

COURSE PROJECT

Universidad Peruana de Ciencias Aplicadas



**Ingeniería de Software - 2025-2
Aplicaciones para Dispositivos Móviles - 12612
Profesor: David Gerardo Quevedo Velasco**

Informe del Trabajo Final

**Startup: Stoq
Producto: StockWise**

Team Members:

Member	Code
Almerco Rojas, Jocelyn Damaly	U20221g068
Sanchez Rios, Camila Cristina	U202210973
Chi Cruzatt, Kevin Jorge	U202313655
Oroncoy Almeyda, Alejandro Daniel	U202313397
Paucar Meneses, Jeremy Alion	U201719449

Abril, 2025

Registro de versiones del Informe

Versión	Fecha	Autor	Descripción de modificaciones
TB1	17/09/2025	<ul style="list-style-type: none">Jocelyn AlmercoCamila SanchezKevin ChiAlejandro OroncoyJeremy Paucar	<ul style="list-style-type: none">Capítulo I: PresentacionCapítulo II: Requirements Development and Software Solution DesignAvance de Conclusiones, Bibliografía y Anexos

Project Report Collaboration Insights

Link de repositorio del reporte: <https://github.com/upc-pre-202520-1ACC0238-12612-Stoq/Report>

Contenido

Student Outcome

Objetivos SMART

Capítulo I: Presentación

- 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 Development and Software Solution Design

- COURSE PROJECT
- Project Report Collaboration Insights
- Contenido
- Student Outcome
- Objetivos SMART
- Capítulo I: Presentación
 - 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
- Capítulo II: Requirements Development and Software Solution Design
 - 2.1. Competidores
 - 2.1.1. Análisis competitivo
 - 2.1.2. Estrategias y tácticas frente a competidores
 - Estrategias Generales de Posicionamiento
 - Estrategias Ofensivas frente a Competidores
 - Estrategias Defensivas
 - 2.2. Entrevistas
 - 2.2.1. Diseño de entrevistas
 - 2.2.2. Registro de entrevistas
 - 2.2.3. Análisis de entrevistas
 - 2.3. Needfinding
 - 2.3.1. User Personas
 - 2.3.2. User Task Matrix
 - 2.3.3. User Journey Mapping
 - 2.3.4. Empathy Mapping
 - 2.3.5. Ubiquitous Language
 - 2.4. Requirements specification
 - 2.4.1. User Stories
 - Technical Stories
 - Spike Story
 - 2.4.2. Impact Mapping
 - Segmento Objetivo 1: Bodegas especializadas por rubro
 - Segmento Objetivo 2: Startups y emprendedores en expansión con necesidades logísticas
 - 2.4.3. Product Backlog
 - 2.5. Strategic-Level Domain-Driven Design
 - 2.5.1. EventStorming
 - 2.5.1.1. Candidate Context Discovery
 - Step 1: Unstructured exploration
 - Step 2 : Timeline construction

- Step 3: Pain Points
- Step 4: Pivotal Points
- Step 5: Commands
- Step 6: Policies
- Step 7 & 8: Read Models & External Systems
- Step 9 & 10: Aggregates & Bounded Contexts
- 2.5.1.2. Domain Message Flows Modeling
- 2.5.1.3. Bounded Context Canvases
- 2.5.2. Context Mapping
- 2.5.3. Software Architecture
 - 2.5.3.1. Software Architecture Context Level Diagrams
 - 2.5.3.2. Software Architecture Container Level Diagrams
 - 2.5.3.3. Software Architecture Deployment Diagrams
- 2.6. Tactical-Level Domain-Driven Design
 - 2.6.1. Bounded Context: AlertStockManagement Context
 - 2.6.1.1. Domain Layer
 - 2.6.1.2. Application Layer
 - 2.6.1.3. Interface Layer
 - 2.6.1.4. Infrastructure Layer
 - 2.6.1.5. Bounded Context Software Architecture Component Level Diagrams
 - 2.6.1.6. Bounded Context Software Architecture Code Level Diagrams
 - 2.6.1.6.1. Bounded Context Domain Layer Class Diagrams
 - 2.6.2. Bounded Context: Iam Context
 - 2.6.2.1. Domain Layer
 - 2.6.2.2. Interface Layer
 - 2.6.2.3. Application Layer
 - 2.6.2.4. Infrastructure Layer
 - 2.6.2.5. Bounded Context Software Architecture Component Level Diagrams
 - 2.6.2.6. Bounded Context Software Architecture Code Level Diagrams
 - 2.6.2.6.1. Bounded Context Domain Layer Class Diagrams
 - 2.6.2.6.2. Bounded Context Database Design Diagram
 - 2.6.3. Bounded Context: Inventory Context
 - 2.6.3.1. Domain Layer
 - 2.6.3.2. Application Layer
 - 2.6.3.3. Interface Layer
 - 2.6.3.4. Infrastructure Layer
 - 2.6.3.5. Bounded Context Software Architecture Component Level Diagrams
 - 2.6.3.6. Bounded Context Software Architecture Code Level Diagrams
 - 2.6.3.6.1. Bounded Context Domain Layer Class Diagrams
 - Domain: Model
 - Domain: Services
 - 2.6.3.6.2. Bounded Context Database Design Diagram
 - 2.6.4.1. Domain Layer
 - 2.6.4.2. Application Layer
 - 2.6.4.3. Interface Layer
 - 2.6.4.4. Infrastructure Layer
 - 2.6.4.5. Bounded Context Software Architecture Component Level Diagrams
 - 2.6.4.6. Bounded Context Software Architecture Code Level Diagrams
 - 2.6.4.6.1. Bounded Context Domain Layer Class Diagrams
 - Domain: Model
 - Domain: Services
 - 2.6.4.6.2. Bounded Context Database Design Diagram
 - 2.6.5. Bounded Context: Reports Context
 - 2.6.5.1. Domain Layer
 - 2.6.5.2. Application Layer
 - 2.6.5.3. Interface Layer
 - 2.6.5.4. Infrastructure Layer
 - 2.6.5.5. Bounded Context Software Architecture Component Level Diagrams
 - 2.6.5.6. Bounded Context Software Architecture Code Level Diagrams
 - 2.6.5.6.1. Bounded Context Domain Layer Class Diagrams
 - Domain: Model
 - Domain: Services
 - 2.6.5.6.2. Bounded Context Database Design Diagram

- Capítulo III: Solution UI/UX Design
 - 3.1. Product design
 - 3.1.1. Style Guidelines
 - 3.1.1.1. General Style Guidelines
 - 3.1.2. Information Architecture
 - 3.1.2.1. Organization Systems
 - 3.1.2.2. Labelling Systems
 - 3.1.2.3. SEO Tags and Meta Tags
 - 3.1.2.4. Searching Systems
 - 3.1.2.5. Navigation Systems
 - 3.1.3. Landing Page UI Design
 - 3.1.3.1. Landing Page Wireframe
 - 3.1.3.2. Landing Page Mock-up
 - 3.1.4. Mobile Applications UX/UI Design
 - 3.1.4.1. Mobile Applications Wireframes
 - 3.1.4.2. Mobile Applications Wireflow Diagrams
 - 3.1.4.3. Mobile Applications Mock-ups
 - 3.1.4.4. Mobile Applications User Flow Diagrams
 - 3.1.4.5. Mobile Applications Prototyping
- Capítulo IV: Product Implementation & Validation
 - 4.1. Software Configuration Management
 - 4.1.1. Software Development Environment Configuration
 - 4.1.2. Source Code Management
 - 4.1.3. Source Code Style Guide & Conventions
 - 4.1.4. Software Deployment Configuration
 - 4.2. Landing Page & Mobile Application Implementation
 - 4.2.1. Sprint n
 - 4.2.1.1. Sprint Planning n
 - 4.2.1.2. Sprint Backlog n
 - 4.2.1.3. Development Evidence for Sprint Review
 - 4.2.1.4. Testing Suite Evidence for Sprint Review
 - 4.2.1.5. Execution Evidence for Sprint Review
 - 4.2.1.6. Services Documentation Evidence for Sprint Review
 - 4.2.1.7. Software Deployment Evidence for Sprint Review
 - 4.2.1.8. Team Collaboration Insights during Sprint
 - 4.3. Validation Interviews
 - 4.3.1. Diseño de Entrevistas
 - 4.3.2. Registro de Entrevistas
 - 4.3.3. Evaluaciones según heurísticas
- Conclusiones y recomendaciones
- Video App Validation
- Video About-the-Product
- Video About-the-team
- Bibliografía
- Anexos

Student Outcome

ABET - EAC - Student Outcome 7

Criterio: La capacidad de adquirir y aplicar nuevos conocimientos según sea necesario, utilizando estrategias de aprendizaje apropiadas.

En el siguiente cuadro se describe las acciones realizadas y enunciados de conclusiones por parte del grupo, que permiten sustentar el haber alcanzado el logro del ABET – EAC - Student Outcome 7.

Criterio específico	Acciones realizadas	Conclusiones
Actualiza conceptos y conocimientos necesarios para su desarrollo profesional y en especial para su proyecto en soluciones de software.	Jocelyn Almerco TB1: Reconocí que participar en la edición del video de entrevistas y en las herramientas de análisis (Capítulo II, matrices, mapeos y backlog) me exige	TB1: Como equipo logramos actualizar nuestros conocimientos al aplicar diferentes técnicas y herramientas en el proyecto, lo que nos permitió

un aprendizaje permanente para mejorar mi aporte en proyectos de software.

TP1:

..

TB2:

..

TF:

..

mejorar nuestra formación profesional y fortalecer el desarrollo de la solución de software.

TP1:

..

TB2:

..

TF:

..

Camila Sanchez Rios

TB1:

Me encargué de crear la organización y los repositorios en el github. Colaboré con Capítulo I: Presenatcion, Capítulo II: Entrevistas, Competidores

TP1:

TB2:

TF:

Kevin Chi

TB1:

Puse en práctica lo que sabía y reforcé mis conocimientos diseñando el C4 y el modelo de base de datos para el proyecto.

TP1:

TB2:

TF:

Alejandro Oroncoy

TB1:

Mejoré mis conocimientos aplicando el Tactical DDD y organizando el dominio del proyecto de una manera más clara.

TP1:

TB2:

TF:

Jeremy Paucar

TB1:

Aprendí nuevas formas de mapear el flujo de eventos y actualicé mi conocimiento aplicando EventStorming en el proyecto.

TP1:

TB2:

TF:

Reconoce la necesidad del aprendizaje permanente para el desempeño profesional y el desarrollo de proyectos en soluciones de software.

Jocelyn Almerco**TB1:**

Me encargué de editar el video de las entrevistas. Colaboré con Capítulo II: Entrevistas, User Task Matrix, User Journey Mapping, Empathy Mapping, Requirements specification, User Stories, Impact Mapping, Product Backlog.

TP1:

..

TB2:

..

TF:

..

Camila Sanchez**TB1:**

Me encargué de crear la organización y los repositorios en el github. Colaboré con Capítulo I: Presenatcion, Capítulo II: Entrevistas, Competidores

TP1:

..

TB2:

..

TF:**Kevin Chi****TB1:**

Me di cuenta de que al trabajar con C4 y el diagrama de base de datos necesito seguir aprendiendo sobre cómo representar bien la arquitectura y los datos de un sistema.

TP1:

..

TB2:

..

TF:**Alejandro Oroncoy****TB1:**

Comprendí que usar las tácticas de DDD requiere aprendizaje constante de patrones como entidades y agregados para aplicarlos bien en el dominio.

TP1:**TB1:**

Es importante contar con la capacidad de trabajar en equipo, ya que permite compartir ideas, resolver problemas juntos y mejorar los resultados del proyecto.

TP1:

...

TB2:

...

TF:

...

...

TB2:

...

...

TF:

...

Jeremy Paucar

TB1:

Entendí que al hacer EventStorming es clave seguir capacitándome para identificar de forma correcta los eventos y procesos del sistema.

TP1:

...

TB2:

...

TF:

...

Objetivos SMART

Capítulo I: Presentación

1.1. Startup Profile

1.1.1. Descripción de la Startup

StockWise es un aplicativo móvil de gestión de inventarios dirigida a pequeñas y medianas empresas, startups y bodegas especializadas. Su objetivo principal es facilitar el control eficiente de entradas y salidas de productos, la gestión de usuarios, la configuración de alertas inteligentes, la generación de reportes detallados y de boleta de venta, todo a través de una interfaz intuitiva y accesible desde cualquier dispositivo.

La solución responde a una problemática concreta: muchos negocios aún utilizan métodos manuales (como hojas de cálculo o registros en papel) para administrar sus inventarios y pagos, lo cual genera errores, desorganización, sobrecompras y pérdidas económicas. StockWise busca resolver este problema digitalizando y centralizando el control del inventario y de las ganancias de la tienda, permitiendo a los negocios tomar decisiones basadas en datos y optimizar su operación mediante cuatro planes de funcionalidad avanzada:

- **Plan A – Entrada por voz:** Permite registrar productos mediante comandos de voz (ejemplo: “Aregar 20 botellas de agua al inventario”), ofreciendo mayor rapidez y comodidad, especialmente en situaciones donde el personal no pueda usar el teclado.
- **Plan B – Geolocalización (GPS):** Integra funciones de ubicación para optimizar la trazabilidad y distribución de productos, permitiendo identificar su procedencia, registrar puntos de entrega y visualizar en un mapa interactivo las distintas sedes o sucursales vinculadas a la tienda principal.
- **Plan C – Localización y predicción inteligente:**
 - Localiza: Utiliza un mapa interactivo con integración de códigos QR para ubicar productos dentro del almacén de manera precisa.
 - Predice: Implementa un sistema de reabastecimiento inteligente basado en patrones de ventas, que sugiere cuándo y cuánto stock reponer para evitar quiebres de inventario.
- **Plan D – Escaneo por lotes con cámara rápida:** En lugar de códigos de barras, la app toma una foto del producto o lote y una API de visión (como Google ML Kit) devuelve etiquetas genéricas. El usuario confirma el producto exacto y la cantidad en un menú antes de registrarla, pudiendo ver directamente su ubicación en el almacén virtual.

Misión: Brindar a pequeñas y medianas empresas una solución de gestión de inventarios y pagos sencilla, accesible y eficiente, que les permita digitalizar su operación, reducir errores logísticos y tomar decisiones basadas en datos reales, apoyando así su crecimiento sostenible.

Visión: StockWise se enfoca en ser una aplicación preferida de los empresarios que buscan una gestión de inventarios, herramientas de alto valor con un modelo escalable, adaptable y centrado en el usuario.

1.1.2. Perfiles de integrantes del equipo



Jocelyn Damaly Almerco Rohas

Soy estudiante de Ingeniería de Software y actualmente curso el 5to ciclo. En mi tiempo libre me gusta tocar ukelele, leer libros, resolver sudoku, escuchar música y ver series.



Jeremy Paucar Meneses

Estudiante de Ingeniería de Software, apasionado a programar, tengo conocimientos en c++, js, c#, y algunos frameworks como Vue, dentro de mis pasatiempos esta escuchar musica, ver futbol y ver series



Sánchez Ríos, Camila Cristina

Soy estudiante de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas, actualmente me encuentro en el octavo ciclo. Me gusta escuchar música y leer en los ratos libres y aprender más sobre la carrera.



Chi Cruzatt, Kevin Jorge

Actualmente estoy cursando la carrera de ingeniería de Software y me encuentro en el 6º ciclo. Tengo experiencia utilizando diferentes metodologías de diseño, como Domain Driven Design (DDD) y CMV, además de frameworks como Laravel y Spring en lenguajes como Java, PHP, etc. Me considero una persona paciente y perseverante en situaciones difíciles.

Oroncoy Almeyda, Alejandro Daniel

Mi nombre es Alejandro Oroncoy. Tengo 19 años, soy estudiante de la carrera de ingeniería de software, estoy en 6to ciclo. Me considero una persona proactiva, autodidacta y orientada a objetivos.



1.2. Solution Profile

1.2.1 Antecedentes y problemática

- **Who (¿Quiénes?)**

Emprendedores, startups y pequeñas/medianas empresas con bodegas físicas que almacenan productos de distintos rubros como ropa, calzado, electrodomésticos, ferretería o alimentos.

- **What (¿Qué sucede?)**

A medida que sus negocios escalan, la gestión manual del inventario con hojas de cálculo o registros físicos se vuelve ineficiente, generando pérdidas, errores, quiebres de stock, sobrecompras y desorden logístico.

- **When (¿Cuándo ocurre?)**

En el momento en que el negocio empieza a crecer, aumentar su variedad de productos o abrir múltiples canales de venta, como tiendas físicas y plataformas online.

- **Where (¿Dónde ocurre?)**

En bodegas físicas propias, espacios alquilados o incluso en el hogar del emprendedor, especialmente en etapas tempranas o de expansión del negocio.

- **Why (¿Por qué es un problema?)**

La falta de un sistema centralizado y en tiempo real impide tomar decisiones estratégicas basadas en datos. Esto afecta la planificación de compras, genera pérdidas económicas, y daña la experiencia del cliente final.

- **How (¿Cómo lo solucionan hoy?)**

Mediante herramientas manuales como cuaderno o hojas de papel, inventarios escritos o software no especializado, que resultan limitados, propensos a errores y poco escalables.

- **How much (¿Cuánto cuesta no resolverlo?)**

El costo se traduce en pérdidas económicas significativas por productos no vendidos, errores de stock, tiempo invertido en tareas manuales y menor competitividad frente a negocios más organizados.

1.2.2 Lean UX Process

1.2.2.1. Lean UX Problem Statements

El propósito de StockWise es proporcionar una aplicación intuitiva y accesible para que pequeñas y medianas empresas, startups y bodegas especializadas puedan gestionar su inventario de forma eficiente, digitalizada y sin complicaciones técnicas ni altos costos.

El problema se encuentra en que muchos de estos negocios aún dependen de métodos manuales como hojas de cálculo, cuadernos físicos o herramientas no especializadas para registrar y controlar su stock. Esto genera errores frecuentes, pérdidas de productos, compras innecesarias, falta de trazabilidad y escasa visibilidad sobre la operación logística.

Hemos observado que esta situación impacta negativamente en la productividad del negocio, la satisfacción del cliente y la capacidad de tomar decisiones basadas en datos. A medida que estos negocios escalan, el desorden operativo se vuelve insostenible, provocando sobrecostos y

afectando su crecimiento.

¿Cómo podríamos diseñar una aplicación de gestión de inventarios que sea lo suficientemente simple, funcional y adaptable para cubrir las necesidades reales de estos negocios en expansión, facilitando el control del inventario, reduciendo errores y mejorando la toma de decisiones, con métricas que midan eficiencia operativa, precisión del stock y satisfacción del usuario?

1.2.2.2. Lean UX Assumptions

Business Assumptions:

1. Creemos que los negocios emergentes necesitan digitalizar su gestión de inventarios a través de una solución móvil accesible.
2. Estas necesidades se pueden satisfacer con una aplicación móvil intuitiva, escalable y de bajo costo.
3. Nuestros clientes iniciales serán emprendedores, startups y pequeñas empresas con bodegas especializadas que operan de manera ágil.
4. El principal valor que busca el cliente es tener control y visibilidad total de su inventario en tiempo real, desde cualquier lugar.
5. El cliente obtendrá además alertas inteligentes, reportes automáticos, generación de boletas de venta y una experiencia de usuario móvil optimizada.
6. Adquiriremos la mayoría de nuestros clientes mediante publicidad en redes sociales dirigida, ASO (Optimización de la App Store) y asociaciones con comunidades de emprendedores.
7. Nuestro modelo de ingresos se basará en un esquema freemium dentro de la app, con upgrade a planes premium que desbloqueen funcionalidades avanzadas.
8. Nuestra competencia directa e indirecta incluye aplicaciones genéricas como hojas de cálculo móviles (Excel, Sheets), recordatorios básicos (Calendar) y ERPs complejos con apps móviles poco intuitivas.
9. Nuestra ventaja competitiva radicará en la simplicidad móvil, el enfoque específico en las pymes y la especialización en la gestión de inventarios sobre la marcha.
10. El mayor riesgo para el negocio es una baja tasa de adopción o la percepción de que la app es compleja o redundante frente a métodos manuales.
11. Mitigaremos este riesgo realizando pruebas de usabilidad móvil con usuarios reales, iteraciones rápidas basadas en feedback y una estrategia de onboarding dentro de la app que guíe al usuario paso a paso.

User Assumptions:

- **¿Quién es el usuario?** Dueños de negocios, encargados de bodegas o logística en pymes/startups que necesitan gestionar inventarios de forma remota.
- **¿Qué problemas busca resolver nuestro producto?** La falta de control inmediato del stock, los errores por registros manuales en papel, los sobrecostos por pérdidas y las ventas fallidas debido a quiebres de stock inesperados.
- **¿Qué características son importantes?** Un registro rápido de productos (por voz, cámara o manual), un historial de movimientos accesible, alertas push personalizables, reportes visuales simplificados y la generación de boletas de venta directamente desde el dispositivo móvil.
- **¿Dónde encaja nuestro producto en su trabajo?** Se integra en su flujo de trabajo diario para la gestión del inventario en la bodega, en el punto de venta o durante las rondas de reposición, permitiendo decisiones informadas desde cualquier lugar.
- **¿Cuándo y cómo es usado nuestro producto?** Se utiliza de manera frecuente a lo largo del día, directamente desde sus teléfonos inteligentes, para registrar entradas/salidas al momento, consultar niveles de stock en tiempo real y generar comprobantes al instante.
- **¿Cómo debe verse y comportarse?** Debe tener una interfaz de usuario (UI) móvil limpia, simple y con navegación táctil intuitiva. Debe ser rápida, responsive y funcionar sin conexión para casos de uso críticos.

Feature Assumptions:

Creemos que la aplicación móvil debe contar con una interfaz táctil intuitiva y diseños adaptados a dispositivos móviles que permitan a emprendedores y administradores adoptarla sin dificultad, minimizando la curva de aprendizaje.

Creemos que la app debe proporcionar notificaciones push y alertas personalizables (por stock bajo, fechas de vencimiento o pedidos de proveedores) que mantendrán a los usuarios informados de manera proactiva para prevenir errores operativos.

Creemos que el sistema debe incluir un módulo de reportes y dashboards visuales optimizados para móvil que permitan visualizar de un vistazo métricas clave (productos más vendidos, niveles de stock, tendencias), facilitando la toma de decisiones ágiles.

1.2.2.3. Lean UX Hypothesis Statements

Hipótesis 1: Creemos que los usuarios valorarán las alertas inteligentes push.

Sabremos que es cierto.

Cuando al menos el 80% de los usuarios activos tengan las notificaciones activadas y reporten una reducción en quiebres de stock en encuestas de satisfacción.

Hipótesis 2: Creemos que los usuarios encontrarán valiosos los reportes visuales optimizados para móvil.

Sabremos que es cierto.

el módulo de reportes muestre una tasa de uso semanal superior al 40% y contribuya a un aumento del 20% en la retención de usuarios.

Hipótesis 3: Creemos que ofrecer una versión gratuita con funcionalidades básicas incentivará la adopción inicial y las conversiones a premium.

Sabremos que es cierto

si la tasa de conversión de usuarios gratuitos a premium supera el 5% y el tiempo hasta la conversión es menor a 90 días.

Hipótesis 4: Creemos que implementar alertas inteligentes por stock bajo y fechas de vencimiento reducirá significativamente los quiebres de inventario.

Sabremos que tenemos razón

Cuando los usuarios que activan las alertas experimenten una reducción del 30% en incidentes de stock cero en un período de 3 meses.

Hipótesis 5: Creemos que implementar la generación de boletas de venta directamente en la app agilizará el proceso de cierre de ventas y el control de ingresos.

Sabremos que tenemos razón,

Cuando los usuarios que utilizan la función reporten un ahorro de tiempo mínimo de 1 hora diaria en administración y un control más preciso de su flujo de caja diario.

1.2.2.4. Lean UX Canvas

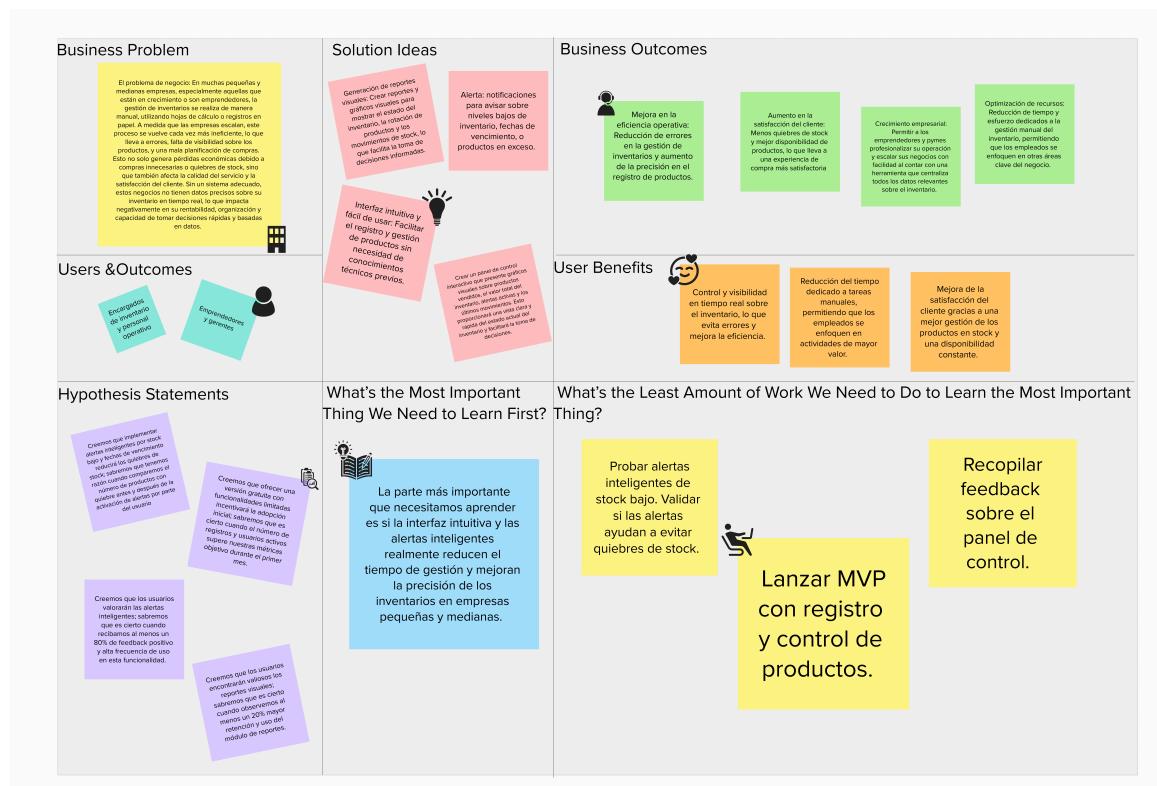


Imagen (Nº1). Elaboración propia. Realizado en Canva

Capítulo II: Requirements Development and Software Solution Design

2.1. Competidores

2.1.1. Análisis competitivo

Objetivo del análisis: Identificar el posicionamiento competitivo de StockWise en el mercado de aplicaciones móviles de gestión de inventarios para pymes, entendiendo las ventajas diferenciales y oportunidades de mejora.				
	Zoho Inventory 	Odoo Inventory 	TradeGecko 	
PERFIL				
Overview	Solución móvil integral especializada en gestión de inventarios para pymes y emprendedores	Suite de negocio completa con módulo de inventario como parte de su ecosistema	Software de gestión open-source con módulo de inventario personalizable	Plataforma de gestión de inventario centrada en comercio electrónico y retail
Ventaja competitiva	Especialización móvil, interfaz intuitiva, planes funcionales específicos (voz, GPS, IA) y enfoque en pymes	Integración completa con ecosistema Zoho y terceros, robustez funcional	Flexibilidad y personalización por ser open-source	Fuerte integración con canales de e-commerce y plataformas de venta
PERFIL DE MARKETING				
Mercado objetivo	Pymes, emprendedores, startups y bodegas especializadas	Medianas empresas y pymes en crecimiento	Empresas de todos los tamaños que buscan personalización	Negocios de e-commerce y retail
Estrategias de marketing	Marketing digital, ASO, comunidades de emprendedores, modelo freemium	Marketing B2B, fuerza de ventas, contenido educativo	Comunidad open-source, partners de implementación	Marketing especializado en e-commerce, ferias sectoriales
PERFIL DE PRODUCTO				
Productos & Servicios	App móvil con: gestión inventory, alertas, reportes, boletas, planes voz/GPS/IA/scanner	Software web y móvil con: inventory, órdenes venta, envíos, facturación	Suite modular web/móvil con: inventory, manufactura, ventas, CRM	Plataforma web/móvil: inventory, pedidos, logística, analytics
Precios & Costos	Modelo freemium con planes premium escalables por funcionalidad	Planes por volumen de órdenes, desde gratis hasta enterprise	Freemium modular, costos de implementación y personalización	Planes premium por volumen de ventas y usuarios
Canales de distribución	Móvil (iOS/Android)	Web y móvil (iOS/Android)	Web y móvil (iOS/Android)	Web y móvil (iOS/Android)

Análisis SWOT Fortalezas - Especialización móvil nativa

- Interfaz intuitiva
- Funcionalidades innovadoras (voz, GPS, IA)
- Enfoque específico en pymes - Ecosistema integrado

- Solución completa
- Reconocimiento de marca
- Escalabilidad - Flexibilidad y personalización
- Comunidad activa
- Modelo open-source
- Múltiples módulos integrados - Especialización en e-commerce
- Integraciones con plataformas de venta
- Analytics avanzado **Debilidades** - Marca desconocida
- Ecosistema limitado vs competidores
- Recursos limitados vs grandes competidores - Curva de aprendizaje más pronunciada
- Puede ser complejo para pymes muy pequeñas
- Costo elevado para planes avanzados - Requiere personalización para funcionar óptimamente
- Soporte puede ser limitado en versión community - Enfoque muy específico en e-commerce
- Menos adaptable a otros modelos de negocio
- Precios elevados **Oportunidades** - Crecimiento mercado pymes digitales
- Tendencia hacia soluciones móviles
- Expansión a Latinoamérica
- Alianzas con comunidades de emprendedores - Expansión a nuevos mercados
- Desarrollo de más integraciones
- Consolidación como suite integral - Crecimiento de la comunidad de desarrolladores
- Expansión de funcionalidades móviles
- Alianzas con implementadores - Crecimiento del e-commerce global
- Expansión a nuevos verticales de retail
- Desarrollo de analytics predictivo **Amenazas** - Entrada de competidores establecidos al mercado móvil
- Commoditización de funcionalidades básicas
- Dificultad para captar usuarios frente a alternativas gratuitas - Competencia de soluciones más especializadas
- Precios más competitivos de alternativas
- Cambios en regulaciones internacionales - Fragmentación de la comunidad
- Calidad variable de implementaciones
- Competencia de soluciones SaaS más pulidas - Competencia de plataformas de e-commerce con módulos nativos
- Cambios en APIs de plataformas de venta
- Consolidación del mercado

2.1.2. Estrategias y tácticas frente a competidores

Estrategias Generales de Posicionamiento

1. Especialización en Experiencia Móvil

Objetivo: Diferenciarnos como la solución móvil nativa más intuitiva del mercado.

Tácticas:

- Desarrollar funcionalidades exclusivas móviles (voz, GPS, escaneo por cámara)
- Optimizar la interfaz para uso táctil con gestos intuitivos
- Garantizar funcionamiento offline para operaciones críticas
- Implementar notificaciones push contextuales y personalizables

2. Enfoque en Usabilidad para PYMES

Objetivo: Ofrecer la curva de aprendizaje más corta del mercado.

Tácticas:

- Crear sistema de onboarding guiado dentro de la aplicación
- Desarrollar tutoriales interactivos por funcionalidad
- Implementar asistente inteligente para tareas complejas
- Diseñar plantillas preconfiguradas por tipo de negocio

Estrategias Ofensivas frente a Competidores

1. Contra Zoho Inventory

Debilidad a explotar: Complejidad para PYMES pequeñas

Tácticas:

- Campañas comparativas destacando simplicidad vs complejidad

- Programa de migración asistida desde Zoho
- Planes precio más competitivos para segmento pequeño-mediano
- Funcionalidades "justo lo necesario" sin sobrecarga de opciones

2. Contra Odoo Inventory

Debilidad a explotar: Requiere personalización e implementación

Tácticas:

- Mensajería centrada en "funciona desde el día 1"
- Precios transparentes sin costos ocultos de implementación
- Casos de éxito de implementación en menos de 24 horas
- Servicio de configuración inicial incluido en planes premium

3. Contra TradeGecko

Debilidad a explotar: Enfoque limitado a e-commerce

Tácticas:

- Posicionamiento como solución multi-canal integral
- Funcionalidades para negocios físicos + digitales
- Campañas dirigidas a retail tradicional con interés en digitalizarse
- Precios más accesibles para negocios mixtos

Estrategias Defensivas

1. Ante posible commoditización

Tácticas:

- Desarrollo continuo de funcionalidades innovadoras (voz, IA, realidad aumentada)
- Programa de fidelización con beneficios escalables
- Integraciones exclusivas con proveedores locales
- Comunidad de usuarios para feedback y co-creación

2. Protección frente a entrada de grandes competidores

Tácticas:

- Alianzas estratégicas con asociaciones de PYMES
- Contratos a largo plazo con precios congelados
- Desarrollo de especialización vertical por industria
- Branding fuerte centrado en el enfoque PYME

2.2. Entrevistas

La finalidad de realizar entrevistas es obtener un alcance más completo sobre las experiencias, perspectivas y opiniones de los segmentos de mercado definidos. Nuestro objetivo es recolectar información valiosa que nos permita conocer mejor a nuestro público objetivo. Con estas entrevistas se tendrá una visión más clara de las necesidades y expectativas de nuestros usuarios.

2.2.1. Diseño de entrevistas

Segmento #1: Bodegas especializadas por rubro

Preguntas principales

1. ¿Podrías contarme cómo gestionas actualmente el inventario de tu bodega?
2. ¿Cuáles son los mayores retos que enfrentas al momento de organizar tus productos?
3. ¿Has tenido pérdidas o problemas por errores en el inventario? ¿Cómo los resolviste?
4. ¿Qué tan importante es para ti tener un control en tiempo real de tu stock?
5. ¿Utilizas algún sistema o herramienta digital? ¿Cuál y cómo te va con ella?

Preguntas complementarias

1. ¿Cómo te enteras cuando falta un producto o está a punto de vencerse?
2. ¿Qué tipo de reportes o información te gustaría tener sobre tu inventario?
3. ¿Qué dispositivos usas más en tu trabajo (laptop, celular, tablet)?
4. ¿Cómo crees que una plataforma digital podría ayudarte a mejorar tu operación?

Segmento #2: Startups y emprendedores en expansión con necesidades logísticas

Preguntas principales

1. ¿Cómo manejas actualmente el inventario de tu negocio?
2. ¿En qué momentos has sentido que el control del stock te limita o te hace perder tiempo?
3. ¿Cómo llevas el registro de entradas y salidas de productos?
4. ¿Qué te gustaría mejorar de tu proceso logístico actual?
5. ¿Has considerado usar alguna plataforma para gestionar tu inventario? ¿Por qué sí o por qué no?

Preguntas complementarias

1. ¿Qué herramientas digitales usas en tu negocio actualmente?
2. ¿Dónde almacenas tus productos?
3. ¿Qué tan seguido necesitas revisar el stock?
4. ¿Qué redes sociales o canales digitales usas para vender?

2.2.2. Registro de entrevistas

Segmento 1: Bodegas especializadas por rubro

Entrevista	1	Nombre	Milagros Almerco
Edad	28	Distrito	San Martín de Porres
Captura de la entrevista:	En la entrevista, la estudiante Milagros Almerco comenta que lleva el control de su bodega en un cuaderno, anotando manualmente las ventas y compras. Reconoce que su sistema es básico y que a veces se olvida de registrar productos, lo que dificulta llevar un control claro del inventario. Además, le resulta complicado encontrar información específica entre tantas hojas. Señala como uno de sus principales problemas la falta de control sobre las fechas de vencimiento, lo que ha ocasionado pérdidas por productos que no logró vender a tiempo. También menciona que no siempre actualiza los cambios de precio. Para prevenir esto, trata de revisar los estantes y adelantar los productos por vencer. Respecto a herramientas digitales, indica que usa el celular y la calculadora por practicidad. Pero le gustaría contar con un sistema que le avise qué productos están por vencer, cuáles debe reponer y cómo están cambiando los precios. Además, considera importante poder visualizar su inventario en tiempo real. Le gustaría recibir alertas y reportes que le ayuden a organizar mejor su negocio.		
URL de la grabación	Ver grabación		
Timing	00:00 -5:48		

Entrevista	2	Nombre	Elvis Aranga Mesa
Edad	31	Distrito	Santiago de Surco
Captura de la entrevista:	Elvis Aranga Meza, de 31 años y encargado de la administración en su bodega en Santiago de Surco, gestiona el inventario de forma manual, sin un sistema digital, basándose en el control visual y el registro de ventas totales. Reconoce que esta falta de control sistematizado dificulta hacer proyecciones, identificar los productos con mayor rotación o margen de ganancia, y manejar adecuadamente las mermas, especialmente en productos perecibles como frutas y abarrotes. Ha enfrentado pérdidas por productos vencidos y robos, y señala la necesidad de implementar un sistema digital para actualizar precios en tiempo real, controlar vencimientos y facilitar la gestión por parte del personal. Considera que una plataforma digital mejoraría la eficiencia, permitiría delegar tareas y evitar errores en la rotación y precios, aportando mayor libertad y control en la operación diaria.		
URL de la grabación	Ver grabación		
Timing	5:48 - 11:47		
Entrevista	3	Nombre	Catalina Villa Guerra
Edad	28	Distrito	Breña
Captura de la entrevista:	Catalina Guerra, de 28 años encargada de administrar una bodega familiar en el distrito de Breña, ella cuenta que gestiona la bodega de manera manual y lo pasa a una tabla de excel, también nos comenta lo difícil que es tener control de todas las ventas, ella menciona mucho algunas dificultades cuando llega nuevos lotes de productos, ya que no se lleva un control estricto de esto, por otro lado, ella hace enfasis de las perdidas monetarias que enfrenta por no poder gestionar bien las entradas y salidas de productos, en este contexto también nos cuenta sobre la expiración de algunos producto que ella no puede gestionar de manera correcta y por lo cual le va generando muchas perdidas. Ella considera que si tuviera acceso a una plataforma orientada a su rubro de trabajo podría solventar algunos de los problemas ya mencionados anteriormente y también de esta manera dejar de lado el uso de apuntes manuales que es pesado. También dice que le permitiría delegar de mejor manera los inventarios.		

URL de la grabación	Ver grabación
Timing	11:48 - 21:31

Segmento 2: Startups y emprendedores en expansión con necesidades logísticas

Entrevista	4	Nombre	Juan Carlos Ramírez
Edad	49	Distrito	Surquillo
Captura de la entrevista:	Juan Carlos Ramírez, emprendedor de 49 años, gestiona su inventario principalmente de forma manual, registrando entradas y salidas mediante boletas y facturas sin un control digital completo. Actualmente está en proceso de digitalizar su sistema con un programa y una máquina POS que escanea códigos de barra, aunque aún no emite boletas ni facturas electrónicas. La revisión del stock se realiza visualmente y las ventas se gestionan principalmente por WhatsApp con entrega a domicilio. Busca mejorar la eficiencia y liberar tiempo mediante la implementación de herramientas digitales más integradas.		
	Ver grabación		
Timing	21:31 - 25:14		
Entrevista	5	Nombre	Leonardo Gamboa
Edad	26	Distrito	San Miguel
Captura de la entrevista:	Leonardo G., emprendedor, maneja actualmente su inventario de forma manual utilizando principalmente hojas de Excel y, en ocasiones, una libreta para registrar entradas y salidas de productos. La actualización no siempre está al día, lo que le genera retrasos al atender pedidos y, en algunos casos, ventas de productos agotados que afectan la experiencia de sus clientes. Sus ventas se coordinan principalmente a través de WhatsApp, mientras que la promoción la realiza en Facebook e Instagram. Revisa su stock casi a diario debido a la demanda constante y almacena sus productos en un cuarto adaptado en su casa. Aunque ha considerado implementar plataformas digitales para gestionar su inventario, aún no lo hace por temor a los costos y a la complejidad de uso. Su principal objetivo es contar con un sistema más rápido y automático que le avise cuando se esté quedando sin stock y que le permita organizar mejor los pedidos, optimizando así su tiempo y eficiencia.		
	Ver grabación		
Timing	25:14 - 29:36		
Entrevista	6	Nombre	Gael Rivera
Edad	xx	Distrito	xx
Captura de la entrevista:	La entrevista trata sobre Eduardo de Rivera Sosa, un emprendedor que importa productos desde Asia para venderlos en Perú. Actualmente, gestiona su inventario manualmente con Excel, lo que le consume mucho tiempo y recursos, generando pérdidas. Busca una solución automatizada y fácil de usar para controlar mejor su stock y evitar la necesidad de estar actualizando constantemente la información. También usa Notion para organizar pedidos y vende a través de Facebook, Instagram y TikTok. En resumen, la entrevista aborda los retos y necesidades en la gestión de inventarios en su negocio de importación y venta.		
	Ver grabación		
Timing	29:36 - 33:36		

2.2.3. Análisis de entrevistas

Segmento 1: Bodegas especializadas por rubro

Los entrevistados (Milagros, Elvis, Catalina) gestionan inventarios de forma manual, con apuntes en cuadernos o Excel, lo que genera dificultades para mantener un control preciso y actualizado. Las principales problemáticas incluyen falta de control sobre fechas de vencimiento, pérdidas por productos no vendidos a tiempo, ausencia de automatización, y falta de sistemas que permitan actualizar precios en tiempo real o visualizar el inventario de manera dinámica. Los entrevistados manifiestan la necesidad de plataformas digitales con alertas y reportes para mejorar la gestión, delegar tareas y reducir errores humanos, lo que apuntaría a una alta demanda por soluciones tecnológicas accesibles y específicas para bodegas tradicionales.

Segmento 2: Startups y emprendedores en expansión con necesidades logísticas

Los entrevistados Juan Carlos, Leonardo y Eduardo manejan inventarios mayormente manuales o semiautomatizados (uso de Excel, cuadernos, y sistemas en proceso de digitalización). Tienen una mayor actividad y volumen, con operaciones en crecimiento, y venden usando canales digitales

como WhatsApp y redes sociales. Aunque reconocen la importancia de digitalizar y automatizar sus procesos para ganar eficiencia, enfrentan obstáculos como la complejidad o altos costos de las soluciones disponibles. Buscan sistemas simples, automáticos, que avisen sobre niveles de stock y faciliten la organización para poder escalar su negocio sin perder control.

Ambos segmentos comparten retos similares: control manual, dificultades con productos perecibles, pérdidas por falta de actualización o visibilidad, y la necesidad de automatización. Sin embargo, los bodegueros tradicionales tienen una menor adopción y experiencia con herramientas digitales y requieren soluciones accesibles y adaptadas a su realidad, mientras que las startups muestran interés en herramientas integradas y avanzadas, pero preocupadas por la complejidad y costos. Una solución escalable, intuitiva y que pueda adaptarse a distintos tamaños de negocios sería ideal para cubrir ambos mercados.

2.3. Needfinding

En el siguiente apartado, analizaremos a nuestros segmentos objetivos para identificar sus necesidades y en base a esto ofrecerles soluciones óptimas a sus problemas.

2.3.1. User Personas

Segmento 1: Bodegas especializadas por rubro

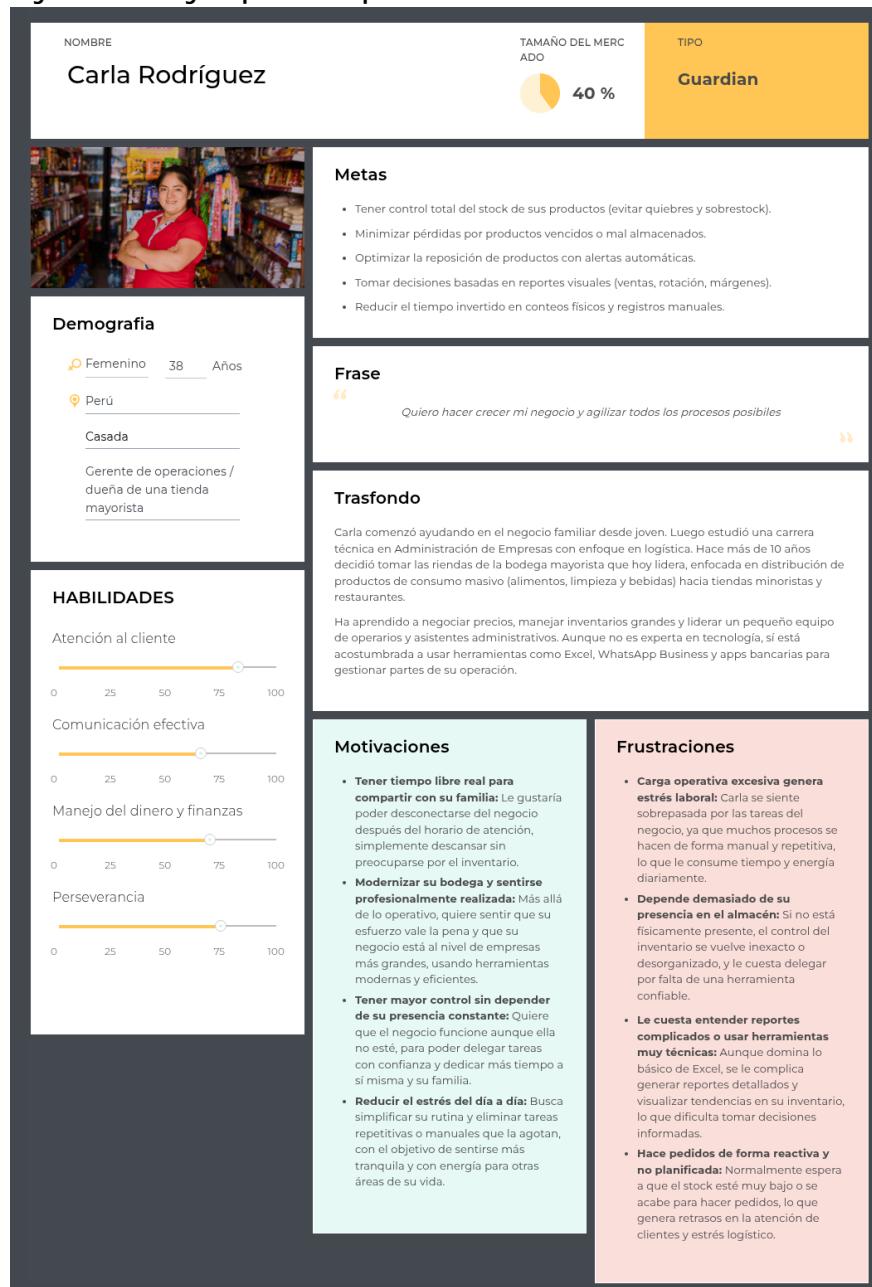


Imagen (N°2). Elaboración propia. Realizado en UXPressia

Segmento 2: Startups y emprendedores en expansión con necesidades logísticas

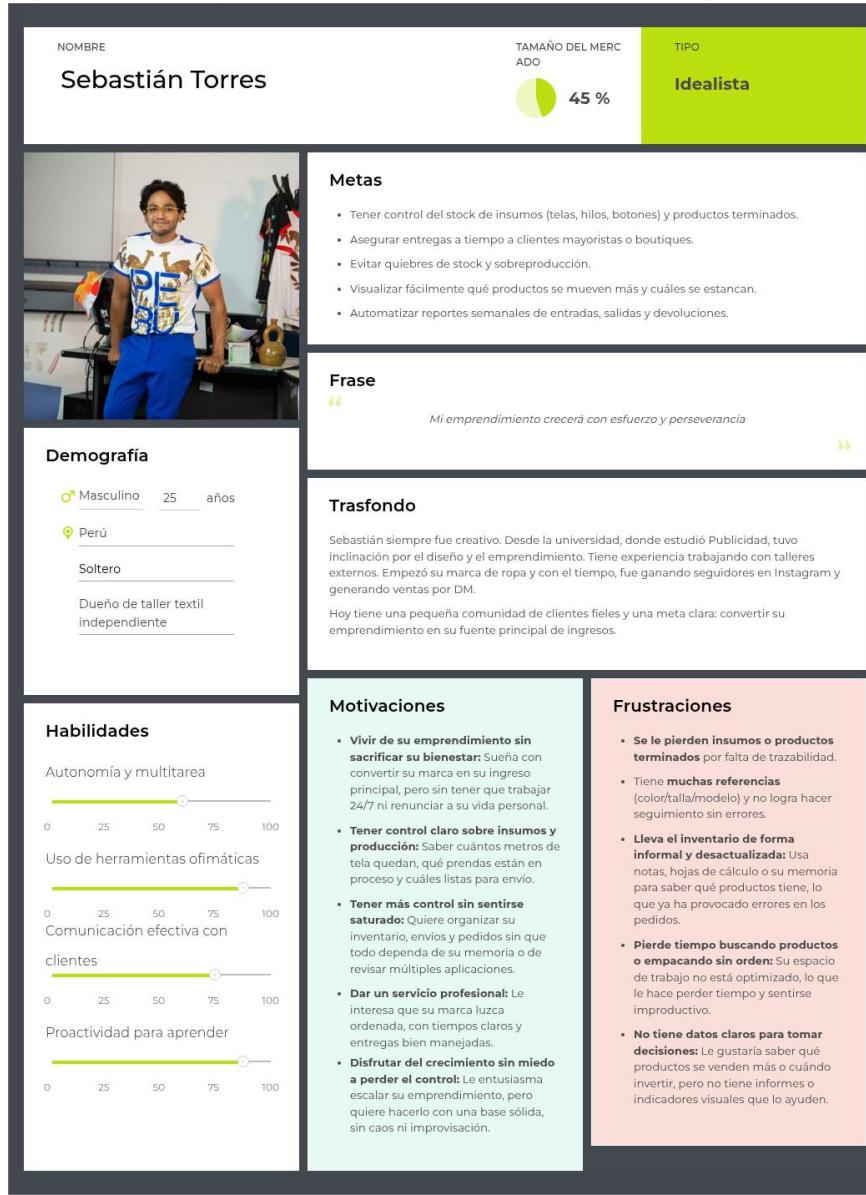


Imagen (Nº3). Elaboración propia. Realizado en UXPressia

2.3.2. User Task Matrix

Segmento 1: Bodegas especializadas por rubro

Task Matrix	Frecuencia	Importancia
Supervisar el stock y revisar niveles de inventario	Alta	Alta
Realizar conteos físicos o auditorías manuales	Media	Alta
Negociar precios y coordinar con proveedores	Alta	Alta
Revisar reportes de ventas, rotación y márgenes	Media	Alta
Ingresar datos en Excel o sistemas básicos de control	Media	Media
Delegar tareas a sus asistentes o empleados	Media	Alta
Atender clientes en tienda	Alta	Alta
Coordinar pedidos con mayoristas o distribuidores	Alta	Alta

Task Matrix	Frecuencia	Importancia
Capacitarse en nuevas herramientas tecnológicas	Baja	Media
Resolver errores de inventario (<i>sobrestock</i> , productos vencidos)	Alta	Alta
Registrar productos por voz y confirmar datos	Media	Alta
Escanear lotes con cámara rápida y validar	Alta	Alta
Ubicar productos en almacén con QR/mapa	Alta	Alta
Revisar alertas de predicción de stock	Media	Alta
Visualizar sedes y rutas en mapa (GPS)	Baja	Media

Segmento 2: Startups y emprendedores en expansión con necesidades logísticas

Task Matrix	Frecuencia	Importancia
Supervisar la producción (propia o con talleres externos)	Alta	Alta
Gestionar inventario de insumos y productos terminados (telas, hilos, prendas)	Alta	Alta
Registrar entradas, salidas y devoluciones de productos	Media	Alta
Coordinar pedidos con clientes o puntos de venta	Alta	Alta
Diseñar nuevas piezas o adaptar modelos a la demanda	Alta	Alta
Publicar contenido y gestionar redes sociales	Alta	Media
Empaquetar productos y organizar despachos	Media	Alta
Analizar qué productos se venden más y qué insumos se usan más	Media	Alta
Actualizar listas de precios, catálogos o colecciones	Media	Media
Registrar ventas y organizar información contable o administrativa	Media	Alta
Aprender herramientas nuevas para mejorar su operativa (apps, control, automatización)	Media	Alta
Dictar insumos o productos por voz	Alta	Alta
Registrar clientes y entregas con GPS	Alta	Alta
Analizar predicciones de producción/ventas	Media	Alta
Escanear productos en lotes y asignar ubicación	Alta	Alta
Recibir alertas de sobrestock o quiebre	Media	Alta

2.3.3. User Journey Mapping

Segmento 1: Bodegas especializadas por rubro

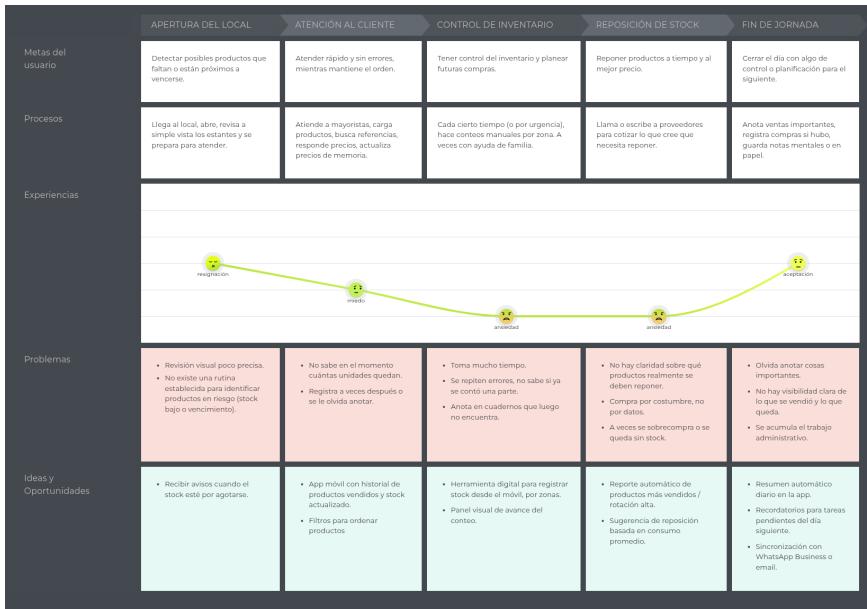


Imagen (N°4). Elaboración propia. Realizado en UXPressia

Segmento 2: Startups y emprendedores en expansión con necesidades logísticas

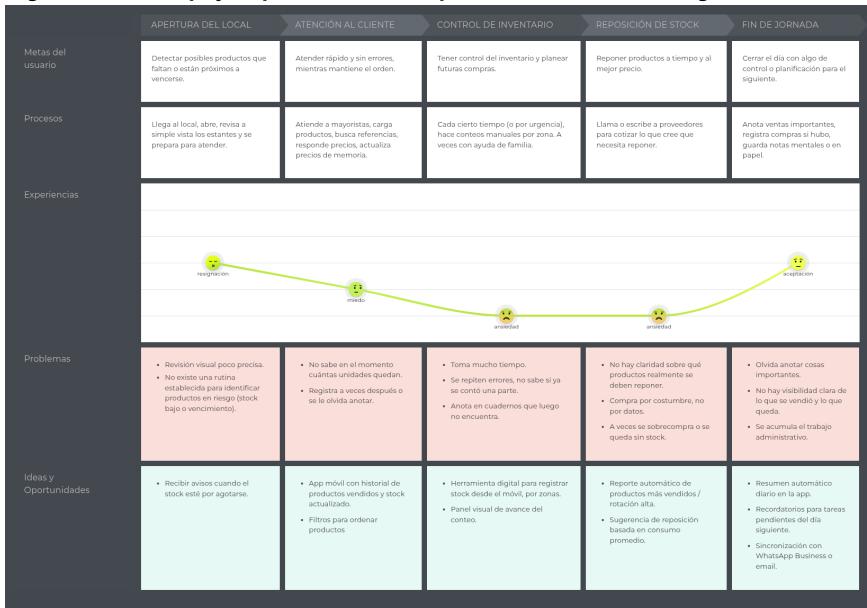


Imagen (N°5). Elaboración propia. Realizado en UXPressia

2.3.4. Empathy Mapping

Segmento 1: Bodegas especializadas por rubro

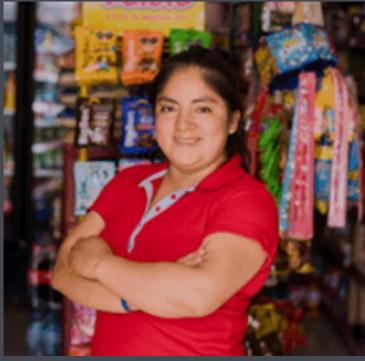
<p>1.¿Con quién estamos empatizando?</p> <p>Carla es una mujer peruana que es gerente de una tienda mayorista la cual desea organizarse mejor para hacer crecer su negocio y llevar un mejor balance en todos los aspectos de su vida.</p>	<p>7.¿Qué es lo que piensa y siente?</p> <p>“ Estoy cansada de hacer todo a mano, pero tampoco quiero perder el control. Mi trabajo no termina cuando cierro la tienda. Me llevo el estrés a casa.”</p>	<p>2.¿Qué es lo que necesita hacer?</p> <p>Estar dispuesta a aceptar cambios para el progreso de su bodega. Informarse de nuevas tecnologías para no quedarse atrás.</p>
<p>6.¿Qué es lo que escucha?</p> <p>Recomendaciones de amigos comerciantes que le hablan de apps para controlar ventas o inventario. Opiniones de su familia sobre lo mucho que trabaja y lo poco que descansa.</p>		
<p>5.¿Qué es lo que hace?</p> <p>Reconoce los puntos débiles de su gestión. Busca nuevas tecnologías y las compara para elegir la que más se adapte a sus necesidad.</p>		<p>3.¿Qué es lo que ve?</p> <p>Un mercado competitivo, con negocios similares que ofrecen precios agresivos. Que otros negocios usan herramientas tecnológicas, pero le parecen muy complicadas.</p>
<p>Dolores</p> <p>¿Cuáles son sus frustraciones y miedos? Tareas repetitivas que consumen tiempo valioso. No tener información clara para tomar decisiones. Dependencia de su presencia física para que todo funcione.</p>	<p>Motivaciones</p> <p>¿Cuáles son sus necesidades, esperanzas y deseos? Automatizar procesos sin perder el control. Tomar decisiones informadas (con reportes visuales, simples). Sentirse profesionalizada y al nivel de empresas grandes.</p>	<p>4.¿Qué es lo que dice?</p> <p>“ Tengo que seguir esforzándome por darle un servicio de calidad a mis clientes sin sacrificar mi tiempo libre.”</p>

Imagen (Nº6). Elaboración propia. Realizado en UXPressia

Segmento 2: Startups y emprendedores en expansión con necesidades logísticas

<p>1.¿Con quién estamos empatizando?</p> <p>Sebastián es el fundador de su propio emprendimiento digital el cual es una tienda de ropa en redes. Quiero expandirme para escalar más rápido y posicionar su negocio entre los mejores.</p>	<p>7.¿Qué es lo que piensa y siente?</p> <p>“Siento que pierdo dinero en materiales y tiempo. Estoy cansado de repetir los mismos errores por no tener control.” Se</p>	<p>2.¿Qué es lo que necesita hacer?</p> <p>Reconocer los puntos débiles en la logística de su emprendimiento para mejorarlo. Informarse acerca de nuevas tecnologías.</p>
<p>6.¿Qué es lo que escucha?</p> <p>A talleristas diciendo: “Ya entregué todo, revisa tu lista”. A su entorno recomendándole que digitalice su operación.</p>		<p>3.¿Qué es lo que ve?</p> <p>Sus estantes llenos, pero sin saber qué está realmente disponible. A sus talleres o trabajadores haciendo su parte sin mucha supervisión.</p>
<p>5.¿Qué es lo que hace?</p> <p>Elige las tecnologías que se adaptan a sus necesidades y trabaja para suplir con las fallas en su gestión que reconoció antes.</p>		
<p>Dolores</p> <p>¿Cuáles son sus frustraciones y miedos? Pérdidas de materiales sin explicación clara. Errores en entregas por falta de trazabilidad. Falta de información para planificar o comprar inteligentemente.</p>	<p>Motivaciones</p> <p>¿Cuáles son sus necesidades, esperanzas y deseos? Tener control real del negocio, sin depender 100% de su presencia. Reducir errores y estrés operativo. Escalar su empresa con procesos profesionales, sin perder tiempo ni creatividad.</p>	<p>4.¿Qué es lo que dice?</p> <p>“No puedo seguir así, tengo que organizarme por mí y mi negocio.”</p>

Imagen (Nº7). Elaboración propia. Realizado en UXPressia

2.3.5. Ubiquitous Language

El siguiente glosario presenta los términos clave utilizados a lo largo del desarrollo del proyecto StockWise. Este lenguaje común busca asegurar que todos los miembros del equipo (tanto técnicos como no técnicos) comparten una comprensión unificada de los conceptos centrales del sistema, facilitando así la comunicación y el diseño colaborativo.

Palabra	Descripción
Landing Page	Página principal de presentación al llegar al sitio oficial de StockWise.
Inventario	Conjunto de productos almacenados en una bodega con detalles como cantidad, ubicación y fecha de vencimiento.
Producto	Artículo registrado en la plataforma con nombre, descripción, stock mínimo, unidades, fecha de vencimiento y categoría.
Stock	Cantidad disponible de un producto específico en el inventario.
Stock bajo	Estado que indica que la cantidad de un producto está por debajo del mínimo definido por el usuario.
Panel de Control	Interfaz principal para visualizar métricas, movimientos y reportes del inventario.

Palabra	Descripción
Movimiento de Inventario	Registro de cambios en cantidades de productos por compras, ventas, pérdidas o ajustes.
Reporte	Documento visual o estadístico que resume información clave del inventario para la toma de decisiones.
Bodega	Negocio pequeño con venta de productos de primera necesidad, como alimentos y bebidas.
Usuario Administrador	Persona con acceso total que puede gestionar usuarios, configuraciones y datos.
Emprendedor	Usuario en etapa de crecimiento que busca profesionalizar la gestión de su negocio.
Dueña de bodega	Usuario con experiencia comercial que desea optimizar la gestión operativa con herramientas digitales.
Versión Freemium	Modelo que permite acceso gratuito a funciones básicas y pago por funciones premium.
Onboarding	Proceso de bienvenida y guía para enseñar a nuevos usuarios a usar la plataforma.
MVP (Producto Mínimo Viable)	Versión inicial con funciones mínimas para validar hipótesis clave con usuarios reales.

2.4. Requirements specification

El siguiente glosario presenta los términos clave utilizados a lo largo del desarrollo del proyecto StockWise. Este lenguaje común busca asegurar que todos los miembros del equipo (tanto técnicos como no técnicos) comparten una comprensión unificada de los conceptos centrales del sistema, facilitando así la comunicación y el diseño colaborativo.

1. **Landing Page:** Página de presentación a la que llegan los usuarios cuando visitan el sitio oficial de stockwise.
2. **Inventory:** Conjunto de productos almacenados en una bodega. Incluye información como cantidad disponible, ubicación, fecha de vencimiento, entre otros.
3. **Producto:** Bien que se registra en la plataforma, con atributos específicos como nombre, descripción, stock mínimo, unidades, fecha de vencimiento, y categoría.
4. **Stock:** Cantidad disponible de un producto exacto en el inventario
5. **Stock bajo:** Estado de un producto que ha superado la mínima cantidad definida por el usuario.
6. **Panel de Control:** Interfaz principal de la plataforma donde el usuario puede visualizar métricas clave del inventario, movimientos y reportes
7. **Movimiento de Inventario:** Registro de cualquier cambio en la cantidad de productos: entrada (nueva compra), salida (venta o pérdida), ajuste manual, etc.
8. **Reporte:** Documento visual o estadístico que resume información clave del inventario
9. **Bodega:** Negocios pequeños donde se venden productos de primera necesidad, como alimentos y bebidas.
10. **Usuario Administrador:** Persona que tiene acceso total a la plataforma, puede crear usuarios, modificar configuraciones y visualizar todos los datos.
11. **Emprendedor:** Usuario que está en proceso de expansión de su negocio, y busca profesionalizar su gestión
12. **Dueña de bodega:** Usuario generalmente con experiencia en el rubro comercial que busca optimizar su gestión operativa con herramientas digitales
13. **Versión Freemium:** Modelo de negocio que permite a los usuarios acceder a funciones básicas de forma gratuita, con la posibilidad de pagar por funciones premium.
14. **Onboarding:** Proceso de bienvenida y guía para nuevos usuarios que les enseña cómo usar la plataforma y aprovechar sus funcionalidades.
15. **MVP (Producto Mínimo Viable):** Versión inicial de la plataforma con las funciones mínimas necesarias para validar hipótesis clave con usuarios reales.
16. **Registro por Voz:** Funcionalidad que permite ingresar datos dictando productos, cantidades o insumos, validados por el sistema.
17. **Escaneo QR:** Proceso de registrar productos o lotes mediante el uso de lector, facilitando la velocidad de captura.
18. **Mapa/GPS:** Herramienta que permite ubicar bodegas, almacenes o rutas de distribución, y registrar entregas.
19. **Alertas de Stock:** Notificaciones automáticas que advierten sobre riesgos de quiebre, sobrestock o vencimiento de productos.
20. **Rotación de Productos:** Indicador que mide la velocidad de venta o uso de un producto en el inventario.
21. **Gestión de Producción:** Capacidad de registrar y controlar procesos productivos.
22. **Contabilidad Básica:** Registro y organización de ventas, costos y márgenes, integrados a reportes financieros simples.

2.4.1. User Stories

Las User Stories son clave en metodologías ágiles porque traducen los requisitos funcionales desde la mirada del usuario. Cada historia especifica una necesidad concreta, lo que permite planificar, priorizar y construir el sistema de forma iterativa. Así se asegura que cada función aporte valor real y permanezca alineada con las expectativas del usuario final.

Story ID	User	Priority	Epic
US01	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario

Title Registra producto nuevo Description Como usuario, quiero registrar un nuevo producto en mi inventario, para tener un control detallado de su existencia desde el comienzo. Acceptance Criteria

Escenario 01: Registro exitoso de producto

- Dado que el sistema recibe datos válidos para un producto,
- Cuando procesa el registro,
- Entonces guarda el producto en la base de datos
- Y lo incluye en el inventario.

Escenario 02: Interno de registro en campos vacíos

- Dado que el sistema recibe un producto con campos requeridos vacíos,
- Cuando intenta registrar el producto,
- Entonces el sistema rechaza la operación
- Y muestra mensajes de validación sobre los campos faltantes.

Story ID	User	Priority	Epic
US02	Usuario de inventario	Alta	EP01 - Funciones básicas de inventario

Title Editar información de producto Description Como usuario, quiero editar la información de un producto registrado, para corregir o actualizar datos. Acceptance Criteria

Escenario 01: Edición con éxito

- **Dado que** el usuario ya tiene un producto en el inventario
- **Cuando** modifica campos de información del producto
- **Y** procesa el registro adecuadamente
- **Entonces** el sistema guarda los cambios

Escenario 02: Edición con datos inválidos

- **Dado que** el usuario está editando información de un producto
- **Cuando** ingrese datos inválidos
- **Entonces** el sistema le muestra los errores
- **Y** no se guardan los cambios

Story ID	User	Priority	Epic
US03	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario

Title Registrar salida de producto Description Como usuario, quiero registrar la salida de productos del inventario, para mantener actualizado el stock en tiempo real. Acceptance Criteria

Escenario 01: Registro válido de salida de stock

- **Dado que** un producto tiene suficiente stock disponible
- **Cuando** el sistema recibe una salida con cantidad permitida,
- **Entonces** descuenta la cantidad del stock
- **Y** registra el movimiento.

Escenario 02: Intento de salida mayor al stock disponible

- **Dado que** que el stock actual es menor que la cantidad solicitada
- **Cuando** valida la salida
- **Entonces** rechaza la operación
- **Y** expone el motivo "cantidad supera stock disponible"

Story ID	User	Priority	Epic
US04	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario
Title	Ver historial de movimientos		
	Description		
	Como usuario, quiero ver el historial de entrada y salida de mis productos, para rastrear los cambios y movimientos en el inventario.		
	Acceptance Criteria		

Escenario 01: Visualización del historial completo

- Dado que el usuario está en la sección historial de un producto
- Cuando este producto tiene movimientos registrados
- Entonces el sistema muestra una lista con toda la información del producto

Escenario 02: Producto sin historial de movimientos

- Dado que el usuario accede al historial de un producto nuevo o sin registro
- Cuando no hay datos previos
- Entonces el sistema muestra un mensaje indicando que no hay movimientos registrados

Story ID	User	Priority	Epic			
US05	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario			
Title	Generar alertas por bajo stock					
Description						
Como usuario, quiero recibir alertas cuando un producto está debajo del stock mínimo, para reabastecerme a tiempo.						
Acceptance Criteria						
Escenario 01: Activación automática de alerta por stock bajo						
<ul style="list-style-type: none">• Dado que un producto tiene un límite mínimo configurado• Cuando su stock baja por debajo del mínimo• Entonces el sistema genera una alerta visual y notifica al usuario						
Escenario 02: Personalización de la alerta						
<ul style="list-style-type: none">• Dado que el usuario desea configurar alertas específicas• Cuando accede a la edición del producto y define un nuevo valor de stock mínimo• Entonces el sistema guarda la alerta y la activa con el nuevo valor						

Story ID	User	Priority	Epic			
US06	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario			
Title	Buscar productos en inventarios					
Description						
Como usuario, quiero buscar productos en el inventario por nombre o categoría, para encontrarlo de manera rápida y fácil.						
Acceptance Criteria						
Escenario 01: Búsqueda por nombre o código						
<ul style="list-style-type: none">• Dado que el usuario está en la sección "Inventario"• Cuando ingresa el nombre o código de un producto en el buscador• Entonces se muestran los productos que coinciden con los términos de búsqueda						
Escenario 02: Búsqueda sin coincidencias						
<ul style="list-style-type: none">• Dado que el usuario realiza una búsqueda con término no registrado• Cuando busca el producto• Entonces el sistema muestra el mensaje "No se encontraron resultados"						

Story ID	User	Priority	Epic			
US07	Usuario de inventario		EP01 – Funciones básicas de inventario			
Title	Añadir etiquetas para productos					
Description						
Como usuario, quiero poder clasificar los productos mediante etiquetas, para organizar el stock y agilizar búsquedas.						

Acceptance Criteria			
Escenario 01: Añadir etiqueta correctamente			
<ul style="list-style-type: none"> • Dado que el usuario está en el formulario de "Nuevo producto" • Cuando escribe una o varias etiquetas personalizadas y lo guarda exitosamente • Entonces las etiquetas se registran con el producto y son visibles en su detalle 			
Escenario 02: Usar etiquetas existentes			
<ul style="list-style-type: none"> • Dado que el usuario comienza a escribir una etiqueta • Cuando ya existe una similar en el sistema • Entonces el sistema sugiere etiquetas existentes para evitar duplicados 			
Escenario 03: Filtrar por etiquetas			
<ul style="list-style-type: none"> • Dado que el usuario está en el módulo de inventario • Cuando selecciona una etiqueta desde el filtro • Entonces se muestran solo los productos que contienen esa etiqueta 			

Story ID	User	Priority	Epic			
US08	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado			
Title	Estadística de productos más vendidos					
Description						
Como usuario, quiero ver un listado con los productos más vendidos durante un periodo de tiempo, para identificar los de mayor demanda y planificar mejor.						
Acceptance Criteria						
Escenario 01: Ver productos más vendidos en un periodo						
<ul style="list-style-type: none"> • Dado que el usuario accede al módulo de estadísticas • Cuando selecciona un rango de fechas • Entonces el sistema muestra un ranking de productos más vendidos y permite ordenarlos por unidades o ingresos 						
Escenario 02: Exportar reporte						
<ul style="list-style-type: none"> • Dado que el usuario visualiza el listado • Cuando solicita la exportación • Entonces el sistema genera un archivo PDF con el ranking 						

Story ID	User	Priority	Epic			
US09	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado			
Title	Estadística de categoría más vendida					
Description						
Como usuario, quiero visualizar qué categorías o líneas de productos son las que más se venden, para saber cuál promocionar o reducir del catálogo.						
Acceptance Criteria						
Escenario 01: Ver categorías más vendidas						
<ul style="list-style-type: none"> • Dado que el usuario accede al módulo de estadísticas • Cuando selecciona un rango de fechas • Entonces se muestra un gráfico con la participación de cada categoría en las ventas 						
Escenario 02: Ver detalle de productos por categoría						
<ul style="list-style-type: none"> • Dado que el usuario visualiza las categorías más vendidas • Cuando busca en categoría • Entonces se muestra la lista con los productos que la componen y sus ventas 						

Story ID	User	Priority	Epic		
US10	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado		
Title		Ticket promedio			
Description					
Como usuario, quiero conocer el ticket promedio de mis ventas para entender cuánto gasta en promedio cada cliente y evaluar la efectividad de los precios.					
Acceptance Criteria					
Escenario 01: Visualizar ticket promedio					
<ul style="list-style-type: none"> Dado que el usuario accede al módulo de estadísticas Cuando selecciona un rango de fechas Entonces se muestra el ticket promedio, el total de ingresos y el número de ventas 					
Escenario 02: Comparar ticket con periodo anterior					
<ul style="list-style-type: none"> Dado que el usuario ve el ticket promedio actual Cuando habilita la opción de comparación Entonces el sistema muestra la variación respecto al periodo anterior 					

Story ID	User	Priority	Epic		
US11	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado		
Title		Estadística de stock promedio			
Description					
Como usuario, quiero visualizar el promedio de stock disponible para identificar niveles óptimos y evitar quiebres o exceso de stock.					
Acceptance Criteria					
Escenario 01: Visualizar stock promedio					
<ul style="list-style-type: none"> Dado que el usuario accede al módulo de estadísticas Cuando selecciona un producto o categoría y un rango de fechas Entonces se muestra el promedio de stock disponible en ese periodo 					
Escenario 02: Detectar stock por fuera del rango óptimo					
<ul style="list-style-type: none"> Dado que el usuario ve el reporte de stock promedio Cuando el valor está por debajo o por encima del stock ideal Entonces el sistema resalta los productos con riesgo de quiebre o sobrestock y puede sugerir acciones 					

Story ID	User	Priority	Epic		
US12	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado		
Title		Generar reportes de inventario			
Description					
Como usuario, quiero generar reportes detallados del inventario para analizar el estado general y tomar decisiones informadas.					
Acceptance Criteria					
Escenario 01: Generar reporte completo					
<ul style="list-style-type: none"> Dado que el usuario accede al módulo de reportes Cuando selecciona parámetros como fecha, información, categorías o stock mínimo Entonces el sistema genera un reporte con esa información 					
Escenario 02: Exportar reporte					

- Dado que el usuario visualiza el reporte generado
- Cuando hace clic en "Exportar"
- Entonces el sistema descarga el reporte en PDF o Excel

Story ID	User	Priority	Epic		
US13	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado		
Title		Controlar productos caducados			
Description					
Como usuario, quiero identificar productos próximos a caducar o ya caducados, para evitar pérdidas y mantener la calidad del inventario.					
Acceptance Criteria					
Escenario 01: Visualizar productos próximos a caducar					
<ul style="list-style-type: none"> • Dado que el usuario accede al módulo de inventario • Cuando filtra por fecha de caducidad próxima • Entonces el sistema muestra una lista con los productos que caducan pronto 					
Escenario 02: Alerta de productos caducados					
<ul style="list-style-type: none"> • Dado que un producto está caducado • Cuando el usuario accede al inventario • Entonces el sistema muestra una alerta visual destacando esos productos 					

Story ID	User	Priority	Epic		
US14	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado		
Title		Registrar devolución de productos			
Description					
Como usuario, quiero registrar devoluciones de productos al inventario, para mantener el stock actualizado y controlar incidencias.					
Acceptance Criteria					
Escenario 01: Registro de devolución válida					
<ul style="list-style-type: none"> • Dado que un producto fue devuelto por un cliente • Cuando el usuario ingresa la cantidad devuelta y la razón y confirma • Entonces el sistema incrementa el stock y guarda el movimiento 					
Escenario 02: Devolución con cantidad inválida					
<ul style="list-style-type: none"> • Dado que el usuario intenta registrar una devolución con cantidad negativa o mayor a la vendida • Cuando hace clic en "Guardar" • Entonces el sistema muestra un error y no registra la devolución 					

Story ID	User	Priority	Epic		
US15	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado		
Title		Configurar roles y permisos de usuario			
Description					
Como usuario, quiero definir roles y permisos para los usuarios del sistema.					
Acceptance Criteria					
Escenario 01: Crear nuevo rol con permisos específicos					
<ul style="list-style-type: none"> • Dado que el administrador accede a la configuración de usuarios • Cuando crea un nuevo rol y asigna permisos • Entonces el rol se guarda y puede ser asignado 					

Escenario 02: Asignar rol a usuario

- Dado que el administrador selecciona un usuario existente
- Cuando asigna un rol específico
- Entonces el sistema muestra los permisos del rol asignado

Story ID	User	Priority	Epic			
US16	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario			
Title	Gestionar proveedores					
Description						
Como usuario, quiero registrar y gestionar la información de proveedores, para mantener contactos actualizados y facilitar la reposición.						
Acceptance Criteria						
Escenario 01: Registro exitoso de proveedor						
<ul style="list-style-type: none">• Dado que el usuario está en "Nuevo proveedor"• Cuando completa los campos obligatorios y guarda• Entonces el proveedor se registra y aparece en la lista						
Escenario 02: Edición de datos de proveedor						
<ul style="list-style-type: none">• Dado que el usuario selecciona un proveedor existente• Cuando modifica la información y guarda• Entonces el sistema actualiza los datos del proveedor						

Story ID	User	Priority	Epic			
US17	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario			
Title	Registrar lote de productos					
Description						
Como usuario, quiero registrar productos por lote, para controlar el ingreso masivo con un mismo proveedor, fecha y condiciones.						
Acceptance Criteria						
Escenario 01: Registro de lote con múltiples productos						
<ul style="list-style-type: none">• Dado que el usuario accede a "Nuevo lote"• Cuando añade varios productos con fecha, proveedor y condiciones• Entonces el sistema guarda el lote y vincula los productos						
Escenario 02: Visualizar lote registrado						
<ul style="list-style-type: none">• Dado que el usuario visualiza la lista de productos• Cuando selecciona "Ver lote"• Entonces se muestra la información común del lote						

Story ID	User	Priority	Epic			
US18	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado			
Title	Visualizar historial de stock					
Description						
Como usuario, quiero ver el historial de stock de un producto, para analizar su comportamiento y tomar mejores decisiones de compra.						
Acceptance Criteria						
Escenario 01: Visualizar historial por producto						
<ul style="list-style-type: none">• Dado que el usuario accede a la ficha de un producto						

- Cuando selecciona "Ver historial"
- Entonces se muestra una gráfica con variaciones de stock en el tiempo

Escenario 02: Filtro por periodo

- Dado que el usuario visualiza el historial de stock
- Cuando aplica un filtro por rango de fechas
- Entonces la gráfica se actualiza con el periodo seleccionado

Story ID	User	Priority	Epic			
US19	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado			
Title	Configurar alertas de stock mínimo					
Description						
Como usuario, quiero configurar alertas cuando un producto alcance su stock mínimo, para poder reabastecerme a tiempo.						
Acceptance Criteria						
Escenario 01: Configurar stock mínimo						
<ul style="list-style-type: none"> • Dado que el usuario accede a la configuración de productos • Cuando establece un valor de stock mínimo por producto • Entonces el sistema guarda ese valor para alertas 						
Escenario 02: Alerta de bajo stock						
<ul style="list-style-type: none"> • Dado que un producto llega o baja del stock mínimo configurado • Cuando el usuario accede al panel o inventario • Entonces se muestra una alerta visual indicando que se debe reponer 						

Story ID	User	Priority	Epic			
US20	Usuario de inventario	Baja	EP03 – Innovación y valor agregado			
Title	Añadir etiquetas a productos					
Description						
Como usuario, quiero añadir etiquetas personalizadas a los productos, para facilitar su clasificación y búsqueda.						
Acceptance Criteria						
Escenario 01: Etiquetado desde el formulario del producto						
<ul style="list-style-type: none"> • Dado que el usuario está registrando o editando un producto • Cuando escribe nuevas etiquetas o selecciona existentes • Entonces las etiquetas se asocian al producto guardado 						
Escenario 02: Buscar productos por etiqueta						
<ul style="list-style-type: none"> • Dado que el usuario está en el módulo de búsqueda o lista • Cuando ingresa el nombre de una etiqueta • Entonces se filtran los productos asociados a esa etiqueta 						

Story ID	User	Priority	Epic			
US21	Usuario de inventario	Alta	EP01 - Funciones básicas de inventario			
Title	Buscar productos en inventario					
Description						
Como usuario, quiero buscar productos en el inventario por nombre o categoría, para encontrarlos de forma rápida y eficiente.						
Acceptance Criteria						

Escenario 01: Búsqueda por nombre

- Dado que el usuario accede al buscador
- Cuando ingresa parte del nombre del producto
- Entonces se muestran resultados coincidentes en tiempo real

Escenario 02: Búsqueda por categoría

- Dado que el usuario está en el buscador
- Cuando selecciona una categoría del filtro
- Entonces se muestran solo productos de esa categoría

Story ID	User	Priority	Epic			
US22	Usuario de inventario	Baja	EP03 – Innovación y valor agregado			
Title	Generar alertas por productos caducados					
Description						
Como usuario, quiero recibir alertas automáticas de productos caducados, para poder retirarlos del inventario a tiempo.						
Acceptance Criteria						
Escenario 01: Activar alertas automáticas						
<ul style="list-style-type: none"> • Dado que el usuario accede a configuración de alertas • Cuando activa la opción de alerta por caducidad • Entonces el sistema enviará notificaciones cuando haya productos vencidos 						
Escenario 02: Visualización de productos vencidos						
<ul style="list-style-type: none"> • Dado que hay productos caducados en el inventario • Cuando el usuario accede al sistema • Entonces se muestra un panel o lista resaltando estos productos 						

Story ID	User	Priority	Epic			
US23	Usuario de inventario	Alta	EP01 – Funciones básicas de inventario			
Title	Registrar ingreso de productos					
Description						
Como usuario, quiero registrar el ingreso de productos al inventario, para mantener actualizado el stock disponible.						
Acceptance Criteria						
Escenario 01: Ingreso individual de producto						
<ul style="list-style-type: none"> • Dado que el usuario accede al formulario de ingreso • Cuando registra los datos del producto y la cantidad recibida • Entonces el sistema actualiza el stock del producto 						
Escenario 02: Ingreso masivo por archivo						
<ul style="list-style-type: none"> • Dado que el usuario cuenta con un archivo de productos • Cuando carga el archivo en el sistema • Entonces se actualiza el inventario con los productos del archivo 						

Story ID	User	Priority	Epic			
US24	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado			
Title	Visualizar resumen de stock por categoría					
Description						
Como usuario, quiero ver un resumen del stock agrupado por categoría, para conocer el estado general del inventario de forma ordenada.						
Acceptance Criteria						

Acceptance Criteria

Escenario 01: Visualización del resumen por categoría

- Dado que el usuario accede al módulo de inventario
- Cuando selecciona "Resumen por categoría"
- Entonces se muestra una tabla con cada categoría, cantidad total de productos y stock promedio

Escenario 02: Acceso al detalle desde el resumen

- Dado que el usuario visualiza el resumen por categoría
- Cuando hace clic en una categoría
- Entonces se despliega la lista de productos que la conforman

Story ID	User	Priority	Epic			
US25	Usuario de inventario	Media	EP02 – Funciones de analítica, reportes y control avanzado			
Title	Comparar ventas entre periodos					
Description						
Como usuario, quiero comparar las ventas entre dos períodos, para identificar tendencias y evaluar el impacto de promociones.						
Acceptance Criteria						
<p>Escenario 01: Comparación de períodos seleccionados</p> <ul style="list-style-type: none"> • Dado que el usuario accede al módulo de estadísticas • Cuando selecciona dos períodos a comparar • Entonces se muestra una gráfica con la diferencia de ventas <p>Escenario 02: Comparación por categoría o producto</p> <ul style="list-style-type: none"> • Dado que el usuario visualiza la comparación general • Cuando filtra por categoría o producto • Entonces la gráfica se actualiza con los datos seleccionados 						

Story ID	User	Priority	Epic			
US26	Usuario de inventario	Baja	EP03 – Innovación y valor agregado			
Title	Registrar productos por voz					
Description						
Como usuario, quiero registrar productos mediante comandos de voz (ej. "Aregar 20 botellas de agua al inventario"), para ganar rapidez y comodidad cuando no pueda usar el teclado.						
Acceptance Criteria						

Escenario 01: Comando válido

- Dado que el usuario activa el modo voz,
- Cuando dicta "Aregar <cantidad> <producto> al inventario",
- Entonces el sistema reconoce cantidad y producto, muestra un resumen y al confirmar registra el movimiento.

Escenario 02: Ambigüedad o error

- Dado que el audio no es claro o hay varias coincidencias,
- Cuando el sistema no puede identificar el producto,
- Entonces solicita confirmación/sugerencias y permite corregir o escribir manualmente.

Escenario 03: Dictado continuo

- Dado que el usuario permanece en modo voz,
- Cuando dicta varios comandos consecutivos,
- Entonces el sistema procesa cada ítem y permite "cancelar" o "terminar" por voz.

Story ID	User	Priority	Epic			
US27	Usuario de inventario	Baja	EP03 - Innovación y valor agregado			
Title	Geolocalización de productos y entregas					
Description						
Como usuario, quiero integrar GPS para registrar procedencia, puntos de entrega y visualizar en un mapa las sedes vinculadas a la tienda principal, para optimizar trazabilidad y distribución.						
Acceptance Criteria						
<p>Escenario 01: Captura de ubicación</p> <ul style="list-style-type: none"> • Dado que el usuario registra un movimiento/entrega, • Cuando concede permisos de ubicación, • Entonces el sistema guarda latitud/longitud y muestra un pin en el mapa del producto o entrega. <p>Escenario 02: Mapa de sedes</p> <ul style="list-style-type: none"> • Dado que el usuario abre el mapa, • Cuando selecciona "Sedes y rutas", • Entonces visualiza marcadores por sede y puede filtrar por estado. <p>Escenario 03: Sin permisos</p> <ul style="list-style-type: none"> • Dado que el usuario deniega GPS, • Cuando intenta registrar ubicación, • Entonces el sistema muestra aviso y permite ingresar dirección manual. 						

Story ID	User	Priority	Epic			
US28	Usuario de inventario	Baja	EP03 - Innovación y valor agregado			
Title	Localizar productos con mapa y QR					
Description						
Como usuario, quiero ubicar productos dentro del almacén usando un mapa interactivo e integración de códigos QR, para encontrarlos con precisión.						
Acceptance Criteria						
<p>Escenario 01: Asociar ubicación por QR</p> <ul style="list-style-type: none"> • Dado que el usuario escanea un QR de estantería/pasillo, • Cuando guarda la ubicación en la ficha del producto, • Entonces el sistema vincula el producto a ese punto del mapa. <p>Escenario 02: Búsqueda y ruta</p> <ul style="list-style-type: none"> • Dado que el usuario busca un producto, • Cuando selecciona "Ver ubicación", • Entonces el mapa señala pasillo/estante y puede mostrar ruta sugerida. <p>Escenario 03: QR inválido</p> <ul style="list-style-type: none"> • Dado que el QR no existe o está dañado, • Cuando intenta asociarlo, • Entonces el sistema alerta y permite ingresar ubicación manual. 						

Story ID	User	Priority	Epic			
US29	Usuario de inventario	Baja	EP03 – Innovación y valor agregado			
Title	Predicción de reabastecimiento					
Description						
Como usuario, quiero recibir sugerencias de cuándo y cuánto reponer según patrones de ventas y stock mínimo, para evitar quiebres y						

sobrestock.

Acceptance Criteria			
Escenario 01: Sugerencia de compra			
<ul style="list-style-type: none">• Dado que el sistema analiza histórico y lead time,• Cuando el stock proyectado cae bajo el nivel objetivo,• Entonces propone fecha y cantidad de reabastecimiento y permite aceptarla/editarla.			
Escenario 02: Alerta de quiebre inminente			
<ul style="list-style-type: none">• Dado que el consumo supera la previsión,• Cuando se detecta riesgo de quiebre,• Entonces se muestra alerta prioritaria y recomendaciones de acción.			
Escenario 03: Parámetros ajustables			
<ul style="list-style-type: none">• Dado que el usuario abre "Predicción",• Cuando ajusta stock mínimo, tiempo de reposición y estacionalidad,• Entonces el modelo recalculta y actualiza las sugerencias.			

Story ID	User	Priority	Epic			
US30	Usuario de inventario	Baja	EP03 – Innovación y valor agregado			
Title	Escaneo por lotes con cámara rápida					
Description						
Como usuario, quiero registrar lotes tomando una foto del producto/lote para que una API de visión devuelva etiquetas genéricas; luego confirmar producto exacto y cantidad y ver su ubicación en el almacén virtual.						
Acceptance Criteria						
Escenario 01: Registro por visión						
<ul style="list-style-type: none">• Dado que el usuario abre "Escaneo rápido",• Cuando toma una foto del lote,• Entonces la API (p.ej., ML Kit) devuelve etiquetas; el usuario elige el producto, ingresa cantidad y confirma el registro del lote.						
Escenario 02: Etiquetas ambiguas						
<ul style="list-style-type: none">• Dado que la API devuelve varias opciones o no encuentra coincidencias,• Cuando el usuario revisa sugerencias,• Entonces puede seleccionar/corregir manualmente y continuar sin perder el flujo.						
Escenario 03: Ubicación en almacén virtual						
<ul style="list-style-type: none">• Dado que el lote quedó registrado,• Cuando el usuario finaliza,• Entonces el sistema muestra o permite elegir la ubicación del lote en el mapa del almacén.						

Technical Stories

En esta sección se describen las historias técnicas que desarrollamos para implementar las funcionalidades clave. Cada historia define tareas específicas que el equipo de desarrollo debe realizar, como crear endpoints, manejar validaciones, controlar el stock, generar reportes, entre otros.

Story ID	User	Priority	Epic			
TS01	Developer	Alta	EP04 - Cuentas y seguridad			
Title	API Registro de usuarios					
Description						
Como desarrollador, necesito exponer endpoints para registrar nuevos usuarios, asegurando validaciones y persistencia correcta en el backend.						
Acceptance Criteria						
Escenario 01: Registro exitoso						

Dado que el endpoint de registro está disponible y la app valida email/contraseña,
 Cuando el usuario envía datos válidos desde el móvil,
 Entonces el backend crea la cuenta (201), retorna token y la app lo guarda de forma segura (Keychain/Keystore).

Escenario 02: Datos inválidos o duplicados

Dado el endpoint de registro,
 Cuando el email ya existe o los campos no cumplen las reglas (formato/fortaleza),
 Entonces se responde 400/409 con mensaje claro y la app muestra los errores sin duplicar el envío.

Story ID	User	Priority	Epic			
TS02	Developer	Alta	EP04 - Cuentas y seguridad			
Title	API Autenticación de usuarios					
Description						
Como desarrollador, necesito implementar endpoints de autenticación para que la app móvil inicie sesión de forma segura (emisión y validación de tokens).						
Acceptance Criteria						
Escenario 01: Login válido Dado el endpoint de autenticación, Cuando se envían credenciales válidas desde la app, Entonces se retorna 200 con access/refresh token y datos mínimos del perfil; la app inicia sesión persistente y puede habilitar biometría.						
Escenario 02: Login inválido Dado el endpoint de autenticación, Cuando las credenciales son incorrectas o el usuario está inactivo, Entonces se retorna 401/403 con mensaje adecuado y la app ofrece "recuperar contraseña".						

Story ID	User	Priority	Epic			
TS03	Developer	Alta	EP01 - Funciones básicas de inventario			
Title	API Creación de productos					
Description						
Como desarrollador, necesito exponer endpoints para registrar productos con validaciones de campos requeridos y unicidad.						
Acceptance Criteria						
Escenario 01: Producto creado Dado que la app permite capturar foto/código de barras, Cuando se envían datos correctos desde el móvil, Entonces el backend crea el producto (201) y la app sincroniza la imagen en segundo plano.						
Escenario 02: Datos inválidos Dado el endpoint de productos, Cuando el código está duplicado o faltan campos requeridos, Entonces se retorna 400/409 con detalle y la app resalta los campos a corregir.						

Story ID	User	Priority	Epic			
TS04	Developer	Alta	EP01 - Funciones básicas de inventario			
Title	API Consulta de productos					
Description						
Como desarrollador, necesito listar productos con soporte de paginación y orden para su visualización en la app móvil.						
Acceptance Criteria						
Escenario 01: Lista disponible Dado que la app lista con "infinite scroll", Cuando consulta el endpoint, Entonces se retorna colección paginada (200) mostrando placeholders y usando caché local si no hay conexión.						

Escenario 02: Sin registros

Dado el endpoint de consulta,
 Cuando no existen productos,
 Entonces se retorna 200 con lista vacía y la app muestra estado vacío con CTA para crear.

Story ID	User	Priority	Epic		
TS05	Developer	Alta	EP01 - Funciones básicas de inventario		
Title		API Productos por categoría			
Description					
Como desarrollador, necesito filtrar productos por categoría para agilizar búsquedas desde la app.					
Acceptance Criteria					
Escenario 01: Categoría existente Dado que el usuario selecciona una categoría (chips/filtros), Cuando se consulta el endpoint con la categoría, Entonces se retorna 200 con resultados ignorando mayúsculas/acentos y la app recuerda el filtro.					
Escenario 02: Sin coincidencias Dado el filtro por categoría, Cuando no hay productos asociados, Entonces se retorna 200 con lista vacía y la app permite limpiar filtros.					

Story ID	User	Priority	Epic		
TS06	Developer	Alta	EP01 - Funciones básicas de inventario		
Title		API Productos por etiqueta			
Description					
Como desarrollador, necesito permitir la consulta de productos por etiqueta para mejorar la organización y búsqueda.					
Acceptance Criteria					
Escenario 01: Etiqueta válida Dado que la app ofrece autocompletado de etiquetas recientes, Cuando el usuario selecciona una etiqueta y consulta, Entonces el backend retorna los productos (200) y la app permite combinar con otros filtros activos.					
Escenario 02: Etiqueta sin resultados Dado que se consulta por una etiqueta inexistente o sin datos, Cuando el backend no encuentra coincidencias, Entonces responde 200 con lista vacía y la app sugiere etiquetas relacionadas.					

Story ID	User	Priority	Epic		
TS07	Developer	Alta	EP01 - Funciones básicas de inventario		
Title		API Registro de inventario			
Description					
Como desarrollador, necesito endpoints para registrar movimientos/stock de inventario con validaciones de negocio.					
Acceptance Criteria					
Escenario 01: Registro válido Dado que la app permite escanear código/QR, Cuando se registra una entrada o salida con cantidad válida, Entonces el backend guarda el movimiento (201) y retorna el stock actualizado para mostrar en la app.					
Escenario 02: Registro inválido Dado el endpoint de inventario,					

Cuando la cantidad es negativa o la salida supera el stock,
Entonces se retorna 400 con detalle y la app resalta el campo con error.

Story ID	User	Priority	Epic
TS08	Developer	Alta	EP02 – Funciones de analítica, reportes y control avanzado
Title			API Inventarios por precio
Description			
Como desarrollador, necesito consultar inventarios filtrados por rango de precio para análisis desde la app.			
Acceptance Criteria			
Escenario 01: Rango válido Dado que el usuario ajusta un rango (slider), Cuando se consulta el endpoint con la moneda/locale del dispositivo, Entonces se retornan resultados (200) y pueden ordenarse por precio asc/desc.			
Escenario 02: Sin resultados Dado un rango sin coincidencias, Cuando el backend no encuentra registros, Entonces retorna 200 con lista vacía y la app sugiere ampliar el rango.			

Story ID	User	Priority	Epic
TS09	Developer	Alta	EP02 – Funciones de analítica, reportes y control avanzado
Title			API inventario por producto
Description			
Como desarrollador, necesito consultar el inventario de un producto específico para mostrar su disponibilidad.			
Acceptance Criteria			
Escenario 01: Producto existente Dado que la app escanea o selecciona un producto, Cuando se consulta por su ID, Entonces el backend retorna 200 con stock actual y últimos movimientos.			
Escenario 02: Producto inexistente Dado el endpoint por producto, Cuando el ID no existe o está inactivo, Entonces se retorna 404 con mensaje "no encontrado" y la app ofrece buscar otro producto.			

Story ID	User	Priority	Epic
TS10	Developer	Alta	EP02 – Funciones de analítica, reportes y control avanzado
Title			API Inventarios por fecha de ingreso
Description			
Como desarrollador, necesito filtrar inventarios por fecha de ingreso para consultas cronológicas desde la app.			
Acceptance Criteria			
Escenario 01: Fecha con registros Dado que el usuario selecciona un rango con el date picker del móvil, Cuando la app consulta con la zona horaria del dispositivo, Entonces el backend retorna 200 con los registros del período.			
Escenario 02: Fecha sin registros Dado el endpoint por fecha, Cuando el rango no tiene datos, Entonces se retorna 200 con lista vacía y la app propone rangos cercanos.			

Story ID	User	Priority	Epic
-----------------	-------------	-----------------	-------------

TS11	Developer	Alta	EP02 – Funciones de analítica, reportes y control avanzado			
Title	API Inventarios por cantidad					
Description						
Como desarrollador, necesito consultar inventarios por cantidad exacta o rango para monitorear niveles críticos o excedentes.						
Acceptance Criteria						
<p>Escenario 01: Rango/cantidad válida Dado que la app permite filtrar por cantidad o rango, Cuando se consulta el endpoint, Entonces se retornan productos (200) e indicadores de "bajo stock" si están bajo el mínimo configurado.</p> <p>Escenario 02: Sin coincidencias Dado el filtro por cantidad/rango, Cuando no hay resultados, Entonces se retorna 200 con lista vacía y la app permite guardar el filtro.</p>						

Story ID	User	Priority	Epic			
TS12	Developer	Alta	EP02 – Funciones de analítica, reportes y control avanzado			
Title	API Inventarios por proveedor					
Description						
Como desarrollador, necesito consultar inventarios asociados a un proveedor para análisis y gestión.						
Acceptance Criteria						
<p>Escenario 01: Proveedor existente Dado que el usuario selecciona un proveedor en la app, Cuando se consulta el endpoint con su identificador, Entonces retorna 200 con la lista paginada de inventarios.</p> <p>Escenario 02: Proveedor sin productos Dado el endpoint por proveedor, Cuando no hay inventarios vinculados, Entonces se retorna 200 con lista vacía y la app muestra un estado informativo.</p>						

Story ID	User	Priority	Epic			
TS13	Developer	Alta	EP05 - Alertas y notificaciones			
Title	API Registro de alertas					
Description						
Como desarrollador, necesito registrar alertas del sistema (p.ej., bajo stock) para su notificación en la app móvil.						
Acceptance Criteria						
<p>Escenario 01: Alerta registrada Dado un evento de bajo stock detectado en la app o backend, Cuando se envía la alerta con datos válidos al endpoint, Entonces se registra (201) y la app puede disparar notificación local/push.</p> <p>Escenario 02: Datos inválidos Dado el endpoint de alertas, Cuando el tipo es inválido o faltan campos requeridos, Entonces se retorna 400 con detalle de validaciones.</p>						

Story ID	User	Priority	Epic
TS14	Developer	Alta	EP05 - Alertas y notificaciones
Title	API Consulta de alertas		
Description			

Como desarrollador, necesito listar alertas para su visualización y seguimiento en la aplicación móvil.

Acceptance Criteria

Escenario 01: Listado disponible

Dado que la app muestra un feed con "pull to refresh",

Cuando consulta el endpoint,

Entonces se retornan alertas paginadas (200) ordenadas por severidad/fecha y se pueden marcar como leídas.

Escenario 02: Sin alertas

Dado el endpoint de consulta de alertas,

Cuando no existen registros,

Entonces se retorna 200 con lista vacía y la app ofrece ir a configuración de alertas.

Story ID	User	Priority	Epic		
TS15	Developer	Alta	EP01 - Funciones básicas de inventario		
Title		Description			
Como desarrollador, necesito endpoints para registrar combos de productos respetando reglas de composición.					
Acceptance Criteria					
Escenario 01: Combo registrado Dado que el usuario arma un combo en la app seleccionando productos y cantidades, Cuando envía la creación, Entonces el backend valida reglas (stock mínimo, no repetidos) y crea el combo (201) de forma atómica.					
Escenario 02: Datos inválidos Dado el endpoint de combos, Cuando la composición es inválida o hay productos inexistentes, Entonces se retorna 400 con detalle y la app conserva la selección para corregir.					

Story ID	User	Priority	Epic		
TS16	Developer	Alta	EP01 - Funciones básicas de inventario		
Title		Description			
Como desarrollador, necesito consultar combos registrados para mostrarlos como opciones agrupadas en la app móvil.					
Acceptance Criteria					
Escenario 01: Listado de combos Dado que la app lista combos con imágenes y precio total, Cuando consulta el endpoint, Entonces el backend retorna 200 con lista paginada y la app permite ordenar y guardar favoritos.					
Escenario 02: Sin combos Dado el endpoint de combos, Cuando no existen registros, Entonces se retorna 200 con lista vacía y la app muestra un estado sin datos con botón para crear combo.					

Spike Story

Story ID	User	Priority	Epic		
SPK01	Equipo de desarrollo	Baja	EP03 – Innovación y valor agregado		
Title		Description			
Como equipo de desarrollo, investiga gramática mínima y extracción de entidades (cantidad, producto) para registrar movimientos por voz con manejo de ambigüedad.					

Acceptance Criteria

Escenario 01: Extracción de entidades con confianza mínima

- Dado que se recibe un comando válido,
- Cuando procesa audio o texto,
- Entonces extrae cantidad y referencia de producto con un umbral de confianza definido.

Escenario 02: Manejo de ambigüedad

- Dado que existen múltiples coincidencias,
- Cuando detecta ambigüedad,
- Entonces solicita desambiguación dentro del flujo técnico.

Escenario 03: Dictado consecutivo

- Dado que el usuario dicta varios comandos,
- Cuando los recibe en secuencia,
- Entonces procesa cada uno y conserva estado hasta terminación.

Story ID	User	Priority	Epic
SPK02	Equipo de desarrollo	Alta	EP04 – Cuentas y seguridad
Title			
Roles y permisos en API			
Description			
Como equipo de desarrollo, investiga un modelo RBAC con matriz rol permiso y puntos de control en API para operaciones de inventario y administración.			
Acceptance Criteria			
Escenario 01: Autorización según política			
<ul style="list-style-type: none"> • Dado que un rol tiene permisos definidos, • Cuando solicita una operación restringida, • Entonces autoriza o rechaza de acuerdo con la política. 			
Escenario 02: Intentos no autorizados			
<ul style="list-style-type: none"> • Dado que un usuario carece del permiso, • Cuando intenta modificar recursos, • Entonces rechaza la acción y registra el intento. 			
Escenario 03: Efecto de cambios de rol			
<ul style="list-style-type: none"> • Dado que se actualizan las asignaciones de rol, • Cuando el cambio entra en vigor, • Entonces aplica permisos vigentes sin inconsistencias. 			

Story ID	User	Priority	Epic
SPK03	Equipo de desarrollo	Baja	EP03 – Innovación y valor agregado
Title			
Geolocalización y modelo de ubicación (GPS y QR)			
Description			
Como equipo de desarrollo, investiga captura de lat/long, precisión, almacenamiento, permisos y asociación de ubicación por QR para productos y entregas.			
Acceptance Criteria			
Escenario 01: Persistencia de coordenadas			
<ul style="list-style-type: none"> • Dado que los permisos de ubicación están concedidos, • Cuando captura la ubicación, • Entonces persiste coordenadas con precisión y timestamp. 			

Escenario 02: Flujo sin permisos

- Dado que los permisos son denegados,
- Cuando se requiere ubicación,
- Entonces documenta flujo de entrada manual de dirección o coordenadas.

Escenario 03: Asociación por QR

- Dado que se dispone de un QR válido,
- Cuando asocia la ubicación,
- Entonces vincula el producto al punto definido y lo hace consultable.

Story ID	User	Priority	Epic			
SPK04	Equipo de desarrollo	Baja	EP03 – Innovación y valor agregado			
Title	Predicción de reabastecimiento					
Description						
Como equipo de desarrollo, investiga una fórmula base (media móvil + lead time) con parámetros ajustables para sugerir fecha y cantidad de reabastecimiento.						
Acceptance Criteria						
Escenario 01: Sugerencia con supuestos explícitos						
<ul style="list-style-type: none">• Dado que existe histórico de ventas y stock mínimo,• Cuando calcula demanda proyectada,• Entonces sugiere fecha y cantidad con supuestos explícitos.						
Escenario 02: Recálculo por parámetros						
<ul style="list-style-type: none">• Dado que los parámetros son ajustables,• Cuando actualiza valores,• Entonces recálcula sugerencias con trazabilidad de versiones.						
Escenario 03: Detección de quiebre inminente						
<ul style="list-style-type: none">• Dado que el consumo supera la previsión,• Cuando se detecta desvío relevante,• Entonces marca riesgo de quiebre y registra recomendación prioritaria.						

Story ID	User	Priority	Epic			
SPK054	Equipo de desarrollo	Baja	EP03 – Innovación y valor agregado			
Title	Escaneo por lotes con cámara rápida (visión ML)					
Description						
La app toma una foto al lote; una API de visión (p.ej., Google ML Kit) devuelve etiquetas genéricas. El usuario confirma producto exacto y cantidad antes de registrar; se puede saltar a su ubicación en el almacén virtual.						
Acceptance Criteria						
Escenario 01: Etiquetado y confirmación						
<ul style="list-style-type: none">• Dado una imagen de lote,• Cuando el servicio devuelve top-N etiquetas,• Entonces el usuario selecciona producto y define cantidad en un paso.						
Escenario 02: Registro por lotes						
<ul style="list-style-type: none">• Dado múltiples fotos consecutivas,• Cuando procesa cada captura,• Entonces registra movimientos en cola con feedback de éxito/error por ítem.						
Escenario 03: Acceso a ubicación						

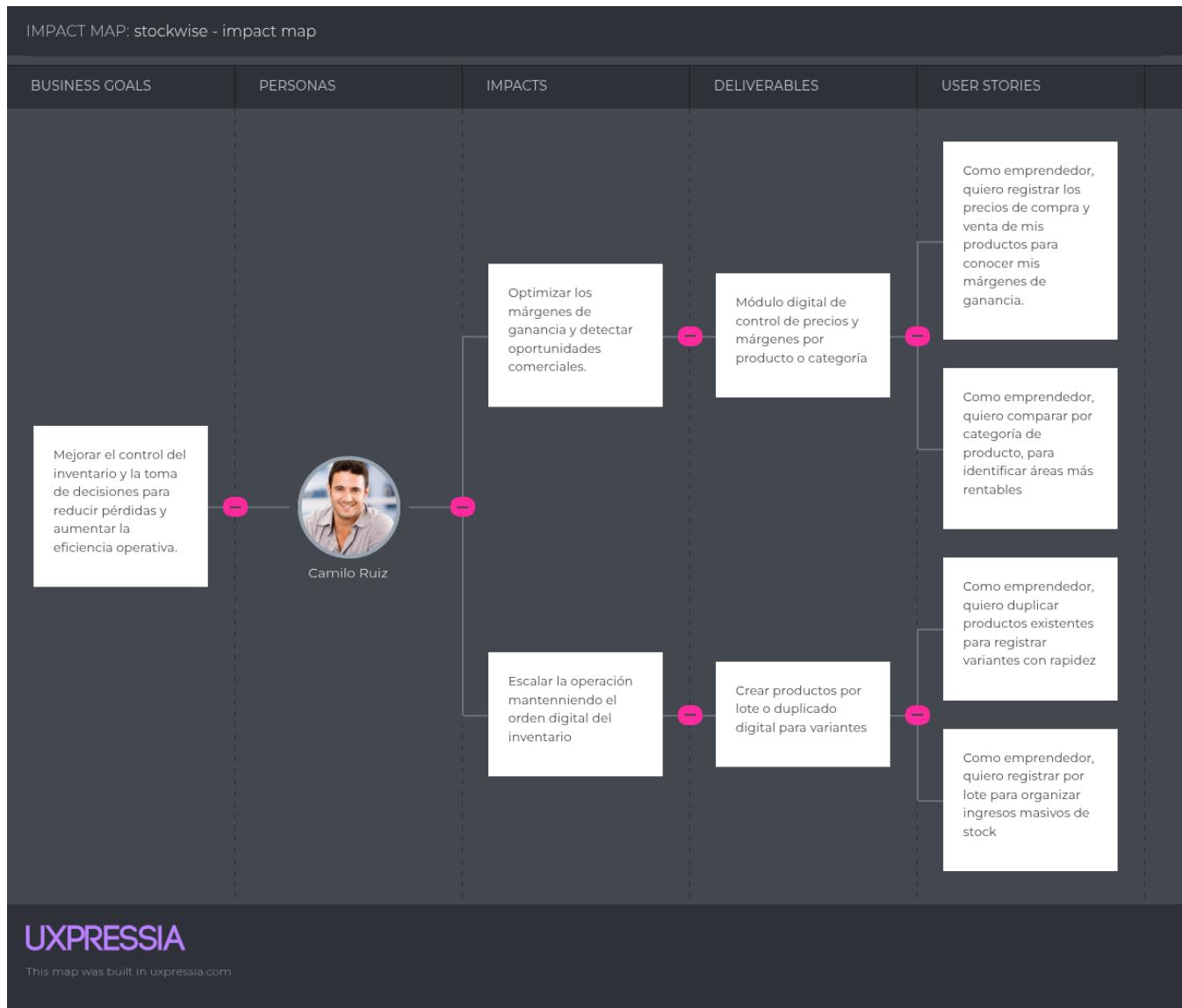
- Dado un producto confirmado,
- Cuando el usuario lo solicita,
- Entonces navega a su ubicación (QR o GPS) en el mapa del almacén.

2.4.2. Impact Mapping

Segmento Objetivo 1: Bodegas especializadas por rubro



Segmento Objetivo 2: Startups y emprendedores en expansión con necesidades logísticas



2.4.3. Product Backlog

#	User Story ID	Título	Descripción	Story Points
1	US01	Registra producto nuevo	Como usuario, quiero registrar un nuevo producto en mi inventario, para tener un control detallado desde el inicio.	5
2	US05	Generar alertas por bajo stock	Como usuario, quiero recibir alertas por bajo stock, para reabastecerme a tiempo.	5
3	US07	Añadir etiquetas para productos	Como usuario, quiero clasificar productos con etiquetas, para organizar el stock y facilitar búsquedas.	5
4	US08	Estadística de productos más vendidos	Como usuario, quiero ver los productos más vendidos para identificar los de mayor demanda.	5
5	US09	Estadística de categoría más vendida	Como usuario, quiero saber qué categorías son las más vendidas para tomar decisiones comerciales.	5
6	US11	Estadística de stock promedio	Como usuario, quiero ver el stock promedio para detectar quiebres o sobrestock.	5

#	User Story ID	Título	Descripción	Story Points
7	US15	Configurar roles y permisos	Como administrador, quiero definir roles y permisos para proteger la información del sistema.	5
8	US17	Registrar lote de productos	Como usuario, quiero registrar productos por lote para facilitar el control masivo.	5
9	US20	Crear kits o combos de productos	Como usuario, quiero agrupar productos en kits para venderlos como un solo paquete.	5
10	US26	Registrar productos por voz	Como usuario, quiero registrar productos mediante comandos de voz para ganar rapidez y comodidad.	5
11	US27	Geolocalización de productos y entregas	Como usuario, quiero integrar GPS para registrar procedencia y puntos de entrega para optimizar trazabilidad.	5
12	US28	Localizar productos con mapa y QR	Como usuario, quiero ubicar productos dentro del almacén usando mapa y QR para encontrarlos con precisión.	5
13	US02	Editar información de producto	Como usuario, quiero editar la información de un producto registrado, para corregir o actualizar datos.	3
14	US03	Registrar salida de producto	Como usuario, quiero registrar la salida de productos, para mantener actualizado el stock en tiempo real.	3
15	US04	Ver historial de movimientos	Como usuario, quiero ver el historial de movimientos, para rastrear los cambios en el inventario.	3
16	US06	Buscar productos en inventario	Como usuario, quiero buscar productos por nombre o categoría, para encontrarlos rápidamente.	3
17	US10	Ticket promedio	Como usuario, quiero conocer el ticket promedio para evaluar efectividad de precios.	3
18	US12	Generar reportes de inventario	Como usuario, quiero generar reportes detallados del inventario para tomar decisiones informadas.	3
19	US13	Controlar productos caducados	Como usuario, quiero identificar productos caducados para evitar pérdidas.	3
20	US14	Registrar devolución de productos	Como usuario, quiero registrar devoluciones al inventario para mantener el stock actualizado.	3
21	US16	Gestionar proveedores	Como usuario, quiero registrar y gestionar proveedores para facilitar la reposición de productos.	3
22	US18	Registrar precios de compra y venta	Como usuario, quiero registrar precios para calcular márgenes de ganancia.	3
23	US19	Registrar unidades de medida	Como usuario, quiero definir unidades de medida para cada producto.	3
24	US21	Agregar ubicación de producto	Como usuario, quiero registrar la ubicación del producto para facilitar su localización.	3
25	US22	Personalizar columnas en inventario	Como usuario, quiero personalizar la vista del inventario según mis necesidades.	3
26	US23	Duplicar ficha de producto	Como usuario, quiero duplicar productos para registrar variantes fácilmente.	3
27	US25	Ver inventario por estantería	Como usuario, quiero ver inventario por ubicación física para facilitar el control por zonas.	3
28	US29	Exportar reportes	Como usuario, quiero exportar reportes a PDF o Excel para compartir y analizar fuera de la plataforma.	3
29	US30	Personalizar notificaciones	Como usuario, quiero configurar notificaciones personalizadas para alertas de stock, caducidad y otras condiciones.	3
30	US24	Añadir notas internas a productos	Como usuario, quiero añadir comentarios internos para registrar incidencias sin mostrarlas al cliente.	2

2.5. Strategic-Level Domain-Driven Design

En esta sección se describe el proceso utilizado en las decisiones del negocio, aplicando el enfoque de Domain Driven Design. A través de esta metodología, se busca identificar y definir los distintos Bounded Contexts que conforman el sistema, es decir, subconjuntos con límites lógicos y naturales en función del dominio del problema. Para lograrlo, el equipo emplea herramientas como EventStorming y el Bounded Context Canvas, las cuales permiten visualizar, analizar y delimitar las responsabilidades, flujos de eventos y relaciones clave dentro del sistema.

2.5.1. EventStorming

En este apartado se documenta el proceso del Event Storming, una herramienta clave para construir una primera aproximación al modelado general del dominio del negocio. Esta técnica colaborativa permite identificar eventos relevantes, actors, commands y aggregates dentro del sistema. Con ello, el objetivo principal es explorar nuestro dominio para descubrir patrones de software relevantes, límites y conexiones claras entre nuestros bounded context y sentar las bases para una buena arquitectura. Para ello, se realizaron los siguientes pasos en la herramienta Lucidchart bajo la normativa del DDD Estratégico. https://miro.com/app/board/uXjVJlfQcmY=/?share_link_id=739758304264

2.5.1.1. Candidate Context Discovery

A continuación se detalla el proceso y la evolución final del event storming.

Step 1: Unstructured exploration

Primero tuvimos una lluvia de ideas en la cual nos propusimos identificar exhaustivamente todos los eventos que ocurrirán en nuestro aplicativo, para asegurarnos de abarcar una amplia gama de posibilidades y ofrecer una experiencia completa y satisfactoria a nuestros usuarios.

A continuación, se presentan una serie de figuras que evidencian este proceso de exploración inicial.

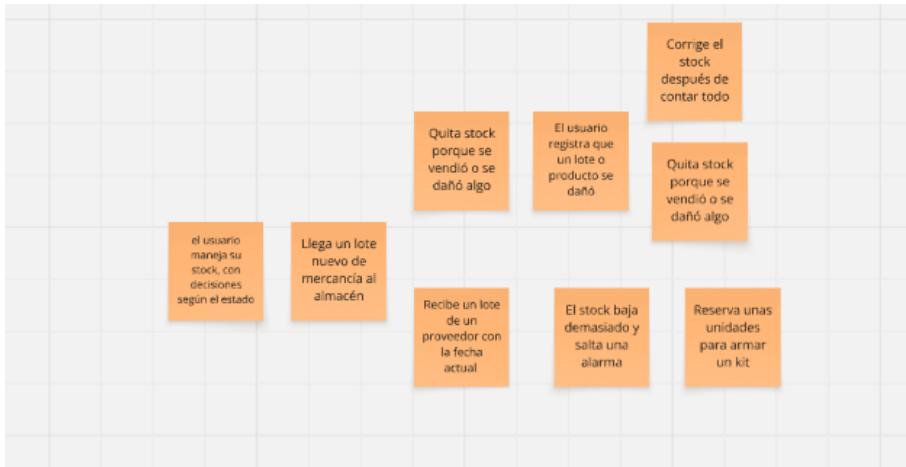
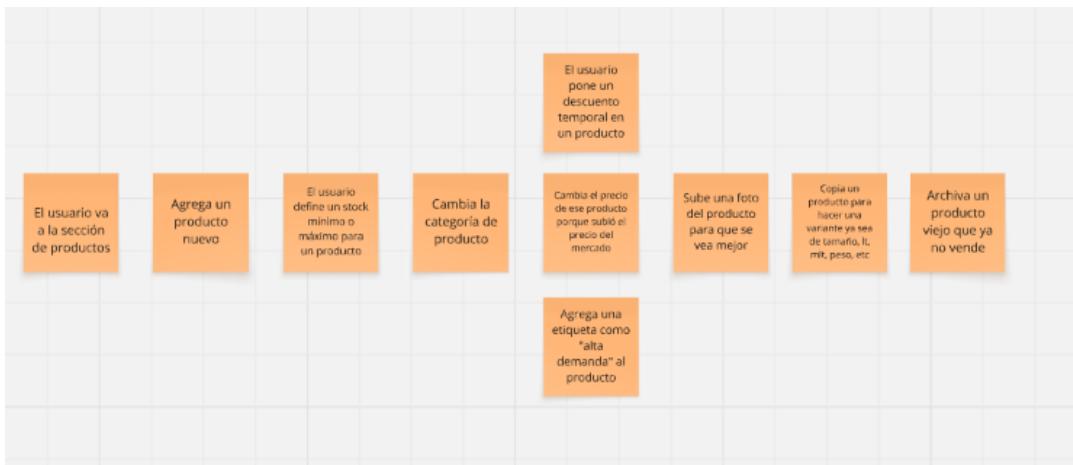
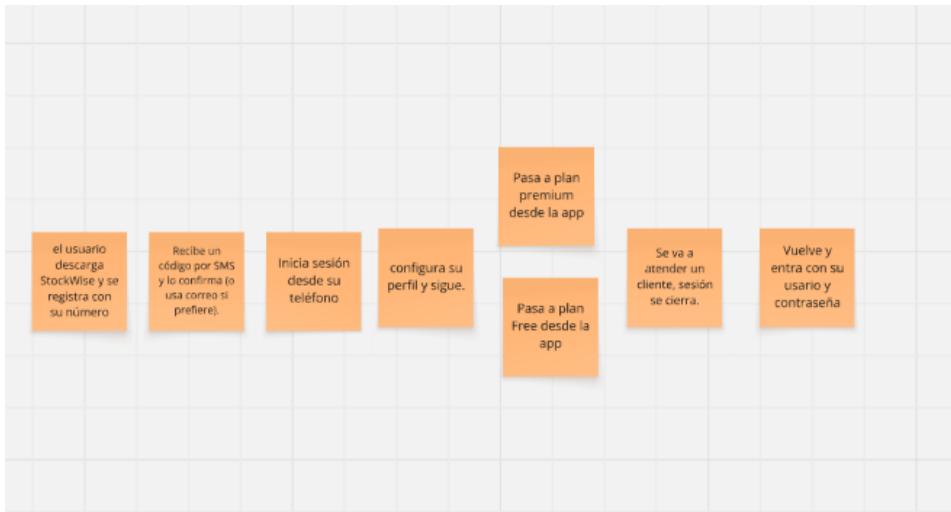
Unstructured exploration									
Alguien crea una cuenta nueva en la app	El usuario entra con su mail y contraseña.	Se cierra la sesión porque se va a almorzar.	Un jefe le da permisos de admin a un empleado.	El usuario cambia su foto de perfil o el idioma a inglés.	Pide resetear la contraseña porque la olvidó.	Termina de cambiar la contraseña y ya puede entrar de nuevo.	Sube de plan gratis a uno pagado para más productos.	Baja de plan porque ya no necesita tanto.	La cuenta se duerme por no usarla en meses.
Agrega un producto nuevo, como "camisetas rojas".	Cambia el precio de ese producto porque subió el dólar.	Copia un producto para hacer una variante, tipo talla S.	Archiva un producto viejo que ya no vende.	Pone fecha de caducidad a un yogur que acaba de llegar.	Agrega una etiqueta como "alta demanda" al producto.	Quita una etiqueta porque ya no aplica.	Sube una foto del producto para que se vea mejor.	Cambia la categoría de "ropa" a "accesorios".	Ajusta el precio por una promo de fin de semana.
Llega un lote nuevo de mercancía al almacén.	Quita stock porque se vendió o se dañó algo.	Corrige el stock después de contar todo manualmente.	Recibe un lote de un proveedor con la fecha de hoy.	Un lote de comida se vence y hay que tirarlo.	Termina el conteo mensual y ve si hay errores.	El stock baja demasiado y salta una alarma.	Reserva unas unidades para armar un kit.	Liberá esa reserva porque al final no lo usa.	Toma una foto rápida del inventario actual para un reporte.
Salta una alerta porque el stock de leche está bajo.	Alerta de que algo se va a vencer en una semana.	Avisa que hay demasiado stock de algo que no se mueve.	El usuario dice "ok, vi la alerta, la marco como hecha".	Ignora una alerta porque ya lo manejó de otra forma.	Manda la alerta al jefe porque es grave.	Envía un mail o notificación al teléfono del usuario.	Cambia el límite de stock bajo de 10 a 5 unidades.	Procesa un montón de alertas al final del día.	Guarda el historial viejo de alertas en un archivo.
Arma un kit nuevo con camisetas y pantalones.	Cambia el kit agregando más de un producto.	Publica el kit para que lo vean en la tienda online.	Desarma el kit porque ya no lo quiere.	Chequea si hay stock suficiente para vender el kit.	Vende un kit y resta todo el stock de adentro.	Genera un reporte de qué se vendió esta semana.	Exporta el reporte a Excel para mandárselo al contador.	Registra una entrada o salida en el historial.	Captura números como "stock promedio" para ver tendencias.
El usuario confirma su correo tras registrarse, para activar su cuenta y empezar a usar la app	La sesión se cierra automáticamente tras un tiempo sin actividad, para proteger la cuenta.	El usuario pone un descuento temporal en un producto, como "20% off en algún producto".	El usuario define un stock mínimo o máximo para un producto, para personalizar alertas.	El usuario registra que un lote o producto se dañó (e.g., latas aplastadas), ajustando el inventario.	El usuario mueve stock de una bodega a otra (útil para startups con múltiples almacenes)	El usuario crea una alerta personalizada, como "avisar si el stock de un producto baja de 10".	El usuario marca una alerta como "urgente" para que salte primero en el dashboard.	El usuario crea un reporte que muestra tendencias de ventas (e.g., qué productos se mueven más).	El usuario envía el reporte por email o lo comparte con el equipo desde la app

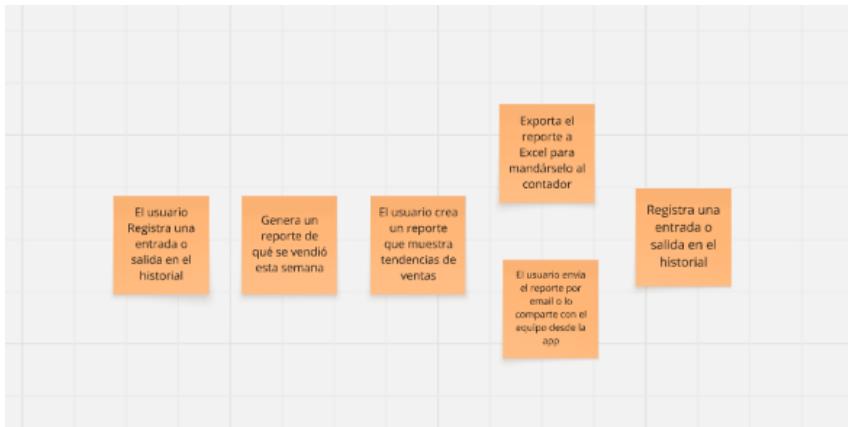
En esta figura se aprecia un mural con todos los eventos del dominio que fueron rescatados de la lluvia de ideas que realizó el equipo.

Step 2 : Timeline construction

Una vez identificados los eventos, organizamos la secuencia cronológica en la que estos ocurren dentro del flujo del sistema. Esto nos permitió identificar la secuencia natural del negocio y clarificar la narrativa del dominio.

A continuación, se presentan una serie de capturas de pantalla que ilustran cómo estructuramos esta línea temporal.

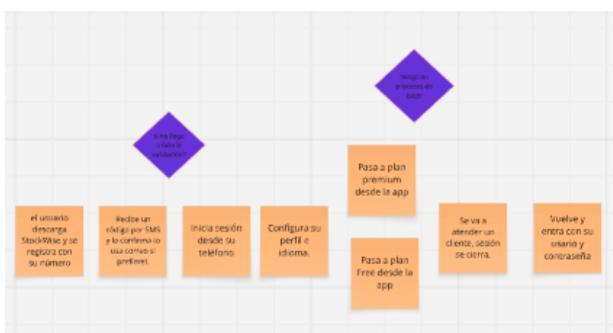




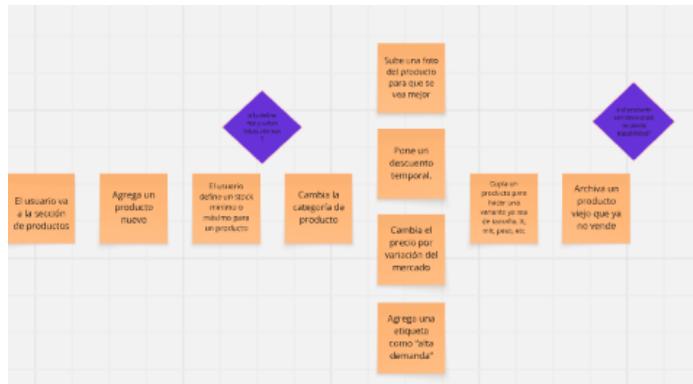
Step 3: Pain Points

En este paso identificamos los puntos críticos que requieren especial atención dentro de nuestro dominio, ya sea por su nivel de impacto en el negocio, riesgo potencial o complejidad. Para facilitar su análisis y visualización, se ha elaborado un diagrama colaborativo que refleja dichos puntos críticos.

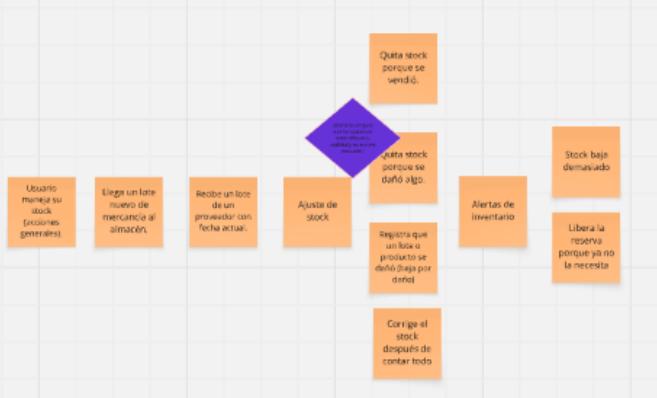
A continuación, se presentan una serie de figuras que evidencian el proceso realizado durante la identificación y análisis de los puntos críticos del dominio. Estas capturas complementan el diagrama general elaborado en Miro y permiten observar con mayor detalle los elementos clave identificados.



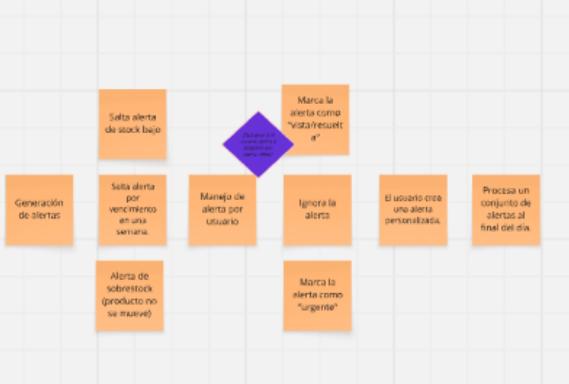
En esta imagen se aprecia la identificación de incertidumbres sobre cómo manejar la verificación (canal, reintentos, expiración y fallback) y cuándo activar la cuenta si el código no llega o falla. Al resolver estas interrogantes —reenvío OTP, cambio de canal, rate-limit y soporte— mejoraremos el alta de usuarios y reduciremos abandonos.



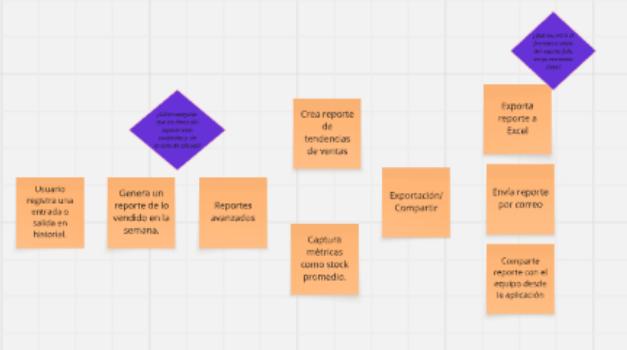
En esta imagen se aprecia la identificación de incertidumbres sobre de dónde salen los valores por defecto, si son obligatorios y cómo impactan las alertas. Al imponer validación y defaults por categoría, habilitamos reposición oportuna y alertas útiles sin fricción para el usuario.



En esta imagen se aprecia la identificación de incertidumbres sobre cómo cerrar una alerta: si se exige confirmación de atención o escalamiento previo, quién puede marcarla como "vista/resuelta", y cómo tratar estados como ignorada o urgente cuando hay alertas por stock bajo, vencimiento o sobrestock; además, hay dudas sobre la creación de alertas personalizadas y el procesamiento por lotes al final del día.



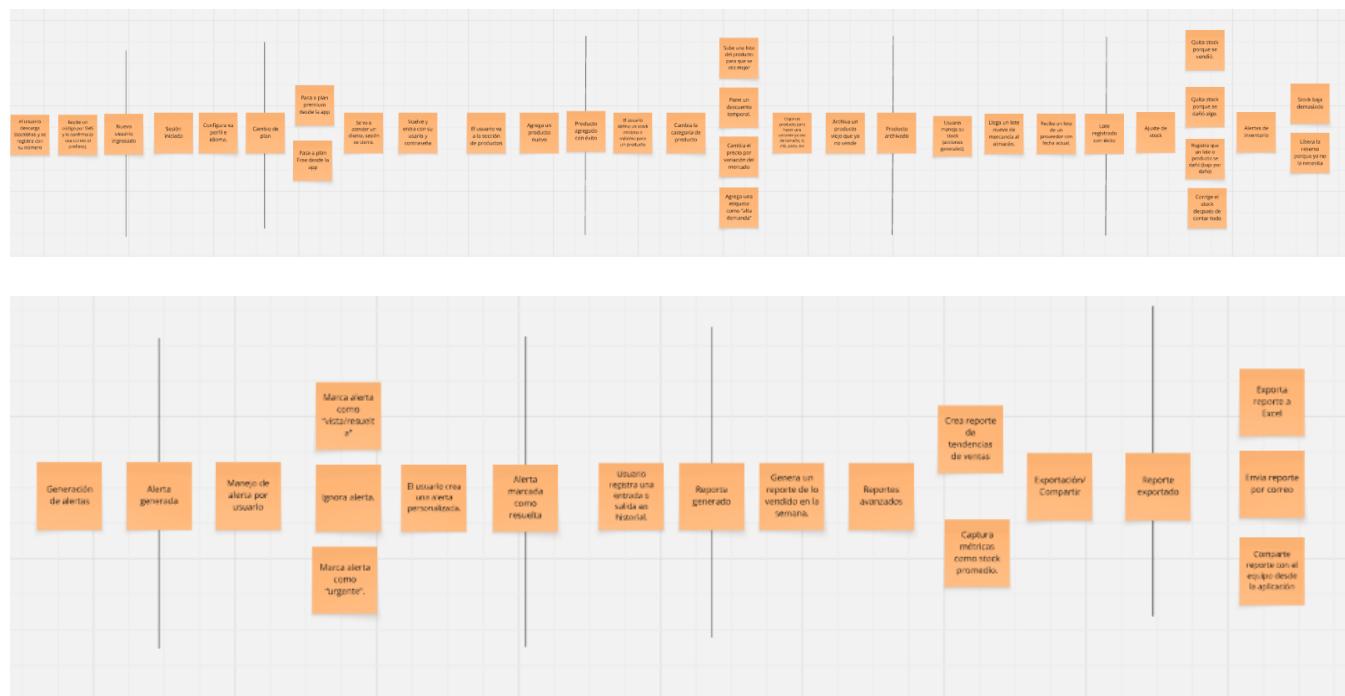
En esta imagen se aprecia la identificación de incertidumbres sobre qué datos "validados" deben alimentar la auditoría y los reportes (para evitar inconsistencias), así como el alcance y filtros de los informes (ventas semanales, tendencias, métricas como stock promedio) y quién puede exportar/compartir (Excel, correo) según el plan.



En esta imagen se aprecia la identificación de incertidumbres sobre cómo garantizar que un "ajuste de stock" refleje la realidad y no oculte errores o maniobras: cuándo corresponde usar ajuste versus quitar stock por venta o baja por daño, qué evidencias (motivo, usuario, documento/foto) y doble validación exigir, cómo conciliar el conteo físico sin duplicar salidas previas, cómo impactar lotes y reservas (p. ej., liberar reservas ya innecesarias) y cómo disparar alertas por umbral después de cada movimiento.

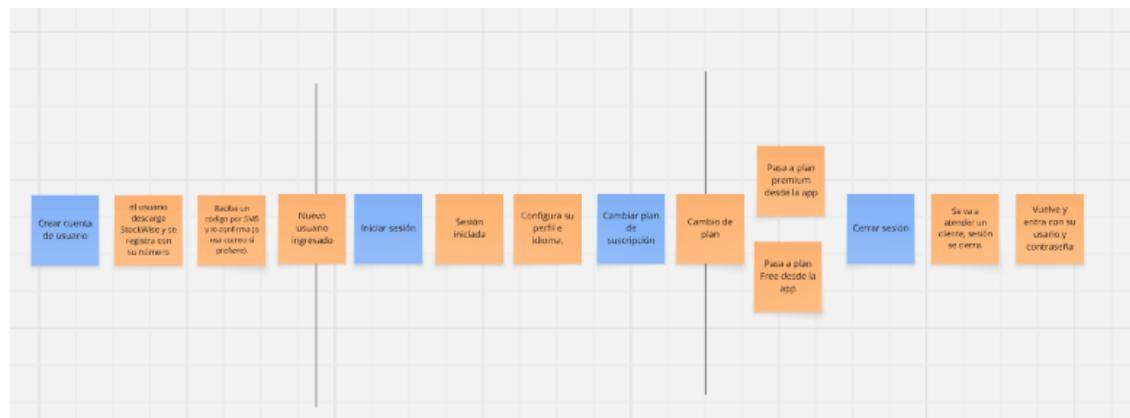
Step 4: Pivotal Points

En esta sección se marcaron los puntos clave que denotan transiciones significativas dentro del proceso de desarrollo, especialmente aquellos eventos o decisiones que implican un cambio en el comportamiento del sistema o en la forma en que los distintos Bounded Contexts interactúan entre sí. Como se puede observar, los pivotal points son sumamente importantes. Por ello, para poder comprender y apreciar el diseño realizado en este paso, se muestran las siguientes imágenes.

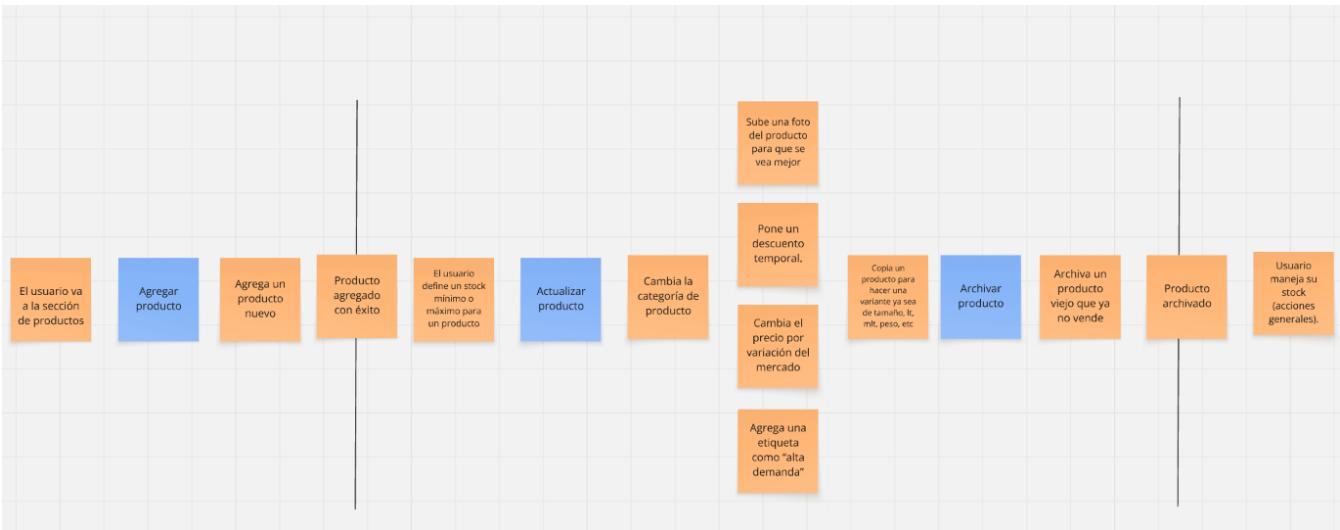


Step 5: Commands

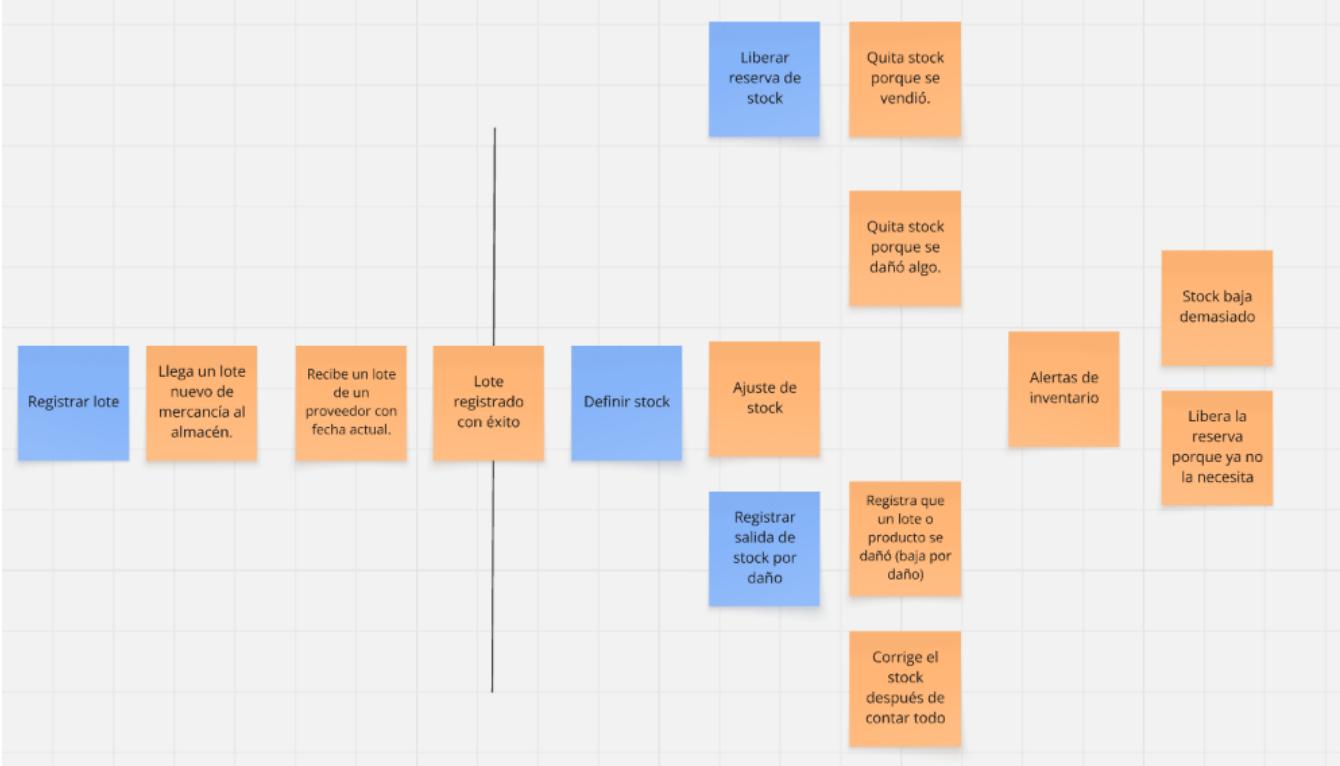
En este paso se definen los comandos que representan las acciones que los usuarios pueden ejecutar en el sistema. Estos comandos son fundamentales para desencadenar eventos y modificar el estado del sistema, y deben estar alineados con los eventos identificados previamente. A continuación, se muestran los comandos relevantes para nuestro dominio.



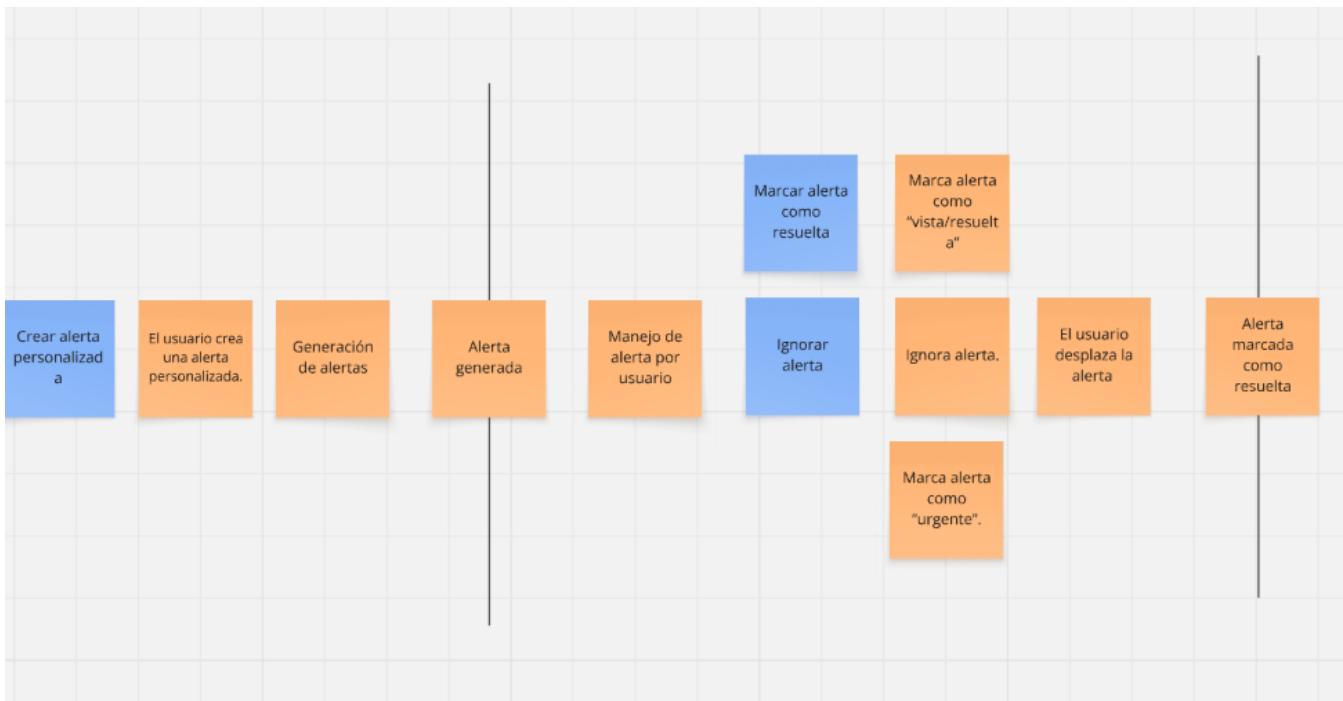
Crear cuenta de usuario registra la cuenta y dispara la verificación (SMS/Email); Iniciar sesión autentica factores y abre una sesión válida; Cambiar plan de suscripción solicita upgrade/downgrade, coordina el cobro y aplica el plan activo tras confirmación; Cerrar sesión invalida la sesión/token en curso.



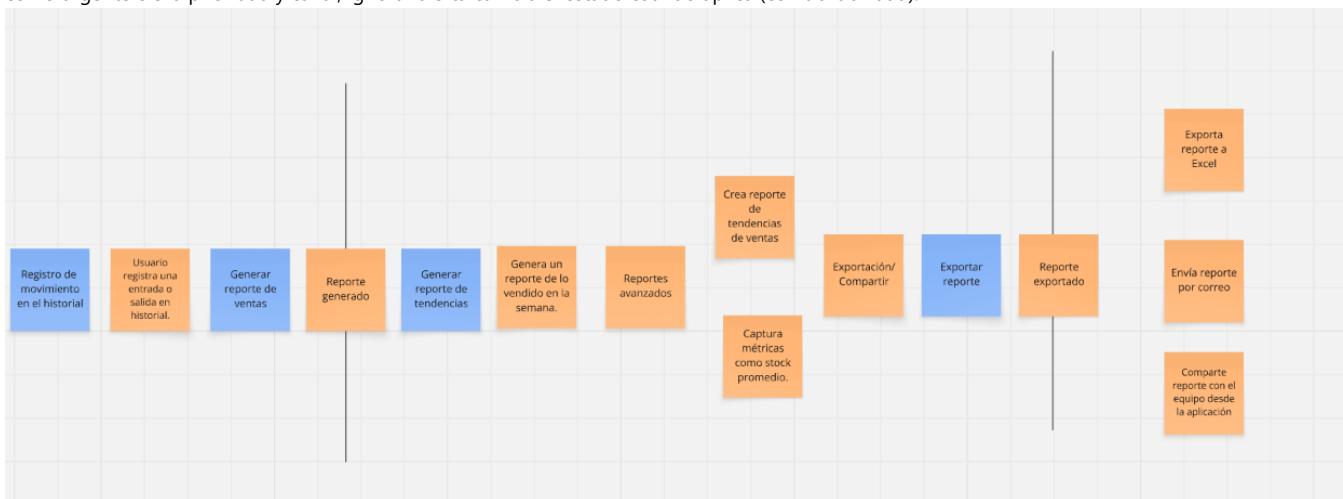
Agregar producto da de alta un ítem con sus datos base; Actualizar producto modifica atributos (nombre, descripción, imágenes, etc.); Cambiar la categoría de producto reasigna su clasificación; Archivar producto cambia el estado a “archivado” cuando no existen existencias ni reservas activas.



Registrar lote ingresa stock por lote (cantidad, almacén, vencimiento); Definir stock establece umbrales mínimo/máximo para alertas y reposición; Ajuste de stock corrige existencias con doble validación cuando procede; Liberar reserva de stock devuelve a disponible las cantidades apartadas que ya no se necesitan.



Generar alerta crea una alerta con severidad y canales según reglas; Marcar alerta como resuelta cierra el caso tras confirmación de atención; Marcar como urgente eleva prioridad y canal; Ignorar alerta cambia el estado cuando aplica (con trazabilidad).



Generar reporte ejecuta una consulta/KPI sobre datos validados; Exportar reporte materializa el resultado (Excel/PDF) con sello/firma; Enviar/Compartir reporte distribuye el archivo por correo o canal elegido y registra la entrega.

Step 6: Policies

En este paso se describen políticas que rigen decisiones y acciones dentro del sistema, asegurando que este se aline con las necesidades del usuario y los objetivos del negocio.

Los siguientes frames contienen políticas que indican diferentes restricciones o condiciones descritas a continuación:

-Validar contacto antes de activar la cuenta. El correo o teléfono debe confirmarse (OTP) antes de permitir inicio de sesión o uso de la app.

-Un solo plan activo por cuenta. La suscripción vigente es única (Free o Premium) y el cambio de plan solo se aplica cuando el pago está confirmado.

-Producto con datos mínimos obligatorios. Todo producto debe tener categoría y precio; además se definen umbrales de stock min/máx para integrarse con inventario.

-No archivar con stock/Reserva. Prohibido archivar si existen existencias o reservas; primero se libera/traslada el stock.

-Perecibles con fecha de vencimiento. Lotes de productos perecibles deben registrar expiryDate.

-Ajuste con doble validación. Cualquier ajuste de inventario requiere aprobación en un segundo paso.

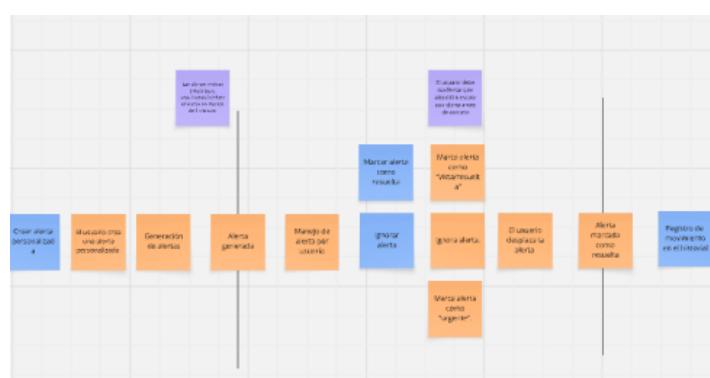
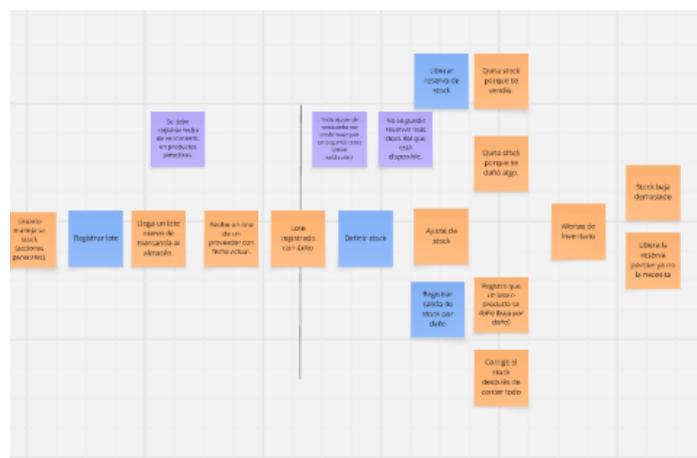
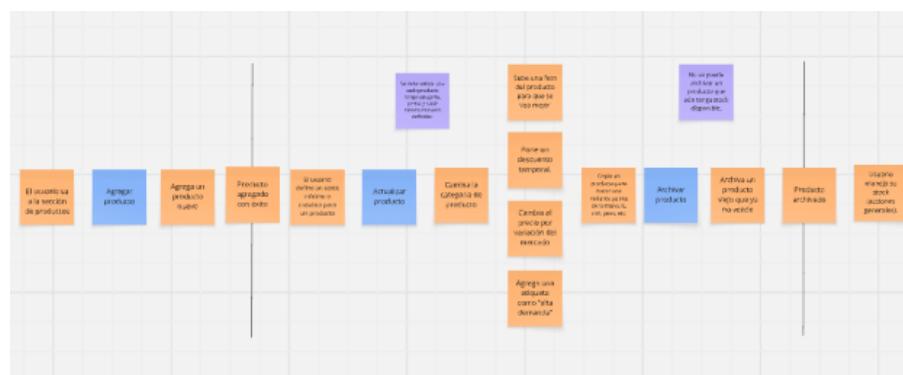
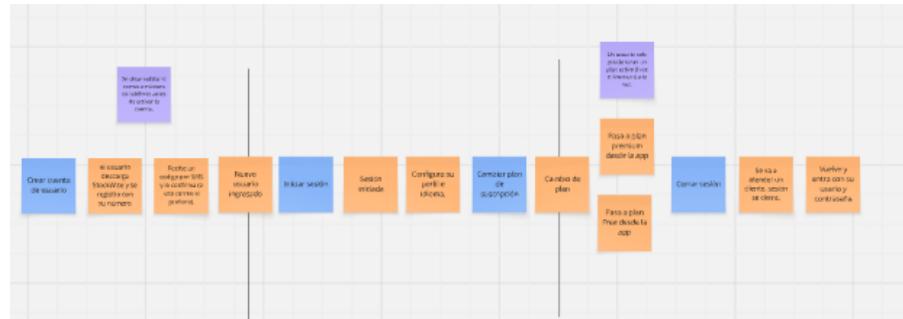
-No reservar más que disponible. Las reservas nunca pueden exceder el stock disponible.

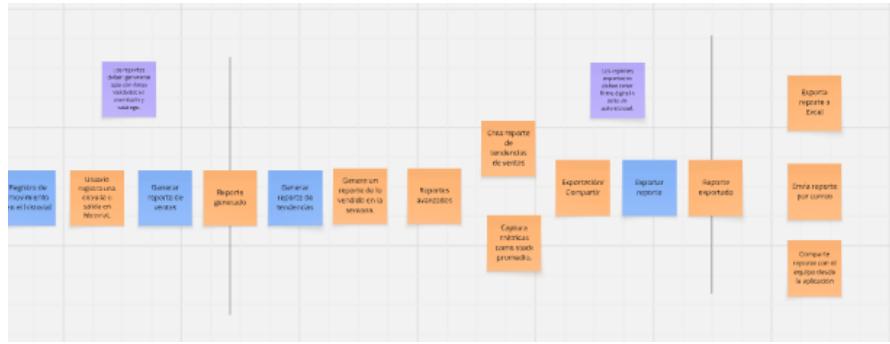
-Críticas en < 1 minuto. Las alertas de alta severidad (stock bajo, vencimiento) deben generarse y enviarse en menos de un minuto.

-ACK obligatorio antes de cerrar. El usuario debe reconocer/escalar una alerta antes de marcarla como resuelta.

-Reportes solo con datos validados. Los informes se generan a partir de eventos confirmados (inventario/catálogo).

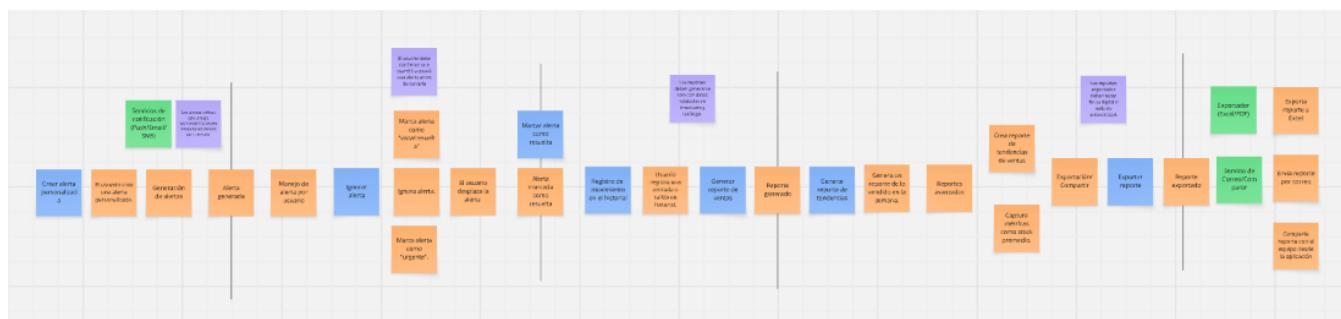
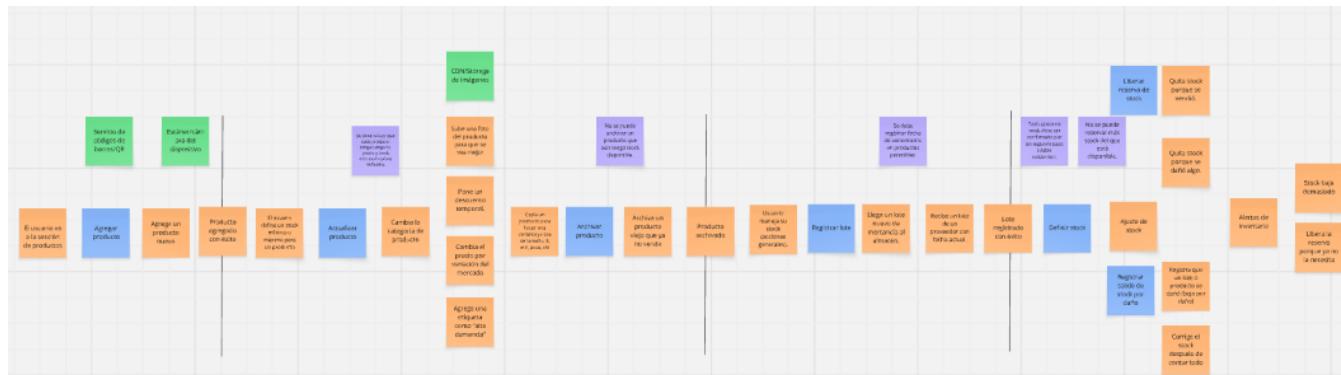
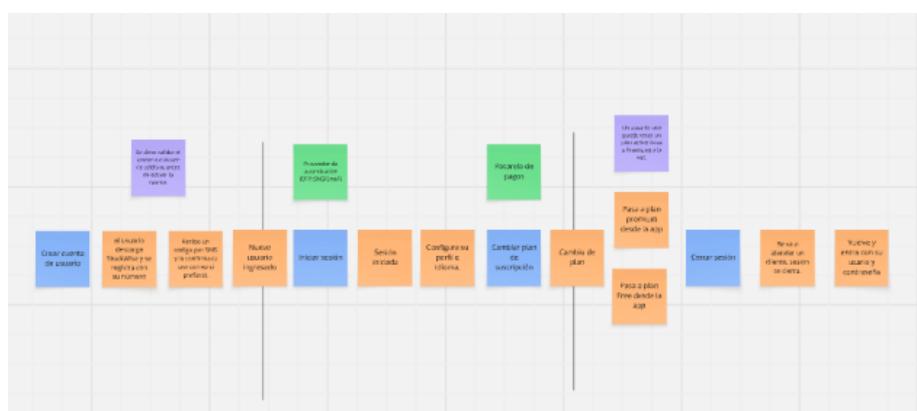
-Exportos con firma/sello. Los archivos exportados (Excel/PDF) deben incluir firma digital o sello de autenticidad.





Step 7 & 8: Read Models & External Systems

Estos dos pasos conforman la aparición de los read models y los external systems respectivamente. Por un lado, los read models son proyecciones que representan vistas personalizadas, dashboards, o reportes. Por otro lado, los external systems en DDD representan aquellos componentes o servicios externos con los que el sistema necesita interactuar, como pasarelas de pago, servicios de autenticación, sistemas de terceros, o integraciones con otras plataformas empresariales.

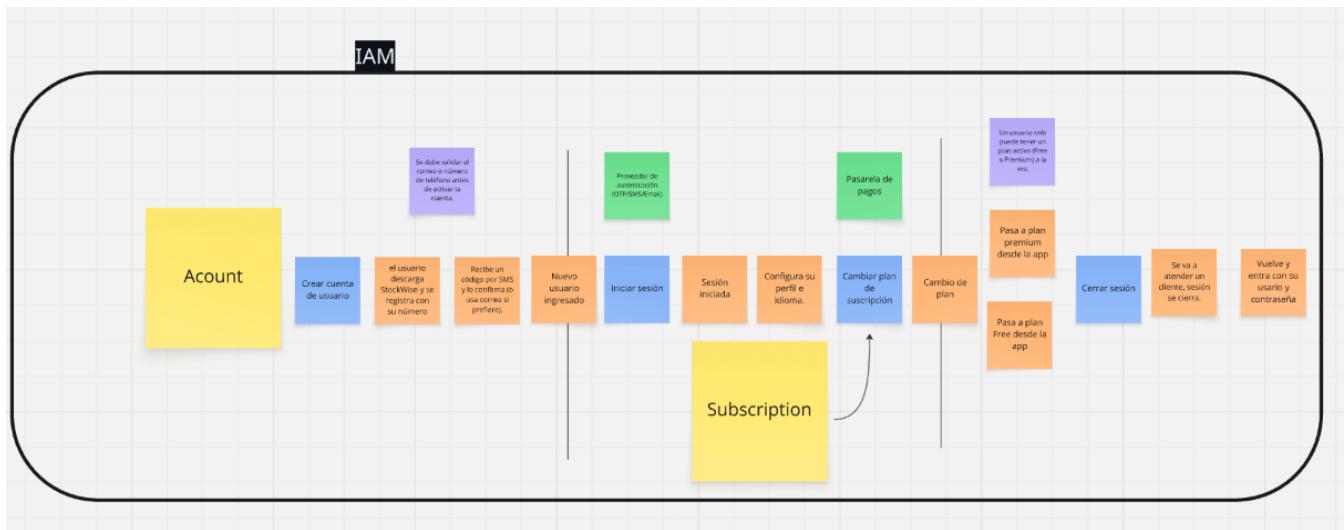


Step 9 & 10: Aggregates & Bounded Contexts

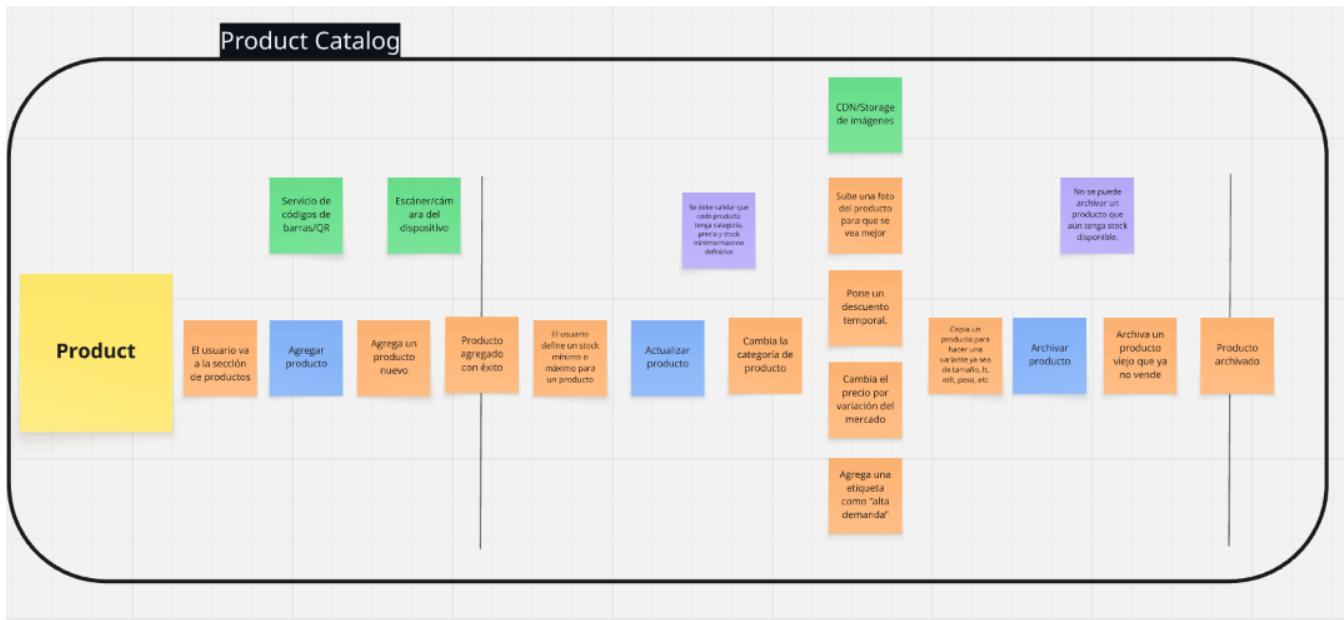
En los pasos 9 y 10 se introducen dos conceptos clave del Domain-Driven Design: Aggregates y Bounded Contexts. Los aggregates representan unidades de consistencia dentro del modelo de dominio, agrupando entidades y objetos de valor que deben mantenerse coherentes bajo una misma

raíz (aggregate root). Por otro lado, los bounded contexts definen límites en los cuales un modelo tiene significado y coherencia, estableciendo una frontera clara para evitar ambigüedades conceptuales.

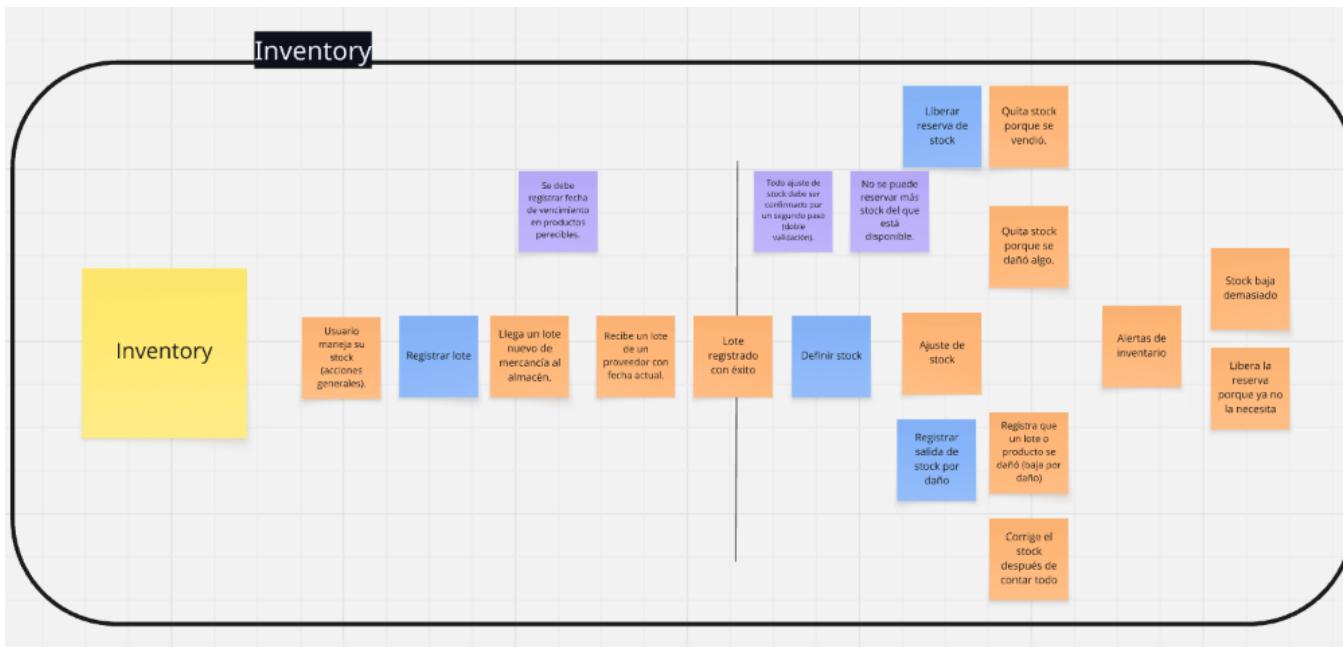
Bounded Context: IAM



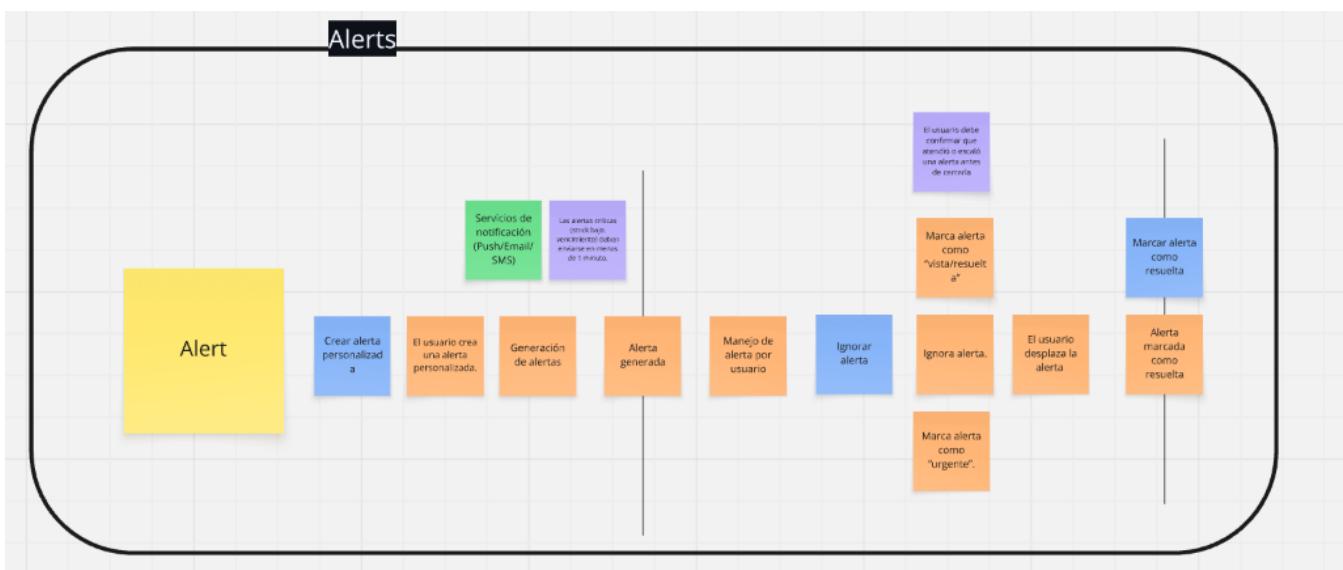
Bounded Context: Product catalog



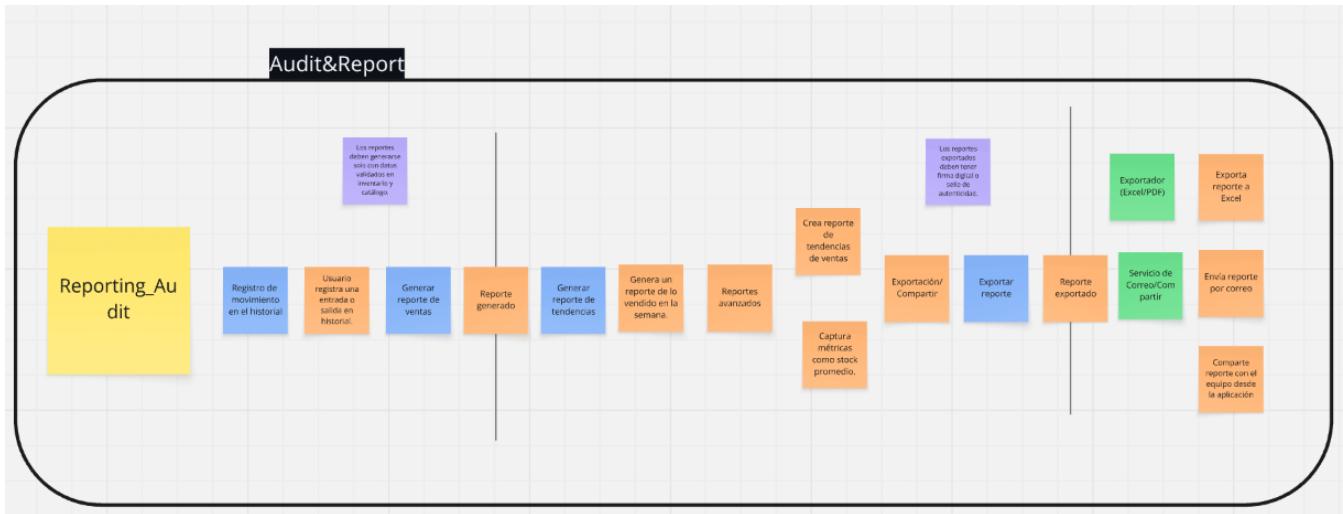
Bounded Context: Inventory



Bounded Context: Alerts

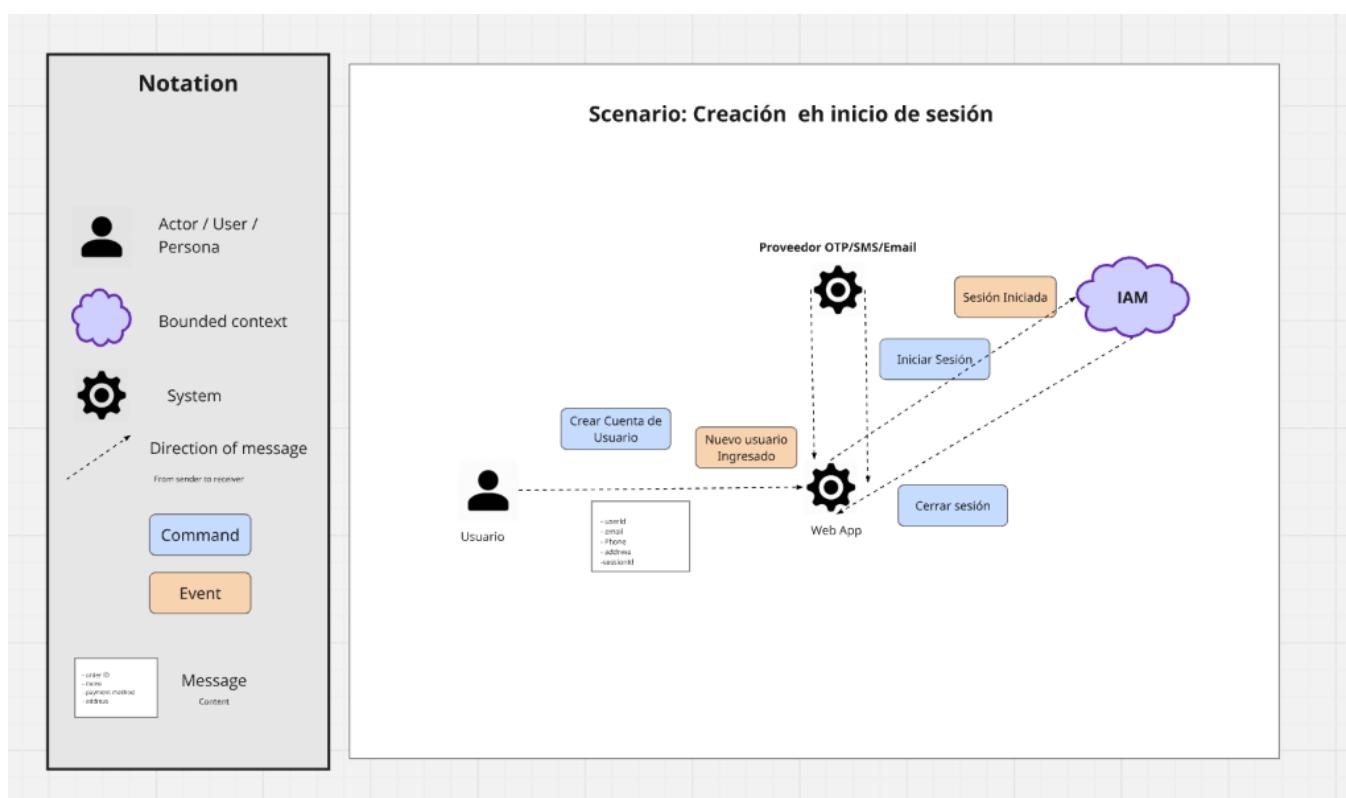


Bounded Context: AuditReport

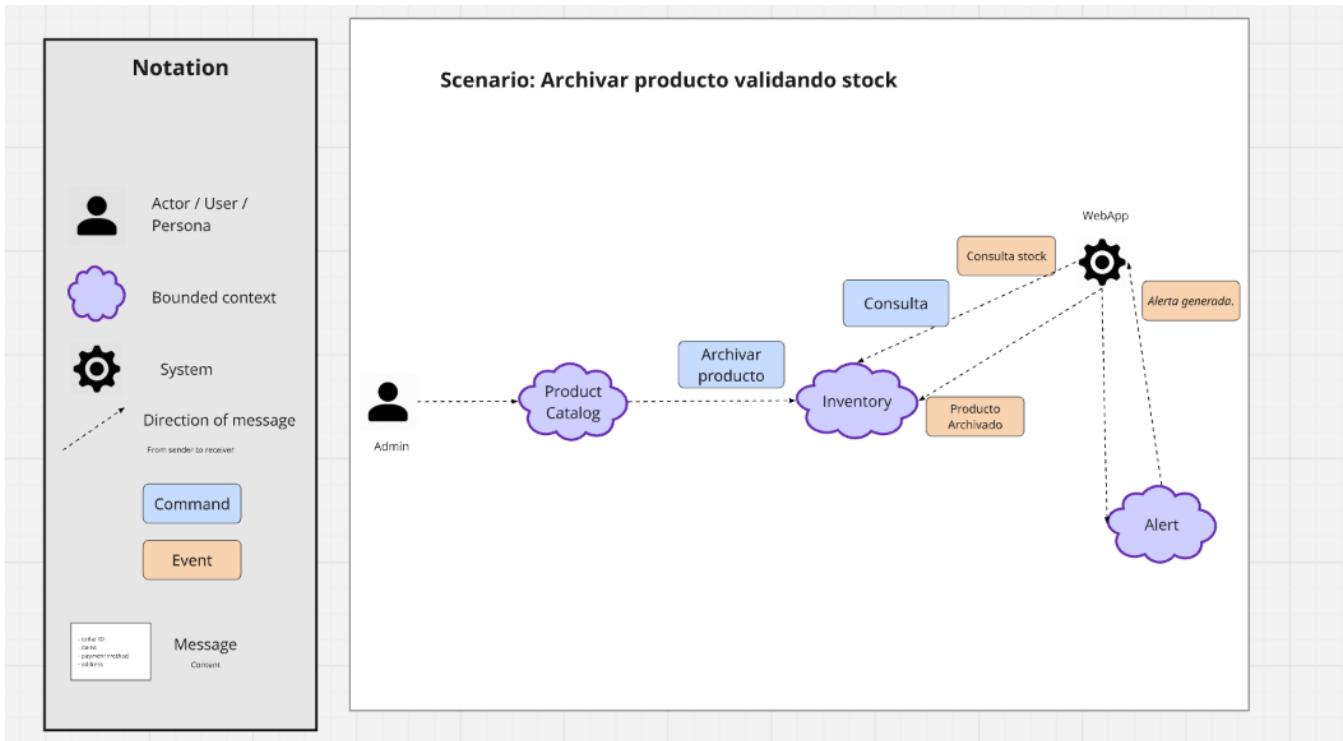


2.5.1.2. Domain Message Flows Modeling

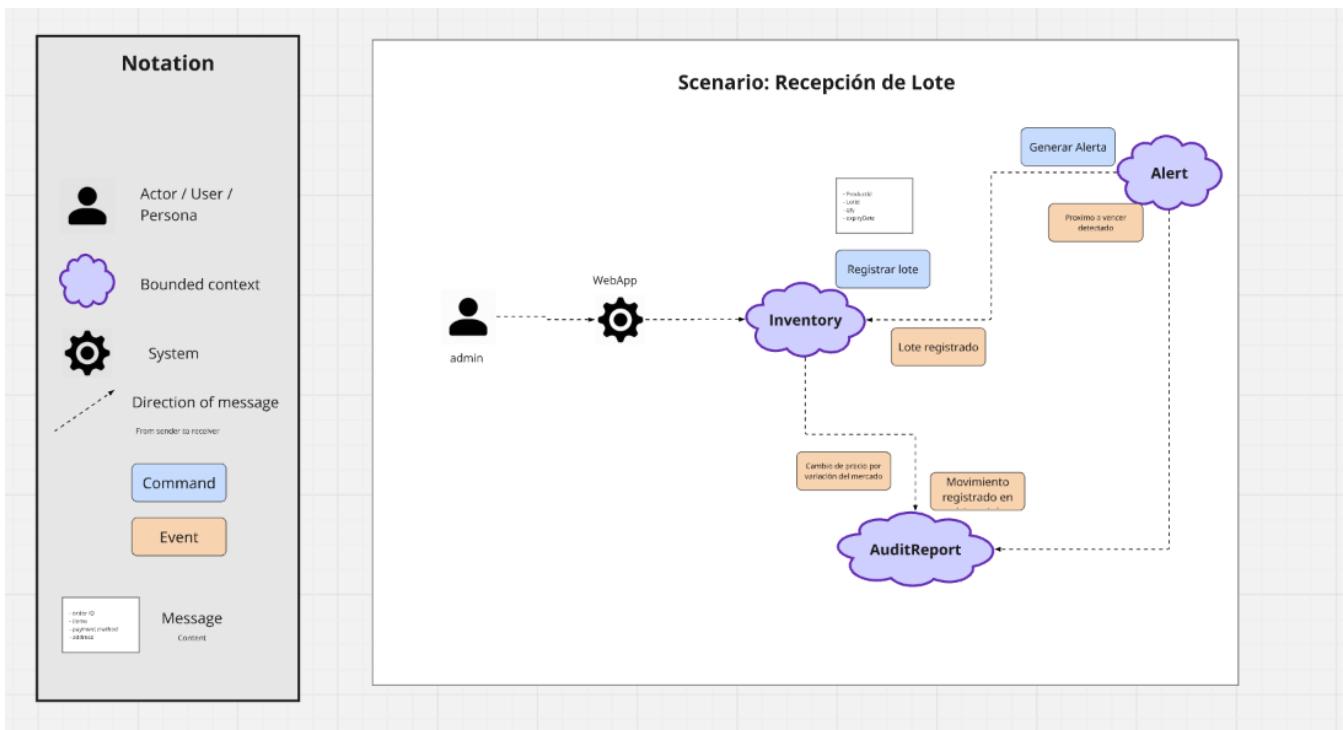
En los siguientes gráficos, se representa el proceso seguido para visualizar cómo deben colaborar los bounded contexts para resolver distintos escenarios que pueden presentarse en el negocio.



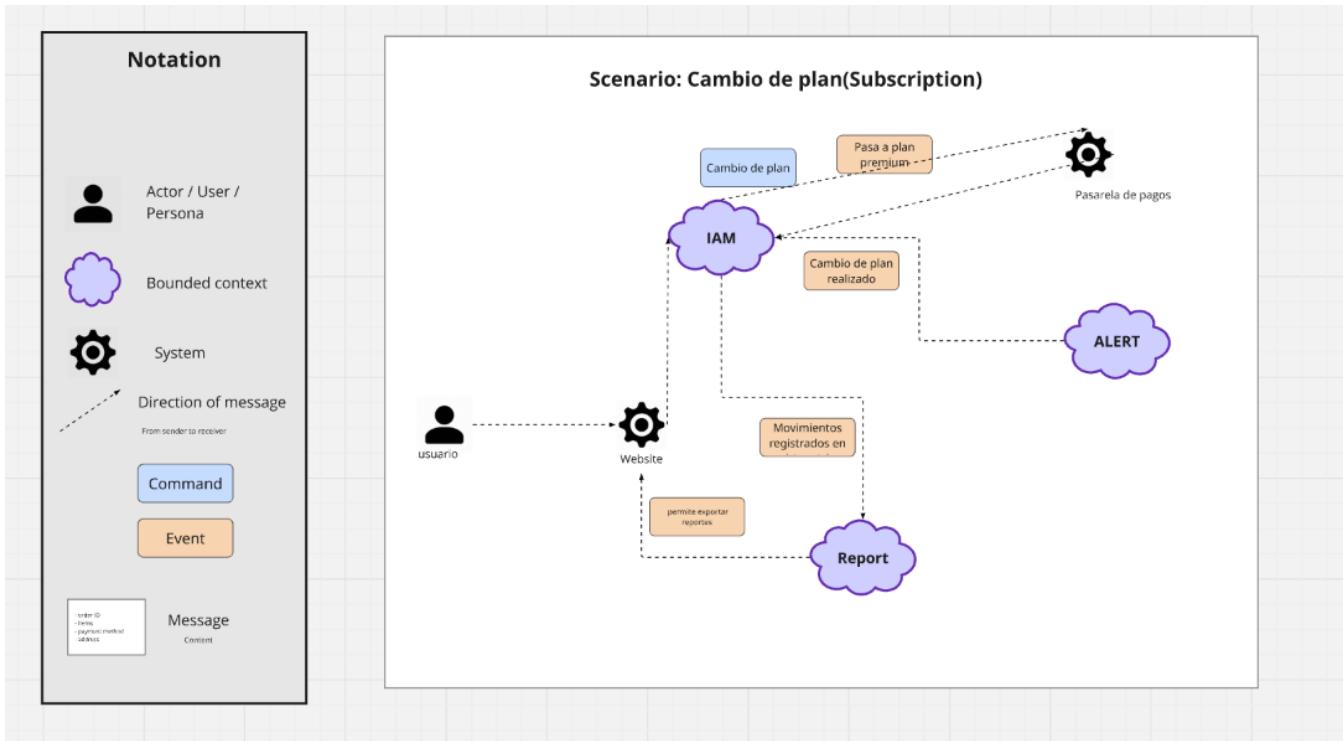
Este es un escenario de creación e inicio/cierre de sesión para nuestra app de bodega: primero, el usuario registra su cuenta en la Web App enviando su correo o teléfono; la aplicación remite el comando a IAM, que coordina con el proveedor OTP/SMS/Email para verificar el contacto y, una vez confirmado, publica el evento “Nuevo usuario ingresado” dejando la cuenta activa. A continuación, el usuario solicita iniciar sesión; la Web App envía el comando a IAM, que valida credenciales/OTP y emite “Sesión iniciada”, habilitando el acceso a las demás funciones. Finalmente, cuando el usuario decide salir, la Web App manda “Cerrar sesión” a IAM, que invalida el token/sesión y concluye el ciclo de autenticación.



Este es un escenario de archivado de producto con validación de stock: el Admin solicita en Product Catalog el comando Archivar producto y, antes de ejecutar el cambio, el catálogo consulta a Inventory el estado de existencias y reservas. Con la Consulta de stock, Inventory responde si hay unidades o reservas activas; si aún existe stock o reservas, desde la Web App se genera la notificación correspondiente y se envía a Alerts (Alerta generada) para informar y bloquear la acción. Si el resultado confirma stock=0 y sin reservas, el catálogo procede a completar la operación y se publica el evento Producto archivado, quedando el ítem fuera de publicación.



Este es un escenario de recepción de lote: el admin, a través de la Web App, envía a Inventory el comando Registrar lote con productId, lotId, qty y expiryDate; Inventory valida la política de perecibles (exigir fecha de vencimiento) y confirma la alta con el evento Lote registrado; a partir de ese hecho aplica sus reglas: si el lote es perecible y cae dentro del umbral, detecta "Próximo a vencer" y ordena en Alerts el comando Generar Alerta para notificar al usuario; en paralelo, registra la trazabilidad de la operación en AuditReport mediante el evento Movimiento registrado en historial, dejando el ingreso listo para reportes e indicadores.

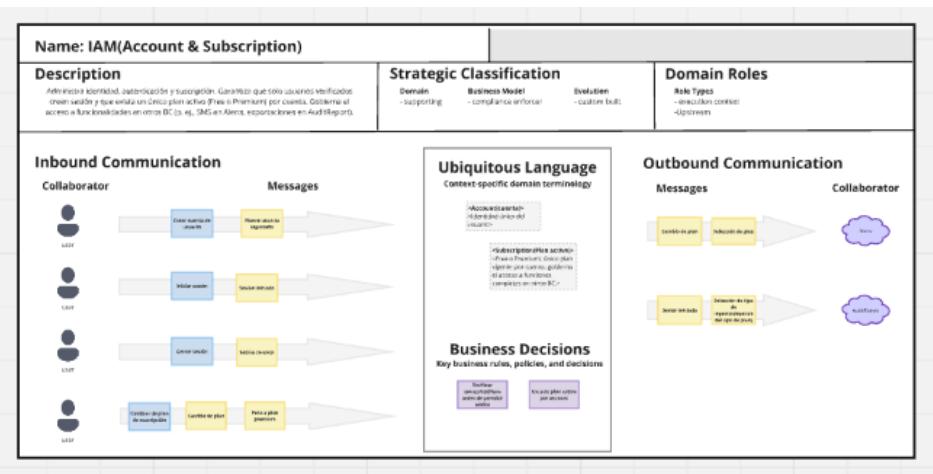


Este es un escenario de cambio de plan (Subscription): el usuario solicita en el Website el comando Cambio de plan, que es atendido por el BC IAM; IAM coordina con la pasarela de pagos para capturar el cobro y, una vez confirmado, publica los eventos “Cambio de plan realizado” y “Pasa a plan premium”, los cuales habilitan capacidades en los demás contextos: Alerts reconfigura sus canales (por ejemplo, activa SMS para notificaciones críticas) y Report registra el movimiento de suscripción en la auditoría; como efecto visible, el Website recibe el permiso de exportar reportes (gating por plan) y el usuario continúa operando con las nuevas funcionalidades asociadas a su plan activo..

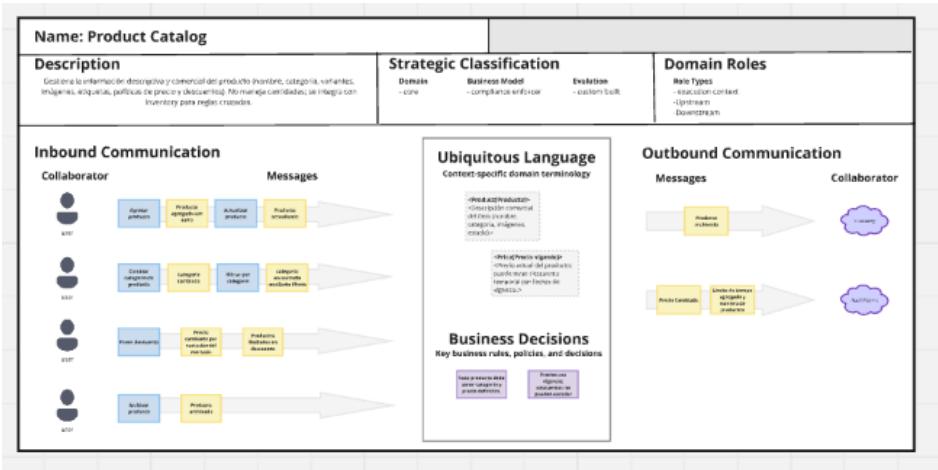
2.5.1.3. Bounded Context Canvases

El Bounded Context Canvas es una herramienta visual del Domain-Driven Design (DDD) que permite definir, entender y comunicar de forma clara los límites, responsabilidades y elementos clave de un Bounded Context. Ayuda a alinear al equipo en torno a una visión compartida del dominio, identificando entidades, eventos, comandos y relaciones externas. Además, elaborar un diseño modular del sistema de manera coherente es posible gracias a las nomenclaturas establecidas por un Bounded Context Canvas

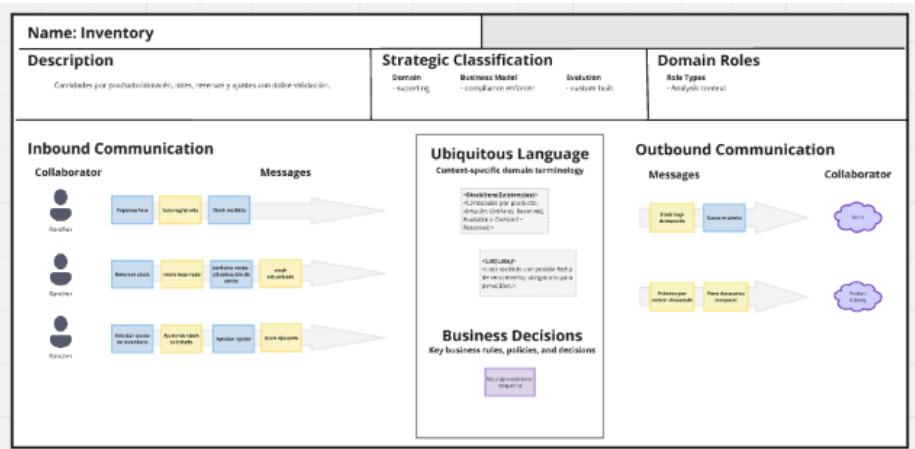
Bounded Context: IAM



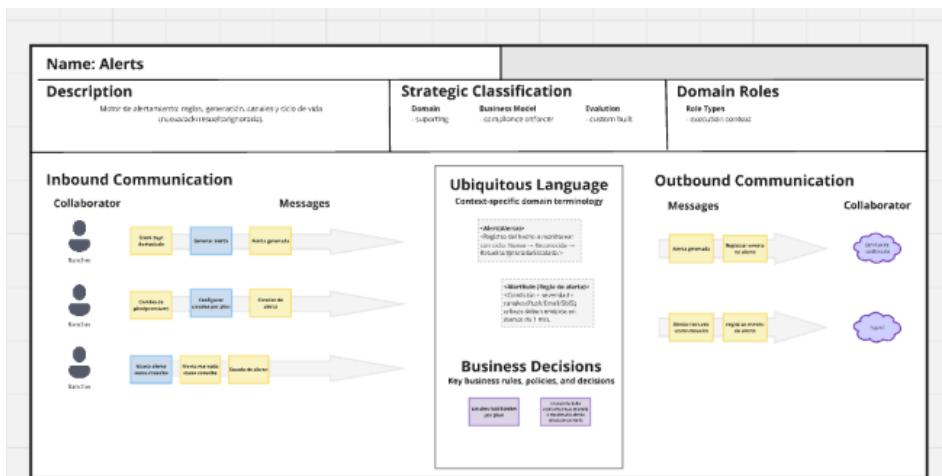
Bounded Context: Product catalog



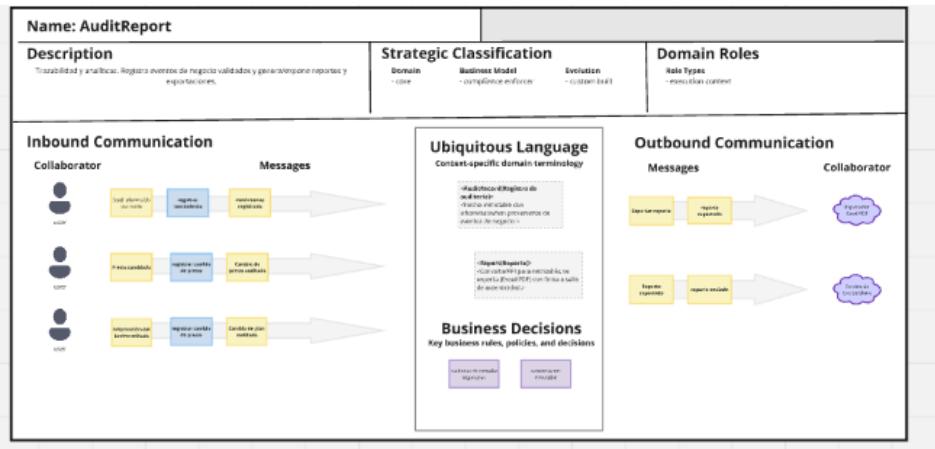
Bounded Context: Inventory



Bounded Context: Alerts



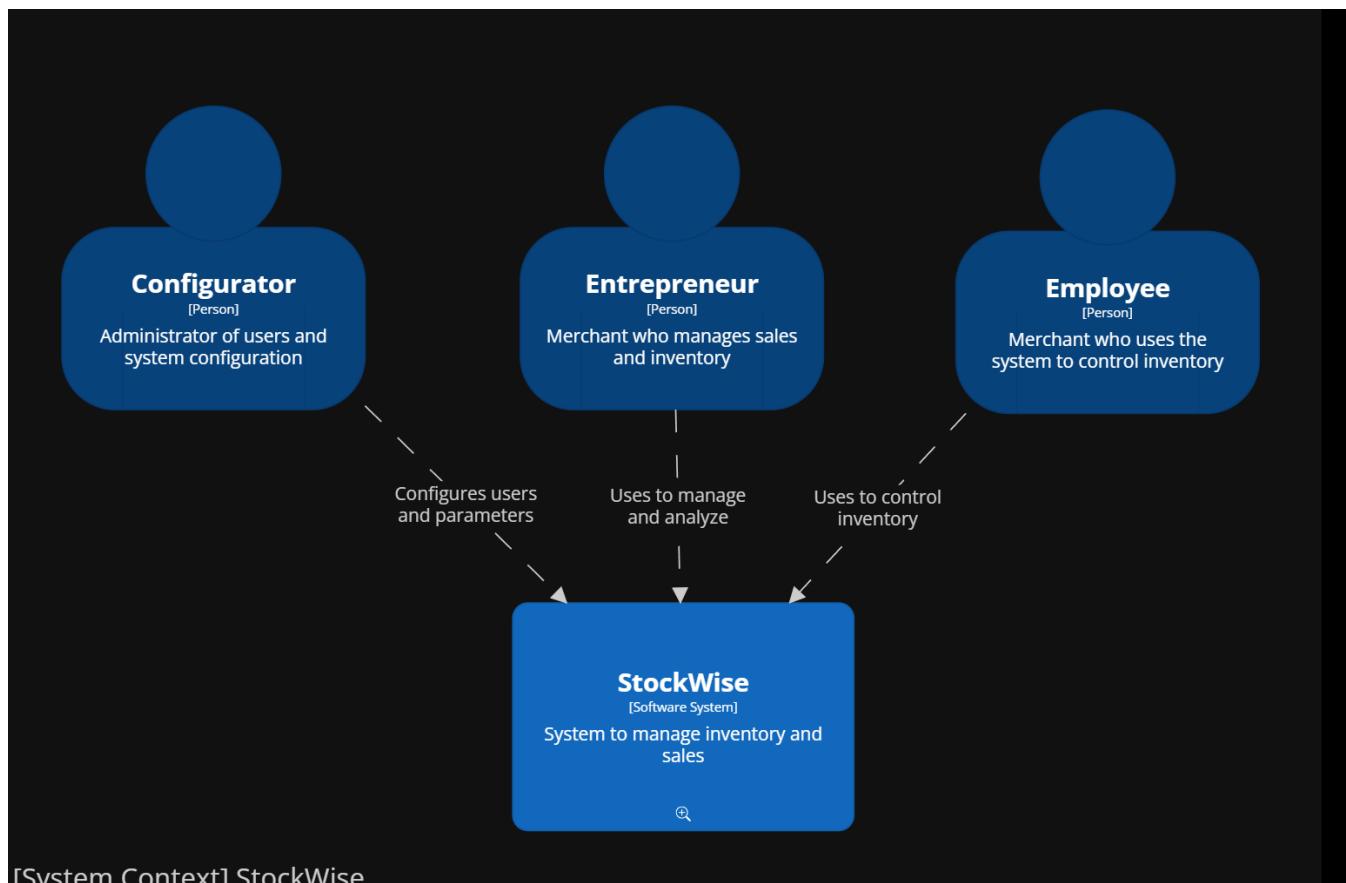
Bounded Context: AuditReport



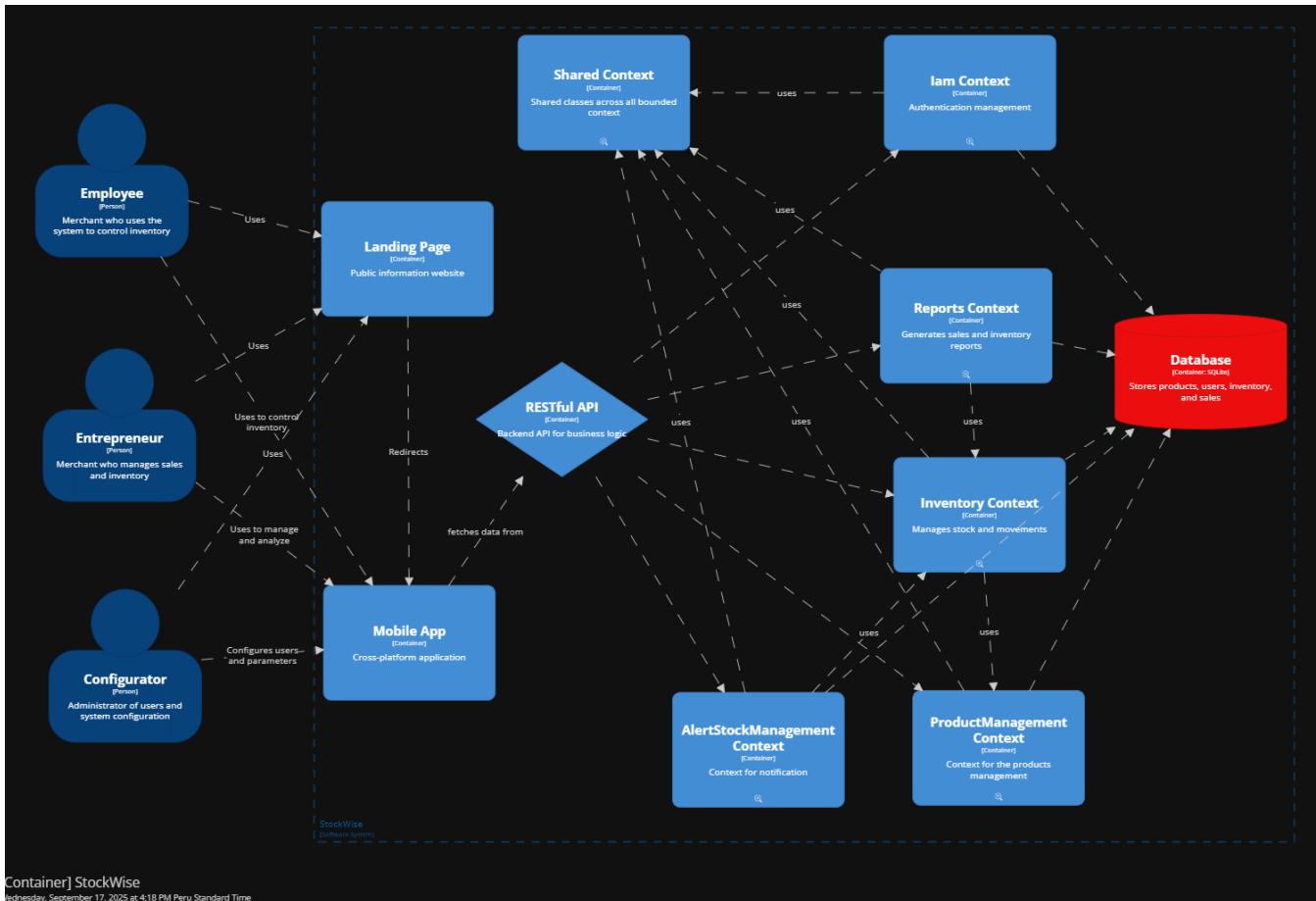
2.5.2. Context Mapping

2.5.3. Software Architecture

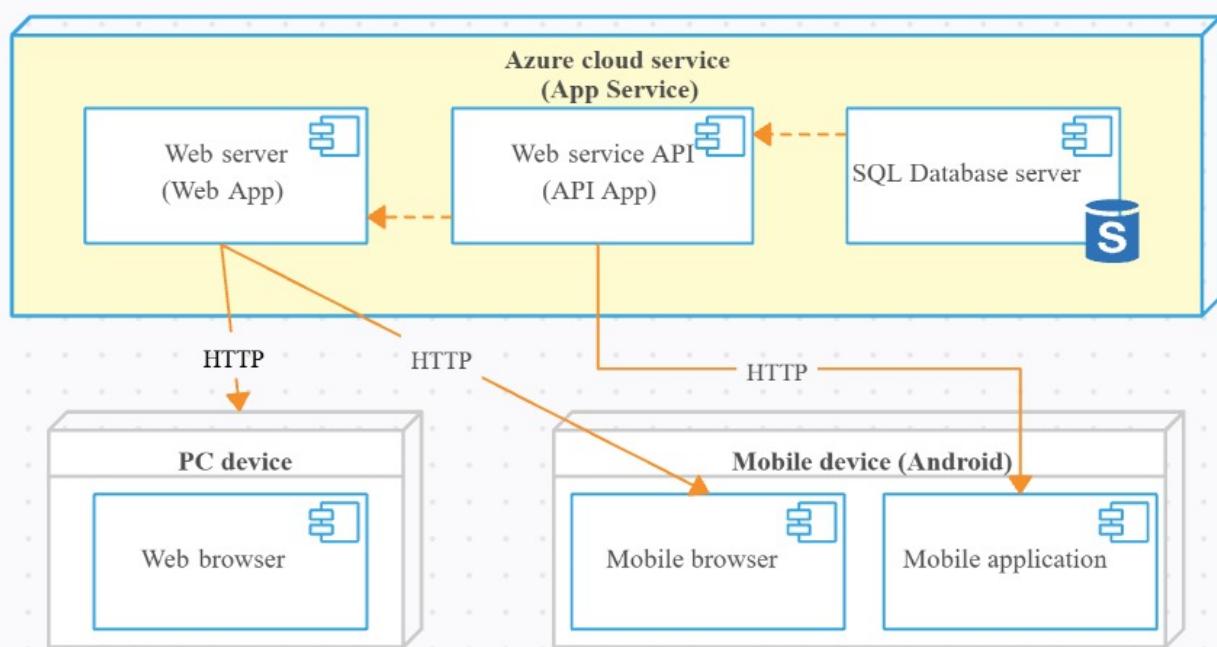
2.5.3.1. Software Architecture Context Level Diagrams



2.5.3.2. Software Architecture Container Level Diagrams



2.5.3.3. Software Architecture Deployment Diagrams



2.6. Tactical-Level Domain-Driven Design

2.6.1. Bounded Context: AlertStockManagement Context

2.6.1.1. Domain Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
----------------	------	-----------	---------------------

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
StockAlertItem	Aggregate Root (Entity)	Representa una alerta de stock para un producto	ProductName, Quantity, MinStock, EntryDate, IsLowStock
StockAlertQuery	Query	Criterio de consulta de alertas de stock	IncludeLowStock
IStockAlertQueryService	Domain Service (Interface)	Define contrato para consultas de alertas de stock	GetAlertsAsync(StockAlertQuery): Task<IEnumerable>
IInventoryReadRepository	Repository (Interface)	Lectura de inventario para generar alertas	GetStockAlertsAsync(StockAlertQuery): Task<List>
IStockAlertReadRepository	Repository (Interface)	Lectura de alertas de stock ya calculadas	GetStockAlertsAsync(StockAlertQuery): Task<List>

2.6.1.2. Application Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
StockAlertQueryService	Application Service	Implementa consultas de alertas de stock contra inventario	repository, GetAlertsAsync(StockAlertQuery): Task<List>
IStockAlertQueryService	Application Service (Interface)	Contrato de aplicación para consultas de alertas de stock	GetAlertsAsync(StockAlertQuery): Task<List>

2.6.1.3. Interface Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
StockAlertResource	Resource (DTO)	Representa alerta de stock en respuestas REST	ProductName, Quantity, MinStock, EntryDate, IsLowStock
StockAlertTransform	Assembler	Convierte entidad StockAlertItem en DTO	ToResource(StockAlertItem): StockAlertResource
StockAlertController	REST Controller	Expone endpoints para consultar alertas de stock	_service, GetAlerts(): Task

2.6.1.4. Infrastructure Layer

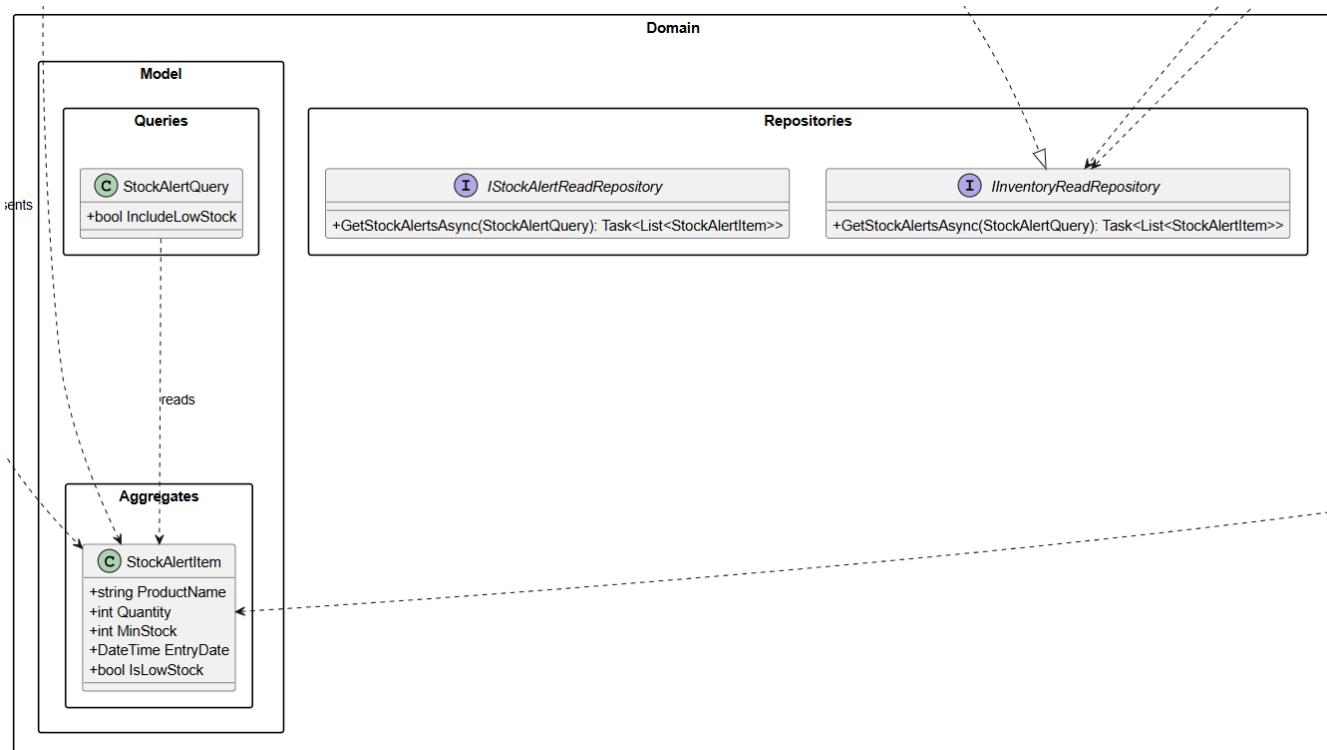
Clase	Tipo	Propósito	Atributos / Métodos
InventoryReadRepository	Repository Implementation	Implementación EF Core para leer stock y generar alertas	context, GetStockAlertsAsync(StockAlertQuery): Task<List>

2.6.1.5. Bounded Context Software Architecture Component Level Diagrams



2.6.1.6. Bounded Context Software Architecture Code Level Diagrams

2.6.1.6.1. Bounded Context Domain Layer Class Diagrams



2.6.2. Bounded Context: Iam Context

2.6.2.1. Domain Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
User	Aggregate Root (Entity)	Representa un usuario dentro del sistema IAM	Id, Name, LastName, Email, Password, Role, User(), User(SignInCommand), User(SignUpCommand)

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
UserRole	Enum (Value Object)	Define los roles posibles de usuario	Employee, Administrator
SignUpCommand	Command	Solicitud para registrar un nuevo usuario	Name, LastName, Email, Password
SignInCommand	Command	Solicitud para iniciar sesión	Email, Password
ChangeUserRoleCommand	Command	Solicitud para cambiar el rol de un usuario	UserId, NewRole
 GetUserByIdQuery	Query	Consulta para obtener un usuario por ID	Id
IUserCommandService	Domain Service (Interface)	Define el contrato de servicios para manejar comandos de usuario	Handle(SignUpCommand), Handle(SignInCommand), Handle(ChangeUserRoleCommand)
IUserQueryService	Domain Service (Interface)	Define el contrato de servicios para consultas de usuario	Handle(GetUserByIdQuery)
IUserRepository	Repository (Interface)	Acceso abstracto a la persistencia de usuarios	FindByEmailAsync, FindByIdAsync, AddAsync, UpdateUserRoleAsync

2.6.2.2. Interface Layer

Clase	Tipo	Propósito	Atributos / Métodos
SignUpResource	Resource (DTO)	Representa los datos de entrada para registrarse	Name, LastName, Email, Password
SignInResource	Resource (DTO)	Representa los datos de entrada para iniciar sesión	Email, Password
UserResource	Resource (DTO)	Representa un usuario simplificado en respuesta	Id, Name, Email
AuthenticatedUserResource	Resource (DTO)	Representa un usuario autenticado junto con token	Id, Name, LastName, Token, Role
ChangeUserRoleResource	Resource (DTO)	Datos de entrada para cambiar el rol de un usuario	UserId, NewRole
SignUpCommandFromResourceAssembler	Assembler	Convierte un recurso en un comando de dominio	ToCommandFromResource(SignUpResource)
SignInCommandFromResourceAssembler	Assembler	Convierte un recurso en un comando de dominio	ToCommandFromResource(SignInResource)
UserResourceFromEntityAssembler	Assembler	Convierte una entidad en un recurso DTO	ToResourceFromEntity(User)

Clase	Tipo	Propósito	Atributos / Métodos
ChangeUserRoleCommandFromResourceAssembler	Assembler	Convierte un recurso en un comando de dominio	ToCommandFromResource(ChangeUserRoleResource)
AuthenticatedUserResourceFromEntityAssembler	Assembler	Convierte entidad y token en recurso de autenticación	ToResourceFromEntity(User, Token)
AuthenticationController	REST Controller	Expone endpoints de autenticación y gestión de usuario	SignUp(SignUpResource), SignIn(SignInResource), ChangeUserRole(ChangeUserRoleResource)

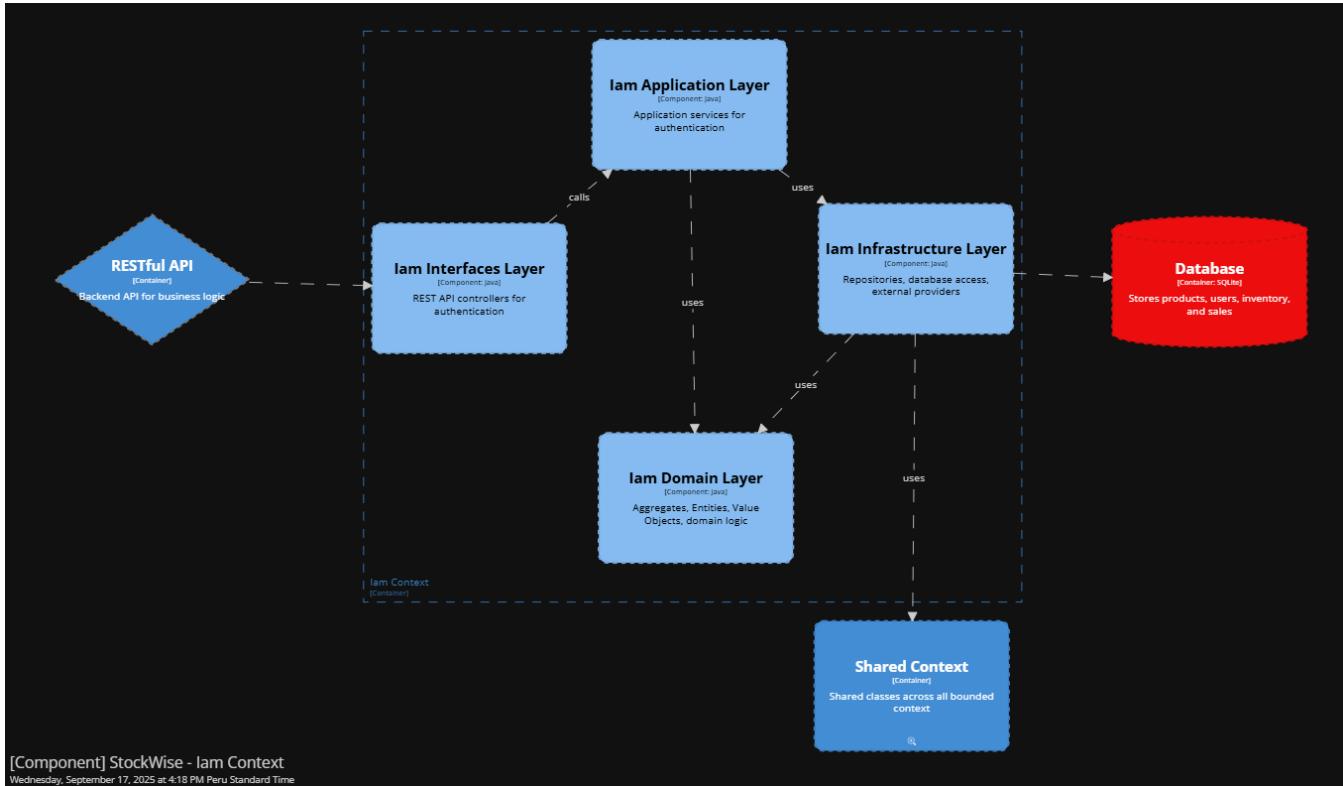
2.6.2.3. Application Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
UserCommandService	Application Service	Implementa los servicios de comando de usuarios	userRepository, tokenService, hashingService, unitOfWork, Handle(SignUpCommand), Handle(SignInCommand), Handle(ChangeUserRoleCommand)
UserQueryService	Application Service	Implementa los servicios de consulta de usuarios	userRepository, GetAllAsync(), FindByIdAsync(int)
ITokenService	Outbound Service (Interface)	Define generación y validación de tokens	GenerateToken(User), ValidateToken(string)
IHashingService	Outbound Service (Interface)	Define operaciones de hashing de contraseñas	GenerateHash(string), VerifyHash(string, string)

2.6.2.4. Infrastructure Layer

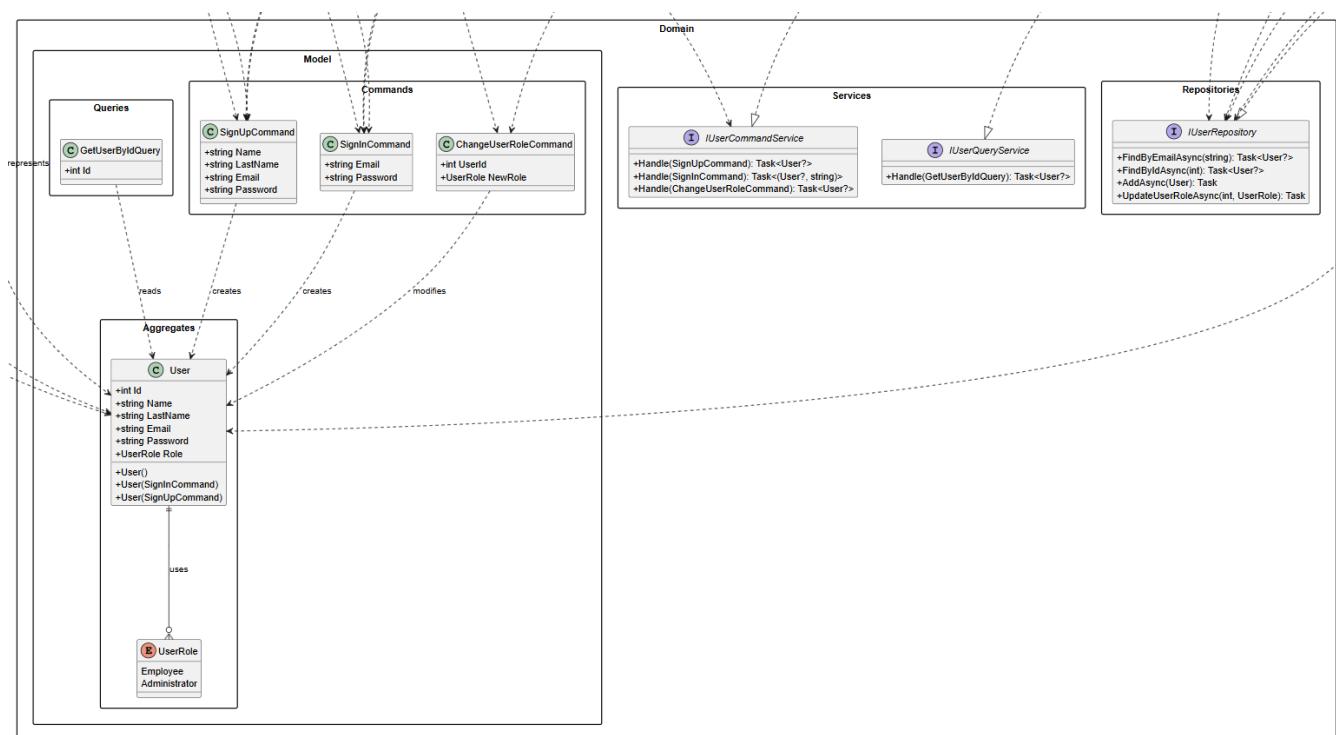
Clase	Tipo	Propósito	Atributos / Métodos
UserRepository	Repository (Implementation)	Implementación de acceso a datos de usuarios	context, FindByEmailAsync, FindByIdAsync, AddAsync, UpdateUserRoleAsync
TokenService	Outbound Service Implementation	Implementación de generación y validación de tokens JWT	GenerateToken(User), ValidateToken(string)
HashingService	Outbound Service Implementation	Implementación de hashing de contraseñas con BCrypt	GenerateHash(string), VerifyHash(string, string)
TokenSettings	Configuration	Configuración de secret key para JWT	Secret
AuthorizeRolesAttribute	Authorization Filter	Atributo para restringir acceso por roles en controllers	_roles, OnAuthorization(AuthorizationFilterContext)

2.6.2.5. Bounded Context Software Architecture Component Level Diagrams



2.6.2.6. Bounded Context Software Architecture Code Level Diagrams

2.6.2.6.1. Bounded Context Domain Layer Class Diagrams



2.6.2.6.2. Bounded Context Database Design Diagram

Users					
		comment			
Id	INT	NULL	default	comment	
Name	VARCHAR(255)	N-N	default	comment	
LastName	VARCHAR(255)	N-N	default	comment	
Email	VARCHAR(255)	N-N	default	comment	
Password	VARCHAR(255)	N-N	default	comment	
Role	ENUM(Employee,Administrator)	N-N	Administrator	comment	
CreatedAt	TIMESTAMP	NULL	CURRENT_TIMESTAMP	comment	
UpdatedAt	TIMESTAMP	NULL	CURRENT_TIMESTAMP	comment	

2.6.3. Bounded Context: Inventory Context

2.6.3.1. Domain Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
InventoryByProduct	Aggregate Root (Entity)	Representa inventario por producto	Id, Categoria, Producto, FechaEntrada, Cantidad, Precio, StockMinimo, UnidadMedida, InventoryByProduct(), InventoryByProduct(...)
InventoryByBatch	Aggregate Root (Entity)	Representa inventario por lotes	Id, Proveedor, Producto, FechaEntrada, Cantidad, Precio, Unidad, Total, InventoryByBatch(), InventoryByBatch(...)
Cantidad	Value Object	Representa una cantidad	Value, Cantidad(int)
Precio	Value Object	Representa un precio monetario	Value, Precio(decimal)
StockMinimo	Value Object	Define el stock mínimo de un producto	Value, StockMinimo(int)
Unidad	Value Object	Representa la unidad de medida	Value, Unidad(string)
CreateInventoryByProductCommand	Command	Crear inventario por producto	Categoría, Producto, FechaEntrada, Cantidad, Precio, StockMinimo, UnidadMedida
CreateInventoryByBatchCommand	Command	Crear inventario por lote	Proveedor, Producto, FechaEntrada, Cantidad, Precio, Unidad
GetInventoryByBatchQuery	Query	Consultar todos los inventarios por lote	-
GetInventoryByProductQuery	Query	Consultar todos los inventarios por producto	-
GetInventoryByProductIdQuery	Query	Consultar inventario por producto por Id	Id
GetInventoryByBatchByIdQuery	Query	Consultar inventario por lote por Id	Id
GetGeneralInventoryQuery	Query	Consultar inventario general	-
IInventoryByProductCommandService	Domain Service (Interface)	Define contrato para comandos de inventario por producto	Handle(CreateInventoryByProductCommand), DeleteAsync(int)
IInventoryByProductQueryService	Domain Service (Interface)	Define contrato para consultas de inventario por producto	GetAllAsync(), GetByldAsync(int)
IInventoryByBatchCommandService	Domain Service (Interface)	Define contrato para comandos de inventario por lote	Handle(CreateInventoryByBatchCommand), DeleteAsync(int)

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
IInventoryByBatchQueryService	Domain Service (Interface)	Define contrato para consultas de inventario por lote	Handle(GetInventoryByBatchQuery)
IInventoryByProductRepository	Repository (Interface)	Acceso abstracto a inventarios por producto	ListAsync(), FindByIdAsync(int), AddAsync(InventoryByProduct), DeleteAsync(int)
IInventoryByBatchRepository	Repository (Interface)	Acceso abstracto a inventarios por lote	ListAsync(), FindByIdAsync(int), AddAsync(InventoryByBatch), DeleteAsync(int)

2.6.3.2. Application Layer

Clase	Tipo	Propósito	Atributos / Métodos
InventoryByProductCommandService	Application Service	Implementa comandos de inventario por producto	repository, unitOfWork, Handle(CreateInventoryByProductCommand), DeleteAsync(int)
InventoryByBatchCommandService	Application Service	Implementa comandos de inventario por lote	repository, unitOfWork, Handle(CreateInventoryByBatchCommand), DeleteAsync(int)
InventoryByProductQueryService	Application Service	Implementa consultas de inventario por producto	repository, GetAllAsync(), GetByIdAsync(int)
InventoryByBatchQueryService	Application Service	Implementa consultas de inventario por lote	repository, Handle(GetInventoryByBatchQuery)

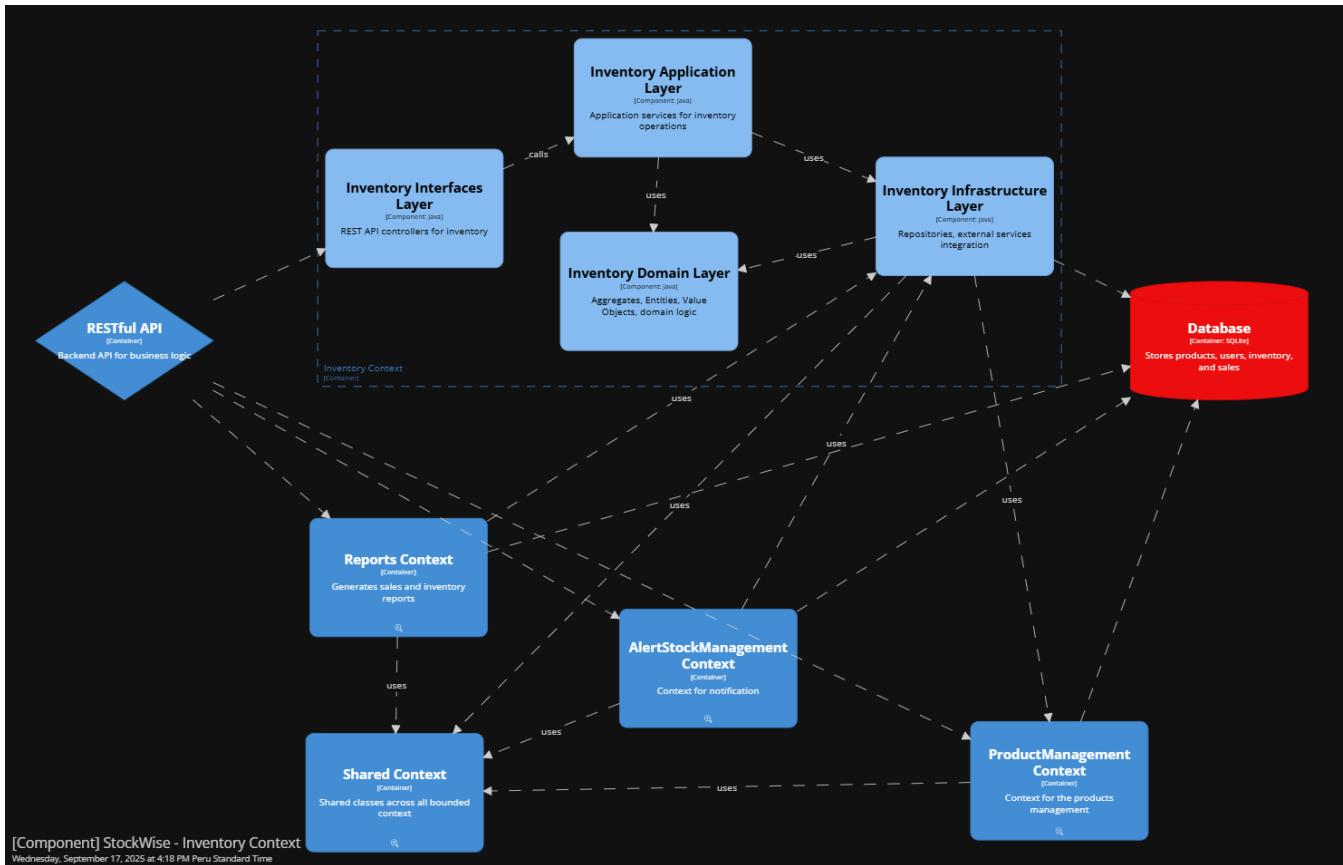
2.6.3.3. Interface Layer

Clase	Tipo	Propósito	Atributos / Métodos
InventoryGeneralResource	Resource (DTO)	Representa el inventario general (productos y lotes)	Productos, Lotes
InventoryByProductResource	Resource (DTO)	Representa inventario por producto en respuestas	Id, Categoria, Producto, FechaEntrada, Cantidad, Precio, StockMinimo, UnidadMedida
InventoryByBatchResource	Resource (DTO)	Representa inventario por lote en respuestas	Id, Proveedor, Producto, FechaEntrada, Cantidad, Precio, Unidad, Total
InventoryTransform	Assembler	Convierte entidades de inventario a recursos DTO	ToResourceFromEntity(InventoryByProduct), ToResourceFromEntity(InventoryByBatch)
InventoryController	REST Controller	Expone endpoints de inventarios	GetAllInventory()

2.6.3.4. Infrastructure Layer

Clase	Tipo	Propósito	Atributos / Métodos
InventoryByProductRepository	Repository Implementation	Implementación acceso a inventario por producto (EF Core)	context, ListAsync(), FindByIdAsync(int), AddAsync(InventoryByProduct), DeleteAsync(int)
InventoryByBatchRepository	Repository Implementation	Implementación acceso a inventario por lote (EF Core)	context, ListAsync(), FindByIdAsync(int), AddAsync(InventoryByBatch), DeleteAsync(int)

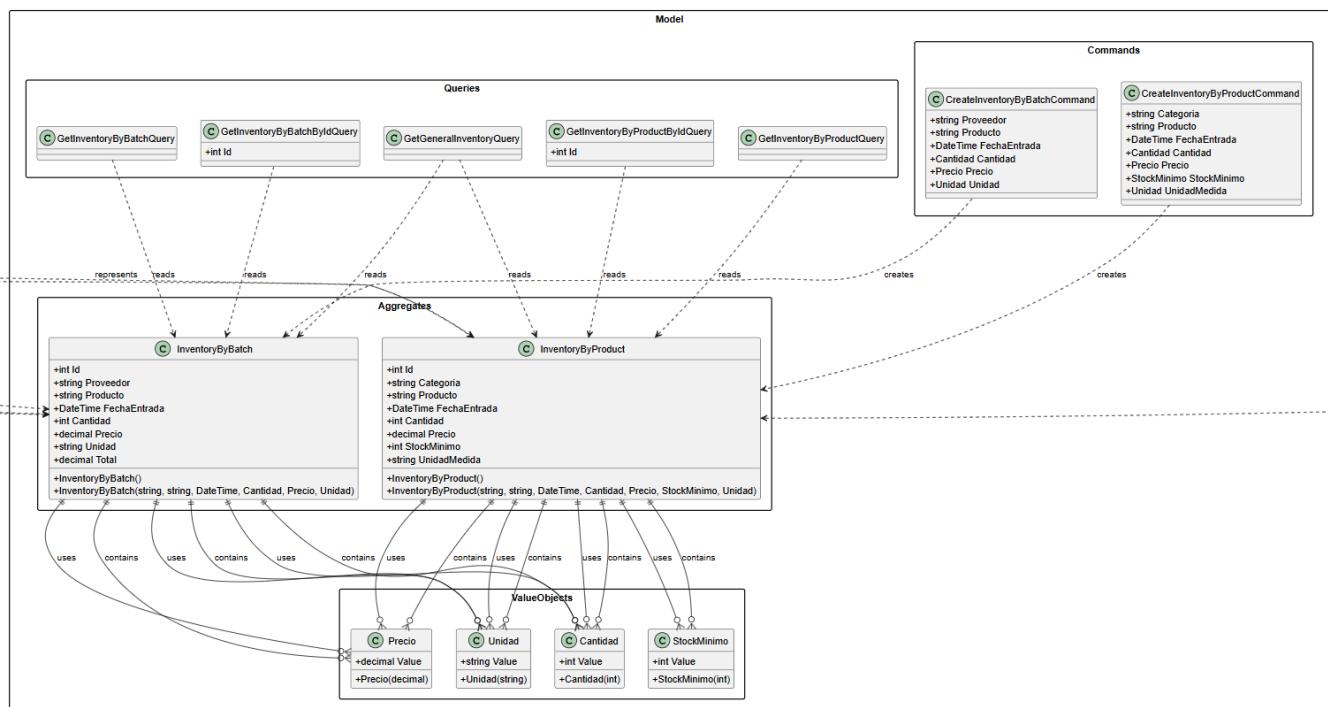
2.6.3.5. Bounded Context Software Architecture Component Level Diagrams



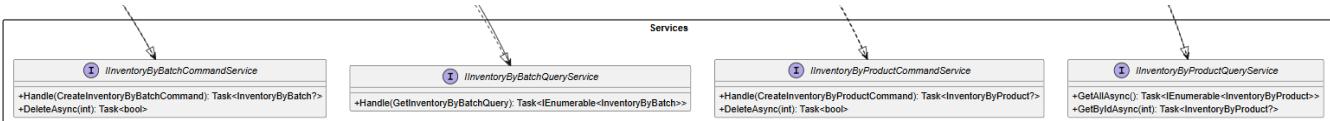
2.6.3.6. Bounded Context Software Architecture Code Level Diagrams

2.6.3.6.1. Bounded Context Domain Layer Class Diagrams

Domain: Model



Domain: Services



2.6.3.6.2. Bounded Context Database Design Diagram



2.6.4.1. Domain Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
Product	Aggregate Root (Entity)	Representa un producto dentro del sistema	Id, Name, Description, PurchasePrice, SalePrice, InternalNotes, CategoryId, UnitId, Category, Unit, ProductTags, Product(), Product(...), Product(CreateProductCommand), UpdateProduct(...), AddTag(int), RemoveTag(int)
Category	Entity	Agrupa productos por categoría	Id, Name, Products, Category(), Category(string), UpdateName(string)
Unit	Entity	Representa la unidad de medida de un producto	Id, Name, Abbreviation, Products, Unit(), Unit(string, string), UpdateUnit(string, string)
ProductTag	Entity (Join)	Relaciona productos con etiquetas	ProductId, TagId, ProductTag(int, int)
Tag	Entity	Representa una etiqueta asociada a productos	Id, Name, Description, Tag(), Tag(string, string), UpdateTag(string, string)
Combo	Entity	Representa un combo de productos	Id, Name, Description, Price, ComboItems, Combo(), Combo(...), UpdateCombo(...)
ComboItem	Entity	Relaciona un producto con un combo	Id, ComboId, ProductId, Quantity, Combo, Product, ComboItem(), ComboItem(int, int, int), UpdateQuantity(int)
ProductId	Value Object	Identificador de producto	Value
UnitId	Value Object	Identificador de unidad	Value
TagId	Value Object	Identificador de etiqueta	Value
CategoryId	Value Object	Identificador de categoría	Value

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
CreateProductCommand	Command	Crear un nuevo producto	Name, Description, PurchasePrice, SalePrice, InternalNotes, CategoryId, UnitId, TagIds
UpdateProductCommand	Command	Actualizar datos de un producto	Id, Name, Description, PurchasePrice, SalePrice, InternalNotes, CategoryId, UnitId
DeleteProductCommand	Command	Eliminar un producto	Id
CreateComboCommand	Command	Crear un combo	Name, Description, Price
UpdateComboCommand	Command	Actualizar datos de un combo	Id, Name, Description, Price
CreateComboItemCommand	Command	Crear ítem en un combo	Comboid, ProductId, Quantity
UpdateComboItemCommand	Command	Actualizar ítem de un combo	Id, Quantity
DeleteComboItemCommand	Command	Eliminar ítem de un combo	Id
DeleteComboCommand	Command	Eliminar un combo	Id
GetProductByIdQuery	Query	Consultar producto por Id	Id
GetAllProductsQuery	Query	Consultar todos los productos	-
GetProductsByCategoryQuery	Query	Consultar productos por categoría	CategoryId
GetProductsByTagQuery	Query	Consultar productos por etiqueta	TagId
GetProductsByPriceRangeQuery	Query	Consultar productos por rango de precios	MinPrice, MaxPrice
GetComboByIdQuery	Query	Consultar combo por Id	Id
GetAllCombosQuery	Query	Consultar todos los combos	-
GetComboItemByIdQuery	Query	Consultar ítem de combo por Id	Id
GetComboItemsByComboidQuery	Query	Consultar ítems de un combo	Comboid
IProductCommandService	Domain Service (Interface)	Define el contrato para manejar comandos de productos	Handle(CreateProductCommand), Handle(UpdateProductCommand), Handle>DeleteProductCommand
IProductQueryService	Domain Service (Interface)	Define el contrato para consultas de productos	Handle(GetProductByIdQuery), Handle(GetAllProductsQuery), Handle(GetProductsByCategoryQuery), Handle(GetProductsByTagQuery), Handle(GetProductsByPriceRangeQuery)
IComboCommandService	Domain Service (Interface)	Define el contrato para comandos de combos	Handle(CreateComboCommand), Handle(UpdateComboCommand)

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
IComboQueryService	Domain Service (Interface)	Contrato para consultas de combos	GetAllAsync()
IUnitQueryService	Domain Service (Interface)	Contrato para consultas de unidades	GetAllAsync()
ITagQueryService	Domain Service (Interface)	Contrato para consultas de etiquetas	GetAllAsync()
IProductRepository	Repository (Interface)	Acceso abstracto a productos	FindProductByNameAsync, FindAllProductsAsync, FindProductsByCategoryAsync, FindProductsByTagAsync, FindProductsByPriceRangeAsync, FindProductWithRelationsAsync
ICategoryRepository	Repository (Interface)	Acceso abstracto a categorías	FindCategoryByNameAsync, FindAllCategoriesAsync
IUnitRepository	Repository (Interface)	Acceso abstracto a unidades	FindUnitByNameAsync, FindAllUnitsAsync
ITagRepository	Repository (Interface)	Acceso abstracto a etiquetas	FindTagByNameAsync, FindAllTagsAsync, FindTagsByIdsAsync
IComboRepository	Repository (Interface)	Acceso abstracto a combos	FindComboByNameAsync, FindAllCombosAsync

2.6.4.2. Application Layer

Clase	Tipo	Propósito	Atributos / Métodos
ProductCommandService	Application Service	Implementa comandos de producto	productRepository, tagRepository, unitOfWork, Handle(CreateProductCommand), Handle(UpdateProductCommand)
ProductQueryService	Application Service	Implementa consultas de productos	productRepository, Handle(GetProductByIdQuery), Handle(GetAllProductsQuery), Handle(GetProductsByCategoryQuery), Handle(GetProductsByTagQuery), Handle(GetProductsByPriceRangeQuery)
ComboCommandService	Application Service	Implementa comandos de combos	comboRepository, unitOfWork, Handle(CreateComboCommand), Handle(UpdateComboCommand)
ComboQueryService	Application Service	Consultas sobre combos	comboRepository, GetAllAsync()
UnitQueryService	Application Service	Consultas sobre unidades	unitRepository, GetAllAsync()
TagQueryService	Application Service	Consultas sobre etiquetas	tagRepository, GetAllAsync()

2.6.4.3. Interface Layer

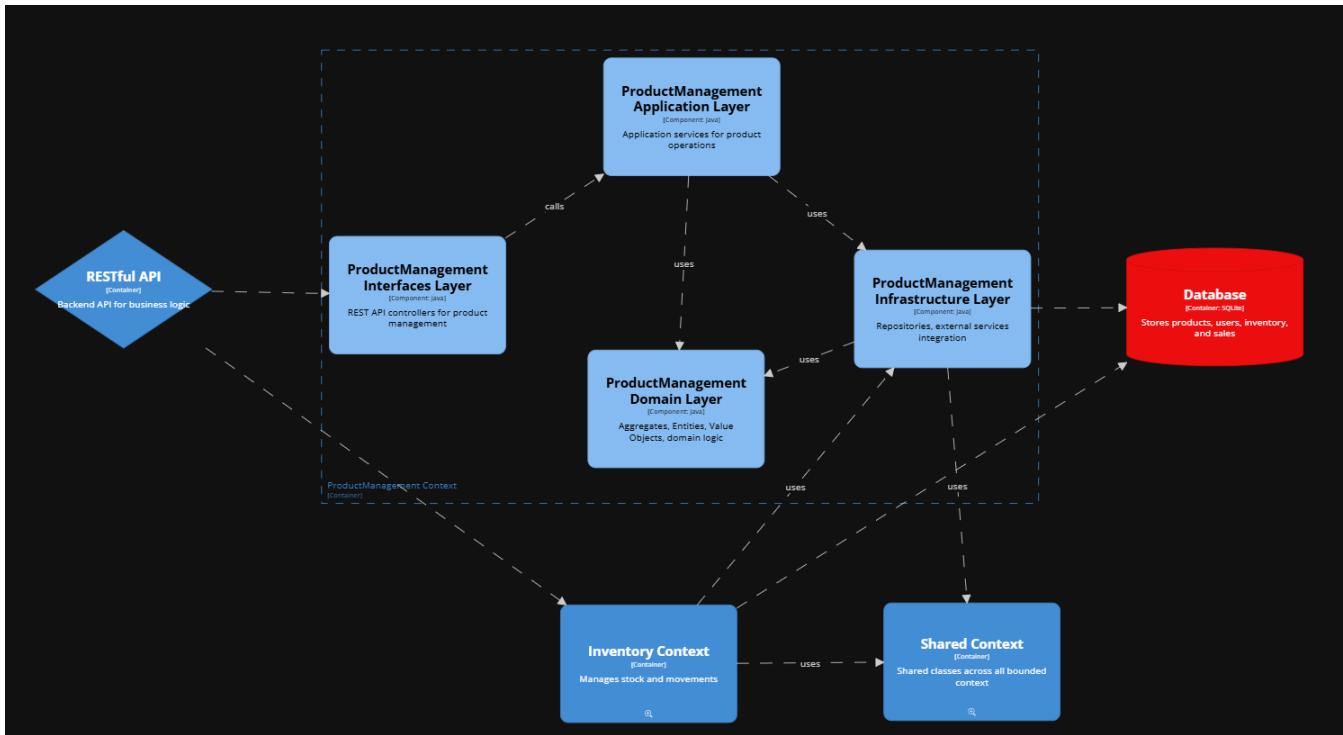
Clase	Tipo	Propósito	Atributos / Métodos
CreateProductResource	Resource (DTO)	Datos de entrada para crear producto	Name, Description, PurchasePrice, SalePrice, InternalNotes, CategoryId, UnitId, TagIds
UpdateProductResource	Resource (DTO)	Datos de entrada para actualizar producto	Id, Name, Description, PurchasePrice, SalePrice, InternalNotes, CategoryId, UnitId

Clase	Tipo	Propósito	Atributos / Métodos
ProductResource	Resource (DTO)	Representa un producto completo en respuesta	Id, Name, Description, PurchasePrice, SalePrice, InternalNotes, CategoryId, UnitId, Category, Unit, Tags
CategoryResource	Resource (DTO)	Representa una categoría	Id, Name
UnitResource	Resource (DTO)	Representa una unidad	Id, Name, Abbreviation
TagResource	Resource (DTO)	Representa una etiqueta	Id, Name, Description
ComboResource	Resource (DTO)	Representa un combo	Id, Name, Description, Price
CreateComboResource	Resource (DTO)	Datos de entrada para crear combo	Name, Description, Price
CreateProductCommandFromResourceAssembler	Assembler	Convierte recurso a comando	ToCommandFromResource(CreateProductResource)
UpdateProductCommandFromResourceAssembler	Assembler	Convierte recurso a comando	ToCommandFromResource(UpdateProductResource)
ProductResourceFromEntityAssembler	Assembler	Convierte entidad a recurso DTO	ToResourceFromEntity(Product)
ProductsController	REST Controller	Expone endpoints de productos	GetProductById, CreateProduct, GetAllProducts, GetProductsByCategory, GetProductsByTag
UnitsController	REST Controller	Expone endpoints de unidades	GetAllUnits
TagsController	REST Controller	Expone endpoints de etiquetas	GetAllTags
CombosController	REST Controller	Expone endpoints de combos	GetComboById, CreateCombo, GetAllCombos

2.6.4.4. Infrastructure Layer

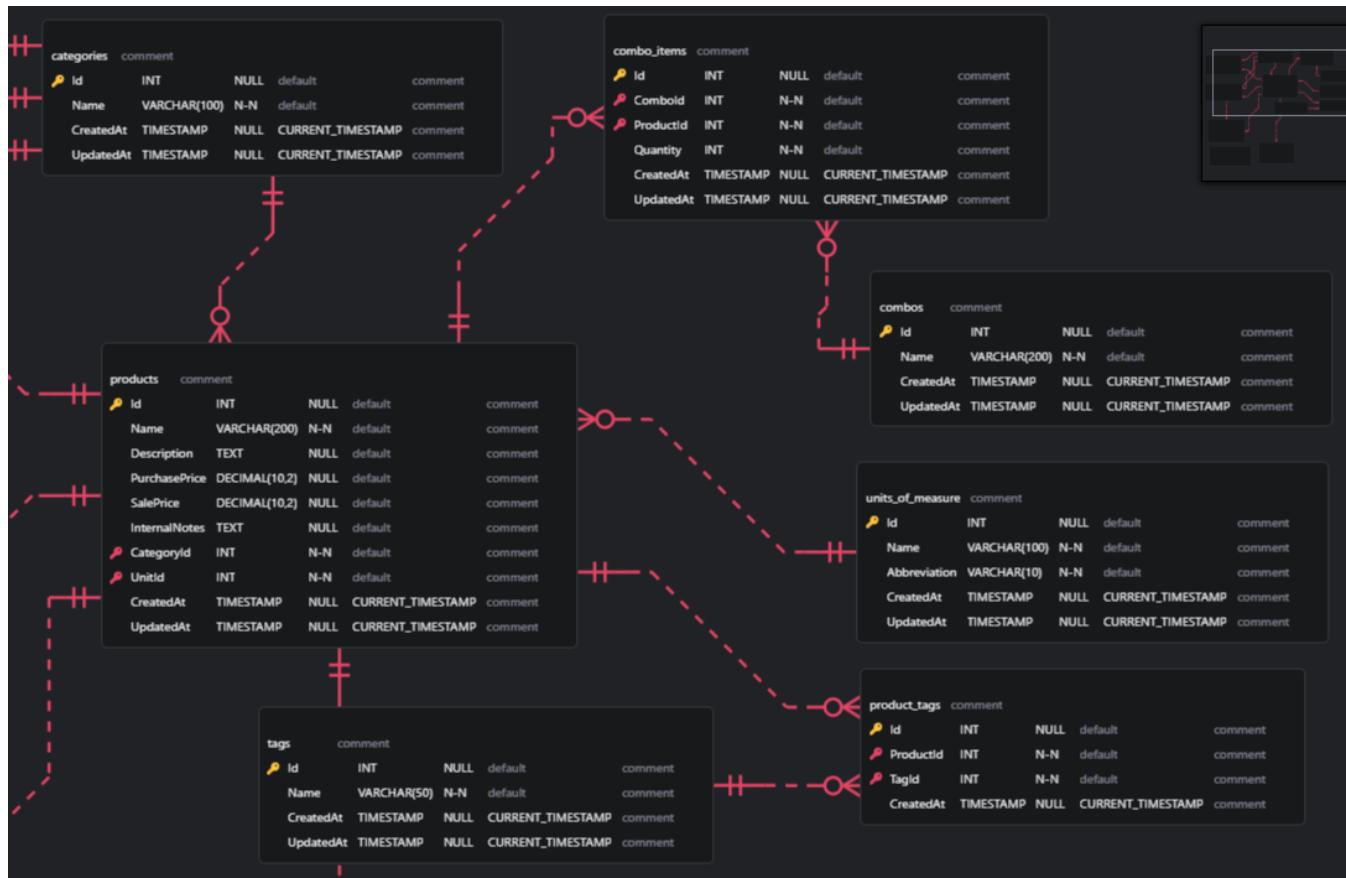
Clase	Tipo	Propósito	Atributos / Métodos
ProductRepository	Repository Implementation	Implementación acceso a productos (EF Core)	context, FindProductByNameAsync, FindAllProductsAsync, FindProductsByCategoryAsync, FindProductsByTagAsync, FindProductsByPriceRangeAsync, FindProductWithRelationsAsync
CategoryRepository	Repository Implementation	Implementación acceso a categorías (EF Core)	context, FindCategoryByNameAsync, FindAllCategoriesAsync
UnitRepository	Repository Implementation	Implementación acceso a unidades (EF Core)	context, FindUnitByNameAsync, FindAllUnitsAsync
TagRepository	Repository Implementation	Implementación acceso a etiquetas (EF Core)	context, FindTagByNameAsync, FindAllTagsAsync, FindTagsByIdsAsync
ComboRepository	Repository Implementation	Implementación acceso a combos (EF Core)	context, FindComboByNameAsync, FindAllCombosAsync

2.6.4.5. Bounded Context Software Architecture Component Level Diagrams

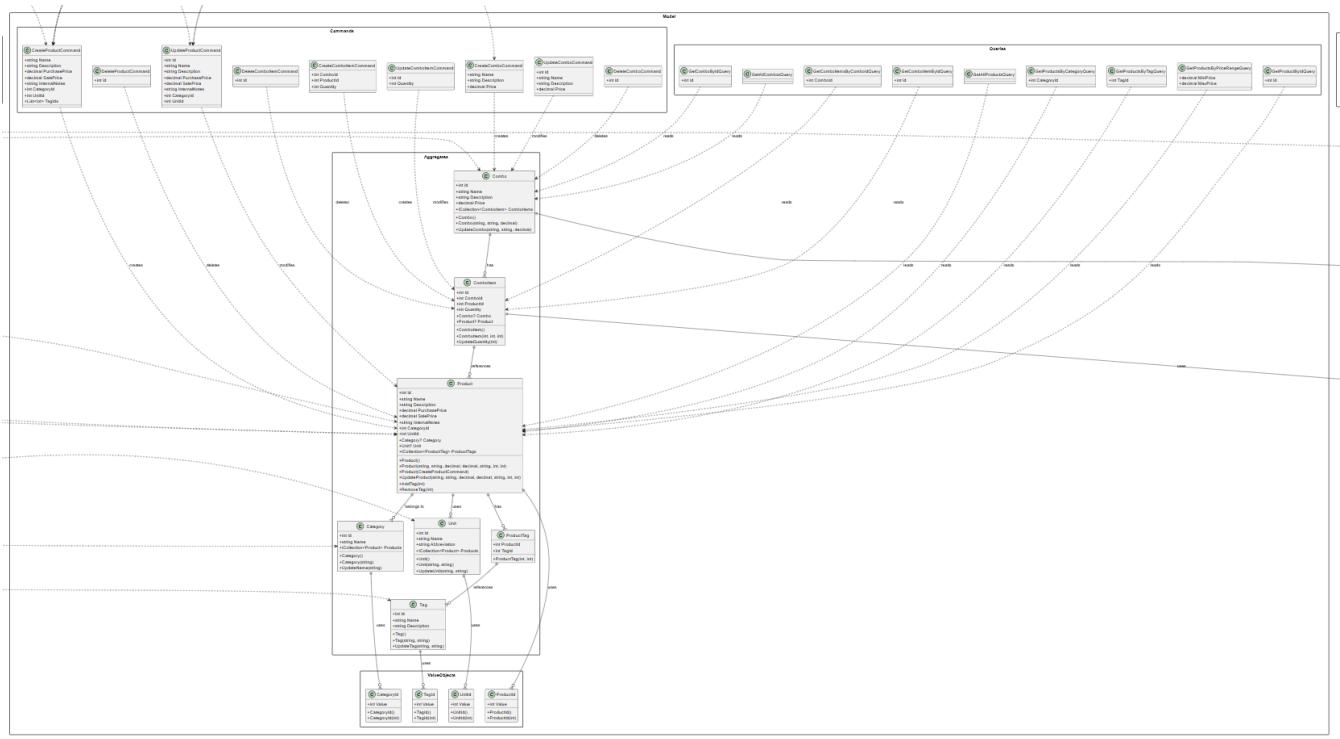


2.6.4.6. Bounded Context Software Architecture Code Level Diagrams

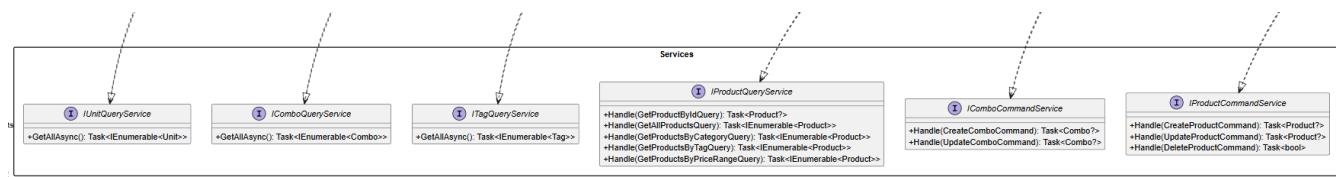
2.6.4.6.1. Bounded Context Domain Layer Class Diagrams



Domain: Model



Domain: Services



2.6.4.6.2. Bounded Context Database Design Diagram

2.6.5. Bounded Context: Reports Context

2.6.5.1. Domain Layer

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
CategoryReport	Aggregate Root (Entity)	Reporte de productos por categoría	Id, Categoria, Producto, FechaConsulta, PrecioUnitario, Cantidad, Total, CategoryReport(), CategoryReport(...), UpdateCantidad(int), UpdatePrecio(decimal)
StockAverageReport	Aggregate Root (Entity)	Reporte de stock promedio por producto y categoría	Id, StockPromedio, Categoria, Producto, FechaConsulta, StockIdeal, Estado, StockAverageReport(...)
ReportType	Enum	Define el tipo de reporte	Category, StockAverage
ReportDate	Value Object	Fecha de un reporte	Value, ReportDate(DateTime)
StockStatus	Value Object	Estado de stock	Value, StockStatus(string)
ProductName	Value Object	Nombre del producto	Value, ProductName(string)
CategoryName	Value Object	Nombre de la categoría	Value, CategoryName(string)
CreateCategoryReportCommand	Command	Crear reporte por categoría	Categoría, Producto, FechaConsulta, PrecioUnitario, Cantidad

Clase/Interfaz	Tipo	Propósito	Atributos / Métodos
CreateStockAverageReportCommand	Command	Crear reporte de stock promedio	StockPromedio, Categoria, Producto, FechaConsulta, StockIdeal, Estado
GetAllCategoryReportsQuery	Query	Consultar todos los reportes de categoría	-
GetCategoryReportsByDateQuery	Query	Consultar reportes de categoría por fecha	FechaConsulta
GetCategoryReportByIdQuery	Query	Consultar reporte de categoría por Id	Id
GetAllStockAverageReportsQuery	Query	Consultar todos los reportes de stock promedio	-
GetStockAverageReportsByDateQuery	Query	Consultar reportes de stock promedio por fecha	FechaConsulta
GetStockAverageReportByIdQuery	Query	Consultar reporte de stock promedio por Id	Id
ICategoryReportCommandService	Domain Service (Interface)	Define contrato para comandos de reportes de categoría	Handle(CreateCategoryReportCommand)
ICategoryReportQueryService	Domain Service (Interface)	Define contrato para consultas de reportes de categoría	Handle(GetAllCategoryReportsQuery), Handle(GetCategoryReportsByDateQuery), Handle(GetCategoryReportByIdQuery)
IStockAverageReportCommandService	Domain Service (Interface)	Define contrato para comandos de reportes de stock promedio	Handle(CreateStockAverageReportCommand)
IStockAverageReportQueryService	Domain Service (Interface)	Define contrato para consultas de reportes de stock promedio	Handle(GetAllStockAverageReportsQuery), Handle(GetStockAverageReportsByDateQuery)
ICategoryReportRepository	Repository (Interface)	Acceso a reportes de categoría	FindAllAsync(), FindByDateAsync(DateTime), AddAsync(CategoryReport), FindByIdAsync(int)
IStockAverageReportRepository	Repository (Interface)	Acceso a reportes de stock promedio	FindAllAsync(), FindByDateAsync(DateTime), AddAsync(StockAverageReport), FindByIdAsync(int)

2.6.5.2. Application Layer

Clase	Tipo	Propósito	Atributos / Métodos
CategoryReportCommandService	Application Service	Implementa comandos de reportes de categoría	repository, unitOfWork, Handle(CreateCategoryReportCommand)
StockAverageReportCommandService	Application Service	Implementa comandos de reportes de stock promedio	repository, unitOfWork, Handle(CreateStockAverageReportCommand)
CategoryReportQueryService	Application Service	Implementa consultas de reportes de categoría	repository, Handle(GetAllCategoryReportsQuery), Handle(GetCategoryReportsByDateQuery), Handle(GetCategoryReportByIdQuery)
StockAverageReportQueryService	Application Service	Implementa consultas de reportes de stock promedio	repository, Handle(GetAllStockAverageReportsQuery), Handle(GetStockAverageReportsByDateQuery)

2.6.5.3. Interface Layer

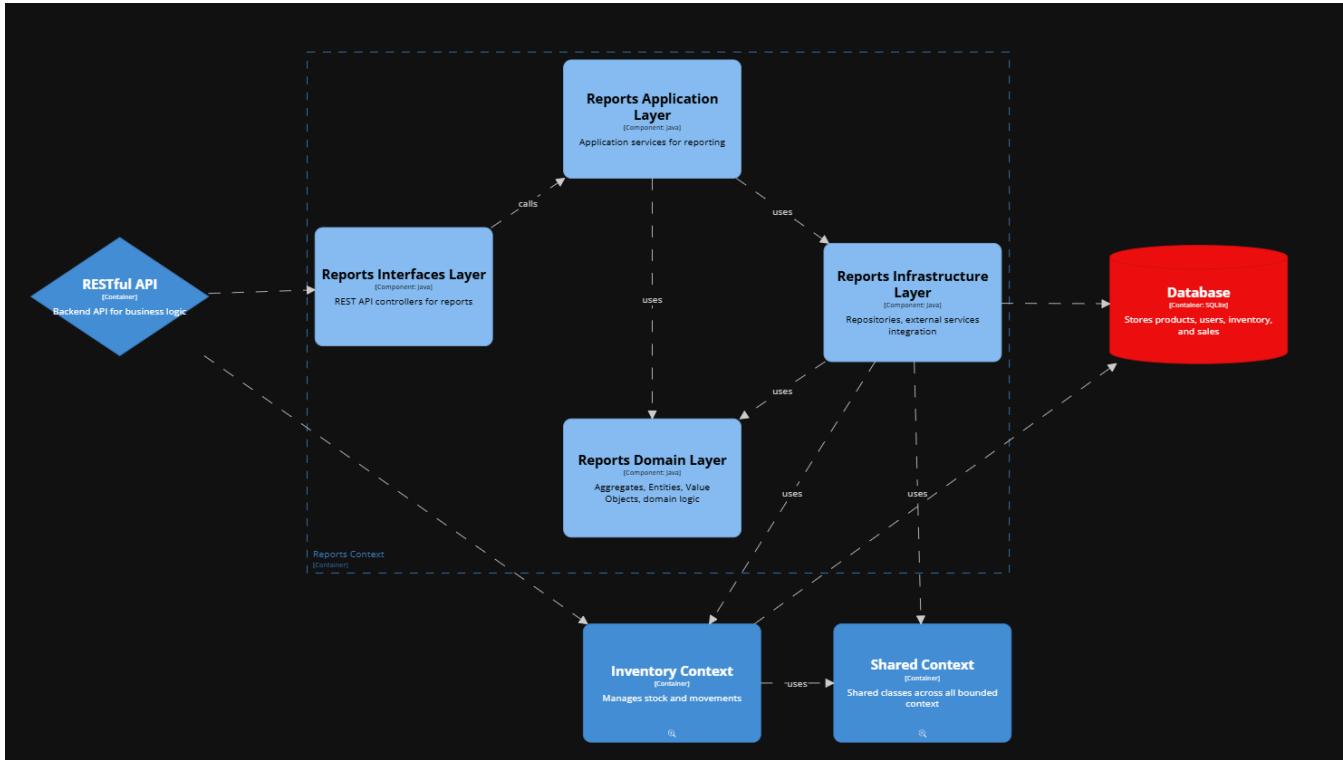
Clase	Tipo	Propósito	Atributos / Métodos

Clase	Tipo	Propósito	Atributos / Métodos
GeneralReportResource	Resource (DTO)	Representa todos los reportes (categoría y stock promedio)	CategoryReports, StockAverageReports
CategoryReportResource	Resource (DTO)	Representa reporte de categoría en respuestas	Id, Categoria, Producto, FechaConsulta, PrecioUnitario, Cantidad, Total
StockAverageReportResource	Resource (DTO)	Representa reporte de stock promedio en respuestas	Id, StockPromedio, Categoria, Producto, FechaConsulta, StockIdeal, Estado
CreateCategoryReportResource	Resource (DTO)	Datos de entrada para crear un reporte de categoría	Categoría, Producto, FechaConsulta, PrecioUnitario, Cantidad
CreateStockAverageReportResource	Resource (DTO)	Datos de entrada para crear un reporte de stock promedio	StockPromedio, Categoria, Producto, FechaConsulta, StockIdeal, Estado
CategoryReportResourceAssembler	Assembler	Convierte entidad CategoryReport en DTO	ToResource(CategoryReport)
StockAverageReportResourceAssembler	Assembler	Convierte entidad StockAverageReport en DTO	ToResource(StockAverageReport)
CreateCategoryReportCommandAssembler	Assembler	Convierte recurso en comando	ToCommand(CreateCategoryReportResource)
CreateStockAverageReportCommandAssembler	Assembler	Convierte recurso en comando	ToCommand(CreateStockAverageReportResource)
ReportController	REST Controller	Expone endpoints para reportes	GetAllReports(), CreateCategory(CreateCategoryReportResource), GetAllCategoryReports(), GetCategoryReportById(int), GetCategoryByDate(DateTime), CreateStockAverage(CreateStockAverageReportResource), GetAllStockAverageReports(), GetStockAverageReportById(int), GetStockAverageByDate(DateTime)

2.6.5.4. Infrastructure Layer

Clase	Tipo	Propósito	Atributos / Métodos
CategoryReportRepository	Repository Implementation	Implementación EF Core para reportes de categoría	context, FindAllAsync(), FindByDateAsync(DateTime), AddAsync(CategoryReport), FindByIdAsync(int)
StockAverageReportRepository	Repository Implementation	Implementación EF Core para reportes de stock promedio	context, FindAllAsync(), FindByDateAsync(DateTime), AddAsync(StockAverageReport), FindByIdAsync(int)

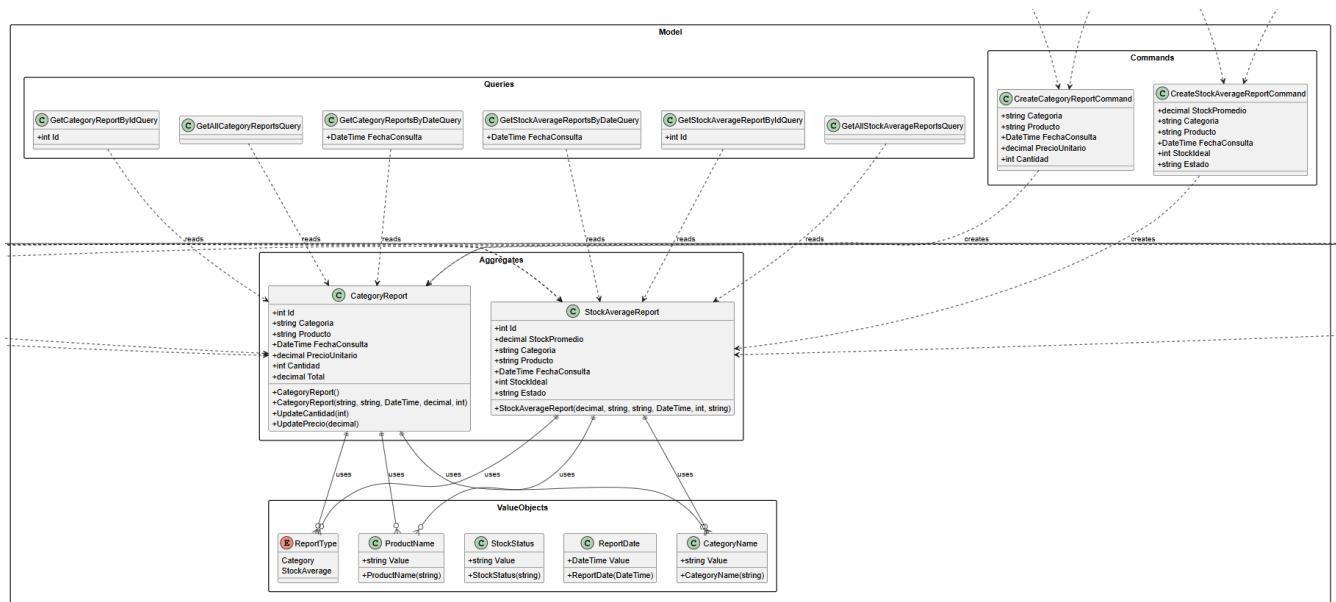
2.6.5.5. Bounded Context Software Architecture Component Level Diagrams



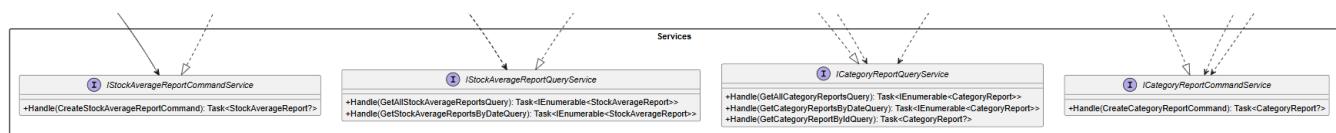
2.6.5.6. Bounded Context Software Architecture Code Level Diagrams

2.6.5.6.1. Bounded Context Domain Layer Class Diagrams

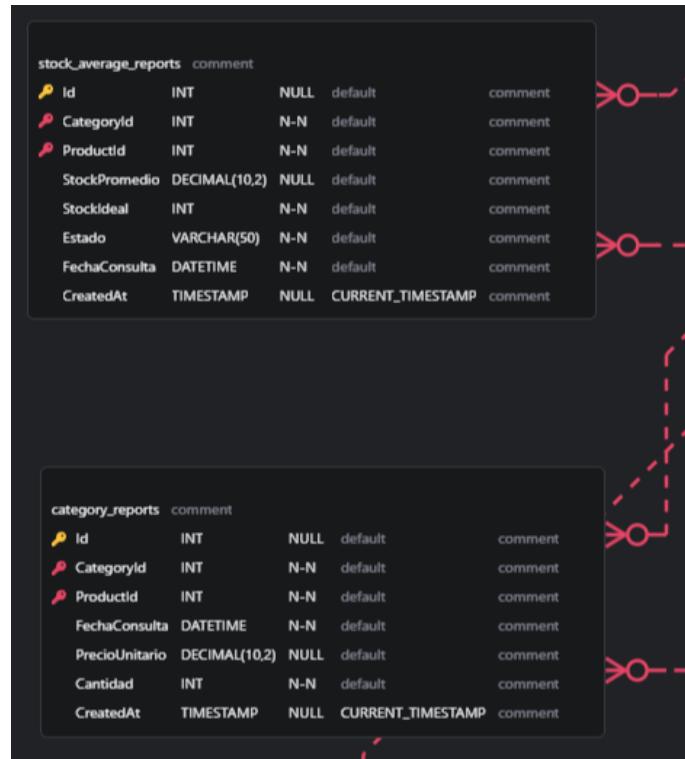
Domain: Model



Domain: Services



2.6.5.6.2. Bounded Context Database Design Diagram



Capítulo III: Solution UI/UX Design

3.1. Product design

3.1.1. Style Guidelines

3.1.1.1. General Style Guidelines

3.1.2. Information Architecture

3.1.2.1. Organization Systems

3.1.2.2. Labelling Systems

3.1.2.3. SEO Tags and Meta Tags

3.1.2.4. Searching Systems

3.1.2.5. Navigation Systems

3.1.3. Landing Page UI Design

3.1.3.1. Landing Page Wireframe

3.1.3.2. Landing Page Mock-up

3.1.4. Mobile Applications UX/UI Design

3.1.4.1. Mobile Applications Wireframes

3.1.4.2. Mobile Applications Wireflow Diagrams

3.1.4.3. Mobile Applications Mock-ups

3.1.4.4. Mobile Applications User Flow Diagrams

3.1.4.5. Mobile Applications Prototyping

Capítulo IV: Product Implementation & Validation

-
- 4.1. Software Configuration Management
 - 4.1.1. Software Development Environment Configuration
 - 4.1.2. Source Code Management
 - 4.1.3. Source Code Style Guide & Conventions
 - 4.1.4. Software Deployment Configuration
 - 4.2. Landing Page & Mobile Application Implementation
 - 4.2.1. Sprint n
 - 4.2.1.1. Sprint Planning n**
 - 4.2.1.2. Sprint Backlog n**
 - 4.2.1.3. Development Evidence for Sprint Review**
 - 4.2.1.4. Testing Suite Evidence for Sprint Review**
 - 4.2.1.5. Execution Evidence for Sprint Review**
 - 4.2.1.6. Services Documentation Evidence for Sprint Review**
 - 4.2.1.7. Software Deployment Evidence for Sprint Review**
 - 4.2.1.8. Team Collaboration Insights during Sprint**

Conclusiones y recomendaciones

El desarrollo de las funcionalidades propuestas para StockWise responde a las necesidades reales de emprendedores, pymes y startups que buscan optimizar su gestión de inventarios de manera simple, rápida y eficiente. A través de un enfoque centrado en el usuario, se priorizan características esenciales como el control de stock, la generación de alertas inteligentes, reportes visuales y una experiencia de uso intuitiva. Además, la estrategia de diferenciación frente a la competencia, basada en la simplicidad, accesibilidad y enfoque especializado, refuerza el potencial de StockWise para posicionarse como una solución líder en el mercado latinoamericano. Con esta base sólida, el proyecto avanza hacia la creación de una aplicación móvil de alto valor, adaptable y preparada para el crecimiento continuo.

Video App Validation

Video About-the-Product

Video About-the-team

Bibliografía

- 1. Dux Software. (2025, 9 abril). Dux Software: El Sistema de Gestión para tu Negocio. <https://www.duxsoftware.com.ar/>
- 2. Mecalux. (s. f.). Store fulfillment. <https://www.mecalux.pe/software/store-fulfillment>
- 3. SoftDolt. (s. f.). Vendus: agiliza la gestión de tu punto de venta. <https://www.softwaredoit.es/vendus/vendus.html>

Anexos

Video de Entrevistas: