

Softwarinos - Report



Universidad Peruana de Ciencias Aplicadas

INGENIERÍA DE SOFTWARE

Ciclo: 8

CURSO: Arquitecturas De Software Emergentes | SECCIÓN 4253

Profesor: De Los Rios Fernandez, Christian Luis

Proyecto de curso

Informe de Trabajo Final

StartUp: Softwarinos

Producto: Detekto

Integrantes:

Integrantes	Codigo
Ampudia Flores, Jose Carlos Isaac	u202112936
De La Piedra Quintanilla, Erwin Miquel	U202112179
Elsner De La Torre Ugarte	u202111654

Integrantes	Codigo		
Gutierrez Zumaeta, Manuel Alonso	U202112353		
Ciclo 2025-10			
Mayo, 2025			
<hr/>			
<h2>Registro de Versiones del informe</h2>			
<hr/>			
Versión	Fecha	Autor	Descripción de modificación
0.1	19-04-2025	Manuel Gutierrez	Redacción de Student Outcome, Startup Profile, Descripción de la Startup y Perfiles del equipo.
0.2	21-04-2025	Jose Ampudia	Elaboración del Solution Profile: Antecedentes, Problemática, Lean UX Process (Problem Statements, Assumptions, Hypotheses, Canvas).
0.3	22-04-2025	Miquel De la Piedra	Desarrollo de Segmentos Objetivo, análisis de Competidores, estrategias y tácticas frente a la competencia.
0.4	23-04-2025	Julio Elsner	Diseño de entrevistas, registro de entrevistas y análisis de entrevistas.
0.5	24-04-2025	Manuel Gutierrez	Elaboración de Needfinding: User Personas, User Task Matrix, Empathy Mapping, As-is Scenario Mapping.
0.6	24-04-2025	Miquel De la Piedra	Creación de Ubiquitous Language.
0.7	25-04-2025	Julio Elsner	Desarrollo del To-Be Scenario Mapping, User Stories e Impact Mapping.
0.8	25-04-2025	Jose Ampudia	Construcción del Product Backlog y definición del Strategic-Level Attribute-Driven Design: Design Purpose, Inputs, Constraints, Architectural Drivers.
0.9	26-04-2025	Manuel Gutierrez	Desarrollo de EventStorming, Candidate Context Discovery, Domain Message Flows Modeling, Context Mapping.

Versión	Fecha	Autor	Descripción de modificación
1.0	26-04-2025	Jose Ampudia	Finalización de Software Architecture: Landscape Diagram, Context Level, Container Level y Deployment Diagrams.
1.1	30-04-2025	Julio Elsner, Manuel Gutierrez	Inicio de redacción del Capítulo V: Tactical-Level Software Design. Revisión de estructura por capas y definición inicial del contexto Object-Recognition.
1.2	02-05-2025	Julio Elsner, Jose Ampudia	Desarrollo de las secciones Domain Layer e Interface Layer del contexto Object-Recognition.
1.3	10-05-2025	Manuel Gutierrez, Jose Ampudia	Desarrollo de Application Layer e Infrastructure Layer del contexto Object-Recognition.
1.4	13-05-2025	Julio Elsner	Redacción del Bounded Context: NOTIFICATIONS. Inicio del Capítulo VI con Style Guidelines y General Style Guidelines.
1.5	16-05-2025	Miquel De la Piedra	Redacción de Web, Mobile & Devices Style Guidelines, y desarrollo de Labeling, Searching y SEO Tags Systems.
1.6	17-05-2025	Jose Ampudia	Avance en Navigation Systems, Landing Page Wireframe y Mock-up. Diseño de Applications Wireframes y Wireflow Diagrams.

Project Report Collaboration Insights

Para el desarrollo del proyecto, se ha utilizado la plataforma de GitHub para el control correcto de versiones y la colaboración de los integrantes del equipo. A continuación se presenta el link directo a la organización del equipo:

Link de organización Softwarinos:

<https://github.com/Emergentes-Softwarinos>

TB1

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

Integrante	Responsabilidad
José Ampudia	Desarrollo del diseño de las entrevistas, registro de entrevistas, Análisis de entrevistas, Needfinding, User Personas, User Task Matrix, Empathy Mapping, As-Is Scenario Mapping, Ubiquitous Language, To-Be Scenario Mapping y User Stories

Integrante	Responsabilidad
Manuel Gutierrez	Desarrollo de Solution Profile, Antecedentes y problemática, Lean UX Process, Lean UX Problem Statements, Lean UX Assumptions, Lean UX Hypothesis Statements, Lean UX Canvas, Segmentos objetivos, Competidores, Análisis competitivo, Estrategias y tácticas frente a competidores y Registro de entrevistas
Miquel De la Piedra	Desarrollo del Impact Mapping, Product Backlog, Strategic-Level Attribute-Driven Design, Design Purpose, Attribute-Driven Design Inputs, Primary Functionality (Primary User Stories), Quality attribute Scenarios, Constraints, Architectural Drivers Backlog, Architectural Design Decisions y Quality Attribute Scenario Refinements y Registro de entrevistas
Julio De la Torre	Desarrollo de Strategic-Level Domain-Driven Design, EventStorming, Candidate Context Discovery, Domain Message Flows Modeling, Bounded Context Canvases, Context Mapping y Software Architecture, Software Architecture System Landscape Diagram, Software Architecture Context Level Diagrams, Software Architecture Container Level Diagrams y Software Architecture Deployment Diagrams y Registro de entrevistas

TB2

Para la entrega de la TB2 se realizó una reunión donde se asignaron las responsabilidades a cada integrante del equipo. A continuación se muestra la siguiente tabla con los detalles:

Integrante	Responsabilidad
José Ampudia	Desarrollo de las secciones Domain Layer, Interface Layer, Application Layer e Infrastructure Layer del contexto Object-Recognition. Participación en diseño de Landing Page (wireframe y mock-up), Applications Wireframes y Wireflow Diagrams.
Manuel Gutierrez	Redacción inicial del Capítulo V: Tactical-Level Software Design. Participación en la definición del contexto Object-Recognition y desarrollo de Application Layer e Infrastructure Layer.
Miquel De la Piedra	Desarrollo de Style Guidelines para Web, Mobile & Devices. Creación de sistemas de etiquetado, búsqueda y SEO (Labeling Systems, Searching Systems y SEO Tags).
Julio Elsner	Revisión de estructura por capas y redacción inicial del Capítulo V. Desarrollo del Bounded Context: NOTIFICATIONS. Redacción de General Style Guidelines e inicio del Capítulo VI. Participación en Domain Layer e Interface Layer del contexto Object-Recognition.

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

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

- José Ampudia (@IsaacAmp24)
- Manuel Gutierrez (@ManuGZ)
- Miquel De la Piedra (@MiquelDlp)
- Julio De la Torre (@JulioElsnerDLTU)

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

Tabla de contenidos

- Student Outcome

Capítulo I: Introducción

- 1.1. Startup Profile
 - 1.1.1. Descripción de la Startup
 - 1.1.2. Perfiles de integrantes del equipo
- 1.2. Solution Profile
 - 1.2.1. Antecedentes y problemática
 - 1.2.2. Lean UX Process
 - 1.2.2.1. Lean UX Problem Statements
 - 1.2.2.2. Lean UX Assumptions
 - 1.2.2.3. Lean UX Hypothesis Statements
 - 1.2.2.4. Lean UX Canvas
- 1.3. Segmentos objetivo

Capítulo II: Requirements Elicitation & Analysis

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

Capítulo III: Requirements Specification

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

Capítulo IV: Strategic-Level Software Design

- 4.1. Strategic-Level Attribute-Driven Design
 - 4.1.1. Design Purpose
 - 4.1.2. Attribute-Driven Design Inputs

- 4.1.2.1. Primary Functionality (Primary User Stories)
- 4.1.2.2. Quality attribute Scenarios
- 4.1.2.3. Constraints
- 4.1.3. Architectural Drivers Backlog
- 4.1.4. Architectural Design Decisions
- 4.1.5. Quality Attribute Scenario Refinements
- 4.2. Strategic-Level Domain-Driven Design
 - 4.2.1. EventStorming
 - 4.2.2. Candidate Context Discovery
 - 4.2.3. Domain Message Flows Modeling
 - 4.2.4. Bounded Context Canvases
 - 4.2.5. Context Mapping
- 4.3. Software Architecture
 - 4.3.1. Software Architecture System Landscape Diagram
 - 4.3.2. Software Architecture Context Level Diagrams
 - 4.3.3. Software Architecture Container Level Diagrams
 - 4.3.4. Software Architecture Deployment Diagrams

Capítulo V: Tactical-Level Software Design

- 5.X. Bounded Context: Nombre del Contexto
 - 5.X.1. Domain Layer
 - 5.X.2. Interface Layer
 - 5.X.3. Application Layer
 - 5.X.4. Infrastructure Layer
 - 5.X.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.X.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.X.7.1. Domain Layer Class Diagrams
 - 5.X.7.2. Database Design Diagram

Capítulo VI: Solution UX Design

- 6.1. Style Guidelines
 - 6.1.1. General Style Guidelines
 - 6.1.2. Web, Mobile & Devices Style Guidelines
- 6.2. Information Architecture
 - 6.2.2. Labeling Systems
 - 6.2.3. Searching Systems
 - 6.2.4. SEO Tags and Meta Tags
 - 6.2.5. Navigation Systems
- 6.3. Landing Page UI Design
 - 6.3.1. Landing Page Wireframe
 - 6.3.2. Landing Page Mock-up
- 6.4. Applications UX/UI Design
 - 6.4.1. Applications Wireframes
 - 6.4.2. Applications Wireflow Diagrams
 - 6.4.3. Applications Mock-ups

- 6.4.4. Applications User Flow Diagrams
- 6.5. Applications Prototyping

Capítulo VII: Product Implementation, Validation & Deployment

- 7.1. Software Configuration Management
 - 7.1.1. Development Environment Configuration
 - 7.1.2. Source Code Management
 - 7.1.3. Code Style Guide & Conventions
 - 7.1.4. Deployment Configuration
- 7.2. Solution Implementation
 - 7.2.X. Sprint n
 - 7.2.X.1. Sprint Planning
 - 7.2.X.2. Sprint Backlog
 - 7.2.X.3. Development Evidence
 - 7.2.X.4. Testing Suite Evidence
 - 7.2.X.5. Execution Evidence
 - 7.2.X.6. Services Documentation
 - 7.2.X.7. Deployment Evidence
 - 7.2.X.8. Team Collaboration Insights
- 7.3. Validation Interviews
 - 7.3.1. Diseño de Entrevistas
 - 7.3.2. Registro de Entrevistas
 - 7.3.3. Evaluaciones según heurísticas
- 7.4. Video About-the-Product

Conclusiones

- Conclusiones y recomendaciones

Video About-the-Team

- Video del equipo

Bibliografía

- Referencias

Anexos

- Anexos

Student Outcome

ABET – EAC - Student Outcome 3: Capacidad de comunicarse efectivamente con un rango de audiencias.

Criterio específico	Acciones realizadas	Conclusiones
---------------------	---------------------	--------------

Comunicó oralmente con efectividad a diferentes rangos de audiencia	Acciones realizadas	Conclusiones
	<p>José Ampudia</p> <p>TB1</p> <p>Participó activamente en entrevistas con usuarios para la recopilación de información clave sobre necesidades reales. Durante estas sesiones, empleó un lenguaje claro y empático para asegurar la comprensión del propósito del proyecto, facilitando así una retroalimentación útil y precisa.</p> <p>TB2</p> <p>Explicó verbalmente la estructura y funcionalidad de las capas Domain, Interface, Application e Infrastructure del contexto Object-Recognition durante sesiones técnicas del equipo, logrando que todos comprendieran la relación entre cada componente y su rol en el sistema.</p>	<p>Las habilidades de comunicación oral del equipo permitieron un entendimiento claro y compartido del proyecto tanto a nivel técnico como no técnico, mejorando la colaboración con usuarios y entre los miembros del equipo.</p>
	<p>Manuel Gutierrez</p> <p>TB1</p> <p>Comunicó de manera clara la problemática y el enfoque estratégico del proyecto durante las entrevistas con usuarios y discusiones del equipo, asegurando que todos comprendieran la visión general del producto. Además, facilitó sesiones explicativas sobre la propuesta de valor frente a competidores.</p> <p>TB2</p> <p>Presentó en reuniones del equipo la redacción inicial del Capítulo V, articulando de forma clara cómo se definió el contexto Object-Recognition y las decisiones tomadas sobre la Application Layer e Infrastructure Layer, facilitando su validación por parte de todos los miembros.</p>	
	<p>Miquel De la Piedra</p> <p>TB1</p> <p>Explicó el propósito y funcionamiento del Impact Mapping a su equipo, así como las decisiones de diseño arquitectónico a nivel estratégico, asegurando que todos los miembros comprendieran la relación entre funcionalidades y metas del proyecto.</p> <p>TB2</p> <p>Expuso verbalmente las Style Guidelines diseñadas para plataformas Web y Mobile, detallando cómo</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>se aplican los sistemas de etiquetado, búsqueda y SEO en el diseño UX, y resolvió dudas técnicas del equipo para alinear la arquitectura de información con el estilo visual.</p> <p>Julio Elsner</p> <p>TB1</p> <p>Facilitó el entendimiento de conceptos técnicos complejos como el EventStorming, Candidate Context Discovery y la arquitectura del sistema mediante presentaciones orales y debates dentro del equipo, adaptando el lenguaje técnico según el nivel de conocimiento de la audiencia.</p> <p>TB2</p> <p>Expuso oralmente el diseño por capas del contexto NOTIFICATIONS, así como la justificación de la redacción inicial del Capítulo VI. Además, facilitó la comunicación entre los miembros para integrar su trabajo con la Domain e Interface Layer del contexto Object-Recognition.</p>	
Comunica por escrito con efectividad a diferentes rangos de audiencia	<p>José Ampudia</p> <p>TB1</p> <p>Redactó documentos como el análisis de entrevistas, User Personas, y escenarios As-Is/To-Be de forma estructurada y clara, facilitando la comprensión por parte del equipo y stakeholders del contexto del usuario y sus necesidades.</p> <p>TB2</p> <p>Documentó las capas Domain, Interface, Application e Infrastructure del contexto Object-Recognition con una redacción técnica clara. También aportó en los wireframes y wireflows de las aplicaciones, describiendo los flujos con enfoque funcional y técnico.</p> <p>Manuel Gutierrez</p> <p>TB1</p> <p>Elaboró entregables como el Lean UX Canvas, Problem Statements y el análisis competitivo con una redacción precisa y orientada a la toma de decisiones estratégicas. Su documentación fue esencial para definir la propuesta de valor del proyecto.</p>	<p>La calidad de los entregables escritos permitió transmitir efectivamente tanto el enfoque de usuario como las decisiones técnicas, mejorando la comunicación y coordinación entre todos los actores involucrados.</p>

Criterio específico	Acciones realizadas	Conclusiones
	TB2 Redactó el borrador estructurado del Capítulo V, describiendo con claridad el enfoque general del diseño táctico. También explicó por escrito las funcionalidades clave de las capas Application e Infrastructure, permitiendo una interpretación consistente por parte de todos.	
	Miquel De la Piedra	
	TB1 Documentó el diseño arquitectónico a través del Attribute-Driven Design y Backlog técnico, asegurando claridad en los lineamientos técnicos para los futuros desarrolladores y partes interesadas.	
	TB2 Escribió las guías de estilo visual y la arquitectura de información para Web y Mobile, incluyendo el desarrollo de los sistemas de etiquetado, búsqueda y SEO. Su redacción estructurada facilitó la implementación posterior por parte del equipo de desarrollo.	
	Julio Elsner	
	TB1 Generó documentación técnica detallada sobre los componentes de la arquitectura del sistema, diagramas y mapeos de contexto, facilitando así la comprensión integral del sistema por parte de audiencias tanto técnicas como no técnicas.	
	TB2 Redactó el contexto NOTIFICATIONS en el Capítulo V, definiendo sus capas y describiendo sus interacciones. También elaboró las secciones introductorias del Capítulo VI, estableciendo las bases conceptuales del diseño UX de forma clara y coherente.	

Capítulo I: Introducción

1.1. Startup Profile

En esta sección presentaremos la descripción del startup y los perfiles de los miembros del team.

1.1.1. Descripción de la Startup

En Softwarinos, estamos desarrollando una solución con respecto al reconocimiento de objetos a través de una aplicación móvil, para facilitar la búsqueda de productos y la venta de estos mismos por parte de vendedores, además de un seguimiento de estas ventas y proyecciones de parte del administrador.

Misión: Crear una aplicación móvil que reconozca objetos de forma rápida y precisa, para ayudar a los vendedores a encontrar y vender productos fácilmente, y permitir a los administradores llevar un mejor control de las ventas y hacer proyecciones para su negocio.

Visión: Convertirnos en una empresa destacada en soluciones tecnológicas para el comercio, facilitando el trabajo de vendedores y administradores a través de herramientas inteligentes que mejoran la gestión y el crecimiento de sus negocios.

1.1.2. Perfiles de integrantes del equipo

Foto	Nombre y Descripción
	<p>Nombre: Manuel Alonso Gutierrez Zumaeta Código: u202112353 Descripción: Soy Manuel Gutierrez, estudiante de la carrera de Ingeniería de Software, tengo 20 años. Tengo experiencia en el desarrollo de aplicaciones web, tanto frontend como backend, programación móvil en Flutter. Me considero una persona responsable y que se propone hacer un buen trabajo.</p>
	<p>Nombre: Julio Elsner De La Torre Ugarte Código: u202111654 Descripción: Estudiante de Ingeniería de software, tengo 21 años. Cuento con experiencia en desarrollo de aplicaciones Web Full Stack (Front, Back-end) y bases de datos. Me considero una persona indagadora y que hace las cosas con una idea plasmada antes de realizarla.</p>

Foto	Nombre y Descripción
	<p>Nombre: Erwin Miquel De la Piedra Quintanilla Código: u202112179 Descripción: Soy Miquel De la Piedra, tengo 21 años, estudiante de la carrera de Ingeniería de Software. Poseo conocimientos relacionados a la programación en C++, Python, HTML y CSS. Además poseo experiencia con el manejo de bases de datos. Considero que estos conocimientos ayudarán a la realización de este trabajo, además de mejorar nuestros métodos de organización y trabajo en equipo.</p>
	<p>Nombre: José Carlos Isaac Ampudia Flores Código: u202112936 Descripción: Soy estudiante de Ingeniería de Software, con sólida formación en desarrollo Backend utilizando frameworks como Spring Boot. Tambiénuento con conocimientos en el desarrollo Frontend, trabajando con herramientas como Angular. Me caracterizo por mi capacidad para trabajar bajo presión, así como por mi enfoque organizado y metódico, lo que me permite adaptarme correctamente a los desafíos del entorno tecnológico.</p>

1.2. Solution Profile

1.2.1 Antecedentes y problemática

- **What (¿Qué?)**

El problema principal es la dificultad que enfrentan ferreterías y comercios de productos físicos para

identificar, buscar y gestionar herramientas y materiales de forma rápida y precisa. Los sistemas actuales no están adaptados al sector y requieren conocimientos técnicos o procesos manuales lentos.

- **When (¿Cuándo?)**

Este problema ocurre diariamente en el momento de la atención al cliente, la búsqueda de productos en almacén, la actualización del inventario y el registro de ventas. Es especialmente crítico durante horas de alta demanda o cuando se requiere atención eficiente.

- **Where (¿Dónde?)**

Sucede en ferreterías, depósitos, puntos de venta físicos y almacenes, donde los empleados necesitan reconocer y registrar productos con rapidez, y los administradores requieren visibilidad clara sobre el movimiento de stock y ventas.

- **Who (¿Quién?)**

Afecta principalmente a los vendedores que deben identificar productos de forma rápida, a los clientes que experimentan demoras, y a los administradores que necesitan gestionar inventario y ventas sin sistemas complejos.

- **Why (¿Por qué?)**

Porque los sistemas actuales como Shopify, Zoho CRM o TradeGecko están diseñados para otros tipos de negocios, no cuentan con funciones de reconocimiento visual, y son demasiado generales o complejos para el día a día de una ferretería tradicional.

- **How (¿Cómo?)**

El problema se intenta resolver actualmente con métodos manuales (catálogos impresos, búsqueda visual, listas), aplicaciones genéricas o incluso sin ningún sistema, lo que lleva a errores, pérdida de tiempo y falta de control.

- **How much (¿Cuánto?)**

El impacto puede medirse en tiempo perdido (minutos por venta), errores en inventario, insatisfacción del cliente, y pérdida de ventas por no poder identificar o encontrar un producto a tiempo. A nivel económico, esto representa una baja en la eficiencia operativa y en la competitividad del negocio.

1.2.2 Lean UX Process.

1.2.2.1. Lean UX Problem Statements.

En las ferreterías, la gestión de productos y la atención al cliente siguen dependiendo en gran medida de métodos tradicionales como la búsqueda manual, el uso de catálogos físicos y sistemas poco especializados. Esta situación genera tiempos de espera prolongados, errores frecuentes en la identificación de productos, una experiencia deficiente para el cliente y una gestión de inventario ineficiente.

Aunque existen plataformas genéricas de ventas y CRM, estas no se adaptan a las particularidades del rubro ferretero. El problema de fondo radica en la falta de soluciones tecnológicas específicas que automatizan y agilicen tareas operativas críticas como la identificación de productos y el control del stock.

Para resolver esta brecha, Detekto propone una solución tecnológica innovadora basada en **tecnologías emergentes** como la **visión computacional**, la **inteligencia artificial** y la **realidad aumentada**. Nuestra aplicación móvil permitirá a los vendedores reconocer herramientas y materiales directamente con la cámara

del dispositivo, facilitando el registro de ventas, la gestión de inventario en tiempo real y la generación de reportes automatizados.

Además, Detekto mejorará la coordinación entre vendedores y administradores a través de una interfaz intuitiva, diseñada para entornos de trabajo con alta demanda, ruido y limitaciones técnicas. El enfoque se orienta a ferreterías pequeñas y medianas que buscan digitalizarse sin incurrir en altos costos o en procesos complejos de implementación.

Sabremos que hemos tenido éxito cuando se logre:

- Reducir significativamente los tiempos de atención al cliente.
- Aumentar la precisión del inventario registrado.
- Mejorar la satisfacción tanto del cliente como del personal.
- Observar un incremento en las ventas derivado de procesos más ágiles y eficientes.

1.2.2.2. Lean UX Assumptions.

- **Business Assumptions**

- Las ferreterías requieren una solución especializada que agilice la identificación de productos y mejore la eficiencia operativa, ya que los sistemas actuales son generalistas y no se adaptan a sus procesos específicos.
- Una aplicación móvil con reconocimiento visual basado en inteligencia artificial y visión computacional puede reducir significativamente los tiempos de atención al cliente y mejorar la precisión en el registro de ventas e inventario.
- El modelo de negocio sostenible será mediante suscripciones mensuales, con niveles diferenciados (básico y premium) que incluyan funcionalidades como reportes avanzados y soporte técnico.
- Las principales soluciones existentes (como Shopify o Zoho CRM) carecen de un enfoque especializado en ferreterías, lo que representa una oportunidad de diferenciación para Detekto.
- La propuesta de valor de Detekto reside en ofrecer una herramienta intuitiva, eficiente y adaptada al entorno ferretero, que aproveche tecnologías emergentes para resolver tareas repetitivas y propensas a error.

- **User Assumptions**

- **¿Quién es el usuario?**

Vendedores de ferreterías, encargados de atención al cliente, personal de almacén y administradores de pequeños y medianos negocios del rubro.

- **¿En qué parte de su trabajo encaja Detekto?**

Se utiliza en el punto de venta o durante la organización del inventario, donde el tiempo y la precisión son críticos. La app actúa como un asistente digital inmediato.

- **¿Qué problema resuelve y cómo?**

El usuario enfrenta demoras y errores al buscar productos manualmente. Detekto lo soluciona usando visión computacional para identificar herramientas y materiales en segundos, optimizando la atención y reduciendo el margen de error.

- **¿Cuándo y cómo se usa Detekto?**

Durante la jornada laboral, principalmente con smartphones, al interactuar con clientes o al organizar productos en estanterías y almacenes.

- **¿Qué características son esenciales para el usuario?**

Rapidez en el reconocimiento visual, registro automático de productos, generación de reportes, interfaz intuitiva y disponibilidad offline o con conectividad limitada.

- **¿Cómo debe comportarse la app?**

Debe ser robusta y confiable, con una experiencia de usuario optimizada para ambientes dinámicos: botones grandes, flujos de navegación simples, interfaz visual clara y tiempos de respuesta inmediatos.

- **User Benefits**

- **Reconocimiento automático de productos:**

Ahorra tiempo y elimina la necesidad de búsqueda manual en catálogos o bases de datos.

- **Mejora de la atención al cliente y las ventas:**

Permite ofrecer respuestas más rápidas, mejorando la experiencia y acelerando los procesos de compra.

- **Control eficiente para administradores:**

Acceso en tiempo real a métricas clave como productos más vendidos y stock disponible, facilitando la toma de decisiones.

- **Reducción de errores humanos:**

La automatización del reconocimiento disminuye las equivocaciones en el registro y facturación de productos.

- **Facilidad de adopción:**

Pensado para usuarios con bajo nivel de alfabetización digital, lo que garantiza una curva de aprendizaje baja.

1.2.2.3. Lean UX Hypothesis Statements.

- **Hipótesis 1**

Creemos que la experiencia de los vendedores mejorará significativamente si se integra reconocimiento automático de productos mediante la cámara del dispositivo móvil, basado en visión computacional.

Sabremos que hemos tenido éxito cuando el tiempo promedio de identificación de productos se reduzca al menos en un 50% durante el primer mes de uso.

- **Hipótesis 2**

Creemos que los administradores tomarán decisiones más efectivas si cuentan con un panel inteligente con reportes automatizados de ventas e inventario.

Sabremos que hemos tenido éxito cuando al menos el 70% de los administradores accedan al panel semanalmente y reporten mejoras concretas en la gestión de stock en los primeros tres meses.

- **Hipótesis 3**

Creemos que una interfaz limpia, visual y optimizada para usuarios con poca experiencia tecnológica facilitará la adopción de la app.

Sabremos que hemos tenido éxito cuando el 80% de los nuevos usuarios completen su primera venta en menos de 10 minutos durante su primer uso.

- **Hipótesis 4**

Creemos que una solución especializada como Detekto, centrada exclusivamente en el sector ferretero, será preferida frente a plataformas genéricas.

Sabremos que hemos tenido éxito cuando al menos el 60% de los usuarios prefieran Detekto sobre otras soluciones según los resultados de encuestas de satisfacción después del primer mes.

1.2.2.4. Lean UX Canvas.

Lean UX Canvas



1.3. Segmentos objetivo.

El estado actual de la gestión de productos en las ferreterías está centrado principalmente en métodos tradicionales, como catálogos físicos y búsqueda manual, lo que resulta en tiempos de espera largos, ineficiencia en la atención al cliente y una gestión de inventario poco precisa. Lo que los sistemas existentes no logran abordar es la integración efectiva de tecnología para agilizar la identificación de productos, mejorar la precisión en el registro de ventas e inventario, y optimizar el seguimiento de las operaciones comerciales.

Nuestro producto, Detekto, resolverá esta brecha mediante el desarrollo de una solución móvil innovadora que utiliza reconocimiento visual a través de la cámara del dispositivo. Nos enfocaremos en proporcionar una

plataforma que permita a los vendedores identificar productos rápidamente, registrar ventas, acceder a informes de inventario en tiempo real y obtener reportes automatizados. Además, simplificaremos la gestión del inventario y mejoraremos la comunicación entre vendedores y administradores, todo desde una interfaz sencilla y fácil de usar.

Nuestro enfoque inicial será dirigirnos a ferreterías y pequeños comercios especializados en herramientas y materiales, proporcionando una solución tecnológica accesible que optimice sus procesos. Sabremos que hemos tenido éxito cuando veamos una disminución significativa en los tiempos de atención al cliente, una mejora en la precisión del inventario y una mayor satisfacción tanto de empleados como de clientes, resultando en un aumento en las ventas y eficiencia operativa.

Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores.

2.1.1. Análisis competitivo.

Competitive Analysis Landscape

¿Por qué llevar a cabo este análisis?	El objetivo de este análisis es identificar las características de los competidores y encontrar maneras de diferenciarnos.				
Startup y Competidores		Detekto	TradeGecko	Zoho CRM	Shopify
Perfil	Overview	App móvil centrada en reconocimiento de herramientas de ferretería mediante cámara. Enfocada en facilitar búsqueda, venta y seguimiento de producto.	Plataforma de gestión de inventario para pequeñas empresas. Adquirido por QuickBooks.	CRM completo con automatización de ventas y análisis, enfocado en todo tipo de industria.	Plataforma de ecommerce para crear y gestionar tiendas online con herramientas integradas de ventas.
	Ventaja competitiva ¿Qué valor ofrece a los clientes?	Reconocimiento de productos en tiempo real con IA + enfoque específico en ferreterías. Integración con ventas y control.	Gestión de inventario multi-canal con buena interfaz, pero sin reconocimiento visual.	Gran capacidad de personalización y automatización, pero no especializado.	Alto poder de personalización en ecommerce, pero más generalista, sin reconocimiento visual.
Perfil de Mercado	Mercado	Ferreterías,	PYMES con	Empresas	Emprendedores,

Marketing	objetivo	vendedores de productos físicos, administradores de puntos de venta.	múltiples canales de venta.	medianas y grandes con procesos de ventas complejos.	pymes y empresas que venden en línea.
	Estrategias de marketing	Enfoque en contenido visual (demostraciones), alianzas con ferreterías, participación en ferias del sector.	Marketing B2B digital, webinars, presencia en comunidades de comercio electrónico.	Publicidad online, inbound marketing, fuerza de ventas directa.	Marketing digital masivo, influencers, SEO, redes sociales.
Perfil de producto	Productos & Servicios	Reconocimiento de objetos, gestión de ventas, seguimiento de stock, reportes para administradores.	Gestión de inventario, pedidos, reportes, integración con ecommerce.	CRM, automatización de marketing, analíticas, gestión de clientes.	Tienda online, pagos, envíos, herramientas de marketing y reportes.
	Precios & Costos	Modelo freemium con funcionalidades avanzadas por suscripción mensual (propuesta).	Planes desde \$39/mes a más de \$200, dependiendo de la funcionalidad.	Planes desde \$14/mes, escalables por necesidades.	Desde \$39/mes hasta planes empresariales. Costos adicionales por plugins.
	Canales de distribución (Web y/o Móvil)	App móvil (iOS/Android), versión web en desarrollo (futuro).	Web app, sin versión móvil independiente.	Web app y apps móviles.	Web app y apps móviles.
Análisis SWOT	Fortalezas	Enfoque especializado en ferreterías. Reconocimiento visual innovador. Facilidad de uso.	Sólido en inventario multi-canal. Buena integración con QuickBooks.	Potente CRM con automatizaciones. Altamente configurable.	Ecosistema completo de ecommerce. Alto reconocimiento de marca.
	Debilidades	Producto nuevo, aún en fase inicial. Menor reconocimiento visual. Puede ser complejo	No incluye funciones de reconocimiento visual. Puede ser complejo	No es específico para el sector ferretería. Curva de aprendizaje.	Competencia feroz, enfoque generalista. Costos ocultos

		para nuevos usuarios.		con apps externas.
Oportunidades	Asociaciones con proveedores de ferretería. Escalabilidad con inteligencia artificial.	Integrarse con nuevas plataformas de ecommerce.	Expandirse en sectores nicho como retail técnico.	Aumentar integraciones con IA y automatización visual.
Amenazas	Entrada de grandes plataformas al nicho. Adopción tecnológica baja en ciertos mercados.	Dependencia de QuickBooks. Competencia creciente.	Saturación del mercado CRM. Nuevos players con IA.	Altos costos de operación. Competencia con marketplaces como Amazon.

2.1.2. Estrategias y tácticas frente a competidores.

Detekto puede diferenciarse en el mercado ofreciendo una solución móvil especializada en el reconocimiento de herramientas y productos de ferretería mediante inteligencia artificial, un enfoque único frente a competidores más generalistas como TradeGecko, Zoho CRM y Shopify. Al centrarse en un mercado específico como el de las ferreterías y comercios de productos físicos, Detekto puede desarrollar funcionalidades para vendedores y administradores, como la identificación instantánea de productos, seguimiento de inventario y reportes automatizados de ventas.

A diferencia de plataformas como Shopify, que están orientadas a la creación de tiendas online generales, Detekto puede destacar por su simplicidad de uso, velocidad en el reconocimiento de objetos y utilidad directa en puntos de venta físicos. Frente a Zoho CRM, que apunta a la gestión de relaciones con clientes de múltiples industrias, Detekto puede posicionarse como una herramienta operativa y directa, que no requiere grandes configuraciones ni curvas de aprendizaje complejas.

Además, el desarrollo de capacidades avanzadas basadas en visión por computadora y el uso de datos para generar proyecciones y sugerencias de ventas permitirá que Detekto evolucione hacia una plataforma inteligente, útil tanto para vendedores como para administradores. Esta tecnología, poco explotada en el sector ferretero, representa una ventaja competitiva significativa.

Para enfrentar a TradeGecko, que ofrece sólidas funciones de gestión de inventario pero sin reconocimiento visual, Detekto puede aprovechar su facilidad de uso desde dispositivos móviles y su interfaz adaptada al trabajo en campo o mostrador. También puede incluir integraciones simples con sistemas contables o puntos de venta, cerrando así el ciclo completo de atención comercial.

Una estrategia de marketing centrada en demostrar el funcionamiento en tiempo real de la app, mediante videos y casos de éxito, junto a alianzas con cadenas de ferreterías, distribuidores de herramientas y cámaras de comercio local, puede permitir a Detekto ganar visibilidad y confianza. Además, capacitar a los usuarios con contenido sencillo y práctico ayudará a reducir barreras tecnológicas en sectores menos digitalizados.

2.2. Entrevistas.

2.2.1. Diseño de entrevistas.

Se han planteado varias interrogantes dirigidas a nuestro público objetivo, con el fin de recolectar información que nos proporcionen requisitos, lo que fortalecerá para el desarrollo de nuestra solución.

- Segmento 1: Vendedor
 - Preguntas generales (Vendedor):
 - ¿Cuál es tu nombre y edad?
 - ¿En qué distrito vives actualmente?
 - ¿Cuál es tu estado civil y con quién vives?
 - ¿Cuál es tu ocupación actual y cuántos años llevas trabajando como vendedor?
 - ¿Con qué frecuencia usas el celular o dispositivos móviles en tu trabajo?
 - Preguntas sobre necesidades (Vendedor)
 - ¿Cómo sueles buscar un producto en la tienda actualmente?
 - ¿Qué haces cuando un cliente no sabe el nombre de lo que necesita, pero lo describe o muestra una imagen?
 - ¿Cómo verificas si hay stock de un producto?
 - ¿Te sientes cómodo usando tecnología como apps móviles para trabajar? ¿Por qué?
 - Preguntas para profundizar (Vendedor)
 - ¿Te ha pasado que pierdes una venta por no encontrar rápido el producto?
 - ¿Usas alguna aplicación o sistema digital actualmente en tu trabajo? ¿Cuál?
 - ¿Qué tipo de celular usas? (marca/modelo)
 - ¿Qué tan frecuentemente accedes a redes sociales o apps desde el celular?
 - ¿Qué esperas de una app que te ayude en tu trabajo como vendedor?
 - ¿Qué es lo que más te frustra de tu día a día en el trabajo?
- Segmento 2: Administrador
 - Preguntas generales (Administrador):
 - ¿Cuál es su nombre y edad?
 - ¿Dónde vive actualmente?
 - ¿Cuál es su ocupación dentro del negocio? ¿Desde cuándo ocupa ese cargo?
 - ¿Qué nivel educativo tiene?
 - ¿Con qué frecuencia utiliza una computadora en su trabajo?
 - Preguntas sobre necesidades (Administrador)
 - ¿Cómo realiza actualmente el seguimiento de productos y stock en la tienda?
 - ¿Qué tipo de información le gustaría ver respecto a las ventas que realizan los vendedores?
 - ¿Cómo hace proyecciones o predicciones de ingresos o ganancias?
 - ¿Usa alguna herramienta digital (Excel, sistema contable, etc.) para la gestión del negocio?
 - ¿Qué tan cómodo se siente usando plataformas web o dashboards de datos?

- Preguntas para profundizar (Administrador)

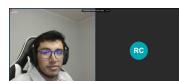
- ¿Qué tipo de decisiones toma a partir de la información de ventas?
- ¿Con qué frecuencia revisa el rendimiento de los vendedores?
- ¿Qué tipo de reportes o gráficos le gustaría ver en una plataforma web?
- ¿Qué le frustra más al momento de hacer seguimiento o gestión del negocio?
- ¿Cuáles son sus objetivos principales como administrador de la tienda?

2.2.2. Registro de entrevistas.

Entrevistas realizadas a Vendedores (Segmento 1):

Entrevista 1	Carlos Raul
Edad	23
Ocupación	Vendedor en una ferretería
Distrito	Villa El Salvador
Fecha	23/04/2025
<p>Carlos, un joven vendedor de 23 años que trabaja en una ferretería desde hace casi dos años, compartió su experiencia diaria atendiendo clientes. Actualmente, realiza búsquedas de productos de forma manual, lo que le toma tiempo, especialmente cuando el cliente no conoce el nombre del artículo. En esos casos, Carlos intenta adivinar a partir de fotos o descripciones, lo que muchas veces genera demoras o incluso pérdida de ventas. Destacó que una de sus principales frustraciones es no encontrar rápidamente los productos o confirmar su disponibilidad, ya que no cuentan con un sistema digital de stock. Carlos usa con frecuencia su celular durante el trabajo y se siente cómodo con la tecnología, por lo que considera muy útil una app con reconocimiento de objetos mediante cámara. Espera que esta herramienta le permita identificar productos rápidamente, verificar su precio, stock y ubicación dentro de la tienda para mejorar su atención y eficiencia.</p> 	
URL de la grabación	Ver grabación
Timming	0:00 - 5:57
Entrevista 2	Roberto Carranza
Edad	25
Ocupación	Vendedor de ferretería
Distrito	Jesús María
Fecha	24/02/2025

Entrevista 2 Roberto Carranza



Roberto es un joven vendedor en una ferretería. Actualmente, en su trabajo utilizan un software antiguo para buscar stock y precios, lo cual resulta lento y ha ocasionado pérdida de ventas. Una de sus principales frustraciones es cuando los clientes traen una muestra sin saber el nombre del producto; él tiene que adivinar el nombre y la marca, ya que el sistema solo muestra resultados si se ingresan exactamente esos datos. Roberto considera que una aplicación que reconozca los productos a partir de una muestra y muestre diferentes marcas con precios y stock disponibles le facilitaría mucho el proceso de venta.

URL de la grabación	Ver grabación
----------------------------	-------------------------------

Timming 5:56 - 13:07

Entrevista 3 Joaquin Rivadeneyra

Edad	30
-------------	----

Ocupación	Vendedor de ferretería
------------------	------------------------

Distrito	San Isidro
-----------------	------------

Fecha	25/02/2025
--------------	------------



La entrevista fue realizada a Joaquín Rivadeneyra, vendedor de una ferretería en San Isidro. Él nos comentó que, aunque actualmente manejan algunas operaciones básicas usando Excel, la actualización manual de la información sobre productos y ventas suele generar problemas, como retrasos y errores. Además, expresó su interés en poder visualizar las ventas de productos directamente desde su celular, de manera organizada por día, semana o mes, a través de una aplicación móvil. Joaquín mostró apertura e interés en adoptar nuevas herramientas digitales que le permitan enfocarse más en brindar una mejor atención al cliente y dejar de lado tareas manuales que podrían automatizarse.

URL de la grabación	Ver grabación
----------------------------	-------------------------------

Timming 13:08 - 14:59

Entrevistas realizadas a Administradores (Segmento 2):
Entrevista 4 Luna Doly

Edad	30
-------------	----

Ocupación	Administradora de una ferretería
------------------	----------------------------------

Distrito	Jesús María, Lima
-----------------	-------------------

Fecha	23/04/2025
--------------	------------

Entrevista 4**Luna Doly**

La entrevista se realizó con Luna, administradora de una ferretería en Lima con más de 7 años de experiencia en el cargo. Actualmente gestiona el negocio utilizando principalmente Excel y realiza el seguimiento del stock y las ventas de forma manual. Está interesada en contar con una plataforma web sencilla que le permita visualizar el rendimiento de los vendedores, los productos más vendidos y hacer proyecciones de ganancias. Aunque se siente cómoda usando computadoras, prefiere herramientas simples y visuales. Sus principales frustraciones incluyen la falta de información actualizada y errores en los registros. Como administradora, su objetivo es optimizar la gestión del negocio, crecer y mejorar la organización interna con ayuda de herramientas digitales.

URL de la grabación
[Ver grabación](#)
Timming

15:00 - 19:29

Entrevista 5**Sandro Alarcon****Edad**

22

Ocupación

Administradora de una ferretería

Distrito

San Juan de Lurigancho

Fecha

25/04/2025



La entrevista fue realizada a Sandro Alarcón, administrador de stock en una ferretería ubicada en Lima. Nos comentó que actualmente sigue utilizando métodos tradicionales como Excel para llevar el control de su inventario y reportes de productos. Sin embargo, señala que este proceso manual suele generar errores y retrasos, ya que depende mucho de la actualización constante y del cruce de información entre áreas. Sandro expresó su interés en adoptar nuevas herramientas digitales que puedan automatizar la gestión de stock y facilitar el trabajo de su equipo, considerando que una mejor organización beneficiaría directamente la eficiencia de las operaciones de la ferretería.

|| **URL de la grabación** | [Ver grabación](#) || **Timming** | 19:30 - 23:54 |

Entrevista 6**Oscar Palacios****Edad**

64

Ocupación

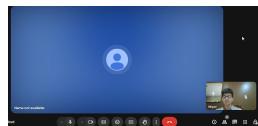
Administradora de una ferretería

Distrito

Ica, Ica

Fecha

25/04/2025

Entrevista 6**Oscar Palacios**

La entrevista fue realizada a Oscar Palacios, administrador de una ferretería en Ica, el nos comento como aun mantiene el uso tecnologías populares como Excel, para el manejo de su stock y sus ventas; sin embargo nos comenta también que esto lo mantiene con complicaciones, ya que es necesario realizar este procedimiento de manera manual y estar revisando constantemente los posibles cambios que puedan realizar, además de mantenerlo actualizado manualmente. Nos comenta también su gusto por utilizar nuevas herramientas que le ayuden con estos procesos y ayuden a sus trabajadores, ya que para él ellos son los principales pilares de su negocio.

URL de la grabación
[Ver grabación](#)
Timing

23:54 - 34:57

2.2.3. Análisis de entrevistas.

En base a las entrevistas realizadas, se ha llegado a la conclusión de 1 análisis para cada segmento objetivo.

Segmento 1: Vendedor

- Entrevista 1: Carlos Raul Carlos es un joven vendedor de 23 años que tiene casi dos años trabajando en una ferretería. Actualmente, se enfrenta a la frustración de buscar productos de forma manual, lo que genera demoras, especialmente cuando los clientes no saben el nombre del producto. Su principal desafío es encontrar rápidamente los productos y confirmar su disponibilidad, ya que no cuentan con un sistema digital de stock. Carlos se siente cómodo utilizando tecnología y considera útil una app con reconocimiento de objetos para mejorar la eficiencia en su trabajo. Esto le permitiría identificar productos, verificar precios y disponibilidad en tiempo real, mejorando la atención al cliente.
- Entrevista 2: Roberto Carranza Roberto, un vendedor de 25 años, utiliza un software antiguo para gestionar el stock y los precios de los productos, lo cual resulta ineficiente y le ha causado pérdidas de ventas. Uno de sus mayores problemas es cuando los clientes traen muestras sin saber el nombre del producto, ya que el sistema solo responde a búsquedas exactas. Roberto ve el valor de una aplicación que pueda reconocer productos a partir de una muestra y mostrar diferentes opciones con precios y stock disponible. Esto facilitaría su trabajo, permitiéndole ofrecer mejores opciones al cliente de manera más rápida.
- Entrevista 3: Joaquín Rivadeneyra Joaquín, un vendedor de 30 años, utiliza Excel para gestionar productos y ventas, pero esto le genera retrasos y errores debido a la actualización manual de la información. Él expresó su interés en una solución que le permita ver las ventas de productos desde su celular, organizadas por día, semana o mes. Está dispuesto a adoptar nuevas herramientas digitales que optimicen su tiempo y le permitan enfocarse más en brindar atención al cliente, delegando tareas administrativas a la tecnología. Esto reflejó una clara necesidad de herramientas que automatizan el proceso de visualización y gestión de ventas.

Segmento 2: Administrador

- Entrevista 4: Luna Doly Luna es una administradora de 30 años con más de 7 años de experiencia en el sector. Actualmente, gestiona el negocio utilizando Excel, lo que le genera frustraciones debido a la falta de información actualizada y errores en los registros. Luna busca una plataforma web sencilla que le permita visualizar el rendimiento de los vendedores, identificar los productos más vendidos y hacer proyecciones de ganancias. A pesar de que se siente cómoda usando computadoras, prefiere herramientas simples y visuales que faciliten la gestión de su negocio, mejoren la organización interna y optimicen los procesos.
- Entrevista 5: Sandro Alarcón Sandro, administrador de 22 años, utiliza Excel para llevar el control del inventario y los reportes de productos. Aunque se siente cómodo con esta herramienta, reconoce que el proceso manual puede generar errores y retrasos, lo que afecta la eficiencia operativa. Sandro está interesado en herramientas digitales que puedan automatizar la gestión del stock, mejorando la organización y eficiencia de las operaciones de la ferretería. Considera que una solución digital permitiría una mejor coordinación entre los equipos y una optimización de los procesos.
- Entrevista 6: Oscar Palacios Oscar, administrador de 64 años en Ica, también utiliza Excel para la gestión de stock y ventas, lo que le ha causado complicaciones debido a la necesidad de actualizar la información manualmente. Aunque sigue siendo dependiente de métodos tradicionales, está abierto a adoptar nuevas herramientas digitales que ayuden a automatizar los procesos y optimicen la gestión de la ferretería. Oscar ve a sus trabajadores como los pilares del negocio y considera que la tecnología podría facilitar su trabajo, mejorando la eficiencia en la operación.

2.3. Needfinding.

2.3.1. User Personas.

A continuación se presentan los perfiles de nuestros segmentos objetivos, tales como los Vendedores, y los Administradores de las ferreterías, esta información es primordial para entender las necesidades específicas y diseñar una herramienta que facilite la detección de objetos de ferretería y de esta manera aumentar sus ventas.

- Segmento 1: Vendedor:

NAME

Juan Fernando

MARKET SIZE



31 %

TYPE

Rational

Demographic

Male 34 years

Lima

Married

Vendedor en Ferretería

Skills

Adaptación: 75

Eficiencia: 80

Conocimiento técnico: 60

Channels

Facebook:

WhatsApp:

TikTok:

Goals

- Atender rápidamente a los clientes y cerrar más ventas.
- Identificar productos de forma sencilla, incluso si no conoce su nombre.
- Tener visibilidad clara del stock disponible para evitar confusiones.
- Mejorar la experiencia con el cliente para generar más recomendaciones y ventas rápidas.

Quote

La eficiencia y productividad de una empresa, es cuando la tecnología está a disposición.

Background

Juan tiene 34 años, trabaja como vendedor en una ferretería desde hace 6 años. No tiene formación técnica, pero ha aprendido sobre los productos en el día a día. Vive en Villa El Salvador con su esposa e hijos. Usa su smartphone principalmente para comunicarse por WhatsApp y ver redes sociales, pero no está muy familiarizado con apps especializadas. Su jornada es larga y suele lidiar con múltiples tareas al mismo tiempo: atender clientes, buscar productos, y revisar el inventario con su supervisor.

Motivations

- Ahorrar tiempo en la atención al cliente
- Evitar errores o confusiones que afecten en la venta
- Ser reconocido por su buen desempeño por parte de su jefe
- Sentirse más eficiente y moderno en su trabajo
- Usar herramientas que le faciliten el trabajo sin complicarlo

Frustrations

- No saber si hay stock hasta averiguar si hay en el almacén
- Clients que no saben el nombre del producto y se impacientan
- Tener que hacer muchas cosas al mismo tiempo sin apoyo tecnológico.
- Perder ventas por la falta de identificación de productos que Juan desconoce y no es capaz de recordar el nombre o encontrarlo rápido en el negocio

Technology



Browsers



UXPRESSIA

This persona was built in uxpressia.com

- Segmento 2: Administrador:

PERSONA: Alexandra

NAME Alexandra	MARKET SIZE  73 %	TYPE Guardian
	Goals <ul style="list-style-type: none"> Supervisar de forma efectiva el rendimiento de los vendedores Tener acceso rápido a reportes de ventas y productos Tomar decisiones informadas para mejorar las ganancias del negocio Tener una visión general del inventario para automatizar los datos y ahorrar tiempo y así evitar errores humanos 	
Demographic <p> Female 43 years</p> <p> Marital Status</p>	Quote <p><i>“Las buenas decisiones se toman mediante la correcta organización del equipo de trabajo”</i></p>	
Skills <p>Organización</p>  <p>Tecnología</p>  <p>Trabajo en equipo</p> 	Background <p>Alexandra tiene 42 años y es administradora de una ferretería familiar desde hace más de 10 años. Vive en San Juan de Lurigancho y tiene estudios técnicos en contabilidad. Está acostumbrada a usar Excel y WhatsApp Web en su día a día. Aunque no es experta en tecnología, sabe usar herramientas digitales básicas. Es muy organizada y meticolosa con los números. Le preocupa no tener visibilidad clara sobre lo que se vende y lo que está en stock, y a veces siente que los vendedores no reportan todo lo necesario.</p>	
Channels <p> WhatsApp  Google</p>	Motivations <ul style="list-style-type: none"> Tener el control total del inventario y las ventas. Ver estadísticas y proyecciones sin tener que calcular todo manualmente. Identificar qué productos se venden más y cuáles menos. Aumentar la rentabilidad del negocio mediante una mejor planificación. Reducir pérdidas por errores o desinformación. 	Frustrations <ul style="list-style-type: none"> No tener acceso a información de ventas en tiempo real. Tener que depender del vendedor para saber lo que se vendió. Usar herramientas que no son intuitivas o que requieren mucho tiempo. No poder proyectar ingresos fácilmente sin fórmulas complejas en Excel. Sentirse limitada por la falta de tecnología adaptada a su negocio.
Technology <p>  Apple Windows</p>	Browsers <p></p>	

UXPRESSIA

This persona was built in uxpressia.com

2.3.2. User Task Matrix.

En esta sección se detallan las tareas principales de los segmentos objetivos, en este caso, los vendedores y administradores de ferreterías, así como su frecuencia e importancia.

Tareas	Vendedores		Administradores	
	Frecuencia	Importancia	Frecuencia	Importancia
Buscar productos en el almacén o tienda	Diario	Alta	Quincenal	Media

Identificar productos a partir de imágenes o descripciones	Diario	Alta	Rara vez	Baja
Consultar stock	Diario	Alta	Semanal	Alta
Reservar productos para un cliente	Varias veces por semana	Alta	Ocasional	Media
Registrar ventas o informar sobre ventas realizadas	Diario	Alta	Semanal	Alta
Hacer seguimiento a ventas realizadas	Rara vez	Media	Semanal	Alta
Analizar productos más vendidos	Rara vez	Media	Semanal	Alta
Hacer proyecciones de ingresos	-	-	Quincenal	Alta
Interactuar con tecnología	Diario	Media	Diario	Alta

Las tareas que realizan en una empresa de ferretería son bastante similares a cualquier empresa de ventas, lo cual nos permite identificar las principales necesidades de nuestro público objetivo. A través de este análisis se pudo entender que, el vendedor realiza principalmente tareas operativas con una frecuencia diaria, relacionadas con la atención al cliente y búsqueda de productos. Por otro lado, la administradora desempeña un rol más estratégico y analítico, con tareas menos frecuentes pero de alta importancia, enfocadas a la supervisión de las ventas. Se destaca que ambos segmentos interactúan con herramientas digitales, aunque el enfoque y los propósitos son distintos.

2.3.3. Empathy Mapping.

Mediante este gráfico se evidencia las percepciones y sentimientos de los usuarios al interactuar con nuestra solución, este análisis nos permite conocer sobre sus desafíos que ellos poseen, y diseñar una propuesta más adecuada y centrada a sus requisitos.

Segmento 1 - Vendedor:

PERSONA: Juan Fernando

1.WHO are we empathizing with? Con Juan, un vendedor de ferretería con varios años de experiencia. Tiene trato directo con los clientes, trabaja bajo presión y busca rapidez y precisión al momento de atender. No es experto en tecnología, pero la usa en su día a día.	2.What do they need to DO? Atender rápidamente a los clientes. Identificar productos a partir de descripciones o imágenes. Consultar disponibilidad de stock. Reservar productos para concretar ventas. Registrar o reportar ventas realizadas. Usar tecnología que no interrumpta su flujo de trabajo.	3.What do they SAY? "A veces el cliente no sabe cómo se llama lo que quiere." "Pierdo tiempo buscando productos." "No sé si hay stock hasta que lo veo yo mismo." "Quisiera algo que me ayude a identificar productos más fácil." "Si la app es complicada, mejor no la uso."
4.What do they SEE? Clientes que llegan apurados o confundidos. Estantes con muchos productos similares. Compañeros que a veces también se frustran. Poca organización o herramientas digitales eficientes. Su celular como herramienta constante de apoyo.		5.What do they DO? Atiende clientes de manera directa. Busca productos físicamente. Usa WhatsApp o el celular para mostrar o recibir imágenes. Intenta resolver rápido, muchas veces improvisando. Pregunta por stock a compañeros o revisa en almacén.
6.What do they HEAR? Clientes: "No tienes esto?" "¿Puedes buscarme rápido?" Supervisores: "¿Ya vendiste eso?" Compañeros: "Siempre falta algo" o "Esto es un caos". Redes/ambiente digital: "Ahora todo es con apps y cámaras".	7.What do they THINK and FEEL? Se siente presionado por la velocidad de atención. Le frustra no tener herramientas que lo ayuden. Piensa que una buena app podría mejorar mucho su trabajo. Se siente inseguro si no encuentra rápido el producto. Tiene apertura al uso de tecnología, pero quiere que sea sencilla.	PAINS Pérdida de tiempo buscando productos. Clientes que se molestan por demoras. Dificultad para identificar productos sin nombre. Falta de visibilidad sobre stock. Herramientas digitales confusas o poco accesibles.
GAINS Identificación de productos desde imágenes o descripciones. Consulta rápida de stock desde el celular. Reservar productos sin posos complicados. Aumento en ventas gracias a mayor agilidad. Mayor seguridad y confianza en la atención al cliente.		

UXPRESSIA

This persona was built in uexpresia.com

Segmento 2 - Administrador:

PERSONA: Alexandra

1.WHO are we empathizing with? Alexandra, 45 años, es administradora en una ferretería mediana. Tiene experiencia en control de inventarios, seguimiento de ventas y supervisión de personal. Su mayor reto es mantener la operación organizada, tener acceso rápido a la información y tomar decisiones sin perder tiempo revisando registros manuales. Usa principalmente una laptop para trabajar, pero también revisa su celular.	2.What do they need to DO? Supervisar el desempeño de los vendedores. Visualizar en tiempo real el inventario disponible. Analizar las ventas por periodo y por producto. Detectar tendencias o problemas con los productos. Generar proyecciones y reportes automáticamente. Coordinar reabastecimiento de stock.	3.What do they SAY? "Necesito un sistema que me diga todo sin tener que buscar entre papeles." "No puedo tomar decisiones si la información no está clara." "Quiero tener control desde una plataforma sencilla." "La digital me interesa, pero debe ser fácil de usar."
4.What do they SEE? Sistemas genéricos y complejos que no se adaptan a su negocio. Pérdida de ventas por falta de organización. Información dispersa entre cuadernos, WhatsApp y Excel. Oportunidad de mejora si tuviera acceso a reportes en tiempo real. Vendedores improvisando cuando no tienen datos del stock.		5.What do they DO? Registra y revisa datos de ventas semanalmente. Actualiza inventarios manualmente o por WhatsApp. Supervisa las operaciones de forma remota. Revisa informes antes de tomar decisiones con proveedores. Se frustra cuando debe corregir errores de los vendedores.
6.What do they HEAR? De vendedores: "No sé si queda stock de ese producto." De clientes: "¿Cuándo responderán este artículo?" De otros negocios: "Ya usan apps que les ahorran tiempo." De su entorno: "Debes digitalizar más el negocio."	7.What do they THINK and FEEL? Quiere sentirse en control con acceso a información clave. Se preocupa por perder ventas por desorganización. Le genera ansiedad depender de registros manuales. Está motivada a usar tecnología, siempre que no complique su rutina. Confía en soluciones simples que estén pensadas para su rubro.	PAINS • Desorden en el manejo de inventario. • Falta de reportes automáticos y en tiempo real. • Depender de Excel, papel o WhatsApp para tomar decisiones. • Pérdida de tiempo y oportunidades por no tener información precisa. • Dificultad para visualizar el desempeño del negocio sin procesos centralizados.
GAINS • Plataforma web centralizada con reportes automáticos. • Acceso rápido al inventario y a ventas por vendedor. • Proyecciones de ingresos con solo un par de clics. • Interfaz amigable, sin curvas de aprendizaje técnicas. • Ahorro de tiempo y mejora en la toma de decisiones estratégicas.		

2.3.4. As-is Scenario Mapping.

En esta sección, se describen los procesos cotidianos, antes de la implementación de nuestra solución, esto nos permite identificar desafíos que sufren nuestro segmento objetivo, y a través de estos, encontrar oportunidades de mejora para mejorar el reconocimiento de objetos y mejorar las ventas en el sector ferretero.

- Segmento 1 - Vendedor:

Segmento Vendedor

Steps	Recibe al cliente	Identifica el producto	Verifica el stock	Procesa la venta	Actualiza el inventario
Doing	- Saluda al cliente - Pregunta qué necesita	Escucha la descripción o el uso del producto Trata de adivinar de qué herramienta se trata	Camina al almacén para revisar manualmente Consulta con compañeros si existe el stock que el cliente pidió	Calcula el precio con calculadora Anota el producto en una hoja	Normalmente lo deja para el final del día Muchas veces no registra las ventas por falta de herramientas
Thinking	“¿Será un cliente nuevo o de los que regatean?” “Ojalá sepa explicar lo que busca”	“¿Dónde he visto antes este producto?” “Capaz lo muestra una herramienta similar”	“No tengo idea si ese producto todavía queda en stock” “Ojalá no haya que decirle que no hay la herramienta que busca”	“Ojalá no se me esté olvidando nada” “Aver si el precio de esta herramienta está actualizada”	“Mejor lo anoto después, ahora no hay tiempo” “Ojalá no se me olvide”
Feeling	- Amable al inicio - Relajado si conoce al cliente	Frustrado si el cliente no se explica bien Ansioso si hay fila esperando	Presionado si no encuentra lo que busca Estresado si es hora punta y tiene muchos clientes	Enfocado en anotar lo que pidió el cliente Con algo de prisa si es hora punta	Desmotivado si ha sido un día pesado Preocupado por no llevar un buen control del stock

- Segmento 2 - Administrador:

Segmento Administrador

Steps	Revisión de ventas del día	Monitores del inventario	Organiza reportes y proyecciones	Toma decisiones operativas	Coordina con el personal
Doing	Pide al vendedor el resumen en papel o verbal Anota en su cuaderno, a veres en Excel	Revisa físicamente el almacén Pregunta a vendedores si hay agotamientos en el stock	Usa fórmulas básicas en Excel Imprime tablas	Evalúa si debe pedir más stock Decide si sube o baja los precios	Da instrucciones claras. Escucha sugerencias o quejas si hay tiempo.
Thinking	“Las ventas han sido buenas hoy?” “¿Faltará registrar algo?”	“¿Cuántas unidades quedaban de este producto?” “Deberían anotar bien los ingresos y saldos”	“¿Esto será suficiente para planear el próximo pedido?” “Ojalá Excel no se congeste”	“¿Vale la pena invertir en más stock ahora?” “¿Cómo justifico los gastos si no tengo datos claros?”	“¿Me habrán entendido bien?” “Sería mejor si todos siguieran un sistema.”
Feeling	Frustración leve si los números no cuadran Duda	Cansancio Preocupación por pérdidas no registradas	Inseguridad Deseo de automatizar pero falta de herramientas simples	Tensión al tomar decisiones con datos incompletos Terror a equivocarse.	Agradecida si el equipo responde bien. Deseosa de más organización

2.4. Ubiquitous Language.

En esta sección, se definen términos clave para el dominio del negocio, basándose en el enfoque de Ubiquitous Language propuesto por Eric Evans en Domain-Driven Design. Estas definiciones buscan asegurar una comunicación clara y efectiva entre los miembros del equipo, alineando el vocabulario técnico con las necesidades del proyecto.

- Product (Producto):** Herramienta o material físico que se encuentra en venta dentro de la ferretería.
- Stock (Stock/Inventario):** Cantidad disponible de cada producto en el almacén o tienda.
- Sale (Venta):** Transacción realizada entre un vendedor y un cliente por uno o más productos.
- Seller (Vendedor):** Persona encargada de atender al cliente, identificar los productos y concretar las ventas.

- **Administrator (Administrador):** Persona responsable de supervisar el negocio, controlar las ventas y monitorear el inventario.
- **Recognition (Reconocimiento visual):** Proceso por el cual la app identifica un producto mediante la cámara del celular.
- **Reservation (Reserva):** Acción de apartar un producto para un cliente hasta que se concrete su compra.
- **Report (Reporte):** Documento o visualización con datos organizados sobre ventas, productos o inventario.
- **Projection (Proyección):** Estimación de ventas o ingresos futuros basada en el análisis de datos pasados.
- **Unavailable Product (Producto no disponible):** Producto que no se encuentra en stock al momento de la búsqueda.
- **Best Seller (Producto más vendido):** Producto que ha tenido mayor número de ventas en un período determinado.
- **Break in Stock (Quiebre de stock):** Situación en la que se agota un producto y no se cuenta con reposición inmediata.
- **Storefront (Mostrador):** Zona de atención al cliente donde el vendedor interactúa directamente y realiza las ventas.
- **Warehouse (Almacén):** Espacio físico donde se guardan los productos no exhibidos.
- **Shift (Turno):** Período asignado de trabajo para los vendedores en el día.
- **Sales Summary (Resumen de ventas):** Informe breve que muestra el total de ventas realizadas en un día o período específico.
- **Restock (Reabastecimiento):** Acción de reponer productos que se han agotado en el inventario.
- **Visual Search (Búsqueda visual):** Método para encontrar un producto a través de una imagen o la cámara, en lugar de una búsqueda por nombre.

Capítulo III: Requirements Specification

3.1. To-Be Scenario Mapping.

En esta sección, se visualiza el rendimiento esperado de la solución propuesta después de su implementación, se presentarán los flujos de trabajo, las interacciones y las mejoras con respecto al estado actual, resaltando cómo la aplicación propuesta resolverá problemas y optimizará los procesos existentes.

Segmento 1: Vendedor:

Segmento Vendedor

Steps	Inicio del turno	Atención al cliente	Consulta de Stock	Venta y registro	Reserva de productos	Fin del turno				
Doing	Abre la app Detekto Revisa stock y avisos	Escanea herramientas que el cliente muestra o describe Atiende más rápido al tener sugerencias visuales en pantalla	Verifica disponibilidad directamente desde la app con un par de toques	Filtrar por categoría, nombre o imagen escaneada	Realiza el registro de venta desde la app (cantidad, producto, cliente)	Marca producto como vendido en inventario automáticamente	Usa la opción "reservar" tras identificar que el cliente necesita un producto que está por agotarse	Registra datos del cliente y fecha de retiro desde el mismo flujo	Accede al resumen del día con ventas, productos más pedidos y clientes atendidos	Se desconecta tranquilo sabiendo que todo quedó registrado
Thinking	"Vemos qué está en baja de stock."	"Qué bueno tener todo centralizado en una app."	"Este cliente no sabe el nombre, pero la cámara lo reconoce al instante."	"Puedo sugerirle algo mejor con la lista de similares."	"Ya no tengo que ir al almacén solo para revisar si hay o no stock."	"Incluso puedo ver sugerencias de reemplazo si falta el producto."	"Todo se guarda automáticamente, no tengo que depender de papeles."	"Puedo consultar luego cualquier detalle si hay un reclamo."	"No perderé esta venta, el cliente se siente contento sabiendo que lo tendrá reservado."	"Antes era un caos al final del día... ahora todo es más fluido, qué falta y qué debí reporar mañana."
Feeling	Preparado	Motivado	Seguro	Confiado en la herramienta	Tranquilo	Menos estresado	Se siente actualizado tecnológicamente	Ordenado	Responsable	Satisfecho

Segmento 2: Administrador:

Segmento Administrador

Steps	Inicio del día	Supervisión de ventas	Gestión del inventario	Análisis y decisiones	Cierre de jornada	
Doing	Abre la app Detekto y revisa el dashboard con resumen del día anterior.	Revisa notificaciones sobre bajo stock o tareas pendientes.	Compara rendimiento entre vendedores.	Revisa alertas de productos con poca rotación.	Genera reportes automáticos de ventas y stock.	Revisa que todo esté en orden.
Thinking	"Tengo todo al alcance sin pedir reportes."	"Ya sé qué áreas necesitan atención."	"Sé qué vendedor está teniendo más impacto."	"Puedo corregir en el momento si algo va mal."	"Gestionar inventario ya no es una pesadilla."	"Estoy actuando basado en hechos, no en coronadas."
Feeling	Organizado	Tranquilo	Seguro	Con visión clara del negocio	Eficiente	Empoderado

3.2. User Stories.

En esta sección, se presentan las User Stories que rescatan las necesidades principales de nuestros usuarios finales, tanto vendedores como los administradores. Esto nos ayuda a definir y a priorizar funcionalidades críticas para asegurar que la aplicación cumpla con las expectativas del usuario.

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
EP-01	Gestión de cuenta	Como usuario, quiero gestionar mi cuenta, incluyendo la creación de mi perfil y configuración de preferencias y personalizar mi perfil.	-	-
EP-02	Gestión de productos mediante escaneo visual	Como vendedor, quiero usar la cámara para reconocer productos, verificar stock, hacer reservas y revisar escaneos recientes, para agilizar mis tareas.	-	-
EP-03	Monitoreo de vendedores en tiempo real	Como administrador, quiero visualizar en tiempo real la ubicación de los vendedores y su actividad para supervisar mejor su desempeño y tomar decisiones oportunas.	-	-
EP-04	Control de asistencia de vendedores	Como administrador quiero tener un registro del horario de entrada y salida laboral de los vendedores	-	-

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-01	Registrar cuenta	Como usuario, quiero crear una cuenta para poder acceder a la aplicación.	<p>Escenario 1: Proceso de registro</p> <p>Dado que el usuario desea crear una cuenta, cuando complete el formulario de registro en la aplicación, entonces la cuenta deberá ser creada y el usuario recibirá una confirmación.</p> <p>Escenario 2: Verificación de cuenta</p> <p>Dado que el usuario ha registrado una cuenta, cuando la cuenta se cree, entonces el usuario deberá verificar su correo electrónico para activar la cuenta y acceder a la aplicación.</p>	EP-01
HU-02	Iniciar sesión	Como usuario, quiero ingresar a mi cuenta para utilizar la aplicación.	<p>Escenario 1: Acceso a la cuenta</p> <p>Dado que el usuario desea ingresar a la aplicación, cuando ingrese sus credenciales correctas, entonces deberá tener acceso a su cuenta y a todas las funcionalidades.</p> <p>Escenario 2: Mensaje de error.</p> <p>Dado que el usuario ha ingresado credenciales incorrectas, cuando intente iniciar sesión, entonces deberá recibir un mensaje de error y la opción de intentar nuevamente.</p>	EP-01
HU-03	Cerrar sesión	Como usuario quiero cerrar mi sesión para mantener mis datos seguros.	<p>Escenario 1: Cierre de sesión</p> <p>Dado que el usuario desea cerrar su sesión, cuando seleccione la opción de cerrar sesión en la aplicación, entonces su sesión deberá finalizar y será redirigido a la pantalla de inicio.</p> <p>Escenario 2: Confirmación de cierre.</p> <p>Dado que el usuario ha cerrado su sesión, cuando el proceso se complete, entonces deberá recibir una confirmación de que ha cerrado sesión exitosamente.</p>	EP-01

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-04	Recuperar contraseña	Como usuario, quiero recuperar mi contraseña en caso de olvidarla para poder acceder a mi cuenta.	<p>Escenario 1: Solicitar recuperación.</p> <p>Dado que el usuario ha olvidado su contraseña, cuando seleccione la opción de recuperación de contraseña o proporcione su dirección de correo electrónico, entonces recibirá un enlace para restablecer la contraseña.</p> <p>Escenario 2: Restablecer contraseña.</p> <p>Dado que el usuario ha recibido el enlace de recuperación, cuando siga el enlace y complete el formulario de restablecimiento de contraseña, entonces la contraseña deberá actualizarse y el usuario recibirá una confirmación de que la contraseña ha sido cambiada exitosamente.</p>	EP-01
HU-05	Escanear productos con la cámara	Como vendedor, quiero identificar productos escaneándolos con la cámara para ahorrar tiempo.	<p>Escenario 1: Producto reconocido correctamente</p> <p>Dado que el vendedor desea reconocer un producto, cuando lo escanee correctamente, entonces el sistema deberá mostrar su nombre, imagen, stock y precio.</p> <p>Escenario 2: Producto no reconocido</p> <p>Dado que el vendedor desea reconocer un producto, cuando el sistema no lo identifique, entonces deberá mostrar un mensaje de error y sugerir búsqueda manual.</p>	EP-02

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-06	Verificar disponibilidad en stock	Como vendedor, quiero conocer el stock de un producto escaneado para saber si puedo venderlo.	<p>Escenario 1: Mostrar stock actualizado.</p> <p>Dado que el vendedor desea verificar disponibilidad, cuando el sistema muestre el producto, entonces deberá indicar su cantidad en stock.</p> <p>Escenario 2: Considerar reservas activas.</p> <p>Dado que el vendedor desea información en tiempo real, cuando consulte el stock, entonces este deberá estar actualizado y reflejar reservas activas.</p>	EP-02
HU-07	Reservar producto escaneado	Como vendedor, quiero reservar productos para asegurar que estén disponibles para el cliente.	<p>Escenario 1: Crear reserva temporal</p> <p>Dado que el vendedor desea reservar un producto, cuando lo seleccione, entonces el sistema deberá marcarlo como reservado.</p> <p>Escenario 2: Liberar producto no vendido</p> <p>Dado que el vendedor desea mantener la reserva por un tiempo, cuando no se concrete la venta, entonces el producto deberá liberarse automáticamente.</p>	EP-02
HU-08	Ver historial de escaneos recientes.	Como vendedor, quiero ver los últimos productos escaneados para acceder a ellos sin repetir el proceso.	<p>Escenario 1: Acceso al historial visual.</p> <p>Dado que el vendedor desea revisar productos anteriores, cuando acceda al historial, entonces deberá ver los últimos 10 escaneos con nombre e imagen.</p> <p>Escenario 2: Acceso rápido a escaneo previo.</p> <p>Dado que el vendedor desea rapidez, cuando seleccione un producto del historial, entonces deberá cargarse directamente su información.</p>	EP-02

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-09	Registrar una venta	Como vendedor, quiero registrar una venta para llevar un control de las transacciones realizadas.	<p>Escenario 1: Registro exitoso de venta</p> <p>Dado que el usuario desea recibir notificaciones de riego, cuando el sistema complete un riego, entonces enviará una notificación con detalles sobre el riego realizado.</p> <p>Escenario 2: Validación antes del registro</p> <p>Dado que el vendedor desea registrar una venta, cuando el stock sea insuficiente o el producto no esté validado, entonces el sistema deberá impedir el registro y mostrar un mensaje.</p>	EP-02
HU-10	Generar recibo de venta	Como vendedor, quiero generar un recibo luego de registrar una venta para entregarlo al cliente como comprobante.	<p>Escenario 1: Emisión automática del recibo</p> <p>Dado que el vendedor ha registrado una venta, cuando el proceso finalice, entonces el sistema deberá generar un recibo con resumen de productos, precios y total.</p> <p>Escenario 2: Visualizar o reenviar recibo</p> <p>Dado que el vendedor desea compartir el recibo, cuando lo seleccione, entonces deberá poder visualizarlo, descargarlo o enviarlo por correo/WhatsApp.</p>	EP-02

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-11	Cancelar venta registrada recientemente	Como vendedor, quiero cancelar una venta reciente en caso de error, para corregir información sin afectar el inventario.	<p>Escenario 1: Cancelación dentro de límite de tiempo</p> <p>Dado que el vendedor desea cancelar una venta, cuando lo haga dentro de los 5 minutos posteriores al registro, entonces el sistema deberá anularla y actualizar el stock.</p> <p>Escenario 2: Restricción tras tiempo límite</p> <p>Dado que el vendedor desea cancelar una venta pasada, cuando haya excedido el tiempo permitido, entonces el sistema deberá bloquear la acción y sugerir contacto con el administrador.</p>	EP-02
HU-12	Revisar ventas realizadas	Como administrador, quiero revisar el historial de ventas realizadas por los vendedores para supervisar el rendimiento y tomar decisiones basadas en datos.	<p>Escenario 1: Consulta de ventas por fecha o vendedor.</p> <p>Dado que el administrador desea revisar el historial de ventas, cuando seleccione un rango de fechas o un vendedor específico, entonces el sistema deberá mostrar una lista con las ventas registradas, incluyendo fecha, productos y montos.</p> <p>Escenario 2: Acceso a detalles de cada venta</p> <p>Dado que el administrador desea analizar información específica, cuando seleccione una venta de la lista, entonces deberá poder ver el detalle completo: productos vendidos, cantidades, método de pago y total.</p>	EP-02

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-13	Ver ubicación de vendedores activos	Como administrador, quiero ver en el mapa la ubicación de los vendedores en tiempo real para supervisar sus rutas.	<p>Escenario 1: Seguimiento en vivo</p> <p>Dado que el administrador desea supervisar al equipo, cuando abra el panel de seguimiento, entonces el sistema deberá mostrar los vendedores con su última posición.</p> <p>Escenario 2: Actualización continua</p> <p>Dado que el administrador desea ver rutas actuales, cuando el vendedor cambie de ubicación, entonces el sistema deberá actualizar automáticamente el mapa.</p>	EP-03
HU-14	Recibir alertas por inactividad o desvíos.	Como administrador, quiero recibir alertas si un vendedor se detiene por mucho tiempo o se desvía de su ruta para poder actuar rápidamente.	<p>Escenario 1: Notificación por inactividad prolongada</p> <p>Dado que el administrador desea detectar problemas, cuando un vendedor no se mueva por más de 20 minutos, entonces el sistema deberá emitir una alerta.</p> <p>Escenario 2: Desvío de zona asignada</p> <p>Dado que el administrador desea asegurar el cumplimiento de zonas, cuando un vendedor salga del área establecida, entonces el sistema deberá notificar al administrador.</p>	EP-03
HU-15	Consultar historial de rutas por fecha	Como administrador, quiero consultar la ruta que siguió un vendedor en un día específico para analizar su productividad.	<p>Escenario 1: Selección de fecha y vendedor</p> <p>Dado que el administrador desea revisar rutas pasadas, cuando seleccione una fecha y vendedor, entonces el sistema deberá mostrar el trayecto registrado.</p> <p>Escenario 2: Detalles de movimiento</p> <p>Dado que el administrador desea analizar el comportamiento, cuando revise la ruta, entonces deberá ver puntos de inicio, paradas, tiempo total y distancia recorrida.</p>	EP-03

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-16	Registro de Entrada	Como vendedor, quiero registrar mi hora de entrada al trabajo para que se contabilice mi asistencia diaria.	<p>Escenario 1: Visualizar opción de marcar entrada</p> <p>Dado que el vendedor inicia su jornada laboral, cuando acceda a la sección de asistencia, entonces deberá ver un botón para marcar su entrada.</p> <p>Escenario 2: Registrar la entrada correctamente</p> <p>Dado que el vendedor presione "Marcar entrada", cuando se confirme el registro, entonces se deberá almacenar la fecha y hora exacta del ingreso.</p>	EP-04
HU-17	Registro de Salida	Como vendedor, quiero registrar mi hora de salida para completar mi jornada laboral.	<p>Escenario 1: Opción habilitada tras marcar entrada</p> <p>Dado que el vendedor ya marcó su entrada, cuando entre nuevamente a la sección de asistencia, entonces deberá ver habilitado el botón de "Marcar salida".</p> <p>Escenario 2: Registrar salida correctamente</p> <p>Dado que el vendedor presione "Marcar salida", cuando se confirme el registro, entonces se deberá almacenar la hora de salida y calcular el tiempo trabajado.</p>	EP-04

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-18	Consultar Historial de Asistencia	Como administrador, quiero consultar el historial de asistencia de los vendedores para hacer seguimiento de su cumplimiento.	<p>Escenario 1: Visualizar historial por vendedor</p> <p>Dado que el administrador accede al historial, cuando seleccione un vendedor y un rango de fechas, entonces el sistema deberá mostrar los registros de entrada y salida correspondientes.</p> <p>Escenario 2: Exportar historial</p> <p>Dado que el administrador ha generado una vista del historial, cuando seleccione "Exportar", entonces el sistema deberá permitir descargar el reporte en PDF o Excel.</p>	EP-04
HU-19	Recordatorio de Marcación	Como vendedor, quiero recibir una notificación si no he marcado entrada para no olvidar registrar mi asistencia.	<p>Escenario 1: Envío de recordatorio automático</p> <p>Dado que es un día laboral y no se ha registrado entrada hasta las 10:00 a.m., cuando el sistema detecte esta condición, entonces enviará una notificación push al dispositivo del vendedor.</p> <p>Escenario 2: No enviar en días no laborales</p> <p>Dado que el día está marcado como no laborable en el sistema, cuando no haya registro de entrada, entonces no se deberá enviar ninguna notificación.</p>	EP-04

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-20	Generar Reporte Mensual	Como administrador, quiero generar un reporte mensual de asistencia para evaluar el rendimiento personal.	<p>Escenario 1: Seleccionar mes y empleado</p> <p>Dado que el administrador accede al módulo de reportes, cuando seleccione un mes y un vendedor, entonces deberá visualizar el resumen de asistencia.</p> <p>Escenario 2: Descargar reporte</p> <p>Dado que el administrador visualiza el resumen mensual, cuando presione "Descargar", entonces el reporte deberá generarse en formato PDF o Excel.</p>	EP-04
HU-21	Visualización de reportes diarios de ventas	Como administrador, quiero ver un reporte de ventas diario para poder revisar el desempeño de las ventas del día y tomar decisiones sobre el stock.	<p>Escenario 1: Reporte de ventas diarias</p> <p>Dado que el administrador ha llegado al final del día, cuando accede al dashboard de reportes, entonces podrá ver un reporte detallado de las ventas del día.</p> <p>Escenario 2: Datos de ventas filtrados</p> <p>Dado que el administrador tiene el reporte de ventas diarias, cuando filtra las ventas por categoría o nombre, entonces podrá ver un reporte detallado de las ventas del día.</p>	EP-05

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-22	Visualización sobre sugerencias sobre compras	Como administrador, quiero recibir sugerencias de compras basadas en los reportes de ventas para tomar decisiones precisas.	<p>Escenario 1: Sugerencia sobre compras</p> <p>Dado que el administrador está visualizando el reporte de ventas, cuando hace clic en un producto específico en el reporte, entonces el sistema mostrará una sugerencia automática sobre el restock del producto.</p> <p>Escenario 2: Proyección de compras futuras</p> <p>Dado que el administrador está revisando el reporte de ventas, cuando de clic en "Proyecciones de compras futuras", entonces el sistema mostrará una proyección basada en las ventas pasadas.</p>	EP-05
HU-23	Visualizaciones de proyecciones mensuales o en un rango	Como administrador, quiero ver proyecciones mensuales de ventas y stock para poder planificar compras a largo plazo	<p>Escenario 1: Proyecciones mensuales de ventas.</p> <p>Dado que el administrador desea planificar compras, cuando accede al panel de proyecciones mensuales, entonces el sistema mostrará una proyección de ventas para los próximos 30 días o un rango basado en datos históricos de ventas.</p> <p>Escenario 2: Proyecciones del stock</p> <p>Dado que el administrador desea gestionar el stock, cuando visualiza las proyecciones de ventas, entonces, el sistema calculará y mostrará el stock necesario para cubrir las proyecciones de ventas del mes siguiente.</p>	EP-05

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
HU-24	Exportar reportes a formatos descargables	Como administrador, quiero exportar los reportes generados en formatos como PDF o Excel para poder analizarlos fuera de la aplicación.	<p>Escenario 1: Exportación en formato PDF.</p> <p>Dado que el administrador está viendo un reporte en la aplicación, cuando de clic al botón "Exportar en PDF", entonces el sistema generará y descarga el reporte en formato PDF.</p> <p>Escenario 2: Error al exportar en PDF</p> <p>Dado que el administrador está observando el reporte en la aplicación y se va el internet o sucede un error externo, cuando de clic a "Exportar en PDF", entonces la aplicación mostrará un error "Error al exportar PDF"</p>	EP-05
TS-01	Desarrollar API para el registro del usuario	Como Developer, quiero desarrollar una API RESTful que permita a los usuarios registrar sus cuentas en la aplicación.	<p>Escenario 1: Solicitar creación de cuenta</p> <p>Dado que el usuario proporciona su nombre, correo electrónico y contraseña en el formulario de registro, cuando la API recibe la solicitud de registro, entonces debe almacenar los datos en la base de datos y devolver un código de respuesta HTTP 201 (Creado), junto con un mensaje de éxito.</p> <p>Escenario 2: Error en la creación de cuenta</p> <p>Dado que el usuario proporciona datos incompletos o inválidos, cuando la API recibe la solicitud, entonces debe devolver un código de respuesta HTTP 400 (Bad Request), con un mensaje que indique los errores de validación (por ejemplo, "El correo electrónico ya está registrado").</p>	EP-01

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
TS-02	Desarrollar API para autenticación de usuario	Como Developer, quiero crear una API RESTful que valide las credenciales de los usuarios (correo electrónico y contraseña) para generar un token de autenticación y permitir el acceso a la aplicación.	<p>Escenario 1: Inicio de sesión exitoso.</p> <p>Dado que el usuario proporciona credenciales válidas (correo y contraseña), cuando la API recibe la solicitud de inicio de sesión, entonces debe devolver un código de respuesta HTTP 200, junto con el token de autenticación JWT.</p> <p>Escenario 2: Error en inicio de sesión</p> <p>Dado que el usuario ingresa credenciales incorrectas, cuando la API recibe la solicitud de inicio de sesión, entonces debe devolver un código de respuesta HTTP 401 (Unauthorized), con un mensaje "Credenciales incorrectas".</p>	EP-01
TS-03	Desarrollar API para escanear productos	Como Developer, quiero crear una API RESTful que reciba la imagen escaneada de un producto, lo procese y devuelva información relevante como el nombre, precio y stock del producto.	<p>Escenario 1: Escaneo exitoso</p> <p>Dado que el vendedor escanea el producto desde la cámara de su smartphone, cuando la API recibe la solicitud con el código de barras, entonces debe devolver un código de respuesta HTTP 200, con los datos del producto: nombre, precio, stock disponible.</p> <p>Escenario 2: Escaneo fallido</p> <p>Dado que el vendedor escanea el producto desde la cámara de su smartphone, cuando la API no encuentra el producto en la base de datos, entonces debe devolver un código de respuesta HTTP 404 (Not Found), con un mensaje "Producto no encontrado".</p>	EP-02

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
TS-04	Desarrollar API para registrar una venta	Como Developer, quiero crear una API RESTful que registre una venta, incluyendo los productos vendidos, cantidades, precios y el total de la transacción.	<p>Escenario 1: Registro de venta exitoso</p> <p>Dado que el vendedor proporciona los productos vendidos, sus cantidades y precios, cuando la API recibe la solicitud, entonces debe almacenar los datos de la venta en la base de datos y devolver un código de respuesta HTTP 201 (Creado), junto con los detalles de la venta (productos, cantidades, monto total).</p> <p>Escenario 2: Error en registro de venta debido a stock insuficiente</p> <p>Dado que el stock de algún producto es insuficiente, cuando la API recibe la solicitud de venta, entonces debe devolver un código de respuesta HTTP 400 (Bad Request), con el mensaje "Stock insuficiente para el producto [nombre del producto]".</p>	EP-02
TS-05	Desarrollar API para generar reportes de ventas	Como Developer, quiero desarrollar una API RESTful que permita generar reportes de ventas, filtrados por fecha, vendedor, o categoría de productos, y devolver los datos en formato JSON.	<p>Escenario 1: Solicitar reporte de ventas filtrado</p> <p>Dado que el administrador selecciona un rango de fechas y un vendedor, cuando la API recibe la solicitud, entonces debe devolver un código de respuesta HTTP 200, con un listado de ventas que incluyan productos, cantidades, fechas y montos totales.</p> <p>Escenario 2: Error al generar reporte</p> <p>Dado que el rango de fechas o los filtros proporcionados son inválidos, cuando la API procesa la solicitud, entonces debe devolver un código de respuesta HTTP 400, con el mensaje "Rango de fechas inválido".</p>	EP-05

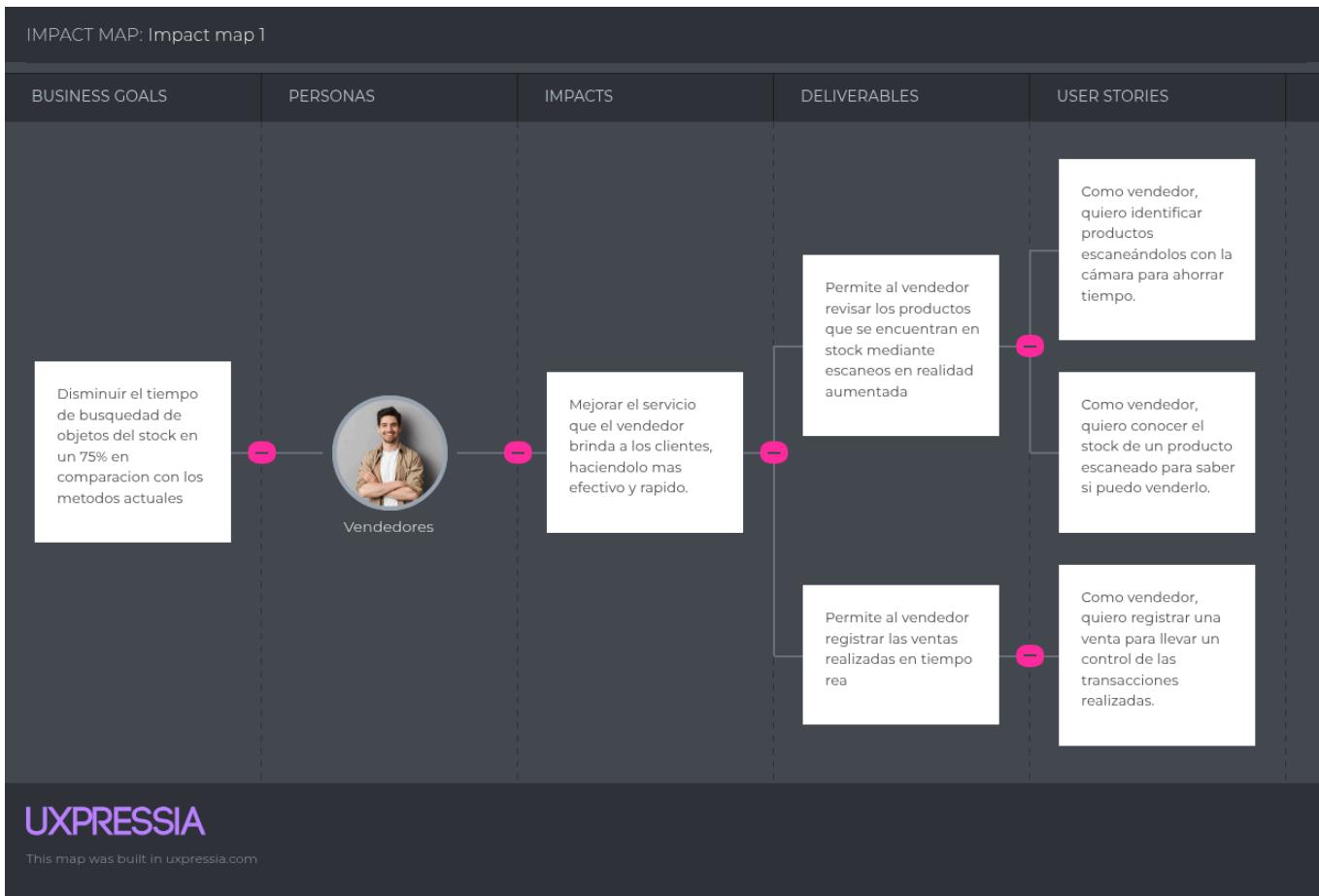
Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
TS-06	Desarrollar API para exportación de reportes	Como Developer, quiero crear una API RESTful que permita exportar los reportes generados en formatos como PDF o Excel, para que el administrador pueda descargarlos.	<p>Escenario 1: Exportación en formato PDF</p> <p>Dado que el administrador ha generado un reporte de ventas, cuando selecciona la opción de exportar en formato PDF, entonces la API debe generar el archivo PDF y devolverlo para su descarga con un código de respuesta HTTP 200.</p> <p>Escenario 2: Error al exportar reporte</p> <p>Dado que ocurre un problema técnico (por ejemplo, conexión a internet caída), cuando la API intenta generar el reporte, entonces debe devolver un código de respuesta HTTP 500 (Internal Server Error), con el mensaje "Error al exportar reporte".</p>	EP-05
TS-07	Desarrollar API para geolocalización de vendedores	Como Developer, quiero crear una API RESTful que reciba las coordenadas de los vendedores y las registre en la base de datos para que el administrador pueda ver las ubicaciones en tiempo real.	<p>Escenario 1: Enviar ubicación del vendedor</p> <p>Dado que el vendedor envía su ubicación actual (latitud y longitud), cuando la API recibe la solicitud, entonces debe actualizar la base de datos con la nueva ubicación y devolver un código de respuesta HTTP 200.</p> <p>Escenario 2: Error al enviar ubicación</p> <p>Dado que la ubicación enviada es inválida o está fuera de un rango permitido, cuando la API procesa la solicitud, entonces debe devolver un código de respuesta HTTP 400, con el mensaje "Ubicación inválida".</p>	EP-03

Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
TS-08	API para registrar entrada de asistencia	Como Developer, quiero desarrollar un endpoint que permita registrar la hora de entrada de un vendedor, para que el sistema almacene correctamente su inicio de jornada	<p>Escenario 1: Registrar entrada por primera vez en el día</p> <p>Dado que el vendedor ha iniciado sesión y no ha registrado entrada en la fecha actual, Cuando realiza un POST a /api/asistencia/entrada con su ID, Entonces el sistema debe guardar la fecha y hora actuales como hora de entrada Y retornar un mensaje de confirmación</p> <p>Escenario 2: Intentar registrar entrada más de una vez</p> <p>Dado que el vendedor ya ha registrado su entrada en la fecha actual, cuando realiza un nuevo POST a /api/asistencia/entrada Entonces el sistema debe retornar un mensaje de error indicando que la entrada ya fue registrada</p>	EP-04

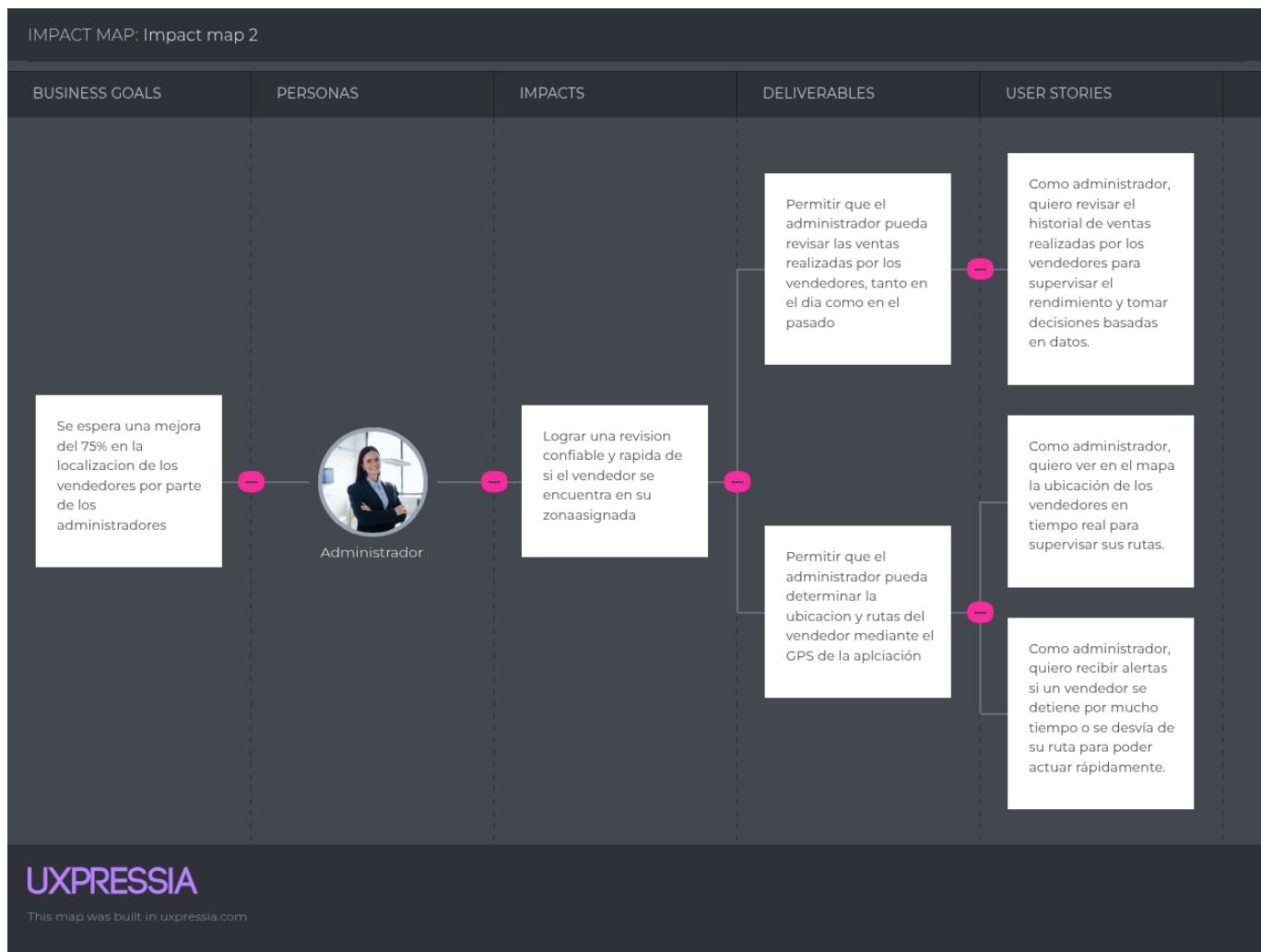
Epic/User Story ID	Título	Descripción	Criterios de Aceptación	Relación (EPIC ID)
TS-09	API para registrar salida de asistencia	Como Developer, quiero desarrollar un endpoint que registre la hora de salida del vendedor, para que se calcule automáticamente su tiempo trabajado.	<p>Escenario 1: Registrar salida correctamente</p> <p>Dado que el vendedor ya registró su entrada en la fecha actual y aún no ha registrado su salida, Cuando realiza un POST a /api/asistencia/salida con su ID Entonces el sistema debe guardar la hora actual como salida.</p> <p>Escenario 2: Intentar registrar salida sin haber registrado entrada</p> <p>Dado que el vendedor no ha registrado su entrada en la fecha actual, Cuando realiza un POST a /api/asistencia/salida Entonces el sistema debe retornar un mensaje de error indicando que no se puede registrar la salida sin una entrada previa.</p> <p>Escenario 3: Intentar registrar salida más de una vez</p> <p>Dado que el vendedor ya ha registrado su salida en la fecha actual, Cuando realiza otro POST a /api/asistencia/salida Entonces el sistema debe retornar un mensaje de error indicando que la salida ya fue registrada.</p>	EP-04

3.3. Impact Mapping.

- Vendedor



- Administrador



UXPRESSIA

This map was built in upressoia.com

3.4. Product Backlog.

Orden	User Story ID	Título	Descripción	Story Points
1	HU-01	Registrar cuenta	Como usuario, quiero crear una cuenta para poder acceder a la aplicación.	2
2	HU-02	Iniciar sesión	Como usuario, quiero ingresar a mi cuenta para utilizar la aplicación.	1
3	HU-05	Escanear productos con la cámara	Como vendedor, quiero identificar productos escaneándolos con la cámara para ahorrar tiempo.	5
4	HU-06	Verificar disponibilidad en stock	Como vendedor, quiero conocer el stock de un producto escaneado para saber si puedo venderlo.	3
5	HU-07	Reservar producto escaneado	Como vendedor, quiero reservar productos para asegurar su disponibilidad.	3
6	HU-08	Ver historial de escaneos recientes	Como vendedor, quiero ver los últimos productos escaneados para agilizar consultas.	2

Orden	User Story ID	Título	Descripción	Story Points
7	HU-09	Registrar una venta	Como vendedor, quiero registrar una venta para llevar control de las transacciones.	5
8	HU-10	Generar recibo de venta	Como vendedor, quiero generar un recibo luego de registrar una venta para el cliente.	3
9	HU-11	Cancelar venta registrada recientemente	Como vendedor, quiero cancelar una venta reciente en caso de error.	3
10	HU-16	Registro de Entrada	Como vendedor, quiero registrar mi hora de entrada al trabajo.	2
11	HU-17	Registro de Salida	Como vendedor, quiero registrar mi hora de salida para completar la jornada.	2
12	HU-19	Recordatorio de Marcación	Como vendedor, quiero recibir una notificación si olvido marcar entrada.	2
13	HU-03	Cerrar sesión	Como usuario, quiero cerrar sesión para mantener mis datos seguros.	1
14	HU-04	Recuperar Contraseña	Como usuario, quiero recuperar mi contraseña si la olvido.	2
15	HU-12	Revisar ventas realizadas	Como administrador, quiero revisar ventas realizadas por los vendedores.	5
16	HU-21	Visualización de reportes diarios de ventas	Como administrador, quiero ver reportes diarios de ventas para gestionar stock.	5
17	HU-22	Visualización sobre sugerencias de compras	Como administrador, quiero recibir sugerencias automáticas de compras.	3
18	HU-23	Visualización de proyecciones mensuales o rango	Como administrador, quiero ver proyecciones de ventas y stock.	3
19	HU-24	Exportar reportes a formatos descargables	Como administrador, quiero exportar los reportes en PDF o Excel.	2
20	HU-13	Ver ubicación de vendedores activos	Como administrador, quiero ver en el mapa la ubicación de vendedores.	5
21	HU-14	Recibir alertas por inactividad o desvíos	Como administrador, quiero recibir alertas si un vendedor está inactivo o desviado.	5

Orden	User Story ID	Título	Descripción	Story Points
22	HU-15	Consultar historial de rutas por fecha	Como administrador, quiero consultar rutas seguidas por los vendedores.	3
23	HU-18	Consultar Historial de Asistencia	Como administrador, quiero consultar el historial de asistencia de vendedores.	3
24	HU-20	Generar Reporte Mensual	Como administrador, quiero generar un reporte mensual de asistencia.	2
25	TS-01	Desarrollar API para registro de usuario	Como Developer, quiero desarrollar una API que permita registrar cuentas de usuario.	3
26	TS-02	Desarrollar API para autenticación de usuario	Como Developer, quiero crear una API que valide credenciales y genere token de autenticación.	3
27	TS-03	Desarrollar API para escanear productos	Como Developer, quiero crear una API que reciba imagen o código de barras y devuelva información del producto.	5
28	TS-04	Desarrollar API para registrar una venta	Como Developer, quiero crear una API que registre productos vendidos, cantidades y total.	5
29	TS-05	Desarrollar API para generar reportes de ventas	Como Developer, quiero crear una API que genere reportes filtrados por fechas, vendedores o categorías.	5
30	TS-06	Desarrollar API para exportación de reportes	Como Developer, quiero crear una API que permita exportar los reportes en PDF o Excel.	3
31	TS-07	Desarrollar API para geolocalización de vendedores	Como Developer, quiero crear una API que registre coordenadas de vendedores en tiempo real.	5

Capítulo IV: Strategic-Level Software Design.

4.1. Strategic-Level Attribute-Driven Design.

4.1.1. Design Purpose.

El propósito del diseño es desarrollar una solución tecnológica compuesta por una aplicación móvil para vendedores y una aplicación web para administradores. Esta solución debe facilitar la **gestión de productos, ventas y stock**, optimizando los tiempos de atención y asegurando la trazabilidad de las operaciones.

4.1.2. Attribute-Driven Design Inputs.

- Como **vendedor**, quiero usar reconocimiento de objetos para identificar productos rápidamente.
- Como **vendedor**, quiero verificar el stock en tiempo real para confirmar la disponibilidad del producto.
- Como **vendedor**, quiero reservar productos para asegurar que estén disponibles para el cliente.
- Como **vendedor**, quiero registrar las ventas para llevar un control de las transacciones.
- Como **vendedor**, quiero generar facturas para entregar un comprobante de compra al cliente.
- Como **administrador**, quiero revisar las ventas para mantener el control financiero.
- Como **administrador**, quiero ver y actualizar la lista de productos desde la web.
- Como **administrador**, quiero gestionar el inventario para evitar desabastecimientos o excesos.
- Como **administrador**, quiero crear y gestionar usuarios con diferentes roles y permisos.
- Como **administrador**, quiero generar reportes de ventas e inventario para la toma de decisiones.
- Como **cliente**, quiero ver el catálogo de productos con sus descripciones y precios.

4.1.2.1. Primary Functionality (Primary User Stories).

Con el objetivo de priorizar las historias de usuario clave para el funcionamiento básico del sistema y la satisfacción de los usuarios finales, se listaron aquellas con mayor impacto funcional y valor de negocio en el *product backlog*.

ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US01	Reconocimiento de Producto por Imagen	Como vendedor, quiero escanear un producto con la cámara del celular para obtener su información automáticamente, de forma rápida y precisa.	<p>Escenario 1: Reconocimiento exitoso. Dado que el vendedor escanea un producto, cuando el sistema reconoce el objeto, entonces se muestra su información (nombre, precio, stock).</p> <p>Escenario 2: Error en reconocimiento. Dado que el sistema no reconoce el producto, cuando el vendedor lo escanea, entonces se muestra un mensaje indicando que no se encontró y se sugiere verificar la imagen o buscar manualmente.</p>	EP01

ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
US02	Registro de Venta	Como vendedor, quiero registrar rápidamente la venta de un producto reconocido, para agilizar el proceso de atención al cliente.	<p>Escenario 1: Venta registrada correctamente. Dado que el producto fue reconocido, cuando el vendedor selecciona "Registrar venta", entonces la venta se almacena en el sistema y se actualiza el stock.</p> <p>Escenario 2: Error en el registro. Dado un fallo en el sistema, cuando el vendedor intenta registrar la venta, entonces se muestra un mensaje de error y se sugiere reintentar.</p>	EP02
US03	Seguimiento de Ventas	Como administrador, quiero acceder a un panel donde pueda ver un resumen de las ventas realizadas, para monitorear el rendimiento del negocio.	<p>Escenario 1: Visualización de estadísticas. Dado que el administrador accede al panel de ventas, cuando selecciona un rango de fechas, entonces el sistema muestra gráficos y datos agregados.</p> <p>Escenario 2: Fallo en la carga del panel. Dado un error del sistema, cuando el administrador intenta visualizar las estadísticas, entonces se muestra un mensaje indicando la falla.</p>	EP03
US04	Proyecciones de Ventas	Como administrador, quiero ver proyecciones de ventas futuras basadas en el historial, para tomar decisiones informadas sobre inventario y estrategia comercial.	<p>Escenario 1: Proyección generada. Dado que el sistema cuenta con suficientes datos históricos, cuando el administrador solicita una proyección, entonces se muestra un gráfico con estimaciones de venta.</p> <p>Escenario 2: Datos insuficientes. Dado que el historial de ventas es insuficiente, cuando se solicite una proyección, el sistema informará que no puede generar predicciones precisas aún.</p>	EP04

4.1.2.2. Quality attribute Scenarios.

Aspecto	Escenario	Respuesta
Rendimiento	Cuando un vendedor escanea un objeto, la aplicación debe mostrar la información del producto en 2 segundos.	Alta prioridad, debe cumplirse para asegurar la satisfacción del usuario.
Usabilidad	La interfaz de la aplicación móvil debe ser intuitiva, permitiendo que un nuevo vendedor aprenda las funciones básicas en 10 minutos de uso inicial.	El diseño debe priorizar la claridad y la navegación sencilla.
Seguridad	El acceso a datos sensibles como los registros de ventas debe requerir mecanismos de autenticación y autorización.	Crítico para proteger la información del negocio.
Disponibilidad	La aplicación debe estar disponible el 99.9% del tiempo durante las horas de trabajo.	Necesario para operaciones diarias confiables.
Mantenibilidad	El código debe ser modular y bien documentado para permitir actualizaciones y depuraciones sencillas.	Mejora la eficiencia del desarrollo y soporte a largo plazo.

4.1.2.3. Constraints.

A continuación se detallan las restricciones arquitectónicas que impactan el desarrollo de la aplicación:

Topología de la red:

La aplicación debe operar correctamente en redes locales heterogéneas, ya que será utilizada por distintas ferreterías que pueden tener infraestructuras de red limitadas o no estandarizadas. El sistema debe garantizar una conexión estable con APIs privadas de cada negocio y permitir reintentos en caso de fallos por latencia o pérdida de conexión.

Proveedor de base de datos:

No se utilizará una base de datos centralizada para toda la información operacional de las ferreterías. En su lugar, cada ferretería mantendrá su propio sistema de gestión (ERP, inventario, etc.), y Detekto deberá conectarse mediante APIs específicas provistas por cada cliente. Esto implica que el sistema debe ser capaz de adaptarse a distintos esquemas de datos y formatos de respuesta, usando adaptadores o middlewares para la integración. Internamente, Detekto podrá mantener una base temporal para almacenamiento de eventos o ventas, con consultas diarias sincronizadas.

Entorno web:

La aplicación web para administradores debe estar preparada para interactuar con APIs externas (propias de cada ferretería) y con el backend de Detekto. Debe permitir configuración de endpoints por cliente, manejo de autenticación por API Key o token JWT, y ser compatible con navegadores modernos (Chrome, Firefox, Safari). Debe alojarse en servidores seguros con configuración HTTPS y protección CORS activa.

Servidores:

El backend de Detekto será desplegado en contenedores Docker bajo infraestructura cloud (Railway, Render o similares). Debido a las limitaciones de estos entornos (recursos, concurrencia, tiempo de actividad), se deben evitar procesos de sincronización pesados o que dependan de disponibilidad constante de las APIs externas. Se recomienda establecer mecanismos de consulta programada (ej. cron jobs nocturnos) para actualizar información crítica como ventas e inventario.

Software de terceros:

Detekto debe integrarse con servicios externos según las necesidades de cada ferretería. Esto incluye:

APIs propias del cliente (ERP, stock, catálogo)

Servicios de notificación (email, WhatsApp Business API)

Herramientas de localización o seguimiento (Google Maps API u open source alternativos) La aplicación debe manejar límites de consumo (rate limits) y fallos en integraciones de forma desacoplada.

Normas y regulaciones:

Detekto debe garantizar que los datos de cada ferretería se manejen de forma aislada y segura (multitenancy). Cada instancia debe respetar la privacidad del cliente, evitando compartir o almacenar de forma cruzada información sensible. Además, se debe cumplir con la Ley N° 29733 de Protección de Datos Personales. Toda la comunicación debe realizarse mediante conexiones cifradas (TLS), y los accesos deben estar protegidos mediante autenticación robusta por rol (administrador, vendedor, cliente).

4.1.2.2. Technical Stories

Estas historias técnicas representan tareas esenciales para el soporte de funcionalidades clave, así como la mantenibilidad, rendimiento y seguridad de la aplicación.

Technical Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
TS01	Integración del Reconocimiento por Cámara	Implementar la funcionalidad que permite acceder a la cámara del dispositivo móvil para capturar imágenes y analizarlas con el modelo de IA.	<p>Escenario 1: Acceso correcto a la cámara. Cuando el usuario activa la cámara desde la app, entonces se debe abrir correctamente y permitir capturar una imagen.</p> <p>Escenario 2: Fallo de acceso. Si la cámara no puede abrirse, mostrar error.</p>	EP01

Technical Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
TS02	Entrenamiento del Modelo de IA	Entrenar y validar un modelo de machine learning capaz de identificar productos desde imágenes capturadas.	Escenario 1: Precisión mayor al 90%. Al evaluar el modelo, debe alcanzar al menos un 90% de precisión en el reconocimiento. Escenario 2: Generación de reporte de métricas de desempeño del modelo.	EP01
TS03	API de Ventas	Diseñar e implementar una API RESTful para registrar y consultar ventas realizadas a través de la aplicación móvil.	Escenario 1: Registro exitoso. Cuando se envía una venta por POST, debe almacenarse correctamente y devolver una respuesta 200. Escenario 2: Validación de campos obligatorios en el request.	EP02
TS04	Panel de Control para Administradores	Crear el módulo backend y frontend para mostrar las estadísticas de ventas y proyecciones a los administradores.	Escenario 1: El administrador puede consultar estadísticas por fecha y ver gráficos. Escenario 2: El sistema debe alertar si los datos son insuficientes para proyecciones.	EP03 / EP04
TS05	Sistema de Autenticación	Implementar un sistema de autenticación segura con tokens JWT para proteger rutas de administrador y vendedor.	Escenario 1: Login válido. Dado un usuario registrado, cuando introduce sus credenciales, recibe un token JWT válido. Escenario 2: Acceso restringido sin token. Si no hay token, no se puede acceder a rutas protegidas.	EP03

4.1.3. Architectural Drivers Backlog

Esta sección identifica los impulsores clave de la arquitectura que influyen significativamente en las decisiones de diseño del sistema.

Driver ID	Título de Driver	Descripción	Importancia para Stakeholders	Impacto en Architecture Technical Complexity
AD01	Rendimiento en Reconocimiento de Objetos	El sistema debe reconocer objetos en menos de 2 segundos para asegurar fluidez en la experiencia de venta.	High	High
AD02	Seguridad de Datos de Ventas	Los datos de ventas y usuarios deben estar protegidos mediante autenticación y autorización robusta.	High	Medium
AD03	Escalabilidad del Sistema	El sistema debe poder escalar a múltiples usuarios y dispositivos sin pérdida significativa de rendimiento.	Medium	High
AD04	Disponibilidad de la Aplicación	La app debe estar disponible el 99.9% del tiempo durante horarios de venta.	High	Medium
AD05	Mantenibilidad del Código	El sistema debe tener un código modular y documentado para facilitar actualizaciones futuras.	Medium	Medium
AD06	Interfaz Intuitiva	La interfaz debe ser lo suficientemente sencilla como para que un nuevo usuario la aprenda en 10 minutos.	High	Low

4.1.4. Architectural Design Decisions.

Esta sección documenta las decisiones arquitectónicas clave tomadas durante el desarrollo del sistema, incluyendo el contexto, las alternativas evaluadas y las razones de la elección final.

Decisión ID	Título de la Decisión	Descripción	Justificación
DD01	Uso de Arquitectura Basada en Componentes	Se decidió estructurar la aplicación móvil y backend en componentes independientes (reconocimiento, ventas, autenticación, administración, etc.).	Favorece la mantenibilidad, permite trabajo paralelo y facilita futuras migraciones a microservicios.
DD02	Integración de Modelo IA con TensorFlow Lite	Se optó por usar TensorFlow Lite para correr el modelo de reconocimiento de objetos directamente en dispositivos móviles.	Alta precisión y velocidad en dispositivos móviles; buena compatibilidad con Android/iOS.
DD03	Almacenamiento en Firebase y Firestore	Se eligió Firebase Authentication y Firestore para el backend de autenticación y almacenamiento de datos.	Reduce complejidad operativa, ofrece escalabilidad y se integra fácilmente con apps móviles.

Decisión ID	Título de la Decisión	Descripción	Justificación
DD04	Uso de JWT para Autenticación	Se decidió implementar autenticación basada en tokens JWT para proteger endpoints y gestionar sesiones de usuario.	Es una solución moderna, segura y ampliamente utilizada en aplicaciones móviles/web.
DD05	Framework Flutter para el Frontend	Se seleccionó Flutter como tecnología base para el desarrollo de la app móvil.	Permite desarrollo multiplataforma nativo (Android/iOS), alta velocidad de desarrollo y buena experiencia UX.
DD06	API RESTful para Comunicación entre Módulos	Se adoptó REST como estilo arquitectónico para la comunicación entre frontend y backend.	Simplicidad, amplia adopción y fácil de consumir desde cualquier cliente HTTP.
DD07	Diseño Responsivo y Accesibilidad	Se tomó la decisión de diseñar la interfaz siguiendo principios de diseño accesible y adaptable a distintos tamaños de pantalla.	Mejora la experiencia del usuario y asegura inclusión de distintos perfiles de usuarios.

4.1.5. Quality Attribute Scenario Refinements

Refinamientos detallados para los escenarios de atributos de calidad, con el objetivo de precisar sus condiciones, respuestas esperadas y métricas asociadas.

Quality Attribute	Escenario Original	Refinamiento
Rendimiento	Mostrar información del producto en 2 segundos tras escanear un objeto	El sistema debe garantizar que el tiempo total desde que el usuario abre la cámara hasta mostrar los resultados no excede los 2000 ms.
Usabilidad	Interfaz intuitiva que permite aprender funciones básicas en 10 minutos	Al menos el 90% de los nuevos usuarios debe poder completar una venta simulada en menos de 10 minutos sin ayuda externa.
Seguridad	Requiere autenticación para acceder a datos sensibles	El acceso a datos sensibles debe requerir autenticación multifactor y roles de usuario (admin/vendedor) diferenciados.
Disponibilidad	Disponible el 99.9% del tiempo durante horas de trabajo	Debe haber monitoreo activo y recuperación automática ante fallos para garantizar la disponibilidad mínima de 99.9% entre 8am y 8pm.
Mantenibilidad	Código modular y documentado	Todo nuevo módulo debe incluir pruebas unitarias, comentarios explicativos y cumplir con las guías internas de estilo y arquitectura.

4.2. Strategic-Level Domain-Driven Design.

4.2.1. EventStorming.

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

Step 1: Unstructured Exploration

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



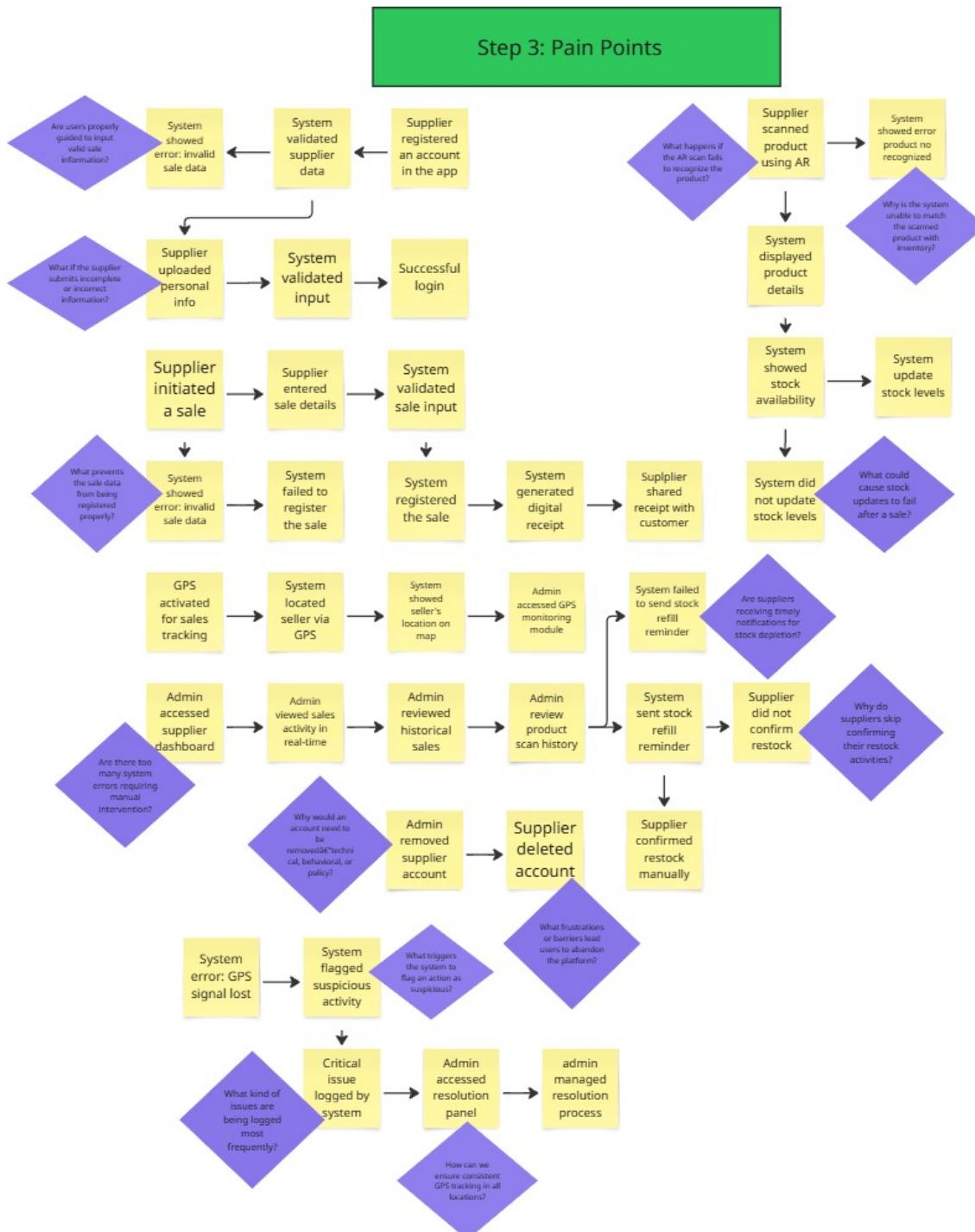
Step 2: Timelines:

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



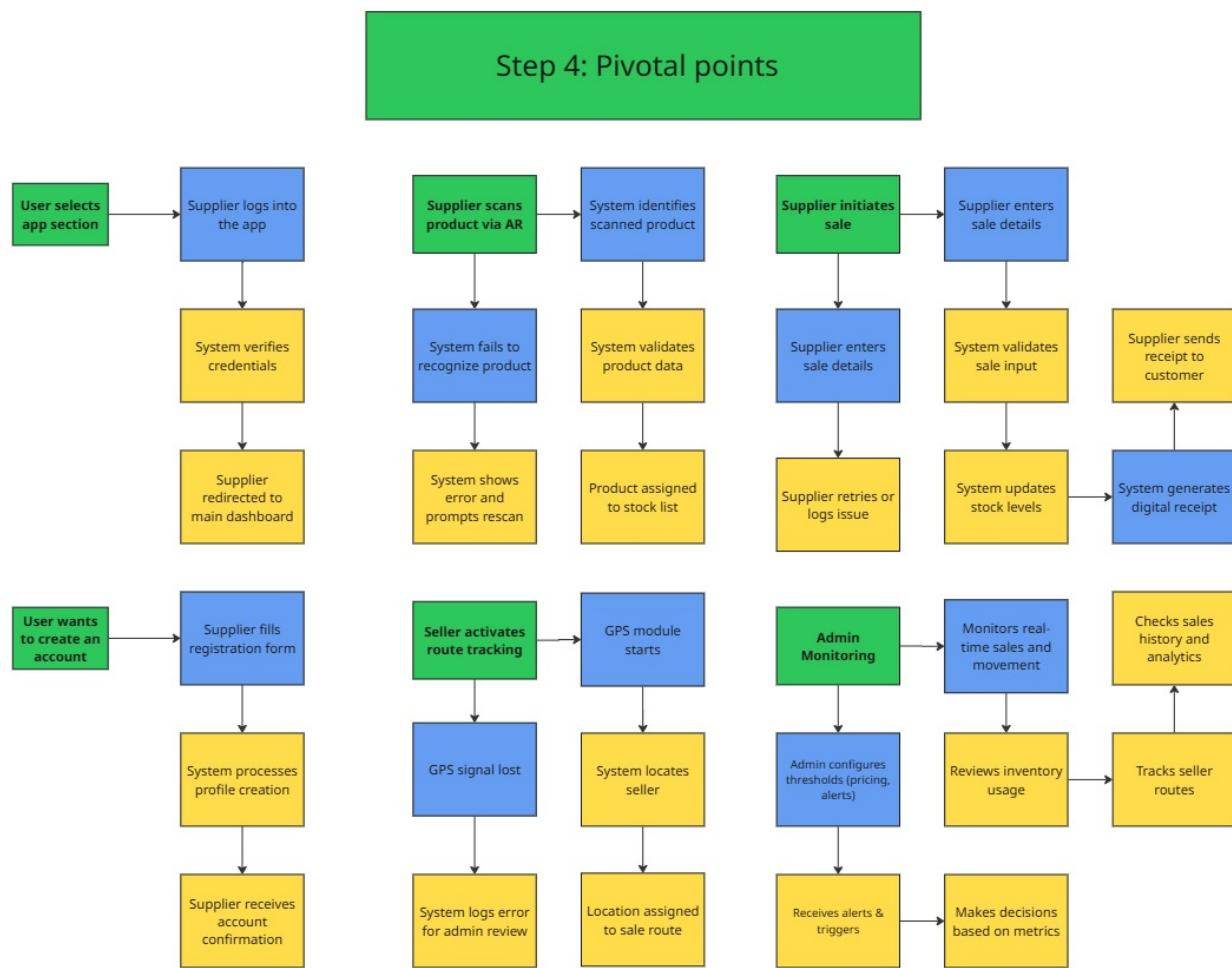
Step 3: Pain points

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



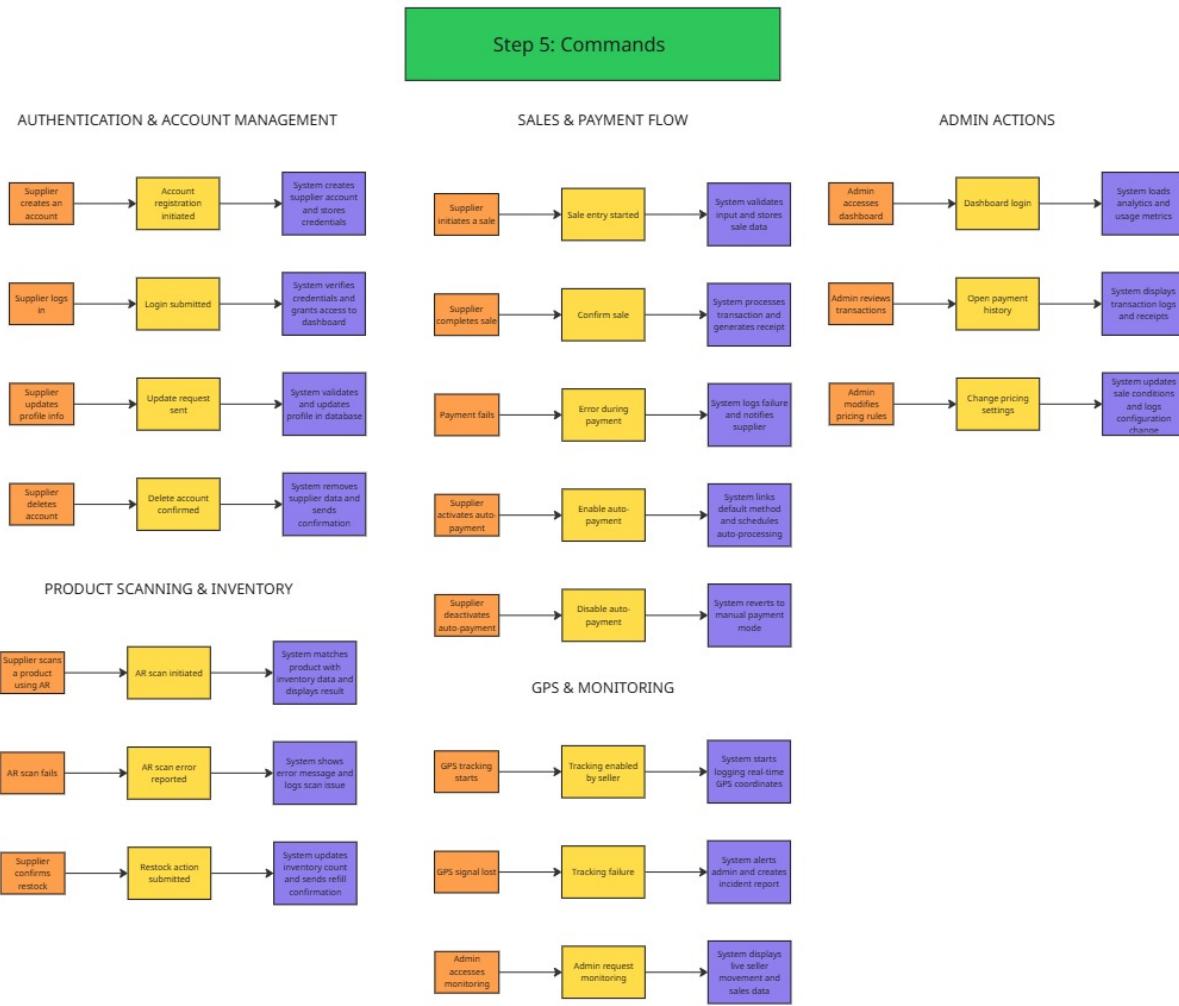
Step 4: Pivotal points

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



Step 5: Commands

En este paso se identificó los comandos realizados por sus respectivos actores.



Step 6: Policies

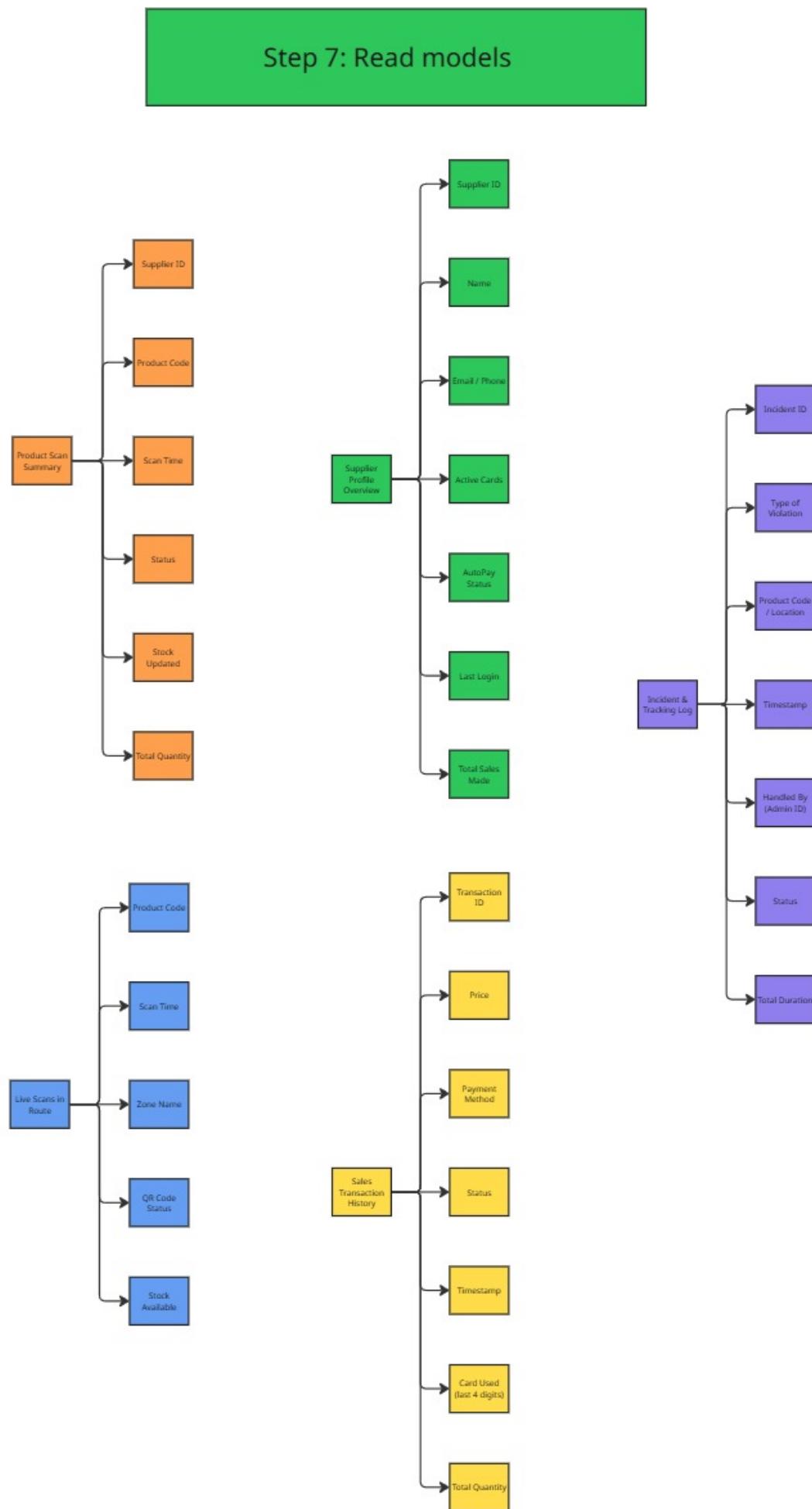
En este paso se incluyeron las políticas, escenarios donde la aplicación ejecuta alguna acción.

Step 6: Policies

Suppliers must be authenticated before accessing sales, stock, or tracking features.	All sensitive updates (e.g., payment methods, business info) must be confirmed by the user before saving.	Account deletion must be irreversible and require final confirmation.	Past transaction data and analytics must only be visible to authenticated users.	Only authorized admins can access the real-time GPS dashboard and stock analytics.
All critical system events (e.g., location changes, stock failures) must be logged for admin review.	Auto-payment can only be activated if at least one valid card is linked.	Deleted payment methods must be confirmed as removed before allowing new ones.	All payment actions must validate the card before processing.	If payment fails, the system must guide the user with recovery steps and retry options.
Sales receipts must be auto-generated and stored for at least 30 days.	Product scans must match a registered SKU to be accepted into the inventory system.	Duplicate scans must be flagged or ignored to prevent stock manipulation.	Failed AR scans must trigger a fallback photo/manual entry method.	Suppliers exiting before registering a sale must trigger a manual admin review.
Real-time location tracking should only activate after login and confirmation by the supplier.	If the GPS signal is lost, the system must raise a flag for potential review.	Multiple lost signal events in a short period should trigger user monitoring.	Sales timeout or delay must notify the user at least 5 minutes before auto-deactivation.	If a supplier repeatedly skips confirming sales or restocks, the system should monitor the pattern.
Admins can mark areas or periods as 'inventory lockdown' or 'manual override' in special situations.	If no transaction history is found, the system must offer the user guidance or support contact options.			

Step 7: Read Models

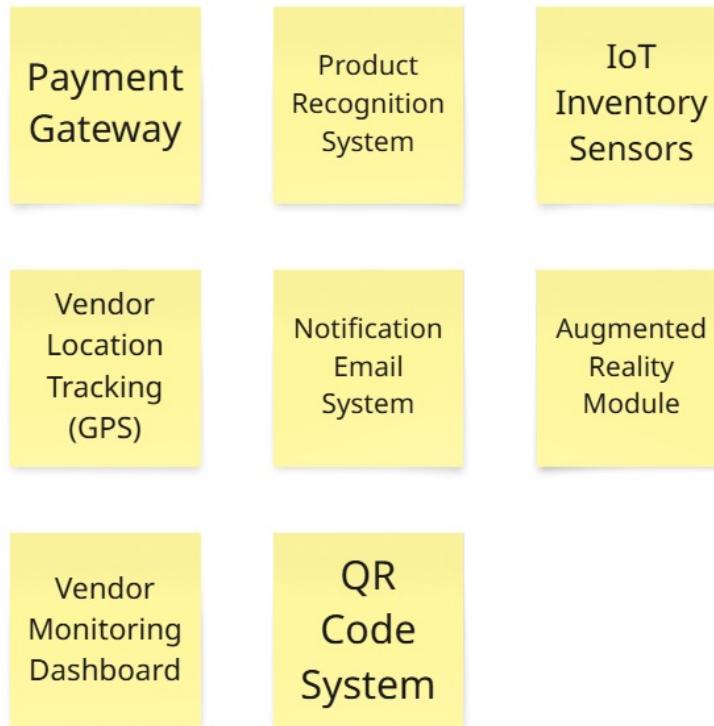
En este paso se realizó el modelo de lectura de cada sistema de la aplicación.



Step 8: External systems

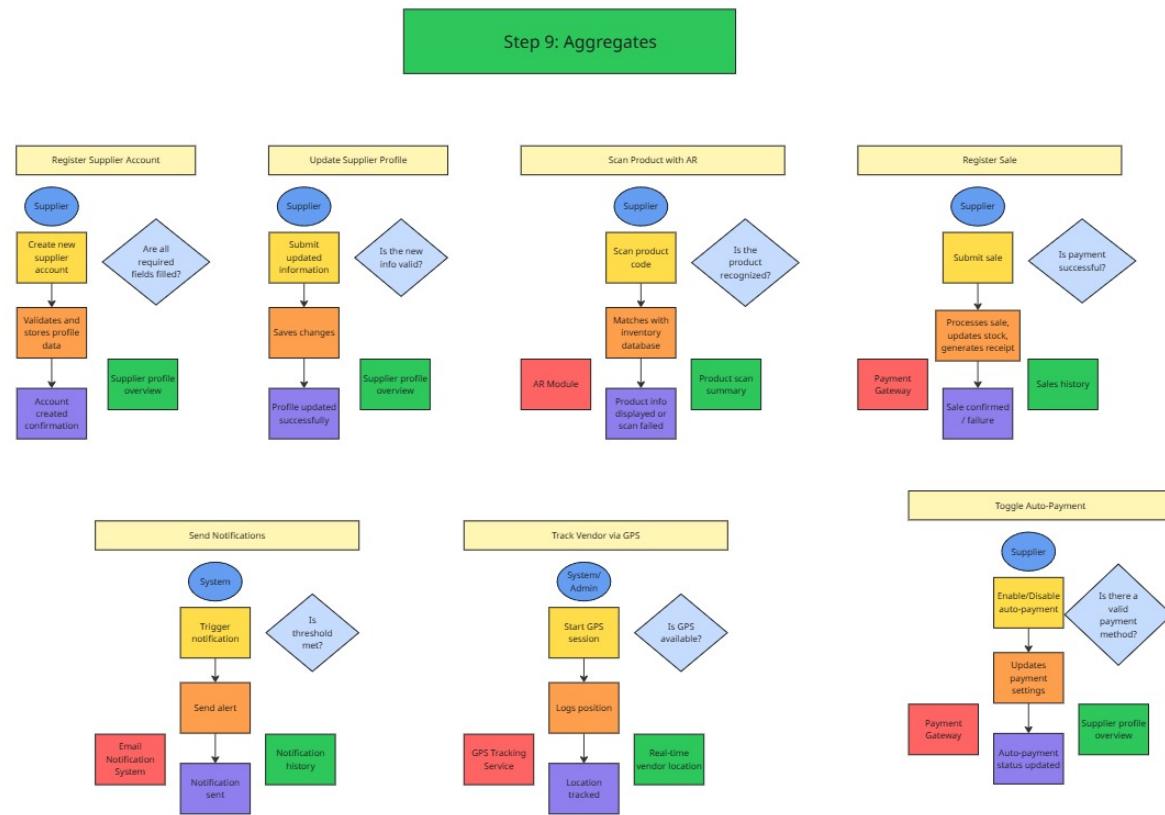
En este paso se identificaron algunos sistemas externos.

Step 8: External Systems



Step 9: Aggregates

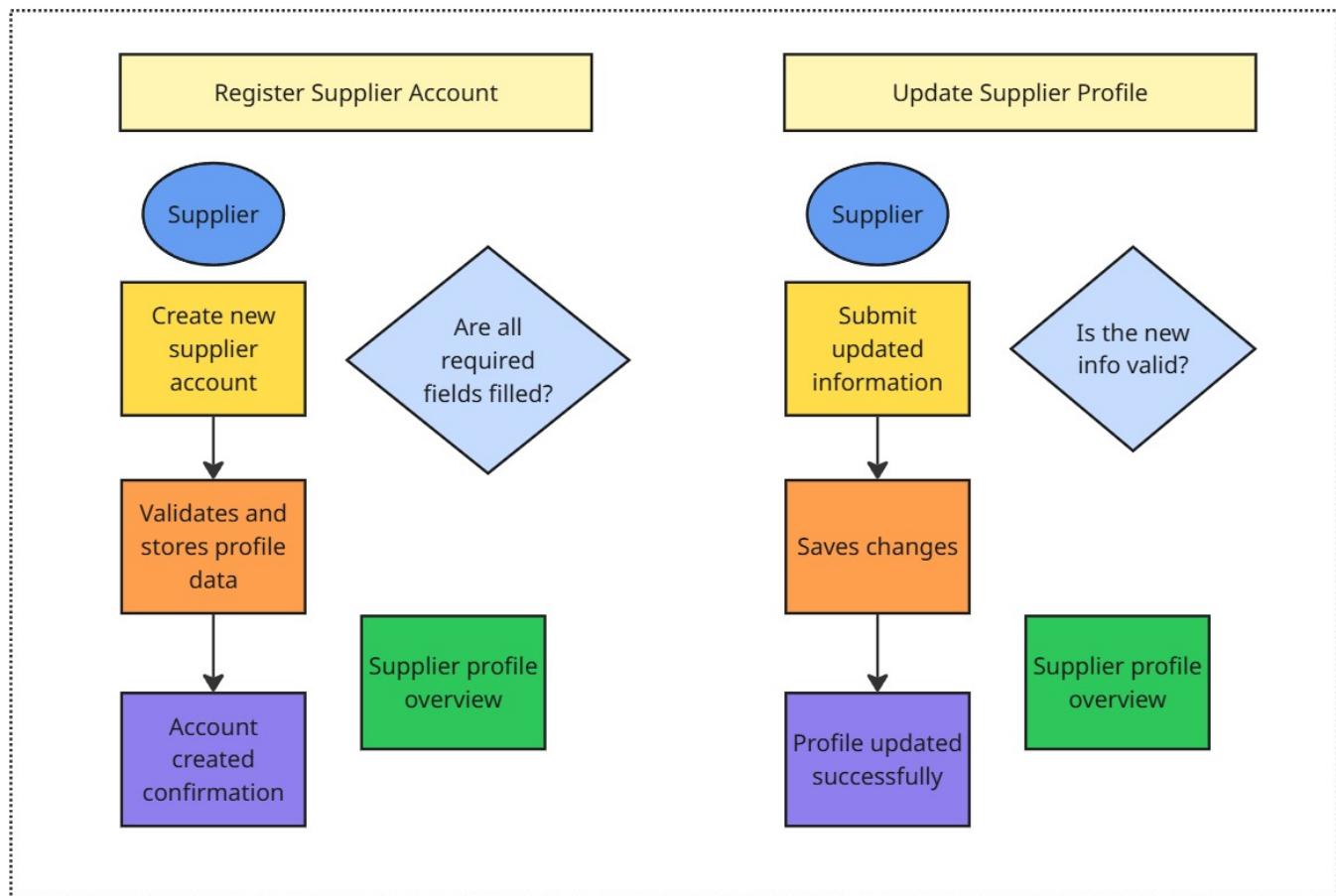
En este paso se implementan los agregados de cada sistema de la aplicación.



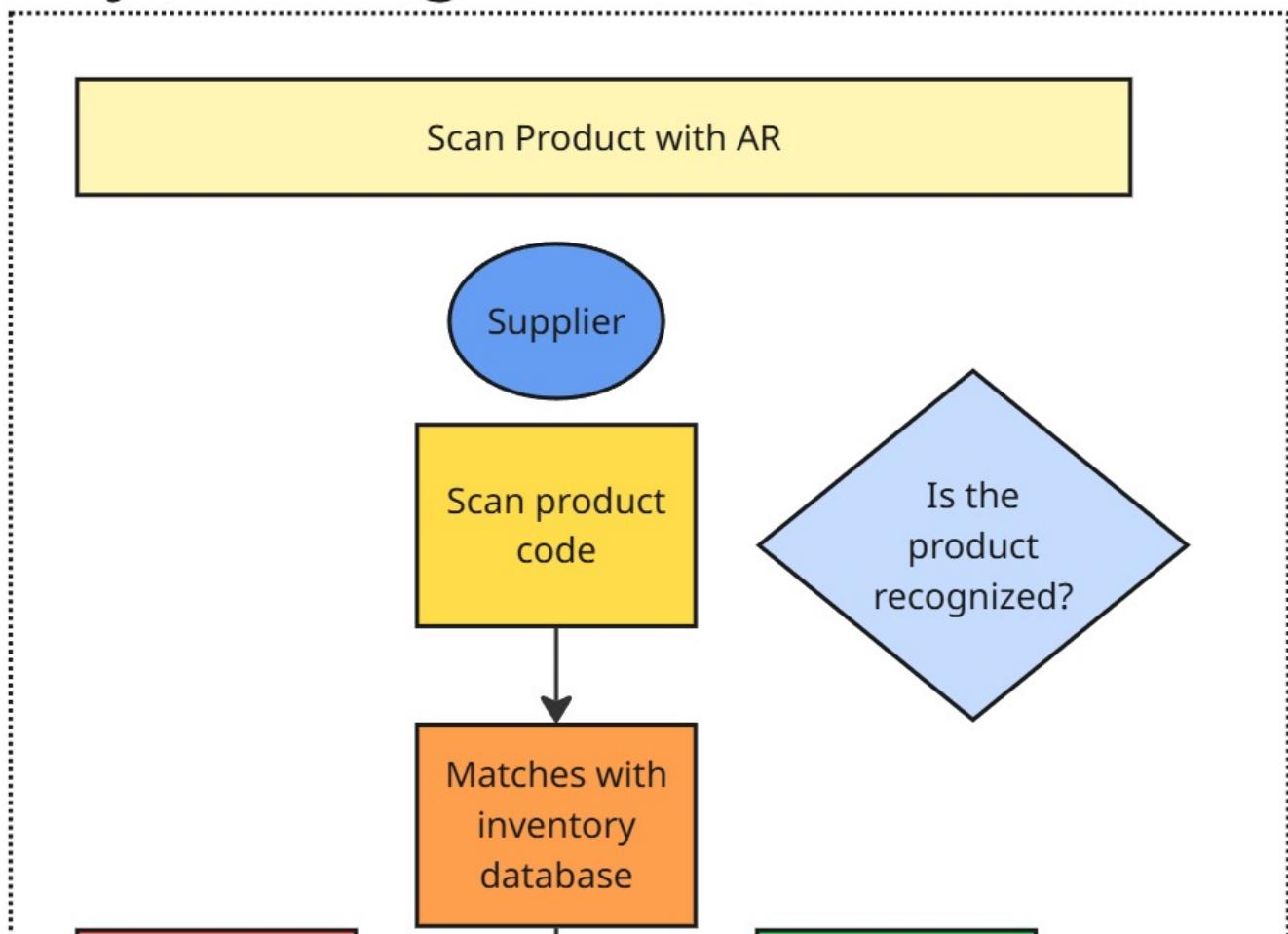
4.2.2. Candidate Context Discovery.

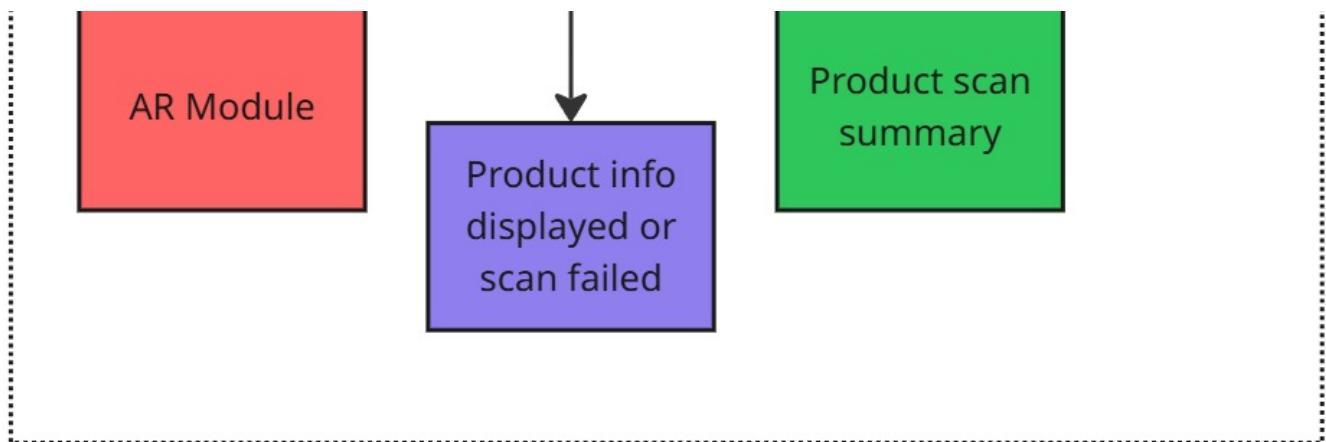
En este punto luego de la realización del Event storming, se identificaron los bounded context a trabajar. En este caso se lograron identificar 5.

IAM



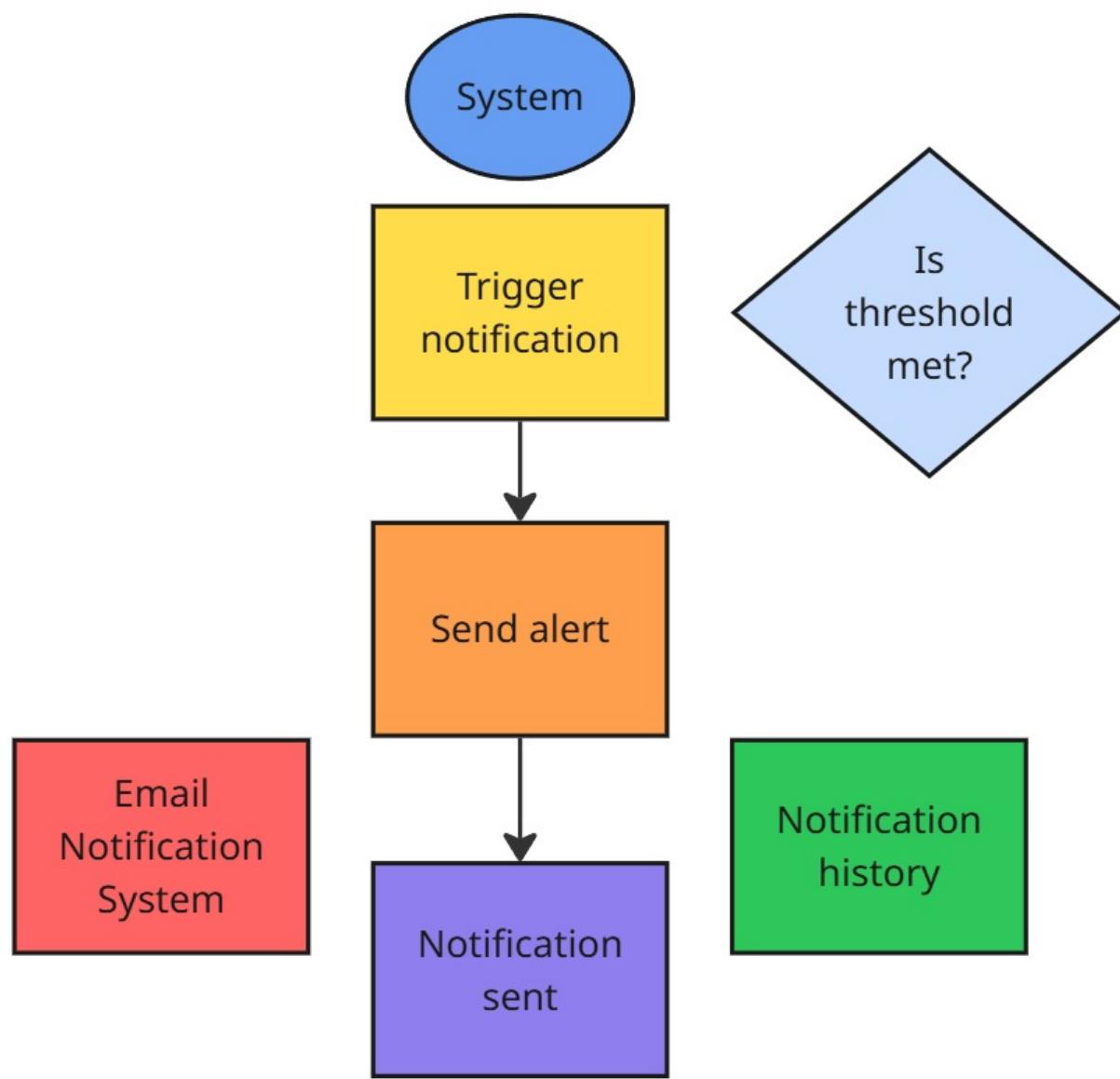
Object Recognition





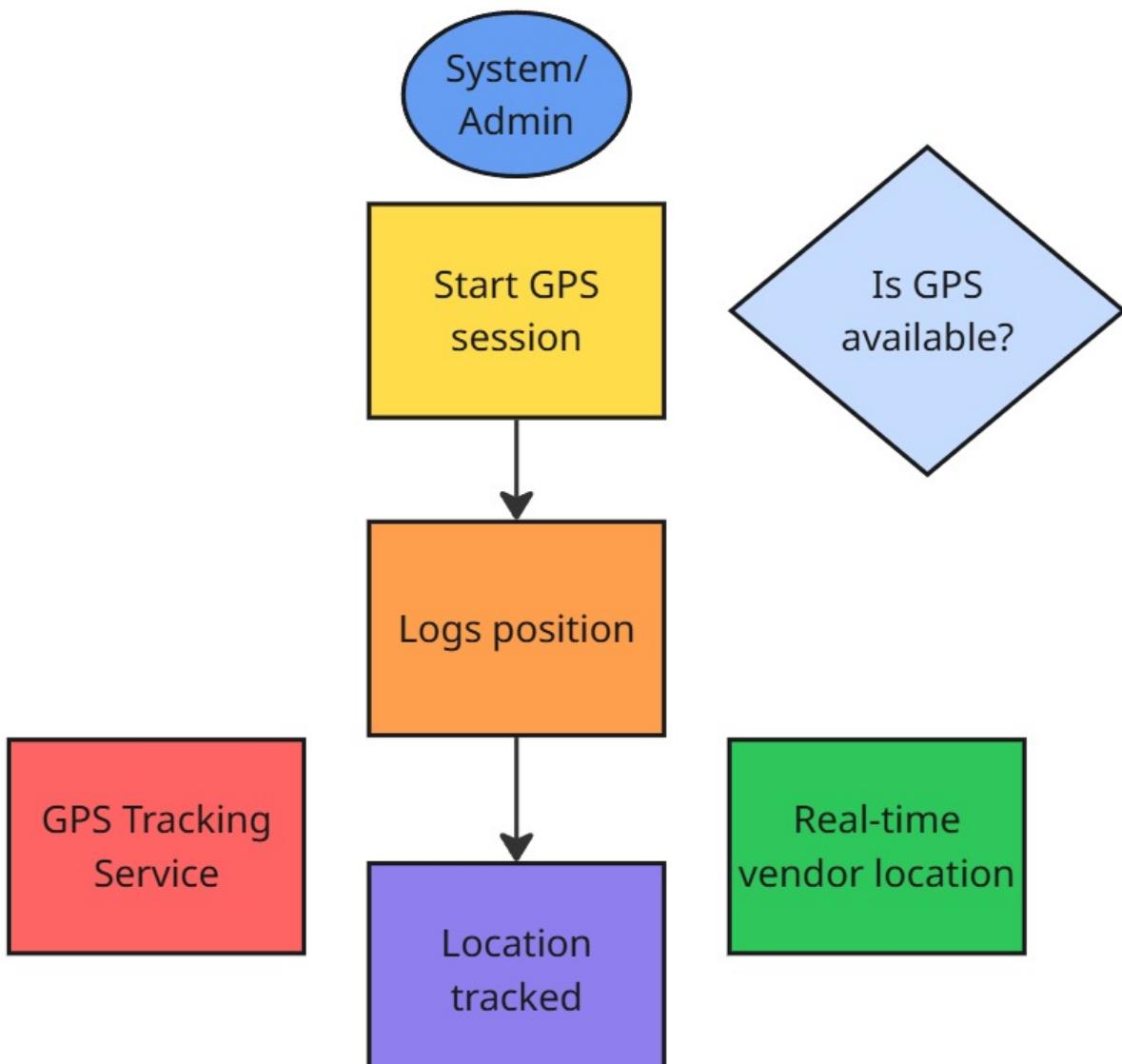
Notifications

Send Notifications



Tracking & Monitoring

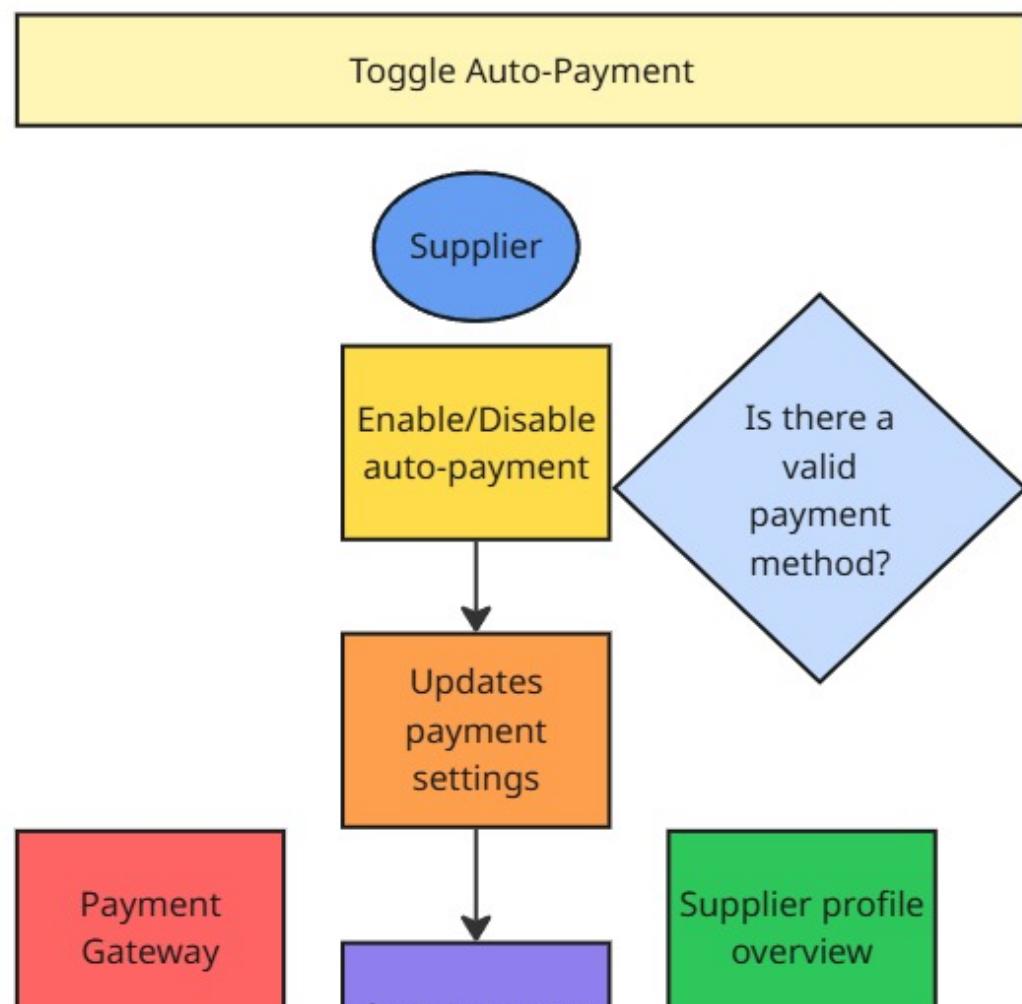
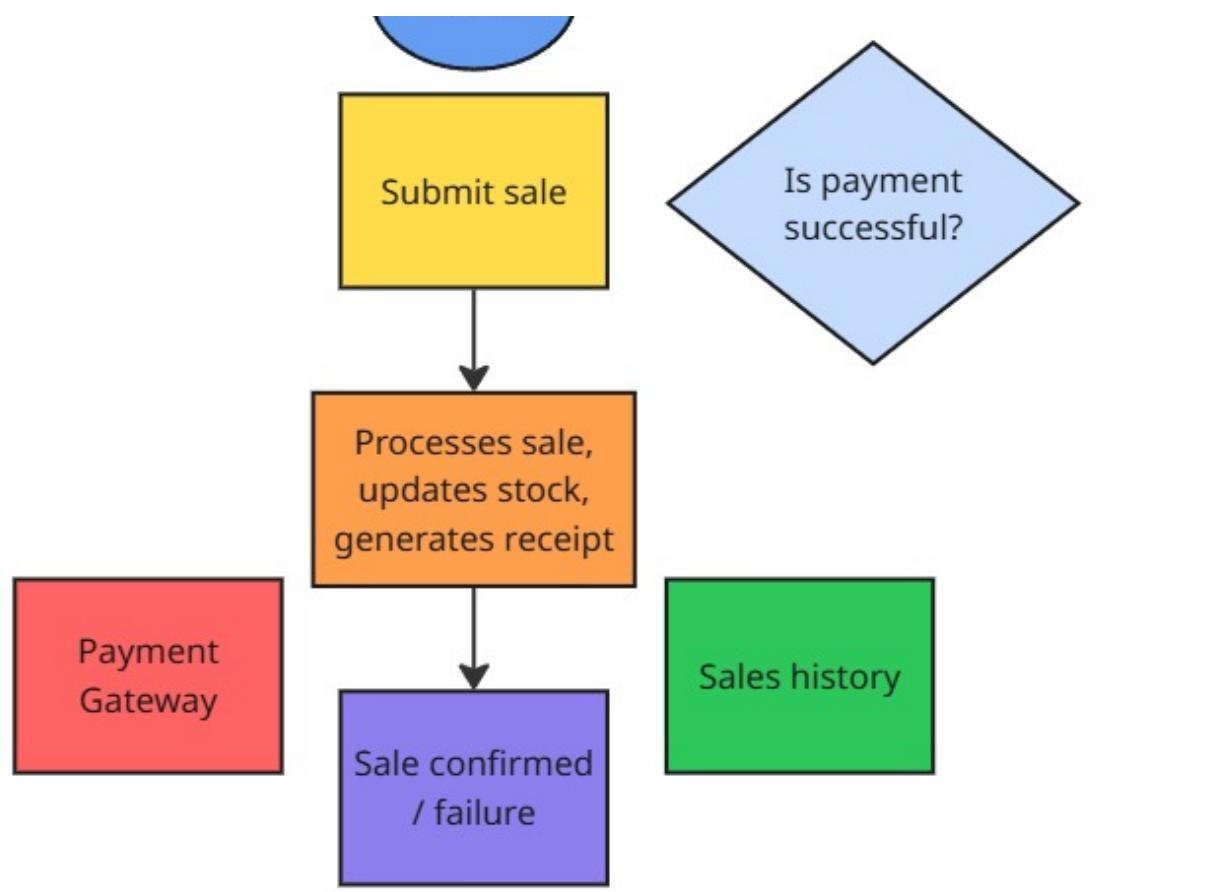
Track Vendor via GPS

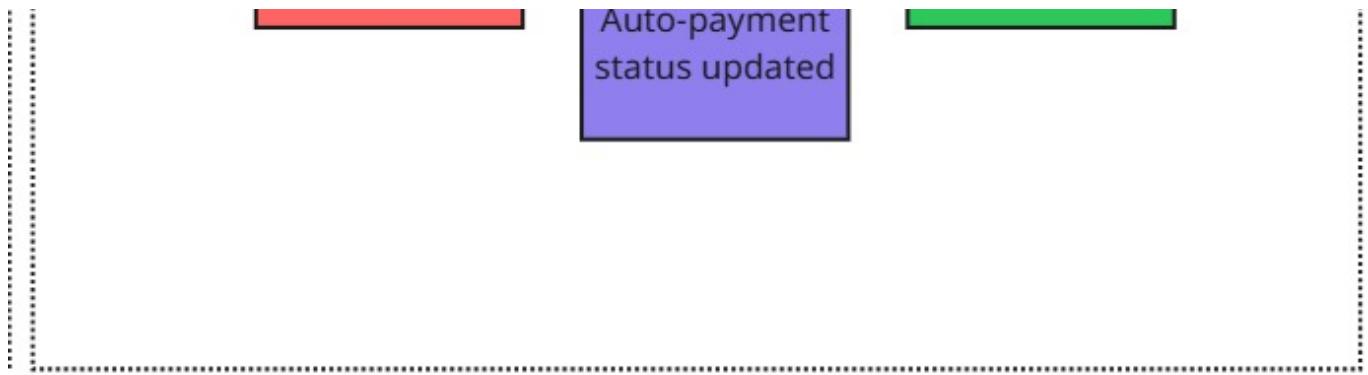


Sales

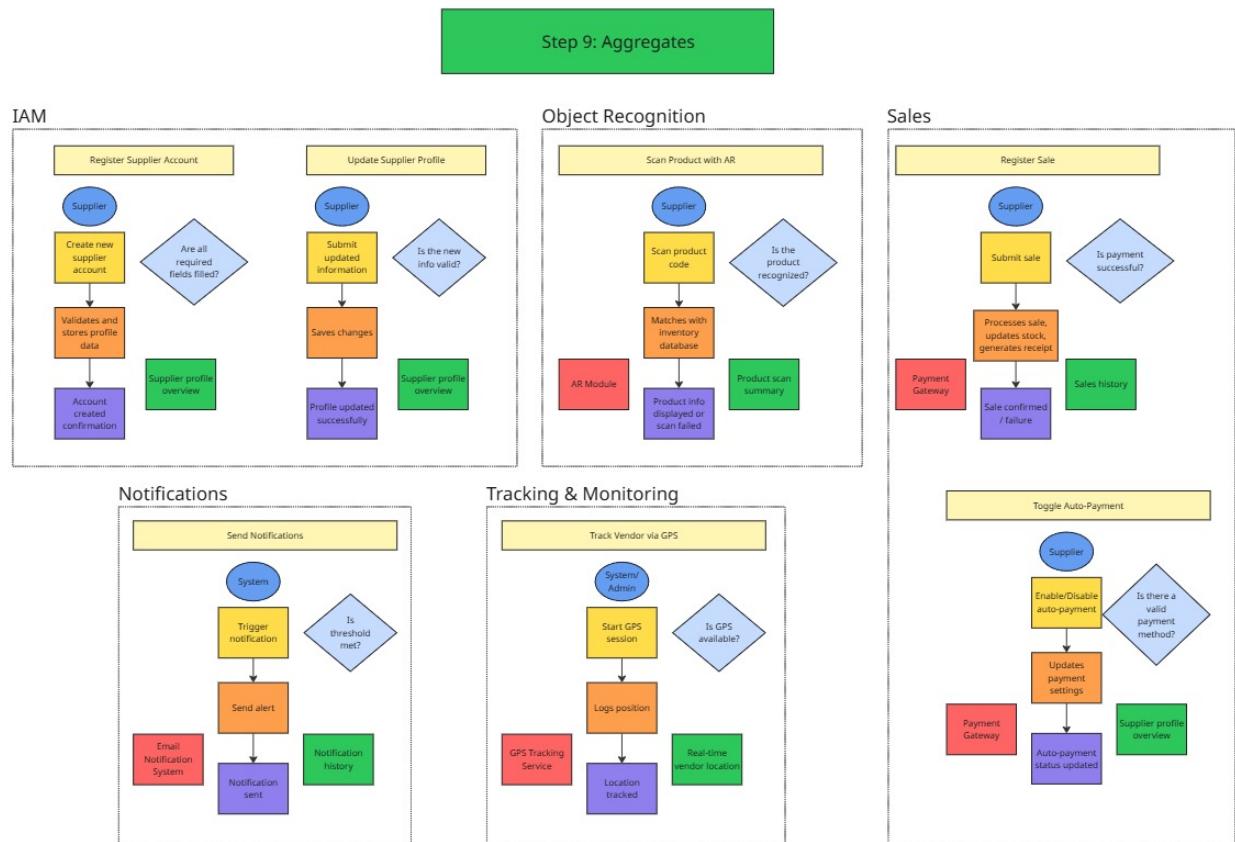
Register Sale

Supplier





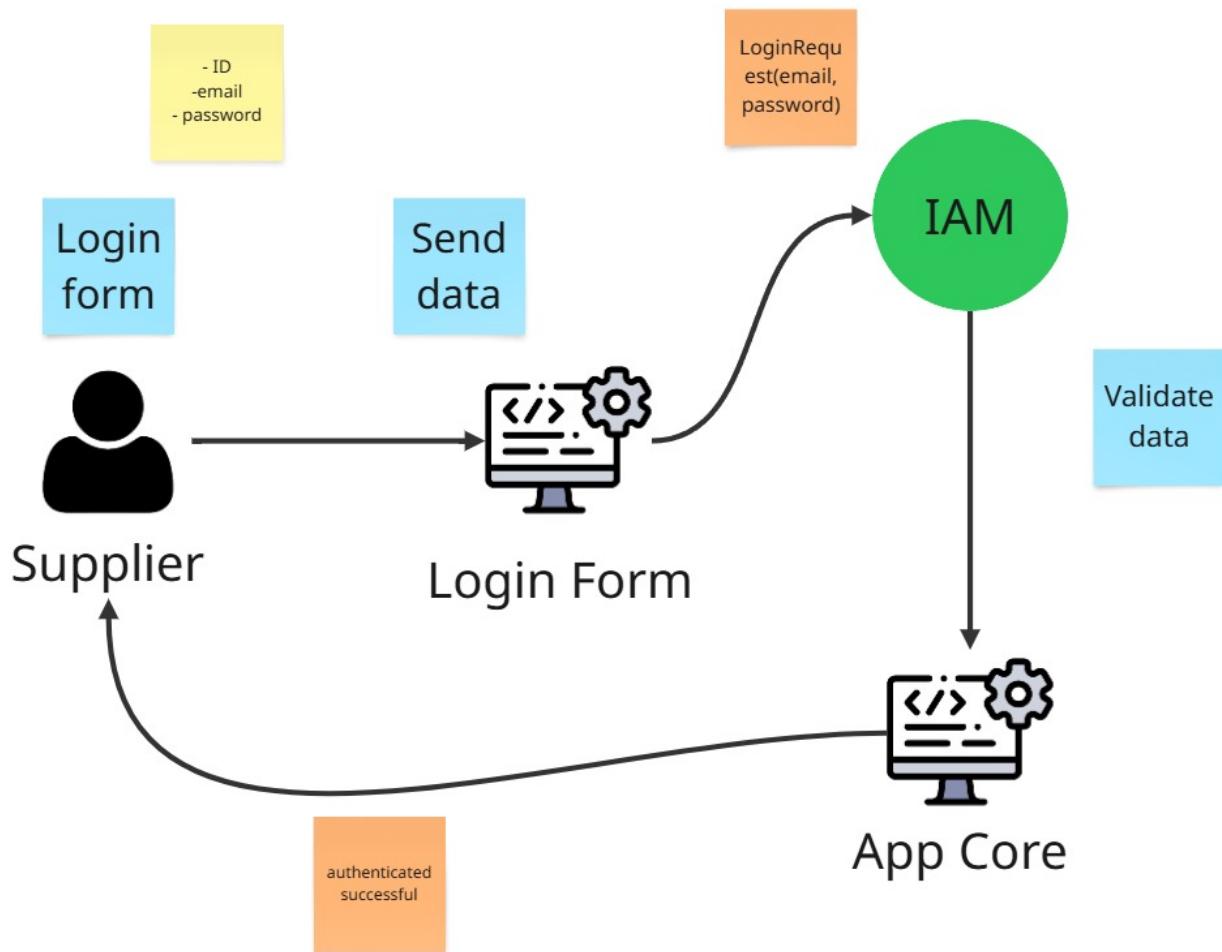
Aquí podemos apreciarlo completo:



4.2.3. Domain Message Flows Modeling.

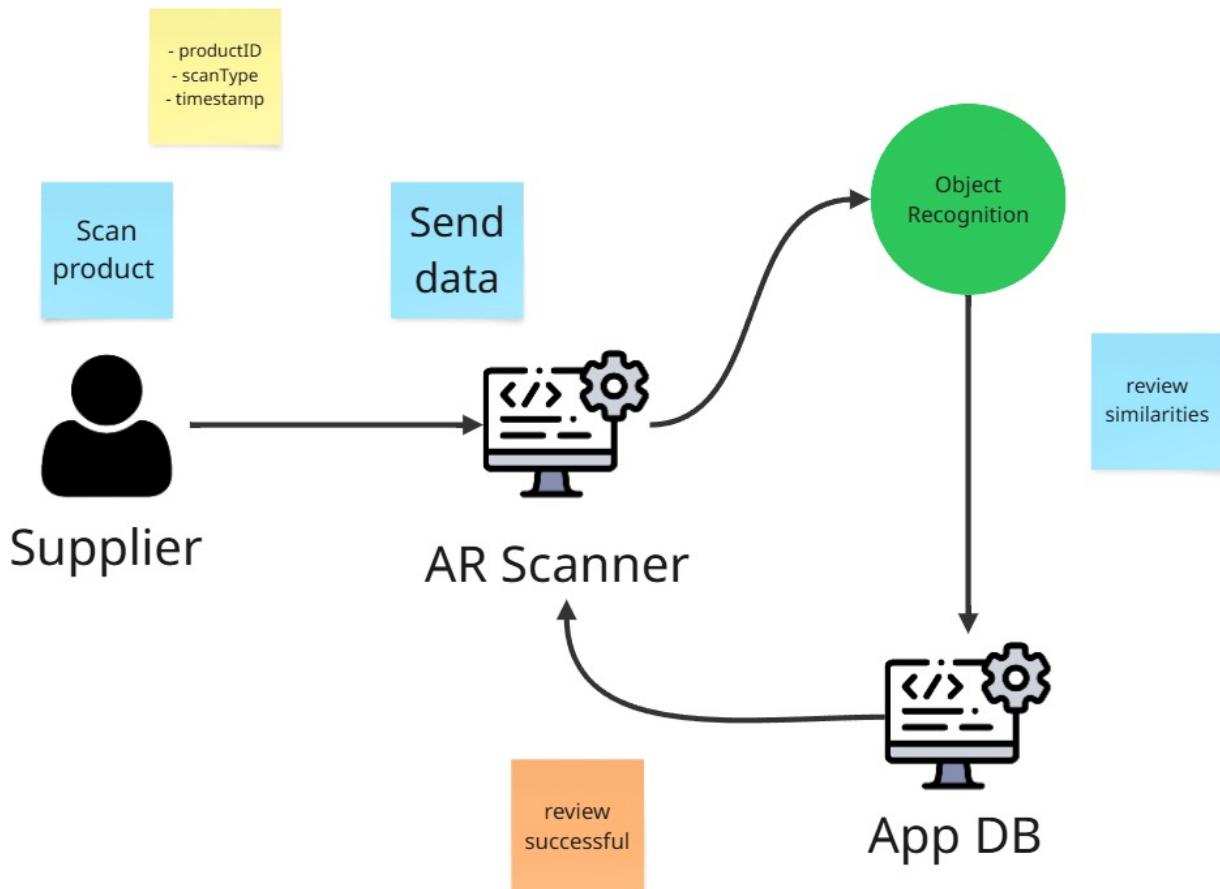
Actor: Supplier

Escenario: El proveedor inicia sesión



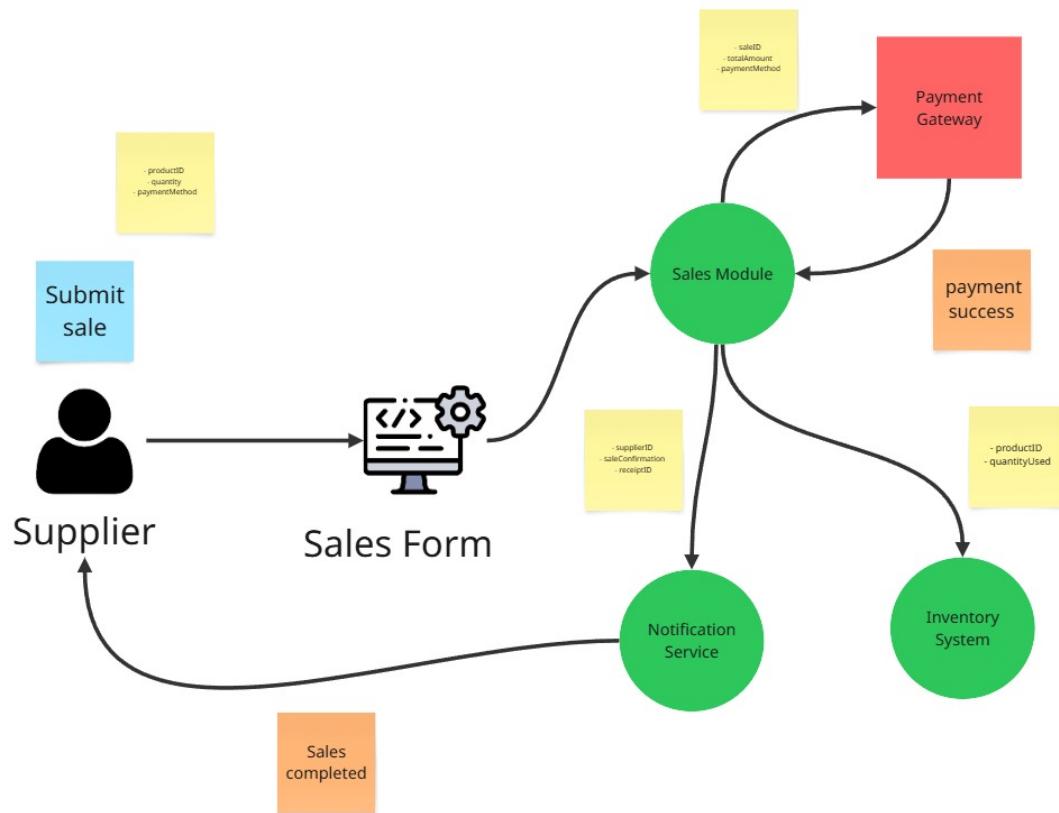
Actor: Supplier

Escenario: El proveedor registra una venta luego del escaneo exitoso



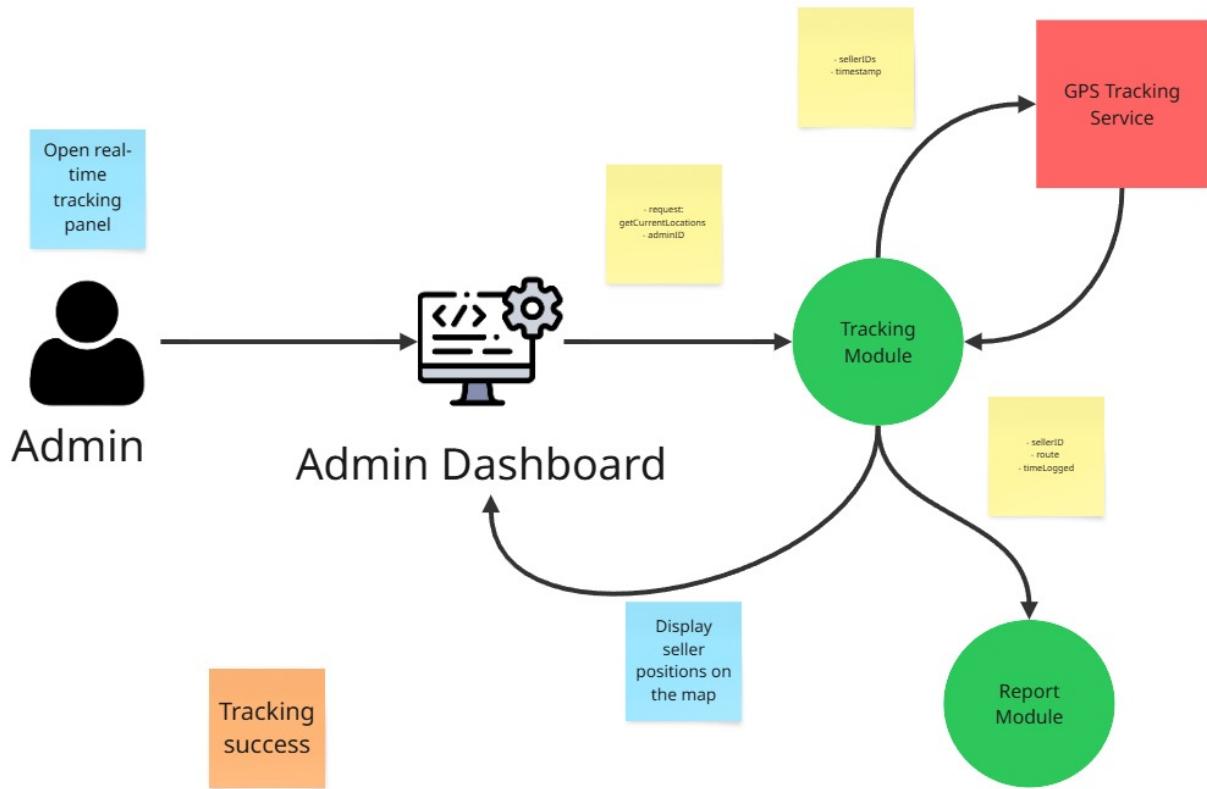
Actor: Supplier

Escenario: El proveedor registra una venta luego del escaneo exitoso



Actor: Admin

Escenario: El administrador revisa rutas y estado en vivo de vendedores



4.2.4. Bounded Context Canvases.

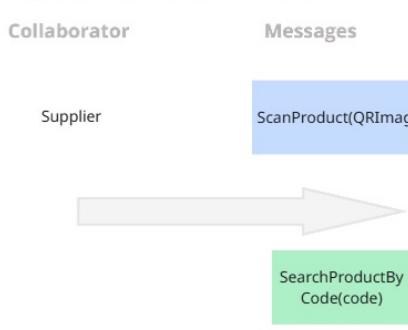
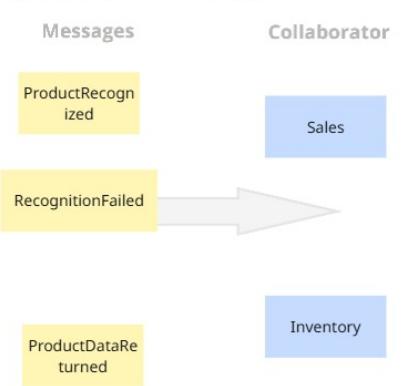
- IAM:

Name: Sales		V5 github.com/ddd-crew/bounded-context-canvas
Purpose Manage and register product sales made by suppliers, including payment confirmation, stock updates, and sales logs.	Strategic Classification - Domain: Core Domain - Business Model: Custom Build - Evolution: Differentiator - Domain Roles: Execution Context	Domain Roles
Inbound Communication		
Collaborator	Messages	
Supplier	SubmitSale(productID, quantity, paymentMethod)	
AR module	ConfirmPayment(transactionID)	
Ubiquitous Language		
Sale Receipt Payment status Stock updated Registered product Auto-payment		
Business Decisions		
Payment must be validated before completing sale Sales auto-confirm only if auto-payment is active Stock deduction is mandatory and must be atomic with sale registration Sale fails if product is out of stock or not recognized		
Outbound Communication		
Collaborator	Messages	
Inventory System	SaleConfirmed	
Notification Service	StockUpdated	
Payment Gateway (external)	SaleFailed	
Payment Gateway (external)	ReceiptGenerated	
Payment Gateway (external)	NotifySupplier	
Assumptions Suppliers input sale details via mobile app Payments are handled through an integrated external API Products must be scanned or selected from inventory before submission		Verification Metrics Number of sales per supplier per day Successful vs. failed sale attempts Stock discrepancies flagged per week
		Open Questions - Should discounts be managed in this context? - Who handles fraud or duplicate sale detection?

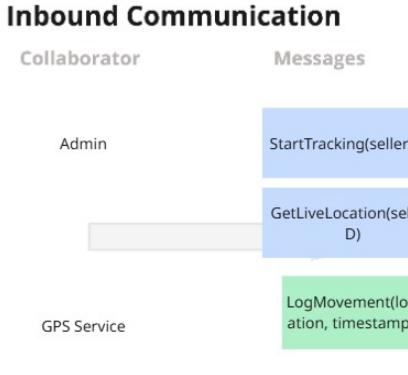
- Sales:

Name: IAM		V5 github.com/ddd-crew/bounded-context-canvas
Purpose Authenticate suppliers and manage login, logout, and identity validation for secure access to app features.	Strategic Classification - Domain: Generic - Business Model: Compliance Enforcer - Evolution: Commodity - Domain Roles: Execution Context	Domain Roles
Inbound Communication		
Collaborator	Messages	
Supplier	Login(email, password)	
Security	Logout()	
Security	ValidateToken(token)	
Ubiquitous Language		
Login Logout Session Authenticated Token Credentials		
Business Decisions		
Login is required before any product interaction Token expires after defined inactivity Password resets require multi-step verification		
Outbound Communication		
Collaborator	Messages	
Profile	Authentication Success	
Sales	AuthenticationFailed	
AR Module	TokenExpired	
Assumptions Users must authenticate through the login page Sessions are managed securely using encrypted tokens		Verification Metrics Daily active sessions Failed login attempts Token expiration rate
		Open Questions - Should biometric login be included?

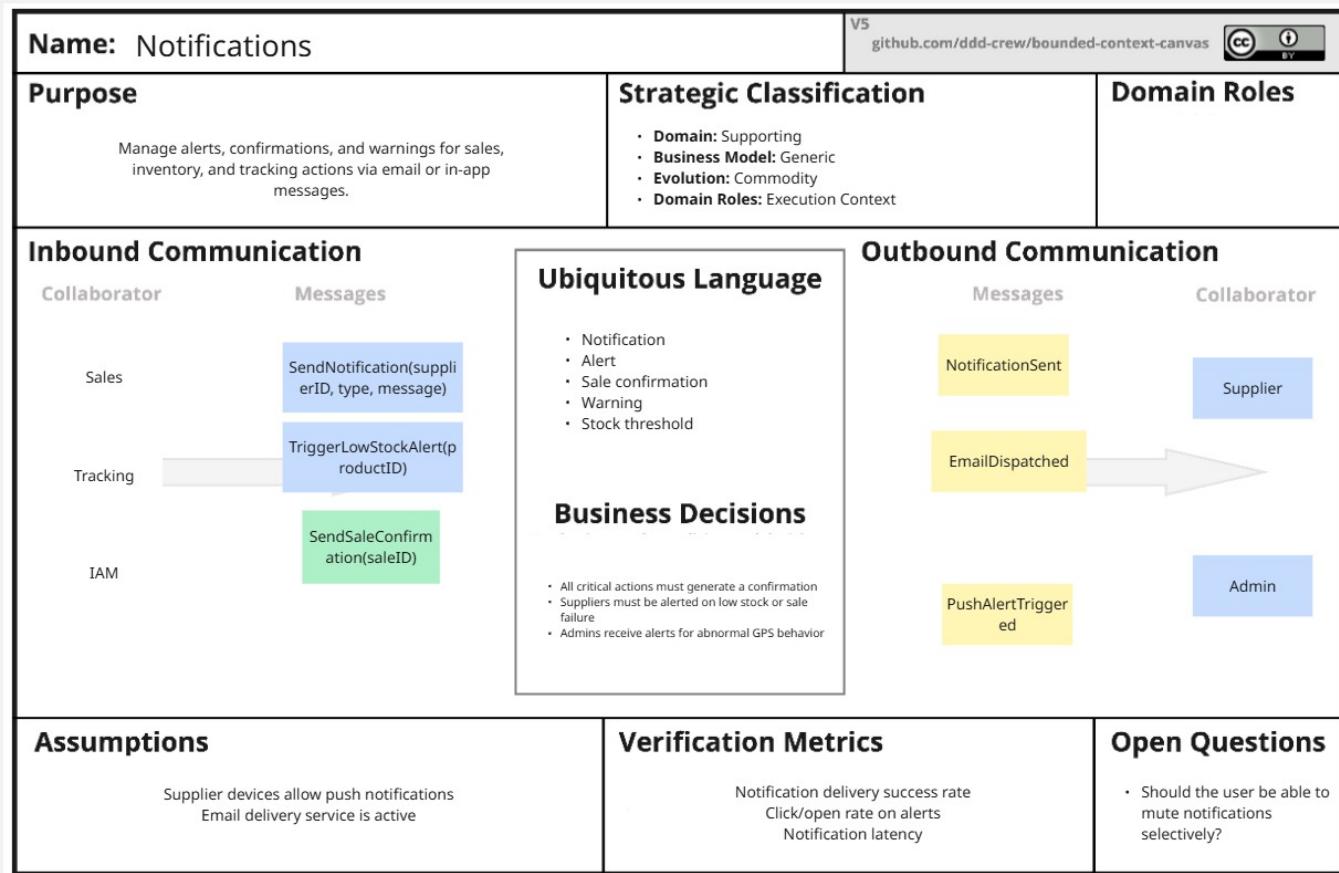
- Object Recognition:

Name: Object Recognition		V5 github.com/ddd-crew/bounded-context-canvas 
Purpose Allow suppliers to scan products using augmented reality or QR code and identify them within the system.	Strategic Classification <ul style="list-style-type: none">• Domain: Core Domain• Business Model: Custom Build• Evolution: Differentiator• Domain Roles: Execution Context	Domain Roles
Inbound Communication  <p>The diagram shows a Supplier sending a message 'ScanProduct(QRImage)' to a central Ubiquitous Language box. The Ubiquitous Language box contains a list of terms: Product code, AR scan, Recognition status, and Validated product. It also contains a section titled 'Business Decisions' with rules: Only scanned or validated products can be sold and A fallback must exist for failed scans. The Ubiquitous Language box then sends messages back to the Supplier ('SearchProductByCode(code)') and to an Outbound Communication box.</p>	Outbound Communication  <p>The Outbound Communication box receives messages from the Ubiquitous Language box: 'ProductRecognized' (sent to Sales) and 'RecognitionFailed' (sent to Inventory). It also sends a message 'ProductDataRetuned' back to the Ubiquitous Language box.</p>	
Assumptions Product packaging includes scannable QR or is visually distinct Database of products is synchronized and up to date	Verification Metrics Number of scans per session Scan success rate Most scanned products	Open Questions Should the system support offline product recognition?

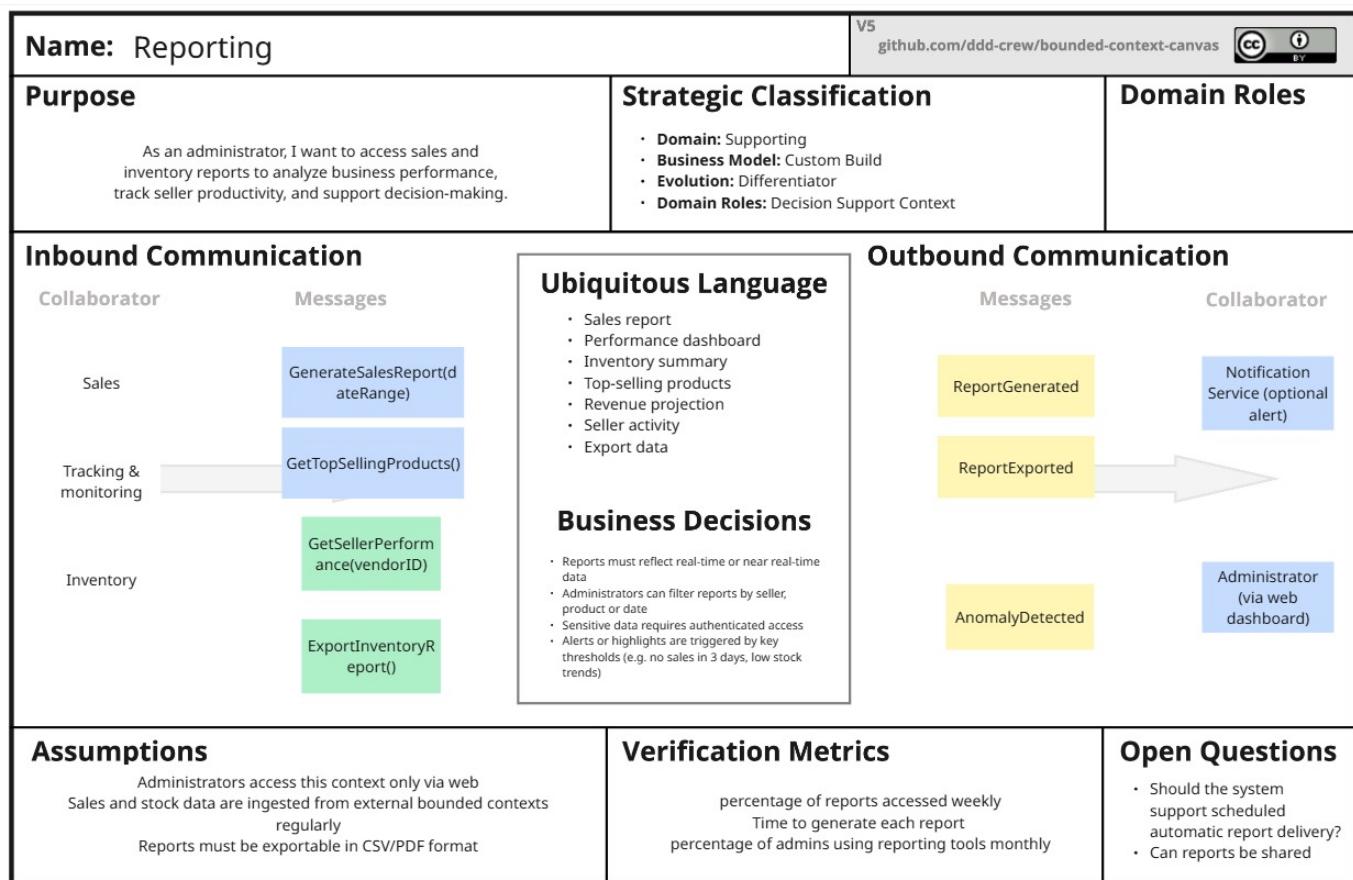
- Tracking & Monitoring:

Name: Tracking & Monitoring		V5 github.com/ddd-crew/bounded-context-canvas 
Purpose Provide admins with real-time GPS tracking and behavior analysis of suppliers in the field.	Strategic Classification <ul style="list-style-type: none">• Domain: Core Domain• Business Model: Custom Build• Evolution: Differentiator• Domain Roles: Execution Context	Domain Roles
Inbound Communication  <p>The diagram shows an Admin sending a message 'StartTracking(sellerID)' to a central Ubiquitous Language box. The Ubiquitous Language box contains a list of terms: Route, GPS ping, Active session, Deviation, and Zone. It also contains a section titled 'Business Decisions' with rules: Admins can define "valid routes" or "delivery zones", Excessive inactivity triggers alerts, and Tracking must be paused during breaks or disconnection. The Ubiquitous Language box then sends messages back to the Admin ('GetLiveLocation(sellerID)') and to an Outbound Communication box.</p>	Outbound Communication  <p>The Outbound Communication box receives messages from the Ubiquitous Language box: 'LocationTracked' (sent to Reporting), 'VendorOffRoute' (sent to Notification Service), and 'TrackingStopped' (sent to Notification Service).</p>	
Assumptions GPS service is enabled and available on the supplier's device Routes are updated in real time	Verification Metrics Number of daily tracking sessions Off-route alerts triggered Tracking success vs. signal loss	Open Questions Should vendors be notified when they're being tracked?

- Notifications:



- Reporting:



4.1.2 Context Mapping

Después de realizar la técnica de **EventStorming**, se identificaron los siguientes **4 Bounded Contexts**:

- **Stock Management**

Encargado de la gestión automática de inventario mediante reconocimiento de objetos usando modelos de TensorFlow.

- **Sales Tracking**

Encargado de registrar, visualizar y analizar las ventas realizadas a lo largo del día mediante la aplicación web.

- **User Management**

Responsable de la creación, autenticación y gestión de usuarios en el sistema.

- **Product Catalog**

Encargado de almacenar y administrar información detallada de los productos (nombre, descripción, precio, imagen de referencia, etc).

Se ha decidido utilizar los patrones de **Conformist** y **Customer/Supplier** para definir la relación entre los Bounded Contexts:

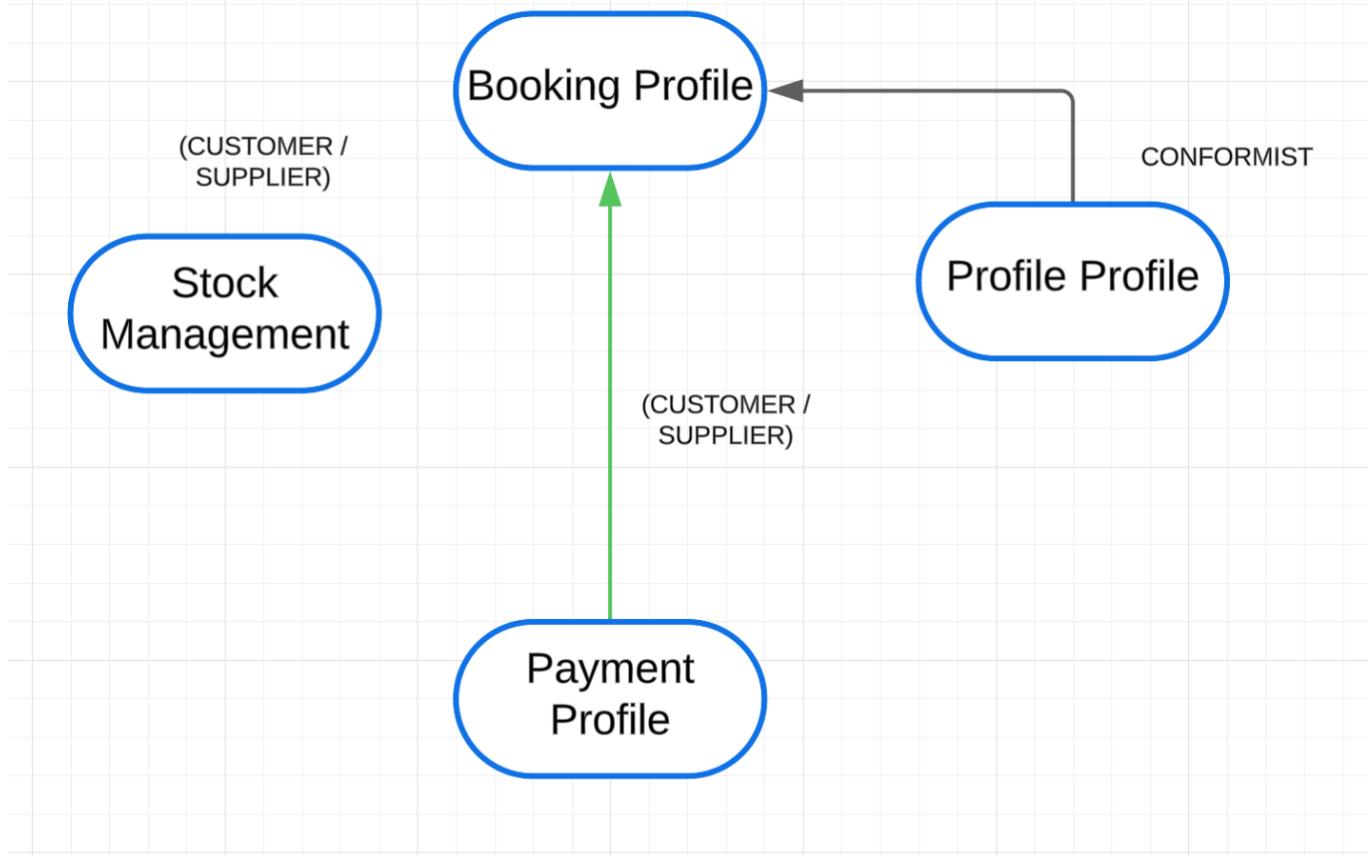
Conformist

- **Definición:**

Los Bounded Contexts tienen modelos de dominio similares y pueden adaptarse fácilmente entre sí sin comprometer la integridad de sus modelos. Se adopta un enfoque conformista cuando no se requiere modificar el modelo del proveedor.

- **Uso en el sistema:**

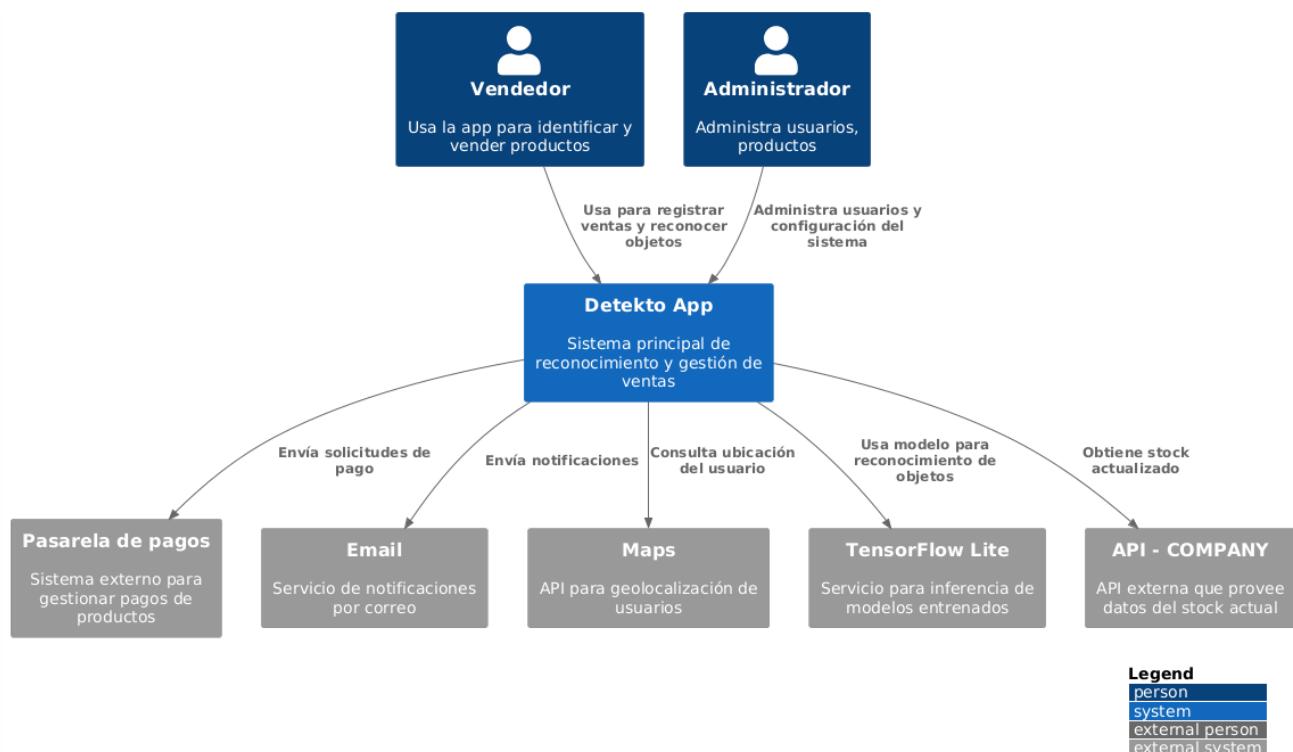
El Bounded Context de **Sales Tracking** necesita acceder a datos de productos gestionados por **Product Catalog** (por ejemplo, nombre del producto, precio) para poder registrar ventas correctamente. Dado que el modelo de producto es estable y compatible, **Sales Tracking** actúa como **Conformist** respecto a **Product Catalog**.



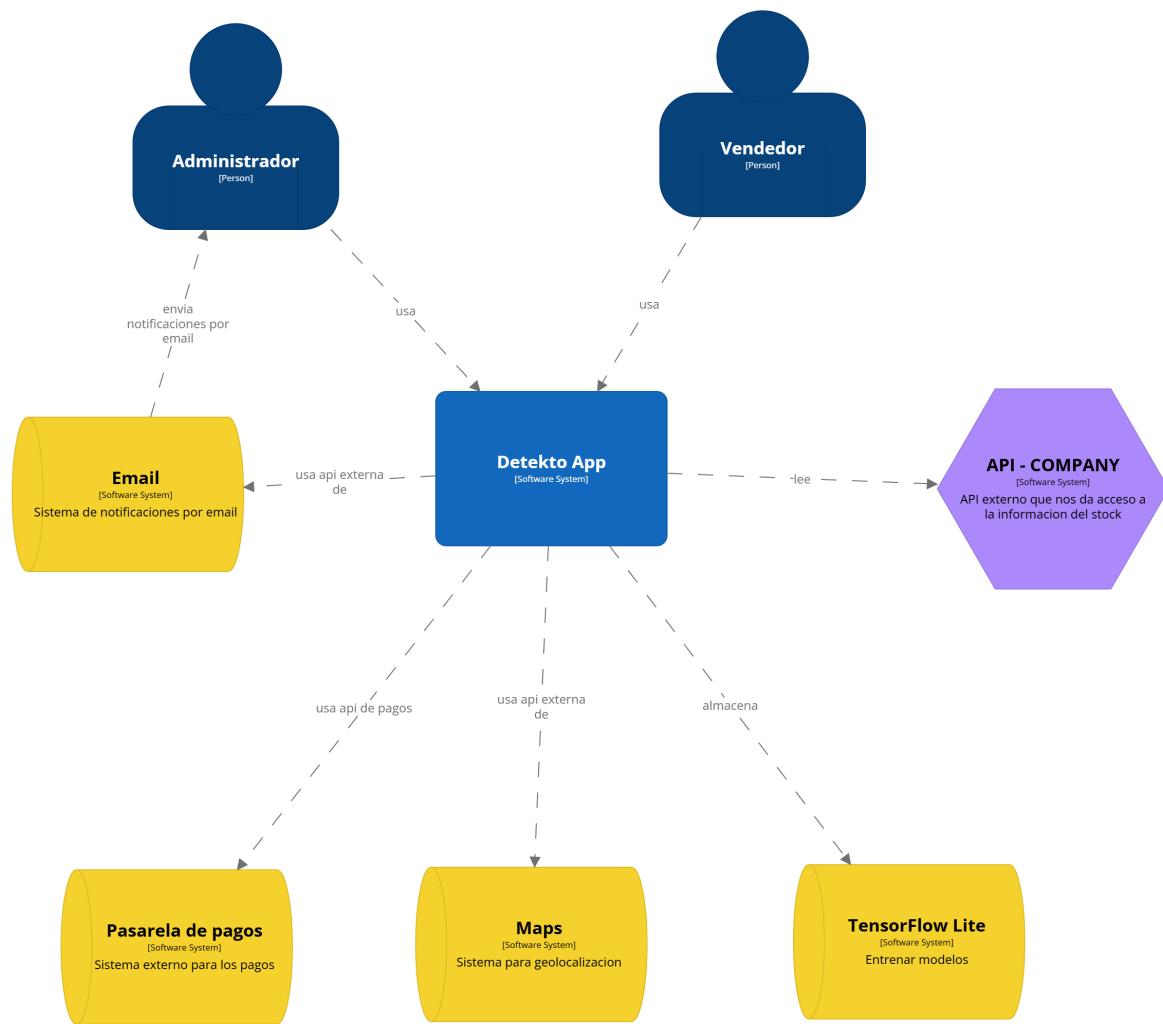
Link del LucidChart: https://lucid.app/lucidchart/ebfb06fe-73a6-4849-b4dc-4d220b98fe36/edit?viewport_loc=-1926%2C-1255%2C1933%2C1013%2C0_0&invitationId=inv_c91eb06c-12ca-4864-a806-19eaf5e528d8

4.3. Software Architecture.

4.3.1. Software Architecture System Landscape Diagram.

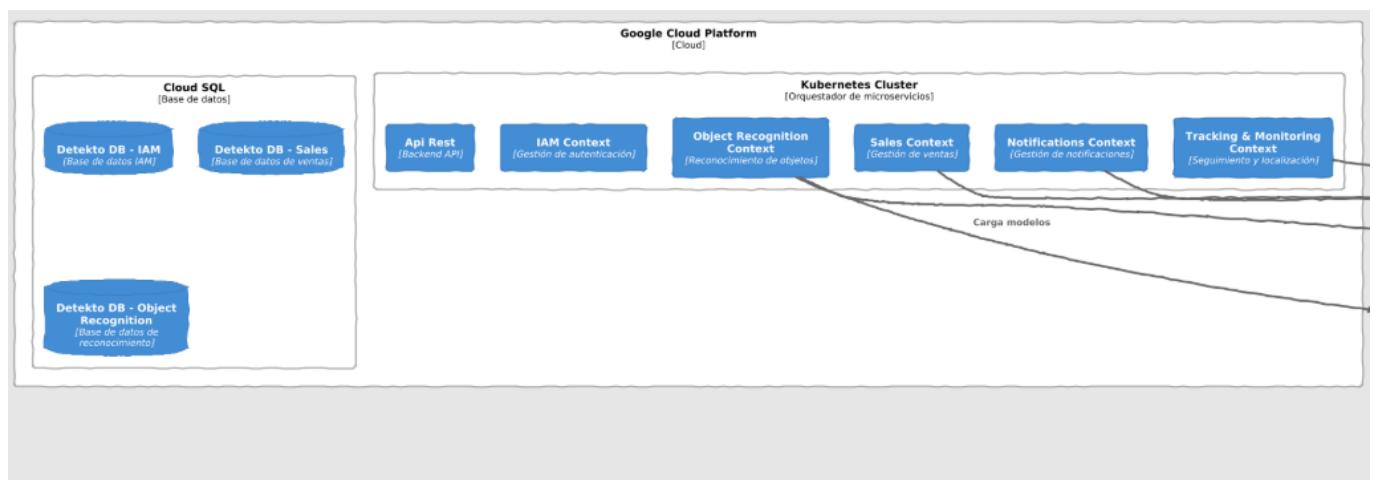


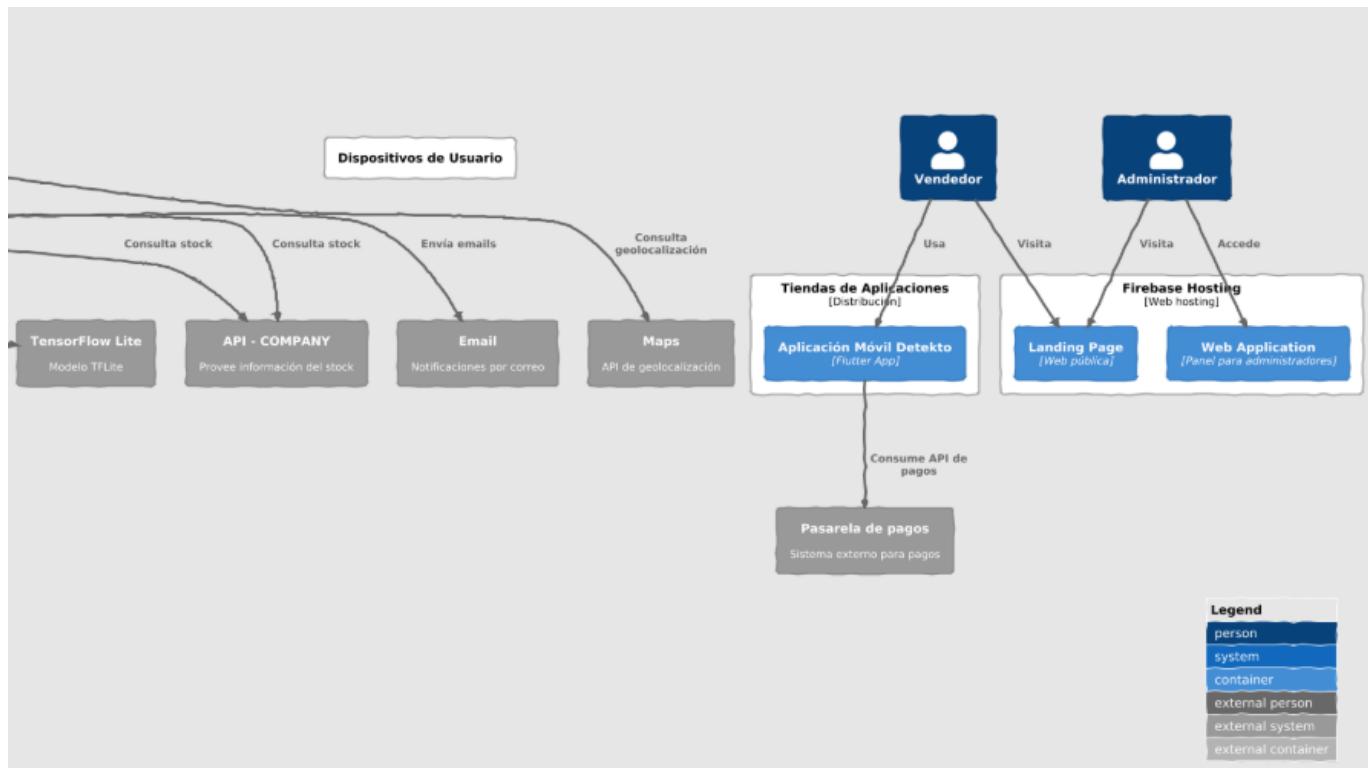
4.3.2. Software Architecture Context Level Diagrams.



[System Context] Detekto App
 Saturday, May 17, 2025 at 12:16 PM Peru Standard Time

4.3.3. Software Architecture Deployment Diagrams.





Capítulo V: Tactical-Level Software Design.

5.1. Bounded Context: IAM

5.1.1. Domain Layer.

En la capa de dominio del contexto de IAM de la aplicación, se definen las entidades del usuario y roles. El usuario representa a diferentes tipos de usuarios que interactúan con la aplicación. Los roles definen los permisos y privilegios de cada usuario.

Aggregate 1: Usuario

Nombre	Categoría	Propósito
Usuario	Entidad	Representa a un usuario de la aplicación de Detekto, y contiene información sobre la identidad y los roles del usuario.

Atributos del Usuario

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único del usuario.
nombre	String	Privado	Nombre del usuario.
email	String	Privado	Correo electrónico del usuario.
password	String	Privado	Contraseña del usuario.
fechaCreacion	DateTime	Privado	Fecha de creación del usuario.

Metodos del Usuario

Nombre	Tipo de retorno	Visibilidad	Descripción
Constructor	Void	Public	Constructor de la clase Usuario.
authenticateUser	Boolean	Public	Autentica al usuario utilizando su correo electrónico y contraseña.

Aggregate 2: Role

Nombre	Categoría	Propósito
Role	Entidad	Representa un rol de usuario en la aplicación de Detekto, y contiene información sobre los permisos asociados a ese rol.

Atributos del Role

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único del rol.
nombre	String	Privado	Nombre del rol.
permisos	List	Privado	Lista de permisos asociados al rol.

Metodos del Role

Nombre	Tipo de retorno	Visibilidad	Descripción
Constructor	Void	Public	Constructor de la clase Role.
addPermission	Void	Public	Agrega un permiso al rol.
getPermissions	List	Public	Devuelve la lista de permisos asociados al rol.

5.1.2. Interface Layer.

En la capa de interfaz del contexto de IAM de la aplicación, se definen los controladores y servicios que manejan las solicitudes y respuestas de la API. Los controladores son responsables de recibir las solicitudes HTTP y devolver las respuestas correspondientes.

Controller 1: UsersController

Nombre	Categoría	Propósito
UsersController	Controlador	Maneja las solicitudes relacionadas con los usuarios, como la autenticación y la creación de nuevos usuarios.

Atributos del UsersController

Nombre	Tipo de dato	Visibilidad	Descripción
userService	UserService	Privado	Servicio que maneja la lógica de negocio relacionada con los usuarios.
roleService	RoleService	Privado	Servicio que maneja la lógica de negocio relacionada con los roles.

Metodos del UsersController

Nombre	Tipo de retorno	Visibilidad	Descripción
authenticateUser	ResponseEntity	Public	Metodo para el inicio de sesion de un usuario usando credenciales y devuelve un token JWT.
registerUser	ResponseEntity	Public	Metodo para registrar un nuevo usuario en la aplicacion.
getAllUsers	ResponseEntity	Public	Metodo para obtener todos los usuarios registrados en la aplicacion.
getUserById	ResponseEntity	Public	Metodo para obtener un usuario por su ID.
updateUser	ResponseEntity	Public	Metodo para actualizar la informacion de un usuario.
deleteUser	ResponseEntity	Public	Metodo para eliminar un usuario de la aplicacion.
getUserRoles	ResponseEntity	Public	Metodo para obtener los roles de un usuario.

Controller 2: RolesController

Nombre	Categoría	Proposito
RolesController	Controlador	Maneja las solicitudes relacionadas con los roles, como la creación y asignación de roles a usuarios.

Atributos del RolesController

Nombre	Tipo de dato	Visibilidad	Descripción
roleService	RoleService	Privado	Servicio que maneja la lógica de negocio relacionada con los roles.
userService	UserService	Privado	Servicio que maneja la lógica de negocio relacionada con los usuarios.

Metodos del RolesController

Nombre	Tipo de retorno	Visibilidad	Descripción
getAllRoles	ResponseEntity	Public	Metodo para obtener todos los roles disponibles en la aplicacion.
assignRoleToUser	ResponseEntity	Public	Metodo para asignar un rol a un usuario.
removeRoleFromUser	ResponseEntity	Public	Metodo para eliminar un rol de un usuario.

5.1.3. Application Layer.

En la capa de aplicacion del contexto de IAM de la aplicacion, se definen los servicios que manejan la logica de negocio relacionada con los usuarios y roles. Estos servicios son utilizados por los controladores para realizar las operaciones necesarias.

Service 1: UserService

Nombre	Categoría	Propósito
UserService	Servicio	Maneja la logica de negocio relacionada con los usuarios, como la autenticacion y la gestion de usuarios.

Atributos del UserService

Nombre	Tipo de dato	Visibilidad	Descripción
userRepository	UserRepository	Privado	Repositorio que maneja la persistencia de los usuarios.
passwordEncoder	PasswordEncoder	Privado	Codificador de contraseñas para la autenticacion de usuarios.
jwtHandler	JwtHandler	Privado	Manejador de tokens JWT para la autenticacion de usuarios.

Metodos del UserService

Nombre	Tipo de retorno	Visibilidad	Descripción
authenticateUser	User	Public	Autentica a un usuario utilizando su correo electronico y contraseña.
registerUser	User	Public	Registra un nuevo usuario en la aplicacion.
getAllUsers	List	Public	Devuelve todos los usuarios registrados en la aplicacion.
updateUser	User	Public	Actualiza la informacion de un usuario.
deleteUser	Void	Public	Elimina un usuario de la aplicacion.

Nombre	Tipo de retorno	Visibilidad	Descripción
getUserById	User	Public	Devuelve un usuario por su ID.
getUserByEmail	User	Public	Devuelve un usuario por su correo electronico.
getUserRoles	List	Public	Devuelve los roles de un usuario.
assignRoleToUser	Void	Public	Asigna un rol a un usuario.

Service 2: RoleService

Nombre	Categoría	Proposito
RoleService	Servicio	Maneja la logica de negocio relacionada con los roles, como la creacion y asignacion de roles a usuarios.

Atributos del RoleService

Nombre	Tipo de dato	Visibilidad	Descripción
roleRepository	RoleRepository	Privado	Repositorio que maneja la persistencia de los roles.

Metodos del RoleService

Nombre	Tipo de retorno	Visibilidad	Descripción
getAllRoles	List	Public	Devuelve todos los roles disponibles en la aplicacion.
assignRoleToUser	Void	Public	Asigna un rol a un usuario.
removeRoleFromUser	Void	Public	Elimina un rol de un usuario.

5.1.4. Infrastructure Layer.

En la capa de infraestructura del contexto de IAM de la aplicacion, se definen los repositorios que manejan la persistencia de los usuarios y roles. Estos repositorios son utilizados por los servicios para realizar las operaciones necesarias.

Repository 1: UserRepository

Nombre	Categoría	Proposito
UserRepository	Repositorio	Maneja la persistencia de los usuarios en la base de datos.

Metodos del UserRepository

Nombre	Tipo de retorno	Visibilidad	Descripción

Nombre	Tipo de retorno	Visibilidad	Descripción
findByEmail	User	Public	Busca un usuario por su correo electronico.
findById	User	Public	Busca un usuario por su ID.
existsByEmail	Boolean	Public	Verifica si un usuario existe por su correo electronico.
save	User	Public	Guarda un usuario en la base de datos.
delete	Void	Public	Elimina un usuario de la base de datos.

Repository 2: RoleRepository

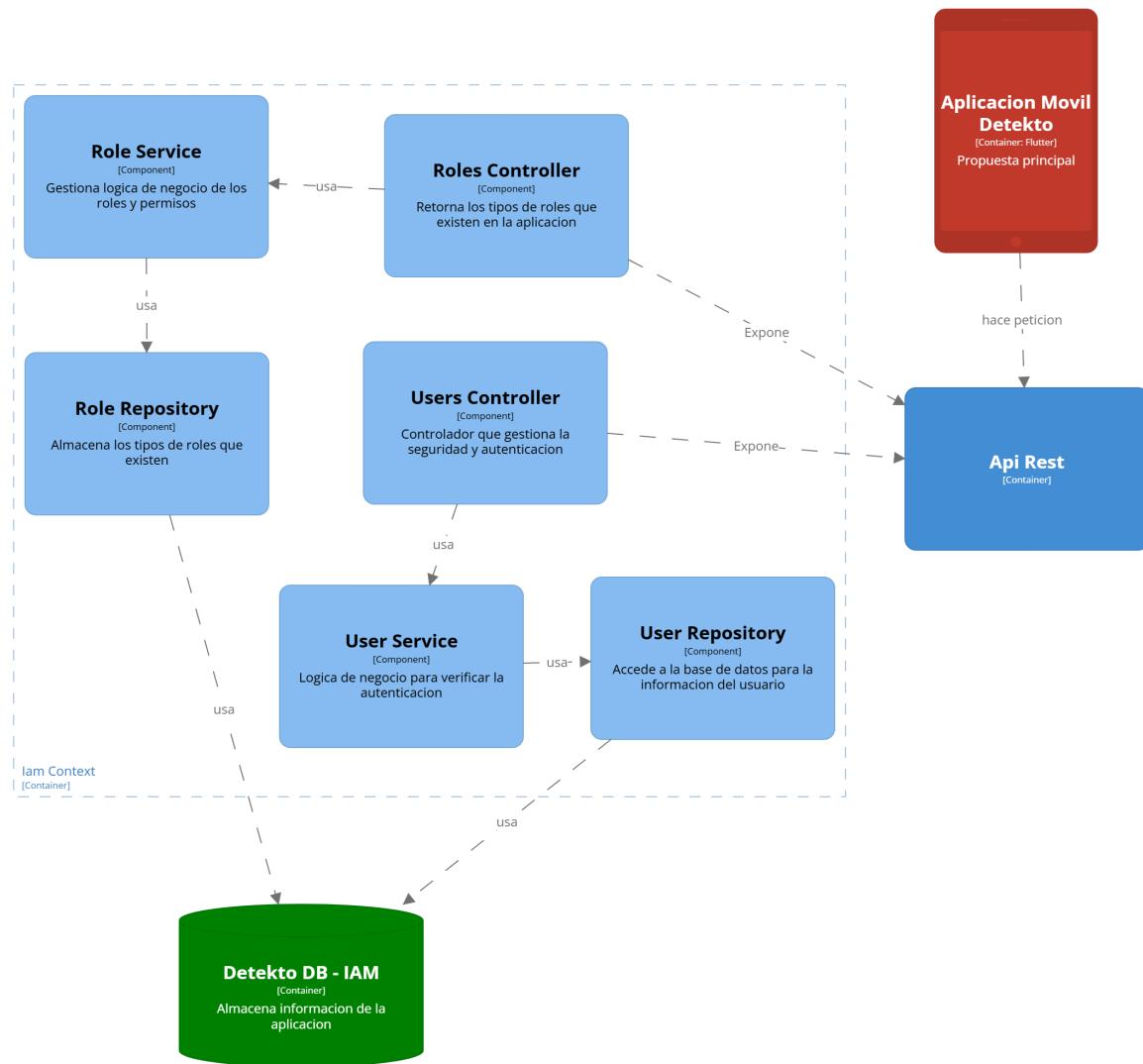
Nombre	Categoría	Propósito
RoleRepository	Repositorio	Maneja la persistencia de los roles en la base de datos.

Metodos del RoleRepository

Nombre	Tipo de retorno	Visibilidad	Descripción
findById	Role	Public	Busca un rol por su ID.
findByName	Role	Public	Busca un rol por su nombre.
save	Role	Public	Guarda un rol en la base de datos.
deleteById	Void	Public	Elimina un rol de la base de datos.

5.1.6. Bounded Context Software Architecture Component Level Diagrams.

Esta sección presenta los diagramas de componentes de la arquitectura de software del contexto de IAM. Estos diagramas muestran la estructura y las relaciones entre los diferentes componentes del sistema.



[Component] Detekto App - Iam Context
 Saturday, May 17, 2025 at 12:16 PM Peru Standard Time

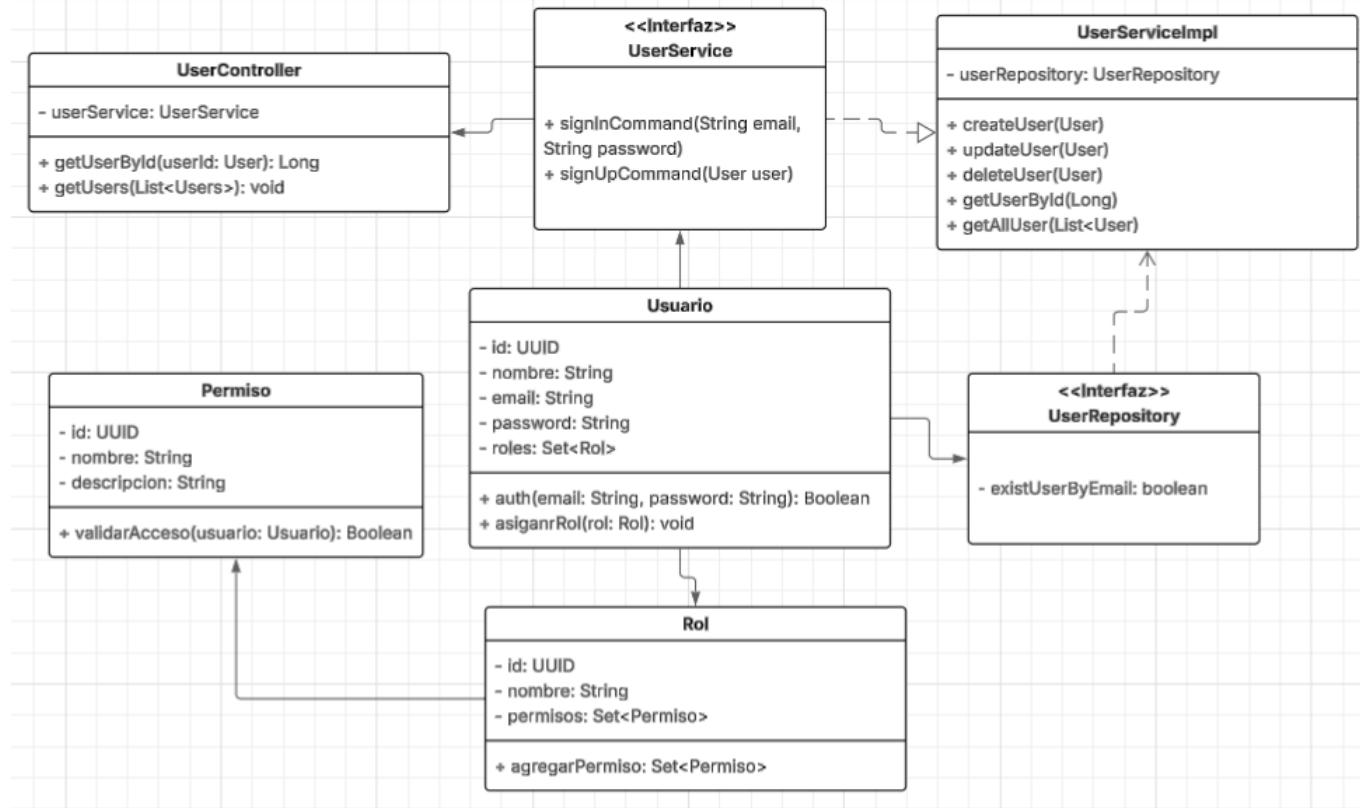
5.1.7. Bounded Context Software Architecture Code Level Diagrams.

En esta sección, el equipo de Softwarinos presenta la implementación de los componentes dentro de cada contexto.

- **Domain Layer Class Diagrams:** Muestra la estructura de las clases y sus relaciones en el contexto de IAM.
- **Database Design Diagram:** Presenta el diseño de la base de datos, incluyendo las tablas y sus relaciones.

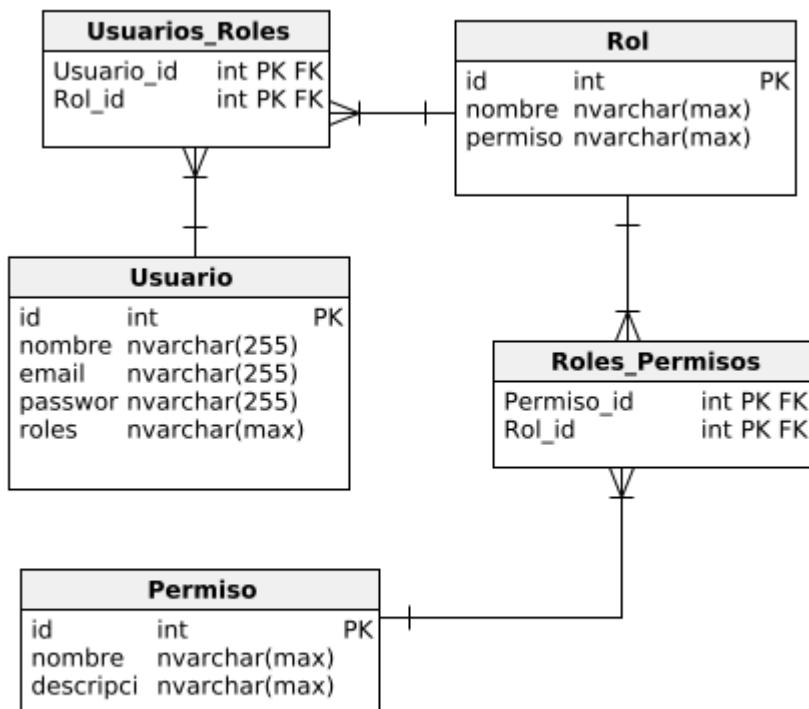
5.1.7.1. Bounded Context Domain Layer Class Diagrams.

En esta sección se presentan los diagramas de clases del contexto de IAM, en el que se muestran las entidades claves para la autenticación del usuario, los roles, junto la relación que tienen los mismos.



5.1.7.2. Bounded Context Database Design Diagram.

En esta sección, el diagrama de base de datos nos muestra la estructura de las tablas y sus relaciones en el contexto de IAM. Este diagrama es fundamental para entender cómo se almacenan y gestionan los datos en la aplicación.



5.2. Bounded Context: OBJECT RECOGNITION

5.2.1. Domain Layer.

En la capa de dominio del contexto de Object-Recognition de la aplicación, se definen las entidades centrales que permiten registrar y gestionar los datos generados por el reconocimiento visual de herramientas y materiales. Este contexto no realiza inferencias directamente, sino que registra los resultados obtenidos desde la app móvil y permite el seguimiento de versiones del modelo de reconocimiento.

Aggregate 1: ImageRecognition

Nombre	Categoría	Propósito
ImageRecognition	Entidad	Representa una imagen escaneada por la app móvil que ha sido identificada mediante reconocimiento visual. Guarda el resultado, la confianza del modelo, y el producto reconocido.

Atributos del ImageRecognition

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único de la imagen reconocida.
nombreArchivo	String	Privado	Nombre del archivo de la imagen escaneada.
resultado	String	Privado	Resultado del reconocimiento visual (Ej: "Destornillador").
fechaEscaneo	LocalDateTime	Privado	Fecha y hora en que se realizó el escaneo.
productoid	UUID	Privado	Identificador al producto reconocido (referencia al contexto externo SALES).
estado	EstadoReconocimiento	Privado	Estado del reconocimiento (Ej: "Pendiente", "Exitoso", "Fallido").

Métodos del ImageRecognition

Nombre	Tipo de retorno	Visibilidad	Descripción
Constructor	Void	Public	Constructor de la clase ImageRecognition.
asociarProducto	Void	Public	Asocia un producto al reconocimiento de la imagen.
actualizarEstado	Void	Public	Actualiza el estado del reconocimiento.

Aggregate 2: VersionModeloReconocimiento

Nombre	Categoría	Propósito
--------	-----------	-----------

Nombre	Categoría	Propósito
VersionModeloReconocimiento	Entidad	Representa una versión del modelo de reconocimiento visual utilizado en la app Detekto, incluyendo información sobre la fecha de implementación y el estado del modelo.

Atributos del VersionModeloReconocimiento

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único de la versión del modelo.
version	String	Privado	Número de versión del modelo (Ej: "1.0.0").
rutaModelo	String	Privado	Ruta del modelo .tflite utilizado para el reconocimiento.
fechaimplementacion	LocalDateTime	Privado	Fecha y hora en que se implementó la versión del modelo.

Métodos del VersionModeloReconocimiento

Nombre	Tipo de retorno	Visibilidad	Descripción
Constructor	Void	Public	Constructor de la clase VersionModeloReconocimiento.
esVersionActual	Boolean	Public	Verifica si la versión del modelo es la actual.

Enum 1: EstadoReconocimiento

Nombre	Categoría	Propósito
EstadoReconocimiento	Enumeración	Enum que representa los posibles estados de un reconocimiento visual.

Valores del Enum

- PENDIENTE:** El reconocimiento está pendiente de ser procesado.
- EXITOSO:** El reconocimiento se realizó con éxito y se obtuvo un resultado.
- FALLIDO:** El reconocimiento falló y no se obtuvo un resultado.

5.2.2. Interface Layer.

En esta capa se define el controlador encargado de recibir solicitudes relacionadas con el registro y gestión de imágenes reconocidas. Aunque el reconocimiento ocurre en la app móvil, el backend permite guardar y auditar estos eventos.

Controller: ReconocimientoController

Nombre	Categoría	Propósito
ReconocimientoController	Controlador	Maneja las solicitudes HTTP relacionadas con la recepción de imágenes reconocidas desde la app móvil.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
registrarImagen	ResponseEntity	Público	Recibe una imagen reconocida y la registra en el sistema.
obtenerPorProducto	ResponseEntity	Público	Devuelve las imágenes asociadas a un determinado producto.
obtenerEstadisticas	ResponseEntity	Público	Devuelve estadísticas sobre los resultados del reconocimiento.

5.2.3. Application Layer.

En esta capa se orquesta la lógica del dominio. El servicio principal permite registrar imágenes y asociarlas con una versión del modelo.

Service: ReconocimientoService

Nombre	Categoría	Propósito
ReconocimientoService	Servicio	Encapsula la lógica de negocio para registrar imágenes reconocidas y asociarlas a modelos y productos.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
registrarImagen	ImagenReconocida	Público	Registra una imagen reconocida con su predicción y versión del modelo.
asociarProducto	Void	Público	Asocia un producto a una imagen previamente registrada.
actualizarEstado	Void	Público	Cambia el estado de una imagen reconocida.

Service: VersionModeloService

Nombre	Categoría	Propósito
ModelVersionService	Servicio	Maneja la lógica de negocio relacionada con el control de versiones del modelo de reconocimiento visual.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
registrarNuevaVersion	VersionModeloReconocimiento	Público	Registra una nueva versión del modelo.
establecerComoActual	Void	Público	Marca una versión como la actual.
obtenerVersionActual	VersionModeloReconocimiento	Público	Devuelve la versión actual en uso.
listarVersiones	List	Público	Retorna todas las versiones del modelo.

5.2.4. Infrastructure Layer.

Esta capa se encarga de manejar la persistencia de las entidades del dominio, así como integrarse con sistemas externos si se requiere almacenar imágenes o versiones del modelo.

Repository: ImagenReconocidaRepository

Nombre	Categoría	Propósito
ImagenReconocidaRepository	Repositorio	Accede a la base de datos para guardar y recuperar imágenes reconocidas.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
save	ImagenReconocida	Público	Guarda una nueva imagen reconocida.
findByProductid	List	Público	Recupera imágenes por ID de producto.

Repository: VersionModeloRepository

Nombre	Categoría	Propósito
VersionModeloRepository	Repositorio	Accede a las versiones de los modelos cargados.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
save	VersionModeloReconocimiento	Público	Guarda una nueva versión del modelo.
findActual	VersionModeloReconocimiento	Público	Recupera la versión del modelo marcada como actual.

5.2.6. Bounded Context Software Architecture Component Level Diagrams.

Esta sección presenta los diagramas de componentes de la arquitectura de software del contexto de **OBJECT RECOGNITION**. Estos diagramas muestran la estructura y las relaciones entre los diferentes componentes del sistema.



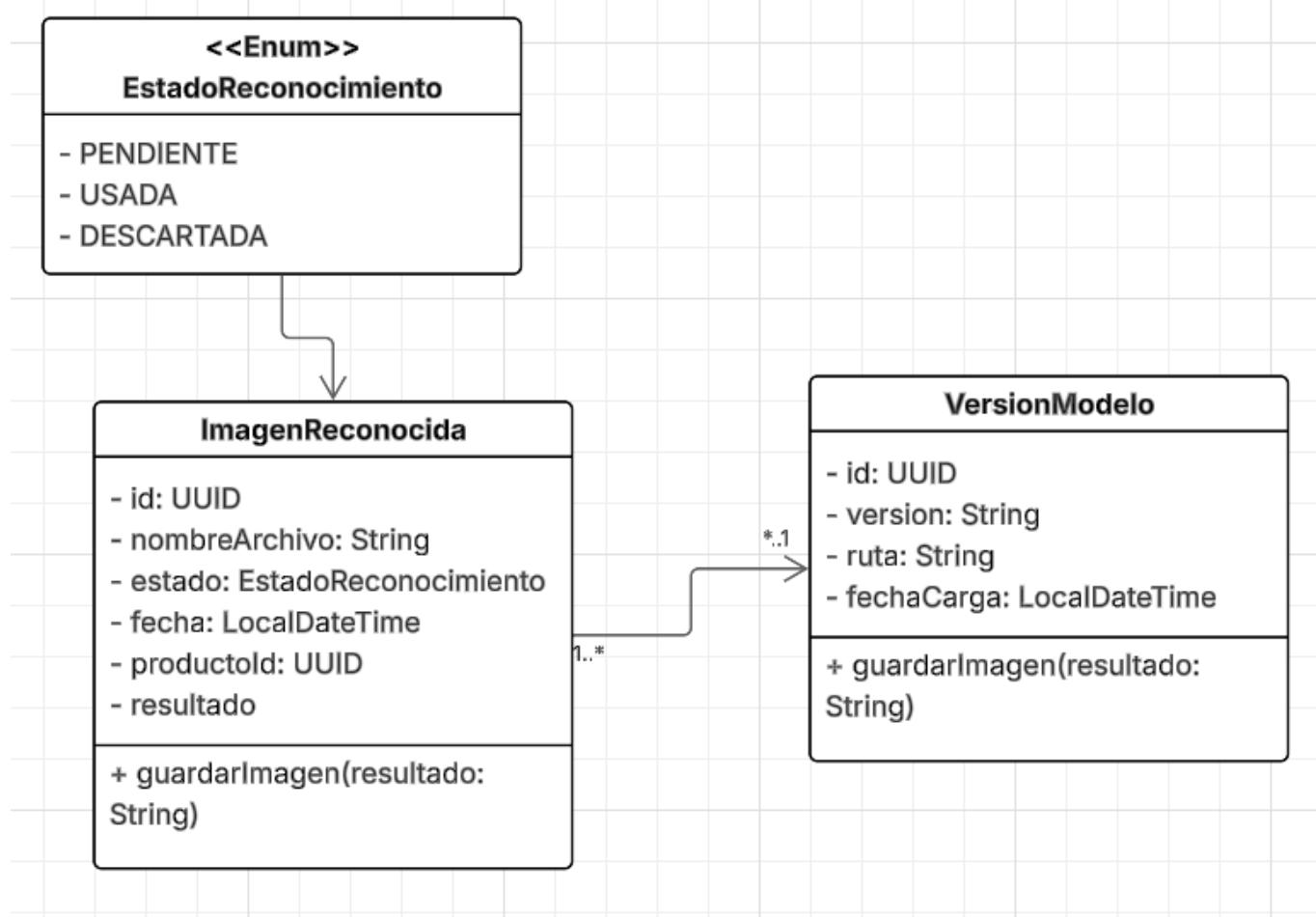
[Component] Detekto App - Object Recognition Context
Saturday, May 17, 2025 at 12:16 PM Peru Standard Time



5.2.7. Bounded Context Software Architecture Code Level Diagrams.

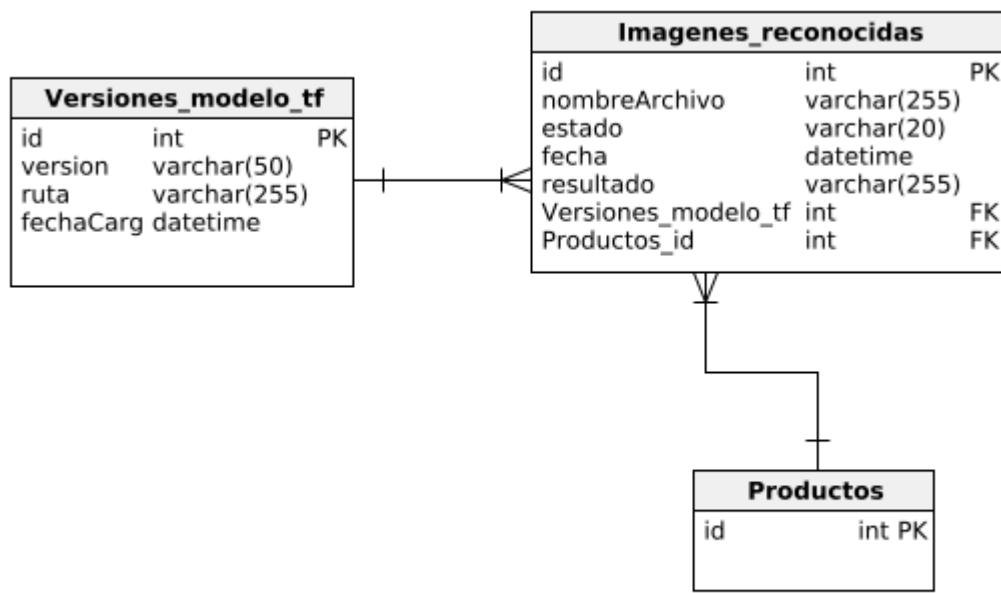
5.2.7.1. Bounded Context Domain Layer Class Diagrams.

En esta sección se presentan los diagramas de clases del contexto de OBJECT RECOGNITION, en el que se muestran las entidades claves para la gestión del reconocimiento de objetos, junto la relación que tienen los mismos.



5.2.7.2. Bounded Context Database Design Diagram.

En esta sección, el diagrama de base de datos nos muestra la estructura de las tablas y sus relaciones en el contexto de OBJECT RECOGNITION. Este diagrama es fundamental para entender cómo se almacenan y gestionan los datos en la aplicación.



5.3. Bounded Context: SALES

5.3.1. Domain Layer.

En la capa de dominio del contexto de SALES de la aplicación, se definen las entidades principales que permiten registrar y gestionar las ventas realizadas por los usuarios.

Una venta contiene productos, cantidades, totales y la información del cliente que realizó la compra.

Aggregate 1: Sale

Nombre	Categoría	Propósito
Sale	Entidad	Representa una venta realizada dentro de la aplicación, incluyendo la fecha, los productos vendidos, el cliente y el estado de la transacción.

Atributos del Sale

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único de la venta.
fecha	DateTime	Privado	Fecha en la que se realizó la venta.
total	Decimal	Privado	Monto total de la venta.
metodoPago	String	Privado	Medio de pago utilizado por el cliente.
usuarioid	UUID	Privado	ID del usuario que realizó la venta.
estado	EstadoVenta	Privado	Estado actual de la venta.
items	List	Privado	Lista de productos vendidos en la transacción.

Métodos del Sale

Nombre	Tipo de retorno	Visibilidad	Descripción
calcularTotal	Decimal	Public	Calcula el monto total de la venta.
marcarComoPagada	Void	Public	Cambia el estado de la venta a "PAGADA".
cancelar	Void	Public	Cancela la venta cambiando su estado.

Aggregate 2: Product

Nombre	Categoría	Propósito
Product	Entidad	Representa un producto incluido en una venta, con su cantidad y precio unitario.

Atributos del Product

Nombre	Tipo de dato	Visibilidad	Descripción

Nombre	Tipo de dato	Visibilidad	Descripción
productold	UUID	Privado	ID del producto vendido.
cantidad	Int	Privado	Cantidad vendida del producto.
precioUnitario	Decimal	Privado	Precio por unidad del producto en la venta.

Métodos del Product

Nombre	Tipo de retorno	Visibilidad	Descripción
calcularSubtotal	Decimal	Public	Calcula el subtotal del ítem (cantidad * precio).

Enumerado: EstadoVenta

Nombre	Tipo	Propósito
EstadoVenta	Enum	Define el estado de una venta dentro del sistema.

Valores posibles:

- PENDIENTE
- PAGADA
- CANCELADA

5.3.2. Interface Layer.

En la capa de interfaz del contexto de SALES de la aplicación, se definen los controladores y servicios que manejan las solicitudes y respuestas de la API. Los controladores son responsables de recibir las solicitudes HTTP y devolver las respuestas correspondientes.

Controller 1: SalesController

Nombre	Categoría	Propósito
SalesController	Controlador	Expone endpoints públicos para la gestión de ventas.

Atributos del SalesController

Nombre	Tipo de dato	Visibilidad	Descripción
saleService	SaleService	Privado	Servicio que maneja la lógica de negocio de las ventas.

Metodos del SalesController

Nombre	Tipo de retorno	Visibilidad	Descripción
registerSale	ResponseType	Public	Método para registrar una nueva venta en el sistema.

Nombre	Tipo de retorno	Visibilidad	Descripción
getAllSales	ResponseEntity	Public	Método para obtener todas las ventas registradas en la aplicación.
getSaleById	ResponseEntity	Public	Método para obtener los detalles de una venta por su ID.
deleteSale	ResponseEntity	Public	Método para eliminar una venta existente.
getSalesByUserId	ResponseEntity	Public	Método para obtener todas las ventas realizadas por un usuario específico.

5.3.3. Application Layer.

En la capa de aplicación del contexto de IAM de la aplicación, se definen los servicios que manejan la lógica de negocio relacionada con los usuarios y roles. Estos servicios son utilizados por los controladores para realizar las operaciones necesarias.

Service 1: SalesService

Nombre	Categoría	Propósito
SalesService	Servicio	Maneja la lógica de negocio relacionada con el registro, consulta y gestión de ventas.

Atributos del SalesService

Nombre	Tipo de dato	Visibilidad	Descripción
salesRepository	SalesRepository	Privado	Repositorio que gestiona la persistencia de las ventas.

Métodos del SalesService

Nombre	Tipo de retorno	Visibilidad	Descripción
registerSale	Sale	Public	Registra una nueva venta en el sistema.
getAllSales	List	Public	Retorna todas las ventas registradas en la aplicación.
getSaleById	Sale	Public	Obtiene los detalles de una venta a partir de su ID.
deleteSale	Void	Public	Elimina una venta existente por su ID.
getSalesByUserId	List	Public	Devuelve las ventas asociadas a un usuario específico.

5.3.4. Infrastructure Layer.

En la capa de infraestructura del contexto de SALES de la aplicación, se definen los repositorios que manejan la persistencia de los usuarios y roles. Estos repositorios son utilizados por los servicios para realizar las operaciones necesarias.

Repository 1: SalesRepository

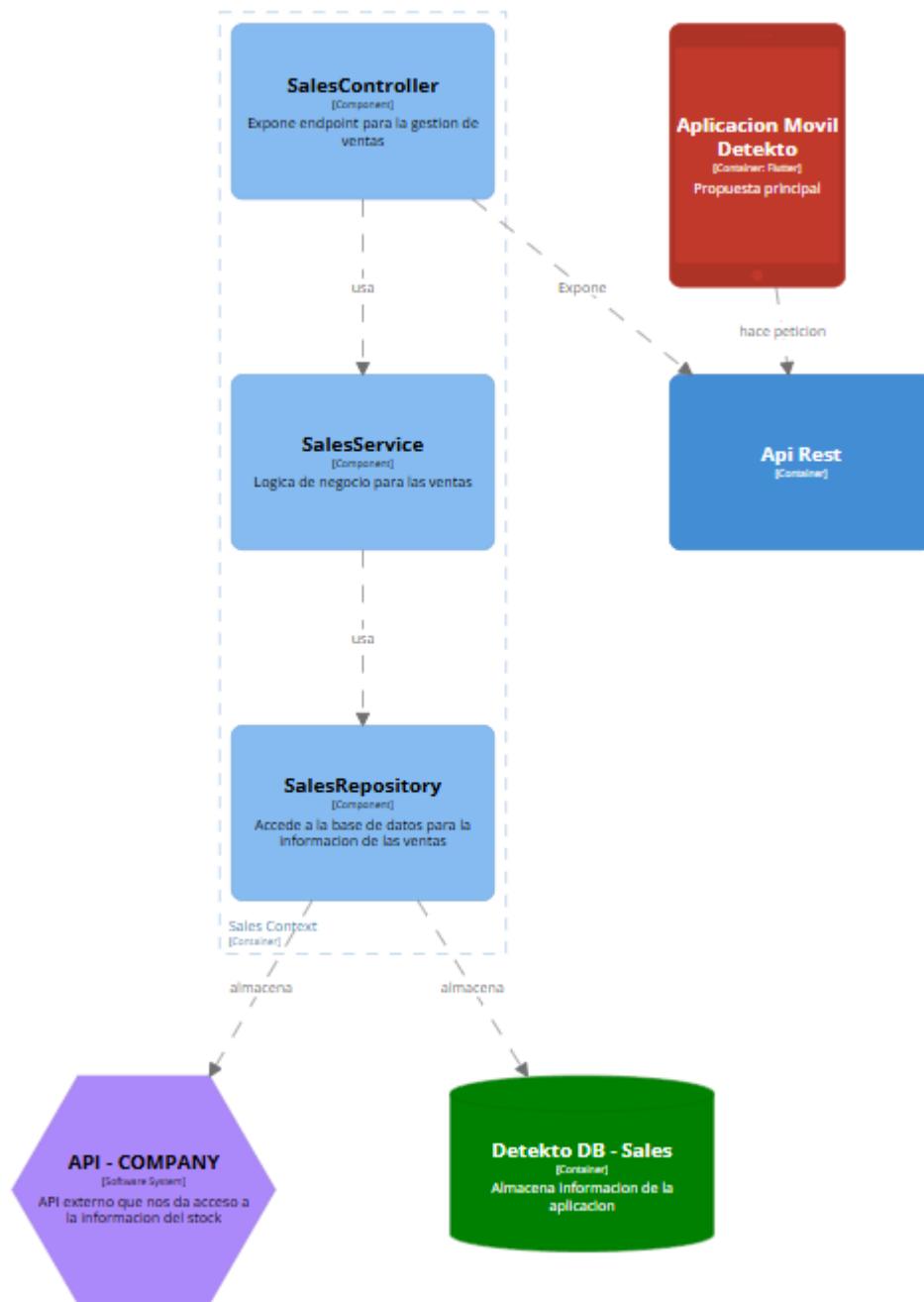
Nombre	Categoría	Propósito
SalesRepository	Repositorio	Maneja la persistencia de las ventas en la base de datos.

Métodos del SalesRepository

Nombre	Tipo de retorno	Visibilidad	Descripción
findAll	List	Public	Retorna todas las ventas almacenadas.
findById	Sale	Public	Busca una venta por su ID.
findByIdUser	List	Public	Devuelve las ventas asociadas a un usuario específico.
save	Sale	Public	Guarda una venta en la base de datos.
deleteById	Void	Public	Elimina una venta específica de la base de datos.

5.3.6. Bounded Context Software Architecture Component Level Diagrams.

Esta sección presenta los diagramas de componentes de la arquitectura de software del contexto de **SALES**. Estos diagramas muestran la estructura y las relaciones entre los diferentes componentes del sistema.



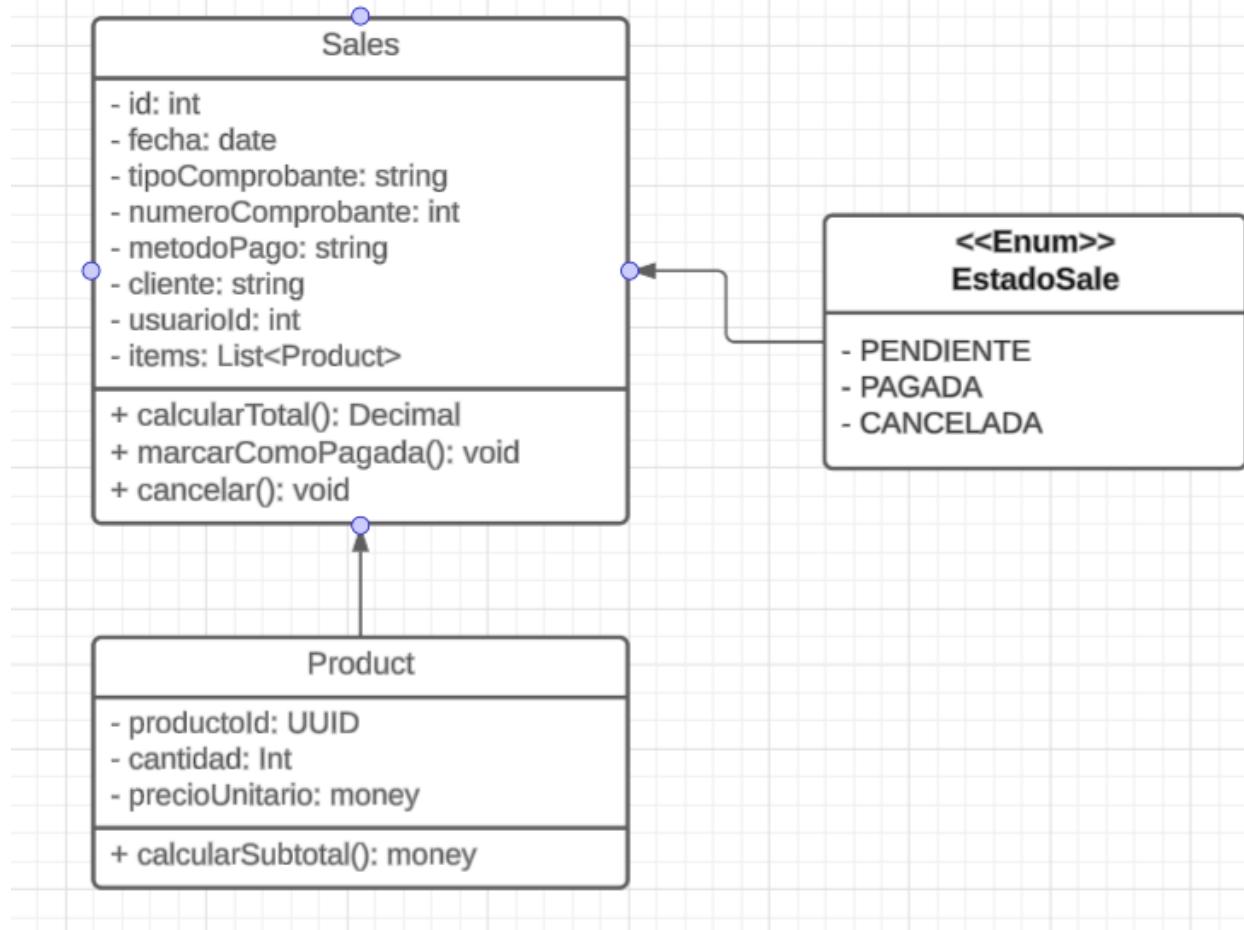
5.3.7. Bounded Context Software Architecture Code Level Diagrams.

En esta sección, el equipo de Softwarinos presenta la implementación de los componentes dentro de cada contexto.

- **Domain Layer Class Diagrams:** Muestra la estructura de las clases y sus relaciones en el contexto de SALES.
- **Database Design Diagram:** Presenta el diseño de la base de datos, incluyendo las tablas y sus relaciones.

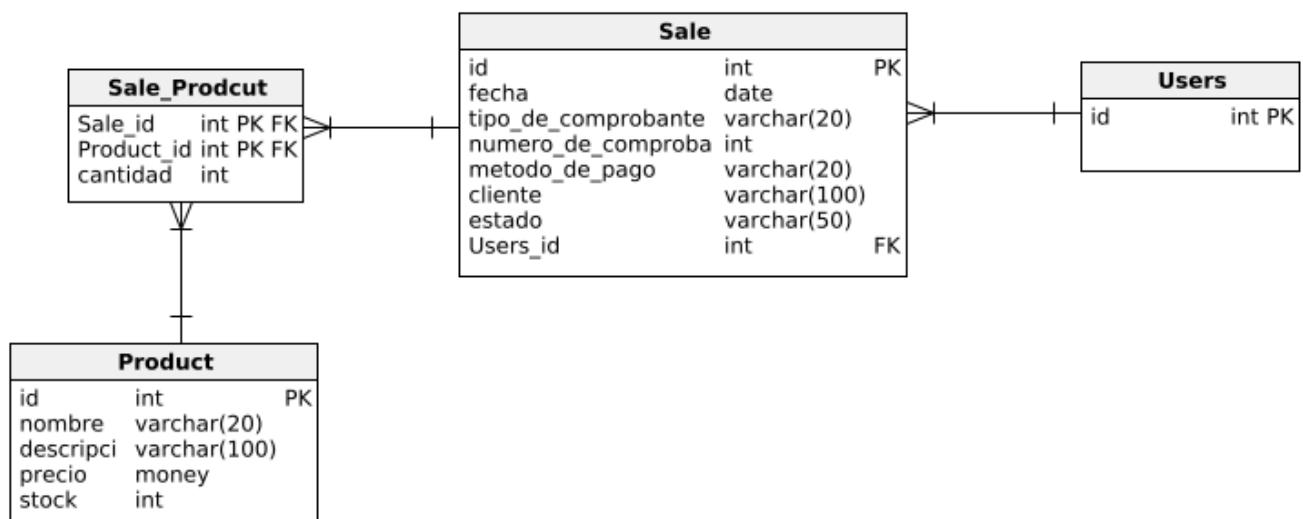
5.3.7.1. Bounded Context Domain Layer Class Diagrams.

En esta sección se presentan los diagramas de clases del contexto de SALES, en el que se muestran las entidades clave para la autenticación del usuario, los roles, junto la relación que tienen los mismos



5.3.7.2. Bounded Context Database Design Diagram.

En esta sección, el diagrama de base de datos nos muestra la estructura de las tablas y sus relaciones en el contexto de SALES. Este diagrama es fundamental para entender cómo se almacenan y gestionan los datos en la aplicación.



5.4. Bounded Context: NOTIFICATIONS

5.4.1. Domain Layer.

En la capa de dominio del contexto de Notifications, se definen las entidades y servicios que permiten gestionar el envío de notificaciones hacia los administradores o usuarios del sistema. Las notificaciones pueden enviarse a través de canales externos como el correo electrónico, y se lleva un registro de su estado.

Aggregate 1: Notificacion

Nombre	Categoría	Propósito
Notificacion	Entidad	Representa un mensaje dirigido a un usuario, el cual puede ser enviado mediante un canal externo como el correo electrónico.

Atributos

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único de la notificación.
destinatarioEmail	String	Privado	Correo del destinatario de la notificación.
asunto	String	Privado	Título o asunto de la notificación.
mensaje	String	Privado	Contenido del mensaje enviado.
fechaEnvio	LocalDateTime	Privado	Fecha y hora en la que se intentó enviar la notificación.
estado	EstadoNotificacion	Privado	Estado actual de la notificación (enviada, fallida, pendiente).

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
Constructor	Void	Público	Constructor de la clase.
marcarComoEnviada	Void	Público	Cambia el estado a ENVIADA.
marcarComoFallida	Void	Público	Cambia el estado a FALLIDA.

Enum: EstadoNotificacion

Nombre	Categoría	Propósito
EstadoNotificacion	Enumeración	Enum que representa el estado actual de una notificación enviada.

Valores del Enum

- ENVIADA:** La notificación fue enviada exitosamente.
- FALLIDA:** La notificación no pudo ser enviada.
- PENDIENTE:** La notificación está pendiente de envío.

5.4.2. Interface Layer.

En esta capa se define el controlador que expone los endpoints para consultar o generar notificaciones desde otros módulos de la aplicación.

Controller: NotificacionesController

Nombre	Categoría	Propósito
NotificacionesController	Controlador	Maneja las solicitudes relacionadas con la generación y visualización de notificaciones.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
enviarNotificacion	ResponseType	Público	Recibe los datos de una notificación y la envía al destinatario.
listarNotificaciones	ResponseType	Público	Devuelve la lista de notificaciones registradas.
obtenerPorEstado	ResponseType	Público	Devuelve las notificaciones según su estado.

5.4.3. Application Layer.

En esta capa se orquesta el envío de notificaciones, validando la lógica del negocio antes de delegar a la infraestructura el envío final.

Service: NotificacionService

Nombre	Categoría	Propósito
NotificacionService	Servicio	Se encarga de validar, registrar y gestionar el proceso de envío de una notificación.

Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
crearYEnviarNotificacion	Notificacion	Público	Registra una nueva notificación y gestiona su envío.
listarNotificaciones	List	Público	Devuelve la lista de notificaciones registradas.
obtenerPorEstado	List	Público	Devuelve notificaciones según su estado.

External Service Interface: EmailSender

Nombre	Categoría	Propósito

Nombre	Categoría	Propósito
EmailSender	Interface	Representa un proveedor externo de correo que permite el envío de notificaciones.

Métodos

Nombre	Tipo de retorno	Descripción
enviar	Boolean	Envía un correo al destinatario y retorna si fue exitoso.

5.4.4. Infrastructure Layer.

Esta capa se encarga de la persistencia de las notificaciones y de su envío a través de sistemas externos como servicios de correo.

Repository: NotificacionRepository

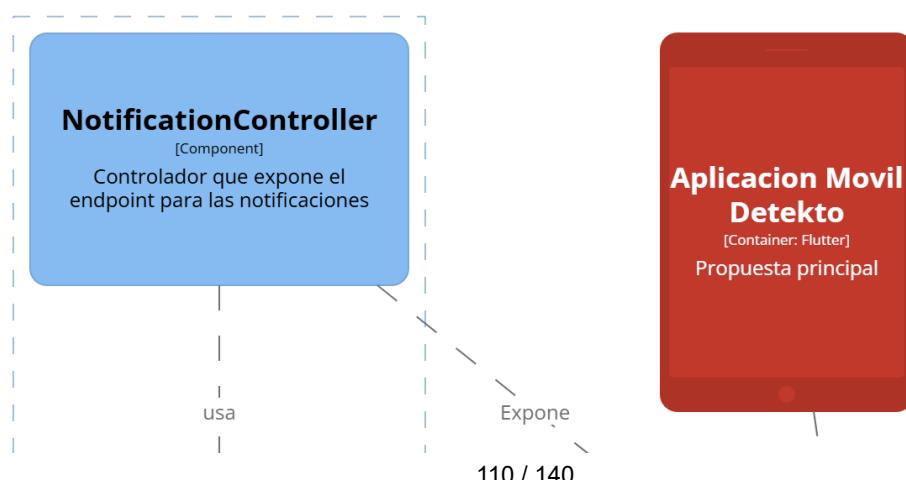
Nombre	Categoría	Propósito
NotificacionRepository	Repositorio	Maneja la persistencia de las notificaciones.

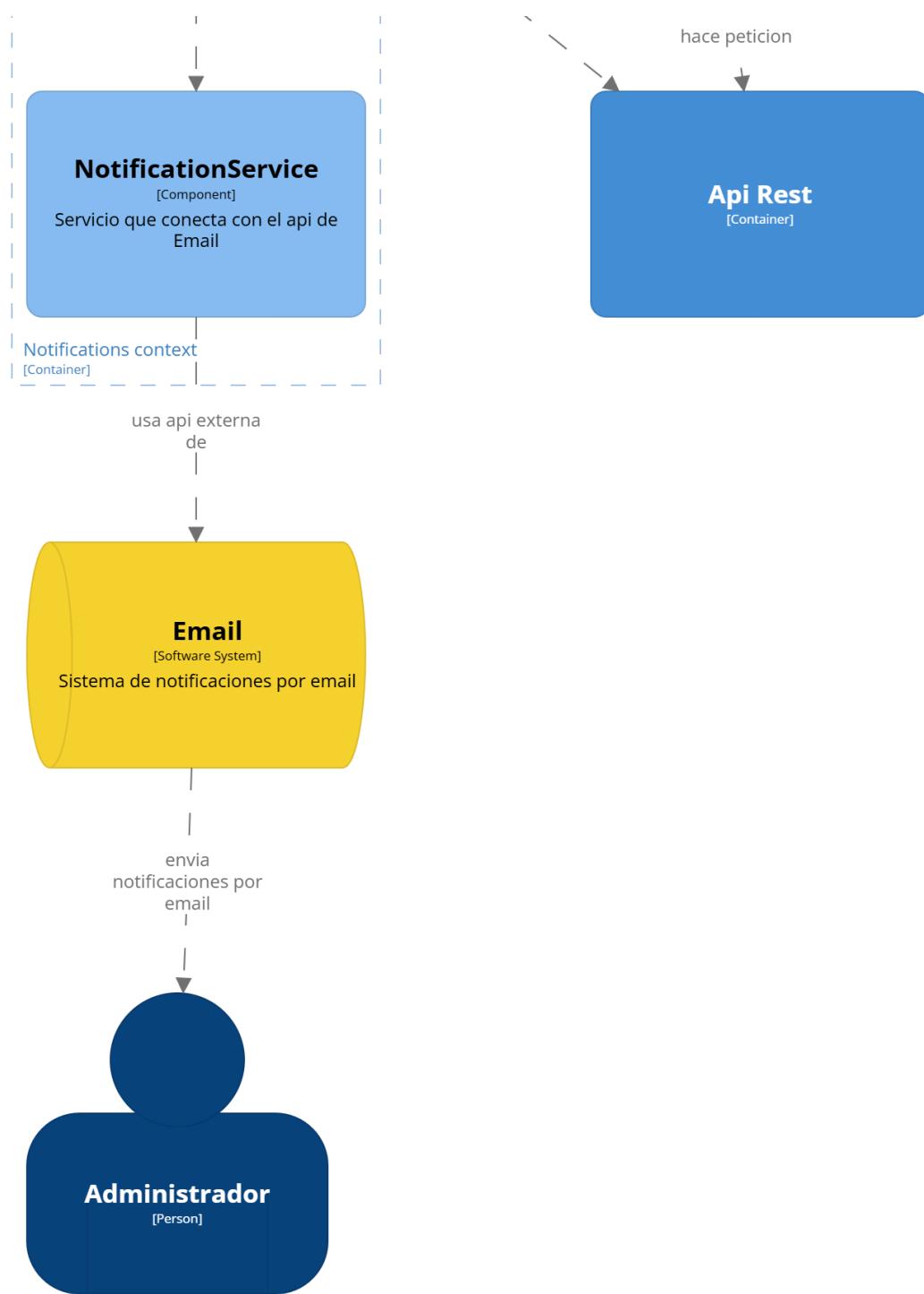
Métodos

Nombre	Tipo de retorno	Visibilidad	Descripción
save	Notificacion	Público	Guarda una notificación en la base de datos.
findByEstado	List	Público	Recupera notificaciones filtradas por estado.
findAll	List	Público	Devuelve todas las notificaciones registradas.

5.4.6. Bounded Context Software Architecture Component Level Diagrams.

Esta sección presenta los diagramas de componentes de la arquitectura de software del contexto de **NOTIFICATIONS**. Estos diagramas muestran la estructura y las relaciones entre los diferentes componentes del sistema.





[Component] Detekto App - Notifications context

jueves, 15 de mayo de 2025, 19:30 hora estándar de Perú

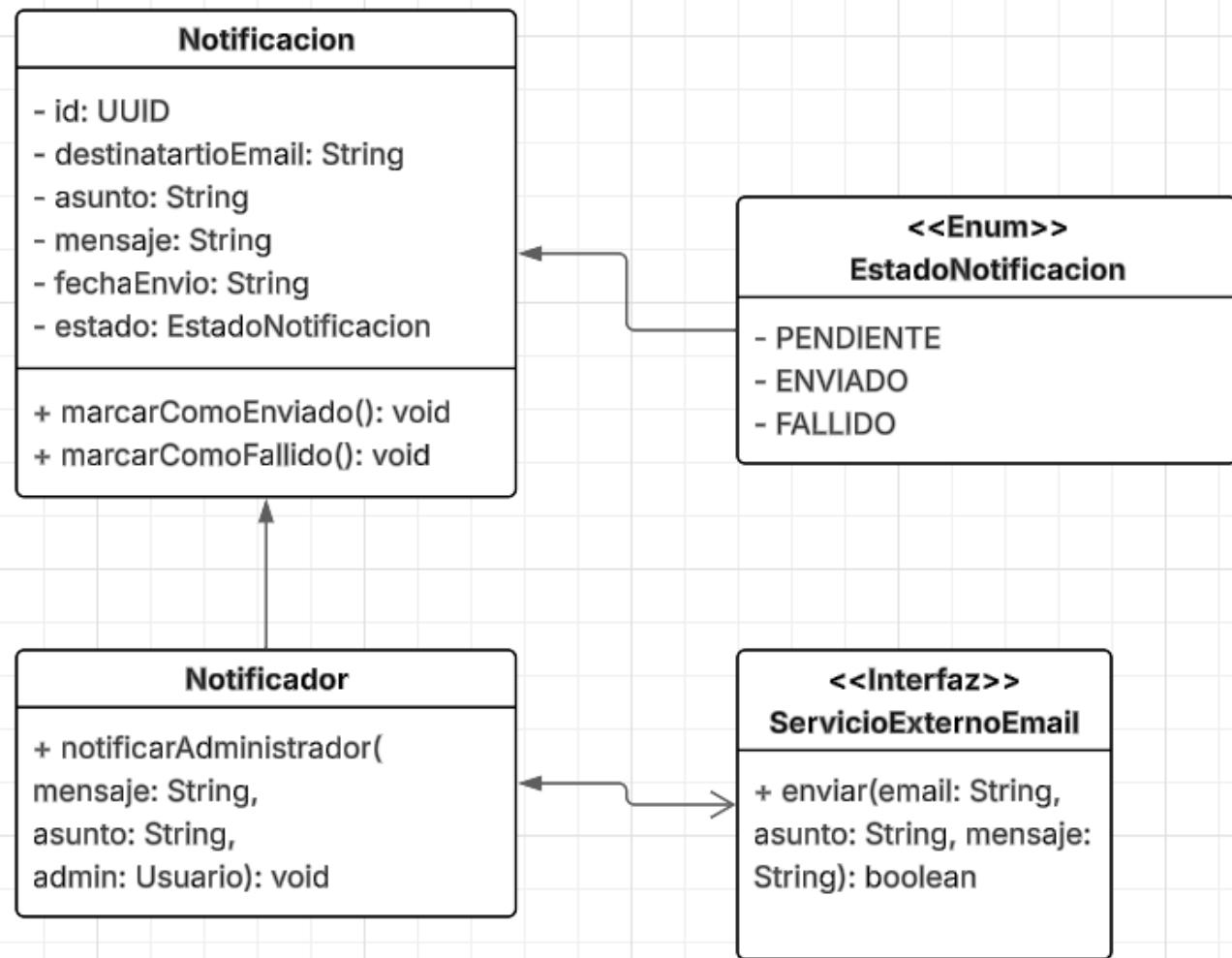
5.4.7. Bounded Context Software Architecture Code Level Diagrams.

Esta sección presenta la implementación de los componentes dentro de cada contexto.

- **Domain Layer Class Diagrams:** Muestra la estructura de las clases y sus relaciones en el contexto de NOTIFICATIONS.
- **Database Design Diagram:** Presenta el diseño de la base de datos, incluyendo las tablas y sus relaciones.

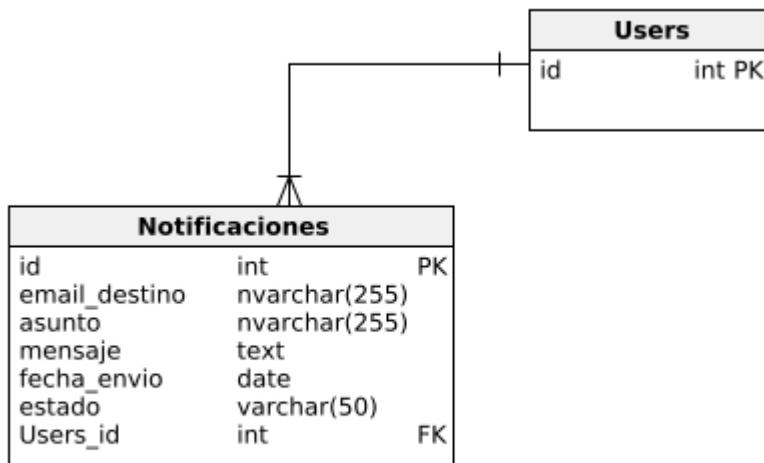
5.4.7.1. Bounded Context Domain Class Diagrams.

En esta sección se presenta el diagrama de clases del contexto de NOTIFICATIONS, en el que se muestran las entidades primordiales para la gestión de las notificaciones, junto la relación que tienen los mismos.



5.4.7.2. Bounded Context Database Design Diagram.

En esta sección, el diagrama de base de datos nos muestra la estructura de las tablas y sus relaciones en el contexto de NOTIFICATIONS. Este diagrama es fundamental para entender cómo se almacenan y gestionan los datos en la aplicación.



5.5. Bounded Context: TRACKING & MONITORING

5.5.1. Domain Layer.

En la capa de dominio del contexto de **Tracking & Monitoring**, se definen las entidades responsables de representar la ubicación de los usuarios y el seguimiento en tiempo real.

Este módulo permite almacenar, actualizar y consultar la posición geográfica de un usuario para fines de monitoreo, seguridad o logística.

Aggregate 1: Location

Nombre	Categoría	Propósito
Location	Entidad	Representa la ubicación geográfica de un usuario en un momento determinado.

Atributos del Location

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único de la ubicación.
userId	UUID	Privado	Identificador del usuario al que pertenece.
latitud	Double	Privado	Coordenada de latitud.
longitud	Double	Privado	Coordenada de longitud.
fechaHora	DateTime	Privado	Fecha y hora en la que se registró la ubicación.

Métodos del Location

Nombre	Tipo de retorno	Visibilidad	Descripción
actualizar	Void	Public	Permite actualizar latitud y longitud.

Nombre	Tipo de retorno	Visibilidad	Descripción
esReciente	Boolean	Public	Indica si la ubicación fue registrada recientemente.

Aggregate 2: TrackingSession

Nombre	Categoría	Propósito
TrackingSession	Entidad	Representa una sesión de seguimiento activo de un usuario.

Atributos del TrackingSession

Nombre	Tipo de dato	Visibilidad	Descripción
id	UUID	Privado	Identificador único de la sesión.
userId	UUID	Privado	Usuario que está siendo monitoreado.
inicio	DateTime	Privado	Fecha y hora de inicio del seguimiento.
fin	DateTime?	Privado	Fecha y hora de finalización (puede ser nula).
ubicaciones	List	Privado	Lista de ubicaciones registradas durante la sesión.

Métodos del TrackingSession

Nombre	Tipo de retorno	Visibilidad	Descripción
agregarUbicacion	Void	Public	Añade una nueva ubicación al historial de la sesión.
finalizarSesion	Void	Public	Marca la sesión como finalizada.
obtenerRecorrido	List	Public	Devuelve la lista de ubicaciones ordenadas cronológicamente.

5.5.2. Interface Layer.

En la capa de interfaz del contexto de TRACKING & MONITORING de la aplicación, se definen los controladores y servicios que manejan las solicitudes y respuestas de la API. Los controladores son responsables de recibir las solicitudes HTTP y devolver las respuestas correspondientes.

Controller 1: MonitoringController

Nombre	Categoría	Propósito
MonitoringController	Controlador	Expone endpoints públicos para geolocalizar usuarios.

Atributos del MonitoringController

Nombre	Tipo de dato	Visibilidad	Descripción

Nombre	Tipo de dato	Visibilidad	Descripción
monitoringService	MonitoringService	Privado	Servicio que contiene la lógica de negocio de localización.

Métodos del MonitoringController

Nombre	Tipo de retorno	Visibilidad	Descripción
getCurrentLocation	ResponseEntity	Public	Retorna la ubicación actual del usuario.
trackUserLocation	ResponseEntity	Public	Realiza el seguimiento de un usuario en tiempo real.

5.5.3. Application Layer.

En la capa de aplicación del contexto de TRACKING & MONITORING de la aplicación, se definen los servicios que manejan la lógica de negocio relacionada con los usuarios y roles. Estos servicios son utilizados por los controladores para realizar las operaciones necesarias.

Service 1: MonitoringService

Nombre	Categoría	Propósito
MonitoringService	Servicio	Gestiona la lógica de negocio para la obtención de ubicación.

Atributos del MonitoringService

Nombre	Tipo de dato	Visibilidad	Descripción
mapsApiClient	MapsAPIClient	Privado	Cliente que consume la API de geolocalización externa.

Métodos del MonitoringService

Nombre	Tipo de retorno	Visibilidad	Descripción
fetchLocation	Location	Public	Consulta la ubicación del usuario mediante el sistema externo.
trackInRealtime	Stream	Public	Inicia la monitorización en tiempo real de un usuario.

5.5.4. Infrastructure Layer.

Esta capa define el repositorio responsable de acceder y persistir los datos relacionados con la ubicación y sesiones de seguimiento en la base de datos del sistema.

Repository 1: LocationRepository

Nombre	Categoría	Propósito
LocationRepository	Repositorio	Gestiona la persistencia de las ubicaciones geográficas de los usuarios.

Métodos del LocationRepository

Nombre	Tipo de retorno	Visibilidad	Descripción
save	Location	Public	Guarda una nueva ubicación en la base de datos.
findById	List	Public	Obtiene todas las ubicaciones asociadas a un usuario.
findLatestById	Location	Public	Retorna la última ubicación registrada de un usuario.
deleteById	Void	Public	Elimina las ubicaciones asociadas a un usuario.

Repository 2: TrackingSessionRepository

Nombre	Categoría	Propósito
TrackingSessionRepository	Repositorio	Administra el acceso a las sesiones de seguimiento de los usuarios.

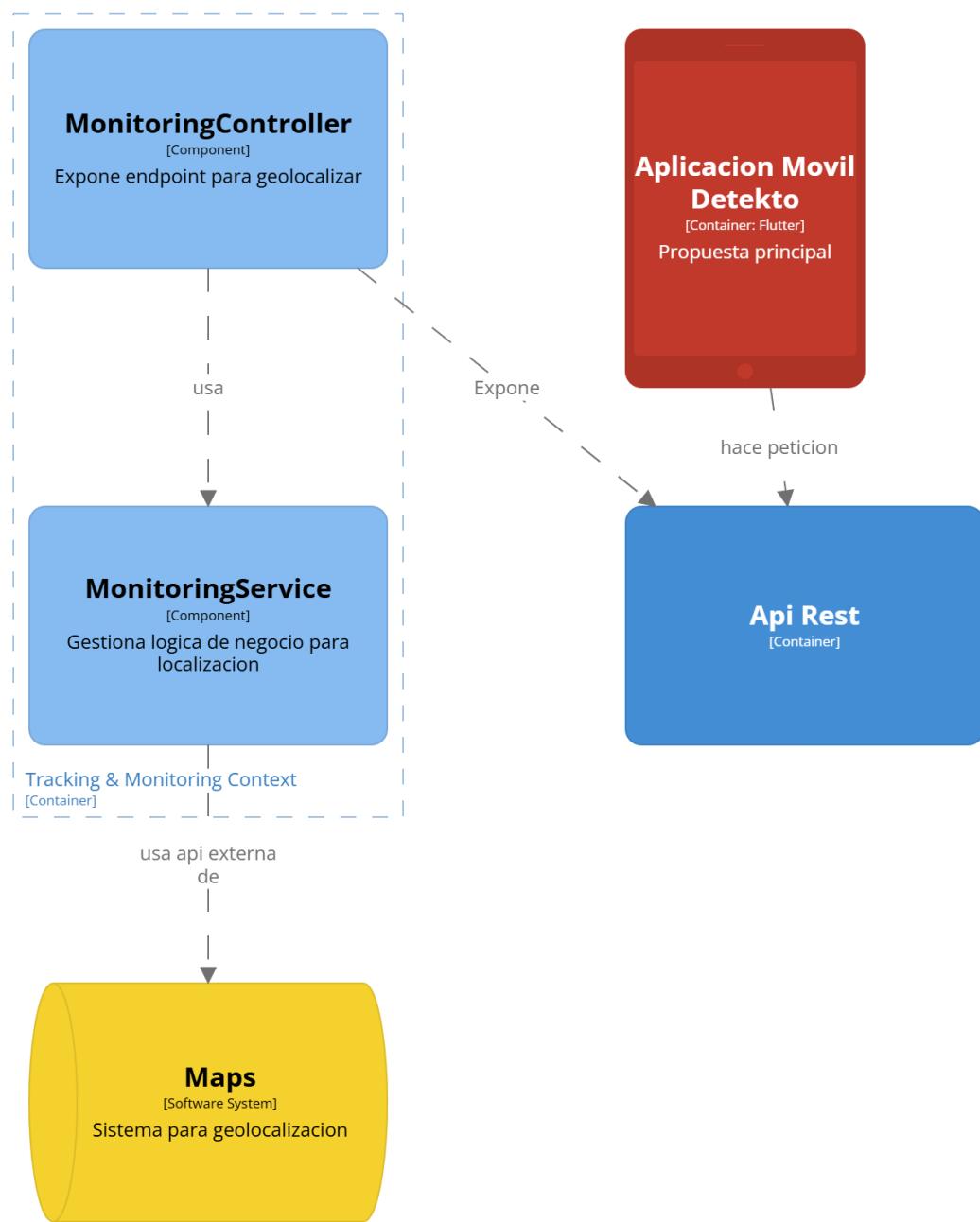
Métodos del TrackingSessionRepository

Nombre	Tipo de retorno	Visibilidad	Descripción
save	TrackingSession	Public	Guarda una nueva sesión de seguimiento.
findActiveById	TrackingSession	Public	Devuelve la sesión activa actual de un usuario, si existe.
findById	TrackingSession	Public	Busca una sesión por su identificador.
closeSessionById	Void	Public	Finaliza una sesión marcándola con fecha de cierre.
getSessionHistoryById	List	Public	Retorna todas las sesiones de seguimiento de un usuario.

5.5.6. Bounded Context Software Architecture Component Level Diagrams.

Esta sección presenta los diagramas de componentes de la arquitectura de software del contexto de TRACKING & MONITORING. Estos diagramas muestran la estructura y las relaciones entre los diferentes

componentes del sistema.



[Component] Detekto App - Tracking & Monitoring Context

jueves, 15 de mayo de 2025, 19:30 hora estndar de Per

5.5.7. Bounded Context Software Architecture Code Level Diagrams.

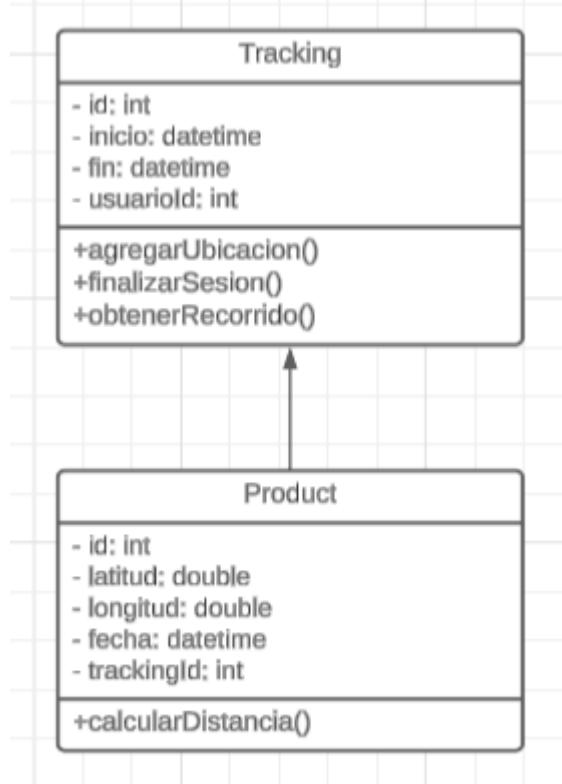
En esta sección, el equipo de Softwarinos presenta la implementación de los componentes dentro de cada contexto.

- **Domain Layer Class Diagrams:** Muestra la estructura de las clases y sus relaciones en el contexto de TRACKING & MONITORING.

- **Database Design Diagram:** Presenta el diseño de la base de datos, incluyendo las tablas y sus relaciones.

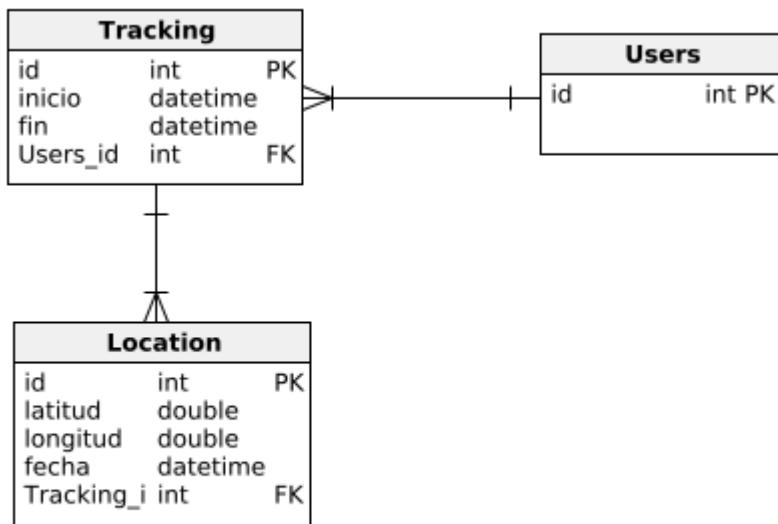
5.5.7.1. Bounded Context Domain Layer Class Diagrams.

En esta sección se presentan los diagramas de clases del contexto de TRACKING & MONITORING, en el que se muestran las entidades claves para la autenticación del usuario, los roles, junto la relación que tienen los mismos



5.5.7.2. Bounded Context Database Design Diagram.

En esta sección, el diagrama de base de datos nos muestra la estructura de las tablas y sus relaciones en el contexto de TRACKING & MONITORING. Este diagrama es fundamental para entender cómo se almacenan y gestionan los datos en la aplicación.



Capítulo VI: Solution UX Design

6.1. Style Guidelines.

En esta sección se definirán las guías visuales clave para el diseño de DETEKTO, incluyendo colores, tipografía, logotipo y tono de comunicación. Estos lineamientos asegurarán una identidad coherente y profesional que refleje los valores de confianza e innovación de la marca.

6.1.1. General Style Guidelines.

Es fundamental considerar las decisiones y elementos visuales que respaldan los principios generales de diseño para DETEKTO. Por esta razón, resulta esencial definir aspectos clave como la identidad de marca, la paleta de colores y la tipografía. Además, es necesario establecer el tono de comunicación y el lenguaje utilizado, que abarcan características como divertido o serio, formal o casual, respetuoso o irreverente, y entusiasta o sereno.

6.1.2. Web, Mobile & Devices Style Guidelines.



DETEKTO representa innovación, confianza y eficiencia en la gestión y ventas y stock con reconocimiento por objetos con IA. Nuestra esencia radica en ofrecer soluciones tecnológicas modernas que permiten monitorear, optimizar y mejorar sus ventas. La marca es sinónimo de precisión, confiabilidad y una experiencia de usuario simplificada, orientada a un entorno urbano inteligente.

Logotipo: El logotipo de DETEKTO refleja la identidad de una marca moderna, segura y tecnológica. Mediante una combinación de colores vivos y un diseño minimalista pero sofisticado, el logo simboliza la eficiencia, la confianza y el uso de tecnología de vanguardia en la vigilancia de parqueo. Este logotipo debe transmitir una imagen clara de control, innovación y accesibilidad para usuarios y operadores. Colores:

	HEX 1E1E1E RGB 30, 30, 30		HEX 2E3447 RGB 42, 52, 71		HEX 4F566B RGB 79, 86, 107		HEX 8E9099 RGB 142, 144, 153		HEX CBCED5 RGB 203, 206, 213
--	------------------------------------	--	------------------------------------	--	-------------------------------------	--	---------------------------------------	--	---------------------------------------

Se ha utilizado los colores blanco, negro y azul oscuro como colores principales para nuestro diseño.

● Azul oscuro (#2E3447): El color azul oscuro se utiliza como color principal de la marca, aportando confianza, seguridad y modernidad. Representa la tecnología y la eficiencia que caracterizan a DETEKTO en su enfoque hacia la optimización del estacionamiento urbano. Es protagonista en los botones principales, encabezados y elementos destacados de la interfaz.

- Blanco (#FFFFFF): El blanco se emplea como color de fondo predominante, transmitiendo limpieza, claridad y simplicidad en la experiencia del usuario. Sirve como base para resaltar los demás colores y garantizar una lectura fluida, orden visual y una navegación intuitiva.
- Gris (#4F566B / #8E9099): Los tonos grises se utilizan para los textos secundarios, bordes de campos de entrada, íconos y elementos informativos. Brindan equilibrio visual, profesionalismo y elegancia. Este color apoya al azul oscuro sin competir con él, manteniendo una estética sobria y tecnológica.

Tipografía:

Inter Regular

Inter Regular

Inter Italic

Inter Italic

Inter Thin BETA

Inter Thin BETA

Inter Thin Italic BETA

Inter Thin Italic BETA

Inter Extra Light BETA

Inter Extra Light BETA

Inter Extra Light Italic BETA

Inter Extra Light Italic BETA

Inter Light BETA

Inter Light BETA

La tipografía utilizada para DETEKTO es Inter , la cual es, moderna y legible, con líneas limpias y claras. Se ha elegido una fuente que refleje la tecnología y la seguridad de la marca, mientras mantiene un aspecto novedoso y actual. Usaremos las variantes Regular, Medium, Semi-Bold y Bold.

Tonos de Comunicación:

- Formal / Confiable: En DETEKTO adoptamos un tono formal y profesional que transmite confianza y responsabilidad, especialmente en lo relacionado a la seguridad del usuario y la gestión de sus reservas. Sin embargo, mantenemos una comunicación cercana y comprensible, accesible para todos los usuarios que buscan una solución eficiente para estacionar.

- Respetuoso / Cercano: Siempre nos comunicamos con respeto, claridad y empatía. Entendemos las frustraciones comunes en la búsqueda de estacionamiento y respondemos con soluciones y mensajes que conectan con las verdaderas necesidades de nuestros usuarios, manteniendo una relación transparente y honesta.
- Entusiasta / Sereno: Transmitimos entusiasmo por cambiar la forma en que las personas se estacionan en la ciudad, destacando nuestra innovación tecnológica y eficiencia. Al mismo tiempo, mantenemos un tono sereno y confiable que le brinda al usuario la tranquilidad de saber que su experiencia con DETEKTOUp será rápida, segura y sin complicaciones.

6.2. Information Architecture.

6.2.2. Labeling Systems.

En esta sección se presenta el sistema de etiquetado que Detekto utilizará para facilitar la comprensión y navegación dentro de la plataforma, tanto en la aplicación como en la landing page. Los encabezados estarán organizados de forma clara, accesible y adaptada a los distintos perfiles de usuarios.

Inicio/Home: Vista principal con acceso a funcionalidades clave, como escaneo de productos, sincronización de stock, historial de ventas y alertas de inventario. Se incluirá una introducción breve sobre la misión de Detekto: optimizar la gestión de ventas y control de inventario en tiempo real usando tecnologías inteligentes.

Funcionalidades/Features: Se describirán las principales funcionalidades, como el reconocimiento automático de objetos, actualización en tiempo real del stock, generación de reportes de ventas, integración con sistemas de gestión y facilidad de uso para pequeños y medianos negocios. También se destacará el valor agregado de la automatización del proceso de inventariado mediante escaneo móvil.

Preguntas Frecuentes/FAQs: Se responderán preguntas comunes sobre cómo registrarse, realizar escaneos, sincronizar datos con sistemas externos, revisar el historial de stock y ventas, así como la configuración de notificaciones automáticas.

Contáctanos/Contact Us: Incluirá canales de atención como correo electrónico, WhatsApp y un formulario de contacto directo para soporte técnico o consultas comerciales.

6.2.3. Searching Systems.

En esta sección se presenta el sistema de etiquetado que **Detekto** utilizará para facilitar la comprensión y navegación dentro de la plataforma, tanto en la aplicación como en la landing page. Los encabezados estarán organizados de forma clara, accesible y adaptada a los distintos perfiles de usuarios.

Inicio/Home: Vista principal con acceso a funcionalidades clave, como escaneo de productos, sincronización de stock, historial de ventas y alertas de inventario. Se incluirá una introducción breve sobre la misión de Detekto: optimizar la gestión de ventas y control de inventario en tiempo real usando tecnologías inteligentes.

Funcionalidades/Features: Se describirán las principales funcionalidades, como el reconocimiento automático de objetos, actualización en tiempo real del stock, generación de reportes de ventas, integración con sistemas de gestión y facilidad de uso para pequeños y medianos negocios. También se destacará el valor agregado de la automatización del proceso de inventariado mediante escaneo móvil.

Preguntas Frecuentes/FAQs: Se responderán preguntas comunes sobre cómo registrarse, realizar escaneos, sincronizar datos con sistemas externos, revisar el historial de stock y ventas, así como la configuración de notificaciones automáticas.

Contáctanos/Contact Us: Incluirá canales de atención como correo electrónico, WhatsApp y un formulario de contacto directo para soporte técnico o consultas comerciales.

5.2.3. SEO Tags and Meta Tags

Para optimizar la visibilidad de **Detekto** en los motores de búsqueda, se definirán las siguientes etiquetas SEO y metadatos para su landing page:

Title: Detekto | Gestión inteligente de stock y ventas con escaneo móvil.

Description: Detekto - Escanea, reconoce productos y actualiza tu inventario en tiempo real desde tu celular.

Keywords: gestión de ventas, control de stock, escaneo de productos, inventario inteligente, reconocimiento de objetos, Detekto app.

Author: Detekto Team

Canonical: [URL principal de la landing page, por definir]

6.2.3. Searching Systems

El sistema de búsqueda en **Detekto** ha sido diseñado para brindar rapidez, precisión y facilidad de uso, adaptándose a las necesidades operativas de usuarios que gestionan inventario desde dispositivos móviles. Existen dos modalidades principales de búsqueda implementadas dentro de la plataforma:

1. Búsqueda por escaneo (Scan-to-Search)

El núcleo de Detekto es su sistema de reconocimiento de objetos mediante cámara o escáner móvil. Al escanear un producto, el sistema:

- Reconoce el objeto utilizando visión por computadora y machine learning.
- Busca automáticamente el producto en la base de datos del inventario.
- Devuelve los datos asociados, como nombre del producto, cantidad en stock, precio, historial de ventas y alertas relacionadas.

Esta búsqueda es ideal para situaciones de alta rotación de productos o actualización inmediata de inventario.

2. Búsqueda textual (Text Search)

La aplicación también permite la búsqueda manual mediante texto para mayor flexibilidad. El usuario puede introducir:

- Nombre del producto.
- Código SKU.
- Categoría o etiqueta personalizada.

El motor de búsqueda implementa funcionalidades de *autocomplete*, *tolerancia a errores tipográficos (fuzzy search)* y *filtros por estado de stock*, lo que facilita encontrar productos incluso en catálogos extensos.

Características adicionales:

- Historial de búsquedas recientes.
- Resultados ordenados por relevancia y frecuencia de actualización.
- Acceso rápido desde la barra de navegación superior.

Estas funcionalidades permiten que tanto operadores de almacén como encargados de ventas accedan de forma inmediata a la información necesaria, optimizando la toma de decisiones y reduciendo errores operativos.

6.2.4. SEO Tags and Meta Tags.

Para optimizar la visibilidad de Detekto en los motores de búsqueda, se definirán las siguientes etiquetas SEO y metadatos para su landing page:

Title: Detekto | Gestión inteligente de stock y ventas con escaneo móvil.

Description: Detekto - Escanea, reconoce productos y actualiza tu inventario en tiempo real desde tu celular.

Keywords: gestión de ventas, control de stock, escaneo de productos, inventario inteligente, reconocimiento de objetos, Detekto app.

Author: Detekto Team

Canonical: [URL principal de la landing page, por definir]

6.2.5. Navigation Systems.

La navegación en **Detekto** se diseñará para ofrecer una experiencia fluida, intuitiva y centrada en las tareas más frecuentes del usuario relacionadas con la gestión de ventas e inventario.

Menú de navegación principal: Estará disponible en la parte inferior (en dispositivos móviles) o lateral (en tabletas), con acceso directo a secciones como "Inicio", "Escanear Producto", "Inventario", "Ventas", "Alertas" y "Perfil". Este menú prioriza las acciones operativas más comunes en el flujo de trabajo de supervisores y vendedores.

Búsqueda centralizada: Se incluirá un campo de búsqueda visible y accesible desde la pantalla principal para localizar rápidamente productos por nombre, código SKU o categoría. También podrá activarse el escáner directamente desde esta barra de búsqueda.

Botones de acción estratégicos (CTAs): Botones como "Escanear Ahora", "Actualizar Stock", "Registrar Venta", "Enviar Reporte" y "Ver Detalles" estarán claramente identificados y ubicados en lugares clave dentro de cada flujo, facilitando la interacción eficiente con la aplicación.

Navegación coherente y simplificada: Toda la plataforma mantendrá una estructura de navegación consistente, con iconografía clara, rutas predecibles, menú de retroceso accesible y jerarquías visuales definidas. Esto permite que tanto nuevos usuarios como operadores experimentados puedan moverse rápidamente dentro de la aplicación sin curva de aprendizaje significativa.

6.3. Landing Page UI Design.

6.3.1. Landing Page Wireframe.

Sección Home

Wireframe - Home

Detekto

Inicio Nosotros Aplicacion Equipo Contacto

Bienvenido a Softwarinos

Detekto es la app Inteligente para ferreterías: reconoce herramientas, organiza tu stock y potencia tus ventas.

Otros más

Sección Nosotros

Wireframe - About

Detekto

Inicio Nosotros Aplicacion Equipo Contacto

Sobre Nosotros

En Softwarinos, desarrollamos una app móvil con reconocimiento de objetos para facilitar a los vendedores la búsqueda y venta de productos, y permitir a los administradores dar seguimiento y proyectar sus ventas.



Misión

Visión

Sección About the product/Aplicacion

Wireframe - Aplicacion

Dtekto

Inicio Nosotros Aplicacion Equipo Contacto

Sobre la Aplicación

Con Dtekto podrás identificar herramientas apuntando con tu cámara, ver productos disponibles por ranura, gestionar tu inventario y realizar reservas en segundos.

Servicio 1

Loren ipsum dolor sit amet, consectetur adipiscing elit. Sed iaculis erat quis lectus interdum, eu suscipit massa auctor. Aenean quis consequat nulla. Etiam vulputate, tortor non aliquam varius.



Servicio 2

Loren ipsum dolor sit amet, consectetur adipiscing elit. Sed iaculis erat quis lectus interdum, eu suscipit massa auctor. Aenean quis consequat nulla. Etiam vulputate, tortor non aliquam varius.



Seccion About the team/Equipo

Wireframe - Aplicacion

Dtekto

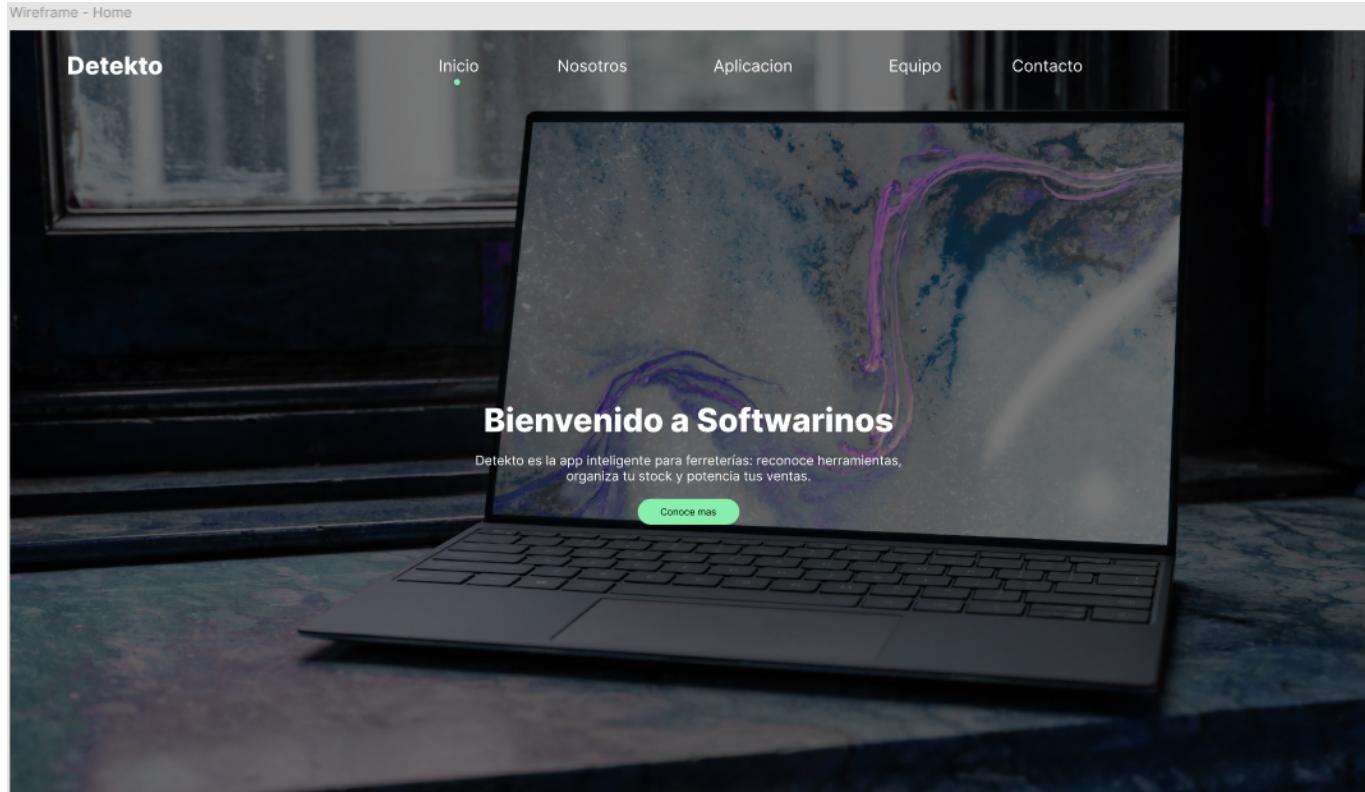
Inicio Nosotros Aplicacion Equipo Contacto

Sobre el Equipo



6.3.2. Landing Page Mock-up.

Seccion Home



Sección Nosotros

Wireframe - About

The wireframe displays the 'About' section of the website. It features a header with the 'Detekto' logo and a navigation bar with links to 'Inicio', 'Nosotros', 'Aplicacion', 'Equipo', and 'Contacto'. Below the header, the page title 'Sobre Nosotros' is centered. To the left of the text, there is a paragraph describing the company's mission: 'En Softwarinos, desarrollamos una app móvil con reconocimiento de objetos para facilitar a los vendedores la búsqueda y venta de productos, y permitir a los administradores dar seguimiento y proyectar sus ventas.' To the right of the text is a photograph of a workspace with a laptop, monitor, and desk lamp. At the bottom, two dark blue rectangular buttons contain the words 'Misión' and 'Visión' in white text.

Sección About the product/Aplicacion

Wireframe - Aplicacion

Detekto

Inicio Nosotros Aplicacion Equipo Contacto

Sobre la Aplicación

Con Detekto podrás identificar herramientas apuntando con tu cámara, ver productos disponibles por marca, gestionar tu inventario y realizar reservas en segundos.

Reconocimiento de Herramientas

Detecta automáticamente productos como martillos, destornilladores o sierras y muestra las marcas disponibles.





Gestión de Inventario

Organiza productos, visualiza tu stock en tiempo real y evita quiebres de stock.

Seccion About the team/Equipo

Wireframe - Aplicacion

Detekto

Inicio Nosotros Aplicacion Equipo Contacto

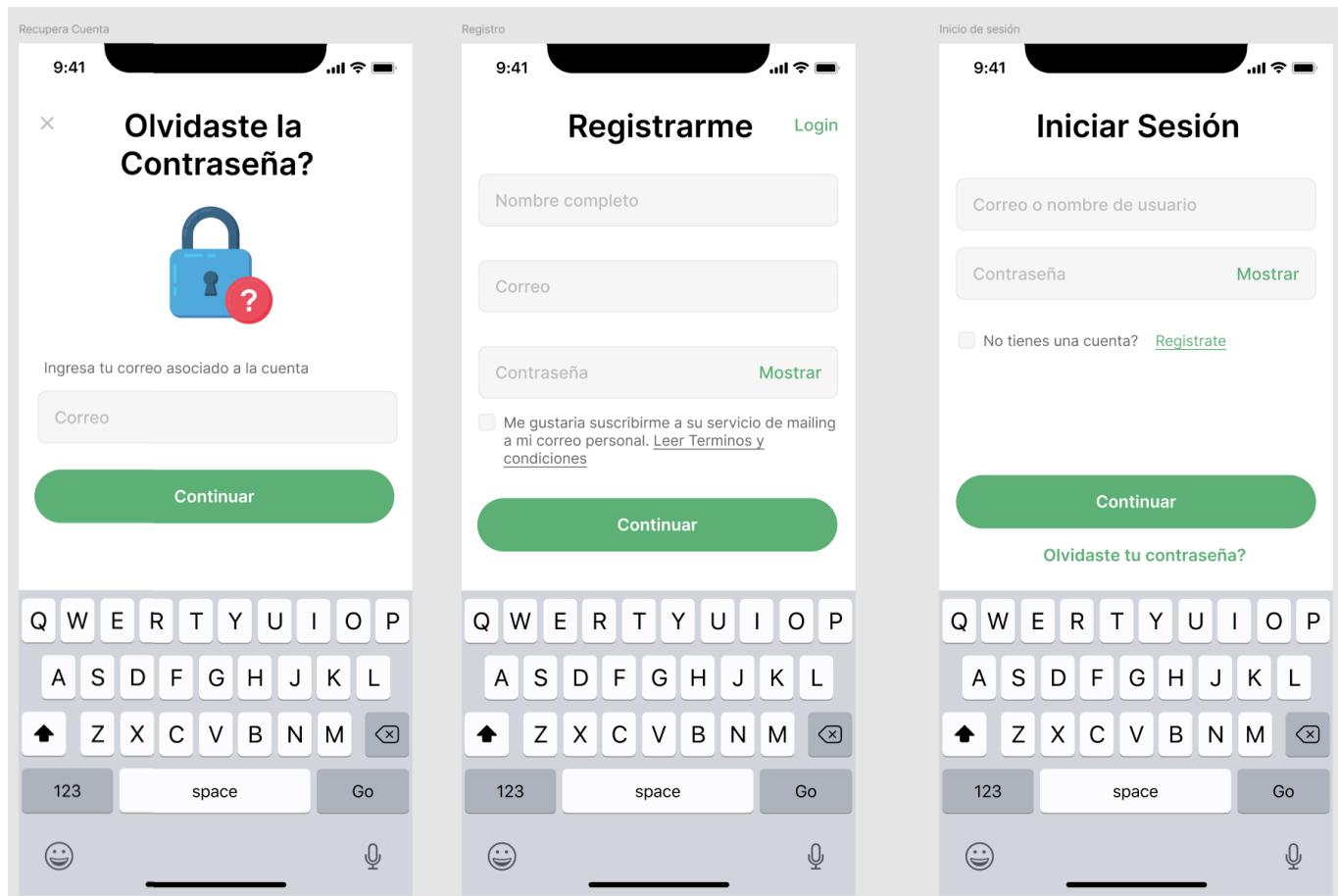
Sobre el Equipo

Somos Softwarinos, un equipo de apasionados por la tecnología y la innovación para la industria ferretera.

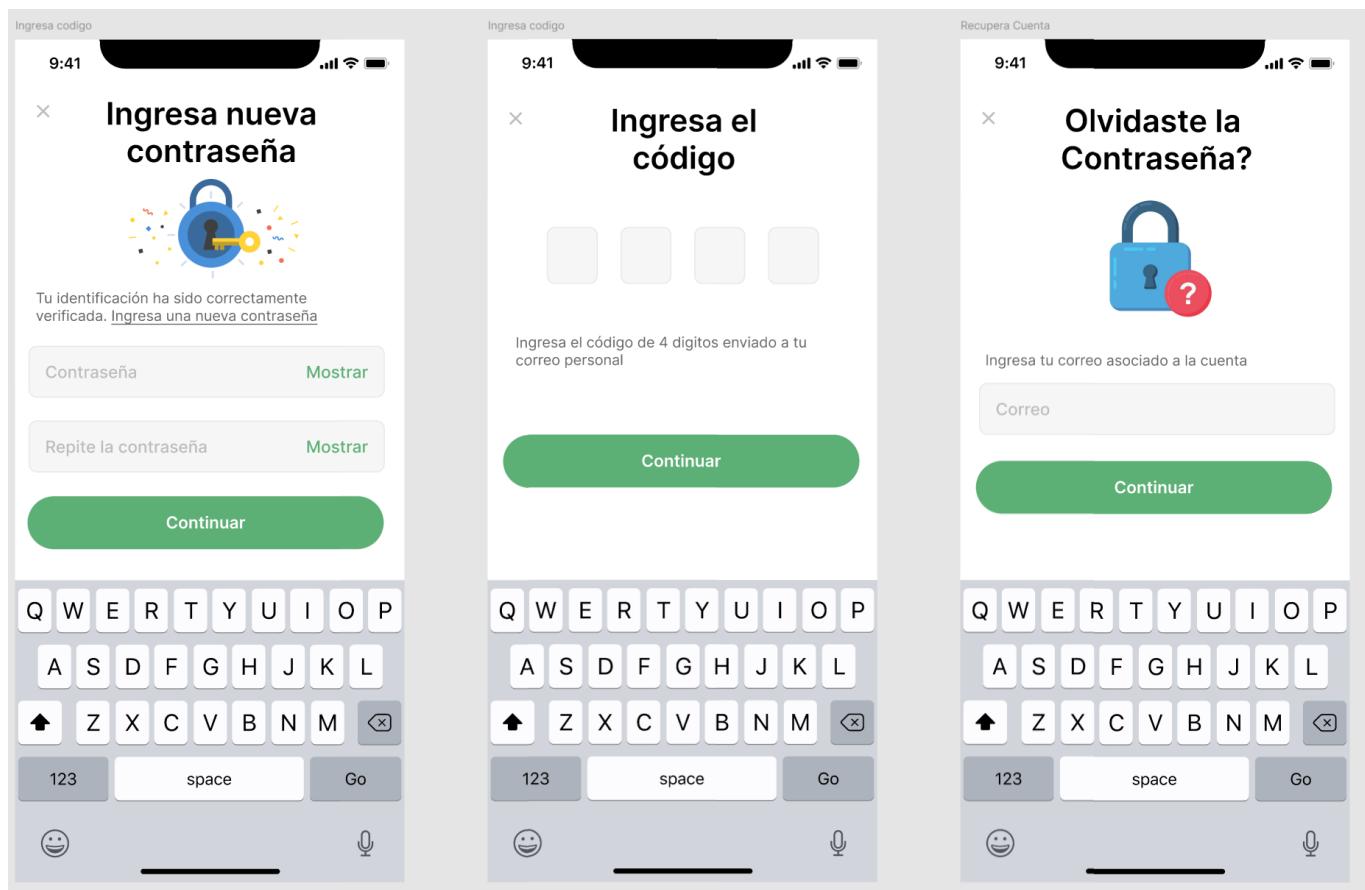


6.4. Applications UX/UI Design.

Inicia sesion, Registrom, Recupera tu cuenta:



Recupera tu cuenta:



Inicio y Ventas:

Inicio

9:41

Inicio **Filtrar**

Buscar

- Martillos y Destornilladores** hace 8m
Venta de productos sección Herramientas manuales
- Taladro, amoladora** hace 15m
Venta de productos sección Herramientas eléctricas
- Cemento, cal, arena** hace 30m
Venta de productos sección Materiales de construcción
- Tubos PVC, grifería** hace 1h
Venta de productos sección Fontanería

GRAFICO DE BARRAS DE VENTAS POR CATEGORIA
2025

Ventas

9:41

Ventas **Filtrar**

Buscar

Martillos y Destornilladores
Compra de 30 unidades de Martillos y 20 Destor....
8m ago

Ventas Totales

9:41

Ventas Totales **Filtrar**

Buscar

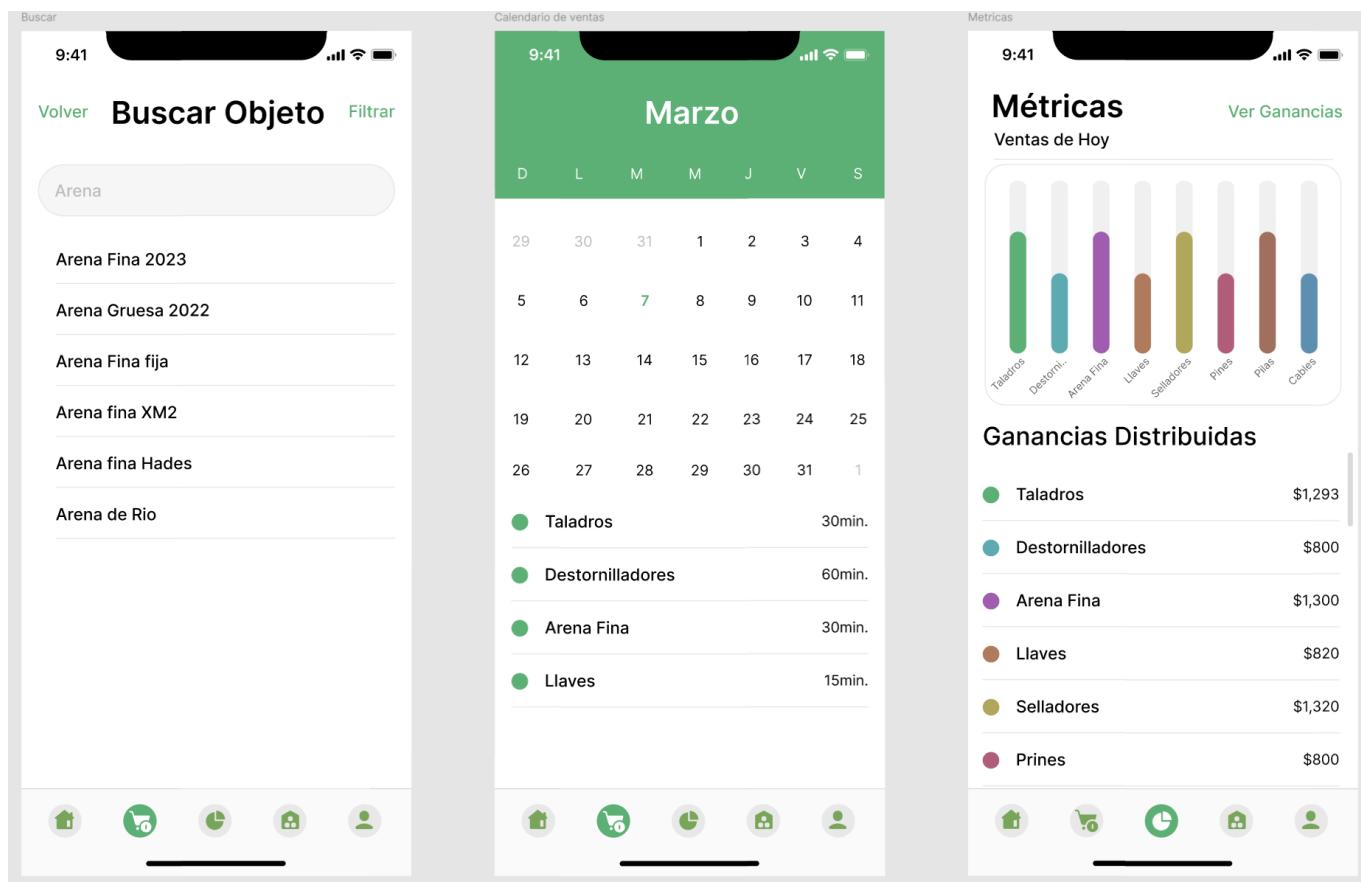
Más vendidos

Imagen	Producto	Precio
	Martillo TitanForce 5000	\$19.99
	Destornillador MaxGrip Pro	\$9.99
	Llave Ajustable FlexiTool	\$29.99

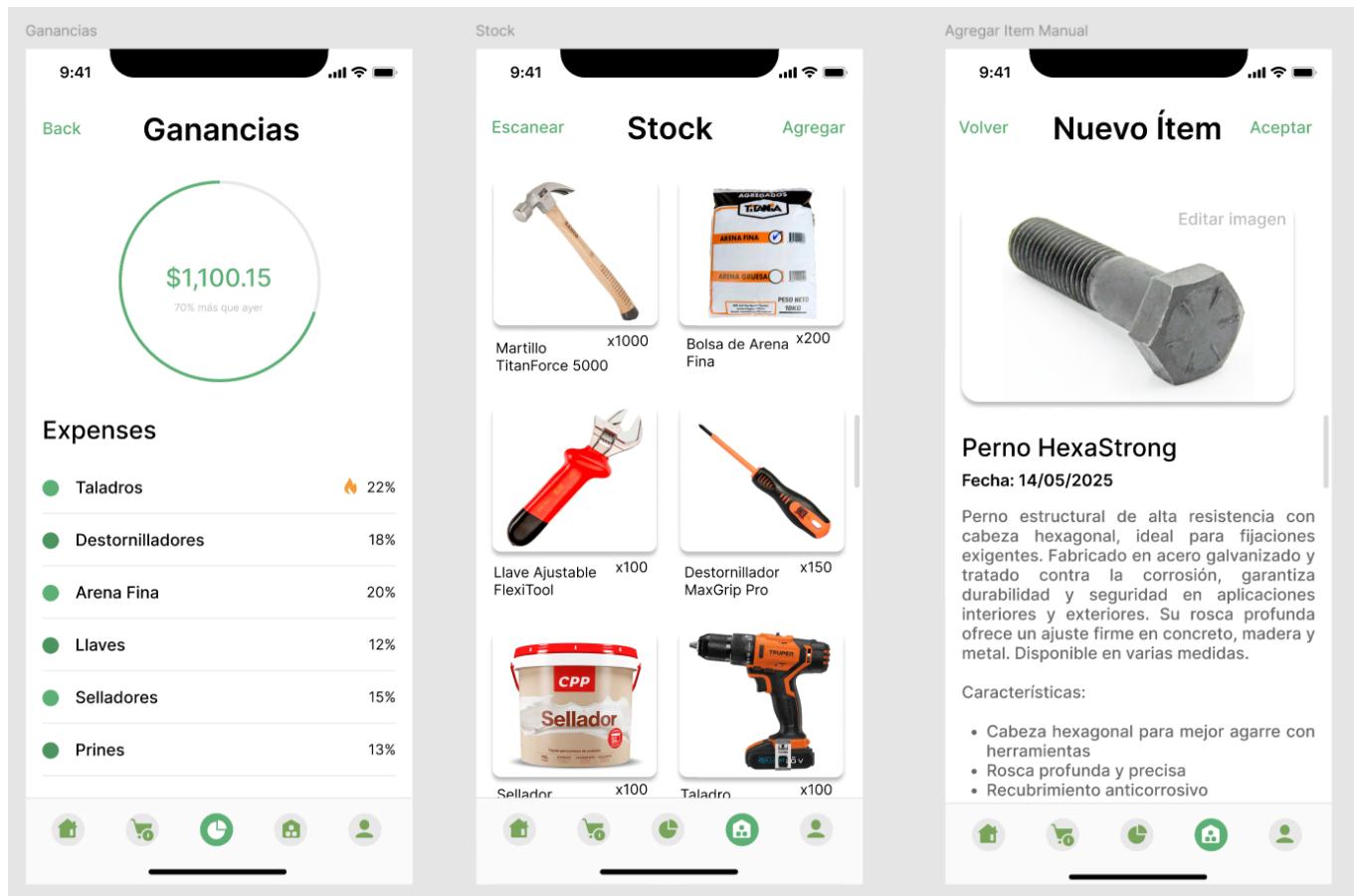
Lista Completa

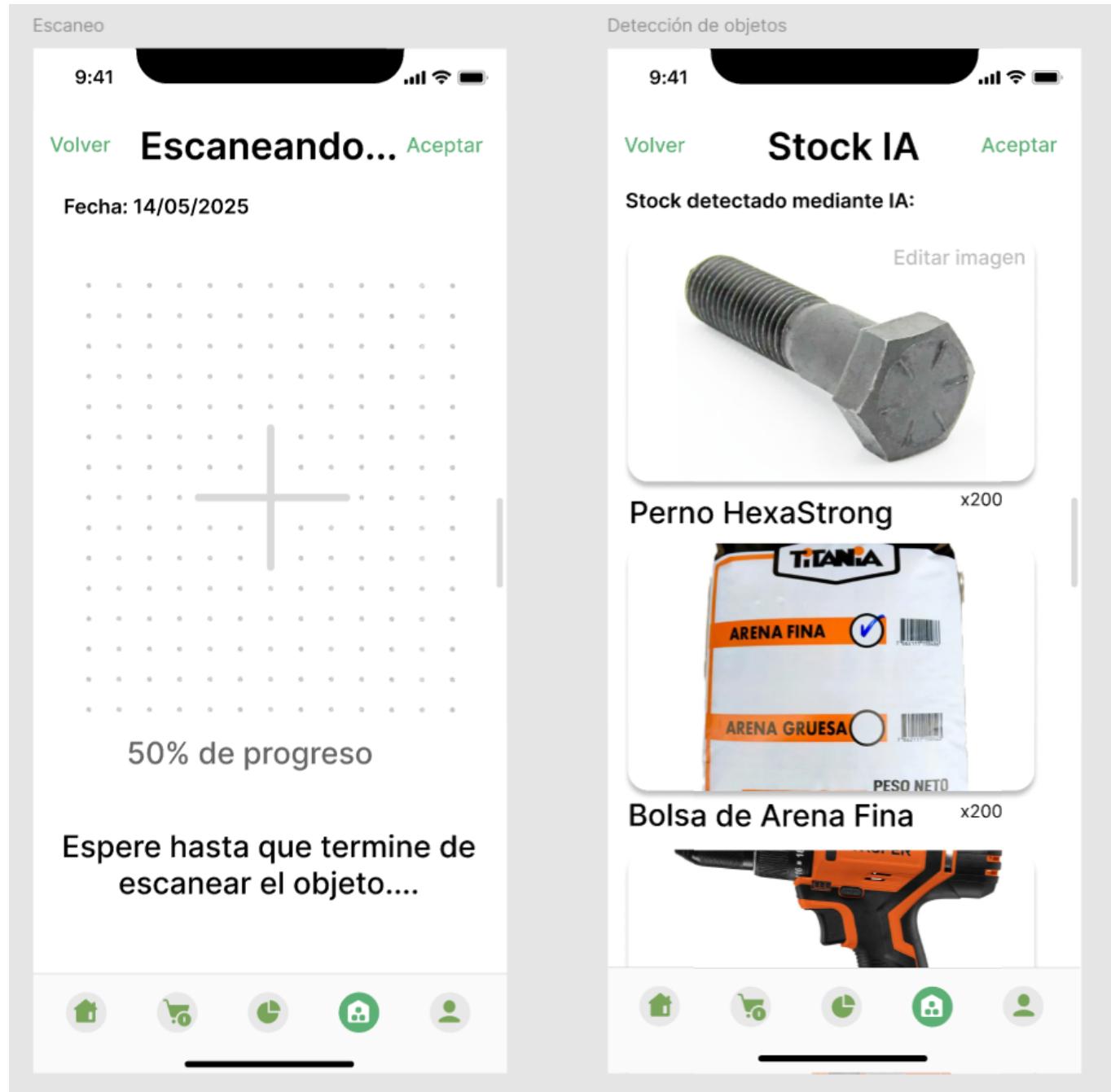
Imagen	Producto	Precio
	Taladro PowerDrill	\$49.99
	Arena Fina ConstruPlus	\$8.99
	Sellador PrimeGuard	\$5.99

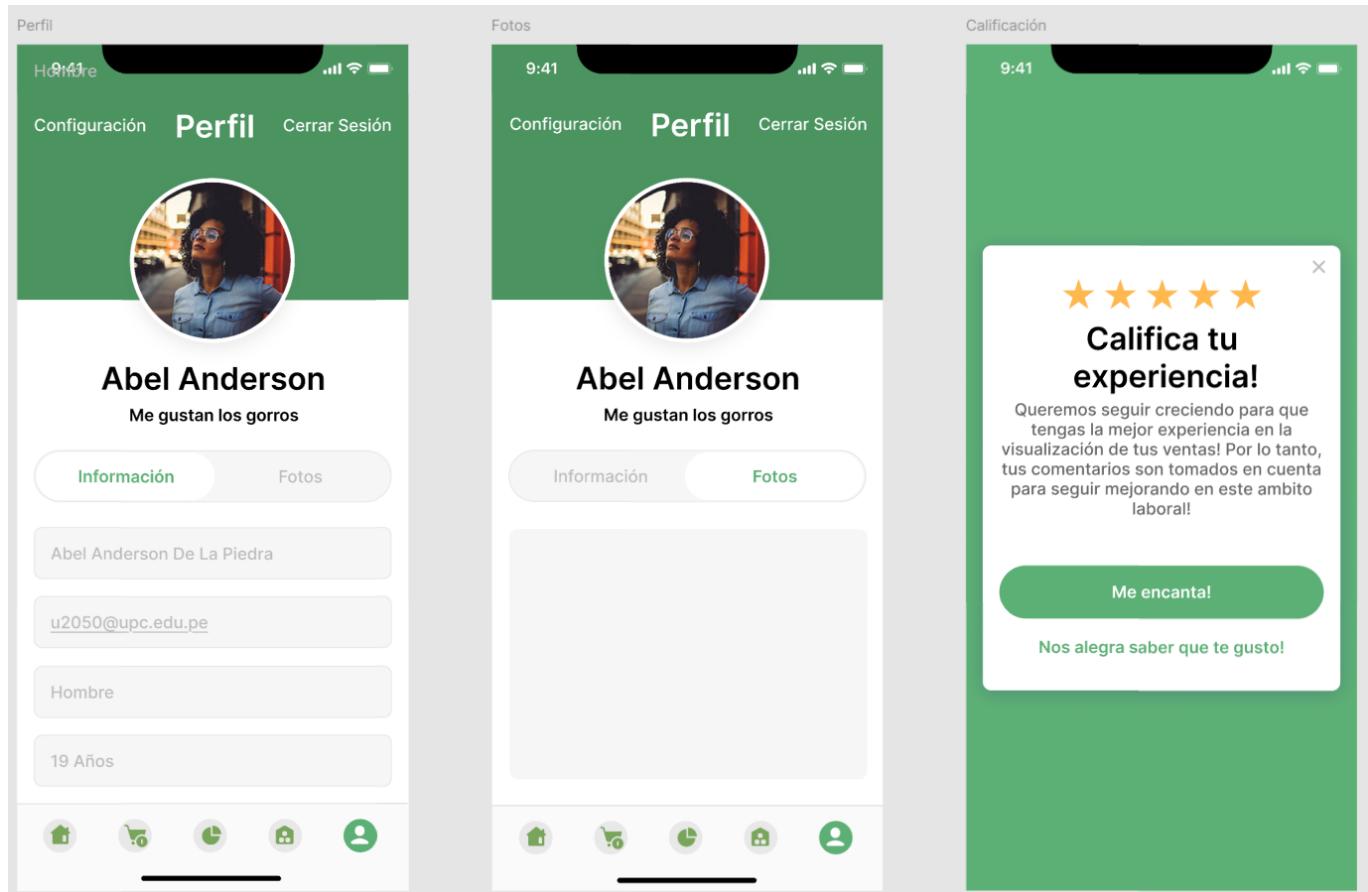
Buscar Objeto y Métricas



Ganancias, Stock y Agregar Ítem:

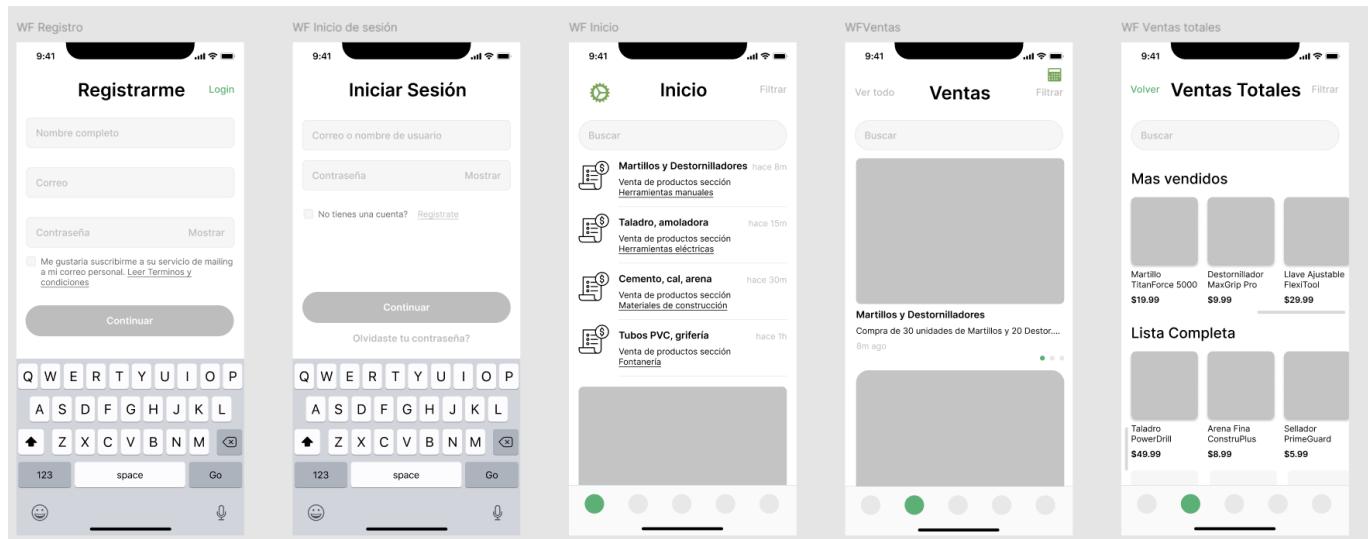


Escaneo de objetos, Stock IA:**Perfil:**

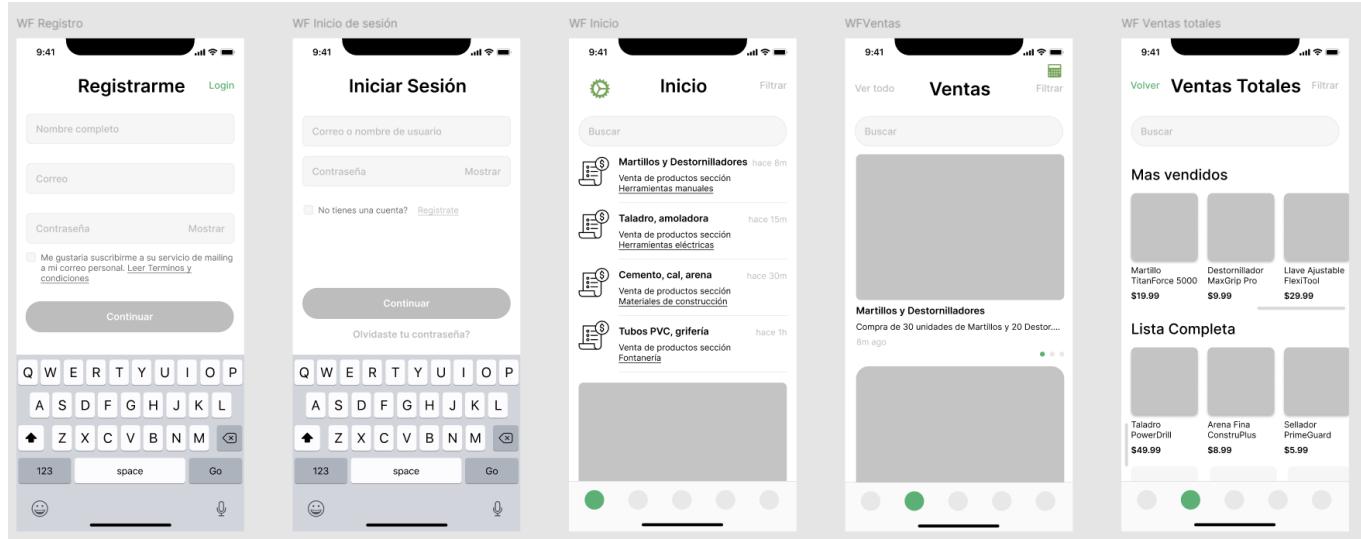


6.4.1. Applications Wireframes.

PAGINA 1:



PAGINA 2:



6.4.2. Applications Wireflow Diagrams.

Los siguientes diagramas de wireflow representan visualmente las pantallas y transiciones más relevantes de la aplicación móvil DETEKTO, diseñada para la gestión de ventas en ferreterías. A diferencia del user flow, los wireflows integran tanto la lógica de navegación como la disposición preliminar de los elementos visuales en cada vista, lo cual permite validar tempranamente la experiencia de usuario (UX) antes del diseño definitivo.

login-wireflow-mov

-Este wireflow refleja el proceso de autenticación en la aplicación DETEKTO. Comprende las pantallas de registro e inicio de sesión, incluyendo validaciones básicas y opciones como el acceso a términos y condiciones. Se prioriza una interfaz simple y clara para garantizar un acceso ágil desde dispositivos móviles.

The image displays two wireframe prototypes side-by-side, both titled "WF Registro" (WF Registration) at the top.

WF Registro (Left): Registrarme

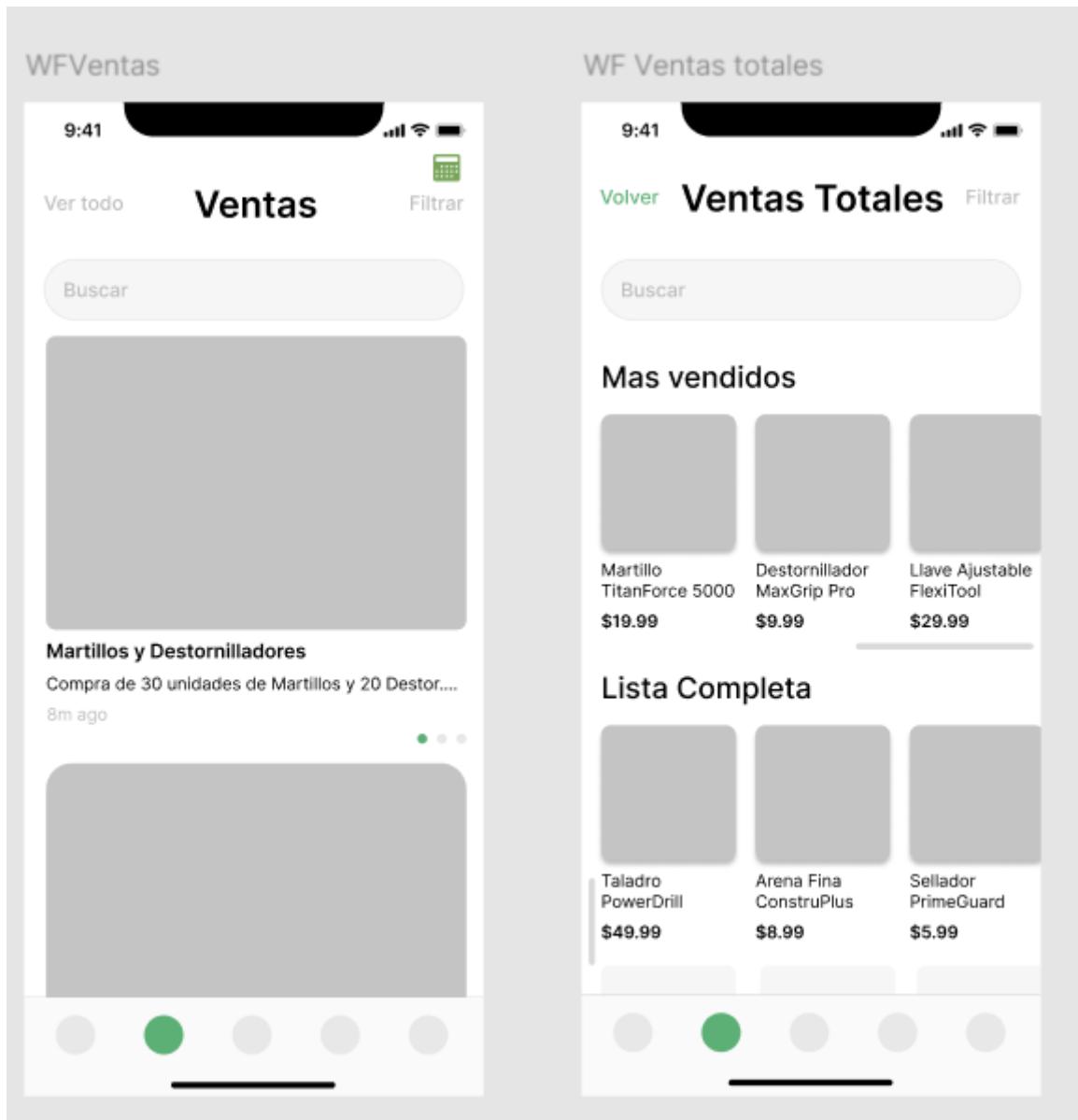
- Header: "Registrarme" and "Login" button.
- Text input fields: "Nombre completo" (Name), "Correo" (Email).
- Text input field: "Contraseña" (Password) with a "Mostrar" (Show) link.
- Checkboxes: "Me gustaría suscribirme a su servicio de mailing a mi correo personal. [Leer Terminos y condiciones](#)" (I would like to subscribe to your mailing service to my email. [Read Terms and Conditions](#)).
- Buttons: "Continuar" (Continue) and a large grey keyboard area below.

WF Inicio de sesión (Right): Iniciar Sesión

- Text input fields: "Correo o nombre de usuario" (Email or user name) and "Contraseña" (Password) with a "Mostrar" (Show) link.
- Text: "No tienes una cuenta? [Regístrate](#)" (Don't you have an account? [Register](#)).
- Buttons: "Continuar" (Continue) and "Olvidaste tu contraseña?" (Forgot your password?).
- Keyboard area: A standard QWERTY keyboard with "123", "space", and "Go" buttons, along with smiley face and microphone icons.

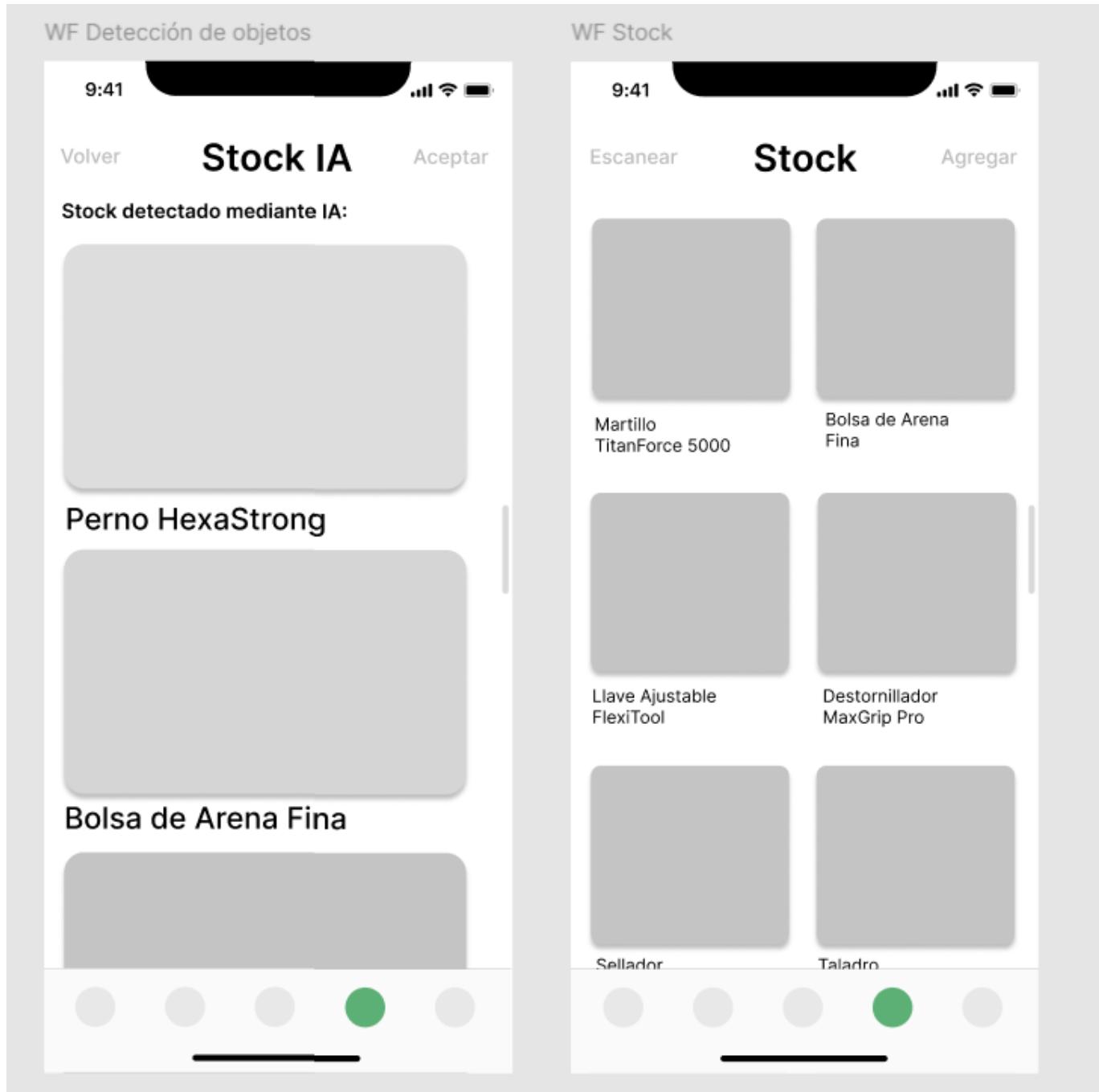
ventas-wireflow-mov

- Este flujo simula el proceso de revisión de ventas. Desde la pantalla de resumen hasta el detalle de productos más vendidos, permite al usuario filtrar por fecha o categoría y visualizar datos clave que apoyan la toma de decisiones en tiempo real.



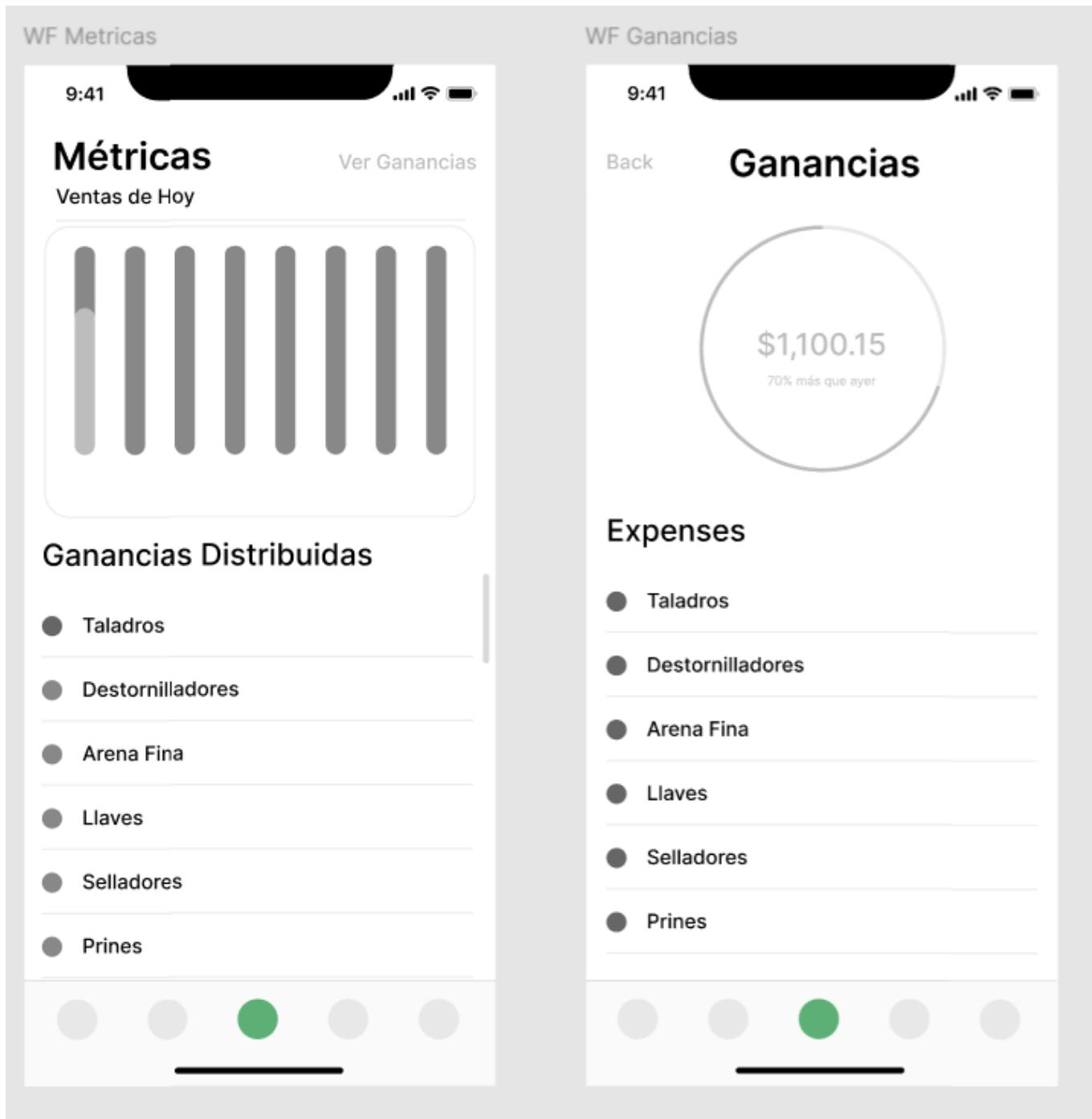
EscaneoIA-wireflow

- Uno de los elementos diferenciadores de DETEKTO es su capacidad de detectar productos mediante visión por computadora. Este flujo muestra cómo el usuario puede escanear el entorno de su almacén o tienda, y la app detecta automáticamente productos mediante IA, listando los artículos reconocidos y facilitando su registro en el inventario sin intervención manual.



métricas-wireflow-mov

- Aquí se representa la visualización de indicadores clave como ventas del día, productos con mayor rotación y distribución de ingresos. Este flujo permite al usuario analizar rápidamente el rendimiento del negocio y detectar oportunidades de mejora.



perfil-wireflow-mov

- Describe el flujo de navegación hacia la configuración del perfil del usuario. Se incluye la posibilidad de editar información personal y cerrar sesión, manteniendo una estructura simple y funcional.

