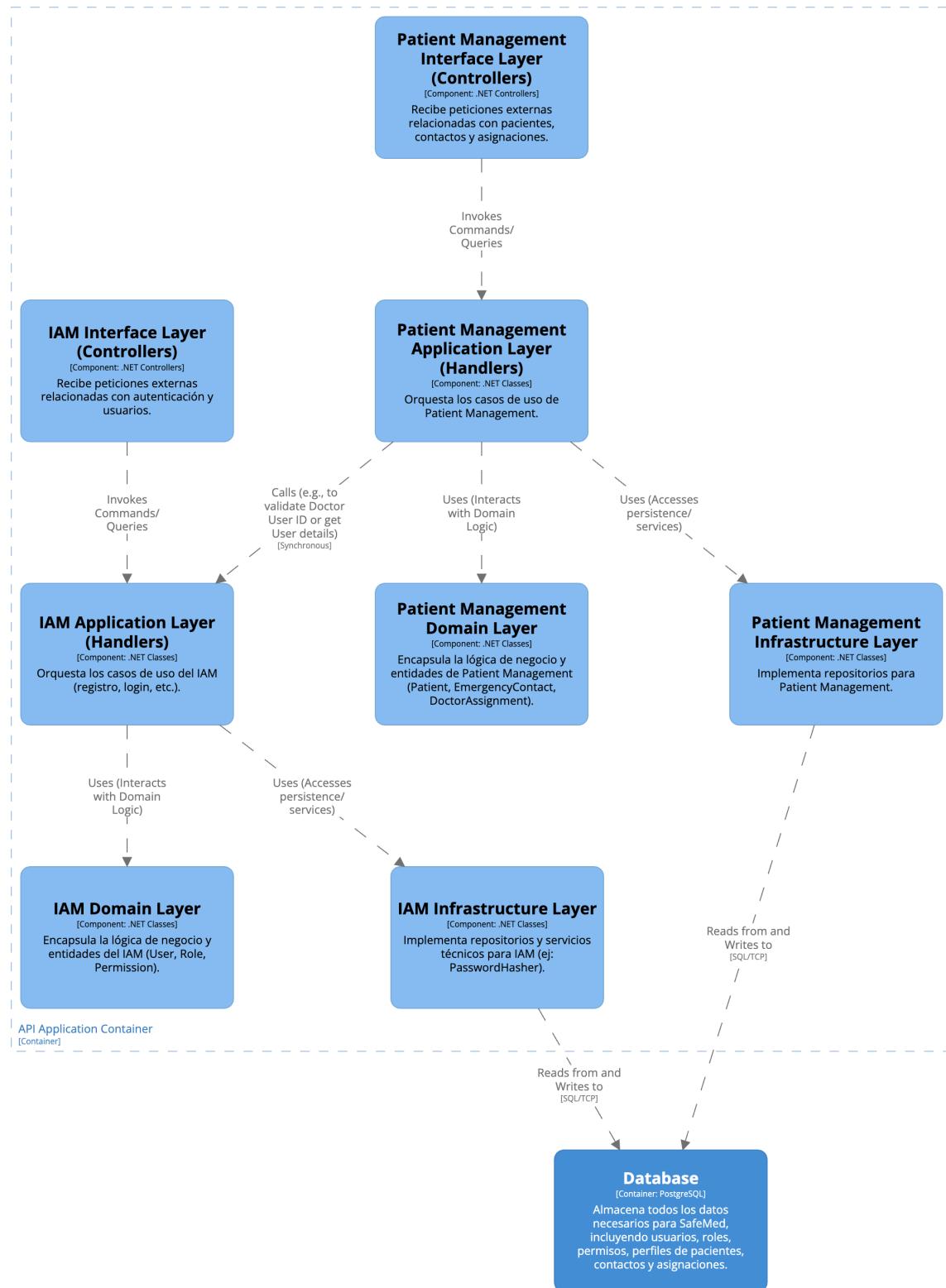
Los componentes principales de esta capa incluyen las implementaciones concretas de los **repositorios** y, potencialmente, **servicios de bajo nivel** específicos para este contexto.

Componentes de Infraestructura Clave

Componente	Tipo	Descripción	Tecnología/Implementación Típica
PatientRepository	Implementación Repositorio	Implementa la interfaz IPatientRepository . Gestiona la persistencia transaccional del agregado completo Patient (incluyendo EmergencyContact y DoctorAssignment).	ORM
PatientManagementDbContext	Contexto DB/ORM	Configuración y gestión del acceso a la base de datos específico para el esquema de Patient Management.	ORM
EmergencyContactRepository	Implementación Repositorio	Podría ser gestionado por PatientRepository si se modela como entidad interna, o como repositorio secundario para acceso directo.	ORM

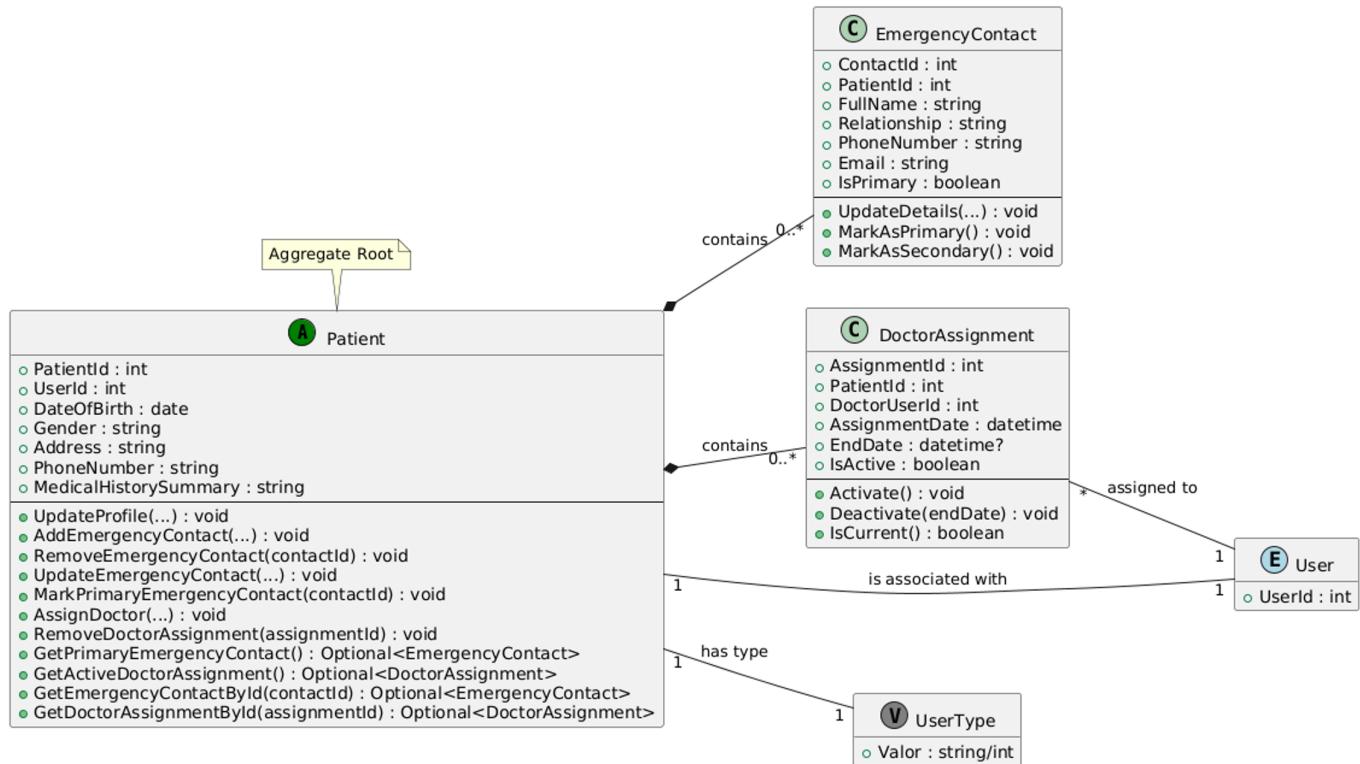
Componente	Tipo	Descripción	Tecnología/Implementación Típica
DoctorAssignmentRepository	Implementación Repository	Podría ser gestionado por PatientRepository o como repositorio secundario si requiere un ciclo de vida independiente.	ORM

4.2.2.5. Bounded Context Software Architecture Component Level Diagrams.

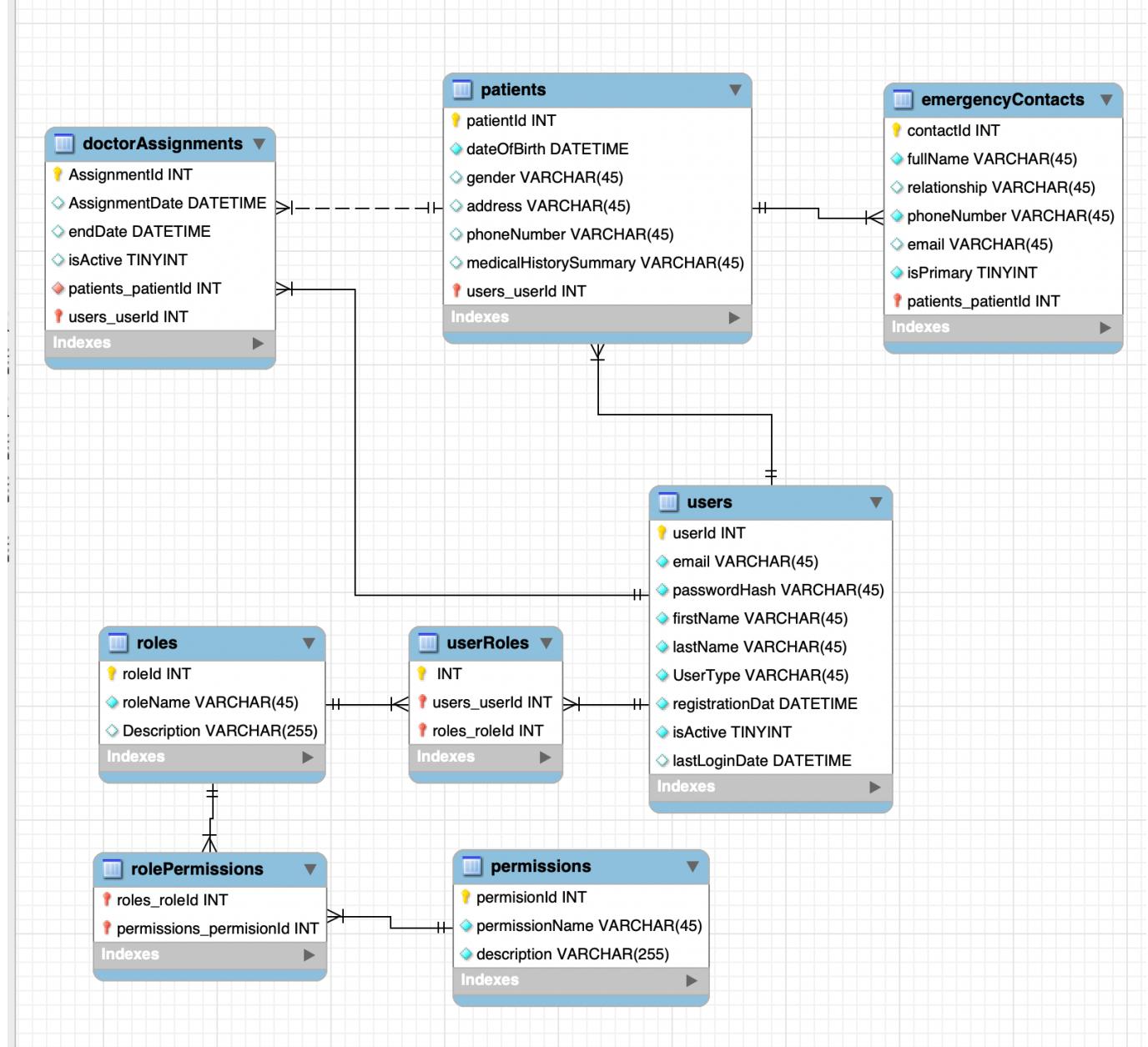


4.2.2.6. Bounded Context Software Architecture Code Level Diagrams.

4.2.2.6.1. Bounded Context Domain Layer Class Diagrams.



4.2.2.6.2. Bounded Context Database Design Diagram.



4.2.3. Bounded Context: Sensor Monitoring

El dominio de **Sensor Monitoring** en SafeMed es fundamental para la supervisión en tiempo real de la salud del paciente. Su objetivo principal es capturar, almacenar y visualizar los datos del **ritmo cardíaco** provenientes de sensores conectados, permitiendo a pacientes y doctores realizar un seguimiento continuo y seguro de los signos vitales. Este contexto asegura que los datos biométricos se registren de manera precisa y estén disponibles en la aplicación para su visualización, análisis y toma de decisiones médicas.

Diccionario de Clases

En esta sección, presentamos el **Diccionario de Clases** para el contexto de **Sensor Monitoring**, detallando las entidades principales que gestionan la captura y manejo del ritmo cardíaco.

1. HeartRateReading

Atributo	Tipo de Dato	Descripción
ReadingId	GUID	Identificador único de la lectura.

Atributo	Tipo de Dato	Descripción
PatientId	GUID	Identificador del paciente asociado.
Timestamp	datetime	Momento exacto de la lectura.
HeartRateBPM	int	Ritmo cardíaco registrado (latidos por minuto).

2. SensorDevice

Atributo	Tipo de Dato	Descripción
DeviceId	GUID	Identificador único del dispositivo.
SerialNumber	string	Número de serie del sensor.
PatientId	GUID	Identificador del paciente asignado.
IsActive	boolean	Estado de activación del sensor.

3. MonitoringSession

Atributo	Tipo de Dato	Descripción
SessionId	GUID	Identificador único de la sesión de monitoreo.
PatientId	GUID	Paciente monitoreado en esta sesión.
StartTime	datetime	Inicio de la sesión.
EndTime	datetime?	Fin de la sesión (opcional mientras esté activa).

4.2.3.1. Domain Layer

En la capa de dominio se modelan los conceptos centrales del contexto **Sensor Monitoring** siguiendo los principios de **Domain-Driven Design**, adaptados a **SafeMed**.

El agregado raíz **MonitoringSession** representa una sesión activa o histórica de monitoreo del paciente. Cada **MonitoringSession** está asociada a múltiples **HeartRateReading**, que representan las mediciones capturadas.

El **SensorDevice** se modela como una entidad que establece el vínculo físico entre el paciente y el sistema.

Aggregates

Aggregate MonitoringSession

- **SessionId** (GUID)
- **PatientId** (GUID)
- **StartTime** (datetime)
- **EndTime** (datetime?)
- **List readings**

Métodos:

- **StartSession(patientId): MonitoringSession**
- **EndSession(): void**
- **AddHeartRateReading(heartRateBPM, timestamp): void**

- `GetAverageHeartRate(): double`

Entities

Entity `HeartRateReading`

- **ReadingId** (GUID)
- **PatientId** (GUID)
- **Timestamp** (datetime)
- **HeartRateBPM** (int)

Entity `SensorDevice`

- **DeviceId** (GUID)
- **SerialNumber** (string)
- **PatientId** (GUID)
- **IsActive** (boolean)

Value Objects

No se definen Value Objects complejos en este contexto básico, pero podría incluirse más adelante uno para medidas de señales biométricas.

Domain Services

`RealTimeMonitoringService`

- `ProcessSensorData(sensorInput): HeartRateReading`
- `DetectAnomalies(heartRateReading): AnomalyAlert?`

4.2.3.2. Interface Layer

La **Capa de Interfaz** para el contexto de **Sensor Monitoring** en SafeMed sirve como la puerta de entrada para recibir las lecturas del sensor y mostrar los datos en tiempo real a los usuarios.

Se encarga de:

- Recibir las lecturas de los sensores.
- Validar datos de entrada.
- Disparar casos de uso en la Capa de Aplicación.
- Formatear y enviar respuestas.

Controladores

Controlador	Responsabilidad Principal
<code>SensorDataController</code>	Recibir y registrar datos de ritmo cardíaco.
<code>MonitoringSessionController</code>	Gestionar sesiones de monitoreo de pacientes.

Endpoints Clave por Controlador

Controlador	Método HTTP	Endpoint	Descripción
-------------	----------------	----------	-------------

Controlador	Método HTTP	Endpoint	Descripción
SensorDataController	POST	/api/v1/sensors/data	Recibir nueva lectura de ritmo cardíaco.
MonitoringSessionController	POST	/api/v1/sensors/sessions/start	Iniciar una nueva sesión de monitoreo.
MonitoringSessionController	POST	/api/v1/sensors/sessions/end	Finalizar una sesión de monitoreo activa.
MonitoringSessionController	GET	/api/v1/sensors/sessions/{sessionId}	Obtener detalles de una sesión de monitoreo.
MonitoringSessionController	GET	/api/v1/sensors/sessions/{sessionId}/readings	Obtener lecturas asociadas a una sesión.

Enfoque

Esta capa se centra en la comunicación externa (sensores, apps móviles, web), validando la estructura de los datos y orquestando la lógica de aplicación.

4.2.3.3. Application Layer

La **Capa de Aplicación** coordina el flujo de trabajo entre las capas de Interfaz, Dominio e Infraestructura para los casos de uso de monitoreo de ritmo cardíaco.

Comandos (Commands)

Comando	Descripción
StartMonitoringSessionCommand	Iniciar una nueva sesión de monitoreo para un paciente.
EndMonitoringSessionCommand	Finalizar una sesión de monitoreo activa.
RegisterHeartRateReadingCommand	Registrar una nueva lectura de ritmo cardíaco.

Queries

Query	Descripción
GetMonitoringSessionQuery	Obtener los detalles de una sesión específica.
GetHeartRateReadingsQuery	Obtener todas las lecturas de una sesión.

Command Handlers

Command Handler	Comandos que Maneja
StartMonitoringSessionCommandHandler	StartMonitoringSessionCommand
EndMonitoringSessionCommandHandler	EndMonitoringSessionCommand
RegisterHeartRateReadingCommandHandler	RegisterHeartRateReadingCommand

Query Handlers

Query Handler	Queries que Maneja
GetMonitoringSessionQueryHandler	GetMonitoringSessionQuery
GetHeartRateReadingsQueryHandler	GetHeartRateReadingsQuery

Otros Componentes

Se podrían incluir Event Handlers para alertas de anomalías detectadas durante el monitoreo en tiempo real.

4.2.3.4. Infrastructure Layer

La **Capa de Infraestructura** maneja la persistencia de datos de sesiones y lecturas de ritmo cardíaco provenientes de los sensores.

Repositorios Clave

Infrastructure Repository **MonitoringSessionRepository**

- **Implementa:** `IMonitoringSessionRepository`
- **Tecnología:** ORM (ej: Entity Framework Core)

Métodos:

- `StartSession(patientId: GUID): MonitoringSession`
- `EndSession(sessionId: GUID): void`
- `FindById(sessionId: GUID): Optional<MonitoringSession>`

Infrastructure Repository **HeartRateReadingRepository**

- **Implementa:** `IHeartRateReadingRepository`
- **Tecnología:** ORM (ej: Entity Framework Core)

Métodos:

- `AddReading(reading: HeartRateReading): void`
- `FindReadingsBySession(sessionId: GUID): List<HeartRateReading>`

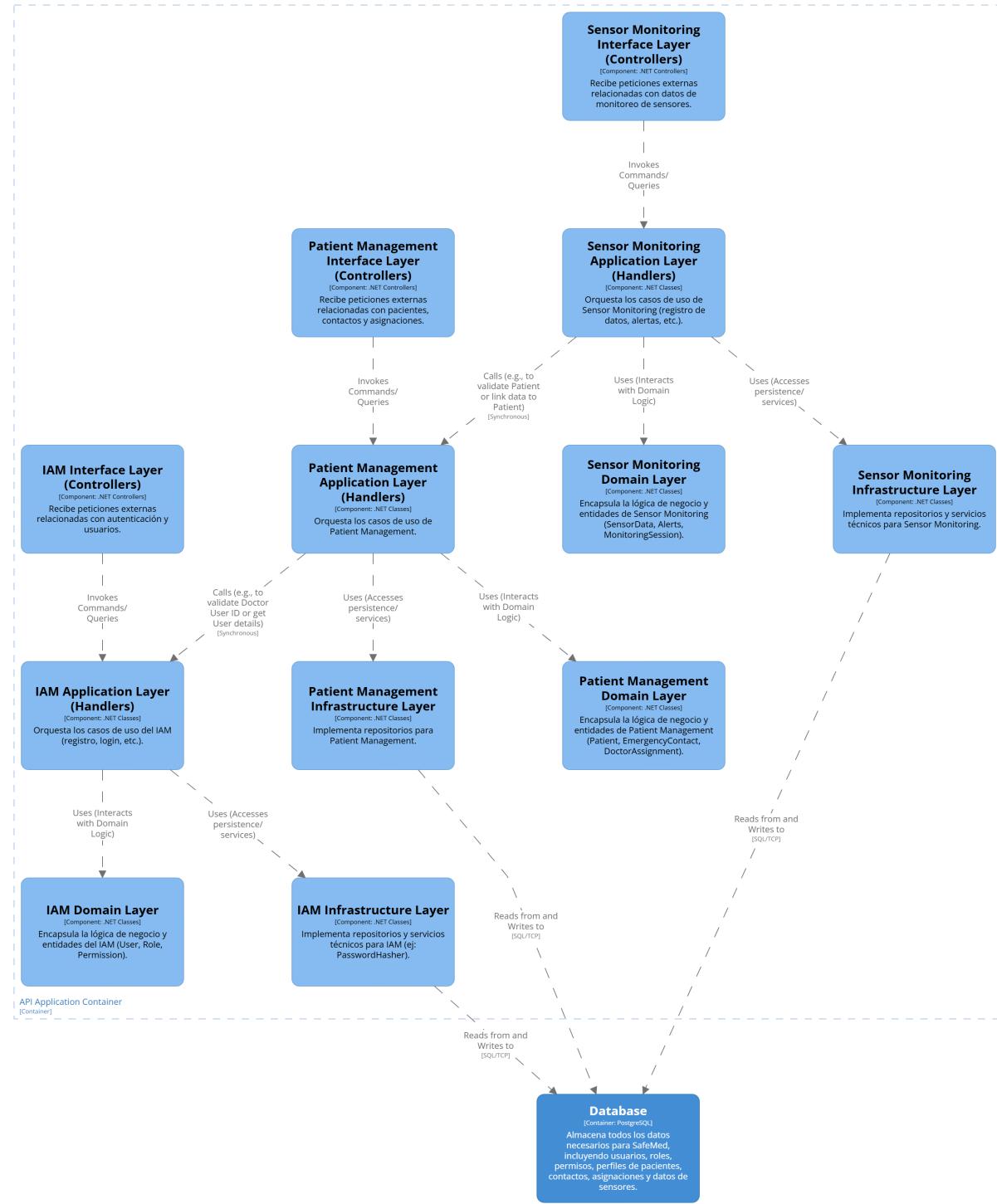
Infrastructure Repository SensorDeviceRepository

- **Implementa:** `ISensorDeviceRepository`
- **Tecnología:** ORM (ej: Entity Framework Core)

Métodos:

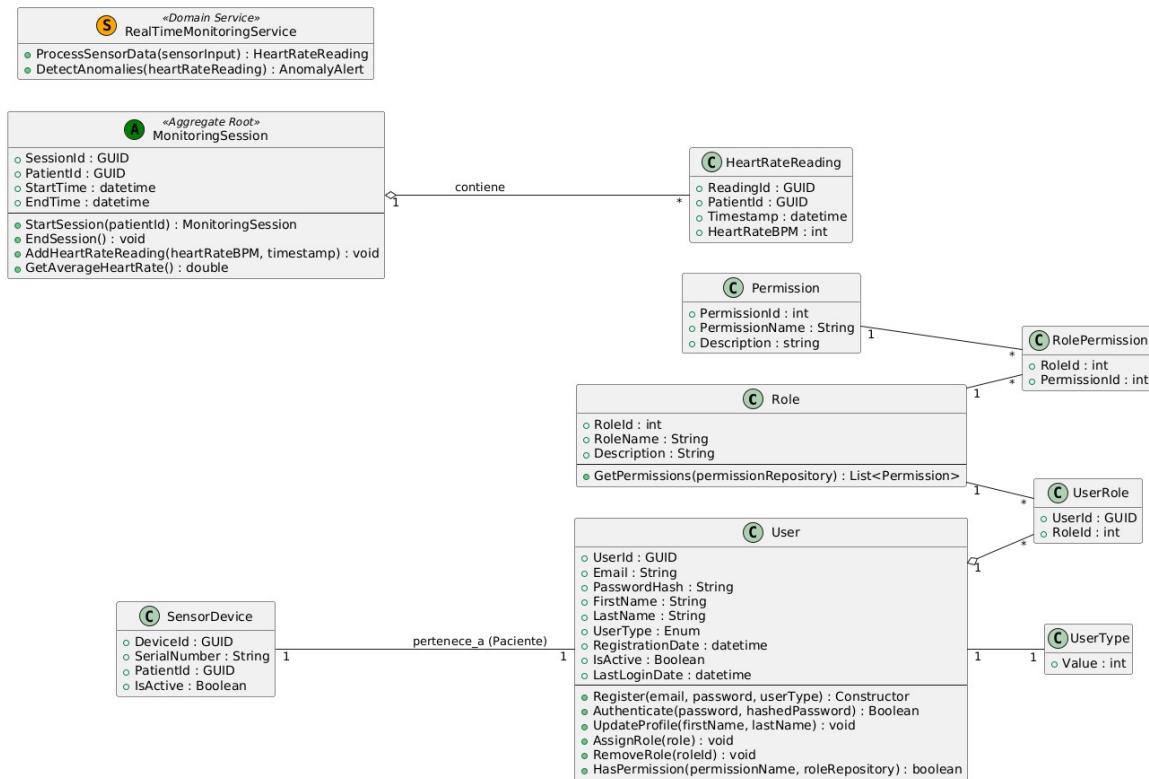
- `FindBySerialNumber(serialNumber: string): Optional<SensorDevice>`
- `ActivateSensor(deviceId: GUID): void`
- `DeactivateSensor(deviceId: GUID): void`

4.2.3.6. Bounded Context Software Architecture Code Level Diagrams.

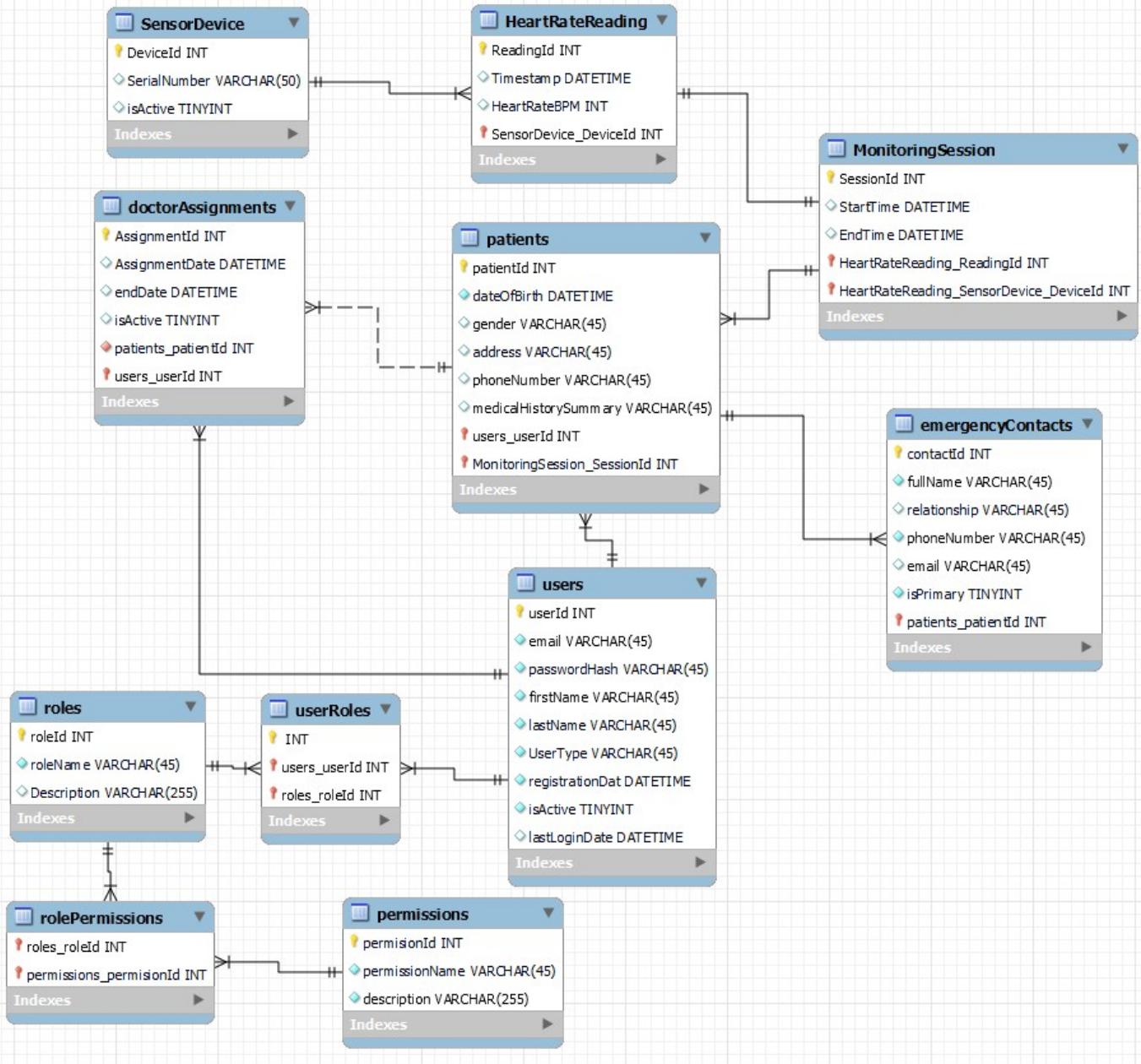


[Component] SafeMed - API Application Container
 Diagrama de Componentes del Contenedor API (incluye IAM, Patient Management y Sensor Monitoring).
 Saturday, April 26, 2025 at 4:40 PM Peru Standard Time

4.2.3.6.1. Bounded Context Domain Layer Class Diagrams.



4.2.3.6.2. Bounded Context Database Design Diagram.



Capítulo V: Solution UI/UX Design

5.1. Style Guidelines.

5.1.1. General Style Guidelines.

Logotipo

El logotipo actual de MedSystem es un isologotipo tipográfico. La palabra MED aparece en un azul claro que evoca confianza y salud, mientras que SYSTEM se muestra en negro para reforzar seriedad y fiabilidad. La línea inferior horizontal remata el conjunto y hace de ancla visual. Aunque en el prototipo original existía un ícono de cruz, en la versión vigente se ha optado por esta marca denominativa para simplificar la lectura y mejorar la escalabilidad en distintos dispositivos.

MEDSYSTEM

Tipografía

- **Primaria – Poppins:** empleada en títulos, subtítulos y cuerpo de texto por su excelente legibilidad y estilo contemporáneo.
- **Secundaria – Oswald:** se utiliza en subtítulos o cifras destacadas para aportar contraste.
- **Respaldo – sans-serif genérico:** se declara como *fallback* por si la carga de las familias anteriores falla.

FONTS

Poppins Light AaBbCcDdEeFfGgHhIiJj

Poppins Light | 997 Glyphs

Poppins AaBbCcDdEeFfGgHhIiJjKkLl

Poppins Regular | 997 Glyphs

Poppins Medium AaBbCcDdEeFfGg

Poppins Medium | 997 Glyphs

Poppins SemiBold AaBbCcDdEeFfGg

Poppins SemiBold | 997 Glyphs

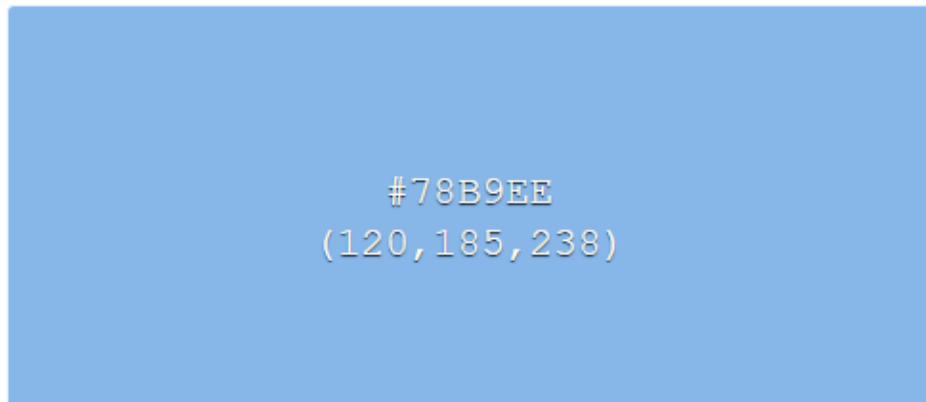
Poppins Bold AaBbCcDdEeFfGgHh

Poppins Bold | 997 Glyphs

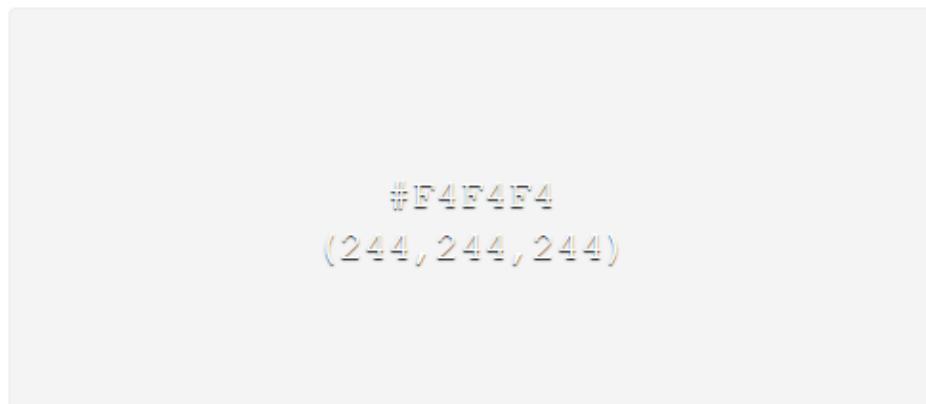
Oswald DemiBold
 Oswald DemiBold
Oswald DemiBold
Oswald DemiBold
Oswald DemiBold
Oswald DemiBold
Oswald DemiBold
Oswald DemiBold

Paleta de Colores

Uso específico	Color	Hex
Títulos destacados, palabras clave (ej. "MED")	Azul principal	#103EF5
Fondo de botones primarios, gradientes oscuros	Azul oscuro	#103EF5
Degradiados de fondo, botones secundarios	Azul claro	#78B9EE
Fondos de secciones, tarjetas, áreas de contenido	Blanco	#FFFFFF
Texto general, encabezados secundarios	Negro	#000000



#78B9EE
(120, 185, 238)



#F4F4F4
(244, 244, 244)



#103EF5
(16, 62, 245)

5.1.2. Web, Mobile and IoT Style Guidelines.

Se prioriza una experiencia limpia y enfocada. Los azules transmiten calma y profesionalismo; el blanco aporta amplitud y facilita la lectura. Los tamaños de fuente se ajustan para garantizar accesibilidad. Todo el sistema visual mantiene consistencia entre la *landing* y la aplicación web.

Botones

- **Primario:** relleno con gradiente horizontal (#003A64 → #60D4FF), texto blanco, bordes redondeados grandes.
- **Secundario:** fondo blanco, borde 1 px con el mismo gradiente y texto en azul principal.

Colores

#78B9EE
(120, 185, 238)

#F4F4F4
(244, 244, 244)

#103EF5
(16, 62, 245)

Estilo de letras

FONTS

Poppins Light AaBbCcDdEeFfGgHhIiJj

Poppins Light | 997 Glyphs

Poppins AaBbCcDdEeFfGgHhIiJjKkLl

Poppins Regular | 997 Glyphs

Poppins Medium AaBbCcDdEeFfGg

Poppins Medium | 997 Glyphs

Poppins SemiBold AaBbCcDdEeFfGg

Poppins SemiBold | 997 Glyphs

Poppins Bold AaBbCcDdEeFfGgHh

Poppins Bold | 997 Glyphs

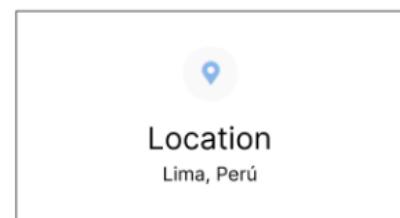
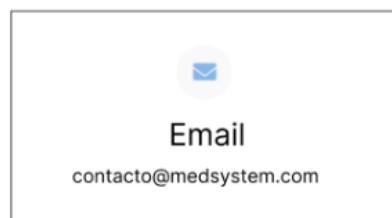
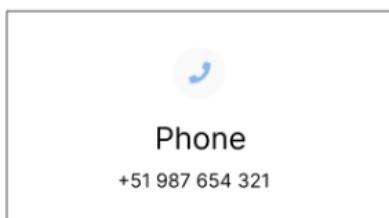
» Oswald DemiBold
» Oswald DemiBold
» **Oswald DemiBold**
» **Oswald DemiBold**
» **Oswald DemiBold**
Oswald DemiBold
Oswald DemiB

Botones

Navbar



Footer

[CONTACT](#)

Quick Links <ul style="list-style-type: none"> → Home → About us → Product → Team → Contact 	Product <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Make appointments <input checked="" type="checkbox"/> View Medical Records <input checked="" type="checkbox"/> Receive Analyses 	Follow Us <ul style="list-style-type: none"> Facebook GitHub Twitter 	Contact <ul style="list-style-type: none"> Email Phone Location
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------

5.2. Information Architecture.

5.2.1. Organization Systems.

Se utiliza una estructura jerárquica para la *landing page*: Inicio → Secciones temáticas (About Us, Product, Pricing, Team, Contact). Dentro de la aplicación, la navegación se divide por roles (pacientes, médicos, laboratorio) y se accede mediante un menú lateral.

5.2.2. Labeling Systems.

- **Appointments** – programación y listado de citas.
- **Chat** – mensajería médico-paciente con finalidad exclusiva de orientación clínica.
- **Treatments for patient** – seguimiento y edición de tratamientos prescritos.
- **Request history** – historial clínico y tratamientos previos del paciente.

5.2.3. SEO Tags and Meta Tags

Contexto	Title	Description	Keywords	Author
Landing	MedSystem	MedSystem – SafeMed Official Landing Page	appointments, treatments, doctors, patients, laboratories	SafeMed Team
Web app	MedSystem	MedSystem – SafeMed Official Web Site	appointments, clinic history, treatment tracking, medical tests, test results, registration, diagnosis	SafeMed Team

5.2.4. Searching Systems.

La aplicación integra un motor de búsqueda optimizado que filtra rápidamente pacientes, chats, historiales y resultados directamente desde la base de datos, incluso en escenarios de gran volumen.

5.2.5. Navigation Systems.

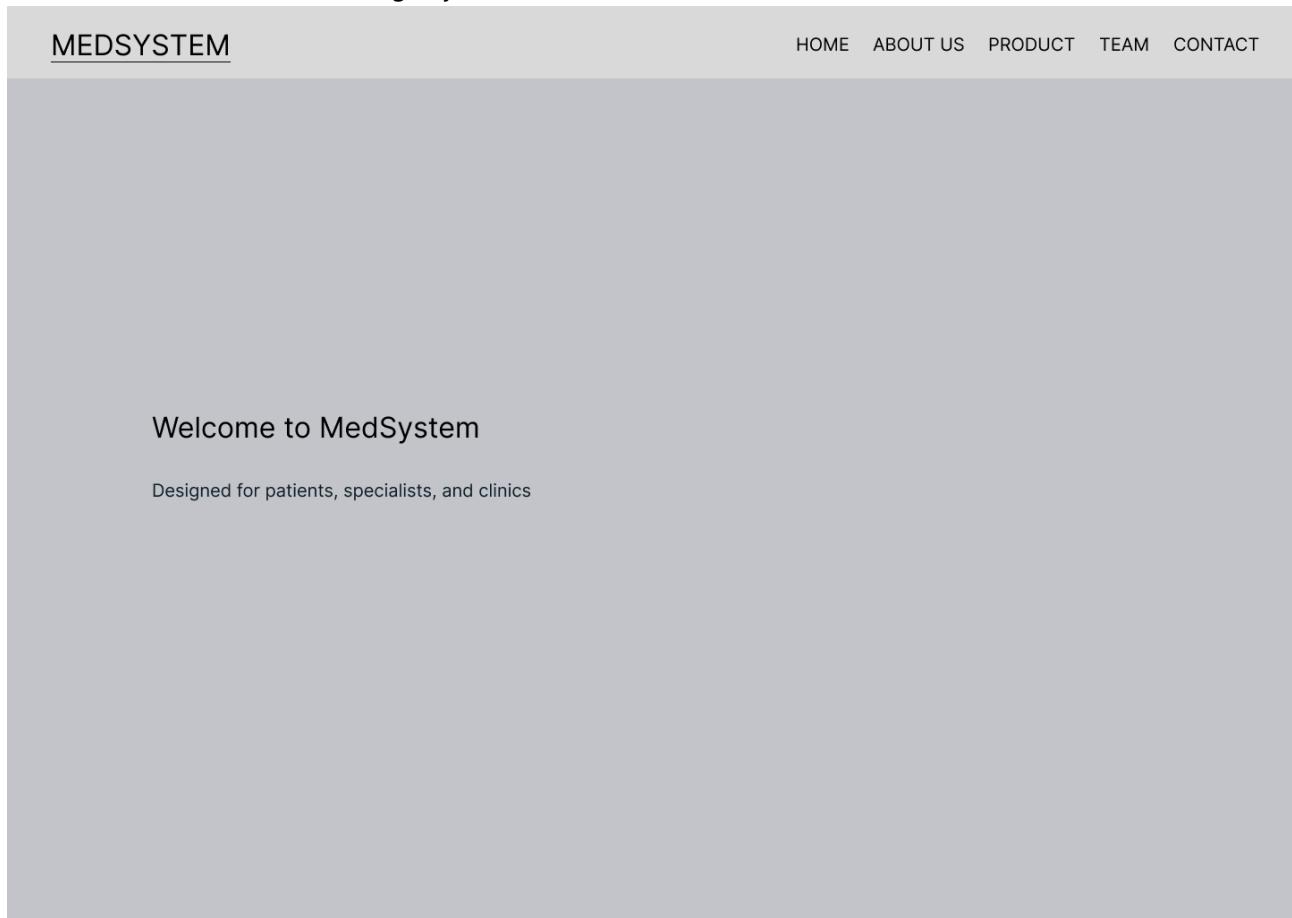
- **Landing:** barra superior fija con anclas internas a cada sección.
- **Aplicación:** barra lateral con iconos y rótulos; se adapta responsivamente a *drawer* en móvil.

5.3. Landing Page UI Design.

5.3.1. Landing Page Wireframe.

El diseño final mantiene la lógica de los wireframes originales, pero introduce una imagen de héroe de ancho completo con superposición oscura para mejorar el contraste del eslogan. Las secciones se alternan entre texto-izquierda/imagen-derecha y viceversa para conservar ritmo visual.

1. **Hero** – título de bienvenida, eslogan y *call to action*.



2. About Us – descripción de la empresa y valores (compromiso, trabajo en equipo, organización).

The screenshot shows the 'ABOUT US' page of the MedSystem website. At the top, there is a navigation bar with links for HOME, ABOUT US, PRODUCT, TEAM, and CONTACT. Below the navigation bar, the word 'ABOUT US' is centered in a large, bold, black-bordered box. To the left, under the heading 'SafeMed', there is a brief description of the company: 'SafeMed is an innovative startup founded by Software Engineering students from the Faculty of Engineering at the Peruvian University of Applied Sciences. Our company focuses on developing a web solution, called MedSystem, using modern technologies to optimize processes in orthopedic clinics. MedSystem is designed to meet the needs of both orthopedic specialists and patients.' Below this text are three small boxes with icons and labels: 'Commitment' (a person icon), 'Teamwork' (a group of people icon), and 'Organization' (an office building icon). To the right, there is a large, stylized icon of a person's head and shoulders inside a square frame.

3. Product – beneficios segmentados: pacientes y médicos, con ilustraciones.

The screenshot shows the 'Patients' section of the MedSystem website. At the top, there is a navigation bar with links for HOME, ABOUT US, PRODUCT, TEAM, and CONTACT. Below the navigation bar, there is a large, stylized icon of a person's head and shoulders inside a square frame. To the right, the word 'Patients' is displayed in bold. Below 'Patients', there is a paragraph of text: 'MedSystem is an intuitive and accessible platform that provides you with the peace of mind to access your medical records securely and conveniently. With our application, you can schedule appointments, receive reminders, and access relevant information about your treatment, all in one place, so you can focus on your well-being without hassle.'

MEDSYSTEM

HOME ABOUT US PRODUCT TEAM CONTACT

Doctors / Specialists

MedSystem is your ally in efficiently managing your medical practice. With our platform, you can easily organize your schedule, quickly access patients' medical history, and collaborate effectively with other specialists. Simplify your daily work and provide quality care with MedSystem.

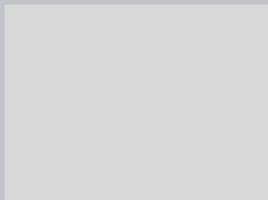
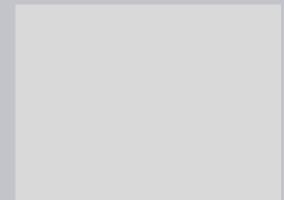
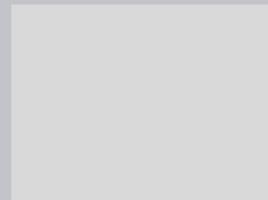
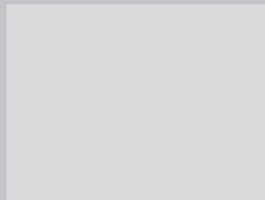


4. **Features** – tarjetas con íconos y breve texto explicativo; separadas por receptor (Patients / Doctors).

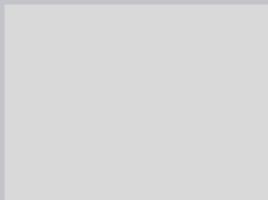
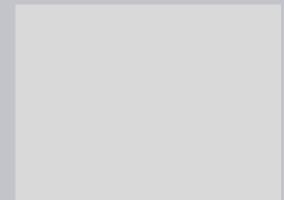
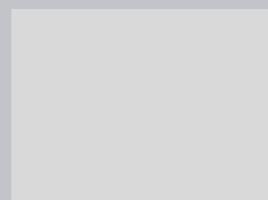
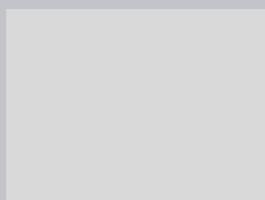
MEDSYSTEM

HOME ABOUT US PRODUCT TEAM CONTACT

FOR PATIENTS



FOR DOCTORS



5. **Team** – galería adaptativa con foto y rol de cada integrante.

The screenshot shows a light gray header with the word "MEDSYSTEM" in bold. Below it is a large gray rectangular area. In the center of this area is a white-bordered box containing the text "OUR TEAM". Below this box are five smaller white-bordered boxes, each containing a black silhouette of a person's head and shoulders. To the right of each silhouette is a small gray circle with a number: "1", "2", "3", "4", and "5".

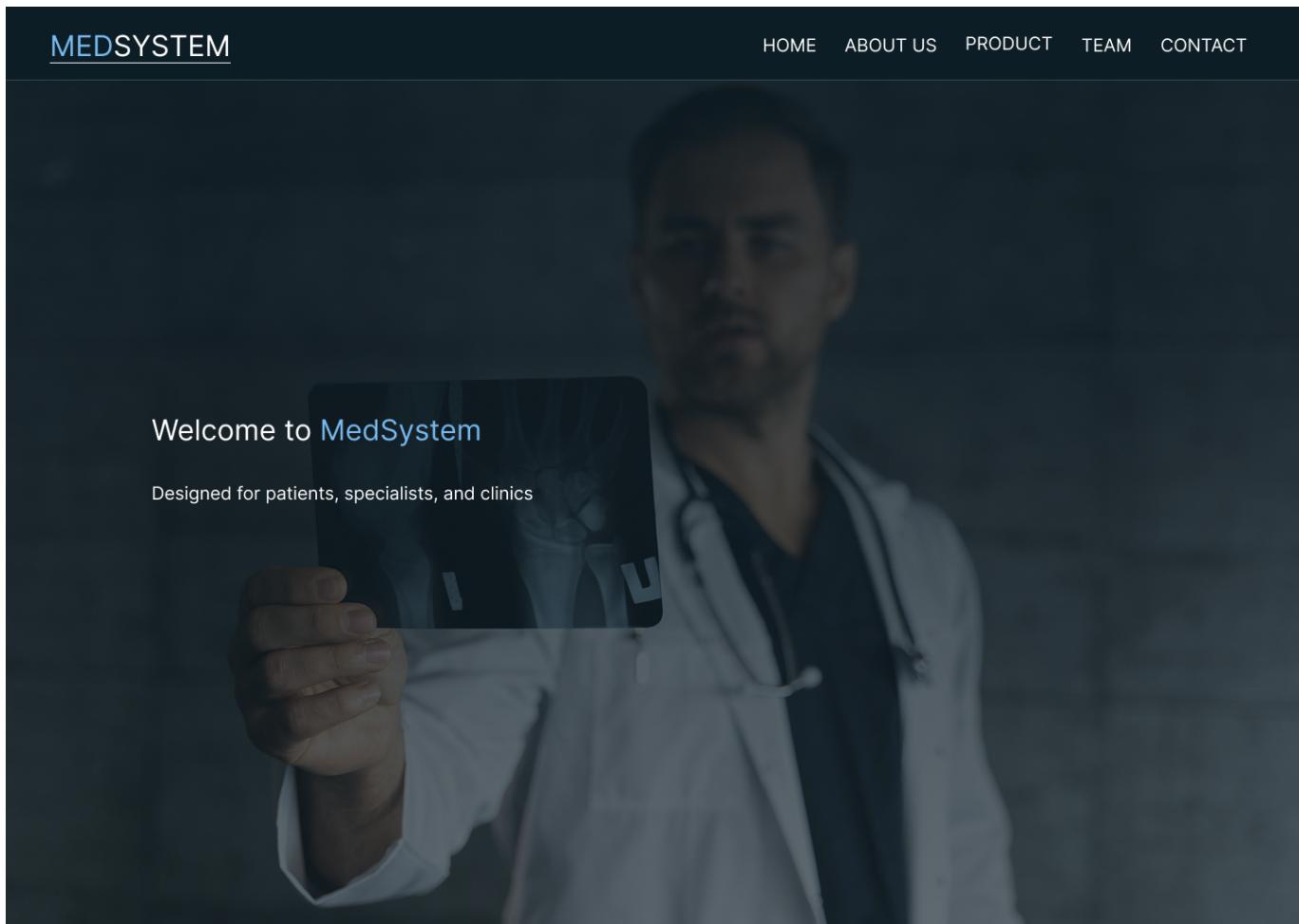
6. **Contact** – información de teléfono, correo y ubicación; seguido de footer con enlaces rápidos y redes sociales.

The screenshot shows a light gray header with the word "MEDSYSTEM" in bold. Below it is a large gray rectangular area. In the center of this area is a white-bordered box containing the text "CONTACT". Below this box are three separate white-bordered boxes, each containing an icon of a camera and the word "Number", "Email", or "Location" followed by a link. The "Number" box has the link "+51 987 654 321". The "Email" box has the link "contacto@medsystem.com". The "Location" box has the link "Lima, Perú".

Footer Links:

- Home
- About us
- Product
- Team
- Contact
- Make appointments
- View Medical Records
- Receive Analyses
- Facebook
- Github
- Twitter
- Email
- Phone
- Location

5.3.2. Landing Page Mock-up.



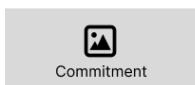
MEDSYSTEM

HOME ABOUT US PRODUCT TEAM CONTACT

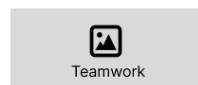
ABOUT US

SafeMed

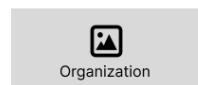
SafeMed is an innovative startup founded by Software Engineering students from the Faculty of Engineering at the Peruvian University of Applied Sciences. Our company focuses on developing a web solution, called MedSystem, using modern technologies to optimize processes in orthopedic clinics. MedSystem is designed to meet the needs of both orthopedic specialists and patients.



Commitment



Teamwork



Organization





Patients

MedSystem is an intuitive and accessible platform that provides you with the peace of mind to access your medical records securely and conveniently. With our application, you can schedule appointments, receive reminders, and access relevant information about your treatment, all in one place, so you can focus on your well-being without hassle.

Doctors / Specialists

MedSystem is your ally in efficiently managing your medical practice. With our platform, you can easily organize your schedule, quickly access patients' medical history, and collaborate effectively with other specialists. Simplify your daily work and provide quality care with MedSystem.



MEDSYSTEM

HOME ABOUT US PRODUCT TEAM CONTACT

FOR PATIENTS



Make Appointments

Schedule appointments your way, quickly and securely



Track Treatments

Follow your treatments step by step



Schedule Medical Tests

You can schedule medical tests according to your doctor's instructions



Receive Results

Receive your test results, virtually or in person

FOR DOCTORS



Revisar opciones de tratamiento

Te brinda opciones de tratamiento, de acuerdo a diagnósticos pasados



Evaluuar progreso del paciente

Visualiza como va evolucionando el paciente



Interpretar resultados de análisis

Muestra información sintetizada de un análisis



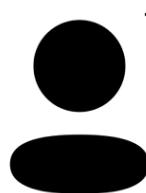
Comunicación segura con pacientes

Chat dentro de la aplicación, para una comunicación directa

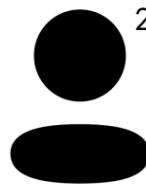
MEDSYSTEM

HOME ABOUT US PRODUCT TEAM CONTACT

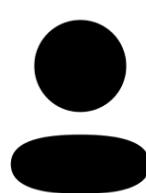
OUR TEAM



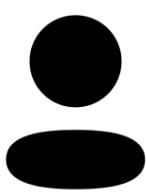
1



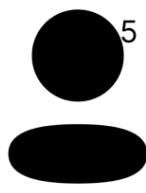
2



3



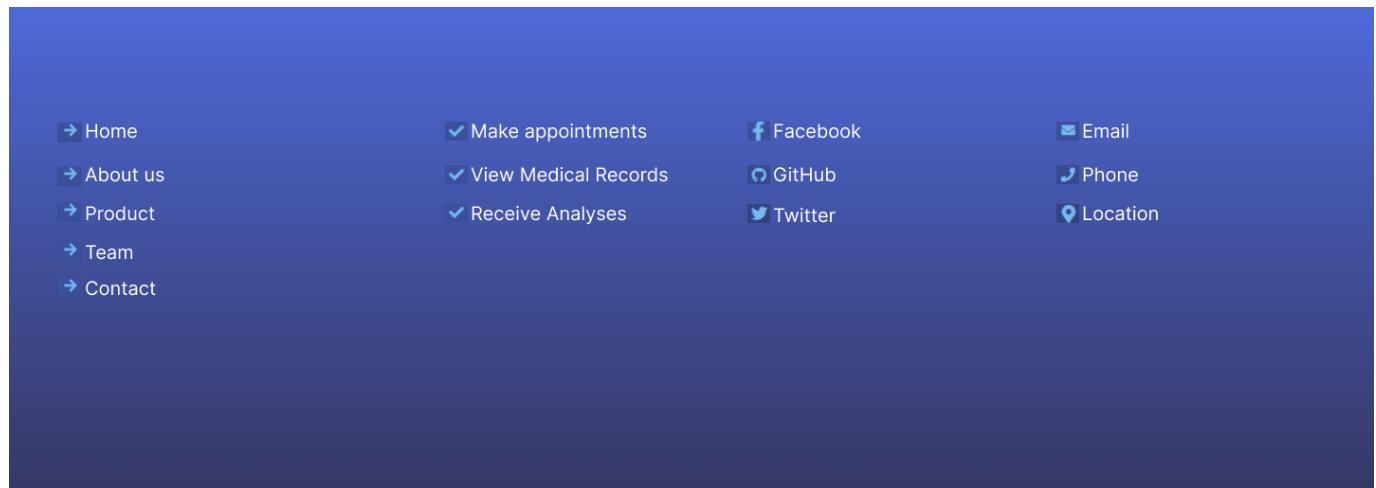
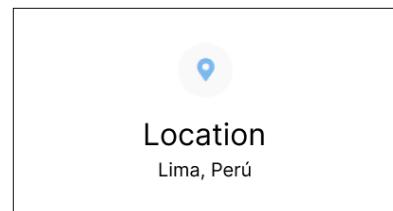
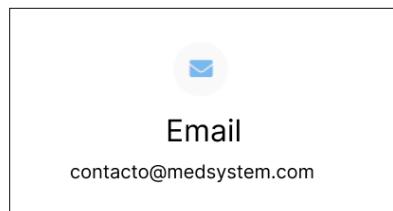
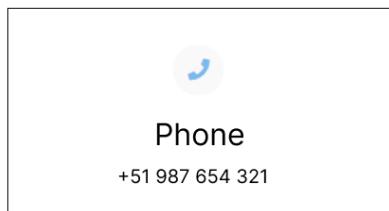
4



5

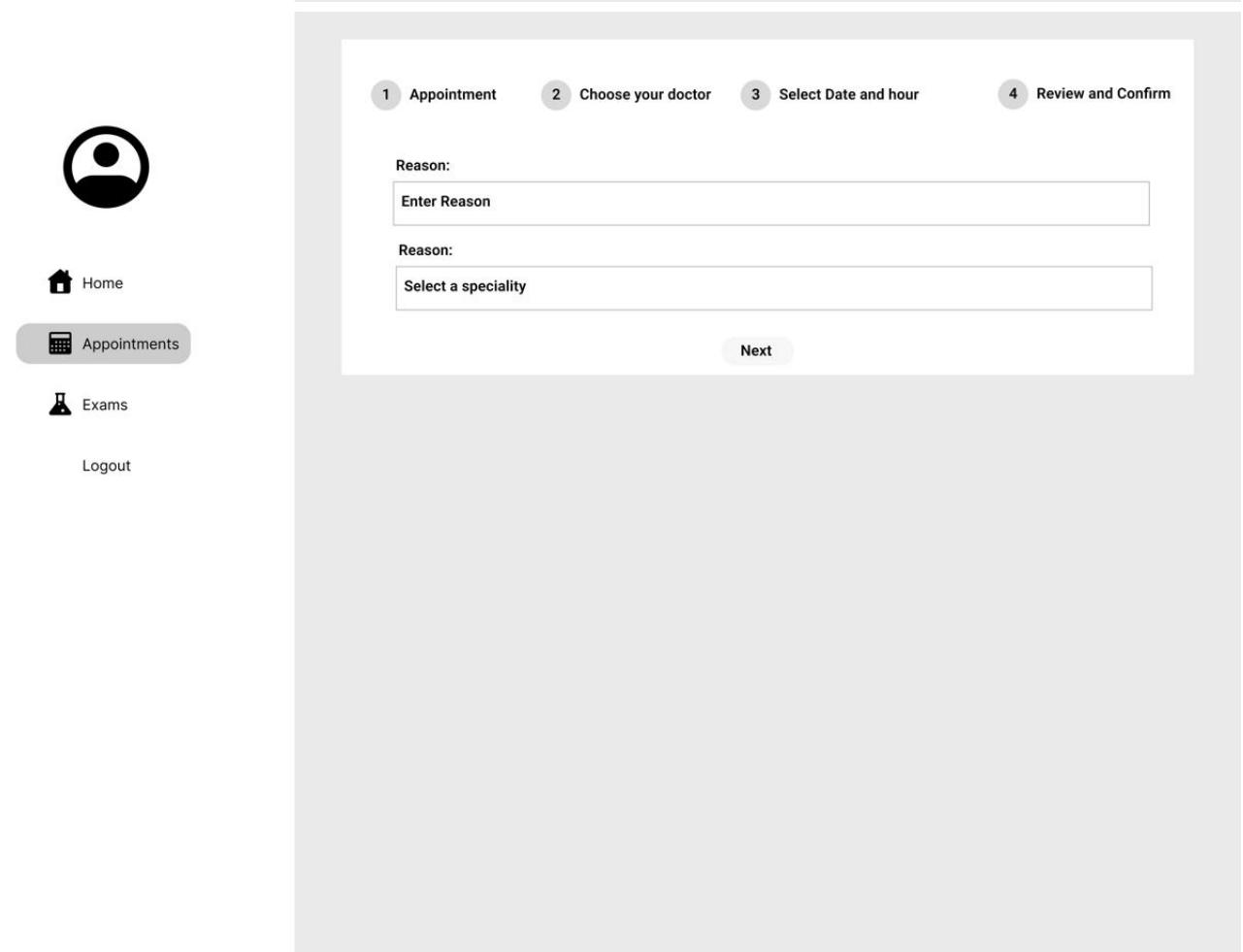
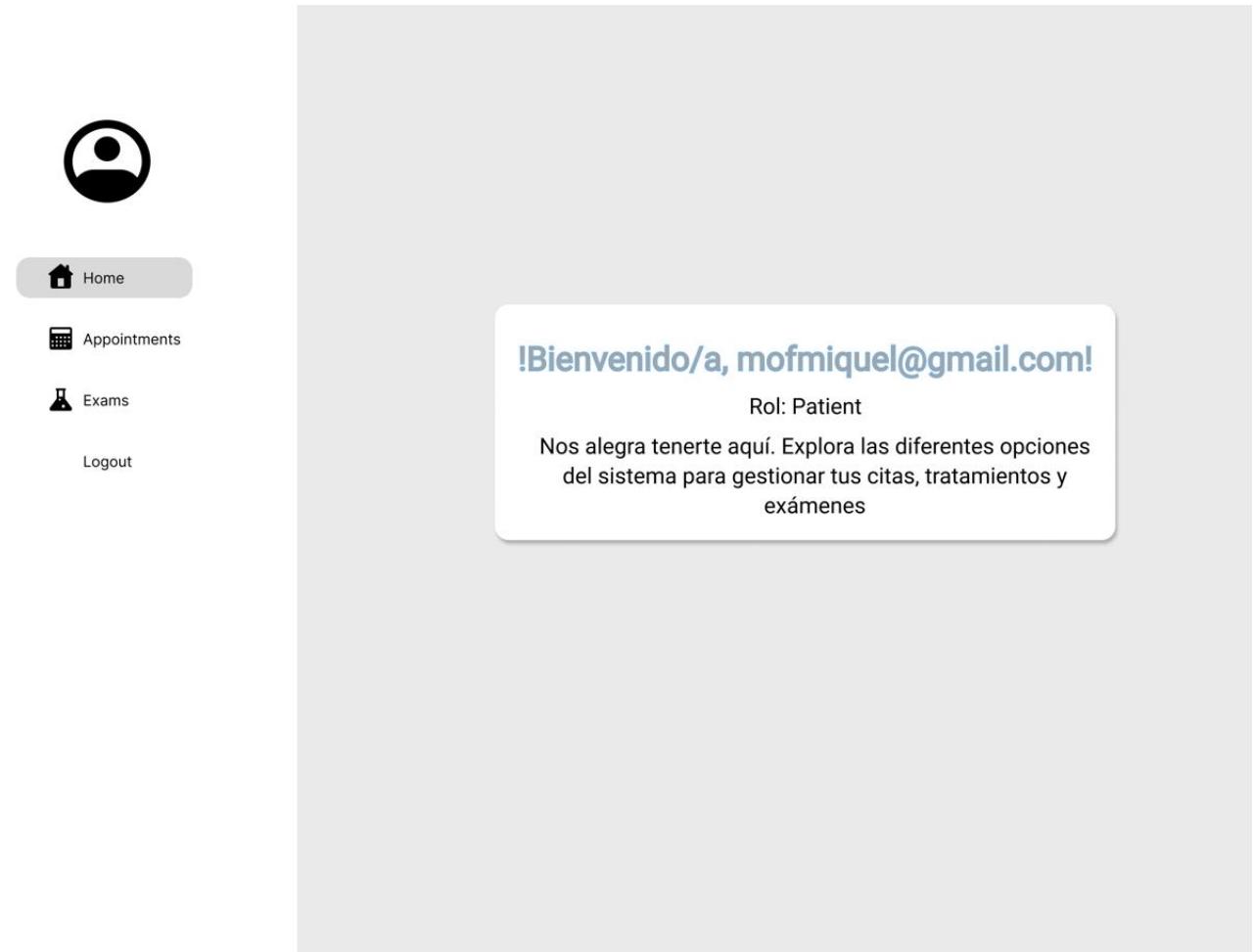
MEDSYSTEM

CONTACT



5.4. Applications UX/UI Design.

5.4.1. Applications Wireframes.



1 Appointment 2 Choose your doctor 3 Select Date and hour 4 Review and Confirm

Choose a doctor:

Select a doctor:

Home Appointments Exams Logout

1 Appointment 2 Choose your doctor 3 Select Date and hour 4 Review and Confirm

Reason: Heart
Specialty: Cardiology
Doctor:
Date: May 6, 2025
Time: 00:00

Back Submit

5.4.2. Applications Mock-ups.

Sign In

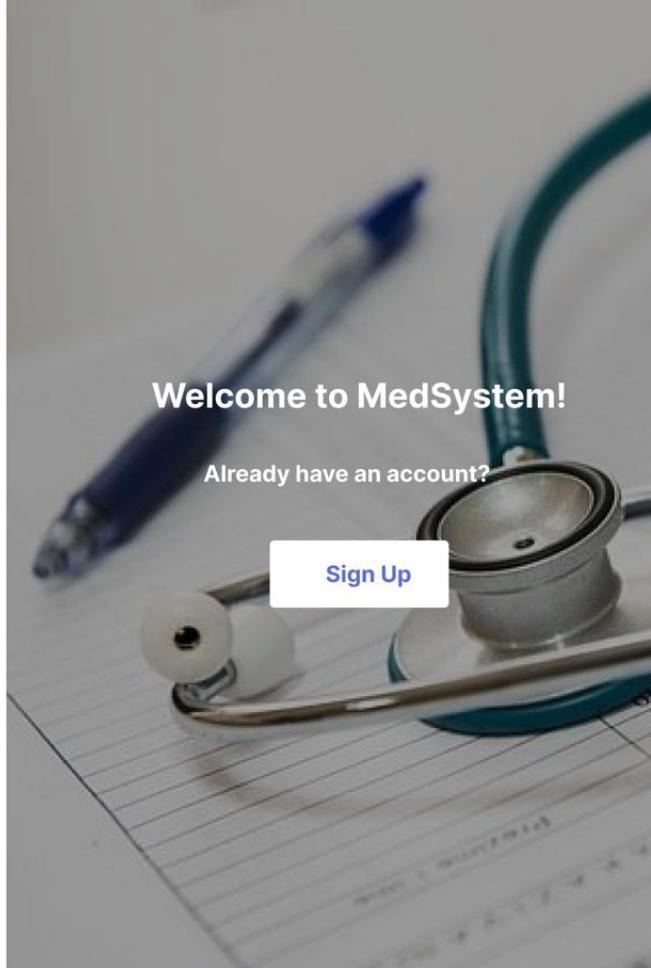
Email

Password

Remember Me

[Forgot Password?](#)

[Sign In](#)



Welcome to MedSystem!

[Already have an account?](#)

[Sign Up](#)

Sign Up

[Doctor](#)

[Patient](#)

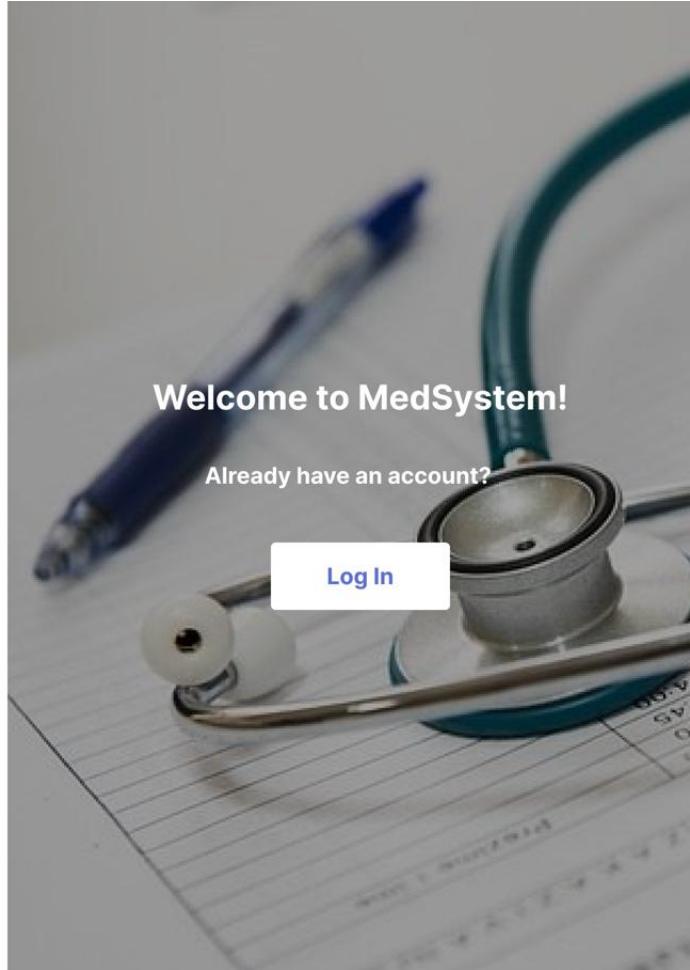
Email

Password

First Name

Last Name

[Sign Up](#)



Welcome to MedSystem!

[Already have an account?](#)

[Log In](#)

Sign Up

[Doctor](#)[Patient](#)

Email

Enter your email

Password

Enter your password

First Name

Enter your first name

Last Name

Enter your last name

License Number

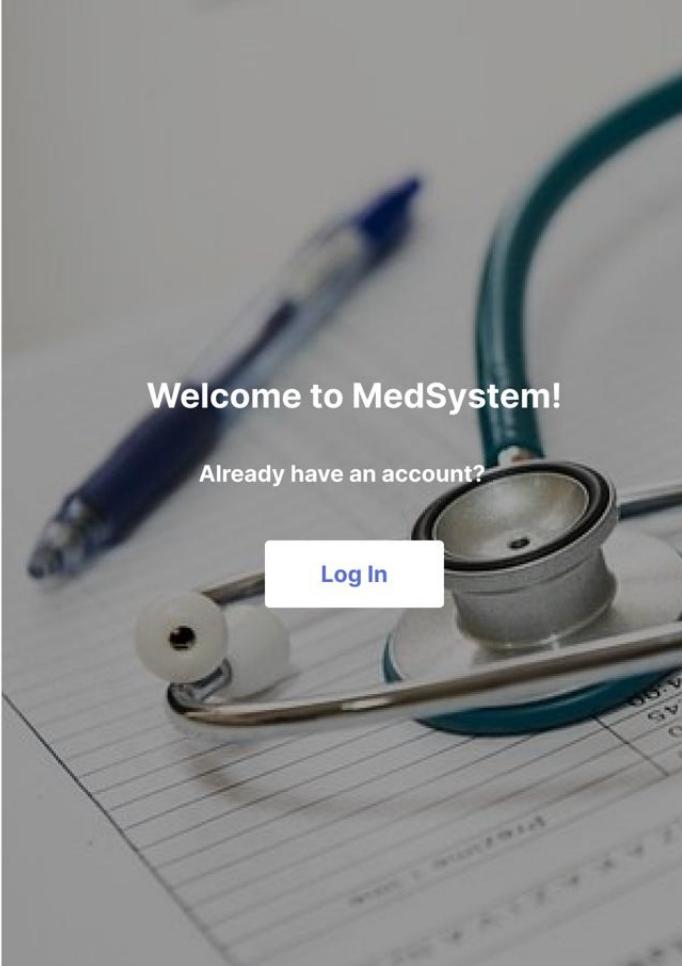
Enter your license number

Speciality

Enter your license speciality

Phone Number

Enter your license phone number

[Sign Up](#)A photograph of a stethoscope with a blue tube and a pen resting on a white, lined clipboard.

Already have an account?

[Log In](#)

Sign Up

[Doctor](#)[Patient](#)

Email

Enter your email

Password

Enter your password

First Name

Enter your first name

Last Name

Enter your last name

Age

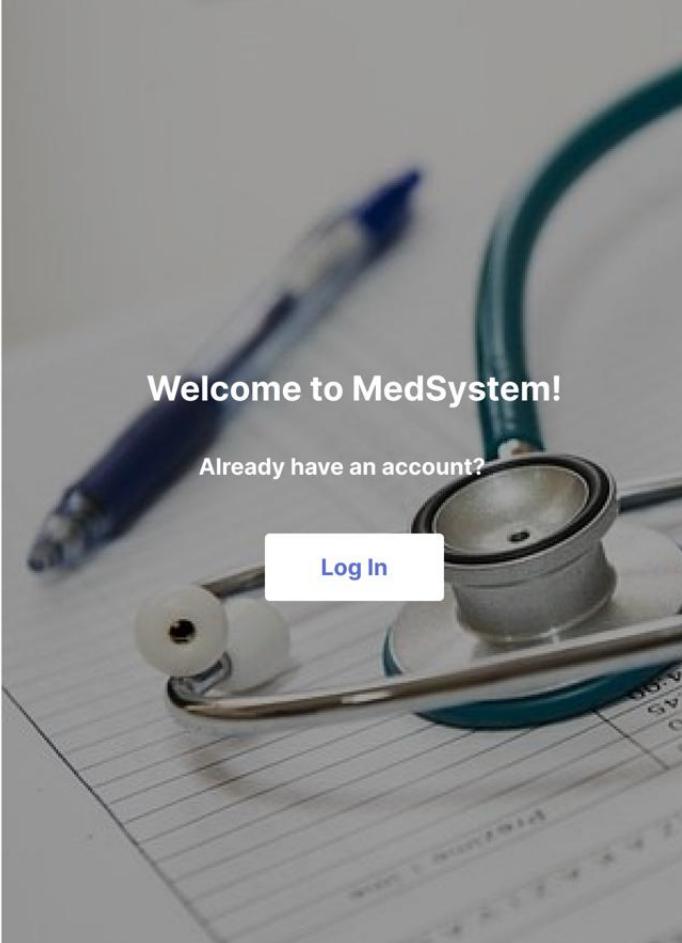
Enter your license age

Address

Enter your license address

Phone Number

Enter your license phone number

[Sign Up](#)A photograph of a stethoscope with a blue tube and a pen resting on a white, lined clipboard.

Already have an account?

[Log In](#)

The screenshot shows the application's main interface. On the left, there is a sidebar with a user profile icon and the email address "mofmiquel@gmail.com". Below the profile are five navigation links: "Home" (selected), "Appointments", "Treatments", "Exams", and "Logout". The main content area features a welcome message: "¡Bienvenido/a, mofmiquel@gmail.com!" followed by "Rol: Patient". A brief message below states: "Nos alegra tenerte aquí. Explora las diferentes opciones del sistema para gestionar tus citas, tratamientos y exámenes.".

This screenshot shows the "Your Appointments" page. At the top, it displays the title "Your Appointments". Below the title is a table header with columns: "Doctor", "Patient", "Date", "Reason", "Speciality", and "Actions". Underneath the table, there is a pagination control showing "Items per page: 5" and "0 of 0" results. A green button labeled "Add New Appointment" is located at the bottom of the table area.

5.5. Applications Prototyping.

Capítulo VI: Product Implementation, Validation & Deployment

6.1. Software Configuration Management.

6.1.1. Software Development Environment Configuration.

En esta sección se describen brevemente todas las herramientas que se han utilizado para la elaboración del proyecto, se dividirán en dos partes.

UX/UI Design:

- Miro: Miro es una herramienta en línea colaborativa en tiempo real, esta permite crear diversos diagramas, mapas mentales, diagramas de flujo u otros artefactos que se necesiten en el proyecto. Link: <https://miro.com/es/>
- UXPressia: UXPressia es una herramienta enfocada en lo que es la experiencia de usuario, ya que esta permite crear User Personas, Customer Journey, Impact maps, etc. Link: <https://uxpressia.com/>

- Figma: Figma es una herramienta que se enfoca en el diseño de interfaces para los usuarios y prototipado de las páginas webs, además de que este también es colaborativo por lo cual todos pueden trabajar en un mismo proyecto en simultáneo. Link: <https://miro.com/es/>
- Structurizr: Es una página web donde nos permite desarrollar modelos en C4, de la arquitectura de nuestro proyecto describiendo los contenedores, componentes y contextos de nuestra solución web. Link: <https://structurizr.com/>

Software Development:

- Github: Herramienta para guardar, a través de repositorios, los proyecto que se desarollen, además que permite guardar un historial de versiones, utilizando Git Link: <https://github.com>
- Github Pages: Herramienta dentro de github que permite alojar una pagina web de un repositorio HTML: Un lenguaje de marcado para crear tanto la estructura y el contenido de una página web.
- JavaScript: Lenguaje de programación para agregar funcionalidad a las páginas webs
- CSS: Lenguaje utilizado para dar estilo a las páginas webs
- Visual Studio Code: Editor de código que se utiliza para realizar distintos trabajos con respecto a la programación, este soporta una gran cantidad de lenguajes, como Python, JavaScript, HTML, entre otros. Link: <https://code.visualstudio.com/>

Project Management:

Para el desarrollo del proyecto hemos usado distintas plataformas para hacer las coordinaciones de los avances, el primero fue un grupo por medio de WhatsApp donde compartimos las opiniones del trabajo y los avances realizados, después para las reuniones utilizamos Discord que nos permite hacer llamadas para una correcta coordinación del trabajo. En cuanto a los avances del trabajo utilizamos github donde hacíamos commit con cada avance significativo que hacemos en los repositorios

6.1.2. Source Code Management.

Hemos creado los repositorios que usaremos a lo largo de el desarrollo del proyecto esto incluye el de la documentacion, landing page, frontend y backend.

- URL del repositorio de documentacion: <https://github.com/SafeMedUPC/InformeSafeMed/tree/main>
- URL del repositorio de la landing page: <https://github.com/SafeMedUPC/SafeMed-LandingPage>
- URL del repositorio del Frontend:
- URL del repositorio del Backend: <https://github.com/SafeMedUPC/Medsystem-backend>

6.1.3. Source Code Style Guide & Conventions.

Para el desarrollo de nuestra solucion web emplearemos distintos lenguajes de programacion para la landing page el frontend y backend:

HTML:

- Indentación: Mantener una indentación clara para mejorar la legibilidad.
- Semántica: Utilizar elementos HTML semánticos para una estructura clara y accesible.
- Comentarios: Documentar el código HTML con comentarios claros y concisos.
- Atributos Alt: Proporcionar textos alternativos para imágenes y otros elementos.

CSS:

- Nomenclatura de Clases: UtilizaR una convención de nomenclatura consistente, como BEM.
- Separación de Intereses: Separar estilos del HTML utilizando clases y selectores específicos.
- Reutilización de Estilos: Identificar estilos comunes y reutilizarlos para mantener la coherencia.

JavaScript:

- Nomenclatura de Variables: Utilizar nombres descriptivos, significativos y breves.
- Modularidad: Dividir el código en módulos pequeños y reutilizables.
- Manejo de Errores: Implementar un manejo de errores robusto.

6.1.4. Software Deployment Configuration.

Para gestionar el despliegue de nuestra aplicación, hemos optado por utilizar GitHub y GitHub Pages.

GitHub es una plataforma para alojar repositorios de Git. Permite colaborar de manera eficiente en proyectos de software, facilitando el seguimiento de cambios, la revisión de código y la gestión de versiones.

Despues tambien esta GitHub Pages que es una función de GitHub que permite alojar sitios web estáticos directamente desde un repositorio de GitHub. Con GitHub Pages, permitiendonos alojar nuestra pagina web sin tener que pagar un servidor externo.

6.2. Landing Page, Services & Applications Implementation.

6.2.1. Sprint 1

6.2.1.1. Sprint Planning 1.

Se planeó el primer sprint a través de una reunión dentro de la plataforma Discord. Para esto, se realizó una tabla que permitió registrar toda la información discutida.

Sprint #	Sprint 1
Sprint	
Planning	
Background	
Date	2024-06-10
Time	12:30 PM
Location	Virtual
Prepared by	Gutierrez Zumaeta, Manuel Alonso

Sprint #	Sprint 1
Attendees (to planning meeting)	De La Piedra Quintanilla, Erwin Miquel Gutierrez Zumaeta, Manuel Alonso Mendez Rosales, Marco Aurelio Roca Huapaya, Orlando Arturo Ventura Chancafe, Eduardo Renato
Sprint 1 - Review Summary	Sprint 1 - Review Summary
Sprint 1 - Retrospective Summary	El equipo destacó como puntos positivos la buena organización en las tareas y el cumplimiento del cronograma. Como mejora, se mencionó la necesidad de definir con más precisión los criterios de aceptación de las historias de usuario y reforzar la documentación de los endpoints generados en este sprint.
Sprint Goal & User Stories	
Sprint 1 Goal	El objetivo principal de este sprint fue implementar la funcionalidad de agendamiento de citas médicas desde la app móvil para pacientes, y su visualización en la plataforma web por parte de los doctores. Además, se comenzó el desarrollo de la funcionalidad para el monitoreo del ritmo cardíaco en tiempo real.
Sprint 1 Velocity	18
Sum of Story Points	19

6.2.1.2. Aspect Leaders and Collaborators.

A continuación se presenta la asignación de líderes (L) y colaboradores (C) para cada aspecto del proyecto, basada en las tareas asignadas en las historias de usuario. Cada miembro del equipo es líder en el aspecto relacionado a sus responsabilidades principales y colaborador en los demás.

Team Member	GitHub Username	Interfaz	Backend	Testing	Documentación	Integración
Miquel De la Piedra	@MiquelDlp	L	C	C	C	C
Manuel Gutierrez	@ManuGZ	L	C	C	C	C
Marco Mendez	@cuak123	L	C	C	C	C
Orlando Roca	@OrlandoRoca	C	L	C	C	C
Eduardo Ventura	@Eduvent	C	L	C	C	C

Notas:

- *Interfaz:* Diseño y desarrollo de las vistas y componentes frontend.
- *Backend:* Programación y mantenimiento de endpoints y lógica del servidor.
- *Testing:* Validación y pruebas funcionales del sistema.
- *Documentación:* Elaboración de manuales, reportes y documentación técnica.
- *Integración:* Coordinación entre módulos y despliegue de la aplicación.

6.2.1.3. Sprint Backlog 1.

User Story ID	User Story Title	Task ID	Task Title	Description	Estimation(hours)	Assigned to	Status
US01	Agendar una cita desde la app	T01	Diseñar interfaz de agendamiento en app	Crear vista para que el paciente seleccione fecha, hora y motivo de la cita.	4	De La Piedra Quintanilla, Erwin Miquel	In Progress
			Desarrollar lógica de agendamiento en frontend	Implementar funciones que validen el formulario y preparen los datos para el backend.	3	Mendez Rosales, Marco Aurelio	To Do
		T03	Crear endpoint POST para agendar cita	Programar endpoint en backend para registrar la cita médica con los datos del paciente.	4	Ventura Chancafe, Eduardo Renato	In Progress
US02	Ver citas programadas	T04	Diseñar vista de citas programadas	Desarrollar pantalla donde el paciente pueda ver sus citas agendadas con detalles.	3	Gutierrez Zumaeta, Manuel Alonso	To Do
				Programar lógica de backend que retorne las citas		Roca Huapaya, Orlando Arturo	

			programadas por el usuario.		
US03	Cancelar o reprogramar una cita	T06	Desarrollar función para cancelar o reprogramar	Incluir opciones en frontend para modificar o eliminar citas.	Gutierrez Zumaeta, Manuel Alonso To Do
US04	Sincronización con agenda del doctor	T07	Crear vista de agenda para doctor	Diseñar sección donde el doctor puede ver sus citas ordenadas por día y hora.	De La Piedra Quintanilla, Erwin Miquel To Do
US01	Visualizar ritmo cardíaco en tiempo real	T08	Crear gráfico en frontend	Implementar componente que muestre los datos del ritmo cardíaco en tiempo real.	Mendez Rosales, Marco Aurelio To Do
		T09	Conectar frontend con datos simulados	Programar lectura y renderizado dinámico desde fuente simulada.	Roca Huapaya, Orlando Arturo To Do

6.2.1.4. Development Evidence for Sprint Review.

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
SafeMedUPC/CAP5	main	f5d581f	feat: added student outcomes		15/05/2025
SafeMedUPC/CAP5	main	dfd614f	Mockups and wireflows		15/05/2025
SafeMedUPC/CAP5	main	b2ad1a4	Add files via upload		15/05/2025
SafeMedUPC/CAP5	main	515b21c	Merge pull request #14 from SafeMedUPC/CAP5		15/05/2025

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
SafeMedUPC/CAP5	main	9535efa	Merge branch 'main' into CAP5		15/05/2025
SafeMedUPC/CAP5	main	3345e13	add		15/05/2025
SafeMedUPC/CAP5	main	8440ca6	Update upc-pre-202510-1asi0572-2941-SafeMed-report-tb1.md		15/05/2025
SafeMedUPC/CAP5	main	4108947	feat(cap5): add chapter from 5 to 5.3.2		15/05/2025
SafeMedUPC/CAP5	main	cf4c064	Merge remote-tracking branch 'origin/main'		15/05/2025
SafeMedUPC/CAP5	main	3990400	feat(cap5): add chapter from 5 to 5.3.2		15/05/2025
SafeMedUPC/CAP5	main	c9cdcd2	Merge pull request #11 from SafeMedUPC/CAP5		15/05/2025
SafeMedUPC/CAP5	main	f6427d8	feat: sprint backlog 1		15/05/2025
SafeMedUPC/CAP5	main	8ce9b1e	feat:add images		15/05/2025
SafeMedUPC/CAP5	main	be9552a	Merge pull request #10 from SafeMedUPC/CAP5		15/05/2025
SafeMedUPC/MedSystembackend	main	42709b4	Initial commit		14/05/2025
SafeMedUPC/MedSystembackend	master	c8b5a03	fix: fix endpoints for authentication and doctors/patients		14/05/2025
SafeMedUPC/MedSystemFrontend	master	42709b4	Initial commit		14/05/2025

6.2.1.5. Testing Suite Evidence for Sprint Review.

Feature: Gestión de citas médicas y monitorización

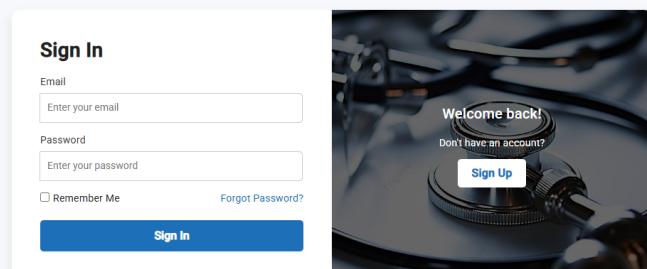
- User Story US01 Scenario: Agendar una cita desde la app *Given* que el paciente accede a la interfaz de agendamiento *When* crea una cita seleccionando fecha, hora y motivo *Then* la vista debe permitir validar el formulario y preparar los datos para enviar al backend *And* el backend debe registrar la cita médica con los datos del paciente
- User Story US02 Scenario: Ver citas programadas *Given* que el paciente desea revisar sus citas agendadas *When* accede a la vista de citas programadas *Then* debe visualizar las citas con todos los detalles *And* el backend debe devolver las citas programadas por el usuario

- User Story US03 Scenario: Cancelar o reprogramar una cita *Given* que el paciente tiene una cita agendada *When* decide cancelar o reprogramar la cita *Then* debe poder modificar o eliminar la cita desde el frontend
- User Story US04 Scenario: Sincronización con agenda del doctor *Given* que el doctor desea revisar su agenda *When* accede a la vista de agenda *Then* debe visualizar sus citas ordenadas por día y hora
- User Story US01 (Monitorización) Scenario: Visualizar ritmo cardíaco en tiempo real *Given* que el paciente desea monitorear su ritmo cardíaco *When* accede al componente gráfico en la app *Then* debe mostrar datos en tiempo real *And* la app debe conectarse con los datos simulados y renderizarlos dinámicamente

6.2.1.6. Execution Evidence for Sprint Review.

- Imagen del la aplicacion web corriendo en la consola de firebase y en la web

The screenshot shows the Firebase Hosting dashboard for the project 'MedSystem'. The left sidebar includes sections for Descripción general, Accesos directos a proyectos, Hosting (selected), Categorías de producto, Compilación, Ejecución, Analytics, IA, Todos los productos, and Herramientas de desarrollo relacionadas. The main area displays the 'Administrador sitio' section with tabs for Panel and Uso. It shows the 'Versión actual' (version 15/5/25, 5:23 p.m.) and 'Versiones anteriores' (version 15/5/25, 6:23 p.m.). It also lists 'Dominios' (medsystem-b45f8.web.app and medsystem-b45f8.firebaseioapp.com, both set to 'Predeterminado') and provides options to 'Agregar un dominio personalizado' and 'Ver los 2 dominios'. A note at the bottom encourages using preview channels to test changes before publishing. The browser address bar shows 'medsystem-b45f8.web.app/sign-in'.



6.2.1.7. Services Documentation Evidence for Sprint Review.

Imagenes del backend corriendo localmente, de los bounded context mas importantes

Results Results Management Endpoints

- POST** /api/v1/results
- GET** /api/v1/results/patientId/{patientId}
- GET** /api/v1/results/doctorId/{doctorId}

Reports Report Management Endpoints

- GET** /api/v1/reports
- POST** /api/v1/reports
- GET** /api/v1/reports/{id}
- GET** /api/v1/reports/patientId/{patientId}

Treatments Treatment Management Endpoints

- GET** /api/v1/treatments
- POST** /api/v1/treatments
- GET** /api/v1/treatments/patientId/{patientId}
- DELETE** /api/v1/treatments/treatmentName/{treatmentName}

Security Service for MedSystem API v1.0.0 OAS 3.0

/v3/api-docs

Security for application REST API documentation.

MedSystem 2.0

ACME Learning Platform Wiki Documentation

Servers

Authorize

Authentication Authentication API

- POST** /api/v1/auth/sign-up
- POST** /api/v1/auth/sign-in

Roles Roles API

- GET** /api/v1/roles

Users Users API

- GET** /api/v1/users
- GET** /api/v1/users/{userId}

Schemas

ACME Learning Platform API v1.0.0 OAS 3.0

/v3/api-docs

ACME Learning Platform application REST API documentation.

Apache 2.0

ACME Learning Platform Wiki Documentation

Servers

http://localhost:8003 - Generated server url ▾

Appointments Endpoints for managing appointments ^

PUT /api/v1/appointments/{appointmentId}/reason

PUT /api/v1/appointments/{appointmentId}/date

GET /api/v1/appointments

POST /api/v1/appointments

GET /api/v1/appointments/{id}

GET /api/v1/appointments/patientId/{patientId}

GET /api/v1/appointments/doctorId/{doctorId}

DELETE /api/v1/appointments/{appointmentId}

6.2.1.8. Software Deployment Evidence for Sprint Review.

Usamos firebase para el deployment

En la terminal escribimos lo siguiente:

- npm install -g firebase-tools
 - firebase login
 - firebase init Nos debe aparecer esto

Luego ponemos :

- npm run build Por ultimo :

- firebase deploy

```

Terminal Local + 
Lazy chunk files | Names | Raw size | Estimated transfer size
chunk-MIAKXSH.js   browser      60.07 kB    16.64 kB

Application bundle generation complete. [5.766 seconds]
Output location: C:\Users\PC\Downloads\MedSystemFrontend\MedSystemFrontend\dist\med-system-frontend
PS C:\Users\PC\Downloads\MedSystemFrontend\MedSystemFrontend> firebase deploy
! JSON error trying to load C:\Users\PC\Downloads\MedSystemFrontend\MedSystemFrontend\firebase.json
*** Deploying to 'medsystem-b45f8'...
  1 deploying hosting
  i hosting[medsystem-b45f8]: beginning deploy...
  i hosting[medsystem-b45f8]: Found 8 files in dist/med-system-frontend/browser
  p  hosting: uploading new files [0/8] (0%)
  
```

Nos dara el link: <https://medsystem-b45f8.web.app>

6.2.1.9. Team Collaboration Insights during Sprint.



Anexos:

- Video Exposicion TB1: https://upcedupe-my.sharepoint.com/_/g/personal/u202112353_upc_edu_pe/ERYan0Gr4ddPj0vWCWtqHtcBz1ElzzOZSzZhc3GBszgTUg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJtDHzJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rliwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D&e=1GoFPi
- Video Exposicion TP: https://upcedupe-my.sharepoint.com/_/g/personal/u202112353_upc_edu_pe/EegUrsyT9mtFjhif5v_swawB--FbLpaINq9MCwlbGLrzZg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcmI2ZUZvckJ1c2luZXNzliwicmVmZXJyYWxBc

HBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=WLzORb