



Universidad Peruana de Ciencias Aplicadas

Ingeniería de software

Periodo 202502

1ACC0238| Aplicaciones para Dispositivos Móviles

NRC: 1807

Docente: Jorge Luis Mayta Guillermo

Informe del Trabajo Final

Nombre del StartUp: SyncCafe

Nombre del Producto: iCafe

U202319431 - Choquehuanca Nuñez, Luciana Carolina

U202312391 - Hallasi Saravia, Miguel Angel

U20231C111 - Inga Orihuela, Christian Fabrizio

U202222846 - Ramos Calagua, Sebastian Alexander

U202310008 - Urrutia Peña, Jasmin Adriana

U202117342 - Velasquez Chambi, Ruben

Diciembre 2025

Project Report Collaboration Insights

Enlace de la organización para el reporte del proyecto: <https://github.com/orgs/1ACC0238-2520-1807/dashboard>

TB1

Para el desarrollo del informe correspondiente a la entrega TB1, se estableció la implementación de secciones de la siguiente manera para cada integrante del equipo:

Integrante	Tareas Asignadas
Ruben Velasquez Chambi	User Stories, Impact Mapping, Product Backlog, Software Architecture Context Level Diagrams, Software Architecture Container Level Diagrams, Bounded Context: Inventory
Christian Fabrizio Inga Orihuela	Análisis competitivo, Estrategias y tácticas frente a competidores, Diseño de entrevistas, Registro de entrevistas, Análisis de entrevistas, Domain Message Flows Modeling, y Bounded Context: Product
Hallasi Saravia, Miguel Angel	User Personas, User Task Matrix, User Journey Mapping, Empathy Mapping, Ubiquitous Language, Context Mapping, Software Architecture Deployment Diagrams, Bounded Context: Finances
Urrutia Peña, Jasmin Adriana	Descripción de la Startup, Solution Profile, Antecedentes y problemática, Diseño de entrevistas, Registro de entrevistas, Análisis de entrevistas, EventStorming, Candidate Context Discovery y Bounded Context: Contacts
Sebastian Alexander Ramos Calagua	Lean UX Process, Bounded Context Canvases, Bounded Context: Sales

TP

Para el desarrollo del informe correspondiente a la entrega TP, cada integrante realizo la siguiente parte:

Integrante	Tareas Asignadas
Ruben Velasquez Chambi	Sprint 1 y Creacion del boundent Iam y Contacts en el backend
Christian Fabrizio Inga Orihuela	Implementacion de boundent Products en el frontend
Hallasi Saravia, Miguel Angel	Sprint 1 Y Creacion del boundent Inventory y Products en el backend
Urrutia Peña, Jasmin Adriana	Capítulo III: Solution UI/UX Design e implementación del boundent Iam , Contacts e Inventory en el frontend
Sebastian Alexander Ramos Calagua	Correcciones TB1 y Software Configuration Management

TB2

Para el desarrollo del informe correspondiente a la entrega TB2, cada integrante realizo la siguiente parte:

Integrante	Tareas Asignadas
Ruben Velasquez Chambi	Culminacion de Backend, spring planning, spring backlog.
Christian Fabrizio Inga Orihuela	Aplicación en Flutter, Corrección TP1 y video About the Product.
Hallasi Saravia, Miguel Angel	Culminación de Backend, Sprint 2
Urrutia Peña, Jasmin Adriana	Culminación de la Aplicación Android, Conclusiones y Recomendaciones
Sebastian Alexander Ramos Calagua	Aplicación en Flutter y Corrección TP1

TF

Para el desarrollo del informe correspondiente a la entrega TF, cada integrante realizo la siguiente parte:

Integrante	Tareas Asignadas
Ruben Velasquez Chambi	Documentacion y correccion del informe
Christian Fabrizio Inga Orihuela	Aplicación en Flutter
Hallasi Saravia, Miguel Angel	Arreglar Aplicación Android
Urrutia Peña, Jasmin Adriana	Arreglar Aplicación Android
Sebastian Alexander Ramos Calagua	Aplicación en Flutter
Luciana Carolina Choquehuanca Nuñez	Aplicación en Flutter

Registro de Versiones del Informe

Versión	Fecha	Autor	Descripción de modificación
TB1	17/09/25	Ruben Velasquez Chambi Christian Fabrizio Inga Orihuela Hallasi Saravia, Miguel Angel Urrutia Peña, Jasmin Adriana Sebastian Alexander Ramos Calagua	Se completaron el Capítulo I y Capítulo II, incluyendo análisis de problemática, entrevistas y arquitectura inicial.
TP1	08/10/2025	Ruben Velasquez Chambi Christian Fabrizio Inga Orihuela Hallasi Saravia, Miguel Angel Urrutia Peña, Jasmin Adriana Sebastian Alexander Ramos Calagua	Se integraron los resultados de entrevistas, análisis competitivo y se definieron los primeros Bounded Contexts.
TB2	13/11/2025	Ruben Velasquez Chambi Christian Fabrizio Inga Orihuela Hallasi Saravia, Miguel Angel Urrutia Peña, Jasmin Adriana Sebastian Alexander Ramos Calagua	Se termino la Landing Page, el Backend y la Aplicacion Android.
TF	04/12/2025	Ruben Velasquez Chambi Christian Fabrizio Inga Orihuela Hallasi Saravia, Miguel Angel Urrutia Peña, Jasmin Adriana Sebastian Alexander Ramos Calagua Luciana Carolina Choquehuanca Nuñez	Se termino la aplicacion android y flutter.

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
 - 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. Ubiquitous Language
 - 2.4. Requirements specification
 - 2.4.1. User Stories
 - 2.4.2. Impact Mapping
 - 2.4.3. Product Backlog
 - 2.5. Strategic-Level Domain-Driven Design
 - 2.5.1. EventStorming
 - 2.5.1.1. Candidate Context Discovery
 - 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: Contacts
 - 2.6.1.1. Domain Layer
 - 2.6.1.2. Interface Layer
 - 2.6.1.3. Application 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
 - 2.6.1.6.2. Bounded Context Database Design Diagram
 - 2.6.2. Bounded Context: Product
 - 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
 - 2.6.3.1. Domain Layer
 - 2.6.3.2. Interface Layer
 - 2.6.3.3. Application 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
 - 2.6.3.6.2. Bounded Context Database Design Diagram
 - 2.6.4. Bounded Context: Sales

- 2.6.4.1. Domain Layer
- 2.6.4.2. Interface Layer
- 2.6.4.3. Application 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
 - 2.6.4.6.2. Bounded Context Database Design Diagram
- 2.6.5. Bounded Context: Finances
 - 2.6.5.1. Domain Layer
 - 2.6.5.2. Interface Layer
 - 2.6.5.3. Application 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
 - 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

- Conclusiones y recomendaciones
- Video App Validation
- Video About the product
- Video About the team
- Glosario
- Bibliografía
- Anexos

Student Outcome

ABET 7:

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

Este proyecto permitió aplicar el Student Outcome ABET 7. A lo largo del desarrollo se investigaron metodologías ágiles, arquitectura limpia y herramientas de integración, aplicándolas en la práctica para resolver problemas reales.

Criterio específico	Acciones realizadas	Conclusiones
7.c1. Actualizan conceptos y conocimientos necesarios para su desarrollo profesional y en especial para su proyecto en soluciones de software.	<p>Ruben Genaro Velasquez Chambi</p> <p>TB1</p> <p>Me encargué de la elaboración de los User Stories, así como de las Spike Stories para investigar las bibliotecas y tecnologías que se podrían utilizar en el proyecto. Además, estructuré el ProductBacklog y participé en la creación del diagrama C4 para definir los niveles de arquitectura del sistema.</p> <p>TP</p> <p>En esta entrega me encargué de registrar el Sprint Planning correspondiente al Sprint 1, definiendo las tareas, prioridades y objetivos a cumplir dentro del equipo. Además, realicé correcciones en el diagrama C4, mejorando la claridad en la estructura de los componentes. También desarrollé la configuración inicial del proyecto backend y móvil, asegurando la correcta integración de dependencias y estructura base del proyecto.</p> <p>TB2</p> <p>En este entregable culminé el desarrollo del backend, asegurando su integración con el frontend. Además, participé en la elaboración del Sprint Planning y Sprint Backlog del Sprint 2, coordinando las tareas del equipo y verificando el cumplimiento de objetivos técnicos. Esta experiencia reforzó mis conocimientos en organización de proyectos ágiles y desarrollo de servicios escalables.</p> <p>TF</p> <p>Durante este sprint reforcé y actualicé mis conocimientos relacionados con la documentación técnica y la redacción formal de informes de desarrollo de software. Esto incluyó mejorar la estructura, coherencia y claridad del documento del proyecto, aplicar buenas prácticas de documentación ágil y corregir inconsistencias</p> <p>Jasmin Adriana Urrutia Peña</p> <p>TB1</p> <p>Para esta entrega, completé tareas como el perfil de la startup, antecedentes, problemática y segmento objetivo,</p>	<p>La elaboración de user stories y spike stories nos permitió actualizar y reforzar nuestros conocimientos sobre metodologías ágiles y la forma en que estas ayudan a definir los requerimientos y las tecnologías adecuadas para el desarrollo de software. Asimismo, trabajar en el product backlog y el diagrama C4 nos ayudó a comprender mejor la importancia de la planificación y la arquitectura en el ciclo de vida del software. La identificación de Bounded Contexts fue fundamental para dividir el sistema en componentes manejables y cohesionados. Cada integrante reforzó habilidades prácticas y teóricas en esta última etapa, logrando integrar todos los módulos y cerrar el ciclo de desarrollo con un producto funcional.</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>lo que me permitió obtener una visión más clara y estructurada del proyecto.</p> <p>TP</p> <p>Para esta entrega realice capítulo 4 de UX design lo cual nos permitió tener una vision clara al momento de realizar el frontend.</p> <p>TB2</p> <p>En este entregable finalicé la aplicación Android, integrando las funcionalidades principales y asegurando su correcto funcionamiento. Además, elaboré las conclusiones y recomendaciones del proyecto, lo que me permitió reflexionar sobre el proceso de desarrollo y consolidar aprendizajes en diseño de interfaces, pruebas y experiencia de usuario.</p> <p>Sebastian Alexander Ramos Calaguia</p> <p>TB1</p> <p>Investigué y definí los Bounded Contexts clave del sistema. Colaboré en el diseño del Context Map para visualizar las relaciones entre los subsistemas.</p> <p>TP</p> <p>Durante esta entrega trabajé en la configuración y documentación técnica del entorno de desarrollo para iCafe, asegurando consistencia entre el backend en Java/Docker y la app móvil en Kotlin. Además, revisé la estructura de gestión de código fuente, convenciones de estilo y configuración de despliegue. Documenté la evidencia de pruebas realizadas mediante Swagger, validando los endpoints desarrollados y alineándolos con los criterios de aceptación definidos.</p> <p>TB2</p> <p>En esta entrega participé en el desarrollo de la aplicación en Flutter y en la corrección del TP1, fortaleciendo mis habilidades en desarrollo multiplataforma y control de versiones. Este proceso me permitió afianzar conocimientos sobre integración entre capas y la importancia de la revisión continua en proyectos colaborativos.</p> <p>Christian Fabrizio Inga Orihuela</p> <p>TB1</p> <p>En esta entrega entrevisté a una persona de nuestro segmento objetivo, de esta forma entendiendo sus necesidades. Por otro lado, el realizar los Domain Message Flows Modeling me ayudó a comprender mejor la forma en que nuestra app funcionará, esto también ayudándome en mi desarrollo profesional.</p> <p>TP</p> <p>En esta entrega desarrollé la parte del frontend de Products, detectando algunos errores de comunicación con el backend, pero que finalmente fueron arreglados con ayuda del grupo y así comprendiendo los errores que pueden haber en el desarrollo de software.</p> <p>TB2</p> <p>Durante esta entrega implementé mejoras en la</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>aplicación Flutter y realicé la corrección del TP1, lo que me ayudó a consolidar mis conocimientos en desarrollo móvil y en la depuración de errores. Este proceso fortaleció mi comprensión sobre la importancia del trabajo en equipo y la revisión iterativa en la entrega de software funcional.</p> <p>Miguel Hallasi Saravia TB1 Participé activamente en la elaboración de los artefactos User Personas, User Task Matrix, User Journey Mapping y Empathy Mapping. Esta experiencia me permitió reforzar conocimientos clave sobre diseño centrado en el usuario, análisis de necesidades y comportamiento, y estructuración de flujos de interacción. Estos conceptos son fundamentales para el desarrollo de soluciones de software efectivas y orientadas al usuario.</p> <p>TP Para esta entrega implementé los Bounded Contexts en el backend utilizando Java, lo cual me permitió aplicar de manera práctica los conceptos de DDD aprendidos anteriormente. Esta implementación reforzó mi comprensión sobre la separación de responsabilidades y la importancia de mantener una arquitectura limpia y escalable en proyectos de software.</p> <p>TB2 En esta fase culminé el desarrollo del backend, completando las funcionalidades planificadas en el Sprint 2. Este trabajo me permitió aplicar conceptos avanzados de arquitectura y consolidar mi experiencia en desarrollo de APIs y control de versiones. Además, fortalecí mi capacidad para coordinar entregas técnicas dentro de un entorno ágil.</p> <p>Luciana Carolina Choquehuanca Núñez TF Durante el trabajo final me encargué del diseño e implementación de las pantallas en Flutter para la gestión de empleados y la gestión de proveedores, definiendo los flujos de registro, edición y listado. Este trabajo me permitió actualizar y reforzar conceptos de desarrollo móvil, como la composición de widgets, manejo de formularios y validaciones, navegación entre pantallas e integración con servicios del backend. Gracias a ello, consolidé mis conocimientos en interfaces orientadas a la usabilidad y alineadas con los requerimientos funcionales del sistema, contribuyendo directamente a la solución de software desarrollada por el equipo.</p>	
7.c2. Reconocen la necesidad del aprendizaje permanente profesional y el desarrollo de proyectos en	<p>Ruben Genaro Velasquez Chambi TB1 En este proyecto aprendí que al momento de investigar tecnologías mediante las Spike Stories, es fundamental estar en constante actualización para seleccionar las bibliotecas más adecuadas. También reconocí la importancia de aplicar herramientas de análisis y diseño</p>	La experiencia con la investigación de bibliotecas y el diseño arquitectónico refuerza la idea de que un profesional debe mantenerse actualizado y ser capaz de aplicar nuevas técnicas y herramientas en el desarrollo de soluciones de software. En el TB2, cada integrante fortaleció su compromiso con el aprendizaje continuo , evidenciando una evolución técnica y metodológica a lo largo del proyecto.

Criterio específico	Acciones realizadas	Conclusiones
<p>soluciones de software.</p>	<p>como el C4 Model, que requieren un aprendizaje continuo para mejorar las habilidades de diseño arquitectónico.</p> <p>TP</p> <p>Durante esta entrega reconocí la importancia del aprendizaje continuo al trabajar en la configuración y desarrollo del bounded context "Contacts" en el backend. Implementar este módulo me permitió profundizar en conceptos de arquitectura orientada al dominio (DDD), entendiendo cómo estructurar entidades, servicios y repositorios de forma coherente con los principios de diseño de software. Asimismo, el registro del sprint 1 me ayudó a comprender la importancia de la documentación y la mejora en el proceso de desarrollo.</p> <p>TB2</p> <p>Durante esta entrega consolidé mis conocimientos en gestión ágil y backend avanzado, comprendiendo la importancia del aprendizaje continuo para mantener la calidad y escalabilidad del sistema. Cada reto técnico me permitió mejorar mi capacidad de resolución de problemas y mi criterio como desarrollador.</p> <p>TF</p> <p>El trabajo realizado en este sprint me permitió reconocer la importancia del aprendizaje continuo dentro del desarrollo de software. Las tareas de redacción, revisión y mejora del informe evidenciaron que siempre existen oportunidades para perfeccionar la documentación, adoptar mejores metodologías y refinar la comunicación del proyecto.</p> <p>Jasmin Adriana Urrutia Peña</p> <p>TB1</p> <p>Realicé el Event Storming lo que me ayudó a darme cuenta de la importancia que tiene saber reconocer los bounded context ya que es importante para un buen inicio de trabajo.</p> <p>TP</p> <p>Para esta entrega ayudé a implementar el registro de proveedores, empleados e insumos en el frontend, lo que me ayudó a practicar.</p> <p>TB2</p> <p>En esta entrega final aprendí la importancia de la constancia y mejora continua al culminar la aplicación Android y elaborar las conclusiones del proyecto. Esta experiencia me ayudó a fortalecer mis habilidades técnicas, reflexivas y comunicativas, esenciales en el desarrollo profesional dentro del área de software.</p> <p>Sebastian Alexander Ramos Calagua</p> <p>TB1</p> <p>Analicé las interacciones entre Bounded Contexts para garantizar la consistencia y escalabilidad del sistema. Estudié patrones de diseño DDD (Domain-Driven Design) para aplicar en la arquitectura del proyecto.</p> <p>TP</p> <p>La configuración del entorno de desarrollo y la documentación técnica me permitieron consolidar conocimientos sobre control de versiones, despliegue y</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>buenas prácticas de arquitectura. La integración de Swagger como herramienta de validación fortaleció mi comprensión sobre pruebas de integración y documentación de APIs.</p> <p>TB2</p> <p>En esta etapa final reconocí el valor de la revisión continua y el aprendizaje autónomo al trabajar en la aplicación Flutter. La necesidad de adaptarse a distintos entornos y lenguajes reafirmó mi compromiso con la formación profesional constante en el desarrollo multiplataforma.</p> <p>Christian Fabrizio Inga Orihuela</p> <p>TB1</p> <p>Participé en el diseño de las entrevistas y en la realización de una de ellas, lo que me permitió comprender mejor las necesidades reales de los usuarios finales. Además, colaboré en el Domain Message Flows Modeling y en la elaboración de diagramas arquitectónicos del sistema, fortaleciendo mis conocimientos en DDD.</p> <p>TP</p> <p>El desarrollar el bounded context de Products en el frontend me hizo comprender diferentes aspectos del desarrollo de aplicaciones, así como aplicar conocimientos aprendidos en clase, lo cual a su vez me ayudará en mi futuro profesional.</p> <p>TB2</p> <p>En esta entrega comprendí la importancia de seguir aprendiendo nuevas herramientas y metodologías, especialmente al trabajar en Flutter y en la revisión del TP1. Este proceso reforzó mi capacidad de análisis y adaptación frente a los desafíos técnicos del desarrollo de software.</p> <p>Miguel Hallasi Saravia</p> <p>TB1</p> <p>Colaboré en el desarrollo del Bounded Context: Finances, aplicando principios de Domain-Driven Design (DDD) para delimitar responsabilidades y cohesión dentro del sistema.</p> <p>TP</p> <p>Para esta entrega implementé los Bounded Contexts en el backend utilizando Java, lo cual me permitió aplicar de manera práctica los conceptos de DDD aprendidos anteriormente.</p> <p>TB2</p> <p>Durante esta fase reafirmé la importancia del aprendizaje permanente al culminar la implementación del backend en el Sprint 2. Cada iteración fue una oportunidad para optimizar procesos y aplicar mejores prácticas de programación, consolidando mis competencias técnicas y profesionales.</p> <p>Luciana Carolina Choquehuanca Núñez</p> <p>TF</p>	

Criterio específico	Acciones realizadas	Conclusiones
	Durante el trabajo final comprendí la importancia del aprendizaje permanente al implementar las pantallas de gestión de empleados y proveedores en Flutter. Para lograrlo, tuve que revisar documentación oficial, ejemplos y buenas prácticas de diseño de interfaces, así como adaptar componentes a las necesidades específicas del proyecto. Este proceso me mostró que el desarrollo móvil exige estar en constante actualización y me motivó a seguir profundizando en Flutter y en patrones de diseño de UI para futuros proyectos de soluciones de software.	

Miembro del Equipo	Objetivo SMART 1	Objetivo SMART 2
Ruben Velasquez Chambi	En los próximos 12 meses, conseguir una posición de practicante en una empresa de software que utilice tecnologías móviles (Kotlin,java, Flutter o Jetpack Compose). Para lograrlo, construiré un portafolio con al menos 3 aplicaciones móviles publicadas en GitHub	En los 18 meses posteriores a mi graduación, completar una certificación en pruebas de software (como ISTQB Foundation Level) dedicando al menos 5 horas semanales de estudio y práctica en plataformas de testing. Con ello, fortaleceré mis conocimientos en aseguramiento de calidad y aumentaré mis oportunidades para aplicar a puestos QA dentro del área de desarrollo de software.
Christian Fabrizio Inga Orihuela	En los próximos 12 meses, desarrollar al menos 3 módulos funcionales para aplicaciones móviles utilizando Flutter y Kotlin, publicando el código en GitHub y aplicando principios de arquitectura limpia, para fortalecer mi portafolio como desarrollador móvil.	Para la finalización del presente ciclo e inicios de verano, conseguir una posición de prácticas preprofesionales como desarrollador móvil en una empresa tecnológica, enviando al menos 15 postulaciones y asistiendo a 3 eventos de networking, mientras dedico 8 horas semanales a reforzar mis conocimientos técnicos.
Hallasi Saravia, Miguel Angel	Durante este ciclo me comprometo a dominar los fundamentos del desarrollo móvil utilizando Flutter y Kotlin	Para finales del ciclo 2025-2, implementaré un proyecto móvil completo que integre autenticación, base de datos y al menos 3 funcionalidades principales, documentando el proceso.
Urrutia Peña, Jasmin Adriana	En el próximo año, quiero comenzar mis prácticas por lo que obtendré una posición de practicante en una empresa de desarrollo frontend que utilice tecnologías como React.js o JavaScript avanzado.	Conseguir un puesto de trabajo como desarrollador frontend junior en una empresa tecnológica, aplicando a al menos 20 vacantes y asistiendo a 5 eventos de networking en los primeros 9 meses tras mi graduación en diciembre de 2026.
Sebastian Alexander Ramos Calagua	Desarrollar y publicar el módulo de gestión de inventario de iCafe en 6 meses, permitiendo el registro y tracking de insumos en tiempo real, con al menos 3 cafeterías piloto usando la funcionalidad.	Obtener la certificación AWS Cloud Practitioner en los próximos 4 meses, dedicando 6 horas semanales al estudio, para mejorar la infraestructura técnica de SyncCafe con soluciones escalables en la nube.
Luciana Carolina Choquehuanca Nuñez	En los próximos 12 meses, diseñar y desarrollar al menos 3 interfaces completas de aplicaciones móviles (pantallas de inicio de sesión, dashboard y detalle) utilizando Flutter y buenas prácticas de UX/UI, publicando los resultados en mi portafolio de GitHub y validándolos con mínimo 5 usuarios cada uno mediante breves encuestas.	En los próximos 18 meses, preparar y aprobar una certificación introductoria en cloud computing (como AWS Cloud Practitioner o Azure Fundamentals), dedicando 5 horas semanales de estudio, realizando 3 simulacros de examen y aplicando estos conocimientos en al menos un proyecto académico o personal.

Capítulo I: Presentación

1.1. Startup Profile

1.1.1. Descripción de la Startup

Nuestra startup se llama SyncCafe y surge con el propósito de transformar la manera en que las cafeterías administran sus operaciones diarias. Actualmente, muchos de estos negocios todavía dependen de registros manuales o herramientas dispersas como libretas, hojas de cálculo o aplicaciones poco integradas, lo que genera desorganización, pérdida de información y costos innecesarios que afectan directamente la rentabilidad y el crecimiento. Para responder a esta problemática, desarrollamos iCafe, una plataforma digital diseñada para que los dueños y administradores de cafeterías puedan centralizar sus procesos en un solo lugar, controlando costos de manera más precisa, registrando proveedores y empleados y gestionando inventarios. Nuestro objetivo es simplificar la gestión sin necesidad de conocimientos técnicos avanzados, ofreciendo una solución clara, eficiente y accesible que permita a los propietarios enfocarse en lo más importante: brindar una mejor experiencia a sus clientes, asegurar la calidad de sus productos y hacer crecer su negocio.



1.1.2. Perfiles de integrantes del equipo

A continuación se presentan los perfiles de los integrantes del equipo de desarrollo de SyncCafe. Cada miembro aporta habilidades complementarias en diseño, desarrollo, análisis y arquitectura de software, fortaleciendo el enfoque multidisciplinario del proyecto iCafe.

Foto	Información
A portrait photo of Christian Inga Orihuela, a young man with dark hair and glasses, wearing a red shirt.	Nombres y apellidos: Christian Inga Orihuela Código: 20231C111 Carrera: Ingeniería de Software Descripción: Estudiante del sexto ciclo con conocimientos en C++, Python y JavaScript. Poseo capacidad de adaptación a distintos entornos de trabajo y me considero una persona responsable y confiable.
A portrait photo of Miguel Hallasi Saravia, a young man with dark hair and glasses, wearing a grey shirt.	Nombres y apellidos: Miguel Hallasi Saravia Código: U202312391 Carrera: Ingeniería de Software Descripción: Estudiante del sexto ciclo, motivado por el aprendizaje continuo y la adquisición de nuevas experiencias en desarrollo móvil, diseño de interfaces y trabajo colaborativo.
A portrait photo of Rubén Genaro Velásquez Chambi, a young man with dark hair and glasses, wearing a black hoodie.	Nombres y apellidos: Rubén Genaro Velásquez Chambi Código: U202117342 Carrera: Ingeniería de Software Descripción: Estudiante del séptimo ciclo con experiencia en C++, Node.js, TypeScript, Java, Angular y Vue.js. Me especializo en eficiencia de equipo, arquitectura limpia y desarrollo bajo principios SOLID.
A portrait photo of Jasmin Adriana Urrutia Peña, a young woman with long dark hair, wearing a light-colored top.	Nombres y apellidos: Jasmin Adriana Urrutia Peña Código: U202310008 Carrera: Ingeniería de Software Descripción: Estudiante con enfoque en frontend y diseño UX. Me considero responsable, creativa y empática, con conocimientos en C++, HTML y CSS. Valoro el trabajo en equipo y el aprendizaje compartido.

Foto	Información
	<p>Nombres y apellidos: Sebastian Alexander Ramos Calagua Código: U202222846 Carrera: Ingeniería de Software Descripción: Desarrollador full-stack con experiencia en arquitectura modular. Me apasiona aplicar tecnología para resolver desafíos sociales en educación, salud y justicia, priorizando claridad, escalabilidad y utilidad.</p>
	<p>Nombres y apellidos: Luciana Carolina Choquehuanca Nuñez Código: U202319431 Carrera: Ingeniería de Software Descripción: Soy Luciana, tengo 20 años y actualmente curso el sexto ciclo de Ingeniería de Software. Me entusiasman los proyectos innovadores que buscan transformar la experiencia del usuario en diversos entornos. Disfruto participar en propuestas desafiantes, especialmente aquellas que combinan creatividad, diseño centrado en el usuario y un impacto tangible en la vida real.</p>

1.2.1. Antecedentes y problemática

En esta sección se analiza el contexto actual de las cafeterías en relación con la gestión operativa, identificando las principales dificultades que enfrentan y justificando la necesidad de una solución tecnológica adaptada a sus características. Este análisis fundamenta el desarrollo del producto iCafe como respuesta a una problemática concreta del sector.

Descripción de la problemática

Actualmente, muchas cafeterías gestionan sus operaciones de forma manual o mediante herramientas poco integradas, como hojas de cálculo, libretas físicas o aplicaciones aisladas. Este enfoque genera desorganización, errores humanos, falta de trazabilidad y escaso control sobre los costos e insumos. Los administradores enfrentan dificultades para acceder a información clara y en tiempo real, lo que se traduce en pérdidas económicas, sobrecarga de trabajo y baja productividad.

Si bien existen sistemas de gestión empresarial en el mercado, la mayoría están orientados a grandes compañías, presentan costos elevados o requieren conocimientos técnicos avanzados. Estas soluciones no se ajustan a la dinámica de una cafetería pequeña o mediana, que demanda rapidez, simplicidad y accesibilidad.

Según el estudio de Gómez et al. (2022) publicado por IEEE, más del 65% de microempresas en el sector gastronómico en Latinoamérica aún utilizan métodos manuales para la administración de inventarios y ventas, lo que impacta directamente en su rentabilidad y capacidad de crecimiento.

En este contexto, surge la necesidad de una herramienta digital centralizada, intuitiva y económica que permita a los administradores de cafeterías gestionar sus operaciones de manera más eficiente. Una solución que simplifique el control de costos, facilite el registro de proveedores y empleados, y organice los inventarios. Con este propósito nace iCafe, un producto desarrollado por la startup SyncCafe, cuyo enfoque es brindar un sistema moderno y adaptado a las necesidades reales de cafeterías independientes y cadenas pequeñas.

Aplicación de la metodología 5W y 2H

-What (¿Qué?)

Las cafeterías enfrentan problemas de gestión por depender de procesos manuales y dispersos. Esto ocasiona errores en el control de costos, inventarios y personal, lo que afecta directamente la rentabilidad del negocio.

-Who (¿Quién?)

Los principales afectados son los dueños y administradores de cafeterías, así como los empleados que deben trabajar sin lineamientos claros en cuanto a recetas o insumos disponibles.

-Where (¿Dónde?)

Este problema se presenta en cafeterías independientes, cadenas pequeñas y medianas, así como en cafeterías institucionales dentro de universidades o empresas que carecen de herramientas digitales adaptadas a sus necesidades.

-When (¿Cuándo?)

La problemática se da todos los días en las actividades básicas de una cafetería: compras de insumos, registro de proveedores, control de inventarios y organización de personal.

-Why (¿Por qué?)

La falta de digitalización y de un sistema centralizado obliga a llevar los procesos manualmente. Esto no solo consume tiempo, sino que también aumenta la posibilidad de errores, genera desorden y dificulta la toma de decisiones basadas en datos.

-How (¿Cómo?)

Actualmente los administradores utilizan métodos tradicionales como libretas, hojas sueltas o documentos de Excel, lo que fragmenta la información. Esto ocasiona duplicidad de tareas, pérdida de registros y falta de visibilidad sobre los recursos disponibles.

-How much (¿Cuánto?)

Las pérdidas varían según el tamaño de la cafetería, pero en muchos casos pueden representar entre el 10% y 20% de los ingresos mensuales, ya sea por compras innecesarias, desperdicio de insumos o falta de control en costos. Una solución digital adaptada a este sector permitiría reducir significativamente estos porcentajes, generando mayor eficiencia y rentabilidad.

1.2.2. Lean UX Process

1.2.2.1. Lean UX Problem Statements

En el contexto actual, los administradores de cafeterías enfrentan dificultades para gestionar sus operaciones de forma eficiente. La falta de herramientas digitales integradas genera desorganización, pérdida de información y errores humanos que afectan la rentabilidad del negocio.

Hemos identificado que el principal problema radica en la ausencia de una solución móvil que permita gestionar inventarios, proveedores, empleados y recetas de forma centralizada y accesible desde cualquier lugar.

¿Cómo podemos ayudar a los administradores de cafeterías a gestionar sus operaciones diarias de forma más eficiente y ordenada, utilizando una aplicación móvil intuitiva y adaptada a sus necesidades reales?

1.2.2.2. Lean UX Assumptions

¿Quiénes son nuestros usuarios?

Dueños y administradores de cafeterías pequeñas y medianas que buscan mejorar la gestión de sus negocios sin depender de herramientas complejas o costosas.

¿Dónde encaja nuestro servicio en su vida?

iCafe se integra en la rutina diaria de los administradores, permitiéndoles registrar compras, controlar inventarios, gestionar personal y compartir recetas desde su dispositivo móvil, en tiempo real.

¿Qué problemas podría tener nuestro producto?

- Resistencia al cambio por parte de usuarios acostumbrados a métodos tradicionales.
- Dificultad para adaptar la interfaz a distintos niveles de experiencia tecnológica.
- Necesidad de sincronización entre dispositivos si hay más de un administrador.

¿Cómo y cuándo se usa nuestro producto?

Se utiliza durante las operaciones diarias de la cafetería: al recibir insumos, registrar proveedores, actualizar inventarios, capacitar al personal y revisar recetas. El acceso móvil permite su uso en cualquier momento, incluso fuera del local.

¿Qué características son importantes?

- Interfaz simple y amigable.
- Registro rápido de insumos, proveedores y empleados.
- Visualización clara de costos e inventarios.
- Compartir recetas estandarizadas con el equipo.
- Seguridad y respaldo de datos.

¿Cómo debe verse y comportarse nuestro producto?

Debe ser visualmente limpio, con navegación intuitiva, accesible desde dispositivos Android e iOS, y con funcionalidades que se adapten al flujo de trabajo de una cafetería.

1.2.2.3. Lean UX Hypothesis Statements

Business Hypothesis

Creemos que al ofrecer una aplicación móvil especializada en la gestión de cafeterías, podremos posicionar a SyncCafe como una solución líder en el sector gastronómico independiente.

Sabremos que hemos logrado nuestra visión

Cuando observemos que iCafe sea adoptada por cafeterías en distintas regiones del país, con una base de usuarios activa y creciente.

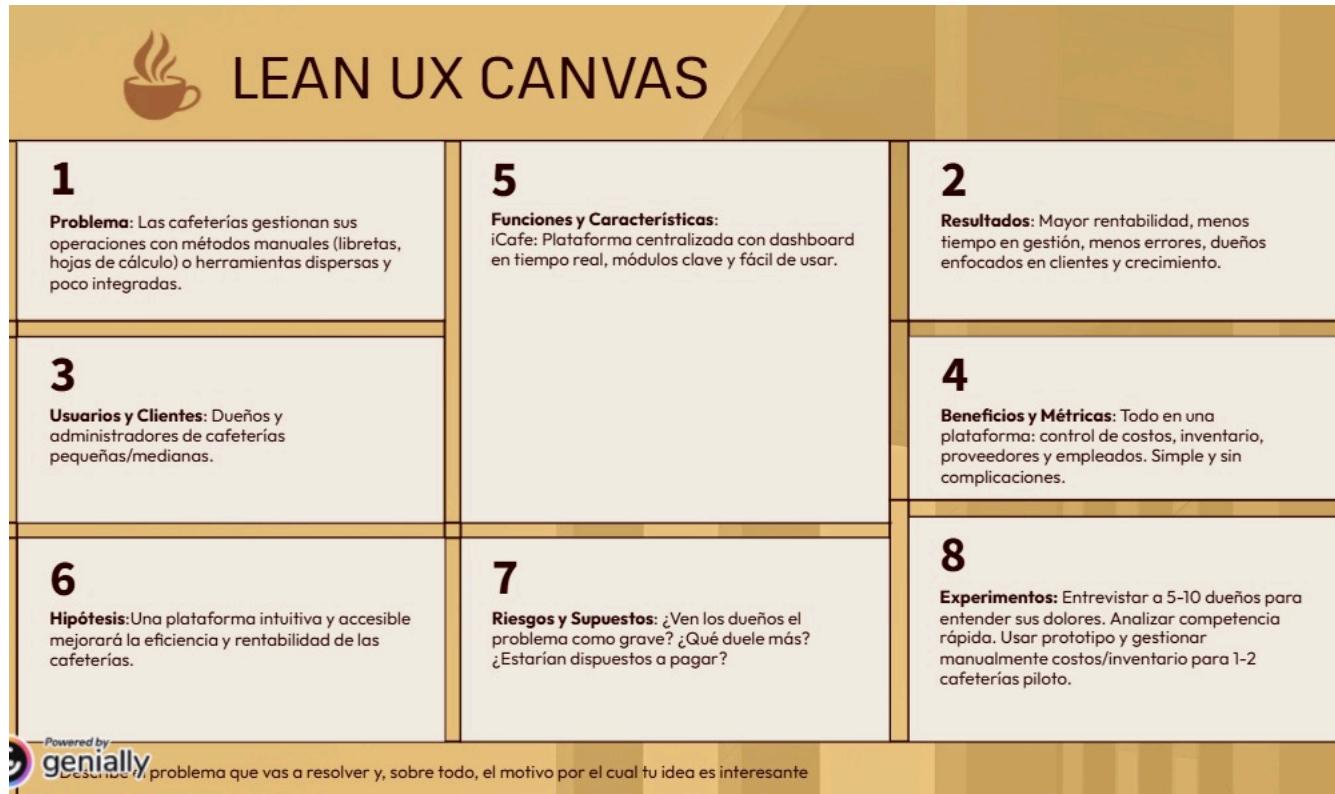
User Hypothesis

Creemos que los administradores de cafeterías valorarán una herramienta móvil que les permita gestionar sus operaciones de forma rápida, ordenada y sin necesidad de conocimientos técnicos avanzados.

Sabremos que hemos logrado satisfacer sus necesidades

Cuando los usuarios reporten una mejora en la organización del negocio, reducción de errores y mayor control sobre sus costos e inventarios.

1.2.2.4. Lean UX Canvas



1.3. Segmentos objetivo

Segmento 1: Dueños de Cafeterías

Personas que cuentan con un emprendimiento de cafeterías, en su mayoría pequeños y medianos negocios, y que administran directamente las operaciones del día a día. Se trata de propietarios que buscan mejorar la organización interna de su local y optimizar el uso de recursos, pero que aún dependen de registros manuales o herramientas poco integradas. Generalmente son emprendedores jóvenes y adultos, entre 25 y 45 años, con un nivel básico de familiaridad con soluciones digitales, interesados en reducir pérdidas, controlar costos e inventarios, y gestionar de manera más eficiente a sus proveedores y personal.

Capítulo II: Requirements Development and Software Solution Design

2.1. Competidores

Plataforma SaaS enfocada en control de inventarios y costos (COGS) con gestión de proveedores y reportes avanzados. Es robusta y escalable, ideal para negocios con mayor complejidad operativa. Para el segmento de cafeterías de 1–3 locales puede resultar costosa/excesiva, pero marca el estándar en costeo y control.

Stock & Inventory Simple

App Android de control de stock ligera: alta/baja de productos, cantidades, costos básicos, alertas y exportaciones. Su propuesta es la simplicidad y la rápida adopción, pero sin profundidad en métricas de costos ni analítica para la toma de decisiones del dueño.

Cafe Manager

App móvil orientada a operaciones diarias (ventas/pedidos) con inventario simple. Es rápida y accesible, pero no prioriza la gestión interna (métricas de desperdicio, análisis de costos, precios sugeridos), por lo que se alinea parcialmente con el enfoque “solo dueño”.

2.1.1. Análisis competitivo

Competitive Analysis Landscape					
¿Por qué llevar a cabo este análisis?	Identificar el posicionamiento de iCafe frente a software cloud/móvil que resuelve (total o parcialmente) la gestión interna de cafeterías: inventario, costos y métricas. Este análisis permite definir foco de valor (simplicidad móvil, costo bajo, enfoque cafeterías de 1–3 locales, decisiones rápidas con datos) y tácticas de entrada.				
		iCafe (SyncCafe)	MarketMan	Stock & Inventory Simple	Cafe Manager
Perfil	Overview	App móvil (Android → Flutter) centrada en el dueño: recetas, inventario, métricas de desperdicio y gestor de costos con reportes PDF.	Plataforma cloud para inventarios, costos y proveedores, con módulos avanzados de control (más robusta y premium).	App Android genérica de control de stock y costos básicos; muy ligera y simple.	App móvil enfocada en operaciones básicas (ventas/pedidos) con inventario simple; menos alineada a gestión interna pura.
	Ventaja competitiva	Mobile-first, simplicidad, enfoque cafetero, métricas de desperdicio, precios sugeridos y PDF listo para compartir.	Costeo/COGS sólido, gestión de proveedores, analítica avanzada y escalabilidad.	Curva de aprendizaje mínima, exportaciones simples, alertas de stock; muy accesible.	Interfaz simple y rápida para operaciones diarias.
	Clientes	Dueños de cafeterías 1–3 locales en zonas urbanas/semiurbanas.	Restaurantes/cafeterías con foco fuerte en control de costos e inventario.	Emprendedores y micro-negocios que requieren control básico de stock.	Dueños que priorizan rapidez operativa y registro de ventas simples.
Perfil de Marketing	Mercado objetivo	Cafeterías independientes y micro-cadenas en latinoamérica.	F&B global que prioriza COGS, inventario y compliance.	Negocios pequeños que buscan control básico de inventario en móvil.	Micro-negocios que requieren operar rápido desde el teléfono.
	Estrategias de marketing	Contenido educativo, demos rápidas, alianzas con proveedores de café.	Casos de éxito, integraciones y ROI explícito en costos/mermas.	ASO/Play Store, reseñas, simplicidad como propuesta de valor.	Visibilidad en tiendas de apps y marketing de boca a boca.
Perfil de Producto	Productos y Servicios	Productos, inventario, métricas de desperdicio,	Inventario, costos, proveedores, reportes	Inventario básico, movimientos,	Ventas/pedidos e inventario simple

		gestor de costos, precios sugeridos y PDF.	avanzados.	costos simples, exportaciones.	(menos foco en métricas/costos).
	Precios y Costos	Aplicación Gratuita	Planes por sede/usuario; inversión mayor que iCafe.	Gratis o muy bajo costo; funciones limitadas.	Muy accesible; alcance funcional reducido.
	Canales de distribución	App Android (Kotlin) y luego Flutter; API REST propia; landing.	Web + app móvil (SaaS).	App móvil Android.	App móvil Android.
ANÁLISIS SWOT	Fortalezas	UX sencilla, mobile-first, enfoque cafeterías, métricas visuales y PDF compatible.	Costeo/COGS sólido, gestión de proveedores, analítica y escalabilidad.	Simplicidad extrema, curva de aprendizaje baja, exportaciones rápidas.	Rapidez operativa para ventas simples; fácil de usar.
	Debilidades	Marca nueva, menos integraciones; sin historial de mercado.	Curva de adopción y costo para micro-negocios.	Limitado en métricas de costos y sin módulos de decisiones.	Menor foco en métricas y costos; menos alineada a gestión interna pura.
	Oportunidades	Digitalización de cafeterías; nicho LATAM poco atendido con mobile simple.	Planes para SMBs y verticalización por segmentos.	Evolucionar a métricas/costos para F&B pequeños.	Añadir módulo de costos para mayor valor al dueño.
	Amenazas	Entrada de suites consolidadas al nicho cafetero; inercia de Excel.	Competencia de apps móviles más baratas y simples.	Desplazamiento por soluciones más completas del sector.	Ser reemplazada por apps con mejor gestión interna.

2.1.2. Estrategias y tácticas frente a competidores

Disponibilidad 100 % gratuita y sin registros extensos: acceso inmediato solo con nombre de negocio.

Interfaz minimalista y clara: colores acorde al tema, pocas pantallas, iconografía directa, prioriza velocidad sobre densidad de funciones.

Visualización de métricas inmediatas: gráficos de stock, consumo, desperdicio y costos en el inicio para dar valor apenas se usan los datos.

Alertas automáticas de gestión: avisos de stock bajo, altos niveles de desperdicio y sobrecostos detectados.

Generación de reportes PDF simples: exportación directa de inventario, costos y recetas para respaldar decisiones administrativas.

Optimización offline: permitir funcionamiento sin conexión para que los dueños puedan registrar datos desde el local sin depender de internet.

2.2. Entrevistas

2.2.1. Diseño de entrevistas

En esta sección se plantearán las preguntas para validar nuestra aplicación y conoces las necesidades de nuestro segmento objetivo.

Preguntas Generales:

¿Cómo te llamas y a qué te dedicas?

Preguntas Específicas:

¿Cómo anotas lo que entra y sale de tus ingredientes en la cafetería?

¿Cada cuánto miras o actualizas tu lista de productos en stock?

¿Te ha pasado quedarte sin algo o tener de más? ¿O tirar comida? ¿Cómo lo arreglas?

¿Te gustaría una app que te muestre si usas bien los ingredientes y cuánto tiras?

¿Sabes cuánto te cuesta de verdad hacer un café? ¿Cómo pones el precio para venderlo?

¿Piensas en gastos como sueldos o renta al poner precios? ¿Cada cuánto chequeas tus gastos totales?

¿Qué usas ahora para organizar tu cafetería? ¿Como Excel, cuadernos o alguna app?

¿Qué usas más para manejar tu negocio: celular, computadora u otro? ¿Del 1 al 5, qué tan fácil te resulta usar apps en el celular?

¿Si tuvieras una pantalla rápida con números clave, qué querrías ver primero? ¿Y qué opción nueva te haría usar una app todos los días?

2.2.2. Registro de entrevistas

Entrevista 1:

Datos del entrevistador:

- Nombre: Inga Orihuela, Christian Fabrizio

Datos del entrevistado:

- Nombre: Jazmin
- Edad: 30 años
- Residencia: Lima
- Enlace: <https://youtu.be/CFkBNErNY6s>



Resumen de la entrevista:

Jazmín es dueña y administradora total de su cafetería. Lleva el control de insumos en un cuaderno y a veces de memoria, revisa el stock solo cuando nota faltantes y ha tenido pérdidas por sobrecompra (frutas) y quiebres de stock (leche, pan). No calcula con exactitud los costos de sus productos, fija precios por referencia a la competencia y revisa sus gastos mensualmente. Usa más el celular que la computadora y se siente cómoda usando apps (4/5). Le gustaría una app que le muestre ventas diarias, stock bajo, alertas de gasto excesivo y uso de ingredientes.

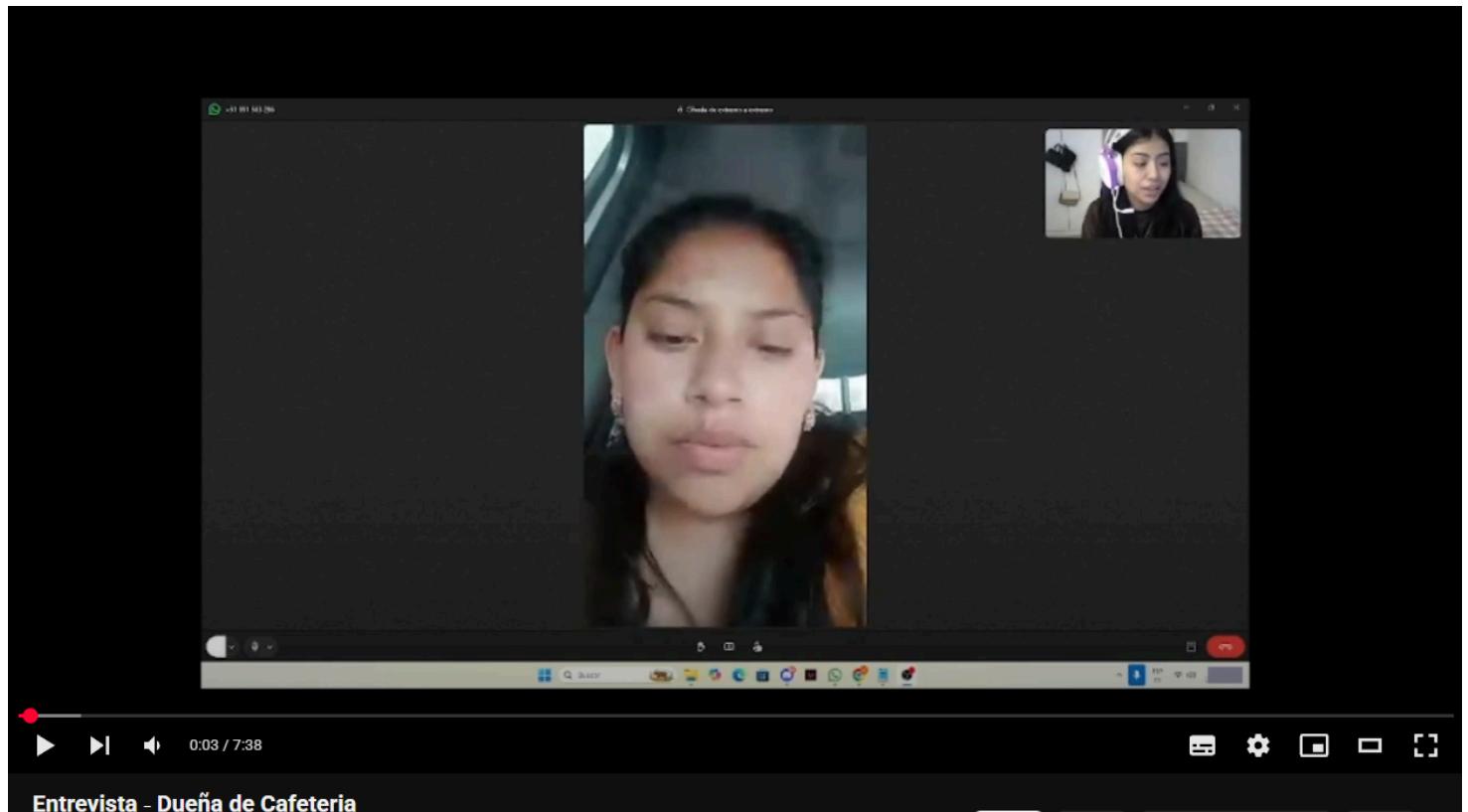
Entrevista 2:

Datos del entrevistador:

- Nombre: Urrutia Peña, Jasmin Adriana

Datos del entrevistado:

- Nombre: Melanie Espinoza
- Edad: 34 años
- Residencia: Huancayo
- Enlace: <https://youtu.be/i0tp6LbB0A>



Resumen de la entrevista:

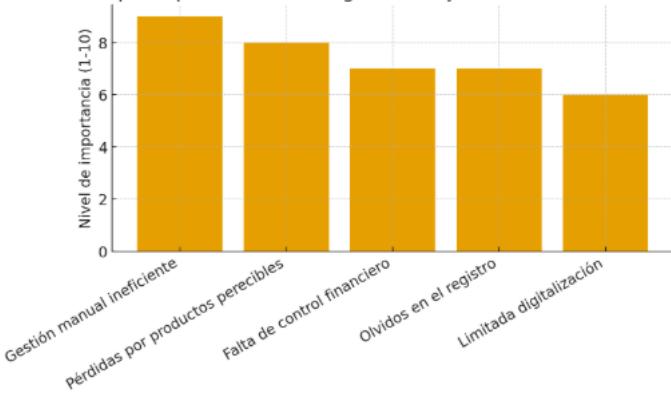
La entrevista con Melanie Espinoza, dueña de la cafetería Fryhna, revela que su principal problema es la falta de organización en la gestión de inventario y costos, ya que actualmente depende de libretas y hojas de Excel que resultan tediosas, poco confiables y propensas a errores u olvidos por parte del personal. Esta situación genera pérdidas económicas debido a la caducidad de productos perecibles como la leche y a la escasez de insumos clave como el café. Además, no cuentan con un control estructurado de gastos fijos ni sueldos, lo que dificulta la planificación financiera. Frente a ello, la aplicación propuesta representa una oportunidad clara al ofrecer una solución móvil, accesible y automatizada que permitiría centralizar la información, generar alertas de stock y vencimiento, facilitar el control de costos y mejorar la rentabilidad y eficiencia del negocio.

2.2.3. Análisis de entrevistas

Análisis del Primer Segmento Objetivo: Dueños de Cafeterías.

Los dueños de cafeterías suelen enfrentar problemas comunes relacionados con la gestión de inventario, ya que muchos todavía dependen de métodos manuales como libretas o archivos de Excel. Esto genera errores, olvidos y pérdidas económicas por productos perecibles o desbalance entre insumos. Asimismo, existe una falta de control financiero estructurado, pues no registran de manera organizada gastos como sueldos, alquileres o costos de insumos. A esto se suma la limitada digitalización, ya que suelen contar con una laptop básica en la tienda, pero no sistemas especializados que faciliten el registro en tiempo real. Pese a ello, muestran alta disposición a adoptar soluciones móviles porque valoran la accesibilidad y simplicidad, especialmente desde sus teléfonos. En resumen, el segmento tiene una necesidad latente de herramientas digitales accesibles que integren inventario, costos y proveedores, con recordatorios y reportes que optimicen su rentabilidad y reduzcan pérdidas.

Principales problemas del segmento objetivo: Dueños de cafeterías



2.3. Needfinding

2.3.1. User Personas

PERSONA: Ramsés Martínez

NAME

Ramsés Martínez

Goals
 Mejorar la eficiencia operativa de la cafetería
 Reducir costos y aumentar la rentabilidad
 Mantener un alto estándar de calidad en productos y servicios

Background
 Javier lleva más de 10 años en el sector de la restauración. Empezó como camarero y desde hace tres años, gestiona "Café del Sol".

Motivations
 El éxito y crecimiento del negocio
 Recibir comentarios positivos de los clientes
 La posibilidad de abrir una nueva sucursal en el futuro

Frustrations
 La falta de un sistema integrado para gestionar inventario, proveedores y empleados
 El tiempo que invierte en tareas manuales y repetitivas

Technology





Browsers


Demographic
 ♂ Male 35 years
 Perú, Lima
 Married
 Administrador de una cafetería mediana

Skills
 Colaborativo

 0 25 50 75 100
 Analítico

 0 25 50 75 100
 Paciente

 0 25 50 75 100

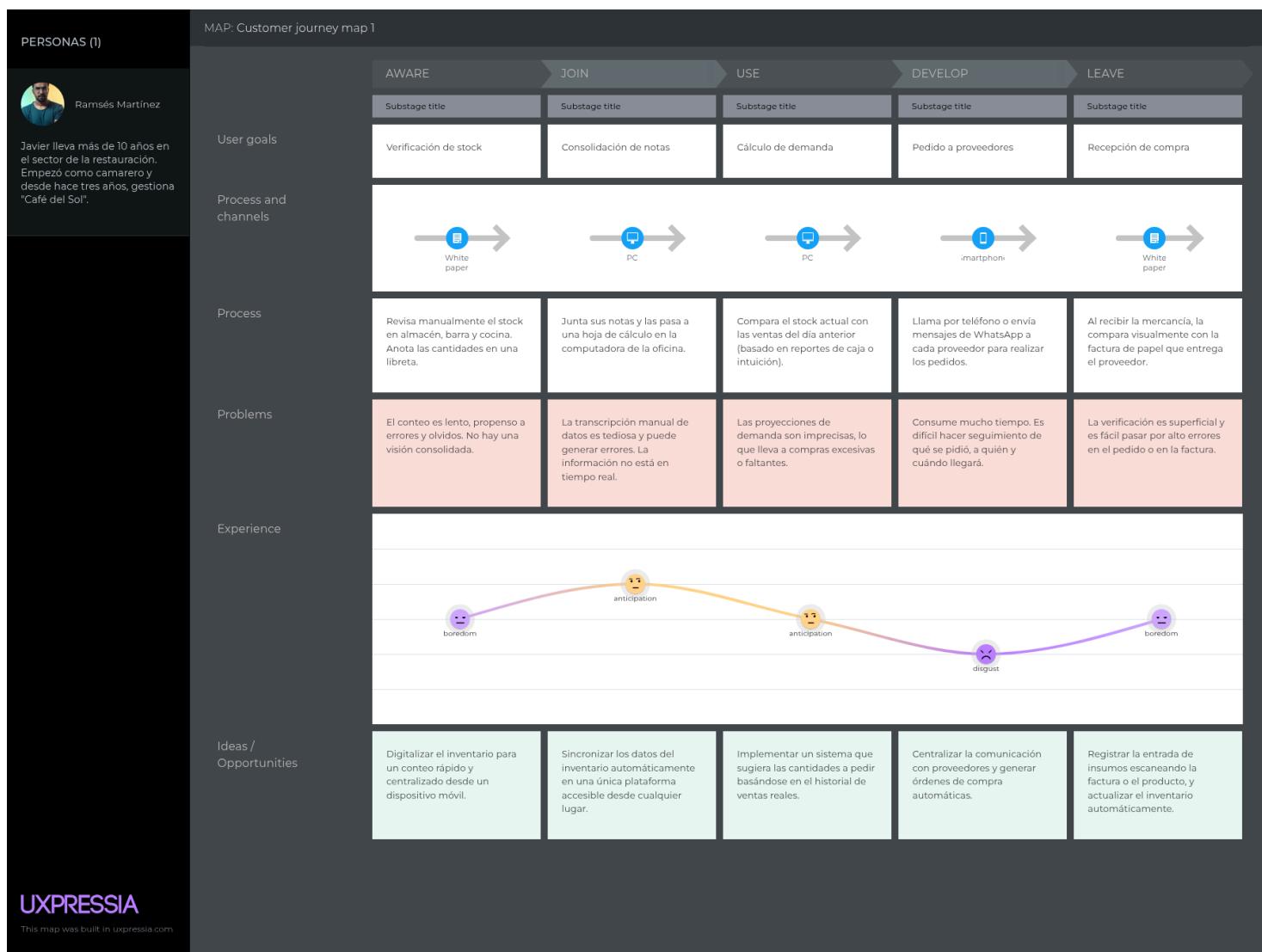
UXPRESSIA
This persona was built in uxpresa.com

2.3.2. User Task Matrix

Tarea (Task)	Frecuencia	Importancia
Gestionar el inventario de insumos	Alta	Alta
Realizar pedidos a proveedores	Media	Alta

Tarea (Task)	Frecuencia	Importancia
Controlar la calidad de los insumos recibidos	Media	Alta
Gestionar el personal (horarios, turnos)	Alta	Alta
Capacitar a nuevos empleados	Baja	Media
Supervisar la preparación de recetas	Alta	Alta
Asegurar la estandarización de recetas	Alta	Alta
Realizar el cuadro de caja diario	Alta	Alta
Analizar las ventas y el rendimiento	Media	Alta
Gestionar la atención al cliente y quejas	Alta	Alta
Mantener la limpieza y orden del local	Alta	Media

2.3.3. User Journey Mapping



2.3.4. Empathy Mapping

PERSONA: Empathy map

1.WHO are we empathizing with?

Administrador de una cafetería mediana en Lima. Busca eficiencia y control para hacer crecer su negocio

7.What do they THINK and FEEL?

“
¿Tendremos suficiente stock para el fin de semana?", "Debo encontrar una forma de reducir las mermas”

2.What do they need to DO?

Necesita gestionar el inventario, supervisar al personal, controlar la calidad de los cafés, realizar pedidos a proveedores y analizar las ventas para asegurar la rentabilidad

6.What do they HEAR?

Quejas de clientes por la demora, comentarios de empleados sobre la desorganización, noticias sobre el alza de precios y sugerencias para modernizar el negocio



3.What do they SEE?

Clientes, facturas en papel, hojas de cálculo, desorden en el almacén, empleados bajo presión y la competencia cercana

5.What do they DO?

Anota pedidos en una libreta, llama a proveedores, supervisa al personal, realiza el cuadro de caja manualmente y atiende las quejas de los clientes.

PAINS

Pérdida de tiempo en tareas manuales y repetitivas.
Errores en la gestión de inventario que generan pérdidas económicas.
Dificultad para tomar decisiones estratégicas por falta de datos fiables.

GAINS

What are their needs, hopes, wants and dreams?
Aumentar la eficiencia, reducir costos y mejorar la rentabilidad.
Disponer de más tiempo para su familia y para planificar el crecimiento del negocio.

4.What do they SAY?

“
"Necesito el reporte de ventas de ayer."
"El proveedor se volvió a retrasar."
"Revisen la receta antes de preparar ese café.”

2.3.5. Ubiquitous Language

Término	Definición
Inventory	Stock de todos los insumos y productos disponibles en la cafetería.
Provider	empresa que abastece de insumos a la cafetería.
Recipe	Instrucciones estandarizadas para la preparación de un producto, garantizando consistencia y calidad.
Product	Artículo final que se ofrece a la venta al cliente.
Employer	Empleado de la cafetería responsable de las operaciones diarias y atención al cliente.
Purchase	Proceso de adquisición de insumos y materiales necesarios para el funcionamiento de la cafetería.
Sales	Transacciones comerciales realizadas con los clientes para la venta de productos y servicios.

2.4. Requirements specification

2.4.1. User Stories

Epics

Epic ID	Título	Descripción
EP001	Autenticación y Roles	Permite el acceso seguro al sistema diferenciando dueños y empleados con permisos específicos.
EP002	Gestión de Empleados	Permite a los dueños crear, editar, eliminar y asignar credenciales a empleados por sucursal.
EP003	Gestión de Proveedores	Administración de proveedores: creación, edición, búsqueda y eliminación.
EP004	Gestión de Recetas y Productos	Administración de recetas, insumos, preparación y vinculación con inventario.
EP005	Gestión de Costos	Registro y clasificación de costos (fijos, variables, directos, indirectos) para calcular la rentabilidad.
EP006	Gestión de Inventario	Control de entradas, salidas, categorías de insumos y búsqueda de inventario en tiempo real.
EP007	Gestión de Precios de Venta	Definición de precios de venta basados en costos e insumos de cada producto.
EP008	Búsquedas Optimizada	Búsqueda rápida de empleados, proveedores e inventario.
EP009	Reportes y Estadísticas	Reportes de costos, ventas, rentabilidad y gráficos
EP010	Funcionalidades para Empleados	Acceso simplificado a recetas, precios y búsqueda de productos.
EP011	Landing Page	Proporcionar un sitio web estático con información clara sobre el producto
EP012	Spike Story	historias de usuario de investigación de recursos para el proyecto

User Stories

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
US001	Registro de dueño	Como dueño, quiero registrarme en la aplicación para administrar mi cafetería.	Escenario 1 – Registro exitoso Given que el dueño ingresa datos válidos When confirma el registro Then el sistema crea su cuenta. Escenario 2 – Email duplicado Given que el email ya existe When intenta registrarse Then el sistema muestra un error. Escenario 3 – Campos faltantes	EP001

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
			Given que no llena todos los campos When intenta registrarse Then el sistema muestra validaciones obligatorias.	
US002	Login seguro	Como usuario, quiero iniciar sesión con mis credenciales para acceder según mi rol.	Escenario 1 – Login exitoso Given credenciales correctas When hace login Then accede al sistema según rol. Escenario 2 – Contraseña incorrecta Given credenciales inválidas When hace login Then el sistema rechaza acceso. Escenario 3 – Bloqueo por intentos fallidos Given múltiples intentos fallidos When excede límite Then la cuenta queda bloqueada temporalmente.	EP001
US003	Listar empleados	Como dueño, quiero ver la lista de empleados para gestionarlos fácilmente.	Escenario 1 – Lista con datos Given empleados existentes When abro listado Then muestra empleados. Escenario 2 – Lista vacía Given no hay empleados When abro listado Then muestra mensaje vacío.	EP002
US004	Actualizar empleado	Como dueño, quiero actualizar datos de empleados para mantener información al día.	Escenario 1 – Actualización exitosa Given datos válidos When actualizo Then se guardan cambios. Escenario 2 – Campos inválidos Given datos inválidos When actualizo Then muestra error.	EP002
US005	Eliminar empleado	Como dueño, quiero eliminar empleados en caso de despido o renuncia.	Escenario 1 – Eliminación exitosa Given empleado válido When elimino Then se elimina de la lista. Escenario 2 – Empleado con registros asociados Given empleado vinculado When elimino Then sistema pide confirmación.	EP002
US006	Crear proveedor	Como dueño, quiero registrar proveedores para gestionar insumos y compras.	Escenario 1 – Registro exitoso Given datos completos When guardo proveedor Then se almacena correctamente. Escenario 2 – Duplicado Given proveedor ya existe When registro Then muestra error. Escenario 3 – Campos obligatorios Given datos incompletos When registro Then muestra validación.	EP003

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
US007	Listar proveedores	Como dueño, quiero listar proveedores para ver a quién comprar.	<p>Escenario 1 – Lista con datos Given proveedores existen When abro listado Then muestra proveedores.</p> <p>Escenario 2 – Lista vacía Given no hay proveedores When abro listado Then muestra mensaje vacío.</p>	EP003
US008	Actualizar proveedor	Como dueño, quiero actualizar datos de proveedores para mantener información al día.	<p>Escenario 1 – Actualización exitosa Given datos válidos When actualizo Then se guardan cambios.</p> <p>Escenario 2 – Campos inválidos Given datos inválidos When actualizo Then muestra error.</p>	EP003
US09	Crear receta	Como dueño, quiero registrar recetas con ingredientes para estandarizar preparación.	<p>Escenario 1 – Registro exitoso Given ingredientes válidos When guardo receta Then se registra correctamente.</p> <p>Escenario 2 – Ingrediente inexistente Given ingrediente no registrado When intento usarlo Then muestra error.</p> <p>Escenario 3 – Sin nombre de receta Given falta nombre When registro Then muestra validación.</p>	EP004
US010	Actualizar receta	Como dueño, quiero actualizar recetas para modificar ingredientes o cantidades.	<p>Escenario 1 – Actualización exitosa Given datos válidos When actualizo receta Then se guardan cambios.</p> <p>Escenario 2 – Campos inválidos Given datos incompletos When actualizo Then muestra error.</p>	EP004
US011	Registrar costos	Como dueño, quiero registrar costos fijos, variables, directos e indirectos para calcular rentabilidad.	<p>Escenario 1 – Registro exitoso Given datos completos When guardo costo Then se registra.</p> <p>Escenario 2 – Tipo inválido Given tipo no válido When guardo Then muestra error.</p> <p>Escenario 3 – Falta monto Given monto vacío When guardo Then muestra validación.</p>	EP005

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
US012	Registrar entrada inventario	Como dueño, quiero registrar entradas de inventario para tener control de insumos.	<p>Escenario 1 – Registro exitoso Given insumo válido When ingreso cantidad Then se actualiza stock.</p> <p>Escenario 2 – Cantidad inválida Given cantidad negativa When ingreso Then sistema rechaza.</p> <p>Escenario 3 – Insumo inexistente Given insumo no registrado When ingreso Then muestra error.</p> <p>Escenario 4 – Insumo duplicado Given insumo ya existe When registro Then muestra error.</p>	EP006
US013	Actualizar inventario	Como dueño, quiero actualizar inventario para reflejar entradas y salidas.	<p>Escenario 1 – Actualización exitosa Given datos válidos When actualizo Then se guardan cambios.</p>	EP006
US014	Definir precio de venta	Como dueño, quiero definir precios de venta basados en costos para maximizar rentabilidad.	<p>Escenario 1 – Precio calculado Given receta registrada When defino precio Then sistema sugiere precio con margen.</p> <p>Escenario 2 – Precio manual Given quiero definir precio When ingreso manual Then sistema lo acepta.</p>	EP007
US015	Búsqueda de empleados	Como dueño, quiero buscar empleados por nombre o correo para localizarlos rápidamente.	<p>Escenario 1 – Coincidencia exacta Given nombre válido When busco Then muestra resultados.</p> <p>Escenario 2 – Parcial Given letras parciales When busco Then lista posibles coincidencias.</p> <p>Escenario 3 – Sin resultados Given no existe When busco Then muestra mensaje vacío.</p>	EP008
US016	Buscar proveedores	Como dueño, quiero buscar proveedores rápidamente para encontrarlos fácilmente.	<p>Escenario 1 – Búsqueda por nombre Given proveedores existen When busco por nombre Then muestra coincidencias.</p> <p>Escenario 2 – Sin resultados Given no hay coincidencias When busco Then muestra mensaje vacío.</p>	EP008

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
US017	Buscar inventario	Como dueño, quiero buscar inventario por nombre o categoría para localizar insumos.	<p>Escenario 1 – Búsqueda exitosa Given insumos existen When busco por categoría Then muestra coincidencias.</p> <p>Escenario 2 – Sin resultados Given no hay coincidencias When busco Then muestra mensaje vacío.</p>	EP008
US018	Reporte de costos	Como dueño, quiero generar reportes de costos por sucursal para analizar rentabilidad.	<p>Escenario 1 – Reporte exitoso Given costos registrados When genero reporte Then sistema muestra tabla.</p> <p>Escenario 2 – Sin datos Given no hay costos When genero Then muestra mensaje vacío.</p> <p>Escenario 3 – Multisucursal Given dueño con varias sucursales When selecciona Then genera reporte específico.</p>	EP009
US019	Ver recetas	Como empleado, quiero consultar recetas estandarizadas para preparar correctamente.	<p>Escenario 1 – Receta existente Given receta válida When consulto Then muestra ingredientes y pasos.</p> <p>Escenario 2 – Filtrar recetas Given múltiples recetas When busco por nombre Then lista filtrada.</p> <p>Escenario 3 – Receta no encontrada Given receta no existe When consulto Then muestra error.</p>	EP010
US020	Ver sección Home	Como visitante (dueño de cafetería) , quiero ver una introducción clara con imagen destacada y texto breve.	<p>Escenario 1 – Carga exitosa Given que ingreso a la página When se carga el sitio Then visualizo logo, imagen principal y mensaje de bienvenida.</p> <p>Escenario 2 – Fallo de carga Given error When cargo sitio Then muestra mensaje de error.</p>	EP011
US021	Ver sección About	Como visitante , quiero ver una explicación sobre la aplicación y su propósito, para entender su valor.	<p>Escenario 1 – Visualización exitosa Given que estoy en la landing When hago scroll Then veo la sección "Sobre la aplicación" con texto e imágenes.</p> <p>Escenario 2 – Sección vacía Given falta contenido</p>	EP011

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
			When ingreso Then muestra mensaje temporal.	
US022	Ver sección Features	Como visitante , quiero conocer las funcionalidades principales de la app, para evaluar si cubre mis necesidades.	Escenario 1 – Lista de funcionalidades Given que estoy en landing When navego a “Features” Then visualizo lista de funcionalidades. Escenario 2 – Sin funcionalidades Given no cargan When ingreso Then muestra mensaje temporal.	EP011
US023	Botón Call to Action (CTA)	Como visitante , quiero un botón visible para descargar la app móvil, para obtener acceso fácilmente.	Escenario 1 – Botón visible Given que estoy en landing When veo CTA Then aparece botón “Descargar App”. Escenario 2 – Click CTA Given que hago click When interactúo Then me redirige a Google Play/App Store.	EP011
US024	Ver sección Contact	Como visitante , quiero ver información de contacto, para poder comunicarme con soporte o ventas.	Escenario 1 – Información visible Given que estoy en landing When voy a “Contact” Then visualizo email, teléfono o formulario. Escenario 2 – Envío de formulario vacío Given no ingreso datos When envío Then muestra validación.	EP011
TS001	API de registro de dueño	Como Developer , quiero implementar un endpoint REST para registrar dueños de cafetería, para que el sistema guarde la información.	Escenario 1 – Registro exitoso Given request con datos válidos When POST /owners Then responde 201 con el ID del dueño. Escenario 2 – Email duplicado Given request con email existente When POST /owners Then responde 400 con mensaje de error.	EP001
TS002	API de login de dueño	Como Developer , quiero implementar un endpoint REST para el login de dueños, validando credenciales contra la base de datos.	Escenario 1 – Login exitoso Given request con credenciales correctas When POST /auth/login Then responde 200 con token JWT. Escenario 2 – Credenciales inválidas Given request con email o password incorrecto When POST /auth/login Then responde 401 Unauthorized.	EP001
TS003	API de gestión de empleados	Como Developer , quiero implementar endpoints REST para CRUD de	Escenario 1 – Alta de empleado Given request válido When POST /employees	EP002

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
		empleados, de forma que se almacenen sus datos.	<p>Then responde 201 con ID generado.</p> <p>Escenario 2 – Consulta</p> <p>Given GET /employees/{id} existente</p> <p>When envío request</p> <p>Then responde 200 con datos del empleado.</p>	
TS004	API de eliminación de empleado	Como Developer , quiero implementar un endpoint REST para eliminar empleados por renuncia o despido.	<p>Escenario 1 – Eliminación exitosa</p> <p>Given DELETE /employees/{id} existente</p> <p>When envío request</p> <p>Then responde 200 con confirmación.</p> <p>Escenario 2 – Empleado inexistente</p> <p>Given DELETE con ID no existente</p> <p>When envío request</p> <p>Then responde 404 Not Found.</p>	EP002
TS005	API de proveedores	Como Developer , quiero implementar endpoints REST para CRUD de proveedores, de forma que se administren los datos en el sistema.	<p>Escenario 1 – Registro proveedor</p> <p>Given request válido</p> <p>When POST /providers</p> <p>Then responde 201 con ID del proveedor.</p> <p>Escenario 2 – Actualización proveedor</p> <p>Given request con datos correctos</p> <p>When PUT /providers/{id}</p> <p>Then responde 200 con proveedor actualizado.</p>	EP003
TS006	API de inventario	Como Developer , quiero exponer endpoints REST para CRUD de insumos y productos, para que el inventario esté actualizado.	<p>Escenario 1 – Registro producto</p> <p>Given request válido</p> <p>When POST /products</p> <p>Then responde 201 con ID del producto.</p> <p>Escenario 2 – Consulta inventario</p> <p>Given GET /products</p> <p>When envío request</p> <p>Then responde 200 con listado de productos.</p>	EP006
TS007	API de recetas	Como Developer , quiero exponer endpoints REST para CRUD de recetas, para mantener actualizadas las preparaciones de los productos.	<p>Escenario 1 – Creación receta</p> <p>Given request válido</p> <p>When POST /recipes</p> <p>Then responde 201 con ID de receta.</p> <p>Escenario 2 – Actualización receta</p> <p>Given request válido</p> <p>When PUT /recipes/{id}</p> <p>Then responde 200 con receta actualizada.</p>	EP004
TS008	API de reportes financieros	Como Developer , quiero implementar un endpoint REST para generar reportes financieros, de forma que se obtengan métricas clave.	<p>Escenario 1 – Reporte generado</p> <p>Given parámetros válidos</p> <p>When GET /reports/financial?from=2025-01-01&to=2025-01-31</p> <p>Then responde 200 con JSON del reporte.</p> <p>Escenario 2 – Rango inválido</p> <p>Given parámetros con fechas incorrectas</p>	EP009

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
			When envío request Then responde 400.	
TS009	API de seguridad (JWT)	Como Developer , quiero implementar autenticación con JWT en los endpoints, para proteger el acceso a datos sensibles.	Escenario 1 – Token válido Given request con token válido When accedo a /employees Then responde 200 con datos. Escenario 2 – Token inválido Given request con token incorrecto When envío request Then responde 401 Unauthorized.	EP001
TS010	API de registro de costos	Como Developer , quiero implementar un endpoint REST para registrar costos (fijos, variables, directos, indirectos) por sucursal.	Escenario 1 – Registro exitoso Given request válido When POST /costs Then responde 201 con ID del costo. Escenario 2 – Datos incompletos Given request sin categoría o monto When POST /costs Then responde 400 con mensaje de validación.	EP005
TS011	API de clasificación de costos	Como Developer , quiero implementar lógica en el backend para clasificar automáticamente costos según su tipo y sucursal.	Escenario 1 – Clasificación exitosa Given request con tipo válido (fijo, variable, directo, indirecto) When POST /costs Then se guarda en la categoría correspondiente. Escenario 2 – Tipo inválido Given request con tipo no reconocido When POST Then responde 400.	EP005
TS012	API de consulta de costos	Como Developer , quiero implementar un endpoint REST para consultar los costos de una sucursal en un periodo de tiempo.	Escenario 1 – Consulta exitosa Given GET /costs?Sucursal=1&From=2025-01-01&To=2025-01-31 When envío request Then responde 200 con listado de costos. Escenario 2 – Fechas inválidas Given parámetros con rango inválido When GET /costs Then responde 400 con error.	EP005
SS001	Investigación sobre biblioteca Charts.kt	Como equipo de desarrollo, quiero investigar y prototipar la biblioteca Charts.kt para la generación de gráficos en aplicaciones Kotlin/Android, para evaluar si es viable mostrar dashboards de costos, ventas y rentabilidad.	Escenario 1 – Revisión de documentación Given que la biblioteca Charts.kt ofrece distintas opciones When el desarrollador revisa su documentación oficial y ejemplos Then identifica y documenta qué tipos de gráficos soporta. Escenario 2 – Prototipo funcional Given la necesidad de validación técnica When se construye un prototipo con al menos 2	EP012

Story ID	Título	Descripción	Criterios de aceptación	Relación con (Epic ID)
			gráficos (ej. barras y líneas) Then se documenta su facilidad de integración y limitaciones.	
SS002	Investigación de servicios externos para autenticación y autorización	Como equipo de desarrollo, quiero investigar la viabilidad de usar un servicio externo como Firebase Authentication o Supabase Auth para el manejo de autenticación y autorización de usuarios, para decidir si es conveniente delegar estas funciones críticas en lugar de desarrollarlas internamente.	<p>Escenario 1 – Revisión de documentación Given que Firebase y Supabase ofrecen servicios de autenticación When el desarrollador revisa su documentación oficial Then identifica los flujos de login soportados (email/password, OAuth, proveedores sociales).</p> <p>Escenario 2 – Análisis de costos Given que cada servicio tiene planes gratuitos y de pago When se comparan los modelos de precios Then se documenta cuál se ajusta mejor al presupuesto y escalabilidad del proyecto.</p> <p>Escenario 3 – Validación técnica con prototipo Given la necesidad de evaluar integración When se implementa un prototipo de registro/login de usuarios Then se documenta la facilidad de integración, tiempo de configuración y limitaciones.</p>	EP012
SS003	Investigación de servicios externos para almacenamiento de imágenes	Como equipo de desarrollo, quiero investigar la viabilidad de usar un servicio externo como Firebase Storage o AWS S3 para el almacenamiento de imágenes, para decidir si conviene delegar esta funcionalidad en lugar de implementarla en infraestructura propia.	<p>Escenario 1 – Revisión de documentación Given que Firebase Storage y AWS S3 ofrecen APIs para manejo de archivos When el desarrollador revisa la documentación oficial Then identifica opciones de subida, descarga, escalabilidad y permisos de acceso.</p> <p>Escenario 2 – Análisis de costos Given que cada servicio maneja un modelo de precios distinto (ej. GB almacenados, transferencias) When se comparan los costos de Firebase Storage y AWS S3 Then se documenta cuál es más conveniente según el volumen esperado de imágenes.</p> <p>Escenario 3 – Validación técnica con prototipo Given la necesidad de probar integración real When se implementa un prototipo que suba y recuperé al menos una imagen desde el servicio Then se documenta la facilidad de integración, rendimiento y limitaciones encontradas.</p>	EP012

2.4.2. Impact Mapping



2.4.3. Product Backlog

Esta sección servirá como guía para organizar el trabajo del equipo, asegurando que siempre avancemos lo más importante del proyecto.

# Orden	User Story Id	Título	Story Points (1/2/3/5/8)
1	SS001	Investigación sobre biblioteca Charts.kt	3
2	SS002	Investigación de servicios externos para autenticación y autorización	3
3	SS003	Investigación de servicios externos para almacenamiento de imágenes	3
4	US020	Ver sección Home	1
5	US021	Ver sección About	1
6	US022	Ver sección Features	1
7	US023	Botón Call to Action (CTA)	1
8	US024	Ver sección Contact	2
9	US011	Registrar costos	5
10	US012	Registrar entrada inventario	5
11	US013	Actualizar inventario	3
12	US014	Definir precio de venta	5
13	US018	Reporte de costos	8
14	US009	Crear receta	3
15	US010	Actualizar receta	3
16	US006	Crear proveedor	5
17	US008	Actualizar proveedor	5
18	US004	Actualizar empleado	5
19	US005	Eliminar empleado	2
20	US003	Listar empleados	3
21	US007	Listar proveedores	3
22	US015	Búsqueda de empleados	3
23	US016	Buscar proveedores	3
24	US017	Buscar inventario	3
25	US019	Ver recetas	5
26	US001	Registro de dueño	3
27	US002	Login seguro	5

2.5. Strategic-Level Domain-Driven Design

2.5.1. EventStorming

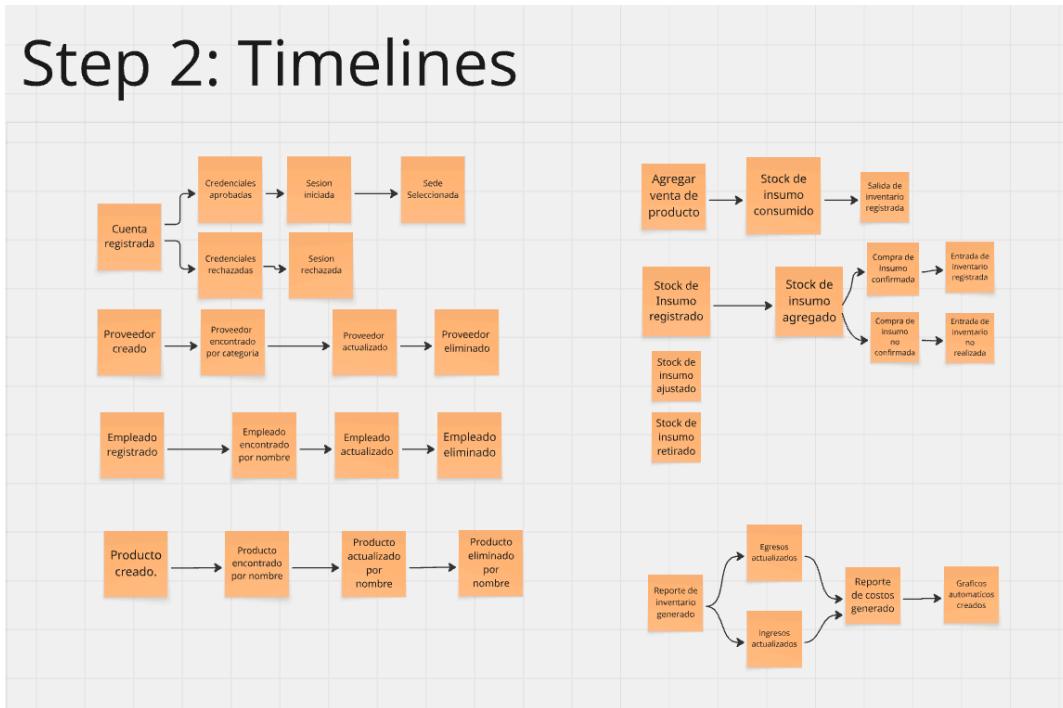
En esta sección se mostrará la aplicación del Domain-Driven Design (DDD) a nivel estratégico dentro del proyecto. El objetivo es identificar los distintos bounded context que intervienen en el dominio, así como las relaciones entre ellos, con el fin de establecer una arquitectura clara que permita escalar la solución y adaptarla a las necesidades del negocio.

2.5.1.1. Candidate Context Discovery

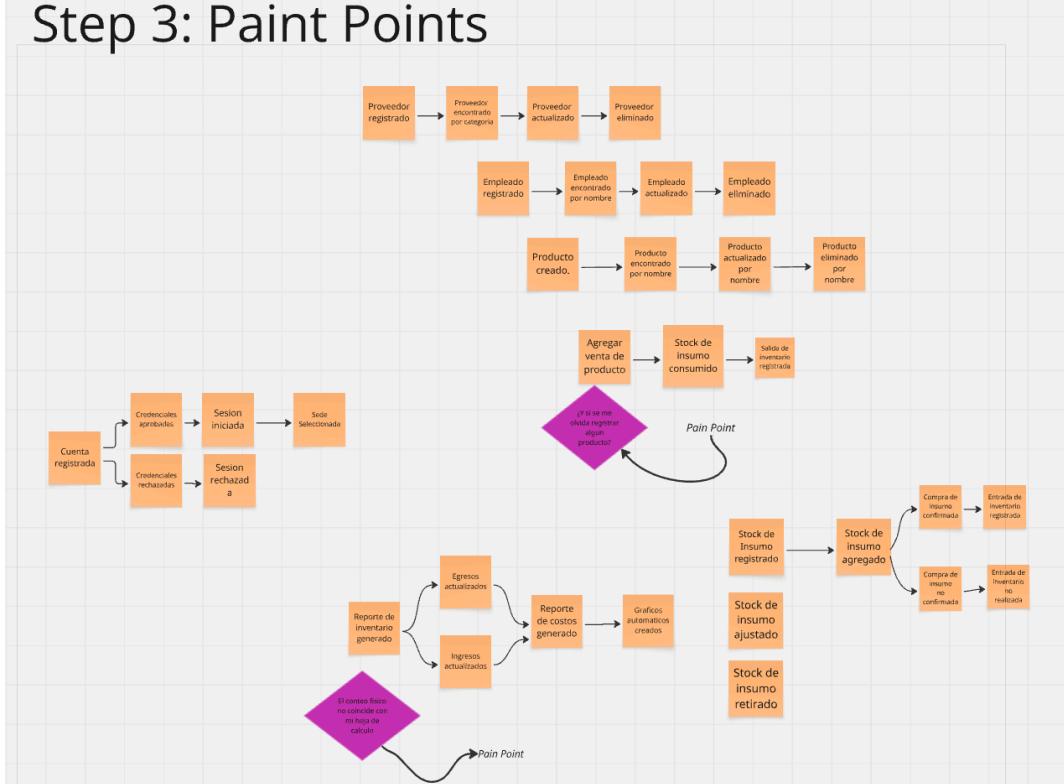
Step 1 - Collect Domain Events



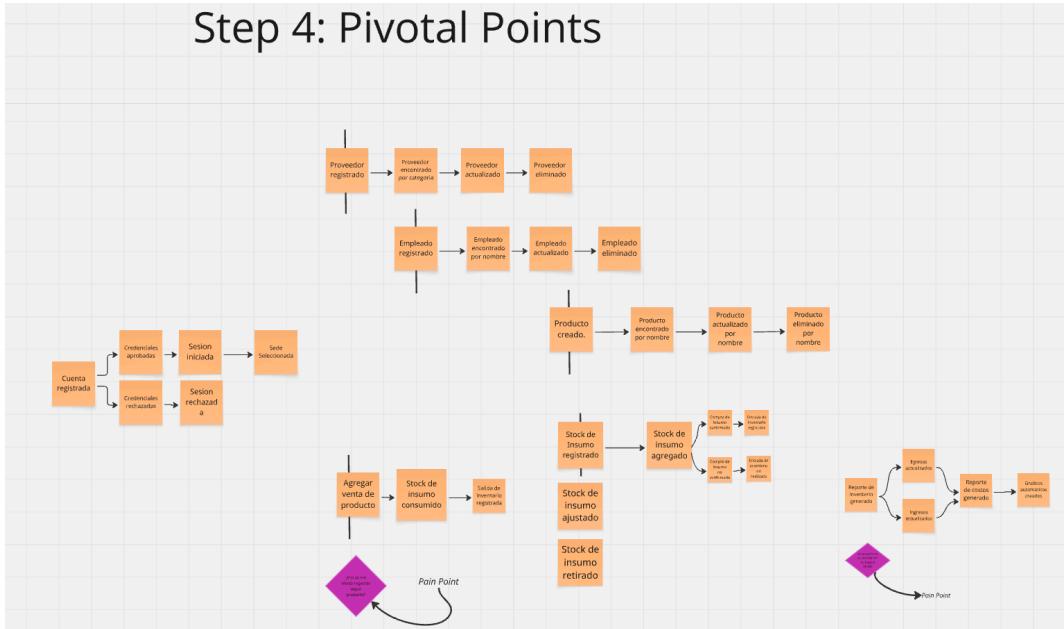
Step 2: Timelines



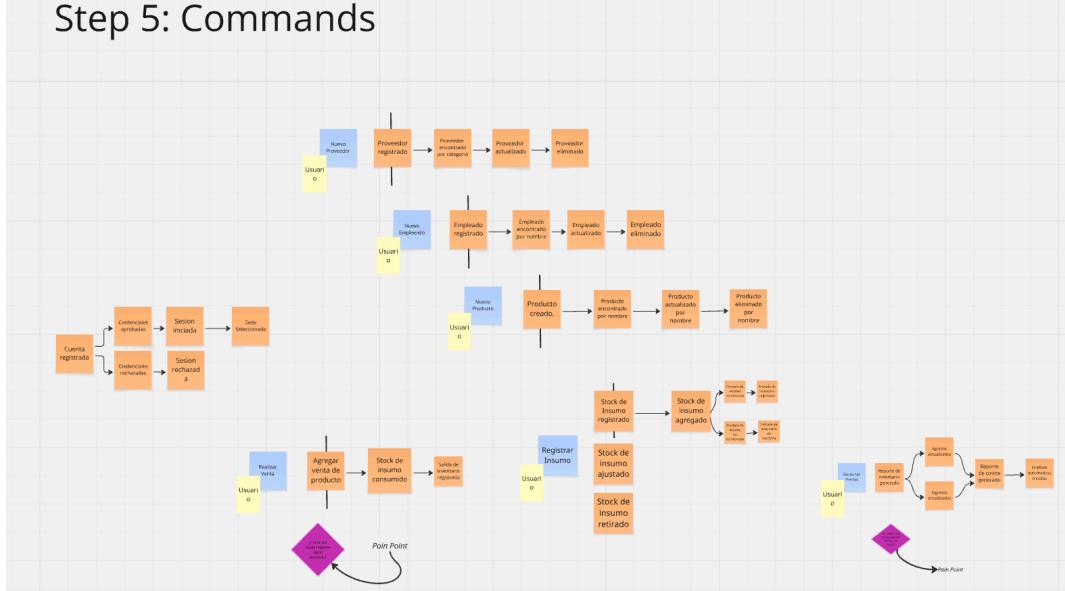
Step 3: Paint Points



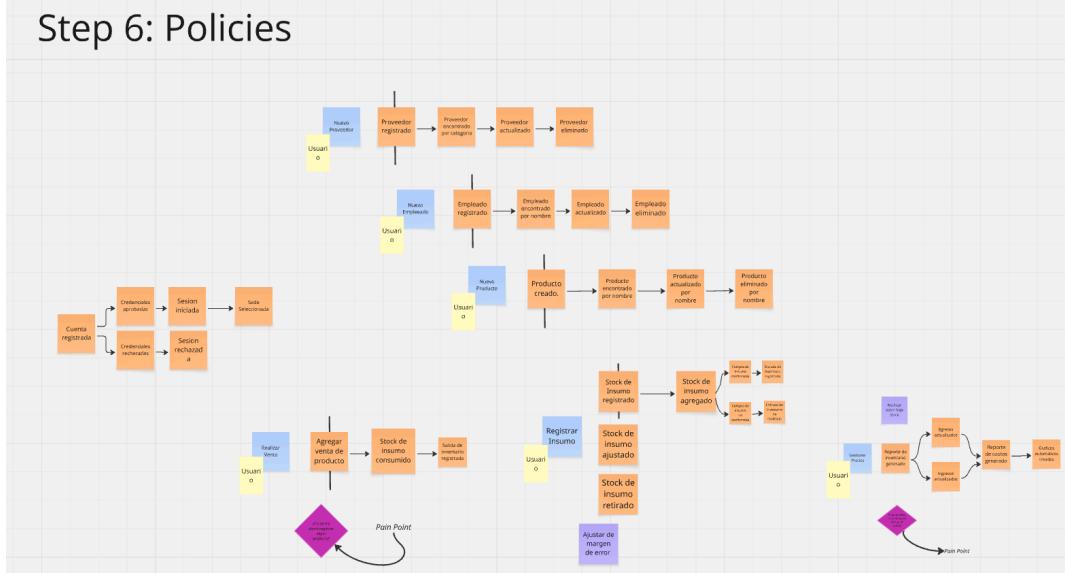
Step 4: Pivotal Points



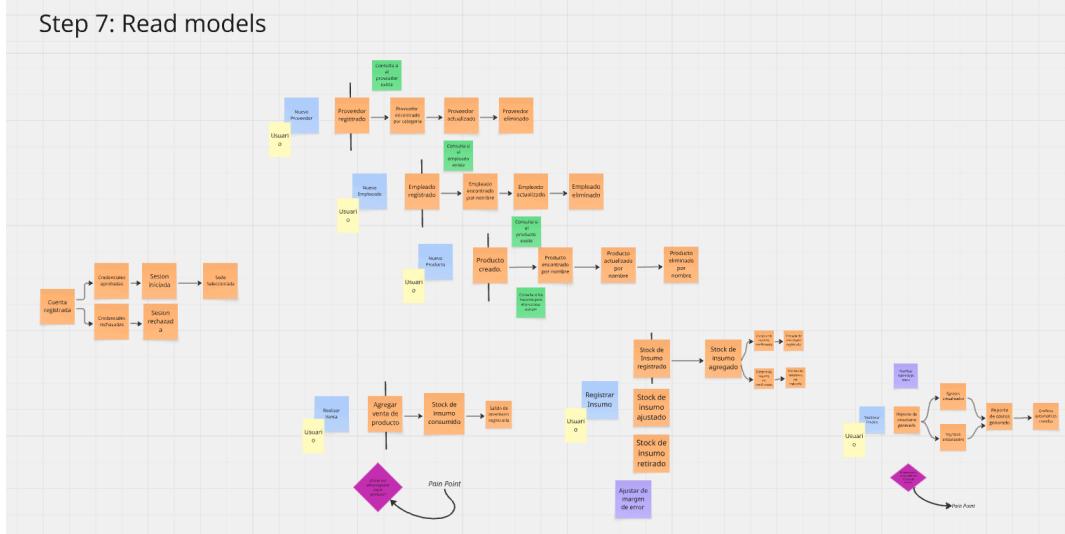
Step 5: Commands



Step 6: Policies



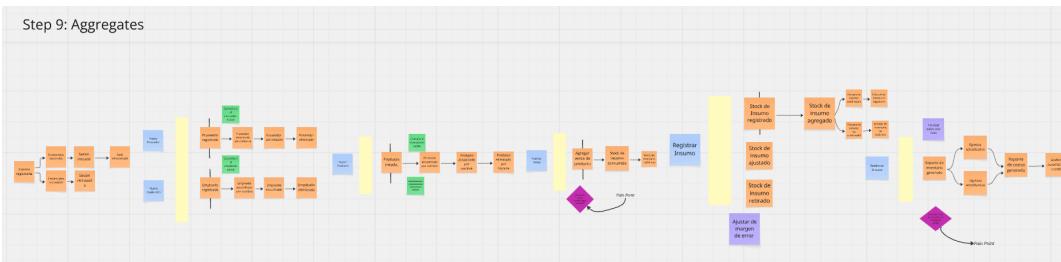
Step 7: Read models



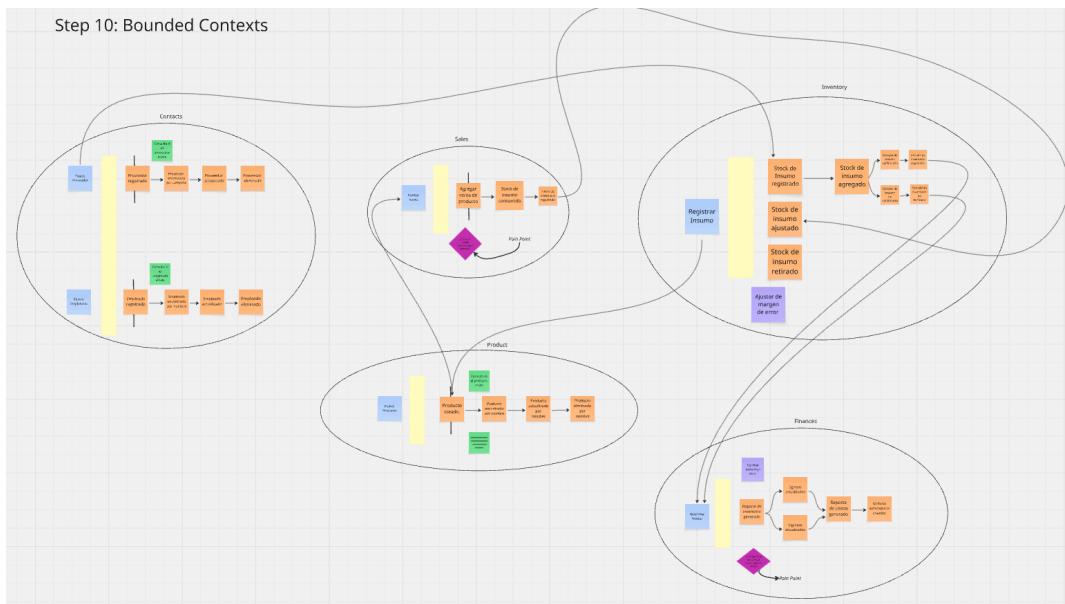
Step 8: External Systems and Actors



Step 9: Aggregates



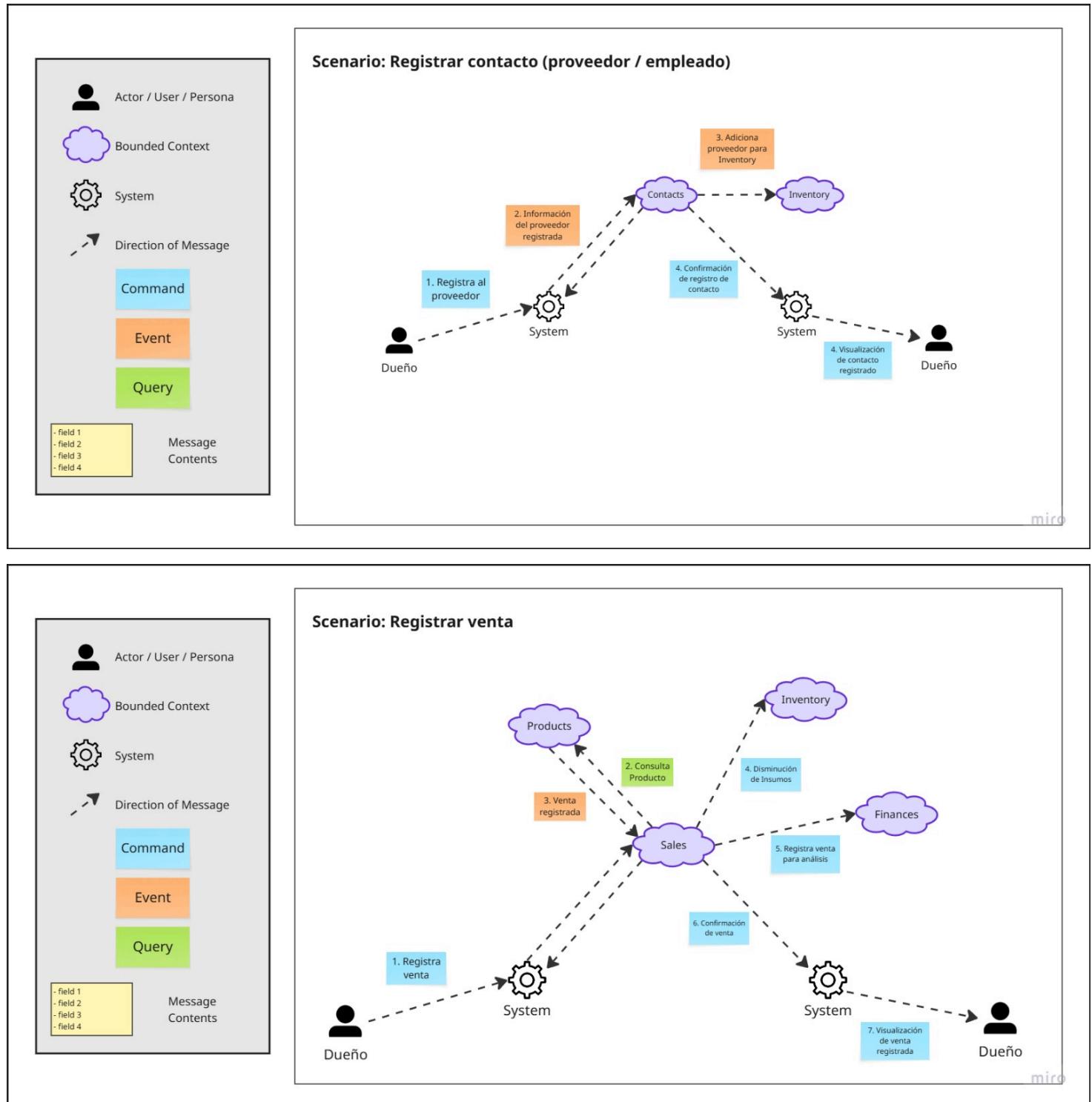
Step 10: Bounded Contexts

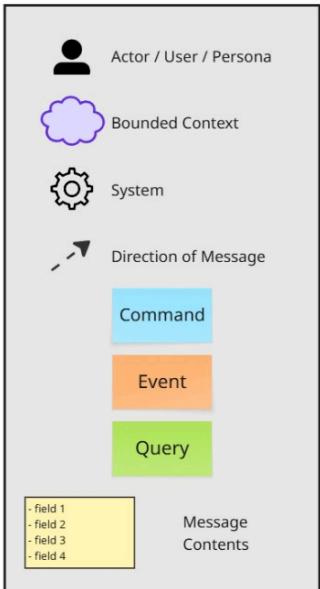


Link:

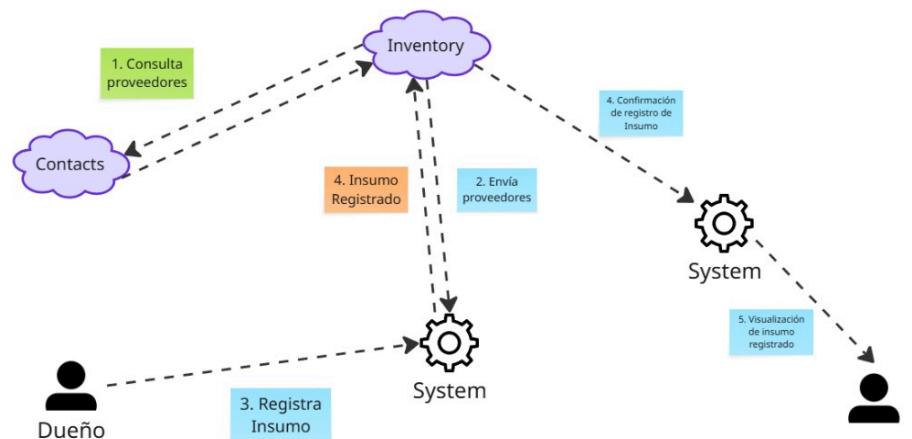
https://miro.com/welcomeonboard/TzdJdEdvSWJQa2pYS2FlU5DWE9ZV2w5MjkycmZRZzB2OUFQWkhOemVGUTd6ekE4R0tlVW11TmNMMVlrd1lnWHRQbk5lWHN2VEs3M0ttajJ5TG1LUXZ6SVhjZ255RHJzTjZ5eEZRSXFkZisvOVo3VjhaRmdSL0s4T0tBMmhQd1F3VHhHVHd5UWtSM1BidUtUYmxycDRnPTohdjE=?share_link_id=804149752797

2.5.1.2. Domain Message Flows Modeling

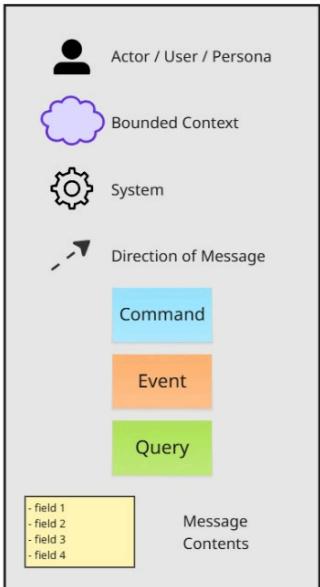




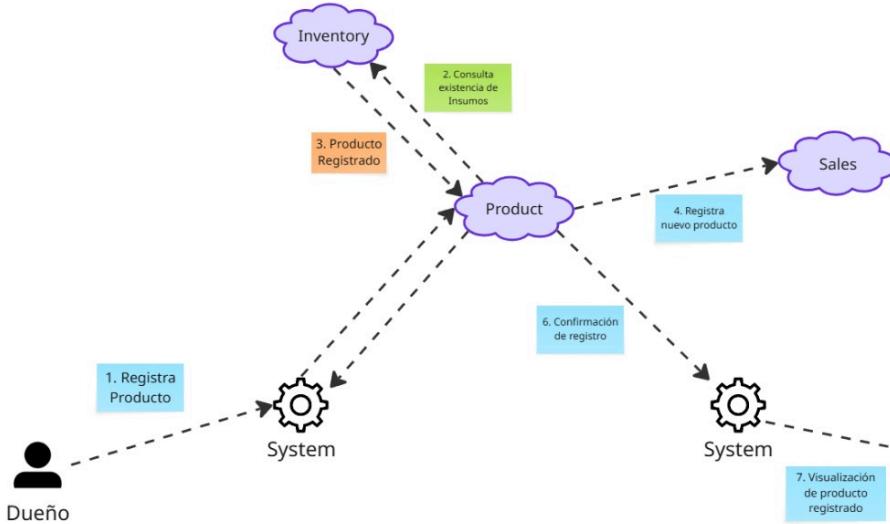
Scenario: Registrar insumo a inventario



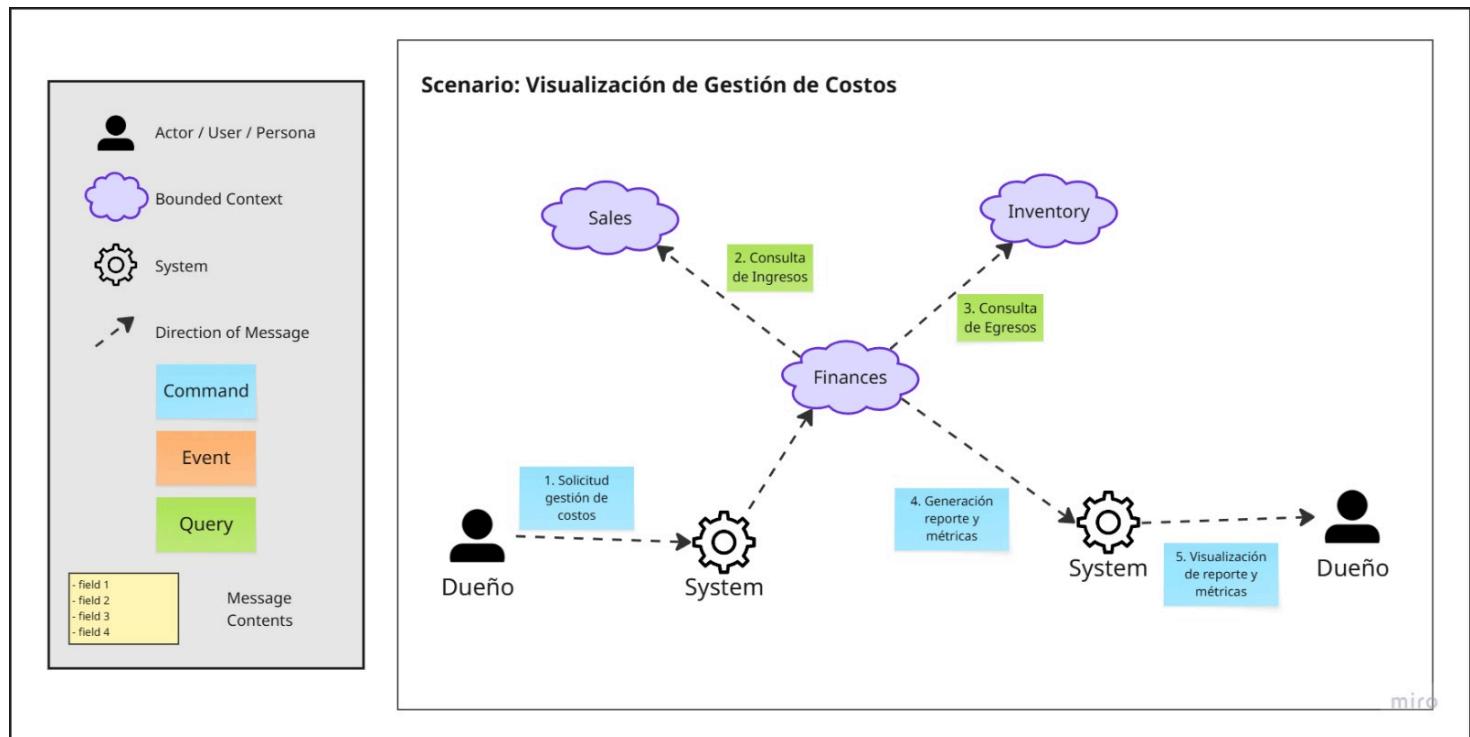
miro



Scenario: Registrar producto

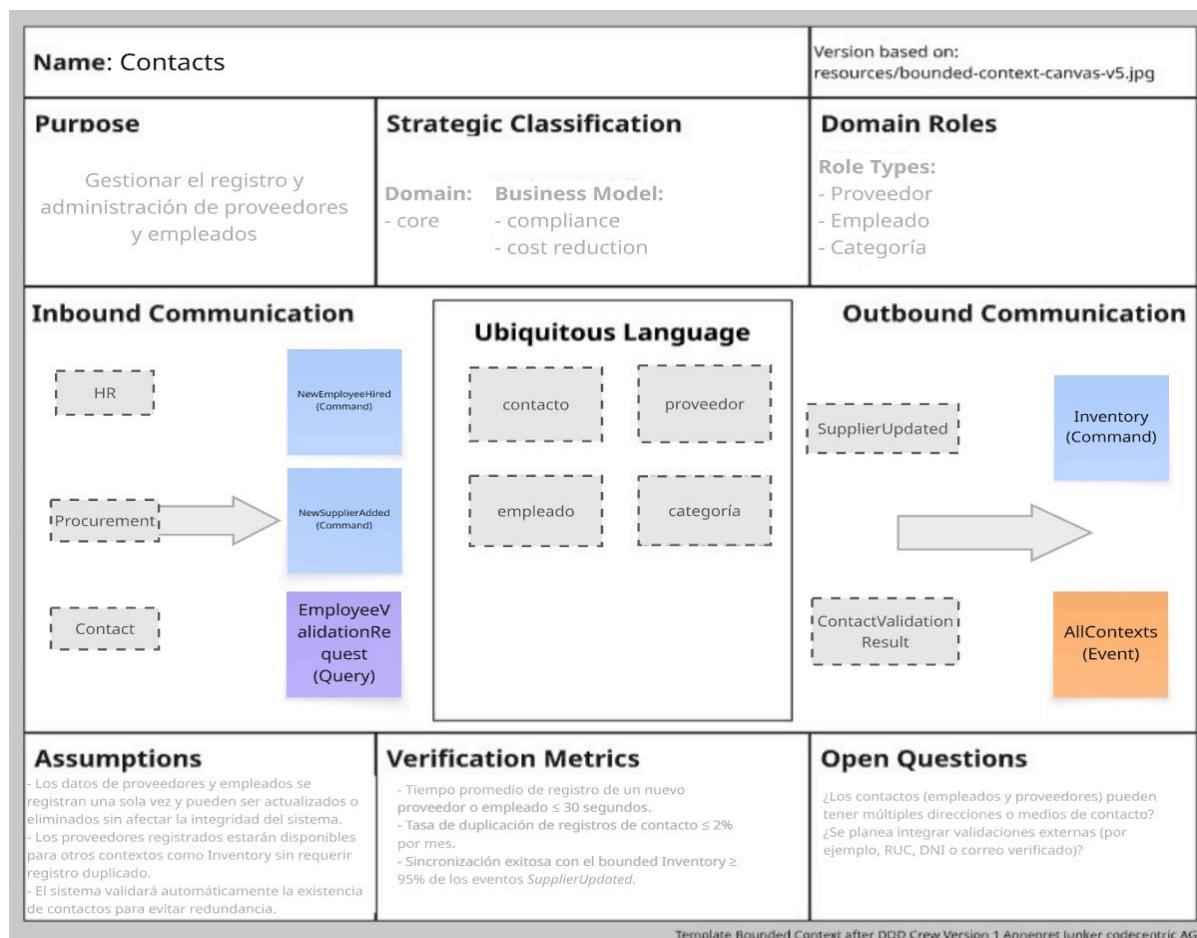


miro



Link: [Domain Message Flows Modeling](#)

2.5.1.3. Bounded Context Canvases



Template Bounded Context after DDD Crew Version 1 Ansgret Junker codecentric AG

Name Sales	Version based on: resources/bounded-context-canvas-v5.jpg	
Purpose Gestionar órdenes de venta, pagos y actualización de inventario.	Strategic Classification Domain: - core Business Model: - revenue - cost reduction	Domain Roles Role types: - Orden - Item - Pago - Empleado
Inbound Communication		
	Ubiquitous Language orden item pago estado método de pago	Outbound Communication InventoryUpdated INVENTORY (Event) PaymentConfirmed FINANCES (Event)
Assumptions Cada orden debe tener al menos un producto. El pago debe estar asociado a una orden válida. Verification Metrics 100% de órdenes registradas, tiempo de respuesta <500ms, 0% errores en pagos. Open Questions ¿Cómo validar productos omitidos en la orden?		
Template Bounded Context after DDD Crew Version 1 Ansgret Junker codecentric AG		

Name Inventory	Version based on: resources/bounded-context-canvas-v5.jpg	
Purpose Controlar insumos, salidas, ajustes y consolidación de stock.	Strategic Classification Domain: - supporting Business Model: - cost reduction	Domain Roles Role types: - Insumo, Stock, Ajuste, Salida
Inbound Communication		
	Ubiquitous Language insumo, stock, ajuste, salida, consolidación	Outbound Communication StockAlert FINANCES (Event)
Assumptions No se permite stock negativo. Las salidas deben estar justificadas. Verification Metrics 100% de movimientos registrados, 0% inconsistencias de stock. Open Questions ¿Cómo auditar ajustes manuales de inventario?		
Template Bounded Context after DDD Crew Version 1 Ansgret Junker codecentric AG		

Name Finances	Version based on: resources/bounded-context-canvas-v5.jpg	
Purpose	Strategic Classification	Domain Roles
Gestionar precios, reportes financieros y análisis de ingresos/gastos.	<p>Domain:</p> <ul style="list-style-type: none"> - supporting <p>Business Modell:</p> <ul style="list-style-type: none"> - cost reduction - revenue 	<p>Role types:</p> <ul style="list-style-type: none"> - Precio, Ingreso, Gasto, Reporte
Inbound Communication	Ubiquitous Language	Outbound Communication
<p>Sales</p> <p>PaymentConfirmed (Event)</p> <p>Inventory</p> <p>StockAlert (Event)</p>	<p>Ubiquitous Language</p> <p>precio, ingreso, gasto, reporte, gráfico</p>	<p>FinancialReportGenerated</p> <p>Outbound Communication</p> <p>ADMIN (Event)</p>
Assumptions	Verification Metrics	Open Questions
Los precios deben reflejar el costo real. Los reportes deben ser consistentes.	100% de reportes generados, 0% errores en cálculos.	¿Cómo manejar diferencias entre conteo físico y sistema?

Template Bounded Context after DDD Crew Version 1 Ansgret Junker codecentric AG

Name Product	Version based on: resources/bounded-context-canvas-v5.jpg	
Purpose	Strategic Classification	Domain Roles
Gestionar productos, validación de componentes y ciclo de vida.	<p>Domain:</p> <ul style="list-style-type: none"> - core <p>Business Modell:</p> <ul style="list-style-type: none"> - revenue 	<p>Role types:</p> <ul style="list-style-type: none"> - Producto, Componente, Receta
Inbound Communication	Ubiquitous Language	Outbound Communication
<p>Inventory</p> <p>ComponentAvailability (Query)</p>	<p>Ubiquitous Language</p> <p>producto, componente, receta, stock</p>	<p>ProductCreated</p> <p>Outbound Communication</p> <p>SALES (Event)</p>
Assumptions	Verification Metrics	Open Questions
Un producto debe tener componentes disponibles en stock.	100% de productos validados, 0% duplicados por nombre.	¿Cómo manejar eliminación de productos en uso?

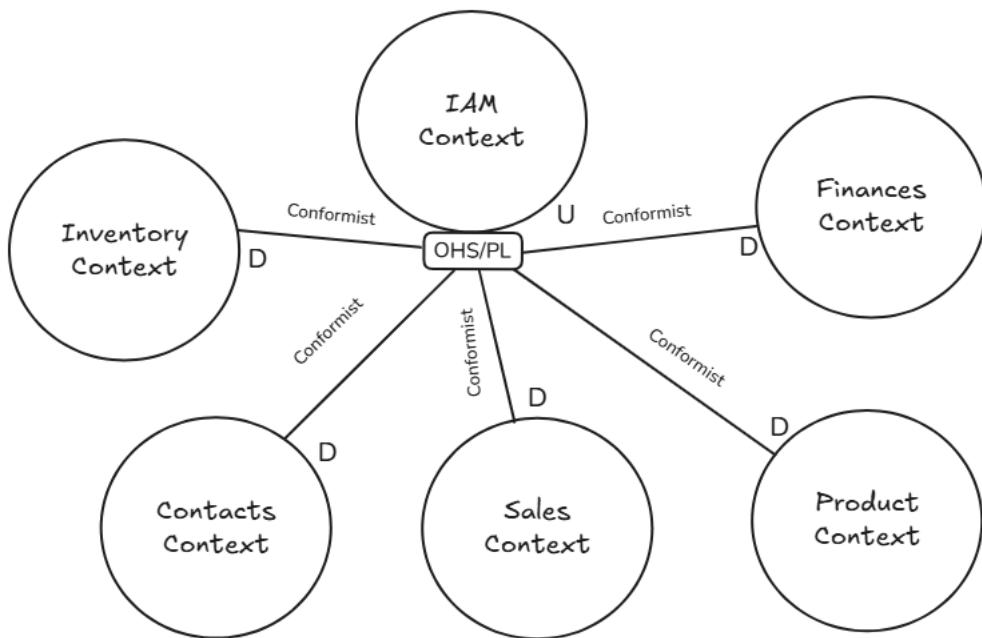
Template Bounded Context after DDD Crew Version 1 Ansgret Junker codecentric AG

2.5.2. Context Mapping

Context Mapping es la práctica de definir los límites de cada Bounded Context, dispone de su propia tecnología, lenguaje y arquitectura para reflejar fielmente su parte del dominio. A través de las Context Maps se describen las relaciones entre diferentes contextos, estableciendo reglas de traducción (Translation Maps) y señalando quién expone servicios (Upstream) y quién los consume (Downstream) para mantener la coherencia e independencia de cada modelo.

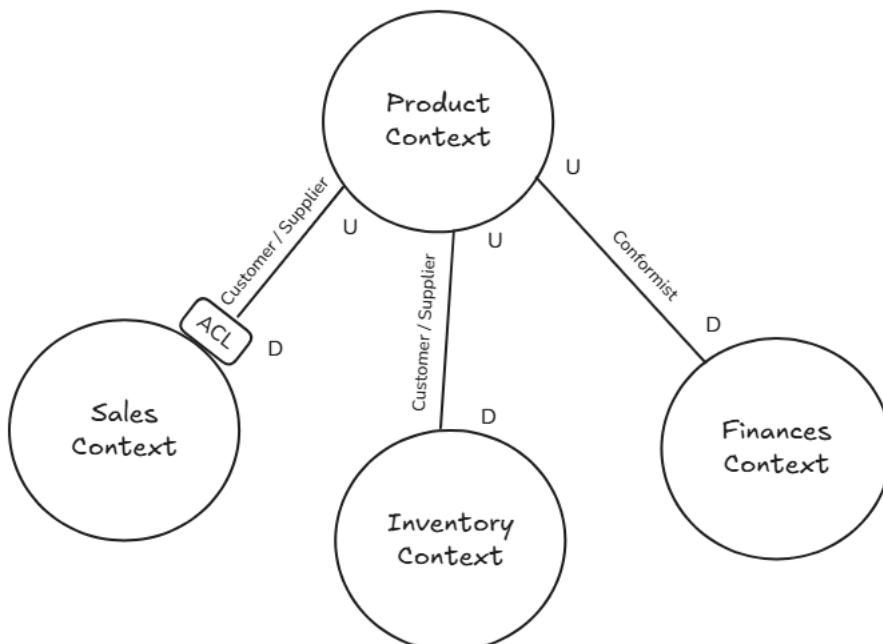
Autenticación y autorización (IAM Context)

IAM expone verificación de identidad y emisión/validación de tokens en JSON como OHS/PL. Sales, Inventory, Product, Finances y Contacts consumen este contrato sin transformación bajo un patrón Conformist.



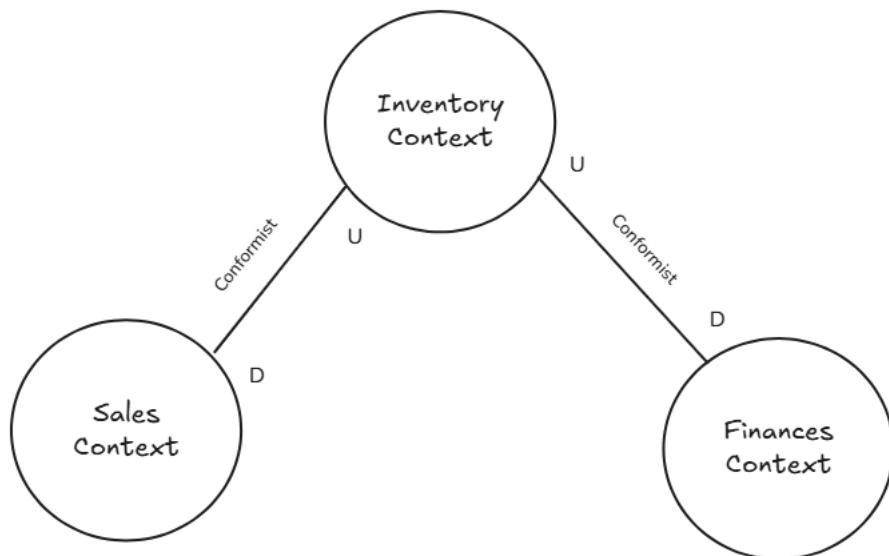
Catálogo y recetas (Product Context)

Product expone catálogo y recetas. Sales traduce y negocia atributos (ACL + Customer–Supplier), Inventory influye en la estructura (Customer–Supplier) y Finances consume el modelo estándar (Conformist).



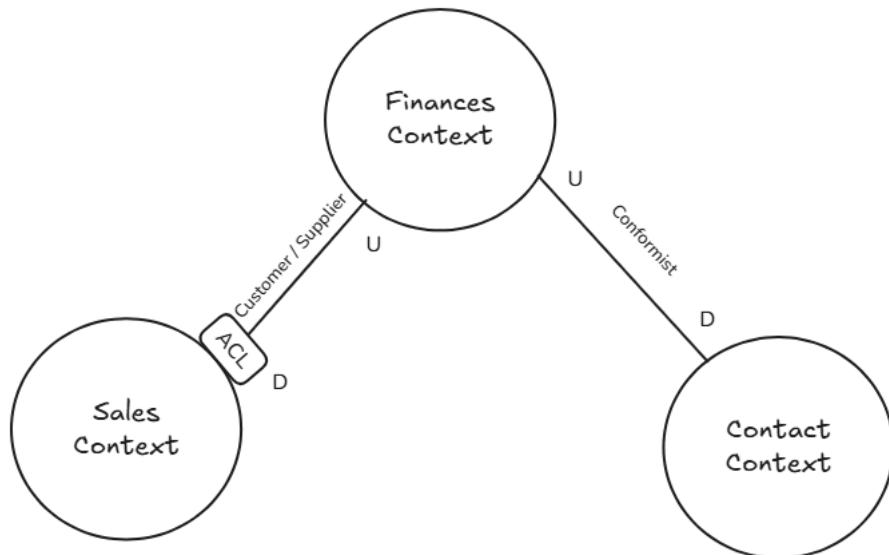
Stock (Inventory Context)

Inventory ofrece disponibilidad. Sales y Finances consultan estos datos directamente bajo un patrón Conformist.



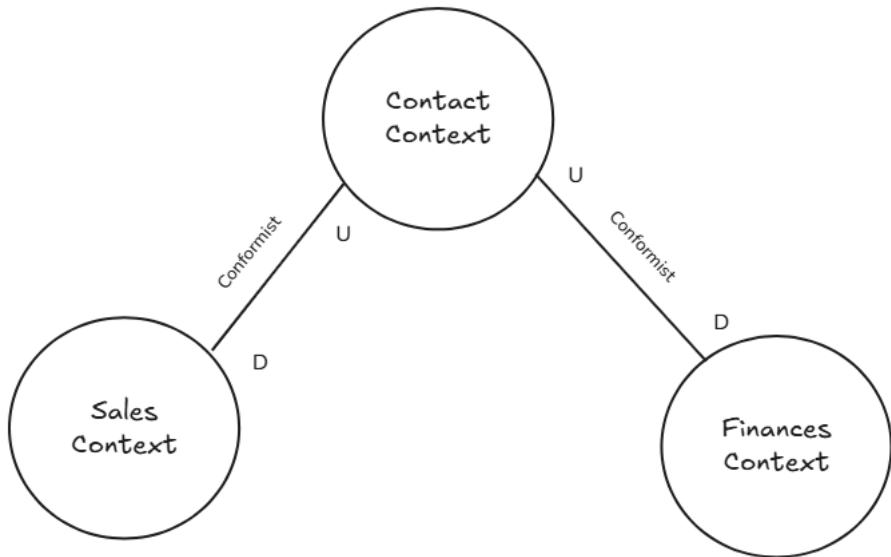
Reportes y comprobantes (Finances Context)

Finances expone generación de reportes y comprobantes. Sales traduce requisitos fiscales y negocia formatos (ACL + Customer–Supplier), mientras Contacts consume datos contables sin adaptaciones (Conformist).



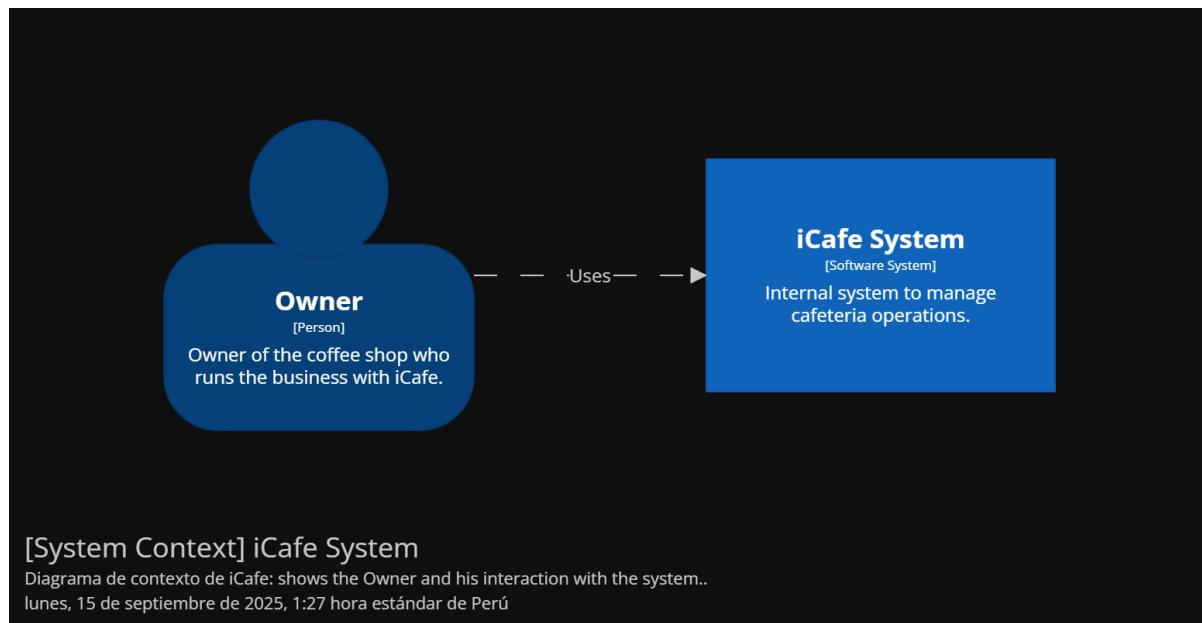
Contactos (Contact Context)

Contacts expone datos de los clientes/proveedores. Sales y Finances consumen esta información sin transformación bajo un patrón Conformist.

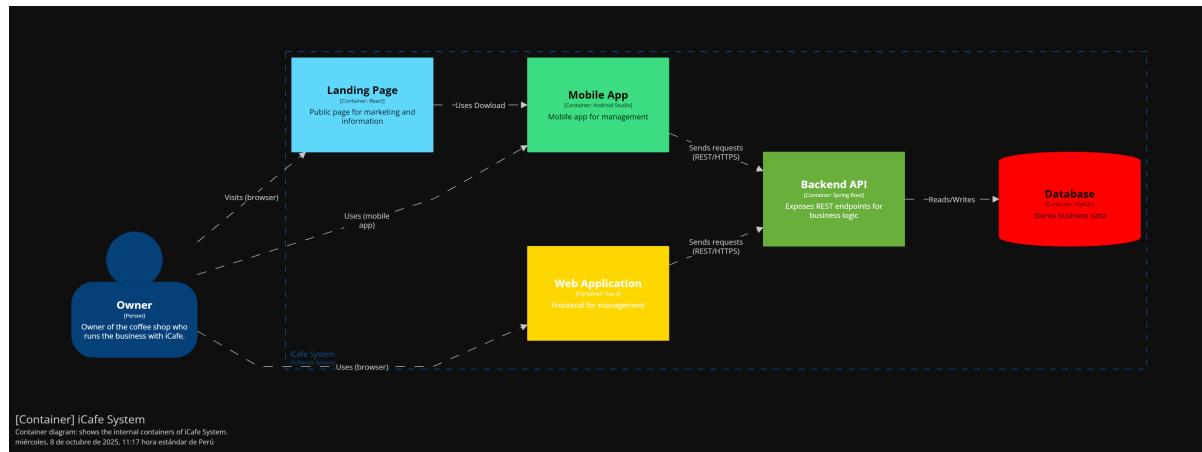


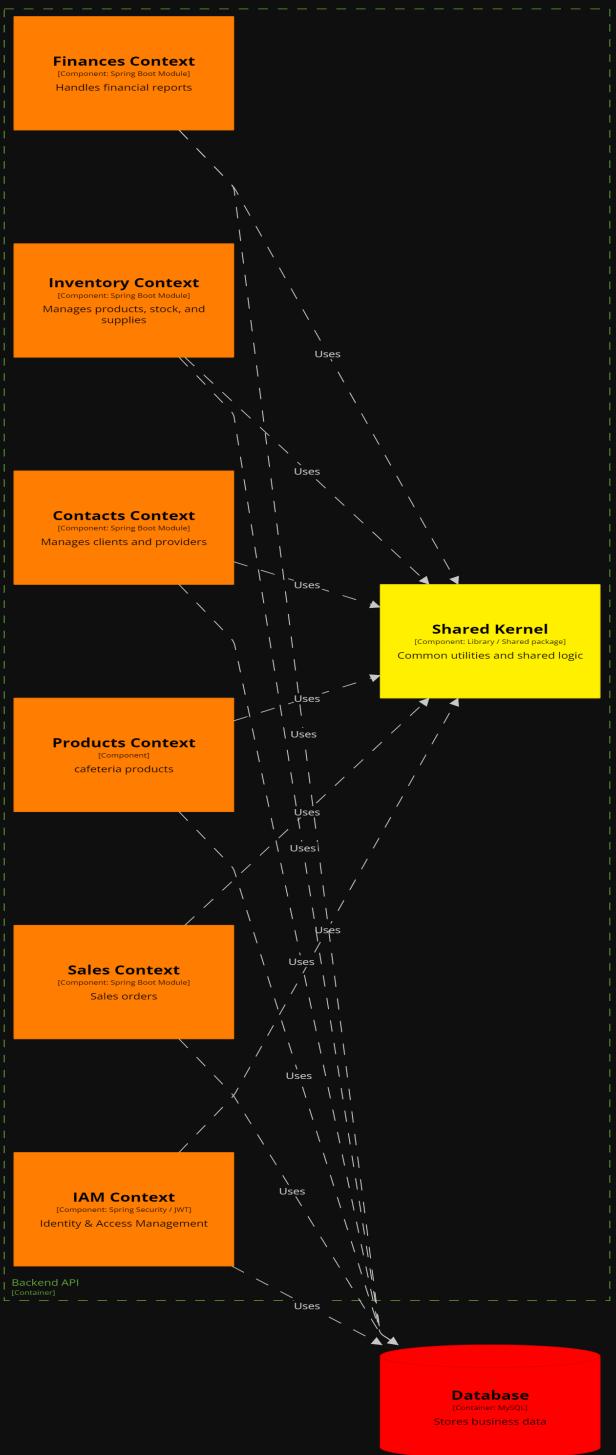
2.5.3. Software Architecture

2.5.3.1. Software Architecture Context Level Diagrams



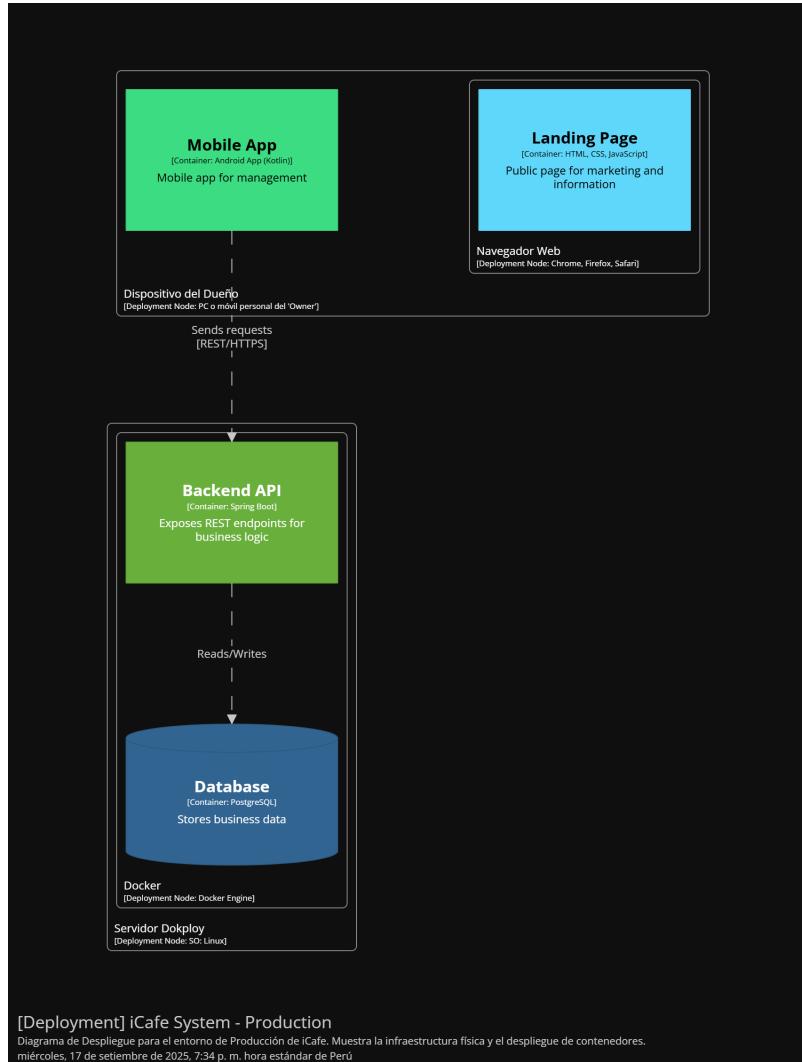
2.5.3.2. Software Architecture Container Level Diagrams





[Component] iCafe System - Backend API
Component diagram: shows internal modules of Backend API.
lunes, 15 de septiembre de 2025, 19:23 hora estándar de Perú

2.5.3.3. Software Architecture Deployment Diagrams



2.6. Tactical-Level Domain-Driven Design

2.6.1. Bounded Context: Contacts

El bounded context Contacts se encarga de la gestión de proveedores y empleados de la cafetería. Su objetivo es registrar, actualizar y eliminar estos contactos, permitiendo al dueño listar todos los registros para una administración eficiente y centralizada de las relaciones externas e internas.

2.6.1.1. Domain Layer

En esta capa se definen los elementos principales del dominio de contactos. Se modelan los agregados, entidades, value objects y servicios de dominio que representan la lógica central de cómo se gestionan los proveedores y empleados. El objetivo es mantener la consistencia del negocio y las reglas que garantizan el control y la validación de los datos de contactos.

Tipo	Clase / Nombre	Descripción	Atributos / Valores
Aggregate	ContactManagement	Gestión de contactos como una unidad de consistencia	suppliers, employees
Root	Supplier	Representa un proveedor del inventario	id, name, contactInfo, category
Entity	Employee	Representa un empleado de la cafetería	id, name, position, contactInfo
Value Object	Address	Representa una dirección física o de contacto	street, city, zipCode, country
Value Object	PhoneNumber	Representa un número de teléfono con tipo	number, type (MOBILE, OFFICE)
Domain Service	ContactValidationService	Valida la unicidad y consistencia de datos de contactos	validateSupplier(), validateEmployee()

Tipo	Clase / Nombre	Descripción	Atributos / Valores
Enum	ContactType	Define los tipos de contactos disponibles	SUPPLIER, EMPLOYEE

2.6.1.2. Interface Layer

En esta capa se encuentran los controladores y objetos de transferencia (DTOs) que sirven como punto de comunicación entre el sistema y los usuarios o clientes externos. Su función principal es exponer endpoints REST para la gestión de proveedores y empleados, facilitando la interacción con la aplicación de una forma clara y estructurada.

Tipo	Clase / Nombre	Descripción	Métodos / Endpoints principales
Controller	ContactController	Expone endpoints REST para la gestión general de contactos. Llama a la capa de aplicación.	<ul style="list-style-type: none"> - GET /contacts — listar todos los contactos - GET /contacts/{id} — obtener detalle
Controller	SupplierController	Maneja operaciones relacionadas con proveedores (registrar, actualizar, eliminar).	<ul style="list-style-type: none"> - POST /suppliers — registrar nuevo proveedor - PUT /suppliers/{id} — actualizar datos - DELETE /suppliers/{id} — eliminar proveedor
Controller	EmployeeController	Maneja operaciones relacionadas con empleados (registrar, actualizar, eliminar).	<ul style="list-style-type: none"> - POST /employees — registrar nuevo empleado - PUT /employees/{id} — actualizar datos - DELETE /employees/{id} — eliminar empleado
DTO	SupplierResource	Objeto de transferencia que devuelve datos de proveedores.	id, name, contactInfo, category
DTO	EmployeeResource	Objeto de transferencia que devuelve datos de empleados.	id, name, position, contactInfo
DTO	ContactCommandResource	Objeto de transferencia para recibir comandos de creación/actualización.	type, name, contactInfo

2.6.1.3. Application Layer

Esta capa maneja los flujos de procesos de contactos mediante el uso de command handlers y event handlers. Su rol es coordinar las operaciones del negocio, orquestando las acciones que se solicitan desde la interfaz y delegando la lógica al dominio. Así se asegura que cada comando o evento dispare el proceso adecuado dentro del contexto de contactos.

Tipo	Clase / Nombre	Descripción	Método / Comandos manejados
Command Handler	RegisterSupplierHandler	Maneja el proceso de registrar un nuevo proveedor en el sistema.	- handle(RegisterSupplierCommand)
Command Handler	UpdateSupplierHandler	Maneja la actualización de datos de un proveedor existente.	- handle(UpdateSupplierCommand)
Command Handler	RemoveSupplierHandler	Maneja la eliminación de un proveedor.	- handle(RemoveSupplierCommand)
Command Handler	RegisterEmployeeHandler	Maneja el proceso de registrar un nuevo empleado.	- handle(RegisterEmployeeCommand)
Command Handler	UpdateEmployeeHandler	Maneja la actualización de datos de un empleado existente.	- handle(UpdateEmployeeCommand)
Command Handler	RemoveEmployeeHandler	Maneja la eliminación de un empleado.	- handle(RemoveEmployeeCommand)

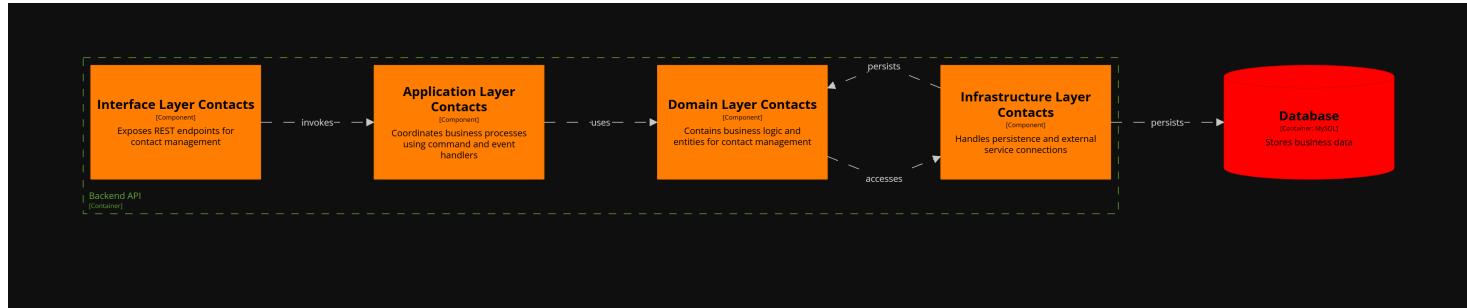
Tipo	Clase / Nombre	Descripción	Método / Comandos manejados
Event Handler	ContactUpdatedHandler	Escucha eventos de dominio de actualización de contactos y dispara notificaciones o auditoría.	- on(ContactUpdatedEvent)

2.6.1.4. Infrastructure Layer

En esta capa se implementa la conexión con servicios externos, principalmente la base de datos. Incluye los repositorios que persisten la información de proveedores y empleados utilizando JPA/Hibernate. De esta manera, el dominio se mantiene independiente de la tecnología, mientras la infraestructura garantiza el acceso confiable a los datos.

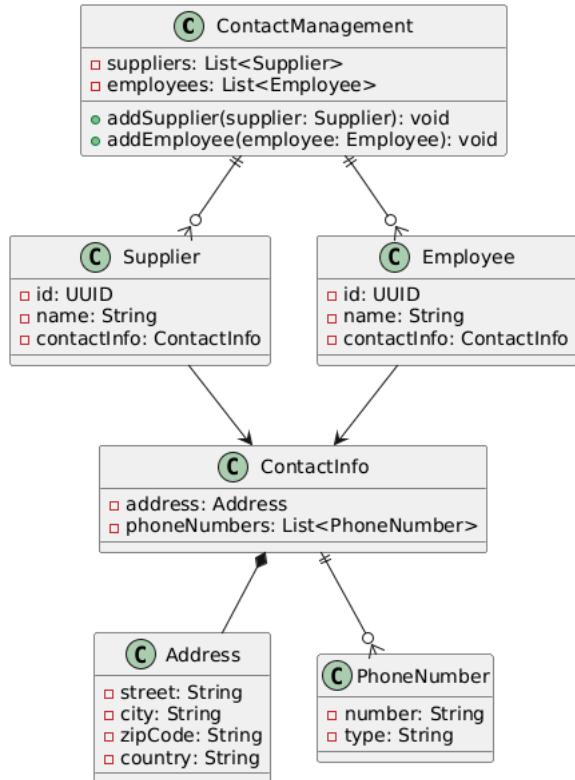
Tipo	Clase / Nombre	Descripción	Notas Técnicas
Repository	SupplierRepository	Implementación JPA/Hibernate para persistir proveedores.	Mapa Supplier a tabla suppliers.
Repository	EmployeeRepository	Implementación JPA/Hibernate para persistir empleados.	Mapa Employee a tabla employees.

2.6.1.5. Bounded Context Software Architecture Component Level Diagrams

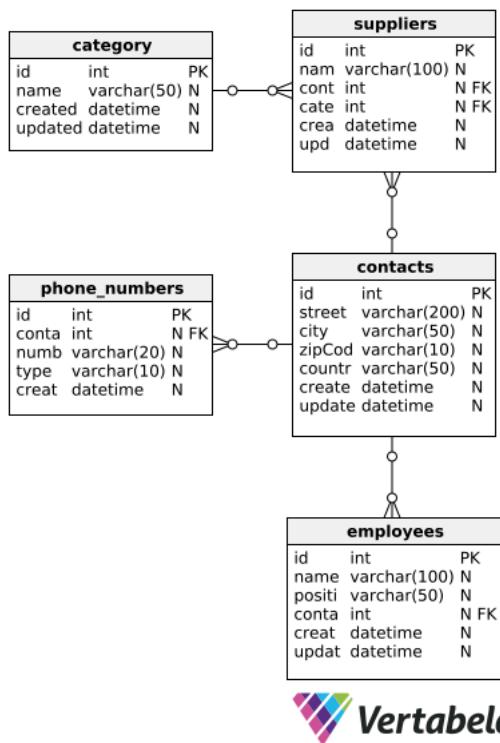


2.6.1.6. Bounded Context Software Architecture Code Level Diagrams

2.6.1.6.1. Bounded Context Domain Layer Class Diagrams



2.6.1.6.2. Bounded Context Database Design Diagram



2.6.2. Bounded Context: Product

2.6.2.1. Domain Layer

El bounded context Product se encarga de definir el estándar del dueño para cada producto del menú. Su objetivo es crear, actualizar, archivar o eliminar productos garantizando consistencia: nombre único por sucursal, componentes válidos y control de versionado para que otros contextos (p. ej., Sales) calculen consumos con el producto correcto.

Tipo	Clase / Nombre	Descripción	Atributos / Valores clave
Aggregate	Product	Agregado raíz que modela un producto del menú, soportando modalidad SIMPLE o COMPOSED con control de versión	id, ownerId, sucursalId, name, category, type (SIMPLE, COMPOSED), portions, steps, version, directItem (DirectItemSpec?), components: List, createdAt, updatedAt
Root	Product	Root del agregado; garantiza invariantes y emite eventos	—
Value Object	RecipeItem	Ítem de la receta COMPOSED: insumo y cantidad por porción	itemId, qtyPerPortion (> 0), unit (g, ml, unidad)
Value Object	DirectItemSpec	Especificación para producto SIMPLE: vínculo al insumo de inventario y conversión a porción	itemId, portionFactor (> 0), unit (unidad por porción)
Domain Service	ProductPolicy	Reglas transversales: unicidad de nombre, normalización de unidades, validación de referencias a Inventory	ensureUniqueName(), normalizeUnits(), validateItemsExist()
Domain Event	ProductCreated	Producto creado con receta inicial o referencia directa	productId, name, type, version, components[]?, directItem?
Domain Event	ProductUpdated	Cambios en datos generales y/o receta; si cambia COMPOSED o portions, version++	productId, changedFields, version, components[]?, directItem?
Domain Event	ProductArchived	Producto archivado	productId

Tipo	Clase / Nombre	Descripción	Atributos / Valores clave
Domain Event	ProductActivated	Producto reactivado	productId
Domain Event	ProductDeleted (opcional)	Eliminación lógica/física según política	productId

2.6.2.2. Interface Layer

En esta capa se modela el agregado Product (root) y sus value objects, junto con reglas de negocio e invariantes.

Tipo	Clase / Nombre	Descripción	Métodos / Endpoints principales
Controller	ProductController	CRUD de productos y lectura de receta vigente	POST /products · PUT /products/{id} · PATCH /products/{id}/archive · PATCH /products/{id}/activate · DELETE /products/{id} · GET /products/{id} · GET /products
Controller	RecipeExportController (opcional)	Solicita exportación de recetario PDF (delegado a Documents)	POST /products/exports/recipes-pdf
DTO (in)	ProductCommandResource	Payload de creación/actualización con soporte SIMPLE o COMPOSED	name, category, type, portions, steps, directItem: {itemId, portionFactor, unit}?, components: [{itemId, qtyPerPortion, unit}]?
DTO (out)	ProductResource	Respuesta de producto normalizada	id, name, category, type, portions, steps, status, version, directItem?, components[], createdAt, updatedAt
DTO (out)	RecipeltemResource	Ítem de receta para COMPOSED	itemId, name, qtyPerPortion, unit
DTO (out)	DirectItemResource	Especificación de producto SIMPLE	itemId, name, portionFactor, unit

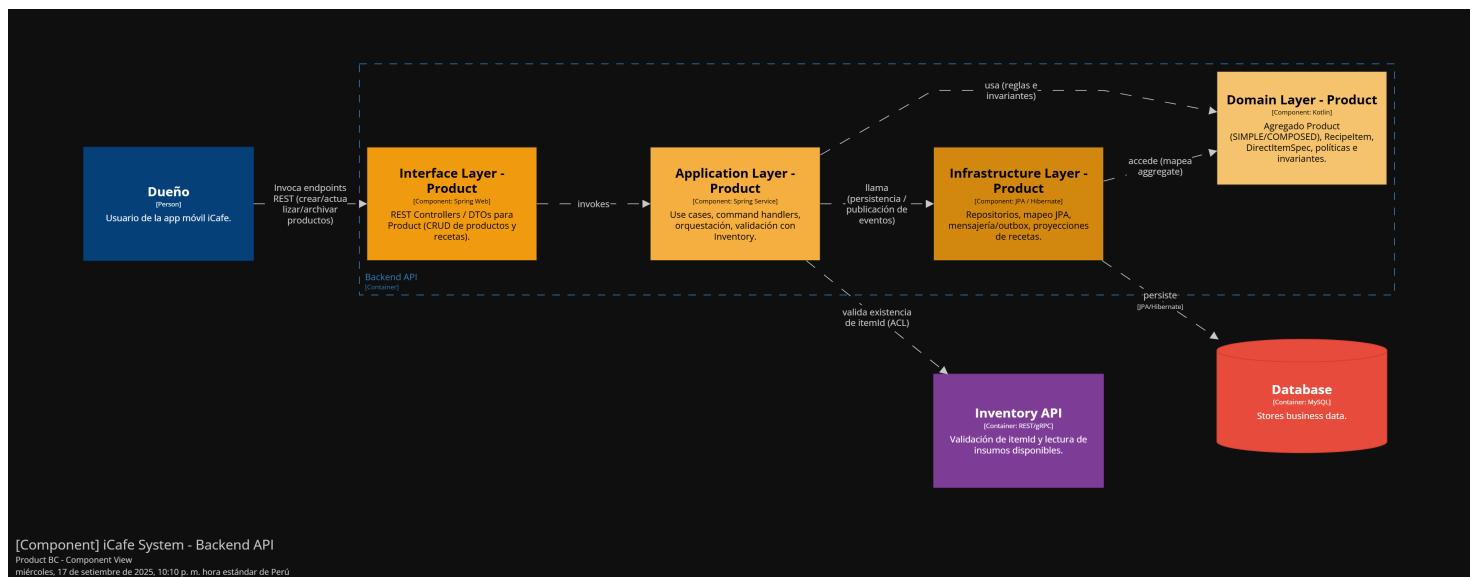
2.6.2.3. Application Layer

Tipo	Clase / Nombre	Descripción	Métodos / Comandos manejados
Command Handler	CreateProductHandler	Crea producto. Valida unicidad, existencia de itemId en Inventory (ACL) y coherencia entre type/directItem/components	handle(CreateProductCommand)
Command Handler	UpdateProductHandler	Actualiza datos; si cambia components o portions ⇒ version++	handle(UpdateProductCommand)
Command Handler	ArchiveProductHandler	Marca producto como ARCHIVED	handle(ArchiveProductCommand)
Command Handler	ActivateProductHandler	Reactiva producto	handle(ActivateProductCommand)
Command Handler	DeleteProductHandler (opcional)	Elimina o soft-delete según política	handle>DeleteProductCommand)
Event Handler	ProductChangedProjection	Actualiza read models para listados y exportación	on(ProductCreated, ProductUpdated, ProductArchived, ProductActivated)
Event Handler	ProductChangedPublisher	Publica cambios para invalidar cachés o actualizar catálogos en Sales	on(ProductCreated, ProductUpdated)

2.6.2.4. Infrastructure Layer

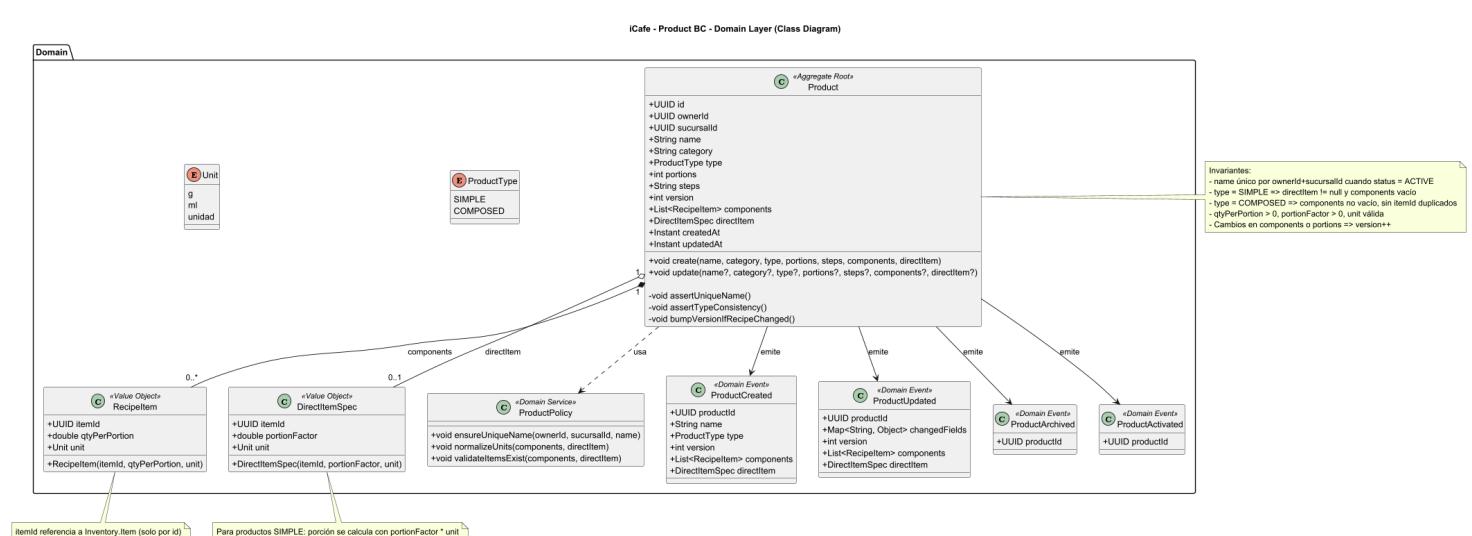
Tipo	Clase / Nombre	Descripción	Notas Técnicas
Repository	ProductRepository	Persistencia del agregado Product	JPA/Hibernate o equivalente; tablas products y product_components; para SIMPLE, tabla product_direct_item; índices por owner+Sucursal+name
Mapper	ProductJpaMapper	Mapear entidades JPA a dominio (VOs Recipelm y DirectItemSpec)	Conversión de unidades consistente; collections embebidas versionadas
Messaging	DomainEventOutbox	Publicación transaccional de eventos de dominio	Tabla outbox_events + publicador asíncrono
Integration	InventoryACLClient	Validar existencia de itemID contra Inventory	REST/gRPC GET /inventory/items/{id} con timeout/retry/circuit breaker
Projection	ProductReadModelRepository	Proyección denormalizada para listados y exportación PDF	Vistas materializadas con name, type, version, components/directItem

2.6.2.5. Bounded Context Software Architecture Component Level Diagrams

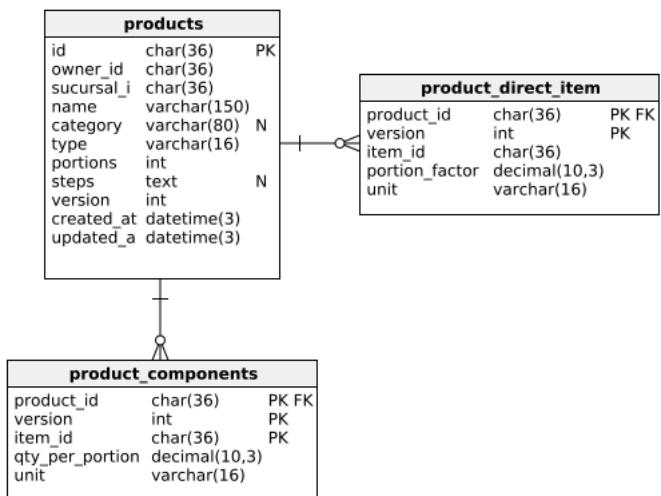


2.6.2.6. Bounded Context Software Architecture Code Level Diagrams

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

El bounded context Inventory se encarga de la gestión de los insumos de la cafetería, como café, azúcar o leche. Su objetivo es registrar cada insumo disponible y controlar el flujo de entradas y salidas del stock, ya sea por compras, consumo en pedidos, desperdicio o ajustes de inventario. Además, permite generar alertas cuando un insumo alcanza niveles bajos para garantizar la continuidad en la preparación de productos.

2.6.3.1. Domain Layer

En esta capa se definen los elementos principales del dominio del inventario. Aquí se modelan los agregados, entidades, value objects y servicios de dominio que representan la lógica central de cómo se gestionan los insumos y sus movimientos. El objetivo es mantener la consistencia del negocio y las reglas que garantizan el control del stock.

Tipo	Clase / Nombre	Descripción	Atributos / Valores
Aggregate	SupplyManagement	Gestiona el inventario de insumos como una unidad de consistencia.	supplyItems, transactions
Aggregate Root	SupplyItem	Representa cada insumo dentro del inventario.	id, nombre, unidadMedida, cantidadActual, puntoDeReorden
Entity	InventoryTransaction	Registra un movimiento de inventario.	id, tipoMovimiento, cantidad, fecha, referencia
Value Object	Quantity	Representa una cantidad con su medida, garantiza consistencia.	valor, unidadMedida
Value Object	UnitOfMeasure	Define la unidad de medida utilizada.	gramos, kilogramos, litros, unidades, etc.
Value Object	ReorderPoint	Valor mínimo que dispara la alerta de reabastecimiento.	cantidadMinima
Domain Service	StockConsumptionService	Encapsula la lógica de reducción de stock cuando se consumen insumos.	consumirInsumo(supplyItem, quantity)
Domain Service	StockReplenishmentService	Encapsula la lógica de reabastecimiento al registrar nuevas entradas.	reabastecer(supplyItem, quantity)
Enum	TransactionType	Define los tipos de movimientos en el inventario.	ENTRADA, SALIDA, AJUSTE
Enum	UnitMeasureType	Define los tipos de unidades disponibles para medir insumos.	GRAMOS, KILOGRAMOS, LITROS, UNIDADES

2.6.3.2. Interface Layer

En esta capa se encuentran los controladores y objetos de transferencia (DTOs) que sirven como punto de comunicación entre el sistema y los usuarios o clientes externos. Su función principal es exponer endpoints REST para la gestión de insumos y movimientos, facilitando la interacción con la aplicación de una forma clara y estructurada.

Tipo	Clase / Nombre	Descripción	Métodos / Endpoints principales
Controller	InventoryController	Expone endpoints REST para la gestión del inventario. Llama a la capa Application.	<ul style="list-style-type: none"> - GET /inventory/items → listar insumos - GET /inventory/items/{id} → obtener detalle de un insumo - POST /inventory/items → registrar un nuevo insumo - PUT /inventory/items/{id} → actualizar datos de un insumo - DELETE /inventory/items/{id} → eliminar insumo
Controller	StockMovementController	Maneja operaciones relacionadas con entradas y salidas de inventario.	<ul style="list-style-type: none"> - POST /inventory/movements → registrar un movimiento (entrada, salida, ajuste) - GET /inventory/movements → listar movimientos - GET /inventory/movements/{id} → detalle de un movimiento
DTO	InventoryTransactionResource	Objeto de transferencia para recibir datos de movimientos de inventario.	tipoMovimiento, cantidad, referencia
DTO	SupplyItemResource	Objeto de transferencia que devuelve datos de insumos	id, nombre, unidadMedida, cantidadActual, puntoDeReorden

2.6.3.3. Application Layer

Esta capa maneja los flujos de procesos del inventario mediante el uso de command handlers y event handlers. Su rol es coordinar las operaciones del negocio, orquestando las acciones que se solicitan desde la interfaz y delegando la lógica al dominio. Aquí se asegura que cada comando o evento dispare el proceso adecuado dentro del contexto de inventario.

Tipo	Clase / Nombre	Descripción	Métodos / Comandos manejados
Command Handler	RegisterSupplyItemHandler	Maneja el proceso de registrar un nuevo insumo en el inventario.	- handle(RegisterSupplyItemCommand)
Command Handler	UpdateSupplyItemHandler	Maneja la actualización de datos de un insumo existente.	- handle(UpdateSupplyItemCommand)
Command Handler	RemoveSupplyItemHandler	Maneja la eliminación de un insumo del inventario.	- handle(RemoveSupplyItemCommand)
Command Handler	RegisterInventoryTransactionHandler	Maneja el registro de movimientos de inventario (entrada, salida, ajuste).	- handle(RegisterInventoryTransactionCommand)
Event Handler	LowStockEventHandler	Escucha el evento de dominio StockBajoDetectado y dispara acciones como notificación o reabastecimiento.	- on(StockBajoDetectado)

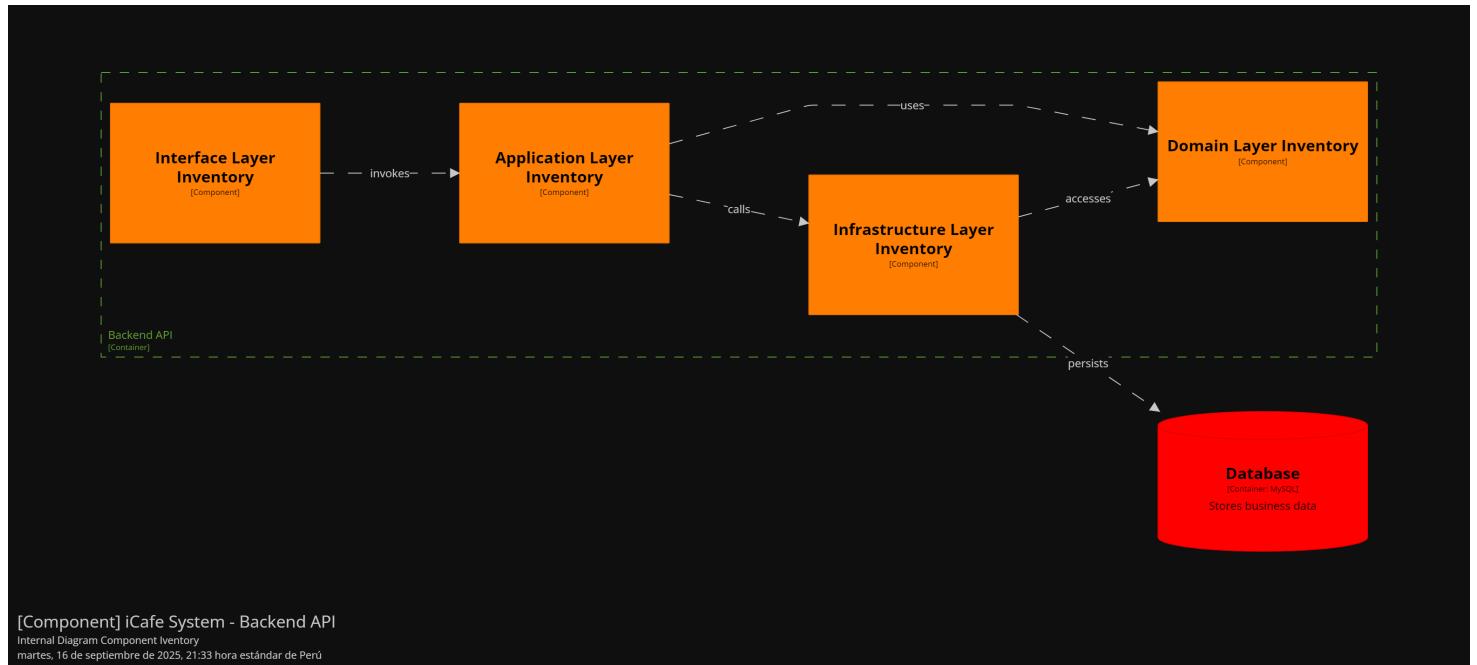
2.6.3.4. Infrastructure Layer

En esta capa se implementa la conexión con servicios externos, principalmente la base de datos. Incluye los repositorios que persisten la información de los insumos y transacciones de inventario utilizando JPA/Hibernate. De esta manera, el dominio se mantiene independiente de la tecnología, mientras la infraestructura garantiza el acceso confiable a los datos.

Tipo	Clase / Nombre	Descripción	Notas Técnicas
Repository Impl	SupplyItemRepository	Implementación de SupplyItemRepository usando JPA/Hibernate para persistir insumos.	Mapea SupplyItem a tabla supply_items .

Tipo	Clase / Nombre	Descripción	Notas Técnicas
Repository Impl	InventoryTransactionRepository	Implementación de <code>InventoryTransactionRepository</code> usando JPA/Hibernate.	Mapea <code>InventoryTransaction</code> a tabla <code>inventory_transactions</code> .

2.6.3.5. Bounded Context Software Architecture Component Level Diagrams

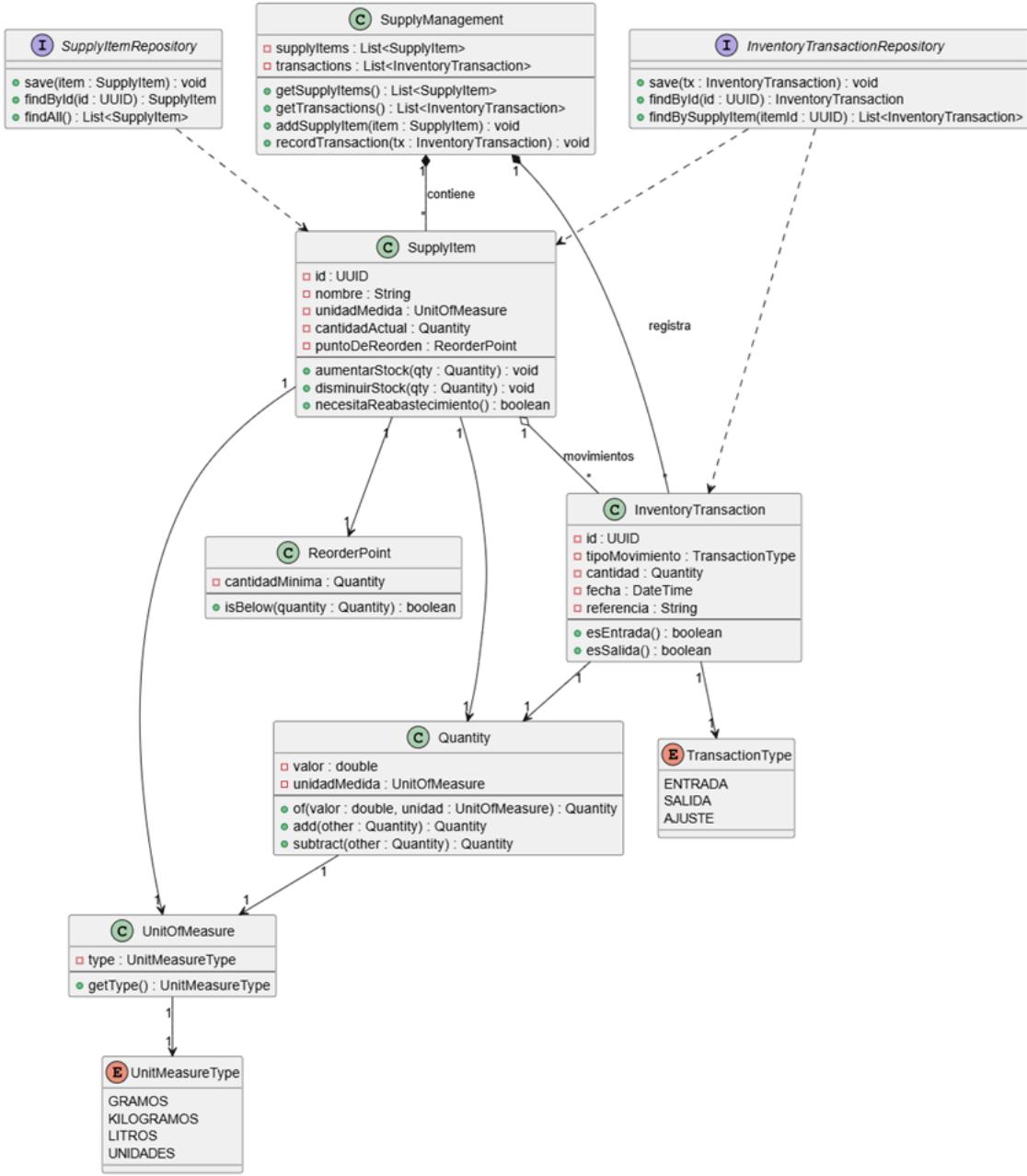


2.6.3.6. Bounded Context Software Architecture Code Level Diagrams

2.6.3.6.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Inventory representa los principales elementos del Domain Layer, incluyendo agregados, entidades, value objects, servicios de dominio, enumeraciones e interfaces de repositorios. Su objetivo es mostrar las relaciones y responsabilidades de cada componente dentro de la gestión de insumos y movimientos de inventario en la cafetería. Para su elaboración se utilizó PlantUML.

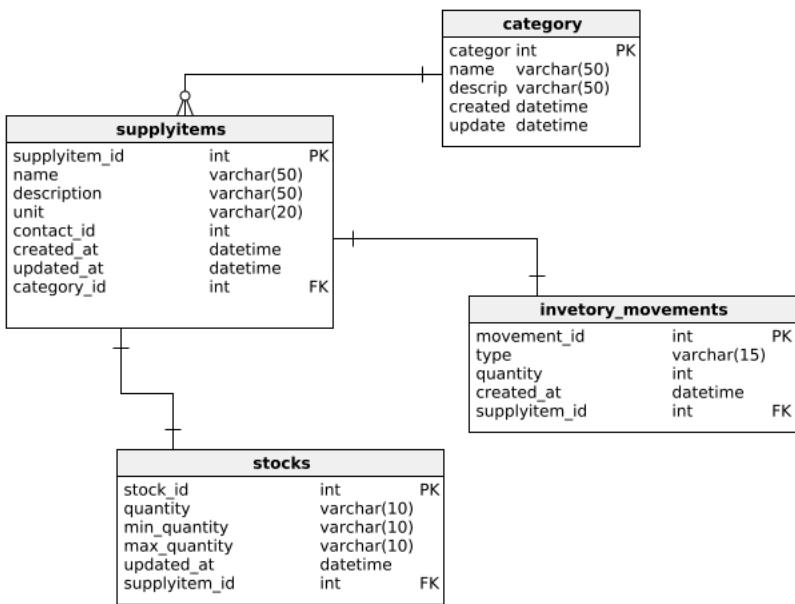
Class Diagram - Domain Layer (Inventory Bounded Context)



2.6.3.6.2. Bounded Context Database Design Diagram

En esta sección se diseña el diagrama de base de datos relacional de el bounded context inventory.

- **Category** : agrupa los insumos en familias o tipos
- **supplyItems**: insumo gestionado en el inventario
- **stocks**: disponibilidad de insumos
- **inventory_movements**: el movimiento dentro del inventario.



2.6.4. Bounded Context: Sales

2.6.4.1. Domain Layer

En esta capa se definen los elementos principales del dominio de ventas. Aquí se modelan los agregados, entidades, value objects y servicios de dominio que representan la lógica central de cómo se gestionan las órdenes, pagos y transacciones comerciales.

Tipo	Clase / Nombre	Descripción	Atributos / Valores
Aggregate	OrderManagement	Gestiona el ciclo de vida completo de una orden como unidad de consistencia.	orders, payments
Aggregate Root	Order	Representa una orden de venta con todos sus items.	id, fecha, estado, total, empleadoId, items
Entity	OrderItem	Representa un item dentro de una orden.	id, productoId, cantidad, precioUnitario, subtotal
Entity	Payment	Registra el pago asociado a una orden.	id, ordenId, monto, metodoPago, fecha, estado
Value Object	Money	Representa valores monetarios con currency.	monto, currency
Value Object	OrderStatus	Define los estados posibles de una orden.	CREADA, PAGADA, EN_PROCESO, COMPLETADA, CANCELADA
Value Object	PaymentMethod	Define los métodos de pago aceptados.	EFFECTIVO, TARJETA, TRANSFERENCIA
Domain Service	OrderCalculationService	Calcula totales, impuestos y descuentos de una orden.	calcularTotal(order)
Domain Service	PaymentProcessingService	Gestiona el procesamiento de pagos.	procesarPago(order, paymentMethod)
Domain Event	OrderCreatedEvent	Evento disparado cuando se crea una nueva orden.	orderId, fecha, total
Domain Event	PaymentReceivedEvent	Evento disparado cuando se recibe un pago.	orderId, monto, metodoPago
Enum	OrderStatusType	Define los tipos de estado de orden.	CREATED, PAID, PROCESSING, COMPLETED, CANCELLED

Tipo	Clase / Nombre	Descripción	Atributos / Valores
Enum	PaymentMethodType	Define los tipos de métodos de pago.	CASH, CARD, TRANSFER

2.6.4.2. Interface Layer

En esta capa se encuentran los controladores y DTOs que sirven como interfaz entre el sistema y los usuarios/clientes externos.

Tipo	Clase / Nombre	Descripción	Métodos / Endpoints principales
Controller	SalesController	Expone endpoints REST para la gestión de ventas.	- POST /sales/orders → crear nueva orden - GET /sales/orders → listar órdenes - GET /sales/orders/{id} → obtener orden específica - PUT /sales/orders/{id} → actualizar orden - DELETE /sales/orders/{id} → cancelar orden
Controller	PaymentController	Maneja operaciones relacionadas con pagos.	- POST /sales/payments → procesar pago - GET /sales/payments → listar pagos - GET /sales/payments/{id} → detalle de pago
DTO	OrderRequest	Objeto para recibir datos de creación de orden.	items, empleadoid, metodoPago
DTO	OrderResponse	Objeto para devolver datos de orden.	id, fecha, estado, total, items
DTO	PaymentRequest	Objeto para recibir datos de pago.	ordenId, monto, metodoPago
DTO	PaymentResponse	Objeto para devolver datos de pago.	id, ordenId, monto, estado, fecha

2.6.4.3. Application Layer

Esta capa maneja los flujos de procesos de ventas mediante command handlers y event handlers.

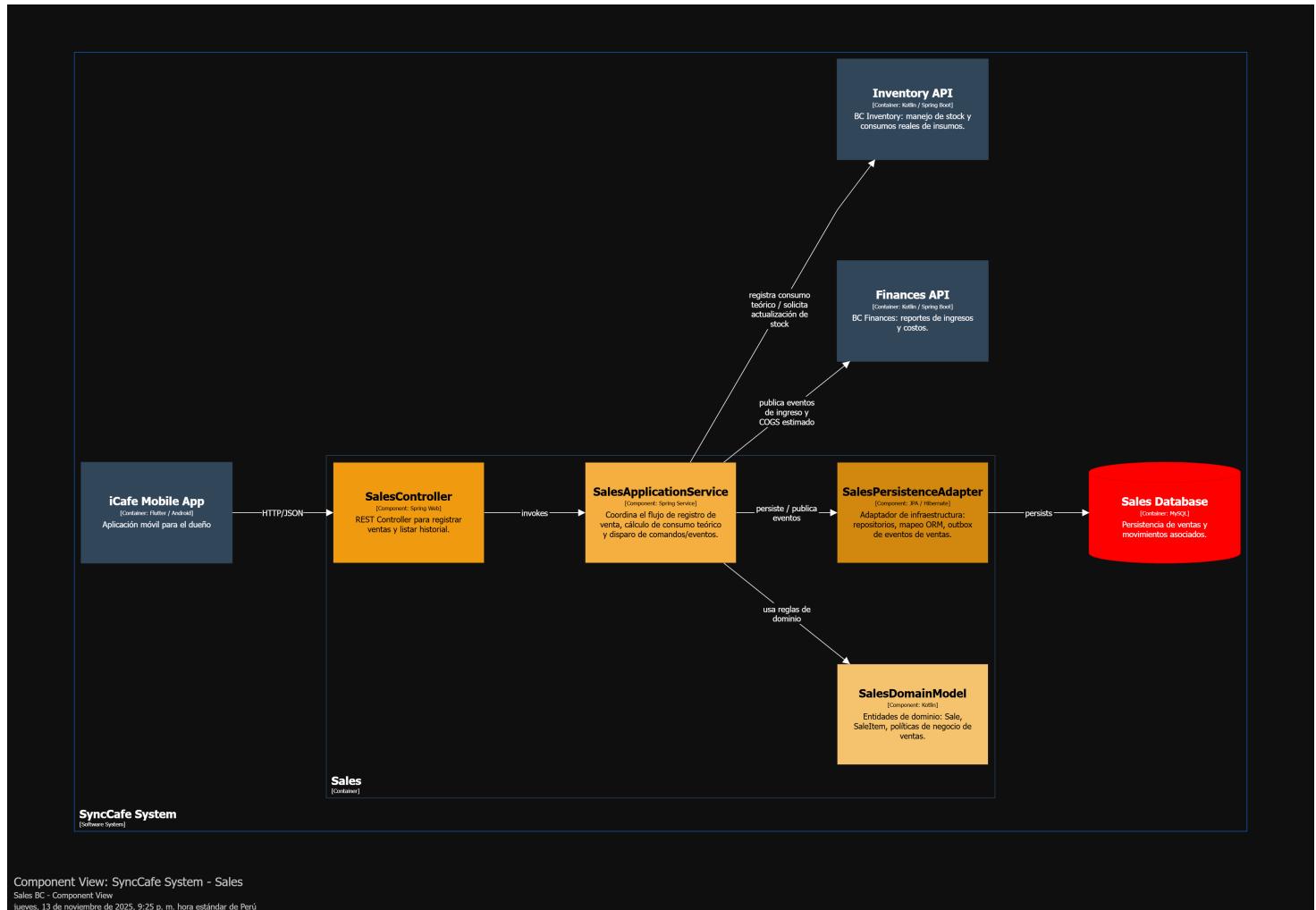
Tipo	Clase / Nombre	Descripción	Métodos / Comandos manejados
Command Handler	CreateOrderHandler	Maneja la creación de una nueva orden.	- handle(CreateOrderCommand)
Command Handler	ProcessPaymentHandler	Maneja el procesamiento de pagos.	- handle(ProcessPaymentCommand)
Command Handler	UpdateOrderStatusHandler	Maneja la actualización del estado de una orden.	- handle(UpdateOrderStatusCommand)
Command Handler	CancelOrderHandler	Maneja la cancelación de órdenes.	- handle(CancelOrderCommand)
Event Handler	InventoryUpdateEventHandler	Escucha eventos de actualización de inventario.	- on(InventoryUpdatedEvent)
Event Handler	PaymentConfirmedEventHandler	Escucha eventos de confirmación de pago.	- on(PaymentConfirmedEvent)

2.6.4.4. Infrastructure Layer

En esta capa se implementa la conexión con servicios externos y la persistencia de datos.

Tipo	Clase / Nombre	Descripción	Notas Técnicas
Repository Impl	OrderRepositoryImpl	Implementación de OrderRepository usando JPA/Hibernate.	Mapea Order a tabla orders
Repository Impl	PaymentRepositoryImpl	Implementación de PaymentRepository usando JPA/Hibernate.	Mapea Payment a tabla payments
External Service	InventoryServiceClient	Cliente para comunicación con servicio de inventario.	REST client para Inventory API
External Service	PaymentGatewayClient	Cliente para comunicación con pasarela de pagos.	Integración con Stripe/PayPal

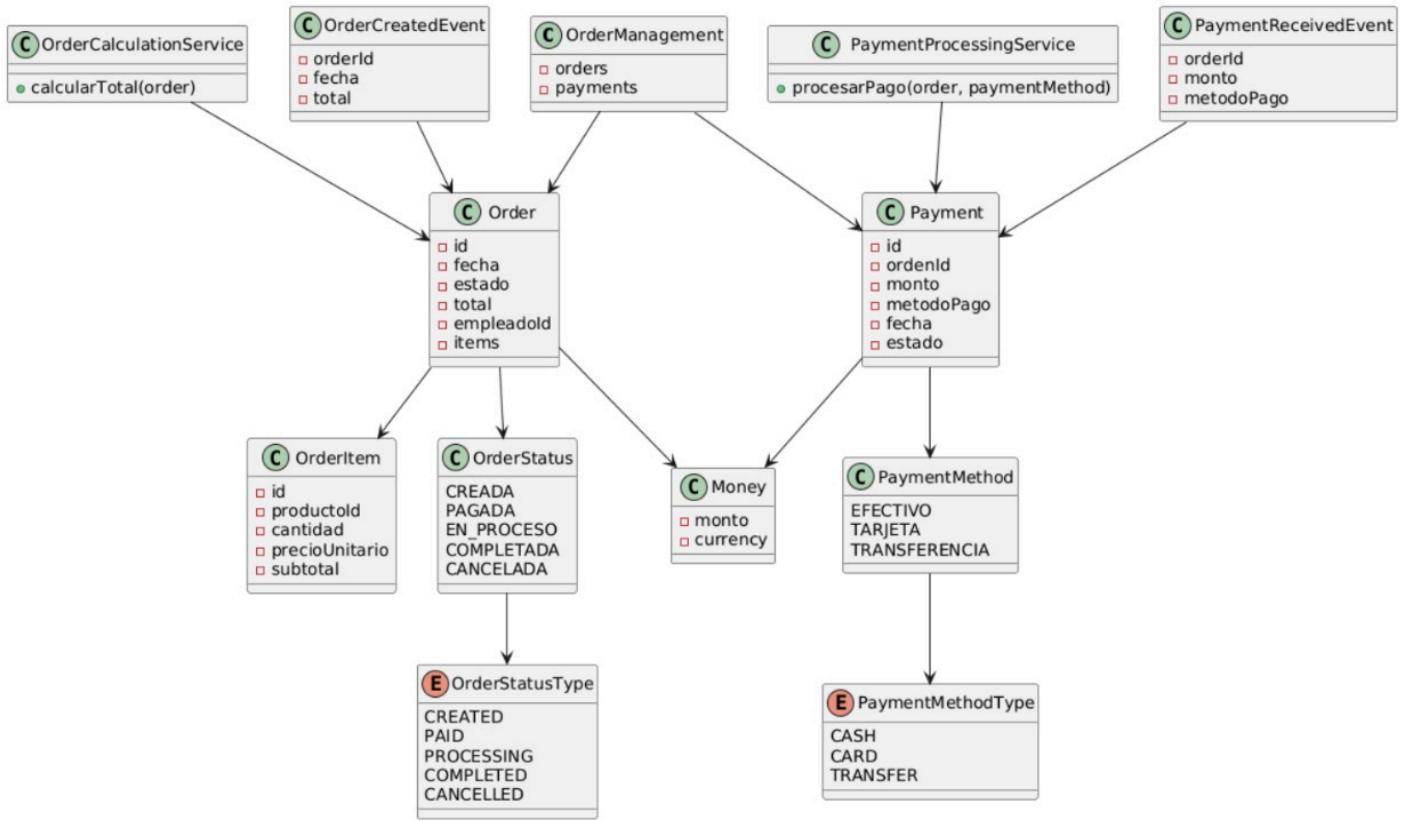
2.6.4.5. Bounded Context Software Architecture Component Level Diagrams



2.6.4.6. Bounded Context Software Architecture Code Level Diagrams

2.6.4.6.1. Bounded Context Domain Layer Class Diagrams

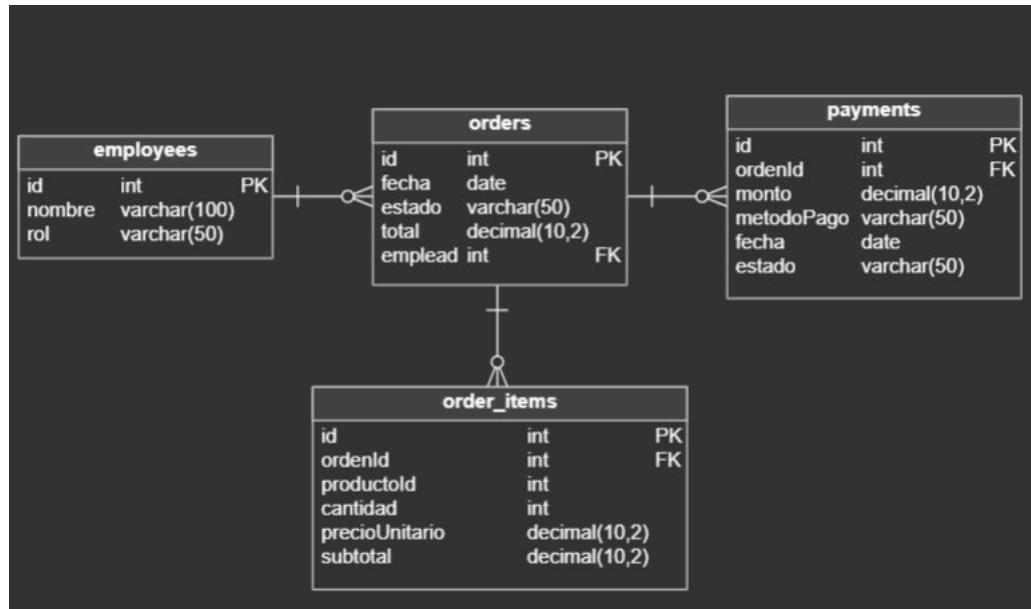
El diagrama de clases del bounded context Sales representa los principales elementos del Domain Layer, mostrando las relaciones entre órdenes, ítems de orden, pagos y servicios de dominio. El diagrama fue elaborado usando PlantUML.



2.6.4.6.2. Bounded Context Database Design Diagram

En esta sección se diseña el diagrama de base de datos relacional para el bounded context Sales.

- **orders**: almacena las órdenes de venta
- **order_items**: almacena los items de cada orden
- **payments**: registra los pagos asociados a las órdenes
- **employees**: información de empleados que realizan ventas



2.6.5. Bounded Context: Finances

El bounded context Finances se encarga de la gestión financiera de la cafetería, incluyendo el registro de ingresos, gastos, control de costos, generación de reportes financieros y análisis de rentabilidad.

2.6.5.1. Domain Layer

En esta capa se definen los elementos principales del dominio financiero. Aquí se modelan los agregados, entidades, value objects y servicios de dominio que representan la lógica central.

Tipo	Clase / Nombre	Descripción	Atributos / Valores
Aggregate	FinancialManagement	Gestiona las operaciones financieras como una unidad de consistencia.	transactions, budgets, reports
Aggregate Root	Transaction	Representa una transacción financiera (ingreso o gasto).	id, tipo, monto, fecha, categoría, descripción, referencia
Entity	Budget	Define un presupuesto para un período y categoría específicos.	id, categoría, montoAsignado, montoUtilizado, período
Entity	FinancialReport	Representa un informe financiero generado para un período.	id, tipo, período, fechaGeneración, datos
Value Object	Money	Representa valores monetarios con su moneda.	valor, moneda
Value Object	Period	Define un período de tiempo para análisis financiero.	fechalinicio, fechaFin
Value Object	Category	Categoría para clasificar transacciones financieras.	id, nombre, tipo
Domain Service	ProfitCalculationService	Calcula la rentabilidad basada en ingresos y gastos.	calcularRentabilidad(period)
Domain Service	BudgetAnalysisService	Analiza el cumplimiento de presupuestos y desviaciones.	analizarPresupuesto(budget, period)
Domain Event	TransactionRegisteredEvent	Evento disparado cuando se registra una nueva transacción.	transactionId, tipo, monto, fecha
Domain Event	BudgetExceededEvent	Evento disparado cuando un gasto supera el presupuesto asignado.	budgetId, categoría, montoExcedido
Enum	TransactionType	Define los tipos de transacciones financieras.	INGRESO, GASTO
Enum	ReportType	Define los tipos de informes financieros disponibles.	BALANCE, FLUJO_CAJA, RENTABILIDAD, PRESUPUESTO

2.6.5.2. Interface Layer

En esta capa se encuentran los controladores y DTOs que sirven como interfaz entre el sistema y los usuarios o clientes externos.

Tipo	Clase / Nombre	Descripción	Métodos / Endpoints principales
Controller	TransactionController	Expone endpoints REST para la gestión de transacciones financieras.	<ul style="list-style-type: none"> - POST /finances/transactions → registrar transacción - GET /finances/transactions → listar transacciones - GET /finances/transactions/{id} → obtener detalle - PUT /finances/transactions/{id} → actualizar transacción - DELETE /finances/transactions/{id} → eliminar transacción
Controller	BudgetController	Maneja operaciones relacionadas con presupuestos.	<ul style="list-style-type: none"> - POST /finances/budgets → crear presupuesto - GET /finances/budgets → listar presupuestos - GET /finances/budgets/{id} → detalle de presupuesto - PUT /finances/budgets/{id} → actualizar presupuesto

Tipo	Clase / Nombre	Descripción	Métodos / Endpoints principales
Controller	ReportController	Gestiona la generación y consulta de informes financieros.	- POST /finances/reports → generar informe - GET /finances/reports → listar informes - GET /finances/reports/{id} → obtener informe específico
DTO	TransactionRequest	Objeto para recibir datos de creación de transacción.	tipo, monto, categoría, descripción, referencia
DTO	TransactionResponse	Objeto para devolver datos de transacción.	id, tipo, monto, fecha, categoría, descripción
DTO	BudgetRequest	Objeto para recibir datos de creación de presupuesto.	categoría, montoAsignado, fechalinicio, fechaFin
DTO	ReportRequest	Objeto para solicitar generación de informes.	tipo, fechalinicio, fechaFin

2.6.5.3. Application Layer

Esta capa maneja los flujos de procesos financieros mediante command handlers y event handlers.

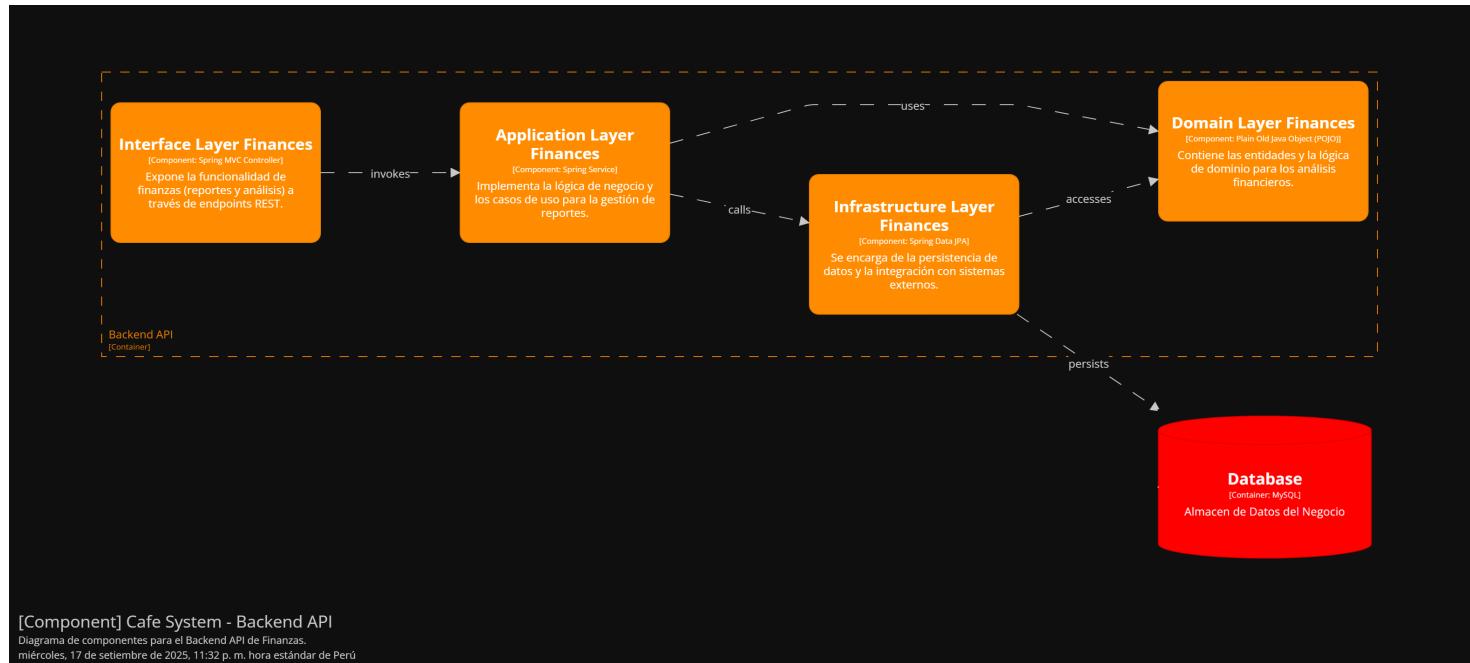
Tipo	Clase / Nombre	Descripción	Métodos / Comandos manejados
Command Handler	RegisterTransactionHandler	Maneja el registro de nuevas transacciones financieras.	- handle(RegisterTransactionCommand)
Command Handler	CreateBudgetHandler	Maneja la creación de nuevos presupuestos.	- handle(CreateBudgetCommand)
Command Handler	GenerateFinancialReportHandler	Maneja la generación de informes financieros.	- handle(GenerateFinancialReportCommand)
Command Handler	UpdateBudgetHandler	Maneja la actualización de presupuestos existentes.	- handle(UpdateBudgetCommand)
Event Handler	SaleCompletedEventHandler	Escucha eventos de ventas completadas para registrar ingresos.	- on(SaleCompletedEvent)
Event Handler	InventoryPurchaseEventHandler	Escucha eventos de compras de inventario para registrar gastos.	- on(InventoryPurchaseEvent)
Event Handler	BudgetExceededEventHandler	Maneja acciones cuando se excede un presupuesto.	- on(BudgetExceededEvent)

2.6.5.4. Infrastructure Layer

En esta capa se implementa la conexión con servicios externos, principalmente la base de datos. Incluye los repositorios que persisten la información financiera utilizando JPA/Hibernate.

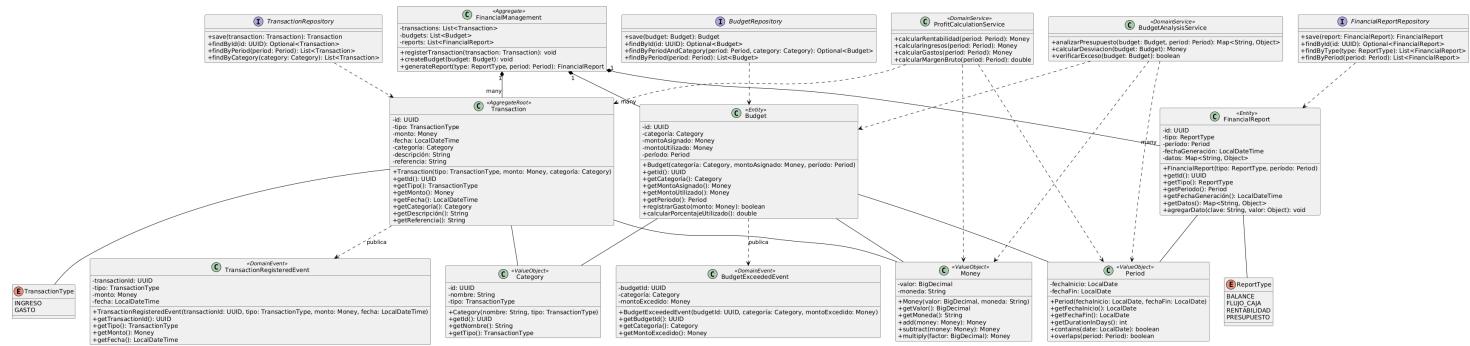
Tipo	Clase / Nombre	Descripción	Notas Técnicas
Repository Impl	TransactionRepositoryImpl	Implementación de TransactionRepository usando JPA/Hibernate.	Mapea Transaction a tabla transactions
Repository Impl	BudgetRepositoryImpl	Implementación de BudgetRepository usando JPA/Hibernate.	Mapea Budget a tabla budgets
Repository Impl	FinancialReportRepositoryImpl	Implementación de FinancialReportRepository usando JPA/Hibernate.	Mapea FinancialReport a tabla financial_reports
External Service	SalesServiceClient	Cliente para comunicación con servicio de ventas.	REST client para Sales API
External Service	InventoryServiceClient	Cliente para comunicación con servicio de inventario.	REST client para Inventory API
External Service	ReportGenerationService	Servicio para generación de reportes en diferentes formatos (PDF, Excel).	Utiliza bibliotecas como Apache POI, iText

2.6.5.5. Bounded Context Software Architecture Component Level Diagrams



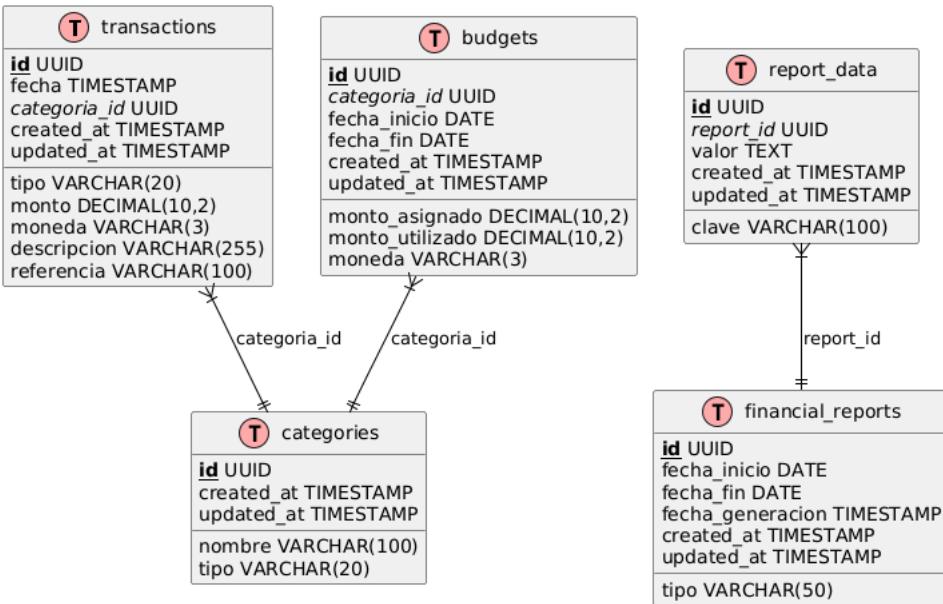
2.6.5.6. Bounded Context Software Architecture Code Level Diagrams

2.6.5.6.1. Bounded Context Domain Layer Class Diagrams



2.6.5.6.2. Bounded Context Database Design Diagram

- **transactions**: almacena todas las transacciones financieras (ingresos y gastos)
- **categories**: categorías para clasificar transacciones
- **budgets**: presupuestos asignados por categoría y período
- **financial_reports**: informes financieros generados
- **report_data**: datos específicos incluidos en cada informe



Capítulo III: Solution UI/UX Design

3.1. Product design

Esta sección describe el diseño del producto para **iCafe**, una plataforma digital creada para optimizar la gestión de cafeterías. El enfoque principal es la simplicidad, eficiencia y accesibilidad, pensada para propietarios y administradores sin necesidad de conocimientos técnicos avanzados.

3.1.1. Style Guidelines

Las guías de estilo definen la identidad visual de **iCafe**, asegurando consistencia en pantallas y componentes.

Se adopta un **enfoque minimalista** con colores inspirados en el café, tipografías modernas e íconos discretos.

De esta manera, la interfaz es **profesional, amigable y accesible**, alineada con la meta de simplificar operaciones.

3.1.1.1. General Style Guidelines

Esta sección presenta las Directrices Generales de Estilo, que definen la identidad visual de la aplicación.

Logo:

Se diseñó un logo que, a simple vista, se percibe como una taza de café, ya que nuestro público objetivo son personas con cafeterías.



Paleta de Colores:

Transmiten una atmósfera cálida y acogedora:

- Gris Oscuro: #51514F → textos y bordes.
- Gris Medio: #929190 → textos secundarios, íconos.
- Gris Claro: #D9D9D9 → fondos sutiles.
- Marrón Oscuro: #8F4C32 → botones principales, encabezados.

- Marrón Medio: #77574B → navegación.
- Verde Oliva: #695E30 → indicadores de éxito, sostenibilidad.
- Durazno Claro: #FFDBCE → tarjetas, secciones de ventas/inventario.
- Rosa Suave: #FFF8F6 → fondos claros, formularios.
- Blanco: #FFFFFF → espacios principales, limpieza visual.



Los botones:

- **Primarios:** marrón oscuro (#8F4C32).
 - **Secundarios:** durazno claro (#FFDBCE).
 - **Terciarios:** verde oliva (#695E30).
- Todos con esquinas redondeadas y aspecto táctil.



Tipografía:

- Fuente Principal: **Poppins** (sans-serif) → encabezados, cuerpo de texto y botones.
- Fuente Secundaria: **Pridi** (serif) → subtítulos o textos decorativos.

Directrices:

- Encabezados: Poppins bold (24--32 pt).
- Texto principal: Poppins regular (14--16 pt).
- Contraste sutil con Pridi en etiquetas o subtítulos.

POPPINS
poppins

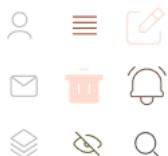
PRIDI
pridi

POPPINS
poppins

PRIDI
pridi

Íconos:

- Íconos minimalistas en gris medio (#929190) o colores de acento.
- Ejemplos: usuario, menú, editar, notificaciones, buscar.
- Logotipo: taza de café humeante en tonos marrones.
- Ilustraciones simples y planas, sin sobrecargar la interfaz.



Diseño y Espaciado:

- Fondos claros (#FFF8F6 / #FFFFFF).
- Amplio espacio en blanco.
- Botones con forma de píldora.
- Campos con esquinas redondeadas y sombras suaves.
- Diseño **mobile**.

Accesibilidad:

- Tema claro por defecto.
- Contraste alto según normas **WCAG**.
- Texto escalable.
- Compatible con daltonismo.
- Áreas de interacción mínimas: **48x48 dp**.

3.1.2. Information Architecture

La arquitectura de información organiza el contenido de manera lógica para facilitar acceso rápido a funciones esenciales: registro, inicio de sesión, panel de control y herramientas de operación.

Se prioriza la estructura jerárquica, optimizada para dispositivos móviles y orientada a flujos de tareas como registro de ventas o seguimiento de inventario.

3.1.2.1. Organization Systems

La aplicación se estructura a través de diferentes sistemas de organización, según el tipo de información y la necesidad del usuario.

Jerárquica:

- **Inventario:** Los productos (café, insumos, utensilios) se listan según categorías definidas: Materias primas, Bebidas, Complementos.
- **Personal:** Lista de empleados organizada por roles (barista, cajero, administrador).
- **Transacciones:** Historial de ventas presentado cronológicamente, mostrando primero las más recientes.
- **Panel de control:** Muestra indicadores clave (ventas diarias, stock bajo, desempeño del personal).

Secuencial:

- **Registro de ventas:** Paso a paso para seleccionar producto, cantidad y medio de pago.
- **Registro de empleados:** Proceso que guía desde datos personales hasta asignación de rol.
- **Onboarding inicial:** Explicación guiada desde el inicio de sesión hasta la configuración básica de la cafetería.

Matricial:

- **Gestión de sedes:** Cuando la cafetería posee más de un local, el sistema muestra un panel tipo cuadrícula con cada sede. Desde allí se puede acceder al inventario, ventas y personal de forma independiente.
- **Calendario de turnos:** Agenda visual que organiza horarios de empleados según días y horas.

3.1.2.2. Labelling Systems

Se aplicará un etiquetado simple y claro, con el formato **verbo + sustantivo**, acompañado de íconos.

- Ejemplos: *Registrar Ventas, Gestionar Inventario, Añadir Empleado*.
- Campos de entrada con etiquetas breves: *Nombre de Cafetería, Correo electrónico, Contraseña*.
- Jerarquía visual mediante tipografías diferenciadas para títulos, subtítulos y contenido.
- Todo el contenido estará localizado en español, con planes de traducción futura al inglés.

3.1.2.3. SEO Tags and Meta Tags

Para la Landing Page:

- **Title:** iCafe – Gestión inteligente para tu cafetería
- **Description:** Plataforma digital para administrar ventas, inventario y personal en tu cafetería de manera sencilla y eficiente.
- **Keywords:** cafeterías, gestión de inventario, control de ventas, administración de café, dueños de cafeterías.
- **Author:** Equipo iCafe

Para la Aplicación Móvil (ASO):

- **App Title:** iCafe
- **App Subtitle:** Administración de cafeterías en tu bolsillo
- **App Keywords:** café, cafetería, ventas, inventario, gestión
- **App Description:** iCafe es la app que ayuda a dueños de cafeterías a organizar su negocio de forma eficiente: controla ventas, inventario, personal y proveedores desde tu celular.

3.1.2.4. Searching Systems

El sistema de búsqueda permitirá localizar rápidamente información clave:

- **Global:** búsqueda transversal de productos, empleados y transacciones.
- **Facetada:** filtros por categorías de producto (café, insumos, postres), proveedores o costos.
- **Autocompletado:** sugerencias al escribir precios, nombres de productos o empleados.
- **Resultados:** presentados en listas con miniaturas (para productos) o tarjetas (para empleados).
- **Offline:** búsqueda indexada para casos sin conexión a internet.

3.1.2.5. Navigation Systems

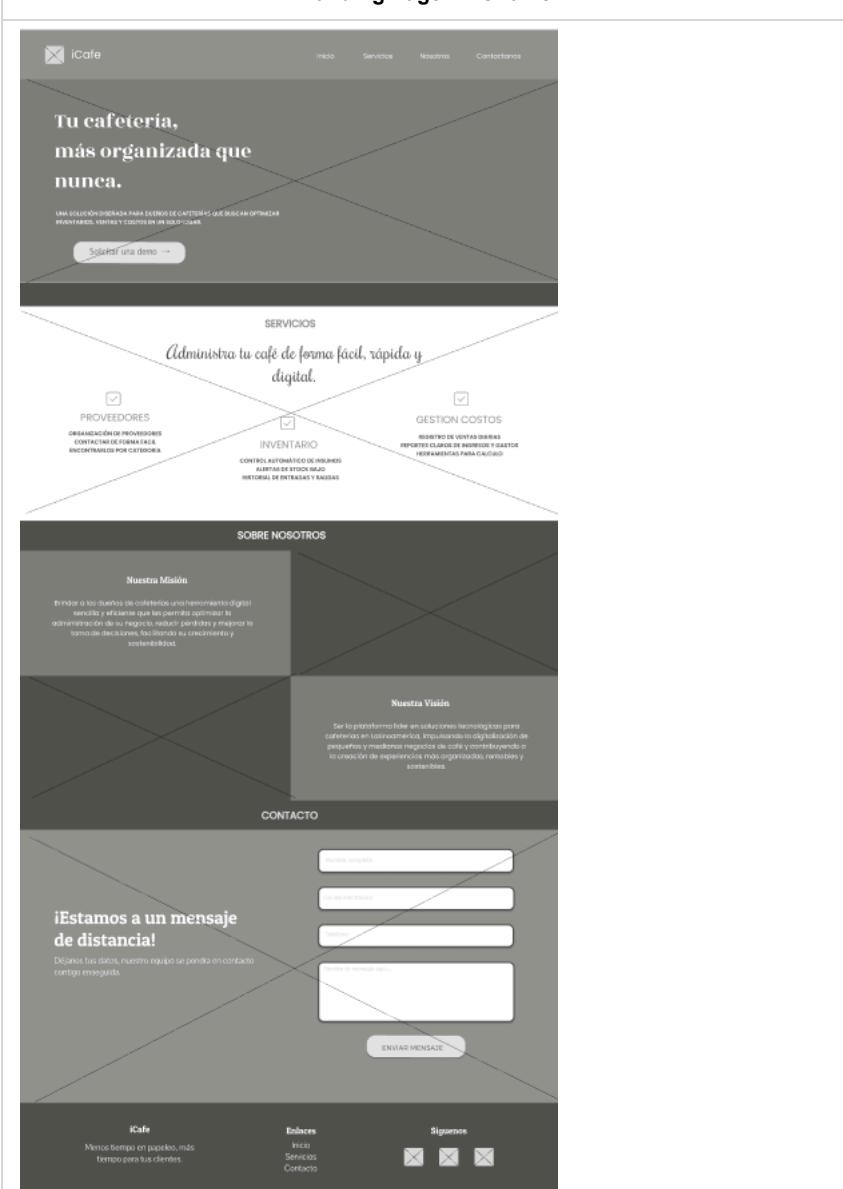
Se implementan diferentes formas de navegación, adaptadas a la web y la app móvil:

- **Landing Page:** navegación superior con secciones: *Inicio, Servicios, Beneficios, Costos, Contacto*.
- **Aplicación móvil:** navegación principal mediante pestañas inferiores: *Panel, Inventario, Personal, Gestión de Costos, Registro de compra y venta*.
- **Secundaria:** menús desplegables y tarjetas para acceder a funciones dentro de cada módulo (ejemplo: *Inventario → Materias Primas*).
- **Gestos:** *swipe lateral* para cambiar entre listas de ventas o turnos de empleados.

3.1.3. Landing Page UI Design

3.1.3.1. Landing Page Wireframe

En esta sección se presentan los wireframes de la landing page, los cuales representan la estructura básica de la interfaz. El objetivo es definir la distribución de los elementos principales que comunicarán de manera clara los servicios de la aplicación y facilitarán la navegación a los usuarios.

Landing Page Wireframe	Landing Page Wireframe Responsive
 <p>The wireframe illustrates the layout of the landing page. At the top, there's a header with the logo 'iCafe' and navigation links for 'Inicio', 'Servicios', 'Nosotros', and 'Contacto'. Below the header is a main section with the heading 'Tu cafetería, más organizada que nunca.' and a sub-section 'Una solución organizada para dueños de cafeterías que buscan optimizar inventarios, ventas y costos en un solo lugar.' A button labeled 'Solicita una demo →' is present. This is followed by a 'SERVICIOS' section with two main categories: 'PROVEEDORES' (checkboxes for 'Gestión de proveedores', 'Contratación de forma fácil', and 'Encuentros por catálogo') and 'INVENTARIO' (checkboxes for 'Control automático de inventario', 'Gestionar stock rápidamente', and 'Historial de inventario y salidas'). Below these are 'GESTIÓN COSTOS' (checkboxes for 'Análisis de ventas diarias', 'Reportes de ingresos y gastos', and 'Herramientas para ahorrar'). The next section is 'SOBRE NOSOTROS' with 'Nuestra Misión' and 'Nuestra Visión' sub-sections. 'Nuestra Misión' describes the mission to provide coffee shop owners with a digital tool to manage inventories, sales, and costs. 'Nuestra Visión' aims to be a platform that facilitates communication between coffee shops and suppliers, promotes digitalization, and creates better experiences. The 'CONTACTO' section features a form for sending a message with fields for 'Nombre completo', 'Email electrónico', and 'Contenido de tu mensaje aquí...', and a 'ENVIAR MENSAJE' button. The footer contains social media icons for 'Facebook', 'Twitter', 'Instagram', and 'LinkedIn', along with the text 'iCafe' and 'Menos tiempo en papeles, más tiempo para tus clientes.'</p>	 <p>The responsive wireframe shows the landing page's layout on a mobile device. The header 'iCafe' and navigation links are at the top. The main headline 'Tu cafetería, más organizada que nunca.' is followed by a brief description. The 'SERVICIOS' section is collapsed into a single button labeled 'Ver servicios'. The 'SOBRE NOSOTROS' section includes 'Nuestra Misión' and 'Nuestra Visión'. The 'CONTACTO' section is also collapsed into a single button labeled 'Envíame un mensaje'. The footer includes social media icons and the text 'iCafe'.</p>

-Link: <https://www.figma.com/design/c61JnVMaZZcReGqcVzwikH/Untitled?node-id=0-1&t=gB15ysllrlfRAw-1>

3.1.3.2. Landing Page Mock-up

Aquí se muestran los mock-ups de la landing page, donde se integran aspectos visuales como colores, tipografía e imágenes. Estos diseños permiten visualizar con mayor fidelidad cómo se verá la página final y cómo transmitirá la identidad de la marca enfocada en cafeterías.

LandingPage Mock-up	LandingPage Mock-up Responsive
<p>SERVICIOS Administra tu café de forma fácil, rápida y digital.</p> <p>PROVEEDORES Gestión de proveedores Entregas directas a tu local Encuentra proveedores para tu negocio</p> <p>INVENTARIO Control automático de inventario Alarma de stock bajo</p> <p>GESTIÓN COSTOS Reportes de ventas y gastos Reportes claves de negocio y análisis</p> <p>CONTACTO ¡Estamos a un mensaje de distancia! Envíanos tu mensaje y te responderemos en el menor tiempo posible.</p>	<p>MISIÓN Administrá tu café de forma fácil, rápida y digital.</p> <p>PROVEEDORES Gestión de proveedores Entregas directas a tu local Encuentra proveedores para tu negocio</p> <p>INVENTARIO Control automático de inventario Alarma de stock bajo</p> <p>GESTIÓN COSTOS Reportes de ventas y gastos Reportes claves de negocio y análisis</p> <p>SOBRE NOSOTROS Nuestra Misión Brindar a los dueños de cafeterías una herramienta digital, un salto y evolución que les permita operar mejor su negocio, reducir el trabajo y mejorar la forma en que administran su negocio y sus recursos.</p> <p>Nuestra Visión Ser la plataforma más líder en soluciones tecnológicas para cafeterías en Latinoamérica, ofreciendo una digitalización de procesos y relaciones negociales de calidad a los dueños de negocios y administradores de cafeterías.</p> <p>CONTACTO ¡Estamos a un mensaje de distancia! Envíanos tu mensaje y te responderemos en el menor tiempo posible.</p>

-Link: <https://www.figma.com/design/c61JnVMaZZcReGqcVzwikH/Untitled?node-id=0-1&t=gB15yslrlfRAw-1>

3.1.4. Mobile Applications UX/UI Design

Esta sección aborda el diseño de la experiencia de usuario (UX) y la interfaz gráfica (UI) de la aplicación móvil. Se busca crear una plataforma intuitiva, atractiva y funcional que permita a los administradores de cafeterías gestionar sus negocios de forma sencilla y eficiente.

3.1.4.1. Mobile Applications Wireframes

Se presentan los wireframes de la aplicación móvil, que representan la base estructural de cada pantalla. Estos bocetos iniciales permiten definir la jerarquía de información y la organización de los elementos.

Wireframes de iCafe

Se muestran las pantallas en baja fidelidad

Wireframes de iCafe



Link: <https://www.figma.com/design/c61JnVMaZZcReGqcVzwikH/Untitled?node-id=0-1&t=gB15ysllrLlfRAw-1>

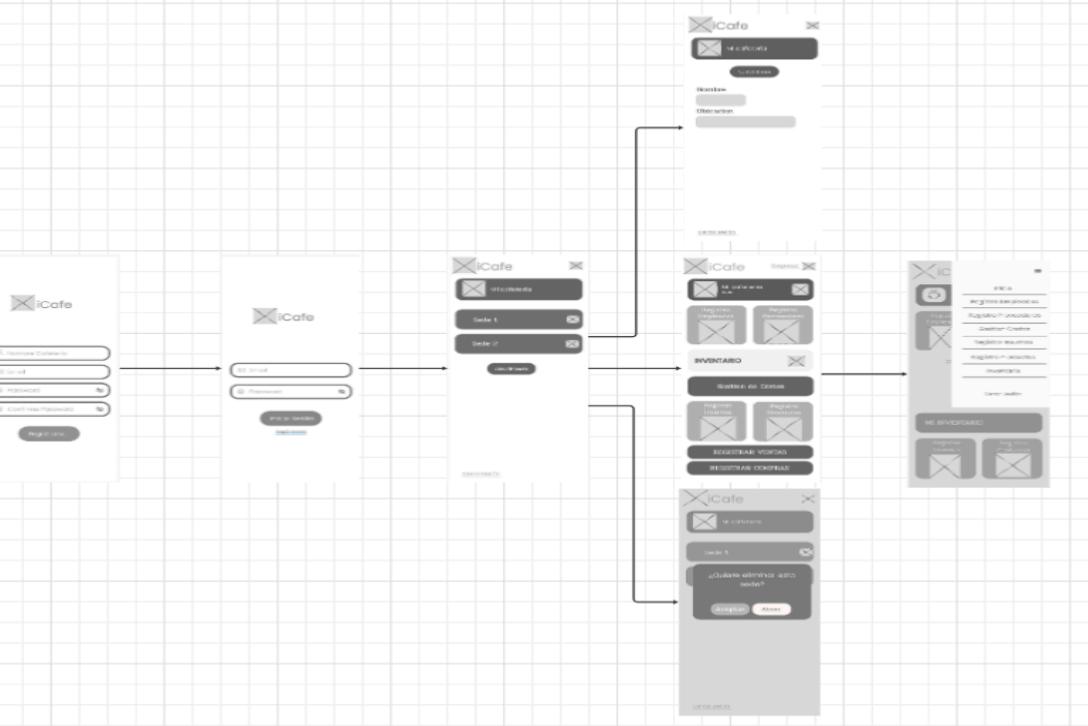
3.1.4.2. Mobile Applications Wireflow Diagrams

En este apartado se incluyen los wireflow diagrams, los cuales combinan los wireframes con el flujo de navegación. Esto permite visualizar cómo interactúa el usuario al desplazarse por las distintas pantallas de la aplicación.

User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo registrarme e iniciar sesión antes de visualizar el panel principal.

User Persona: Dueños de cafeterias.



User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo agregar, editar y eliminar empleados cuando yo quiera.



User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo agregar, editar y eliminar Insumos cuando yo quiera.

User Persona: Dueños de cafeterias.



User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo agregar, editar y eliminar proveedor cuando yo quiera.



User Persona: Dueños de cafeterias.

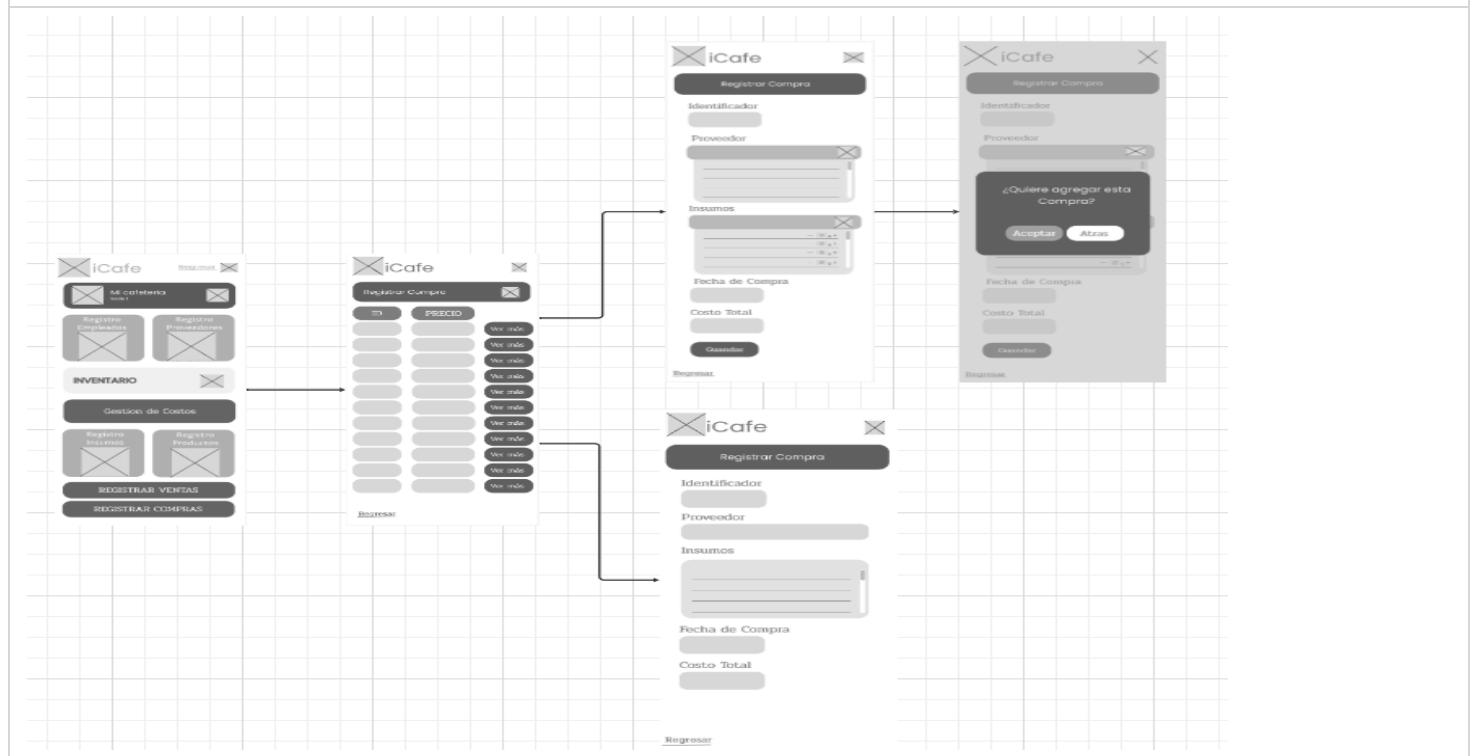
User Goal: Como usuario, deseo agregar, editar y eliminar Producto cuando yo quiera.

User Persona: Dueños de cafeterias.



User Persona: Dueños de cafeterias.

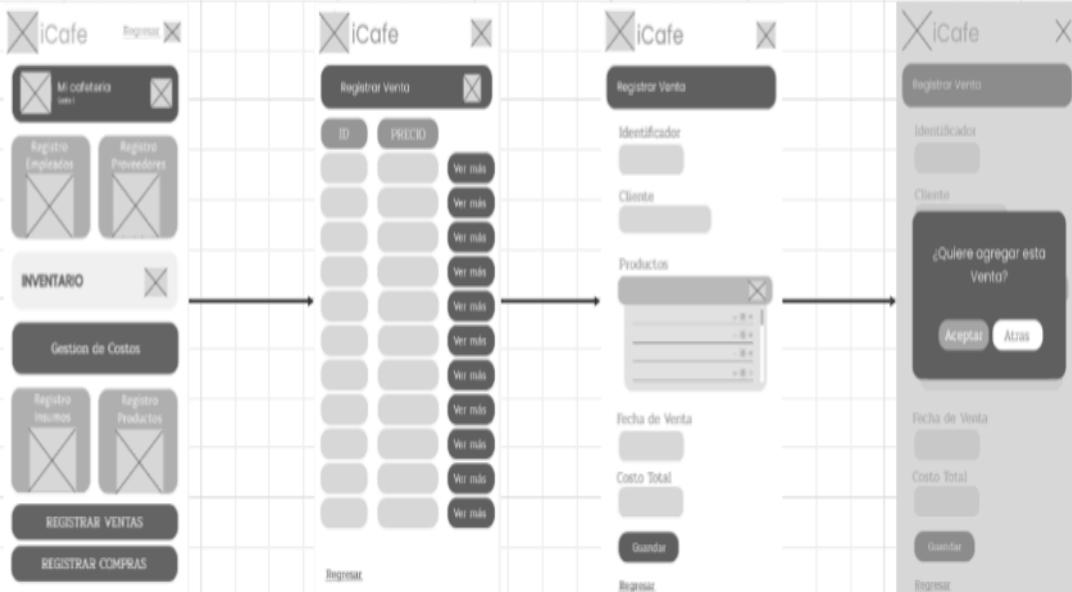
User Goal: Como usuario, deseo registrar compras para llevar un recuento de mis inversiones.



User Persona: Dueños de cafeterias.

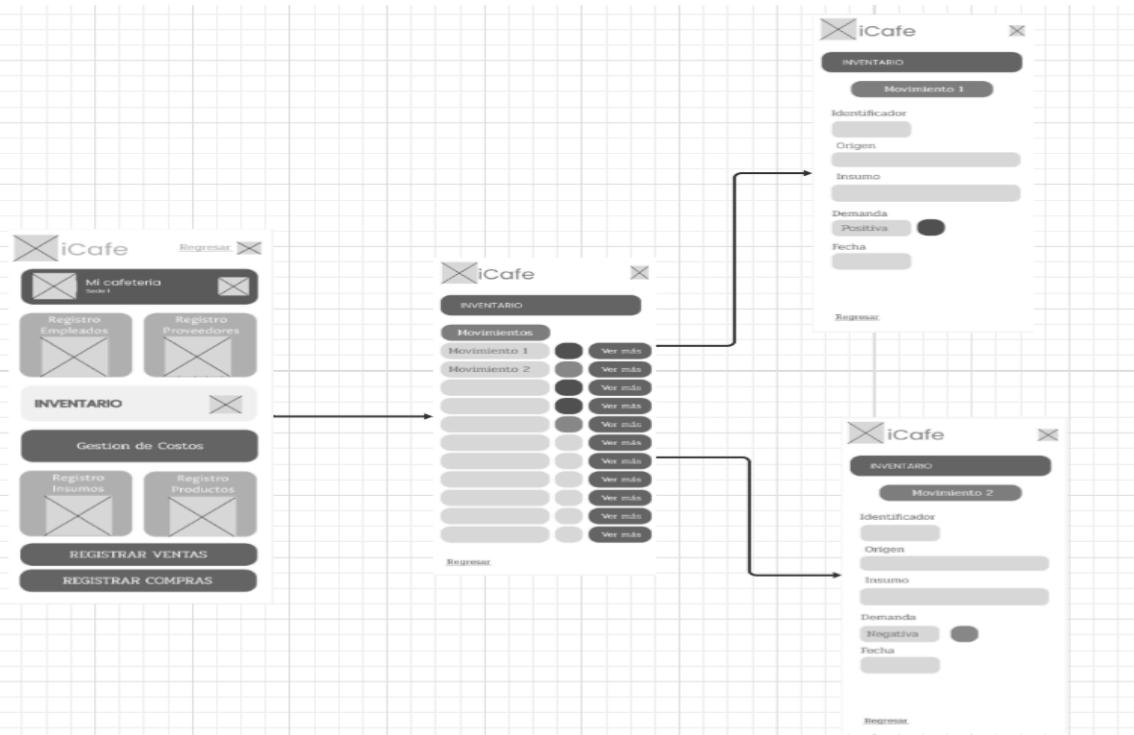
User Goal: Como usuario, deseo registrar ventas para llevar un recuento de mis ganancias.

User Persona: Dueños de cafeterias.



User Persona: Dueños de cafeterias.

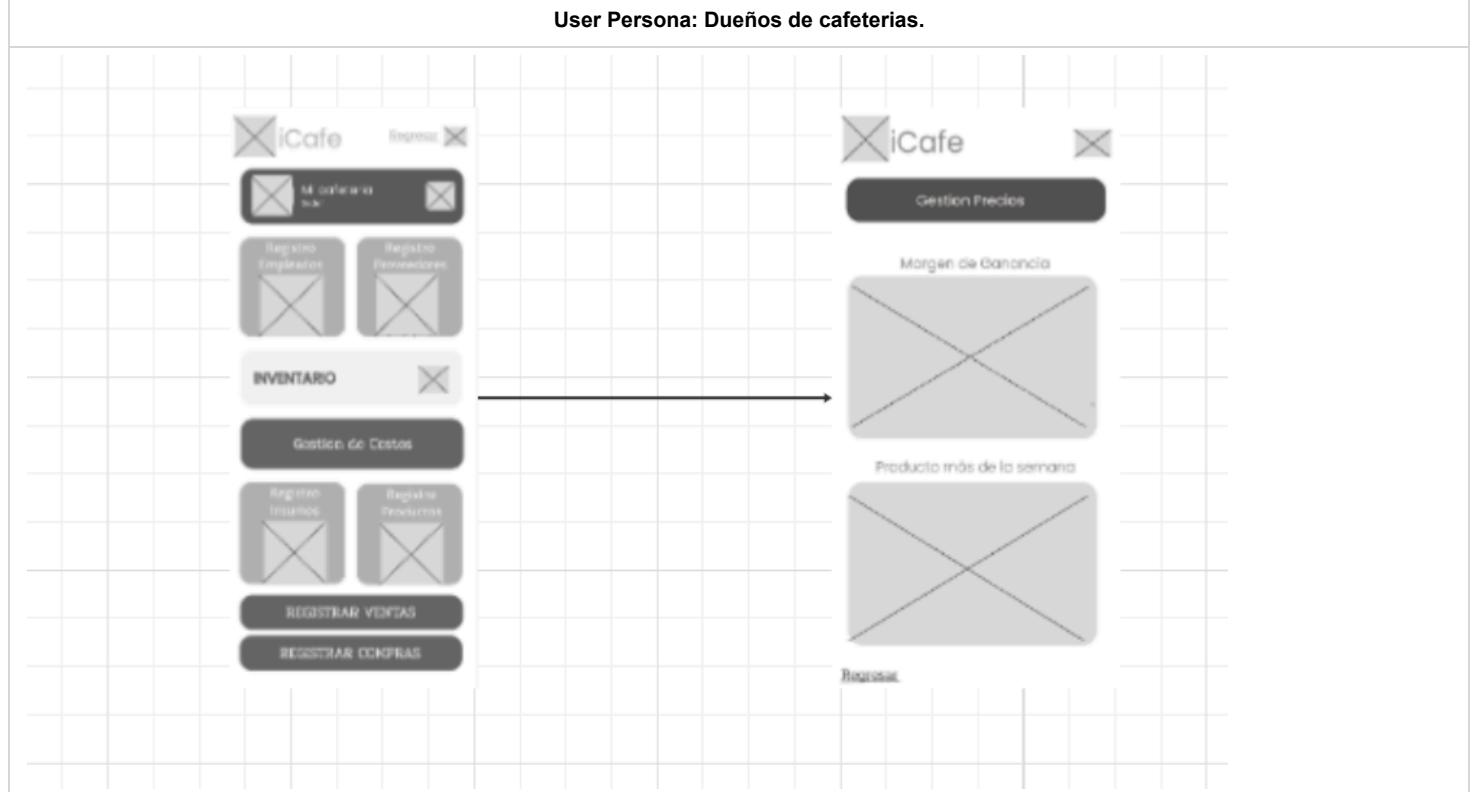
User Goal: Como usuario, deseo una sección de inventario donde tenga el recuento de movimientos.



User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo una sección de Gestion de costos donde vea estadisticas en graficos.

User Persona: Dueños de cafeterías.



- Link: https://lucid.app/lucidchart/22f7abe1-aca2-4970-9925-f24fdd1c166f/edit?viewport_loc=-18984%2C-19367%2C24587%2C35020%2CHWEp-vi-RSFO&invitationId=inv_72f97110-024a-4131-a9e3-c460ec619fa9

3.1.4.3. Mobile Applications Mock-ups

Aquí se presentan los mock-ups de la aplicación móvil, en los que se incorporan los elementos visuales definitivos. Estos diseños muestran cómo se verán las pantallas finales, alineadas con la identidad gráfica de la marca.

Registro e Inicio de Sesión
Muestra las pantallas de registro e inicio sesión.

Registro e Inicio de Sesión



Email

Password

Iniciar Sesión

Registrarse

Registro e Inicio de Sesión

Muestra el panel principal con estadísticas importantes.

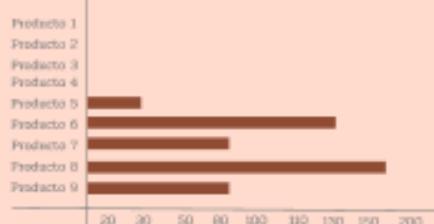


[Regresar](#)



Mi cafetería
Sede 1

Producto más vendido de la semana



Total de Ventas
Noviembre

36 clientes

Octubre

30 clientes

Septiembre

25 clientes

Total de Empleados

15

[Ver más](#)

Total de Proveedores

20

[Ver más](#)



Agregar Empleados

Muestra las pantallas donde se realiza el registro de empleados

Agregar Empleados

The grid displays six screenshots illustrating the employee management process:

- Empleados**: Shows a list of employees with a search bar and a button to "Agregar Empleado".
- Editar Empleado**: Shows a form for editing an employee's details, including fields for Nombre, Rol, Correo, Telefono, and Direccion.
- Editar Empleado**: Shows a confirmation dialog asking "¿QUIERE GUARDAR SUS CAMBIOS?", with "Aceptar" and "Atrás" buttons.
- Ver Empleado**: Shows a detailed view of an employee named "Empleado 1" with fields for Nombre, Rol, and Correo.
- Ver Empleado**: Shows a confirmation dialog asking "¿Quiere eliminar este empleado?", with "Aceptar" and "Cancelar" buttons.
- Editar Empleado**: Shows a form for adding a new employee, identical to the one in the second screenshot.
- Editar Empleado**: Shows a confirmation dialog asking "¿Quiere agregar este empleado?", with "Aceptar" and "Cancelar" buttons.

Agregar Proveedores

Muestra las pantallas donde se realiza el registro de proveedores

Agregar Proveedores

The screenshot displays several screens of the iCafe mobile application related to managing suppliers (Proveedores):

- Editar Proveedor:** A form screen with fields for Nombre, Gmail, and Telefono. It includes a "Guardar Cambios" button at the bottom.
- Editar Proveedor:** A confirmation dialog asking "¿QUIERE GUARDAR SUS CAMBIOS?" with "Aceptar" and "Atras" buttons.
- Proveedores:** A list screen showing a grid of supplier names. A specific row for "PROVEEDOR 1" is highlighted.
- Ver Proveedor:** A details screen for "PROVEEDOR 1" showing fields for Ruc, Gmail, Telefono, and Insumo. It includes a "Eliminar" button.
- Ver Proveedor:** A confirmation dialog asking "¿Quiere eliminar este Proveedor?" with "Aceptar" and "Atras" buttons.
- Editar Proveedor:** A form screen for "PROVEEDOR 1" with fields for Nombre, Gmail, and Telefono. It includes a "Quitar" button.
- Editar Proveedor:** A confirmation dialog asking "¿Quiere agregar este Proveedor?" with "Aceptar" and "Atras" buttons.

Agregar Insumos

Muestra las pantallas donde se realiza el registro de insumos

Agregar Insumos

The screenshot displays the iCafe software interface for managing ingredients (Insumos). It includes:

- Editar Insumos (Add Ingredient):** A form with fields for Proveedor (Supplier), Precio por gramo (Price per gram), Fecha de Compra (Purchase date), and Fecha de vencimiento (Expiry date). Buttons include Guardar Cambios (Save changes) and Regresar (Back).
- Editar Insumos (Edit Ingredient):** A confirmation dialog asking "¿QUERE GUARDAR SUS CAMBIOS?" (Do you want to save your changes?) with Acceptar (Accept) and Atras (Cancel) buttons.
- Insumos (Ingredients):** A list view showing a grid of ingredients with columns for Nombre (Name), FV (Expiration date), and Unidad (Unit). Buttons include Registro Insumo (Register ingredient) and FV (Expiration date).
- Ver Insumos (View Ingredients):** A detailed view of an ingredient with fields for Proveedor, Precio por gramo, Fecha de Compra, and Fecha de vencimiento. Buttons include Eliminar (Delete) and Regresar (Back).
- Ver Insumos (View Ingredients):** A confirmation dialog asking "¿Quiere eliminar este insumo?" (Do you want to delete this ingredient?) with Acceptar (Accept) and Atras (Cancel) buttons.
- Editar Insumos (Add Ingredient):** A form with fields for Nombre, Proveedor, Precio por gramo, Fecha de Compra, and Fecha de vencimiento. Buttons include Agregar (Add) and Regresar (Back).
- Editar Insumos (Edit Ingredient):** A confirmation dialog asking "¿Quiere agregar este insumo?" (Do you want to add this ingredient?) with Acceptar (Accept) and Atras (Cancel) buttons.

Agregar Producto

Muestra las pantallas donde se realiza el registro de productos

Agregar Producto

The grid consists of six wireframe screens arranged in two columns and three rows:

- Row 1:** Two "Editar Producto" (Edit Product) screens. The left one shows a search interface for "Insumos" (Supplies) with a list of items like "Miel", "Leche", etc. The right one shows a confirmation dialog asking "¿QUERE GUARDAR SUS CAMBIOS?" (Do you want to save changes?) with "Aceptar" (Accept) and "Cancelar" (Cancel) buttons.
- Row 2:** Two "Ver Producto" (View Product) screens. The left one shows a list of products with columns for "Nombre" (Name), "Precio" (Price), and "Insumos". The right one shows a confirmation dialog asking "¿Quiere eliminar este Insumo?" (Do you want to delete this supply?) with "Aceptar" (Accept) and "Cancelar" (Cancel) buttons.
- Row 3:** Two "Editar Producto" screens. The left one shows a search interface for "Insumos" with a list of items like "Miel", "Leche", etc. The right one shows a confirmation dialog asking "¿Quiere agregar este Insumo?" (Do you want to add this supply?) with "Aceptar" (Accept) and "Cancelar" (Cancel) buttons.

Registrar Compra

Muestra las pantallas donde se realiza el registro de compra de insumos a proveedores

Registrar Compra

The first screen shows a list of purchases with columns for ID and PRECIO, each with a 'Ver más' button. A 'Registrar Compra' button is at the top. The second screen is a detailed view of a single purchase, showing fields for Identificador, Proveedor, Insumos, Fecha de Compra, and Costo Total, with a 'Guardar' button. A modal asks if you want to add the purchase, with 'Aceptar' and 'Atras' buttons. The third screen shows a list of purchases with fields for Identificador, Proveedor, Insumos, Fecha de Compra, and Costo Total, with a 'Guardar' button.

Registrar Venta

Muestra las pantallas donde se realiza el registro de venta de productos a clientes

The first screen shows a list of sales with columns for ID and PRECIO, each with a 'Ver más' button. A 'Registrar Venta' button is at the top. The second screen is a detailed view of a single sale, showing fields for Identificador, Cliente, Productos, Fecha de Venta, and Costo Total, with a 'Guardar' button. A modal asks if you want to add the sale, with 'Aceptar' and 'Atras' buttons. The third screen shows a list of sales with fields for Identificador, Cliente, Productos, Fecha de Venta, and Costo Total, with a 'Guardar' button.

Inventario y Gestión de Costos

Muestra las pantallas donde se ven los movimientos y analíticas

Inventario y Gestión de Costos

Inventario

Inventario

Gestión de Costos

Inventario

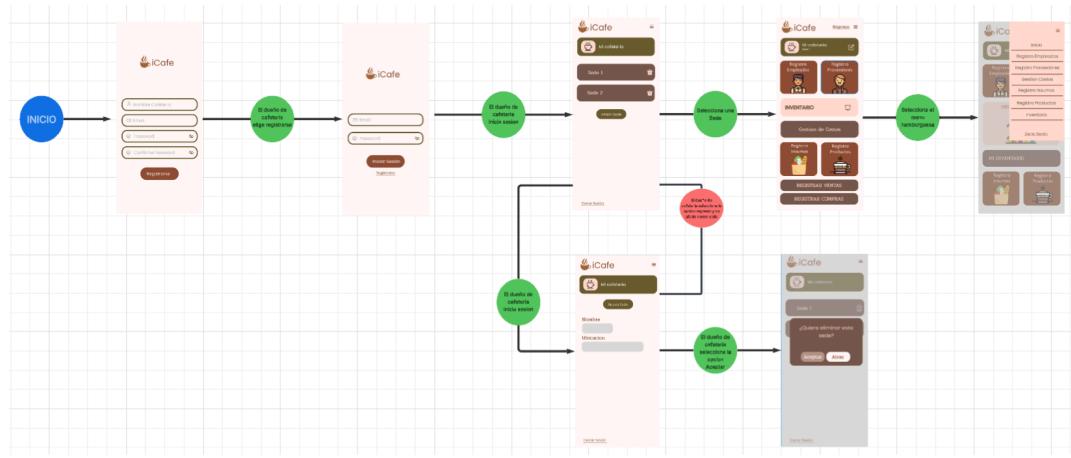
Link: <https://www.figma.com/design/c61JnVMaZZcReGqcVzwikH/Untitled?node-id=0-1&t=gB15ysllfRAw-1>

3.1.4.4. Mobile Applications User Flow Diagrams

Este apartado expone los user flow diagrams, que ilustran los caminos que puede seguir un usuario dentro de la aplicación. Su objetivo es asegurar una experiencia de uso fluida y coherente en cada interacción.

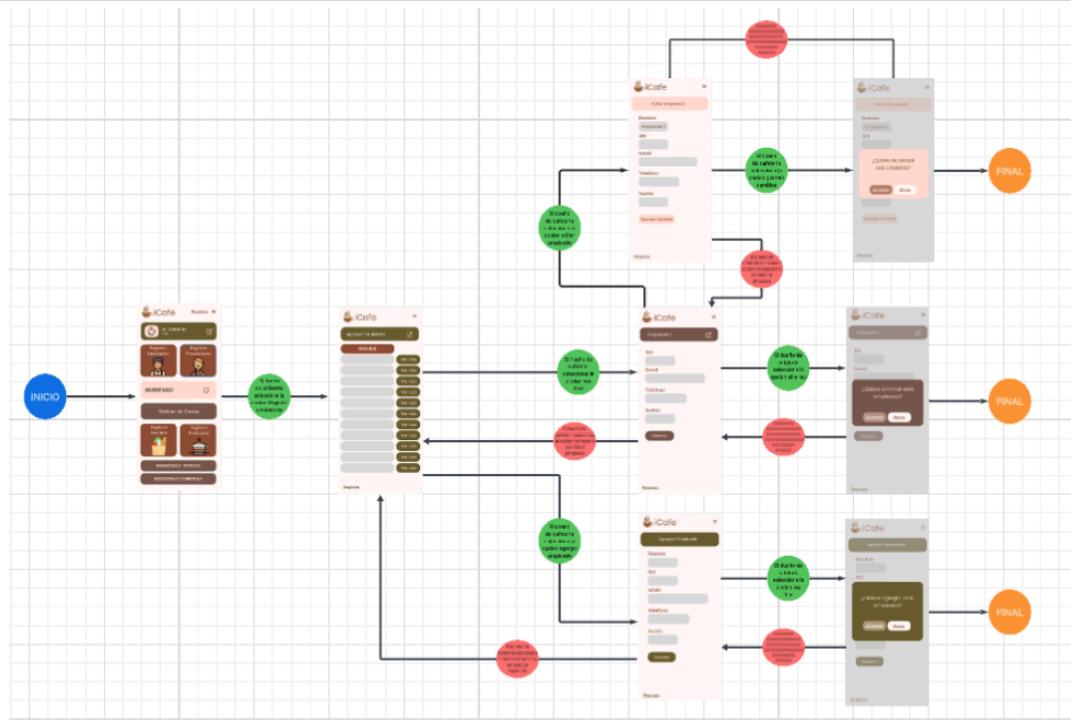
User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo registrarme e iniciar sesion antes de visualizar el panel principal.



User Persona: Dueños de cafeterias.

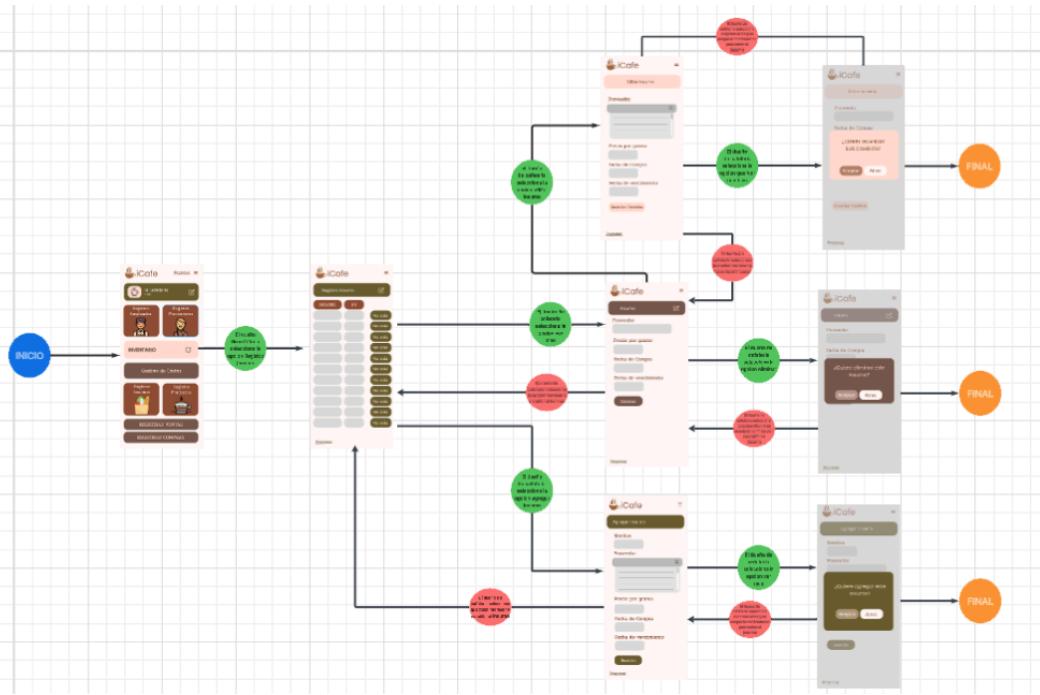
User Goal: Como usuario, deseo agregar, editar y eliminar empleados cuando yo quiera.



User Persona: Dueños de cafeterias.

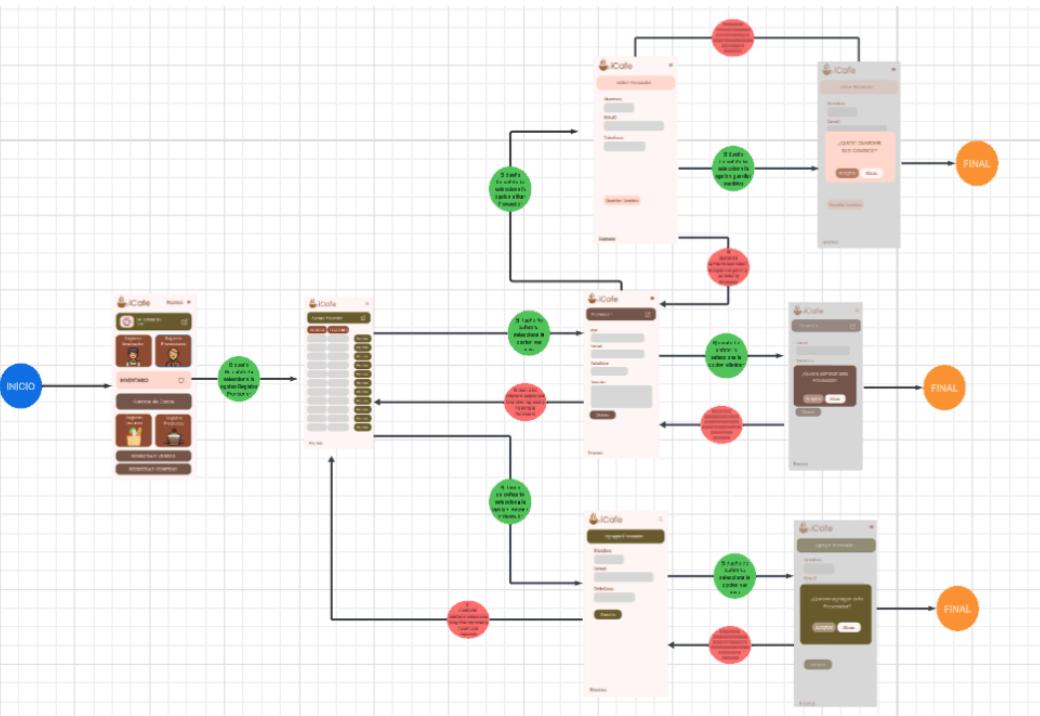
User Goal: Como usuario, deseo agregar, editar y eliminar Insumos cuando yo quiera.

User Persona: Dueños de cafeterias.



User Persona: Dueños de cafeterias.

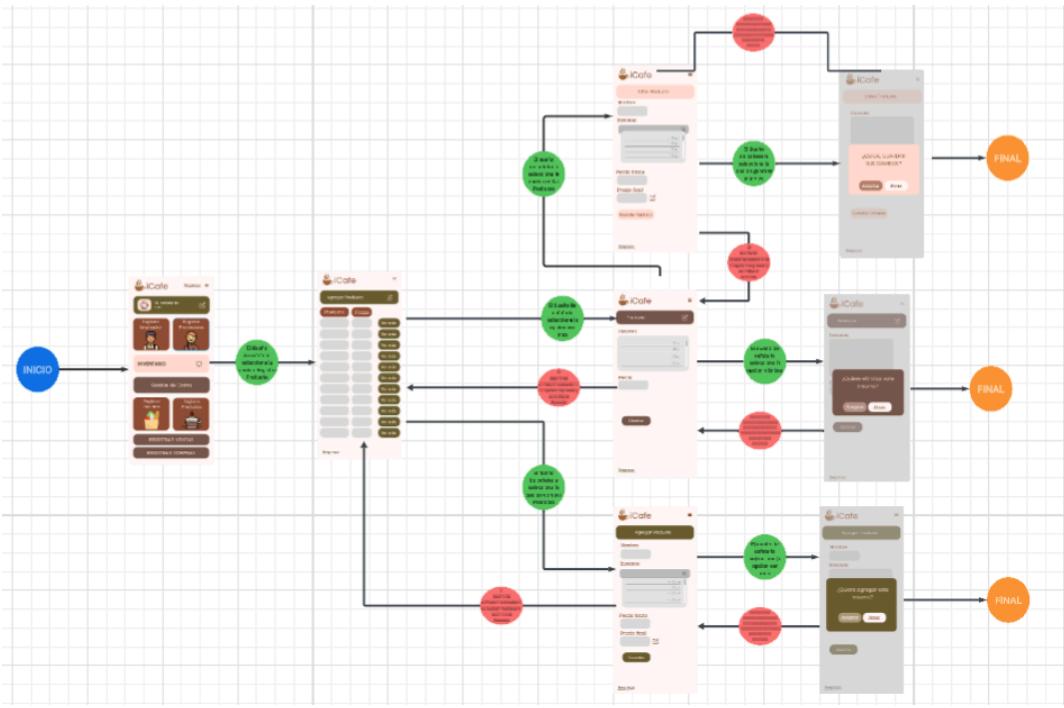
User Goal: Como usuario, deseo agregar, editar y eliminar proveedor cuando yo quiera.



User Persona: Dueños de cafeterias.

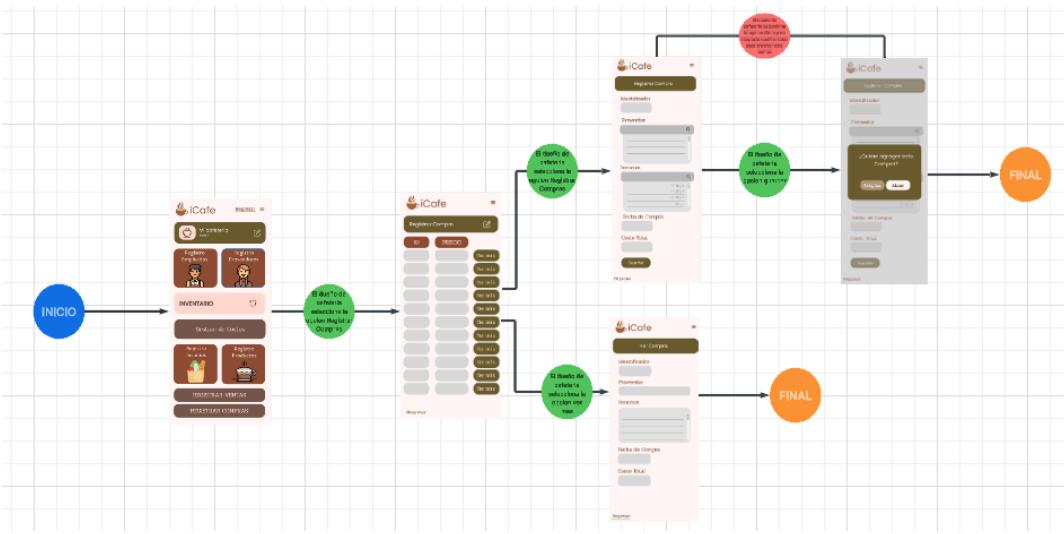
User Goal: Como usuario, deseo agregar, editar y eliminar Producto cuando yo quiera.

User Persona: Dueños de cafeterías.



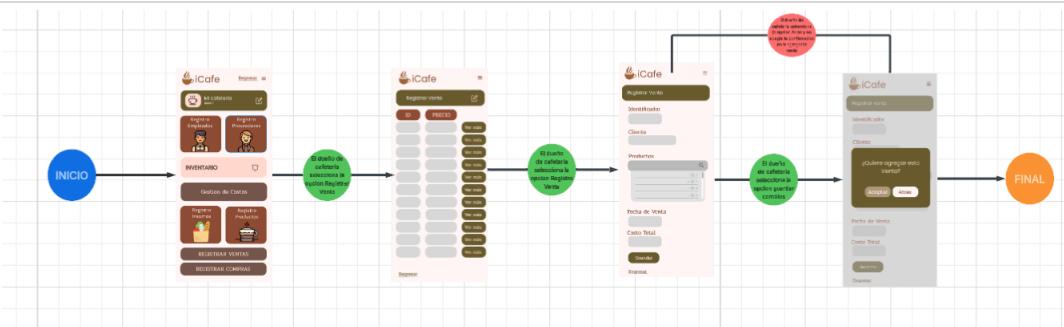
User Persona: Dueños de cafeterías.

User Goal: Como usuario, deseo registrar compras para llevar un recuento de mis inversiones.



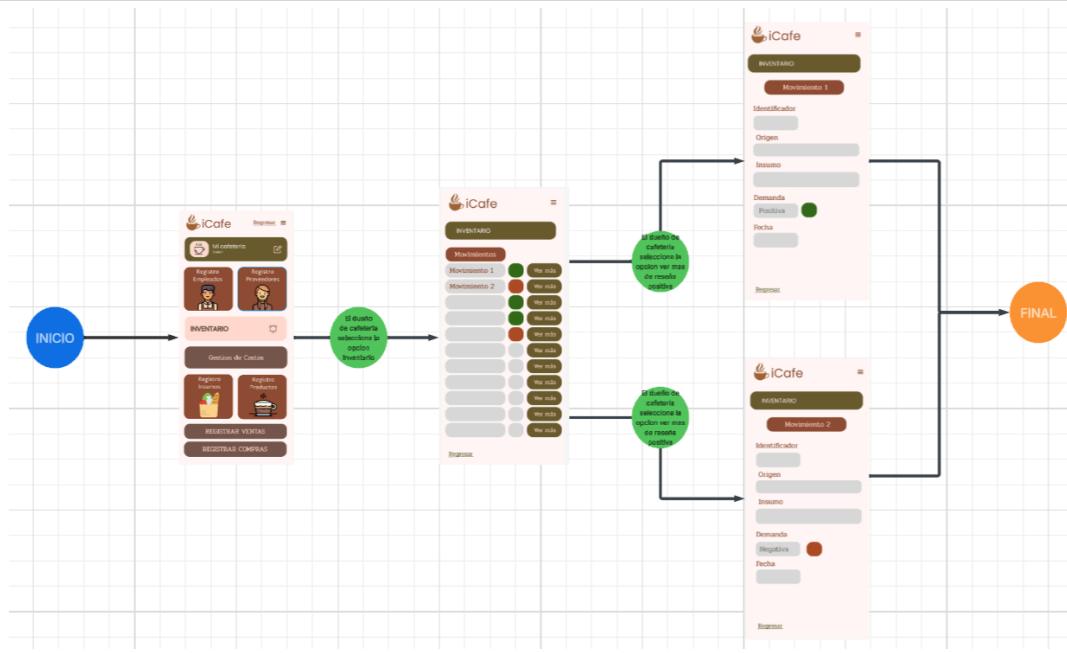
User Persona: Dueños de cafeterías.

User Goal: Como usuario, deseo registrar ventas para llevar un recuento de mis ganancias.



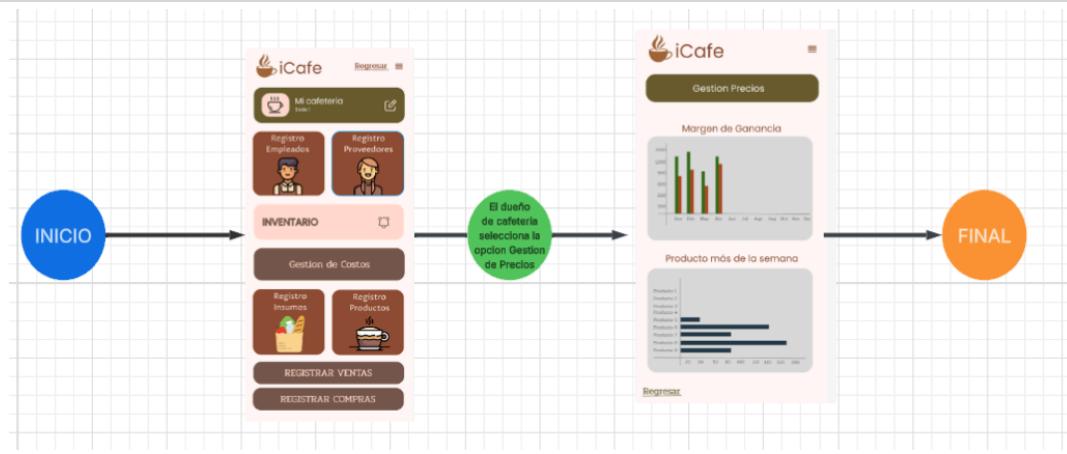
User Persona: Dueños de cafeterias.

User Goal: Como usuario, deseo una sección de inventario donde tenga el recuento de movimientos.



User Persona: Dueños de cafeterias.

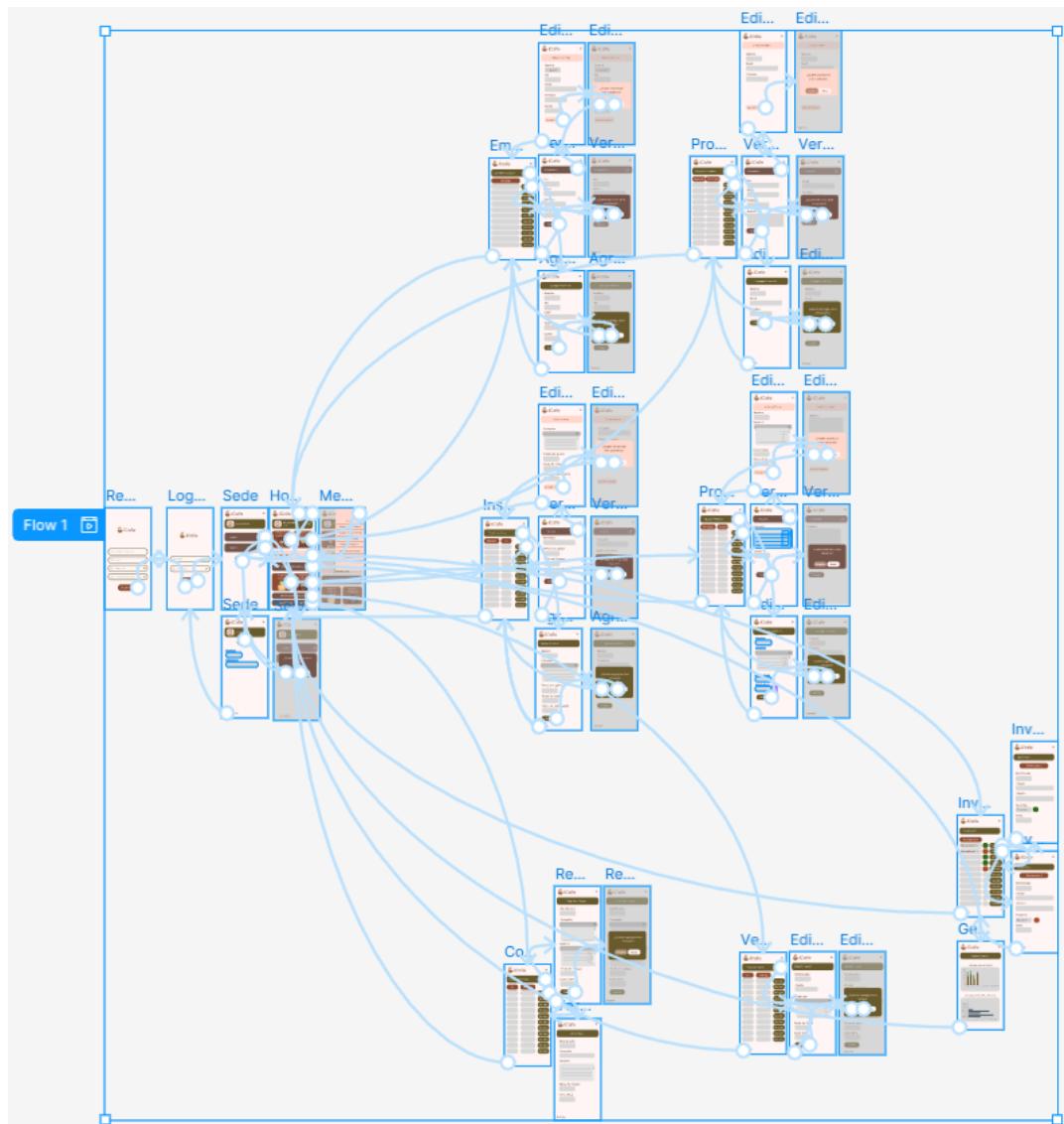
User Goal: Como usuario, deseo una sección de Gestión de costos donde vea estadísticas en graficos.



- Link: https://lucid.app/lucidchart/98d23ad7-1af5-4a5f-b91e-c6a9cfe2c04c/edit?viewport_loc=-7294%2C-7059%2C35610%2C16552%2C0_0&invitationId=inv_cf223d6e-96b8-480b-9205-aeb8095ac5fb

3.1.4.5. Mobile Applications Prototyping

Finalmente, se presentan los prototipos interactivos de la aplicación móvil, los cuales permiten simular la navegación real. Esto facilita la validación temprana de la experiencia de usuario y permite realizar mejoras antes del desarrollo final.



Link: <https://youtu.be/Hd8nV5grxeg>

Link Figma: <https://www.figma.com/design/54NyDHkxRtBDyw7VWojNK/Untitled?node-id=0-1&t=V2Z4IR3eL3gC4gol-1>

3.1.5. Referencias de diseño de aplicaciones similares

Para el diseño de la aplicación **iCafe**, se tomaron como referencia diversas propuestas visuales de aplicaciones similares disponibles en plataformas de diseño como Dribbble y Behance. Estas referencias permitieron identificar tendencias actuales en interfaces móviles, así como buenas prácticas en la organización de dashboards, navegación y experiencia de usuario.

Referencias consultadas:

- [Barbershop App Dashboard – Dribbble](#)
- [Coffee Shop Mobile App UI – Dribbble](#)
- [Restaurant Food Ordering App – Behance](#)

Inspiración aplicada en iCafe:

- **Minimalismo visual:** Se adoptó un estilo limpio y moderno, con predominio de colores neutros y acentos cálidos, similar al dashboard de Barbershop.
- **Jerarquía clara de información:** La disposición de menús y secciones sigue un orden lógico que facilita la navegación, inspirado en apps de pedidos de comida y bebidas.
- **Tipografía legible y consistente:** Se priorizó el uso de fuentes sans-serif para mejorar la lectura en pantallas móviles.
- **Elementos visuales atractivos:** Se incorporaron íconos y microinteracciones que refuerzan la experiencia del usuario, tomando como referencia las animaciones y transiciones vistas en Dribbble y Behance.
- **Enfoque en la experiencia del cliente:** Al igual que en las apps de cafeterías y restaurantes, se buscó que el usuario pueda realizar pedidos y gestionar su cuenta de manera rápida y sencilla.

Este análisis comparativo permitió que el diseño de **iCafe** no solo cumpla con los objetivos funcionales del proyecto, sino que también ofrezca una experiencia visual atractiva y alineada con estándares modernos de aplicaciones móviles.

3.1.6. Wireframe textual de la navegación principal de iCafe

El siguiente wireframe textual describe la estructura y flujo principal de la aplicación **iCafe**, inspirado en las referencias de diseño analizadas. No se incluyen detalles gráficos, sino una representación en palabras de cómo se organiza la navegación:

Pantalla de Inicio (Landing Page)

- Logo de SyncCafe y nombre de la aplicación.
- Botón destacado: *"Ingresar a la aplicación"*.
- Enlaces secundarios: *Acerca de*, *Contacto*, *Descargar App*.

Pantalla de Login / Registro

- Campos: correo electrónico, contraseña.
- Botón: *Iniciar sesión*.
- Enlace: *¿No tienes cuenta? Regístrate*.
- Registro rápido con nombre, correo y contraseña.

Dashboard Principal

- Barra superior con saludo al usuario y acceso a perfil.
- Menú inferior con iconos de navegación:
 - Inicio
 - Productos
 - inventario
 - Ventas
 - Contactos
- Sección central con accesos rápidos:
 - *Nuevo pedido*
 - *Ver reportes*
 - *Promociones activas*

Pantalla de Productos

- Lista de productos con imagen, nombre y precio.
- Botón flotante: *Agregar nuevo producto*.
- Filtros: categoría, disponibilidad.

Pantalla de inventario

- Tabla con stock disponible.
- Alertas visuales para productos con bajo inventario.
- Botón: *Actualizar stock*.

Pantalla de Ventas

- Historial de ventas recientes.
- Gráfico simple de ingresos por día/semana.
- Botón: *Registrar nueva venta*.

Pantalla de Contactos

- Lista de clientes frecuentes.
- Opción de agregar contacto nuevo.
- Integración con datos de pedidos anteriores.

Pantalla de Perfil / Configuración

- Datos del usuario.
- Opciones de configuración: idioma, notificaciones, seguridad.

- Botón: *Cerrar sesión*.

Capítulo IV: Product Implementation & Validation

4.1. Software Configuration Management

La gestión de configuración de software en el proyecto iCafe se ha diseñado para asegurar la trazabilidad, consistencia y calidad en cada etapa del desarrollo. Se han definido entornos, herramientas y convenciones que permiten al equipo colaborar de forma eficiente, mantener control sobre el código fuente y facilitar el despliegue continuo.

4.1.1. Software Development Environment Configuration

Para garantizar un entorno de desarrollo homogéneo entre los integrantes del equipo, se establecieron las siguientes configuraciones:

- **Lenguajes principales:**
 - Backend: Java y Dockerfile.
 - Aplicación móvil: Kotlin.
- **Frameworks y bibliotecas:**
 - Spring Boot para servicios backend.
 - Firebase para autenticación y base de datos en tiempo real.
 - Android Jetpack para arquitectura móvil.
- **IDE recomendados:**
 - IntelliJ IDEA (Java backend).
 - Android Studio (Kotlin móvil).
- **Gestores de paquetes:**
 - Maven para backend Java.
 - Gradle para Android.
- **Control de versiones:** Git con GitHub como repositorio central.
- **Configuración local:** Se documentó en el archivo `README.md` los pasos para instalar dependencias, configurar variables de entorno y ejecutar el proyecto en modo desarrollo.

4.1.2. Source Code Management

El código fuente se gestiona bajo un esquema de ramas que permite trabajo paralelo y control de calidad:

- **Repositorio principal:** [GitHub SyncCafe Organization](#)
- **Estrategia de ramas:**
 - `main` : versión estable para producción.
 - `develop` : integración de nuevas funcionalidades.
 - `feature/*` : desarrollo de módulos específicos (ej. `feature/inventory-tracking`).
 - `hotfix/*` : correcciones urgentes en producción.
- **Pull Requests:** Todo cambio debe pasar por revisión de código (code review) antes de ser fusionado.
- **Commits:** Se siguen convenciones semánticas (`feat:`, `fix:`, `docs:`, `refactor:`) para facilitar la trazabilidad.
- **Integración continua:** Se configuró GitHub Actions para ejecutar pruebas unitarias y validaciones de estilo en cada push.

4.1.3. Source Code Style Guide & Conventions

Para mantener la calidad y legibilidad del código, se definieron guías de estilo por tecnología:

- **Java (Backend):**
 - Convenciones estándar de Java y Spring Boot.
 - Uso de anotaciones `@Service`, `@Repository`, `@RestController`.
 - Separación por paquetes: `controller`, `service`, `model`, `repository`.
- **Kotlin (App móvil):**
 - Convenciones de Android Jetpack.
 - Uso de `ViewModel` y `LiveData` para arquitectura MVVM.

- Separación por capas: ui , data , domain .
- **Documentación:** Cada módulo incluye comentarios Javadoc/KDoc y archivos README.md explicativos.

4.1.4. Software Deployment Configuration

El despliegue de iCafe se ha planificado para facilitar pruebas piloto y escalabilidad futura:

- **Landing Page:**
 - Desplegada en GitHub Pages desde rama gh-pages .
 - Optimizada con etiquetas SEO y diseño responsive.
- **Aplicación móvil:**
 - Compilada en modo debug para pruebas internas.
 - Se utilizará Firebase App Distribution para compartir builds con cafeterías piloto.
- **Backend:**
 - Desplegado en [Render.com](#) con base de datos PostgreSQL.
 - Contenerizado con Docker para facilitar el despliegue.
 - Variables de entorno gestionadas desde el panel de Render.
- **Próximos pasos:**
 - Automatización del despliegue con CI/CD.
 - Configuración de monitoreo con herramientas como Sentry o LogRocket.

4.2. Landing Page & Mobile Application Implementation

4.2.1. Sprint 1

4.2.1.1. Sprint Planning 1

Para el primer sprint el equipo establecio que el desarrollo de las tareas serian aproximadamente 22 horas.

Sprint #	Sprint 1
Sprint Planning Background	
Date	2025-010-06
Time	10:30 PM
Location	Discord
Prepared by	Urrutia Peña Jasmin Adriana
Atendees (to planning meeting)	<ul style="list-style-type: none"> • Velaquez Chambi Ruben Genaro • Christian Fabrizio inga Orihuela • Hallasi Saravia Miguel Angel • Sebastian Alexander Ramos Calagua
Sprint 1 Review Summary	Este es el primer sprint a realizar por el equipo
Sprint 1 Retrospective Summary	El equipo estuvo de acuerdo con empezar con el desarrollo de la landing page, aplicación móvil y el backend
Sprint Goal & User Stories	
Sprint 1 Goal	<ul style="list-style-type: none"> • Nuestro enfoque está en desarrollar una landing page moderna y adaptable, conectada con el módulo de contactos del backend. Creemos que esto brindará una mayor visibilidad y participación de usuarios, atrayendo a más personas interesadas en la aplicación móvil. Esto se confirmará cuando al menos 30 usuarios envíen formularios de contacto • Nuestro enfoque está en construir la arquitectura central del backend siguiendo los principios de Domain-Driven Design (DDD) e implementar los bounded contexts de contacts, product e inventory. Creemos que esto brindará una base escalable y mantenible, que facilitará la extensión de la lógica de negocio y futuras integraciones. Esto se confirmará cuando todos los módulos de dominio estén operativos, pasen las pruebas unitarias y expongan correctamente sus endpoints RESTful documentados en Swagger/OpenAPI.

Sprint #	Sprint 1
	<ul style="list-style-type: none"> Nuestro enfoque está en diseñar e implementar la aplicación móvil utilizando Jetpack Compose, MVVM y Arquitectura Limpia, totalmente integrada con el backend. Creemos que esto brindará una experiencia de usuario fluida e intuitiva. Esto se confirmará cuando la aplicación pueda obtener, mostrar y enviar datos correctamente hacia y desde el backend durante las pruebas.
Sprint 1 Velocity	30
Sum of story points	37

4.2.1.2. Sprint Backlog 1

Sprint #	Sprint 1						
	User Story		Work-item/Task				
ID	Title	ID	Title	Description	Estimation (Hours)	Assigned To	Status (To-do/In-Process/To-Review/Done)
ID	US020 Seccion Home	TA01	Diseñar el Home	Diseñar la seccion home de la lading page en html y css	0.5	Urrutia Peña Jasmin Adriana	Done
ID	US021 Seccion About	TA01	Diseñar el About	Diseñar la seccion About de la lading page en html y css	0.5	Urrutia Peña Jasmin Adriana	Done
ID	US022 Seccion Features	TA01	Diseñar el features	Diseñar la seccion feature de la lading page en html y css	0.5	Urrutia Peña Jasmin Adriana	Done
ID	US024 seccion contact	TA01	Formulario	Diseñar un formulario de contacto en html y css	0.5	Urrutia Peña Jasmin Adriana	Done
ID	US012 Registrar entrada inventario	TA01	Endpoint para el registro de entrada de un insumo	Se debe registrar la entrada de un insumo en el inventario	2	Hallasi Saravia Miguel Angel	Done
		TA02	Diseño en la aplicacion movil de inventario	Se debe de crear la actividad de registro de entrada de un insumo en el inventario	1	Hallasi Saravia Miguel Angel	Done
ID	US013 Actualizar inventario	TA01	Endpoint para actualizar datos del inventario	Se debe crear un endpoint para actualizar datos del inventario	1	Hallasi Saravia Miguel Angel	Done
		TA02	Vista para actualizar inventario	Se debe crear la actividad para actualizar datos del inventario	1	Hallasi Saravia Miguel Angel	In-process
ID	US009 Crear receta	TA01	endpoint para crear una receta	Se debe crear un endpoint para crear una receta	1	Ruben Velasquez Chambi	To-do
		TA02	Vista para crear receta	Se debe crear la actividad para crear una receta	1	Ruben Velasquez Chambi	To-do
ID	US006 Crear proveedor	TA01	enpoint para registrar	Se debe de crear un portafolio de proveedores	2	Ruben Velasquez	Done

			proveedores	que para acada usuario		Chambi	
		TA02	Vista para registro de proveedores	Se debe de crear la actividad para registrar proveedores	1	Ruben Velasquez Chambi	Done
ID	US008 Actualizar proveedor	TA01	enpoint para actualizar proveedores	Se debe de poder modificar los datos de los proveedores	1	Ruben Velasquez Chambi	Done
		TA02	Vista para actualizar de proveedores	Se debe de crear la actividad para actualizar proveedores	1	Ruben Velasquez Chambi	Done
ID	US004 Actualizar empleados	TA01	enpoint para actualizar empleados	Se debe de poder modificar los datos de los empleados	1	Ruben Velasquez Chambi	Done
		TA02	Vista para actualizar de empleados	Se debe de crear la actividad para actualizar empleados	1	Ruben Velasquez Chambi	Done
ID	US005 Eliminar empleados	TA01	enpoint para actualizar empleados	Se debe de poder eliminar los empleados	0.5	Ruben Velasquez Chambi	Done
ID	US003 Listar empleados	TA01	enpoint para listar empleados	Se debe poder consultar la lista de los empleados	1	Ruben Velasquez Chambi	Done
		TA02	Vista de lista de empleados	Se debe de crear la actividad para visualizar la lista de empleados	1	Ruben Velasquez Chambi	Done
		TA03	Implementar el backend en la aplicacion movil	Se debe de consumir el endpoint correspondiente	1	Ruben Velasquez Chambi	To-do
ID	US007 Listar proveedores	TA01	enpoint para listar proveedores	Se debe poder consultar la lista de los proveedores	1	Ruben Velasquez Chambi	Done
		TA02	Vista de lista de proveedores	Se debe de crear la actividad para visualizar la lista de proveedores	1	Ruben Velasquez Chambi	Done
		TA03	Implementar el backend en la aplicacion movil	Se debe de consumir el endpoint correspondiente	1	Ruben Velasquez Chambi	To-do

4.2.1.3. Development Evidence for Sprint Review

Repository	Branch	Commit ID		Commit message
https://github.com/1ACC0238-2520-1807/LandingPage-iCafe	main	303e4706fb23d21bfb860ae99bdd672c01ea919c		chore: initial commit
	main	0ae643b60a5e5096960a618b8310a35448a95147		feat: Add files via upload
	main	6e46bb204654992384f5364f437f8cae16f940ee		feat: Create CNAME
	main	95e03a811f89e7d8de20692d5b78ffc441ade31c		feat: Update index.html
	main	0faf3074244b70240578d560275c40d47f1ba254		Update script.js

https://github.com/1ACC0238-2520-1807/backend-icafe	main	737c8c177cffc724783a3c4ab9199ea14f098022	feat(jam): bound context terminado y base de proyecto
	main	fe33dccc92221d7354f8b1aebcf3e2c04811667f	feat(contacts): se agrego la capa domain y infrastructure
	develop	a48c5c6998e8f2b19fbb42e435c232d4c64ca133	feat(portfolio): cuando se crea una nueva cuenta, se crea un nuevo portfolio que tiene como referencia el userID
	feature/contacts	a48c5c6998e8f2b19fbb42e435c232d4c64ca133	feat(contacts): se implemento update y delete para provider
	feature/contacts	a48c5c6998e8f2b19fbb42e435c232d4c64ca133	feat(contacts): se implemento todo contacts, pero falta limpieza de codigo
	feature/products	a48c5c6998e8f2b19fbb42e435c232d4c64ca133	feat: product
	develop/td>	625041b48d14e4441d960ccb8d4eee1e5f66bf0d	feat(docker): add docker configuration
	develop/td>	d34cace9052543fe8285432699bd6ef35d85ba03	Merge pull request #2 from 1ACC0238-2520-1807/develop
https://github.com/1ACC0238-2520-1807/Android-icafe	main	2a5a7cd5098ed3171233e769cb5e34de03bba9bd	first commit
	develop	238ad865911d8882dd6cd79d53813f81f622a9b0	feat(auth): se creo la presentacion de login y register
	develop	2d2e039a27f854fbe67fd3ac32f49bed6e2bb27b0	feat(AppNav): Se creo la navegacion base
	feature/contacts	1e048bc5ef06f70d52b01ccb52b5d592c89d2c46	Delete app/src/main/java/com/example/icafe/core directory
	feature/contacts	47daa00738f52ef28a9a5e865c4ca38d2648137d	Delete app/src/main/java/com/example/icafe/ui/theme directory
	feature/contacts	86804944b385927c8e16dd0db8be1a904507789c	Update AndroidManifest.xml
	feature/contacts	bf5818fbce0e9f3d9dc82017ef4940bb9f86f4c4	Update libs.versions.toml
	develop	ffbc7c8e409c137f5d4368535c9a7b148292f815	Merge pull request #1 from 1ACC0238-2520-1807/contacts

4.2.1.4. Testing Suite Evidence for Sprint Review

Objetivo

Demostrar que las funcionalidades desarrolladas durante el sprint han sido validadas mediante pruebas automatizadas y/o manuales, garantizando su correcto funcionamiento y cumplimiento de criterios de aceptación.

Evidencia de pruebas realizadas

Tipo de prueba	Herramienta utilizada	Cobertura	Resultado	Historias cubiertas
Unit Testing	JUnit	85%	Passed	US012, US014, US019
API Testing	Swagger	100%	Passed	US006, US012, US017, US018
Manual Testing	Checklist funcional	—	Passed	US001-US024

Evidencia técnica adjunta

- Capturas de pantalla de pruebas exitosas.
- Logs de ejecución de test suite.
- Video corto de pruebas UI automatizadas.
- Enlace al repositorio con carpeta `/tests` documentada: [GitHub SyncCafe](#)

Alineación con criterios de aceptación

Las pruebas fueron diseñadas para validar los criterios de aceptación definidos en las historias de usuario, siguiendo el formato **Dado–Cuando–Entonces**.

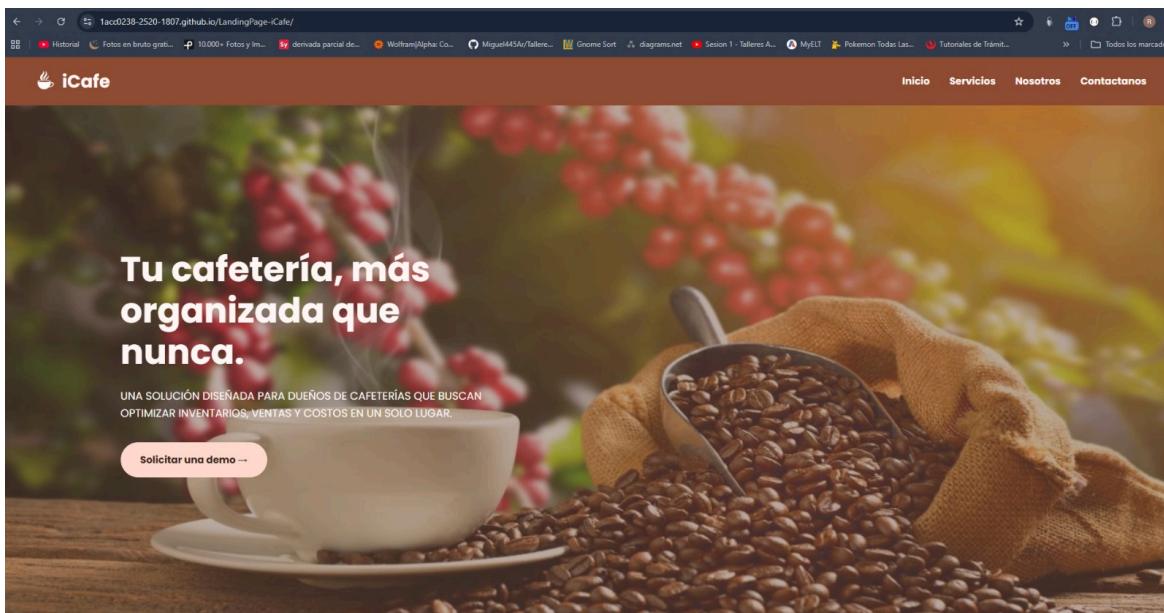
Conclusión

La suite de pruebas ejecutada durante el sprint valida las funcionalidades entregadas, cubre los criterios de aceptación definidos y permite avanzar con confianza hacia el despliegue. Se recomienda mejorar la cobertura de pruebas UI en el próximo sprint.

4.2.1.5. Execution Evidence for Sprint Review

En esta sección describiré las actividades desarrolladas durante el Sprint 1, las cuales abarcan el trabajo realizado en los tres componentes principales del proyecto: la Landing Page, el Backend y la Aplicación Móvil. Se presentarán los avances obtenidos en cada uno de estos módulos, acompañados de capturas de pantalla que evidencian las vistas implementadas y su funcionamiento. Además, se incluirá un enlace a un video demostrativo donde se podrá observar la visualización y navegación logradas durante este Sprint.

Landing Page



SERVICIOS
Administra tu café de forma fácil, rápida y digital.

PROVEEDORES

ORGANIZACIÓN DE PROVEEDORES
CONTACTAR DE FORMA FÁCIL
ENCONTRARLOS POR CATEGORÍA

INVENTARIO

CONTROL AUTOMÁTICO DE INSUMOS
ALERTAS DE STOCK BAJO
HISTÓRICO DE ENTRADAS Y SALIDAS

GESTIÓN COSTOS

REGISTRO DE VENTAS DIARIAS
REPORTES CLAROS DE INGRESOS Y GASTOS
HERRAMIENTAS PARA CÁLCULO

SOBRE NOSOTROS

Nuestra Misión

Nuestra Misión

Brindar a los dueños de cafeterías una herramienta digital sencilla y eficiente que les permita optimizar la administración de su negocio, reducir pérdidas y mejorar la toma de decisiones, facilitando su crecimiento y sostenibilidad.

Nuestra Visión

Ser la plataforma líder en soluciones tecnológicas para cafeterías en Latinoamérica, impulsando la digitalización de pequeños y medianos negocios de café y contribuyendo a la creación de experiencias más organizadas, rentables y sostenibles.

iCafe

¡ESTAMOS A UN MENSAJE DE DISTANCIA!

Nombre completo

Correo electrónico

Teléfono

Escribe tu mensaje aquí...

ENVIAR MENSAJE

La Landing Page fue diseñada para presentar la aplicación móvil a los dueños de cafeterías, destacando sus beneficios principales y motivando la descarga mediante un botón de "Call to Action". Se priorizó un diseño limpio, moderno y adaptable a distintos dispositivos.

Link: <https://1acc0238-2520-1807.github.io/LandingPage-iCafe/>

Backend:

Desarrollada con **Kotlin** y **Jetpack Compose**, la aplicación móvil permite a los dueños de cafeterías gestionar empleados, proveedores e inventario desde su dispositivo. Se implementó arquitectura **MVVM** y principios de **Clean Architecture** para asegurar mantenibilidad y escalabilidad.

The screenshot shows the Swagger UI for the iCafe API. The top navigation bar includes links for 'PLAY', 'API', 'DOC', and 'Explore'. The main content area is titled 'iCafe API' and displays the following sections:

- Authentication**: Contains endpoints for sign-up and sign-in.
- Contact-Portfolio**: Contains endpoints for managing contact portfolios and their providers.
- Inventory**: Contains endpoints for managing inventory items.
- Roles**: Contains an endpoint for managing roles.
- Stock Movements**: Contains endpoints for managing stock movements.
- Products**: Contains endpoints for managing products, including creating, updating, deleting, and archiving them.
- Users**: Contains endpoints for managing user accounts.
- Provider-Contact**: Contains endpoints for managing provider contacts.

Each endpoint is listed with its method (e.g., GET, POST, PUT, DELETE), URL, and a brief description. The background features a light blue gradient with a subtle grid pattern.

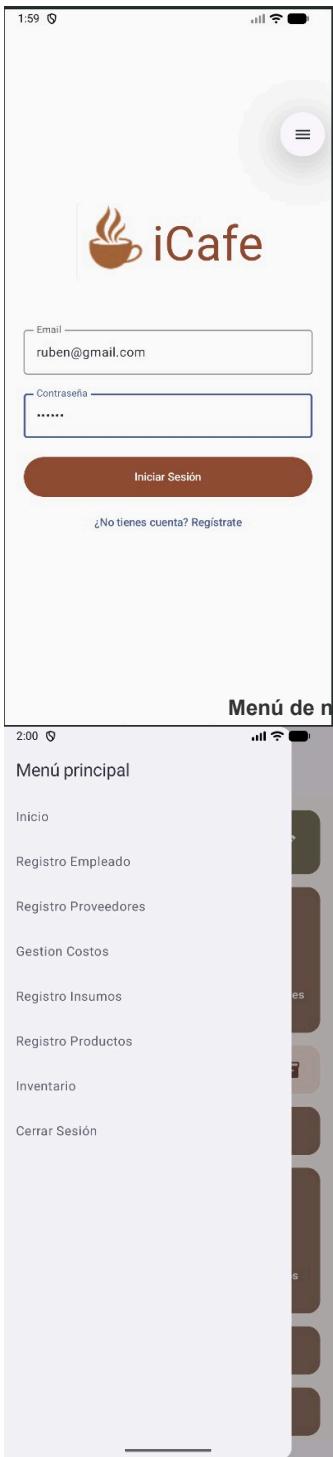
Link: https://drive.google.com/file/d/11aM-DKEcy-WijTvHAGVb-tzOcH0vTmTB/view?usp=drive_link

Aplicacion Moviele:

Desarrollada con **Kotlin** y **Jetpack Compose**, la aplicación móvil permite a los dueños de cafeterías gestionar empleados, proveedores e inventario desde su dispositivo. Se implementó arquitectura **MVVM** y principios de **Clean Architecture** para asegurar mantenibilidad y escalabilidad.

Pantalla de inicio de sesión

Pantalla principal



Registrar nuevo empleado

The dashboard has a header 'Mi Cafetería Portfolio #4'. It includes sections for 'Registro Empleados' (Employee Registration), 'Registro Proveedores' (Supplier Registration), 'INVENTARIO' (Inventory), 'Gestion de Costos' (Cost Management), 'Registro Insumos' (Registration of Raw Materials), 'Registro Productos' (Registration of Products), 'REGISTRAR VENTAS' (Register Sales), and 'REGISTRAR COMPRAS' (Register Purchases). On the right, there's a 'Lista de empleados' (List of Employees) section with a sub-header 'Empleados' and a button 'Agregar Empleado' (Add Employee). Below it is a table with two rows: 'jhon' and 'federico', each with a 'Ver más' (View more) button.

Lista de empleados

NOMBRE	
jhon	Ver más
federico	Ver más

Editar información de empleado

2:01 ⓘ

← Agregar Empleado ⌂

Agregar Empleado

Nombre

Rol

Gmail

Lista de proveedores

2:01 ⓘ

← Proveedores ⌂

Agregar Proveedor +

NOMBRE TELÉFONO

empaniditas 987564123 Ver más

2:01 ⓘ

← Ver más Empleado ⌂

jhon

Rol

barrrendero

Gmail

jhon@gmail.com

Teléfono

Registrar nuevo proveedor

2:01 ⓘ

← Agregar Proveedor ⌂

Agregar Proveedor

Nombre

RUC

Gmail

Teléfono

Guardar

2:02 ⓘ

← Ver Proveedor ⌂

empaniditas S.A

RUC:

78564564

Gmail:

empa@gmail.com

Teléfono:

987564123

Eliminar

2:02 ⓘ

← Insumos ⌂

Registrar nuevos insumos al inventario

Nombre	Cantidad
azucar	10.0 kilogramos
Leche	10.0 litros
cafe	200.0 gramos

+

Video demostrativo: [Ver en Google Drive](#)

Link: <https://drive.google.com/file/d/1FjrpWygjRiYmw7yVLpyuLXEac2Lg016r/view?usp=sharing>

4.2.1.6. Services Documentation Evidence for Sprint Review

En esta sección se incluye la relación de Endpoints documentados con OpenAPI, relacionados con el alcance del Sprint. Durante este Sprint se logró implementar y documentar completamente la API RESTful del backend de iCafe, utilizando Spring Boot y Swagger/OpenAPI para generar documentación interactiva y detallada de todos los endpoints disponibles.

Logros Alcanzados en Documentación de Web Services:

- Implementación completa de documentación OpenAPI 3.0 para todos los bounded contexts desarrollados
- Configuración de Swagger UI para interacción en tiempo real con la API
- Documentación detallada de esquemas de request/response para cada endpoint
- Especificación de códigos de estado HTTP y manejo de errores
- Ejemplos de uso y datos de muestra para facilitar la integración

Tabla de Endpoints Documentados:

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/providers	GET	GET /api/v1/contact-portfolios/1/providers
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/providers	POST	POST /api/v1/contact-portfolios/1/providers
Inventory	/api/v1/inventory/movements	POST	POST /api/v1/inventory/movements
Inventory	/api/v1/inventory/movements/{branchId}	GET	GET /api/v1/inventory/movements/1
Inventory	/api/v1/inventory/stock/{branchId}/{supplyItemId}	GET	GET /api/v1/inventory/stock/1/10
Products	/api/v1/products	GET	GET /api/v1/products
Products	/api/v1/products	POST	POST /api/v1/products
Products	/api/v1/products/{productId}	GET	GET /api/v1/products/1
Products	/api/v1/products/{productId}	PUT	PUT /api/v1/products/1
Products	/api/v1/products/{productId}/activate	POST	POST /api/v1/products/1/activate
Products	/api/v1/products/{productId}/archive	POST	POST /api/v1/products/1/archive
Products	/api/v1/products/{productId}/ingredients	POST	POST /api/v1/products/1/ingredients
Products	/api/v1/products/{productId}/ingredients/{supplyItemId}	DELETE	DELETE /api/v1/products/1/ingredients/10
Products	/api/v1/products/branch/{branchId}	GET	GET /api/v1/products/branch/1

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada
Purchase Orders	/api/v1/purchase-orders	POST	POST /api/v1/purchase-orders
Purchase Orders	/api/v1/purchase-orders/branch/{branchId}	GET	GET /api/v1/purchase-orders/branch/1
Sales	/api/v1/sales	GET	GET /api/v1/sales
Sales	/api/v1/sales	POST	POST /api/v1/sales
Sales	/api/v1/sales/{saleId}	GET	GET /api/v1/sales/1
Sales	/api/v1/sales/{saleId}/cancel	PUT	PUT /api/v1/sales/1/cancel
Sales	/api/v1/sales/{saleId}/complete	PUT	PUT /api/v1/sales/1/complete
Sales	/api/v1/sales/branch/{branchId}	GET	GET /api/v1/sales/branch/1
SupplyItem	/api/v1/supply-items	GET	GET /api/v1/supply-items
SupplyItem	/api/v1/supply-items	POST	POST /api/v1/supply-items
SupplyItem	/api/v1/supply-items/{branchId}/branch	GET	GET /api/v1/supply-items/1/branch
SupplyItem	/api/v1/supply-items/{id}	DELETE	DELETE /api/v1/supply-items/1
SupplyItem	/api/v1/supply-items/{id}	GET	GET /api/v1/supply-items/1
SupplyItem	/api/v1/supply-items/{id}	PUT	PUT /api/v1/supply-items/1
SupplyItem	/api/v1/supply-items/{id}	GET	GET /api/v1/supply-items/1
SupplyItem	/api/v1/supply-items/{id}	PUT	PUT /api/v1/supply-items/1
Sales	/api/v1/sales	POST	POST /api/v1/sales
Sales	/api/v1/sales/{id}	GET	GET /api/v1/sales/1
Inventory	/api/v1/inventories/{id}	GET	GET /api/v1/inventories/1
Inventory	/api/v1/inventories/{id}	PUT	PUT /api/v1/inventories/1
Products	/api/v1/products/{id}	GET	GET /api/v1/products/1
Products	/api/v1/products/{id}	PUT	PUT /api/v1/products/1
Purchase Orders	/api/v1/purchase-orders	POST	POST /api/v1/purchase-orders
SupplyItem	/api/v1/supply-items/{id}	GET	GET /api/v1/supply-items/1
SupplyItem	/api/v1/supply-items/{id}	PUT	PUT /api/v1/supply-items/1

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/providers/{providerId}	GET	GET /api/v1/contact-portfolios/1/providers/1
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/providers/{providerId}	PUT	PUT /api/v1/contact-portfolios/1/providers/1
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/providers/{providerId}	DELETE	DELETE /api/v1/contact-portfolios/1/providers/1
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/employees	GET	GET /api/v1/contact-portfolios/1/employees
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/employees	POST	POST /api/v1/contact-portfolios/1/employees
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/employees/{employeeId}	GET	GET /api/v1/contact-portfolios/1/employees/1
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/employees/{employeeId}	PUT	PUT /api/v1/contact-portfolios/1/employees/1
Contact-Portfolio	/api/v1/contact-portfolios/{portfolioId}/employees/{employeeId}	DELETE	DELETE /api/v1/contact-portfolios/1/employees/1
Provider-Contact	/api/v1/providercontacts	GET	GET /api/v1/providercontacts
Provider-Contact	/api/v1/providercontacts	POST	POST /api/v1/providercontacts
Products	/api/v1/products	GET	GET /api/v1/products?ownerId=1&branchId=1
Products	/api/v1/products	POST	POST /api/v1/products
Products	/api/v1/products/{productId}	GET	GET /api/v1/products/1
Products	/api/v1/products/{productId}	PUT	PUT /api/v1/products/1
Products	/api/v1/products/{productId}	DELETE	DELETE /api/v1/products/1
Products	/api/v1/products/{productId}/archive	PATCH	PATCH /api/v1/products/1/archive
Products	/api/v1/products/{productId}/activate	PATCH	PATCH /api/v1/products/1/activate

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada
Inventory	/api/v1/inventory/items	GET	GET /api/v1/inventory/items
Inventory	/api/v1/inventory/items	POST	POST /api/v1/inventory/items
Inventory	/api/v1/inventory/items/{id}	GET	GET /api/v1/inventory/items/1
Inventory	/api/v1/inventory/items/{id}	PUT	PUT /api/v1/inventory/items/1
Inventory	/api/v1/inventory/items/{id}	DELETE	DELETE /api/v1/inventory/items/1
Stock Movements	/api/v1/inventory/movements	GET	GET /api/v1/inventory/movements
Stock Movements	/api/v1/inventory/movements	POST	POST /api/v1/inventory/movements
Stock Movements	/api/v1/inventory/movements/{id}	GET	GET /api/v1/inventory/movements/1
Authentication	/api/v1/authentication/sign-up	POST	POST /api/v1/authentication/sign-up
Authentication	/api/v1/authentication/sign-in	POST	POST /api/v1/authentication/sign-in
Users	/api/v1/users	GET	GET /api/v1/users
Users	/api/v1/users/{userId}	GET	GET /api/v1/users/1
Roles	/ap/v1/roles	GET	GET /ap/v1/roles

Documentación Interactiva Desplegada:

La documentación completa de la API está disponible en: <http://upc-icafbackend-3sger0-aa823d-31-97-13-234.traefik.me/swagger-ui/index.html>

Capturas de Interacción con la Documentación:

The screenshot shows the iCafe API documentation using the Swagger UI. The left sidebar lists various bounded contexts:

- Authentication**: Authentication Objects
 - POST /api/v1/authentication/login** (Login)
 - POST /api/v1/authentication/logout** (Logout)
- Contact-Portfolio**: Contact Profile Management Endpoints
 - GET /api/v1/contact-profile/list/{providerId}** (Get contact-profile list by providerId)
 - PUT /api/v1/contact-profile/list/{providerId}** (Update contact-profile list by providerId)
 - DELETE /api/v1/contact-profile/list/{providerId}** (Delete contact-profile list by providerId)
 - GET /api/v1/contact-profile/list/{providerId}/empLOYEE/{empLOYEE}** (Get employee by providerId and empLOYEE)
 - PUT /api/v1/contact-profile/list/{providerId}/empLOYEE/{empLOYEE}** (Update employee by providerId and empLOYEE)
 - DELETE /api/v1/contact-profile/list/{providerId}/empLOYEE/{empLOYEE}** (Delete employee by providerId and empLOYEE)
 - GET /api/v1/contact-profile/list/{providerId}/president** (Get president by providerId)
 - PUT /api/v1/contact-profile/list/{providerId}/president** (Update president by providerId)
 - GET /api/v1/contact-profile/list/{providerId}/empLOYEE** (Get empLOYEE by providerId)
 - PUT /api/v1/contact-profile/list/{providerId}/empLOYEE** (Update empLOYEE by providerId)
- Inventory**: Inventory Management Objects
 - GET /api/v1/inventory/item/{ID}**
 - PUT /api/v1/inventory/item/{ID}**
 - DELETE /api/v1/inventory/item/{ID}**
 - GET /api/v1/inventory/item**
 - POST /api/v1/inventory/item**
- Roles**: Risk Management Endpoints
 - GET /api/v1/roles**
- Stock Movements**: Stock Movement Management Endpoints
 - GET /api/v1/inventory/documents**
 - POST /api/v1/inventory/documents**
 - GET /api/v1/inventory/documents/{ID}**
- Products**: Product management operations
 - GET /api/v1/products/{productID}** (Get product by ID)
 - PUT /api/v1/products/{productID}** (Update product)
 - DELETE /api/v1/products/{productID}** (Delete product)
 - GET /api/v1/products** (Get all products)
 - POST /api/v1/products** (Create a new product)
 - PUT /api/v1/products/{productID}/archive** (Archive product)
 - DELETE /api/v1/products/{productID}/archive** (Unarchive product)
- Users**: User Management Endpoints
 - GET /api/v1/users**
 - GET /api/v1/users/{userId}**
- Provider-Contact**: Provider Contact Management Endpoints
 - GET /api/v1/provider-contacts**
 - POST /api/v1/provider-contacts**

Figura 1: Vista general de la documentación Swagger UI mostrando todos los bounded contexts disponibles

This screenshot shows a detailed view of the **Contacts** endpoint under the **Provider-Contact** bounded context.

Operations:

- GET /api/v1/provider-contacts/{providerId}** (Get provider contacts)
- PUT /api/v1/provider-contacts/{providerId}** (Update provider contacts)
- DELETE /api/v1/provider-contacts/{providerId}** (Delete provider contacts)
- GET /api/v1/provider-contacts** (Get all provider contacts)

Parameters:

Name	Description
ownerId	Owner ID
branchId	Branch ID

Responses:

Example response (JSON format):

```

{
  "id": 4,
  "name": "John Doe",
  "branch": 1,
  "owner": 1
}
  
```

Figura 2: Documentación detallada de endpoints del bounded context Contacts con ejemplos de request/response

- **URL del Repositorio:** <https://github.com/1ACC0238-2520-1807/iCafe-Backend>

4.2.1.7. Software Deployment Evidence for Sprint Review

Configuración del Despliegue del Backend (Web Services)

Para el despliegue del backend de iCafe, se implementó una estrategia de contenerización utilizando Docker Compose, lo que permite una gestión eficiente y escalable de los servicios. El proceso de despliegue se realizó mediante las siguientes tecnologías y configuraciones:

Tecnologías Utilizadas:

- **Docker Compose:** Para la orquestación de contenedores y gestión de servicios
- **Dokploy:** Como plataforma de despliegue y gestión de aplicaciones contenerizadas
- **OpenJDK 17 (openjdk:17-jdk-slim):** Imagen base optimizada para ejecutar aplicaciones Java
- **Spring Boot:** Framework principal para el desarrollo del backend

Proceso de Despliegue:

- Contenerización:** El backend fue empaquetado en contenedores Docker utilizando la imagen `openjdk:17-jdk-slim`, que proporciona un entorno ligero y optimizado para aplicaciones Java con Spring Boot.
- Configuración con Docker Compose:** Se configuró un archivo `docker-compose.yml` que define el servicio de la aplicación backend.
- Despliegue con Dokploy:** Se utilizó Dokploy como plataforma de despliegue, que actúa como una solución de Platform-as-a-Service (PaaS) simplificando la gestión de contenedores y proporcionando funcionalidades de monitoreo, escalado automático y gestión de recursos.

Evidencia Visual del Despliegue:

The screenshot shows the Dokploy interface with the following sections:

- General:** Shows the deployment status as "Running".
- Deploy Settings:** Includes options for "Display", "Record", "Stop", "Open Terminal", and "Autodeploy".
- Provider:** Set to GitHub, with the URL `dokployapp23236363133`.
- Repository:** Set to `backend-scale`.
- Branch:** Set to `main`.
- Compose Path:** Set to `/Docker-compose.yml`.
- Trigger Type:** Set to "On Push".
- Watch Paths:** Set to `src/main/java`.
- Converted Compose:** Displays the generated Docker Compose file with added domains. It includes environment variables like `SPRING_PROFILES_ACTIVE=prod`, `SPRING_DATASOURCE_URL=jdbc:mysql://rs-frosty-sealess-addy:3306/...?autoReconnect=true&useSSL=false`, and `SPRING_DATASOURCE_PASSWORD=zef15m6A5tD`. It also specifies ports and services for the backend and database.
- Domains:** Shows a list of domains: `iCafe-backend-Sgep0-aab23d-31-07-13-214.traefik.me`. It includes fields for "Path" (set to "/"), "Port" (set to 8080), and "Protocol" (set to "HTTP").
- Deployment:** Shows deployment logs with 160 lines of output. The logs include build steps for Spring Boot (e.g., "INFO: #44 11.95 [INFO] ... resources/3.2/testResources (default-testResources) # [iCafe ...]", "INFO: #44 11.96 [INFO] skip non existing resourceDirectory /app/src/test/resources"), compilation ("INFO: #44 11.94 [INFO] ... compiler:3.4.3-b2023-09-03-150950!jar"), and packaging ("INFO: #44 11.95 [INFO] Compiling 1 source file with Java (debug parameters released-17) to target/test-classes"). It also shows the deployment of the application ("INFO: #44 12.41 [INFO] ... iCafe-0.0.1-SNAPSHOT.jar was deployed to /app/target/iCafe-0.0.1-SNAPSHOT.jar"), tests being skipped ("INFO: #44 12.55 [INFO] Tests are skipped"), and the final success message ("INFO: #44 16.26 [INFO] total time: 14.439 s").

Configuración del Despliegue de la Landing Page

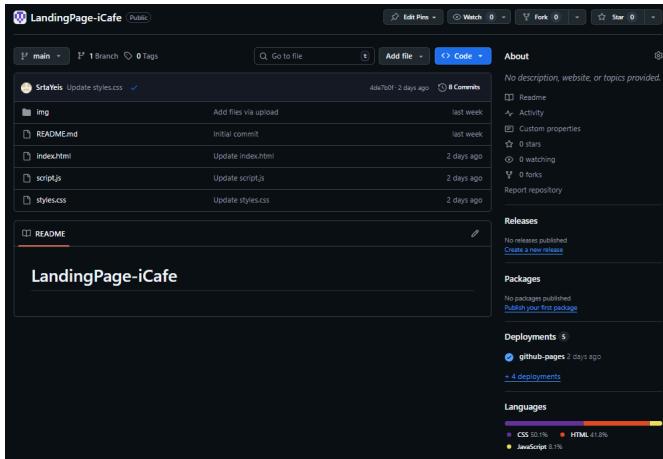
Para el despliegue de la Landing Page de iCafe, se implementó una solución web estática utilizando tecnologías fundamentales del desarrollo web y una plataforma de hosting gratuita.

Tecnologías Utilizadas:

- **HTML**: Para la estructura y contenido de la página web
- **CSS**: Para el diseño visual, estilos y responsive design
- **GitHub Pages**: Como plataforma de despliegue para sitios web estáticos

Proceso de Despliegue:

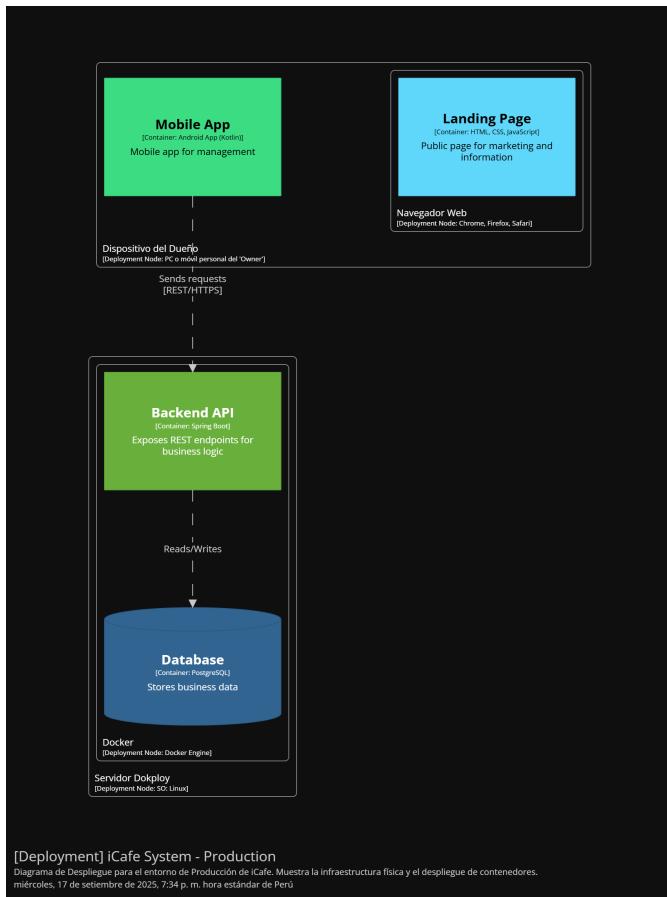
1. **Desarrollo**: La Landing Page fue desarrollada utilizando HTML semántico y CSS moderno, implementando un diseño responsive que se adapta a diferentes dispositivos.
2. **Repositorio**: El código fuente se almacenó en un repositorio de GitHub, manteniendo un control de versiones adecuado.



3. **Despliegue con GitHub Pages**: Se configuró GitHub Pages para servir automáticamente la Landing Page desde el repositorio, proporcionando una URL pública accesible para los usuarios.

The screenshot shows the GitHub Pages settings for the 'LandingPage-iCafe' repository. On the left, there's a sidebar with various GitHub features like General, Access, Code and automation, and Integrations. The main area is titled 'GitHub Pages' and shows the repository is live at <https://lacc0238-2520-1807.github.io/LandingPage-iCafe/>. It was last deployed by SraYess 2 days ago. There are buttons for 'Visit site' and 'Unpublish site'. Below that, under 'Build and deployment', it says 'Deploy from a branch' (set to 'main') and 'Source' (set to '/ (root)'). It also shows a preview of the site. Under 'Custom domain', it says 'Custom domains allow you to serve your site from a domain other than lacc0238-2520-1807.github.io'. Under 'Visibility', it says 'Start free for 30 days'.

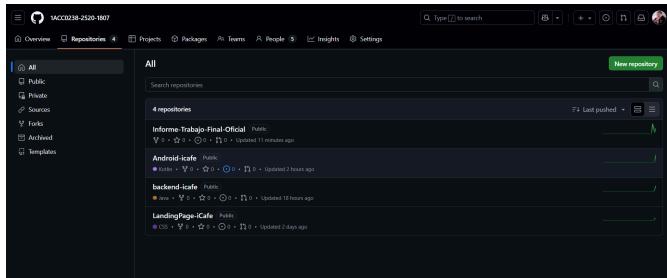
Diagrama C4:



4.2.1.8. Team Collaboration Insights during Sprint

En esta sección se presentarán las analíticas de colaboración del equipo dentro de la plataforma GitHub, donde se evidenciará el trabajo realizado en los tres repositorios principales del proyecto: Landing Page, Backend y Aplicación Móvil. A través de métricas como commits, ramas creadas, pull requests, revisiones y contribuciones por integrante, se busca demostrar la participación activa, la división de responsabilidades y la coordinación continua del equipo durante el desarrollo de cada componente del sistema.

Organizacion: Nuestra organizacion para el curso cuenta con 4 repositorios, donde 3 de estos son para el desarrollo del sistema propuesto.



Colaboradores:

Landing Page:

En el repositorio de la Landing Page, el trabajo fue desarrollado principalmente por un solo integrante del equipo, quien asumió la responsabilidad completa del diseño, desarrollo e implementación de esta parte del proyecto.

Backend:

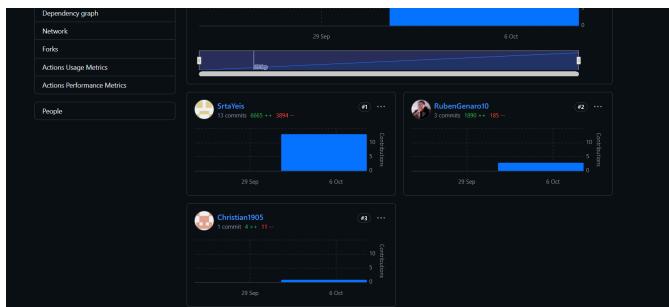
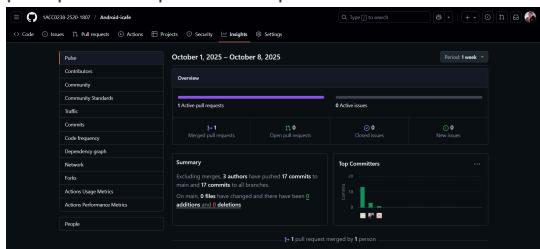
En el repositorio del backend, se evidenció una colaboración activa entre varios miembros del equipo, aplicando correctamente la metodología GitFlow para mantener un flujo de trabajo ordenado y controlado. Cada desarrollador trabajó en ramas específicas (feature, develop, main), lo que permitió integrar cambios de forma segura y realizar revisiones antes de las fusiones finales. Durante este sprint se implementaron los bounded contexts principales: IAM, Products, Contact e Inventory, los cuales constituyen la base funcional del sistema y servirán como infraestructura esencial para los próximos sprints.

En el siguiente gráfico se describe las ramas del repositorio backend donde se muestra una estructura ramificada y bien organizada, propia del uso de GitFlow.

Aplicación Móvil:

El gráfico de colaboración del repositorio correspondiente a la aplicación móvil, desarrollada con Kotlin y Jetpack Compose, muestra una participación continua de los miembros encargados del desarrollo del front-end móvil. Se evidencia una actividad constante en commits y pushes, principalmente en

ramas de tipo feature/ orientadas al diseño de interfaces, integración con el backend y configuración inicial del proyecto siguiendo la arquitectura MVVM y principios de arquitectura limpia.



4.2.2. Sprint 2

Durante el Sprint 2, el equipo de desarrollo de iCafe se enfocará en completar al 100% la implementación del backend, asegurando la correcta integración de todos los módulos y servicios necesarios. Además, se trabajará en finalizar la aplicación móvil para Android, optimizando su funcionalidad y diseño. Paralelamente, se desarrollará una versión inicial de la aplicación en Flutter.

4.2.2.1. Sprint Planning 2

Sprint #	Sprint 2
Sprint Planning Background	
Date	2025-11-05
Time	10:30 PM
Location	Discord
Prepared by	Urrutia Peña Jasmin Adriana
Atendees (to planning meeting)	<ul style="list-style-type: none"> Velaquez Chambi Ruben Genaro Christian Fabrizio inga Orihuela Hallasi Saravia Miguel Angel Sebastian Alexander Ramos Calagua
Sprint 1 Review Summary	<ul style="list-style-type: none"> Se desarrolló una landing page moderna donde se describen nuestros servicios Se construyó la arquitectura central del backend bajo principios de Domain-Driven Design (DDD), implementando los bounded contexts de contacts, iam. Se diseñó e implementó la base de la aplicación móvil usando Jetpack Compose, MVVM y Arquitectura Limpia, logrando la comunicación efectiva con el backend para obtener y enviar datos correctamente
Sprint 1 Retrospective Summary	Durante la retrospectiva del Sprint 1, el equipo coincidió en que la planificación técnica fue acertada, especialmente en la organización por bounded contexts, lo cual facilitó la integración entre módulos.
Sprint Goal & User Stories	
Sprint 2 Goal	<ul style="list-style-type: none"> Implementación final de los bounded contexts de Inventory, Product y Sales, asegurando su integración total dentro del backend. Finalizar la aplicación móvil en Android, consolidando todas sus funcionalidades y conexión con los servicios del backend. Inicio al desarrollo de una versión inicial en Flutter

Sprint #	Sprint 2		
Sprint 2 Velocity	30		
Sum of story points	45		

4.2.2.2. Sprint Backlog 2

Sprint #	Sprint 2						
	User Story		Work-item/Task				
ID	Title	ID	Title	Description	Estimation (Hours)	Assigned To	Status (To-do/In-Process/To-Review/Done)
ID	US012 Registrar entrada inventario	TA01	Endpoint para el registro de entrada de un insumo	Se debe registrar la entrada de un insumo en el inventario	2	Hallasi Saravia Miguel Angel	Done
		TA02	Diseño en la aplicacion movil de inventario	Se debe de crear la actividad de registro de entrada de un insumo en el inventario	1	Hallasi Saravia Miguel Angel	Done
ID	US013 Actualizar inventario	TA01	Endpoint para actualizar datos del inventario	Se debe crear un endpoint para actualizar datos del inventario	1	Hallasi Saravia Miguel Angel	Done
		TA02	Vista para actualizar inventario	Se debe crear la actividad para actualizar datos del inventario	1	Hallasi Saravia Miguel Angel	In-process
ID	US009 Crear receta	TA01	endpoint para crear una receta	Se debe crear un endpoint para crear una receta	1	Ruben Velasquez Chambi	To-do
		TA02	Vista para crear receta	Se debe crear la actividad para crear una receta	1	Ruben Velasquez Chambi	To-do
ID	US007 Listar proveedores	TA01	enpoint para listar proveedores	Se debe poder consultar la lista de los proveedores	1	Ruben Velasquez Chambi	Done
		TA02	Vista de lista de proveedores	Se debe de crear la actividad para visualizar la lista de proveedores	1	Ruben Velasquez Chambi	Done
		TA03	Implementar el backend en la aplicacion movil	Se debe de consumir el endpoint correspondiente	1	Ruben Velasquez Chambi	To-do
ID	US014 Definir precio de venta	TA04	Add Item and Product Section	Se implementó la sección de productos y items dentro de la pagina Alimentos y definir los precios .	2	Jasmin Adriana Urrutia Peña	Done
		TA05	Add Sales and Orders	Se agregó el módulo de ventas y pedidos dentro del sistema.	2	Jasmin Adriana Urrutia Peña	Done

		TA06	Add Finances and Movements	Se desarrolló la sección de finanzas y movimientos, integrando control de transacciones.	2	Jasmin Adriana Urrutia Peña	Done
--	--	------	----------------------------	--	---	-----------------------------	------

4.2.2.3. Development Evidence for Sprint Review

Repository	Branch	Commit ID	Commit message	Commit Message body
https://github.com/1ACC0238-2520-1807/LandingPage-iCafe	main	7dea72270a28a56a1c6139f1984992a8731687f6	Update mobile content	---
	feature/product	ae86845f77f3db881b1bd41bf67c2b5e41dd848d	Se agrego una nueva entidad SupplyItem	---
	feature/product	ba3c106b9d2f51db631778b166c8c07053c613e9	Se mapeo todos los servicios, resource y assembler de SupplyItem, pero falta los controllers	---
	feature/product	2fe06948d62338d53bc69237ac283b6917dee88f	Se limpio el codigo de la version anterior	---
	develop	800648bf1d14398d57bf3c53572cc231813c6d7b	fix: providerId	---
	feature/sales	800648bf1d14398d57bf3c53572cc231813c6d7b	test(sales): update test to use primitive double instead of BigDecimal	---
	feature/products	5a5b03f5d6d73d5f77ff0e5068ed46d53055b916	feat(product): se agrego el endpoint para buscar por branchId en los supplyItem	---
	develop	7782fed28a6336d894945a939866290e354134c1	refactor(purchase-order): extract domain event registration to separ	---
	develop	33ab18af44a4d59550603d62268b5a38db8115f1	use BranchId value object consistently across repositories	---
	main	2c693efaed84c42eb0575902c88bd14b4fa75baa	Update iCafe Design	---
https://github.com/1ACC0238-2520-1807/backend-icafe	develop	729ff1f9dbfc798f578badd4f423b90b5d1360d0	Add Item and Product Section	---
	develop	c7c0984bf860a77464e4158df8fd03d0444daa75	Add Sales and Orders	---
	develop	7193d7727c30b06feef292766d8381ee4abfc83	fix Product Detail	---
	develop	398243bee21af2fc99c004b3d5c1434a2116ceb1	Refactor InventoryMovementsScreen layout and states	---
				1

4.2.2.4. Testing Suite Evidence for Sprint Review

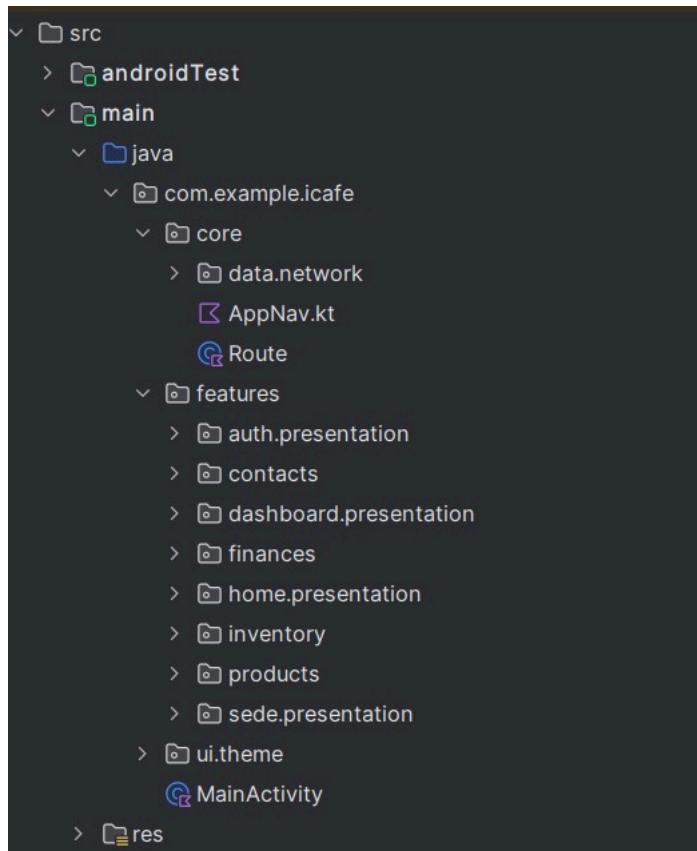
En el Sprint 2, se implementaron las pruebas de aceptación para las historias de usuario y tareas que fueron creadas en el flujo Android.

Repository	Commit ID	Commit Message	Date
Informe-Trabajo-Final-Oficial	04466ad	feat: add feature files for US019-US024 landing pages	12/11/2025

Repository	Commit ID	Commit Message	Date
Informe-Trabajo-Final-Oficial	f272d69	feat: add feature files for inventory, suppliers, pricing and reports	12/11/2025
Informe-Trabajo-Final-Oficial	a52507e	feat: add feature files for US007-US012	12/11/2025
Informe-Trabajo-Final-Oficial	4736d37	feat: add feature files for user stories US001-US006	12/11/2025

Repository <https://github.com/1ACC0238-2520-1807/test>

4.2.2.5. Execution Evidence for Sprint Review

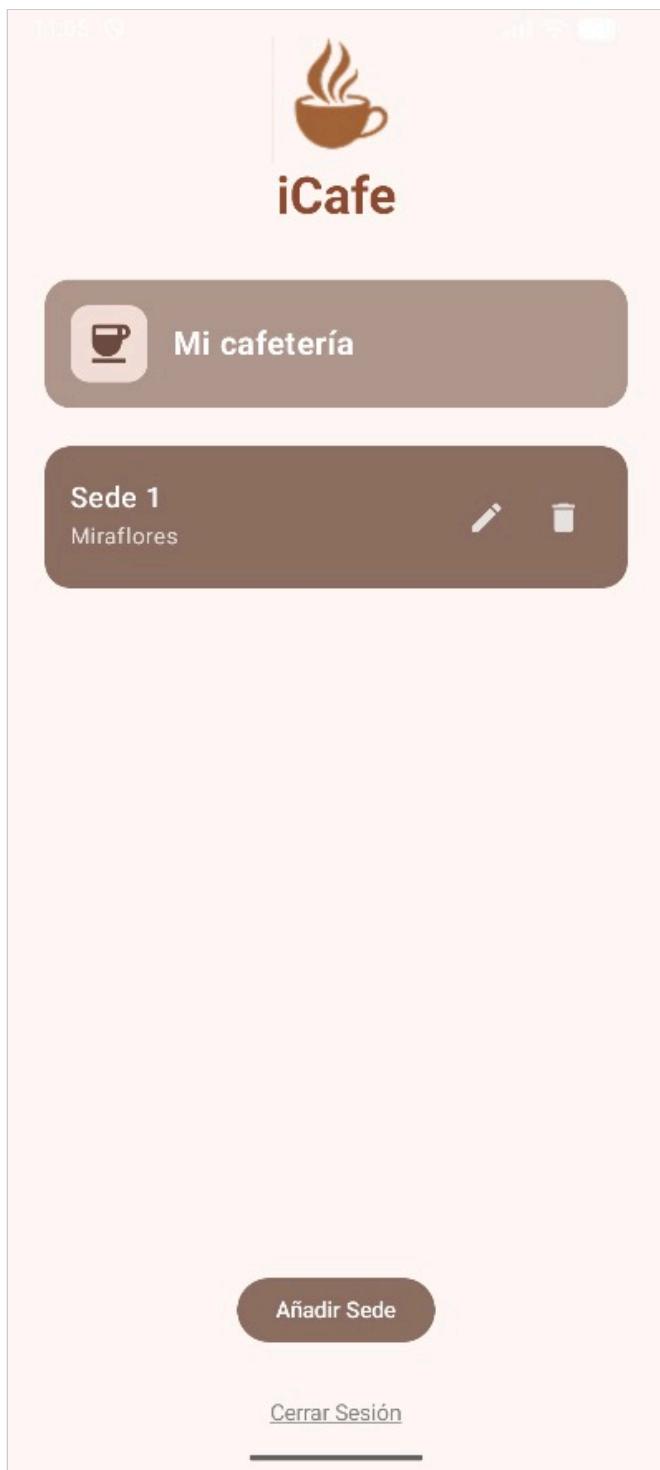


```

v  └─ src
  > └─ androidTest
  < └─ main
    < └─ java
      < └─ com.example.icafe
        < └─ core
          > └─ data.network
            ┌─ AppNav.kt
            └─ Route
        < └─ features
          > └─ auth.presentation
          > └─ contacts
          > └─ dashboard.presentation
          > └─ finances
          > └─ home.presentation
          > └─ inventory
          > └─ products
          > └─ sede.presentation
        > └─ ui.theme
        └─ MainActivity
  > └─ res

```

Vista de las sedes



Vista de la implementación de las sedes dentro de la aplicación Android.

Vista de estadísticas



Vista de estadísticas y métricas dentro de la aplicación Android.

Vista de compras y ventas

Gestión Financiera
Administra tus ventas y compras

Administrador Ventas

Administrador Compras

Vista de movimientos

Movimientos

Movimiento 50

Identificador:	50
Origen:	Compra - Orden #7
Insumo:	Cafe
Tipo de Movimiento:	ENTRADA 200.0 GRAMOS
Fecha:	2025-11-12T21:01:10.084627

Vista de compras y ventas integradas en la aplicación Android.

4.2.2.6. Services Documentation Evidence for Sprint Review

Tabla de Endpoints Documentados:

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada	Parámetros
Inventory	/api/v1/inventory/movements	POST	POST /api/v1/inventory/movements	Body: RegisterStockIV

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada	Parámetros
Inventory	/api/v1/inventory/movements/{branchId}	GET	GET /api/v1/inventory/movements/1	branchId (path)
Inventory	/api/v1/inventory/stock/{branchId}/{supplyItemId}	GET	GET /api/v1/inventory/stock/1/10	branchId , supplyItemId (path)
Products	/api/v1/products	GET	GET /api/v1/products	—
Products	/api/v1/products	POST	POST /api/v1/products	Body: CreateProduct
Products	/api/v1/products/{productId}	GET	GET /api/v1/products/1	productId (path)
Products	/api/v1/products/{productId}	PUT	PUT /api/v1/products/1	productId (path) UpdateProduct
Products	/api/v1/products/{productId}/activate	POST	POST /api/v1/products/1/activate	productId (path)
Products	/api/v1/products/{productId}/archive	POST	POST /api/v1/products/1/archive	productId (path)
Products	/api/v1/products/{productId}/ingredients	POST	POST /api/v1/products/1/ingredients	productId (path) AddIngredients
Products	/api/v1/products/{productId}/ingredients/{supplyItemId}	DELETE	DELETE /api/v1/products/1/ingredients/10	productId , supplyItemId (path)
Products	/api/v1/products/branch/{branchId}	GET	GET /api/v1/products/branch/1	branchId (path)
Purchase Orders	/api/v1/purchase-orders	POST	POST /api/v1/purchase-orders	Body: CreatePurchaseOrder
Purchase Orders	/api/v1/purchase-orders/branch/{branchId}	GET	GET /api/v1/purchase-orders/branch/1	branchId (path)
Sales	/api/v1/sales	GET	GET /api/v1/sales	—
Sales	/api/v1/sales	POST	POST /api/v1/sales	Body: CreateSale
Sales	/api/v1/sales/{saleId}	GET	GET /api/v1/sales/1	saleId (path)
Sales	/api/v1/sales/{saleId}/cancel	PUT	PUT /api/v1/sales/1/cancel	saleId (path)
Sales	/api/v1/sales/{saleId}/complete	PUT	PUT /api/v1/sales/1/complete	saleId (path)
Sales	/api/v1/sales/branch/{branchId}	GET	GET /api/v1/sales/branch/1	branchId (path)

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada	Parámetros
SupplyItem	/api/v1/supply-items	GET	GET /api/v1/supply-items	—
SupplyItem	/api/v1/supply-items	POST	POST /api/v1/supply-items	Body: CreateSupplyItem
SupplyItem	/api/v1/supply-items/{branchId}/branch	GET	GET /api/v1/supply-items/1/branch	branchId (path)
SupplyItem	/api/v1/supply-items/{id}	DELETE	DELETE /api/v1/supply-items/1	id (path)
SupplyItem	/api/v1/supply-items/{id}	GET	GET /api/v1/supply-items/1	id (path)
SupplyItem	/api/v1/supply-items/{id}	PUT	PUT /api/v1/supply-items/1	id (path), Body: UpdateSupplyItem

Swagger del backend desplegado: <http://upc-icafbackend-3sger0-aa823d-31-97-13-234.traefik.me/swagger-ui/index.html>

Sales Sales Management Endpoints

- PUT /api/v1/sales/{saleId}/complete Complete a sale
- PUT /api/v1/sales/{saleId}/cancel Cancel a sale
- GET /api/v1/sales Get all sales
- POST /api/v1/sales Create a new sale
- GET /api/v1/sales/{saleId} Get sale by ID
- GET /api/v1/sales/branch/{branchId} Get sales by branch ID

SupplyItem SupplyItem management operations

- GET /api/v1/supply-items/{id}
- PUT /api/v1/supply-items/{id}
- DELETE /api/v1/supply-items/{id}
- GET /api/v1/supply-items
- POST /api/v1/supply-items
- GET /api/v1/supply-items/{branchId}/branch

Products Product management operations		^
GET	/api/v1/products/{productId}	🔒 ↴
PUT	/api/v1/products/{productId}	🔒 ↴
GET	/api/v1/products	🔒 ↴
POST	/api/v1/products	🔒 ↴
POST	/api/v1/products/{productId}/ingredients	🔒 ↴
POST	/api/v1/products/{productId}/archive	🔒 ↴
POST	/api/v1/products/{productId}/activate	🔒 ↴
GET	/api/v1/products/branch/{branchId}	🔒 ↴
DELETE	/api/v1/products/{productId}/ingredients/{supplyItemId}	⬇️ 🔒 ↴
Purchase Orders Purchase order management operations		^
PUT	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}/confirm	Confirm purchase order
PUT	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}/complete	Complete purchase order
PUT	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}/cancel	Cancel purchase order
POST	/api/v1/purchase-orders	Create purchase order
GET	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}	Get purchase order by ID
GET	/api/v1/purchase-orders/branch/{branchId}	Get purchase orders by branch

- **URL del Repositorio:** <https://github.com/1ACC0238-2520-1807/iCafe-Backend>

4.2.2.7. Software Deployment Evidence for Sprint Review

Configuración del Despliegue del Backend (Web Services)

Para el despliegue del backend de iCafe, se implementó una estrategia de contenerización utilizando Docker Compose, lo que permite una gestión eficiente y escalable de los servicios. El proceso de despliegue se realizó mediante las siguientes tecnologías y configuraciones:

Tecnologías Utilizadas:

- **Docker Compose:** Para la orquestación de contenedores y gestión de servicios
- **Dokploy:** Como plataforma de despliegue y gestión de aplicaciones contenerizadas
- **OpenJDK 17 (openjdk:17-jdk-slim):** Imagen base optimizada para ejecutar aplicaciones Java
- **Spring Boot:** Framework principal para el desarrollo del backend

Proceso de Despliegue:

1. **Contenerización:** El backend fue empaquetado en contenedores Docker utilizando la imagen `openjdk:17-jdk-slim`, que proporciona un entorno ligero y optimizado para aplicaciones Java con Spring Boot.
2. **Configuración con Docker Compose:** Se configuró un archivo `docker-compose.yml` que define el servicio de la aplicación backend.
3. **Despliegue con Dokploy:** Se utilizó Dokploy como plataforma de despliegue, que actúa como una solución de Platform-as-a-Service (PaaS) simplificando la gestión de contenedores y proporcionando funcionalidades de monitoreo, escalado automático y gestión de recursos.

Evidencia Visual del Despliegue:

The screenshot shows the iCafe Backend deployment interface. At the top, there are tabs for General, Environment, Domains, Deployments, Backups, Schedules, Volume Backups, Logs, Monitoring, and Advanced. Under the Deploy Settings tab, there are options for Deploy, Retried, Stop, Open Terminal, and Autodeploy. The Provider section shows GitHub selected as the provider, with a GitHub Account dropdown set to 'iCafeBackend' and a Repository dropdown set to 'backend-icafe'. The Dockerfile is shown as 'backend-icafe/Dockerfile'. The Docker Path is set to 'docker-compose.yml'. The Trigger Type is set to 'On Push'. Watch Paths are listed as 'src/main'. Below this, the Converted Compose section displays the converted docker-compose file:

```

1 services:
2   iCafe-app:
3     build:
4       context: .
5       dockerfile: Dockerfile
6       container_name: iCafe-backend
7     ports:
8       - "8080:8080"
9     environment:
10      - SPRING_PROFILES_ACTIVE=prod
11      - SPRING_DATASOURCE_URL=jdbc:postgresql://cp-frosty-smoke-addyou-pooler.c-2.us-east-1.aws.neon.tech/neondb?sslmode=require&reconnect=true
12      - SPRING_DATASOURCE_USERNAME=neon
13      - SPRING_DATASOURCE_PASSWORD=216968d9
14      - SPRING_H2_INMEMORY_MODE=TRUE
15      - SPRING_H2_SHOW_SQL=False
16      - AUTHENTICATION_JWT_SECRET=yourSecretStringOrTokenSigningCredentials
17      - AUTHENTICATION_JWT_EXPIRATION_DAYS=7
18     restart: unless-stopped
19   networks:
20     - default
21     - docker-network
22   labels:
23     - traefik.http.routers.iCafe-backend.ingress=true
24     - traefik.http.routers.iCafe-backend.ingress-0000=8080
25     - traefik.http.services.iCafe-backend.ingress-0000=web
26     - traefik.http.services.iCafe-backend.ingress-0000=service-iCafe-backend
27     - traefik.docker.network=docker-network
28     - traefik.http.routers.iCafe-backend.ingress-0000=8080.ruleHost=iCafe.vh1-logics.tech

```

The Domains section lists 'iCafe-app' with a URL 'http://iCafe-backend-3ge0-aab23d-31-97-13-234.traefik.me'. The Deployment section shows a log of Maven build steps, indicating a successful build and deployment to Docker:

```

[INFO] #44 10.50 [INFO] --- resources:3.3.1:testResources (default-testResources) @ iCafe ...
[INFO] #44 10.50 [INFO] skip non existing resource directory /app/testResources
[INFO] #44 11.94 [INFO] --- compiler:1.4.13:compile (default-testCompile) @ iCafe ...
[INFO] #44 11.95 [INFO] Recompiling the module because of changed dependency.
[INFO] #44 11.95 [INFO] Compiling 1 source file with Java (using parameters release 17) to target/test-classes
[INFO] #44 12.41 [INFO] --- surefire:5.4:client (default-test) @ iCafe ...
[INFO] #44 12.41 [INFO] No tests were skipped.
[INFO] #44 12.55 [INFO] ...
[INFO] #44 12.55 [INFO] --- Serl:4.2.4:jar (default-jar) @ iCafe ...
[INFO] #44 13.38 [INFO] Building jar: /app/target/iCafe-0.1-SNAPSHOT.jar
[INFO] #44 13.58 [INFO] ...
[INFO] #44 13.58 [INFO] --- spring-boot:3.5.6:repackage (repackage) @ iCafe ...
[INFO] #44 14.20 [INFO] Reloading main artifact /app/target/iCafe-0.1-SNAPSHOT.jar with repackaged archive, adding nested dependency
[INFO] #44 14.20 [INFO] The original artifact has been renamed to /app/target/iCafe-0.1-SNAPSHOT.[ar.original]
[INFO] #44 14.20 [INFO] ...
[INFO] #44 14.20 [INFO] BUILD SUCCESS
[INFO] #44 14.20 [INFO] ...
[INFO] #44 14.20 [INFO] Total time: 14.438 s
[INFO] #44 14.20 [INFO] Finished at: 2025-10-08T08:42:04Z
[INFO] #44 14.20 [INFO] -----
[INFO] #45 Reporting to Image
[INFO] #45 Exporting layers
[INFO] #45 Exporting layers 0-24 done
[INFO] #45 Exporting layers 25-56 done
[INFO] #45 iCafe-backend-3ge0-iCafe-app done
[INFO] #45 iCafe-backend-3ge0-iCafe-app done
[INFO] #45 DONE: 0.2s
[INFO] #45 Wrote manifest provence for metadata file
[INFO] #45 OK: 0.6s
[INFO] iCafe-app Built
[INFO] Container iCafe-backend Running
[INFO] Docker Compose Deployed! ✅

```

Configuración del Despliegue de la Landing Page

Para el despliegue de la Landing Page de iCafe, se implementó una solución web estática utilizando tecnologías fundamentales del desarrollo web y una plataforma de hosting gratuita.

Tecnologías Utilizadas:

- HTML:** Para la estructura y contenido de la página web
- CSS:** Para el diseño visual, estilos y responsive design
- GitHub Pages:** Como plataforma de despliegue para sitios web estáticos

Proceso de Despliegue:

- Desarrollo:** La Landing Page fue desarrollada utilizando HTML semántico y CSS moderno, implementando un diseño responsive que se adapta a diferentes dispositivos.
- Repositorio:** El código fuente se almacenó en un repositorio de GitHub, manteniendo un control de versiones adecuado.

3. Despliegue con GitHub Pages: Se configuró GitHub Pages para servir automáticamente la Landing Page desde el repositorio, proporcionando una URL pública accesible para los usuarios.

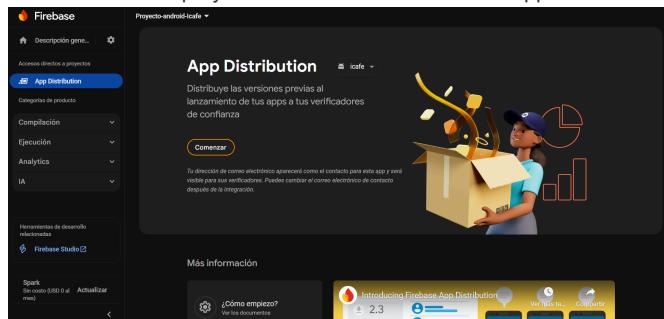
Link: <https://1acc0238-2520-1807.github.io/LandingPage-iCafe/>

Diagrama C4:

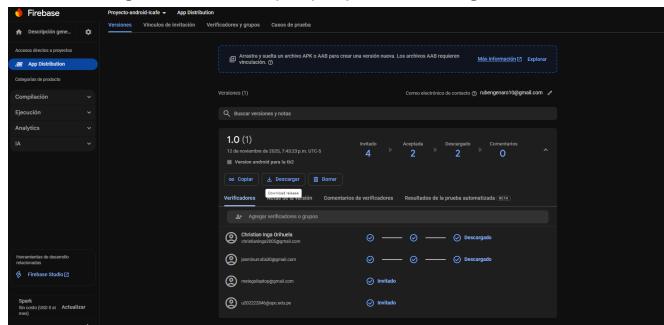


Despliegue en aplicación android en firebase

Se crea un nuevo proyecto en la consola de firebase app distribution



Se debe cargar el archivo .apk que previamente generaste en android studio y añadir los tester a traves de correo.



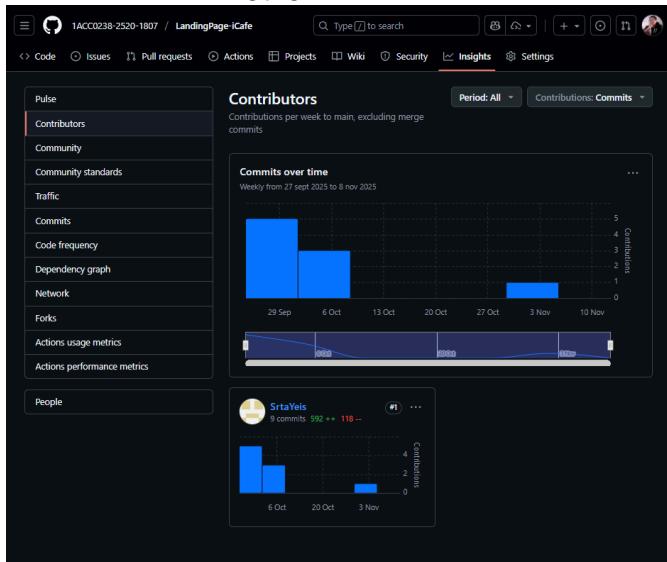
link: https://appdistribution.firebaseio.google.com/testerapps/1:892002161154:android:b7ff4a0dbfb60be018cd1b/releases/2mvv6lj5b2l4o?utm_source=firebase-console

4.2.2.8. Team Collaboration Insights during Sprint

En la siguiente sección se presentarán las evidencias del trabajo colaborativo realizado durante el Sprint 2. En este periodo se llevaron a cabo mejoras en la landing page, el backend desarrollado en Spring Boot, la aplicación Android y la aplicación Flutter. Para sustentar este trabajo conjunto, se mostrarán las gráficas generadas a partir de la actividad registrada en GitHub.

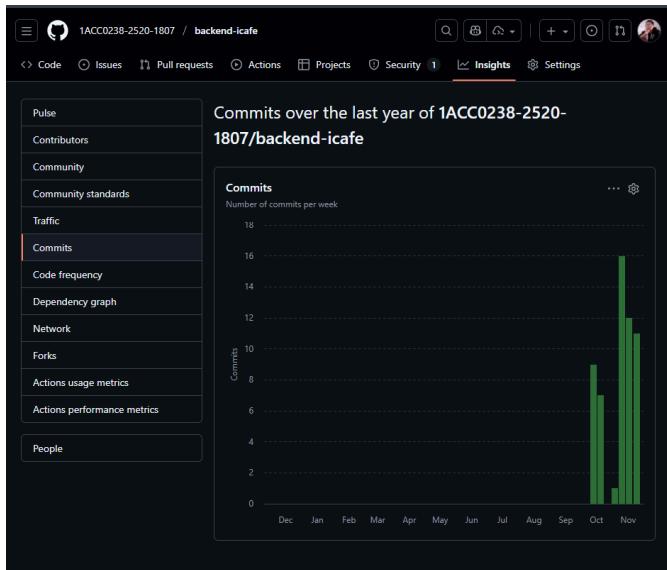
Landing page

Actualización de la landing page

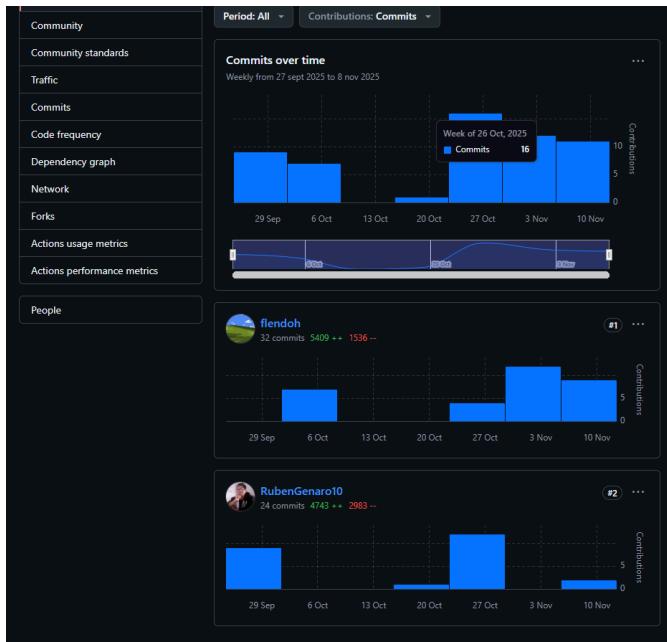


Aplicacion Backend

Fechas en donde se desarrollo el backend durante el sprint 2

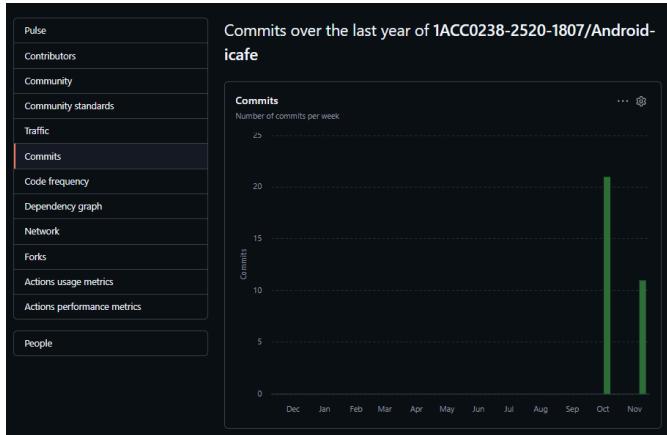


Colaboradores que participaron en el desarrollo de la aplicación backend



Aplicacion Android

Fechas en donde se desarrollo la aplicación Android durante el sprint 2

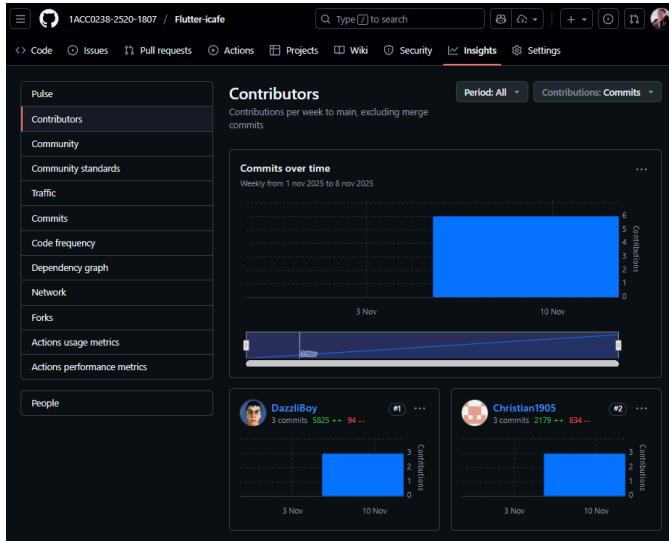


Colaboradores que participaron en el desarrollo de la aplicación android



Aplicacion Flutter

Colaboradores que participaron en el desarrollo de la aplicación flutter



4.2.3. Sprint 3

Durante el Sprint 3, el equipo de desarrollo de iCafe se enfocará en completar la versión integral de la aplicación en Flutter, tanto a nivel de interfaz como de lógica de negocio. Se implementarán todos los módulos pendientes, garantizando la integración total con el backend ya desarrollado, y se refinará la experiencia de usuario mediante mejoras en el diseño, la navegación y el rendimiento. Al finalizar este sprint, se espera contar con una versión estable y funcional de iCafe en Flutter, lista para pruebas y validación con usuarios finales.

4.2.3.1. Sprint Planning 3

Sprint #	Sprint 3
Sprint Planning Background	
Date	2025-11-24
Time	8:30 PM
Location	Discord
Prepared by	Urrutia Peña Jasmin Adriana
Atendees (to planning meeting)	<ul style="list-style-type: none"> Velaquez Chambi Ruben Genaro Christian Fabrizio Inga Orihuela Hallasi Saravia Miguel Angel Sebastian Alexander Ramos Calagua Luciana Carolina Choquehuanca Nuñez
Sprint 2 Review Summary	<ul style="list-style-type: none"> Se completó al 100% la implementación del backend, integrando correctamente todos los módulos y servicios necesarios. Se finalizó la aplicación móvil para Android, consolidando sus funcionalidades y garantizando la comunicación estable con el backend. Se desarrolló una versión inicial de la aplicación en Flutter que replica las funcionalidades básicas de la app Android.
Sprint 2 Retrospective Summary	Durante la retrospectiva del Sprint 2, el equipo coincidió en que priorizar la finalización del backend y la app de Android fue una decisión acertada, ya que permitió reducir riesgos técnicos y validar los principales flujos de negocio. Sin embargo, se identificó la necesidad de mejorar la estimación del esfuerzo en tareas de integración y pruebas, así como de reforzar la documentación de los servicios para facilitar el trabajo en Flutter.
Sprint Goal & User Stories	
Sprint 3 Goal	<ul style="list-style-type: none"> Completar la versión integral de iCafe en Flutter, implementando todas las pantallas, flujos y casos de uso definidos para la aplicación móvil. Integrar la aplicación Flutter con el backend existente, consumiendo los servicios de los bounded contexts relevantes y asegurando la coherencia de datos. Optimizar la experiencia de usuario mediante mejoras en el diseño visual, la navegación y el rendimiento de la aplicación. Preparar una versión estable de iCafe en Flutter lista para pruebas exhaustivas internas y validación con usuarios finales.

Sprint #	Sprint 3					
Sprint 3 Velocity	30					
Sum of story points	45					

4.2.3.2. Sprint Backlog 3

Sprint #	Sprint 3						
User Story		Work-item/Task					
ID	Title	ID	Title	Description	Estimation (Hours)	Assigned To	Status (To-do/In-Process/To-Review/Done)
US025	Registro de cliente	TA01	Pantalla de registro cliente	Diseñar e implementar interfaz de registro para clientes	3	Luciana Carolina Choquehuanca Nuñez	To-do
		TA02	Validación de campos	Implementar validaciones para email, contraseña y datos personales	2	Jasmin Adriana Urrutia Peña	To-do
		TA03	Integración con API	Conectar con endpoint de registro de clientes	2	Hallasi Saravia Miguel Angel	To-do
US026	Ver menú de productos	TA01	Catálogo de productos	Diseñar interfaz para mostrar lista de productos disponibles	4	Christian Fabrizio Inga Orihuela	To-do
		TA02	Filtros y categorías	Implementar filtros por categoría (cafés, postres, etc.)	3	Luciana Carolina Choquehuanca Nuñez	To-do
		TA03	Integración con API	Conectar con endpoint de productos del menú	2	Ruben Velasquez Chambi	To-do
US027	Realizar pedido	TA01	Carrito de compras	Implementar funcionalidad de carrito para agregar productos	5	Sebastian Alexander Ramos Calagua	To-do
		TA02	Proceso de checkout	Diseñar flujo de confirmación y pago del pedido	4	Christian Fabrizio Inga Orihuela	To-do
		TA03	Integración con API	Conectar con endpoint de creación de pedidos	3	Ruben Velasquez Chambi	To-do
US028	Ver estado del pedido	TA01	Pantalla de seguimiento	Diseñar interfaz para mostrar estado del pedido en tiempo real	3	Luciana Carolina Choquehuanca Nuñez	To-do
		TA02	Actualizaciones automáticas	Implementar actualización automática del estado	2	Hallasi Saravia Miguel Angel	To-do
		TA03	Integración con API	Conectar con endpoint de estado de pedidos	2	Jasmin Adriana Urrutia Peña	To-do
US029	Historial de pedidos	TA01	Lista de pedidos anteriores	Diseñar interfaz para mostrar historial de	3	Christian Fabrizio Inga Orihuela	To-do

			pedidos			
		TA02	Detalles de pedido	Implementar vista de detalles para cada pedido histórico	2	Sebastian Alexander Ramos Calagua
		TA03	Integración con API	Conectar con endpoint de historial de pedidos	2	Luciana Carolina Choquehuanca Nuñez
US030	Sistema de valoraciones	TA01	Interfaz de calificación	Diseñar pantalla para calificar productos y servicio	3	Jasmin Adriana Urrutia Peña
		TA02	Envío de reseñas	Implementar funcionalidad para enviar valoraciones	2	Ruben Velasquez Chambi
		TA03	Integración con API	Conectar con endpoint de envío de valoraciones	2	Hallasi Saravia Miguel Angel

4.2.3.3. Development Evidence for Sprint Review

Repository	Branch	Commit ID	Commit message	Commit Message body	C o
https://github.com/1ACC0238-2520-1807/Flutter-icafe	main	efd96a85e1fb3a9b5d3943e812804679dd0ffee6	Feat: Refactor architecture and add homepage	---	13
	feature/product	ba3c106b9d2f51db631778b166c8c07053c613e9	Feat: Added add and edit branch page	---	13
	feature/product	2fe06948d62338d53bc69237ac283b6917dee88f	feat: add create provider and employee feature	---	01
	feature/product	f917c3d61b6d15f306878af7d437ba7bb8eb5380	feat: add Inventory screen.	---	03
https://github.com/1ACC0238-2520-1807/Android-icafe	main	2c693efaed84c42eb0575902c88bd14b4fa75baa	Refactor InventoryMovementsScreen layout and states	---	03
	develop	729ff1f9dbfc798f578badd4f423b90b5d1360d0	Fix: product	---	3/

4.2.3.4. Testing Suite Evidence for Sprint Review

En el Sprint 3, se implementaron las pruebas de aceptación para las historias de usuario y tareas que fueron creadas en el flujo Flutter.

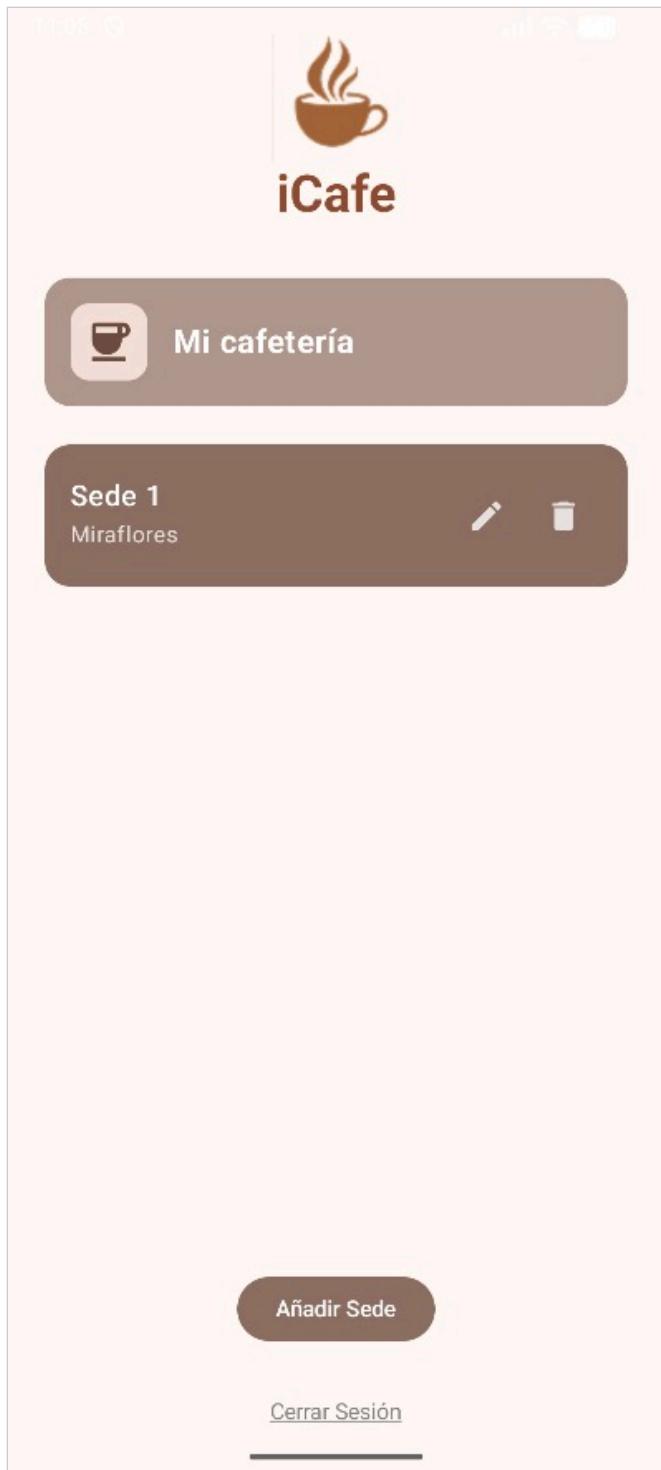
Repository	Commit ID	Commit Message	Date
Informe-Trabajo-Final-Oficial	04466ad	feat: add feature files for US019-US024 landing pages	01/12/2025
Informe-Trabajo-Final-Oficial	f272d69	feat: add feature files for inventory, suppliers, pricing and reports	02/12/2025
Informe-Trabajo-Final-Oficial	a52507e	feat: add feature files for US007-US012	02/12/2025
Informe-Trabajo-Final-Oficial	4736d37	feat: add feature files for user stories US001-US006	02/12/2025

Repositorio <https://github.com/1ACC0238-2520-1807/test>

4.2.3.5. Execution Evidence for Sprint Review

En esta sección describiré las actividades desarrolladas durante el Sprint 2, las cuales abarcan el trabajo realizado en los dos componentes principales del proyecto: la Aplicación Móvil Android y Apliccion Flutter. Se presentarán los avances obtenidos en cada uno de estos módulos, acompañados de capturas de pantalla que evidencian las vistas implementadas y su funcionamiento.

Vista de las sedes



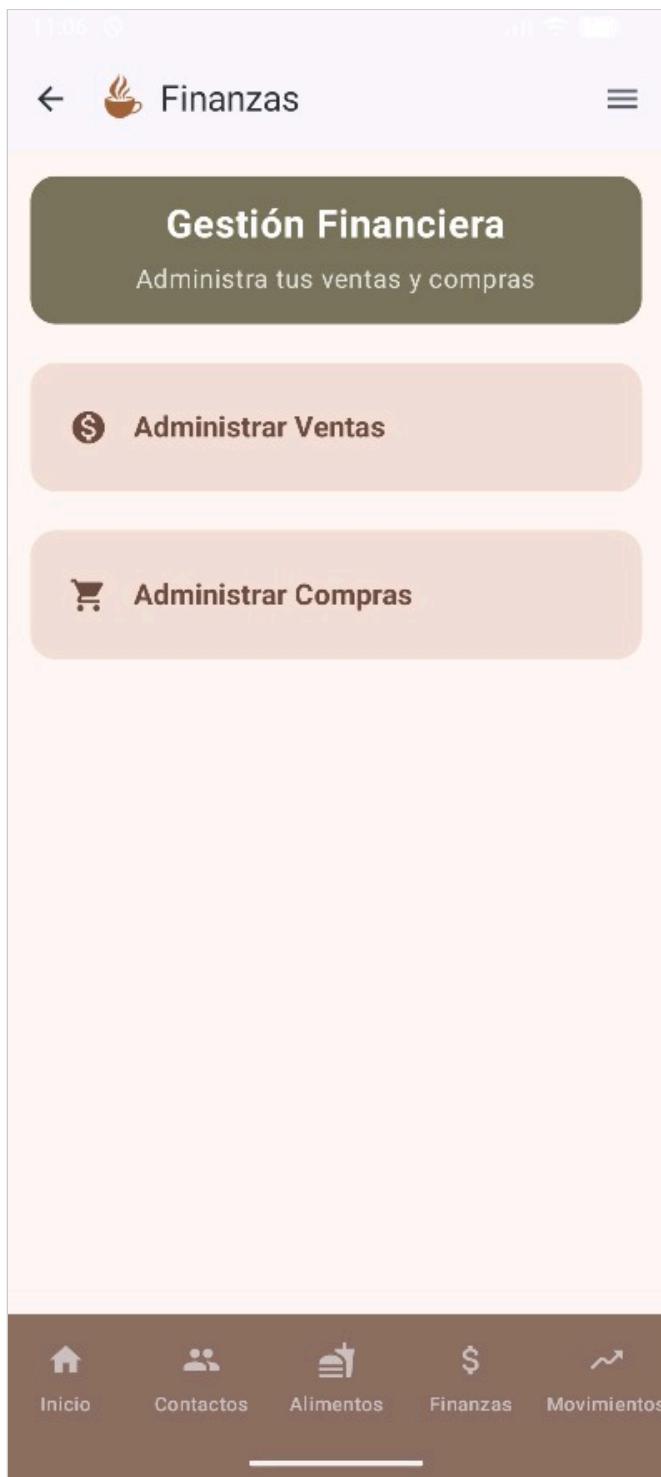
Vista de la implementación de las sedes dentro de la aplicación Android.

Vista de estadísticas



Vista de estadísticas y métricas dentro de la aplicación Android.

Vista de compras y ventas

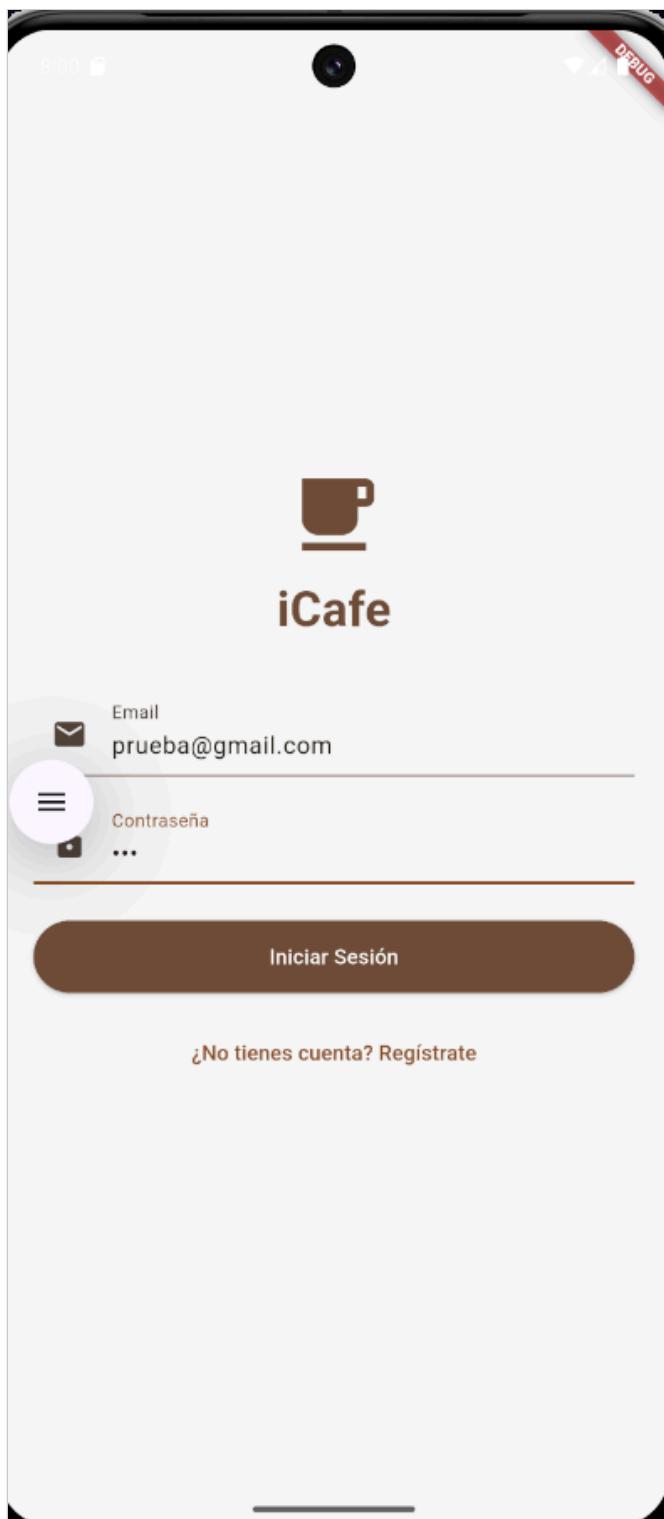


Vista de movimientos



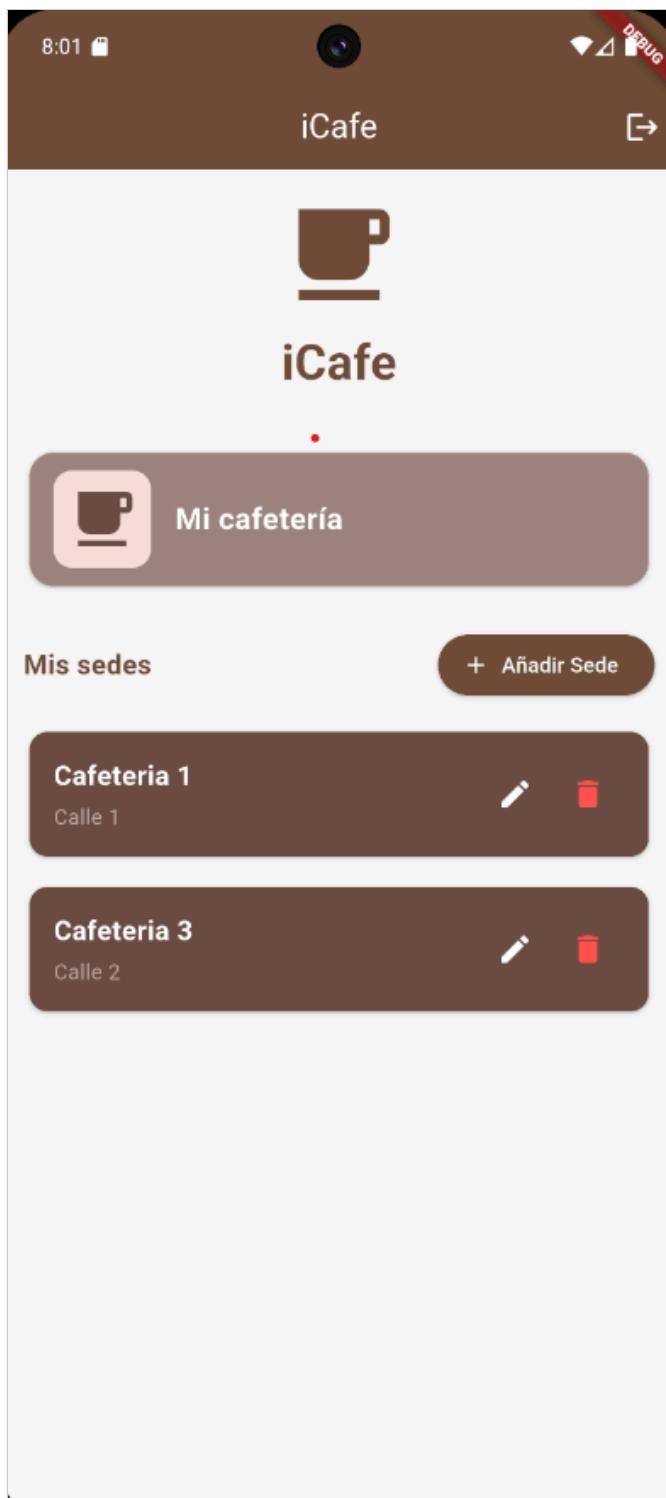
Vista de movimientos y transacciones registradas en la aplicación Android.

Inicio de sesión



Vista al iniciar la aplicación para iniciar sesión con correo y contraseña

Elección de sede



Vista donde podrás elegir la sede por la sede, además de editar la información o en el caso de querer eliminar la sede

Home y resumen

Sede Actual: Cafeteria 1

Resumen de Ventas

Total Vendido	Nº Ventas	Promedio Venta
S/. 212.76	2	S/. 106.38

Empleados
3

Proveedores
2

Insumos
5

Productos
5

Ventas Totales
212.76
S/.

Compras Totales
2200.00
S/.

Inicio **Inventario**

Gestión de Alimentos
Administra tus insumos y productos

Administrar Insumos
Gestiona tus insumos y materias primas

Administrar Productos
Gestiona los productos de tu cafetería

Inventario

Gestión de Alimentos
Administra tus insumos y productos

Administrar Insumos
Gestiona tus insumos y materias primas

Administrar Productos
Gestiona los productos de tu cafetería

Inicio **Contactos** **Alimentos** (highlighted) **Finanzas** **Movimiento**

Vista donde podras visualizar los insumos y productos que tiene tu sede

4.2.3.6. Services Documentation Evidence for Sprint Review

Tabla de Endpoints Documentados:

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada	Parámetros
Inventory	/api/v1/inventory/movements	POST	POST /api/v1/inventory/movements	Body: RegisterStockIV

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada	Parámetros
Inventory	/api/v1/inventory/movements/{branchId}	GET	GET /api/v1/inventory/movements/1	branchId (path)
Inventory	/api/v1/inventory/stock/{branchId}/{supplyItemId}	GET	GET /api/v1/inventory/stock/1/10	branchId , supplyItemId (path)
Products	/api/v1/products	GET	GET /api/v1/products	—
Products	/api/v1/products	POST	POST /api/v1/products	Body: CreateProduct
Products	/api/v1/products/{productId}	GET	GET /api/v1/products/1	productId (path)
Products	/api/v1/products/{productId}	PUT	PUT /api/v1/products/1	productId (path) UpdateProduct
Products	/api/v1/products/{productId}/activate	POST	POST /api/v1/products/1/activate	productId (path)
Products	/api/v1/products/{productId}/archive	POST	POST /api/v1/products/1/archive	productId (path)
Products	/api/v1/products/{productId}/ingredients	POST	POST /api/v1/products/1/ingredients	productId (path) AddIngredients
Products	/api/v1/products/{productId}/ingredients/{supplyItemId}	DELETE	DELETE /api/v1/products/1/ingredients/10	productId , supplyItemId (path)
Products	/api/v1/products/branch/{branchId}	GET	GET /api/v1/products/branch/1	branchId (path)
Purchase Orders	/api/v1/purchase-orders	POST	POST /api/v1/purchase-orders	Body: CreatePurchaseOrder
Purchase Orders	/api/v1/purchase-orders/branch/{branchId}	GET	GET /api/v1/purchase-orders/branch/1	branchId (path)
Sales	/api/v1/sales	GET	GET /api/v1/sales	—
Sales	/api/v1/sales	POST	POST /api/v1/sales	Body: CreateSale
Sales	/api/v1/sales/{saleId}	GET	GET /api/v1/sales/1	saleId (path)
Sales	/api/v1/sales/{saleId}/cancel	PUT	PUT /api/v1/sales/1/cancel	saleId (path)
Sales	/api/v1/sales/{saleId}/complete	PUT	PUT /api/v1/sales/1/complete	saleId (path)
Sales	/api/v1/sales/branch/{branchId}	GET	GET /api/v1/sales/branch/1	branchId (path)

Bounded Context	Endpoint	HTTP Verb	Sintaxis de Llamada	Parámetros
SupplyItem	/api/v1/supply-items	GET	GET /api/v1/supply-items	—
SupplyItem	/api/v1/supply-items	POST	POST /api/v1/supply-items	Body: CreateSupplyItem
SupplyItem	/api/v1/supply-items/{branchId}/branch	GET	GET /api/v1/supply-items/1/branch	branchId (path)
SupplyItem	/api/v1/supply-items/{id}	DELETE	DELETE /api/v1/supply-items/1	id (path)
SupplyItem	/api/v1/supply-items/{id}	GET	GET /api/v1/supply-items/1	id (path)
SupplyItem	/api/v1/supply-items/{id}	PUT	PUT /api/v1/supply-items/1	id (path), Body: UpdateSupplyItem

Swagger del backend desplegado: <http://upc-icafbackend-3sger0-aa823d-31-97-13-234.traefik.me/swagger-ui/index.html>

Sales Sales Management Endpoints

- PUT /api/v1/sales/{saleId}/complete Complete a sale
- PUT /api/v1/sales/{saleId}/cancel Cancel a sale
- GET /api/v1/sales Get all sales
- POST /api/v1/sales Create a new sale
- GET /api/v1/sales/{saleId} Get sale by ID
- GET /api/v1/sales/branch/{branchId} Get sales by branch ID

SupplyItem SupplyItem management operations

- GET /api/v1/supply-items/{id}
- PUT /api/v1/supply-items/{id}
- DELETE /api/v1/supply-items/{id}
- GET /api/v1/supply-items
- POST /api/v1/supply-items
- GET /api/v1/supply-items/{branchId}/branch

Products		Product management operations	^
GET	/api/v1/products/{productId}		🔒 ↴
PUT	/api/v1/products/{productId}		🔒 ↴
GET	/api/v1/products		🔒 ↴
POST	/api/v1/products		🔒 ↴
POST	/api/v1/products/{productId}/ingredients		🔒 ↴
POST	/api/v1/products/{productId}/archive		🔒 ↴
POST	/api/v1/products/{productId}/activate		🔒 ↴
GET	/api/v1/products/branch/{branchId}		🔒 ↴
DELETE	/api/v1/products/{productId}/ingredients/{supplyItemId}		⬇️ 🔒 ↴
Purchase Orders		Purchase order management operations	^
PUT	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}/confirm	Confirm purchase order	🔒 ↴
PUT	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}/complete	Complete purchase order	🔒 ↴
PUT	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}/cancel	Cancel purchase order	🔒 ↴
POST	/api/v1/purchase-orders	Create purchase order	🔒 ↴
GET	/api/v1/purchase-orders/{purchaseOrderId}/branch/{branchId}	Get purchase order by ID	🔒 ↴
GET	/api/v1/purchase-orders/branch/{branchId}	Get purchase orders by branch	🔒 ↴

- **URL del Repositorio:** <https://github.com/1ACC0238-2520-1807/iCafe-Backend>

4.2.3.7. Software Deployment Evidence for Sprint Review

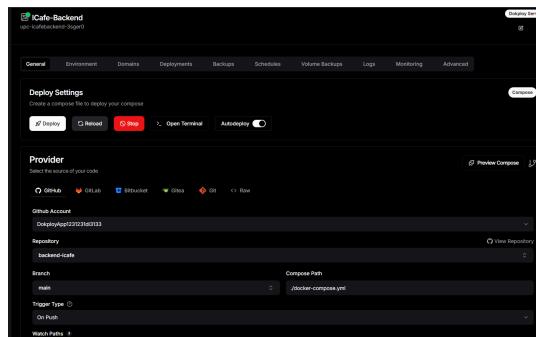
Configuración del Despliegue del Backend (Web Services)

Para el despliegue del backend de iCafe, se empleó un enfoque basado en contenedores, utilizando Docker Compose como herramienta principal para la orquestación. Esta estrategia permite una gestión más eficiente, escalable y consistente de los servicios en distintos entornos. A continuación, se presentan las tecnologías utilizadas y el flujo seguido durante el despliegue:

Tecnologías Utilizadas:

- **Docker Compose:** Para la orquestación de contenedores y gestión de servicios
- **Dokploy:** Como plataforma de despliegue y gestión de aplicaciones contenerizadas
- **OpenJDK 17 (openjdk:17-jdk-slim):** Imagen base optimizada para ejecutar aplicaciones Java
- **Spring Boot:** Framework principal para el desarrollo del backend

Evidencia Visual del Despliegue:



The screenshot shows the Docker Compose interface with three main sections:

- Converted Compose:** Displays the converted Docker Compose file with added domains.
- Domains:** Shows the domain configuration for the application.
- Deployment:** Displays the deployment logs for the iCafe service.

```

version: '3'
services:
  iCafeApp:
    build:
      context: .
      dockerfile: Dockerfile
      container_name: iCafe-backend
    environment:
      - SPRING_DATASOURCE_URL=jdbc:postgresql://cp-frosty-smoke-addyourpooler.c-2.un-east-1.aws.neon.tech/reordb?sslmode=require&charSet=UTF8
      - SPRING_DATASOURCE_USERNAME=neonb_owner
      - SPRING_DATASOURCE_PASSWORD=Zqf5MqAysRQ
      - SPRING_DATASOURCE_MAX_WAIT=10000
      - SPRING_DATASOURCE_MAX_UPDATE=1000
      - SPRING_H2_SHOW_SQL=true
      - AUTHORIZATION_NFT_SECRET=writehereYourSecretStringOrTokenSignature!credentials
      - AUTHORIZATION_NFT_EXPIRATION_DAYS=7
    ports:
      - 8080:8080
    networks:
      - iCafe-network
      - proxy-network
    labels:
      - traefik.http.routers.iCafebackend.ingress=true
      - traefik.http.routers.iCafebackend.ingress-regex=^http://iCafe\.backend\.(?P<port>\d{2})\.iCafe\.net$&traefik.http.routers.iCafebackend.ingress-regex-priority=10000
      - traefik.http.routers.iCafebackend.ingress-regex-path=/api/v1/web
      - traefik.http.routers.iCafebackend.ingress-regex-path-prefix=/api/v1/web
      - traefik.docker.network=proxy-network
      - traefik.http.routers.iCafebackend.ingress-regex=^http://iCafe\.api\.iCafe\.net$&traefik.http.routers.iCafebackend.ingress-regex-path=/api/v1/loics.tech"
networks:
  iCafe-network
  proxy-network
  labels:
    - traefik.http.routers.iCafebackend.ingress=true
    - traefik.http.routers.iCafebackend.ingress-regex=^http://iCafe\.backend\.(?P<port>\d{2})\.iCafe\.net$&traefik.http.routers.iCafebackend.ingress-regex-priority=10000
    - traefik.http.routers.iCafebackend.ingress-regex-path=/api/v1/web
    - traefik.http.routers.iCafebackend.ingress-regex-path-prefix=/api/v1/web
    - traefik.docker.network=proxy-network
    - traefik.http.routers.iCafebackend.ingress-regex=^http://iCafe\.api\.iCafe\.net$&traefik.http.routers.iCafebackend.ingress-regex-path=/api/v1/loics.tech"

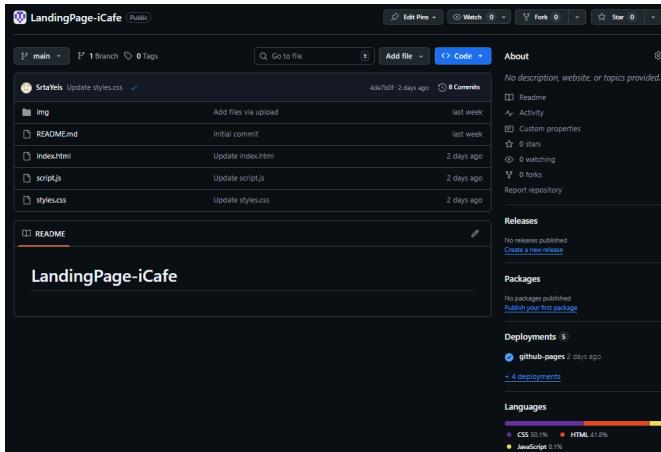
```

Configuración del Despliegue de la Landing Page

La Landing Page del proyecto iCafe fue desplegada utilizando un enfoque de sitio estático, apoyado en tecnologías web fundamentales y utilizando una plataforma de hosting gratuita que facilita su publicación.

Tecnologías Utilizadas:

- HTML:** Para la estructura y contenido de la página web
- CSS:** Para el diseño visual, estilos y responsive design
- GitHub Pages:** Como plataforma de despliegue para sitios web estáticos



Despliegue con GitHub Pages: Se configuró GitHub Pages para servir automáticamente la Landing Page desde el repositorio, proporcionando una URL pública accesible para los usuarios.

Link: <https://1acc0238-2520-1807.github.io/LandingPage-iCafe/>

GitHub Pages

Your site is live at <https://iacc0238-2520-1807.github.io/LandingPage-iCafe/>

Last deployed 2 days ago

[Visit site](#) [Unpublish site](#)

Build and deployment

Source: Deploy from a branch

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

Save

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

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment workflow](#). [Learn more about deploying to GitHub Pages using custom workflows](#).

Custom domain

Custom domains allow you to serve your site from a domain other than [iacc0238-2520-1807.github.io](#). [Learn more about configuring custom domains](#).

[Save](#) [Remove](#)

Enforce HTTPS

— Required for your site because you are using the default domain (iacc0238-2520-1807.github.io)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more about securing your GitHub Pages site with HTTPS](#).

Visibility [GitHub Enterprise](#)

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

[Start free for 30 days](#)

iCafe

Inicio Servicios Nosotros Contactos

Tu cafetería, más organizada que nunca.

UNA SOLUCIÓN DISEÑADA PARA DURMIOS DE CAFETERÍAS QUE BUSCAN OPTIMIZAR INVENTARIOS, VENTAS Y COSTOS EN UN SOLO LUGAR.

Solicitar una demo →

Despliegue en aplicación android en firebase

Para distribuir la aplicación Android, se utilizó Firebase App Distribution, permitiendo compartir builds con testers de forma rápida y sencilla.

Se creó un proyecto en la consola de Firebase destinado exclusivamente a la distribución de la app.

Firebase

Proyecto-android-taste

App Distribution

Distribuye las versiones previas al lanzamiento de tus apps a tus verificadores de confianza

Comenzar

Más información

Cómo empiezo? Ver los documentos 2.3 Introducing Firebase App Distribution We Use You Contacto

Posteriormente, se cargó el archivo .apk generado desde Android Studio y se agregaron los testers mediante correo electrónico.

App Distribution

Verificadores

Anímate a subir un archivo APK o AAB para crear una versión nueva. Los archivos AAB requieren Google Play Protect.

Verificadores (0) Correo electrónico de contacto: nihengnrao@gmail.com

1.0 (1) Invitado 4 > Aceptada 2 > Descargado 2 > Comentada 0

Verificadores

Verificadores invitados Comentarios de la prueba automatizada (0)

Agregar verificadores o grupos

Destinatarios (0) Correo electrónico de contacto: nihengnrao@gmail.com

nikunjrao@gmail.com

nikunjrao@gmail.com

nihengnrao@gmail.com

nihengnrao@gmail.com

link: <https://appdistribution.firebaseio.google.com/testerapps/1:892002161154:android:b7ff4a0dbfb60be018cd1b/releases/2mvv6lj5b2l4o>

utm_source=firebase-console

4.2.3.8. Team Collaboration Insights during Sprint

4.3. Validation Interviews

En esta sección, validaremos nuestro producto final con entrevistas a nuestro segmento objetivo.

4.3.1. Diseño de Entrevistas

Objetivo:

Validar la aplicación móvil con dueños de cafeterías pequeñas, evaluando la usabilidad, el diseño y la utilidad de la herramienta en la gestión de su negocio.

Preguntas de validación:

- ¿Considera que el diseño de la aplicación es visualmente atractivo y adecuado para su negocio?
- ¿Diría que la interfaz es intuitiva y amigable, incluso sin explicaciones previas?
- ¿Las funciones disponibles (como registrar ventas, agregar productos o ver movimientos financieros) le parecen útiles para su cafetería?
- ¿Considera que esta aplicación podría facilitar la administración diaria de su negocio?
- ¿Hubo algo que no entendió o que le pareció poco claro en la interfaz?
- ¿Cree que esta aplicación podría ahorrarle tiempo o ayudarle a tener un mejor control de su cafetería?
- ¿Qué mejoras o sugerencias haría para que la aplicación se adapte mejor a las necesidades de su cafetería?
- Si la aplicación estuviera disponible en el mercado, ¿estaría dispuesta(o) a utilizarla de manera constante?

4.3.2. Registro de Entrevistas

Entrevista 1:

Datos del entrevistador:

- Nombre: Inga Orihuela, Christian Fabrizio

Datos del entrevistado:

- Nombre: Jazmin
- Edad: 30 años
- Residencia: Lima
- Enlace: dfg



Resumen de la entrevista:

fdgdfg

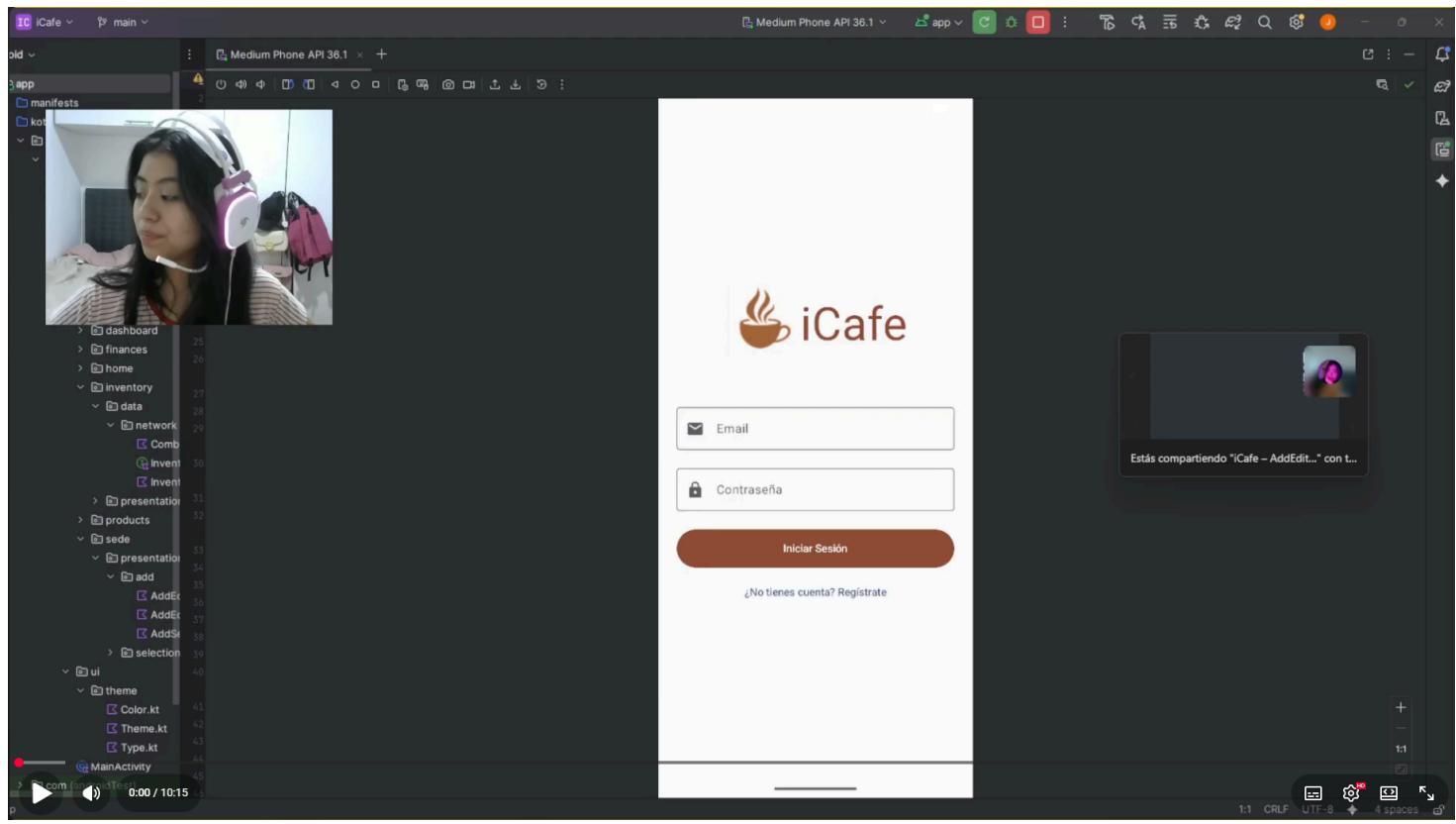
Entrevista 2:

Datos del entrevistador:

- Nombre: Urrutia Peña, Jasmin Adriana

Datos del entrevistado:

- Nombre: Melanie Espinoza
- Edad: 34 años
- Residencia: Huancayo
- Enlace: https://youtu.be/IQ_PUh31_XY



Resumen de la entrevista:

La entrevistada considera que la aplicación es útil, clara y sencilla de usar. Destaca que le permitirá mejorar el control y la administración de su cafetería, especialmente en cuanto al inventario, registro de insumos, ventas y compras. Menciona que actualmente existe desperdicio de insumos, y que la aplicación le ayudaría a reducirlo gracias a un manejo más ordenado. También señala que tener estadísticas y un historial de movimientos le permitiría tomar mejores decisiones y obtener una rentabilidad más real y precisa. En general, ve la propuesta como una solución práctica, alineada con la tecnología actual, y afirma que sí estaría interesada en usarla.

4.3.3. Evaluaciones según heurísticas

UX Heuristics & Principles Evaluation

Usability – Inclusive Design – Information Architecture

CARRERA: Ingeniería de Software

CURSO: CC238

SECCIÓN: 1807

PROFESORES: Todos

AUDITOR: SyngCafe

CLIENTE(S): Melanie espinoza y Jazmin

SITE o APP A EVALUAR

iCafe

TAREAS A EVALUAR

1. Registrar empleados
2. Registrar proveedores
3. Registrar insumos
4. Registrar productos

5. Registrar una venta
6. Registrar una compra
7. Visualizar historial de movimientos
8. Consultar estadísticas de ventas y compras

No incluidas:

1. Generación de reportes avanzados
2. Exportación de datos
3. Gestión de usuarios con roles
4. Integración con facturación externa

ESCALA DE SEVERIDAD

Nivel	Descripción
1	Problema superficial
2	Problema menor
3	Problema mayor
4	Problema muy grave

TABLA RESUMEN

#	Problema	Severidad	Heurística violada
1	No hay botón para regresar al menú	2	Libertad y control del usuario
2	Botones poco descriptivos	2	Visibilidad del estado
3	Sin buscador en listas	3	Is it findable?
4	Campos sin validación	4	Reconocimiento y corrección

DESCRIPCIÓN DE PROBLEMAS

PROBLEMA #1

Heurística violada: Usability – Libertad y control del usuario

Problema:

En secciones como “Registrar insumos” o “Registrar productos”, no existe un botón visible para regresar al menú principal. Esto obliga al usuario a retroceder manualmente varias veces y genera pérdida de tiempo.

Recomendación:

Incluir un botón de “Volver al menú” en todas las pantallas internas.

PROBLEMA #2

Heurística violada: Usability – Visibilidad del estado del sistema

Problema:

Botones como “Agregar” no especifican si corresponde a un proveedor, insumo o empleado, causando confusión.

Recomendación:

Renombrar botones a: “Agregar proveedor”, “Agregar insumo”, “Agregar empleado”, etc.

PROBLEMA #3

Heurística violada: Information Architecture – Is it findable?

Problema:

Cuando la lista crece, es difícil encontrar elementos sin una barra de búsqueda o filtros.

Recomendación:

Agregar buscador y filtros por categoría, proveedor o stock.

PROBLEMA #4

Severidad: 4

Heurística violada: Usability – Ayuda al usuario a reconocer y corregir errores

Problema:

El sistema permite ingresar valores inválidos (stock negativo, precios en cero), afectando la confiabilidad del inventario.

Recomendación:

Agregar validaciones y mensajes de error claros: "El stock no puede ser negativo".

Conclusiones

Conclusiones y recomendaciones

Conclusiones:

- El proyecto abordó eficazmente la desorganización y la falta de herramientas digitales en cafeterías pequeñas y medianas, ofreciendo una solución clara para la gestión de inventario, proveedores, empleados y costos.
- La adopción de una arquitectura robusta con Domain-Driven Design (DDD) y Bounded Contexts bien definidos proporciona una base escalable y mantenible para el sistema.
- El enfoque en el usuario, evidenciado por las entrevistas, User Personas y User Journey Mappings, resultó en un diseño de interfaz intuitivo y fácil de usar.
- La implementación de una Landing Page, un backend funcional (con Swagger) y una aplicación Android operativa demuestra la capacidad del equipo para transformar ideas en un producto tangible.
- La colaboración efectiva, la planificación de sprints y el uso de herramientas de control de versiones reflejan una buena organización y coordinación del equipo.

Recomendaciones:

- Priorizar la ejecución y documentación de las entrevistas de validación con usuarios reales para obtener retroalimentación directa y refinar la aplicación.
- Completar la implementación de todas las funcionalidades críticas en la aplicación móvil para Android, especialmente las relacionadas con ventas, finanzas y reportes detallados, y consolidar el desarrollo en Flutter.
- Implementar herramientas de monitoreo de rendimiento, errores y uso en el backend y la aplicación móvil una vez que el producto esté en manos de los usuarios.
- Desarrollar una estrategia de crecimiento y marketing más amplia, explorando marketing digital, alianzas con proveedores o demostraciones dirigidas para expandir la base de usuarios.
- Continuar el aprendizaje en nuevas áreas como la analítica de datos o la inteligencia artificial para ofrecer funcionalidades más avanzadas y añadir valor al producto.

Video App Validation

Link: [https://www.canva.com/design/DAG4mwAyték/GYEclk_B8KDLbPRLAUUmKw/edit?](https://www.canva.com/design/DAG4mwAyték/GYEclk_B8KDLbPRLAUUmKw/edit?utm_content=DAG4mwAyték&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

[utm_content=DAG4mwAyték&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton](https://www.canva.com/design/DAG4mwAyték/GYEclk_B8KDLbPRLAUUmKw/edit?utm_content=DAG4mwAyték&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

Video About the product

Link: <https://youtu.be/wF37vzb0DtA>

Video About the team

Link: <https://youtu.be/Cj0DfL8WPmA>

Glosario

Bibliografía

- Cuéllar, J. (2016). *Domain-Driven Design: Context Maps*. Jose Cuéllar .net. <https://josecuellar.net/domain-driven-design-episodio-ii-context-maps/>
- Huang, Z. (2023). A comparative analysis between chain coffee shops and independent coffee shops from consumer satisfaction perspective. In Proceedings of the 2022 3rd International Conference on E-commerce and Internet Technology (ECIT 2022) (pp. 1438-1444). Atlantis Press. https://doi.org/10.2991/978-94-6463-005-3_148
- Lee, K.-S., & Ruck, K. J. (2022). Barista diary: An autoethnography studying the operational experience of third-wave coffee shop baristas. International Journal of Hospitality Management, 102, Article 103182. <https://doi.org/10.1016/j.ijhm.2022.103182>
- Molina, S. A. C. (2023). Desarrollar una aplicación web responsiva para la gestión de reservas y administración de restaurante [Tesis de pregrado, Pontificia Universidad Católica del Ecuador]. Repositorio PUCE. <https://repositorio.puce.edu.ec/handle/123456789/41306>
- Sidabutar, G. D. G. E., Seah, J. A., & Singgalen, Y. A. (2023). Analysis and design of web-based information system for coffeeshop management using design thinking methodology: Case of Kopi KurangLebih. Journal of Information Systems and Informatics, 5(1), 217-231. <https://doi.org/10.51519/journalisi.v5i1.455>

Anexos

- Link Repositorio Informe: <https://github.com/1ACC0238-2520-1807/Informe-Trabajo-Final-Oficial>
- Link Repositorio Landing Page: <https://github.com/1ACC0238-2520-1807/LandingPage-iCafe>
- Landing Page Deployado: <https://1acc0238-2520-1807.github.io/LandingPage-iCafe/>
- Link Repositorio Backend: <https://github.com/1ACC0238-2520-1807/backend-icafe>
- Backend Deployado: <http://upc-icafebackend-3sger0-aa823d-31-97-13-234.traefik.me/swagger-ui/index.html#>
- Link Repositorio Android: <https://github.com/1ACC0238-2520-1807/Android-icafe>
- Android Deployado: <https://appdistribution.firebaseio.dev/i/996291fcf7f9d0d1>
- Link Repositorio Flutter: <https://github.com/1ACC0238-2520-1807/Flutter-icafe>
- Link PPT: https://www.canva.com/design/DAGzPbC_J7U/kQe16BT9So7GJgFTXdkFGg/edit?utm_content=DAGzPbC_J7U&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton