



# POSGRADOS

## MAESTRÍA EN SOFTWARE CON MENCIÓN EN DISEÑO DE ARQUITECTURA DE SISTEMAS

RPC-SO-34-NO.778-2021

### OPCIÓN DE TITULACIÓN:

PROYECTO DE TITULACIÓN CON  
COMPONENTES DE INVESTIGACIÓN  
APLICADA Y/O DE DESARROLLO

### TEMA:

APLICACIÓN DE PRINCIPIOS DEVOPS Y  
METODOLOGÍAS ÁGILES PARA EL DESARROLLO  
DEL MÓDULO DEL HISTORIAL FINANCIERO DE  
LA COOPERATIVA DE AHORRO Y CRÉDITO  
JARDÍN AZUAYO.

### AUTORES:

EDUARDO GUILLERMO AGUILAR RIERA  
JUAN GABRIEL NAULA SANISACA

### DIRECTOR:

CINTYA DE LA NUBE AGUIRRE BRITO

CUENCA – ECUADOR  
2025

## **Autores:**



### **Eduardo Guillermo Aguilar Riera**

Ingeniero de Sistemas.

Candidato a Magíster en Software con mención en Diseño de Arquitectura de Sistemas por la Universidad Politécnica Salesiana – Sede Cuenca.  
eaguilarr1@est.ups.edu.ec|



### **Juan Gabriel Naula Sanisaca**

Ingeniero de Sistemas Informáticos.

Candidato a Magíster en Software con mención en Diseño de Arquitectura de Sistemas por la Universidad Politécnica Salesiana – Sede Cuenca.  
jnaulas@est.ups.edu.ec

## **Dirigido por:**



### **Cintya de la Nube Aguirre Brito**

Ingeniera de Sistemas.

Máster Of Science In Software Engineering.  
cintya.aguirre@gmail.com

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

#### **DERECHOS RESERVADOS**

2025 © Universidad Politécnica Salesiana.

CUENCA – ECUADOR – SUDAMÉRICA

EDUARDO GUILLERMO AGUILAR RIERA

JUAN GABRIEL NAULA SANISACA

Aplicación de principios DEVOPS y metodologías ágiles para el desarrollo del módulo del historial financiero de la Cooperativa de Ahorro y Crédito Jardín Azuayo.

## **DEDICATORIA**

A mis padres Guillermo y Marcia, por su amor, por su apoyo incondicional en cada paso de este camino, por enseñarme a no darme por vencido y que cuando uno se cae tiene que volver a levantarse con más fuerza.

A mis hermanas Sandy y Caro por su compañía, por sus consejos, por sus chistes, por su aliento constante, por ayudarme y pese a ser las menores ser mis hermanas mayores en muchos momentos.

Al Milo, mi fiel compañero, que siempre supo cómo hacerme sonreír en los momentos difíciles, por ser una fuente inagotable de alegría y compañía en los momentos de cansancio y estrés.

Finalmente, gracias a mi amor por siempre motivarme a alcanzar esta meta, recordándome que el esfuerzo vale la pena, por todo el amor que me brindas, por estar conmigo pese a lo difícil que puede parecer el camino y por no dejarme caer nunca, por fin te puedo llamar colega.

Gracias a todos por ser parte de este camino.

Edú.

## **AGRADECIMIENTO**

Quiero expresar mi más profundo agradecimiento a mis padres, quienes con su amor, apoyo incondicional y consejos siempre han sido mi guía y fortaleza en todos los momentos complicados. A mis hermanas, por ser mis compañeras en cada etapa de mi vida, brindándome siempre su ánimo y su cariño inagotable.

De igual manera un agradecimiento especial a la cooperativa por apoyarme para lograr conseguir esta meta, por permitirme formar profesional y académicamente.

A mi directora de tesis y amiga Chin quien nos ayudó cuando nos dábamos por vencidos y que nos presionaba siempre para hacer mucho más, gracias por las habladas nos sirvieron de mucho.

A Juan, mi compañero de tesis, gracias por el esfuerzo compartido, las largas horas de trabajo y por todo el apoyo durante todo este proceso.

A mis amigas Angie y Moni por su gran apoyo en la implementación de una nueva metodología, gracias por toda su dedicación, por ayudarme a leer el libro pese a que no era parte de su trabajo.

A mi amor, por siempre motivarme a que termine la tesis, por creer en mí cuando más lo necesitaba; tu confianza y aliento han sido esenciales para alcanzar esta meta, gracias por siempre apoyarme y no dejar que me derrumbe, gracias por tu amor incondicional.

Finalmente, a todos mis compañeros quienes de una u otra manera estuvieron ahí para brindarnos su apoyo a fin de conseguir este título profesional.

A todos ustedes, gracias por formar parte de este camino y por darme la fuerza para culminarlo.

Edú.

## **DEDICATORIA**

A mi esposa Jenny Angelica y mis bellos hijos Karolain Angelica, Ian Gabriel, Thais Valentina y mi princesita en el cielo, quienes desde el primer momento han sido mis pilares principales durante todo este proceso, quienes, con su cariño, amor, ternura y sus sonrisas fueron ese impulso en cada momento cuando creía no poder avanzar.

A mis padres, Teresa Narcisa y Juan Mecias, gracias por darme ese impulso en cada momento, por sus consejos, que han sido la guía en cada uno de los pasos de mi vida.

A mis hermanos, quienes de una u otra manera han sabido darme esas palabras de aliento para poder avanzar en este proceso.

A mis suegros, quienes de una u otra manera han sabido ayudarme para poder cumplir esta meta.

Juan.

## **AGREDECIMIENTO**

Primeramente, Dios y María Santísima por brindarme salud, vida y sabiduría para poder alcanzar esta meta soñada durante muchos años.

A mi amada esposa por su apoyo incondicional desde el primer instante que se abordó el tema de la Maestría; gracias por siempre estar junto a mí, por su compañía durante las largas veladas luego de la jornada laboral; en sí no hay palabra para poder agradecerle.

A mis bellos hijos, gracias por sus sonrisas de aliento, por saberme comprender y afrontar esta etapa un poquito apretada, pero siempre juntos, palabras de mis bellos hijos "Juntos como familia". Gracias, amores míos.

A mi tutora de Tesis Cintya, gracias infinitas primeramente por tomar esa decisión de apoyarme en esta tutoría, al mismo tiempo por sus jalones de orejas por sus consejos que han sido de gran ayuda para esforzarme y lograr un mejor resultado.

A mi equipo de trabajo, Mónica y Angy, gracias infinitas por su apoyo para iniciar con pie firme el proceso en miras a la implementación de una nueva metodología, gracias infinitas por ese gran apoyo.

A la Cooperativa Jardín Azuayo un agradecimiento infinito por el apoyo brindado, ya que su apoyo ha sido de gran importancia para poder cumplir este objetivo.

Y como no a Eduardo, mi compañero de tesis, gracias por ese apoyo, esfuerzo y dedicación para poder alcanzar esta meta.

Juan.

# TABLA DE CONTENIDO

---

Resumen .....	11
ABSTRACT.....	14
1. INTRODUCCIÓN .....	17
Sinopsis Capítulo 2.....	18
2. DETERMINACIÓN DEL PROBLEMA.....	19
2.1 DESCRIPCIÓN DEL PROBLEMA.....	19
2.2 OBJETIVOS .....	20
2.2.1 OBJETIVO GENERAL. ....	20
2.2.2 OBJETIVOS ESPECÍFICOS .....	20
Sinopsis Capítulo 3.....	21
3. MARCO TEÓRICO REFERENCIAL .....	23
3.1 DEVOPS.....	23
3.1.1 DEVOPS EN LAS FINANZAS .....	26
3.1.2 DEVOPS NIVEL EMPRESARIAL.....	27
3.1.3 LOS TRES PILARES DE DEVOPS.....	27
3.1.4 NECESIDAD .....	28
3.2 PRINCIPIOS ÁGIL .....	28
3.2.1 AGILIDAD EN EL ECOSISTEMA FINANCIERO .....	30
3.3 CONCEPTOS LEAN.....	30
3.3.2 LEAN INCEPTION .....	31
3.4 INTERACCIÓN ENTRE CONCEPTOS LEAN, AGILE Y DEVOPS .....	34
3.5 CASOS DE ÉXITO EN IMPLEMENTACIÓN DE PRACTICAS DEVOPS Y AGILE .....	34
3.6 DORA METRICS. ....	36
3.7 IMPACTO EN LA TOMA DE DECISIONES PARA EL DISEÑO Y DESARROLLO DEL SISTEMA.....	39
3.7.1 DEFINICIÓN DEL ALCANCE: LEAN INCEPTION Y EL MVP.....	40
3.7.2 PLANIFICACIÓN Y EJECUCIÓN: METODOLOGÍAS ÁGILES.....	40
3.7.3 IMPLEMENTACIÓN Y ENTREGA: DEVOPS .....	41
3.7.4. EVALUACIÓN DEL DESEMPEÑO: MÉTRICAS DORA.....	42
Sinopsis Capítulo 4.....	43
4. DESARROLLO DEL PROYECTO. ....	44

4.1	DISEÑO DE LA INVESTIGACIÓN.....	44
4.2	MODALIDAD DE LA INVESTIGACIÓN. ....	45
4.3	EJECUCIÓN DE LA METODOLOGÍA DE LA INVESTIGACIÓN.....	46
4.3.1	DEFINICIÓN DEL GRUPO DE TRABAJO.....	47
4.3.2	DEFINICIÓN DE REQUERIMIENTOS MEDIANTE LA AGENDA LEAN INCEPTION. .....	48
4.3.3	LANZAMIENTO (KICK-OFF).....	50
4.3.4	ESCRIBIENDO LA VISIÓN DEL PRODUCTO .....	50
4.3.5	EL PRODUCTO ES, NO ES / HACE, NO HACE. ....	51
4.3.6	DESCRIBIENDO LAS PERSONAS.....	52
4.3.7	LLUVIA DE IDEAS DE LAS FUNCIONALIDADES. ....	53
4.3.8	ENTENDIMIENTO TÉCNICO DE NEGOCIO Y DE EXPERIENCIA DE LAS FUNCIONALIDADES.....	54
4.3.9	MOSTRANDO LOS VIAJES DE USUARIOS. ....	56
4.3.10	CONSTRUCCIÓN SECUENCIADOR DE FUNCIONALIDADES. ....	67
4.3.11	CONSTRUYENDO UN CANVAS MVP.....	68
4.3.12	SHOW CASE DE LA LEAN INCEPTION.....	69
4.3.13	DEFINIR ALCANCE. ....	70
4.3.14	DEFINICIÓN DE MVP (PRODUCTO MINIMO VIABLE).....	70
4.4	DISEÑO DE PROTOTIPOS. ....	71
4.5	CONFIGURACIÓN DE ENTORNOS DE DESARROLLO.....	73
4.5.1	STACK TECNOLÓGICO. ....	73
4.5.2	ENFOQUE AGIL - LEAN INCEPTION. ....	75
4.5.3	METODOLOGÍA AGIL - SCRUM. ....	76
4.5.4	HERRAMIENTA GESTIÓN DE PROYECTOS CLICKUP. ....	77
4.5.5	STRUCTURIZR.....	78
4.5.6	POSTGRESQL.....	78
4.5.7	VISUAL STUDIO CODE. ....	79
4.5.8	ECLIPSE.....	80
4.5.9	REACTJS.....	80
4.5.10	SPRING BOOT.....	81
4.5.11	OPENAPI. ....	82
4.5.12	MAVEN.....	83
4.5.13	JAVA. ....	83
4.5.14	GITLAB. ....	84
4.5.15	DOCKER.....	85

4.5.16 DOCKER COMPOSE .....	85
4.5.17 GOOGLE CLOUD PLATAFORM. ....	85
4.5.18 GOOGLE CLOUD RUN.....	86
4.5.19 GOOGLE CLOUD SQL.....	86
4.6 CREACIÓN DE CUENTAS Y CONFIGURACION DE HERRAMIENTA DE CONTROL DE VERSIONES.....	86
4.7 SELECCIÓN DE IDE DE DESARROLLO BACK-END Y FRONT-END.....	89
4.8 CONFIGURACION DE AMBIENTES APEGADOS A LAS BUENAS PRACTICAS DE DEVOPS.....	90
4.9 SELECCIÓN DE HERRAMIENTAS DE AUTOMATIZACIÓN .....	93
4.10 CONSTRUCCIÓN DE MVP.....	95
4.11 DESARROLLO DEL MVP.....	97
4.12 DESPLIEGUE DEL MVP. ....	100
4.13 ANÁLISIS CI /CD .....	103
4.14 ANÁLISIS PIPELINE .....	106
4.15 FUNCIONALIDAD Y USO DE LAS REVISIONES.....	117
4.16 FEEDBACK CONTINUO. ....	121
Sinopsis Capítulo 5.....	122
5. RESULTADOS Y DISCUSIÓN .....	123
5.1 MEDICIÓN DE RESULTADOS .....	123
5.1.1 Dora Métrica antes de DevOps .....	124
5.1.2 Dora Métrica después de DevOps.....	126
5.2 Discusión de los objetivos planteados .....	130
5.2.1 Definir de manera temprana los requisitos .....	130
5.2.2 Diseño de prototipos del MVP .....	133
5.2.3 Seleccionar herramientas adecuadas.....	133
5.2.4 Construcción de un MVP .....	134
5.2.5 Medir la percepción de los usuarios.....	135
6. Conclusiones.....	143
6.1 Dificultades presentadas. ....	145
6.2 Posibles Mejoras.....	146
7. Bibliografía.....	148
8 Anexos. ....	152

APLICACIÓN DE  
PRINCIPIOS DEVOPS  
Y METODOLOGÍAS  
ÁGILES PARA EL  
DESARROLLO DEL  
MODULO DEL  
HISTORIAL  
FINANCIERO DE LA  
COOPERATIVA DE  
AHORRO Y CRÉDITO  
JARDÍN AZUAYO

AUTOR(ES):

EDUARDO GUILLERMO AGUILAR RIERA Y  
JUAN GABRIEL NAULA SANISACA

## RESUMEN

La Cooperativa de Ahorro y Crédito Jardín Azuayo enfrenta múltiples desafíos al momento de introducir nuevos productos y realizar los mantenimientos de sus sistemas. Estas dificultades pueden ser atribuibles, entre otras cosas, a la gestión manual de procesos y a la falta de una metodología eficiente para la recopilación de requisitos. Esta situación resulta en retrasos en los plazos de entrega y en problemas durante el desarrollo, tales como estimaciones imprecisas de tiempos y esfuerzos, así como errores en producción debido a pruebas limitadas y manuales. Además, el sistema actual, utilizado para gestionar el historial financiero de los socios no cumple con los requisitos necesarios para evaluar adecuadamente su capacidad de pago.

Con el fin de enfrentar estos desafíos, se sugiere crear un Producto Mínimo Viable (MVP) en el cual su proceso de construcción integre principios DevOps, así como enfoques ágiles como Scrum y metodologías como Lean Inception. Este enfoque incluirá la automatización de pruebas y despliegues, mejorando tanto la eficiencia operativa como la calidad del software. Los objetivos principales abarcan la definición temprana de requisitos, la creación de prototipos para obtener retroalimentación del usuario, y la selección de herramientas que simplifiquen adoptar un modelo de integración y despliegue continuo (CI/CD).

La metodología Lean Inception, permite identificar los requisitos esenciales y validar ideas de manera ágil, facilitando el desarrollo de un MVP que aglutine la información financiera de los socios de forma efectiva. Este proceso se complementa con la flexibilidad necesaria para adaptarse a cambios y asegurar la entrega constante de valor al sistema de la cooperativa.

Por otro lado, el enfoque ágil de Scrum se implementará para gestionar de forma eficaz el proceso de desarrollo, aplicando ceremonias esenciales como “Sprint Planning”, “Daily Scrum”, “Sprint Review” y “Sprint Retrospective”. Estas prácticas permitirán un seguimiento continuo y una mejora constante en el desarrollo.

Además, se abordará el despliegue de los diferentes artefactos utilizando Google Cloud Platform (GCP), que agiliza el proceso de despliegue y permite que los desarrolladores no inviertan esfuerzos en configurar la infraestructura, ya que esta es administrada por Google. De este modo, el equipo podrá concentrarse en el desarrollo del código y en garantizar la funcionalidad de la aplicación, satisfaciendo así las demandas y expectativas de los usuarios. Las diversas funcionalidades de GCP, como Cloud Run para el despliegue del backend y frontend, así como Cloud SQL para la base de datos, contribuirán significativamente a la eficiencia del proceso.

Finalmente, la utilización de GitLab como herramienta de control de versiones facilitará la administración de diferentes versiones del código y su automatización, gracias a sus robustas capacidades de integración y despliegue continuos (CI/CD). Esta plataforma no solo permite una gestión eficaz de versiones, sino que también optimiza la comunicación dentro del equipo mediante la semántica de Git, lo que favorece un flujo de trabajo más colaborativo.

GitLab ofrece funcionalidades adicionales que enriquecen la experiencia de desarrollo. La planificación de trabajo se ve optimizada a través de la creación de issues, que permiten notificar sobre diversos tipos de actividades, como errores, mejoras y nuevas funcionalidades. Esta característica asegura un mejor seguimiento de las tareas asignadas a los desarrolladores, facilitando la priorización y gestión del trabajo en curso.

Además, GitLab integra herramientas de seguimiento que permiten visualizar el estado de los proyectos en tiempo real, fomentando una mayor claridad en el proceso de desarrollo. La posibilidad de asignar tareas y establecer fechas límite proporciona claridad en las responsabilidades de cada integrante del equipo, contribuyendo a una entrega más puntual y eficiente.

En resumen, GitLab no solo mejora la gestión del código, sino que también establece un entorno de trabajo colaborativo donde las actividades pueden ser monitorizadas y evaluadas, asegurando así un desarrollo más ágil y efectivo.

---

**Palabras claves:**

DevOps Development (desarrollo) y Operations (operaciones), Lean, MVP Minimum Viable Product (Producto Mínimo Viable), Mockup (Prototipo desarrollado antes del trabajo), Pipeline (Serie de pasos automatizados que se utiliza para desarrollar, evaluar y lanzar software de forma eficaz), GCP (Google Cloud Platform).

## ABSTRACT

---

The Jardín Azuayo Savings and Credit Cooperative faces multiple challenges in introducing new products and maintaining its systems. These difficulties are mainly attributable to the manual handling of processes and the absence of an efficient methodology for requirement gathering. This situation results in delays in delivery deadlines and problems during development, including incorrect estimates of time and effort, along with production errors due to limited and manual testing. Moreover, the current system used to manage the financial history of the partners does not meet the necessary requirements to adequately assess their payment capacity.

To tackle these challenges, the creation of a Minimum Viable Product (MVP) is proposed, integrating DevOps principles, as well as agile approaches like Scrum and methodologies such as Lean Inception. This approach will include the automation of testing and deployments, improving both operational efficiency and software quality. The main objectives include the early definition of requirements, the creation of prototypes to obtain user feedback, and the selection of tools that facilitate the adoption of a continuous integration and continuous delivery (CI/CD) model.

Lean Inception methodology, which combines Design Thinking and Lean Startup, allows for identifying essential requirements and validating ideas quickly, facilitating the development of an MVP that effectively centralizes the financial information of the partners. This process is complemented by the necessary flexibility to adapt to changes and ensure the constant delivery of value to the cooperative's system.

On the other hand, the agile approach of Scrum will be implemented to effectively manage the software development process, applying essential ceremonies such as

“Sprint Planning”, “Daily Scrum”, “Sprint Review”, “Sprint Retrospective”. These practices will allow for continuous monitoring and constant improvement in development.

Additionally, the deployment of the different artifacts using Google Cloud Platform (GCP) will be addressed, which streamlines the deployment process and allows the development team not to invest efforts in infrastructure configuration, as it is managed by Google. This approach allows the team to concentrate on coding and ensuring the application's functionality, thus meeting the requirements and anticipations of the users. The various GCP services, such as Cloud Run for deploying the backend and frontend, as well as Cloud SQL for the database, will greatly enhance the productivity of the process.

Finally, the use of GitLab as a version control tool will facilitate the management of different code versions and their automation, because of its solid capabilities for continuous integration and continuous deployment (CI/CD). This platform not only allows for effective version management but also improves communication within the team through Git semantics, fostering a more collaborative workflow.

GitLab offers additional features that enrich the development experience. Work planning is optimized through the creation of issues, which allow for notifications about various types of activities, such as bugs, improvements, and new features. This feature ensures better tracking of tasks assigned to developers, facilitating the prioritization and management of ongoing work.

Additionally, GitLab integrates tracking tools that allow for real-time visualization of project status, promoting greater transparency in the development process. The ability to assign tasks and set deadlines provides clarity in the responsibilities of each team member, which contributes to more timely and efficient delivery.

In summary, GitLab not only improves code management but also establishes a collaborative work environment where activities can be monitored and evaluated, thus ensuring a more agile and effective development.

---

**Keywords:**

DevOps Development and Operations (operaciones), Lean, MVP Minimum Viable Product (Producto Mínimo Viable), Mockup (Prototipo desarrollado antes del trabajo), Pipeline (A set of automated procedures designed to build, test, and deploy software in an efficient and consistent manner), GCP (Google Cloud Platform).

---

# 1. INTRODUCCIÓN

---

La Cooperativa de Ahorro y Crédito Jardín Azuayo, fundada en 1996 desde sus inicios, ha estado comprometida con el desarrollo financiero de sus socios mediante la provisión de productos y servicios que fomenten el ahorro y el acceso al crédito.

En la era digital actual, la información financiera se ha convertido en un activo invaluable tanto para individuos como para organizaciones. La capacidad de acceder, analizar y comprender el historial financiero de manera rápida y precisa es fundamental para la toma de decisiones informadas y la gestión eficiente de recursos. Sin embargo, los enfoques tradicionales para el desarrollo y mantenimiento de sistemas de historial financiero a menudo se ven obstaculizados por procesos lentos, falta de colaboración entre equipos y dificultades para adaptarse a los cambios.

Ante este panorama, surge la necesidad de adoptar enfoques innovadores que permitan optimizar el desarrollo y la gestión de sistemas de historial financiero. Los Principios DevOps y las Metodologías ágiles se presentan como una solución prometedora para abordar estos desafíos. DevOps, al promover la colaboración y la automatización en todo el ciclo de vida del desarrollo de software, permite acelerar la entrega de valor y mejorar la calidad de los productos. Por su parte, las Metodologías ágiles, al priorizar la flexibilidad y la adaptación a los cambios, facilitan la creación de sistemas que se ajustan a las necesidades dinámicas del entorno financiero.

En este contexto, la Cooperativa de Ahorro y Crédito Jardín Azuayo (COAC) se enfrenta a desafíos significativos en su proceso de introducción de nuevos productos y en el mantenimiento de los existentes. La gestión manual de los procesos y la falta de una metodología eficiente para la recopilación de requisitos han resultado en tiempos prolongados de desarrollo y estimaciones inexactas de los plazos y esfuerzos necesarios para la implementación de nuevos proyectos. La falta de una estructura ágil y automatizada también afecta las fases de prueba,

---

limitando el número de pruebas realizadas y provocando la aparición de errores en producción. Adicionalmente, el sistema actual de obtención del historial financiero de los socios no cumple con las expectativas en cuanto a la evaluación de la capacidad de pago.

Para superar estos retos, este trabajo se centra en la implementación de un modelo de Integración y Entrega Continua (CI/CD) para el módulo del Historial Financiero de la COAC Jardín Azuayo. Proponemos la creación de un Producto Mínimamente Viable (MVP) que integre principios DevOps y metodologías ágiles, con el objetivo de agilizar los procesos de desarrollo y despliegue mediante la automatización. La automatización de los despliegues y las pruebas permitirá reducir el riesgo de errores humanos, garantizar la estabilidad del software y mejorar la eficiencia operativa. Además, se busca optimizar la comunicación entre los equipos de desarrollo, operaciones y demás departamentos, eliminando procesos manuales y acortando los tiempos de entrega en los entornos productivos.

Este enfoque transformará el ciclo de desarrollo del módulo del Historial Financiero, lo que resultará en una entrega continua de valor y una retroalimentación temprana y constante del usuario. A través de la automatización de las tareas y la mejora de la calidad y eficiencia, se busca establecer una base sólida para futuros desarrollos dentro de la cooperativa. Esperamos que los resultados de esta investigación contribuyan a la mejora de la práctica del desarrollo de sistemas de historial financiero y sienten las bases para futuras investigaciones en este campo, específicamente en el contexto de la Cooperativa de Ahorro y Crédito Jardín Azuayo.

## SINOPSIS CAPÍTULO 2

Este capítulo describe los principales problemas que enfrenta la Cooperativa de Ahorro y Crédito Jardín Azuayo en la gestión de su historial financiero. Se detallan las dificultades asociadas a los procesos manuales, la falta de metodologías eficientes en la recopilación de requisitos y la deficiente estimación de tiempos en

---

el desarrollo de nuevas aplicaciones. Asimismo, se abordan la falta de automatización y el impacto de estos factores en la calidad del software. Finalmente, se plantea la aplicación de metodologías ágiles para la captura de requisitos, así como la implementación de un MVP con enfoque DevOps como solución para optimizar el desarrollo, pruebas y despliegue de software.

## 2. DETERMINACIÓN DEL PROBLEMA

### 2.1 DESCRIPCIÓN DEL PROBLEMA

La Cooperativa de Ahorro y Crédito Jardín Azuayo enfrenta actualmente desafíos significativos en la introducción de nuevos productos al mercado y en el mantenimiento de los ya existentes. Estos problemas se originan principalmente en la gestión manual de la mayoría de los procesos y en la ausencia de una metodología eficiente para la recopilación de requisitos, lo que provoca que estos procedimientos se prolonguen durante varias semanas.

Además, el desarrollo de nuevas aplicaciones se complica debido a estimaciones inexactas de los tiempos y esfuerzos necesarios para su implementación, lo que resulta en la fijación de plazos poco realistas para la finalización de los proyectos. Durante el proceso de desarrollo, es común que los usuarios soliciten funcionalidades adicionales no previstas inicialmente, incrementando el tiempo y esfuerzo requeridos y generando retrasos en las fechas comprometidas.

En las fases de pruebas, se presentan dificultades para preparar los entornos y ejecutar pruebas funcionales, ya que las instalaciones no están automatizadas y las pruebas se realizan manualmente por uno o varios usuarios. Esto limita el número de pruebas ejecutadas, las cuales no abarcan la totalidad del sistema, resultando en la aparición de errores en producción que no fueron detectados durante la fase de pruebas.

---

Actualmente, la cooperativa dispone de un sistema obsoleto para obtener el historial financiero de sus socios, el cual no cumple con las expectativas al momento de obtener información sobre la capacidad de pago de los mismos.

Para abordar estos desafíos, se propone la creación de un Producto Mínimamente Viable (MVP) que incorpore principios DevOps y metodologías ágiles. El objetivo es promover la entrega continua y la automatización, agilizando el ciclo de desarrollo y despliegue. La automatización de los despliegues y pruebas permitirá reducir errores humanos, garantizar la estabilidad del software y mejorar la eficiencia operativa. Además, esta metodología fomentará una comunicación más efectiva entre los equipos de desarrollo, operaciones y otros departamentos, reduciendo costos al eliminar procesos manuales y acortando los tiempos de entrega en entornos productivos.

## 2.2 OBJETIVOS

### 2.2.1 OBJETIVO GENERAL.

Implementación de un modelo CI/CD alineado a la propuesta de valor para el Módulo del Historial Financiero de la COAC Jardín Azuayo.

### 2.2.2 OBJETIVOS ESPECÍFICOS

Definir de manera temprana los requisitos del módulo del Historial Financiero, mediante la utilización de Lean Inception para definir el mínimo producto viable (MVP) que se implementará en la COAC Jardín Azuayo.

Diseño de prototipos del MVP del módulo del Historial Financiero, en base al Product Backlog, mediante Mockups, para obtener una retroalimentación del usuario con respecto al diseño planteado.

Seleccionar las herramientas adecuadas para la implementación de CI/CD, mediante la creación de un pipeline, repositorios de versionamiento, configuración de ambientes de desarrollo para automatizar las tareas y lograr un impacto significativo en la eficiencia, la calidad y la agilidad del proceso de desarrollo.

---

Construcción de un MVP del módulo del Historial Financiero, mediante la automatización de procesos de: desarrollo, pruebas, seguridades, para garantizar una automatización efectiva y una entrega y despliegue continuo de software de calidad que puedan servir de base para los futuros desarrollos de la cooperativa.

Medir la percepción y satisfacción de los usuarios, a través de evaluaciones de satisfacción, comparación de tiempos de entrega de valor anteriores versus los tiempos actuales aplicando metodologías ágiles, para evaluar si los usuarios están complacidos con el MVP del Historial Financiero entregado y poder obtener una retroalimentación para mejorar los siguientes sprints.

## SINOPSIS CAPÍTULO 3

En el capítulo anterior, se pudo identificar los principales desafíos que enfrenta la Cooperativa de Ahorro y Crédito Jardín Azuayo en el proceso de desarrollo de Software para este caso puntual el Módulo del Historial Financiero, tales como procesos manuales ineficientes y una deficiente estimación de tiempos en el desarrollo de nuevas aplicaciones o mejoras a los existentes. Para abordar esta problemática se propuso la aplicación de metodologías ágiles y la implementación de un Producto Mínimo Viable (MVP) con enfoque DevOps.

En este contexto, el capítulo "MARCO TEÓRICO REFERENCIAL" profundiza en los fundamentos conceptuales y metodológicos que sustenta dicha propuesta. Se examina la evolución histórica de DevOps, destacando su origen en la necesidad de integrar los procesos de desarrollo y operaciones para mejorar la eficiencia y calidad en la entrega de software. También se exploran los conceptos de la filosofía Lean, enfocándose en la eliminación de desperdicios y la optimización de procesos para maximizar el valor entregado al cliente.

En este marco, se introduce la metodología Lean Inception, una práctica que combina Design Thinking y Lean Startup para definir el Producto Mínimo Viable (MVP) a través de talleres colaborativos. Esta metodología es especialmente útil para alinear al equipo en la visión del producto y establecer una dirección común

---

desde el inicio del proyecto. Lean Inception se compone de una serie de actividades estructuradas que facilitan la comprensión de las características clave del producto, la identificación de los posibles clientes y la construcción de un lienzo que formula las características del MVP. Al integrar Lean Inception en la adopción de prácticas DevOps, la organización puede asegurar que el desarrollo de soluciones tecnológicas esté alineado con las necesidades reales de los usuarios, promoviendo una entrega continua y eficiente de valor.

Asimismo, se destaca la necesidad de una transformación cultural en la organización, orientada hacia la adopción de prácticas ágiles y de automatización que permitan una entrega continua y eficiente de soluciones tecnológicas. Esta transformación no solo mejora la experiencia de los socios, sino que también mantiene la competitividad en el sector financiero al asegurar que los productos y servicios ofrecidos estén alineados con las necesidades y expectativas de los clientes.

Además, se profundiza en la implementación de la entrega continua (CD) utilizando Google Cloud Platform (GCP). GCP ofrece herramientas como Cloud Shell, Cloud Build y Cloud Deploy que facilitan la automatización del ciclo de vida del software, desde la construcción hasta el despliegue en producción. Estas herramientas permiten gestionar versiones de manera eficiente, asegurando la disponibilidad inmediata de los servicios y la capacidad de revertir cambios en caso de errores. La integración de estas herramientas en el proceso de DevOps contribuye a una entrega más rápida y confiable de las soluciones tecnológicas.

Finalmente, como todo trabajo es necesario medir el resultado de la aplicación de estos principios para lo cual se enfatiza en la importancia de evaluar la efectividad de DevOps mediante el uso de métricas clave. Para lo cual se analizan las métricas DORA (DevOps Research and Assessment), que proporcionan indicadores esenciales para medir el rendimiento de los equipos de desarrollo y operaciones. Estas métricas incluyen:

---

Frecuencia de despliegue (Deployment Frequency): Mide cuántas veces se despliega código en producción en un período determinado.

Tiempo medio de entrega de cambios (Mean Lead Time for Changes): Indica el tiempo promedio desde que se realiza un cambio en el código hasta que se despliega en producción.

Tasa de fallos de cambios (Change Failure Rate): Representa el porcentaje de cambios que resultan en fallos en producción que requieren una corrección inmediata.

Tiempo medio de recuperación (Mean Time to Recovery - MTTR): Mide el tiempo promedio que se tarda en restaurar el servicio después de una interrupción en producción.

Estas métricas permiten a las organizaciones evaluar la eficiencia y estabilidad de sus procesos de entrega de software, facilitando la identificación de áreas de mejora y la implementación de prácticas que optimicen el rendimiento general.

## 3. MARCO TEÓRICO REFERENCIAL

---

El marco referencial de una tesis de grado es una sección fundamental que proporciona el contexto teórico y conceptual necesario para comprender el problema de investigación. Su objetivo principal es situar el estudio dentro del conjunto de conocimientos existentes, identificando teorías, conceptos clave, antecedentes y, en caso necesario, el marco legal relacionado con el tema.

### 3.1 DEVOPS.

A principios de la década de 2000, el desarrollo de aplicaciones y la gestión de operaciones eran dos mundos distintos. El desarrollo se consideraba estratégico, mientras que las operaciones se percibían como tácticas, a menudo relegadas o subcontratadas. Los desarrolladores se centraban en la creación de código,

---

---

mientras que las operaciones se enfocaban en mantener el servicio disponible para los clientes (Kim, Humble, Debois, & Willis, 2016) Esta dinámica generaba problemas como incumplimientos de plazos, versiones de software extensas y falta de coordinación, lo que retrasaba la entrega de valor hasta la finalización del proyecto. Además, los errores se detectaban tardíamente, lo que aumentaba los costos de corrección de la deuda técnica (Kim, Humble, Debois, & Willis, 2016)

El despliegue de cada nueva versión de software era un desafío, ya que podía causar interrupciones en el servicio debido a procesos manuales, lo que a menudo resultaba en sanciones por el impacto negativo en el cliente (Kim, Humble, Debois, & Willis, 2016). Para abordar estos problemas, se adoptaron conceptos de la metodología Ágil, implementando procesos de desarrollo por Sprint. Esto permitió realizar entregas y mejoras continuas en pequeñas iteraciones, lo que facilitó una mejor comprensión de las necesidades de los usuarios (Debois, 2008).

Con cada Sprint, surgía la necesidad de desplegar la aplicación, lo que requería una nueva fase de configuración del servidor por parte del equipo de Operaciones de TI. Este proceso, caracterizado por pruebas iterativas, brindaba una oportunidad constante para mejorar la configuración y optimizar el entorno (Debois, 2008).

A partir de estas experiencias, evolucionó un conjunto de prácticas que hoy conocemos como DevOps. El término fue acuñado por Debois en la Agile Conference de 2008 (Debois, 2008). DevOps promueve la colaboración estrecha entre todos los actores involucrados en el ciclo de vida del software, desde propietarios de productos y arquitectos hasta desarrolladores, equipos de QA, operaciones de TI y profesionales de seguridad (Kim et al., 2016). El objetivo es alcanzar metas compartidas, donde el trabajo en equipo contribuya a un ciclo de desarrollo más eficiente. DevOps fomenta un cambio cultural y una comunicación efectiva, lo que permite comprender mejor las necesidades del cliente y entregar software de alta calidad de manera más rápida, gracias a la automatización de los procesos de integración y despliegue (Debois, 2008).

---

En esencia, DevOps es una filosofía y cultura de ingeniería de software que impulsa a equipos multifuncionales a crear, probar y desplegar software de forma más rápida y confiable mediante la automatización (Macarthy, 2021).

Sus principios clave, como la integración y entrega continua, optimizan el ciclo de vida del software, garantizando estabilidad, eficiencia, confiabilidad, disponibilidad y seguridad (Nasreen Azad & Hyrynsalmi, 2023).

DevOps fomenta la colaboración para agilizar la respuesta a cambios, requerimientos y actualizaciones (Nasreen Azad & Hyrynsalmi, 2023).

Este enfoque organizacional moderno promueve una relación fluida entre desarrolladores y operaciones, utilizando metodologías ágiles para lanzamientos frecuentes que aseguren sostenibilidad y rentabilidad. El éxito de DevOps radica en un cambio cultural que fomente la satisfacción y motivación de los empleados (Baertschi, 2023).

DevOps es un movimiento cultural y un conjunto de buenas prácticas que busca reducir la brecha entre desarrollo y operaciones, causada por su separación estructural. Su objetivo es acelerar el desarrollo de software, permitiendo lanzamientos más frecuentes y una respuesta ágil a las demandas de los clientes, sin comprometer la calidad del software. Esto se logra mediante una comunicación estrecha entre desarrolladores y operaciones (Su, 2023).

DevOps es una solución a las restricciones de tiempo en el desarrollo de software, que busca aumentar la frecuencia y velocidad de entrega mediante procesos automatizados. Sin embargo, DevOps no se trata solo de herramientas, sino de una transformación cultural que prioriza la comunicación, la automatización, la medición y el intercambio de información (Amitkumar V. Jha, 2023).

En la ingeniería de software moderna, la implementación de DevOps y metodologías ágiles se ha vuelto esencial. (Akbar, 2024) enfatiza la importancia de estas prácticas, destacando que los canales de CI/CD son fundamentales para mejorar la productividad, la calidad y los tiempos de respuesta. La integración

---

continua de código (CI) permite la detección temprana de errores, facilitando ajustes oportunos y reduciendo el tiempo de depuración. La automatización, otra ventaja clave, libera a los desarrolladores para que se centren en tareas de desarrollo, evitando así desperdicios. La entrega continua (CD) extiende el concepto de CI al automatizar el proceso de implementación en entornos de producción una vez superadas las pruebas necesarias. CI y CD constituyen la base de DevOps, permitiendo a los equipos entregar software de alta calidad de manera más rápida y confiable. La automatización del proceso de desarrollo e implementación facilita iteraciones rápidas y la mejora continua de los productos de software.

### 3.1.1 DEVOPS EN LAS FINANZAS

La aplicación de DevOps en la industria financiera es esencial en un entorno altamente competitivo donde el crecimiento continuo y el cumplimiento de objetivos a corto plazo son cruciales (Bird, 2017).

Los servicios financieros recurren a DevOps para introducir nuevos productos y servicios más rápido, pero deben cumplir con estrictos requisitos de actividad y rendimiento (Bird, 2017). Aunque la adopción de DevOps en este sector implica desafíos y costos elevados, los beneficios son innegables, al igual que los riesgos de no entregar valor a los clientes con la suficiente rapidez y perderlos frente a competidores, especialmente empresas disruptivas en línea impulsadas por DevOps (Bird, 2017).

Los principios ágiles (priorizar el software sobre la documentación, la entrega frecuente, la colaboración cara a cara y un enfoque en la excelencia técnica y la automatización) forman la base de DevOps. La entrega continua, el marco de control para DevOps, se basa en una práctica fundamental del desarrollo ágil: la integración continua, donde los desarrolladores se aseguran de que el código crece y se ejecuta correctamente cada vez que se registra un cambio. La entrega continua lleva esto al siguiente nivel (Bird, 2017). Busca aprovisionar o configurar entornos de pruebas que coincidan en lo posible con la producción de forma automática; empaquetar el código e implementarlo en entornos de pruebas automáticamente;

---

ejecutar pruebas de estrés, aceptación, rendimiento y seguridad para lograr que el sistema esté siempre listo para implementarse en producción (Bird, 2017).

DevOps implica la distribución conjunta de la supervisión del sistema entre los equipos de desarrollo y operaciones. Esto implica la exposición de métricas de producción, registros de excepciones y la emisión de alertas a los desarrolladores (Bird, 2017).

### 3.1.2 DEVOPS NIVEL EMPRESARIAL

El conjunto de buenas prácticas de DevOps ha demostrado su valor en numerosas implementaciones, especialmente en grandes empresas de Internet como Google, Netflix, Amazon y Etsy. Estas empresas han transformado su arquitectura, prácticas, técnicas y, lo más importante, su cultura de trabajo colaborativo, lo que les ha permitido obtener resultados sorprendentes y aumentar su competitividad.

### 3.1.3 LOS TRES PILARES DE DEVOPS

DevOps se basa en tres pilares fundamentales:

**Cultura:** Fomenta la cooperación, comunicación, transparencia, responsabilidad compartida, confianza y comprensión entre los equipos

**La automatización:** Permite acelerar el proceso de desarrollo y entrega de software al automatizar tareas repetitivas, pruebas y cualquier otra implementación sistemática.

**Integración continua:** Permite liberar software de manera rápida y confiable. La Integración Continua (CI) implica la integración regular y automatizada de nuevo código en un repositorio compartido, seguida de pruebas automáticas para asegurar la ausencia de errores.

**Entrega Continua (CD)** lleva este concepto un paso adelante al automatizar el despliegue del software a ambientes productivos o pruebas, garantizando que el software esté siempre en condiciones de ser desplegado.

---

Estos tres pilares se combinan para crear un entorno ágil, colaborativo y eficiente que facilita que las organizaciones desarrollen, prueben y entreguen software con mayor rapidez como se puede ver en la imagen 1.

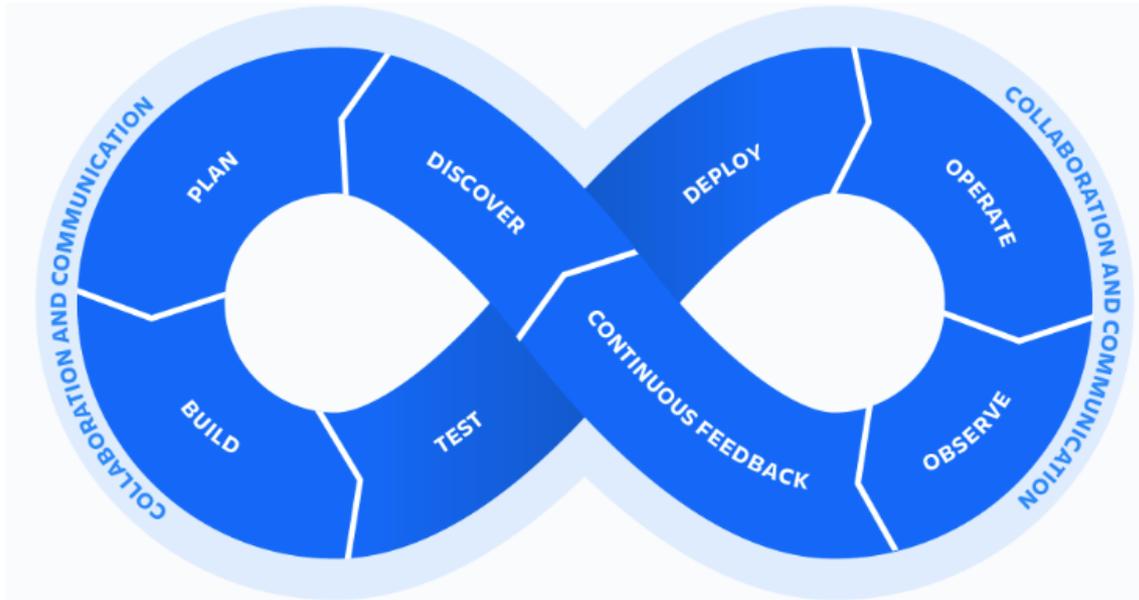


Imagen 1 Pilares de DevOps (Atlassian, 2024)

### 3.1.4 NECESIDAD

DevOps es actualmente un componente esencial y necesario para el proceso de desarrollar software, no solo para la Cooperativa, sino para todas las empresas tecnológicas. Permite reducir los tiempos de respuesta y lograr una comprensión efectiva de las necesidades reales en la entrega de productos que generen valor al socio. Estos productos se integran y entregan de manera continua con el fin de brindar una mejor experiencia a los socios.

## 3.2 PRINCIPIOS ÁGIL

En el pasado, la planificación de proyectos de desarrollo solía llevarse a cabo durante varios años antes de que comenzara la fase de programación, lo que a menudo resultaba en desastres costosos. Era común tener que reconstruir por

---

completo el software después de haber invertido sumas significativas de dinero. Para abordar estos desafíos, en el año 2001, un grupo de 17 especialistas en desarrollo de software se reunió para discutir las técnicas y procesos de desarrollo. En esta reunión, se introdujo el término "Metodologías ágiles" para referirse a los nuevos enfoques que estaban surgiendo como alternativas a los métodos convencionales, los cuales se caracterizaban por ser excesivamente rígidos y depender en gran medida de una planificación exhaustiva antes de iniciar el proceso.

Con este fin, firmaron el Manifiesto Ágil, que se fundamenta en los siguientes principios:

Ejecución de software en lugar de depender excesivamente de documentación extensiva.

Priorización de las personas y sus interacciones sobre herramientas y procesos.

Adaptación y respuesta ágil al cambio en lugar de adherirse estrictamente a un plan.

Colaboración activa con el cliente en lugar de depender principalmente de acuerdos contractuales.

A partir de estos principios se han desarrollado una variedad de marcos de trabajo y metodologías, siendo Scrum una de las principales y más utilizadas. Según algunos autores, Scrum es un enfoque ágil que brinda apoyo a individuos, grupos y entidades, permitiéndoles crear valor mediante la implementación de soluciones flexibles para desafíos complicados (Schwaber & Sutherland, 2020).

Además, según Schwaber & Sutherland, Scrum se fundamenta en el empirismo y la filosofía Lean. El empirismo defiende que el conocimiento se origina en la experiencia y las decisiones fundamentadas en la observación, mientras que el enfoque Lean se orienta a reducir el desperdicio y concentrarse en lo esencial (Schwaber & Sutherland, 2020).

---

### 3.2.1 AGILIDAD EN EL ECOSISTEMA FINANCIERO

En la situación actual del sector financiero, caracterizado por un rápido avance tecnológico, cambios normativos y una creciente demanda de los clientes, la agilidad ha surgido como un factor crítico para prosperar en un entorno empresarial dinámico y cambiante. Las instituciones financieras están adoptando enfoques ágiles para mejorar su capacidad de respuesta ante estos desafíos. Esto implica la habilidad de adaptarse rápidamente a cambios, implementar eficientemente nuevas tecnologías y ajustar estrategias según las demandas del mercado.

Desde la incorporación de enfoques ágiles en el desarrollo hasta la implementación de estrategias ágiles en la administración de proyectos y procesos, la industria financiera busca no solo mantenerse al día con los cambios, sino también liderar la innovación y ofrecer servicios financieros más flexibles y orientados al cliente. La agilidad se ha vuelto un elemento esencial para la supervivencia y el éxito en un sector que exige una adaptabilidad constante.

## 3.3 CONCEPTOS LEAN

La metodología Lean tiene sus raíces en Japón, especialmente en Toyota (Womack, J. P., & Jones, D. T. (2003). *Lean thinking: Banish waste and create wealth in your corporation*. Simon & Schuster), y se fundamenta en su sistema de producción Toyota Production System (TPS). Se trata de una filosofía que, mediante herramientas y técnicas, busca transformar la cultura organizacional para adoptar prácticas que mejoren procesos u operaciones. Su objetivo es reducir desperdicios, optimizar flujos y priorizar las necesidades del cliente, todo ello en un marco de entregas más ágiles, sin comprometer la calidad.

El enfoque Lean ha ganado prominencia como la estrategia preferida en muchas industrias de manufactura, y sus principios principales se resumen en seis conceptos esenciales:

---

Definición de valor desde el punto de vista del cliente: Un producto o servicio tiene valor cuando responde directamente a una necesidad específica del cliente; todo lo demás carece de valor agregado.

Identificación del flujo de valor: Esto abarca todas las actividades de los diversos departamentos involucrados en la entrega conjunta de un producto o servicio. Representa el proceso completo que añade valor al cliente, desde la concepción hasta la producción.

Creación de flujo al eliminar desperdicios: Al analizar el flujo de valor, se descubre que solo un pequeño porcentaje de actividades agrega valor en la industria de servicios, pero esto puede aumentar significativamente. Eliminar desperdicios asegura un flujo ininterrumpido hacia el cliente, enfocándose en el producto, flexibilizando prácticas y adoptando nuevas herramientas.

Atender la demanda del cliente: Entender y ajustar el proceso para satisfacer la demanda del cliente, produciendo lo que desea en el momento preciso para evitar la generación de productos no deseados.

Mejora continua en busca de la excelencia: Comenzar con una reorganización completa de los pasos del proceso y conectarlos progresivamente para identificar y eliminar desperdicios, aspirando a una perfección teórica donde cada paso agregue valor al cliente.

Compromiso y empoderamiento de los empleados: El involucramiento y formación de los trabajadores son cruciales. La capacitación adecuada es esencial para implementar Lean y se debe promover una cultura que incluya métricas enfocadas en el valor. Medir el rendimiento de acuerdo con esta estrategia es vital para evitar prácticas no alineadas con Lean.

### 3.3.2 LEAN INCEPTION

Una "inception" es una actividad que se lleva a cabo al comienzo de un proyecto para reunir a las personas involucradas y establecer una dirección común y un estilo

---

de trabajo continuo. La "lean inception" es una forma enfocada de inception, que se puede realizar en una sola semana. Durante este tiempo, comprendemos las características clave, los posibles clientes del producto, y construimos un lienzo para formular las características de un Producto Mínimo Viable.

Una "inception" plantea un método colaborativo de exploración y clarificación, en el cual los participantes colaboran en una serie de actividades para comprender las opciones y desarrollar el Producto Mínimo Viable (Caroli, 2018).

Lean Inception es una metodología que sirve como guía, compuesta por una serie de actividades colaborativas y dinámicas orientadas a crear un lienzo del Mínimo Producto Viable (MVP), una visualización del avance del producto "lean" y su estrategia de desarrollo. Comienza con la actividad de "Definir la visión del producto" y concluye con "Construir el lienzo MVP". En lugar de planear un producto final con todas las funcionalidades posibles, se centra en un producto inicial que pueda lanzarse rápidamente y del que se pueda obtener aprendizaje: el Producto Mínimo Viable (Caroli, 2018).

MVP (Producto Mínimo Viable): en el ciclo "construir, medir y aprender", el MVP se presenta como un componente esencial, como se ilustra en el gráfico a continuación imagen 2, donde el MVP se visualiza como el artefacto a desarrollar (Caroli, Lean Inception Creando Conversaciones hacia un producto exitoso., 2018).

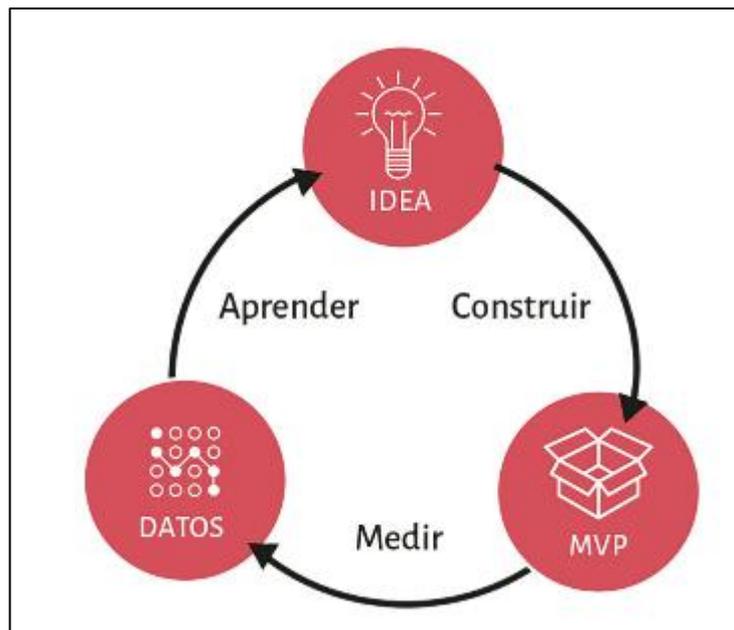


Imagen 2 Producto Mínimo Viable (Caroli, *Lean inception: creando conversaciones hacia un producto exitoso*, 2019)

Cuando el producto mínimo viable está desarrollado, se somete a pruebas para recopilar datos que permitan evaluar su uso y, en consecuencia, obtener el aprendizaje esperado.

Recibe el nombre de Lean Inception por dos motivos: en primer lugar, la duración de la inception es más breve, excluyendo aspectos no relacionados con el producto, como arquitectura o gestión de proyectos, lo que la hace más ágil. En segundo lugar, el resultado principal de la inception es la claridad sobre el Producto Mínimo Viable (MVP), un elemento clave en la filosofía Lean Startup el cual es un enfoque para crear nuevos servicios y productos enfocados en la eficiencia y la adaptación rápida (Caroli, *Lean Inception Creando Conversaciones hacia un producto exitoso.*, 2018).

Finalmente se puede considerar que Lean Inception es una serie de técnicas que nos ayudarán a construir un MVP (producto mínimo viable) el cual nos otorgará valor de una manera rápida, dichos MVP van a ir creciendo de manera iterativa e incremental, el principal objetivo del MVP es poder determinar si vale o no la pena seguir construyendo el producto.

---

## 3.4 INTERACCIÓN ENTRE CONCEPTOS LEAN, AGILE Y DEVOPS

Las tres estrategias comparten un conjunto común de metas, que abarcan desde acelerar el tiempo de introducción al mercado y el tiempo para lograr valor, hasta conseguir mayor eficiencia, calidad mejorada, mayor satisfacción del cliente, ahorro de costos, retroalimentación en tiempo real y mejora continua a través de la entrega ininterrumpida. La conexión entre las tres también puede ser definida mediante un conjunto de definiciones en constante evolución, indicando que las metodologías mismas están en un estado de constante perfeccionamiento. Algunos ven a Agile como una filosofía, dado que se fundamenta en diversas metodologías como Scrum, programación extrema, Kanban (utiliza tableros visuales para representar el flujo de trabajo), programación en pares, entre otras. La constante que une a Lean, Agile y DevOps es el concepto de "valor". Sea que este valor se alcance mediante mejoras más frecuentes en el software, logrando una disminución en el tiempo de llegada al mercado, así como mayores niveles de satisfacción del cliente o una, la meta última es proporcionar un valor superior a los clientes.

## 3.5 CASOS DE ÉXITO EN IMPLEMENTACIÓN DE PRACTICAS DEVOPS Y AGILE

Del análisis de la adopción de DevOps en la industria bancaria Suiza, basado en el método obtenidos en base a los elementos de Gutzwiller, el autor realiza la evaluación del método aplicado, arrojando resultados abrumadores que demuestran el éxito generado de la aplicación de DevOps (Baertschi, 2023) Como se describe a continuación:

Al año de aplicado la metodología y en pleno proceso de transición de DevOps en la industria bancaria suiza, en el equipo de DevOps existentes realizaron las visitas diarias y la socialización de la metodología aplicada de lo cual realizaron cuestionarios y una evaluación a los 53 empleados con diferentes roles como Producto Owner, Scrum Máster e ingenieros DevOps.

---

Las temáticas de la evaluación fueron:

Comprensión y utilidad del método utilizado

Evaluación del método DevOps introducido en la entidad financiera

Evaluación de si el método creado habría agregado valor a la implementación de DevOps en la institución financiera

Los resultados de aplicar la encuesta demostraron que:

El 92,45 % de los encuestados considera comprensible o muy comprensible el método aplicado.

El 84,91 % dio respuestas correctas a las preguntas de control individual.

Una quinta parte del equipo encuestado considera que hubiese sido valioso presentar el método al inicio de la transformación cultural.

El 79,25 % considera la importancia de la automatización, y el 64,15 % la califica como muy grande o grande.

El 32,08 % considera que el método habría creado valor solo si se hubiese revisado por completo, y el 47,17 % considera que el método existente habría tenido que ser revisado para que genere valor a la empresa.

Como resultado de este análisis, se puede observar que el cambio de cultura dentro del equipo es vital para la adopción de DevOps. Este cambio ayuda a que el equipo se empodere y comprenda el valor del método aplicado (Baertschi, 2023).

Por otro lado, del estudio realizado por (Su, 2023), podemos ver el éxito generado en la adopción de DevOps en un equipo de desarrollo dentro de una Organización financiera tomando como base de una investigación cualitativa en base a tres métodos de investigación como encuestas, retrospectivas y entrevistas arrojando resultados abrumadores como se detalla a continuación.

---

De cinco encuestas repetitivas distribuidas durante cuatro meses a 47 participantes del equipo de desarrollo, recibimos 40 respuestas completas de 235 formularios enviados con una tasa de respuesta del 17%. El 30% de los encuestados eran directores de proyectos que lideraban el desarrollo de aplicaciones y el 70% eran desarrolladores. Los resultados mostraron que 27 de 40 encuestados activaron el proceso de CI/CD al menos una vez en los últimos 14 días, y el 62% de ellos lo activó más de tres veces. La Figura 1 muestra las veces que se activaron si los participantes han utilizado la canalización de CI/CD en los últimos 14 días, correspondiente a la cantidad de horas reportadas como ahorradas por los encuestados (Su, 2023).

### 3.6 DORA METRICS.

En el desarrollo de software aplicando principios ágiles, resulta fundamental el uso de métricas para realizar un seguimiento y evaluación de la implementación de prácticas DevOps. En este contexto, las métricas DORA, desarrolladas por Nicole Forsgren, Gene Kim y Jez Humble (Brennan Wilkes, 2023) se destacan por su capacidad para medir y comparar el rendimiento de DevOps en diversas organizaciones. Estas métricas permiten evaluar el desempeño de dos maneras: en primer lugar, miden la eficiencia operativa a través de la Frecuencia de Despliegue y el Tiempo de Espera para el Cambio; en segundo lugar, permiten valorar la estabilidad del producto mediante la Tasa de Fallos de Cambios y el Tiempo de Recuperación. Esta evaluación integral facilita a las organizaciones optimizar sus procesos y asegurar la calidad y confiabilidad en la entrega continua de software.

Para aplicar correctamente la metodología, es esencial detallar cada métrica, entre las cuales se encuentra la Frecuencia de Despliegue (Deployment Frequency). Esta es una de las métricas clave que evalúa el rendimiento de los equipos en términos de velocidad de entrega. La Frecuencia de Despliegue mide con qué frecuencia los equipos despliegan cambios en un ambiente de producción, siendo un indicador crucial de la agilidad y capacidad de respuesta ante cambios y nuevas funcionalidades requeridas. Para este cálculo, se establecen valores de referencia en la frecuencia de despliegue, tales como:

---

Elite: Varias veces al día (idealmente, más de un despliegue diario).

Alta: Entre una vez al día y una vez por semana.

Media: Entre una vez por semana y una vez al mes.

Baja: Menos de una vez al mes.

Estas categorías ayudan a clasificar el rendimiento de los equipos y contribuyen a la toma de decisiones enfocadas en optimizar la frecuencia y eficacia de los despliegues en función de los objetivos organizacionales.

Por otro lado, la métrica Tiempo de Espera para el Cambio (Lead Time for Change) permite evaluar el tiempo que transcurre desde el versionamiento de un cambio hasta su implementación en producción. Esta métrica refleja la agilidad y capacidad del equipo para responder de manera eficiente a los cambios requeridos. Al igual que con la Frecuencia de Despliegue, se establecen valores de referencia que facilitan su interpretación y comparación. Los niveles de rendimiento considerados en esta métrica son los siguientes:

Elite: Menos de una hora.

Alto Desempeño: Entre una hora y un día.

Desempeño Medio: Entre uno y siete días.

Bajo Desempeño: Más de una semana.

Estos niveles permiten a las organizaciones identificar áreas de mejora y optimizar los procesos de desarrollo y despliegue, asegurando así una respuesta ágil y eficaz a los cambios y demandas del mercado.

En cuanto a la estabilidad del producto, se utiliza la métrica denominada Tasa de Fallo de Cambio (Change Failure Rate), que permite medir el porcentaje de cambios que resultan en errores en producción. Esta métrica proporciona una visión de la calidad y estabilidad de las entregas en producción. Cabe destacar que la Tasa de

---

Fallo de Cambio es una métrica fundamental en DORA, ya que evalúa la frecuencia de fallos en los cambios implementados, lo que ayuda a analizar la consistencia y confiabilidad de los despliegues.

El valor referencial para esta tasa suele expresarse en porcentaje. En general, una baja tasa de fallos (idealmente, inferior al 15%) indica que la mayoría de los cambios son estables y no requieren corrección. Según los informes de DORA, los equipos de alto rendimiento suelen mantener una Tasa de Fallo de Cambio por debajo del 15%, mientras que los equipos de rendimiento menor pueden superar el 30%.

Para el cálculo de este valor se aplica al formula:

$$\text{Change Failure Rate} = (\text{Número de Fallos} / \text{Número total de implementaciones}) * 100$$

Tiempo de Recuperacion (Mean Time to Recovery).

Otra métrica importante es el Tiempo de Recuperación o Time to Restore Services, que mide el tiempo promedio necesario para resolver incidentes en producción tras un fallo. Esta métrica evalúa la capacidad del equipo para restaurar rápidamente el servicio, lo que resulta fundamental para comprender la rapidez de respuesta y la resiliencia del sistema frente a problemas imprevistos. Su objetivo es minimizar el tiempo de interrupción para reducir el impacto en los usuarios finales.

Valores de Referencia para el Tiempo de Recuperación

En el contexto de DORA, los equipos de alto rendimiento suelen restaurar servicios en menos de una hora tras un fallo. Los valores de referencia de DORA son los siguientes:

Alto rendimiento: Menos de 1 hora.

Medio rendimiento: Entre unas horas y 1 día.

Bajo rendimiento: Más de 1 día.

---

La métrica de Tiempo de Restauración del Servicio se calcula siguiendo estos pasos:

Identificar cada incidente dentro de un período específico (por ejemplo, una semana o un mes).

Medir el tiempo de restauración de cada incidente, contabilizando el tiempo desde que ocurre el fallo hasta que el servicio se restablece completamente.

Sumar todos los tiempos de restauración obtenidos.

Dividir la suma entre el número total de incidentes para obtener el tiempo promedio de restauración.

Por ejemplo, si en una semana ocurren tres incidentes con tiempos de restauración de 20, 45, y 30 minutos, el cálculo del tiempo promedio de recuperación sería:

Este valor representa la eficiencia del equipo para restaurar el servicio después de un fallo, y un valor menor indica una respuesta más rápida y eficaz.

Estas métricas son de vital importancia para los equipos de DevOps ya que permiten a los equipos de desarrollo evaluar tanto su eficiencia como su confiabilidad y son cruciales para mejorar prácticas y resultados en DevOps. Una organización que logra buenos valores en estas métricas puede considerarse de alto rendimiento en cuanto a la entrega de software y capacidad de recuperación ante fallos. Optimizar estos aspectos contribuye a mejorar no solo la velocidad de entrega, sino también la estabilidad operacional del equipo.

### 3.7 IMPACTO EN LA TOMA DE DECISIONES PARA EL DISEÑO Y DESARROLLO DEL SISTEMA.

La toma de decisiones en el diseño y desarrollo del sistema Historial Financiero se ve profundamente influenciada por la integración de tres enfoques clave: DevOps, metodologías ágiles y Lean Inception. Estos conceptos trabajan en conjunto para

---

transformar la manera en que se estructuran y gestionan los proyectos, garantizando un proceso más ágil, eficiente y centrado en el cliente. A continuación, se describe cómo cada uno de estos enfoques impacta las decisiones a lo largo de las diferentes etapas del ciclo de vida del sistema.

### 3.7.1 DEFINICIÓN DEL ALCANCE: LEAN INCEPTION Y EL MVP

En la etapa inicial del desarrollo, la definición del alcance del proyecto es esencial para asegurar que los esfuerzos estén alineados con las necesidades del cliente y los objetivos del negocio. En este contexto, Lean Inception influye directamente en la toma de decisiones:

**Definición del MVP:** Lean Inception guía la decisión de desarrollar un Producto Mínimo Viable (MVP) que contemple solo las funcionalidades esenciales para validar el valor del sistema en el mercado. Esta decisión es clave para reducir el riesgo, permitiendo que el equipo obtenga retroalimentación temprana antes de invertir grandes recursos en características adicionales.

**Validación Temprana:** Con la definición del MVP, se decide realizar validaciones tempranas con usuarios para asegurar que el producto satisfaga sus necesidades y expectativas. Esto permite ajustar el rumbo del proyecto basándose en datos reales y no en suposiciones.

### 3.7.2 PLANIFICACIÓN Y EJECUCIÓN: METODOLOGÍAS ÁGILES

Una vez definido el alcance, las metodologías ágiles influyen en cómo se planifican y ejecutan las tareas de desarrollo:

---

**Flexibilidad para Adaptarse a Cambios:** La metodología ágil permite que las decisiones de diseño evolucionen conforme a los cambios que surjan en los requisitos del cliente a lo largo del proceso de desarrollo. A diferencia de los enfoques tradicionales, ágiles promueve la adaptabilidad sin afectar la calidad del producto final.

**Priorización de Funcionalidades:** Mediante prácticas como Scrum, se toma la decisión de priorizar las funcionalidades del sistema según su valor para el cliente y la viabilidad técnica. Esta priorización asegura que el equipo se enfoque en entregar las características más relevantes primero.

**Entrega Iterativa:** Se decide trabajar en sprints, entregando versiones funcionales del sistema al final de cada ciclo. Esto permite obtener retroalimentación continua del cliente, facilitando la validación de funcionalidades y ajustando el diseño a medida que se avanza en el desarrollo.

### 3.7.3 IMPLEMENTACIÓN Y ENTREGA: DEVOPS

La implementación y entrega del sistema Historial Financiero también se ve influenciada por las prácticas de DevOps, que aportan eficiencia y rapidez al ciclo de vida del software:

**Automatización del Ciclo de Vida:** En la etapa de implementación, se toma la decisión de automatizar procesos clave como pruebas, integración y despliegue mediante CI/CD (Integración y Despliegue Continuo). Esto reduce el tiempo de comercialización y minimiza los errores humanos, garantizando la consistencia y fiabilidad del producto en cada versión entregada.

**Colaboración entre Equipos Multifuncionales:** DevOps fomenta la decisión de crear equipos multifuncionales que trabajen de manera conjunta desde el inicio del proyecto. Esta colaboración entre desarrolladores, operaciones y personal de

---

seguridad mejora la comunicación y la rapidez en la resolución de problemas, acelerando el proceso de desarrollo.

**Entrega Continua:** A través de la entrega continua, se toma la decisión de hacer que el sistema sea capaz de recibir actualizaciones frecuentes y de bajo riesgo. Esto permite adaptarse rápidamente a los cambios del mercado y responder a las necesidades del cliente con mayor agilidad.

### 3.7.4. EVALUACIÓN DEL DESEMPEÑO: MÉTRICAS DORA

Una vez que el sistema está en funcionamiento, la evaluación de su desempeño se basa en el uso de métricas DORA, que permiten tomar decisiones informadas para la mejora continua del proyecto:

**Frecuencia de Despliegue:** Las métricas de DORA influyen en la decisión de medir la frecuencia con la que se realizan despliegues del sistema. Esta métrica es fundamental para evaluar la rapidez con que se lanzan nuevas versiones y se corrigen errores.

**Tiempo de Espera para el Cambio:** Se decide medir el tiempo que transcurre desde que se identifica la necesidad de un cambio hasta que se implementa en producción. Esta métrica ayuda a mejorar la velocidad del ciclo de desarrollo y la capacidad de adaptación a nuevas demandas.

**Tasa de Fallos y Tiempo de Recuperación:** Las decisiones sobre cómo manejar incidencias en producción se basan en las métricas de DORA, que permiten medir la tasa de fallos y el tiempo necesario para restaurar el sistema en caso de errores. Esta información es crucial para optimizar la estabilidad del sistema y reducir tiempos de inactividad.

---

## SINOPSIS CAPÍTULO 4

En el capítulo anterior, se identificaron los fundamentos conceptuales y metodológicos que sustentan la propuesta de implementación de principios DevOps y metodologías ágiles para el desarrollo del MVP del historial financiero, integrando un modelo de CI/CD.

En este contexto, el presente capítulo se centra en el desarrollo integral del proyecto, iniciando con la aplicación de la metodología Lean Inception. Esta estrategia permitirá generar un lienzo con las principales funcionalidades que debe contener el MVP, tomando como base los aportes de los participantes.

Una vez definidas estas funcionalidades, se procederá al diseño de mockups, los cuales facilitarán un primer acercamiento con los usuarios respecto al diseño y las características del aplicativo. Este proceso será clave para obtener retroalimentación temprana y realizar ajustes oportunos.

Con los mockups validados, el siguiente paso será la selección de herramientas para la implementación del modelo CI/CD, abarcando tanto la configuración como la integración de diversas soluciones que apoyarán el ciclo de vida del historial financiero.

Posteriormente, se procederá con el desarrollo del historial financiero, incluyendo la integración del código fuente, su construcción y despliegue en los diferentes entornos establecidos en los pipelines de GitLab.

Finalmente, para evaluar y mejorar la implementación de las prácticas DevOps, se emplearán las métricas DORA. Estas métricas clave —frecuencia de despliegue, tiempo de entrega de cambios, tasa de fallos de cambios y tiempo medio de recuperación— permitirán monitorear el desempeño del equipo y optimizar el proceso de desarrollo.

---

## 4. DESARROLLO DEL PROYECTO.

### 4.1 DISEÑO DE LA INVESTIGACIÓN.

En la presente tesis se emplean diversos enfoques de diseño de investigación para garantizar una implementación efectiva de los principios DevOps y las metodologías ágiles en el desarrollo del módulo Historial Financiero.

En primer lugar, se adopta un enfoque de investigación aplicada, cuyo objetivo es mejorar el proceso de desarrollo de software mediante la construcción de un Minimum Viable Product (MVP) funcional y eficiente. Para ello, se implementa un modelo de integración y despliegue continuo (CI/CD), que optimiza la entrega de nuevas funcionalidades y mejora la calidad del software.

En segundo lugar, se incorpora la investigación experimental, que permite evaluar de manera sistemática la eficiencia del ciclo de vida del software. A través de pruebas y mediciones, se analizan indicadores clave como tiempos de desarrollo, estabilidad de las versiones y velocidad de implementación, lo que facilita la identificación de mejoras en el proceso.

Finalmente, se emplea un enfoque de investigación en acción, basado en ciclos iterativos de planificación, acción, observación y reflexión. A medida que se desarrolla el MVP, se recopila retroalimentación tanto de los usuarios finales como del equipo de desarrollo. Esta información es utilizada para realizar mejoras continuas en el producto y en los procesos de desarrollo, asegurando así una evolución constante del software en función de las necesidades detectadas.

Este enfoque integral permite validar y perfeccionar la aplicación de DevOps y metodologías ágiles en el desarrollo del módulo Historial Financiero, fomentando una cultura de mejora continua y entrega eficiente de valor al usuario.

---

## 4.2 MODALIDAD DE LA INVESTIGACIÓN.

La recopilación de requisitos es una fase esencial en el desarrollo de software, ya que permite definir las funcionalidades y características clave del producto final. Para este propósito, se emplea la metodología ágil Lean Inception, que facilita la identificación y comprensión de las necesidades reales del cliente, optimizando la definición del Producto Mínimo Viable (MVP).

En este contexto, la investigación adopta un enfoque mixto, combinando métodos cualitativos y cuantitativos para garantizar un análisis integral del proceso de desarrollo.

El enfoque cualitativo se aplica en la fase inicial de recopilación de requisitos y validación de necesidades mediante Lean Inception, que integra los principios de Design Thinking y Lean Startup. Design Thinking permite comprender y resolver problemas desde la perspectiva del usuario a través de las etapas de Empatizar, Definir, Prototipar y Testear, mientras que Lean Startup reduce la incertidumbre en el desarrollo mediante el ciclo Construir, Medir y Aprender, asegurando la validación temprana de las ideas. Para aplicar esta metodología, se coordinan reuniones con el equipo de desarrollo siguiendo la agenda establecida por Paulo Caroli, utilizando sesiones colaborativas, entrevistas y dinámicas grupales para identificar y refinar los requisitos esenciales del MVP.

El enfoque cuantitativo se emplea para evaluar la efectividad de las decisiones tomadas en la fase de desarrollo e implementación del MVP. Se utilizan métricas y análisis de datos para medir aspectos como la eficiencia en el desarrollo, la cantidad de iteraciones requeridas para ajustar el MVP según la retroalimentación, y la precisión en la consolidación de datos financieros, asegurando que la información refleje correctamente las operaciones de crédito, ingresos y captaciones de los socios.

Este enfoque mixto permite obtener una comprensión detallada de los requisitos del sistema y medir objetivamente el impacto de Lean Inception en la mejora del

---

proceso de desarrollo, garantizando que el MVP sea funcional y cumpla con los requerimientos del cliente.

## 4.3 EJECUCIÓN DE LA METODOLOGÍA DE LA INVESTIGACIÓN.

---

En la **imagen 5** se presenta la metodología seleccionada para la implementación del proyecto, basada en un enfoque DevOps que integra prácticas ágiles y de desarrollo continuo.

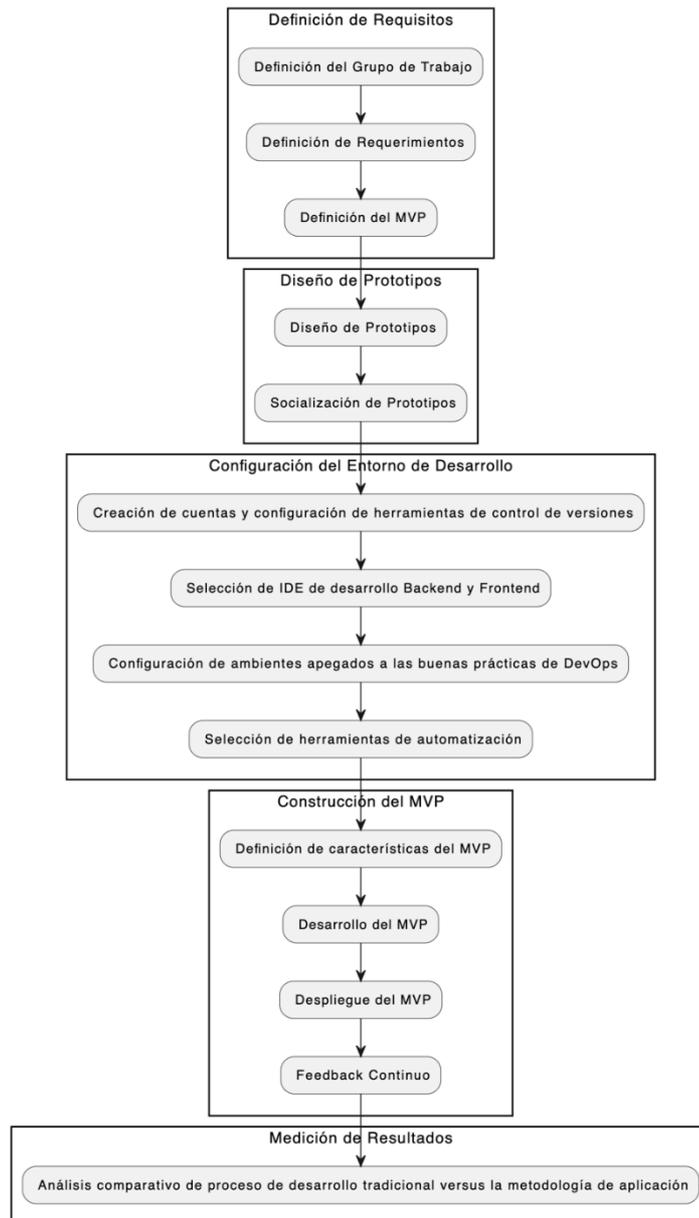


Imagen 3 Metodología implementación DevOps.

### 4.3.1 DEFINICIÓN DEL GRUPO DE TRABAJO.

Para garantizar el éxito en el levantamiento de requisitos, se consideró fundamental la realización de reuniones de trabajo orientadas a la selección del personal idóneo, en función de las capacidades, funciones y requerimientos del proyecto. Como

---

resultado, se constituyó un equipo multifuncional conformado por un Product Owner, un Scrum Master, los Stakeholders así como un Development Team.

Para asegurar un levantamiento de requisitos efectivo y estructurado, se estableció un equipo de trabajo conformado por profesionales con roles clave dentro del proyecto. La selección del equipo se realizó considerando las capacidades, funciones y necesidades específicas del desarrollo. Como resultado, se conformó un equipo multifuncional, compuesto por:

**Product Owner:** Responsable de representar las necesidades del cliente y priorizar los requisitos del producto.

**Scrum Master:** Encargado de facilitar las sesiones de trabajo y garantizar la aplicación correcta de la metodología ágil.

**Stakeholders:** Actores clave que aportan una visión estratégica y requisitos esenciales del negocio.

**Development Team:** Grupo de desarrolladores encargados de la implementación técnica de las funcionalidades definidas.

Esta estructura organizativa permitió optimizar la comunicación, agilizar la toma de decisiones y garantizar una correcta alineación con los objetivos del proyecto.

### 4.3.2 DEFINICIÓN DE REQUERIMIENTOS MEDIANTE LA AGENDA LEAN INCEPTION.

Con el objetivo de llevar un control adecuado del tiempo, poder capturar y entender los requisitos tanto funcionales como técnicos se decidió crear reuniones de trabajo durante una semana en la cual se trabajó en cada punto de la agenda en sesiones de una hora por la mañana y otra por la tarde. Dichas sesiones de trabajo se basaron en la agenda recomendada por Paulo Caroli (Caroli, Lean Inception Creando Conversaciones hacia un producto exitoso., 2018) para iniciar la implementación de Lean Inception.

Este enfoque, promueve la colaboración entre equipos multidisciplinares, lo que permite una comprensión más clara y rápida de los objetivos del proyecto. La agenda Lean Inception se puede observar en la **imagen 6**.

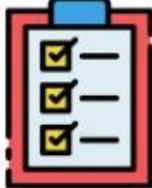
AGENDA PARA TALLER LEAN INCEPTION			
	MAÑANA	TARDE	
LUNES	 Kick-off	 Visión del Producto	 El producto es / no es hace / no hace
MARTES	 Identificando a las personas.	 Descubriendo las funcionalidades.	
MIERCOLES	 Entendimiento Técnico y de experiencia de usuario.	 Describiendo los viajes.	
JUEVES	 Exponiendo funcionalidades en los viajes.	 Secuenciando las funcionalidades.	
VIERNES	 Construyendo el Canvas MVP.	 Presentando los resultados.	

Imagen 4 Agenda Lean Inception

---

### 4.3.3 LANZAMIENTO (KICK-OFF).

Esta es la primera actividad que se realizará en la mañana, el equipo de trabajo Product owner, stakeholders, personal de desarrollo, usuarios finales y personas clave son reunidos con el objetivo de establecer una comprensión compartida del proyecto, su propósito, sus objetivos y los criterios de éxito. Es una actividad muy importante ya que los participantes se sentirán involucrados ya que sus ideas serán escuchadas participarán de una manera más activa, además en esta actividad se debe explicar cómo funcionará la metodología Lean Inception, definir las horas de reunión y de preferencia reunirse en la misma sala durante toda la agenda.

### 4.3.4 ESCRIBIENDO LA VISIÓN DEL PRODUCTO

Una vez que los participantes entendieron la metodología, se procedió a solicitar a los participantes sus opiniones referentes hacia dónde se ve el producto en el tiempo, de las cuales surgen varios enunciados, una vez compartidas las ideas con todos se obtuvo como resultado la siguiente visión del producto: "Se aspira contar con un sistema de información financiera integral que transforme la forma en que los asesores cooperativos acceden y utilizan los datos financieros de los socios. Se busca ofrecer una plataforma que optimice la presentación de información, agilice la toma de decisiones y mejore la eficiencia general del proceso".

### 4.3.5 EL PRODUCTO ES, NO ES / HACE, NO HACE.

Para este punto, se utilizó una pizarra, la cual fue dividida en 4 partes, se solicitó a cada participante compartir sus percepciones sobre qué es el producto y qué no, así como las funciones deben cumplir y cuáles no. Luego se procedió a verificar las respuestas y discutir las con el equipo de trabajo para ver las más relevantes y agrupar las que estén repetidas.

Finalmente se agrupó los puntos clave y se los consolidó en una sola frase y se obtuvo el siguiente resultado: "Es un aplicativo web dinámico y amigable para los usuarios, de uso interno, que permite obtener información consolidada acompañada de sus respectivos detalles sobre el historial financiero de los socios" como se puede visualizar en la **imagen 7**.



Imagen 5 Descripción del producto.

---

### 4.3.6 DESCRIBIENDO LAS PERSONAS.

En nuestro segundo día, se trabajó con cada uno de los participantes para obtener una descripción de las diferentes personas que utilizarán el sistema. Para ello, se proporcionan hojas de papel en blanco donde cada uno realiza una descripción clara de los usuarios como se puede ver en la imagen 8, lo que resulta en el siguiente conjunto de descripciones.

- Asesor Cooperativo.
- Asesor de centro de atención virtual.
- Analista de soporte operativo.
- Analista de innovación y desarrollo.



*Imagen 6 Describiendo las Personas*

---

### 4.3.7 LLUVIA DE IDEAS DE LAS FUNCIONALIDADES.

Una vez que los diferentes usuarios del sistema han sido definidos claramente, es esencial que se comprendan las funcionalidades que el sistema debe llevar a cabo para alcanzar los objetivos establecidos por cada uno de los usuarios. Por lo tanto, en colaboración con el equipo, se realiza una sesión de lluvia de ideas utilizando post-it y colocándolos en la pizarra para describir las funcionalidades requeridas, como se puede observar en la imagen 9. Posteriormente, una vez que las funcionalidades han sido obtenidas, se procede con las validaciones necesarias con el fin de eliminar funcionalidades duplicadas, unificar funcionalidades cuyo objetivo sea el mismo, así como la respectiva priorización en un orden secuencial de las diferentes funciones requeridas que les permita a cada uno de los usuarios alcanzar sus objetivos.



*Imagen 7 Lluvia de ideas de las funcionalidades*



Por otro lado, se ha considerado la certeza técnica, que se refiere a la claridad del equipo técnico sobre cómo construir la funcionalidad requerida. Por consiguiente, cada funcionalidad ha sido clasificada según el nivel de certeza técnica, la confianza en el entendimiento del negocio y la experiencia del usuario. Esto ha proporcionado una posición relativa respecto al nivel de certidumbre. Se ha empleado un gráfico denominado "gráfico del semáforo", como se observa en la **imagen 11**, donde los colores indican el nivel de precaución necesario: verde para continuar, amarillo para prestar atención y rojo para detenerse y esperar. Al finalizar la revisión, todas las funcionalidades han sido evaluadas y analizadas.

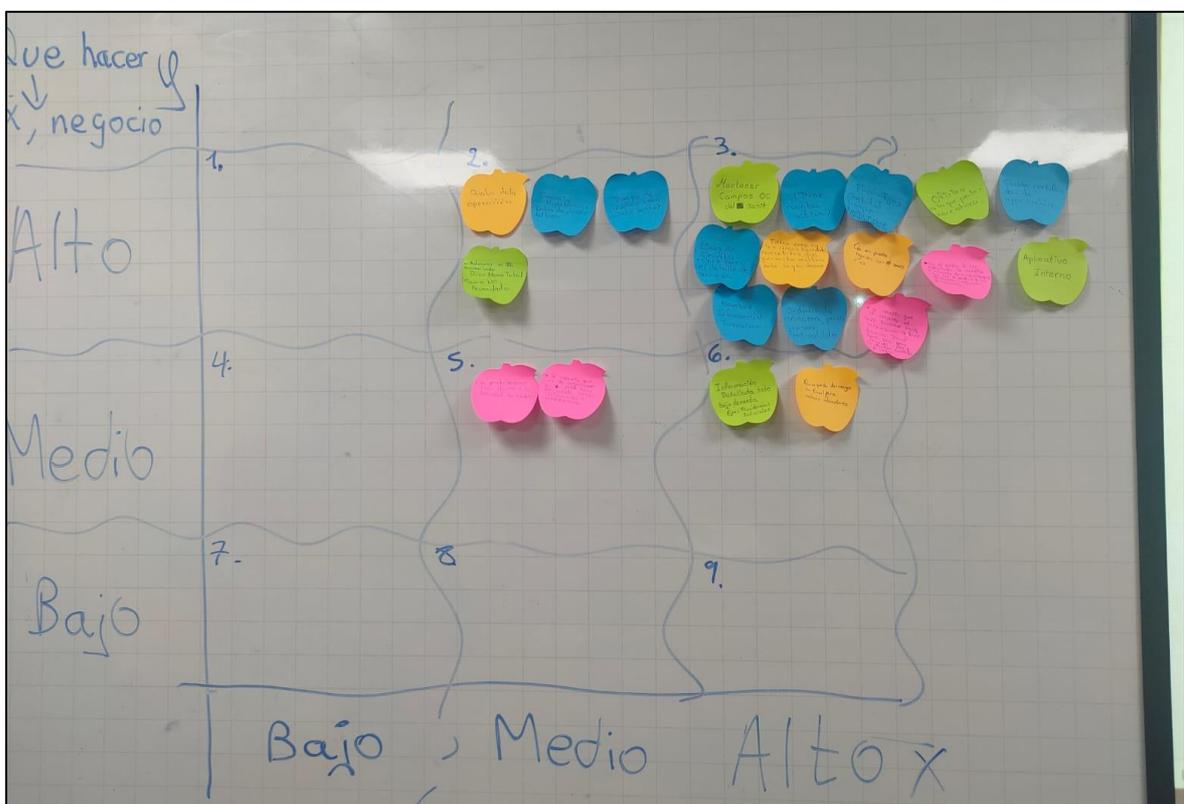


Imagen 9 Gráfico del semáforo

El desarrollo de esta actividad fue crucial para que el equipo adquiriera un conocimiento más amplio sobre qué funcionalidades requieren un mayor esfuerzo y cuáles considerar descartar. Cada funcionalidad se evalúa en términos técnicos, de experiencia de usuario y de valor de negocio. Esto se realiza primero mapeando las funcionalidades en el gráfico del semáforo y luego marcándolas según una tabla que mide el esfuerzo, la experiencia de usuario y el valor de negocio. El color y las

---

marcas asignadas ayudan al equipo en actividades posteriores de priorización, estimación y planificación.

### 4.3.9 MOSTRANDO LOS VIAJES DE USUARIOS.

Una vez comprendidas a fondo las funcionalidades de la aplicación, se procede a su validación con el equipo y usuarios expertos. Se detalla, paso a paso, la secuencia del uso diario actual para alcanzar el objetivo, aplicando el modelo C4 para lograr una mejor visualización. Este modelo facilita la comprensión de la estructura del software y sus interacciones con usuarios y sistemas externos. Para ello, se utilizan los diferentes diagramas que componen este modelo:

**Diagramas de Contexto**, nos permite ver el panorama general como está estructurado el Software y los usuarios que interactúan con él, así como los Sistemas externos con los cuales interactúa nuestro sistema (Brown, s.f.).

**Diagrama de Contenedor**, permite visualizar de manera general la arquitectura del software permitiendo ver a cada integrante como una única unidad ejecutable por separado que forma parte del sistema (Brown, s.f.).

**Diagrama de Componentes**, permite visualizar cómo un contenedor está formado por una serie de componentes, así como sus responsabilidades y los detalles de tecnología e implementación (Brown, s.f.) Esta información se ilustra en las tareas realizadas en las **imágenes 12, 13, 14**.

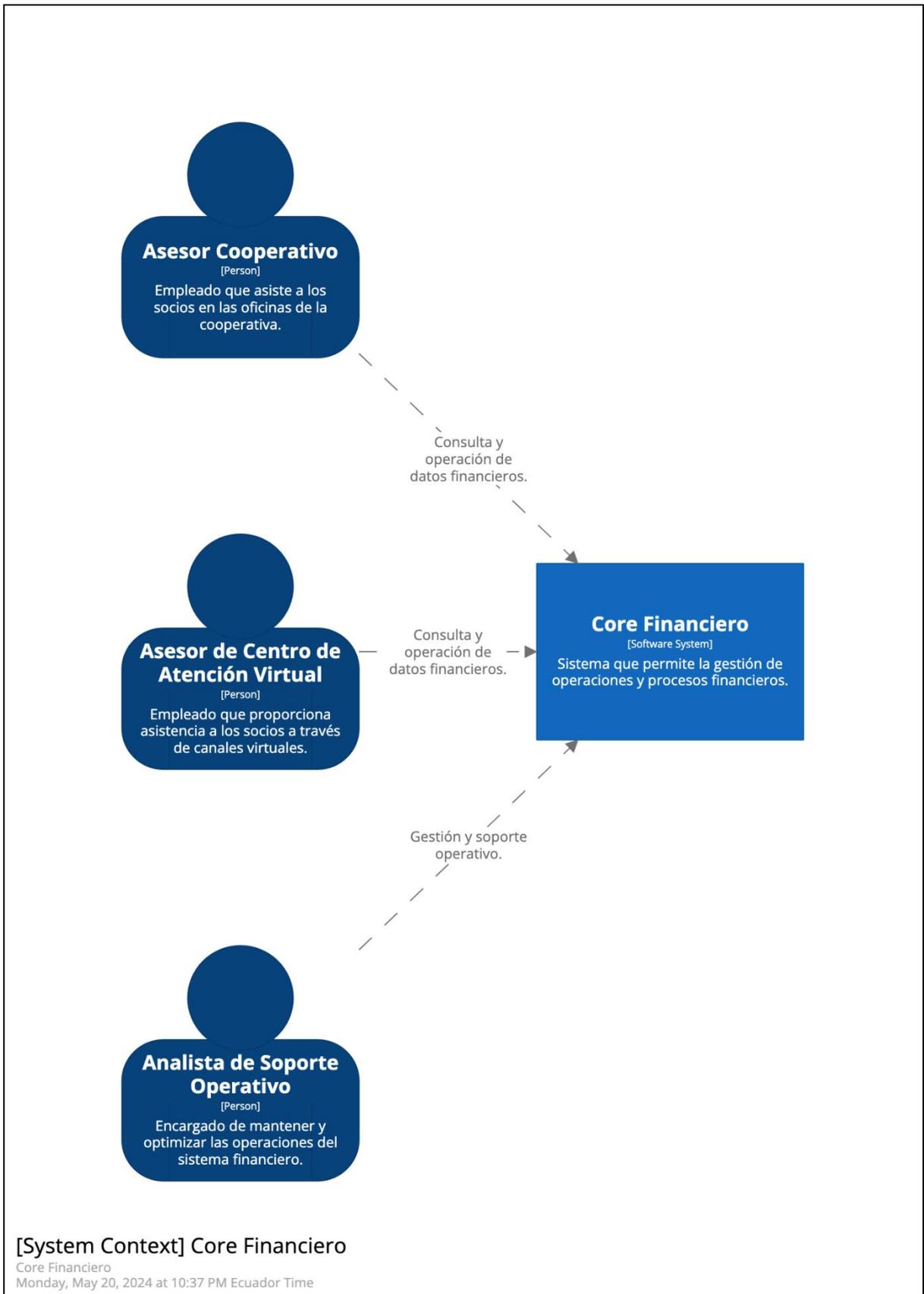


Imagen 10 Diagrama de Contexto – Secuencia de funcionalidades actual del historial financiero

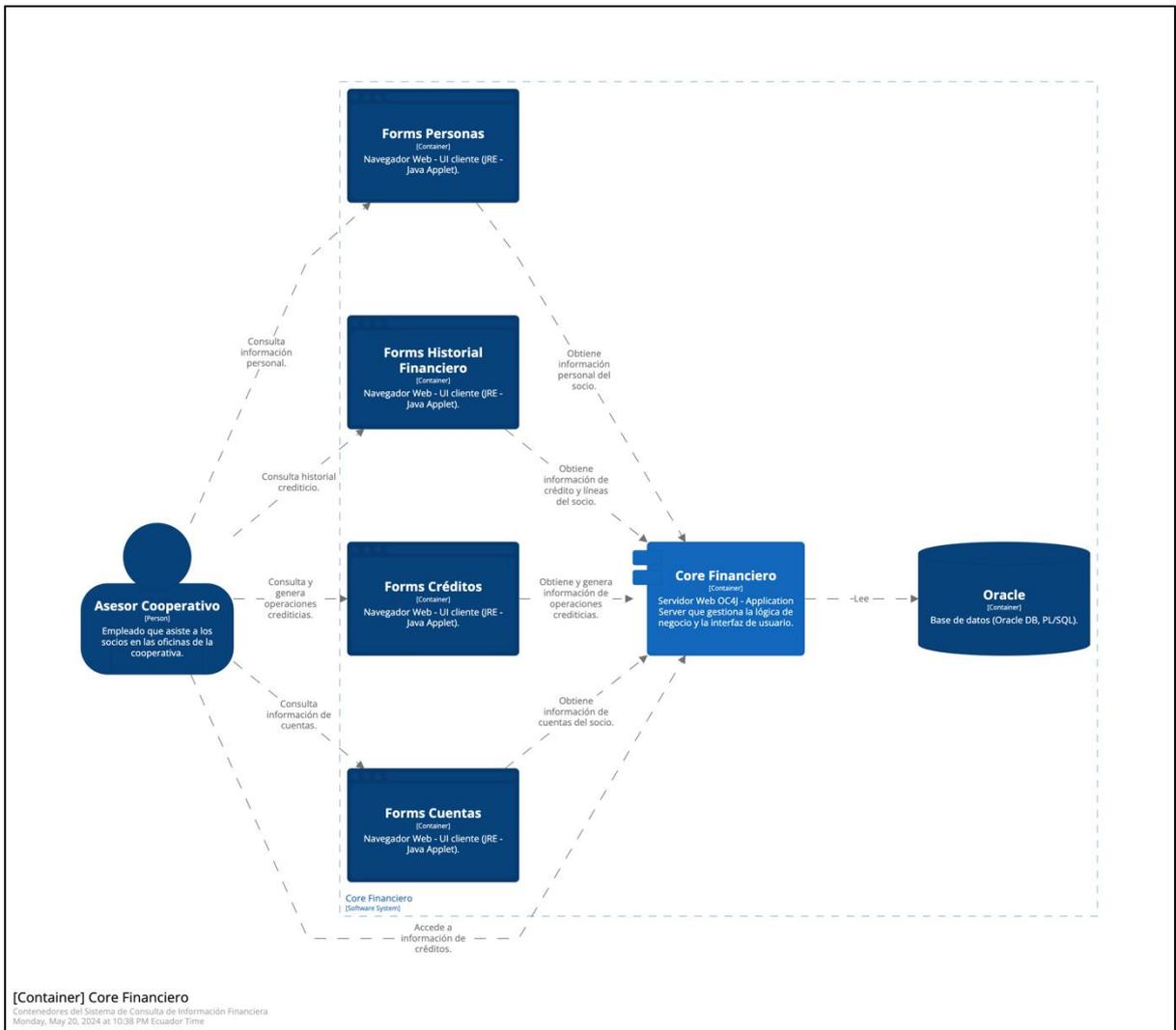


Imagen 11 Diagrama de Contenedor – Secuencia de funcionalidades actual del historial financiero

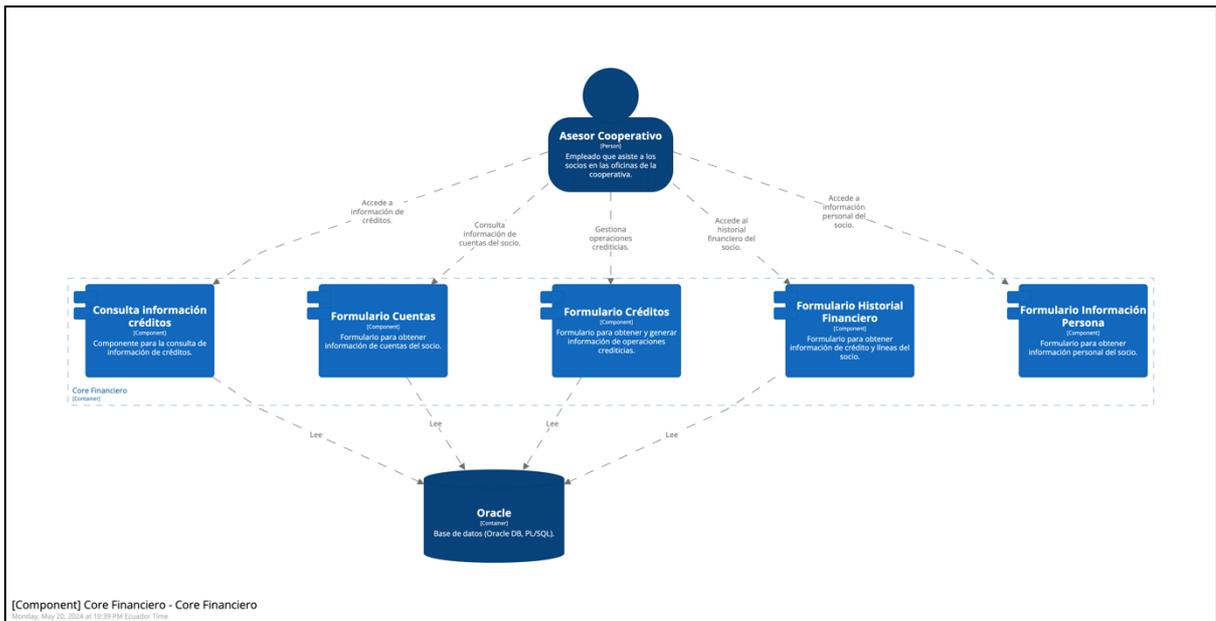


Imagen 12 Diagrama de Componente – Secuencia de funcionalidades actual del historial financiero

Sin embargo, la descripción detallada de estos pasos, especialmente en relación con la obtención de información crediticia de un socio, ha sido excesiva. Por consiguiente, en consonancia con la secuencia de pasos actual, se ha trabajado en colaboración con el equipo para consolidar esta secuencia en un único punto de acceso. Esta iniciativa tiene como objetivo mejorar la eficiencia de las actividades diarias de los usuarios y acelerar el proceso de validación del historial financiero. Para ello, se propone una nueva secuencia de funcionalidades, detalladas mediante el uso de la herramienta modelo C4, como se explicó en pasos anteriores, que permite visualizar una mejora significativa, tal como se evidencia en las **imágenes 15, 16, 17, 18, 19.**

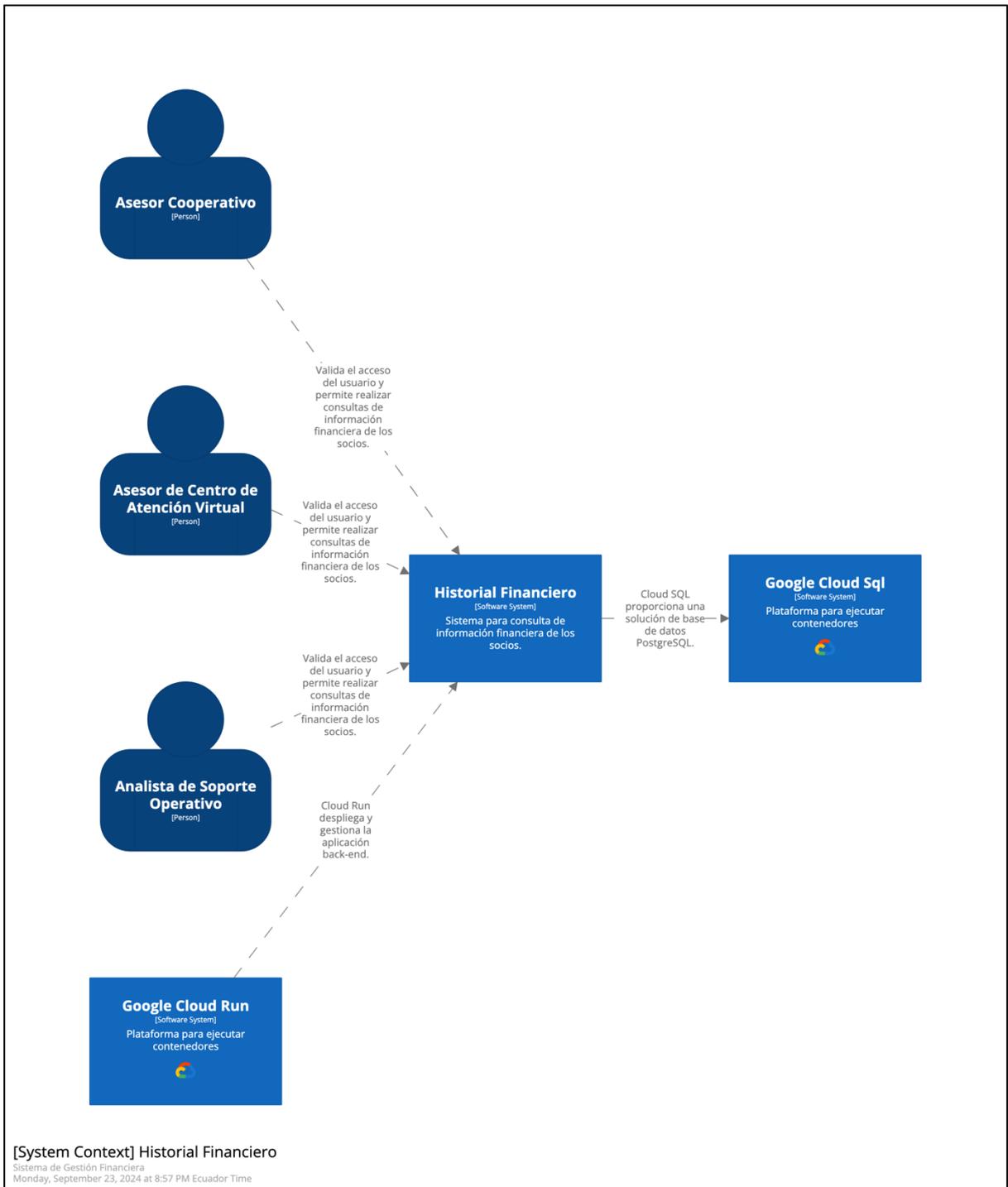


Imagen 13

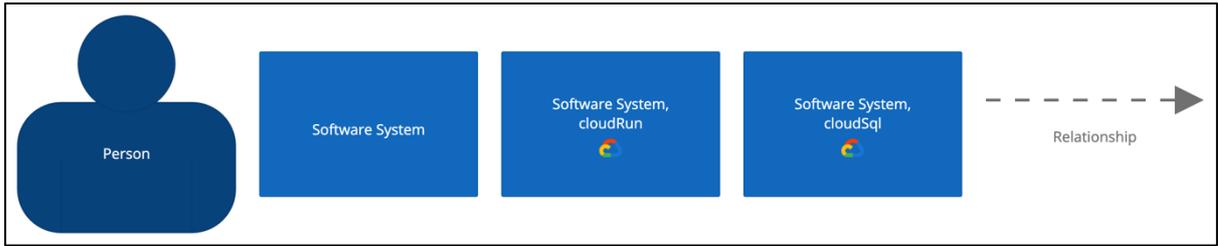


Imagen 14 Secuencia de funcionalidades -Diagrama de Contexto

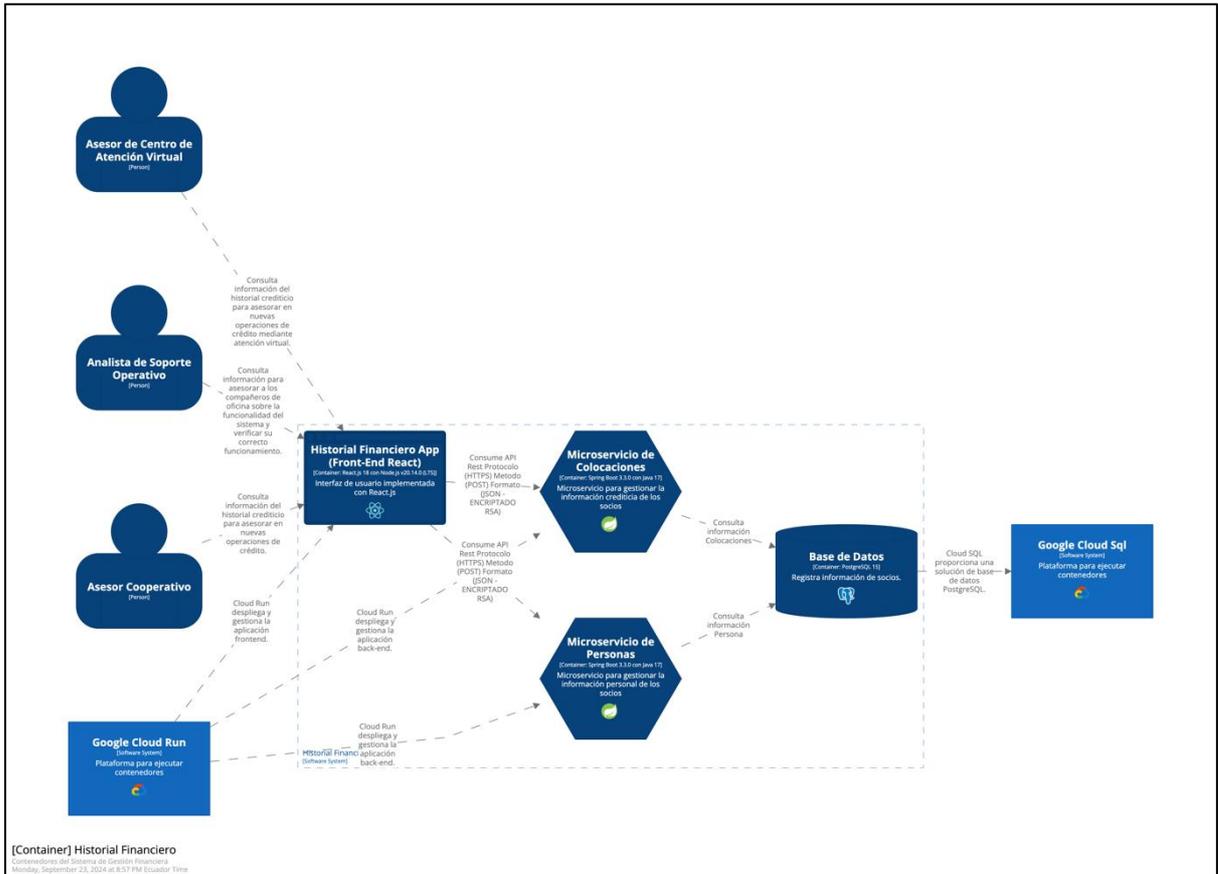


Imagen 15 Secuencia de funcionalidades – Diagrama de contenedor

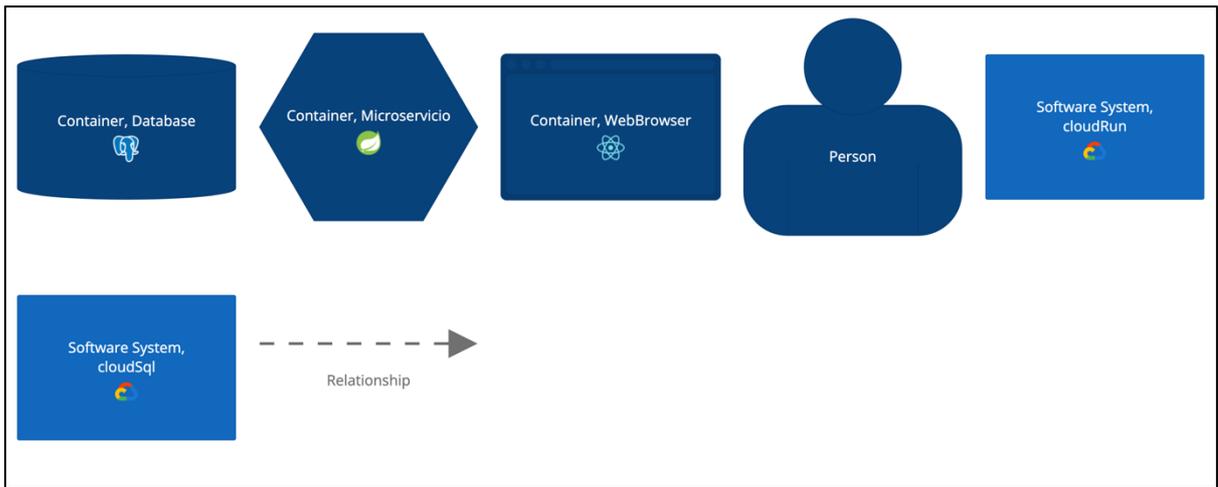


Imagen 16

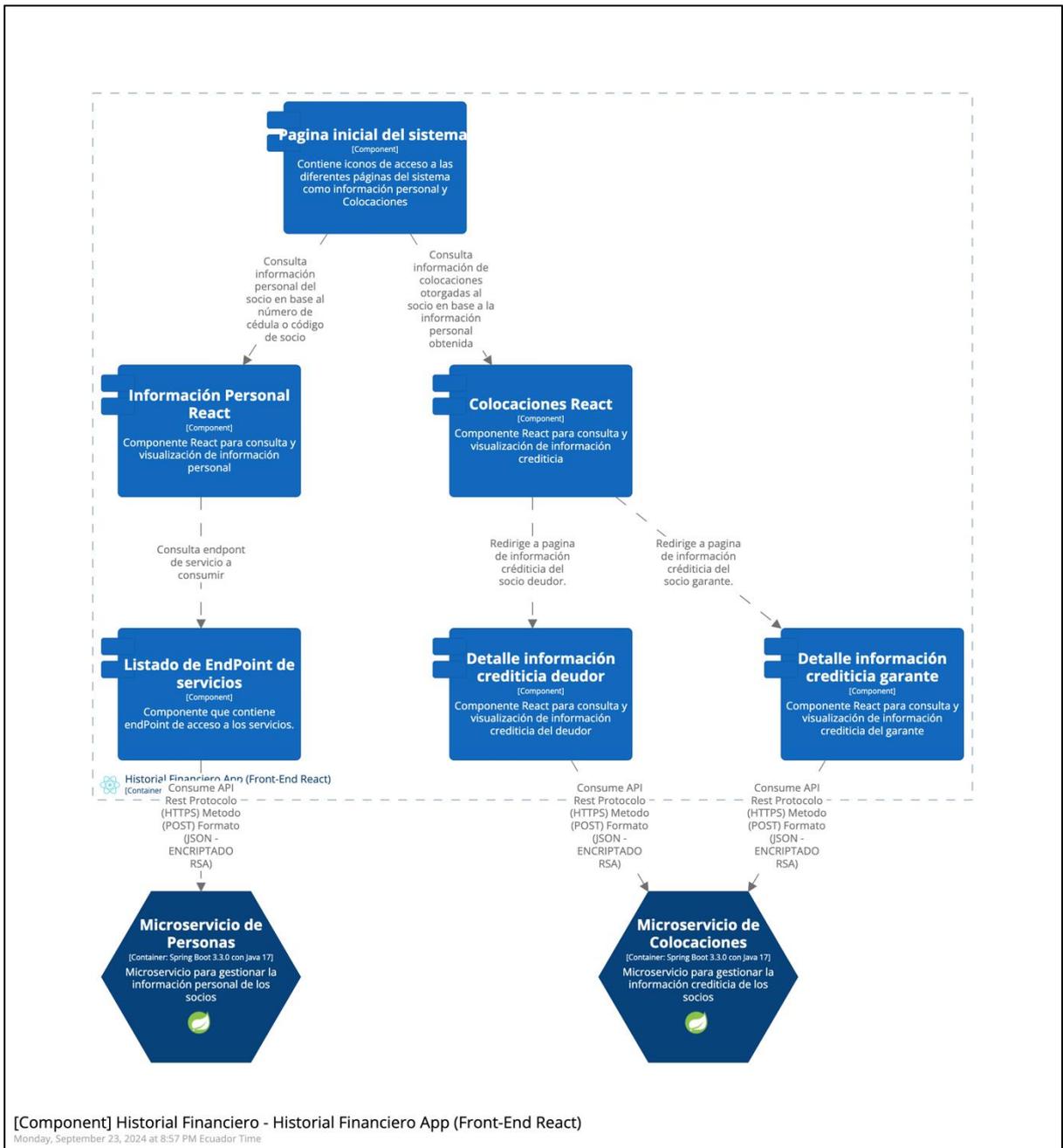


Imagen 17 Secuencia de funcionalidades – Diagrama de API de componente Front-End

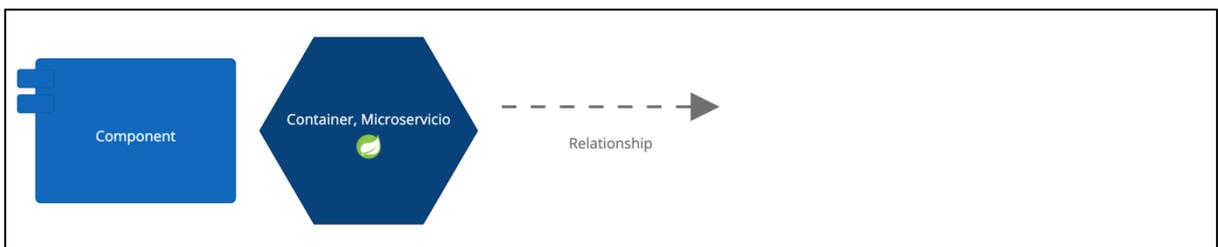
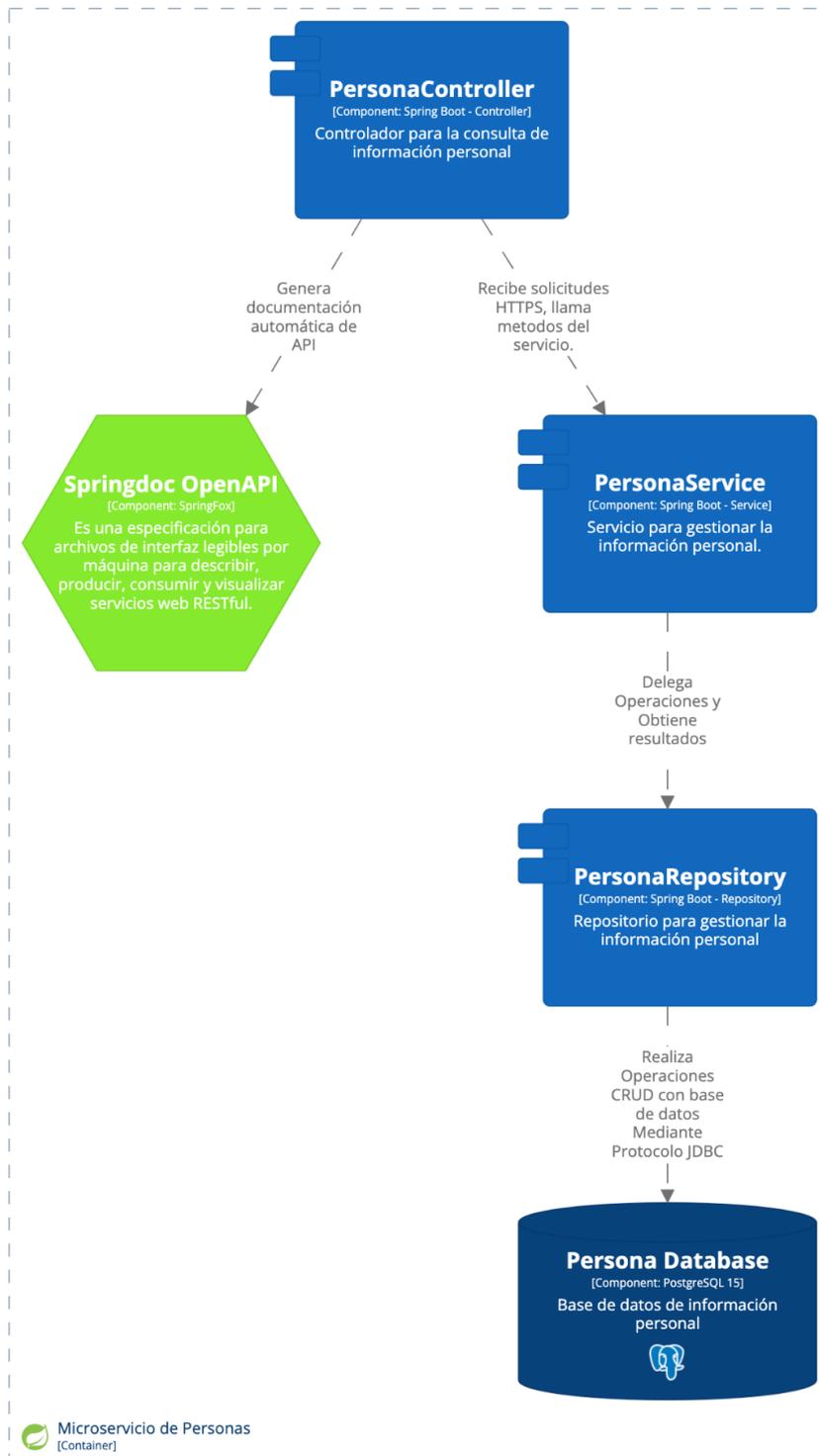


Imagen 18



[Component] Historial Financiero - Microservicio de Personas

Tuesday, September 24, 2024 at 12:38 PM Ecuador Time

Imagen 19 Secuencia de funcionalidades – Diagrama de componente – Microservicio Personas

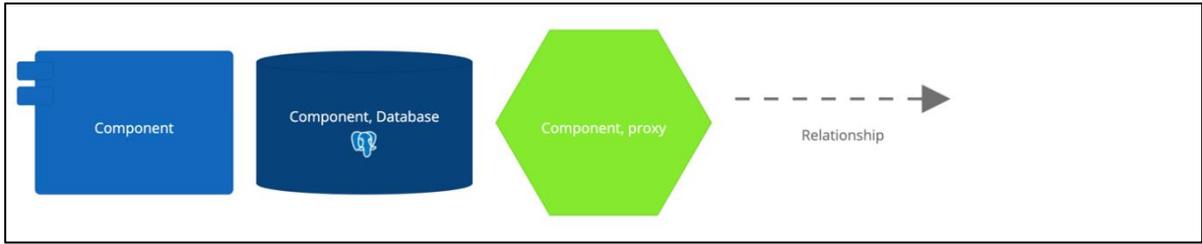


Imagen 20

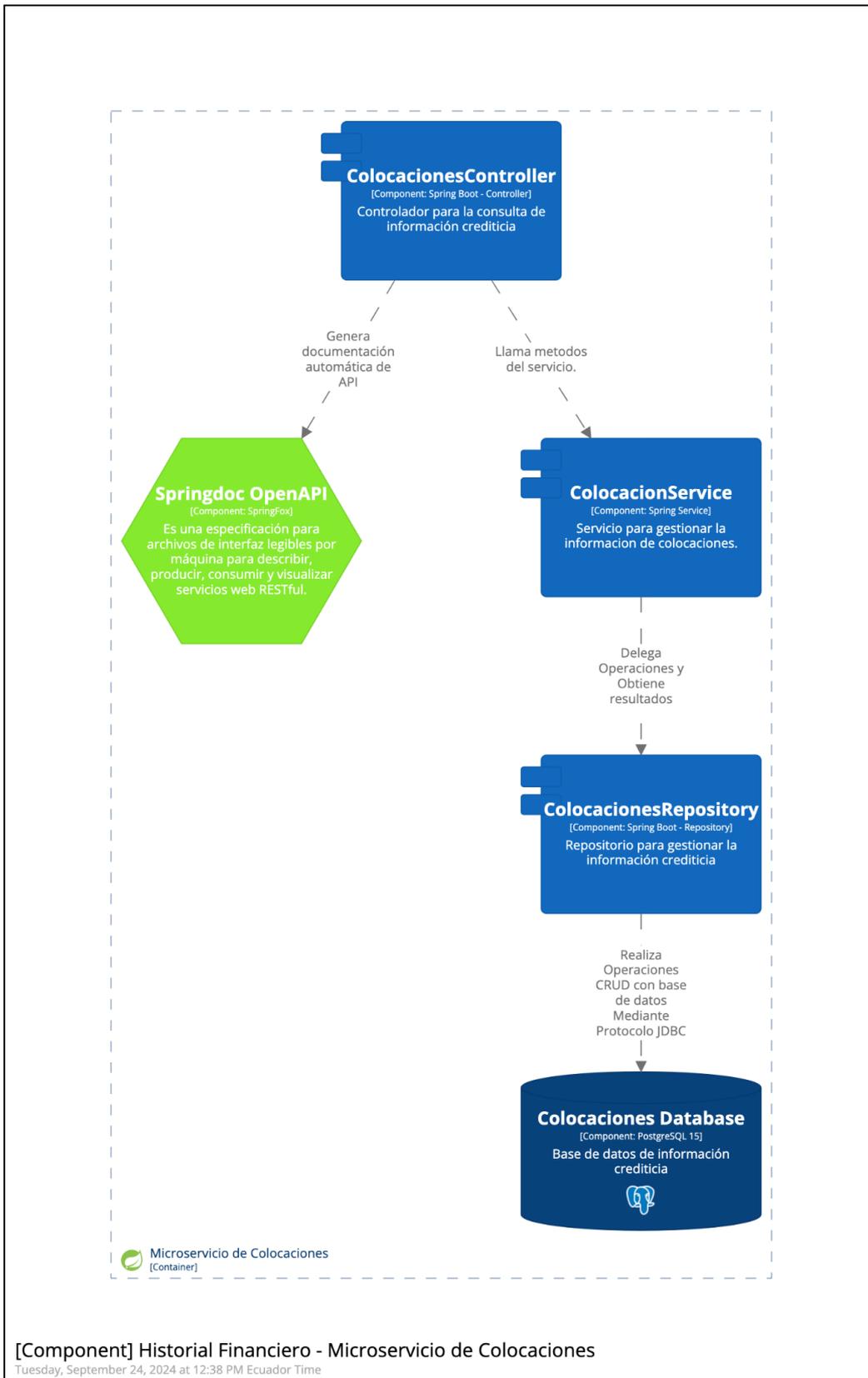


Imagen 21 Secuencia de funcionalidades – Diagrama de contenedor – Microservicio de Colocaciones.

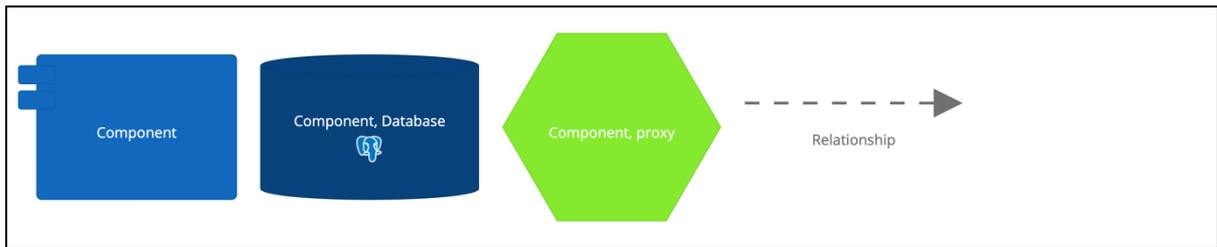


Imagen 22

#### 4.3.10 CONSTRUCCIÓN SECUENCIADOR DE FUNCIONALIDADES.

Una vez identificadas las funcionalidades junto con sus evaluaciones en términos de esfuerzo, valor para el negocio, experiencia del usuario y certeza técnica, nos enfrentamos a la pregunta crucial: "¿Cuál de estas dos funcionalidades es la más prioritaria?". Responder a esta pregunta requirió un punto de partida claro.

Para encontrar ese punto de partida, se recurrió a la identificación de la persona más relevante y el viaje del usuario más prioritario. Esto proporcionó el inicio necesario, comenzando con la primera funcionalidad de este viaje. Se colaboró estrechamente con el equipo de trabajo para determinar cuál de estas tiene la más alta prioridad, considerando las valoraciones previamente establecidas, tales como el valor para el negocio, el esfuerzo requerido, la experiencia del usuario y el nivel de certeza técnica.

Luego, en conjunto, se seleccionaron las funcionalidades que ayuden a cumplir con los objetivos de los usuarios y permitan la creación de un producto mínimo viable (MVP). Aunque la regla indica que no debe haber más de tres funcionalidades en una secuencia, se decidió agregar cuatro funcionalidades después de deliberar con el equipo, debido a su estrecha relación y su importancia como base fundamental para el MVP. Este MVP permitirá validar las hipótesis planteadas y, basándonos en sus resultados, decidir si continuamos con las siguientes secuencias o Sprints. El resultado de esta decisión se visualiza en la **Tabla 2**.

## FUNCIONALIDADES

1	Mantener Campos OC del JASIT.	Ingresar con Código Socio.	Filtros de Estado de crédito.	Nombre: Información Financiera.
2	Enlace al detalle de pago de crédito.	Cuenta Activa	Oficina a la que pertenece el socio.	Saldos promedio.
3	Enlace a los detalles de los avances de línea de Crédito.	Muestra la decisión de la Operación Crediticia.	Valor la cuota de cada operación.	Saldos de Certificados de aportación.
4	Providencias Judiciales.	Se puede Imprimir para adjunta a las solicitudes de Crédito.	Garantías Hipotecarias	Plazo de vencimiento de la operación crediticia.

**Tabla 2.** *Secuencia de funcionalidades para la creación del MVP.*

### 4.3.11 CONSTRUYENDO UN CANVAS MVP.

Una vez alcanzado el paso crucial de la construcción del CANVAS MVP dentro del proceso continuo según la metodología, se procede a la documentación del MVP final, ya ejecutada conforme al cronograma establecido. Este proceso ha sido fundamentado en los resultados obtenidos durante la socialización con el equipo de trabajo, siguiendo una plantilla específica proporcionada por Canvas MVP. En esta plantilla se encuentra una estructura clara y organizada para la captura de los elementos esenciales del MVP. Las etapas de este proceso han sido ilustradas en la imagen 25, las cuales han servido de guía en la creación de un MVP sólido y efectivo.

<p><b>2. Segmentos de las personas</b></p>  <ul style="list-style-type: none"> <li>* Asesor Cooperativo</li> <li>* Asesor de Centro de Atención Virtual</li> <li>* Analista de Soporte Operativo</li> <li>* Analista de Innovación y Diseño</li> </ul>	<p><b>1. Propuesta o visión del MVP</b></p>  <ul style="list-style-type: none"> <li>* Migrar el historial crediticio actual a un sistema consolidado para disponer de datos ordenados, claros y concisos.</li> <li>* Proporcionar datos relevantes para disminuir el tiempo de respuesta al socio.</li> </ul>	<p><b>5. Resultado Esperado</b></p>  <ul style="list-style-type: none"> <li>* Disminución de tiempos en obtener la información del socio.</li> <li>* Evitar reprocesos al verificar la información presentando datos relevantes de manera ordenada clara y concisa.</li> </ul>
<p><b>Viajes de usuario</b></p>  <ul style="list-style-type: none"> <li>* Ingresar al sistema</li> <li>* Realizar búsqueda por identificación o código de socio.</li> <li>* Analizar la información personal del socio.</li> <li>* Analizar el historial de créditos como deudor o cónyuge.</li> <li>* Analizar el historia de créditos como garante.</li> </ul>	<p><b>4. Funcionalidades</b></p>  <ul style="list-style-type: none"> <li>Mantener campos OC del JASIT</li> <li>Ingresar con Número de identificación o código de socio</li> <li>Filtros de estado de crédito</li> <li>Nombre: Información Financiera</li> </ul>	<p><b>6. Métricas para la validación de hipótesis de negocio</b></p>  <ul style="list-style-type: none"> <li>* Bajar el tiempo de análisis de la información financiera por parte del asesor en un 25%.</li> <li>* Reducir el uso de 6 pantallas que se tiene actualmente a 2 pantallas.</li> </ul>
<p><b>7. Cronograma</b></p>  <ul style="list-style-type: none"> <li>* Planificación y Configuración Inicial</li> <li>* Desarrollo Inicial</li> <li>* Integración con GitLab</li> <li>* Desarrollo y Pruebas Continuas</li> <li>* Despliegue y Mantenimiento</li> </ul>		

Imagen 23 Canvas MVP

### 4.3.12 SHOW CASE DE LA LEAN INCEPTION.

Dentro del marco de la Lean Inception, el siguiente paso es la realización del “Show Case”. Después de haber completado el cronograma propuesto por la metodología y haber documentado el MVP final utilizando la plantilla proporcionada por Canvas MVP, se llega al punto crucial de mostrar los avances y logros alcanzados durante el proceso. Este evento, conocido como SHOW CASE, representa una oportunidad para que los avances y logros sean presentados al equipo de trabajo y otras partes interesadas relevantes.

Durante el SHOW CASE, se hicieron destacar los aspectos clave del MVP final, resaltando cómo se alinean con los objetivos del proyecto y cómo se abordan las necesidades identificadas. Además, se ofreció una visión general de los pasos a seguir, delineando claramente el camino a seguir para avanzar con éxito en el proyecto.

---

Con la información anterior, el SHOW CASE se convierte en un punto clave en el proceso de la Lean Inception, consolidando el progreso realizado hasta el momento y allanando el camino para el éxito continuo del proyecto.

### 4.3.13 DEFINIR ALCANCE.

Ahora que hemos identificado los problemas actuales con el Historial Financiero de la COAC Jardín Azuayo, estamos dando pasos concretos hacia la solución. En colaboración estrecha con el equipo de trabajo, hemos comenzado a elaborar un backlog de productos. Este paso inicial nos ha brindado la oportunidad de priorizar las historias que abordaremos en los primeros sprints. Centrados en los desafíos identificados, hemos estructurado nuestras tareas de manera estratégica, asegurando que atendamos de forma efectiva y eficiente las necesidades más urgentes y fundamentales. Este proceso de priorización no solo nos proporciona una hoja de ruta clara para nuestros esfuerzos iniciales, sino que también sienta una base sólida para futuras iteraciones y mejoras continuas en el sistema de Historial Financiero. En acuerdo conjunto, estamos definiendo el alcance de nuestro Producto Mínimo Viable (MVP), enfocándonos inicialmente en el primer elemento relacionado con la obtención de información sólida y centralizada. Este primer punto se centra en la Información de Créditos en sus diferentes segmentos, proporcionando datos consolidados y oportunos como único punto de acceso.

### 4.3.14 DEFINICIÓN DE MVP (PRODUCTO MINIMO VIABLE)

Una vez priorizadas las actividades, nos sumergimos en una fase crucial de colaboración con el equipo Scrum. Junto con el equipo, nos dedicamos a determinar con precisión cuál sería nuestro Producto Mínimo Viable (MVP), la piedra angular de nuestro proyecto, diseñado para satisfacer las necesidades fundamentales del Historial Financiero. En este proceso de negociación y deliberación, nos esforzamos por identificar las características esenciales que nos permitirían cumplir con los requisitos establecidos, manteniendo siempre un enfoque ágil y adaptable. Nuestra

---

meta era alcanzar un consenso sólido y claro sobre los aspectos del producto prioritarios para garantizar su viabilidad y utilidad desde el principio. Como resultado del análisis realizado, procedimos a seleccionar nuestro Producto Mínimo Viable, que servirá como punto de partida, tal como se muestra en la **tabla 3**.

### FUNCIONALIDADES

1	Mantener Campos OC del JASIT.	Ingresar con Código Socio.	Filtros de Estado de crédito.	Nombre: Información Financiera.
---	-------------------------------	----------------------------	-------------------------------	---------------------------------

**Tabla 3** Funcionalidad para Producto Mínimo Viable a desarrollar.

## 4.4 DISEÑO DE PROTOTIPOS.

---

Una vez que se tiene la claridad de las funcionalidades que tendrá el aplicativo es necesario el diseño de prototipos los cuales proporcionan una serie de beneficios cruciales que pueden mejorar significativamente el proceso de desarrollo. Ofrecen una perspectiva clara de cómo se anticipa que operen los procesos y flujos de trabajo dentro del sistema, lo que permite identificar de manera temprana, cuellos de botella, áreas de mejora y riesgos potenciales antes de la implementación completa en el entorno de producción. Además, el diseño de prototipos en el contexto de DevOps facilita la realización de pruebas de concepto rápidas, lo que permite validar ideas y enfoques de manera eficiente. Esto promueve una iteración temprana y frecuente, lo que a su vez ayuda a minimizar los costos y riesgos vinculados al desarrollo de software. Estas son áreas clave que exploraré en nuestra investigación para comprender mejor cómo el diseño de prototipos permite optimizar el proceso de desarrollo en el entorno de DevOps.

Asimismo, los prototipos proporcionan una plataforma para obtener feedback temprano y continuo de los stakeholders, incluidos los usuarios finales, los desarrolladores y los equipos de operaciones. Esta retroalimentación es esencial para asegurar que el producto final cumpla con las necesidades y expectativas de todos los participantes. Además, el diseño de prototipos fomentó la colaboración y la alineación entre los equipos de desarrollo, operaciones y otros stakeholders, lo

que facilitó una mayor eficacia en alcanzar los objetivos del proyecto DevOps. En resumen, el diseño de los prototipos fue una herramienta útil para optimizar el entendimiento técnico y facilitar la implementación exitosa de prácticas DevOps. Al visualizar y probar implementaciones antes de la implementación completa, los equipos pudieron identificar y abordar problemas de manera proactiva, lo que conduce a un desarrollo de mayor calidad y más eficiente.

Posteriormente, junto con el equipo de trabajo, utilizamos la herramienta de diseño colaborativo Figma para crear prototipos que representen los resultados esperados por parte de los usuarios al finalizar el producto. Los prototipos se fundamentan en las necesidades y expectativas recopiladas durante el proceso de definición de requisitos. Los resultados de este proceso pueden visualizarse en las **imágenes de la 26 y 27.**

Prototipo de un formulario de consulta de información personal. El formulario está dividido en varias secciones con campos de entrada:

- Número de Identificación
- Código de Socio:
- Tipo de Identificación:
- Número de identificación:
- Código de Socio:
- Apellidos y Nombres
- Fecha de nacimiento
- Edad
- Estado Civil
- Cargas Familiares
- Sucursal
- Estado de Solicitud
- Fecha de Apertura Cuenta
- Cuenta Activa
- Persona Actualizada

En la parte inferior del formulario hay una barra gris con tres íconos: un cuadrado con cuatro pequeños cuadrados, tres puntos verticales y una lupa.

*Imagen 24 Consulta información Personas*

INFORMACIÓN CREDITICIA																
Buscar																
DEUDOR																
Sucursal	Actúa como	Tipo Operación	Producto	N° Operación	Fecha Operación	Fecha vencimiento	Cantidad Solicitada	Saldo	Cuota	N° Cuotas Mora Total	Días Mora Acumulado	N° Cuotas Mora Actual	Días Mora Actual	Clasificación	Estado	
Cuenca	Titular	Crédito	Extraordinario	56890	10/02/2012	10/01/2013	\$ 2.000,00	\$ 0,00	145,50	0	0	0	0	Original	Cancelado	
Deudor Principal:				Cónyuge:				Tipo Garantía:								
Subsegmento				Código Póliza Seguro:				Observación:								
Cuenca	Titular	Crédito	Con Ahorro	96993	22/05/2019	22/11/2021	\$ 10.000,00	\$ 0,00	228,88	0	0	0	0	Original	Cancelado	ver
Cuenca	Cónyuge	Crédito	Línea de Consumo Ordinaria	990117	16/09/2023	16/03/2025	\$ 5.000,00	\$ 2.000,00	308,00	3	20	1	5	Original	Liquidado	ver
Buscar																
GARANTE																
Sucursal	Actúa como	Tipo Operación	Producto	N° Operación	Fecha Operación	Fecha vencimiento	Cantidad Solicitada	Saldo	Cuota	N° Cuotas Mora Total	Días Mora Acumulado	N° Cuotas Mora Actual	Días Mora Actual	Clasificación	Estado	
Cuenca			Extraordinario	56890			\$ 2.000,00		145,50	0			0	Original		ver
Deudor Principal:				Cónyuge:				Tipo Garantía:								
Subsegmento				Código Póliza Seguro:				Observación:								
Buscar																
CARTA DE GARANTÍA																
Sucursal	Código Garantía	Fecha Garantía	Beneficiario	RUC beneficiario	Nombre contrato	Monto	Fecha inicio	Fecha vencimiento	Días vigencia	Tipo garantía	Estado					
Loja	3924	21/02/2024	Agencia de Regulación y Control de las Telecomunicaciones ARCOTEL	1768181900001	Registro de Servicios de acceso a Internet y concesión de uso y explotación de frecuencias no esenciales del espectro radioeléctrico	\$425,00	28/02/2024	14/03/2025	380	Fiel cumplimiento	Ejecutada					
Buscar																
GARANTÍA HIPOTECARIA																
Cod. Valoración	Código bien	N° Operación	Socio	Tipo Bien	Fecha Aprobación	Propietario	Cédula	Val. Tot. Realización	Saldo Hipotecable	Estado Bien	Obligación Financiera					
28023	21218	486526	486526	EDIFICACIÓN	11-09-2018	AGUILAR RIERA EDUARDO GUILLERMO	0105189146	\$ 75.726,95	69.668,70	Activa	SI	ver				
GARANTÍA FIDUCIARIA																
Cod. Valoración	Código bien	N° Operación	Socio	Tipo Bien	Fecha Aprobación	Propietario	Cédula	Val. Tot. Realización	Saldo Hipotecable	Estado Bien	Obligación Financiera					
28023	21218	486526	486526	EDIFICACIÓN	11-09-2018	AGUILAR RIERA EDUARDO GUILLERMO	0105189146	\$ 75.726,95	69.668,70	Activa	SI	ver				
Código	Cod. Valo	Cod. Bien	Valor Debe	# Operación	Suc. Con.	Estado	Cod. Tip. Doc	Valor Haber	Cuenta Código	Cred. Debi	Tipo_Garantía	Fecha_Registro				
1718959	0	0	10.000	13806	41	S	2	0	73.0110.01.02	D	1	19/02/2024	ver			

Imagen 25 Información crediticia de socio

## 4.5 CONFIGURACIÓN DE ENTORNOS DE DESARROLLO.

### 4.5.1 STACK TECNOLÓGICO.

Para desarrollar el Producto Mínimo Viable (MVP), se realiza un análisis exhaustivo orientado a seleccionar las herramientas más apropiadas para el proyecto. Dichos recursos, que se detallan en el stack tecnológico (conjunto de tecnologías,

herramientas o lenguajes de programación que se emplean juntos al momento de desarrollar aplicaciones) ilustrado en la imagen 28, son seleccionadas en función de su escalabilidad, facilidad de integración, soporte técnico disponible y su alineación con los requisitos específicos del sistema. Este proceso asegura que las tecnologías elegidas no solo contribuyan efectivamente al cumplimiento de los objetivos del MVP, sino que también faciliten su mantenimiento y evolución futura.

HISTORIAI FINANCIERO	
GESTIÓN DE PROYECTOS Y METODOLOGÍAS AGILES	
ARQUITECTURA	
BASE DE DATOS	
HERRAMIENTAS DE DESARROLLO	
FRONT-END	
BACK-END	
DEVOPS	
INFRAESTRUCTURA	

Imagen 26 Stack Tecnológico

Estas decisiones garantizan un entorno robusto y coherente para la creación de software, facilitando la integración eficiente entre los diferentes componentes y simplificando el proceso de entrega y despliegue continuo.

En cuanto a la selección de los entornos de desarrollo (IDE) para el back-end y el front-end, así como para las herramientas de prueba de servicios REST, se sigue la arquitectura definida para la cooperativa, basada en metodologías, principios y buenas prácticas recomendadas.

---

Adicionalmente, los entornos de desarrollo se configuran conforme a las mejores prácticas de DevOps, seleccionando las herramientas de automatización apropiadas para optimizar tanto la entrega continua como el despliegue continuo del MVP. Esto permite validar de manera temprana el Producto Mínimo Viable.

## 4.5.2 ENFOQUE AGIL - LEAN INCEPTION.

### **Enfoque Metodológico: Lean Inception (Un Inicio Ágil y Centrado en el Valor).**

Lean Inception es un enfoque ágil creado para poner en marcha proyectos de desarrollo de manera eficaz, centrada en el valor. Esta técnica se caracteriza por su capacidad para alinear rápidamente a todo el equipo en torno a una visión común del producto, priorizando las funcionalidades que generan mayor valor para el cliente y minimizando el desperdicio (Caroli, Lean Inception Creando Conversaciones hacia un producto exitoso., 2018).

### **Objetivos de Lean Inception:**

- **Alinear al equipo:** Facilita la creación de un entendimiento compartido sobre los objetivos del proyecto, el alcance y las expectativas, lo que es crucial para el éxito del desarrollo.
- **Priorizar funcionalidades:** Permite identificar las características más importantes del producto desde la perspectiva del cliente, asegurando que se enfoquen los esfuerzos en lo que realmente importa.
- **Minimizar el desperdicio:** Contribuye a la reducción de trabajo innecesario y a la optimización de recursos, asegurando que cada tarea y esfuerzo esté alineado con las necesidades del cliente.

- 
- Crear un backlog inicial: Establece una base sólida para el Product Backlog en metodologías como Scrum o Kanban, facilitando la planificación y organización del trabajo futuro.
  - Al implementar Lean Inception en el desarrollo del MVP, se garantiza que el equipo esté preparado para abordar el proyecto de manera ágil y enfocada en el valor, lo que contribuye a una mayor eficacia en el proceso de desarrollo de software.

### 4.5.3 METODOLOGÍA AGIL - SCRUM.

Para el desarrollo del MVP (Minimum Viable Product), se ha seleccionado Scrum como el marco de trabajo ágil. Scrum se utiliza de manera generalizada en proyectos de software complejos y en continua evolución, gracias a su énfasis en la colaboración, la flexibilidad y la entrega constante de valor al cliente.

#### **¿Cómo se aplica Scrum en la creación del MVP?**

Scrum organiza el trabajo en ciclos breves llamados sprints, los cuales, en este caso tendrán una duración de entre dos y cuatro semanas. Cada sprint tendrá como objetivo entregar un incremento funcional del producto, lo que permitirá evaluar y validar el progreso del MVP de forma continua.

Al iniciar cada sprint, se definirán claramente las tareas o funcionalidades que serán abordadas, priorizando aquellas que aporten el mayor valor al desarrollo del MVP.

Al final de cada sprint, el equipo revisará el avance, ajustará el plan si es necesario y preparará el siguiente ciclo de trabajo.

---

## Pilares de Scrum en la Implementación del MVP

- **Transparencia:** Todos los aspectos del desarrollo del MVP serán visibles tanto para el equipo como para las partes interesadas, permitiendo que se tomen decisiones informadas durante el proceso.
- **Inspección:** Al final de cada sprint, se inspeccionará el incremento del producto para identificar posibles mejoras. Esta evaluación periódica garantiza que el MVP se ajuste a los objetivos planteados y que el equipo esté alineado con las expectativas del proyecto.
- **Adaptación:** Scrum permite una rápida adaptación a los cambios. Si durante el desarrollo del MVP surgen nuevas necesidades o se identifican áreas de mejora, el plan de trabajo se ajustará de manera ágil, asegurando que el producto evolucione de acuerdo con los requerimientos del proyecto y de los usuarios.

### 4.5.4 HERRAMIENTA GESTIÓN DE PROYECTOS CLICKUP.

En el desarrollo del MVP y la gestión del equipo, se ha optado por utilizar ClickUp como la herramienta principal de gestión de proyectos. ClickUp es una plataforma en línea altamente flexible que ha ganado reconocimiento en los últimos años gracias a su habilidad para ajustarse a las necesidades de equipos de diversos tamaños y sectores.

Entre las principales razones para seleccionar ClickUp se encuentran su versatilidad y su amplia gama de funcionalidades. Estas incluyen la gestión de tareas, la asignación de responsabilidades, la creación de diagramas de Gantt, la monitorización del avance en tiempo real y la integración con múltiples herramientas, lo que facilita una visión centralizada del proyecto. Además, su

---

enfoque colaborativo permite que todos los integrantes del equipo se mantengan alineados con respecto a los objetivos y el estado del desarrollo del MVP, mejorando la eficiencia del trabajo.

Gracias a estas características, ClickUp no solo mejora la organización y planificación de los sprints dentro del marco de trabajo ágil Scrum, sino que también facilita la coordinación de todo el equipo en cada etapa del ciclo de vida de la construcción del sistema.

#### 4.5.5 STRUCTURIZR.

Structurizr es una herramienta de modelado y documentación de arquitectura que emplea diagramas C4 (Context, Containers, Components, and Code). Esta metodología proporciona una forma clara y estructurada de visualizar la arquitectura de software, facilitando la comprensión y comunicación entre los integrantes del equipo y el resto de las partes involucradas.

Aunque Structurizr se accede generalmente de forma online, recientemente se ha lanzado una versión "lite" que permite su ejecución local mediante un contenedor. Esta opción incrementa la flexibilidad en el uso de la herramienta, permitiendo a los equipos colaborar de manera independiente y adaptarse a sus entornos de desarrollo específicos.

Al integrar Structurizr en el stack tecnológico del proyecto, se asegura una documentación adecuada de la arquitectura, lo que resulta crucial para el desarrollo eficiente y para mantener una visión coherente a lo largo del ciclo de vida del software.

#### 4.5.6 POSTGRESQL.

PostgreSQL, comúnmente pronunciado como "Post-GRES", es un sistema de gestión de bases de datos relacionales (SGBDR) open source. Destacado por su fiabilidad, flexibilidad y cumplimiento con estándares técnicos abiertos, PostgreSQL se posiciona como una de las bases de datos más avanzadas y populares a nivel mundial.

---

Este SGBDR ofrece una amplia gama de características, incluyendo soporte para transacciones ACID, un sólido sistema de extensibilidad, y la capacidad de manejar grandes volúmenes de datos. Además, su arquitectura permite la implementación de consultas complejas y la integración con múltiples lenguajes de programación, lo que lo convierte en una opción ideal para aplicaciones que requieren un manejo eficiente y robusto de datos.

La elección de PostgreSQL en el stack tecnológico del proyecto asegura un rendimiento óptimo, escalabilidad y seguridad, facilitando así el desarrollo de aplicaciones que se alineen con los requisitos actuales y futuros de la arquitectura de software.

#### 4.5.7 VISUAL STUDIO CODE.

Es un editor de texto gratuito y de código abierto creado por Microsoft, que se ha posicionado como una de las herramientas más utilizadas entre los programadores. Su reconocida ligereza y velocidad, combinadas con una amplia gama de funcionalidades, lo convierten en un entorno de desarrollo altamente personalizable.

Este editor soporta múltiples lenguajes de programación y proporciona características avanzadas como autocompletado de código, resaltado de sintaxis, depuración integrada y un terminal incorporado. Además, su amplia gama de extensiones ofrece a los desarrolladores la posibilidad de adaptar el entorno de trabajo a sus necesidades particulares, lo que mejora la productividad y facilita la colaboración en proyectos de software.

La inclusión de Visual Studio Code en el stack tecnológico del proyecto asegura un desarrollo ágil y eficiente, ofreciendo un entorno que favorece la implementación de buenas prácticas de programación y el manejo de complejidades inherentes al desarrollo de (Microsoft, s.f.).

---

## 4.5.8 ECLIPSE.

En el marco del desarrollo del MVP, se ha seleccionado Eclipse como el entorno de desarrollo integrado (IDE) principal. Eclipse es una herramienta de código abierto ampliamente reconocida y utilizada por programadores en todo el mundo debido a su versatilidad y robustez.

Este IDE actúa como una plataforma fundamental para desarrollar software, permitiendo a los desarrolladores construir y gestionar proyectos en una gran variedad de lenguajes de programación, como Python, Java, C++, entre otros. Su arquitectura modular facilita integrar múltiples herramientas, así como plugins, lo cual facilita adaptar el entorno de acuerdo con las necesidades particulares del proyecto.

Eclipse no solo mejora la productividad del desarrollo al ofrecer funciones como la edición de código, gestionar versiones, permite depurar y también promueve la colaboración entre los integrantes del equipo, al brindar herramientas que simplifican la integración y el control del código fuente. Gracias a su amplia comunidad y a la disponibilidad de recursos, Eclipse se convierte en una opción perfecta para el desarrollo eficiente y organizado del MVP en el contexto de la arquitectura de software.

## 4.5.9 REACTJS.

Como siguiente paso para desarrollar el front end, se decide utilizar ReactJS debido a sus numerosas ventajas y a su aceptación en los equipos de desarrollo de aplicaciones web. ReactJS es una biblioteca de JavaScript de código abierto, mantenida por Facebook, utilizado para crear interfaces de usuario dinámicas e interactivas.

Algunas de las ventajas clave de ReactJS incluyen:

**Componentización:** ReactJS utiliza un enfoque basado en componentes, lo cual simplifica la división de la interfaz de usuario en componentes modulares y reutilizables. Esto facilita la construcción, mantenimiento y reutilización del código.

---

**Virtual DOM:** ReactJS emplea un Virtual DOM para mejorar el rendimiento de las actualizaciones de la interfaz de usuario. En lugar de modificar directamente el DOM (Document Object Model Virtual es una estructura que representa la página web como un árbol de elementos HTML) completo cada vez que el estado de la aplicación se modifica, React compara el Virtual DOM con el DOM real y solo actualiza los elementos que han cambiado, o que se traduce en un mayor rendimiento y eficiencia.

**JSX:** ReactJS utiliza JSX (JavaScript XML), una extensión de sintaxis que permite incorporar código HTML dentro de JavaScript. Esto simplifica la creación de componentes React y aumenta la legibilidad del código.

**Reactividad:** ReactJS simplifica el manejo del estado de la aplicación y las actualizaciones de la interfaz de usuario mediante su modelo de programación reactivo. Los cambios en el estado de la aplicación generan actualizaciones automáticas en la interfaz de usuario, lo que simplifica el desarrollo y disminuye la probabilidad de errores.

**Amplia comunidad y ecosistema:** ReactJS tiene una extensa comunidad de desarrolladores y una variada colección de bibliotecas y herramientas adicionales, lo que simplifica la resolución de problemas, el aprendizaje y la ampliación de funcionalidades

En resumen, al elegir ReactJS para el desarrollo de nuestro front end, estamos optando por una tecnología sólida, eficiente y altamente adaptable que nos permitirá crear interfaces de usuario modernas y responsivas para nuestra aplicación.

#### 4.5.10 SPRING BOOT.

Como siguiente paso y de vital importancia, hemos seleccionado Spring Boot para el desarrollar nuestro backend, el cual es un framework de código abierto que permite la creación rápida y sencilla de aplicaciones Java. Spring Boot resalta por su

---

capacidad para facilitar el desarrollo., reducir la configuración necesaria y aumentar la productividad del equipo.

Entre las ventajas clave de Spring Boot se encuentran:

**Facilidad de configuración:** Spring Boot nos ofrece un enfoque de "opinión sobre la configuración", el cual ofrece ajustes predeterminados sensatos para que el desarrollador pueda comenzar rápidamente sin tener que configurar cada detalle.

**Integración sencilla:** Permite la integración fácil y rápida con una variedad de tecnologías y marcos de trabajo, como Spring Framework, Hibernate, JPA.

**Despliegue rápido:** Spring Boot incluye un servidor integrado, lo que facilita el despliegue de la aplicación sin la necesidad de configurar un servidor de aplicaciones externo.

**Actuación y escalabilidad:** Spring Boot proporciona características que ayudan a mejorar el rendimiento de la aplicación y permite escalarla fácilmente según las necesidades del negocio.

**Amplia comunidad y soporte:** Al ser parte del ecosistema de Spring Framework, Spring Boot cuenta con una amplia comunidad de desarrolladores y una documentación completa, lo que facilita el aprendizaje y la resolución de problemas.

En resumen, Spring Boot nos brinda las herramientas necesarias para desarrollar un backend robusto, eficiente y escalable, lo que nos permite enfocarnos en la implementación de nuestras soluciones sin preocuparnos por la complejidad de la infraestructura subyacente y centrando nuestra atención en el proceso de desarrollo nuestro MVP.

#### 4.5.11 OPENAPI.

Open API es un lenguaje de especificación estandarizado que permite describir las APIs de manera independiente del lenguaje de programación utilizado. Esta

---

característica facilita la transferencia de conocimiento entre proveedores y consumidores de APIs. A través de Open API, se pueden detallar aspectos fundamentales como el nombre del servicio, la URL del endpoint, los parámetros requeridos, los métodos disponibles, el tipo de respuesta y sus respectivas descripciones (Foundation, s.f.) .

#### 4.5.12 MAVEN.

Maven es una herramienta diseñada para gestionar y comprender proyectos de software en Java que proporciona un marco de trabajo para el desarrollo, la construcción y la gestión de proyectos. Sus diversas funcionalidades incluyen la gestión de dependencias, la construcción del proyecto, la estandarización, el uso de plugins y la integración continua (Apache, s.f.) lo que lo convierte en una poderosa herramienta a la hora la creación de los proyectos de software.

#### 4.5.13 JAVA.

Es una plataforma tecnológica y un lenguaje de programación creado por Sun Microsystems en 1995. Desde su lanzamiento, ha evolucionado hasta convertirse en un pilar esencial para numerosos servicios y aplicaciones digitales actuales, así como de productos innovadores diseñados para el futuro. Aunque muchas aplicaciones Java modernas integran el entorno de ejecución y la propia aplicación, todavía hay aplicaciones y sitios web que requieren la instalación de Java para escritorio. El sitio web [Java.com](http://Java.com) está destinado a consumidores que necesitan Java para sus aplicaciones de escritorio, especialmente aquellas compatibles con Java 8. Los desarrolladores y usuarios interesados en aprender Java pueden visitar [dev.java](http://dev.java), mientras que los usuarios comerciales deben consultar [oracle.com/java](http://oracle.com/java) para obtener más información (Oracle, s.f.).

---

#### 4.5.14 GITLAB.

GitLab es la herramienta ideal para desarrollar MVPs, ya que permite a los equipos adoptar las mejores prácticas de DevOps. Al ofrecer una suite integral de funcionalidades que abarcan desde la gestión del repositorio de código hasta la implementación automatizada. Su enfoque integrado permite al equipo gestionar todo el ciclo de vida del desarrollo de software en un solo lugar, lo que facilita la configuración y el mantenimiento de los pipelines de CI/CD (GitLab, 2024).

Una de las ventajas clave de GitLab es su enfoque en la automatización, así como la integración continua. Los pipelines de CI/CD se pueden definir y configurar fácilmente utilizando un archivo YAML, lo que permite a los equipos especificar todas las etapas necesarias para la construcción, prueba y despliegue de su aplicación de manera transparente y reproducible. Además, GitLab proporciona una amplia gama de corredores y servicios integrados para ejecutar estas etapas, lo que garantiza una ejecución rápida y eficiente de los pipelines (GitLab, 2024).

Otra característica destacada de GitLab es su enfoque en la seguridad y el cumplimiento. La plataforma ofrece capacidades avanzadas de escaneo de seguridad y análisis estático de código, lo que permite a los equipos identificar y corregir vulnerabilidades en su código de manera proactiva. Además, GitLab ofrece características de cumplimiento integradas, como la gestión de acceso basada en roles y la auditoría de cambios, que ayudan a garantizar que los pipelines de CI/CD cumplan con los estándares de seguridad y regulaciones aplicables (GitLab, 2024).

En resumen, GitLab es una herramienta integral y potente para la implementación de prácticas de CI/CD en el desarrollo de software. Su enfoque integrado, su énfasis

---

---

en la automatización y la seguridad, y su amplia gama de características lo convierten en una opción sólida para cualquier equipo que busque mejorar su proceso de desarrollo y despliegue de software. En nuestra investigación, exploraremos en detalle cómo aprovechar al máximo las capacidades de GitLab para optimizar nuestros pipelines de CI/CD y optimizar la eficiencia y calidad de nuestro desarrollo de software.

#### 4.5.15 DOCKER.

una plataforma de software que facilita la creación, implementación y administración de aplicaciones en contenedores. Los contenedores son entornos portátiles y livianos que contienen todas las dependencias necesarias para que una aplicación funcione correctamente, garantizando su funcionamiento consistente en diferentes entornos, ya sea en máquinas locales, servidores o en la nube.

Docker optimiza la virtualización a nivel del sistema operativo, permitiendo la ejecución aislada de múltiples contenedores en un mismo sistema. Esto no solo mejora el uso eficiente de los recursos, sino que también favorece la escalabilidad y la eficiencia en el desarrollo y la implementación de aplicaciones.

Una de sus principales ventajas es la capacidad de simplificar la gestión de arquitecturas basadas en microservicios, permitiendo desplegar y escalar de forma independiente cada componente de la aplicación (Inc, s.f.).

#### 4.5.16 DOCKER COMPOSE

Se presenta un cliente de Docker que permite trabajar con aplicaciones compuestas por un conjunto de contenedores. Esta herramienta simplifica la gestión y orquestación de contenedores, optimizando el desarrollo, mantenimiento y despliegue de aplicaciones fundamentadas en microservicios.

#### 4.5.17 GOOGLE CLOUD PLATAFORM.

Google Cloud Platform (GCP) es un conjunto integral de servicios en la nube ofrecido por Google que te permite desarrollar, implementar y escalar aplicaciones

---

de forma ágil y eficiente. Es como tener un centro de datos virtual a tu disposición, pero sin la molestia de administrar la infraestructura física (Google, s.f.).

#### 4.5.18 GOOGLE CLOUD RUN.

Es un servicio de computación completamente administrado sin servidor ofrecido por Google Cloud Platform. El cual permite ejecutar contenedores de forma escalable y sin tener que preocuparte por la gestión de la infraestructura subyacente. Es decir, tú te encargas de tu código y Google se encarga de todo lo demás: escalado automático, balanceo de carga, administración de recursos (Google, s.f.).

#### 4.5.19 GOOGLE CLOUD SQL.

Google Cloud SQL es un servicio totalmente administrado que permite ejecutar instancias de bases de datos relacionales ampliamente utilizadas, como MySQL, PostgreSQL y SQL Server, en la nube de Google. Esto facilita la creación, administración y escalabilidad de tus bases de datos sin necesidad de gestionar la infraestructura subyacente (Google, s.f.).

### 4.6 CREACIÓN DE CUENTAS Y CONFIGURACION DE HERRAMIENTA DE CONTROL DE VERSIONES.

En el desarrollo de software, la creación de cuentas y la configuración de un sistema de gestión de versiones son pasos esenciales que garantizan un manejo eficaz del código y promueven la colaboración entre los integrantes del equipo de trabajo. Para lograrlo se implementan las siguientes acciones:

Creación de cuentas: Se asignan cuentas en la plataforma de control de versiones, en este caso GitLab, a cada desarrollador como se puede ver en la imagen 29. Esto asegura que cada miembro del equipo tenga acceso al repositorio del código fuente, facilitando la colaboración en el desarrollo del proyecto.

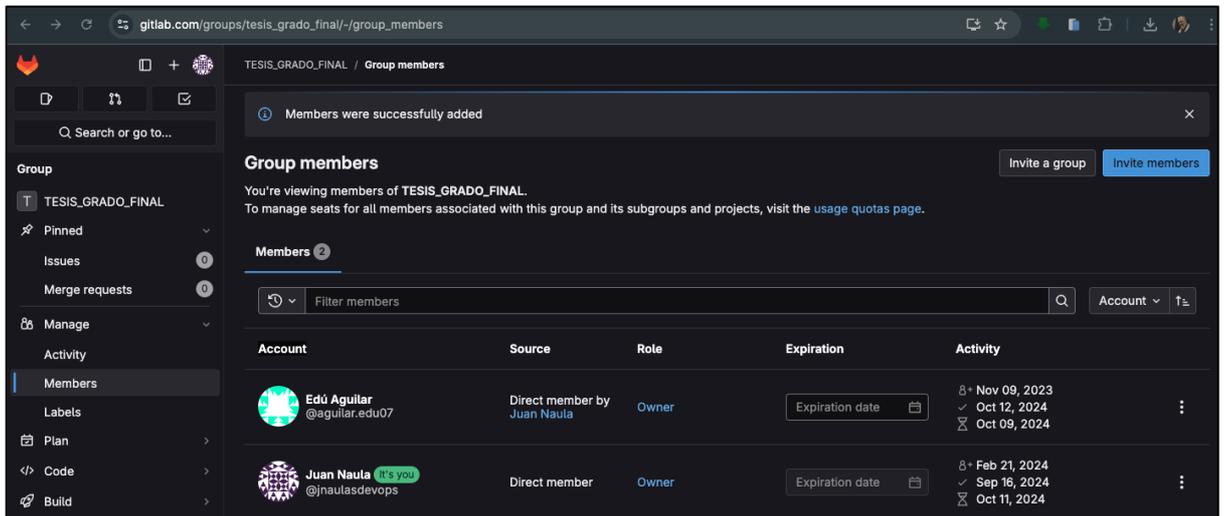
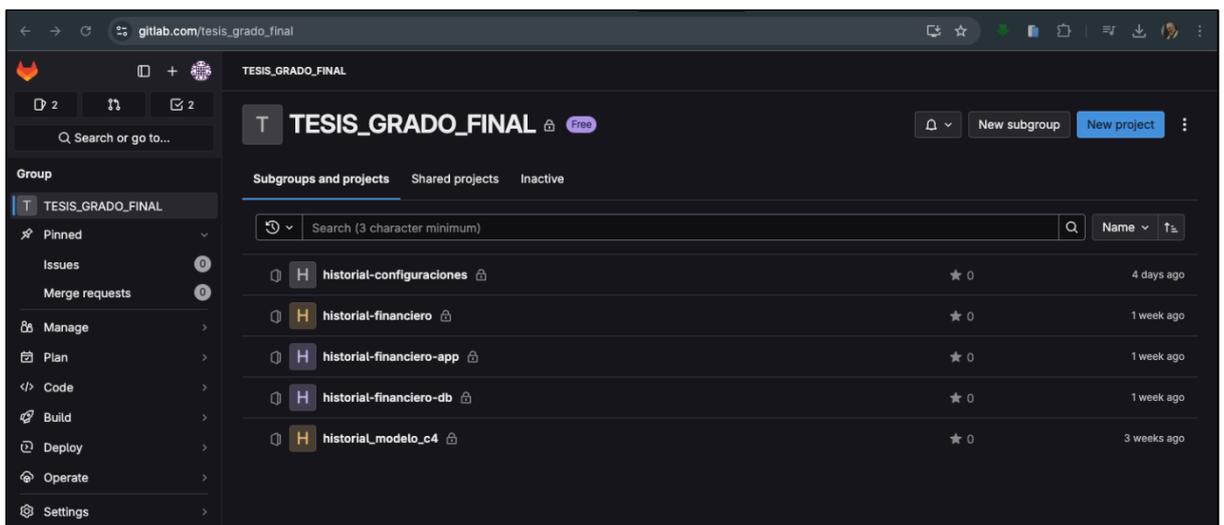


Imagen 27 Creación cuentas

**Configuración del repositorio:** Se establece un repositorio donde se alojará el código fuente. La estructura del repositorio se organiza de manera clara y funcional, definiendo la disposición de carpetas y archivos, lo que facilita la navegación y el mantenimiento del proyecto como se puede ver en la imagen 30, donde se definen los repositorios para back-end, front-end, base de datos, modelo c4 y configuraciones.



**Implementación de estrategias de branching:** Se adoptan estrategias de gestión de ramas (branching) que permiten desarrollar nuevas características y corregir errores de forma ordenada. Entre las ramas configuradas se incluyen las de development, main como se puede ver en la imagen 31.

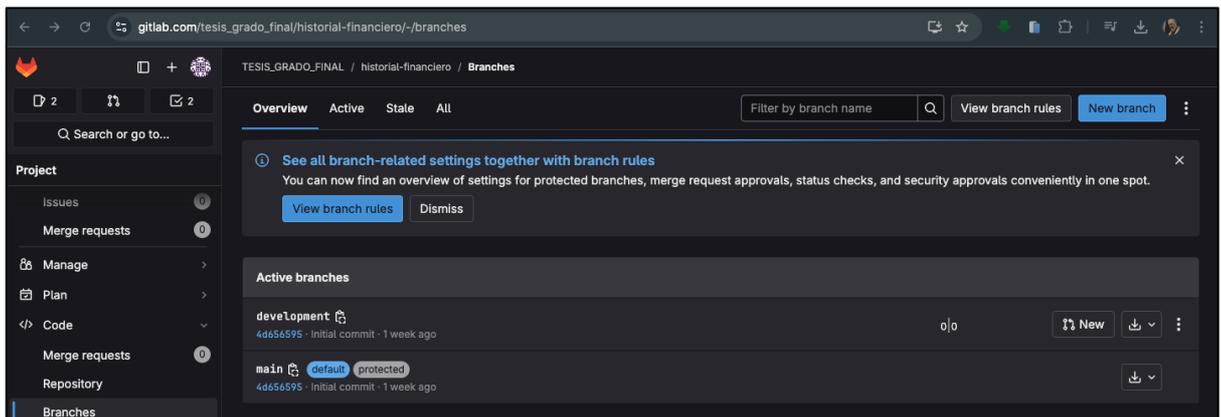


Imagen 29 Ramas proyectos gitlab

De igual manera dentro del control de versiones, se implementó la semántica de Git con el objetivo de gestionar de manera estructurada y eficiente los cambios en el código fuente del proyecto. Esta semántica incluye el uso de convenciones claras para los mensajes de commit, siguiendo un formato preestablecido que permite identificar rápidamente el propósito de cada cambio, ya sea una nueva funcionalidad, una corrección de error o una optimización, etc. Entre las etiquetas más utilizadas se encuentran:

**feat:** Para identificar commits que añaden una nueva funcionalidad al software o mejoran una existente.

**fix:** Utilizada cuando el commit corrige un error en el código de la aplicación.

**docs:** Se emplea cuando se añade o modifica la documentación del proyecto.

**style:** Para aquellos cambios que afectan el formato del código, como ajustes de estilo (por ejemplo, la corrección de paréntesis, llaves, etc.) sin alterar la lógica de producción.

---

**refactor:** Usada cuando se modifica el código de producción sin agregar nuevas funcionalidades, por ejemplo, al renombrar variables o simplificar un método.

**test:** Esta etiqueta se aplica cuando se añaden o modifican pruebas, sin impactar directamente el código de producción.

**chore:** Se utiliza para tareas rutinarias o de mantenimiento, como la actualización de dependencias o la modificación de archivos de configuración, sin alterar el código de producción.

Esta implementación de la semántica de Git dentro del control de versiones no solo mejora la trazabilidad y organización del proyecto, sino que también optimiza la colaboración entre los desarrolladores, asegurando que el código se mantenga en un estado coherente y fácilmente desplegable.

Configuración de permisos: Se asignan permisos adecuados a cada miembro del equipo, asegurando que cada usuario tenga acceso a las funciones necesarias dentro del repositorio, protegiendo al mismo tiempo las áreas críticas del código fuente.

## 4.7 SELECCIÓN DE IDE DE DESARROLLO BACK-END Y FRONT-END.

Para desarrollar software, la elección de los entornos de desarrollo integrados (IDE) es fundamental para optimizar el proceso de programación y garantizar la productividad de los equipos. Para desarrollar el back-end, se seleccionó Eclipse, un IDE ampliamente utilizado en la industria por su robustez y soporte nativo para proyectos en Java. Eclipse proporciona una gran variedad de herramientas, plugins para facilitar la administración de proyectos de gran escala, así como la integración con otras tecnologías necesarias en el desarrollo del sistema.

Por otro lado, para el front-end, se ha optado por Visual Studio Code, una herramienta ligera y altamente flexible que permite trabajar de manera eficiente con tecnologías web como HTML, CSS, y JavaScript. Visual Studio Code destaca por

---

su amplia gama de extensiones y su habilidad para integrarse con sistemas de control de versiones (Git por ejemplo), lo que lo hace una opción perfecta para el desarrollo de interfaces de usuario.

## 4.8 CONFIGURACION DE AMBIENTES APEGADOS A LAS BUENAS PRACTICAS DE DEVOPS.

Siguiendo las buenas prácticas de DevOps, se busca implementar un cambio de cultura organizacional que promueva una colaboración estrecha, la automatización y la integración continua. En este sentido, se procede con la configuración adecuada de los diferentes ambientes del sistema, los cuales incluyen los entornos de development y main, como se ha explicado en los pasos anteriores.

Una vez configuradas las cuentas en GitLab y organizados los proyectos, se procede con la creación de variables de entorno, como se muestra en la tabla 4, que permiten gestionar de manera segura los datos requeridos para la adecuada realización de los procesos de integración continua. Dichas variables se configuran para garantizar que los parámetros críticos, como claves de acceso o configuraciones específicas, estén disponibles sin comprometer la seguridad del sistema. Además, facilitan la automatización de los procesos, evitando incluir información codificada directamente en los ductos, ya que esta es gestionada a través de las variables de entorno.

Posteriormente, se procede con la creación de los archivos de configuración denominados pipelines, los cuales, en GitLab, se especifican como: `.gitlab-ci.yml`. Estos archivos definen la secuencia de pasos a ejecutar dentro del proceso de integración y entrega continua de software, siendo procesados por los corredores de GitLab. Los pipelines automatizan las fases de compilación, prueba e implementación mediante el uso de Docker, lo que facilita la implementación de un ciclo eficiente de Integración y Entrega Continua (CI/CD). Esta automatización no

---

solo minimiza el riesgo de errores manuales, sino que también optimiza la velocidad y consistencia en el ciclo de vida del software.

Una de las principales ventajas que ofrece la automatización mediante pipelines, junto con el uso de variables de entorno, es la capacidad de gestionar despliegues automáticos directamente desde la configuración del código. Esto permite identificar de forma precisa el ambiente en el cual se está versionando e implementando despliegues dinámicos en distintos entornos, tales como los proyectos de Google Cloud Platform (GCP), que en este caso particular incluyen los entornos de desarrollo y producción.

Como se ha mencionado previamente, el despliegue del sistema se realiza en Google Cloud Platform (GCP), donde se gestionan diversos proyectos que alojan los servicios del sistema. Para el proyecto de historial financiero, se han configurado dos entornos: `historial-financiero-grc-dev` para desarrollo e `historial-financiero-grc-produ` para producción. En ambos entornos se despliega el servicio `historial-financiero-app` (front-end) e `historial-financiero` (back-end) utilizando Cloud Run, que ofrece un entorno escalable y gestionado para ejecutar las aplicaciones en contenedores.

Además, en el entorno de producción dentro del proyecto `historial-financiero-grc-produ`, se utiliza Cloud SQL para la gestión de base de datos, lo que asegura la persistencia y manejo eficiente de la información de los usuarios. Esta arquitectura en la nube basada en Cloud Run y Cloud SQL, permite una escalabilidad dinámica, segura y controlada del proyecto, garantizando que los servicios puedan responder a un crecimiento en la demanda sin comprometer la disponibilidad y el rendimiento.

De esta manera se asegura la estabilidad del sistema, al mismo tiempo que se habilita una escalabilidad flexible, lo que permite ajustar los recursos conforme aumentan las demandas del proyecto y de los usuarios, optimizando el uso de la infraestructura y garantizando una operación continua.

Grupo	Proyecto	Variables	Descripción
TESIS_GRADO_FINAL	historial-financiero-app, historial-financiero, historial-financiero-db	GITLAB_TOKEN	Otorga permisos que permite interactuar con las api de Gitlab
TESIS_GRADO_FINAL	historial-financiero-app, historial-financiero, historial-financiero-db	GCP_REGION	Contiene nombre de ubicación geográfica donde Google alojara nuestros servicios.
TESIS_GRADO_FINAL	historial-financiero-app, historial-financiero, historial-financiero-db	GCP_PROJECT_ID_DEV	Identificador único del proyecto creado en Google Cloud Plataform para development
TESIS_GRADO_FINAL	historial-financiero-app, historial-financiero, historial-financiero-db	GCP_PROJECT_ID_PROD	Identificador único del proyecto creado en Google Cloud Plataform para Production
TESIS_GRADO_FINAL	historial-financiero-app, historial-financiero, historial-financiero-db	GCLOUD_SERVICE_KEY_DEV	Contiene clave del proyecto gcp para development codificado en base 64 que permite conectarse con el servicio de GCP
TESIS_GRADO_FINAL	historial-financiero	GCLOUD_SERVICE_KEY_PROD	Clave del proyecto gcp para producción codificado en base 64 que permite conectarse con el servicio de GCP
TESIS_GRADO_FINAL	historial-financiero-db	DB_NAME	Nombre de la base de datos a deployar.
TESIS_GRADO_FINAL	historial-financiero-db	DB_VERSION	Versión de base de datos para nuestro caso PostgreSQL
TESIS_GRADO_FINAL	historial-financiero-db	INSTANCE_NAME	Contiene nombre de la instancia de BD.
TESIS_GRADO_FINAL	historial-financiero-db	DB_USER_ADM_DEV	Contiene nombre de usuario administrador de la base de datos en ambiente development
TESIS_GRADO_FINAL	historial-financiero-db	DB_USER_APP_DEV	Contiene nombre de usuario de aplicación de la base de datos en ambiente development
TESIS_GRADO_FINAL	historial-financiero-db	DB_USER_ADM_PROD	Contiene nombre de usuario administrador de la base de datos en ambiente producción
TESIS_GRADO_FINAL	historial-financiero-db	DB_USER_APP_PROD	Contiene nombre de usuario de aplicación de la base de datos en ambiente producción
TESIS_GRADO_FINAL	historial-financiero-db	DB_PASSWORD_ADM_DEV	Contiene clave de usuario de acceso a la base de datos en ambiente development como administrador.
TESIS_GRADO_FINAL	historial-financiero-db	DB_PASSWORD_APP_DEV	Contiene clave de usuario de acceso a la base de datos en ambiente development como aplicación.

TESIS_GRADO_FINAL	historial-financiero-db	DB_PASSWORD_ADM_PROD	Clave de acceso a la BD en ambiente productivo con privilegios de administrador.
TESIS_GRADO_FINAL	historial-financiero-db	DB_PASSWORD_APP_PROD	Contiene clave de acceso a la base de datos en ambiente productivo con privilegios de aplicación.
TESIS_GRADO_FINAL	historial-financiero-db	VPC_NAME	Nombre de red virtual privada dentro del proveedor de servicio de GCP.
TESIS_GRADO_FINAL	historial-financiero-db	SUBNET_NAME	Contiene el nombre la subred específica dentro de la VPC.
TESIS_GRADO_FINAL	historial-financiero-db	SUBNET_RANGE	Contiene rango de dirección IP para la subred.
TESIS_GRADO_FINAL	historial-financiero-db	VPC_CONNECTOR_NAME	Contiene conector que permite la comunicación de los servicios sin servidor como Cloud Run.
TESIS_GRADO_FINAL	historial-financiero-db	CONNECTOR_RANGE	Contiene rango de dirección IP para el conector.
TESIS_GRADO_FINAL	historial-financiero	DATASOURCE_DEV	Contiene Data Source de conexión de back-end en ambiente development con la base de datos.
TESIS_GRADO_FINAL	historial-financiero	DATASOURCE_PROD	Contiene Data Source de conexión de back-end en ambiente Production con la base de datos.
TESIS_GRADO_FINAL	historial-financiero-app	URL_BACKEND_DEV	Contiene URL de servicio back-end que permite la comunicación desde el front-end en ambiente development.
TESIS_GRADO_FINAL	historial-financiero-app	URL_BACKEND_PROD	Contiene URL de servicio back-end que permite la comunicación desde el front-end en ambiente productivo.

**Tabla 4** Tabla variables de entorno gitlab para proceso CI/CD.

## 4.9 SELECCIÓN DE HERRAMIENTAS DE AUTOMATIZACIÓN

La selección de herramientas de automatización es un paso crítico para asegurar la eficiencia en el ciclo de desarrollo y entrega de software. Dentro del contexto de DevOps, se adopta un enfoque de integración continua (CI) y entrega continua (CD),

---

lo que posibilita la automatización de tareas recurrentes y optimizar los procesos de compilación, prueba y despliegue.

En este caso, se ha seleccionado GitLab CI/CD como la herramienta principal para la automatización del ciclo de vida del software. GitLab CI/CD proporciona un marco completo que permite definir y gestionar pipelines automatizados mediante archivos de configuración `.gitlab-ci.yml`. Esto permite que, tras cada cambio en el código, se realicen automáticamente las fases de construcción, pruebas y despliegue, garantizando un flujo constante de desarrollo sin intervención manual.

El uso de estas herramientas no solo asegura una mayor velocidad en la entrega de nuevas funcionalidades y correcciones, sino que también mejora la calidad del código al detectar problemas de manera temprana, gracias a las pruebas automatizadas integradas en los pipelines. Además, su integración con Docker y Google Cloud facilita el despliegue y la escalabilidad de los servicios, permitiendo que el sistema responda eficientemente a cambios en la demanda.

La adopción de estas prácticas y herramientas asegura un desarrollo más ágil y fiable, promoviendo la automatización como una parte integral del proceso de entrega continua para lo cual se procedió con la configuración dentro de gitlab como se puede ver en la imagen 32 detallada a continuación.

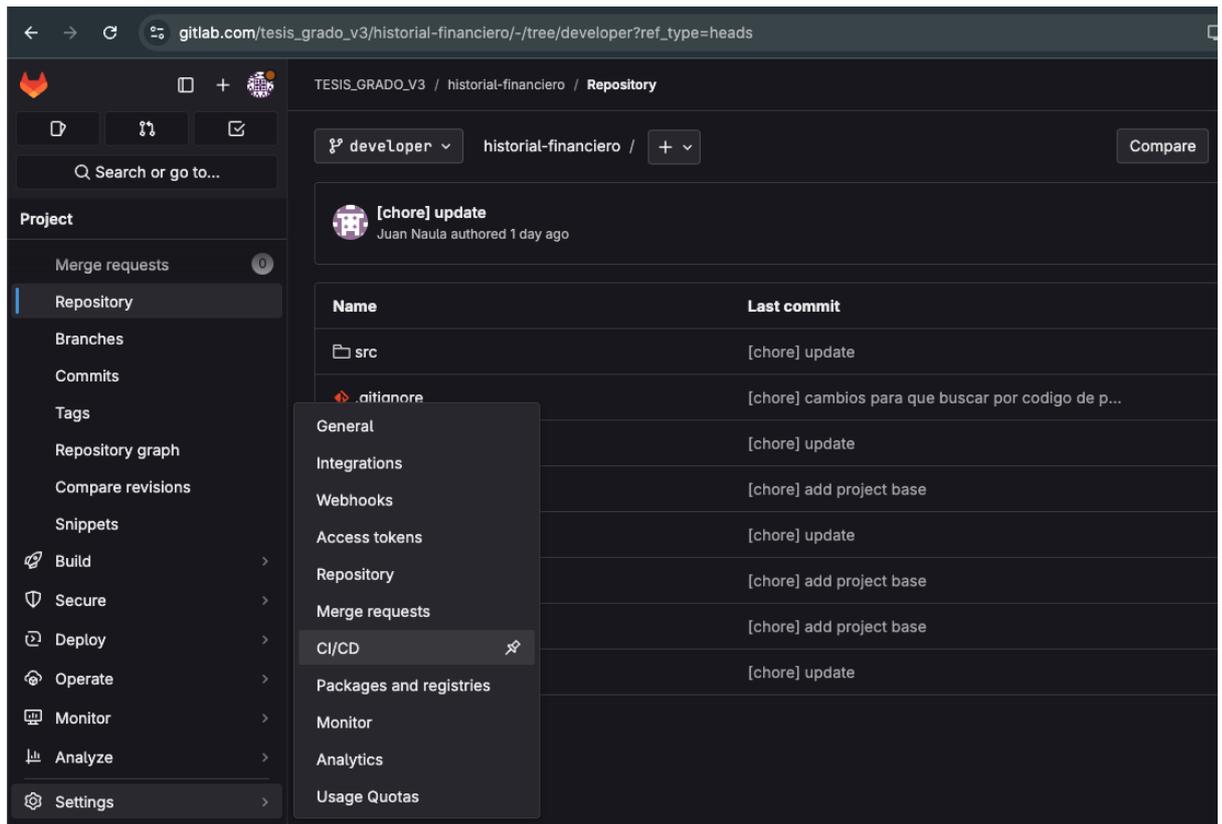


Imagen 30 Configuración herramienta de automatización

## 4.10 CONSTRUCCIÓN DE MVP.

Para la etapa de construcción del MVP, es fundamental organizar el flujo de trabajo dividiéndolo en actividades específicas. Para ello, se emplea el marco de trabajo Scrum, que mejora significativamente el proceso de desarrollo de software en relación con el esquema de trabajo antiguo. Una de las ceremonias clave de Scrum aplicadas en este proceso es el Sprint Planning, la cual asegura que el equipo esté alineado y trabajando de manera efectiva. Esta ceremonia resulta crucial, ya que fomenta la colaboración, la revisión continua y contribuye a mejorar el rendimiento del equipo.

Con el objetivo de garantizar una correcta planificación de las actividades, se implementa la ceremonia de Sprint Planning, donde se reúne al equipo de desarrollo, al Scrum Master y al Product Owner. Durante esta reunión, el equipo elige los elementos del Product Backlog que se trabajarán durante el sprint, según

las prioridades definidas por el Product Owner. Esta permite definir el Incremento que se espera lograr en función de las funcionalidades descritas previamente. Para la administración de estas actividades y el monitoreo de las tareas, se utiliza la herramienta ClickUp como se puede ver en la imagen 33, lo que facilita la organización y control del progreso.

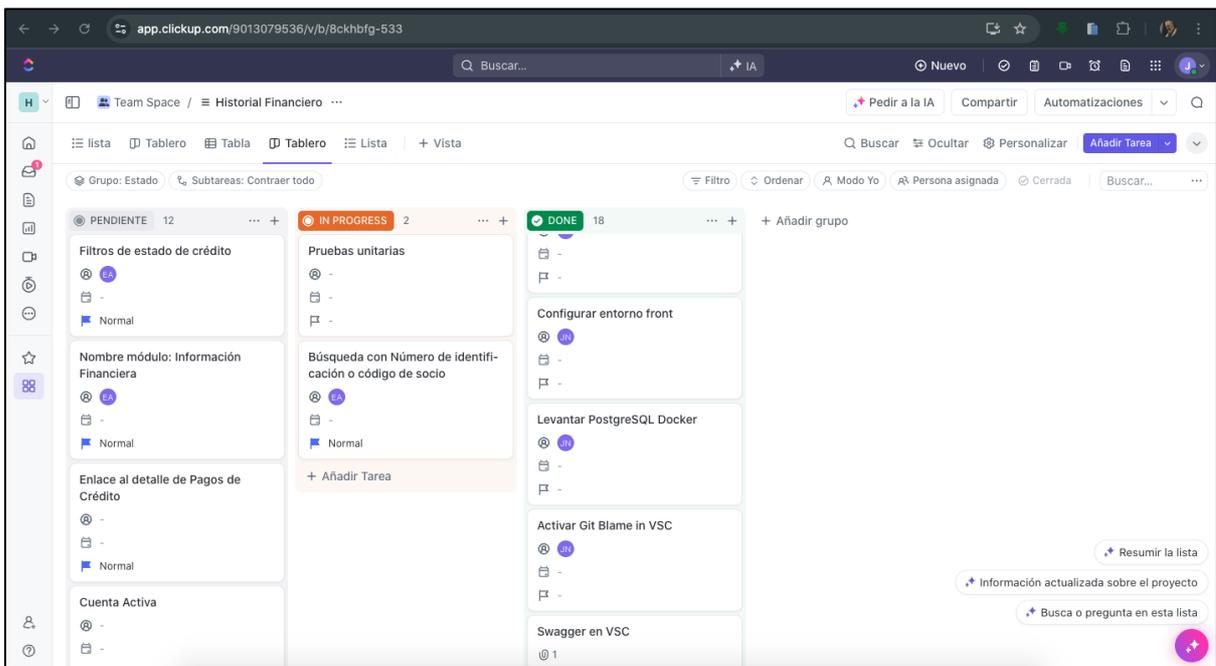


Imagen 31 Planificación ClickUp

En esta fase de planificación, se define el Producto Mínimo Viable (MVP), que consiste en seleccionar las funcionalidades esenciales que deben implementarse para validar el producto en una etapa temprana. El enfoque se basa en metodologías ágiles, lo que permite una construcción iterativa y entregas frecuentes en cortos períodos de tiempo. En cada iteración, se prioriza el valor entregado al usuario final, sin perder de vista los objetivos generales del proyecto.

Posteriormente, una vez configurados los entornos y seleccionadas las herramientas de automatización, se procede con la planificación detallada de las fases de construcción del MVP. Durante este proceso, ClickUp se utiliza para gestionar los distintos sprints definidos en la fase inicial de captura de requisitos, la cual se realizó utilizando la metodología Lean Inception. Esta planificación permite

---

gestionar de forma eficiente las iteraciones del desarrollo, asegurando que el MVP del sistema de historial financiero cumpla con los objetivos de funcionalidad y entrega temprana establecidos.

De esta forma, la arquitectura del sistema se construye de manera ordenada y eficiente, garantizando que las funcionalidades críticas sean entregadas dentro del tiempo establecido, mientras se mantiene la flexibilidad para ajustarse a los cambios en virtud de la retroalimentación del usuario o las necesidades del proyecto.

## 4.11 DESARROLLO DEL MVP.

El desarrollo del MVP se fundamenta en la implementación de las funcionalidades esenciales definidas previamente durante la fase de planificación. Por lo cual, para gestionar eficientemente las etapas de desarrollo, se emplean enfoques ágiles denominados sprints, los cuales permiten generar entregables de manera incremental, obteniendo un producto listo para ser probado. El enfoque se centra en la agilidad y eficiencia, con el objetivo de obtener un prototipo funcional en el menor tiempo posible, garantizando que el sistema esté disponible en su versión mínima para el negocio y que este pueda validar de forma temprana sus diferentes funcionalidades.

De acuerdo con la planificación realizada en ClickUp, se procede con la implementación del código fuente en los diversos componentes del sistema, que incluyen:

La base de datos PostgreSQL.

El back-end, desarrollado con Java Spring Boot.

El front-end, implementado con React JS y JavaScript.

Durante esta fase, se integran las funcionalidades clave del sistema de historial financiero, priorizando la integración continua y la automatización, lo que facilita un desarrollo controlado y permite identificar posibles mejoras o ajustes de manera temprana antes del despliegue final.

Asimismo, se subraya la importancia de las pruebas realizadas en este proceso. Para asegurar la funcionalidad y estabilidad de los componentes, se utiliza Docker Compose para integrar los servicios y ejecutar pruebas locales, antes de proceder con el versionamiento y despliegue en los servidores de Google Cloud Platform (GCP). Como resultado, se logra el despliegue y la integración completa del sistema, tal como se observa en las imágenes 34 a la 37.

```
docker-compose.yml M x  README.md
1 # 2. Define listado de servicios para desplegar la aplicacion. #4 2024-06-28 Juan Naula
2 # Define version #1 2024-06-03 Juan Naula
3 version: '3.8' #4 2024-06-28 Juan Naula
4 # Defina listado de servicios a ejecutar

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
able (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
backend | 2024-10-16T20:30:03.811Z INFO 1 --- [historial-financiero] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFac
backend | 2024-10-16T20:30:04.370Z INFO 1 --- [historial-financiero] [main] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If ap
backend | 2024-10-16T20:30:05.913Z WARN 1 --- [historial-financiero] [main] JpaBaseConfigurations$JpaWebConfiguration : spring.jpa.open-in-view is enabl
backend | 2024-10-16T20:30:06.995Z INFO 1 --- [historial-financiero] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (htt
front | 2024-10-16T20:30:07.026Z INFO 1 --- [historial-financiero] [main] e.f.j.h.HistorialFinancieroApplication : Started HistorialFinancieroAppli
front | Compiled with warnings.
front | Failed to parse source map from '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map' file: Error: ENOENT: no such
front | file or directory, open '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map'
front | Search for the keywords to learn more about each warning.
front | To ignore, add // eslint-disable-next-line to the line before.
front | WARNING in ./node_modules/react-double-scrollbar/dist/DoubleScrollbar.js
front | Module Warning (from ./node_modules/source-map-loader/dist/cjs.js):
front | Failed to parse source map from '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map' file: Error: ENOENT: no such
front | file or directory, open '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map'
front | webpack compiled with 1 warning
front | Compiling...
front | Compiled with warnings.
front | Failed to parse source map from '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map' file: Error: ENOENT: no such
front | file or directory, open '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map'
front | Search for the keywords to learn more about each warning.
front | To ignore, add // eslint-disable-next-line to the line before.
front | WARNING in ./node_modules/react-double-scrollbar/dist/DoubleScrollbar.js
front | Module Warning (from ./node_modules/source-map-loader/dist/cjs.js):
front | Failed to parse source map from '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map' file: Error: ENOENT: no such
front | file or directory, open '/historial-financiero-app/node_modules/react-double-scrollbar/dist/DoubleScrollbar.js.map'
front | webpack compiled with 1 warning
historial_db | 2024-10-16 20:34:53.908 UTC [66] LOG: checkpoint starting: time
historial_db | 2024-10-16 20:34:58.480 UTC [66] LOG: checkpoint complete: wrote 46 buffers (0.3%); 0 WAL file(s) added, 0 removed, 0 recycled; write=4.531 s, sync=0.02
3 s, total=4.572 s; sync files=13, longest=0.020 s, average=0.002 s; distance=240 kB, estimate=240 kB
```

Imagen 32 Ejecución herramienta docker-compose

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
juannaula@Juans-MacBook-Pro historial-configuraciones % docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED   STATUS    PORTS                               NAMES
9efbfa78c48   ja-services/historial-financiero:v01 "java -jar historial..." 4 minutes ago Up 4 minutes 0.0.0.0:8080->8080/tcp   backend
60850f02402e   ja-services/local-postgres-db:v01  "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 0.0.0.0:5432->5432/tcp   historial_db
f4f69328ea76   ja-services/mas-front:1.0         "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 0.0.0.0:3000->3000/tcp, 8080/tcp   front
juannaula@Juans-MacBook-Pro historial-configuraciones %
```

Imagen 33 Resultado contenedores en ejecución

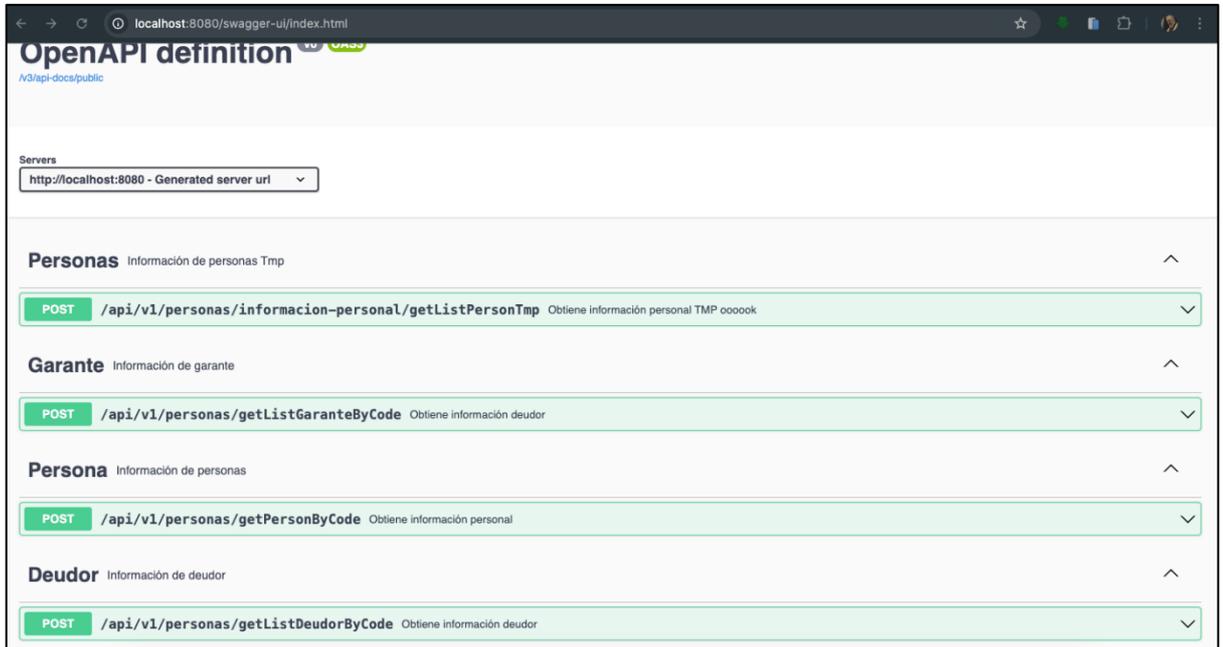


Imagen 34 Despliegue back-end

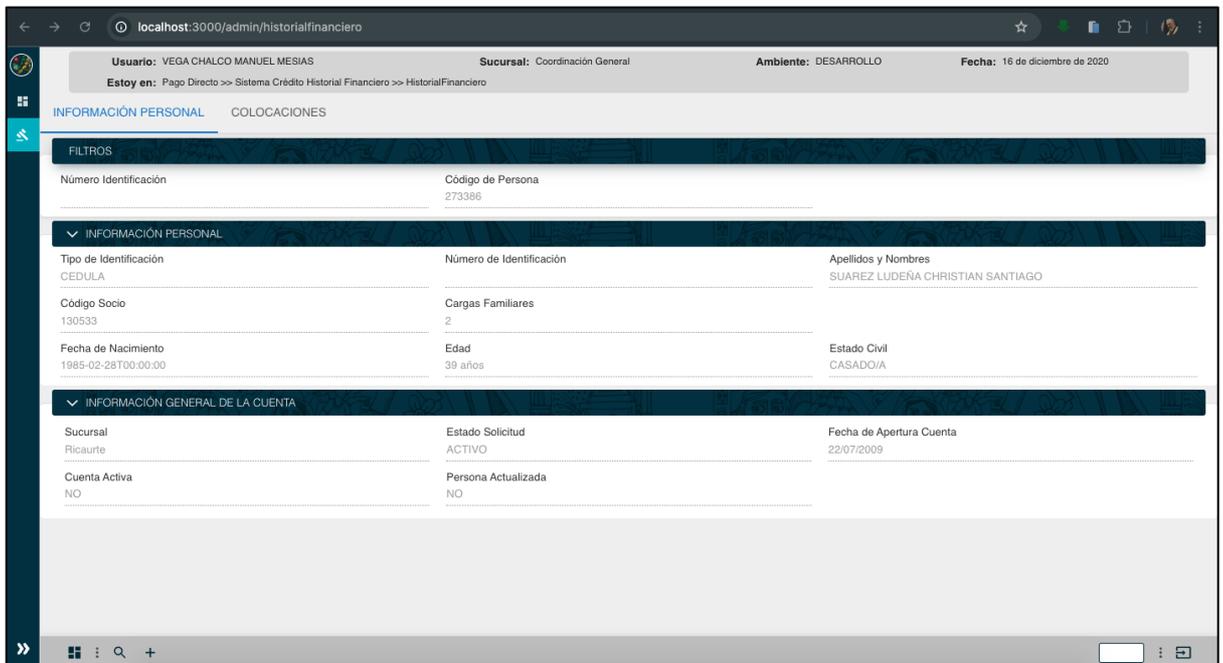


Imagen 35 Despliegue front-end.

## 4.12 DESPLIEGUE DEL MVP.

Una vez finalizada la fase de desarrollo y realizadas las pruebas funcionales a nivel local mediante el uso de la herramienta Docker Compose, se procede con el despliegue de los diferentes artefactos resultantes. Estos artefactos incluyen la base de datos, el back-end y el front-end, que se despliegan en Google Cloud Platform (GCP) utilizando herramientas de integración continua integradas en GitLab, como se puede observar en las **imágenes 38 al 50**.

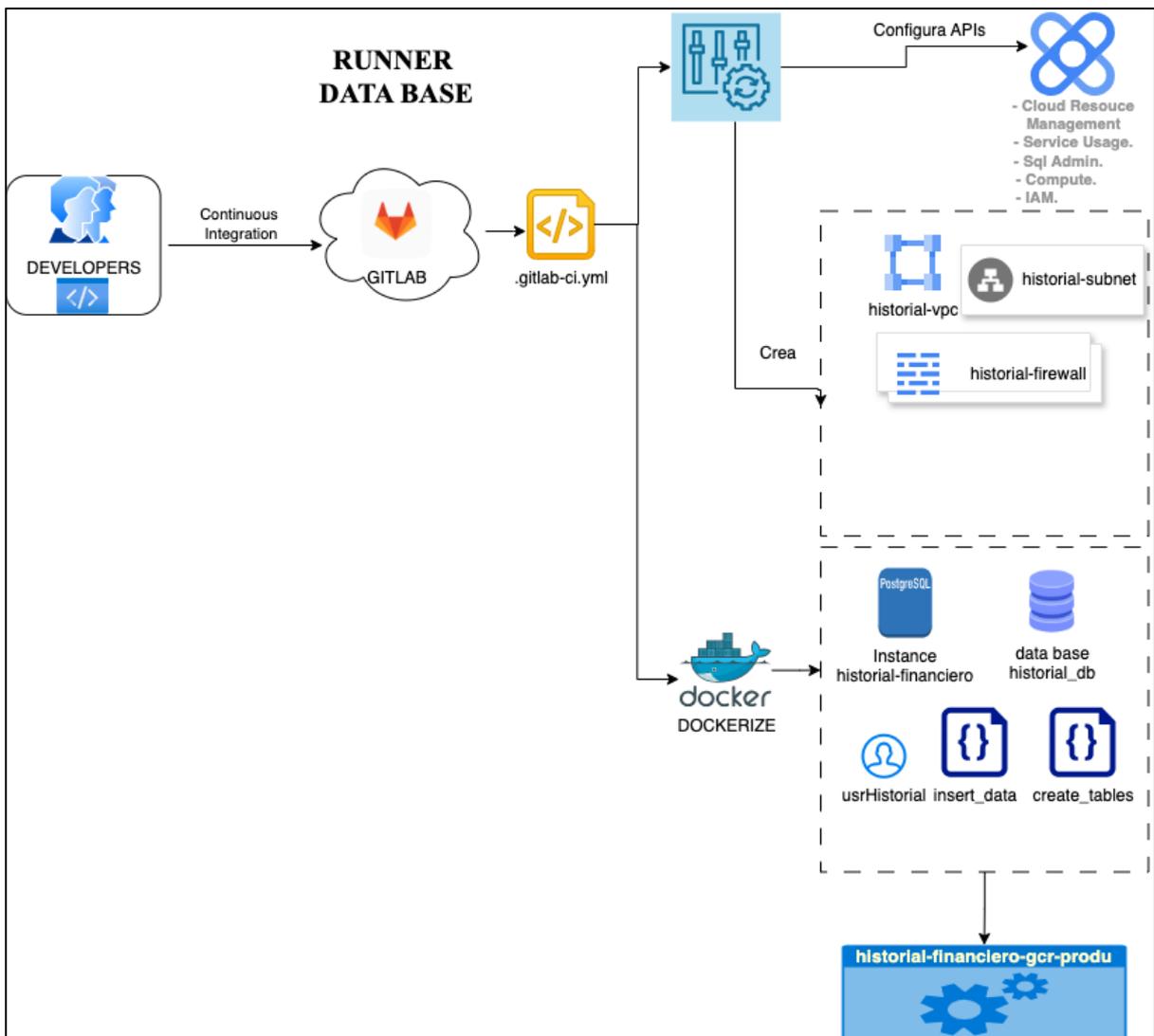


Imagen 36 Despliegue base de datos en Google Cloud Platform servicio Cloud SQL.

La **imagen 38** ilustra el proceso general del despliegue del servicio de base de datos, que comienza con el push realizado por los colaboradores en los repositorios de GitLab. Al detectar cambios, GitLab ejecuta automáticamente el pipeline que contiene una secuencia de pasos iniciada en el stage `setup_environment`. En esta etapa se utiliza una imagen Docker con una versión oficial del SDK de Google Cloud, lo cual permite interactuar con los servicios de GCP, incluyendo la autenticación en la cuenta de servicio y el proyecto (en este caso, `historial-financiero-grc-produ`). Se configuran las APIs necesarias para el funcionamiento adecuado del servicio, como `sqladmin.googleapis.com`, `compute.googleapis.com` y `servicenetworking.googleapis.com`, y se asignan roles como `cloudsql.admin`.

Tras la configuración de las APIs y los roles, se establece la configuración de la red, incluyendo VPC, subredes y VPC Peering, lo cual facilita la conectividad entre los servicios desplegados en Cloud Run y Cloud SQL.

En la siguiente etapa, `deploy_database`, se despliega la base de datos. Primero, se autentica el proyecto para el despliegue, y a continuación se ejecutan una serie de pasos, tales como la creación de instancias, bases de datos, usuarios, y la ejecución de consultas necesarias para el funcionamiento del servicio de base de datos logrando obtener el resultado esperado como se puede ver en las **imágenes 39, 40, 41**.

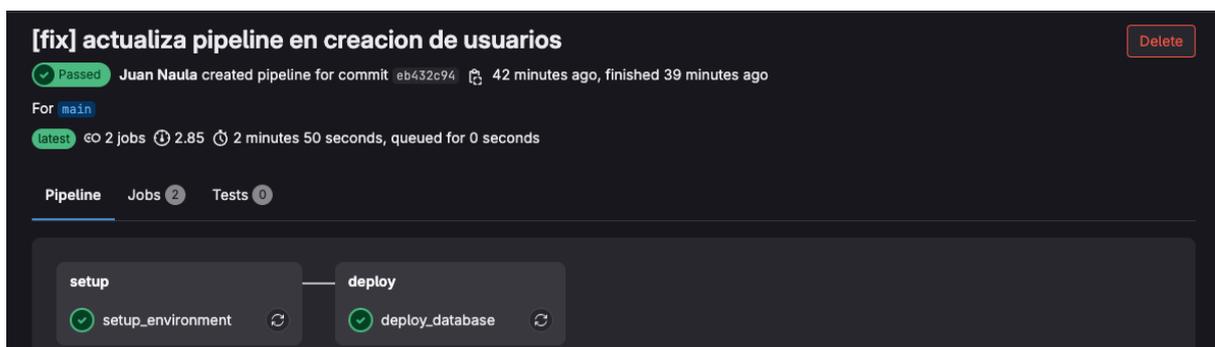


Imagen 39 Resultado ejecución pipeline de base de datos en Gitlab.

Instance ID	Issues	Cloud SQL edition	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Actions
historial-instance		Enterprise	PostgreSQL 15	34.55.142.50	10.8.0.15	historial-financiero-gcr...	ENABLE	us-central1-c	

*Imagen 40 Resultado despliegue base de datos en ambiente Development.*

Instance ID	Issues	Cloud SQL edition	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Actions
historial-instance		Enterprise	PostgreSQL 15	34.46.107.202	10.66.118.9	historial-financiero-gcr...	ENABLE	us-central1-a	

*Imagen 41 Resultado despliegue base de datos en ambiente Production*

En la imagen 42 se muestran los resultados de la ejecución del pipeline de base de datos, donde se observan tanto los casos de éxito como los de error. Es importante señalar que las fallas ocurrieron debido a errores de autenticación en el proceso de despliegue, los cuales fueron corregidos oportunamente, permitiendo la ejecución exitosa del despliegue.

Status	Pipeline	Created by	Stages
Passed 00:06:49 1 hour ago	[fix] actualiza pipeline en creacion de usu... #1517466683 main eb432c94 latest		✓✓
Failed 00:06:25 2 days ago	[feat] actualiza pipeline con validacion de ... #1517457420 main f0160889		✓✗
Passed 00:06:52 1 hour ago	[feat] actualiza pipeline con validacion de ... #1516674513 development f0160889 latest		✓✓
Passed 00:06:53 3 days ago	[feat] actualiza pipeline con validacion de ... #1516607985 development b5fad783		✓✓
Passed 00:06:36 3 days ago	[feat] agrega creacion de usuarios de ad... #1515462788 development 0983a421		✓✓
Failed 00:06:46 3 days ago	[feat] agrega creacion de usuarios de ad... #1515454757 development 42be2ea3		✓✗
Passed 00:07:02	[feat] agrega fuente proyecto de base de ... #1515443123 development 1a2880a8		✓✓

Imagen 42 Resultado ejecución pipelines data base.

## 4.13 ANÁLISIS CI /CD

A continuación, en las imágenes 43(base de datos), 44 (backend) y 45 (frontend) se presenta el análisis de CI/CD proporcionado por GitLab. En el eje horizontal (X) se observan los meses en los que se realizaron los versionamientos, mientras que en el eje vertical (Y) se muestra, en color violeta, el total de versionamientos generados dentro de los rangos de meses y en color verde los despliegues exitosos realizados. Esta visualización permite evaluar la frecuencia y el éxito de los despliegues en distintos períodos, facilitando el análisis de tendencias en el ciclo de desarrollo.

Este análisis de CI/CD tiene como propósito ofrecer una visión general del rendimiento de los ductos a lo largo del tiempo. La métrica de tiempo se basa en períodos prolongados, como semanas o meses, lo que facilita identificar tendencias en el rendimiento de los ductos, permitiendo una evaluación sobre la estabilidad.

En términos de datos visualizados, se incluyen tanto los ductos exitosos como los fallidos durante cada período, brindando una visión integral sobre la estabilidad de

las implementaciones. También permite comparar los pipelines exitosos frente a aquellos que presentaron fallas o que requirieron más tiempo de

El uso principal de este tipo de análisis es proporcionar a los equipos una perspectiva histórica y ayudar a evaluar la efectividad de los cambios implementados en la infraestructura CI/CD, como las optimizaciones y automatizaciones. Esta información es esencial para tomar decisiones informadas sobre mejoras continuas, ajustes en el proceso y optimización del ciclo de desarrollo.



Imagen 43 Análisis CI/CD, grafico de embudos historial-financiero-db.

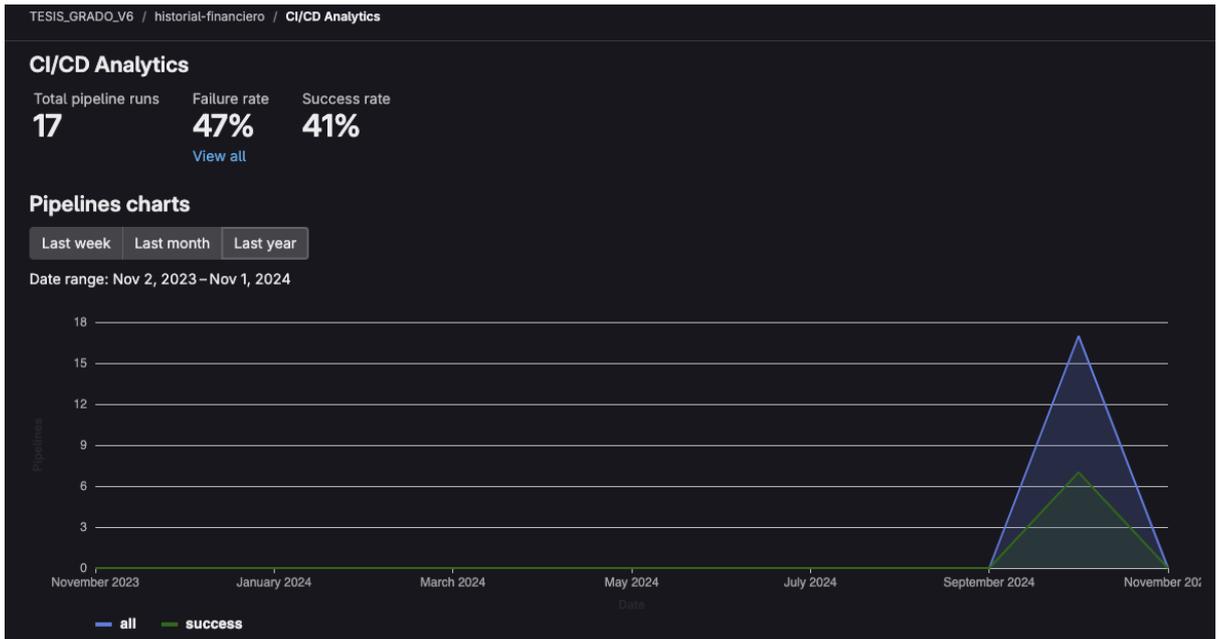


Imagen 44 Análisis CI/CD, grafico de embudos historial-financiero.



Imagen 45 Análisis CI/CD, grafico de embudos historial-financiero-app.

## 4.14 ANÁLISIS PIPELINE



Imagen 44 Resultado duración pipeline en los últimos 30 commits para historial-financiero-db.

En las imágenes 44 (base de datos), 51 (backend) y 55 (frontend) se presenta el análisis de la duración del pipeline para las últimas 30 confirmaciones (commits) en GitLab. Este análisis tiene como propósito evaluar el rendimiento reciente de los pipelines, centrándose en los tiempos de ejecución específicos de los últimos 30 commits.

En cuanto a la métrica de tiempo, se muestran los tiempos que tomaron ejecutar cada uno de los últimos 30 pipelines, lo cual facilita identificar variaciones en la duración de las ejecuciones recientes y posibles problemas de rendimiento. Los datos visualizados representan el tiempo de duración de cada pipeline, lo que permite detectar si algún compromiso en particular ha generado un aumento en el tiempo de ejecución del pipeline. Este incremento podría señalar la necesidad de realizar ajustes o correcciones en cambios recientes que afectarán el desempeño.

Este análisis es especialmente útil para realizar diagnósticos rápidos de rendimiento y resolver problemas de manera inmediata, ayudando a los equipos a optimizar la velocidad de las tuberías en tiempo real.

Como resultado de este análisis, es posible identificar de forma clara las diferencias clave entre el análisis de CI/CD general y el análisis específico de los últimos 30

commits, facilitando así la comprensión del rendimiento reciente en comparación con las tendencias históricas, como se observa en la tabla.

Característica	Análisis de CI/CD	Duración del pipeline para los últimos 30 confirmaciones
Rango de Tiempo	Largo plazo (semanas o meses)	Corto plazo (últimos 30 commits)
Propósito	Análisis de tendencias y estabilidad general	Monitoreo de duración y rendimiento reciente
Visualización	Cantidad de oleoductos exitosos/fallidos	Duración de cada pipeline
Uso ideal	Evaluar mejoras generales en CI/CD	Diagnosticar problemas específicos en el corto plazo

Tabla 5 Tabla diferencias clave entre CI/CD Analytics y 30 últimos commits.

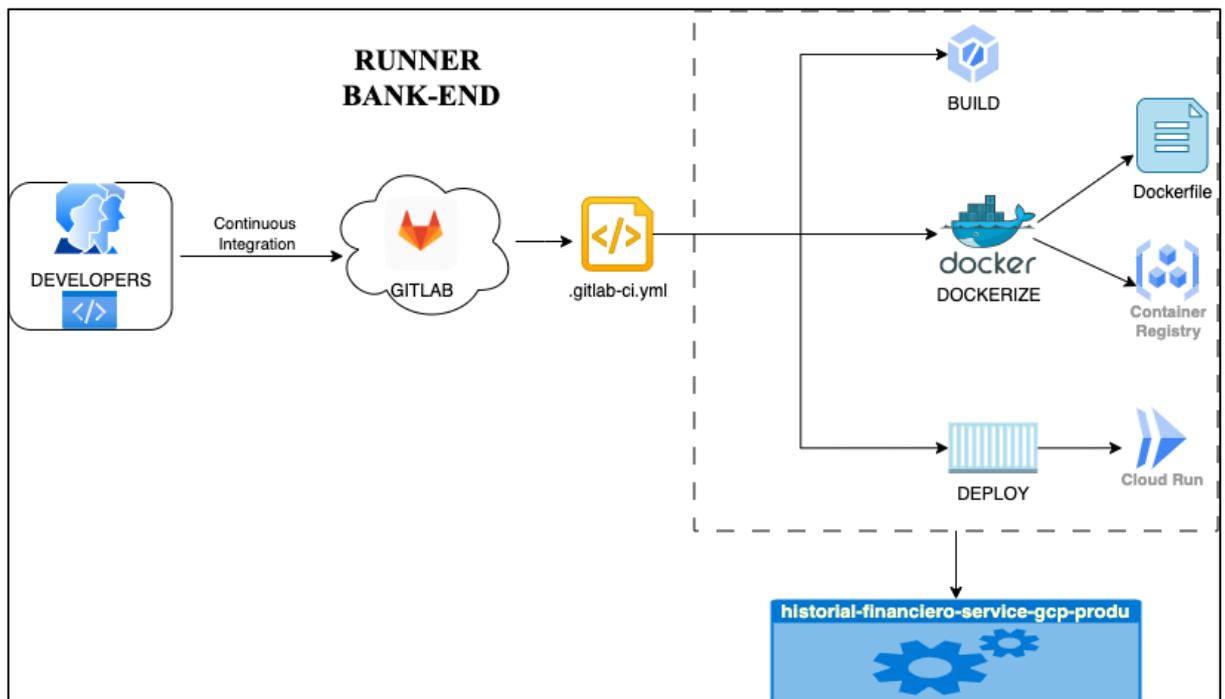


Imagen 45 Diagrama despliegue artefacto back-end, front-end en Google Cloud Platform servicio Cloud Run

La imagen 43 ilustra el proceso de despliegue de los servicios tanto de backend como de frontend, en el cual se sigue una secuencia de etapas (stages) para alcanzar este objetivo. El proceso inicia cuando los colaboradores realizan un "push" de sus cambios en cualquier rama del repositorio. Una vez que GitLab detecta un nuevo cambio, el pipeline se ejecuta automáticamente, desencadenando las distintas

etapas definidas para cada componente. Para el backend, las etapas incluyen build, dockerize, create-vpc y deploy. En el caso del frontend, las etapas ejecutadas son build, dockerize y deploy. Este flujo permite un despliegue automatizado y continuo de las aplicaciones, asegurando que los cambios se implementen de manera controlada y efectiva.

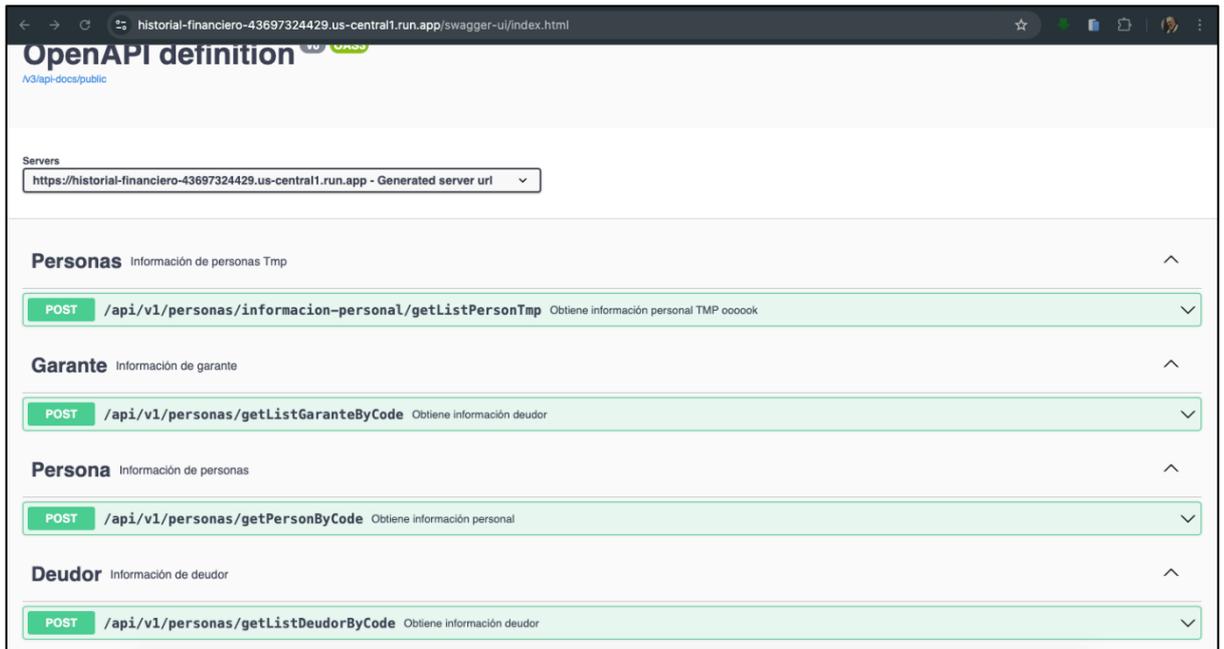


Imagen 46 Resultado Despliegue back-end cloud run

\* Se colocó estos nombres debido a que así se maneja actualmente en la cooperativa

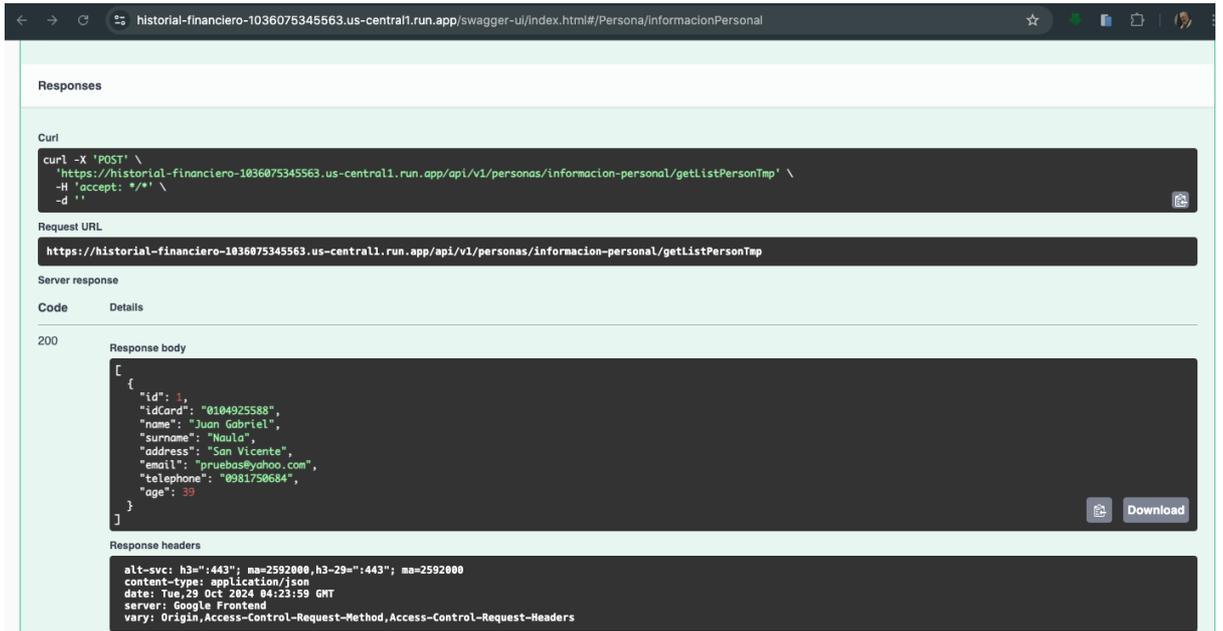


Imagen 47 Resultado consumo servicios back-end

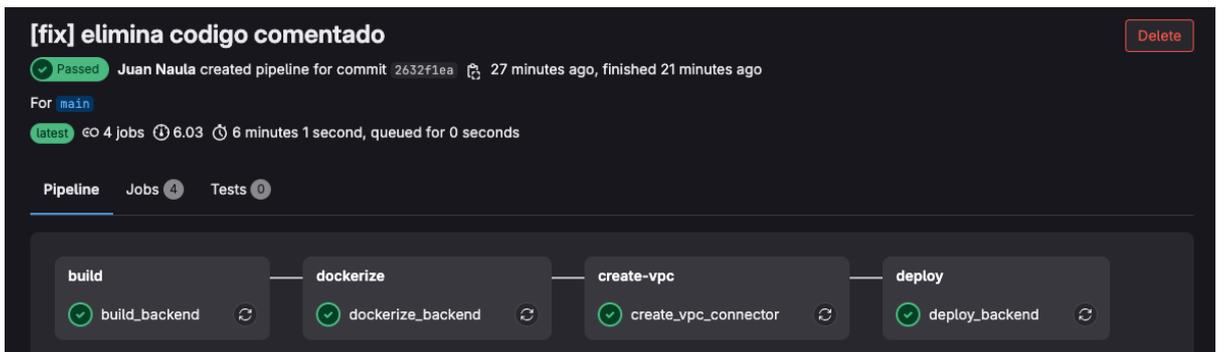


Imagen 48 Resultado ejecución secuencia de pasos pipeline back-end en gitlab.

Status	Pipeline	Created by	Stages
Passed 00:05:54 1 hour ago	update #1517533876 development 7ec3114b latest	[Avatar]	4 stages (all passed)
Passed 00:08:14 2 days ago	update #1517532912 development e3e69cb8	[Avatar]	4 stages (all passed)
Passed 00:08:05 1 hour ago	[fix] agrega políticas de cors para ambient... #1517519000 main 5d1a3121 latest	[Avatar]	4 stages (all passed)
Passed 00:05:54 2 days ago	[fix] agrega políticas de cors para ambient... #1517510893 main b5b9487d	[Avatar]	4 stages (all passed)
Passed 00:08:01 2 days ago	[fix] elimina código comentado #1517481163 main 2632f1ea	[Avatar]	4 stages (all passed)
Passed 00:05:43 2 days ago	[fix] elimina código comentado #1517480860 development 2632f1ea	[Avatar]	4 stages (all passed)
Passed 00:05:54	[chore] update #1516992442 development d045dca0	[Avatar]	4 stages (all passed)

Imagen 49 Resultado ejecución pipelines backend.

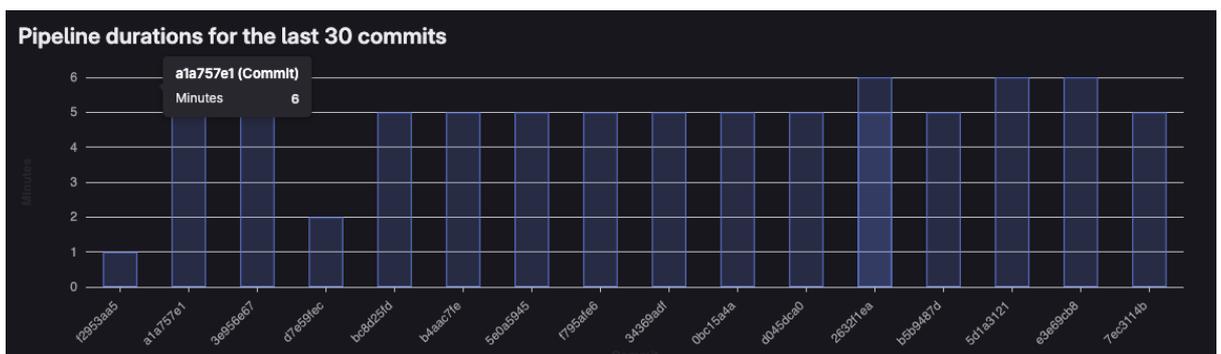


Imagen 51 Resultado duración pipeline en los últimos 30 commits para historial-financiero.

Las Imágenes 43, 44, 50, 51, 54 y 55 ilustran la frecuencia de implementaciones realizadas, lo que permite evaluar tanto la velocidad como la estabilidad en la entrega de software. Esta métrica es fundamental en el ámbito de DevOps, especialmente dentro de las DORA Metrics (métricas de DevOps Research and Assessment), que son indicadores clave para medir el rendimiento de los equipos de desarrollo y operaciones en términos de agilidad y confiabilidad, proporcionando una evaluación crítica del flujo de trabajo.

La frecuencia de implementación es una de las cuatro métricas principales de DORA, junto con el Tiempo de Espera para Cambios (Lead Time for Changes), Tasa de Fallo

---

de Cambios (Change Failure Rate) y Tiempo de Recuperación (Time to Restore Service). Estas métricas brindan una visión integral del desempeño en los procesos DevOps, facilitando la identificación de oportunidades de mejora en la agilidad y efectividad de los equipos.

Los datos presentados en las imágenes permiten obtener una visión crítica de las prácticas actuales, lo que fomenta mejoras continuas en los procesos de despliegue y soporte. La frecuencia de implementaciones refleja la capacidad del equipo para entregar cambios de manera regular y confiable, mejorando la capacidad para responder ante la demanda de nuevas funcionalidades o correcciones en producción. Esto representa un avance significativo en comparación con la realidad actual de la cooperativa, donde el proceso desde el desarrollo hasta las pruebas unitarias, compilaciones y despliegues, se lleva a cabo de forma manual. Este análisis se detalla en las tablas 6.

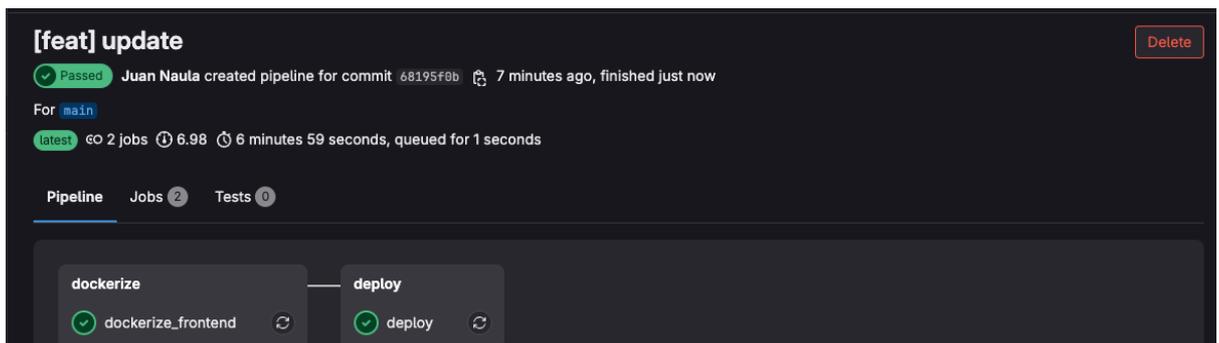


Imagen 52 Resultado ejecución secuencia de pasos pipeline front-end en gitlab.

Status	Pipeline	Created by	Stages
Passed 00:06:50 57 minutes ago	Merge branch 'development' into 'main' #1522296087 latest		2 stages
Passed 00:09:33 1 hour ago	[chore] update #1522268547 latest		3 stages
Passed 00:06:59 2 days ago	[feat] update #1517495165		2 stages
Passed 00:09:20 2 days ago	[feat] update #1517429713		3 stages
Passed 00:10:52 2 days ago	[chore] update #1517381504		3 stages
Passed	[chore] update #1517260081		3 stages

Imagen 53 Resultado ejecución secuencia de pasos pipeline front-end en gitlab.

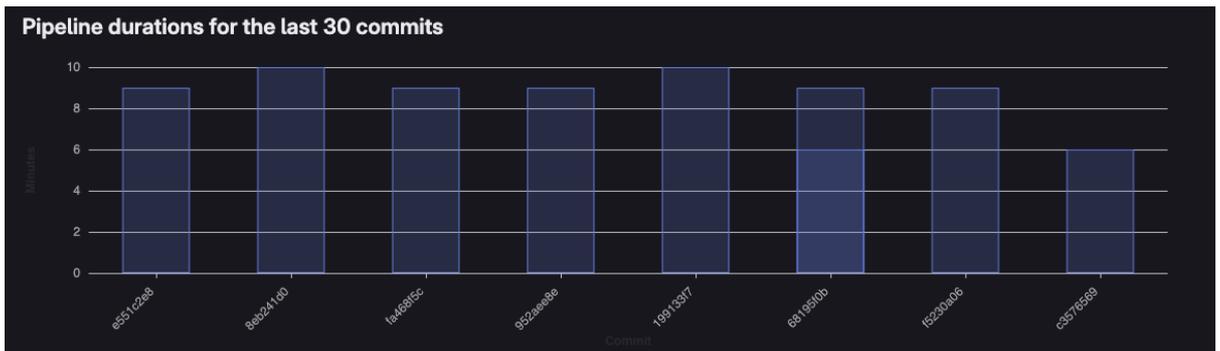


Imagen 55 Resultado duración pipeline en los últimos 30 commits para historial-financiero-app.

Name	Deployment type	Req/sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
historial-financiero	Container	0	us-central1	Allow unauthenticated	All	2 RECOMMENDATIONS	11 minutes ago	historial-financiero-gcr-dev.iam.gserviceaccount.cc
historial-financiero-app	Container	0	us-central1	Allow unauthenticated	All	SECURITY	1 hour ago	historial-financiero-gcr-dev.iam.gserviceaccount.cc

Imagen 56 Resultado despliegue Front-end y back-end development en Cloud run.

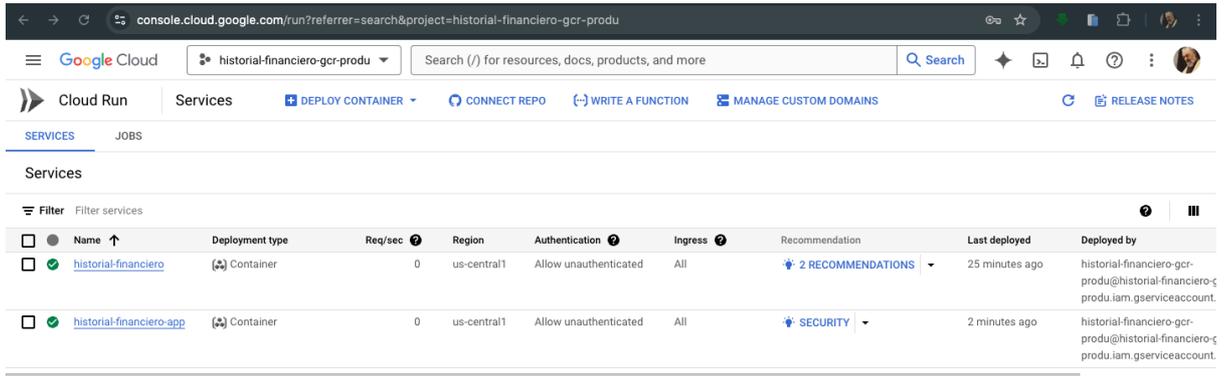


Imagen 57 Resultado despliegue Front-end y back-end production en Cloud run.

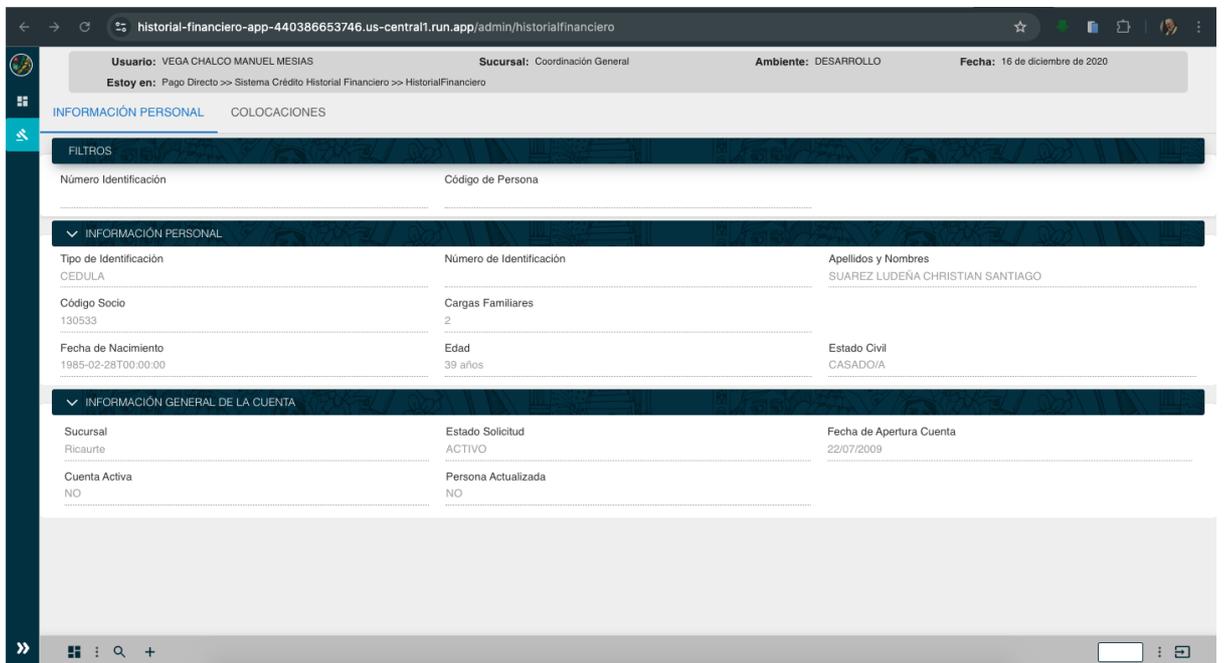


Imagen 58 Resultado aplicación en ejecución ambiente development.

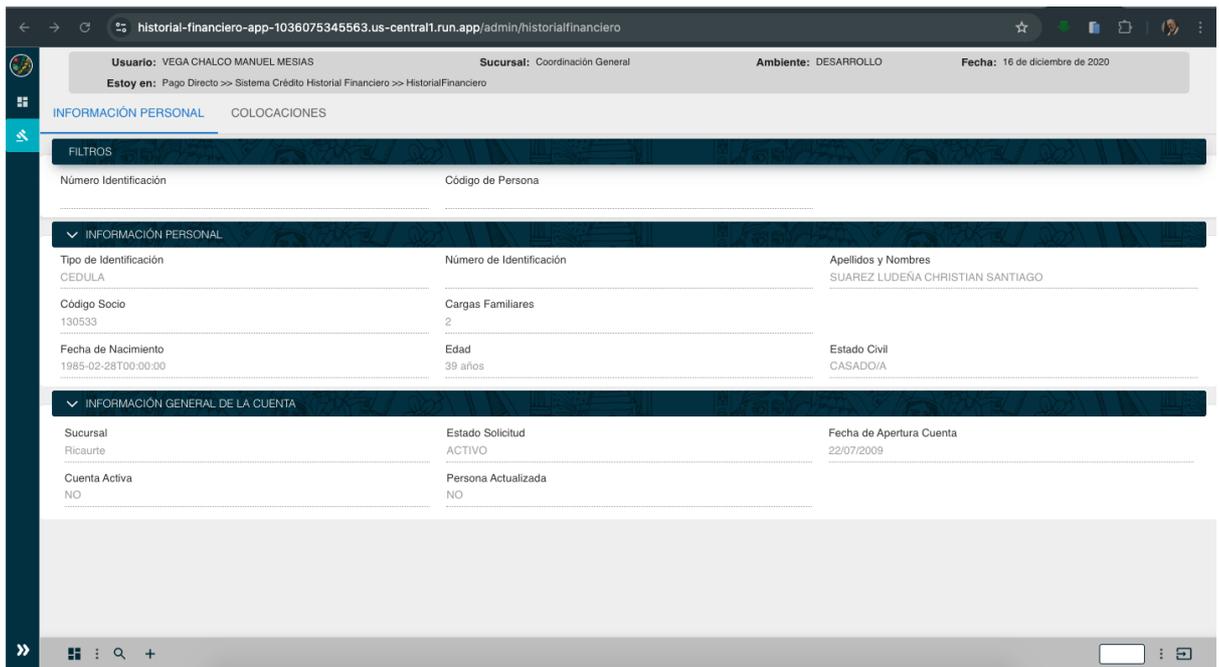


Imagen 59 Resultado aplicación en ejecución ambiente production.

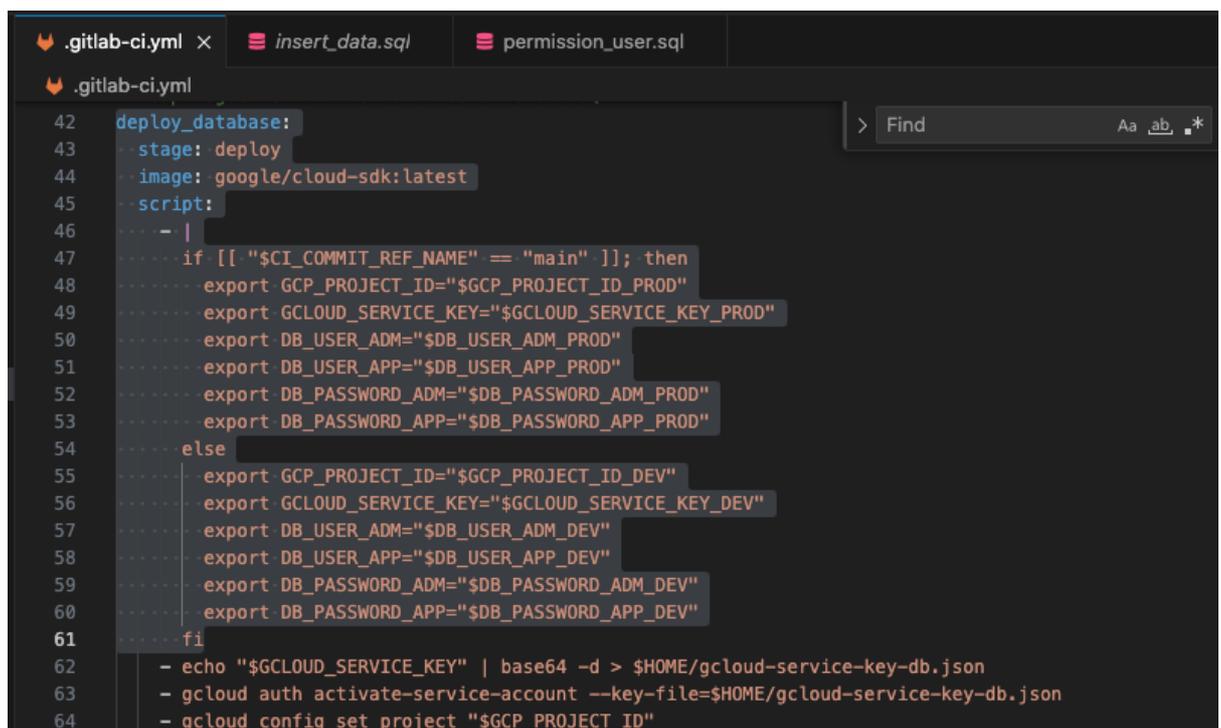
La estrategia de despliegue sigue un enfoque automatizado mediante pipelines de CI/CD (Integración Continua/Despliegue Continuo), lo cual asegura la coherencia y eficiencia en cada etapa del proceso. GitLab CI, junto con Docker, facilita el empaquetamiento de los contenedores y su posterior despliegue en servicios de Google Cloud, tales como Cloud Run para el backend y frontend, y Cloud SQL para la base de datos.

Este enfoque garantiza que los distintos componentes del sistema se desplieguen de manera coherente y escalable en la nube, lo que facilita la gestión y actualización del sistema de historial financiero. La automatización minimiza errores humanos, acelera la entrega y asegura un entorno de producción robusto y confiable, preparado para respaldar el aumento de la demanda de manera segura.

En un entorno DevOps, el uso de pipelines que permiten la selección dinámica del ambiente de despliegue según la variable `CI_COMMIT_REF_NAME` aporta múltiples beneficios en el desarrollo de aplicaciones. Este enfoque permite configurar el entorno adecuado de manera automatizada según la rama de GitLab desde la cual se ejecute el pipeline.

Para entornos diferenciados (producción y desarrollo), la lógica condicional para definir variables de entorno, tales como credenciales y configuraciones específicas de la base de datos, asegura que los despliegues sean consistentes y seguros. En las **imágenes 61 al 63** se observa la codificación de selección dinámica del ambiente de despliegue; en el paso `deploy_database`, el pipeline evalúa el valor de `CI_COMMIT_REF_NAME`: si corresponde a la rama `main`, se seleccionan automáticamente los parámetros de producción (`GCP_PROJECT_ID_PROD`, `GCLOUD_SERVICE_KEY_PROD`, etc.). Para otras ramas, se eligen los valores de desarrollo (`GCP_PROJECT_ID_DEV`, `GCLOUD_SERVICE_KEY_DEV`, etc.), aplicándose esta lógica al despliegue dinámico de base de datos, backend y frontend.

Esta estrategia suprime la necesidad de intervención manual, disminuye la probabilidad de errores en la configuración y optimiza el flujo de trabajo, permitiendo que cada despliegue se ajuste automáticamente al ambiente adecuado. En este contexto, la práctica no solo mejora la eficiencia operativa, sino que también establece una arquitectura robusta y adaptable, adecuada para entornos de desarrollo ágiles y seguros.



```
.gitlab-ci.yml x  insert_data.sql  permission_user.sql
.gitlab-ci.yml
42  deploy_database:
43    stage: deploy
44    image: google/cloud-sdk:latest
45    script:
46    - |
47      if [[ "$CI_COMMIT_REF_NAME" == "main" ]]; then
48        export GCP_PROJECT_ID="$GCP_PROJECT_ID_PROD"
49        export GCLOUD_SERVICE_KEY="$GCLOUD_SERVICE_KEY_PROD"
50        export DB_USER_ADM="$DB_USER_ADM_PROD"
51        export DB_USER_APP="$DB_USER_APP_PROD"
52        export DB_PASSWORD_ADM="$DB_PASSWORD_ADM_PROD"
53        export DB_PASSWORD_APP="$DB_PASSWORD_APP_PROD"
54      else
55        export GCP_PROJECT_ID="$GCP_PROJECT_ID_DEV"
56        export GCLOUD_SERVICE_KEY="$GCLOUD_SERVICE_KEY_DEV"
57        export DB_USER_ADM="$DB_USER_ADM_DEV"
58        export DB_USER_APP="$DB_USER_APP_DEV"
59        export DB_PASSWORD_ADM="$DB_PASSWORD_ADM_DEV"
60        export DB_PASSWORD_APP="$DB_PASSWORD_APP_DEV"
61      fi
62    - echo "$GCLOUD_SERVICE_KEY" | base64 -d > $HOME/gcloud-service-key-db.json
63    - gcloud auth activate-service-account --key-file=$HOME/gcloud-service-key-db.json
64    - gcloud config set project "$GCP_PROJECT_ID"
```

Imagen 60 Pipeline deploy base de datos, historial-financiero

```
application-prod.properties application-dev.properties application.properties Dockerfile .gitlab-ci.yml X
.gitlab-ci.yml
82 # Despliegue en Google Cloud Run
83 deploy_backend:
84   stage: deploy
85   image: google/cloud-sdk:latest
86   script:
87     - |
88       if [[ "$CI_COMMIT_REF_NAME" == "main" ]]; then
89         export GCP_PROJECT_ID="$GCP_PROJECT_ID_PROD"
90         export GLOUD_SERVICE_KEY="$GLOUD_SERVICE_KEY_PROD"
91         export CLOUD_SQL_CONNECTION_NAME="$GCP_PROJECT_ID:$GCP_REGION:$INSTANCE_NAME"
92         export DATASOURCE="$DATASOURCE_PROD"
93         export DB_USER="$DB_USER_ADM_PROD"
94         export DB_PASSWORD="$DB_PASSWORD_ADM_PROD"
95       else
96         export GCP_PROJECT_ID="$GCP_PROJECT_ID_DEV"
97         export GLOUD_SERVICE_KEY="$GLOUD_SERVICE_KEY_DEV"
98         export CLOUD_SQL_CONNECTION_NAME="$GCP_PROJECT_ID:$GCP_REGION:$INSTANCE_NAME"
99         export DATASOURCE="$DATASOURCE_DEV"
100        export DB_USER="$DB_USER_ADM_DEV"
101        export DB_PASSWORD="$DB_PASSWORD_ADM_DEV"
102      fi
103      - echo "$GLOUD_SERVICE_KEY" | base64 -d > $HOME/gcloud-service-key.json
104      - gcloud auth activate-service-account --key-file=$HOME/gcloud-service-key.json
105      - gcloud config set project "$GCP_PROJECT_ID"
```

Imagen 61 Pipeline deploy back-end, historial-financiero.

```
Dockerfile JS endpoints.js .gitlab-ci.yml X
.gitlab-ci.yml
75
76 deploy:
77   stage: deploy
78   image: google/cloud-sdk:latest
79   script:
80     - |
81       if [[ "$CI_COMMIT_REF_NAME" == "main" ]]; then
82         export GCP_PROJECT_ID="$GCP_PROJECT_ID_PROD"
83         export GLOUD_SERVICE_KEY="$GLOUD_SERVICE_KEY_PROD"
84         export GCP_SERVICE_NAME="historial-financiero-app"
85         export INSTANCE_NAME="$INSTANCE_NAME_PROD"
86       else
87         export GCP_PROJECT_ID="$GCP_PROJECT_ID_DEV"
88         export GLOUD_SERVICE_KEY="$GLOUD_SERVICE_KEY_DEV"
89         export GCP_SERVICE_NAME="historial-financiero-app"
90         export INSTANCE_NAME="$INSTANCE_NAME_DEV"
91       fi
92       # Autenticación en Google Cloud
93       - echo "$GLOUD_SERVICE_KEY" | base64 -d > $HOME/gcloud-service-key-front.json
94       - gcloud auth activate-service-account --key-file=$HOME/gcloud-service-key-front.json
95       - gcloud config set project "$GCP_PROJECT_ID"
96       - gcloud auth configure-docker
97       # Leer la imagen desde el archivo generado en el stage anterior
98       - export GCP_IMAGE=$(cat src/image.txt)
```

Imagen 62 Pipeline deploy front-end, historial-financiero-app

---

Finalizado el despliegue de los servicios, resulta fundamental resaltar los beneficios que proporcionan los servicios en la nube, especialmente las revisiones de implementación en Google Cloud Run, ilustradas en las imágenes 63 y 64. Las revisiones representan versiones inmutables de un servicio que se crean con cada despliegue o actualización de configuraciones, ya sea de código, variables de entorno, configuración de red o límites de recursos. Este sistema proporciona un control detallado de los cambios efectuados en el servicio, lo que facilita la administración y la estabilidad operativa.

## 4.15 FUNCIONALIDAD Y USO DE LAS REVISIONES

**Creación de una Revisión:** Con cada despliegue o cambio en el servicio, ya sea una nueva imagen de contenedor o una modificación en la configuración, Cloud Run genera una nueva revisión que captura el estado completo del servicio en ese momento. Esta revisión es inmutable, lo que significa que no se puede modificar una vez creada.

**Selección de Revisión para Enrutamiento:** Es posible controlar el tráfico dirigido a diferentes revisiones, lo cual facilita una transición gradual y pruebas de nuevas versiones.

**División de tráfico:** Se puede asignar un porcentaje del tráfico a distintas revisiones, una práctica útil para pruebas de nuevas versiones en producción (canary releases o blue-green deployments) y para garantizar que los cambios no afecten negativamente a los usuarios finales.

**Rollbacks:** En caso de que una nueva revisión presente problemas, es posible redirigir el tráfico rápidamente a una revisión anterior, minimizando el impacto en los usuarios.

---

**Seguimiento de Cambios y Auditoría:** Al mantener el estado exacto de cada despliegue, las revisiones permiten una auditoría sencilla, facilitando el historial de despliegues y el seguimiento de versiones de configuración o código desplegadas.

**Control de Entornos de Desarrollo y Producción:** En entornos de desarrollo, las revisiones permiten probar y validar cambios antes de implementarlos en producción, sin afectar las versiones estables en producción.

**Ventajas de las Revisiones:**

**Facilidad de Rollback:** Las revisiones permiten retroceder a una versión anterior del servicio de forma rápida en caso de problemas en producción.

**Despliegues Graduales (Canary/Blue-Green):** Facilitan pruebas controladas en producción al asignar un porcentaje de tráfico a nuevas revisiones para validarlas.

**Auditoría y Control de Cambios:** Cada revisión guarda una copia inmutable de la configuración, lo cual resulta esencial para auditorías y rastreo de cambios.

**Resiliencia y Seguridad:** Al ser inmutables, las revisiones garantizan que las versiones de despliegue no sean alteradas accidentalmente, asegurando la estabilidad y confiabilidad del servicio en producción.

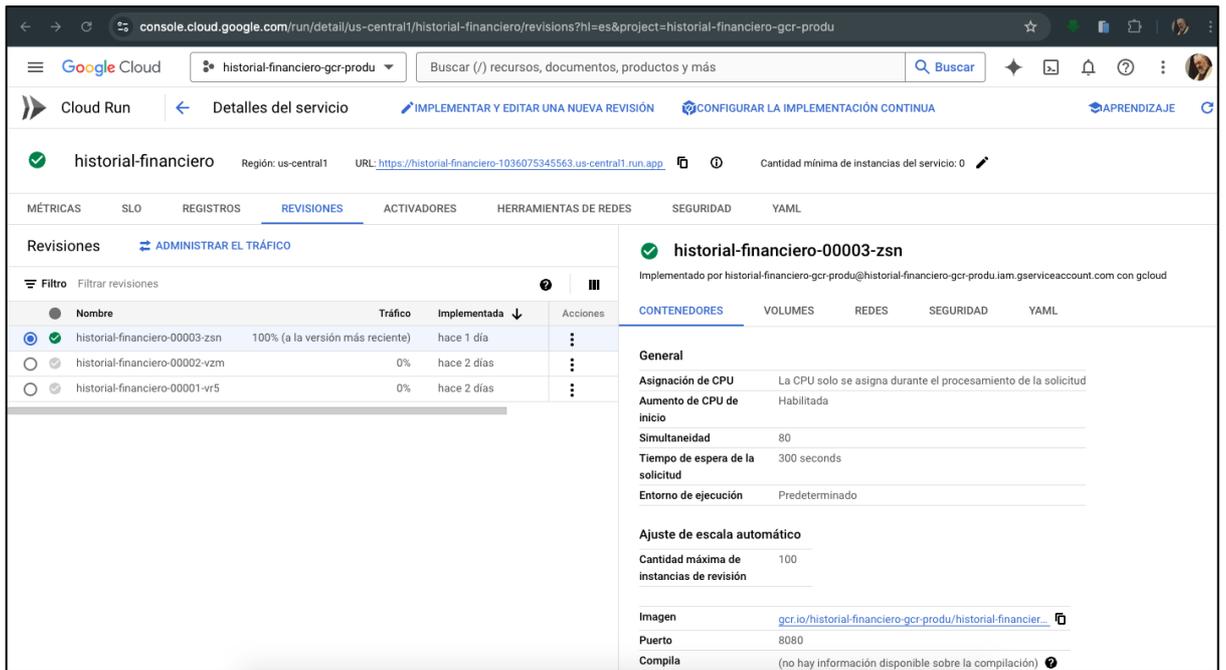


Imagen 63 Revisiones servicio backend historial-financiero

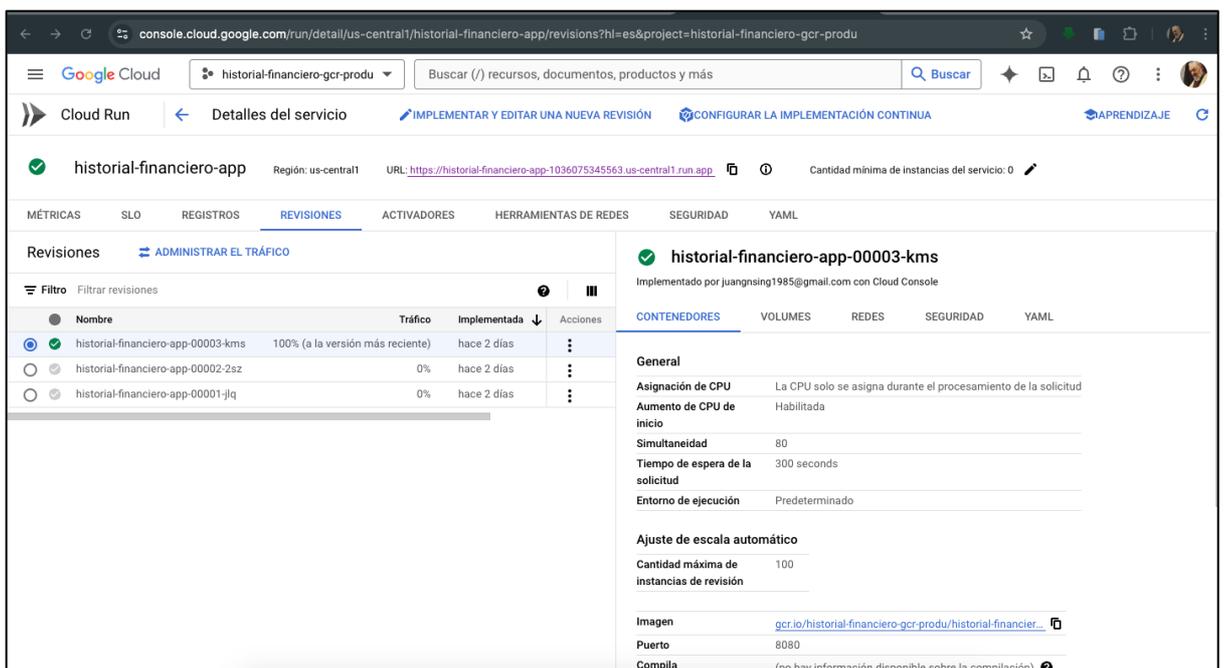


Imagen 64 Revisiones frontend historial-financiero-app

Por otra parte, la implementación de Cloud SQL Studio en el despliegue del servicio historial-financiero-db ofrece una administración centralizada y eficiente de la base de datos en la nube. Esta herramienta permite gestionar instancias, ejecutar consultas SQL y realizar tareas de mantenimiento de forma sencilla, sin requerir

---

configuraciones complejas ni acceso directo a la infraestructura subyacente, como se ilustra en la imagen 65 donde se puede destacar.

### **Funcionalidades Principales en el Contexto del Despliegue.**

- **Creación y Configuración de Instancias:** Cloud SQL Studio permite configurar instancias específicamente para historial-financiero-db, con opciones personalizables de escalabilidad, respaldo y seguridad, adaptándose a los requisitos del servicio.
- **Gestión de Usuarios y Permisos:** A través de esta herramienta se facilita la administración de usuarios y la definición de permisos específicos, garantizando que el acceso a la base de datos cumpla con las políticas de seguridad establecidas.
- **Ejecución de Consultas SQL:** Cloud SQL Studio permite ejecutar y validar consultas SQL directamente en su interfaz, lo cual resulta esencial para monitorear las operaciones y el estado de la base de datos sin interrumpir el servicio historial-financiero-db.
- **Supervisión y Monitoreo de Rendimiento:** Las herramientas de monitoreo en tiempo real permiten observar el rendimiento de historial-financiero-db, identificar posibles cuellos de botella y realizar ajustes para optimizar el servicio.
- **Automatización de Respaldos y Restauración:** La plataforma facilita la configuración de respaldos automáticos, la restauración de información, permitiendo garantizar la continuidad operativa y la integridad de los datos ante posibles fallos.

### **Beneficios en el Despliegue de historial-financiero-db**

La integración de Cloud SQL Studio optimiza tanto el despliegue como la gestión de historial-financiero-db, proporcionando una interfaz intuitiva y herramientas avanzadas para la administración de bases de datos en Google Cloud. Esto no solo incrementa la eficiencia operativa, sino que también refuerza la seguridad y la

estabilidad del servicio, consolidando a Cloud SQL Studio como una solución integral para la gestión y el monitoreo de bases de datos en entornos de nube.

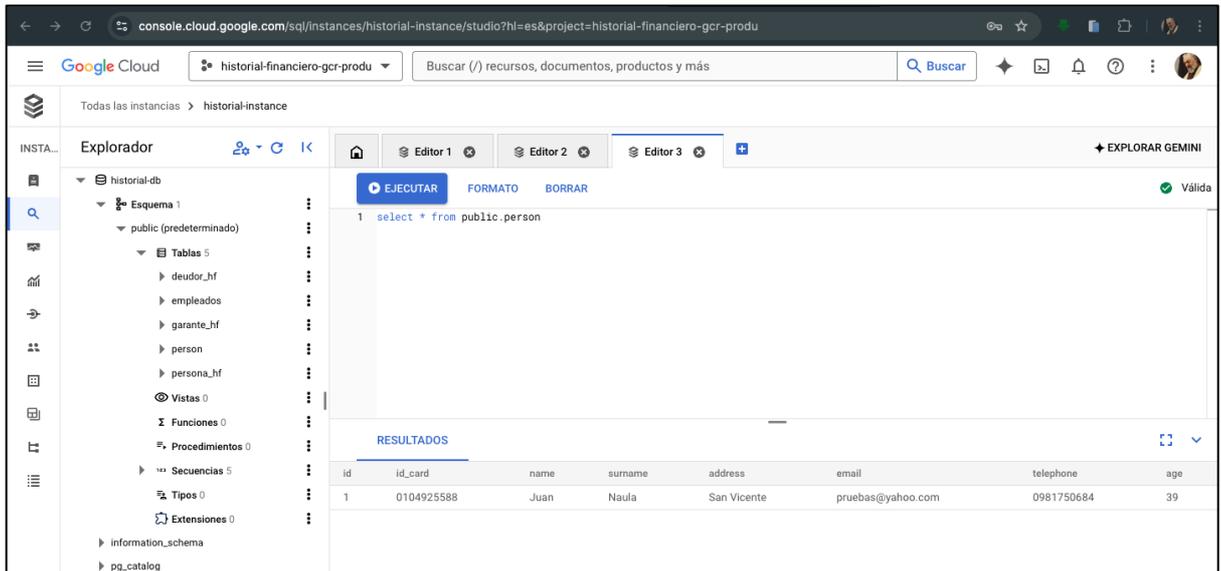


Imagen 65 Herramienta de gestión centralizada Cloud SQL Studio

## 4.16 FEEDBACK CONTINUO.

Durante el proceso de desarrollo, se implementaron actividades clave, basadas en marcos ágiles como Scrum, con el propósito de obtener retroalimentación en cada fase del ciclo de desarrollo. Las ceremonias de Scrum, como el “Daily Scrum”, el “Sprint Review” y el “Sprint Retrospective”, se estructuraron de la siguiente manera:

**Daily Scrum (Reunión Diaria):** Se realizaron reuniones diarias en las que cada miembro del equipo reportó el progreso, identificando bloqueos y oportunidades de mejora. Estas reuniones permitieron un flujo constante de información, asegurando la sincronización del equipo y la detección temprana de problemas.

**Sprint Review (Revisión del Sprint):** Al concluir cada Sprint, se presentaron los incrementos del MVP a los interesados clave, quienes proporcionaron retroalimentación valiosa. Esta actividad permitió ajustar las funcionalidades

---

entregadas de acuerdo con las expectativas y requerimientos del usuario final, garantizando la entrega de valor constante.

Sprint Retrospective (Retrospectiva del Sprint): Después de cada Sprint, se realizó una retrospectiva para analizar las ventajas y aspectos de mejora del equipo. Este proceso facilitó la implementación de ajustes en el enfoque de trabajo y promovió un ambiente de mejora continua. Nos permitió poder identificar los procesos que nos demandaban más tiempo y los cuellos de botella que se estaban formando, las actividades que no estábamos realizando de manera correcta lo cual nos permitió tener un mejor tiempo de respuesta y desarrollo para los siguientes sprint.

## SINOPSIS CAPÍTULO 5

En el capítulo anterior, se detalló la implementación de principios DevOps y metodologías ágiles en el módulo de historial financiero. Este proceso permitió optimizar el ciclo de vida del software, alineándolo con la propuesta de valor del proyecto. Se describieron las prácticas y herramientas clave utilizadas, así como los beneficios obtenidos en términos de eficiencia, calidad y colaboración.

En este capítulo, el objetivo principal es analizar el impacto de la implementación de DevOps y metodologías ágiles en el módulo de historial financiero. Se evaluarán los resultados obtenidos en relación con los objetivos planteados, identificando los puntos fuertes y las áreas de mejora. Además, se explorarán las lecciones aprendidas y las mejores prácticas que pueden ser aplicadas a otros módulos del sistema.

---

# 5. RESULTADOS Y DISCUSIÓN

## 5.1 MEDICIÓN DE RESULTADOS

Para abordar de manera efectiva el tema de Medición de Resultados en el contexto de las métricas DORA, se presenta una tabla que refleja el desempeño de los procesos de desarrollo e implementación de software en la Cooperativa. Las métricas DORA, que incluyen Frecuencia de Implementación, Plazo de Cambio, Tasa de Fallas y Tiempo de Recuperación, fueron fundamentales para evaluar la eficiencia y calidad del proceso actual en comparación con los beneficios potenciales de la implementación de DevOps.

Estas métricas se derivan del análisis del estado actual del proceso de desarrollo dentro de la Cooperativa. Cabe destacar que varios aspectos aún dependen de tareas manuales y carecen de priorización. Este análisis establece una línea de base que luego se puede comparar con los resultados posteriores a la adopción de prácticas DevOps. Con DevOps, se anticipan mejoras en los indicadores actuales, incluido un ciclo de desarrollo optimizado, una mayor frecuencia de implementación y reducciones tanto en las tasas de fallas como en los tiempos de recuperación.

Asimismo, un aspecto fundamental para el cálculo de la **Tasa de Fallo de Cambios** es la correcta aplicación de la fórmula que determina este indicador de rendimiento. En este caso, el cálculo se realiza tomando el número de implementaciones fallidas que requieren intervención correctiva y dividiéndolo entre el número total de implementaciones en un período específico. Este cociente, multiplicado por 100, expresa el porcentaje de cambios que presentan fallos.

Este valor permite evaluar la estabilidad y confiabilidad del proceso de despliegue, brindando información clave sobre la calidad de las prácticas de desarrollo y pruebas. Un resultado elevado sugiere áreas de mejora en los procesos previos al despliegue, tales como revisiones de código y pruebas automatizadas.

---

$$\text{Change Failure Rate} = \left( \frac{\text{Número de fallos}}{\text{Número total de implementaciones}} \right) \times 100$$

En el Tiempo de Restauración del Servicio es esencial aplicar correctamente la fórmula recomendada, que permite obtener un valor promedio de restauración. Esta métrica evalúa la rapidez con la que un equipo puede restablecer un servicio tras una interrupción. La fórmula se expresa de la siguiente manera:

$$\text{Tiempo Promedio de Restauración} = \frac{\sum(\text{Tiempo de Restauración de cada incidente})}{\text{Número de incidentes}}$$

Esta expresión permite calcular el promedio del tiempo que el sistema permanece inactivo o con fallos antes de ser completamente restaurado. La precisión en el registro de cada incidente y su tiempo de resolución es fundamental para obtener datos fiables, que a su vez ofrecen una visión clara de la capacidad de recuperación del equipo y del sistema en su conjunto.

Este enfoque contribuye a la identificación de áreas de mejora en los procedimientos de respuesta ante fallos, permitiendo optimizar la estabilidad y continuidad del servicio.

### 5.1.1 DORA MÉTRICA ANTES DE DEVOPS

La **Tabla 6** de resultados sirvió como punto de referencia inicial, ayudando a identificar áreas de mejora y a establecer objetivos claros dentro de una cultura DevOps. Este enfoque enfatiza la integración continua y la automatización de procesos, lo que permite un análisis más exhaustivo de las métricas de DORA y facilita la evaluación del desempeño del proceso de desarrollo e implementación a lo largo del tiempo.

<b>Métrica</b>	<b>Objetivo</b>	<b>Valores</b>	<b>Observación</b>
<b>Frecuencia de implementación (Cantidad de despliegues a producción entre una semana y una vez al mes)</b>	Medir la capacidad de respuesta ante los requerimientos o solución de incidencias presentadas.	<b>ACTUAL:</b> 1 vez al mes.  <b>REFERENCIA:</b> Entre una vez por semana y una vez al mes.	La frecuencia de despliegue actual, en el mejor de los casos, es de una vez al mes. Esto se debe a la falta de una definición clara de los requisitos, su priorización, así como a los procesos manuales vigentes dentro de la cooperativa, como las pruebas unitarias, el proceso de compilación y el despliegue manual en ambiente productivo, así como a la priorización de cambios que en muchos de los casos existen varios requerimientos desarrollados por diferentes grupos que tocan el mismo aplicativo y en este caso es necesario priorizar cuál de ellos debe desplegarse primero.
<b>Tiempo de espera para Cambios (Tiempo promedio desde el desarrollo hasta el despliegue en producción)</b>	Reducir el tiempo de entrega de cambios.	<b>ACTUAL:</b> 7 días.  <b>REFERENCIA:</b> Desempeño medio entre uno y siete días.	El tiempo mínimo que se requiere para que una solución este en producción desde su reporte, desarrollo, pruebas, compilación y despliegue en ambiente productivo es de mínimo 7 días a un mes debido a los procesos manuales que se lleva dentro de la cooperativa
<b>Tasa de Fallo de Cambios (Porcentaje de cambios desplegados en producción que requieren una corrección o un Rollback)</b>	Mantener el valor por debajo del 15%	<b>ACTUAL:</b> 100%.  <b>REFERENCIA:</b> < 15%	La alta tasa de fallos se debe a la falta de automatización en los procesos de pruebas, compilación y despliegue, los cuales se ejecutan solo al final del desarrollo. Esto ocurre porque se espera completar el desarrollo del aplicativo completo antes de su puesta en producción, sin una priorización adecuada que permita aplicar el ciclo de desarrollar, medir y aprender para implementar soluciones tempranas, así como como requisitos poco claros, priorización inadecuada o pruebas insuficientes, que se señalan en la tabla como factores que afectan la frecuencia y la confiabilidad de la implementación
<b>Tiempo de Restauración del servicio (Tiempo promedio para recuperar el servicio después de una falla en producción)</b>	Reducir el tiempo de entrega de recuperación	<b>ACTUAL:</b> Bajo rendimiento.  <b>REFERENCIA:</b> Más de un día	Actualmente, el proceso de recuperación del servicio ante un error es muy costoso, ya que puede tardar desde un día en el mejor de los casos hasta una semana, lo cual genera pérdidas para la cooperativa y provoca insatisfacción entre los socios. Esto se debe a la dependencia de procesos manuales y a la falta de una priorización efectiva para brindar soluciones oportunas.

**Tabla 6** DORA Metric implementaciones base de datos, backend, frontend previo DevOps.

## 5.1.2 DORA MÉTRICA DESPUÉS DE DEVOPS

A partir de estas métricas obtenidas en el proceso actual previo a la implementación de DevOps, se presenta la **Tabla 7**, en la que se refleja los beneficios significativos derivados de la adopción de prácticas DevOps dentro de los proyectos de la cooperativa.

Métrica	Objetivo	Valores	Observación
<b>Frecuencia de implementación (Cantidad de despliegues a producción entre una semana y una vez al mes)</b>	Medir la capacidad de respuesta ante los requerimientos o solución de incidencias presentadas.	<b>ACTUAL:</b> 4 veces al mes.  <b>REFERENCIA:</b> Entre una vez al día y una vez por semana	La aplicación de metodologías ágiles en la captura de requisitos ha permitido comprender de mejor manera las verdaderas necesidades de los usuarios y priorizarlas adecuadamente. Al organizar las tareas con enfoques ágiles, se facilita la entrega de componentes más pequeños y manejables. Asimismo, la adopción de principios DevOps, como la integración y entrega continua, respaldados por GitLab y la construcción de pipelines, ha automatizado los procesos de construcción, pruebas y despliegue. Esto ha permitido generar un flujo continuo de retroalimentación, mejorando gradualmente el desarrollo y la implementación del proyecto.
<b>Tiempo de espera para Cambios (Tiempo promedio desde el desarrollo hasta el despliegue en producción)</b>	Reducir el tiempo de entrega de cambios.	<b>ACTUAL:</b> 1 día.  <b>REFERENCIA:</b> De una hora a un día.	Con la implementación de los principios DevOps, el tiempo de espera para el despliegue se ha reducido a minutos gracias a la automatización del ciclo de vida del software mediante pipelines. Estos pipelines eliminan la necesidad de procesos manuales, permitiendo que el despliegue se realice simplemente al ejecutar las aplicaciones de merge según el ambiente en el que se requiera implementar como se puede observar en los extractos de código de las imágenes 60,61,62 detalladas en pasos anteriores.
<b>Tasa de Fallo de Cambios (Porcentaje de cambios desplegados en producción que requieren una corrección o un Rollback)</b>	Mantener el valor por debajo del 15%	<b>ACTUAL:</b> 5%.  <b>REFERENCIA:</b> < 15%	Tras adoptar DevOps, se observa una mejora significativa en esta tasa de fallos, de cuatro implementaciones, se tiene: <ul style="list-style-type: none"> <li>• A nivel de base de datos solo una presenta errores (Es importante señalar que estos errores suelen originarse por problemas de permisos durante el despliegue de servicios dockerizados en entornos de Cloud SQL)</li> <li>• A nivel de Back End no presenta errores.</li> </ul>

			<ul style="list-style-type: none"> <li>• A nivel de Front End no presenta errores.</li> </ul> <p>Podemos concluir que la implementación de DevOps no solo reduce la frecuencia de estos fallos, sino que también facilita la identificación y resolución ágil de los mismos, mejorando así la estabilidad y confiabilidad del proceso de implementación.</p>
<b>Tiempo de Restauración del servicio (Tiempo promedio para recuperar el servicio después de una falla en producción)</b>	Reducir el tiempo de entrega de recuperación	<b>ACTUAL:</b> Bajo rendimiento.  <b>REFERENCIA:</b> Entre unas horas y 1 día	<p>La adopción de prácticas como la automatización de implementaciones, el monitoreo continuo y la integración de herramientas de alerta ha reducido el tiempo necesario para restablecer servicios tras una falla o interrupción. Anteriormente, la restauración de bases de datos en Cloud SQL podía tardar entre unas horas y un día. Con DevOps, este proceso se ha optimizado, permitiendo identificar y resolver incidentes en cuestión de minutos u horas en la mayoría de los casos.</p> <p>La implementación del backend en Google Cloud Run ha aportado numerosas ventajas, especialmente en términos de automatización, administración de versiones y gestión de tráfico. GCP permite implementar revisiones de cada implementación, lo cual facilita revertir a versiones anteriores en caso de incidentes, y controlar el tráfico de manera precisa hacia cada versión, asegurando así una alta disponibilidad y un bajo riesgo al introducir cambios en producción</p> <p>La implementación del frontend en Google Cloud Run ha aportado numerosas ventajas, especialmente en términos de automatización, administración de versiones y gestión de tráfico. GCP permite implementar revisiones de cada implementación, lo cual facilita revertir a versiones anteriores en caso de incidentes, y controlar el tráfico de manera precisa hacia cada versión, asegurando así una alta disponibilidad y un bajo riesgo al introducir cambios en producción.</p> <p>Esto no solo disminuye el impacto en la disponibilidad del servicio, sino que también incrementa la confianza de los usuarios al asegurar una respuesta rápida y efectiva ante cualquier problema.</p>

**Tabla 7DORA Metric implementaciones base de datos, backend, frontend aplicando principios DevOps.**

---

De acuerdo con los resultados presentados en las Tablas 6 y 7, la implementación de prácticas DevOps ha tenido un impacto significativo en el rendimiento del equipo de desarrollo y en las operaciones de software, mejorando todos los aspectos claves evaluados por la métrica DORA.

**Frecuencia de despliegues:** Ha aumentado considerablemente, permitiendo despliegues continuos y más frecuentes gracias a la automatización, pasando de ciclos largos a múltiples despliegues diarios.

**Tiempo de ciclo:** El tiempo entre el desarrollo y la producción se ha reducido, gracias a la integración continua y las herramientas de automatización, lo que facilita una entrega más ágil y eficiente.

**Tasa de fallas de implementación:** La tasa de errores ha disminuido debido a la integración de pruebas automatizadas, monitoreo continuo y validación temprana de los cambios, lo que contribuye a despliegues más estables.

**Tiempo de recuperación:** La recuperación ante fallas se ha acelerado, reduciendo el tiempo de inactividad de horas o días a minutos u horas, gracias a la infraestructura como código, monitoreo constante y mecanismos de reversión rápida de versiones.

En resumen, la adopción de DevOps ha optimizado el ciclo de vida del software, mejorando su calidad, reduciendo los tiempos de inactividad y aumentando la confianza de los usuarios al garantizar una respuesta rápida y confiable ante cualquier incidente.

Gracias a la adopción de metodologías ágiles, fue posible identificar oportunamente diversos inconvenientes en la cooperativa. Inicialmente, se buscaba implementar Docker en los servidores de la cooperativa y se habían asignado tiempos de desarrollo dentro de los horarios laborales, ya que el módulo de Historial Financiero era un requerimiento del área de servicios financieros. Además, se estableció la necesidad de implementar servidores dockerizados en ambientes de estabilización,

---

certificación y producción para cumplir con las políticas de la cooperativa en cuanto a entornos de prueba.

No obstante, surgieron dificultades con la comunicación de red debido a las direcciones IP utilizadas, lo cual generó complicaciones. Dado que se trataba de un proyecto crítico y con una implementación compleja, se decidió adoptar un enfoque tradicional de desarrollo, elaborando un cronograma y entregando el producto completo al final del proceso.

Se procedió a generar el proyecto utilizando bases de datos locales, todo en un entorno dockerizado y desplegado en la nube de GCP para probar diversas hipótesis. Este enfoque permitió observar varios problemas de la metodología tradicional. Durante la entrega del producto, se presentaron múltiples errores y se debieron realizar cambios en las pantallas, solicitar nuevos campos y funcionalidades, además de identificar que algunos campos no contenían la información deseada. También surgieron problemas de coordinación con otros equipos, ya que todos compartían las mismas bases de datos y, a veces, los cambios en paquetes comunes afectaban el funcionamiento del sistema. La validación se tornaba más compleja, ya que el equipo de calidad debía probar módulos completos, y el equipo técnico identificó errores que llevaron a reestructurar el proyecto, lo que generó un retraso de aproximadamente cuatro semanas respecto a la fecha de entrega prevista para la salida a producción.

En contraste, al implementar ambientes dockerizados, aunque el avance inicial parecía más lento debido al tiempo dedicado a automatizar tareas, crear Dockerfiles, pipelines, y establecer la comunicación entre la base de datos, el backend y el frontend, se logró mayor eficiencia a largo plazo. La necesidad de aprender a manejar GCP, una plataforma nueva para el equipo añadió inicialmente una curva de aprendizaje pronunciada, lo cual ralentizó el progreso en comparación con el proyecto anterior, generando incertidumbre sobre la finalización del proyecto. Sin embargo, comenzaron a emerger ventajas significativas: al usar bases

---

de datos locales, se eliminó el riesgo de interferencia entre compañeros, se optimizó el tiempo de despliegue gracias a la automatización, y las reuniones diarias facilitaron el seguimiento de las actividades y la resolución de inconvenientes. Los cambios solicitados se implementaron de manera más ágil, lo cual permitió concluir el desarrollo del MVP en el plazo acordado.

En conclusión, aunque la adopción de una nueva metodología requiere una curva de aprendizaje más extensa en comparación con el enfoque tradicional, los beneficios se hicieron evidentes en términos de reducción de tiempos de entrega, mejora en la comunicación entre los equipos de la cooperativa (procesos, innovación, desarrollo y calidad), detección temprana de errores y menor costo de corrección al trabajarse con entregables periódicos. Además, la automatización de tareas permitió que todas las áreas se enfocaran en actividades de mayor valor y redujeran la carga de tareas repetitivas.

## 5.2 DISCUSIÓN DE LOS OBJETIVOS PLANTEADOS

En esta sección analizaremos en detalle los resultados obtenidos a partir de los objetivos planteados en la investigación. Nuestro propósito es evaluar en qué medida se han alcanzado dichos objetivos y, a partir de ello, determinar si las hipótesis formuladas al inicio del estudio han sido confirmadas o refutadas.

### 5.2.1 DEFINIR DE MANERA TEMPRANA LOS REQUISITOS

De acuerdo con el objetivo: “Definir de manera temprana los requisitos del módulo del Historial Financiero, mediante la utilización de Lean Inception para definir el mínimo producto viable (MVP) que se implementará en la COAC Jardín Azuayo”, se pudo identificar que los tiempos de construcción de los requisitos mejoraron significativamente, se pudo identificar a todos los usuarios del sistema, las funcionalidades que requieren los usuarios y las limitaciones del sistema.

A continuación, realizaremos un comparativo del antes y después de utilizar la metodología de Lean Inception.

<b>PARAMETRO</b>	<b>ANTES</b>	<b>DESPUES</b>
<b>Tiempo de construcción del requerimiento</b>	Antes de implementar la metodología de Lean Inception el tiempo para presentar un requerimiento nuevo era de aproximadamente un mes	Con la implementación de Lean Inception se mejoró el tiempo ya que tomó solamente una semana ya que ese es el tiempo que se necesita para aplicar la metodología.
<b>Personas que utilizarán el sistema</b>	Cuando se entregaba el sistema para pruebas de certificación o incluso en etapas de producción aparecían nuevos usuarios que necesitaban utilizar el sistema con sus propios requerimientos lo cual ampliaba el tiempo de desarrollo.	Al incluir desde las primeras etapas a todos los usuarios de todos los departamentos de la cooperativa se puede identificar quienes son los actores y cuáles son sus requerimientos.
<b>Funcionalidades.</b>	Para solicitar un requerimiento las áreas de Innovación de la cooperativa se reunían con el área solicitante del nuevo requerimiento y trataban de plasmar en un documento lo que necesitaba el área para posteriormente presentar dicho documento al área de desarrollo, aquí se tenían inconvenientes ya que al presentar la solución por parte del área de desarrollo al área solicitante muchas veces no era	Al reunirse desde un inicio tanto el área solicitante, el departamento de innovación y el departamento de desarrollo se puede plasmar de mejor manera el requerimiento del área solicitante, el equipo de desarrollo entiende oportunamente lo que se desea, también se pueden llegar a consensos sobre los entregables con funcionalidad deseada en cada uno de los sprint.

	lo requerido, ya que no se tenía la funcionalidad deseada por el usuario.	
<b>Entendimiento del sistema y experiencia de usuario</b>	Como se explicó en el punto anterior, antes de aplicar la metodología, se presentaba un documento a desarrollo, sin embargo, el equipo de desarrollo no tenía clara la funcionalidad del nuevo requerimiento, y se hacía lo que se entendía del documento, existían limitaciones en la tecnología ocupada (Oracle forms) y se diseñaban las pantallas de acuerdo con el gusto del desarrollador ya que no existía un prototipo de lo que deseaba y generaba inconformidad en el usuario final.	Al reunirse al inicio del requerimiento, todas las áreas que van a utilizar el sistema se mejora la comprensión por parte del equipo de desarrollo sobre qué es lo que requiere el usuario final, se puede explicar al usuario las limitaciones que puede llegar a tener el sistema, se pueden presentar prototipos de pantalla por parte del usuario final, de esta forma el usuario está contento ya que siente que sus peticiones son escuchadas y el equipo de desarrollo entiende que se espera del sistema.

Una de las principales dificultades que enfrentamos durante la implementación de la metodología fue la disponibilidad de tiempo por parte de los usuarios. Dado que las reuniones diarias eran esenciales para el desarrollo del proyecto, no siempre fue posible coordinar la asistencia de todos los participantes. Esta limitación impactó la continuidad del trabajo, generando retrasos en la toma de decisiones y afectando la eficiencia del proceso.

---

## 5.2.2 DISEÑO DE PROTOTIPOS DEL MVP

De acuerdo con el objetivo: “Diseño de prototipos del MVP del módulo del Historial Financiero, en base al Product Backlog, mediante Mockups, para obtener una retroalimentación del usuario con respecto al diseño planteado”.

Al aplicar la metodología de Lean Inception se pudo identificar las funcionalidades del sistema que son requeridas por el usuario, posterior a ello al trabajar con la metodología Scrum se pudo realizar un Product Back Log en el cual se definió el MVP a construir y las tareas de los primeros sprint, una de las tareas fue la construcción de Mockups con las pantallas a construirse, y que simulen la funcionalidad deseada por el usuario, para ello se utilizó la herramienta Figma, la cual fue muy útil para presentar un prototipo a los usuarios y que estos nos brinden una retroalimentación temprana de que les parece las pantallas, que falta y que campos están sobrando, y se pudo entender de mejor manera el flujo deseado, luego de algunas revisiones se llegó a un acuerdo en el que todas las partes estaban satisfechas con los prototipos.

## 5.2.3 SELECCIONAR HERRAMIENTAS ADECUADAS

De acuerdo con el objetivo: “Seleccionar las herramientas adecuadas para la puesta en marcha de Integración Continua y Despliegue Continuo, mediante la creación de un pipeline, repositorios de versionamiento, configuración de ambientes de desarrollo para automatizar las tareas, a fin de lograr un impacto significativo en la agilidad, la calidad y la eficiencia, del proceso de desarrollo”.

Para este punto se consideraron algunas herramientas (que las describimos a detalle en el capítulo 4.5 Configuración de Entornos de desarrollo), y tras realizar análisis de algunas herramientas, seleccionar las herramientas que se nos hagan más familiares y las que ya posee la cooperativa seleccionamos herramientas como PostgreSQL, GitLab, Docker, Google Cloud, Figma, Spring Boot entre otros por su

---

capacidad de integración, escalabilidad y alineación con los requerimientos del proyecto. Cada herramienta desempeñó un rol clave en la automatización y optimización del ciclo de desarrollo entre las principales ventajas tenemos:

- **Gestión Eficiente del Código y la Colaboración.** - GitLab permitió una gestión centralizada del código fuente, facilitando la integración continua (CI/CD) y asegurando un flujo de trabajo eficiente mediante control de versiones y automatización de despliegues. Además, Figma contribuyó significativamente a la colaboración en el diseño de interfaces, optimizando la comunicación entre desarrolladores y diseñadores.
- **Contenerización y Despliegue Automatizado.** - La implementación de Docker permitió la contenerización de aplicaciones, asegurando entornos de desarrollo homogéneos y facilitando la portabilidad entre servidores. Su combinación con Google Cloud optimizó la infraestructura, proporcionando escalabilidad y disponibilidad a las aplicaciones desplegadas.
- **Base de Datos y Desarrollo Backend.** - PostgreSQL se consolidó como la solución de base de datos más adecuada por su robustez, soporte para transacciones complejas y capacidad de manejar grandes volúmenes de datos. En cuanto al desarrollo backend, Spring Boot permitió agilizar la creación de microservicios, reduciendo tiempos de desarrollo y facilitando la integración con otras herramientas.

## 5.2.4 CONSTRUCCIÓN DE UN MVP

De acuerdo con el objetivo: “Construcción de un MVP o Producto Mínimo Viable del módulo del Historial Financiero, al automatizar procesos de desarrollo, pruebas, seguridades, para garantizar una automatización efectiva, una entrega y despliegue continuo de software de calidad que pueda servir de base para los futuros desarrollos en la cooperativa”

Al construir nuestro MVP del Historial Financiero, se pudieron establecer ciertos criterios claves que los podemos ver a continuación:

- 
- Automatización y Eficiencia en el Desarrollo. - La construcción del Producto Mínimo Viable (MVP) del módulo de Historial Financiero permitió optimizar los procesos de desarrollo mediante la integración de herramientas DevOps. La automatización de tareas clave, como la gestión del código, pruebas y despliegue, redujo significativamente los tiempos de entrega y minimizó los errores humanos.
  - Entrega y Despliegue Continuo. - La adopción de prácticas de integración y entrega continua (CI/CD) permitió una implementación ágil del software, asegurando que cada versión del módulo se desplegara de manera confiable en los diferentes entornos. Esto mejoró la capacidad de respuesta ante cambios y nuevas necesidades del sistema y nos facilitó como desarrolladores para tener cada uno un entorno de trabajo y no tener que depender de otros desarrolladores trabajando en las mismas bases que nosotros lo cual entorpece las pruebas.
  - Base para Futuros Desarrollos. - El MVP desarrollado no solo cumple con los requisitos iniciales, sino que también sirve como una base escalable y reutilizable para futuros módulos dentro de la cooperativa. La modularidad y estandarización del código facilitan la incorporación de nuevas funcionalidades sin afectar la estabilidad del sistema.
  - Impacto en la Cooperativa. - La automatización de procesos dentro del módulo de Historial Financiero optimizó la gestión y consulta de datos, mejorando la eficiencia operativa y la experiencia de los usuarios. Además, la implementación de metodologías ágiles y herramientas DevOps establece un precedente para la implementación de otros servicios en la cooperativa.

## 5.2.5 MEDIR LA PERCEPCIÓN DE LOS USUARIOS

De acuerdo con el objetivo: “Medir la percepción y satisfacción de los usuarios, a través de evaluaciones de satisfacción, comparación de tiempos de entrega de valor anteriores versus los tiempos actuales aplicando metodologías ágiles, para evaluar

---

si los usuarios están complacidos con el MVP del Historial Financiero entregado y poder obtener una retroalimentación para mejorar los siguientes sprints”

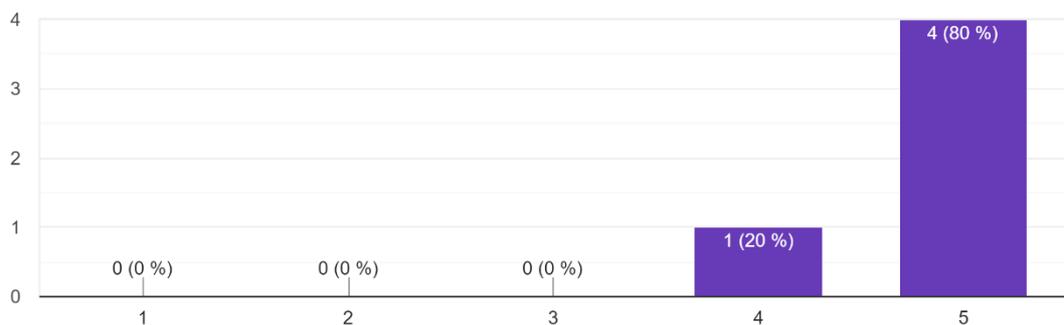
Para medir la percepción de los usuarios con respecto a la metodología Lean Inception se creó una encuesta con los participantes en el taller de Lean Inception, en dicha encuesta se plantearon preguntas para medir la satisfacción con respecto a los tiempos de desarrollo, si cumple el MVP con las necesidades del usuario, la evaluación de levantar las historias de usuario utilizando la nueva metodología, en la cual se observa que existe una gran satisfacción por parte del equipo de trabajo que formó parte de este proyecto, razón por la cual esta metodología será considerada para futuros proyectos que se implementen en la cooperativa debido a la comprensión por parte de todos los participantes de las funcionalidades del sistema, el corto tiempo en el que se puede determinar lo que se necesita hacer para el sistema y los sprints necesarios para crear el MVP del nuevo producto.

Se obtuvieron los siguientes resultados:

1. **¿Fue clara la metodología usada en los talleres de Lean Inception?** Con una escala del 1 al 5 siendo 1 mala y 5 buena se obtuvo el siguiente resultado en el que se observa que la percepción de la metodología implementada es buena

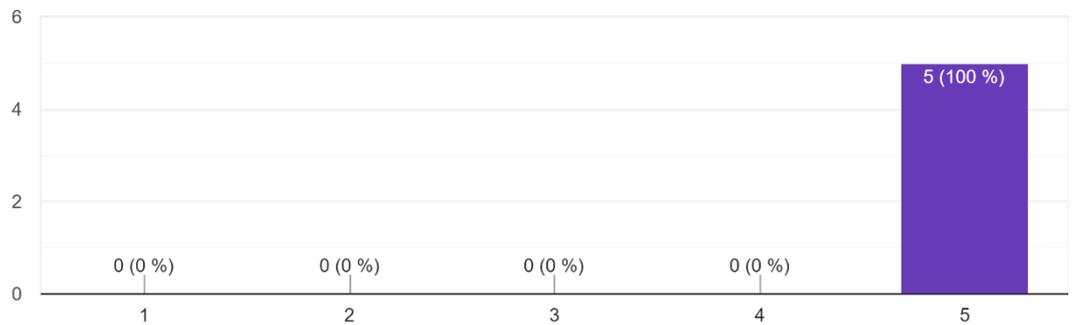
1.- ¿Fue clara la metodología usada en los talleres de Lean Inception?

5 respuestas



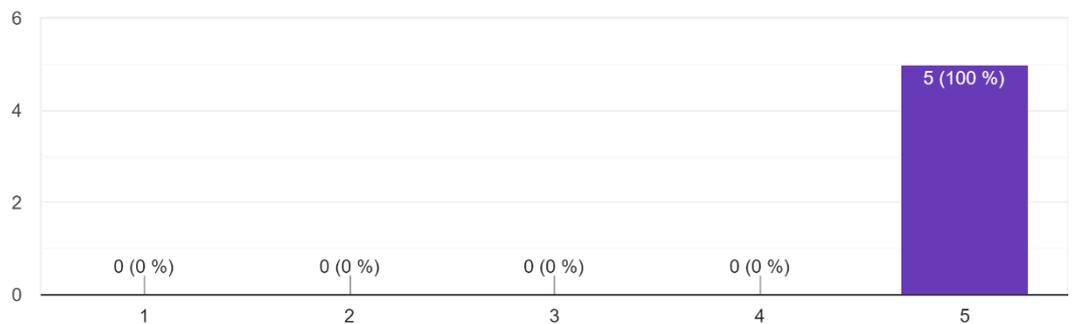
2. **¿Qué tan satisfecho se sintió con las sesiones para la implementación de Lean Inception?** Con una escala del 1 al 5, siendo 1 Muy Insatisfecho y 5 Muy Satisfecho se observa que todas las personas estuvieron muy satisfechas con las sesiones para la implementación de los talleres de Lean Inception

2.- ¿Que tan satisfecho se sintió con las sesiones para la implementación de Lean Inception?  
5 respuestas



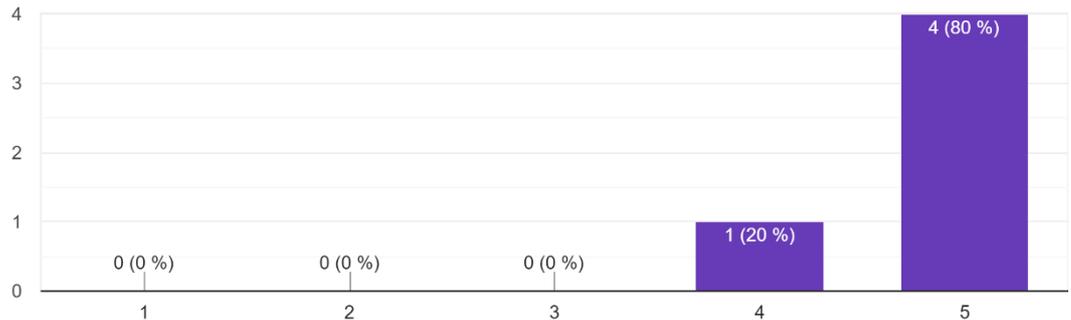
3. **¿Qué comprensión tiene del MVP que se va a desarrollar para el Historial Financiero?** Con una escala del 1 al 5, siendo 1 Muy mala y 5 Muy buena se observa que todos los integrantes comprenden que se va a implementar en el MVP para el Historial Financiero

3.- ¿Que comprensión tiene del MVP que se va a desarrollar para el Historial Financiero?  
5 respuestas



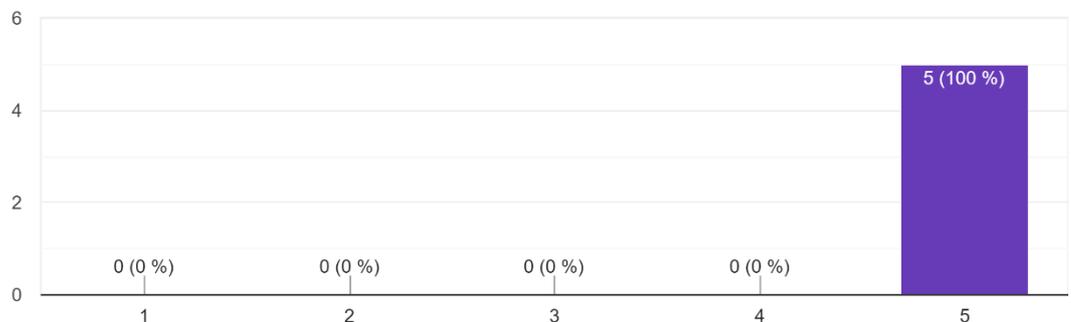
- 4. ¿Se tiene claras las necesidades de los usuarios al finalizar los talleres de Lean Inception?** Con una escala del 1 al 5, siendo 1 Poco Claras y 5 Muy Claras, se puede observar que los usuarios tienen claras las necesidades de los usuarios con respecto a la funcionalidad del Historial Financiero

4.- ¿Se tiene claras las necesidades de los usuarios al finalizar los talleres de Lean Inception?  
5 respuestas



- 5. ¿Está de acuerdo en que se debe implementar primero un MVP antes que el proyecto completo?** Con una escala del 1 al 5, Siendo 1 Totalmente en desacuerdo y 5 Totalmente de acuerdo se puede identificar que los usuarios comprendieron las ventajas de implementar un MVP antes que el sistema completo

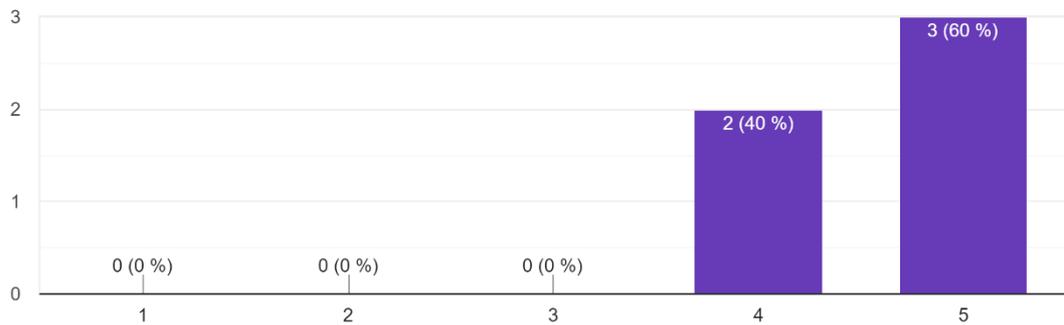
5.- ¿Está de acuerdo en que se debe implementar primero un MVP antes que el proyecto completo?  
5 respuestas



**6. ¿El MVP que se pretende realizar cumple sus necesidades?** Con una escala del 1 al 5, Siendo 1 Totalmente en desacuerdo y 5 Totalmente de acuerdo, se pudo identificar que no todos están de acuerdo con las necesidades que se van a solventar en el MVP, esto se debe a que se implementarán ciertas funcionalidades y no todas las que pretendía el usuario

6.- ¿El MVP que se pretende realizar cumple sus necesidades?

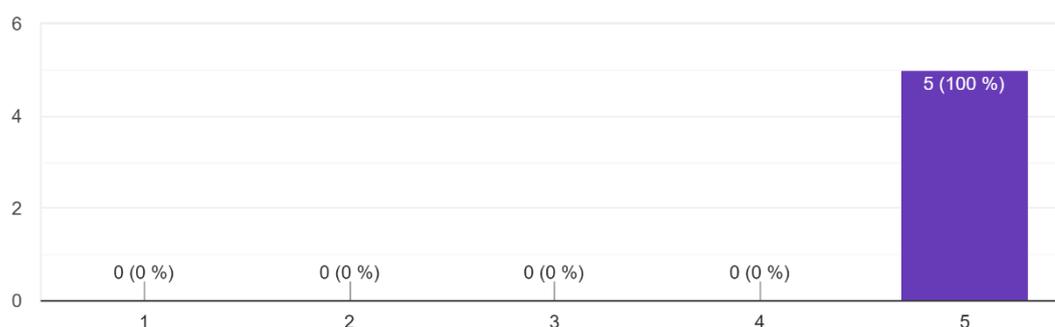
5 respuestas



**7. ¿Está Ud. de acuerdo en que utilizar Lean Inception mejoró la comprensión del producto que se pretende realizar?** Con una escala del 1 al 5, siendo 1 Totalmente en desacuerdo y 5 totalmente de acuerdo, se puede observar que con la implementación de Lean Inception todos los participantes comprendieron que se pretende realizar al implementar el Historial Financiero

7.- ¿Está ud de acuerdo en que utilizar Lean Inception mejoró la comprensión del producto que se pretende realizar?

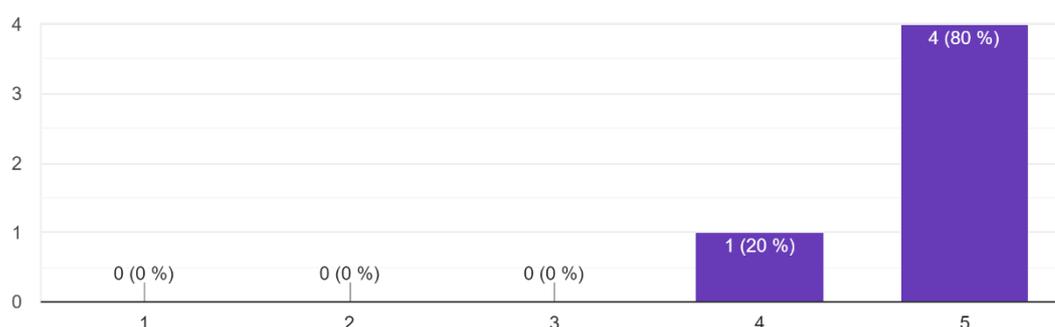
5 respuestas



**8. 8.- ¿Está de acuerdo en que utilizar Lean Inception mejoró los tiempos utilizados para entender el producto que se pretende entregar?** Con una escala del 1 al 5, siendo 1 Totalmente en desacuerdo y 5 Totalmente de acuerdo, se puede evidenciar que los usuarios están de acuerdo con que los tiempos de entrega mejoraron respecto a los tiempos que se enviaban con la metodología anterior.

8.- ¿Está de acuerdo en que utilizar Lean Inception mejoró los tiempos utilizados para entender el producto que se pretende entregar?

5 respuestas



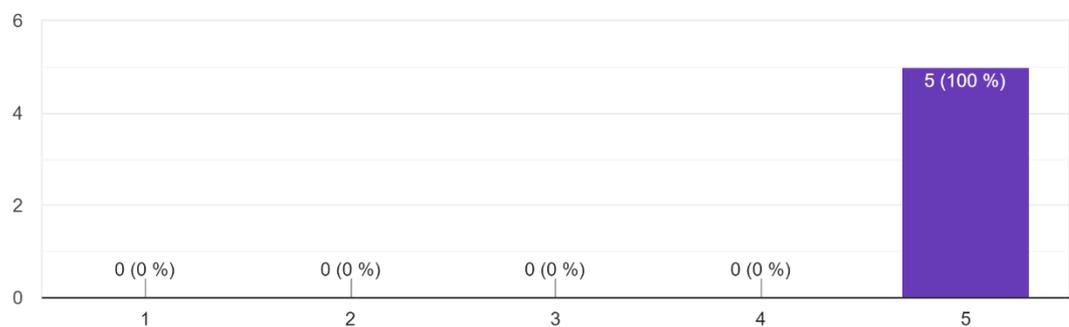
**9. ¿La implementación de Lean Inception ha ayudado a optimizar los tiempos de desarrollo?** Con una escala del 1 al 5, siendo 1 Totalmente en desacuerdo y 5 Totalmente de acuerdo, se puede observar que los usuarios están

---

contentos con los tiempos de la fase de desarrollo, ya que anteriormente siempre existían retrasos ya que no se consideraban correctamente los tiempos, había problemas en los cambios de ambientes y aparecían errores producto de funcionalidades no desarrolladas ya que no se entendía correctamente el requerimiento, este es uno de los puntos más relevantes para nuestra validación.

9.- ¿La implementación de Lean Inception ha ayudado a optimizar los tiempos de desarrollo ?

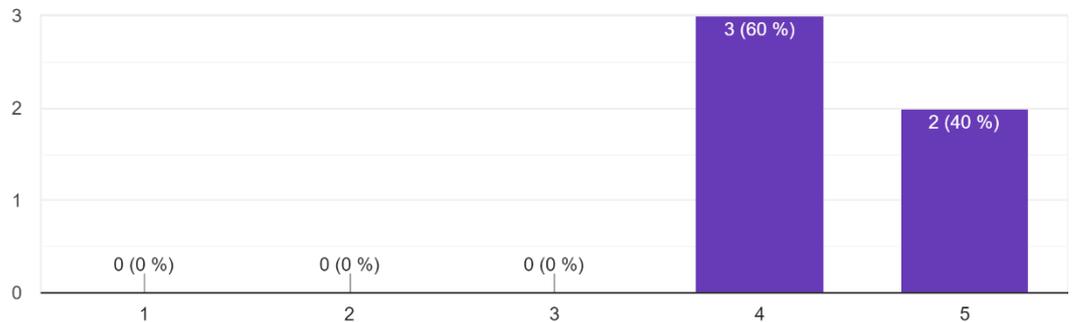
5 respuestas



**10. ¿Se han cumplido los tiempos ofrecidos para desarrollar el MVP?** Con una escala del 1 al 5, Siendo 1 Totalmente en desacuerdo y 5 Totalmente de acuerdo, Se puede observar que los usuarios están de acuerdo con los tiempos ofrecidos, ya no totalmente de acuerdo ya que hubo inconvenientes en los despliegues en ambientes de certificación en la cooperativa y nos tocó armar servidores de pruebas en otros ambientes.

10.- ¿Se han cumplido los tiempos ofrecidos para desarrollar el MVP?

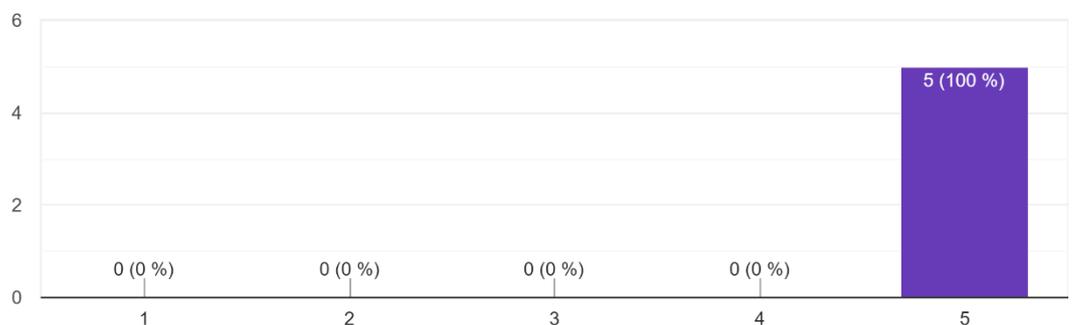
5 respuestas



**11. ¿Qué tan satisfecho está con la utilización de la metodología de Lean Inception?** Con una escala del 1 al 5, Siendo 1 Muy insatisfecho y 5 Muy satisfecho, se puede determinar que todos están satisfechos con la implementación de la metodología Lean Inception.

11.- ¿Que tan satisfecho está con la utilización de la metodología de Lean Inception?

5 respuestas



Se puede concluir con las encuestas realizadas que los usuarios están muy satisfechos con la implementación de Lean Inception en la cooperativa, ya que nos ayudó a reducir los tiempos de aprendizaje para entender los que los usuarios necesitan, así como incluir a todos los actores y levantar los requerimientos en tan solo una semana.

---

## 6. CONCLUSIONES.

---

Al implementar la metodología de Lean Inception se lograron levantar las historias de usuario en 1 semana, mientras que de la manera tradicional es una actividad que en la cooperativa toma 4 semanas, por tal motivo existe una reducción del 75% del tiempo lo cual permitirá levantar más historias de otros proyectos y también reducir los tiempos de entrega del software solicitado por las diferentes áreas de la cooperativa.

Al utilizar Lean Inception se pudo comprender de mejor manera por parte de todo el equipo de trabajo (personal de Procesos, de Innovación, de calidad y de desarrollo) que se pretende entregar a los usuarios, reduciendo la curva de aprendizaje comparada con la metodología tradicional cuando se escribía un documento en el cual se pretende plasmar que se requería implementar, en muchos casos olvidándose de ciertas funcionalidades requeridas y por otra parte entendiéndose mal por parte del equipo de desarrollo lo solicitado en las historias. Por lo tanto, se puede entregar un software de mejor calidad de acuerdo con las necesidades de la institución.

Al utilizar Lean Inception se plantea la creación de un MVP del Historial Financiero, al implementarlo se puede ver de manera rápida si es que lo desarrollado es lo que los usuarios pretendían, y se pueden realizar a tiempo todas las correcciones para los siguientes sprints, en cambio en la metodología tradicional estos cambios solo se los podía hacer al finalizar la entrega del producto completo y realizar cambios era una tarea muy complicada ya que no se contaba con el tiempo suficiente y se tenía que cambiar en demasiados lugares la codificación.

Al iniciar la implementación de las ceremonias de Scrum eran unas tareas que parecían aburridas, no se tenía el compromiso total de los miembros del equipo, pero poco a poco se fue viendo la efectividad de las mismas y que ya no se tenía problemas de cuellos de botella, o falta de comunicación entre el equipo, de igual

---

manera se obtenía una retroalimentación temprana sobre el producto y también se podía ver los puntos de fallo y de mejora para poder trabajar de una manera más rápida y efectiva.

Al utilizar herramientas como ClickUp se pudo evidenciar por parte de todo el equipo de trabajo las actividades que se encontraban haciendo los miembros del equipo, así como evidenciar en que etapa se encontraba el proyecto y las tareas que estaban retrasadas para darles un tratamiento especial a fin de no atrasar la entrega propuesta del proyecto.

La implementación de un flujo de trabajo DevOps utilizando GitLab CI/CD en combinación con Google Cloud Platform (GCP) ha sido fundamental para la construcción eficiente del MVP del sistema de historial financiero. Gracias a la integración automatizada de procesos de compilación, pruebas, y despliegue, se logró mejorar la productividad del equipo y optimizar el tiempo de desarrollo. La adopción de GCP para la infraestructura en la nube permitió garantizar escalabilidad y disponibilidad, aspectos clave para un proyecto que maneja datos sensibles y debe mantener altos estándares de seguridad y rendimiento.

El uso de contenedores Docker en la configuración y despliegue de los servicios backend y frontend en Cloud Run, junto con la configuración de bases de datos en Cloud SQL, consolidó una arquitectura flexible y modular. Además, esta configuración facilita futuras actualizaciones o expansiones del MVP sin impactar en la experiencia de usuario.

A medida que el sistema evolucione, se pueden explorar mejoras adicionales, como la implementación de monitoreo continuo y alertas de seguridad, optimizando aún más el despliegue en producción. En conclusión, esta arquitectura DevOps basada en GitLab y GCP proporciona una base sólida para el crecimiento y escalabilidad del sistema, sentando un precedente para futuras aplicaciones y asegurando una experiencia ágil y eficiente en el desarrollo de software financiero.

La aplicación de las métricas DORA permite evaluar de manera precisa el estado actual de los procesos en la cooperativa en comparación con los beneficios

---

---

obtenidos tras la implementación de prácticas DevOps. Al medir indicadores clave como la frecuencia de implementaciones, el tiempo de ciclo, la tasa de fallas de implementación y el tiempo de recuperación, se evidencian mejoras significativas en el rendimiento de los procesos de desarrollo.

Sin DevOps, los procesos presentaban largos tiempos de ciclo, baja frecuencia de despliegues, una mayor tasa de fallas, y tiempos de recuperación extensos, lo cual afectaba tanto la eficiencia operativa como la experiencia del usuario. Sin embargo, con la adopción de DevOps, se observa una notable optimización en cada uno de estos aspectos: los despliegues se realizan de manera continua y frecuente, el tiempo de ciclo se ha reducido, la estabilidad de los despliegues ha mejorado, y los tiempos de recuperación ante fallas se han acortado drásticamente.

En conclusión, la implementación de DevOps en la cooperativa ha transformado sus procesos, alineándolos con los estándares de alto rendimiento que promueven las métricas DORA. Este cambio no solo incrementa la eficiencia y la confiabilidad del ciclo de desarrollo, sino que también refuerza la capacidad de respuesta ante incidentes, contribuyendo a una mayor satisfacción y confianza de los usuarios.

## 6.1 DIFICULTADES PRESENTADAS.

Entre las principales dificultades que se tuvieron durante la implementación de la tesis tenemos:

Al momento de realizar las reuniones de Lean Inception fue complicado coordinar los horarios de todas las áreas involucradas, ya que no le veían como algo importante y asignaban sus tiempos a otras actividades, lograr cambiar la mentalidad de las personas y convencerles que el trabajo que se realiza en los talleres pese a que es cansado y parece que es un tiempo perdido al final es mucho tiempo que se ahorra ya que se entiende claramente el requerimiento, se identifican las funcionalidades y se logra descubrir todos los actores que utilizarán el sistema.

---

Al momento de construir los Mockups para las pantallas a entregarse en el MVP, no se tenía plantillas, no se tenía los botones a utilizar, la distribución de estos, las fuentes de las letras, ni un conocimiento en el manejo de la herramienta Figma, por lo cual se tuvo que aprender sobre la marcha y construir los prototipos desde cero lo cual demandó bastante tiempo.

Al implementar las herramientas para el desarrollo fue un reto, ya que no sabíamos utilizar muchas cosas por ejemplo Dockers, que gracias a la maestría nos dieron las pautas principales para su utilización, sin embargo, también se identificaron desafíos, como la curva de aprendizaje inicial y la necesidad de configurar adecuadamente los entornos para maximizar el rendimiento.

Fue complicado convencer a las personas que íbamos a implementar solo un MVP, ya que la cultura de la cooperativa siempre se entrega los sistemas completos, pasando algunos meses en desarrollo, luego en estabilización y certificación para finalmente pasar a producción y en esa etapa es cuando aparecían los errores o inconformidades con el sistema.

Una de las mayores dificultades en la implementación de DevOps fue que al inicio íbamos a poder desarrollar el Historial Financiero y podíamos instalar en los ambientes de Docker que nos iban a facilitar en la cooperativa, tanto en ambientes de estabilización como certificación y se nos iba a permitir desarrollar con este enfoque, pero al existir otras demandas en la cooperativa se tuvieron que realizar sobre la tecnología actual, por este motivo tuvimos que desarrollar el Historial Financiero sin implementación de Docker y el módulo completo para subir a ambientes productivos y paralelamente nos tocó generar el Historial Financiero en Docker y subirlos a servicios de Google en la nube para validar las teorías.

## 6.2 POSIBLES MEJORAS.

Cuando se necesite un nuevo requerimiento se pudiera coordinar con los jefes de las diferentes áreas para que se asigne un tiempo a los integrantes que realizarán

---

---

el nuevo requerimiento, de esta manera podrán realizar el taller de Lean Inception sin descuidar sus actividades diarias y sin tener que trabajar horas extras por asistir a los talleres.

Se debe estandarizar la construcción de Mockups, en la cual se tenga ya plantillas bases para los botones, grillas, cuadros de texto, los iconos de los botones para cada una de las funcionalidades, de esta manera se tendrán diseños más homogéneos y será más fácil para el usuario acostumbrarse a las nuevas pantallas del sistema.

Se recomienda continuar explorando mejoras en la integración de herramientas, especialmente en la automatización de despliegues y monitoreo de infraestructura. Además, fomentar la capacitación en prácticas DevOps permitirá optimizar aún más los flujos de trabajo y la eficiencia operativa.

Finalmente, como se pudo realizar con éxito el Historial Financiero y cumplir con los tiempos establecidos, mejorar la comunicación entre los equipos, tener menos errores al cambiarnos de ambientes, tener un código más ordenado, y una mejor forma de trabajo, la cooperativa tomará como base la tesis realizada con la intención de implementar un plan piloto para construir un nuevo sistema con la intención de migrar el Core principal, en el cuál va a ser muy importante implementar metodologías como Lean Inception para poder identificar a tiempo los requisitos, los usuarios que van a utilizar el sistema, y las principales funcionalidades con el objetivo de crear un MVP, también se debe utilizar una metodología como Scrum para poder realizar un seguimiento y un desarrollo rápido del producto, se pretende utilizar Dockers, una vez se tenga un MVP del nuevo sistema se pretende someter a pruebas de carga para identificar si va a poder soportar toda la transaccionalidad, si son efectivas las pruebas se pretende migrar toda nuestra tecnología, y en caso de que falle buscar nuevas alternativas ya no construyendo todo el sistema (que tomaría años) sino MVP que permitan probar las teorías de una manera más rápida y sin perder tanto tiempo para validar.

---

## 7. BIBLIOGRAFÍA.

---

- Akbar, M. A. (2024). DevOps project management success factors: A decision-making framework. *Software: Practice and Experience*, 54(2), 257--280.
- Amitkumar V. Jha, R. T. (2023). From theory to practice: Understanding DevOps culture and mindset. *Cogent Engineering*, 10(1), 2251758. doi:10.1080/23311916.2023.2251758
- Apache, F. d. (s.f.). *Maven*. Obtenido de <https://maven.apache.org/what-is-maven.html>
- Appium. Documentación oficial. (2023). *Appium Documentation*. Obtenido de Appium Documentation: <https://appium.io/docs/en/2.2/intro/>
- Appium.io. (17 de 02 de 2014). *Documentación de Appium*. Obtenido de Documentación de Appium: <https://appium.io/docs/ja/latest/>
- Atlassian. (2024). *devops*. Obtenido de <https://www.atlassian.com/es/devops/what-is-devops>
- Babativa Novoa, C. A. ( 2017). *Investigación cuantitativa*.
- Baertschi, S. a. (2023). A Method for the Adoption of DevOps in the Banking Industry. *2023 International Conference on Information Management (ICIM)*. doi:10.1109/ICIM58774.2023.00012
- Bird, J. (2017). *DevOps for finance*. O'Reilly Media.
- Brennan Wilkes, A. M.-A. (2023). Un marco para automatizar la medición de métricas de investigación y evaluación de DevOps (DORA). *Conferencia internacional IEEE 2023 sobre mantenimiento y evolución de software (ICSME)*. Bogotá, Colombia: IEEE.
- Brown, S. (s.f.). *C4 Model*. Obtenido de <https://c4model.com/diagrams/system-context>
- Calles-García, J., & González-Pérez, P. (2011). *La Biblia del Footprinting*.
- Caroli, P. (2018). *Lean Inception Creando Conversaciones hacia un producto exitoso*. Porto Alegre.
- Caroli, P. (2019). *Lean inception: creando conversaciones hacia un producto exitoso*. Sao Paulo: Editora Caroli. doi:978-85-94377-09-8
- Cusco Mejía, B. F. (2022). *Desarrollo e implementación de una arquitectura DevOps para un sistema web basado en microservicios en infraestructuras basadas en código (Bachelor's thesis)*.
- DAVILA NICANOR, et al. (2015). DAVILA NICANOR, L. E. T. I. C. I. A., Marín Guerrero, O., Aguilar Juárez, I., & AYALA DE LA VEGA, J. O. E. L. Obtenido de Pruebas de Regresión Funcional Mediante el Uso de Patrones de Diseño.: <https://appium.io/docs/en/2.2/intro/>
- Debois, P. (2008). *Agile Infrastructure and Operations: How Infra-gile are You?* doi:10.1109/Agile.2008.42
- Easy, S. (02 de 01 de 2020). *What is Selenium WebDriver | Architecture, Advantage*. Obtenido de What is Selenium WebDriver | Architecture, Advantage: <https://www.scientecheasy.com/2020/07/selenium-webdriver-architecture.html/>
-

- 
- Engström, et al. (2010). *A systematic review on regression test selection techniques. Information and Software Technology, 52(1), 14-30.*
- Esquivel, A. L. E. . (2023). *Clasificación de variables. CISA, 4(4), 43-53.*
- Farley, J. H. (2010). *Reliable Software Releases through Build, Test, and Deployment Automation.* Addison-Wesley Professional.
- Foundation, L. (s.f.). *OPENAPI INITIATIVE.* Obtenido de <https://www.openapis.org/what-is-openapi>
- Foundation, OpenJS. (17 de 02 de 2024). *Appium.io.* Obtenido de Introduccion: <https://appium.io/docs/en/latest/>
- Fradejas Pascual, D. (2019). *Sistema de integración continua para el desarrollo del software y hardware en servidores y estaciones de trabajo de empresa.*
- García Gutiérrez, et al. (2021). *Automated Driver Management for Selenium WebDriver.*
- Garg, S. (2016). *Appium Recipes. .*
- GitLab. (2024). Obtenido de GitLab: <https://about.gitlab.com/es/topics/ci-cd/cicd-pipeline/#ci-cd-pipeline-overview>
- Gojare, S., et al. (2015). *Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. Procedia Computer Science, 50, 341-346.*
- Google. (s.f.). *Descripción general de Google Cloud.* Obtenido de [https://cloud.google.com/docs/overview?hl=es-419&\\_gl=1\\*1xkx437\\*\\_up\\*MQ..&gclid=Cj0KCQjwo8S3BhDeARIsAFRmkOMpqSX8Rqnp4JS8YBI91-8LckFojjzdGQ-TGIMMLnC\\_TO7rY4GQLUaAgpLEALw\\_wcB&gclsrc=aw.ds](https://cloud.google.com/docs/overview?hl=es-419&_gl=1*1xkx437*_up*MQ..&gclid=Cj0KCQjwo8S3BhDeARIsAFRmkOMpqSX8Rqnp4JS8YBI91-8LckFojjzdGQ-TGIMMLnC_TO7rY4GQLUaAgpLEALw_wcB&gclsrc=aw.ds)
- Gundecha, U. (2015). *Selenium Testing Tools Cookbook. Packt Publishing Ltd.*
- Gutierrez Zapata, D. L. ( 2021). Implementación de plataforma de automatización de procesos usando “Selenium Web Driver” para optimizar las pruebas de regresión en San Isidro.
- Hans, M. (2015). *Appium Essentials.*
- Herrera Pérez, S. (2017). *Análisis e Implementación de la Integración Continua en empresas de software.*
- Inc, D. (s.f.). *docker-overview.* Obtenido de <https://docs.docker.com/get-started/docker-overview/>
- Kanglin Li, M. W. (2006). *Effective Software Test Automation.* Wiley.
- Kim, G. a. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.* IT Revolution Press.
- Kim, G., Humble, J., Debois, P., & Willis, P. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.*
- Kumar, M., Sharma, A., & Kumar, R. (2014). *Fuzzy entropy-based framework for multi-faceted test case classification and selection: An empirical study. IET Software, 8(3), 103-112.*
- Leticia, et al. (s.f.). *Pruebas de Regresión Funcional Mediante el Uso de Patrones de Diseño.*
- López Cano, José L. . (1989). *Método e Hipótesis Científicos. Trillas. México.*
- Macarthy, R. W. (2021). The Role of Skillset in the Determination of DevOps implementation Strategy. doi:10.1109/ICSSP-ICGSE52873.2021.00015}
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software.*
-

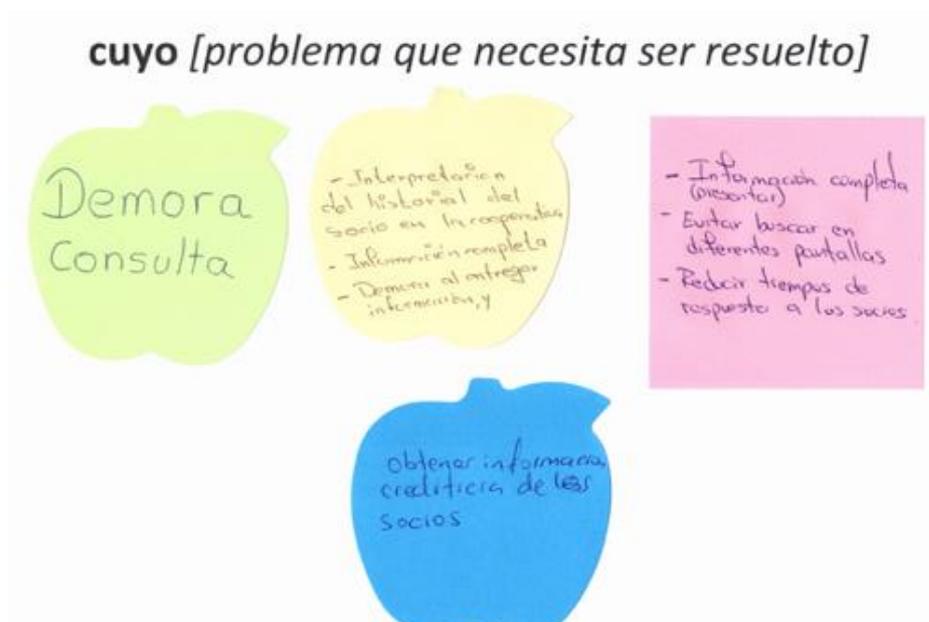
- 
- Masache Romero, J. G. (2021). *Uso de Scrum, Devops y MobileD en el desarrollo de aplicaciones móviles (Bachelor's thesis)*.
- Mena Barrera, M. (2021). *Análisis de herramientas para realizar pruebas automatizadas a aplicaciones móviles desarrolladas en Flutter*.
- Mena Barrera, M. F. . (2021). *Análisis de herramientas para realizar pruebas automatizadas a aplicaciones móviles desarrolladas en Flutter*.
- Microsoft. (s.f.). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com/docs>
- Morales, et al. (2017). *Desarrollo de un modelo de pruebas funcionales de software basado en la herramienta SELENIUM*. *Industrial data*, 20(1), 139-147.
- Morales, et al. (2021). *Research in the area of Software Quality: A Mapping Study of the Last Decade*. *2021 Mexican International Conference on Computer Science (ENC)*, 1-6.
- Murthy, A. C. (2019). *Mobile Application Development, Usability, and Security*. IGI Global.
- Oracle. (s.f.). *JAVA*. Obtenido de [https://www.java.com/en/download/help/whatis\\_java.html](https://www.java.com/en/download/help/whatis_java.html)
- Orozco Díaz, J. A. (2022). *Implementación de pruebas funcionales automatizadas con Selenium, Appium y Rest-Assured en la empresa Valid Colombia*.
- Paños Medina, A. (2021). *Desarrollo de arquitectura genérica para pruebas automatizadas en un marco de desarrollo de software dirigido por modelos*. Valencia.
- Parsa, S., & Khalilian, A. . (2009). *A Bi-objective Model Inspired Greedy Algorithm for Test Suite Minimization*. En Y. Lee, T. Kim, W. Fang, & D. Ślęzak (Eds.), *Future Generation Information Technology (Vol. 5899, pp. 208-215)*.
- Pérez, et al. (2022). *Desarrollo de pruebas funcionales con Selenium WebDriver y Python*. *Serie Científica de la Universidad de las Ciencias Informáticas*, 15(5), 23-40.
- Pham, H. (2003). *Regression Testing of Software Components*. Springer.
- Pressman, R. S. (2010). *INGENIERÍA DEL SOFTWARE 7ED UN ENFOQUE PRÁCTICO*. University of Connecticut. McGraw-Hill.
- Ramesh, Srinivasan Desikan y Gopalaswamy. (2016). *Software Testing: Principles and Practices*. Pearson.
- Rivera Martínez, C. A. (2018). *Automatización de pruebas de regresión*. .
- Rodríguez Burgos, K. E. (2012). *Investigación cuantitativa: Diseño, técnicas, muestreo y análisis cuantitativo*. *Investigación Cuantitativa: Diseño, técnicas, muestreo y análisis cuantitativo*, 137-157.
- Rothermel, et al. (1994). *A framework for evaluating regression test selection techniques*. *Proceedings of 16th International Conference on Software Engineering*, 201-210.
- Rothermel, G., & Harrold, M. J. (1993). *A safe, efficient algorithm for regression test selection*. *1993 Conference on Software Maintenance*, 358-367.
- Selendroid Selenium for android*. (s.f). Obtenido de Selendroid Selenium for android: <http://selendroid.io/>
- Selenium Developers Group. (2016). *Selenium Web Browser Automation*.
- Su, L. a. (2023). *A Case Study of DevOps Adoption within a Large Financial Organisation*. doi:10.1109/ICSME58846.2023.00053}

- 
- Ted Schadler, J. B. (2014). *he mobile mind shift: Engineer your business to win in the mobile moment*. Greenleaf Book Group.
- Tran, H. M. (2023). Automation Testing with Appium Framework in IP Multimedia Subsystem. *14th International Conference on Information and Communication Technology Convergence (ICTC)*, 579-582.
- Vega Llobell, A. T. (2018). *Pruebas funcionales automatizadas para aplicaciones Web: Usando Selenium para aplicar pruebas de regresión automatizadas (Doctoral dissertation, Universitat Politècnica de València)*.
- Velásquez, S. M., Sossa, D. E. M., Zapata, M. E., Adasme, M. E. G., & Ríos, J. P. . (2019). *Pruebas a aplicaciones móviles: avances y retos. Lámpsakos (revista discontinuada)*, (21), 39-50.
- Verma, N. (2017). *Mobile Test Automation With Appium*. Packt Publishing Ltd.
- Webb, R. T. (2023). Architectural views. 20.
- www.elhacker.net. (s.f.). *www.elhacker.net*. Obtenido de [https://www.elhacker.net/trucos\\_google.html](https://www.elhacker.net/trucos_google.html)
- Zein, et al. (2016). *A systematic mapping study of mobile application testing techniques*.
- Zúñiga, et al. (2023). *Metodología de la investigación científica: guía práctica. Ciencia Latina Revista Científica Multidisciplinar*, 7(4), 9723-9762.

Caroli, P. (2018). Lean inception: creando conversaciones hacia un producto exitoso. [www.caroli.org](http://www.caroli.org)

Nasreen Azad, J. P., & Hyrynsalmi, S. (2023). DevOps critical success factors — A systematic literature review. *Information and Software Technology*, 157, 107150. <https://doi.org/https://doi.org/10.1016/j.infsof.2023.107150>

## 8 ANEXOS.



**es una [categoría de producto]**



**El nombre del producto]**



**eso [beneficios clave, motivo para comprarlo]**

A mind map for 'eso' featuring a central pink square and four apple-shaped notes. The pink square contains the text: '- Disminuir tiempo en oficina para atención a los socios'. The green notes contain: 'Conocer mejor socio' and 'Rapidez Consulta'. The yellow note contains: '- Evitar demoras en la entrega de información' and '- Entendible'. The blue note contains: 'Califico al socio - obtener información real'.

**nuestro producto [diferencia clave]**

A mind map for 'nuestro producto' featuring a central pink square and three apple-shaped notes. The pink square contains the text: '- Información completa de créditos'. The blue note contains: 'Acceso unico a la información'. The yellow note contains: 'Información Adecuada.'. The green note contains: 'Información Detallada'.

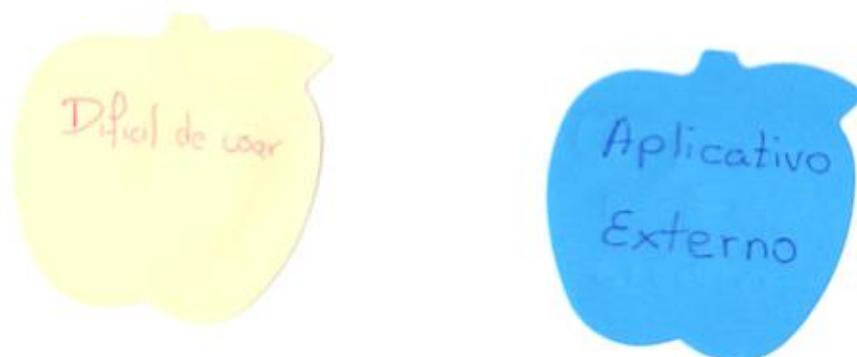
**nuestro producto [diferencia clave]**

A mind map for 'nuestro producto' featuring a central pink square and three apple-shaped notes. The pink square contains the text: '- Información completa de créditos'. The blue note contains: 'Acceso unico a la información'. The yellow note contains: 'Información Adecuada.'. The green note contains: 'Información Detallada'.

Escribiendo la Visión del producto.



## NO ES



Apellido # OC

Información detallada de operación crediticia

Muestra Créditos Socio

## HACE

Días  
Mora de mora  
de crédito  
papeo / garantías

→ Me de los día de mora por cuota y por meses. NO acumulados.

Buena  
Información del socio  
credito

Presentar resumen  
Días mora

→ Que me envíe a la pantalla de un pago judicial por desmor sus pagar reducidos

Presentar Garantías atadas a la OC

→ Me envíe para ver las gras del conyuge sus citados.

## HACE

→ Me de Información de las demandas Judiciales (Asesor Judicial)

→ Me de la información si puede acceder a créditos (caso conyugal con demanda.)

Pantalla para verificar nivel de endeudamiento interno del socio/garante

No dirige a más información de forma directa.

Consulta Buró de Crédito

Sistema de base  
→ Nivel de endeudamiento interno y externo

## NO HACE

Presenta datos específicos del caso en una sola pantalla

Presentar Datos del Socio Información

Transacciones financieras

Edita Información

Modifica Datos

## NO HACE

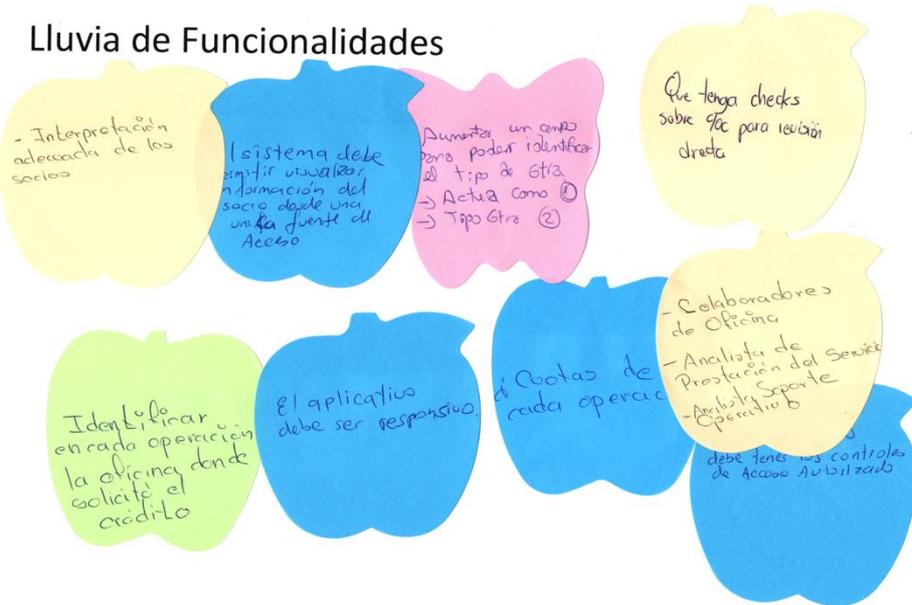
~~Activación de~~  
Solicitudes de Créditos

Activación de Cuenta

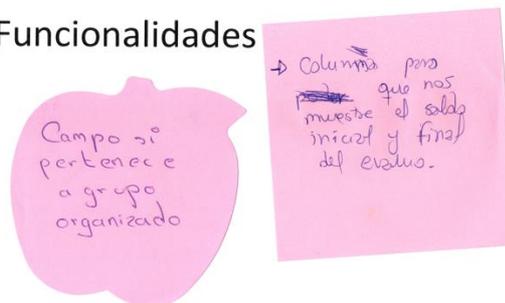
No detalla adeudo crédito los pagos realizados

Descripción del producto.

## Lluvia de Funcionalidades



## Lluvia de Funcionalidades



Brainstorming de las funcionalidades

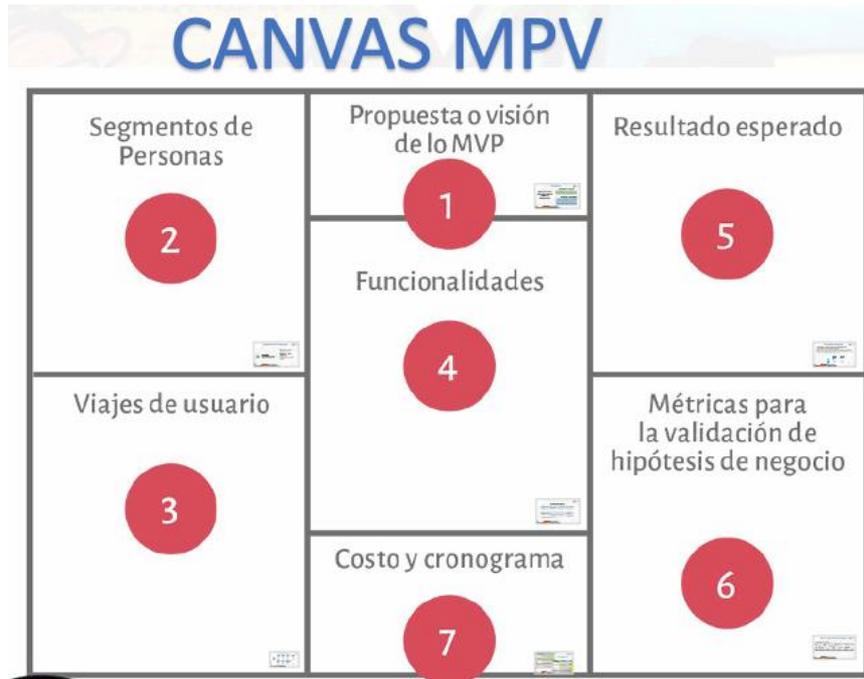


Imagen 13 Canvas MVP.

## Segmentos de Personas

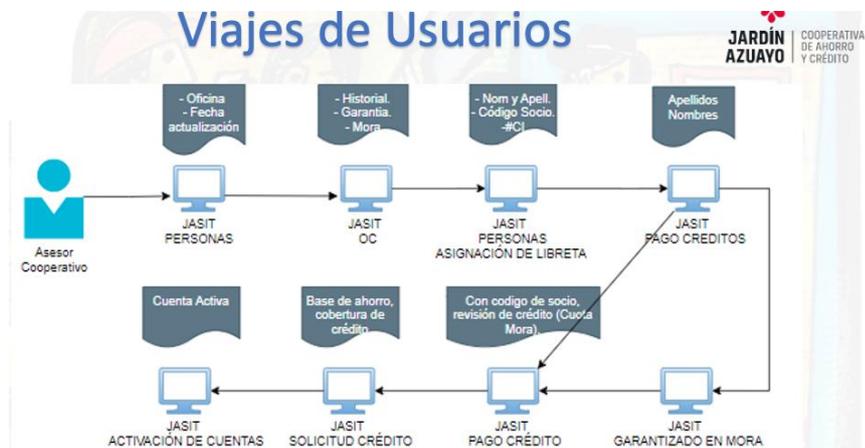


### ASESOR COOPERATIVO

- Asesor de Centro de Atención Virtual
- Analista de Soporte Operativo
- Analista de Innovación y Diseño

**JARDÍN AZUAYO** COOPERATIVA DE AHORRO Y CRÉDITO

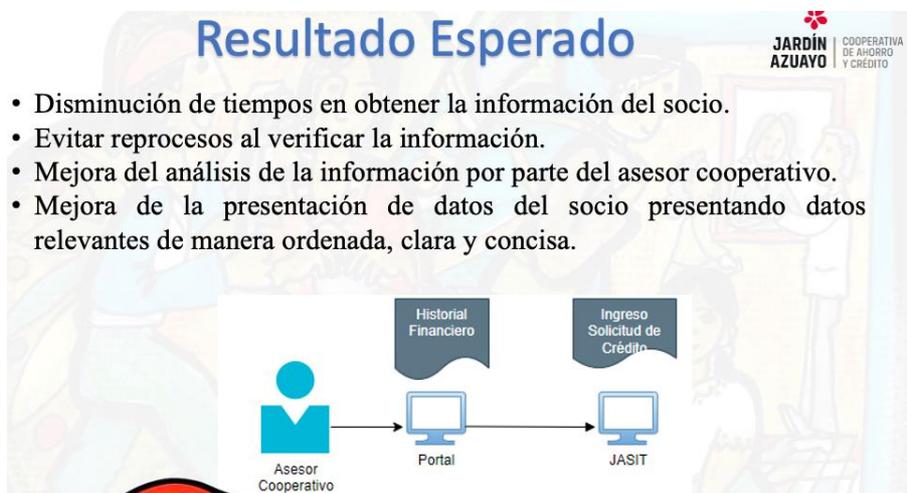
Segmento de personas



Viajes de usuario actual historial financiero.

FUNCIONALIDADES				
1	Mantener campos OC del JASIT	Ingresar con Número de identificación o código de socio	Filtros de estado de crédito	Nombre: Información Financiera
2	Enlace al detalle de Pagos de Crédito	Cuenta Activa	Oficina a la que pertenece el socio	Saldos promedio
3	Enlace al detalle de los avances de la Línea de Crédito	Muestra la decisión de la operación crediticia	Valor de la cuota de cada operación	Saldo de Certificados de aportación
4	Providencias Judiciales	Se puede imprimir para adjuntar a la solicitud de crédito	Garantía hipotecaria	Plazo de vencimiento de la operación crediticia

Secuenciador de funcionalidades.



Resultado Esperado.

# Métricas para la Validación del negocio

Tiempos de atención al socio por observación

Petición del socio	JASIT		Información Financiera					Tabla de Crédito /Excel	Portal	despedida	Observaciones	Tiempo Total
	Personas /Oficina	OC	Jasit/ Cambio Oficina	Buro	Estado de Cuenta	Pago Crédito	Jasit/ Saldos promedio					
9:54:00	9:54:00	9:56:00					9:56:00	9:57:00	9:57:00	10:05:00	Asesoría(pre)	0:11:00
10:28:00	10:29:00	10:29:00		10:30:00				10:30:00		10:32:00	Asesoría	0:04:00
10:40:00	10:41:00	10:41:00		10:42:00		10:45:00		10:44:00		10:46:00		0:06:00
10:58:00	10:58:00	10:59:00		11:00:00				11:01:00		11:03:00	Sin HC	0:05:00
11:52:00	11:53:00	11:53:00						11:53:00		12:05:00	Inf. Emergente	0:13:00

Imagen 19 Métricas para validación de negocio.

N°	Descripción de la tarea	Duración de la etapa (días)	Tarea dependiente	Comienzo Planificado	Fin Planificado	Responsable
<b>Análisis - Investigación</b>						
1	Informe de Análisis - Investigación (Documento - presentación)	5	No Aplica	01/02/24	08/02/24	Mónica González
2	Presentación a la Dirección (power Point)	1	1	16/02/24	19/02/24	Mónica González
3	Presentación al Comité (power point)	1	1	16/02/24	19/02/24	Mónica González
<b>Solicitud para construir</b>						
5	Esperar para iniciar la construcción.	1	3	19/02/24	20/02/24	Mónica González
<b>Diseño</b>						
7	Definición del flujo, características del servicio y diseño pantallas inicial, Historias usuario base (documento borrador)	4	5	19/02/24	23/02/24	
8	Historias de Usuario /Criterios de Aceptación. (documento-firma)	2	7	21/02/24	23/02/24	
9	Especificación Técnica Integral(Documento y aprobación)	2	7	21/02/24	23/02/24	
10	Plan de pruebas técnicas y funcionales incluido listado de personas que dan el ok para subir a producción. (documento)	2	7	21/02/24	23/02/24	
11	Plan de promoción (documento) y solicitud de imágenes.	2	7	21/02/24	23/02/24	
<b>Desarrollo</b>						
13	Actividades de desarrollo	20	10	26/02/24	26/03/24	Eduardo Aguilar
14	Pruebas Unitarias	1	13	22/03/24	25/03/24	Eduardo Aguilar
15	Manual de instalación (documento)	21	10	23/02/24	26/03/24	Eduardo Aguilar
16	Estabilización Funcional (acta)	4	14	25/03/24	29/03/24	Yesmenia Apolo
17	Estabilización Técnica (acta)	4	14	25/03/24	29/03/24	Cristian Idrovo
18	Estabilización Seguridad (acta)	4	14	25/03/24	29/03/24	NO
19	Validación de imágenes	4	14	25/03/24	29/03/24	NO
20	Actualización de la documentación	25	10	22/02/24	28/03/24	Todos
21	Prerequisitos y pase a certificación	4	14	26/03/24	01/04/24	Eduardo Aguilar
22	Despliegue y validación	1	21	29/03/24	01/04/24	Cristian Idrovo

## Cronograma

<b>Certificación</b>						
24	Certificación Funcional (acta)	4	22	01/04/24	05/04/24	Yesmenia Apolo
25	Certificación Técnica (acta)	4	22	01/04/24	05/04/24	Cristian Idrovo
26	Certificación Seguridad (acta)	4	22	01/04/24	05/04/24	NO APLICA
27	Capacitación Técnica (acta)	4	22	02/04/24	08/04/24	Desarrollo
28	Capacitación Operativa (acta)	4	22	05/04/24	11/04/24	Yesmenia Apolo
29	Manual funcional / instructivo (documento)	4	22	05/04/24	11/04/24	Yesmenia Apolo
30	Insumos promocionales	4	22	05/04/24	11/04/24	Mónica González
31	Preparación de prerequisites	4	22	05/04/24	11/04/24	Cristian Idrovo
32	Plan de Implementación de software (documento)	4	22	08/04/24	12/04/24	Producción
33	Actualización de la documentación	4	22	05/04/24	11/04/24	Todos
34	Autorización de pase a producción	1	24	11/04/24	12/04/24	Mónica González
35	Pase a producción	1	34	11/04/24	12/04/24	Cristian Idrovo
<b>Despliegue</b>						
37	Puesta en producción	1	35	12/04/24	15/04/24	Producción
<b>Implementación - Evaluación</b>						
39	Soporte Postproducción	0	37	15/04/24	15/04/24	Todos
40	Evaluación / plan de evaluación	0	37	12/04/24	12/04/24	Mónica González
41	Plan de implementación masiva	0	37	15/04/24	15/04/24	Mónica González

Cronograma de actividades inicial.

INFORMACIÓN PERSONAL
CAPTACIONES
COLOCACIONES

Información Personal

**FILTROS**

Número de Identificación	Código de Socio:
--------------------------	------------------

**INFORMACIÓN PERSONAL**

Tipo de Identificación:	Número de Identificación:	Código de Socio:
Apellidos y Nombres	Fecha de nacimiento	Edad
Estado Civil	Cargas Familiares	

**INFORMACIÓN GENERAL**

Sucursal	Estado de Solicitud	Fecha de Apertura Cuenta
Cuenta Activa	Personas Actualizada	

☰
⋮
🔍

⋮
🏠

*Prototipos Información personal.*

INFORMACIÓN PERSONAL
CAPTACIONES
COLOCACIONES

Frame 1

SERVICIOS VIRTUALES

JAVirtual Contratado    JAMovil Contratado    JAPagos Contratado    Tarjeta Contratado

INFORMACIÓN ADICIONAL

Grupo Organizado: <u>ELECAUSTRO</u>	Fondo Administrado: <u>SI</u>	Vinculado: <u>SI</u>
Providencias Judiciales: <u>SI</u>	Base de Ahorro: <u>1,456,90</u>	Listas negras: <u>SI</u>

La tabla sólo se puede visualizar cuando las Providencias Judiciales (SI)

CUENTAS DEL SOCIO

Producto	Nombre Cuenta	Número Cuenta	Saldo Total	Saldo Disponible	Saldo Bloqueado	
AHORROS A LA VISTA	AHORRO	2248078	800,00	400,00	400,00	<a href="#">ver</a>
AHORRO PROGRAMADO - MEDIANO PLAZO	AHORRO	5590785	1.000,00	1.000,00	0,00	<a href="#">ver</a>
AHORROS A LA VISTA	AHORRO	2248078	800,00	400,00	400,00	<a href="#">ver</a>
CERTIFICADOS DE APORTACIÓN	CERTIFICADOS	67890	100,00	95,00	5,00	<a href="#">ver</a>

PLAZOS FIJOS

Código Plazo	Capital	Plazo	Tasa interés	Valor Interés	Total	Fecha Captación	Fecha Vencimiento
908234	4.000,00	365	8,75%	354,88	4.354,88	20/02/2024	20/02/2025

Prototipo Información Captaciones.

INFORMACIÓN PERSONAL
CAPTACIONES
COLOCACIONES

Frame 2

INFORMACIÓN CREDITICIA

DEUDOR

Sucursal	Actúa como	Tipo Operación	Producto	N° Operación	Fecha Operación	Fecha vencimiento	Cantidad Solicitada	Saldo	Cuota	N° Cuotas Mora Total	Días Mora Acumulados	N° Cuotas Mora Actual	Días Mora Actual	Calificación	Estado	
Cuenca	Titular	Crédito	Extraordinario	56890	10/02/2012	10/01/2013	\$ 2.000,00	\$ 0,00	145,50	0	0	0	0	Original	Cancelado	<a href="#">ver</a>

Deudor Principal:	Cónyuge:	Tipo Garantía:
Subsegmento:	Código Póliza Seguro:	Observación:

Cuenca	Titular	Crédito	Con Ahorro Línea de Consumo Ordinaria	96993	22/05/2019	22/11/2021	\$ 10.000,00	\$ 0,00	220,00	0	0	0	0	Original	Cancelado	<a href="#">ver</a>
Cuenca	Cónyuge	Crédito	Con Ahorro Línea de Consumo Ordinaria	990117	16/09/2023	16/03/2025	\$ 5.000,00	\$ 2.000,00	308,00	3	20	1	5	Original	Liquidado	<a href="#">ver</a>

GARANTE

Sucursal	Actúa como	Tipo Operación	Producto	N° Operación	Fecha Operación	Fecha vencimiento	Cantidad Solicitada	Saldo	Cuota	N° Cuotas Mora Total	Días Mora Acumulados	N° Cuotas Mora Actual	Días Mora Actual	Calificación	Estado	
Cuenca			Extraordinario	56890			\$ 2.000,00		145,50	0			0	Original		<a href="#">ver</a>

Deudor Principal:	Cónyuge:	Tipo Garantía:
Subsegmento:	Código Póliza Seguro:	Observación:

CARTA DE GARANTÍA

Sucursal	Código Garantía	Fecha Garantía	Beneficiario	RUC beneficiario	Nombre contrato	Monto	Fecha Inicio	Fecha Vencimiento	Días vigencia	Tipo garantía	Estado
Loja	3924	21/02/2024	Agencia de Regulación y Control de las Telecomunicaciones ARCTEL	176811900001	Registro de Servicios de acceso a Internet y concesión de uso y explotación de frecuencias no esenciales del espectro radioeléctrico	\$425,00	28/02/2024	14/03/2025	300	Ful cumplimiento	Ejecutado

Prototipo Información Captaciones.

Frame 3

TABLA DE AMORTIZACION CALCULADA

Número Crédito	Nº Cuota	Fecha Final	Saldo	Capital	Días	Interés	Días Mora	Mora	Interés Diferido	Valor Pendiente	Seguro Desgravamen	Total	Estado
602006	1	02/07/2022	75000	441,95	15	281,95	0	0	\$ 0,00	0,00	0,00	723,20	PENDIENTE

Prototipo Tabla de amortización

The screenshot shows a ClickUp workspace for 'Historial Financiero'. The main view is a task list with columns for 'Pendiente', 'Terminado', and 'Delegado'. The 'Terminado' column is active, showing a list of completed tasks such as 'Definir tecnologías a usar', 'Buscar en dónde deployar CGP', and 'Levantar PostgreSQL Docker'. The left sidebar contains navigation options like 'Inicio', 'Bandeja de entrada', 'Documentos', and 'Espacios'. The top navigation bar includes a search bar and a 'Gestionar tarjetas' button.

Planificación de actividades Spring Planning ClickUp

```
.gitlab-ci.yml x
.gitlab-ci.yml
35 lockerize_frontend:
36   stage: dockerize
37   image: google/cloud-sdk:latest # Imagen Docker para construir y empujar la imagen Docker
38   variables:
39     DOCKER_HOST: tcp://docker:2375/
40     DOCKER_TLS_CERTDIR: ""
41   script:
42     - |
43       if [[ "$CI_COMMIT_REF_NAME" == "main" ]]; then
44         echo "Deploying to production environment"
45         echo "Project ID: $GCP_PROJECT_ID_PROD"
46         export GCP_PROJECT_ID="$GCP_PROJECT_ID_PROD"
47         export GCP_SERVICE_KEY="$GCP_SERVICE_KEY_PROD"
48         export GCR_IMAGE="gcr.io/$GCP_PROJECT_ID/historial-financiero-app"
49         echo "IMAGEN RESULTANTE"
50         echo "$GCR_IMAGE"
51       else
52         echo "Deploying to development environment"
53         echo "Project ID: $GCP_PROJECT_ID_DEV"
54         export GCP_PROJECT_ID="$GCP_PROJECT_ID_DEV"
55         export GCP_SERVICE_KEY="$GCP_SERVICE_KEY_DEV"
56         export GCR_IMAGE="gcr.io/$GCP_PROJECT_ID/historial-financiero-app"
57         echo "IMAGEN RESULTANTE"
58         echo "$GCR_IMAGE"
59       fi
60   # Autenticación en Google Cloud y configuración del registro Docker
61   - echo "$GCP_SERVICE_KEY" | base64 -d > $HOME/gcloud-service-key.json
62   - gcloud auth activate-service-account --key-file=$HOME/gcloud-service-key.json
63   - gcloud auth configure-docker
64   - docker build -t "$GCR_IMAGE" -f Dockerfile .
65   - docker push "$GCR_IMAGE"
66   - echo "$GCR_IMAGE" > gcr_image.txt # Guardar la imagen en un archivo para pasarla al siguiente stage
67   artifacts:
68     paths:
69       - gcr_image.txt # Pasar el archivo al siguiente stage
70   dependencies:
71     - build_frontend
72   only:
73     - development
74     - main
```

*Runner historial financiero app.*