

# COURSE PROJECT

---

Universidad Peruana de Ciencias Aplicadas



Ingeniería de Software - 2025-1  
Aplicaciones Web - 4366  
Profesor: Alex Humberto Sánchez Ponce

Informe del Trabajo Final

Startup: AyniTech  
Producto:

Team Members:

Member	Code
Jocelyn Damaly Almerco Rojas	U20221g068
Sanchez Rios, Camila Cristina	U202210973
Henry Kalet Esteban Roman	U202310210
Maria Fernanda Peña Riofrio	U202113279
Fabiola Del Rocio Saldaña Ayala	U202313773

Abril, 2025

## Registro de versiones del Informe

---

Versión	Fecha	Autor	Descripción de modificaciones
TB1	07/04/2025	• ..	• Capítulo I: Introducción
		• Camila Sanchez	• Capítulo II: Requirements Elicitation & Analysis
		• Maria	• Capítulo III: Requirements Specification
		• Fernanda Peña	• Capítulo IV: Product Design
		• ..	• Avance del Capítulo V: Product Implementation, Validation & Deployment hasta el punto 5.2.1.8
			• Avance de Conclusiones, Bibliografía y Anexos

# Project Report Collaboration Insights

Link de repositorio del reporte:: <https://github.com/upc-pre-202510-1asi0730-4366-AyniTech/Report>

## Contenido

### Student Outcome

#### Capítulo I: Introducción

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

#### Capítulo II: Requirements Elicitation & Analysis

- COURSE PROJECT
- Project Report Collaboration Insights
- Contenido
- Student Outcome
- Capítulo I: Introducción
  - 1.1. Startup Profile
    - 1.1.1. Descripción de la Startup
    - 1.1.2. Perfiles de integrantes del equipo
  - 1.2. Solution Profile
    - 1.2.1 Antecedentes y problemática
    - 1.2.2 Lean UX Process

- 1.2.2.1. Lean UX Problem Statements
  - 1.2.2.2. Lean UX Assumptions
  - 1.2.2.3. Lean UX Hypothesis Statements
  - 1.2.2.4. Lean UX Canvas
- 1.3. Segmentos objetivo
- Capítulo II: Requirements Elicitation & Analysis
  - 2.1. Competidores
    - 2.1.1. Análisis competitivo
    - 2.1.2. Estrategias y tácticas frente a competidores
  - 2.2. Entrevistas
    - 2.2.1. Diseño de entrevistas
    - 2.2.2. Registro de entrevistas
    - 2.2.3. Análisis de entrevistas
  - 2.3. Needfinding
    - 2.3.1. User Personas
    - 2.3.2. User Task Matrix
    - 2.3.3. User Journey Mapping
    - 2.3.4. Empathy Mapping
    - 2.3.5. As-is Scenario Mapping
  - 2.4. Ubiquitous Language
- Capítulo III: Requirements Specification
  - 3.1. To-Be Scenario Mapping
  - 3.2. User Stories
  - 3.3. Impact Mapping
  - 3.4. Product Backlog
- Capítulo IV: Product Design
  - 4.1. Style Guidelines
    - 4.1.1. General Style Guidelines
    - 4.1.2. Web Style Guidelines
  - 4.2. Information Architecture
    - 4.2.1. Organization Systems.
    - 4.2.2. Labeling Systems.
    - 4.2.3. SEO Tags and Meta Tags
    - 4.2.4. Searching Systems.
    - 4.2.5. Navigation Systems.
  - 4.3. Landing Page UI Design.
    - 4.3.1. Landing Page Wireframe.
    - 4.3.2. Landing Page Mock-up.
  - 4.4. Web Applications UX/UI Design.
    - 4.4.1. Web Applications Wireframes.
    - 4.4.2. Web Applications Wireflow Diagrams.
    - 4.4.2. Web Applications Mock-ups.
    - 4.4.3. Web Applications User Flow Diagrams.
  - 4.5. Web Applications Prototyping.
  - 4.6. Domain-Driven Software Architecture.
    - 4.6.1. Software Architecture Context Diagram.

- 4.6.2. Software Architecture Container Diagrams.
  - 4.6.3. Software Architecture Components Diagrams.
- 4.7. Software Object-Oriented Design.
  - 4.7.1. Class Diagrams.
  - 4.7.2. Class Dictionary.
- 4.8. Database Design.
  - 4.8.1. Database Diagram.
- Capítulo V: Product Implementation, Validation & Deployment
  - 5.1. Software Configuration Management.
    - 5.1.1. Software Development Environment Configuration.
    - 5.1.2. Source Code Management.
    - 5.1.3. Source Code Style Guide & Conventions.
    - 5.1.4. Software Deployment Configuration.
  - 5.2. Landing Page, Services & Applications Implementation
    - 5.2.1. Sprint 1
      - 5.2.1.1. Sprint Planning 1
      - 5.2.1.2. Aspect Leaders and Collaborators.
      - 5.2.1.3. Sprint Backlog n.
      - 5.2.1.4. Development Evidence for Sprint Review.
      - 5.2.1.5. Execution Evidence for Sprint Review.
      - 5.2.1.6. Services Documentation Evidence for Sprint Review.
      - 5.2.1.7. Software Deployment Evidence for Sprint Review.
      - 5.2.1.8. Team Collaboration Insights during Sprint.
  - 5.3. Validation Interviews.
    - 5.3.1. Diseño de Entrevistas.
    - 5.3.2. Registro de Entrevistas.
    - 5.3.3. Evaluaciones según heurísticas.
  - 5.4. Video About-the-Product.
- Conclusiones
  - Conclusiones y recomendaciones.
- Video About-the-Team.
- Bibliografía
- Anexos

### Capítulo III: Requirements Specification

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

### Capítulo IV: Product Design

- COURSE PROJECT
- Project Report Collaboration Insights
- Contenido
- Student Outcome
- Capítulo I: Introducción

- 1.1. Startup Profile
  - 1.1.1. Descripción de la Startup
  - 1.1.2. Perfiles de integrantes del equipo
- 1.2. Solution Profile
  - 1.2.1 Antecedentes y problemática
  - 1.2.2 Lean UX Process
    - 1.2.2.1. Lean UX Problem Statements
    - 1.2.2.2. Lean UX Assumptions
    - 1.2.2.3. Lean UX Hypothesis Statements
    - 1.2.2.4. Lean UX Canvas
- 1.3. Segmentos objetivo
- Capítulo II: Requirements Elicitation & Analysis
  - 2.1. Competidores
    - 2.1.1. Análisis competitivo
    - 2.1.2. Estrategias y tácticas frente a competidores
  - 2.2. Entrevistas
    - 2.2.1. Diseño de entrevistas
    - 2.2.2. Registro de entrevistas
    - 2.2.3. Análisis de entrevistas
  - 2.3. Needfinding
    - 2.3.1. User Personas
    - 2.3.2. User Task Matrix
    - 2.3.3. User Journey Mapping
    - 2.3.4. Empathy Mapping
    - 2.3.5. As-is Scenario Mapping
  - 2.4. Ubiquitous Language
- Capítulo III: Requirements Specification
  - 3.1. To-Be Scenario Mapping
  - 3.2. User Stories
  - 3.3. Impact Mapping
  - 3.4. Product Backlog
- Capítulo IV: Product Design
  - 4.1. Style Guidelines
    - 4.1.1. General Style Guidelines
    - 4.1.2. Web Style Guidelines
  - 4.2. Information Architecture
    - 4.2.1. Organization Systems.
    - 4.2.2. Labeling Systems.
    - 4.2.3. SEO Tags and Meta Tags
    - 4.2.4. Searching Systems.
    - 4.2.5. Navigation Systems.
  - 4.3. Landing Page UI Design.
    - 4.3.1. Landing Page Wireframe.
    - 4.3.2. Landing Page Mock-up.
  - 4.4. Web Applications UX/UI Design.
    - 4.4.1. Web Applications Wireframes.

- 4.4.2. Web Applications Wireflow Diagrams.
  - 4.4.2. Web Applications Mock-ups.
  - 4.4.3. Web Applications User Flow Diagrams.
- 4.5. Web Applications Prototyping.
- 4.6. Domain-Driven Software Architecture.
  - 4.6.1. Software Architecture Context Diagram.
  - 4.6.2. Software Architecture Container Diagrams.
  - 4.6.3. Software Architecture Components Diagrams.
- 4.7. Software Object-Oriented Design.
  - 4.7.1. Class Diagrams.
  - 4.7.2. Class Dictionary.
- 4.8. Database Design.
  - 4.8.1. Database Diagram.
- Capítulo V: Product Implementation, Validation & Deployment
  - 5.1. Software Configuration Management.
    - 5.1.1. Software Development Environment Configuration.
    - 5.1.2. Source Code Management.
    - 5.1.3. Source Code Style Guide & Conventions.
    - 5.1.4. Software Deployment Configuration.
  - 5.2. Landing Page, Services & Applications Implementation
    - 5.2.1. Sprint 1
      - 5.2.1.1. Sprint Planning 1
      - 5.2.1.2. Aspect Leaders and Collaborators.
      - 5.2.1.3. Sprint Backlog n.
      - 5.2.1.4. Development Evidence for Sprint Review.
      - 5.2.1.5. Execution Evidence for Sprint Review.
      - 5.2.1.6. Services Documentation Evidence for Sprint Review.
      - 5.2.1.7. Software Deployment Evidence for Sprint Review.
      - 5.2.1.8. Team Collaboration Insights during Sprint.
  - 5.3. Validation Interviews.
    - 5.3.1. Diseño de Entrevistas.
    - 5.3.2. Registro de Entrevistas.
    - 5.3.3. Evaluaciones según heurísticas.
  - 5.4. Video About-the-Product.
- Conclusiones
  - Conclusiones y recomendaciones.
- Video About-the-Team.
- Bibliografía
- Anexos

## Capítulo V: Product Implementation, Validation & Deployment

- COURSE PROJECT
- Project Report Collaboration Insights
- Contenido
- Student Outcome
- Capítulo I: Introducción

- 1.1. Startup Profile
  - 1.1.1. Descripción de la Startup
  - 1.1.2. Perfiles de integrantes del equipo
- 1.2. Solution Profile
  - 1.2.1 Antecedentes y problemática
  - 1.2.2 Lean UX Process
    - 1.2.2.1. Lean UX Problem Statements
    - 1.2.2.2. Lean UX Assumptions
    - 1.2.2.3. Lean UX Hypothesis Statements
    - 1.2.2.4. Lean UX Canvas
- 1.3. Segmentos objetivo
- Capítulo II: Requirements Elicitation & Analysis
  - 2.1. Competidores
    - 2.1.1. Análisis competitivo
    - 2.1.2. Estrategias y tácticas frente a competidores
  - 2.2. Entrevistas
    - 2.2.1. Diseño de entrevistas
    - 2.2.2. Registro de entrevistas
    - 2.2.3. Análisis de entrevistas
  - 2.3. Needfinding
    - 2.3.1. User Personas
    - 2.3.2. User Task Matrix
    - 2.3.3. User Journey Mapping
    - 2.3.4. Empathy Mapping
    - 2.3.5. As-is Scenario Mapping
  - 2.4. Ubiquitous Language
- Capítulo III: Requirements Specification
  - 3.1. To-Be Scenario Mapping
  - 3.2. User Stories
  - 3.3. Impact Mapping
  - 3.4. Product Backlog
- Capítulo IV: Product Design
  - 4.1. Style Guidelines
    - 4.1.1. General Style Guidelines
    - 4.1.2. Web Style Guidelines
  - 4.2. Information Architecture
    - 4.2.1. Organization Systems.
    - 4.2.2. Labeling Systems.
    - 4.2.3. SEO Tags and Meta Tags
    - 4.2.4. Searching Systems.
    - 4.2.5. Navigation Systems.
  - 4.3. Landing Page UI Design.
    - 4.3.1. Landing Page Wireframe.
    - 4.3.2. Landing Page Mock-up.
  - 4.4. Web Applications UX/UI Design.
    - 4.4.1. Web Applications Wireframes.

- 4.4.2. Web Applications Wireflow Diagrams.
  - 4.4.2. Web Applications Mock-ups.
  - 4.4.3. Web Applications User Flow Diagrams.
- 4.5. Web Applications Prototyping.
- 4.6. Domain-Driven Software Architecture.
  - 4.6.1. Software Architecture Context Diagram.
  - 4.6.2. Software Architecture Container Diagrams.
  - 4.6.3. Software Architecture Components Diagrams.
- 4.7. Software Object-Oriented Design.
  - 4.7.1. Class Diagrams.
  - 4.7.2. Class Dictionary.
- 4.8. Database Design.
  - 4.8.1. Database Diagram.
- Capítulo V: Product Implementation, Validation & Deployment
  - 5.1. Software Configuration Management.
    - 5.1.1. Software Development Environment Configuration.
    - 5.1.2. Source Code Management.
    - 5.1.3. Source Code Style Guide & Conventions.
    - 5.1.4. Software Deployment Configuration.
  - 5.2. Landing Page, Services & Applications Implementation
    - 5.2.1. Sprint 1
      - 5.2.1.1. Sprint Planning 1
      - 5.2.1.2. Aspect Leaders and Collaborators.
      - 5.2.1.3. Sprint Backlog n.
      - 5.2.1.4. Development Evidence for Sprint Review.
      - 5.2.1.5. Execution Evidence for Sprint Review.
      - 5.2.1.6. Services Documentation Evidence for Sprint Review.
      - 5.2.1.7. Software Deployment Evidence for Sprint Review.
      - 5.2.1.8. Team Collaboration Insights during Sprint.
  - 5.3. Validation Interviews.
    - 5.3.1. Diseño de Entrevistas.
    - 5.3.2. Registro de Entrevistas.
    - 5.3.3. Evaluaciones según heurísticas.
  - 5.4. Video About-the-Product.
- Conclusiones
  - Conclusiones y recomendaciones.
- Video About-the-Team.
- Bibliografía
- Anexos

## Conclusiones

- Conclusiones y recomendaciones
- Video About-the-Team

## Bibliografía

## Anexos



# Student Outcome

ABET – EAC - Student Outcome 5

Criterio: La capacidad de funcionar efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos.

Criterio específico	Acciones realizadas	Conclusiones
Comunica oralmente con efectividad a diferentes rangos de audiencia.	name	TB1:
	TB1:	TP1:
	..	.
	TP1:	TB2:
	,...	TF:
	TB2:	
	.	
	TF:	
	.	
	name	
	TB1:	
	TP1:	
	TB2:	
	TF:	
	.	
	name	
	TB1:	
	TP1:	
	.	
	TB2:	
	TF:	
	.	

TB1:  
TP1:  
TB2:  
TF:

Comunica por escrito con efectividad a diferentes rangos de audiencia.

name	TB1:
TB1:	..
..	TP1:
TP1:	...
..	TB2:
TB2:	..
..	TF:
TF:	...

..

name
TB1:
..
TP1:
..
TB2:
...
TF:
.

name
TB1:
..
TP1:
..
TB2:

...

**TF:**

.

**name**

**TB1:**

...

**TP1:**

...

**TB2:**

...

**TF:**

...

# Capítulo I: Introducción

---

## 1.1. Startup Profile

### 1.1.1. Descripción de la Startup

StockWise es una plataforma web 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 y la generación de reportes detallados, 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, lo cual genera errores, desorganización, sobrecompras y pérdidas económicas. StockWise busca resolver este problema digitalizando y centralizando el control del inventario, permitiendo a los negocios tomar decisiones basadas en datos y optimizar su operación.

**Misión:** Brindar a pequeñas y medianas empresas una solución de gestión de inventarios 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 la plataforma 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



Foto de perfil de

name

..



---

**Saldaña Ayala, Fabiola del Rocio**

Soy estudiante de la carrera de Ingeniería de Software, actualmente me encuentro en el quinto ciclo. Me gusta viajar, aprender nuevos idiomas, leer y me interesa aprender más acerca de nuevas tecnologías.



---

**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 setimo ciclo. Me gusta escuchar música y leer en los ratos libres y aprender más sobre la carrera.



---

**Peña Riofrio, Maria Fernanda**

Soy estudiante de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas, actualmente me encuentro en el sexto ciclo. En mismo tiempos libres me gusta leer sobre temas diversos y ver películas.



---

**Esteban Román, Henr Kalet**

Soy estudiante de la carrera de Ingeniería de Software. Actualmente me encunetro en 5to ciclo y me gusta mucho escuchar música y pasear.

## 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 hojas de Excel, 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 plataforma web 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 plataforma 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.
2. Estas necesidades se pueden satisfacer con una plataforma web intuitiva, escalable y accesible.
3. Nuestros clientes iniciales serán emprendedores y start ups con bodegas especializadas.
4. El valor más importante que un cliente quiere de nuestros servicios es control y visibilidad de su inventario.
5. El cliente también va a obtener alertas inteligentes, reportes automáticos y una experiencia intuitiva.
6. Vamos a obtener la mayoría de los clientes mediante publicidad en línea y asociaciones con comunidades de emprendedores.
7. Vamos a obtener ingresos mediante un modelo freemium con upgrade a planes premium.
8. Nuestra competencia en el mercado serán herramientas genéricas como Excel, Google Sheets y ERPs complejos.
9. Vamos a tener ventaja frente a nuestra competencia debido a nuestra simplicidad, enfoque en pymes, y especialización en inventarios.
10. El mayor riesgo del servicio es la baja adopción o la percepción de que es complejo o costoso.
11. Lo resolveremos realizando pruebas con usuarios reales, iteraciones constantes y una estrategia clara de onboarding.

User Assumptions:

¿Quién es el usuario? Dueños de negocios, encargados de bodegas o logística en pymes/startups.

¿Qué problemas tiene nuestro producto que busca resolver? Falta de control de stock, errores manuales, sobre costos, quiebres de stock.

¿Qué características son importantes? Registro de productos, historial de movimientos, alertas, reportes, roles y permisos.

¿Dónde encaja nuestro producto en su vida/trabajo? En la gestión diaria del inventario, decisiones de compra, y control de logística.

¿Cuándo y cómo es usado nuestro producto? A diario, desde computadoras o móviles, al inicio y cierre de turnos, y durante reposiciones.

¿Cómo debe verse y comportarse? Simple, rápida, intuitiva, con interfaz limpia, adaptable a cualquier dispositivo.

### Feature Assumptions:

Creemos que la aplicación debe contar con una interfaz de usuario intuitiva y responsiva que permitirá a los encargados de almacén, emprendedores y administradores adoptarla sin dificultad, reduciendo la curva de aprendizaje.

Creemos que la plataforma debe proporcionar notificaciones y alertas personalizables (como stock bajo o fechas próximas de vencimiento) que mantendrán a los usuarios informados de manera oportuna para evitar errores logísticos y quiebres de stock.

Creemos que el sistema debe incluir herramientas de generación de reportes y gráficos que permitan visualizar fácilmente información clave del inventario (productos más rotados, niveles de stock, tendencias de ventas), ayudando a los usuarios a tomar decisiones informadas.

#### 1.2.2.3. Lean UX Hypothesis Statements

Creemos que los usuarios valorarán las alertas inteligentes.

sabremos que es cierto.

cuando recibamos al menos un 80% de los usuarios, un feedback positivo y alta frecuencia de uso en esta funcionalidad.

Creemos que los usuarios encontrarán valiosos los reportes visuales.

sabremos que es cierto .

cuando observemos mínimamente un 20% mayor retención y uso del módulo de reportes.

Creemos que ofrecer una versión gratuita con funcionalidades limitadas incentivará la adopción inicial.

sabremos que es cierto.

cuando el número de registros y usuarios activos supere nuestras métricas objetivo durante el primer mes.

Creemos que implementar alertas inteligentes por stock bajo y fechas de vencimiento reducirá los quiebres de stock.

Sabremos que tenemos razón

Cuando comparemos el número de productos con quiebre antes y después de la activación de alertas por parte del usuario.

#### 1.2.2.4. Lean UX Canvas

### 1.3. Segmentos objetivo

#### Segmento 1: Bodegas especializadas por rubro

Este segmento incluye negocios que poseen bodegas físicas para almacenar productos específicos según su rubro. Suelen ser tiendas mayoristas o minoristas que necesitan tener un control formal del inventario sobre productos que varían en características como tallas, colores, fechas de vencimiento o dimensiones técnicas. Estas bodegas enfrentan desafíos como la alta rotación de productos, pérdidas por falta de trazabilidad o desorden, y decisiones de compra poco informadas. StockWise les permite organizar su inventario de forma clara, generar alertas inteligentes según tipo de producto, y tomar decisiones basadas en reportes visuales y detallados.

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

Son negocios en crecimiento que han comenzado a escalar su operación, ya sea vendiendo online (Shopify, MercadoLibre, Instagram) o desde un punto físico, y que almacenan sus productos en bodegas propias, alquiladas o incluso en casa. En esta etapa, el desorden en el inventario puede afectar la atención al cliente, las ventas y la planificación de compras. No cuentan con un sistema formal, pero necesitan profesionalizar su operación para seguir creciendo.

## Capítulo II: Requirements Elicitation & Analysis

---

### 2.1. Competidores

#### 2.1.1. Análisis competitivo

#### 2.1.2. Estrategias y tácticas frente a competidores

### 2.2. Entrevistas

#### 2.2.1. Diseño de entrevistas

#### 2.2.2. Registro de entrevistas

#### 2.2.3. Análisis de entrevistas

### 2.3. Needfinding

#### 2.3.1. User Personas

#### 2.3.2. User Task Matrix

#### 2.3.3. User Journey Mapping

#### 2.3.4. Empathy Mapping

#### 2.3.5. As-is Scenario Mapping

### 2.4. Ubiquitous Language

## Capítulo III: Requirements Specification

---

### 3.1. To-Be Scenario Mapping

### 3.2. User Stories

### 3.3. Impact Mapping

### 3.4. Product Backlog

## Capítulo IV: Product Design

---



## 4.1. Style Guidelines

### 4.1.1. General Style Guidelines

### 4.1.2. Web Style Guidelines

## 4.2. Information Architecture

### 4.2.1. Organization Systems.

### 4.2.2. Labeling Systems.

### 4.2.3. SEO Tags and Meta Tags

### 4.2.4. Searching Systems.

### 4.2.5. Navigation Systems.

## 4.3. Landing Page UI Design.

### 4.3.1. Landing Page Wireframe.

### 4.3.2. Landing Page Mock-up.

## 4.4. Web Applications UX/UI Design.

### 4.4.1. Web Applications Wireframes.

### 4.4.2. Web Applications Wireflow Diagrams.

### 4.4.2. Web Applications Mock-ups.

### 4.4.3. Web Applications User Flow Diagrams.

## 4.5. Web Applications Prototyping.

## 4.6. Domain-Driven Software Architecture.

### 4.6.1. Software Architecture Context Diagram.

### 4.6.2. Software Architecture Container Diagrams.

### 4.6.3. Software Architecture Components Diagrams.

## 4.7. Software Object-Oriented Design.

### 4.7.1. Class Diagrams.

### 4.7.2. Class Dictionary.

## 4.8. Database Design.

#### 4.8.1. Database Diagram.

## Capítulo V: Product Implementation, Validation & Deployment

---

### 5.1. Software Configuration Management.

#### 5.1.1. Software Development Environment Configuration.

#### 5.1.2. Source Code Management.

#### 5.1.3. Source Code Style Guide & Conventions.

#### 5.1.4. Software Deployment Configuration.

### 5.2. Landing Page, Services & Applications Implementation

#### 5.2.1. Sprint 1

##### **5.2.1.1. Sprint Planning 1**

##### **5.2.1.2. Aspect Leaders and Collaborators.**

##### **5.2.1.3. Sprint Backlog n.**

##### **5.2.1.4. Development Evidence for Sprint Review.**

##### **5.2.1.5. Execution Evidence for Sprint Review.**

##### **5.2.1.6. Services Documentation Evidence for Sprint Review.**

##### **5.2.1.7. Software Deployment Evidence for Sprint Review.**

##### **5.2.1.8. Team Collaboration Insights during Sprint.**

### 5.3. Validation Interviews.

#### 5.3.1. Diseño de Entrevistas.

#### 5.3.2. Registro de Entrevistas.

#### 5.3.3. Evaluaciones según heurísticas.

### 5.4. Video About-the-Product.

## Conclusiones

---

### Conclusiones y recomendaciones.

## Video About-the-Team.

---

## Bibliografía

---

## Anexos

---