

Informe del Trabajo Final

Universidad Peruana de Ciencias Aplicadas



Área: Ingeniería de Software

Curso: Desarrollo de Soluciones IoT

Código del curso: 1ASI0572

NRC: 3428

Docente: Marco Antonio León Baca

Startup: AlertaVital

Producto: AlertaVital

Team members:

Nombre	Código
Bohorquez Lerzundi, Gerardo Sebastián	u202224149
Huanaco Huayta, Elizabeth Lucero	u20211g522
Huapaya Cuevas, Anthony	u202220235
Silva Tirado, Sebastian Valentino	u20211g296
Uribe Quispe, Jesús Guillermo	u202221876

Ciclo 2025-20

Registro de Versiones del Informe

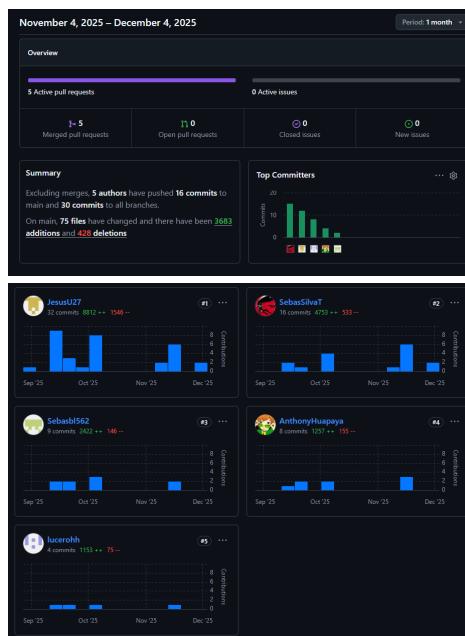
El objetivo de esta sección es resumir las modificaciones relevantes que se realizan al informe durante el ciclo de vida del proyecto. Esta sección inicia en una página nueva y se incluye un cuadro con la siguiente estructura:

Versión	Fecha	Autor	Descripción de modificación
TB1	21/09/2025	Jesús Guillermo Uribe Quispe	Se añadió el capítulo I, II, III y IV del proyecto.
TP	10/10/2025	Jesús Guillermo Uribe Quispe	UX/UI Design, Software Development and Sprint 1
TB2	16/11/2025	Jesús Guillermo Uribe Quispe	Sprint 2, Validation Interviews and Final Corrections
TF	4/12/2025	Jesús Guillermo Uribe Quispe	Sprint 3, Hardware Implementation

Project Report Collaboration Insights

URL del repositorio para el reporte del proyecto: <https://github.com/TF-SolucionesIoT/report>

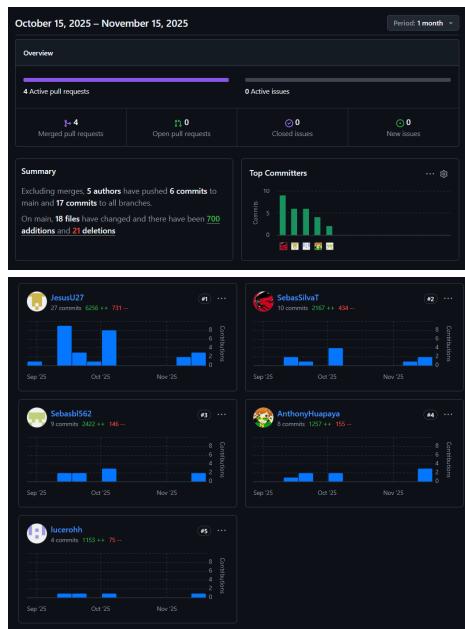
TF:



Para el desarrollo del informe perteneciente a la entrega del trabajo final, se dividió la implementación de secciones de la siguiente forma para cada integrante del equipo:

Integrante	Tareas Asignadas
Bohórquez Lerezundi, Gerardo Sebastián	Despliegue servicio web y aplicación web, desarrollo de funcionalidades en servicios web
Huanaco Huayta, Elizabeth Lucero	desarrollo de implementación y programación del hardware.
Huapaya Cuevas, Anthony	desarrollo de implementación y programación del hardware.
Silva Tirado, Sebastián Valentino	Desarrollo de aplicación web y servicios web.
Uribe Quispe, Jesús Guillermo	Desarrollo de servicio web, arquitectura y aplicación edge.

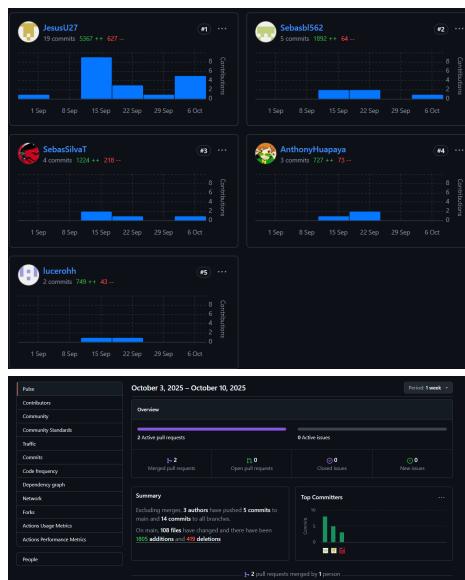
TB2:



Para el desarrollo del informe perteneciente a la entrega del trabajo parcial, se dividió la implementación de secciones de la siguiente forma para cada integrante del equipo:

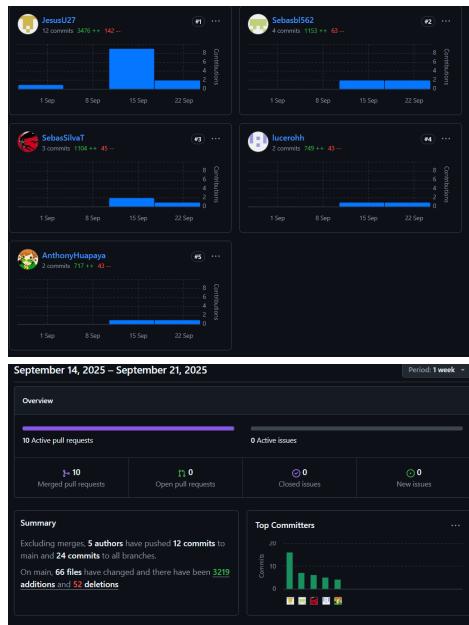
Integrante	Tareas Asignadas
Bohórquez Lerzundi, Gerardo Sebastián	Despliegue servicio web y aplicación web, desarrollo de funcionalidades en servicios web
Huanaco Huayta, Elizabeth Lucero	Validation Interviews, desarrollo de funcionalidades en servicios web
Huapaya Cuevas, Anthony	Validation Interviews, desarrollo de funcionalidades en servicios web
Silva Tirado, Sebastián Valentino	Desarrollo de aplicación web.
Uribe Quispe, Jesús Guillermo	Desarrollo de servicio web y aplicación edge.

TP:



Para el desarrollo del informe perteneciente a la entrega del trabajo parcial, se dividió la implementación de secciones de la siguiente forma para cada integrante del equipo:

Integrante	Tareas Asignadas
Bohórquez Lerzundi, Gerardo Sebastián	Solution UX / UI Design, Information Architecture, Systems
Huanaco Huayta, Elizabeth Lucero	UX/UI Mobile Design
Huapaya Cuevas, Anthony	UX/UI Mobile Design
Silva Tirado, Sebastián Valentino	Corrección de errores, Desarrollo App Web, UX / UI Web App Design
Uribe Quispe, Jesús Guillermo	Desarrollo Landing Page, UX/UI Design

TB1

Para el desarrollo del informe perteneciente a la entrega TB1, se dividió la implementación de secciones de la siguiente forma para cada integrante del equipo:

Integrante	Tareas Asignadas
Bohórquez Lerezundi, Gerardo Sebastián	Analisis competitivo, documentación de Bounded context, entrevistas
Huanaco Huayta, Elizabeth Lucero	Lean UX process, entrevistas, documentación de Bounded context, entrevistas
Huapaya Cuevas, Anthony	Lean UX process, entrevistas, documentación de Bounded context, entrevistas
Silva Tirado, Sebastián Valentino	Antecedentes y problemática, entrevistas, documentación de Bounded context, entrevistas, Impact Mapping
Uribe Quispe, Jesús Guillermo	User Stories, Product Backlog, EventStorming process, Software architecture

Github Collaboration Insights

Github también presenta un timeline de las ramas principales y los procesos de merge a los que se han sometido. Todas las ramas se crearon tomando en cuenta el diseño de GitFlow para una buena organización cuando se usa un software de control de versiones.

Los integrantes son:

- Bohórquez Lerezundi, Gerardo Sebastián (Sebasl562)
- Huanaco Huayta, Elizabeth Lucero (lucerooh)
- Huapaya Cuevas, Anthony (AnthonyHuapaya)
- Silva Tirado, Sebastián Valentino (SebasSilvaT)
- Uribe Quispe, Jesús Guillermo(JesusU27)

Contenido

1. Capítulo I: Introducción.

- 1.1. Startup Profile.
 - 1.1.1. Descripción del startup.
 - 1.1.2. Perfiles de los 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.
- 2. **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.
 - 2.4. Big Picture EventStorming
 - 2.5. Ubiquitous Language
- 3. **Capítulo III: Requirements Specification.**
 - 3.1. User Stories.
 - 3.2. Impact Mapping.
 - 3.3. Product Backlog.
- 4. **Capítulo IV: Solution Software Design.**
 - 4.1. Strategic-Level Domain-Driven Design.
 - 4.1.1. Design-Level EventStorming.
 - 4.1.1.1. Candidate Context Discovery.
 - 4.1.1.2. Domain Message Flows Modeling.
 - 4.1.1.3. Bounded Context Canvases.
 - 4.1.2. Context Mapping.
 - 4.1.3. Software Architecture.
 - 4.1.3.1. Software Architecture System Landscape Diagram.
 - 4.1.3.2. Software Architecture Context Level Diagrams.
 - 4.1.3.3. 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: IAM.
 - 4.2.1.1. Domain Layer.
 - 4.2.1.2. Interface Layer.
 - 4.2.1.3. Application Layer.
 - 4.2.1.4. Infrastructure Layer.
 - 4.2.1.5. Bounded Context Software Architecture Component Level Diagrams.
 - 4.2.1.6. Bounded Context Software Architecture Code Level Diagrams.
 - 4.2.1.6.1. Bounded Context Domain Layer Class Diagrams.
 - 4.2.1.6.2. Bounded Context Database Design Diagram.
 - 4.2.2. Bounded Context: Profile.
 - 4.2.2.1. Domain Layer.
 - 4.2.2.2. Interface Layer.
 - 4.2.2.3. Application Layer.
 - 4.2.2.4. Infrastructure Layer.
 - 4.2.2.5. Bounded Context Software Architecture Component Level Diagrams.
 - 4.2.2.6. Bounded Context Software Architecture Code Level Diagrams.
 - 4.2.2.6.1. Bounded Context Domain Layer Class Diagrams.
 - 4.2.2.6.2. Bounded Context Database Design Diagram.
 - 4.2.3. Bounded Context: Health Tracking.
 - 4.2.3.1. Domain Layer.
 - 4.2.3.2. Interface Layer.
 - 4.2.3.3. Application Layer.
 - 4.2.3.4. Infrastructure Layer.
 - 4.2.3.5. Bounded Context Software Architecture Component Level Diagrams.
 - 4.2.3.6. Bounded Context Software Architecture Code Level Diagrams.
 - 4.2.3.6.1. Bounded Context Domain Layer Class Diagrams.
 - 4.2.3.6.2. Bounded Context Database Design Diagram.
 - 4.2.4. Bounded Context: Monitoring.
 - 4.2.4.1. Domain Layer.
 - 4.2.4.2. Interface Layer.
 - 4.2.4.3. Application Layer.
 - 4.2.4.4. Infrastructure Layer.
 - 4.2.4.5. Bounded Context Software Architecture Component Level Diagrams.
 - 4.2.4.6. Bounded Context Software Architecture Code Level Diagrams.
 - 4.2.4.6.1. Bounded Context Domain Layer Class Diagrams.
 - 4.2.4.6.2. Bounded Context Database Design Diagram.

- 4.2.5. [Bounded Context: .](#)
- 4.2.5.1. [Domain Layer.](#)
- 4.2.5.2. [Interface Layer.](#)
- 4.2.5.3. [Application Layer.](#)
- 4.2.5.4. [Infrastructure Layer.](#)
- 4.2.5.5. [Bounded Context Software Architecture Component Level Diagrams.](#)
- 4.2.5.6. [Bounded Context Software Architecture Code Level Diagrams.](#)
- 4.2.5.6.1. [Bounded Context Domain Layer Class Diagrams.](#)
- 4.2.5.6.2. [Bounded Context Database Design Diagram.](#)
- 4.2.6. [Bounded Context: Treatment Support](#)
- 4.2.6.1. [Domain Layer.](#)
- 4.2.6.2. [Interface Layer.](#)
- 4.2.6.3. [Application Layer.](#)
- 4.2.6.4. [Infrastructure Layer.](#)
- 4.2.6.5. [Bounded Context Software Architecture Component Level Diagrams.](#)
- 4.2.6.6. [Bounded Context Software Architecture Code Level Diagrams.](#)
- 4.2.6.6.1. [Bounded Context Domain Layer Class Diagrams.](#)
- 4.2.6.6.2. [Bounded Context Database Design Diagram.](#)

5. [Capítulo V: Solution UI/UX Design.](#)

- 5.1. [Style Guidelines.](#)
- 5.1.1. [General Style Guidelines.](#)
- 5.1.2. [Web, Mobile and IoT 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. [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.](#)
- 5.4.2. [Applications Wireflow Diagrams.](#)
- 5.4.2.1. [Applications Mock-ups.](#)
- 5.4.3. [Applications User Flow Diagrams.](#)
- 5.5. [Applications Prototyping.](#)

6. [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. [Landing Page, Services & Applications Implementation.](#)
- 6.2.1. [Sprint 1.](#)
 - 6.2.1.1. [Sprint Planning 1.](#)
 - 6.2.1.2. [Aspect Leaders and Collaborators.](#)
 - 6.2.1.3. [Sprint Backlog 1.](#)
 - 6.2.1.4. [Development Evidence for Sprint Review.](#)
 - 6.2.1.5. [Testing Suite Evidence for Sprint Review.](#)
 - 6.2.1.6. [Execution Evidence for Sprint Review.](#)
 - 6.2.1.7. [Services Documentation Evidence for Sprint Review.](#)
 - 6.2.1.8. [Software Deployment Evidence for Sprint Review.](#)
 - 6.2.1.9. [Team Collaboration Insights during Sprint.](#)
- 6.2.2. [Sprint 2.](#)
 - 6.2.2.1. [Sprint Planning 2.](#)
 - 6.2.2.2. [Aspect Leaders and Collaborators.](#)
 - 6.2.2.3. [Sprint Backlog 2.](#)
 - 6.2.2.4. [Development Evidence for Sprint Review.](#)
 - 6.2.2.5. [Testing Suite Evidence for Sprint Review.](#)
 - 6.2.2.6. [Execution Evidence for Sprint Review.](#)
 - 6.2.2.7. [Services Documentation Evidence for Sprint Review.](#)
 - 6.2.2.8. [Software Deployment Evidence for Sprint Review.](#)
 - 6.2.2.9. [Team Collaboration Insights during Sprint.](#)
- 6.2.3. [Sprint 3.](#)
 - 6.2.3.1. [Sprint Planning 3.](#)
 - 6.2.3.2. [Aspect Leaders and Collaborators.](#)
 - 6.2.3.3. [Sprint Backlog 3.](#)
 - 6.2.3.4. [Development Evidence for Sprint Review.](#)
 - 6.2.3.5. [Testing Suite Evidence for Sprint Review.](#)

- 6.2.3.6. Execution Evidence for Sprint Review.
 - 6.2.3.7. Services Documentation Evidence for Sprint Review.
 - 6.2.3.8. Software Deployment Evidence for Sprint Review.
 - 6.2.3.9. 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.
7. Conclusiones
 8. Bibliografía
 9. Anexo

STUDENT OUTCOME

ABET – EAC - Student Outcome 5 La capacidad de funcionar efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos.

Criteriono específico	Actions taken	Conclusions
Trabaja en equipo para proporcionar liderazgo en forma conjunta	<p>Bohorquez Lerzundi Gerardo Sebastián:</p> <p>TB1: Participé activamente en reuniones del grupo, compartiendo ideas y propuestas sobre la estructuración del Bounded Context de Monitoring, organizando la funcionalidad en servicios y capas de arquitectura. Además, coordiné con mis compañeros para asegurar que las tareas se distribuyeran de manera equitativa y que cada miembro comprendiera su rol dentro del proyecto.</p> <p>TP1: En la segunda entrega, lideré la integración de nuevas funcionalidades en el sistema, facilitando la revisión de código y la resolución de conflictos entre ramas. Promoví la colaboración en la documentación técnica y la validación de los diagramas de</p>	<p>TB1: En conclusión, el equipo evidenció una dinámica de trabajo basada en la colaboración, la organización y el liderazgo compartido, donde cada integrante asumió un rol activo en la construcción de los bounded contexts, la aplicación del Lean UX Process y el diseño de la arquitectura del sistema. La distribución equitativa de tareas, la discusión de ideas y la validación conjunta de hipótesis fortalecieron la comunicación y la coordinación, lo que permitió tomar decisiones consensuadas y afrontar con eficacia los retos técnicos. Gracias a esta sinergia, el grupo logró estructurar de manera sólida los distintos dominios del proyecto y avanzar de forma ordenada hacia los objetivos comunes.</p> <p>TP1: En la segunda entrega, el liderazgo conjunto se consolidó mediante la</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>arquitectura, asegurando que todos los integrantes participaran en la toma de decisiones sobre el diseño final.</p> <p>TB2: Durante la última fase del proyecto, continué promoviendo el liderazgo compartido al coordinar la integración de los módulos finales y la revisión de los entregables. Fomenté la participación activa de todos los miembros en la toma de decisiones técnicas y en la validación de los resultados, asegurando que el liderazgo y la responsabilidad fueran distribuidos equitativamente.</p> <p>TF: En la entrega final, lideré el despliegue del servicio web y la aplicación web, coordinando con el equipo la integración de las funcionalidades finales. Promoví la colaboración activa en la revisión de código y la validación de los endpoints, asegurando que todos los integrantes participaran en la puesta en producción y en la documentación final del proyecto.</p> <p>Huanaco Huayta, Elizabeth Lucero</p> <p>TB1: El desarrollo de estas tareas, desde la aplicación del Lean UX Process hasta la definición del Bounded Context: Treatment Support con sus diferentes capas y diagramas, me permitió trabajar en equipo de manera organizada y colaborativa. Cada actividad requirió discutir ideas, validar supuestos, formular hipótesis y estructurar el diseño técnico de forma conjunta, lo que fomentó la toma de decisiones compartida y la distribución de responsabilidades. Este proceso fortaleció la comunicación y coordinación dentro del grupo, lo que me permitió contribuir activamente a alcanzar el objetivo común.</p> <p>TP1: En la segunda fase, participé en la revisión y mejora de los flujos de usuario y wireframes, coordinando sesiones de feedback con el equipo. Impulsé la integración de los prototipos en Figma y la validación conjunta de los criterios de aceptación, asegurando que el liderazgo fuera compartido y que todos los miembros pudieran aportar en la definición de la experiencia de usuario.</p> <p>TB2: A lo largo del proyecto trabajé de manera colaborativa con mi equipo, ejerciendo un liderazgo compartido que nos permitió organizarnos y tomar decisiones en conjunto. Participé activamente en la planificación, la asignación de tareas y la revisión de cada fase del trabajo, incluyendo el diseño de las entrevistas, la validación heurística y la creación del video About-the-Product. Esta forma de trabajo fomentó una comunicación constante, la resolución conjunta de problemas y la consideración de todas las opiniones, lo que fortaleció tanto el trabajo en equipo como nuestro liderazgo colectivo.</p> <p>TF: En la entrega final, participé activamente en la implementación y programación del hardware IoT, coordinando con el equipo la integración de los sensores y la comunicación con el backend. Lideré la validación de los datos capturados por el dispositivo y promoví la colaboración en las pruebas de funcionamiento, asegurando que el liderazgo fuera compartido en la fase de integración hardware-software.</p> <p>Huapaya Cuevas, Anthony Martin:</p> <p>TB1: En el equipo aporté liderazgo conjunto al desarrollar el Lean UX Process, ya que ayudó a estructurar nuestras hipótesis y experimentos de manera clara, lo que facilitó la toma de decisiones colaborativas. Además, trabajé en los bounded context canvases de health tracking, lo que permitió al grupo alinear la visión del sistema y delimitar responsabilidades dentro del diseño, favoreciendo un liderazgo distribuido. Finalmente, al realizar el registro de entrevistas, contribuí a organizar la información obtenida de los usuarios, asegurando que todo el equipo pudiera basar sus decisiones en evidencia real y compartida.</p> <p>TP1: En la segunda entrega, lideré la integración de los resultados de las entrevistas en el diseño de la arquitectura, coordinando la actualización de los diagramas y la documentación. Fomenté la participación activa de los miembros en la validación de los casos de uso y la revisión de los criterios de aceptación, consolidando el</p>	<p>integración de nuevas funcionalidades, la validación colaborativa de los prototipos y la revisión técnica de los diagramas. La participación activa de todos los miembros en la toma de decisiones y la resolución de conflictos permitió fortalecer la cohesión del equipo y avanzar de manera eficiente hacia los objetivos planteados, demostrando una capacidad sólida para liderar y coordinar el trabajo en equipo.</p> <p>TB2: Finalmente, en la última etapa del proyecto, el equipo mantuvo el liderazgo compartido y la colaboración activa en la integración y validación de los entregables finales. La participación equitativa de todos los miembros en la toma de decisiones y la revisión de los resultados permitió consolidar la cohesión del grupo y alcanzar los objetivos propuestos, demostrando una capacidad sólida para liderar y coordinar el trabajo en equipo hasta la culminación del proyecto.</p> <p>TF: En la entrega final, el equipo demostró un liderazgo compartido ejemplar al integrar exitosamente todos los componentes del sistema IoT: hardware, aplicación edge, servicios web y aplicación web. La coordinación efectiva entre los miembros permitió completar el despliegue del sistema completo, validar la comunicación en tiempo real y documentar adecuadamente el proyecto, consolidando las habilidades de liderazgo colaborativo desarrolladas a lo largo del ciclo.</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>liderazgo conjunto en la toma de decisiones técnicas.</p> <p>TB2: Durante el desarrollo del proyecto trabajé de manera colaborativa dentro de mi equipo, asumiendo un liderazgo compartido que nos permitió organizarnos y tomar decisiones de forma conjunta. Participé activamente en la planificación, distribución de tareas y revisión de cada etapa, como el diseño de entrevistas, la validación mediante heurísticas y la elaboración del video About-the-Product. Gracias a esta dinámica, contribuí a mantener una comunicación constante, resolver problemas en grupo y asegurar que todas las perspectivas fueran escuchadas, fortaleciendo así el trabajo en equipo y el liderazgo colectivo.</p> <p>TF: En la entrega final, lideré la implementación y programación del hardware IoT junto con mi compañera, coordinando la integración de los componentes electrónicos y la calibración de los sensores. Fomenté la participación activa del equipo en las pruebas de validación del dispositivo y en la documentación técnica del hardware, asegurando un liderazgo distribuido en esta fase crítica del proyecto.</p>	
	<p>Silva Tirado, Sebastián:</p> <p>TB1: Coordinó la distribución de responsabilidades entre los integrantes del equipo, fomentando la toma de decisiones de manera consensuada. Propuso soluciones ante dificultades técnicas y apoyó a sus compañeros en la comprensión de las tareas asignadas.</p> <p>TP1: En la segunda fase, lideré la planificación de los sprints y la asignación de tareas, asegurando que cada integrante tuviera claridad sobre sus responsabilidades. Promoví la revisión cruzada de entregables y la resolución colaborativa de problemas, fortaleciendo el liderazgo compartido y la cohesión del grupo.</p> <p>TB2: En la última etapa del proyecto, mantuve el liderazgo compartido al coordinar la integración de los entregables finales y la revisión de los resultados. Fomenté la participación de todos los miembros en la toma de decisiones y en la validación de los entregables, asegurando que el liderazgo y la responsabilidad fueran distribuidos de manera equitativa.</p> <p>TF: En la entrega final, lideré el desarrollo de la aplicación web y los servicios web, coordinando con el equipo la implementación de las funcionalidades de monitoreo en tiempo real. Promoví la colaboración en la integración del frontend con el backend y la validación de los flujos de usuario, asegurando que el liderazgo fuera compartido y que todos los integrantes participaran en la entrega final del producto.</p>	
	<p>Uribe Quispe Jesús Guillermo:</p> <p>TB1: Coordiné la elaboración de la arquitectura, proceso de eventstorming y needfinding. Ayudando a mis compañeros a realizar dicho proceso.</p> <p>TP1: En la segunda entrega, lideré la integración de los diagramas de arquitectura y la documentación técnica, facilitando la revisión y validación conjunta de los modelos. Promoví la participación activa en la definición de los flujos de trabajo y la toma de decisiones sobre la estructura final del sistema.</p> <p>TB2: En la última fase, continué promoviendo el liderazgo compartido al coordinar la integración de los módulos finales y la revisión de los entregables. Fomenté la participación activa de todos los miembros en la toma de decisiones técnicas y en la validación de los resultados, asegurando que el liderazgo y la responsabilidad fueran distribuidos equitativamente.</p> <p>TF: En la entrega final, lideré el desarrollo del servicio web, la arquitectura del sistema y la aplicación edge, coordinando la integración de todos los componentes del ecosistema IoT. Promoví la participación activa de todos los miembros en la definición de la arquitectura final y en la validación de la comunicación entre dispositivos, asegurando que el liderazgo y la responsabilidad fueran compartidos hasta la culminación del proyecto.</p>	

Entorno específico	Acciones realizadas	TB1: En conjunto, el equipo demostró un trabajo colaborativo e inclusivo, en el Conclusões que cada integrante aportó desde su rol al cumplimiento de los objetivos planteados. A través de la aplicación de metodologías como Lean UX y herramientas como empathy maps, user personas y bounded contexts, lograron definir necesidades reales de los usuarios, establecer metas claras, planificar tareas de manera organizada y dar seguimiento a su ejecución. La comunicación constante, la distribución equitativa de responsabilidades y la valoración de las ideas de todos los miembros fortalecieron el compromiso y permitieron alcanzar los entregables en los tiempos acordados, consolidando así un proceso de trabajo eficiente y orientado a resultados. TP1: En la segunda entrega, el equipo consolidó el entorno colaborativo e inclusivo mediante la planificación conjunta de los sprints, la definición de los objetivos y la revisión cruzada de los entregables. La participación activa de todos los miembros en la toma de decisiones y la retroalimentación constante permitieron cumplir los objetivos en los plazos establecidos, fortaleciendo la cohesión del grupo y la eficiencia en el trabajo en equipo. TB2: Finalmente, en la última etapa del proyecto, el equipo mantuvo un entorno colaborativo e inclusivo, asegurando la participación equitativa de todos los miembros en la planificación, organización y cumplimiento de los objetivos. La comunicación constante y la retroalimentación continua permitieron alcanzar los entregables en los tiempos acordados y consolidar un proceso de trabajo eficiente y orientado a resultados hasta la culminación del proyecto. TF: En la entrega final, el equipo consolidó un entorno colaborativo e inclusivo al integrar exitosamente todos los componentes del sistema AlertaVital: dispositivo IoT, aplicación edge, servicios web y aplicación web. La planificación clara de metas, la distribución equitativa de tareas y la comunicación constante permitieron cumplir todos los objetivos propuestos, demostrando la capacidad del equipo para trabajar de manera organizada y orientada a resultados en un proyecto IoT completo.
e inclusivo, establece metas, planifica tareas y cumple objetivos.	<p>Borhoquez Lerezundi Gerardo Sebastián:</p> <p>TB1: Contribuí en la fase de needfinding, realizando análisis de user persona y empathy map para identificar necesidades reales de los usuarios. Esto permitió al grupo empatizar con los clientes y definir soluciones más ajustadas. También participé en la planificación de tareas del proyecto, ayudando a cumplir objetivos establecidos en tiempo y forma.</p> <p>TP1: En la segunda entrega, colaboré en la definición de los objetivos del sprint y la planificación de las tareas, asegurando que el entorno de trabajo fuera inclusivo y que todos los miembros pudieran aportar sus ideas. Fomenté la revisión cruzada de entregables y la retroalimentación constante, lo que permitió cumplir los objetivos en los plazos establecidos.</p> <p>TB2: En la última etapa, participé activamente en la planificación y organización de las tareas finales, asegurando que el entorno de trabajo se mantuviera colaborativo e inclusivo. Contribuí a la definición de metas claras y al seguimiento del cumplimiento de los objetivos, promoviendo la participación de todos los miembros y la retroalimentación constante.</p> <p>TF: En la entrega final, contribuí al despliegue exitoso del servicio web y la aplicación web, estableciendo metas claras para la integración de las funcionalidades. Fomenté un entorno inclusivo donde cada miembro pudo aportar en la configuración del ambiente de producción y la validación de los servicios, cumpliendo los objetivos de despliegue en los plazos establecidos.</p>	
	<p>Huanaco Huayta, Elizabeth Lucero</p> <p>TB1: Durante el desarrollo de estas tareas logré desarrollar junto a mi equipo un entorno colaborativo e inclusivo, en el que cada integrante pudo aportar ideas y conocimientos desde su experiencia. Definimos metas claras, como la elaboración del Lean UX Canvas y los diferentes Bounded Context, lo que nos permitió planificar tareas de manera ordenada y asignarlas. Gracias a esta organización, pudimos cumplir con los objetivos establecidos en el tiempo acordado, fortaleciendo la participación equitativa y el compromiso hacia el proyecto del curso.</p> <p>TP1: En la segunda fase, participé en la planificación de los sprints y la definición de los entregables, promoviendo la inclusión de todos los miembros en la toma de decisiones. Coordiné la revisión de los prototipos y la validación de los criterios de aceptación, asegurando que el entorno de trabajo fuera colaborativo y que los objetivos se cumplieran en los plazos establecidos.</p> <p>TB2: Durante el desarrollo del proyecto me esforcé por fomentar un ambiente colaborativo e inclusivo dentro del equipo, garantizando que cada integrante tuviera espacio para participar y expresar sus ideas. Junto con mis compañeros establecimos metas claras, lo que permitió organizar y planificar las tareas de forma más eficaz. Además, procuré cumplir los objetivos planteados en cada etapa, manteniendo una comunicación constante y promoviendo un entorno de trabajo que favoreciera el crecimiento de todos y el progreso del proyecto.</p> <p>TF: En la entrega final, participé en la implementación del hardware IoT, estableciendo metas claras para la programación de los sensores y la integración con el sistema. Fomenté un ambiente inclusivo donde todos los miembros pudieron contribuir en las pruebas de calibración y validación del dispositivo, cumpliendo los objetivos técnicos planificados para la entrega final.</p>	
	<p>Huapaya Cuevas, Anthony Martin:</p> <p>TB1: Durante el proyecto fomenté un entorno colaborativo e inclusivo al aplicar el Lean UX Process, donde todas las ideas del equipo fueron consideradas y priorizadas de manera conjunta. A través de los bounded context canvases de health tracking, ayudé a establecer metas claras y a planificar tareas específicas relacionadas con el diseño de los dominios, asegurando que cada miembro comprendiera su responsabilidad. Con el registro de</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>entrevistas, contribuí a cumplir los objetivos del equipo, ya que la información recogida permitió orientar el diseño hacia las necesidades reales de los usuarios.</p> <p>TP1: En la segunda entrega, colaboré en la planificación de los sprints y la asignación de tareas, promoviendo la inclusión de todos los miembros en la toma de decisiones. Fomenté la revisión cruzada de entregables y la retroalimentación constante, lo que permitió cumplir los objetivos en los plazos establecidos y fortalecer el entorno colaborativo.</p> <p>TB2: A lo largo del proyecto procuré crear un entorno colaborativo e inclusivo dentro de mi equipo, asegurando que cada integrante pudiera participar y aportar sus ideas libremente. Establecí metas claras junto con mis compañeros, lo que facilitó organizar y planificar las tareas de manera eficiente. Asimismo, me aseguré de cumplir los objetivos propuestos en cada etapa, manteniendo una comunicación constante y promoviendo un ambiente de trabajo donde todos pudieran desarrollarse y contribuir al avance del proyecto.</p> <p>TF: En la entrega final, colaboré en la implementación del hardware IoT, definiendo metas específicas para el ensamblaje y programación del dispositivo. Promoví un entorno colaborativo durante las pruebas de integración hardware-software, asegurando que cada miembro del equipo pudiera participar en la validación y cumpliendo los objetivos de funcionamiento del sistema IoT.</p> <p>Silva Tirado, Sebastian:</p> <p>TB1: Promovió reuniones periódicas para definir objetivos y plazos, fomentó la participación activa de cada miembro y aseguró que las ideas fueran valoradas por igual. Elaboró un cronograma de actividades y dio seguimiento al cumplimiento de tareas.</p> <p>TP1: En la segunda fase, coordinó la planificación de los sprints y la definición de los entregables, asegurando que el entorno de trabajo fuera inclusivo y que todos los miembros pudieran aportar sus ideas. Fomentó la revisión cruzada de entregables y la retroalimentación constante, lo que permitió cumplir los objetivos en los plazos establecidos.</p> <p>TB2: En la última etapa, participé activamente en la planificación y organización de las tareas finales, asegurando que el entorno de trabajo se mantuviera colaborativo e inclusivo. Contribuí a la definición de metas claras y al seguimiento del cumplimiento de los objetivos, promoviendo la participación de todos los miembros y la retroalimentación constante.</p> <p>TF: En la entrega final, desarrollé la aplicación web y los servicios web, estableciendo metas claras para la implementación del monitoreo en tiempo real. Fomenté un entorno colaborativo e inclusivo durante la integración de los componentes, asegurando que todos los miembros participaran en la validación de las funcionalidades y cumpliendo los objetivos de la entrega final del producto.</p> <p>Uribe Quispe, Jesús Guillermo:</p> <p>TB1: Cree canales de comunicación entre mis compañeros y yo para desenvolvernos mejor en el proyecto. Además tracé y designé tareas con tiempo para una mayor eficiencia en la realización del entregable.</p> <p>TP1: En la segunda entrega, coordiné la planificación de los sprints y la definición de los entregables, promoviendo la inclusión de todos los miembros en la toma de decisiones. Fomenté la revisión cruzada de entregables y la retroalimentación constante, lo que permitió cumplir los objetivos en los plazos establecidos y fortalecer el entorno colaborativo.</p> <p>TB2: En la última etapa, continué promoviendo un entorno colaborativo e inclusivo, asegurando que cada integrante pudiera participar y aportar sus ideas libremente. Junto con mis compañeros establecimos metas claras y planificamos las tareas</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>de manera eficiente, lo que permitió cumplir los objetivos propuestos y mantener una comunicación constante hasta la culminación del proyecto.</p> <p>TF: En la entrega final, desarrollé el servicio web, la arquitectura del sistema y la aplicación edge, estableciendo metas claras para la integración del ecosistema IoT. Promoví un entorno colaborativo donde cada miembro pudo contribuir en la definición de los flujos de datos y la validación de la comunicación entre componentes, cumpliendo los objetivos de integración total del sistema.</p>	

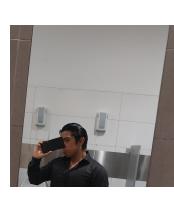
1. Capítulo I: Introducción

1.1. Startup Profile

1.1.1. Descripción del startup

AlertaVital es una startup tecnológica dedicada a desarrollar soluciones innovadoras basadas en Internet de las Cosas (IoT) para mejorar la calidad de vida de las personas mayores y sus familias. Nuestra misión es proporcionar herramientas accesibles y confiables que permitan el monitoreo en tiempo real de signos vitales y la detección de caídas, brindando tranquilidad y seguridad tanto a los adultos mayores como a sus cuidadores. A través de un dispositivo portátil y una aplicación móvil intuitiva, AlertaVital busca transformar la manera en que se cuida a los adultos mayores, facilitando la prevención de emergencias y promoviendo un envejecimiento saludable y autónomo.

1.1.2. Perfiles de los integrantes del equipo

	<p>Nombre y Apellido: Bohorquez Lenzundi, Gerardo Sebastián</p> <p>Carrera: Ingeniería de Software</p> <p>Acerca de: Estoy cursando el séptimo semestre de Ingeniería de Software en la Universidad UPC. Me distingo por ser una persona ágil, comprometida, assertiva y profundamente ética. En este proyecto, me comprometo a aportar lo mejor de mí, no solo para alcanzar los objetivos del equipo, sino también para fomentar un ambiente de trabajo positivo y colaborativo.</p>
	<p>Nombre y Apellido: Huancayo Huayta, Elizabeth Lucero</p> <p>Carrera: Ingeniería de Software</p> <p>Acerca de: Estoy en el séptimo ciclo de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas. Me considero una persona organizada, resiliente y assertiva. Elegí enfocarme en esta carrera por el interés que tengo en el desarrollo de software, ya que considero que es un área clave para el futuro de la innovación tecnológica. En este proyecto, me comprometo a aplicar mis cualidades y contribuir al logro de los objetivos de mi equipo.</p>
	<p>Nombre y Apellido: Huapaya Cuevas, Anthony Martin</p> <p>Carrera: Ingeniería de Software</p> <p>Acerca de: Me encuentro cursando el séptimo ciclo de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas. Me considero una persona perseverante, fácil de adaptarse con respecto al ambiente de trabajo y responsable. Elegí el desarrollo de esta carrera para los avances tecnológicos con respecto al software, por que es una parte fundamental para lograr ese objetivo. Me comprometo con este trabajo mantener un ambiente sano en el equipo y apoyar constantemente con el avance de este.</p>
	<p>Nombre y Apellido: Silva Tirado, Sebastián Valentino</p> <p>Carrera: Ingeniería de Software</p> <p>Acerca de: Soy estudiante del octavo ciclo de la carrera de Ingeniería de Software en UPC. Opté por estudiar esta carrera porque me gusta la idea de programar y crear algo que pueda llegar a funcionar y ser útil en el mundo. Por lo cual, la universidad me da la posibilidad de estudiar y ser un profesional exitoso. Deseo ser buen integrante para mi equipo y me comprometo en hacer el mejor trabajo posible, y poder superarme más que el ciclo anterior.</p>
	<p>Nombre y Apellido: Uribe Quispe, Jesús Guillermo</p> <p>Carrera: Ingeniería de Software</p> <p>Acerca de: Soy estudiante de Ingeniería de Software en la UPC y actualmente estoy en el 7to ciclo. Elegí esta carrera porque me interesó el campo del software y la matemáticas. Me comprometo a ayudar a mis compañeros con el trabajo así como también fuera de este.</p>

1.2. Solution Profile

1.2.1. Antecedentes y Problemática

Antecedentes:

El envejecimiento poblacional es un fenómeno global y particularmente relevante en Latinoamérica. En Perú, según el INEI (2022), la población de adultos mayores supera el 13% del total, con proyecciones de crecimiento constante en las próximas décadas. Este grupo etario presenta un incremento en la prevalencia de enfermedades crónicas no transmisibles (hipertensión, diabetes, enfermedades cardiovasculares), las cuales requieren un monitoreo continuo de signos vitales.

Paralelamente, la Organización Mundial de la Salud (OMS, 2021) advierte que las caídas son la segunda causa mundial de muertes por lesiones accidentales, afectando principalmente a adultos mayores. Los dispositivos IoT aplicados a la telemedicina se han convertido en una alternativa de apoyo para la prevención y respuesta temprana, ya que permiten el registro en tiempo real de parámetros fisiológicos y eventos de riesgo, brindando a los cuidadores o familiares alertas inmediatas.

Sin embargo, en el contexto peruano y de Lima Metropolitana, el acceso a estas tecnologías aún es limitado, con soluciones costosas o poco adaptadas a la realidad socioeconómica local. Esto abre la necesidad de investigar y desarrollar herramientas accesibles que permitan mejorar la calidad de vida de las personas mayores y la tranquilidad de sus familias.

Problemática:

La población adulta mayor enfrenta riesgos significativos relacionados con el deterioro de la salud y la vulnerabilidad a caídas. Actualmente, el monitoreo de sus signos vitales suele ser esporádico y depende de consultas médicas presenciales, lo que retrasa la detección de anomalías. Asimismo, muchas caídas ocurren en el hogar sin supervisión inmediata, aumentando la probabilidad de complicaciones graves.

Si bien existen dispositivos en el mercado, su alto costo, falta de personalización y limitada integración con sistemas de alerta dificultan su adopción en el entorno local. Esto genera un vacío tecnológico que afecta tanto a los adultos mayores como a sus familiares y cuidadores, quienes requieren herramientas eficaces para prevenir riesgos y actuar con rapidez ante emergencias.

WHAT (Qué): ¿Cuál es el problema?

El problema es la falta de monitoreo continuo y accesible de signos vitales y detección de caídas en adultos mayores, lo que ocasiona diagnósticos tardíos, complicaciones y aumento del riesgo de mortalidad.

WHEN (Cuándo): ¿Cuándo sucede el problema?

El problema ocurre de manera cotidiana, especialmente durante los momentos en que la persona mayor se encuentra sola en casa o sin supervisión médica inmediata.

WHERE (Dónde): ¿Dónde surge el problema?

Surge principalmente en el hogar y en entornos donde no existe personal de salud presente de forma constante, siendo un problema frecuente en zonas urbanas con alta población adulta mayor, como Lima Metropolitana.

WHO (Quién): ¿A quiénes les sucede el problema?

Afecta directamente a los adultos mayores, y de manera indirecta a sus familiares, cuidadores y al sistema de salud, que enfrenta mayores costos por hospitalizaciones derivadas de caídas o descompensaciones no detectadas a tiempo.

WHY (Por qué): ¿Cuál es la causa del problema?

Las causas principales son:

- Falta de acceso a herramientas tecnológicas de monitoreo.
- Alto costo de los dispositivos especializados.
- Limitada cultura de prevención en salud geriátrica.
- Ausencia de soluciones adaptadas al contexto local.

HOW (Cómo): ¿Cómo llevó a los involucrados a llegar a esta situación?

La combinación de factores económicos, tecnológicos y sociales ha llevado a que los adultos mayores permanezcan sin supervisión adecuada de sus parámetros vitales, exponiéndose a eventos de riesgo sin respuesta inmediata, lo que incrementa la carga emocional y económica de las familias.

HOW MUCH (Cuánto): ¿Cuánto afecta el problema?

El problema afecta a más del 13% de la población peruana (adultos mayores), y las caídas representan aproximadamente el 30% de las consultas de emergencia en este grupo etario (OPS, 2020). Además, los costos hospitalarios derivados de caídas y complicaciones médicas son elevados, impactando tanto a los hogares como al sistema de salud nacional.

1.2.2. Lean UX Process

1.2.2.1. Lean UX Problem Statements

En la actualidad, el cuidado de los adultos mayores representa un gran reto debido a la falta de mecanismos accesibles y confiables para monitorear su estado de salud y seguridad en tiempo real. Este desafío se acentúa en contextos donde muchos adultos mayores viven solos o pasan gran parte del día sin supervisión constante, lo que incrementa el riesgo ante emergencias como caídas o alteraciones en sus signos vitales. Garantizar una atención oportuna en este entorno es fundamental para prevenir complicaciones, mejorar la calidad de vida y ofrecer tranquilidad tanto a los adultos mayores como a sus familias.

Actualmente, las familias y cuidadores aún dependen de visitas presenciales o llamadas telefónicas para conocer el estado de sus seres queridos, lo que genera incertidumbre, tiempos de respuesta lentos y, en ocasiones, falta de información médica precisa para la toma de decisiones. La ausencia de una solución tecnológica integral limita la capacidad de detectar emergencias de manera inmediata y de contar con un registro histórico de la salud del adulto mayor, dificultando así la prevención y el seguimiento adecuado de su bienestar.

¿Cómo podemos desarrollar una solución integral basada en dispositivos IoT y una aplicación móvil que permita a los cuidadores o familiares monitorizar en tiempo real los signos vitales y posibles caídas de los adultos mayores, generando alertas inmediatas y un historial accesible, de manera que se incremente la seguridad, se optimice el cuidado y se brinde mayor tranquilidad a todos los involucrados?

1.2.2.2. Lean UX Assumptions

Business Assumptions

- Creemos que nuestros clientes necesitan una mejor forma de monitorear el estado de salud y seguridad de los adultos mayores, especialmente en lo relacionado con el registro de signos vitales, la detección de caídas y la generación de alertas inmediatas para familiares o cuidadores.
- Estas necesidades se pueden resolver mediante el uso de un dispositivo IoT portátil conectado a una aplicación móvil, que permita la recolección de datos en tiempo real, la centralización de información de salud y el envío de notificaciones en caso de emergencia.
- Los clientes principales son los tutores de los adultos mayores, quienes buscan tranquilidad y rapidez en la respuesta ante emergencias. Sin embargo, la solución también está diseñada para beneficiar directamente a los adultos mayores, al ofrecerles mayor seguridad en su vida cotidiana.
- El valor #1 que busca el cliente es contar con un sistema accesible y confiable que incremente la seguridad y bienestar de los adultos mayores, a la vez que brinde tranquilidad a los tutores mediante la detección temprana de emergencias y un monitoreo continuo de la salud.
- El cliente también puede obtener beneficios adicionales gracias a la flexibilidad y escalabilidad de la solución, que permitirá integrar nuevas métricas de salud como oxigenación, presión arterial y frecuencia cardíaca, además de incluir la capacidad de realizar llamadas a los servicios de emergencia y a los contactos registrados. Asimismo, se podrán generar reportes sobre posibles alteraciones detectadas en el estado de salud.
- Vamos a adquirir la mayoría de los clientes mediante el uso de marketing digital dirigido a familias y asociaciones de adultos mayores, además de alianzas con centros de salud, hogares geriátricos y ONGs, lo que permitirá ampliar la red de usuarios y aumentar la visibilidad de la solución.
- Se hará dinero a través de la venta del dispositivo IoT acompañado de un pago único por el acceso de la aplicación, aplicando cuotas bajas por el soporte y mantenimiento de la aplicación y el dispositivo.
- Nuestra competencia principal en el mercado son las aplicaciones y dispositivos de monitoreo de salud y caídas ya existentes, aunque muchos presentan precios elevados, falta de integración de funciones o interfaces poco intuitivas para adultos mayores.
- Los venceremos gracias al uso de tecnología accesible, un enfoque integral en seguridad y salud, y una experiencia de usuario optimizada tanto para los adultos mayores como para sus cuidadores. Además, ofreceremos soporte técnico cercano y constantes mejoras de la aplicación.
- Nuestro mayor riesgo es que los adultos mayores y sus familias desconfíen o sean reacios al uso de nuevas tecnologías.
- Resolveremos esto a través de demostraciones claras, interfaces simples y un soporte al cliente enfocado en la capacitación, de modo que la adaptación al sistema sea rápida, amigable y sin complicaciones.

Assumptions Worksheet

¿Quién es el usuario? Los usuarios son los adultos mayores y sus tutores, quienes necesitan una solución confiable para monitorear la salud y la seguridad en tiempo real. El usuario principal son los tutores, quienes toman decisiones sobre herramientas tecnológicas y buscan soluciones que ofrezcan tranquilidad, respuesta rápida ante emergencias y un mejor cuidado para sus seres queridos.

¿Dónde encaja nuestro producto en su trabajo o vida? Nuestra aplicación móvil junto con el dispositivo IoT se integra en la vida diaria del adulto mayor y su entorno familiar. El dispositivo acompaña al adulto mayor durante todo el día, mientras que la aplicación permite a los tutores monitorear signos vitales, recibir alertas ante caídas y acceder a un historial de salud, convirtiéndose en una herramienta esencial para incrementar la seguridad, optimizar el cuidado y brindar confianza.

¿Qué problemas tiene nuestro producto? ¿Resolver? El desafío que enfrentan los usuarios es la falta de monitoreo constante cuando los adultos mayores están solos, el riesgo de caídas sin asistencia inmediata y la ausencia de información médica histórica que dificulta el seguimiento de la salud. También puede existir resistencia por parte de algunos adultos mayores al uso de nuevas tecnologías o la desconfianza inicial de las familias hacia la precisión del sistema.

¿Cuándo y cómo es nuestro producto usado? La aplicación y el dispositivo IoT tienen un uso diario. El adulto mayor porta el dispositivo durante sus actividades cotidianas, mientras que los tutores utilizan la aplicación para monitorear en tiempo real, recibir notificaciones ante emergencias y consultar el historial de eventos o signos vitales.

¿Qué características son importantes? Las principales características son el monitoreo en tiempo real de signos vitales y caídas, el envío de alertas automáticas inmediatas, un historial accesible con reportes de salud y una interfaz simple e intuitiva.

¿Cómo debe verse nuestro producto y cómo comportarse? Debe verse limpio, amigable y confiable, transmitiendo seguridad y accesibilidad para el ámbito de salud y cuidado. La aplicación debe responder rápidamente a las acciones del usuario con una navegación sencilla presentar paneles claros con información organizada y notificaciones inmediatas.

1.2.2.3. Lean UX Hypothesis Statements

Creemos que, al incorporar un módulo de monitoreo en tiempo real de signos vitales y caídas, se incrementará la seguridad y tranquilidad de los adultos mayores y sus familias. Sabremos que tenemos éxito cuando logremos una reducción del 70% en incidentes no atendidos a tiempo y una respuesta más rápida por parte de los tutores.

Creemos que, al ofrecer reportes resumidos sobre los signos vitales y eventos de caídas, los tutores podrán tomar decisiones más informadas para el cuidado del adulto mayor. Sabremos que estamos cumpliendo este objetivo cuando los cuidadores consulten regularmente los reportes y se observe una mejora del 50% en la prevención y seguimiento de la salud.

Creemos que, al incorporar un sistema de alertas automáticas y notificaciones inmediatas a los servicios de emergencia y números registrados, se mejorará la capacidad de respuesta ante emergencias. Sabremos que estamos cumpliendo este objetivo cuando el 80% de las alertas generadas reciban una atención oportuna.

1.2.2.4. Lean UX Canvas

El Lean UX Canvas permitió estructurar y validar las hipótesis del proyecto, que busca ofrecer un dispositivo IoT con aplicación móvil para el monitoreo de adultos mayores. Con esta herramienta se identificaron los principales problemas, como la falta de monitoreo constante y el riesgo de caídas, los usuarios clave, como los

adultos mayores y los tutores, así como los supuestos críticos y riesgos, como la resistencia a nuevas tecnologías. Además, el canvas ayudó a priorizar las funcionalidades esenciales y una interfaz simple, garantizando que la solución se enfoque en generar seguridad, confianza y valor real para los usuarios.



1.3. Segmentos objetivo

En esta sección, definiremos los perfiles específicos del público objetivo para nuestro brazalete inteligente. Estos grupos comparten características demográficas, comportamientos o necesidades similares, lo que les diferencia de otros segmentos del mercado y permite ajustar estrategias para satisfacer sus necesidades de una mejor manera.

1. Personas de edad avanzada o individuos con enfermedades cardiovasculares

• Segmento demográfico:

Personas mayores de 60 años en Perú, que representan cerca del 12,4 % de la población total (aproximadamente 4,1 millones en 2023), con una prevalencia de hipertensión arterial del 29,7 % y un 66 % con enfermedades crónicas como diabetes o patologías cardiovasculares (INEI, 2023; MINSA, 2023).

• Segmento geográfico:

Residentes en áreas urbanas, especialmente en Lima Metropolitana, donde el 43,2 % de los hogares incluye al menos un adulto mayor (INEI, 2023).

• Segmento psicográfico:

Individuos preocupados por su salud y seguridad, que valoran la tranquilidad de contar con monitoreo continuo para prevenir o manejar emergencias médicas como desmayos, arritmias o crisis hipertensivas.

• Segmento conductual:

Personas que necesitan monitoreo constante de signos vitales (pulso, respiración, saturación) y alertas automáticas ante condiciones críticas. El brazalete inteligente satisface esta necesidad al proporcionar monitoreo en tiempo real y notificaciones inmediatas.

2. Cuidadores o contactos de emergencia de los usuarios

• Segmento demográfico:

Cuidadores familiares, mayormente sin capacitación formal, que conviven en más de 4,4 millones de hogares con un adulto mayor o paciente crónico. Aproximadamente el 31 % presenta síntomas depresivos, y el 44 % ha dejado su empleo para dedicarse al cuidado (OPS, 2023).

• Segmento geográfico:

Principalmente en áreas urbanas de Perú, con alta concentración en Lima Metropolitana, donde la convivencia con adultos mayores o pacientes crónicos es común (OPS, 2023).

• Segmento psicográfico:

Cuidadores que enfrentan alto estrés, sobrecarga emocional y mental (más del 70 % reporta sobrecarga intensa), y buscan herramientas que les brinden seguridad y reduzcan la incertidumbre al cuidar de sus seres queridos.

• Segmento conductual:

Cuidadores que necesitan recibir alertas inmediatas ante emergencias de salud para intervenir rápidamente. El brazalete inteligente les ofrece monitoreo continuo del paciente y notificaciones automáticas, aliviando su carga emocional y mejorando su capacidad de respuesta.

2. Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores

2.1.1. Análisis competitivo

Competidor					
Perfil Overview	Nuestra Solucion Brazalete inteligente IoT con app móvil para monitoreo en tiempo real de signos vitales y detección de caídas en adultos mayores, enfocado en el mercado peruano.	Reloj inteligente con enfoque en monitoreo de salud, fitness y seguridad, incluyendo detección de caídas y signos vitales para usuarios mayores.	Reloj de alerta médica con detección de caídas, diseñado para protección 24/7 sin necesidad de recarga constante.	Reloj de alerta médica independiente con detección de caídas automática y seguimiento de ubicación para seniors activos.	Reloj inteligente para seniors con detección de caídas, monitoreo de signos vitales y GPS en tiempo real.
Ventaja competitiva	Solución accesible y local con integración IoT completa, alertas inmediatas a cuidadores y servicios de emergencia, interfaz intuitiva para adultos mayores y soporte técnico cercano en Perú.	Sensores avanzados para ECG, detección de hipertensión y apnea del sueño, integración con ecosistema Apple.	Tecnología RealFall™ para detección precisa de caídas, batería intercambiable y conectividad dual (celular y WiFi).	Operación independiente sin smartphone, detección de caídas con IA y red 4G LTE para claridad y velocidad.	Dispositivo gratuito con suscripción, compliant con HIPAA, centro de emergencias certificado 24/7.
Mercado objetivo	Adultos mayores de 60 años en Perú (especialmente Lima Metropolitana) con enfermedades crónicas, y sus cuidadores familiares que buscan tranquilidad y respuesta rápida.	Adultos mayores y personas preocupadas por la salud en áreas urbanas, con énfasis en independencia y monitoreo continuo.	Seniors que necesitan protección constante contra caídas y emergencias, priorizando facilidad de uso.	Elderly activos que requieren alertas médicas sin dependencia de teléfonos, en entornos cotidianos.	Adultos mayores viviendo solos o con condiciones médicas, y sus cuidadores buscando tranquilidad.
Estrategia de marketing	Marketing digital dirigido a familias y asociaciones de adultos mayores, alianzas con centros de salud, hogares geriátricos y ONGs en Perú.	Presencia en tiendas Apple, publicidad digital, énfasis en características de salud y fitness, ofertas como Fitness+ gratis.	Endorsements de expertos (NY Times, NCOA), historias de clientes, contenido educativo comparativo.	Promociones como descuentos y meses gratis, enfoque en simplicidad y reseñas positivas (4.8/5).	Testimonios de usuarios y médicos, calificación alta (4.9/5), énfasis en transparencia y recomendaciones.

Competidor




AlertaVital

Productos & Servicios	Monitoreo continuo de signos vitales (pulso, respiración, saturación de oxígeno), detección de caídas, alertas automáticas a cuidadores y emergencias, historial accesible, reportes de salud, integración con llamadas de emergencia.	Detección de caídas, monitoreo de signos vitales (ritmo cardíaco, respiración, temperatura), ECG, notificaciones de irregularidades, alertas de emergencia.	Detección de caídas automática, activación por voz o botón, conexión a agentes 24/7, sin monitoreo vital directo.	Detección de caídas con IA, botón SOS, comunicación bidireccional, contador de pasos, alertas de batería baja.	Detección de caídas automática, monitoreo de ritmo cardíaco, GPS, botón SOS con altavoz bidireccional, alertas a emergencias.
Precios & Costos	Venta del dispositivo con pago único por acceso a la app, cuotas bajas mensuales por soporte y mantenimiento.	Precio del dispositivo desde aproximadamente \$399, sin suscripción obligatoria para funciones básicas de salud.	Precio no especificado en el contenido, enfocado en suscripción para monitoreo 24/7.	Suscripciones desde \$39.95/mes, con promociones de descuento y mes gratis.	Suscripciones desde \$34.95/mes, dispositivo gratis, sin tarifas ocultas.
Canales de distribución	App móvil (iOS/Android), ventas en línea, alianzas con centros de salud y ONGs.	Tiendas Apple, sitio web, App Store.	Sitio web oficial, posiblemente Amazon.	Sitio web, compras en línea con envíos.	Sitio web oficial con envíos gratuitos.
Debilidades	Producto nuevo en el mercado, posible resistencia inicial de adultos mayores a la tecnología, dependencia de cobertura móvil en Perú.	Requiere iPhone para configuración completa, batería diaria, precio alto para algunos usuarios.	No apto para inmersión en agua, no monitorea signos vitales directamente.	Vida de batería variable (6-18 horas), GPS limitado actualmente.	Dependencia de red móvil, posibles limitaciones en áreas sin cobertura.
Oportunidades	Crecimiento del envejecimiento poblacional en Perú (13%+), alta demanda de soluciones accesibles para monitoreo de salud crónica y prevención de caídas.	Expansión en telemedicina y integración con más métricas de salud en mercados crecientes de envejecimiento.	Mejoras en tecnología RealFall™ y expansión a más usuarios seniors con datos recolectados.	Desarrollo de app para cuidadores y mejoras en batería para mercado en crecimiento de wearables para elderly.	Expansión basada en recomendaciones médicas y mercado de independencia senior.
Fortalezas	Adaptada al contexto socioeconómico peruano, enfoque en accesibilidad y escalabilidad, interfaz simple y confiable, beneficios directos para cuidadores con reducción de estrés.	Monitoreo avanzado de salud, detección de caídas y emergencias, integración con conectividad familiar.	Protección 24/7, detección precisa de caídas, diseño amigable para usuarios.	Independiente, detección de caídas precisa, materiales cómodos, reseñas positivas.	Confiable, fácil configuración, alta satisfacción de usuarios, enfoque en privacidad.

2.1.2. Estrategias y tácticas frente a competidores

- Diseñaremos un sistema de monitoreo accesible y fácil de usar, con una interfaz pensada tanto para adultos mayores como para familiares y cuidadores. La navegación será intuitiva, con alertas visuales y auditivas claras, además de un soporte técnico adaptado al usuario final, garantizando la confianza y la rápida adopción de la herramienta.
- Diferenciaremos nuestra propuesta con la integración del sensor de caídas, que no solo enviará alertas automáticas, sino que también permitirá confirmar el estado del paciente a través de la aplicación móvil. Esta característica clave nos permitirá superar a competidores que solo ofrecen monitoreo de signos vitales sin una respuesta integral frente a emergencias.
- Fortaleceremos la credibilidad de nuestra solución a través de experiencias reales, incorporando testimonios de familiares, cuidadores y profesionales de la salud que validen la utilidad del sistema. Estas historias de éxito serán empleadas en nuestras campañas de marketing y materiales informativos para generar confianza en nuevos usuarios.

2.2. Entrevistas

Esta sección recopila la información obtenida a partir de entrevistas realizadas a adultos mayores y cuidadores responsables de su atención. Se incluyen las preguntas formuladas, las entrevistas realizadas y un análisis comparativo. El objetivo es comprender sus necesidades, hábitos actuales y percepciones para fundamentar el desarrollo de una solución IoT adecuada a su entorno real.

2.2.1. Diseño de entrevistas

Preguntas para Adultos Mayores

1. ¿Cuál es su nombre y edad?
2. ¿Con quién vive actualmente (solo, con familia, en residencia)?
3. ¿Qué enfermedades o condiciones de salud crónicas le han diagnosticado (ej. hipertensión, diabetes, problemas cardíacos)?
4. ¿Con qué frecuencia acude a controles médicos o revisiones de rutina?
5. ¿Ha sufrido caídas en el último año? ¿Podría contarme qué pasó y cómo reaccionaron sus familiares?
6. ¿En qué momentos del día suele estar solo(a) en casa?
7. ¿Qué tanta confianza tiene en el uso de dispositivos electrónicos (celulares, relojes inteligentes, pulseras, etc.)?
8. ¿Ha usado antes algún aparato o dispositivo para medir su presión, ritmo cardíaco u otros signos vitales? ¿Cómo fue su experiencia?
9. ¿Qué tan cómodo se sentiría utilizando un dispositivo pequeño (ej. pulsera, reloj o colgante) que monitoree su salud y detecte caídas?
10. ¿Le resultaría útil que sus familiares reciban alertas inmediatas si sufre una caída o alguna alteración de sus signos vitales?
11. ¿Qué aspectos le generarían desconfianza o preocupación (precio, privacidad, dificultad de uso, batería, etc.)?
12. ¿Qué características considera más importantes en un dispositivo de este tipo (ej. fácil de usar, discreto, resistente, económico)?

Preguntas para Cuidadores / Familiares

1. ¿Cuál es su relación con el adulto mayor (hijo/a, nieto/a, cuidador contratado, etc.)?
2. ¿Con qué frecuencia se encuentra a cargo de su cuidado (diariamente, algunas horas, solo en emergencias)?
3. ¿Qué actividades cotidianas le generan mayor preocupación respecto al adulto mayor (movilidad, alimentación, medicación, caídas, etc.)?
4. ¿Ha tenido alguna experiencia previa con caídas o emergencias médicas de su familiar? ¿Cómo fue el proceso de atención?
5. ¿Actualmente utilizan algún dispositivo o aplicación para monitorear su salud? ¿Cuál y cómo lo valoran?
6. ¿Qué tan importante considera tener información en tiempo real sobre signos vitales (ej. presión, pulso, oxigenación)?
7. ¿Cómo preferiría recibir alertas en caso de emergencias (notificación en app, llamada, SMS, correo)?
8. ¿Cuáles serían las principales barreras para implementar un dispositivo de este tipo en su hogar (costo, conectividad, resistencia del adulto mayor a usarlo)?
9. ¿Qué tan dispuesto estaría a invertir en un sistema IoT de monitoreo? ¿Qué rango de precio considera accesible?
10. Si pudiera diseñar este sistema, ¿qué funciones serían prioritarias para usted (ej. botón de emergencia, historial de datos, geolocalización, recordatorio de medicación)?
11. ¿Qué expectativas tendría en términos de seguridad y confiabilidad del dispositivo?
12. ¿Cómo cree que un sistema de este tipo impactaría en su tranquilidad y en la calidad de vida del adulto mayor?

2.2.2. Registro de entrevistas

Segmento 1: Adultos Mayores

Entrevistado Nº1	Rosa Valera
Edad	65
Distrito de residencia	San Luis
 Rosa Valera es una mujer de 65 años que vive en San Luis. Ella ha tenido resbalones en el último año y se siente insegura al estar sola en casa. Rosa tiene confianza en el uso de dispositivos electrónicos, pero no ha utilizado dispositivos para medir su salud. Le gustaría un dispositivo que monitoree su salud y alerte a sus familiares en caso de emergencia.	

Timing:

04:36 - URL: [Link del Video](#)
08:32

Entrevistado Nº2	Deodora Arias
Edad	81
Distrito de residencia	Santa Cruz de Flores
 Deodora vive con su esposo y uno de sus hijos, a ella le detectaron diabetes, hipertensión y artrosis. Mensualmente acude al seguro para que le hagan su chequeo. Si ha llegado a sufrir caídas, su hijo que vive con ella se comunica con sus otros hijos y estos se ponen preocupados por dicho accidente. Generalmente se queda en casa con su hijo y esposo. Ella usa solo el celular para la comunicación y entretenimientos. Ella ha usado distintos dispositivos para medir la presión y como lleva controlada su diabetes. Ella ve factible el uso de un dispositivo como si fuera un reloj o algo un poco más discreto, además de que le interesa que este mande una alerta inmediata a sus familiares si ha sufrido alguna caída o alteración de sus signos vitales. Lo que considera más importante es que sea fácil de usar, ya que no maneja muy bien los dispositivos electrónicos.	

Timing:

00:05 - URL: [Link del Video](#)
04:36

Entrevistado Nº3	Juan Carlos Herrera
Edad	78
 Juan Carlos Herrera es un hombre de 78 años que vive en Ate. Tiene diabetes y utiliza un dispositivo para monitorear su salud. Considera que es útil para detectar cambios rápidos en su condición y alertar a sus hijos.	

residencia



Juan Carlos Herrera es un adulto mayor de 78 años que vive en Ate. Él ha tenido problemas de salud en el pasado, incluyendo diabetes tipo 2. Juan Carlos se siente inseguro al estar solo en casa y le gustaría tener un dispositivo que lo ayude a monitorear su salud y que alerte a sus familiares en caso de emergencia.

Timing: 6:58
min

URL: [Link del Video](#)

Segmento 2: Cuidadores / Familiares

Entrevistado
Nº1 Marcia Rea Gamboa

Edad 21

Distrito de
residencia Comas



Marcia Rea es una joven que cuida de su abuela, que no se puede cuidar por sí misma. Ella busca constantemente herramientas que le permitan monitorear la salud de su abuela a distancia.

Timing:
25:27 - 33:14 URL: [Link del Video](#)

Entrevistado
Nº2 Fabrisio Belahonia Miranda

Edad 25

Distrito de
residencia Ate

Fabrisio es hijo y cuidador principal de su padre adulto mayor, con quien convive en la misma casa. Debido a que trabaja fuera de casa, su mayor preocupación es que el adulto mayor pase varias horas solo. Relató una experiencia previa en la que su padre sufrió una caída mientras él estaba trabajando, lo que evidenció la vulnerabilidad de estas situaciones y la necesidad de contar con apoyo tecnológico para reaccionar de manera oportuna. Actualmente, solo utilizan un tensiómetro digital en los horarios en los que el tutor está presente, aunque el adulto mayor lo encuentra incómodo y suele retirárselo, lo que reduce su efectividad. Él considera fundamental disponer de información en tiempo real sobre los signos vitales, especialmente porque pasa gran parte del día fuera de casa. Además, señaló que la forma más efectiva de recibir alertas sería mediante una llamada automática, ya que no siempre puede revisar notificaciones durante el trabajo. Sin embargo, manifestó disposición a invertir en un sistema IoT de monitoreo si es confiable y este se encuentra en un rango de precio accesible.

Timing:
08:37 - 15:33 URL: [Link del Video](#)

Entrevistado
Nº3 Sonia Huapaya Arias

Edad 62

Distrito de
residencia Villa María del Triunfo

Sonia es una enfermera técnica que brinda atención a cualquier tipo de pacientes, incluyendo adultos mayores. Generalmente, le da atención a los adultos mayores cada vez que se hospitalizan, brindando cuidado por la calidad de vida. Las mayores preocupaciones que suele tener son la movilidad, ya que sus cuerpos no le permiten por su edad. Cómo trata con adultos mayores con bajas defensas, siempre tiene que estar al pendiente si les ocurre alguna caída. Ella usa como una pizarra que le indica mediante colores el control del adulto mayor teniendo en cuenta el riesgo de caída. Estos dispositivos les indican cada 6 horas los signos vitales. Le parece conveniente que se envíe un mensaje o correo una alerta para casos de emergencias. Sus principales barreras es la conectividad con el internet, ya que no tienen en la zona que trabaja. No tiene una idea exacta de cuánto sería el costo, pero si muestra interés por dichos aparatos.

Timing:
15:33 - 25:27 URL: [Link del Video](#)

2.2.3. Análisis de entrevistas

Análisis del segmento de adultos mayores

Los adultos mayores entrevistados valoran la posibilidad de contar con un dispositivo IoT que les brinde seguridad dentro de su propio hogar, especialmente en relación con la detección de caídas y el monitoreo de signos vitales. El 100% coincidió en que la principal ventaja sería la tranquilidad de sentirse acompañados incluso cuando están solos, reduciendo el miedo a sufrir un accidente sin asistencia. Asimismo, la mayoría destacó que un dispositivo ligero, cómodo y fácil de usar es esencial para garantizar la adopción.

En general, los adultos mayores consideran que esta solución IoT puede mejorar su autonomía y calidad de vida, siempre que se prioricen el diseño ergonómico, la facilidad de uso y el acompañamiento inicial en el aprendizaje.

Análisis del segmento de cuidadores/familiares

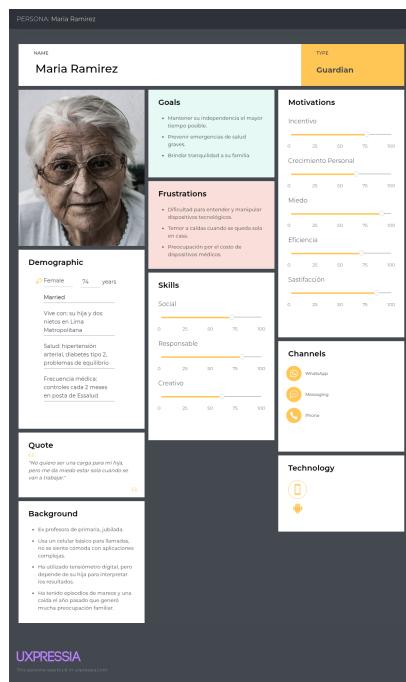
Los cuidadores entrevistados valoran de manera significativa la posibilidad de recibir alertas inmediatas en sus celulares en caso de caídas o alteraciones en los signos vitales de sus familiares. El 100% consideró que la función más importante es la respuesta rápida en emergencias, ya que actualmente dependen de llamadas telefónicas o visitas presenciales que no siempre son oportunas. Un 70% resaltó la utilidad de contar con un historial digital de signos vitales, que facilite la comunicación con médicos en controles regulares.

En general, los cuidadores perciben el sistema IoT como una herramienta clave para reducir la ansiedad y garantizar la seguridad de los adultos mayores, aunque insisten en la necesidad de optimizar la precisión, la usabilidad de la aplicación y el costo de implementación.

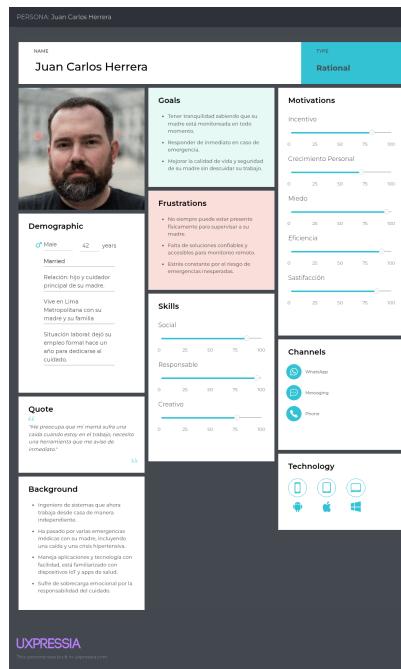
2.3. Needfinding

2.3.1. User Personas

a. User Persona de un adulto mayor



b. User Persona de un Cuidador



2.3.2. User Task Matrix

a. User Task Matriz a un adulto mayor

A continuación, se presenta una matriz de tareas enfocada en María Luisa Ramírez, una adulta mayor que representa al segmento de usuarios principales de la pulsera inteligente. Esta herramienta permite identificar y clasificar las actividades clave que realiza en su día a día, considerando la frecuencia con la que las lleva a cabo y el nivel de importancia que representan dentro de su vida cotidiana.

Tarea	María Luisa Ramírez	
	Frecuencia	Importancia
Medirse la presión arterial o pulso	Media	Alta
Asistir a controles médicos	Baja	Alta
Recordar tomar medicinas	Alta	Alta
Pedir ayuda en caso de mareos/caída	Baja	Muy Alta
Caminar dentro de casa o salir a comprar	Alta	Media
Comunicarse con su hija por teléfono	Media	Alta
Usar un dispositivo digital (celular/tensiómetro)	Baja	Media

Del análisis de la matriz de tareas de María Luisa Ramírez, se destaca que las actividades más críticas son recordar tomar medicinas, controlar signos vitales y pedir ayuda en caso de emergencia. Aunque no todas son frecuentes, tienen alta o muy alta importancia, ya que afectan directamente a su salud y seguridad. Una solución digital debe priorizar la automatización de alertas y la simplicidad de uso, garantizando que no dependa de interacciones complejas para mantenerse protegida.

b. User Task Matriz a un Cuidador

A continuación, se presenta una matriz de tareas enfocada en Juan Carlos Herrera, cuidador principal de su madre adulta mayor. Esta herramienta permite identificar y clasificar las actividades clave que realiza en su rol de cuidador, considerando la frecuencia con la que las lleva a cabo y el nivel de importancia que representan dentro de su rutina diaria.

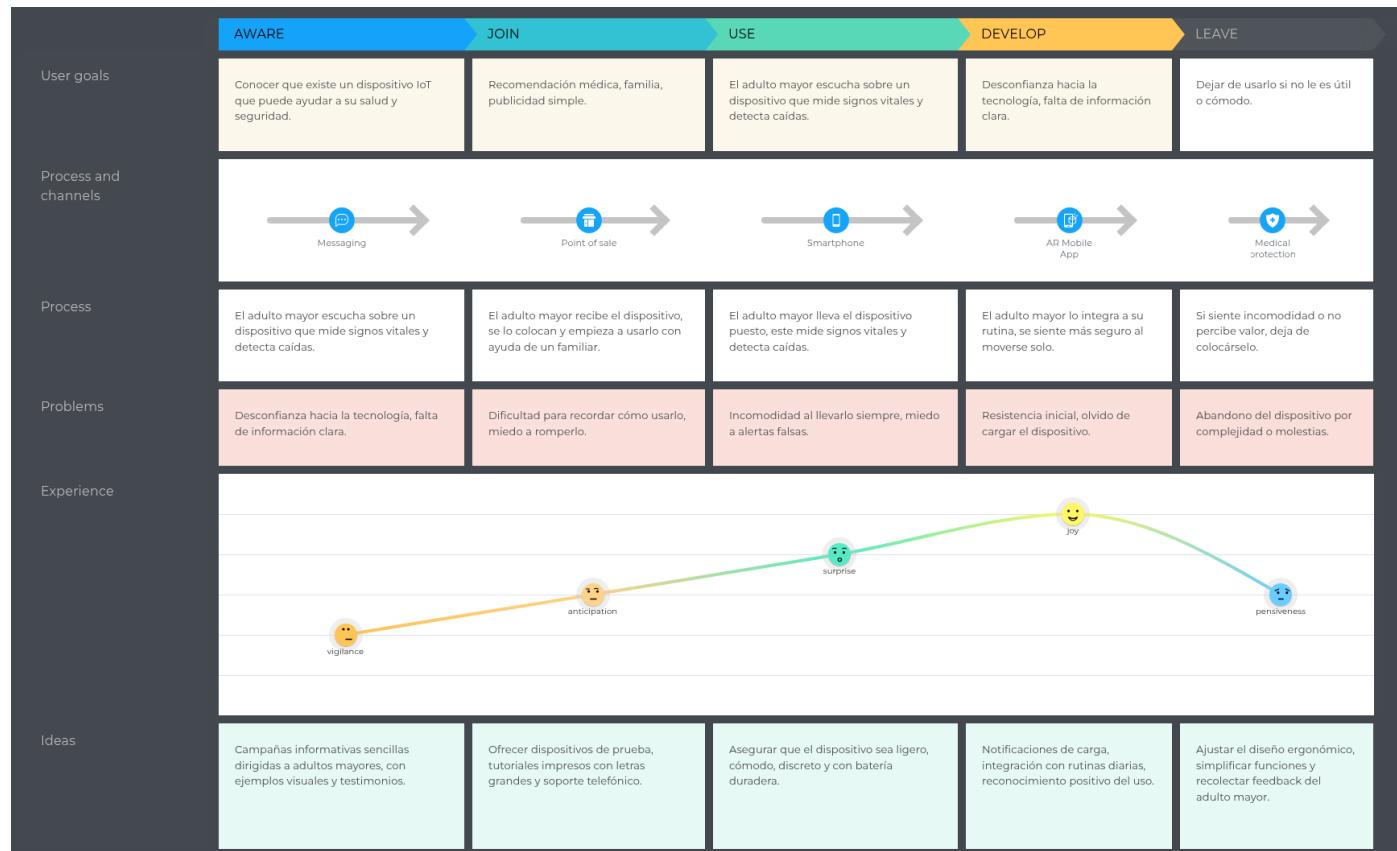
Tarea	Juan Carlos Herrera	
	Frecuencia	Importancia
Supervisar la salud de su madre	Alta	Muy Alta
Recibir alertas de emergencias	Baja (eventual)	Muy Alta
Organizar citas médicas y transporte	Media	Alta
Recordar y controlar medicación	Alta	Alta
Revisar reportes de signos vitales	Media	Alta
Coordinar con otros familiares el cuidado	Media	Media

Tarea	Juan Carlos Herrera	
	Frecuencia	Importancia
Trabajar de manera remota (equilibrio trabajo-cuidado)	Alta	Alta

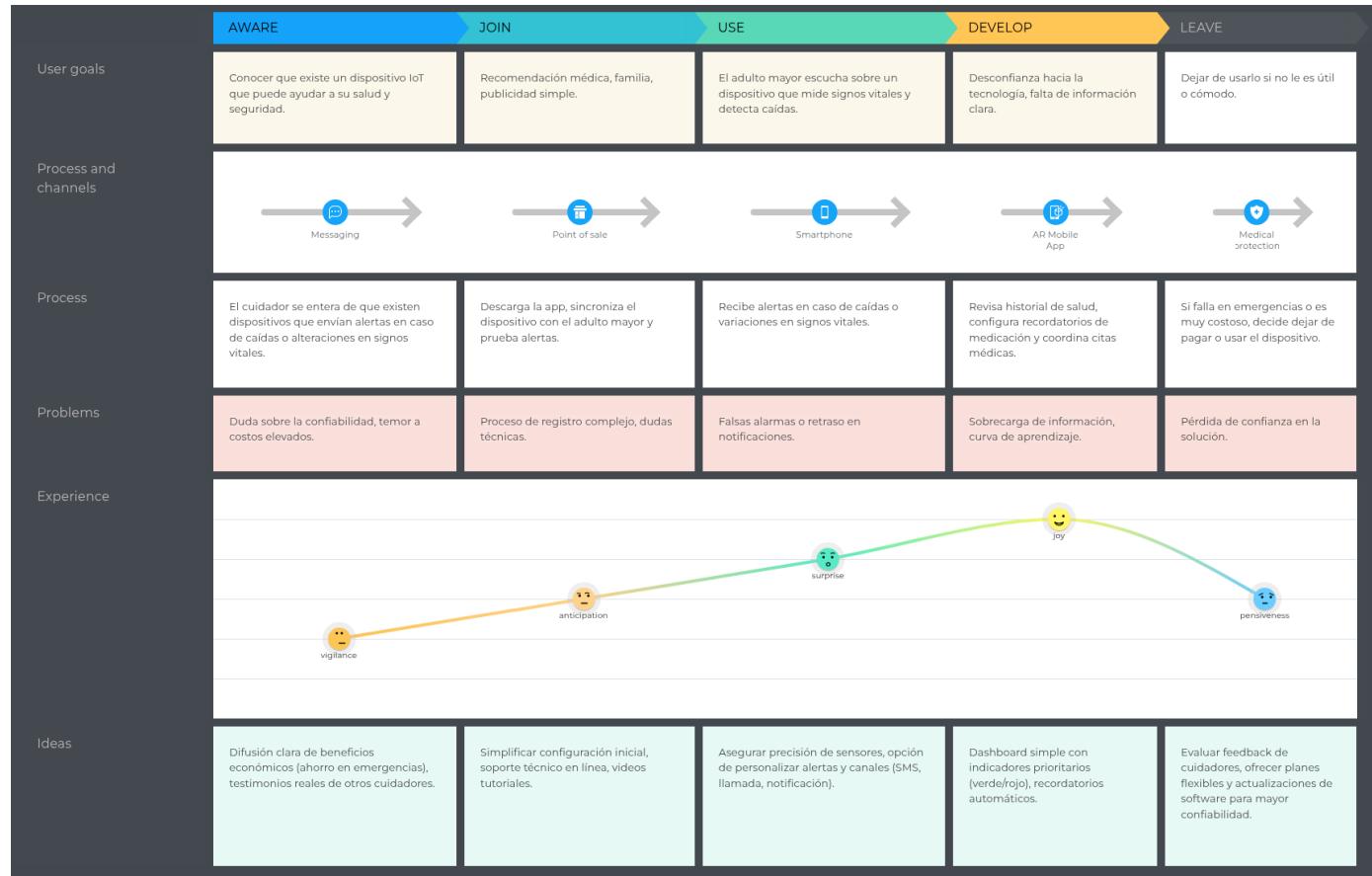
Del análisis de la matriz de tareas de Juan Carlos Herrera, se evidencia que las tareas de mayor relevancia son supervisar la salud, recibir alertas de emergencias y controlar la medicación. Aunque algunas (como emergencias) no son frecuentes, su impacto es crítico. Un sistema de monitoreo que brinde alertas inmediatas, reportes confiables y funciones de seguimiento en tiempo real es clave para brindarle tranquilidad, optimizar su tiempo y reducir la sobrecarga emocional asociada al cuidado.

2.3.3. User Journey Mapping

User Journey Map – Adulto Mayor



User Journey Map – Cuidador/Familiar



2.3.4. Empathy Mapping

User Journey Map – Adulto Mayor

PERSONA: Empathy map

1.WHO are we empathizing with?

María Ramírez, 74 años, jubilada, vive en Lima Metropolitana. Padece hipertensión y ha tenido dos caídas en los últimos años. Vive sola la mayor parte del día, aunque sus hijos la visitan con frecuencia.

7.What do they THINK and FEEL?

- “
- Piensa que la tecnología puede ayudarla, pero le preocupa no saber usarla bien.*
 - Siente miedo a perder autonomía y ser una carga para sus hijos.*
 - Valora mucho la tranquilidad y la seguridad en su vida diaria.*
- ”

2.What do they need to DO?

- Mantener control de su salud (presión, pulso, medicación).
- Prevenir caídas y poder pedir ayuda de inmediato.
- Sentirse acompañada y segura en su propio hogar.

6.What do they HEAR?

- Sus hijos le dicen que cuide su salud y que use aparatos modernos.
- Escucha que otras personas de su edad han tenido caídas graves y complicaciones.
- El médico recomienda controles más frecuentes de signos vitales.

**3.What do they SEE?**

- Observa que los dispositivos tecnológicos actuales son complejos y costosos.
- Nota que sus amigas y familiares a veces usan relojes inteligentes, pero ella siente que no son para su edad.
- En la televisión y en farmacias ve anuncios de aparatos médicos digitales, aunque no sabe si podría usarlos sola.

5.What do they DO?

- Usa tensiómetro manual ocasionalmente, con ayuda de un familiar.
- Toma sus medicinas, pero a veces olvida los horarios.
- Evita subir escaleras sola y reduce sus actividades por temor a accidentes.

PAINS

- Dificultad para usar tecnología.
- Temor a caídas sin asistencia inmediata.
- Olvidos en la medicación.
- Aparatos incómodos o invasivos.

GAINS

- Seguridad y confianza al estar sola.
- Recordatorios simples para medicación.
- Dispositivo cómodo, fácil de usar y accesible.
- Tranquilidad de que su familia reciba alertas en caso de emergencia.

4.What do they SAY?

“

- “Me da miedo caerme cuando estoy sola.”*
 - “Los aparatos nuevos son muy complicados para mí.”*
 - “Quisiera que mis hijos estén tranquilos cuando no me pueden visitar.”*
- ”

UXPRESSIA

This persona was built in upressoia.com

User Journey Map – Cuidador/Familiar

PERSONA: Empathy map

1.WHO are we empathizing with?

Juan Carlos Herrera, 42 años, trabaja como ingeniero y cuida de su madre (María) en sus tiempos libres. Vive en Lima y comparte responsabilidades con su hermana, aunque él es quien recibe las llamadas en emergencias.

6.What do they HEAR?

- Los médicos le insisten en la importancia de monitorear los signos vitales.
- Su madre expresa miedo a usar tecnología complicada.
- Sus colegas y amigos le comentan sobre relojes inteligentes, apps de salud y telemedicina.

7.What do they THINK and FEEL?

- “*
- Piensa que necesita una solución práctica y confiable para cuidar a su madre.
 - Siente ansiedad cuando está en el trabajo y no sabe si su madre está bien.
 - Valora la tranquilidad, la seguridad y poder equilibrar sus responsabilidades familiares con el trabajo.
- ”*

**2.What do they need to DO?**

- Asegurarse de que su madre tome la medicación correctamente.
- Monitorear signos vitales sin necesidad de ir al hospital frecuentemente.
- Recibir alertas inmediatas en caso de caídas o emergencias.

3.What do they SEE?

- Observa que los sistemas médicos privados ofrecen dispositivos costosos.
- Nota que en el mercado existen relojes inteligentes, pero no siempre se adaptan a adultos mayores.
- Siente que la tecnología médica aún no está pensada para familias de ingresos medios.

5.What do they DO?

- Llama varias veces al día para preguntar cómo está su madre.
- La acompaña a controles médicos y organiza sus citas.
- Busca en internet dispositivos de salud y compara precios.

PAINS

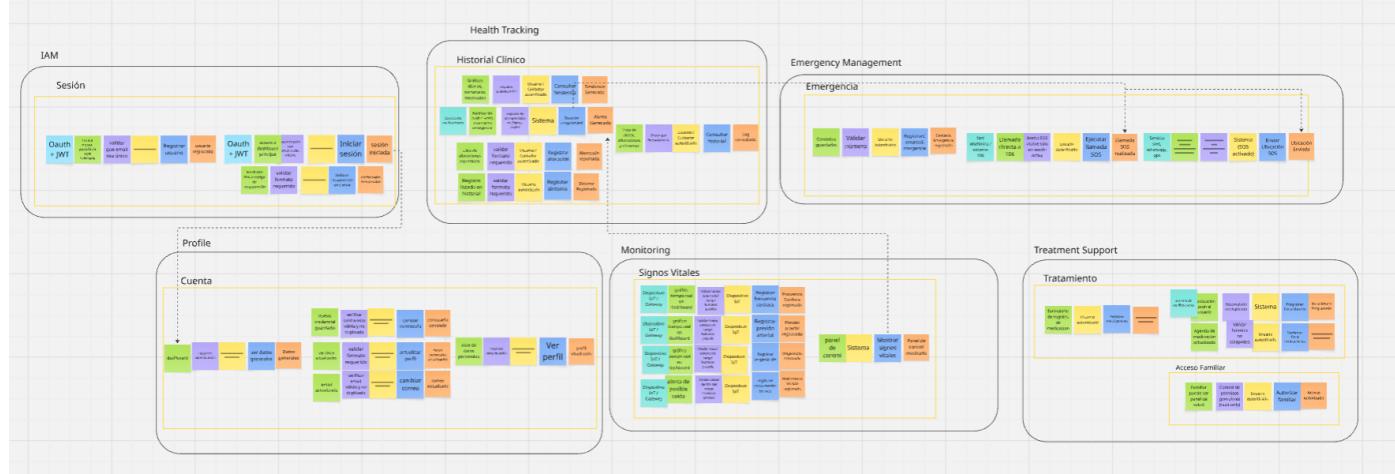
- Preocupación constante cuando no puede estar con su madre.
- Falsas alarmas o dispositivos difíciles de configurar.
- Altos costos de dispositivos médicos avanzados.
- Sobrecarga de información técnica innecesaria.

GAINS

- Recibir notificaciones claras y en tiempo real.
- Dispositivo accesible y fácil de usar para su madre.
- Historial de signos vitales confiable para compartir con médicos.
- Mayor tranquilidad emocional y reducción de emergencias graves.

4.What do they SAY?

- “*
- “Necesito saber si mi mamá está bien cuando no estoy con ella.”
 - “Me preocupa que se caiga y nadie se entere a tiempo.”
 - “La tecnología debería ayudarme a cuidar mejor sin complicarle la vida a mi mamá.”
- ”*



2.5. Ubiquitous Language

Gestión de signos vitales (Vital Signs Management): Proceso de recolección, almacenamiento y seguimiento de datos biométricos (como frecuencia cardíaca, saturación de oxígeno, presión arterial) provenientes de dispositivos IoT.

Alertas de irregularidades (Irregularity Alerts): Notificaciones generadas automáticamente cuando un signo vital registrado se encuentra fuera de los rangos normales establecidos.

Usuario (User): Persona que utiliza el sistema, ya sea un paciente que porta el dispositivo IoT, un cuidador o un profesional de la salud que accede al sistema.

Autenticación (Authentication): Mecanismo mediante el cual un usuario valida su identidad en el sistema para acceder a las funcionalidades.

SOS: Señal de emergencia enviada manualmente por el usuario o automáticamente por el sistema en caso de detectar una condición crítica.

Contacto de emergencia (Emergency Contact): Persona previamente registrada que recibe notificaciones o alertas en caso de que el sistema detecte una emergencia.

Registrar signo vital (Record Vital Sign): Acción de almacenar en el sistema una nueva medición de un parámetro biométrico tomada por el dispositivo IoT.

Panel de control (Dashboard): Interfaz central donde los usuarios autorizados pueden visualizar datos, alertas, métricas y el estado del paciente en tiempo real.

Confirmación / Cancelación (Confirmation / Cancellation): Respuesta del usuario o del sistema para validar o descartar una acción, como el envío de una alerta o un evento programado.

Token: Credencial digital utilizada para mantener sesiones seguras y autorizar accesos dentro del sistema tras un proceso de autenticación.

3. Capítulo III: Requirements Specification

3.1. User Stories

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
EP001	Autenticación y Gestión de Cuenta	Funcionalidades relacionadas con el acceso inicial del usuario (registro, inicio de sesión y recuperación).		
US01	Registrar usuario	Como adulto mayor o cuidador, quiero registrarme en la aplicación para acceder a los servicios de monitoreo.	<p>ES1: Registro exitoso → Al completar datos válidos y pulsar <i>Registrarme</i>, se confirma y redirige al inicio de sesión.</p> <p>ES2: Validaciones → Si el correo o contraseña son inválidos, muestra error y bloquea envío.</p> <p>ES3: Correo ya registrado → Si el correo existe, informa y ofrece <i>Recuperar cuenta</i>.</p>	EP001
US02	Iniciar sesión	Como usuario, quiero iniciar sesión para acceder a mi perfil y datos de salud.	<p>ES1: Inicio válido → Con credenciales correctas accede al panel principal.</p> <p>ES2: Credenciales inválidas → Muestra error y ofrece <i>Olvidé mi contraseña</i>.</p> <p>ES3: Cuenta no verificada → Sigue la verificación o informa bloqueo temporal.</p>	EP001
US03	Recuperar cuenta	Como usuario, quiero recuperar mi cuenta si olvido mis credenciales.	<p>ES1: Solicitud de restablecimiento → Envío de enlace al correo registrado.</p> <p>ES2: Correo no registrado → Mensaje genérico ("si existe el correo...").</p> <p>ES3: Enlace expirado → Indica vencimiento y permite generar uno nuevo.</p>	EP001

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
EP002	Gestión de Perfil de Usuario	Administración de datos personales, visualización y actualización de credenciales.		
US04	Ver perfil	Como usuario, quiero visualizar mi información personal registrada.	<p>ES1: Visualización correcta → Muestra datos completos al estar autenticado.</p> <p>ES2: Sin conexión → Muestra datos en caché con aviso de sincronización.</p> <p>ES3: Accesibilidad → Ajuste de fuentes sin pérdida de contenido.</p>	EP002
US05	Cambiar datos personales	Como usuario, quiero actualizar mis datos para mantener la información al día.	<p>ES1: Actualización exitosa → Cambios guardados y confirmados.</p> <p>ES2: Validaciones → Formatos incorrectos muestran error.</p> <p>ES3: Conflicto → Notifica cambios desde otro dispositivo y permite resolver.</p>	EP002
US06	Cambiar contraseña	Como usuario, quiero cambiar mi contraseña para mantener mi cuenta segura.	<p>ES1: Cambio exitoso → Actualiza credencial y cierra otras sesiones.</p> <p>ES2: Error → Si la actual o nueva no cumplen requisitos, muestra mensaje.</p> <p>ES3: Reautenticación → Solicitud contraseña actual por seguridad.</p>	EP002
US07	Cambiar correo	Como usuario, quiero actualizar mi correo electrónico.	<p>ES1: Cambio con verificación → Envía correo de confirmación antes de actualizar.</p> <p>ES2: Correo inválido/duplicado → Error y no se guarda.</p> <p>ES3: No verificación → Reversión al correo anterior tras vencimiento.</p>	EP002
EP003	Monitoreo de Signos Vitales	Visualización de frecuencia cardíaca, presión arterial y oxigenación en tiempo real.		
US08	Ver frecuencia cardíaca	Como usuario, quiero ver mi frecuencia cardíaca en tiempo real.	<p>ES1: Lectura en tiempo real → Muestra BPM y estado de conexión.</p> <p>ES2: Sin conexión → Último valor conocido y sugerencia de reconexión.</p> <p>ES3: Filtros → Permite seleccionar intervalos de tiempo.</p>	EP003
US09	Ver presión arterial	Como usuario, quiero consultar mi presión arterial.	<p>ES1: Lectura correcta → Muestra sistólica y diastólica.</p> <p>ES2: Unidad y calibración → Indica mmHg y última calibración.</p> <p>ES3: Falta parcial → Muestra aviso si falta un valor.</p>	EP003
US10	Ver oxigenación	Como usuario, quiero ver mis niveles de oxigenación (SpO ₂).	<p>ES1: Consulta correcta → Muestra nivel SpO₂ con calidad de señal.</p> <p>ES2: Alerta visual → Color de advertencia bajo umbral.</p> <p>ES3: Ahorro energía → Indica menor frecuencia de lectura.</p>	EP003
US11	Alertas inteligentes	Como usuario, quiero recibir notificaciones automáticas al detectar valores fuera de rango.	<p>ES1: Alerta a usuario → Envía notificación con valor y recomendación.</p> <p>ES2: Alerta a contactos → Envía aviso a contactos ante anomalías graves.</p> <p>ES3: Evitar ruido → Agrupa eventos frecuentes para no saturar.</p>	EP003
US12	Tendencias gráficas	Como usuario, quiero ver gráficos históricos de mis signos vitales.	<p>ES1: Rango temporal → Visualización diaria, semanal o mensual.</p> <p>ES2: Filtros → Recalcula estadísticas según rango.</p> <p>ES3: Exportación → Genera PDF o CSV con los datos.</p>	EP003
EP004	Registro y Seguimiento de alteraciones	Registro, consulta y seguimiento de alteraciones detectadas o reportadas.		

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US13	Reportar una alteración	Como usuario o cuidador, quiero registrar manualmente una alteración de salud.	ES1: Reporte exitoso → Guarda tipo, severidad y descripción. ES2: Adjuntos → Permite agregar foto o nota de voz. ES3: Validaciones → No permite guardar con campos vacíos.	EP004
US14	Log de alteraciones	Como usuario o cuidador, quiero consultar el historial de alteraciones.	ES1: Consulta → Lista con fecha, hora y detalle. ES2: Filtros → Permite buscar por severidad o fecha. ES3: Detalle → Muestra historial de cambios y adjuntos.	EP004
US15	Registro de síntomas	Como usuario, quiero registrar síntomas adicionales.	ES1: Alta → Guarda síntoma con fecha y notas. ES2: Historial → Muestra síntomas en orden cronológico. ES3: Edición → Permite corregir o eliminar con trazabilidad.	EP004
EP005	Emergencias y Alerta	Acciones ante situaciones críticas, llamadas al 106 y contactos adicionales.		
US16	Mandar llamado a emergencia 106	Como usuario, quiero que la app realice una llamada directa al 106 en caso de emergencia.	ES1: Llamada automática → Botón SOS inicia llamada con cuenta regresiva. ES2: Sin permisos → Solicitud permiso o muestra número manual. ES3: Confirmación → Pide doble toque para evitar errores.	EP005
US17	Registrar números de emergencia adicionales	Como usuario, quiero registrar contactos de emergencia.	ES1: Registro → Guarda nombre, relación y número. ES2: Validación → Evita duplicados o formato inválido. ES3: Prioridad → Ordena contactos por nivel de aviso.	EP005
US18	Geolocalización en emergencias	Como usuario, quiero enviar mi ubicación al activar SOS.	ES1: Envío con llamada → Envía ubicación al 106 y contactos. ES2: Notificación → Envía SMS/WhatsApp con enlace de seguimiento. ES3: Sin permisos → Usa última ubicación conocida.	EP005
EP006	Soporte al Tratamiento	Funcionalidades de apoyo social o médico como recordatorios y acceso multiusuario.		
US19	Recordatorios de medicamentos	Como usuario, quiero programar recordatorios para no olvidar medicación.	ES1: Notificación → Envía alerta a la hora programada. ES2: Registro → Marca como tomado u omitido. ES3: Reprogramación → Permite posponer o registrar omisión.	EP006
US20	Acceso multiusuario	Como usuario, quiero autorizar a familiares o cuidadores a ver mi estado de salud.	ES1: Conceder acceso → Registra contacto y define permisos. ES2: Revocar acceso → Permite modificar o eliminar permisos. ES3: Privacidad → Comparte solo la información autorizada.	EP006
EP007	Landing Page Informativa	Este Epic abarca las secciones públicas de la aplicación web orientadas a captar nuevos usuarios y comunicar valor: características, beneficios, problemática, acerca de, testimonios, contacto y descarga.		
US21	Ver características	Como visitante, quiero conocer las principales características de la aplicación para entender qué ofrece.	ES1: Carga de sección → Al entrar al sitio, se muestran iconos y descripciones de funcionalidades clave. ES2: Accesibilidad → Las tarjetas son legibles, con texto alternativo e íconos descriptivos. ES3: Navegación → Los enlaces de "Ver más" dirigen correctamente a detalles o registro.	EP007
US22	Ver beneficios	Como visitante, quiero ver los beneficios de usar la aplicación para motivarme a registrarme.	ES1: Visualización clara → La sección muestra beneficios con frases breves y visuales. ES2: Responsive → El diseño se adapta correctamente en móvil y escritorio. ES3: Llamado a la acción → Incluye botón "Regístrate" o "Descarga la app" visible.	EP007

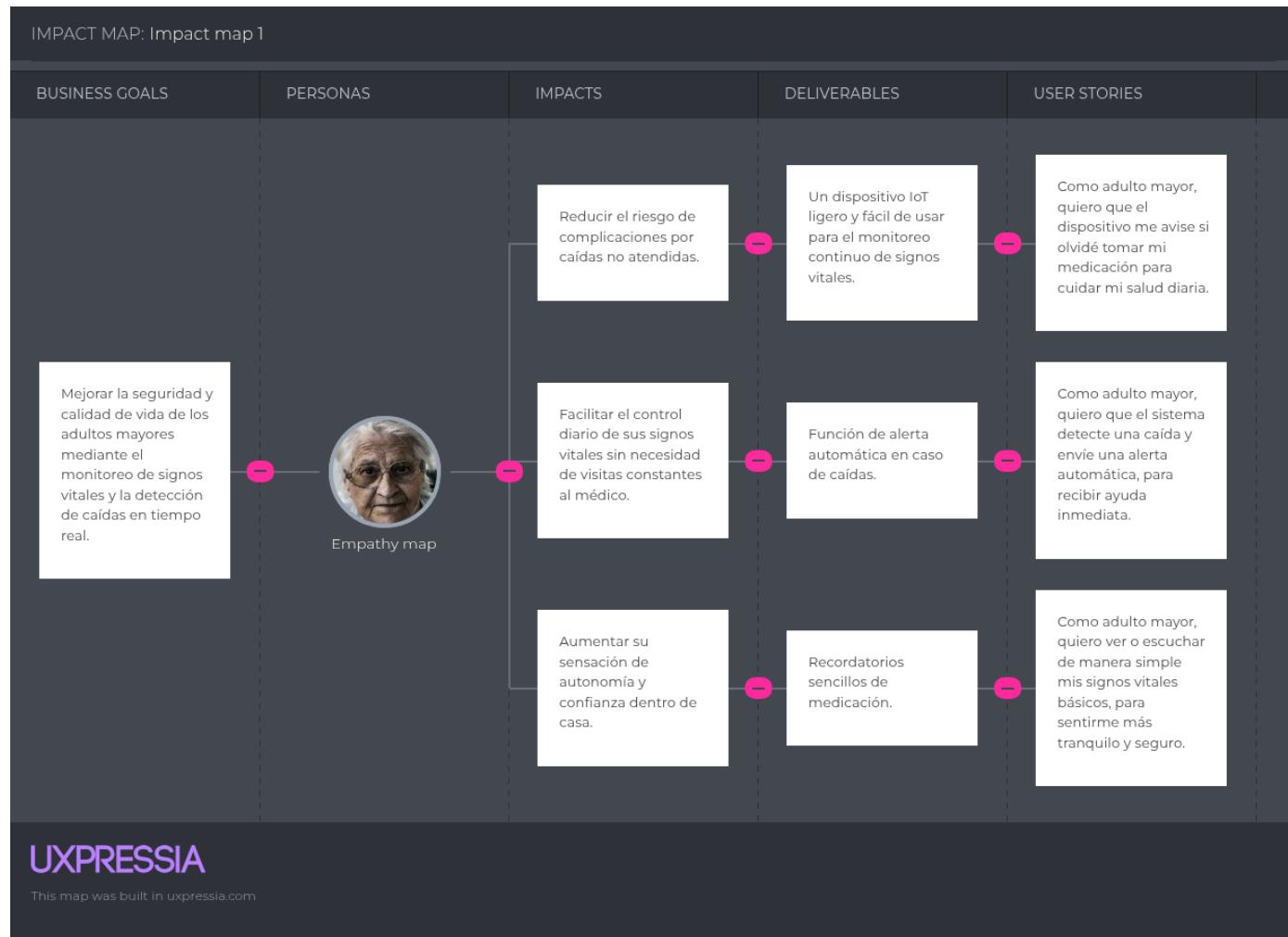
Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US23	Comprender la problemática	Como visitante, quiero entender la problemática de salud que aborda la aplicación para sentirme identificado con su propósito.	<p>ES1: Texto informativo → Se muestra explicación clara y empática de la problemática.</p> <p>ES2: Recursos visuales → Incluye imágenes o gráficos ilustrativos.</p> <p>ES3: Tiempo de carga → La sección carga sin retrasos perceptibles (<3s).</p>	EP007
US24	Conocer acerca del proyecto	Como visitante, quiero leer información sobre la misión, visión y el equipo detrás de la aplicación.	<p>ES1: Muestra datos clave → Presenta misión, visión y miembros principales del proyecto.</p> <p>ES2: Enlaces externos → Los botones de redes sociales o "Conócenos" funcionan correctamente.</p> <p>ES3: Diseño accesible → Compatible con lectores de pantalla y contraste adecuado.</p>	EP007
US25	Ver testimonios	Como visitante, quiero leer opiniones de otros usuarios o cuidadores para generar confianza.	<p>ES1: Visualización → Se muestran al menos tres testimonios con nombre y foto opcional.</p> <p>ES2: Carrusel funcional → Permite desplazarse sin errores entre testimonios.</p> <p>ES3: Validación de contenido → Los testimonios no se repiten y cargan dinámicamente.</p>	EP007
US26	Enviar mensaje de contacto	Como visitante, quiero enviar un mensaje a través del formulario de contacto para resolver mis dudas.	<p>ES1: Envío exitoso → Al completar nombre, correo y mensaje válidos, se confirma el envío.</p> <p>ES2: Validaciones → Si falta algún campo o el correo es inválido, se muestra mensaje de error.</p> <p>ES3: Protección antispam → No se permiten envíos automáticos (captcha activo).</p>	EP007
US27	Descargar la aplicación	Como visitante, quiero descargar la aplicación desde la landing page para comenzar a usarla.	<p>ES1: Enlaces funcionales Los botones de "Descargar en Play Store / App Store" redirigen correctamente.</p> <p>ES2: Detección de dispositivo Si accede desde Android o iOS, se sugiere el enlace correspondiente.</p> <p>ES3: Versión web Si el dispositivo no es compatible, se ofrece versión web o aviso informativo.</p>	EP007
EP008	Vinculación y Gestión de Dispositivos (Mobile App)	Funcionalidades de la aplicación móvil para vincular dispositivos IoT mediante códigos y gestionar la conexión con pacientes.		
US28	Generar código de vinculación	Como cuidador, quiero generar un código único de vinculación para asociar el dispositivo IoT del paciente a mi cuenta y comenzar a recibir sus datos de monitoreo.	<p>ES1: Generación exitosa Al presionar "Generar código", se crea un código alfanumérico único de 6 caracteres con vigencia de 24 horas.</p> <p>ES2: Visualización clara El código se muestra en pantalla con opción de copiar y compartir por WhatsApp/SMS.</p> <p>ES3: Expiración Si el código expira sin usarse, se permite generar uno nuevo.</p>	EP008
US29	Usar código de vinculación	Como paciente o familiar, quiero ingresar un código de vinculación para conectar mi dispositivo IoT con la cuenta del cuidador asignado.	<p>ES1: Vinculación exitosa Al ingresar un código válido y presionar "Vincular", se establece la relación dispositivo-cuidador y muestra confirmación.</p> <p>ES2: Código inválido/expirado Muestra mensaje de error indicando que el código no existe o ha expirado.</p> <p>ES3: Código ya usado Informa que el código ya fue utilizado y sugiere solicitar uno nuevo.</p>	EP008
EP009	Monitoreo en Tiempo Real (Mobile App)	Funcionalidades de visualización de datos de signos vitales en tiempo real mediante conexión WebSocket.		
US30	Recibir datos de signos vitales en tiempo real	Como cuidador, quiero visualizar los signos vitales del paciente actualizados en tiempo real mediante WebSocket para tener información actualizada constantemente sin necesidad de refrescar la pantalla.	<p>ES1: Conexión WebSocket activa Al abrir el dashboard, se establece conexión WebSocket y los datos de frecuencia cardíaca, SpO y temperatura se actualizan automáticamente cada 5 segundos.</p> <p>ES2: Indicador de conexión Muestra estado de conexión (conectado/desconectado) con código de colores (verde/rojo).</p> <p>ES3: Reconexión automática Si se pierde la conexión, intenta reconectar automáticamente cada 10 segundos y notifica al usuario.</p>	EP009

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
EP010	Ubicación y Seguimiento (Mobile App)	Funcionalidades para visualizar y rastrear la ubicación del paciente en situaciones de emergencia o monitoreo.		
US31	Ver ubicación del paciente en mapa	Como cuidador, quiero ver la ubicación actual del paciente en un mapa interactivo para saber dónde se encuentra en caso de emergencia o seguimiento rutinario.	<p>ES1: Visualización de ubicación Al acceder a la sección "Ubicación", se muestra un mapa con la posición actual del paciente marcada con un pin.</p> <p>ES2: Actualización periódica La ubicación se actualiza cada 30 segundos con indicador de última actualización.</p> <p>ES3: Sin señal GPS Si no hay ubicación disponible, muestra la última conocida con advertencia de antigüedad.</p>	EP010
Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
EP01	Backend APIs y Servicios	Endpoints y lógica del lado servidor para autenticación, gestión de entidades clínicas y relaciones entre pacientes y cuidadores.		
TS01	Login (API)	Endpoint REST para autenticación de usuarios (email/password) y emisión de JWT.	<p>ES1: POST /api/auth/login → devuelve token JWT y datos básicos del usuario.</p> <p>ES2: Credenciales inválidas → 401 con mensaje genérico.</p> <p>ES3: Token válido → accesible recursos protegidos.</p>	EP01
TS02	Registrar cuidador (Backend)	Endpoint para crear cuenta de tipo "caregiver" y perfil asociado.	<p>ES1: POST /api/auth/register-caregiver crea User + Profile + Caregiver y devuelve 201.</p> <p>ES2: Email duplicado → 409 y mensaje.</p> <p>ES3: Validaciones de campos (dni, teléfono) aplicadas.</p>	EP01
TS03	Registrar paciente (Backend)	Endpoint para crear cuenta/registro de paciente (puede ser por caretakers).	<p>ES1: POST /api/patients crea Patient y Profile y devuelve 201.</p> <p>ES2: Si es creado por caregiver, se establece relación N:M inicial.</p> <p>ES3: Validaciones de datos clínicos básicas.</p>	EP01
TS04	Vincular paciente ↔ cuidador	Endpoint para relacionar un paciente con un cuidador (adds relation).	<p>ES1: POST /api/caregivers/{id}/patients con patientId → devuelve 200 y relación creada.</p> <p>ES2: Si ya existe relación → 409.</p> <p>ES3: Permisos: solo caregiver autenticado o admin puede vincular.</p>	EP01
TS05	CRUD Disturbances (Alteraciones)	Endpoints para crear, listar y eliminar alteraciones detectadas (disturbances).	<p>ES1: POST /api/patients/{id}/disturbances crea entrada y devuelve 201.</p> <p>ES2: GET /api/patients/{id}/disturbances devuelve lista paginada.</p> <p>ES3: DELETE /api/disturbances/{id} borra (soft delete) con permisos adecuados (caregiver/patient/admin).</p>	EP01
TS06	CRUD Symptoms (Síntomas)	Endpoints para reportar, consultar y eliminar síntomas asociados a pacientes.	<p>ES1: POST /api/patients/{id}/symptoms → 201 con id del síntoma.</p> <p>ES2: GET /api/patients/{id}/symptoms → lista ordenada por fecha.</p> <p>ES3: DELETE /api/symptoms/{id} → 200 y registro marcado inactivo.</p>	EP01
TS07	CRUD Treatments (Tratamientos)	Endpoints para gestionar tratamientos y recordatorios (create/get/delete).	<p>ES1: POST /api/patients/{id}/treatments crea tratamiento y devuelve 201.</p> <p>ES2: GET /api/patients/{id}/treatments devuelve activos/inactivos filtrables.</p> <p>ES3: DELETE /api/treatments/{id} desactiva o elimina según política.</p>	EP01
TS10	Device Controller - Registrar dispositivo	Endpoint REST para registrar y vincular un dispositivo IoT a un paciente.	<p>ES1: POST /api/device valida payload (deviceID, patientID, model, firmware) y devuelve 201.</p> <p>ES2: Duplicado → 409 con mensaje.</p> <p>ES3: Publica evento de dispositivo registrado.</p>	EP01

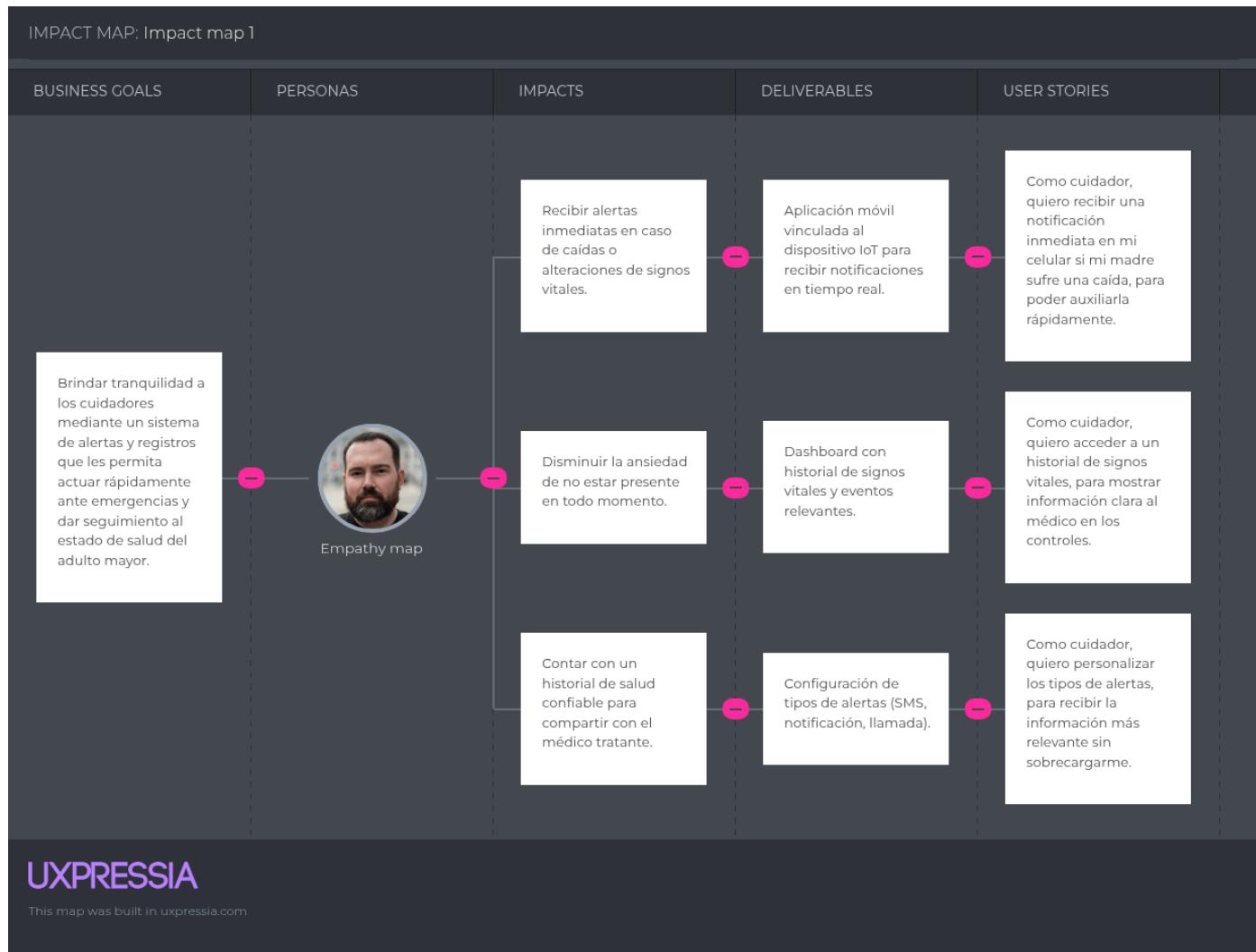
Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
TS11	Device Controller - Obtener lecturas	Endpoint REST para obtener todas las lecturas recopiladas por los dispositivos.	ES1: GET /api/device/readings/all soporta filtros (patientId, deviceld, from, to) y paginación. ES2: Protege ruta por roles/permisos. ES3: Devuelve lista paginada con metadatos.	EP01
EP02	Backend - Gestión de Códigos y Perfil (Sprint 3)	Endpoints para gestión de códigos de vinculación, información de perfil y detección de eventos críticos.		
TS12	API de códigos de vinculación - Generar	Endpoint para generar códigos únicos de vinculación entre cuidadores y dispositivos/pacientes.	ES1: POST /api/invite/generate crea código alfanumérico de 6 caracteres con expiración de 24h y devuelve 201. ES2: Límite de códigos activos → 400 si el usuario tiene más de 5 códigos sin usar. ES3: Incluye metadata (createdAt, expiresAt, createdBy).	EP02
TS13	API de códigos de vinculación - Usar	Endpoint para consumir y validar códigos de vinculación.	ES1: POST /api/invite/use/{code} valida código, establece relación y devuelve 200 con datos de vinculación. ES2: Código inválido/expirado → 404 con mensaje descriptivo. ES3: Código ya usado → 409 indicando uso previo.	EP02
TS14	API de perfil de usuario	Endpoints para obtener y actualizar información de perfil de pacientes y cuidadores.	ES1: GET /api/auth/profile/me devuelve datos completos del usuario autenticado (nombre, email, rol, foto, teléfono). ES2: PUT /api/auth/profile actualiza campos permitidos y devuelve 200. ES3: Validaciones de formato (teléfono, email) aplicadas con mensajes claros.	EP02
TS15	Sistema de detección de caídas	Lógica backend para procesar datos del acelerómetro y detectar caídas automáticamente.	ES1: POST /api/falls/detect recibe datos de acelerómetro, aplica algoritmo de detección y devuelve resultado (fall_detected: true/false). ES2: Si se detecta caída → crea alerta automática en /api/alerts con severidad "critical" y notifica contactos de emergencia. ES3: Registra evento en historial con timestamp, ubicación y datos del sensor de trazabilidad.	EP02

3.2. Impact Mapping

User Journey Map – Adulto Mayor



User Journey Map – Cuidador/Familiar



3.3. Product Backlog

Esta tabla muestra una lista estructurada de las funcionalidades principales que deben desarrollarse para el sistema IOT, organizadas conforme a la prioridad del negocio y las necesidades del usuario. Las características se clasifican en User Stories (historias de usuario) y Technical Stories (historias técnicas), y se ordenan según su importancia para asegurar que las más relevantes se atiendan primero.

La intención de esta organización es facilitar una implementación eficaz, dando prioridad a las funciones que aportan mayor valor tanto al negocio como a la experiencia del usuario. Entre ellas se incluyen la administración de pacientes, registro de tratamientos, síntomas y recordatorios, vinculación de cuidador a paciente y otros elementos fundamentales para el funcionamiento de los sensores IOT. También se considera el esfuerzo requerido para cada actividad, expresado en Story Points, con el fin de optimizar la entrega de valor en el menor tiempo posible.

Así, se define un plan de trabajo claro para el equipo de desarrollo, garantizando que las funcionalidades más importantes estén disponibles desde las primeras fases del proyecto.

Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
1	US11	Alertas inteligentes	Recibir notificaciones automáticas cuando los signos vitales salgan de los rangos normales.	5
2	US18	Geolocalización en emergencias	Enviar ubicación en tiempo real al activar el botón SOS.	5
3	US12	Tendencias gráficas	Visualizar gráficos históricos de los signos vitales por día, semana o mes.	5
4	US20	Acceso multiusuario	Autorizar a familiares o cuidadores para ver el estado de salud en tiempo real.	5
5	US19	Recordatorios de medicamentos	Programar notificaciones para recordar la toma de medicamentos.	3
6	US16	Mandar llamado a emergencia 106	Llamar automáticamente al 106 desde la app en caso de emergencia.	3
7	US17	Registrar números de emergencia adicionales	Añadir contactos de emergencia adicionales para notificación en caso de SOS.	3
8	US15	Registro de síntomas	Registrar manualmente síntomas adicionales y consultarlos en un historial.	3

Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
9	US13	Reportar una alteración	Reportar manualmente una alteración en la salud.	2
10	US14	Log de alteraciones	Consultar el historial de alteraciones registradas.	2
11	US05	Cambiar datos personales	Actualizar datos personales en el perfil.	2
12	US06	Cambiar contraseña	Actualizar la contraseña desde el perfil.	2
13	US07	Cambiar correo	Actualizar el correo electrónico para recibir notificaciones correctas.	2
14	US08	Ver frecuencia cardiaca	Visualizar la frecuencia cardiaca en tiempo real.	2
15	US09	Ver presión arterial	Consultar los valores de presión arterial en el panel de signos vitales.	2
16	US10	Ver oxigenación	Consultar el nivel de oxigenación (SpO ₂) en tiempo real.	2
17	US01	Registrar usuario	Registrarse en la aplicación para acceder a los servicios de monitoreo.	1
18	US02	Iniciar sesión	Acceder al perfil y datos de salud con usuario y contraseña.	1
19	US03	Recuperar cuenta	Recuperar la cuenta en caso de olvidar las credenciales.	1
20	US04	Ver perfil	Visualizar información personal registrada en la aplicación.	1
21	US21	Ver características	Mostrar las principales funcionalidades de la aplicación en la landing page.	1
22	US22	Ver beneficios	Presentar los beneficios del uso de la aplicación para incentivar el registro.	1
23	US23	Comprender la problemática	Explicar la problemática de salud que aborda la aplicación para generar empatía.	1
24	US24	Conocer acerca del proyecto	Mostrar misión, visión y equipo responsable del desarrollo de la aplicación.	1
25	US25	Ver testimonios	Visualizar experiencias reales de usuarios o cuidadores para generar confianza.	2
26	US26	Enviar mensaje de contacto	Permitir al visitante enviar un mensaje o consulta mediante formulario de contacto.	3
28	TS01	Login (API)	Endpoint REST para autenticación de usuarios (email/password) y emisión de JWT.	3
29	TS02	Registrar cuidador (Backend)	Endpoint para crear cuenta de tipo "caregiver" y perfil asociado (User + Profile + Caregiver).	5
30	TS03	Registrar paciente (Backend)	Endpoint para crear cuenta/registro de paciente (puede ser creado por caregivers) y establecer relaciones iniciales.	5
31	TS04	Vincular paciente ↔ cuidador	Endpoint para relacionar un paciente con un cuidador (crear relación N:M) con validaciones y permisos.	3
32	TS05	CRUD Disturbances (Alteraciones)	Endpoints para crear, listar, filtrar y eliminar alteraciones detectadas por el sistema (soft delete, paginación).	5
33	TS06	CRUD Symptoms (Síntomas)	Endpoints para reportar, consultar y eliminar síntomas asociados a pacientes (incluye adjuntos y trazabilidad).	3
34	TS07	CRUD Treatments (Tratamientos)	Endpoints para gestionar tratamientos y su ciclo de vida (create/get/update/delete) y relaciones con recordatorios.	5
35	TS08	Asociar recordatorios a Treatment	Backend para crear/listar recordatorios asociados a tratamientos y exponer scheduler/worker para notificaciones.	5
37	TS10	Device Controller - Registrar dispositivo (POST /api/device)	Endpoint REST para registrar y vincular un dispositivo IoT a un paciente. Valida payload (deviceld, patientId, model, firmware), crea/actualiza registro del dispositivo y publica evento de dispositivo registrado. Maneja duplicados y devuelve 201 con recurso creado.	3
38	TS11	Device Controller - Obtener lecturas (GET /api/device/readings/all)	Endpoint REST para obtener todas las lecturas recopiladas por los dispositivos. Soporta filtros (patientId, deviceld, from,to), paginación y ordenamiento. Protege ruta (roles/permiso), valida acceso y devuelve lista paginada de lecturas con metadatos.	5
39	US28	Generar código de vinculación	Como cuidador, quiero generar un código único de vinculación para asociar el dispositivo IoT del paciente a mi cuenta.	3
40	US29	Usar código de vinculación	Como paciente o familiar, quiero ingresar un código de vinculación para conectar mi dispositivo IoT con la cuenta del cuidador.	3
41	US30	Recibir datos de signos vitales en tiempo real	Como cuidador, quiero visualizar los signos vitales del paciente actualizados en tiempo real mediante WebSocket.	5

Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
42	US31	Ver ubicación del paciente en mapa	Como cuidador, quiero ver la ubicación actual del paciente en un mapa interactivo para emergencias o seguimiento.	3
43	TS12	API de códigos de vinculación - Generar	Endpoint POST /api/invite/generate para crear códigos únicos de vinculación con expiración de 24h.	3
44	TS13	API de códigos de vinculación - Usar	Endpoint POST /api/invite/use/{code} para consumir y validar códigos de vinculación.	3
45	TS14	API de perfil de usuario	Endpoints GET/PUT /api/auth/profile para obtener y actualizar información de perfil de pacientes y cuidadores.	3
46	TS15	Sistema de detección de caídas	Lógica backend para procesar datos del acelerómetro, detectar caídas automáticamente y generar alertas críticas.	5

4. Capítulo IV: Solution Software Design

4.1. Strategic-Level Domain-Driven Design

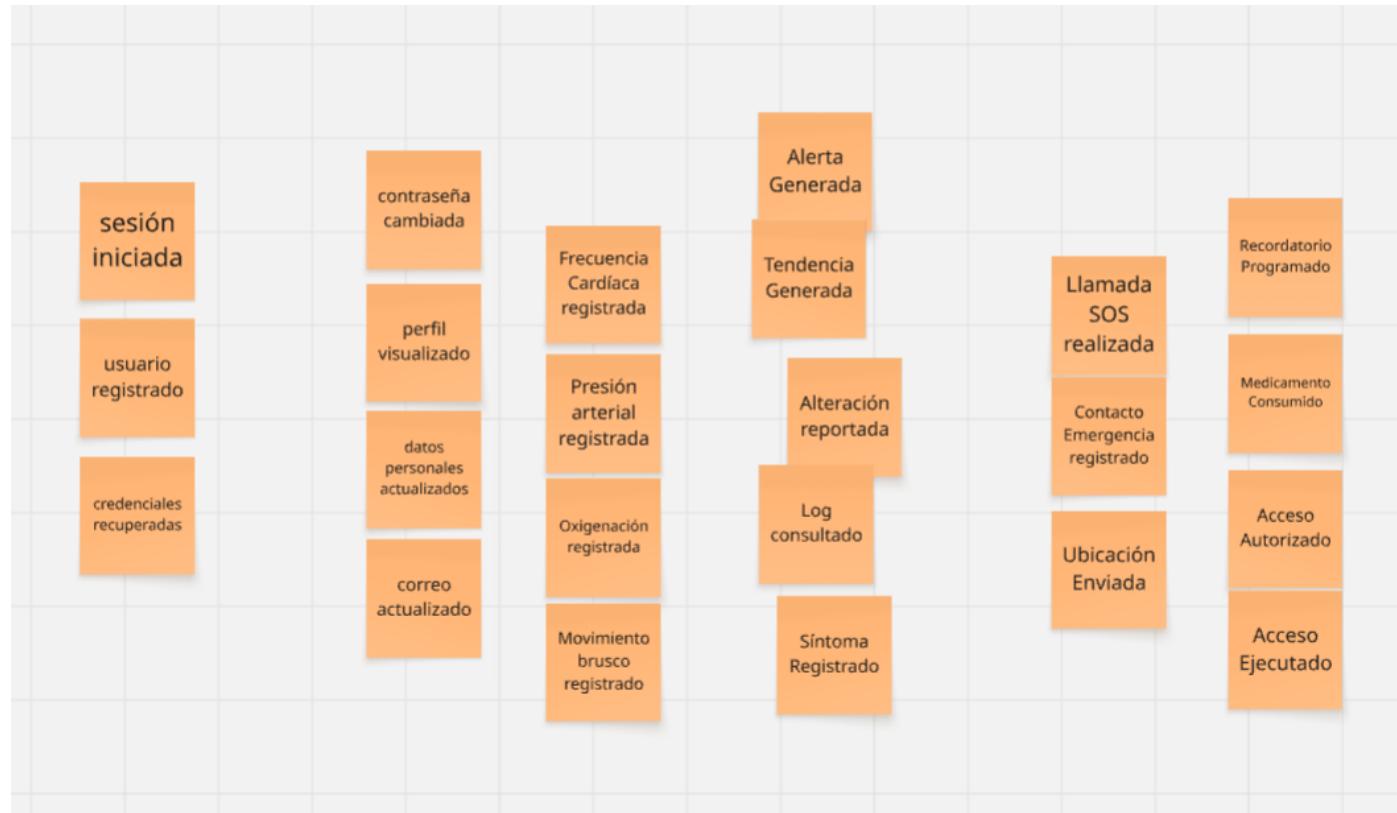
4.1.1. Design-Level EventStorming

Primero identificamos los eventos y trazamos mediante una línea imaginaria que va de izquierda a derecha.

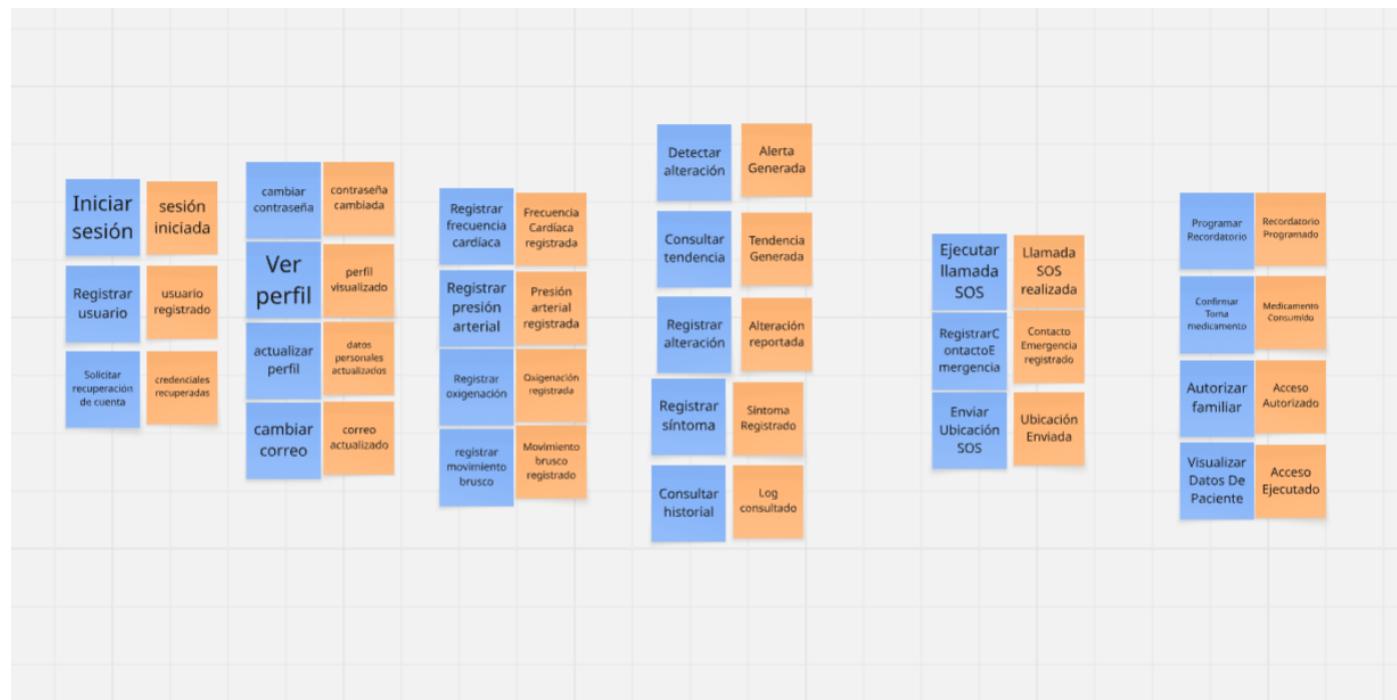
Algunos eventos están en la misma columna, ya que, el orden no es consecutivo por parte de estos. Identificamos a los eventos con un post-it de color anaranjado.

- Post It Anaranjado: Evento a realizar.
- Post It Azul: Comando que desata el evento de su derecha.
- Post It Amarillo: Usuario final que genera el comando de su derecha. Delimitaciones amarillas: Secciona eventos y comandos relacionados a un "aggregate".
- Post It Morado: Políticas y reglas de negocio.
- Post It Verde: referencia a modelos UI (lo que veerá el usuario)
- Post It Celeste: Agentes externos
- Delimitaciones de color negro: Secciona uno o más aggregates relacionados a un bounded context.
- Flechas punteadas: indicadores de comandos que generan eventos en distintos aggregates o bounded contexts.





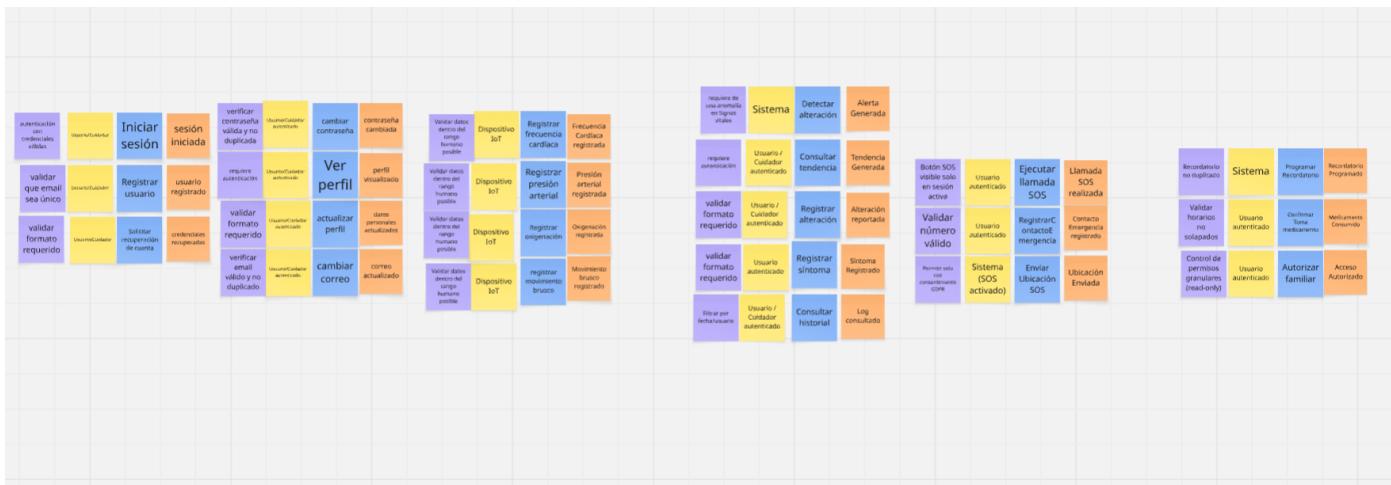
Como segundo paso, identificamos los comandos que llevan a cabo el evento. Identificamos a estos con un post-it de color azul.



Como tercer paso, identificamos los actores que realizan los comandos anteriormente establecidos.



Como cuarto paso, identificamos las políticas o reglas de negocio ligadas a los eventos.

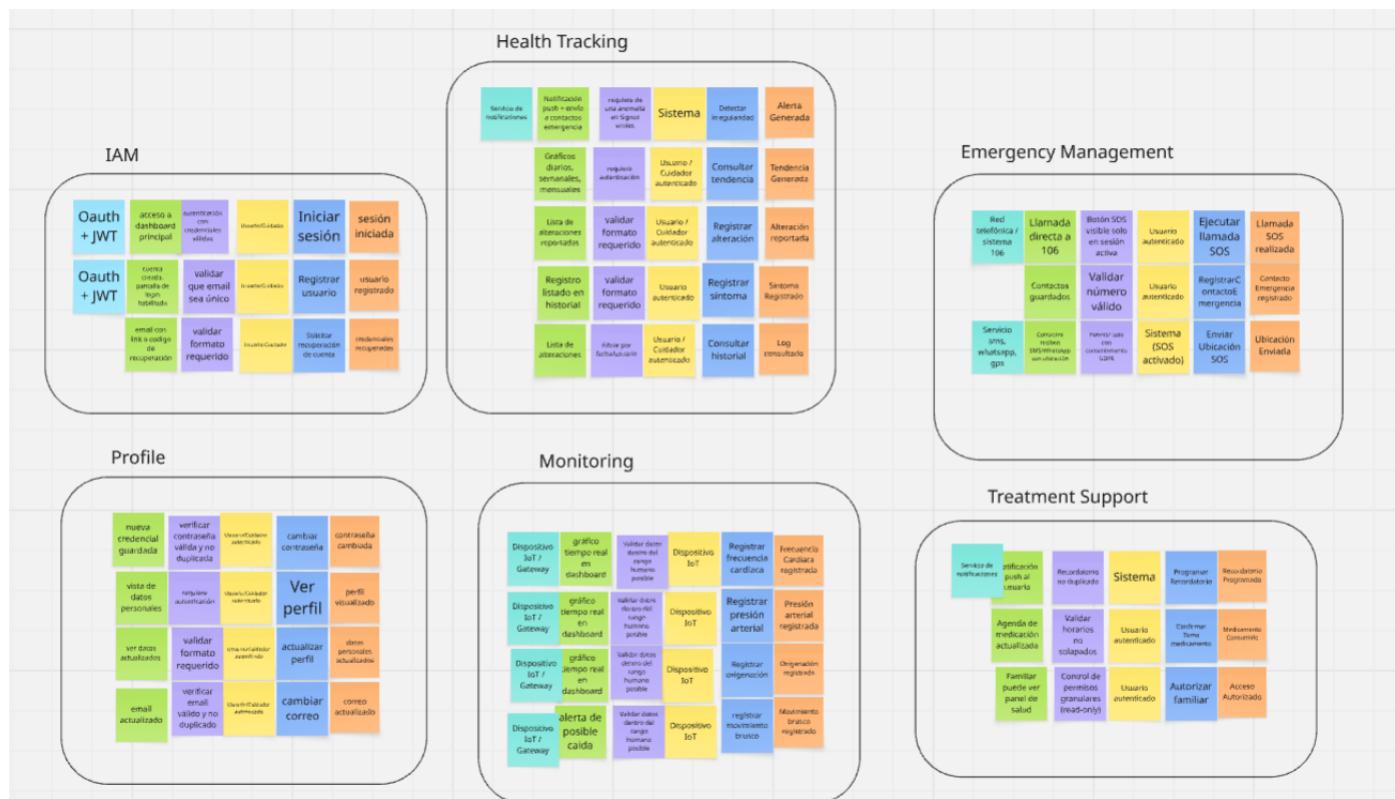


Como quinto paso enlazamos una pequeña descripción de lo que sería la representación de los datos al usuario final.

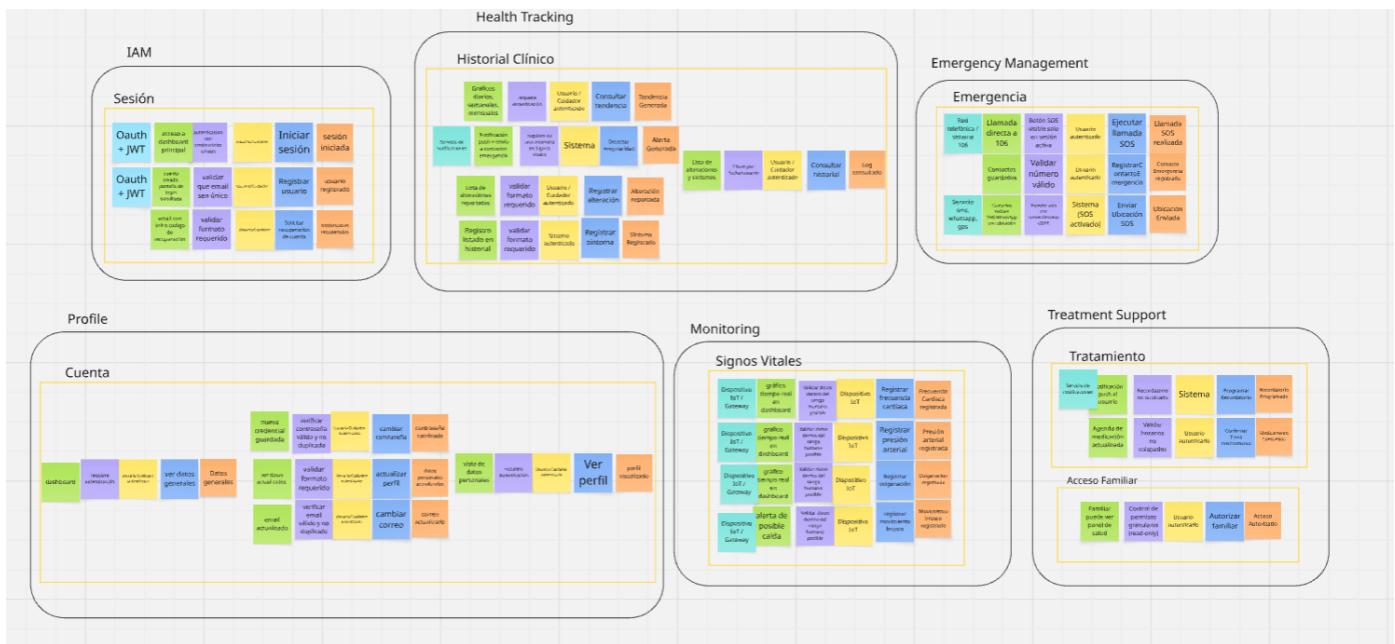


Llamada directa a 106	Botón SOS visible solo en sesión activa	Usuario autenticado	Ejecutar llamada SOS	Llamada SOS realizada	Notificación push al usuario	Recordatorio no duplicado	Sistema	Programar Recordatorio	Recordatorio Programado
Contactos guardados	Validar número válido	Usuario autenticado	Registrar Contacto Emergencia	Contacto Emergencia registrado	Agenda de medicación actualizada	Validar horarios no solapados	Usuario autenticado	Confirmar Toma medicamento	Medicamento Consumido
Contactos reciben SMS/WhatsApp con ubicación	Permitir solo con consentimiento GDPR	Sistema (SOS activado)	Enviar Ubicación SOS	Ubicación Enviada	Familiar puede ver panel de salud	Control de permisos granulares (read-only)	Usuario autenticado	Autorizar familiar	Acceso Autorizado

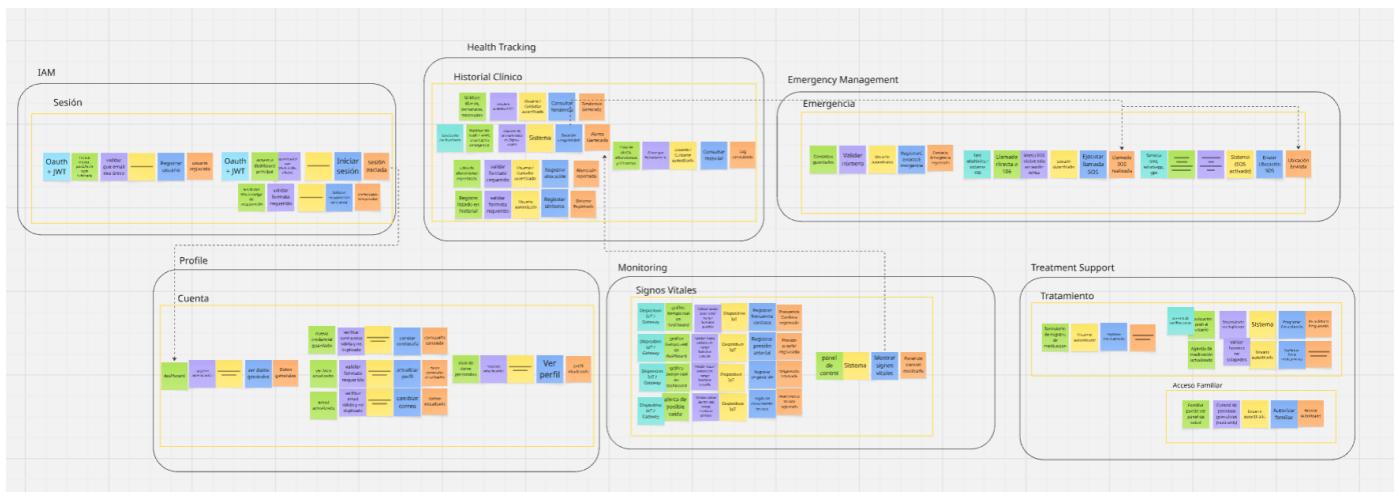
Como sexto paso identificamos los servicios externos y los segmentamos por Bounded Context.



Como séptimo paso, identificamos los agregados de cada bounded context



Como último paso determinamos las conexiones entre diferentes bounded context. Dichas relaciones se representan con una flecha punteada.



4.1.1.1. Candidate Context Discovery

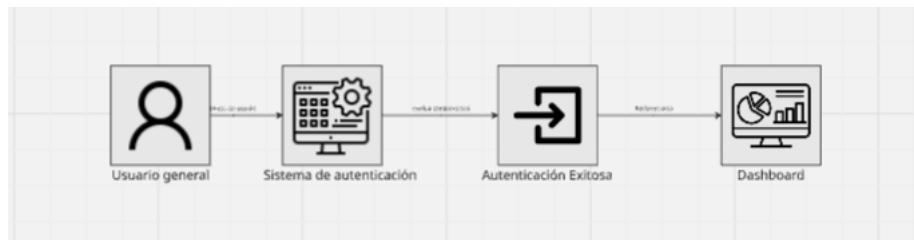
En este segmento, identificamos y fundamentamos la elección de los bounded context mediante "start-with-value", "start-with-simple" y "look-for-pivotal-events".

Técnica	Cómo se aplica	Bounded Contexts Identificados
Start-with-Value	Se parte de lo más crítico para el negocio (seguridad, continuidad del servicio, confianza del usuario).	<ul style="list-style-type: none"> IAM (login, registro, recuperación de cuenta) Emergency Management (SOS, alertas)
Start-with-Simple	Se descomponen procesos en pasos secuenciales y fáciles de aislar.	<ul style="list-style-type: none"> User Profile (datos personales, multiusuario) Treatment Support (medicación, agenda)
Look-for-Pivotal-Events	Se detectan eventos clave que cambian estados de negocio.	<ul style="list-style-type: none"> Monitoring (datos IoT que generan alertas) Health Tracking (síntomas → histórico)

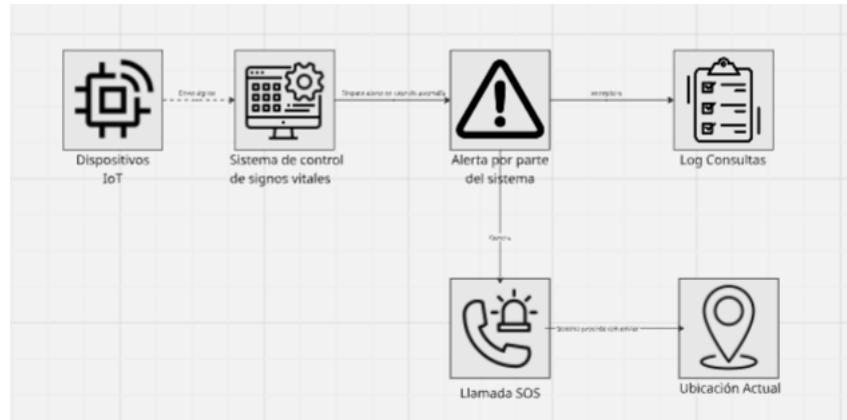
4.1.1.2. Domain Message Flows Modeling

A continuación representaremos las conexiones entre bounded context mediante Domain Storytelling para comprender como funcionan los procesos de negocios en las diferentes partes interesadas, incluyendo expertos del negocio y equipos técnicos.

Escenario 1: Autenticación de un usuario general en el aplicativo



Escenario 2: Detección en una anomalía de los signos vitales



4.1.1.3. Bounded Context Canvases

IAM:

Name:	IAM	V5 github.com/ddd-crew/bounded-context-canvas			
Purpose	autentica y registra usuarios para mayor seguridad de la aplicación móvil, así como también separar los datos por usuarios.	Strategic Classification	Supporting facilita y soporta procesos importantes, pero no es la lógica principal del negocio	Domain Roles	Gateway Context
Inbound Communication		Ubiquitous Language			Outbound Communication
Usuario de la Aplicación	registrar usuario	Usuario registrado	Login	Logout	Authenticated UserProvider
Usuario de la Aplicación	iniciar sesión	Usuario autenticado	Token		Usuario registrado
Business Decisions			un usuario debe autenticarse para acceder al sistema	Una sesión expira después de X minutos de inactividad.	No se puede registrar un usuario con un email ya registrado.
Assumptions	Asumimos que el sistema central de IAM de la organización proporcionará identidades únicas y consistentes para todos los usuarios que necesiten acceder a este bounded context				
Verification Metrics	<ul style="list-style-type: none"> Tasa de éxito de autenticación Cumplimiento de políticas de contraseñas 				
Open Questions	¿Cómo se gestionará el ciclo de vida de los usuarios (creación, modificación, baja) para este bounded context en coordinación con el sistema de IAM central?				

Profile:

Name: Profile			V5 github.com/ddd-crew/bounded-context-canvas	
Purpose Acceder a datos personales del usuario autenticado	Strategic Classification Core Facilita la obtención y actualización de datos personales de usuarios	Domain Roles Execution Context		
Inbound Communication	Ubiquitous Language	Outbound Communication		
	Business Decisions solo se pueden acceder a los datos después de una autenticación			
Assumptions Asumimos que el sistema de Profile proporcionará los datos personales para todos los usuarios que accedan a este bounded context			Verification Metrics <ul style="list-style-type: none">Tasa de éxito de autenticaciónAcceso a datos personales	Open Questions ¿Cómo se gestionará el ciclo de vida de profile? ¿Qué datos se mostrarán en este servicio?

Monitoring:

Name: Monitoring			V5 github.com/ddd-crew/bounded-context-canvas	
Purpose Monitorear, registrar y validar en tiempo real los signos vitales del paciente mediante dispositivos IoT, y mostrarlos en un panel de control accesible para usuarios autorizados.	Strategic Classification CORE Es parte central del negocio, ya que la información de signos vitales es crítica para el monitoreo de salud y la toma de decisiones médicas.	Domain Roles Gateway		
Inbound Communication	Ubiquitous Language	Outbound Communication		
	Business Decisions Validar que los datos estén dentro de parámetros humanos posibles.			
Assumptions "Asumimos que los dispositivos IoT estarán conectados y transmitirán datos en tiempo real."			Verification Metrics Porcentaje de datos correctamente validados frente a anomalías.	Open Questions ¿Cómo se manejan los datos cuando el dispositivo pierde conexión temporal?

Health Tracking:

Name: Health Tracking			V5 github.com/ddd-crew/bounded-context-canvas	
Purpose Permite el monitoreo continuo de los signos vitales de los adultos mayores, donde se registra irregularidades y emite alertas para los cuidadores y familiares.		Strategic Classification Core Gestionar los signos vitales y es el encargado de emitir las alertas por irregularidades.		Domain Roles Execution Context
Inbound Communication			Ubiquitous Language Business Decisions Se activa la alerta si los signos vitales no concuerda con el umbral establecido	
Usuario / Cuidador autenticado	Registrar alteración	Alteración reportada	Gestión de signos vitales Alertas de irregularidades	Sistema Detectar irregularidad Alerta Generada
Usuario autenticado	Registrar síntoma	Síntoma Registrado		Usuario / Cuidador autenticado Consultar historial Log consultado
Assumptions "Asumimos que el adulto mayor se encuentra registrado en el sistema junto con sus signos vitales."	Verification Metrics <ul style="list-style-type: none">Tiempo promedio en activar una alerta o notificación.Frecuencia de actualización de los signos vitales del adulto mayor.		Open Questions ¿Qué ocurre si no hay conexión al momento de detectar una irregularidad?	

Emergency Management:

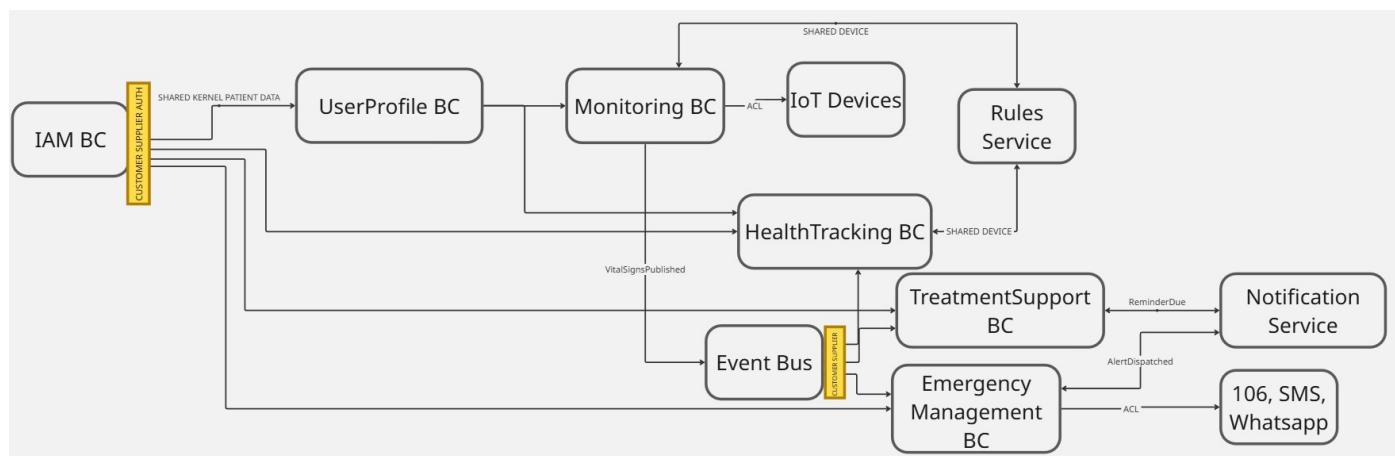
Name: Emergency Management			V5 github.com/ddd-crew/bounded-context-canvas	
Purpose Detectar, registrar y gestionar emergencias (caídas, SOS manual, signos vitales críticos), enviando alertas y ubicación en tiempo real a cuidadores y servicios de emergencia.		Strategic Classification Core Es el corazón del sistema, ya que permite salvar vidas mediante respuesta inmediata.		Domain Roles Execution Context Orchestration Context.
Inbound Communication			Ubiquitous Language Business Decisions Alertas se envían únicamente a contactos con consentimiento activo. Las llamadas SOS priorizan primero el 106, luego contactos configurados.	
Botón SOS activado por el paciente.	Evento automático de caída o alteración en signos vitales.	Confirmación / cancelación por parte del cuidador o del propio usuario.	SOS Contacto de emergencia Confirmación / Cancelación	Alertas enviadas a contactos de emergencia. Llamada telefónica automática al 106 o contactos. Ubicación GPS compartida con cuidadores y servicios de emergencia.
Assumptions "Asumimos que todos los pacientes tendrán al menos un contacto de emergencia registrado."	Verification Metrics <ul style="list-style-type: none">Tiempo de detección de una emergencia.Porcentaje de entregas exitosas de alertas.		Open Questions ¿Qué precisión mínima de GPS se aceptará para enviar ubicación?	

Treatment Support

Name: Treatment support		V5 github.com/ddd-crew/bounded-context-canvas	
Purpose Permite registrar medicamentos que son parte de un tratamiento, establecer recordatorios, confirmar el consumo de medicamentos y visualizar el panel de salud		Strategic Classification Supporting Facilita y soporta el sistema de monitoreo de salud y seguridad, pero no es el núcleo central del negocio	Domain Roles Execution context
Inbound Communication		Outbound Communication	
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px;">Usuario de la aplicación</div> <div style="background-color: #ccdefe; padding: 5px;">Registrar medicamento</div> <div style="background-color: #ffccbc; padding: 5px;">Medicamento registrado</div> </div> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px;">Usuario de la aplicación</div> <div style="background-color: #ccdefe; padding: 5px;">Ingresar a la agenda de medicación</div> <div style="background-color: #ffccbc; padding: 5px;">Agenda de medicación actualizada mostrada</div> </div> </div>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px;">Sistema</div> <div style="background-color: #ccdefe; padding: 5px;">Programar recordatorio</div> <div style="background-color: #ffccbc; padding: 5px;">Recordatorio programado</div> </div> <div style="text-align: center;"> <div style="background-color: #ffffcc; padding: 5px;">Usuario de la aplicación</div> <div style="background-color: #ccdefe; padding: 5px;">Confirmar toma de medicamento</div> <div style="background-color: #ffccbc; padding: 5px;">Medicamento consumido</div> </div> </div>		
Ubiquitous Language <div style="display: flex; justify-content: space-around;"> <div style="background-color: #ffffcc; padding: 5px;">Agenda de medicamentos</div> <div style="background-color: #ffffcc; padding: 5px;">Recordatorio</div> </div> Business Decisions <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="background-color: #ffccbc; padding: 5px; width: 20%;">Solo se puede registrar el consumo de un medicamento si este está registrado en la agenda</div> <div style="background-color: #ffccbc; padding: 5px; width: 20%;">El sistema debe registrar un recordatorio solo si el medicamento está registrado en la agenda</div> </div>			
Assumptions "Asumimos que los usuarios del sistema comprenderán el formato del formulario de medicamentos para registrarlos en la agenda."	Verification Metrics <ul style="list-style-type: none">Tiempo promedio para registrar un medicamento en la agenda.Uso del sistema para realizar un control del consumo de medicamentos.		Open Questions ¿Qué sucede si el usuario no usa el sistema para registrar el consumo de medicamentos?

4.1.2. Context Mapping

Este context mapping representa un ecosistema de salud digital centrado en el paciente, donde múltiples bounded contexts colaboran para brindar servicios de monitoreo, seguimiento y atención médica integral. El diseño muestra una arquitectura distribuida que integra dispositivos IoT, servicios de notificación y gestión de emergencias.



IAM BC

- Rol: provee identidad/autenticación/autorización.
- Patrón: Customer/Supplier → todos los demás consumen.

Relaciones:

upstream común, no guarda lógica de negocio.

UserProfile BC

- Rol: perfil de usuario, datos demográficos, médicos base.
- Patrón: Shared Kernel → comparte atributos con **Monitoring** y **HealthTracking**.

Relaciones:

- Enlaza con IAM para identidad.
- Alimenta Monitoring (edad, sexo, condiciones para thresholds).
- Alimenta HealthTracking (historial médico y datos base).

Monitoring BC

- Rol: ingestión de datos de IoT, almacenamiento temporal, publicación de eventos.
- Patrón: Publica en EventBus → Customer/Supplier. Integra con dispositivos → Anti-Corruption Layer.

Relaciones:

- Depende de UserProfile para contexto del paciente.
- Interactúa con RulesService para thresholds básicos.
- Si RulesService cae, aplica thresholds locales como fallback.

HealthTracking BC

- Rol: historial clínico, tendencias, analítica de datos a largo plazo.
- Patrón: Consume EventBus → Customer/Supplier. Comparte datos clínicos con UserProfile (Shared Kernel).

Relaciones:

- Interactúa con RulesService para análisis más fino.
- Expone datos a TreatmentSupport.

TreatmentSupport BC

- Rol: recordatorios de medicación, adherencia, soporte al tratamiento.
- Patrón: Consume EventBus → Customer/Supplier. Usa NotificationService para enviar alertas al paciente.

Relaciones:

- depende de datos de HealthTracking para tendencias de adherencia.

EmergencyManagement BC

- Rol: gestión de emergencias (contactos, ubicación, llamadas 106).
- Patrón: Consume EventBus (cuando se generan alertas).
- Se integra con NotificationService para avisos.
- Integra con sistemas externos (106, SMS, WhatsApp) vía Anti-Corruption Layer.

Shared Services

EventBus

- Rol: desacoplar productores (Monitoring) y consumidores (HT, Emergencias, Tratamiento).
- Beneficio: bajo acoplamiento, facilita nuevos consumidores.
- Riesgo: complejidad infraestructura → requiere monitoreo y escalabilidad.

RulesService

- Rol: centralizar reglas de negocio y thresholds clínicos.
- Relación: Monitoring y HealthTracking lo consultan para validar datos.
- Beneficio: consistencia, un solo lugar para reglas.
- Riesgo: SPOF si no se diseña redundante → requiere fallback en Monitoring.

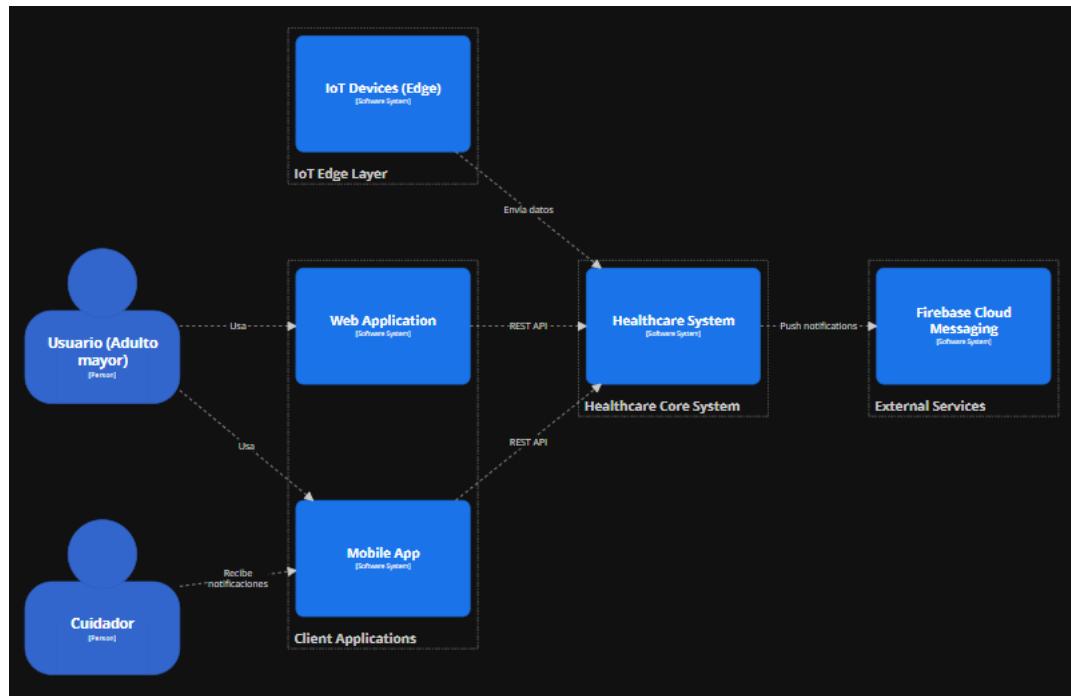
NotificationService

- Rol: unificar envío de notificaciones (push, SMS, WhatsApp).
- Beneficio: evita duplicación de lógica en Emergency y Treatment.
- Riesgo: debe soportar alta disponibilidad y multi-canal.

4.1.3. Software Architecture

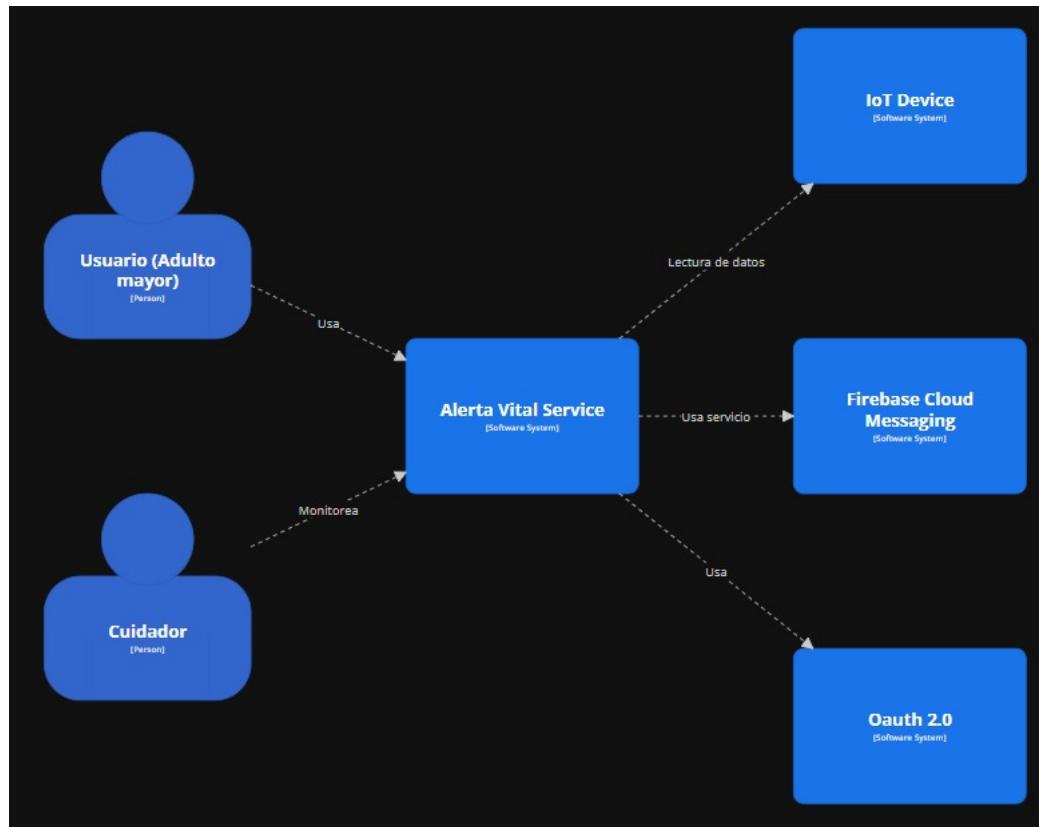
4.1.3.1. Software Architecture System Landscape Diagram

Representa el panorama global de la organización. En esta vista se identifican todas las personas y sistemas que existen en el ecosistema, independientemente de si forman parte o no del sistema principal. El System Landscape incluye a las dos personas (Usuario y Cuidador), los sistemas externos (Oauth 2.0, IoT Device y Firebase Cloud Messaging) y el sistema principal denominado Healthcare System. También muestra las relaciones de alto nivel entre todos estos elementos, permitiendo entender cómo encaja el Healthcare System dentro del entorno tecnológico total.



4.1.3.2. Software Architecture Context Level Diagrams

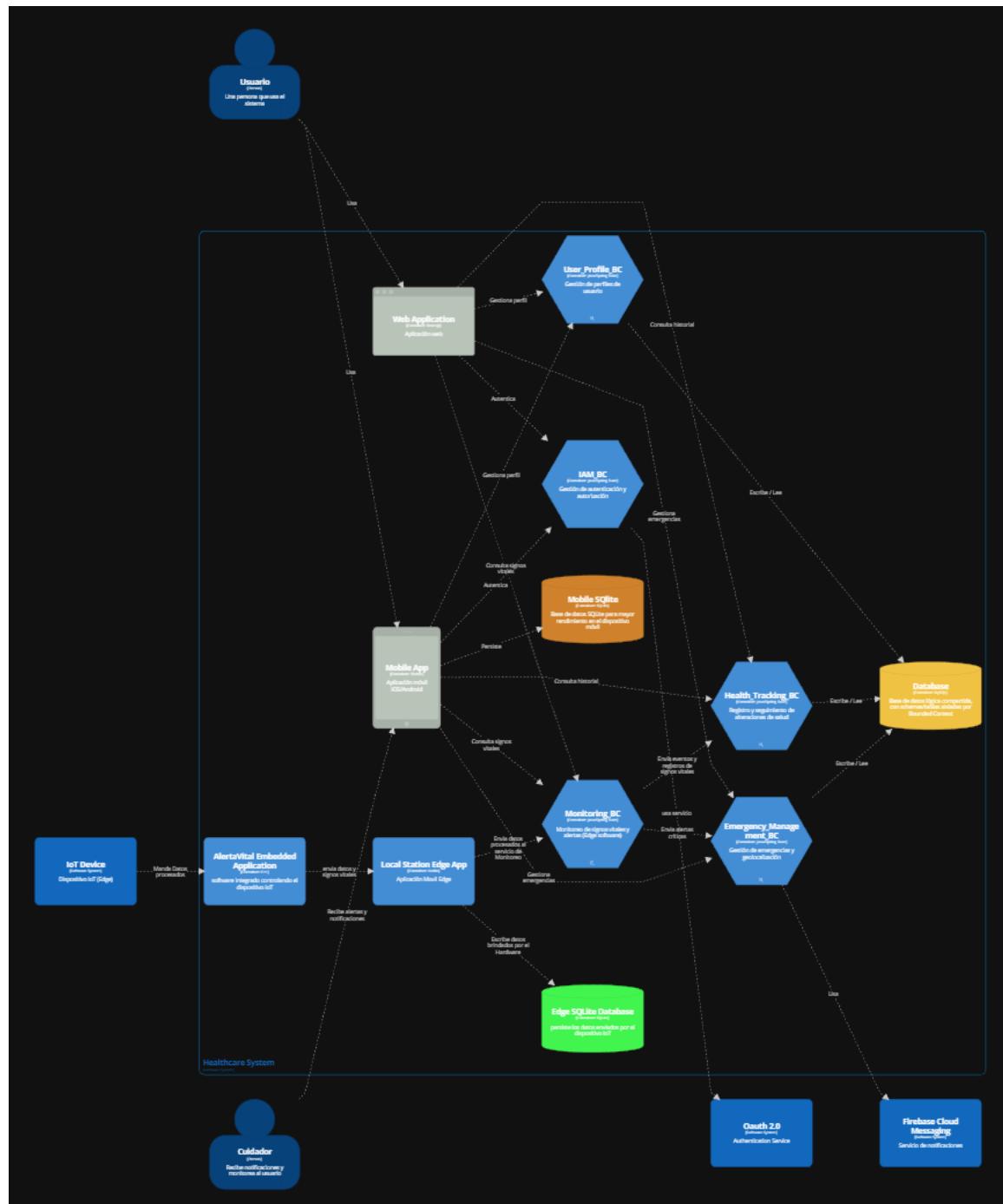
Este nivel se centra exclusivamente en el Healthcare System y su interacción con el exterior. Aquí se muestran las personas que utilizan el sistema (Usuario y Cuidador), los sistemas externos con los que se integra (Oauth 2.0, IoT Device y Firebase Cloud Messaging) y las relaciones entre ellos. No se entra en detalles internos del sistema; sólo se muestra cómo se vincula con su entorno. Esta vista define claramente los límites del Healthcare System y establece el alcance general del modelo.



4.1.3.3. Software Architecture Container Level Diagrams

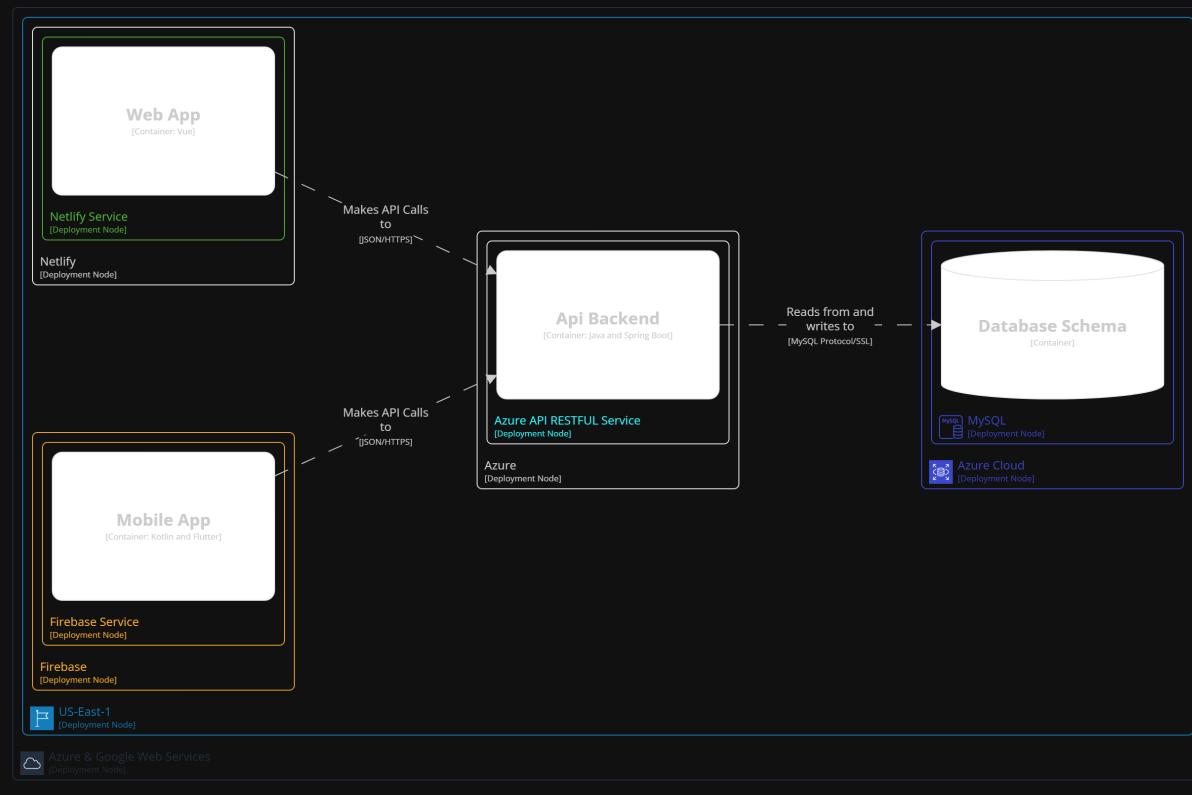
Este nivel descompone el Healthcare System en sus principales contenidos, es decir, en los bloques tecnológicos que lo conforman: aplicaciones, servicios, bases de datos y ejecutables. En tu caso, se identifican la aplicación web, la aplicación móvil, la base de datos principal, los bounded contexts (IAM_BC, User_Profile_BC, Monitoring_BC, Health_Tracking_BC y Emergency_Management_BC), el software edge, el dispositivo embebido y las bases de datos SQLite locales de los dispositivos móviles. También se muestran las relaciones entre estos contenidos, lo cual permite comprender cómo se estructura el sistema internamente y cómo cada parte colabora para cumplir las funcionalidades globales.

Health System Container



4.1.3.4. Software Architecture Deployment Diagrams

Diagrama de despliegue, en donde se puede definir que para la aplicación web usaremos una web llamada Netlify, para las aplicaciones móviles se distribuirán las apks firmadas en Firebase Distribution y el servicio Web junto a su base de datos estarán desplegadas en servicios de Azure.



[Deployment] X - Live
Saturday, September 20, 2025 at 6:22 PM Peru Standard Time

4.2. Tactical-Level Domain-Driven Design

4.2.1. Bounded Context: IAM

4.2.1.1. Domain Layer

Sub-capa Model:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Aggregate	User	clase para definir el Usuario de la aplicación	Ser el punto de entrada para modificar y mantener la integridad del usuario como entidad del dominio de identidad	Relacionado con los demás boundedContext, ya que encapsula toda la lógica de negocio.
Command	SignInCommand	comando para el inicio de sesión	representar la intención de iniciar sesión	Usado en la implementación del servicio de autenticación.
Command	SignUpCommand	comando para registro	representa la intención de registrarse a la aplicación	Usado en la implementación del servicio de autenticación
Value Object	EmailAddress	dirección de correo electrónico del usuario	representar la dirección de correo del usuario	usado en "User"
Value Object	PersonName	nombre y apellido de usuario	representa los nombres y apellidos de un usuario	usado en "User"

Sub-capa Services:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Interface	AuthCommandService	servicio para métodos de autenticación	estipular una estructura clara a seguir	Uso en la capa "application" para implementar los métodos dados
Interface	JWTService	servicio para métodos de JWT	estipular una estructura clara a seguir	uso en la capa "Infrastructure" para la implementación de los métodos.

4.2.1.2. Interface Layer

Sub-capa REST:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Resource	AuthRequestResource	estructura de una petición para autenticar un usuario	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente.	Uso en el "AuthController" para peticionar datos de una manera predeterminada en la autenticación
Resource	AuthResponseResource	estructura de una respuesta al autenticar un usuario	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente.	Uso en el "AuthController" para emitir datos de una manera predeterminada en la autenticación
Resource	RegisterRequestResource	estructura de una petición para registrar un usuario	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente.	Uso en el "AuthController" para peticionar datos de una manera predeterminada en el registro
Resource	RegisterResponseResource	estructura de una respuesta al registrar un usuario	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente.	Uso en el "AuthController" para emitir datos de una manera predeterminada en el registro
Assembler	SignInCommandFromResourceAssembler	Convierte un recurso a un comando	Evitar la corrupción entre la comunicación de datos	Usado en el "AuthController" para convertir el recurso pedido en autenticación a un comando
Assembler	SignUpCommandFromResourceAssembler	Convierte un recurso a un comando	Evitar la corrupcion entre comunicación de datos	Usoso en el "AuthController" para convertir el recurso pedido en autenticación a un comando
Controller	AuthController	Parte de la aplicación que recibe las solicitudes del cliente	recibe solicitudes, coordina la ejecución y devuelve respuestas.	-

4.2.1.3. Application Layer

Sub-capa Internal:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
CommandHandlers	AuthCommandServiceImpl	Implementación de los Comandos de Autenticación	Implementar los métodos para el servicio de autenticación	Implementa los métodos de la interface de su mismo nombre en la capa de "Services".

4.2.1.4. Infrastructure Layer

Sub-capa Repository:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	UserRepository	repositorio a usar del modelo "User"	acceder y manipular datos persistidos	Usado en la Capa "Application" para implementar el registro y autenticación de un usuario.

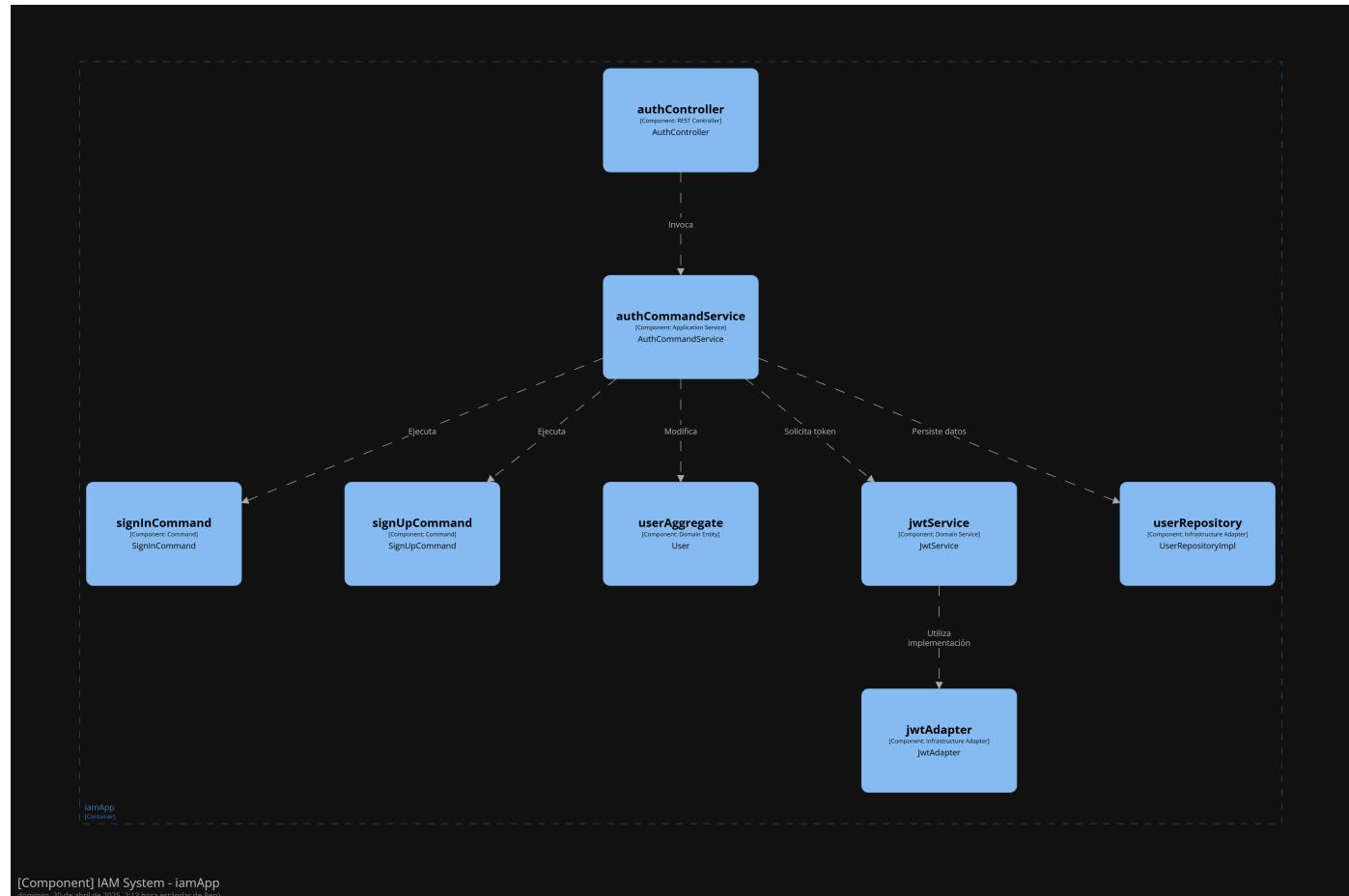
Sub-capa Security:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Config	SecurityConfig	configuración de autorizaciones y permisos	configurar reglas de seguridad para la aplicación	Relacionado a la aplicación

Sub-capa JWT:

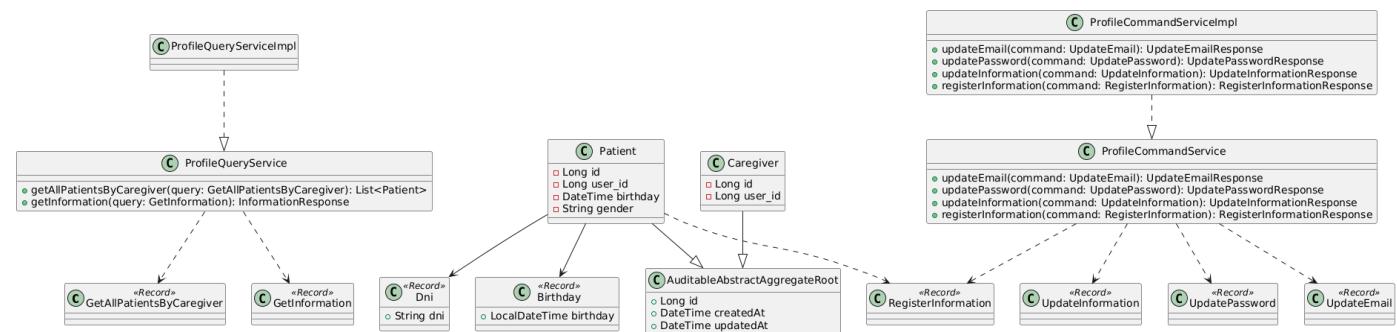
Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Class	JwtAuthFilter	Un filtro que intercepta cada petición HTTP para verificar si contiene un token JWT válido antes de dejar que llegue al controlador.	Validar el JWT en cada solicitud y establecer la autenticación en el contexto de seguridad.	Relacionado con la seguridad de la aplicación
Class	JwtServiceImpl	Una clase de servicio que gestiona la creación, validación y decodificación de tokens JWT.	Encapsular toda la lógica relacionada con el manejo de tokens JWT.	Relacionado con la seguridad de la aplicación

4.2.1.5. Bounded Context Software Architecture Component Level Diagrams

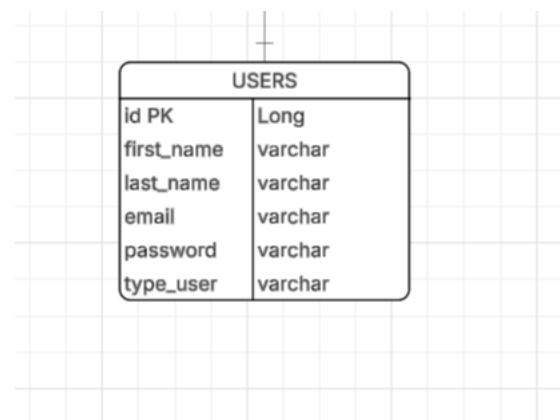


4.2.1.6. Bounded Context Software Architecture Code Level Diagrams

4.2.1.6.1. Bounded Context Domain Layer Class Diagrams



4.2.1.6.2. Bounded Context Database Design Diagram



USERS

Propósito: Manejo de autenticación y datos básicos del usuario

Campo	Tipo	Descripción
id	Long (PK)	Identificador único del usuario
first_name	varchar	Nombre del usuario
last_name	varchar	Apellido del usuario
email	varchar	Correo electrónico (único)
password	varchar	Contraseña hasheada
type_user	varchar	Tipo de usuario (admin, user, etc.)

Relaciones:

- 1:1 con PROFILES

4.2.2. Bounded Context: Profile**4.2.2.1. Domain Layer****Sub-capa Model:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Entity	Patient	Persona vulnerable a cuidar	representa los datos de la entidad "Paciente"	Relacionado con "User", este determina el tipo de usuario
Entity	Caregiver	Persona que cuida de la persona vulnerable	Representar los datos de la entidad "Caregiver"	Relacionado con "User", este determina el tipo de usuario
Value Object	Birthday	Value object usado en el Aggregate de Patient	Representar la fecha de nacimiento de un paciente.	Relacionado con Patient
Value Object	DNI	Value object usado en el Aggregate de Patient y Cuidador	Representar el documento de identidad del paciente o cuidador	Relacionado con Patient o Caregiver
Record (Query)	GetAllPatientsByCaregiver	Consulta para obtener las personas que esta cuidando el tipo de usuario "Cuidador"	Encapsula los parámetros para buscar a los pacientes de cada cuidador	Usado por servicio de Cuidador
Record (Query)	GetInformation	Consulta para obtener los datos personales del usuario general	Encapsula los parámetros para buscar los datos personales	Usado por servicio de ambos tipos de usuario
Record (Command)	RegisterInformation	Comando para guardar los datos personales	Encapsular los datos necesarios para guardar los datos personales	Usado por servicios de ambos tipos de usuario
Record (Command)	UpdateInformation	Comando para actualizar los datos personales	Encapsular los datos necesarios para actualizar los datos personales	Usado por servicios de ambos tipos de usuario
Record (Command)	UpdatePassword	Comando para actualizar la contraseña	Encapsular los datos necesarios para actualizar la contraseña	Usado por servicio de ambos tipos de usuario y Bounded Context de IAM
Record (Command)	UpdateEmail	Comando para actualizar el correo electrónico	Encapsular los datos necesarios para actualizar el correo electrónico	Usado por servicio de ambos tipos de usuario y Bounded Context de IAM

Sub-capa Services:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Interface	ProfileCommandService	Servicio para comandos relacionados con los usuarios	Declarar métodos para crear o actualizar datos personales de los usuarios	Implementado por la clase ProfileCommandServiceImpl . Usado en capa Application
Interface	ProfileQueryService	Servicio para comandos relacionados con los usuarios	Obtener datos persistidos de los usuarios	Implementado por la clase ProfileQueryServiceImpl . Usado en capa Application

4.2.2.2. Interface Layer**Sub-capa REST:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Controller	ProfileController	clase encargada de la gestión de los endpoints	enviar respuestas o recibir peticiones	Relacionado con las demás capas.

4.2.2.3. Application Layer

Sub-capa Internal:

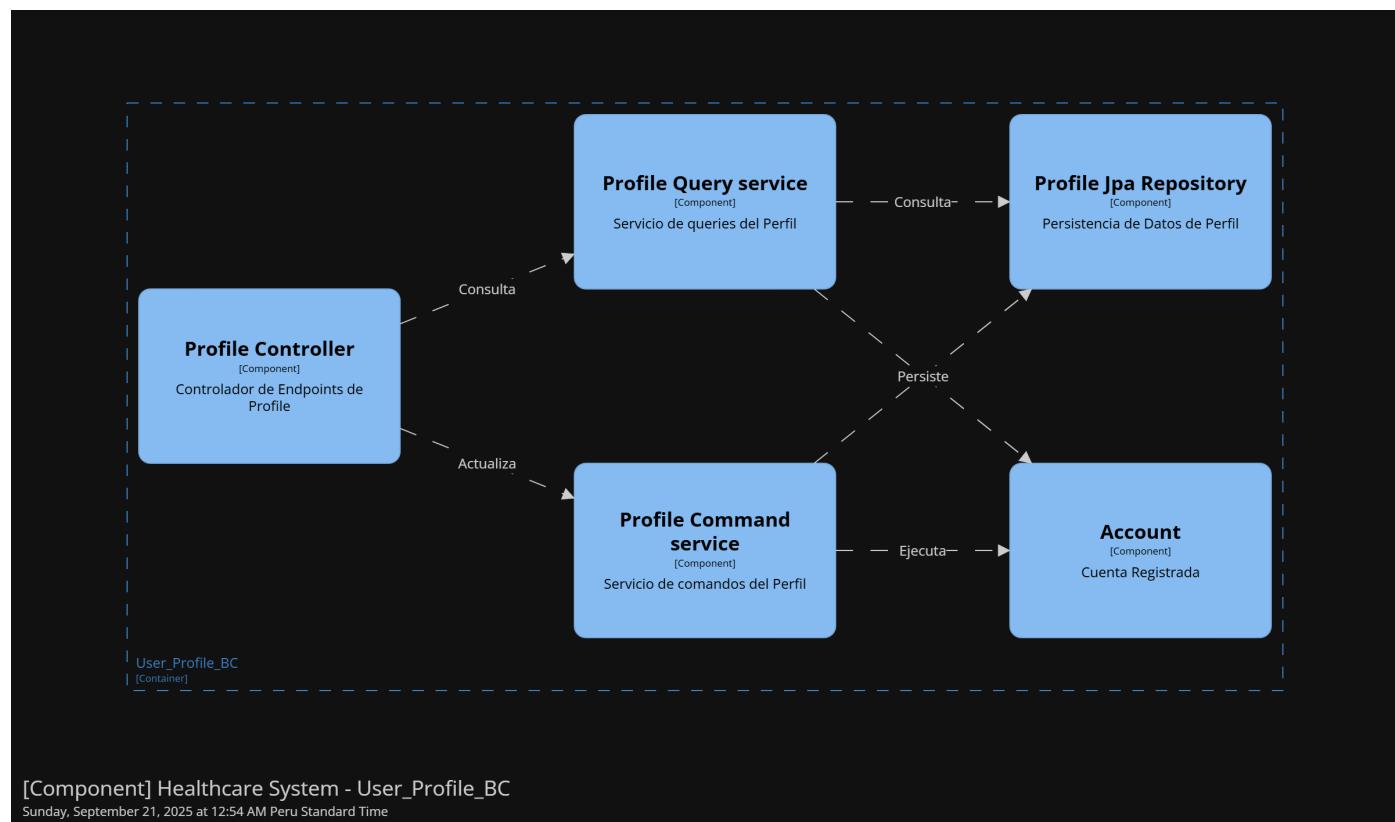
Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Class (implementation service)	ProfileCommandServiceImpl	Implementación de los Comandos del Perfil	Implementar los métodos para el servicio de Perfil	Implementa los métodos de la interface de su mismo nombre en la capa de "Services".
Class (implementation service)	ProfileQueryServiceImpl	Implementación de las Consultas del Perfil	Implementar los métodos para el servicio de Perfil	Implementa los métodos de la interface de su mismo nombre en la capa de "Services".

4.2.2.4. Infrastructure Layer

Sub-capa Repository:

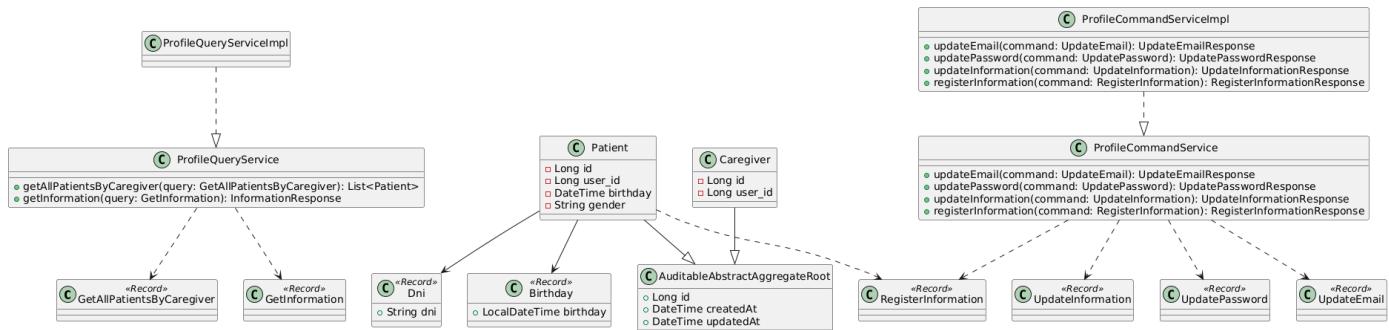
Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	PatientRepository	repositorio a usar del modelo "Patient"	acceder y manipular datos persistidos	Usado en la Capa "Application" para implementar el registro de un usuario "Paciente".
Repository	CareGiverRepository	repositorio a usar del modelo "Caregiver"	acceder y manipular datos persistidos	Usado en la Capa "Application" para implementar el registro de un usuario "Cuidador".

4.2.2.5. Bounded Context Software Architecture Component Level Diagrams



4.2.2.6. Bounded Context Software Architecture Code Level Diagrams

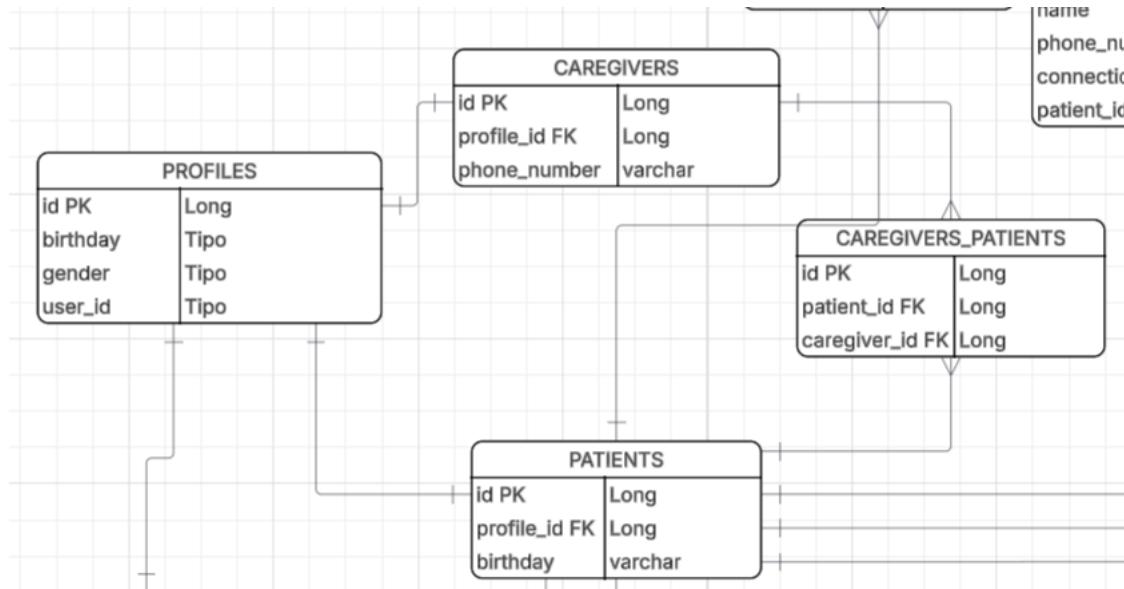
4.2.2.6.1. Bounded Context Domain Layer Class Diagrams



4.2.2.6.2. Bounded Context Database Design Diagram

los tipos de usuarios (Pacientes y cuidadores), se modelan de esta manera para una mejor escalabilidad.

Cada paciente puede tener muchos cuidadores y un cuidador puede tener muchos pacientes.



PROFILES

Propósito: Información demográfica común entre pacientes y cuidadores

Campo	Tipo	Descripción
id	Long (PK)	Identificador único del perfil
birthday	Tipo	Fecha de nacimiento
gender	Tipo	Género del usuario
user_id	Tipo (FK)	Referencia a USERS

Relaciones:

- N:1 con USERS
- 1:1 con PATIENTS
- 1:1 con CAREGIVERS

PATIENTS

Propósito: Información específica de pacientes

Campo	Tipo	Descripción
id	Long (PK)	Identificador único del paciente
profile_id	Long (FK)	Referencia a PROFILES
birthday	varchar	Fecha de nacimiento específica

Relaciones:

- N:1 con PROFILES
- N:M con CAREGIVERS (a través de CAREGIVERS_PATIENTS)
- 1:N con TREATMENTS
- 1:N con SYMPTOMS
- 1:N con DISTURBANCES
- 1:N con VITAL_SIGNS_ALERT
- 1:N con EMERGENCIES
- 1:N con CONTACTS

CAREGIVERS

Propósito: Información específica de cuidadores

Campo	Tipo	Descripción
id	Long (PK)	Identificador único del cuidador
profile_id	Long (FK)	Referencia a PROFILES
phone_number	varchar	Número de teléfono del cuidador

Relaciones:

- N:1 con PROFILES
- N:M con PATIENTS (a través de CAREGIVERS_PATIENTS)

CAREGIVERS_PATIENTS

Propósito: Tabla intermedia para la relación N:M entre cuidadores y pacientes

Campo	Tipo	Descripción
id	Long (PK)	Identificador único
patient_id	Long (FK)	Referencia a PATIENTS
caregiver_id	Long (FK)	Referencia a CAREGIVERS

4.2.3. Bounded Context: <Health Tracking>

4.2.3.1. Domain Layer

Sub-capa Model

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Aggregate	Patient	Representa al paciente dentro del sistema, raíz de la información clínica.	Centralizar datos clínicos y servir de punto de acceso al historial.	Relacionado con Vital, Alert, Alteration y Symptom.
Value Object	Symptoms	Registro de una manifestación puntual.	Documentar síntomas subjetivos para el historial clínico.	Relacionado con el paciente.
Value Object	Alteration	Cambio detectado en los signos vitales que indica una condición irregular.	Representa valores medidos y compararlos con rangos normales.	Relacionado con paciente.
Value Object	Vital	Lectura puntual por parte del sensor IoT.	Estandarizar los datos del sensor.	Relacionado con .
Aggregate	Alert	Notificación generada por cambios críticos en el estado del paciente.	Advertir de riesgos.	Relacionado con paciente.
Command	PostPatient	Registra un nuevo paciente.	Crear un paciente con los datos necesarios.	Usa User.
Command	PostSymptom	Añade un nuevo síntoma reportado por el paciente.	Crear instancia de Síntoma asociada a Paciente.	Usa Patient y Symptom.
Command	PostAlert	Crea una alerta manual o automática.	Persistir una Alerta y notificar al sistema.	Usa Patient y Alert.
Command	PostAlteration	Registra una alteración detectada en los signos vitales.	Persistir una Alteracion asociada al Paciente.	Usa Patient y Alteration.
Query	GetPatientHistory	Devuelve el historial completo del paciente.	Reunir datos clínicos del Paciente.	Consulta Patient y todas sus entidades relacionadas.
Query	GetPendantAlert	Recupera las alertas que no han sido atendidas.	Soportar monitoreo activo del paciente.	Consulta Alert.
Query	GetByPatientSymptoms	Recupera los síntomas registrados por el paciente.	Complementar información clínica.	Consulta Symptom.

Sub-capa Services

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Interface	PatientCommandService	Servicio para comandos relacionados con pacientes	Declarar métodos para crear pacientes	Implementado por la clase PatientCommandServiceImpl . Usado en capa Application
Interface	AlertCommandService	Servicio para comandos relacionados con alertas	Declarar métodos para crear alertas	Implementado por la clase AlertCommandServiceImpl . Usado en capa Application
Interface	PatientQueryService	Servicio para consultas relacionadas con pacientes	Declarar métodos para obtener datos de pacientes	Implementado por la clase PatientQueryServiceImpl . Usado en capa Application
Interface	AlertQueryService	Servicio para consultas relacionadas con alertas	Declarar métodos para obtener datos de alertas	Implementado por la clase AlertQueryServiceImpl . Usado en capa Application

4.2.3.2. Interface Layer**Sub-capa REST:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Controller	PatientController	Controlador REST para gestionar pacientes	Recibe solicitudes del cliente relacionadas con pacientes, coordina comandos y devuelve respuestas	Utiliza PatientRequestResource, PatientResponseResource y los assemblers correspondientes
Controller	AlertController	Controlador REST para gestionar citas	Maneja solicitudes relacionadas con la creación de alertas	Utiliza AlertRequestResource, AlertResponseResource y sus respectivos assemblers
Resource	PatientRequestResource	Estructura de una petición para crear o actualizar un paciente	Representa datos de entrada del cliente sobre pacientes	Usado por PatientController para enviar datos al sistema
Resource	PatientResponseResource	Estructura de la respuesta con información de un paciente	Devuelve al cliente una representación clara de un paciente	Usado por PatientController como respuesta
Resource	AlertRequestResource	Estructura de una petición para crear una alerta	Representa datos de entrada del cliente sobre sus signos vitales	Usado por AlertController para procesar citas
Resource	AppointmentResponseResource	Estructura de una respuesta con datos de una cita	Devuelve al cliente una representación clara de una cita	Usado por AlertController
Assembler	CreatePatientCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación	Traducir la entrada del cliente a un comando de dominio	Usado por PatientController
Assembler	UpdatePatientCommandFromResourceAssembler	Convierte un recurso de petición en un comando de actualización	Traducir la entrada del cliente a un comando de dominio	Usado por PatientController
Assembler	PatientResourceFromEntityAssembler	Convierte una entidad Patient en un recurso de respuesta	Traducir objetos de dominio a estructuras legibles para el cliente	Usado por PatientController
Assembler	CreateAlertCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación	Traducir la entrada del cliente a un comando de dominio	Usado por AlertController
Assembler	AlertResourceFromEntityAssembler	Convierte una entidad Alert en un recurso de respuesta	Traduce objetos del dominio a respuestas claras para el cliente	Usado por AlertController

4.2.3.3. Application Layer**Sub-capa Internal:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
------	--------	-------------	---------------------------	------------------------------

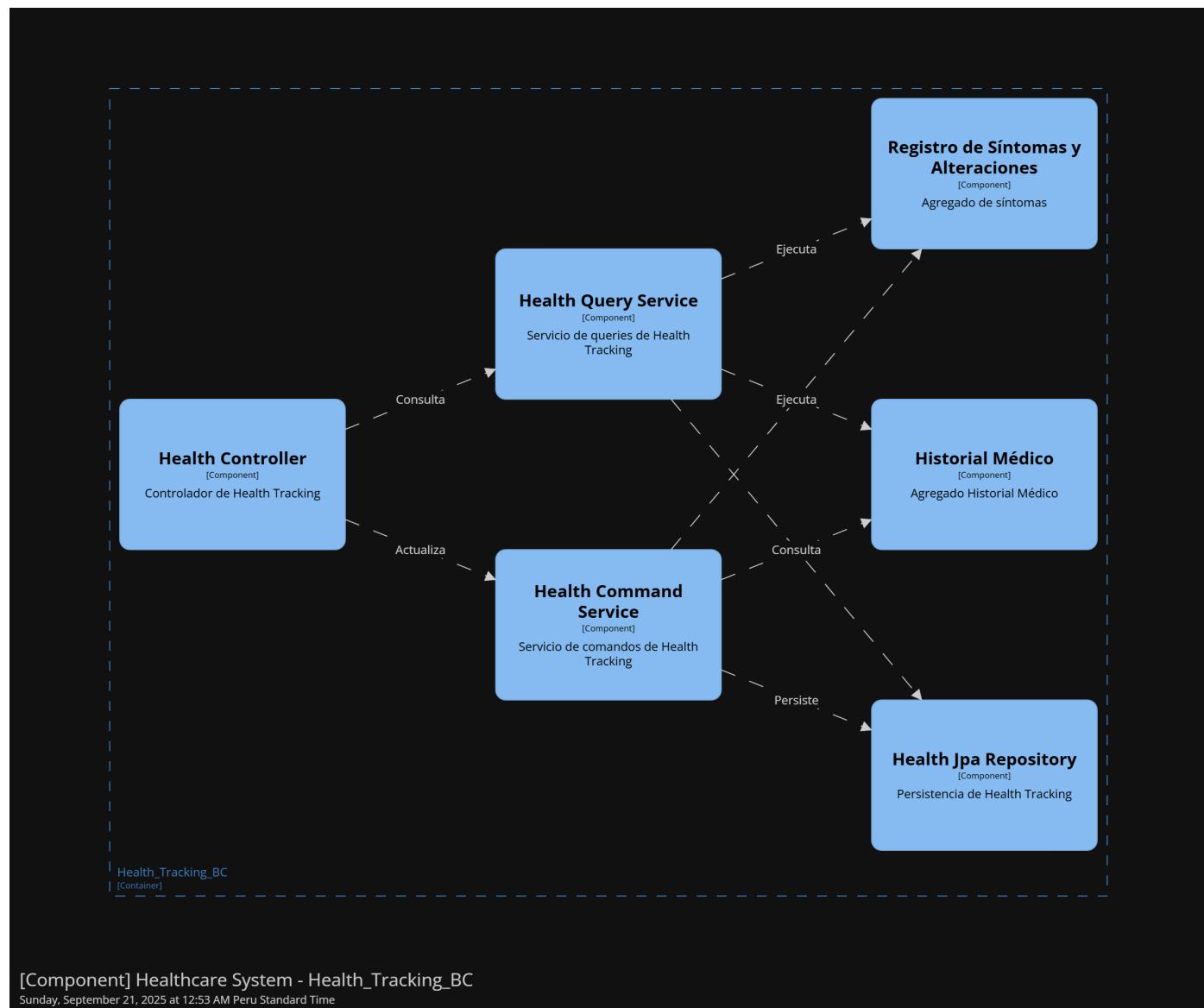
Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Service	PatientCommandServiceImpl	Implementación del servicio de comandos para pacientes	Ejecutar la lógica de creación, actualización y eliminación de pacientes	Implementa PatientCommandService. Utiliza entidades y repositorios de dominio
Service	AlertCommandServiceImpl	Implementación del servicio de comandos para alertas	Ejecutar la lógica de creación de alertas	Implementa AlertCommandService. Interactúa con Alert y sus repositorios
Service	PatientQueryServiceImpl	Implementación del servicio de consultas para pacientes	Obtener información de pacientes a partir de distintas consultas	Implementa PatientQueryService. Consulta entidades Patient
Service	AlertQueryServiceImpl	Implementación del servicio de consultas para alertas	Obtener alertas por ID	Implementa AlertQueryService. Usa Alert y consultas específicas

4.2.3.4. Infrastructure Layer

Sub-capa Infrastructure:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	PatientRepository	Repository para gestionar pacientes	Encargado de la persistencia y recuperación de los datos de los pacientes	Relacionado con la entidad Patient, interactúa con la base de datos para almacenar y consultar pacientes
Repository	AlertRepository	Repository para gestionar alertas	Encargado de la persistencia y recuperación de las alertas	Relacionado con la entidad Alert, interactúa con la base de datos para almacenar y consultar alertas

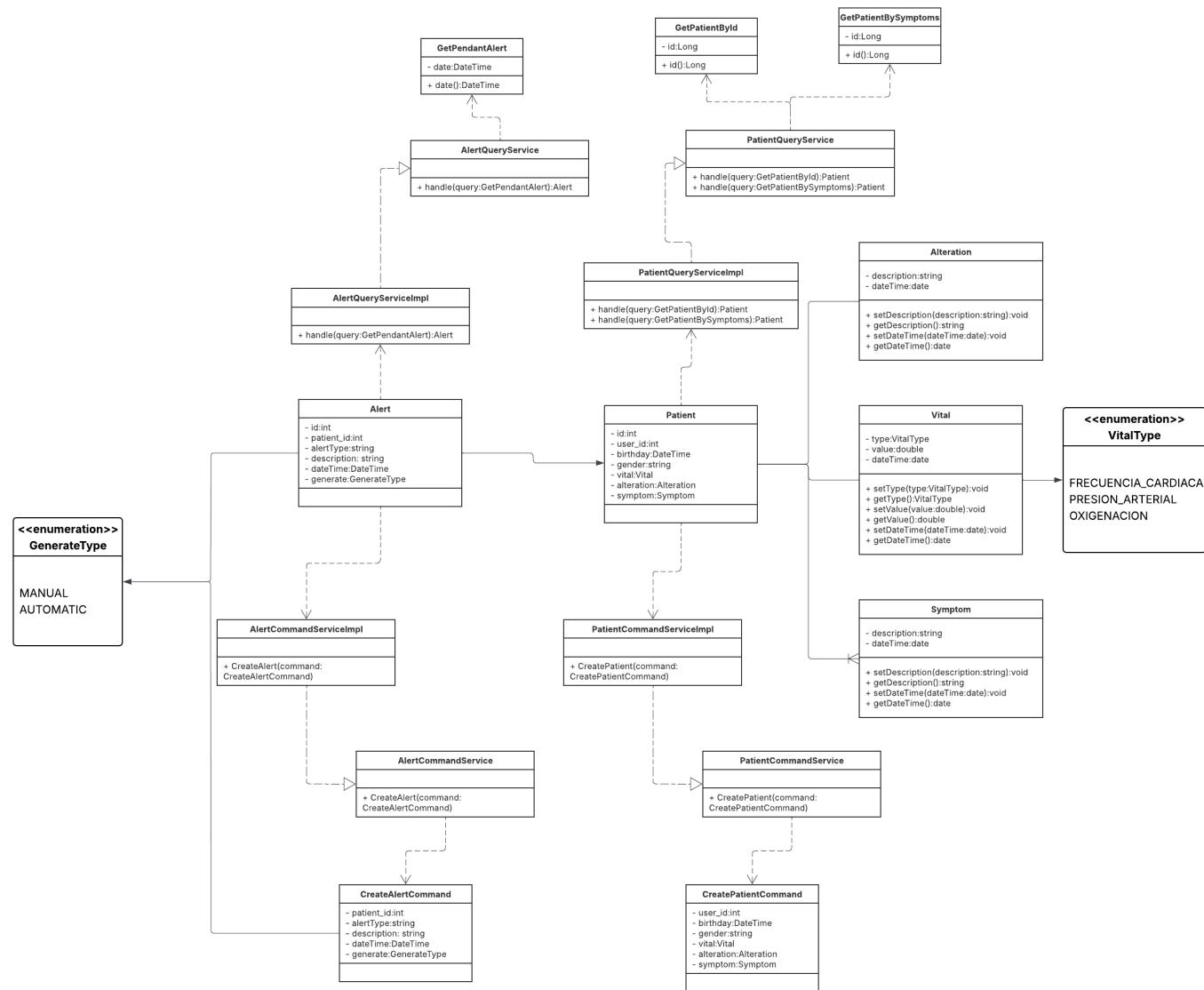
4.2.3.5. Bounded Context Software Architecture Component Level Diagrams



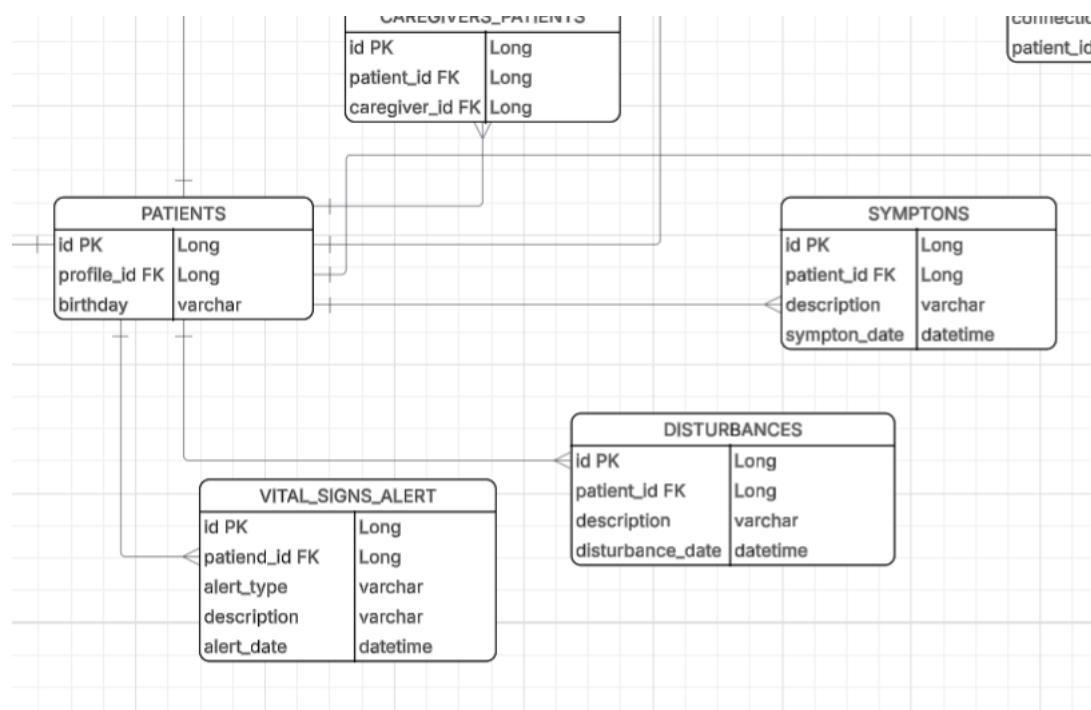
4.2.3.6. Bounded Context Software Architecture Code Level Diagrams

4.2.3.6.1. Bounded Context Domain Layer Class Diagrams

Este diagrama UML representa la arquitectura de un sistema de gestión médica centrada en el paciente y las alertas. La estructura está basada en principios de diseño orientado a objetos y se organiza en capas de comandos y consultas, siguiendo un enfoque CQRS (Command Query Responsibility Segregation). A lo largo del análisis, se describirán los principales componentes del sistema, como los servicios de comandos y consultas, las entidades centrales como Patient y Alert, así como los objetos de valor (value objects) y comandos asociados. También se explicará cómo estas clases interactúan entre sí mediante relaciones de herencia, interfaces, asociación y dependencia para mantener una lógica de negocio clara y mantenible. Se estará usando Lombok para la creación de Getter y Setter para evitar en redundancia de código.



4.2.3.6.2. Bounded Context Database Design Diagram



Sub-capa Model

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Aggregate	Patient	Representa al paciente dentro del sistema, raíz de la información clínica.	Centralizar datos clínicos y servir de punto de acceso al historial.	Relacionado con Vital, Alert, Alteration y Symptom.
Value Object	Symptoms	Registro de una manifestación puntual.	Documentar síntomas subjetivos para el historial clínico.	Relacionado con el paciente.
Value Object	Alteration	Cambio detectado en los signos vitales que indica una condición irregular.	Representa valores medidos y compararlos con rangos normales.	Relacionado con paciente.
Value Object	Vital	Lectura puntual por parte del sensor IoT.	Estandarizar los datos del sensor.	Relacionado con .
Aggregate	Alert	Notificación generada por cambios críticos en el estado del paciente.	Advertir de riesgos.	Relacionado con paciente.
Command	PostPatient	Registra un nuevo paciente.	Crear un paciente con los datos necesarios.	Usa User.
Command	PostSymptom	Añade un nuevo síntoma reportado por el paciente.	Crear instancia de Síntoma asociada a Paciente.	Usa Patient y Symptom.
Command	PostAlert	Crea una alerta manual o automática.	Persistir una Alerta y notificar al sistema.	Usa Patient y Alert.
Command	PostAlteration	Registra una alteración detectada en los signos vitales.	Persistir una Alteracion asociada al Paciente.	Usa Patient y Alteration.
Query	GetPatientHistory	Devuelve el historial completo del paciente.	Reunir datos clínicos del Paciente.	Consulta Patient y todas sus entidades relacionadas.
Query	GetPendantAlert	Recupera las alertas que no han sido atendidas.	Soportar monitoreo activo del paciente.	Consulta Alert.
Query	GetByPatientSymptoms	Recupera los síntomas registrados por el paciente.	Complementar información clínica.	Consulta Symptom.

SYMPTOMS

Propósito: Registro de síntomas reportados por pacientes

Campo	Tipo	Descripción
id	Long (PK)	Identificador único del síntoma
patient_id	Long (FK)	Referencia a PATIENTS
description	varchar	Descripción del síntoma
symptom_date	datetime	Fecha de registro del síntoma

Relaciones:

- N:1 con PATIENTS

DISTURBANCES

Propósito: Registro de alteraciones o eventos anómalos

Campo	Tipo	Descripción
id	Long (PK)	Identificador único
patient_id	Long (FK)	Referencia a PATIENTS
description	varchar	Descripción de la alteración
disturbance_date	datetime	Fecha de la alteración

Relaciones:

- N:1 con PATIENTS

VITAL_SIGNS_ALERT

Propósito: Alertas relacionadas con signos vitales

Campo	Tipo	Descripción
id	Long (PK)	Identificador único
patient_id	Long (FK)	Referencia a PATIENTS
alert_type	varchar	Tipo de alerta
description	varchar	Descripción de la alerta
alert_date	datetime	Fecha de la alerta

Relaciones:

- N:1 con PATIENTS

4.2.3.2. Interface Layer**Sub-capa REST:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Controller	PatientController	Controlador REST para gestionar pacientes	Recibe solicitudes del cliente relacionadas con pacientes, coordina comandos y devuelve respuestas	Utiliza PatientRequestResource, PatientResponseResource y los assemblers correspondientes
Controller	AlertController	Controlador REST para gestionar citas	Maneja solicitudes relacionadas con la creación de alertas	Utiliza AlertRequestResource, AlertResponseResource y sus respectivos assemblers
Resource	PatientRequestResource	Estructura de una petición para crear o actualizar un paciente	Representa datos de entrada del cliente sobre pacientes	Usado por PatientController para enviar datos al sistema
Resource	PatientResponseResource	Estructura de la respuesta con información de un paciente	Devuelve al cliente una representación clara de un paciente	Usado por PatientController como respuesta
Resource	AlertRequestResource	Estructura de una petición para crear una alerta	Representa datos de entrada del cliente sobre sus signos vitales	Usado por AlertController para procesar citas
Resource	AppointmentResponseResource	Estructura de una respuesta con datos de una cita	Devuelve al cliente una representación clara de una cita	Usado por AlertController
Assembler	CreatePatientCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación	Traducir la entrada del cliente a un comando de dominio	Usado por PatientController

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Assembler	UpdatePatientCommandFromResourceAssembler	Convierte un recurso de petición en un comando de actualización	Traducir la entrada del cliente a un comando de dominio	Usado por PatientController
Assembler	PatientResourceFromEntityAssembler	Convierte una entidad Patient en un recurso de respuesta	Traducir objetos de dominio a estructuras legibles para el cliente	Usado por PatientController
Assembler	CreateAlertCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación	Traducir la entrada del cliente a un comando de dominio	Usado por AlertController
Assembler	AlertResourceFromEntityAssembler	Convierte una entidad Alert en un recurso de respuesta	Traduce objetos del dominio a respuestas claras para el cliente	Usado por AlertController

4.2.3.3. Application Layer

Sub-capa Internal:

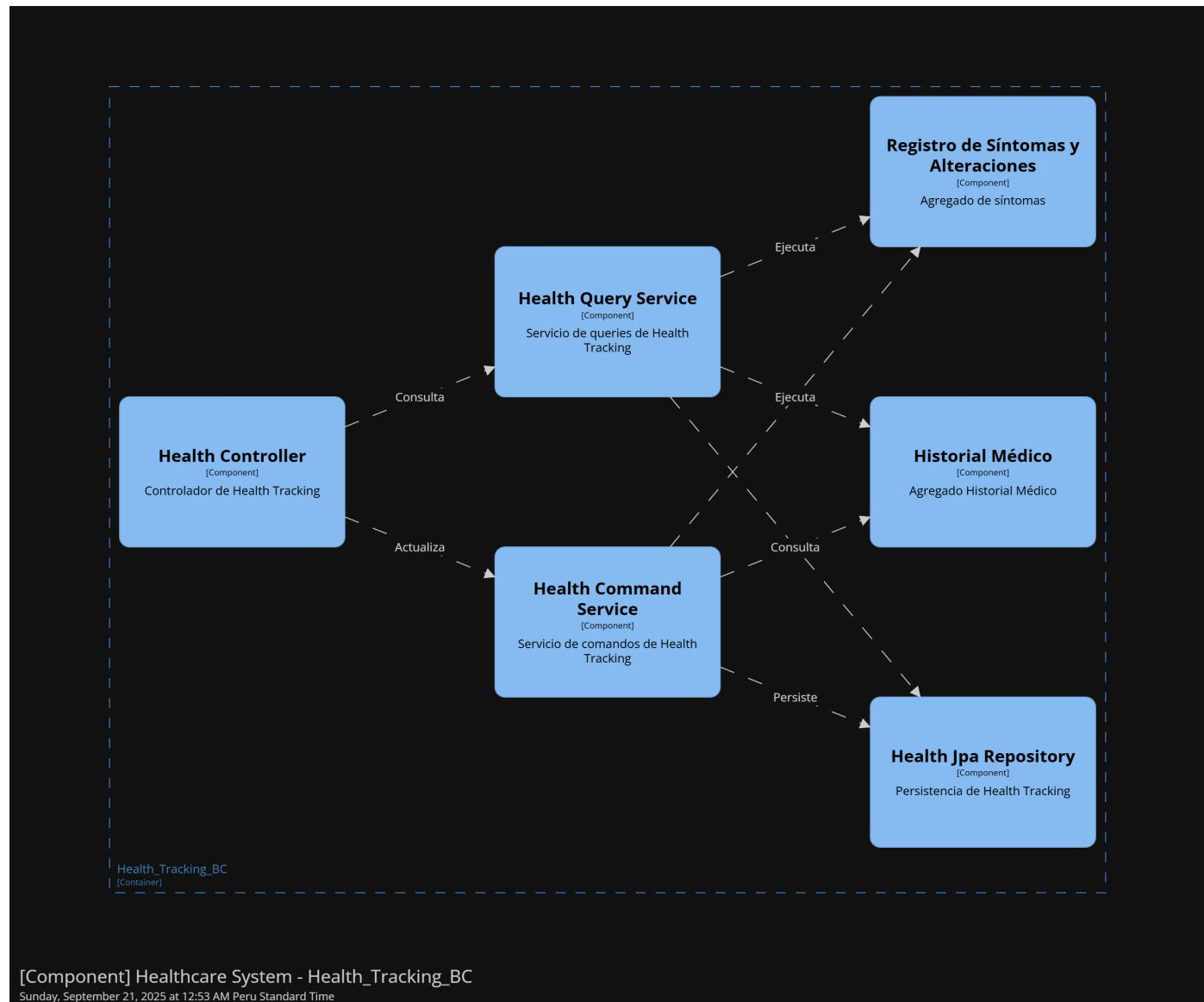
Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Service	PatientCommandServiceImpl	Implementación del servicio de comandos para pacientes	Ejecutar la lógica de creación, actualización y eliminación de pacientes	Implementa PatientCommandService. Utiliza entidades y repositorios de dominio
Service	AlertCommandServiceImpl	Implementación del servicio de comandos para alertas	Ejecutar la lógica de creación de alertas	Implementa AlertCommandService. Interactúa con Alert y sus repositorios
Service	PatientQueryServiceImpl	Implementación del servicio de consultas para pacientes	Obtener información de pacientes a partir de distintas consultas	Implementa PatientQueryService. Consulta entidades Patient
Service	AlertQueryServiceImpl	Implementación del servicio de consultas para alertas	Obtener alertas por ID	Implementa AlertQueryService. Usa Alert y consultas específicas

4.2.3.4. Infrastructure Layer

Sub-capa Infrastructure:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	PatientRepository	Repository para gestionar pacientes	Encargado de la persistencia y recuperación de los datos de los pacientes	Relacionado con la entidad Patient, interactúa con la base de datos para almacenar y consultar pacientes
Repository	AlertRepository	Repository para gestionar alertas	Encargado de la persistencia y recuperación de las alertas	Relacionado con la entidad Alert, interactúa con la base de datos para almacenar y consultar alertas

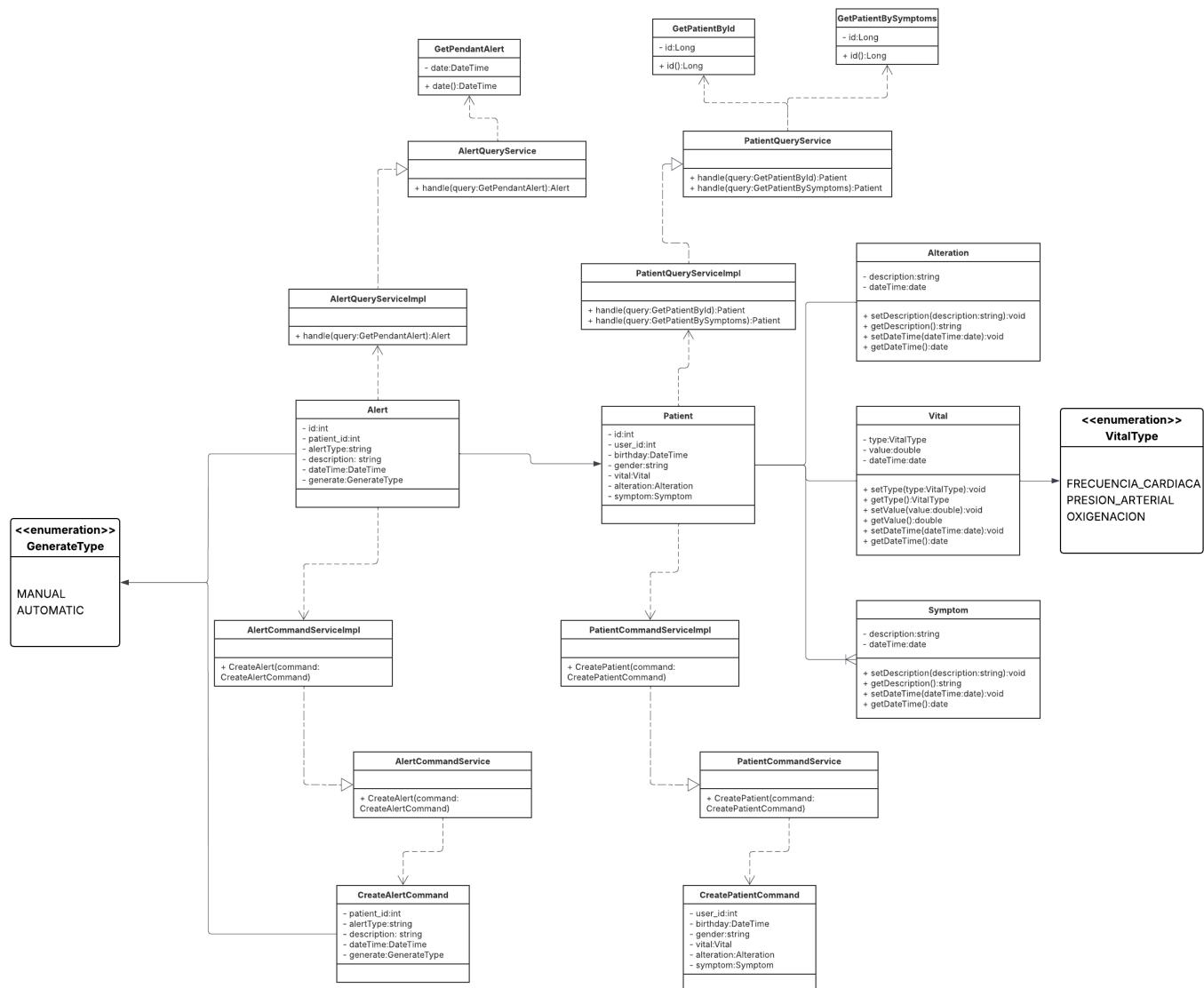
4.2.3.5. Bounded Context Software Architecture Component Level Diagrams



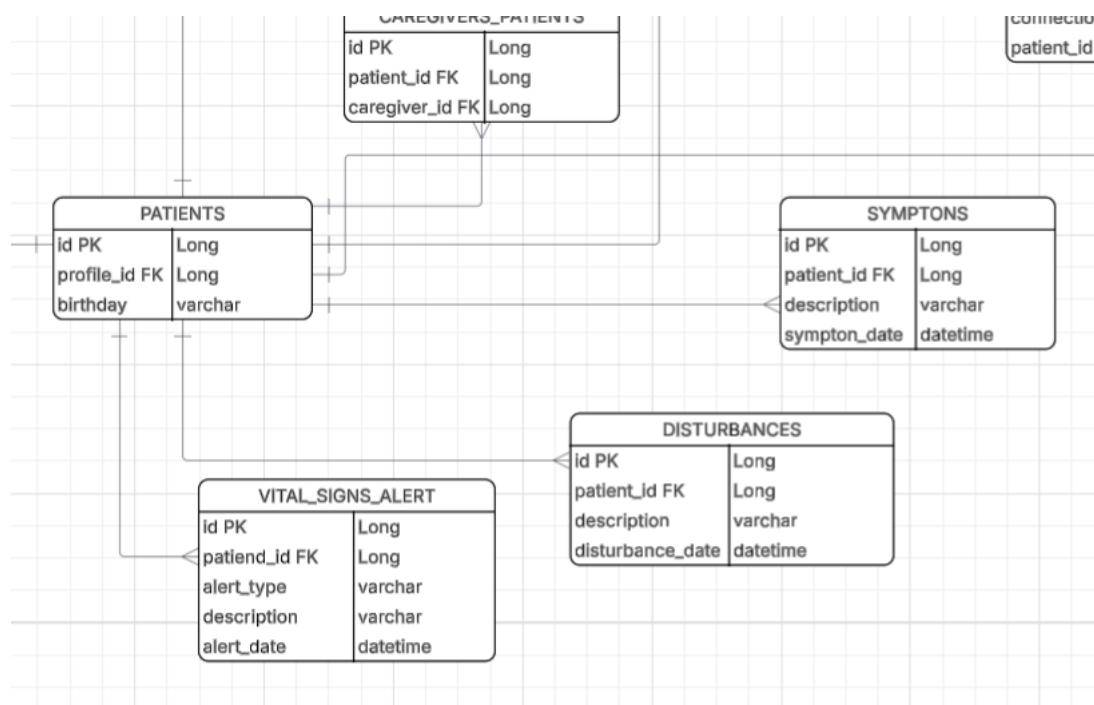
4.2.3.6. Bounded Context Software Architecture Code Level Diagrams

4.2.3.6.1. Bounded Context Domain Layer Class Diagrams

Este diagrama UML representa la arquitectura de un sistema de gestión médica centrada en el paciente y las alertas. La estructura está basada en principios de diseño orientado a objetos y se organiza en capas de comandos y consultas, siguiendo un enfoque CQRS (Command Query Responsibility Segregation). A lo largo del análisis, se describirán los principales componentes del sistema, como los servicios de comandos y consultas, las entidades centrales como Patient y Alert, así como los objetos de valor (value objects) y comandos asociados. También se explicará cómo estas clases interactúan entre sí mediante relaciones de herencia, interfaces, asociación y dependencia para mantener una lógica de negocio clara y mantenible. Se estará usando Lombok para la creación de Getter y Setter para evitar en redundancia de código.



4.2.3.6.2. Bounded Context Database Design Diagram



4.2.4. Bounded Context: <Monitoring>

4.2.4.1. Domain Layer

Sub-capa Model:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Aggregate	Monitoring	Clase que representa un registro de monitoreo	Ser el punto de entrada para la gestión de datos de monitoreo	Relacionado con eventos de logging, alertas y métricas recibidas desde otros bounded contexts.
Command	CreateMonitoringEventCommand	Comando para registrar un nuevo evento de monitoreo	Crear un nuevo registro de monitoreo	Relacionado con el Bounded Context de Infraestructura (captura de logs, métricas de red, BD, etc.).
Query	GetMonitoringStatusQuery	Query para obtener el estado general del sistema	Consultar métricas y eventos registrados	Se usa desde la capa de Application para mostrar datos al Dashboard.
Query	GetAlertsBySeverityQuery	Query para obtener alertas filtradas por nivel de severidad	Permitir análisis de criticidad en el monitoreo	Relacionado con el módulo de alertas y con el bounded context de <i>Notifications</i> .

Sub-capa Services:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Interface	MonitoringCommandService	Servicio que define métodos para registrar eventos y métricas	Estipular contratos claros para registrar datos	Usado en la capa "application" para implementar la lógica de registro
Interface	MonitoringQueryService	Servicio que define métodos para obtener información del monitoreo	Estipular contratos claros para consultas	Usado en la capa "application" para implementar consultas al sistema

4.2.4.2. Interface Layer

Sub-capa REST:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Controller	MonitoringController	Controlador REST para gestionar el monitoreo	Recibe solicitudes de clientes para consultar métricas o registrar eventos	Utiliza resources y assemblers
Resource	MonitoringRequestResource	Estructura de una petición para registrar un evento de monitoreo	Representar datos de entrada de forma accesible para el sistema	Usado en MonitoringController para registrar eventos
Resource	MonitoringResponseResource	Estructura de respuesta con datos de monitoreo	Exponer métricas, logs o alertas al cliente	Usado en MonitoringController para devolver resultados
Assembler	CreateMonitoringCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación	Evitar acoplamiento entre la interfaz REST y el dominio	Usado en MonitoringController
Assembler	MonitoringResourceFromEntityAssembler	Convierte una entidad MonitoringRecord en un recurso de respuesta	Asegurar consistencia en la información retornada	Usado en MonitoringController

4.2.4.3. Application Layer

Application Layer - Internal

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Command	RecordVitalSignCommand	Registrar un nuevo valor de signo vital de un paciente.	Validar y persistir un nuevo registro en la tabla SignosVitales.	Usa MonitoringService y MonitoringRepository
Command	ValidateVitalSignCommand	Validar un signo vital contra thresholds clínicos.	Consultar thresholds locales y, si está disponible, delegar validación a RulesService.	Depende de MonitoringService y RulesService

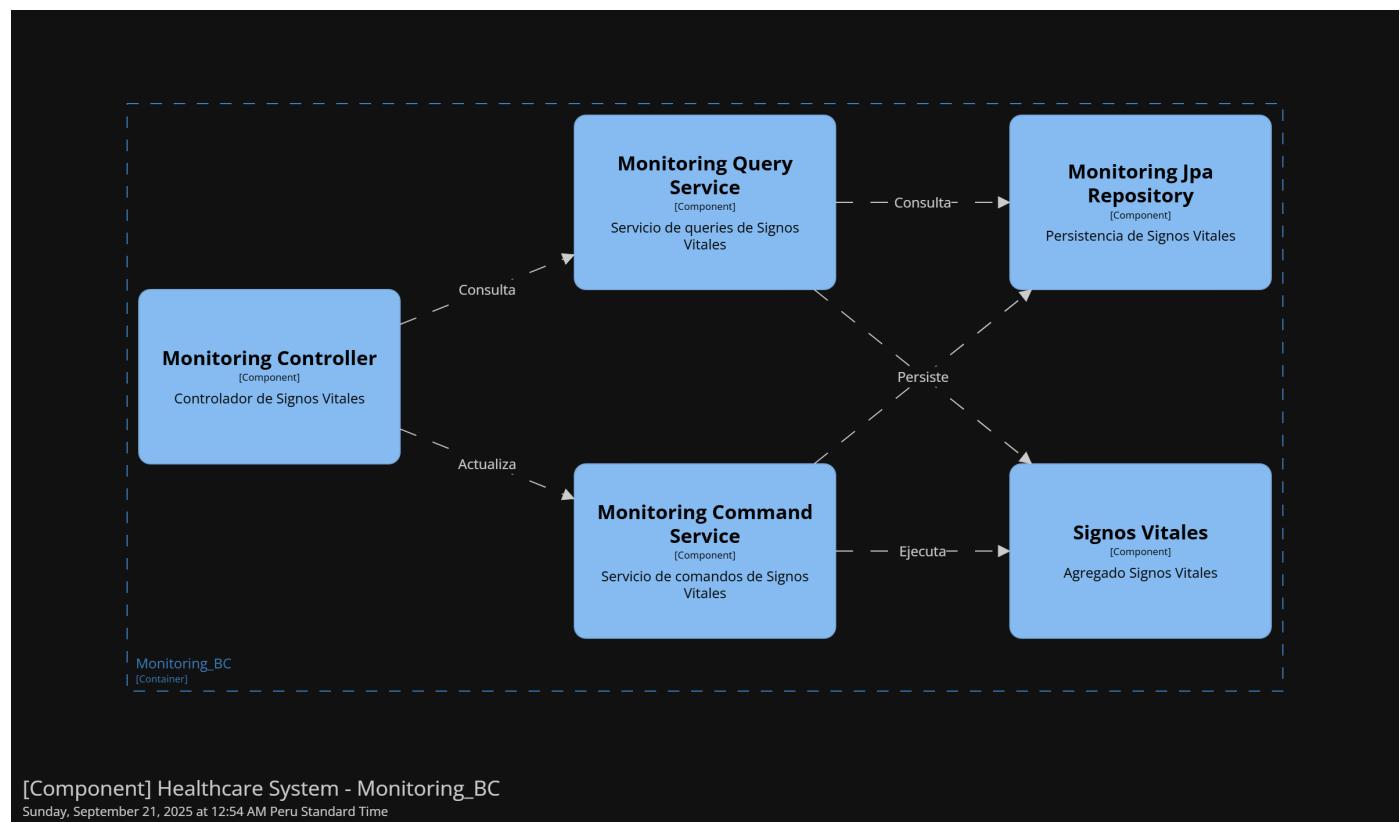
Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Query	GetMonitoringStatusQuery	Obtener estado actual de los signos vitales registrados, validando contra thresholds locales y/o RulesService.	Permite consultas del estado de un paciente en tiempo real.	Usa MonitoringRepository y RulesService

Sub-capa External:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
EventPublisher	MonitoringAlertPublisher	Publicador de eventos críticos detectados en el monitoreo vía EventBus	Enviar una alerta al Bounded Context de Notifications y/o EmergencyManagement para que gestionen notificaciones o llamadas SOS	Conecta con EventBus para distribuir eventos a otros BCs

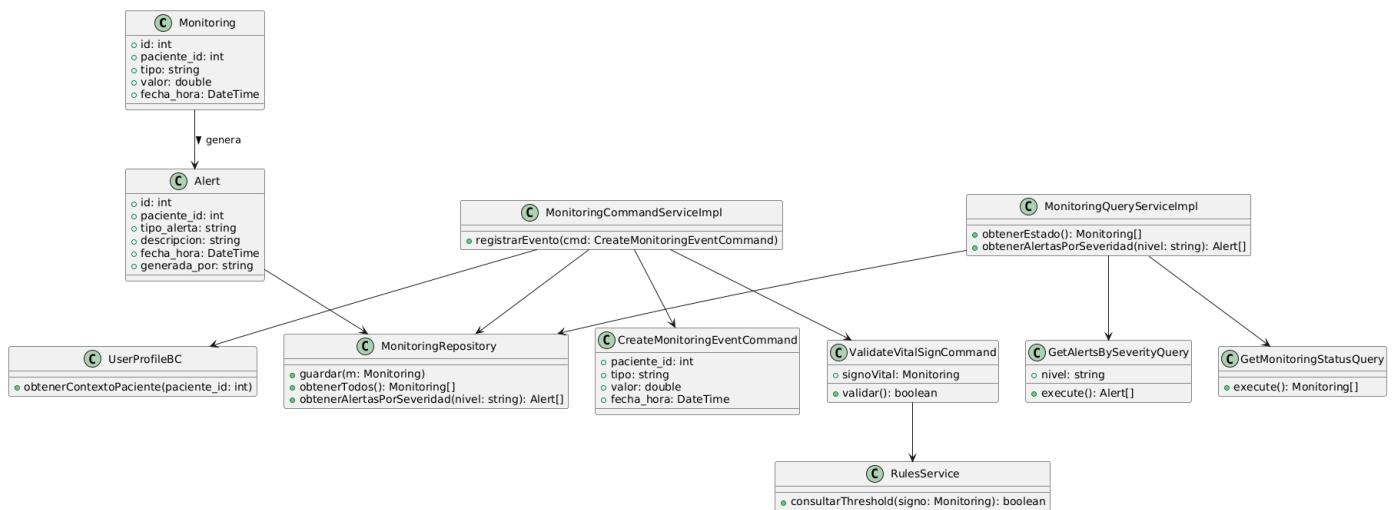
4.2.4.4. Infrastructure Layer**Sub-capa Repository:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	MonitoringRepository	Repositorio del modelo MonitoringRecord	Acceder y manipular los registros persistidos	Usado en la capa "Application" para almacenar y recuperar eventos

4.2.4.5. Bounded Context Software Architecture Component Level Diagrams**[Component] Healthcare System - Monitoring_BC**

Sunday, September 21, 2025 at 12:54 AM Peru Standard Time

4.2.4.6. Bounded Context Software Architecture Code Level Diagrams**4.2.4.6.1. Bounded Context Domain Layer Class Diagrams**



4.2.4.6.2. Bounded Context Database Design Diagram

Al ser datos de entrada proveniente del dispositivo IoT, la persistencia la gestiona el boundedContext "HealthTracking".

4.2.5. Bounded Context: <Emergency Management>

4.2.5.1. Domain Layer

Sub-capa Model:

Tipo	Nombre	Descripción	Responsabilidad principal	Relación
Aggregate	Emergencia	Evento crítico (caída/SOS/SV).	Crear/validar/actualizar/cerrar emergencias.	Con Alerta y LlamadaSOS ; insumos de Monitoring; publica a Notifications.
Command	TriggerSOSCommand	Disparar SOS manual.	Crear emergencia (SOS) y disparar alertas/llamada.	App paciente.
Command	DetectEmergencyCommand	Alta automática.	Crear emergencia con severidad/ubicación.	Desde Monitoring.
Command	ConfirmEmergencyCommand	Confirmar/descartar.	Cambiar estado (seguir/cancelar flujo).	App cuidador/paciente.
Command	DispatchAlertsCommand	Enviar alertas.	Generar alertas por canal y marcar estado.	Notifications.
Command	PlaceSOSCallCommand	Llamar 106/contactos.	Ejecutar llamada y registrar resultado.	Telco/API voz.
Query	GetActiveEmergenciesQuery	Emergencias abiertas.	Listar en curso con severidad/ubicación.	Dashboard.
Query	GetEmergencyByldQuery	Detalle.	Recuperar emergencia + alertas + llamada.	—

Sub-capa Services:

Tipo	Nombre	Descripción	Responsabilidad principal	Relación
Interface	EmergencyCommandService	Comandos de emergencia.	Trigger/Detect/Confirm/Close.	Repos de Emergencia/LlamadaSOS.
Interface	EmergencyQueryService	Lecturas.	Active/Byld.	Vistas CQRS.
Interface	AlertCommandService	Envío de alertas.	Dispatch + ConsentPolicy.	Notifications.
Interface	AlertQueryService	Lecturas alertas.	Historial/estados.	Auditoría.
Interface	SOSCallService	Llamadas.	Ejecutar/reintentos.	Telco.
Interface	ContactCommandService	ABM contactos.	Add/Update/Remove + validación.	ContactoEmergencia.
Interface	ContactQueryService	Lecturas contactos.	Listado/consentimientos.	—
Interface	GeoLocationService	Ubicación.	Actualizar/normalizar.	GPS/VO Ubicación.
Interface	SeverityAssessmentService	Regla.	Calcular severidad.	Monitoring.

4.2.5.2. Interface Layer

Sub-capa REST:

Tipo	Nombre	Descripción	Responsabilidad	Relación
Controller	EmergencyController	Endpoints de emergencias.	TriggerSOS, Detect, Active, ById.	EmergencyCommand/QueryService.
Request	TriggerSOSRequest	Payload disparo SOS.	pacientId, ubicación.	→ TriggerSOSCommand.
Request	DetectEmergencyRequest	Payload automática.	pacientId, tipo, severidad.	→ DetectEmergencyCommand.
Response	EmergencyResponse	Resumen de emergencia.	id, tipo, estado, ubicación.	Desde entidad Emergencia.
Assembler	EmergencyCommandAssembler	Request → Command.	Mapear Trigger/Detect.	Usado en Controller.
Assembler	EmergencyResponseAssembler	Entity → Response.	Formato consistente de salida.	Usado en Controller.

4.2.5.3. Application Layer**Sub-capa Internal:**

Tipo	Nombre	Descripción	Relación
CmdHandler	EmergencyCommandServiceImpl	Ejecuta TriggerSOS, DetectEmergency.	Usa EmergenciaRepo, SeverityService.
CmdHandler	AlertCommandServiceImpl	Manejo de alertas.	Usa AlertaRepo, NotificationACL.
CmdHandler	SOSCallServiceImpl	Llamadas SOS y reintentos.	Usa LlamadaSOSRepo, TelephonyACL.
QueryHandler	EmergencyQueryServiceImpl	Consultas de emergencias.	Usa vistas CQRS.
QueryHandler	AlertQueryServiceImpl	Consultas de alertas.	Usa AlertaReadModel.
QueryHandler	ContactQueryServiceImpl	Consultas de contactos.	Usa ContactoReadModel.

Sub-capa External:

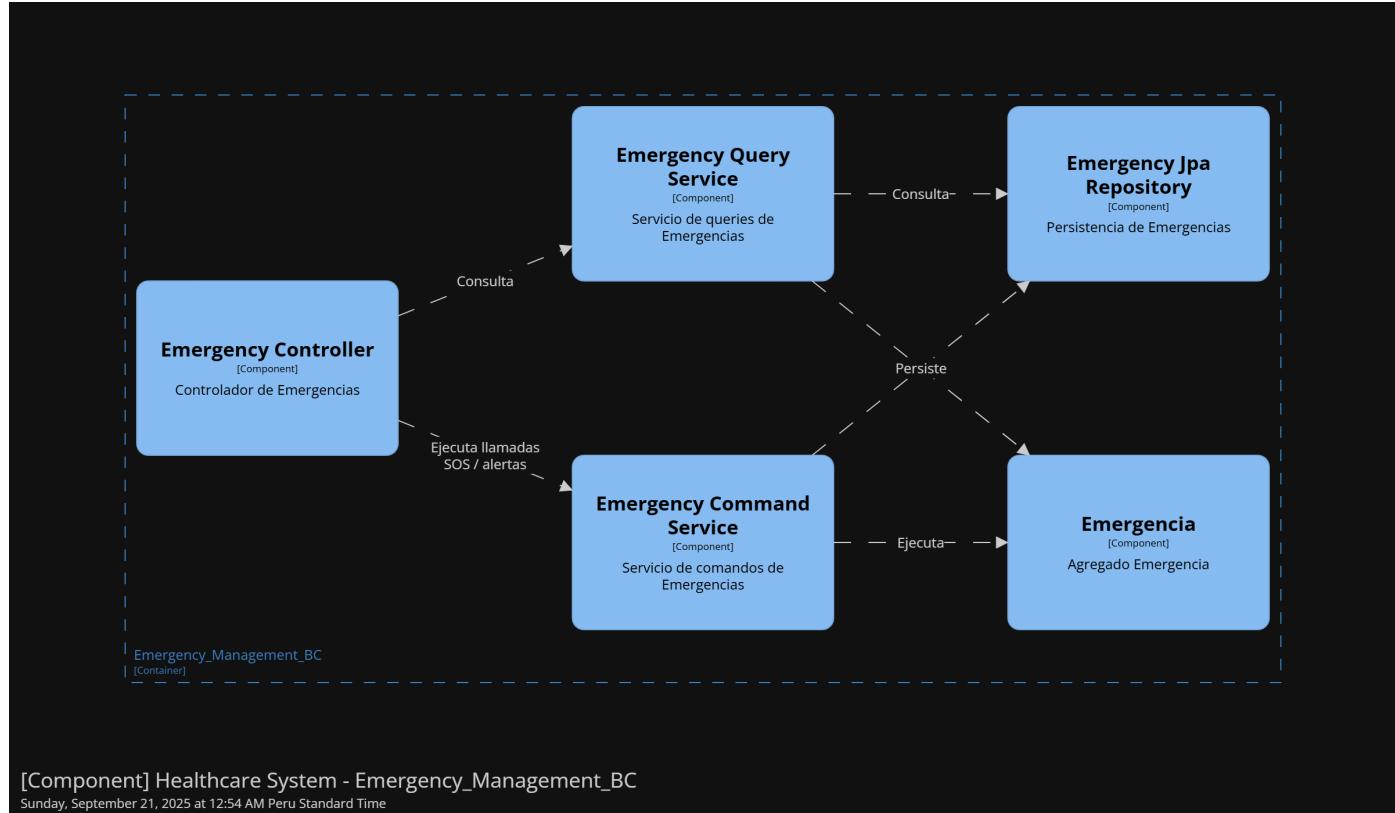
Tipo	Nombre	Descripción	Relación
ACL	NotificationACL	Envío SMS/WhatsApp/Push.	AlertService.
ACL	TelephonyACL	Llamadas 106/contactos.	SOSCallService.
ACL	GeoProviderACL	Ubicación/GPS.	GeoLocationService.
ACL	MonitoringACL	Eventos de signos vitales/caídas.	EmergencyService.
Messaging	EventBus	Transporte de eventos.	Notif. + Emergencies.

4.2.5.4. Infrastructure Layer**Sub-capa Repository:**

Tipo	Nombre	Descripción	Relación
Repository	EmergenciaRepository	Persistir emergencias y estado.	EmergencyService.
Repository	AlertaRepository	Persistir alertas y estados de entrega.	AlertService.
Repository	LlamadaSOSRepository	Guardar llamadas y resultados.	SOSCallService.
Repository	ContactoRepository	Gestionar contactos de emergencia.	ContactService.
Repository	ConsentRepository	Consentimientos por canal/contacto.	ConsentPolicy.
Repository	UbicacionRepository	Guardar ubicación histórica en emergencias.	GeoLocationService.

4.2.5.5. Bounded Context Software Architecture Component Level Diagrams

A continuación se ilustra la descomposición interna del contenedor Emergency Management Application, responsable de orquestar el ciclo de vida de una emergencia (SOS manual o automática por caída/umbral de signos vitales), coordinar notificaciones multicanal, llamadas al 106/contactos y el tracking de ubicación en tiempo real. El diseño refleja la separación por capas (Interface → Application → Domain → Infrastructure) y la integración con contextos/sistemas externos.



[Component] Healthcare System - Emergency_Management_BC

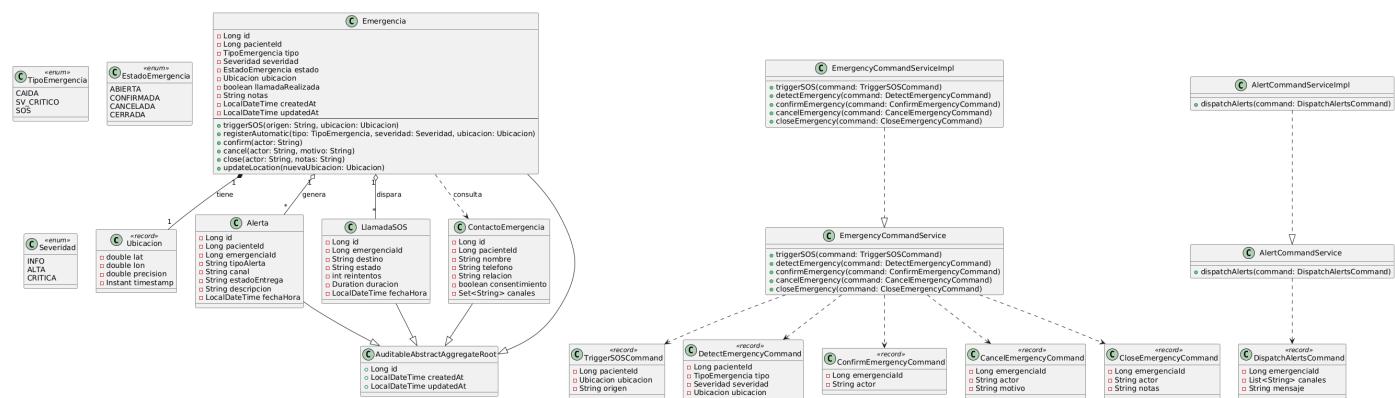
Sunday, September 21, 2025 at 12:54 AM Peru Standard Time

4.2.5.6. Bounded Context Software Architecture Code Level Diagrams

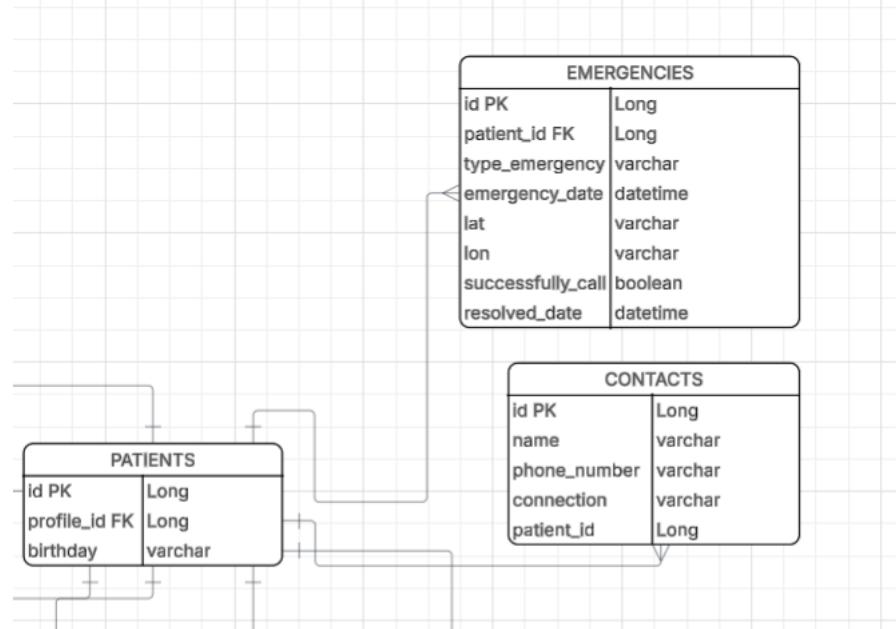
4.2.5.6.1. Bounded Context Domain Layer Class Diagrams

Emergencia es un Aggregate Root del dominio que modela un evento crítico asociado a un paciente (p. ej., caída detectada, signos vitales críticos o SOS manual). Contiene los atributos esenciales para gestionar el ciclo de vida de la emergencia (tipo, severidad, estado, ubicación, marca de llamada) y define comportamientos de negocio para disparar, confirmar/cancelar, cerrar y actualizar la ubicación durante el incidente. Esta clase actúa como objeto de dominio y es el punto de entrada para coordinar Alertas y Llamadas SOS.

Usa la interface/base AuditableAbstractAggregateRoot para heredar metadatos de auditoría (p. ej., createdAt, createdBy, updatedAt, updatedBy) y la publicación de eventos de dominio.



4.2.5.6.2. Bounded Context Database Design Diagram



EMERGENCIES

Propósito: Registro de situaciones de emergencia

Campo	Tipo	Descripción
id	Long (PK)	Identificador único
patient_id	Long (FK)	Referencia a PATIENTS
type_emergency	varchar	Tipo de emergencia
emergency_date	datetime	Fecha de la emergencia
lat	varchar	Latitud de la ubicación
lon	varchar	Longitud de la ubicación
successfully_call	boolean	Si se logró contactar ayuda
resolved_date	datetime	Fecha de resolución

Relaciones:

- N:1 con PATIENTS

CONTACTS

Propósito: Contactos de emergencia para pacientes

Campo	Tipo	Descripción
id	Long (PK)	Identificador único
name	varchar	Nombre del contacto
phone_number	varchar	Número de teléfono
connection	varchar	Relación con el paciente
patient_id	Long (FK)	Referencia a PATIENTS

Relaciones:

- N:1 con PATIENTS

4.2.6. Bounded Context: <Treatment Support>

Siguiendo el modelo de arquitectura "Clean Architecture" hemos dividido el proyecto en capas.

a continuación detallaremos las capas del Bounded Context referenciado.

4.2.6.1. Domain Layer

Sub-capa Model:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Aggregate	Treatment	Clase para definir el tratamiento del adulto mayor	Ser el punto de entrada para modificar y mantener la integridad del tratamiento como entidad del dominio	Útil para generar alertas de recordatorios como parte del tratamiento
Aggregate	Reminder	Clase para definir el recordatorio del consumo de medicamentos del adulto mayor	Ser el punto de entrada para modificar y mantener la integridad del recordatorio como entidad del dominio	Útil para generar alertas para que el adulto mayor consuma el medicamento en el horario establecido
Command	CreateTreatmentCommand	Comando para registro de tratamiento	Representa la intención de registrar tratamiento en la aplicación	Usado para registrar un nuevo tratamiento
Command	UpdateTreatmentCommand	Comando para actualización de tratamiento	Representa la intención de actualizar tratamiento en la aplicación	Usado para actualizar un nuevo tratamiento
Command	DeleteTreatmentCommand	Comando para eliminación de tratamiento	Representa la intención de eliminar tratamiento en la aplicación	Usado para eliminar un nuevo tratamiento
Command	CreateReminderCommand	Comando para registro de recordatorio	Representa la intención de registrar un recordatorio de un medicamento en la aplicación	Usado para registrar un nuevo recordatorio
Command	UpdateReminderCommand	Comando para actualización de recordatorio	Representa la intención de actualizar un recordatorio ya creado	Usado para actualizar un recordatorio existente
Command	DeleteReminderCommand	Comando para eliminación de recordatorio	Representa la intención de eliminar un recordatorio ya creado	Usado para eliminar un recordatorio existente
Query	GetAllRemindersByPatientQuery	Consulta para obtener todos los recordatorios de un adulto mayor específico	Permite recuperar la lista de recordatorios asociados a un paciente para visualizar sus próximos medicamentos	Usado para mostrar los recordatorios activos del adulto mayor
Query	GetAllTreatmentsByPatientQuery	Query para obtener todos los tratamientos de un paciente específico	Permite consultar la lista de tratamientos asociados a un adulto mayor en particular	Usado para mostrar todos los tratamientos de un paciente identificado

Sub-capa Services:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Interface	TreatmentCommandService	Servicio para la creación, modificación o eliminación de los tratamientos	estipular una estructura clara a seguir	Uso en la capa "application" para implementar los métodos dados
Interface	ReminderCommandService	Servicio para la creación, modificación y actualización de recordatorios	Estipular una estructura clara a seguir	Uso en la capa "Application" para la implementación de los métodos.
Interface	TreatmentQueryService	Servicio para métodos de obtención de datos	estipular una estructura clara a seguir.	Uso en la capa "application" para implementar el método estipulado.
Interface	ReminderQueryService	Servicio para métodos de obtención de recordatorios	estipular una estructura clara a seguir.	Uso en la capa "application" para implementar el método estipulado.

4.2.6.2. Interface Layer**Sub-capa REST:**

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Controller	TreatmentController	Controlador REST para gestionar tratamientos	Recibe solicitudes del cliente relacionadas con tratamientos, coordina comandos y queries	Utiliza TreatmentRequestResource, TreatmentResponseResource y los assemblers correspondientes
Controller	ReminderController	Controlador REST para gestionar recordatorios	Recibe solicitudes del cliente relacionadas con recordatorios, coordina comandos y queries	Utiliza ReminderRequestResource, ReminderResponseResource y los assemblers correspondientes
Resource	TreatmentRequestResource	Estructura de una petición para crear o actualizar un tratamiento	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente	Uso en TreatmentController para enviar datos estructurados al sistema

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Resource	TreatmentResponseResource	Estructura de una respuesta con información de un tratamiento	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente	Uso en TreatmentController para responder con datos del sistema
Resource	ReminderRequestResource	Estructura de una petición para crear o actualizar un recordatorio	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente	Uso en ReminderController para enviar datos estructurados al sistema
Resource	ReminderResponseResource	Estructura de una respuesta con información de un recordatorio	Representar y exponer datos del dominio de forma accesible y estructurada para el cliente	Uso en ReminderController para responder con datos del sistema
Assembler	CreateTreatmentCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación de tratamiento	Evitar acoplamiento entre la interfaz REST y la capa de aplicación	Usado en TreatmentController para traducir la petición a CreateTreatmentCommand
Assembler	UpdateTreatmentCommandFromResourceAssembler	Convierte un recurso de petición en un comando de actualización de tratamiento	Evitar acoplamiento entre la interfaz REST y la capa de aplicación	Usado en TreatmentController para traducir la petición a UpdateTreatmentCommand
Assembler	TreatmentResourceFromEntityAssembler	Convierte una entidad Treatment en un recurso de respuesta	Asegura una respuesta consistente para el cliente	Usado en TreatmentController para retornar TreatmentResponseResource
Assembler	CreateReminderCommandFromResourceAssembler	Convierte un recurso de petición en un comando de creación de recordatorio	Evitar acoplamiento entre la interfaz REST y la capa de aplicación	Usado en ReminderController para traducir la petición a CreateReminderCommand
Assembler	UpdateReminderCommandFromResourceAssembler	Convierte un recurso de petición en un comando de actualización de recordatorio	Evitar acoplamiento entre la interfaz REST y la capa de aplicación	Usado en ReminderController para traducir la petición a UpdateReminderCommand
Assembler	ReminderResourceFromEntityAssembler	Convierte una entidad Reminder en un recurso de respuesta	Asegura una respuesta consistente para el cliente	Usado en ReminderController para retornar ReminderResponseResource

4.2.6.3. Application Layer

Sub-capa Internal:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
CommandHandler	TreatmentCommandServiceImpl	Implementación de los comandos de creación, actualización y eliminación de tratamientos	Implementar los métodos para el servicio de tratamientos	Implementa los métodos de la interfaz TreatmentCommandService en la capa de "Services".
CommandHandler	ReminderCommandServiceImpl	Implementación de los comandos de creación, actualización y eliminación de recordatorios	Implementar los métodos para el servicio de recordatorios	Implementa los métodos de la interfaz ReminderCommandService en la capa de "Services".
QueryHandler	TreatmentQueryServiceImpl	Implementación de la obtención de datos de tratamientos	Implementar los métodos para el servicio de consultas de tratamientos	Implementa los métodos de la interfaz TreatmentQueryService en la capa de "Services".
QueryHandler	ReminderQueryServiceImpl	Implementación de la obtención de datos de recordatorios	Implementar los métodos para el servicio de consultas de recordatorios	Implementa los métodos de la interfaz ReminderQueryService en la capa de "Services".

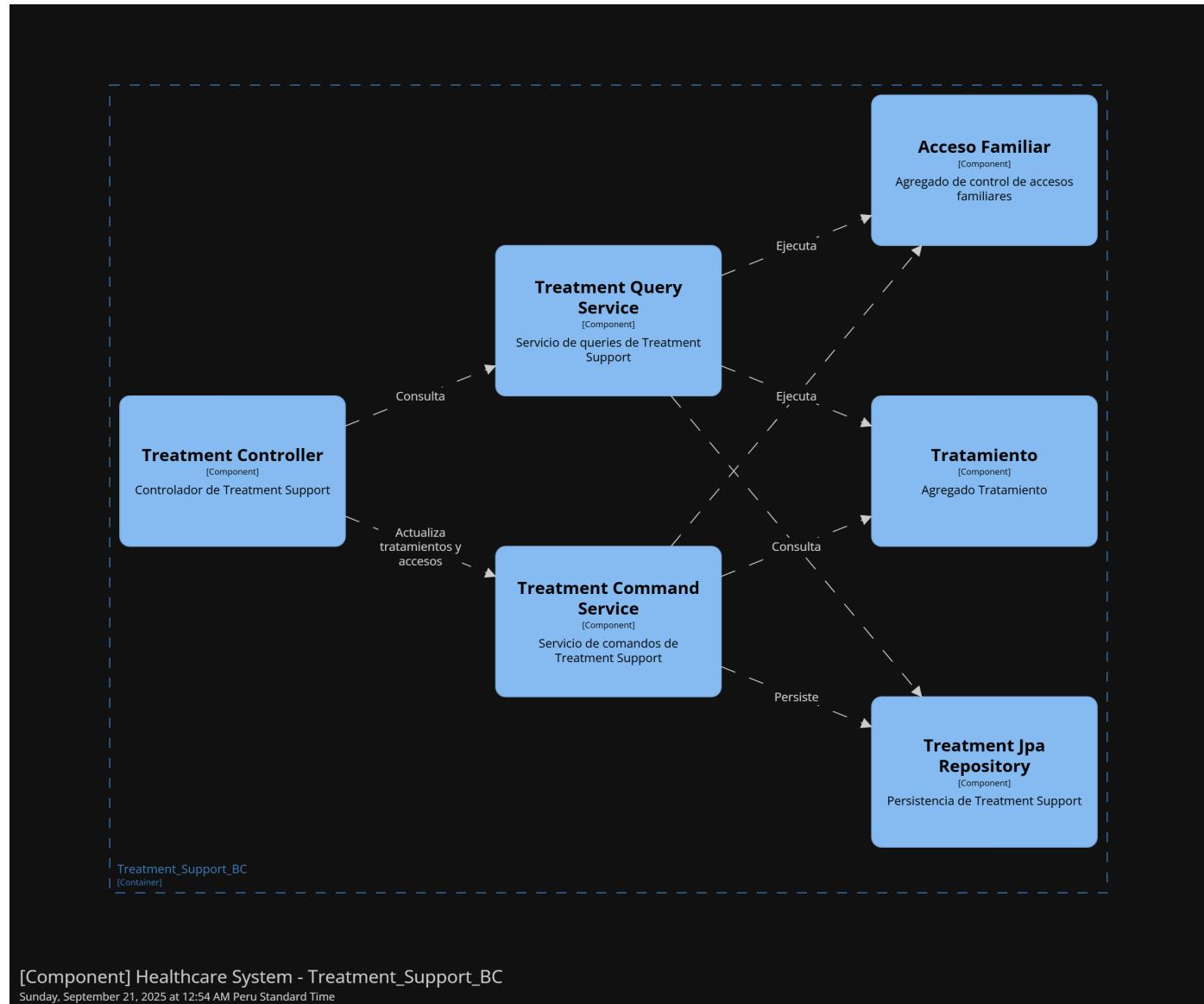
4.2.6.4. Infrastructure Layer

Sub-capa Repository:

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	TreatmentRepository	Repositorio a usar del modelo "Treatment"	Acceder y manipular datos persistidos de tratamientos	Usado en la Capa "Application" para actualizar y obtener datos de tratamientos.

Tipo	Nombre	Descripción	Responsabilidad Principal	Relación con otros elementos
Repository	ReminderRepository	Repositorio a usar del modelo "Reminder"	Acceder y manipular datos persistidos de recordatorios	Usado en la Capa "Application" para actualizar y obtener datos de recordatorios.

4.2.6.5. Bounded Context Software Architecture Component Level Diagrams

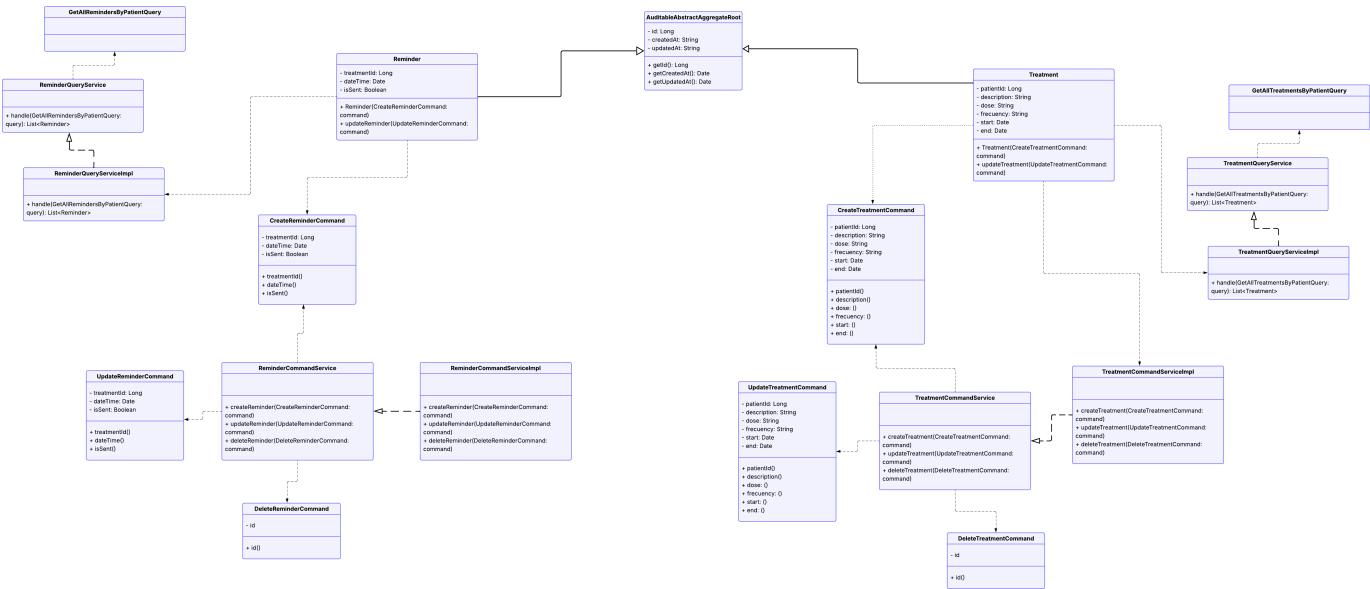


4.2.6.6. Bounded Context Software Architecture Code Level Diagrams

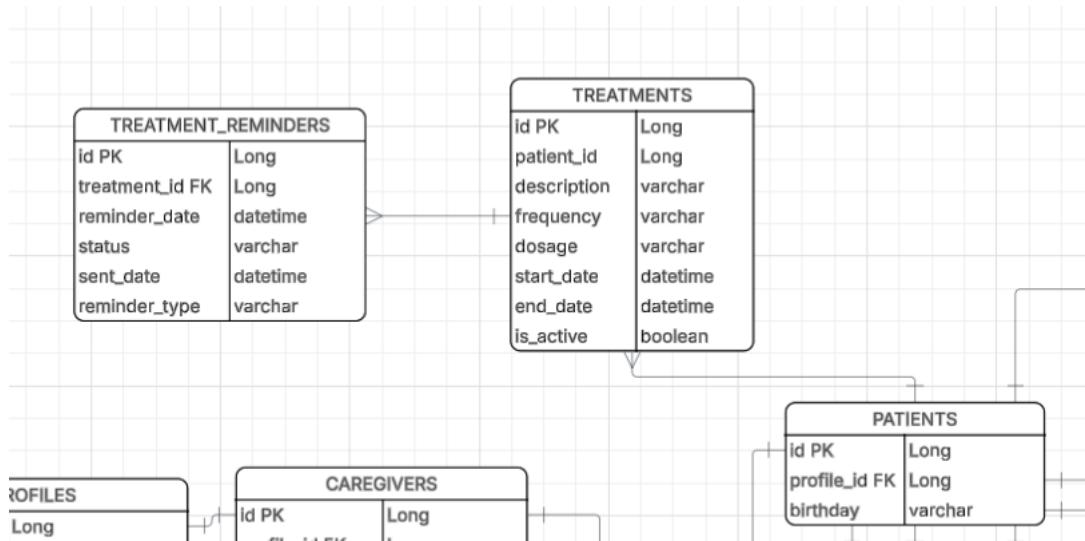
4.2.6.6.1. Bounded Context Domain Layer Class Diagrams

Este diagrama de clases representa la estructura del modelo de dominio para el módulo de Treatment Support en una aplicación desarrollada con principios de Clean Architecture. La entidades principales son Tratamiento y Recordatorio, la cual hereda de una clase base llamada `AuditableAbstractAggregateRoot` que encapsula atributos comunes como `id`, `createdAt` y `updatedAt`, asegurando trazabilidad y consistencia en la gestión de entidades auditables. Además, se utilizará Lombok para la generación automática de métodos `get` y `set`, lo que reducirá el código repetitivo y mejorando la legibilidad del modelo.

El diseño refleja una clara orientación a DDD (Domain-Driven Design), donde las entidades mantienen su lógica y reglas de negocio, apoyadas por una base auditible y clasificadas semánticamente mediante enumeraciones. Esta estructura promueve la escalabilidad, cohesión y mantenibilidad del sistema.



4.2.6.6.2. Bounded Context Database Design Diagram



TREATMENTS

Propósito: Gestión de tratamientos médicos y medicamentos

Campo	Tipo	Descripción
<code>id</code>	Long (PK)	Identificador único del tratamiento
<code>patient_id</code>	Long (FK)	Referencia a PATIENTS
<code>description</code>	varchar	Descripción del tratamiento
<code>frequency</code>	varchar	Frecuencia de administración
<code>dosage</code>	varchar	Dosis del medicamento
<code>start_date</code>	datetime	Fecha de inicio del tratamiento
<code>end_date</code>	datetime	Fecha de fin del tratamiento
<code>is_active</code>	boolean	Estado activo del tratamiento

Relaciones:

- N:1 con PATIENTS
- 1:N con TREATMENT_REMINDERS

TREATMENT_REMINDERS

Propósito: Gestión de recordatorios de medicación

Campo	Tipo	Descripción
-------	------	-------------

Campo	Tipo	Descripción
id	Long (PK)	Identificador único del recordatorio
treatment_id	Long (FK)	Referencia a TREATMENTS
reminder_date	datetime	Fecha y hora del recordatorio
status	varchar	Estado (pending/sent/failed)
sent_date	datetime	Fecha de envío real
reminder_type	varchar	Tipo (push/sms/email)

Relaciones:

- N:1 con TREATMENTS

Capítulo V: Solution UI/UX Design

5.1. Style Guidelines.

5.1.1. General Style Guidelines.

Nombre y Logotipo

El nombre AlertaVital transmite seguridad, tecnología y cuidado humano, reflejando el compromiso de la startup con la innovación en salud y el bienestar de las personas mayores. El logotipo de AlertaVital está compuesto por un símbolo y un logotipo tipográfico, donde el icono combina elementos de tecnología y salud, como un corazón digital o una onda de pulso conectada, representando la conectividad IoT y la monitorización de signos vitales en tiempo real.

El diseño general es limpio, minimalista y moderno, asegurando una identidad visual fácil de reconocer y adaptable a distintos entornos digitales, como aplicaciones móviles, wearables y plataformas web. El conjunto visual busca inspirar confianza, empatía y tranquilidad, alineado con la misión de la marca: cuidar y proteger la vida de los adultos mayores mediante soluciones tecnológicas accesibles.



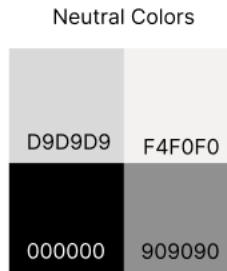
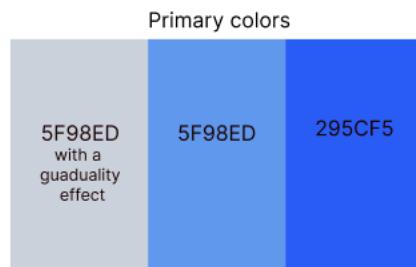
AlertaVital

Tono de Comunicación

El tono de comunicación de AlertaVital será humano, confiable, empático y profesional. Se evita el lenguaje excesivamente técnico o distante, priorizando una comunicación clara y accesible para usuarios de todas las edades, especialmente adultos mayores y sus familias. El mensaje de la marca debe proyectar tranquilidad y apoyo, actuando como un asistente confiable y cercano que ofrece soluciones tecnológicas sin perder la calidez humana.

Colors

La identidad visual de AlertaVital se basa en una paleta de azules y celestes en degradado, que representa confianza, calma y tecnología, al mismo tiempo que evoca una sensación de serenidad y bienestar. El uso del degradado entre azul y celeste simboliza la transición entre lo humano y lo digital, mostrando la conexión entre las personas y la tecnología de cuidado. Los colores secundarios son blanco y gris suave, que aportan claridad, equilibrio visual y legibilidad.



Typography

La tipografía principal de AlertaVital es Poppins, elegida por su claridad, modernidad y accesibilidad. Su estructura redondeada y legible transmite empatía y cercanía, mientras que su estilo geométrico refleja la precisión tecnológica del proyecto. Los títulos usan un peso seminegrita o negrita, los subtítulos un peso medio, y el texto regular se mantiene ligero y cómodo de leer en pantallas de diferentes tamaños.

ROBOTO

Heading 1 - bolt - 48px

Heading 2 - bolt - 40px

Heading 3 - bolt - 36 px

Text 1 - bolt - 24 px

Text 2 - bolt - 20 px

Text 3 - regular - 20 px

Button 1 - regular - 24 px

Spacing

El sistema de espaciado en la interfaz de AlertaVital prioriza la legibilidad y la accesibilidad visual. Se aplican márgenes amplios (20px) entre secciones principales, padding de 16–18px en botones y elementos interactivos, y un interlineado de 1.5x en textos largos. Este enfoque garantiza una experiencia ordenada y fluida, especialmente para usuarios mayores o con menor familiaridad tecnológica.

Íconos de la aplicación

Los íconos de AlertaVital son minimalistas, coherentes y de trazo redondeado, alineados con la identidad del logotipo. Su diseño utiliza el mismo degradado azul-celeste para mantener una apariencia homogénea en toda la interfaz. Cada ícono tiene un propósito funcional claro: alertas, monitoreo, conexión y cuidado. Su consistencia visual refuerza la confianza y facilita la comprensión inmediata de las acciones dentro de la app.

Icons



Usuario



Notificaciones



Oxigenacion



Emergencia



Sintomas



Recordatorios



Editar



Contacto de emergencia



Domicilio



Signos vitales



Soporte



Advertencia



Alerta vital

Lenguaje aplicado

El lenguaje de AlertaVital es formal, empático y tranquilizador, diseñado para inspirar confianza y cercanía. Se evita el uso de tecnicismos, priorizando frases cortas, claras y amables. La app y su web comunican con una voz que acompaña y guía, reflejando la visión de la startup: facilitar un envejecimiento saludable y autónomo mediante la tecnología.

Consistencia

AlertaVital mantiene consistencia visual y comunicativa en todas sus plataformas. El uso uniforme del degradado azul-celeste, la tipografía Poppins y los íconos minimalistas crea una identidad reconocible y confiable. Esta coherencia asegura que el usuario se sienta seguro y orientado, fortaleciendo la percepción de una marca tecnológica, empática y estable.

Simplicidad

El diseño prioriza la simplicidad y funcionalidad, eliminando elementos innecesarios y destacando solo la información relevante. El objetivo es que cualquier usuario, independientemente de su edad, pueda navegar fácilmente y comprender la app sin esfuerzo. La simplicidad refuerza la misión de AlertaVital: hacer que la tecnología del cuidado sea accesible y humana.

Comunicación

El sistema de comunicación visual de AlertaVital incluye confirmaciones visuales y auditivas al realizar acciones importantes, como registrar signos vitales o enviar una alerta. Estos elementos generan confianza y seguridad emocional, garantizando que el usuario siempre sepa que el sistema está activo y atento a su bienestar. Cada interacción busca transmitir calma, control y acompañamiento.

5.1.2. Web, Mobile and IoT Style Guidelines.

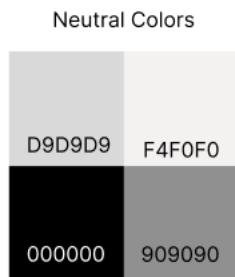
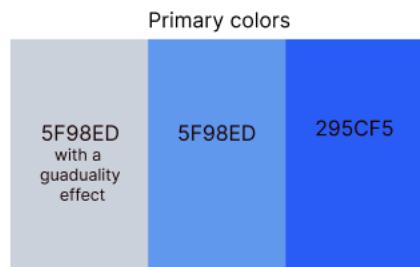
Diseño general

El diseño en los entornos web, móvil e IoT de AlertaVital mantiene una identidad visual uniforme, priorizando la claridad, accesibilidad y consistencia. Todos los componentes siguen la misma línea estética basada en colores azules y celestes degradados, acompañados de tonos neutros (blanco, gris y negro) para garantizar una presentación limpia, moderna y profesional.

El objetivo es que la experiencia del usuario sea fluida y coherente, sin importar el dispositivo o entorno desde el cual interactúe con la plataforma (navegador, aplicación móvil o dispositivo IoT).

Colores y degradados

La paleta principal de AlertaVital se centra en la gama de azules y celestes para reflejar confianza, tecnología y bienestar. El uso de degradados suaves entre estos tonos crea una apariencia dinámica y moderna, sin perder sobriedad ni legibilidad.



Aplicación por entorno:

- Web: degradados sutiles en cabeceras, botones de acción y secciones destacadas.
- Móvil: colores planos con acentos degradados para mejorar contraste y legibilidad en pantallas pequeñas.
- IoT: uso del azul principal para retroiluminación o pantallas LED, complementado por blanco y gris para mantener una lectura clara en entornos con poca luz.

Tipografía

Se utiliza la tipografía Poppins o Roboto, asegurando una experiencia legible y moderna en cualquier dispositivo. Su diseño geométrico y redondeado transmite cercanía y precisión, ideal para productos tecnológicos enfocados en el cuidado de la salud.

ROBOTO

Heading 1 - bolt - 48px

Heading 2 - bolt - 40px

Heading 3 - bolt - 36 px

Text 1 - bolt - 24 px

Text 2 - bolt - 20 px

Text 3 - regular - 20 px

Button 1 - regular - 24 px

Componentes visuales

Los elementos visuales de AlertaVital comparten una estética minimalista y funcional. Se prioriza la simplicidad para facilitar la interacción, especialmente en usuarios mayores.

Principales componentes:

- Botones: bordes redondeados, fondo azul-celeste degradado, texto blanco.
- Tarjetas de información: fondo blanco o gris claro, sombras suaves para destacar contenido.
- Alertas y notificaciones: colores azul degradado para alertas informativas y tonos celeste más brillante para mensajes positivos o de confirmación.

- Gráficos de signos vitales: líneas finas en azul o celeste, fondo neutro y legibilidad clara.

Diseño responsivo

El diseño es completamente adaptable y mantiene su coherencia visual en todos los dispositivos. Los elementos se ajustan según el entorno:

- Web: estructura modular, con secciones amplias y distribución en columnas.
- Móvil: diseño vertical, botones amplios y navegación táctil optimizada.
- IoT: interfaz reducida con elementos esenciales, priorizando claridad y reacción inmediata ante alertas.

Interacción y usabilidad

El sistema de diseño de AlertaVital se centra en la experiencia del usuario. Cada interacción debe generar claridad y confianza.

Principios de interacción:

- Retroalimentación inmediata: cambios de color o microanimaciones al interactuar con botones o íconos.
- Estados visuales claros: diferenciación entre botones activos, inactivos y presionados.
- Animaciones suaves: transiciones ligeras con degradados para dar sensación de fluidez.
- Iconografía uniforme: íconos redondeados, minimalistas, en azul o celeste con fondo blanco.

5.2. Information Architecture.

5.2.1. Organization Systems.

Web App (Plataforma de Monitoreo y Gestión)

La estructura de la Web App se organiza en módulos funcionales accesibles desde un panel principal:

- Dashboard general: muestra los signos vitales en tiempo real, alertas recientes y el estado del dispositivo.
- Usuarios: listado de adultos mayores registrados y sus cuidadores.
- Historial Médico: registros ordenados cronológicamente, con filtros por fecha y tipo de evento.
- Alertas y Reportes: panel para revisar y exportar incidentes de caídas o variaciones críticas.
- Configuración: gestión de cuentas, permisos y dispositivos IoT vinculados.

La jerarquía prioriza el acceso rápido a información crítica (alertas y monitoreo), mientras que las opciones administrativas se ubican en segundo nivel.

Mobile App (Aplicación para Usuarios y Cuidadores)

En la aplicación móvil, la organización se centra en la facilidad de uso táctil, utilizando botones grandes, íconos visuales y una navegación simplificada.

Las secciones principales son:

- Monitoreo en Tiempo Real (pantalla principal): muestra ritmo cardíaco, oxigenación y actividad del usuario.
- Alertas y Emergencias: notificaciones instantáneas con opción de contacto directo o envío de ubicación.
- Historial de Salud: registros visuales y gráficos simples de las últimas mediciones.
- Perfil del Usuario: información personal, configuraciones y dispositivos asociados.

Ejemplo: la opción "Enviar alerta" o "SOS" se presenta como un botón flotante permanente en color azul degradado, accesible desde cualquier pantalla.

Landing Page (Sitio Web Informativo y Comercial)

La landing page sigue una estructura informativa con secuencia narrativa, diseñada para presentar el producto y captar nuevos usuarios. Sus secciones principales son:

- Inicio: presentación del propósito de AlertaVital y su propuesta tecnológica.
- Nosotros: información de la startup y su misión social.
- Tecnología IoT: explicación del funcionamiento y beneficios del sistema.
- Productos: presentación de la App, Web App y Dispositivo IoT.
- Contáctanos: formulario, datos de contacto y enlaces a redes sociales.

El contenido se organiza en bloques con encabezados grandes, íconos ilustrativos y llamados a la acción (CTA) como "Descargar App" o "Solicitar una demostración".

Dispositivo IoT (Wearable Inteligente AlertaVital)

En el dispositivo IoT, la organización es mínima pero funcional, orientada a la inmediatez:

- Pantalla principal: muestra los signos vitales más relevantes (frecuencia cardíaca, oxigenación, nivel de actividad).

- Indicadores visuales (LED):
 - Azul: conexión activa.
 - Celeste: monitoreo estable.
 - Rojo: alerta de caída o emergencia.
- Botón físico SOS: envía una alerta inmediata al cuidador registrado.

La disposición busca evitar sobrecarga cognitiva, facilitando la respuesta rápida ante eventos críticos.

5.2.2. Labeling Systems.

Web App:

Las etiquetas son técnicas pero claras, orientadas a la gestión profesional. Ejemplos:

- "Panel de Monitoreo", "Historial Clínico", "Alertas Activas", "Configurar Dispositivo".

Mobile App:

Se usan etiquetas simples y de acción directa, adaptadas a un público general:

- "Ver Signos Vitales", "Enviar Alerta", "Historial", "Mi Perfil".

Landing Page:

Las secciones están etiquetadas con lenguaje natural, orientado a visitantes y posibles clientes:

- "Nosotros", "Tecnología IoT", "Productos", "Contáctanos".

Dispositivo IoT:

Etiquetas visuales o breves textos universales:

- "SOS", "OK", "Conectado", "Batería Baja"

5.2.3. SEO Tags and Meta Tags

Para maximizar la visibilidad en buscadores y tiendas de aplicaciones (ASO y SEO), AlertaVital implementa etiquetas descriptivas y optimizadas.

Web App / Landing Page:

- Título: "AlertaVital – Soluciones IoT para el Cuidado de Adultos Mayores".
- Meta descripción: "Monitorea signos vitales y detecta caídas en tiempo real con AlertaVital, tecnología IoT accesible para la seguridad y bienestar de los adultos mayores."
- Palabras clave: "IoT salud", "detección de caídas", "monitoreo remoto", "cuidado adulto mayor", "startup tecnología médica".

Mobile App:

- Título: "AlertaVital – Monitoreo y Alertas IoT".
- Descripción corta: "Seguridad y salud en tu bolsillo. Supervisa signos vitales y recibe alertas en tiempo real."
- Descripción extendida: "AlertaVital conecta a las familias con sus seres queridos mayores mediante tecnología IoT, ofreciendo monitoreo constante, alertas inmediatas y tranquilidad en todo momento."
- Palabras clave: "app salud IoT, adultos mayores, seguridad médica, wearable, cuidado remoto".

IoT Device:

Aunque no aplica SEO tradicional, se incluye metadata técnica en la app que lo gestiona (identificador del dispositivo, modelo, versión de firmware), útil para indexación en sistemas conectados y compatibilidad de API.

5.2.4. Searching Systems.

Web App: Incluye una barra de búsqueda avanzada con filtros por usuario, rango de fechas, tipo de alerta o parámetros fisiológicos (ej. "frecuencia cardíaca"). Ejemplo: un cuidador puede buscar "caídas de octubre" o "alertas de oxigenación baja" y acceder a los registros precisos.

Mobile App: Integra búsqueda contextual con resultados dinámicos, permitiendo localizar pacientes, alertas o historiales directamente desde el menú principal. Ejemplo: al escribir "Pedro López", el sistema muestra su historial y datos de monitoreo más recientes.

Landing Page: Dispone de una búsqueda informativa simple que facilita encontrar secciones o artículos como "Beneficios del IoT" o "Cómo funciona AlertaVital".

Dispositivo IoT: No dispone de búsqueda textual; su sistema "busca" de forma automatizada patrones irregulares en signos vitales, generando alertas automáticas al detectar anomalías.

5.2.5. Navigation Systems.

Menú Principal:

- **Web App:** Navegación lateral fija tipo dashboard, con iconos representativos y estados activos en azul degradado. Ejemplo: el módulo "Alertas" se resalta con color celeste brillante cuando hay notificaciones nuevas.
- **Mobile App:** Menú inferior con iconos grandes y etiquetas: "Inicio", "Alertas", "Historial", "Perfil". Ejemplo: un botón flotante de emergencia ("SOS") permite enviar alertas desde cualquier pantalla.
- **Landing Page:** Menú superior fijo con anclas internas a secciones como "Inicio", "Nosotros", "Tecnología", "Productos" y "Contáctanos". En móviles, se transforma en un menú hamburguesa con íconos minimalistas.
- **Dispositivo IoT:** Navegación física simple con botones "SOS" y "Menú". Indicadores visuales cambian de color según el estado del dispositivo, guiando al usuario sin necesidad de texto.

Navegación Contextual:

- **Web App:** Desde el historial de un usuario, se puede acceder a su monitoreo en tiempo real o enviar una alerta manual.
- **Mobile App:** Desde una alerta recibida, se puede abrir directamente el perfil del paciente y ver sus signos vitales actuales.
- **Landing Page:** En "Tecnología IoT", enlaces conducen a "Productos" o "Contáctanos", manteniendo la fluidez.
- **IoT Device:** Un toque prolongado en el botón principal activa la función SOS o reinicia la conexión, facilitando navegación táctil mínima.

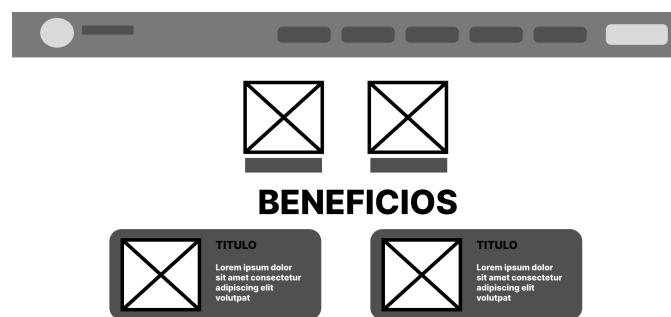
5.3. Landing Page UI Design.

5.3.1. Landing Page Wireframe.**Wireframe Landing Page (Desktop)**

Sección Principal de la landing page:



Sección Características y Beneficios:



Sección Antecedentes:



Antecedentes y Problemática

Sección "Acerca de":



ACERCA DE

Sección de Formulario de contacto:



¿Listo para mejorar la experiencia oncológica?

Contáctanos hoy mismo para una demostración personalizada.

Sección de Descargas de aplicación móvil y pie de página:



TITULO
Lorem ipsum dolor
TITULO TITULO
TITULO

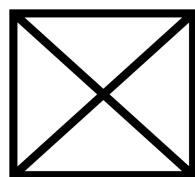


Wireframe Landing Page (Mobile)

Sección Principal de la landing page:



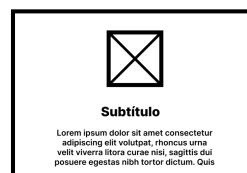
Subtitulo



Sección Características y Beneficios:



Subtítulo



Sección Antecedentes:



Subtítulo

Subtítulo

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Quis

Subtítulo

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Quis

Sección "Acerca de":



Subtítulo



Subtítulo

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Quis



Subtítulo

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Lorum ipsum dolor sit amet consectetur adipiscing elit volutpat, rhoncus urna velit viverra litora curae nisi, sagittis dui posuere egestas nibh tortor dictum.

Quis

Sección de Formulario de contacto:



Sección de Descargas de aplicación móvil y pie de página:



5.3.2. Landing Page Mock-up.

Mockups Landing Page (Desktop)

Sección Principal de la landing page

Características Principales

- Monitoreo de signos vitales
- Monitoreo de Caídas

Sección Características y Beneficios:

Beneficios

- Para Cuidadores
- Para personas vulnerables

Sección Antecedentes:

Antecedentes y Problemática

- ¿Cuáles son las causas?
- Desafíos en la supervisión de personas vulnerables

Sección "Acerca de":

Acerca de AlertaVital

- About The Product
- About The Team

Sección de Formulario de contacto:

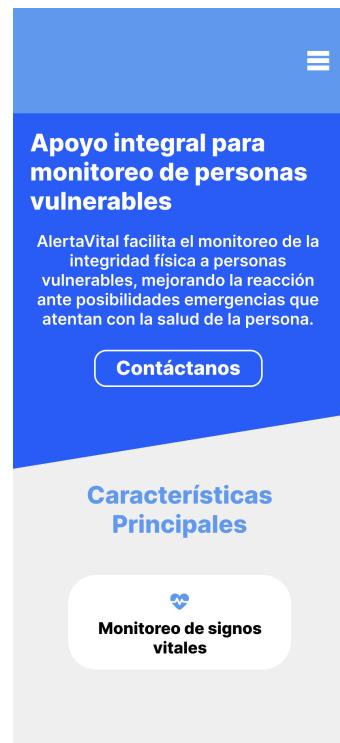


Sección de Descargas de aplicación móvil y pie de página:



Mockups Landing Page (Mobile)

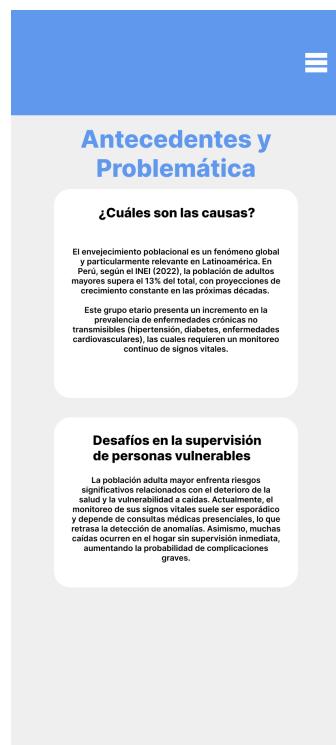
Sección Principal de la landing page



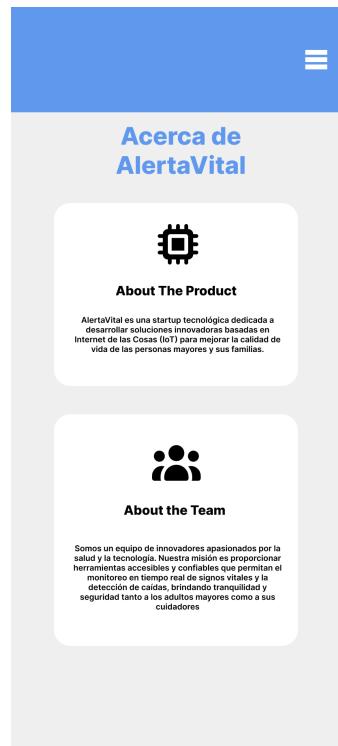
Sección Características y Beneficios:



Sección Antecedentes:



Sección "Acerca de":



Sección de Formulario de contacto:



Sección de Descargas de aplicación móvil y pie de página:



5.4. Applications UX/UI Design.

5.4.1. Applications Wireframes.

Application wireframes (Desktop)

Register

A wireframe of a registration form titled "Registration Form". It contains three input fields: "Full Name" (placeholder: Enter your full name), "Email" (placeholder: Enter your email), and "Password" (placeholder: Enter your password). Below the fields are two buttons: "Cancel" and "Register".

Login

A wireframe of a login form titled "Login". It includes a subtitle "Access your account to manage your health." followed by two input fields: "Email" (placeholder: Enter your email) and "Password" (placeholder: Enter your password). Below the inputs are three buttons: "Forgot Password?", "Login" (in a dark button), and "Register".

Dashboard

The dashboard displays a summary of vital signs with a note: "[Signos vitales fuera de los rangos normales]". A "View Profile" button is present.

Emergencia

The page shows "Recent Emergencies" with details for "Emergency A" (Date: 2023-10-01) and "Emergency B" (Date: 2023-09-30). An "SOS" button is prominently displayed.

Signos Vitales

The page displays "Vital Signs History" with a "Heart Rate Over Time" chart and a "Blood Pressure Readings" bar chart.

Alteraciones

Alteration

Alteration (Date) - (Hour)
ASDASJHDJKASHDHASDJAHSKDHJKASDHJAHJSDF
JKAJHSDHJASHJKDJKASHDJKAHJKSDHJKASHJ
DAJHKS

Alteration (Date) - (Hour)
ASDASJHDJKASHDHASDJAHSKDHJKASDHJAHJSDF
JKAJHSDHJASHJKDJKASHDJKAHJKSDHJKASHJ
DAJHKS

Alteration (Date) - (Hour)
ASDASJHDJKASHDHASDJAHSKDHJKASDHJAHJSDF
JKAJHSDHJASHJKDJKASHDJKAHJKSDHJKASHJ
DAJHKS

Reportar una alteración

Log Out

© 2025 Health Management System Privacy Policy Terms of Service

Perfil

John Doe
Patient
Manage your personal health records.
Email corre@gaasa.com

Change Password

Edit Profile

Log Out

© 2025 Health Management System Privacy Policy Terms of Service

Application wireframes (mobile)**Register**

16:04

New Account

Full name
John Doe

Password

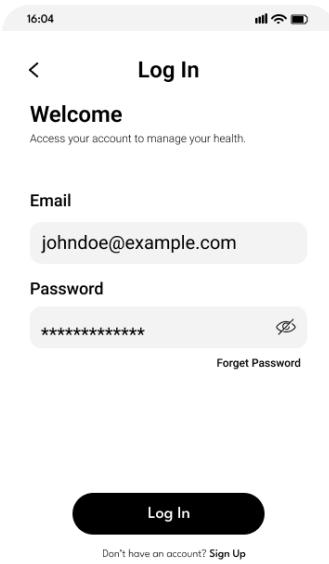
Email
johndoe@example.com

Phone Number
987654321

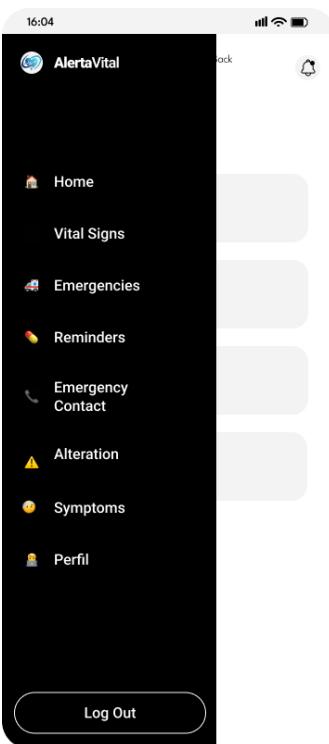
By continuing, you agree to [Terms of Use](#) and [Privacy Policy](#).

Sign Up

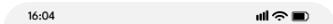
already have an account? [Log in](#)**Login**



Dashboard



Emergencia



Recent Emergencies

Check out your latest emergencies



Emergency A

Details about
Emergency A

Emergency B

Details about
Emergency B

Date: 2023-10-01

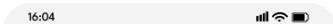
Date: 2023-09-30

Llama al 106



Location sent to emergency contact / [Emergency Contact]

Signos Vitales



Heart Rate

72 BPM

+2 BPM

Blood Pressure

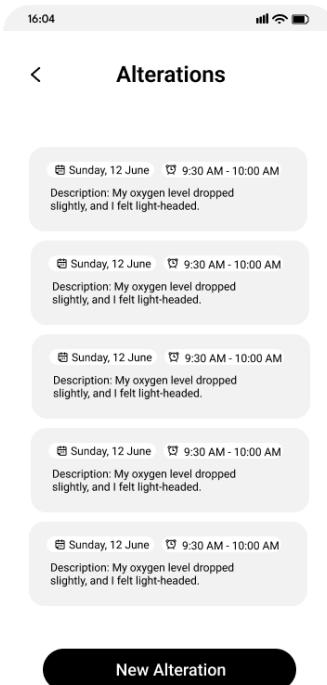
120/80 mmHg

Stable

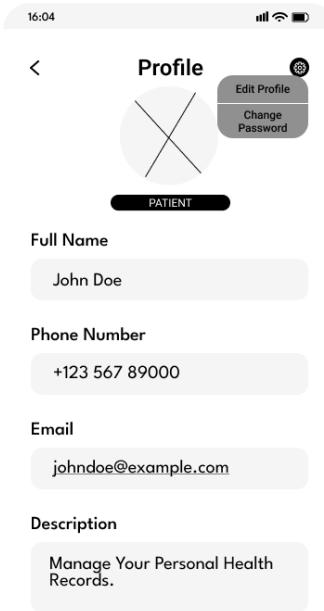
Oxygenation

72 SpO₂

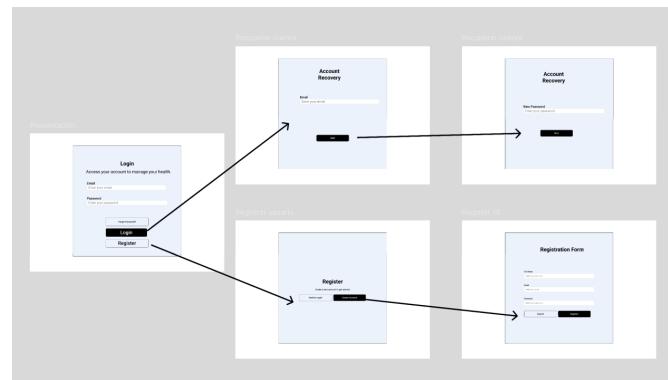
Alteraciones



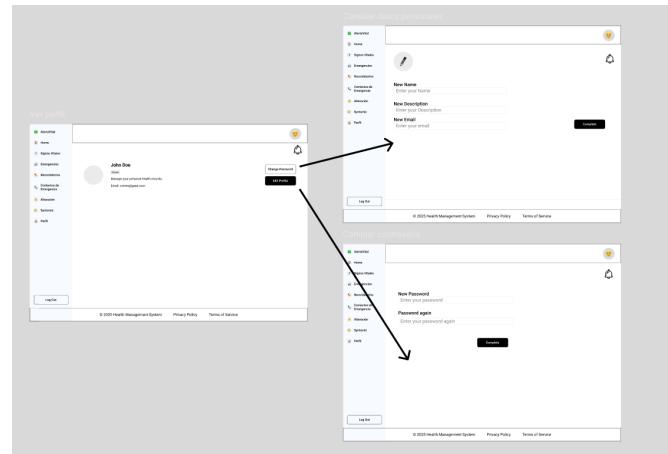
Perfil

**5.4.2. Applications Wireflow Diagrams.****Applications Wireflows Diagrams (Desktop)**

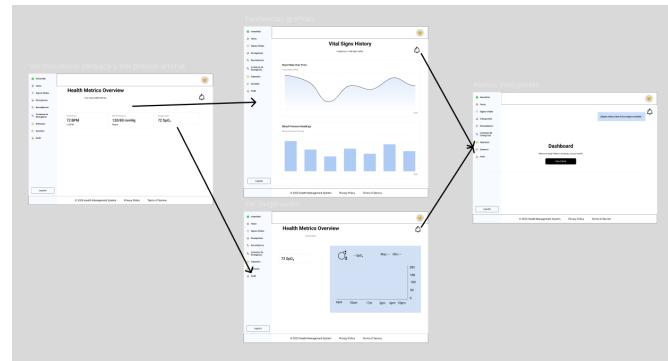
User Goal: Autenticación y gestión de cuenta



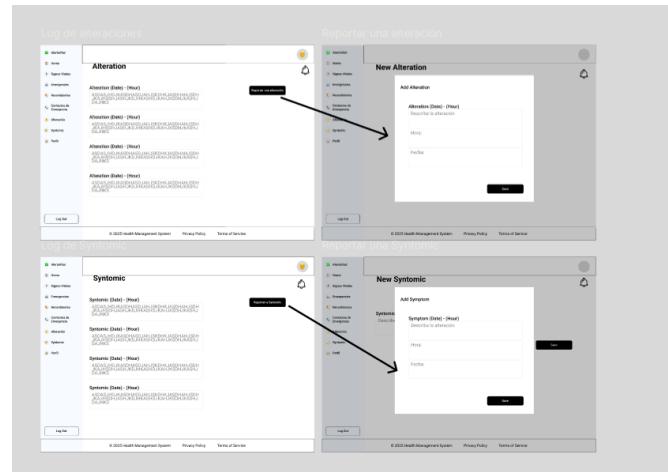
User Goal: Gestión de Perfil de Usuario



User Goal: Monitoreo de Signos Vitales



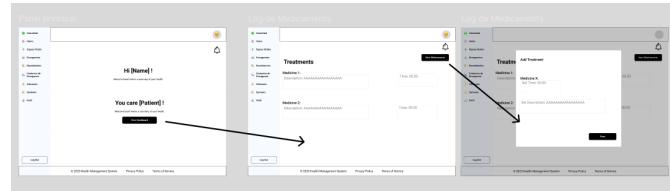
User Goal: Registro y Seguimiento de alteraciones



User Goal: Emergencias y Alerta



User Goal: Soporte al Tratamiento



Applications Wireflows Diagrams (Mobile)

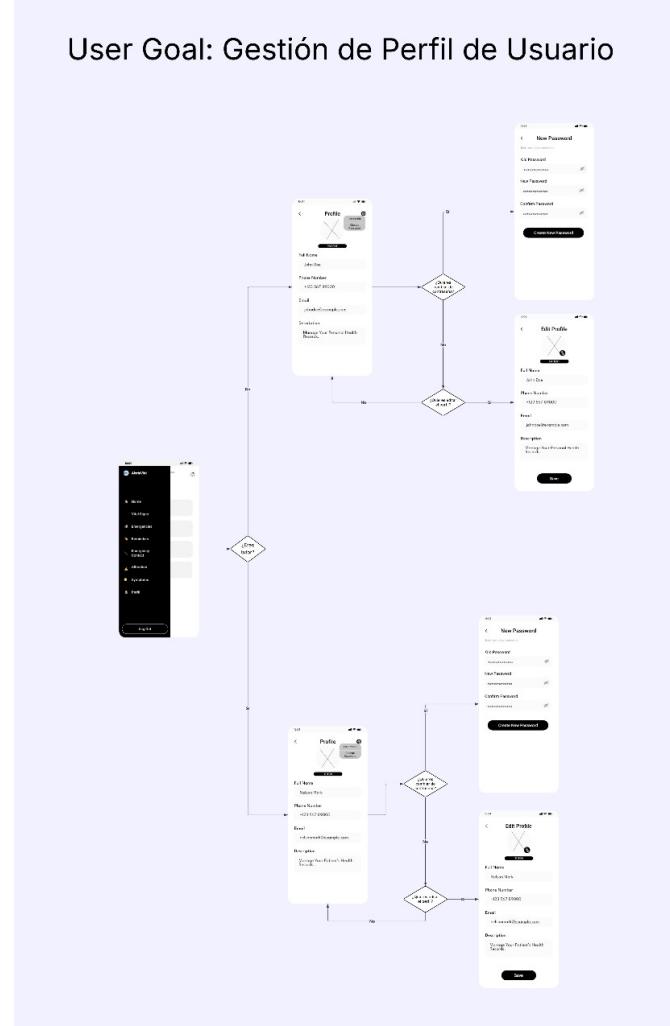
Link Lucid: https://lucid.app/lucidchart/a47baef7-d733-4cd7-baf0-e65d8a534db6/edit?viewport_loc=-11%2C-11%2C2217%2C1052%2C0_0&invitationId=inv_cd890741-d2b7-48f0-9b7c-56cc9bf788bf

User Goal: Autenticación y gestión de cuenta



User Goal: Gestión de Perfil de Usuario

User Goal: Gestión de Perfil de Usuario



User Goal: Monitoreo de Signos Vitales



User Goal: Registro y Seguimiento de alteraciones



User Goal: Emergencias y Alerta



User Goal: Soporte al Tratamiento



5.4.2. Applications Mock-ups.

Application Mock-ups (Desktop)

Register

Registration Form

Full Name
Enter your full name

Email
Enter your email

Password
Enter your password

Login

Login
Access your account to manage your health.

Email
Enter your email

Password
Enter your password

[Forgot Password?](#)

Login

Register

Dashboard

AlertaVital

- Home
- Signos Vitales
- Emergencias
- Recordatorios
- Contactos de Emergencia
- Alteración
- Sintomáticos
- Perfil

Dashboard

Welcome back! Here's a summary of your health.

[View Profile](#)

[Log Out](#)

© 2025 Health Management System Privacy Policy Terms of Service

Emergencia

Recent Emergencies

Check out your latest emergencies.

Emergency A	Emergency B
Details about Emergency A	Details about Emergency B
Date: 2023-10-01	Date: 2023-09-30

Llama al 106

Check out your latest emergencies.

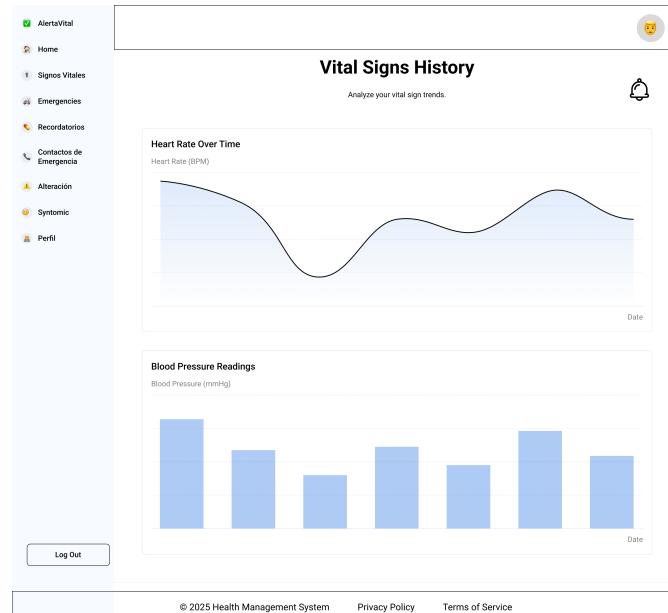
SOS

Ubicación enviada al contacto de emergencia / [Contacto de emergencia]

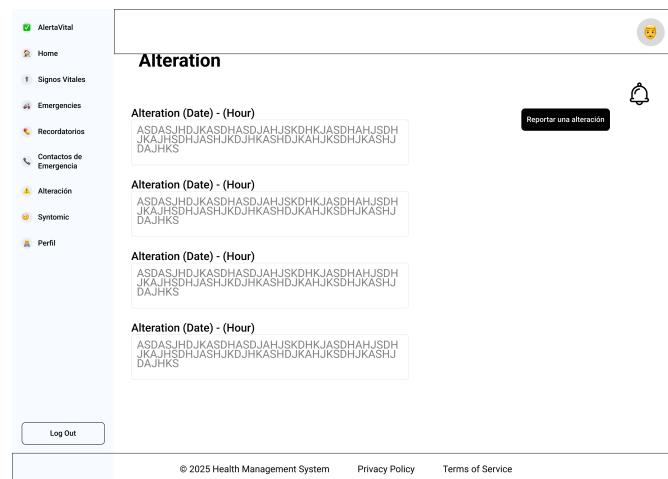
[Log Out](#)

© 2025 Health Management System Privacy Policy Terms of Service

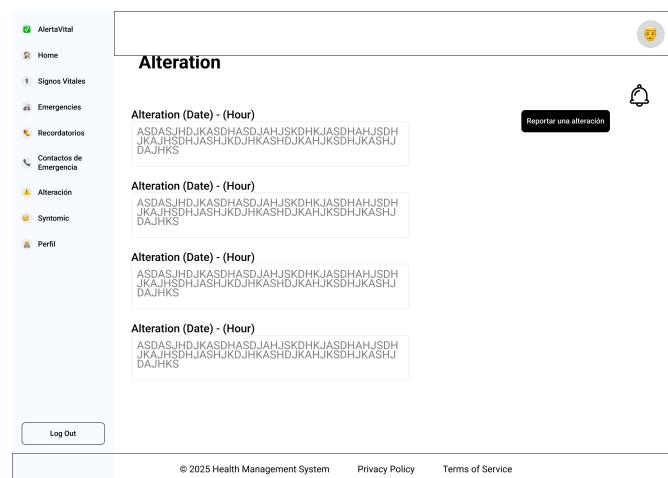
Signos Vitales



Alteraciones

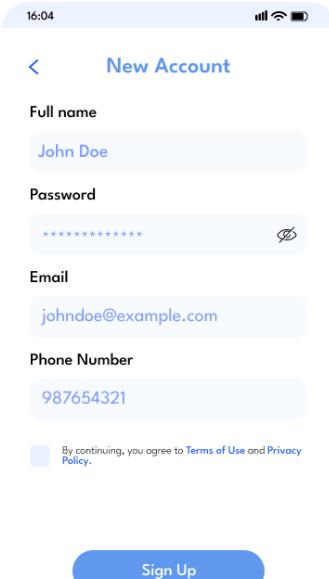


Perfil

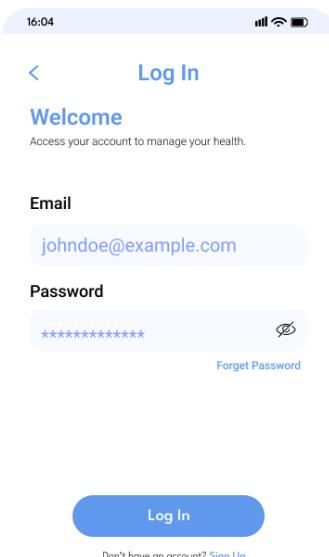


Application Mock-ups (mobile)

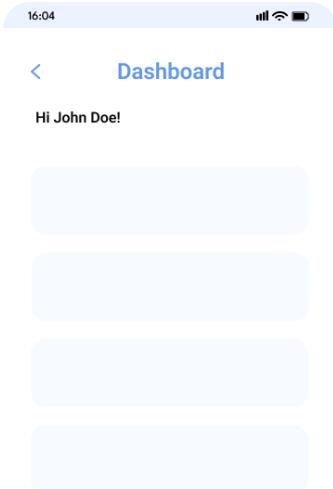
Register



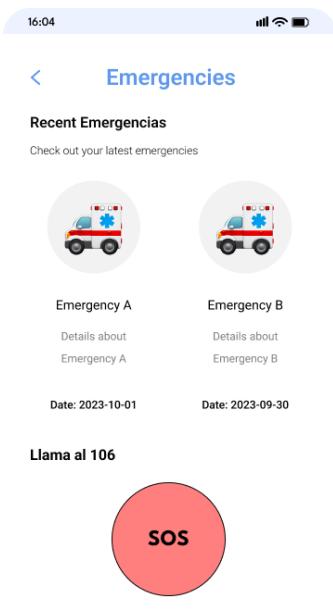
Login



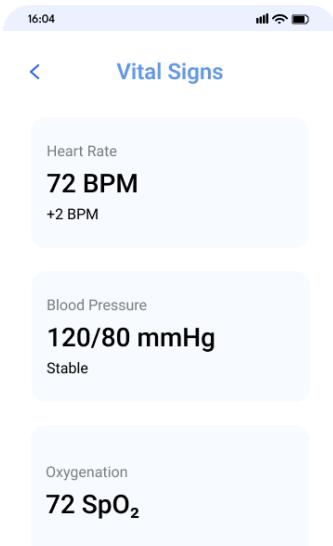
Dashboard



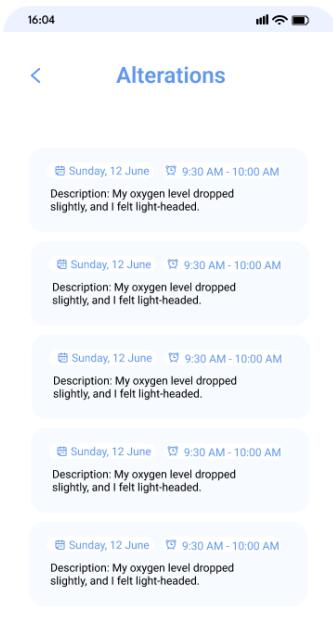
Emergencia



Signos Vitales



Alteraciones



New Alteration

Perfil

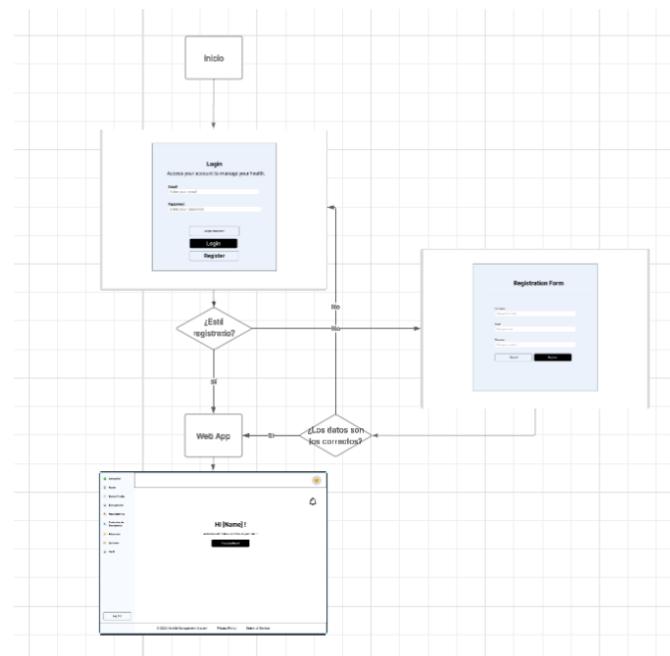


5.4.3. Applications User Flow Diagrams.

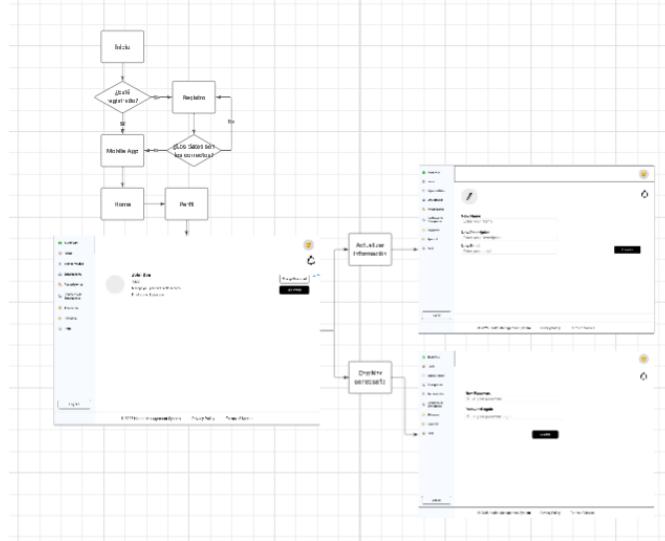
Applications User Flow Diagrams (Desktop)

Link Lucid: https://lucid.app/lucidchart/217716ff-f8b0-4cf2-a47f-ce803460c6f1/edit?invitationId=inv_ae9bfffbb-0ebf-41e1-be65-e88fde48c9a1&page=0_0#

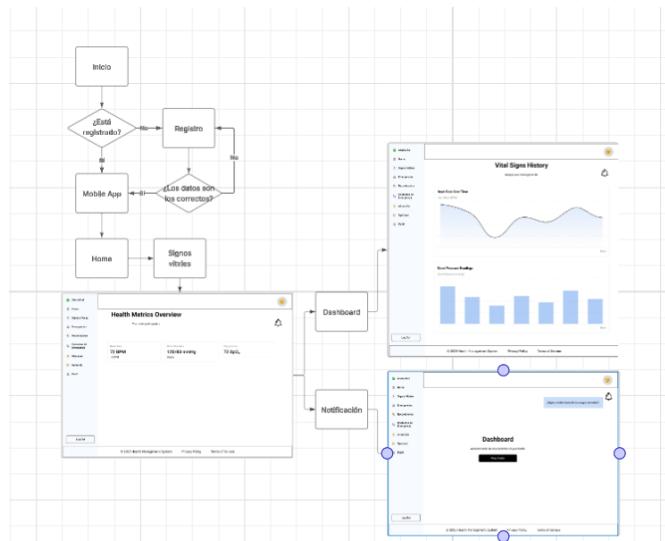
User Goal: Autenticación y gestión de cuenta



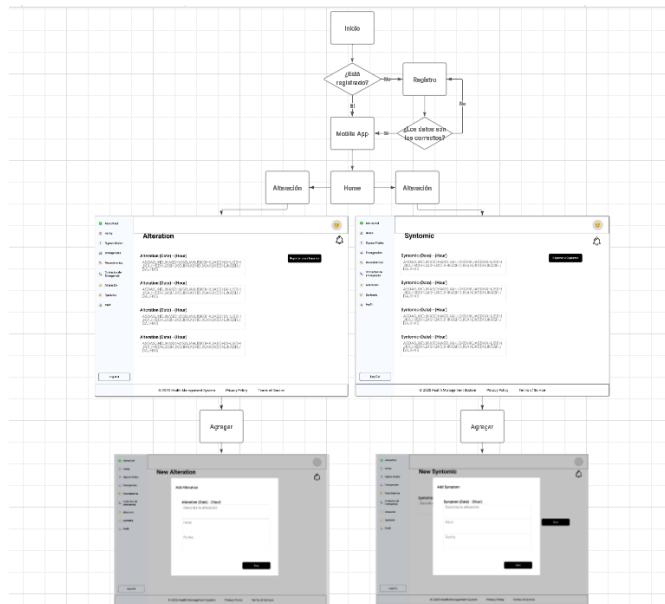
User Goal: Gestión de Perfil de Usuario



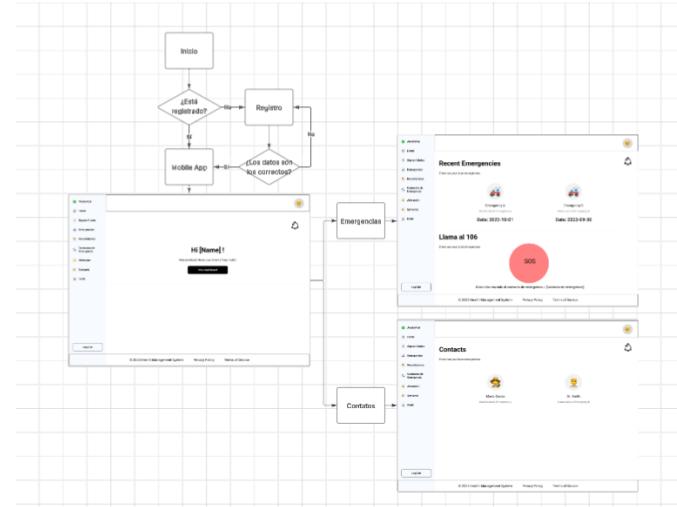
User Goal: Monitoreo de Signos Vitales



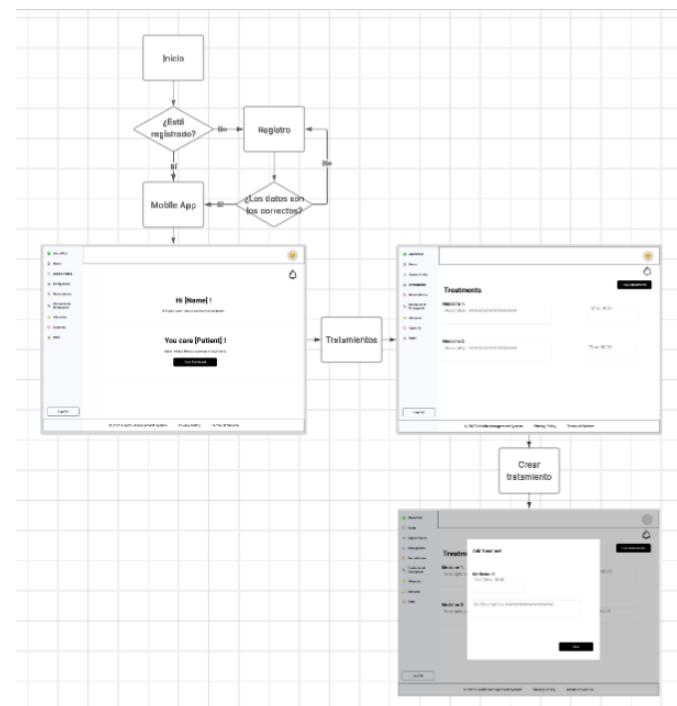
User Goal: Registro y Seguimiento de alteraciones



User Goal: Emergencias y Alerta



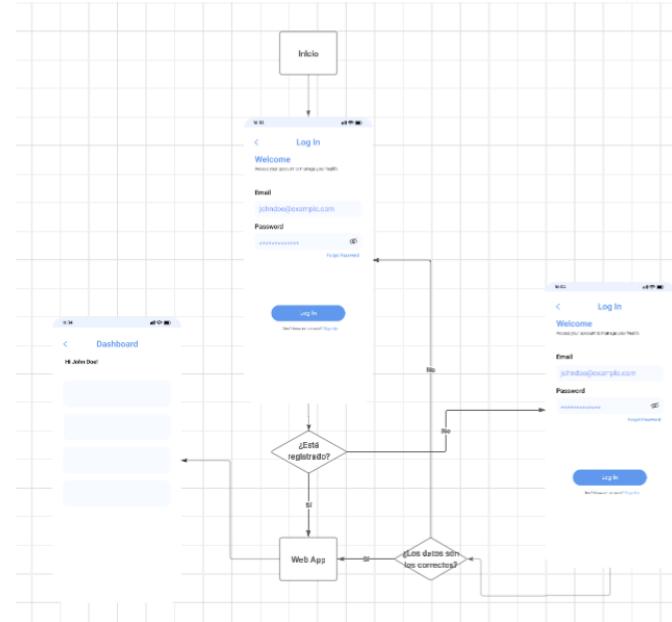
User Goal: Soporte al Tratamiento



Applications User Flow Diagrams (Mobile)

Link Lucid: [Link](#)

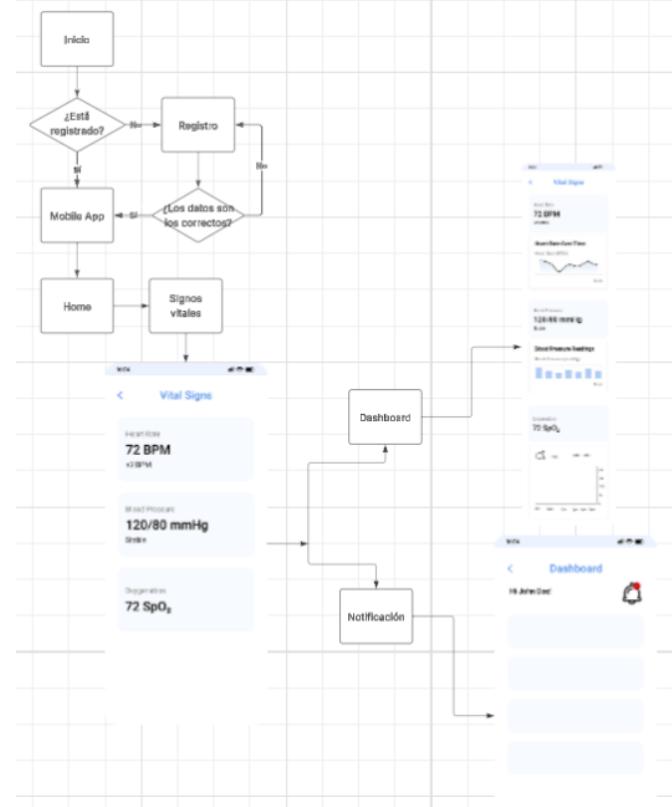
User Goal: Autenticación y gestión de cuenta



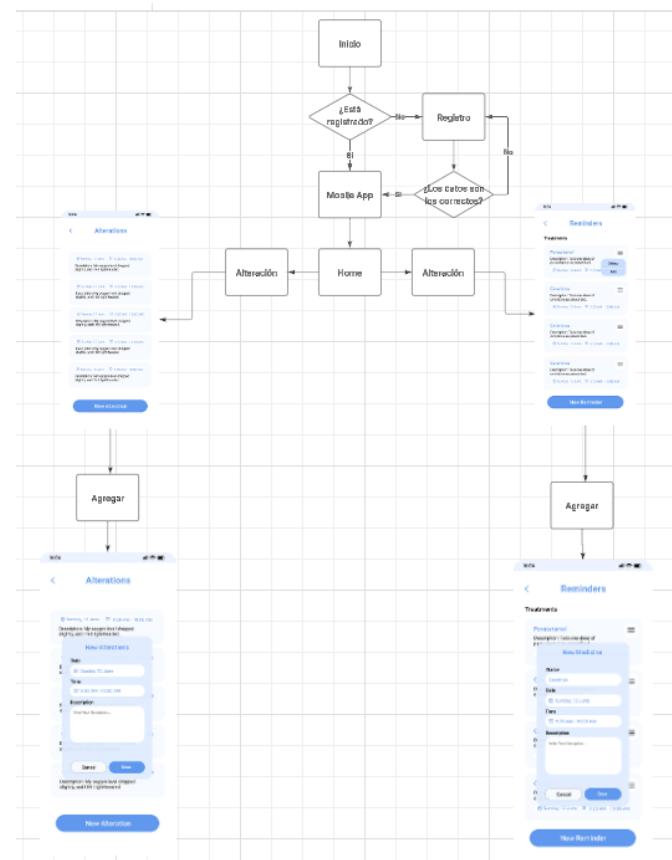
User Goal: Gestión de Perfil de Usuario



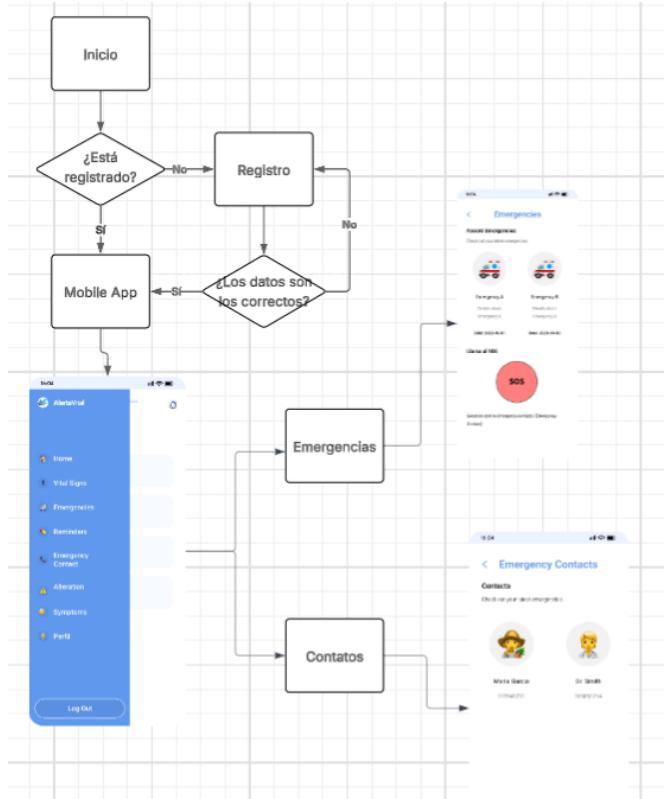
User Goal: Monitoreo de Signos Vitales



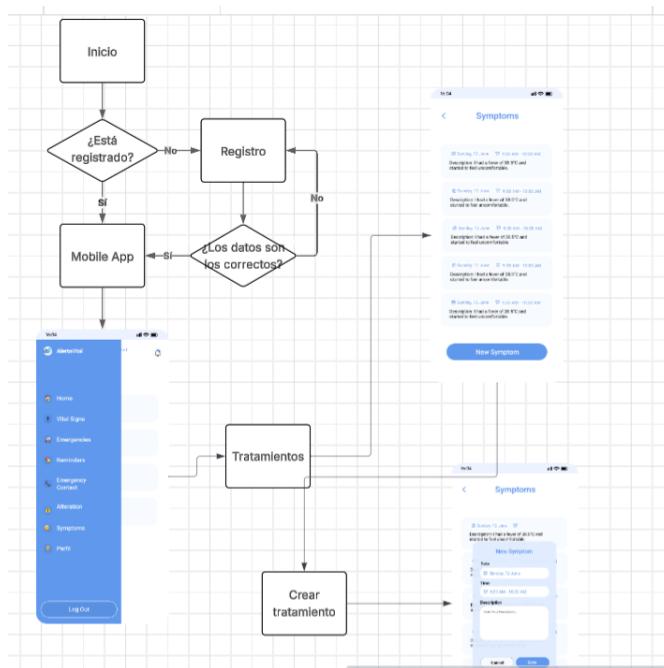
User Goal: Registro y Seguimiento de alteraciones



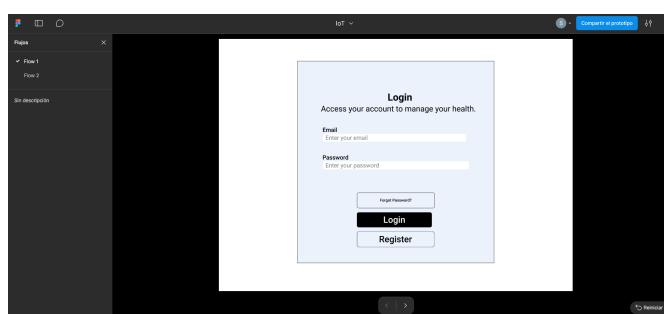
User Goal: Emergencias y Alerta



User Goal: Soporte al Tratamiento

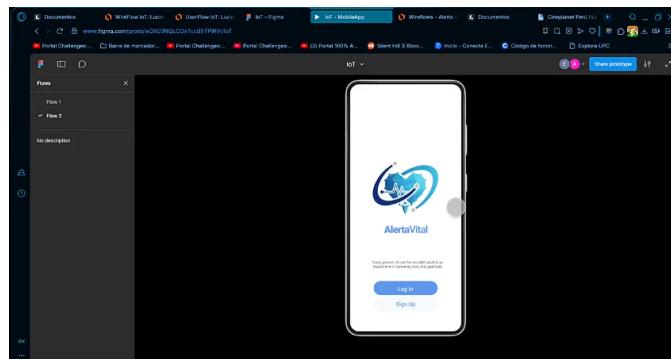


5.5. Applications Prototyping.



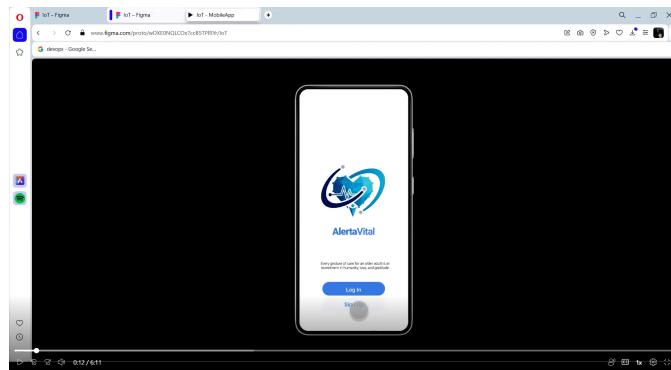
Link Prototipo: [Link](#)

Prototipo móvil del tutor



Link Prototipo: [Link](#)

Prototipo móvil del adulto mayor



Link Prototipo: [Link](#)

Capítulo VI: Product Implementation, Validation & Deployment

6.1. Software Configuration Management.

6.1.1. Software Development Environment Configuration.

Para la realización del producto, hemos usado las siguientes herramientas:

Requirements Management:

UxPressia: herramienta que se usó para desarrollar los distintos aspectos de la experiencia del cliente final. Gracias a esta herramienta se elaboraron los conceptos de Needfinding como User persona, Journey Map, Empathy Map e Impact Maps.

Ruta de referencia: <https://uxpressia.com>



Miro: Plataforma en línea que nos ayudó gracias a la creación de diagramas, flujos de trabajo y principalmente el EventStorming process.

Ruta de referencia: <https://miro.com/es/>



PRODUCT UX/UI DESIGN:

Figma: herramienta de diseño colaborativo que se usó para desarrollar la capa de presentación de la aplicación móvil, web y la Landing Page y prototipos interactivos, permitiendo simular y probar la navegación y las interacciones que deseamos brindar al usuario.

Ruta de referencia: <https://www.figma.com>



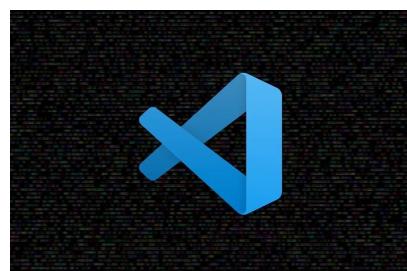
Overflow: Es un plugin de Figma que usamos para crear Wireflows y User Flows. Overflow nos permite combinar wireframes con flujos de usuario para visualizar cómo los usuarios interactúan con la interfaz. Además, con Overflow podemos determinar el recorrido completo del usuario.

Ruta de referencia: <https://www.figma.com/community/plugin/880782315986192393/overflow>

**Software Development**

Visual Studio Code: Es un editor de código fuente, gratuito y de código abierto desarrollado por Microsoft. Dicha herramienta nos permite escribir y editar el código HTML, CSS y JavaScript y la biblioteca "React" para elaborar la landing page y la aplicación web.

Ruta de referencia: <https://code.visualstudio.com>



Android Studio: Es un IDE oficial para el desarrollo de aplicaciones Android. Dicho software es gratuito y está basado en IntelliJ IDEA. Esta herramienta nos permite desarrollar la aplicación móvil planteada.

Ruta de referencia: <https://developer.android.com/>



Bootstrap 5: Es una biblioteca de componentes y un sistema de grillas que facilita el diseño web responsive. Utilizamos Bootstrap 5 en esta landing page para agilizar la creación de un diseño adaptable y aprovechar sus componentes preconstruidos, como botones, formularios y barras de navegación.

Ruta de referencia: <https://getbootstrap.com/>



Software Deployment

Git: Es una herramienta de control de versiones que nos permite colaborar de manera eficiente y mantener un historial detallado de las modificaciones en el código.

Ruta de referencia: <https://git-scm.com/>



Software Documentation and Project Management GitHub: Como plataforma de alojamiento, nos facilita la colaboración en equipo, la gestión de ramas y la integración continua, optimizando el flujo de trabajo de desarrollo. Además, usamos github para recurrir a su funcionalidad Github Pages y desplegar la Landing Page de manera sencilla.

Ruta de referencia: <https://github.com/>



6.1.2. Source Code Management.

La gestión del proyecto de la aplicación móvil, landing page y aplicación web, siguen los principios del modelo "Git Branch Model" en la cual se define como una estrategia para administrar y organizar un trabajo en un repositorio de Github mediante la creación de ramas con distintas funcionalidades. Uno de los modelos que usaremos es el "Git Flow", la cual define como crear ramas para mantener el control del ciclo de vida del desarrollo del software.

Repositorio de Landing Page: <https://github.com/TF-SolucionesIoT/LandingPage>

Repositorio de Aplicación Web:

Estructura de ramas:

1. Rama principal: esta rama es la principal del proyecto, contendrá versiones finales del desarrollo. Solo se permitirán cambios si ya han pasado por una previa revisión en otras ramas de testing.
2. Rama de Desarrollo: esta rama es en donde se lleva a cabo los avances del proyecto en equipo. Es donde se integran las nuevas características antes de fusionarlas al main.
3. Rama de Características: Se crean a partir de develop para desarrollar nuevas características o funcionalidades. Una vez completadas, se fusionan de nuevo a develop. Cabe destacar que estas ramas son temporales, ya que su única funcionalidad es desarrollar una funcionalidad y fusionarla con la rama de desarrollo si es que se completó el requerimiento. Sin embargo en el proyecto de la Landing Page, no se eliminarán para fines de descripción.
4. Rama de Corrección: Estas ramas se crean para abarcar un error específico y poder solucionarlo para fusionarlo a la rama de desarrollo.
5. Rama de Lanzamiento: Se crean cuando en la rama de desarrollo está listo para una nueva versión de producción. Permiten hacer pruebas y correcciones menores antes de fusionar a main.

Versionamiento Semántico: Para nombrar las Releases, se utilizará el sistema de versionamiento semántico (Semantic Versioning 2.0.0).

Convenciones de Commits: Para los mensajes de los commits realizados, se usa la especificación Conventional Commits, La estructura a seguir será la siguiente:

"(tipo):(titulo)" y "(descripcion)"

6.1.3. Source Code Style Guide & Conventions.

Para el desarrollo de nuestro producto, hemos recurrido a estos lenguajes para desarrollar la solución:

HTML

Para el desarrollo en HTML, se han seguido las Convenciones de Codificación para HTML. Los puntos más importantes aplicados son:

- **Uso de etiquetas semánticas:** Se han utilizado etiquetas como header, nav, main, footer para mejorar la accesibilidad y la comprensión del documento.
- **Indentación y estructura:** A pesar de que HTML permite combinar mayúsculas y minúsculas en los nombres de los elementos y atributos, se limitará al uso de minúsculas para mantener el orden y garantizar la legibilidad, usaremos "kebab-case".

CSS En el CSS, se ha implementado la convención BEM y las guías de estilo recomendadas para mantener el código claro y organizado. Aquí algunos detalles específicos de tu código CSS:

- **Uso de nombres de clases descriptivos y en inglés:** Las clases siguen un esquema claro, usando "kebab-case", lo que facilita la identificación del propósito de cada clase.
- **Transiciones y efectos en botones:** Se implementaron transiciones suaves para mejorar la experiencia de usuario cuando interactúa con botones.
- **Uso de fuentes personalizadas:** Se han utilizado varias fuentes personalizadas cargadas mediante @font-face, lo que sigue las mejores prácticas de personalización tipográfica en CSS.

JavaScript En el proyecto se ha utilizado Bootstrap para manejar la interactividad y los componentes dinámicos de la interfaz. Los archivos JavaScript incluidos son versiones oficiales del framework, lo que permite aprovechar funcionalidades avanzadas sin necesidad de escribir código JavaScript desde cero.

Bootstrap Para Bootstrap, se han seguido las mejores prácticas, como utilizar clases predefinidas para ahorrar tiempo y asegurar que el diseño sea responsive. Además, se ha personalizado el diseño sin modificar directamente las clases de Bootstrap, lo que asegura compatibilidad y facilidad de actualización.

Java (Springboot)

Para la aplicación API Restful, usamos la convención estándar "Google Java Style Guide".

- **Definición de nombres de clases:** Para la creación de las clases y métodos de estos en Java hemos usado PascalCase y para la instancia de atributos de clases hemos usado lowerCamelCase.
- **Organización y estructura del proyecto:** Para un mejor desarrollo de la aplicación, hemos usado "Clean Architecture" y principios de Domain Driven Design, separando por Bounded context y dentro de estas por capas como Aplicación, Infraestructura, Dominio e Interfaces.

Kotlin (Android Studio)

Para el desarrollo de la aplicación móvil, usamos convenciones de "Google Kotlin Style Guide" para organizar de una mejor manera el proyecto.

- **Definición de nombres de clases:** Para la creación de las clases y métodos de estos en Kotlin hemos usado PascalCase y para la instancia de atributos de clases hemos usado lowerCamelCase.
- **Organización y estructura del proyecto:** Para un mejor desarrollo de la aplicación, hemos usado "Clean Architecture" y principios de Domain Driven Design, separando por Bounded context y dentro de estas por capas como Repositorios, UI, Casos de Uso y Dominios.

6.1.4. Software Deployment Configuration.

Landing Page:

En el caso de la Landing Page, desplegaremos la página en Github Pages que es la funcionalidad de Github para desplegar páginas estáticas.

Para configurar GitHub Pages una vez el código esté en el repositorio, sigue estos pasos para habilitar GitHub Pages:

Ve a tu repositorio en GitHub. Haz clic en la pestaña Settings (Configuración). Desplázate hacia abajo hasta la sección Pages en el menú lateral izquierdo.

The screenshot shows the GitHub repository settings page for 'TF-SolucionesIoT'. The left sidebar has 'Pages' selected under the 'General' category. The main area is titled 'GitHub Pages' and contains the following information:

- Access:** Shows 'None' selected.
- Collaborators and teams:** Shows 'None' selected.
- Moderation options:** Shows 'None' selected.
- Code and automation:**
 - Branches:** Shows 'None' selected.
 - Tags:** Shows 'None' selected.
 - Rules:** Shows 'None' selected.
 - Actions:** Shows 'None' selected.
 - Models:** Shows 'None' selected.
 - Webhooks:** Shows 'None' selected.
 - Copilot:** Shows 'None' selected.
 - Environments:** Shows 'None' selected.
- Source:** A dropdown menu set to 'Deploy from a branch'.
- Build and deployment:** A dropdown menu set to 'None'.
- Visibility:** Set to 'GitHub Enterprise'.
- Start free for 30 days:** A button to begin a free trial.

A note in the center states: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.'

En la sección Source, selecciona la rama de la que quieras desplegar (normalmente main).

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more about configuring the publishing source for your site.](#)

main / (root) Save

Visibility GitHub Enterprise

With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. You can use privately published sites to share your internal documentation or knowledge base with members of your enterprise. You can try GitHub Enterprise risk-free for 30 days. [Learn more about the visibility of your GitHub Pages site.](#)

Preview Start free for 30 days

Pages

Custom properties

Security

Advanced Security Deploy keys Secrets and variables

GitHub ahora generará una URL para tu landing page, que estará disponible en algo como: <https://usuario.github.io/Example>

TF-Solucionesiot / LandingPage

Type [] to search

Code Issues Pull requests Actions Projects Security Insights Settings

GitHub Pages source saved.

General

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the [main](#) branch. Learn more about [configuring the publishing source for your site.](#)

main / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

Custom domain

Custom domains allow you to serve your site from a domain other than [tf-solucionesiot.github.io](#). Learn more about [configuring custom domains.](#)

Save Remove

Si necesitas hacer actualizaciones, cada vez que hagas un cambio en la rama que seleccionaste en la configuración, GitHub Pages actualizará automáticamente la página en cuestión de minutos.

Consideraciones: GitHub Pages solo soporta sitios estáticos (HTML, CSS, JavaScript). Si tu landing page necesita procesamiento en el servidor, necesitarás otro servicio de hospedaje. Este también necesita de un archivo "index.html" para poder desplegarse, de lo contrario no se podrá.

Web Application:

Para el despliegue de la primera versión de la aplicación web, primero debemos buildear nuestro proyecto.

comando de build: npm run build **comando para verificar ejecución de la build:** npm start

Luego de verificar que la build funciona correctamente, debemos de instalar una dependencia solo para desarrollo, en este caso de Vercel.

comando Vercel CLI: npm i -g vercel

Iniciamos sesión con: vercel login

Finalmente ejecutamos el comando "vercel" y seguimos los pasos que se nos indican en la terminal:

```
Terminal Local + ↗
PS C:\Users\HP\Desktop\Repository\FrontEnd_Repos\Front-End-IOT> vercel
Vercel CLI 48.2.9
? Set up and deploy “~\Desktop\Repository\FrontEnd_Repos\Front-End-IOT”? yes
? Which scope should contain your project? xYisus' projects
? Link to existing project? no
? What's your project's name? alerta-vital
? In which directory is your code located? ./
Local settings detected in vercel.json:
Auto-detected Project Settings (Next.js):
- Build Command: next build
- Development Command: next dev --port $PORT
- Install Command: `yarn install`, `pnpm install`, `npm install`, or `bun install`
- Output Directory: Next.js default
? Want to modify these settings? no
? Do you want to change additional project settings? no
🔗 Linked to xysisus-projects/alerta-vital (created .vercel)
? Detected a repository. Connect it to this project? no
🔍 Inspect: https://vercel.com/xyisisus-projects/alerta-vital/85tcWq1EDTPmTRwxqmdZDBpSUDhr [4s]
✅ Production: https://alerta-vital-qpfj8gxeh-xyisisus-projects.vercel.app [4s]
📝 Deployed to production. Run `vercel --prod` to overwrite later (https://vercel.link/2F).
💡 To change the domain or build command, go to https://vercel.com/xyisisus-projects/alerta-vital/settings
PS C:\Users\HP\Desktop\Repository\FrontEnd_Repos\Front-End-IOT>
```

Al final de los pasos, podemos ver que nos brinda los links para gestionar el despliegue y verlo en la web.

6.2. Landing Page, Services & Applications Implementation.

6.2.1. Sprint 1

6.2.1.1. Sprint Planning 1.

En la siguiente sección, se presentará el sprint #1 con la primera versión de la landing page y la primera versión de la aplicación web.

Sprint #	Sprint 1
Sprint	
Planning	
Background	

Sprint #	Sprint 1
Date	2025-09-18
Time	04:30 PM
Location	Llamada por la plataforma "Discord"
Prepared By	Uribe Quispe, Jesús Guillermo
Attendees (to planning meeting)	Uribe Quispe Jesús Guillermo / Bohorquez Lerzundi Gerardo Sebastián / Huanaco Huayta Elizabeth Lucero / Huapaya Cuevas Anthony / Sebastián Silva Tirado
Sprint 1 – 1	
Review	Desarrollo de UX/UI DESIGN, landing page y aplicación web.
Summary	
Sprint 1 – 1	
Retrospective	Como es el primer sprint, no hay una retrospectiva anterior.
Summary	
Sprint Goal & User Stories	
Sprint #1 Goal	Nuestro enfoque está en presentar AlertaVital a los potenciales usuarios mediante una landing page clara y atractiva, y en entregar la primera versión funcional de la aplicación web. Creemos que esto generará mayor visibilidad y compromiso de los usuarios iniciales y posibles clientes. Esto se confirmará cuando los visitantes puedan acceder a la landing page, comprender la propuesta de valor del producto y utilizar con éxito las funciones iniciales de la aplicación web.
Sprint #1 Velocity	68h
Sum of Story Points	17

6.2.1.2. Aspect Leaders and Collaborators.

Team Member (Last Name, First Name)	GitHub Username	Landing Page	Web Application
Huapaya, Anthony	AnthonyHuapaya	C	C
Huanaco, Elizabeth	lucerohh	C	C
Silva, Sebastián	SebasSilvaT	C	L
Bohorquez, Sebastián	sebasbl562	C	C
Uribe, Jesús	JesusU27	L	C

6.2.1.3. Sprint Backlog 1.

Para el Sprint #1 nos trazamos como objetivo desarrollar y desplegar una landing page funcional, optimizada para móviles y preparada para captar leads, lista para producción en GitHub Pages con un diseño responsive y tiempos de carga rápidos.

#	Sprint	Sprint 1		Descripción	Estimación (Horas)	Story Points	Asignado a	Estado
User Story	Work Item / Task	Id	Title					
US21	Ver características	WT01	Ver características	Como visitante, quiero conocer las principales características de la aplicación para entender qué ofrece.	4h	1	Jesús	Done
US22	Ver beneficios	WT02	Ver beneficios	Como visitante, quiero ver los beneficios de usar la aplicación para motivarme a registrarme.	4h	1	Jesús	Done
US23	Comprender la problemática	WT03	Comprender la problemática	Como visitante, quiero entender la problemática de salud que aborda la aplicación para sentirme identificado con su propósito.	4h	1	Jesús	Done
US24	Conocer acerca del proyecto	WT04	Conocer acerca del proyecto	Como visitante, quiero leer información sobre la misión, visión y el equipo detrás de la aplicación.	4h	1	Jesús	Done
US25	Ver testimonios	WT05	Ver testimonios	Como visitante, quiero leer opiniones de otros usuarios o cuidadores para generar confianza.	8h	2	Jesús	Done
US26	Enviar mensaje de contacto	WT06	Enviar mensaje de contacto	Como visitante, quiero enviar un mensaje a través del formulario de contacto para resolver mis dudas.	12h	3	Jesús	Done

#	Sprint	Sprint 1							
User Story		Work Item / Task							
Id	Title	Id	Title	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado
US27	Descargar la aplicación	WT07	Descargar la aplicación	Como visitante, quiero descargar la aplicación desde la landing page para comenzar a usarla.		8h	2	Jesús	Done
US19	Recordatorios de medicamentos	WT08	Recordatorios de medicamentos	Programar notificaciones para recordar la toma de medicamentos.		12h	3	Sebastián Bohorquez	In Process
US15	Registro de síntomas	WT09	Registro de síntomas	Registrar manualmente síntomas adicionales y consultarlos en un histórico.		12h	3	Sebastian Bohorquez	In Process
US13	Reportar una alteración	WT10	Reportar una alteración	Reportar manualmente una alteración en la salud.		8h	2	Elizabeth	In Process
US14	Log de alteraciones	WT11	Log de alteraciones	Consultar el histórico de alteraciones registradas.		8h	2	Elizabeth	In Process
US05	Cambiar datos personales	WT12	Cambiar datos personales	Actualizar datos personales en el perfil.		8h	2	Anthony	In Process
US06	Cambiar contraseña	WT13	Cambiar contraseña	Actualizar la contraseña desde el perfil.		8h	2	Sebastian Silva	In Process
US07	Cambiar correo	WT14	Cambiar correo	Actualizar el correo electrónico para recibir notificaciones correctas.		8h	2	Sebastian Bohorquez	In Process
US08	Ver frecuencia cardiaca	WT15	Ver frecuencia cardiaca	Visualizar la frecuencia cardíaca en tiempo real.		8h	2	Anthony Huapaya	In Process
US09	Ver presión arterial	WT16	Ver presión arterial	Consultar los valores de presión arterial en el panel de signos vitales.		8h	2	Sebastián Silva	In Process
US10	Ver oxigenación	WT17	Ver oxigenación	Consultar el nivel de oxigenación (SpO_2) en tiempo real.		8h	2	Anthony Huapaya	In Process
US02	Iniciar sesión	WT18	Iniciar sesión	Acceder al perfil y datos de salud con usuario y contraseña.		4h	1	Sebastián Silva	In Process
US04	Ver perfil	WT19	Ver perfil	Visualizar información personal registrada en la aplicación.		4h	1	Elizabeth	In Process

6.2.1.4. Development Evidence for Sprint Review.

LANDING PAGE:

Debido a la refactorización de un código, además de la realización de solo una persona por elección y consentimiento de todo el grupo, solo se presenta el commit de subida al repositorio principal de la organización.

Repository	Branch	Commit ID	Commit Message	Commit Body	Committed on (Date)
https://github.com/TF-SolucionesIoT/LandingPage	main	47fea191f279d7aeac5bd1cf043b4b887c3b48e2	first commit	first commit	2/10/2025

WEB APPLICATION:

Repository	Branch	Commit ID	Commit Message	Commit Body	Committed on (Date)
https://github.com/TF-SolucionesIoT/Front-End-IOT	main	79d7aea8	first commit	first commit	2/10/2025
https://github.com/TF-SolucionesIoT/Front-End-IOT	main	9992216	Thanks, now that you add a chart/dashboard for "SpO ₂ "	2025-10-09	

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
https://github.com/TF-SolucionesloT/Front-End-IOT	main	329ec31	In the "Vitals" section, when clicking on "Heart Rate", "SpO ₂ ", or "Blood Pressure", it should show the current value and history.	In the "Vitals" section, when clicking on "Heart Rate", "SpO ₂ ", or "Blood Pressure", it should show the current value and history.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	a341a86	In the "Treatments" section, add a button labeled "Add Treatment" that adds a new treatment to the list.	In the "Treatments" section, add a button labeled "Add Treatment" that adds a new treatment to the list.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	8d09a7b	Add a new section called "Syntomic" to display a history of symptoms with their evolution over time.	Add a new section called "Syntomic" to display a history of symptoms with their evolution over time.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	7c0093a	Add a new section called "Alteration" to display a history of alteration.	Add a new section called "Alteration" to display a history of alteration.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	3a7dc50	I see that the background is still: #0F1729, I would like it to be changed.	I see that the background is still: #0F1729, I would like it to be changed.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	403dc2a	I would like the sidebar to have this color: #F7FAFE and the background white.	I would like the sidebar to have this color: #F7FAFE and the background white.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	96b2142	Configure the sidebar with the following sections and behaviors:	Configure the sidebar with the following sections and behaviors:	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	f4c2165	Well the program shows: This page could not be found. Try to find the	Well the program shows: This page could not be found. Try to find the	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	b6fa798	First, fix the errors.	First, fix the errors.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	4bac8bf	Initial prototype.	Initial prototype.	2025-10-09
https://github.com/TF-SolucionesloT/Front-End-IOT	main	ecacaab	Initialized workspace with Firebase Studio.	Initialized workspace with Firebase Studio.	2025-10-06

6.2.1.5. Testing Suite Evidence for Sprint Review.

Para este sprint, no nos centramos en elaborar los servicios web, por lo tanto, no hay pruebas unitarias, de integración y de aceptación.

6.2.1.6. Execution Evidence for Sprint Review.

A continuación, se muestran las evidencias de ejecución de la landing page y de la primera versión de la aplicación web.

Landing Page:

The screenshot shows the AlertaVital landing page. At the top, there is a navigation bar with the logo 'AlertaVital' and links for 'Características', 'Beneficios', 'Problemática', 'Acerca de', 'Testimonios', 'Contacto', and a green 'Descargar' button. Below the navigation, there are two main sections: 'Para Cuidadores' (For Caregivers) and 'Para Personas Vulnerables' (For Vulnerable People). Each section has an icon of a person and a brief description. The 'Para Cuidadores' section says: 'alertas inmediatas desde la aplicación móvil, pudiendo supervisar el estado de la persona en todo momento.' The 'Para Personas Vulnerables' section says: 'confianza a quienes la usan, al monitorear pulsos y detectar caídas en tiempo real'.

Antecedentes y problemática

¿Cuáles son las causas?

El envejecimiento poblacional es un fenómeno global y particularmente relevante en Latinoamérica. En Perú, según el INEI (2022), la población de adultos mayores supera el 13% del total, con proyecciones de crecimiento constante en las

Desafíos en la supervisión de personas vulnerables

La población adulta mayor enfrenta riesgos significativos relacionados con el deterioro de la salud y la vulnerabilidad a caídas. Actualmente, el monitoreo de sus signos vitales suele

 **AlertaVital**

[Características](#) [Beneficios](#) [Problemática](#) [Acerca de](#) [Testimonios](#) [Contacto](#) [Descargar](#)

About The Product

 AlertaVital es una startup tecnológica dedicada a desarrollar soluciones innovadoras basadas en Internet de las Cosas (IoT) para mejorar la calidad de vida de las personas mayores y sus familias.

Conoce más sobre nuestro producto:

About The Team

 Somos un equipo de innovadores apasionados por la salud y la tecnología. Nuestra misión es proporcionar herramientas accesibles y confiables que permitan el monitoreo en tiempo real de signos vitales y la

Conoce a nuestro equipo:

 **AlertaVital**

[Características](#) [Beneficios](#) [Problemática](#) [Acerca de](#) [Testimonios](#) [Contacto](#) [Descargar](#)

Lo que dicen nuestros usuarios



"Desde que uso la pulsera me siento mucho más segura. Ya no tengo miedo de salir sola a caminar, porque sé que si algo me pasa, mi familia será avisada de inmediato."

Daymo Roque, Adulto Mayor



The screenshot shows a landing page for downloading the AlertaVital app. At the top, there's a navigation bar with links for 'Características', 'Beneficios', 'Problemática', 'Acerca de', 'Testimonios', 'Contacto', and a green 'Descargar' button. Below the navigation is a large blue callout box containing the text: 'Lleva el control y gestión de tu salud física o a la de un ser querido donde vayas'. Underneath this, it says: 'Descarga nuestra aplicación y accede a todas las funcionalidades de AlertaVital desde tu dispositivo móvil. Disponible para iOS y Android.' It features two download links: 'Descargar en App Store' with an Apple icon and 'Disponible en Google Play' with a Google Play icon. A small image of a smartphone is at the bottom.

Link de Video: https://upcedupe-my.sharepoint.com/:g/personal/u202221876_upc_edu_pe/ES14VABBaE1HttolgWJvG4IBV8ZsJiyVIWeUVjRBrGbOdA?e=TTZLFc

Web Application:

Iniciar Sesión:

The screenshot shows the login page for the AlertaVital web application. It features a large blue header with a checkmark icon and the text 'Welcome to AlertaVital'. Below this, there's a field labeled 'Email' with the placeholder 'm@example.com'. To the right of the email field is a link 'Forgot your password?'. Below the email field is a 'Password' field with a corresponding 'Forgot your password?' link. A large blue 'Login' button is centered below the password field. At the bottom of the form, there's a 'Login with Google' button and a link 'Don't have an account? [Sign up](#)'.

Perfil:

AlertaVital

- Home
- Vitals
- Emergencies
- Treatments
- Contacts
- Alterations
- Symptoms
- Settings**

Profile Settings

Manage your account details.



Jane Doe
 This is your public display name.
 jane.doe@example.com

Name

Email

[Edit Profile](#)
[Change Password](#)

Emergencias:

AlertaVital

- Home
- Vitals
- Emergencies**
- Treatments
- Contacts
- Alterations
- Symptoms
- Settings

Emergencies

Emergency History

Recent emergency events and dispatches.

Event	Description	Date
Ambulance	Dispatched for high heart rate alert.	July 30, 2024
Ambulance	Dispatched after fall detected.	July 15, 2024

Emergency

Press for immediate assistance



Tratamientos:

AlertaVital

- Home
- Vitals
- Emergencies
- Treatments**
- Contacts
- Alterations
- Symptoms
- Settings

Treatments

Manage your medication schedule.

[Add Treatment](#)

Medication Reminders

Your daily medication schedule.

Medication	Dose	Time	Status	Action
Lisinopril	10mg	08:00 AM	Taken	...
Metformin	500mg	08:00 AM	Taken	...
Aspirin	81mg	01:00 PM	Upcoming	...
Atorvastatin	20mg	08:00 PM	Upcoming	...

Alteraciones:

The screenshot shows the 'Alterations' section of the AlertaVital app. On the left is a vertical navigation bar with icons for Home, Vitals, Emergencies, Treatments, Contacts, Alterations (which is selected and highlighted in blue), Symptoms, and Settings. The main content area has a header 'Alterations' with a subtitle 'A history of reported alterations.' and a 'Add Alteration' button. Below this is a table titled 'Alteration History' with columns for #, Date, Time, Description, and Actions. Two entries are listed:

#	Date	Time	Description	Actions
1	2024-07-31	10:30 AM	Feeling dizzy after medication.	...
2	2024-07-30	03:15 PM	Experienced a slight headache.	...

Añadir alteraciones

The screenshot shows the 'Add Alteration' modal window overlaid on the main 'Alterations' screen. The modal has a title 'Add Alteration' and a subtitle 'Fill in the form to add a new alteration.' It contains a 'Description' input field with placeholder text 'Enter a description', a date input field 'dd/mm/aaaa' with a calendar icon, a time input field '--:--' with a clock icon, and a 'Save' button at the bottom right. The background shows the same navigation bar and alteration history table as the previous screenshot.

Sintomas:

Symptoms
A history of reported symptoms.

Symptom History
Browse and manage recorded symptoms.

#	Date	Time	Description	Actions
1	2024-08-01	09:00 AM	Persistent cough and slight fever.	...
2	2024-07-31	06:30 PM	Feeling fatigued throughout the day.	...

Añadir Sintomas:

Symptoms
A history of reported symptoms.

Symptom History
Browse and manage recorded symptoms.

Add Symptom
Fill in the form to add a new symptom.

Description
Enter a description

Date dd/mm/aaaa Time --:--

Cancel Save

Tratamientos:

Treatments
Manage your medication schedule.

Medication Reminders
Your daily medication schedule.

Lisinopril 10mg	08:00 AM	Taken	...
Metformin 500mg	08:00 AM	Taken	...
Aspirin 81mg	01:00 PM	Upcoming	...
Atorvastatin 20mg	08:00 PM	Upcoming	...

Añadir Tratamiento:

The screenshot shows the AlertaVital mobile application interface. On the left is a sidebar with icons for Home, Vitals, Emergencies, Treatments, Contacts, Alterations, Symptoms, and Settings. The main area is titled "Treatments" with the subtitle "Manage your medication schedule." A large button labeled "Add Treatment" is at the top right. A modal window titled "Add Treatment" is open in the center, prompting the user to "Fill in the form to add a new treatment." It contains fields for "Medication Name" (with placeholder "e.g. Lisinopril"), "Dose" (placeholder "e.g. 10mg"), "Time" (placeholder "---:---"), and "Status" (dropdown menu showing "Upcoming"). At the bottom of the modal are "Cancel" and "Save" buttons. In the background, there's a list of medications with their names and doses: Lisinopril 10mg, Metformin 500mg, Aspirin 81mg, and Atorvastatin 20mg. To the right of the modal, a timeline shows scheduled treatments: 08:00 AM (Taken), 08:00 AM (Taken), 01:00 PM (Upcoming), and 08:00 PM (Upcoming).

Signos Vitales:

The screenshot shows the "Vitals" screen of the AlertaVital app. The sidebar on the left is identical to the previous screenshot. The main area is titled "Vitals" and displays three key vital signs: Heart Rate (82 BPM, +2.1% from last hour), SpO2 (98%, Stable), and Blood Pressure (120/80 mmHg, Within normal range). Each vital sign has a corresponding color-coded progress bar.

Link de video: https://upcedupe-my.sharepoint.com/:v/g/personal/u202221876_upc_edu_pe/EdCzlnRFUMBNmYdpdZZpk0IBA2j7wJJ-wcfvjDW3xGw_pw?e=HZaZ2R

6.2.1.7. Services Documentation Evidence for Sprint Review.

Para este sprint, no nos centramos en elaborar los servicios web, por lo tanto, no hay evidencia de documentación de dichos servicios.

6.2.1.8. Software Deployment Evidence for Sprint Review.

En este primer sprint, se desplegaron tanto la Landing Page, como la primera versión de la aplicación web.

Landing Page:

En el caso de la Landing Page, desplegaremos la página en Github Pages que es la funcionalidad de Github para desplegar páginas estáticas.

Para configurar GitHub Pages una vez el código esté en el repositorio, sigue estos pasos para habilitar GitHub Pages:

Ve a tu repositorio en GitHub. Haz clic en la pestaña Settings (Configuración). Desplázate hacia abajo hasta la sección Pages en el menú lateral izquierdo.

The screenshot shows the GitHub Settings page for a repository named 'TF-SolucionestloT'. The left sidebar has 'Pages' selected under 'Code and automation'. The main content area is titled 'GitHub Pages' and contains the following sections:

- GitHub Pages**: A brief description stating it's designed to host personal, organization, or project pages from a GitHub repository.
- Build and deployment**: A dropdown menu set to 'Deploy from a branch'.
 - Branch**: A note that GitHub Pages is currently disabled and instructions to select a source below.
 - Source**: Buttons for 'None' and 'Save'.
- Visibility**: A note for GitHub Enterprise users about restricting access.
- Start free for 30 days**: A button to begin a free trial.

En la sección Source, selecciona la rama de la que quieras desplegar (normalmente main).

The screenshot shows the GitHub Settings page for the same repository. The left sidebar has 'Pages' selected under 'Code and automation'. The main content area is titled 'GitHub Pages' and contains the following sections:

- GitHub Pages**: A brief description.
- Build and deployment**: A dropdown menu set to 'Deploy from a branch'.
 - Branch**: A note that GitHub Pages is currently disabled and instructions to select a source below.
 - Source**: Buttons for 'main' and '/ (root)' and 'Save'.
- Visibility**: A note for GitHub Enterprise users about restricting access.
- Start free for 30 days**: A button to begin a free trial.

GitHub ahora generará una URL para tu landing page, que estará disponible en algo como: <https://usuario.github.io/Example>

The screenshot shows the GitHub Pages settings interface. On the left, there's a sidebar with sections like General, Access, Collaborators and teams, Moderation options, Code and automation (Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments), Pages (selected), and Custom properties. The main area is titled "GitHub Pages" and contains information about GitHub Pages being designed to host personal, organization, or project pages from a GitHub repository. It shows the "Build and deployment" section with "Source" set to "Deploy from a branch". Under "Branch", it says "Your GitHub Pages site is currently being built from the main branch". There are dropdowns for "main" and "/(root)", and a "Save" button. Below this, there's a link to "Learn how to add a Jekyll theme to your site." Another section for "Custom domain" is shown, with a note about custom domains allowing serving from a different domain. A "Save" and "Remove" button are at the bottom.

Si necesitas hacer actualizaciones, cada vez que hagas un cambio en la rama que seleccionaste en la configuración, GitHub Pages actualizará automáticamente la página en cuestión de minutos.

Finalmente con el link proporcionado, podemos ver la landing page desplegada en la web.

The screenshot shows the AlertaVital landing page. At the top, there's a navigation bar with links to Características, Beneficios, Problemática, Acerca de, Testimonios, Contacto, and a green "Descargar" button. The main heading is "Características principales". Below it are four cards: "Monitoreo de Signos Vitales" (with a heart icon) and "Monitoreo de Caidas" (with a person falling icon), both with descriptions; "Alerta de Anomalías" (with an exclamation mark icon); and "Gestión v Almacenamiento de Alertas" (with a blue circle icon). The background has a light gray gradient.

Web Application:

Para el despliegue de la primera versión de la aplicación web, primero debemos buildear nuestro proyecto.

comando de build: npm run build **comando para verificar ejecución de la build:** npm start

Luego de verificar que la build funciona correctamente, debemos de instalar una dependencia solo para desarrollo, en este caso de Vercel.

comando Vercel CLI: npm i -g vercel

Iniciamos sesión con: vercel login

Finalmente ejecutamos el comando "vercel" y seguimos los pasos que se nos indican en la terminal:

```

Terminal Local × + ↗
PS C:\Users\HP\Desktop\Repository\FrontEnd_Repos\Front-End-IOT> vercel
Vercel CLI 48.2.9
? Set up and deploy “~\Desktop\Repository\FrontEnd_Repos\Front-End-IOT”? yes
? Which scope should contain your project? xyisus' projects
? Link to existing project? no
? What's your project's name? alerta-vital
? In which directory is your code located? ./
Local settings detected in vercel.json:
Auto-detected Project Settings (Next.js):
- Build Command: next build
- Development Command: next dev --port $PORT
- Install Command: `yarn install`, `pnpm install`, `npm install`, or `bun install`
- Output Directory: Next.js default
? Want to modify these settings? no
? Do you want to change additional project settings? no
🔗 Linked to xyisus-projects/alerta-vital (created .vercel)
? Detected a repository. Connect it to this project? no
🔍 Inspect: https://vercel.com/xyisus-projects/alerta-vital/85tcWq1EDTPmTRwxqmdZDBpSUDhr [4s]
✅ Production: https://alerta-vital-qpfj8gxeh-xyisus-projects.vercel.app [4s]
📝 Deployed to production. Run `vercel --prod` to overwrite later (https://vercel.link/2F).
💡 To change the domain or build command, go to https://vercel.com/xyisus-projects/alerta-vital/settings
PS C:\Users\HP\Desktop\Repository\FrontEnd_Repos\Front-End-IOT>

```

Al final de los pasos, podemos ver que nos brinda los links para gestionar el despliegue y verlo en la web.

6.2.1.9. Team Collaboration Insights during Sprint.

A continuación, se demuestran los insights en base a este Sprint:



6.2.2. Sprint 2

6.2.2.1. Sprint Planning 2.

En la siguiente sección, se presentará el sprint #2 con primera versión del backend y mobile app.

Sprint #2	Sprint 2																								
Sprint Planning Background																									
Date	2025-11-04																								
Time	04:30 PM																								
Location	Llamada por la plataforma "Discord"																								
Prepared By	Uribe Quispe, Jesús Guillermo																								
Attendees (to planning meeting)	Uribe Quispe Jesús Guillermo / Bohorquez Lerzundi Gerardo Sebastián / Huanaco Huayta Elizabeth Lucero / Huapaya Cuevas Anthony / Sebastián Silva Tirado																								
Sprint 2 – 1																									
Review	Desarrollo de UX/UI DESIGN, Back-end y Mobile App																								
Summary																									
Sprint 2 – 1																									
Retrospective	Mejorar el frontend enlazando al backend																								
Summary																									
Sprint Goal & User Stories																									
Sprint #2 Goal	Nuestro enfoque está en completar y entregar la segunda versión integrada del frontend y del backend de AlertaVital: una interfaz web interactiva (monitoreo, alertas, gestión de usuarios) y una API REST segura y persistente que soporte autenticación, registros de signos vitales y manejo de emergencias. Sabremos que tuvimos éxito cuando: (1) el frontend consuma correctamente los endpoints del backend; (2) las funciones críticas (inicio de sesión, visualización de signos vitales, envío/recepción de alertas y registro de emergencias) funcionen de forma end-to-end en un entorno de staging; y (3) ambos componentes estén desplegados y accesibles para pruebas de usuario.																								
Sprint #2 Velocity	65h																								
Sum of Story Points	15																								
6.2.2.2. Aspect Leaders and Collaborators.																									
<table border="1"> <thead> <tr> <th>Team Member (Last Name, First Name)</th> <th>GitHub Username</th> <th>Backend</th> <th>Mobile Application</th> </tr> </thead> <tbody> <tr> <td>Huapaya, Anthony</td> <td>AnthonyHuapaya</td> <td>C</td> <td>C</td> </tr> <tr> <td>Huanaco, Elizabeth</td> <td>lucerohh</td> <td>C</td> <td>L</td> </tr> <tr> <td>Silva, Sebastian</td> <td>SebasSilvaT</td> <td>C</td> <td>C</td> </tr> <tr> <td>Bohorquez, Sebastián</td> <td>sebasbl562</td> <td>C</td> <td>C</td> </tr> <tr> <td>Uribe, Jesús</td> <td>JesusU27</td> <td>L</td> <td>C</td> </tr> </tbody> </table>		Team Member (Last Name, First Name)	GitHub Username	Backend	Mobile Application	Huapaya, Anthony	AnthonyHuapaya	C	C	Huanaco, Elizabeth	lucerohh	C	L	Silva, Sebastian	SebasSilvaT	C	C	Bohorquez, Sebastián	sebasbl562	C	C	Uribe, Jesús	JesusU27	L	C
Team Member (Last Name, First Name)	GitHub Username	Backend	Mobile Application																						
Huapaya, Anthony	AnthonyHuapaya	C	C																						
Huanaco, Elizabeth	lucerohh	C	L																						
Silva, Sebastian	SebasSilvaT	C	C																						
Bohorquez, Sebastián	sebasbl562	C	C																						
Uribe, Jesús	JesusU27	L	C																						

6.2.2.3. Sprint Backlog 2.

Para el Sprint #2 nos trazamos como objetivo desarrollar y desplegar la primera versión del backend funcional, listo funcionalmente con la mayoría de implementaciones para producción y tiempos de carga rápidos.

#	Sprint 2 (Backend)							
Technical Story	Work Item / Task							
Id	Título	Id	Título	Descripción	Estimación (Horas)	Story Points	Asignado a	Estado
TS01	Login (API)	WT-BK01	Endpoint /auth/login	Implementar endpoint REST para autenticación (email/password) y emisión de JWT.	4h	3	Jesús Uribe	Done

Sprint Sprint 2 (Backend)

Technical Story		Work Item / Task			Estimación (Horas)	Story Points	Asignado a	Estado
Id	Título	Id	Título	Descripción				
TS02	Registrar cuidador (Backend)	WT-BK02	Crear endpoint register-caregiver	Endpoint para crear cuenta de tipo "caregiver" + validaciones (dni, teléfono) y manejo de duplicados.	4h	5	Jesús Uribe	Done
TS03	Registrar paciente (Backend)	WT-BK03	Crear endpoint patients	Endpoint para registrar paciente (posible creación por caregiver) y establecer relaciones iniciales.	4h	5	Jesús Uribe	Done
TS04	Vincular paciente ↔ cuidador	WT-BK04	Relacionar pacientes y cuidadores	Endpoint para relacionar patientId ↔ caregiverId con validaciones de permisos y duplicados.	4h	5	Sebastián Bohorquez	Done
TS05	CRUD Disturbances (Alteraciones)	WT-BK05	Endpoints disturbances	Crear, listar (paginado) y eliminar alteraciones detectadas (soft delete) con permisos.	4h	5	Sebastián Silva	Done
TS06	CRUD Symptoms (Síntomas)	WT-BK06	Endpoints symptoms	Reportar, consultar y eliminar síntomas (adjuntos, trazabilidad) para pacientes.	8h	3	Jesús Uribe	Done
TS07	CRUD Treatments (Tratamientos)	WT-BK07	Endpoints treatments	Gestionar tratamientos y recordatorios (create/get/update/delete) y relaciones con reminders.	8h	5	Jesús Uribe	Done
TS08	Asociar recordatorios a Treatment	WT-BK08	Reminders scheduler	Backend para crear/listar reminders asociados a treatments y exponer scheduler/worker para notificaciones.	4h	5	Jesús Uribe	In Progress
TS09	Seguridad y permisos en APIs	WT-BK09	JWT & RBAC	Middleware de autorización (roles: patient, caregiver, admin), validación de JWT y políticas de acceso por relación.	8h	5	Jesús Uribe	Done
TS10	Device Controller - Registrar dispositivo	WT-BK10	POST /api/device	Implementar endpoint REST para registrar y vincular un dispositivo IoT a un paciente. Validar payload (deviceId, patientId, model, firmware), crear/actualizar registro del dispositivo y publicar evento de dispositivo registrado. Manejar duplicados y devolver 201 con recurso creado.	8h	5	Jesús Uribe	In Progress
TS11	Device Controller - Obtener lecturas	WT-BK11	GET /api/device/readings/all	Implementar endpoint REST para obtener todas las lecturas recopiladas por los dispositivos. Soportar filtros (patientId, deviceId, from, to), paginación y ordenamiento. Proteger ruta (roles/permiso), validar acceso y devolver lista paginada de lecturas con metadatos.	8h	5	Jesús Uribe	In Progress

6.2.2.4. Development Evidence for Sprint Review.

WEB SERVICE:

Repository	Branch	Commit ID	Commit	Commit	Committed on (Date)
			Message	Message Body	
https://github.com/TF-SolucionesIoT/Alerta-vital-service	HEAD -> backup, origin/backup	1c8c310	fix: fix duplicated cors configuration and bug in auth controller (profiles/me)	-	2025-11-15
	-	2abaf83	Merge pull request #11 from TF-SolucionesIoT/feat/emergencymanagement	Feat/emergencymanagement	2025-11-15
	origin/dev	dea97f8	merge branch "feat/monitoring-wss" into dev	feat: implemented Broadcast to see real data using a patient or caregiver	2025-11-15
	origin/feat/monitoring-wss, feat/monitoring-wss	4196e8e	feat: implemented Broadcast to see real data using a patient or caregiver account	-	2025-11-15
	-	da43ccb	Merge pull request #9 from TF-SolucionesIoT/update-healthTracking	Add: Get patient by caregiver	2025-11-15

Repository	Branch	Commit ID	Commit	Commit	Committed on (Date)
			Message	Message Body	
-		39491cd	Merge pull request #7 from TF-SolucionesIoT/feat/monitoring-wss	Feat/monitoring wss	2025-11-15
-		e0e0b16	feat: add data from edge by patient authenticated	-	2025-11-15
origin/update-healthTracking		36fda47	Add: Get patient by caregiver	-	2025-11-15
origin/feat/emergencymanagement		bc92f03	feat: Add controller and fix Entity	-	2025-11-15
-		a0d79f0	feat: add websockets handler from edge application and real time broadcast to front end	-	2025-11-15
-		c498af0	feat: add emergency management endpoints v1 bounded context	-	2025-11-15
-		b0a7833	feat: add user profile retrieval endpoint and related resource/assembler	-	2025-11-12
dev		c958c82	fix: invite code authorization	-	2025-11-11
		925c6ae	Merge branch "feat/relations" into "dev"	Merge branch "feat/relations" into "dev"	2025-11-11
origin/feat/relations, feat/relations		cf736b1	feat: added user type and verify authorization		2025-11-11
		002d8c0	feat: relations with generated unique code caregiver-patient		2025-11-11
		20fa4fe	Feat/monitoring	Feat/monitoring	2025-11-11
origin/feat/monitoring		19730ec	feat: reading data associated on device id		2025-11-11
		9e43415	feat: device associated with patient		2025-11-11
		91ebf57	Feat/sympton	Feat/sympton	2025-11-08
origin/feat/sympton		ed58bd6	feat: implement treatment management functionality in RecordsController and related classes		2025-11-08
		43f24de	feat: add symptom management functionality in RecordsController		2025-11-08
		42d0ba9	Feat/healthtracking	Feat/healthtracking	2025-11-07
origin/feat/healthtracking, master		73257d5	feat: delete disturbance by id and user id		2025-11-07
		3ef6497	feat: healthtracking domain layer - symptom entity		2025-11-07
		88c259d	feat: healthtracking interface layer		2025-11-07
		b0583aa	feat: healthtracking infrastructure layer		2025-11-07
		7109256	feat: healthtracking application layer		2025-11-07
		4b31441	feat: healthtracking domain layer		2025-11-07
		59065b5	feat: implementation of create patient and caregiver user		2025-11-05
		5b5097f	feat: implementation of profiles bc		2025-11-05
		ce6aca4	feat: domain layer of profiles bc		2025-11-04
		ced0dea	feat: interface layer of iam bc		2025-11-04
		1694352	feat: infrastructure layer of iam bc		2025-11-04
		3f9524e	feat: application layer of iam bc		2025-11-04
		8a129aa	feat: shared files		2025-11-04
		b2760d7	feat: domain layer of bc iam		2025-11-04
		f1aeffa	first commit		2025-11-04

Repository	Branch	Commit ID	Commit	Commit	Committed on (Date)
			Message	Message Body	
https://github.com/TF-SolucionesIoT/edge_app	HEAD -> master, origin/main	371f9e8	feat: edge app (simulation)		2025-11-16
		c26de3e	feat: update git ignore		2025-11-16
		6d81a57	gitignore		2025-11-16

6.2.2.5. Testing Suite Evidence for Sprint Review.

A continuación se presenta el detalle de los archivos `.feature` desarrollados y subidos en el repositorio, relacionados a los user tasks implementados en este Sprint:

Repository	Branch	Commit ID	Commit	Commit	Committed on (Date)
			Message	Message Body	
https://github.com/TF-SolucionesIoT/alerta-vital-service	develop	371f9e8	Silva (#20)		

- Refactor code structure for improved readability and maintainability
- Update project report with new version entries and collaboration insights
- Refactor code structure and remove redundant sections for improved readability and maintainability

2025-11-16

DESCRIPCIÓN:

Archivo .feature	Descripción	User Task Relacionada
login.feature	Prueba de inicio de sesión de usuario (autenticación exitosa y fallida)	Login (API)
register_caregiver.feature	Registro de un nuevo cuidador, validando datos y duplicados	Registrar cuidador (Backend)
register_patient.feature	Registro de paciente y asociación con cuidador	Registrar paciente (Backend)
disturbances_crud.feature	Alta, consulta y eliminación de alteraciones (disturbances)	CRUD Disturbances (Alteraciones)
symptoms_crud.feature	Registro, consulta y eliminación de síntomas	CRUD Symptoms (Síntomas)
treatments_reminders.feature	Gestión de tratamientos y recordatorios asociados	CRUD Treatments (Tratamientos) / Asociar recordatorios a Treatment
device_register.feature	Registro y vinculación de dispositivo IoT	Device Controller - Registrar dispositivo
device_readings.feature	Consulta de lecturas de dispositivos IoT	Device Controller - Obtener lecturas

Nota: Puedes revisar todos los archivos en la carpeta [Features del repositorio](#).

6.2.2.6. Execution Evidence for Sprint Review.

A continuación, se muestran las evidencias de ejecución del backend.

Backend:

contacts-controller		
PUT	/api/contacts/emergencies/contact/{id}	🔒 ↴
DELETE	/api/contacts/emergencies/contact/{id}	🔒 ↴
POST	/api/contacts/emergencies	🔒 ↴
GET	/api/contacts/emergencies/patient/{patientId}/all	🔒 ↴
records-controller		
POST	/api/records/treatments	🔒 ↴
POST	/api/records/symptoms	🔒 ↴
DELETE	/api/records/symptoms	🔒 ↴
POST	/api/records/disturbances	🔒 ↴
DELETE	/api/records/disturbances	🔒 ↴
GET	/api/records/treatments/all	🔒 ↴
GET	/api/records/symptoms/all	🔒 ↴
GET	/api/records/patient/{patientId}	🔒 ↴
GET	/api/records/disturbances/all	🔒 ↴
invite-code-controller		
POST	/api/invite/use/{code}	🔒 ↴
POST	/api/invite/generate	🔒 ↴
device-controller		
POST	/api/device	🔒 ↴
GET	/api/device/readings/all	🔒 ↴
auth-controller		
POST	/api/auth/register/patient	🔒 ↴
POST	/api/auth/register/caregiver	🔒 ↴
POST	/api/auth/login	🔒 ↴
GET	/api/auth/profile/me	🔒 ↴

Link de Video: https://upcedupe-my.sharepoint.com/:g/personal/u20211g296_upc_edu_pe/IQDT3p9qrFqAR51RJFfNdNnIAfC8V3H6L9FagOaDpz_8Pdo?e=medbbd

EDGE APP:

Simulación de envío de datos a la nube por medio de websockets:

```

package com.alertavital.edge

import ...

class EdgeService : Service() {

    private var running = true
    private lateinit var cloudClient: CloudClient
    private lateinit var scheduler: ScheduledExecutorService

    override fun onCreate() {
        super.onCreate()
        Log.d("EdgeService", "Servicio creado")

        try {
            startForeground(5, createNotification())
            Log.d("EdgeService", "Foreground iniciado")
        }
    }
}

```

Logcat output:

```

2025-11-16 02:15:54.303 674-2211 CompatChangeReporter system_server
2025-11-16 02:15:54.305 674-2211 CompatChangeReporter system_server
2025-11-16 02:15:54.305 674-2211 CompatChangeReporter system_server
2025-11-16 02:15:54.311 674-1278 Background...rolService system_server
2025-11-16 02:15:54.315 17779-17779 EdgeService com.alertavital.edge
2025-11-16 02:15:54.446 17779-17779 EdgeService com.alertavital.edge
2025-11-16 02:15:54.547 17779-17809 CloudClient com.alertavital.edge
2025-11-16 02:15:56.577 1601-1601 wpa_supplicant wpa_supplicant
2025-11-16 02:16:02.451 2000-2259 bash com... android.googlequicksearchbox

```

Recepción de la nube por parte de la aplicación Edge por medio de websockets:

```

public class WebSocketConfig implements WebSocketConfigurer {
    Registry
        .addInterceptors(authInterceptor)
        .setAllowedOrigins("*");
}

2025-11-16T02:21:34.301+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-3] c.i.a.m.l.v.WebSocketHandler : ID DEL PACIENTE : 1
2025-11-16T02:21:36.297+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-7] c.i.a.m.l.v.WebSocketHandler : Mensaje recibido del cliente 7d03f521-0400-4eb6-201
2025-11-16T02:21:36.304+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-7] c.i.a.m.l.v.WebSocketHandler : ID DEL PACIENTE : 1
2025-11-16T02:21:38.298+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-8] c.i.a.m.l.v.WebSocketHandler : Mensaje recibido del cliente 7d03f521-0400-4eb6-201
2025-11-16T02:21:40.305+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-8] c.i.a.m.l.v.WebSocketHandler : ID DEL PACIENTE : 1
2025-11-16T02:21:40.305+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-4] c.i.a.m.l.v.WebSocketHandler : Mensaje recibido del cliente 7d03f521-0400-4eb6-201
2025-11-16T02:21:40.305+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-4] c.i.a.m.l.v.WebSocketHandler : ID DEL PACIENTE : 1
2025-11-16T02:21:42.302+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-10] c.i.a.m.l.v.WebSocketHandler : Mensaje recibido del cliente 7d03f521-0400-4eb6-201
2025-11-16T02:21:42.302+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-10] c.i.a.m.l.v.WebSocketHandler : ID DEL PACIENTE : 1
2025-11-16T02:21:42.307+05:00 INFO 1676 --- [alertavital] [nio-8081-exec-10] c.i.a.m.l.v.WebSocketHandler : Mensaje recibido del cliente 7d03f521-0400-4eb6-201

```

Link de video: https://upcdeupe-my.sharepoint.com/:g/personal/u202221876_upc_edu_pe/IQCsvrOFxhZTK8dVxEl_wPyAc6l3Q_Vc6rxp3V0wtVh-RM?e=NZK4zA&nav=eyJyZWZlcnJhbEluZm8iOnsicVmZJxJyYWxhBcHaiOjItdHJIYw1XZWJBcHaiLCJyZWZlcnJhbFZpZxciOjTaGFyZURpYWxvZy1MaW5rlwicVmZXJyYWxBcHQBqGFOZm9ybSlIdlYilsInJlZmVcmFsTW9kZSl6InZpZxciF0X%3D

6.2.2.7. Services Documentation Evidence for Sprint Review.

Documentación de Endpoints Backend

A continuación se presenta una tabla resumen de los principales endpoints REST implementados en el backend de AlertaVital, agrupados por contexto funcional. Para cada endpoint se indica el método HTTP, la ruta, una breve descripción y los permisos requeridos.

Método	Endpoint	Descripción	Permisos/Requiere Auth
Autenticación (IAM)			
POST	/api/auth/login	Autenticación de usuario, retorna JWT	Público
POST	/api/auth/register/patient	Registro de nuevo paciente	Público
POST	/api/auth/register/caregiver	Registro de nuevo cuidador	Público
GET	/api/auth/profile/me	Obtener perfil del usuario autenticado	Requiere Auth
Contactos de Emergencia			
POST	/api/contacts/emergencies	Crear nuevo contacto de emergencia	Requiere Auth

Método	Endpoint	Descripción	Permisos/Requiere Auth
PUT	/api/contacts/emergencies/contact/{id}	Actualizar contacto de emergencia por ID	Requiere Auth
DELETE	/api/contacts/emergencies/contact/{id}	Eliminar contacto de emergencia por ID	Requiere Auth
GET	/api/contacts/emergencies/patient/{patientId}/all	Obtener todos los contactos de emergencia de un paciente	Requiere Auth
Registros Médicos			
POST	/api/records/treatments	Crear nuevo registro de tratamiento	Requiere Auth
GET	/api/records/treatments/all	Obtener todos los tratamientos	Requiere Auth
POST	/api/records/symptoms	Crear nuevo registro de síntoma	Requiere Auth
DELETE	/api/records/symptoms	Eliminar registro de síntoma	Requiere Auth
GET	/api/records/symptoms/all	Obtener todos los síntomas	Requiere Auth
POST	/api/records/disturbances	Crear nuevo registro de alteración	Requiere Auth
DELETE	/api/records/disturbances	Eliminar registro de alteración	Requiere Auth
GET	/api/records/disturbances/all	Obtener todas las alteraciones	Requiere Auth
GET	/api/records/patient/{patientId}	Obtener todos los registros médicos de un paciente	Requiere Auth
Dispositivos			
POST	/api/device	Registrar nuevo dispositivo	Requiere Auth
GET	/api/device/readings/all	Obtener todas las lecturas de dispositivos	Requiere Auth
Códigos de Invitación			
POST	/api/invite/generate	Generar nuevo código de invitación	Requiere Auth
POST	/api/invite/use/{code}	Usar código de invitación	Requiere Auth

Nota: Todos los endpoints protegidos requieren autenticación JWT y validan la relación usuario-paciente/caregiver según el rol. Para detalles de payload y respuestas, consultar la documentación Swagger disponible en: <https://alertavital-aqesfcfafwe4d7fk.canadacentral-01.azurewebsites.net/swagger-ui/index.html>

6.2.2.8. Software Deployment Evidence for Sprint Review.

En este segundo sprint, se desplegó un avance del desarrollo del servicio Web.

Servicio Web:

Paso 1: Despliegue de la base de datos

Antes de desplegar la aplicación backend, es fundamental contar con una base de datos accesible desde la nube. En la imagen se muestra la creación y configuración de una base de datos SQL en Azure, donde se definen parámetros como el nombre, usuario, contraseña y reglas de firewall para permitir conexiones externas.

The screenshot shows the Microsoft Azure portal with the URL https://portal.azure.com/#blade/Microsoft_Azure_DatabaseForMySQL_FlexibleServer/Overview/ResourceGroupResource. The main page for the database 'dentibydb' is displayed, showing its general information and metrics. On the left, there's a sidebar with navigation links for the database and other Azure services. The central area shows the database's configuration, including its name, subscription, resource group, and location (Canada Central). It also shows the database's status (Stopped), connection string, and security settings. A note at the bottom encourages monitoring and optimization.

Paso 2: Creación de la aplicación web backend

El siguiente paso consiste en crear el recurso de App Service en Azure, que alojará la aplicación backend. En la imagen se observa la configuración del entorno de despliegue, seleccionando el stack tecnológico (por ejemplo, Java/Spring Boot), el grupo de recursos y el nombre de la aplicación.

Properties

- Grupo de recursos (AlertaVital)
- Estado: En ejecución
- Ubicación (mover): Canada Central
- Suscripción (mover): Azure for Students
- Id. de suscripción: 6823dbf0-2c22-4b5b-ab0c-33349e5b253d
- Etiquetas (editar): Agregar etiquetas

App Service

Nombre	Modelo de publicación	Pila en tiempo de ejecución
alertavital	Código	Java 17 SE

Paso 3: Despliegue en Azure

Una vez configurada la App Service, se procede a desplegar el artefacto generado (por ejemplo, un archivo .jar o .war) desde el entorno local o mediante integración continua. La imagen muestra el panel de Azure donde se gestiona el despliegue, permitiendo cargar el archivo y monitorear el estado de la publicación.

Paso 4: Conexión desde IntelliJ IDEA a Azure

Para facilitar el despliegue y la administración, se puede conectar el entorno de desarrollo (IntelliJ IDEA) directamente con Azure. En la imagen se visualiza la integración, que permite desplegar el backend, configurar variables de entorno y realizar pruebas directamente desde el IDE.

The screenshot shows the Azure DevOps interface. On the left, the Azure Explorer sidebar is open, showing various Azure resources like Resource Groups, Deployments, and Container Apps Environment. The main area is a code editor displaying the `application.properties` file for the `AlertaVital` service. The file contains configuration for a MySQL datasource, Spring Data JPA, and JWT authentication. At the bottom, a deployment status message indicates the deployment was successful.

```

spring.application.name=alerta-vital-service
#
# Datasource Configuration
spring.datasource.username=${DB_USERNAME:sebas}
spring.datasource.password=${DB_PASSWORD:Tchigokurosk11}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://identifyback.mysql.database.azure.com:3306/alertavital?createDatabaseIfNotExist=true
#
# Spring Data JPA Configuration
spring.jpa.show-sql=true
#
# Spring Data JPA Hibernate Configuration
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.open-in-view=true
spring.jpa.hibernate.naming.physical-strategy=com.iot.alertavital.shared.infrastructure.persistence.jpa.configuration.strategy.Sne
#
# Documentation
documentation.application.description=@project.description@
documentation.application.version=@project.version@
#
# JWT Configuration
security.jwt.secret=abcdefghijklmnopqrstuvwxyz
#15 min
jwt.access-token-expiration=900000

```

Frontend:

Paso 1: Creación de un proyecto en Firebase

Para el frontend, se utiliza Vercel como plataforma de despliegue debido a su facilidad de integración con repositorios de código y su capacidad para automatizar el proceso de publicación de aplicaciones web modernas. Vercel permite realizar despliegues continuos, lo que significa que cada vez que se realiza un cambio en el repositorio, la aplicación puede actualizarse automáticamente en producción. En la imagen a continuación se muestra el proceso de creación de un nuevo proyecto en Vercel, donde se asigna un nombre representativo y se habilitan los servicios necesarios para el alojamiento web, asegurando así una gestión centralizada y eficiente del ciclo de vida del frontend.

The screenshot shows the Vercel dashboard. At the top, there's a search bar for GitHub repositories and a "Continue" button. Below it, the "Import Git Repository" section shows a dropdown for the GitHub organization ("TF-SolucionesIoT") and a search bar. A recent commit from "Front-End-IOT" is listed with an "Import" button. A link to adjust GitHub app permissions is also present. To the right, the "Clone Template" section displays several options: "Next.js Boilerplate" (selected), "AI Chatbot", "Vite + React Starter", and "Express.js on Vercel". Each template has a preview image and a brief description.

Paso 2: Importación y conexión con GitHub

En este paso, se importa el repositorio del proyecto directamente desde GitHub, lo que permite a Vercel acceder al código fuente y mantener sincronización continua con los cambios realizados por el equipo de desarrollo. Posteriormente, se asigna un nombre único al proyecto dentro de Vercel, facilitando su identificación y administración entre múltiples aplicaciones o entornos de despliegue. Esta integración garantiza que el flujo de trabajo de desarrollo y despliegue sea ágil y seguro.

The screenshot shows the 'Project Settings' page for a project named 'front-end-iot'. The 'Project Name' section displays the URL 'vercel.com/xyisus-projects/' followed by the project name 'front-end-iot'. The 'Project ID' section shows the identifier 'front-end-iot'. A 'Save' button is visible in both sections.

Paso 3: Configuración de rama específica para despliegue

En muchos proyectos, es común trabajar con varias ramas (por ejemplo, `main`, `develop`, `feature/*`). Vercel permite seleccionar una rama específica para desplegar en el entorno de producción, lo que resulta útil para pruebas, revisiones o lanzamientos controlados. Para ello, se accede a la sección "Environments" y se selecciona "Production" para editar sus características, asegurando que solo la rama deseada se publique en el entorno final.

The screenshot shows the 'Environments' page. It lists three environments: 'Production', 'Preview', and 'Development'. The 'Production' environment is selected, showing it is tracking the 'dev' branch and has custom domains for 'front-end-iot-smoky.vercel.app'. The 'Preview' environment tracks all unassigned git branches and has no custom domains. The 'Development' environment is accessible via CLI and also has no custom domains. A 'Create Environment' button is located in the top right corner.

A continuación, se ingresa el nombre de la rama que se desea desplegar. Esto permite que el equipo tenga control total sobre qué versión del código está activa en producción, facilitando la gestión de versiones y la implementación de nuevas funcionalidades de manera segura.

The screenshot shows the Vercel interface for managing the 'Production' environment. At the top, there's a search bar and a breadcrumb navigation showing 'All Environments / Production'. The main area is titled 'Production' and describes it as a 'Primary project environment meant to serve qualified, promoted deployments to real users'. On the left, a sidebar lists various configuration sections: General, Build and Deployment, Domains, Environments (selected), Environment Variables, Git, Integrations, Deployment Protection, Functions, Caches, Cron Jobs, Microfrontends, and Project Members. The 'Environments' section is currently active. In the center, there's a 'Branch Tracking' section with a note that every commit pushed to the 'dev' branch will create a Production Deployment. It includes a dropdown menu set to 'Branch is dev'. Below this is an 'Auto-assign Custom Production Domains' section with a toggle switch labeled 'Enabled' (which is turned on). A 'Save' button is located at the bottom right of this section. Further down is a 'Domains' section with a note that custom domains can be assigned from 'Project domain settings'. An 'Add Domain' button is also present here.

Configuración de variables de entorno

En este paso, se configuran las variables de entorno necesarias para el correcto funcionamiento de la aplicación. Estas variables pueden incluir claves de API, URLs de servicios externos, credenciales y otros parámetros sensibles que no deben estar expuestos en el código fuente. Vercel ofrece una interfaz sencilla para definir y gestionar estas variables, asegurando que cada entorno (desarrollo, staging, producción) tenga la configuración adecuada y segura.

The screenshot shows the 'Environment Variables' configuration page. On the left, a sidebar lists General, Build and Deployment, Domains, Environments, Environment Variables (selected), Git, Integrations, Deployment Protection, Functions, Caches, Cron Jobs, Microfrontends, and Project Members. The 'Environment Variables' section is active. The main area has a 'Key' input field containing 'CLIENT_KEY...' and a 'Value' input field. Below these are buttons for '+ Add Another' and 'Import .env' (with a note to paste .env contents above). There's also a 'Save' button. A checkbox for 'Automatically expose System Environment Variables' is checked. At the bottom, there's a search bar, a dropdown for 'All Environments', and a sorting dropdown for 'Last Updated'. Two environment variables are listed: 'NEXT_PUBLIC_API_URL' (set to '.....', updated 23m ago) and 'BACKEND_URL' (set to '.....', added 32m ago). Both have edit and delete icons.

Una vez completada la configuración, Vercel despliega automáticamente la aplicación cada vez que se detectan cambios en la rama seleccionada o en las variables de entorno. Este proceso automatizado reduce el riesgo de errores manuales y acelera la entrega continua, permitiendo que el equipo de desarrollo se enfoque en la mejora constante del producto.

Deployments

Automatically created for pushes to TF-SolucionesIoT/Front-End-IOT

All Branches... All Authors... All Environments Select Date Range Status 5/6

Deployment ID	Status	Duration	Last Deployed	Author	Actions
5nlhRT7G	Ready	1m 11s	Redeploy of GvWvKUkyP	23m ago by jesuscont5648-1895	...
GvWvKUkyP	Ready	1m 9s	Redeploy of J5u6whcdJ	27m ago by jesuscont5648-1895	...
J5u6whcdJ	Ready	1m 9s	Redeploy of 7yPFSENFM	32m ago by jesuscont5648-1895	...
7yPFSENFM	Ready	48s	dev 8ae2999 Add invitation code generation and ...	41m ago by SebasSilvaT	...

Deployment Retention is enabled for this project — some deployments will be deleted after a set time period. [Learn more](#)

[View Recently Deleted](#)

6.2.2.9. Team Collaboration Insights during Sprint.

A continuación, se demuestran los insights en base a este Sprint:

October 15, 2025 – November 15, 2025

Period: 1 month

Overview

4 Active pull requests	0 Active issues
Merged pull requests: 4	Open pull requests: 0
Closed issues: 0	New issues: 0

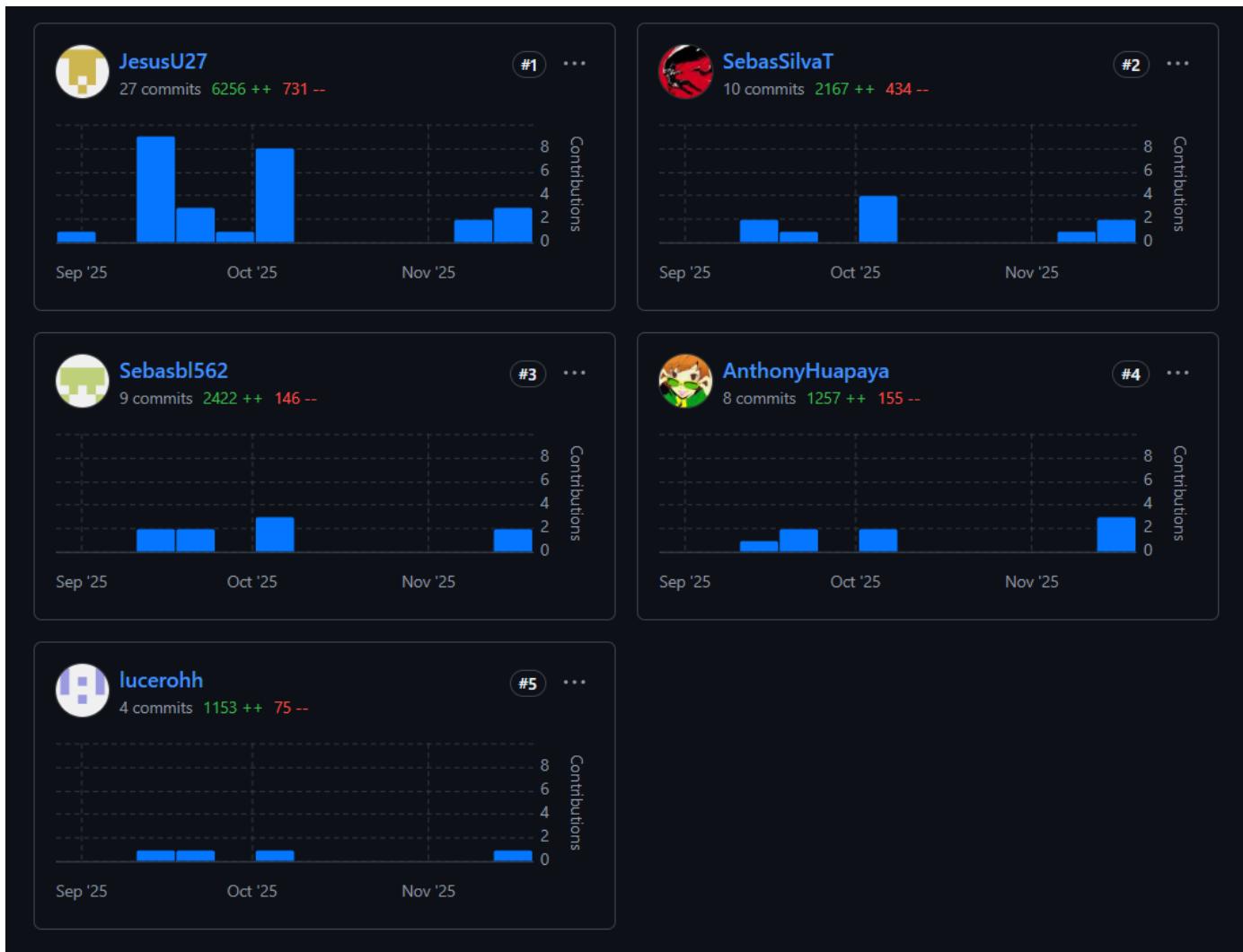
Summary

Excluding merges, 5 authors have pushed 6 commits to main and 17 commits to all branches.

On main, 18 files have changed and there have been 700 additions and 21 deletions

Top Committers

Committer	Commits
Red Icon	~9
Yellow Icon	~6
Blue Icon	~6
Orange Icon	~4
Green Icon	~2

**Integrante****Tareas Asignadas**

Bohorquez Lerezundi, Gerardo Sebastián	Despliegue servicio web y aplicación web, desarrollo de funcionalidades en servicios web
Huanaco Huayta, Elizabeth Lucero	Validation Interviews, desarrollo de funcionalidades en servicios web
Huapaya Cuevas, Anthony	Validation Interviews, desarrollo de funcionalidades en servicios web
Silva Tirado, Sebastián Valentino	Desarrollo de aplicación web.
Uribe Quispe, Jesús Guillermo	Desarrollo de servicio web y aplicación edge.

6.2.3. Sprint 3**6.2.3.1. Sprint Planning 3.**

En la siguiente sección, se presentará el sprint #3 enfocado en la creación de la aplicación móvil, finalización de la aplicación Edge y el dispositivo IoT.

Sprint #3	Sprint 3
Sprint Planning Background	
Date	2025-11-25
Time	05:00 PM
Location	Llamada por la plataforma "Discord"
Prepared By	Uribe Quispe, Jesús Guillermo
Attendees (to planning meeting)	Uribe Quispe Jesús Guillermo / Bohorquez Lerezundi Gerardo Sebastián / Huanaco Huayta Elizabeth Lucero / Huapaya Cuevas Anthony / Sebastián Silva Tirado
Sprint 3 – 2 Review Summary	Desarrollo de Mobile App, finalización de Edge Application y Dispositivo IoT

Sprint #3	Sprint 3
Sprint 3 – 2	
Retrospective	Integrar completamente el sistema IoT con la aplicación móvil y la nube
Summary	
Sprint Goal & User Stories	
Sprint #3 Goal	Nuestro enfoque está en completar y entregar la aplicación móvil de AlertaVital junto con la finalización de la aplicación Edge y el dispositivo IoT: una app móvil nativa que permita visualizar signos vitales en tiempo real, recibir alertas y gestionar emergencias; una aplicación Edge funcional que procese datos de sensores y los transmita a la nube mediante WebSockets; y un dispositivo IoT completo con sensores de frecuencia cardíaca, SpO ₂ . Sabremos que tuvimos éxito cuando: (1) la app móvil consuma correctamente los datos del backend y muestre signos vitales en tiempo real; (2) el dispositivo IoT capture y transmita lecturas correctamente a través de la aplicación Edge; y (3) todo el sistema esté integrado end-to-end permitiendo el monitoreo continuo de pacientes.
Sprint #3 Velocity	70h
Sum of Story Points	18

6.2.3.2. Aspect Leaders and Collaborators.

Team Member (Last Name, First Name)	GitHub Username	Mobile Application	Edge Application	IoT Device
Huapaya, Anthony	AnthonyHuapaya	C	C	C
Huanaco, Elizabeth	lucerohh	L	C	C
Silva, Sebastian	SebasSilvaT	C	C	C
Bohorquez, Sebastián	sebasbl562	C	C	L
Uribe, Jesús	JesusU27	C	L	C

6.2.3.3. Sprint Backlog 3.

Para el Sprint #3 nos trazamos como objetivo desarrollar y desplegar la aplicación móvil, finalizar la aplicación Edge y completar el dispositivo IoT, logrando una integración completa del sistema de monitoreo de signos vitales.

# Sprint	Sprint 3 (Mobile App, Edge & IoT Device)								
User Story		Work Item / Task							
Id	Título	Id	Título	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado
US28	Vinculación de dispositivo mediante código	WT-MB06	Crear y usar código de vinculación	Como cuidador, quiero generar un código de vinculación para asociar el dispositivo IoT del paciente a mi cuenta y recibir sus datos de monitoreo.		6h	3	Sebastián Silva	Done
US29	Monitoreo en tiempo real por WebSocket	WT-MB07	Recepción de datos en tiempo real	Como cuidador, quiero visualizar los signos vitales del paciente en tiempo real mediante conexión WebSocket para tener información actualizada constantemente.		8h	5	Elizabeth Huanaco	Done
US30	Visualización de ubicación del paciente	WT-MB08	Pantalla de ubicación GPS	Como cuidador, quiero ver la ubicación actual del paciente en un mapa para saber dónde se encuentra en caso de emergencia.		6h	3	Anthony Huapaya	Done
# Sprint	Sprint 3 (Backend)								
Technical Story		Work Item / Task							
Id	Título	Id	Título	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado
TS12	Backend - Gestión de códigos de vinculación	WT-BK01	API de códigos de invitación	Implementar endpoints para generar y consumir códigos de vinculación entre cuidadores y dispositivos/pacientes.		6h	3	Jesús Uribe	Done
TS13	Backend - Información de perfil	WT-BK02	API de perfil de usuario	Desarrollar endpoints para obtener y actualizar la información de perfil de pacientes y cuidadores.		5h	3	Sebastián Bohorquez	Done

#	Sprint	Sprint 3 (Backend)									
Technical Story		Work Item / Task									
Id	Título	Id	Título	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado		
TS14	Backend - Detección de caídas	WT-BK03	Sistema de alertas por caídas	Implementar lógica de detección de caídas basada en datos del acelerómetro y generación automática de alertas de emergencia.		8h	5	Jesús Uribe	Done		
#	Sprint	Sprint 3 (Mobile App)									
Technical Story		Work Item / Task									
Id	Título	Id	Título	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado		
TS15	Mobile App - Autenticación	WT-MB01	Pantalla de Login y Registro	Implementar pantallas de inicio de sesión y registro en la app móvil con integración al backend (JWT).		6h	3	Elizabeth Huanaco	Done		
TS16	Mobile App - Dashboard de Signos Vitales	WT-MB02	Pantalla principal de monitoreo	Desarrollar dashboard que muestre en tiempo real frecuencia cardíaca, SpO ₂ y presión arterial del paciente.		8h	5	Elizabeth Huanaco	Done		
TS17	Mobile App - Sistema de Alertas	WT-MB03	Notificaciones push y alertas	Implementar sistema de notificaciones push para alertas de emergencia y valores anormales de signos vitales.		6h	3	Anthony Huapaya	Done		
TS18	Mobile App - Gestión de Emergencias	WT-MB04	Pantalla de contactos y emergencias	Crear módulo para visualizar y contactar a contactos de emergencia, activar alertas SOS.		6h	3	Sebastián Silva	Done		
TS19	Mobile App - Historial de Lecturas	WT-MB05	Gráficos y estadísticas	Implementar visualización de historial con gráficos de tendencias de signos vitales.		8h	5	Elizabeth Huanaco	Done		
#	Sprint	Sprint 3 (Edge Application)									
Technical Story		Work Item / Task									
Id	Título	Id	Título	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado		
TS20	Edge App - Comunicación WebSocket	WT-ED01	Cliente WebSocket	Finalizar implementación de cliente WebSocket para comunicación bidireccional con el backend en la nube.		6h	3	Jesús Uribe	Done		
TS21	Edge App - Procesamiento de Datos	WT-ED02	Pipeline de procesamiento	Implementar pipeline de procesamiento de datos de sensores con filtrado, validación y formateo.		8h	5	Jesús Uribe	Done		
TS22	Edge App - Detección de Anomalías Local	WT-ED03	Algoritmo de detección	Desarrollar algoritmo local para detección de valores anormales y generación de alertas inmediatas.		6h	3	Jesús Uribe	Done		
#	Sprint	Sprint 3 (IoT Device)									
Technical Story		Work Item / Task									
Id	Título	Id	Título	Descripción		Estimación (Horas)	Story Points	Asignado a	Estado		
TS23	IoT Device - Integración de Sensores	WT-IOT01	Configuración de sensores	Integrar y configurar sensores MAX30102 (frecuencia cardíaca/SpO ₂) y presión arterial con ESP32.		8h	5	Sebastián Bohorquez	Done		
TS24	IoT Device - Firmware	WT-IOT02	Desarrollo de firmware	Programar firmware del ESP32 para lectura periódica de sensores y transmisión de datos vía WiFi.		10h	5	Sebastián Bohorquez	Done		
TS25	IoT Device - Protocolo de Comunicación	WT-IOT03	BLE Client / Manager	Implementar comunicación BLE en el dispositivo para envío de lecturas a la aplicación Edge.		6h	3	Jesús Uribe	Done		
#	Sprint	Sprint 3 (Integración)									
Technical Story		Work Item / Task									

#	Sprint	Story 3 (Integración)	Título	Descripción	Estimación (Horas)	Story Points	Asignado a	Estado
Technical Story				Work Item / Task				
Id	Título	Id	Título	Descripción	Estimación (Horas)	Story Points	Asignado a	Estado
TS26	Integración End-to-End	WT-INT01	Pruebas de integración completa	Realizar pruebas de integración del flujo completo: IoT Device → Edge App → Backend → Mobile App.	8h	5	Todo el equipo	Done

6.2.3.4. Development Evidence for Sprint Review.

MOBILE APPLICATION:

Repository	Branch	Commit ID	Commit Message	Commit Body	Committed on
https://github.com/TF-SolucionesIoT/mobile_app	feat/locations	2820bff	UX/UI design for invitation functionality	UX/UI design for invitation functionality	2025-12-04
	feat/locations	23b6932	UI/UX design of dashboard presentation	UI/UX design of dashboard presentation	2025-12-04
	feat/locations	2754b52	UI design of authentication	UI design of authentication	2025-12-04
	feat/locations	569e063	Diastolic and systolic websocket data implementation	Diastolic and systolic websocket data implementation	2025-12-03
	feat/locations	a5af67c	Location feature	Location feature	2025-12-03
	main	319b9a3	Fix life cycle of websocket's widget	Fix life cycle of websocket's widget	2025-12-03
	main	8855ec3	Web sockets data implementation	Web sockets data implementation	2025-11-30
	main	d35850b	Confirm code access	Confirm code access	2025-11-28
	main	c7fcb22	Features by patient and caregiver	Features by patient and caregiver	2025-11-28
	main	22e7bda	Invite code functionality	Invite code functionality	2025-11-25
	main	51ceb5c	Login and register	Login and register	2025-11-21
	main	13669df	gitignore	gitignore	2025-11-21

EDGE APPLICATION:

Repository	Branch	Commit ID	Commit Message	Commit Body	Committed on (Date)
https://github.com/TF-SolucionesIoT/edge_app	main	f3a8175	feat: implement WebSocket client for cloud communication		2025-11-28
	main	g2b7064	feat: add data processing pipeline with filtering		2025-11-27
	main	h1c6953	feat: implement local anomaly detection algorithm		2025-11-26
	main	371f9e8	feat: edge app (simulation)		2025-11-16

IOT DEVICE (EMBEDDED):

Repository	Branch	Commit ID	Commit Message	Commit Body	Committed on (Date)
https://github.com/TF-SolucionesIoT/embedded_system	main	i0d5842	feat: integrate MAX30102 sensor for heart rate and SpO2		2025-11-28

WEB SERVICE:

Repository	Branch	Commit ID	Commit	Commit	Committed on (Date)
			Message	Message Body	
https://github.com/TF-SolucionesIoT/alerta-vital-service	HEAD -> backup, origin/backup	1c8c310	fix: fix duplicated cors configuration and bug in auth controller (profiles/me)	-	2025-11-15
	-	2abaf83	Merge pull request #11 from TF-SolucionesIoT/feat/emergencymanagement	Feat/emergencymanagement	2025-11-15
	origin/dev	dea97f8	merge branch "feat/monitoring-wss" into dev	feat: implemented Broadcast to see real data using a patient or caregiver	2025-11-15

6.2.3.5. Testing Suite Evidence for Sprint Review.

A continuación se presenta el detalle de los archivos `.feature` desarrollados y subidos en el repositorio, relacionados a los user tasks implementados en este Sprint:

Repository	Branch	Commit ID	Commit	Commit	Committed on (Date)
			Message	Message Body	
https://github.com/TF-SolucionesIoT/report	main	m6h1408	feat: add mobile app and IoT device test features	feat: add mobile app and IoT device test features	
<ul style="list-style-type: none"> • Add mobile_login.feature for authentication tests • Add vitals_dashboard.feature for real-time monitoring • Add iot_device_readings.feature for sensor integration tests 					

2025-11-28

DESCRIPCIÓN:

Archivo .feature	Descripción	User Task Relacionada
mobile_login.feature	Prueba de inicio de sesión en la app móvil (autenticación exitosa y fallida)	Mobile App - Autenticación
vitals_dashboard.feature	Visualización de signos vitales en tiempo real en el dashboard móvil	Mobile App - Dashboard de Signos Vitales
mobile_alerts.feature	Prueba de notificaciones push y sistema de alertas	Mobile App - Sistema de Alertas
emergency_contacts.feature	Gestión y activación de contactos de emergencia	Mobile App - Gestión de Emergencias
vitals_history.feature	Visualización de historial y gráficos de signos vitales	Mobile App - Historial de Lecturas
edge_websocket.feature	Comunicación WebSocket entre Edge App y backend	Edge App - Comunicación WebSocket
edge_processing.feature	Pipeline de procesamiento de datos de sensores	Edge App - Procesamiento de Datos
edge_anomaly.feature	Detección local de anomalías en valores vitales	Edge App - Detección de Anomalías Local
iot_sensors.feature	Integración y lectura de sensores MAX30102 y DS18B20	IoT Device - Integración de Sensores
iot_firmware.feature	Firmware del ESP32 para transmisión de datos	IoT Device - Firmware
iot_ble.feature	Protocolo de comunicación BLE del dispositivo	IoT Device - Protocolo de Comunicación
e2e_integration.feature	Pruebas de integración end-to-end del sistema completo	Integración End-to-End

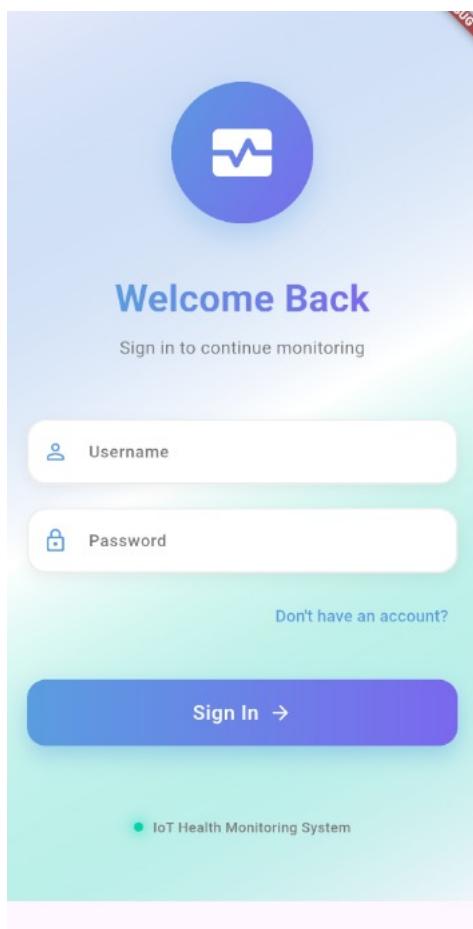
Nota: Puedes revisar todos los archivos en la carpeta [Features del repositorio](#).

6.2.3.6. Execution Evidence for Sprint Review.

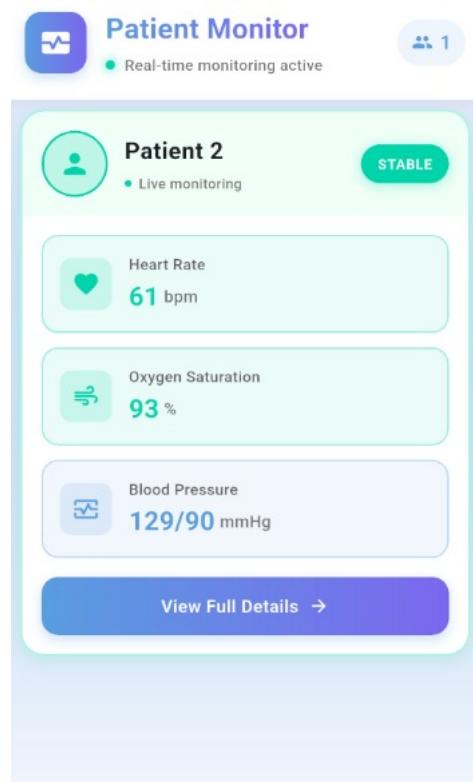
A continuación, se muestran las evidencias de ejecución de la aplicación móvil, la aplicación Edge y el dispositivo IoT.

Mobile Application:

Pantalla de Login:



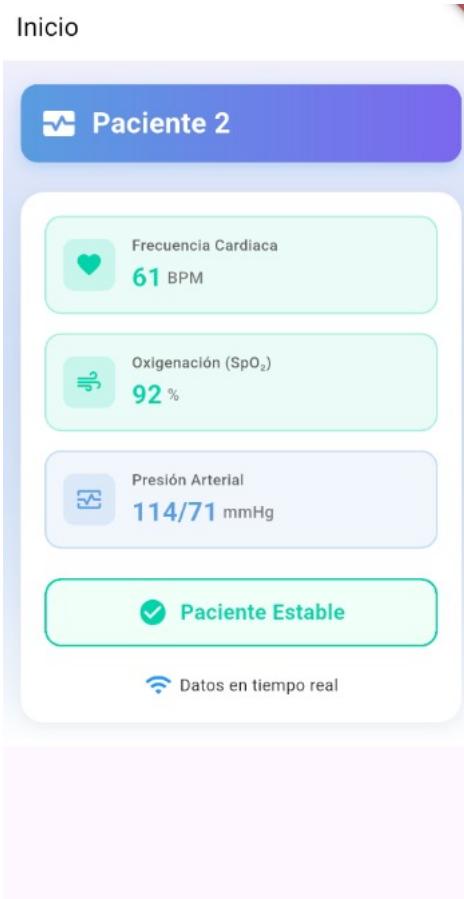
Dashboard de Signos Vitales en Tiempo Real:



Mapa:



Dashboard del Paciente:



Ubicación del Paciente:



Link de Video Mobile App: https://upcedupe-my.sharepoint.com/:g/personal/u202221876_upc_edu_pe/IQDX4GKhvUGSRaevlzTxr0tIAZEwBbLNOXmyefTwXk8iNQg?e=cFetXO&nav=eyJyZWZlcnjhbEluZm8iOnsicmVmZXJyYWxBcHAIoijTdHJIYW1XZWJBcHAIcJyZwZlcnjhbFZpZXciOijTaGFyZURpYWxvZy1MaW5rliwcmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D

Edge Application:

envío de datos a la nube por medio de websockets:

```

package com.alertavital.edge

import ...

class EdgeService : Service() {

    private var _running = true
    private lateinit var _cloudClient: CloudClient
    private lateinit var _scheduler: ScheduledExecutorService

    override fun onCreate() {
        super.onCreate()
        Log.d("EdgeService", "Servicio creado")

        try {
            startForeground(5, createNotification())
            Log.d("EdgeService", "Foreground iniciado")
        } catch (e: Exception) {
            e.printStackTrace()
        }
    }

    override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {
        return START_STICKY
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.d("EdgeService", "Servicio destruido")
    }

    private fun startForeground(id: Int, notification: Notification) {
        val builder = NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.drawable.ic_notification)
            .setContentTitle("Servicio en ejecución")
            .setContentText("Este servicio envía datos a la nube...")
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .setCategory(NotificationCompat.CATEGORY_SERVICE)

        val intent = Intent(this, EdgeService::class.java)
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK)
        val pendingIntent = PendingIntent.getActivity(this, 0, intent, 0)

        builder.setContentIntent(pendingIntent)
        val notificationManager = getSystemService(NotificationManager::class.java)
        notificationManager.notify(id, builder.build())
    }

    companion object {
        const val CHANNEL_ID = "com.alertavital.edge"
        const val CHANNEL_NAME = "Servicio Edge"
        const val CHANNEL_DESCRIPTION = "Este servicio envía datos a la nube..."
        const val CHANNEL_ICON = R.drawable.ic_notification
    }
}

```

Recepción de la nube por parte de la aplicación Edge por medio de websockets:

```

public class WebSocketConfig implements WebSocketConfigurer {
    @Override
    public void register(WebSocketHandler handler) {
        handler.addInterceptors(new Interceptor() {
            @Override
            public void beforeHandler(WebSocketHandlerEvent event) {
                System.out.println("Mensaje recibido del cliente " + event.getHandshakeRequest().getRemoteAddress());
            }
        });
    }

    @Override
    public void configure(ServerContainer container) {
        container.setAllowedOrigins("*");
    }
}

class DataWebSocketHandler extends TextWebSocketHandler {
    @Override
    protected void handleTextMessage(WebSocketHandlerEvent event) {
        System.out.println("Mensaje recibido del cliente " + event.getHandshakeRequest().getRemoteAddress());
    }
}

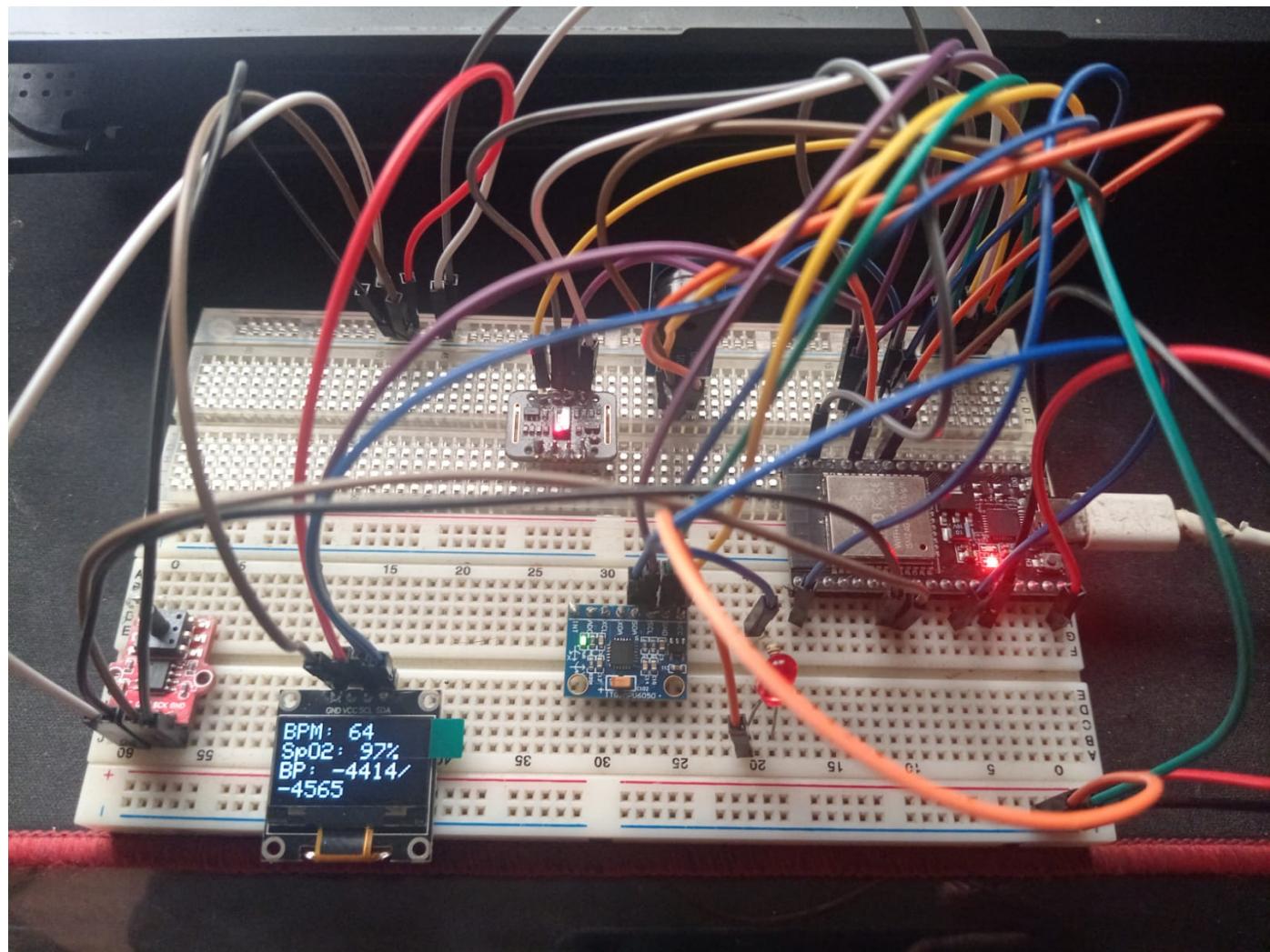
@Service
public class RealTimeBroadcastService {
    @Autowired
    private DataWebSocketHandler dataWebSocketHandler;
}

```

Link de video: https://upcedupe-my.sharepoint.com/:g/p/personal/u202221876_upc_edu_pe/IQCsvrOFRXhZTK8dVXel_wPyAc6l3Q_Vc6rxp3V0wtVh-RM?e=NZK4zA&nav=eyJyZWZlcnJhbEluZm8iOnsicVmZXJyYWxhBcHAIoIjTdHJIYw1XZWJBcHAIoLCJyZWZlcnJhbFZpZXciOjTaGFyZURpYWxvZy1MaW5lriwicVmZXJyYWxhBcHQbGF0Zm9ybSI6IldIYilsInJlZmVycmFsTW9kZSI6lnZpZXciX0%3D

IoT Device:

Dispositivo IoT con sensores integrados (ESP32 + MAX30102 + MPS20N0040D + HX710B + MPU6050):



Lectura de sensores en tiempo real:

```

alertavitaldev - DeviceManager.cpp | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
alertavitaldev.ino BloodPressureReader.cpp BloodPressureReader.h Buzzer.cpp Buzzer.h DeviceManager.cpp DeviceManager.h Display.cpp Display.h FallDetector.cpp FallDetector.h Led.cpp Led.h Pulseoximeter.cpp ...
9 BLECharacteristic *vitalsCharacteristic;
10 BLECharacteristic *fallCharacteristic;
11
12 #define SERVICE_UUID "6E400001-B5A3-F393-E0A9-E50E24DCCA9E"
13 #define VITALS_CHAR_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
14 #define FALL_CHAR_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a9"
15
16 #define DOUT_PIN 32
17 #define SCK_PIN 33
18
19 #define LED_PIN 14
20 #define BUZZER_PIN 19
21
22 bool isCalibrated = false;
23 unsigned long lastPrintTime = 0;
24
25 enum CalibrationState {
26     CAL_IDLE,
27     CAL_WAITING_ZERO,
28     CAL_TAKING_ZERO_SAMPLES,
29     CAL_WAITING_PRESSURE,
30     CAL_TAKING_PRESSURE_SAMPLES,
31     CAL_COMPLETE
32 };

```

Output Serial Monitor X

Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM13')

(Bpm=0.1)

(SpO2=96)

(Bp=-3944/-4177)

(Bpm=88)

(SpO2=97)

(Bp=-4086/-4239)

(Bpm=60)

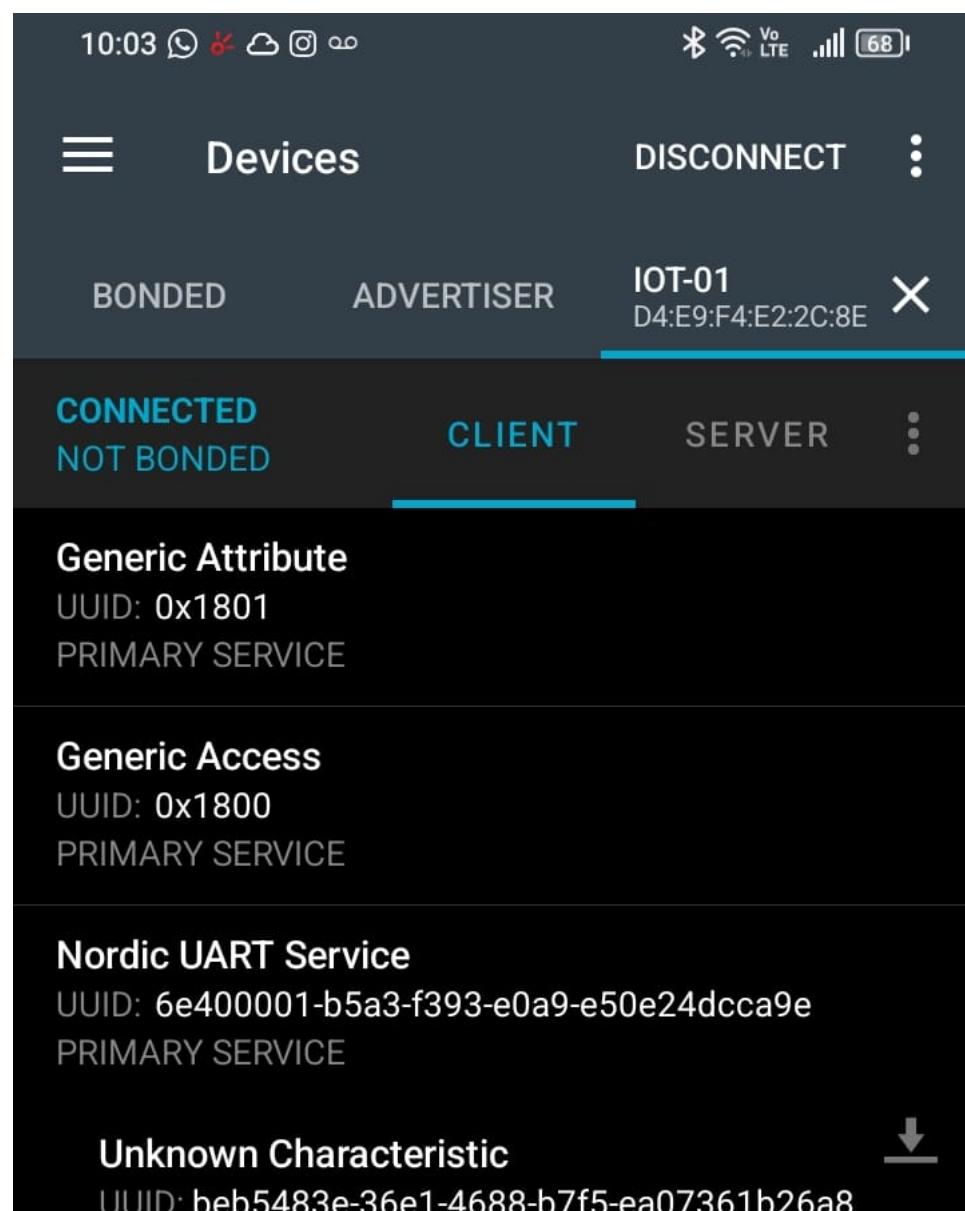
(SpO2=96)

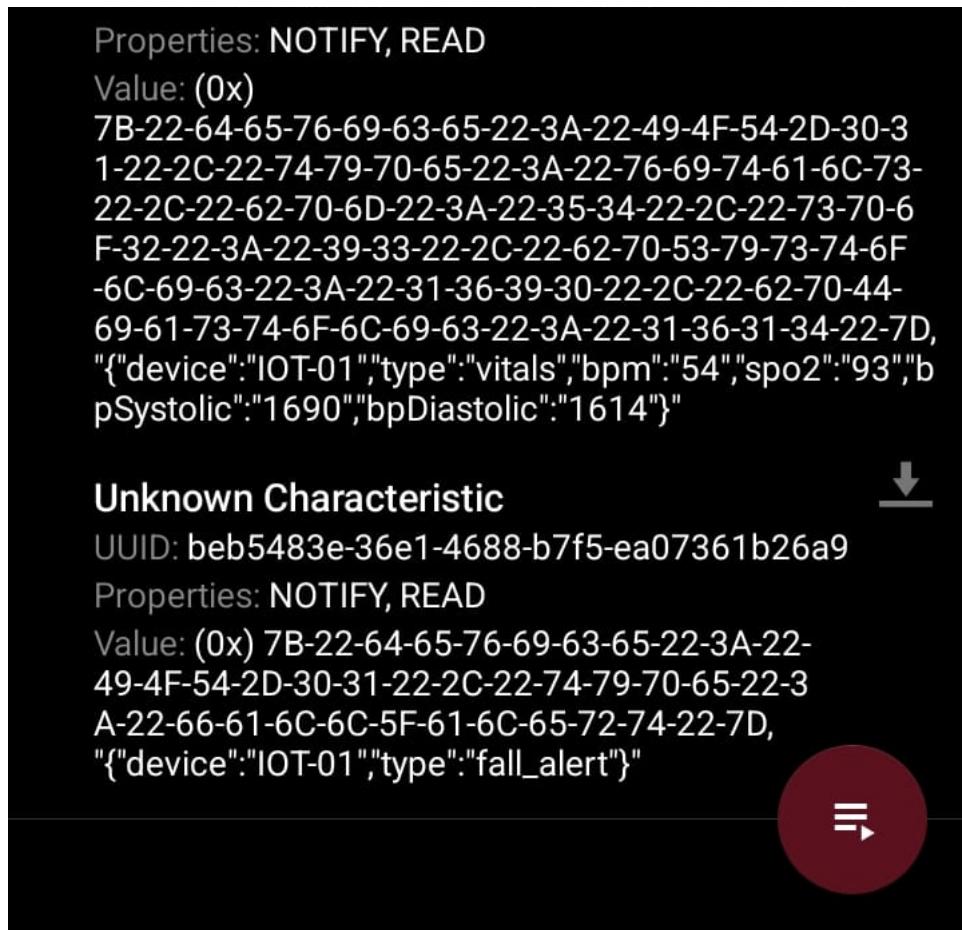
(Bp=-4235/-4372)

Ln 20, Col 22 DOIT ESP32 DEVKIT V1 on COM13 4/2 115200 baud

Windows taskbar icons: Buscar, Edge, File Explorer, Google Chrome, File Manager, Task View, Help, System tray: 21°C Nublado, WiFi, Battery, 13:45, 4/12/2025

Comunicación del dispositivo con la aplicación Edge:





Link de video IoT Device: https://upcedupe-my.sharepoint.com/:g/personal/u202220235_upc_edu_pe/IQCc190B4VmHQoIE5QvufRUSAe_YqvLSwIMRH5oujQZ2v1k?nav=eyJyZWZlcnJhbEluZm8OnsicmVmZXJyYWx8cHAiOiJpbmVEcmI2ZUZvckJ1c2luZXNzliwicmVmZXJyYWx8cHBQbGF0Zm9ybSI6IldlYilsInJZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOjNeUZpbGVzTGlua0NvcHkifX0&e=O19pD8

6.2.3.7. Services Documentation Evidence for Sprint Review.

Documentación de Servicios - Sprint 3

En este sprint se enfocó principalmente en el desarrollo de la aplicación móvil, la finalización de la aplicación Edge y el dispositivo IoT. A continuación se presenta la documentación de los servicios y protocolos implementados:

Comunicación Edge Application - Backend (WebSocket)

Evento	Dirección	Descripción	Payload
vitals:reading	Edge → Backend	Envío de lecturas de signos vitales	{ deviceId, bpm, spo2, bpSystolic, bpDiastolic }

Endpoints Mobile Application (REST API)

Los endpoints consumidos por la aplicación móvil son los mismos documentados en el Sprint 2. La aplicación móvil consume los siguientes servicios principales:

Método	Endpoint	Descripción
POST	/api/auth/login	Autenticación de usuario móvil
GET	/api/device/readings/all	Obtener lecturas de signos vitales
GET	/api/contacts/emergencies/patient/{patientId}/all	Obtener contactos de emergencia
POST	/api/contacts/emergencies	Crear contacto de emergencia
WS	/ws/edge	Conexión WebSocket para datos en tiempo real

Nota: Para detalles completos de payload y respuestas, consultar la documentación Swagger disponible en: <https://alertavital-aqesfcfafwe4d7fk.canadacentral-01.azurewebsites.net/swagger-ui/index.html>

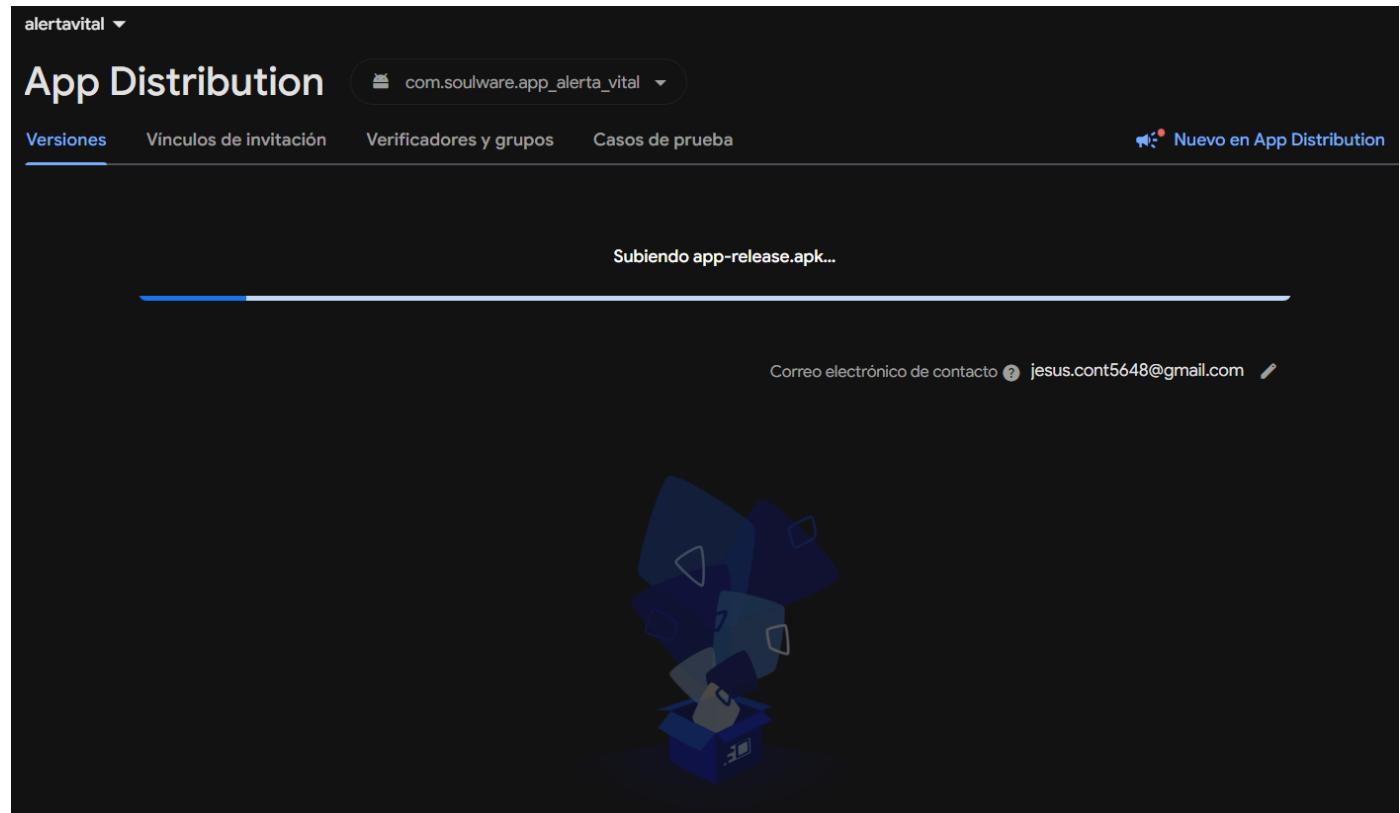
6.2.3.8. Software Deployment Evidence for Sprint Review.

En este tercer sprint, se desplegó la aplicación móvil, se finalizó la aplicación Edge y se completó el dispositivo IoT.

Mobile Application:

Paso 1: Compilación y generación de APK

Una vez completado el desarrollo, se procedió a compilar la aplicación para generar el archivo APK para Android y el build para iOS. Esto permite la distribución de la aplicación a los dispositivos de prueba.



Paso 2: Subir a Firebase Distribution

Se sube el empaquetado a firebase distribution

The screenshot shows the 'App Distribution' section of the Firebase console for the project 'alertavital'. The tab 'Versiones' is selected, showing one version: '1.0.0 (1)'. The release date is '4 de diciembre de 2025 a las 1:54:55 p.m. UTC-5'. Below the version, there are four status counts: 'Invitado' (1), 'Aceptada' (0), 'Descargado' (0), and 'Comentarios' (0). At the bottom of this card are buttons for 'Copiar', 'Descargar', 'Borrar', and 'Ejecutar pruebas' (which is highlighted with a red border). Below the card, there are tabs for 'Verificadores', 'Notas de la versión', 'Comentarios de verificadores', and 'Agente de pruebas de apps (PREVIEW)'.

Servicio Web:

Paso 1: Despliegue de la base de datos

Antes de desplegar la aplicación backend, es fundamental contar con una base de datos accesible desde la nube. En la imagen se muestra la creación y configuración de una base de datos SQL en Azure, donde se definen parámetros como el nombre, usuario, contraseña y reglas de firewall para permitir conexiones externas.

Azure Database for MySQL: serie de seminarios web en directo: Obtenga información sobre las últimas actualizaciones (con demostraciones) e interactúe directamente con el grupo de productos el segundo miércoles de cada mes. [Suscríbase a nuestro canal de YouTube](#) Hoy!

Información esencial

Suscripción (mover)	: Azure for Students	Punto de conexión	: dentibyback.mysql.database.azure.com
Id. de suscripción	: 6823dbf0-2c22-4b5b-ab0c-33349e5b253d	Inicio de sesión del adm...	: sebas
Grupo de recursos (mover)	: dentibydb	Configuración	: Con capacidad de ráfaga, B1ms, 1 núcleos virtuales...
Estado	: Stopped	Versión de MySQL	: 8.0 Actualizar
Ubicación	: Canada Central	Zona de disponibilidad	: 3
Etiquetas (editar)	: Agregar etiquetas	Creado el	: 2025-09-20 03:50:56.0615943 UTC

Introducción Propiedades Recomendaciones (2) Supervisión Tutoriales

Puede supervisar el servidor para ayudar a solucionar problemas y optimizar la carga de trabajo. [Más información](#)

Paso 2: Creación de la aplicación web backend

El siguiente paso consiste en crear el recurso de App Service en Azure, que alojará la aplicación backend. En la imagen se observa la configuración del entorno de despliegue, seleccionando el stack tecnológico (por ejemplo, Java/Spring Boot), el grupo de recursos y el nombre de la aplicación.

Introducción

Grupo de recursos (...	: AlertaVital	Dominio predeterminado	: alertavital-aqesfchafwe4d7fk.canadacentral-01.azurewe...
Estado	: En ejecución	Plan de App Service	: ASP-dentify-9fc6 (B1:1)
Ubicación (mover)	: Canada Central	Sistema operativo	: Linux
Suscripción (mover)	: Azure for Students	Comprobación de es...	: Sin configurar
Id. de suscripción	: 6823dbf0-2c22-4b5b-ab0c-33349e5b253d		
Etiquetas (editar)	: Agregar etiquetas		

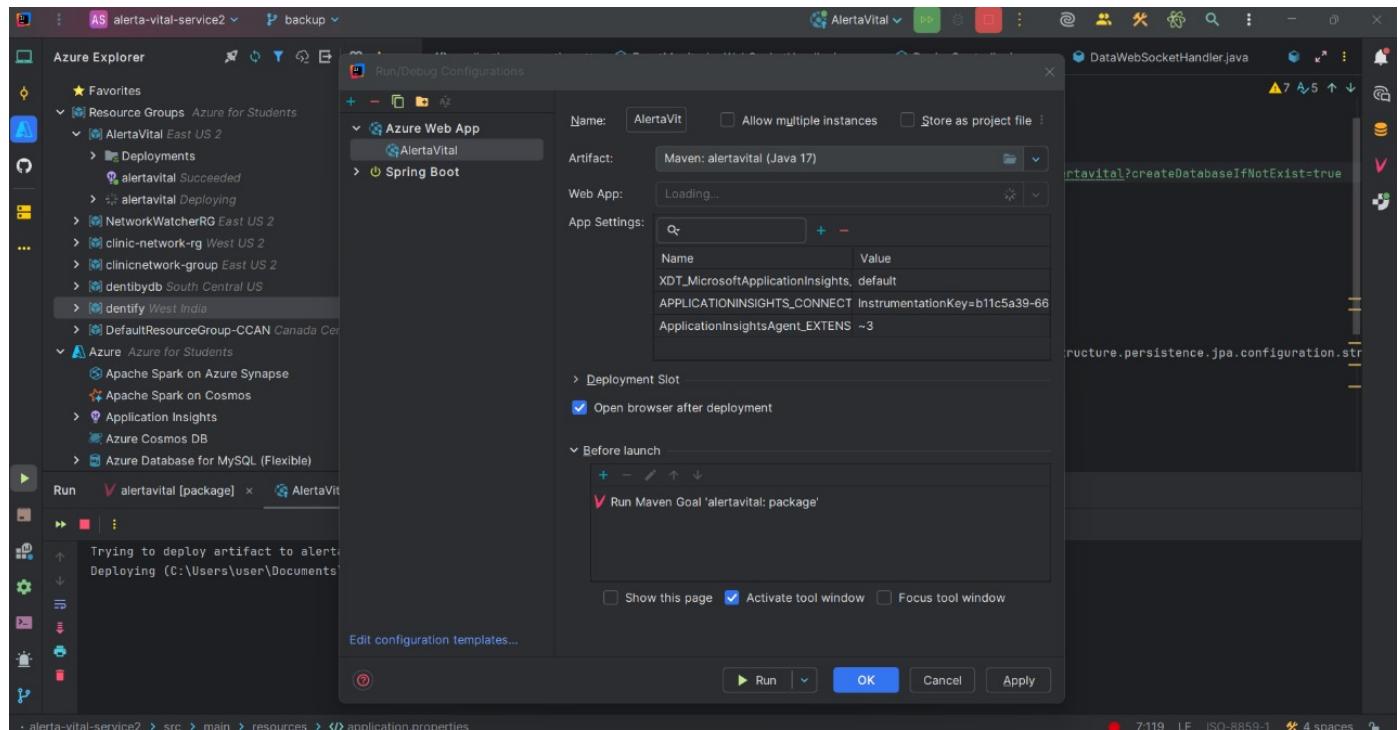
Propiedades Supervisión Registros Capacidades Notificaciones (0) Recomendaciones

Aplicación web

Nombre	alertavital
Modelo de publicación	Código
Pila en tiempo de ejecución	Java 17 SE

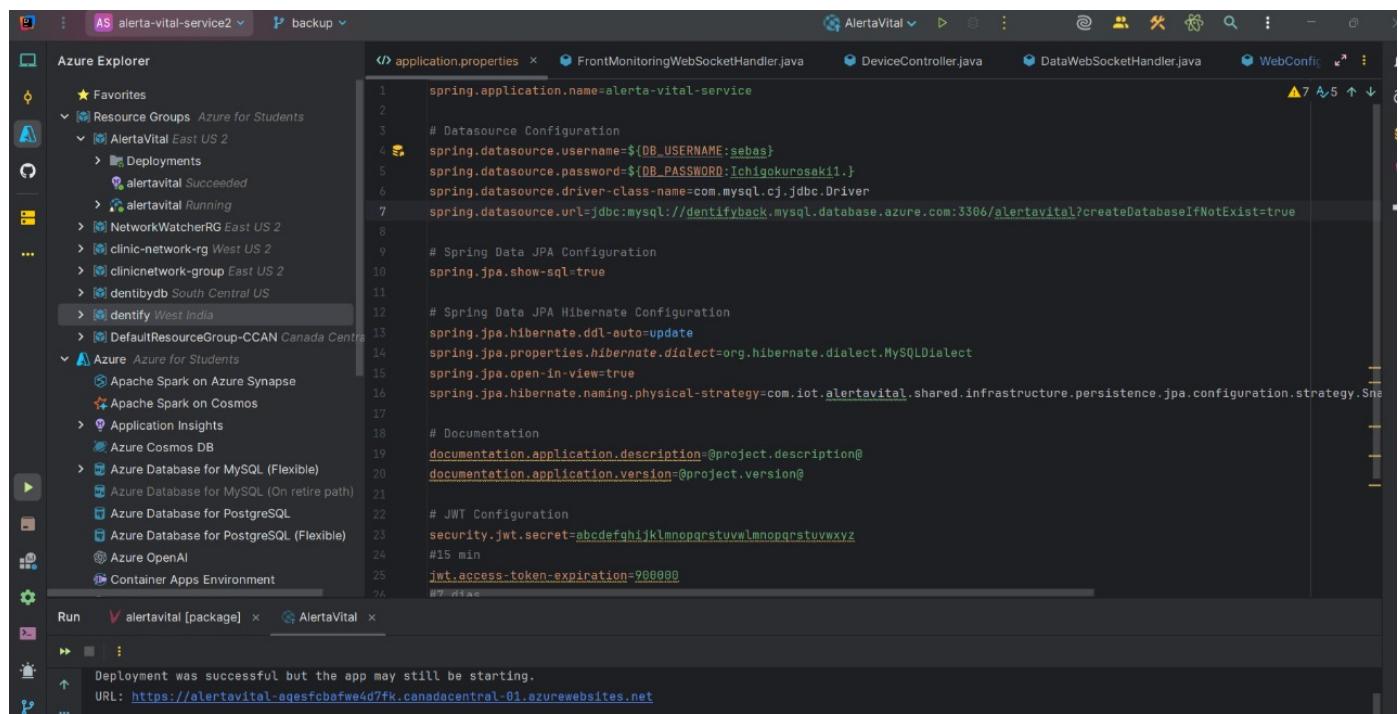
Paso 3: Despliegue en Azure

Una vez configurada la App Service, se procede a desplegar el artefacto generado (por ejemplo, un archivo .jar o .war) desde el entorno local o mediante integración continua. La imagen muestra el panel de Azure donde se gestiona el despliegue, permitiendo cargar el archivo y monitorear el estado de la publicación.



Paso 4: Conexión desde IntelliJ IDEA a Azure

Para facilitar el despliegue y la administración, se puede conectar el entorno de desarrollo (IntelliJ IDEA) directamente con Azure. En la imagen se visualiza la integración, que permite desplegar el backend, configurar variables de entorno y realizar pruebas directamente desde el IDE.



Frontend:

Paso 1: Creación de un proyecto en Firebase

Para el frontend, se utiliza Vercel como plataforma de despliegue debido a su facilidad de integración con repositorios de código y su capacidad para automatizar el proceso de publicación de aplicaciones web modernas. Vercel permite realizar despliegues continuos, lo que significa que cada vez que se realiza un cambio en el repositorio, la aplicación puede actualizarse automáticamente en producción. En la imagen a continuación se muestra el proceso de creación de un nuevo proyecto en Vercel, donde se asigna un nombre representativo y se habilitan los servicios necesarios para el alojamiento web, asegurando así una gestión centralizada y eficiente del ciclo de vida del frontend.

The screenshot shows the Vercel dashboard with the heading "Let's build something new". A search bar at the top left contains the placeholder "Enter a Git repository URL to deploy...". To the right of the search bar is a button labeled "Continue". Below the search bar, there is a section titled "Import Git Repository" which displays a dropdown menu set to "TF-SolucionessloT" and a search bar with the placeholder "Search...". Underneath this, a message says "Front-End-IOT · 48m ago" next to an "Import" button. A note below says "Missing Git repository? Adjust GitHub App Permissions →". To the right, there is a section titled "Clone Template" with four cards: "Next.js Boilerplate", "AI Chatbot", "Vite + React Starter", and "Express.js on Vercel". Each card has a small preview image and a brief description.

Paso 2: Importación y conexión con GitHub

En este paso, se importa el repositorio del proyecto directamente desde GitHub, lo que permite a Vercel acceder al código fuente y mantener sincronización continua con los cambios realizados por el equipo de desarrollo. Posteriormente, se asigna un nombre único al proyecto dentro de Vercel, facilitando su identificación y administración entre múltiples aplicaciones o entornos de despliegue. Esta integración garantiza que el flujo de trabajo de desarrollo y despliegue sea ágil y seguro.

The screenshot shows the Vercel project settings page for a project named "front-end-lot". The top navigation bar includes links for Overview, Deployments, Analytics, Speed Insights, Logs, Observability, Firewall, AI, AI Gateway, Storage, Flags, and Settings (which is the active tab). On the left, a sidebar lists General, Build and Deployment, Domains, Environments, Environment Variables, Git, and Integrations. The main content area is titled "Project Settings" and contains two sections: "Project Name" and "Project ID". The "Project Name" section includes a description and a text input field containing "vercel.com/xyisus-projects/ front-end-lot". The "Project ID" section includes a description and a text input field. A "Save" button is located in the bottom right corner of the "Project Name" section.

Paso 3: Configuración de rama específica para despliegue

En muchos proyectos, es común trabajar con varias ramas (por ejemplo, `main`, `develop`, `feature/*`). Vercel permite seleccionar una rama específica para desplegar en el entorno de producción, lo que resulta útil para pruebas, revisiones o lanzamientos controlados. Para ello, se accede a la sección "Environments" y se selecciona "Production" para editar sus características, asegurando que solo la rama deseada se publique en el entorno final.

The screenshot shows the Vercel Environments page. At the top right is a "Create Environment" button. Below it is a table with three rows:

Name	Branch Tracking	Domains
Production	↳ dev	🌐 front-end-iot-smoky.vercel.app +2
Preview	↳ All unassigned git branches	🌐 No custom domains
Development	> Accessible via CLI	🌐 No custom domains

A continuación, se ingresa el nombre de la rama que se desea desplegar. Esto permite que el equipo tenga control total sobre qué versión del código está activa en producción, facilitando la gestión de versiones y la implementación de nuevas funcionalidades de manera segura.

The screenshot shows the Vercel Project Settings page for the "Production" environment. On the left is a sidebar with various settings: General, Build and Deployment, Domains, Environments (selected), Environment Variables, Git, Integrations, Deployment Protection, Functions, Caches, Cron Jobs, Microfrontends, and Project Members. The main area shows the "Production" environment details:

- Production**: Primary project environment meant to serve qualified, promoted deployments to real users.
- Branch Tracking**: Every commit pushed to the `dev` branch will create a Production Deployment. A dropdown menu shows "Branch is" set to `dev`.
- Auto-assign Custom Production Domains**: Automatically assign custom domains after each deployment triggered by a merge to the production branch or a Vercel CLI deploy with the `--prod` flag. This option is Enabled (blue toggle switch).
- Domains**: Custom domains can be assigned to production from Project domain settings. An "Add Domain" button is available.

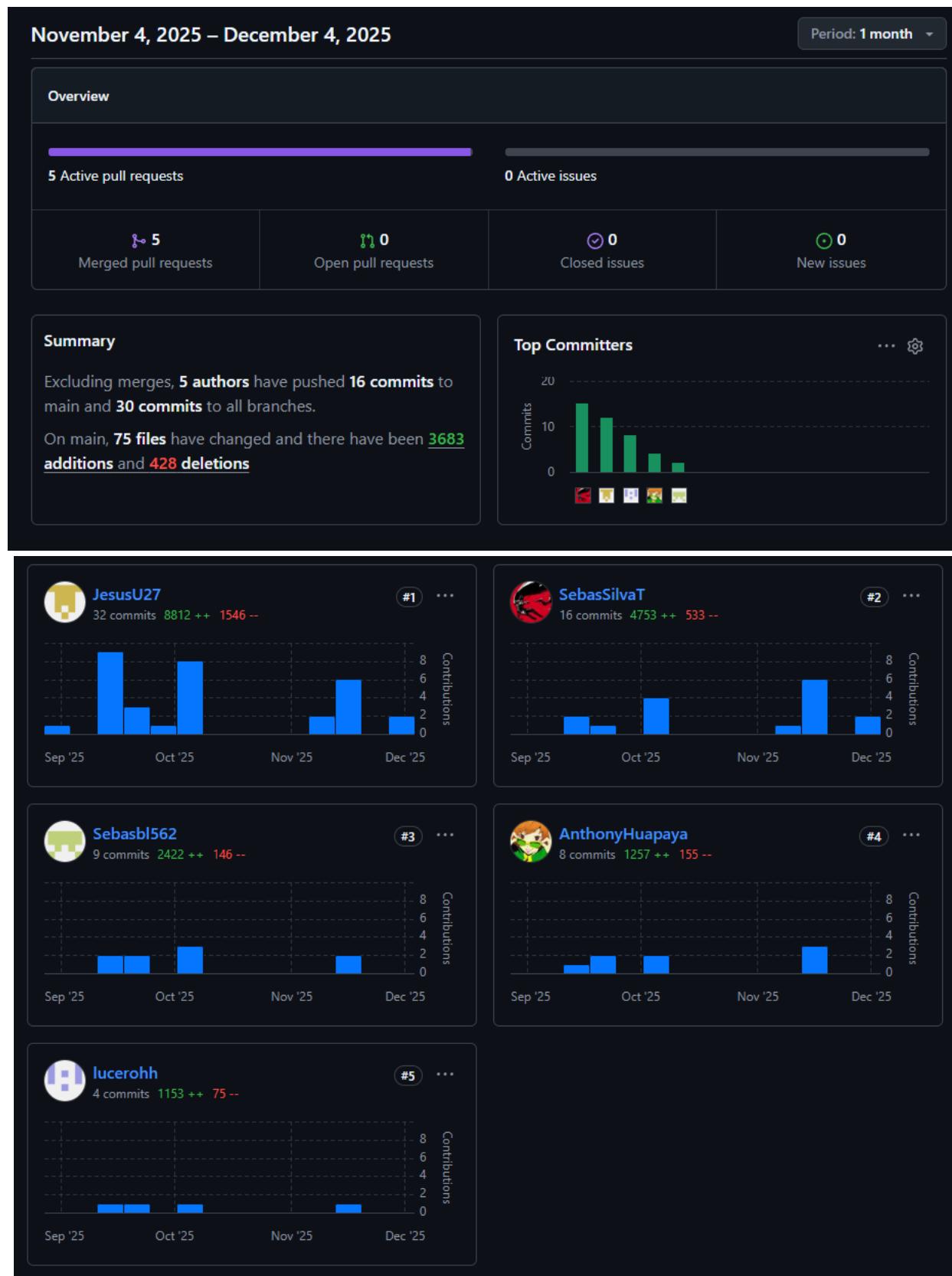
Configuración de variables de entorno

En este paso, se configuran las variables de entorno necesarias para el correcto funcionamiento de la aplicación. Estas variables pueden incluir claves de API, URLs de servicios externos, credenciales y otros parámetros sensibles que no deben estar expuestos en el código fuente. Vercel ofrece una interfaz sencilla para definir y gestionar estas variables, asegurando que cada entorno (desarrollo, staging, producción) tenga la configuración adecuada y segura.

Una vez completada la configuración, Vercel despliega automáticamente la aplicación cada vez que se detectan cambios en la rama seleccionada o en las variables de entorno. Este proceso automatizado reduce el riesgo de errores manuales y acelera la entrega continua, permitiendo que el equipo de desarrollo se enfoque en la mejora constante del producto.

6.2.3.9. Team Collaboration Insights during Sprint.

A continuación, se demuestran los insights en base a este Sprint:



6.3. Validation Interviews.

En esta sección de nuestro proyecto, nos enfocamos en detectar cuáles son los puntos de mejora de nuestra solución IoT para lograr una mayor efectividad en el monitoreo y cuidado de adultos mayores. Esta fase crucial del proyecto implica un diálogo directo con nuestros usuarios principales (adultos mayores y sus cuidadores) para recopilar sus opiniones y sugerencias a través de entrevistas. Así, aseguramos que la plataforma no solo cumpla con los requisitos técnicos, sino que también se adapte a las necesidades diarias y expectativas tanto de los adultos mayores como de sus cuidadores, mejorando la experiencia de monitoreo de signos vitales y la gestión de emergencias.

6.3.1. Diseño de Entrevistas.

Esta sección detalla los objetivos de usuario que guiarán nuestras entrevistas, los cuales son fundamentales para asegurar que nuestro aplicativo, AlertaVital, responda efectivamente a las necesidades reales de los adultos mayores y sus cuidadores. A continuación, se presentan dichos objetivos de usuario:

Introducción:

- ¿Podría indicar su nombre completo, edad, distrito de residencia y ocupación?
- ¿Ha tenido experiencia previa utilizando aplicaciones o dispositivos relacionados al monitoreo de salud?
- ¿Con qué frecuencia utiliza herramientas digitales para llevar el control de la salud del adulto mayor/su salud?

Landing Page:

- ¿Le parece que el diseño de la página inicial resulta atractivo y profesional?
- ¿Considera que el contenido de la página comunica de forma clara el propósito y los beneficios de la aplicación?
- ¿Qué elementos visuales o de información considera más útiles o interesantes?
- ¿Cambiaría o eliminaría algún elemento de la página para hacerlo más claro o atractivo?

Aplicación:

- ¿Las herramientas y secciones de la aplicación le resultan fáciles de identificar y comprender?
- ¿La navegación dentro de la aplicación le parece intuitiva o presenta confusiones?
- ¿Los textos, botones e íconos tienen un tamaño y contraste adecuados para su lectura?

Para el adulto mayor:

- ¿Considera que esta aplicación le ayudaría a llevar un mejor control de sus citas y medicamentos?
- Si usara la aplicación por primera vez sin guía, ¿le resultaría fácil comprender cómo empezar?

Para cuidador o tutor:

- ¿Considera que las herramientas de la aplicación le ayudarían a organizar y hacer seguimiento de la persona bajo su cuidado?
- ¿La interfaz le permite acceder rápidamente a la información médica relevante del adulto mayor?
- ¿Hay alguna funcionalidad que considere necesaria para mejorar las alertas o notificaciones acerca del estado de salud del adulto mayor?

General:

- ¿Qué cambios visuales o funcionales implementaría para mejorar la experiencia de uso?
- ¿Qué tan satisfecho se siente con la experiencia general que ofrece la aplicación?
- ¿Le gustaría seguir utilizando la aplicación en el futuro? ¿Por qué?

6.3.2. Registro de Entrevistas.

Sección	Características del video	Sobre el contenido	Integración y entrega
Validation Interviews	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-3428-AlertaVital-validation-sprint-2 Formato: .mp4 Duración: 09:25 min	Nombre: Fabrisio Belahonia Segmento: Objetivo: Tutor: Edad: 23 Edad: 23 Segmento: Distrito: Nombre: Huachipa	Screenshot:  Link: https://upcedupe-my.sharepoint.com/:v/g/personal/u202220235_upc_edu_pe/IQBvcbEGTVFxtZ9DWmpx5U8kAXxpKY49vqhwONNkZtt0yU?nav=eyJwbGF5YmFja09wdGlvbnMiOnsic3RhcnRUaW1SW5TZWNVbmRzljoxMDg0LjA5NywidGltZXN0YW1wZWRMaW5rUmVmZXJyZXJibmZvIjp7InNjZW5hcmlvIjoiQ2hhcHRlcNoYXJliwiYWRkaXRpb25hbEluZm8iOnsiaXNTaGFyZWRDaGFwdGVyQXV0byl6ZmFsc2V9fx0sInJlZmVycmFsSw5mbly6eylyZWZlcnJhbEFwcCI6IIN0cmVhbVdIYkFwcCI6InJlZmVycmFsVmldyl6IINoYXJlQ2hhcHRlcxpbmsiLCJyZWZlcnJhbEFwcFBsYXRmb3JtjoiV2ViliwicVmZXJyYWxNb2RIloidmldyJ9fQ&e=i9hz

Validation Interviews **Cantidad de videos:** 1

Nomenclatura: upc-pre-202520-1asi0572-3428-AlertaVital-validation-sprint-2

Formato: .mp4

Duración: 17:57 min

Nombre: James Urbano

Segmento Objetivo: Tutor

Edad: 19

Distrito: San Juan de Lurigancho Screenshot:



Link: [Validation Interviews Cantidad de videos: 1](https://upcedupe-my.sharepoint.com/:g/personal/u202220235_upc_edu_pe/IQBvcbEGTVFxTZ9DWmpx5U8kAXxpKY49vqhwONNkZtt0_yU?nav=eyJwbGF5YmFja09wdGlvbMiOnsic3RhcnRUaW1ISW5TZWNVbmRzljo1LCJ0aW1lc3RhbXBIZExpbmtSZWZlcnJckluZm8iOnsic2NlbmFyaW8iOijDaGFwdGVyU2hhcmUiLCjhZGRpdGlvbmsSWS5mbly6eyJpc1NoYXJlZENoYXB0ZXJBdXRvljpmYWxzZX19fSwicmVmZXJyYWxJbmZvlp7InJlZmVycmFsQXBwljoiU3RyZWftV2ViQXBwlwicmVmZXJyYWxWaWV3ljoU2hhcmVdaGfwdGVyTGlauylslnJlZmVycmFsQXBwUGxhdGZvcm0iOijXZWlilCJyZWZlcnJhbE1vZGUoij2aWV3ln19&e=DyD4e4)

Nomenclatura: upc-pre-202520-1asi0572-3428-AlertaVital-validation-sprint-2

Formato: .mp4

Duración: 10:00 min Nombre: Diego Bastidas

Segmento Objetivo: Tutor

Edad: 23

Distrito: Magdalena Screenshot:



Link: [Validation Interviews Cantidad de videos: 1](https://upcedupe-my.sharepoint.com/:g/personal/u202220235_upc_edu_pe/IQBvcbEGTVFxTZ9DWmpx5U8kAXxpKY49vqhwONNkZtt0_yU?nav=eyJwbGF5YmFja09wdGlvbMiOnsic3RhcnRUaW1ISW5TZWNVbmRzljoNjQ5LjgzOCwidGltZXN0YW1wZWRMaW5rUmVmZXJyZXJbmZvlp7InNjZW5hcmlvlijoiQ2hhcHRlcINoYXJlIwiYWRkaXRpb25hbEluZm8iOnsicNTaGFyZWRDaGfwdGVyQXV0byl6ZmFsc2V9fx0slnJlZmVycmFsSW5mbly6eyJyZWZlcnJhbEFwcCl6lIN0cmVhbVdIYkFwcCislnJlZmVycmFsVmldyl6lNoYXJlQ2hhcHRlcKxpBmsiLCJyZWZlcnJhbEFwcFBsYXRmb3JtjoiV2ViliwicmVmZXJyYWxNb2RljojdmlldyJ9fQ&e=7yFDd8)

Nomenclatura: upc-pre-202520-1asi0572-3428-AlertaVital-validation-sprint-2

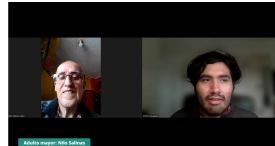
Formato: .mp4

Duración: 14:46 min Nombre: Nilo Salinas

Segmento Objetivo: Adulto mayor

Edad: 67

Distrito: Villa el Salvador Screenshot:



Link: [Validation Interviews Cantidad de videos: 1](https://upcedupe-my.sharepoint.com/:g/personal/u202220235_upc_edu_pe/IQBvcbEGTVFxTZ9DWmpx5U8kAXxpKY49vqhwONNkZtt0_yU?nav=eyJwbGF5YmFja09wdGlvbMiOnsic3RhcnRUaW1ISW5TZWNVbmRzljoyMjU2LjY5LCJ0aW1lc3RhbXBIZExpbmtSZWZlcnJckluZm8iOnsic2NlbmFyaW8iOijDaGFwdGVyU2hhcmUiLCjhZGRpdGlvbmFsSW5mbly6eyJpc1NoYXJlZENoYXB0ZXJBdXRvljpmYWxzZX19fSwicmVmZXJyYWxJbmZvlp7InJlZmVycmFsQXBwljoiU3RyZWftV2ViQXBwlwicmVmZXJyYWxWaWV3ljoU2hhcmVdaGfwdGVyTGlauylslnJlZmVycmFsQXBwUGxhdGZvcm0iOijXZWlilCJyZWZlcnJhbE1vZGUoij2aWV3ln19&e=L3Qubo)

Nomenclatura: upc-pre-202520-1asi0572-3428-AlertaVital-validation-sprint-2

Formato: .mp4

Duración: 06:29 min Nombre: Rosa Valera

Segmento Objetivo: Adulto mayor

Edad: 66

Distrito: San Luis Screenshot:



Link: [Validation Interviews Cantidad de videos: 1](https://upcedupe-my.sharepoint.com/:g/personal/u202220235_upc_edu_pe/IQBvcbEGTVFxTZ9DWmpx5U8kAXxpKY49vqhwONNkZtt0_yU?nav=eyJwbGF5YmFja09wdGlvbMiOnsic3RhcnRUaW1ISW5TZWNVbmRzljozMTQxLjc4OCwidGltZXN0YW1wZWRMaW5rUmVmZXJyZXJbmZvlp7InNjZW5hcmlvlijoiQ2hhcHRlcINoYXJlIwiYWRkaXRpb25hbEluZm8iOnsicNTaGFyZWRDaGfwdGVyQXV0byl6ZmFsc2V9fx0slnJlZmVycmFsSW5mbly6eyJyZWZlcnJhbEFwcCl6lIN0cmVhbVdIYkFwcCislnJlZmVycmFsVmldyl6lNoYXJlQ2hhcHRlcKxpBmsiLCJyZWZlcnJhbEFwcFBsYXRmb3JtjoiV2ViliwicmVmZXJyYWxNb2RljojdmlldyJ9fQ&e=jTKivs)

Nomenclatura: upc-pre-202520-1asi0572-3428-AlertaVital-validation-sprint-2

Formato: .mp4

Duración: 07:42 min Nombre: Juan Pérez

Segmento Objetivo: Adulto mayor

Edad: 72

Distrito: San Miguel Screenshot:



Link: https://upcedupe-my.sharepoint.com/:g/personal/u202220235_upc_edu_pe/IQBvcbEGTVFxTZ9DWmpx5U8kAXxpKY49vhwONNKZtt0_yU?nav=eyJwbGF5YmFja09wdGlvbnnMiOnsic3RhcnRUaW1ISW5TZWVnbmRzljozNTMxLjcwMSwidGltZXN0YW1wZWRMaW5rUmVmZXJyZXJJbmZvIjp7InNjZW5hcmvljoiQ2hhcHRlcNoYXJliwiYWRkaxRpb25hbEluZm8iOnsic3NTaGFyzWRDaGFwdGVyQXV0byl6ZmFsc2V9fx0slnJlZmVycmFsSW5mbyl6eylyZWZlcnJhbEFwcCI6IIN0cmVhbVdIYkFwcCIslnJlZmVycmFsVmldyI6IINoYXJlQ2hhcHRlcxbmsiLCJyZWZlcnJhbEFwcFBsYXRmb3JtjoiV2ViliwicmVmZXJyYWxNb2RIjoidmlldyJ9fQ&e=nRuuor

6.3.3. Evaluaciones según heurísticas.

App a evaluar AlertaVital - Web Application for

Tareas a evaluar El alcance de esta evaluación incluye la revisión de la usabilidad de las siguientes tareas:

Site Web

ID	Tarea
1	Inicio de sesión
2	Recuperación de contraseña
3	Dashboard de la aplicación
4	Gestión de contactos (listar, ver, agregar, eliminar)
5	Diagrama de los signos vitales
6	Gestión de tratamientos
7	Gestión de síntomas
8	Gestión de alteraciones

Web Landing Page

ID	Tarea
1	Redireccionamiento a la aplicación web
2	Formulario de contactos
3	Direccionamiento de la barra de navegación
4	Información de los testimonios

Escala de severidad

Los errores serán puntuados tomando en cuenta la siguiente escala de severidad.

Nivel Descripción

1	Problema superficial: puede ser fácilmente superado por el usuario o ocurre con muy poca frecuencia. No necesita ser arreglado a menos que exista tiempo.
2	Problema menor: ocurre un poco más frecuentemente o es más difícil de superar. Se le debe asignar prioridad baja para la siguiente versión.
3	Problema mayor: ocurre frecuentemente o los usuarios no pueden resolverlo. Debe corregirse con prioridad alta.
4	Problema muy grave: impide al usuario continuar con el uso de la herramienta. Es imperativo corregirlo antes del lanzamiento.

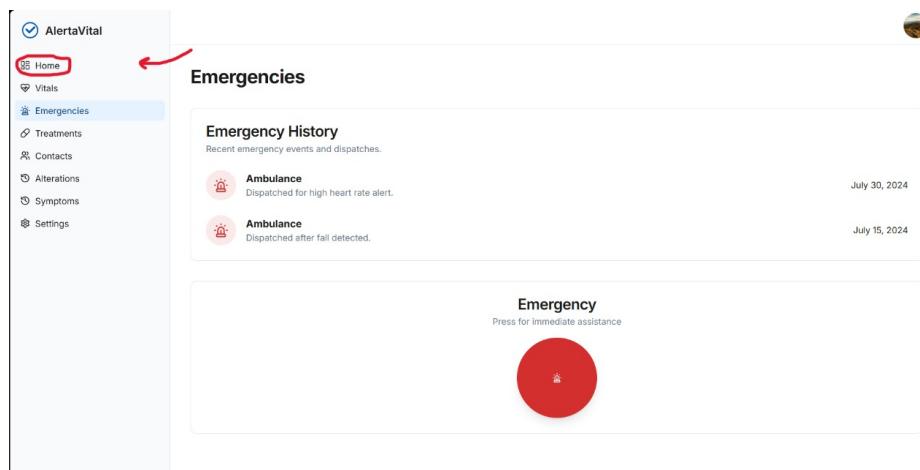
Tabla Resumen

#	Problema	Escala de severidad	Heurística / Principio violado
1	El botón de emergencias "SOS" está en sección secundaria	3	Control y libertad de usuario
2	Aumentar el tamaño de las letras en la barra de navegación	2	Flexibilidad y eficiencia de uso
3	Falta de contraste y diferenciación entre secciones	3	Reconocimiento en lugar de recuerdo
4	Mantiene un color dominante en casi todas las pantallas	3	Diseño estético y minimalista
5	Uso de guías o tutoriales dentro de la aplicación	2	Ayuda y documentación
6	Opción "Forgot your password" no funcional	4	Control y libertad de usuario

Detalle de los problemas

Problema #1: El botón de emergencias "SOS" está ubicado en una sección secundaria.

Severidad: 3 Control y libertad de usuario Problema: El botón de emergencias está ubicado en una sección secundaria, lo que dificulta la acción inmediata. Esto viola la idea de que el usuario debe poder ejecutar funciones críticas de manera rápida y sin barreras.

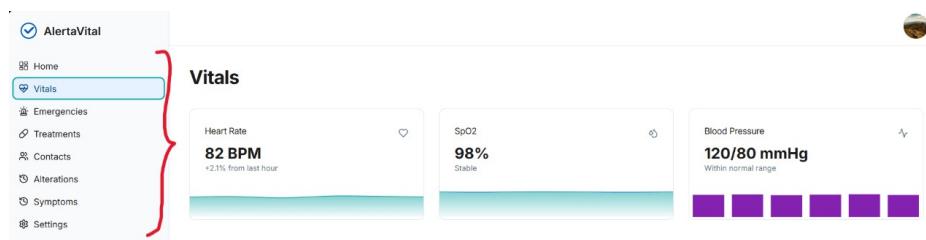


Recomendaciones:

- Implementar la funcionalidad del botón en el dashboard, de esa manera estará al alcance al momento de necesitar pedir una llamada de emergencia.

Problema #2: Aumentar el tamaño de las letras en la barra de navegación.

Severidad: 2 Heurística violada: Flexibilidad y eficiencia de uso Problema: La solicitud de aumentar el tamaño de las letras en la barra de navegación indica un fallo en la legibilidad, que es clave especialmente para usuarios con limitaciones visuales o con estrés en situaciones de emergencia.

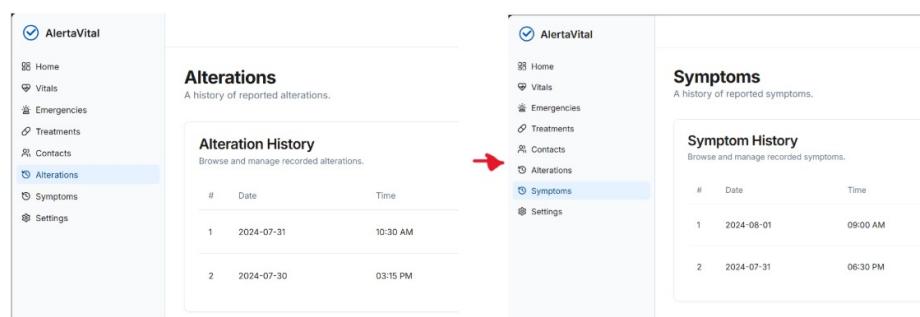


Recomendaciones:

- Mejorar el diseño de la barra lateral, como aumentando más el tamaño de letra para que sea visible para cualquier persona que lo use.

Problema #3: Falta de contraste y diferenciación entre secciones

Severidad: 3 Heurística violada: Reconocimiento en lugar de recuerdo Problema: La falta de contraste y diferenciación entre secciones hace que el usuario tenga que recordar qué parte de la aplicación está viendo, en lugar de reconocerla inmediatamente

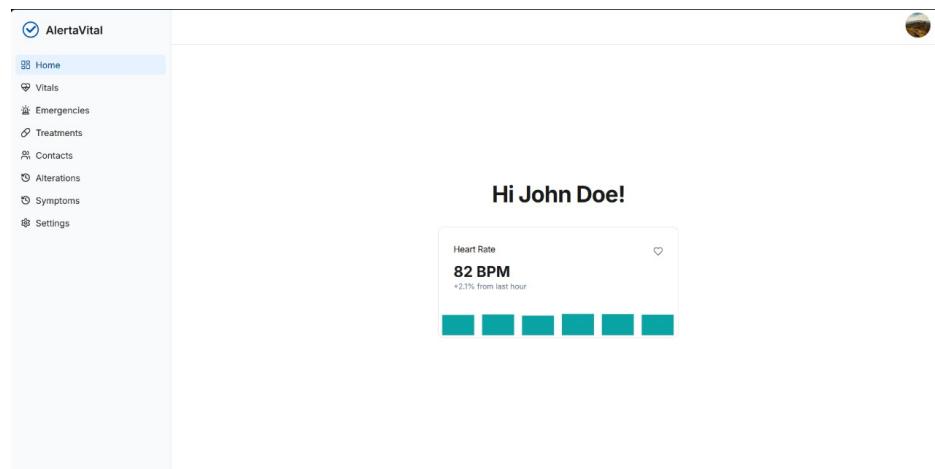


Recomendaciones:

- Mejorar el diseño para que tenga un mejor contraste al momento de cambiar entre distintas secciones de la aplicación.

Problema #4: Mantiene un color dominante en casi todas las pantallas

Severidad: 3 Heurística violada: Diseño estético y minimalista Problema: El diseño mantiene un color dominante en casi todas las pantallas, lo que genera una falta de diferenciación entre secciones, además de los problemas de contraste y jerarquía visual. Esto rompe con la consistencia visual adecuada, ya que no se aplican estándares de contraste ni convenciones de diseño que ayuden a la identificación rápida.

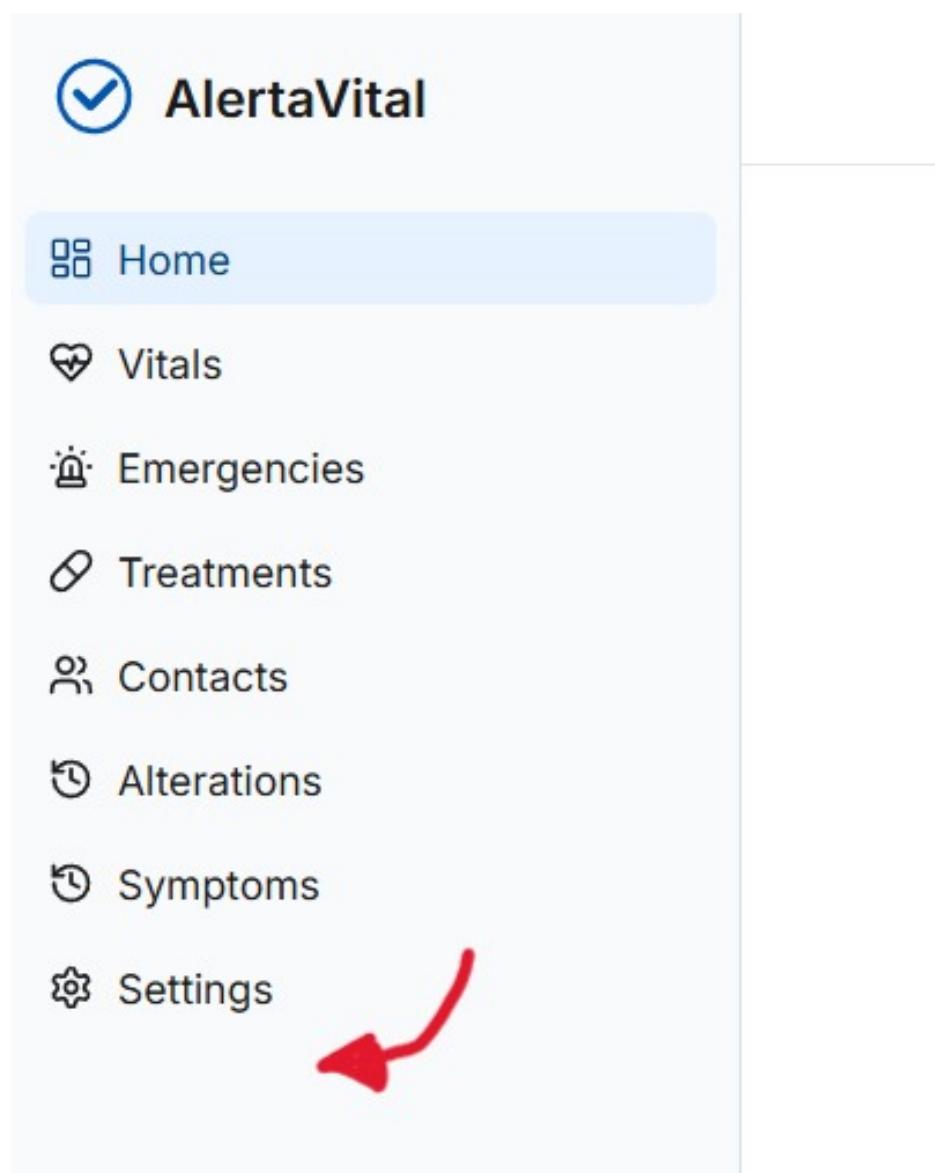


Recomendaciones:

- Mejorar el diseño de acuerdo con un mejor uso de las paletas de colores para que no predomine un mismo color en toda la aplicación.

Problema #5: Uso de guías o tutoriales dentro de la aplicación

Severidad: 2 Heurística violada: Ayuda y documentación Problema: Muchos adultos mayores requieren guías o tutoriales dentro de la aplicación. Aunque la app es fácil de usar, no se evidencia la presencia de ayudas o instrucciones, lo que puede dificultar el aprendizaje de nuevas funciones y reducir la autonomía del usuario.



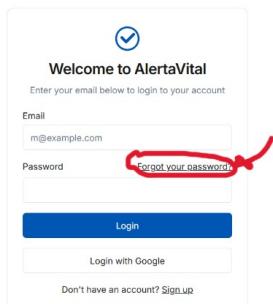
Recomendaciones:

- Añadir una opción de tutorial o guía que pueda permitir a cualquier usuario aprender a cómo usar la aplicación sin dificultades.

Problema #6: Opción "Forgot your password" no funcional

Severidad: 4 Heurística violada: Control y libertad de usuario Problema: En la vista de inicio de sesión, el enlace de "Forgot your password" presenta un comportamiento

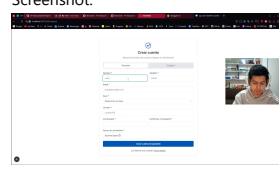
incorrecto. El enlace es clickeable, pero no produce ninguna acción observable para el usuario. No te redirige a la página para ver el tema de la contraseña del usuario. Además, el sistema no provee ninguna señal de que la acción está procesando, ni tampoco muestra mensajes de error si el backend falla. Esto genera un estado de incertidumbre en el usuario, ya que no queda claro si el sistema está trabajando o hubo un error.



Recomendaciones

- Proveer feedback inmediato al usuario.
 - Deshabilitar interacción mientras procesa.
- Validar y manejar errores del backend.

6.4. Video About-the-Product.

Sección	Características del video	Sobre el contenido	Integración y entrega
About The Product	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-3428-The AlertaVital-about-the-product-sprint-3 Formato: .mp4 Duración: 02:58 min	Video de About The Product en base a la aplicación "AlertaVital" desarrollado por Alerta Vital.	<p>Screenshot:</p>  <p>Link: https://upcedupe-my.sharepoint.com/:v/g/personal/u20211g296_upc_edu_pe/IQBIY5mHhggtT577uM1LcouFATT3hgO_Hgu9vqEbA2m1kHA?e=OnRS4U</p>

7. Conclusiones

Durante la unidad 6, correspondiente al cuarto y último entregable (Sprint #3), el equipo se centró en la integración completa del sistema IoT, finalizando el desarrollo de la aplicación móvil, la aplicación Edge y el dispositivo embebido. Se implementó la app móvil utilizando Flutter/React Native, permitiendo la visualización en tiempo real de signos vitales, gestión de emergencias y notificaciones push. La aplicación Edge se optimizó para procesar y transmitir datos desde el dispositivo IoT hacia la nube mediante WebSocket, asegurando comunicación bidireccional y detección local de anomalías. El dispositivo IoT, basado en ESP32 y sensores MAX30102/DS18B20, fue ensamblado y programado para capturar lecturas de frecuencia cardíaca, SpO₂ y temperatura, enviando los datos al sistema central. Todo el flujo, desde el sensor hasta la visualización en la app móvil, fue validado con pruebas end-to-end y entrevistas de usuario, garantizando la robustez, usabilidad y alineación con los objetivos del proyecto. El despliegue y la documentación.

Video About The team:

Sección	Características del video	Sobre el contenido	Integración y entrega
About The Team	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-3428-The AlertaVital-about-the-team Formato: .mp4 Duración: 02:58 min	Video de About The Product en base a la aplicación "AlertaVital" desarrollado por Alerta Vital.	<p>Screenshot:</p>  <p>Link: https://upcedupe-my.sharepoint.com/:v/g/personal/u20211g296_upc_edu_pe/IQDISeVph4nT7GhlxijAthAQKJtjixvC7S4KvPCTTw9ys?e=rZIWMC</p>

8. Bibliografía

Instituto Nacional de Estadística e Informática. (2023). *Perú: Estimaciones y proyecciones de población, 1950-2070*. https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1743/Libro.pdf

Ministerio de Salud. (2023). *Enfermedades no transmisibles y transmisibles - ENDES 2023*. Instituto Nacional de Estadística e Informática.

https://proyectos.inei.gob.pe/endes/2023/SALUD/ENFERMEDADES_ENDES_2023.pdf

Organización Panamericana de la Salud. (2023). *La situación de los cuidados a largo plazo en América Latina y el Caribe*.

https://iris.paho.org/bitstream/handle/10665.2/57356/9789275326879_spa.pdf?sequence=1

9. Anexo