

**Universidad Peruana de Ciencias Aplicadas****Ingeniería de Software****Ciclo 2024-2****Desarrollo de Soluciones IoT****Sección:** WS71 **Profesor:** Angel Augusto Velasquez Nuñez**INFORME DEL TRABAJO FINAL****Startup:** SocialTech**Producto:** MIAM (Monitoreo Inteligente para Adultos Mayores)**Integrantes:**

Achamizo Huamani, Jean Carlos

Aliaga Trevejo, Lucía Guadalupe

Raymundo Guevara, Rodrigo Alejandro

Sagastegui Rodriguez, Luis Jesús

Siancas Reategui, Luis Alberto

Trujillo Lopez, Luis Alberto

Setiembre, 2024**Registro de Versiones del Informe**

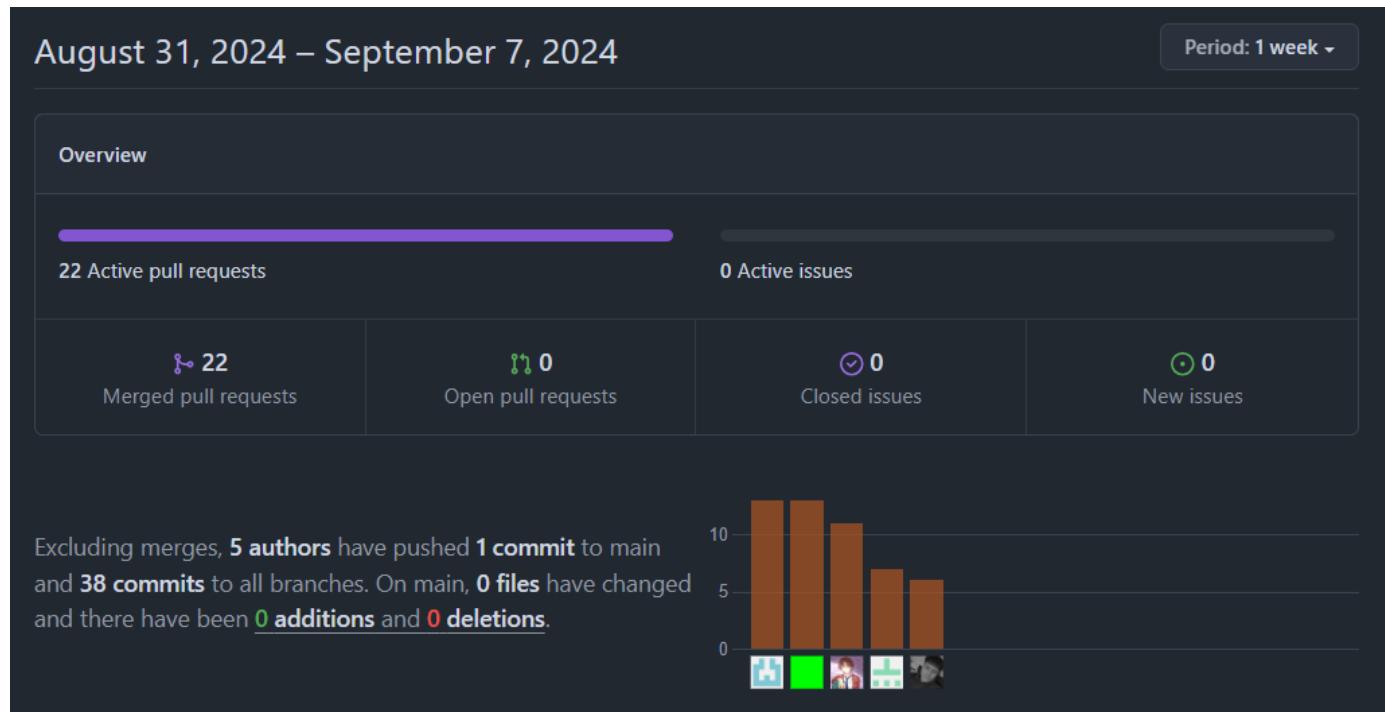
Versión	Fecha	Autor	Descripción de la modificación
1.0	06/09/2024	Achamizo Huamani Jean Carlos, Aliaga Trevejo Lucía Guadalupe, Raymundo Guevara Rodrigo Alejandro, Siancas Reategui Luis Alberto, Trujillo Lopez Luis Alberto	Se añadieron los elementos correspondientes al entregable de la TB1 (capítulos 1 al 4)
1.1	25/09/2024	Aliaga Trevejo Lucía Guadalupe	Se añadieron y corrigieron los elementos correspondientes en el capítulo de Needfinding
1.2	25/09/2024	Achamizo Huamani Jean Carlos, Siancas Reategui Luis Alberto	Se añadieron y corrigieron los elementos correspondientes en el capítulo strategic level software design, como la descripción y las imágenes de cada diagrama.
1.3	26/09/2024	Achamizo Huamani Jean Carlos	Se corrigieron los elementos correspondientes del Lean UX Process, los apartados de Lean UX Assumptions y Lean UX Hypothesis Statements
1.3	27/09/2024	Raymundo Guevara Rodrigo Alejandro	Se añadieron los mockups y wireframes en la sección de diseño de interfaz.
1.4	27/09/2024	Sagastegui Rodriguez Luis Jesus	Se añadieron los cuadros de planificación del sprint en la sección de metodología ágil.
1.5	27/09/2024	Trujillo Lopez Luis Alberto	Se completaron los cuadros de planificación del sprint con detalles adicionales y se validaron las fechas y tareas.
2.0	31/10/2024	Aliaga Trevejo Lucía Guadalupe	Se agregó el capítulo correspondiente al Sprint 2

Project Report Collaboration Insights

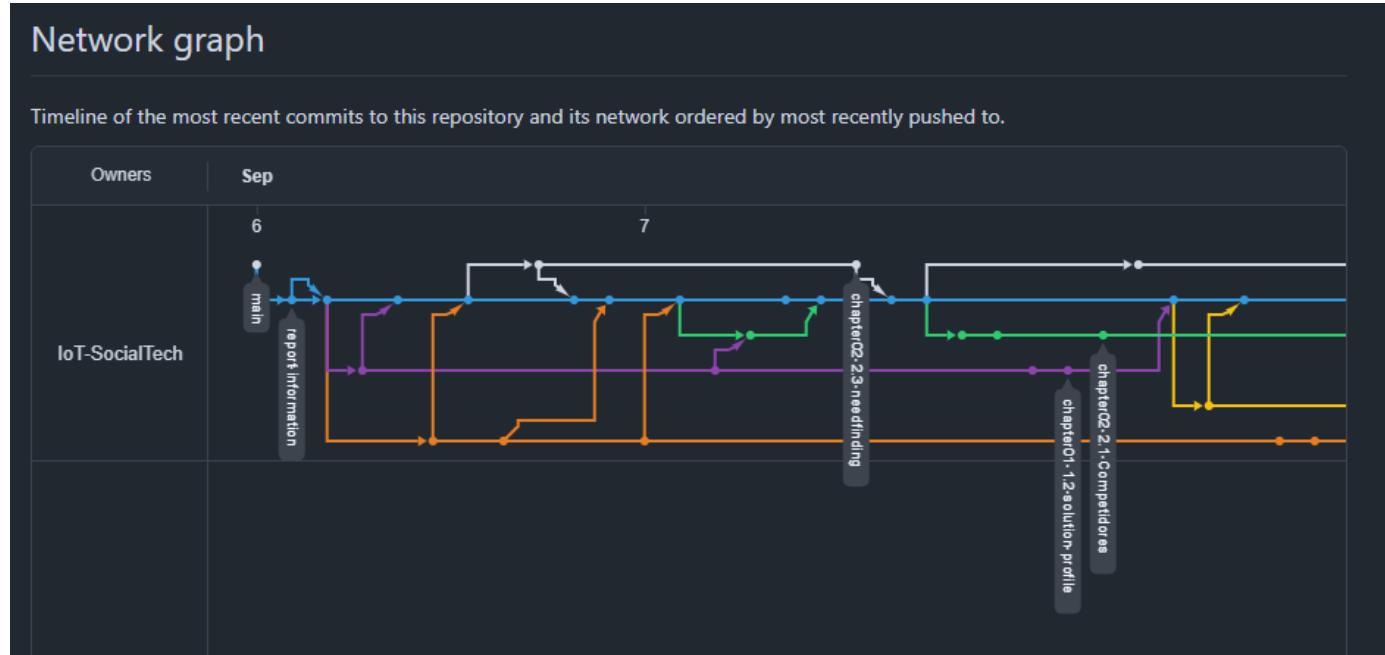
URL del repositorio: <https://github.com/IoT-SocialTech/iot-report>

TB1

El equipo colaboró activamente en el repositorio de GitHub para la elaboración del informe, realizando un total de 38 commits para el primer entregable.

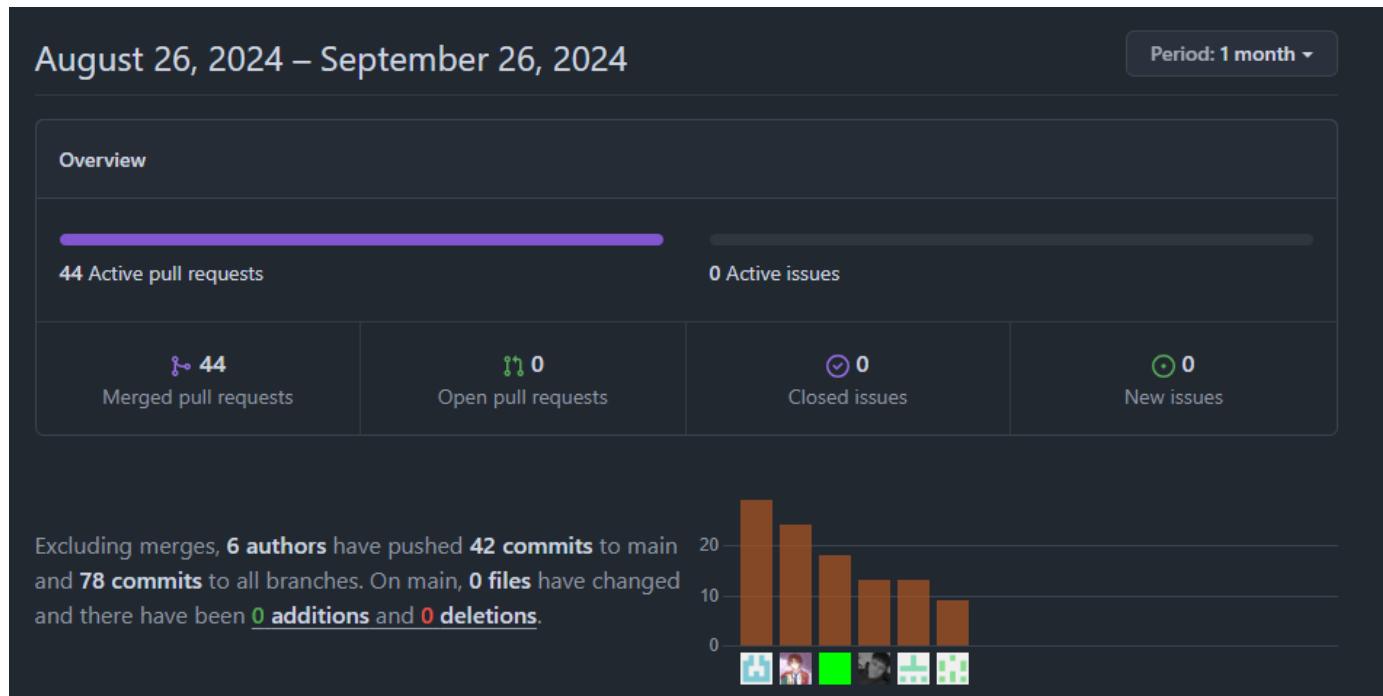


El uso de ramas paralelas permitió a los miembros del equipo trabajar simultáneamente en diferentes secciones del informe, integrando cambios de manera eficiente y reduciendo conflictos. Este enfoque colaborativo garantizó que todos los miembros contribuyeran al desarrollo del informe de manera equilibrada.



TP1

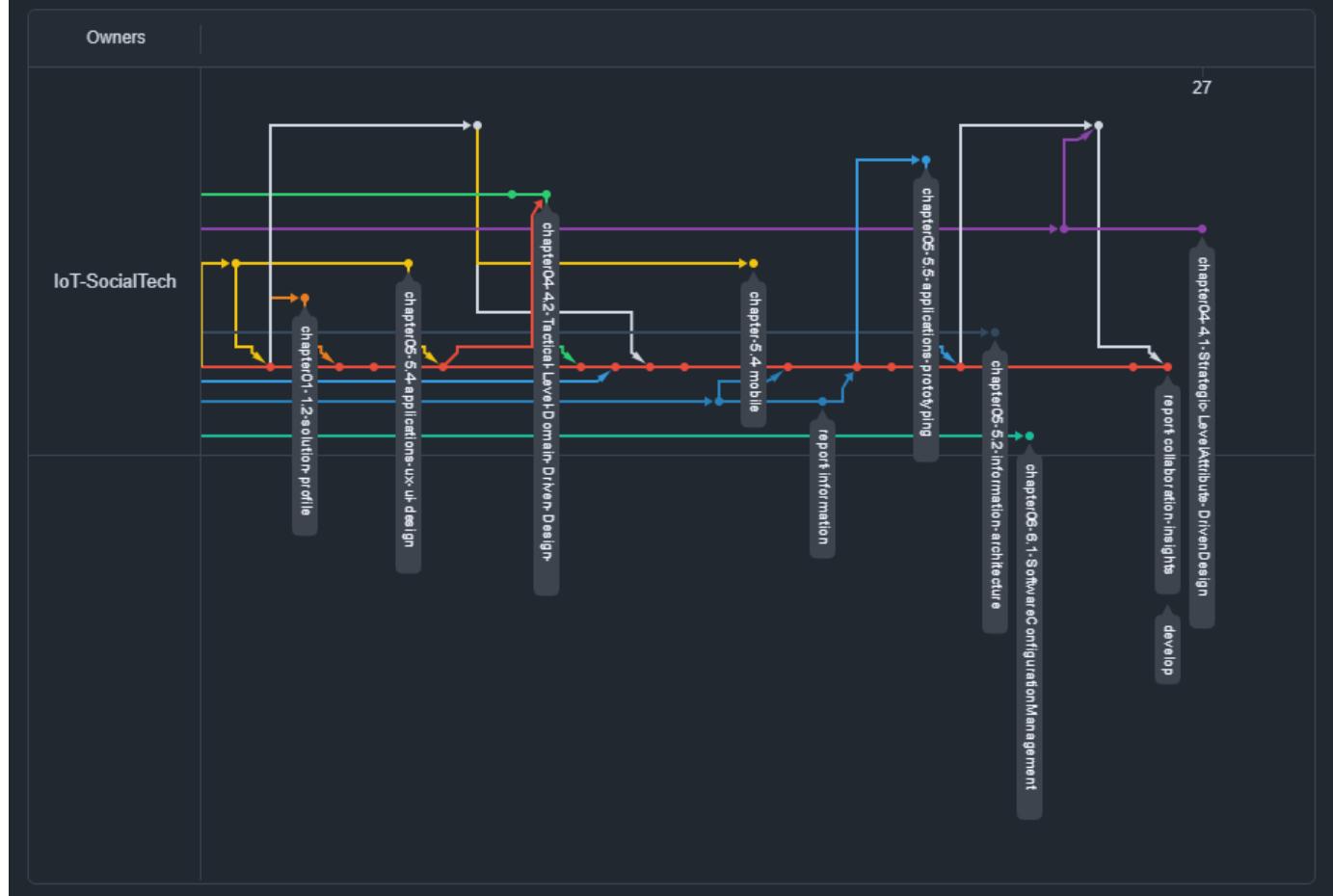
Para la segunda entrega, el equipo logró un total de 78 commits en todas las ramas.



Se continuó el trabajo en ramas paralelas para integrar las correcciones y nuevos elementos. Gracias a este enfoque se evitaron conflictos entre las nuevas versiones del reporte.

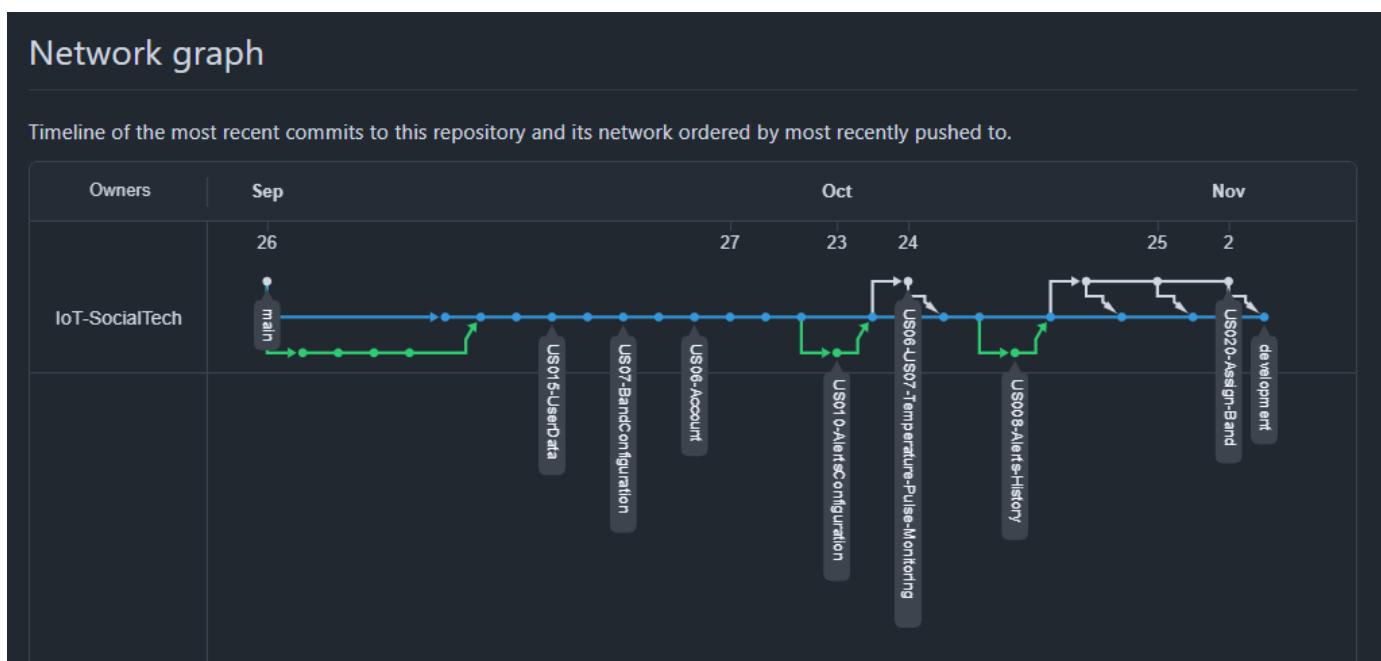
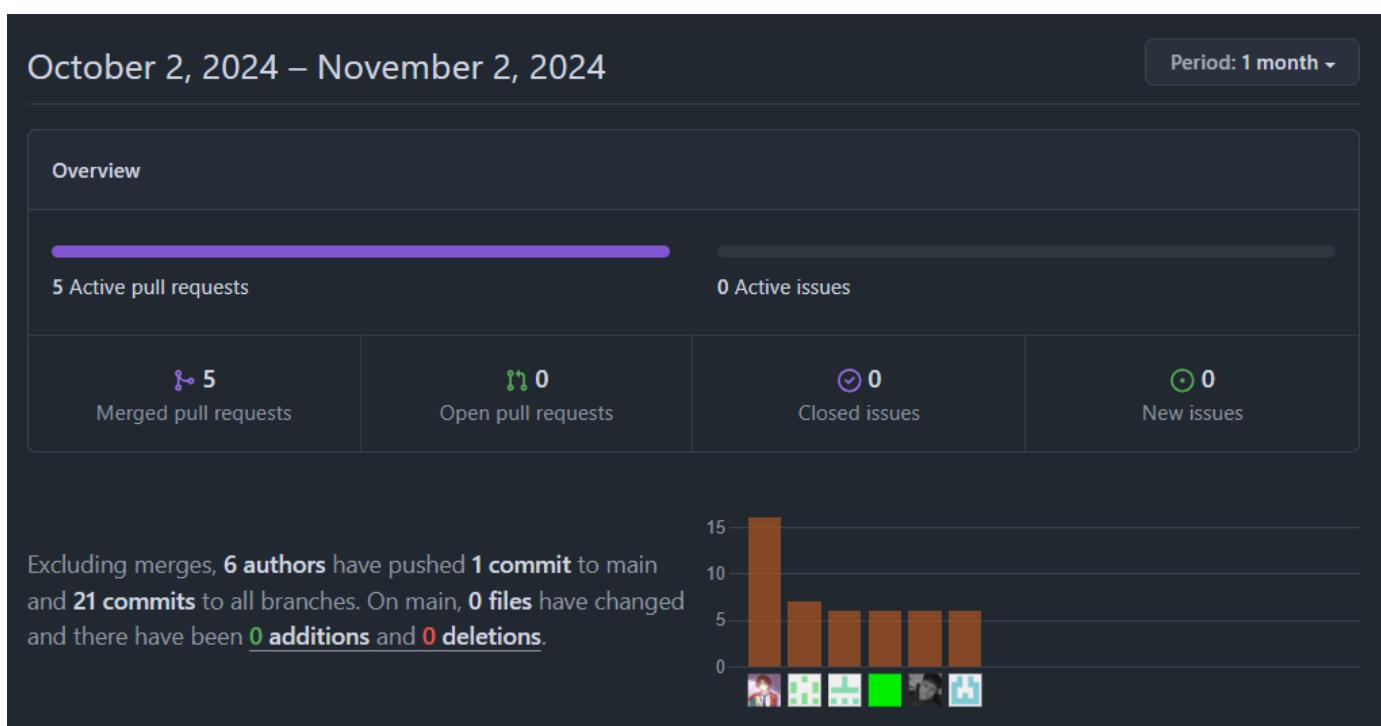
Network graph

[Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.](#)



TB2

Para la tercera entrega se adicionan los commits en los repositorios de los productos:

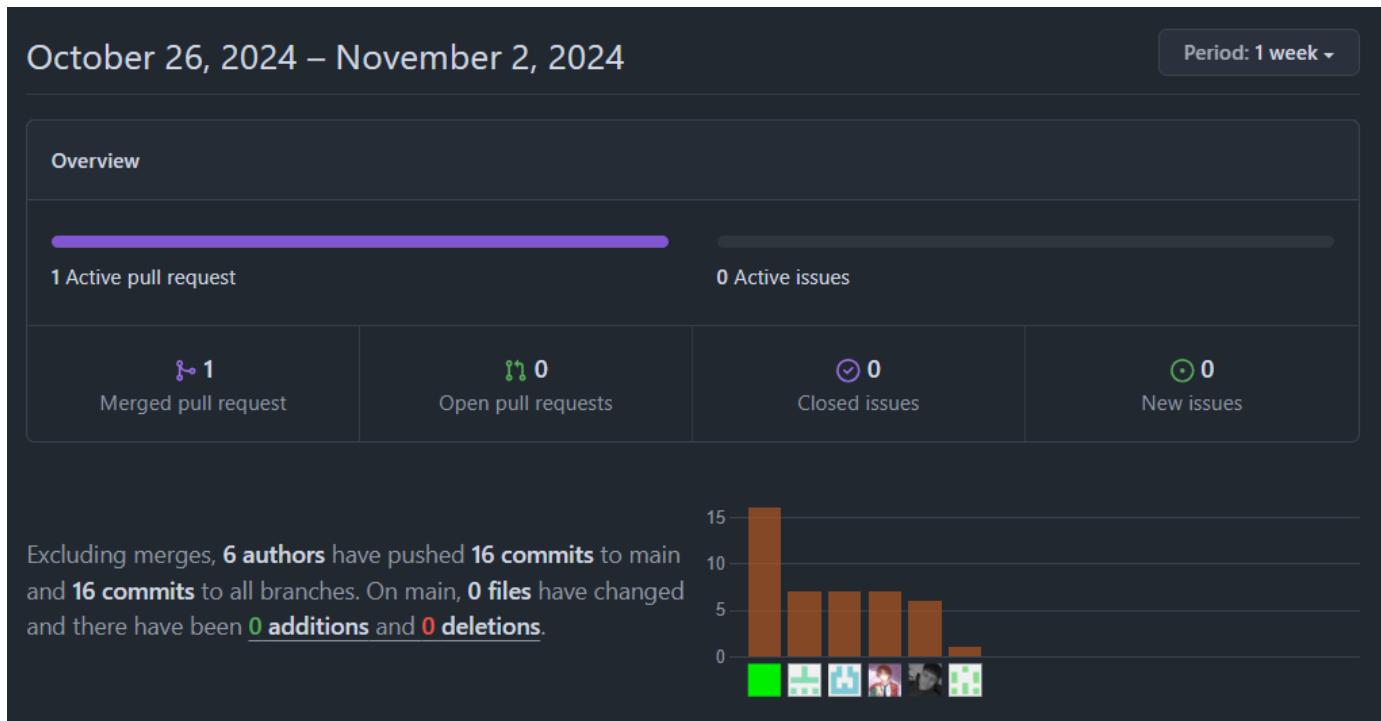
Web Application*Mobile Application*

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Embedded Application



Network graph

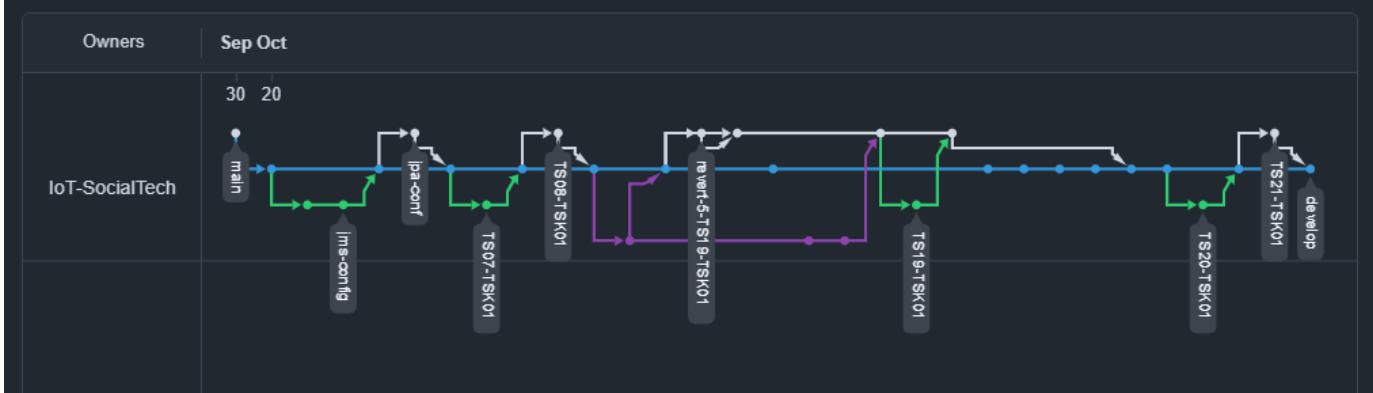
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Edge API

Network graph

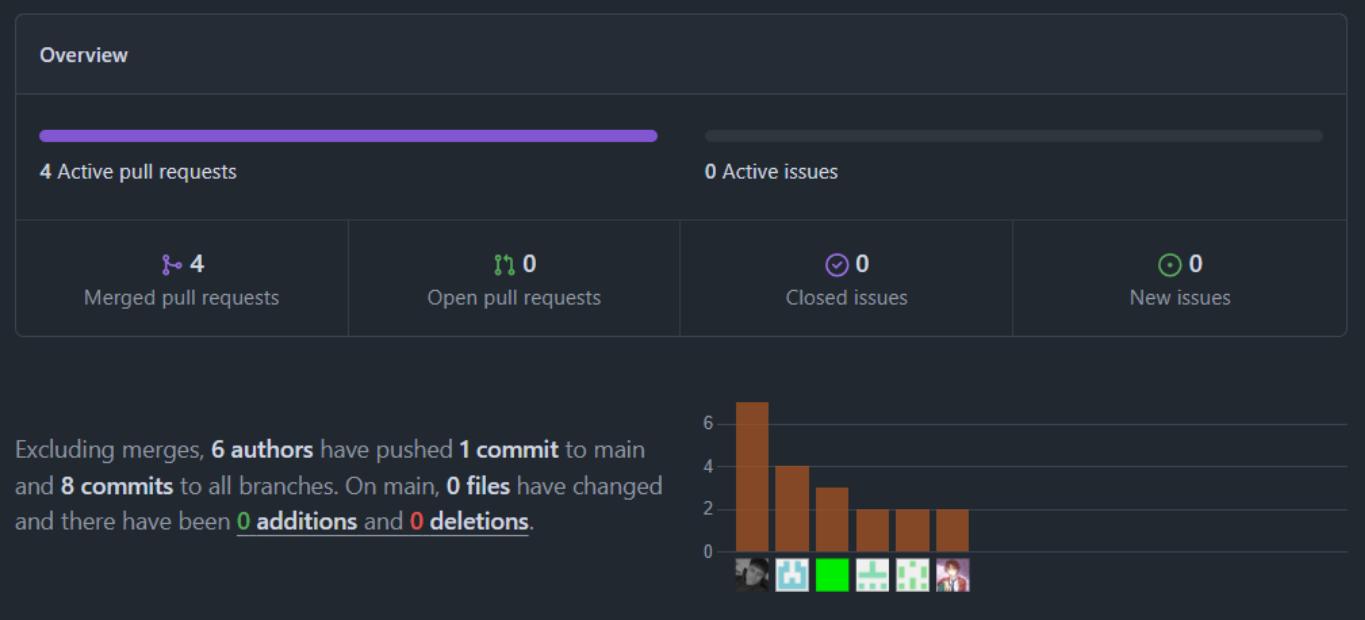
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Cloud API

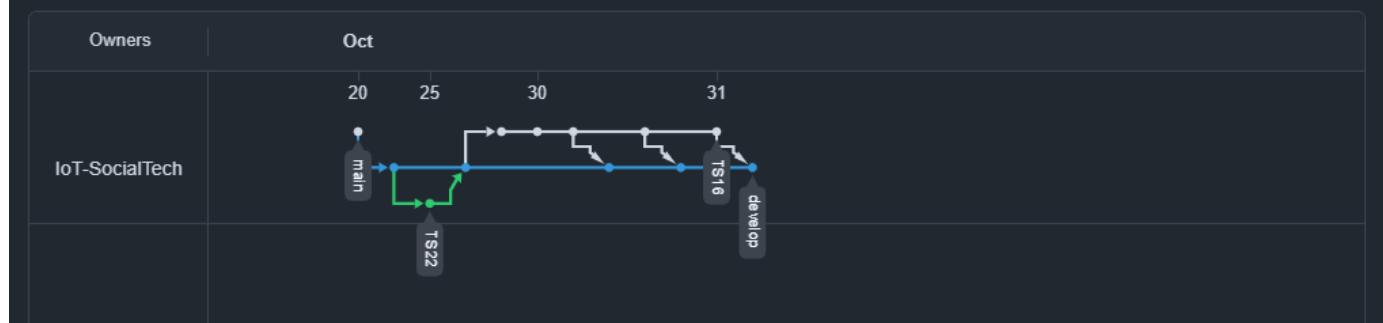
October 2, 2024 – November 2, 2024

Period: 1 month ▾



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Se continuó el trabajo en ramas paralelas para integrar las correcciones y nuevos elementos. Gracias a este enfoque se evitaron conflictos entre las nuevas versiones del reporte.

Contenido

Tabla de Contenidos

- Registro de Versiones del Informe
 - | 1.3 | 26/09/2024 | Achamizo Huamani Jean Carlos| Se corrigieron los elementos correspondientes del Lean UX Process, los apartados de Lean UX Assumptions y Lean UX Hypothesis Statements |
- Project Report Collaboration Insights
- Contenido
 - 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: Strategic-Level Software Design
 - 4.1. Strategic-Level Attribute-Driven Design
 - 4.1.1. Event Storming
 - 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
 - Proceso para Crear el Context Mapping y Análisis de Alternativas
 - 1. Pasos para Crear el Context Mapping
 - 1.1. Identificación de los Bounded Contexts
 - 1.2. Identificación de Relaciones Iniciales
 - 2. Análisis de Alternativas y Preguntas Clave
 - 2.1. ¿Qué pasaría si movemos este capability a otro bounded context?
 - 2.2. ¿Qué pasaría si descomponemos este capability y movemos uno de los sub-capabilities a otro bounded context?
 - 2.3. ¿Qué pasaría si partimos el bounded context en múltiples bounded contexts?
 - 2.4. ¿Qué pasaría si tomamos este capability de estos 3 contexts y lo usamos para formar un nuevo context?
 - 2.5. ¿Qué pasaría si duplicamos una funcionalidad para romper la dependencia?
 - 2.6. ¿Qué pasaría si creamos un shared service para reducir la duplicación entre múltiples bounded contexts?
 - 2.7. ¿Qué pasaría si aislamos los core capabilities y movemos los otros a un context aparte?
 - 3. Alternativa Recomendada de Context Mapping
 - 4. Patrones de Relaciones Sugeridos
 - 4.1.3. Software Architecture
 - 4.1.3.1. Software Architecture System Landscape Diagram
 - 4.1.3.2. Software Architecture Context Level Diagrams
 - 4.1.3.3. Software Architecture Container Level Diagrams
 - 4.1.3.4. Software Architecture Deployment Diagrams
 - 4.2. Tactical-Level Domain-Driven Design
 - 4.2.1. Bounded Context: Device Context
 - 4.2.1.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.1.2. Interface Layer
 - Controllers
 - 4.2.1.3. Application Layer
 - Command Handlers
 - Event Handlers
 - Query Handlers
 - 4.2.1.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.1.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Edge API):
 - 4.2.1.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.1.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.1.7.2. Bounded Context Database Design Diagram
 - 4.2.2. Bounded Context: Edge Context
 - 4.2.2.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.2.2. Interface Layer
 - Controllers
 - 4.2.2.3. Application Layer
 - Command Handlers

- Event Handlers
 - Query Handlers
 - 4.2.2.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.2.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Edge API):
 - 4.2.2.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.2.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.2.7.2. Bounded Context Database Design Diagram
-
- 4.2.3. Bounded Context: Notification Context
 - 4.2.3.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.3.2. Interface Layer
 - Controllers
 - 4.2.3.3. Application Layer
 - Command Handlers
 - Event Handlers
 - Query Handlers
 - 4.2.3.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.3.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Web API):
 - 4.2.3.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.3.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.3.7.2. Bounded Context Database Design Diagram
-
- 4.2.4. Bounded Context: Metrics Context
 - 4.2.4.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.4.2. Interface Layer
 - Controllers
 - 4.2.4.3. Application Layer
 - Command Handlers
 - Event Handlers
 - Query Handlers
 - 4.2.4.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.4.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Edge API):
 - 4.2.4.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.4.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.4.7.2. Bounded Context Database Design Diagram
- 4.2.5. Bounded Context: Payment Context
 - 4.2.5.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.5.2. Interface Layer
 - Controllers
 - 4.2.5.3. Application Layer

- Command Handlers
 - Event Handlers
 - Query Handlers
 - 4.2.5.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.5.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Web API):
 - 4.2.5.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.5.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.5.7.2. Bounded Context Database Design Diagram
 - 4.2.6. Bounded Context: Configuration Context
 - 4.2.6.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.6.2. Interface Layer
 - Controllers
 - 4.2.6.3. Application Layer
 - Command Handlers
 - Event Handlers
 - Query Handlers
 - 4.2.6.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.6.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Edge API):
 - 4.2.6.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.6.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.6.7.2. Bounded Context Database Design Diagram
 - 4.2.7. Bounded Context: Account Context
 - 4.2.7.1. Domain Layer
 - Entities
 - Value Objects
 - Domain Services
 - Aggregates y Aggregate Root
 - Repositories (Interfaces)
 - 4.2.7.2. Interface Layer
 - Controllers
 - 4.2.7.3. Application Layer
 - Command Handlers
 - Event Handlers
 - Query Handlers
 - 4.2.7.4. Infrastructure Layer
 - Repositories (Implementaciones)
 - 4.2.7.6. Bounded Context Software Architecture Component Level Diagrams
 - Components (Web API):
 - 4.2.7.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.7.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.7.7.2. Bounded Context Database Design Diagram
- Capítulo V: Solution UI/UX Design
 - 5.1. Strategic-Level Attribute-Driven Design
 - 5.1.1. General Style Guidelines
 - 5.1.1.1. Branding
 - 5.1.1.2. Logotipo
 - 5.1.1.3. Typography
 - 5.1.1.4. Colors
 - 5.1.1.5. Spacing
 - 5.1.2. Web, Mobile and IoT Style Guidelines
 - 5.1.2.1. Web Style Guidelines

- 5.1.2.2. Mobile Style Guidelines
 - 5.2. Information Architecture
 - 5.2.1. Organization Systems.
 - 5.2.2. Labeling Systems.
 - 5.2.3. SEO Tags and Meta Tags.
 - 5.2.4. Navigation Systems.
 - 5.2.5. Navigation Systems.
 - 5.2.3. SEO Tags and Meta Tags
 - 5.2.4. Searching Systems.
 - 5.2.5. Navigation Systems.
 - 5.3. Landing Page UI Design
 - 5.3.1. Landing Page Wireframe
 - 5.3.2. Landing Page Mock-up
 - 5.4. Applications UX/UI Design
 - 5.4.1. Applications Wireframes
 - 5.4.1.1. Web Application Wireframes
 - 5.4.1.2. Mobile Application Wireframes
 - 5.4.2. Applications Wireflow Diagrams
 - 5.4.2.1. Web Application Wireflow Diagrams
 - 5.4.2.2. Mobile Application Wireflow Diagrams
 - 5.4.3. Applications Mock-ups
 - 5.4.3.1. Web Application Mock-ups
 - 5.4.3.2. Mobile Application Mock-ups
 - 5.4.4. Applications User Flow Diagrams
 - 5.4.4.1. Web Application User Flow Diagrams
 - 5.4.4.2. Mobile Application User Flow Diagrams
 - 5.5 Applications Prototyping
- Capítulo VI: Product Implementation, Validation & Deployment
 - 6.1. Software Configuration Management
 - 6.1.1. Software Development Environment Configuration
 - 6.1.2. Source Code Management
 - 6.1.3. Source Code Style Guide & Conventions
 - 6.1.4. Software Deployment Configuration
 - 6.2.1. Sprint 1
 - 6.2.1.1. Sprint Planning 1
 - 6.2.1.2. Sprint Backlog 1
 - 6.2.1.3. Development Evidence for Sprint Review
 - 6.2.1.4. Testing Suite Evidence for Sprint Review
 - 6.2.1.5. Execution Evidence for Sprint Review
 - 6.2.1.6. Services Documentation Evidence for Sprint Review
 - 6.2.1.7. Software Deployment Evidence for Sprint Review
 - 6.2.1.8. Team Collaboration Insights during Sprint
 - 6.2.2. Sprint 2
 - 6.2.2.1. Sprint Planning 2
 - 6.2.2.2. Sprint Backlog 2
 - 6.2.2.3. Development Evidence for Sprint Review
 - 6.2.2.4. Testing Suite Evidence for Sprint Review
 - 6.2.2.5. Execution Evidence for Sprint Review
 - 6.2.2.6. Services Documentation Evidence for Sprint Review
 - 6.2.2.7. Software Deployment Evidence for Sprint Review
 - 6.2.2.8. Team Collaboration Insights during Sprint
 - 6.3 Validation Interviews.
 - 6.3.1. Diseño de Entrevistas.
 - 6.3.2. Registro de Entrevistas.
 - 6.3.3. Evaluaciones según heurísticas.
 - 6.4. Video About-the-Product.
- Conclusiones
- Bibliografía
- Anexos
 - Videos de Exposición
 - Diagrama C4 MIAM

Student Outcome

Criterio específico	Acciones realizadas	Conclusión
Trabaja en equipo para proporcionar liderazgo en forma conjunta	<p>Jean Carlos Achamizo Huamani</p> <p>TB1:</p> <p>Implementé inicialmente los apartados de assumptions e hipótesis, estructurando las bases del proyecto y facilitando la comprensión común de los objetivos. Esta acción ayudó a definir un enfoque claro y unificado para el equipo, asegurando que todos los miembros estuvieran alineados desde el principio.</p> <p>TP1:</p> <p>Corregí los assumptions originales, añadiendo cinco nuevos y elaboré hipótesis detalladas utilizando la plantilla proporcionada. Esto permitió al equipo validar los objetivos del proyecto y adaptar las actividades según los requerimientos específicos, lo que mejoró la toma de decisiones conjunta y la planificación estratégica.</p> <p>TB2:</p> <p>Colaboré en el desarrollo de funcionalidades del backend, optimizando las llamadas de API y estructurando la lógica de negocio. En el frontend, implementé componentes interactivos para mejorar la experiencia de usuario en la landing page, lo que reforzó el liderazgo conjunto y permitió una visión integral del proyecto.</p>	<p>Conclusión TB1: El equipo estableció una base sólida para el proyecto al definir claramente las responsabilidades y objetivos iniciales, promoviendo un liderazgo conjunto y una toma de decisiones participativa. Se crearon los apartados iniciales y se definieron roles, lo que facilitó el trabajo en equipo.</p> <p>Conclusión TP1: El equipo mejoró la planificación y ejecución del proyecto al revisar y corregir los elementos iniciales. Cada miembro contribuyó a profundizar y perfeccionar sus responsabilidades, permitiendo una mayor coordinación y efectividad en la consecución de los objetivos. Todos los roles estuvieron alineados, lo que reforzó el liderazgo conjunto y la cohesión del equipo.</p> <p>Conclusión TB2: El equipo trabajó de manera integrada en diversas áreas del proyecto (frontend, backend, device y landing page), fortaleciendo el liderazgo compartido. Esta colaboración permitió que todos los miembros estuvieran alineados con los objetivos generales, facilitando una visión integral del desarrollo y una toma de decisiones en conjunto.</p>
Luis Alberto Trujillo Lopez	<p>TB1:</p> <p>Añadí las páginas web de los competidores y sus logos, proporcionando una base inicial para el análisis competitivo. Esto facilitó la comprensión del panorama del mercado y permitió al equipo definir estrategias de manera más informada.</p> <p>TP1:</p> <p>Expandí esta información agregando descripciones detalladas de cada competidor y reorganizando las imágenes del Event Storming en secciones claras con descripciones para cada fase. Al hacer esto, ayudé a clarificar los roles y responsabilidades dentro del equipo, promoviendo un liderazgo compartido y un trabajo en equipo más cohesivo.</p> <p>TB2:</p> <p>Participé en la implementación del diseño de la landing page y colaboré en el backend configurando los endpoints necesarios para la funcionalidad de los dispositivos, fortaleciendo el liderazgo compartido y garantizando un trabajo en equipo cohesivo.</p>	
Lucía Guadalupe Aliaga Trevejo	<p>TB1:</p> <p>Realicé los primeros resúmenes de entrevistas y la descripción inicial de los User Personas. Facilitando las discusiones del equipo, me aseguré de que todas las opiniones fueran consideradas, promoviendo un entorno de trabajo colaborativo.</p>	

Criterio específico	Acciones realizadas	Conclusión
	<p>TP1:</p> <p>Mejoré los enlaces y resúmenes de entrevistas, incluyendo detalles sobre marcas y personalidad del entrevistado. Corregí la descripción de los User Personas basándome en el análisis de entrevistas. Esto ayudó a alinear la visión del equipo con las necesidades reales de los usuarios, fortaleciendo el liderazgo conjunto en la toma de decisiones.</p> <p>TB2:</p> <p>Apoyé en el frontend en la creación de formularios interactivos en la landing page y colaboré en el device, asegurando que la comunicación entre frontend y backend fuera eficiente, facilitando un entorno de liderazgo compartido.</p>	
	<p>Luis Alberto Siancas Reategui</p> <p>TB1:</p> <p>Comencé el análisis inicial de entrevistas, incluyendo detalles generales sobre los entrevistados y tecnologías utilizadas. Esto proporcionó al equipo una comprensión básica del perfil de usuario.</p> <p>TP1:</p> <p>Añadí detalles específicos sobre las tecnologías utilizadas por los entrevistados, sus marcas preferidas y las influencias que afectan su toma de decisiones. Esta información fue clave para que el equipo pudiera tomar decisiones informadas y desarrollar estrategias que respondieran efectivamente a las necesidades identificadas.</p> <p>TB2:</p> <p>Desarrollé configuraciones adicionales en el backend para mejorar la gestión de dispositivos y ayudé en el frontend para asegurar una estructura visual consistente en la landing page, promoviendo la colaboración y el liderazgo cohesivo.</p>	
	<p>Rodrigo Alejandro Raymundo Guevara</p> <p>TB1:</p> <p>Definí las primeras User Stories y Epics, estableciendo la base del backlog del proyecto. Esto permitió al equipo organizar el trabajo en tareas claras y priorizadas, promoviendo una coordinación efectiva.</p> <p>TP1:</p> <p>Unifiqué el cuadro de User Stories y Epics, mejoré la redacción de las User Stories y sus escenarios en español, e incluí las User Stories de la landing page. Esto ayudó a estructurar mejor el trabajo del equipo, asegurando que todos estuvieran alineados con las prioridades del proyecto y facilitando un liderazgo compartido.</p> <p>TB2:</p> <p>Realicé ajustes en las User Stories para incluir nuevos requisitos del device y aseguré que la lógica de backend fuera compatible con el frontend en la landing page, promoviendo la sinergia del equipo y el liderazgo compartido.</p>	
	<p>Luis Jesús Sagastegui Rodríguez</p> <p>TB1:</p>	

Criterio específico	Acciones realizadas	Conclusión
	<p>Empecé la definición de las reglas de negocio básicas para los aggregates y el diseño inicial de la base de datos. Esto proporcionó una base técnica para la planificación del proyecto.</p> <p>TP1:</p> <p>Añadí reglas de negocio detalladas para los aggregates, elaboré diagramas de clases y de base de datos. Estas acciones facilitaron una mejor comprensión técnica del proyecto, permitiendo al equipo tomar decisiones técnicas de manera más colaborativa y eficiente.</p> <p>TB2:</p> <p>Aporté en la creación y ajuste de las reglas de negocio en el backend, permitiendo una comunicación fluida con el frontend para la funcionalidad del device y desarrollé el diseño de base de datos para la landing page, fortaleciendo la colaboración y el liderazgo técnico compartido.</p>	
Crea un entorno colaborativo e inclusivo, establece metas, planifica tareas y cumple objetivos	<p>Jean Carlos Achamizo Huamani</p> <p>TB1:</p> <p>Facilitó la creación de un entorno inclusivo al asegurar que todas las voces fueran escuchadas durante las discusiones del equipo. Definí metas claras y organicé la planificación de tareas iniciales. Esto ayudó a establecer una base sólida para que el equipo trabajara de manera coordinada hacia objetivos comunes.</p> <p>TP1:</p> <p>Corregí y expandí los assumptions, añadiendo cinco nuevos y elaborando hipótesis más específicas para guiar el trabajo del equipo. Estas acciones permitieron establecer objetivos más precisos y alineados con la visión del proyecto, facilitando un entorno colaborativo donde todos entendieron claramente su rol.</p> <p>TB2:</p> <p>Promoví un entorno inclusivo para asegurar que todos los miembros contribuyeran en el frontend y backend, colaborando en la planificación de tareas para garantizar que todos comprendieran sus responsabilidades y los objetivos comunes.</p> <p>Luis Alberto Trujillo Lopez</p> <p>TB1:</p> <p>Promoví la inclusión en el equipo al proporcionar la base del análisis competitivo con las páginas web y logos de competidores, facilitando la planificación inicial del análisis. Esto ayudó al equipo a comprender mejor el entorno competitivo y a definir metas realistas.</p> <p>TP1:</p> <p>Añadí descripciones detalladas de los competidores, mejorando la calidad del análisis y facilitando la colaboración del equipo en la definición de metas estratégicas. Al clarificar las diferencias y similitudes con la competencia, ayudé a que el equipo pudiera enfocar sus esfuerzos en áreas de oportunidad específicas.</p> <p>TB2:</p> <p>Definí metas en conjunto con el equipo, dividiendo las tareas para frontend, backend y device, asegurando que se cumplieran los objetivos en tiempo, facilitando una</p>	<p>Conclusión TB1: El equipo trabajó de manera efectiva en la planificación inicial del proyecto, estableciendo metas claras y asegurando la participación equitativa de todos los miembros. Cada miembro contribuyó a la creación de un entorno colaborativo e inclusivo, estableciendo las bases para un trabajo coordinado y enfocado en objetivos comunes.</p> <p>Conclusión TP1: Se consolidó la colaboración y la inclusión, revisando y mejorando los elementos iniciales y planificando de manera detallada las tareas. Cada miembro del equipo ajustó sus responsabilidades y contribuciones en función de los datos y análisis obtenidos, lo que permitió una mejor planificación y ejecución de los objetivos establecidos.</p> <p>Conclusión TB2: En esta fase, el equipo trabajó de manera colaborativa en frontend, backend, device y landing page, estableciendo metas y tareas claras para cada área. Se promovió un entorno inclusivo que facilitó el cumplimiento de los objetivos y permitió una planificación efectiva, asegurando que cada miembro contribuyera de manera significativa a la consecución de los objetivos del proyecto.</p>

Criterio específico	Acciones realizadas	Conclusión
	<p>planificación coordinada.</p> <p>Lucía Guadalupe Aliaga Trevejo</p> <p>TB1:</p> <p>Garantizé que las tareas y objetivos fueran distribuidos equitativamente. Realicé los resúmenes iniciales de entrevistas y la descripción de User Personas, maximizando el potencial de cada miembro del equipo. Esto permitió al equipo tener un punto de partida claro para entender las necesidades de los usuarios.</p> <p>TP1:</p> <p>Mejoré los resúmenes de entrevistas, incluyendo detalles importantes, y consolidé la User Task Matrix en un solo cuadro, describiendo tareas comunes y corrigiendo el User Journey Mapping. Estas acciones permitieron al equipo tener una visión más clara de las tareas y su importancia, facilitando la planificación y el cumplimiento de objetivos.</p> <p>TB2:</p> <p>Apoyé en la organización y planificación de las tareas, asignando objetivos claros en cada área (device y frontend) para asegurar que el equipo mantuviera un enfoque inclusivo y colaborativo.</p>	
	<p>Luis Alberto Siancas Reategui</p> <p>TB1:</p> <p>Supervisé el progreso del equipo en relación con las metas establecidas y facilité la planificación de tareas iniciales basadas en el análisis preliminar de entrevistas. Esto aseguró que el equipo estuviera alineado y enfocado en las actividades prioritarias.</p> <p>TP1:</p> <p>Completé el análisis de entrevistas, detallando las tecnologías utilizadas, marcas preferidas e influencias de los entrevistados. Esta información fue crucial para ajustar las metas y estrategias del equipo, asegurando que las tareas planificadas respondieran a las expectativas de los usuarios y cumplieran con los objetivos del proyecto.</p> <p>TB2:</p> <p>Colaboré en la planificación de tareas para el backend y device, asegurando que los miembros del equipo entendieran sus roles y responsabilidades, fortaleciendo la colaboración y cumplimiento de objetivos.</p>	
	<p>Rodrigo Alejandro Raymundo Guevara</p> <p>TB1:</p> <p>Colaboré en la creación de un ambiente de trabajo respetuoso y colaborativo, donde cada miembro pudo contribuir con sus ideas en la planificación de tareas iniciales y la redacción de User Stories. Esto permitió al equipo organizar el trabajo de manera efectiva desde el inicio.</p> <p>TP1:</p> <p>Mejoré la redacción de las User Stories y sus criterios de aceptación, prioricé las User Stories de la landing page y unifiqué las Epics y User Stories. Estas acciones ayudaron al equipo a tener una dirección clara y a organizar las tareas de manera que todos entendieran qué se</p>	

Criterio específico	Acciones realizadas	Conclusión
	<p>esperaba de cada uno, facilitando el cumplimiento de objetivos.</p> <p>TB2:</p> <p>Apoyé en la organización y división de tareas entre frontend y backend, estableciendo objetivos claros para la landing page y device, manteniendo al equipo alineado en un entorno inclusivo.</p> <p>Luis Jesús Sagastegui Rodríguez</p> <p>TB1:</p> <p>Implementé prácticas colaborativas e inclusivas, asegurando que todos los miembros estuvieran enfocados en las tareas iniciales establecidas y facilitando la planificación de objetivos. Esto permitió una mejor organización y claridad en el trabajo del equipo.</p> <p>TP1:</p> <p>Añadí reglas detalladas de los aggregates, diagramas de clases y estructura de la base de datos, organizando mejor la planificación de tareas y el cumplimiento de objetivos técnicos del proyecto. Estas acciones proporcionaron un marco técnico sólido para el equipo, permitiendo una colaboración más efectiva en el desarrollo de la solución.</p> <p>TB2:</p> <p>Participé en la planificación de tareas del backend y colaboré en la integración de las reglas de negocio con el frontend, manteniendo un entorno colaborativo y alineado en las metas y objetivos de la landing page y device.</p>	

Capítulo I: Introducción

1.1. Startup Profile

1.1.1. Descripción de la Startup

1.1.2. Perfiles de integrantes del equipo

Perfil	Foto
Nombre: Achamizo Huamani, Jean Carlos Carrera: Ingeniería de Software Descripción: Como estudiante de Ingeniería de Software en la Universidad UPC, mi entusiasmo por el aprendizaje constante me impulsa a buscar siempre nuevas maneras de perfeccionar mis habilidades académicas y aplicarlas en situaciones prácticas, también poseo aptitudes en comunicación asertiva y escucha activa. Reconozco la importancia de una comunicación clara y efectiva en cualquier proyecto de software, yuento con la capacidad de escuchar de manera atenta para comprender las necesidades y expectativas de los clientes y colegas de equipo.	
Nombre: Aliaga Trevejo, Lucía Guadalupe Carrera: Ingeniería de Software Descripción: Soy una estudiante de Ingeniería de Software, actualmente interesada en el desarrollo web front y back end. Me gusta crear interfaces intuitivas y llamativas para mejorar la experiencia de usuario. Me considero una persona responsable y centrada. Mis habilidades blandas incluyen comunicación efectiva, trabajo en equipo, liderazgo y adaptabilidad. Aspiro a contribuir de manera significativa en proyectos innovadores para seguir expandiendo mis conocimientos técnicos y capacidades.	
Nombre: Raymundo Guevara, Rodrigo Alejandro	

Carrera: Ingeniería de Software

Descripción: Soy estudiante de la carrera de Ingeniería de Software y un apasionado de la tecnología. Me gusta descubrir nuevas formas de solucionar problemas con diversos frameworks y algoritmos. Me considero una persona responsable y empática. Me gusta mucho trabajar en equipo y ayudar a mis compañeros



Nombre: Siancas Reategui, Luis Alberto

Carrera: Ingeniería de Software

Descripción: Soy estudiante de la carrera de Ingeniería de Software enfocado en el desarrollo backend. Me gusta crear API's las cuales uso para mis proyectos, por otro lado, me gusta realizar el despliegue de mis aplicaciones así como también usar diferentes diseños de arquitectura para que mi aplicación tenga un funcionamiento estable y tenga alta disponibilidad. En cuanto a mis habilidades blandas, soy empático y tengo escucha activa, esto me ayuda en gran parte a la hora de trabajar en equipos ya que facilita la comunicación y por ende los resultados son mejores.



Nombre: Trujillo Lopez, Luis Alberto

Carrera: Ingeniería de Software

Descripción: Soy un estudiante de Ingeniería de Software interesado en las tecnologías de aprendizaje autónomo. Me gusta desarrollar en un entorno web páginas que se vean llamativas y que tengan múltiples funcionalidades enfocándome en la experiencia del usuario y en la creación de API's. Poseo algunas habilidades que me permiten relacionarme en equipo, soy una persona responsable, con un enfoque centrado, comunicación asertiva, liderazgo y adaptabilidad. Mi meta es contribuir en el desarrollo de proyectos que me permitan mejorar y encontrar nuevas habilidades para adquirir mayores conocimientos laboral y profesionalmente.



Nombre: Sagastegui Rodriguez, Luis Jesus

Carrera: Ingeniería de Software

Descripción: Soy un estudiante de la carrera de ingeniería de Software y me apasiona mucho lo que hago. Siempre estoy interesado en aprender nuevas tecnologías y lenguajes de programación. Me gusta ver proyectos de las demás personas y poder compartir ideas. Me considero una persona responsable y muy amable que siempre le gusta ayudar a las demás personas. Siempre trato de dar lo mejor de mi para poder destacar.



1.2. Solution Profile

1.2.1 Antecedentes y problemática

Los adultos mayores representan una población vulnerable no solo a enfermedades relacionadas con el envejecimiento, sino también a lesiones que pueden deteriorar significativamente su calidad de vida (Dorri, Zabolinezhad, & Sattari, 2023). De acuerdo con la Organización Mundial de la Salud, las caídas son la principal causa de lesiones y mortalidad en personas mayores de 75 años (Giulianelli et al., 2017). Ante este panorama, es esencial proporcionarles un entorno seguro que contribuya a prevenir discapacidades y complicaciones en su salud (Dorri, Zabolinezhad, & Sattari, 2023).

A continuación, se presenta un análisis detallado de esta problemática empleando la metodología '5W2H'.

Who

Los adultos mayores en Perú, especialmente aquellos que viven en casas de reposo o en sus propios domicilios, representan una población vulnerable debido a las limitaciones propias del envejecimiento, como la pérdida de reflejos y del equilibrio (Suárez et al., 2020). Por otro lado, los cuidadores, tanto formales (como enfermeros y técnicos) como informales (familiares), son los responsables de brindar atención diaria. Estos cuidadores a menudo enfrentan desafíos adicionales como las exigencias de su trabajo y los recursos limitados. Esta situación puede dificultar la supervisión y la atención constante que los adultos mayores necesitan.

What

La principal problemática es la alta incidencia de caídas entre los adultos mayores, que no solo provoca lesiones físicas como fracturas y traumatismos, sino que también tiene consecuencias psicológicas como el miedo a caer de nuevo, lo cual reduce aún más su movilidad e independencia. Para los cuidadores, cada caída significa un aumento significativo en la carga de trabajo, implicando cuidados adicionales.

Where

Las caídas ocurren principalmente en las casas de reposo y domicilios de los adultos mayores. En las casas de reposo, la falta de personal y la infraestructura deficiente pueden contribuir al riesgo de caídas. En los domicilios, los adultos mayores a menudo se enfrentan a condiciones peligrosas como escaleras sin barandas, pisos resbaladizos, y la ausencia de dispositivos de asistencia como andadores o bastones.

When

Las caídas pueden suceder en cualquier momento del día, pero son más comunes durante actividades cotidianas como levantarse de la cama, caminar al baño, o asearse. Las noches y madrugadas son momentos críticos debido a la disminución de la visibilidad y la menor capacidad de reacción tanto de los adultos mayores como de los cuidadores.

Why

Las caídas suelen suceder por una combinación de factores, como la fragilidad del adulto mayor y una supervisión insuficiente.

How

Los adultos mayores pueden perder el equilibrio al realizar actividades diarias sin la asistencia adecuada. Los cuidadores no son capaces de realizar supervisión constante debido a que tienen varios pacientes a cargo, en especial en las casas de reposo, lo que puede llevar a descuidos involuntarios, aumentando el riesgo de caídas.

How Much

Uno de cada tres adultos mayores de 65 años sufre caídas, de las cuales el 68% resultan en lesiones, incluyendo fracturas. Además, el 60% de las caídas ocurren en el hogar, 30% en sitios públicos y 10% en instituciones de salud, teniendo consecuencias físicas y psicológicas para el adulto mayor (Ministerio de Salud, 2018).

1.2.2 Lean UX Process

1.2.2.1. Lean UX Problem Statements

El estado actual de la seguridad para adultos mayores se ha centrado principalmente en productos como botones de emergencia y sensores de movimiento básicos. Sin embargo, estos productos no logran ofrecer una supervisión continua ni adaptarse a los entornos específicos en los que viven los adultos mayores. Esto deja una brecha crítica en la prevención de caídas, lo que resulta en lesiones físicas y una reducción en la calidad de vida de los adultos mayores. Los cuidadores se enfrentan a la dificultad de supervisar constantemente y gestionar los riesgos sin herramientas adecuadas para ser más proactivos.

Nuestro producto abordará esta brecha mediante el desarrollo de un sistema innovador de monitoreo basado en IoT, diseñado para ofrecer una supervisión continua y alertas en tiempo real adaptadas a los entornos específicos de cada adulto mayor. Nuestro enfoque inicial serán los cuidadores y dueños de casas de reposo, con el objetivo de mejorar la seguridad y la calidad de vida de los adultos mayores.

Sabremos que tenemos éxito cuando veamos una reducción en los incidentes de caídas, una mayor satisfacción de los cuidadores y una mejora significativa en la calidad de vida de los adultos mayores.

1.2.2.2. Lean UX Assumptions

En esta sección se formulan las hipótesis basadas en las suposiciones previas, estableciendo relaciones claras entre las características del producto, los comportamientos esperados de los usuarios y los resultados de negocio deseados. Con puntos

1. Business Assumptions: Asumimos que los dueños de casas de reposo y cuidadores profesionales requieren una solución integrada y confiable para monitorear la salud y seguridad de los adultos mayores, reduciendo el riesgo de incidentes y mejorando la eficiencia en el cuidado diario.
2. Business Outcome Assumptions: Asumimos que los dueños de casas de reposo y cuidadores medirán el éxito de nuestro servicio por su capacidad para reducir en un 30% los incidentes de caídas y mejorar la respuesta a emergencias en un 50% durante el primer año de implementación.
3. User Assumptions: Asumimos que los cuidadores profesionales y los administradores de casas de reposo estarán dispuestos a adoptar nuevas tecnologías si estas les permiten monitorear de manera proactiva la salud de los residentes y reducir la carga laboral asociada a la supervisión manual.
4. User Outcome Assumptions: Asumimos que los cuidadores y administradores de casas de reposo experimentarán una mejora en la satisfacción laboral, una disminución del 20% en las tareas repetitivas de supervisión y un aumento del 40% en la percepción de seguridad por parte de los familiares de los residentes durante el primer semestre de uso del sistema.
5. Feature Assumptions: Asumimos que la combinación de sensores de movimiento y signos vitales, junto con alertas en tiempo real personalizables según el estado de salud de cada residente, será la característica clave que diferenciará nuestra solución de la competencia y captará al menos el 25% del mercado objetivo en el primer año.

1.2.2.3. Lean UX Hypothesis Statements

En esta sección se presentan las suposiciones clave que sustentan el diseño y desarrollo del producto. Estas suposiciones se basan en el entendimiento actual de las necesidades y comportamientos de los usuarios, así como en las condiciones del mercado. Identificar y documentar estas suposiciones permite al equipo alinearse en torno a expectativas comunes y establecer una base sólida para el proceso de validación a través de pruebas e iteraciones futuras.

Hypothesis Statements 1: Creemos que al implementar un sistema de monitoreo integrado, los dueños de casas de reposo y cuidadores verán una reducción del 30% en los incidentes de caídas y una mejora del 20% en la eficiencia operativa dentro de los primeros seis meses. Sabremos que esto es cierto cuando se observe una disminución documentada en los reportes de incidentes y una mejora en la satisfacción y eficiencia reportada por los usuarios.

Hypothesis Statements 2: Creemos que al utilizar nuestro sistema, los dueños de casas de reposo y cuidadores notarán una disminución del 30% en los incidentes de caídas y un 50% de mejora en el tiempo de respuesta a emergencias dentro del primer año. Sabremos que hemos tenido

éxito cuando se registren mejoras documentadas en los reportes de incidentes y tiempos de respuesta durante este período.

Hypothesis Statements 3: Creemos que al ofrecer una solución tecnológica que facilite la supervisión proactiva, al menos el 70% de los cuidadores profesionales y administradores adoptarán nuestro sistema en los primeros seis meses. Sabremos que esto es cierto cuando el 70% de los usuarios registrados utilicen activamente la solución y reporten una disminución significativa en la carga laboral.

Hypothesis Statements 4: Creemos que al usar nuestro sistema de monitoreo, los cuidadores y administradores de casas de reposo reportarán una mejora del 30% en su satisfacción laboral y una disminución del 20% en las tareas de supervisión repetitivas dentro del primer semestre. Sabremos que hemos tenido éxito cuando los resultados de encuestas de satisfacción y reportes de tiempo de trabajo lo reflejen, y al menos el 40% de los familiares indiquen una mayor percepción de seguridad.

Hypothesis Statements 5: Creemos que la integración de sensores de movimiento y signos vitales con alertas personalizables en tiempo real aumentará la adopción de nuestro sistema en un 25% del mercado objetivo durante el primer año. Sabremos que hemos tenido éxito cuando el 25% del mercado potencial (casas de reposo y cuidadores profesionales) hayan adquirido y utilicen activamente nuestra solución, reportando mejoras en la seguridad y satisfacción del cuidado.

1.2.2.4. Lean UX Canvas

Business Problem	Solution Ideas	Business Outcomes
Los cuidadores de personas mayores enfrentan dificultades para monitorear el estado de salud de sus pacientes en tiempo real. Los familiares y cuidadores necesitan una forma eficiente de ser notificados sobre emergencias, como caídas o problemas de salud, para brindar atención inmediata y prevenir complicaciones graves.	<p>Monitorear la temperatura y el ritmo cardíaco de las personas mayores en tiempo real.</p> <p>Recibir notificaciones inmediatas en caso de emergencias o valores fuera de rango.</p> <p>Permitir a los usuarios solicitar ayuda a través de un botón en el brazalete.</p> <p>Generar reportes automáticos con datos de salud y patrones a lo largo del tiempo.</p> <p>Acceso al sistema desde dispositivos móviles y web para cuidadores y familiares.</p>	<p>Mejorar la capacidad de respuesta ante emergencias en un 25%.</p> <p>Reducir el tiempo de respuesta a incidentes de salud en personas mayores en un 20%.</p> <p>Incrementar la satisfacción de los clientes mediante la integración de reportes y notificaciones automatizadas.</p>
User & Customers		User Benefits
<p>Usuarios: Cuidadores y familiares de personas mayores que necesitan monitoreo constante.</p> <p>Clientes: Clínicas geriátricas, residencias de ancianos, y familias con personas mayores que requieren monitoreo continuo.</p>		<p>Cuidadores: Mayor visibilidad en tiempo real del estado de salud de las personas mayores, mejor toma de decisiones basadas en datos y reducción en la necesidad de monitoreo físico constante.</p> <p>Personas Mayores: Mayor seguridad y tranquilidad al saber que están siendo monitoreados de manera constante, y facilidad para solicitar ayuda en caso de emergencia.</p>
Hypotheses	What's the most important thing we need to learn first?	What's the least amount of work we need to do to learn the most important thing?
<p>Monitoreo en Tiempo Real: Implementar un sistema de monitoreo en tiempo real reducirá los incidentes de salud no detectados en un 25% y mejorará la capacidad de respuesta de los cuidadores en un 20%.</p> <p>Notificaciones Instantáneas: Las notificaciones instantáneas mejorarán la capacidad de los cuidadores para actuar rápidamente, reduciendo los riesgos de complicaciones.</p> <p>Reportes Automáticos: La generación de reportes automáticos ayudará a los cuidadores y familiares a tomar decisiones mejor informadas sobre el estado de salud de las personas mayores.</p>	<p>Lo más importante que necesitamos aprender primero es: La efectividad del sistema de monitoreo en tiempo real para detectar y notificar problemas de salud en personas mayores.</p>	<p>Mínimo trabajo necesario: Desarrollar un prototipo básico del sistema de monitoreo que incluya monitoreo de temperatura, ritmo cardíaco y un botón de solicitud de ayuda. Realizar pruebas piloto en un entorno controlado para validar la efectividad del sistema y recopilar retroalimentación de los usuarios antes del lanzamiento completo.</p>

1.3. Segmentos objetivo

Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores

Nombre del Competidor	Descripción
Life Alert   https://www.lifealert.com/	Life Alert es una de las marcas más reconocidas en sistemas de alerta médica, famosa por su eslogan "¡He caído y no puedo levantarme!". Ofrece sistemas de emergencia diseñados para ayudar a personas mayores a solicitar asistencia rápidamente. Su principal dispositivo es un botón de emergencia que puede usarse en el hogar o fuera de él, enviando alertas a un centro de monitoreo disponible las 24 horas del día.

 https://www.mobilehelp.com/	MobileHelp es un proveedor líder de sistemas de alerta médica que ofrece dispositivos portátiles y basados en el hogar para personas mayores. Sus dispositivos están diseñados para alertar a los servicios de emergencia en caso de una caída u otro incidente. Además de las soluciones de emergencia en el hogar, MobileHelp destaca por su cobertura fuera de casa gracias a su integración con redes móviles, lo que ofrece a los usuarios mayor independencia y movilidad.
--	--

 CarePredict	CarePredict es una empresa tecnológica que se enfoca en el monitoreo preventivo para adultos mayores. Utiliza dispositivos portátiles equipados con sensores para rastrear patrones de comportamiento y signos vitales. A través de la inteligencia artificial, CarePredict predice posibles problemas de salud antes de que ocurran, como caídas o infecciones, y permite la intervención proactiva de cuidadores y familiares.
---	--

<https://www.carepredict.com/>

2.1.1. Análisis competitivo

Competitive Analysis Landscape					
¿Por qué llevar a cabo este análisis?	Comparar las características y funcionalidades clave de nuestra solución con las de la competencia para identificar ventajas competitivas y posibles áreas de mejora.				
Competidores	MIAM 	Life Alert  https://www.lifealert.com/	Carepredict 	MobileHelp 	
				https://www.mobilehelp.com/	https://www.carepredict.com/
Perfil	Overview	MIAM ofrece un avanzado sistema de monitoreo inteligente para la prevención	Life Alert es un sistema de alerta médica diseñado para proteger a las personas mayores en caso de emergencia sanitaria en el hogar, les permite ser independientes y enviar	CarePredict es una empresa de tecnología que ofrece soluciones de monitoreo para el cuidado de adultos mayores, usando inteligencia artificial y sensores. Su sistema predice cambios en el comportamiento	MobileHelp ofrece dispositivos médicos que permiten a los adultos mayores vivir de forma independiente mientras brindan tranquilidad a sus familiares, incluyen sistemas de alerta de emergencia médica que

	de accidentes en adultos mayores, a través del seguimiento de su movimiento y signos vitales de manera rápida y precisa, genera reportes de los datos del adulto mayor y proporciona un fácil monitoreo por su interfaz.	ayuda rápidamente en caso de una emergencia médica.	para prevenir problemas de salud como caídas o infecciones.	funcionan en el hogar y en exteriores.
Ventaja competitiva	Ofrece un enfoque detallado e íntegro para el monitoreo completo del adulto mayor evitando posibles accidentes.	Envía la ayuda que necesita rápidamente, las 24 horas del día, los 7 días de la semana.	Tecnología avanzada de sensores portátiles y el uso de inteligencia artificial que predice problemas de salud antes de que ocurran.	Combinación de dispositivos médicos avanzados con opciones de movilidad, es decir, sus sistemas no están limitados al hogar y ofrecen cobertura en cualquier lugar donde haya señal celular.
Perfil de Marketing	MIAM ofrece una solución completa para ayudar en el cuidado del adulto mayor, dirigidas a aquellos familiares que poseen familia con una edad avanzada y deseen un cuidado especializado para sus seres queridos.	Ofrece una solución especializada en la respuesta rápida ante los distintos accidentes que pueda sufrir un adulto mayor que se encuentre solo.	CarePredict se posiciona como una solución innovadora en el sector de salud para personas mayores, enfocándose en la prevención de problemas de salud a través de tecnología avanzada.	MobileHelp se posiciona como una solución accesible y fiable en el mercado de dispositivos de seguridad para personas mayores, suelen centrarse en la seguridad y tranquilidad que brindan a los usuarios.
Mercado objetivo				
Estrategias de marketing	Usará el marketing digital, realizará posts, campañas publicitarias y colaboración con influencers en	Campañas publicitarias en televisión y videos con testimonios de los adultos mayores a los que le ayudó el producto ante una emergencia.	Utilizan contenido educativo y testimonios de clientes para atraer a los usuarios. Además, aprovechan redes sociales y asociaciones con organizaciones de atención médica para ampliar su alcance.	Utilizan una estrategia multicanal, que incluye anuncios en televisión, redes sociales y asociaciones con proveedores de atención médica..

		internet para atraer a familiares que quieran un aparato que ayude en el cuidado de su adulto mayor.		
Productos & Servicios	Ofrecemos un plan de suscripción mensual de bajo costo para el plan básico o un plan de suscripción personalizado para el plan empresarial	Ofrece 3 productos para la respuesta rápida ante una emergencia, "help button", "micro voice pendant system" y "on-the-go + GPS".	Ofrecen sensores portátiles que monitorean las actividades diarias y detectan cambios en los patrones de comportamiento.	MobileHelp ofrece sistemas de alerta médica, monitores de salud, y accesorios como sensores de caídas y relojes inteligentes con capacidades de monitoreo.
Perfil de Producto	Ofrecemos un plan de suscripción mensual de bajo costo para el plan básico o un plan de suscripción personalizado para el plan empresarial.	Entregan el producto después de realizar una llamada al número proporcionado en su página web.	El modelo de precios se basa en suscripciones mensuales por el uso de los dispositivos y el acceso a la plataforma de monitoreo.	Sus precios varían según el tipo de dispositivo y plan de servicio, generalmente en formato de suscripción mensual.
Canales de distribución	Ventas directas a partir de los canales de la aplicación web y móvil.	Ventas directas y personalizadas para cada llamada al número proporcionado por la página web.	Acceso a través de una plataforma web y aplicaciones móviles que permiten a los cuidadores y familiares monitorear la salud de los adultos mayores en tiempo real.	MobileHelp distribuye sus productos principalmente a través de su sitio web y socios minoristas.

Análisis SWOT

Fortalezas Tecnología IoT avanzada para un monitoreo preciso del adulto mayor, acompañado de aplicativo móvil y web intuitivo con una gran personalización y proporciona información recolectada. Enfoque centrado en la respuesta rápida ante los accidentes del adulto mayor, solución sencilla y simple pero eficiente. Tecnología avanzada, predicción proactiva de problemas de salud. Amplia cobertura fuera del hogar, sistemas fáciles de usar y monitoreo 24/7. Debilidades Posible alta inversión en el área de investigación y desarrollo, necesidad de establecerse en un mercado altamente competitivo. Limitación en solo enviar alertas cuando el botón sea presionado, posible dependencia de su reputación. Dependencia de la tecnología, lo que puede limitar la adopción por parte de usuarios menos tecnológicos. Dependencia de las redes celulares para algunas funciones, lo que puede limitar la cobertura en áreas rurales. Oportunidades Creciente población de adultos mayores en el mundo, promedio de vida en alza, mejoras en la respuesta de salud, demanda de soluciones tecnológicas en el área y expansión a mercados globales. Expansión del área de salud y cuidado de adultos mayores, promedio de vida de las personas más alta, expansión en mercados globales. Creciente demanda de soluciones de cuidado para personas mayores. Creciente demanda de soluciones de salud para la población envejecida. Amenazas Competencia fuerte de empresas ya establecidas y consolidadas en el campo, cambios en las tendencias tecnológicas que puedan cambiar o afectar a la demanda del servicio. Competencia de otras soluciones más completas en el mercado, creciente avance de la tecnología puede dejar anticuado el simple sistema. Competencia de otros dispositivos de monitoreo y avances en tecnología. Competencia creciente de dispositivos de alerta médica portátiles y cambios en las regulaciones del sector.

2.1.2. Estrategias y tácticas frente a competidores

OPORTUNIDADES

AMENAZAS

- Creciente población de adultos mayores a nivel mundial.
- Aumento en la demanda de soluciones tecnológicas para el cuidado de la salud.
- Expansión global hacia mercados con alta demanda de productos de salud y monitoreo.
- Mejoras en la expectativa de vida y respuesta de salud preventiva.
- Competencia fuerte de empresas establecidas en el mercado.
- Rápido cambio tecnológico, lo que puede requerir actualizaciones constantes.
- Baja barrera de entrada para nuevos competidores en el sector de monitoreo.
- Cambios en las tendencias de salud y políticas que afecten la demanda del servicio.

FORTALEZAS ESTRATEGIAS FO (Ofensivas) ESTRATEGIAS FA (Defensivas)

- Tecnología IoT avanzada para el monitoreo preciso de adultos mayores.
- Aplicación web y móvil intuitiva con gran personalización.
- Capacidad de monitoreo en tiempo real de signos vitales y movimiento.
- Generación de reportes detallados para cuidadores y familiares.
- F1 + O1/O2: Aprovechar la avanzada tecnología IoT para expandirse en el mercado global de cuidado de adultos mayores, resaltando las capacidades de monitoreo proactivo en campañas dirigidas a mercados en crecimiento.
- F3 + O3: Utilizar el monitoreo en tiempo real y los reportes personalizados para formar alianzas estratégicas con hospitales y centros de salud, ingresando a nuevos mercados internacionales donde la demanda de tecnología de salud está en aumento.
- F2 + A1: Usar la ventaja de la interfaz intuitiva para diferenciarse de competidores más tradicionales, enfocando las campañas de marketing en la facilidad de uso y el soporte continuo que MIAM ofrece a las familias.
- F4 + A3: Generar innovaciones en los reportes y análisis de datos para mantenerse a la vanguardia frente a nuevos competidores que puedan entrar al mercado. Añadir funcionalidades basadas en inteligencia artificial para mejorar la calidad del servicio.

DEBILIDADES ESTRATEGIAS DO (Reorientación) ESTRATEGIAS DA (Supervivencia)

- Alta inversión en I+D para mantenerse competitivos.
- Falta de reconocimiento frente a competidores ya establecidos.
- Necesidad de una estrategia sólida de penetración de mercado.
- Dependencia de la tecnología y conectividad para un monitoreo eficiente.
- D1 + O2: Invertir en investigación y desarrollo de manera escalonada, comenzando con mejoras en los productos más demandados, para capitalizar el aumento en la demanda de soluciones tecnológicas para adultos mayores.
- D3 + O3: Desarrollar una estrategia de marketing digital sólida para aumentar la penetración en el mercado, aprovechando la tendencia creciente de la atención a adultos mayores para ganar visibilidad frente a competidores más grandes.
- D2 + A1/A2: Trabajar en la mejora del reconocimiento de marca a través de colaboraciones con influencers y campañas publicitarias que resalten las características distintivas de MIAM, mitigando la fuerte competencia y la posible obsolescencia tecnológica.
- D4 + A4: Implementar sistemas de monitoreo basados en la nube que puedan adaptarse fácilmente a cambios tecnológicos y políticas, reduciendo la dependencia de la infraestructura tecnológica y asegurando una respuesta ágil a cualquier cambio en las tendencias de salud.

2.2. Entrevistas

2.2.1. Diseño de entrevistas

Segmento 1: Cuidadores de adultos mayores (Caregivers)

- ¿Cuál es su nombre?
- ¿Cuántos años tiene y cuál es su estado civil?
- ¿En qué distrito vive?
- ¿Cuál es su profesión?
- ¿Qué dispositivos tecnológicos utiliza regularmente?
- ¿Hay alguna marca de dispositivos que prefiera o utilice más? ¿Por qué?
- ¿Usa alguno para el monitoreo de los adultos mayores? (Ej. pulseras inteligentes, cámaras de vigilancia, etc.)
- ¿Qué medios o redes sociales utiliza para informarse sobre temas relacionados a su trabajo o comunicarse? (Ej. WhatsApp, Facebook, foros especializados, etc.)
- ¿Se considera una persona que se adapta fácilmente a las nuevas tecnologías?
- ¿En su labor, prefiere seguir procedimientos establecidos o es más de buscar soluciones tecnológicas?
- ¿Cuáles son los principales desafíos que enfrenta al monitorear y cuidar a un adulto mayor?
- ¿Qué aspectos de su trabajo le generan más satisfacción o facilitan su labor?
- ¿Podría describir cómo es un día típico cuidando a un adulto mayor, en particular en relación con la prevención de caídas?
- ¿Qué pasos sigue o qué herramientas utiliza para asegurarse de que el adulto mayor esté seguro en todo momento?
- Si pudiera diseñar una herramienta para ayudarle en el monitoreo de adultos mayores, ¿qué características considera que serían imprescindibles?
- ¿Qué funciones adicionales le gustaría que tuviera esta herramienta?

Segmento 2: Dueños de casas de reposo (Nursing Home Owners)

- ¿Cuál es su nombre?

- ¿Cuántos años tiene y cuál es su estado civil?
- ¿En qué distrito vive?
- ¿Cuál es su profesión y su rol dentro de la casa de reposo?
- ¿Qué dispositivos tecnológicos utiliza regularmente?
- ¿Hay alguna marca de dispositivos que prefiera o utilice más? ¿Por qué?
- ¿Utiliza actualmente alguna herramienta para el monitoreo de los adultos mayores residentes? (Ej. sistemas de alarma, sensores, cámaras, etc.)
- ¿Qué medios o redes sociales utiliza para estar al tanto de información y comunicarse? (Ej. LinkedIn, WhatsApp , etc.)
- ¿Cómo se describe en cuanto a la adopción de nuevas tecnologías en su establecimiento?
- ¿Estaría interesado en invertir en tecnología de punta para el monitoreo de los residentes? ¿Por qué?
- ¿Cuáles son los principales desafíos que enfrenta al gestionar la seguridad y bienestar de los residentes?
- ¿Qué mejoras o innovaciones le han resultado más útiles en la gestión de la casa de reposo?
- ¿Podría describir el proceso que sigue en un día típico de trabajo?
- ¿Qué protocolos o herramientas son esenciales en su día a día para garantizar la seguridad y prevenir caídas de los residentes?
- Si pudiera integrar un nuevo sistema de monitoreo en su casa de reposo, ¿qué características serían imprescindibles para usted?
- ¿Qué otras funciones le gustaría que tuviera este sistema para mejorar la seguridad y de los residentes?

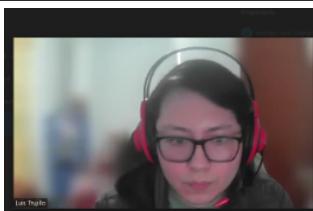
2.2.2. Registro de entrevistas

Segmento 1: Cuidadores de adultos mayores (Caregivers)

<p>Nombres y apellidos: Katty Salgado Edad: 29 años Distrito: Los Olivos Timing: 0:00 - 10:06 minutos Duración: 10:06 minutos</p>	
<p>Enlace: https://goo.su/Ao2Pm</p> <p>Resumen: Katty es una enfermera que se dedica al cuidado de los adultos mayores. En la entrevista se muestra como una persona amable y empática. Ella comenta que sus dispositivos más usados son su celular, su tablet y laptop, los cuales utiliza para comunicarse y gestionar sus horarios de trabajo. Asimismo, sus marcas preferidas son las de sus dispositivos de uso diario: Xiaomi y Apple. No utiliza dispositivos tecnológicos para el cuidado de los adultos mayores, pero ha identificado que en algunos hogares las familias utilizan cámaras para monitorearlos. Las redes sociales que más usa son Whatsapp, Facebook e Instagram y considera que se adapta a las nuevas tecnologías fácilmente. Dentro de su rutina de trabajo realiza funciones como la toma de signos vitales, aseo del adulto mayor, tendido de cama y apoyo con la alimentación. Katty considera que uno de los desafíos en su trabajo es evitar las caídas de los adultos mayores, por lo cual tiene que ayudarlos constantemente a movilizarse. Por otra parte, lo que le causa más satisfacción es el agradecimiento del paciente y su familia. Considera que un sistema de monitoreo debería ayudarle a monitorear los signos vitales (pulso, temperatura, saturación de oxígeno, etc.) y prevenir las caídas. También considera útil el envío de notificaciones cuando alguno de los signos vitales esté alterado y recordatorios del tratamiento (medicación) del paciente.</p>	
<p>Nombres y apellidos: Sara Villanueva Edad: 39 años Distrito: Callao Timing: 10:06 - 13:42 minutos Duración: 3:36 minutos</p>	

Enlace: <https://goo.su/Ao2Pm>

Resumen: Sara es una enfermera que trabaja cuidando adultos mayores. Se muestra como una persona seria, concisa y práctica, adaptada al uso de la tecnología en su trabajo. El dispositivo que más utiliza en su día a día es su celular, ya que le ayuda a crear alarmas para administrar a tiempo el tratamiento a sus pacientes. Su marca favorita es Motorola. Sara considera que se adapta a la tecnología y le gusta incorporar soluciones tecnológicas en su labor. Su medio de comunicación más frecuente es a través de Whatsapp y también utiliza Instagram y TikTok. Uno de los principales desafíos que enfrenta al cuidar a los adultos mayores son las caídas, en especial en aquellos que no son independientes, y algo que le causa satisfacción es el agradecimiento y cariño que le dan sus pacientes. Con el fin de evitar caídas realiza procedimientos como colocar almohadas y ubicarlos en una posición adecuada, además de estar siempre pendiente todo el tiempo de ellos. Ella considera que un sistema de monitoreo del adulto mayor debe incluir cámaras para supervisarlos y el envío de alertas al cuidador.

Nombres y apellidos: William Ramos Edad: 22 años Distrito: Jesus Maria Timing: 13:42 - 22:03 minutos Duración: 8:21 minutos		
---	--	---

Enlace: <https://goo.su/Ao2Pm>

Resumen: William Ramos es una persona que terminó su carrera profesional y se dedica a trabajar de forma parcial en el cuidado de su vecina que es una adulta mayor. Es un joven extrovertido y práctico que muestra preocupación por el cuidado del adulto a su cargo. Los dispositivos que más utiliza son su computadora y celular, siendo sus marcas preferidas Samsung, Xiaomi y Asus. Las redes que más usa son Instagram y Whatsapp. Menciona que es una persona que se relaciona mucho con la tecnología a diferencia del adulto mayor al que se encarga de cuidar. Comenta que es una tarea un tanto complicada el hecho de cuidar al adulto mayor por la constante supervisión que se debe hacer y la preocupación de que pueda pasar algo. Menciona que en ocasiones el adulto mayor que cuida le gusta ser independiente y hacer varias acciones por su cuenta, le parece llamativo una solución que le permita monitorear si la persona que cuida se encuentra en perfecto estado y en qué ubicación está, ya que de esa manera podría darle autonomía sin descuidar el correcto cuidado que debe realizar. Comenta que le interesa una aplicación que permita saber el estado del adulto mayor y que permita al adulto mayor dar mensajes ante cualquier emergencia.

Segmento 2: Dueños de casas de reposo (Nursing Home Owners)

Nombres y apellidos: Gabriel Hachamizo Edad: 27 años Distrito: Los Aquijes - Ica Timing: 22:03 - 28:14 minutos Duración: 06:11 minutos	
---	--

Enlace: <https://goo.su/Ao2Pm>

Resumen: Gabriel es un licenciado en enfermería que trabaja en una casa de reposo para adultos mayores, desempeñando un rol administrativo. Es una persona organizada y práctica que busca mejorar la eficiencia en su labor. En la entrevista, comentó que sus dispositivos más utilizados son celulares y laptops y su marca preferida es Lenovo, ya que nunca ha tenido imprevistos con sus dispositivos. Para el monitoreo de los pacientes, se utilizan cámaras de seguridad, mientras que la comunicación con los familiares se realiza principalmente a través de WhatsApp. Facebook se emplea para promocionar los servicios de la casa de reposo. Gabriel considera que la adopción de nuevas tecnologías es positiva y puede mejorar el cuidado de los pacientes. Sin embargo, enfrenta desafíos como el manejo de una gran cantidad de pacientes y la insuficiencia del protocolo de atención para garantizar una atención adecuada. También describe su rol en la empresa, incluyendo la priorización de actividades según su importancia. Además, Gabriel sugiere el desarrollo de un software para optimizar la administración y el registro de pacientes, así como para recopilar datos que contribuyan a una mejor atención.

Nombres y apellidos: Miriam Guevara Edad: 54 Distrito: Trujillo Timing: 28:14 - 36:29 minutos Duración: 7:54 minutos	
---	--



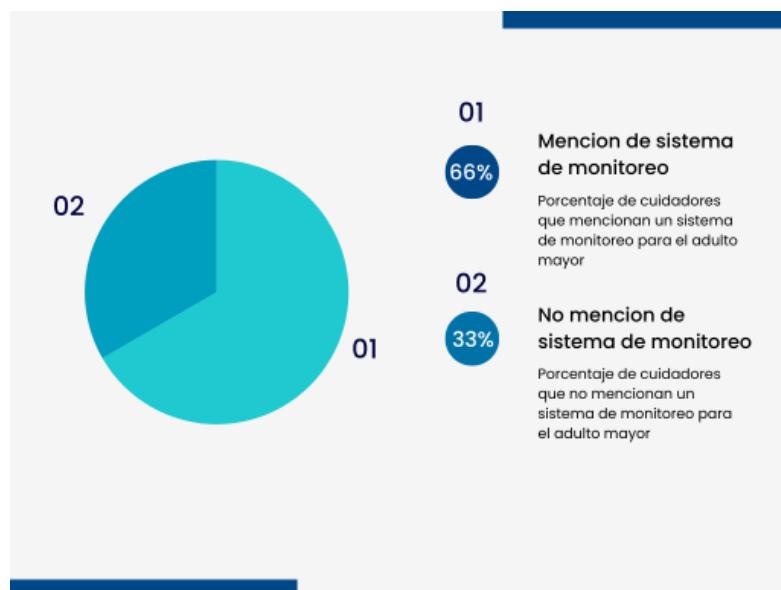
Enlace: <https://goo.su/Ao2Pm>

Resumen: Miriam es una enfermera que ha puesto su casa de reposo para adultos mayores llamdo Divino Amor. Se muestra como una persona empática, amable y extrovertida que valora la organización. Durante la entrevista, comenta que utiliza con frecuencia su celular y laptop, siendo sus marcas favoritas Apple y Lenovo. El canal que más usa para comunicarse es a través de WhatsApp. Considera que la tecnología es importante, así que en su casa de reposo utiliza cámaras de seguridad para el monitoreo de los pacientes. Sin embargo, comenta que para los aduluos mayores es complicado adaptarse. Por el momento, ellos disponen de una pequeña campana que hacen sonar cuando necesitan ayuda de sus cuidadores. Uno de los desafíos que identifica es el monitoreo constante del adulto mayor, para lo cuál las cámaras han sido de ayuda. La labor del personal inicia antes de que los pacientes se levanten por la mañana, ya que deben organizar qué actividades realizarán y verificar los medicamentos que administrarán a los adultos mayores. Considera que lo mas dificil es tratar a todos los adultos mayores al mismo tiempo, sobre todo cuando despiertan.

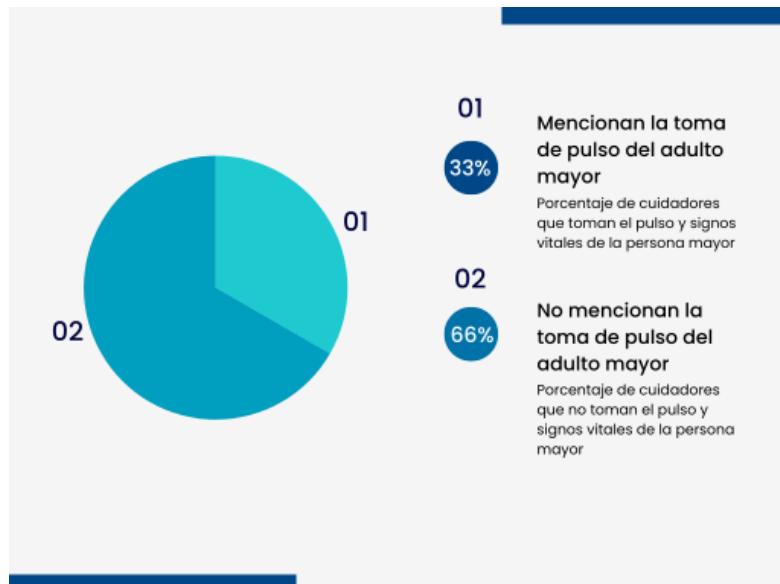
2.2.3. Análisis de entrevistas

Segmento Objetivo -> Cuidadores:

- Características Objetivas:



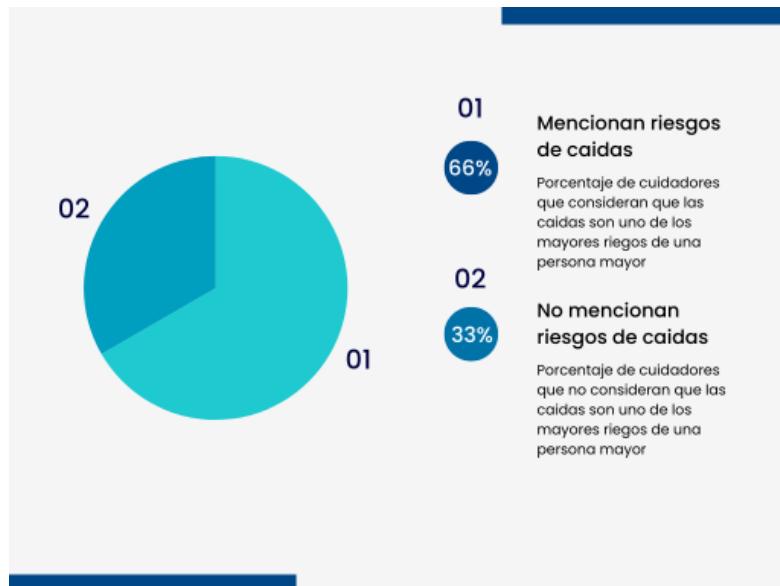
La gráfica muestra el porcentaje de cuidadores que han adoptado un sistema de monitoreo para personas mayores, comparado con aquellos que no lo han hecho. Se observa que una gran mayoría de cuidadores ya utiliza algún tipo de tecnología para supervisar la salud y seguridad de los adultos mayores, lo que sugiere una tendencia creciente hacia la digitalización y automatización en el cuidado de la tercera edad.



Indica el porcentaje de cuidadores que realizan un monitoreo activo de los signos vitales, como el pulso, de los adultos mayores bajo su cuidado, frente a aquellos que no realizan estas mediciones. Esto refleja la importancia que se le da al seguimiento de la salud en tiempo real para prevenir posibles complicaciones y mantener un control constante de la condición física del adulto mayor.

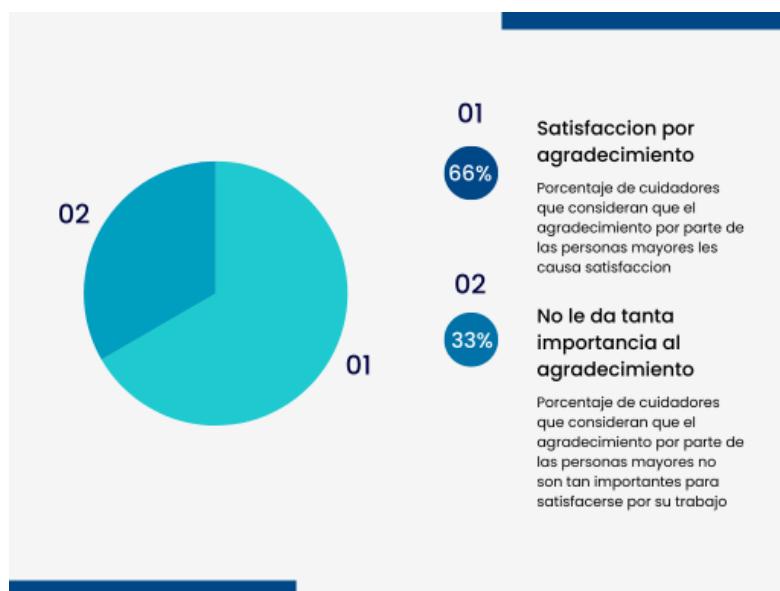


Muestra el porcentaje de cuidadores que enfrentan desafíos significativos en su labor diaria. Los desafíos más comunes incluyen la falta de tiempo, el estrés emocional y la necesidad de equilibrar el cuidado con otras responsabilidades personales. Esta gráfica resalta la necesidad de brindar más apoyo y recursos a los cuidadores.

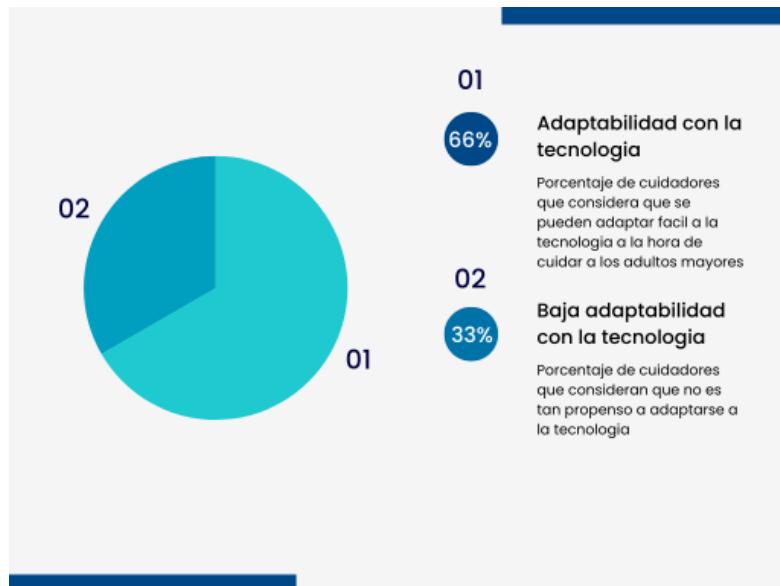


Representa el porcentaje de cuidadores que consideran las caídas como uno de los mayores riesgos para las personas mayores, en comparación con aquellos que no le dan tanta importancia a este factor. Las caídas son una causa principal de hospitalizaciones y lesiones graves en adultos mayores, por lo que es crucial implementar medidas preventivas.

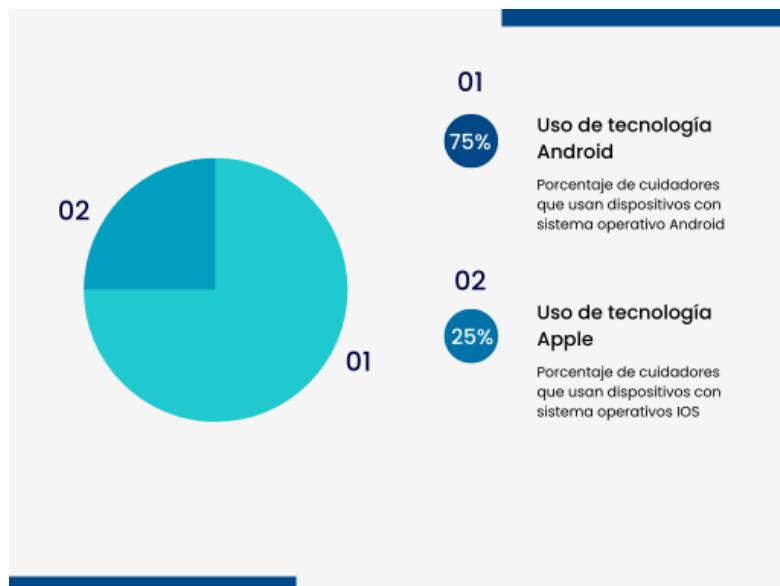
- Características Subjetivas:



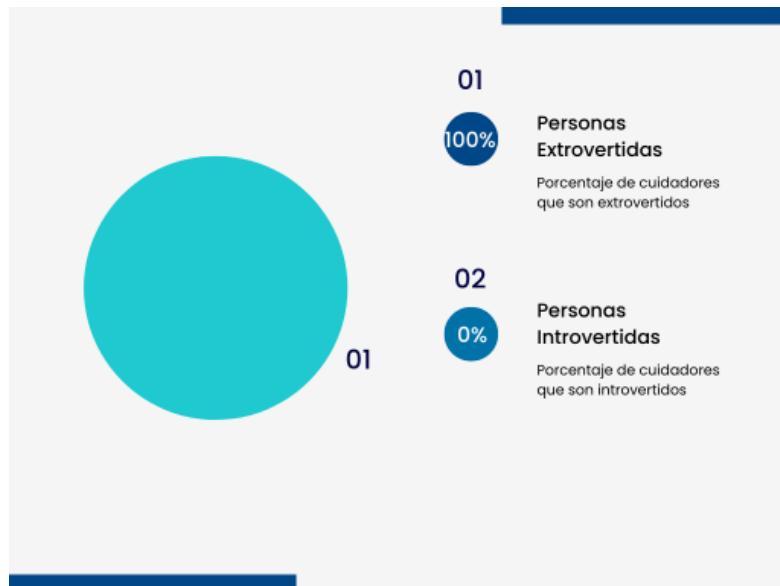
Compara la satisfacción personal de los cuidadores que reciben gratitud por parte de los adultos mayores, frente a aquellos que no experimentan este tipo de reconocimiento. La gratitud tiene un impacto positivo en la motivación y bienestar emocional del cuidador, subrayando la importancia del reconocimiento y apoyo en su rol.



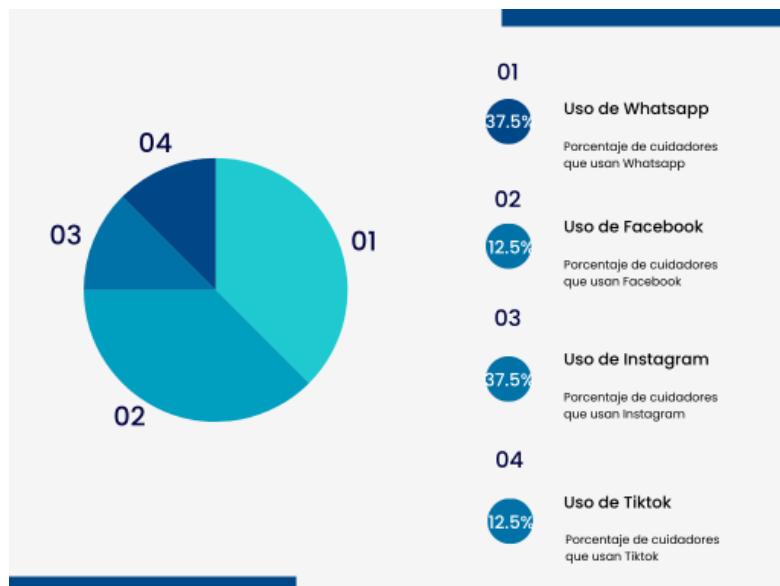
Refleja la capacidad de adaptabilidad de los cuidadores hacia el uso de tecnología en el cuidado de adultos mayores. Aquellos con alta adaptabilidad suelen sentirse más cómodos utilizando dispositivos y aplicaciones para monitorear la salud, mientras que los de baja adaptabilidad pueden requerir capacitación adicional para integrar la tecnología en su rutina.



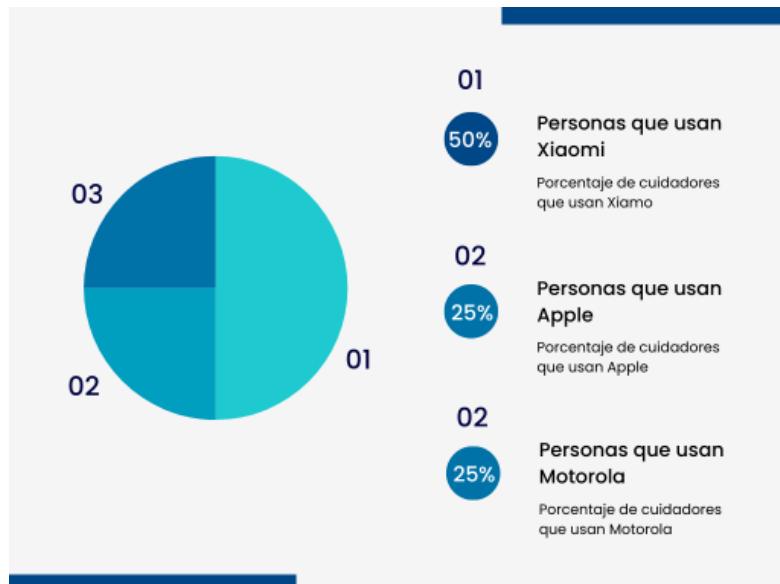
Muestra la preferencia en el uso de sistemas operativos móviles entre los cuidadores, comparando el uso de Android frente a iOS. Esta preferencia puede influir en la adopción de ciertas aplicaciones o herramientas de monitoreo, y es importante considerarlo al desarrollar soluciones tecnológicas para el cuidado.



Compara la personalidad de los cuidadores, destacando la proporción de extrovertidos frente a introvertidos. Los extrovertidos pueden encontrar más satisfacción en roles que implican interacción constante con los adultos mayores y otros cuidadores, mientras que los introvertidos pueden preferir tareas más orientadas al seguimiento y administración.



Indica el uso de diferentes redes sociales entre los cuidadores, incluyendo WhatsApp, Facebook, Instagram y TikTok. Esta información es útil para diseñar estrategias de comunicación y apoyo, así como para entender las preferencias de interacción social de los cuidadores en plataformas digitales.



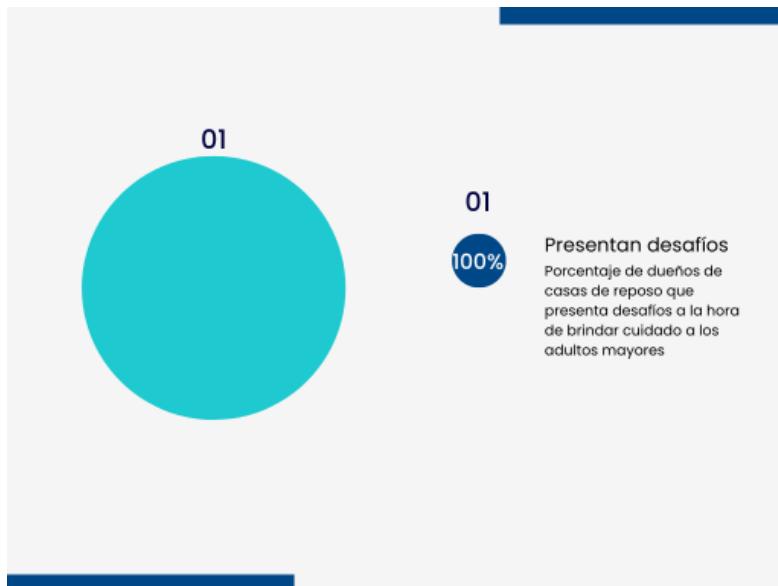
Compara la preferencia por marcas de dispositivos móviles, mostrando una alta presencia de marcas como Xiaomi, Apple y Motorola. Este dato es relevante para el desarrollo de aplicaciones móviles, asegurando compatibilidad y optimización en las marcas más utilizadas por los cuidadores.

Segmento Objetivo -> Dueños de casas de reposo:

- Características Objetivas:



Muestra el uso de cámaras de monitoreo en casas de reposo, resaltando la importancia de la vigilancia para la seguridad de los residentes. Los propietarios de casas de reposo que implementan sistemas de cámaras suelen tener un mejor control y respuesta ante situaciones de emergencia.



Indica los principales desafíos que enfrentan los dueños de casas de reposo, como la gestión eficiente del personal y la implementación de tecnología para mejorar la atención. Este gráfico destaca la necesidad de soluciones tecnológicas que simplifiquen la administración y mejoren la calidad del servicio.



Resalta el interés de los dueños de casas de reposo en implementar tecnología para mejorar sus servicios. La mayoría está abierta a adoptar nuevas herramientas que les permitan monitorear mejor a los residentes y optimizar los recursos disponibles.

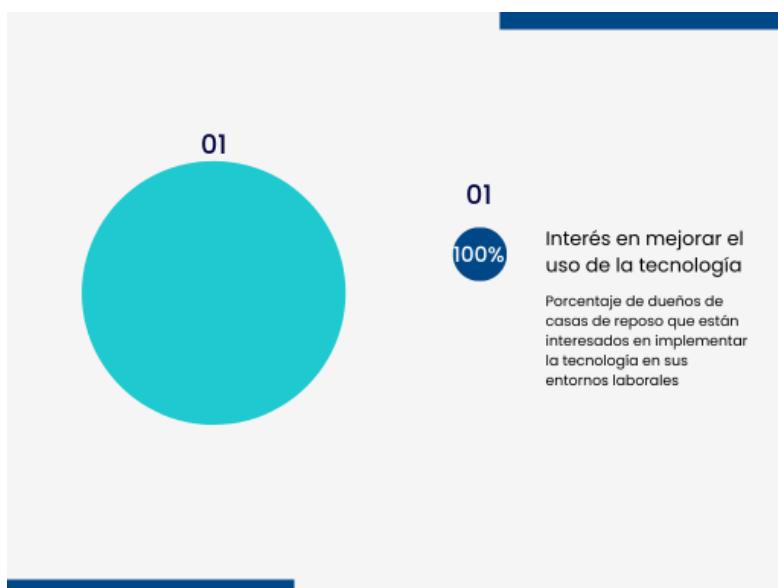
- Características Subjetivas:



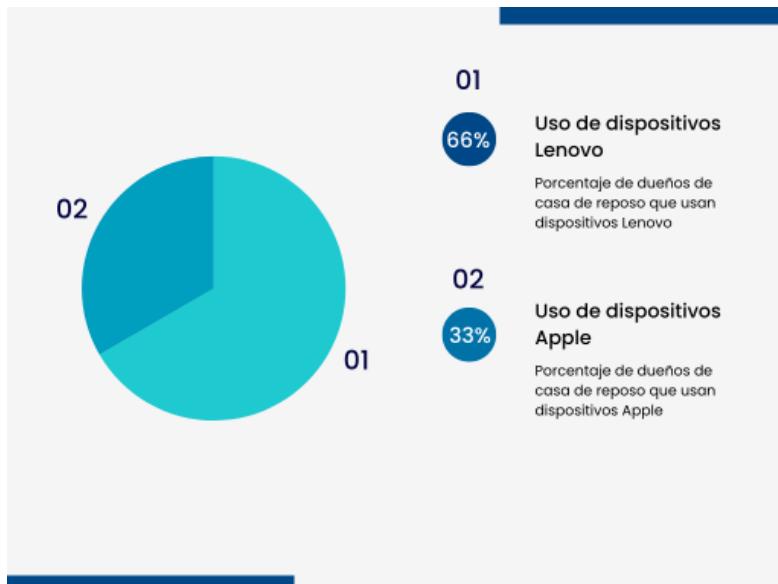
Muestra el nivel de satisfacción de los dueños de casas de reposo al brindar cuidado a los adultos mayores. La satisfacción personal es un indicador de la calidad del servicio y la motivación para continuar mejorando el entorno y la atención brindada.



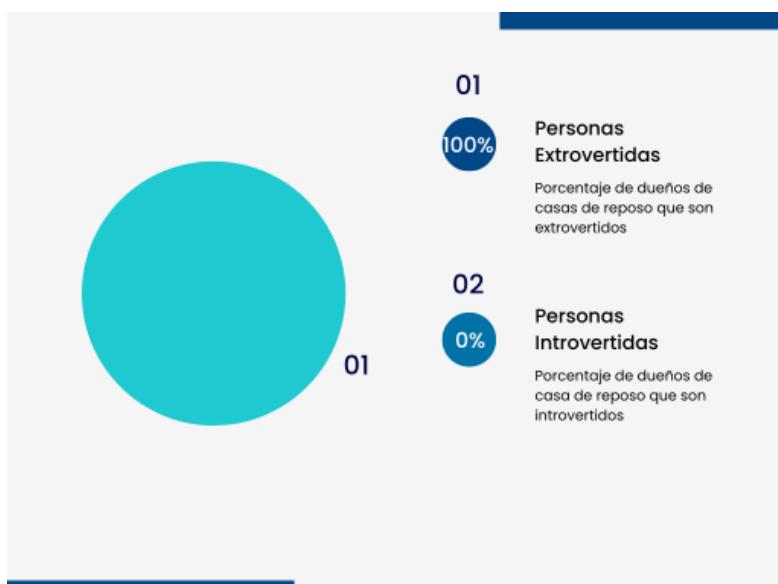
Indica el grado de adaptabilidad a nuevas tecnologías de los dueños de casas de reposo. Aquellos con alta adaptabilidad suelen estar más dispuestos a implementar sistemas avanzados de monitoreo y gestión en sus instalaciones.



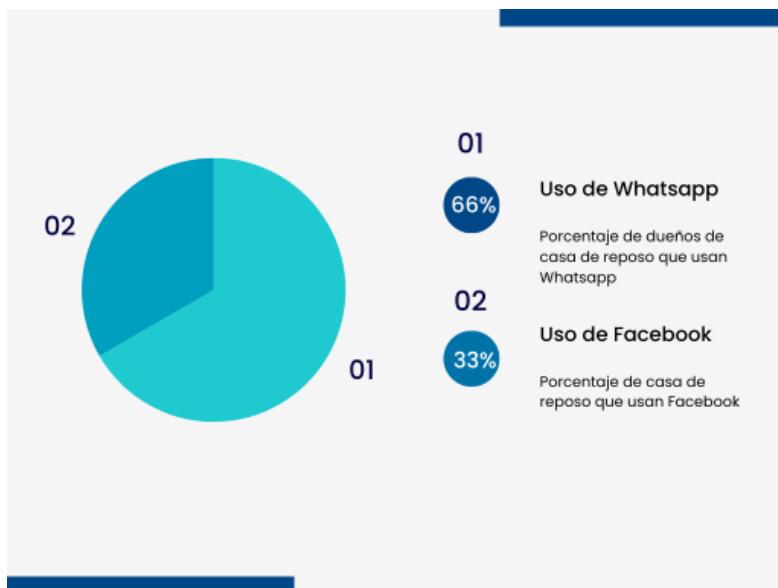
Refleja el interés en mejorar el uso de la tecnología en el entorno laboral de las casas de reposo. La mayoría busca soluciones que integren funciones como monitoreo de salud, administración de medicamentos y comunicación con familiares.



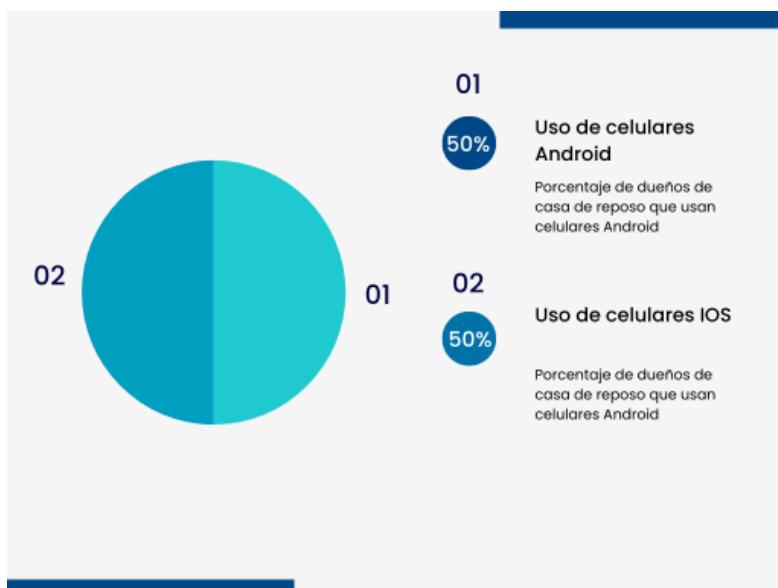
Compara la preferencia por dispositivos Lenovo y Apple entre los dueños de casas de reposo. Esta información puede influir en la decisión de adquirir herramientas tecnológicas compatibles y optimizadas para los dispositivos más utilizados.



Muestra la proporción de dueños de casas de reposo que son extrovertidos frente a introvertidos. Los extrovertidos tienden a liderar con un enfoque más comunicativo y abierto, mientras que los introvertidos pueden centrarse más en la organización interna y la eficiencia operativa.



Indica el uso de redes sociales como WhatsApp y Facebook entre los dueños de casas de reposo. Estas plataformas son clave para la comunicación con familiares y el equipo de trabajo, facilitando el intercambio de información y la coordinación.



Compara el uso de celulares Android frente a iOS entre los dueños de casas de reposo. Esta preferencia tecnológica puede afectar la adopción de aplicaciones y sistemas de gestión diseñados para dispositivos móviles, destacando la importancia de crear soluciones multiplataforma.

2.3. Needfinding

2.3.1. User Personas

Segmento 1: Cuidadores de adultos mayores (Caregivers)

Ana Martínez es una enfermera de 32 años que trabaja en Lima, Perú, especializada en el cuidado de adultos mayores. Ella es una cuidadora comprometida que se siente motivada por el bienestar y la seguridad de sus pacientes, y siempre busca ofrecer el mejor servicio posible. Su enfoque está en mejorar la eficiencia en el cuidado de sus pacientes, a pesar de enfrentar desafíos como las caídas y la falta de tecnología para facilitar su trabajo. A Ana le interesa incorporar nuevas tecnologías en su rutina para monitorear signos vitales de manera más eficiente y se siente frustrada por la dependencia de métodos manuales. Utiliza canales de comunicación como WhatsApp e Instagram para conectarse con los demás y sigue marcas tecnológicas como Xiaomi.

PERSONA: Ana Martinez

NAME	TYPE		
Ana Martinez	Guardian		
			
Quote  Lo más importante es asegurarme de que mis pacientes estén seguros y cómodos 	Goals <ul style="list-style-type: none"> Mejorar la eficiencia en el cuidado de sus pacientes Ofrecer el mejor servicio posible Frustrations <ul style="list-style-type: none"> Le resulta difícil prevenir caídas y mantener la seguridad de los pacientes Depende de métodos manuales para monitorear los signos vitales de sus pacientes Frustrada por la falta de tecnología que facilite su trabajo 	Motivations <p>Incentive</p>  <p>Fear</p>  <p>Achievement</p>  <p>Growth</p>  <p>Social</p> 	
Demographic <p>Femenino</p> <p>32 años</p> <p>Lima, Perú</p> <p>Cuidador de Adultos Mayores</p>	Background <p>Ana Martinez es una enfermera y trabaja en el cuidado de adultos mayores. Se adapta bien a las nuevas tecnologías y está interesada en incorporarlas en su trabajo. Su día a día incluye tomar signos vitales, asear a los pacientes y prevenir caídas. Se siente motivada por el cariño y agradecimiento que recibe de los pacientes y sus familias.</p>	Channels <p> WhatsApp  Instagram</p>	Brands and influencers <p></p>
Technology <p>   </p>			

UXPRESSIA
This persona was built in uxpressia.com

Segmento 2: Dueños de casas de reposo (Nursing Home Owners)

Gabriel Lopez es un enfermero de 40 años que dirige una casa de reposo para adultos mayores en Lima, Perú. Como dueño y administrador, su principal objetivo es mejorar la eficiencia en la atención a los pacientes a través de un sistema que le permita gestionar el estado de estos de manera más efectiva. Aunque tiene una amplia experiencia en la gestión de personal y pacientes, uno de sus mayores retos es atender a todos sus pacientes simultáneamente, especialmente durante los momentos de mayor actividad. Gabriel valora profundamente la tecnología y busca constantemente formas de integrarla para optimizar tanto la calidad de los cuidados como la administración en su lugar de trabajo. Utiliza WhatsApp como su principal canal de comunicación y se siente influenciado por marcas como Lenovo.

PERSONA: Gabriel Lopez

NAME	TYPE
Gabriel Lopez	Guardian
	
Goals	Motivations
<ul style="list-style-type: none"> Mejorar la eficiencia en la atención a los pacientes Tener un sistema que le permita gestionar el estado de los pacientes 	<p>Incentive</p>  <p>0 25 50 75 100</p> <p>Fear</p>  <p>0 25 50 75 100</p> <p>Achievement</p>  <p>0 25 50 75 100</p> <p>Growth</p>  <p>0 25 50 75 100</p> <p>Social</p>  <p>0 25 50 75 100</p>
Quote	Background
<p>“Mi mayor satisfacción es ver a nuestros pacientes felices y bien cuidados”</p>	<p>Gabriel es un licenciado en enfermería que tiene una casa de reposo para adultos mayores. Su experiencia laboral le ha permitido reconocer la importancia de la tecnología en la mejora del cuidado y la administración en su lugar de trabajo.</p> <p>En su día a día, Gabriel se enfrenta a la tarea de gestionar un gran número de pacientes y de coordinar las actividades del personal, lo que le lleva a buscar constantemente formas de mejorar la calidad de su servicio.</p>
Demographic	Brands and influencers
Masculino 40 años Lima, Perú Dueño de casa de reposo	
Technology	Channels
	 WhatsApp

UXPRESSIA
This persona was built in uxpressia.com

2.3.2. User Task Matrix

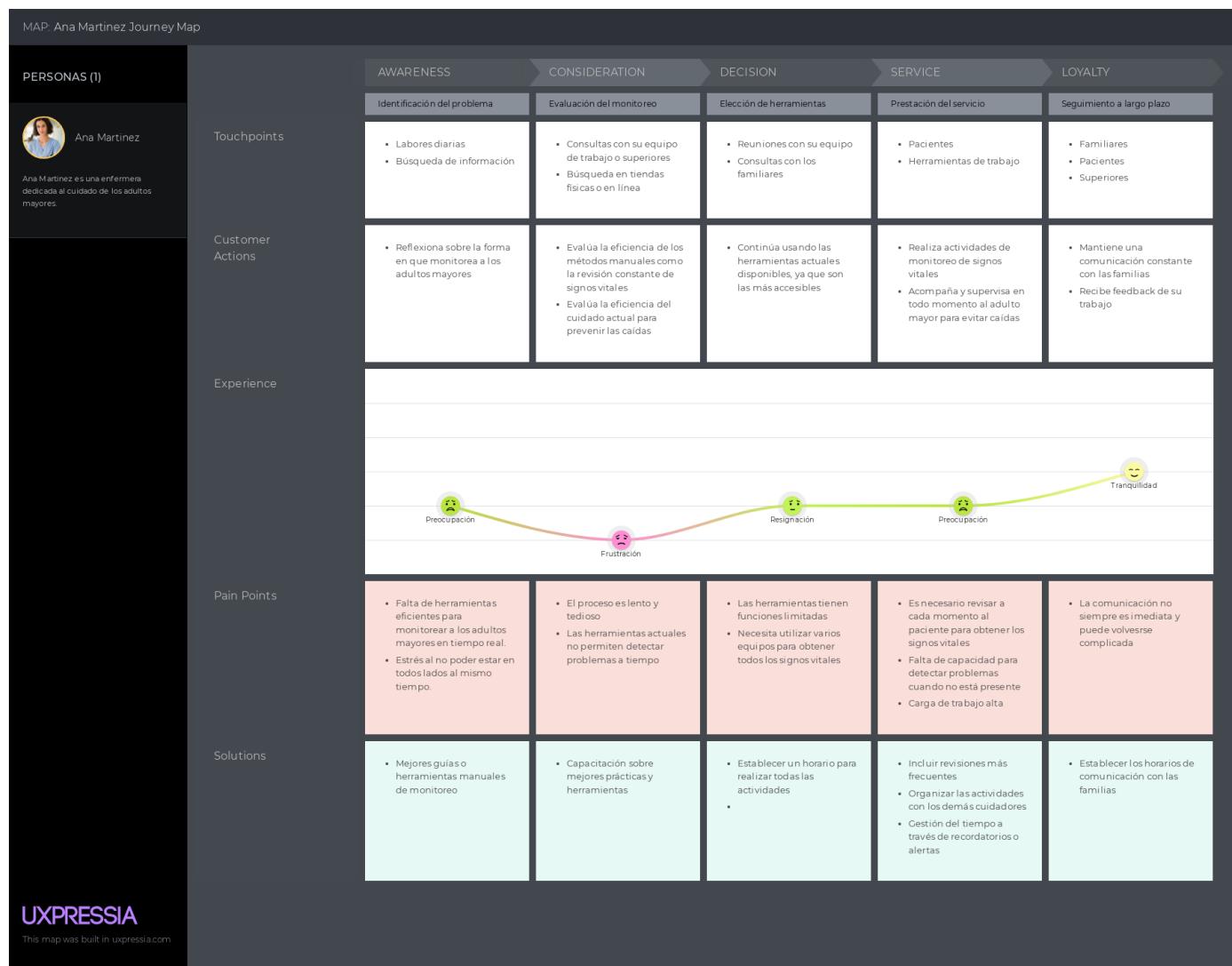
El User Task Matriz presenta las actividades en común que más realizan los usuarios Ana Martinez y Gabriel Lopez. Entre las más importantes se encuentran la supervisión del estado de los adultos mayores, gestión de los cuidados y medicación y la revisión de actividades diarias. Esto sugiere que tienen prioridades similares en su trabajo.

User Task	Ana Martinez	Gabriel Lopez		
	Frequency	Importance	Frequency	Importance
Supervisión del estado de los pacientes	SIEMPRE	ALTA	SIEMPRE	ALTA
Comunicación y coordinación con familias de los pacientes	USUALMENTE	ALTA	USUALMENTE	MEDIA
Gestión de cuidados y medicación de los pacientes	SIEMPRE	ALTA	SIEMPRE	ALTA
Revisar y priorizar actividades diarias	SIEMPRE	ALTA	SIEMPRE	ALTA

2.3.3. User Journey Mapping

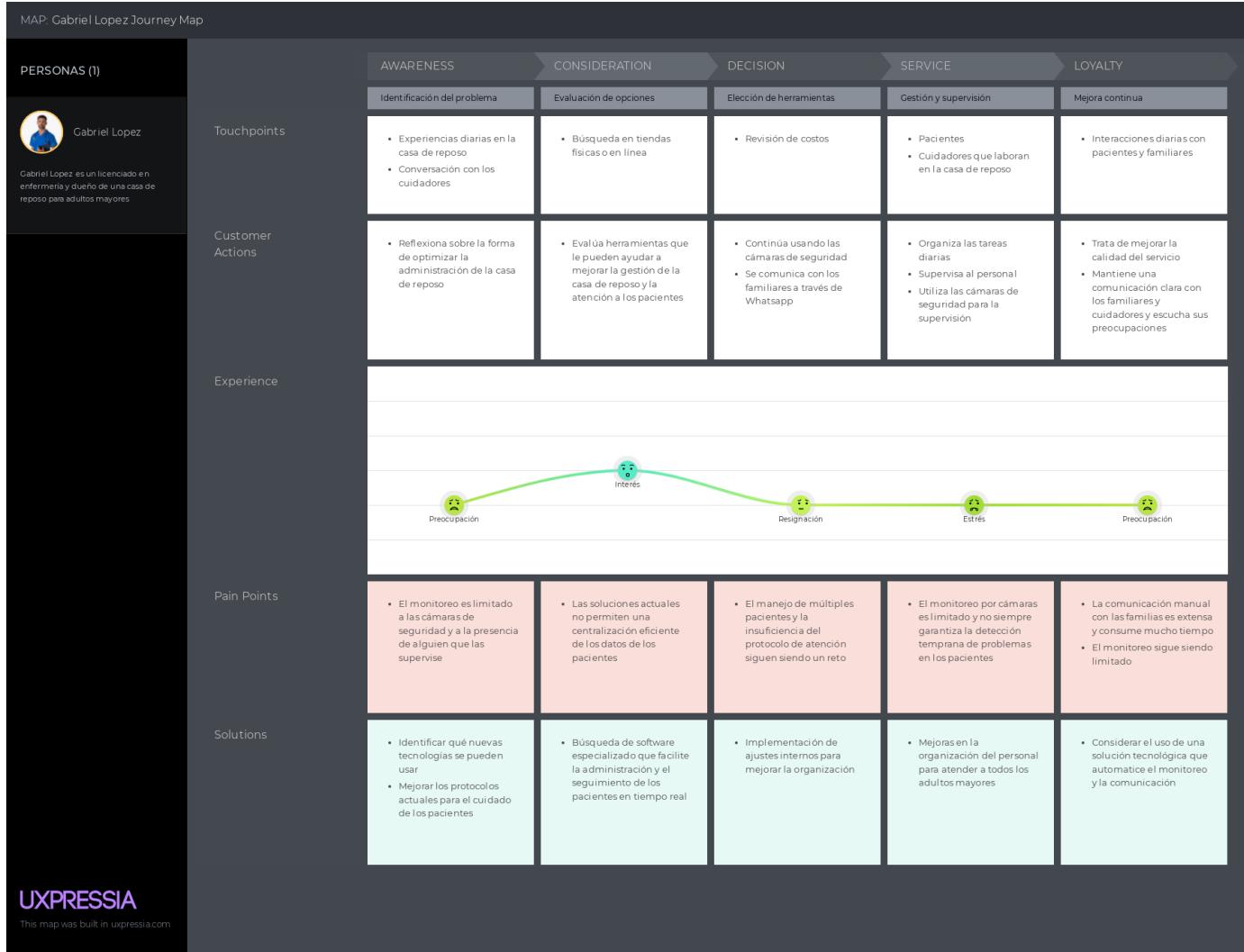
User Persona: Ana Martinez

El User Journey Mapping de Ana Martínez destaca las dificultades que enfrenta a lo largo de su labor diaria. Durante todo el proceso, Ana experimenta una carga significativa debido a la falta de herramientas eficientes que faciliten el monitoreo continuo de los pacientes. Esta situación genera estrés y preocupación constante, especialmente en relación con la prevención de caídas y la precisión en la toma de signos vitales.



User Persona: Gabriel Lopez

El User Journey Mapping de Gabriel López uestra su enfoque en la eficiencia de su personal y la mejora del bienestar de los pacientes. A lo largo de las fases, Gabriel se enfrenta a desafíos como la falta de un protocolo adecuado y la carga de trabajo, pero valora la adopción de nuevas tecnologías para optimizar la atención. Aunque se comunica activamente con los familiares y monitorea a los pacientes mediante cámaras, aún percibe la necesidad de una solución tecnológica que automatice estos procesos, mejorando tanto la atención de los pacientes como la tranquilidad de sus familias y cuidadores.



2.3.4. Empathy Mapping

User Persona: Ana Martinez

En el mapa de empatía de Ana Martinez, observamos que es una persona profundamente comprometida con la seguridad y bienestar de sus pacientes, pero que se siente frustrada por la falta de herramientas tecnológicas que faciliten su labor. A diario, organiza su trabajo mediante horarios y recordatorios, realiza el monitoreo de signos vitales, y ayuda a los pacientes a evitar caídas, mientras escucha tanto las necesidades de estos como el agradecimiento de sus familias. Aunque enfrenta dificultades para garantizar que los pacientes se movilicen con seguridad, ve en la tecnología una oportunidad para optimizar sus tareas y mejorar la calidad de su trabajo.

Think and Feel

- "Quiero asegurarme de que mis pacientes estén siempre seguros y cómodos"
- "Tengo que estar al lado de mi paciente todo el tiempo, esa es la labor de un cuidador"
- Se siente frustrada cuando no tiene las herramientas adecuadas para monitorear a los pacientes
- Se siente satisfecha al recibir el agradecimiento y cariño de los pacientes y sus familias

Hear

Escucha los comentarios sobre las necesidades de los pacientes

Escucha los agradecimientos de las familias por su trabajo



See

Ve que los pacientes en su entorno están expuestos a caídas

Revisa información sobre el cuidado de los adultos mayores

Ve la necesidad de realizar múltiples tareas de forma simultánea durante su jornada laboral

Say and Do

- Elabora un horario de trabajo
- Coloca alarmas para recordar la medicación del paciente
- Realiza el monitoreo de los signos vitales del paciente
- Ayuda a los pacientes a movilizarse para evitar caídas

PAINS

La ausencia de tecnología específica que facilite el monitoreo y cuidado de los pacientes

Dificultades para asegurar la movilidad segura de los pacientes

GAINS

- Posibilidad de optimizar las tareas y mejorar la calidad de su trabajo gracias a la tecnología
- Oportunidades para mantenerse al día con las mejores prácticas y conocimientos sobre su labor
- Que su trabajo sea reconocido por sus pacientes y los familiares

UXPRESSIA

This persona was built in uxpressia.com

User Persona: Gabriel Lopez

En el mapa de empatía de Gabriel Lopez, observamos que está preocupado por cómo gestionar eficientemente la atención de todos sus pacientes y asegurar que todos reciban el cuidado adecuado. Escucha al personal expresar dificultades en el monitoreo y en la atención de los pacientes, así como comentarios sobre la necesidad de mejorar la atención y la comunicación. Observa un entorno con múltiples tareas y una carga de trabajo alta, y nota que las tecnologías pueden ser difíciles de usar para los pacientes. Su objetivo es elevar la calidad del cuidado y aumentar la satisfacción de los pacientes y sus familias.

Think and Feel

- Le preocupa cómo manejar la gran cantidad de pacientes y asegurar que todos reciban la atención adecuada
- Desea ofrecer el mejor cuidado posible y optimizar la gestión de la casa de reposo

Hear

- Escucha al personal expresar dificultades en el monitoreo y atención de pacientes
- Oye comentarios sobre la necesidad de una mejor atención y comunicación



See

- Observa un entorno de trabajo con múltiples tareas y una carga de pacientes alta
- Ve que las tecnologías pueden ser difíciles de usar para los pacientes

Say and Do

- Prioriza las actividades según su importancia
- Busca formas de implementar mejoras en el cuidado y la administración

PAINS

- Dificultad para manejar y priorizar la atención de todos los pacientes simultáneamente
- La tecnología actual no siempre es intuitiva o eficaz para los cuidadores y puede ser difícil de manejar

GAINS

- Mejorar la gestión de la casa de reposo
- Mejorar la calidad del cuidado y la atención al paciente
- Mejorar la satisfacción de sus pacientes y familias

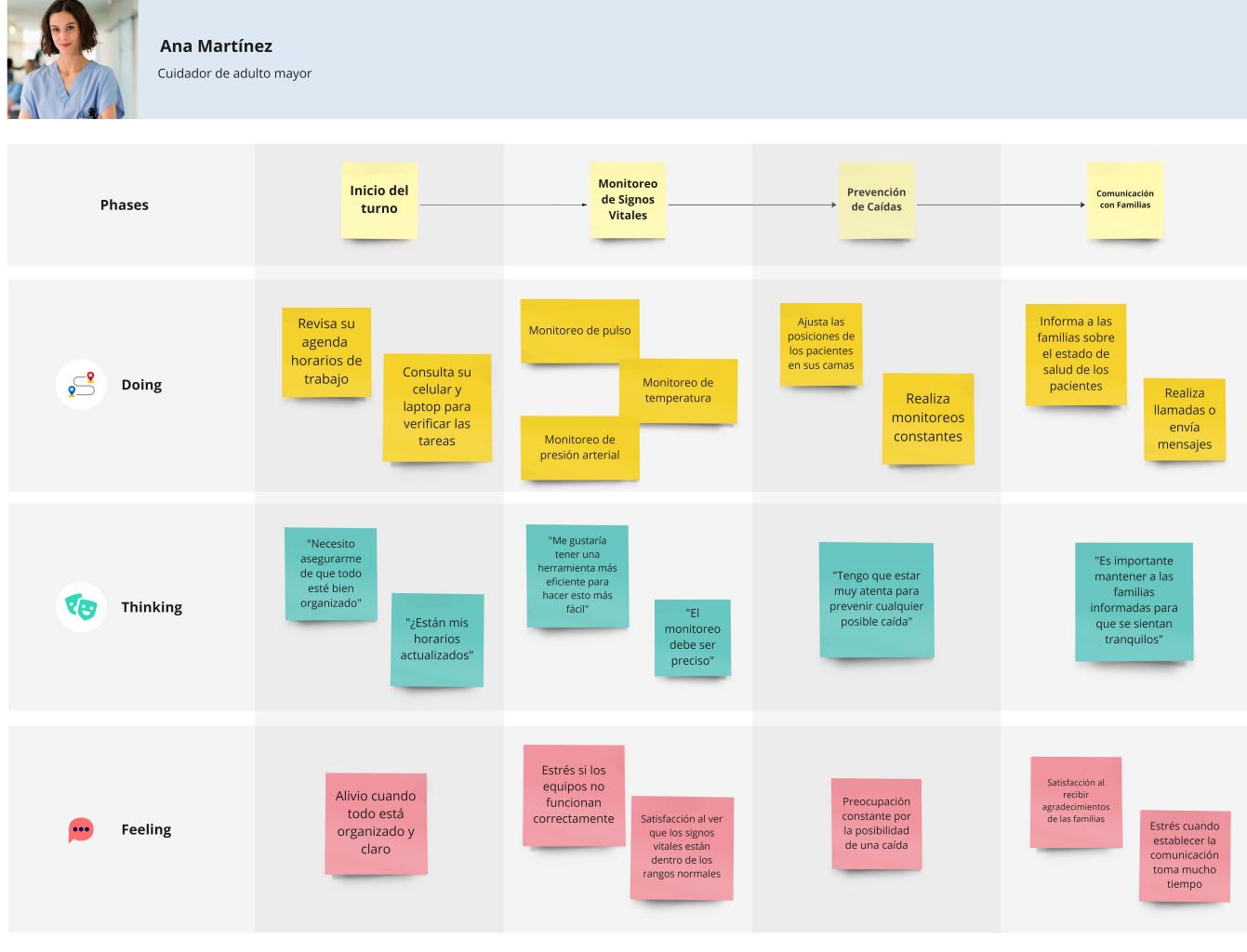
UXPRESSIA

This persona was built in upressoia.com

2.3.5. As-is Scenario Mapping

User Persona: Ana Martinez

Este mapa muestra el flujo de trabajo de Ana Martínez en la actualidad, desde el inicio de su turno hasta la comunicación con las familias. En cada fase, Ana realiza tareas como la revisión de su agenda, monitoreo de signos vitales (pulso, presión arterial y temperatura), ajustes en las posiciones de los pacientes para evitar las caídas y la comunicación constante con sus familias. Sus pensamientos están centrados en la organización y la precisión, mientras que sus emociones varían entre alivio cuando todo está en orden, estrés y preocupación constante por la prevención de caídas.



Áreas Positivas

- Comunicación con las familias
- Satisfacción por hacer bien su trabajo
- Responsabilidad y compromiso con su labor

Áreas Negativas

- Información dispersa en diferentes dispositivos
- El monitoreo constante del adulto mayor es desafiante

Blank Areas

- Qué tecnologías se pueden emplear para mejorar su trabajo

User Persona: Gabriel Lopez

Este mapa muestra el flujo de trabajo de Gabriel Lopez en la actualidad. Sus principales actividades son la asignación de tareas a su personal, supervisión y la comunicación con los familiares de sus pacientes. En cada fase realiza tareas como priorizar las tareas de los cuidadores, supervisión a través de cámaras de seguridad e informar a las familias sobre el estado de los adultos mayores. Sus pensamientos están centrados en la efectividad del cuidado que se brinda en su casa de reposo, mientras que sus emociones varían entre frustración, insatisfacción y preocupación.



Áreas Positivas

- Apertura a la implementación de tecnología
- Compromiso con el bienestar de los pacientes
- Evaluación de oportunidades de mejora

Áreas Negativas

- El monitoreo constante del adulto mayor es desafiante
- Las herramientas actuales no son muy eficientes

Blank Areas

- Con qué frecuencia presenta problemas con los dispositivos que usa actualmente

2.4. Ubiquitous Language

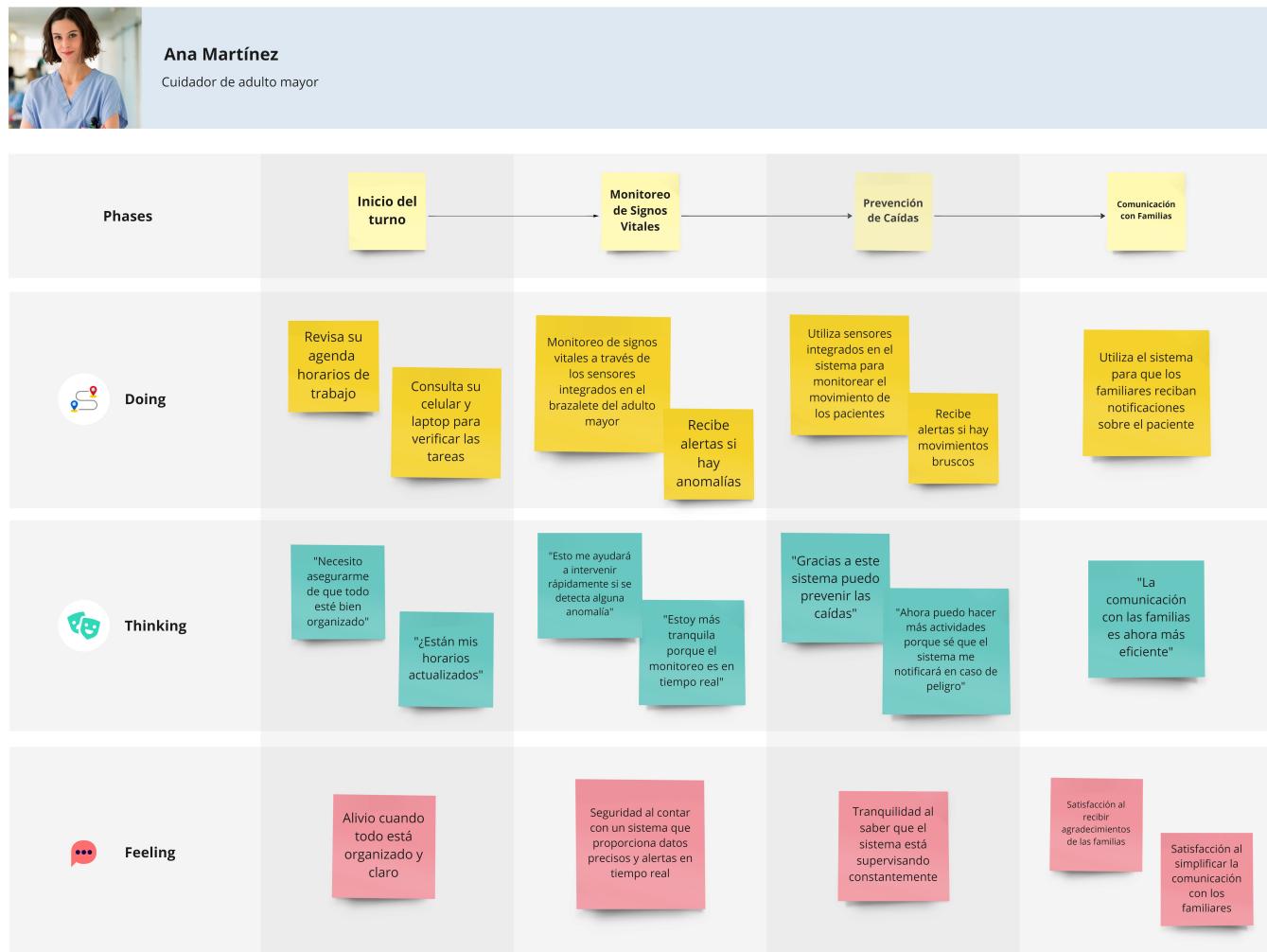
Término	Definición
Elderly	Personas mayores que son el objetivo principal del sistema.
Vital Signs	Indicadores de salud como la frecuencia cardíaca y la temperatura.
Caregiver	Persona encargada de cuidar y asistir al adulto mayor.

Término	Definición
Health Data	Información sobre el estado de salud del adulto mayor.
Emergency Response	Acciones tomadas en caso de una situación urgente, como una caída o un cambio en los signos vitales.
Activity Tracking	Seguimiento de las actividades del adulto mayor para detectar patrones inusuales.
Safety Protocol	Conjunto de reglas y procedimientos para asegurar la seguridad del adulto mayor.
Wellness	Estado general de salud y bienestar del adulto mayor.
Medication Reminders	Notificaciones para recordar al adulto mayor que tome sus medicamentos.
Heart Rate	Frecuencia con la que late el corazón, medida en pulsaciones por minuto (ppm).
Body Temperature	Medida de la temperatura corporal, que puede indicar fiebre o infecciones.
Fall Risk	Riesgo de caídas del adulto mayor.

Capítulo III: Requirements Specification

3.1. To-Be Scenario Mapping

Segmento 1: Cuidadores de adultos mayores (Caregivers)



Segmento 2: Dueños de casas de reposo (Nursing Home Owners)



3.2. User Stories

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
EP001	Información del Producto	Como Visitante, quiero desplazarme entre las secciones de la Página de Inicio para leer la información sobre el producto.	Dado que el visitante está en la Página de Inicio, cuando se desplaza por la sección de beneficios, entonces se muestra la información de los beneficios del producto.	
US001	Beneficios del Producto	Como Visitante, quiero leer información sobre los beneficios del producto para evaluar cómo satisface mis necesidades.	Dado que el visitante está en la Página de Inicio, cuando se desplaza por la sección de beneficios, entonces se muestra la información de los beneficios del producto.	EP001

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US002	Modelo de Negocio	Como Visitante, quiero leer información sobre el modelo de negocio para evaluar si el producto beneficia mis objetivos y necesidades.	Dado que el visitante está en la Página de Inicio, cuando se desplaza por la sección de modelo de negocio, entonces se muestra la información del modelo de negocio del producto.	EP001
US003	Reseñas de Clientes	Como Visitante, quiero leer reseñas de clientes para tomar una decisión informada sobre si comprar o usar un producto o servicio.	Dado que el visitante está en la Página de Inicio, cuando se desplaza por la sección de reseñas, entonces se muestran las reseñas de clientes. Dado que no hay reseñas disponibles, cuando no se encuentren datos, entonces se muestra un mensaje indicando la ausencia de reseñas.	EP001
US004	Contactar a Miembros de la Compañía	Como Visitante, quiero contactar a los miembros de la compañía para solicitar más información o recibir soporte antes de completar una compra.	Dado que el visitante está en la sección de contacto, cuando se envía una solicitud de contacto, entonces el sistema muestra un mensaje de confirmación. Dado que no se puede enviar la solicitud, cuando ocurre un error, entonces el sistema muestra un mensaje indicando el problema.	EP001
US005	Información de Miembros de la Compañía	Como Visitante, quiero leer información sobre los miembros de la compañía para entender su experiencia y antecedentes.	Dado que el visitante está en la Página de Inicio, cuando se desplaza por la sección de miembros de la compañía, entonces se muestra la información de cada miembro. Dado que no hay información disponible, cuando no se encuentran datos, entonces el sistema muestra un mensaje indicando la ausencia de información.	EP001
EP002	Monitoreo de Salud	Como Caregiver, quiero que la pulsera monitoree la temperatura y el ritmo cardíaco en tiempo real para recibir alertas en caso de valores anormales.		

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US006	Monitoreo de Temperatura en Tiempo Real	Como Caregiver, quiero que el sistema monitoree la temperatura en tiempo real para recibir alertas si la temperatura está fuera del rango saludable.	<p>Dado que la pulsera está en uso,</p> <p>cuando la temperatura es monitoreada,</p> <p>entonces el sistema muestra el valor actual y envía una alerta si está fuera del rango predefinido.</p> <p>Dado que el rango de temperatura no está configurado,</p> <p>cuando el sistema no encuentra un rango,</p> <p>entonces se muestra un mensaje solicitando la configuración.</p>	EP002
US007	Monitoreo del Ritmo Cardíaco en Tiempo Real	Como Caregiver, quiero que el sistema monitoree el ritmo cardíaco en tiempo real para recibir alertas si está fuera del rango saludable.	<p>Dado que la pulsera está en uso,</p> <p>cuando el ritmo cardíaco es monitoreado,</p> <p>entonces el sistema muestra el valor actual y envía una alerta si está fuera del rango predefinido.</p> <p>Dado que el sensor de ritmo cardíaco no funciona correctamente,</p> <p>cuando el sistema detecta un fallo,</p> <p>entonces se muestra un mensaje indicando el problema.</p>	EP002
US008	Alertas de Salud	Como Caregiver, quiero recibir alertas en caso de valores anormales de temperatura o ritmo cardíaco para tomar acciones rápidamente.	<p>Dado que se detectan valores anormales,</p> <p>cuando el sistema envía una alerta,</p> <p>entonces el cuidador recibe una notificación en su dispositivo móvil.</p> <p>Dado que el dispositivo no está vinculado,</p> <p>cuando el sistema no puede enviar la alerta,</p> <p>entonces se muestra un mensaje solicitando la vinculación.</p>	EP002

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US009	Historial de Datos de Salud	Como Caregiver, quiero ver el historial de datos de salud para tener un registro completo de la temperatura y ritmo cardíaco a lo largo del tiempo.	<p>Dado que el cuidador está en la sección de historial, cuando selecciona una fecha específica, entonces el sistema muestra el historial con gráficos o tablas de la temperatura y el ritmo cardíaco.</p> <p>Dado que no hay datos disponibles, cuando se selecciona un rango sin información, entonces se muestra un mensaje indicando la ausencia de datos.</p>	EP002
US010	Configuración de Alertas Personalizadas	Como Caregiver, quiero configurar alertas personalizadas para temperatura y ritmo cardíaco para adaptar las notificaciones a necesidades específicas.	<p>Dado que el cuidador está en la sección de configuración, cuando se configuran los umbrales de temperatura y ritmo cardíaco, entonces el sistema guarda los umbrales personalizados para enviar alertas.</p> <p>Dado que se intenta guardar un umbral no válido, cuando se ingresa un valor incorrecto, entonces se muestra un mensaje de error indicando la configuración incorrecta.</p>	EP002
EP003	Solicitud de Ayuda	Como Caregiver, quiero que los usuarios puedan solicitar ayuda rápidamente en caso de necesidad para poder actuar de manera oportuna.	<p>Dado que el usuario presiona el botón de emergencia, cuando se envía la notificación, entonces el cuidador recibe un mensaje con la ubicación actual y el estado de emergencia.</p>	
US011	Envío de Notificación de Emergencia	Como Caregiver, quiero que se envíe una notificación cuando se presione el botón de emergencia para recibir ayuda rápidamente en caso de necesidad.	<p>Dado que no se registra la confirmación de recepción, cuando el sistema no recibe respuesta, entonces intenta reenviar la notificación y registra el fallo.</p>	EP003

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US012	Recepción de Notificaciones de Ayuda	Como Caregiver, quiero recibir notificaciones en mi dispositivo móvil cuando se presione el botón de emergencia para actuar rápidamente.	<p>Dado que el botón de emergencia es presionado, cuando se envía la notificación, entonces el cuidador recibe la notificación con la ubicación y detalles de la emergencia.</p> <p>Dado que la notificación no es recibida, cuando el sistema detecta un fallo, entonces informa al administrador y registra el problema.</p>	EP003
US013	Gestión de Notificaciones	Como Caregiver, quiero gestionar las notificaciones recibidas para revisar y dar seguimiento a las alertas de emergencia.	<p>Dado que el cuidador está en la sección de notificaciones, cuando se revisa el historial, entonces se muestran todas las notificaciones de emergencia recibidas con sus detalles.</p> <p>Dado que no se encuentran notificaciones, cuando no hay datos en el historial, entonces se muestra un mensaje indicando la ausencia de alertas.</p>	EP003
US014	Configuración de Alertas de Emergencia	Como Caregiver, quiero configurar alertas de emergencia para ajustar los parámetros según necesidades y preferencias.	<p>Dado que el cuidador está en la sección de configuración de emergencias, cuando se configuran las preferencias de notificación, entonces el sistema guarda y aplica estas preferencias a futuras emergencias.</p> <p>Dado que se intenta guardar una configuración no válida, cuando se ingresa un valor incorrecto, entonces se muestra un mensaje de error solicitando la corrección.</p>	EP003
EP004	Gestión de Pulseras	Como Nursing Home Owner, quiero gestionar las pulseras alquiladas para actualizar la información de alquiler, rastrear pulseras disponibles y generar informes de uso para la facturación.		

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US015	Acceso a Datos de Usuario	Como Nursing Home Owner, quiero acceder a los datos de usuario para generar informes y realizar un seguimiento del uso de las pulseras.	<p>Dado que el administrador está en la sección de datos de usuario,</p> <p>cuando selecciona los datos de temperatura o ritmo cardíaco,</p> <p>entonces el sistema muestra el historial del usuario.</p> <p>Dado que no hay datos disponibles,</p> <p>cuando se selecciona un usuario sin información,</p> <p>entonces se muestra un mensaje indicando la ausencia de información.</p>	EP004
US016	Gestión de Pulseras Alquiladas	Como Nursing Home Owner, quiero gestionar las pulseras alquiladas para actualizar información, rastrear su uso y emitir facturas.	<p>Dado que el administrador está en la sección de gestión de pulseras,</p> <p>cuando se actualiza el estado de una pulsera alquilada,</p> <p>entonces el sistema refleja el cambio en el inventario y en la facturación.</p> <p>Dado que el inventario no coincide,</p> <p>cuando el sistema detecta una discrepancia,</p> <p>entonces se muestra un mensaje de error indicando el problema.</p>	EP004
US017	Visualización de Pulseras Disponibles	Como Nursing Home Owner, quiero ver una lista de pulseras disponibles para saber cuáles están en alquiler y cuáles están disponibles para nuevos usuarios.	<p>Dado que el administrador está en la sección de pulseras,</p> <p>cuando selecciona la opción de pulseras disponibles,</p> <p>entonces el sistema muestra una lista de pulseras con su estado actual.</p> <p>Dado que la lista no se actualiza,</p> <p>cuando el sistema detecta un error,</p> <p>entonces muestra un mensaje solicitando la verificación manual.</p>	EP004

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US018	Generación de Informes de Uso	Como Nursing Home Owner, quiero generar informes sobre el uso de las pulseras para analizar su utilización y rendimiento.	<p>Dado que el administrador está en la sección de informes, cuando selecciona generar un informe de uso, entonces el sistema muestra un informe detallado con la información de uso de las pulseras.</p> <p>Dado que no se puede exportar el informe, cuando el sistema detecta un fallo, entonces muestra un mensaje indicando el problema y solicita un nuevo intento.</p>	EP004
US019	Monitoreo de Sensores de Pulseras	Como Nursing Home Owner, quiero monitorear el estado de los sensores de las pulseras para asegurarme de que funcionen correctamente en todo momento.	<p>Dado que el administrador está en la sección de monitoreo de sensores, cuando selecciona una pulsera específica, entonces el sistema muestra el estado de sus sensores, incluyendo batería, conectividad y precisión.</p> <p>Dado que se detecta una falla, cuando el sistema encuentra un problema en el sensor, entonces envía una alerta al administrador indicando el fallo.</p>	EP004
US020	Asignación de Pulsera a un Usuario	Como Nursing Home Owner, quiero asignar una pulsera a un nuevo usuario para que pueda comenzar a utilizar el sistema de monitoreo de salud de inmediato.	<p>Dado que el administrador está en la sección de asignación, cuando selecciona un nuevo usuario y una pulsera disponible, entonces el sistema asigna la pulsera al usuario y actualiza el estado en el inventario.</p> <p>Dado que la asignación no se realiza correctamente, cuando el sistema detecta un error, entonces muestra un mensaje solicitando la verificación manual.</p>	EP004

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US021	Generación de Informes Financieros	Como Nursing Home Owner, quiero generar informes financieros para analizar los ingresos generados por el alquiler de pulseras.	<p>Dado que el administrador está en la sección de informes financieros,</p> <p>cuando selecciona generar un informe,</p> <p>entonces el sistema muestra un informe financiero detallado con ingresos por alquiler y pagos recibidos.</p> <p>Dado que no hay datos en el rango seleccionado,</p> <p>cuando el sistema no encuentra información,</p> <p>entonces muestra un mensaje indicando la ausencia de información.</p>	EP004
US022	Configuración de Pagos Automatizados	Como Nursing Home Owner, quiero configurar pagos automatizados para facilitar la gestión de los cobros por el alquiler de pulseras.	<p>Dado que el administrador está en la sección de configuración,</p> <p>cuando selecciona configurar pagos automáticos,</p> <p>entonces el sistema procesa los pagos automáticamente según el intervalo configurado.</p> <p>Dado que no se puede procesar un pago,</p> <p>cuando el sistema detecta un fallo en el proceso,</p> <p>entonces muestra una alerta indicando el problema y solicita la verificación manual.</p>	EP004
US023	Cancelación de Alquiler de Pulseras	Como Nursing Home Owner, quiero cancelar el alquiler de una pulsera para liberar la pulsera y actualizar el estado en el inventario.	<p>Dado que el administrador está en la sección de gestión de alquileres,</p> <p>cuando selecciona cancelar un alquiler,</p> <p>entonces el sistema libera la pulsera en el inventario y actualiza el estado del alquiler.</p> <p>Dado que no se puede liberar la pulsera,</p> <p>cuando el sistema detecta un fallo,</p> <p>entonces muestra un mensaje solicitando la verificación manual.</p>	EP004

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US024	Actualización de Tarifas de Alquiler	Como Nursing Home Owner, quiero actualizar las tarifas de alquiler de pulseras para reflejar cambios en los costos actuales de alquiler.	<p>Dado que el administrador está en la sección de tarifas, cuando selecciona actualizar una tarifa, entonces el sistema aplica la nueva tarifa a futuros alquileres.</p> <p>Dado que se intenta aplicar la tarifa retroactivamente, cuando el sistema detecta una inconsistencia, entonces muestra un mensaje solicitando la verificación manual.</p>	EP004
EP005	Interfaz de Usuario	Como Visitor, Caregiver o Nursing Home Owner, quiero una interfaz fácil de usar para visualizar y entender los datos de la pulsera, acceder a informes y gestionar notificaciones de manera intuitiva.		
US025	Acceso Rápido a Datos de la Pulsera	Como Caregiver, quiero acceder rápidamente a los datos actuales de la pulsera desde la pantalla principal para ver información de salud sin navegar por múltiples páginas.	<p>Dado que el cuidador está en la pantalla principal de la aplicación, cuando abre la aplicación, entonces el sistema muestra la temperatura y el ritmo cardíaco actuales sin necesidad de navegar por diferentes menús.</p> <p>Dado que no se pueden actualizar los datos en tiempo real, cuando el sistema detecta una pérdida de conexión, entonces muestra un mensaje indicando el problema y solicita un nuevo intento.</p>	EP005
US026	Navegación Intuitiva	Como Caregiver, quiero una navegación intuitiva en la aplicación para acceder fácilmente a las diferentes funcionalidades sin confusión.	<p>Dado que el cuidador está navegando en la aplicación, cuando selecciona una sección, entonces puede cambiar de una sección a otra a través de una barra de navegación o botones accesibles.</p> <p>Dado que se intenta acceder a una sección no disponible, cuando el sistema detecta un acceso indebido, entonces muestra un mensaje indicando la opción no habilitada.</p>	EP005

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US027	Visualización Clara de Alertas	Como Caregiver, quiero que las alertas importantes se destaque visualmente para identificar rápidamente las notificaciones críticas.	<p>Dado que el cuidador está en la aplicación, cuando se recibe una alerta crítica, entonces el sistema destaca visualmente la alerta utilizando colores y tamaño de texto.</p> <p>Dado que la alerta no se visualiza correctamente, cuando el sistema detecta un fallo en la visualización, entonces muestra un mensaje indicando el problema y solicita la verificación manual.</p>	EP005
US028	Personalización de la Interfaz	Como Caregiver, quiero personalizar algunos aspectos de la interfaz para adaptarla a mis preferencias de visualización.	<p>Dado que el cuidador está en la sección de configuración, cuando selecciona opciones de personalización, entonces el sistema permite cambiar el esquema de colores y el tamaño de texto.</p> <p>Dado que el sistema no guarda las preferencias, cuando se intenta guardar la configuración, entonces se muestra un mensaje indicando el problema y solicita un nuevo intento.</p>	EP005
US029	Notificaciones en Tiempo Real	Como Caregiver, quiero recibir notificaciones en tiempo real sobre cambios en la salud para reaccionar rápidamente en caso de emergencias.	<p>Dado que el sistema detecta un cambio significativo en la salud, cuando se envía la notificación, entonces el cuidador recibe una notificación instantánea en su dispositivo móvil o aplicación web.</p> <p>Dado que la notificación no llega al dispositivo, cuando el sistema detecta un fallo en la entrega, entonces registra el fallo e intenta reenviar la notificación.</p>	EP005
US030	Resumen Diario de Salud	Como Caregiver, quiero recibir un resumen diario del estado de salud para estar informado sobre el bienestar general sin revisar constantemente los datos.	<p>Dado que el sistema está registrando los datos de salud, cuando finaliza el día, entonces se genera y envía un resumen diario con los datos de salud clave al usuario.</p> <p>Dado que el resumen no se genera correctamente, cuando el sistema detecta un problema, entonces muestra un mensaje indicando el fallo y solicita la verificación manual.</p>	EP005

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
EP006	Generación de Informes	Como Nursing Home Owner, quiero generar informes detallados con los datos de la pulsera para analizar la salud de los usuarios, generar informes de uso y hacer recomendaciones basadas en los datos recopilados.	<p>Dado que el administrador está en la sección de informes, cuando selecciona un período de tiempo, entonces el sistema genera un informe con gráficos y análisis de los datos de salud recopilados.</p>	
US031	Generación de Informes de Salud	Como Nursing Home Owner, quiero generar informes detallados sobre la salud de los usuarios para analizar patrones y tendencias a lo largo del tiempo.	<p>Dado que no hay datos en el período seleccionado, cuando el sistema detecta la ausencia de información, entonces muestra un mensaje indicando la falta de datos.</p>	EP006
US032	Filtros Avanzados en Informes	Como Nursing Home Owner, quiero aplicar filtros avanzados al generar informes para obtener la información más relevante de manera rápida y efectiva.	<p>Dado que el administrador está generando un informe, cuando selecciona filtros como fecha, tipo de alerta y rango de valores, entonces el sistema muestra los datos relevantes basados en los filtros aplicados.</p> <p>Dado que los filtros no producen resultados, cuando el sistema no encuentra datos coincidentes, entonces muestra un mensaje indicando la ausencia de información.</p>	EP006
US033	Análisis de Tendencias de Salud	Como Nursing Home Owner, quiero ver un análisis de tendencias basado en los datos recopilados para identificar posibles problemas de salud a largo plazo.	<p>Dado que el administrador está en la sección de informes, cuando selecciona análisis de tendencias, entonces el sistema genera gráficos que muestran patrones y posibles riesgos de salud a partir de los datos de los usuarios.</p> <p>Dado que no hay datos suficientes, cuando el sistema detecta la ausencia de datos, entonces muestra un mensaje indicando la falta de información para el análisis.</p>	EP006

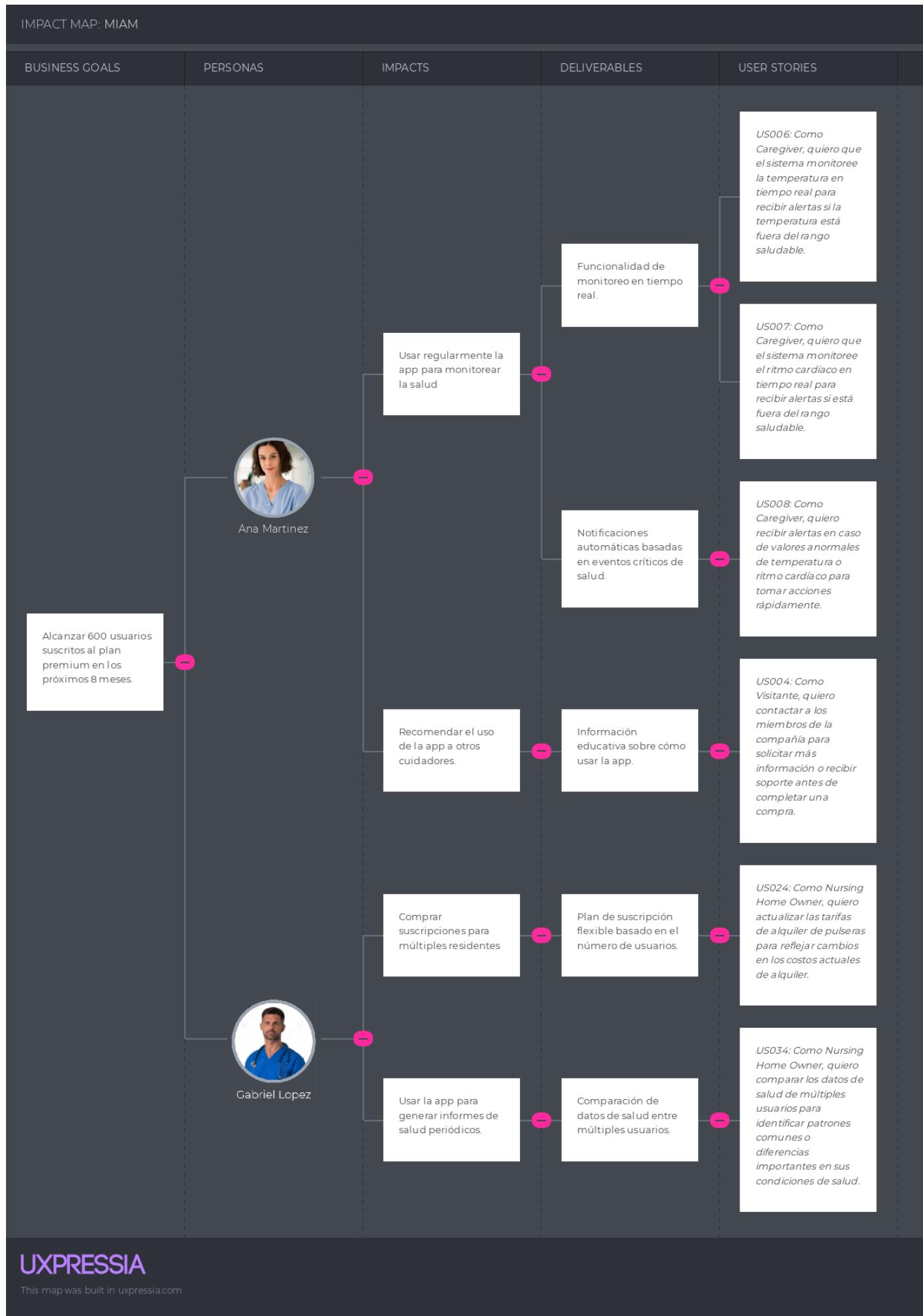
Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
US034	Comparación de Datos entre Usuarios	Como Nursing Home Owner, quiero comparar los datos de salud de múltiples usuarios para identificar patrones comunes o diferencias importantes en sus condiciones de salud.	<p>Dado que el administrador tiene múltiples usuarios en el sistema,</p> <p>cuando selecciona usuarios específicos,</p> <p>entonces el sistema genera un informe comparativo con gráficos y datos fáciles de interpretar.</p> <p>Dado que no hay datos suficientes para la comparación,</p> <p>cuando el sistema no encuentra información,</p> <p>entonces muestra un mensaje indicando la falta de datos para la comparación.</p>	EP006
US035	Exportación de Datos para Análisis Externo	Como Nursing Home Owner, quiero exportar todos los datos de salud en formato bruto para un análisis más detallado con herramientas externas.	<p>Dado que el administrador necesita realizar un análisis externo,</p> <p>cuando selecciona exportar datos,</p> <p>entonces el sistema permite la exportación de todos los datos de salud en formato CSV o JSON.</p> <p>Dado que la exportación no se realiza correctamente,</p> <p>cuando el sistema detecta un problema,</p> <p>entonces muestra un mensaje indicando el fallo y solicita un nuevo intento.</p>	EP006
EP007	Desarrollo de API	Como desarrollador integrador, quiero una API bien documentada con funcionalidades clave para integrar la aplicación con otros sistemas, gestionar datos de la pulsera, recibir notificaciones y acceder a informes de manera eficiente.	<p>Dado que el desarrollador necesita acceder a datos de la pulsera,</p> <p>cuando la API recibe una solicitud de datos,</p> <p>entonces devuelve los datos de temperatura y ritmo cardíaco en tiempo real.</p> <p>Dado que no se permite el acceso,</p> <p>cuando la API detecta un problema de acceso,</p> <p>entonces muestra un mensaje indicando la falta de acceso a los datos solicitados.</p>	EP007
TS036	Implementación de API para Integración	Como desarrollador, quiero implementar una API para integrar los datos de la pulsera con otros sistemas de monitoreo para asegurar que los datos sean accesibles desde otras plataformas.	<p>Dado que el desarrollador necesita acceder a datos de la pulsera,</p> <p>cuando la API recibe una solicitud de datos,</p> <p>entonces devuelve los datos de temperatura y ritmo cardíaco en tiempo real.</p> <p>Dado que no se permite el acceso,</p> <p>cuando la API detecta un problema de acceso,</p> <p>entonces muestra un mensaje indicando la falta de acceso a los datos solicitados.</p>	EP007

Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
TS037	Creación de Endpoints de Notificación	Como desarrollador, quiero crear endpoints para gestionar notificaciones en la API para que las alertas de salud se configuren y envíen correctamente.	<p>Dado que el desarrollador está implementando la API, cuando se configura una alerta, entonces los endpoints permiten la creación, edición y eliminación de notificaciones.</p> <p>Dado que no se permiten notificaciones, cuando la API detecta un problema, entonces muestra un mensaje indicando la falta de acceso a la funcionalidad y solicita un nuevo intento.</p>	EP007
TS038	Seguridad de la API	Como desarrollador, quiero asegurarme de que la API tenga la autenticación y autorización adecuadas para proteger los datos sensibles de los usuarios de la pulsera.	<p>Dado que el usuario intenta acceder a los datos de la API, cuando se realiza una solicitud, entonces la API requiere un token de autenticación válido para acceder a los datos de salud.</p> <p>Dado que un usuario intenta acceder sin autenticación, cuando la API detecta el acceso no autorizado, entonces bloquea la solicitud e informa la necesidad de autenticación.</p>	EP007
TS039	Documentación de la API	Como desarrollador, quiero que la API esté documentada de manera clara y completa para que otros desarrolladores puedan integrarse fácilmente.	<p>Dado que la API está lista para su uso, cuando un desarrollador necesita información, entonces la documentación incluye ejemplos de uso de todos los endpoints y parámetros disponibles.</p> <p>Dado que no se encuentra la documentación completa, cuando un desarrollador intenta acceder a un endpoint, entonces se muestra un mensaje solicitando la actualización de la documentación.</p>	EP007

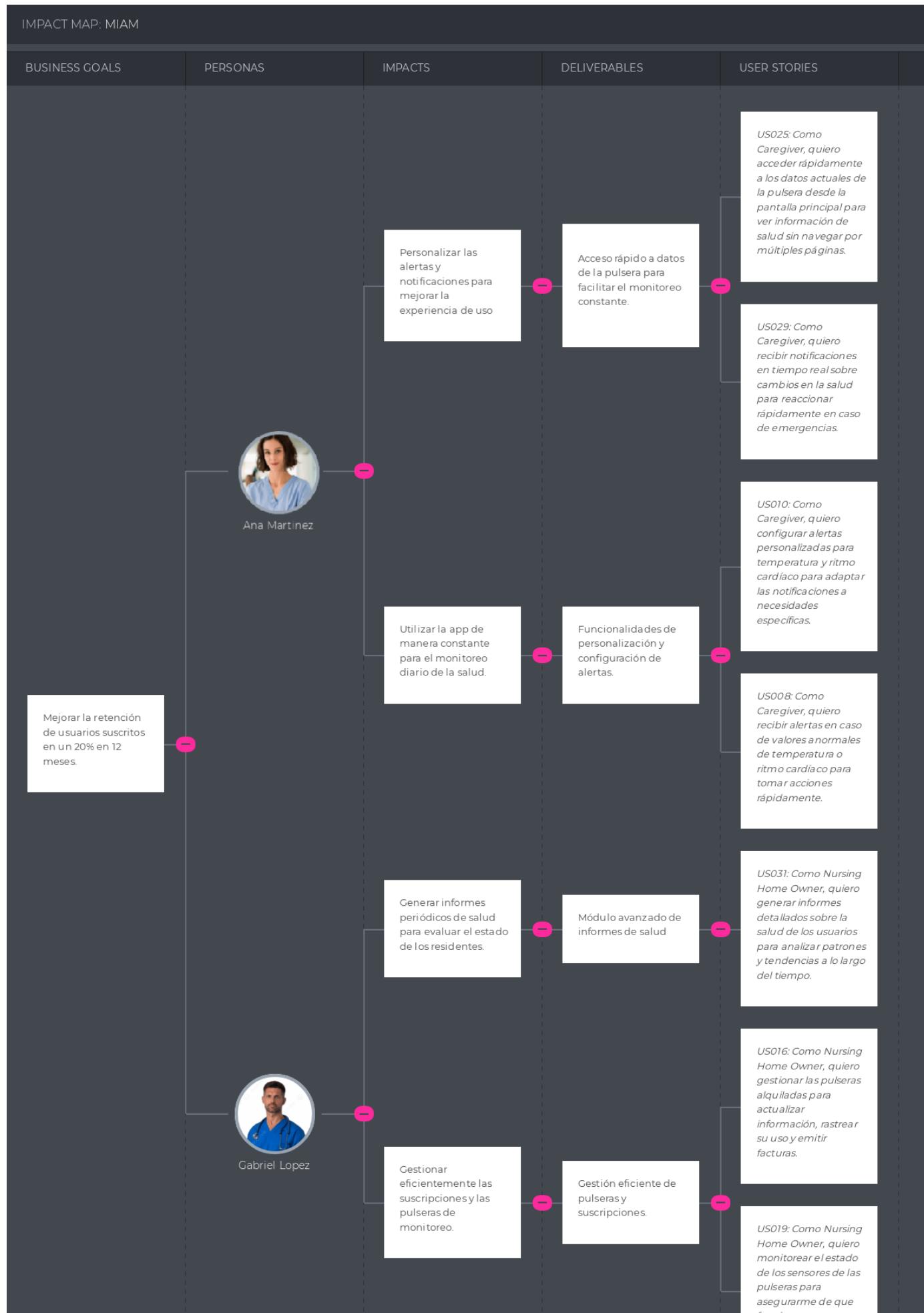
Epic/Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con Epic ID
TS040	Optimización del Rendimiento de la API	Como desarrollador, quiero optimizar la API para que las respuestas sean rápidas y eficientes, mejorando la experiencia del usuario final.	Dado que la API está en producción, cuando se realizan consultas, entonces la API tiene tiempos de respuesta inferiores a 200ms.	Dado que se realizan consultas repetitivas, cuando la API detecta patrones de uso, entonces utiliza técnicas de caché para mejorar el rendimiento.
EP006	Autenticación de Usuario	Como Visitante, quiero poder iniciar sesión de manera segura en la aplicación para acceder a las funcionalidades personalizadas.		
US006	Inicio de Sesión	Como Visitante, quiero iniciar sesión utilizando mi correo electrónico y contraseña para acceder a las funcionalidades personalizadas de la aplicación.	Dado que el usuario está en la pantalla de login, cuando ingresa su correo y contraseña correctos, entonces se permite el acceso a su cuenta y se redirige a la página principal.	EP002
US007	Recuperación de Contraseña	Como Visitante, quiero recuperar mi contraseña en caso de haberla olvidado para poder acceder nuevamente a mi cuenta.	Dado que el usuario ha olvidado su contraseña, cuando selecciona la opción de "Recuperar Contraseña" e ingresa su correo electrónico, entonces el sistema envía un enlace de recuperación a su correo. Dado que se intenta recuperar con un correo no registrado, cuando se produce este error, entonces se muestra un mensaje indicando la ausencia de cuenta asociada.	EP002

3.3. Impact Mapping

Este Impact Mapping se desarrolló para el business goal de alcanzar 600 usuarios suscritos al plan premium en los próximos 8 meses. Se identificaron dos actores clave: Ana Martínez (Caregiver) y Gabriel López (Nursing Home Owner)



Este Impact Mapping se desarrolló para el business goal de mejorar la retención de usuarios suscritos en un 20% en 12 meses.. Se identificaron dos actores clave: Ana Martínez (Caregiver) y Gabriel López (Nursing Home Owner)





3.4. Product Backlog

Orden	User Story			Story Points(1 / 2 / 3 / 5 / 8)
	Id	Título	Descripción	
1	US001	Beneficios del Producto	Como Visitante, quiero leer información sobre los beneficios del producto para evaluar cómo satisface mis necesidades.	2
2	US002	Modelo de Negocio	Como Visitante, quiero leer información sobre el modelo de negocio para evaluar si el producto beneficia mis objetivos y necesidades.	2
3	US003	Reseñas de Clientes	Como Visitante, quiero leer reseñas de clientes para tomar una decisión informada sobre si comprar o usar un producto o servicio.	2
4	US004	Contactar a Miembros de la Compañía	Como Visitante, quiero contactar a los miembros de la compañía para solicitar más información o recibir soporte antes de completar una compra.	3
5	US005	Información de Miembros de la Compañía	Como Visitante, quiero leer información sobre los miembros de la compañía para entender su experiencia y antecedentes.	3
6	US006	Monitoreo de Temperatura en Tiempo Real	Como Caregiver, quiero que el sistema monitoree la temperatura en tiempo real para recibir alertas si la temperatura está fuera del rango saludable.	5
7	US007	Monitoreo del Ritmo Cardíaco en Tiempo Real	Como Caregiver, quiero que el sistema monitoree el ritmo cardíaco en tiempo real para recibir alertas si está fuera del rango saludable.	5
8	US008	Alertas de Salud	Como Caregiver, quiero recibir alertas en caso de valores anormales de temperatura o ritmo cardíaco para tomar acciones rápidamente.	5
9	US009	Historial de Datos de Salud	Como Caregiver, quiero ver el historial de datos de salud para tener un registro completo de la temperatura y ritmo cardíaco a lo largo del tiempo.	5
10	US010	Configuración de Alertas Personalizadas	Como Caregiver, quiero configurar alertas personalizadas para temperatura y ritmo cardíaco para adaptar las notificaciones a necesidades específicas.	5
11	US011	Envío de Notificación de Emergencia	Como Caregiver, quiero que se envíe una notificación cuando se presione el botón de emergencia para recibir ayuda rápidamente en caso de necesidad.	3
12	US012	Recepción de Notificaciones de Ayuda	Como Caregiver, quiero recibir notificaciones en mi dispositivo móvil cuando se presione el botón de emergencia para actuar rápidamente.	3
13	US013	Gestión de Notificaciones	Como Caregiver, quiero gestionar las notificaciones recibidas para revisar y dar seguimiento a las alertas de emergencia.	5
14	US014	Configuración de Alertas de Emergencia	Como Caregiver, quiero configurar alertas de emergencia para ajustar los parámetros según necesidades y preferencias.	5
15	US025	Acceso Rápido a Datos de la Pulsera	Como Caregiver, quiero acceder rápidamente a los datos actuales de la pulsera desde la pantalla principal para ver información de salud sin navegar por múltiples páginas.	3

User Story	Story Points(1 / 2 / 3 / 5 / 8)			
Orden	Story Id	Título	Descripción	
16	US026	Navegación Intuitiva	Como Caregiver, quiero una navegación intuitiva en la aplicación para acceder fácilmente a las diferentes funcionalidades sin confusión.	3
17	US027	Visualización Clara de Alertas	Como Caregiver, quiero que las alertas importantes se destaque visualmente para identificar rápidamente las notificaciones críticas.	3
18	US028	Personalización de la Interfaz	Como Caregiver, quiero personalizar algunos aspectos de la interfaz para adaptarla a mis preferencias de visualización.	3
19	US029	Notificaciones en Tiempo Real	Como Caregiver, quiero recibir notificaciones en tiempo real sobre cambios en la salud para reaccionar rápidamente en caso de emergencias.	5
20	US030	Resumen Diario de Salud	Como Caregiver, quiero recibir un resumen diario del estado de salud para estar informado sobre el bienestar general sin revisar constantemente los datos.	5
21	US015	Acceso a Datos de Usuario	Como Nursing Home Owner, quiero acceder a los datos de usuario para generar informes y realizar un seguimiento del uso de las pulseras.	5
22	US016	Gestión de Pulseras Alquiladas	Como Nursing Home Owner, quiero gestionar las pulseras alquiladas para actualizar información, rastrear su uso y emitir facturas.	5
23	US017	Visualización de Pulseras Disponibles	Como Nursing Home Owner, quiero ver una lista de pulseras disponibles para saber cuáles están en alquiler y cuáles están disponibles para nuevos usuarios.	3
24	US018	Generación de Informes de Uso	Como Nursing Home Owner, quiero generar informes sobre el uso de las pulseras para analizar su utilización y rendimiento.	5
25	US019	Monitoreo de Sensores de Pulseras	Como Nursing Home Owner, quiero monitorear el estado de los sensores de las pulseras para asegurarme de que funcionen correctamente en todo momento.	5
26	US020	Asignación de Pulsera a un Usuario	Como Nursing Home Owner, quiero asignar una pulsera a un nuevo usuario para que pueda comenzar a utilizar el sistema de monitoreo de salud de inmediato.	5
27	US021	Generación de Informes Financieros	Como Nursing Home Owner, quiero generar informes financieros para analizar los ingresos generados por el alquiler de pulseras.	5
28	US022	Configuración de Pagos Automatizados	Como Nursing Home Owner, quiero configurar pagos automatizados para facilitar la gestión de los cobros por el alquiler de pulseras.	5
29	US023	Cancelación de Alquiler de Pulseras	Como Nursing Home Owner, quiero cancelar el alquiler de una pulsera para liberar la pulsera y actualizar el estado en el inventario.	5
30	US024	Actualización de Tarifas de Alquiler	Como Nursing Home Owner, quiero actualizar las tarifas de alquiler de pulseras para reflejar cambios en los costos actuales de alquiler.	3
31	US031	Generación de Informes de Salud	Como Nursing Home Owner, quiero generar informes detallados sobre la salud de los usuarios para analizar patrones y tendencias a lo largo del tiempo.	5
32	US032	Filtros Avanzados en Informes	Como Nursing Home Owner, quiero aplicar filtros avanzados al generar informes para obtener la información más relevante de manera rápida y efectiva.	3
33	US033	Análisis de Tendencias de Salud	Como Nursing Home Owner, quiero ver un análisis de tendencias basado en los datos recopilados para identificar posibles problemas de salud a largo plazo.	5
34	US034	Comparación de Datos entre Usuarios	Como Nursing Home Owner, quiero comparar los datos de salud de múltiples usuarios para identificar patrones comunes o diferencias importantes en sus condiciones de salud.	5

User Story Id	Título	Descripción	Story Points(1 / 2 / 3 / 5 / 8)
35 US035	Exportación de Datos para Análisis Externo	Como Nursing Home Owner, quiero exportar todos los datos de salud en formato bruto para un análisis más detallado con herramientas externas.	3
36 TS036	Implementación de API para Integración	Como desarrollador, quiero implementar una API para integrar los datos de la pulsera con otros sistemas de monitoreo para asegurar que los datos sean accesibles desde otras plataformas.	5
37 TS037	Creación de Endpoints de Notificación	Como desarrollador, quiero crear endpoints para gestionar notificaciones en la API para que las alertas de salud se configuren y envíen correctamente.	5
38 TS038	Seguridad de la API	Como desarrollador, quiero asegurarme de que la API tenga la autenticación y autorización adecuadas para proteger los datos sensibles de los usuarios de la pulsera.	8
39 TS039	Documentación de la API	Como desarrollador, quiero que la API esté documentada de manera clara y completa para que otros desarrolladores puedan integrarse fácilmente.	3
40 TS040	Optimización del Rendimiento de la API	Como desarrollador, quiero optimizar la API para que las respuestas sean rápidas y eficientes, mejorando la experiencia del usuario final.	5

Capítulo IV: Strategic-Level Software Design

4.1. Strategic-Level Attribute-Driven Design

4.1.1. Event Storming

EventStorming es una técnica colaborativa e iterativa de modelado que permite explorar en profundidad una problemática compleja y de gran escala, facilitando la identificación de la mayor cantidad de detalles y desafíos posibles.



Enlace del Miroo para verlo completo: https://miro.com/app/board/uXjVKldWbQI/?share_link_id=811519345320

4.1.1.1. Candidate Context Discovery

Step 1: Unstructured Exploration

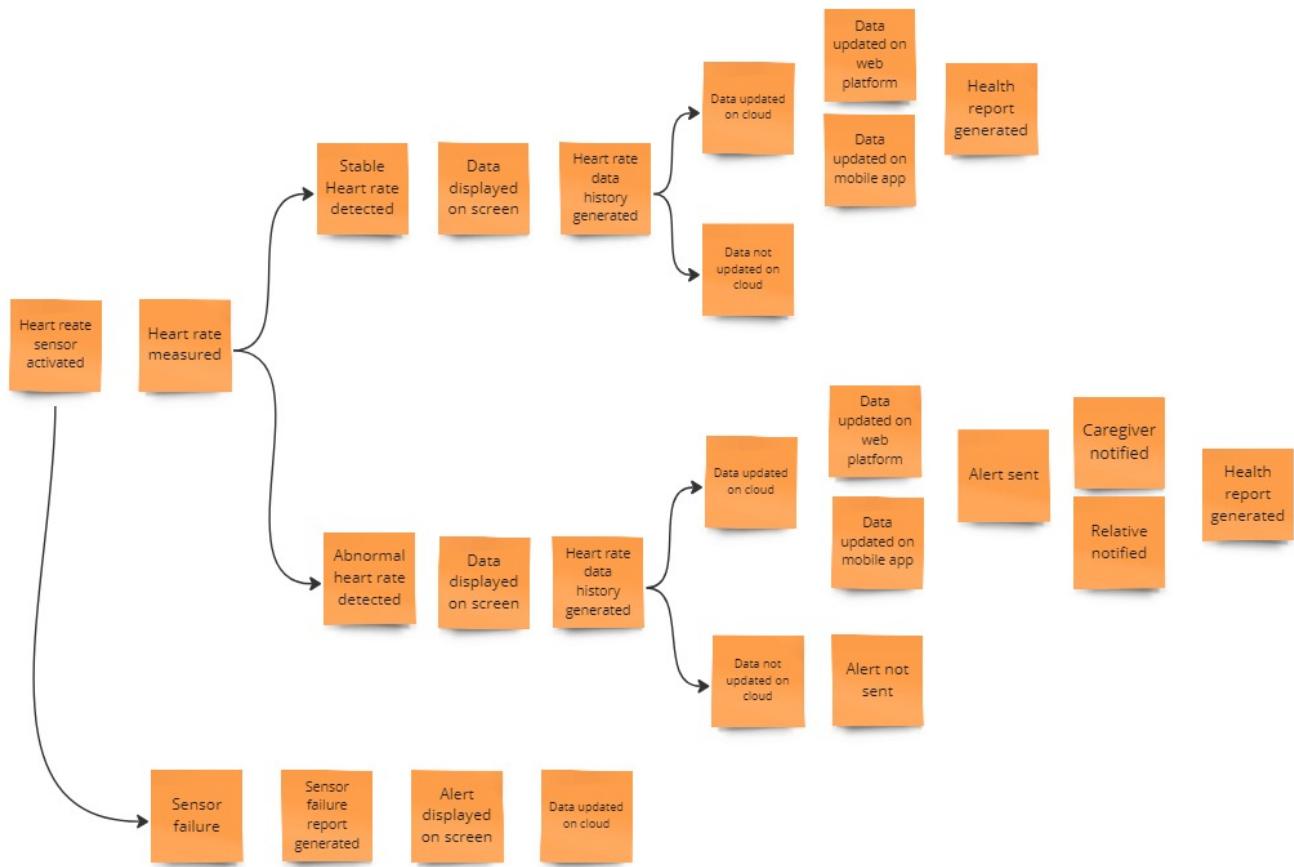
En el primer paso del EventStorming, se inicia con una sesión de lluvia de ideas enfocada en identificar los eventos del dominio relacionados con el negocio en estudio. Es fundamental formular estos eventos en tiempo pasado, describiendo lo que ha ocurrido en el sistema o proceso.

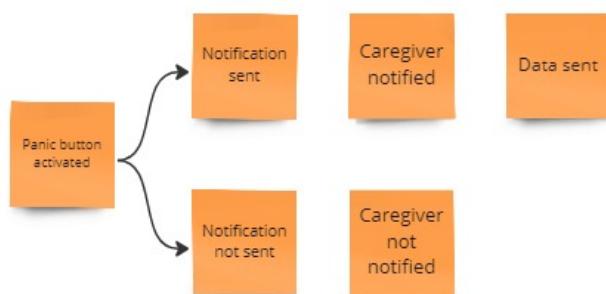
Step 1: Unstructured Exploration				
Movement sensor activated	Data sent	Data displayed on screen	Account created	Relative unlinked
Alert sent	Relative linked	Caregiver linked	Relative notified	Caregiver notified
Heart rate measured	Notification sent	Caregiver unlinked	Temperature measured	IoT Band configured
Subscription paid	Upgraded subscription to pro	Payment history requested	Account verified	Registration form sent
Alert displayed on screen	Rushed movement detected	Health report dashboard shown	Abnormal heart rate detected	Health report generated
Panic button activated	Presence in risky area detected	Data updated on web platform	Data updated on mobile app	Abnormal temperature detected
Connection with band lost	Sensor failure	Login requested	Data updated in cloud	IoT Band connected
Band firmware updated	Movement detected	Pulse data history generated	Temperature data history generated	IoT Band synchronized
Normal temperature detected	Heart rate sensor activated	Temperature sensor activated	Sensor failure report generated	Stable Heart rate detected

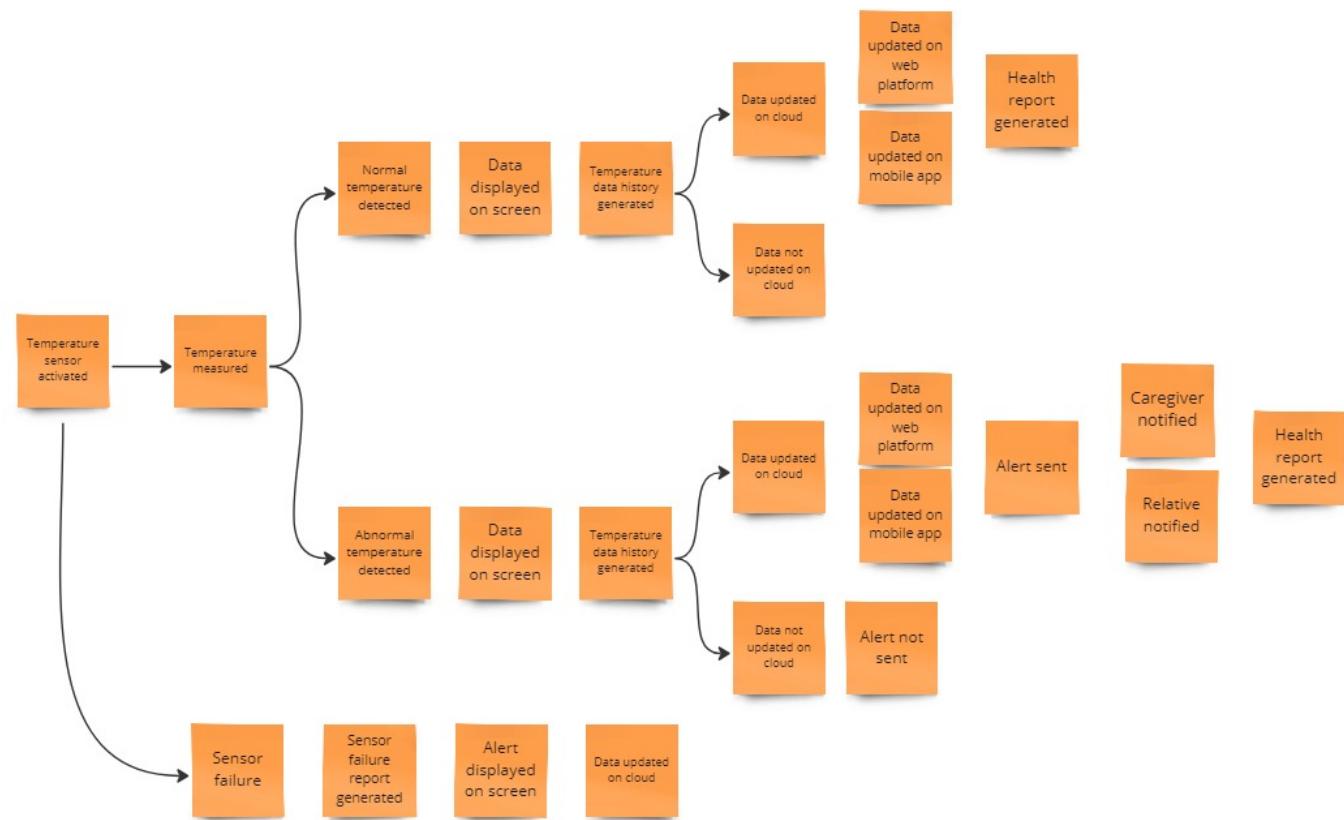
Realizamos una lluvia de ideas para identificar todos los eventos, datos y acciones relevantes que el sistema IoT para el cuidado de adultos mayores debería manejar utilizando una pulsera inteligente y aplicaciones web. Se incluyeron eventos relacionados con la activación de sensores y monitoreo de signos vitales, como la detección de movimiento, ritmo cardíaco y temperatura, para garantizar el seguimiento de la salud del usuario. También se consideraron acciones del sistema, como envío de alertas, notificaciones a familiares y cuidadores, así como la gestión de cuentas y configuración del dispositivo. Se añadieron posibles fallos en la conexión o en los sensores para asegurar la robustez del sistema ante fallos técnicos. Además, se contemplaron aspectos de gestión de datos y sus actualizaciones en diferentes plataformas, así como la interacción con servicios adicionales como suscripciones y pagos. Estos pos-its reflejan un mapeo inicial de todas las posibles interacciones y eventos que el sistema debería manejar para brindar una solución integral y segura para el cuidado de adultos mayores.

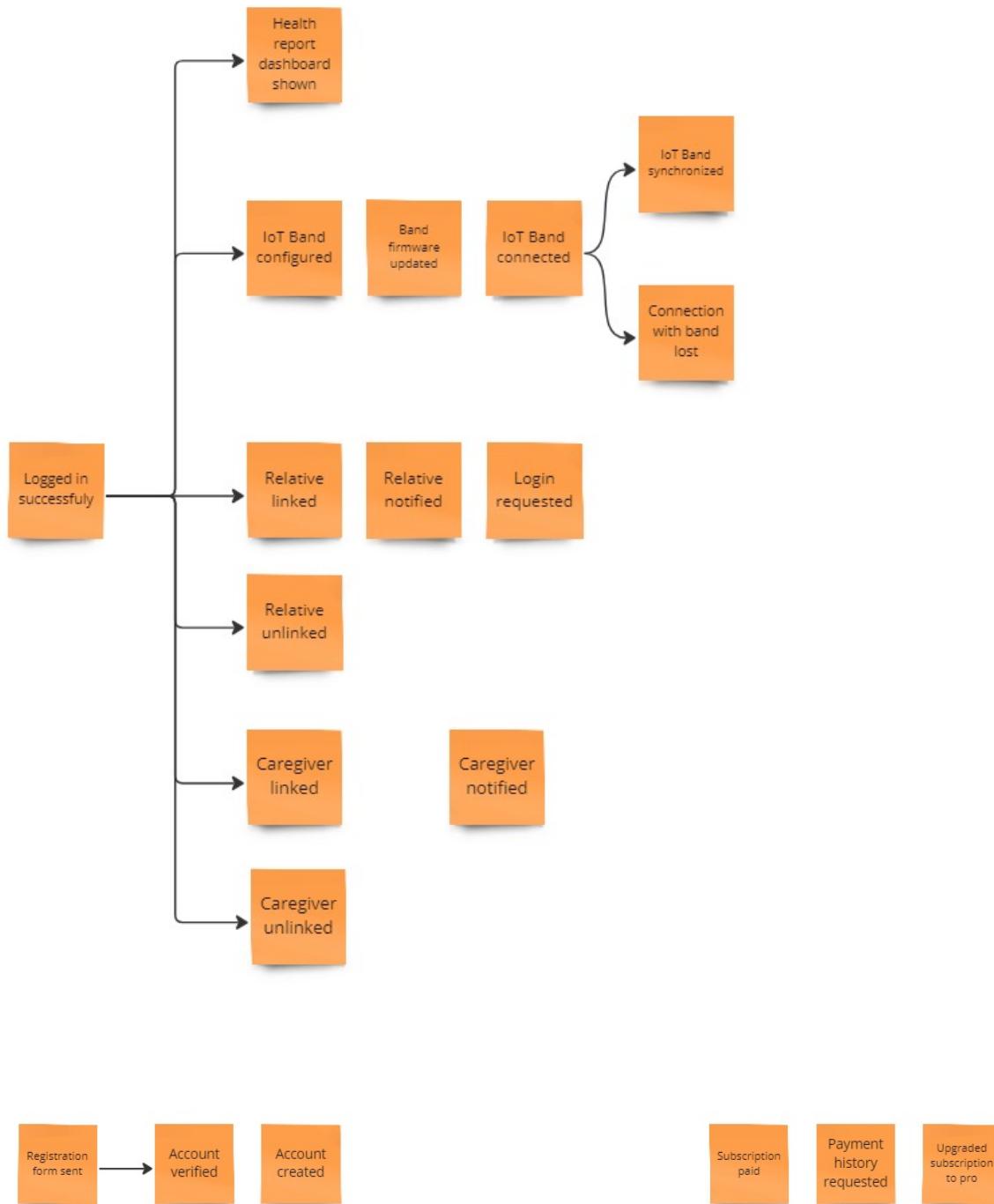
Step 2: Timelines

En este segundo paso, se revisan los eventos de dominio generados y se organizan en el orden en que ocurren dentro del dominio. Primero, se debe construir un happy path, es decir, un escenario en el que todo funciona correctamente y el proceso comercial es exitoso. Una vez que se ha completado este camino ideal, se pueden agregar escenarios alternativos que contemplen variaciones, fallos, o situaciones excepcionales.





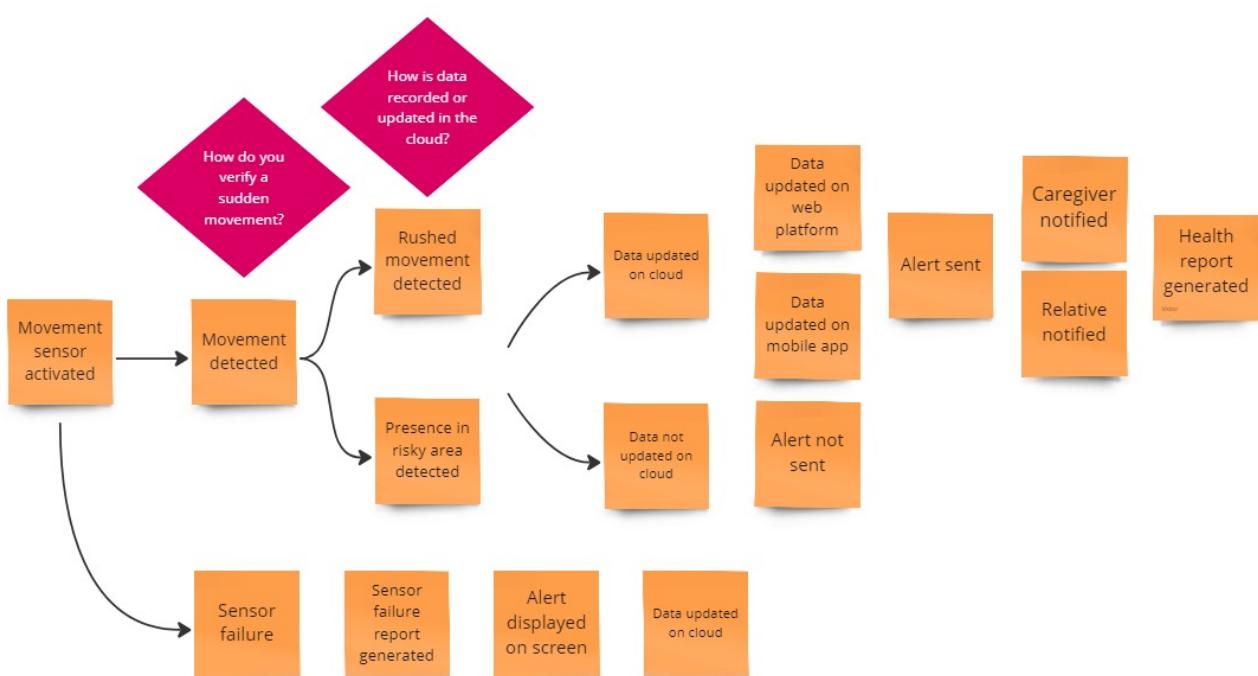
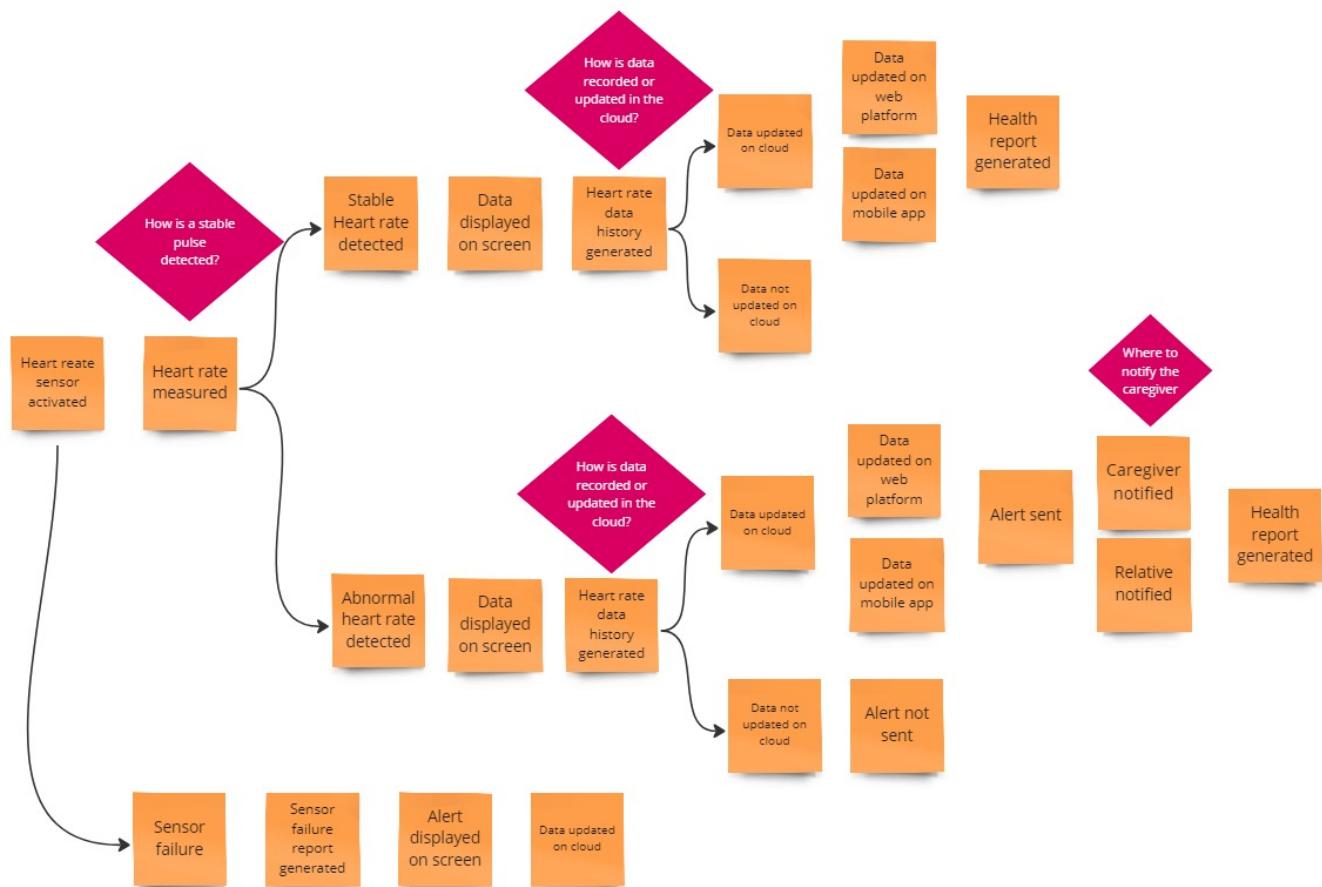


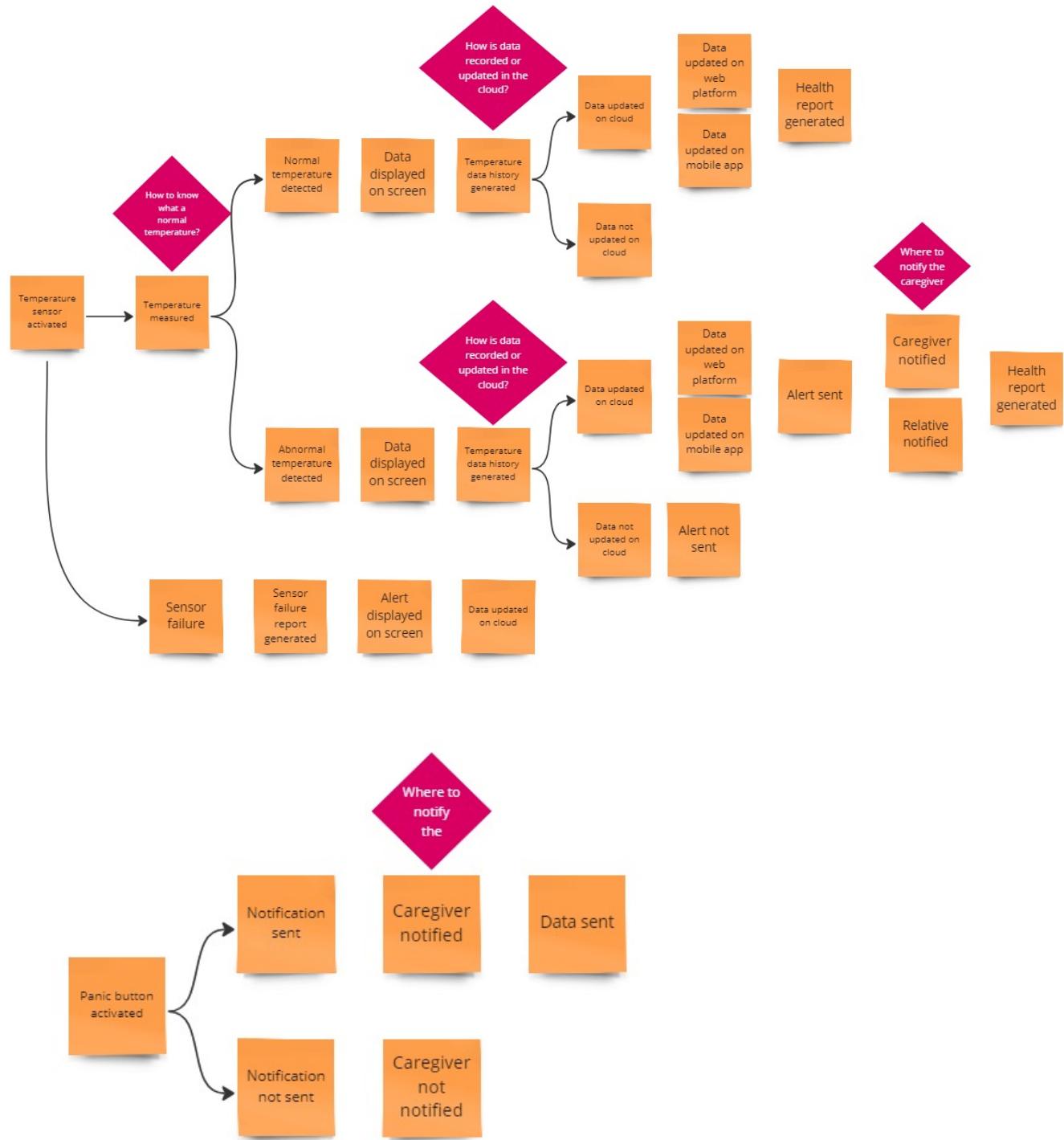


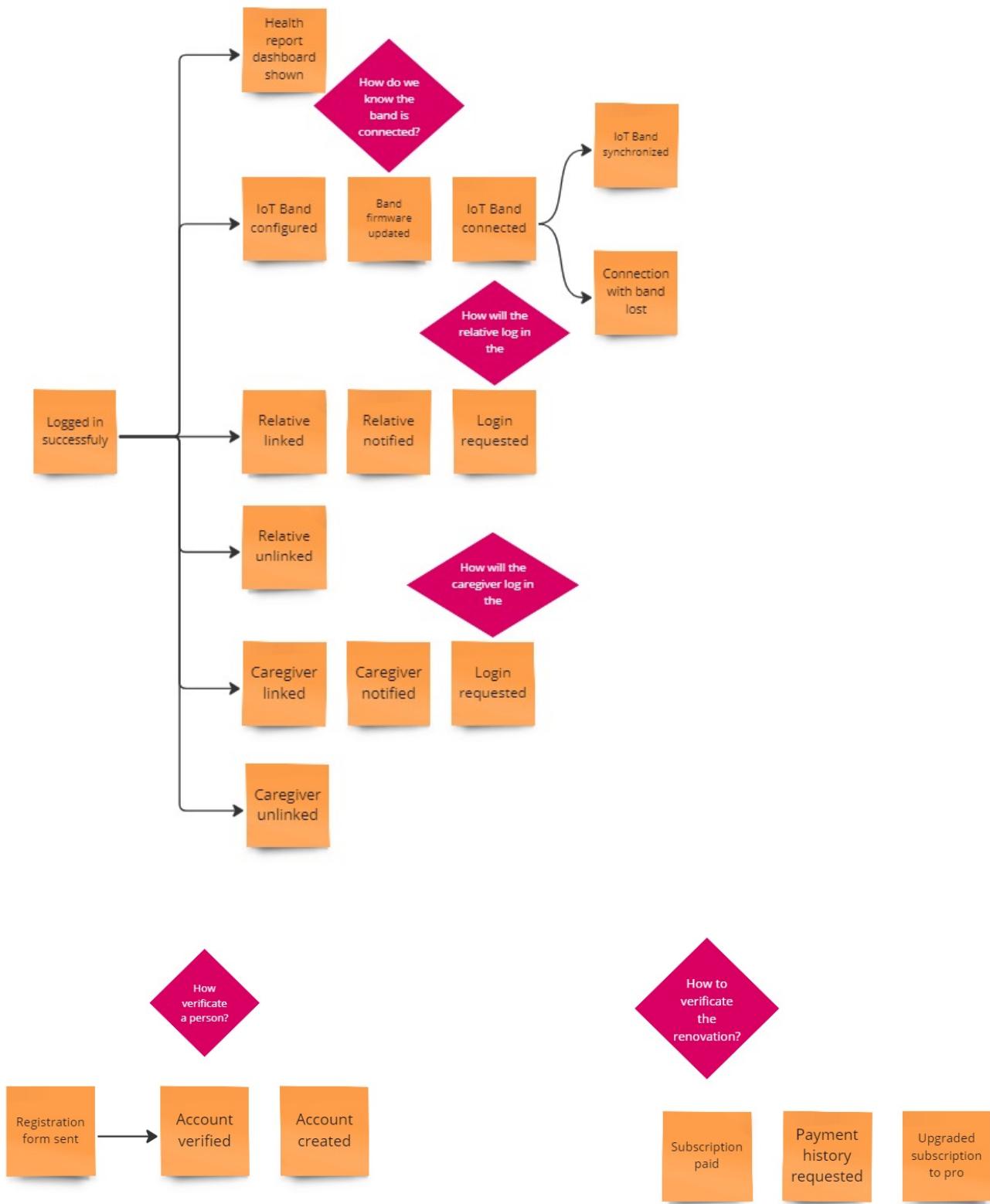
Hemos organizado la información generada en la fase de exploración no estructurada mediante flujos de eventos específicos, identificando así cómo debería reaccionar el sistema ante distintas situaciones relacionadas con el monitoreo de salud y seguridad del adulto mayor. Cada flujo visualiza un conjunto de eventos y acciones correlacionadas, desde la activación de sensores y la medición de parámetros vitales hasta la generación de alertas y la notificación a familiares o cuidadores. Se detallan diferentes escenarios como la detección de movimientos inusuales, alteraciones en la frecuencia cardíaca o temperatura, así como la gestión de fallos en el sistema. Además, se incluye la actualización de datos en distintas plataformas (nube, aplicaciones móviles y web) y la generación de reportes de salud. Este paso permite visualizar las posibles trayectorias que seguiría el sistema en respuesta a cada evento, estableciendo la base para desarrollar un flujo de trabajo eficiente y cohesivo que garantice una respuesta rápida y adecuada a cada situación monitorizada.

Step 3: Paint Points

Después de organizar los eventos en una línea de tiempo, aprovechamos esta vista general para identificar puntos de interés a lo largo del proceso. Estos puntos de interés pueden incluir cuellos de botella, pasos manuales que podrían ser automatizados, falta de documentación o carencias de conocimiento del dominio.



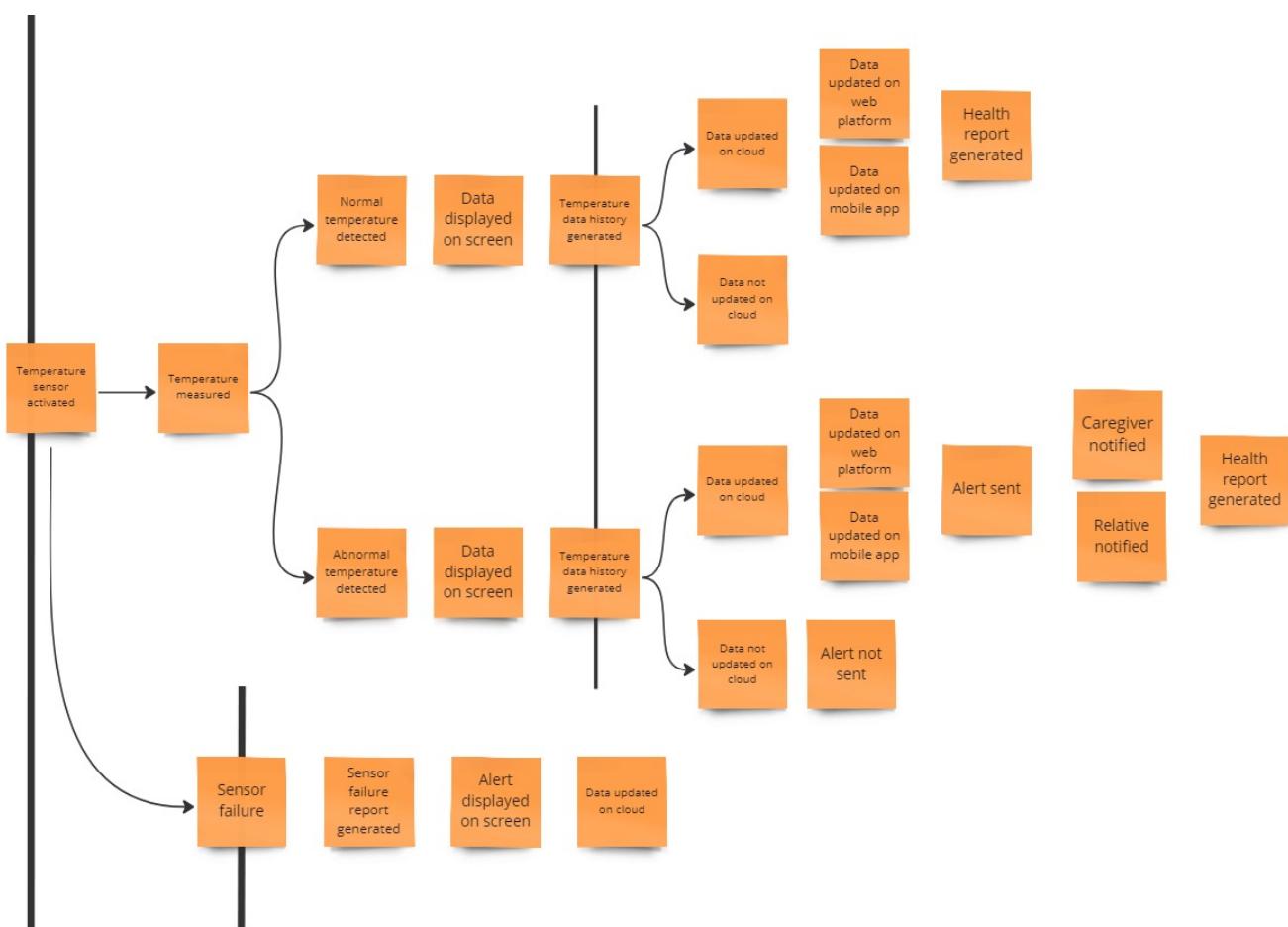
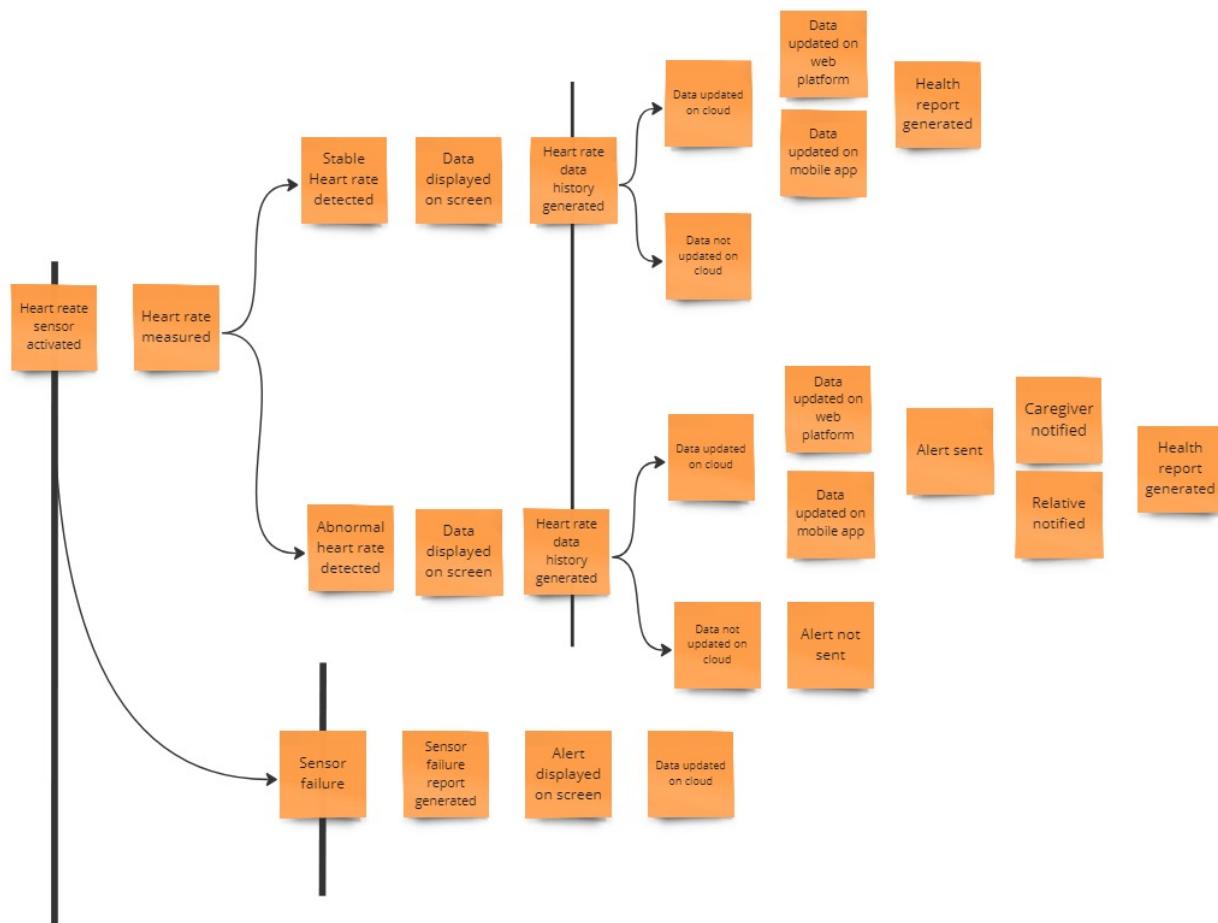


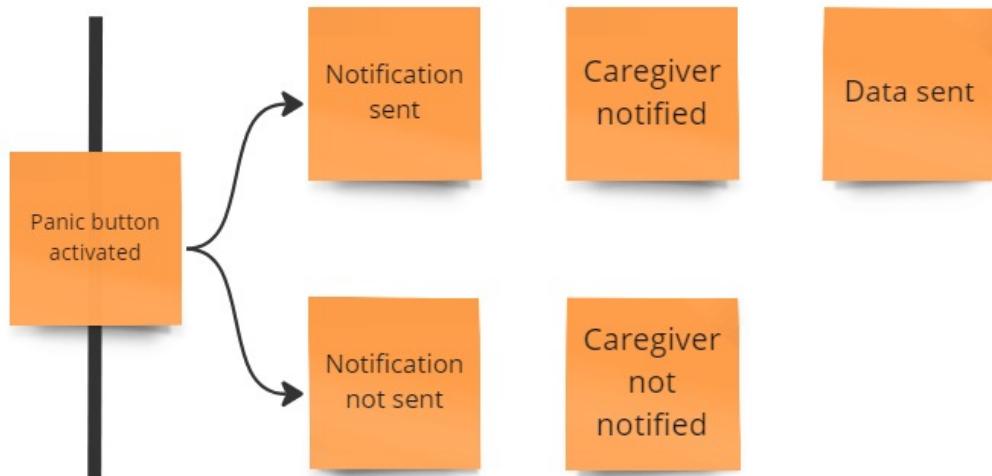
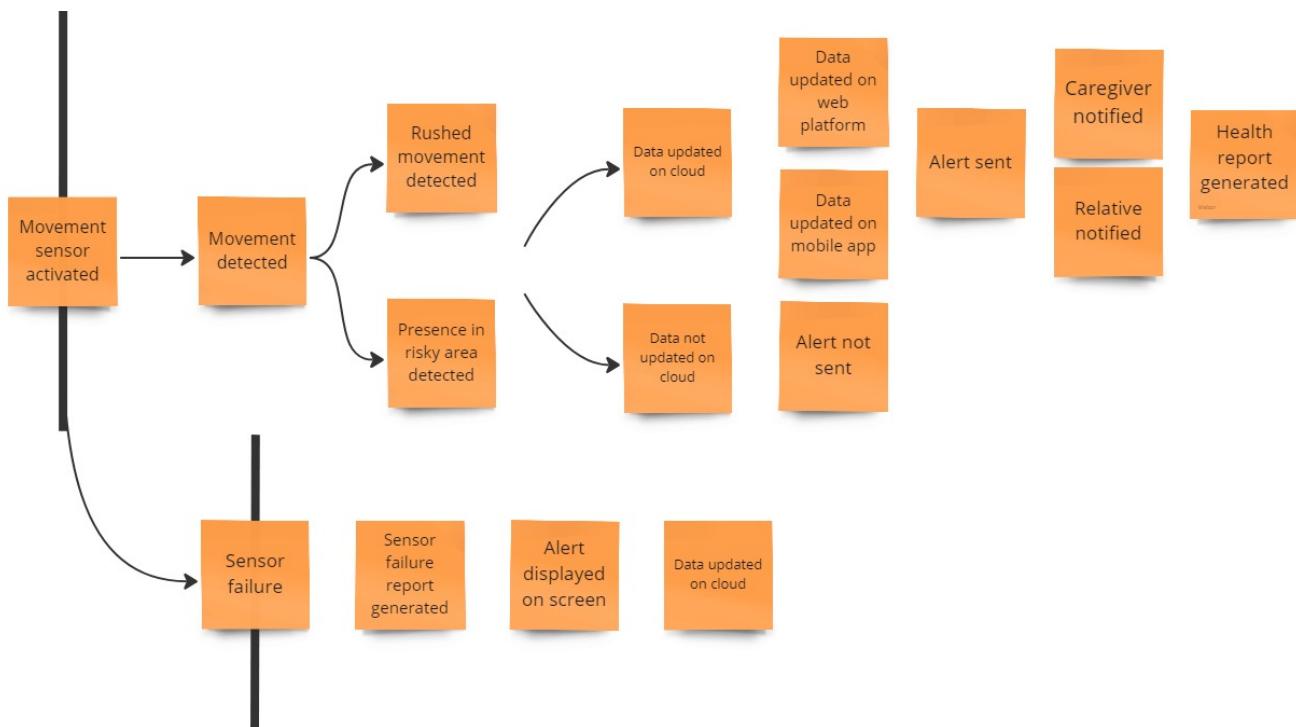


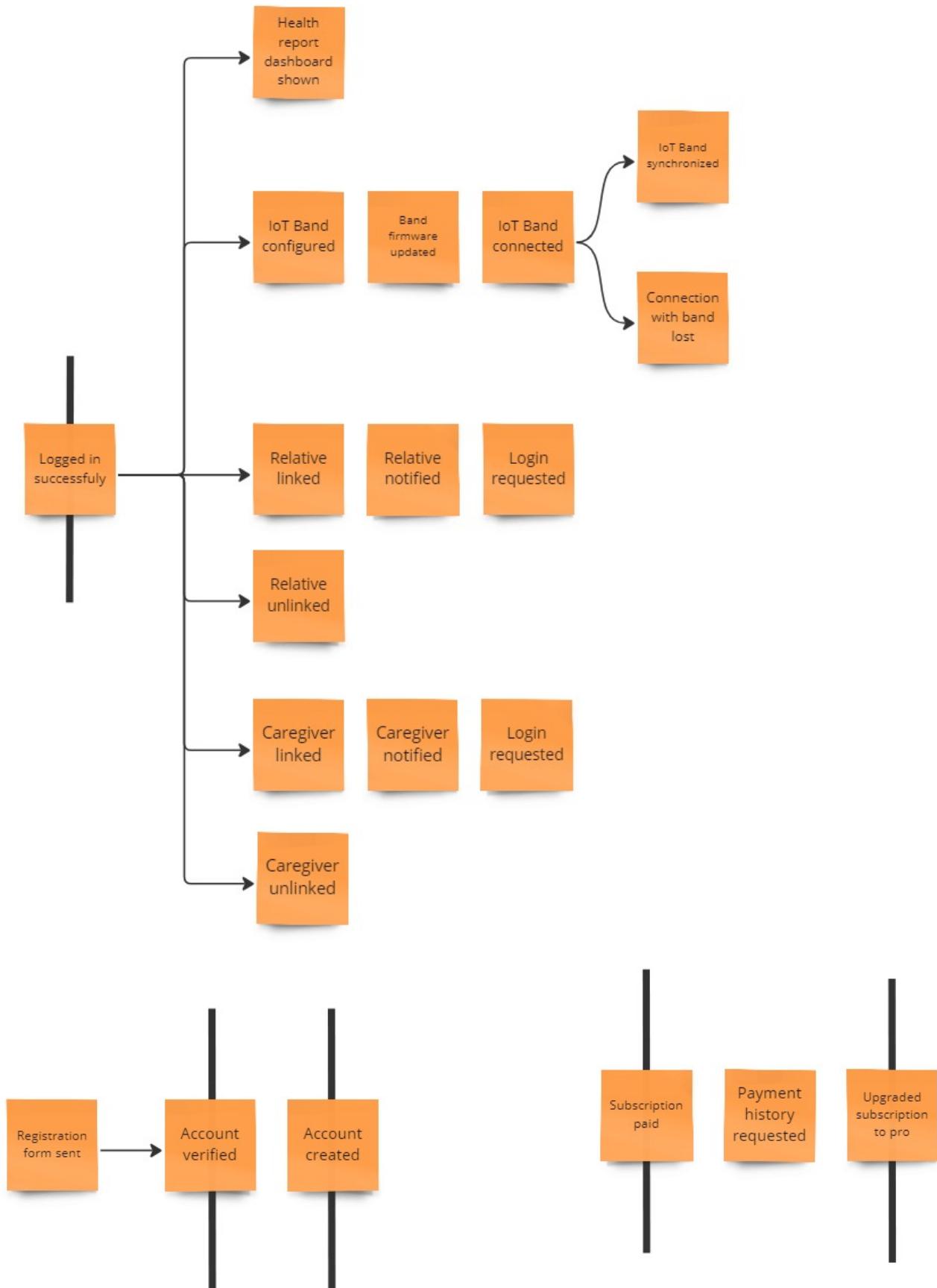
Hemos identificado y documentado los puntos críticos y decisiones clave en cada flujo de eventos. Estos flujos se detallan con preguntas clave que deben ser respondidas para garantizar el correcto funcionamiento del sistema, como la forma de detectar un pulso estable, la verificación de un movimiento repentino o la identificación de una temperatura normal. Además, se analiza cómo se registran y actualizan los datos en la nube, cómo se notifica al cuidador en caso de eventos anómalos. Se incluyen flujos adicionales para la gestión de la conexión de la banda IoT, la configuración y vinculación de cuentas, así como la verificación de usuarios y renovaciones de suscripción. Este paso permite una mayor claridad sobre los requerimientos específicos del sistema, asegurando que se cubran todos los posibles escenarios y puntos de fallo, y se establezcan procesos claros para la toma de decisiones en situaciones críticas, mejorando así la robustez y la confiabilidad de la solución IoT propuesta.

Step 4: Pivotal Points

Una vez que tenemos nuestra línea de eventos completa, incluyendo los pain points, buscamos eventos comerciales clave que marquen un cambio en el contexto o en la fase del proceso. Estos se denominan eventos principales y los señalamos con una barra vertical que separa los eventos anteriores de los posteriores a dicho evento.





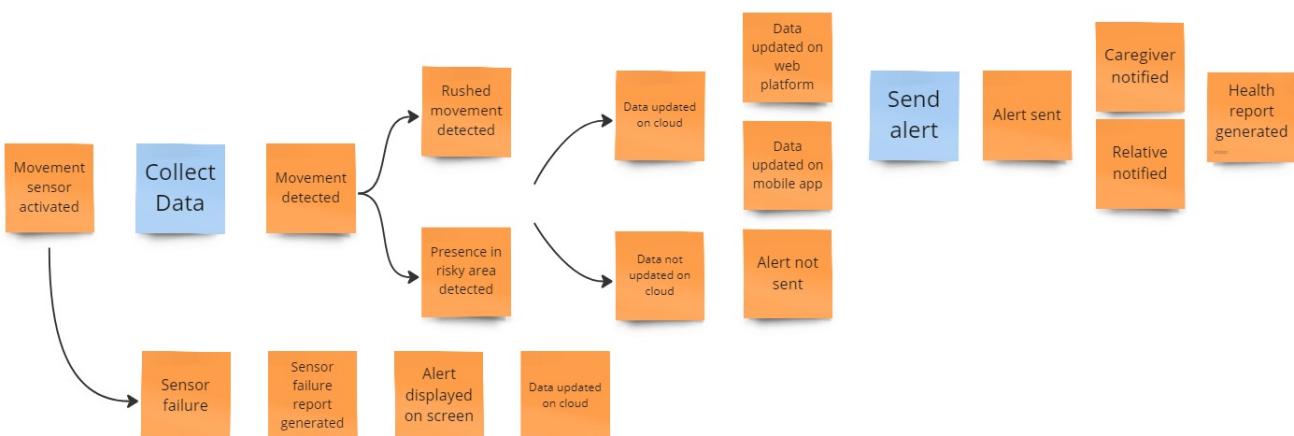
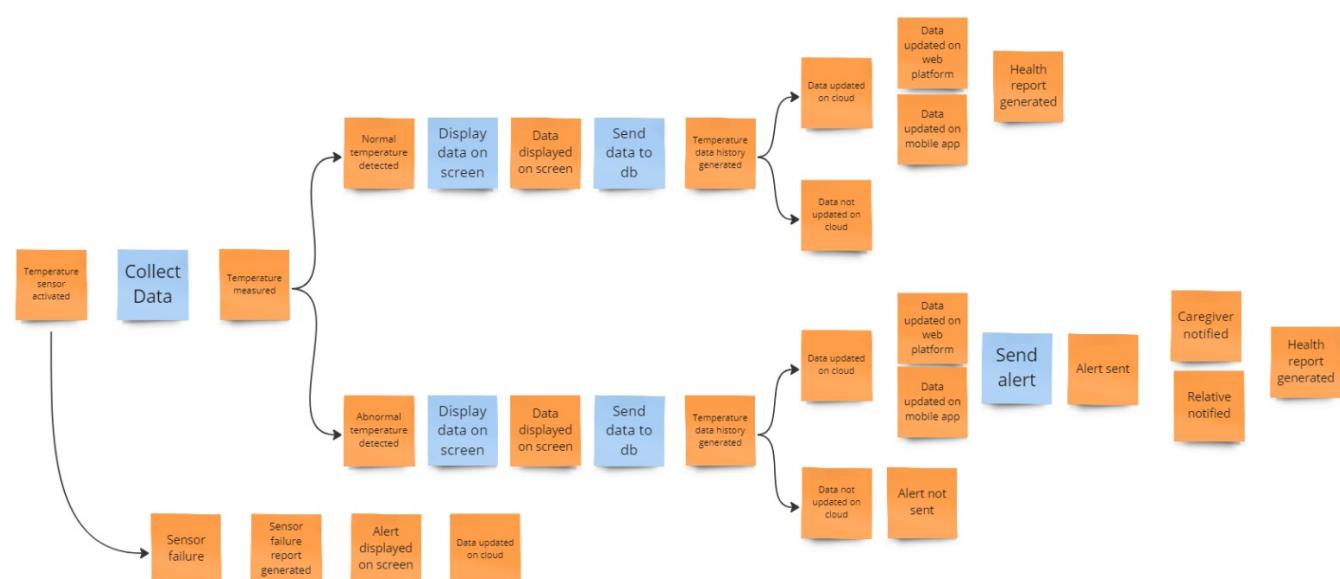
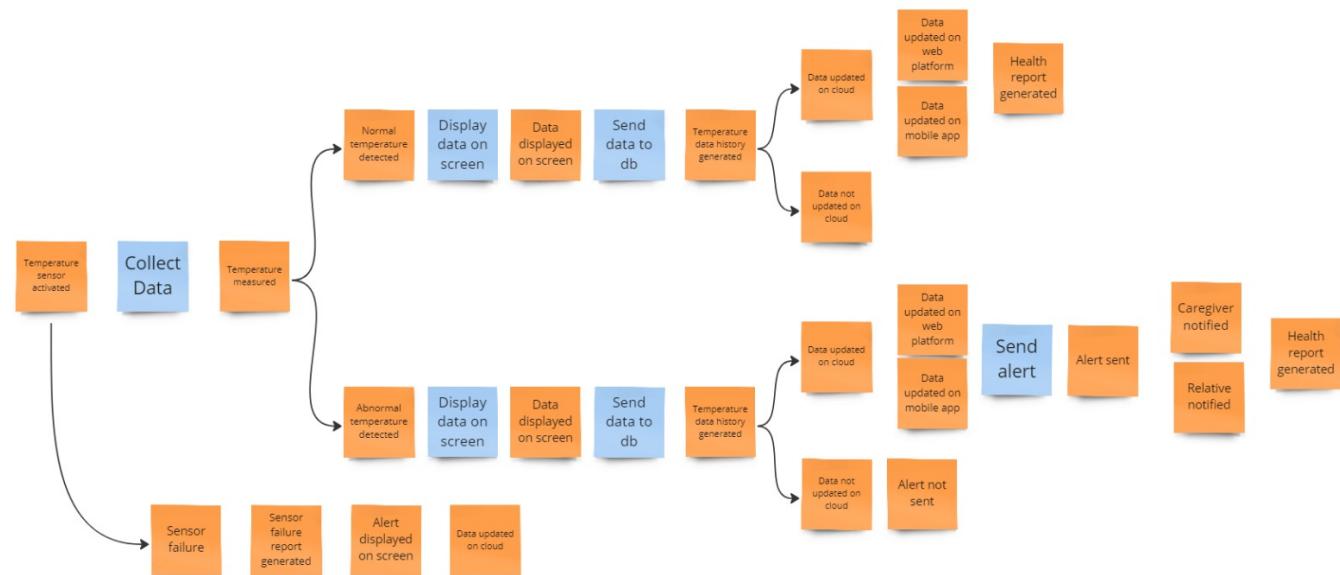


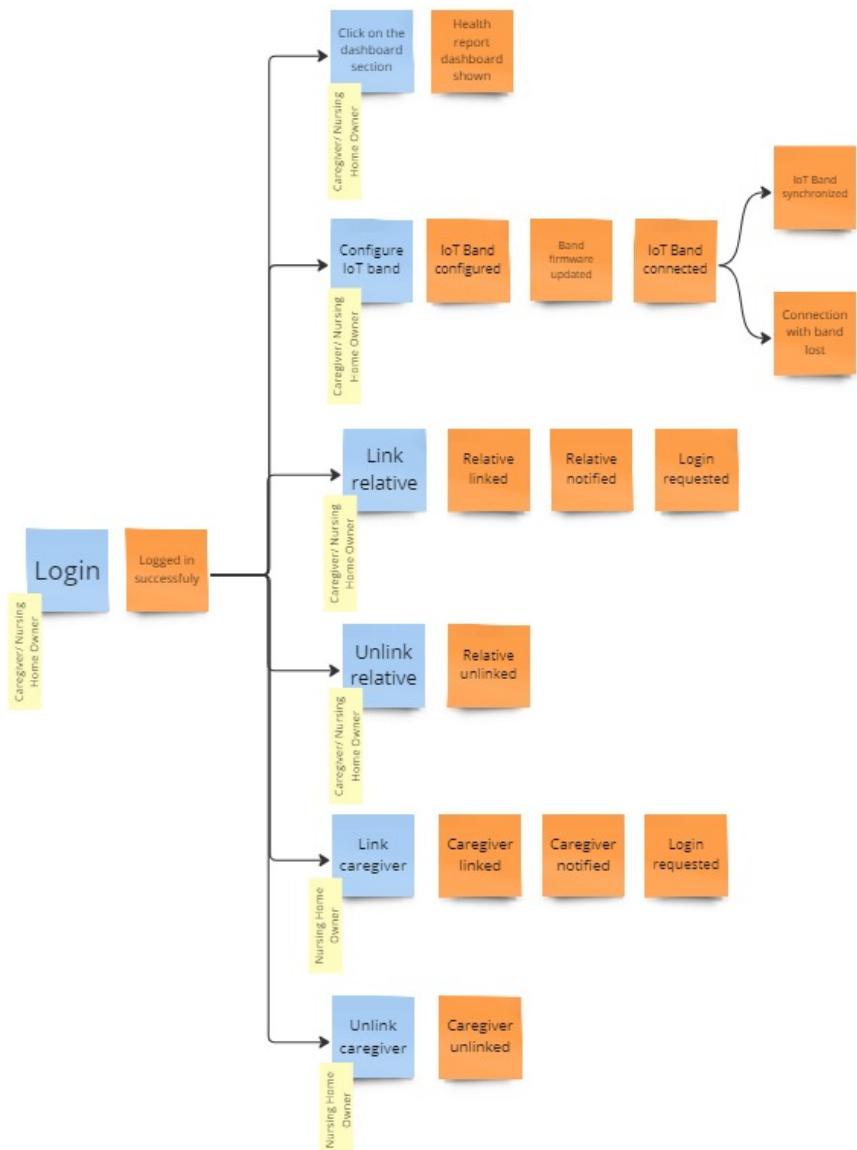
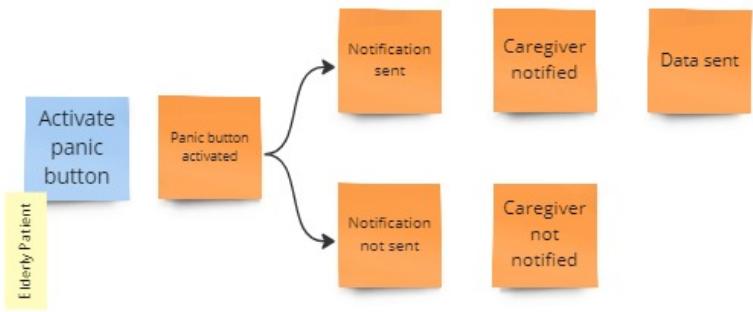
Hemos identificado los puntos críticos (pivotal points) que son esenciales para la correcta operación y toma de decisiones del sistema IoT de cuidado de adultos mayores. Estos puntos son momentos donde se requiere validar decisiones importantes para garantizar que el sistema funcione de manera adecuada. Por ejemplo, se definieron puntos de verificación como la detección de ritmo cardíaco y temperatura estables, la actualización de datos en la nube y la generación de alertas a cuidadores y familiares. También se incluyeron decisiones sobre la configuración de la banda IoT, la verificación de usuarios y la gestión de suscripciones. Estos puntos críticos permiten asegurar que, ante cualquier eventualidad, el

sistema tenga claras las acciones a seguir para brindar una respuesta eficiente y oportuna, minimizando el riesgo de fallos y mejorando la confiabilidad del sistema.

Step 5: Commands

En este paso también introducimos los comandos, los cuales describen la causa de un evento o el flujo de eventos. A diferencia de los eventos de dominio, los comandos son expresados en modo imperativo, describiendo las operaciones que deben ejecutarse en el sistema.



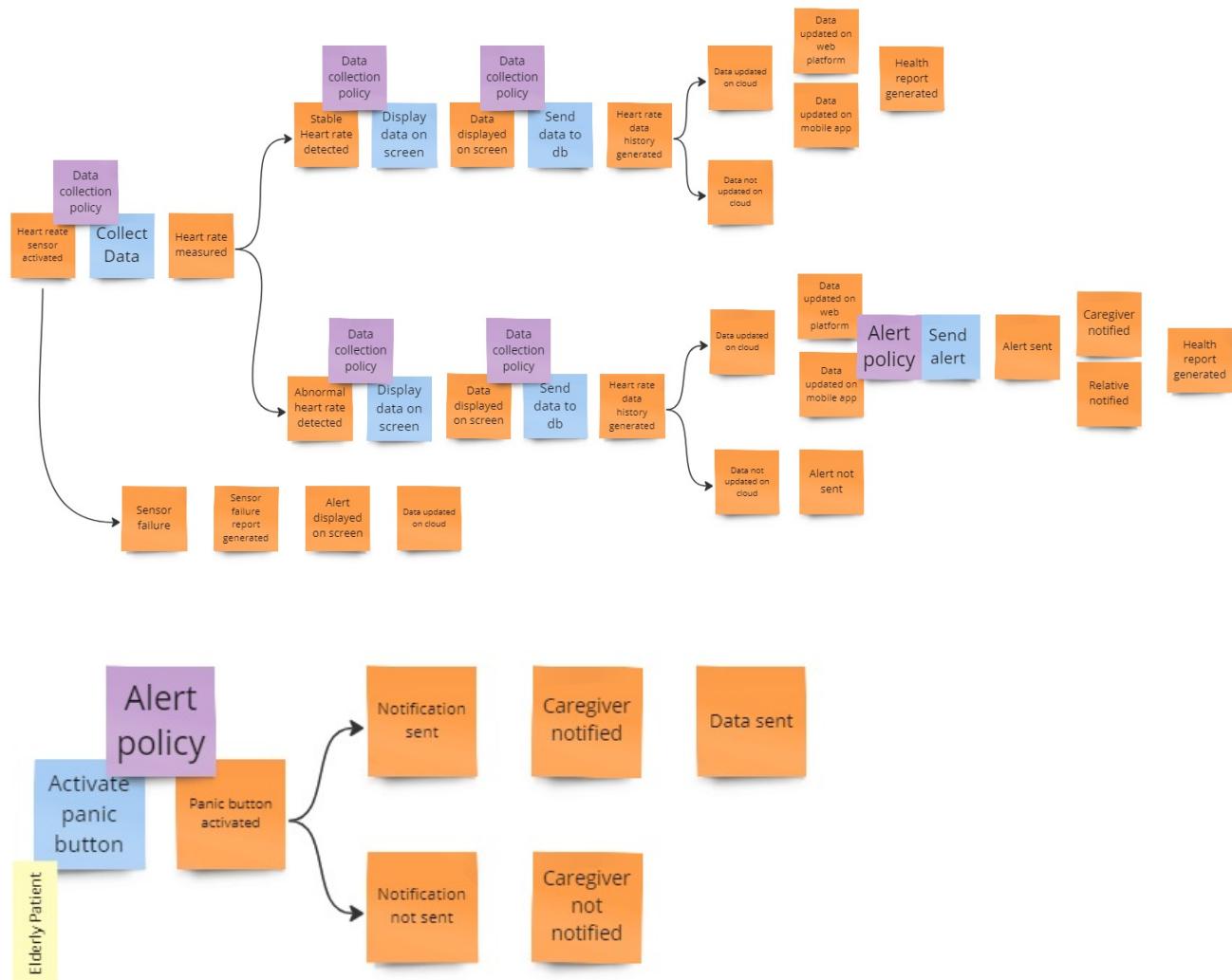


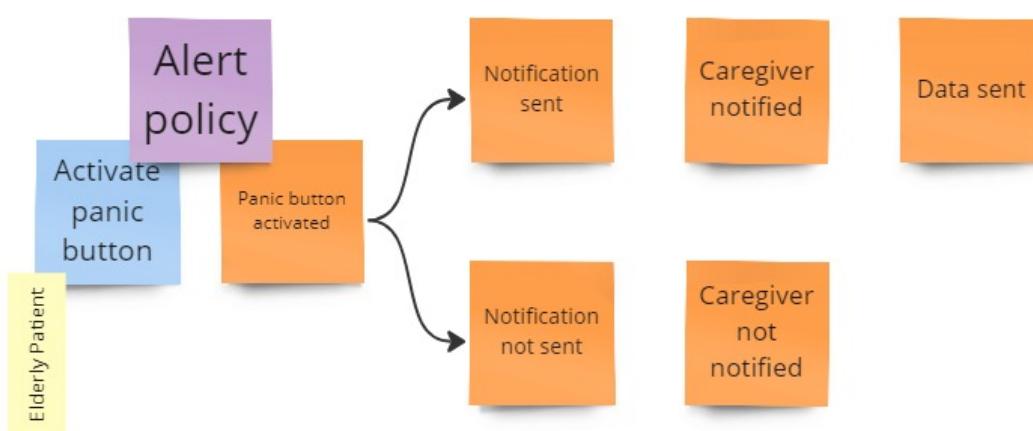
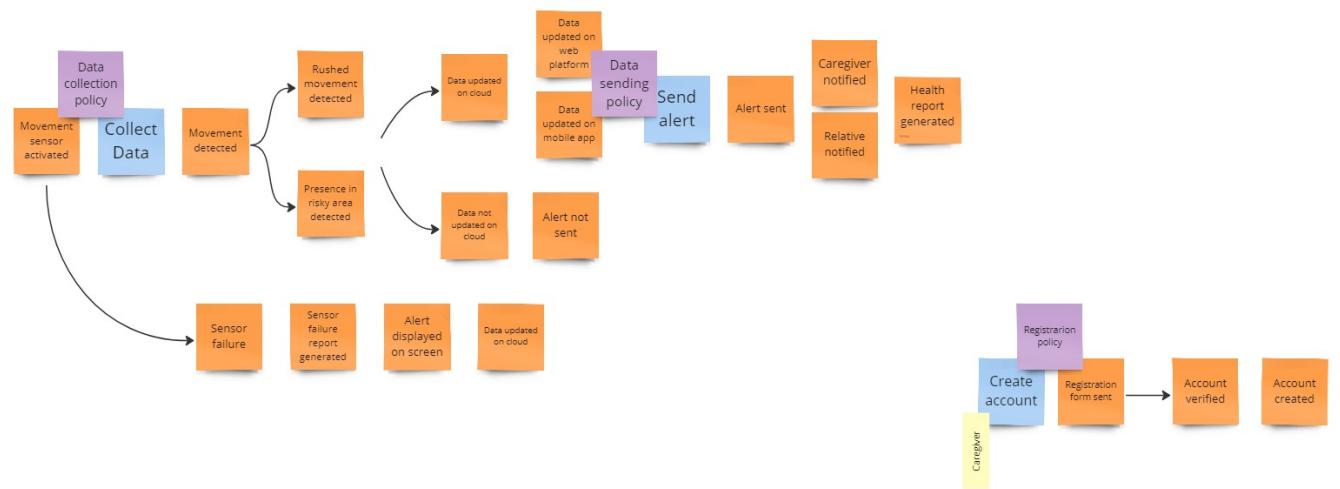
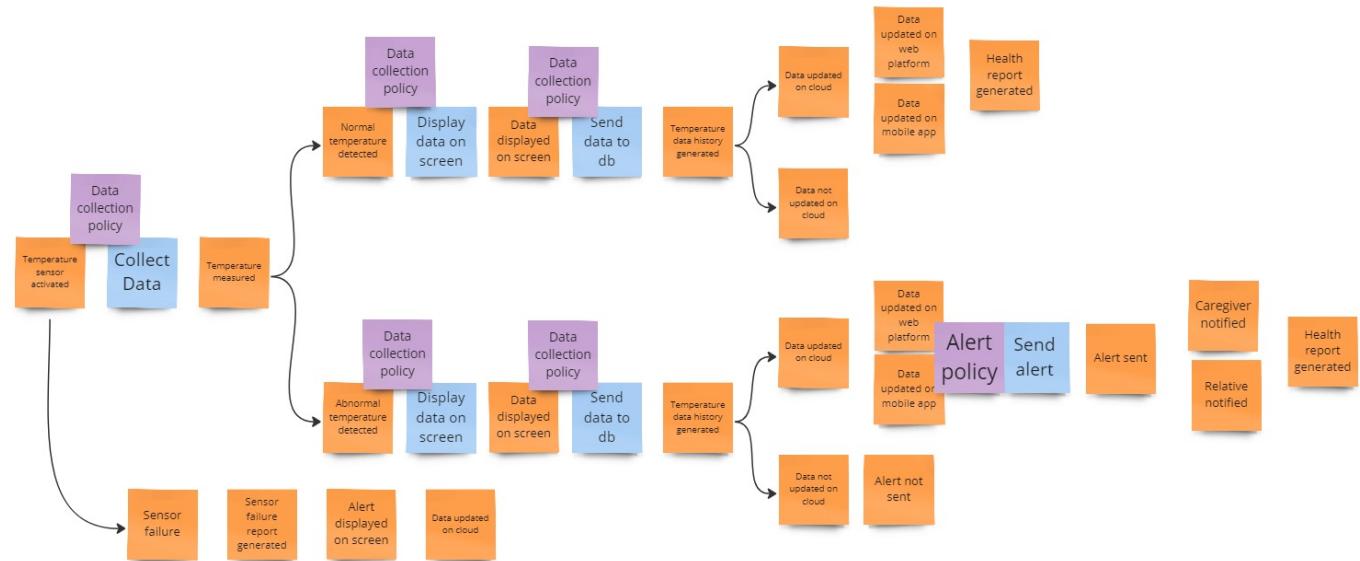


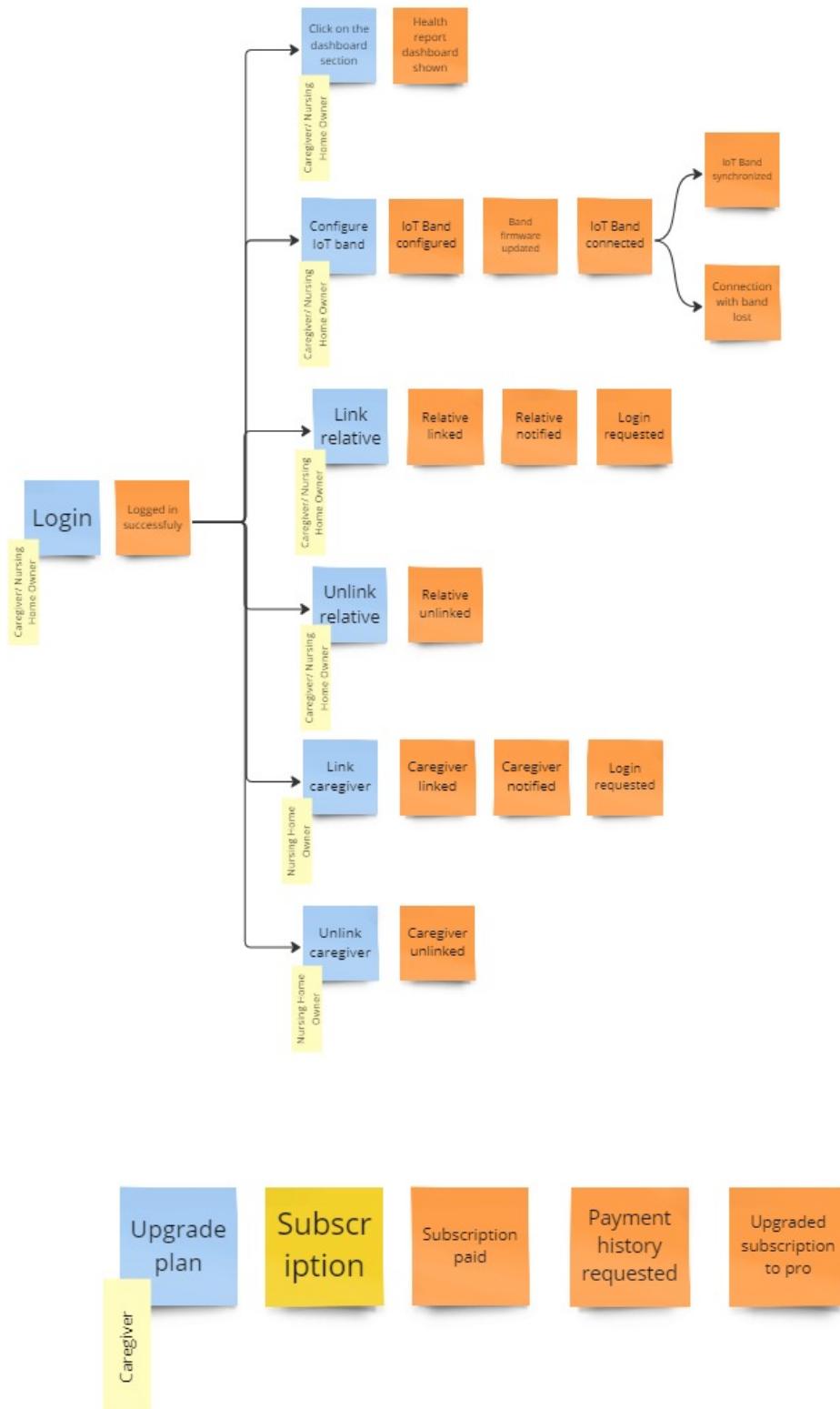
Estructuramos las funcionalidades y acciones clave del sistema IoT para el cuidado de adultos mayores, identificando los actores involucrados y sus interacciones con el sistema. Se definieron procesos específicos para la recolección de datos de los sensores, su visualización y almacenamiento en la base de datos, y la generación de alertas en caso de situaciones anómalas. Además, se detallaron los flujos de registro, vinculación y desvinculación de usuarios, cuidadores y familiares, así como la configuración de la banda IoT y su sincronización. Este paso permitió mapear con claridad cómo cada actor interactúa con el sistema y cómo se gestionan los datos críticos, asegurando que todas las funcionalidades necesarias estén integradas para ofrecer una solución completa y eficiente.

Step 6: Policies

En este punto, buscamos automation policies (políticas de automatización) que puedan ejecutar estos comandos. Esto significa que un evento específico del dominio desencadena automáticamente la ejecución de un comando. En otras palabras, cuando ocurre un evento determinado, el comando correspondiente se ejecuta de manera automática.





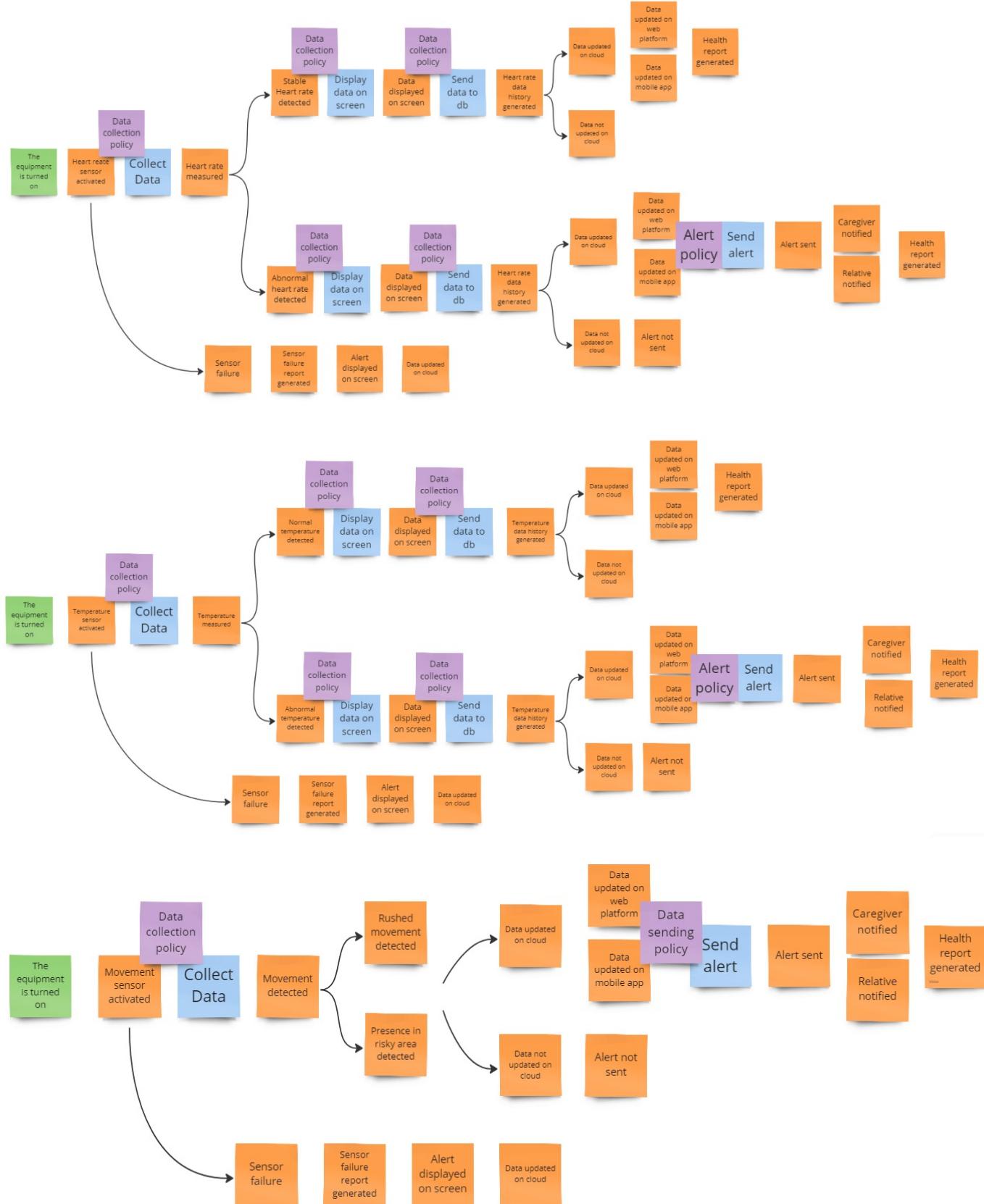


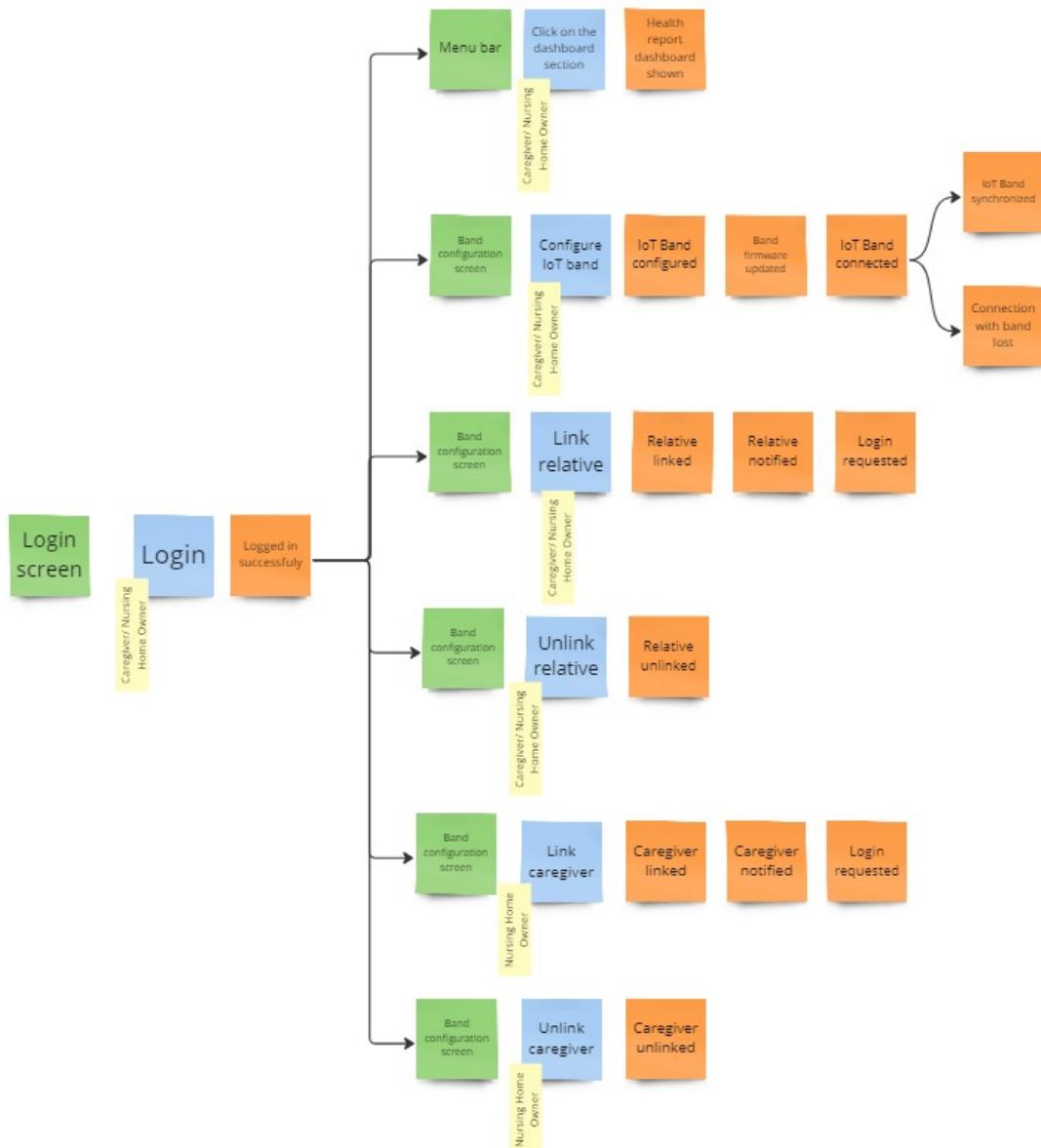
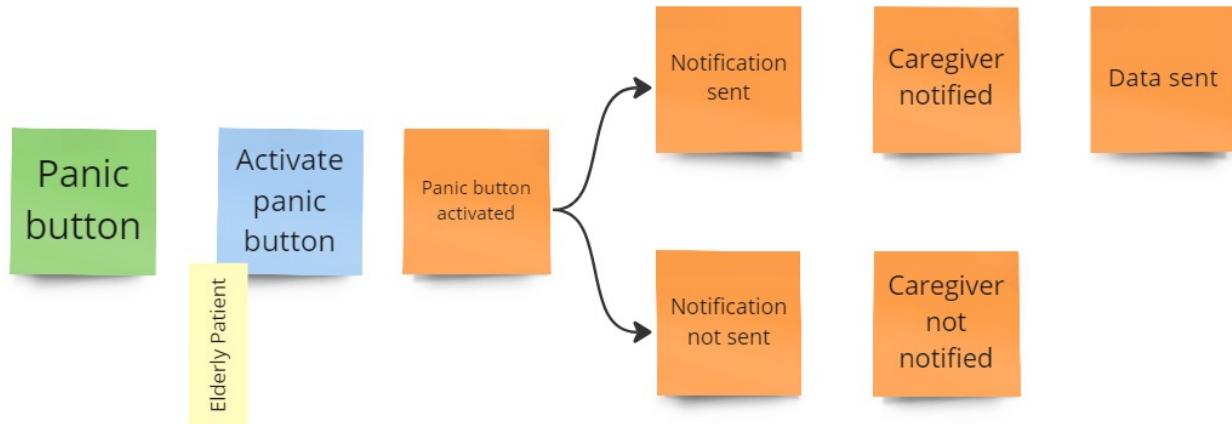
Definimos y organizamos las políticas de recolección y envío de datos, alertas, registro de usuarios y gestión de suscripciones. Se establecieron los eventos para la activación de sensores, detección de condiciones críticas, generación de alertas y notificaciones a cuidadores y familiares, así como el manejo de fallos de sensores. También se estructuraron los procesos para la creación, verificación y administración de cuentas, y la actualización de planes de suscripción, asegurando una correcta integración y flujo de información en la aplicación.

Step 7: Read Models

En este paso, introducimos el modelo de lectura, que es la representación de datos del dominio que un agente utiliza para decidir si debe ejecutar o no un comando. Por esta razón, definimos una vista de datos para cada comando, como monitores del sistema, informes, notificaciones, entre

otros.



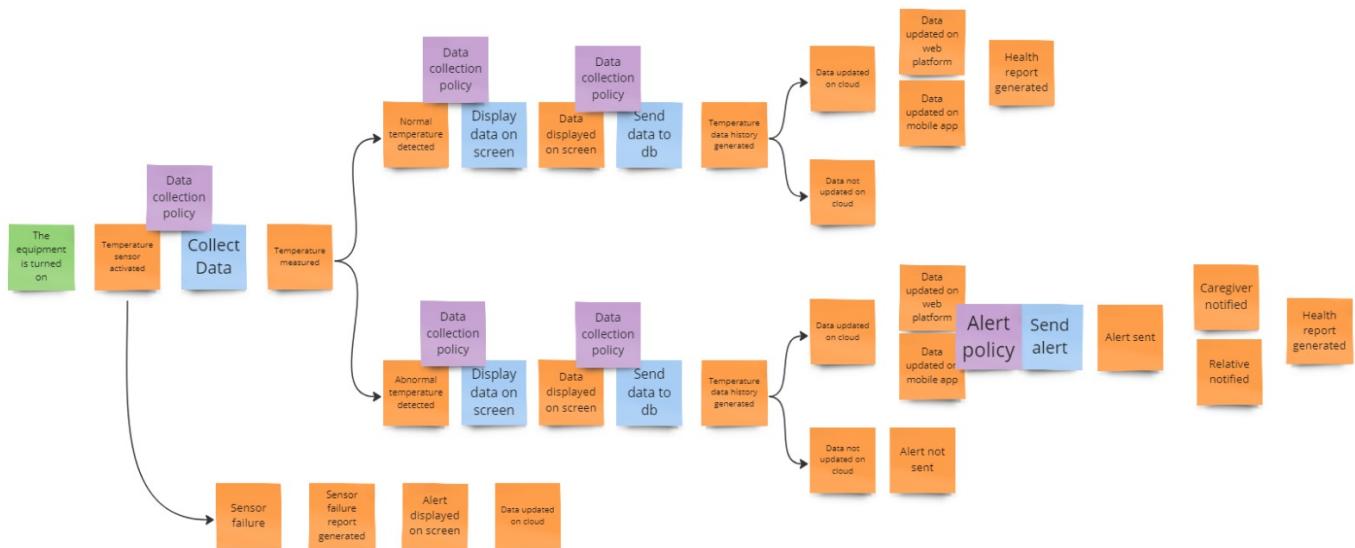
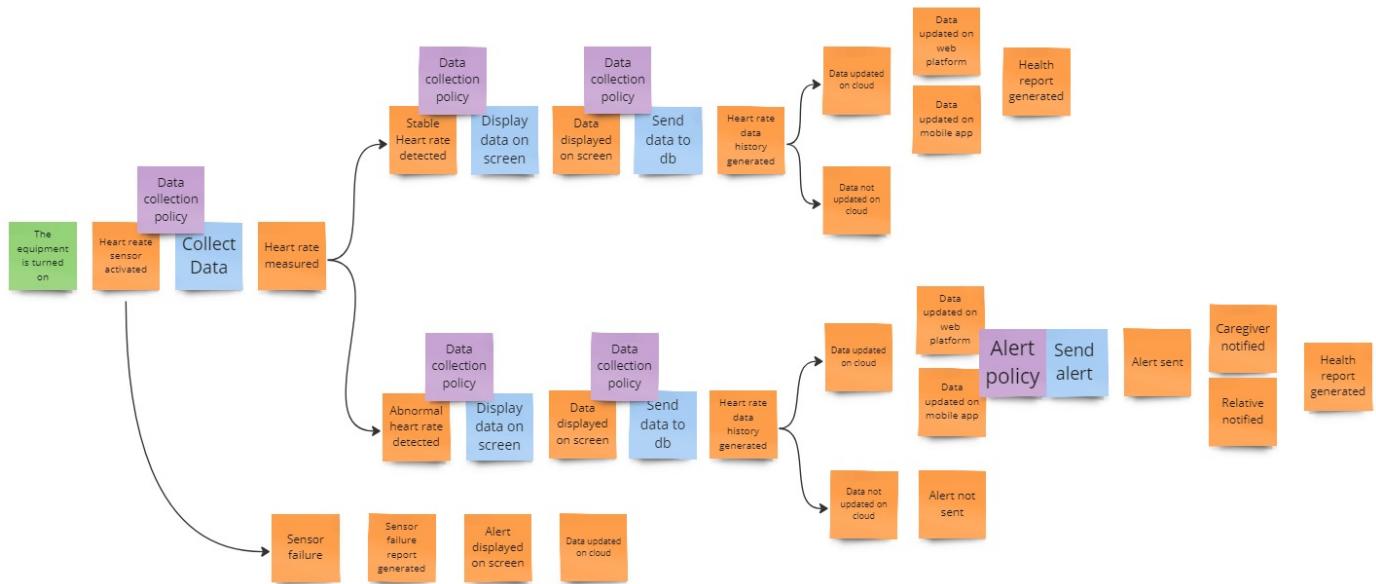


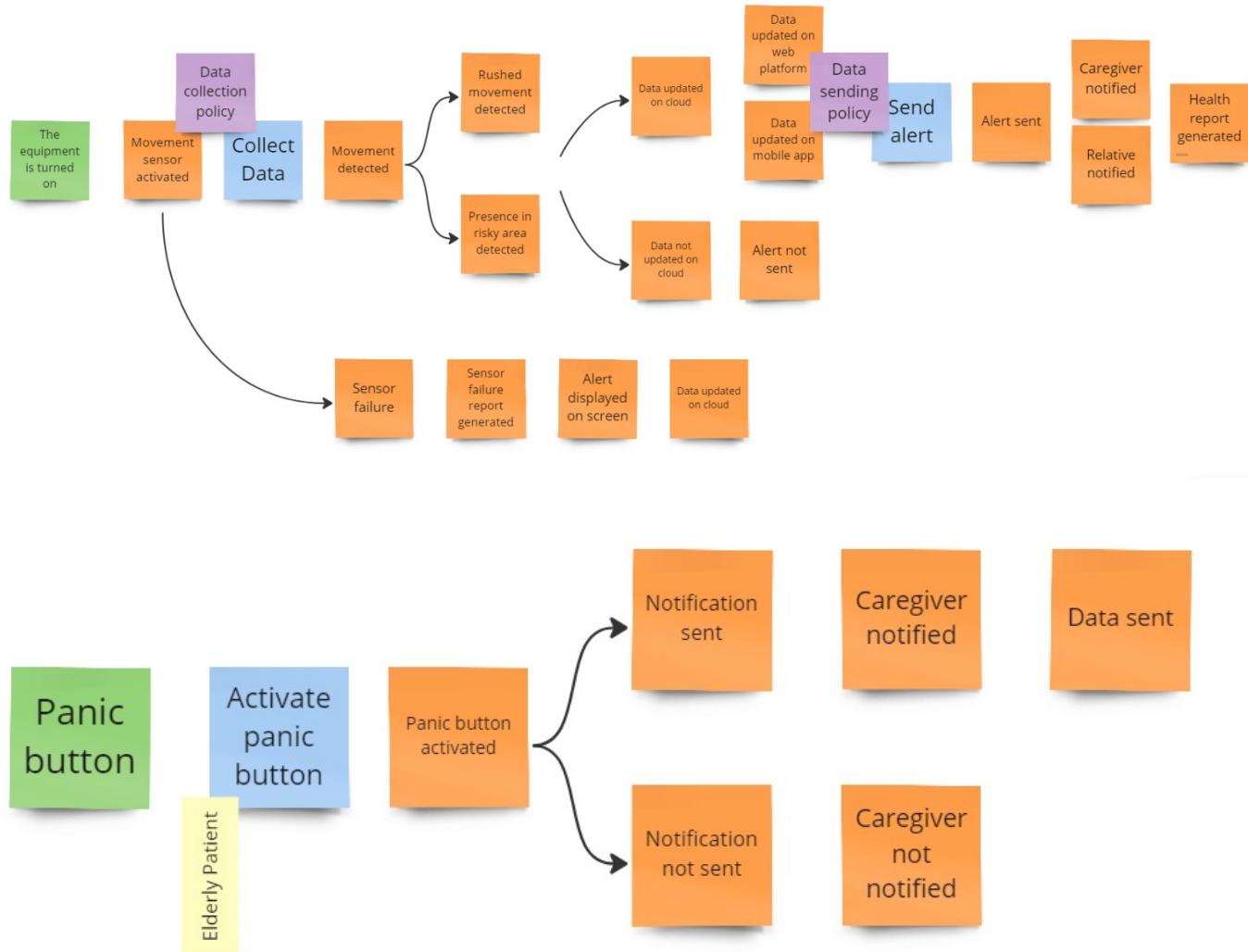


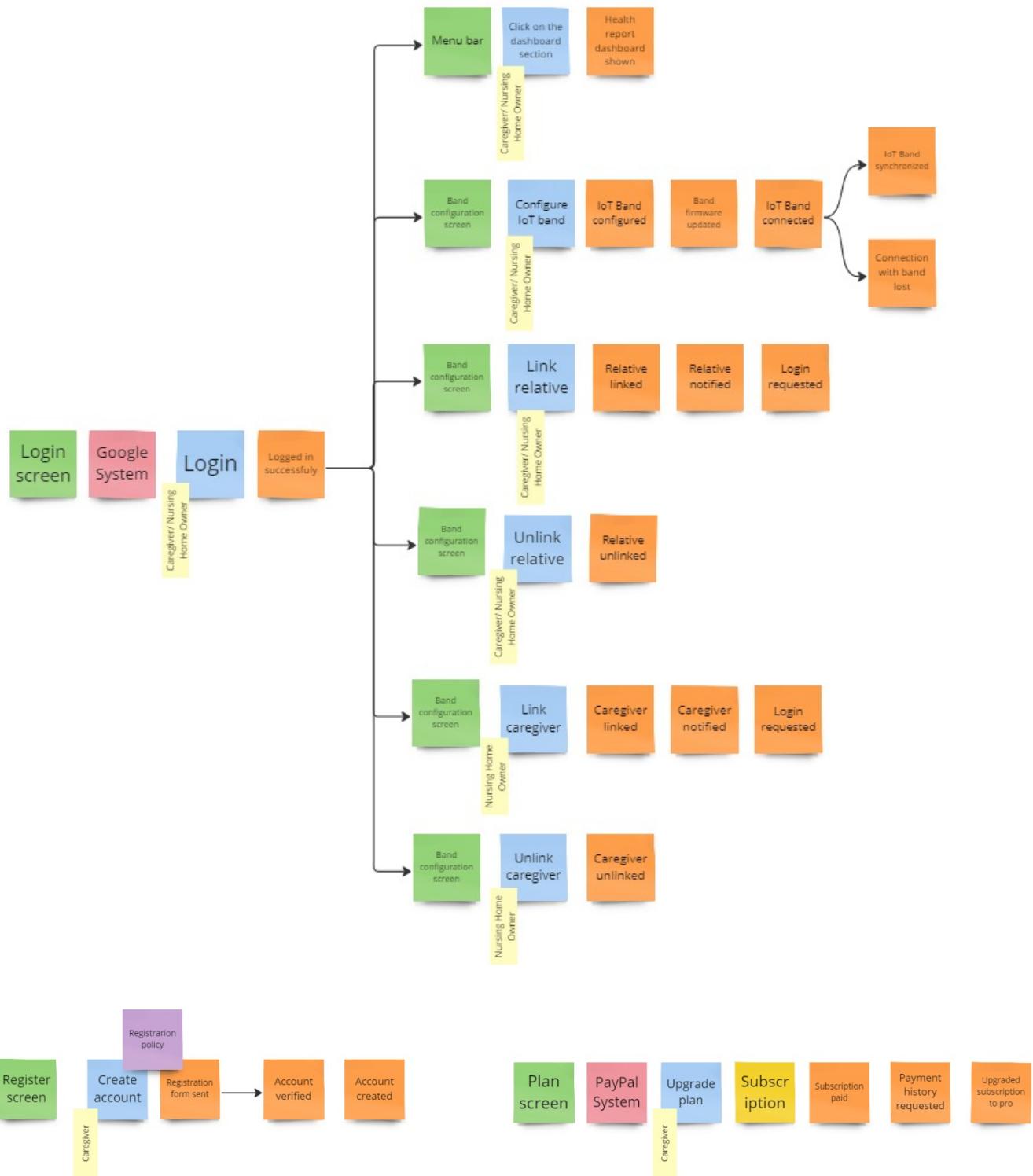
Identificamos las políticas de negocio y cómo interactúan con los eventos y comandos ya establecidos. Cada política fue colocada en relación con un evento o comando específico, con el objetivo de definir reglas y restricciones que deben cumplirse para la ejecución de ciertas acciones. Estas políticas incluyen aspectos como la recolección de datos, el envío de alertas y el registro de cuentas, garantizando que las operaciones del sistema se realicen bajo ciertos parámetros. Además, se introdujeron políticas para la gestión de registros, suscripciones y actualizaciones de planes, detallando cómo y cuándo deben aplicarse estas reglas en función del contexto y estado del sistema.

Step 8: External Systems

A continuación, completamos el modelo incluyendo los sistemas externos. Un sistema externo es cualquier sistema que no pertenece al dominio en el que estamos trabajando. Estos sistemas pueden ejecutar comandos (entrada) o recibir notificaciones sobre eventos (salida).



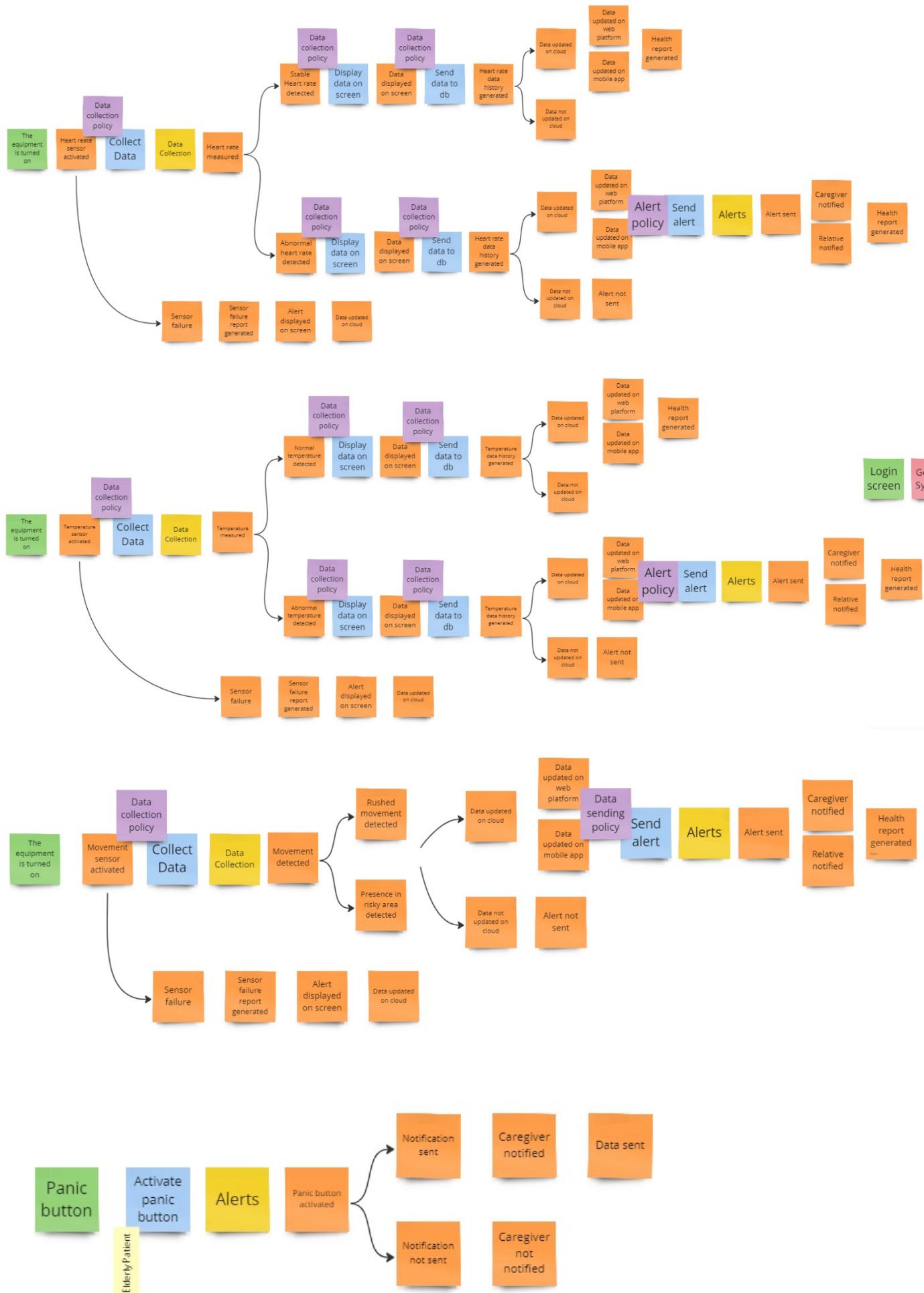


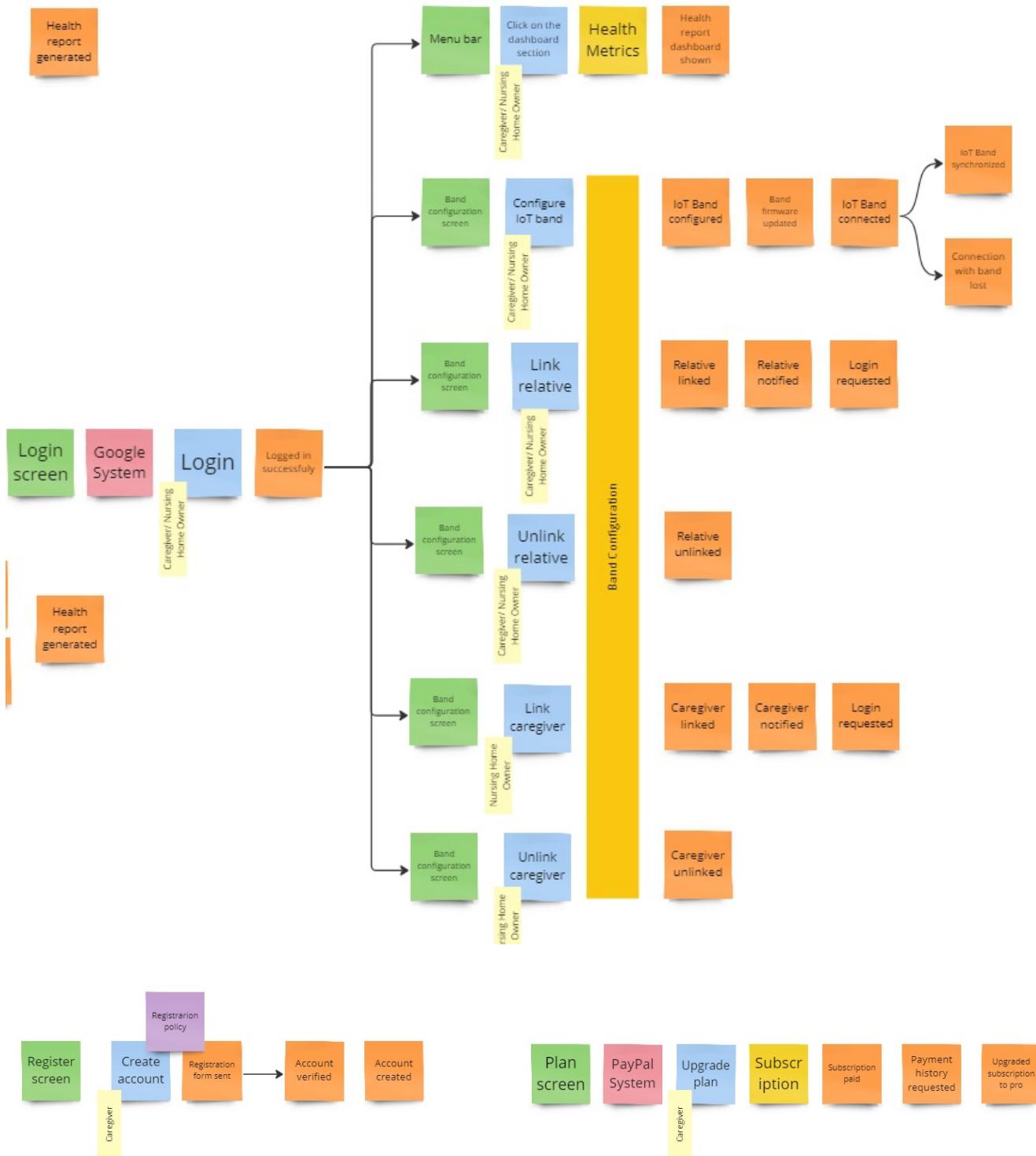


Incorporamos eventos relacionados con el encendido del equipo y el inicio de la recolección de datos, representados con etiquetas verdes. Además, se añadieron interacciones adicionales para diferentes sistemas de monitoreo, como sensores de movimiento, ritmo cardíaco y temperatura, destacando la política de colección de datos en color morado y la política de alertas en color lila. También se detallaron las acciones de inicio de sesión, enlace y desenlace de familiares y cuidadores, junto con la configuración del dispositivo IoT, resaltando la importancia del menú de navegación y los métodos de autenticación como el sistema de Google. En general, se completó la visualización de procesos desde la recolección de datos hasta la gestión de alertas y la actualización de información en la nube y plataformas web, estructurando el flujo de eventos para cada subsistema involucrado.

Step 9: Aggregates

Luego de presentar todos los eventos y comandos, comenzamos a agrupar los conceptos relacionados en agregados, que son las unidades que reciben comandos y generan eventos.

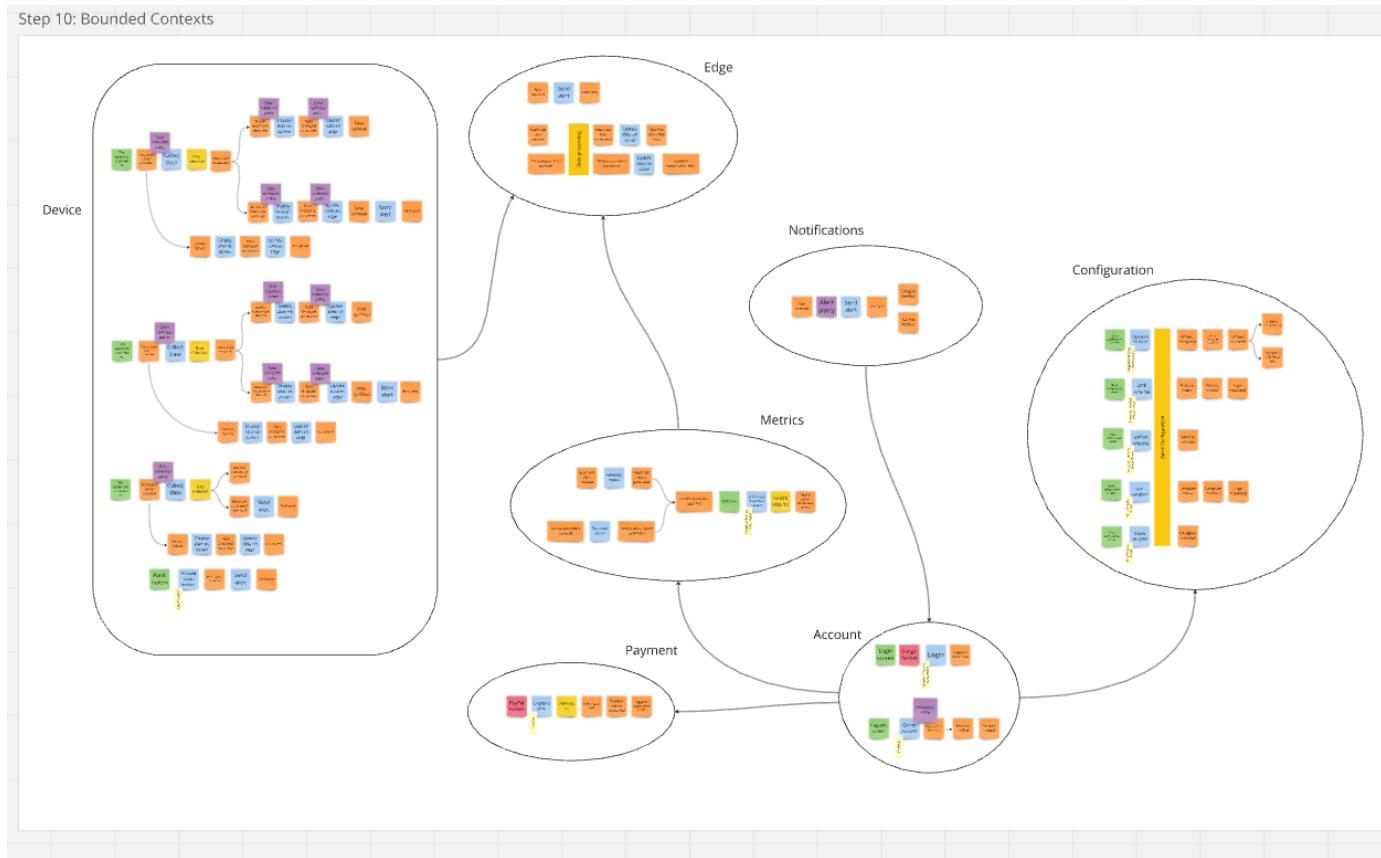




Cada diagrama inicia con la activación del equipo y la recolección de datos correspondientes (frecuencia cardíaca, temperatura, movimiento), procediendo a la detección de eventos normales o anormales. Si se detecta una anomalía, se sigue una política de recolección de datos y de alerta, que conlleva la generación de alertas y notificaciones hacia cuidadores y familiares a través de distintas plataformas (web, aplicación móvil). Además, se introdujo el flujo de inicio de sesión y creación de cuentas para configurar el sistema y asociar cuidadores o familiares con los pacientes, facilitando la personalización de alertas y la administración de las notificaciones.

Step 10: Bounded Contexts

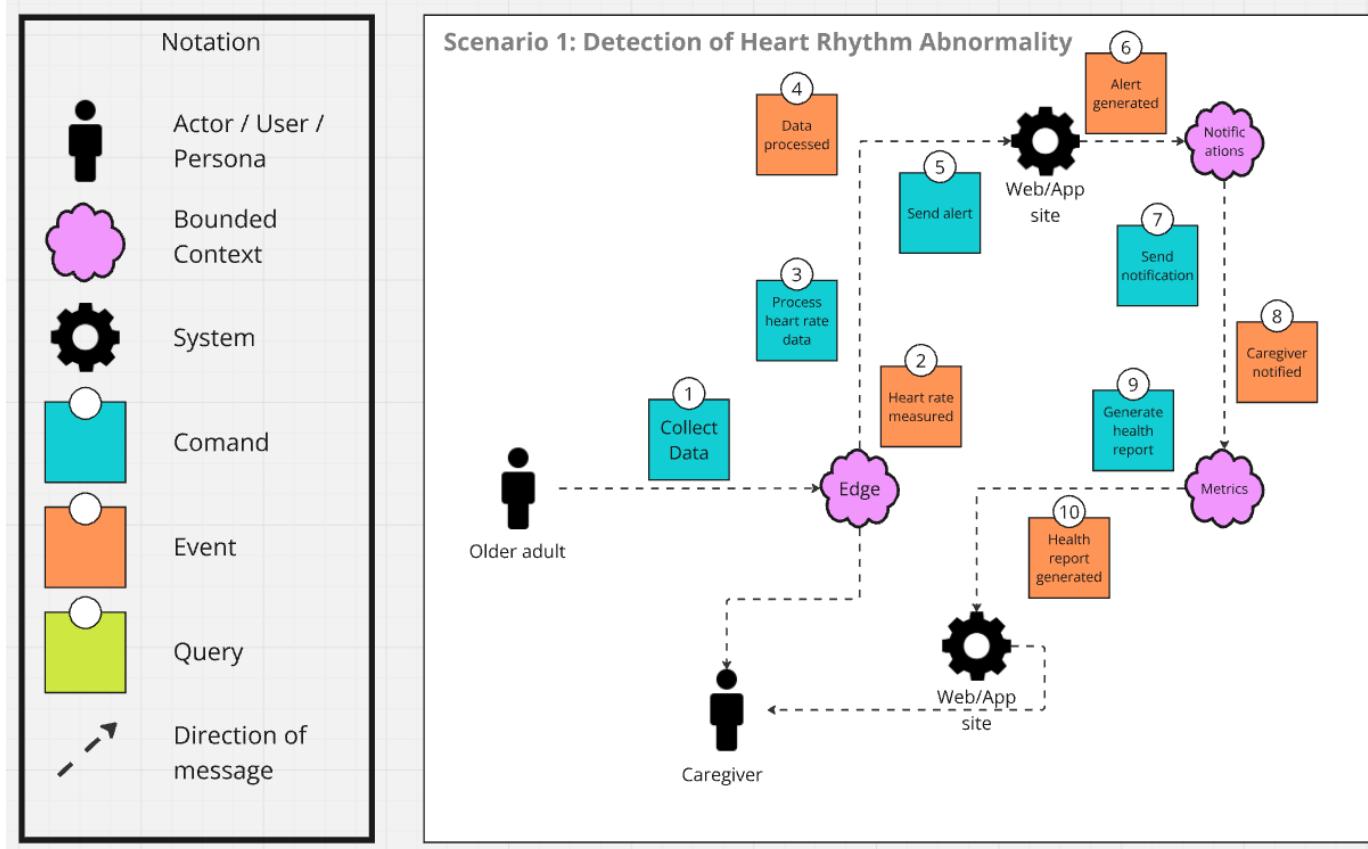
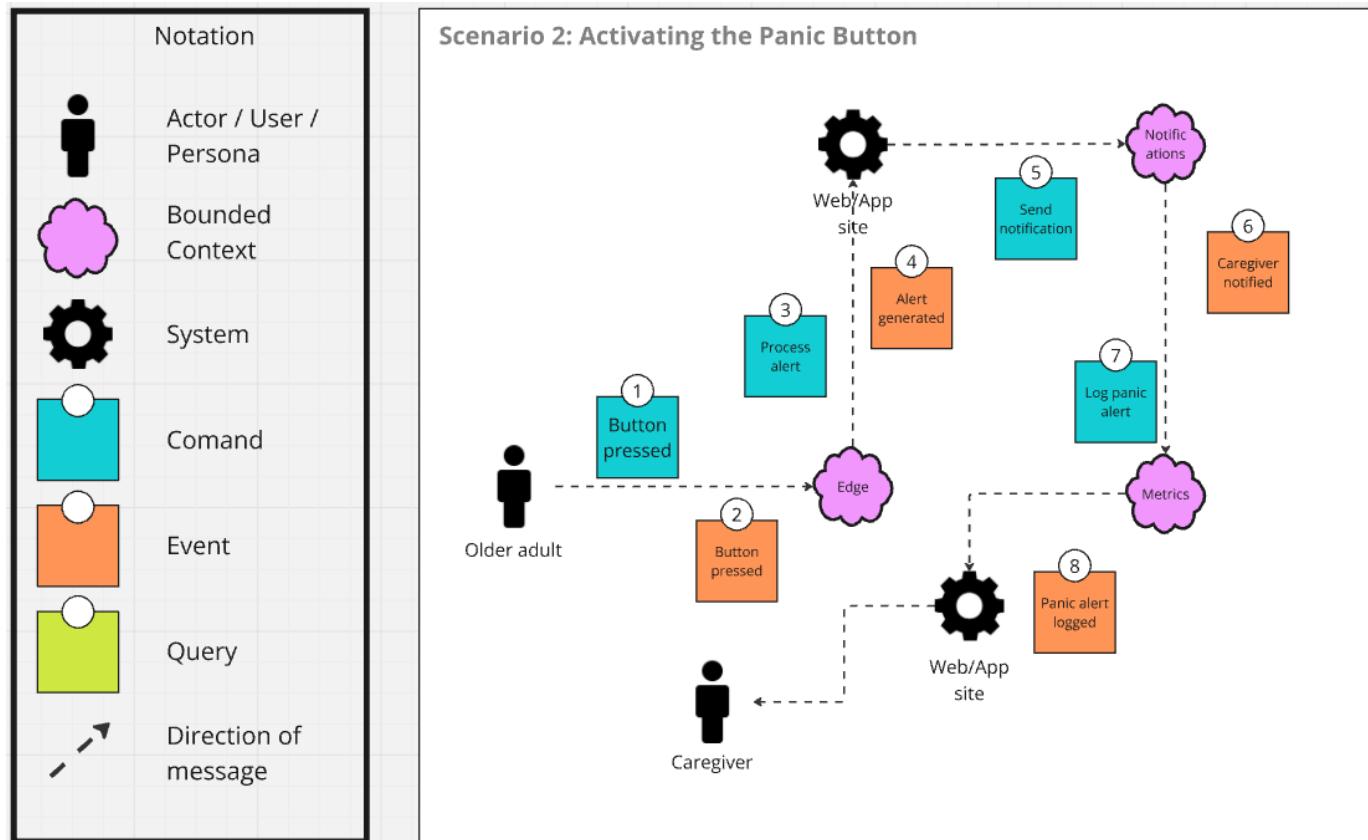
Finalmente, identificamos los agregados que están relacionados entre sí y que son relevantes por representar funciones estrechamente vinculadas o porque están conectados según ciertas políticas. Estos grupos de agregados sirven como candidatos naturales para definir los Bounded Contexts (contextos delimitados) dentro del sistema.

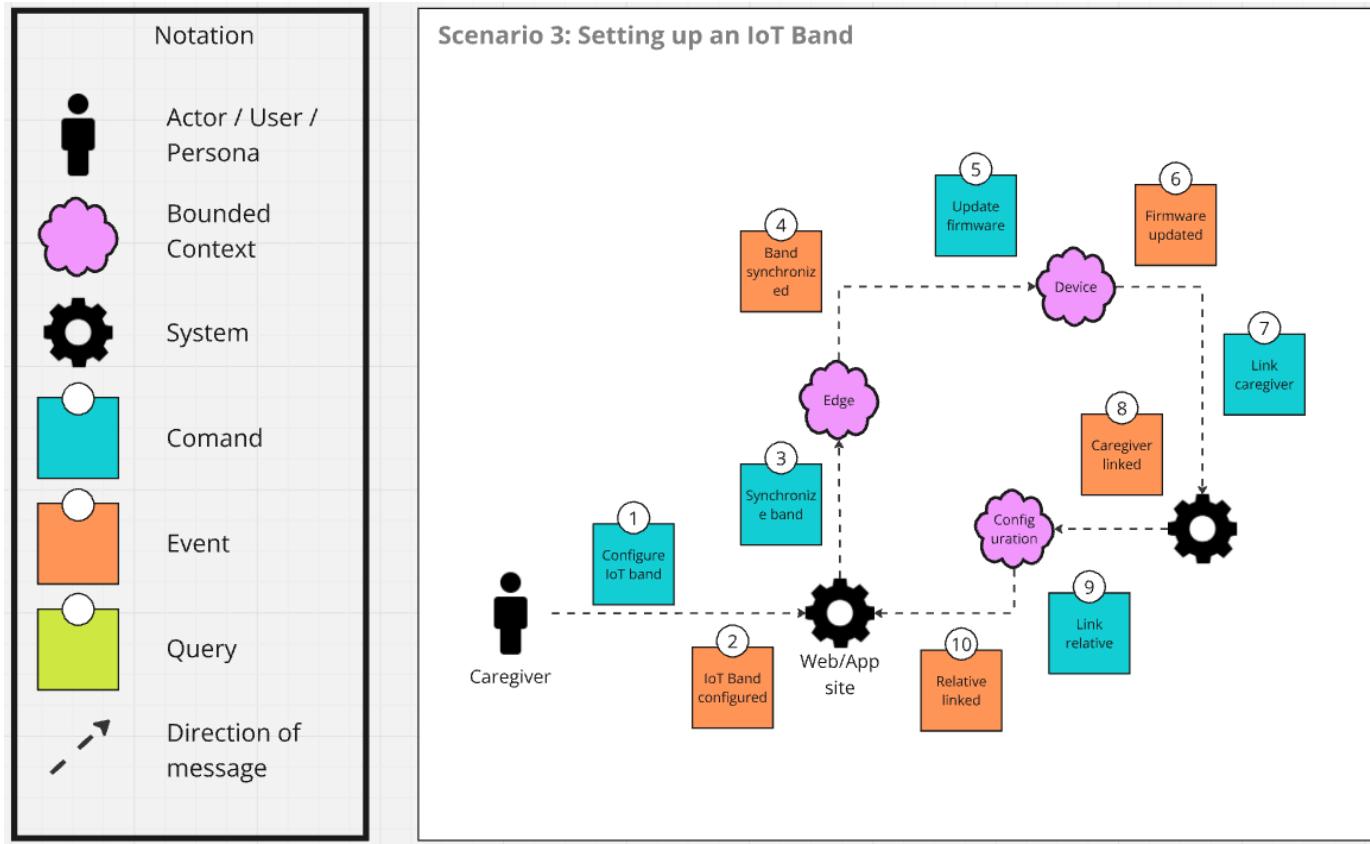
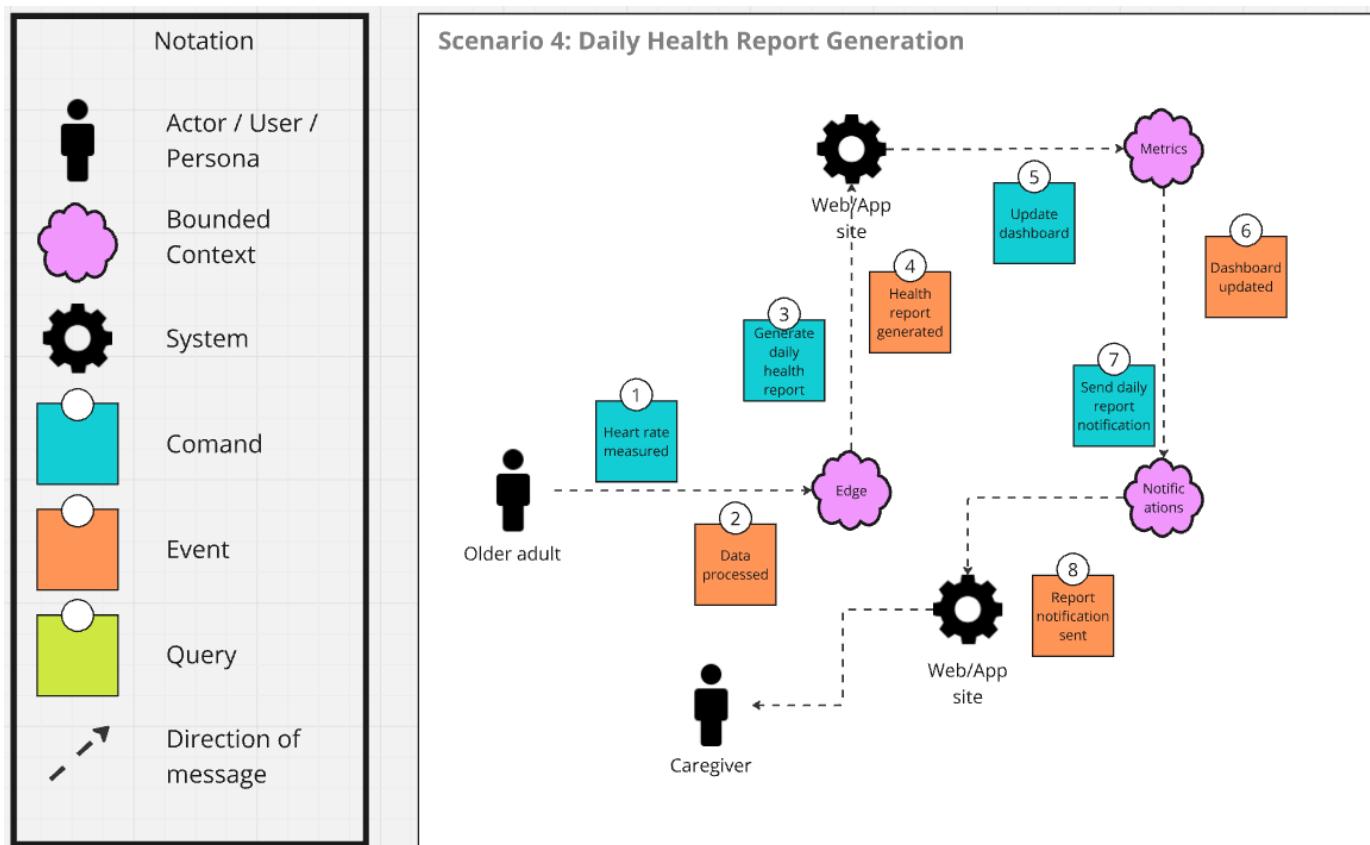


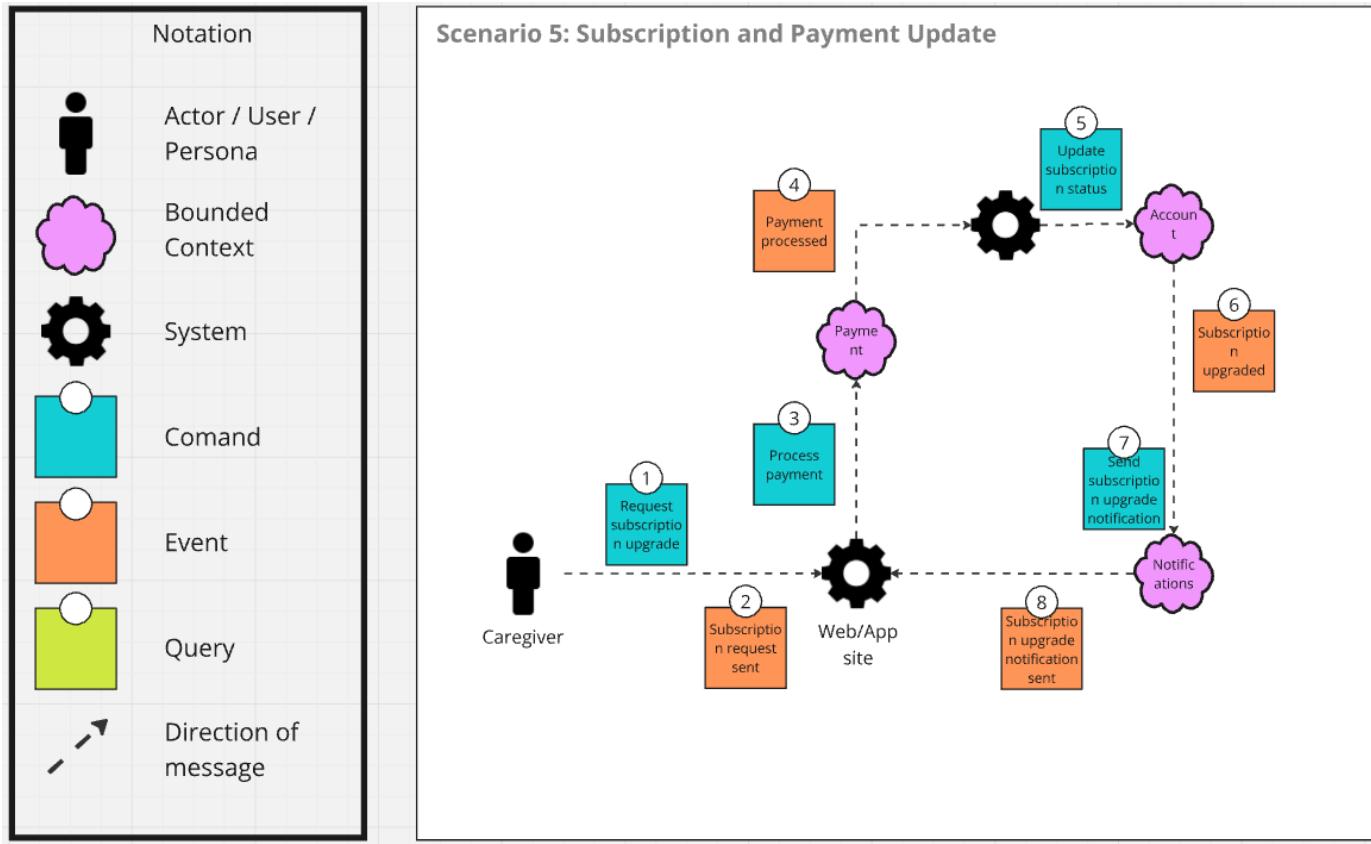
Identificamos y agrupamos los eventos del sistema en diferentes Bounded Contexts para definir áreas funcionales específicas: Device gestiona la interacción con dispositivos y sensores; Edge procesa datos en tiempo real y genera alertas; Notifications envía notificaciones a cuidadores y familiares; Metrics recolecta y gestiona métricas del sistema; Payment maneja suscripciones y pagos; Account se encarga de la autenticación y gestión de usuarios; y Planning (antes Configuration) permite la planificación y configuración del sistema. Estos contextos delimitan claramente las responsabilidades y facilitan la modularidad del sistema.

4.1.1.2 Domain Message Flows Modeling

Scenario: Detection of Heart Rhythm Abnormality

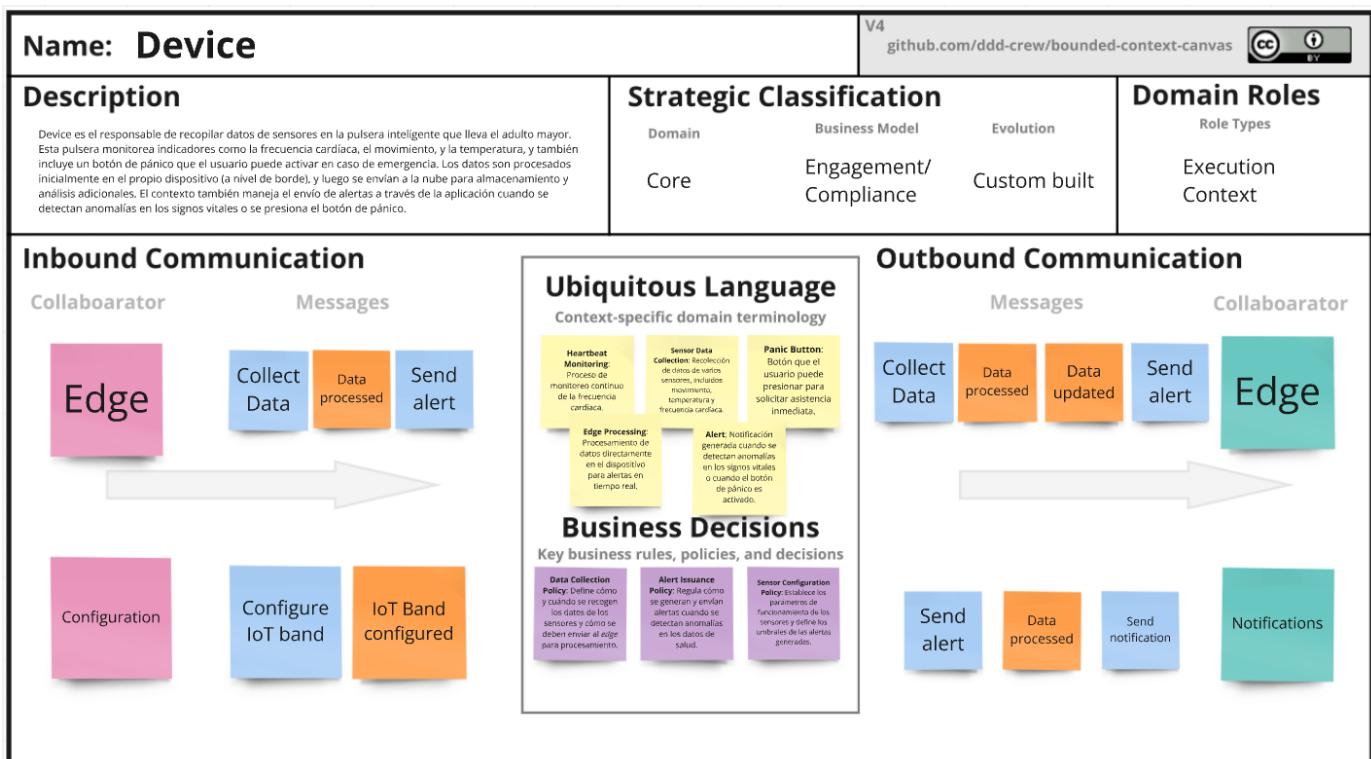
**Scenario: Activating the Panic Button****Scenario: Setting up an IoT Band**

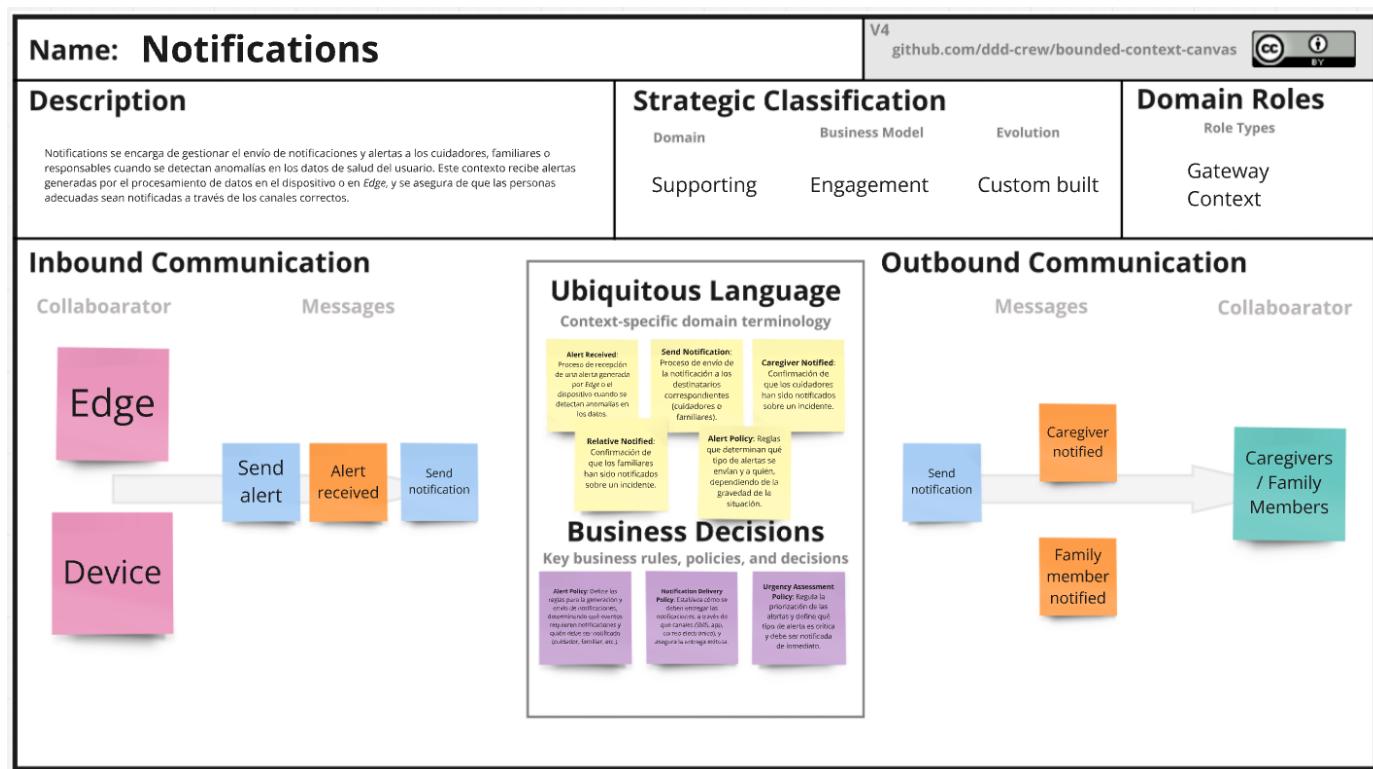
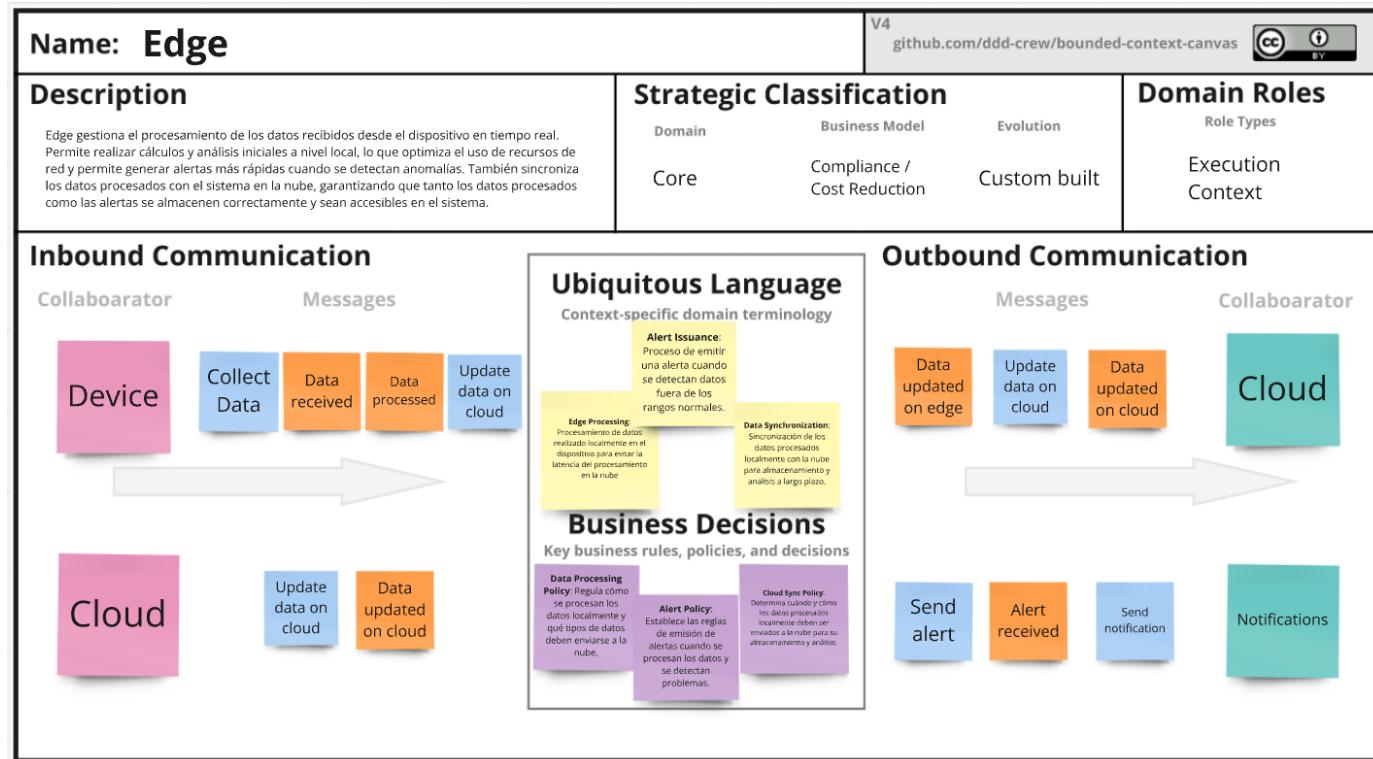
**Scenario: Daily Health Report Generation****Scenario: Subscription and Payment Update**

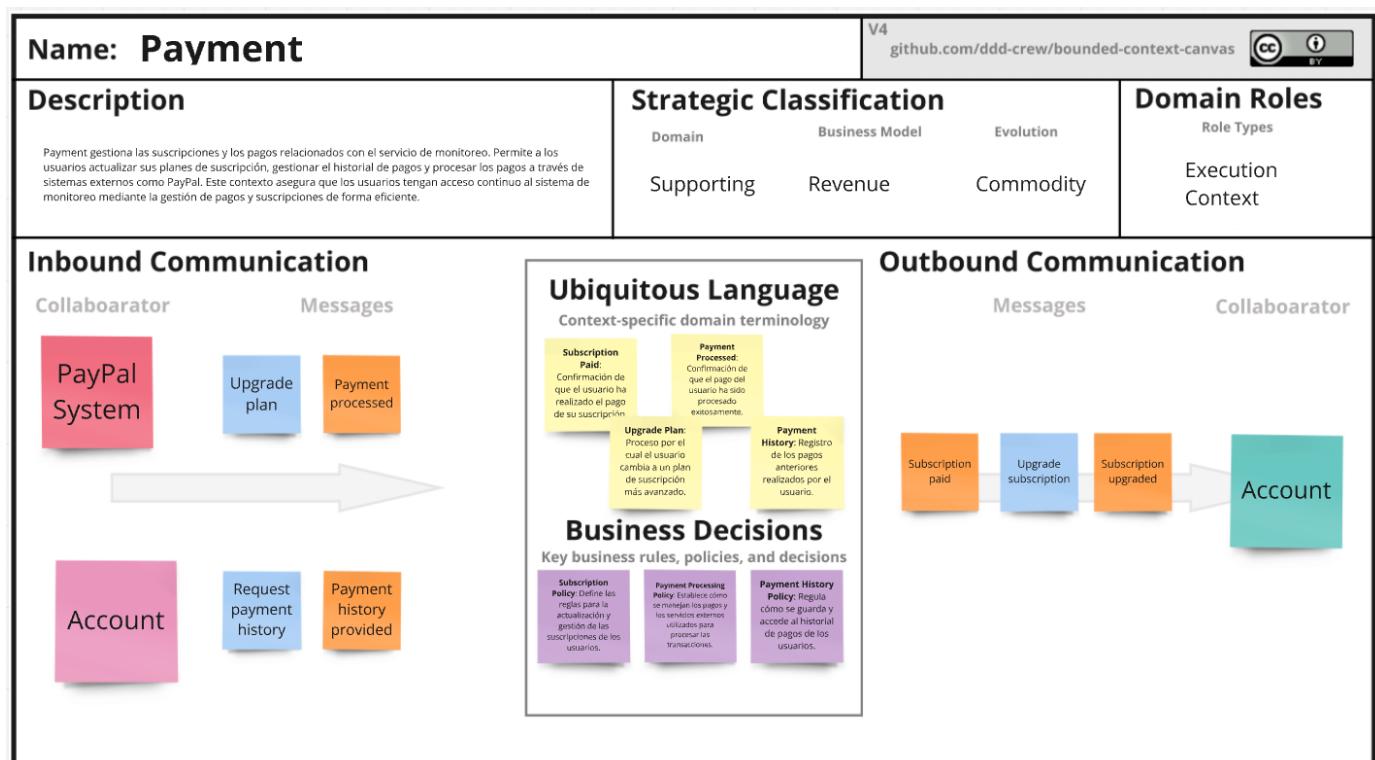
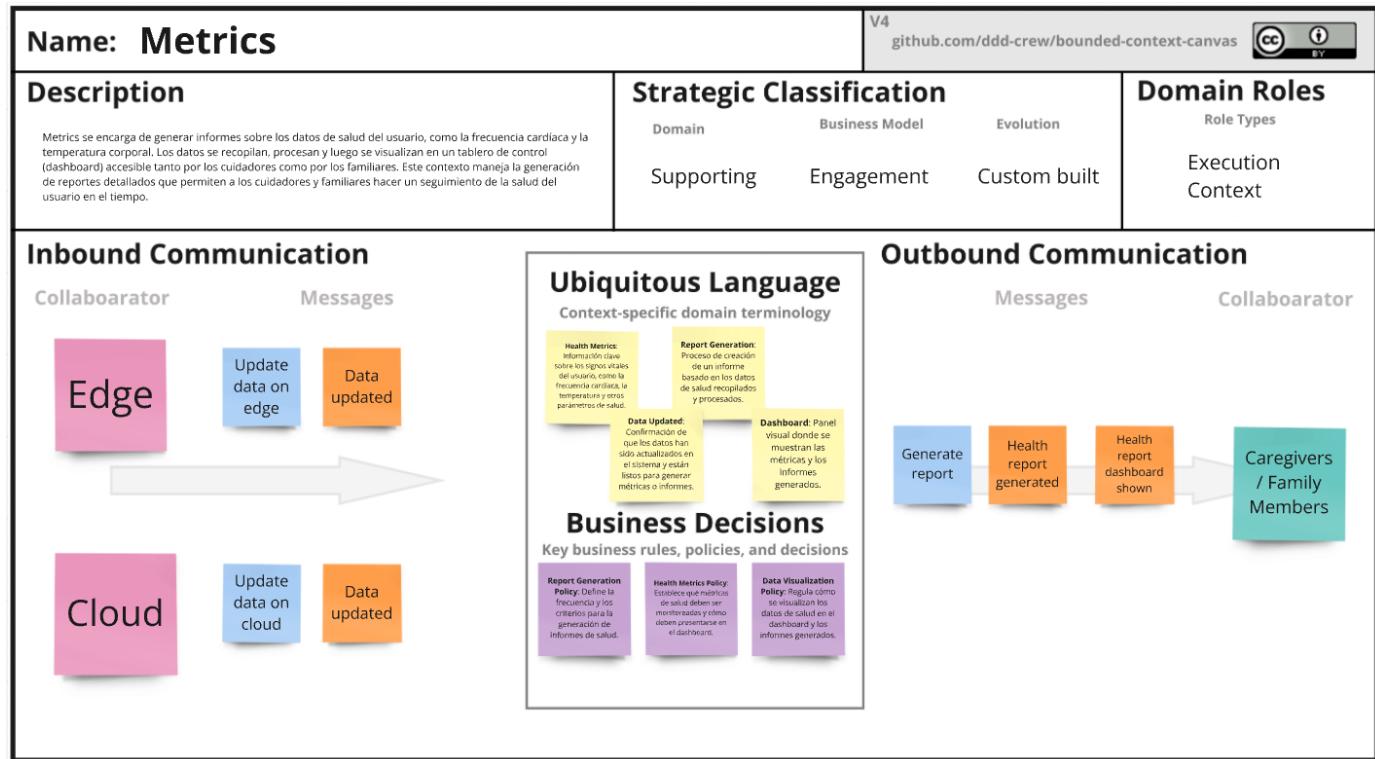


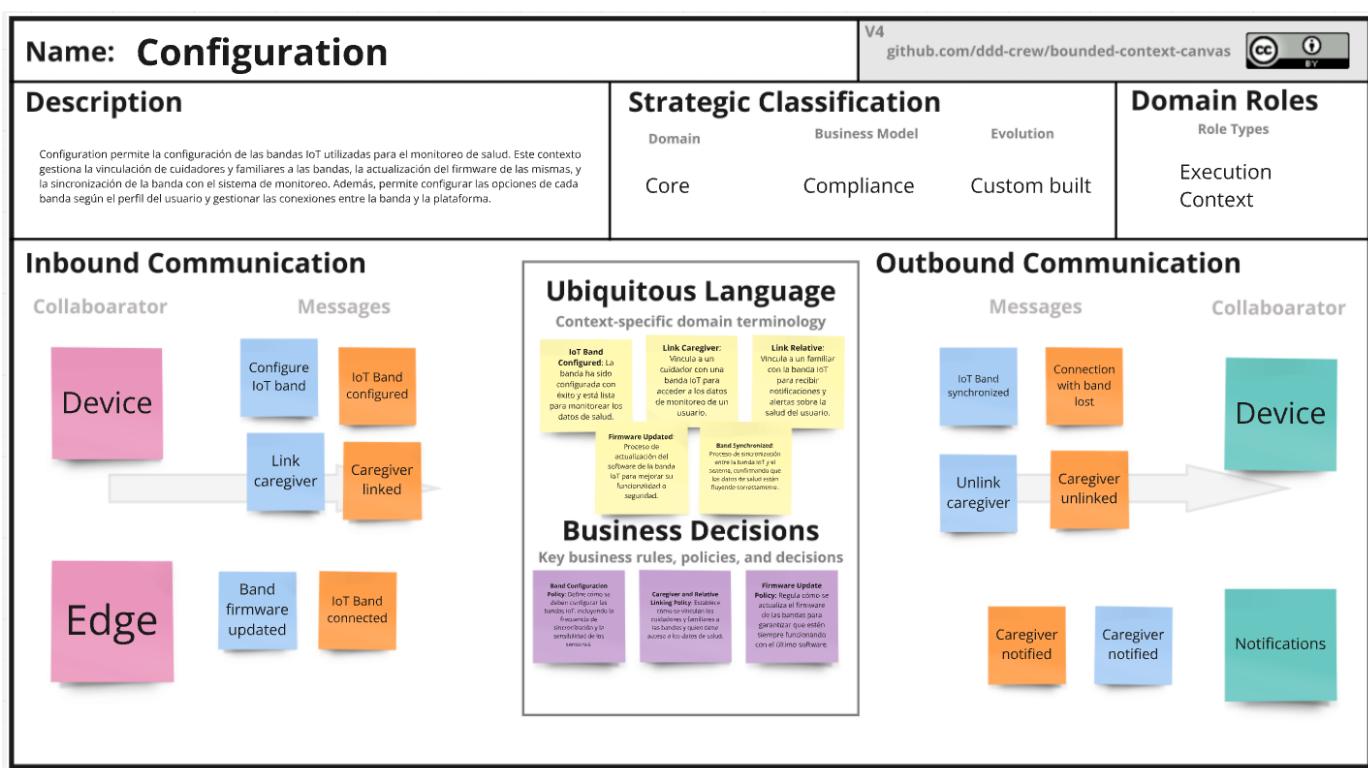
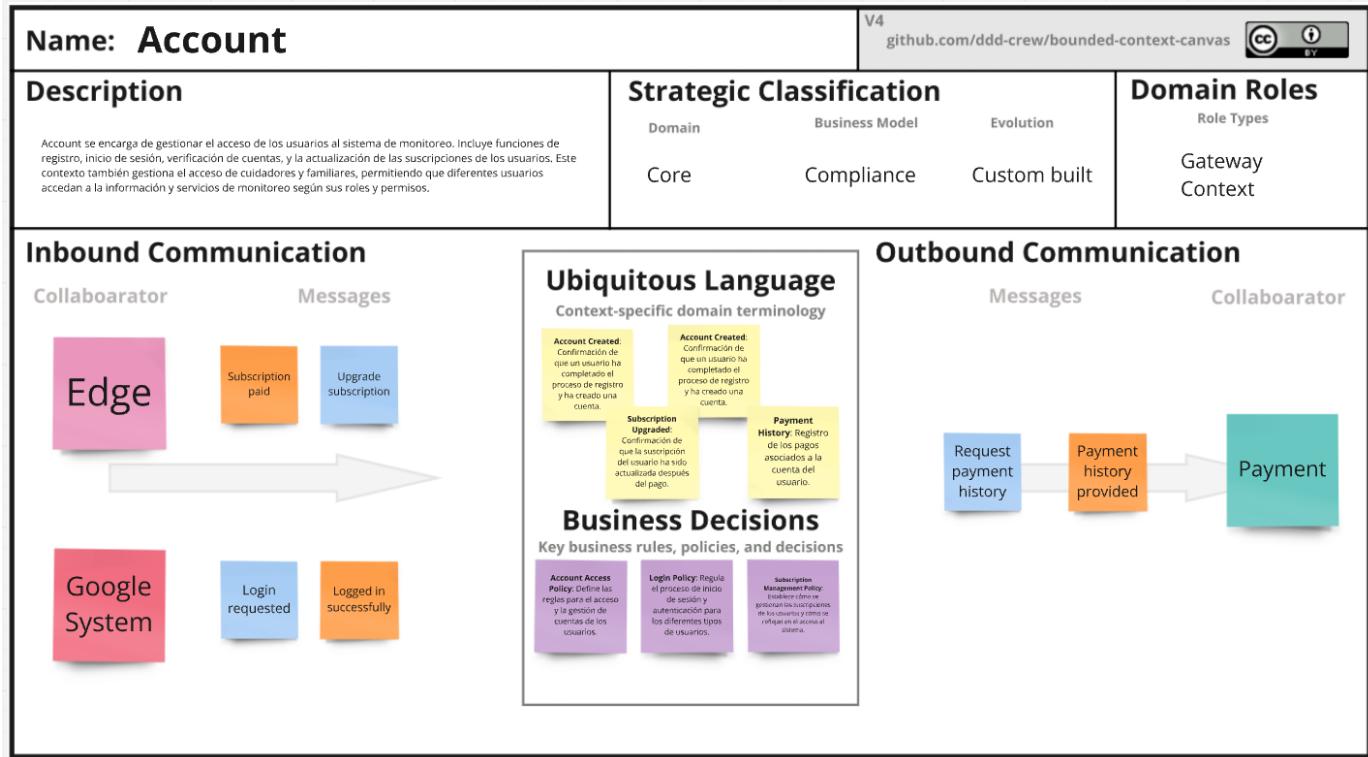
Enlace del Miroo para verlo completo: https://miro.com/app/board/uXjVKhkf7G8=/?share_link_id=819060613108

4.1.1.3 Bounded Context Canvases









Enlace del Miroo para verlo completo: https://miro.com/app/board/uXjVKhkFAVo=/?share_link_id=337128963652

4.1.2. Context Mapping

Proceso para Crear el Context Mapping y Análisis de Alternativas

1. Pasos para Crear el Context Mapping

1.1. Identificación de los Bounded Contexts

- Configuration
- Notification
- Account
- Metrics

- **Payment**
- **Edge**
- **Device**

1.2. Identificación de Relaciones Iniciales

- 1. Account → Configuration:** Relación de **Customer/Supplier**.
 - *Account* provee la configuración inicial de usuarios, mientras *Configuration* consume y ajusta las preferencias.
- 2. Notification → Device:** Relación de **Conformist**.
 - *Notification* se adapta a las características del *Device* para enviar alertas específicas.
- 3. Metrics → Device:** Relación de **Shared Kernel**.
 - Comparten datos comunes sobre el estado y rendimiento del dispositivo.
- 4. Payment → Account:** Relación de **Customer/Supplier**.
 - *Account* proporciona la información de facturación y el estado de la suscripción, mientras que *Payment* gestiona los pagos.
- 5. Edge → Device:** Relación de **Published Language**.
 - *Edge* se comunica con *Device* utilizando un lenguaje de mensajes estandarizado para gestionar la integración.

2. Análisis de Alternativas y Preguntas Clave

2.1. ¿Qué pasaría si movemos este capability a otro bounded context?

- **Caso Considerado:** Mover la capacidad de gestión de notificaciones desde *Notification* hacia *Account*.
- **Impacto:**
 - *Account* tendría la responsabilidad completa de administrar las preferencias y notificaciones de usuario.
 - Incrementaría el acoplamiento y la complejidad de *Account*.
- **Discusión:**
 - Se recomienda mantener la separación, ya que *Notification* debe ser independiente y responder a eventos de otros contextos.

2.2. ¿Qué pasaría si descomponemos este capability y movemos uno de los sub-capabilities a otro bounded context?

- **Caso Considerado:** Descomponer *Notification* en sub-capabilities como *EmailNotification* y *PushNotification* y mover *EmailNotification* a *Account*.
- **Impacto:**
 - Permitiría a *Account* gestionar las notificaciones relacionadas con el usuario de manera más específica.
 - *PushNotification* seguiría dependiendo de eventos generados por dispositivos.
- **Discusión:**
 - La descomposición permitiría un enfoque más específico en la entrega de notificaciones, pero incrementaría la complejidad de *Account*. Podría ser una buena opción dividirlo en dos bounded contexts independientes, pero no moverlo completamente a *Account*.

2.3. ¿Qué pasaría si partimos el bounded context en múltiples bounded contexts?

- **Caso Considerado:** Partir *Configuration* en *UserPreferences* y *SystemSettings*.
- **Impacto:**
 - Separar las configuraciones específicas del usuario de las configuraciones del sistema permitiría una gestión más eficiente y clara de las configuraciones, evitando la sobrecarga de un solo contexto.
- **Discusión:**
 - Se recomienda realizar esta separación para reducir la complejidad y mejorar la escalabilidad, evitando mezclar configuraciones que puedan tener diferentes ciclos de vida y dependencias.

2.4. ¿Qué pasaría si tomamos este capability de estos 3 contexts y lo usamos para formar un nuevo context?

- **Caso Considerado:** Crear un nuevo contexto llamado *SubscriptionManagement* que combine capacidades de *Account*, *Payment* y *Notification* relacionadas con la gestión de suscripciones y pagos.
- **Impacto:**
 - Centralizaría la lógica de gestión de suscripciones, pagos y notificaciones relacionadas.
 - Reduciría la duplicación y simplificaría la integración de capacidades.
- **Discusión:**
 - La creación de un nuevo contexto es viable y podría simplificar la arquitectura. Se debe evaluar si la complejidad de integración se reduce suficientemente para justificar la creación de este nuevo contexto.

2.5. ¿Qué pasaría si duplicamos una funcionalidad para romper la dependencia?

- **Caso Considerado:** Duplicar la funcionalidad de gestión de dispositivos en *Edge* y *Device* para reducir la dependencia mutua.
- **Impacto:**
 - Rompería la dependencia, pero aumentaría la duplicación de lógica y datos, lo que puede generar inconsistencias.
- **Discusión:**
 - No se recomienda la duplicación, ya que aumentaría la complejidad de mantenimiento. Mejor optar por la utilización de un servicio compartido o una capa anticorrupción.

2.6. ¿Qué pasaría si creamos un shared service para reducir la duplicación entre múltiples bounded contexts?

- **Caso Considerado:** Crear un servicio compartido de notificaciones para *Notification* y *Account*.
- **Impacto:**
 - Reduciría la duplicación en la lógica de envío y manejo de notificaciones, centralizando esta capacidad en un único servicio reutilizable.
- **Discusión:**
 - Se recomienda implementar un servicio compartido de notificaciones que permita reducir la duplicación de lógica y simplifique la integración con otros contextos.

2.7. ¿Qué pasaría si aislamos los core capabilities y movemos los otros a un context aparte?

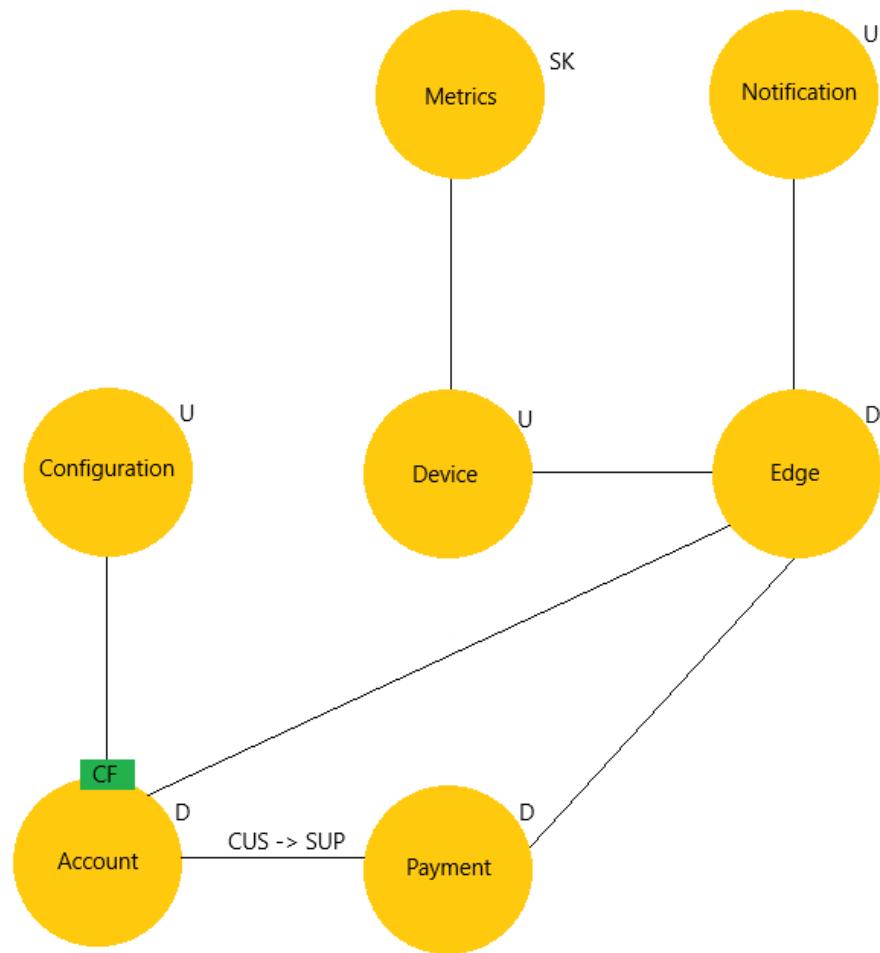
- **Caso Considerado:** Aislar las capacidades core de monitoreo de salud en *Device* y mover capacidades de gestión de dispositivos a un contexto aparte llamado *DeviceManagement*.
- **Impacto:**
 - Se reduciría la sobrecarga en el contexto de *Device*, permitiendo una mayor flexibilidad y escalabilidad en el manejo de dispositivos.
- **Discusión:**
 - Aislar capacidades core puede mejorar la escalabilidad y el mantenimiento del sistema. Se recomienda implementar esta alternativa para simplificar el contexto *Device*.

3. Alternativa Recomendada de Context Mapping

1. **Crear un contexto independiente de *SubscriptionManagement*** que maneje suscripciones, pagos y notificaciones relacionadas con suscripciones. Esto simplificaría las dependencias entre *Account*, *Payment* y *Notification*.
2. **Separar *Configuration* en *UserPreferences* y *SystemSettings*** para reducir la complejidad al separar configuraciones de usuario y de sistema.
3. **Implementar un servicio compartido de notificaciones** para centralizar la lógica de envío de notificaciones.
4. **Aislar las capacidades core de monitoreo de salud en *Device* y mover la gestión de dispositivos a *DeviceManagement*** para mejorar la escalabilidad y la gestión de dispositivos.

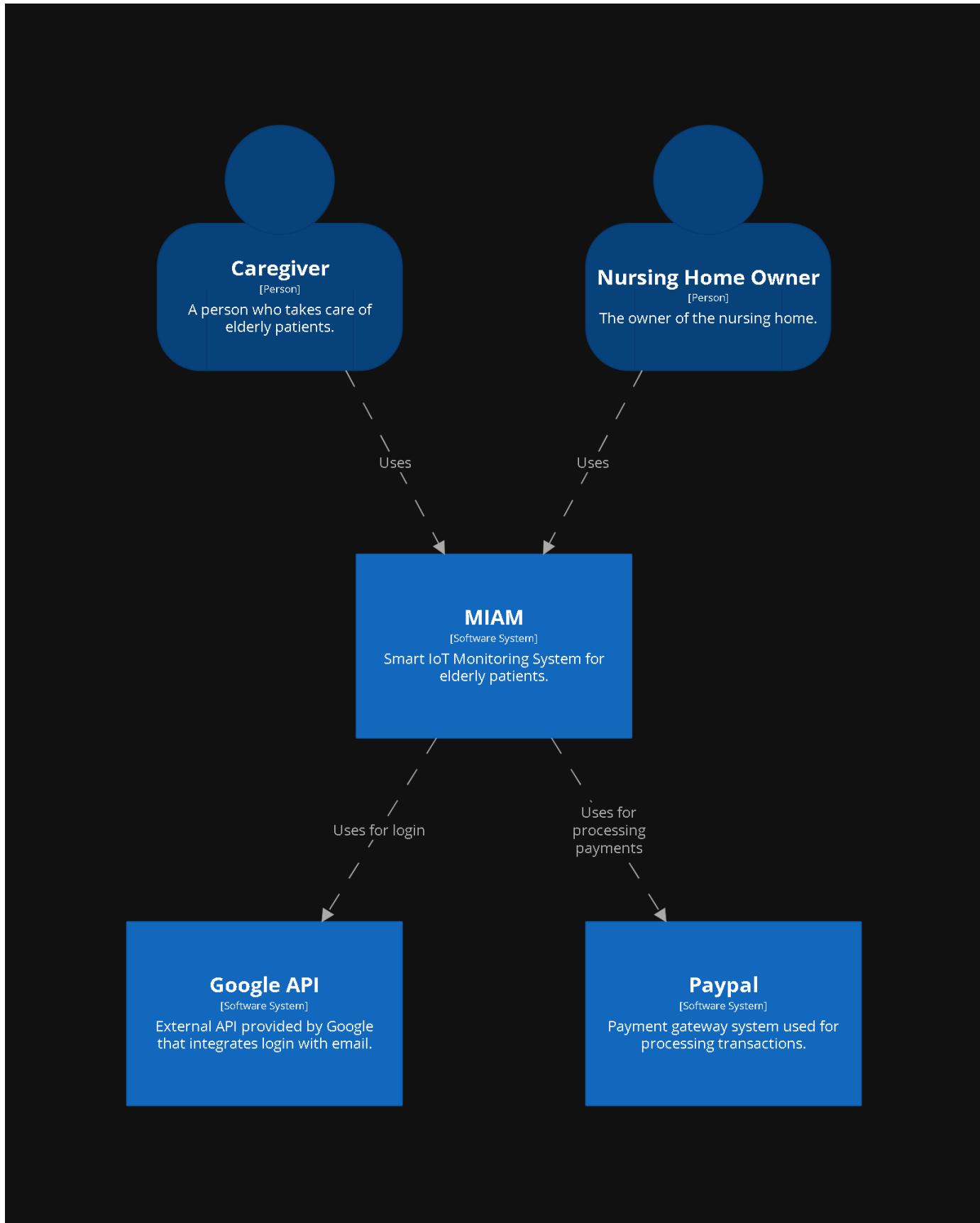
4. Patrones de Relaciones Sugeridos

- **Anti-corruption Layer (ACL):** Para proteger el contexto *Device* de cambios en *DeviceManagement*.
- **Conformist:** Para que *Notification* se adapte a los datos que recibe de *SubscriptionManagement*.
- **Shared Kernel:** Entre *Device* y *Metrics* para compartir información de monitoreo.
- **Customer/Supplier:** Relación entre *Account* y *SubscriptionManagement*.

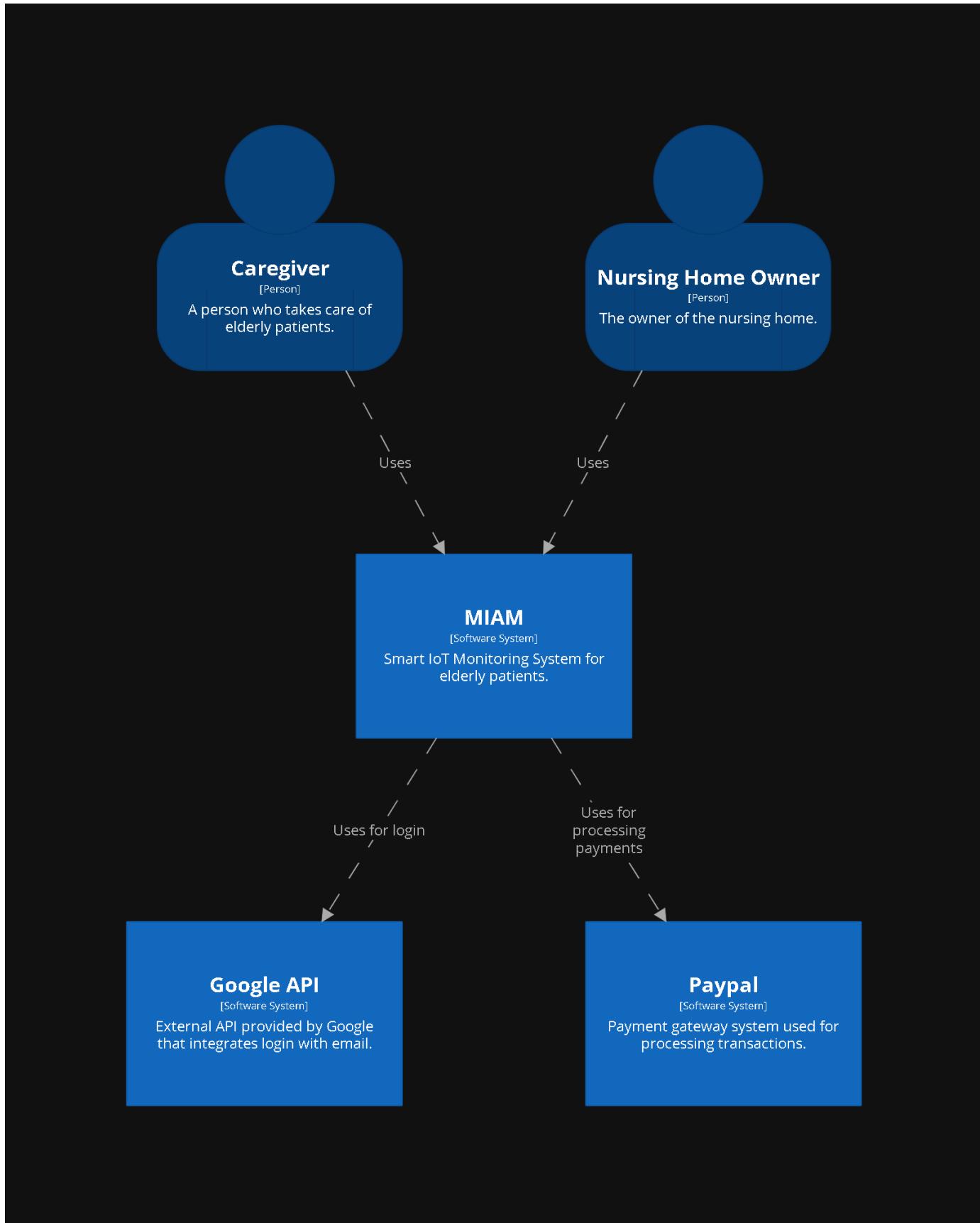


4.1.3. Software Architecture

4.1.3.1. Software Architecture System Landscape Diagram



4.1.3.2. Software Architecture Context Level Diagrams

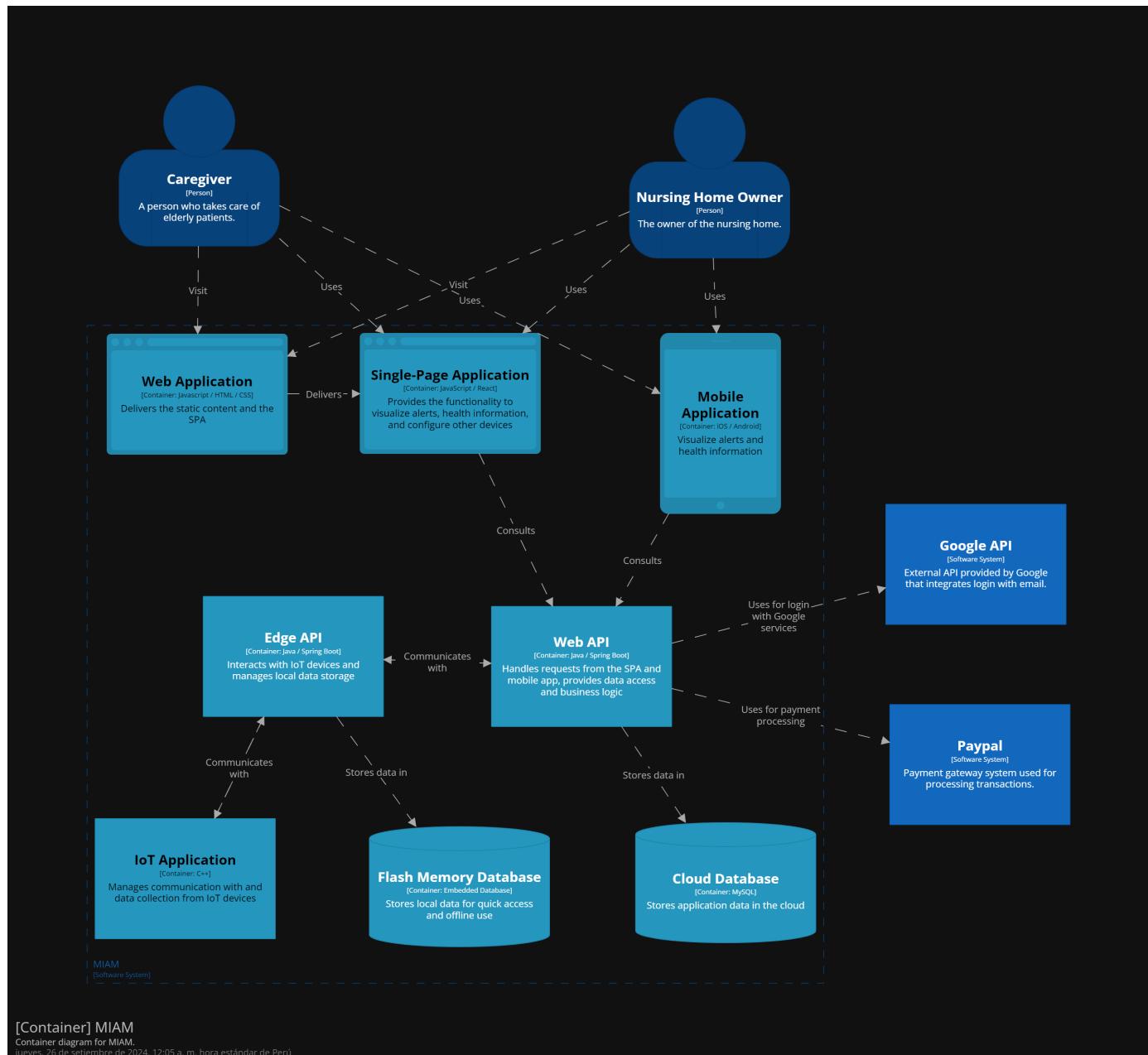


4.1.3.3. Software Architecture Container Level Diagrams

En este apartado se presenta un diagrama con una visión detallada de la arquitectura interna de MIAM, mostrando cómo se organiza el sistema en diferentes contenedores que manejan distintas responsabilidades. Los servicios externos de Google API y Paypal aseguran una experiencia de usuario fluida y segura para la creación de cuentas y la realización de pagos, integrándose perfectamente con la arquitectura del sistema.

Contenedores Principales del Sistema MIAM:

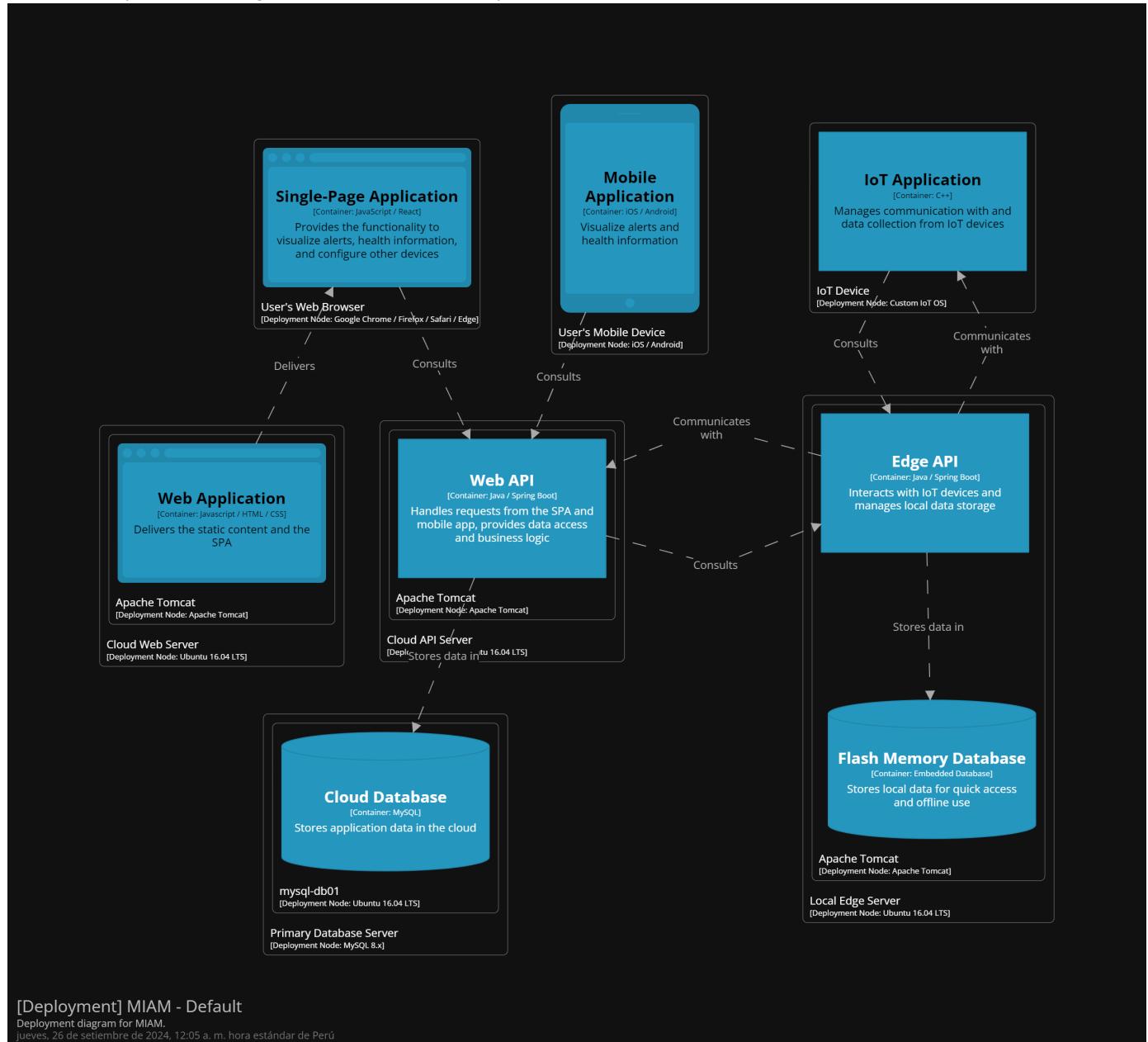
- Web Application: Proporciona el contenido estático y sirve como punto de entrada para la Single-Page Application (SPA). Permite a los usuarios ver información relevante y realizar acciones en la plataforma a través del navegador web.
- Single-Page Application (SPA): Permite a los usuarios visualizar alertas, información de salud y configurar dispositivos. Es la interfaz interactiva principal que se ejecuta en el navegador del usuario.
- Mobile Application: Ofrece una versión adaptada de la funcionalidad de la SPA en dispositivos móviles, permitiendo a los usuarios visualizar alertas e información de salud de manera cómoda en sus teléfonos móviles.
- Web API: Maneja las solicitudes de las aplicaciones web y móviles. Proporciona acceso a la lógica de negocio y los datos, actuando como intermediario entre el frontend y las bases de datos en la nube.
- Edge API: Interactúa directamente con los dispositivos IoT y gestiona el almacenamiento de datos locales. Este contenedor está diseñado para operar en el entorno local del centro de cuidados, reduciendo la latencia y permitiendo respuestas rápidas.
- IoT Application: Gestiona la comunicación y la recopilación de datos desde dispositivos IoT, como pulseras inteligentes utilizadas por los pacientes.
- Flash Memory Database: Almacena datos localmente en el entorno de Edge API para acceso rápido y sin conexión, asegurando la continuidad del servicio incluso en caso de problemas de conectividad.
- Cloud Database: Almacena los datos de la aplicación en la nube. Actúa como el repositorio principal para toda la información relevante de la aplicación.



4.1.3..4. Software Architecture Deployment Diagrams

La Web Application se despliega en el Cloud Web Server y entrega el contenido estático al User's Web Browser, que ejecuta la Single-Page Application. La Mobile Application y la Single-Page Application consultan la Web API para obtener datos y ejecutar lógica de negocio. La Web API se comunica con la Cloud Database para almacenar y recuperar datos. La Edge API, desplegada localmente, se comunica con los dispositivos IoT y

la Flash Memory Database para gestionar datos en tiempo real y almacenamiento local.



4.2. Tactical-Level Domain-Driven Design

4.2.1. Bounded Context: Device Context

4.2.1.1. Domain Layer

Entities

1. Device:

- **Descripción:** Representa un dispositivo IoT en el sistema, el cual recopila datos de salud y permite la comunicación con la infraestructura de borde.
- **Atributos:**
 - **id:** Identificador único del dispositivo.
 - **model:** Modelo del dispositivo.
 - **status:** Estado actual del dispositivo (activo, inactivo).
 - **heartRate:** Frecuencia cardíaca registrada por el dispositivo.
 - **temperature:** Temperatura registrada por el dispositivo.
 - **location:** Ubicación actual del dispositivo.

2. IoTBand:

- **Descripción:** Representa la banda IoT que se usa en el dispositivo para recopilar y transmitir datos de salud.

- **Atributos:**

- `id`: Identificador único de la banda.
- `firmwareVersion`: Versión del firmware de la banda.
- `status`: Estado actual de la banda (operativa, no operativa).

3. Configuration:

- **Descripción:** Almacena las configuraciones específicas de un dispositivo IoT, como umbrales de alertas y parámetros de monitoreo.

- **Atributos:**

- `id`: Identificador único de la configuración.
- `configName`: Nombre de la configuración.
- `value`: Valor de la configuración.
- `Device_id`: Referencia al dispositivo al que pertenece la configuración.

Value Objects

1. DeviceStatus:

- **Descripción:** Objeto de valor que representa el estado de un dispositivo. Asegura que el estado cumpla con ciertas reglas de negocio (por ejemplo, solo valores como 'activo' o 'inactivo').
- **Atributos:**
 - `status`: Estado del dispositivo (activo, inactivo, mantenimiento).

Domain Services

1. DeviceDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con los dispositivos, como el cambio de estado y la verificación de configuraciones.
- **Métodos:**
 - `updateDeviceStatus(Device device, String status)`: Actualiza el estado del dispositivo después de validar que es un estado permitido.

Aggregates y Aggregate Root

1. DeviceAggregateRoot:

- **Justificación:** `Device` es el aggregate root que controla la consistencia y la lógica de negocio de los dispositivos y sus configuraciones. Asegura que las configuraciones aplicadas a un dispositivo sean válidas y consistentes.

Repositories (Interfaces)

1. DeviceRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de dispositivos.
- **Métodos:**
 - `findById(String id)`: Busca un dispositivo por su ID.
 - `save(Device device)`: Guarda o actualiza un dispositivo en la base de datos.
 - `delete(String id)`: Elimina un dispositivo de la base de datos.

2. IoTBandRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de bandas IoT.
- **Métodos:**
 - `findById(String id)`: Busca una banda por su ID.
 - `save(IoTBand band)`: Guarda o actualiza una banda en la base de datos.
 - `delete(String id)`: Elimina una banda de la base de datos.

3. ConfigurationRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de configuraciones.
- **Métodos:**
 - `findById(String id)`: Busca una configuración por su ID.
 - `save(Configuration config)`: Guarda o actualiza una configuración en la base de datos.
 - `delete(String id)`: Elimina una configuración de la base de datos.

4.2.1.2. Interface Layer

Controllers

1. DeviceController:

- **Descripción:** Expone endpoints para la gestión de dispositivos, incluyendo la creación, actualización y recuperación de dispositivos.
- **Métodos:**
 - `registerDevice(Device device)`: Endpoint para registrar un nuevo dispositivo.
 - `updateDeviceStatus(String id, String status)`: Endpoint para actualizar el estado de un dispositivo.
 - `getDeviceById(String id)`: Endpoint para obtener los detalles de un dispositivo por su ID.

2. IoTBandController:

- **Descripción:** Controlador que gestiona las bandas IoT, permitiendo su registro, actualización y consulta.
- **Métodos:**
 - `registerIoTBand(IoTBand band)`: Endpoint para registrar una nueva banda IoT.
 - `updateIoTBand(String id, IoTBand band)`: Endpoint para actualizar una banda IoT.
 - `getIoTBandById(String id)`: Endpoint para obtener los detalles de una banda por su ID.

3. ConfigurationController:

- **Descripción:** Controlador que gestiona las configuraciones de dispositivos, permitiendo su creación, actualización y consulta.
- **Métodos:**
 - `createConfiguration(Configuration configuration)`: Endpoint para crear una nueva configuración.
 - `updateConfiguration(Configuration configuration)`: Endpoint para actualizar una configuración.
 - `getConfigurationById(String id)`: Endpoint para obtener los detalles de una configuración por su ID.

4.2.1.3. Application Layer

Command Handlers

1. UpdateDeviceStatusHandler:

- **Descripción:** Maneja el comando que actualiza el estado de un dispositivo específico.
- **Métodos:**
 - `handle(UpdateDeviceStatusCommand command)`: Valida y aplica el comando para actualizar el estado de un dispositivo.

2. RegisterIoTBandHandler:

- **Descripción:** Maneja el comando para registrar una nueva banda IoT en el sistema.
- **Métodos:**
 - `handle(RegisterIoTBandCommand command)`: Valida y registra una nueva banda IoT en el sistema.

3. UpdateConfigurationHandler:

- **Descripción:** Maneja el comando para actualizar la configuración de un dispositivo específico.
- **Métodos:**
 - `handle(UpdateConfigurationCommand command)`: Valida y aplica el comando para actualizar la configuración de un dispositivo.

Event Handlers

1. DeviceUpdatedEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando un dispositivo es actualizado, asegurando que cualquier cambio sea procesado adecuadamente.
- **Métodos:**
 - `handle(DeviceUpdatedEvent event)`: Procesa las acciones necesarias después de que un dispositivo ha sido actualizado.

2. IoTBandRegisteredEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando se registra una nueva banda IoT, realizando tareas de inicialización.
- **Métodos:**
 - `handle(IoTBandRegisteredEvent event)`: Ejecuta las acciones necesarias después del registro de una nueva banda IoT.

Query Handlers

1. GetDeviceDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un dispositivo específico.
- **Métodos:**
 - `handle(GetDeviceDetailsQuery query)`: Devuelve los detalles del dispositivo solicitado.

2. GetIoTBandDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de una banda IoT específica.
- **Métodos:**
 - `handle(GetIoTBandDetailsQuery query)`: Devuelve los detalles de la banda solicitada.

3. GetConfigurationDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de una configuración específica.
- **Métodos:**
 - `handle(GetConfigurationDetailsQuery query)`: Devuelve los detalles de la configuración solicitada.

4.2.1.4. Infrastructure Layer

Repositories (Implementaciones)

1. DeviceRepositoryImpl:

- **Descripción:** Implementación de `DeviceRepository` que define cómo se almacenan y recuperan los dispositivos desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un dispositivo por su ID.
 - `save(Device device)`: Implementación para guardar o actualizar un dispositivo en la base de datos.
 - `delete(String id)`: Implementación para eliminar un dispositivo de la base de datos.

2. IoTBandRepositoryImpl:

- **Descripción:** Implementación de `IoTBandRepository` que define cómo se almacenan y recuperan las bandas IoT desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una banda por su ID.
 - `save(IoTBand band)`: Implementación para guardar o actualizar una banda en la base de datos.
 - `delete(String id)`: Implementación para eliminar una banda de la base de datos.

3. ConfigurationRepositoryImpl:

- **Descripción:** Implementación de `ConfigurationRepository` que define cómo se almacenan y recuperan las configuraciones desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una configuración por su ID.
 - `save(Configuration config)`: Implementación para guardar o actualizar una configuración en la base de datos.
 - `delete(String id)`: Implementación para eliminar una configuración de la base de datos.

4.2.1.6. Bounded Context Software Architecture Component Level Diagrams

Components (Edge API):

1. DeviceService Component:

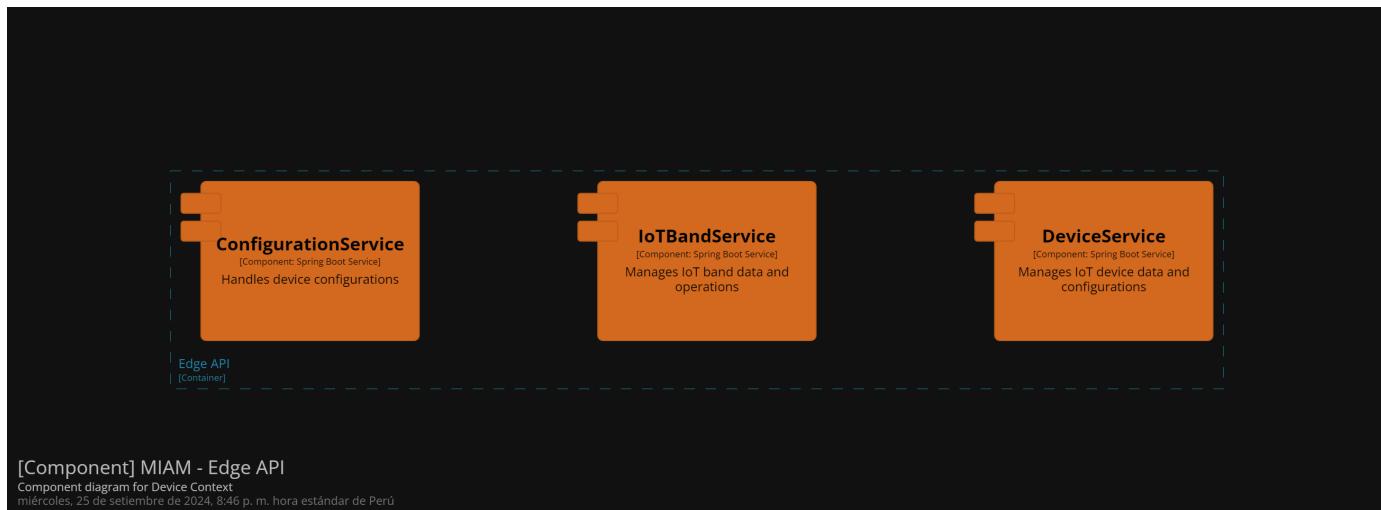
- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de dispositivos. Interactúa con el `DeviceRepository` para registrar y actualizar dispositivos.

2. IoTBandService Component:

- **Descripción:** Componente encargado de la gestión de bandas IoT. Interactúa con el `IoTBandRepository` para gestionar el registro y la actualización de bandas.

3. ConfigurationService Component:

- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de configuraciones. Interactúa con el `ConfigurationRepository` para registrar y actualizar configuraciones.



4.2.1.7. Bounded Context Software Architecture Code Level Diagrams

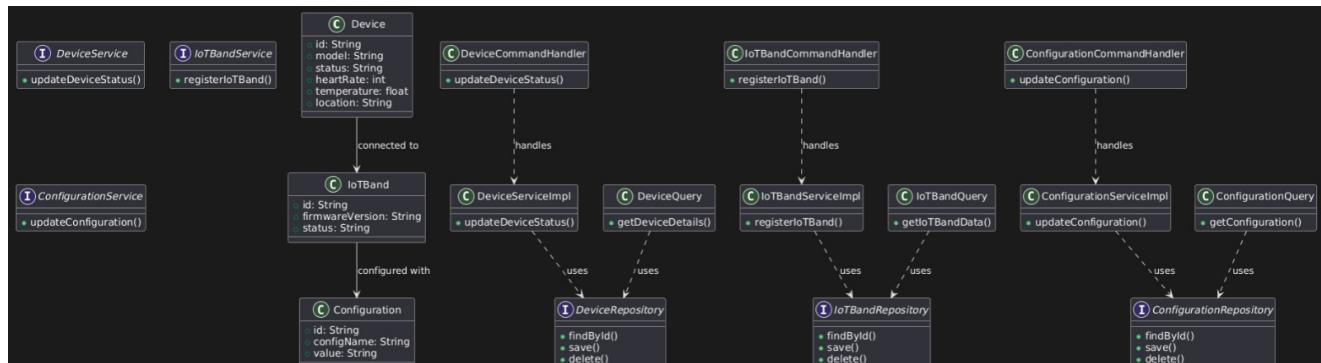
4.2.1.7.1. Bounded Context Domain Layer Class Diagrams

- **Clases importantes:**

- **Device**
- **IoTBand**
- **Configuration**

- **Repositories:**

- **DeviceRepository**
- **IoTBandRepository**
- **ConfigurationRepository**



4.2.1.7.2. Bounded Context Database Design Diagram

- **Tablas:**

- **Device:**

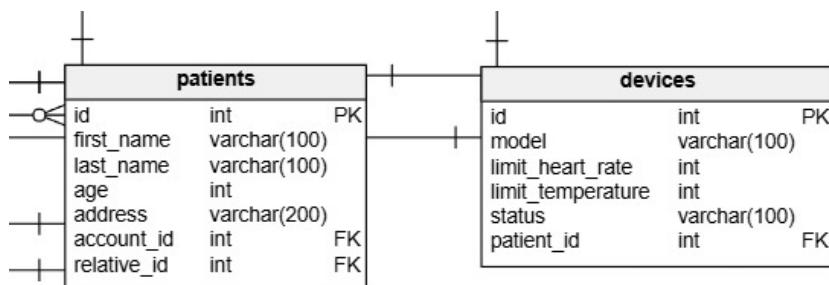
- Columnas: **id**, **model**, **status**, **Patient_id**.
 - Llave Foránea: **Patient_id**.

- **IoTBand:**

- Columnas: **id**, **firmwareVersion**, **status**.

- **Configuration:**

- Columnas: **id**, **configName**, **value**, **Device_id**.
 - Llave Foránea: **Device_id**.



- **Relaciones:**
 - **Configuration** se relaciona con **Device** a través de **Device_id**.
 - **Device** puede tener varias configuraciones asociadas.

4.2.2. Bounded Context: Edge Context

4.2.2.1. Domain Layer

Entities

1. Metrics:

- **Descripción:** Representa los datos de salud recopilados por un dispositivo, incluyendo frecuencia cardíaca y temperatura.
- **Atributos:**
 - **id:** Identificador único de las métricas.
 - **averageHeartRate:** Frecuencia cardíaca promedio registrada.
 - **averageTemperature:** Temperatura promedio registrada.
 - **alertsGenerated:** Número de alertas generadas basadas en las métricas.

2. Patient:

- **Descripción:** Representa al paciente al que se le realiza el monitoreo de salud a través del dispositivo IoT.
- **Atributos:**
 - **id:** Identificador único del paciente.
 - **name:** Nombre del paciente.
 - **age:** Edad del paciente.
 - **address:** Dirección del paciente.
 - **emergencyContact:** Contacto de emergencia para el paciente.

3. Device:

- **Descripción:** Representa un dispositivo IoT en el sistema, utilizado para recopilar datos de salud del paciente.
- **Atributos:**
 - **id:** Identificador único del dispositivo.
 - **model:** Modelo del dispositivo.
 - **status:** Estado actual del dispositivo (activo, inactivo, en mantenimiento).
 - **Patient_id:** Referencia al paciente al que está asignado el dispositivo.

Value Objects

1. HealthMetric:

- **Descripción:** Objeto de valor que representa una métrica de salud específica, como la frecuencia cardíaca o la temperatura. Asegura que las métricas de salud cumplan con los requisitos establecidos.
- **Atributos:**
 - **metricName:** Nombre de la métrica de salud (por ejemplo, frecuencia cardíaca, temperatura).
 - **value:** Valor de la métrica.

Domain Services

1. MetricsDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con las métricas de salud, como el cálculo de promedios y la generación de alertas.

- **Métodos:**

- `calculateAverage(Metric metric)`: Calcula el promedio de una métrica de salud.
- `generateAlertIfNecessary(Metric metric)`: Genera una alerta si la métrica supera ciertos umbrales.

Aggregates y Aggregate Root

1. PatientAggregateRoot:

- **Justificación:** `Patient` es el aggregate root que controla la consistencia y la lógica de negocio de los datos del paciente y sus métricas. Asegura que las métricas asociadas a un paciente sean válidas y consistentes.

Repositories (Interfaces)

1. MetricsRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de métricas de salud.
- **Métodos:**
 - `findById(String id)`: Busca una métrica por su ID.
 - `save(Metric metric)`: Guarda o actualiza una métrica en la base de datos.
 - `delete(String id)`: Elimina una métrica de la base de datos.

2. PatientRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de pacientes.
- **Métodos:**
 - `findById(String id)`: Busca un paciente por su ID.
 - `save(Patient patient)`: Guarda o actualiza un paciente en la base de datos.
 - `delete(String id)`: Elimina un paciente de la base de datos.

3. DeviceRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de dispositivos.
- **Métodos:**
 - `findById(String id)`: Busca un dispositivo por su ID.
 - `save(Device device)`: Guarda o actualiza un dispositivo en la base de datos.
 - `delete(String id)`: Elimina un dispositivo de la base de datos.

4.2.2. Interface Layer

Controllers

1. MetricsController:

- **Descripción:** Expone endpoints para la gestión de métricas, permitiendo crear, actualizar y consultar métricas de salud.
- **Métodos:**
 - `createMetric(Metric metric)`: Endpoint para crear una nueva métrica de salud.
 - `updateMetric(String id, Metric metric)`: Endpoint para actualizar una métrica existente.
 - `getMetricById(String id)`: Endpoint para obtener los detalles de una métrica por su ID.

2. PatientController:

- **Descripción:** Expone endpoints para la gestión de pacientes, permitiendo crear, actualizar y consultar información del paciente.
- **Métodos:**
 - `registerPatient(Patient patient)`: Endpoint para registrar un nuevo paciente.
 - `updatePatient(String id, Patient patient)`: Endpoint para actualizar la información de un paciente.
 - `getPatientById(String id)`: Endpoint para obtener los detalles de un paciente por su ID.

4.2.2.3. Application Layer

Command Handlers

1. UpdateMetricsHandler:

- **Descripción:** Maneja el comando para actualizar las métricas de salud de un paciente específico.
- **Métodos:**
 - `handle(UpdateMetricsCommand command)`: Valida y aplica el comando para actualizar las métricas de salud.

2. RegisterPatientHandler:

- **Descripción:** Maneja el comando para registrar un nuevo paciente en el sistema.
- **Métodos:**
 - `handle(RegisterPatientCommand command)`: Valida y registra un nuevo paciente en el sistema.

Event Handlers

1. MetricAlertEventHandler:

- **Descripción:** Maneja los eventos de alerta generados por métricas de salud. Asegura que las alertas se procesen y se notifique a los contactos de emergencia si es necesario.
- **Métodos:**
 - `handle(MetricAlertEvent event)`: Procesa las acciones necesarias después de que se genere una alerta basada en las métricas de salud.

Query Handlers

1. GetPatientMetricsHandler:

- **Descripción:** Maneja las consultas para obtener las métricas de salud de un paciente específico.
- **Métodos:**
 - `handle(GetPatientMetricsQuery query)`: Devuelve las métricas de salud del paciente solicitado.

2. GetDeviceDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un dispositivo específico.
- **Métodos:**
 - `handle(GetDeviceDetailsQuery query)`: Devuelve los detalles del dispositivo solicitado.

4.2.2.4. Infrastructure Layer

Repositories (Implementaciones)

1. MetricsRepositoryImpl:

- **Descripción:** Implementación de `MetricsRepository` que define cómo se almacenan y recuperan las métricas de salud desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una métrica por su ID.
 - `save(Metric metric)`: Implementación para guardar o actualizar una métrica en la base de datos.
 - `delete(String id)`: Implementación para eliminar una métrica de la base de datos.

2. PatientRepositoryImpl:

- **Descripción:** Implementación de `PatientRepository` que define cómo se almacenan y recuperan los pacientes desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un paciente por su ID.
 - `save(Patient patient)`: Implementación para guardar o actualizar un paciente en la base de datos.
 - `delete(String id)`: Implementación para eliminar un paciente de la base de datos.

3. DeviceRepositoryImpl:

- **Descripción:** Implementación de `DeviceRepository` que define cómo se almacenan y recuperan los dispositivos desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un dispositivo por su ID.
 - `save(Device device)`: Implementación para guardar o actualizar un dispositivo en la base de datos.
 - `delete(String id)`: Implementación para eliminar un dispositivo de la base de datos.

4.2.2.6. Bounded Context Software Architecture Component Level Diagrams

Components (Edge API):

1. MetricsService Component:

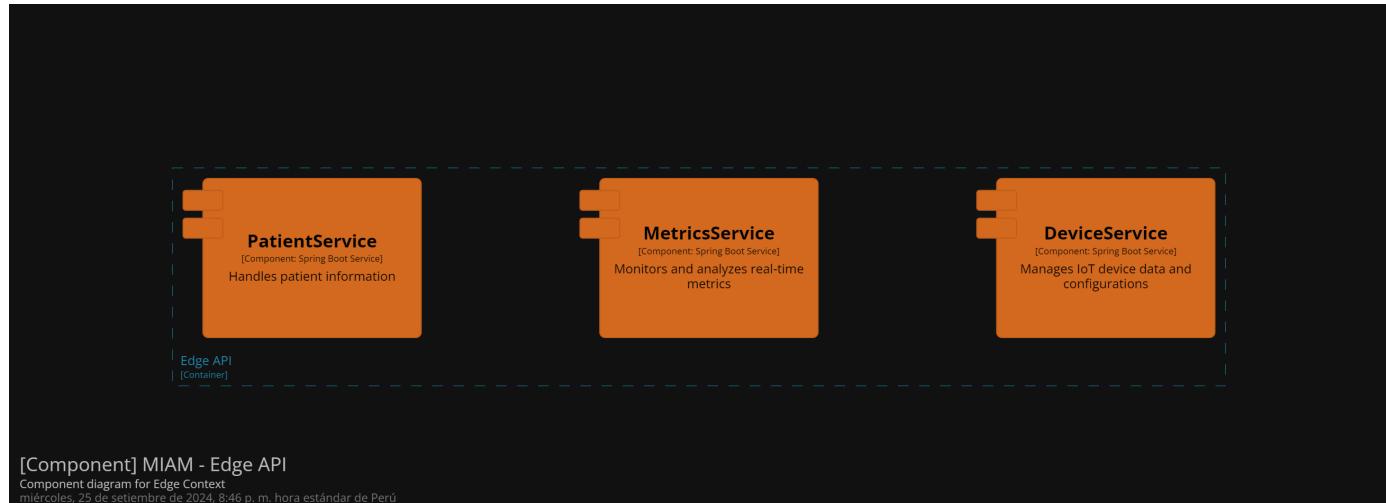
- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de métricas de salud. Interactúa con el **MetricsRepository** para registrar y actualizar métricas.

2. PatientService Component:

- **Descripción:** Componente encargado de la gestión de pacientes. Interactúa con el **PatientRepository** para gestionar el registro y la actualización de pacientes.

3. DeviceService Component:

- **Descripción:** Componente encargado de la gestión de dispositivos IoT. Interactúa con el **DeviceRepository** para gestionar el registro y la actualización de dispositivos.



4.2.2.7. Bounded Context Software Architecture Code Level Diagrams

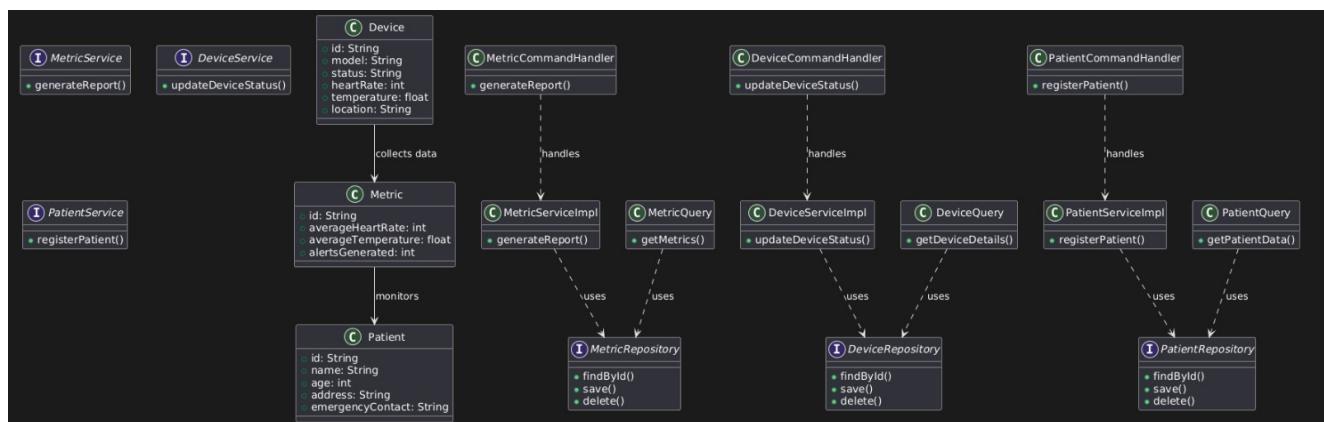
4.2.2.7.1. Bounded Context Domain Layer Class Diagrams

- **Clases importantes:**

- **Metric**
- **Patient**
- **Device**

- **Repositories:**

- **MetricsRepository**
- **PatientRepository**
- **DeviceRepository**



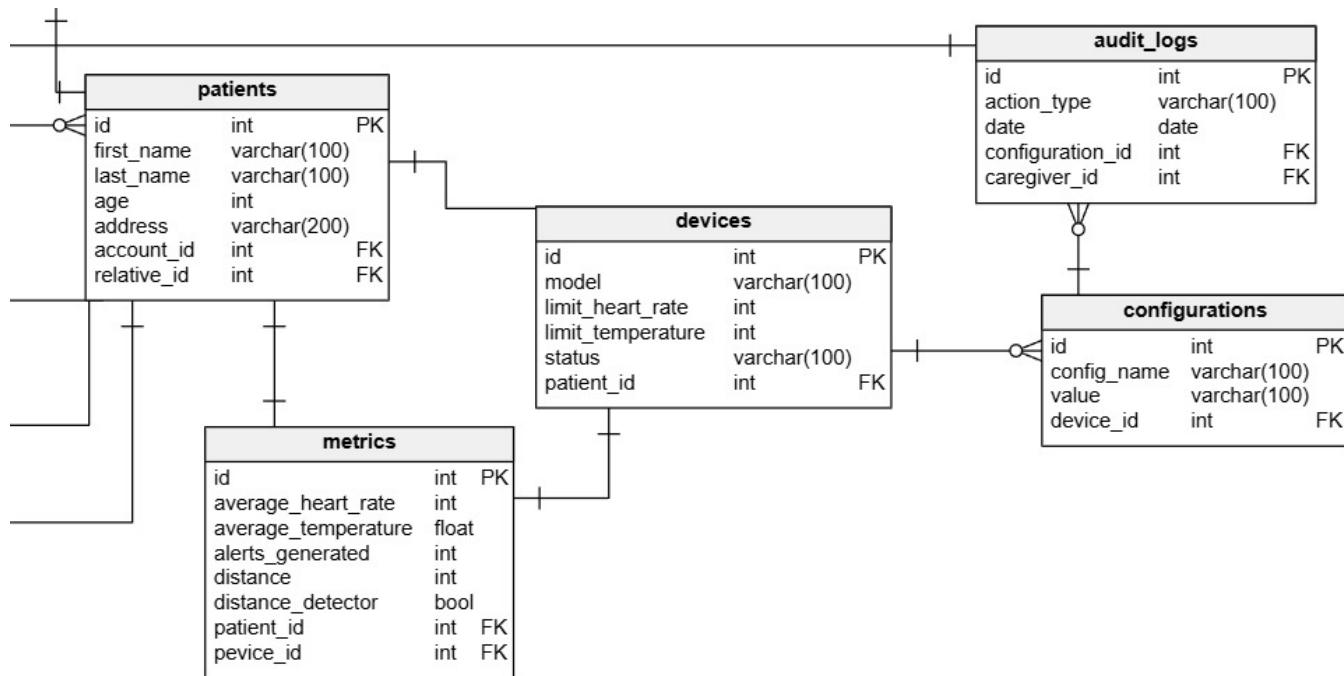
4.2.2.7.2. Bounded Context Database Design Diagram

- **Tablas:**

- **Metric:**

- Columns: **id**, **averageHeartRate**, **averageTemperature**, **alertsGenerated**, **Patient_id**.

- Llave Foránea: `Patient_id`.
- **Patient:**
 - Columnas: `id, name, age, address, emergencyContact`.
- **Device:**
 - Columnas: `id, model, status, Patient_id`.
 - Llave Foránea: `Patient_id`.
- **Relaciones:**
 - `Metric` se relaciona con `Patient` a través de `Patient_id`.
 - `Device` se relaciona con `Patient` a través de `Patient_id`.
 - `Patient` puede tener varias métricas y dispositivos asociados.



4.2.3. Bounded Context: Notification Context

4.2.3.1. Domain Layer

Entities

1. Notification:

- **Descripción:** Representa una notificación enviada a los cuidadores y familiares en caso de anomalías detectadas en la salud del paciente.
- **Atributos:**
 - `id`: Identificador único de la notificación.
 - `message`: Mensaje de la notificación.
 - `notificationType`: Tipo de notificación (emergencia, recordatorio, informe).
 - `sentDate`: Fecha de envío de la notificación.
 - `Caregiver_id`: Referencia al cuidador que recibe la notificación.

2. Caregiver:

- **Descripción:** Representa al cuidador asignado para monitorear la salud del paciente y recibir notificaciones en caso de emergencia.
- **Atributos:**
 - `id`: Identificador único del cuidador.
 - `name`: Nombre del cuidador.
 - `phone`: Número de teléfono del cuidador.

Value Objects

1. NotificationMessage:

- **Descripción:** Objeto de valor que representa el mensaje de una notificación. Asegura que el mensaje cumpla con ciertos requisitos (por ejemplo, longitud máxima, formato adecuado).
- **Atributos:**
 - `message`: Contenido del mensaje de la notificación.

Domain Services

1. NotificationDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con la gestión de notificaciones, como la validación y el envío de mensajes.
- **Métodos:**
 - `sendNotification(Notification notification)`: Envía una notificación a un cuidador específico.
 - `validateNotification(Notification notification)`: Valida que la notificación cumpla con los requisitos establecidos.

Aggregates y Aggregate Root

1. NotificationAggregateRoot:

- **Justificación:** `Notification` es el aggregate root que controla la consistencia y la lógica de negocio para la gestión de notificaciones. Asegura que las notificaciones sean válidas antes de enviarlas a los cuidadores.

Repositories (Interfaces)

1. NotificationRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de notificaciones.
- **Métodos:**
 - `findById(String id)`: Busca una notificación por su ID.
 - `save(Notification notification)`: Guarda o actualiza una notificación en la base de datos.
 - `delete(String id)`: Elimina una notificación de la base de datos.

2. CaregiverRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de cuidadores.
- **Métodos:**
 - `findById(String id)`: Busca un cuidador por su ID.
 - `save(Caregiver caregiver)`: Guarda o actualiza un cuidador en la base de datos.
 - `delete(String id)`: Elimina un cuidador de la base de datos.

4.2.3.2. Interface Layer

Controllers

1. NotificationController:

- **Descripción:** Expone endpoints para la gestión de notificaciones, permitiendo crear, actualizar y consultar notificaciones.
- **Métodos:**
 - `createNotification(Notification notification)`: Endpoint para crear una nueva notificación.
 - `updateNotification(String id, Notification notification)`: Endpoint para actualizar una notificación existente.
 - `getNotificationById(String id)`: Endpoint para obtener los detalles de una notificación por su ID.

2. CaregiverController:

- **Descripción:** Expone endpoints para la gestión de cuidadores, permitiendo registrar, actualizar y consultar información del cuidador.
- **Métodos:**
 - `registerCaregiver(Caregiver caregiver)`: Endpoint para registrar un nuevo cuidador.
 - `updateCaregiver(String id, Caregiver caregiver)`: Endpoint para actualizar la información de un cuidador.
 - `getCaregiverById(String id)`: Endpoint para obtener los detalles de un cuidador por su ID.

4.2.3.3. Application Layer

Command Handlers

1. SendNotificationHandler:

- **Descripción:** Maneja el comando para enviar una notificación a un cuidador específico.
- **Métodos:**
 - `handle(SendNotificationCommand command)`: Valida y aplica el comando para enviar una notificación.

2. RegisterCaregiverHandler:

- **Descripción:** Maneja el comando para registrar un nuevo cuidador en el sistema.
- **Métodos:**
 - `handle(RegisterCaregiverCommand command)`: Valida y registra un nuevo cuidador en el sistema.

Event Handlers

1. NotificationSentEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando una notificación es enviada. Asegura que cualquier acción posterior se realice de manera adecuada.
- **Métodos:**
 - `handle(NotificationSentEvent event)`: Procesa las acciones necesarias después de que una notificación ha sido enviada.

Query Handlers

1. GetNotificationDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de una notificación específica.
- **Métodos:**
 - `handle(GetNotificationDetailsQuery query)`: Devuelve los detalles de la notificación solicitada.

2. GetCaregiverDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un cuidador específico.
- **Métodos:**
 - `handle(GetCaregiverDetailsQuery query)`: Devuelve los detalles del cuidador solicitado.

4.2.3.4. Infrastructure Layer

Repositories (Implementaciones)

1. NotificationRepositoryImpl:

- **Descripción:** Implementación de `NotificationRepository` que define cómo se almacenan y recuperan las notificaciones desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una notificación por su ID.
 - `save(Notification notification)`: Implementación para guardar o actualizar una notificación en la base de datos.
 - `delete(String id)`: Implementación para eliminar una notificación de la base de datos.

2. CaregiverRepositoryImpl:

- **Descripción:** Implementación de `CaregiverRepository` que define cómo se almacenan y recuperan los cuidadores desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un cuidador por su ID.
 - `save(Caregiver caregiver)`: Implementación para guardar o actualizar un cuidador en la base de datos.
 - `delete(String id)`: Implementación para eliminar un cuidador de la base de datos.

4.2.3.6. Bounded Context Software Architecture Component Level Diagrams

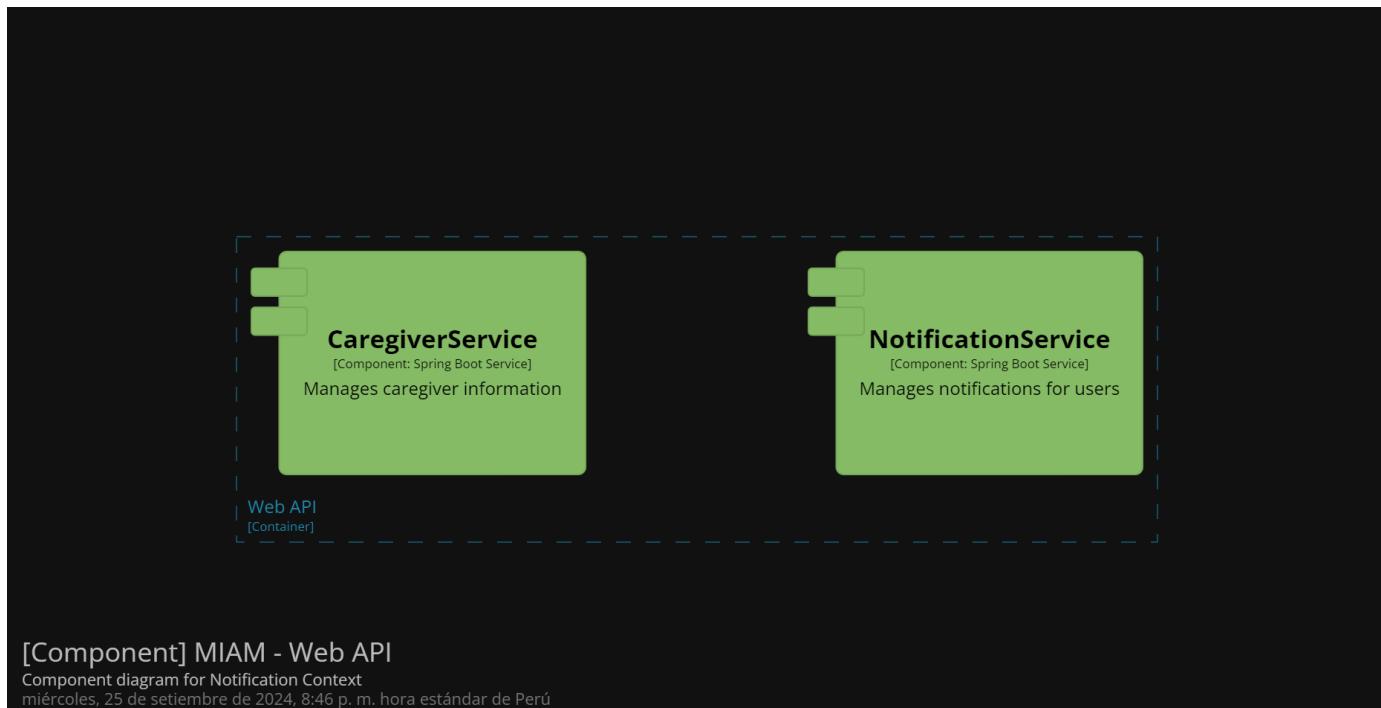
Components (Web API):

1. NotificationService Component:

- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de notificaciones. Interactúa con el `NotificationRepository` para registrar y enviar notificaciones.

2. CaregiverService Component:

- **Descripción:** Componente encargado de la gestión de cuidadores. Interactúa con el `CaregiverRepository` para gestionar el registro y la actualización de cuidadores.



4.2.3.7. Bounded Context Software Architecture Code Level Diagrams

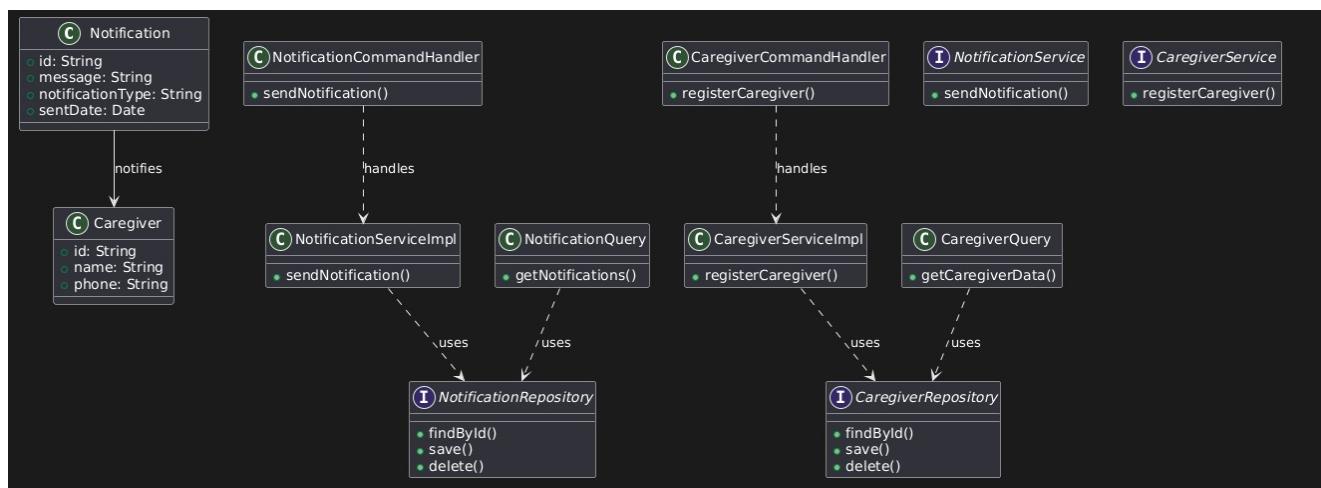
4.2.3.7.1. Bounded Context Domain Layer Class Diagrams

- **Clases importantes:**

- **Notification**
- **Caregiver**

- **Repositories:**

- **NotificationRepository**
- **CaregiverRepository**



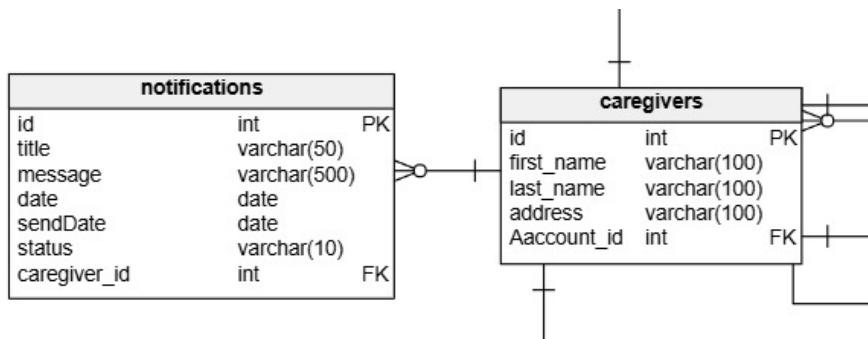
4.2.3.7.2. Bounded Context Database Design Diagram

- **Tablas:**

- **Notification:**
 - Columnas: `id`, `message`, `notificationType`, `sentDate`, `Caregiver_id`.
 - Llave Foránea: `Caregiver_id`.
- **Caregiver:**
 - Columnas: `id`, `name`, `phone`.

- **Relaciones:**

- Notification se relaciona con Caregiver a través de Caregiver_id.
- Caregiver puede recibir varias notificaciones.



4.2.4. Bounded Context: Metrics Context

4.2.4.1. Domain Layer

Entities

1. Metrics:

- Descripción:** Representa los datos de salud recopilados por un dispositivo, incluyendo frecuencia cardíaca y temperatura.
- Atributos:**
 - id:** Identificador único de las métricas.
 - averageHeartRate:** Frecuencia cardíaca promedio registrada.
 - averageTemperature:** Temperatura promedio registrada.
 - alertsGenerated:** Número de alertas generadas basadas en las métricas.
 - Patient_id:** Referencia al paciente al que pertenecen las métricas.

2. Patient:

- Descripción:** Representa al paciente al que se le realiza el monitoreo de salud a través del dispositivo IoT.
- Atributos:**
 - id:** Identificador único del paciente.
 - name:** Nombre del paciente.
 - age:** Edad del paciente.
 - address:** Dirección del paciente.
 - emergencyContact:** Contacto de emergencia para el paciente.

Value Objects

1. HealthMetric:

- Descripción:** Objeto de valor que representa una métrica de salud específica, como la frecuencia cardíaca o la temperatura. Asegura que las métricas de salud cumplan con los requisitos establecidos.
- Atributos:**
 - metricName:** Nombre de la métrica de salud (por ejemplo, frecuencia cardíaca, temperatura).
 - value:** Valor de la métrica.

Domain Services

1. MetricsDomainService:

- Descripción:** Contiene la lógica de negocio relacionada con las métricas de salud, como el cálculo de promedios y la generación de alertas.
- Métodos:**
 - calculateAverage(Metric metric):** Calcula el promedio de una métrica de salud.
 - generateAlertIfNecessary(Metric metric):** Genera una alerta si la métrica supera ciertos umbrales.

Aggregates y Aggregate Root

1. PatientAggregateRoot:

- Justificación:** Patient es el aggregate root que controla la consistencia y la lógica de negocio de los datos del paciente y sus métricas. Asegura que las métricas asociadas a un paciente sean válidas y consistentes.

Repositories (Interfaces)

1. MetricsRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de métricas de salud.
- **Métodos:**
 - `findById(String id)`: Busca una métrica por su ID.
 - `save(Metric metric)`: Guarda o actualiza una métrica en la base de datos.
 - `delete(String id)`: Elimina una métrica de la base de datos.

2. PatientRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de pacientes.
- **Métodos:**
 - `findById(String id)`: Busca un paciente por su ID.
 - `save(Patient patient)`: Guarda o actualiza un paciente en la base de datos.
 - `delete(String id)`: Elimina un paciente de la base de datos.

4.2.4.2. Interface Layer

Controllers

1. MetricsController:

- **Descripción:** Expone endpoints para la gestión de métricas, permitiendo crear, actualizar y consultar métricas de salud.
- **Métodos:**
 - `createMetric(Metric metric)`: Endpoint para crear una nueva métrica de salud.
 - `updateMetric(String id, Metric metric)`: Endpoint para actualizar una métrica existente.
 - `getMetricById(String id)`: Endpoint para obtener los detalles de una métrica por su ID.

2. PatientController:

- **Descripción:** Expone endpoints para la gestión de pacientes, permitiendo crear, actualizar y consultar información del paciente.
- **Métodos:**
 - `registerPatient(Patient patient)`: Endpoint para registrar un nuevo paciente.
 - `updatePatient(String id, Patient patient)`: Endpoint para actualizar la información de un paciente.
 - `getPatientById(String id)`: Endpoint para obtener los detalles de un paciente por su ID.

4.2.4.3. Application Layer

Command Handlers

1. UpdateMetricsHandler:

- **Descripción:** Maneja el comando para actualizar las métricas de salud de un paciente específico.
- **Métodos:**
 - `handle(UpdateMetricsCommand command)`: Valida y aplica el comando para actualizar las métricas de salud.

2. RegisterPatientHandler:

- **Descripción:** Maneja el comando para registrar un nuevo paciente en el sistema.
- **Métodos:**
 - `handle(RegisterPatientCommand command)`: Valida y registra un nuevo paciente en el sistema.

Event Handlers

1. MetricAlertEventHandler:

- **Descripción:** Maneja los eventos de alerta generados por métricas de salud. Asegura que las alertas se procesen y se notifique a los contactos de emergencia si es necesario.
- **Métodos:**
 - `handle(MetricAlertEvent event)`: Procesa las acciones necesarias después de que se genere una alerta basada en las métricas de salud.

Query Handlers

1. GetPatientMetricsHandler:

- **Descripción:** Maneja las consultas para obtener las métricas de salud de un paciente específico.
- **Métodos:**
 - `handle(GetPatientMetricsQuery query)`: Devuelve las métricas de salud del paciente solicitado.

2. GetMetricsDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de una métrica específica.
- **Métodos:**
 - `handle(GetMetricsDetailsQuery query)`: Devuelve los detalles de la métrica solicitada.

4.2.4.4. Infrastructure Layer

Repositories (Implementaciones)

1. MetricsRepositoryImpl:

- **Descripción:** Implementación de `MetricsRepository` que define cómo se almacenan y recuperan las métricas de salud desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una métrica por su ID.
 - `save(Metric metric)`: Implementación para guardar o actualizar una métrica en la base de datos.
 - `delete(String id)`: Implementación para eliminar una métrica de la base de datos.

2. PatientRepositoryImpl:

- **Descripción:** Implementación de `PatientRepository` que define cómo se almacenan y recuperan los pacientes desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un paciente por su ID.
 - `save(Patient patient)`: Implementación para guardar o actualizar un paciente en la base de datos.
 - `delete(String id)`: Implementación para eliminar un paciente de la base de datos.

4.2.4.6. Bounded Context Software Architecture Component Level Diagrams

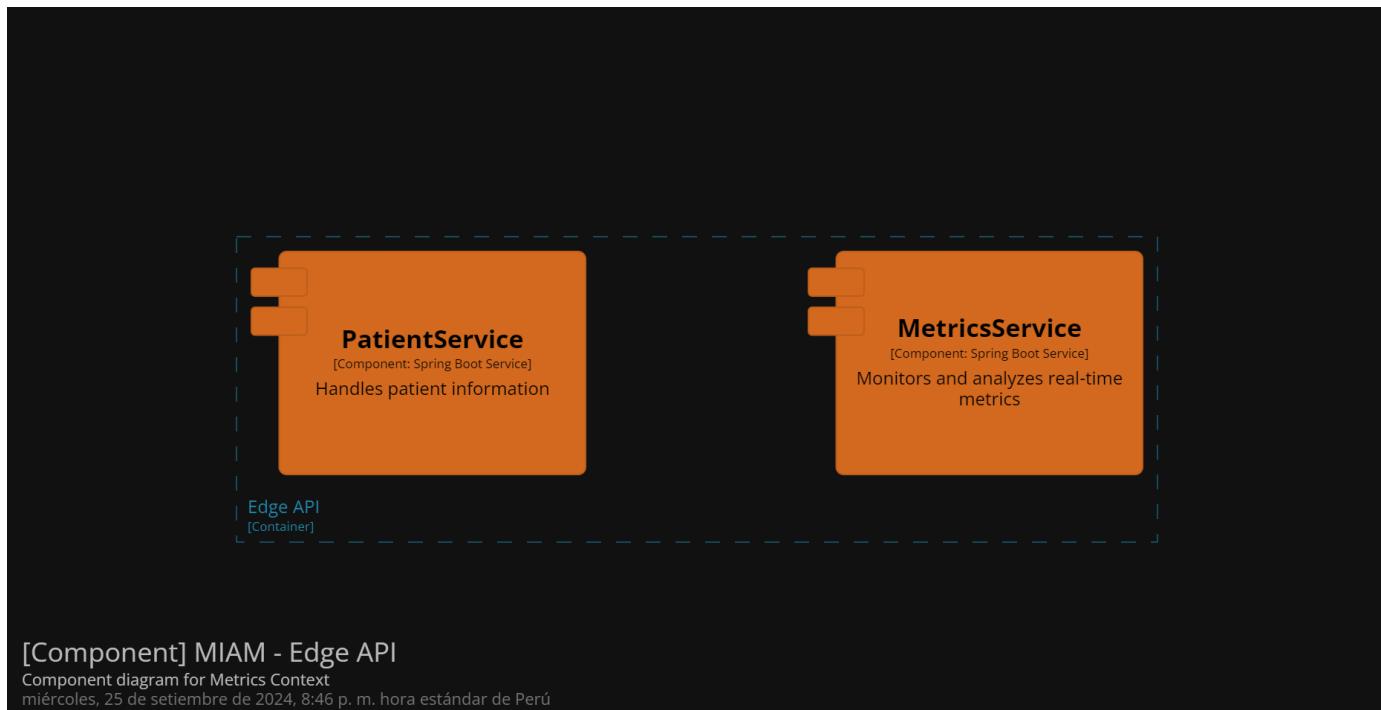
Components (Edge API):

1. MetricsService Component:

- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de métricas de salud. Interactúa con el `MetricsRepository` para registrar y actualizar métricas.

2. PatientService Component:

- **Descripción:** Componente encargado de la gestión de información de los pacientes. Interactúa con el `PatientRepository`. Cada componente interactúa con sus respectivos repositorios y handlers para ejecutar la lógica de negocio y responder a las solicitudes de forma coherente con el patrón CQRS.



4.2.4.7. Bounded Context Software Architecture Code Level Diagrams

4.2.4.7.1. Bounded Context Domain Layer Class Diagrams

En esta sección se detallan las clases del Domain Layer, sus atributos, métodos y relaciones. Clases importantes:

Metrics:

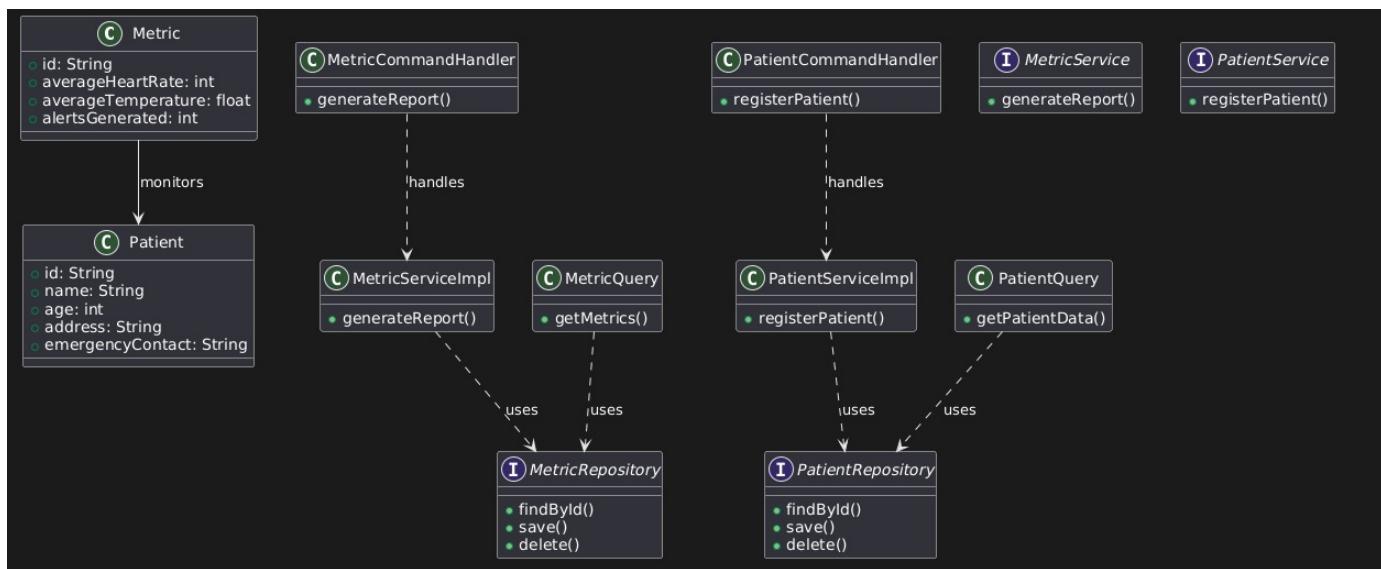
- **Atributos:** id, averageHeartRate, averageTemperature, alertsGenerated, Patient_id.
- **Métodos:** calculateAverage(), generateAlert(threshold).

Patient:

- **Atributos:** id, name, age, address, emergencyContact.
- **Métodos:** updateContact(newContact).

Repositories:

MetricsRepository, PatientRepository.

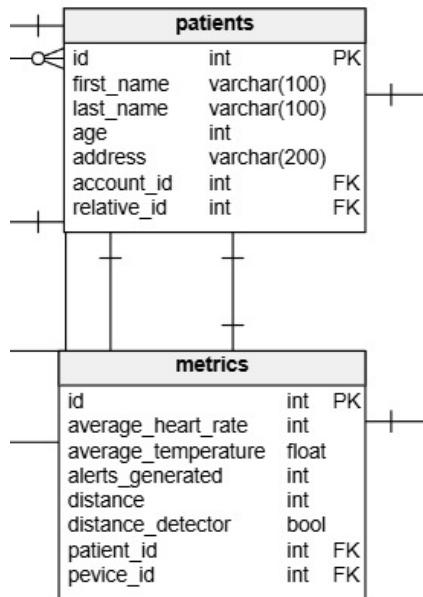


4.2.4.7.2. Bounded Context Database Design Diagram

En esta sección se define el diseño de la base de datos para persistir las entidades de este contexto.

Tablas:

- **Metrics:** Columnas: id, averageHeartRate, averageTemperature, alertsGenerated, Patient_id. Llave Foránea: Patient_id.
- **Patient:** Columnas: id, name, age, address, emergencyContact. Relaciones: Metrics se relaciona con Patient a través de Patient_id.



4.2.5. Bounded Context: Payment Context

4.2.5.1. Domain Layer

Entidades**1. Account:**

- **Descripción:** Representa la cuenta de usuario en el sistema, incluyendo información de contacto y métodos de pago.
- **Atributos:**
 - **id:** Identificador único de la cuenta.
 - **username:** Nombre de usuario asociado a la cuenta.
 - **password:** Contraseña de la cuenta.
 - **role:** Rol del usuario dentro del sistema.
 - **phoneNumber:** Número de teléfono del usuario.

2. Payment:

- **Descripción:** Representa un pago realizado por el usuario para la suscripción del servicio.
- **Atributos:**
 - **id:** Identificador único del pago.
 - **amount:** Monto del pago realizado.
 - **paymentDate:** Fecha en la que se realizó el pago.
 - **paymentStatus:** Estado actual del pago (pendiente, completado, fallido).
 - **Account_id:** Referencia a la cuenta asociada al pago.

3. Subscription:

- **Descripción:** Representa una suscripción al servicio que tiene un usuario, incluyendo fechas de inicio y fin, y el plan asociado.
- **Atributos:**
 - **id:** Identificador único de la suscripción.
 - **startDate:** Fecha de inicio de la suscripción.
 - **endDate:** Fecha de finalización de la suscripción.
 - **status:** Estado de la suscripción (activa, inactiva, cancelada).
 - **Account_id:** Referencia a la cuenta a la que pertenece la suscripción.
 - **Plan_id:** Referencia al plan al que está suscrito el usuario.

4. Plan:

- **Descripción:** Representa un plan de suscripción disponible en el sistema, con sus características y precio.
- **Atributos:**
 - `id`: Identificador único del plan.
 - `planName`: Nombre del plan de suscripción.
 - `description`: Descripción del plan.
 - `price`: Precio del plan de suscripción.

5. CreditCard:

- **Descripción:** Representa la tarjeta de crédito asociada a una cuenta para realizar los pagos.
- **Atributos:**
 - `id`: Identificador único de la tarjeta.
 - `number`: Número de la tarjeta de crédito.
 - `cvv`: Código de seguridad de la tarjeta.
 - `expirationDate`: Fecha de expiración de la tarjeta.
 - `Account_id`: Referencia a la cuenta asociada a la tarjeta.

Value Objects

1. PaymentAmount:

- **Descripción:** Objeto de valor que representa el monto del pago. Asegura que el monto sea positivo y esté dentro de los límites permitidos.
- **Atributos:**
 - `amount`: Monto del pago.

2. SubscriptionStatus:

- **Descripción:** Objeto de valor que representa el estado de una suscripción (activa, inactiva, cancelada).
- **Atributos:**
 - `status`: Estado de la suscripción.

Domain Services

1. PaymentDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con la gestión de pagos, como la validación de montos y la gestión del estado de los pagos.
- **Métodos:**
 - `processPayment(Payment payment)`: Procesa el pago después de validar que todos los datos son correctos.
 - `validatePaymentAmount(PaymentAmount amount)`: Valida que el monto del pago esté dentro de los límites permitidos.

2. SubscriptionDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con la gestión de suscripciones, como la activación y cancelación de suscripciones.
- **Métodos:**
 - `activateSubscription(Subscription subscription)`: Activa una suscripción después de verificar que el pago fue completado.
 - `cancelSubscription(Subscription subscription)`: Cancela una suscripción y gestiona los reembolsos si corresponde.

Aggregates y Aggregate Root

1. AccountAggregateRoot:

- **Justificación:** `Account` es el aggregate root que controla la consistencia y la lógica de negocio para la gestión de cuentas, pagos y suscripciones. Asegura que todos los pagos y suscripciones relacionados con una cuenta sean válidos y consistentes.

Repositories (Interfaces)

1. AccountRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de cuentas de usuario.
- **Métodos:**
 - `findById(String id)`: Busca una cuenta por su ID.
 - `save(Account account)`: Guarda o actualiza una cuenta en la base de datos.

- `delete(String id)`: Elimina una cuenta de la base de datos.

2. PaymentRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de pagos.
- **Métodos:**
 - `findById(String id)`: Busca un pago por su ID.
 - `save(Payment payment)`: Guarda o actualiza un pago en la base de datos.
 - `delete(String id)`: Elimina un pago de la base de datos.

3. SubscriptionRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de suscripciones.
- **Métodos:**
 - `findById(String id)`: Busca una suscripción por su ID.
 - `save(Subscription subscription)`: Guarda o actualiza una suscripción en la base de datos.
 - `delete(String id)`: Elimina una suscripción de la base de datos.

4. PlanRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de planes de suscripción.
- **Métodos:**
 - `findById(String id)`: Busca un plan por su ID.
 - `save(Plan plan)`: Guarda o actualiza un plan en la base de datos.
 - `delete(String id)`: Elimina un plan de la base de datos.

5. CreditCardRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de tarjetas de crédito.
- **Métodos:**
 - `findById(String id)`: Busca una tarjeta de crédito por su ID.
 - `save(CreditCard creditCard)`: Guarda o actualiza una tarjeta en la base de datos.
 - `delete(String id)`: Elimina una tarjeta de la base de datos.

4.2.5.2. Interface Layer

Controllers

1. AccountController:

- **Descripción:** Expone endpoints para la gestión de cuentas de usuario, permitiendo registrar, actualizar y consultar cuentas.
- **Métodos:**
 - `registerAccount(Account account)`: Endpoint para registrar una nueva cuenta.
 - `updateAccount(String id, Account account)`: Endpoint para actualizar la información de una cuenta.
 - `getAccountById(String id)`: Endpoint para obtener los detalles de una cuenta por su ID.

2. PaymentController:

- **Descripción:** Expone endpoints para la gestión de pagos, permitiendo realizar, consultar y actualizar pagos.
- **Métodos:**
 - `processPayment(Payment payment)`: Endpoint para realizar un pago.
 - `updatePayment(String id, Payment payment)`: Endpoint para actualizar la información de un pago.
 - `getPaymentById(String id)`: Endpoint para obtener los detalles de un pago por su ID.

3. SubscriptionController:

- **Descripción:** Expone endpoints para la gestión de suscripciones, permitiendo registrar, actualizar y consultar suscripciones.
- **Métodos:**
 - `createSubscription(Subscription subscription)`: Endpoint para crear una nueva suscripción.
 - `updateSubscription(String id, Subscription subscription)`: Endpoint para actualizar la información de una suscripción.
 - `getSubscriptionById(String id)`: Endpoint para obtener los detalles de una suscripción por su ID.

4. PlanController:

- **Descripción:** Expone endpoints para la gestión de planes de suscripción, permitiendo registrar, actualizar y consultar planes.
- **Métodos:**

- `createPlan(Plan plan)`: Endpoint para crear un nuevo plan de suscripción.
- `updatePlan(String id, Plan plan)`: Endpoint para actualizar la información de un plan.
- `getPlanById(String id)`: Endpoint para obtener los detalles de un plan por su ID.

4.2.5.3. Application Layer

Command Handlers

1. `ProcessPaymentHandler`:

- **Descripción:** Maneja el comando para procesar un pago en el sistema.
- **Métodos:**
 - `handle(ProcessPaymentCommand command)`: Valida y aplica el comando para procesar el pago.

2. `CreateSubscriptionHandler`:

- **Descripción:** Maneja el comando para crear una nueva suscripción en el sistema.
- **Métodos:**
 - `handle(CreateSubscriptionCommand command)`: Valida y crea una nueva suscripción en el sistema.

3. `CreateAccountHandler`:

- **Descripción:** Maneja el comando para registrar una nueva cuenta en el sistema.
- **Métodos:**
 - `handle(CreateAccountCommand command)`: Valida y crea una nueva cuenta en el sistema.

4. `CreatePlanHandler`:

- **Descripción:** Maneja el comando para registrar un nuevo plan de suscripción en el sistema.
- **Métodos:**
 - `handle(CreatePlanCommand command)`: Valida y crea un nuevo plan en el sistema.

Event Handlers

1. `PaymentProcessedEventHandler`:

- **Descripción:** Maneja los eventos que se disparan cuando un pago ha sido procesado, actualizando el estado de la suscripción asociada.
- **Métodos:**
 - `handle(PaymentProcessedEvent event)`: Procesa las acciones necesarias después de que un pago ha sido procesado.

2. `SubscriptionCreatedEventHandler`:

- **Descripción:** Maneja los eventos que se disparan cuando se crea una nueva suscripción.
- **Métodos:**
 - `handle(SubscriptionCreatedEvent event)`: Ejecuta las acciones necesarias después de la creación de una suscripción.

Query Handlers

1. `GetAccountDetailsHandler`:

- **Descripción:** Maneja las consultas para obtener los detalles de una cuenta específica.
- **Métodos:**
 - `handle(GetAccountDetailsQuery query)`: Devuelve los detalles de la cuenta solicitada.

2. `GetPaymentDetailsHandler`:

- **Descripción:** Maneja las consultas para obtener los detalles de un pago específico.
- **Métodos:**
 - `handle(GetPaymentDetailsQuery query)`: Devuelve los detalles del pago solicitado.

3. `GetSubscriptionDetailsHandler`:

- **Descripción:** Maneja las consultas para obtener los detalles de una suscripción específica.
- **Métodos:**
 - `handle(GetSubscriptionDetailsQuery query)`: Devuelve los detalles de la suscripción solicitada.

4. `GetPlanDetailsHandler`:

- **Descripción:** Maneja las consultas para obtener los detalles de un plan específico.
- **Métodos:**
 - `handle(GetPlanDetailsQuery query)`: Devuelve los detalles del plan solicitado.

4.2.5.4. Infrastructure Layer

Repositories (Implementaciones)

1. AccountRepositoryImpl:

- **Descripción:** Implementación de `AccountRepository` que define cómo se almacenan y recuperan las cuentas desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una cuenta por su ID.
 - `save(Account account)`: Implementación para guardar o actualizar una cuenta en la base de datos.
 - `delete(String id)`: Implementación para eliminar una cuenta de la base de datos.

2. PaymentRepositoryImpl:

- **Descripción:** Implementación de `PaymentRepository` que define cómo se almacenan y recuperan los pagos desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un pago por su ID.
 - `save(Payment payment)`: Implementación para guardar o actualizar un pago en la base de datos.
 - `delete(String id)`: Implementación para eliminar un pago de la base de datos.

3. SubscriptionRepositoryImpl:

- **Descripción:** Implementación de `SubscriptionRepository` que define cómo se almacenan y recuperan las suscripciones desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una suscripción por su ID.
 - `save(Subscription subscription)`: Implementación para guardar o actualizar una suscripción en la base de datos.
 - `delete(String id)`: Implementación para eliminar una suscripción de la base de datos.

4. PlanRepositoryImpl:

- **Descripción:** Implementación de `PlanRepository` que define cómo se almacenan y recuperan los planes desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un plan por su ID.
 - `save(Plan plan)`: Implementación para guardar o actualizar un plan en la base de datos.
 - `delete(String id)`: Implementación para eliminar un plan de la base de datos.

5. CreditCardRepositoryImpl:

- **Descripción:** Implementación de `CreditCardRepository` que define cómo se almacenan y recuperan las tarjetas de crédito desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una tarjeta de crédito por su ID.
 - `save(CreditCard creditCard)`: Implementación para guardar o actualizar una tarjeta en la base de datos.
 - `delete(String id)`: Implementación para eliminar una tarjeta de la base de datos.

4.2.5.6. Bounded Context Software Architecture Component Level Diagrams

Components (Web API):

1. PaymentService Component:

- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de pagos. Interactúa con el `PaymentRepository` para registrar y actualizar pagos.

2. SubscriptionService Component:

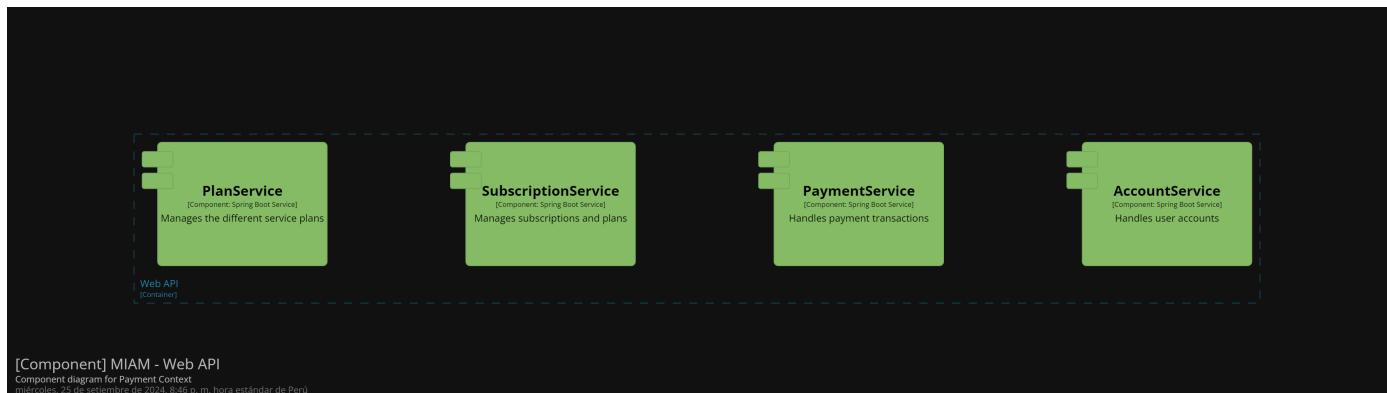
- **Descripción:** Componente encargado de la gestión de suscripciones. Interactúa con el `SubscriptionRepository` para gestionar el registro y la actualización de suscripciones.

3. AccountService Component:

- **Descripción:** Componente encargado de la gestión de cuentas de usuario. Interactúa con el **AccountRepository** para gestionar el registro y la actualización de cuentas.

4. PlanService Component:

- **Descripción:** Componente encargado de la gestión de planes de suscripción. Interactúa con el **PlanRepository** para gestionar el registro y la actualización de planes.



4.2.5.7. Bounded Context Software Architecture Code Level Diagrams

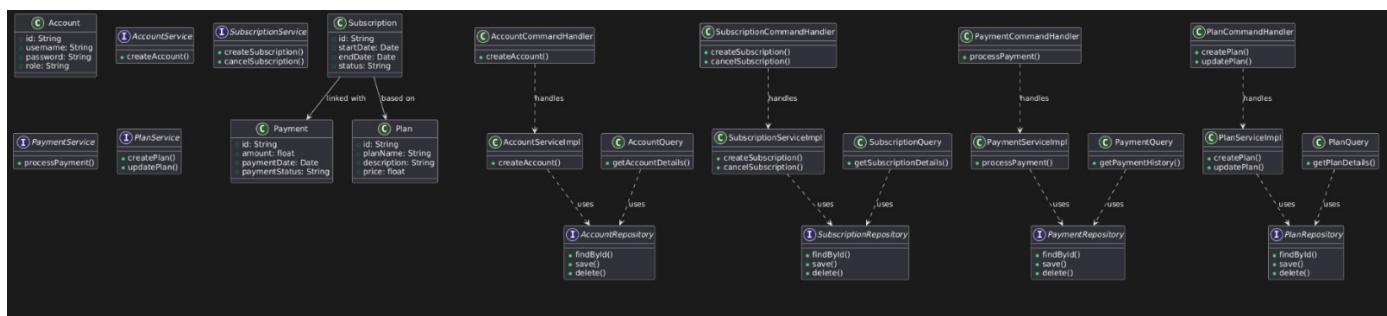
4.2.5.7.1. Bounded Context Domain Layer Class Diagrams

- **Clases importantes:**

- **Account**
- **Payment**
- **Subscription**
- **Plan**
- **CreditCard**

- **Repositories:**

- **AccountRepository**
- **PaymentRepository**
- **SubscriptionRepository**
- **PlanRepository**
- **CreditCardRepository**

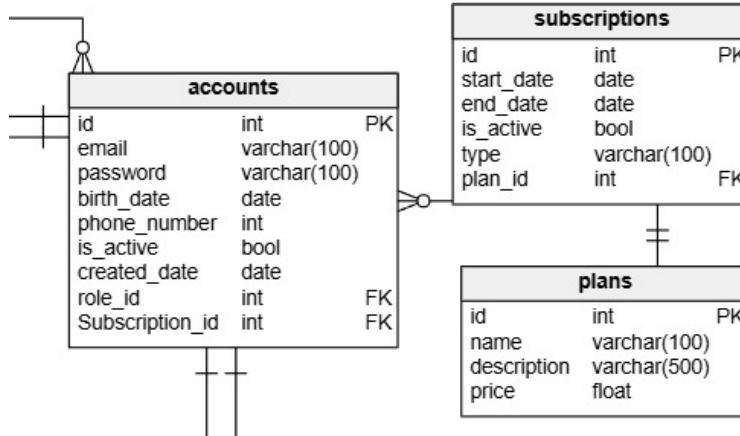


4.2.5.7.2. Bounded Context Database Design Diagram

- **Tablas:**

- **Account:**
 - Columnas: `id`, `username`, `password`, `role`, `phoneNumber`.
- **Payment:**
 - Columnas: `id`, `amount`, `paymentDate`, `paymentStatus`, `Account_id`.
 - Llave Foránea: `Account_id`.
- **Subscription:**
 - Columnas: `id`, `startDate`, `endDate`, `status`, `Account_id`, `Plan_id`.
 - Llave Foránea: `Account_id`, `Plan_id`.
- **Plan:**

- Columnas: `id`, `planName`, `description`, `price`.
- **CreditCard:**
 - Columnas: `id`, `number`, `cvv`, `expirationDate`, `Account_id`.
 - Llave Foránea: `Account_id`.
- **Relaciones:**
 - `Payment` se relaciona con `Account` a través de `Account_id`.
 - `Subscription` se relaciona con `Account` y `Plan` a través de `Account_id` y `Plan_id`.
 - `CreditCard` se relaciona con `Account` a través de `Account_id`.



4.2.6. Bounded Context: Configuration Context

4.2.6.1. Domain Layer

Entities

1. Configuration:

- **Descripción:** Representa las configuraciones específicas de un dispositivo IoT, como umbrales de alertas y parámetros de monitoreo.
- **Atributos:**
 - `id`: Identificador único de la configuración.
 - `configName`: Nombre de la configuración.
 - `value`: Valor de la configuración.
 - `Device_id`: Referencia al dispositivo al que pertenece la configuración.

2. Device:

- **Descripción:** Representa un dispositivo IoT en el sistema, utilizado para recopilar datos de salud del paciente.
- **Atributos:**
 - `id`: Identificador único del dispositivo.
 - `model`: Modelo del dispositivo.
 - `status`: Estado actual del dispositivo (activo, inactivo).
 - `location`: Ubicación del dispositivo.

Value Objects

1. ConfigParameter:

- **Descripción:** Objeto de valor que representa un parámetro de configuración específico. Asegura que el valor del parámetro esté dentro de los límites permitidos.
- **Atributos:**
 - `name`: Nombre del parámetro.
 - `value`: Valor del parámetro.

Domain Services

1. ConfigurationDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con la gestión de configuraciones, como la validación de parámetros y la aplicación de cambios.

- **Métodos:**

- `applyConfiguration(Device device, Configuration config)`: Aplica una configuración a un dispositivo después de validar que todos los parámetros son correctos.
- `validateConfigParameter(ConfigParameter param)`: Valida que un parámetro de configuración esté dentro de los límites permitidos.

Aggregates y Aggregate Root

1. ConfigurationAggregateRoot:

- **Justificación:** `Configuration` es el aggregate root que controla la consistencia y la lógica de negocio para la gestión de configuraciones. Asegura que todas las configuraciones aplicadas a un dispositivo sean válidas y consistentes.

Repositories (Interfaces)

1. ConfigurationRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de configuraciones.
- **Métodos:**
 - `findById(String id)`: Busca una configuración por su ID.
 - `save(Configuration config)`: Guarda o actualiza una configuración en la base de datos.
 - `delete(String id)`: Elimina una configuración de la base de datos.

2. DeviceRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de dispositivos.
- **Métodos:**
 - `findById(String id)`: Busca un dispositivo por su ID.
 - `save(Device device)`: Guarda o actualiza un dispositivo en la base de datos.
 - `delete(String id)`: Elimina un dispositivo de la base de datos.

4.2.6.2. Interface Layer

Controllers

1. ConfigurationController:

- **Descripción:** Expone endpoints para la gestión de configuraciones, permitiendo crear, actualizar y consultar configuraciones.
- **Métodos:**
 - `createConfiguration(Configuration configuration)`: Endpoint para crear una nueva configuración.
 - `updateConfiguration(String id, Configuration configuration)`: Endpoint para actualizar una configuración existente.
 - `getConfigurationById(String id)`: Endpoint para obtener los detalles de una configuración por su ID.

2. DeviceController:

- **Descripción:** Expone endpoints para la gestión de dispositivos, permitiendo crear, actualizar y consultar dispositivos.
- **Métodos:**
 - `registerDevice(Device device)`: Endpoint para registrar un nuevo dispositivo.
 - `updateDeviceStatus(String id, String status)`: Endpoint para actualizar el estado de un dispositivo.
 - `getDeviceById(String id)`: Endpoint para obtener los detalles de un dispositivo por su ID.

4.2.6.3. Application Layer

Command Handlers

1. UpdateConfigurationHandler:

- **Descripción:** Maneja el comando para actualizar una configuración de un dispositivo específico.
- **Métodos:**
 - `handle(UpdateConfigurationCommand command)`: Valida y aplica el comando para actualizar la configuración.

2. RegisterDeviceHandler:

- **Descripción:** Maneja el comando para registrar un nuevo dispositivo en el sistema.
- **Métodos:**
 - `handle(RegisterDeviceCommand command)`: Valida y registra un nuevo dispositivo en el sistema.

Event Handlers

1. ConfigurationUpdatedEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando se actualiza una configuración. Asegura que cualquier cambio sea procesado adecuadamente.
- **Métodos:**
 - `handle(ConfigurationUpdatedEvent event)`: Procesa las acciones necesarias después de que una configuración ha sido actualizada.

2. DeviceRegisteredEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando se registra un nuevo dispositivo.
- **Métodos:**
 - `handle(DeviceRegisteredEvent event)`: Ejecuta las acciones necesarias después del registro de un dispositivo.

Query Handlers

1. GetConfigurationDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de una configuración específica.
- **Métodos:**
 - `handle(GetConfigurationDetailsQuery query)`: Devuelve los detalles de la configuración solicitada.

2. GetDeviceDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un dispositivo específico.
- **Métodos:**
 - `handle(GetDeviceDetailsQuery query)`: Devuelve los detalles del dispositivo solicitado.

4.2.6.4. Infrastructure Layer

Repositories (Implementaciones)

1. ConfigurationRepositoryImpl:

- **Descripción:** Implementación de `ConfigurationRepository` que define cómo se almacenan y recuperan las configuraciones desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una configuración por su ID.
 - `save(Configuration config)`: Implementación para guardar o actualizar una configuración en la base de datos.
 - `delete(String id)`: Implementación para eliminar una configuración de la base de datos.

2. DeviceRepositoryImpl:

- **Descripción:** Implementación de `DeviceRepository` que define cómo se almacenan y recuperan los dispositivos desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un dispositivo por su ID.
 - `save(Device device)`: Implementación para guardar o actualizar un dispositivo en la base de datos.
 - `delete(String id)`: Implementación para eliminar un dispositivo de la base de datos.

4.2.6.6. Bounded Context Software Architecture Component Level Diagrams

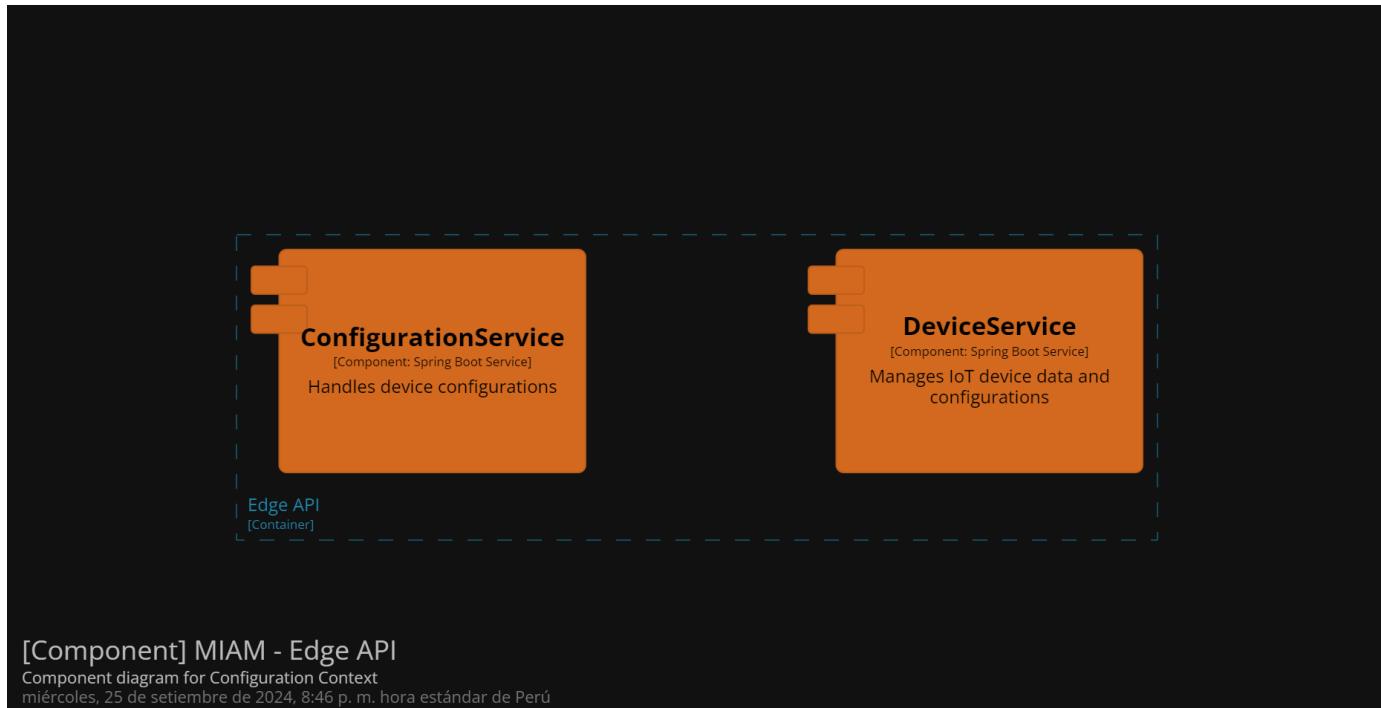
Components (Edge API):

1. ConfigurationService Component:

- **Descripción:** Componente que encapsula la lógica de negocio relacionada con la gestión de configuraciones. Interactúa con el `ConfigurationRepository` para registrar y actualizar configuraciones.

2. DeviceService Component:

- **Descripción:** Componente encargado de la gestión de dispositivos IoT. Interactúa con el `DeviceRepository` para gestionar el registro y la actualización de dispositivos.



4.2.6.7. Bounded Context Software Architecture Code Level Diagrams

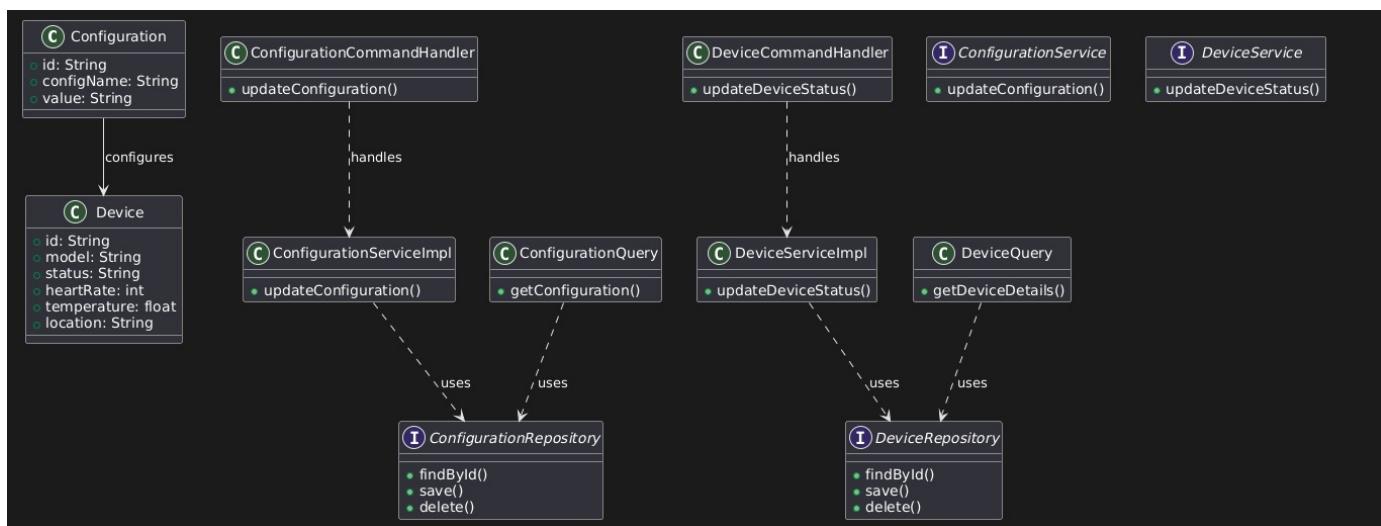
4.2.6.7.1. Bounded Context Domain Layer Class Diagrams

- **Clases importantes:**

- Configuration
- Device

- **Repositories:**

- ConfigurationRepository
- DeviceRepository



4.2.6.7.2. Bounded Context Database Design Diagram

- **Tablas:**

- **Configuration:**

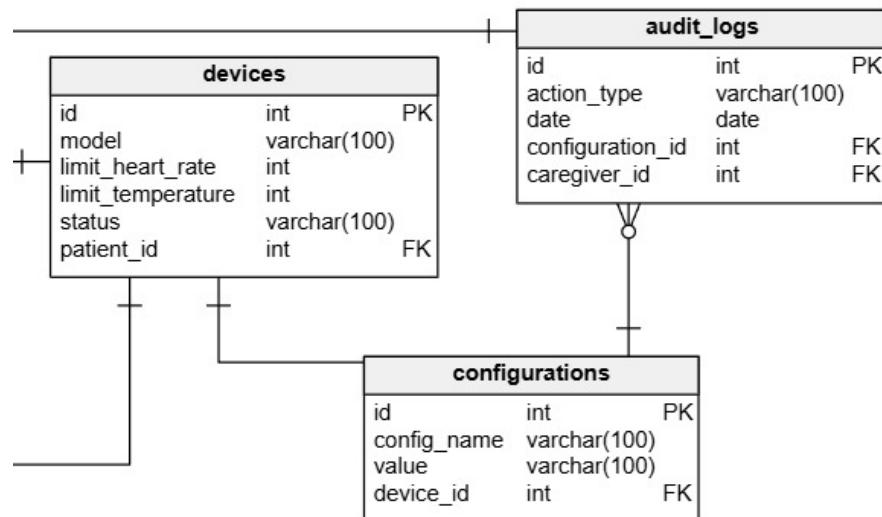
- Columnas: `id`, `configName`, `value`, `Device_id`.
- Llave Foránea: `Device_id`.

- **Device:**

- Columnas: `id`, `model`, `status`, `location`.

- **Relaciones:**

- Configuration se relaciona con Device a través de Device_id.
- Device puede tener varias configuraciones asociadas.



4.2.7. Bounded Context: Account Context

4.2.7.1. Domain Layer

Entities

1. Account:

- **Descripción:** Representa la cuenta de usuario en el sistema, incluyendo información personal, roles y métodos de pago.
- **Atributos:**
 - **id:** Identificador único de la cuenta.
 - **username:** Nombre de usuario asociado a la cuenta.
 - **password:** Contraseña de la cuenta.
 - **email:** Dirección de correo electrónico asociada a la cuenta.
 - **phoneNumber:** Número de teléfono del usuario.
 - **role:** Rol del usuario dentro del sistema.

2. Role:

- **Descripción:** Define los roles de usuario dentro del sistema, como administrador, cuidador, etc.
- **Atributos:**
 - **id:** Identificador único del rol.
 - **roleName:** Nombre del rol (por ejemplo, Admin, Caregiver).
 - **description:** Descripción del rol y sus permisos.

3. Patient:

- **Descripción:** Representa al paciente que es monitoreado mediante la plataforma.
- **Atributos:**
 - **id:** Identificador único del paciente.
 - **name:** Nombre del paciente.
 - **age:** Edad del paciente.
 - **address:** Dirección del paciente.
 - **emergencyContact:** Contacto de emergencia para el paciente.
 - **Account_id:** Referencia a la cuenta a la que pertenece el paciente.

4. Caregiver:

- **Descripción:** Representa a la persona encargada de cuidar al paciente, que tiene acceso a la información de salud del paciente.
- **Atributos:**
 - **id:** Identificador único del cuidador.
 - **name:** Nombre del cuidador.
 - **phone:** Número de teléfono del cuidador.

- **address**: Dirección del cuidador.
- **Account_id**: Referencia a la cuenta a la que pertenece el cuidador.

Value Objects

1. ContactInformation:

- **Descripción:** Objeto de valor que encapsula la información de contacto de un usuario, asegurando que los datos como el correo y el teléfono sean válidos.
- **Atributos:**
 - **email**: Dirección de correo electrónico válida.
 - **phoneNumber**: Número de teléfono válido.

Domain Services

1. AccountDomainService:

- **Descripción:** Contiene la lógica de negocio relacionada con la gestión de cuentas, como la verificación de roles y la actualización de información.
- **Métodos:**
 - **verifyRole(Account account, String role)**: Verifica si una cuenta tiene un rol específico.
 - **updateContactInformation(Account account, ContactInformation contactInfo)**: Actualiza la información de contacto de la cuenta.

Aggregates y Aggregate Root

1. AccountAggregateRoot:

- **Justificación:** **Account** es el aggregate root que controla la consistencia y la lógica de negocio de los usuarios y sus roles. Asegura que todos los roles y datos de contacto relacionados con una cuenta sean válidos y consistentes.

Repositories (Interfaces)

1. AccountRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de cuentas de usuario.
- **Métodos:**
 - **findById(String id)**: Busca una cuenta por su ID.
 - **save(Account account)**: Guarda o actualiza una cuenta en la base de datos.
 - **delete(String id)**: Elimina una cuenta de la base de datos.

2. RoleRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de roles.
- **Métodos:**
 - **findById(String id)**: Busca un rol por su ID.
 - **save(Role role)**: Guarda o actualiza un rol en la base de datos.
 - **delete(String id)**: Elimina un rol de la base de datos.

3. PatientRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de pacientes.
- **Métodos:**
 - **findById(String id)**: Busca un paciente por su ID.
 - **save(Patient patient)**: Guarda o actualiza un paciente en la base de datos.
 - **delete(String id)**: Elimina un paciente de la base de datos.

4. CaregiverRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de cuidadores.
- **Métodos:**
 - **findById(String id)**: Busca un cuidador por su ID.
 - **save(Caregiver caregiver)**: Guarda o actualiza un cuidador en la base de datos.
 - **delete(String id)**: Elimina un cuidador de la base de datos.

4.2.7.2. Interface Layer

Controllers

1. AccountController:

- **Descripción:** Expone endpoints para la gestión de cuentas de usuario, permitiendo registrar, actualizar y consultar cuentas.
- **Métodos:**
 - `registerAccount(Account account)`: Endpoint para registrar una nueva cuenta.
 - `updateAccount(String id, Account account)`: Endpoint para actualizar la información de una cuenta.
 - `getAccountById(String id)`: Endpoint para obtener los detalles de una cuenta por su ID.

2. RoleController:

- **Descripción:** Expone endpoints para la gestión de roles de usuario, permitiendo registrar, actualizar y consultar roles.
- **Métodos:**
 - `createRole(Role role)`: Endpoint para crear un nuevo rol.
 - `updateRole(String id, Role role)`: Endpoint para actualizar la información de un rol.
 - `getRoleById(String id)`: Endpoint para obtener los detalles de un rol por su ID.

3. PatientController:

- **Descripción:** Expone endpoints para la gestión de pacientes, permitiendo registrar, actualizar y consultar pacientes.
- **Métodos:**
 - `registerPatient(Patient patient)`: Endpoint para registrar un nuevo paciente.
 - `updatePatient(String id, Patient patient)`: Endpoint para actualizar la información de un paciente.
 - `getPatientById(String id)`: Endpoint para obtener los detalles de un paciente por su ID.

4. CaregiverController:

- **Descripción:** Expone endpoints para la gestión de cuidadores, permitiendo registrar, actualizar y consultar cuidadores.
- **Métodos:**
 - `registerCaregiver(Caregiver caregiver)`: Endpoint para registrar un nuevo cuidador.
 - `updateCaregiver(String id, Caregiver caregiver)`: Endpoint para actualizar la información de un cuidador.
 - `getCaregiverById(String id)`: Endpoint para obtener los detalles de un cuidador por su ID.

4.2.7.3. Application Layer

Command Handlers

1. CreateAccountHandler:

- **Descripción:** Maneja el comando para registrar una nueva cuenta en el sistema.
- **Métodos:**
 - `handle(CreateAccountCommand command)`: Valida y crea una nueva cuenta en el sistema.

2. CreateRoleHandler:

- **Descripción:** Maneja el comando para crear un nuevo rol en el sistema.
- **Métodos:**
 - `handle(CreateRoleCommand command)`: Valida y crea un nuevo rol en el sistema.

3. RegisterPatientHandler:

- **Descripción:** Maneja el comando para registrar un nuevo paciente en el sistema.
- **Métodos:**
 - `handle(RegisterPatientCommand command)`: Valida y registra un nuevo paciente en el sistema.

4. RegisterCaregiverHandler:

- **Descripción:** Maneja el comando para registrar un nuevo cuidador en el sistema.
- **Métodos:**
 - `handle(RegisterCaregiverCommand command)`: Valida y registra un nuevo cuidador en el sistema.

Event Handlers

1. AccountCreatedEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando se crea una nueva cuenta.
- **Métodos:**
 - `handle(AccountCreatedEvent event)`: Ejecuta las acciones necesarias después de la creación de una cuenta.

2. RoleCreatedEventHandler:

- **Descripción:** Maneja los eventos que se disparan cuando se crea un nuevo rol.
- **Métodos:**
 - `handle(RoleCreatedEvent event)`: Ejecuta las acciones necesarias después de la creación de un rol.

Query Handlers

1. GetAccountDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de una cuenta específica.
- **Métodos:**
 - `handle(GetAccountDetailsQuery query)`: Devuelve los detalles de la cuenta solicitada.

2. GetRoleDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un rol específico.
- **Métodos:**
 - `handle(GetRoleDetailsQuery query)`: Devuelve los detalles del rol solicitado.

3. GetPatientDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un paciente específico.
- **Métodos:**
 - `handle(GetPatientDetailsQuery query)`: Devuelve los detalles del paciente solicitado.

4. GetCaregiverDetailsHandler:

- **Descripción:** Maneja las consultas para obtener los detalles de un cuidador específico.
- **Métodos:**
 - `handle(GetCaregiverDetailsQuery query)`: Devuelve los detalles del cuidador solicitado.

4.2.7.4. Infrastructure Layer

Repositories (Implementaciones)

1. AccountRepositoryImpl:

- **Descripción:** Implementación de `AccountRepository` que define cómo se almacenan y recuperan las cuentas desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar una cuenta por su ID.
 - `save(Account account)`: Implementación para guardar o actualizar una cuenta en la base de datos.
 - `delete(String id)`: Implementación para eliminar una cuenta de la base de datos.

2. RoleRepositoryImpl:

- **Descripción:** Implementación de `RoleRepository` que define cómo se almacenan y recuperan los roles desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un rol por su ID.
 - `save(Role role)`: Implementación para guardar o actualizar un rol en la base de datos.
 - `delete(String id)`: Implementación para eliminar un rol de la base de datos.

3. PatientRepositoryImpl:

- **Descripción:** Implementación de `PatientRepository` que define cómo se almacenan y recuperan los pacientes desde la base de datos.
- **Métodos:**
 - `findById(String id)`: Implementación para buscar un paciente por su ID.
 - `save(Patient patient)`: Implementación para guardar o actualizar un paciente en la base de datos.
 - `delete(String id)`: Implementación para eliminar un paciente de la base de datos.

4. CaregiverRepositoryImpl:

- **Descripción:** Implementación de `CaregiverRepository` que define cómo se almacenan y recuperan los cuidadores desde la base de datos.
- **Métodos:**

- `findById(String id)`: Implementación para buscar un cuidador por su ID.
- `save(Caregiver caregiver)`: Implementación para guardar o actualizar un cuidador en la base de datos.
- `delete(String id)`: Implementación para eliminar un cuidador de la base de datos.

4.2.7.6. Bounded Context Software Architecture Component Level Diagrams

Components (Web API):

1. AccountService Component:

- **Descripción:** Componente encargado de la lógica de negocio relacionada con la gestión de cuentas. Interactúa con el `AccountRepository` para gestionar el registro y la actualización de cuentas.

2. RoleService Component:

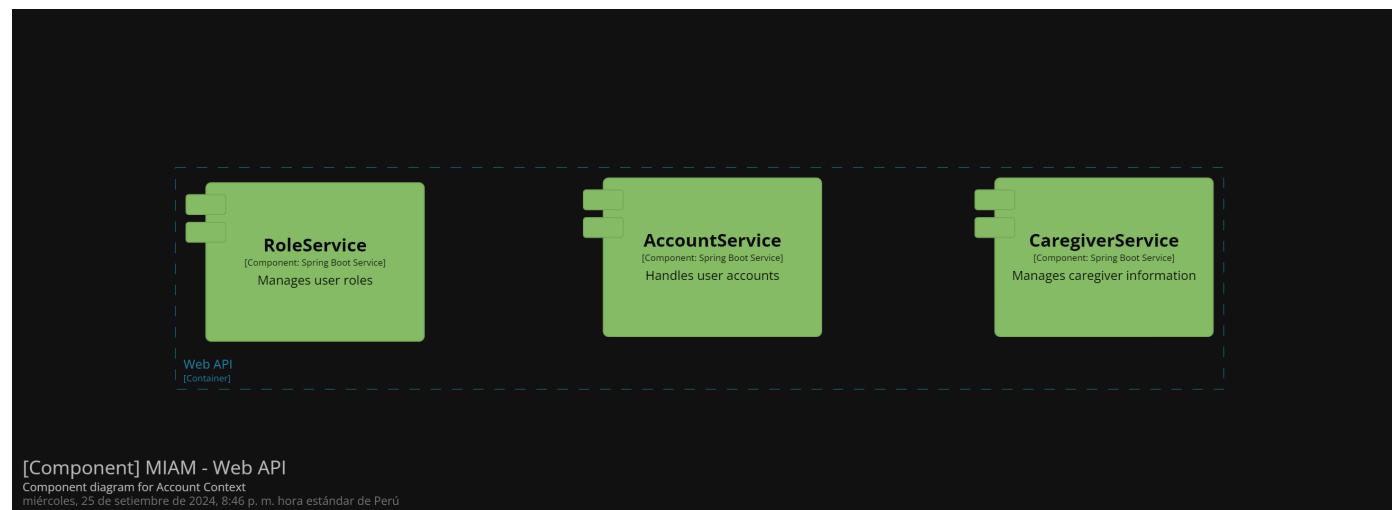
- **Descripción:** Componente encargado de la lógica de negocio relacionada con la gestión de roles. Interactúa con el `RoleRepository` para gestionar el registro y la actualización de roles.

3. PatientService Component:

- **Descripción:** Componente encargado de la lógica de negocio relacionada con la gestión de pacientes. Interactúa con el `PatientRepository` para gestionar el registro y la actualización de pacientes.

4. CaregiverService Component:

- **Descripción:** Componente encargado de la lógica de negocio relacionada con la gestión de cuidadores. Interactúa con el `CaregiverRepository` para gestionar el registro y la actualización de cuidadores.



4.2.7.7. Bounded Context Software Architecture Code Level Diagrams

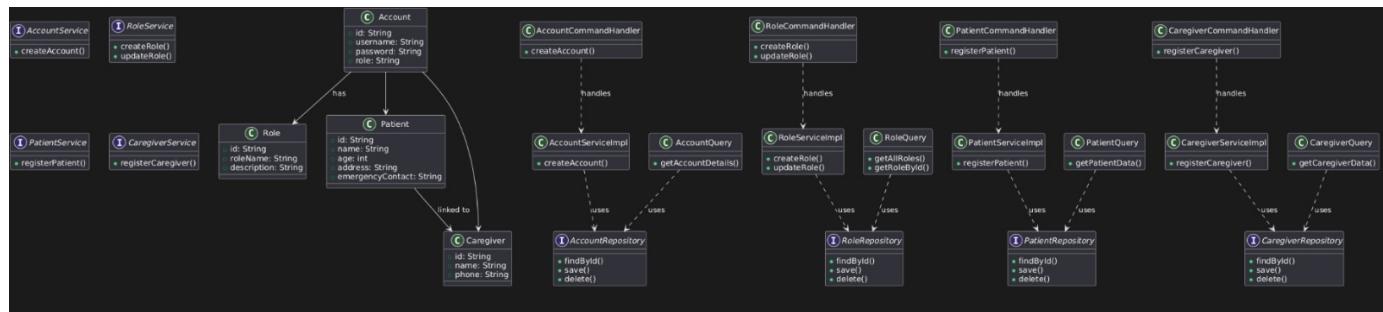
4.2.7.7.1. Bounded Context Domain Layer Class Diagrams

• Clases importantes:

- `Account`
- `Role`
- `Patient`
- `Caregiver`

• Repositories:

- `AccountRepository`
- `RoleRepository`
- `PatientRepository`
- `CaregiverRepository`



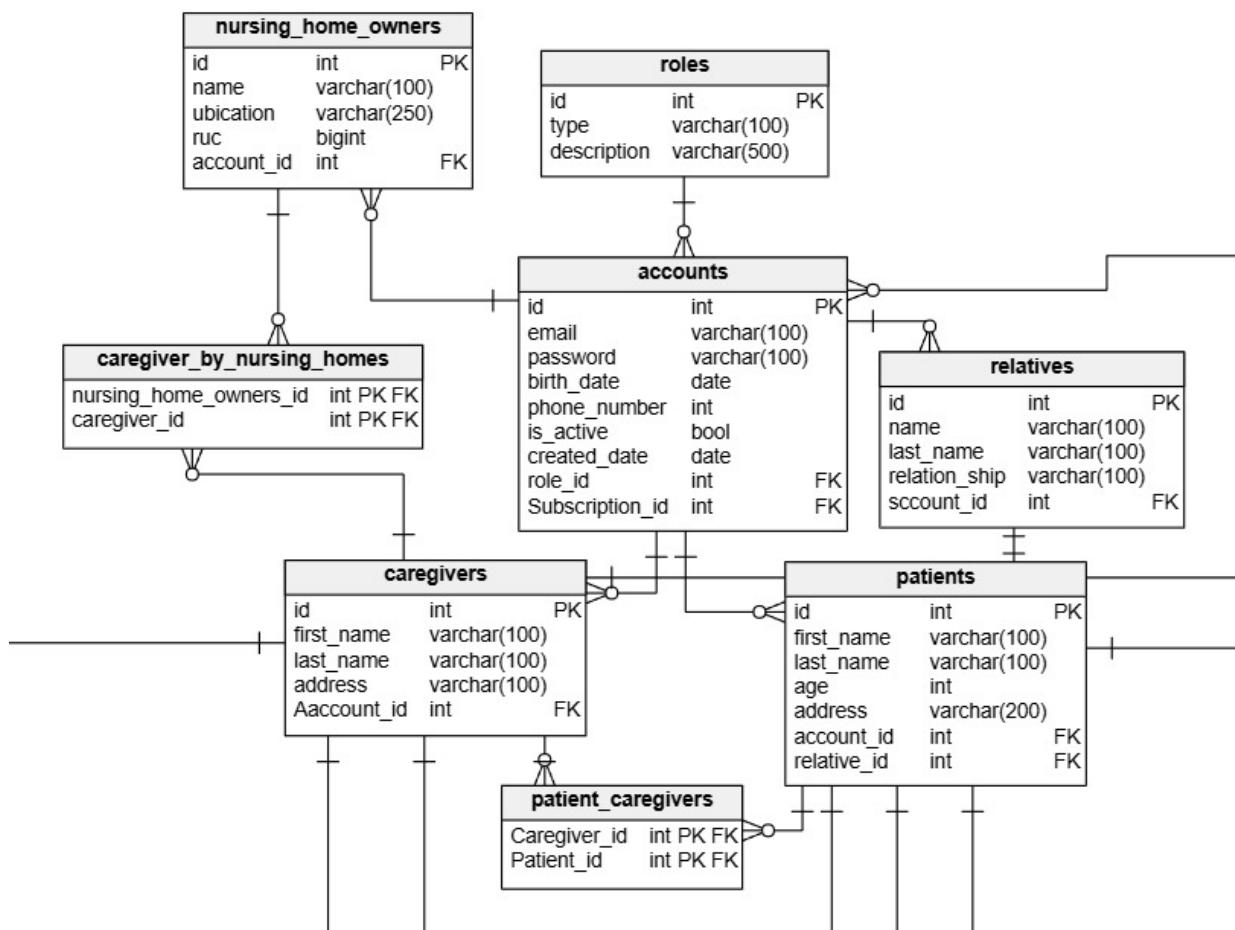
4.2.7.7.2. Bounded Context Database Design Diagram

- **Tablas:**

- **Account:**
 - Columnas: **id**, **username**, **password**, **email**, **phoneNumber**, **role**.
- **Role:**
 - Columnas: **id**, **roleName**, **description**.
- **Patient:**
 - Columnas: **id**, **name**, **age**, **address**, **emergencyContact**, **Account_id**.
 - Llave Foránea: **Account_id**.
- **Caregiver:**
 - Columnas: **id**, **name**, **phone**, **address**, **Account_id**.
 - Llave Foránea: **Account_id**.

- **Relaciones:**

- **Patient** se relaciona con **Account** a través de **Account_id**.
- **Caregiver** se relaciona con **Account** a través de **Account_id**.
- **Account** puede tener múltiples pacientes y cuidadores asociados.



Capítulo V: Solution UI/UX Design

5.1. Strategic-Level Attribute-Driven Design

5.1.1. General Style Guidelines

5.1.1.1. Branding

El sistema de Monitoreo Integral del Adulto Mayor está diseñado para transmitir una sensación de confianza y profesionalismo, fundamental en el contexto de la salud. Se busca un estilo visual que sea accesible y amigable tanto para cuidadores como para usuarios con menos experiencia tecnológica, sin perder el rigor necesario para una aplicación de monitoreo.

El diseño busca reflejar valores como seguridad, precisión, y cuidado. Esto se observa a través de:

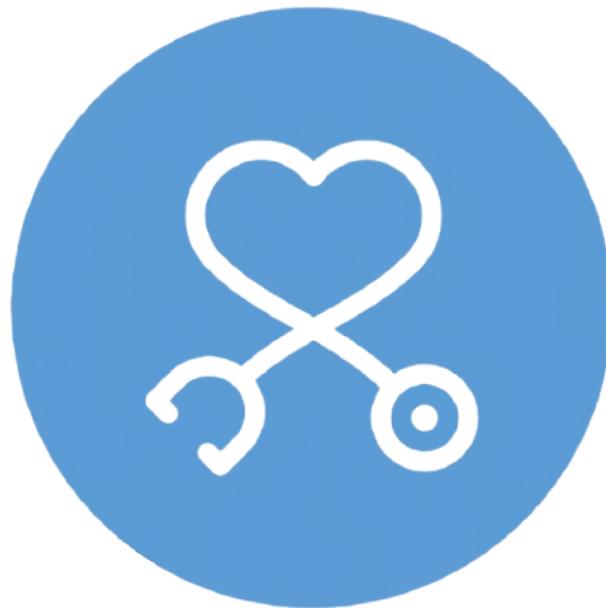
- Paleta de colores: Tonos suaves pero contrastantes, alineados con el contexto de la salud, que generan una sensación de calma y confianza.
- Tipografía moderna: Elegida por su legibilidad clara y estética limpia, contribuyendo a una experiencia visual armónica.
- Diseño minimalista: Enfocado en la funcionalidad, reduciendo la sobrecarga visual para que los usuarios puedan interactuar fácilmente con la interfaz.

Tanto en las aplicaciones como el landing page, se adopta un tono de comunicación respetuoso y sereno, alineado con la naturaleza sensible del cuidado de la salud. El lenguaje es formal, pero sin llegar a ser excesivamente técnico, para que tanto profesionales de la salud como cuidadores y usuarios con menor experiencia tecnológica puedan entender fácilmente la información proporcionada.

5.1.1.2. Logotipo

El logo sigue un diseño minimalista, con líneas limpias y formas simples. El logotipo es fácilmente reconocible y se integra bien en la interfaz sin distraer del contenido principal.

Se presentan dos variaciones principales:



MIAM

La versión a color, utilizada en fondos claros, resalta la identidad de la marca. Es ideal para el uso en el Landing Page, donde el contraste con el fondo es fundamental para captar la atención del usuario.



La versión en blanco, diseñada para fondos oscuros o más complejos. Esta versión asegura que la marca se mantenga presente de manera elegante y sin perder visibilidad. Se utiliza principalmente en situaciones donde el logo necesita integrarse sin competir visualmente con otros elementos, como en el sidebar o áreas más visualmente cargadas.

5.1.1.3. Typography

La tipografía principal utilizada en los encabezados y el cuerpo es Inter, una fuente sans-serif moderna y versátil, reconocida por su excelente legibilidad tanto en dispositivos móviles como en pantallas de alta resolución. Inter se caracteriza por su diseño limpio y espaciado óptimo

Thin

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Extra Light

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Light

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Regular

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Medium

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Semi Bold

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Bold

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Extra Bold

AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

Black

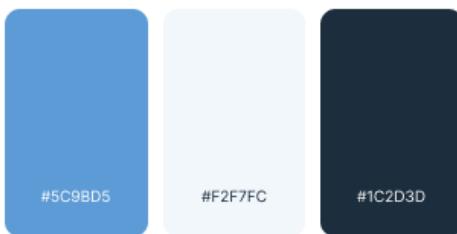
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%
AaBbDdEeGgMmOoRrSs
1234567890 !?()[]{}@\$#%

La estructura tipográfica se organiza mediante variaciones en el tamaño, grosor y estilo de la fuente, estableciendo una jerarquía visual clara que facilita la navegación.

Los títulos se presentan en negrita y con un tamaño mayor, destacándose para captar la atención del usuario de inmediato. En contraste, el cuerpo del texto y las descripciones utilizan tamaños más pequeños y estilos regulares, lo que ofrece una lectura fluida y cómoda en bloques más extensos de información.

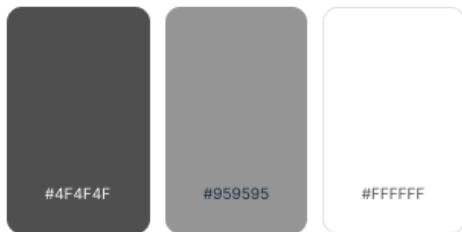
5.1.1.4. Colors

Las aplicaciones web y móvil emplean una paleta cuidadosamente seleccionada que equilibra profesionalismo y tranquilidad, con un enfoque en la accesibilidad y la funcionalidad visual.

Paleta Principal

- #5C9BD5:** Actúa como el color principal, utilizado en títulos y elementos como enlaces o botones de acción secundarios. Este tono de azul está asociado con la tecnología y la seguridad, lo que transmite confianza y estabilidad.
- #F2F7FC:** Se utiliza como color de fondo tanto en las pantallas de la versión web como móvil, creando un entorno limpio que facilita la lectura y mejora la experiencia de usuario.
- #1C2D3D:** Se emplea para proporcionar un contraste oscuro que resalta los íconos y las opciones de navegación.

Colores neutros:



- **#4F4F4F:** Es la elección predominante para todo el texto principal y algunos títulos, asegurando una legibilidad cómoda.
- **#959595:** Se utiliza para textos pequeños o información que no debe competir visualmente con otros elementos.
- **#FFFFFF:** El color blanco se emplea para textos y elementos sobre fondos oscuros.

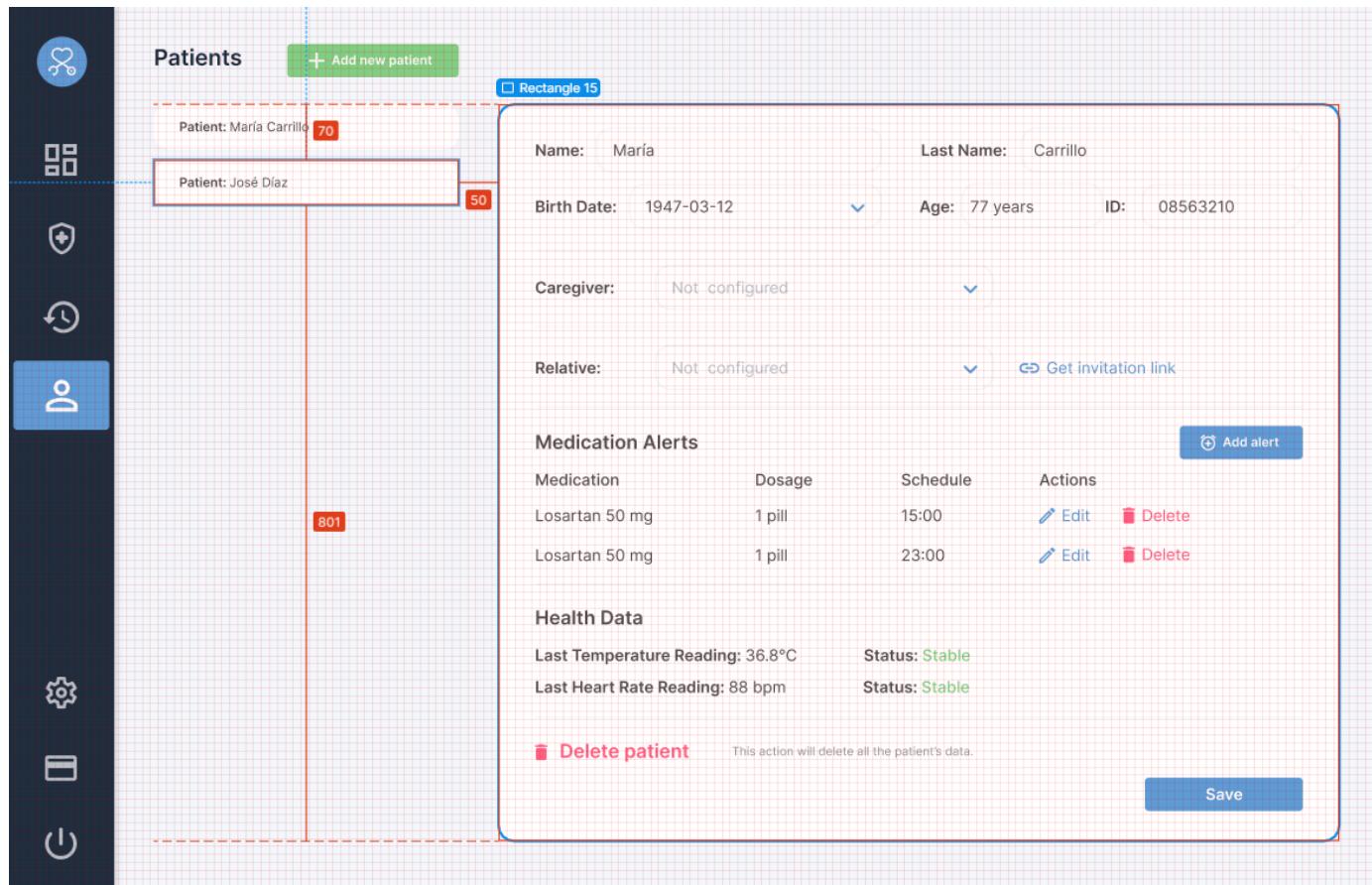
Colores de estado:



- **#80CD83:** Representa estabilidad y situaciones normales, siendo empleado en indicadores de salud, como los signos vitales.
- **#FBBC05:** Indica advertencias o situaciones que requieren precaución.
- **#FE5E81:** Se utiliza para notificar estados críticos o de alto riesgo.

5.1.1.5. Spacing

Se utiliza un espaciado amplio y consistente entre los diferentes elementos de la interfaz, de al menos 10px entre cada elemento. Esto facilita la lectura y la navegación, y evita que la interfaz se vea sobrecargada.



En cuanto a la aplicación móvil, los botones y áreas interactivas son lo suficientemente grandes como para ser utilizados fácilmente en pantallas táctiles.

5.1.2. Web, Mobile and IoT Style Guidelines

5.1.2.1. Web Style Guidelines

Paleta de Colores: Se utilizó la paleta de colores principal para los elementos de la aplicación web. Está compuesta de colores que facilitan la legibilidad y usabilidad.

- **#5C9BD5:** Usado para títulos, enlaces y botones de acción menos importantes.
- **#1C2D3D:** Color del sidebar, proporcionando un contraste fuerte con el fondo.
- **#F2F7FC:** Color de fondo que crea un ambiente ligero y limpio en la interfaz.

Tipografía: Se eligió la fuente Inter. Los títulos se presentan en un tamaño mayor (24px) y un peso más fuerte, mientras que el texto del cuerpo se establece en 16px.

Espaciado y alineación: El diseño mantiene un espaciado consistente entre elementos, con márgenes de al menos 16px para evitar la saturación visual y mejorar la experiencia del usuario.

Botones: Los botones tienen un diseño rectángulo con bordes redondeados, proporcionando una sensación amigable.

Save

Add alert

Add new patient

Gráficos y Visualización de Datos: Los gráficos de signos vitales (temperatura y frecuencia cardíaca) se presentan con líneas claras y colores contrastantes para facilitar su interpretación.

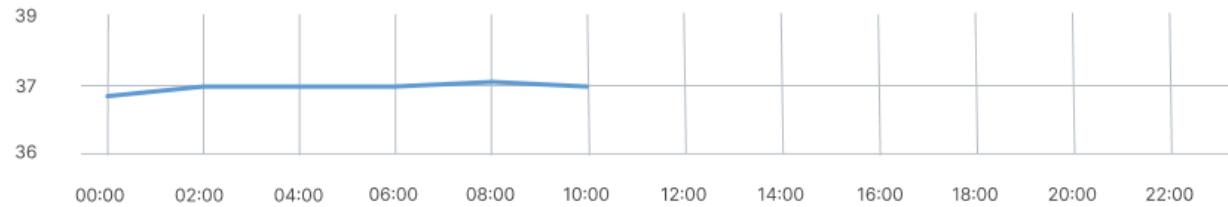
Vital Signs Monitoring

All Patients



September 24h, 2024

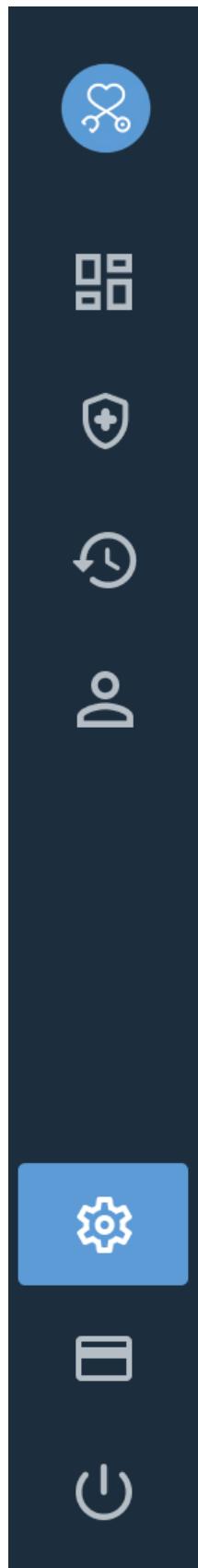
Temperature (°C)



Heart Rate (bpm)



Navegación: La navegación se organiza mediante un menú lateral, permitiendo un acceso fácil a todas las secciones de la aplicación. Las opciones de menú se destacan con iconos intuitivos y etiquetas que mejoran la usabilidad.



5.1.2.2. Mobile Style Guidelines

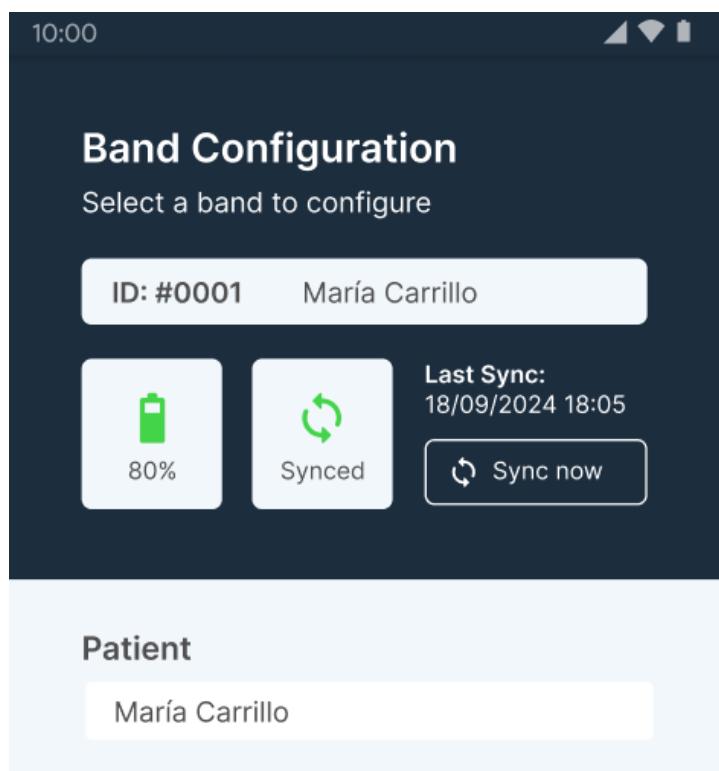
Paleta de Colores: Se utilizó la paleta de colores principal para los elementos de la aplicación web. Está compuesta de colores que facilitan la legibilidad y usabilidad.

- **#5C9BD5:** Usado para títulos, enlaces y botones de acción menos importantes.
- **#1C2D3D:** Color del sidebar, proporcionando un contraste fuerte con el fondo.
- **#F2F7FC:** Color de fondo que crea un ambiente ligero y limpio en la interfaz.

Tipografía: Las fuentes son consistentes con las de la versión web, asegurando que el texto sea claro y fácil de leer en pantallas más pequeñas. Los títulos se presentan en 20px y el texto de cuerpo en 14px.

Espaciado y alineación: El diseño mantiene un espaciado adecuado entre elementos, con un margen mínimo de 12px, evitando la saturación en pantallas reducidas.

Botones: Los botones son amplios y accesibles, con un tamaño mínimo de 44px de alto para facilitar la interacción táctil. Se implementan colores de estado claros que indican interacciones.



Gráficos y Visualización de Datos: Los gráficos de signos vitales se adaptan a la pantalla, utilizando líneas y etiquetas legibles. Se asegura que la información sea fácilmente interpretable en dispositivos móviles.

Navegación: La navegación se coloca en una barra inferior, ofreciendo accesibilidad a las funciones principales de la aplicación. Los iconos son grandes y fáciles de reconocer, permitiendo una navegación intuitiva.



5.2. Information Architecture

5.2.1. Organization Systems.

Se utilizaron los tres sistemas de organización visual: jerárquica, secuencial y matricial. Cada uno de estos enfoques se aplica según la naturaleza de la información o tarea a realizar, asegurando que la navegación y la experiencia del usuario sean claras, eficientes y optimizadas para diferentes contextos.

- Visual Hierarchy:** La organización jerárquica se utiliza principalmente en el dashboard y las pantallas de visualización de información clave, como el monitoreo de signos vitales y las alertas. La finalidad de usar una jerarquía visual en estas áreas es priorizar la información importante para los cuidadores.

The dashboard features a vertical sidebar with icons for heart, grid, shield, clock, user, gear, folder, and power. The main area has four main sections:

- Alerts:** Shows two alerts: "High risk of falling for María Carrillo Immediate attention required" (pink box) and "High temperature recorded for José Diaz Current temperature: 38.5°C" (yellow box).
- Total Patients:** Shows 2 total patients. Below it, the **General Health Status** section includes icons for Temperature (Follow-Up Required), Pulse (All Stable), and Fall Risk (Follow-Up Required).
- Medication:** Shows medication history for María Carrillo (Losartan 50 mg - 1 pill) at 15:00 and José Diaz (Glibenclamide 5 mg - 1/2 pill) at 20:00, both cared for by Carolina Suarez.
- Caregivers:** Shows a dropdown menu set to "Carolina Suarez".
- Vital Signs Monitoring:** Shows temperature and heart rate trends from September 24th, 2024. Temperature remains stable around 37°C. Heart rate fluctuates between 70 and 80 bpm.

- Step-by-Step to Accomplish:** La organización secuencial se aplica en las funciones que requieren la ejecución de tareas que siguen un proceso definido y donde cada etapa depende de la anterior, como es el caso de la sección Billing & Plans.

The sidebar icons include heart, grid, shield, clock, user, gear, folder, and power. The main content shows an upgrade process:

- Upgrade your plan for more features**: Current Plan: Basic. Step 1: Choose (highlighted), Step 2: Customize, Step 3: Payment, Step 4: Confirmation.
- Basic Plan** (includes 1 Caregiver account, Up to 3 Family Member accounts, Up to 3 health bands; designed for individual caregivers). Includes a "CUSTOMIZE" button.
- Enterprise Plan** (includes 1 Administrator account, 1+ Caregiver accounts, 3+ Family Member accounts, Customized health bands; perfect for nursing homes). Includes an "UPGRADE NOW" button.

- Matricial:** Se utilizó este tipo de organización para la vista de tablas en la sección de Alerts y Billing History. Esta proporciona una vista global de los datos que ayuda a manejar la información de manera simultánea.

DATE ↑	HOUR	PATIENT ↓	ALERT TYPE ↓	DESCRIPTION	CAREGIVER ↓	ACTIONS ↓
2024-09-24	09:49 AM	María Carrillo	High risk of falling	Increased risk of falling	Carolina Suarez	Monitored
2024-09-24	09:49 AM	José Díaz	High temperature	Current temperature: 38.5°C	Carolina Suarez	Monitored

5.2.2. Labeling Systems.

- a. Etiquetas Simplificadas y Claras:** El uso de etiquetas simples y claras será clave para facilitar la navegación y comprensión de los usuarios. Algunas etiquetas propuestas:

Home → "Inicio" Our Service → "Nuestros Servicios" Plan → "Planes" Log In → "Iniciar Sesión" Contact → "Contáctanos"

- b. Asociaciones Clave:** Relacionar términos de monitoreo como "seguridad", "alertas", "reporte en tiempo real" con los beneficios principales para que los usuarios los identifiquen rápidamente. Usar un mínimo de palabras en las etiquetas, evitando tecnicismos innecesarios.

5.2.3. SEO Tags and Meta Tags.

- a. SEO para el sitio web:**

Landing Page:

Title: Sistema de Monitoreo Avanzado para Pacientes - Seguridad y Tranquilidad **Meta Description:** Nuestro sistema de monitoreo todo en uno mejora la seguridad y el bienestar de pacientes y seres queridos con monitoreo en tiempo real y alertas personalizadas. **Keywords:** monitoreo de pacientes, sistema de seguridad, alertas personalizadas, cuidado remoto, monitoreo en tiempo real **Author:** Nombre de la startup

Our Service Page:

Title: Nuestros Servicios de Monitoreo - Seguridad 24/7 para Pacientes **Meta Description:** Descubre los servicios que ofrecemos para el monitoreo en tiempo real de pacientes y seres queridos, con alertas automáticas y reportes detallados. **Keywords:** monitoreo de salud, servicios de cuidado remoto, alertas de seguridad, monitoreo de pacientes, bienestar de pacientes.

5.2.4. Navigation Systems.

- El sistema de navegación debe ser claro y directo. Algunas estrategias para guiar a los usuarios:**

Barra de navegación principal (Navbar en la parte superior) con las rutas clave: Inicio, Nuestros Servicios, Planes, Contáctanos, Iniciar Sesión

Breadcrumbs (Migas de pan) para ayudar al usuario a saber dónde está en todo momento dentro del sitio.

Navegación de pie de página: Repetir las rutas clave como "Términos y condiciones", "Política de privacidad", "Soporte", y enlaces a redes sociales.

5.2.5. Navigation Systems.

- La implementación de un sistema de búsqueda sólido ayudará a los usuarios a encontrar información específica sin perderse en la plataforma. Este sistema debe incluir:

Barra de búsqueda visible en la parte superior de todas las páginas.

Resultados claros: Presentar los resultados con un resumen breve que contenga enlaces directos a las páginas correspondientes.

5.2.3. SEO Tags and Meta Tags

5.2.4. Searching Systems.

5.2.5. Navigation Systems.

5.3. Landing Page UI Design

Se diseñó el Landing Page con un enfoque centrado en la experiencia del usuario, donde las decisiones de diseño y la arquitectura de la información se han basado en principios clave como la simplicidad, la accesibilidad y la claridad de comunicación.

Cada componente de la interfaz responde a la necesidad de brindar confianza y facilitar la comprensión del servicio ofrecido, tanto en versiones de escritorio como en dispositivos móviles.

5.3.1. Landing Page Wireframe

El wireframe para el navegador de escritorio presenta una estructura organizada en bloques que prioriza el flujo de información y facilita la exploración de contenido de manera natural. En la parte superior, se encuentra un encabezado con el menú de navegación, manteniendo visibles las secciones más relevantes.

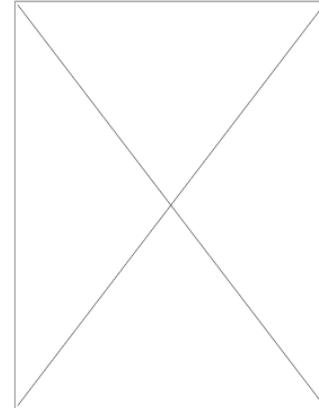
El diseño sigue los principios de la jerarquía visual al posicionar la información más importante en el área superior de la pantalla, lo que mejora la usabilidad. También se aplican principios de diseño inclusivo, con un enfoque en la legibilidad de los textos y la facilidad para navegar por el contenido.

[About](#)[Our Service](#)[Plans](#)[Contact](#)[LOG IN](#)

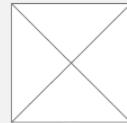
ADVANCED MONITORING FOR SENIORS

POWERED BY THE BEST TECHNOLOGY

An all-in-one monitoring system designed to offer safety and reassurance for your patients and loved ones.

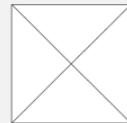
[HOW IT WORKS](#)

How our monitoring system works



Easy setup

Start by setting up the devices. Our system is designed for quick installation.



Configure the band

Customize the monitoring experience by configuring the band. Designate additional caregivers or family members to receive alerts.



Real-Time monitoring

Once installed, the health band begins tracking vital health metrics and movement in real-time.



Immediate Notifications

Receive timely notifications about health irregularities directly on your device.



Health Insights

Access reports and historical data through our web or mobile app.

Why choose us?

Discover the key benefits of our advanced monitoring system designed to enhance the safety and well-being of your patients and loved ones. Here's how we make a difference:



Enhanced Safety

Continuous monitoring ensures early detection of irregularities like abnormal heart rates, movement and high temperatures.



Health Management

Receive timely notifications that allow for prompt action, ensuring better management of health conditions.



Easy-to-Use Interface

Our user-friendly app and intuitive dashboard make it simple to set up, monitor, and manage alerts.



Family Integration

Caregivers can link the bands to family members, allowing them to receive real-time alerts.



Emergency Button

Each band includes an emergency panic button, enabling patients to quickly call for help if they need immediate assistance.

Plans

Choose the plan that best fits your needs and start enhancing the safety and well-being of those you care about. Whether you're an individual caregiver or managing a nursing home, we have a solution tailored for you.

Basic

Includes:

1 Caregiver account
Up to 3 Family Member accounts
Up to 3 health bands

Designed for individual caregivers who need to monitor up to three elderly individuals.

[GET PLAN](#)

Enterprise

Includes:

1 Administrator account
1+ Caregiver accounts
3+ Family Member accounts
Customized health bands

Perfect for nursing homes and larger care facilities. This plan offers advanced features tailored to your specific needs.

[GET PLAN](#)

What our clients are saying

"A Game-Changer for Our Facility"
"Since implementing the Enterprise Plan, our facility has seen a significant improvement in both caregiver efficiency and resident satisfaction. The ability to monitor health metrics in real-time and keep family members informed has made a world of difference. The customized health bands are especially useful and comfortable for our residents."
— Maria González, Director of Sunshine Nursing Home

Meet the team
behind your care solution

At SocialTech, we are passionate about innovative technology. Our team of dedicated professionals combines extensive experience in healthcare, technology, and customer support to deliver a solution that truly makes a difference.

Jean Achamizo Huamani
Lead Developer

Lucía Aliaga Trevejo
Front-End Developer

Rodrigo Raymundo Guevara
DevOps Engineer

Luis Siancas Reategui
Back-End Developer

Luis Trujillo Lopez
Full-Stack Developer

Contact

We're here to help you with any questions or support you need. Feel free to reach out to us through the following channels:

General Inquiries
Email: info@socialtech.com
Phone: (123) 456-7890

Sales
Email: sales@socialtech.com
Phone: (123) 456-7890

Customer Support
Email: support@socialtech.com
Phone: (123) 456-7890

We look forward to hearing from you and assisting you with any needs or inquiries!

 MIAM

Info
How it works
Why choose us
Plans
Testimonials

SocialTech
About us
Contact

Follow us
  

El mock-up para la versión de escritorio del Landing Page refleja la aplicación concreta de los principios de diseño establecidos, como la consistencia visual, con el uso de una paleta de colores neutros y azulados que generan una atmósfera de confianza y tranquilidad. Se han empleado tipografías legibles y contrastes adecuados para mejorar la accesibilidad, siguiendo las pautas de diseño inclusivo.

Además, el Hero Section incluye el Call to action "How it Works", que invita a los usuarios a conocer más el sistema.

The screenshot shows a desktop landing page for a senior monitoring system. At the top, there is a navigation bar with a logo (a blue circle with a heart and a person icon), the word "MIAM", and links for "About", "Our Service", "Plans", "Contact", and a green "LOG IN" button. Below the navigation bar, the main headline reads "ADVANCED MONITORING FOR SENIORS" and "POWERED BY THE BEST TECHNOLOGY". A subtext below the headline states: "An all-in-one monitoring system designed to offer safety and reassurance for your patients and loved ones." To the right of the text is an image of a light blue fitness tracker with a woven strap. Below the headline is a blue button labeled "HOW IT WORKS". In the center of the page, under the heading "How our monitoring system works", there is a horizontal flow of five circular icons connected by lines, each representing a step in the process: 1. Easy setup (showing a red device). 2. Configure the band (showing a blue band). 3. Real-Time monitoring (showing a hand wearing a blue band). 4. Immediate Notifications (showing a smartphone screen with a notification). 5. Health Insights (showing a line graph titled "Sleep Monitoring"). Each icon has a corresponding title and description below it.

ADVANCED MONITORING FOR SENIORS

POWERED BY THE BEST TECHNOLOGY

An all-in-one monitoring system designed to offer safety and reassurance for your patients and loved ones.

HOW IT WORKS

How our monitoring system works

- Easy setup**
Start by setting up the devices. Our system is designed for quick installation.
- Configure the band**
Customize the monitoring experience by configuring the band. Designate additional caregivers or family members to receive alerts.
- Real-Time monitoring**
Once installed, the health band begins tracking vital health metrics and movement in real-time.
- Immediate Notifications**
Receive timely notifications about health irregularities directly on your device.
- Health Insights**
Access reports and historical data through our web or mobile app.

Why choose us?

Discover the key benefits of our advanced monitoring system designed to enhance the safety and well-being of your patients and loved ones. Here's how we make a difference:



Enhanced Safety

Continuous monitoring ensures early detection of irregularities like abnormal heart rates, movement and high temperatures.



Health Management

Receive timely notifications that allow for prompt action, ensuring better management of health conditions.



Easy-to-Use Interface

Our user-friendly app and intuitive dashboard make it simple to set up, monitor, and manage alerts.



Family Integration

Caregivers can link the bands to family members, allowing them to receive real-time alerts.



Emergency Button

Each band includes an emergency panic button, enabling patients to quickly call for help if they need immediate assistance.

Plans

Choose the plan that best fits your needs and start enhancing the safety and well-being of those you care about. Whether you're an individual caregiver or managing a nursing home, we have a solution tailored for you.

Basic

Includes:

1 Caregiver account
Up to 3 Family Member accounts
Up to 3 health bands

Designed for individual caregivers who need to monitor up to three elderly individuals.

[GET PLAN](#)

Enterprise

Includes:

1 Administrator account
1+ Caregiver accounts
3+ Family Member accounts
Customized health bands

Perfect for nursing homes and larger care facilities. This plan offers advanced features tailored to your specific needs.

[GET PLAN](#)

What our clients are saying



"A Game-Changer for Our Facility"

"Since implementing the Enterprise Plan, our facility has seen a significant improvement in both caregiver efficiency and resident satisfaction. The ability to monitor health metrics in real-time and keep family members informed has made a world of difference. The customized health bands are especially useful and comfortable for our residents."

— Maria González, Director of Sunshine Nursing Home

< >

Meet the team

behind your care solution

At SocialTech, we are passionate about innovative technology. Our team of dedicated professionals combines extensive experience in healthcare, technology, and customer support to deliver a solution that truly makes a difference.



Jean Achamizo Huamani
Lead Developer



Lucía Aliaga Trevejo
Front-End Developer



Rodrigo Raymundo Guevara
DevOps Engineer



Luis Siancas Reategui
Back-End Developer



Luis Trujillo Lopez
Full-Stack Developer

Contact

We're here to help you with any questions or support you need. Feel free to reach out to us through the following channels:

General Inquiries
Email: info@socialtech.com
Phone: (123) 456-7890

Sales
Email: sales@socialtech.com
Phone: (123) 456-7890

Customer Support
Email: support@socialtech.com
Phone: (123) 456-7890

We look forward to hearing from you and assisting you with any needs or inquiries!



Info

How it works
Why choose us
Plans
Testimonials

SocialTech

About us
Contact

Follow us



5.4. Applications UX/UI Design

5.4.1. Applications Wireframes

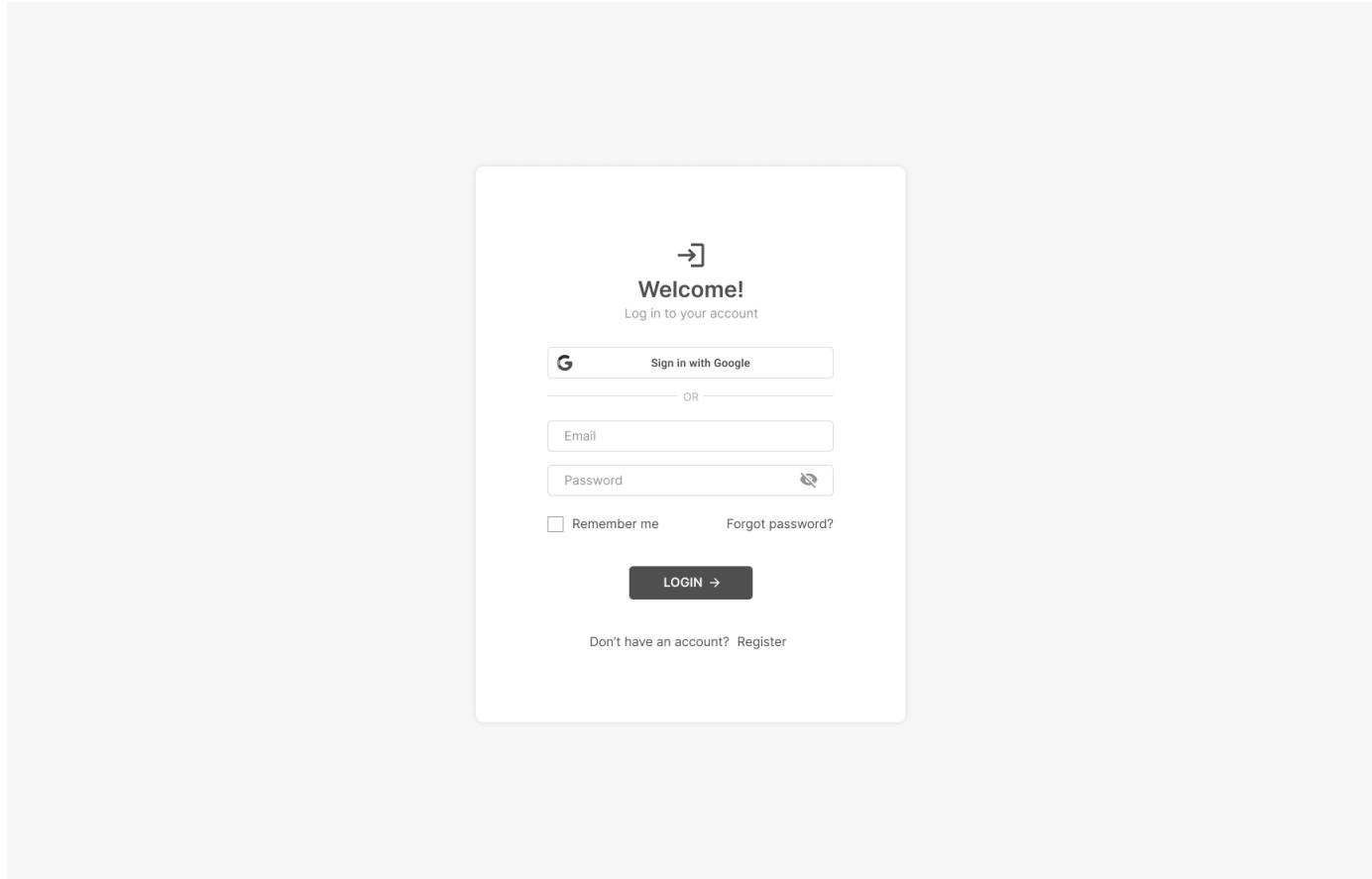
5.4.1.1. Web Application Wireframes

Los wireframes sirven como una representación inicial del diseño. En ellos se ha aplicado cuidadosamente la arquitectura de información, asegurando que los elementos clave se ubiquen de manera lógica y accesible para los usuarios.

Los principios de jerarquía visual y alineación sirven para guiar al usuario a través de la interfaz de forma intuitiva, destacando las acciones más importantes y asegurando una fácil navegación entre las diferentes secciones.

A continuación, se muestran los wireframes de las secciones más importantes de la aplicación:

- Log in



- Dashboard

Alerts

- High risk of falling for María Carrillo
Immediate attention required
- High temperature recorded for José Diaz
Current temperature: 38.5°C

Total Patients

2

Medication

15:00	Maria Carrillo Losartan 50 mg - 1 pill	Caregiver: Carolina Suarez
20:00	José Diaz Glibenclamide 5 mg - 1/2 pill	Caregiver: Carolina Suarez

Caregivers

Carolina Suarez

Vital Signs Monitoring

September 24h, 2024

Temperature (°C)

Heart Rate (bpm)

Alerts

- High risk of falling for María Carrillo
Immediate attention required
- High temperature recorded for José Diaz
Current temperature: 38.5°C

Total Patients

2

Medication

15:00	Maria Carrillo Losartan 50 mg - 1 pill	Caregiver: Carolina Suarez
20:00	José Diaz Glibenclamide 5 mg - 1/2 pill	Caregiver: Carolina Suarez

Caregivers

Carolina Suarez

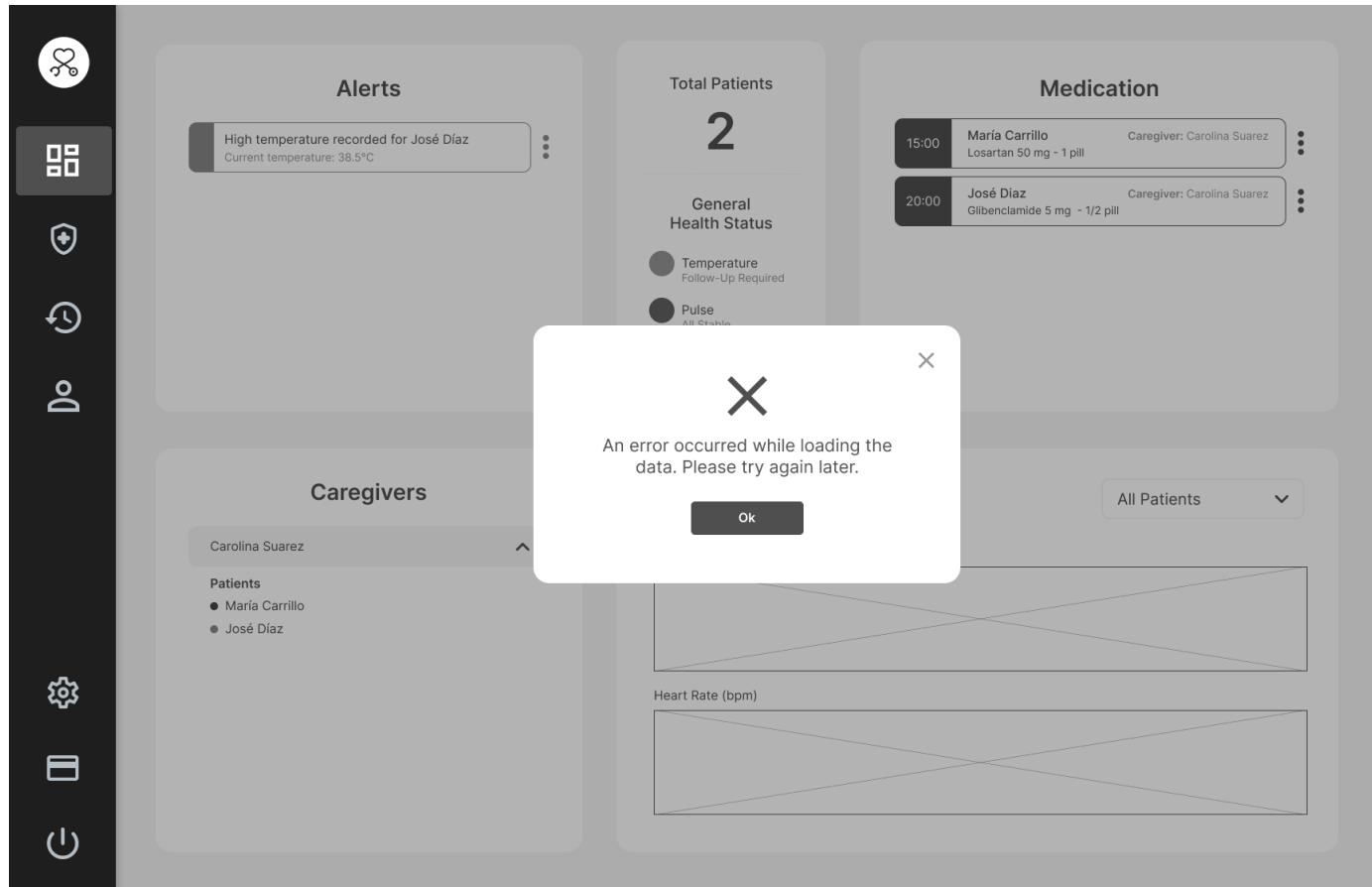
Vital Signs Monitoring

September 24, 2024

Temperature (°C)

Maria Carrillo
José Diaz

Heart Rate (bpm)



- **Band configuration**

The screenshot shows the 'Band Configuration' screen. On the left is a vertical navigation bar with icons for heart rate, square grid, shield (selected), circular arrow, person, gear, card, and power. The main area displays three entries:

- ID: #0001 Patient: María Carrillo
- ID: #0002 Patient: José Díaz
- ID: #0003 Patient: Not configured

The screenshot shows the 'Band Configuration' screen for band ID #0003. The left navigation bar is identical to the previous screenshot. The main area is divided into sections:

- Band ID: #0003**, **Battery: 80%**, **Status: Synchronized**, **Last Sync: 2024-09-24 10:12**, **Sync now**
- Patient**: Assigned to: Elena Farfán
- Caregiver**: Assigned to: Not configured
- Alerts**:
 - Alert if temperature is below 35 °C or higher than 38 °C
 - Alert if pulse is below 60 bpm or higher than 100 bpm
 - Alert me if fall risk is high
- Relative**: Related to: Not configured
- Alerts**:
 - Alert if temperature is below 35 °C or higher than 38 °C
 - Alert if pulse is below 60 bpm or higher than 100 bpm
 - Alert me if fall risk is high
- Reset band**: Resetting this band will unlink all patient data.

A 'Save' button is located at the bottom right.

- **Alerts**

Alerts History						
DATE ↑	HOUR	PATIENT ▾	ALERT TYPE ▾	DESCRIPTION	CAREGIVER ▾	ACTIONS ▾
2024-09-24	09:49 AM	Maria Carrillo	High risk of falling	Increased risk of falling	Carolina Suarez	Monitored
2024-09-24	09:49 AM	José Díaz	High temperature	Current temperature: 38.5°C	Carolina Suarez	Monitored

Generate Report

Select a time period to generate the report

Last week ▾

Cancel Download

- Patients

The screenshot shows a medical software interface. On the left is a dark sidebar with icons for heart, squares, shield, clock, user, gear, card, and power. The main area has a header "Patients" and a button "+ Add new patient". On the left, there's a list of patients: "Patient: María Carrillo", "Patient: José Díaz", and "Patient: New patient". On the right, there's a form to add a new patient: "Name: Patient's name", "Last Name: Patient's last name", "Birth Date: Select date", "ID: ID number", and a "Create" button.

The screenshot shows the same medical software interface after adding a new patient. The patient list now includes "Patient: María Carrillo", "Patient: José Díaz", and "Patient: Elena Farfán". The new patient form is filled out: "Name: Elena", "Last Name: Farfán", "Birth Date: 1947-03-12", "Age: 77 years", "ID: 08563217". Under "Caregiver", it says "Not configured" with a dropdown menu showing "Carolina Suarez". Under "Relative", it says "Elena Torres". There's a link "Get invitation link" next to the relative field. Below this is a section for "Medication Alerts" with a "Add alert" button. A message "No alerts" is displayed. At the bottom is a "Health Data" section showing "Last Temperature Reading: No data" and "Status: No data", and "Last Heart Rate Reading: No data" and "Status: No data". A "Delete patient" button with a warning message "This action will delete all the patient's data." and a "Save" button are at the bottom.

- **Billing**

Billing & Plans

Current Plan:
Basic

Nº Health Bands: 3
Price: \$ 60.00
Renewal: October 18, 2024

Features:

- 1 Caregiver account
- Up to 3 Family Member accounts
- Up to 3 health bands

[CUSTOMIZE](#)

Billing History

# INVOICE	DATE	AMOUNT	STATUS	PAYMENT METHOD
#00002	2024-09-16	\$60.00	Paid	PayPal
#00001	2024-08-15	\$60.00	Paid	PayPal

More features. Better experience!

Upgrade to Enterprise and unlock:

- 1 Administrator account
- 1+ Caregiver accounts
- 3+ Family Member accounts
- Customized health bands

[UPGRADE NOW](#)

Upgrade your plan for more features

Current Plan: Basic

1 Choose 2 Customize 3 Payment 4 Confirmation

Basic

Includes:

1 Caregiver account
Up to 3 Family Member accounts
Up to 3 health bands

Designed for individual caregivers who need to monitor up to three elderly individuals.

[CUSTOMIZE](#)

Enterprise

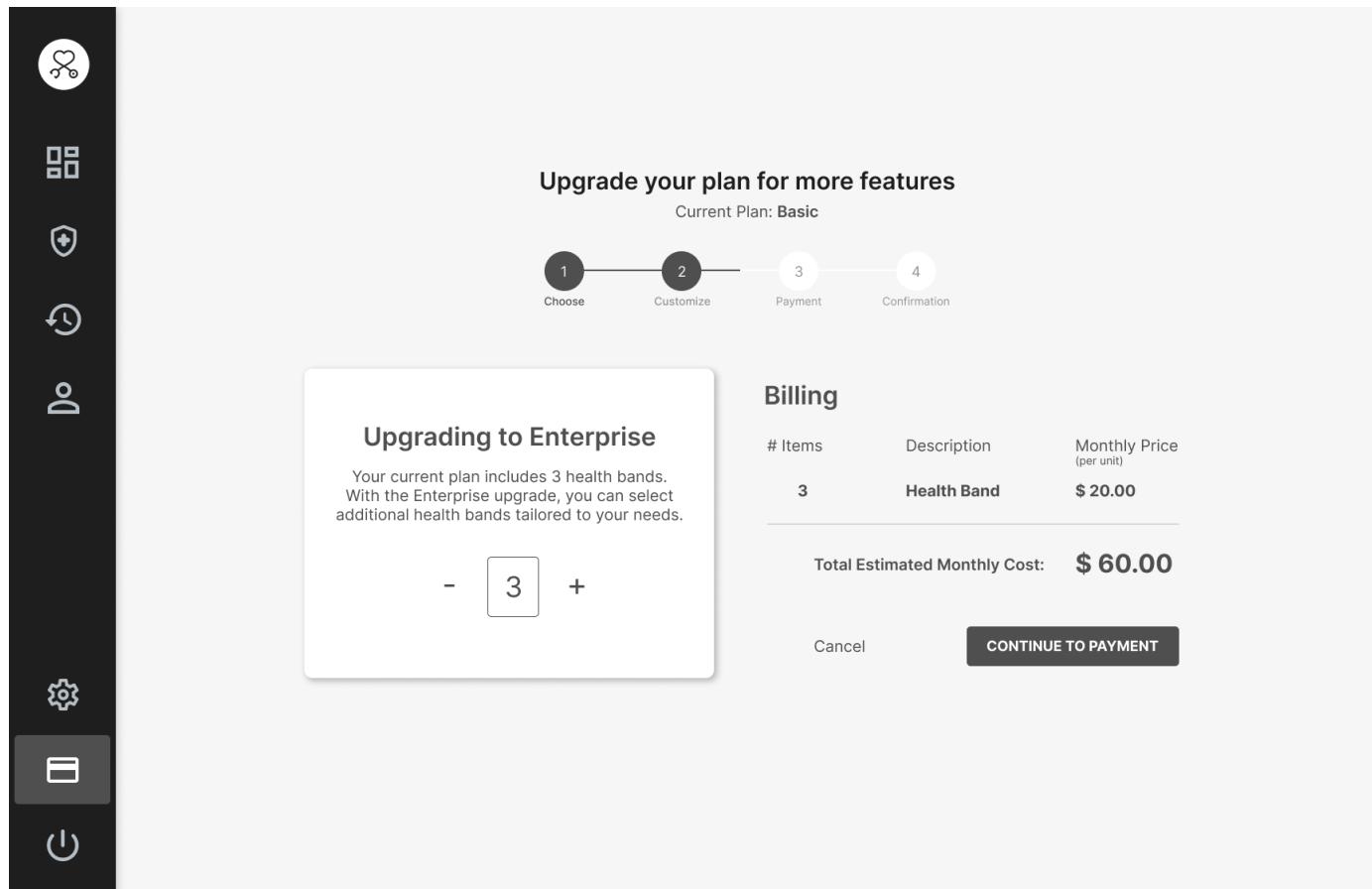
MOST POPULAR

Includes:

1 Administrator account
1+ Caregiver accounts
3+ Family Member accounts
Customized health bands

Perfect for nursing homes and larger care facilities. This plan offers advanced features tailored to your specific needs.

[UPGRADE NOW](#)



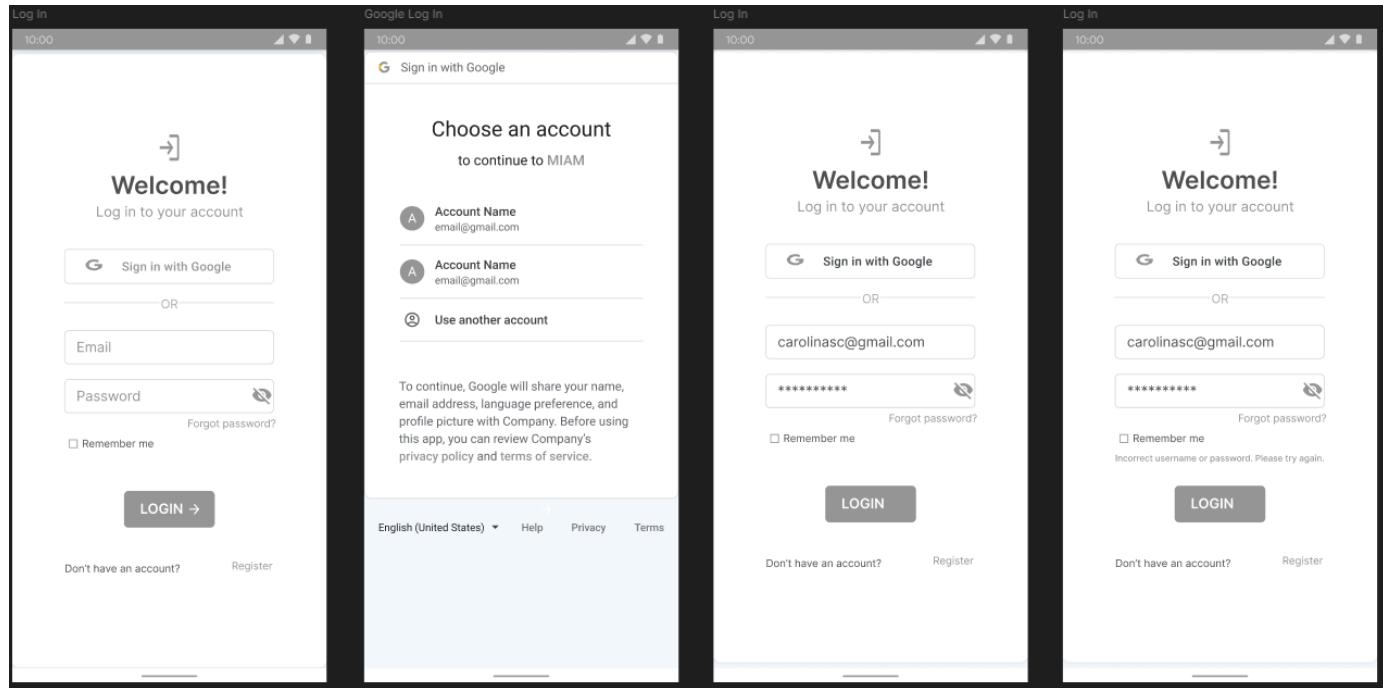
5.4.1.2. Mobile Application Wireframes

Los wireframes móviles representan una versión preliminar del diseño de la aplicación, mostrando cómo se organizará la información en una pantalla más pequeña. Se ha aplicado de manera cuidadosa la arquitectura de información para garantizar que los elementos clave estén ubicados de forma lógica y accesible para los usuarios en sus dispositivos móviles.

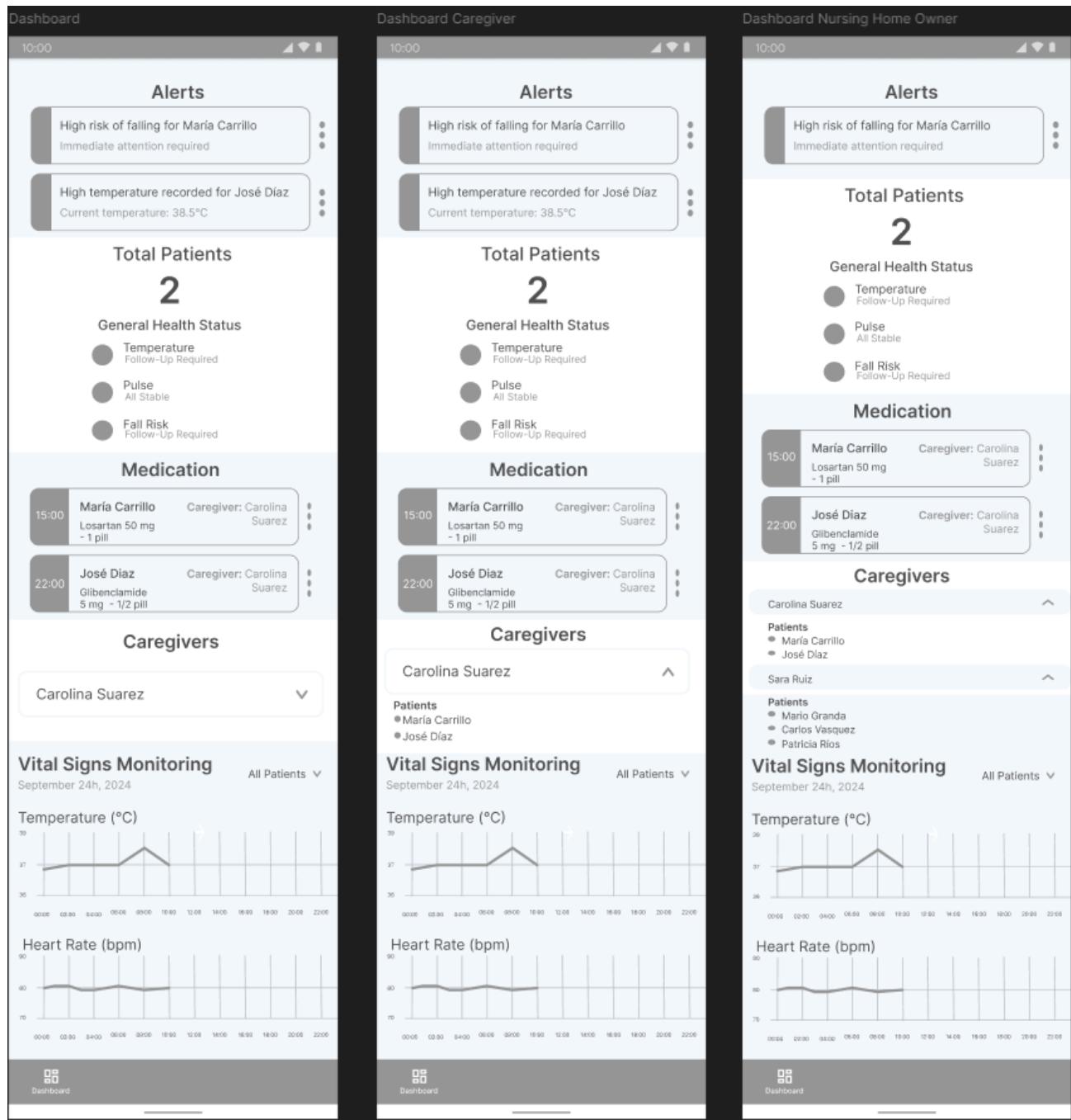
Se han utilizado principios de jerarquía visual y alineación para guiar al usuario a través de la interfaz de forma intuitiva, destacando las acciones más importantes y asegurando una navegación fluida entre las diferentes secciones de la aplicación. El diseño responsive asegura que la experiencia del usuario sea consistente, fácil de usar y sin comprometer la funcionalidad en dispositivos móviles.

A continuación, se presentan los wireframes de las secciones más importantes de la aplicación móvil, mostrando la disposición de los menús, botones de acción y secciones de contenido en un formato optimizado para pantallas de smartphones:

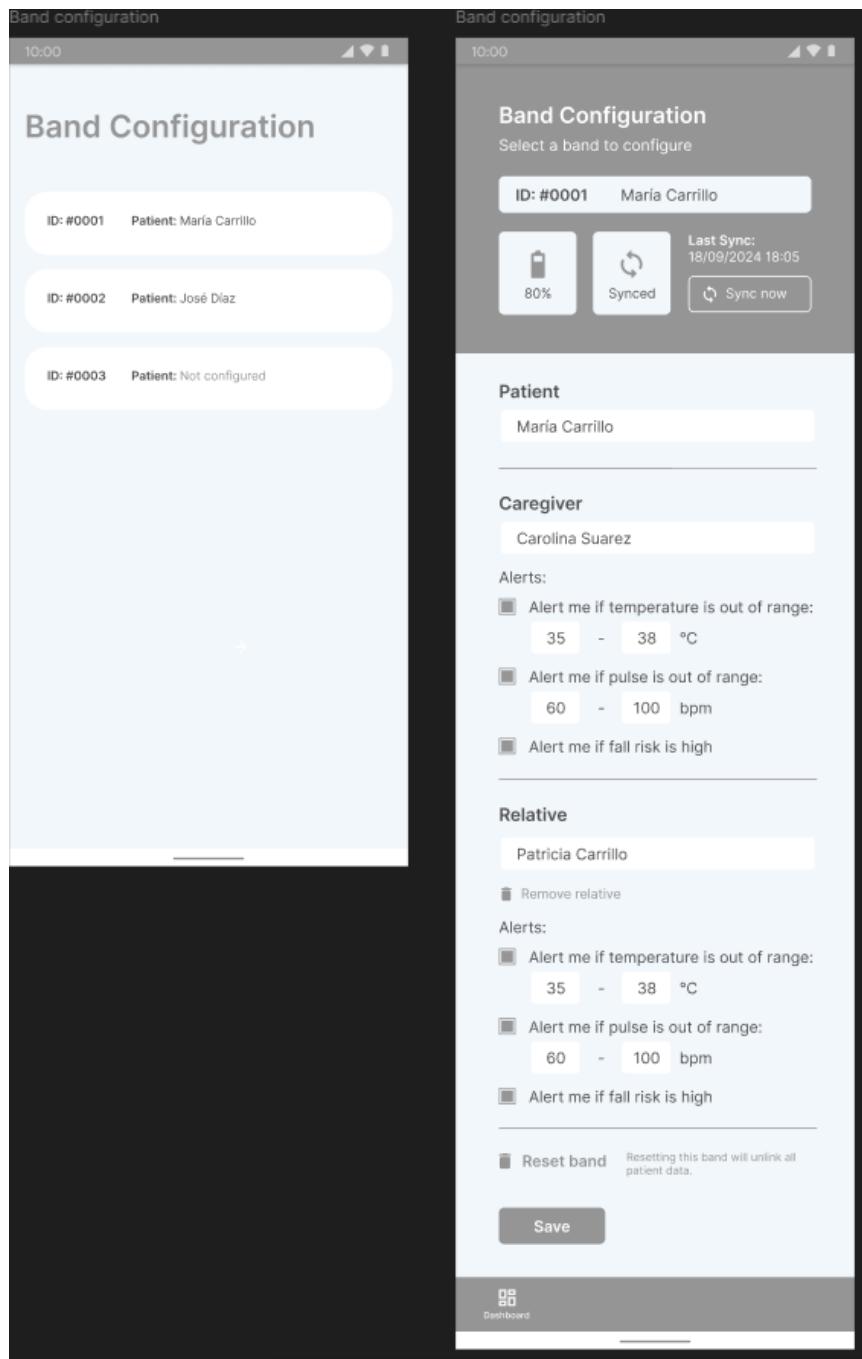
- Log in



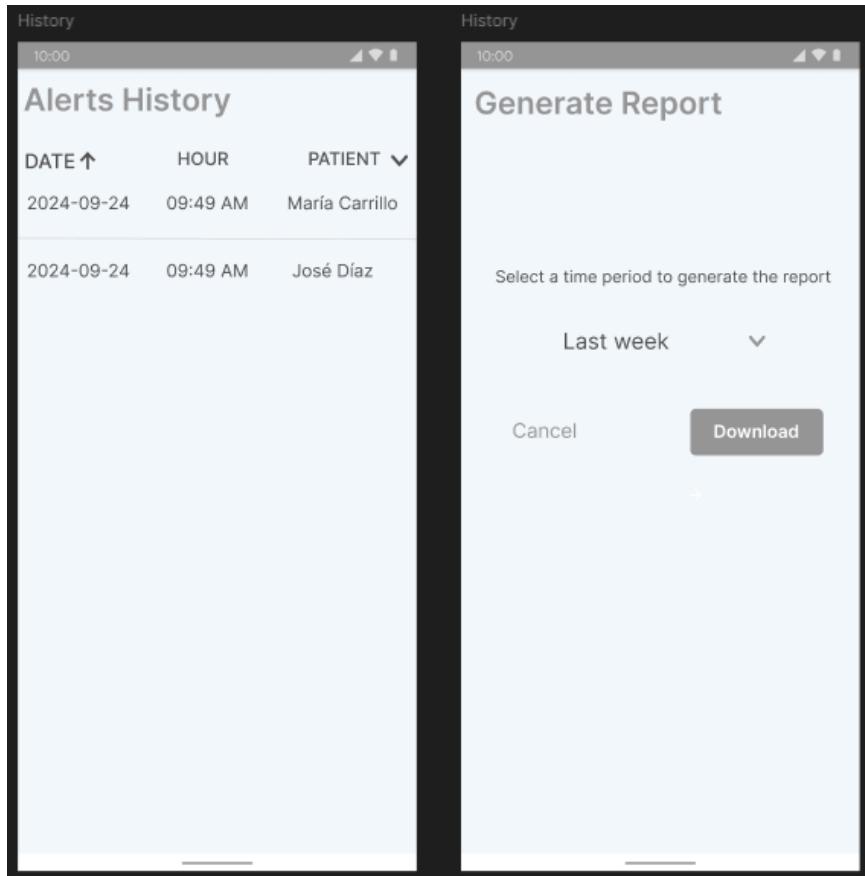
- **Dashboard**



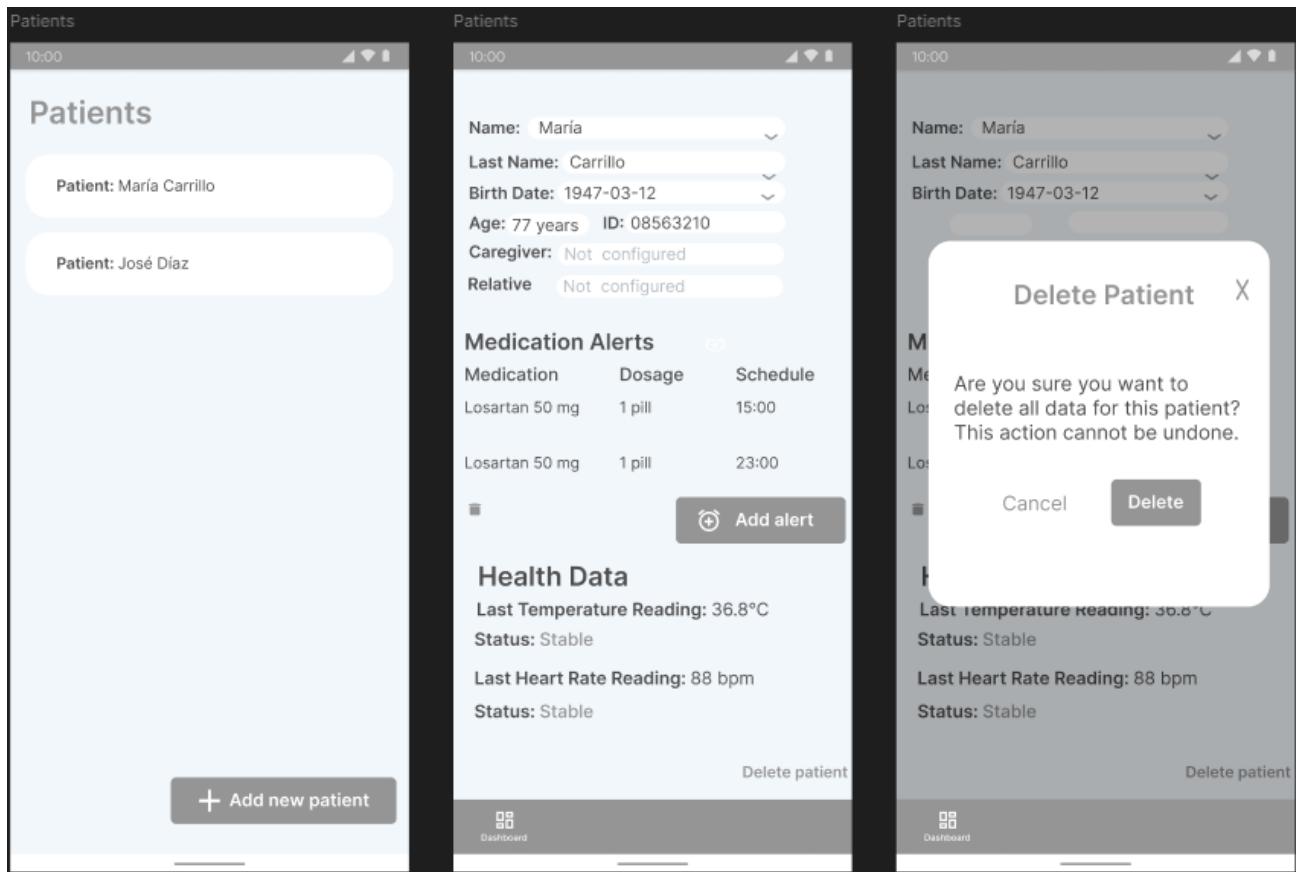
- **Band configuration**



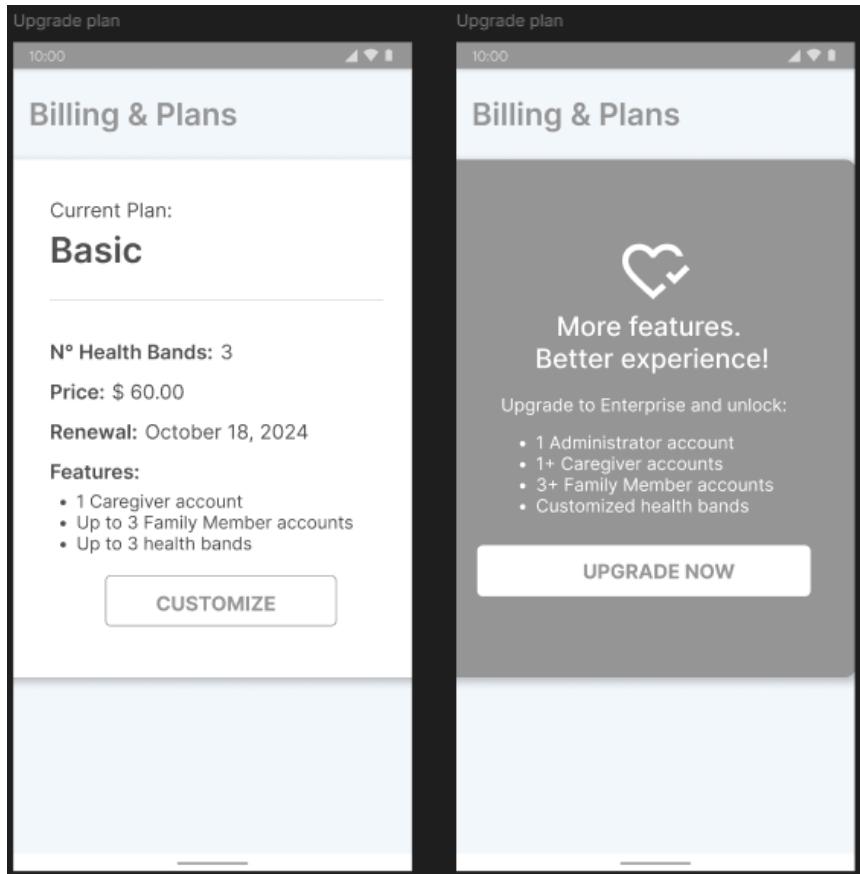
- **Alerts**

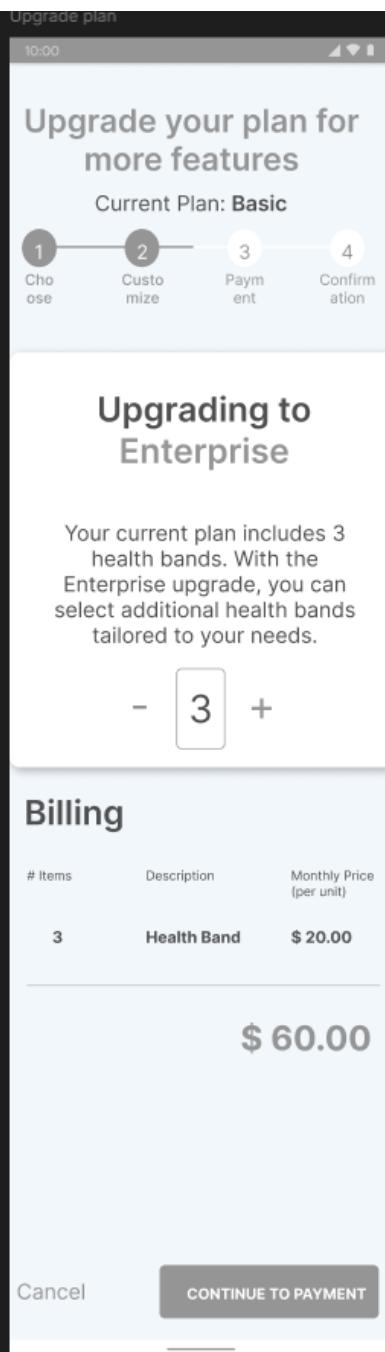
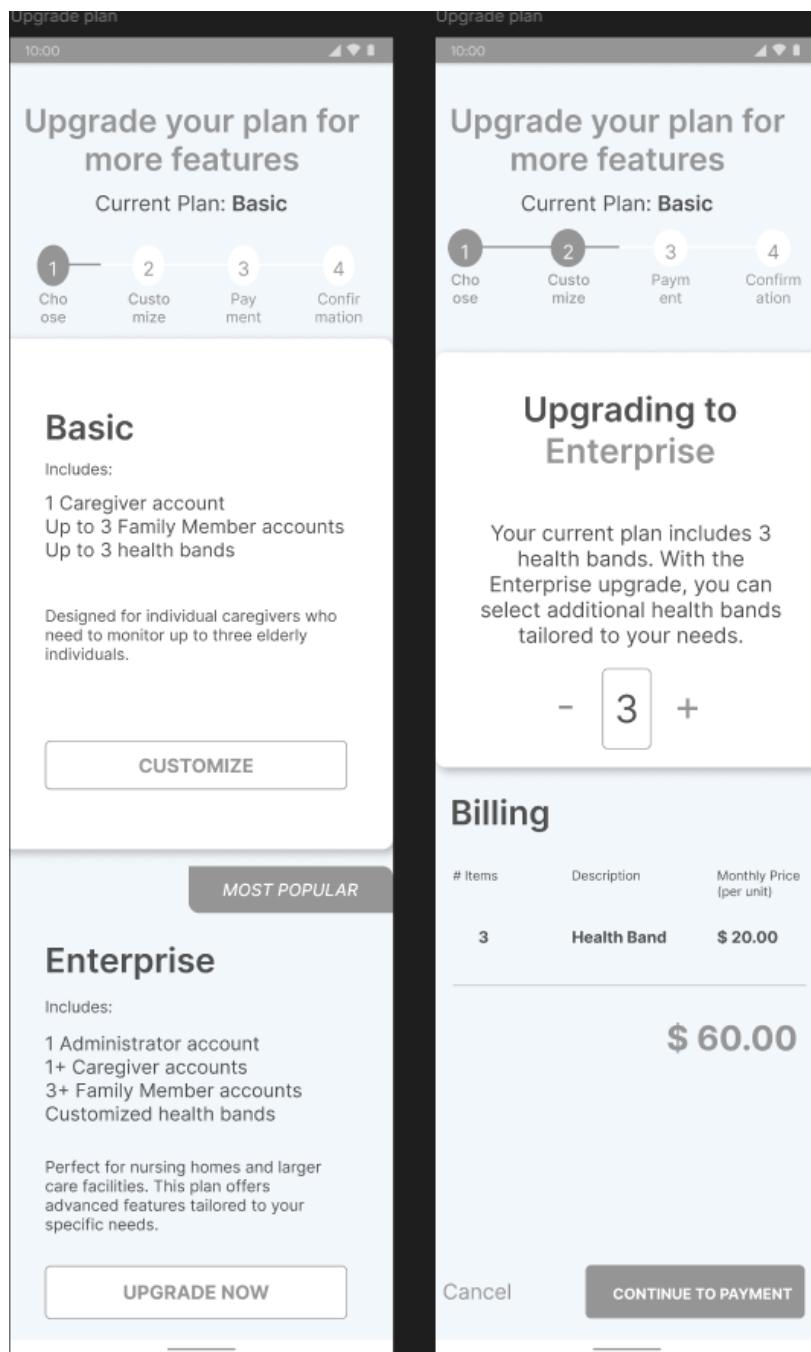


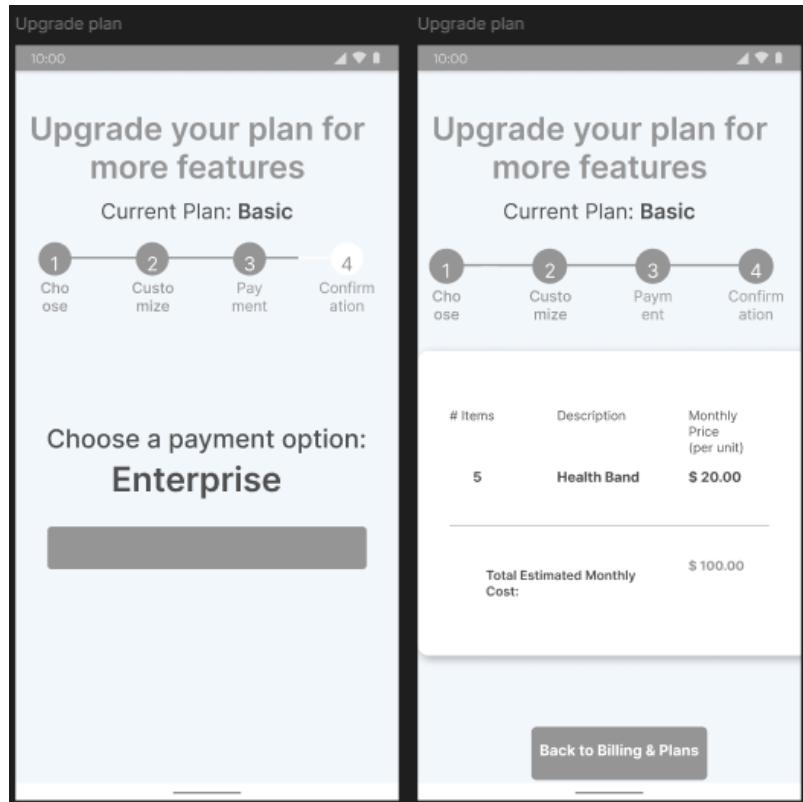
- Patients



- Billing







5.4.2. Applications Wireflow Diagrams

5.4.2.1. Web Application Wireflow Diagrams

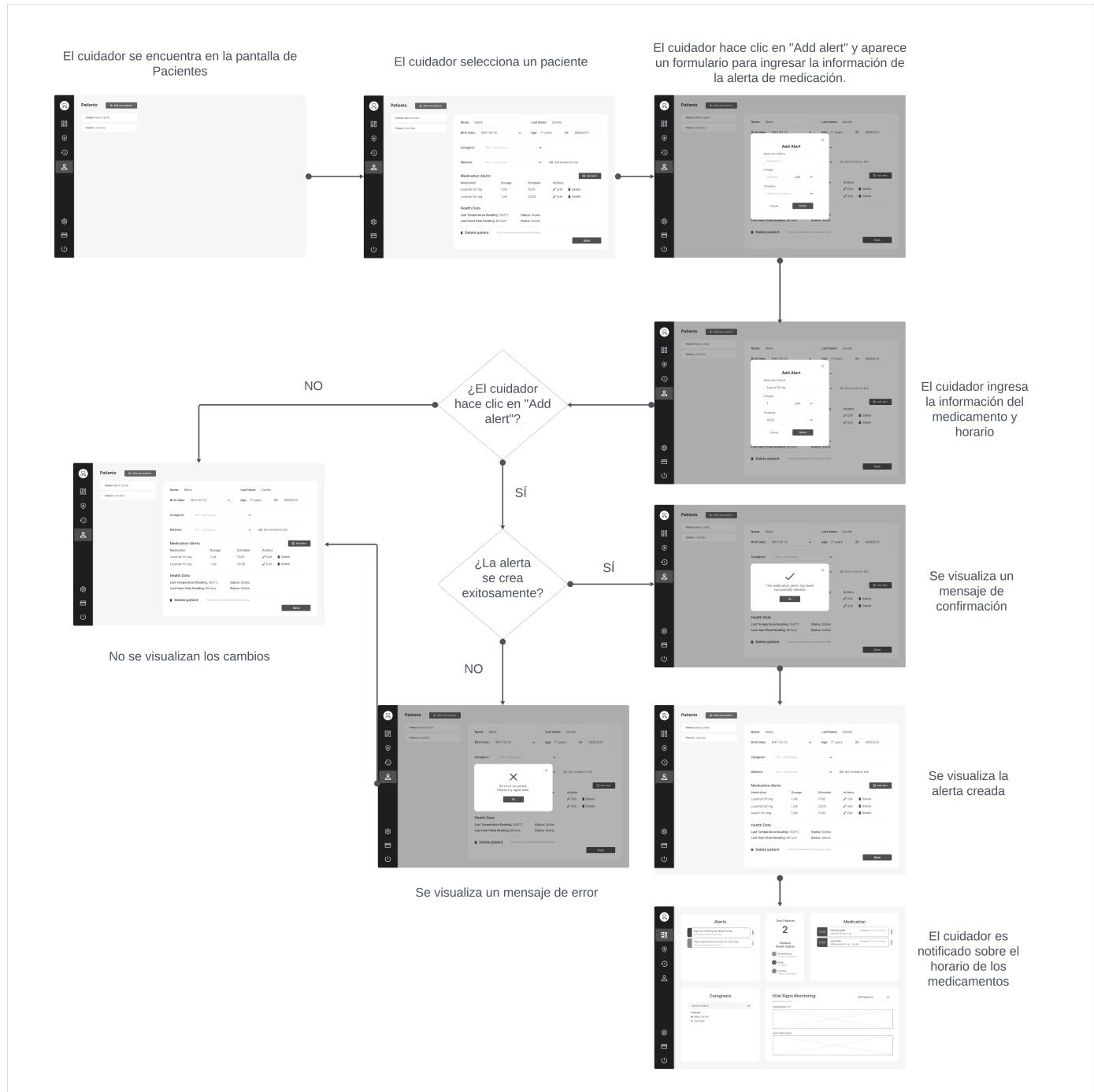
Wireflow 1

User Persona: Ana Martinez

User Goal: Crear una alerta de recordatorio de medicamento del paciente para ser notificada

Descripción: Este flujo ilustra la secuencia de pasos a seguir por Ana cuando quiere crear una alerta de medicamento para ser notificada.

El flujo inicia en la pantalla "Patients". Ana, quien es el cuidador, selecciona un paciente y hace clic en el botón "Add alert". Luego, visualiza un formulario y agrega la información de la alerta. El happy path ocurre la alerta es creada exitosamente y Ana recibe el recordatorio de administración del medicamento.

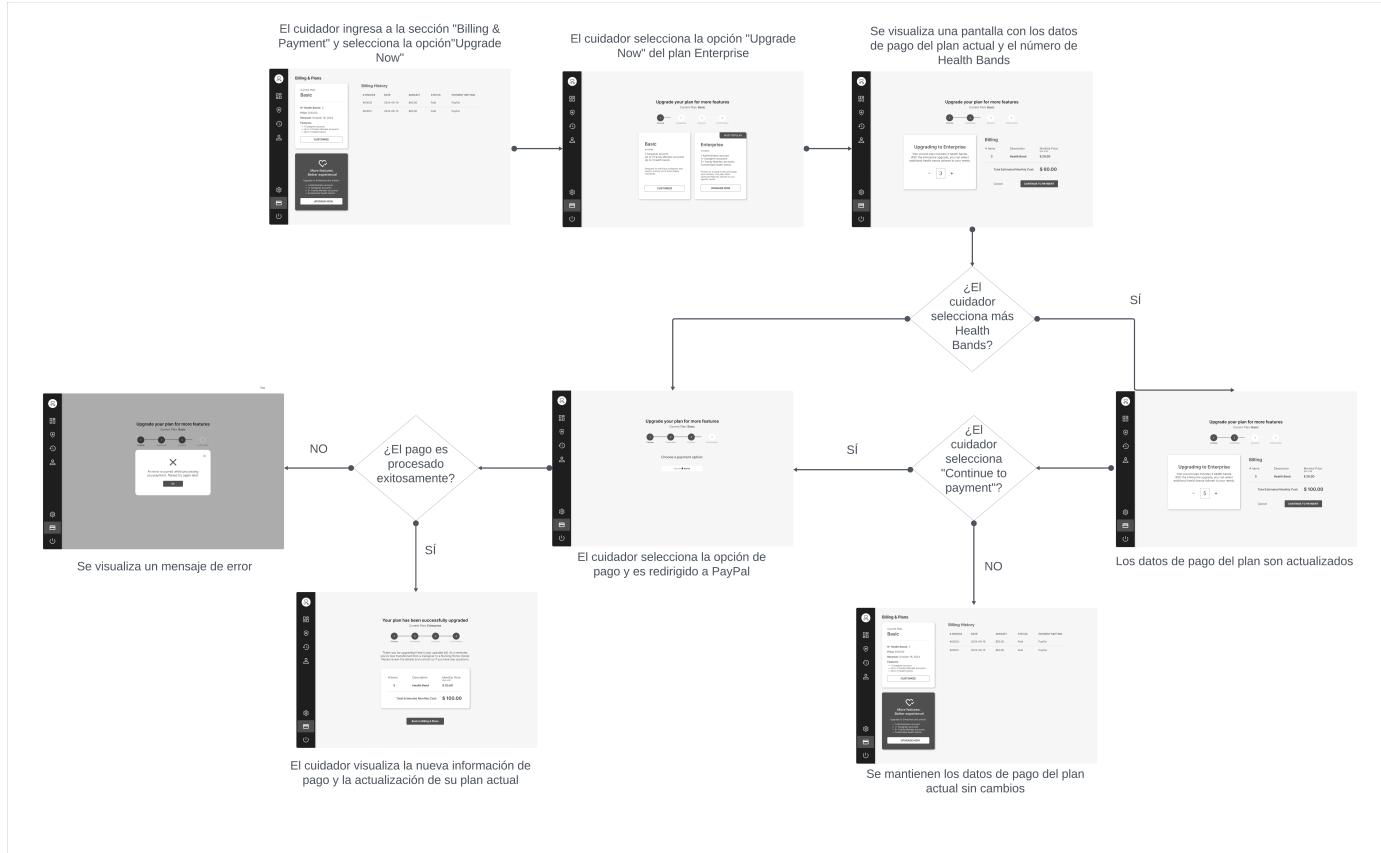


Wireflow 2

User Persona: Ana Martinez

User Goal: Mejorar su plan actual para obtener los beneficios como dueña de una casa de reposo

Descripción: Este flujo ilustra la secuencia de pasos a seguir por Ana al mejorar su plan actual. Inicia cuando Ana ingresa a la pantalla "Billing & Payment", donde se muestra su historial de facturas y datos del plan actual. Cuando Ana hace clic en "Upgrade now", visualiza una sección con la información de los planes disponibles. Al mejorar su plan de Básico a Enterprise, tiene la opción de agregar más Health Bands, lo que se refleja en la pantalla de pago. El happy path ocurre cuando Ana realiza el pago y este es procesado exitosamente, redirigiéndola a una pantalla de confirmación.

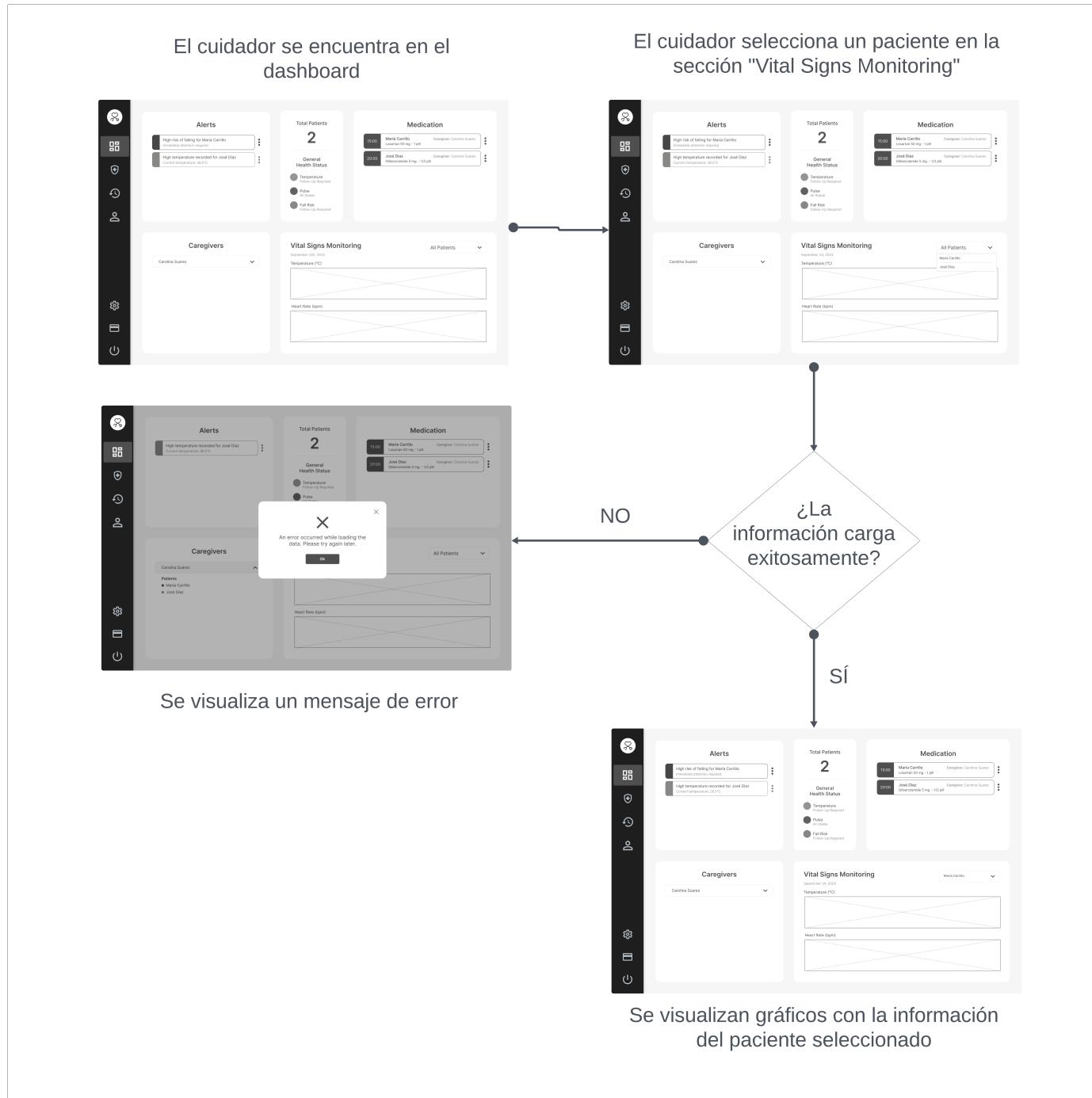


Wireflow 3

User Persona: Ana Martinez & Gabriel Lopez

User Goal: Visualizar los datos vitales de un paciente para tener un monitoreo continuo

Descripción: Este flujo ilustra la secuencia de pasos a seguir, tanto para Ana como para Gabriel, cuando quieren visualizar un resumen de los datos vitales de un paciente. El flujo inicia cuando los usuarios se encuentran en el dashboard, visualizando el apartado de Vital Signs Monitoring. Al hacer clic en el menú desplegable, aparece una lista con todos los pacientes registrados. El happy path ocurre cuando se selecciona un paciente y la gráfica del dashboard se actualiza con los datos de sus funciones vitales.

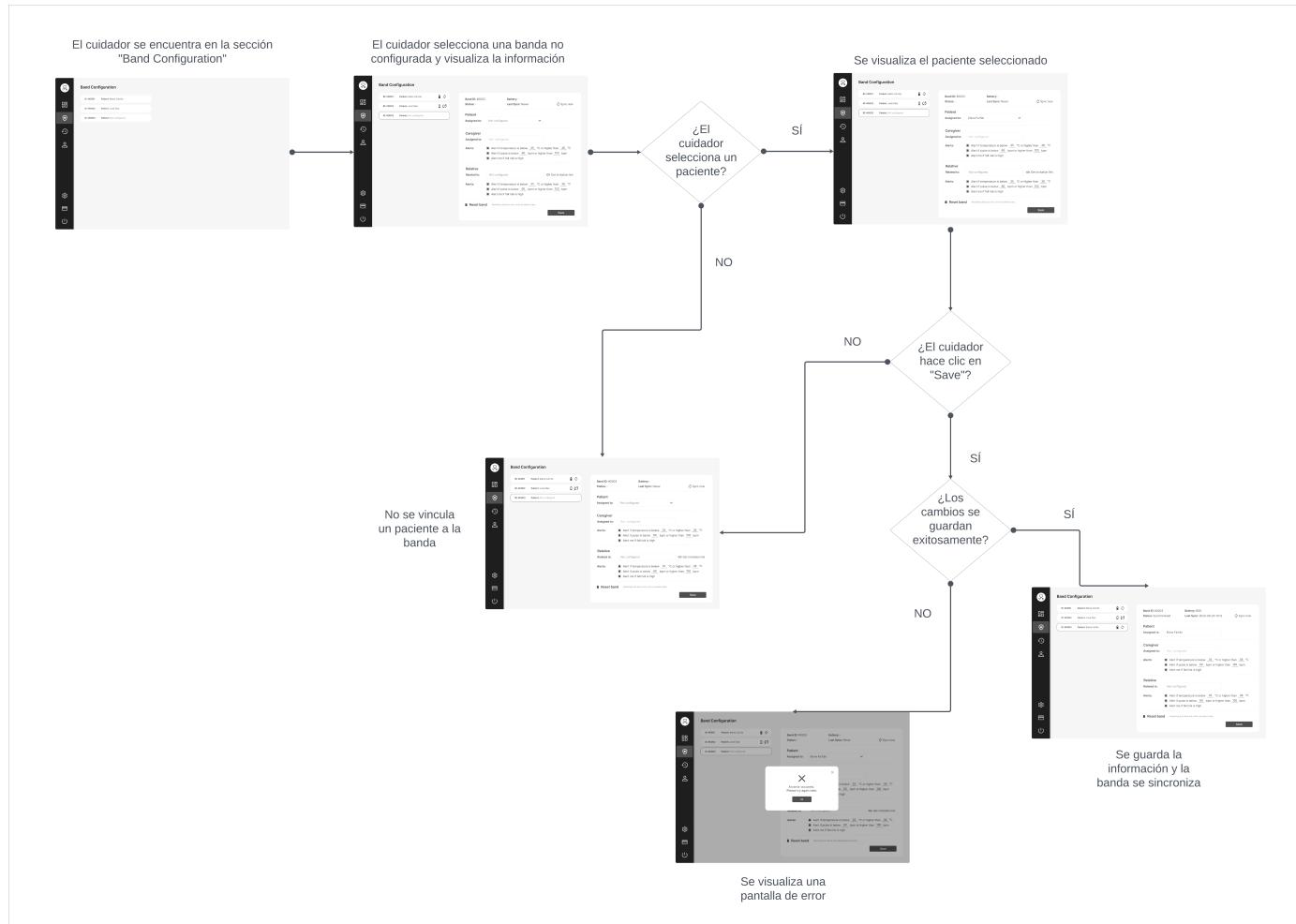


Wireflow 4

User Persona: Ana Martinez & Gabriel Lopez

User Goal: Vincular una banda a un paciente creado

Descripción: Este flujo ilustra la secuencia de pasos a seguir, tanto para Ana como para Gabriel, cuando quieren vincular una banda a un paciente para monitorearlo. El flujo inicia cuando el usuario se encuentra en la vista "Band Configuration" y selecciona una banda que no ha sido configurada. El siguiente paso es elegir un paciente previamente creado del menú desplegable de la sección "Patient". El happy path ocurre cuando, al hacer clic en "Save", la información del paciente se vincula a la banda y esta se sincroniza, empezando a monitorear los signos vitales.

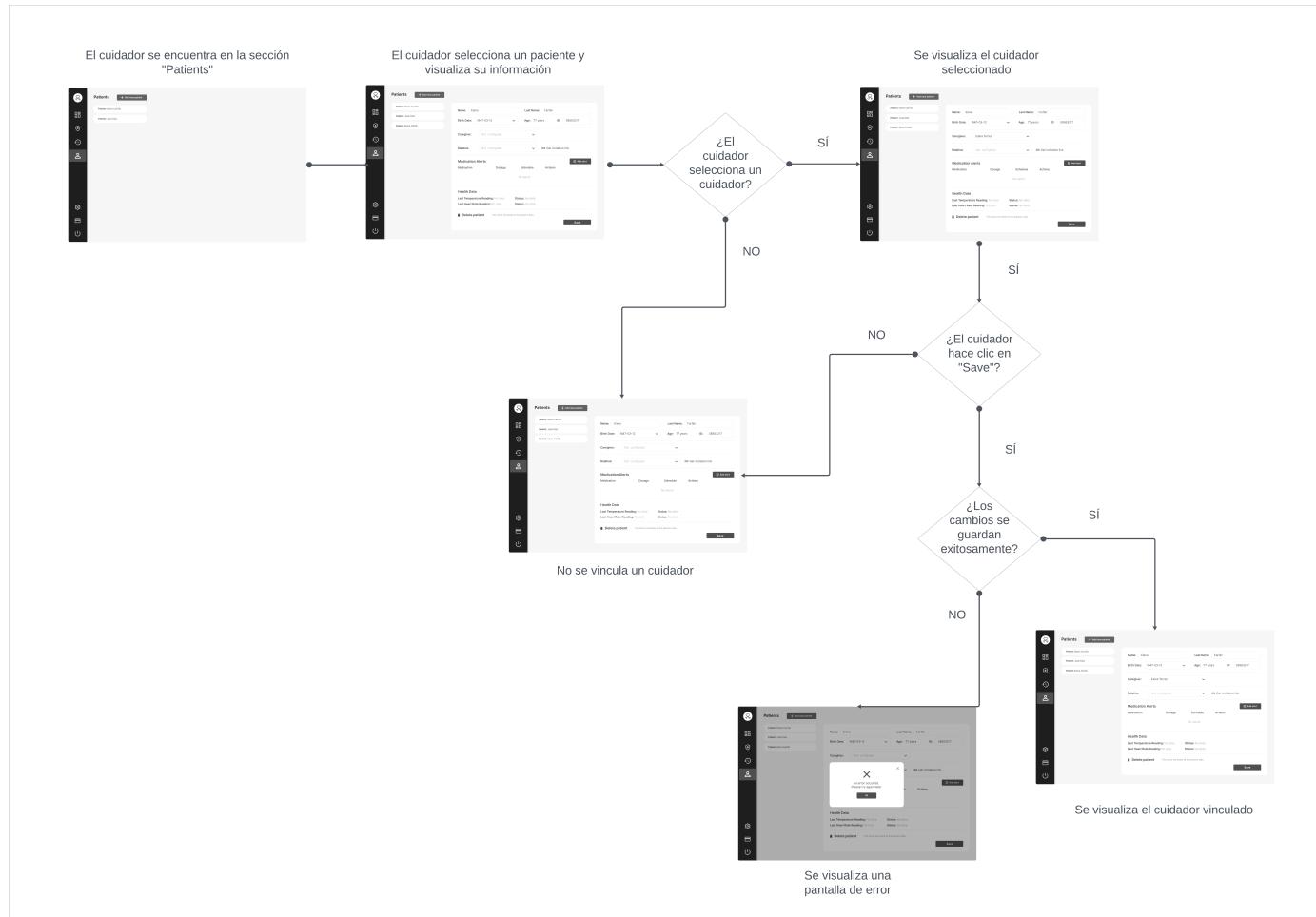


Wireflow 5

User Persona: Gabriel Lopez

User Goal: Vincular una paciente a un cuidador para que reciba las notificaciones

Descripción: Este flujo ilustra la secuencia de pasos a seguir por Gabriel, como dueño de una casa de reposo, para vincular un paciente a un cuidador. El flujo inicia cuando se encuentra en la pantalla "Patients" y selecciona alguno de los pacientes de su lista. Luego, selecciona un cuidador del menú desplegable en la sección "Caregiver", el cual cambia de "No configurado" a mostrar el nombre del cuidador. El happy path ocurre cuando los cambios se guardan exitosamente, el cuidador es vinculado al paciente y puede recibir las alertas.



5.4.2.2. Mobile Application Wireflow Diagrams

5.4.3. Applications Mock-ups

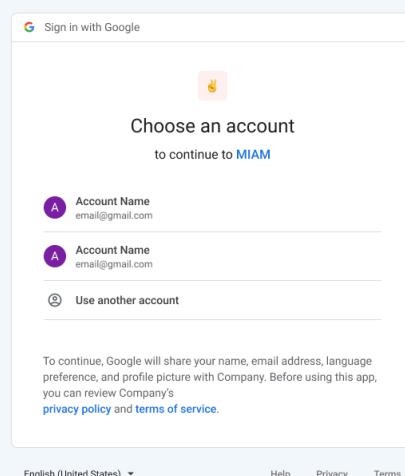
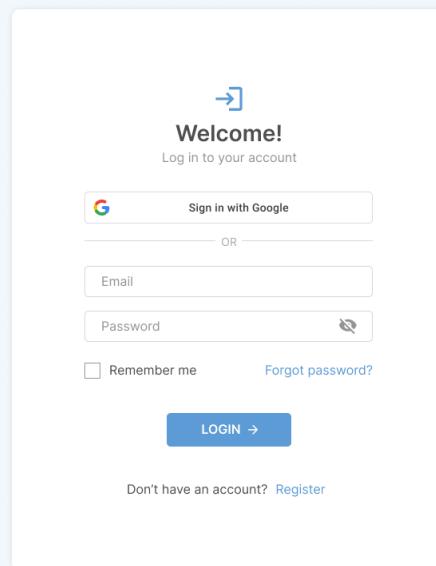
5.4.3.1. Web Application Mock-ups

En esta sección se presentan los mock-ups de la aplicación móvil, mostrando una representación visual más detallada y cercana a la versión final para dispositivos móviles.

Estos mock-ups integran todos los principios de diseño adaptados a la experiencia móvil, como la consistencia visual y la combinación equilibrada de colores, asegurando que la experiencia de usuario sea cohesiva y atractiva en pantallas pequeñas. Cada componente, como botones, tipografías y otros elementos, está diseñado siguiendo las pautas establecidas en el design system, garantizando que se mantenga una coherencia visual a lo largo de toda la aplicación móvil.

A continuación, se muestran los mock-ups de las secciones más importantes de la aplicación móvil, donde se destacan la disposición de los menús, los accesos rápidos y las interacciones táctiles optimizadas para ofrecer una navegación intuitiva y fluida en smartphones:

- Log in



- **Dashboard**

Alerts

- High risk of falling for María Carrillo
Immediate attention required
- High temperature recorded for José Diaz
Current temperature: 38.5°C

Total Patients

2

General Health Status

- Temperature Follow-Up Required
- Pulse All Stable
- Fall Risk Follow-Up Required

Medication

Time	Patient	Medication	Caregiver
15:00	Maria Carrillo	Losartan 50 mg - 1 pill	Carolina Suarez
20:00	José Diaz	Glibenclamide 5 mg - 1/2 pill	Carolina Suarez

Caregivers

Carolina Suarez

Vital Signs Monitoring

September 24h, 2024

Temperature (°C)

Heart Rate (bpm)

Alerts

- High risk of falling for María Carrillo
Immediate attention required
- High temperature recorded for José Diaz
Current temperature: 38.5°C

Total Patients

2

General Health Status

- Temperature Follow-Up Required
- Pulse All Stable
- Fall Risk Follow-Up Required

Medication

Time	Patient	Medication	Caregiver
15:00	Maria Carrillo	Losartan 50 mg - 1 pill	Carolina Suarez
20:00	José Diaz	Glibenclamide 5 mg - 1/2 pill	Carolina Suarez

Caregivers

Carolina Suarez

Vital Signs Monitoring

September 24h, 2024

Temperature (°C)

Heart Rate (bpm)

The dashboard features a sidebar with icons for heart rate, grid, shield, clock, user, gear, card, and power. The main area is divided into four sections:

- Alerts**: Shows two notifications: "High risk of falling for María Carrillo" (immediate attention required) and "High temperature recorded for José Diaz" (current temperature: 38.5°C).
- Total Patients**: Displays a count of 2 patients. Below it, the **General Health Status** is shown with three items: Temperature (Follow-Up Required), Pulse (All Stable), and Fall Risk (Follow-Up Required).
- Medication**: Lists medications for María Carrillo (Losartan 50 mg - 1 pill) at 15:00 and José Diaz (Glibenclamide 5 mg - 1/2 pill) at 20:00, both caregived by Carolina Suarez.
- Caregivers**: Shows a dropdown menu set to "Carolina Suarez".
- Vital Signs Monitoring**: Displays temperature and heart rate trends over 24 hours. Temperature starts at ~36.8°C, rises to a peak of ~37.5°C at 08:00, and ends at ~36.9°C. Heart rate starts at ~80 bpm, dips slightly, and ends at ~79 bpm.

- **Band configuration**

The screenshot shows the 'Band Configuration' screen. On the left is a vertical navigation bar with icons for Home, Devices, Caregivers, Settings, Payments, and Power. The main area is titled 'Band Configuration' and lists three patient bands:

- ID: #0001 Patient: María Carrillo (green battery icon)
- ID: #0002 Patient: José Díaz (red battery icon)
- ID: #0003 Patient: Not configured (green battery icon)

This screenshot shows the detailed configuration for Patient ID #0003. The top header includes the band ID, battery level (80%), status (Synchronized), last sync date (2024-09-24 10:12), and a 'Sync now' button.

Patient
Assigned to: Elena Farfán

Caregiver
Assigned to: Not configured

Alerts:
 Alert if temperature is below 35 °C or higher than 38 °C
 Alert if pulse is below 60 bpm or higher than 100 bpm
 Alert me if fall risk is high

Relative
Related to: Not configured

Alerts:
 Alert if temperature is below 35 °C or higher than 38 °C
 Alert if pulse is below 60 bpm or higher than 100 bpm
 Alert me if fall risk is high

Reset band Resetting this band will unlink all patient data. **Save**

- **Alerts**

DATE ↑	HOUR	PATIENT ▾	ALERT TYPE ▾	DESCRIPTION	CAREGIVER ▾	ACTIONS ▾
2024-09-24	09:49 AM	María Carrillo	High risk of falling	Increased risk of falling	Carolina Suarez	Monitored
2024-09-24	09:49 AM	José Díaz	High temperature	Current temperature: 38.5°C	Carolina Suarez	Monitored

Generate Report

Select a time period to generate the report

Last week

Cancel Download

- Patients

The screenshot shows a mobile application interface for managing patient profiles. On the left is a vertical navigation bar with icons for heart, square grid, shield, clock, settings, card, and power. The main content area has a header "Patients" with a green button "+ Add new patient". Below this, a list shows "Patient: María Carrillo" and "Patient: José Díaz". The main form for "Patient: María Carrillo" includes fields for Name (María), Last Name (Carrillo), Birth Date (1947-03-12), Age (77 years), ID (08563210), Caregiver (Not configured), Relative (Not configured), and a "Get invitation link" button. A "Medication Alerts" section lists two entries: "Losartan 50 mg" at 15:00 and "Losartan 50 mg" at 23:00, each with "Edit" and "Delete" buttons. A "Health Data" section shows "Last Temperature Reading: 36.8°C" (Status: Stable) and "Last Heart Rate Reading: 88 bpm" (Status: Stable). A "Delete patient" button with a warning message is present, along with a "Save" button.

This screenshot shows the same application interface but for a different patient profile. The main form is for "Patient: Elena Farfán". The fields are identical to the first profile: Name (Elena), Last Name (Farfán), Birth Date (1947-03-12), Age (77 years), ID (08563217), Caregiver (Not configured), Relative (Not configured), and a "Get invitation link" button. The "Medication Alerts" section shows "No alerts". The "Health Data" section shows "Last Temperature Reading: No data" (Status: No data) and "Last Heart Rate Reading: No data" (Status: No data). A "Delete patient" button with a warning message is present, along with a "Save" button.

The screenshot shows a patient management interface. On the left is a sidebar with icons for heart, square grid, shield, clock, user (highlighted), gear, card, and power. The main area has a header "Patients" with a "Add new patient" button. Below it, two patients are listed: "Patient: María Carrillo" and "Patient: José Díaz". A modal dialog titled "Delete Patient" is open, asking "Are you sure you want to delete all data for this patient? This action cannot be undone." It contains "Cancel" and "Delete" buttons. To the right of the modal are "Actions" buttons for "Edit" and "Delete" for both María Carrillo and José Díaz. Below the modal, "Health Data" is shown with temperature and heart rate readings, both marked as "Stable". A "Delete patient" button with a warning message is present, along with a "Save" button.

This screenshot is similar to the first one but with a different modal dialog. The "Add Alert" dialog is open, prompting for "Medication Name" (Aspirin 81 mg), "Dosage" (1 pills), and "Schedule" (10:00). It includes "Cancel" and "Delete" buttons. The background shows the same patient list and health data as the first screenshot, with a "Save" button at the bottom.

- **Billing**

Billing & Plans

Current Plan:
Basic

Nº Health Bands: 3
Price: \$ 60.00
Renewal: October 18, 2024

Features:

- 1 Caregiver account
- Up to 3 Family Member accounts
- Up to 3 health bands

[CUSTOMIZE](#)

Billing History

# INVOICE	DATE	AMOUNT	STATUS	PAYMENT METHOD
#00002	2024-09-16	\$60.00	Paid	PayPal
#00001	2024-08-15	\$60.00	Paid	PayPal

More features. Better experience!

Upgrade to Enterprise and unlock:

- 1 Administrator account
- 1+ Caregiver accounts
- 3+ Family Member accounts
- Customized health bands

[UPGRADE NOW](#)

The sidebar on the left contains the following icons: Heart, Grid, Shield, Clock, User, Gear, Wallet, and Power.

Upgrade your plan for more features

Current Plan: Basic

1 Choose 2 Customize 3 Payment 4 Confirmation

Basic

Includes:

- 1 Caregiver account
- Up to 3 Family Member accounts
- Up to 3 health bands

Designed for individual caregivers who need to monitor up to three elderly individuals.

[CUSTOMIZE](#)

Enterprise

MOST POPULAR

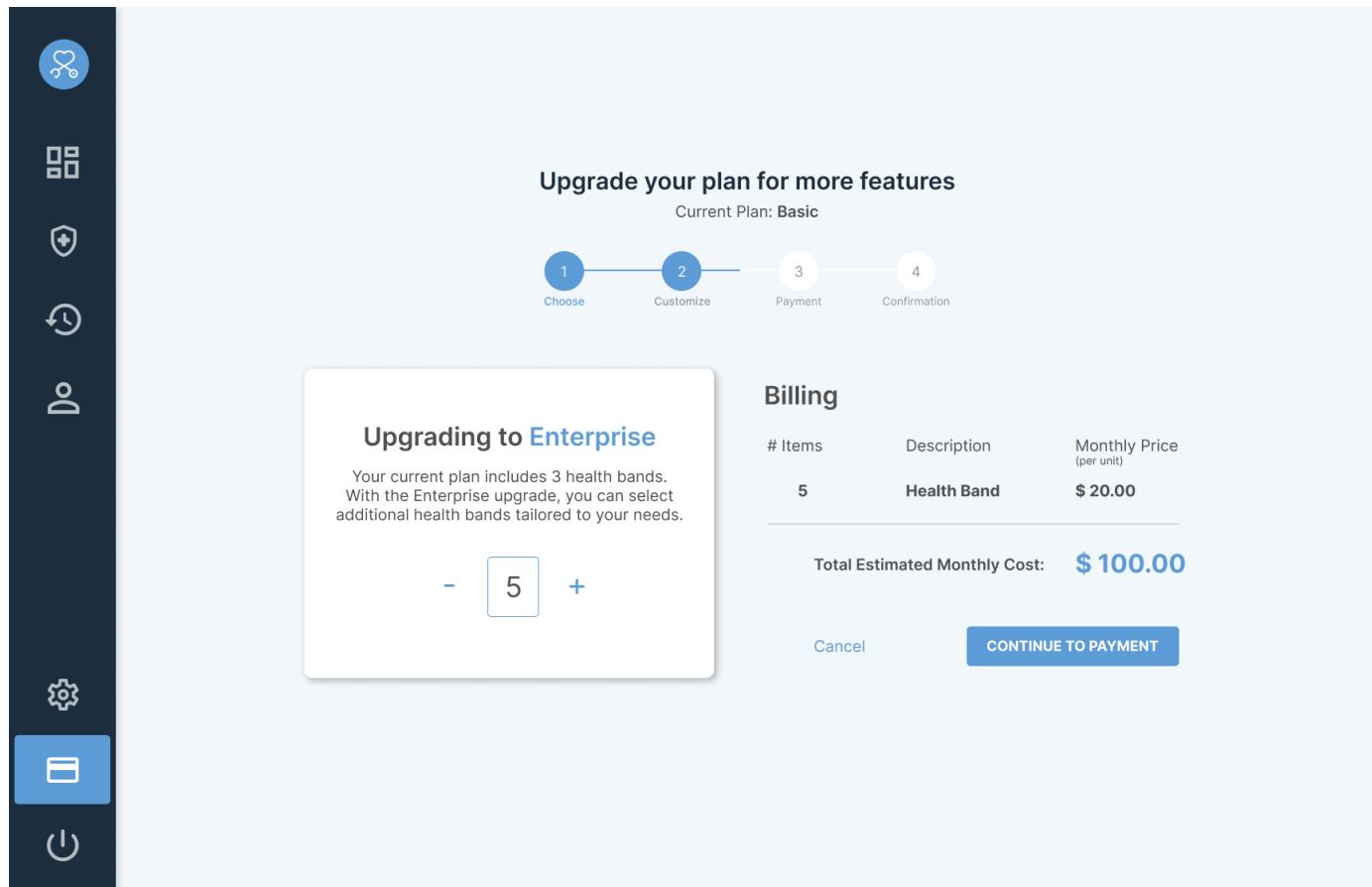
Includes:

- 1 Administrator account
- 1+ Caregiver accounts
- 3+ Family Member accounts
- Customized health bands

Perfect for nursing homes and larger care facilities. This plan offers advanced features tailored to your specific needs.

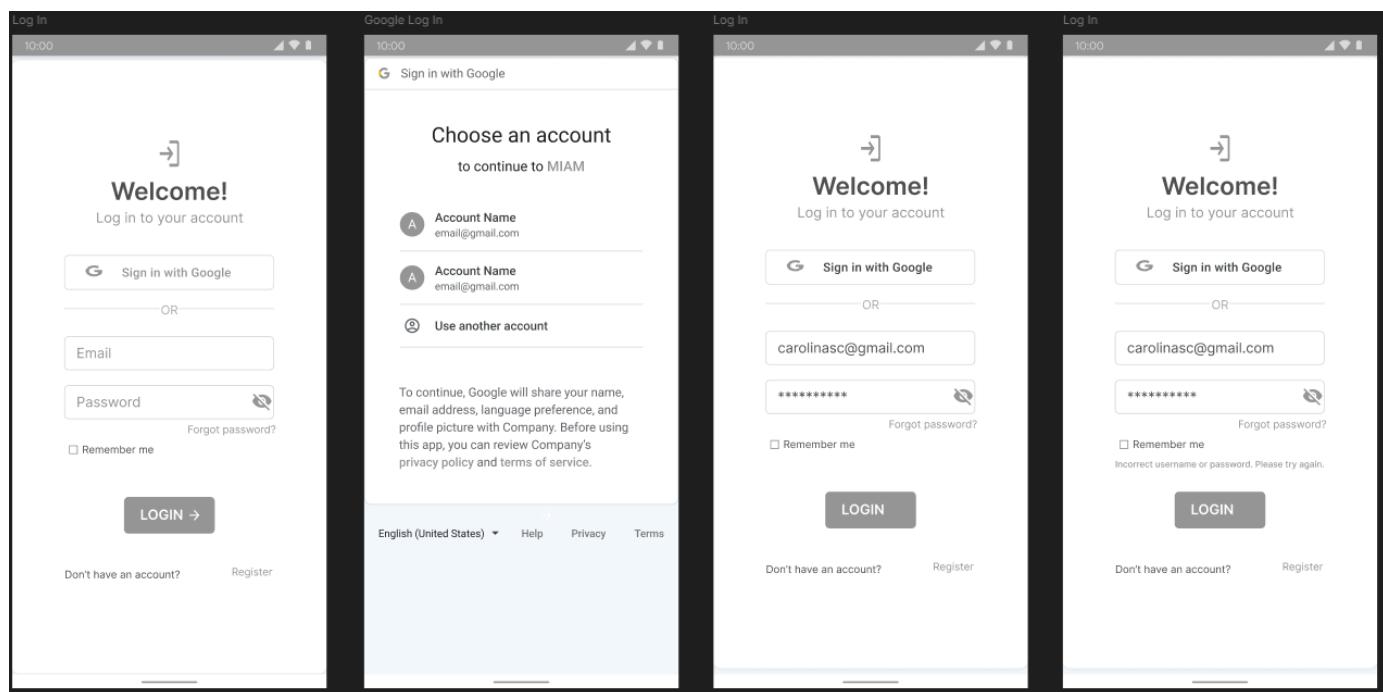
[UPGRADE NOW](#)

The sidebar on the left contains the following icons: Heart, Grid, Shield, Clock, User, Gear, Wallet, and Power.



5.4.3.2. Mobile Application Mock-ups

- Log in



- Dashboard

Alerts

- High risk of falling for María Carrillo
Immediate attention required
- High temperature recorded for José Diaz
Current temperature: 38.5°C

Total Patients

2

Medication

15:00	Maria Carrillo Losartan 50 mg - 1 pill	Caregiver: Carolina Suarez
20:00	José Diaz Glibenclamide 5 mg - 1/2 pill	Caregiver: Carolina Suarez

Caregivers

Carolina Suarez

Vital Signs Monitoring

September 24h, 2024

Temperature (°C)

Heart Rate (bpm)

Alerts

- High risk of falling for María Carrillo
Immediate attention required
- High temperature recorded for José Diaz
Current temperature: 38.5°C

Total Patients

2

Medication

15:00	Maria Carrillo Losartan 50 mg - 1 pill	Caregiver: Carolina Suarez
20:00	José Diaz Glibenclamide 5 mg - 1/2 pill	Caregiver: Carolina Suarez

Caregivers

Carolina Suarez

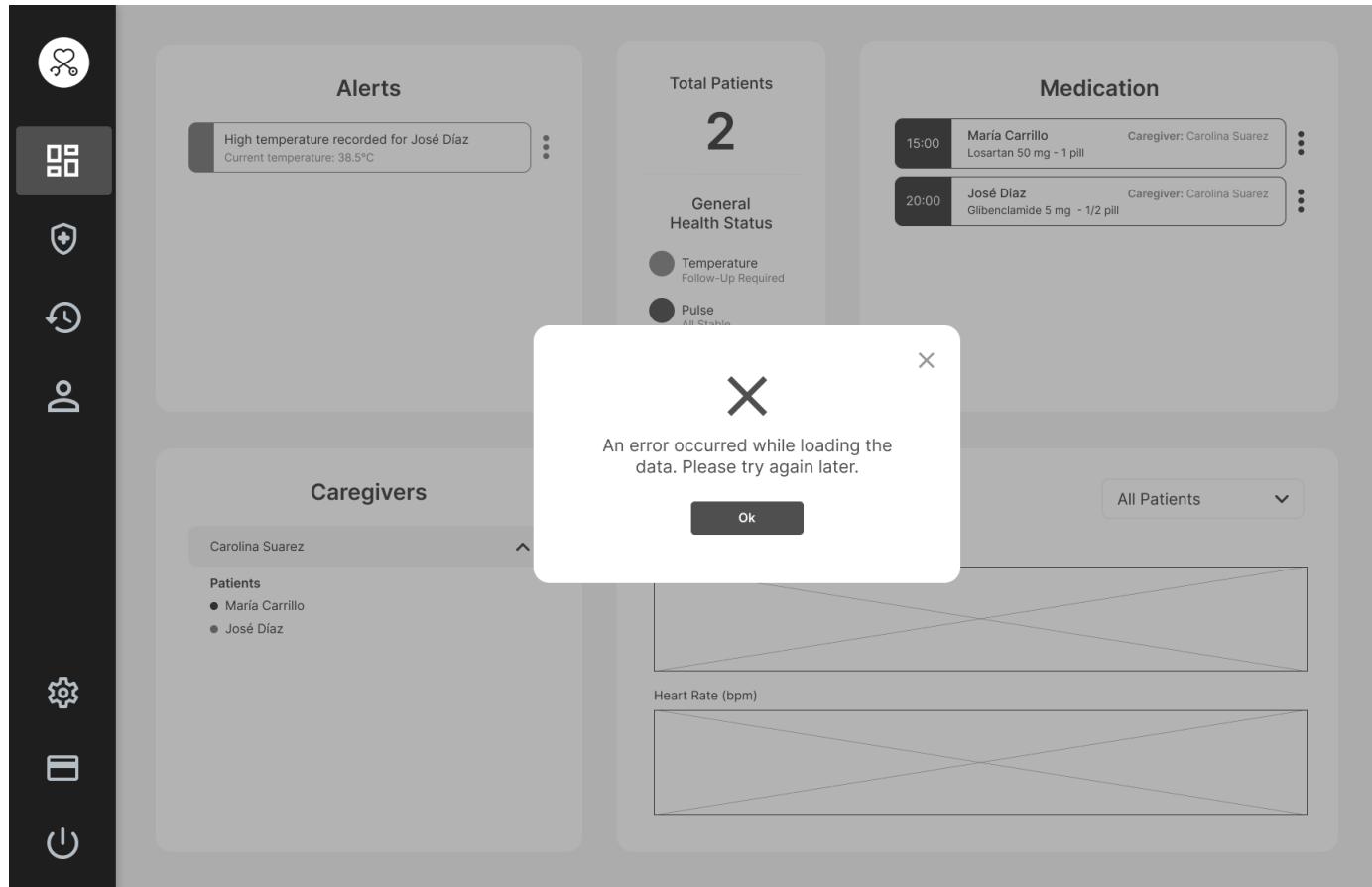
Vital Signs Monitoring

September 24, 2024

Temperature (°C)

Maria Carrillo
José Diaz

Heart Rate (bpm)



- **Band configuration**

The screenshot shows the 'Band Configuration' screen. On the left is a vertical navigation bar with icons for heart rate, square grid, shield (selected), circular arrow, person, gear, card, and power. The main area displays three entries:

- ID: #0001 Patient: María Carrillo
- ID: #0002 Patient: José Díaz
- ID: #0003 Patient: Not configured

The screenshot shows the 'Band Configuration' screen for band ID #0003. The left navigation bar is identical to the first screenshot. The main area is divided into sections:

- Patient:** Assigned to: Elena Farfán
- Caregiver:** Assigned to: Not configured
- Relative:** Related to: Not configured
- Alerts:** Three checkboxes:
 - Alert if temperature is below 35 °C or higher than 38 °C
 - Alert if pulse is below 60 bpm or higher than 100 bpm
 - Alert me if fall risk is high
- Reset band:** A button with a trash icon and the text "Reset band" followed by the note "Resetting this band will unlink all patient data."
- Save:** A large blue "Save" button at the bottom right.

- **Alerts**

Alerts History						
DATE ↑	HOUR	PATIENT ▾	ALERT TYPE ▾	DESCRIPTION	CAREGIVER ▾	ACTIONS ▾
2024-09-24	09:49 AM	Maria Carrillo	High risk of falling	Increased risk of falling	Carolina Suarez	Monitored
2024-09-24	09:49 AM	José Díaz	High temperature	Current temperature: 38.5°C	Carolina Suarez	Monitored

Generate Report

Select a time period to generate the report

Last week ▾

Cancel Download

- Patients

The screenshot shows a medical software interface with a dark sidebar containing icons for heart, square grid, shield, clock, user, gear, card, and power. The main area has a header "Patients" and a button "+ Add new patient". On the left, there's a list of patients: "Patient: María Carrillo", "Patient: José Díaz", and "Patient: New patient". On the right, there's a form to add a new patient with fields for Name, Last Name, Birth Date (with a dropdown menu), ID, and a "Create" button.

The screenshot shows the same medical software interface after adding a new patient. The patient list now includes "Patient: María Carrillo", "Patient: José Díaz", and "Patient: Elena Farfán". The new patient form has been filled out with "Name: Elena", "Last Name: Farfán", "Birth Date: 1947-03-12", "Age: 77 years", "ID: 08563217", "Caregiver: Not configured" (with a dropdown menu showing "Carolina Suarez"), and "Relative: Elena Torres". A "Get invitation link" button is also present. Below this, a "Medication Alerts" section shows a table with columns Medication, Dosage, Schedule, and Actions, and a note "No alerts". The "Health Data" section shows "Last Temperature Reading: No data" and "Status: No data", and "Last Heart Rate Reading: No data" and "Status: No data". At the bottom, a "Delete patient" button with a warning message "This action will delete all the patient's data." and a "Save" button are visible.

- **Billing**

Billing & Plans

Current Plan:
Basic

Nº Health Bands: 3
Price: \$ 60.00
Renewal: October 18, 2024

Features:

- 1 Caregiver account
- Up to 3 Family Member accounts
- Up to 3 health bands

[CUSTOMIZE](#)

Billing History

# INVOICE	DATE	AMOUNT	STATUS	PAYMENT METHOD
#00002	2024-09-16	\$60.00	Paid	PayPal
#00001	2024-08-15	\$60.00	Paid	PayPal

More features. Better experience!

Upgrade to Enterprise and unlock:

- 1 Administrator account
- 1+ Caregiver accounts
- 3+ Family Member accounts
- Customized health bands

[UPGRADE NOW](#)

Upgrade your plan for more features

Current Plan: Basic

1 Choose 2 Customize 3 Payment 4 Confirmation

Basic

Includes:

1 Caregiver account
Up to 3 Family Member accounts
Up to 3 health bands

Designed for individual caregivers who need to monitor up to three elderly individuals.

[CUSTOMIZE](#)

Enterprise

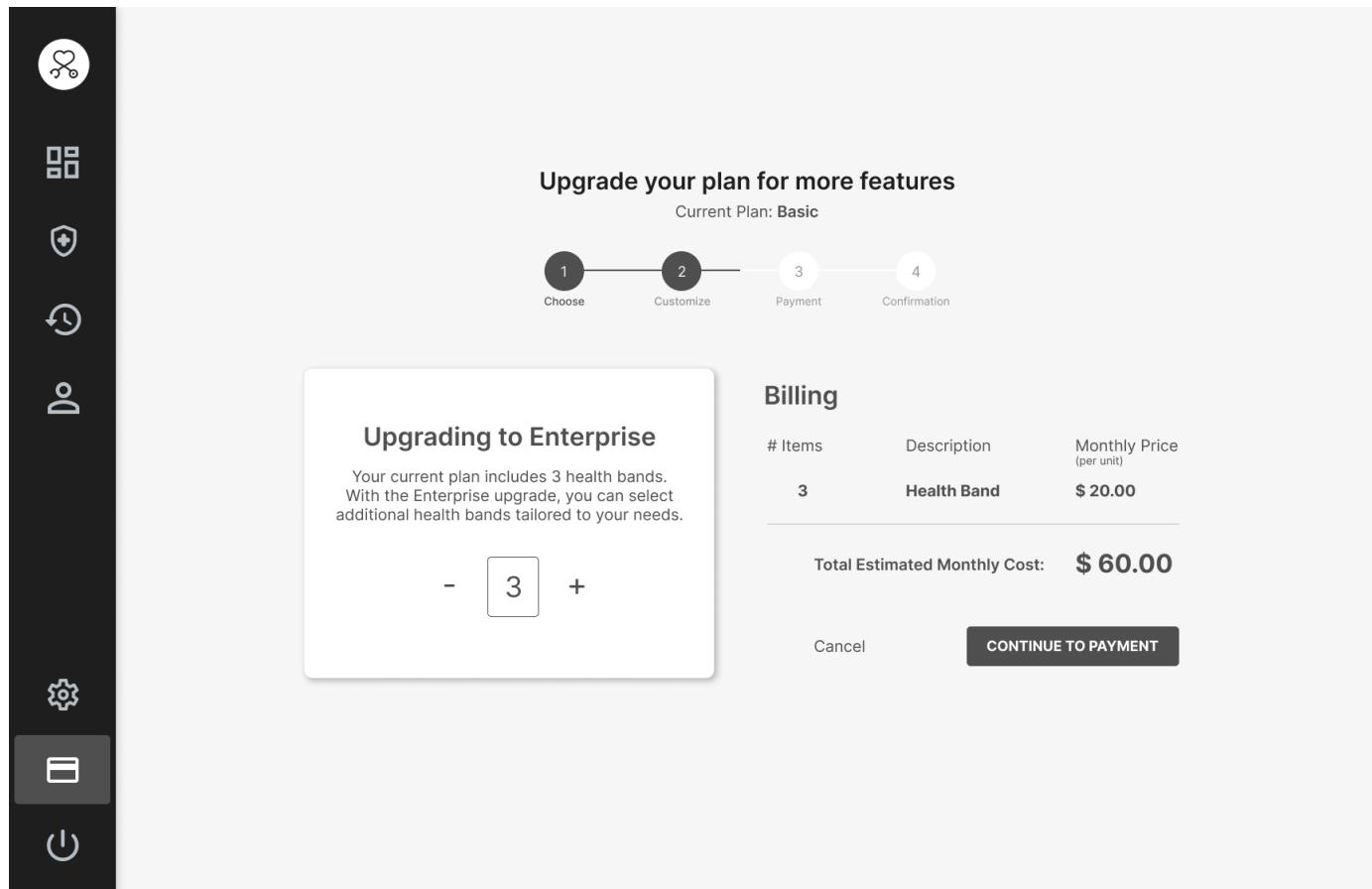
MOST POPULAR

Includes:

1 Administrator account
1+ Caregiver accounts
3+ Family Member accounts
Customized health bands

Perfect for nursing homes and larger care facilities. This plan offers advanced features tailored to your specific needs.

[UPGRADE NOW](#)



5.4.4. Applications User Flow Diagrams

5.4.4.1. Web Application User Flow Diagrams

User Flow 1

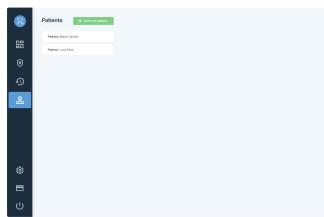
User Persona: Ana Martinez

User Goal: Crear una alerta de recordatorio de medicamento del paciente para ser notificada

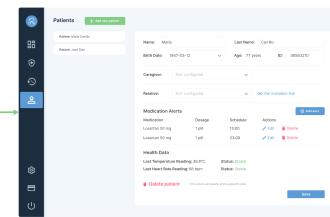
Descripción: Este flujo ilustra la secuencia de pasos a seguir por Ana cuando quiere crear una alerta de medicamento para ser notificada.

El flujo inicia en la pantalla "Patients". Ana, quien es el cuidador, selecciona un paciente y hace clic en el botón "Add alert". Luego, visualiza un formulario y agrega la información de la alerta. El happy path ocurre la alerta es creada exitosamente y Ana recibe el recordatorio de administración del medicamento.

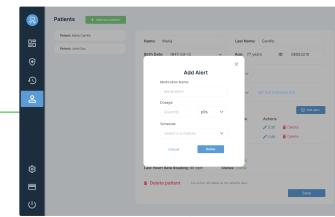
El cuidador se encuentra en la pantalla de Pacientes



El cuidador selecciona un paciente



El cuidador hace clic en "Add alert" y aparece un formulario para ingresar la información de la alerta de medicación.



NO

¿El cuidador hace clic en "Add alert"?

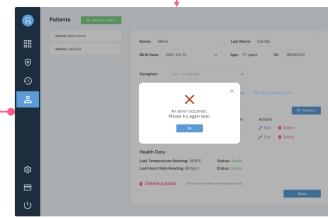
SI

¿La alerta se crea exitosamente?

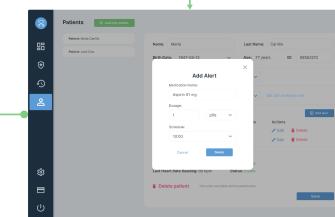
NO

No se visualizan los cambios

⌚ Unhappy Path



Se visualiza un mensaje de error



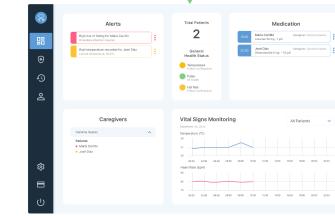
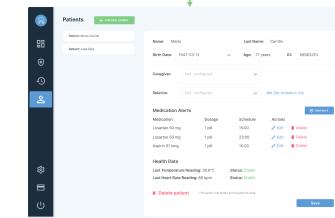
El cuidador ingresa la información del medicamento y horario

Se visualiza un mensaje de confirmación

Se visualiza la alerta creada

El cuidador es notificado sobre el horario de los medicamentos

😊 Happy Path

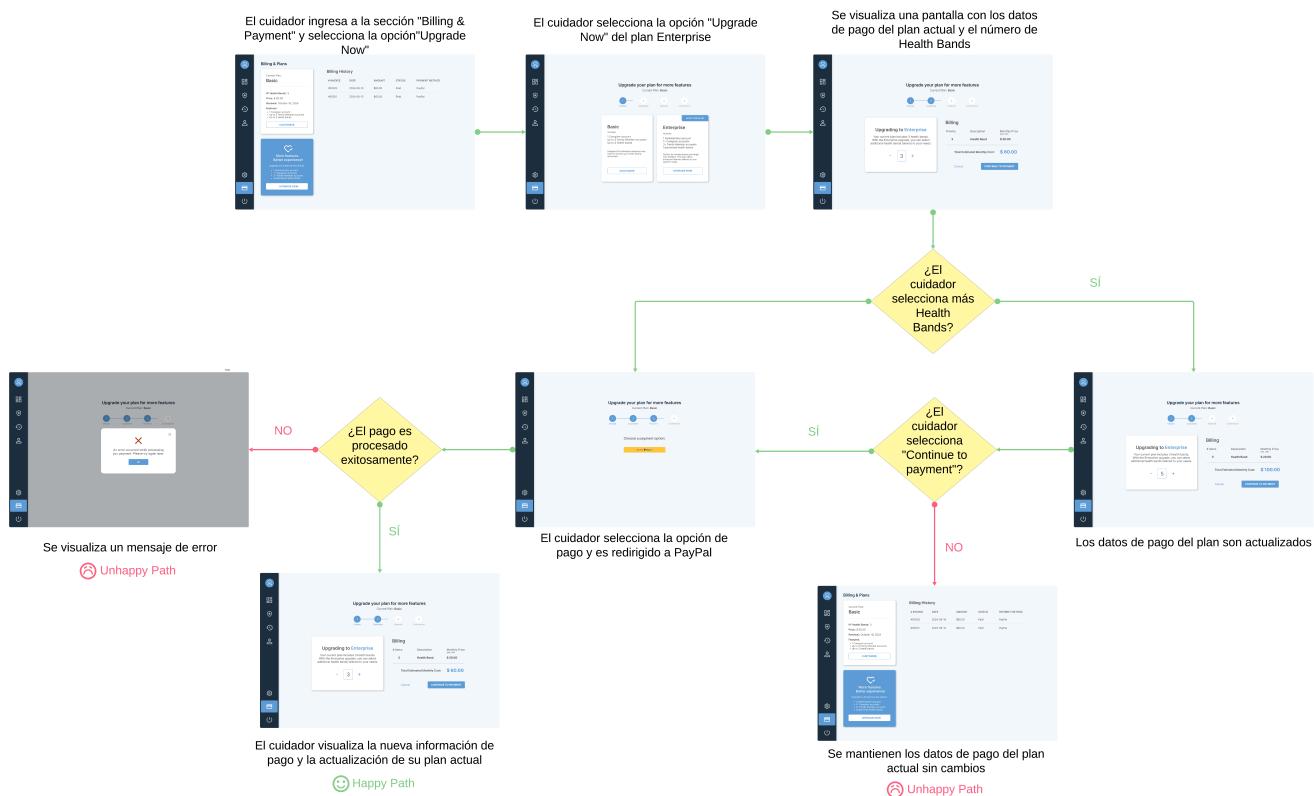


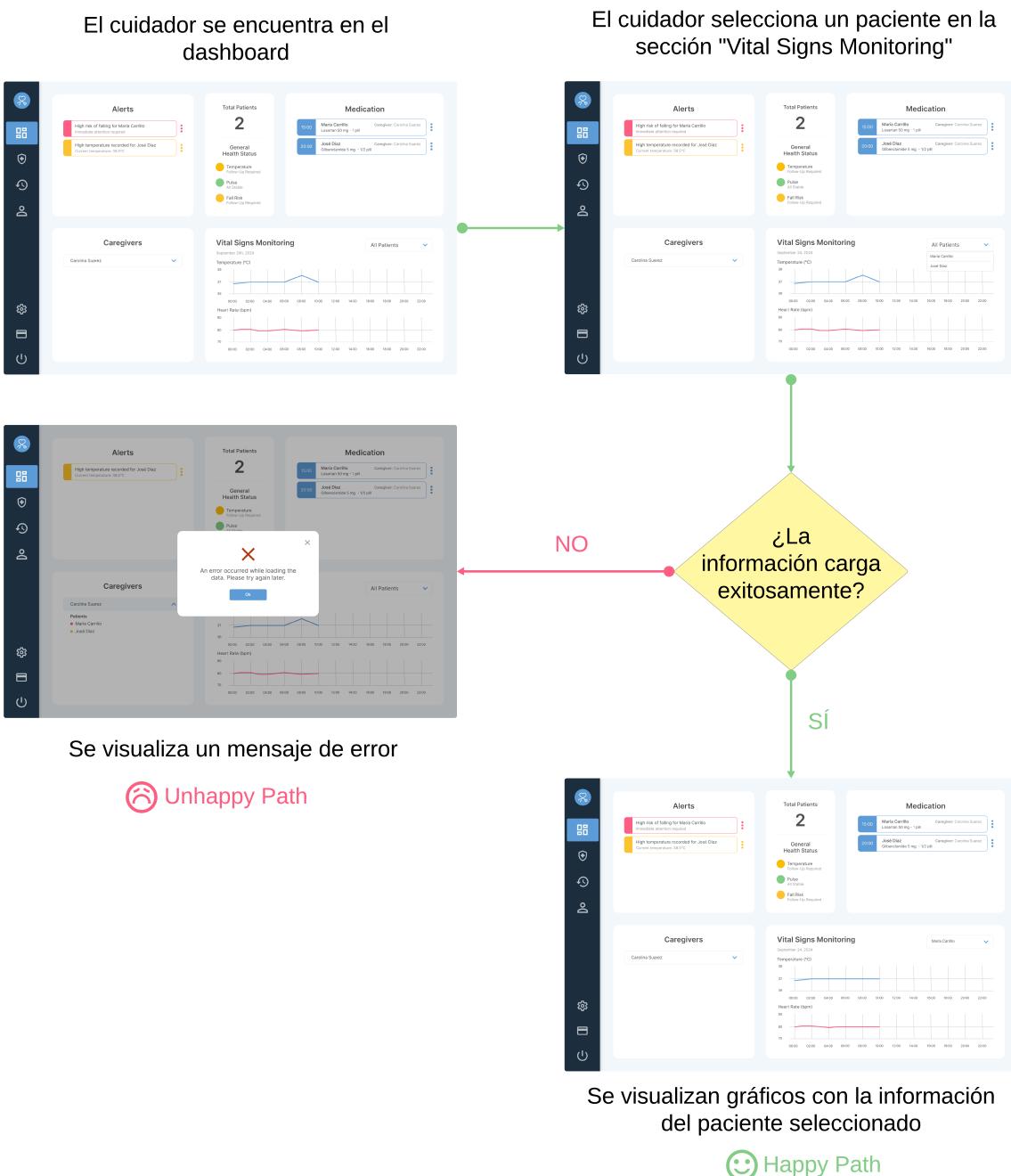
User Flow 2

User Persona: Ana Martinez

User Goal: Mejorar su plan actual para obtener los beneficios como dueña de una casa de reposo

Descripción: Este flujo ilustra la secuencia de pasos a seguir por Ana al mejorar su plan actual. Inicia cuando Ana ingresa a la pantalla "Billing & Payment", donde se muestra su historial de facturas y datos del plan actual. Cuando Ana hace clic en "Upgrade now", visualiza una sección con la información de los planes disponibles. Al mejorar su plan de Básico a Enterprise, tiene la opción de agregar más Health Bands, lo que se refleja en la pantalla de pago. El happy path ocurre cuando Ana realiza el pago y este es procesado exitosamente, redirigiéndola a una pantalla de confirmación.



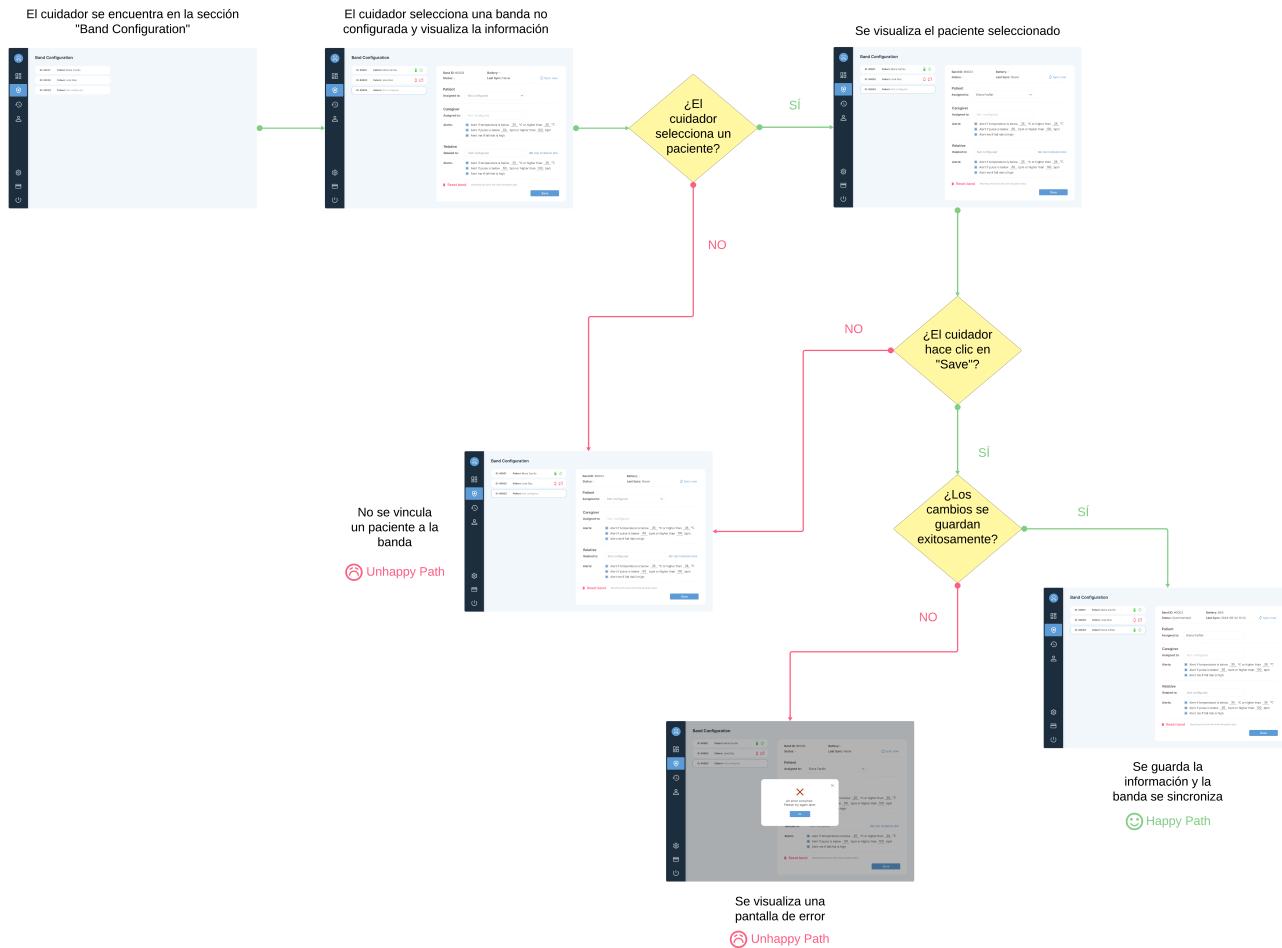


User Flow 4

User Persona: Ana Martinez & Gabriel Lopez

User Goal: Vincular una banda a un paciente creado

Descripción: Este flujo ilustra la secuencia de pasos a seguir, tanto para Ana como para Gabriel, cuando quieren vincular una banda a un paciente para monitorearlo. El flujo inicia cuando el usuario se encuentra en la vista "Band Configuration" y selecciona una banda que no ha sido configurada. El siguiente paso es elegir un paciente previamente creado del menú desplegable de la sección "Patient". El happy path ocurre cuando, al hacer clic en "Save", la información del paciente se vincula a la banda y esta se sincroniza, empezando a monitorear los signos vitales.

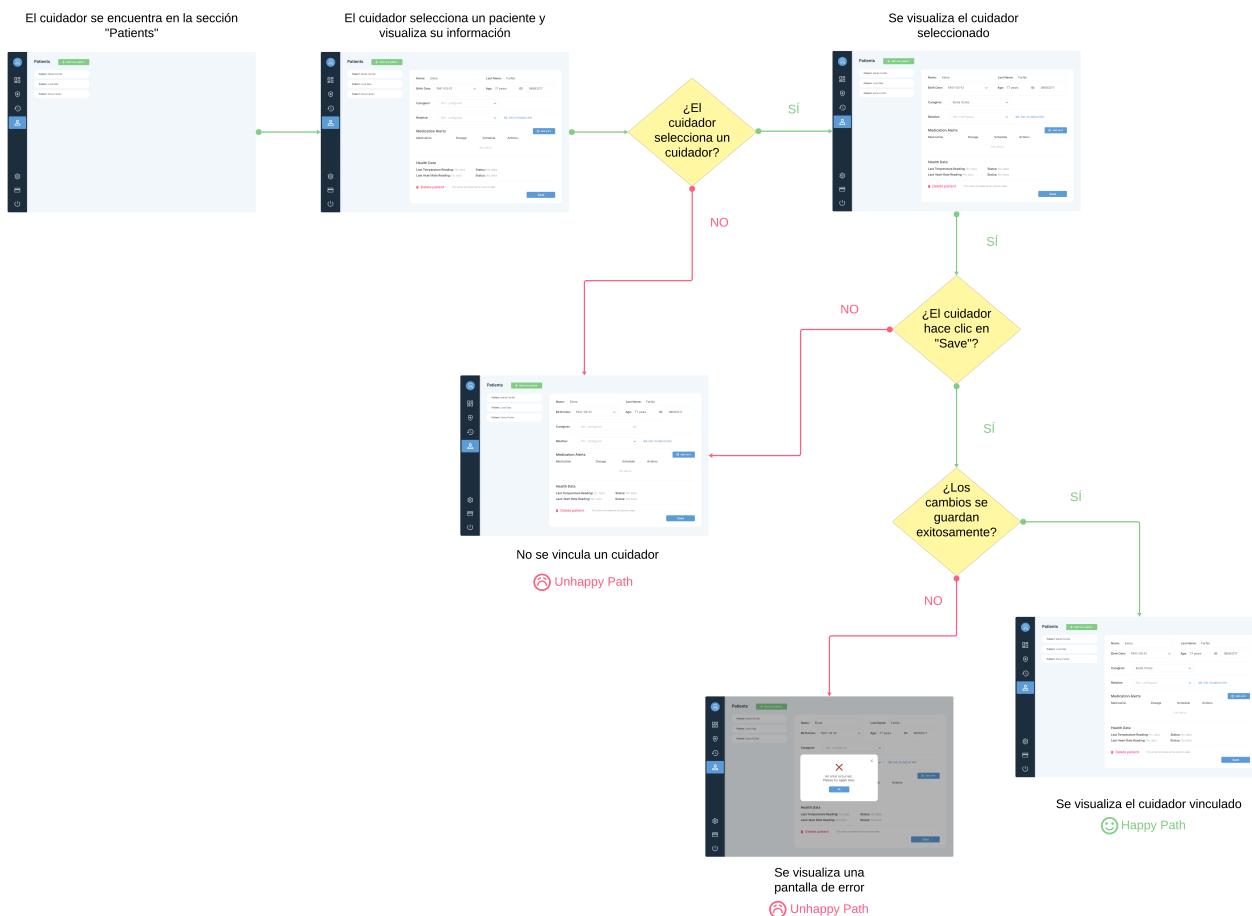


User Flow 5

User Persona: Gabriel Lopez

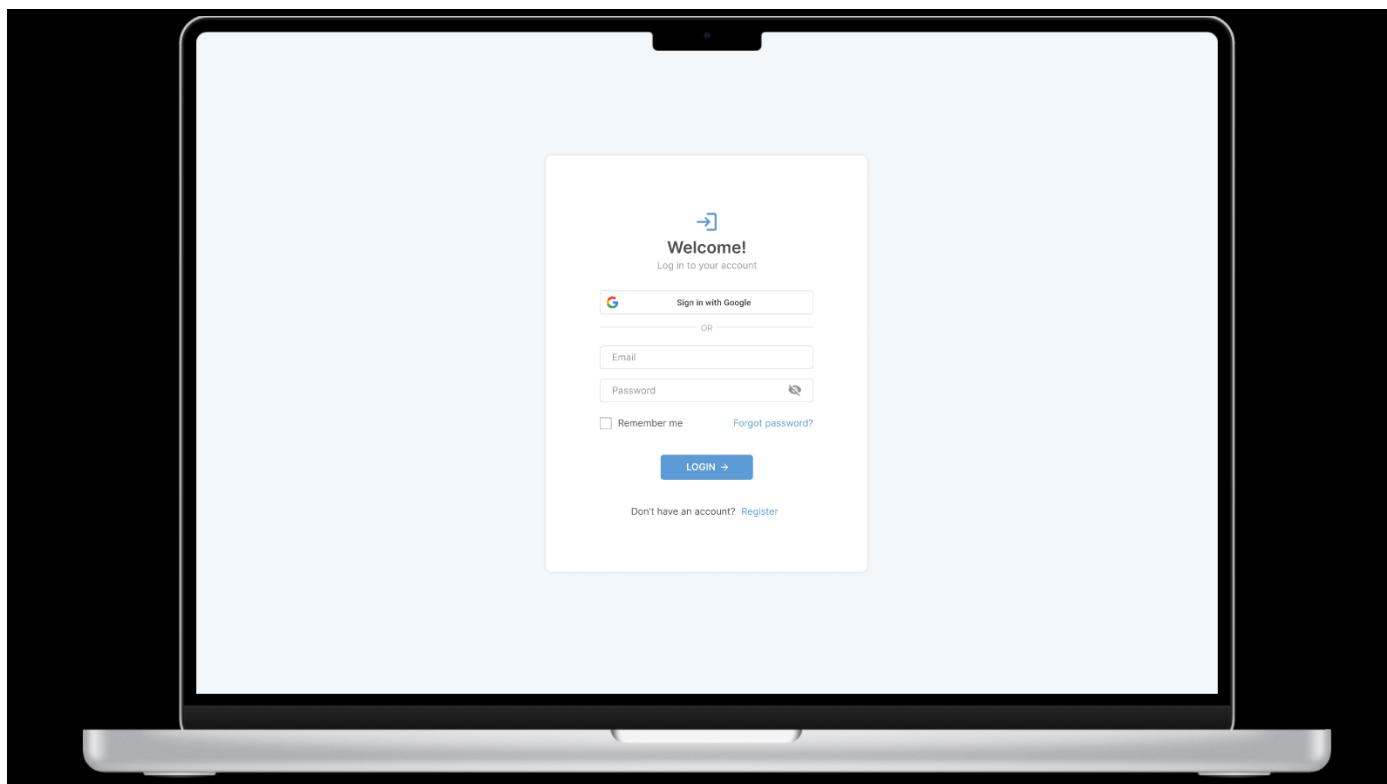
User Goal: Vincular una paciente a un cuidador para que reciba las notificaciones

Descripción: Este flujo ilustra la secuencia de pasos a seguir por Gabriel, como dueño de una casa de reposo, para vincular un paciente a un cuidador. El flujo inicia cuando se encuentra en la pantalla "Patients" y selecciona alguno de los pacientes de su lista. Luego, selecciona un cuidador del menú desplegable en la sección "Caregiver", el cual cambia de "No configurado" a mostrar el nombre del cuidador. El happy path ocurre cuando los cambios se guardan exitosamente, el cuidador es vinculado al paciente y puede recibir las alertas.



5.4.4.2. Mobile Application User Flow Diagrams

5.5 Applications Prototyping



Enlace del prototipo: <https://www.figma.com/proto/HWNBsScQ1yZIRQxCTpWjoF/Mockups?node-id=43-15&node-type=canvas&t=UWTR3uKyOwZLyNxc-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=43-15>

[id=43%3A15&show-sidebar=1](#)

Capítulo VI: Product Implementation, Validation & Deployment

6.1. Software Configuration Management

6.1.1. Software Development Environment Configuration

Project Management Microsoft 365

Alojamiento de los videos de entrevistas, explicación de prototipos y otros relacionados al proyecto Discord

Plataforma de mensajería instantánea donde se realizaron las reuniones, repartición de tarea y desarrollo del proyecto colaborativo.

Whatsapp

Aplicación de mensajería instantánea donde se realizaron recordatorios de las reuniones. Trello

Software de administración Y gestión de proyectos que se utilizó para establecer y designar las tareas Requirements Management

Structurizr

Structurizr es una herramienta de modelado y documentación que permitió el desarrollo de los diagramas C4 MySQL Workbench

Herramienta de diseño para el modelado de diagrama de bases de datos. LucidChart

Herramienta de diseño para el modelado de diagramas UML. Miro

Herramienta de diseño para la creación de los As-Is y To-Be Scenario Mapping Product UX/UI Design Figma

Herramienta que se utilizó para la creación de wireframes, mockups y prototipos. Software Development Git

Es un software de control de versiones para los trabajos en equipos y confiabilidad del desarrollo. Node.js

Node.js es un entorno de ejecución de JavaScript del lado del servidor, que permite desarrollar aplicaciones web escalables y de alto rendimiento fuera del navegador. GitHub

Sistema de control de versiones Git. HTML5

Lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido. CSS

CSS es un lenguaje utilizado para estilizar y dar formato a documentos HTML. JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado y multi-paradigma, utilizado para crear interactividad en páginas web. VScode

Es un editor de código fuente con extensiones que ayudan al desarrollo. Angular Framework

Framework para el desarrollo de frontend Software Deployment Github Pages

Plataforma que nos facilitó realizar el despliegue de nuestro landing page de manera rápida y práctica.

6.1.2. Source Code Management

Utilizamos la metodología de git flow con finalidad es tener un control mayor sobre la gestión del proyecto, avanzar el proyecto de forma simultanea tanto como en la implementación del la aplicación y creación del reporte.

Esto nos permitio separar el proyecto en la rama principal (main) en la que se encuentra la versión presentable del proyecto, que a la vez no cuenta con errores. También tenemos la rama secundario (develop) que sirve para unificar los cambios de las demás ramas. También contamos con más branches una para determinadas funcionalidades amplias de la aplicación y una para cada chapter del reporte.



URL del repositorio del Report en GitHub:

URL del repositorio del Landing Page en GitHub:

URL del repositorio del Frontend en Github:

6.1.3. Source Code Style Guide & Conventions

HTML: Lenguaje utilizado para diseñar páginas web. Este lenguaje utiliza etiquetas para marcar y definir el contenido de la página web; Como un texto, imágenes, videos, etc.

Convenciones:

- Se tiene que declarar el tipo de documento en la primera línea de la siguiente manera:
- Las etiquetas siempre tienen que estar en minúscula, ya que es más claro a la vista y es de fácil escritura para los desarrolladores.
- Usar las comillas para darle un valor a los atributos para una mejor legibilidad.

CSS: Lenguaje que permite dar formatos o estilos a los elementos html. Con este lenguaje se pueden crear diseños web agradables e intuitivos para el usuario.

Convenciones:

- Unir los nombres de las clases compuestas con guiones: box-container{}
- Usar la unidad de medida rem para mantener un diseño responsive: 1rem

6.1.4. Software Deployment Configuration

Para desplegar el landing page, usamos las siguientes herramientas.

- Git: Para realizar el manejo de las versiones del proyecto.
- GitHub: Dentro de la plataforma tenemos un repositorio, lo usamos para almacenar todo el proyecto con sus respectivas versiones.
- GitFlow: Esto permite visualizar y controlar el flujo de trabajo hecho por el equipo de desarrollo.

6.2.1. Sprint 1

6.2.1.1. Sprint Planning 1

Sprint #	Sprint 1
Sprint Planning Background	
Date	05/09/24
Time	16:00
Location	Modalidad remota por Discord
Prepared By	Aliaga Trevejo, Lucía Guadalupe
Attendees (to planning meeting)	<ul style="list-style-type: none"> - Achamizo Huamani, Jean Carlos - Aliaga Trevejo, Lucía Guadalupe - Raymundo Guevara, Rodrigo Alejandro - Siancas Reategui, Luis Alberto - Trujillo Lopez, Luis Alberto - Sagastegui Rodriguez, Luis Jesus
Sprint Goal & User Stories	
Sprint 1 Velocity	12
Sum of Story Points	12

6.2.1.2. Sprint Backlog 1

En esta sección se especifican los detalles del Sprint Backlog, que es una lista de tareas que se han realizado para completar el Sprint.

Sprint #	Sprint 1						
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation(Hours)	Assigned To	Status(To-do/ In-Process/ To-Review/ Done)
US01	Beneficios del Producto	US001-TSK01	Implementar la sección de beneficios	Programar la estructura HTML de la sección de beneficios del producto	1 hora	Jean Achamizo	Done
		US001-TSK02	Agregar estilos de la sección de beneficios Why choose us	Programar estilos de la sección beneficios	0.5 horas	Luis Trujillo	Done
US002	Modelo de	US002-	Implementar la	Programar la	1 hora	Luis	Done

	Negocio	TSK01	sección Our Service	estructura HTML de la sección Our Service		Siancas	
		US002-TSK02	Agregar estilos de la sección Our Service	Programar estilos de la sección Our Service	0.5 horas	Luis Siancas	Done
US003	Reseñas de Clientes	US003-TSK01	Implementar la sección de reseñas	Programar la estructura HTML de la sección de reseñas	1 hora	Jesus Sagastegui	Done
		US003-TSK02	Agregar estilos de la sección de reseñas	Programar estilos de la sección de reseñas	0.5 horas	Lucía Aliaga	Done
US004	Contactar a Miembros de la Compañía	US004-TSK01	Implementar la sección Contact	Programar la estructura HTML de la sección Contact	1 hora	Rodrigo Raymundo	Done
		US004-TSK-02	Agregar estilos de la sección Contact	Programar estilos de la sección Contact	1 hora	Rodrigo Raymundo	Done
US005	Información de Miembros de la Compañía	US005-TSK01	Implementar la sección Meet the team	Programar la estructura HTML de la sección Meet the team	1.5 horas	Luis Trujillo	Done
		US005-TSK02	Agregar estilos de la sección Meet the team	Programar estilos de la sección Meet the team	1 hora	Lucía Aliaga	Done

6.2.1.3. Development Evidence for Sprint Review

En esta sección se presentan los commits realizados en el repositorio de GitHub, donde se puede observar el trabajo colaborativo de cada integrante del equipo en el desarrollo de la Landing Page. A continuación, se detallan las contribuciones individuales en diversas ramas del proyecto, incluyendo las implementaciones de las funcionalidades principales y las integraciones necesarias para el despliegue final del producto.

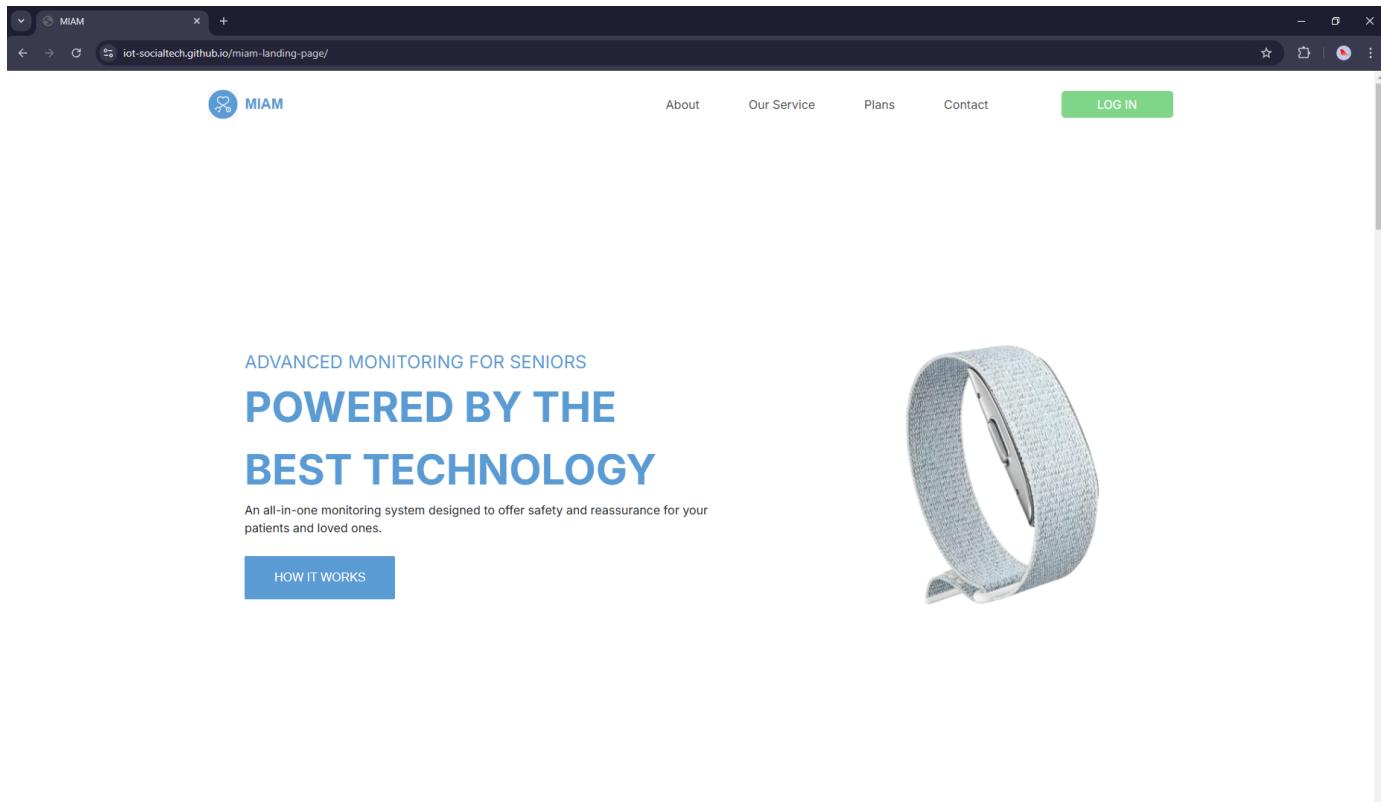
Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
IoT-SocialTech/miam-landing-page	/development	9cc795b	Initial commit	Initial commit	25/09/2024
	/development	41a2388	Feature: NavBar and product information	Feature: NavBar and product information	27/09/2024
	/US01-Product-Information	6aecf7a	Feature: Product benefits and footer	Feature: Product benefits and footer	27/09/2024
	/US02-Business-Model	90281dc	Feature: Information about our Business Model	Feature: Information about our Business Model	27/09/2024
	/US03-CustomerReviews	c2dc9a2	Feature: User Reviews Section	Feature: User Reviews Section	27/09/2024
	/US04-Contact	d407ea8	Feature: Contact Company Members	Feature: Contact Company Members	27/09/2024

/US05-Company-Member-Information	a8e6e9e	Feature: Company Member Information Section	Feature: Company Member Information Section	27/09/2024
/development	b2015d6	Fix: responsive	Fix: responsive	27/09/2024
/development	b160848	fix: responsiveness issues	fix: responsiveness issues	05/10/2024

6.2.1.4. Testing Suite Evidence for Sprint Review

6.2.1.5. Execution Evidence for Sprint Review

Como resultado del primer sprint, se presenta el despliegue de la Landing Page. <https://iot-socialtech.github.io/miam-landing-page/>



6.2.1.6. Services Documentation Evidence for Sprint Review

Link de la landing page: <https://iot-socialtech.github.io/miam-landing-page/>

6.2.1.7. Software Deployment Evidence for Sprint Review

Para el desarrollo de la Landing page, se utilizaron las siguientes tecnologías:

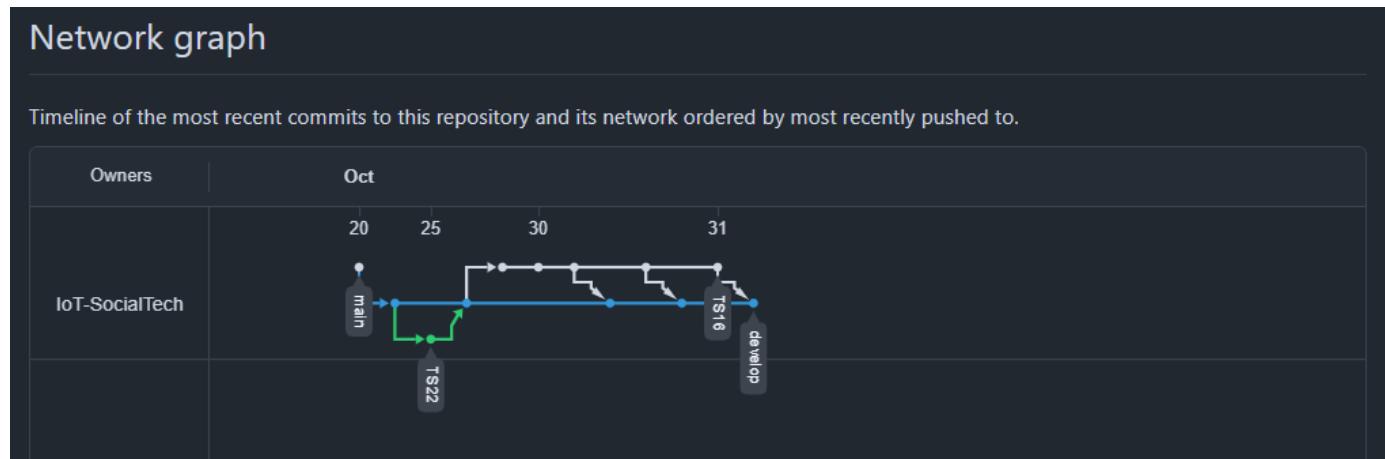
- HTML: Lenguaje de marcado para crear la estructura de una página web. Define la organización del contenido (encabezados, párrafos, imágenes, etc.).
- CSS: Lenguaje de estilo para darle forma y diseño a las páginas web. Controla la apariencia del contenido (colores, tipografías, fondos, etc.).
- Git: Sistema de control de versiones para gestionar el desarrollo de software. Permite realizar un seguimiento de los cambios en el código y colaborar con otros. Sistema de control de versiones que nos ayudó a trabajar en equipo durante el desarrollo del landing page.
- GitHub: Plataforma online para alojar repositorios Git. Permite compartir código, colaborar en proyectos y acceder a una gran comunidad de desarrolladores. Plataforma que nos ayudó al desarrollo colaborativo del equipo para almacenar las versiones de nuestro proyecto.

Igualmente, para el correcto control de versiones y el adecuado desarrollo de la página, se empleó la metodología de trabajo GitFlow Workflow. Es un método para gestionar el desarrollo de software con Git. Define roles y ramas específicas para cada etapa del proceso, como desarrollo, pruebas y lanzamiento. Esto ayuda a organizar el código, facilita la colaboración y reduce el riesgo de errores.

6.2.1.8. Team Collaboration Insights during Sprint

La StartUp ha realizado el presente sprint usando Git y Github como herramientas principales para el control de versiones y GitFlow como metodología de trabajo.

En primer lugar, se creó una organización en Github con el nombre de la StartUp. Luego, se creó un repositorio para la Landing page. En dicho repositorio, el equipo ha creado diferentes ramas (branches) para diferenciar el tipo de trabajo que se realiza. Por ejemplo, la rama `main` se empleó únicamente para el despliegue final de la aplicación; `develop`, para hacer *merge* con demás ramas incluida la `main`; `feature`, para implementar alguna funcionalidad. De esta manera se protege el código funcional y se diferencia de features en desarrollo por los miembros. Después de la creación de cada rama, uno o más usuario pueden acceder a esta y realizar cambios y guardarlos como commits.



6.2.2. Sprint 2

6.2.2.1. Sprint Planning 2

Sprint #	Sprint 2
Sprint Planning Background	
Date	04/10/2024
Time	168:30
Location	Modalidad remota por Discord
Prepared By	Aliaga Trevejo, Lucía Guadalupe
Attendees (to planning meeting)	<ul style="list-style-type: none"> - Achamizo Huamani, Jean Carlos - Aliaga Trevejo, Lucía Guadalupe - Raymundo Guevara, Rodrigo Alejandro - Siancas Reategui, Luis Alberto - Trujillo Lopez, Luis Alberto - Sagastegui Rodriguez, Luis Jesus
Sprint 1 Review Summary	En el sprint anterior, se completó el desarrollo de las secciones de la landing page. Esto incluyó el diseño visual, la estructura de navegación y la implementación de los componentes necesarios para una experiencia de usuario funcional.
Sprint 1 Retrospective Summary	La organización de las tareas para el desarrollo de la landing page fue efectiva y permitió cumplir con el objetivo del sprint. Uno de los principales desafíos fue la gestión del tiempo. Se identificó la necesidad de priorizar y segmentar las tareas complejas para evitar sobrecargas y distribuir el trabajo de manera más equilibrada en el equipo
Sprint Goal & User Stories	
Sprint 2 Goal	Our focus is on establishing the foundational features for user onboarding, bracelet configuration, and initial data collection. We believe it delivers a seamless entry experience for caregivers and nursing home owners, enabling them to begin using the monitoring system effectively and gather essential health data from the start. This will be confirmed when users can successfully register, log in, configure their bracelets, and view initial health data collected from the bracelets in real time.
Sprint 1 Velocity	80
Sum of Story Points	102

6.2.2.2. Sprint Backlog 2

En esta sección se especifican los detalles del Sprint Backlog, que es una lista de tareas que se han realizado para completar el Sprint 2.

El objetivo de este sprint es establecer las características fundamentales necesarias para la creación de usuarios y pacientes en el sistema de monitoreo. Este sprint se enfocará en el desarrollo de funcionalidades para el registro de cuidadores, la configuración de los brazaletes (health band) y la recolección inicial de datos de salud. Al implementar estas características, se busca proporcionar una experiencia de entrada fluida para los cuidadores y propietarios de las casas de reposo, facilitando así el uso efectivo del sistema desde su inicio.

The screenshot shows a Trello board with the following structure:

- Goal:** Our focus is on establishing the foundational features for user onboarding, bracelet configuration, and initial data collection. We believe it delivers a seamless entry experience for caregivers and nursing home owners, enabling them to begin using the monitoring system effectively and gather essential health data from the start. This will be confirmed when users can successfully register, log in, configure their bracelets, and view initial health data collected from the bracelets in real time.
- Stories:**
 - US06: Monitoreo de Temperatura en Tiempo Real (0/5)
 - US07: Monitoreo del Ritmo Cardíaco en Tiempo Real (0/5)
 - US015: Acceso a datos de Usuario (0/5)
 - US011: Envío de Notificación de Emergencia (0/3)
 - US019: Recepción de Notificaciones de Ayuda (0/3)
 - US019: Monitoreo de Sensores de Pulseras (0/5)
 - US020: Asignación de Pulsera a un Usuario (0/5)
 - US022: Configuración de Pagos Automatizados (0/3)
 - TS07: Endpoint para Monitoreo de Temperatura (0/2)
 - TS08: Endpoint para Monitoreo de Ritmo Cardíaco (0/2)
- To Do:** + Añade una tarjeta
- In Process:**
 - US011-TSK02: Implementar la lógica para recibir notificaciones en la aplicación web
 - US012-TSK02: Implementar la interfaz de notificaciones en la aplicación web
 - US02-TSK01: US19-TSK01: Implementar la interfaz móvil y lógica para la configuración de pagos
 - US2-TSK02: Implementar la interfaz web y lógica para la configuración de pagos
 - US19-TSK01: Implementar la interfaz móvil y lógica para visualizar el estado de los sensores
 - US19-TSK02: Implementar la interfaz web y lógica para visualizar el estado de los sensores
 - TS07-TSK02: Crear pruebas unitarias para validar la correcta obtención de la temperatura y manejo de errores
 - TS09-TSK02: Implementar manejo de errores para situaciones como healthband invalido o falta de datos
 - TS09-TSK01: Implementar la funcionalidad para enviar alertas de salud cuando se detecten valores anormales.
- To Review:** + Añade una tarjeta
- To Fix:**
 - US15-TSK03: Implementar el apartado de los pacientes
 - US06-TSK02: Implementar la interfaz de usuario de la aplicación móvil para mostrar la temperatura
 - US06-TSK03: Implementar la interfaz de usuario de la aplicación web para mostrar la temperatura
 - US07-TSK02: Implementar la interfaz de usuario de la aplicación móvil para mostrar el pulso
 - US07-TSK03: Implementar la interfaz de usuario de la aplicación web para mostrar el pulso
- Done:**
 - US06-TSK01: Toma de temperatura en embedded application
 - US07-TSK01: Toma de pulso en embedded application
 - US11-TSK01: Implementar el botón de emergencia en el embedded application
 - US15-TSK01: Implementar formulario para agregar pacientes
 - US12-TSK02: Implementar CRUD para pacientes
 - US11-TSK02: Implementar notificaciones en aplicación móvil
 - US12-TSK01: Implementar interfaz de notificaciones de ayuda en aplicación móvil
 - US20-TSK01: Creación de paciente en aplicación web
 - US20-TSK02: Mostrar la configuración de la banda

Enlace del tablero:

<https://trello.com/invite/b/67127f4727ed7ccde1bbbc90/ATTI96acb467f4e57fe2509686befd5449c2E7848092/sprint-backlog-2>

Sprint 2							
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation(Hours)	Assigned To	Status(To-do/ In-Process/ To-Review/ Done)
US006	Monitoreo de Temperatura en Tiempo Real	US06-TSK01	Toma de temperatura en embedded application	Implementar la lógica en el embedded application para tomar la temperatura	2 horas	Luis Trujillo	Done
			US06-TSK02	Implementar la interfaz de usuario de la aplicación móvil para mostrar la temperatura en tiempo real			

		US06-TSK03	Implementar la interfaz de usuario de la aplicación web para mostrar la temperatura	Implementar la interfaz de usuario de la aplicación web y lógica para mostrar la temperatura en tiempo real	3 horas	Rodrigo Raymundo	To fix
US007	Monitoreo del Ritmo Cardíaco en Tiempo Real	US07-TSK01	Toma de pulso en embedded application	Implementar la lógica en el embedded applicaton para tomar el pulso	2 horas	Luis Trujillo	Done
		T12	Implementar la interfaz de usuario de la aplicación móvil para mostrar el ritmo cardíaco	Implementar la interfaz de usuario de la aplicación móvil y lógica para mostrar el ritmo cardíaco en tiempo real	2 horas		To fix
		T13	Implementar la interfaz de usuario de la aplicación web para mostrar el ritmo cardíaco	Implementar la interfaz de usuario de la aplicación web y lógica para mostrar el ritmo cardíaco en tiempo real	2 horas	Lucía Aliaga	To fix
		US011-TSK01	Implementar el botón de emergencia en el embedded application	Implementar la lógica para el uso del botón de emergencia en el embedded application	Luis Trujillo		Done
US011	Envío de Notificación de Emergencia	US011-TSK02	Implementar notificaciones en aplicación móvil	Implementar la lógica para recibir notificaciones en la aplicación móvil	2 horas	Luis Sagastegui	Done
		US011-TSK03	Recibir notificaciones en la aplicación web	Implementar la lógica para recibir notificaciones en la aplicación web	2 horas	Rodrigo Raymundo	In Process
		US012-TSK01	Implementar interfaz de notificaciones de ayuda en aplicación móvil	Implementar interfaz de notificaciones de ayuda en aplicación móvil	3 horas	Jen Achamizo	Done
US012	Recepción de Notificaciones de Ayuda	US012-TSK02	Interfaz de notificaciones en la aplicación web	Implementar la interfaz de notificaciones en la aplicación web	3 horas	Lucía Aliaga	In Process

US015	Acceso a datos de Usuario	US15-TSK01	Implementar formulario para agregar pacientes	Implementar el formulario para poder agregar la información de los pacientes con los datos necesarios	1.5 horas	Luis Trujillo	Done
		US15-TSK02	Implementar CRUD para pacientes	Implementar las opciones para poder agregar, editar y eliminar la información de los pacientes luego de haberlos agregado	4 horas	Rodrigo Raymundo	Done
		US15-TSK03	Implementar el apartado de los pacientes	Implementar el apartado de los pacientes en donde se guarde toda la información de los pacientes	2 horas	Jesus Sagastegui	To fix
US019	Recepción de Notificaciones de Ayuda	US19-TSK01	Interfaz móvil para visualizar el estado de los sensores	Implementar la interfaz móvil y lógica para visualizar el estado de los sensores	2 horas	Luis Siancas	In Process
		US19-TSK02	Interfaz web para visualizar el estado de los sensores	Implementar la interfaz web y lógica para visualizar el estado de los sensores	3 horas	Luis Trujillo	In Process
US020	Asignación de Pulsera a un Usuario	US20-TSK01	Creación de paciente en aplicación web	Implementar la lógica en la aplicación web para crear un paciente	2 horas	Lucía Aliaga	Done
		US20-TSK02	Mostrar la configuración de la banda	Implementar la interfaz de usuario de la aplicación móvil y lógica para mostrar la configuración de la banda	2 horas	Jean Achamizo	Done
US022	Configuración de Pagos Automatizados	US22-TSK01	Implementar interfaz móvil para la configuración de pagos	Implementar la interfaz móvil y lógica para la configuración de pagos	3 horas	Luis Sagastegui	In Process
		US22-TSK02	Implementar interfaz web para la	Implementar la interfaz web y lógica para la	4 horas	Luis Siancas	In Process

			configuración de pagos	configuración de pagos			
TS07	Endpoint para Monitoreo de Temperatura	TS07-TSK01	Obtener temperatura según healthbandId	Implementar la lógica para obtener la temperatura actual de la pulsera usando el healthbandId	2 horas	Luis Siancas	Done
		TS07-TSK02	Pruebas unitarias para validar la obtención de temperatura	Crear pruebas unitarias para validar la correcta obtención de la temperatura y manejo de errores	3 horas	Rodrigo Raymundo	In Process
TS08	Endpoint para Monitoreo de Ritmo Cardíaco	TS08-TSK01	Obtener ritmo cardíaco según healthbandId	Desarrollar el servicio para recuperar el ritmo cardíaco actual de la pulsera usando el healthbandId	2 horas	Luis Trujillo	Done
		TS08-TSK02	Implementar manejo de errores	Implementar manejo de errores para situaciones como healthbandId inválido o falta de datos	0.5 horas	Luis Trujillo	In Process
TS09	Endpoint para Alertas de Salud	TS09-TSK01	Enviar alertas de salud cuando se detecten valores anormales	Implementar la funcionalidad para enviar alertas de salud cuando se detecten valores anormales	2 horas	Jean Achamizo	In Process
		TS09-TSK02	Integración del sistema de alertas	Integrar el sistema de notificaciones para enviar alertas en tiempo real a los usuarios	2 horas	Luis Siancas	In Process
TS16	Endpoints para acceso a datos del usuario	TS16-TSK01	Creación de los Endpoints CRUD para roles	Creación de los Endpoints CRUD para la creación, modificación y visualización de	2 horas	Luis Trujillo	Done
		TS09-TSK02	Creación de los Endpoints CRUD para pacientes	Creación de los Endpoints CRUD para la creación, modificación y visualización de pacientes	2.5 horas	Lucía Aliaga	Done

TS19	Endpoint para Promedio de Pulso Cardiaco	TS19-TSK01	Implementar la lógica para calcular el promedio de las últimas mediciones de pulso cardiaco	Implementar la lógica para calcular el promedio de las últimas mediciones de pulso cardiaco	1 hora	Rodrigo Raymundo	Done
TS20	Endpoint para Promedio de Temperatura	TS20-TSK01	Calcular el promedio de las últimas mediciones de temperatura	Implementar la lógica para calcular el promedio de las últimas mediciones de temperatura	1 hora	Rodrigo Raymundo	Done
TS21	Lógica para leer y almacenar mensajes de la cola	TS21-TSK01	Lectura de la cola y almacenamiento en la base de datos	Implementar logica de lectura de la cola y almacenamiento en la base de datos	3.5 horas	Luis Siancas	Done
TS22	Consultas de Cloud Api hacia Edge Api	TS22-TSK01	Creación de consultas para el endpoint del promedio del pulso cardiaco en Cloud	Creación de consultas para el endpoint del promedio del pulso cardiaco en Cloud	2 horas	Lucía Aliaga	Done
		TS22-TSK02	Creación de consultas para el endpoint del promedio de la temperatura en Cloud	ICreación de consultas para el endpoint del promedio de la temperatura en Cloud	2 horas	Lucía Aliaga	Done
		TS22-TSK03	Creación de consultas para el endpoint del pulso cardiaco actual en Cloud	Creación de consultas para el endpoint del pulso cardiaco actual en Cloud	2 horas	Luis Siancas	Done
		TS22-TSK04	Creación de consultas para el endpoint de la temperatura actual en Cloud	Creación de consultas para el endpoint de la temperatura actual en Cloud	2 horas	Luis Siancas	Done
TS23	Despliegue de la aplicación web	TS23-TSK01	Despliegue de la aplicación web	Configuración y despliegue de la aplicación web en Netlify	0.5 horas	Lucía Aliaga y Luis Trujillo	Done
TS24	Despliegue de la aplicación móvil	TS24-TSK01	Despliegue de la aplicación móvil	Configuración y despliegue de la aplicación móvil en Firebase App Distribution	1 hora	Jean Achamizo y Luis Sagastegui	Done
TS25	Despliegue del edge API	TS25-TSK01	Despliegue del Edge API	Configuración y despliegue del Edge API en Render	1 hora	Luis Siancas	Done
		TS25-	Despliegue de la	Despliegue de la	1 hora	Luis	Done

		TSK02	base de datos del Edge API	base de datos del Edge API		Siancas	
TS26	Despliegue del cloud API	TS26-TSK01	Despliegue del Edge API	Configuración y despliegue del cloud API en Render	1 hora	Luis Siancas	Done
		TS26-TSK02	Despliegue de la base de datos del cloud API	Despliegue de la base de datos del cloud API	1 hora	Luis Siancas	Done
TS27	Despliegue del embedded application	TS27-TSK01	Despliegue del embedded application	Despliegue del embedded application	0.5 horas	Luis Trujillo	Done
TS28	Endpoints para obtener los planes y suscripciones	TS28-TSK01	Creación y obtención de planes	Implementacion de logica para la creación y la obtenciones de plane	2 horas	Rodrigo Raymundo	In Process
		TS28-TSK02	Creacion y obtencion de planes en los que esta suscripto el cliente	Implementacion de logica para la creacion y obtencion de los planes en los que esta suscripto el cliente	2 horas	Rodrigo Raymundo	In Process
TS29	Generación de notificaciones	TS29-TSK01	Almacenar notificaciones generadas	Implementar logica par almacenar las notificaciones generadas	1 hora	Luis Trujillo	In Process
		TS29-TSK02	Enviar notificaciones a usuario	Crear funcionalidad para que las notificaciones sean enviadas hacia el usuario	1 hora	Luis Siancas	In Process
		TS29-TSK02	Modificacion del status de la notificacion	Crear ndpoints para la modificacion del status de la notificacion	1 hora	Luis Siancas	In Process
TS30	Configuración del dispositivo	TS30-TSK01	Configuración de dispositivos del embedded application	Implementar lógica para la configuración de los dispositivos del embedded application	2 horas	Jean Achamizo	Done
		TS30-TSK02	Implementar apartado de auditoria	Implementar apartado de auditoria para monitoreo constante	2 horas	Luis Trujillo	In Process
TS31	Generar alertas desde	TS31-TSK01	Proximidad con objetos	Implementar la logica en el	2 horas	Lucía Aliaga	Done

	el device		peligrosos en embedded application	embedded application para la proximidad con objetos peligrosos			
	TS31-TSK02	Implementar alerta frente algun accidente o parametro riesgoso	Implementar logica visual de la alerta frente algun accidente o parametro riesgoso	1 hora	Luis Trujillo	Done	
		TS31-TSK02	Generar alerta en cloud con boton de panico	Implementar funcionalidad de boton de panico que genere una alerta en cloud	1 hora	Luis Trujillo	In Process
TS32	Envío de información del device al edge API	TS32-TSK01	Envio de informacion del embedded application al edge API	Envio de informacion de los datos recogidos por el embedded application al edge AP	1 hora	Rodrigo Raymundo	Done

6.2.2.3. Development Evidence for Sprint Review

En esta sección se presentan los commits realizados en el repositorio de GitHub, donde se puede observar el trabajo colaborativo de cada integrante del equipo en el desarrollo de los productos: Web Application, Mobile Application, Edge API, Cloud API y Embedded Application.

A continuación, se detallan las contribuciones individuales en diversas ramas del proyecto, incluyendo las implementaciones de las funcionalidades principales y las integraciones necesarias para el despliegue final del producto.

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed On (Date)
IoT-SocialTech/Front-MIAM	/main	1c57dd1	Initial commit	Initial commit	26/09/2024
	/US06-Account	5906720	Feature: Account registration view completed	Feature: Account registration view completed	26/09/2024
	/US06-Account	e69fbf5	Feature: Creating fake API for testing	Feature: Creating fake API for testing	26/09/2024
	/US06-Account	0500cb3	Feature: Creating the dashboard	Feature: Creating the dashboard	26/09/2024
	/US06-Account	7e85538	Feature: Authentication Settings	Feature: Authentication Settings	26/09/2024
	/US06-Account	f2717f7	Feature: Password recovery form completed	Feature: Password recovery form completed	26/09/2024
	/development	e79d4ca	Feature: SideNav Component	Feature: SideNav Component	26/09/2024
	/US010-AlertsConfiguration	6003c15	Feature: Added band configuration interface	Feature: Added band configuration interface	23/10/2024
	/US06-US07-Temperature-Pulse-Monitoring	122aeda	Feature: Vital signs chart added	Feature: Vital signs chart added	24/10/2024
	/US008-Alerts-History	59ea9d7	Feature: Updated table	Feature: Updated table	25/10/2024

			and alerts information	and alerts information	
Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed On (Date)
IoT-SocialTech/miam-edge-api	/US020-Assign-Band	24fcecc	Feature: Patient information interface added	Feature: Patient information interface added	24/10/2024
	/US020-Assign-Band	13ec953	feat: US020 Integration with fake api	feat: US020 Integration with fake api	25/10/2024
	/US020-Assign-Band	076aee9	fix: mockup api integration	fixed mockup api integration	02/11/2024
	/TS07-TSK01	8ef9c2a	feat: added get temperature functionality	feat: added functionality to return current patient temperature and included shared files	20/10/2024
	/TS07-TSK01	923d98f	feat: added JPA configuration and established connection with database	feat: added JPA configuration and established connection with database	20/10/2024
	/TS07-TSK01	49996fa	Merge pull request #2 from IoT-SocialTech/jpa-conf	Added JPA configuration	20/10/2024
	/TS07-TSK01	695887f	feat: added JMS configuration and established connection with IBM MQ queues	feat: added JMS configuration and established connection with IBM MQ queues	20/10/2024
	/TS07-TSK01	d5a5e9e	feat: added JMS configuration and established connection with IBM MQ queues	feat: added JMS configuration and established connection with IBM MQ queues	20/10/2024
	/TS07-TSK01	2cf7dc7	Initial commit: added base files	Initial commit: added base files	30/09/2024
	/TS08-TSK01	4bd451d	feat: added get heart rate functionality	feat: added functionality to return current patient heart rate	20/10/2024
	/TS08-TSK01	1306aa3	feat: added get temperature functionality	feat: added functionality to return current patient temperature and included shared files	20/10/2024
	/TS08-TSK01	0f970fb	Merge pull request #4 from IoT-SocialTech/TS08-TSK01	feat: added get heart rate functionality	20/10/2024
	/TS19-TSK01	5ab3960	feat: added get average heart rate functionality	feat: added functionality to return average patient heart rate	20/10/2024
	/TS19-TSK01	684cdfe	Merge pull request #7 from IoT-SocialTech/TS19-TSK01	feat: added get average heart rate functionality	20/10/2024
	/TS19-TSK01	4dd6029	feat: added get average heart rate functionality	feat: added functionality to return average patient heart rate	20/10/2024
	/TS20-TSK01	e1b80c7	feat: added get average temperature functionality	feat: added functionality to return average patient temperature	20/10/2024
	/TS20-TSK01	3797c40	fix: error	fix: error	20/10/2024
	/TS20-TSK01	59385b5	Merge remote-tracking branch 'origin/develop' into develop	Conflicts resolved in MetricsController.java	20/10/2024
	/TS20-TSK01	d7aeee0	feat: added get average heart rate functionality	feat: added functionality to return average patient heart rate	20/10/2024
	/TS20-TSK01	b43eed1	feat: added get average heart rate functionality	feat: added functionality to return average patient heart rate	20/10/2024
	/TS21-	5d79f14	feat: added functionality to store	feat: added functionality to store	20/10/2024

TSK01		device measurements in the database	device measurements in the database	
/TS21-TSK01	5291ed4	Merge pull request #9 from IoT-SocialTech/TS20-TSK01	feat: added get average temperature functionality	20/10/2024
/TS21-TSK01	e1b80c7	feat: added get average temperature functionality	feat: added functionality to return average patient temperature	20/10/2024
/TS21-TSK01	03ddda9c	Merge pull request #10 from IoT-SocialTech/TS21-TSK01	feat: added functionality to store device measurements in the database	20/10/2024
/develop	03ddda9c	Merge pull request #10 from IoT-SocialTech/TS21-TSK01	feat: added functionality to store device measurements in the database	20/10/2024
/develop	5d79f14	feat: added functionality to store device measurements in the database	feat: added functionality to store device measurements in the database	20/10/2024
/develop	5291ed4	Merge pull request #9 from IoT-SocialTech/TS20-TSK01	feat: added get average temperature functionality	20/10/2024
/develop	e1b80c7	feat: added get average temperature functionality	feat: added functionality to return average patient temperature	20/10/2024
/develop	3797c40	fix: error	fix: error	20/10/2024
/develop	59385b5	Merge remote-tracking branch 'origin/develop' into develop	Conflicts resolved in MetricsController.java	20/10/2024

Implementación Movie IoT-SocialTech mobile-application-miam

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed On (Date)
IoT-SocialTech/mobile-application-miam	/ts24-mobile-app-implementation	2af1951	feat: seach dependence for change name	Búsqueda de dependencias para cambiar el nombre	01/11/2024
	/ts24-mobile-app-implementation	34a4200	feat: change name for integrate firebase	Cambio de nombre para integración con Firebase	01/11/2024
	/ts24-mobile-app-implementation	469c2eb	feat: add dependence for change name application:change_app_package_name	Agregar dependencia para cambio de nombre de la aplicación: change_app_package_name	01/11/2024
	/ts24-mobile-app-implementation	3286872	Merge pull request #3 from IoT-SocialTech/us012-reception-notifications	Integración de notificaciones de recepción	31/10/2024
	/ts24-mobile-app-implementation	b5e7960	feat: add API logic	Añadir lógica de API	31/10/2024
	/ts24-mobile-app-implementation	3831621	fix: routing in BottomNavigationBarItem	Corregir la ruta en BottomNavigationBarItem	27/10/2024
	/ts24-mobile-app-implementation	3013760	feat: add BottomNavigationBarItem	Añadir BottomNavigationBarItem	27/10/2024
	/ts24-mobile-app-implementation	97ceabe	Merge pull request #2 from IoT-SocialTech/us012-reception-notifications	Integración de notificaciones de recepción	25/10/2024
	/ts24-mobile-app-implementation	5d678bd	feat-tsk01: implementation of interfast notifications	Implementación de notificaciones de interfaz	25/10/2024

/us020-assignment-of-bracelets	6e213c8	fix: fix login screen, configure main screen	Corrección de la pantalla de inicio de sesión y configuración de la pantalla principal	01/11/2024
/us020-assignment-of-bracelets	8b107f4	fix: screen configuration, The cards are modified to display it differently	Configuración de pantalla, las tarjetas se modifican para mostrarse de manera diferente	31/10/2024
/us020-assignment-of-bracelets	0eea0f7	feat: screen configuration, add connection	Configuración de pantalla, se añade conexión	31/10/2024
/us020-assignment-of-bracelets	10a8874	feat: add configuration screen	Añadir pantalla de configuración	31/10/2024
/us020-assignment-of-bracelets	3286872	Merge pull request #3 from IoT-SocialTech/us012-reception-notifications	Integración de notificaciones de recepción	31/10/2024
/us020-assignment-of-bracelets	b5e7960	feat: add API logic	Añadir lógica de API	31/10/2024

6.2.2.4. Testing Suite Evidence for Sprint Review

En esta sección se presenta la evidencia de la suite de pruebas utilizada para la revisión del sprint. Para asegurar la funcionalidad y el cumplimiento de los criterios de aceptación de cada historia de usuario (US) en esta entrega, se implementaron pruebas automáticas con el uso de Gherkin. Cada historia de usuario se describe en lenguaje Gherkin

Repositorio: <https://github.com/IoT-SocialTech/features>

TS21_LogicaLeerAlmacenarMensajesCola.feature

```

● ● ●

1 Feature: Lógica para leer y almacenar mensajes de la cola
2
3 Scenario: Lectura y almacenamiento de mensajes de la cola
4   Given el sistema recibe mensajes en la cola
5   When se implementa la lógica para leer los mensajes
6   Then el sistema almacena los mensajes leídos en la base de datos
7
8 Scenario: Error al leer o almacenar mensajes de la cola
9   Given se intenta leer mensajes de la cola
10  When el sistema detecta un fallo en la lógica de procesamiento
11  Then se registra un error indicando el problema en la lectura o almacenamiento
12 o

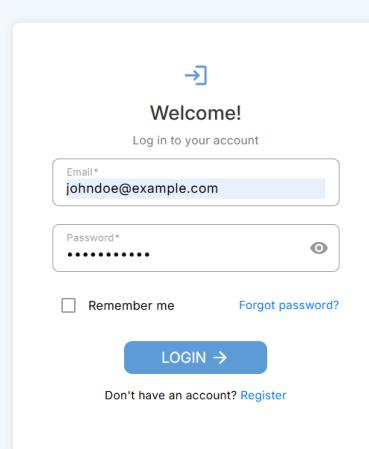
```

US007_MonitoringCardiacRhythm.feature

```
1 Feature: Monitoreo del Ritmo Cardíaco en Tiempo Real
2
3 Scenario: Monitoreo de ritmo cardíaco en tiempo real
4     Given la pulsera está en uso
5     When el ritmo cardíaco es monitoreado
6     Then el sistema muestra el valor actual y envía una alerta si está fuera del rango saludable
7
8 Scenario: Fallo en el sensor de ritmo cardíaco
9     Given el sensor de ritmo cardíaco no funciona correctamente
10    When el sistema detecta un fallo
11    Then se muestra un mensaje indicando el problema
12
```

6.2.2.5. Execution Evidence for Sprint Review**Web Application**

En este Sprint, logramos implementar varias vistas clave en la aplicación web, enfocándonos en mejorar la experiencia de configuración, gestión y monitoreo de los pacientes y sus datos de salud. A continuación, se detalla cada vista y su funcionalidad implementada:



- **Band Configuration:** Desarrollamos la vista para configurar la health band de cada paciente. Esta sección permite seleccionar y ajustar las alertas que recibirán el caregiver y el relative, personalizando así las notificaciones y el tipo de seguimiento que recibirán según las necesidades del paciente.

Band Configuration

ID: 1 Patient: John Doe ID: 2 Patient: Alice Smith

Battery: 80% Last Sync: 2024-09-18 18:05 Sync now

Patient
Name: John

Caregiver
Assigned to: Carolina Suarez
Alerts:
 Alert if temperature is below 35 °C or higher than 38 °C
 Alert if pulse is below 60 bpm or higher than 100 bpm
 Alert me if fall risk is high

Relative
Related to: Carolina Suarez Remove relative

Alerts:
 Alert if temperature is below _____ °C or higher than _____ °C
 Alert if pulse is below _____ bpm or higher than _____ bpm
 Alert me if fall risk is high

- **Patients:** Implementamos la vista de creación y gestión de pacientes, donde se pueden registrar nuevos pacientes y vincularlos tanto a un caregiver como a un relative. Esta funcionalidad permite gestionar de forma centralizada la información relevante de cada paciente.

Patients Add new patient

Patient: John Doe Name: John Last Name: Doe
Birth Date: 1950-04-20 Age: 74 ID:1

Patient: Alice Smith

Caregiver: Not configured

Relative: Jane Doe (Spouse)

Medication Alerts

Medication	Dose	Next Dose	Actions
Aspirin	100 mg	08:00:00	Edit Delete
Ibuprofen	200 mg	12:00:00	Edit Delete

Delete patient

This action will delete all the patient's data.

Save

- **Alerts:** En esta sección, añadimos un historial de alertas para cada paciente, donde se pueden visualizar las notificaciones pasadas y su información detallada, permitiendo un seguimiento preciso de los eventos críticos que han ocurrido en cada caso.

Date	Hour	Patient	Alert Type	Description	Caregiver	Actions
2024-09-24	09:49 AM	John Doe	High risk of falling	Increased risk of falling	Carolina Suarez	i
2024-09-24	08:52 AM	Alice Smith	High temperature	Current temperature: 38.5°C	Carolina Suarez	i

Items per page: 5 1 – 2 of 2

Alert Details

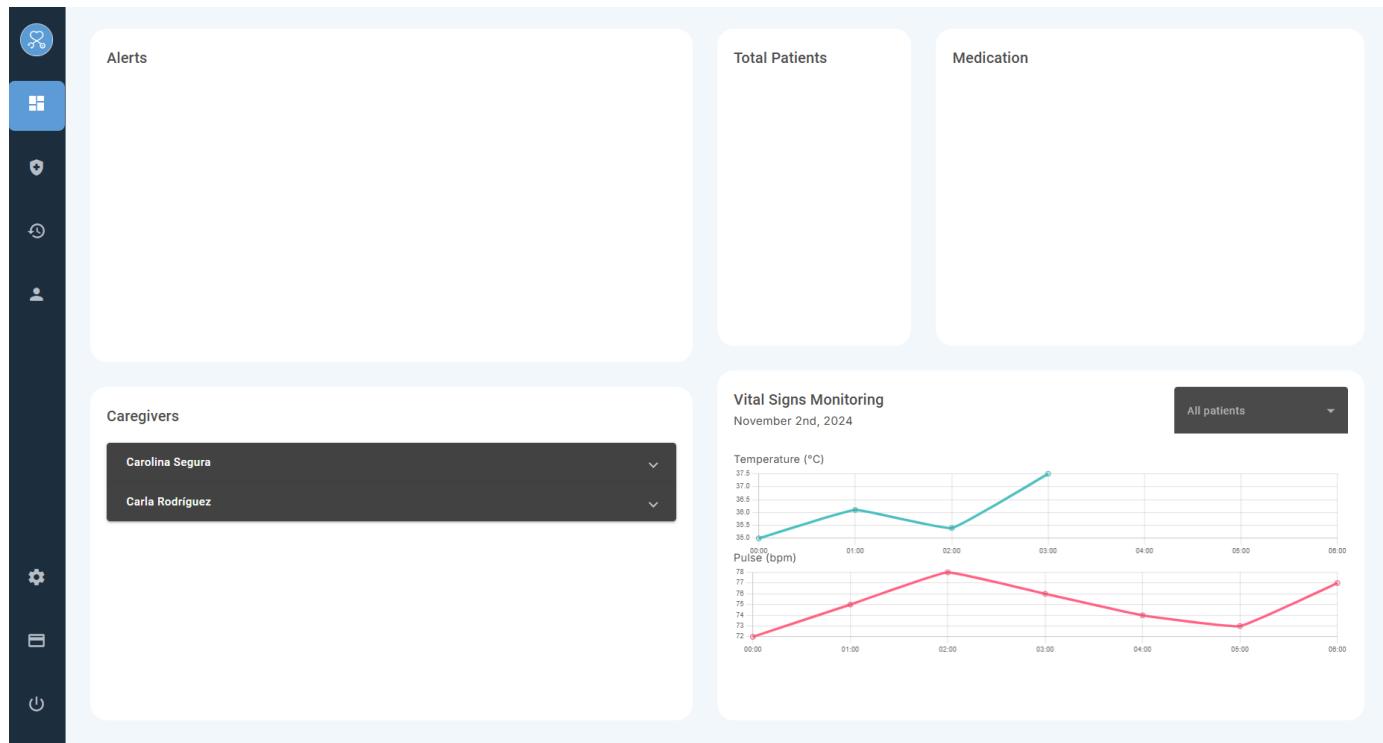
Date: 2024-09-24
Hour: 09:49 AM
Patient: John Doe
Alert Type: High risk of falling
Description: Increased risk of falling
Caregiver: Carolina Suarez

Attention Hour: 2024-09-24 10:00 AM
Time Until Attended: 11 minutes
Caregiver Actions: Patient attended with no incidents
Notes: Patient is feeling dizzy but resting now

[Ok](#)

Items per page: 5 1 – 2 of 2

- **Dashboard:** Avanzamos en el desarrollo del Dashboard, incluyendo gráficos de temperatura y pulso para facilitar el monitoreo de estos parámetros en tiempo real. También se añadió la visualización de los caregivers y sus pacientes.

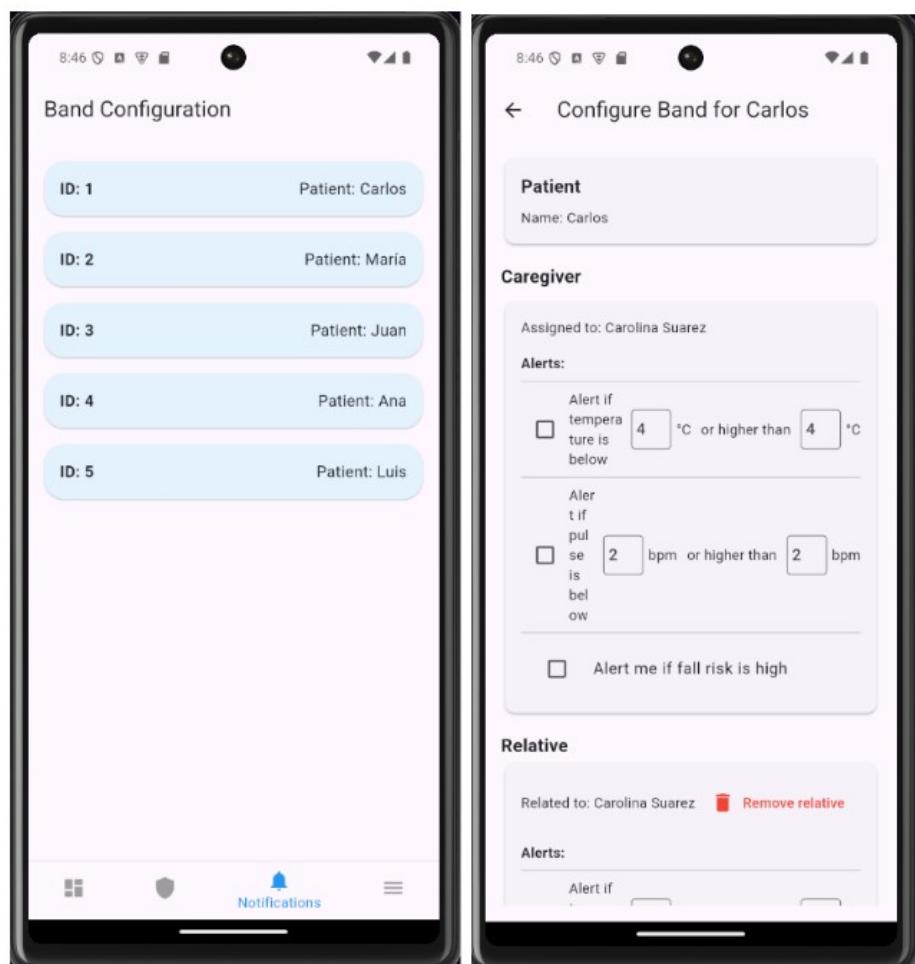
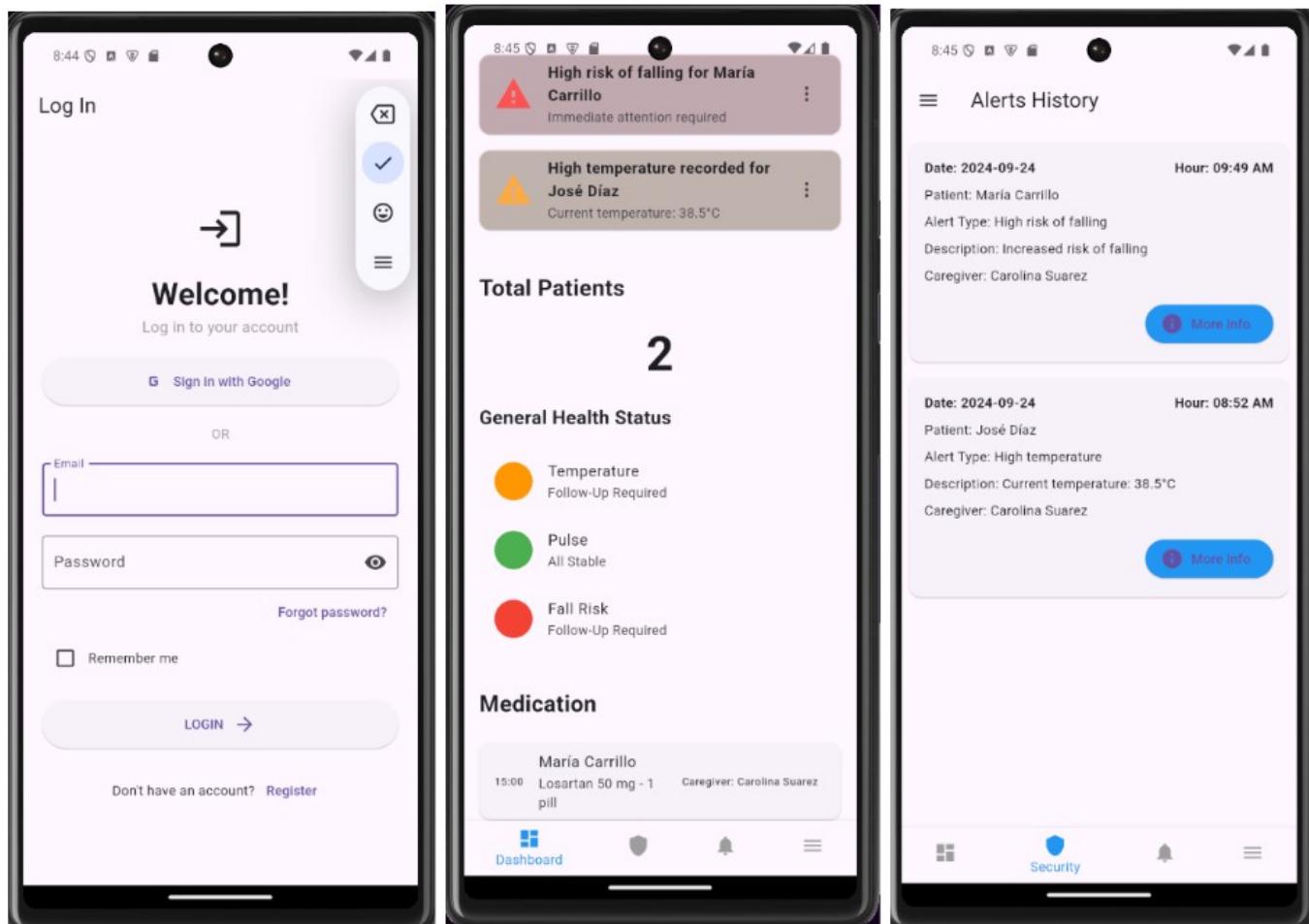


Enlace al video:

<https://goo.su/orFzn6>

Mobile Application

Para la realización de la aplicación móvil se abarcaron varios User Stories, para este Sprint se realizó la pantalla de Login en donde el usuario podrá crearse una cuenta o entrar con su cuenta de Google. Al ingresar a la aplicación, podremos visualizar como primera instancia la creación del Dashboard, en donde se visualiza varios datos importantes de los pacientes como su medicación y el monitoreo de las señales vitales. También se pudo implementar de manera correcta las Alertas de los pacientes en caso suceda algo fuera de lo habitual. La pantalla que más consideramos importante es el Band Configuration. Donde se muestra una lista de Pacientes y en donde se puede gestionar las alertas en base a la temperatura o el pulso del paciente. Para el siguiente Sprint hemos avanzado en distintos User Stories que aún faltan corregir ya que siguen con problemas mínimos, pero consideramos que es un gran avance para este Sprint.



Enlace al video:

https://upcdeupe-my.sharepoint.com/:v/g/personal/u202014249_upc_edu_pe/ETL5B6zfuiNJv6oH_idvWjMBmnbbkiGAgobXI4jsWpWPQSA?e=bB71qy&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiTdHJIYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IlldIYlsInJIZmVycmFsTW9kZSI6InZpZXcifX0%3D

6.2.2.6. Services Documentation Evidence for Sprint Review

En este Sprint, avanzamos en la documentación de los Endpoints desarrollados, asegurando claridad y facilidad de integración mediante OpenAPI. A continuación, se presenta una tabla con las acciones soportadas para cada Endpoint, incluyendo los métodos HTTP, parámetros, ejemplos de llamadas y respuestas.

6.2.2.7. Software Deployment Evidence for Sprint Review

Web Application

Durante este Sprint se realizó el despliegue de la aplicación web en Netlify. Este proceso incluyó la creación y configuración de los recursos necesarios en la plataforma, asegurando la disponibilidad y accesibilidad de la aplicación en un entorno de producción.

Para llevar a cabo el despliegue en Netlify, se realizaron los siguientes pasos:

1. Configuración de la Cuenta y Proyecto en Netlify: Se accedió a una cuenta ya existente y se configuró el repositorio con el código fuente de la aplicación web.
2. Conexión del Repositorio: Se configuró la rama `development` para el despliegue.
3. Configuración de las Opciones de Build y Deployment: Se configuraron las opciones de build en Netlify, especificando el comando de compilación
4. Ejecución del Despliegue Inicial: Una vez configurados los detalles de compilación, Netlify ejecuta automáticamente el primer despliegue. En este paso, Netlify toma el código del repositorio, lo construye y publica la aplicación en un dominio temporal. Posteriormente se configuró el dominio.

Deploys for miam-site

- https://miam-site.netlify.app

Published development@4749359

Deploys

Deploy	Date	Duration
Production: development@4749359 Published	Today at 1:34 AM	Deployed in 1m
Merge pull request #6 from IoT-SocialTech/US020-Assign-Band		
Deploy Preview #6: US020-Assign-Band@076aee9	Today at 1:34 AM	Deployed in 54s
fix: mockup api integration		
Production: development@703ebf3	Oct 25 at 2:33 PM	Deployed in 52s
Merge pull request #5 from IoT-SocialTech/US020-Assign-Band		
Deploy Preview #5: US020-Assign-Band@13ec953	Oct 25 at 12:58 PM	Deployed in 48s
feat: US020 Integration with fake api		
Production: development@c7766fd	Oct 24 at 11:18 PM	Deployed in 51s
Merge pull request #4 from IoT-SocialTech/US020-Assign-Band		
Deploy Preview #4: US020-Assign-Band@24fcecc	Oct 24 at 11:18 PM	Deployed in 49s
Feature: Patient information interface added		
Production: development@404ebce	Oct 24 at 9:57 AM	Deployed in 39s
Merge pull request #3 from IoT-SocialTech/US008-Alerts-History		
Deploy Preview #3: US008-Alerts-History@59ea9d7	Oct 24 at 9:57 AM	Deployed in 41s
Feature: Updated table and alerts information		
Production: development@2df9b0f	Oct 24 at 8:26 AM	Deployed in 46s
Fix: deployment issues		

Published deploy for miam-site

Today at 1:34 AM

Production: development@4749359

Test your site's Lighthouse performance

Want to see how your site will perform before you deploy? Install the Lighthouse plugin for build-time Lighthouse scores and reports. [Learn more](#)

[Install Lighthouse plugin](#)

Deploy summary

- Built using the Angular Runtime
Netlify auto-detected Angular and used the [Angular Runtime](#) to build and deploy your site. [Learn more about deploying with Angular on Netlify](#)
- 1 plugin ran successfully
Select for details.
- Build time: 1m. Total deploy time: 1m
Build started at 1:36:02 AM and ended at 1:37:02 AM. [Learn more about build minutes](#)

Deploy log

Step	Status
Initializing	Complete
Building	Complete
Deploying	Complete
Cleanup	Complete

Enlace: <https://miam-site.netlify.app/>

Mobile Application

Durante este Sprint, se realizó el despliegue de la aplicación móvil utilizando Firebase App Distribution. Este proceso incluyó la configuración y publicación de la aplicación, asegurando que los usuarios de prueba y los equipos de desarrollo pudieran acceder a la versión más reciente en un entorno de pruebas controlado.

Para llevar a cabo el despliegue en Firebase App Distribution, se siguieron los siguientes pasos:

- Configuración de la Cuenta y Proyecto en Firebase: Se accedió a una cuenta ya existente en Firebase y se configuró el proyecto para el despliegue de la aplicación móvil.

Crear un proyecto

Comencemos con el nombre de tu proyecto[?]

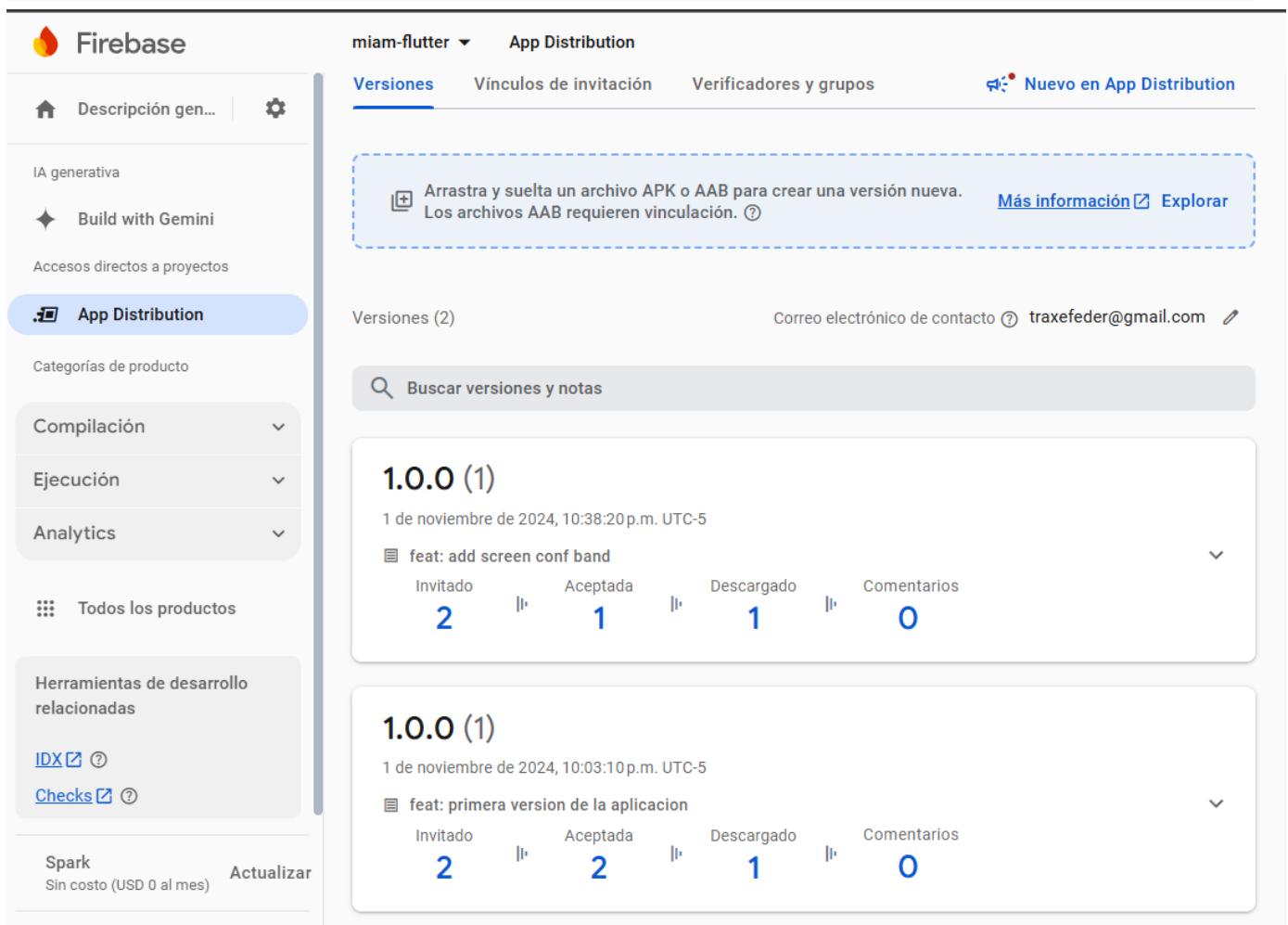
Nombre del proyecto

miam-flutter

 miam-flutter-90aef

Ya tienes un proyecto de Google Cloud?
[Agregar Firebase al proyecto de Google Cloud](#)

Continuar



The screenshot shows the Firebase App Distribution dashboard for the project "miam-flutter". On the left sidebar, under "App Distribution", there are sections for "Descripción gen..." (General description), "IA generativa" (Generative AI), "Build with Gemini", "Accesos directos a proyectos" (Direct links to projects), and "Categorías de producto" (Product categories) which includes "Compilación", "Ejecución", and "Analytics". Below these are "Herramientas de desarrollo relacionadas" (Related development tools) with links to "IDX" and "Checks". At the bottom, it shows "Spark" with a note "Sin costo (USD 0 al mes)" and an "Actualizar" (Update) button.

The main content area shows "App Distribution" for "miam-flutter". It has tabs for "Versiones" (selected), "Vínculos de invitación", and "Verificadores y grupos". A blue button "Nuevo en App Distribution" is visible. A dashed box highlights a placeholder for dropping an APK or AAB file.

Under "Versiones", there are two entries:

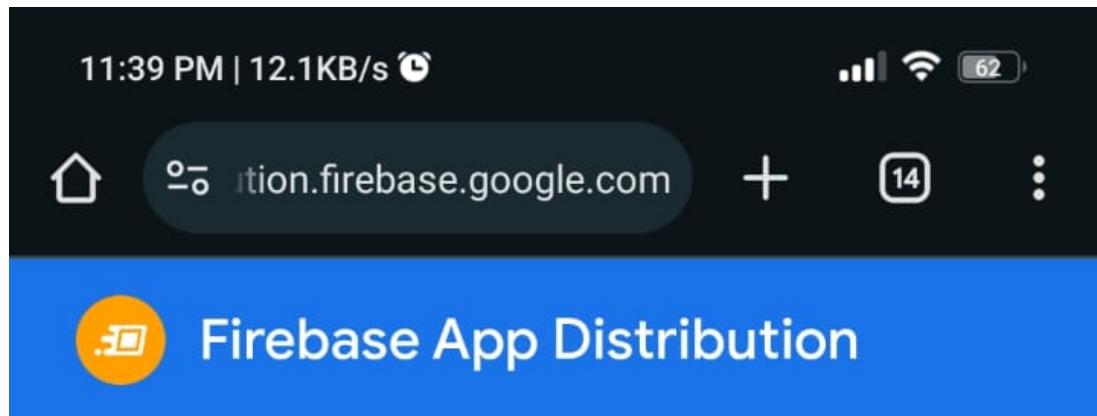
- 1.0.0 (1)** - Last updated on November 1, 2024, at 10:38:20 p.m. UTC-5.
 - feat: add screen conf band
 - Invitado: 2
 - Aceptada: 1
 - Descargado: 1
 - Comentarios: 0
- 1.0.0 (1)** - Last updated on November 1, 2024, at 10:03:10 p.m. UTC-5.
 - feat: primera version de la aplicacion
 - Invitado: 2
 - Aceptada: 2
 - Descargado: 1
 - Comentarios: 0

2. Integración del Repositorio de Código: Se configuró la rama de desarrollo en el repositorio para generar automáticamente builds de la aplicación y enviarlas a Firebase.

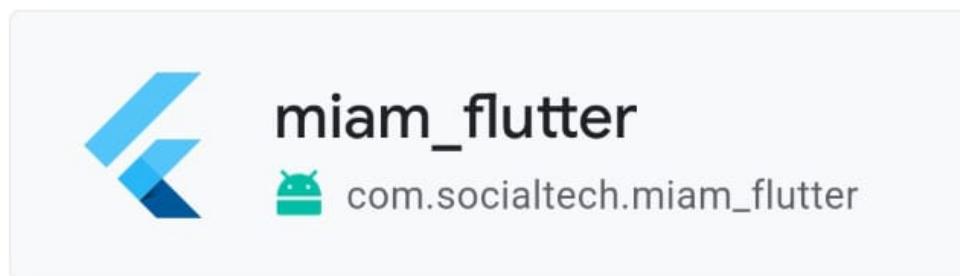
3. Configuración de las Opciones de Build: En el archivo de configuración, se especificaron los detalles de compilación para que la aplicación se genere correctamente antes de ser distribuida. Este proceso incluyó la configuración de los scripts necesarios para la integración continua.

Enlace: <https://appdistribution.firebaseio.dev/i/bc9e72d1b2b27dd6>

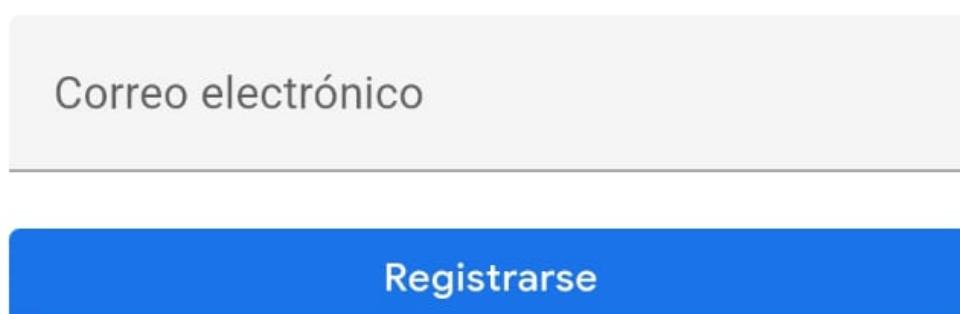
4. Ejecución del Despliegue a Firebase App Distribution: Una vez que se completó la configuración, el proceso de CI/CD envió automáticamente el build a Firebase App Distribution. Desde allí, los usuarios de prueba y el equipo de desarrollo recibieron invitaciones por correo electrónico para descargar e instalar la aplicación en sus dispositivos.



¿Todo listo para probar esta
aplicación?



Regístrate para compartir tu dirección de correo
electrónico **gmail.com** con el desarrollador y
obtener acceso:





Miam Edge API

A screenshot of a web browser displaying the MIAM API documentation using the Swagger UI. The title bar shows the URL "miam-edge-api.onrender.com/swagger-ui/index.html#/". The main header reads "MIAM API 1.0.0 OAS 3.0". Below the header, there's a "Servers" section with a dropdown menu set to "https://miam-edge-api.onrender.com - Generated server url". The "Metrics" section contains four GET requests: "/api/v1/miam-edge-api/metrics/temperature" (Get temperature), "/api/v1/miam-edge-api/metrics/heartRate" (Get heart rate), "/api/v1/miam-edge-api/metrics/averageTemperature" (Get average temperature), and "/api/v1/miam-edge-api/metrics/averageHeartRate" (Get average heart rate). The "Schemas" section lists three items: "ErrorMessage", "ApiResponseTemperatureResponseDto", and "TemperatureResponseDto". The interface has a clean, modern design with a white background and blue highlights for interactive elements.

Miam Cloud API

The screenshot shows the MIAM API documentation generated by Swagger. It includes sections for the Role Controller API, PatientCaregiver API, and Account API. Each section lists methods (GET, POST, PUT, DELETE) with their corresponding URLs and descriptions.

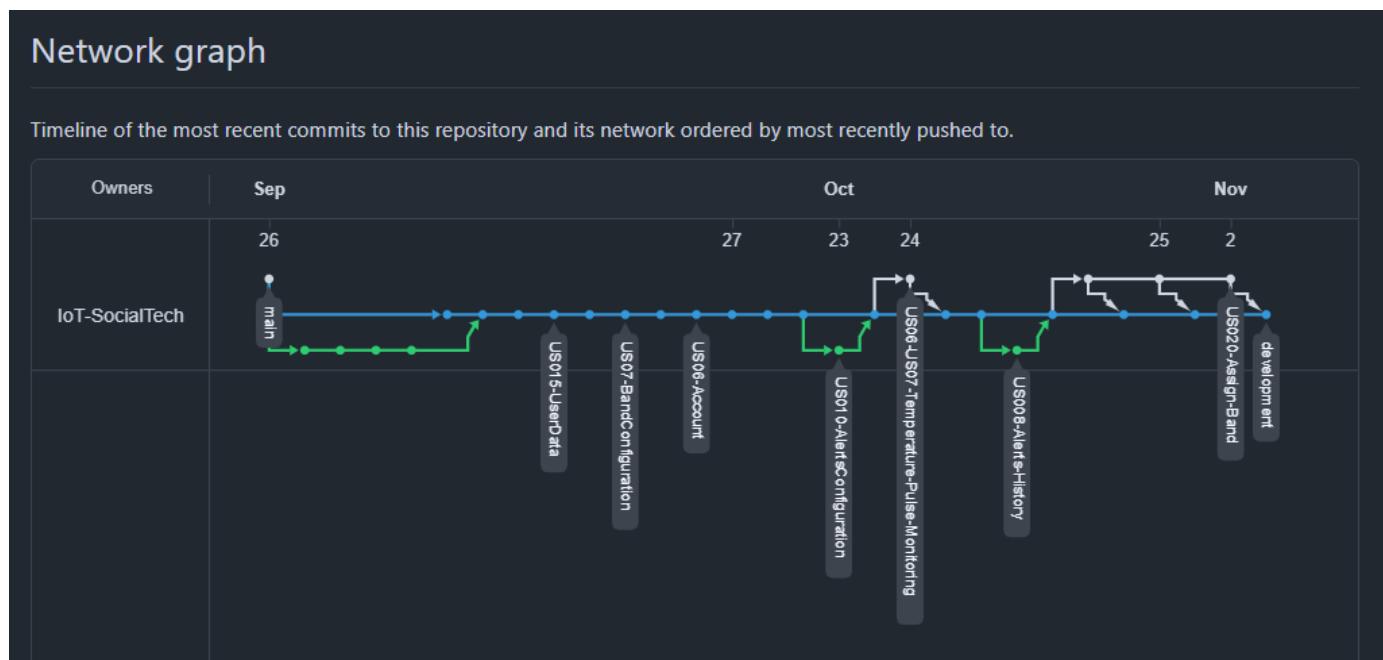
- Role Controller** (Role Controller API)
 - GET /api/v1/miam-cloud-api/roles/{id} Get role by id
 - PUT /api/v1/miam-cloud-api/roles/{id} Update role
 - DELETE /api/v1/miam-cloud-api/roles/{id} Delete role
 - POST /api/v1/miam-cloud-api/roles Create role
- PatientCaregiver** (PatientCaregiver API)
 - GET /api/v1/miam/cloudApi/patientCaregivers/{patientId}/{caregiverId} get a patient caregiver by id
- Account** (Account API)
 - GET /api/v1/miam/cloudApi/accounts/{id} get a account by id
 - PUT /api/v1/miam/cloudApi/accounts/{id} update an existing account

6.2.2.8. Team Collaboration Insights during Sprint

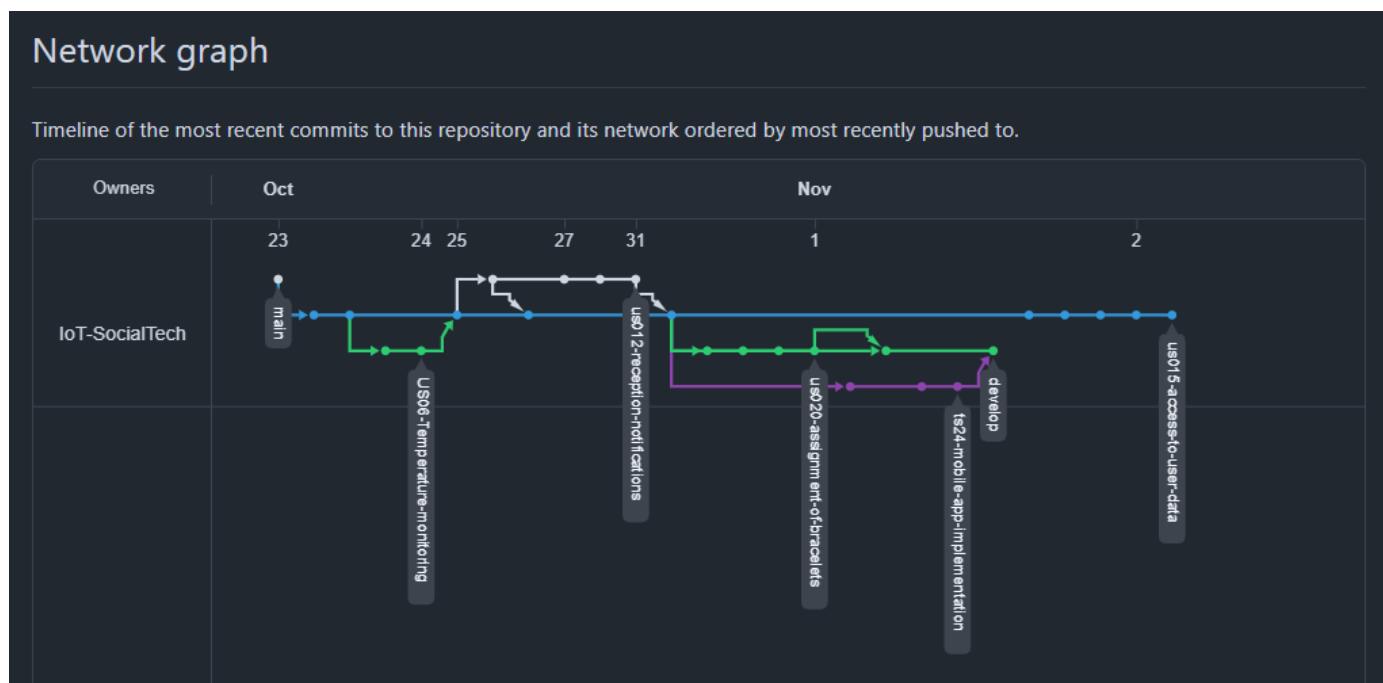
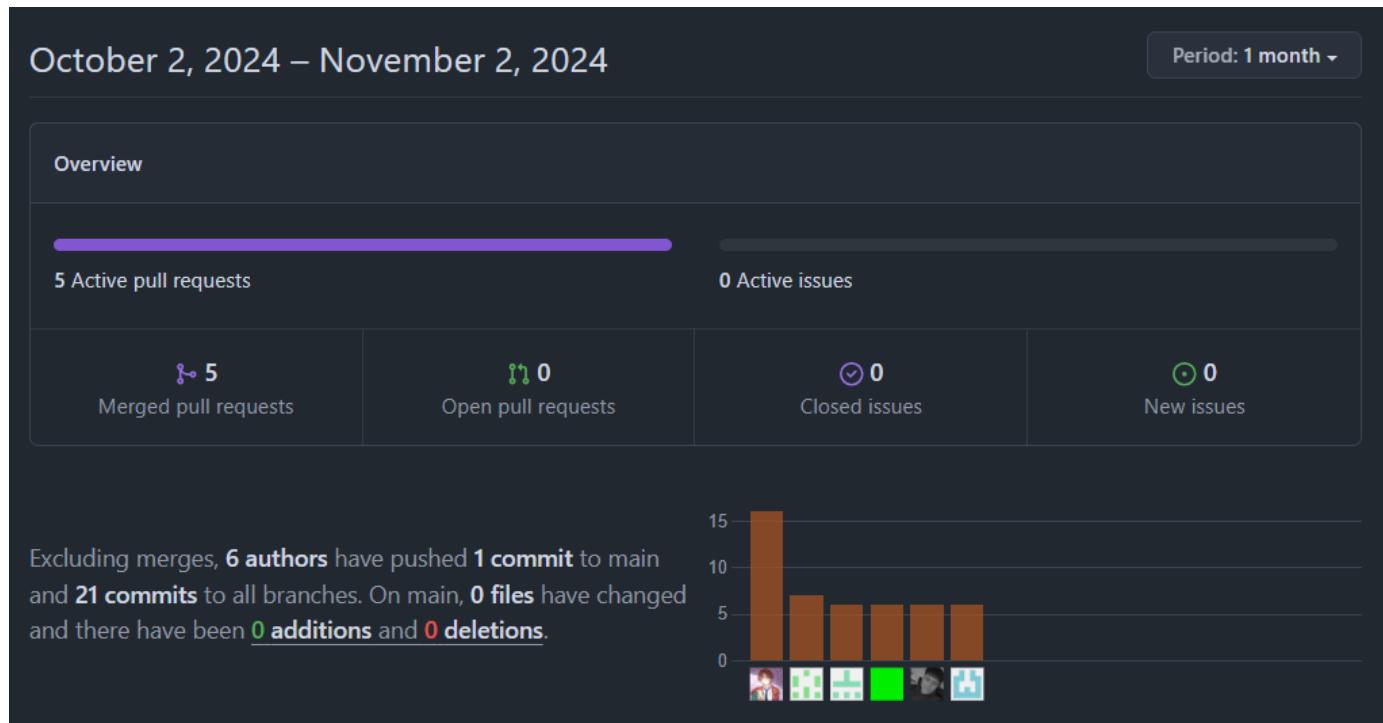
En este Sprint, el equipo trabajó de manera colaborativa en la implementación de los distintos componentes del proyecto, abarcando los Web Services y las Aplicaciones. La contribución de cada miembro quedó reflejada en la actividad de commits y los analíticos de colaboración en GitHub, los cuales se presentan en esta sección con capturas de pantalla.

Para cada producto, se detalla la participación de los miembros del equipo, quienes han contribuido en la implementación de funcionalidades clave.

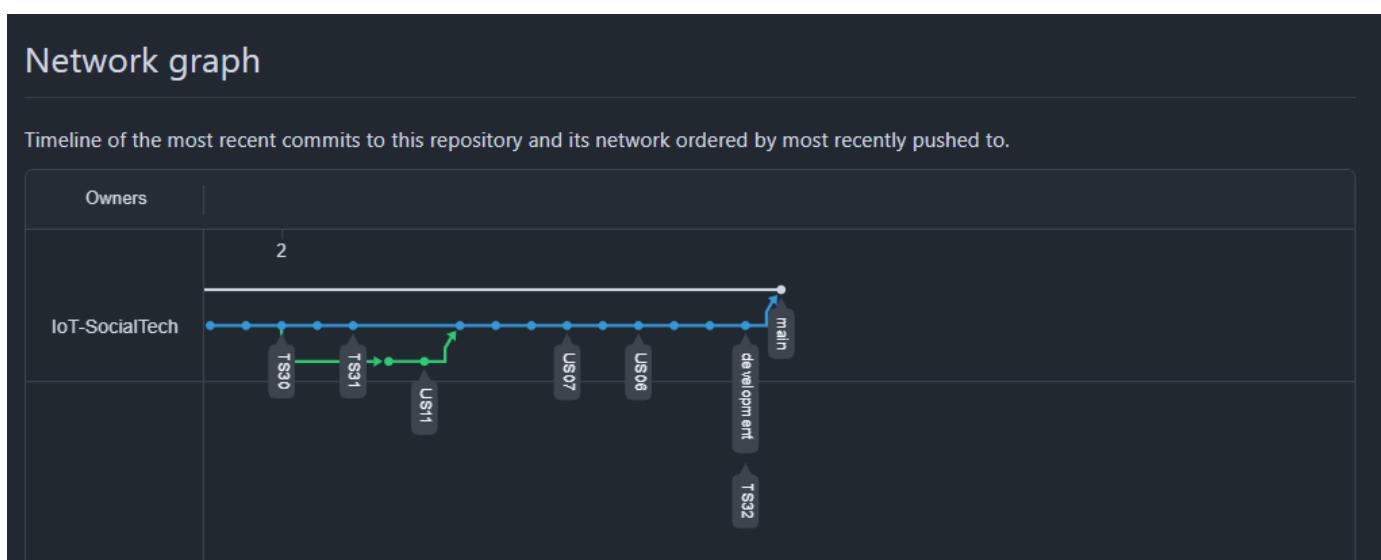
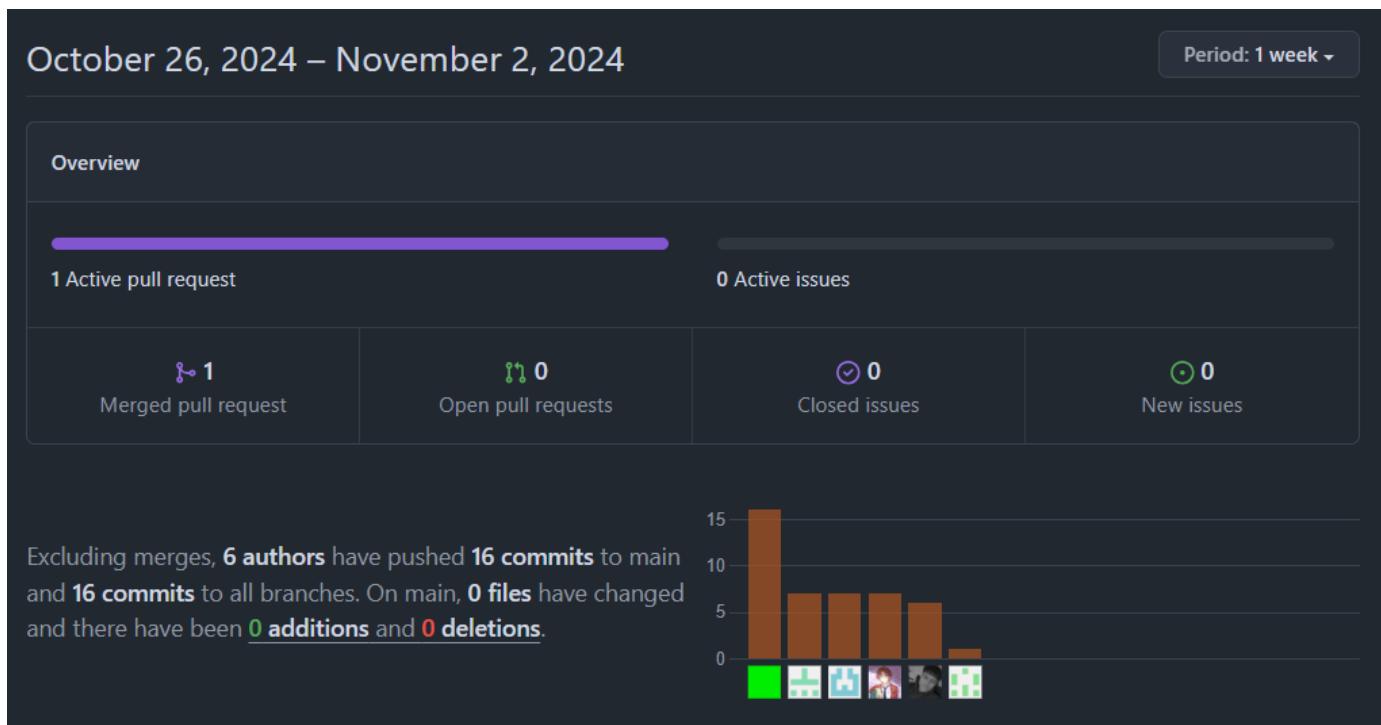
Web Application



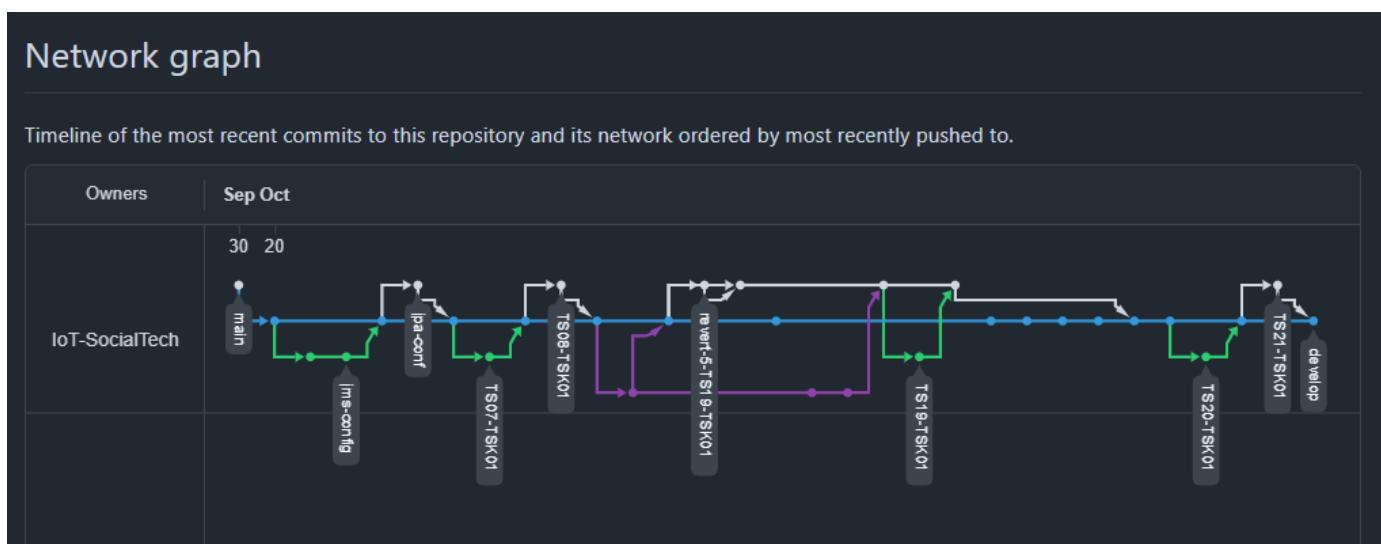
Mobile Application



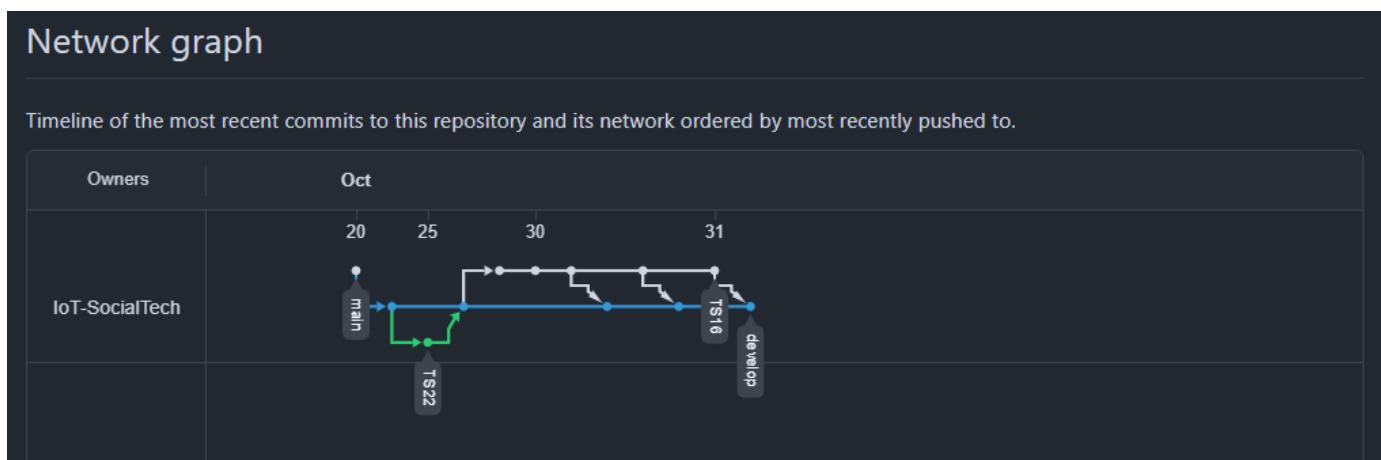
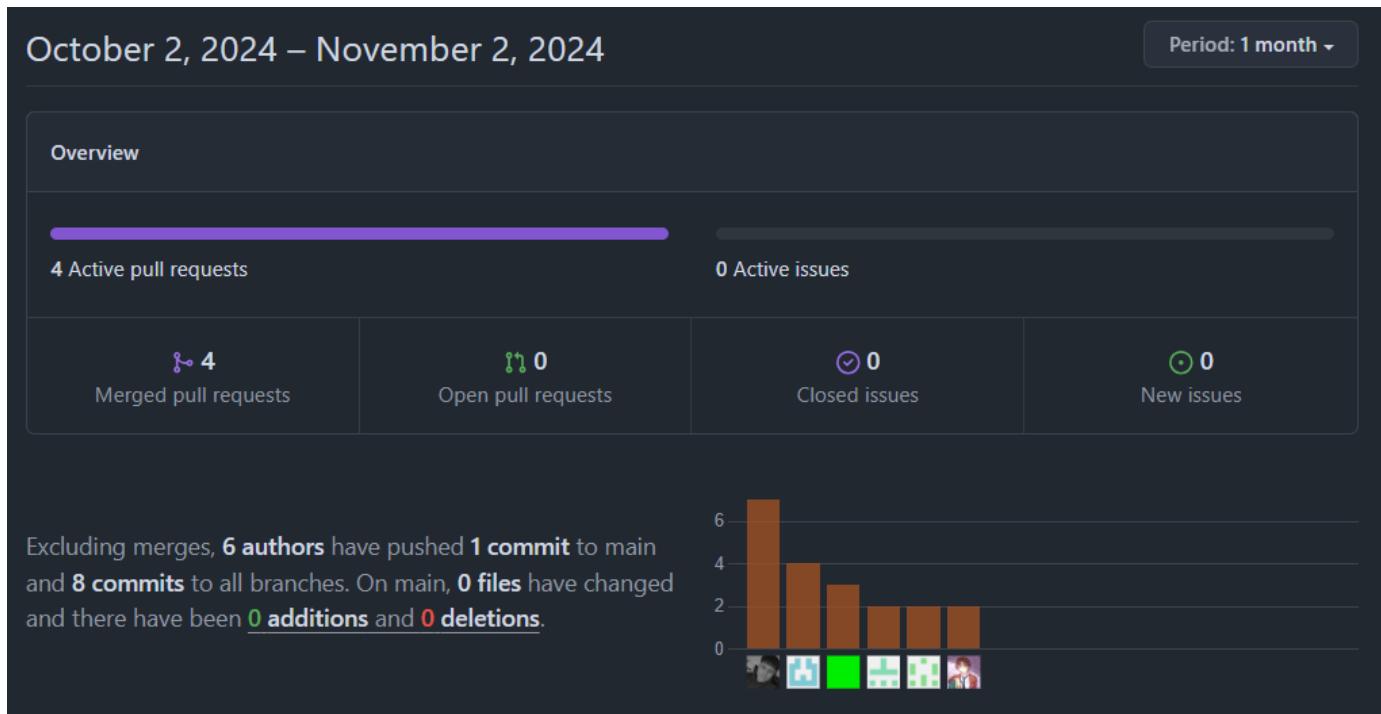
Embedded Application



Edge API



Cloud API



6.3 Validation Interviews.

6.3.1. Diseño de Entrevistas.

Para Cuidadores de Adultos Mayores (Caregivers)

¿Cuáles son los mayores desafíos que enfrentas al monitorear los signos vitales de los adultos mayores a tu cargo?

¿Con qué frecuencia monitorean la temperatura y el pulso de tus pacientes? ¿Cuál es tu método actual para registrar estos datos?

¿Qué tan importante sería para ti tener acceso a estos datos en tiempo real?

¿Qué valor le darías a una alerta inmediata si los signos vitales de un paciente están fuera de los rangos normales?

¿Cómo prefieres acceder a los datos de los pacientes (móvil, tablet, computadora)?

¿Qué problemas o preocupaciones te gustaría que esta aplicación te ayudara a resolver?

¿Te resultaría útil una opción para revisar los historiales de signos vitales de los pacientes en la aplicación? ¿Cómo la usarías en tu rutina?

¿Qué características crees que le faltarían a una herramienta que solo monitorea pulso y temperatura para que sea completa?

¿Hay otras métricas o información sobre el paciente que te gustaría poder rastrear o visualizar desde la misma plataforma?

¿Cuáles son tus expectativas en cuanto a la facilidad de uso de la aplicación? ¿Prefieres una interfaz más simple o más detallada?

Para Dueños de Casas de Reposo (Nursing Home Owners)

¿Cuáles son los principales problemas que experimentan los cuidadores de tu institución al monitorear los signos vitales de los residentes?

¿Qué métodos utilizan actualmente para llevar el control de los signos vitales y cómo se almacenan estos datos?

¿Consideras importante tener acceso centralizado y en tiempo real a la información de salud de tus residentes?

¿Qué impacto crees que tendría la capacidad de recibir alertas sobre signos vitales en el bienestar de los pacientes y en la eficiencia del personal?

¿Cómo se integraría una aplicación de este tipo en la rutina diaria de tu casa de reposo?

¿Cuáles son tus preocupaciones en cuanto a la privacidad y seguridad de los datos de los residentes?

¿Consideras importante que esta información esté disponible en una plataforma web además de la móvil? ¿Por qué?

¿Qué tipo de reportes o análisis te gustaría recibir sobre el estado de salud de tus residentes?

¿Qué funcionalidades crees que serían esenciales en una herramienta de monitoreo de salud para una casa de reposo?

¿Qué tan dispuestos estarían a adoptar esta tecnología si mejora la calidad de atención y facilita el trabajo de los cuidadores?

6.3.2. Registro de Entrevistas.

6.3.3. Evaluaciones según heurísticas.

6.4. Video About-the-Product.

En esta sección se incluye un video promocional que ofrece un primer vistazo a nuestro producto. El video está diseñado para comunicar de manera atractiva y concisa el valor que ofrece nuestro producto, mencionando el público objetivo y beneficios.



CON NUESTRO
SISTEMA DE
MONITOREO
INTELIGENTE DEL
ADULTO MAYOR

No tendrás que preocuparte más por ello

Enlace al video:

<https://goo.su/KISALD2>

Conclusiones

El sistema de pulseras IoT contribuye significativamente a la mejora de la calidad de vida de las personas mayores al ofrecerles una solución de monitoreo constante de sus signos vitales. A través de la detección en tiempo real de anomalías en la temperatura o el ritmo cardíaco, los cuidadores pueden reaccionar rápidamente ante posibles emergencias, lo que proporciona una mayor sensación de seguridad tanto para los usuarios como para sus familiares.

El proyecto tiene un impacto social positivo al ofrecer una solución tecnológica que permite a las personas mayores vivir de manera más independiente, mientras que los cuidadores y familiares pueden estar tranquilos al saber que las personas bajo su cuidado están siendo monitoreadas constantemente. Además, el sistema fomenta la prevención y rápida respuesta ante emergencias, reduciendo el riesgo de incidentes graves y mejorando la seguridad general de los usuarios.

Bibliografía

Dorri, S., Zabolinezhad, H., & Sattari, M. (2023). *The application of Internet of Things for the elderly health safety: A systematic review*. Advances in Biomedical Research, 12, 109. https://doi.org/10.4103/abr.abr_197_22

Giulianelli, D., De Luca, G., De Luca, S., García, G., Carnuccio, E., Valiente, W., & Volker, M. (2017). *Diseño de sistema IoT de monitoreo y alarma para personas mayores*. Repositorio institucional de la UNLP, Universidad Nacional de La Plata.
<https://sedici.unlp.edu.ar/handle/10915/62406>

Ministerio de Salud. (2018, 13 de diciembre). *Uno de cada tres adultos mayores de 65 años sufre una caída*.

<https://www.gob.pe/institucion/minsa/noticias/23629-uno-de-cada-tres-adultos-mayores-de-65-anos-sufre-una-caida>

Suárez, G., Velasco, V., Limones, M., Reyes, H., & Delgado, V. (2020). *Caídas en el adulto mayor y factores de riesgo*. European Journal of Child Development, Education and Psychopathology, 8(1), 47–56. <https://doi.org/10.30552/ejpad.v8i1.130>

Anexos

Videos de Exposición

Enlace	Fecha	Entregable
https://goo.su/sorVJws	07/09/2024	TB1
https://goo.su/8Zz305	26/09/2024	TP
https://goo.su/tYkZtN	02/11/2024	TB2

Diagrama C4 MIAM

<https://structurizr.com/share/95614/diagrams#SystemContext>

6.2.2. Sprint 3

6.2.2.1. Sprint Planning 3

Sprint #	Sprint 3
Sprint Planning Background	
Date	14/10/2024
Time	20:30
Location	Modalidad remota por Discord
Prepared By	Aliaga Trevejo, Lucía Guadalupe
Attendees (to planning meeting)	<ul style="list-style-type: none"> - Achamizo Huamani, Jean Carlos - Aliaga Trevejo, Lucía Guadalupe - Raymundo Guevara, Rodrigo Alejandro - Siancas Reategui, Luis Alberto

	<p>- Trujillo Lopez, Luis Alberto - Sagastegui Rodriguez, Luis Jesus</p>
Sprint 2 Review Summary	En el sprint anterior, se completó la mayoría de las características fundamentales para la creación de usuarios y pacientes en el sistema de monitoreo.
Sprint 2 Retrospective Summary	-
Sprint Goal & User Stories	
Sprint 2 Goal	-
Sprint 1 Velocity	-
Sum of Story Points	-

6.2.2.2. Sprint Backlog 3

En esta sección se especifican los detalles del Sprint Backlog, que es una lista de tareas que se han realizado para completar el Sprint 2.

The screenshot shows a Trello board with the following structure:

- Goal:** Our focus is on enabling users to generate detailed health reports, view real-time notifications and alerts, and manage their subscription plans. We believe this enhances the user experience by providing important insights into the health data, immediate awareness of system alerts, and flexibility in managing their subscription. This will be achieved so that users can successfully generate reports, view notifications and alerts in real time, and update their subscription plan using the implemented endpoints.
- Stories:**
 - US008: Alertas de Salud
 - US009: Historial de Datos de Salud
 - US010: Configuración de Alertas Personalizadas
 - US013: Gestión de Notificaciones
 - US14: Configuración de Alertas de Emergencia
 - US16: Gestión de Pulseras Alquiladas
 - US17: Visualización de Pulseras Disponibles
- To Do:**
 - US028-TSK01: Implementar la interfaz web para implementar las adaptaciones de visualización
 - US028-TSK02: Implementar la interfaz móvil para implementar las adaptaciones de visualización
 - TSK01: Elaboración del prototipo
- In Progress:**
 - US008-TSK03: Implementar la interfaz en la aplicación móvil para la visualización de alertas
 - US008-TSK04: Implementar los mensajes de error en la aplicación móvil
 - US009-TSK01: Implementar filtro de datos por fecha en la aplicación web
 - US009-TSK02: Implementar filtro de datos por fecha en la aplicación móvil
 - US013-TSK01: Implementar la interfaz web para configurar las alertas
 - US013-TSK02: Implementar la interfaz en la aplicación móvil para la visualización de alertas
 - US014-TSK01: Implementar la interfaz web para configurar las alertas
 - US017-TSK01: Implementar la interfaz web para listar las pulseras disponibles
 - US030-TSK01: Implementar la interfaz en la aplicación web que
- To Review:**
 - US013-TSK03: Implementar la interfaz en la aplicación web para la visualización de alertas
 - US013-TSK04: Implementar la lógica en la aplicación web para mostrar el ritmo cardíaco
 - US025-TSK01: Implementar la lógica en la aplicación web para mostrar la temperatura
 - US025-TSK02: Implementar la interfaz en la aplicación móvil para mostrar la temperatura
 - US025-TSK03: Implementar mensaje de error en la aplicación web
 - US025-TSK04: Implementar mensaje de error en la aplicación móvil
 - US027-TSK01: Implementar la interfaz web para mostrar de forma clara las alertas críticas
 - US008-TSK02: Implementar la interfaz en la aplicación web para alertas
- To Fix:**
 - US025-TSK05: Implementar la interfaz móvil para configurar el umbral de ritmo cardíaco
 - US025-TSK06: Implementar la interfaz web para configurar el umbral de temperatura
 - US027-TSK02: Implementar la interfaz móvil para configurar el umbral de temperatura
- Done:**
 - US026-TSK01: Implementar barra de navegación en la aplicación web
 - US026-TSK02: Implementar barra de navegación en la aplicación móvil
 - US026-TSK03: Implementar mensaje de error en la aplicación móvil
 - US027-TSK01: Implementar la interfaz móvil para configurar el umbral de temperatura
 - US027-TSK02: Implementar la interfaz web para configurar el umbral de temperatura

Enlace del tablero:

<https://trello.com/invite/b/67127f4727ed7ccde1bbbc90/ATTI96acb467f4e57fe2509686befd5449c2E7848092/sprint-backlog-2>

Sprint #		Sprint 3					
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation(Hours)	Assigned To	Status(To-do/ In-Process/ To-Review/ Done)
US008	Alertas de Salud	US08-TSK01	Implementar la interfaz en la aplicación web para alertas	Implementar la interfaz en la aplicación web para visualizar las alertas recibidas	- horas	abc	To Fix
			US08-	Implementar	Implementar	abc	To fix

		TSK02	los mensajes de error en la aplicacion web	los mensajes de error en la aplicacion web cuando el dispositivo no esta vinculado			
		US08-TSK03	Implementar la interfaz en la aplicacion movil para alertas	Implementar la interfaz en la aplicacion movil para visualizar las alertas recibidas	- horas	abc	In Process
		US08-TSK04	Implementar los mensajes de error en la aplicacion movil	Implementar los mensajes de error en la aplicacion movil cuando el dispositivo no esta vinculado	- horas	abc	In Process
US009	Historial de Datos de Salud	US09-TSK01	Implementar filtro de datos por fecha en la aplicacion web	Implementar el filtrado de datos de temperatura y ritmo cardiaco por fecha en la aplicacion web	-	-	In Process
		US09-TSK02	Implementar filtro de datos por fecha en la aplicacion movil	Implementar el filtrado de datos de temperatura y ritmo cardiaco por fecha en la aplicacion movil	- horas	-	In Process
US010	Configuración de Alertas Personalizadas	US010-TSK01	Implementar la interfaz web para configurar el umbral de temperatura	Implementar la interfaz en la aplicacion web para configurar el umbral de temperatura	- horas	-	Done
		US010-TSK02	Implementar la interfaz movil para configurar el umbral de temperatura	Implementar la interfaz en la aplicacion movil para configurar el umbral de temperatura	- horas	-	Done
		US010-TSK03	Implementar la interfaz web para configurar el umbral de ritmo cardiaco	Implementar la interfaz en la aplicacion web para configurar el umbral de ritmo cardiaco	- horas	-	Done
		US010-TSK04	Implementar la interfaz movil para	Implementar la interfaz en la aplicacion	- horas	-	Done

			configurar el umbral de ritmo cardiaco	movil para configurar el umbral de ritmo cardiaco			
US013	Gestión de Notificaciones	US013-TSK01	Implementar la interfaz en la aplicacion web para la visualizacion de alertas	Implementar la interfaz en la aplicacion web para la visualizacion de alertas e informacion adicional	- horas	-	To Review
		US013-TSK02	Implementar la interfaz en la aplicacion movil para la visualizacion de alertas	Implementar la interfaz en la aplicacion movil para la visualizacion de alertas e informacion adicional	- horas	-	To Review
US014	Configuración de Alertas de Emergencia	US14-TSK01	Implementar la interfaz movil para configurar las alertas	Implementar la interfaz en la aplicacion movil para configurar las notificaciones de alertas de emergencia	- horas	-	To Review
		US14-TSK02	Implementar la interfaz web para configurar las alertas	Implementar la interfaz en la aplicacion web para configurar las notificaciones de alertas de emergencia	- horas	-	To Review
US016	Gestión de Pulseras Alquiladas	US16-TSK01	Implementar la logica en la aplicacion web para cambiar el estado de una pulsera	Implementar la logica en la aplicacion web para cambiar el estado de una pulsera	- horas	-	To Review
		US16-TSK02	Implementar la logica en la aplicacion web para cambiar el estado de una pulsera	Implementar la logica en la aplicacion web para cambiar el estado de una pulsera	- horas	-	To Review
US017	Visualización de Pulseras Disponibles	US17-TSK01	Implementar la interfaz web para listar las pulseras disponibles	Implementar la interfaz y logica en la aplicacion web para listar las pulseras disponibles	- horas	-	In Process
		US17-TSK02	Implementar la interfaz	Implementar la interfaz y	- horas	-	To Review

			movil para listar las pulseras disponibles	logica en la aplicacion movil para listar las pulseras disponibles			
US018	Generación de Informes de Uso	US18-TSK01	Implementar la logica en la aplicacion web para generar un informe de uso	Implementar la logica en la aplicacion web para generar un informe de uso	- horas	-	To Review
		US18-TSK02	Implementar la interfaz en la aplicacion web para mostrar un informe de uso	Implementar la interfaz en la aplicacion web para mostrar un informe de uso	- horas	-	To Review
		US18-TSK03	Implementar la logica en la aplicacion movil para generar un informe de uso	Implementar la logica en la aplicacion movil para generar un informe de uso	- horas	-	To Review
		US18-TSK04	Implementar la interfaz en la aplicacion movil para mostrar un informe de uso	Implementar la interfaz en la aplicacion movil para mostrar un informe de uso	- horas	-	To Review
US021	Generación de Informes Financieros	US021-TSK01	Implementar la interfaz web y logica para generar un informe financiero	Implementar la interfaz en la aplicacion web y logica para generar un informe financiero	- horas	-	In Process
		US021-TSK02	Implementar la interfaz movil y logica para generar un informe financiero	Implementar la interfaz en la aplicacion movil y logica para generar un informe financiero	- horas	-	In Process
US025	Acceso Rápido a Datos de la Pulsera	US025-TSK01	Implementar la logica en la aplicacion web para mostrar la temperatura	Implementar la logica en la interfaz de la aplicacion web para mostrar el ultimo registro de temperatura del paciente	- horas	-	To Fix
		US025-TSK02	Implementar la logica en la aplicacion web para mostrar	Implementar la logica en la interfaz de la aplicacion web para mostrar el	- horas	-	To Fix

		el ritmo cardiaco	ultimo registro del ritmo cardiaco del paciente				
	US025-TSK03	Implementar mensaje de error en la aplicacion web	Implementar mensaje de error en la aplicacion web cuando se pierde la conexion con la banda	- horas	-	To Fix	
	US025-TSK04	Implementar mensaje de error en la aplicacion movil	Implementar mensaje de error en la aplicacion movil cuando se pierde la conexion con la banda	- horas	-	To Fix	
US026	Navegación Intuitiva	US026-TSK01	Implementar barra de navegacion en la aplicacion web	Implementar la barra de navegacion y redireccion de rutas en la aplicacion web	- horas	-	Done
		US026-TSK02	Implementar barra de navegacion en la aplicacion movil	Implementar la barra de navegacion y redireccion de rutas en la aplicacion movil	- horas	-	Done
US027	Visualización Clara de Alertas	US027-TSK01	Implementar la interfaz web para mostrar de manera clara las alertas criticas	Implementar la interfaz web para mostrar de manera clara las alertas criticas	- horas	-	To Review
		US027-TSK02	Implementar el mensaje de error de las alertas criticas en la aplicacion web	Implementar el mensaje de error de las alertas criticas en la aplicacion web	- horas	-	In Process
		US027-TSK03	Implementar la interfaz móvil para mostrar de manera clara las alertas criticas	Implementar la interfaz móvil para mostrar de manera clara las alertas criticas	- horas	-	To Review
		US027-TSK04	Implementar el mensaje de error de las alertas criticas en la	Implementar el mensaje de error de las alertas criticas	- horas	-	In Process

			aplicacion movil	en la aplicacion movil			
US028	Personalización de la Interfaz	US028-TSK01	Implementar la interfaz web para implementar las adaptaciones de visualizacion	Implementar la interfaz web para implementar las adaptaciones de visualizacion	- horas	-	To Do
		US028-TSK02	Implementar la interfaz movil para implementar las adaptaciones de visualizacion	Implementar la interfaz movil para implementar las adaptaciones de visualizacion	- horas	-	To Do
US029	Notificaciones en Tiempo Real	US029-TSK01	Implementar las notificaciones instantaneas en la aplicacion web	Implementar las notificaciones instantaneas en la aplicacion web	- horas	-	In Process
		US029-TSK02	Implementar el registro y reenvío de la notificación instantanea en caso de fallos de la aplicación web	Implementar el registro y reenvío de la notificación instantanea en caso de fallos de la aplicación web	- horas	-	In Process
		US029-TSK03	Implementar las notificaciones instantaneas en la aplicacion movil	Implementar las notificaciones instantaneas en la aplicacion movil	- horas	-	In Process
		US029-TSK04	Implementar el registro y reenvío de la notificación instantánea en caso de fallos de la aplicación movil	Implementar el registro y reenvío de la notificación instantánea en caso de fallos de la aplicación movil	- horas	-	In Process
US030	Resumen Diario de Salud	US030-TSK01	Implementar la interfaz de la aplicacion web que muestre el resumen del dia	Implementar la interfaz de la aplicacion web que muestre el resumen del dia	- horas	-	In Process
		US030-TSK02	Implementar el mensaje de	Implementar el mensaje de	- horas	-	In Process

		error del resumen del dia en la aplicacion web	error del resumen del dia en la aplicacion web			
	US030-TSK03	Implementar la interfaz de la aplicacion movil que muestre el resumen del dia	Implementar la interfaz de la aplicacion movil que muestre el resumen del dia	- horas	-	In Process
	US030-TSK04	Implementar el mensaje de error del resumen del dia en la aplicacion movil	Implementar el mensaje de error del resumen del dia en la aplicacion movil	- horas	-	In Process