



**Universidad Peruana de Ciencias Aplicadas**

**Carrera de ingeniería de Software**

**Ciclo 202520**

1ASI0572 - Desarrollo de Soluciones IOT

**NRC:** 3443

**Profesor:** Velásquez Núñez, Angel Augusto

**Informe de Trabajo Final**

**Startup:** Los Parkers

**Producto:** CargaSafe

**Relación de integrantes**

<b>Código</b>	<b>Nombre</b>
u20201c410	Garro Vega, Marcelo Fabian
u202113324	Sanchez Ignacio, Jefrey Martin
u20211c273	Aliaga Pimentel, George Arturo
u202113640	Bernardo Eusebio Alessandro Joaquin
u202019577	Chirinos Arevalo Daniel Rodrigo
u202214869	Vera Nuñez Nicolas Alejandro

**Setiembre 2025**

**Registro de Versiones del Informe**

<b>Versión</b>	<b>Fecha</b>	<b>Autor</b>	<b>Descripción de modificación</b>
----------------	--------------	--------------	------------------------------------

Versión	Fecha	Autor	Descripción de modificación
1.0	20/09/2025	Todos los participantes	Se entregó una primera entrega del informe con los siguientes apartados: Carátula, Registro de Versiones, Project Report Collaboration Insights, Contenido, Student Outcome, Capítulo I: Introducción, Capítulo II: Requirements Elicitation & Analysis, Capítulo III: Requirements Specification, Capítulo IV: Solution Software Design y Bibliografía.

## Project Report Collaboration Insights

Repositorios:

- Project Report: <https://github.com/Los-Parkers-IoT/LosParkers-report>

### Entrega TB1

Para la entrega del TB1 se realizaron las actividades necesarias para completar los capítulos I, II, III y IV, resaltando la importancia de la constancia en el trabajo. Como equipo, mantuvimos una frecuencia adecuada de commits y actualizaciones, proyectada como óptima para el desarrollo futuro, y se incluyen en el informe las evidencias de los cambios efectuados.



[Colaboración de repositorios] Evidencia de colaboraciones del repositorio para la entrega TB1 en el informe.

# Contenido

---

## *Tabla de contenidos*

- Registro de Versiones del Informe
- Project Report Collaboration Insights
- Student Outcome
- Capítulo I: Introducción
  - 1.1. Startup Profile
    - 1.1.1. Descripción de la Startup
    - 1.1.2. Perfiles de integrantes del equipo
  - 1.2. Solution Profile
    - 1.2.1. Antecedentes y problemática
    - 1.2.2. Lean UX Process
      - 1.2.2.1. Lean UX Problem Statements
      - 1.2.2.2. Lean UX Assumptions
      - 1.2.2.3. Lean UX Hypothesis Statements
      - 1.2.2.4. Lean UX Canvas
  - 1.3. Segmentos objetivo
    - 1.3.1 Segmento 1: Empresas Clientes
    - 1.3.2 Segmento 2: Clientes Finales
- 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
- Capítulo III: Requirements Specification
  - 3.1. User Stories
  - 3.2. Product Backlog
  - 3.3. Impact Mapping

- 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
      - 4.1.3.2. Software Architecture Container Level Diagrams
      - 4.1.3.3. Software Architecture Deployment Diagrams
  - 4.2. Tactical-Level Domain-Driven Design
    - 4.2.1. Bounded Context: Identity and Access Management
      - 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: Subscriptions and Billing
      - 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: Alerts & Resolution
      - 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: Real-Time 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: Trip Management
  - 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: Fleet Management
  - 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
- 4.2.7. Bounded Context: Profile and Preferences Management
  - 4.2.7.1. Domain Layer
  - 4.2.7.2. Interface Layer
  - 4.2.7.3. Application Layer
  - 4.2.7.4. Infrastructure Layer
  - 4.2.7.5. Bounded Context Software Architecture Component Level Diagrams
  - 4.2.7.6. Bounded Context Software Architecture Code Level Diagrams
    - 4.2.7.6.1. Bounded Context Domain Layer Class Diagrams
    - 4.2.7.6.2. Bounded Context Database Design Diagram
- 4.2.8. Bounded Context: Visualization Analytics
  - 4.2.8.1. Domain Layer
  - 4.2.8.2. Interface Layer
  - 4.2.8.3. Application Layer
  - 4.2.8.4. Infrastructure Layer
  - 4.2.8.5. Bounded Context Software Architecture Component Level Diagrams
  - 4.2.8.6. Bounded Context Software Architecture Code Level Diagrams
    - 4.2.8.6.1. Bounded Context Domain Layer Class Diagrams
    - 4.2.8.6.2. Bounded Context Database Design Diagram

## Student Outcome

---

El curso contribuye de manera directa al desarrollo y cumplimiento del Student Outcome 5 definido por ABET – EAC, asegurando que los estudiantes alcancen las competencias establecidas en dicho resultado.

Criterio: 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. En el siguiente cuadro se describe las acciones realizadas y enunciados de conclusiones por parte del grupo, que permiten sustentar el haber alcanzado el logro del ABET – EAC - Student Outcome 5.

<b>Criterio Específico</b>	<b>Acciones Realizadas por Miembro (por Avance)</b>	<b>Conclusiones del Equipo</b>
Trabaja en equipo para proporcionar liderazgo conjunto	<p>Marcelo Fabian Garro Vega TB1: Durante este avance del proyecto, participé activamente en reuniones presenciales y virtuales, donde contribuí con ideas clave para definir el flujo de negocio, fomentando así la claridad y el consenso entre todos los miembros del equipo. También participé activamente en la elaboración de la arquitectura de software y en el diseño táctico siguiendo los principios de Domain Driven Design.</p> <p>Jefrey Sanchez Ignacio TB1: Durante este avance contribuí a mantener un entorno colaborativo al apoyar en la documentación del Domain Layer, describiendo de manera clara las entidades, value objects y servicios de dominio. Además, participé en la definición de metas de la iteración y en la organización de tareas, lo que permitió que el equipo trabajara con orden, alcanzara los objetivos propuestos y mantuviera consistencia con las guías establecidas por el líder del grupo.</p> <p>Daniel Rodrigo Chirinos Arevalo TB1: Participé en reuniones virtuales y presenciales, aportando ideas sobre cómo abordar el proyecto y colaborando con mis compañeros para tomar decisiones clave respecto al caso propuesto. Además, elaboré de manera colaborativa los user stories y los diagramas de clase y C4. Para ello, realizamos reuniones de planificación y coordinación para asegurar el acuerdo sobre el desarrollo de estos diagramas.</p> <p>Alessandro Joaquin Bernardo Eusebio TB1: Durante este avance del proyecto, participé activamente en reuniones presenciales y virtuales, aportando ideas clave para definir el flujo de negocio y la estructura de la solución. Colaboré en la elaboración de la arquitectura de software y en el diseño táctico bajo los principios de Domain Driven Design, asegurando claridad y consenso entre los miembros del equipo. Además, contribuí en la documentación de los artefactos de usuario y en la organización de tareas, promoviendo un entorno colaborativo y alineado con los objetivos del proyecto.</p>	

Criterio Específico	Acciones Realizadas por Miembro (por Avance)	Conclusiones del Equipo
	<p>Vera Nuñez Nicolas Alejandro</p> <p>TB1:</p> <p>Participé de manera activa en reuniones virtuales y presenciales, aportando propuestas relevantes para la definición de la arquitectura de software del Fleet Management. Lideré la elaboración de los diagramas C4 (contexto, contenedores y componentes) y colaboré en el diseño táctico bajo los principios de Domain Driven Design (DDD). Asimismo, contribuí en la identificación y documentación de los Bounded Contexts, garantizando una delimitación clara y consistente del sistema, y elaboré el Bounded Context Canvases, facilitando la comprensión de las responsabilidades y alcances de cada contexto.</p>	
Crea un entorno colaborativo e inclusivo, establece metas, planifica tareas y cumple objetivos.	<p>Marcelo Fabian Garro Vega</p> <p>TB1: Durante el desarrollo del proyecto, me enfoqué en fomentar un entorno colaborativo e inclusivo dentro del equipo, promoviendo la participación activa de todos los miembros mediante el envío constante de retroalimentaciones constructivas. Contribuí a la estimación de story points y a la planificación de user y technical stories, considerando el valor que estas aportan al negocio. Para ello, tomé en cuenta las opiniones de mis compañeros, lo cual permitió definir metas claras, planificar de manera efectiva y cumplir con los objetivos propuestos bajo los principios de Domain Driven Design.</p>	
	<p>Jefrey Sanchez Ignacio</p> <p>TB1:</p> <p>Durante este avance contribuí a mantener un entorno colaborativo al apoyar en la documentación del Domain Layer, describiendo de manera clara las entidades, value objects y servicios de dominio. Además, participé en la definición de metas de la iteración y en la organización de tareas, lo que permitió que el equipo trabajara con orden, alcanzara los objetivos propuestos y mantuviera consistencia con las guías establecidas por el líder del grupo.</p>	
	<p>Daniel Rodrigo Chirinos Arevalo</p> <p>TB1:</p> <p>Estuve enfocado en fomentar un ambiente de colaboración dentro del equipo. Trabajé en conjunto con mis compañeros para definir los objetivos y tareas del proyecto. Colaboré en el análisis competitivo y en la definición de segmentos clave como el de gestión de transportes y clientes finales. Además, me encargué de crear user stories e impact mapping para estos segmentos, asegurando que el desarrollo del producto estuviera alineado con las necesidades del usuario. Cada semana establecí un límite personal para realizar las secciones de cada capítulo, de esta forma me aseguré de que todas las tareas estuvieran</p>	

Criterio Específico	Acciones Realizadas por Miembro (por Avance)	Conclusiones del Equipo
	<p>bien organizadas y distribuidas para cumplir con los objetivos establecidos.</p> <p>Alessandro Joaquin Bernardo Eusebio  TB1: Durante el desarrollo del proyecto, me enfoqué en fomentar un entorno colaborativo e inclusivo, promoviendo la participación activa de todos los miembros mediante retroalimentación constante y constructiva. Contribuí a la estimación de story points y a la planificación de user y technical stories, considerando el valor que aportan al negocio. Participé en la definición de metas y organización de tareas, asegurando que el equipo trabajara con orden y alcanzara los objetivos propuestos bajo los principios de Domain Driven Design.</p>	
	<p>Vera Nuñez Nicolas Alejandro  TB1:  Promoví un entorno colaborativo mediante la coordinación de los artefactos de arquitectura y la integración de los diagramas en el informe. Contribuí en la planificación de tareas técnicas vinculadas a la arquitectura de software, asegurando la consistencia de los entregables y su alineación con los objetivos del proyecto. Además, facilité retroalimentación constructiva, incentivando la participación activa de todos los miembros del equipo.</p>	

## Capítulo I: Introducción

### 1.1. Startup Profile

#### 1.1.1. Descripción de la Startup

Los Parkers es una startup innovadora, especializada en el desarrollo de soluciones tecnológicas de monitoreo y trazabilidad para el sector logístico y de transporte. Surge de la iniciativa de un equipo multidisciplinario de estudiantes de la Universidad Peruana de Ciencias Aplicadas, quienes comparten una visión común: transformar la manera en que se gestionan y se supervisan las cadenas de suministro de productos sensibles, utilizando tecnología IoT.

Los Parkers se distingue por su enfoque centrado en el usuario, trabajando de forma colaborativa con profesionales del ámbito de la logística para diseñar plataformas intuitivas, inteligentes y adaptadas a las necesidades reales de la industria. Su equipo combina conocimientos técnicos de vanguardia en IoT con una comprensión profunda de los desafíos actuales en materia de cadena de frío, trazabilidad de productos y gestión de riesgos.

Entre sus principales productos destaca CargaSafe, una solución integral que permite a las empresas de transporte y a sus clientes monitorear en tiempo real las condiciones de sus cargas, generando reportes y alertas automáticas respaldadas por tecnología inteligente. Los Parkers emplea metodologías ágiles y

tecnologías modernas para garantizar que sus soluciones sean robustas, escalables y capaces de evolucionar con las necesidades del mercado. Además, la empresa ofrece soporte continuo y mejora constante a sus plataformas, priorizando siempre la experiencia del usuario.

**Visión:** La visión de Los Parkers es convertirse en líder global en el desarrollo de soluciones tecnológicas aplicadas a la logística y la cadena de suministro, empoderando a las empresas para que tomen decisiones más eficientes, informadas y seguras.

**Misión:** La misión de Los Parkers es diseñar y desarrollar herramientas digitales innovadoras, accesibles y personalizadas que contribuyan a mejorar la eficiencia, la transparencia y la seguridad en el transporte de mercancías

### 1.1.2. Perfiles de integrantes del equipo

**Nombre y Apellido:**

Marcelo Fabian Garro Vega

**Código:**

u20201c410

**Carrera:**

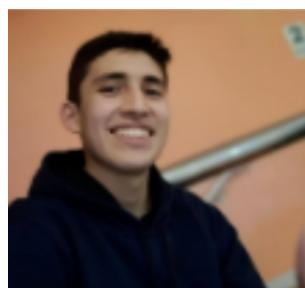
Ingeniería de Software

**Acerca de:**

Soy un estudiante de 22 años apasionado por el desarrollo web y móvil, con un enfoque especial en la experiencia de usuario y el diseño de interfaces (UX/UI). Me especializo en el desarrollo Frontend, siempre buscando crear aplicaciones intuitivas y funcionales. Mi objetivo es aportar tanto en el ámbito creativo como en el técnico, para contribuir al diseño y construcción de soluciones de software que generen impacto real.

**Nombre y Apellido:**

Jefrey Martin Sanchez Ignacio

**Código:**

U202113324

**Carrera:**

Ingeniería de Software

**Acerca de:**

Actualmente cursando el octavo ciclo de mi carrera, Soy una persona responsable, proactiva. Espero aprender mucho del curso y sobretodo de este proyecto.

**Nombre y Apellido:**

George Arturo Aliaga Pimentel

**Código:**

U20211c273



**Carrera:**  
Ingeniería de Software

---

**Acerca de:**  
Me llamo George Arturo Aliaga Pimentel y soy de la carrera de Ingeniería de Software. Estoy cursando el séptimo ciclo. Me considero una persona cooperativa y responsable, lo cual es totalmente requerido para un proyecto grupal. Tengo la meta de ser un gran ingeniero y espero mejorar cada día para lograr un buen desempeño en la carrera.



**Nombre y Apellido:**  
Nicolas Alejandro Vera Nuñez

---

**Código:**  
u202214869

---

**Carrera:**  
Ingeniería de Software

---

**Acerca de:**  
Soy estudiante de la UPC, tengo 21 años. Estoy en la carrera de Ingeniería de Software, ya que, siempre me gustó la tecnología, el análisis de datos y el desarrollo web. Estoy cursando el 7mo ciclo de la carrera y mis habilidades son Angular, Java con Springboot, Docker, Python, HTML y JavaScript. También soy una persona analítica y que encuentra soluciones innovadoras a varios problemas.



**Nombre y Apellido:**  
Daniel Rodrigo Chirinos Arevalo

---

**Código:**  
202019577

---

**Carrera:**  
Ingeniería de Software

---

**Acerca de:**  
Me llamo Daniel Chirinos Arevalo y soy de la carrera de Ingeniería de Software. Mi experiencia en la universidad es bastante normal y me encuentro cursando el sexto ciclo. A lo largo del tiempo, sí que tuve algunas dificultades con algunos cursos, pero logré enfrentarlos. Mi habilidad es ser líder en algunas ocasiones, trabajar con tiempo y lo más importante, ser siempre amistoso con el equipo.

**Nombre y Apellido:**  
Bernardo Eusebio Alessandro Joaquin

---

**Código:**



u202113640

**Carrera:**

Ingeniería de software

**Acerca de:**

Soy Alessandro Joaquin Bernardo Eusebio, estudiante de la carrera de Ingeniería de software. Actualmente estoy bastante orgulloso del progreso que tengo actualmente aprendiendo tanto dentro de la facultad como en mi experiencia autónoma y en el trabajo. Me inclino más a ser un analista de datos y ampliar mis conocimientos sobre Machine Learning, modelos predictivos y AWS.

## 1.2. Solution Profile

### 1.2.1. Antecedentes y problemática

#### What (¿Qué?)

El transporte de mercancías sensibles a la temperatura y las condiciones ambientales sigue siendo un desafío logístico global. Una falla en la cadena de frío o la falta de visibilidad en el recorrido de una carga puede resultar en un deterioro significativo del producto. Según estudios recientes, la industria de alimentos y bebidas en Estados Unidos reportó pérdidas de más de \$6 millones de dólares anuales por empresa en 2022, principalmente debido a daños, pérdidas y retrasos en el transporte (Flock Freight, 2023). En el sector farmacéutico, se proyecta que el mercado de logística de cadena de frío para productos como vacunas y biológicos crecerá en \$12.81 mil millones de dólares entre 2024 y 2028, lo que subraya la criticidad de mantener la integridad de estos productos durante su tránsito (Technavio, 2024).

#### Who (¿Quién?)

Este problema impacta a dos actores clave:

1. Empresas transportistas y de logística: Enfrentan el riesgo financiero de pérdidas de carga, reclamos de clientes y el daño a su reputación por entregas fallidas.
2. El cliente final: Se ve afectado al recibir productos en mal estado, caducados o, en el caso de medicamentos, que han perdido su efectividad, lo cual representa un riesgo para la salud y la seguridad.

#### Where (¿Dónde?)

La problemática se manifiesta a lo largo de toda la cadena de suministro, desde el almacén de origen hasta la entrega final. Es particularmente crítica en los tramos de larga distancia (transporte terrestre, aéreo y marítimo) y en los "puntos de transferencia" entre diferentes vehículos o almacenes, donde la supervisión manual es más difícil. La adopción de tecnologías de monitoreo es una tendencia global que se acelera en mercados con infraestructura logística desarrollada y una creciente demanda de comercio electrónico.

#### When (¿Cuándo?)

La necesidad de visibilidad en tiempo real se ha intensificado desde la pandemia de COVID-19, la cual puso de manifiesto la vulnerabilidad de las cadenas de suministro. El aumento del transporte de productos médicos y la expectativa de los consumidores por entregas rápidas y transparentes han impulsado la demanda de soluciones tecnológicas. Hoy en día, la mayoría de los clientes esperan poder rastrear sus pedidos en tiempo real, lo que convierte la visibilidad de la carga en un estándar de mercado, no solo una ventaja competitiva (Perfect Planner, 2025).

## Why (¿Por qué?)

La principal causa de esta problemática es la falta de información oportuna. Las empresas no tienen acceso a datos críticos sobre la temperatura, ubicación o condiciones de su carga en el momento en que ocurren las desviaciones. Esto impide la toma de acciones correctivas inmediatas, como ajustar el termostato de un camión, cambiar una ruta o notificar al cliente sobre un posible retraso. Sin esta visibilidad, los problemas solo se descubren al final del trayecto, cuando ya es demasiado tarde para evitar la pérdida del producto.

## How (¿Cómo?)

Actualmente, el monitoreo se realiza con métodos ineficientes o no integrados. Muchas empresas aún dependen de registradores de datos manuales que requieren ser revisados al final del viaje o utilizan múltiples sistemas (GPS para ubicación, sensores para temperatura) que no se comunican entre sí. Esta fragmentación reduce la eficiencia operativa y aumenta el riesgo de errores humanos. La falta de una plataforma integral que centralice toda la información limita la capacidad de las empresas para optimizar sus rutas, gestionar riesgos y, en última instancia, ofrecer un servicio de alta calidad (Bogdanov, 2024).

## How much (¿Cuánto?)

El impacto de la falta de un monitoreo efectivo es funcional, operativo y estratégico. Las empresas pierden tiempo y recursos reubicando información o lidiando con problemas logísticos que podrían haberse evitado. Operativamente, esta deficiencia se traduce en mayores costos de seguro y en gastos asociados al desperdicio de productos. Desde una perspectiva de negocio, esta brecha representa una oportunidad clara para monetizar al ofrecer una solución de valor que mejore la fidelización del cliente y construya una reputación de confiabilidad, lo que constituye una ventaja competitiva en el mercado.

### 1.2.2. Lean UX Process

#### 1.2.2.1. Lean UX Problem Statements

Nuestra solución busca proveer a las empresas de transporte y a sus clientes con una plataforma tecnológica integral que les permita monitorear el estado de sus cargas en tiempo real, garantizando la seguridad, calidad y eficiencia de la cadena de suministro. Buscamos que las empresas no sufren pérdidas económicas o daños a su reputación por falta de control, facilitando exponencialmente la supervisión de sus activos críticos, especialmente cuando se trata de productos sensibles que requieren condiciones específicas como temperatura, humedad, o que son susceptibles a vibración y golpes.

Hemos observado que nuestros usuarios, al no tener una plataforma en la cual puedan monitorear la carga en un solo lugar, tienen un problema para estructurar el control de sus activos y guardarlo en un solo lugar, causando que a través de diferentes plataformas o métodos tradicionales guarden sus diferentes recursos, causando además un desorden e incluso pérdida de algunos de estos datos de temperatura, humedad, vibración y ubicación.

¿Cómo podemos ofrecer una plataforma para que las empresas y sus clientes puedan visualizar de forma organizada la información de la carga, consolidando todos los parámetros críticos?

Hemos notado que las empresas que contratan servicios de transporte desean tener visibilidad completa sobre el estado de sus productos. Esta necesidad se presenta normalmente en industrias donde el producto final es perecedero o de alto valor, y el cliente requiere una verificación constante para asegurar que su mercancía llegará en perfectas condiciones, protegiéndola no solo de cambios de temperatura, sino también de golpes o accidentes.

¿Cómo podemos crear una plataforma que mejore la eficiencia y transparencia para todos los actores de la cadena de suministro, proporcionando visibilidad sobre el estado completo de la carga?

Hemos notado que las plataformas que usan las empresas para el monitoreo de cargas no están diseñadas para la simplicidad, tienen un diseño poco atractivo y cuentan con interfaces poco intuitivas o limitan las acciones que se pueden realizar dentro de la misma para algunos usuarios.

¿Cómo podemos diseñar una solución que sea intuitiva y fácil de usar, sin importar el nivel de conocimiento tecnológico del usuario, a pesar de la complejidad de los múltiples datos que monitoreamos?

### 1.2.2.2. Lean UX Assumptions

Business Assumptions:

1. Creemos que hay una demanda significativa de soluciones de monitoreo accesibles que incluyan múltiples parámetros para el sector de transporte en Latinoamérica.
2. Creemos que las empresas estarán dispuestas a pagar una suscripción mensual por una solución que les permita reducir pérdidas y mejorar la confianza de sus clientes al tener control sobre múltiples variables.
3. Creemos que los dispositivos IoT de bajo costo pueden proporcionar la precisión necesaria para el monitoreo de la cadena de frío, además de vibración, ubicación y energía.
4. Creemos que la facilidad de uso es más importante que las funcionalidades avanzadas para nuestro mercado objetivo.
5. Creemos que un modelo de suscripción mensual es más atractivo que la compra de licencias perpetuas para nuestros usuarios.
6. Creemos que la inclusión de parámetros adicionales como humedad, vibración y volcado aumentará el valor percibido de nuestra solución.

User Assumptions:

1. Nuestros usuarios principales son gerentes de operaciones y conductores de empresas de transporte de productos sensibles.
2. Los usuarios necesitan alertas inmediatas cuando se rompe la cadena de frío, hay vibraciones excesivas, o se detecta un volcado para tomar acciones correctivas.

3. Los usuarios quieren dashboards simples e intuitivos que muestren el estado completo de todos sus vehículos en una sola pantalla.
4. Los usuarios valoran la generación automática de reportes que incluyan todos los parámetros monitoreados para sus clientes.
5. Los usuarios prefieren soluciones que no requieran capacitación técnica extensiva.

#### Feature Assumptions:

1. Creemos que el monitoreo en tiempo real de temperatura, humedad, vibración y ubicación es la funcionalidad más crítica.
2. Creemos que las alertas automáticas mediante la aplicación son fundamentales para eventos críticos como volcado y baja energía.
3. Creemos que la gestión de múltiples vehículos y activos desde una sola plataforma es fundamental.
4. Creemos que los reportes históricos son necesarios para el cumplimiento regulatorio y la trazabilidad.
5. Creemos que la integración con dispositivos IoT existentes ampliará nuestro mercado potencial.

#### 1.2.2.3. Lean UX Hypothesis Statements

##### Hypothesis 1:

**Creemos que** implementar un sistema de alertas en tiempo real por incumplimiento de parámetros de temperatura, humedad y volcado **para** empresas de transporte **resultará en** una reducción del 30% en pérdidas de productos y un aumento del 25% en la satisfacción del cliente. *Sabremos que esto es cierto cuando veamos una reducción mensurable en reportes de incidentes y recibamos feedback positivo de al menos el 80% de los usuarios.*

##### Hypothesis 2:

**Creemos que** ofrecer dashboards intuitivos con visualización del estado completo de la flota (temperatura, humedad, vibración, ubicación) **para** gerentes de operaciones **resultará en** una mejora del 40% en la eficiencia operativa y tiempo de respuesta ante incidentes. *Sabremos que esto es cierto cuando observemos una reducción en el tiempo promedio de respuesta a alertas de más de 4 horas a menos de 30 minutos.*

##### Hypothesis 3:

**Creemos que** implementar un modelo de suscripción flexible con diferentes niveles de servicio **para** PYMEs de transporte **resultará en** una adopción del 15% del mercado objetivo en el primer año. *Sabremos que esto es cierto cuando alcancemos 150 empresas suscritas activamente usando la plataforma.*

##### Hypothesis 4:

**Creemos que** proporcionar reportes automatizados y trazabilidad completa **para** clientes finales de servicios de transporte **resultará en** un aumento del 20% en la retención de clientes de nuestros usuarios. *Sabremos*

que esto es cierto cuando nuestros usuarios reporten una reducción en cancelaciones de contratos y solicitudes de nuevos servicios.

#### 1.2.2.4. Lean UX Canvas

<b>1. BUSINESS PROBLEM</b>  Las empresas de transporte de productos sensibles necesitan soluciones accesibles para monitorear múltiples parámetros de la carga (temperatura, humedad, vibración, etc.) y evitar pérdidas económicas por incumplimiento de parámetros.	<b>2. SOLUTION IDEAS</b>  <ul style="list-style-type: none"> <li>• Plataforma IoT de monitoreo en tiempo real (múltiples sensores)</li> <li>• Sistema de alertas automáticas (vibración, volcado, energía, etc.)</li> <li>• Dashboard intuitivo de gestión de flota</li> <li>• Reportes automatizados de trazabilidad completa</li> <li>• Modelo de suscripción escalable</li> </ul>	<b>3. BUSINESS OUTCOMES</b>  <ul style="list-style-type: none"> <li>• 150 empresas suscritas en año 1</li> <li>• Reducción 30% pérdidas de clientes</li> <li>• 80% satisfacción de usuarios</li> <li>• Revenue recurrente de \$180K anuales</li> <li>• Expansión a 3 países LATAM</li> </ul>
<b>4. USERS</b>  <b>Primarios:</b> <ul style="list-style-type: none"> <li>• Gerentes de operaciones de transporte</li> <li>• Conductores de vehículos</li> </ul> <b>Secundarios:</b> <ul style="list-style-type: none"> <li>• Clientes finales de servicios de transporte</li> <li>• Administradores de empresas de transporte</li> </ul>	<b>5. USER OUTCOMES &amp; BENEFITS</b>  <b>Empresas de Transporte:</b> <ul style="list-style-type: none"> <li>• Reducción de pérdidas por múltiples factores (temperatura, vibración, etc.)</li> <li>• Mayor confianza de clientes</li> <li>• Cumplimiento regulatorio automatizado</li> <li>• Optimización de rutas y recursos</li> </ul> <b>Clientes Finales:</b> <ul style="list-style-type: none"> <li>• Visibilidad completa del estado de sus productos</li> <li>• Reportes de trazabilidad en tiempo real</li> <li>• Reducción de riesgos operativos</li> </ul>	<b>6. HYPOTHESES</b>  <b>Creemos que</b> las PYMEs de transporte adoptarán una solución de monitoreo multiparamétrico si es accesible y fácil de usar.  <b>Para validar</b> desarrollaremos un MVP con funcionalidades core (temp, humedad, ubicación) y mediremos adoption rate.
<b>7. WHAT'S THE</b>	<b>8. WHAT'S THE LEAST AMOUNT OF WORK</b>	<b>9. ASSUMPTIONS</b>

MOST IMPORTANT THING WE NEED TO LEARN FIRST?	WE NEED TO DO TO LEARN THE NEXT MOST IMPORTANT THING?	
¿Están las PYMEs de transporte dispuestas a pagar por una solución de monitoreo multiparamétrico IoT y cuál es el precio óptimo que maximiza adopción y rentabilidad?	<ul style="list-style-type: none"> <li>• Desarrollar landing page con pricing y funcionalidades</li> <li>• Realizar 20 entrevistas con empresas target</li> <li>• Crear prototipo de dashboard con múltiples datos</li> <li>• Validar willingness to pay con pre-orders</li> </ul>	<ul style="list-style-type: none"> <li>• Demanda existe en mercado LATAM</li> <li>• Precio \$ 25-150/mes es aceptable</li> <li>• Facilidad de uso &gt; funcionalidades avanzadas</li> <li>• Dispositivos IoT low-cost son suficientes</li> <li>• Modelo de suscripción preferido vs. CAPEX</li> </ul>

## 1.3. Segmentos objetivo

### 1.3.1 Segmento 1: Empresas Clientes

Estas empresas, dedicadas a la logística, distribución o producción de bienes sensibles, necesitan un control exhaustivo sobre sus cargas para asegurar la calidad y evitar pérdidas económicas. Su objetivo es tener una visibilidad completa en tiempo real de sus productos, centralizando toda la información en una sola plataforma para optimizar sus operaciones, cumplir con las normativas y generar confianza en sus propios clientes.

#### Características:

- **Rol:** Gerentes o responsables de logística, calidad o distribución.
- **Ubicación:** Empresas ubicadas en zonas con alta actividad logística y acceso a tecnologías de digitalización.
- **Sector de la industria:** Alimentos perecederos, farmacéuticos, químicos, flores y otros productos que requieren condiciones especiales de conservación.

### 1.3.2 Segmento 2: Clientes Finales

Los clientes finales son los consumidores de los productos transportados por las empresas. Su necesidad principal es la transparencia y la seguridad, ya que buscan la tranquilidad de saber que el producto que adquieren ha sido manejado bajo los estándares de calidad correctos. Valoran la capacidad de verificar el estado de su pedido, desde el origen hasta la entrega, a través de una plataforma sencilla y confiable.

#### Características:

- **Edad:** Mayor a 18 años
- **Ubicación:** Lima, Perú
- **Nivel Socioeconómico:** Medio a alto

## Capítulo II: Requirements Elicitation & Analysis

## 2.1. Competidores

### 2.1.1. Análisis competitivo

#### Análisis Competitivo

¿Por qué llevar a cabo este Análisis? Este análisis permite identificar fortalezas, debilidades y oportunidades en el mercado de soluciones IoT para el monitoreo de cadena de frío, de modo que Macetech pueda priorizar características, precios y estrategias de marketing que maximicen su adopción en el mercado peruano y latinoamericano.

#### Sensitech (Thermo King)



#### Frigga (China)



#### Emerson Cargo Solutions



#### CargaSafe



## Perfil

Overview Multinacional estadounidense líder en monitoreo de la cadena de frío con décadas de experiencia. Fabricante global de dispositivos IoT para cadena de frío, con distribución en más de 60 países. División de Emerson Electric dedicada a soluciones de monitoreo de transporte refrigerado. Startup tecnológica latinoamericana que ofrece monitoreo en tiempo real enfocado en la temperatura del transporte de cargas. ¿Qué valor ofrece a los clientes? Ofrece confianza, cumplimiento de normativas globales (FDA, OMS), cobertura mundial y tecnología robusta. Ofrece sensores desechables/reutilizables de bajo costo, fáciles de implementar en transporte. Seguridad y precisión en tiempo real con analítica avanzada para grandes corporaciones. Propuesta accesible y flexible que asegura la conservación de productos críticos, con alertas inmediatas y dashboards intuitivos.

## Perfil de Marketing

Mercado objetivo Multinacionales farmacéuticas, agroexportadoras y grandes retailers. Exportadores agrícolas y farmacéuticos medianos. Corporaciones de alimentos y farmacéuticas multinacionales. Empresas de transporte, agroexportadores medianos, distribuidores locales de alimentos y fármacos. Estrategias de marketing Presencia en ferias globales, contratos con distribuidores y certificaciones internacionales. Marketing digital, distribuidores locales, precios competitivos. Ventas consultivas, certificaciones globales, contratos a largo plazo. Marketing digital, alianzas con cámaras de comercio, programas de suscripción escalables.

## Perfil de Producto

Productos & Servicios Data loggers, sensores IoT, software de análisis predictivo, soporte técnico 24/7. Data loggers, dispositivos de monitoreo en tiempo real, dashboards básicos. Monitoreo en tiempo real, analítica predictiva, dashboards avanzados. Sensores IoT propios o integrados, aplicación web y móvil, dashboards con métricas clave, alertas en tiempo real. Precios y costos Altos; modelo premium con costo por dispositivo y licencias anuales. Muy competitivos; pago por dispositivo + acceso a plataforma. Elevados; modelo enterprise con contratos anuales. Suscripciones flexibles + costo bajo por dispositivo. Canales de distribución Distribuidores autorizados globales, venta directa enterprise, canal online. Marketplace de e-commerce, distribuidores locales, venta directa. Venta directa corporativa, partners certificados, canal enterprise. Venta directa, partnerships con cámaras de comercio, distribuidores especializados en logística.

## Análisis SWOT

### Fortalezas • Reputación global

- Cumplimiento normativo
- Soporte internacional • Precios accesibles
- Disponibilidad masiva • Marca reconocida
- Integración tecnológica avanzada • Accesibilidad y escalabilidad
- Enfoque en empresas de transporte

- Software amigable Debilidades • Alto costo
- Poca flexibilidad para PYMEs • Limitada personalización de software
- Menor soporte local en LATAM • Precio inaccesible para PYMEs
- Implementación compleja • Respaldo de marca frente a multinacionales
- Mercado nicho especializado Oportunidades • Creciente regulación en transporte farmacéutico y alimentario
- Crecimiento del e-commerce y transporte de alimentos • Demanda en mercados regulados (fármacos, vacunas) • Expansión en LATAM donde grandes competidores no tienen presencia fuerte
- Crecimiento del e-commerce y transporte Amenazas • Startups ágiles con precios más bajos en LATAM • Competidores regionales con soluciones más adaptadas • Startups regionales con mejor relación costo-beneficio • Copia rápida de modelo por competidores grandes o locales
- Regulaciones de transporte cambiantes

## 2.1.2. Estrategias y tácticas frente a competidores

- **Precios accesibles y modelo de suscripción flexible**

Plan básico desde \$29/mes por dispositivo con suscripción mensual sin compromisos a largo plazo, contrastando con licencias anuales costosas de Sensitech y Emerson.

- **Soporte local y personalización regional**

Equipo técnico en español con horarios LATAM, dashboards personalizables con métricas locales y cumplimiento de normativas regionales (SENASA, DIGESA).

- **Implementación rápida y sin complejidad técnica**

Configuración plug-and-play en menos de 24 horas versus semanas de implementación de competidores enterprise, con capacitación incluida.

- **Alianzas estratégicas con el ecosistema local**

Partnerships con cámaras de comercio agrícola, asociaciones de transportistas y distribuidores de dispositivos IoT en mercados emergentes.

- **Transparencia de datos y alertas proactivas**

API abierta para integración con sistemas ERP locales, reportes en tiempo real y alertas vía WhatsApp/SMS, ventajas sobre dashboards cerrados de competidores.

## 2.2. Entrevistas

### 2.2.1. Diseño de entrevistas

#### 1. Preguntas generales

- ¿Cuál es tu nombre y cargo?
- ¿Cuántos años tienes?
- ¿En qué sector o industria trabajas? (alimentos, farmacéutica, logística, etc.)

---

#### 2. Preguntas — **Segmento: Empresa (Gestores de transporte)**

##### 1. Proceso actual de monitoreo

- ¿Cómo monitorean actualmente la temperatura durante el transporte de tus productos?

## 2. Herramientas y tecnología

- ¿Qué dispositivos o sistemas utilizas para el control de cadena de frío y por qué los elegiste?

## 3. Gestión de viajes y rutas

- ¿Cómo planificas y registras los viajes de transporte? ¿Qué información consideras esencial?

## 4. Desafíos principales

- ¿Qué problemas enfrentas cuando se rompe la cadena de frío? ¿Cómo impacta en costos y tiempo?

## 5. Alertas y respuesta a incidentes

- ¿Cómo te enteras cuando hay un problema de temperatura? ¿Qué tan rápido puedes responder?

## 6. Reportes y documentación

- ¿Qué tipo de reportes necesitas generar para clientes o autoridades regulatorias?

## 7. Gestión de dispositivos IoT

- Cuéntame sobre tu experiencia gestionando el mantenimiento y configuración de sensores o dispositivos de monitoreo. ¿Qué desafíos has encontrado?

## 8. Características ideales

- Si pudieras diseñar la plataforma perfecta, ¿qué funciones serían imprescindibles para ti?

## 9. Presupuesto y modelo de pago

- ¿Cuál sería tu modelo de pago preferido para este tipo de servicios y qué factores influyen en esa decisión?

---

## 3. Preguntas — Segmento: Clientes Finales (Consumidores finales)

### 1. Experiencia actual de recepción de productos

- Cuéntame cómo verificas actualmente que los productos que compras llegaron en condiciones óptimas de temperatura.

### 2. Confianza y transparencia en proveedores

- Describe tu nivel de confianza en los reportes de temperatura que te proporcionan tus proveedores. ¿Qué factores aumentarían o disminuirían esa confianza?

### 3. Información requerida sobre el transporte

- ¿Qué información consideras más valiosa tener sobre el transporte de tus productos y cómo te ayudaría en tus operaciones?

### 4. Experiencias con productos dañados

- Comparte alguna experiencia que hayas tenido rechazando productos por problemas de cadena de frío. ¿Cómo identificaste el problema y qué impacto tuvo?

## 5. Preferencias de acceso a información

- Describe cómo prefieres recibir y acceder a información sobre tus pedidos. ¿Qué métodos de comunicación funcionan mejor para tu flujo de trabajo?

## 6. Alertas y notificaciones proactivas

- Cuéntame qué tipo de notificaciones durante el transporte de tus productos serían más útiles para ti y en qué momentos las necesitarías.

## 7. Facilidad de uso y comprensión

- Describe la importancia que tiene para ti que la información técnica sea presentada de manera comprensible. ¿Qué características valoras en las interfaces que usas?

## 8. Características más valoradas

- ¿Qué funcionalidades consideras que agregarían más valor a tu proceso de recepción y validación de productos?

## 9. Expectativas sobre tecnología IoT

- ¿Qué beneficios esperas de un sistema de monitoreo IoT para tus compras de productos sensibles a temperatura y qué preocupaciones tienes al respecto?

### 2.2.2. Registro de entrevistas

#### Segmento 1: Empresa

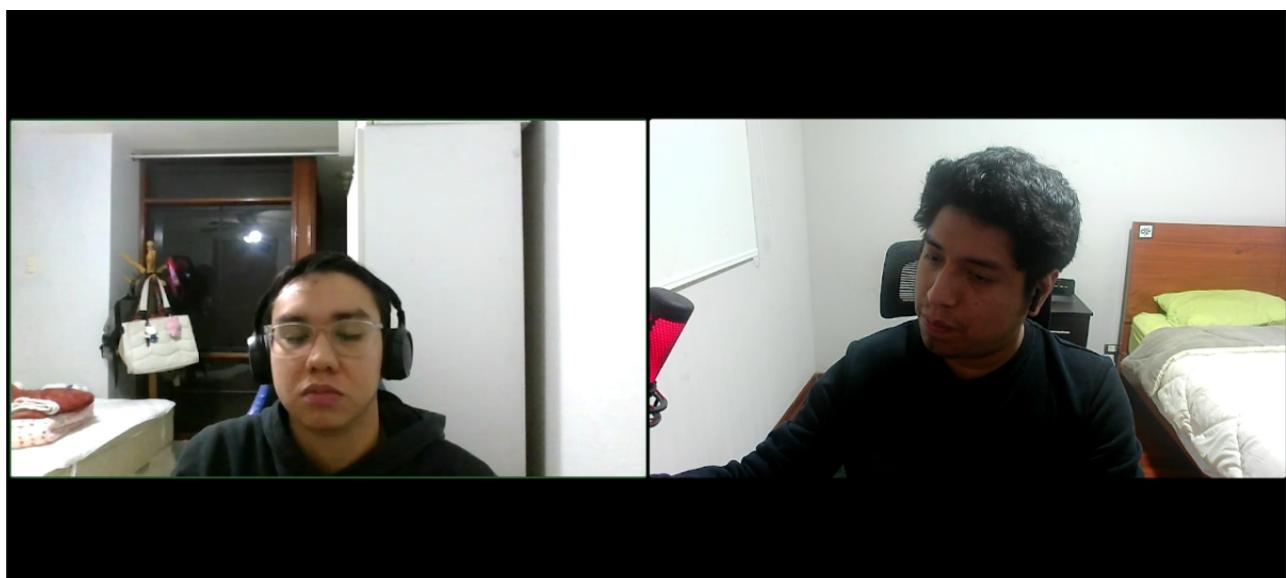
- **Nombre:** Miguel Ruiz
- **Edad:** 28 años
- **Ocupación:** Supervisor
- **Empresa:** Ofertimaq - Distribuidora
- **Enlace:** [Click aquí para ver la entrevista](#)
- **Fecha de entrevista:** 10 de Setiembre del 2025

- **Tiempo inicio - tiempo fin:** 00:00:00 - 00:07:28



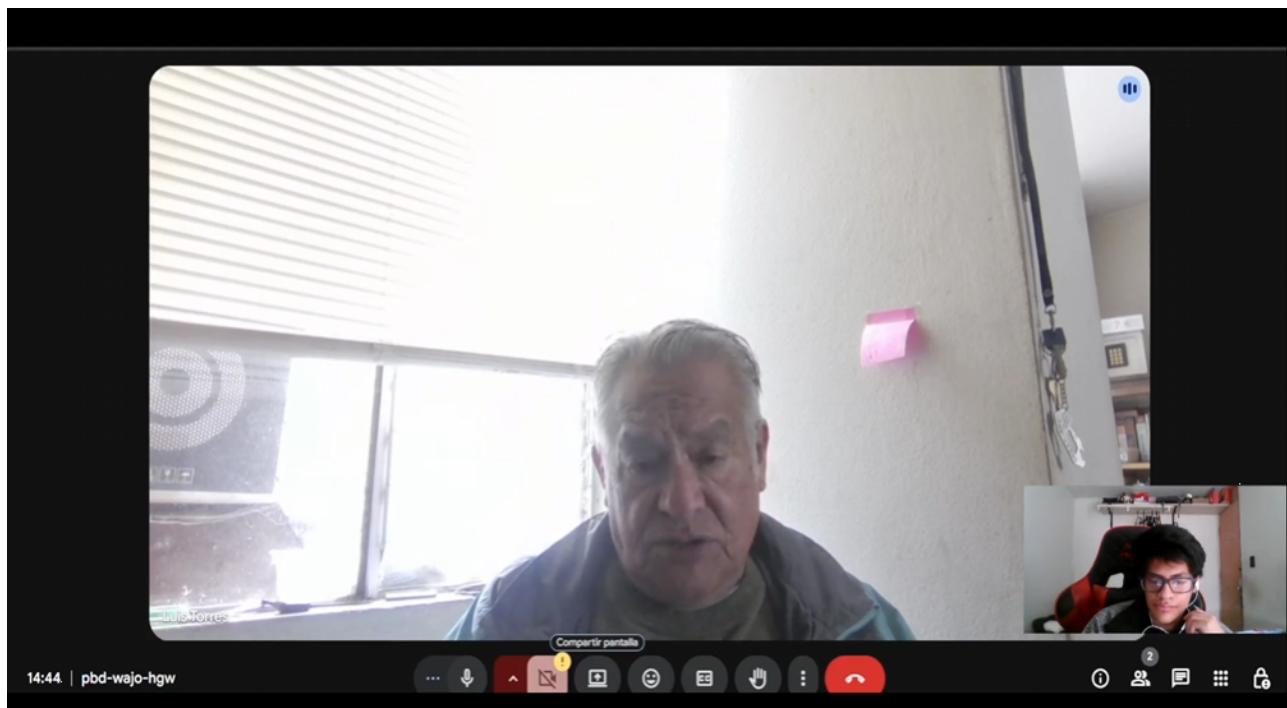
Miguel Ruiz

- **Nombre:** Mario Hinostroza
- **Edad:** 22 años
- **Ocupación:** Operador Logístico
- **Empresa:** Urbano - Distribuidora Ecommerce
- **Enlace:** [Click aquí para ver la entrevista](#)
- **Fecha de entrevista:** 11 de Setiembre del 2025
- **Tiempo inicio - tiempo fin:** 00:07:28 - 00:13:36



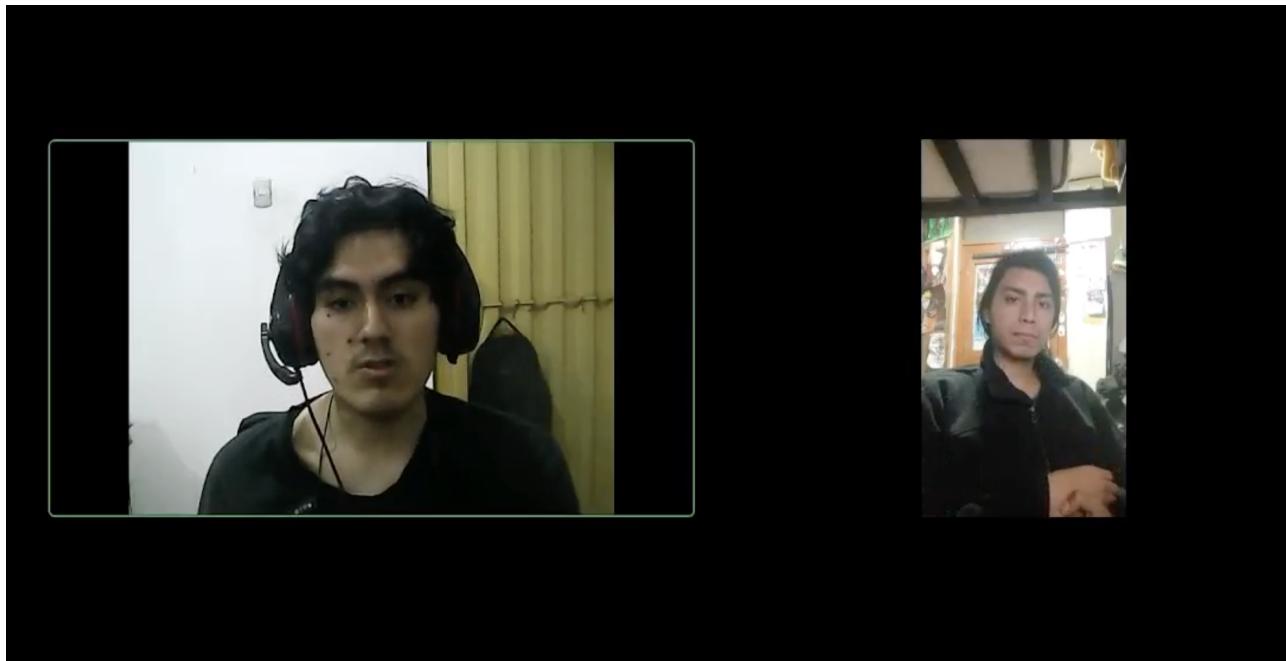
- **Nombre:** Luis Torres Arevalo

- **Edad:** 58 años
- **Ocupación:** Encargado de gestión logística
- **Empresa:** Urbano . Distribuidor de Alimentos
- **Enlace:** [Click aquí para ver la entrevista](#)
- **Fecha de entrevista:** 10 de Setiembre del 2025
- **Tiempo inicio - tiempo fin:** 00:13:36 - 00:20:51



## Segmento 2: Clientes Finales (Consumidores finales)

- **Nombre:** Adrián Zapata
- **Edad:** 23 años
- **Ocupación:** Responsable de parrilla en un negocio de comida rápida
- **Empresa/Sector:** Negocio local de comida rápida / Sector alimentario
- **Enlace:** [URL del video de la entrevista](#)
- **Fecha de entrevista:** 10 de Setiembre del 2025
- **Tiempo inicio - tiempo fin:** 00:20:51 - 00:30:50



- **Nombre:** Brenda Calderon
- **Edad:** 20 años
- **Ocupación:** Trabajadora de medio tiempo en un minimarket, responsable de compras de insumos para refrigeración
- **Empresa/Sector:** Retail alimentario local – Consumo final
- **Enlace:** [URL del video de la entrevista](#)
- **Fecha de entrevista:** 13 de Setiembre del 2025
- **Tiempo inicio - tiempo fin:** 00:31:00 - 00:37:20



- **Nombre:** Gabriel Díaz
- **Edad:** 26 años
- **Ocupación:** Responsable de recepción y control de productos sensibles a temperatura.
- **Empresa/Sector:** Sector de salud
- **Enlace:** [URL del video de la entrevista](#)
- **Fecha de entrevista:** 16 de Setiembre del 2025
- **Tiempo inicio - tiempo fin:** 00:37:20 - 00:43:15



## 2.2.3. Análisis de entrevistas

### Segmento 1: Empresa

- **Nombre:** Miguel Ruiz
- **Entrevistador:** Jefrey Sanchez Ignacio

**Situación Actual:** Miguel Ruiz Sánchez es gestor de transporte de coolers en una empresa logística. Actualmente manejan el monitoreo de temperatura de forma completamente manual - los conductores verifican los coolers durante las paradas usando termómetros digitales y reportan problemas por teléfono. Solo cuentan con GPS básico que no es muy preciso.

**Principales Problemas:** No tienen monitoreo en tiempo real, las alertas dependen de que los conductores llamen cuando algo falla, y esto genera retrasos en las entregas que molestan a los clientes. Cuando un cooler se daña, tienen que enviar otro transporte de emergencia para salvar el producto. Para justificarse ante clientes molestos, solo pueden mostrar reportes de ruta básicos.

**Necesidades Identificadas:** Miguel necesita urgentemente un sistema de monitoreo en tiempo real, alertas automáticas cuando la temperatura llegue a niveles críticos, y reportes sencillos. También mencionó que le gustaría tener un dashboard, aunque no está muy seguro del concepto. Quiere que los clientes puedan verificar remotamente el estado de sus pedidos desde una computadora.

**Disposición de Pago:** Está dispuesto a contratar un servicio que solucione estos problemas, preferiblemente con suscripción mensual o mejor aún, un pago anual para evitar gestiones mensuales. Ve esto como una inversión para recuperar la confianza de los clientes y tener soporte 24/7.

- **Nombre:** Mario Hinostroza
- **Entrevistador:** Marcelo Fabian Garro Vega

Mario Hinostroza, de 22 años, trabaja en el sector de alimentos y actualmente controla la temperatura de la mercadería utilizando el sistema de refrigeración del camión junto a un termómetro portátil. La planificación de viajes la realiza a partir de una hoja de ruta con puntos de entrega y horarios, y lleva un registro manual en una bitácora. Además, entrega reportes que incluyen la hoja de viaje firmada y los datos de temperatura recogidos durante el recorrido.

Entre los principales problemas que enfrenta, destaca que el monitoreo es un proceso manual que demanda mucha atención constante. También existe un alto riesgo de fallas en el motor de frío, lo que provoca pérdida de mercadería y reclamos de los clientes. La detección de incidencias suele depender de su experiencia personal o de lo que muestra el panel del camión, lo que limita la capacidad de respuesta inmediata.

En cuanto a las necesidades identificadas, Mario considera indispensable contar con un sistema automatizado que reduzca la carga manual. Para él sería muy valiosa la integración del panel del camión con una aplicación móvil que alerte en tiempo real sobre variaciones de temperatura, garantizando mayor confiabilidad en la cadena de frío y reduciendo el riesgo de pérdidas y reclamos.

Respecto a la disposición de pago, Mario señala que la decisión recae en la empresa, pero enfatiza que lo fundamental es que el servicio ofrezca un valor tangible. Si la plataforma demuestra un impacto real y positivo en el negocio, la empresa estaría dispuesta a pagar por ella, más allá del costo que implique.

- **Nombre:** Luis Torres Arevalo
- **Entrevistador:** Alessandro Joaquin Bernardo Eusebio

La entrevista con el señor Luis Torres evidencia que la gestión de la cadena de frío se realizaba de manera muy básica y sin tecnología especializada, lo que generaba altos riesgos de pérdida de productos y desconfianza en los clientes. Luis dependía de controles manuales, registros en papel y la responsabilidad del chofer, sin capacidad de reacción ante incidentes durante el trayecto. Esta falta de monitoreo en tiempo real implicaba costos elevados cuando se rompía la cadena de frío y limitaba la capacidad de generar reportes confiables. No obstante, identifica con claridad las funciones clave de una solución ideal: monitoreo en tiempo real, alertas inmediatas y generación automática de reportes, además de preferir un modelo de pago por suscripción mensual que le permita prever costos.

## Segmento 2: Clientes Finales (Consumidores finales)

- **Nombre:** Adrian Zapata
- **Entrevistador:** George Aliaga

Nuestro entrevistado Adrián Zapata, encargado de la parrilla en un negocio de comida rápida, expone una problemática clave en la recepción de productos sensibles a temperatura: la falta de control y visibilidad durante el transporte. Actualmente depende de inspecciones visuales y táctiles, lo que limita la precisión y genera incertidumbre. Señala la importancia del tiempo de entrega, la temperatura constante y la comunicación oportuna, idealmente a través de notificaciones por WhatsApp. También destaca la necesidad de interfaces claras y datos legibles, especialmente en grados Celsius. Su interés en un sistema IoT revela el valor que asigna a la trazabilidad, el monitoreo en tiempo real y la posibilidad de actuar preventivamente ante fallas en la cadena de frío.

- **Nombre:** Brenda Calderon
- **Entrevistador:** Nicolas Vera

La entrevistada Brenda Carolina, compradora frecuente de alimentos online, manifiesta que al recibir productos sensibles a temperatura verifica el empaque, la ausencia de humedad, que los alimentos lleguen fríos al tacto y que la fecha de vencimiento esté visible, ya que ha tenido experiencias negativas como rechazar un lote de yogures en mal estado. Señala que los reportes de temperatura que recibe de los proveedores suelen ser genéricos y poco confiables, por lo que considera fundamental contar con registros más detallados y trazables en tiempo real. Para ella, la información más valiosa incluye la temperatura a lo largo del trayecto, retrasos o eventos que afecten la cadena de frío, y prefiere acceder a estos datos mediante un portal o aplicación con notificaciones breves y claras. Valora interfaces simples con gráficos fáciles de interpretar, así como funcionalidades como alertas en tiempo real, reportes descargables e historial de pedidos, y espera que un sistema IoT le brinde mayor seguridad y control en sus compras, aunque le preocupa el costo de implementación y la confiabilidad técnica de los equipos.

- **Nombre:** Gabriel Díaz
- **Entrevistador:** Daniel Arevalo

El entrevistado Gabriel Díaz, responsable de recepción y control de productos sensibles a temperatura en el sector salud, expone la problemática de confiar únicamente en reportes básicos y revisiones visuales al recibir insumos médicos, lo que genera incertidumbre sobre la cadena de frío. Destaca la necesidad de contar con un monitoreo continuo y alertas en tiempo real para actuar preventivamente, ya que ha debido rechazar pedidos por fallas en la temperatura. Su interés en un sistema IoT refleja la importancia que le asigna a la trazabilidad, la seguridad y la eficiencia, aunque expresa preocupación por los costos y la dependencia de la conectividad.

## 2.3. Needfinding

### 2.3.1. User Personas

- **Segmento: Empresa (Gestores de transporte)**

### User Persona 1: Carlos Méndez



**Responsable** **orientado a resultados**  
**organizado** **pragmático**

**Goals**

- Asegurar que los viajes de transporte cumplan tiempos y parámetros (ej. temperatura).
- Reducir incidencias y pérdidas en la cadena de suministro.
- Monitorear la flota en tiempo real con reportes confiables.
- Optimizar la toma de decisiones mediante dashboards y métricas.

**Motivation**

Incentive	Score
Fear	Medium
Growth	Medium-High
Power	Medium
Social	High

**Frustrations**

- Procesos manuales que generan errores o retrasos.
- Dificultad para monitorear todo en simultáneo.
- Falta de visibilidad de los estados en tiempo real.
- Que el sistema de alertas no sea rápido o efectivo.

**Brands & Influencers**





**Personality**

Introvert	Extrovert
Thinking	Feeling
Sensing	Intuition
Judging	Perceiving

**Bio**

Carlos ha trabajado más de 15 años en logística y transporte. Maneja una flota que mueve productos sensibles (como alimentos y medicamentos), donde mantener la temperatura y tiempos de entrega es crítico. Es exigente con la tecnología: espera precisión y confiabilidad, ya que cualquier falla impacta en su operación.

**Preferred Channels**

Channel Type	Score
Traditional Ads	Medium
Online & Social Media	High
Referral	Medium-High
Guerilla Efforts & PR	Medium

## Carlos Mendoza - Jefe de Logística

El user persona de Carlos representa al gestor experimentado que prioriza la eficiencia operativa y la minimización de riesgos. Muestra la necesidad de herramientas tecnológicas robustas y precisas que le permitan mantener control total sobre la cadena de frío. Su perfil refleja la importancia de la confiabilidad del sistema, ya que cualquier falla puede resultar en pérdidas económicas significativas y problemas regulatorios. Carlos ejemplifica al usuario que valora los datos en tiempo real, reportes detallados y funcionalidades que le permitan tomar decisiones informadas para proteger productos de alto valor.

- **Segmento: Clientes Finales (Consumidores finales)**

## User Persona 2: María Gonzales



Emprendedora    exigente    organizada  
orientada a la calidad

**Goals**

- Asegurar que los insumos lleguen siempre frescos y en condiciones óptimas.
- Evitar pérdidas económicas por productos dañados o fuera de los estándares de calidad.
- Monitorear pedidos y entregas de manera sencilla y en tiempo real.
- Ofrecer a sus clientes finales platos de confianza, respaldados por una cadena de suministro transparente.

**Frustrations**

- Productos en mal estado que generan pérdida y afectan la reputación del negocio.
- Falta de transparencia en el transporte de alimentos perecibles.
- Dificultad para reclamar incidencias o validar cumplimiento de estándares.

**Motivation**

Incentive	Value
Incentive	High
Fear	Medium
Growth	Medium
Power	Medium
Social	Low

**Brands & Influencers**





**Preferred Channels**

Channel	Value
Traditional Ads	Medium
Online & Social Media	Medium
Referral	Low

**Personality**

Dimension	Score
Introvert / Extrovert	Extrovert
Thinking / Feeling	Thinking
Sensing / Intuition	Sensing
Judging / Perceiving	Judging

**Bio**

Maria administra un restaurante reconocido en su ciudad y depende de insumos frescos para mantener la calidad de sus platos. Está acostumbrada a lidiar con proveedores, pero exige que la calidad de sus productos sea transparente y verificable. Valora la transparencia: quiere acceso claro a reportes de temperatura de los insumos que ella solicita, porque cada fallo puede traducirse en clientes insatisfechos y pérdidas.

## María González - Gerente de Compras de Restaurante

El user persona de María representa al consumidor final que valora la transparencia y la calidad en los productos que adquiere para su negocio. Como responsable de compras de un restaurante, necesita la seguridad de que los alimentos que recibe han mantenido la cadena de frío adecuada durante el transporte. Su perfil ilustra la importancia de contar con información clara y accesible sobre el estado de los productos, reportes de cumplimiento fáciles de entender, y la capacidad de verificar la integridad de los alimentos antes de aceptar las entregas. María ejemplifica al usuario que busca confianza y transparencia en el proceso logístico para proteger la reputación de su negocio.

### 2.3.2. User Task Matrix

#### **Segmento: Empresa (Gestores de transporte)**

Tarea	Frecuencia	Importancia
Configurar dispositivos IoT para nuevos vehículos	Media	Alta
Registrar nuevos viajes y establecer parámetros	Alta	Alta
Monitorear temperatura en tiempo real durante viajes	Alta	Alta
Recibir y gestionar alertas de incumplimiento	Media	Alta
Revisar dashboards de estado de flota	Alta	Alta
Generar reportes de viajes para clientes	Media	Alta
Analizar gráficos de incidencias mensuales	Baja	Media
Gestionar suscripciones y pagos del sistema	Baja	Media
Actualizar parámetros de temperatura según producto	Media	Alta
Coordinar respuesta ante emergencias de cadena de frío	Baja	Alta

#### **Segmento: Clientes Finales (Consumidores finales)**

Tarea	Frecuencia	Importancia
Verificar estado de pedidos en tránsito	Alta	Alta
Consultar certificados de cadena de frío	Media	Alta
Revisar reportes de cumplimiento de temperatura	Media	Alta
Rastrear ubicación y progreso de entrega	Alta	Media
Recibir alertas sobre problemas en el transporte	Media	Alta
Validar condiciones del producto al recibirlo	Alta	Alta
Descargar documentación oficial de cumplimiento	Media	Alta
Consultar historial de entregas anteriores	Baja	Media
Reportar productos rechazados por incumplimiento	Baja	Alta
Acceder a soporte para resolver incidencias	Baja	Media

### 2.3.3. User Journey Mapping

#### Journey Map: Carlos Mendoza (Gestor de transporte)

Etapa	Pre-viaje	Configuración	Monitoreo	Gestión de alertas	Post-viaje
Acciones	<ul style="list-style-type: none"> <li>• Planifica ruta y carga</li> <li>• Revisa dispositivos IoT</li> <li>• Define parámetros</li> </ul>	<ul style="list-style-type: none"> <li>• Registra nuevo viaje</li> <li>• Configura temperaturas</li> <li>• Asigna conductor y vehículo real</li> </ul>	<ul style="list-style-type: none"> <li>• Supervisa dashboard</li> <li>• Revisa alertas en tiempo real</li> <li>• Monitorea ubicación</li> </ul>	<ul style="list-style-type: none"> <li>• Recibe notificación de alerta</li> <li>• Evalúa severidad</li> <li>• Coordina respuesta</li> </ul>	<ul style="list-style-type: none"> <li>• Genera reportes</li> <li>• Analiza incidencias</li> <li>• Envía documentación a cliente</li> </ul>
Pensamientos	"Necesito asegurarme de que todo esté configurado correctamente"	"Los parámetros deben ser exactos para este tipo de producto"	"Todo va bien, temperatura estable"	"¿Qué tan grave es esta alerta? ¿Debo actuar inmediatamente?"	"Perfecto, puedo demostrar que cumplimos todos los estándares"
Emociones		👉 Preocupación 🎯 Concentración	👉 Presión 🔥 Determinación	👉 Tranquilidad 🌟 Vigilancia	👉 Ansiedad 🔴 Urgencia
Pain Points	<ul style="list-style-type: none"> <li>• Dispositivos IoT desconfigurados</li> <li>• Falta de historial de viajes similares</li> </ul>	<ul style="list-style-type: none"> <li>• Interface compleja</li> <li>• Muchos parámetros que configurar</li> </ul>	<ul style="list-style-type: none"> <li>• Información dispersa</li> <li>• Alertas poco claras</li> </ul>	<ul style="list-style-type: none"> <li>• Respuesta tardía del sistema</li> <li>• Falta de contexto en alertas</li> </ul>	<ul style="list-style-type: none"> <li>• Reportes difíciles de generar</li> <li>• Formato no compatible con cliente</li> </ul>
Oportunidades	<ul style="list-style-type: none"> <li>• Plantillas de configuración</li> <li>• Verificación automática de dispositivos</li> </ul>	<ul style="list-style-type: none"> <li>• Configuración asistida</li> <li>• Parámetros predefinidos por producto</li> </ul>	<ul style="list-style-type: none"> <li>• Dashboard unificado</li> <li>• Alertas contextualizadas</li> </ul>	<ul style="list-style-type: none"> <li>• Alertas inteligentes con recomendaciones</li> <li>• Escalamiento automático</li> </ul>	<ul style="list-style-type: none"> <li>• Reportes automáticos</li> <li>• Múltiples formatos de exportación</li> </ul>

El journey map de Carlos muestra un proceso de 5 etapas desde la planificación hasta la entrega final. Sus momentos críticos se concentran en la configuración de parámetros correctos y la gestión eficiente de alertas durante el viaje. Las oportunidades principales incluyen simplificar la configuración inicial con plantillas predefinidas, proporcionar dashboards unificados durante el monitoreo, y automatizar la generación de reportes post-viaje. Sus mayores pain points están en la complejidad de configuración y la falta de contexto en las alertas críticas.

## Journey Map: María González (Gerente de Compras de Restaurante)

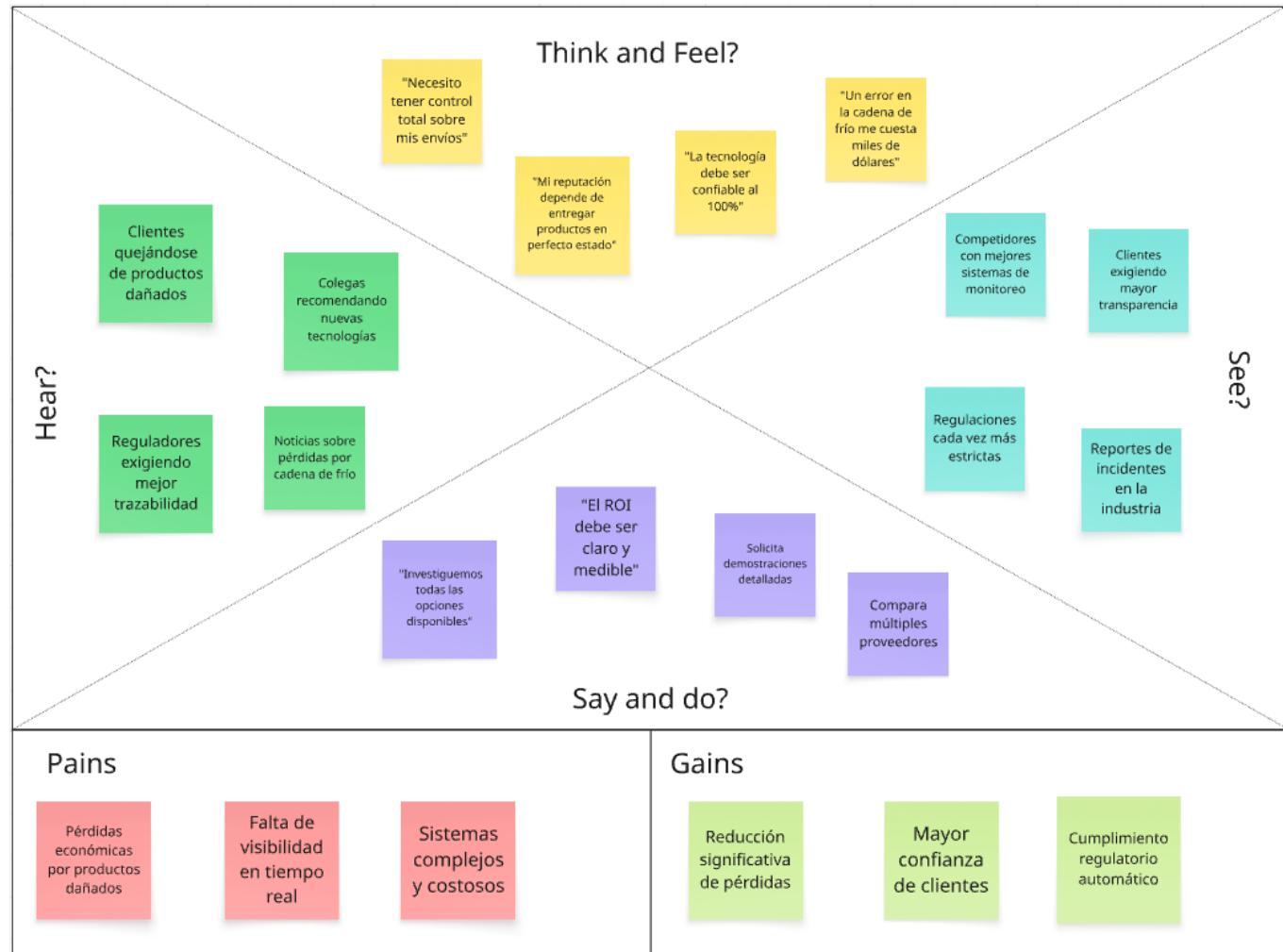
Etapa	Solicitud de Productos	Confirmación de Pedido	Durante el Transporte	Recepción y Verificación	Entrega
Acciones	<ul style="list-style-type: none"> <li>• Identifica necesidad de productos sensibles</li> <li>• Evalúa proveedores disponibles</li> <li>• Solicita cotizaciones</li> </ul>	<ul style="list-style-type: none"> <li>• Revisa términos de entrega</li> <li>• Confirma especificaciones de temperatura</li> <li>• Establece expectativas de entrega</li> </ul>	<ul style="list-style-type: none"> <li>• Espera actualizaciones del proveedor</li> <li>• Monitorea tiempos de entrega</li> <li>• Se mantiene disponible para recepción</li> </ul>	<ul style="list-style-type: none"> <li>• Inspecciona productos físicamente</li> <li>• Solicita documentación de transporte</li> <li>• Decide aceptar o rechazar</li> </ul>	<ul style="list-style-type: none"> <li>• Archiva documentación</li> <li>• Evalúa calidad del servicio</li> <li>• Planifica futuras compras</li> </ul>
Pensamientos	"Necesito asegurarme de que el proveedor garantice la calidad" "¿Cómo sé qué mantendrán la cadena de frío?"	"Espero que cumplan con los rangos de temperatura acordados" "¿Tendré visibilidad del transporte?"	"¿Estará llegando en buenas condiciones?" "No tengo forma de saber si hay problemas"	"¿Puedo confiar en que se mantuvo la temperatura?" "¿Los reportes son realmente precisos?"	"Este proveedor me da confianza" "Puedo demostrar cumplimiento a mis clientes"
Emociones	👉 Neutral 🙄 Cauteloso	👉 Ansioso 🔥 Esperanzado	👉 Preocupado 😰 Frustrado	👉 Estresado 😤 Escéptico	👉 Satisfecho 😊 Tranquilo
Pain Points	<ul style="list-style-type: none"> <li>• Falta de información sobre capacidades de cadena de frío</li> <li>• Dificultad para evaluar confiabilidad</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de transparencia en el proceso</li> <li>• No hay garantías claras de monitoreo</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de visibilidad en tiempo real</li> <li>• No recibe alertas proactivas</li> <li>• Incertidumbre sobre estado del producto</li> </ul>	<ul style="list-style-type: none"> <li>• Reportes poco detallados o poco confiables</li> <li>• Dificultad para verificar historial térmico</li> <li>• Presión de tiempo para decidir</li> </ul>	<ul style="list-style-type: none"> <li>• Documentación difícil de organizar</li> <li>• Falta de métricas de rendimiento del proveedor</li> </ul>
Oportunidades	Mostrar certificaciones y capacidades de monitoreo desde el primer contacto	Proporcionar acceso a plataforma de seguimiento desde confirmación	Enviar notificaciones automáticas y acceso a dashboard en tiempo real	Generar reportes automáticos detallados con gráficos claros y certificación digital	Proporcionar histórico de cumplimiento y métricas de desempeño del proveedor

El journey map de María ilustra un proceso enfocado en la verificación y validación de productos desde la solicitud hasta la aceptación final. Sus momentos críticos se centran en la recepción de productos y la

verificación de que cumplan con los estándares de calidad requeridos. Las oportunidades principales incluyen proporcionar acceso fácil a reportes de cumplimiento, notificaciones proactivas sobre el estado del transporte, y documentación clara que facilite la toma de decisiones de aceptación. Sus mayores pain points están en la falta de transparencia durante el transporte y la dificultad para verificar la integridad de los productos al momento de la entrega.

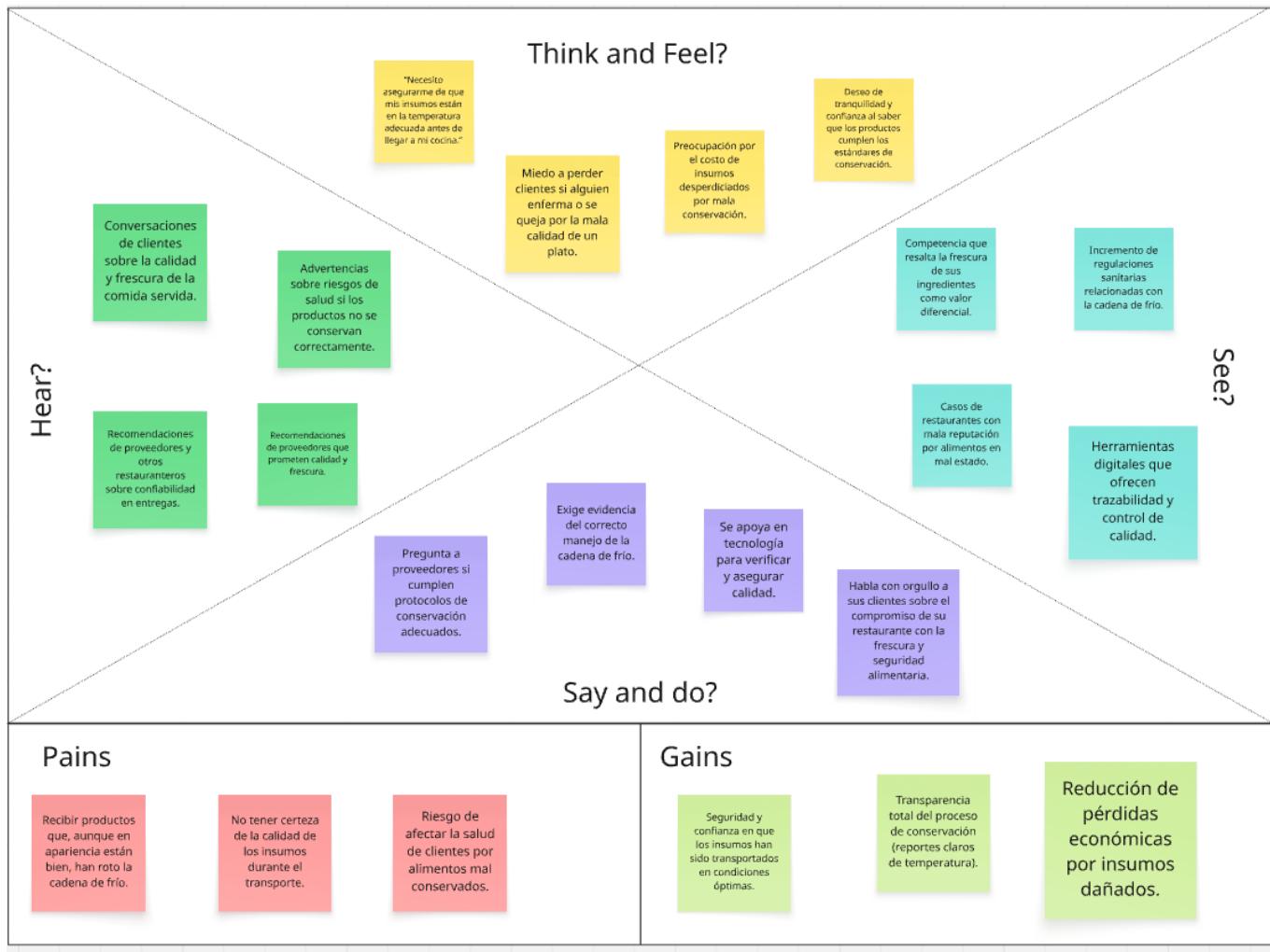
### 2.3.4. Empathy Mapping

#### Segmento: Empresa (Gestores de transporte) - Carlos Mendoza



El empathy map de Carlos revela a un profesional experimentado que busca control total y confiabilidad en los sistemas de monitoreo. Sus principales preocupaciones giran en torno a las pérdidas económicas por fallas en la cadena de frío y la necesidad de mantener la reputación empresarial. Valora la tecnología que le proporcione visibilidad en tiempo real, reportes automáticos y alertas accionables que le permitan responder rápidamente ante incidentes. Su enfoque está en el ROI medible y sistemas que cumplen con regulaciones estrictas.

#### Segmento: Clientes Finales (Consumidores finales) - María González

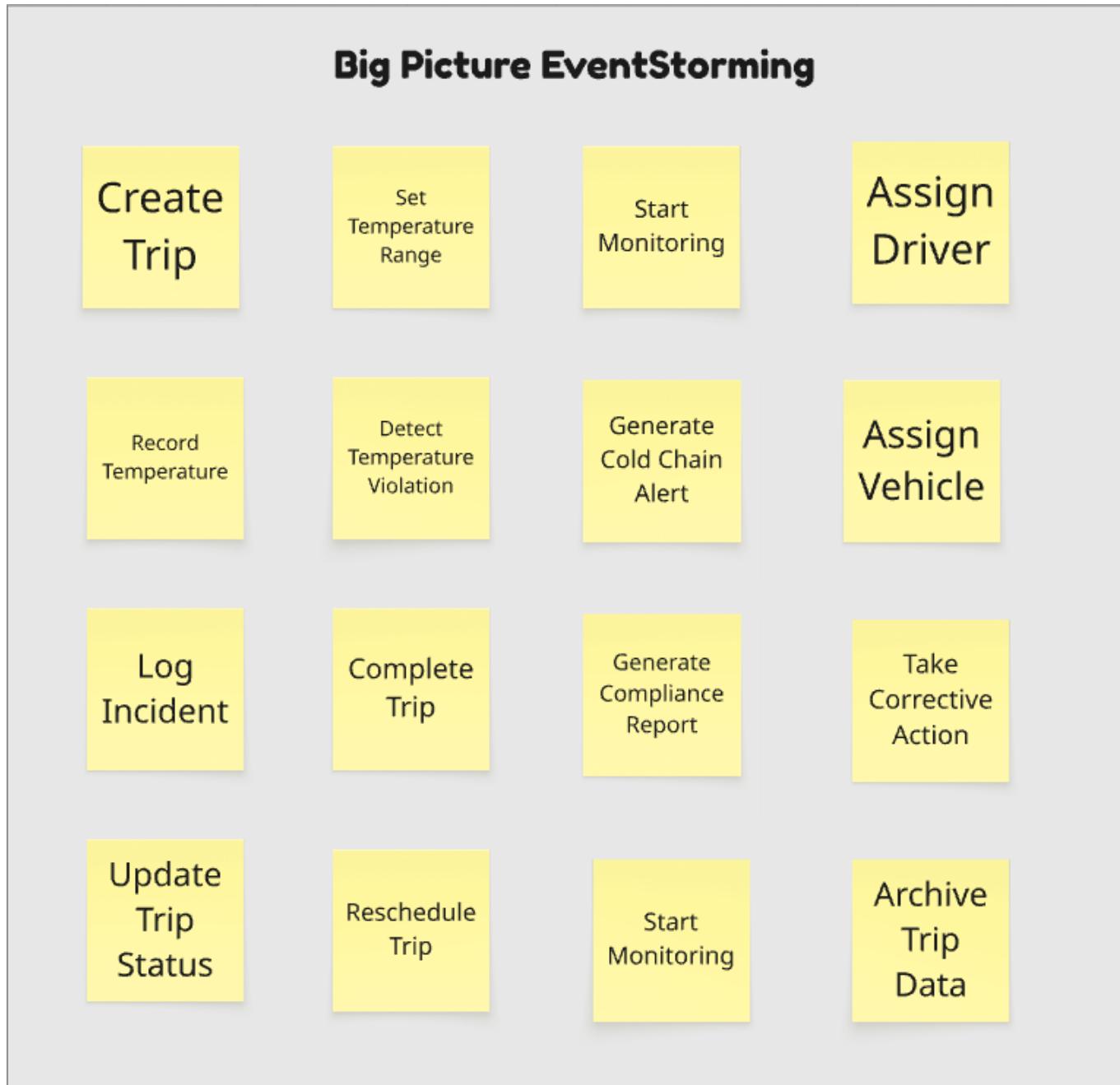


El empathy map de María revela a una profesional responsable que prioriza la calidad y la confianza en sus proveedores. Sus principales preocupaciones se centran en la reputación de su negocio y la satisfacción de sus clientes finales. Valora la transparencia en el proceso de transporte, documentación clara de cumplimiento, y la capacidad de tomar decisiones informadas sobre la aceptación de productos. Su dolor principal es la incertidumbre sobre las condiciones de transporte y la falta de información confiable que le permita verificar la calidad de los productos. Su ganancia principal es tener acceso a información transparente y reportes de cumplimiento que le den confianza para aceptar productos y mantener la calidad en su negocio.

## 2.4. Big Picture EventStorming

El Big Picture Event Storming es una técnica colaborativa de Domain-Driven Design que permitió al equipo identificar y mapear todos los eventos de dominio relevantes para el core business de CargoSafe. Durante el proceso de brainstorming, se siguió una metodología estructurada donde el equipo utilizó sticky notes naranjas para representar eventos de dominio, azules para comandos, amarillas para agregados, y rosadas para hot spots o áreas problemáticas.

El proceso comenzó con una lluvia de ideas donde todos los participantes escribieron eventos que consideraban importantes para el monitoreo de cadena de frío, siguiendo la regla de usar verbos en infinitivo en inglés. Posteriormente, se organizaron temporalmente los eventos de izquierda a derecha, formando el flujo cronológico natural del negocio. Se identificaron actores clave, sistemas externos y se marcaron los hot spots donde existían incertidumbres, riesgos o complejidades que requerían atención especial.



### **Descripción del Event Storming de CargoSafe:**

El Event Storming resultante se enfoca exclusivamente en el core business del monitoreo de cadena de frío durante el transporte. Los eventos están organizados en un flujo cronológico que abarca desde la creación del viaje hasta la generación del reporte final de cumplimiento. Los 10 eventos de dominio identificados representan las actividades críticas que agregan valor al negocio y que deben ser soportadas por la plataforma tecnológica.

## 2.5. Ubiquitous Language

El Ubiquitous Language de CargoSafe establece un vocabulario común y preciso entre el equipo de desarrollo, stakeholders del negocio y usuarios finales, eliminando ambigüedades en la comunicación sobre el dominio de monitoreo de cadena de frío.

### **Términos del Dominio Central**

### **Cold Chain (Cadena de Frío)**

Proceso logístico que mantiene productos perecederos dentro de rangos específicos de temperatura desde el origen hasta el destino final, garantizando su calidad e integridad.

### **Trip (Viaje)**

Instancia de transporte que incluye origen, destino, carga específica, parámetros térmicos requeridos y duración estimada. Representa la unidad básica de monitoreo en el sistema.

### **Temperature Range (Rango de Temperatura)**

Límites mínimos y máximos de temperatura permitidos para una carga específica durante el transporte. Se define según el tipo de producto (farmacéutico, alimentario, etc.).

### **IoT Sensor (Sensor IoT)**

Dispositivo físico que mide y transmite automáticamente lecturas de temperatura en intervalos regulares durante el viaje. Puede ser integrado al vehículo o portable.

### **Temperature Violation (Violación de Temperatura)**

Evento que ocurre cuando las lecturas del sensor salen del rango permitido por un período determinado, comprometiendo potencialmente la integridad del producto.

### **Cold Chain Alert (Alerta de Cadena de Frío)**

Notificación automática generada por el sistema cuando se detecta una violación de temperatura, enviada inmediatamente a los responsables para acción correctiva.

### **Compliance Report (Reporte de Cumplimiento)**

Documento oficial que certifica que el transporte se realizó dentro de los parámetros térmicos requeridos, utilizado para auditorías y certificaciones regulatorias.

## **Términos de Actores**

### **Fleet Manager (Gestor de Flota)**

Usuario responsable de configurar dispositivos, establecer parámetros de viaje, monitorear el estado de múltiples transportes y generar reportes para clientes.

### **Driver (Conductor)**

Usuario operativo que recibe alertas durante el transporte y ejecuta acciones correctivas cuando se detectan problemas en la cadena de frío.

### **Final Client (Cliente Final)**

Receptor de la carga que requiere visibilidad del estado del transporte y documentación de cumplimiento térmico para aceptar o rechazar productos.

## **Términos Técnicos**

### **Real-time Monitoring (Monitoreo en Tiempo Real)**

Capacidad del sistema de recibir, procesar y mostrar datos de temperatura de manera continua con latencia mínima durante el transporte.

### **Temperature Log (Registro de Temperatura)**

Secuencia cronológica de todas las lecturas de temperatura capturadas durante un viaje, almacenada para análisis posterior y generación de reportes.

### **Incident (Incidencia)**

Registro oficial de cualquier evento que comprometa la cadena de frío, incluyendo duración, causa probable y acciones tomadas para su resolución.

### **Dashboard**

Interfaz visual que presenta en tiempo real el estado de múltiples viajes, alertas activas, métricas de rendimiento y resúmenes de cumplimiento.

## **Términos de Procesos de Negocio**

### **Trip Configuration (Configuración de Viaje)**

Proceso de establecer parámetros específicos antes del inicio del transporte: rangos de temperatura, duración estimada, tipo de carga y responsables.

### **Corrective Action (Acción Correctiva)**

Medidas tomadas por el conductor u operador para resolver violaciones de temperatura: ajuste de refrigeración, cambio de ruta, o notificación a supervisores.

### **Temperature Calibration (Calibración de Temperatura)**

Proceso de verificación y ajuste de sensores para garantizar precisión en las mediciones, realizado periódicamente según estándares de calidad.

### **Cold Chain Compliance (Cumplimiento de Cadena de Frío)**

Estado que confirma que un viaje se completó dentro de todos los parámetros térmicos requeridos, cumpliendo con regulaciones y estándares de calidad.

## **Contexto de Métricas y KPIs**

### **Temperature Excursion (Excursión de Temperatura)**

Período específico durante el cual la temperatura estuvo fuera del rango permitido, medido en minutos u horas según criticidad del producto.

### **Compliance Rate (Tasa de Cumplimiento)**

Porcentaje de viajes que se completaron sin violaciones de temperatura en un período determinado, utilizado como KPI principal del servicio.

### **Alert Response Time (Tiempo de Respuesta a Alertas)**

Métrica que mide el tiempo transcurrido entre la generación de una alerta y la implementación de acciones correctivas por parte del equipo operativo.

## **Capítulo III: Requirements Specification**

### **3.1. User Stories**

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
-----------------------	--------	-------------	-------------------------	---------------------------------

<b>Epic / Story ID</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de Aceptación</b>	<b>Relacionado con (Epic ID)</b>
E1	Landing Page	Página principal con secciones informativas y de contacto para captar y orientar a los visitantes.		
E2	Autenticación	Módulo de registro e inicio de sesión seguro para usuarios.		
E3	Gestión de flota	Administración de la flota: registro, actualización y baja de vehículos.		
E4	Planificador de viajes	Creación y actualización de estados de los viajes.		
E5	Monitoreo en tiempo real	Engloba funcionalidades de monitoreo de temperatura en tiempo real y alertas.		
E6	Dashboard de viajes	Engloba pantallas, gráficos e informes relacionados a los viajes.		
E7	Módulo de suscripciones	Engloba funcionalidades de pago, manejo y control de suscripciones.		

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US001	Navegación en landing page	<p><b>Como</b> visitante  <b>quiero</b> navegar entre las secciones de la landing page <b>para</b> acceder fácilmente a la información sobre el servicio.</p>	<p>Scenario: Visualización del menú  Given un visitante ingresa a la landing page  When visualiza la parte superior de la página  Then encuentra un menú con secciones principales (Inicio, Características, Planes, Contacto)</p> <p>Scenario: Navegación entre secciones  Given un visitante hace clic en un enlace del menú  When la página carga la sección correspondiente  Then el visitante accede al contenido correcto  And el menú permanece accesible en todo momento</p>	E1
US002	Sección portada	Como visitante, quiero ver una portada con mensaje principal, para entender rápidamente el propósito de la plataforma.	<p>Scenario: Visualizar mensaje principal  Given un visitante accede a la Landing Page  When la portada se muestra  Then el visitante visualiza un mensaje claro sobre el servicio</p>	E1

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US003	Sección de funcionalidades	Como visitante, quiero visualizar una sección con las funcionalidades principales, para conocer qué ofrece la plataforma.	Scenario: Visualizar funcionalidades Given un visitante accede a la Landing Page When navega a la sección de funcionalidades Then se muestran al menos tres funcionalidades principales de la plataforma	E1
US004	Sección de beneficios	<b>Como</b> visitante, <b>quiero</b> ver una sección con beneficios, <b>para</b> entender qué valor obtengo al usar la plataforma.	Scenario: Visualizar beneficios Given un visitante accede a la Landing Page When navega a la sección de beneficios Then se muestran los beneficios de la plataforma en formato claro y resumido	E1
US005	Sección de testimonios	Como visitante, quiero ver testimonios de otros clientes, para ganar confianza en el servicio.	Scenario: Visualizar testimonios Given un visitante accede a la Landing Page When navega a la sección de testimonios Then se muestran al menos dos opiniones de clientes con sus nombres o cargos	E1

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US006	Sección de contáctanos	<p>Como visitante, quiero acceder a un formulario de contacto, para comunicarme con la empresa y solicitar más información.</p>	<p>Scenario: Enviar mensaje de contacto válido Given un visitante completa el formulario con nombre, correo y mensaje válidos When envía la información Then el sistema confirma el envío exitoso</p> <p>Scenario: Enviar mensaje con datos inválidos Given un visitante omite el correo electrónico When intenta enviar el formulario Then el sistema rechaza el envío y muestra un error de validación</p>	E1
US007	Call to Action a la aplicación web	<p><b>Como</b> visitante <b>quiero</b> encontrar un botón de acceso a la aplicación web <b>para</b> registrarme o iniciar sesión y usar el servicio desde un navegador.</p>	<p>Scenario: Visualización del call to action Given un visitante navega por la landing page When llega a la sección principal Then encuentra un botón de acceso a la aplicación web</p> <p>Scenario: Redirección a la app web Given un visitante hace clic en el botón de acceso a la aplicación web When el sistema procesa la acción Then el visitante es redirigido a la URL de la aplicación web</p>	E1

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US008	Call to Action de descarga de App Móvil	<p><b>Como</b> visitante  <b>quiero</b> encontrar botones de descarga de la aplicación móvil  <b>para</b> instalar la app en mi dispositivo iOS o Android.</p>	<p>Scenario: Visualización de botones de descarga  Given un visitante navega por la landing page  When accede a la sección de descarga  Then visualiza botones con enlaces a App Store y Google Play</p> <p>Scenario: Redirección a la tienda correspondiente  Given un visitante hace clic en un botón de descarga  When el sistema procesa la acción  Then se abre la tienda de aplicaciones correspondiente (App Store o Google Play)</p>	E1
US009	Registro de usuario	<p><b>Como</b> usuario  <b>quiero</b> registrarme en la plataforma  <b>para</b> acceder a mi cuenta y funcionalidades personalizadas.</p>	<p>Scenario: Registro de cuenta válida  Given un usuario se registra  When ingresa sus credenciales en el registro  Then el sistema autentica y permite acceso</p> <p>Scenario: Registro de cuenta inválida  Given un usuario se registra  When ingresa credenciales en el registro incorrectas  Then el sistema rechaza la autenticación y muestra error</p>	E2

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US010	Inicio de sesión	<p><b>Como</b> usuario <b>quiero</b> iniciar sesión en la plataforma <b>para</b> acceder a mi cuenta y funcionalidades personalizadas.</p>	<p>Scenario: Inicio de sesión válido Given un usuario registrado When ingresa credenciales correctas Then el sistema autentica y permite acceso</p> <p>Scenario: Inicio de sesión inválido Given un usuario registrado When ingresa credenciales incorrectas Then el sistema rechaza la autenticación y muestra error</p>	E2
US011	Cerrar sesión	<p><b>Como</b> usuario autenticado <b>quiero</b> cerrar sesión desde la aplicación <b>para</b> que mi cuenta deje de estar accesible en el dispositivo actual.</p>	<p>Scenario 1: Cierre de sesión exitoso</p> <p>Given que el usuario tiene un token válido y está autenticado When solicita cerrar sesión Then se elimina el token del lado del cliente And los siguientes intentos de acceder a recursos requieren autenticación nuevamente</p>	E2
US012	Recuperar contraseña	<p><b>Como</b> usuario <b>quiero</b> recuperar el acceso a mi cuenta mediante un proceso de restablecimiento de contraseña <b>para</b> poder ingresar nuevamente en caso de olvidar mis credenciales.</p>	<p><b>Scenario 1: Solicitud de recuperación</b></p> <p>Given un cliente no recuerda su contraseña When solicita la recuperación en la pantalla de inicio de sesión e ingresa su correo registrado Then el sistema envía un correo electrónico con un</p>	E2

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			enlace seguro para restablecer la contraseña	
			<p><b>Scenario 2: Enlace válido y único</b></p> <p>Given el cliente recibe un enlace de restablecimiento When hace clic en el enlace dentro del tiempo de validez (ej. 15 minutos) Then el sistema muestra un formulario para crear una nueva contraseña And valida que el enlace no haya sido usado previamente</p>	
			<p><b>Scenario 3: Enlace expirado</b></p> <p>Given un cliente recibe un enlace de restablecimiento When intenta usarlo después del tiempo de validez Then el sistema rechaza el intento And muestra el mensaje "El enlace ha expirado, solicita nuevamente la recuperación de contraseña"</p>	
			<p><b>Scenario 4: Contraseña actualizada correctamente</b></p> <p>Given el cliente ingresa una nueva contraseña válida When confirma el restablecimiento Then el sistema actualiza la contraseña en la base</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			de datos And notifica al cliente que el cambio se realizó con éxito	
US013	Registro de vehículos de carga	<b>Como</b> administrador logístico, <b>quiero</b> registrar vehículos en la plataforma, <b>para</b> mantener un inventario actualizado de la flota.	Scenario: Registrar un vehículo válido Given el administrador cuenta con los datos de marca, modelo, placa, capacidad y dimensiones When el administrador registra el vehículo en la plataforma Then el sistema guarda el vehículo como parte de la flota activa	E3
			Scenario: Registro sin datos obligatorios Given el administrador omite campos requeridos como marca, modelo o placa When intenta registrar el vehículo Then el sistema rechaza el registro y muestra un error de validación	
			Scenario: Validar formato de placa Given el administrador ingresa un formato de placa incorrecto When intenta registrar el vehículo Then el sistema rechaza el registro y solicita un formato válido	
			Scenario: Validar dimensiones del vehículo	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			Given el administrador ingresa dimensiones físicas del vehículo (alto, ancho, largo) When las dimensiones son numéricas y positivas Then el sistema acepta los valores y guarda la información	
			Scenario: Validar capacidad de carga Given el administrador ingresa capacidad de carga en kilogramos When la capacidad es mayor a 0 Then el sistema guarda el valor correctamente And el sistema rechaza valores negativos o nulos	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US014	Actualización de datos de vehículos de carga	<p><b>Como</b> administrador logístico, <b>quiero</b> editar la información de los vehículos, <b>para</b> mantener actualizado su estado operativo.</p>	<p>Scenario: Actualizar información básica del vehículo</p> <p>Given existe un vehículo registrado en la plataforma</p> <p>When el administrador modifica la marca, modelo o dimensiones</p> <p>Then el sistema actualiza la información y guarda los cambios en el historial</p> <p>Scenario: Validar actualización de placa duplicada</p> <p>Given un vehículo ya tiene registrada una placa única</p> <p>When el administrador intenta asignar esa misma placa a otro vehículo</p> <p>Then el sistema rechaza la actualización y muestra un error de duplicidad</p>	E3
			<p>Scenario: Mantener consistencia en datos obligatorios</p> <p>Given existe un vehículo registrado</p> <p>When el administrador elimina un dato obligatorio como marca o modelo</p> <p>Then el sistema rechaza la actualización y mantiene los datos anteriores</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			Scenario: Eliminar vehículo activo  Given un vehículo está registrado en la flota  When el administrador elimina el vehículo  Then el sistema cambia su estado a "inactivo"  And conserva su historial de viajes y mantenimientos	
US015	Eliminación de vehículos de carga	<b>Como</b> administrador logístico, <b>quiero</b> eliminar vehículos de la plataforma, <b>para</b> mantener un inventario actualizado y evitar registros obsoletos.	Scenario: Intentar eliminar vehículo con viaje en curso  Given un vehículo tiene un viaje activo  When el administrador intenta eliminarlo  Then el sistema rechaza la eliminación  And muestra un mensaje indicando que el vehículo tiene viajes pendientes	E3
			Scenario: Eliminación exitosa de vehículo sin viajes en curso  Given un vehículo no tiene viajes activos  When el administrador lo elimina  Then el sistema lo marca como eliminado  And ya no aparece en el inventario activo de la flota	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			Scenario: Registro exitoso de dispositivo nuevo Given un dispositivo IoT con un ID único When el administrador lo registra en la plataforma Then el sistema almacena el dispositivo en la base de datos And queda disponible para asignarlo a un vehículo de carga	
US016	Registro de dispositivos IoT	<p><b>Como</b> Administrador Logístico <b>quiero</b> registrar un dispositivo IoT en la plataforma <b>para</b> asociarlo a la flota y comenzar a recibir sus datos.</p>	Scenario: Intentar registrar dispositivo duplicado Given un dispositivo IoT con un ID ya registrado When el administrador intenta registrarlo nuevamente Then el sistema rechaza el registro And muestra un mensaje "El dispositivo ya está registrado"	E3
			Scenario: Validación de campos obligatorios Given un administrador intenta registrar un dispositivo sin datos requeridos como deviceld o tipo When envía la solicitud de registro Then el sistema rechaza el registro And muestra un mensaje indicando los campos faltantes	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p><b>Scenario: Eliminar dispositivo inactivo</b></p> <p><b>Given</b> un dispositivo IoT está registrado en la plataforma y no transmite datos</p> <p><b>When</b> el administrador elimina el dispositivo</p> <p><b>Then</b> el sistema cambia su estado a "<i>inactivo</i>"</p> <p><b>And</b> conserva su historial de transmisiones y alertas</p>	
US017	Eliminar dispositivo IoT	<p><b>Como</b> Administrador Logístico,</p> <p><b>quiero</b> eliminar un dispositivo IoT de la plataforma,</p> <p><b>para</b> darlo de baja en caso de falla o reemplazo.</p>	<p><b>Scenario: Intentar eliminar dispositivo activo</b></p> <p><b>Given</b> un dispositivo está enviando datos en tiempo real</p> <p><b>When</b> el administrador intenta eliminarlo</p> <p><b>Then</b> el sistema rechaza la eliminación</p> <p><b>And</b> muestra un mensaje indicando que el dispositivo sigue activo</p>	E3
			<p><b>Scenario: Eliminación exitosa de dispositivo sin actividad reciente</b></p> <p><b>Given</b> un dispositivo no transmite datos desde hace más de 24h</p> <p><b>When</b> el administrador lo elimina</p> <p><b>Then</b> el sistema lo marca como "<i>eliminado</i>"</p> <p><b>And</b> ya no aparece en la lista de dispositivos activos</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US018	Asignar dispositivo a vehículo de carga	<p><b>Como</b> Administrador Logístico <b>quiero</b> asignar un dispositivo IoT a un vehículo de carga <b>para</b> identificar a qué unidad pertenece cada transmisión de datos</p>	<p>Scenario: Asignar dispositivo a vehículo disponible  Given un dispositivo IoT registrado y sin asignar  When el administrador selecciona un vehículo de la flota  Then el sistema vincula el dispositivo a ese vehículo  And las próximas transmisiones aparecen bajo ese vehículo</p> <p>Scenario: Intentar asignar dispositivo ya vinculado  Given un dispositivo IoT ya asignado a un vehículo  When el administrador intenta asignarlo a otro sin liberar la relación  Then el sistema rechaza la acción  And muestra el mensaje "Dispositivo ya vinculado"</p>	E3

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US019	Cambiar dispositivo de vehiculo	<p><b>Como</b> Administrador Logístico <b>quiero</b> reasignar un dispositivo IoT de un vehículo a otro <b>para</b> reutilizarlo en caso de mantenimiento o rotación de la flota.</p>	<p>Scenario: Reasignar dispositivo a otro vehículo  Given un dispositivo IoT asignado a un vehículo  When el administrador lo reasigna a un vehículo diferente  Then el sistema actualiza la relación  And las transmisiones futuras aparecen bajo el nuevo vehículo</p> <p>Scenario: Confirmar cambio en inventario  Given un dispositivo IoT reasignado de un vehículo a otro  When el administrador consulta el vehículo anterior  Then el dispositivo ya no aparece en su lista de dispositivos asociados</p>	E3

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p><b>Scenario: Consulta de lista de vehículos</b></p> <p>Given existen vehículos registrados en la base de datos</p> <p>When el administrador accede a la sección de vehículos</p> <p>Then el sistema muestra la lista con placa, modelo, estado y capacidad de cada vehículo</p>	
US020	Ver vehículos de carga	<p><b>Como</b> Administrador Logístico <b>quiero</b> visualizar la lista de vehículos de carga registrados en la plataforma <b>para</b> supervisar y gestionar el inventario de transporte.</p>	<p><b>Scenario: Lista vacía de vehículos</b></p> <p>Given no existen vehículos registrados en la base de datos</p> <p>When el administrador consulta la sección de vehículos</p> <p>Then el sistema muestra un mensaje "No hay vehículos registrados"</p>	E3
			<p><b>Scenario: Consulta de detalle de vehículo</b></p> <p>Given un vehículo está registrado en la base de datos</p> <p>When el administrador selecciona un vehículo de la lista</p> <p>Then el sistema muestra su información completa</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p><b>Scenario: Consulta de lista de dispositivos</b></p> <p>Given existen dispositivos IoT registrados en la base de datos</p> <p>When el administrador accede a la sección de dispositivos</p> <p>Then el sistema muestra la lista con ID, tipo, estado y vehículo asignado de cada dispositivo</p>	
US021	Ver dispositivos IoT	<p><b>Como</b> Administrador Logístico <b>quiero</b> visualizar la lista de dispositivos IoT registrados en la plataforma <b>para</b> supervisar su estado y administración en la flota.</p>	<p>Given no existen dispositivos IoT registrados en la base de datos</p> <p>When el administrador consulta la sección de dispositivos</p> <p>Then el sistema muestra un mensaje "No hay dispositivos registrados"</p>	E3
			<p><b>Scenario: Consulta de detalle de dispositivo</b></p> <p>Given un dispositivo IoT está registrado en la base de datos</p> <p>When el administrador selecciona un dispositivo de la lista</p> <p>Then el sistema muestra su información completa</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US022	Ver estado de dispositivo por vehiculo	<p><b>Como</b> Administrador Logístico <b>quiero</b> visualizar el estado de todos los dispositivos asociados a un vehículo en el dashboard <b>para</b> monitorear su conectividad y funcionamiento.</p>	<p>Scenario: Ver dispositivos conectados  Given un vehículo con dispositivos asignados  When el administrador lo selecciona en el dashboard  Then se listan los dispositivos con estado online u offline</p>	
			<p>Scenario: Detectar dispositivo desconectado  Given un dispositivo deja de transmitir datos por más de 5 minutos  When el sistema verifica su estado  Then lo marca como "Desconectado"</p> <p>Scenario: Recuperar estado conectado  Given un dispositivo previamente desconectado  When transmite datos nuevamente  Then la plataforma actualiza su estado a "Conectado"</p>	E3

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US023	Ver estado de dispositivo en el módulo físico	<p><b>Como</b> usuario del dispositivo <b>quiero</b> que el dispositivo tenga un indicador físico de estado (ej. LED) <b>para</b> verificar rápidamente si funciona sin depender de la aplicación.</p>	<p>Scenario: Dispositivo encendido y conectado Given el dispositivo está encendido y con conexión activa When el conductor observa el módulo físico Then el LED verde permanece encendido</p> <p>Scenario: Dispositivo sin conexión Given el dispositivo pierde conectividad When el conductor observa el módulo físico Then el LED rojo parpadea cada 2 segundos</p> <p>Scenario: Batería baja Given el dispositivo tiene nivel de batería inferior al 20% When el conductor observa el módulo físico Then el LED amarillo permanece encendido</p> <p>Scenario: Dispositivo apagado Given el dispositivo ha sido apagado manualmente When el conductor observa el módulo físico Then todos los indicadores permanecen apagados</p>	E3

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US024	Creación de viajes	<p><b>Como</b> administrador logístico, <b>quiero</b> crear viajes asignando vehículo y ruta, <b>para</b> planificar el transporte de mercancías.</p>	<p>Scenario: Crear viaje válido Given existe un vehículo disponible When el administrador crea un viaje con origen y destino Then el sistema registra el viaje con estado "Pendiente"</p>	E4

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p>Scenario: Cambiar estado de viaje a "En ruta"</p> <p>Given existe un viaje con estado "Pendiente"</p> <p>When el administrador cambia su estado a "En ruta"</p> <p>Then el sistema actualiza el estado del viaje a "En ruta"</p> <p>And notifica a los clientes asociados</p>	
US025	Actualización de estados de viaje	<p><b>Como</b> administrador logístico, <b>quiero</b> actualizar el estado de un viaje, <b>para</b> mantener informados a gerentes y clientes sobre el progreso de la entrega.</p>	<p>Scenario: Cambiar estado de viaje a "Finalizado"</p> <p>Given existe un viaje en estado "En ruta"</p> <p>When el administrador cambia el estado a "Finalizado"</p> <p>Then el sistema marca el viaje como finalizado</p> <p>And registra la fecha y hora de finalización</p>	E4

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US026	Reprogramación de viajes	<p><b>Como</b> administrador logístico,  <b>quiero</b> reprogramar un viaje ya creado,  <b>para</b> ajustar fechas y horarios en caso de cambios operativos.</p>	<p>Scenario: Reprogramar un viaje válido  Given existe un viaje programado con estado "Pendiente"  When el administrador modifica la fecha y hora de inicio  Then el sistema actualiza la información del viaje y guarda el cambio en el historial</p> <p>Scenario: Intentar reprogramar un viaje en curso  Given existe un viaje con estado "En ruta"  When el administrador intenta cambiar la fecha u hora de inicio  Then el sistema rechaza la reprogramación y muestra un mensaje de restricción</p>	E4

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US027	Código de viaje para cliente	<p><b>Como</b> cliente final,  <b>quiero</b> recibir un código único de viaje,  <b>para</b> poder consultar fácilmente el estado de mi pedido.</p>	<p>Scenario: Generar código de viaje al crear pedido  Given el administrador crea un viaje nuevo  When el sistema registra el viaje  Then se genera un código único y se envía al cliente por correo o notificación en la aplicación</p> <p>Scenario: Consultar estado con código de viaje  Given un cliente cuenta con el código de su viaje  When ingresa el código en la aplicación  Then el sistema muestra el estado actual del viaje en tiempo real</p>	E4

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US028	Reprogramación de viajes	<p><b>Como</b> administrador logístico, <b>quiero</b> reprogramar un viaje ya creado, <b>para</b> ajustar fechas y horarios en caso de cambios operativos.</p>	<p>Scenario: Reprogramar un viaje válido Given existe un viaje programado con estado "Pendiente" When el administrador modifica la fecha y hora de inicio Then el sistema actualiza la información del viaje y guarda el cambio en el historial</p> <p>Scenario: Intentar reprogramar un viaje en curso Given existe un viaje con estado "En ruta" When el administrador intenta cambiar la fecha u hora de inicio Then el sistema rechaza la reprogramación y muestra un mensaje de restricción</p>	E4
US029	Actualización de temperatura en tiempo real	<p>Como empresa, quiero recibir la temperatura en tiempo real de mis dispositivos IoT, para supervisar la cadena de frío de los viajes.</p>	<p>Given un sensor IoT transmite datos de temperatura When el sistema recibe los datos Then la temperatura se actualiza en la plataforma sin retrasos.</p> <p>Given un dispositivo pierde conexión When intenta enviar datos Then el sistema muestra la última lectura disponible y marca estado de conexión.</p>	E4

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US030	Alertas por incumplimiento de temperatura	Como cliente final, quiero recibir alertas cuando la temperatura sobrepasa los límites definidos, para tomar acciones correctivas.	Given la temperatura excede el rango permitido When el sistema procesa la lectura Then se genera una alerta y se notifica al usuario.	E5
US031	Alertas de conexión IoT	Como empresa, quiero recibir alertas cuando un dispositivo IoT deja de enviar datos, para actuar de inmediato.	Given un usuario tiene notificaciones activadas When recibe una alerta Then visualiza el mensaje en el sistema o correo/SMS.	E5
US032	Roles y permisos de acceso	Como empresa, quiero que el sistema gestione roles y permisos de usuarios (admin, cliente, operador), para controlar accesos.	Given un dispositivo deja de transmitir por más de X minutos When el sistema detecta la ausencia Then genera una alerta para la empresa.	E5
US033	Lista de viajes registrados	Como empresa, quiero ver una lista de todos los viajes registrados para gestionarlos de forma rápida.	Given un usuario con rol administrador When accede al panel de gestión Then puede realizar todas las operaciones.  Given un usuario con rol cliente When accede al sistema Then solo puede ver información de sus viajes y suscripciones.	E5
			Given un usuario autenticado accede al dashboard When solicita la lista de viajes Then el sistema muestra todos los viajes con sus datos principales.	E6

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US034	Detalle de viaje	Como cliente final, quiero consultar el detalle de un viaje, para verificar información específica como ruta, estado y temperatura.	Given un viaje está registrado When el usuario selecciona un viaje de la lista Then se muestran todos sus detalles asociados.	E6
US035	Gráficos de tiempo y temperatura	Como cliente final, quiero ver gráficos de evolución de la temperatura durante el viaje, para verificar el cumplimiento de parámetros.	Given un viaje cuenta con datos de temperatura When se consulta el detalle Then el sistema despliega un gráfico de línea con los valores.	E6
US036	Gráficos de incidencias por mes	Como empresa, quiero ver un gráfico mensual de incidencias para identificar patrones de fallos.	Given existen incidencias en el histórico When el usuario consulta el dashboard Then se muestra un gráfico con número de incidencias por mes.	E6
US037	Filtrado de viajes por fecha	Como empresa, quiero filtrar la lista de viajes por rango de fechas, para analizar un periodo específico.	Given un usuario aplica un filtro de fechas When el sistema procesa la consulta Then se muestran solo los viajes que cumplen el criterio.	E6
US038	Descarga de reporte de viajes	Como cliente final, quiero descargar un reporte en PDF de un viaje con su información y gráficos, para archivarlo o compartirlo.	Given un usuario selecciona un viaje When solicita la exportación Then el sistema genera y entrega un archivo PDF con la información del viaje.	E6

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US039	Cancelar suscripción	Como cliente final, quiero cancelar mi suscripción, para detener los cobros futuros.	Given un cliente con suscripción activa When solicita la cancelación Then el sistema marca la suscripción como cancelada y detiene próximos pagos.	E7
US040	Visualizar información de suscripción	Como cliente final, quiero ver mi estado de suscripción y fecha de expiración, para gestionar mi acceso al servicio.	Given un cliente autenticado accede a la sección de suscripción When consulta su información Then el sistema muestra el plan, estado y fecha de expiración.	E7
US041	Historial de pagos	Como empresa, quiero que los clientes consulten su historial de pagos, para brindar transparencia.	Given un cliente autenticado When accede al historial de pagos Then el sistema muestra todas las transacciones registradas con fecha y monto.	E7
US042	Notificación de renovación próxima	Como cliente final, quiero recibir una notificación antes de que mi suscripción se renueve, para decidir si continúo o cancelo.	Given una suscripción está próxima a renovarse (ej. 3 días antes) When el sistema procesa la fecha Then envía una notificación al cliente.	E7
TS001	API de autenticación	<b>Como developer</b> <b>quiero</b> implementar múltiples endpoints de autenticación (login, logout, refresh y validación de sesión) <b>para</b> que los usuarios puedan gestionar de forma segura su acceso a la plataforma.	<b>Scenario: Login exitoso</b> Given un usuario envía credenciales válidas al endpoint de login When el backend valida usuario y contraseña Then el sistema responde con código <b>200 OK</b> And devuelve un token de acceso y un refresh token	E2

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p><b>Scenario: Login fallido</b></p> <p>Given un usuario envía credenciales inválidas al endpoint de login</p> <p>When el backend intenta validarlas</p> <p>Then el sistema responde con código <b>401 Unauthorized</b></p>	
			<p>And devuelve un mensaje de error de autenticación</p>	
			<p><b>Scenario: Logout exitoso</b></p> <p>Given un usuario tiene sesión activa con un refresh token válido</p> <p>When envía solicitud al endpoint de logout</p> <p>Then el backend responde con código <b>200 OK</b></p> <p>And el refresh token queda invalidado</p>	
			<p><b>Scenario: Registro exitoso de usuario</b></p> <p>Given un nuevo usuario envía datos válidos (correo, contraseña y nombre) al endpoint de registro</p> <p>When el backend valida la información y la almacena en la base de datos</p> <p>Then el sistema responde con código <b>201 Created</b></p> <p>And devuelve un mensaje confirmando la creación del usuario</p>	
			<p><b>Scenario: Registro con correo duplicado</b></p> <p>Given un correo ya existe</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p>en la base de datos</p> <p>When un usuario intenta registrarse con ese mismo correo</p> <p>Then el sistema responde con código <b>409 Conflict</b></p> <p>And devuelve un mensaje "El correo ya está registrado"</p>	<b>Scenario: Refresh token válido</b>  Given un usuario tiene un refresh token válido When envía solicitud al endpoint de refresh Then el backend responde con código <b>200 OK</b> And devuelve un nuevo par de tokens de acceso y refresh
				<b>Scenario: Token inválido o expirado en validación</b>  Given un usuario envía un token inválido o expirado al endpoint de validación When el backend procesa el token Then el sistema responde con código <b>401 Unauthorized</b> And devuelve un mensaje "Token inválido o expirado"
TS002	Servicio de autenticación con JWT	<b>Como</b> developer, <b>quiero</b> implementar autenticación basada en JWT, <b>para</b> asegurar la comunicación entre cliente y servidor.	<b>Scenario: Generación de token JWT</b>  Given un usuario inicia sesión exitosamente When el backend autentica sus credenciales Then genera un token JWT	E2

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			con payload que incluye <code>userId</code> , <code>rol</code> y fecha de expiración	
			<p><b>Scenario: Verificación de token JWT válido</b></p> <p>Given un cliente envía un token JWT válido en la cabecera Authorization</p> <p>When el backend lo verifica con la clave secreta</p> <p>Then acepta la solicitud</p> <p>And permite acceso al recurso protegido</p>	
			<p><b>Scenario: Verificación de token JWT inválido</b></p> <p>Given un cliente envía un token JWT alterado o mal formado</p> <p>When el backend lo valida</p> <p>Then rechaza la solicitud</p> <p>And devuelve un error 401 Unauthorized</p>	
			<p><b>Scenario: Token expirado</b></p> <p>Given un cliente envía un token JWT expirado</p> <p>When el backend intenta validararlo</p> <p>Then rechaza la solicitud</p> <p>And devuelve un error 401 Unauthorized con mensaje "Token expirado"</p>	
			<p><b>Scenario: Renovación de sesión con refresh token</b></p> <p>Given un access token está expirado</p> <p>When el cliente envía un refresh token válido</p> <p>Then el backend genera</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			un nuevo JWT de acceso And responde con los nuevos tokens	
TS003	API de vehículos de carga	<p><b>Como</b> developer, <b>quiero</b> exponer un endpoint para registrar, modificar y consultar vehículos de carga, <b>para</b> que el backend gestione el inventario de la flota.</p>	<p>Scenario: Registro exitoso de vehículo Given un vehículo con datos válidos (placa, modelo, año, capacidad) When se envía una solicitud POST al endpoint de vehículos Then el sistema almacena el vehículo en la base de datos And devuelve un código 201 con el detalle del vehículo registrado</p> <p>Scenario: Intentar registrar vehículo duplicado Given un vehículo con una placa ya existente en el sistema When el developer intenta registrarlo nuevamente Then el sistema rechaza la solicitud And devuelve un código 409 con el mensaje "El vehículo ya está registrado"</p> <p>Scenario: Modificar datos de un vehículo existente Given un vehículo registrado en la base de datos When se envía una solicitud PUT con los campos actualizados Then el sistema actualiza los datos del vehículo</p>	E3

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p>And devuelve un código 200 con el detalle actualizado</p> <p>Scenario: Consultar vehículo por ID Given un vehículo registrado en la base de datos When se envía una solicitud GET al endpoint con su identificador Then el sistema devuelve un código 200 And muestra los datos completos del vehículo</p> <p>Scenario: Consultar lista de vehículos Given existen varios vehículos registrados en la base de datos When se envía una solicitud GET al endpoint de vehículos Then el sistema devuelve un código 200 And muestra la lista completa de vehículos disponibles en el inventario</p>	
TS004	API de dispositivos IoT	Como developer quiero exponer un endpoint de dispositivos para que la plataforma pueda registrar y administrar los dispositivos IoT.	<p>Scenario: Registro exitoso de dispositivo Given un dispositivo IoT con datos válidos (deviceld, tipo, estado inicial) When se envía una solicitud POST al endpoint de dispositivos Then el sistema almacena el dispositivo en la base</p>	E3

<b>Epic / Story ID</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de Aceptación</b>	<b>Relacionado con (Epic ID)</b>
			de datos And devuelve un código 201 con el detalle del dispositivo registrado	
			Scenario: Intentar registrar dispositivo duplicado Given un dispositivo IoT con un deviceld ya existente en la base de datos When se intenta Registrarlo nuevamente Then el sistema rechaza la solicitud And devuelve un código 409 con el mensaje "El dispositivo ya está Registrado"	
			Scenario: Consultar dispositivo por ID Given un dispositivo IoT Registrado en la base de datos When se envía una solicitud GET al endpoint con su identificador Then el sistema devuelve un código 200 And muestra los datos completos del dispositivo	
			Scenario: Consultar lista de dispositivos Given existen varios dispositivos IoT Registrados en la base de datos When se envía una solicitud GET al endpoint de dispositivos Then el sistema devuelve	

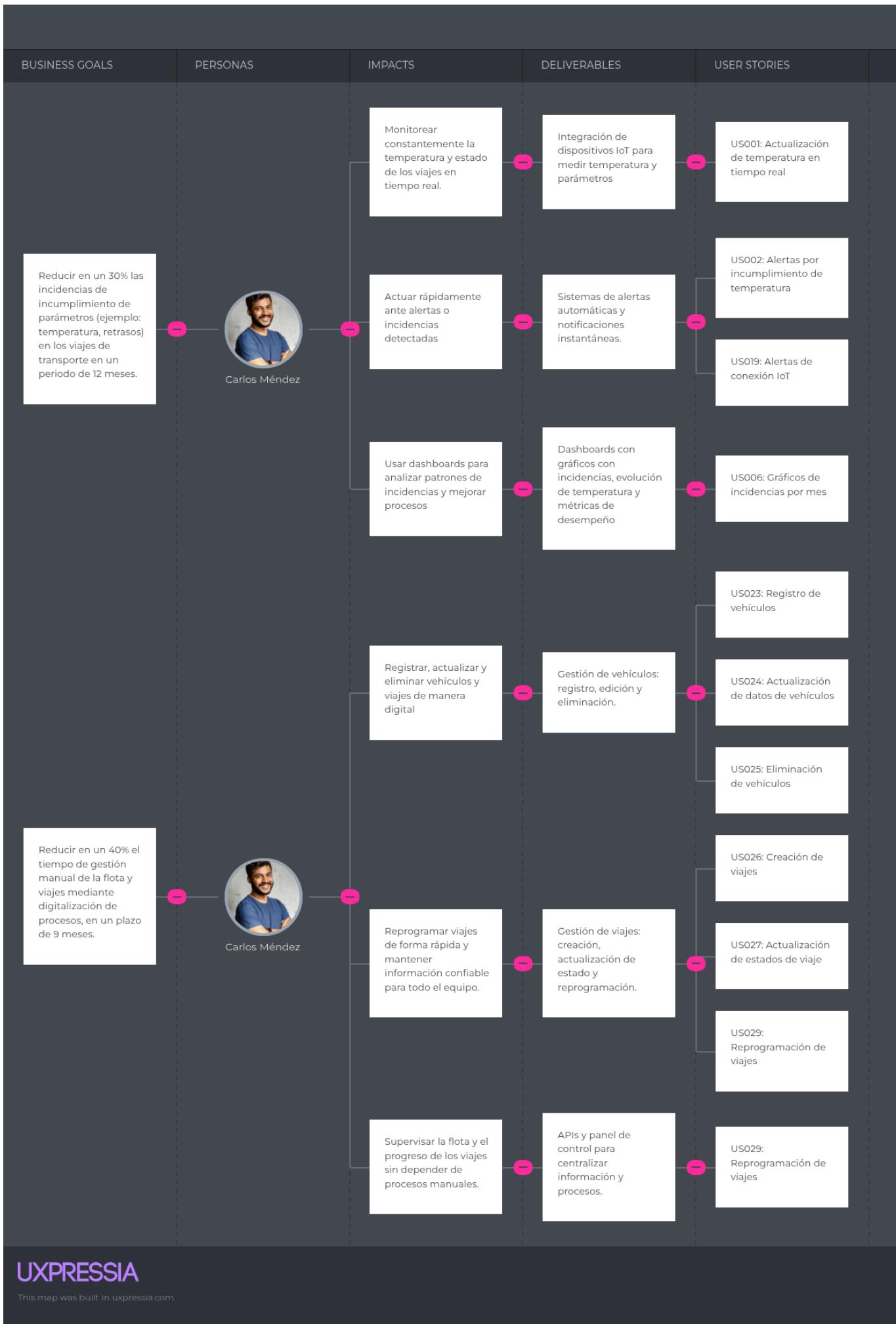
Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p>un código 200        And muestra la lista completa de dispositivos registrados</p> <p>Scenario: Modificar datos de un dispositivo        Given un dispositivo IoT registrado en la base de datos        When se envía una solicitud PUT con campos actualizados (ej. nombre, estado, vehículo asignado)        Then el sistema actualiza los datos del dispositivo        And devuelve un código 200 con el detalle actualizado</p> <p>Scenario: Eliminar dispositivo registrado        Given un dispositivo IoT existe en la base de datos y no tiene transmisiones activas        When se envía una solicitud DELETE al endpoint de dispositivos        Then el sistema cambia su estado a "inactivo"        And conserva su historial de transmisiones y eventos</p>	
TS005	API de viajes	<p><b>Como</b> developer,  <b>quiero</b> exponer un endpoint RESTful para registrar viajes,  <b>para</b> que la aplicación guarde y gestione la información.</p>	<p><b>Scenario: Registro exitoso de viaje</b>        Given un cliente envía los datos válidos de un viaje (vehículo, dispositivo IoT, fecha de inicio, destino) al endpoint de viajes        When el backend valida la</p>	E4

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p>información</p> <p>Then el sistema responde con código <b>201 Created</b></p> <p>And almacena el viaje en la base de datos</p>	
			<p><b>Scenario: Registro con datos incompletos</b></p> <p>Given un cliente envía una solicitud de registro sin todos los campos obligatorios</p> <p>When el backend procesa la solicitud</p> <p>Then el sistema responde con código <b>400 Bad Request</b></p> <p>And devuelve un mensaje indicando los campos faltantes</p>	
			<p><b>Scenario: Registro con vehículo no existente</b></p> <p>Given el identificador de vehículo no existe en la base de datos</p> <p>When el cliente intenta registrar un viaje con ese vehículo</p> <p>Then el sistema responde con código <b>404 Not Found</b></p> <p>And devuelve un mensaje "Vehículo no encontrado"</p>	
			<p><b>Scenario: Registro con dispositivo IoT no existente</b></p> <p>Given el identificador de dispositivo IoT no existe en la base de datos</p> <p>When el cliente intenta registrar un viaje con ese</p>	

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
			<p>dispositivo</p> <p>Then el sistema responde con código <b>404 Not Found</b></p> <p>And devuelve un mensaje "Dispositivo IoT no encontrado"</p> <p><b>Scenario: Validación de solapamiento de viajes</b></p> <p>Given un vehículo ya tiene un viaje activo en la base de datos</p> <p>When se intenta registrar un nuevo viaje con el mismo vehículo en paralelo</p> <p>Then el sistema responde con código <b>409 Conflict</b></p> <p>And devuelve un mensaje "El vehículo ya tiene un viaje activo"</p>	

### 3.2. Impact Mapping

Empresa (Gestores de transporte)



## Clientes Finales (Consumidores finales)



**UXPRESSIA**

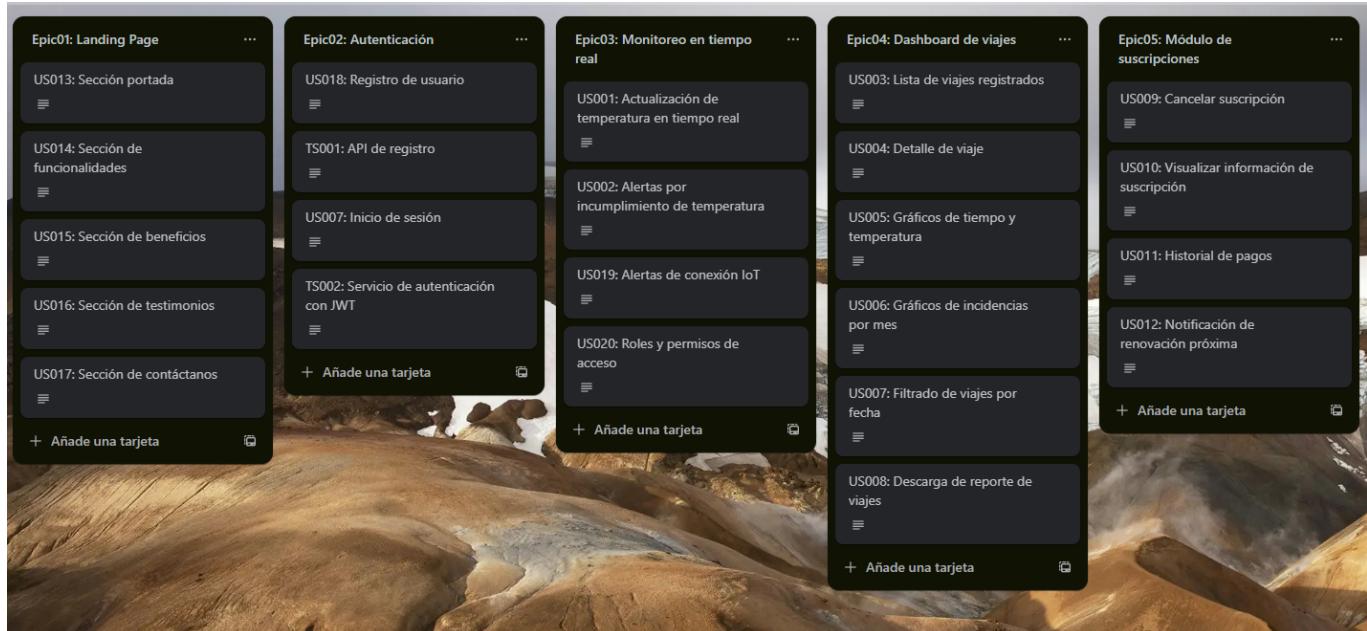
This map was built in [uxpressia.com](https://uxpressia.com)

### 3.3. Product Backlog

# Orden	User Story ID	Título	Descripción	Story Points
1	US013	Sección portada	Como visitante, quiero ver una portada con mensaje principal, para entender rápidamente el propósito de la plataforma.	3
2	US014	Sección de funcionalidades	Como visitante, quiero visualizar una sección con las funcionalidades principales, para conocer qué ofrece la plataforma.	3
3	US015	Sección de beneficios	Como visitante, quiero ver una sección con beneficios, para entender qué valor obtengo al usar la plataforma.	2
4	US016	Sección de testimonios	Como visitante, quiero ver testimonios de otros clientes, para ganar confianza en el servicio.	2
5	US017	Sección de contáctanos	Como visitante, quiero acceder a un formulario de contacto, para comunicarme con la empresa y solicitar más información.	3
6	US018	Registro de usuario	Como usuario, quiero registrarme en la plataforma, para acceder a mi cuenta y funcionalidades personalizadas.	3
7	TS001	API de registro	Como developer, quiero implementar autenticación basada en JWT, para asegurar la comunicación entre cliente y servidor.	8
8	US007	Inicio de sesión	Como usuario registrado, quiero iniciar sesión en la plataforma, para acceder a mi cuenta y funcionalidades personalizadas.	3
9	TS002	Servicio de autenticación con JWT	Como developer, quiero implementar autenticación basada en JWT, para asegurar la comunicación entre cliente y servidor.	8
10	US001	Actualización de temperatura en tiempo real	Como empresa, quiero recibir la temperatura en tiempo real de mis dispositivos IoT, para supervisar la cadena de frío de los viajes.	8
11	US002	Alertas por incumplimiento de temperatura	Como cliente final, quiero recibir alertas cuando la temperatura sobrepasa los límites definidos, para tomar acciones correctivas.	5
12	US019	Alertas de conexión IoT	Como empresa, quiero recibir alertas cuando un dispositivo IoT deja de enviar datos, para actuar de inmediato.	5
13	US020	Roles y permisos de acceso	Como empresa, quiero que el sistema gestione roles y permisos de usuarios (admin, cliente, operador), para controlar accesos.	8

# Orden	User Story ID	Título	Descripción	Story Points
14	US003	Lista de viajes registrados	Como empresa, quiero ver una lista de todos los viajes registrados para gestionarlos de forma rápida.	3
15	US004	Detalle de viaje	Como cliente final, quiero consultar el detalle de un viaje, para verificar información específica como ruta, estado y temperatura.	3
16	US005	Gráficos de tiempo y temperatura	Como cliente final, quiero ver gráficos de evolución de la temperatura durante el viaje, para verificar el cumplimiento de parámetros.	5
17	US006	Gráficos de incidencias por mes	Como empresa, quiero ver un gráfico mensual de incidencias para identificar patrones de fallos.	5
18	US007	Filtrado de viajes por fecha	Como empresa, quiero filtrar la lista de viajes por rango de fechas, para analizar un periodo específico.	3
19	US008	Descarga de reporte de viajes	Como cliente final, quiero descargar un reporte en PDF de un viaje con su información y gráficos, para archivarlo o compartirlo.	5
20	US009	Cancelar suscripción	Como cliente final, quiero cancelar mi suscripción, para detener los cobros futuros.	2
21	US010	Visualizar información de suscripción	Como cliente final, quiero ver mi estado de suscripción y fecha de expiración, para gestionar mi acceso al servicio.	2
22	US011	Historial de pagos	Como empresa, quiero que los clientes consulten su historial de pagos, para brindar transparencia.	3
23	US012	Notificación de renovación próxima	Como cliente final, quiero recibir una notificación antes de que mi suscripción se renueve, para decidir si continúo o cancelo.	3
24	US023	Registro de vehículos	Como administrador logístico, quiero registrar vehículos en la plataforma, para mantener un inventario actualizado de la flota.	5
25	US024	Actualización de datos de vehículos	Como administrador logístico, quiero editar la información de los vehículos, para mantener actualizado su estado operativo.	3
26	US025	Eliminación de vehículos	Como administrador logístico, quiero eliminar vehículos de la plataforma, para mantener un inventario actualizado y evitar registros obsoletos.	3

# Orden	User Story ID	Título	Descripción	Story Points
27	TS003	API de vehículos	Como developer, quiero exponer un endpoint para registrar, modificar y consultar vehículos, para que el backend gestione el inventario de la flota.	8
28	US026	Creación de viajes	Como administrador logístico, quiero crear viajes asignando vehículo y ruta, para planificar el transporte de mercancías.	5
29	US027	Actualización de estados de viaje	Como administrador logístico, quiero actualizar el estado de un viaje, para mantener informados a gerentes y clientes sobre el progreso de la entrega.	5
30	TS004	API de viajes	Como developer, quiero exponer un endpoint RESTful para registrar viajes, para que la aplicación guarde y gestione la información.	8
31	US028	Visualización de viajes	Como cliente final, quiero visualizar el estado de mis pedidos en tiempo real, para tener transparencia en la entrega.	5
32	TS005	API de tracking de viajes	Como developer, quiero exponer un endpoint que devuelva el estado actual de un viaje, para que los clientes puedan consultar el seguimiento en tiempo real.	8
33	US029	Reprogramación de viajes	Como administrador logístico, quiero reprogramar un viaje ya creado, para ajustar fechas y horarios en caso de cambios operativos.	5
34	US030	Código de viaje para cliente	Como cliente final, quiero recibir un código único de viaje, para poder consultar fácilmente el estado de mi pedido.	5
35	US031	Formulario de soporte y reclamos	Como cliente final, quiero acceder a un formulario de soporte y reclamos, para comunicar incidencias o problemas con mis viajes.	3
36	US032	Recepción de reclamos por viajes	Como administrador logístico, quiero recibir los reclamos enviados por los clientes, para analizarlos y dar respuesta oportuna.	3



### Link del Trello:

<https://trello.com/invite/b/68c988fa65c37d7ded09248f/ATTId7fe8515afa46ad0ba2a3eb460ca5c06BCFF0CFB/1osparkersiot-productbacklog>

## Capítulo IV: Solution Software Design

---

### 4.1. Strategic-Level Domain-Driven Design

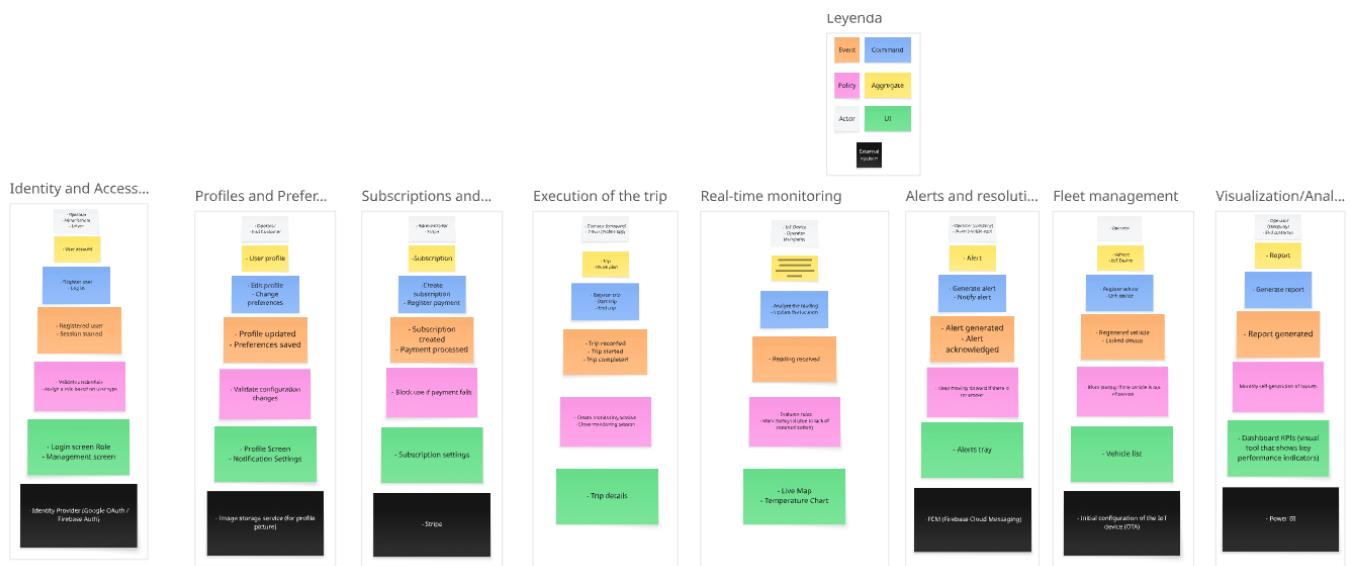
#### 4.1.1. Design-Level EventStorming

##### 4.1.1.1 Candidate Context Discovery

Para esta etapa se llevó a cabo una sesión, la sesión tuvo una duración aproximada de 90 minutos y permitió identificar los bounded contexts del sistema CargaSafe. Durante el proceso se aplicaron las técnicas start-with-value, start-with-simple y look-for-pivotal-events, que facilitaron la agrupación de eventos y entidades según su afinidad y valor para el negocio.

Como resultado, se identificaron ocho bounded contexts:

- **Identity and Access Management:** administración de usuarios, autenticación y control de accesos.
- **Profiles and Preferences Management:** gestión de perfiles de usuario y configuración de preferencias.
- **Fleet management:** gestión de vehículos y dispositivos IoT.
- **Trip management:** creación y ejecución de viajes.
- **Real-time monitoring:** monitoreo de condiciones en tiempo real.
- **Alerts and resolution:** generación de alertas.
- **Visualization/Analytics:** visualización de métricas y reportes.
- **Subscriptions and payments:** gestión de suscripciones y pagos con Stripe.



## Leyenda utilizada en el EventStorming

- **Event**: describe algo que ocurrió en el dominio (Viaje iniciado, Alerta generada).
- **Command**: una instrucción o acción que dispara un evento (Registrar viaje).
- **Policy**: regla de negocio que determina qué ocurre ante ciertas condiciones (Si falta dispositivo → bloquear inicio del viaje).
- **Aggregate**: entidad principal que concentra datos y operaciones (Viaje, Suscripción).
- **UI**: vistas o pantallas del sistema que muestran información al usuario (Dashboard de KPIs).
- **Actor**: roles que interactúan con el sistema (Operador, Conductor).
- **Sistema externo**: integraciones con servicios de terceros (Google Maps, Stripe).

Con esta estructura, el EventStorming permitió organizar y simplificar el dominio de CargaSafe, evidenciando de forma clara los contextos candidatos y la interacción entre actores, procesos y sistemas externos.

[Ver gráfico en Miro](#)

### 4.1.1.2. Domain Message Flows Modeling

En esta etapa se desarrolló el **modelado de flujos de mensajes de dominio (Domain Message Flows)** con el objetivo de visualizar cómo colaboran los bounded contexts identificados en el Candidate Context Discovery para resolver los principales casos de negocio del sistema CargaSafe.

Para la construcción de estos flujos se aplicó la técnica de **Domain Storytelling**, la cual permite describir las interacciones en un lenguaje natural, mostrando cómo un evento generado en un bounded context desencadena comandos o nuevos eventos en otros contextos. De este modo se logra una visión clara de la cooperación entre módulos y del ciclo de vida de la información dentro de la plataforma.

## Historias de dominio (Domain Stories)

### 1. Gestión de identidad y perfiles

- Cuando un *usuario se registra* en **Identity and Access Management**, se genera un evento que es consumido por **Profiles and Preferences**, el cual crea automáticamente el perfil asociado.
- Si un *usuario edita sus preferencias*, se guarda la configuración en **Profiles**, y en caso de referirse a notificaciones, estas se utilizan en **Alerts** para personalizar los canales de envío.

## 2. Control de acceso y suscripciones

- Cuando un *pago es procesado exitosamente* en **Subscriptions & Billing**, se envía un evento a **Identity and Access Management**, que habilita el acceso al sistema.
- Si un *pago falla*, el mismo flujo comunica a IAM que debe restringir o bloquear el acceso del usuario hasta regularizar su situación.

## 3. Gestión de flota y ejecución de viajes

- Al *registrarse un vehículo o dispositivo IoT* en **Fleet Management**, este queda disponible para **Trip Management**, que puede asignarlo a un viaje planificado.
- Cuando un *operador crea e inicia un viaje* en **Trip Management**, se emite un evento que da origen a una sesión de monitoreo en **Monitoring**.

## 4. Monitoreo en tiempo real y alertas

- **Monitoring** recibe continuamente *lecturas de sensores* (temperatura, ubicación, señal). Si se detecta una condición fuera de rango, se genera un evento que es consumido por **Alerts**.
- **Alerts** crea la alerta correspondiente y la notifica a los usuarios, aplicando las preferencias definidas en **Profiles** (por ejemplo, envío por SMS, correo o notificación push).

## 5. Analítica y reportes

- Cada *alerta generada o reconocida* en **Alerts** actualiza los indicadores en **Dashboard & Analytics**, alimentando las métricas de cumplimiento y los reportes de incidentes.
- Cuando **Dashboard & Analytics** genera un *reporte final*, este puede personalizarse de acuerdo con las preferencias almacenadas en **Profiles**, permitiendo al usuario recibir información ajustada a su rol o necesidades.



## Resultados

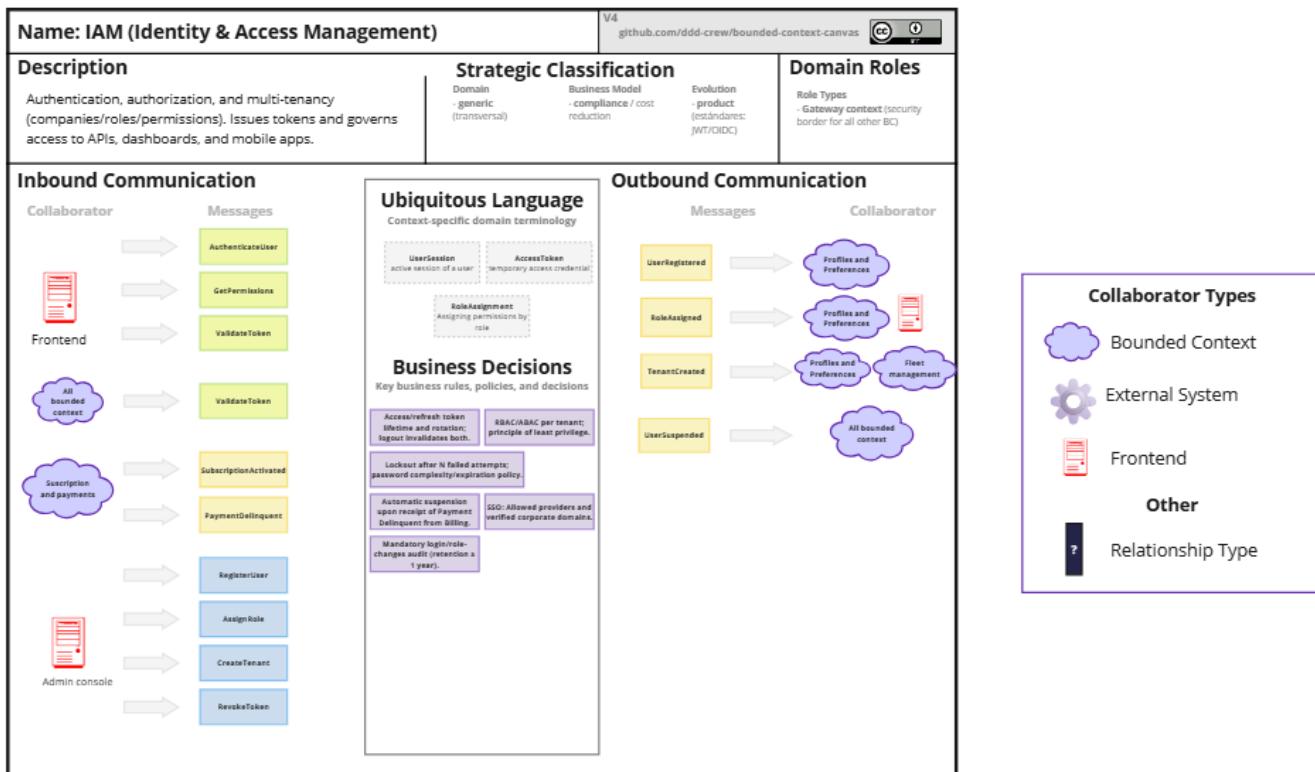
Los flujos de mensajes de dominio evidencian la cooperación entre los ocho bounded contexts de CargaSafe:

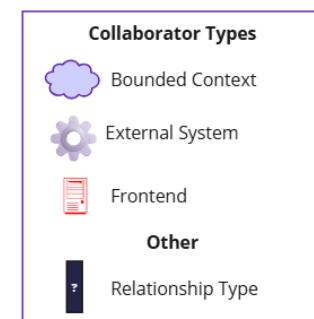
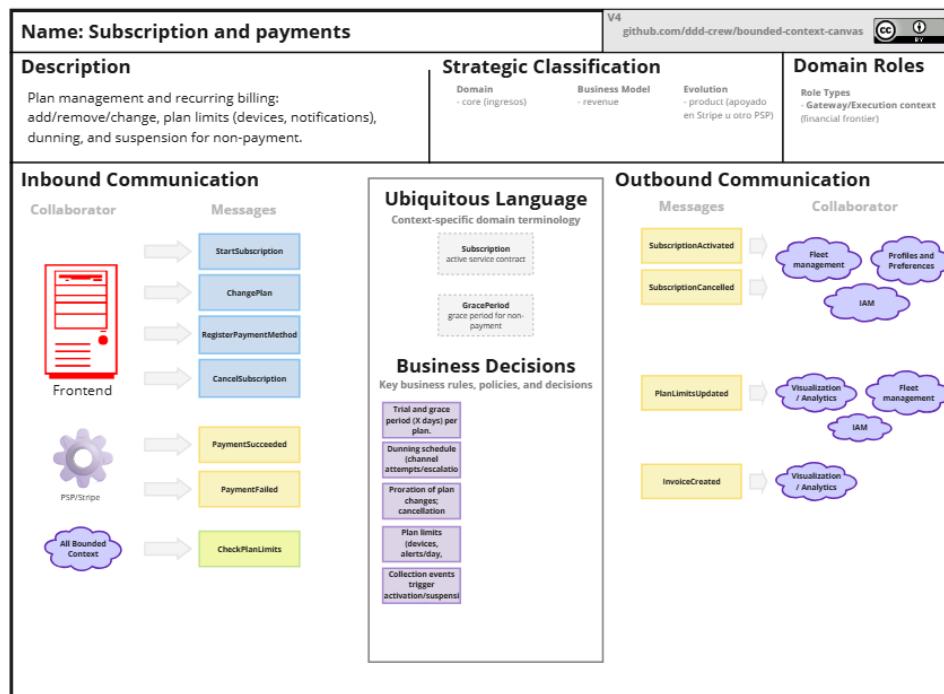
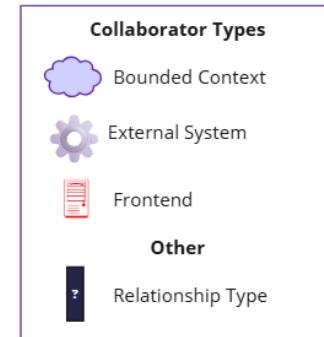
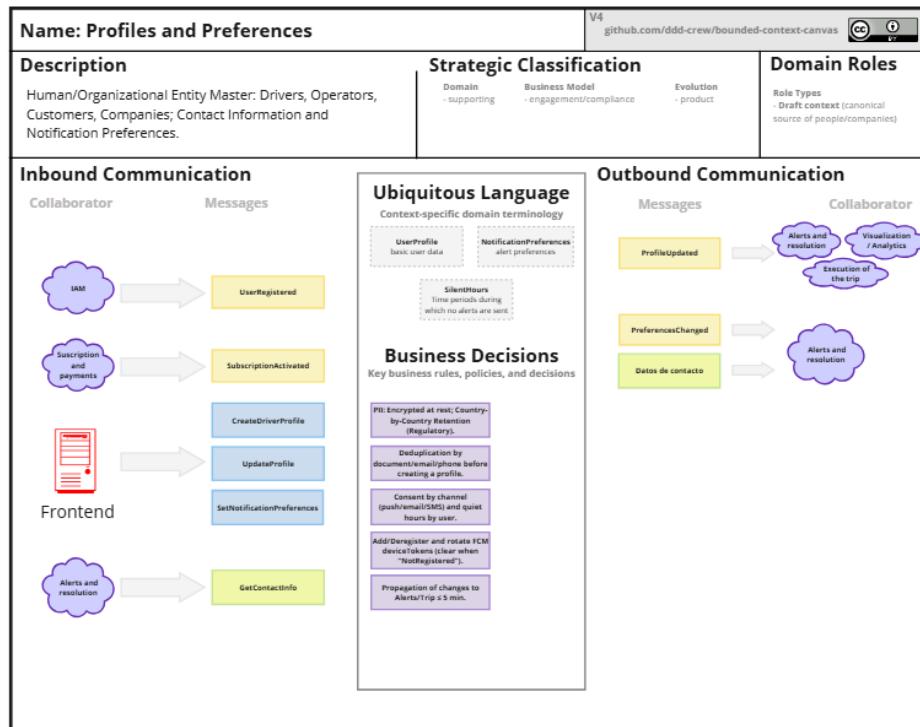
- **Identity and Access Management**
- **Profiles and Preferences Management**
- **Fleet Management**
- **Trip management**
- **Real-time monitoring**
- **Alerts and resolution**
- **Visualization/Analytics**
- **Subscriptions and payments**

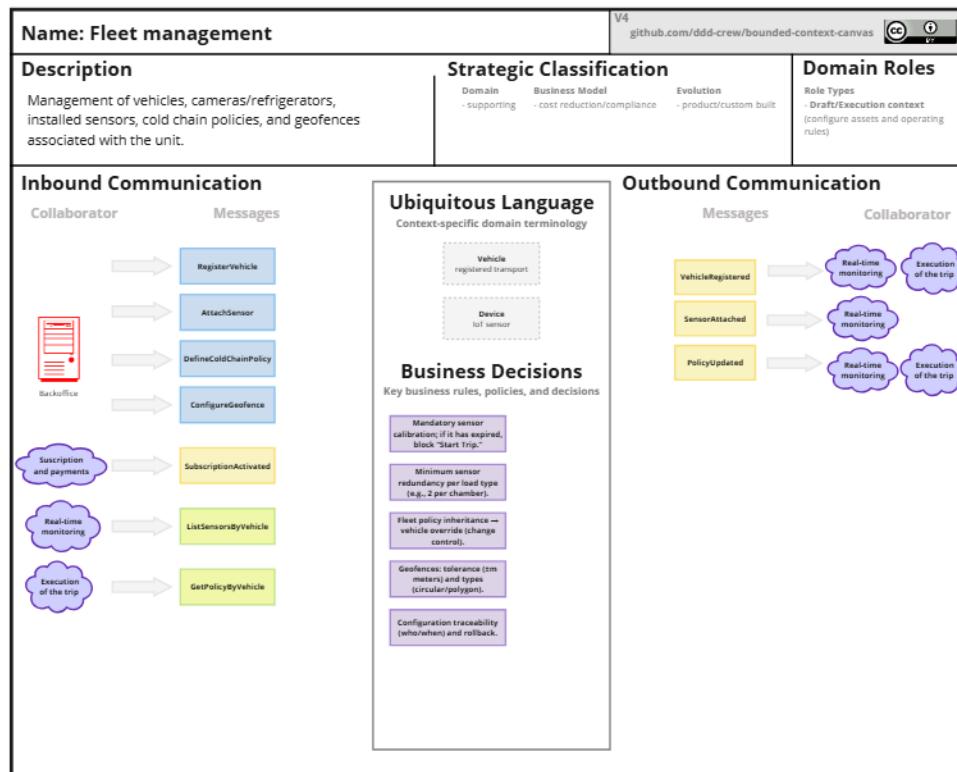
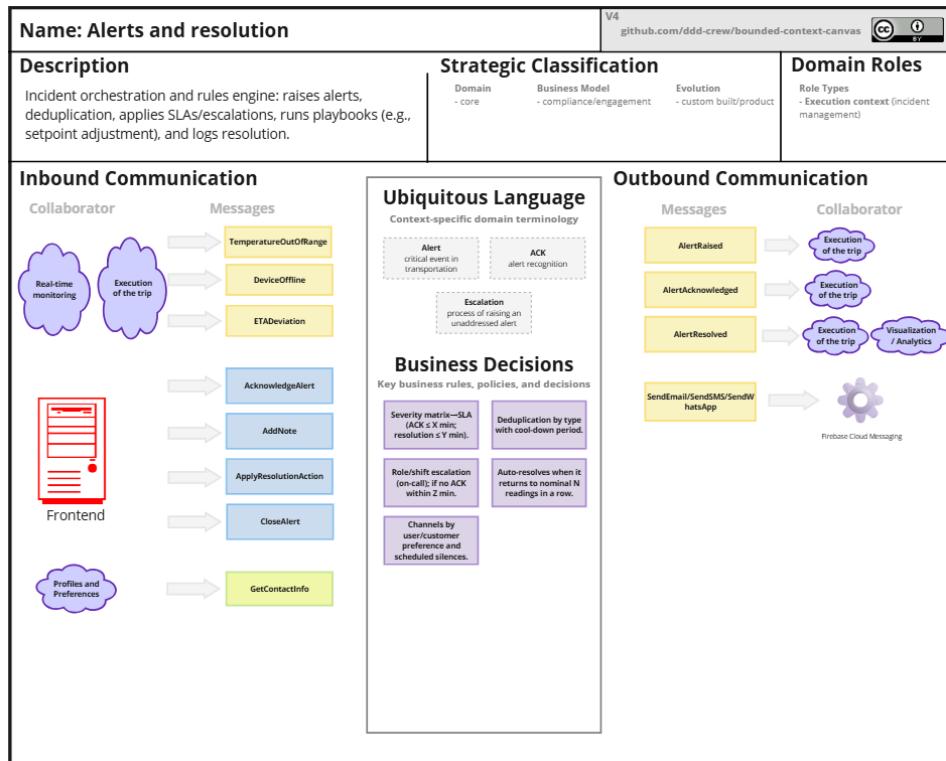
Este ejercicio permitió comprender cómo un evento local en un contexto puede impactar en otros, asegurando la trazabilidad del negocio y la correcta interacción entre los distintos módulos de la solución.

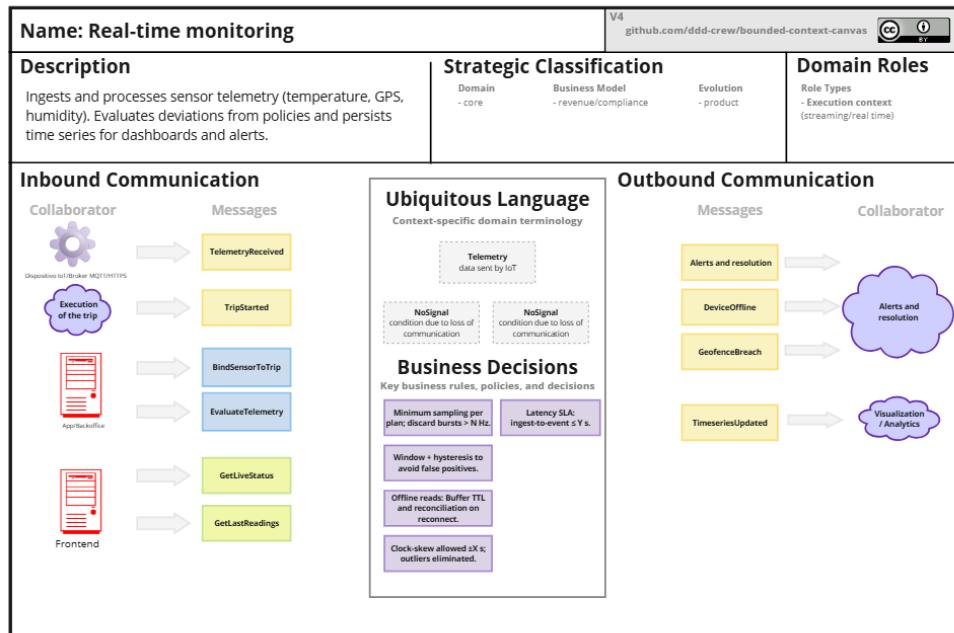
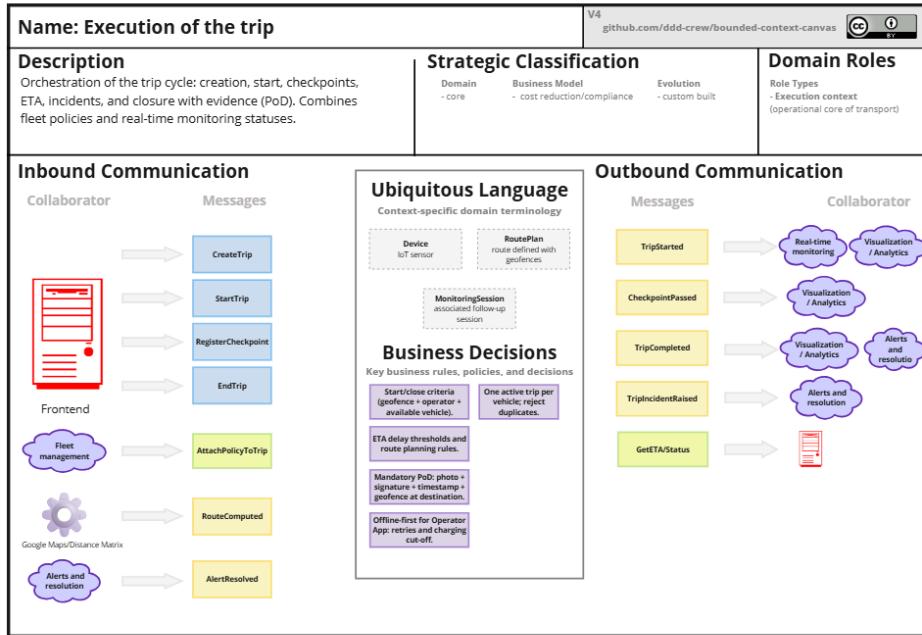
### 4.1.1.3. Bounded Context Canvases

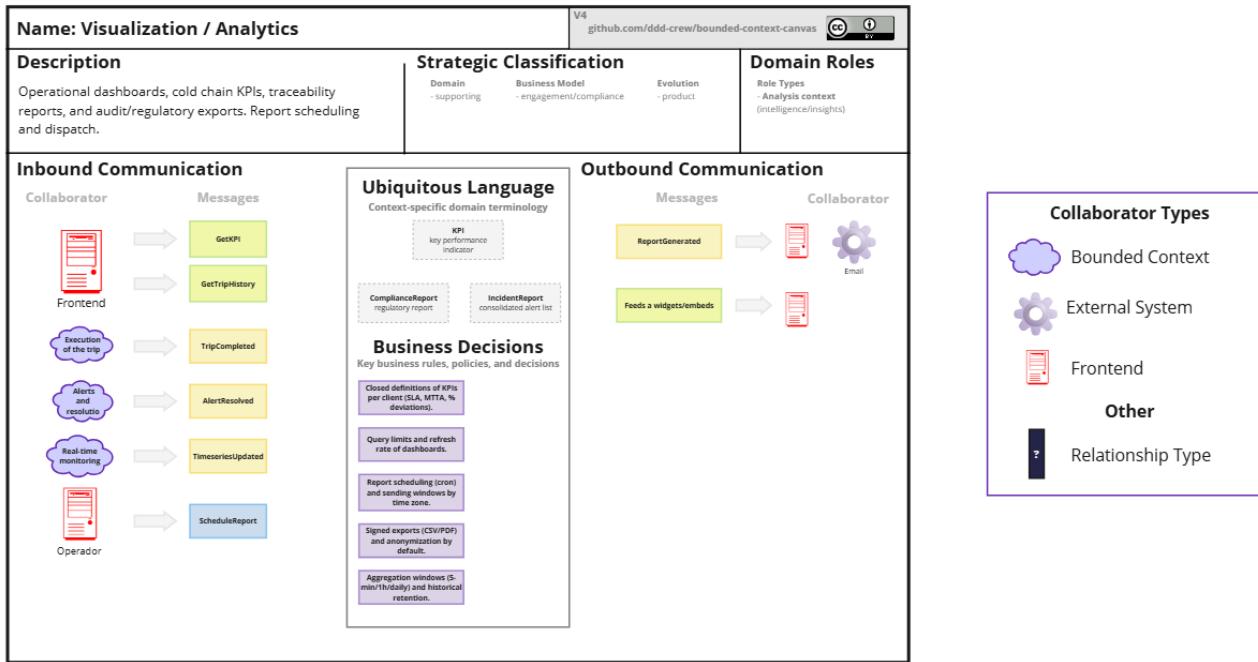
En esta sección se elaboraron los Bounded Context Canvases de CargaSafe para los ocho contextos identificados. El objetivo fue delimitar con precisión responsabilidades, lenguaje ubicuo y decisiones de negocio, además de explicitar las comunicaciones (Queries, Commands y Events) y colaboradores (otros BC, sistemas externos y frontend). Cada canvas documenta: Descripción, Clasificación estratégica (core/supporting/generic), Rol de dominio (draft/execution/analysis/gateway), Inbound/Outbound communication, Ubiquitous Language, Business Decisions y Collaborators. Esta definición fija ownership de datos, reduce ambigüedades y prepara los contratos de integración que se implementarán en APIs y mensajería.











[Ver gráfico en Miro](#)

#### 4.1.2. Context Mapping

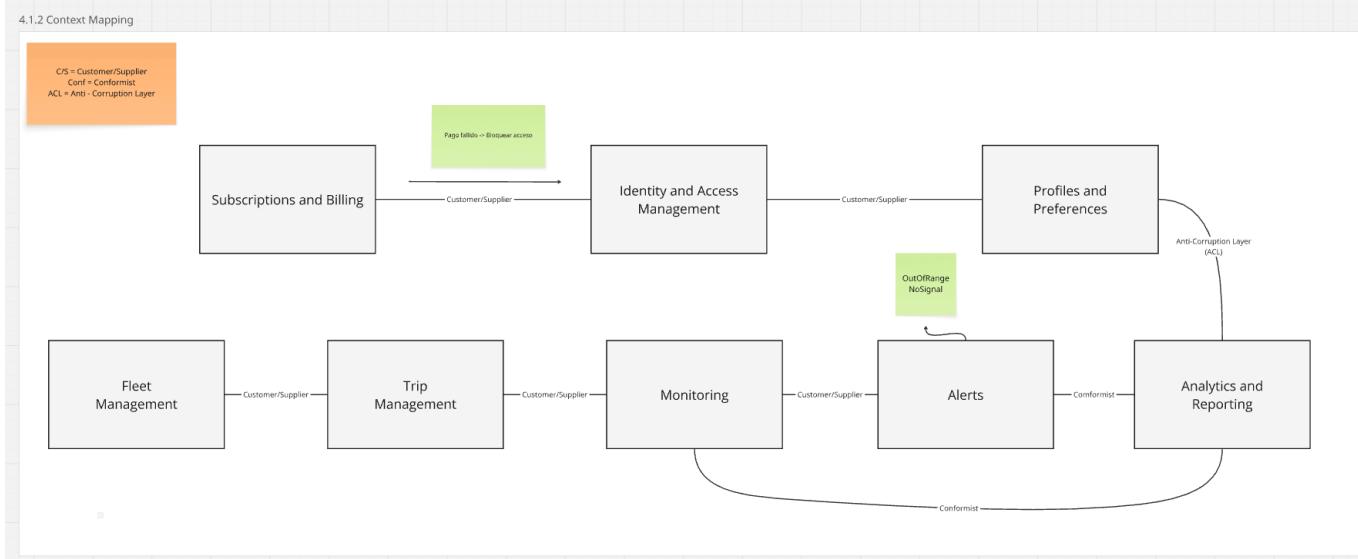
En esta etapa se construyó el **Context Map** de CargaSafe con los ocho bounded contexts identificados. El objetivo fue representar las **relaciones estructurales** entre ellos aplicando patrones de Domain-Driven Design como Customer/Supplier, Conformist y Anti-Corruption Layer (ACL).

#### Resultado

El mapa final permitió:

1. **Visualizar las dependencias entre contextos**, mostrando qué módulos proveen información y cuáles la consumen.
2. **Identificar los contextos core** (Trip Management, Monitoring, Alerts), los de soporte (Fleet, Profiles, Analytics) y los genéricos (IAM, Billing).
3. **Clasificar las relaciones:**
  - Customer/Supplier en la mayoría de flujos operativos (Billing → IAM, Trip → Monitoring, Monitoring → Alerts).
  - Conformist en el consumo de datos por Analytics.
  - Anti-Corruption Layer en la interacción Analytics → Profiles.

De esta manera, el Context Mapping consolida una visión global del sistema, mostrando cómo los distintos contextos colaboran para dar soporte al negocio.



#### 4.1.3. Software Architecture

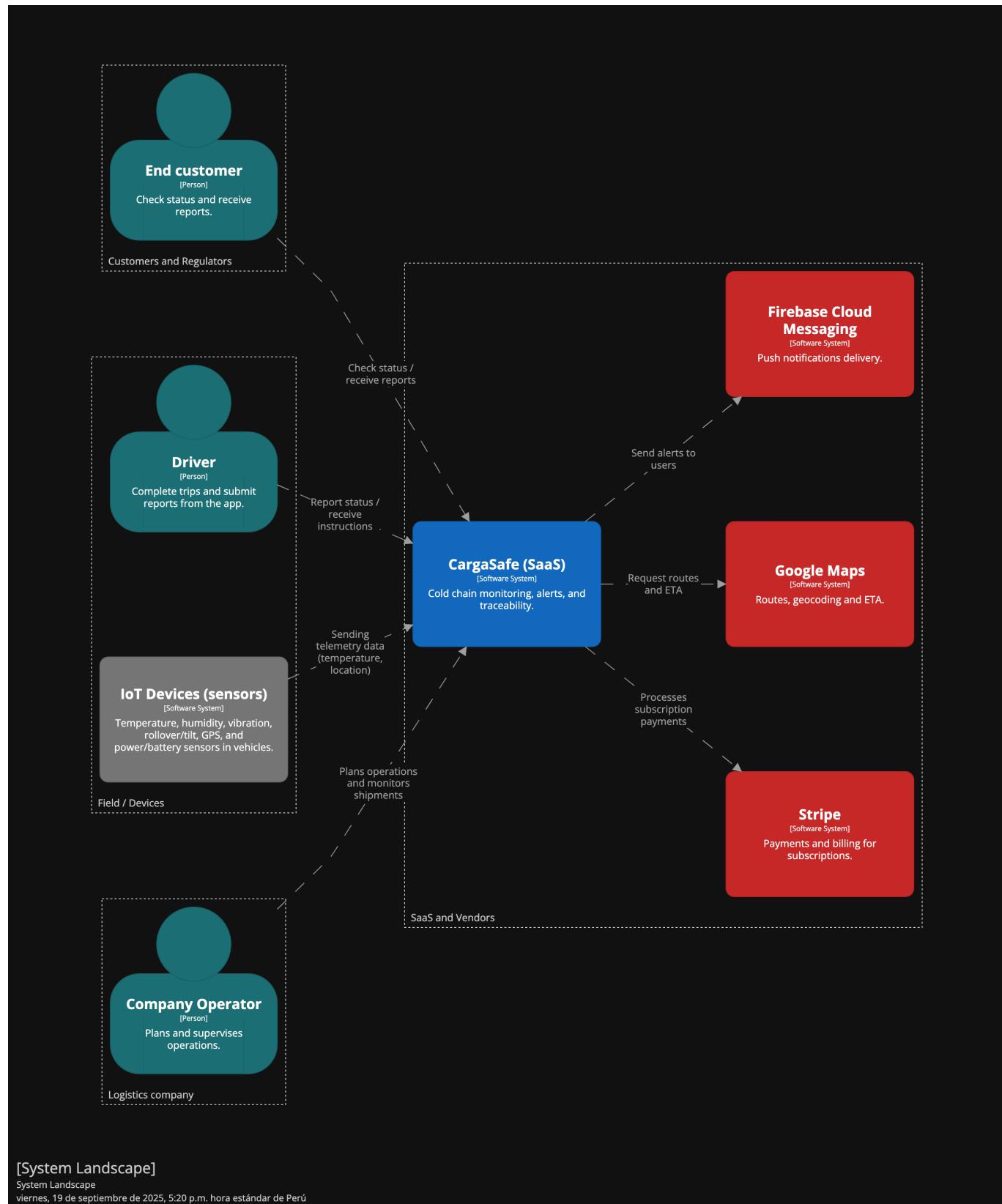
##### 4.1.3.1. Software Architecture System Landscape Diagram

El **System Landscape Diagram** ofrece una visión de alto nivel del **ecosistema empresarial** en el que se integra CargaSafe. Este diagrama no se centra únicamente en un sistema, sino que representa **todas las personas y sistemas de software relevantes**, tanto internos como externos, que participan en la operación logística.

#### Propósito

El objetivo de este diagrama es:

1. Mostrar el alcance de la organización y cómo conviven sus distintos sistemas.
2. Identificar a las **personas, sistemas internos, SaaS externos y proveedores** que colaboran en la cadena de valor.
3. Resaltar cómo **CargaSafe (SaaS)** se conecta dentro de este panorama, en interacción con otros actores y servicios.



## Elementos incluidos

- **Personas:** Company Operator, Driver and End Customer.
- **Sistemas internos:** Logistics Planning and Power BI Data.
- **Sistemas y proveedores externos:** CargaSafe (SaaS), Stripe, Google Maps, Notification Services e IoT Devices (sensors).
- **Grupos:** Se organizaron en cuatro dominios principales:
  - Logistics company

- Field / Devices
- Customers and Regulators
- SaaS and Vendors

## Relaciones principales

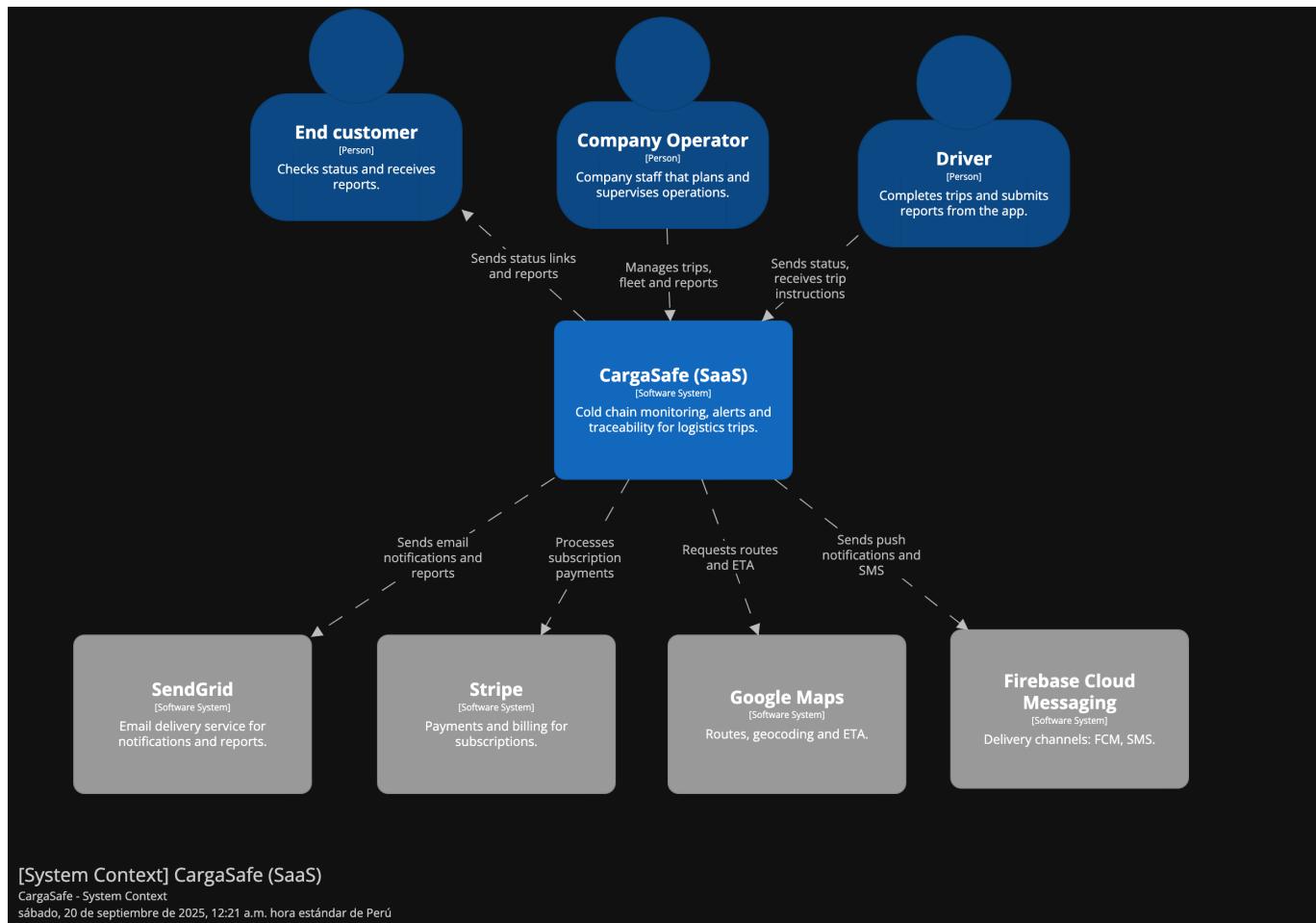
- Logistics Planning → CargaSafe (SaaS): exporta planes y asignaciones de viaje.
- IoT Devices → CargaSafe (SaaS): envía telemetría (temperatura, humedad, vibración, volcado/inclinación, GPS, energía/baterías).
- CargaSafe (SaaS) → Google Maps: consulta rutas y tiempos estimados.
- CargaSafe (SaaS) → Notification Services: envía alertas a los usuarios.
- CargaSafe (SaaS) → Stripe: procesa pagos de suscripción.
- CargaSafe (SaaS) → Power BI Data: exporta datasets consolidados para analítica.
- Company Operator / Driver ↔ CargaSafe (SaaS): planifican, ejecutan y reportan el estado operativo.
- End customer ← CargaSafe (SaaS): consulta estado y recibe reportes.

## Resultado

El diagrama muestra a CargaSafe (SaaS) como el núcleo de integración entre operaciones (Company Operator, Driver, Logistics Planning), telemetría IoT (sensores en campo) y servicios externos (ruteo, notificaciones y pagos), además de su aporte a la inteligencia de negocio mediante Power BI Data. Esta representación proporciona una visión clara e integral de las dependencias y colaboraciones que sustentan la operación logística y la gestión de la cadena de frío.

### 4.1.3.2. Software Architecture Context Level Diagrams

El **Context Diagram** de CargaSafe muestra una visión de alto nivel del sistema y de cómo se relaciona con los actores humanos y los sistemas externos que lo rodean.



En el centro se ubica CargaSafe (SaaS), que representa el sistema principal encargado del monitoreo de la cadena de frío, la trazabilidad y la generación de alertas en los viajes logísticos.

Alrededor del sistema se identifican los siguientes actores:

- *Company Operator*: gestiona viajes, flota y reportes desde la plataforma.
- *Driver*: completa viajes y reporta información desde la aplicación móvil.
- *End customer*: recibe enlaces de estado, alertas y reportes generados por el sistema.

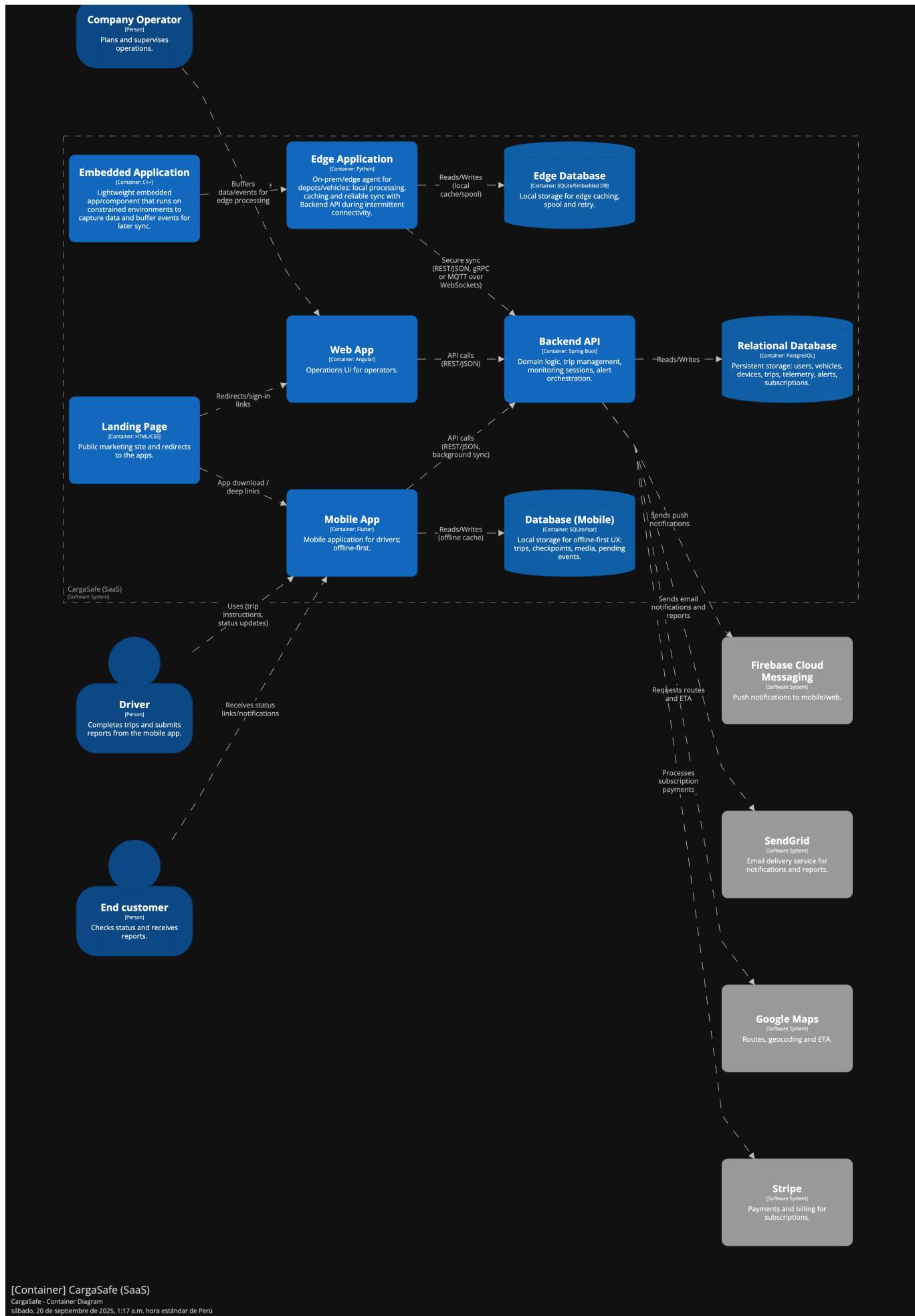
Asimismo, se destacan las interacciones con sistemas externos que complementan las funcionalidades de CargaSafe:

- Google Maps: provee rutas, geocodificación y cálculo de ETA.
- Firebase Cloud Messaging: entrega notificaciones push. Stripe: procesa pagos y facturación de suscripciones. Este diagrama permite visualizar de manera clara las responsabilidades de cada actor y sistema, y cómo CargaSafe se convierte en el núcleo que articula la comunicación entre usuarios, dispositivos IoT y servicios externos, garantizando la operación eficiente y segura de la cadena logística.

#### 4.1.3.2. Software Architecture Container Level Diagrams

En esta parte expandimos el sistema **CargaSafe (SaaS)** para mostrar sus contenedores internos, las tecnologías que utilizamos y cómo se comunican entre sí y con los sistemas externos.





[Container] CargaSafe (SaaS)  
CargaSafe - Container Diagram  
sábado, 20 de septiembre de 2025, 1:17 a.m. hora estándar de Perú

El diagrama de contenedores muestra cómo se organiza internamente CargaSafe (SaaS) y cómo se relaciona con los actores y sistemas externos.

Dentro de la plataforma tenemos varios contenedores:

- *Landing Page*: sitio público que sirve para marketing y como punto de acceso, redirigiendo tanto a la Web App, al Single Web como a la Mobile App (descarga o deeplinks).
- *Web Frontend*: aplicación usada por los operadores para gestionar viajes, flota y reportes.
- *Single Web*: vista pública en línea donde los clientes finales pueden consultar estados y reportes sin necesidad de autenticarse.
- *Mobile App*: aplicación móvil para los conductores, con soporte offline-first. Se conecta a su propia base de datos embebida SQLite para cache y operación sin conexión.
- *Backend API*: núcleo de la lógica de negocio, responsable de gestionar viajes, monitoreo, alertas y suscripciones.
- *Relational Database (PostgreSQL)*: base de datos principal donde se almacenan usuarios, vehículos, dispositivos, viajes, telemetría, alertas y suscripciones.
- *Edge Application (Python)*: agente que corre en instalaciones o vehículos, con capacidad de procesamiento local, cache y sincronización confiable con el backend. Usa su propia Edge Database local para tolerar desconexiones.
- *Embedded Application (C++)*: componente ligero que corre en dispositivos restringidos, captura datos y los envía hacia la aplicación edge para su posterior sincronización.

Los actores principales interactúan con los contenedores:

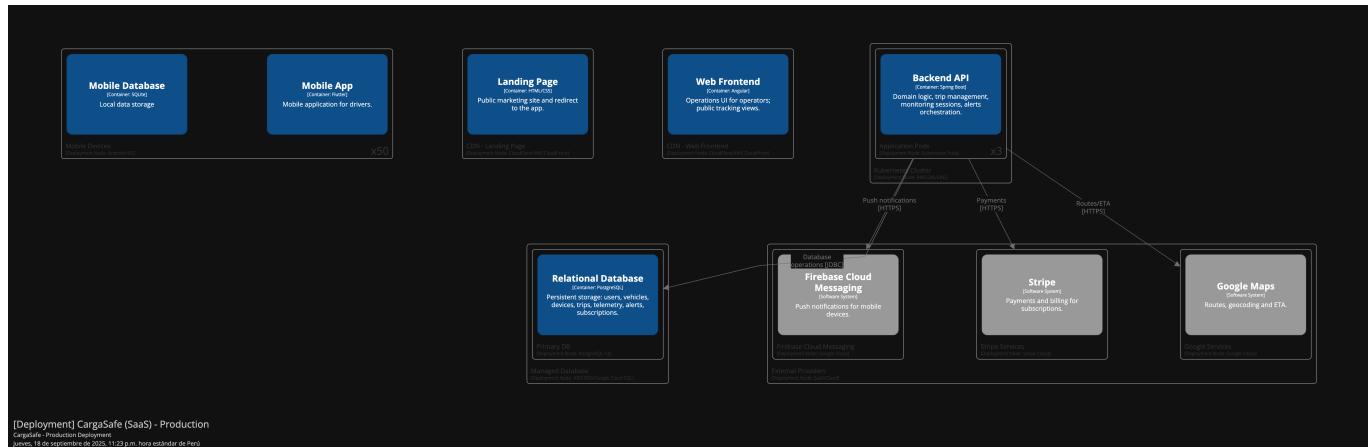
- Company Operator usa la Web App para planificar y supervisar operaciones.
- Driver utiliza la Mobile App para recibir instrucciones y reportar estado de los viajes.
- End Customer accede tanto a la Single Web (para reportes públicos) como a la Mobile App (para recibir notificaciones y links de estado).

Además, CargaSafe se integra con varios sistemas externos:

- *Google Maps*: para rutas, geocodificación y cálculo de ETA.
- *Stripe*: para pagos y facturación de suscripciones.
- *Firebase Cloud Messaging (FCM)*: para notificaciones push hacia aplicaciones móviles y web. En conjunto, el diagrama muestra cómo CargaSafe se estructura en contenedores especializados que soportan las necesidades de operadores, conductores y clientes, asegurando tanto la operación online como offline en distintos puntos de la cadena logística.

#### 4.1.3.3. Software Architecture Deployment Diagrams

El Deployment Diagram de CargaSafe muestra cómo se despliega la solución en un entorno de producción real, representando los nodos de infraestructura, los contenedores de software y las interacciones entre ellos.



## Clientes:

- Los usuarios finales acceden desde navegadores web, donde la Landing Page y el Web Frontend se sirven por separado desde CDNs independientes (CloudFlare/AWS CloudFront) para optimizar la entrega de contenido.
- Los conductores utilizan una aplicación móvil Flutter en dispositivos Android/iOS, que incluye una base de datos SQLite local para almacenamiento offline y sincronización de datos.
- Todas las peticiones de API se realizan mediante HTTPS y son redirigidas hacia el Load Balancer, encargado de enrutar el tráfico hacia los servicios backend.

## Backend y orquestación

- El Backend API (Spring Boot) se despliega dentro de un Kubernetes Cluster en múltiples pods de aplicaciones para alta disponibilidad y escalabilidad.
- El backend centraliza la lógica de negocio, gestiona operaciones de viajes, monitoreo de cadena de frío y orquestación de alertas en tiempo real.

## Base de datos

- El sistema utiliza una base de datos PostgreSQL gestionada (AWS RDS/Google Cloud SQL), con una instancia primaria para operaciones de escritura y réplicas de solo lectura para consultas distribuidas y balanceo de carga.
- Los dispositivos móviles mantienen datos críticos localmente en SQLite para funcionamiento offline durante los viajes.

**Integraciones externas** El backend consume servicios de terceros para extender sus capacidades:

- Google Maps para rutas, geocodificación y cálculo de ETA en tiempo real.
- Stripe para procesamiento de pagos y facturación de subscripciones.
- Firebase Cloud Messaging (FCM) para la entrega de notificaciones push directamente a los dispositivos móviles de los conductores.

**Resultado** El diagrama de despliegue muestra que la solución CargaSafe está organizada bajo una arquitectura cloud-native optimizada, con:

- Separación de responsabilidades: Landing page y aplicación web servidas independientemente
- Capacidades offline: Base de datos local SQLite en dispositivos móviles
- Kubernetes para la orquestación de contenedores del backend

- CDNs separados para optimizar la entrega de contenido estático
- Base de datos gestionada con réplicas para mejorar el rendimiento y disponibilidad
- Notificaciones push nativas a través de FCM

Esta infraestructura permite un sistema escalable, resiliente y con capacidades offline críticas para la operación de conductores en campo, garantizando la continuidad operativa en la gestión de la cadena de frío incluso sin conectividad permanente.

## 4.2. Tactical-Level Domain-Driven Design

### 4.2.1. Bounded Context: Identity and Access Management

#### 4.2.1.1. Domain Layer

##### Entidades Principales

###### User (Aggregate Root)

- **Propósito:** Representa un usuario del sistema con capacidades de autenticación y autorización
- **Atributos principales:**
  - `id`: Identificador único
  - `username`: Nombre de usuario único
  - `email`: Correo electrónico único
  - `passwordHash`: Hash seguro de la contraseña
  - `firstName, lastName`: Datos personales
  - `isEnabled`: Estado activo/inactivo
  - `createdAt, updatedAt`: Timestamps de auditoría
- **Métodos principales:**
  - `authenticate(password)`: Valida credenciales
  - `changePassword(oldPassword, newPassword)`: Cambia contraseña con validación
  - `assignRole(role)`: Asigna rol al usuario
  - `enable(), disable()`: Gestión de estado

###### Role (Entity)

- Propósito: Define roles y permisos en el sistema
- Atributos principales:
  - `id`: Identificador único
  - `name`: Nombre del rol (ADMIN, LOGISTICS\_MANAGER, END\_CUSTOMER)
  - `description`: Descripción del rol
  - `permissions`: Lista de permisos asociados
- **Métodos principales:**
  - `hasPermission(permission)`: Verifica si el rol tiene un permiso específico
  - `addPermission(permission)`: Agrega permiso al rol

###### Token (Entity)

- **Propósito:** Gestiona tokens de acceso y refresh tokens
- **Atributos principales:**
  - `id`: Identificador único

- `token`: Token JWT
- `userId`: Referencia al usuario
- `expiryDate`: Fecha de expiración
- `isRevoked`: Estado de revocación
- **Métodos principales:**
  - `isExpired()`: Verifica si el token ha expirado
  - `revoke()`: Revoca el token

## Value Objects

- **Email**: Valida formato de correo electrónico
- **Password**: Encapsula reglas de contraseñas seguras
- **Permission**: Representa un permiso específico (recurso + acción)
- **TokenClaims**: Información contenida en el JWT

## Domain Services

- **PasswordService**: Gestión de hash y validación de contraseñas
- **TokenService**: Generación y validación de tokens JWT
- **AuthorizationService**: Lógica de autorización basada en roles

## Commands

- **LoginCommand**: Comando para autenticación
- **RegisterUserCommand**: Comando para registro de usuario
- **ChangePasswordCommand**: Comando para cambio de contraseña
- **AssignRoleCommand**: Comando para asignación de roles

## Queries

- **GetUserByIdQuery**: Obtiene usuario por ID
- **GetUserByEmailQuery**: Obtiene usuario por email
- **GetUserRolesQuery**: Obtiene roles de un usuario

## Events

- **UserRegisteredEvent**: Usuario registrado exitosamente
- **UserLoggedInEvent**: Usuario autenticado
- **PasswordChangedEvent**: Contraseña cambiada
- **UserDisabledEvent**: Usuario deshabilitado

## 4.2.1.2. Interface Layer

### Controllers Principales

#### AuthController

- `POST /auth/login`: Autenticación de usuarios
- `POST /auth/logout`: Cierre de sesión
- `POST /auth/refresh`: Renovación de tokens
- `POST /auth/forgot-password`: Solicitud de recuperación de contraseña

## UserController

- `POST /users/register`: Registro de nuevos usuarios
- `GET /users/profile`: Obtiene perfil del usuario actual
- `PUT /users/profile`: Actualiza perfil del usuario
- `PUT /users/change-password`: Cambio de contraseña
- `GET /users/{id}`: Obtiene usuario por ID (solo admins)

### 4.2.1.3. Application Layer

#### Command Services

##### UserCommandService

- Maneja comandos de escritura para usuarios
- Coordina operaciones de creación, actualización y eliminación
- Publica eventos de dominio correspondientes

##### AuthCommandService

- Gestiona procesos de autenticación y autorización
- Maneja tokens y sesiones de usuario
- Coordina flujos de recuperación de contraseña

#### Query Services

##### UserQueryService

- Proporciona consultas de solo lectura para usuarios
- Optimizado para vistas y reportes
- Maneja proyecciones de datos de usuario

##### AuthQueryService

- Consultas relacionadas con autenticación
- Validación de tokens y permisos
- Información de sesiones activas

#### Event Handlers

##### UserRegisteredEventHandler

- Procesa eventos de registro de usuario
- Envía emails de bienvenida
- Configura datos iniciales del usuario

### 4.2.1.4. Infrastructure Layer

#### Repositories

##### UserRepository (implementa IUserRepository)

- Persistencia y consulta de datos de usuarios
- Implementación con Spring Data JPA
- Operaciones CRUD optimizadas

#### **RoleRepository** (implementa IRoleRepository)

- Gestión de roles y permisos
- Consultas para autorización
- Cache de roles frecuentemente usados

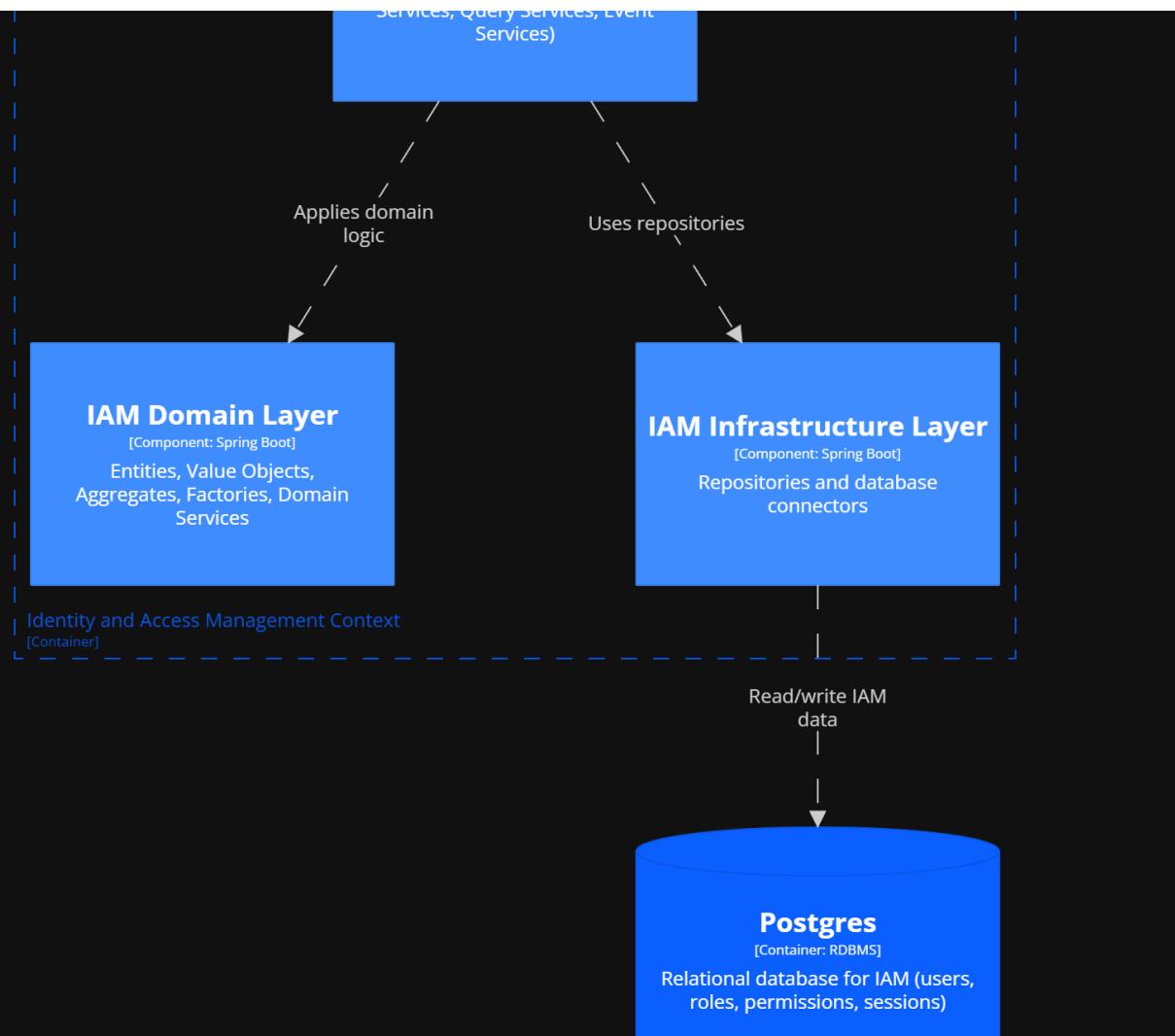
#### **TokenRepository** (implementa ITokenRepository)

- Gestión de tokens JWT
- Limpieza automática de tokens expirados
- Blacklist de tokens revocados

#### **4.2.1.5. Bounded Context Software Architecture Component Level Diagrams**

Diagrama de Componentes - Backend - Identity and Access Management



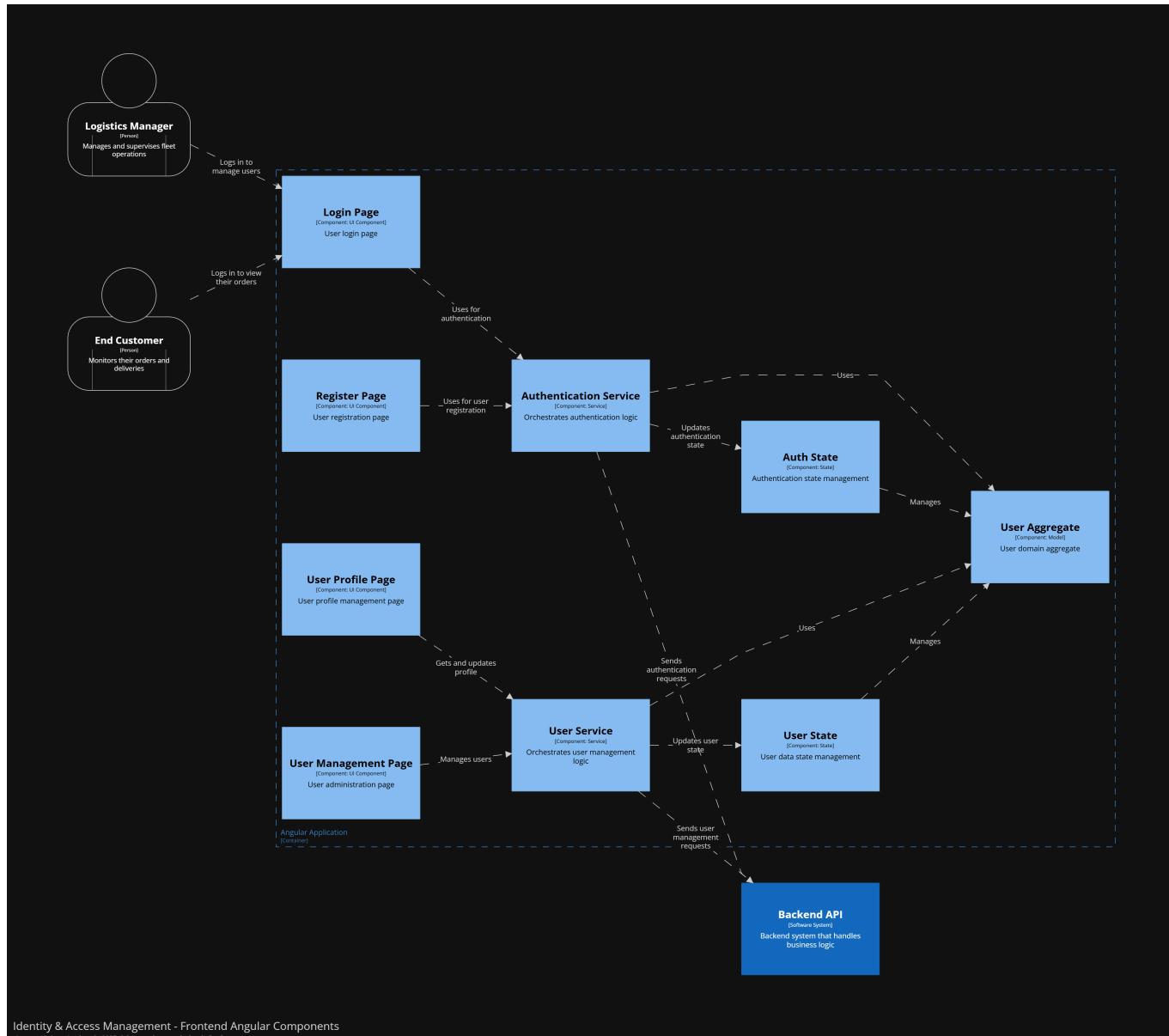


## Identity and Access Management Context - Component Diagram

Internal layered components of the IAM Context inside CargaSafe.  
viernes, 19 de setiembre de 2025, 11:33 p. m. hora estándar de Perú

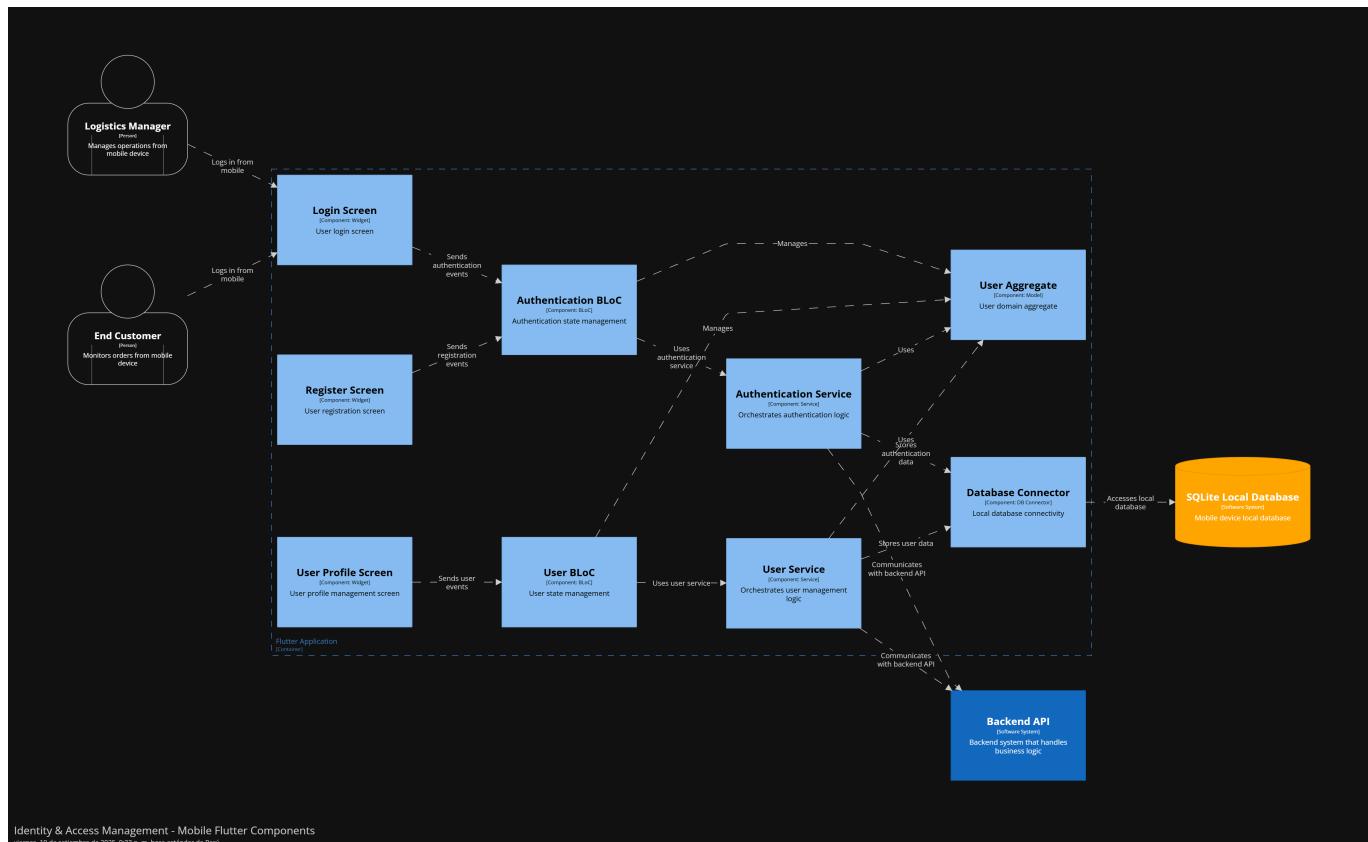
Este diagrama muestra la arquitectura por capas del bounded context IAM en el backend. La separación clara entre Interface, Application, Domain e Infrastructure layers permite un diseño mantenable y testeable. Los controllers en la Interface Layer reciben requests HTTP y delegan a los command/query services en Application Layer, que utilizan el dominio y persisten através de repositories en Infrastructure Layer.

## Diagrama de Componentes - Frontend Web - Identity and Access Management



El diagrama del frontend web muestra los componentes Angular organizados por responsabilidades. Las páginas (Login, Register, User Profile) interactúan con servicios que manejan la lógica de negocio y state management. La comunicación con el backend se realiza através de HTTP services que consumen la API REST.

## Diagrama de Componentes - Mobile - Identity and Access Management

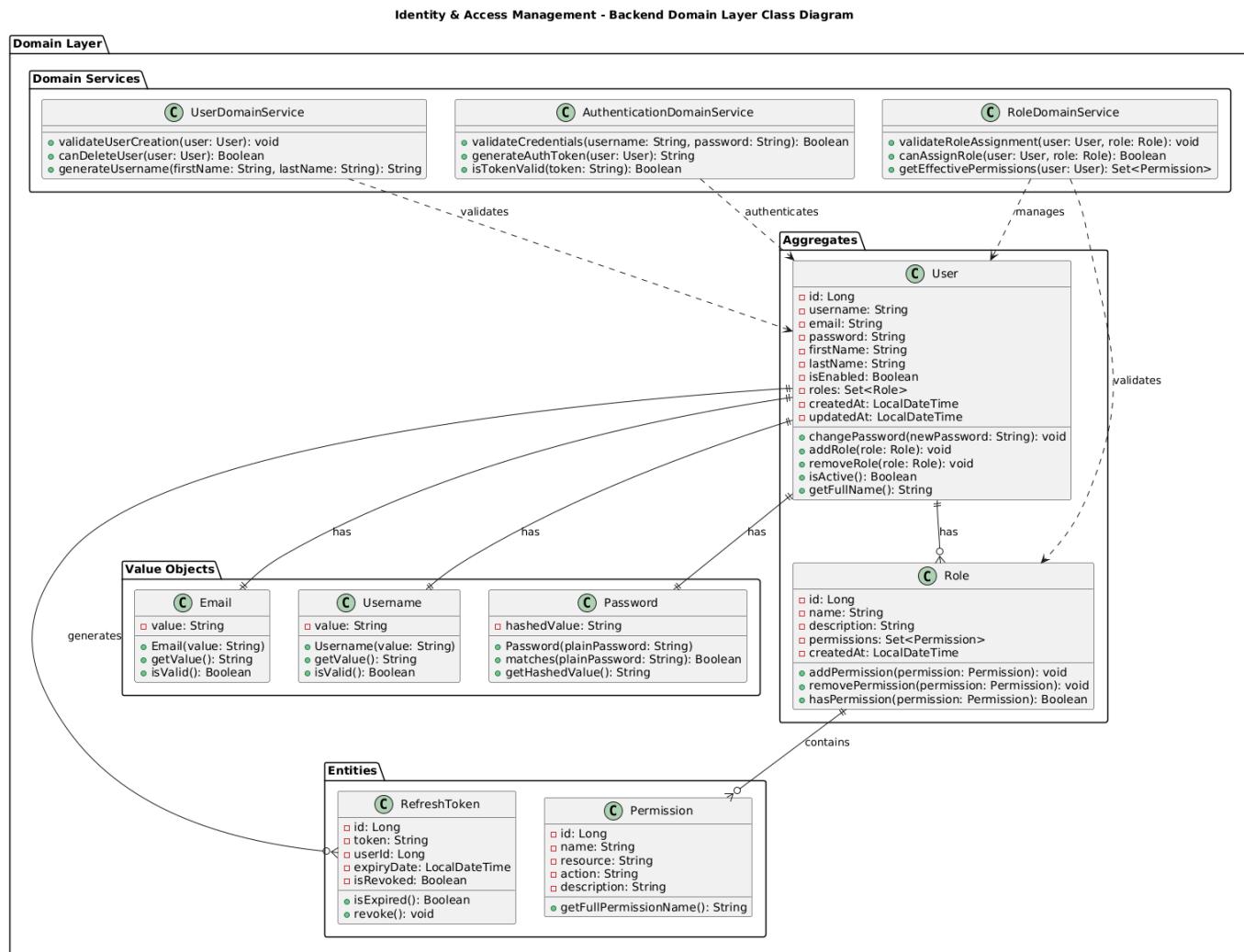


La aplicación móvil utiliza Flutter con arquitectura BLoC para state management. Las pantallas (screens) envían eventos a BLoCs que manejan el estado y coordinan con services. Los services se comunican tanto con el backend API como con la base de datos local SQLite para funcionalidad offline.

#### 4.2.1.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.1.6.1. Bounded Context Domain Layer Class Diagrams

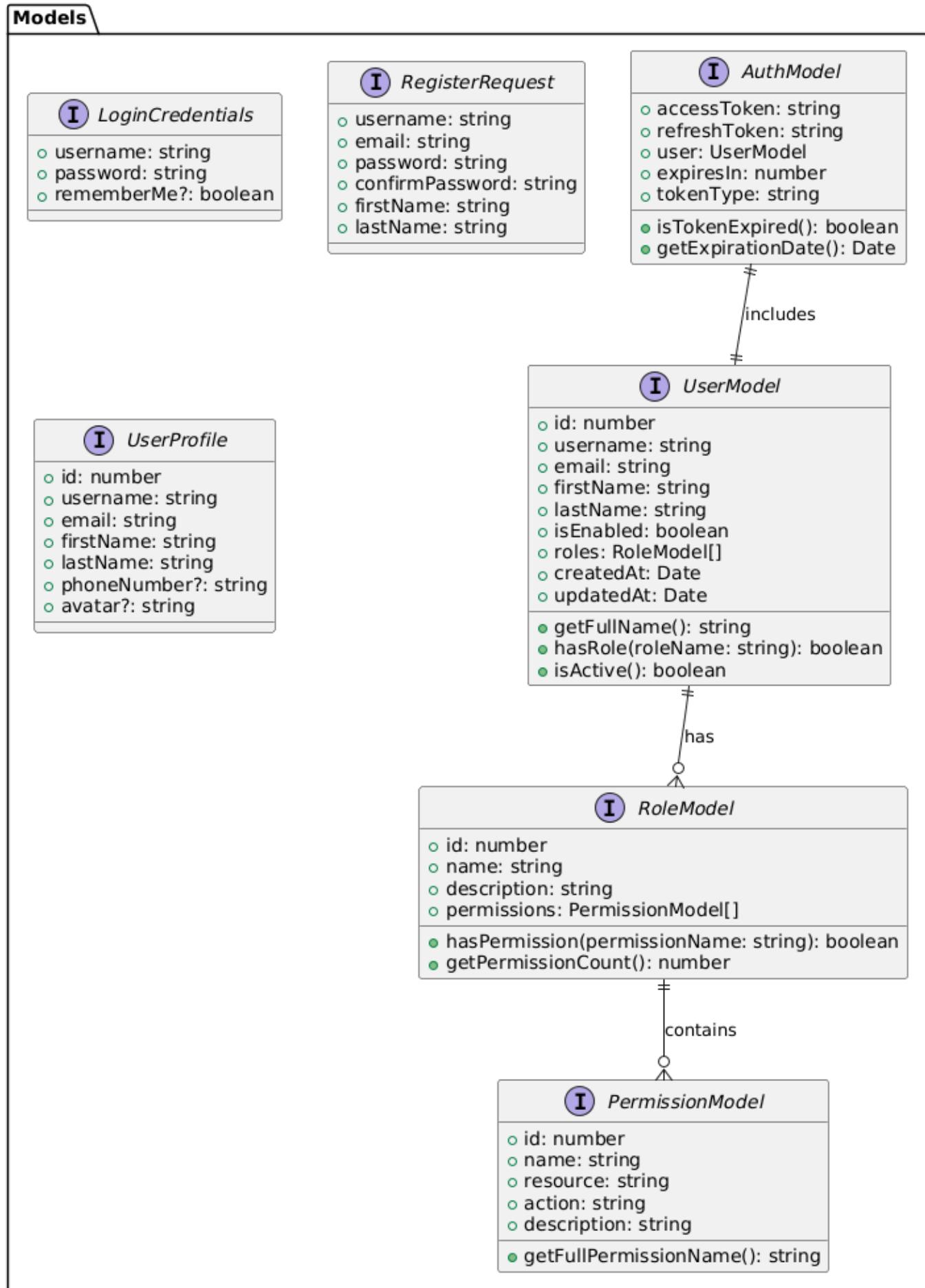
##### Backend - Identity & Access Management Domain Layer Class Diagram



El diagrama de clases del backend muestra las entidades principales del IAM bounded context en la capa de dominio. La entidad User actúa como aggregate root y maneja la lógica de autenticación y autorización. Los roles están conectados através de relaciones many-to-many con usuarios, mientras que los tokens gestionan las sesiones y refresh tokens. La estructura implementa el patrón Repository para la persistencia.

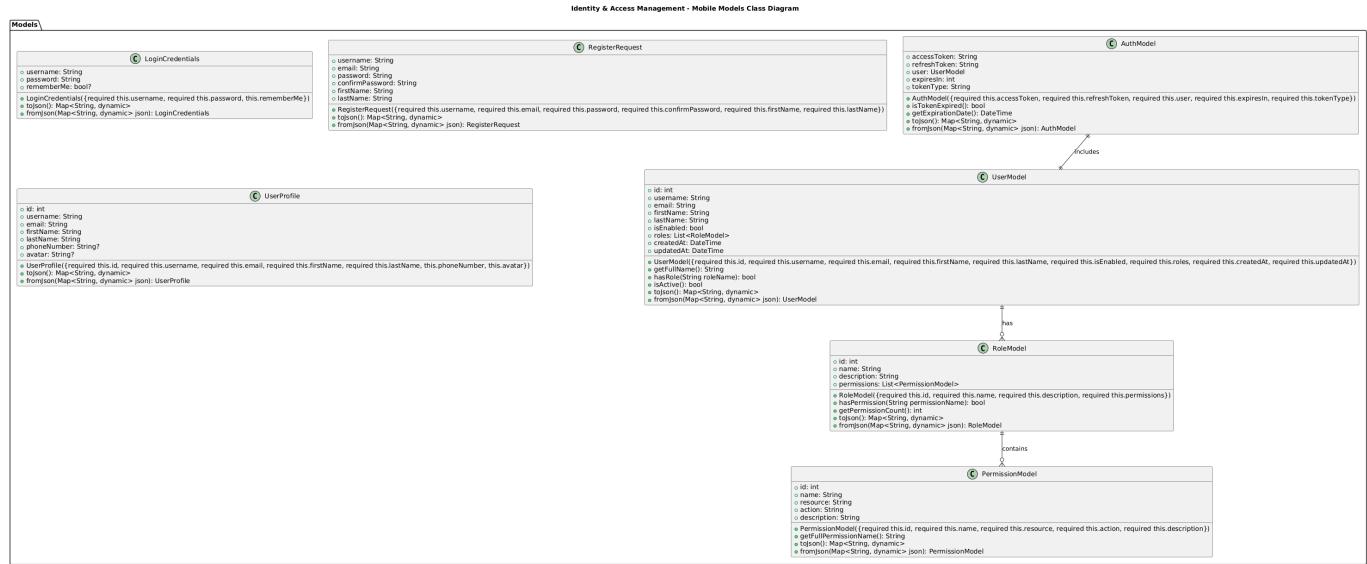
## Frontend - Identity & Access Management Domain Layer Class Diagram

## Identity & Access Management - Frontend Models Class Diagram



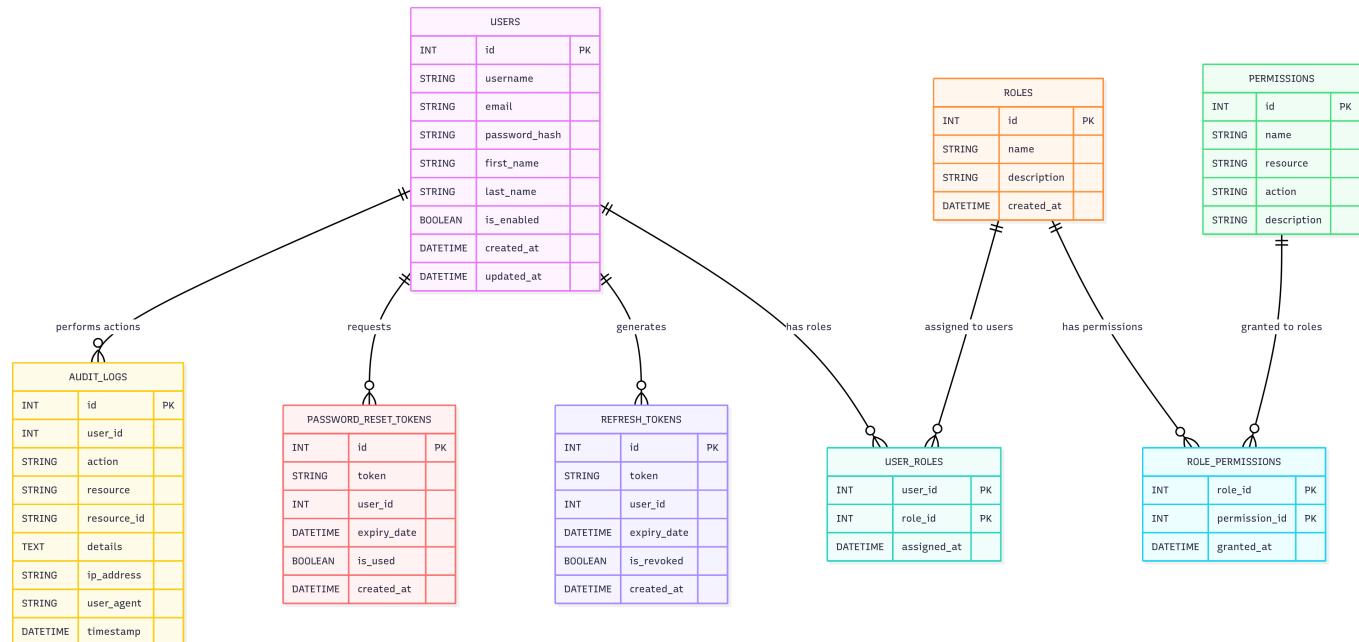
El diagrama del frontend Angular muestra la arquitectura de componentes y servicios para el manejo de identidad. Los componentes (Login, Register, Profile) interactúan con servicios específicos que manejan el estado de autenticación. El AuthService centraliza la lógica de comunicación con el backend API, mientras que los guards protegen las rutas según permisos.

## Mobile - Identity & Access Management Domain Layer Class Diagram



La aplicación móvil Flutter implementa BLoC pattern para el manejo de estado de autenticación. Los BLoCs (AuthBloc, UserBloc) coordinan entre las pantallas y los services, mientras que el local storage permite funcionalidad offline. La arquitectura asegura sincronización de credenciales entre la app y el backend.

### 4.2.1.6.2. Bounded Context Database Design Diagram



El diagrama de base de datos implementa un modelo RBAC (Role-Based Access Control) robusto. Las tablas principales (USERS, ROLES, PERMISSIONS) están conectadas a través de tablas de unión que permiten relaciones many-to-many. Se incluyen tablas auxiliares para tokens de sesión, logs de auditoría y tokens de recuperación de contraseña. La estructura está optimizada para consultas frecuentes de autorización y mantiene integridad referencial.

## 4.2.2. Bounded Context: *Subscriptions and Billing*

### 4.2.2.1. Domain Layer

#### *Entities*

#### **Subscription**

- **Propósito:** Gestionar el ciclo de vida de la suscripción de una empresa.
- **Atributos principales:** subscriptionId, companyId, plan, billingCycle, status (ACTIVE, CANCELED), startedAt, expiresAt.
- **Métodos principales:** activate(), changePlan(newPlan), renew(), cancel().

#### **Payment**

- **Propósito:** Representar pagos asociados a una suscripción.
- **Atributos principales:** paymentId, subscriptionId, amount, status (PENDING, SUCCEEDED, FAILED), date.
- **Métodos principales:** markSucceeded(), markFailed().

#### **Company**

- **Propósito:** Entidad que consume el servicio y depende de su suscripción activa.
- **Atributos principales:** companyId, name, vehicleCount.
- **Métodos principales:** canFitPlan(plan).

#### **Value Objects**

- **Plan:** Define límites y beneficios (code, vehicleLimit, price).
- **BillingCycle:** Periodo de facturación (type, startDate, endDate).
- **GracePeriod:** Tolerancia tras vencimiento (days).

#### **Domain Services**

- **BillingService:** Calcula montos y renovaciones.
- **PaymentPolicy:** Aplica reglas de activación y cancelación según pagos.

#### **Factory**

- **SubscriptionFactory:** Crea una suscripción válida con plan y ciclo inicial.

#### **Commands**

- **CreateSubscriptionCommand:** Crea una nueva suscripción.
- **ChangePlanCommand:** Cambia de plan.
- **CancelSubscriptionCommand:** Cancela una suscripción.
- **RenewSubscriptionCommand:** Renueva periodo.
- **RecordPaymentCommand:** Registra un pago.

#### **Queries**

**GetSubscriptionByIdQuery:** Consulta suscripción por ID. **GetActiveSubscriptionByCompanyQuery:** Consulta suscripción activa de una compañía. **ListPaymentsBySubscriptionQuery:** Lista pagos de una suscripción.

## Events

**SubscriptionCreated:** Suscripción creada. **PlanChanged:** Cambio de plan. **SubscriptionRenewed:** Renovación realizada. **SubscriptionCanceled:** Suscripción cancelada. **PaymentSucceeded / PaymentFailed:** Resultado de pago.

### 4.2.2.2. Interface Layer

#### Controllers

- **SubscriptionController:** Endpoints para crear, renovar, cambiar plan y cancelar suscripciones.
- **PaymentController:** Endpoints para registrar y consultar pagos.
- **PlanController:** Endpoints para listar planes disponibles.
- **CompanyAccessController:** Endpoints para consultar estado de acceso de una empresa.

### 4.2.2.3. Application Layer

#### Command Services

- **SubscriptionCommandService:** Ejecuta comandos de suscripción (crear, cambiar, renovar, cancelar).
- **PaymentCommandService:** Registra pagos y actualiza estado de suscripción.

#### Query Services

**SubscriptionQueryService:** Consulta suscripciones por id, estado o compañía. **PaymentQueryService:** Consulta pagos por suscripción o estado.

#### Event Handlers

- **SubscriptionEventHandler:** Reacciona a eventos de suscripción (creada, renovada, cancelada, cambio de plan).
- **PaymentEventHandler:** Reacciona a pagos exitosos o fallidos.

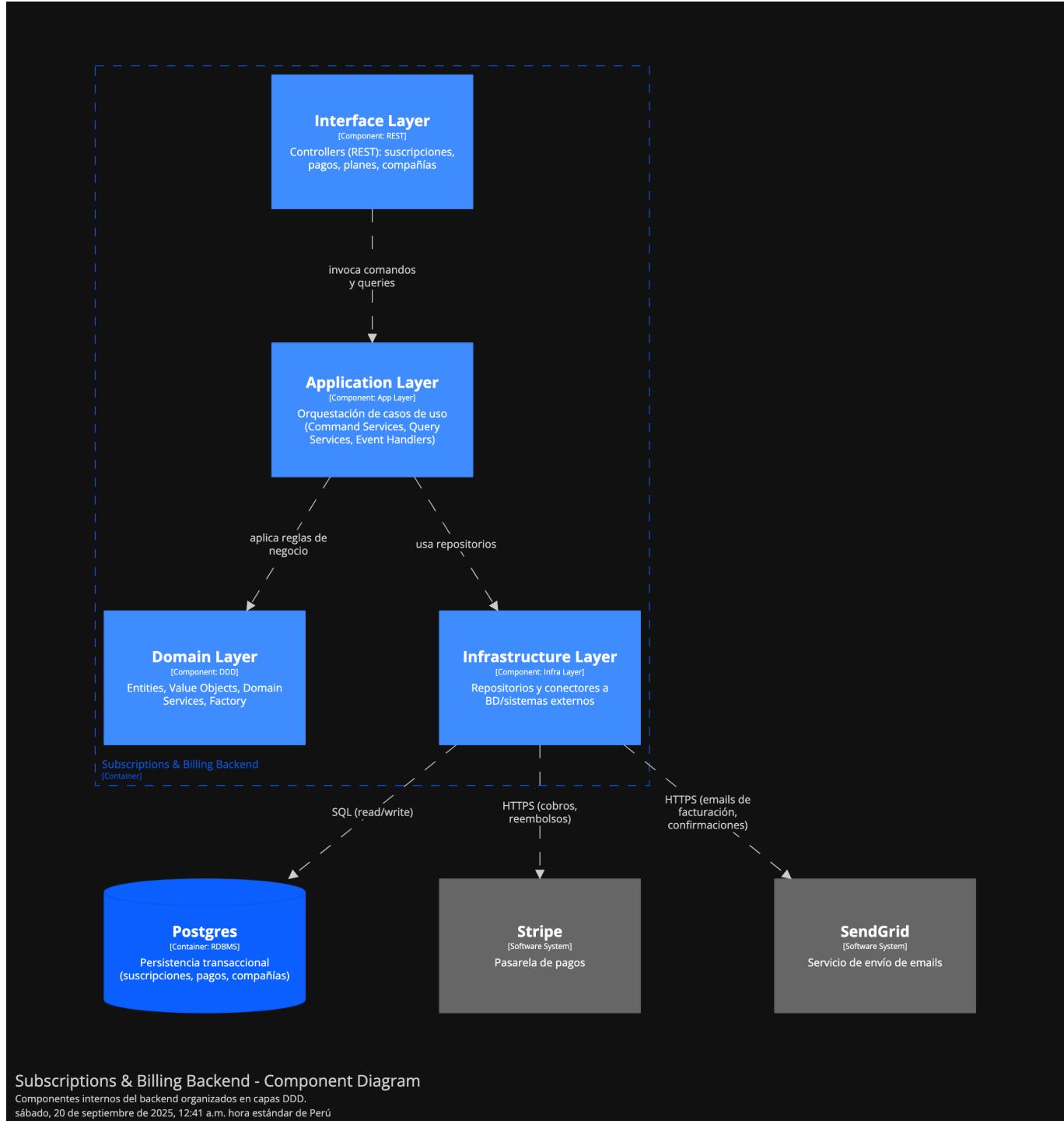
### 4.2.2.4. Infrastructure Layer

#### Repositories (Interfaces)

- **ISubscriptionRepository:** Acceso a datos de suscripciones.
- **IPaymentRepository:** Acceso a datos de pagos.
- **ICompanyRepository:** Acceso a datos de compañías.

### 4.2.2.5. Bounded Context Software Architecture Component Level Diagrams

*Diagrama de componentes - Backend - Subscriptions and Billing*



### Subscriptions & Billing Backend - Component Diagram

Componentes internos del backend organizados en capas DDD.  
sábado, 20 de septiembre de 2025, 12:41 a.m. hora estándar de Perú

El backend del bounded context de Suscripciones y Pagos está organizado en cuatro capas principales:

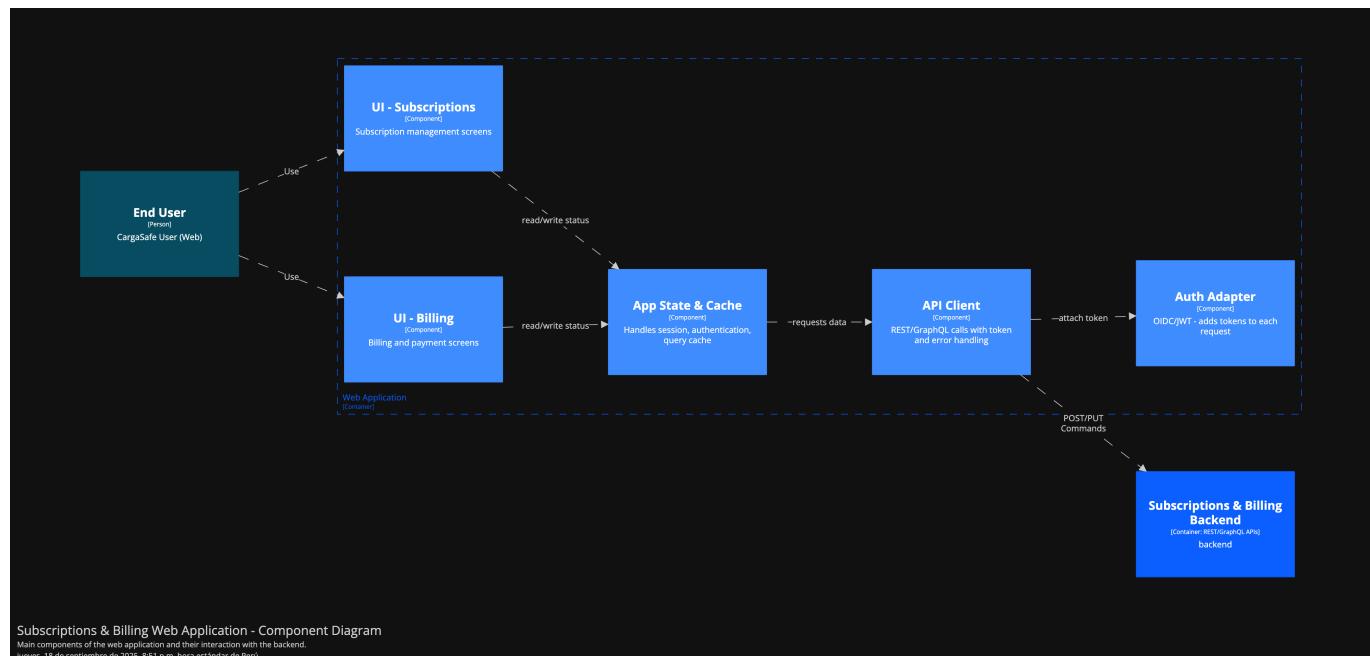
- **Interface Layer:** expone los controladores REST que atienden operaciones de suscripciones, pagos, planes y compañías. Es la puerta de entrada para los usuarios y sistemas que consumen la API.
- **Application Layer:** orquesta los casos de uso mediante Command Services, Query Services y Event Handlers. Aquí se coordinan las operaciones y se invocan las reglas de negocio.
- **Domain Layer:** concentra la lógica de negocio del contexto, con entidades, objetos de valor, servicios de dominio y fábricas. Define las reglas que rigen el ciclo de vida de suscripciones y pagos.
- **Infrastructure Layer:** implementa repositorios y conectores hacia la base de datos y sistemas externos. Se encarga de la persistencia y de la integración técnica.

Las conexiones externas son:

- Postgres para persistencia transaccional (suscripciones, pagos, compañías).

- Stripe para procesamiento de pagos.
- Firebase Cloud Messaging (FCM) para envío de notificaciones push.
- Google Maps para consultas de rutas y tiempos estimados (ETA).

#### Diagrama de componentes - Application Web - Subscriptions and Billing



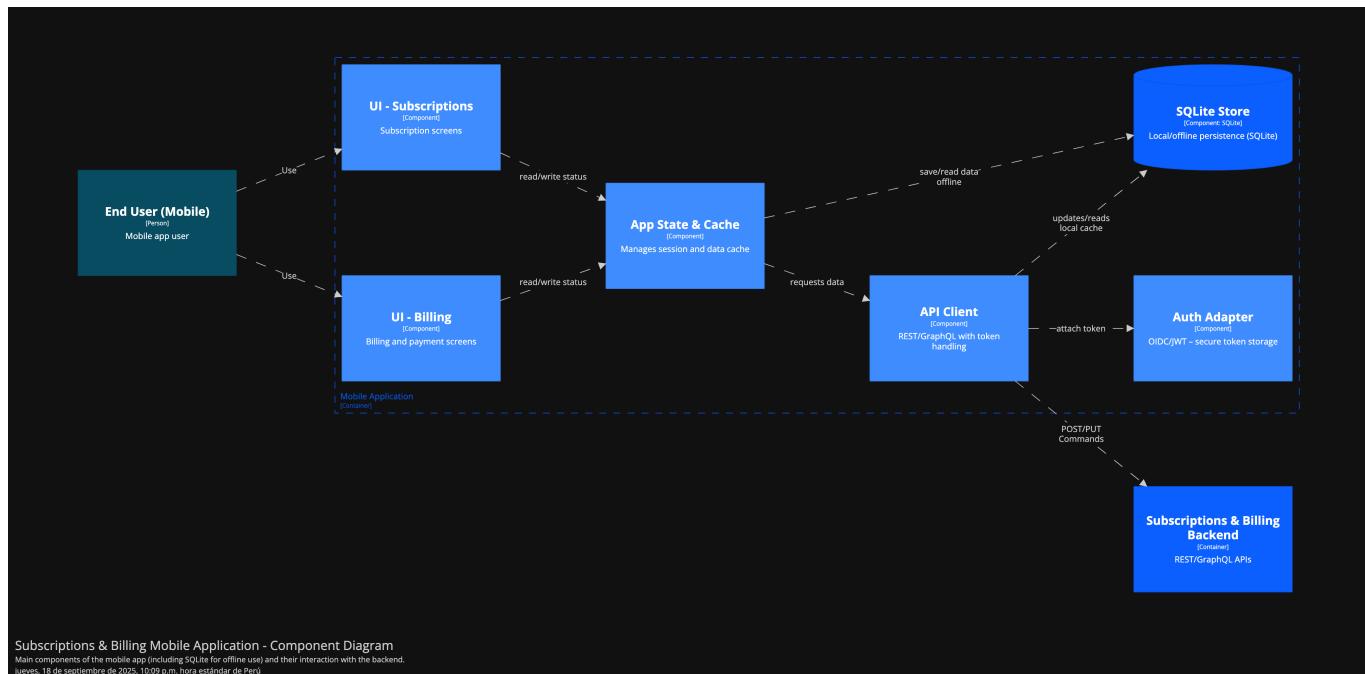
La aplicación web se conecta al bounded context **Subscriptions & Billing** únicamente a través de las APIs: la *Subscriptions API* (para enviar comandos como crear o cancelar una suscripción) y la *Query API* (para consultar datos como facturas o planes activos).

En el lado del cliente, la app se organiza en tres partes:

- **UI (interfaz de usuario)**: pantallas de suscripciones, facturación y pagos.
- **Estado de aplicación**: maneja la sesión del usuario, el cache de consultas y el control de autenticación.
- **Servicios de datos**: cliente HTTP que llama a las APIs, agrega el token de seguridad y gestiona reintentos o errores.

La aplicación web no implementa lógica de negocio propia, solo muestra la información y envía las intenciones del usuario al backend. Todo lo que es reglas, validaciones o persistencia está en el backend.

#### Diagrama de componentes - Mobile Application - Subscriptions and Billing

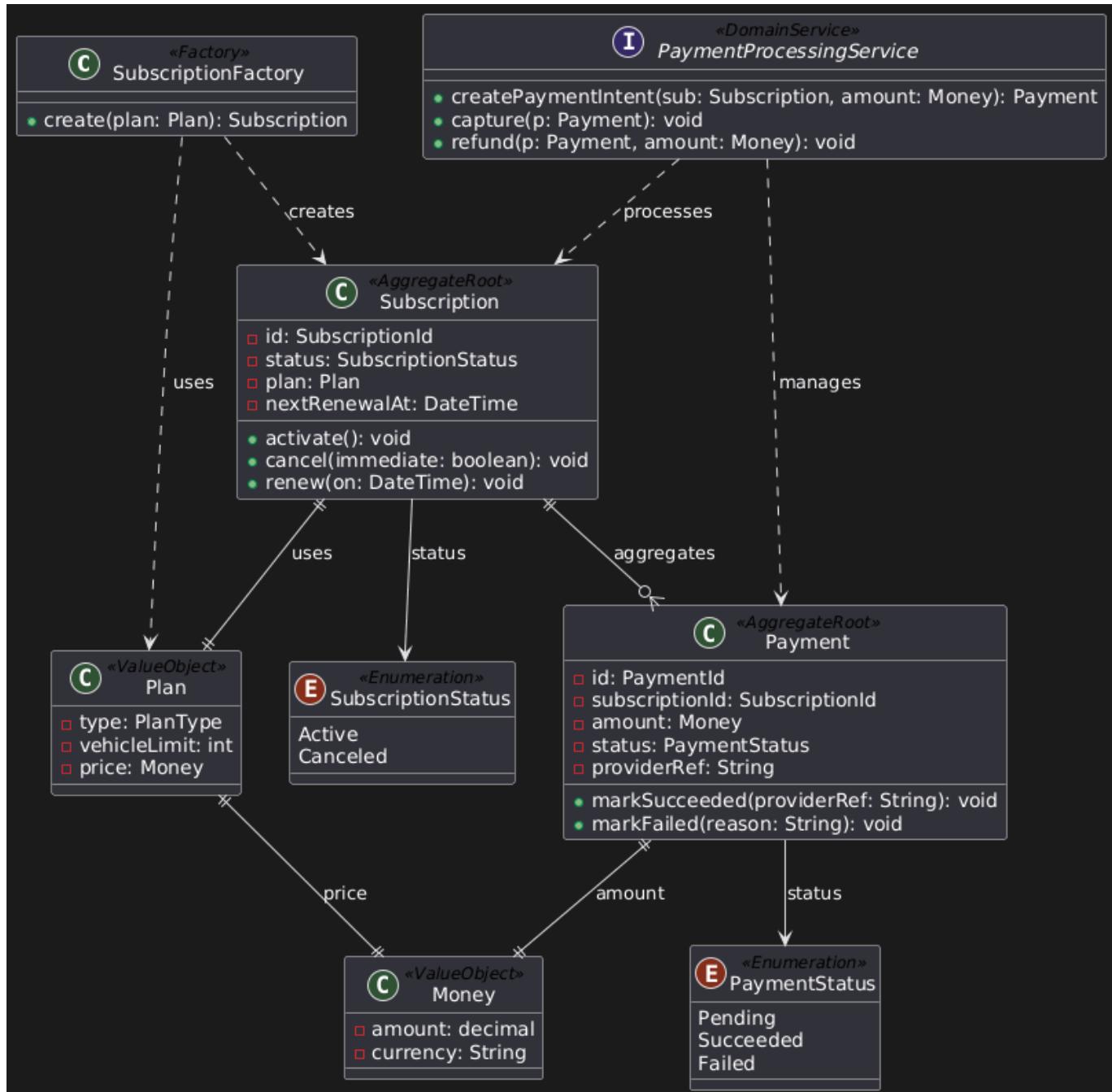


La aplicación móvil de **Subscriptions & Billing** es muy parecido a la versión web, ya que también se conecta al backend por la *Subscriptions API* y la *Query API*. La diferencia es que en el móvil contamos con una base de datos local (SQLite), que nos permite trabajar en modo offline: la app guarda datos y puede seguir operando aunque no haya conexión, y luego sincroniza cuando vuelve el internet.

La app se organiza en pantallas de suscripciones y facturación, un estado de aplicación que maneja la sesión y el cache, y un API Client que envía las solicitudes al backend siempre agregando el token de autenticación. Toda la lógica de negocio sigue estando en el backend; en el cliente solo mostramos información y enviamos las acciones que hace el usuario.

#### 4.2.2.6. Bounded Context Software Architecture Code Level Diagrams

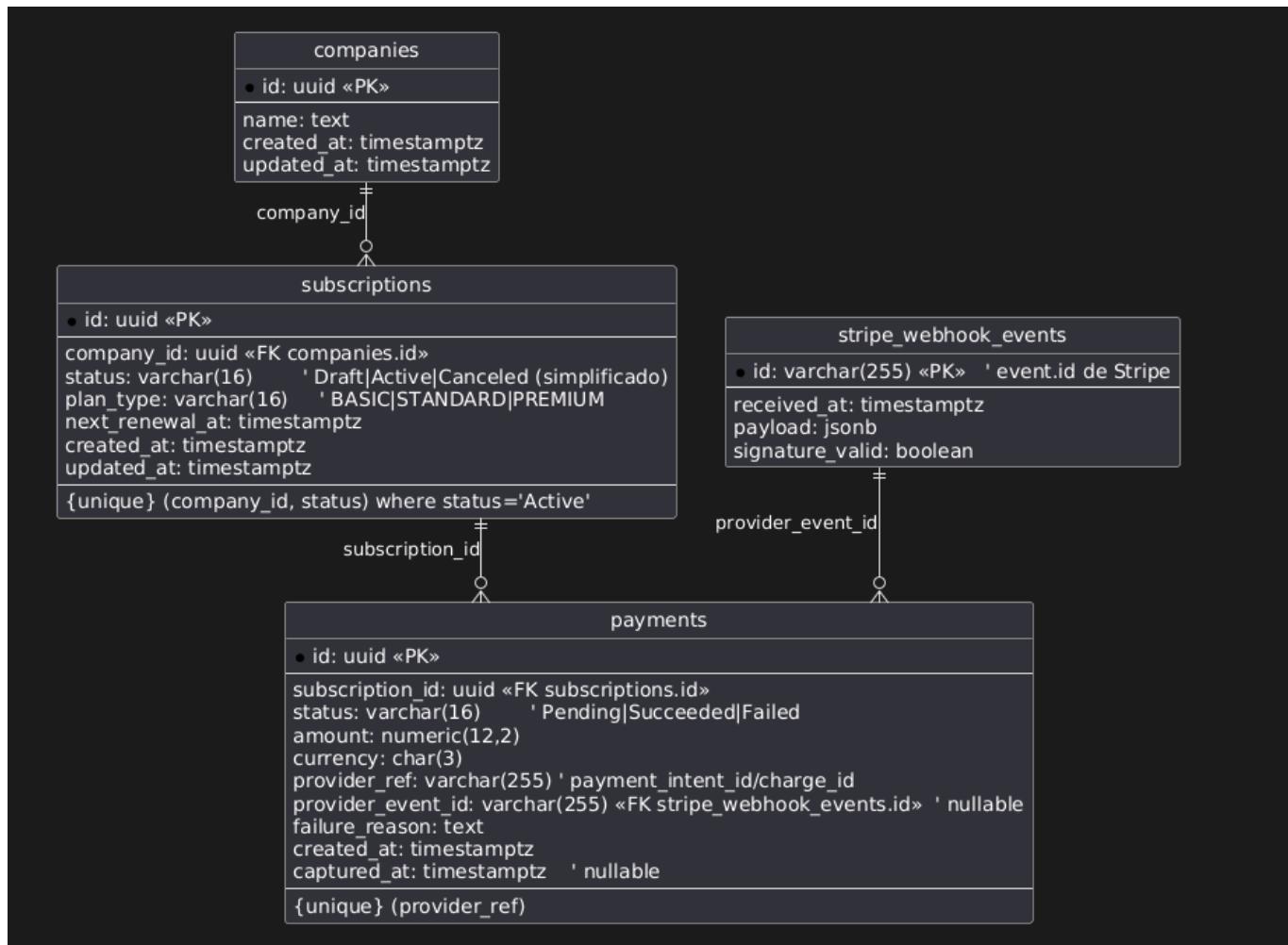
##### 4.2.2.6.1. Bounded Context Domain Layer Class Diagrams



### Explicación del diagrama

El diagrama de clases del Domain Layer muestra a **Subscription** como Aggregate Root, cuyo ciclo de vida se gestiona a través de estados definidos en **SubscriptionStatus** (`Active` y `Canceled`) y su relación con múltiples **Payment**, cada uno con su propio **PaymentStatus** (`Pending`, `Succeeded`, `Failed`). Los Value Objects **Plan** y **Money** encapsulan reglas de negocio como límites de vehículos y montos monetarios. El modelo incluye la **SubscriptionFactory** para la creación controlada de agregados, los Repositories para la persistencia de entidades y el **PaymentProcessingService** como servicio de dominio para la gestión de pagos. En conjunto, este diseño asegura encapsulamiento, claridad en las reglas del negocio e independencia tecnológica en el dominio.

#### 4.2.2.6.2. Bounded Context Database Design Diagram



### Explicación del diagrama

Define la persistencia mínima y suficiente para gestionar compañías, suscripciones y pagos integrados con Stripe. La tabla *companies* centraliza la información de cada cliente. Sobre ella, la tabla *subscriptions* modela el ciclo de vida de la suscripción, incluyendo plan, estado y próxima renovación, con la restricción de que solo puede existir una suscripción activa por compañía. La tabla *payments* registra cada intento de cobro asociado a una suscripción, asegurando unicidad mediante el identificador del proveedor (*provider\_ref*). Finalmente, la tabla **stripe\_webhook\_events** almacena los eventos recibidos desde Stripe y se vincula con los pagos para garantizar trazabilidad e idempotencia en el procesamiento de transacciones.

#### 4.2.3. Bounded Context: *Alerts & Resolution*

##### 4.2.3.1. Domain Layer

##### Entidades (Entities)

##### Entity: Alert (Aggregate Root)

##### Propósito principal

Centralizar la gestión del ciclo de vida de una alerta y garantizar que se cumplan las reglas de negocio.

##### Atributos principales

- alertId: Identificador único de la alerta.
- type: Tipo de alerta (OutOfRange, Offline, RouteDeviation).
- status: Estado actual de la alerta (OPEN, ACKNOWLEDGED, CLOSED).

- sensorType: Tipo de sensor que la generó (TEMPERATURE, HUMIDITY, VIBRATION, TILT, LOCATION, BATTERY).
- createdAt: Fecha y hora de creación de la alerta.
- acknowledgedAt: Momento en que fue reconocida.
- closedAt: Momento en que fue cerrada.

#### **Métodos principales**

- acknowledge(): Marca la alerta como reconocida.
- close(): Cierra la alerta si ya fue reconocida.
- escalate(): Incrementa la criticidad si no fue atendida a tiempo.

### **Entity: Notification**

#### **Propósito principal**

Representar un mensaje enviado a un usuario sobre una alerta.

#### **Atributos principales**

- notificationId: Identificador único de la notificación.
- alertId: Referencia a la alerta asociada.
- channel: Canal de comunicación (EMAIL, SMS, FCM).
- message: Contenido del mensaje.
- sentAt: Fecha y hora de envío.

#### **Métodos principales**

- markAsSent(): Actualiza el estado de la notificación como enviada.

### **Entity: Incident**

#### **Propósito principal**

Registrar un evento relacionado con un viaje que se crea a partir de una alerta.

#### **Atributos principales**

- incidentId: Identificador único del incidente.
- alertId: Referencia a la alerta origen.
- tripId: Identificador del viaje asociado.
- description: Detalle del incidente.
- createdAt: Fecha y hora de creación.

#### **Métodos principales**

- resolve(description): Marca el incidente como resuelto con detalles.

### **Objetos de Valor (Value Objects)**

- AlertType: clasifica los tipos de alertas (OutOfRange, Offline, RouteDeviation).
- AlertStatus: define en qué etapa se encuentra la alerta (Open, Acknowledged, Closed).
- NotificationChannel: indica el medio de comunicación usado (Email, SMS, FCM).
- PersistenceWindow: define el tiempo mínimo que debe cumplirse para que un evento se considere válido como alerta.
- SensorType: clasifica la fuente de monitoreo (TEMPERATURE, HUMIDITY, VIBRATION, TILT, LOCATION, BATTERY).

### **Commands**

**Command: CreateAlertCommand****Parámetros**

- type, sensorType, createdAt.

**Cómo funciona**

Se ejecuta al detectar un evento anómalo. Crea una nueva alerta validando reglas como la ventana de persistencia y evitando duplicación.

**Command: AcknowledgeAlertCommand****Parámetros**

- alertId.

**Cómo funciona**

Permite a un operador reconocer la alerta. Cambia su estado a *ACKNOWLEDGED* y registra la hora.

**Command: CloseAlertCommand****Parámetros**

- alertId.

**Cómo funciona**

Cierra una alerta reconocida, cambiando su estado a *CLOSED* y registrando la fecha de cierre.

**Command: EscalateAlertCommand****Parámetros**

- alertId.

**Cómo funciona**

Incrementa la criticidad de una alerta que lleva demasiado tiempo sin ser reconocida, generando un evento de escalamiento.

**Command: CreateIncidentFromAlertCommand****Parámetros**

- alertId, tripId, description.

**Cómo funciona**

Crea un incidente asociado a un viaje a partir de una alerta específica, permitiendo registrar el detalle del evento.

**Command: SendNotificationCommand****Parámetros**

- alertId, channel, message.

**Cómo funciona**

Ordena enviar una notificación al canal definido (Email, SMS, FCM) para informar al usuario o empresa sobre la alerta.

**Queries****Query: GetAlertByIdQuery****Parámetros**

- alertId.

### Cómo funciona

Recupera los detalles de una alerta específica, incluyendo su estado, tipo y fechas clave.

## Query: GetAlertsByStatusQuery

### Parámetros

- status.

### Cómo funciona

Devuelve todas las alertas con un estado determinado (ej. abiertas, reconocidas, cerradas).

## Query: GetAlertsByTypeQuery

### Parámetros

- type.

### Cómo funciona

Recupera todas las alertas de un tipo específico (ej. RouteDeviation).

## Query: GetNotificationsByAlertIdQuery

### Parámetros

- alertId.

### Cómo funciona

Devuelve todas las notificaciones emitidas en relación con una alerta.

## Query: GetIncidentsByAlertIdQuery

### Parámetros

- alertId.

### Cómo funciona

Obtiene todos los incidentes generados a partir de una alerta determinada.

## Events

### Event: AlertCreatedEvent

Se emite cuando una nueva alerta es registrada en el sistema.

### Event: AlertAcknowledgedEvent

Se emite cuando una alerta es reconocida.

### Event: AlertClosedEvent

Se emite cuando una alerta se cierra exitosamente.

### Event: AlertEscalatedEvent

Se emite cuando una alerta aumenta de criticidad por falta de respuesta.

### Event: NotificationSentEvent

Se emite al enviar una notificación a un usuario o empresa.

### Event: IncidentCreatedEvent

Se emite cuando se genera un incidente a partir de una alerta.

## Fábricas (Factories)

- AlertFactory: encapsula la lógica de creación de una alerta a partir de eventos recibidos (ejemplo: sensor fuera de rango).
- IncidentFactory: crea incidentes asociados a un viaje cuando una alerta lo requiere.

### 4.2.3.2. Interface Layer

En esta capa se definen **Controllers (REST)**.

#### Controllers (REST — Spring Web)

##### AlertController

Este controlador permite crear nuevas alertas a partir de eventos detectados, reconocer (ACK) alertas activas, cerrarlas una vez reconocidas, y obtener tanto el detalle de una alerta específica como la lista de alertas activas (estados OPEN o ACKNOWLEDGED).

##### NotificationController

Su responsabilidad es consultar y actualizar las preferencias de notificación de los usuarios, por ejemplo, los canales permitidos (EMAIL, SMS o FCM) y los tiempos de escalamiento configurados.

##### IncidentController

Permite crear incidentes vinculados a una alerta y un viaje, y consultar el detalle de incidentes registrados.

### 4.2.3.3. Application Layer

#### Command Services

- AlertCommandService: Ejecuta todos los comandos de las alertas.

#### Event Services

- OutOfRangeDetectedEvent: maneja eventos de sensores fuera de rango.
- DeviceOfflineDetectedEvent: maneja eventos de desconexión de dispositivos.
- RouteDeviationDetectedEvent: maneja desvíos de ruta.
- AlertAcknowledgedEvent: actúa tras el reconocimiento de una alerta (ejemplo: detener escalamiento).
- AlertClosedEvent: actúa tras el cierre de una alerta (ejemplo: notificar a analíticas).
- TemperatureOutOfRangeEvent: crea alerta de temperatura.
- HumidityOutOfRangeEvent: crea alerta de humedad.
- VibrationDetectedEvent: maneja vibración anómala.
- TiltOrDumpDetectedEvent: maneja vuelcos o inclinaciones.
- LowBatteryDetectedEvent: maneja alerta de energía.

#### Query Services

- AlertQueryService: Consulta las alertas.

#### Outbound Services

- NotificationService: Servicio para el envío de notificaciones a través de diferentes canales (Email, SMS, Push). Su implementación concreta delega en proveedores externos como Firebase Cloud Messaging

(FCM).

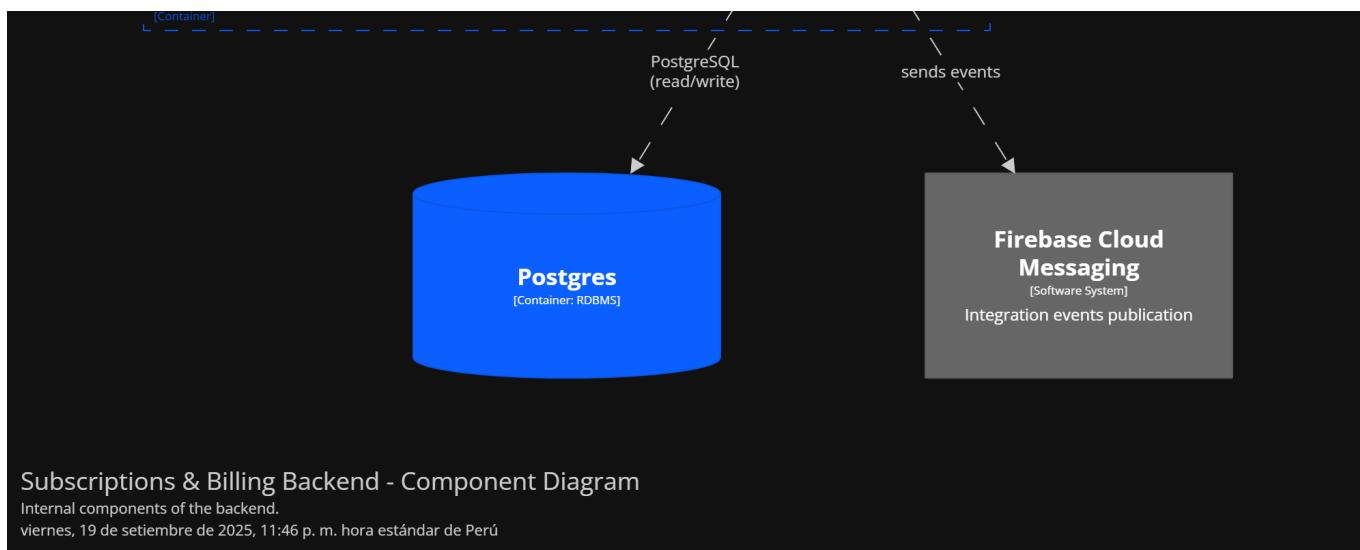
#### 4.2.3.4. Infrastructure Layer

- Notification Repository: Repositorio para acceder a las notificaciones.
- Alert Repository: Repositorio para acceder a las alertas.
- Incident Repository: Repositorio para acceder a los incidentes.

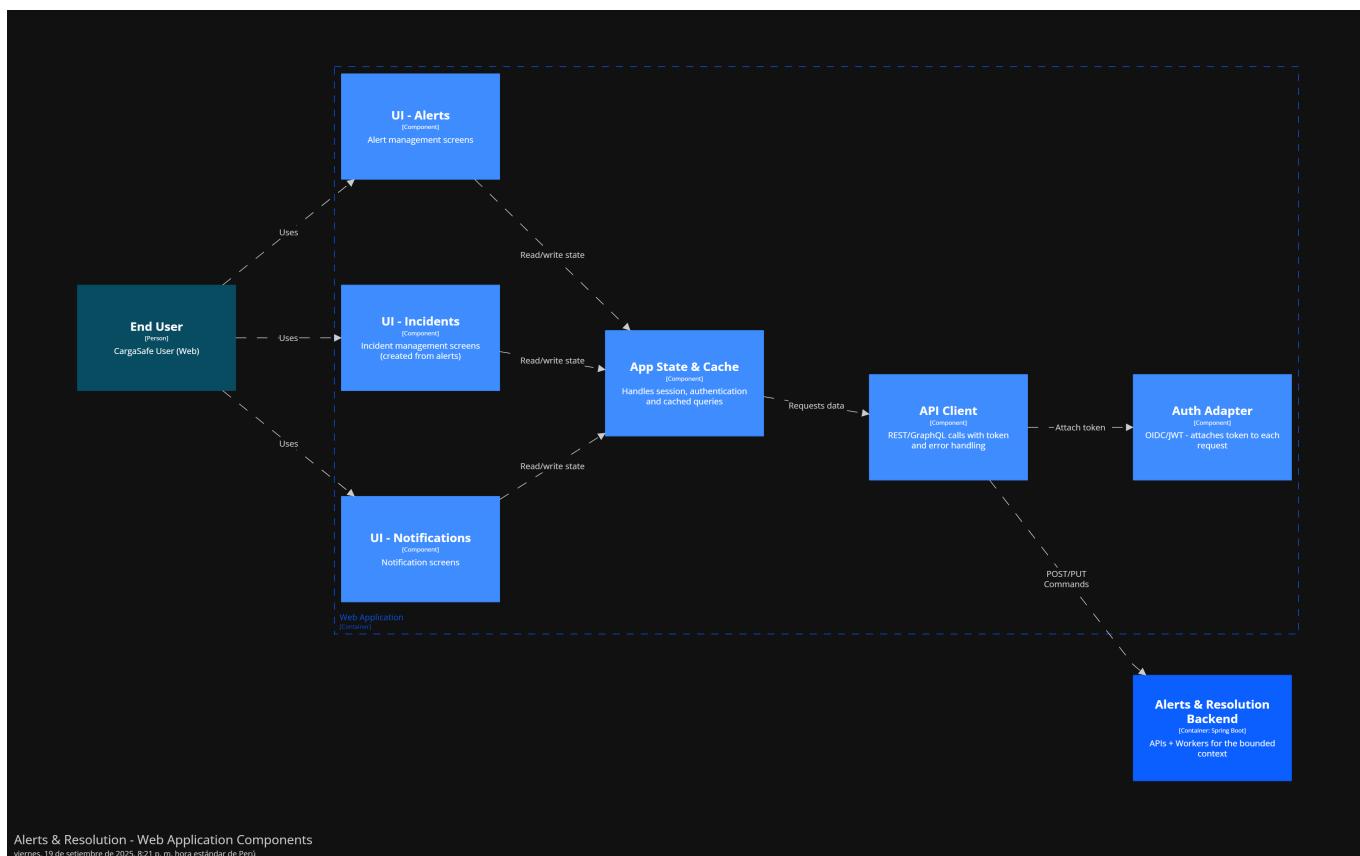
#### 4.2.3.5. Bounded Context Software Architecture Component Level Diagrams

Diagrama de componentes - Backend - Alerts & Resolution

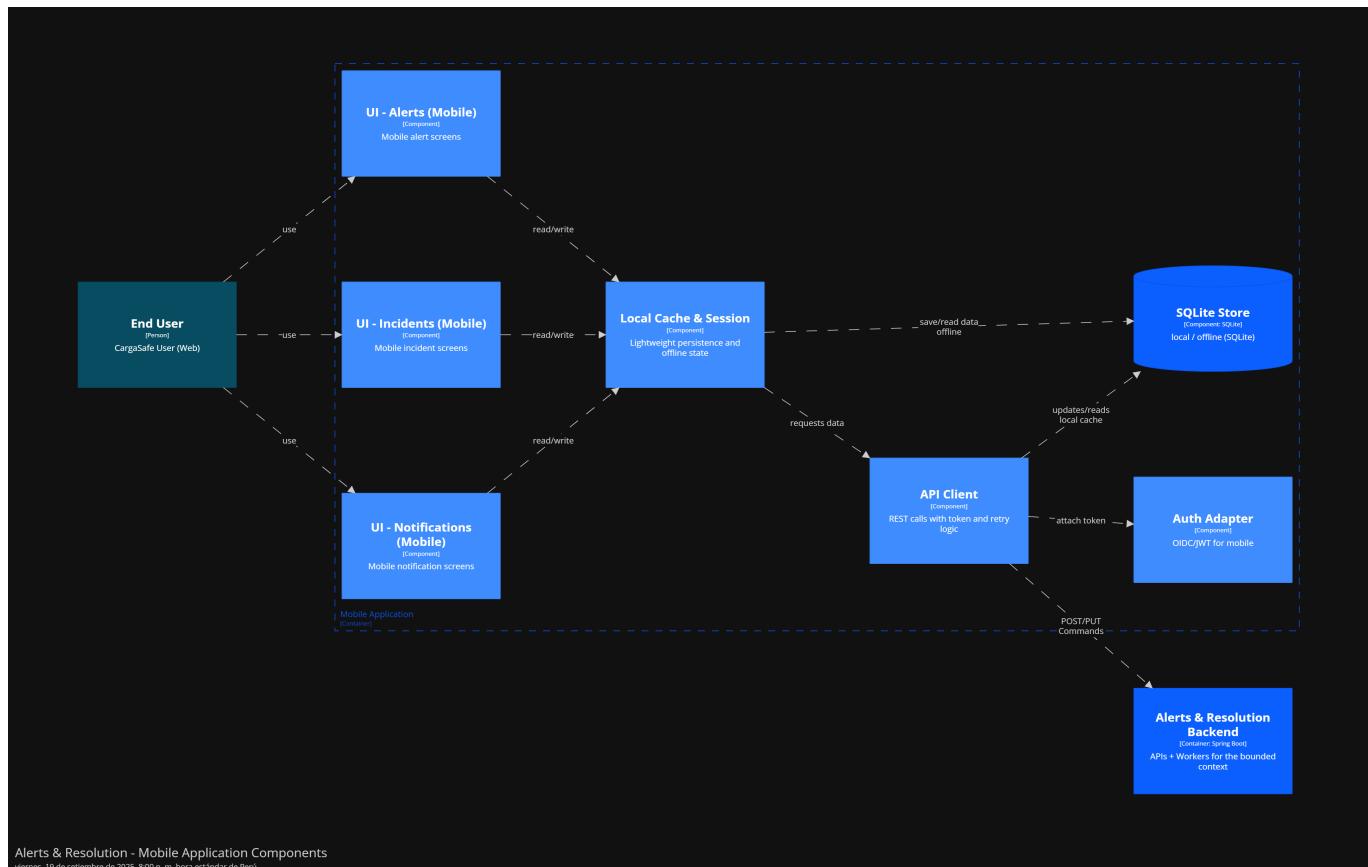




## Diagrama de componentes - Application Web - Alerts & Resolution



## Diagrama de componentes - Mobile App - Alerts & Resolution



#### 4.2.3.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.3.6.1. Bounded Context Domain Layer Class Diagrams

###### Domain Layer Class Diagram

###### Welcome to PlantUML!

You can start with a simple UML Diagram like:

Bob->Alice: Hello

Or

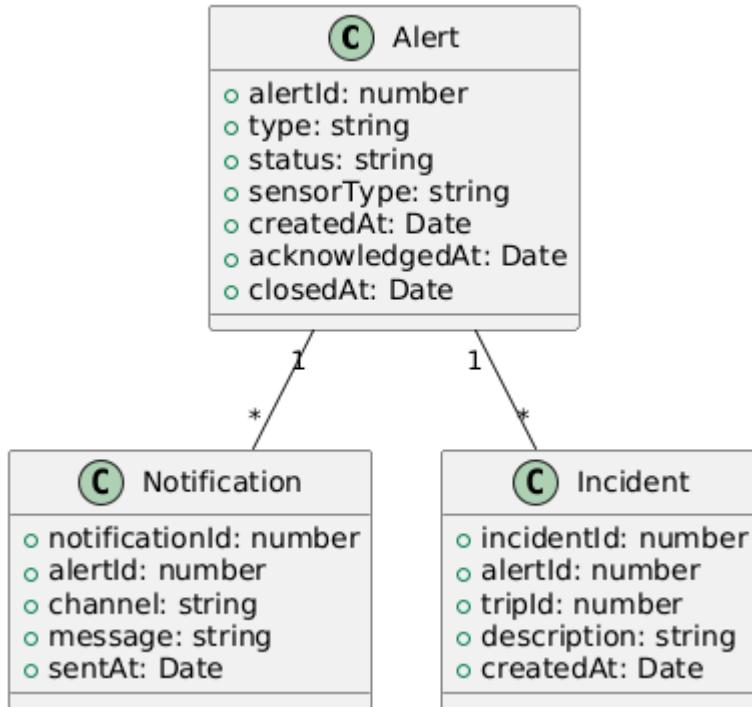
class Example

You will find more information about PlantUML syntax on <https://plantuml.com>

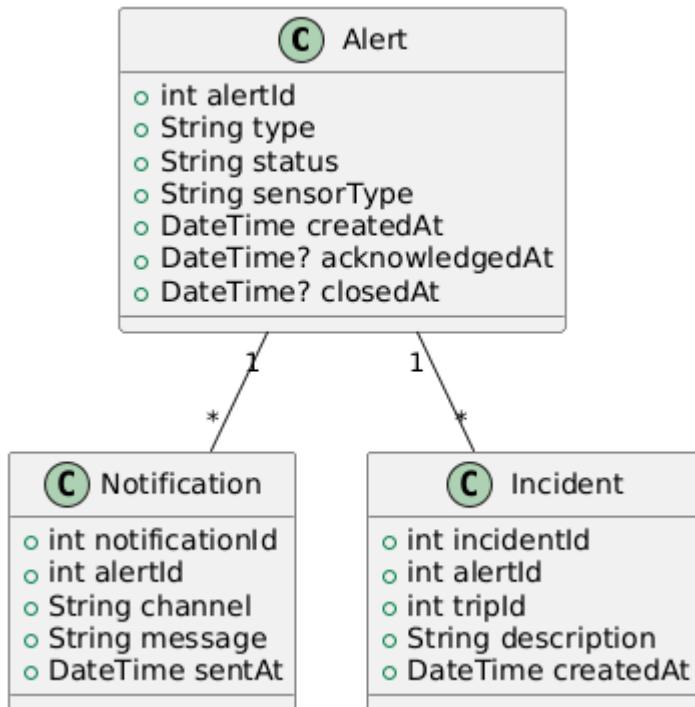
(Details by typing license keyword)



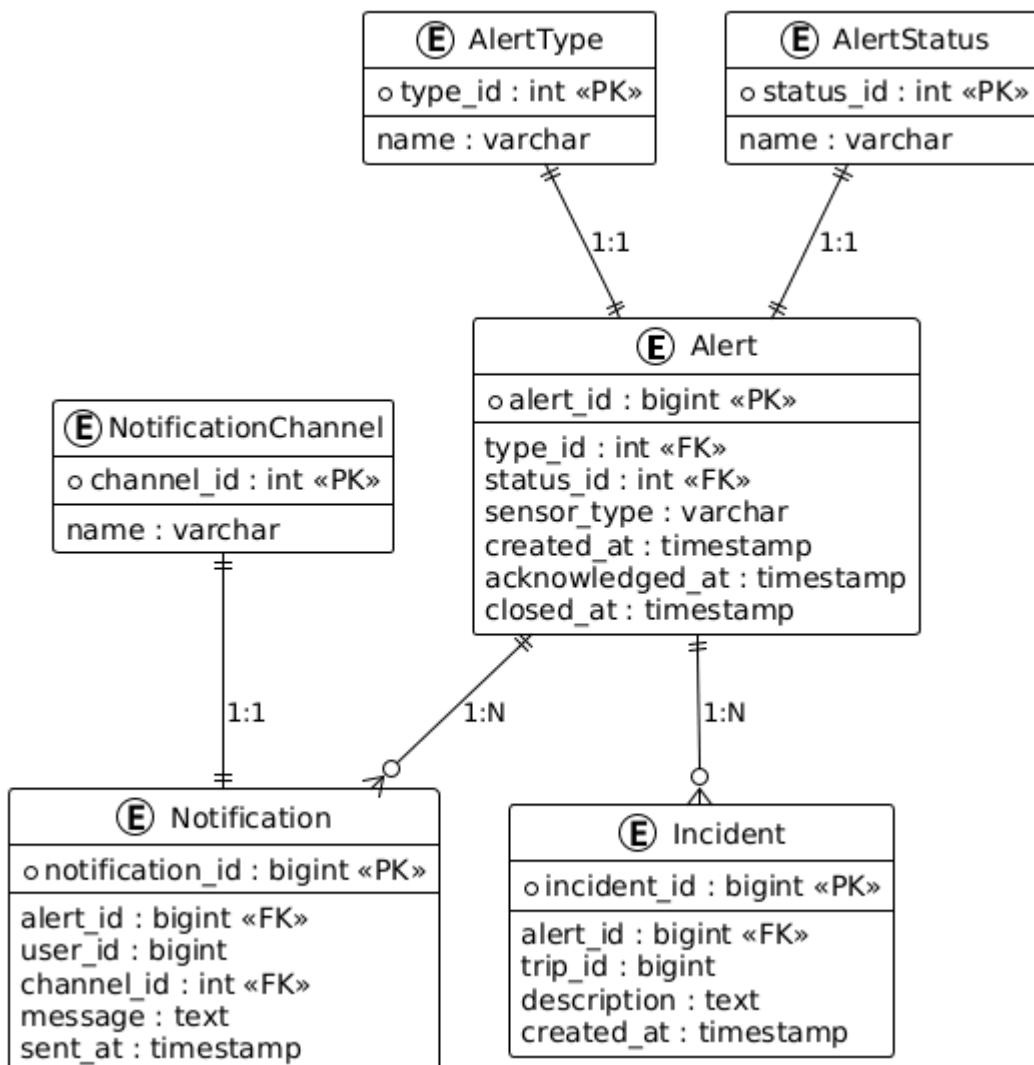
###### WebApp Class Diagram



### Mobile App Class Diagram



#### 4.2.3.6.2. Bounded Context Database Design Diagram



#### 4.2.4. Bounded Context: *Real-Time Monitoring*

##### 4.2.4.1. Domain Layer.

##### **Entity: MonitoringSession (Aggregate Root)**

##### **Propósito principal**

Representar una sesión de monitoreo para un viaje específico y centralizar su ciclo de vida, asegurando que se cumplan las reglas de negocio relacionadas con la captura y validación de datos de telemetría.

##### **Atributos principales**

- sessionId: Identificador único de la sesión de monitoreo.
- tripId: Identificador del viaje asociado.
- status: Estado de la sesión (ACTIVE, INACTIVE, COMPLETED, PAUSED).
- temperatureRange: Rango de temperatura aceptable para el viaje.
- startTime: Fecha y hora de inicio de la sesión.
- endTime: Fecha y hora de finalización de la sesión.

##### **Métodos principales**

- startSession(): Inicia la sesión de monitoreo y cambia su estado a "ACTIVE".

- `endSession()`: Finaliza la sesión de monitoreo y cambia su estado a "COMPLETED".
  - `pauseSession()`: Pausa temporalmente la sesión de monitoreo.
  - `resumeSession()`: Reanuda una sesión pausada.
  - `addTelemetryData(data)`: Agrega una nueva lectura de telemetría a la sesión.
  - `isActive()`: Verifica si la sesión está activa.
- 

## Entity: TelemetryData

### Propósito principal

Representar una única lectura de sensor con identidad propia, capaz de almacenar y gestionar información específica de temperatura, humedad, vibración y ubicación.

### Atributos principales

- `dataId`: Identificador único de la lectura de telemetría.
- `sessionId`: Identificador de la sesión de monitoreo asociada.
- `sensorReading`: Datos específicos de la lectura del sensor.
- `location`: Coordenadas geográficas de la lectura.
- `timestamp`: Fecha y hora de la lectura.
- `deviceID`: Identificador del dispositivo IoT.

### Métodos principales

- `isWithinRange(range)`: Verifica si la lectura está dentro del rango aceptable.
  - `getTemperature()`: Obtiene el valor de temperatura de la lectura.
  - `getHumidity()`: Obtiene el valor de humedad de la lectura.
  - `getVibration()`: Obtiene el valor de vibración de la lectura.
- 

## Entity: Incident

### Propósito principal

Representar un incidente detectado durante el monitoreo como una entidad con identidad propia.

### Atributos principales

- `incidentId`: Identificador único del incidente.
- `sessionId`: Identificador de la sesión de monitoreo asociada.
- `alertType`: Tipo de alerta que generó el incidente.
- `description`: Descripción detallada del incidente.
- `createdAt`: Fecha y hora de creación del incidente.
- `status`: Estado del incidente (OPEN, IN\_PROGRESS, RESOLVED, ESCALATED).

### Métodos principales

- `resolve(description)`: Marca el incidente como resuelto.
  - `escalate()`: Escala el incidente a un nivel superior.
  - `updateStatus(newStatus)`: Actualiza el estado del incidente.
- 

## Value Object: SensorReading

**Propósito principal**

Encapsular los datos de una lectura específica de sensores de forma inmutable.

**Atributos principales**

- temperature: Valor de temperatura medido.
  - humidity: Valor de humedad medido.
  - vibration: Valor de vibración medido.
  - signalStatus: Estado de la señal del dispositivo.
- 

**Value Object: TemperatureRange****Propósito principal**

Definir los límites mínimos y máximos de temperatura aceptables para un viaje.

**Atributos principales**

- minTemperature: Temperatura mínima aceptable.
- maxTemperature: Temperatura máxima aceptable.

**Métodos principales**

- isWithinRange(temperature): Verifica si una temperatura está dentro del rango.
- 

**Value Object: Location****Propósito principal**

Representar un punto geográfico inmutable con coordenadas precisas.

**Atributos principales**

- latitude: Latitud de la ubicación.
- longitude: Longitud de la ubicación.

**Métodos principales**

- distanceTo(other): Calcula la distancia a otra ubicación.
- 

**Value Object: SessionStatus****Propósito principal**

Representar el estado de una sesión de monitoreo en su ciclo de vida.

**Atributos principales**

- status: Valor posible (ACTIVE, INACTIVE, COMPLETED, PAUSED).
- 

**Value Object: SignalStatus**

**Propósito principal**

Indicar el estado de conectividad del dispositivo IoT.

**Atributos principales**

- status: Valor posible (ONLINE, OFFLINE, WEAK\_SIGNAL).
- 

**Value Object: AlertType****Propósito principal**

Clasificar los diferentes tipos de alertas que pueden generarse durante el monitoreo.

**Atributos principales**

- type: Valor posible (TEMPERATURE\_HIGH, TEMPERATURE\_LOW, DEVICE\_OFFLINE, ROUTE\_DEVIATION, VIBRATION\_EXCESSIVE).
- 

**Value Object: Temperature****Propósito principal**

Expresar una magnitud de temperatura con su unidad correspondiente.

**Atributos principales**

- value: Valor numérico de la temperatura.
  - unit: Unidad de medida (ej. °C, °F).
- 

**Value Object: Humidity****Propósito principal**

Expresar un porcentaje de humedad relativa.

**Atributos principales**

- value: Valor numérico de humedad.
  - unit: Unidad de medida (ej. %).
- 

**Value Object: Vibration****Propósito principal**

Expresar una magnitud de vibración con su unidad correspondiente.

**Atributos principales**

- value: Valor numérico de vibración.
  - unit: Unidad de medida (ej. g, m/s<sup>2</sup>).
- 

**Aggregate: MonitoringSessionAggregate**

## Propósito principal

Asegurar la consistencia de una sesión de monitoreo como unidad de negocio, agrupando la sesión con sus datos de telemetría relacionados.

## Métodos principales

- validateSessionReady(): Verifica que la sesión tenga todos los parámetros necesarios antes de iniciar.
  - evaluateRules(): Evalúa las reglas de negocio sobre las lecturas recibidas.
- 

## Factory: MonitoringSessionFactory

### Propósito principal

Crear instancias de **MonitoringSession** en estado inicial válido.

## Métodos principales

- createSession(tripId, temperatureRange): Genera una sesión de monitoreo en estado INACTIVE con todos los datos requeridos.
  - createFromTrip(trip): Crea una sesión basada en los datos de un viaje existente.
- 

## Domain Service: DataIngestionService

### Propósito principal

Encapsular la lógica de procesamiento y validación de datos de telemetría entrantes.

## Métodos principales

- processIncomingData(rawData): Procesa datos brutos de telemetría y los convierte en TelemetryData válidos.
  - validateTelemetryData(data): Valida la integridad y consistencia de los datos de telemetría.
- 

## Domain Service: RuleEvaluationService

### Propósito principal

Analizar las lecturas en tiempo real para detectar violaciones de parámetros y generar alertas.

## Métodos principales

- evaluateTemperatureRules(data, range): Evalúa si las lecturas de temperatura están dentro del rango aceptable.
  - evaluateDeviceConnectivity(lastReading): Verifica el estado de conectividad del dispositivo.
  - detectAnomalies(dataList): Detecta patrones anómalos en las lecturas de sensores.
- 

## Domain Service: DataEnrichmentService

### Propósito principal

Enriquecer los datos de telemetría con información contextual adicional.

## Métodos principales

- `enrichWithLocationData(data)`: Enriquece los datos con información geográfica adicional.
  - `enrichWithRouteInformation(data)`: Agrega información de ruta a los datos de telemetría.
- 

## Command: StartMonitoringSessionCommand

### Propósito

Iniciar una nueva sesión de monitoreo para un viaje específico.

### Parámetros

- `tripId`: Identificador único del viaje.
  - `temperatureRange`: Rango de temperatura aceptable para el monitoreo.
- 

## Command: EndMonitoringSessionCommand

### Propósito

Finalizar una sesión de monitoreo activa, cambiando su estado a COMPLETED.

### Parámetros

- `sessionId`: Identificador único de la sesión de monitoreo.
- 

## Command: PauseMonitoringSessionCommand

### Propósito

Pausar temporalmente una sesión de monitoreo activa.

### Parámetros

- `sessionId`: Identificador único de la sesión de monitoreo.
- 

## Command: ResumeMonitoringSessionCommand

### Propósito

Reanudar una sesión de monitoreo previamente pausada.

### Parámetros

- `sessionId`: Identificador único de la sesión de monitoreo.
- 

## Query: GetMonitoringSessionByIdQuery

### Propósito

Obtener la información completa de una sesión de monitoreo específica mediante su identificador único.

### Parámetros

- `sessionId`: Identificador único de la sesión de monitoreo.
-

## Query: GetTelemetryDataBySessionQuery

### Propósito

Obtener todas las lecturas de telemetría asociadas a una sesión específica de monitoreo.

### Parámetros

- sessionId: Identificador único de la sesión de monitoreo.
  - startTime: Fecha y hora de inicio del rango de consulta (opcional).
  - endTime: Fecha y hora de fin del rango de consulta (opcional).
- 

## Query: GetActiveSessionsQuery

### Propósito

Listar todas las sesiones de monitoreo que se encuentran actualmente activas.

### Parámetros

(No requiere parámetros)

---

## Query: GetSessionsByTripIdQuery

### Propósito

Obtener todas las sesiones de monitoreo asociadas a un viaje específico.

### Parámetros

- tripId: Identificador único del viaje.
- 

## Event: MonitoringSessionStartedEvent

### Propósito

Notificar que una nueva sesión de monitoreo ha sido iniciada.

### Parámetros

- sessionId: Identificador único de la sesión de monitoreo.
  - tripId: Identificador del viaje asociado.
  - temperatureRange: Rango de temperatura configurado.
  - startedAt: Fecha y hora de inicio de la sesión.
- 

## Event: MonitoringSessionCompletedEvent

### Propósito

Notificar que una sesión de monitoreo se ha completado satisfactoriamente.

### Parámetros

- sessionId: Identificador único de la sesión de monitoreo.
- totalReadings: Número total de lecturas procesadas.
- completedAt: Fecha y hora de finalización de la sesión.

**Event: TelemetryDataReceivedEvent****Propósito**

Notificar que se ha recibido y procesado una nueva lectura de telemetría.

**Parámetros**

- sessionId: Identificador único de la sesión de monitoreo.
  - telemetryData: Datos de telemetría recibidos.
  - receivedAt: Fecha y hora de recepción.
- 

**Event: OutOfRangeDetectedEvent****Propósito**

Notificar que se ha detectado una lectura fuera del rango aceptable.

**Parámetros**

- sessionId: Identificador único de la sesión de monitoreo.
  - telemetryData: Datos de telemetría que causaron la alerta.
  - violatedParameter: Parámetro que está fuera de rango.
  - detectedAt: Fecha y hora de detección.
- 

**Event: DeviceOfflineDetectedEvent****Propósito**

Notificar que un dispositivo IoT ha dejado de enviar datos de telemetría.

**Parámetros**

- deviceId: Identificador único del dispositivo.
  - sessionId: Identificador de la sesión de monitoreo afectada.
  - lastSeenTime: Última vez que se recibieron datos del dispositivo.
  - detectedAt: Fecha y hora de detección del problema.
- 

**Event: IncidentCreatedEvent****Propósito**

Notificar que se ha creado un nuevo incidente basado en las alertas detectadas.

**Parámetros**

- incidentId: Identificador único del incidente.
- sessionId: Identificador de la sesión de monitoreo asociada.
- alertType: Tipo de alerta que originó el incidente.
- createdAt: Fecha y hora de creación del incidente.

**4.2.4.2. Interface Layer.**

## Controllers

- MonitoringController: Controlador que maneja las solicitudes relacionadas con el monitoreo de viajes. Atiende operaciones como iniciar una nueva sesión de monitoreo, finalizar o pausar una sesión, así como consultar información de sesiones por identificador, estado del viaje, o recuperar datos de telemetría en tiempo real para visualización en mapas y gráficos.
- TelemetryController: Controlador que maneja las solicitudes relacionadas con los datos de telemetría. Permite consultar lecturas específicas, obtener históricos de datos de sensores y generar reportes de monitoreo para análisis posterior.

### 4.2.4.3. Application Layer.

## Command Services

- MonitoringCommandService: Se encarga de recibir y coordinar los comandos relacionados a las sesiones de monitoreo. Dentro de él se manejan distintos handlers, cada uno especializado en ejecutar un comando específico como iniciar, finalizar, pausar o reanudar sesiones de monitoreo.
- TelemetryCommandService: Se encarga de coordinar los comandos relacionados con el procesamiento de datos de telemetría. Administra la ingesta, validación y enriquecimiento de datos para garantizar que las lecturas sean procesadas correctamente.

---

## Query Services

- MonitoringQueryService: Se encarga de atender las consultas relacionadas a las sesiones de monitoreo. Contiene handlers que procesan queries para obtener información, por ejemplo: consultar una sesión por su identificador, listar sesiones activas o recuperar todas las sesiones de un viaje.
- TelemetryQueryService: Atiende las consultas relacionadas a los datos de telemetría. Permite obtener lecturas específicas, históricos de datos de sensores y generar agregaciones para reportes y visualizaciones.

---

## Event Services

- MonitoringEventService: Se encarga de atender los eventos relacionados a las sesiones de monitoreo. Dentro de él se gestionan distintos servicios especializados que reaccionan a cada evento, como inicio y finalización de sesiones, recepción de datos de telemetría, detección de alertas y creación de incidentes, ejecutando las acciones necesarias después de que ocurren.

### 4.2.4.4. Infrastructure Layer.

## Repositories

- IMonitoringSessionRepository: Repositorio que define las operaciones de acceso a las sesiones de monitoreo, como guardar, actualizar y recuperar información de una sesión.
- ITelemetryDataRepository: Repositorio que define las operaciones de acceso a los datos de telemetría, optimizado para escrituras masivas y consultas de series de tiempo, como registrar nuevas lecturas, consultarlas por sesión y generar agregaciones.

#### 4.2.4.5. Bounded Context Software Architecture Component Level Diagrams

Diagrama de componentes - Backend - Real-Time Monitoring

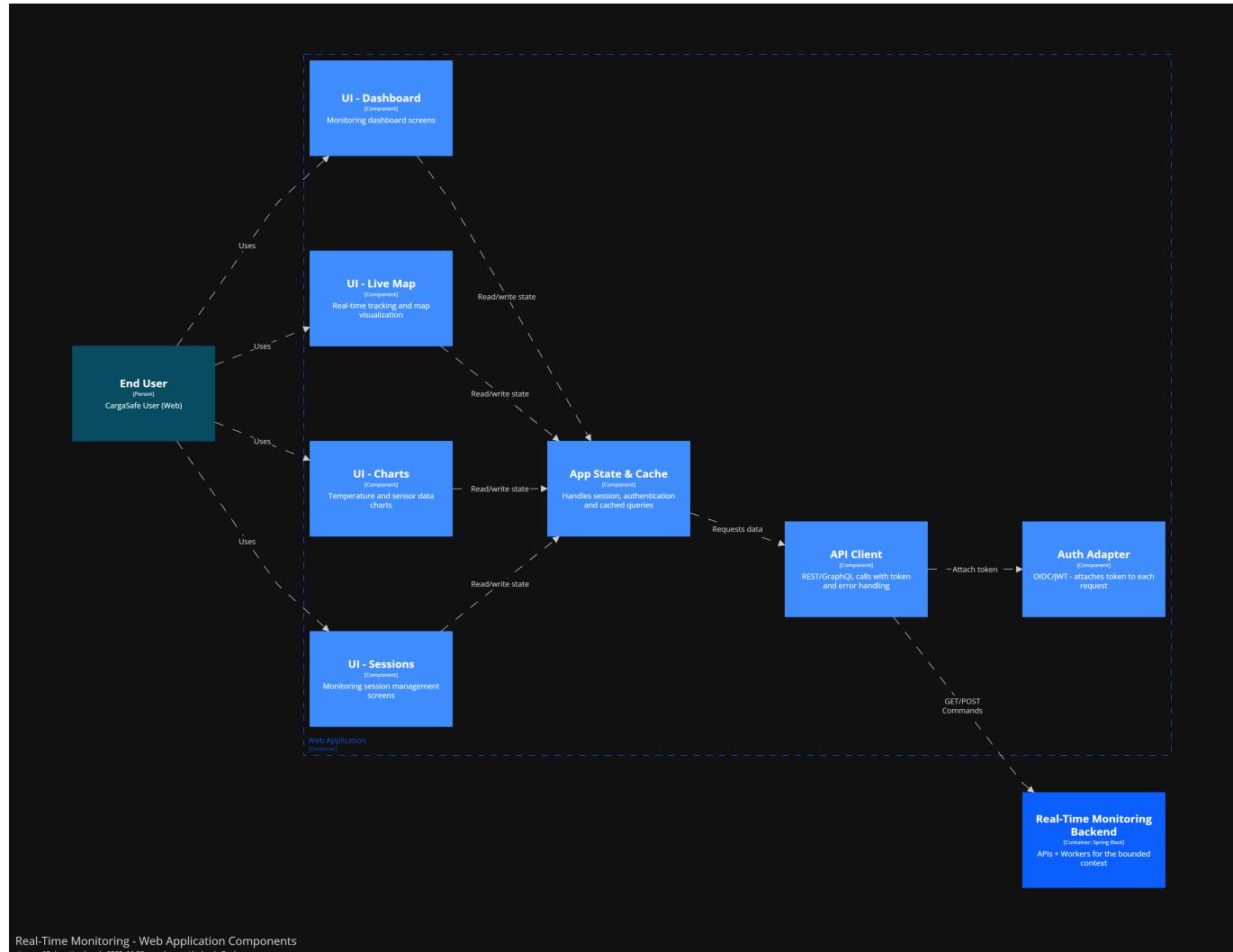


## Real-Time Monitoring Context - Component Diagram

Internal layered components of the Real-Time Monitoring Context inside CargaSafe.

viernes, 19 de setiembre de 2025, 11:43 p. m. hora estándar de Perú

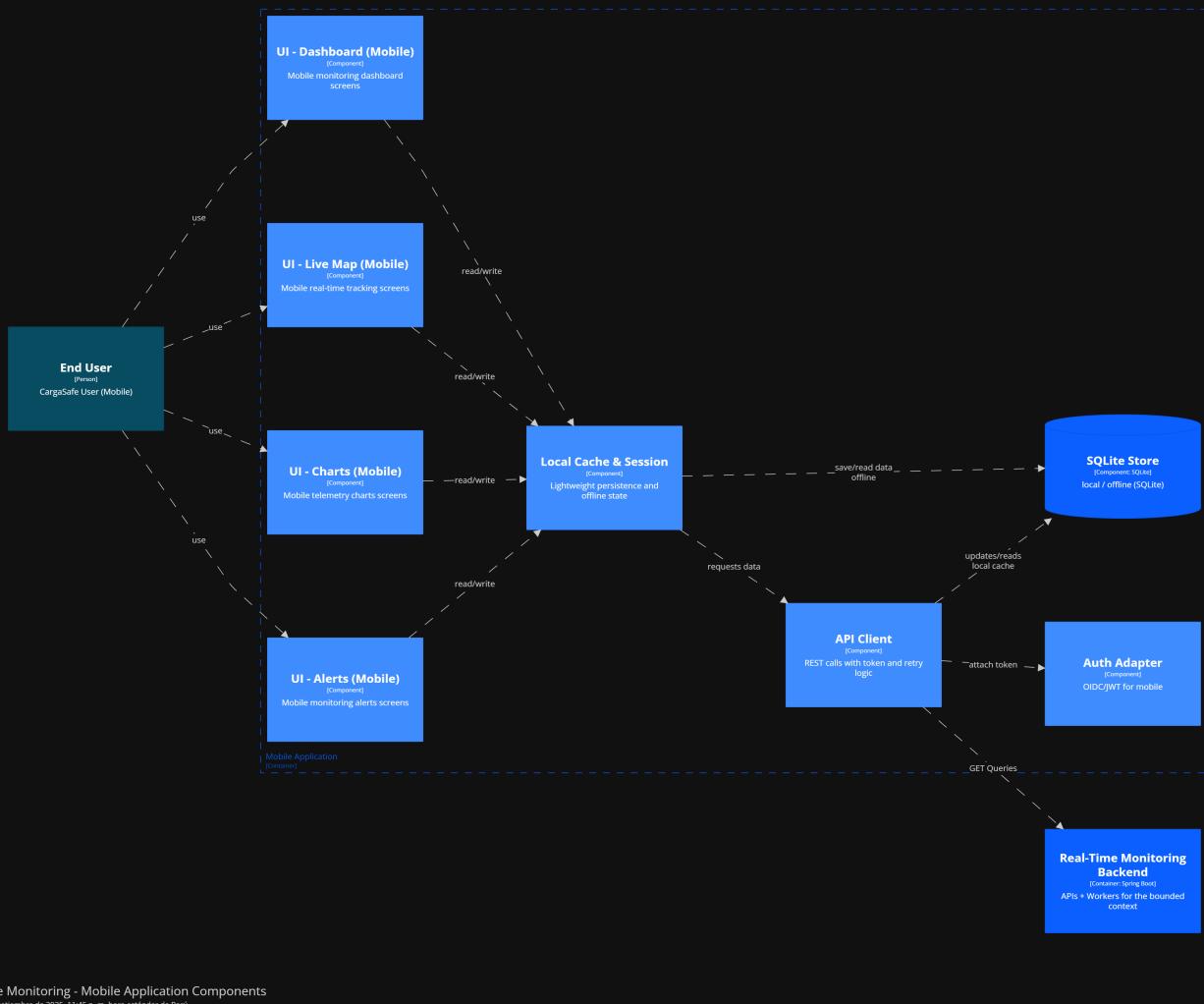
Diagrama de componentes - Application Web - Real-Time Monitoring



Real-Time Monitoring - Web Application Components

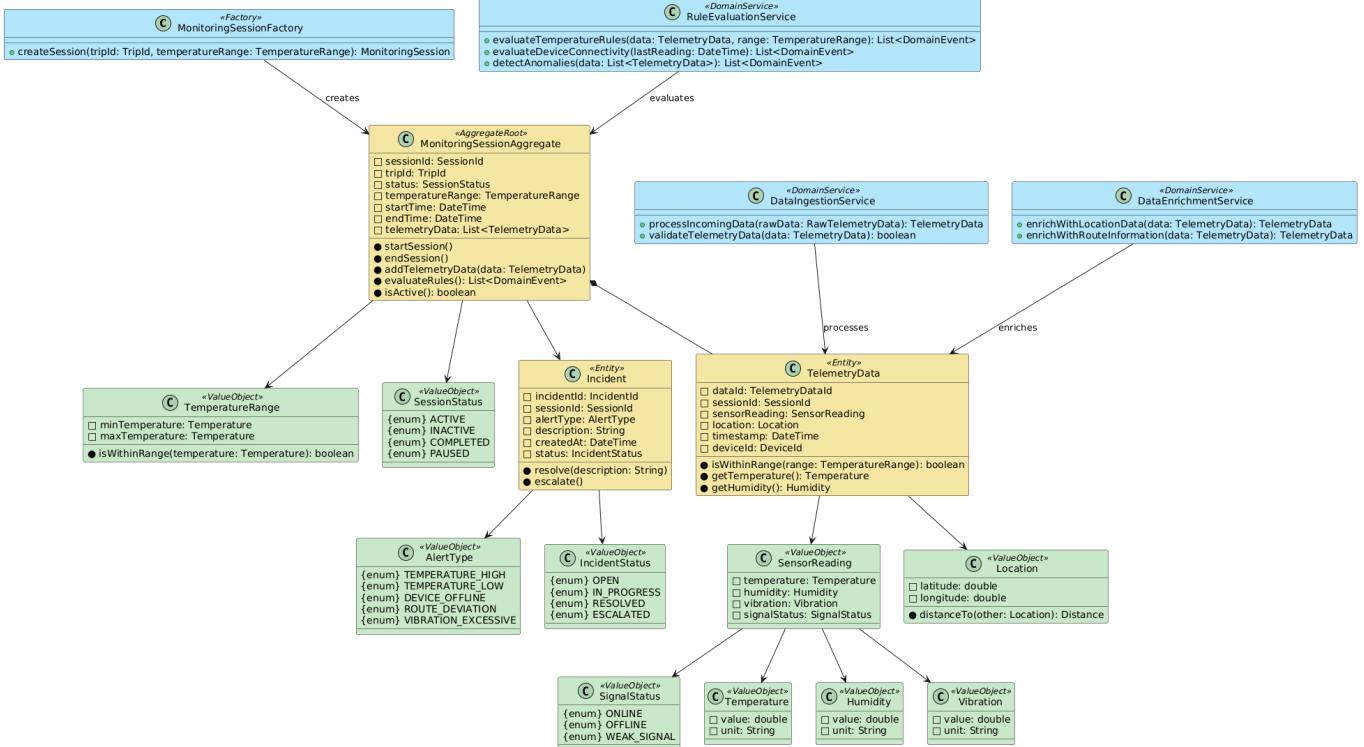
viernes, 19 de setiembre de 2025, 11:53 p. m. hora estándar de Perú

Diagrama de componentes - Mobile App - Real-Time Monitoring

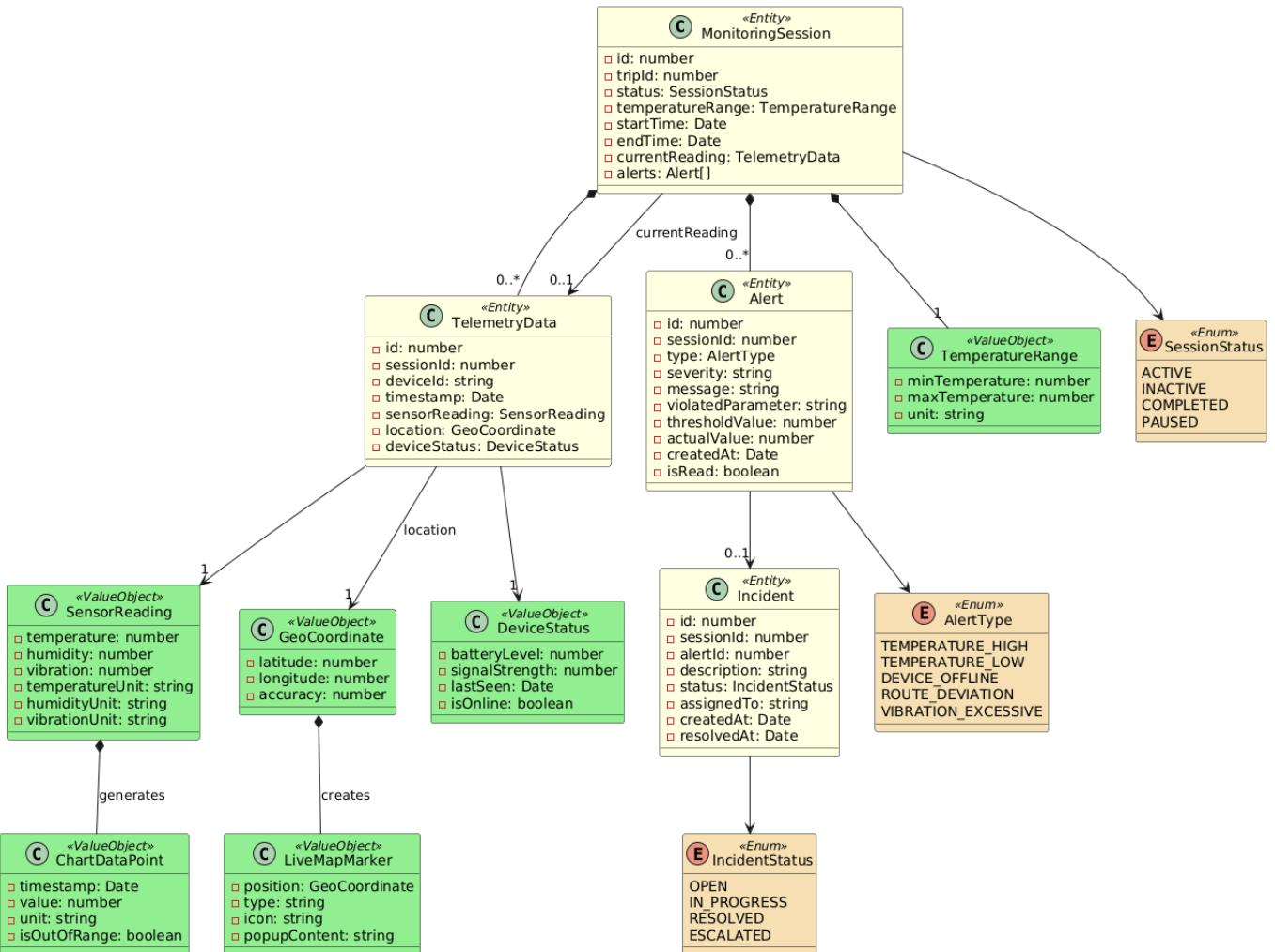


#### 4.2.4.6. Bounded Context Software Architecture Code Level Diagrams

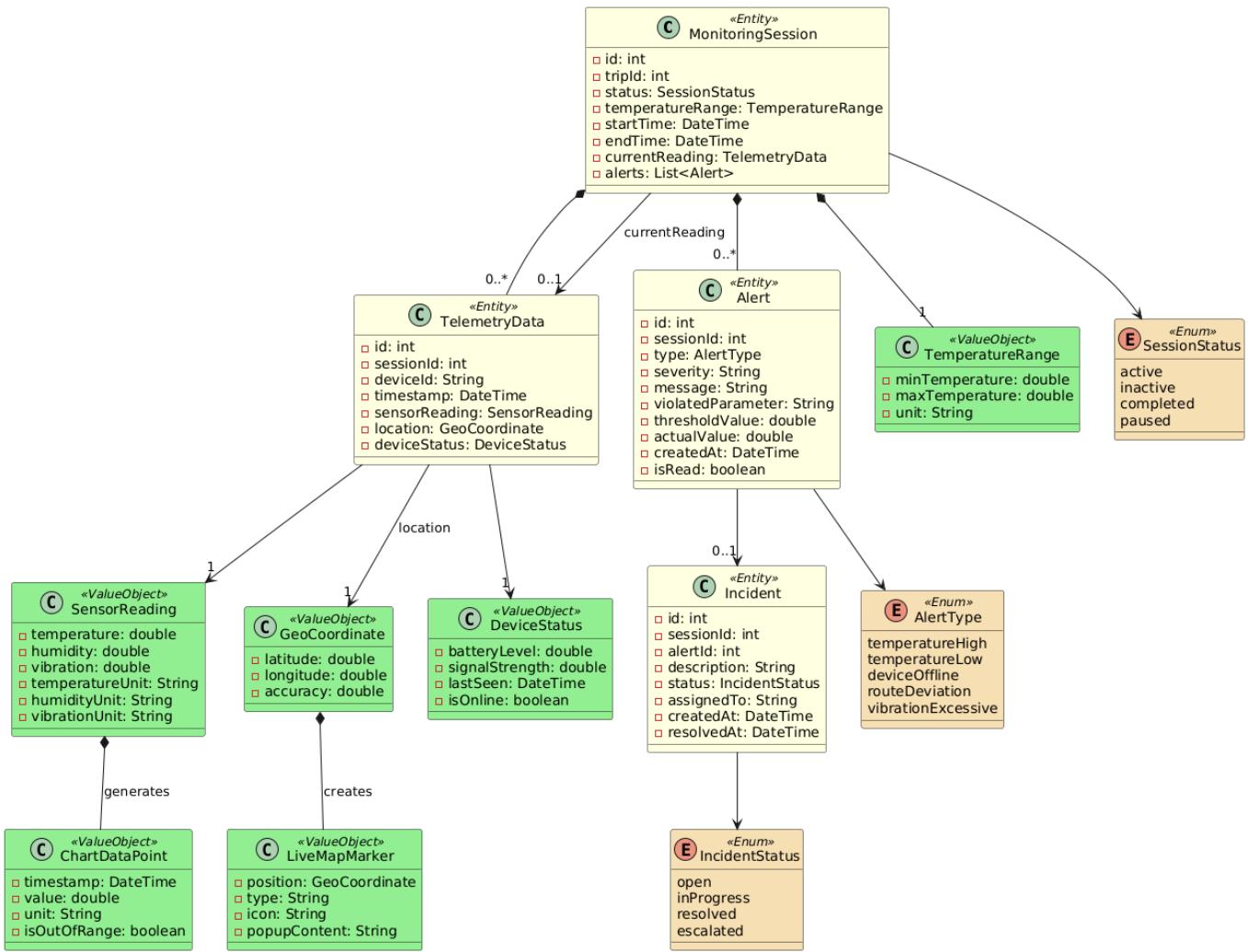
##### 4.2.4.6.1. Bounded Context Domain Layer Class Diagrams



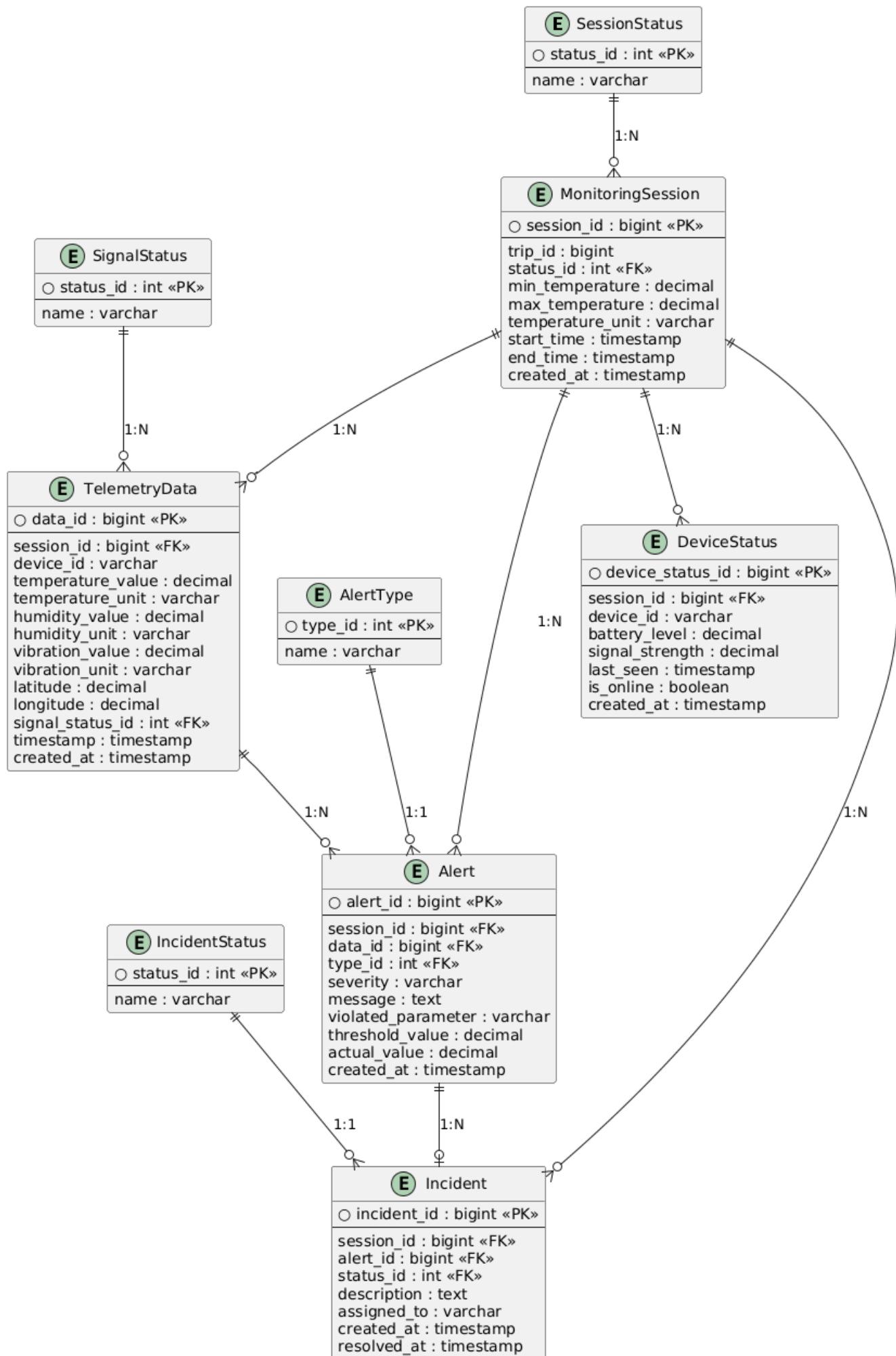
## WebApp Class Diagram



## Mobile Class Diagram



#### 4.2.4.6.2. Bounded Context Database Design Diagram



## 4.2.5. Bounded Context: *Trip management*

### 4.2.5.1. Domain Layer.

#### Entity: Trip (Aggregate Root)

##### Propósito principal

Representar un viaje y centralizar su ciclo de vida, asegurando que se cumplan las reglas de negocio relacionadas con cliente, conductor, vehículo y ruta.

##### Atributos principales

- tripId: Identificador único del viaje.
- clientId: Identificador del cliente.
- driverId: Identificador del conductor.
- vehicleId: Identificador del vehículo.
- route: Ruta definida para el trayecto.
- status: Estado del viaje (CREATED, IN\_PROGRESS, COMPLETED, CANCELLED).
- requestedAt: Fecha y hora de la solicitud.

##### Métodos principales

- assignDriver(driverId): Asigna un conductor al viaje.
- assignVehicle(vehicleId): Vincula un vehículo al viaje.
- startTrip(): Inicia el viaje y cambia su estado a "En curso".
- completeTrip(): Finaliza el viaje y cambia su estado a "Completado".
- cancelTrip(reason): Cancela el viaje y registra el motivo.

---

#### Entity: Route

##### Propósito principal

Representar la ruta de un viaje como una entidad con identidad propia, capaz de almacenar y gestionar la información de los tramos, distancias y duración total.

##### Atributos principales

- routId: Identificador único de la ruta.
- origin: Punto de inicio.
- destination: Punto final.
- segments: Lista de tramos de la ruta.
- totalDistance: Distancia total del viaje.
- totalDuration: Duración total estimada.

##### Métodos principales

- addSegment(segment): Agrega un tramo adicional a la ruta.
- updateDestination(newDestination): Cambia el destino de la ruta antes de iniciar el viaje.
- recalculateTotals(): Recalcula la distancia y la duración total a partir de los segmentos actuales.

## **Value Object: GeoCoordinate**

### **Propósito principal**

Representar un punto geográfico inmutable.

### **Atributos principales**

- latitude: Latitud válida.
  - longitude: Longitud válida.
- 

## **Value Object: RouteSegment**

### **Propósito principal**

Modelar un tramo de ruta entre dos puntos.

### **Atributos principales**

- coordinates: Lista de coordenadas que forman el tramo.
  - distance: Distancia recorrida en el segmento.
  - duration: Tiempo estimado del segmento.
- 

## **Value Object: Distance**

### **Propósito principal**

Expresar una magnitud de distancia.

### **Atributos principales**

- value: Cantidad numérica de la distancia.
  - unit: Unidad de medida (ej. km).
- 

## **Value Object: Duration**

### **Propósito principal**

Expresar un intervalo de tiempo.

### **Atributos principales**

- value: Cantidad numérica de tiempo.
  - unit: Unidad de medida (ej. minutos).
- 

## **Value Object: TripStatus**

### **Propósito principal**

Representar el estado del viaje en su ciclo de vida.

### **Atributos principales**

- status: Valor posible (PENDING, IN\_PROGRESS, COMPLETED, CANCELLED).
-

## Aggregate: TripAggregate

### Propósito principal

Asegurar la consistencia de un viaje como unidad de negocio.

### Métodos principales

- validateTripReady(): Verifica que el viaje tenga cliente, conductor, vehículo y ruta antes de iniciar.
- 

## Factory: TripFactory

### Propósito principal

Crear instancias de **Trip** en estado inicial válido.

### Métodos principales

- createTrip(clientId, driverId, vehicleId, route): Genera un viaje en estado PENDING con todos los datos requeridos.
- 

## Domain Service: RoutePlanningService

### Propósito principal

Encapsular la lógica de planificación de rutas.

### Métodos principales

- generateRoute(origin, destination): Construye una ruta válida con segmentos, distancia y duración.
- 

## Domain Service: TripSchedulerService

### Propósito principal

Validar disponibilidad de recursos antes de asignarlos a un viaje.

### Métodos principales

- checkDriverAvailability(driverId, timeRange): Verifica si un conductor está libre.
  - checkVehicleAvailability(vehicleId, timeRange): Verifica si un vehículo está disponible.
- 

## Command: CreateTripCommand

### Propósito

Crear un nuevo viaje en estado PENDING con las referencias de cliente, conductor, vehículo y ruta.

### Parámetros

- clientId: Identificador del cliente.
  - driverId: Identificador del conductor.
  - vehicleId: Identificador del vehículo.
  - route: Ruta completa del viaje.
-

## **Command: AssignDriverToTripCommand**

### **Propósito**

Asignar un conductor disponible a un viaje existente y actualizar la referencia correspondiente.

### **Parámetros**

- tripId: Identificador único del viaje.
  - driverId: Identificador del conductor.
- 

## **Command: AssignVehicleToTripCommand**

### **Propósito**

Asignar un vehículo disponible a un viaje existente y actualizar la referencia correspondiente.

### **Parámetros**

- tripId: Identificador único del viaje.
  - vehicleId: Identificador del vehículo.
- 

## **Command: StartTripCommand**

### **Propósito**

Iniciar un viaje, cambiando su estado a EN CURSO y registrando la hora exacta de inicio.

### **Parámetros**

- tripId: Identificador único del viaje.
- 

## **Command: CompleteTripCommand**

### **Propósito**

Finalizar un viaje, cambiando su estado a COMPLETADO y registrando la hora de cierre.

### **Parámetros**

- tripId: Identificador único del viaje.
- 

## **Command: CancelTripCommand**

### **Propósito**

Cancelar un viaje, actualizar su estado a CANCELADO y guardar la razón de la cancelación.

### **Parámetros**

- tripId: Identificador único del viaje.
  - reason: Motivo de la cancelación.
- 

## **Command: UpdateRouteForTripCommand**

**Propósito**

Actualizar la ruta de un viaje antes de que inicie, garantizando que la información sea válida y actualizada.

**Parámetros**

- tripId: Identificador único del viaje.
- newRoute: Nueva ruta a asociar.

**Query: GetTripByIdQuery****Propósito**

Obtener la información completa de un viaje específico mediante su identificador único.

**Parámetros**

- tripId: Identificador único del viaje.
- 

**Query: GetTripsByStatusQuery****Propósito**

Listar los viajes filtrados por su estado (Pendiente, En curso, Completado o Cancelado).

**Parámetros**

- status: Estado de los viajes a consultar.
- 

**Query: GetTripsByClientIdQuery****Propósito**

Obtener todos los viajes asociados a un cliente específico.

**Parámetros**

- clientId: Identificador único del cliente.
- 

**Query: GetAllTripsQuery****Propósito**

Recuperar todos los viajes registrados en el sistema, sin aplicar filtros.

**Parámetros**

(No requiere parámetros)

---

**Event: TripCreatedEvent****Propósito**

Notificar que un nuevo viaje ha sido creado en el sistema.

**Parámetros**

- tripId: Identificador único del viaje.

- clientId: Identificador del cliente.
  - driverId: Identificador del conductor asignado.
  - vehicleId: Identificador del vehículo asignado.
  - route: Ruta definida para el viaje.
  - createdAt: Fecha y hora en que se creó el viaje.
- 

## Event: DriverAssignedEvent

### Propósito

Notificar que un conductor fue asignado a un viaje.

### Parámetros

- tripId: Identificador único del viaje.
  - driverId: Identificador del conductor asignado.
  - assignedAt: Fecha y hora de la asignación.
- 

## Event: VehicleAssignedEvent

### Propósito

Notificar que un vehículo fue asignado a un viaje.

### Parámetros

- tripId: Identificador único del viaje.
  - vehicleId: Identificador del vehículo asignado.
  - assignedAt: Fecha y hora de la asignación.
- 

## Event: TripStartedEvent

### Propósito

Notificar que un viaje ha iniciado oficialmente.

### Parámetros

- tripId: Identificador único del viaje.
  - startedAt: Fecha y hora de inicio del viaje.
- 

## Event: TripCompletedEvent

### Propósito

Notificar que un viaje se ha completado satisfactoriamente.

### Parámetros

- tripId: Identificador único del viaje.
  - completedAt: Fecha y hora de finalización del viaje.
-

## Event: TripCancelledEvent

### Propósito

Notificar que un viaje ha sido cancelado.

### Parámetros

- tripId: Identificador único del viaje.
- reason: Motivo de la cancelación.
- cancelledAt: Fecha y hora en que se canceló el viaje.

## 4.2.5.2. Interface Layer.

### Controllers

- TripController: Controlador que maneja las solicitudes relacionadas con los viajes. Atiende operaciones como crear un nuevo viaje, asignar un conductor, actualizar la ruta, iniciar, completar o cancelar un viaje, así como consultar información de viajes por identificador, estado, cliente o recuperar todos los viajes registrados.
- RouteController: Controlador que maneja las solicitudes relacionadas con las rutas de los viajes. Permite registrar una nueva ruta, actualizarla antes del inicio de un viaje y consultar la información de rutas específicas o asociadas a un viaje.

## 4.2.5.3. Application Layer.

### Command Services

- TripCommandService: Se encarga de recibir y coordinar los comandos relacionados a un viaje. Dentro de él se manejan distintos handlers, cada uno especializado en ejecutar un comando específico como iniciar, completar, cancelar o asignar recursos al viaje.
- RouteCommandService: Se encarga de coordinar los comandos relacionados con rutas. Administra la creación, actualización y recálculo de rutas para garantizar que los trayectos estén completos y actualizados antes de iniciar un viaje.

---

### Query Services

- TripQueryService: Se encarga de atender las consultas relacionadas a los viajes. Contiene handlers que procesan queries para obtener información, por ejemplo: consultar un viaje por su identificador, listar viajes por estado o recuperar todos los viajes de un cliente.
- RouteQueryService: Atiende las consultas relacionadas a las rutas de los viajes. Permite obtener información de rutas específicas o de las rutas asociadas a un viaje.

---

### Event Services

- TripEventService: Se encarga de atender los eventos relacionados a un viaje. Dentro de él se gestionan distintos servicios especializados que reaccionan a cada evento, como creación, asignación de recursos, inicio, finalización o cancelación del viaje, ejecutando las acciones necesarias después de que ocurren.

#### 4.2.5.4. Infrastructure Layer.

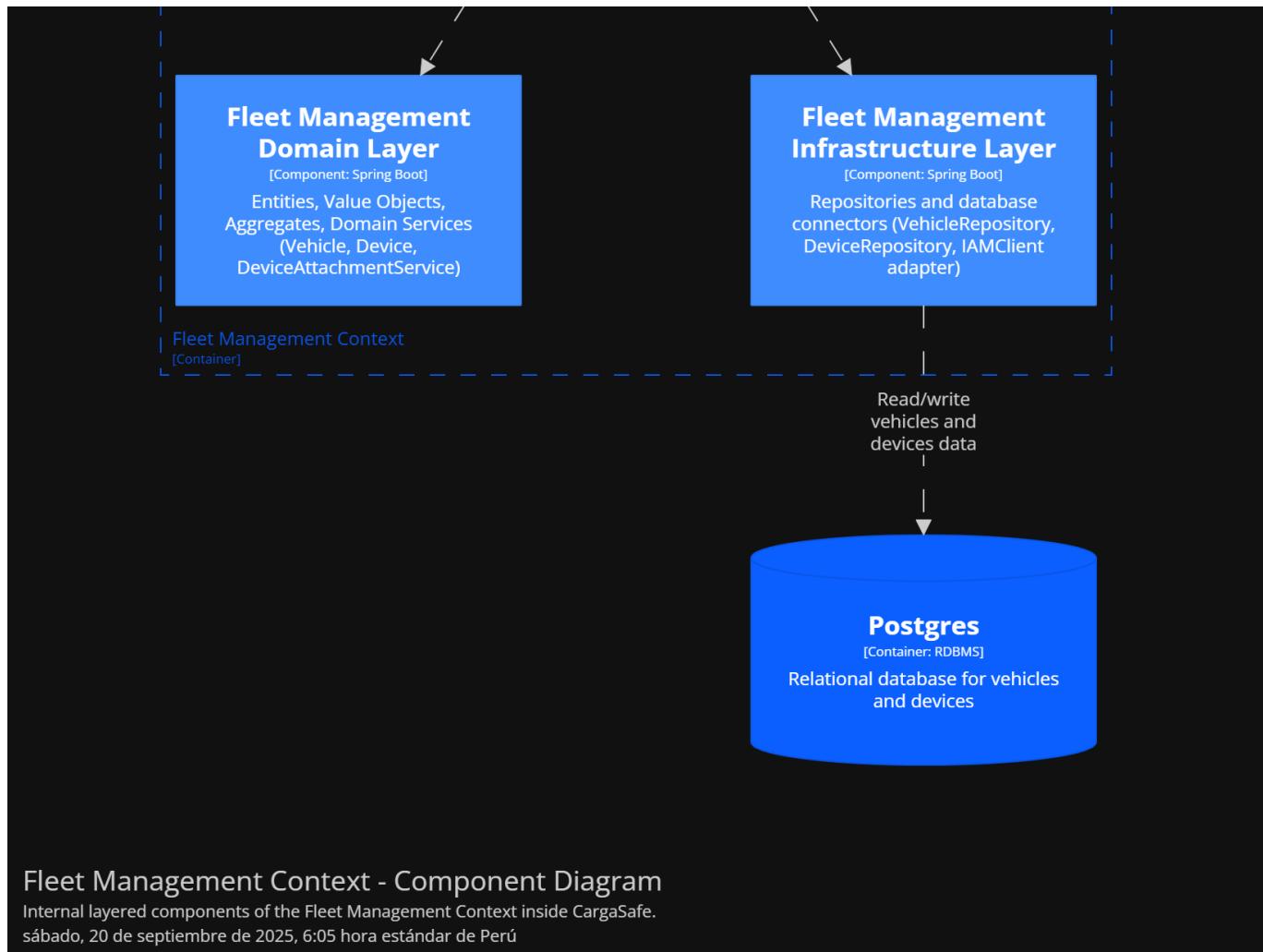
##### Repositories

- ITripRepository: Repositorio que define las operaciones de acceso a los viajes, como guardar, actualizar y recuperar información de un viaje.
- IRouteRepository: Repositorio que define las operaciones de acceso a las rutas, como registrar nuevas rutas, actualizarlas y consultarlas en relación con un viaje.

#### 4.2.5.5. Bounded Context Software Architecture Component Level Diagrams.

Diagrama de componentes - Backend - Trip Management





## Fleet Management Context - Component Diagram

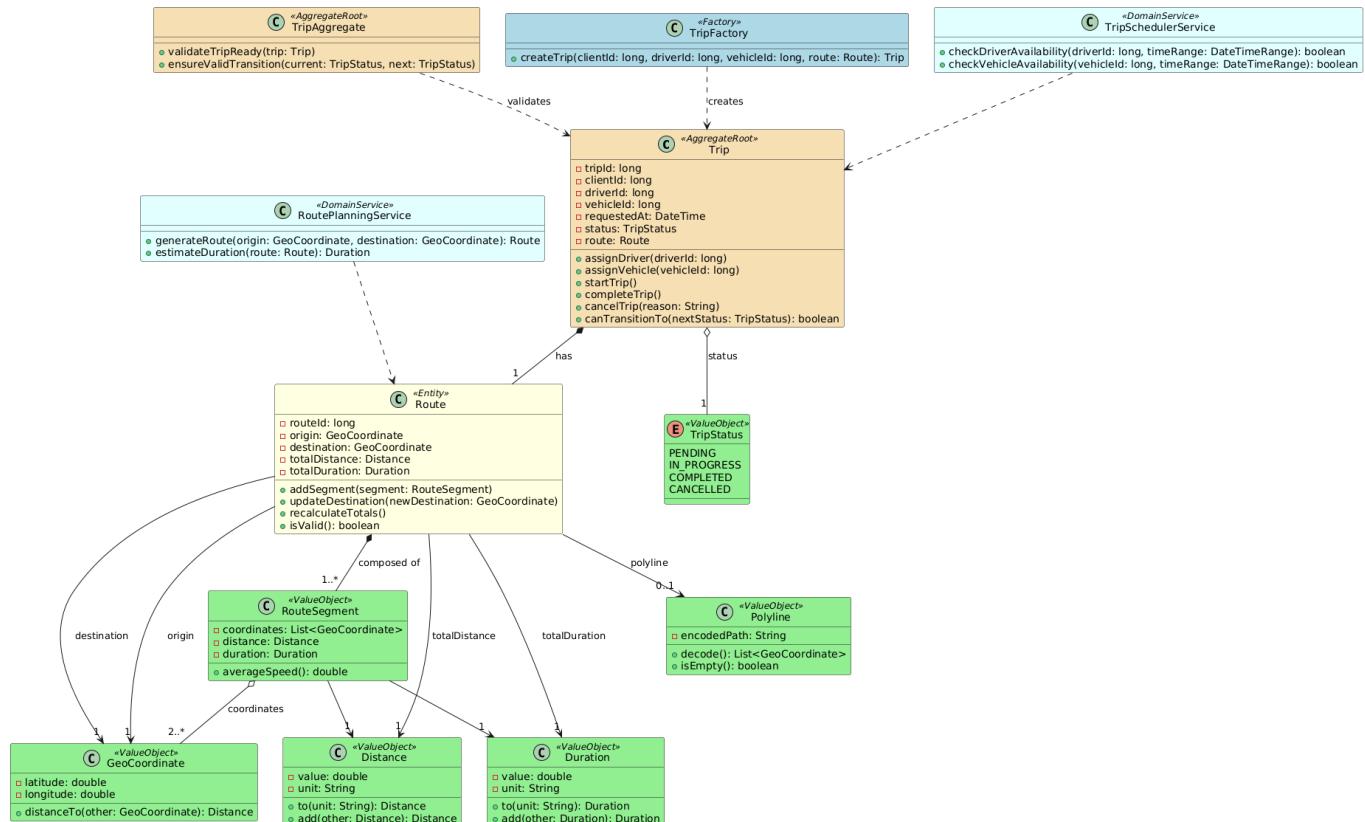
Internal layered components of the Fleet Management Context inside CargaSafe.  
sábado, 20 de septiembre de 2025, 6:05 hora estándar de Perú

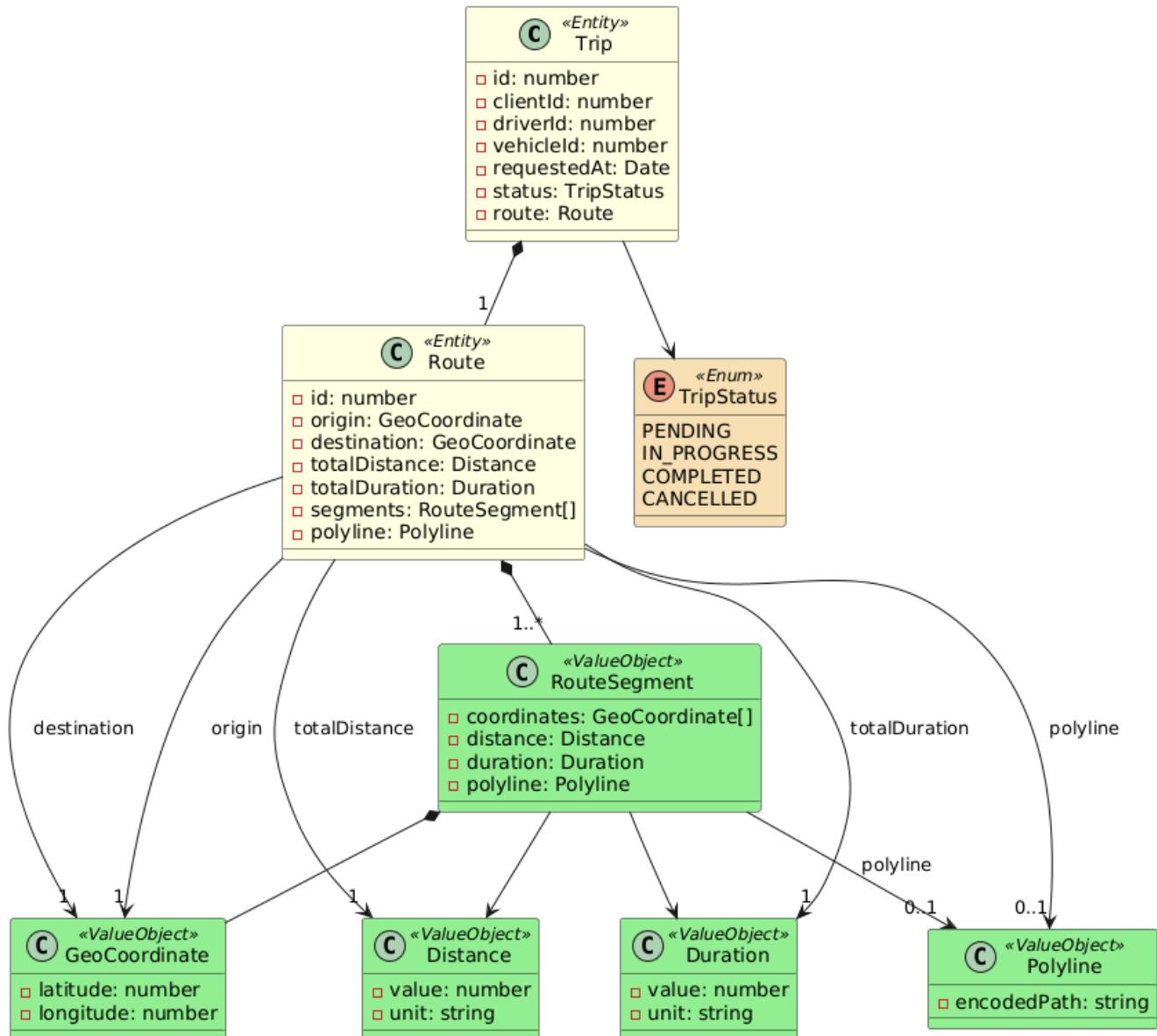
Diagrama de componentes - Application Web - Trip Management

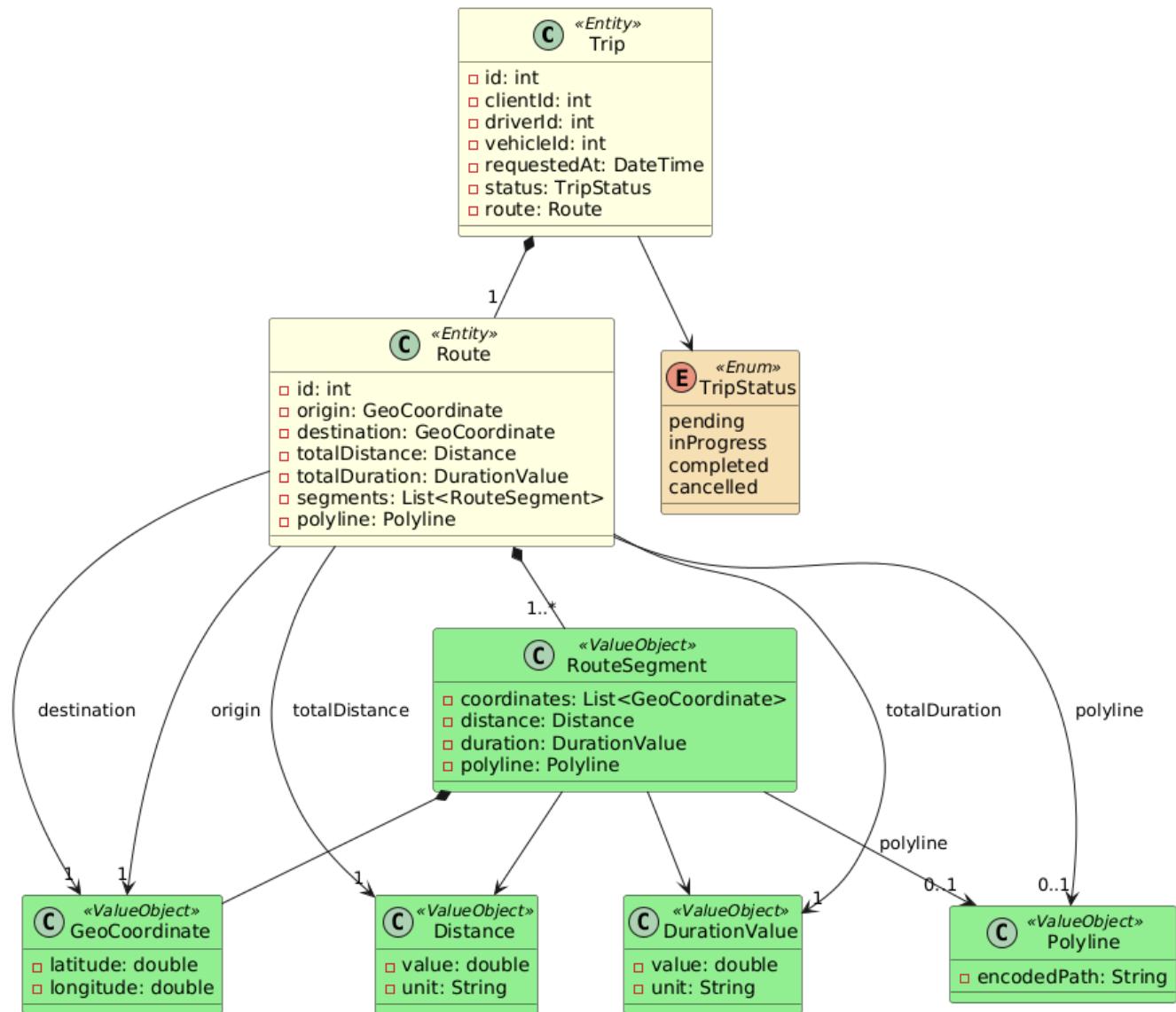
Diagrama de componentes - Mobile App - Trip Management

### 4.2.5.6. Bounded Context Software Architecture Code Level Diagrams.

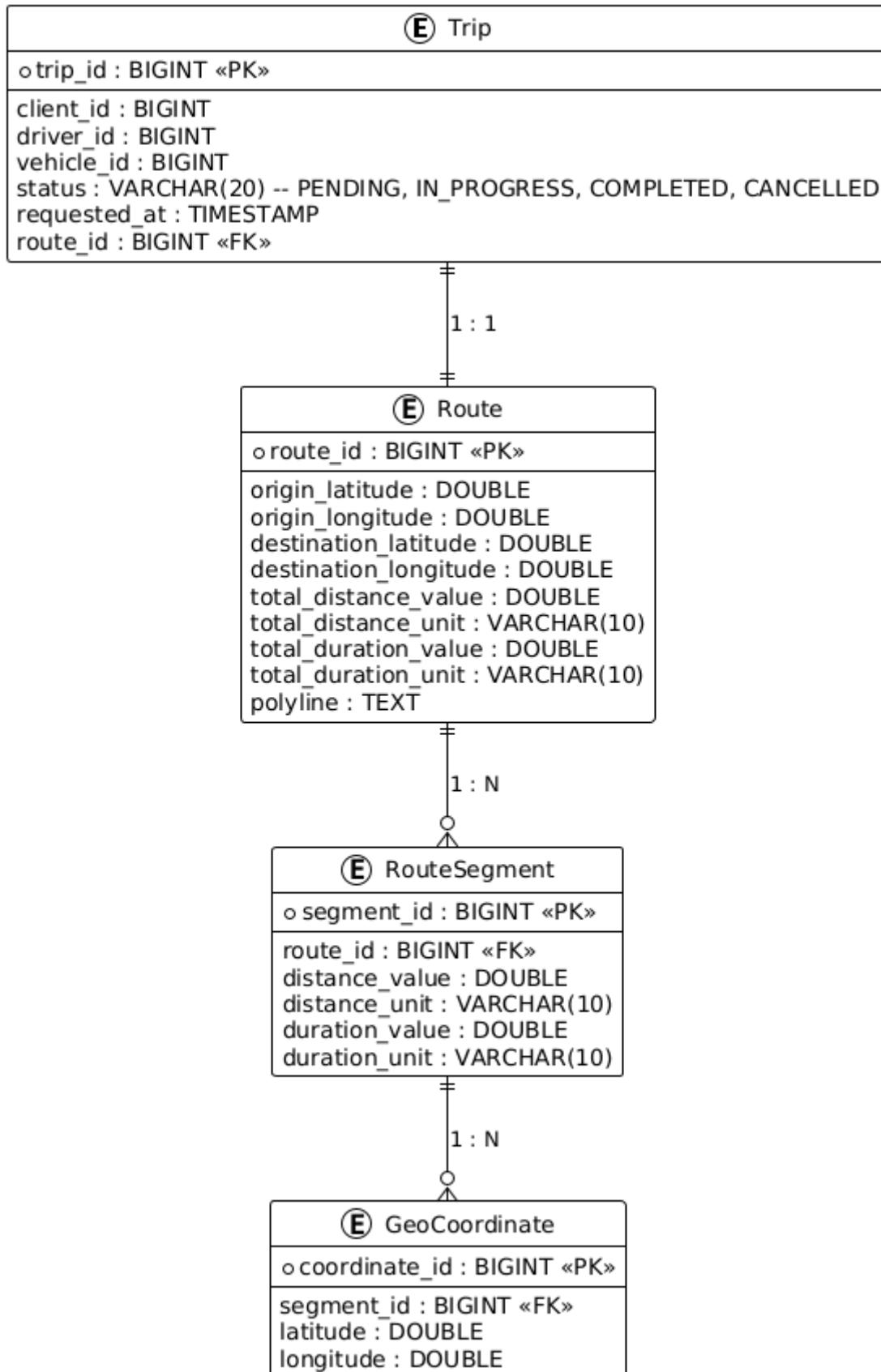
#### 4.2.5.6.1. Bounded Context Domain Layer Class Diagrams.







#### 4.2.5.6.2. Bounded Context Database Design Diagram.



#### 4.2.6. Bounded Context: Fleet Management

##### 4.2.6.1. Domain Layer

###### Entidades Principales

###### Vehicle (Aggregate Root)

- Propósito: Representa una unidad de transporte registrada en el sistema de flota.
- Atributos principales:
  - id (UUID)
  - plate (única)
  - make
  - model
  - active (bool)
  - tenantId
  - createdAt
  - updatedAt
- Métodos principales:
  - register(data), update(data)
  - deactivate() / activate()

## Device (Entity)

- Propósito: Registrar y administrar dispositivos/sensores instalados en unidades.
- Atributos:
  - id,
  - type (temp|gps|humidity|door...)
  - serial (único)
  - lastCalibrationAt?
  - active
  - vehicleId?
  - tenantId
  - createdAt
- Métodos:
  - create(data)
  - update(data)
  - attachToVehicle(vehicleId)
  - detach(),
  - activate()
  - deactivate()

## Value Objects

- Plate (formato y unicidad por país/tenant).

## Domain Services

- DeviceAttachmentService: valida que el vehículo exista y pertenezca al mismo tenantId antes de asociar/desasociar.

## Commands

- RegisterVehicleCommand
- UpdateVehicleCommand
- ActivateVehicleCommand
- DeactivateVehicleCommand
- CreateDeviceCommand
- UpdateDeviceCommand
- AttachDeviceCommand
- DetachDeviceCommand
- ActivateDeviceCommand
- DeactivateDeviceCommand

## Queries

- GetVehicleByIdQuery
- ListVehiclesQuery
- GetDeviceByIdQuery
- ListDevicesQuery (con filtro por vehicleId)

### 4.2.6.2. Interface Layer

#### Controllers Principales (HTTP REST)

##### VehicleController

- POST /vehicles : Registrar vehículo
- PUT /vehicles/{id} : Actualizar datos
- PATCH /vehicles/{id}/activate | /deactivate
- GET /vehicles/{id} | GET /vehicles?search=&page=&size=

##### DeviceController

- POST /devices : Crear sensor
- PUT /devices/{id}
- POST /devices/{id}/attach/{vehicleId}
- POST /devices/{id}/detach
- PATCH /devices/{id}/activate | /deactivate
- GET /devices/{id} | GET /devices?vehicleId=&page=&size=

## Seguridad (ACLs)

- OIDC/JWT con IAM; scopes: fleet:read, fleet:write, fleet:admin.

### 4.2.6.3. Application Layer

#### Command Services

##### **VehicleCommandServiceImpl**

- Propósito: gestionar creación/actualización/activación de vehículos.
- Métodos:
  - handle(RegisterVehicleCommand)
  - handle(UpdateVehicleCommand)
  - handle(ActivateVehicleCommand) / handle(DeactivateVehicleCommand)
- Validaciones: fleet:write/fleet:admin; unicidad plate por tenantId; existencia de vehículo en updates.
- Dependencias: VehicleRepository, IAMClient.

##### **DeviceCommandServiceImpl**

- Propósito: ciclo de vida de dispositivos y asociación a vehículos.
- Métodos:
  - handle(CreateDeviceCommand)
  - handle(UpdateDeviceCommand)
  - handle(AttachDeviceCommand) / handle(DetachDeviceCommand)
  - handle(ActivateDeviceCommand) / handle(DeactivateDeviceCommand)
- Validaciones: fleet:write/fleet:admin; unicidad serial por tenantId; en attach: vehículo existe y mismo tenantId, dispositivo no asociado; en detach: dispositivo asociado.
- Dependencias:
  - DeviceRepository
  - VehicleRepository
  - IAMClient

#### Query Services

##### **FleetQueryServiceImpl**

- Propósito: consultas de solo lectura para vehículos y dispositivos.
- Métodos:
  - handle(GetVehicleByIdQuery) / handle(ListVehiclesQuery)

- handle(GetDeviceByIdQuery) / handle(ListDevicesQuery) (filtros por vehicleId)
- Dependencias:
  - VehicleRepository
  - DeviceRepository
  - IAMClient (scope fleet:read, resolución de tenantId)

## Outbound Services (ACL)

### IAMClient

- assertScope(token, scope) — valida fleet:read|write|admin.
- getTenantId(token) — scoping multi-tenant.
- getUserId(token) — auditoría

## Consideraciones transversales

- Transacciones: 1 comando = 1 transacción (implementada por Infra, p. ej. @Transactional).
- Idempotencia: RegisterVehicle por (tenantId, plate); CreateDevice por (tenantId, serial).
- Auditoría: createdAt/updatedAt (y opcional createdBy/updatedBy).
- Errores estándar: Forbidden (ACL), NotFound, Conflict, ValidationError.

### 4.2.6.4. Infrastructure Layer

#### Repositories

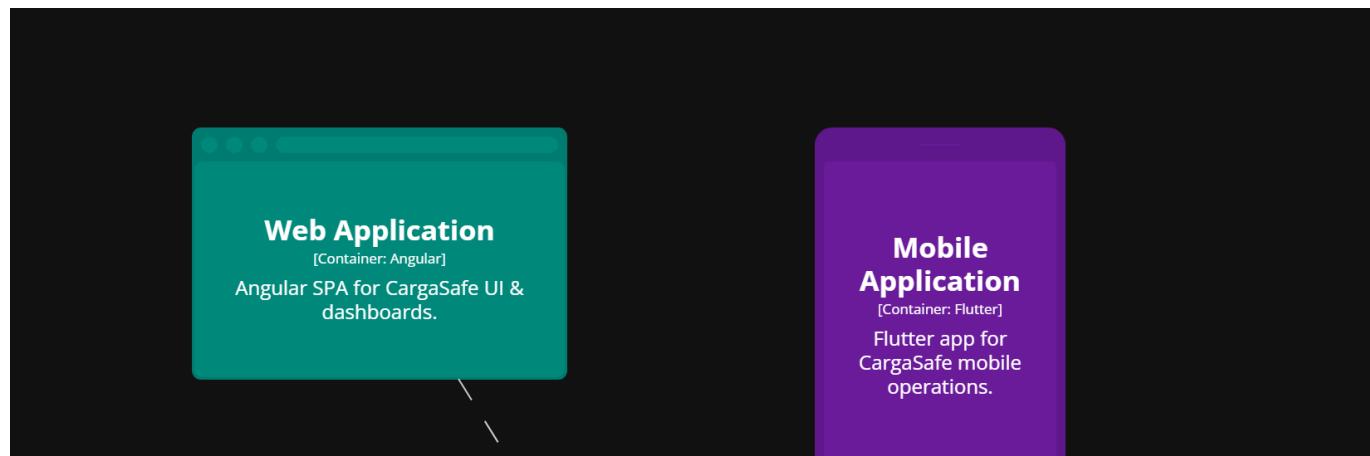
- VehicleRepository (PostgreSQL)
- DeviceRepository (PostgreSQL)

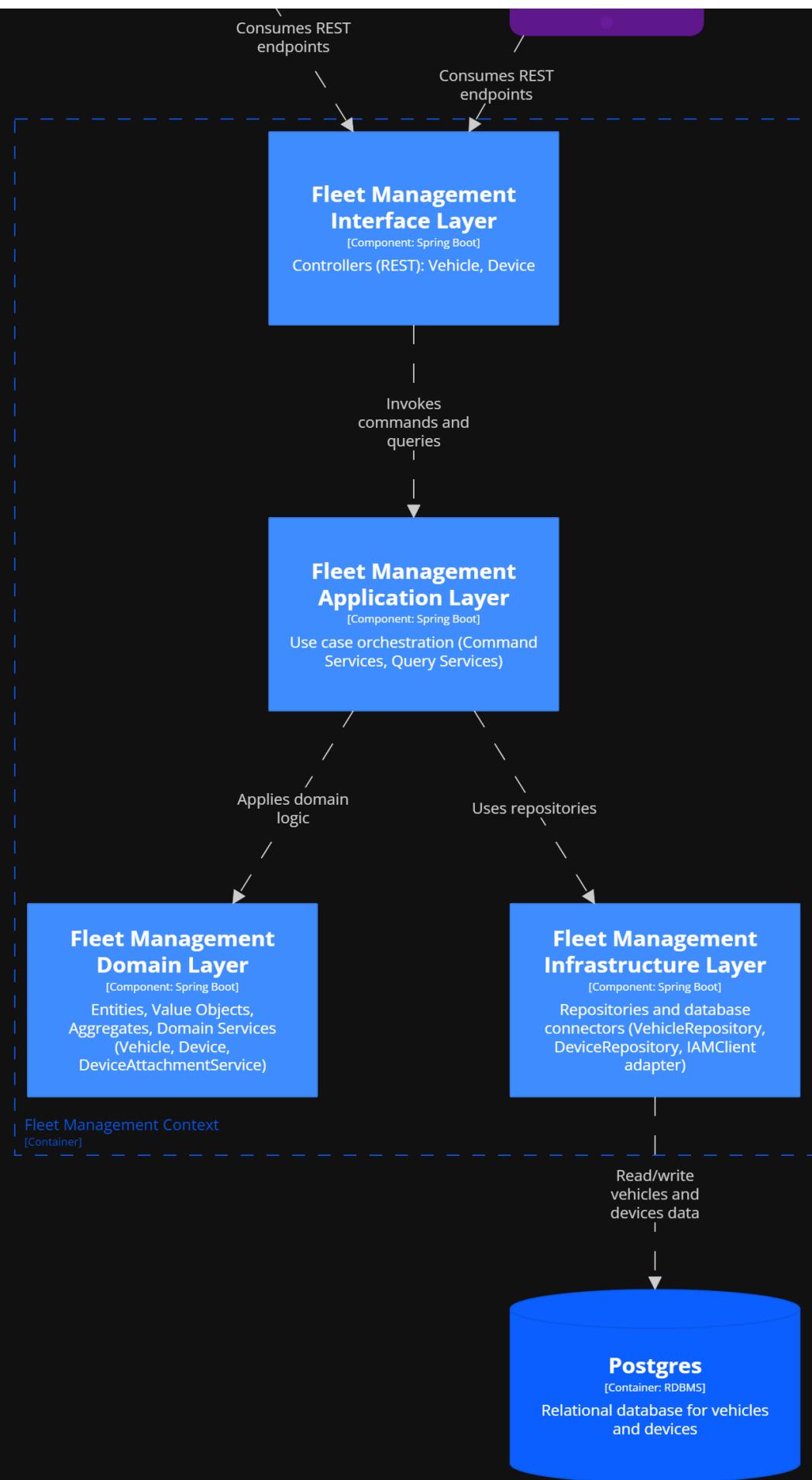
#### Integraciones y seguridad

- IAMClient para validar JWT/claims (ACLS).

### 4.2.6.5. Bounded Context Software Architecture Component Level Diagrams.

Diagrama de componentes - Backend - Fleet Management



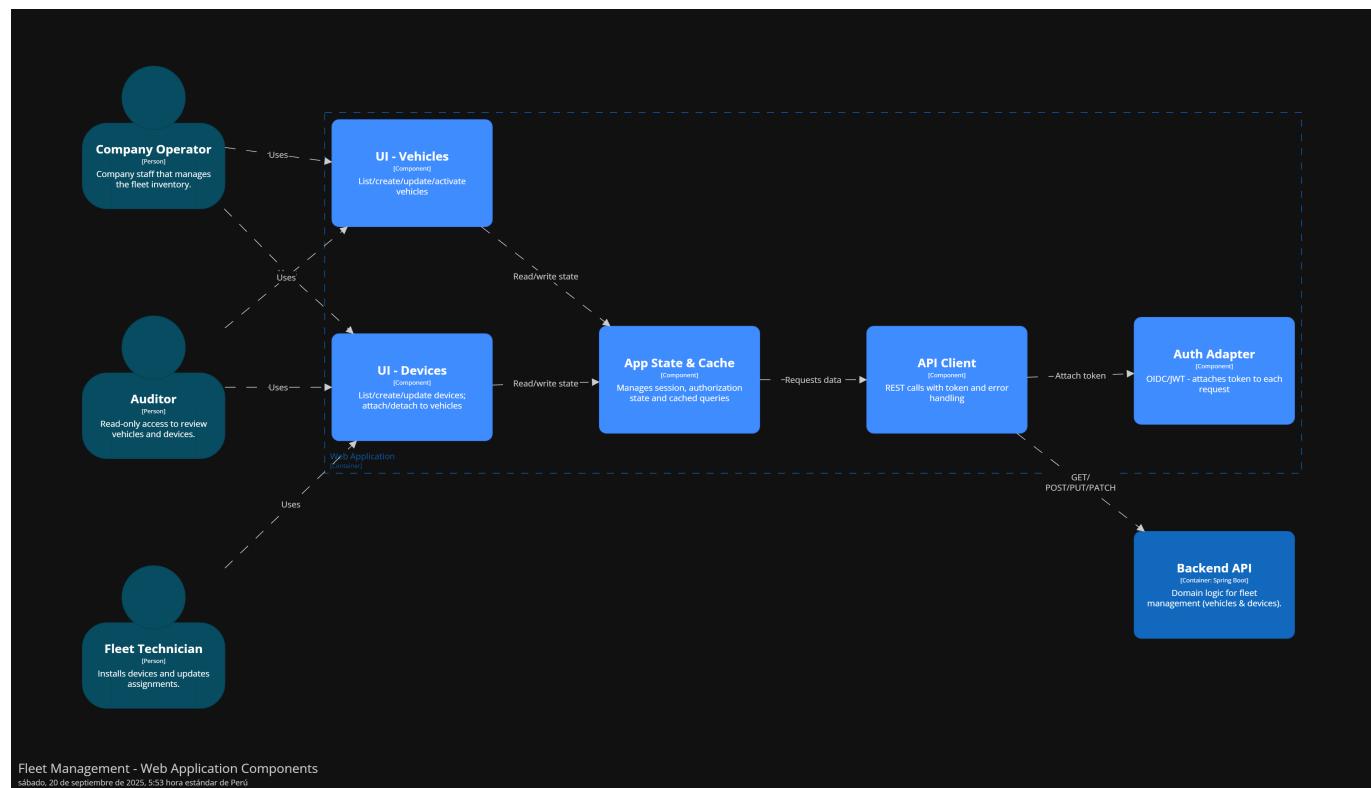


Fleet Management Context - Component Diagram

Internal layered components of the Fleet Management Context inside GameSafe

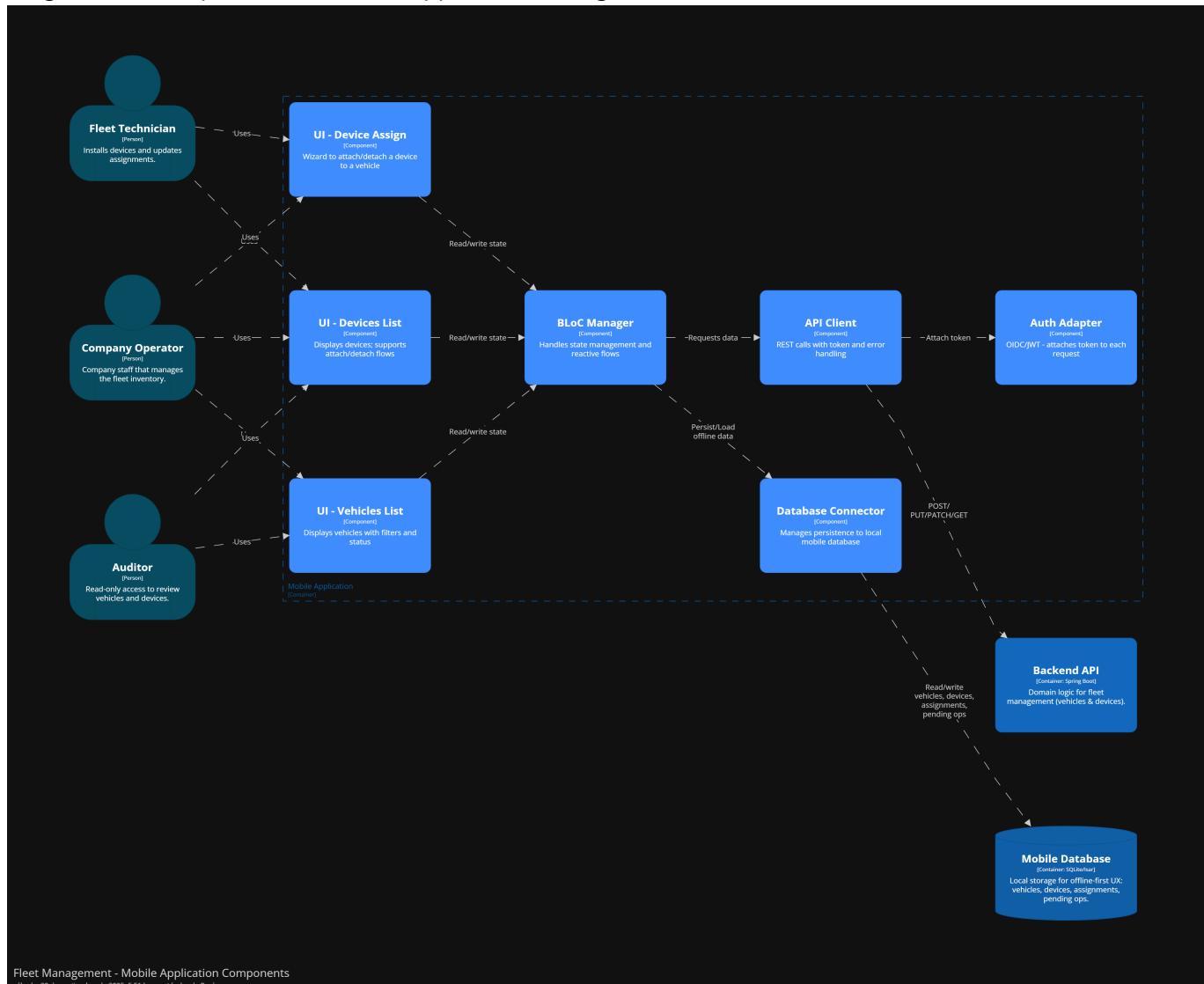
Internal layered components of the Fleet Management context inside Cargasare.  
sábado, 20 de septiembre de 2025, 6:05 hora estándar de Perú

## Diagrama de componentes - Application Web - Fleet Management



Fleet Management - Web Application Components  
sábado, 20 de septiembre de 2025, 5:53 hora estándar de Perú

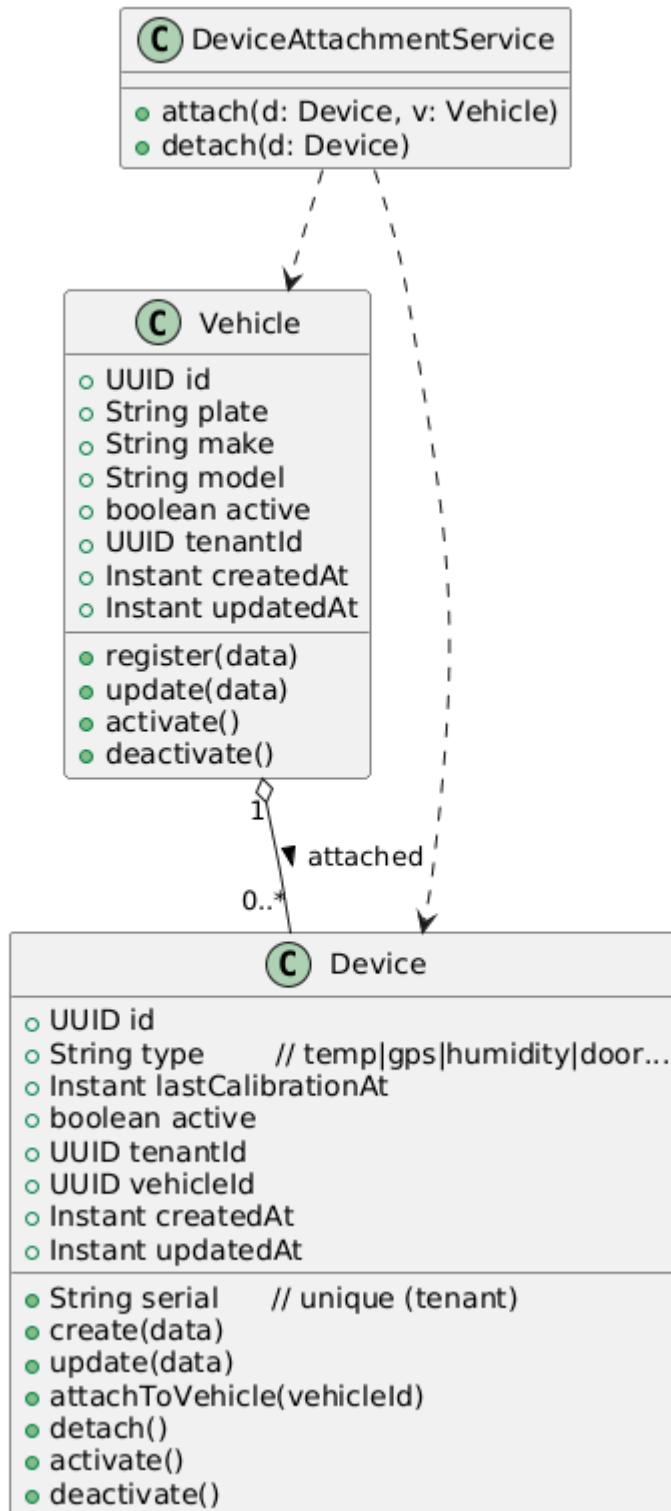
## Diagrama de componentes - Mobile App - Fleet Management

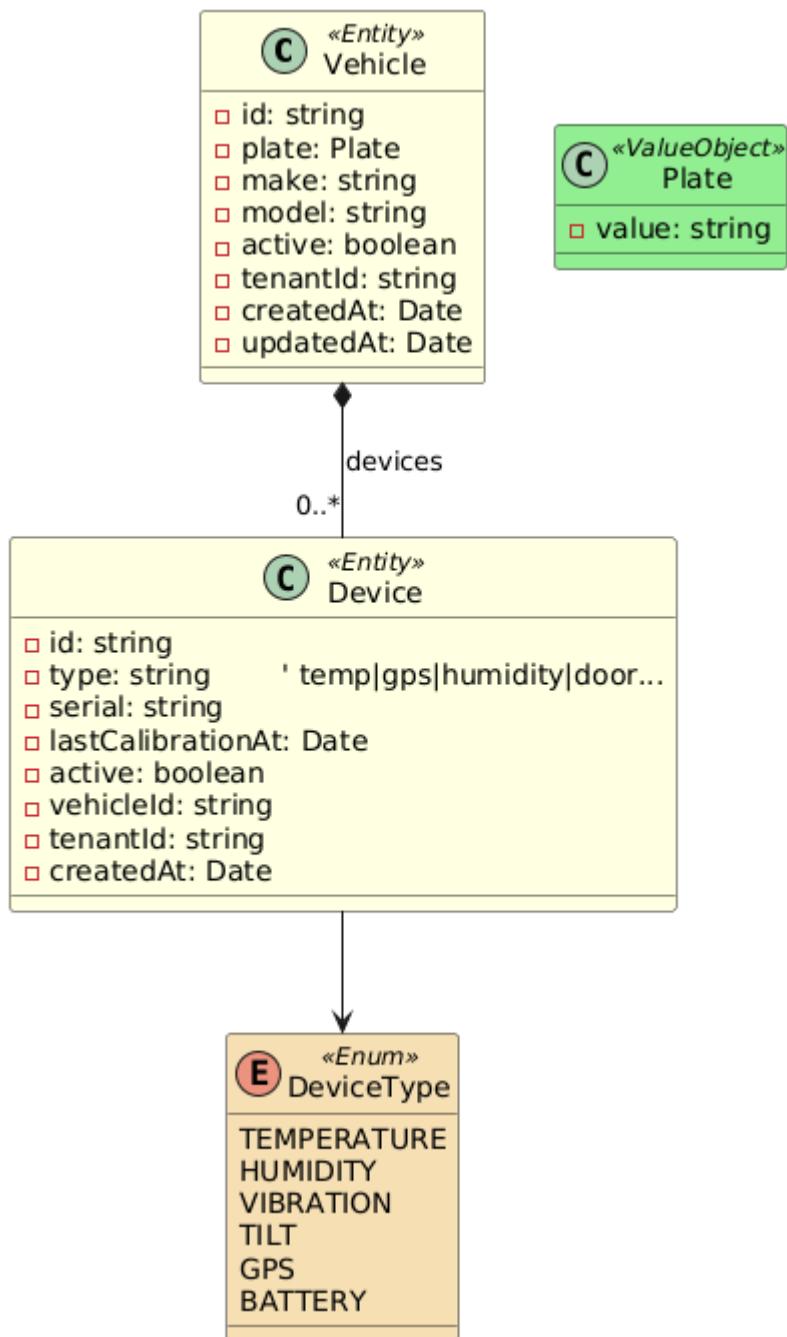


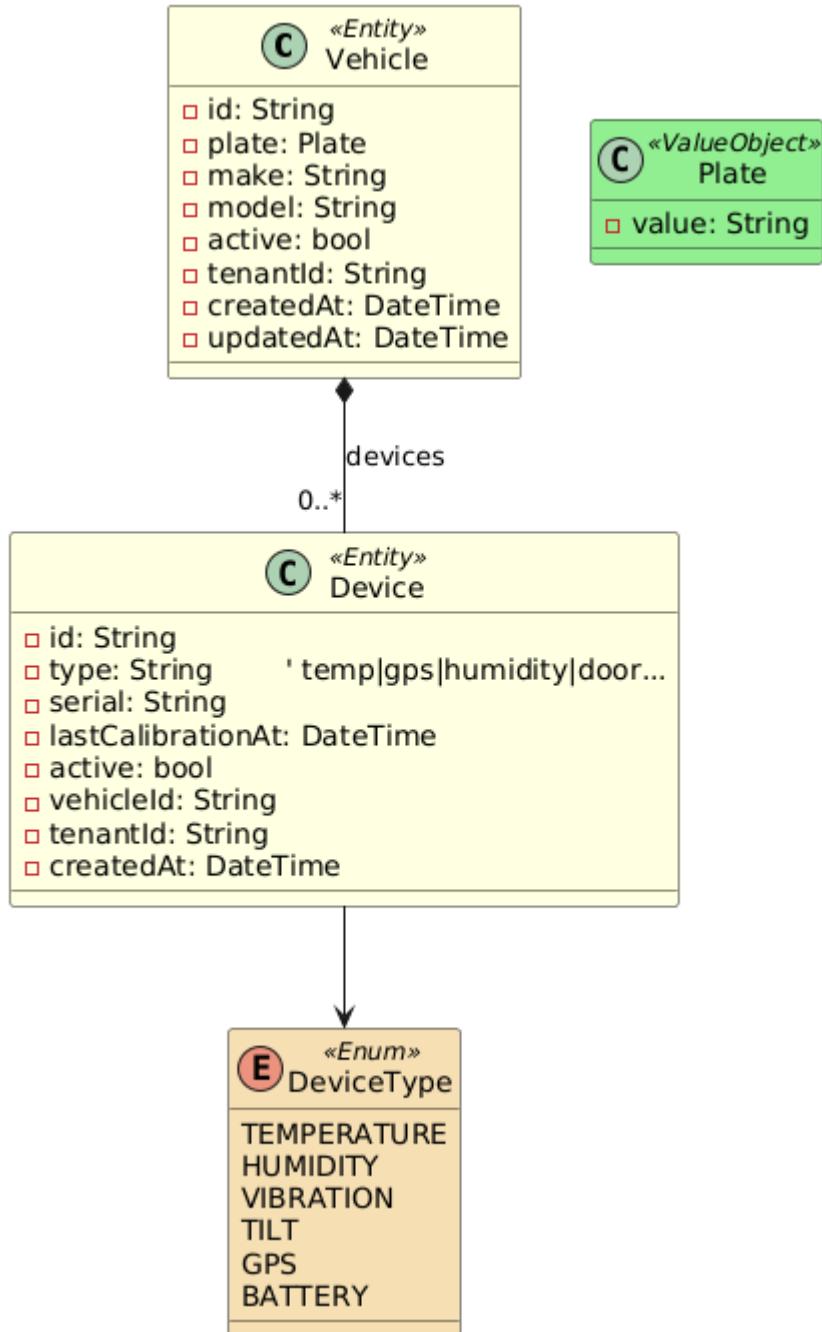
### 4.2.5.6. Bounded Context Software Architecture Code Level Diagrams.

#### 4.2.5.6.1. Bounded Context Domain Layer Class Diagrams.

## Fleet Management - Backend Domain Layer Class Diagram

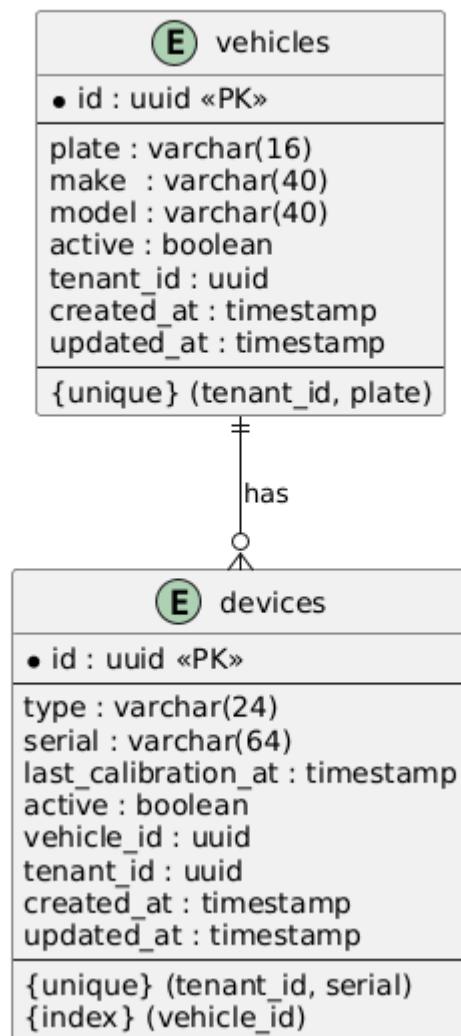






#### 4.2.5.6.2. Bounded Context Database Design Diagram.

## Fleet Management - Database Design (PostgreSQL)



### 4.2.7. Bounded Context: Profile and Preferences Management

#### 4.2.7.1. Domain Layer.

##### Entity: Profile (Aggregate Root)

###### Propósito principal

Representar el perfil dentro del sistema, siempre asociado a un **userId**.

###### Atributos principales

- id: Identificador único del perfil.
- userId: Identificador externo obligatorio asociado al perfil.
- fullName: Nombre completo.
- phoneNumber: Número de contacto (opcional).
- avatarUrl: Imagen de perfil o avatar (opcional).
- createdAt: Fecha de creación.
- updatedAt: Fecha de última actualización.

###### Métodos principales

- updateContactInfo(newPhone)

- updateAvatar(newAvatarUrl)
  - updateName(newName)
- 

## Entity: Preferences

### Propósito principal

Almacenar configuraciones personalizadas de idioma, zona horaria y notificaciones de alertas.

### Atributos principales

- id: Identificador único de las preferencias.
- profileId: Identificador del perfil asociado.
- language: Idioma preferido.
- timeZone: Zona horaria configurada.
- alertEmailEnabled: Recibir alertas por correo electrónico.
- alertPushEnabled: Recibir alertas como notificaciones push en la aplicación.
- alertSmsEnabled: Recibir alertas vía SMS (opcional).

### Métodos principales

- updateLanguage(language)
  - updateTimeZone(timeZone)
  - enableEmailAlerts(flag)
  - enablePushAlerts(flag)
  - enableSmsAlerts(flag)
- 

## Value Object: PhoneNumber

- countryCode
- number

## Value Object: Language

- code (ej. "es", "en")
- name

## Value Object: TimeZone

- code (ej. "UTC-5")
- 

## Aggregate: ProfileAggregate

- updateProfileAndPreferences(profile, preferences)
- 

## Factory: ProfileFactory

- createProfile(fullName, userId)
-

**Domain Service: PreferencesValidationService**

- validateNotificationSettings(emailFlag, pushFlag, smsFlag)
  - validateLanguage(language)
- 

**Command: CreateProfileCommand**

- fullName
- phoneNumber
- avatarUrl
- userId

**Command: UpdateProfileCommand**

- id
- newName
- newPhone
- newAvatarUrl

**Command: UpdatePreferencesCommand**

- id
  - language
  - timeZone
  - alertEmailEnabled
  - alertPushEnabled
  - alertSmsEnabled
- 

**Query: GetProfileByIdQuery**

- id

**Query: GetPreferencesByProfileIdQuery**

- profileId
- 

**Event: ProfileCreatedEvent**

- id
- userId
- createdAt

**Event: ProfileUpdatedEvent**

- id
- updatedAt

**Event: PreferencesUpdatedEvent**

- id

- updatedAt
  - changes
- 

#### 4.2.7.2. Interface Layer.

##### Controllers

- **ProfileController:** Maneja las solicitudes para crear, actualizar y consultar perfiles.
- **PreferencesController:** Maneja las solicitudes para modificar y consultar preferencias (idioma, zona horaria, notificaciones).

#### 4.2.7.3. Application Layer.

##### Command Services

- **ProfileCommandService:** Coordina la ejecución de comandos relacionados con perfiles (creación y actualización).
  - **PreferencesCommandService:** Coordina la ejecución de comandos relacionados con preferencias (idioma, zona horaria, notificaciones).
- 

##### Query Services

- **ProfileQueryService:** Atiende consultas sobre perfiles.
  - **PreferencesQueryService:** Atiende consultas sobre las preferencias de un perfil.
- 

##### Event Services

- **ProfileEventService:** Gestiona eventos relacionados con la creación y actualización de perfiles.
- **PreferencesEventService:** Gestiona eventos relacionados con la actualización de preferencias y su propagación a otros componentes.

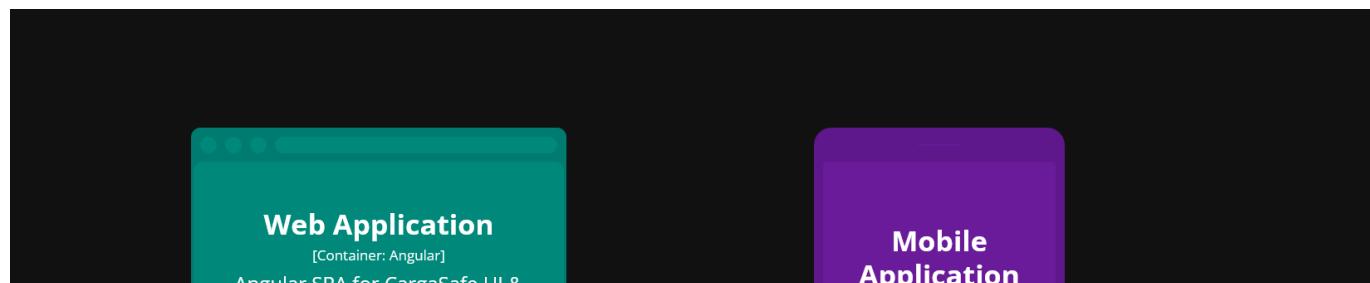
#### 4.2.7.4. Infrastructure Layer.

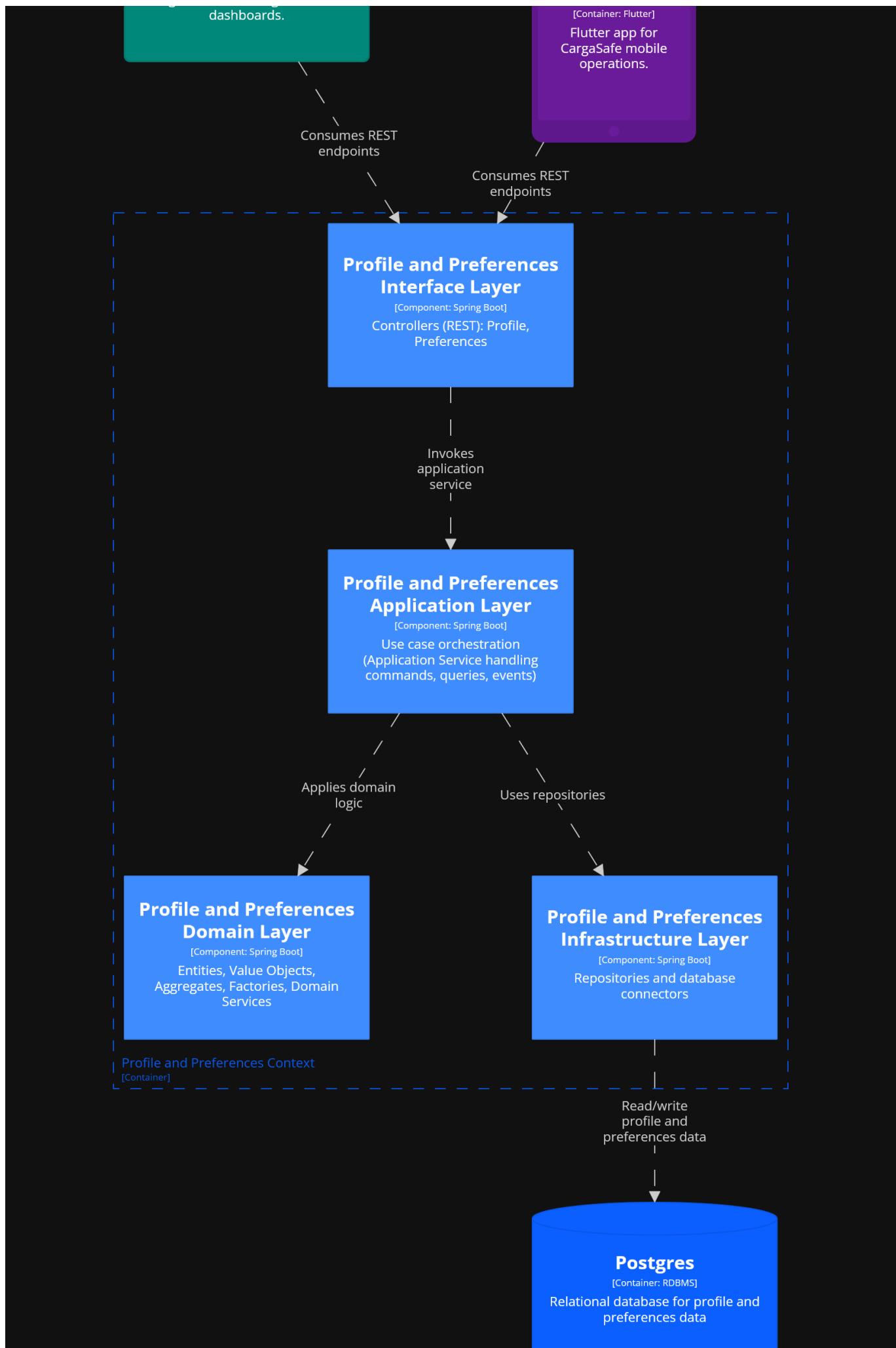
##### Repositories

- **IProfileRepository:** Define operaciones de acceso y persistencia para los perfiles.
- **IPreferencesRepository:** Define operaciones de acceso y persistencia para las preferencias.

#### 4.2.7.5. Bounded Context Software Architecture Component Level Diagrams.

Diagrama de componentes - Backend - Profiles and Preferences Management



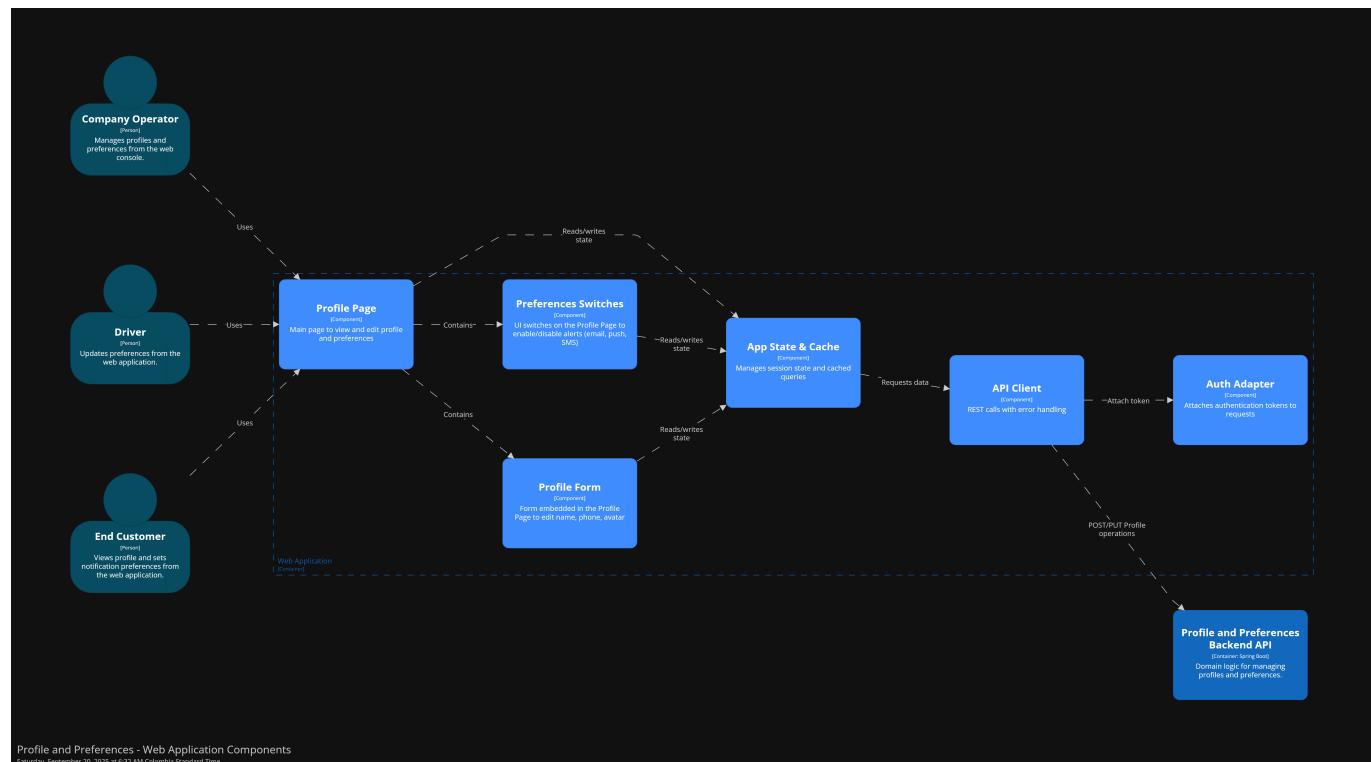


## Profile and Preferences Context - Component Diagram

Internal layered components of the Profile and Preferences Context inside CargaSafe.

Saturday, September 20, 2025 at 6:29 AM Colombia Standard Time

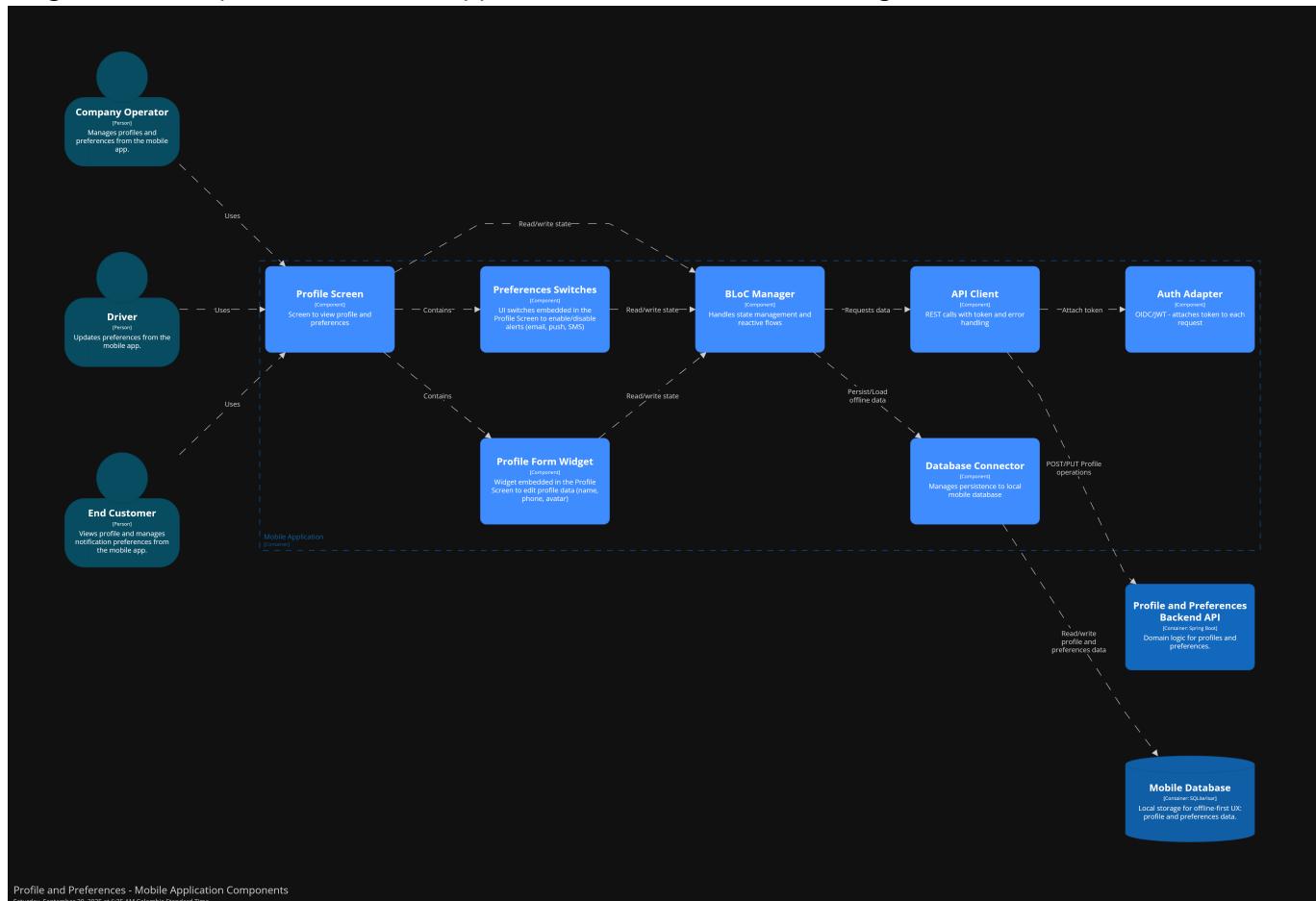
Diagrama de componentes - Application Web - Profiles and Preferences Management



Profile and Preferences - Web Application Components

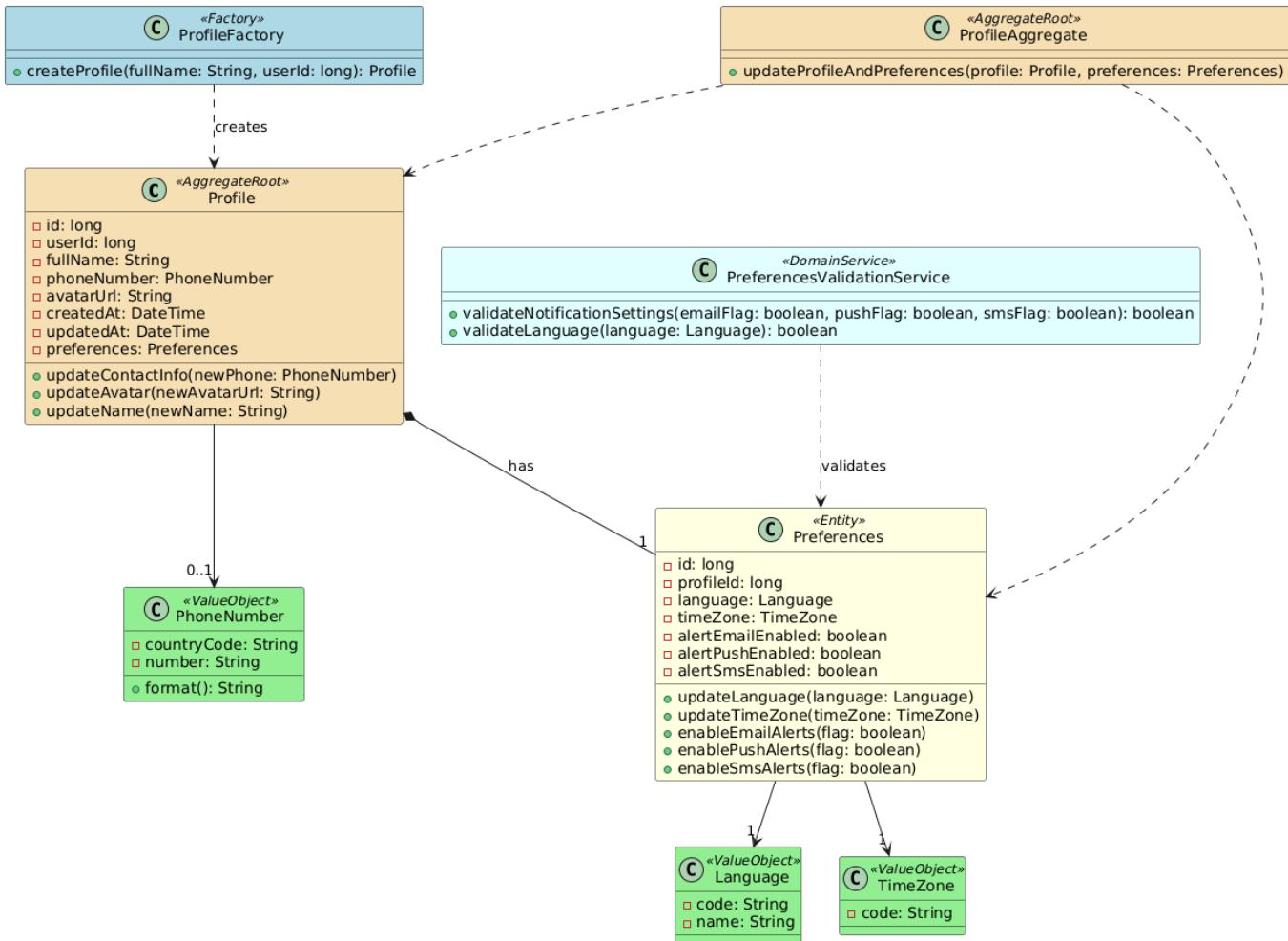
Saturday, September 20, 2025 at 6:32 AM Colombia Standard Time

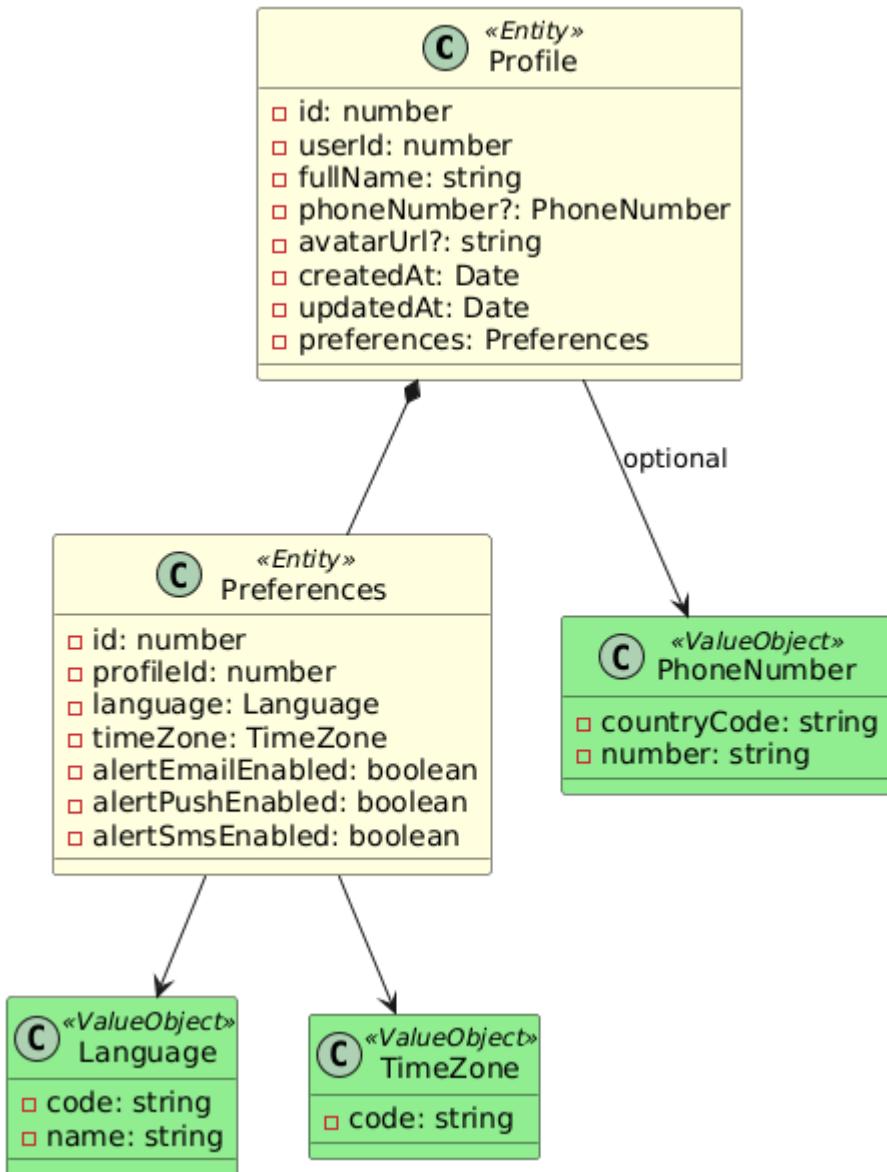
## Diagrama de componentes - Mobile App - Profiles and Preferences Management

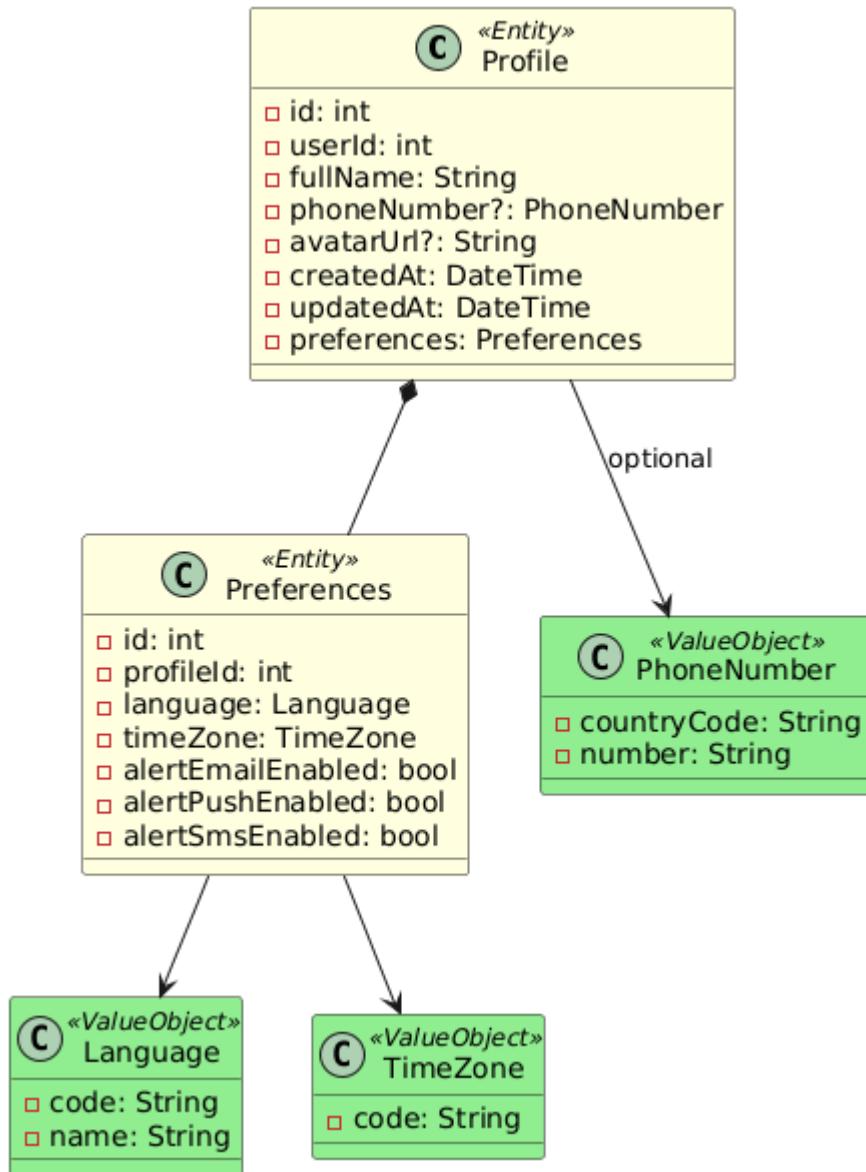


### 4.2.7.6. Bounded Context Software Architecture Code Level Diagrams.

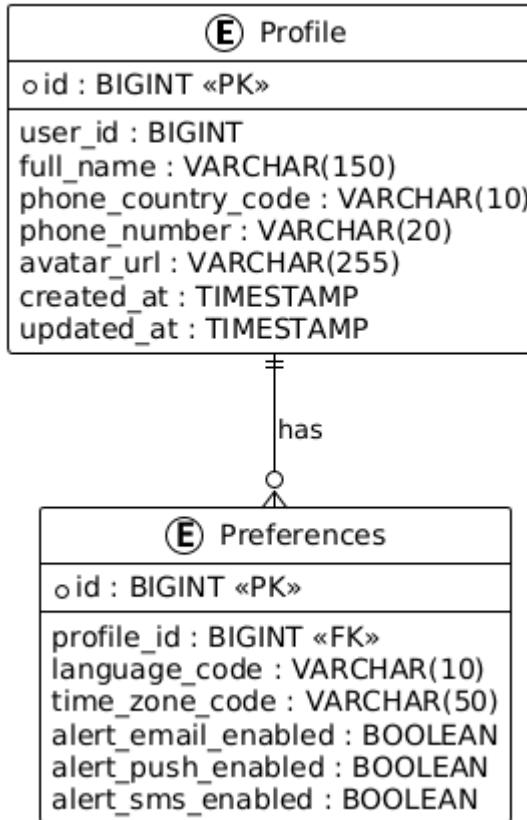
#### 4.2.7.6.1. Bounded Context Domain Layer Class Diagrams.







#### 4.2.7.6.2. Bounded Context Database Design Diagram



#### 4.2.8. Bounded Context: Visualization Analytics

##### 4.2.8.1. Domain Layer

###### Entidades Principales

###### Dashboard (Aggregate Root)

- **Propósito:** Representa un dashboard personalizable con widgets y métricas específicas
- **Atributos principales:**
  - `id`: Identificador único
  - `name`: Nombre del dashboard
  - `userId`: Propietario del dashboard
  - `layout`: Configuración de layout de widgets
  - `isDefault`: Indica si es dashboard por defecto
  - `createdAt, updatedAt`: Timestamps de auditoría
- **Métodos principales:**
  - `addWidget(widget)`: Agrega widget al dashboard
  - `removeWidget(widgetId)`: Remueve widget
  - `updateLayout(layout)`: Actualiza disposición de widgets
  - `clone()`: Crea copia del dashboard

###### Widget (Entity)

- **Propósito:** Componente visual que muestra métricas específicas
- **Atributos principales:**
  - `id`: Identificador único
  - `type`: Tipo de widget (CHART, KPI, TABLE, MAP)

- **title**: Título del widget
- **dataSource**: Fuente de datos
- **configuration**: Configuración específica del widget
- **position**: Posición en el dashboard
- **Métodos principales:**
  - **updateConfiguration(config)**: Actualiza configuración
  - **refresh()**: Refresca datos del widget
  - **validateConfiguration()**: Valida configuración del widget

## Report (Entity)

- **Propósito:** Reporte generado con datos históricos y métricas
- **Atributos principales:**
  - **id**: Identificador único
  - **name**: Nombre del reporte
  - **type**: Tipo de reporte (TRIP\_SUMMARY, COMPLIANCE, PERFORMANCE)
  - **parameters**: Parámetros del reporte
  - **generatedAt**: Fecha de generación
  - **format**: Formato del reporte (PDF, EXCEL, CSV)
- **Métodos principales:**
  - **generate()**: Genera el reporte
  - **schedule(frequency)**: Programa generación automática
  - **export(format)**: Exporta en formato específico

## ChartData (Entity)

- **Propósito:** Datos procesados para visualización en charts
- **Atributos principales:**
  - **id**: Identificador único
  - **chartType**: Tipo de gráfico (LINE, BAR, PIE, SCATTER)
  - **dataPoints**: Puntos de datos
  - **labels**: Etiquetas de los ejes
  - **metadata**: Metadatos adicionales
- **Métodos principales:**
  - **addDataPoint(point)**: Agrega punto de dato
  - **aggregate(groupBy)**: Agrupa datos
  - **filter(criteria)**: Filtra datos

## Value Objects

- **TimeRange**: Rango de tiempo para consultas
- **ChartConfiguration**: Configuración específica de gráficos
- **KPIMetric**: Métrica de rendimiento clave
- **DataFilter**: Filtros aplicados a datos
- **ColorSchema**: Esquema de colores para visualizaciones

## Domain Services

- **DataAggregationService**: Agregación y cálculo de métricas

- **ChartRenderingService**: Lógica de renderizado de gráficos
- **ReportGenerationService**: Generación de reportes complejos
- **MetricsCalculationService**: Cálculo de KPIs y métricas derivadas

## Commands

- **CreateDashboardCommand**: Comando para crear dashboard
- **UpdateWidgetCommand**: Comando para actualizar widget
- **GenerateReportCommand**: Comando para generar reporte
- **RefreshDataCommand**: Comando para refrescar datos

## Queries

- **GetDashboardQuery**: Obtiene dashboard por ID
- **GetTripMetricsQuery**: Obtiene métricas de viajes
- **GetComplianceDataQuery**: Obtiene datos de cumplimiento
- **GetTemperatureHistoryQuery**: Obtiene historial de temperatura

## Events

- **DashboardCreatedEvent**: Dashboard creado
- **ReportGeneratedEvent**: Reporte generado
- **DataRefreshedEvent**: Datos refrescados
- **AlertThresholdExceededEvent**: Umbral de alerta excedido

### 4.2.8.2. Interface Layer

#### Controllers Principales

##### DashboardController

- **GET /dashboards**: Lista dashboards del usuario
- **POST /dashboards**: Crea nuevo dashboard
- **PUT /dashboards/{id}**: Actualiza dashboard
- **DELETE /dashboards/{id}**: Elimina dashboard
- **GET /dashboards/{id}/data**: Obtiene datos del dashboard

##### AnalyticsController

- **GET /analytics/trips**: Métricas de viajes
- **GET /analytics/compliance**: Datos de cumplimiento
- **GET /analytics/performance**: Métricas de rendimiento
- **GET /analytics/temperature-history**: Historial de temperatura

##### ReportController

- **POST /reports/generate**: Genera reporte bajo demanda
- **GET /reports**: Lista reportes generados
- **GET /reports/{id}/download**: Descarga reporte
- **POST /reports/schedule**: Programa reporte automático

## VisualizationController

- `GET /visualizations/chart-data`: Datos para gráficos
- `POST /visualizations/custom-chart`: Genera gráfico personalizado
- `GET /visualizations/kpis`: Obtiene KPIs calculados

### 4.2.8.3. Application Layer

#### Command Services

##### DashboardCommandService

- Maneja creación y modificación de dashboards
- Coordina actualización de widgets
- Gestiona permisos de acceso a dashboards

##### ReportCommandService

- Gestiona generación de reportes
- Maneja programación de reportes automáticos
- Coordina exportación en diferentes formatos

#### Query Services

##### AnalyticsQueryService

- Proporciona métricas y KPIs calculados
- Optimizado para consultas complejas de análisis
- Maneja agregaciones temporales

##### VisualizationQueryService

- Consultas optimizadas para gráficos
- Transformación de datos para visualización
- Cache de datos frecuentemente consultados

#### Event Handlers

##### TripCompletedEventHandler

- Procesa finalización de viajes
- Actualiza métricas de rendimiento
- Genera alertas si es necesario

##### TemperatureViolationEventHandler

- Procesa violaciones de temperatura
- Actualiza métricas de cumplimiento
- Notifica a dashboards relevantes

### 4.2.8.4. Infrastructure Layer

## Repositories

### **DashboardRepository** (implementa IDashboardRepository)

- Persistencia de dashboards y configuraciones
- Optimizado para consultas por usuario
- Cache de dashboards frecuentemente accedidos

### **ReportRepository** (implementa IReportRepository)

- Almacenamiento de reportes generados
- Gestión de archivos de reporte
- Limpieza automática de reportes antiguos

### **MetricsRepository** (implementa IMetricsRepository)

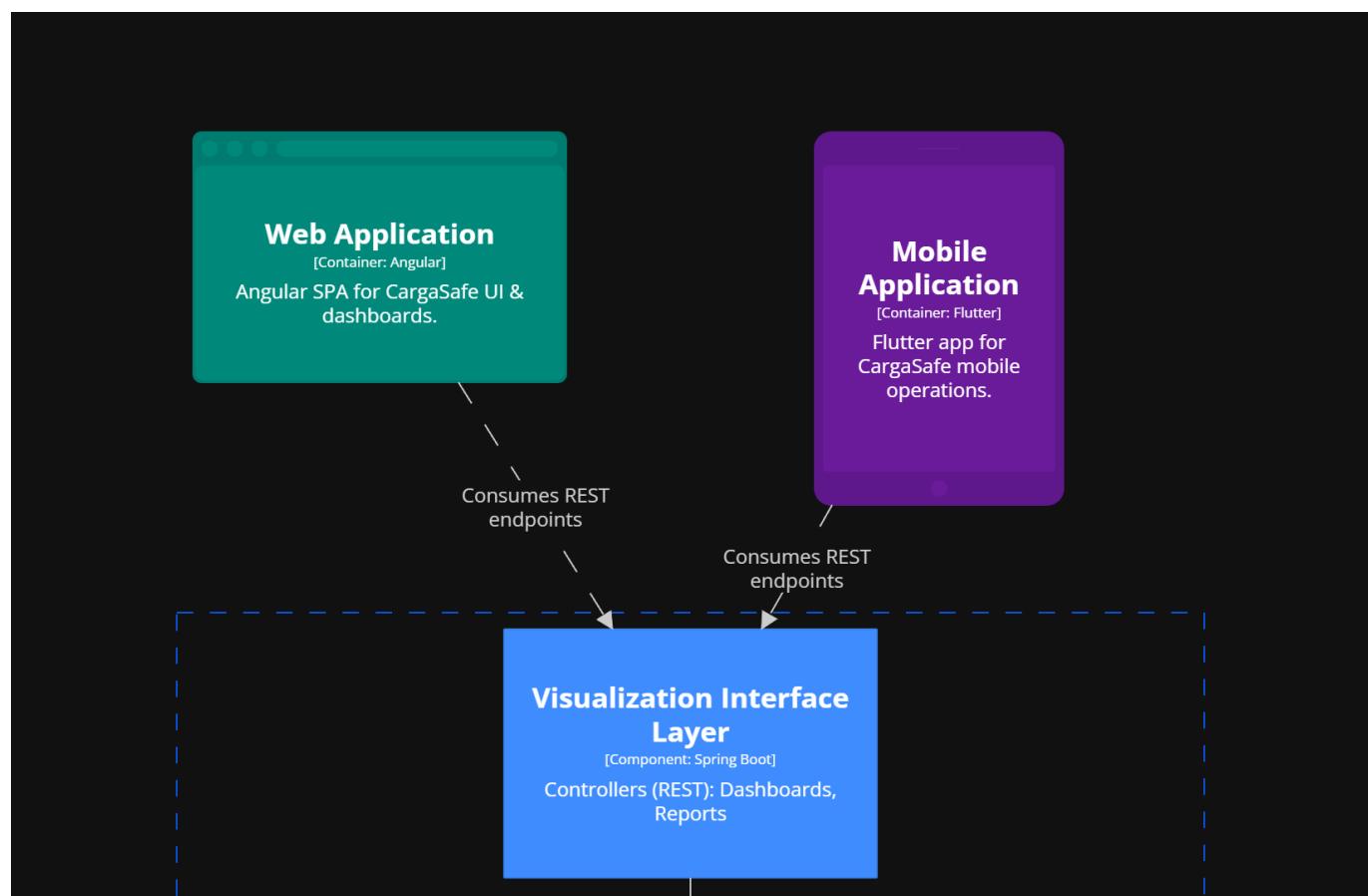
- Consultas optimizadas para métricas agregadas
- Conexión con base de datos de time-series
- Cache de métricas calculadas

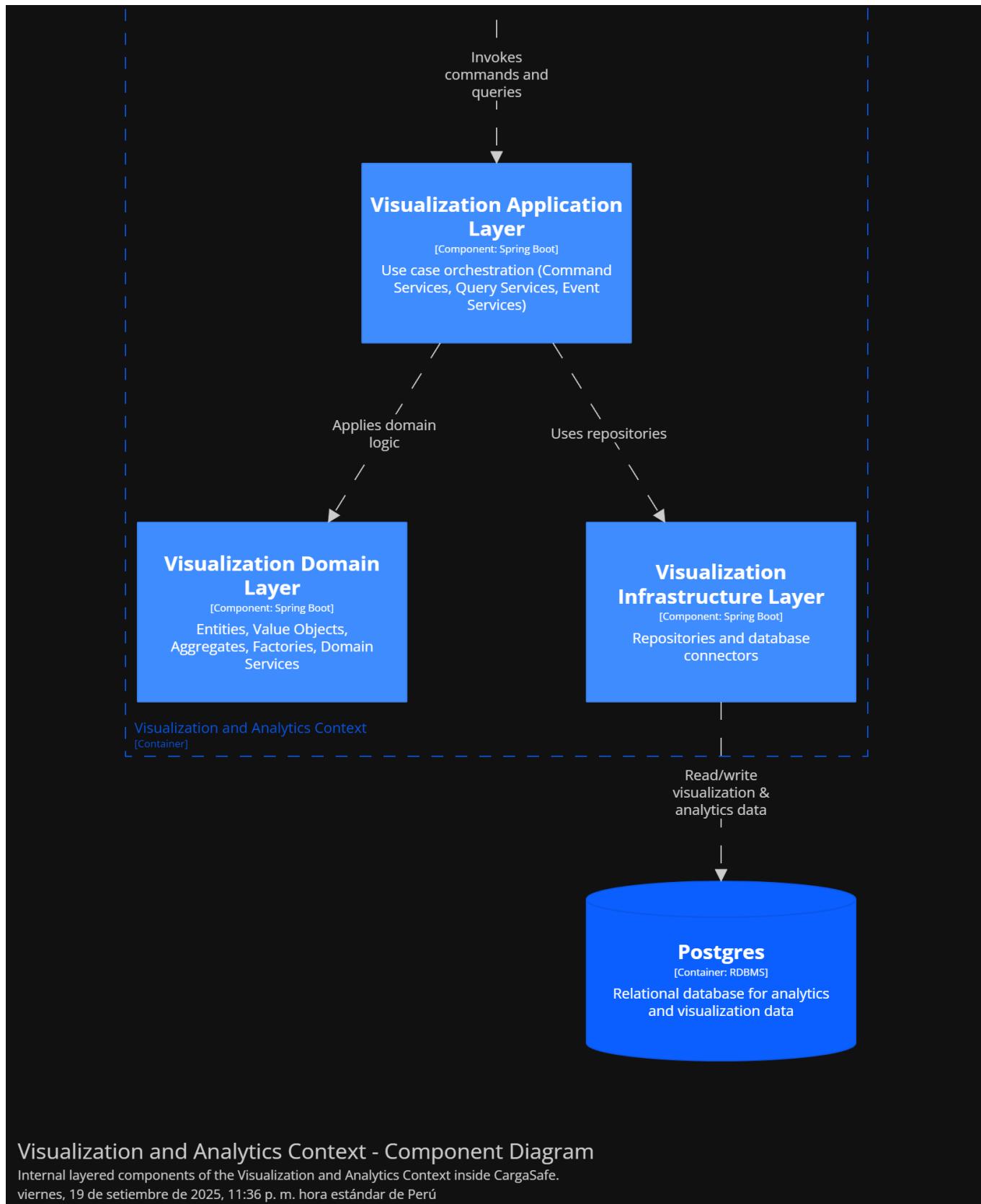
### **ChartDataRepository** (implementa IChartDataRepository)

- Transformación de datos para visualización
- Consultas optimizadas para gráficos
- Manejo de grandes volúmenes de datos temporales

#### 4.2.8.5. Bounded Context Software Architecture Component Level Diagrams

##### Diagrama de Componentes - Backend - Visualization Analytics





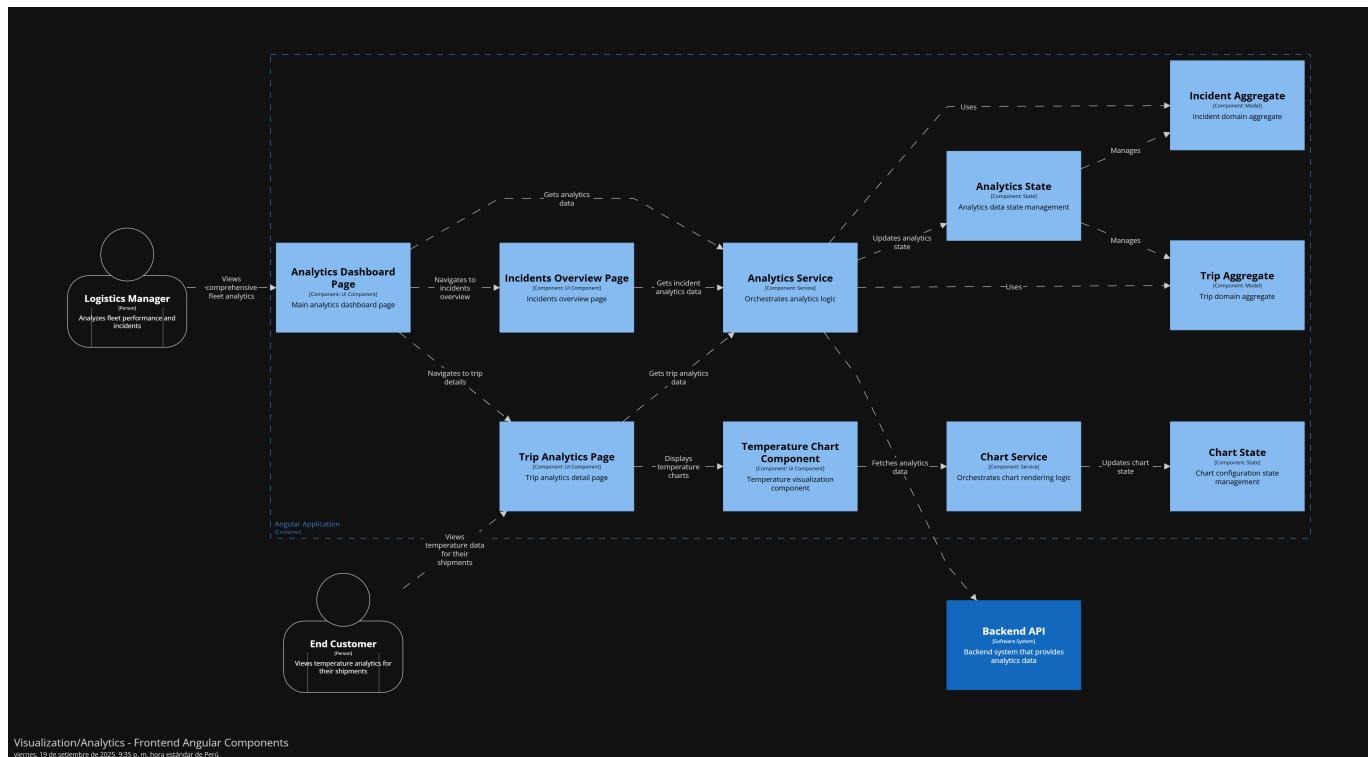
## Visualization and Analytics Context - Component Diagram

Internal layered components of the Visualization and Analytics Context inside CargaSafe.

viernes, 19 de setiembre de 2025, 11:36 p. m. hora estándar de Perú

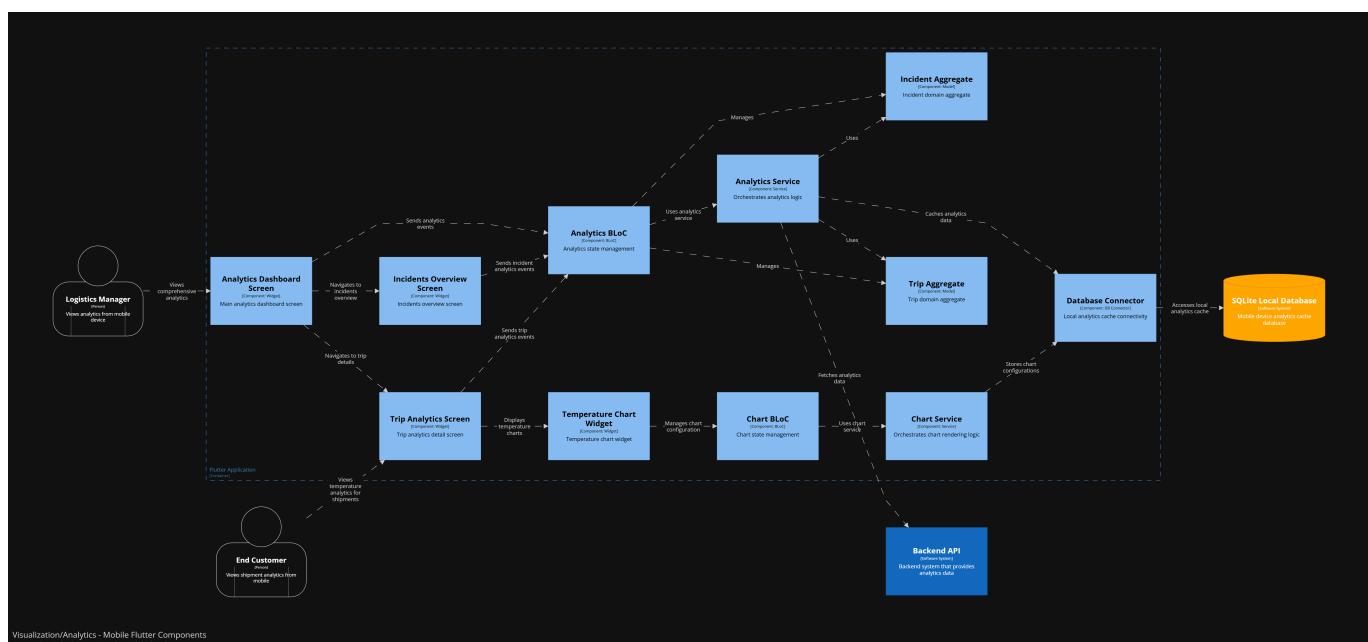
Este diagrama ilustra la arquitectura del bounded context de Visualization Analytics en el backend. Los controllers manejan requests relacionados con dashboards, reportes y análisis. Los services en Application Layer coordinan la lógica de negocio, mientras que los repositories optimizan el acceso a datos tanto transaccionales como de time-series para métricas y visualizaciones.

## Diagrama de Componentes - Frontend Web - Visualization Analytics



El frontend web del módulo de analytics utiliza componentes especializados para visualización de datos. Los chart components renderizarán gráficos interactivos, mientras que dashboard components gestionarán la composición y layout de widgets. Los services manejan la comunicación con APIs de datos y el cache local de métricas.

### Diagrama de Componentes - Mobile - Visualization Analytics

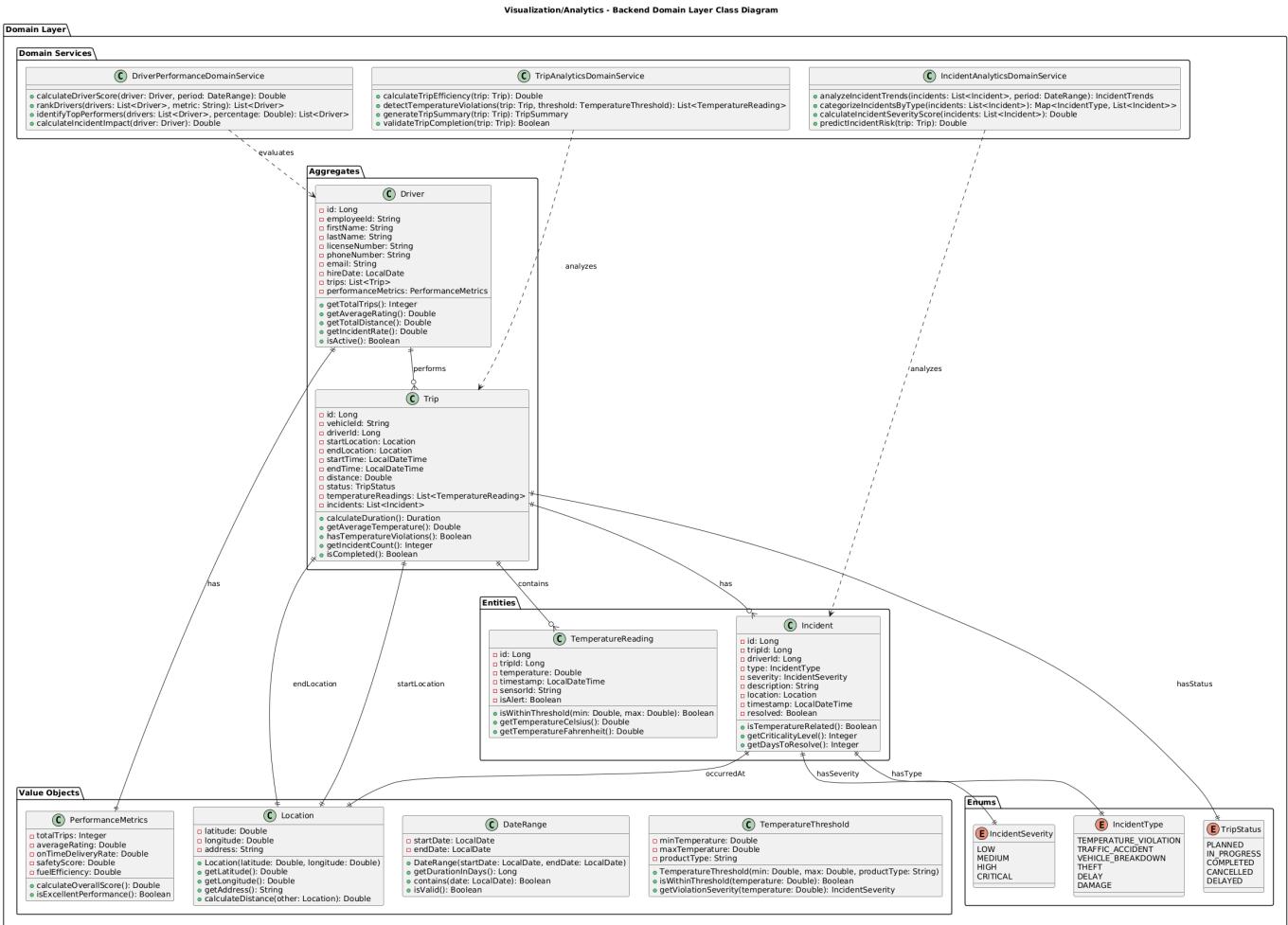


La aplicación móvil prioriza visualizaciones optimizadas para pantallas pequeñas. Los componentes incluyen widgets responsivos y gráficos touch-friendly. El state management a través de BLoC coordina la actualización de datos en tiempo real y gestiona el cache local para funcionalidad offline.

#### 4.2.8.6. Bounded Context Software Architecture Code Level Diagrams

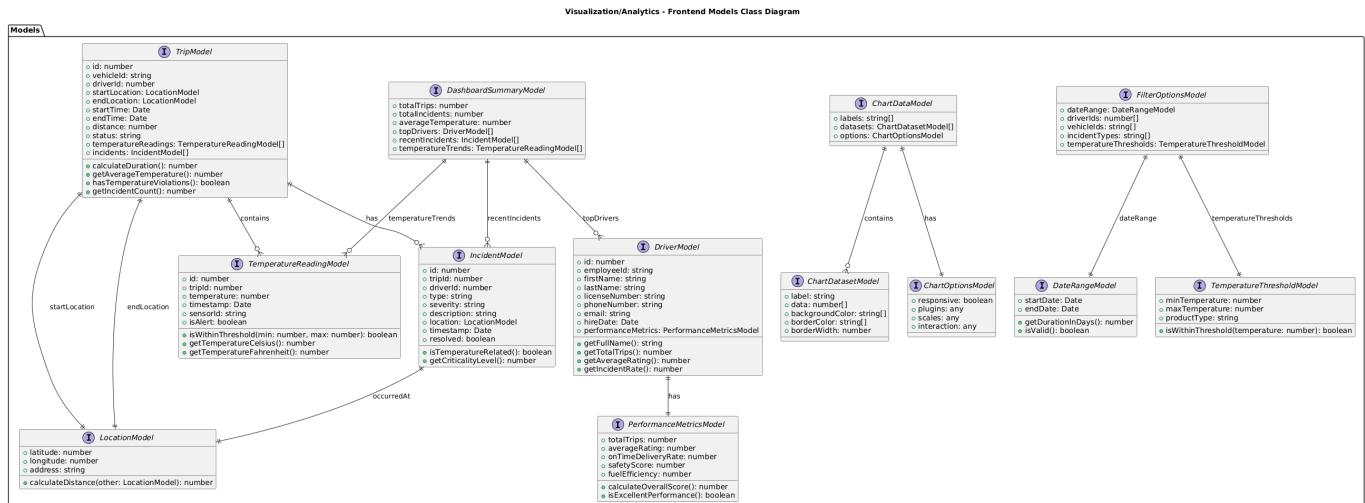
##### 4.2.8.6.1. Bounded Context Domain Layer Class Diagrams

## Backend - Visualization Analytics Domain Layer Class Diagram



El diagrama de clases del backend de Analytics muestra las entidades principales para visualización y análisis de datos. Dashboard actúa como aggregate root conteniendo múltiples Widgets. Los Reports están asociados a usuarios y pueden ser programados para generación automática. ChartData encapsula la información procesada para visualizaciones, mientras que los services coordinan la agregación y cálculo de métricas.

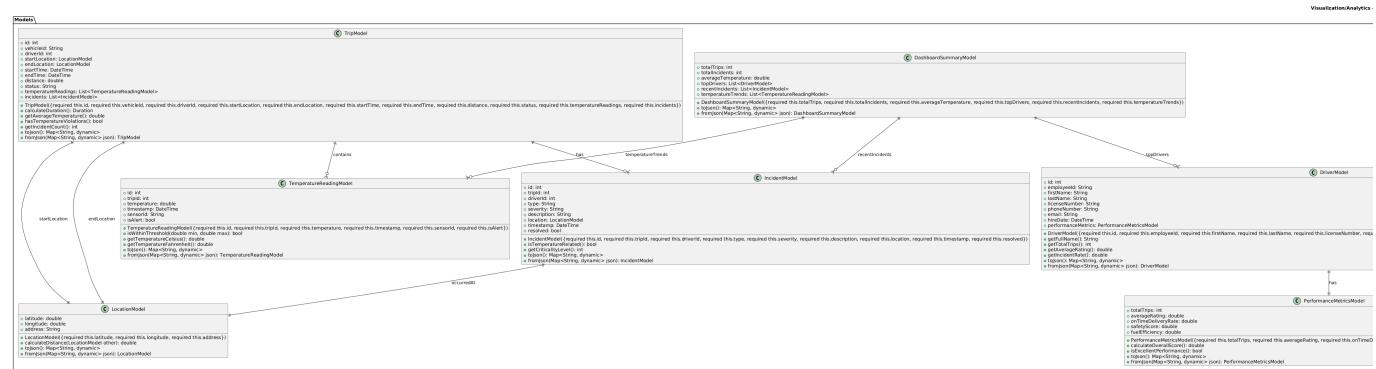
## Frontend - Visualization Analytics Domain Layer Class Diagram



El diagrama del frontend Angular muestra los componentes especializados para visualización de datos. Los chart components renderizan gráficos interactivos usando librerías como Chart.js o D3.js, mientras que

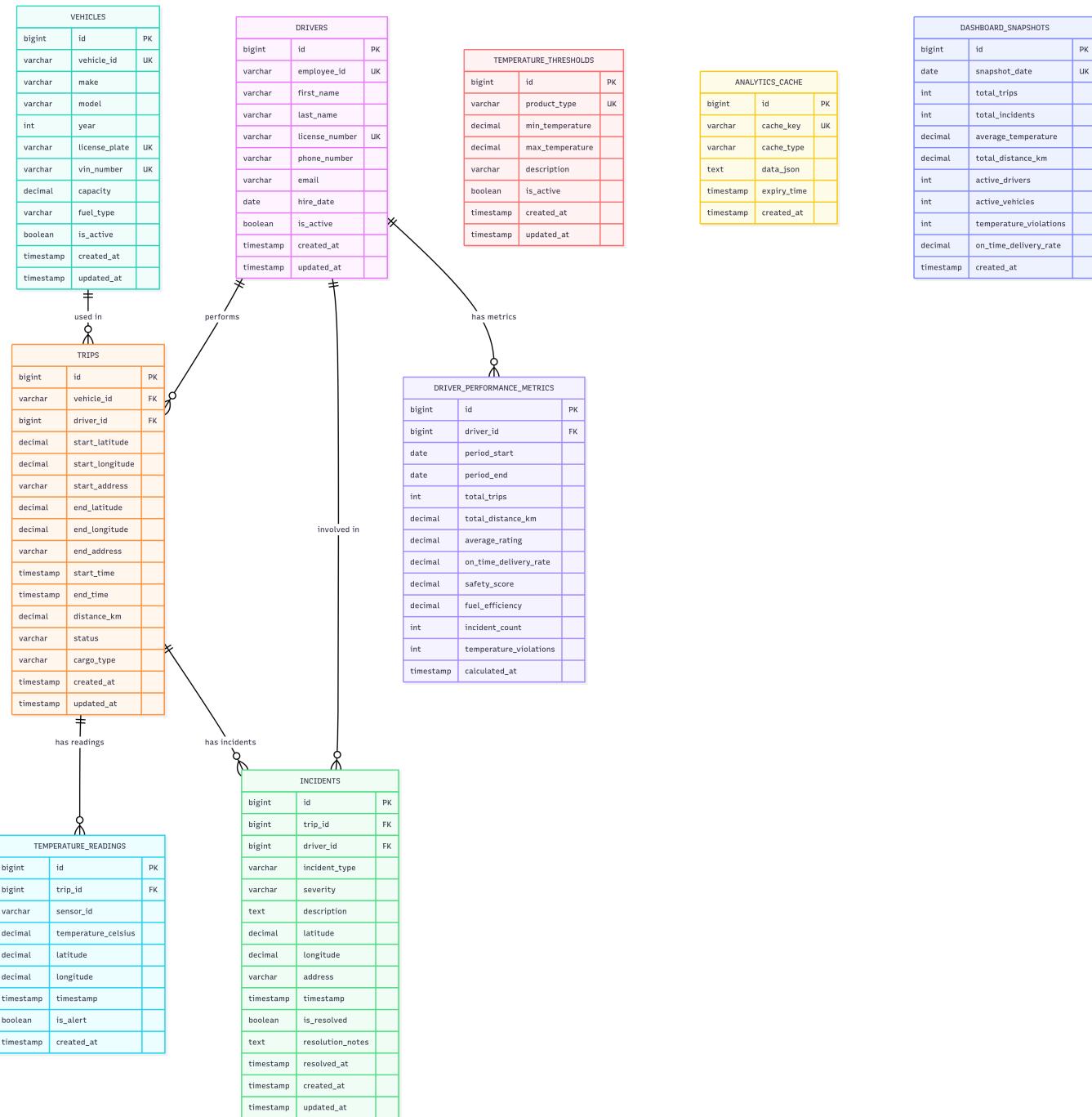
dashboard components gestionan la composición y layout de widgets. Los services manejan la comunicación con APIs de datos y el cache local de métricas para optimizar rendimiento.

## Mobile - Visualization Analytics Domain Layer Class Diagram



La aplicación móvil Flutter prioriza visualizaciones optimizadas para pantallas pequeñas. Los components incluyen widgets responsivos y gráficos touch-friendly. El state management através de BLoC coordina la actualización de datos en tiempo real y gestiona el cache local para funcionalidad offline, permitiendo consulta de métricas básicas sin conectividad.

### 4.2.8.6.2. Bounded Context Database Design Diagram



El diseño de base de datos del módulo Analytics está optimizado para consultas analíticas y agregaciones. Las tablas principales (DASHBOARDS, WIDGETS, REPORTS) mantienen configuraciones de usuario, mientras que las tablas de métricas están desnormalizadas para consultas rápidas. Se incluyen índices especializados para consultas temporales y agregaciones frecuentes.

## Bibliografía

Bogdanov, V. (2024, 23 octubre). *Real-Time Supply Chain Visibility: a Game-Changer*. rinf.tech. <https://www.rinf.tech/real-time-supply-chain-visibility-a-game-changer/>

Flock Freight | 2023 F&B Research Study. (s. f.). <https://www.flockfreight.com/2023-food-beverage-research-study>

Perfectplanner. (2025, 30 enero). *Enhancing Supply Chain Visibility through Real-Time Tracking Technologies*. Perfect Planner. <https://perfectplanner.io/enhancing-supply-chain-visibility/>

Technavio. (2024, 27 marzo). Cold Chain Logistics Market For Pharmaceuticals Industry size is set to grow by USD 12.81 bn from 2024-2028, Agility Public Warehousing Co. K.S.C.P, Air Canada & AVINEX, and more to emerge as Some of the Key Vendors, Technavio. PR Newswire. <https://www.prnewswire.com/news-releases/cold-chain-logistics-market-for-pharmaceuticals-industry-size-is-set-to-grow-by-usd-12-81-bn-from-2024-2028--agility-public-warehousing-co-kscp-air-canada--avinex-and-more-to-emerge-as-some-of-the-key-vendors-technavio-302099252.html>