



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Implementación de un sistema backend de una
plataforma digital para solicitud de microcrédito
dirigido a conductores de taxi aplicativo en Lima
Metropolitana**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Marco Antonio NOLASCO RIVAS

ASESOR

Arturo Alejandro BARTRA MORE

Lima, Perú

2025



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Nolasco, M. (2025). *Implementación de un sistema backend de una plataforma digital para solicitud de microcrédito dirigido a conductores de taxi aplicativo en Lima Metropolitana*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Marco Antonio Nolasco Rivas
Tipo de documento de identidad	DNI
Número de documento de identidad	70432334
URL de ORCID	https://orcid.org/0009-0001-0013-8582
Datos de asesor	
Nombres y apellidos	Arturo Alejandro Bartra More
Tipo de documento de identidad	DNI
Número de documento de identidad	40233946
URL de ORCID	https://orcid.org/0000-0002-3411-7456
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Joel Fernando Machado Vicente
Tipo de documento	DNI
Número de documento de identidad	40476778
Miembro del jurado 1	
Nombres y apellidos	Ciro Rodríguez Rodríguez
Tipo de documento	DNI
Número de documento de identidad	06020241
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento

Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima
	Distrito: Cercado de Lima Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2024
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 19:00 horas del día 14 de enero del año 2025, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Machado Vicente Joel Fernando (Presidente), Dr. Rodríguez Rodríguez Ciro (Miembro) y el Ing. Bartra More Arturo Alejandro (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/jtg-ystx-bsz>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“IMPLEMENTACIÓN DE UN SISTEMA BACKEND DE UNA PLATAFORMA DIGITAL PARA SOLICITUD DE MICROCRÉDITO DIRIGIDO A CONDUCTORES DE TAXI APLICATIVO EN LIMA METROPOLITANA”**, por el Bachiller Nolasco Rivas Marco Antonio; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **18 (DIECIOCHO)**.

A continuación el Presidente de Jurados el Mg. Machado Vicente Joel Fernando, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 19:50 horas, se levantó la sesión.

Presidente

Mg. Machado Vicente Joel Fernando

Miembro

Dr. Rodríguez Rodríguez Ciro

Miembro Asesor

Ing. Bartra More Arturo Alejandro



CERTIFICADO DE SIMILITUD

Yo, BARTRA MORE ARTURO ALEJANDRO en mi condición de asesor acreditado con la Resolución Decanal N° 001059-2024-D-FISI/UNMSM y Oficio N° 005-PTTSP-FISI-2024 del Trabajo de Suficiencia Profesional cuyo título es Implementación de un sistema backend de una plataforma digital para solicitud de microcrédito dirigido a conductores de taxi aplicativo en Lima Metropolitana, presentado por el bachiller NOLASCO RIVAS MARCO ANTONIO para optar el título de Ingeniero de Sistemas CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 8% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título.

Lima, 8 de enero de 2025

Firma del Asesor

Nombres y Apellidos del asesor: BARTRA MORE ARTURO ALEJANDRO

DNI 40233946

DEDICATORIA

Dedico este trabajo con profundo agradecimiento a mis padres Ana y Germán, quienes siempre han creído en mí y me han apoyado en cada paso de mi camino académico. A mi abuelita María, cuya guía desde el cielo ilumina siempre mi sendero. A mi esposa Elizabeth y mis hijos Lucas y Adrián, por su amor y paciencia, que me inspiran a seguir adelante en cada desafío. Y a mi hermano Carlos, por ser una constante fuente de inspiración y motivación para mejorar siempre.

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a mi alma mater, la Universidad Nacional Mayor de San Marcos, por brindarme la oportunidad de formarme en sus aulas. Gracias a esta institución, he podido adquirir no solo conocimientos, sino también valiosas experiencias que me acompañarán siempre.

A mi asesor, Arturo Alejandro Bartra More, le agradezco enormemente por su orientación y sabios consejos a lo largo de todo el proceso de desarrollo de este proyecto. Su apoyo ha sido fundamental para llevar a cabo este trabajo con éxito.

También me gustaría agradecer a la comisión de titulación por darnos la oportunidad de poner en práctica lo aprendido a lo largo de nuestra carrera, y por hacer posible que podamos cerrar este capítulo de nuestra formación profesional.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**Implementación de un sistema backend de una plataforma digital para solicitud
de microcrédito dirigido a conductores de taxi aplicativo
en Lima Metropolitana**

Autor: Nolasco Rivas, Marco Antonio

Asesor: Bartra More, Arturo Alejandro

Título: Trabajo de Suficiencia Profesional para optar por el Título Profesional
de Ingeniero de Sistemas

Fecha: Diciembre de 2024

RESUMEN

Como parte de su compromiso con la inclusión financiera, el banco ha lanzado proyectos piloto en varios sectores, incluyendo el transporte. Uno de estos proyectos está dirigido a los socios conductores de una conocida aplicación de taxis, quienes pueden solicitar microcréditos y apertura de cuentas de ahorro de manera completamente digital y segura, a través de la plataforma creada para ello. Se disponibilizó para los conductores de Lima Metropolitana y comenzó en 2019.

Estos microcréditos, además de fomentar la bancarización, pueden ser utilizados para convertir sus vehículos a gas, lo que representa una opción más económica y ecológica. También pueden servir para cubrir gastos inesperados que surjan durante su trabajo diario.

Para ello se ha desarrollado una plataforma digital que consta de dos partes: frontend y backend. El enfoque principal de este trabajo está en el segundo, que aprovecha los beneficios de las tecnologías como Azure, Java con Spring Boot y Azure SQL Database. Se encarga de aplicar las reglas de negocio y se conecta con diversas APIs, tanto internas como externas, para validar y evaluar las solicitudes. Así, se asegura que las respuestas de aprobación se generen en un plazo máximo de 48 horas.

Palabras clave: Sistema, backend, microcréditos, inclusión, financiera, conductores.

**NATIONAL MAJOR SAN MARCOS UNIVERSITY
FACULTY OF SYSTEMS ENGINEERING AND INFORMATICS
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING**

**Implementation of a backend system of a digital platform for microcredit
application for cab drivers in Metropolitan Lima.**

Author: Nolasco Rivas, Marco Antonio

Advisor: Bartra More, Arturo Alejandro

Title: Professional Sufficiency Work for the Professional Title of
Systems Engineer

Date: December 2024

ABSTRACT

As part of its commitment to financial inclusion, the bank has launched pilot projects in several sectors, including transportation. One of these projects is aimed at driver-partners of a well-known cab app, who can apply for microcredits and open savings accounts in a completely digital and secure way, through the platform created for this purpose. It was made available to drivers in Metropolitan Lima and started in 2019.

These microcredits, in addition to promoting bankarization, can be used to convert their vehicles to gas, which represents a more economical and environmentally friendly option. They can also be used to cover unexpected expenses that arise during their daily work.

For this purpose, a digital platform has been developed consisting of two parts: frontend and backend. The main focus of this work is on the second one, which leverages the benefits of technologies such as Azure, Java with Spring Boot and Azure SQL Database. It is responsible for applying business rules and connects with various APIs, both internal and external, to validate and evaluate requests. This ensures that approval responses are generated within a maximum of 48 hours.

Keywords: System, backend, microloans, inclusion, financial, drivers.

TABLA DE CONTENIDO

DEDICATORIA	iii
AGRADECIMIENTO	iv
RESUMEN	v
ABSTRACT.....	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
INTRODUCCIÓN	1
CAPITULO I. TRAYECTORIA PROFESIONAL	3
CAPITULO II. CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA	12
2.1. Actividad de la empresa	12
2.2. Visión	12
2.3. Misión	12
2.4. Organización de la empresa	12
2.5. Área, cargo y funciones desempeñadas.....	13
2.6 Experiencia profesional realizada en la organización	13
CAPITULO III. ACTIVIDADES DESARROLLADAS	15
3.1. Situación problemática.....	15
3.1.1. Definición del problema.....	15
3.2. Solución.....	16
3.2.1. Objetivos	16
3.2.2. Alcance.....	17
3.2.3. Etapas y metodología	17
3.2.4. Fundamentos utilizados.....	22
3.2.5. Implementación de las áreas, procesos, sistemas y buenas prácticas.....	29
3.3. Evaluación.....	60

3.3.1. Evaluación económica / Evaluación costo-beneficio.....	60
3.3.2. Interpretación del VAN y del TIR	67
CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA	73
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....	75
5.1. Conclusiones	75
5.2. Recomendaciones.....	77
5.3. Fuentes de información	78
5.4. Glosario	81
ANEXOS	82
Anexo 1: Certificaciones.....	82
Anexo 2: Cálculo de Beneficios Esperados	83
Anexo 3: Cálculo de VPC y VPB en Excel	84
Anexo 3: Cálculo del índice Beneficio/Costo	84
Anexo 4: Cálculo del VAN	85
Anexo 5: Cálculo de la TIR	85

ÍNDICE DE FIGURAS

Figura 1 División de Transformación donde se desarrolló el proyecto.....	12
Figura 2 Marco de trabajo Scrum	21
Figura 3 Reglas de negocio definidas	30
Figura 4 Ejemplo de un product backlog.....	31
Figura 5 Ejemplo donde se incluye criterios de aceptación a una historia.	31
Figura 6 Sesión para estimar una historia en el planning.	32
Figura 7 Estimación de una historia de usuario.....	33
Figura 8 Resultados de la estimación	33
Figura 9 Puntajes de la estimación luego de llegar a un consenso.	34
Figura 10 Tablero Jira, con el sprint activo y su objetivo asociado.	35
Figura 11 Historias puntuadas y asignadas.....	35
Figura 12 Diagrama de solución y las diversas interacciones	36
Figura 13 Diagrama de secuencia solicitud de crédito.	37
Figura 14 Modelo entidad relacional para el proyecto backend.....	38
Figura 15 Definición del contrato para el backend.....	39
Figura 16 Creación de un nuevo proyecto.....	40
Figura 17 Estructura inicial del proyecto.....	41
Figura 18 Estructura de carpetas del proyecto.....	41
Figura 19 Carpeta controller con sus clases.	42
Figura 20 Carpeta dto con sus clases.....	42
Figura 21 Carpeta model con sus clases.	43
Figura 22 Carpeta repository con sus clases.....	43
Figura 23 Carpeta service con sus clases.....	44
Figura 24 Carpeta util con su clase asociada.	44
Figura 25 Modelo de Branch Gitflow.....	45
Figura 26 Ejemplos de nombre de las ramas en Bitbucket.....	46
Figura 27 Definición de ramas principales	47
Figura 28 Especificación de ramas principales en Bitbucket.	47
Figura 29 Inicialización repositorio Git local.....	48
Figura 30 Agregar archivos al espacio de trabajo.	48

Figura 31	Cambios comiteados en local.....	49
Figura 32	Agregar repositorio remoto y cambio de nombre de rama.....	49
Figura 33	Subir cambios a la nueva rama.....	49
Figura 34	Bitbucket solicita acceso.	50
Figura 35	Nueva rama feature en Bitbucket.	50
Figura 36	Pull Request desde una rama feature hacia develop.....	50
Figura 37	Revisión de los cambios en un pull request.	51
Figura 38	Ejemplo de unit test.....	52
Figura 39	SonarCloud solicita acceso a Bitbucket	53
Figura 40	Pipeline para integrar con SonarCloud.....	53
Figura 41	App Service que se utiliza para la aplicación backend.	54
Figura 42	Pipeline para el despliegue de la aplicación en un App Service.	55
Figura 43	Reunión técnica para crear el documento de matriz de pruebas.	56
Figura 44	Correo de conformidad de pase a producción.....	57
Figura 45	Presentación de lo trabajado en el sprint en la review.	58
Figura 46	Agenda de la Review.....	58
Figura 47	Logs visualizados en la herramienta Azure Application Insights.	59
Figura 48	Certificación del autor en Azure Fundamentals.....	82
Figura 49	Cálculo de beneficios esperados.....	83
Figura 50	Cálculo de VPC y VPB.	84
Figura 51	Índice de beneficio/costo.....	84
Figura 52	Cálculo del VAN.	85
Figura 53	Cálculo de la TIR	85

ÍNDICE DE TABLAS

Tabla 1 Experiencia profesional en NTTDATA.....	3
Tabla 2 Experiencia profesional en NTTDATA.....	4
Tabla 3 Experiencia profesional en NTTDATA.....	5
Tabla 4 Experiencia profesional en ENCORA	6
Tabla 5 Experiencia profesional en ENCORA	7
Tabla 6 Experiencia profesional en ENCORA	8
Tabla 7 Experiencia profesional en ENCORA	9
Tabla 8 Experiencia profesional en ENCORA	9
Tabla 9 Experiencia profesional en EVERIS.....	10
Tabla 10 Experiencia profesional en DJL.....	11
Tabla 11 Recursos requeridos para la implementación del sistema.....	60
Tabla 12 Infraestructura y servicios para la implementación del sistema.	61
Tabla 13 Herramientas para gestión de trabajo.....	61
Tabla 14 Logros obtenidos por la plataforma desde mayo 2019.	62
Tabla 15 Cálculo de monto aproximado a pagar por un microcrédito.	63
Tabla 16 Beneficios esperados del proyecto en los 3 años.....	64
Tabla 17 Costos esperados del proyecto en los 3 años.	64
Tabla 18 Beneficios y costos esperados en los 3 periodos.	65
Tabla 19 Enfoque de escenario para la inversión inicial del proyecto	67
Tabla 20 Cálculo de monto aproximado a pagar por un microcrédito	67
Tabla 21 Flujo de caja anual neto para el escenario pesimista.	70
Tabla 22 Flujo de caja anual neto para el escenario conservador.....	71
Tabla 23 Flujo de caja anual neto para el escenario optimista.	71
Tabla 24 Cuadro comparativo evaluar la inversión	72

INTRODUCCIÓN

La inclusión financiera se ha convertido en un tema clave para promover el desarrollo económico y social, ya que permite que personas tradicionalmente excluidas del sistema financiero accedan a servicios como cuentas de ahorro y préstamos. La posibilidad de acceder a estos servicios es crucial para mejorar la calidad de vida de sectores vulnerables y potenciar el crecimiento de microempresas, entre otros.

En este contexto, las alianzas estratégicas con socios comerciales juegan un papel fundamental. Estas alianzas permiten implementar pilotos y validar la escalabilidad de nuevos modelos de negocio. En el presente trabajo se ha seleccionado el sector financiero con un enfoque particular en el nicho de transporte; aprovechando la colaboración con una empresa de taxi de aplicativo, como socio estratégico. Esta empresa ofrece un programa de beneficios exclusivos a sus socios conductores, entre ellos la oportunidad de acceder a microcréditos ofrecidos por la entidad financiera.

Estos microcréditos, de montos accesibles y específicos, están diseñados para cubrir necesidades inmediatas relacionadas con el vehículo, herramienta principal de trabajo de los conductores.

Para brindar esta oportunidad, la entidad financiera, ha desarrollado una plataforma digital que facilita a los socios conductores canalizar su solicitud de microcrédito. Esta plataforma se integra con diversos sistemas internos y externos que permiten el análisis de la información, evaluación detallada, cálculo del puntaje crediticio(scoring) y finalmente brindar una respuesta en un plazo no mayor de 48 horas.

A continuación, se detalla la estructura del trabajo:

- En el capítulo I, se desarrolla la trayectoria profesional donde se demuestra la experiencia lograda.
- En el capítulo II, describe el contexto en el que desarrolló la experiencia, además de todo lo relacionado a la empresa, así como la experiencia profesional realizada en la organización.
- En el capítulo III, se detalla las actividades desarrolladas en el presente trabajo, como son la situación problemática, los objetivos, metodología del ciclo de vida y los resultados de la situación propuesta.

- En el capítulo IV se expone la reflexión crítica de la experiencia y el aporte del autor, las necesidades que se atendieron, la experiencia y la capacitación requerida.
- Y finalmente en el capítulo V se detallan las conclusiones y recomendaciones del presente trabajo de suficiencia profesional.

CAPITULO I. TRAYECTORIA PROFESIONAL

El autor del presente trabajo de suficiencia profesional es Bachiller en Ingeniería de Sistemas en la Universidad Nacional Mayor de San Marcos con sólida experiencia en el sector bancario, donde ha participado en diversos proyectos de transformación digital, como parte del equipo de desarrollo, en la creación y optimización de aplicaciones utilizando tecnologías como Java y Azure, aportando soluciones innovadoras para mejorar la eficiencia y calidad del servicio. Además, lideró equipos de desarrollo, aplicando metodologías ágiles y promoviendo las mejores prácticas para garantizar la entrega oportuna de productos de alto impacto. Con un enfoque en la mejora continua y el trabajo colaborativo, los cuales le permitieron potenciar la efectividad de los proyectos.

1.1 EXPERIENCIA PROFESIONAL

NTT DATA

Empresa multinacional que brinda servicios de tecnología de la información y consultoría a las diferentes industrias del país.

Tabla 1

Experiencia profesional en NTTDATA

NTT DATA	LIMA, PERÚ (agosto 2024 – actualidad)
Cargo	Technical Lead
Sector del cliente	Financiero
Proyecto	Migración del servicio Core a APIs; para los procesos de calificación de planillas y desembolso de productos de descuento, cobranza libre y garantía.
Funciones	• Mapeo de los flujos de negocio. • Análisis y resolución de problemas. • Desarrollo de contratos de APIs. • Implementación de APIs. • Pruebas unitarias. • Gestión con equipos.

Nota: Fuente elaboración propia

Tabla 2*Experiencia profesional en NTTDATA*

NTT DATA	LIMA, PERÚ (agosto 2023 – julio 2024)
Cargo	Technical Lead
Sector del Cliente	Financiero
Proyecto	Migración de los servicios Core a APIs; para el cálculo de tasas de los productos de largo y corto plazo para la banca de negocio.
Funciones	• Mapeo del flujo de negocio. • Análisis y resolución de problemas. • Diseño de contratos de APIs. • Implementación de APIs de experiencia. • Manejo de personas a cargo. • Pruebas unitarias. • Gestión con equipos Core.

Nota: Fuente elaboración propia.

Tabla 3*Experiencia profesional en NTTDATA*

NTT DATA	LIMA, PERÚ (junio 2022 – julio 2023)
Cargo	Technical Lead
Sector del cliente	Financiero
Proyecto	Migración de los servicios Core a APIs; para el cálculo de tasa de los productos hipotecario y mayorista.
Funciones	• Mapeo del flujo de negocio. • Análisis y resolución de problemas. • Diseño de contratos de APIs. • Implementación de APIs. • Pruebas unitarias. • Gestión con equipos Core. • Implementación de Datafactory.

Nota: Fuente elaboración propia

ENCORA

Consultora multinacional dedicada a brindar servicios de ingeniería de software con un modelo de negocio enfocado en el cliente.

Tabla 4

Experiencia profesional en ENCORA

ENCORA (AVANTICA)	LIMA, PERÚ (agosto 2019 – junio 2022)
Cargo	Software Engineer II
Sector del cliente	Financiero
Proyecto	Plataforma digital que permite solicitar microcréditos a conductores de taxi aplicativo.
Funciones	• Mapeo del flujo de negocio. • Análisis y resolución de problemas. • Diseño de contratos de APIs. • Implementación de APIs REST. • Integración con APIs externas. • Pruebas unitarias. • Gestión con equipos de Operaciones y Frontend y Core. • Despliegues. • Resolución de Habilitadores de seguridad.

Nota: Fuente elaboración propia.

Tabla 5*Experiencia profesional en ENCORA*

ENCORA (AVANTICA) LIMA, PERÚ (diciembre 2021 – junio 2022)	
Cargo	Software Engineer II
Sector del cliente	Financiero
Proyecto	Plataforma que permite a los clientes de Alicorp del rubro de panificación y gastronomía, solicitar un préstamo.
Funciones	• Creación y/o modificación de stored procedures. • Coordinaciones diversas con los equipos involucrados interno y externos. • Revisión y correcciones de incidencias en landing del proyecto. • Resolución de habilitadores de seguridad. • Modelado de amenazas con el framework Trike para los flujos de la landing.

Nota: Fuente elaboración propia.

Tabla 6*Experiencia profesional en ENCORA*

ENCORA (AVANTICA)	LIMA, PERÚ (diciembre 2021 – junio 2022)
Cargo	Software Engineer II
Sector del cliente	Financiero
Proyecto	Plataforma que permite solicitar un préstamo a los afiliados a una empresa comercializadora de smartphones para que puedan adquirir sus productos.
Funciones	• Resolución de habilitadores de seguridad. • Modelado de amenazas utilizando el framework Trike para el flujo de la landing. • Revisión de incidencias en landing del proyecto.

Nota: Fuente elaboración propia

Tabla 7*Experiencia profesional en ENCORA*

ENCORA (AVANTICA)	LIMA, PERÚ (diciembre 2021 – junio 2022)
Cargo	Software Engineer II
Sector del cliente	Financiero
Proyecto	Plataforma digital para clientes y no clientes de la entidad financiera; con servicios de educación financieras y uso de herramientas digitales para emprendedores.
Funciones	• Implementación de APIs • Implementación de iniciativa para la transferencia de archivos. • Creación y/o modificación de stored procedures. • Coordinaciones diversas con los equipos core y operaciones.

Nota: Fuente elaboración propia.

Tabla 8*Experiencia profesional en ENCORA*

ENCORA (AVANTICA)	LIMA, PERÚ (diciembre 2021 – junio 2022)
Cargo	Software Engineer II
Sector del cliente	Financiero
Proyecto	Plataforma digital que recibe solicitudes de préstamo de aliados físicos.
Funciones	• Análisis de APIs para la implementación. • Coordinaciones diversas con los equipos frontend, core y operaciones.

Nota: Fuente elaboración propia.

Tabla 9

Experiencia profesional en EVERIS

EVERIS(NTTDATA)	LIMA, PERÚ (octubre 2017 – agosto 2019)
Cargo	Solutions Assistant
Sector del cliente	Financiero
Proyecto	Aplicativo móvil para el asesor de negocio de la entidad financiera, para brindar información oportuna a los clientes recurrentes y potenciales.
Funciones	• Implementación de las diversas funcionalidades con arquitectura orientada a microservicios. • Integración con APIs internas. • Implementación de “Pruebas de vida”. • Coordinaciones diversas con los equipos frontend y core. • Despliegue del APIs • Pruebas unitarias.

Nota: Fuente elaboración propia.

Tabla 10

Experiencia profesional en DJL

DJL BUSINESS AND SYSTEMS LIMA, PERÚ (junio 2015 – octubre 2017)	
Cargo	Consultor TI
Sector del cliente	Salud
Proyecto	Desarrollo de nuevas funcionalidades y mantenimiento del sistema Core del cliente.
Funciones	• Revisión y análisis de los requerimientos. • Generación de los documentos funcionales y técnicos. • Implementación de mejoras del sistema core. • Coordinaciones con el equipo de despliegue para el comité y posterior pase a producción.

Nota: Fuente elaboración propia.

CAPITULO II. CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1. Actividad de la empresa

La entidad financiera, se enfoca en las microfinanzas y principalmente se dedica a ofrecer créditos y servicios financieros a las micro y pequeñas empresas (MYPES).

2.2. Visión

Ser una entidad financiera referente en microfinanzas, para los micro y pequeños empresarios; donde destaque el compromiso de brindar soluciones financieras eficaces que impulsen el desarrollo y el progreso de sus clientes (Entidad Financiera, 2024).

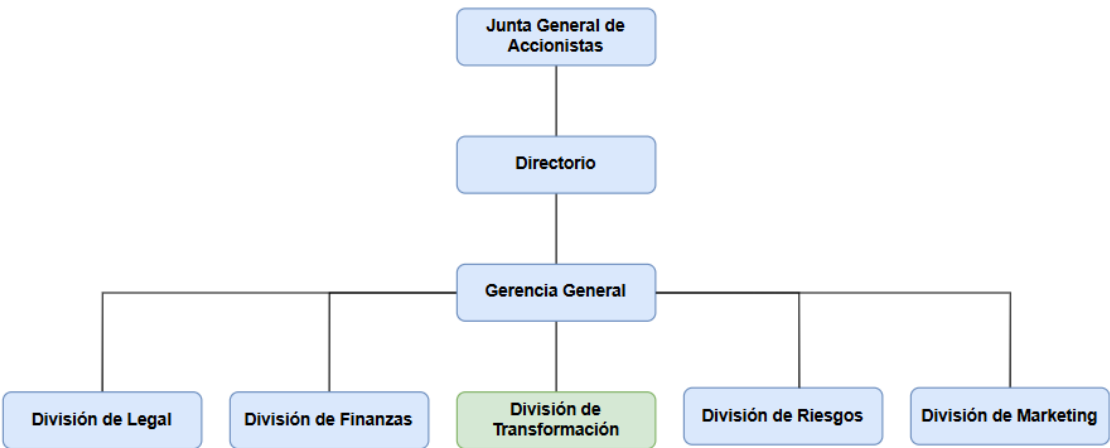
2.3. Misión

Contribuir al desarrollo económico y social del país brindando servicios de inclusión financiera que permitan a las personas mejorar su calidad de vida (Entidad Financiera, 2024).

2.4. Organización de la empresa

Figura 1

División de Transformación donde se desarrolló el proyecto.



Nota: Fuente elaboración propia

2.5. Área, cargo y funciones desempeñadas

Área: División de Transformación

Cargo: Software Engineer II

Funciones desempeñadas:

- Implementación de APIs REST para las diferentes funcionalidades de la landing entre ellas el registro, actualización de datos de los conductores y la sección de contáctanos.
- Integración con APIs externas e internas; para obtener información de los socios conductores.
- Implementación de pruebas unitarias y cobertura de código.
- Implementación de pipeline en Bitbucket para CI/CD.
- Coordinaciones diversas con los equipos Frontend, Core y Operaciones.
- Despliegues de releases al ambiente de producción.
- Documentación de las funcionalidades principales del proyecto.
- Resolución de habilitadores de seguridad.
- Modelado de amenazas utilizando el framework Trike para los flujos de la landing.

2.6 Experiencia profesional realizada en la organización

El autor durante su tiempo en la entidad financiera tuvo la oportunidad de formar parte de varios proyectos los cuales le han permitido potenciar de manera considerable su formación profesional. A continuación, se describe las principales actividades y responsabilidades en los proyectos en el rol de Software Engineer II:

- Participó en el diseño, desarrollo, pruebas, despliegue y mantenimiento de las APIs REST las cuales exponían funcionalidades al equipo de frontend para sus correspondientes integraciones como la solicitud de crédito y de cuenta de ahorro entre otros. Utilizando Java con Spring Boot y varios servicios de Azure, aprovechando los grandes beneficios como el desarrollo ágil, infraestructura y la escalabilidad.
- Coordinó y gestionó requerimientos con equipos internos y externos a la organización algunos de estos proyectos incluyeron: Compra de certificados

digitales, reuniones para definición de soluciones, coordinación para configuración, pruebas del WAF, Ethical hacking e implementaciones con el equipo Core y Frontend.

- Implementó pruebas unitarias para garantizar el correcto funcionamiento de la aplicación, asegurando así la calidad y estabilidad del código, utilizando herramientas como JUnit 5 y Mockito.
- Durante el desarrollo del proyecto adoptó el enfoque Shift Left en las pruebas, lo que permitió realizar las pruebas de calidad en etapas más tempranas del ciclo de vida del software. Esto contribuyó a mejorar tanto la calidad del código como los tiempos de entrega.
- Asimismo, implementó pipelines para automatizar despliegues en los diferentes ambientes, agilizando los procesos de integración y entrega continua.
- Aprovechó herramientas de monitorización de Azure, como Application Insights, para detectar y corregir rápidamente defectos, lo que permitió una resolución eficiente de problemas en tiempo real.
- Adoptó un enfoque ágil que facilitó el trabajo en equipo, promoviendo los beneficios de la colaboración y la adaptabilidad en un entorno dinámico.
- Fomentó sesiones de conocimiento dentro del equipo, buscando formar un grupo multidisciplinario que pudiera colaborar eficazmente en diversas áreas del proyecto.
- Finalmente, a lo largo del proyecto, realizó varias pruebas de concepto (POC) para validar posibles soluciones a necesidades específicas del proyecto antes de su implementación completa; donde se evaluaba la viabilidad técnica de diversas ideas, tecnologías de esta manera poder garantizar que las soluciones propuestas cumplieran con los requisitos del proyecto.

CAPITULO III. ACTIVIDADES DESARROLLADAS

3.1. Situación problemática

En 2019, en Lima Metropolitana, el uso de la aplicación de taxi en estudio había crecido de manera significativa, desde su llegada a Perú el 2014, lo que permitió convertir a muchos conductores en trabajadores independientes, conocidos como socios conductores, que generan sus ingresos utilizando esta plataforma. A la fecha 2024 cuenta con más 320 mil socios conductores (Hernández, 2024).

Sin embargo, este modelo laboral enfrentaba desafíos significativos en términos de inclusión financiera. Muchos de estos socios conductores carecían de acceso a servicios financieros adecuados que les permitían gestionar sus ganancias, ahorrar (Instituto Peruano de Economía, 2023, p. 26) y acceder a préstamos para mejorar sus condiciones de trabajo. Por el simple hecho de no estar bancarizado, no tener historial crediticio o una fuente de ingresos verificable.

Además, abrir una cuenta de ahorros y solicitar un préstamo en cualquier banco implicaba un proceso tedioso, con largas filas y un montón de papeleo. Es por ello los conductores no solo perdían tiempo valioso en gestionar estos trámites, sino que también le restaban horas de trabajo a su día, lo cual limitaba su capacidad de aumentar sus ingresos. Esta situación generaba frustración ante la falta de opciones accesibles que se adaptasen a su estilo de vida.

En ese sentido, era necesaria una solución rápida y accesible que beneficiara a estos conductores.

3.1.1. Definición del problema

El problema radica en la falta de una plataforma digital que permita al denominado socio conductor del taxi aplicativo solicitar y acceder a un microcrédito de manera digital y rápida.

3.2. Solución

Para abordar este problema, se propone el desarrollo de una plataforma digital diseñada para facilitar la solicitud y acceso a microcréditos por parte de los socios conductores. Esta herramienta no solo permitirá a los usuarios acceder a financiamiento de manera ágil y sencilla, sino que también incluirá la opción de crear cuentas de ahorro. La idea es ofrecer una solución integral que simplifique el proceso financiero, eliminando las barreras burocráticas que suelen enfrentar. Al utilizar esta plataforma, los conductores podrán gestionar sus finanzas de manera más efectiva, acceder a créditos que se ajusten a sus necesidades y fomentar el ahorro, todo desde sus dispositivos móvil. De esta manera, se busca empoderar a los conductores, mejorando su estabilidad económica y ayudándoles a enfrentar los retos de su actividad laboral.

3.2.1. Objetivos

3.2.1.1 Objetivo General

Implementación de un sistema backend de una plataforma digital para solicitud de microcrédito dirigido a conductores de taxi aplicativo en Lima Metropolitana.

3.2.1.2. Objetivos Específicos

- 1- Adoptar la metodología Scrum dentro del equipo de desarrollo para facilitar la entrega constante de valor a negocio, fomentar una colaboración efectiva entre los miembros y mejorar continuamente la forma de trabajar.
- 2- Diseñar e implementar una API segura y escalable que incluya la lógica de negocio, reglas e integraciones con sistemas internos y externos a la entidad financiera para el análisis y evaluación de las solicitudes de microcréditos y apertura de cuentas de ahorros de los socios conductores.
- 3- Desarrollar la infraestructura de base de datos para almacenar y gestionar la información de las solicitudes de microcréditos.

- 4- Reducir los procesos manuales a través de la automatización del despliegue de componentes en los diferentes entornos, utilizando herramientas especializadas.
- 5- Asegurar que el desarrollo del sistema backend sea de calidad mediante la realización de pruebas detalladas que verifiquen que todas sus funciones trabajen correctamente y que los errores sean detectados desde las primeras etapas.

3.2.2. Alcance

El alcance de este trabajo se centra en la creación de la infraestructura del backend necesaria para soportar la lógica que gestionará las solicitudes de microcréditos, las evaluaciones de los perfiles de los socios conductores y la solicitud de apertura de cuentas de ahorro. Aunque existen socios conductores en diversas ciudades de Perú, el alcance de este proyecto está limitado a Lima Metropolitana.

- En el flujo de las solicitudes se estará incluyendo en el código las reglas de negocio que permitan realizar las validaciones de los socios conductores.
- Para ello se integrará con APIs internas como base de riesgo y validación de cuenta de ahorros para complementar las validaciones dentro del flujo de solicitudes tanto de crédito como de cuenta de ahorros.
- Se integrará con APIs externas del taxi aplicativo para obtener información del conductor (nombre, teléfono, correo y categorización) y de las ganancias.
- De la misma forma, para la evaluación automatizada nos apalancaremos de un API externo mediante una integración el cual nos brinda la funcionalidad de scoring.

3.2.3. Etapas y metodología

Para la implementación del sistema backend en el proyecto se utilizó Scrum el cual es muy valorado en el desarrollo de software por su enfoque iterativo, flexible y muy orientado a la colaboración.

3.2.3.1 Scrum

“Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos” (Schwaber & Sutherland, 2020, p. 3).

3.2.3.2. Pilares de Scrum

- **Transparencia:** Este pilar del Scrum facilita la inspección, donde el proceso y el trabajo deben ser visibles para todo el equipo.
- **Inspección:** Este pilar facilita la adaptación, pero es necesario para poder detectar alguna desviación de los objetivos.
- **Adaptación:** y este último pilar si se identifica alguna desviación se debe realizar un cambio para poder ajustarlo.

3.2.3.3. Valores de Scrum

Sirven para guiar el comportamiento y la cultura dentro un equipo que utiliza Scrum.

- **Coraje:** Incentiva a que cada miembro del equipo Scrum, pueda afrontar nuevos retos, asumir riesgos y pueda expresar sus puntos de vista sin enfrentamientos.
- **Enfoque:** Motiva a que los miembros del equipo Scrum, se centren en lo importante, en el objetivo del Sprint y eviten distracciones.
- **Compromiso:** Inspira a que los miembros del equipo Scrum, se unan y dediquen a cumplir los objetivos del Sprint, velando siempre por la calidad y la generación de valor.
- **Respeto:** Se crea un espacio colaborativo en el que se valora el aporte de cada persona.
- **Apertura:** Fomenta un entorno donde la transparencia es clave, donde se puede compartir ideas, sugerencias y comentarios sin miedo para generar confianza en el equipo.

3.2.3.4. Equipo Scrum

Está conformado por un Product Owner (PO), un Scrum Máster y los Desarrolladores.

- **Product Owner (PO):** Este rol prioriza el trabajo del producto backlog, según la necesidad y es el encargado de velar por la entrega de valor hacia los stakeholders.

- **Scrum Máster:** Este Rol es el que guía la utilización del Scrum en el equipo, brinda apoyo en el entendimiento de la teoría y práctica del framework asimismo incentiva la autogestión del equipo.
- **Desarrolladores:** Este rol es el encargado de trabajar las diferentes necesidades priorizadas en el producto backlog y generar valor al negocio en cada Sprint.

3.2.3.5. Eventos de Scrum

- **Sprint planning:** Este evento es donde se define lo que se trabajará en el sprint, el PO llega con las priorizaciones del backlog que se desea trabajar y se valida con el equipo los elementos que se pueden tomar del product backlog, así como también se puede definir si hay algún bloqueante por dependencia o por falta de capacity. Una vez llegado a un consenso PO y Team Member, el primero define el objetivo del sprint.
- **Daily Scrum:** Este evento se realiza, como indica el nombre, diariamente por lo general se da al inicio del día con una duración que varía según la cantidad de team members, entre 15 a 30 minutos; donde se inspecciona el avance del objetivo del sprint y si hay algún bloqueante para alertar y el sprint se pueda adaptar a cualquier cambio.

Por lo general la temática de una daily efectiva sería similar a estos ejemplos:

- “Yo estoy avanzando la historia de usuario validación de documentos, identifiqué un bloqueante que va a afectarnos podríamos revisarlo después de esta sesión”.
 - “Ayer terminé las pantallas de solicitud de créditos, hoy estaré enfocado en la pantalla de ahorros, no tengo bloqueantes”
- **Sprint Review:** Este evento se realiza al finalizar el sprint donde los team members presentan el resultado de su trabajo; por lo general se hace con una presentación y una demo del producto. El negocio; PO y stakeholders brindan feedback sobre lo entregado con ello se pueda hacer alguna adaptación y pueda satisfacer las necesidades de los interesados. En algunos casos cuando el sprint ha tenido muchos

bloqueantes y por ello no se ha podido cumplir el objetivo, también se indica que en este evento.

- **Sprint Retrospective:** Este es el último evento del sprint después de haber presentado el producto, se inspecciona porque nos fue bien o porque nos fue mal ahí se analiza diferentes puntos; personas, interacciones, procesos, herramientas y se busca posible origen del porque nos fue mal. Una vez identificado se define una actividad de mejora y los team members se comprometen a trabajar en el siguiente sprint. En la siguiente retrospectiva se evaluará la evolución.

Hay una práctica adicional en Scrum no es evento un formal, pero si importante el **refinamiento del backlog**; es una actividad que se puede realizar de manera continua o esporádica dependiendo de la necesidad del equipo, el propósito de este refinamiento es tener un entendimiento pleno de las historias que se van a trabajar, identificado dependencias, complejidad para que cuando se llegue al planning todo esté claro y sea rápido ese evento.

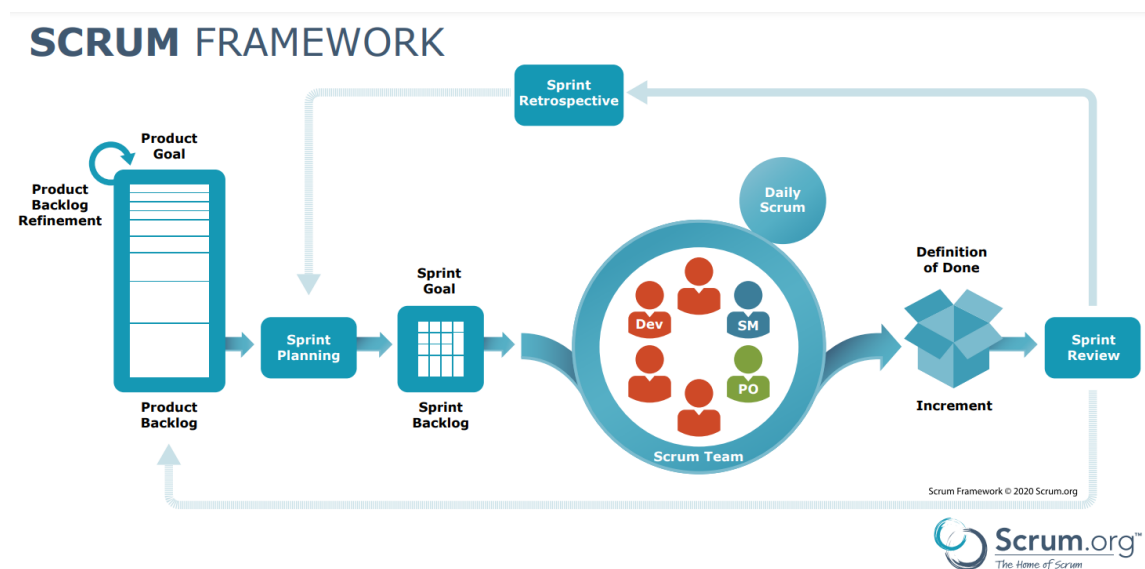
3.2.3.6. Artefactos de Scrum

- **Product Backlog:** Es una lista ordenada de todas las tareas que se necesita para generar y mejorar el producto.
- **Product Goal:** El objetivo del producto es una visión futura del producto, se toma en consideración para que el equipo Scrum pueda planificar sus tareas.
- **Sprint Backlog:** Es la lista de elementos de trabajo pendiente que se va a realizar en el Sprint.
- **Sprint Goal:** El objetivo del Sprint es un compromiso de los desarrolladores que permite enfocarnos en lo que se quiere generar al finalizar el Sprint.

- **Increment:** El incremento es el avance hacia el objetivo del producto, es la suma de todos los elementos del product backlog que han sido completados en los Sprints. Para considerar incremento debe cumplir con la definición de hecho.
- **Definition of Done:** La definición de hecho son las condiciones necesarias que debe cumplir el elemento pendiente del trabajo para considerarse como incremento. El equipo Scrum es el que debe definir y ajustar según considere necesario estas condiciones.

Figura 2

Marco de trabajo Scrum



Nota: gráfico obtenido de Scrum.org. <https://www.scrum.org/resources/scrum-framework-poster>.

3.2.4. Fundamentos utilizados

3.2.4.1. Antecedentes Seleccionados

- **Mulili, F. M. (2024).** *Effects of digital banking channels on customer experience within microfinance institutions in Kenya.*
 - **Objetivo:** Determinar los efectos de la banca digital en la experiencia de los clientes entre las instituciones bancarias de microfinanzas en Kenia.
 - **Metodología:** Encuestas a 400 clientes en las 14 instituciones de microfinanzas de Kenia
 - **Hallazgos:** Los diferentes canales digitales generan satisfacción en el cliente; debido a que les brinda seguridad, rapidez, accesibilidad las 24 hora desde cualquier dispositivo.
 - **Relevancia:** El estudio sugiere que, para mejorar la experiencia de sus clientes, deberían aprovechar los beneficios que brindan los diferentes canales digitales. Esto genera mayor confianza y hay un mayor valor percibido hacia ellos. Pero deberían venir acompañado del aseguramiento de estas tecnologías con el fin de evitar ataques que puedan trasladarse en robos de fondos o información.

- **López, S. P., & Urquía, G. E. (2023).** *Mind the Gap in Financial Inclusion! Microcredit Institutions fieldwork in Peru.*
 - **Objetivo:** Encontrar interrelaciones en inclusión y educación financieras.
 - **Metodología:** Encuesta semiestructurada a 29 personas agentes de microfinanzas.
 - **Hallazgos:** La bancarización y financiamiento para la población es parte de la inclusión financiera, pero es necesario que venga acompañada de educación financiera y debería estar apoyado de las TIC.
 - **Relevancia:** Si bien la inclusión financiera debe ser complementada por la educación financiera esta segunda debe estar adaptada las diferentes necesidades de la población Y las instituciones de microfinanzas deben centrarse en sectores potenciales y prometedores.

- **(Credicorp,2024). Estudio de Índice de inclusión financiera 2024**
 - **Objetivo:** Evolución de la inclusión financiera, en 8 países de la región en el año 2024
 - **Metodología:** Encuesta a 13,443 personas mayores de 18 años de los 8 países.
 - **Hallazgos:** Hay tendencia positiva en la inclusión financiera tanto a nivel Latinoamérica, como Perú además de un aumento en la adopción de las billeteras digitales.
 - **Relevancia:** Del estudio se puede sugerir que, para poder tener más acceso a la inclusión financiera, deberíamos apoyarnos en Los medios digitales
- **inDrive Money (2024):** InDrive lanza un nuevo beneficio de servicio financiero para sus conductores en el que podrán acceder a pequeños préstamos lo pueden hacer directamente de la aplicación que además cuenta con las funciones de visualizar saldo y de realizar el pago directamente.
Esto beneficia en gran medida a los conductores para que puedan realizar mejoras en sus vehículos y así como score crediticio.

3.2.4.2 Spring Boot

Spring Boot es un framework diseñado para Java que facilita la creación de aplicaciones sobre el ecosistema Spring. Su principal ventaja es que permite a los desarrolladores configurar y lanzar aplicaciones de forma ágil y sencilla, eliminando la necesidad de configuraciones complejas (Spring, n.d.).

En su momento se trabajó la versión 2.

3.2.4.3 Spring Data

Spring Data es otra herramienta que es parte del ecosistema Spring y está diseñado para facilitar el acceso a bases de datos y otros sistemas de almacenamiento. Este proyecto permite a los desarrolladores gestionar datos de manera más eficiente, ya que ofrece soporte para diversas tecnologías de almacenamiento. Además, incluye características útiles, como la generación automática de repositorios, lo que simplifica aún más el trabajo con datos (Spring, n.d.).

3.2.4.4 Java 8

Java 8 es una versión del popular lenguaje de programación Java que en su momento trajo consigo una serie de nuevas características. Entre ellas las expresiones lambda y la API de Streams, las cuales son muy utilizados hoy en día en las nuevas versiones, que facilitan la escritura de código más limpio y eficiente. Estas mejoras en la programación funcional permiten a los desarrolladores abordar sus lógicas de una manera más ágil y concisa (Oracle, n.d.).

3.2.4.5 Maven

Maven es una herramienta diseñada para ayudar a gestionar y comprender los proyectos en Java de manera más efectiva. Ofrece funcionalidades que simplifican la construcción del software, la gestión de dependencias y la creación de documentación. Su enfoque en la eficiencia y la organización permite a los desarrolladores mantener sus proyectos en buen orden (Apache Software Foundation, n.d.).

3.2.4.6 SonarQube 9

SonarQube es una herramienta de análisis de código estático que se gestiona de forma automática, lo que te ayuda a asegurar la entrega de código limpio de manera sistemática. Y se adapta a cualquier flujo de trabajo para detectar problemas en el código, facilitando inspecciones continuas en tus proyectos. Esta herramienta es capaz de analizar más de 30 lenguajes de programación distintos y se integra perfectamente en la canalización de CI y en la plataforma DevOps, garantizando que el código desarrollado cumpla con altos estándares de calidad (SonarSource, 2020).

3.2.4.7 Junit5

JUnit 5 es un framework para pruebas en Java que permite escribir y ejecutar pruebas de forma más efectiva. Proporciona una arquitectura modular que soporta pruebas tanto en aplicaciones basadas en Java8 como en versiones posteriores. Su diseño incluye varias mejoras sobre versiones anteriores, como un soporte más robusto para la programación funcional y una mayor flexibilidad en la configuración de las pruebas (JUnit Team, n.d.).

3.2.4.8 Mockito

Mockito es otro framework de pruebas para Java que permite crear objetos simulados (mocks) de forma intuitiva. Este enfoque permite aislar y probar componentes específicos de una aplicación sin depender de sus implementaciones reales (Mockito, n.d.).

3.2.4.9 Azure

Azure es una plataforma de computación en la nube de Microsoft que ofrece una amplia gama de servicios que permiten a las empresas desarrollar y gestionar aplicaciones y servicios en una red global de centros de datos (Microsoft, 2023).

3.2.4.10 App Services

Azure App Services es una plataforma que simplifica el proceso de desarrollo y despliegue de aplicaciones web y servicios backend, permitiendo a los desarrolladores centrarse en la creación de soluciones sin preocuparse por la gestión de la infraestructura (Microsoft, 2024).

3.2.4.11 Git

Git es un sistema de control de versiones distribuido que ayuda a los desarrolladores a supervisar y gestionar los cambios en su código, promoviendo la colaboración efectiva al permitir que cada miembro del equipo trabaje en sus propias copias y combine modificaciones sin perder el historial (Git, n.d.).

3.2.4.12 Bitbucket

Bitbucket es una plataforma de gestión de repositorios de código que permite a los equipos de desarrollo trabajar en proyectos de software mediante el control de versiones y la colaboración, proporciona herramientas para la revisión de código y el seguimiento de problemas, lo que optimiza la integración de procesos y flujos de trabajo ágiles (Bitbucket, n.d.).

3.2.4.13 Lombok

Lombok es una librería de Java que permite a los desarrolladores minimizar el código repetitivo por ejemplo los Getters/Setters mediante el uso de anotaciones que generan automáticamente métodos comunes en tiempo de compilación, facilitando así un enfoque más limpio y enfocado en la lógica del negocio (Lombok, n.d.).

3.2.4.14 Swagger

Swagger es un conjunto de herramientas y especificaciones que ayudan a diseñar y documentar APIs RESTful de forma clara y eficiente, utilizando una estructura estandarizada que fomenta una mejor colaboración entre equipos (Swagger, n.d.).

3.2.4.15 Azure Logic Apps

Azure Logic Apps es una plataforma en la nube que permite a los usuarios crear y ejecutar flujos de trabajo automatizados con poco o ningún conocimiento de programación. Utilizando un diseñador visual y una variedad de operaciones predefinidas, puedes construir rápidamente flujos de trabajo que integren y gestionen aplicaciones, datos, servicios y sistemas (Microsoft, 2024).

3.2.4.16 Azure Blob Storage

Azure Blob Storage es un servicio de almacenamiento en la nube el cual permite gestionar grandes cantidades de datos no estructurados, y proporciona un acceso seguro y eficiente a estos recursos a través de la web, lo que resulta útil para diversas aplicaciones y análisis (Microsoft, 2023).

3.2.4.17 Azure Functions

Azure Functions es un servicio en la nube de computación sin servidor también conocido como serverless, el cual permite a los desarrolladores reducir la cantidad de código que debe escribir y la preocupación por la gestión de la infraestructura y los costos asociados a ello (Microsoft, 2023).

3.2.4.18 PlantUML

PlantUML es una herramienta que permite crear diversos diagramas de manera rápida y sencilla. También conocido como Diagramas como código, su lenguaje es fácil de usar y puedes generar diagramas rápidamente (PlantUML, n.d.).

3.2.4.19 Slack

Slack es una plataforma de comunicación que facilita la colaboración en equipos de trabajo. Permite a los usuarios intercambiar mensajes, compartir archivos y organizar conversaciones en canales específicos, además tiene integrada opciones de videollamadas y gestión de proyectos. Su interfaz es intuitiva y personalizable (Slack, n.d.).

3.2.4.20 Jira

Jira es una herramienta para la gestión de proyectos ágiles que permite a los equipos planificar, supervisar, lanzar y brindar soporte para software de alta calidad de manera segura. Actúa como la principal fuente de información a lo largo de todo el ciclo de vida del desarrollo, lo que permite a los equipos trabajar de forma ágil y mantenerse alineados con la estrategia del proyecto (Atlassian, n.d.).

3.2.4.21 Confluence

Confluence es una herramienta que permite a los equipos combinar tanto conocimiento como colaboración, en este espacio cada miembro puede documentar y colaborar en cualquier iniciativa de esta forma se puede gestionar el conocimiento y no se pierda en el tiempo (Atlassian, n.d.).

3.2.4.22 Visual Studio Code

Visual Studio Code es un editor de código fuente ágil y potente que funciona en el escritorio y es compatible con diferentes sistemas operativos. Tiene soporte para diferentes lenguajes de programación uno de los cuales es Java asimismo tiene un extenso ecosistema de extensiones que facilitan la forma de trabajar (Microsoft, n.d.).

3.2.4.23 SendGrid

SendGrid es una plataforma basada en la nube, que permite el envío de correo transaccional y de marketing, esta solución brindada permite centrarse en el negocio y no tanto en la infraestructura (SendGrid, n.d.).

3.2.4.24 Azure SQL Database

Azure SQL Database es un servicio de base de datos en la nube (PaaS) que se gestiona por sí mismo. Esto significa que se ocupa de tareas como actualizaciones, parches, copias de seguridad y supervisión, todo sin que se intervenga. Permitiendo centrarse en la aplicación y los datos. (Microsoft, 2024).

3.2.5. Implementación de las áreas, procesos, sistemas y buenas prácticas

Para iniciar la implementación del sistema backend, se hará uso del framework de Scrum en la cual se utilizará los diferentes eventos, para alinearse con el objetivo de negocio, el estado actual y los diversos flujos que conlleva realizar la solución al problema indicado. Involucra de manera complementaria reuniones técnicas para definir y entender los detalles de la implementación.

Dado que Scrum nos sugiere como podemos trabajar, a continuación, se narra la forma particular como se ha desarrollado y adoptado este framework en el día a día del equipo.

Recolección de requisitos y definición del backlog

El Product Owner, conociendo la necesidad de negocio, brinda el contexto para que el equipo pueda empaparse de toda la información posible y construir un producto de calidad.

Se utiliza la práctica del refinamiento, en donde se empieza a definir flujos, conceptos, tareas, historias, criterios de aceptación y toda actividad necesaria para la implementación del sistema.

Sincronización con negocio

La entidad bancaria ha realizado una alianza estratégica con una empresa de taxi aplicativo. El objetivo de esta colaboración es que el banco pueda validar un nuevo modelo de negocio aprovechando el nicho de mercado de los conductores de la aplicación. Por otro lado, la empresa de taxi aplicativo busca ofrecer beneficios a los conductores, también conocidos como socios conductores, permitiéndoles acceder a los servicios financieros del banco a través de la plataforma digital, donde podrán solicitar un microcrédito.

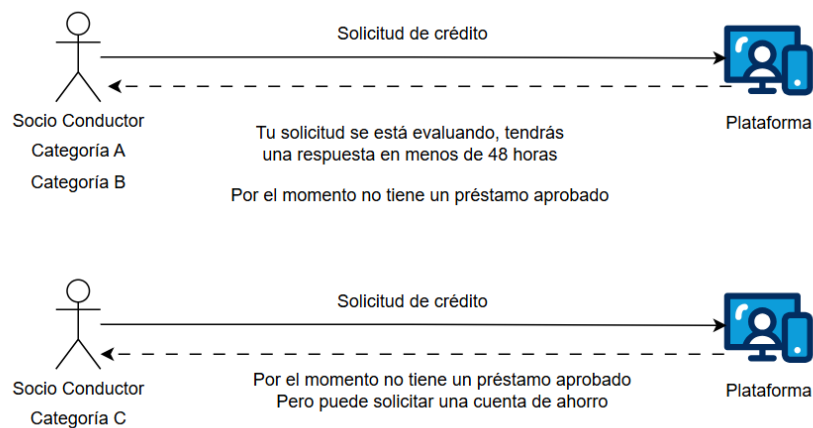
Estos microcréditos son pequeños préstamos que varían entre S/300 y S/1500, y se otorgan tras una evaluación de la situación financiera de los conductores. Además, los socios conductores se clasifican en diferentes categorías según su ingreso semanal; aquellos con mayores ingresos se ubican en categorías más altas, mientras que los de menor ingreso pertenecen a la categoría básica.

Como regla de negocio, los conductores en la categoría básica no podrán solicitar microcréditos por el momento. Sin embargo, podrían abrir una cuenta de ahorros. Las solicitudes de las categorías superiores también se registrarán y luego serán evaluadas tanto por sistemas internos como externos.

Si en el sistema interno un conductor tiene alguna condición financiera desfavorable se le indicará de manera rápida que no cumple con los requisitos para acceder al crédito. Si la solicitud no presenta inconvenientes, el sistema realizará una validación adicional con el sistema externo y el conductor recibirá una respuesta en un plazo de 48 horas.

Figura 3

Reglas de negocio definidas

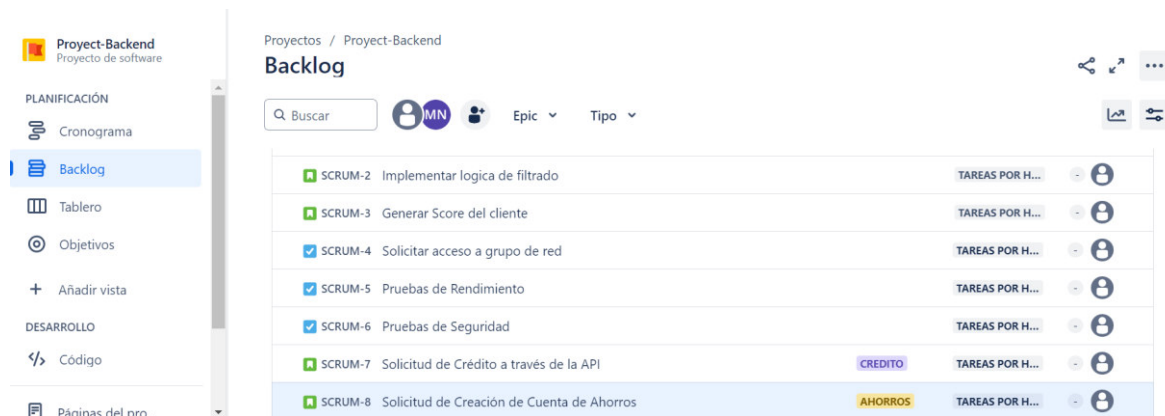


Nota: Fuente elaboración propia

Con el contexto del proyecto, las reglas de negocio, las historias y tareas definidas se agregan al backlog, complementándolo con los criterios de aceptación, que son las condiciones necesarias para considerarla como aceptada. Generalmente esta actividad de agregar las historias en el product backlog lo realiza el Product Owner. Para su posterior selección y puntaje por parte del equipo de desarrolladores en el Sprint Planning.

Figura 4

Ejemplo de un product backlog.

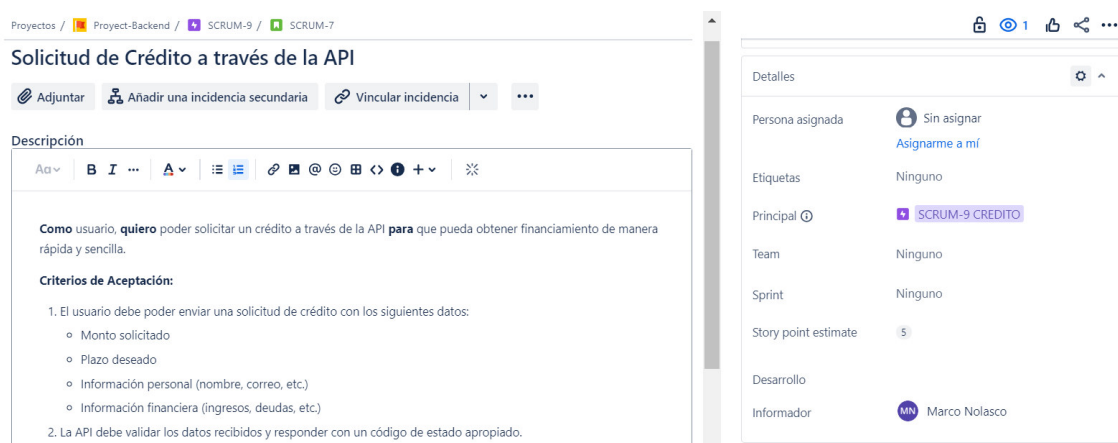


Nota: Fuente elaboración propia

Según las definiciones trabajadas en el proyecto las historias son las que agregan valor al negocio y las tareas son habilitadores para algunas historias, que por lo general son actividades repetitivas o muy técnicas.

Figura 5

Ejemplo donde se incluye criterios de aceptación a una historia.



Nota: Fuente elaboración propia

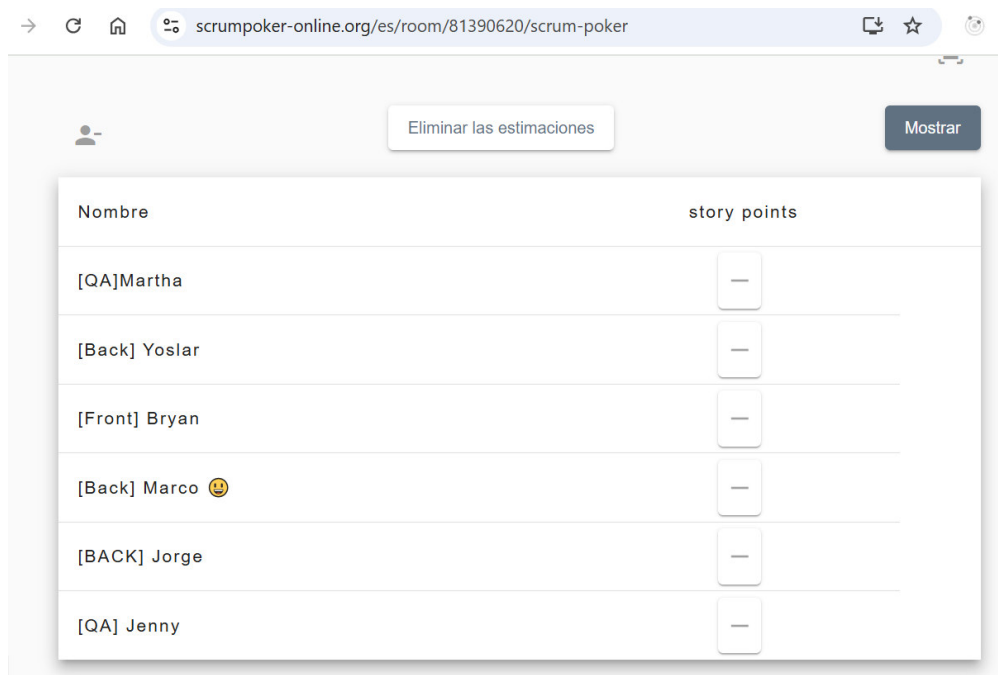
Planificación del sprint: En Sprint Planning, el Product Owner inicia explicando la necesidad del sprint; y cada una de las historias asociadas al objetivo. Con ello el equipo de desarrollo realiza una evaluación de las historias y define un puntaje; considerando la

complejidad, el tiempo que le podría tomar desarrollarlo, si existe alguna dependencia o bloqueante o si hay capacity del equipo.

Para ello se utiliza la herramienta en línea llamada ***scrumpoker-online.org***; el cual permite crear una habitación virtual; donde todos acceden mediante el link de o un código de que nos brinda la persona encargada de llevar la sesión, cada participante ingresa un nombre para poder diferenciarlo.

Figura 6

Sesión para estimar una historia en el planning.

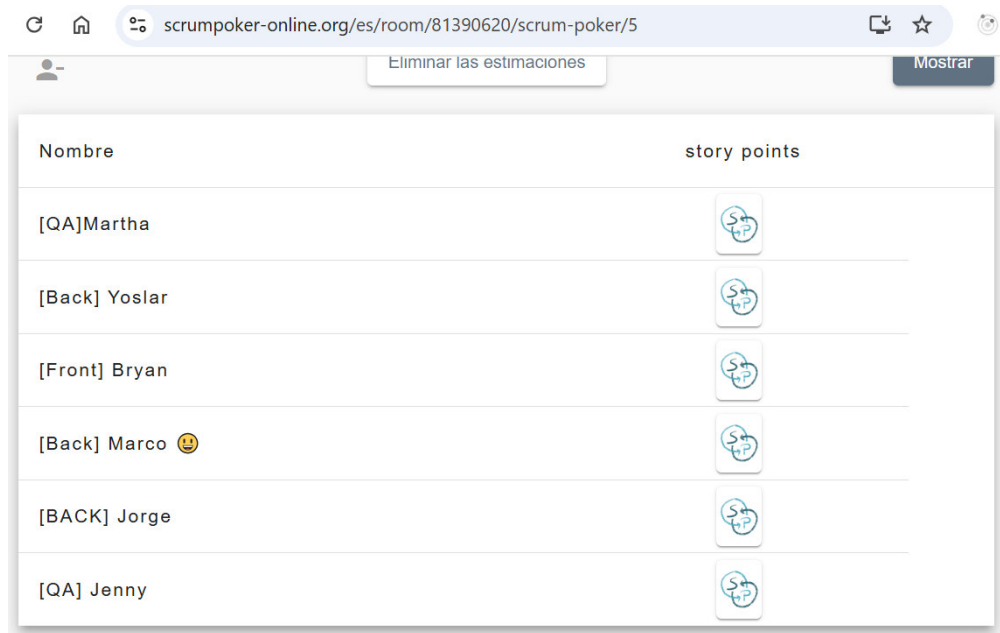


Nombre	story points
[QA]Martha	—
[Back] Yoslar	—
[Front] Bryan	—
[Back] Marco 😊	—
[BACK] Jorge	—
[QA] Jenny	—

Nota: Elaboración propia utilizando la web de ***scrumpoker-online.org***

Figura 7

Estimación de una historia de usuario.

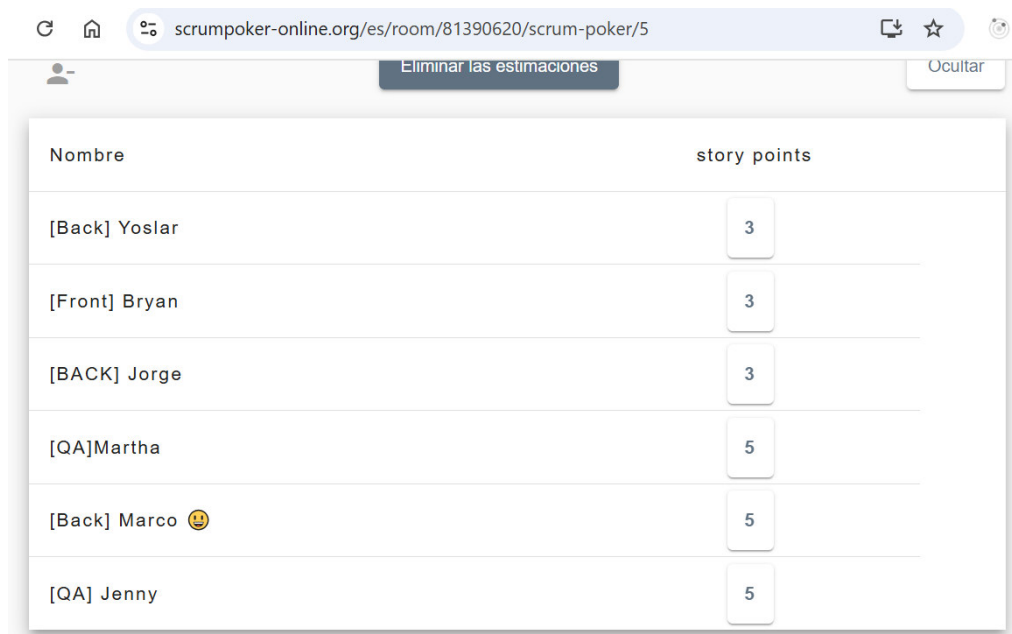


Nombre	story points
[QA]Martha	
[Back] Yoslar	
[Front] Bryan	
[Back] Marco 😊	
[BACK] Jorge	
[QA] Jenny	

Nota: Fuente elaboración propia utilizando herramienta *scrumpoker-online.org*

Figura 8

Resultados de la estimación



Nombre	story points
[Back] Yoslar	3
[Front] Bryan	3
[BACK] Jorge	3
[QA]Martha	5
[Back] Marco 😊	5
[QA] Jenny	5

Nota: Fuente elaboración propia utilizando la herramienta *scrumpoker-online.org*

Luego de mostrar los resultados; en caso todos tengan el mismo puntaje se considera que el equipo está de acuerdo con la complejidad y ha entendido el alcance de la historia.

En caso se muestre resultados similares a la imagen donde hay diferencias un representante del menor y mayor puntaje salen a sustentar porque han definido ese puntaje para la historia, con el sustento de ambos se vuelve a realizar la votación para definir el puntaje si todo ha quedado claro, el puntaje debería salir forma similar a la siguiente imagen.

Figura 9

Puntajes de la estimación luego de llegar a un consenso.

Nombre	story points
[QA]Martha	5
[Back] Yoslar	5
[Front] Bryan	5
[Back] Marco 😊	5
[BACK] Jorge	5
[QA] Jenny	5

Nota: Fuente elaboración propia utilizando la herramienta scrumpoker-online.org

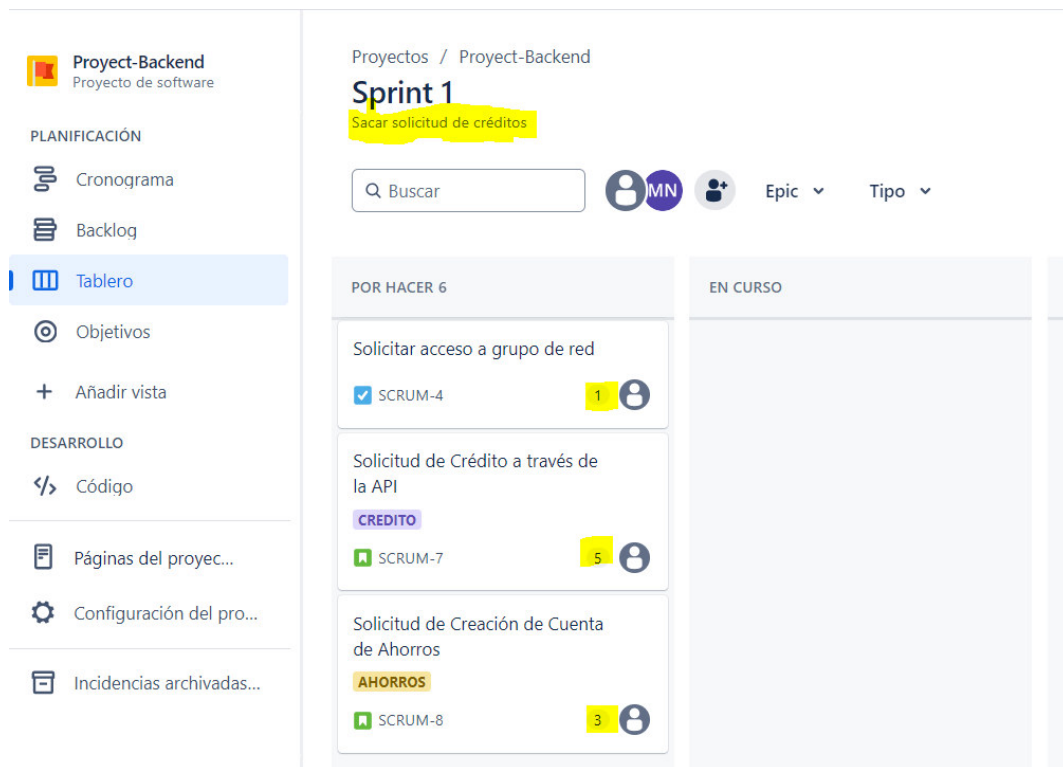
En caso se mantenga con la diferencia de puntajes; se considera el de mayor valor.

Durante esta asignación de puntaje también se aprovecha para indicar que integrante del equipo será responsable de la historia.

Al final la votación de todas las historias y tareas que entrarán en el sprint, el Product Owner define el objetivo del sprint y lo inicia en Jira.

Figura 10

Tablero Jira, con el sprint activo y su objetivo asociado.



Nota: Fuente elaboración propia

Este tipo de evento se realiza cada 2 semanas; el esfuerzo se estima utilizando los números de Fibonacci 1, 2, 3, 5 y 7 donde 1 y 5 son menor y mayor complejidad respectivamente.

Figura 11

Historias puntuadas y asignadas.

SCRUM-7	Solicitud de Crédito a través de la API	CREDITO	TAREAS POR H...	5	
SCRUM-8	Solicitud de Creación de Cuenta de Ahorros	AHORROS	TAREAS POR H...	3	

Nota: Fuente elaboración propia

En caso se estime con un valor mayor de 5, esto depende del equipo donde se trabaje, se entiende como una historia con demasiadas dependencias o el entendimiento de esta es muy gaseoso o ambiguo es por ello se sugiere dividir la historia en historias más simples.

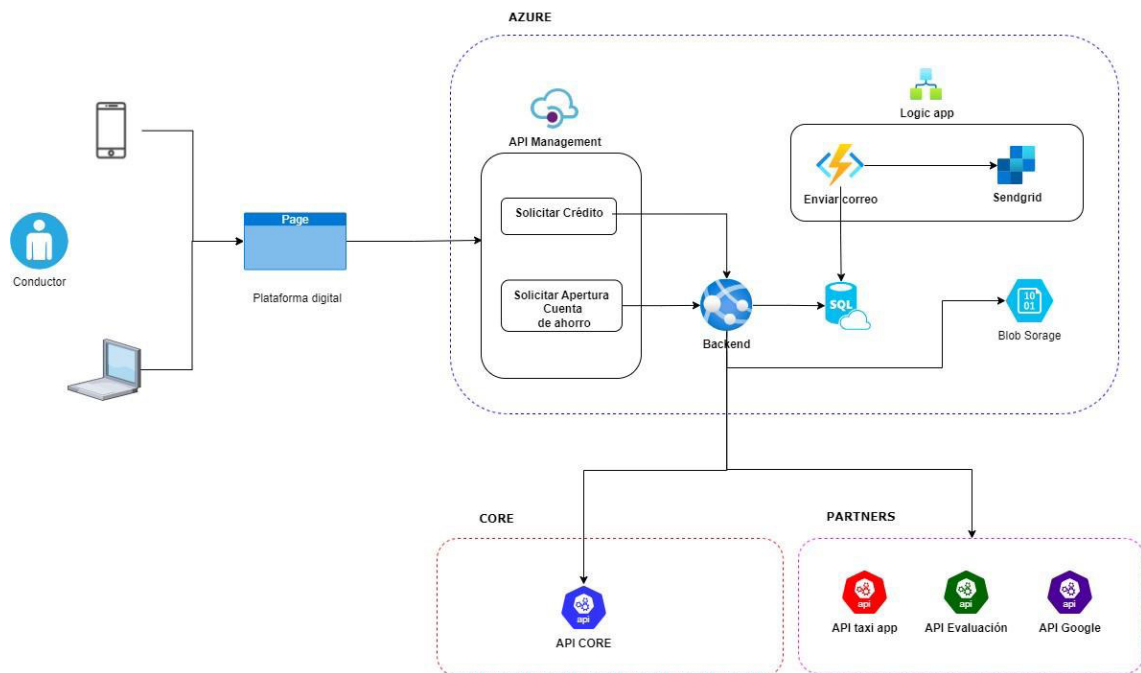
Desarrollo

Con las historias y/o tareas asignadas en Jira, cada responsable empieza a trabajar en una de ellas, tomando en cuenta los criterios de aceptación.

Como primera tarea del proyecto se desea tener los diagramas de arquitectura, componentes y secuencia; los cuales brindan claridad al equipo en las posibles decisiones que puedan tener a lo largo del desarrollo.

Figura 12

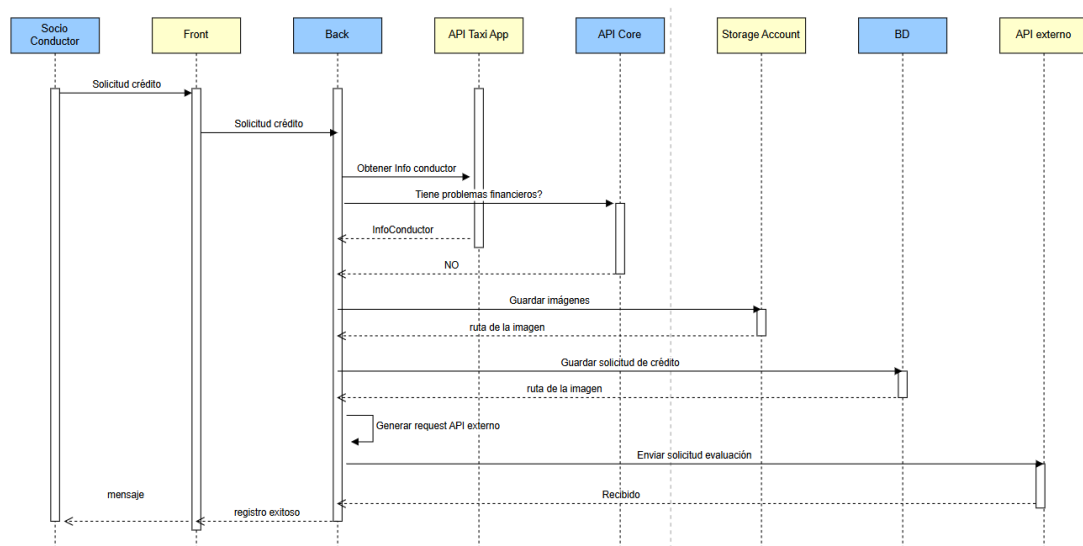
Diagrama de solución y las diversas interacciones



Nota: Fuente elaboración propia

Figura 13

Diagrama de secuencia solicitud de crédito.



Nota: Fuente elaboración propia

Las tareas definidas por el equipo Backend son:

- Definir una base de datos.
- Definir el contrato para el API Backend.
- Desarrollo de la funcionalidad.
- Pruebas unitarias
- Despliegue en ambiente de desarrollo.

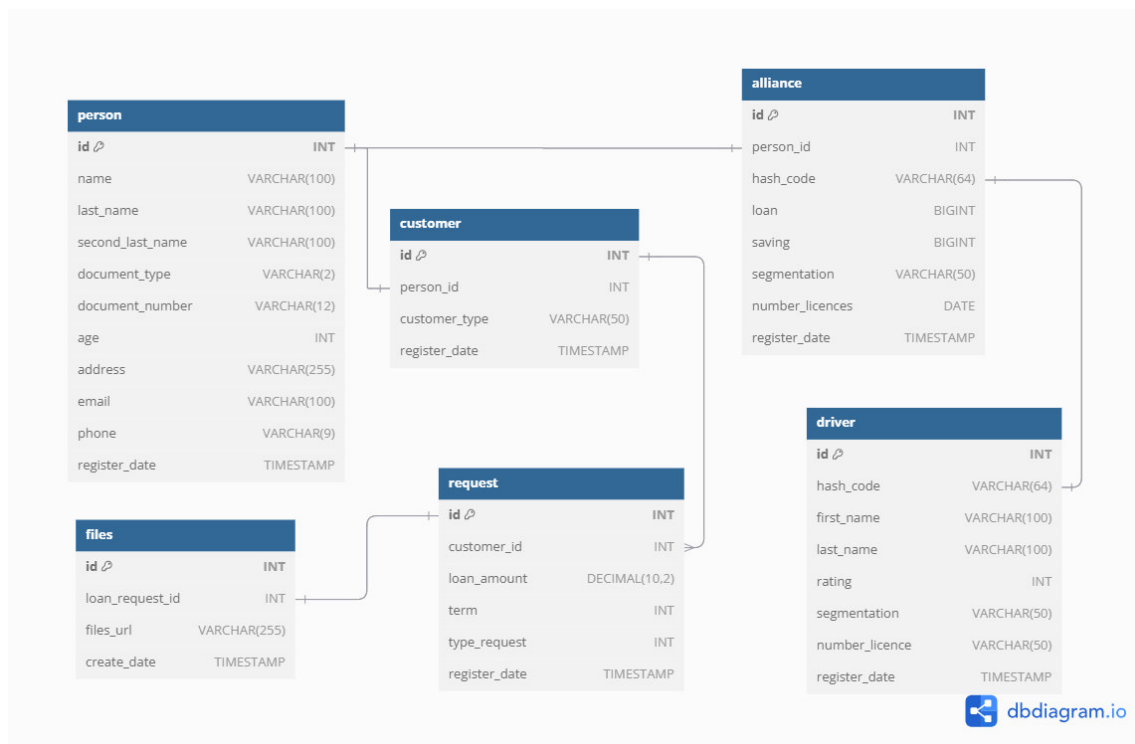
Definir base de datos

El propósito de la base de datos está relacionado con la gestión de solicitudes de microcréditos lo que implica almacenar datos de usuarios y solicitudes créditos. Para ello se utilizarán las entidades **person**, **customer**, **alliance**, **request**, **files**, **driver**

En el cual se almacenarán texto, números, fechas e imágenes; para este propósito se utilizará una base de datos relacional como Azure SQL Database.

Figura 14

Modelo entidad relacional para el proyecto backend.



Nota: Fuente elaboración propia, utilizando herramienta online dbdiagrama.io

Definir el contrato para el API Backend.

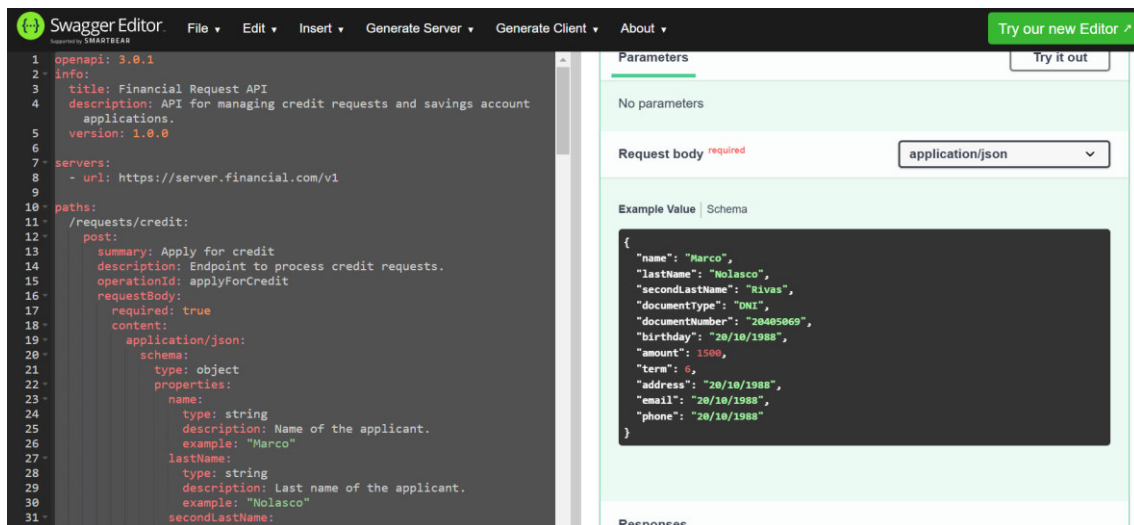
Con el objetivo claro, se define las especificaciones del API.

Se construirá los 2 endpoints uno para solicitud de créditos y otro para cuenta de ahorros.

1. Para cada uno de los endpoints se necesita definir los datos de entrada y salida. y los métodos HTTP para luego documentarlo en la herramienta de diseños de API SwaggerEditor, utilizando OpenAPI esto se conoce el enfoque primero el contrato.
2. Esta definición se realiza con el equipo de frontend para evitar retrabajo, y ambos frentes puedan tener el mismo concepto.

Figura 15

Definición del contrato para el backend.



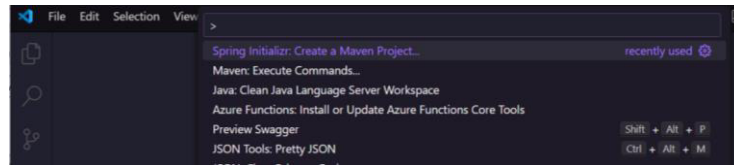
Nota: Fuente elaboración propia utilizando SwaggerEditor.

Definimos la estructura que tendrá nuestro proyecto a nivel de carpetas

Desde el IDE Visual Studio Code, utilizando la opción de Spring Initializr se generará un proyecto Maven; para su posterior organización a nivel de carpetas.

Figura 16

Creación de un nuevo proyecto



Nota: Fuente de elaboración propia

Luego selecciona los siguientes valores para poder generar el proyecto donde se trabajará la funcionalidad en mención.

Project: Maven

Language: Java

Spring Boot: 3.2.11

Project Metadata

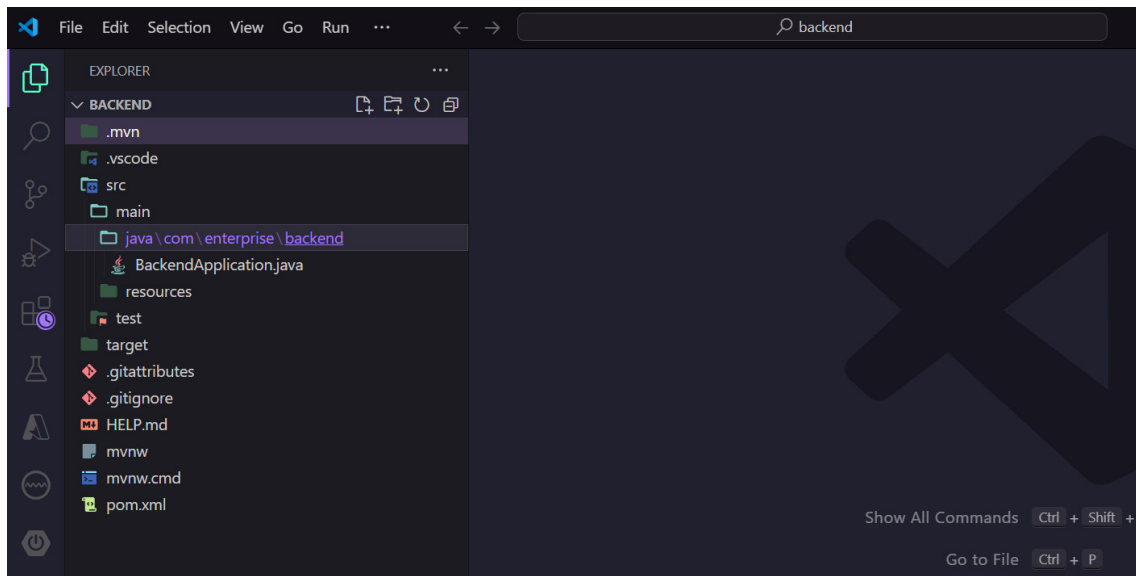
- **Group:** com.enterprise
- **Artifact:** backend
- **Name:** backend
- **Packaging:** JAR
- **Java Version:** 11

Considerar las siguientes dependencias para el proyecto:

- **Spring Web:** Permite utilizar componentes de Web MVC.
- **Spring Data JPA:** Para poder trabajar con la base de datos.
- **Spring Boot DevTools:** Para que cada cambio en el código recarga automáticamente la aplicación. Ayudando a mejorar tiempos
- **Lombok:** Ayuda a evitar agregar código repetitivo, como los Getters y Setters.

Figura 17

Estructura inicial del proyecto

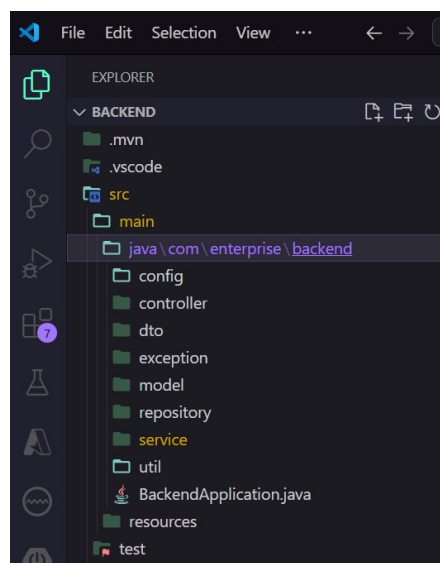


Nota: Fuente elaboración propia

Luego se pasa a estructura el proyecto con las diferentes carpetas en según la imagen

Figura 18

Estructura de carpetas del proyecto



Nota: Fuente elaboración propia

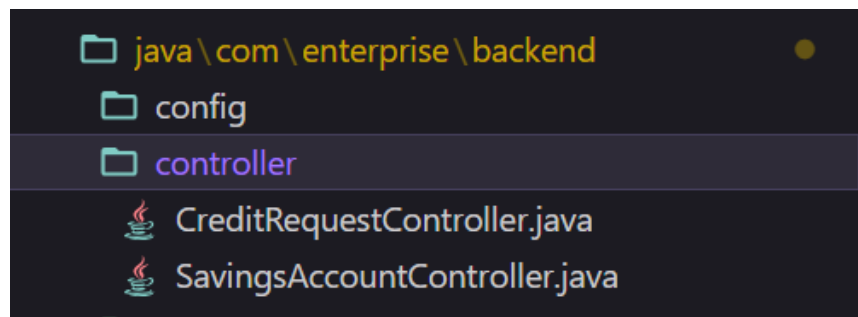
Cerrada la estructura del proyecto, se puede empezar agregar las diferentes funcionalidades.

Cada una de las carpetas del proyecto tienen diferentes responsabilidades:

- **config.** – Contiene las clases de configuración para el proyecto.
- **controller.** – Maneja las solicitudes HTTP, recibe y envía las solicitudes a través de los endpoints REST.

Figura 19

Carpeta controller con sus clases.

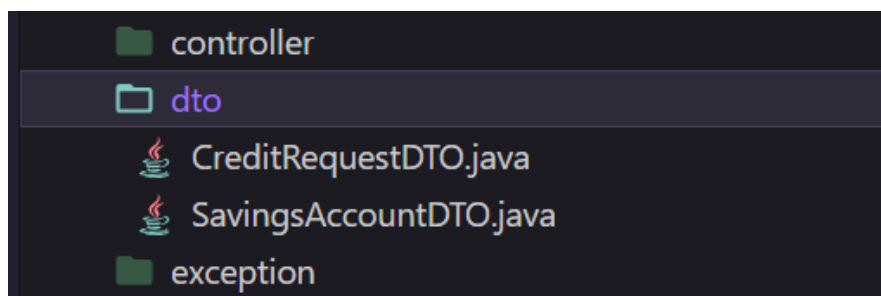


Nota: Fuente elaboración propia

- **dto.** - Contiene las clases para transferir datos entre el controlador y la capa de servicio.

Figura 20

Carpeta dto con sus clases.

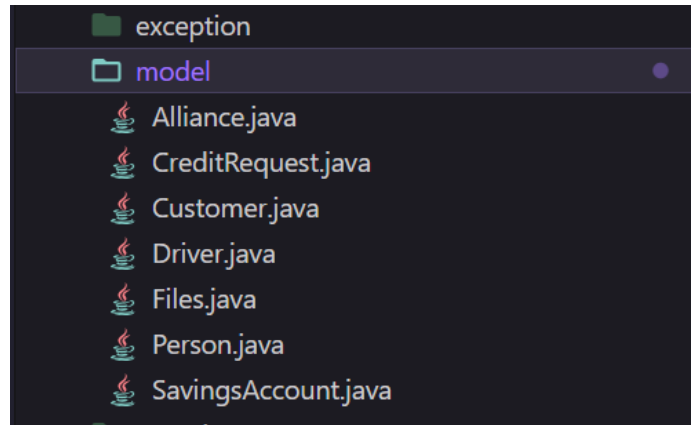


Nota: Fuente elaboración propia

- **model.** -Contiene las clases que representan las entidades de base de datos.

Figura 21

Carpeta model con sus clases.

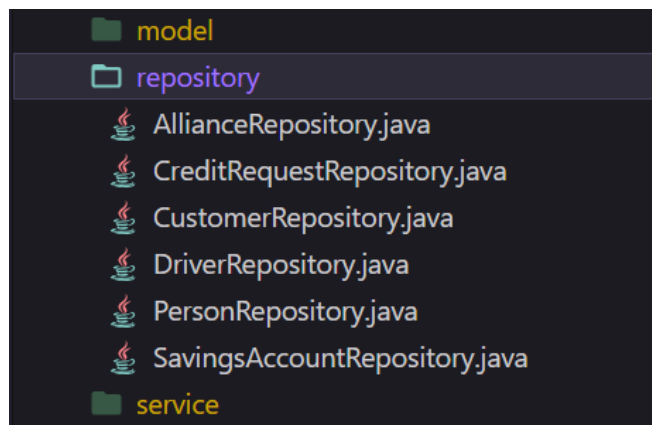


Nota: Fuente elaboración propia

- **repository.** - Contiene las interfaces de los repositorios los cuales permitirán la interacción con la base de datos.

Figura 22

Carpeta repository con sus clases.

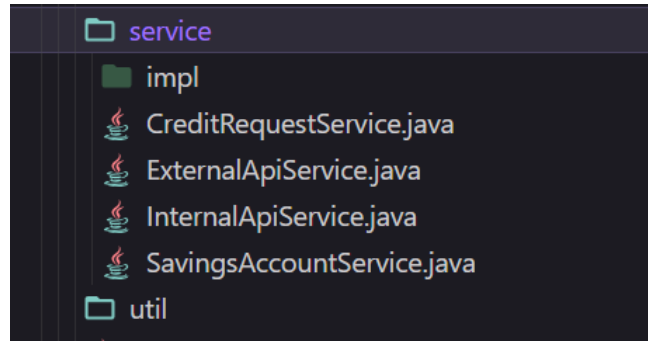


Nota: Fuente elaboración propia

- **services.** - Contiene la lógica de negocio, tanto para las solicitudes de crédito y cuentas de ahorros.

Figura 23

Carpeta service con sus clases.

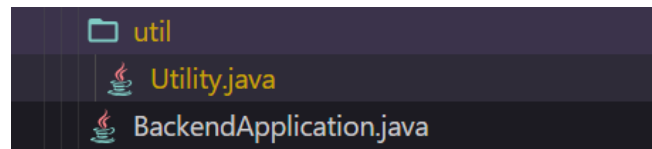


Nota: Fuente elaboración propia

- **util.** – Contiene las clases utilitarias para el proyecto.

Figura 24

Carpeta util con su clase asociada.



Nota: Fuente elaboración propia

Sistema de control de versiones

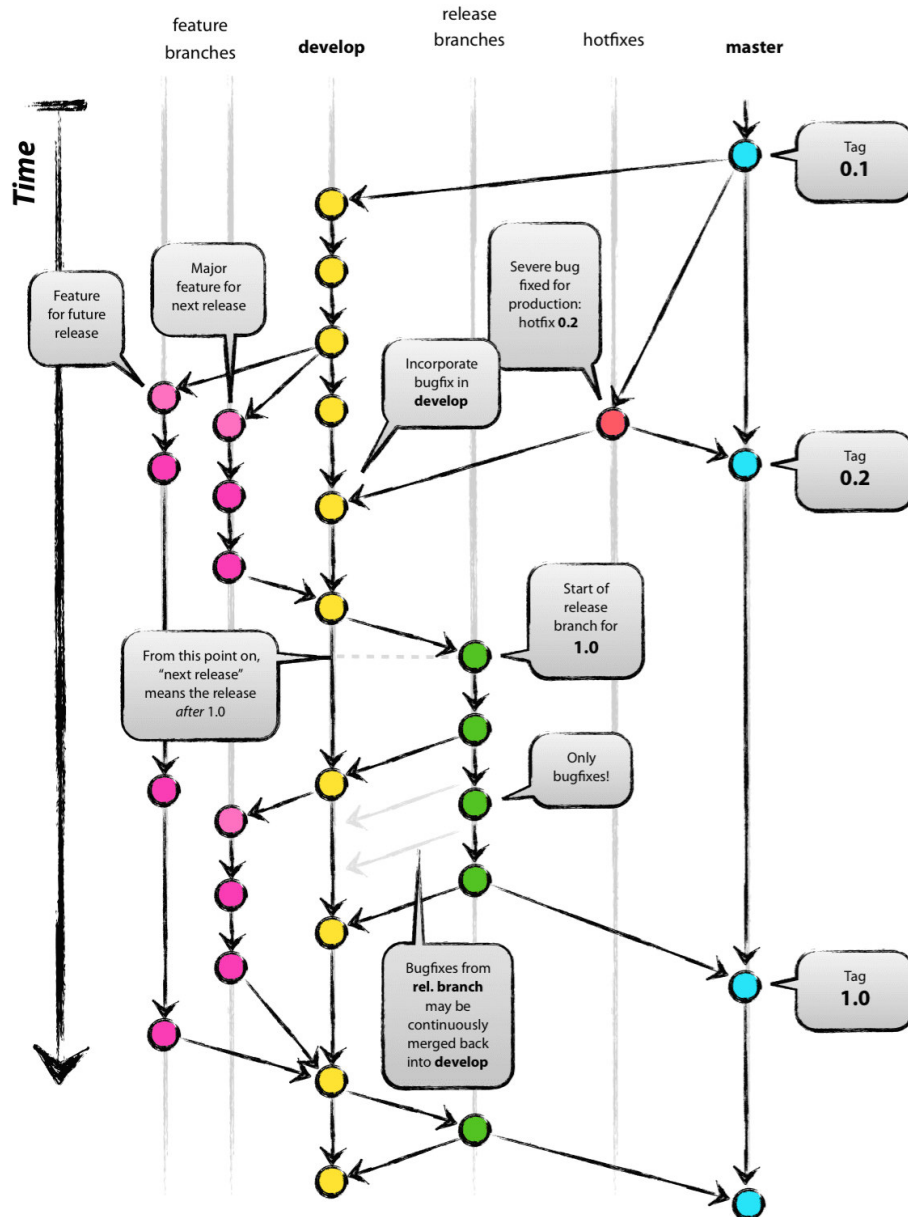
Para gestionar el flujo de trabajo utilizando git, se utiliza una estrategia de ramificación que se conoce como Gitflow el cual permite organizar el trabajo en diferentes ramas entre las cuales tenemos.

- **master**
- **develop**
- **feature**
- **bugfix**
- **hotfix**
- **release**

En la siguiente imagen se puede observar las diferentes ramas, como y cuando se crean y se fusionan.

Figura 25

Modelo de Branch Gitflow.



Nota: Imagen obtenida de <https://nvie.com/posts/a-successful-git-branching-model/>

La rama **develop**; es donde se fusiona el desarrollo de todas las funcionalidades trabajadas, se considera una rama estable, a partir de ella se genera otra llamada release.

Por otro lado, las ramas **feature**, se crean cuando se necesita trabajar en un nuevo requerimiento, se toma como base develop y una vez terminada el desarrollo se fusiona en la rama origen mediante un pull request.

Los pull request (PR) son solicitudes que hace el desarrollador al equipo para poder fusionar los cambios de una rama feature o fix a una rama estable como develop o master.

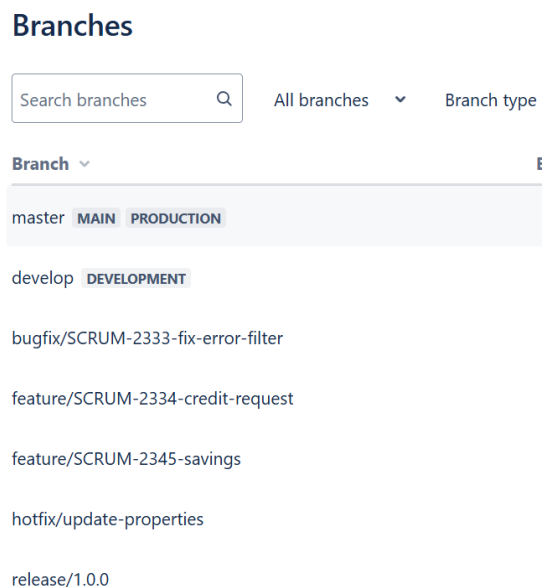
Cada integrante revisa el cambio de código e ingresa un comentario, si hubiera una observación, sino brinda su conformidad. Cuando todos aprueban el PR se realiza el merge o fusión a la rama estable. Si tuviera muchos errores que corregir se puede declinar el pull request para que el encargado de la rama de origen realice las correcciones.

En siguiente lugar se encuentra la rama estable **release**, la que es candidata para desplegar en certificación nace a partir de **develop**, y se genera después de las pruebas en desarrollo. Una vez certificada la rama release se hace un PR hacia master, el cual se utilizará para hacer el despliegue en producción.

En caso hubiera algún error en producción, se utiliza la rama **hotfix**, la cual nace de **master** y se crea para hacer la corrección, finalmente cuando se termina se fusiona el cambio en **master** y **develop**.

Figura 26

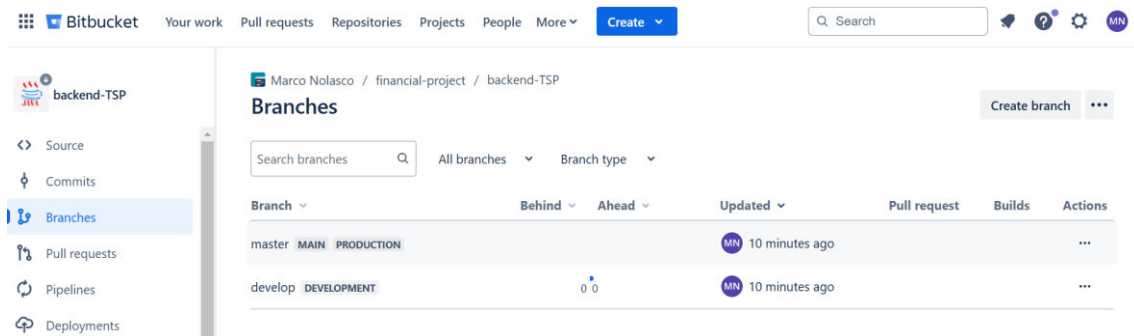
Ejemplos de nombre de las ramas en Bitbucket.



Nota: Fuente elaboración propia

Figura 27

Definición de ramas principales



Nota: Fuente de elaboración propia

En Bitbucket se define como se trabajará las ramas, y se especifica quien será la rama de producción y desarrollo.

Figura 28

Especificación de ramas principales en Bitbucket.

Branching model

Define how you want your branches to be named, and specify your development and production branches, to help us suggest source and target branches. [Learn more](#)

☒ Enable inherited settings

Development branch ⓘ

Use specific branch ▼ develop ▼

Production branch ⓘ

Use main branch ▼ Currently set to master

Nota: Fuente de elaboración propia.

Con el desarrollo avanzando o terminado se sube los cambios al repository en Bitbucket, pero primero inicializamos un repositorio Git Local

Figura 29

Inicialización repositorio Git local.

```
$ git init
Initialized empty Git repository in C:/Users/mnolascr/POCS/Project/backend/.git/
```

Nota: Fuente de elaboración propia

Luego agregaremos todos los archivos que se ha desarrollado en esta funcionalidad

Figura 30

Agregar archivos al espacio de trabajo.

```
$ git add .
warning: in the working copy of '.gitattributes', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.mvn/wrapper/maven-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mvnw.cmd', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'pom.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/java/com/enterprise/backend/BackendApplication.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/application.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/test/java/com/enterprise/backend/BackendApplicationTests.java', LF will be replaced by CRLF the next time Git touches it
```

Nota: Fuente de elaboración propia

Asimismo, se tiene que realizar el commit haciendo una breve descripción de los cambios a subir.

Commits

La forma de trabajar a nivel de commits, se utiliza el concepto de commits atómicos; donde se trata de tener la menor cantidad de commits y los mensajes sean concisos y describan la funcionalidad que se está subiendo a las ramas.

Figura 31

Cambios comiteados en local.

```
$ git commit -m "SCRUM-7 Add endpoint request Credit"
[master (root-commit) 46f0f58] SCRUM-7 Add endpoint request Credit
37 files changed, 918 insertions(+)
create mode 100644 .gitattributes
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/com/enterprise/backend/BackendApplication.java
create mode 100644 src/main/java/com/enterprise/backend/config/AppConfig.java
create mode 100644 src/main/java/com/enterprise/backend/controller/CreditRequestController.java
create mode 100644 src/main/java/com/enterprise/backend/controller/SavingsAccountController.java
create mode 100644 src/main/java/com/enterprise/backend/dto/CreditRequestDto.java
create mode 100644 src/main/java/com/enterprise/backend/dto/SavingsAccountDto.java
create mode 100644 src/main/java/com/enterprise/backend/model/Alliance.java
```

Nota: Fuente de elaboración propia

Finalmente agregamos el repositorio remoto de Bitbucket donde subiremos nuestros cambios, pero antes hacemos el cambio de nombre a una rama feature.

Figura 32

Agregar repositorio remoto y cambio de nombre de rama

```
$ git remote add origin https://mnolascor@bitbucket.org/mnolascor/backend-tsp.git
$ git branch -m feature/SCRUM7-implements-request-credit
```

Nota: Fuente de elaboración propia

Una vez creada la nueva rama, se sube los cambios a develop.

Figura 33

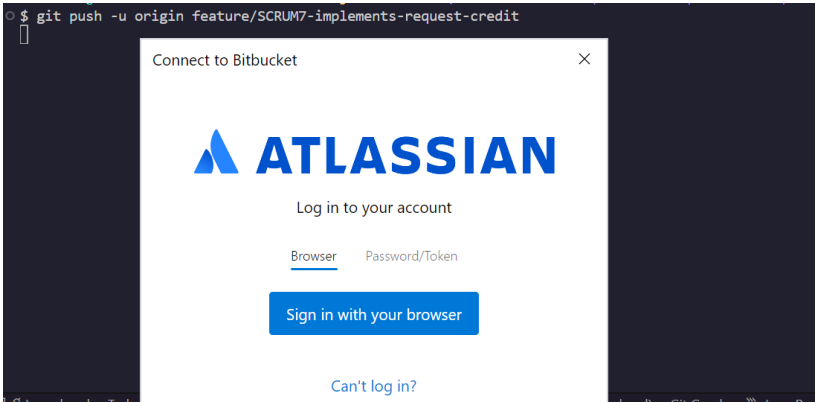
Subir cambios a la nueva rama.

```
$ git push -u origin feature/SCRUM7-implements-request-credit
```

Nota: Fuente de elaboración propia

Nos pedirá conectar con Bitbucket, mostrándonos las opciones disponibles, para luego confirmar el acceso.

Figura 34 Bitbucket solicita acceso.

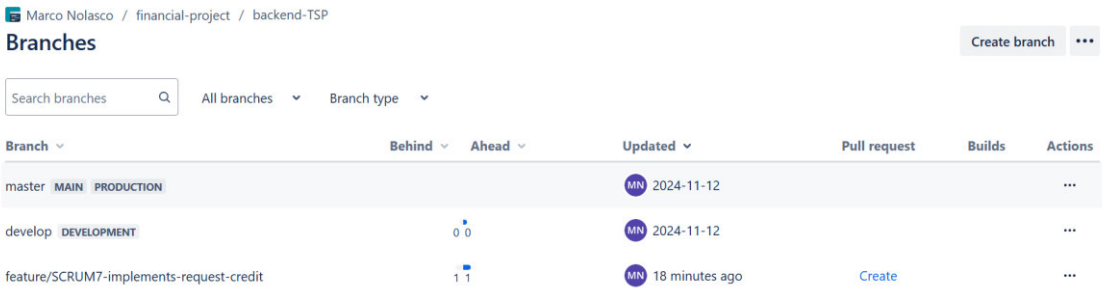


Nota: Fuente de elaboración propia

Finalmente subirá los cambios al repositorio remoto similar a la imagen indicada, Para su posterior fusión en la rama develop.

Figura 35

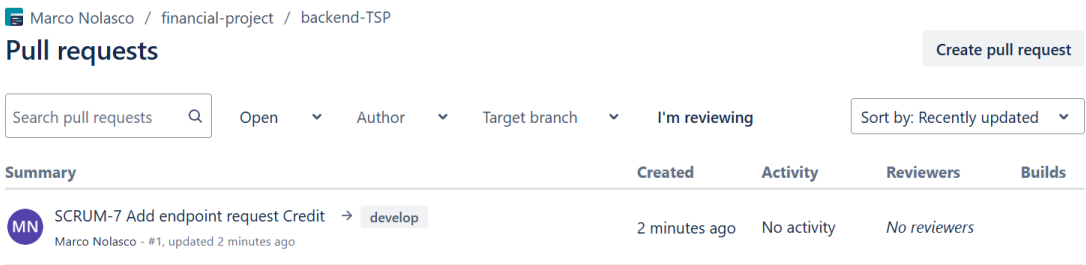
Nueva rama feature en Bitbucket.



Nota: Fuente de elaboración propia

Figura 36

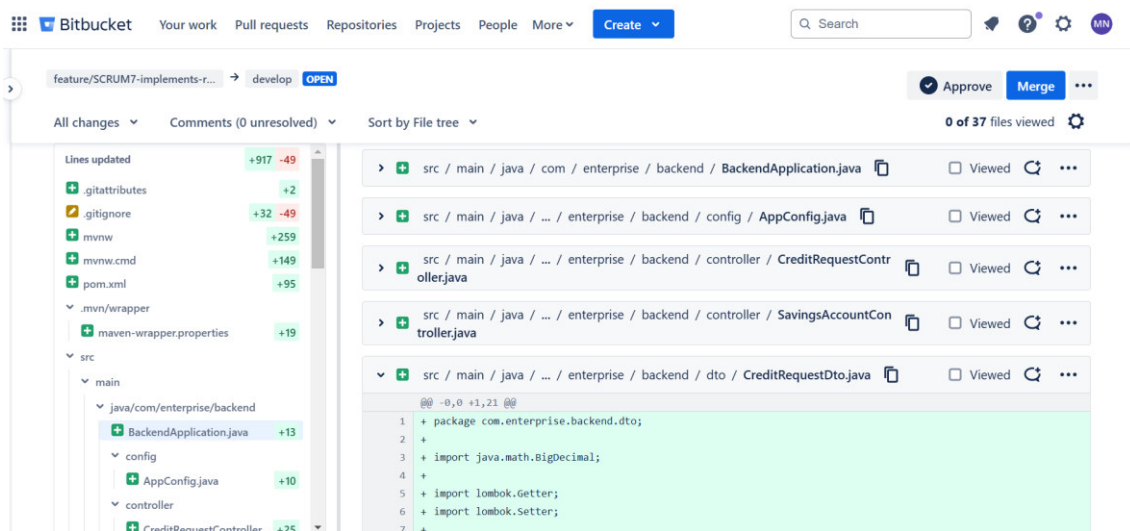
Pull Request desde una rama feature hacia develop.



Nota: Fuente Bitbucket

Figura 37

Revisión de los cambios en un pull request.



Nota: Fuente Bitbucket

Pruebas Unitarias (Unit Test)

Para poder asegurar código desarrollado y que cumpla con lo solicitado. Se realiza las pruebas unitarias, tomando en consideración la matriz de pruebas, y algunos casos adicionales que pueda considerar el desarrollador. Para ellos se utiliza dos frameworks; Junit y Mockito el primero para realizar las pruebas unitarias y simular dependencias externas respectivamente.

La estructura típica de una prueba unitaria cumple con el patrón AAA donde:

- **Preparación** (Arrange): En esta parte se inicializan todas las variables de entrada necesarias.
- **Ejecución** (Act): Aquí se llama al método que se probará.
- **Verificación** (Assert): Finalmente se verifica el resultado obtenido del método sea el mismo del esperado.

Figura 38

Ejemplo de unit test.

```
class CreditRequestServiceImplTest {  
  
    @InjectMocks  
    CreditRequestServiceImpl creditRequestService;  
  
    @Mock  
    InternalApiServiceImpl internalApiService;  
  
    @Test  
    @DisplayName("Response when request credit is reject")  
    void responseWhenRequestCreditIsReject() {  
        //Arrange  
        CreditRequestDto creditRequestDto = mock(CreditRequestDto.class);  
        CreditRequest creditRequest = mock(CreditRequest.class);  
        when(internalApiService.validateNegativeBase(creditRequest)).thenReturn(true);  
        //Act  
        String response = creditRequestService.processCreditRequest(creditRequestDto);  
        //Assert  
        assertEquals("Por el momento no tenemos un crédito para ti.", response);  
    }  
}
```

Nota: Fuente de elaboración propia.

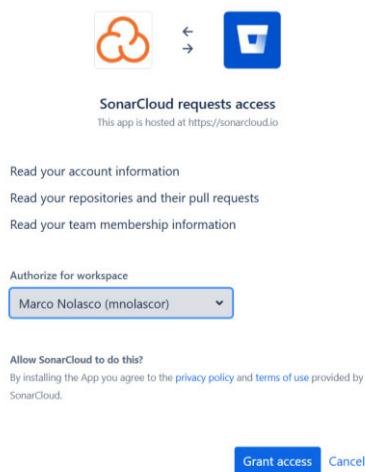
Cobertura de código (Coverage): Es una métrica que nos indica cuanto del código desarrollado ha sido cubierto por las pruebas unitarias, para el proyecto se tiene como lineamiento el coverage debe ser mayor al 85%. Esta métrica nos brinda la herramienta SonarQube o su equivalente en la nube SonarCloud.

Como se utiliza el repositorio de Bitbucket, esta herramienta tiene interacción con otros servicios como Azure, AWS, SonarCloud mediante pipelines.

Para ello en Sonarcloud se conecta con Bitbucket, y así obtener los repositorios para que puedan ser analizados.

Figura 39

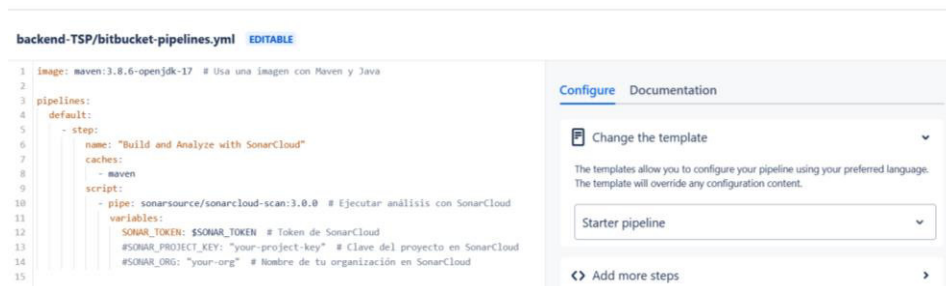
SonarCloud solicita acceso a Bitbucket



Nota: Fuente SonarCloud

Figura 40

Pipeline para integrar con SonarCloud.



Nota: Fuente Bitbucket

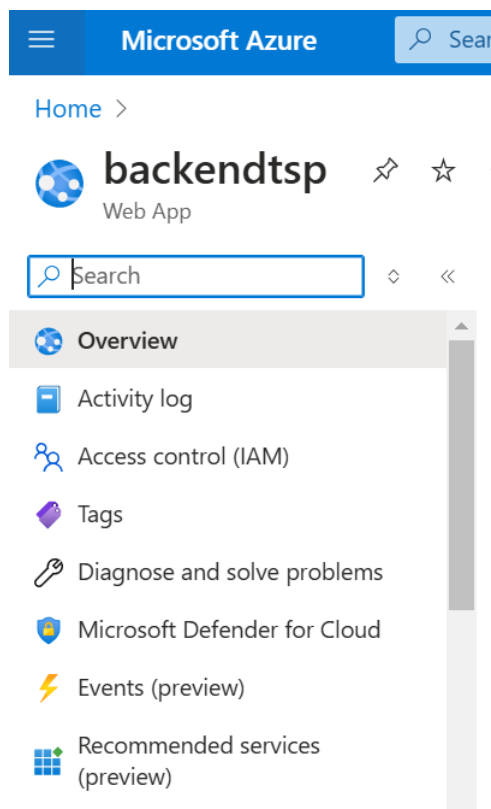
Despliegue:

Para el despliegue de la aplicación utilizamos la herramienta de Azure llamada App Service. Por el cual no necesitamos preocuparnos por la infraestructura esto se conoce como PaaS (Platform as a Service). Es un servicio basado en HTTP para hospedar aplicaciones web, API REST y back-ends para dispositivos móviles. La ejecución y escalamiento de las aplicaciones en entornos de Windows y Linux (Azure, 2024).

Como prerrequisito para el despliegue es tener un App Service en Azure listo para poder hacer el despliegue de la funcionalidad de solicitud de microcréditos

Figura 41

App Service que se utiliza para la aplicación backend.



Nota: Fuente de elaboración propia utilizando Azure.

Una vez implementada el App Service, toca realizar la configuración con el repositorio Bitbucket para poder realizar el despliegue desde esa herramienta, para ello existen los Pipelines.

Se genera el artefacto y mediante ftp se sube y se ejecuta en este servicio. Esto se puede hacer de manera manual, pero resta mucho tiempo por ello se automatiza con los pipelines. Cada ambiente tiene una rama específica para despliegue.

- **Desarrollo:** feature/develop
- **Certificación:** release
- **Producción:** master

Luego de tener la aprobación de los stakeholders sobre la funcionalidad y una fecha pactada para liberar la funcionalidad al ambiente de producción. Se hacen todas las gestiones correspondientes para llevarlo a cabo. QA apoya con las ratificaciones para que se de conformidad que la funcionalidad está operando de manera correcta.

Figura 42

Pipeline para el despliegue de la aplicación en un App Service.

Bitbucket Pipelines Pipe: Azure Web Apps Deploy

Deploys an application to [Azure Web Apps](#). Azure Web Apps enables you to build and host web applications in the programming language of your choice without managing infrastructure. It offers auto-scaling and high availability.

Note: This pipe was forked from <https://bitbucket.org/microsoft/azure-web-apps-deploy> for future maintenance purposes.

YAML Definition

Add the following snippet to the script section of your `bitbucket-pipelines.yml` file:

```
script:
- pipe: atlassian/azure-web-apps-deploy:1.2.2
  variables:
    AZURE_APP_ID: $AZURE_APP_ID
    AZURE_PASSWORD: $AZURE_PASSWORD
    AZURE_TENANT_ID: $AZURE_TENANT_ID
    AZURE_RESOURCE_GROUP: '<string>'
    AZURE_APP_NAME: '<string>'
    ZIP_FILE: '<string>'
    # AZURE_SUBSCRIPTION: '<string>' # Optional.
    # SLOT: '<string>' # Optional.
    # AZURE_CLOUD_ENVIRONMENT: '<string>' # Optional
    # DEBUG: '<boolean>' # Optional.
```

Nota: Fuente elaboración propia

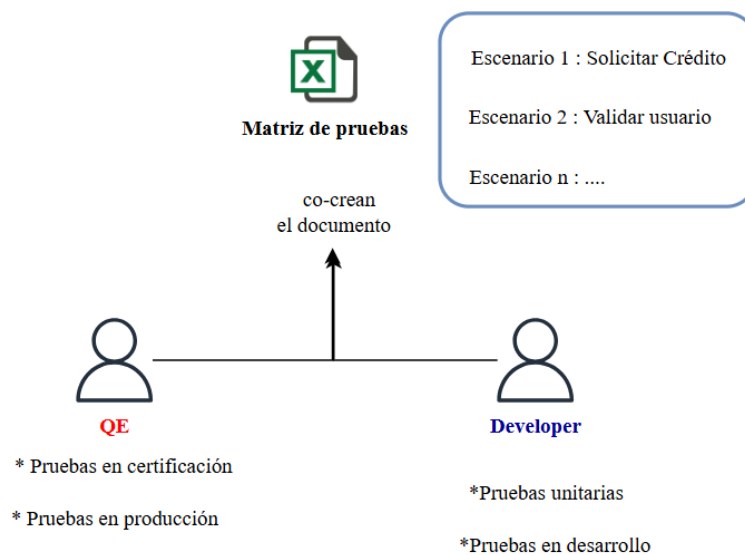
Pruebas

En desarrollo: Como parte del aseguramiento de la calidad, se utiliza un concepto llamado desplazamiento hacia la izquierda o shift-left en el cual se considera que para poder identificar lo más antes posible cualquier error, se realicen pruebas en todo el ciclo de vida del desarrollo de software.

Por eso aparte de las reuniones de refinamiento, hay otras más técnicas entre QE y desarrollador donde se definen una matriz de pruebas considerando los diferentes escenarios a validar. Este documento se lo lleva el desarrollador para incluirlo en sus pruebas unitarias y pruebas funcionales en el ambiente de desarrollo.

Figura 43

Reunión técnica para crear el documento de matriz de pruebas.



Nota: Fuente elaboración propia

En certificación

En esta etapa luego de cerrada la parte de desarrollo con la funcionalidad solicitada el QE, realiza las pruebas funcionales y pruebas integración utilizando la matriz de pruebas. Y da la conformidad en caso cumpla con todos los criterios de aceptación.

También realiza las pruebas de carga y de rendimiento antes de dar su conformidad para el pase a producción.

En producción

Una vez desplegada la aplicación en el ambiente de producción, QE nos ayuda a realizar las pruebas de ratificación para asegurar que la funcionalidad se esté ejecutando correctamente y en caso aplique también las pruebas de regresión, con todas las evidencias obtenidas nos brinda la conformidad.

Si hubiera algún error en las pruebas de ratificación; el QE reporta y el desarrollador tiene que realizar un análisis para encontrar el origen del problema y poder solucionarlo en el menor tiempo posible. En caso el error escapara del alcance del equipo; se procede a realizar una reversión, junto con una prueba de regresión.

Figura 44

Correo de conformidad de pase a producción

Estimados

Se terminó de realizar las pruebas de ratificación en el ambiente de producción con estado **CONFORME**

Saludos

Analista de Calidad

Excelente, gracias.

Muchas gracias.

Conforme, gracias.

Nota: Fuente elaboración propia

Review: En este evento se presenta a los stakeholders una funcionalidad de la aplicación validada por QE, para informar sobre los avances, retos y/o bloqueantes que se tuvo en el sprint y recibir feedback. De acuerdo con ello se realiza los ajustes y se brinda una fecha tentativa de salida a producción.

Para la presentación se utiliza diapositivas con los diferentes puntos como:

Figura 45

Presentación de lo trabajado en el sprint en la review.

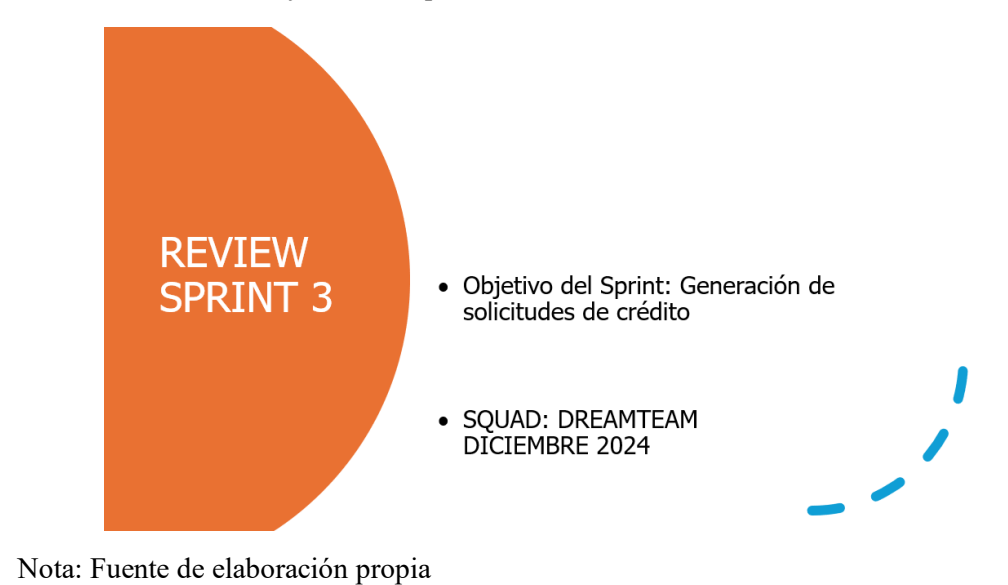


Figura 46

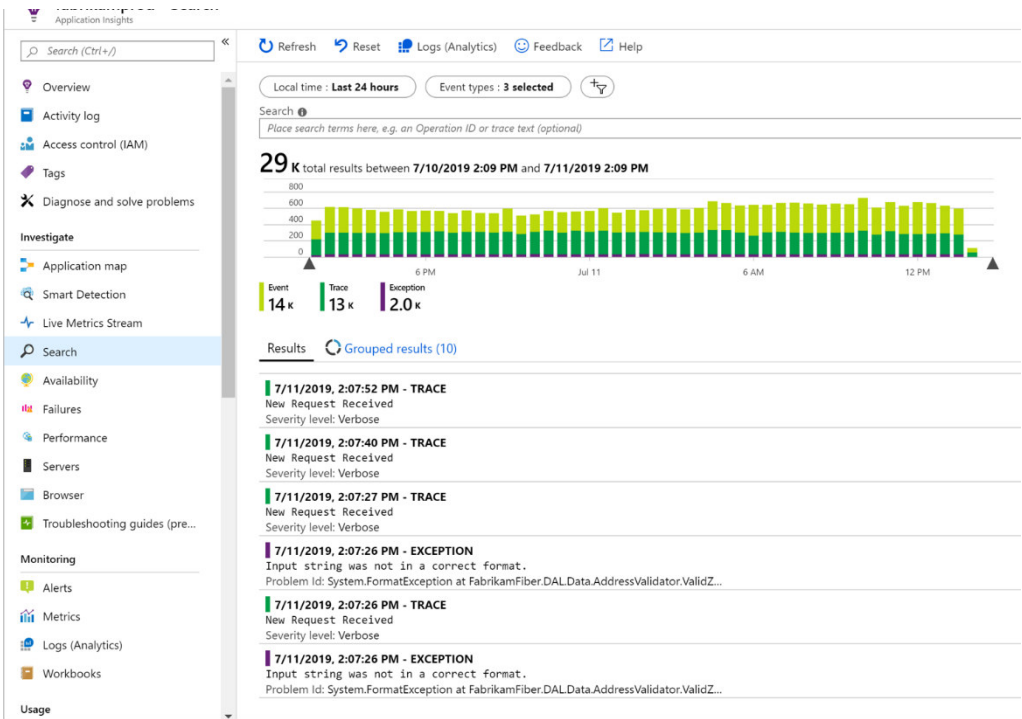
Agenda de la Review.



Monitoreo y Mantenimiento: Finalmente cuando la funcionalidad está desplegada en producción hay que hacerle seguimiento para ver que todo este marchando bien, y en caso salga algún bug o haya alguna otra necesidad se tendrá que realizar un ajuste.

Figura 47

Logs visualizados en la herramienta Azure Application Insights.



Nota: Fuente Portal de Azure

3.3. Evaluación

3.3.1. Evaluación económica / Evaluación costo-beneficio

Se realiza los análisis estimados para valorar los costos y beneficios asociados al proyecto

Tabla 11

Recursos requeridos para la implementación del sistema.

<i>Cargo</i>	<i>Porcentaje de dedicación</i>	<i>Tarifa Mensual(S/)</i>	<i>Costo Total Mensual</i>	<i>Costo Total Anual</i>
<i>Product Owner</i>	<i>50%</i>	<i>S/12,000</i>	<i>S/6,000</i>	<i>S/72,000</i>
<i>Scrum Máster</i>	<i>30%</i>	<i>S/8,000</i>	<i>S/2,400</i>	<i>S/28,800</i>
<i>Desarrollador backend 1</i>	<i>100%</i>	<i>S/5,000</i>	<i>S/5,000</i>	<i>S/60,000</i>
<i>Desarrollador backend 2</i>	<i>100%</i>	<i>S/5,000</i>	<i>S/5,000</i>	<i>S/60,000</i>
<i>Desarrollador frontend</i>	<i>100%</i>	<i>S/5,000</i>	<i>S/5,000</i>	<i>S/60,000</i>
<i>Analista de calidad 1</i>	<i>100%</i>	<i>S/6,000</i>	<i>S/6,000</i>	<i>S/72,000</i>
<i>Analista de calidad 2</i>	<i>30%</i>	<i>S/6,000</i>	<i>S/1,800</i>	<i>S/21,600</i>
<i>Desarrollador Core</i>	<i>50%</i>	<i>S/6,000</i>	<i>S/3,000</i>	<i>S/36,000</i>
<i>Total</i>				<i>S/ 410,400</i>

Nota: Los valores presentados son estimaciones basadas en datos recopilados de diversas fuentes, incluyendo Glassdoor, y han sido ajustados según criterios propios.

Tabla 12

Infraestructura y servicios para la implementación del sistema.

<i>Recurso</i>	<i>Tipo de Recurso</i>	<i>Unidad de medida</i>	<i>Cantidad Estimada</i>	<i>Costo Estimado Mensual (\$)</i>	<i>Costo Estimado Mensual (S/)</i>	<i>Costo Total Estimado Anual (S/)</i>
<i>Azure Blob Storage</i>	Almacenamiento	GB	250	\$146.00	S/562.10	S/6,745.20
<i>Azure function</i>	Computación	Ejecuciones	15	\$0.00	S/0.00	S/0.00
<i>Azure Logic Apps</i>	Integración	Ejecuciones	15	\$0.01	S/0.039	S/0.47
<i>Azure SQL Database</i>	Base de datos	DTUs	20	\$29.43	S/113.306	S/1,359.67
<i>Azure App Services</i>	Aplicaciones Web	Instancias	3	\$438.00	S/1,686.30	S/20,235.60
Total						S/ 28,340.94

Nota: Los costos mensuales estimados han sido calculados utilizando la Calculadora de Precios de Azure y son aproximados. Se han ajustado según las necesidades específicas del proyecto. Para la conversión de costos se ha utilizado un precio de dólar promedio de S/. 3.85.

Tabla 13

Herramientas para gestión de trabajo.

<i>Herramienta</i>	<i>Unidad de Medida</i>	<i>Cantidad</i>	<i>Costo Unitario Mensual (\$)</i>	<i>Costo Unitario Mensual (S/)</i>	<i>Costo Total Mensual (S/)</i>	<i>Costo Total Anual (S/)</i>
<i>Jira</i>	Licencia	10	\$0.00	S/0.00	S/0.00	S/0.00
<i>Bitbucket</i>	Licencia	5	\$0.00	S/0.00	S/0.00	S/0.00
<i>Confluence</i>	Licencia	10	\$0.00	S/0.00	S/0.00	S/0.00
Total					S/0.00	S/0.00

Nota: Los costos mensuales estimados se han obtenido de la página web de Atlassian y son aproximados. Estos valores pueden variar según el plan seleccionado y el número de usuarios.

Para la implementación de este sistema backend podríamos indicar que costo total anual es:

$$\text{CostoRecursos} = CR$$

$$\text{CostoInfraestructuraYservicios} = CIYS$$

$$\text{CostoHerramientasGestión} = CHG$$

$$\text{Costo implementación total anual} = CR + CIYS + CHG$$

$$\text{Costo implementación total anual} = S/410,400.00 + S/28,340.94 + S/0.00$$

$$\text{Costo implementación total anual} = S/ 438,740.94$$

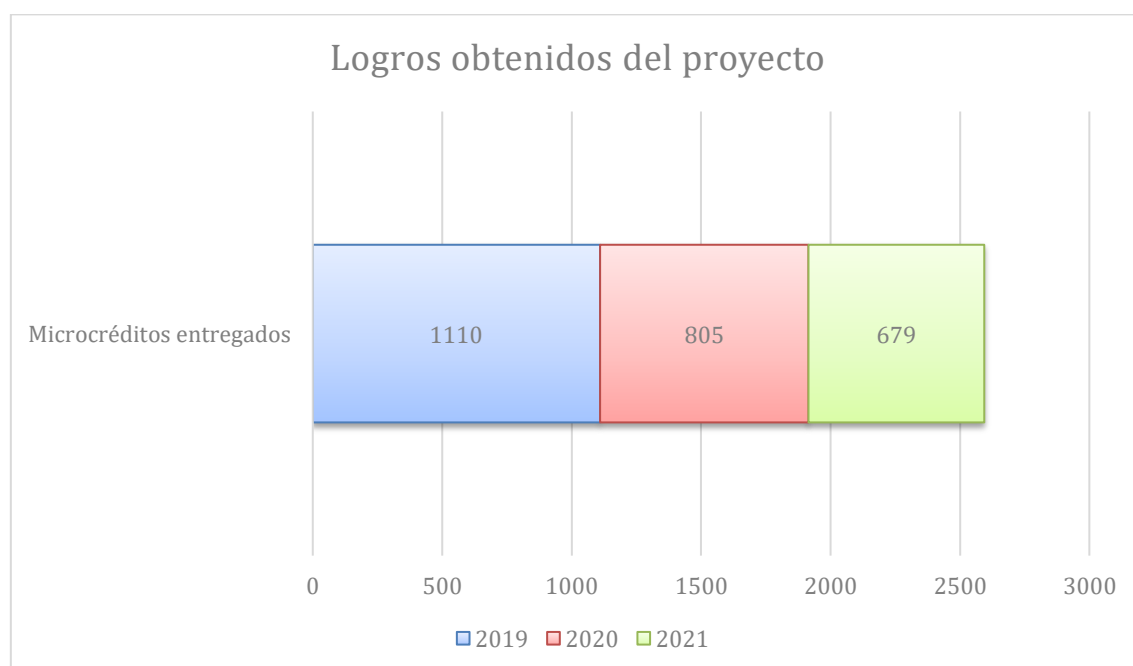
Y que este gasto va a ser un valor constante aproximado en los siguientes años como la figura de mantenimiento.

Beneficios

Ahora se mostrará los logros obtenidos en los 3 períodos de funcionamiento del sistema para revisar si la inversión fue rentable o no.

Tabla 14

Logros obtenidos por la plataforma desde mayo 2019.



Nota: Los datos han sido alterados, con el fin de proteger la confidencialidad de la entidad financiera. Los valores son representativos y no reflejan cifras exactas.

- En el primer periodo fue donde se entregó la mayor cantidad de microcréditos 1110.
- En el segundo periodo dada la coyuntura del COVID 19 se redujo en un 27,5% aproximadamente, con unos 805 microcréditos.
- Y finalmente en el último periodo de análisis se redujo en 15,6% respecto del segundo con unos 679 microcréditos.

En estos 3 periodos de funcionamiento de la plataforma, se colocaron 2594 microcréditos, en base a esa variable realizaremos el ejercicio para obtener el flujo de caja.

Antes de ello, los datos de entrada son aproximados, la tasa de oportunidad vamos a definirlo en base a la TEA, el cual a su vez se está considerando en comparación a otras entidades financieras.

Tasa de oportunidad: 30%

Tabla 15

Cálculo de monto aproximado a pagar por un microcrédito.

Monto microcrédito:	S/700
TEM:	3.51%
TCEA:	64.45%
Plazo	6 meses
Total, a pagar:	Aproximadamente S/831.46

Nota: Fuente elaboración propia, cálculo realizado con simulador de créditos de la entidad financiera, se considera seguro de desgravamen y micro seguros.

Beneficio/Costo

Tenemos los siguientes flujos de caja en beneficios y costos.

Para el cálculo de los beneficios esperados se considera:

$$\# \text{ de colocaciones por periodo} \times \text{monto promedio de pago por un préstamo}$$

Tabla 16

Beneficios esperados del proyecto en los 3 años.

Año	Beneficios Esperados
2019	$1,110 \times 831.46 = \text{S/ } 922,920.60$
2020	$805 \times 831.46 = \text{S/ } 669,325.30$
2021	$679 \times 831.46 = \text{S/ } 564,561.34$

Nota: Fuente elaboración propia.

Para el cálculo de costos esperados se considera el costo total de implementación por año, con si fuera la figura de mantenimiento.

Tabla 17

Costos esperados del proyecto en los 3 años.

Año	Costos Esperados
2019	S/ 438,740.94
2020	S/ 438,740.94
2021	S/ 438,740.94

Nota: Fuente elaboración propia.

Tasa de oportunidad: 30%

Se calcula el **Valor Presente de Beneficios (VPB)** y **Valor Presente de Costos (VPC)**

Formular para obtener el valor presente de beneficios:

$$VPB = \sum \left(\frac{B_t}{(1+r)^t} \right)$$

Dónde:

- **VPB** es el valor presente de los beneficios.
- **B_t** es el beneficio esperado en el período t.
- **r** es la tasa de descuento o tasa de interés.
- **t** es el número del período.

Tabla 18

Beneficios y costos esperados en los 3 periodos.

Año	Periodo	Beneficios Esperados	Costos Esperados
2019	1	922,920.60	438,740.94
2020	2	669,325.30	438,740.94
2021	3	564,561.34	438,740.94

Nota: Fuente elaboración propia.

$$\text{Valor Presente de Beneficios (VPB)} = \frac{922,920.60}{(1+0.3)^1} + \frac{669,325.30}{(1+0.3)^2} + \frac{564,561.34}{(1+0.3)^3}$$

$$VPB = \frac{922,920.60}{1.3} + \frac{669,325.30}{1.69} + \frac{564,561.34}{2.197}$$

$$VPB = S/ 1,362,958.60$$

Fórmula para obtener el valor presente de costos:

$$VPC = \sum \left(\frac{C_t}{(1+r)^t} \right)$$

Donde:

- **VPC** es el valor presente de los costos.
- **C_t** es el costo esperado en el período t.
- **r** es la tasa de descuento o tasa de interés.
- **t** es el número del período.

$$\text{Valor Presente de Costos (VPC)} = \frac{438,740.94}{(1+0.3)^1} + \frac{438,740.94}{(1+0.3)^2} + \frac{438,740.94}{(1+0.3)^3}$$

$$VPC = \frac{438,740.94}{1.3} + \frac{438,740.94}{1.69} + \frac{438,740.94}{2.197}$$

$$VPC = S/ 796,803.07$$

En resumen, podemos indicar que:

Valor Presente de Beneficios (VPB): S/ 1,362,958.60

Valor Presente de Costos (VPC): S/ 796,803.07

Ahora teniendo los 2 valores de beneficios y costos, se puede calcular el índice de **Beneficio/Costo (B/C)**, el cual nos indicará si el proyecto es viable o no, utilizando esta relación de beneficio y costo.

$$B/C = \frac{VALOR PRESENTE DE LOS BENEFICIOS(VPB)}{VALOR PRESENTE DE LOS COSTOS(VPC)}$$

$$B/C = \frac{1,362,958.60}{796,803.07} = 1.71 > 1$$

Como el índice es mayor que 1 nos sugiere que el proyecto es rentable y puede ser considerado para su implementación.

3.3.2. Interpretación del VAN y del TIR

Tanto el **Valor Actual Neto (VAN)** como la **Tasa Interna de Retorno (TIR)** nos ayudarán a evaluar si es rentable la inversión de este proyecto.

Si el VAN es mayor que cero; significa que el proyecto es rentable; lo invertido generará más valor que su costo. Por otro lado; si la TIR es mayor que la tasa de interés también es considerado rentable.

Ahora realizamos el cálculo del valor del VAN. Dado que no se conoce el valor exacto de la inversión inicial se utilizará el enfoque de análisis de escenarios, que permitirá conocer las fluctuaciones de acuerdo con los supuestos valores de la inversión.

Definiremos los 3 escenarios pesimista, conservador y optimista; para el segundo vamos a utilizar el valor del costo anual en la implementación que vimos en la tabla costo consideraremos un valor base y a partir de ello supondremos los valores de los escenarios pesimista y optimista.

Tabla 19

Enfoque de escenario para la inversión inicial del proyecto

Inversión inicial	Valor
Escenario pesimista	S/ 1,000,000.00
Escenario conservador	S/ 431,995.74
Escenario optimista	S/ 100,000.00

Nota: Fuente elaboración propia

Tabla 20

Cálculo de monto aproximado a pagar por un microcrédito

Año	Ingreso Anual	Mantenimiento	Flujo de caja anual neto
2019	S/ 922,920.60	S/ 431,995.74	S/ 490,924.86
2020	S/ 669,325.30	S/ 431,995.74	S/ 237,329.56
2021	S/ 564,561.34	S/ 431,995.74	S/ 132,565.60

Nota: Fuente elaboración propia.

$$VAN = \sum \left(\frac{\text{Flujo de caja}_t}{(1+r)^t} \right) - \text{Inversión inicial}$$

Donde:

- **Flujo de caja:** son los ingresos que se esperan obtener en cada periodo.
- **r:** tasa de interés.
- **t:** es el tiempo o periodo.
- **Inversión inicial:** es el costo inicial del proyecto.

Escenario pesimista:

Inversión inicial: S/ **1,000,000.00**

Costo de mantenimiento anual: S/ **431,995.74**

Tasa de oportunidad o descuento: **30%**

Flujo de caja anual neto: **Valores obtenidos de la tabla 22**

$$VAN_{\text{Pesimista}} = \frac{922920.60}{(1+0.3)^1} + \frac{669325.30}{(1+0.3)^2} + \frac{564561.34}{(1+0.3)^3} - 1,000,000$$

$$VAN_{\text{Pesimista}} = \frac{922920.6}{1.3} + \frac{669325.30}{1.69} + \frac{564561.34}{2.197} - 1,000,000$$

$$VAN_{\text{Pesimista}} = \text{S/ } -421,594.43$$

Esto resultado nos indica que, descontados todos los flujos futuros, el proyecto generará una pérdida de **S/421,594.43 por lo cual el proyecto no es rentable.**

Escenario conservador:

Inversión inicial: S/ **431,995.74**

Costo de mantenimiento anual: S/**431,995.74**

Tasa de oportunidad o descuento: **30%**

Flujo de caja anual neto: **Valores obtenidos de la tabla 22**

$$VAN_{\text{Conservador}} = \frac{922920.60}{(1+0.3)^1} + \frac{669325.30}{(1+0.3)^2} + \frac{564561.34}{(1+0.3)^3} - 431,995.74$$

$$VAN_{\text{Conservador}} = \frac{922920.6}{1.3} + \frac{669325.30}{1.69} + \frac{564561.34}{2.197} - 431,995.74$$

$$VAN_{\text{Conservador}} = \text{S/ } 146,409.83$$

Esto resultado nos indica que, descontados todos los flujos futuros, el proyecto generará **S/146,409.83** más de lo que cuesta la inversión inicial (S/ **431,995.74**), tomando en cuenta la tasa de oportunidad. **Por lo cual el proyecto es rentable.**

Escenario Optimista:

Inversión inicial: S/ **100,000.00**

Costo de mantenimiento anual: S/ **431,995.74**

Tasa de oportunidad o descuento: **30%**

Flujo de caja anual neto: **Valores obtenidos de la tabla 22**

$$VAN_{\text{Optimista}} = \frac{922920.60}{(1+0.3)^1} + \frac{669325.30}{(1+0.3)^2} + \frac{564561.34}{(1+0.3)^3} - 100,000.00$$

$$VAN_{\text{Optimista}} = \frac{922920.6}{1.3} + \frac{669325.30}{1.69} + \frac{564561.34}{2.197} - 100,000.00$$

$$VAN_{\text{Optimista}} = \text{S/ } 478,405.57$$

Esto resultado nos indica que, descontados todos los flujos futuros, el proyecto generará **S/478,405.57** más de lo que cuesta la inversión inicial (**S/ 100,000.00**), tomando en cuenta la tasa de oportunidad. **Por lo cual el proyecto es rentable.**

TIR

$$0 = \sum \left(\frac{\text{Flujo de caja}_t}{(1 + TIR)^t} \right) - I_0$$

Donde:

- Flujo de caja_t: Flujo de caja neto esperado en el periodo t.
- TIR: es la tasa interna de retorno que estamos buscando.
- t: es el periodo.
- I₀ es la inversión inicial.

De la misma forma que el VAN vamos a considerar los 3 escenarios para poder hacer el cálculo.

Escenario Pesimista

Tabla 21

Flujo de caja anual neto para el escenario pesimista.

Año	Flujo de caja anual neto
0	S/ -1,000,000.00
2019	S/ 490,924.86
2020	S/ 237,329.56
2021	S/ 132,565.60

Nota: Fuente elaboración propia

TIR = -9%

Tasa de oportunidad: 30%

Escenario Conservador

Tabla 22

Flujo de caja anual neto para el escenario conservador.

Año	Flujo de caja anual neto
0	S/ -431,995.74
2019	S/ 490,924.86
2020	S/ 237,329.56
2021	S/ 132,565.60

Nota: Fuente elaboración propia.

TIR = 60%

Tasa de oportunidad: 30%

Escenario Optimista

Tabla 23

Flujo de caja anual neto para el escenario optimista.

Año	Flujo de caja anual neto
0	S/ -100,000.00
2019	S/ 490,924.86
2020	S/ 237,329.56
2021	S/ 132,565.60

Nota: Fuente elaboración propia.

TIR = 439%

Tasa de oportunidad: 30%

Tabla 24

Cuadro comparativo evaluar la inversión

Escenario	Inversión Inicial	VAN	TIR
Pesimista	S/ 1,000,000.00	S/ -421,594.43	-9%
Conservador	S/ 431,995.74	S/ 146,409.83	60%
Optimista	S/ 100,000.00	S/ 478,405.57	439%

Nota: Fuente elaboración propia

De este análisis del VAN y la TIR, podríamos concluir

- El escenario pesimista; dado que la **VAN** es negativa (**S/ -421,594.43**) no es rentable y se estaría perdiendo dinero, de igual forma la **TIR (-9%)** es negativa por lo cual definitivamente no es rentable.
- El escenario conservador es favorable debido a que el **VAN** es positivo (**S/ 146,409.83**) y la **TIR (60%)** es alta por ello se puede considerar como una buena inversión.
- El escenario optimista tiene buen potencial con un **VAN (S/ 478,405.57)**, y por el valor alto de la **TIR (439%)** se muy atractivo debido a que generaría una buena rentabilidad

CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El autor de este trabajo desempeñó un papel fundamental en el proyecto de implementación del sistema backend para la solicitud de microcréditos. A lo largo del proceso hubo desafíos que, al analizarlos en retrospectiva, ofrecieron oportunidades de mejora:

- Como reflexión crítica de esta experiencia, se destaca que para implementar algo realmente beneficioso para el usuario, es muy importante comprender sus necesidades desde su punto de vista.
- Aprendí a gestionar las solicitudes de requerimientos con los equipos externos, porque ellos trabajaban a otro ritmo, con sus propias priorizaciones y conseguir atención de algún requerimiento se volvió complicado y como consecuencia a veces no podíamos cumplir nuestros objetivos del sprint por esa dependencia. La experiencia me enseñó que, si hay alguna solicitud hacia un equipo externo, desde el inicio involucrar a un jefe directo de nuestro equipo para que esté al tanto y en caso no haya ninguna respuesta; él podría intervenir y asegurarse de que se le dé seguimiento a la solicitud.
- La comunicación dentro del equipo es clave para evitar malentendidos y retrasos, ya que permite estar alineados respecto a la visión del proyecto, asimismo fomenta la colaboración, identificación y resolución de problemas. Además, no todo es comunicación técnica, a veces es muy necesario explicar las ideas de manera más sencilla y así todos puedan entender cada detalle.
- Si se desea liberar funcionalidad rápidamente; la deuda técnica no debería ser negociable ya que compromete la calidad del producto; esta deuda debe ser gestionada, resuelta pero no ignorada ni postergada.
- Otra lección que considero importante en este proyecto es la documentación del proyecto; todo lo que se pueda plasmar, no necesariamente muy extensa, pero mencione lo importante de esta manera permita al equipo esto permitirá

al equipo ubicarse y absolver cualquier duda, del proyecto y sus funcionalidades, asimismo reduce la curva de aprendizaje para los nuevos integrantes.

- El feedback de los stakeholders; es muy importante porque nos brinda la visión de negocio; con ello podemos tener mejoras significativas en el producto.
- Todas las tareas repetitivas en lo posible se deben automatizar, esto ayuda a optimizar tiempo y ser más eficientes en todo el proceso de desarrollo. Trabajarlo de forma manual, nos ha llevado muchas veces al error y la demora, por lo que se decidió priorizar la automatización.
- La elección del stack tecnológico es muy necesario definirlo al inicio del proyecto para evitar retrabajos que puedan generar demoras en los entregables.
- Utilizar herramientas en la nube a través de las capas gratuitas es muy beneficioso en los inicios de cualquier proyecto, porque puedes experimentar soluciones, con ciertos límites, pero sin afectar la parte financiera.
- Lo aprendido en este piloto se aprovechó en los otros proyectos similares dentro de la misma entidad, permitiendo resolver problemas recurrentes que se habían identificado y resuelto con anterioridad.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

En el capítulo se detallan los puntos más relevantes de la consecución de los objetivos.

5.1. Conclusiones

- Adoptar y utilizar Scrum permitió al equipo entregar valor de manera continua en cada iteración, mejorar la comunicación entre todos los miembros y tener una mejor visibilidad del progreso del proyecto en todo momento.
- La implementación de la API facilitó la integración con otras APIs internas y externas, así como la gestión con la base de datos. Además, ayudó a resaltar la importancia de planificar la infraestructura tecnológica anticipando las necesidades futuras a medida que crecen los requisitos de consumo.
- Las tareas repetitivas, como despliegues en los diferentes ambientes, se vio beneficiado por la utilización de herramientas automatizadas que ayudaron a ser más eficientes y reducir el error humano.
- El enfoque preventivo en las pruebas funcionales durante las primeras etapas del ciclo de desarrollo permitió al equipo identificar fallos de manera temprana y reducir el tiempo necesario para corregirlos, lo que aseguró la calidad del sistema.
- Los costos y beneficios del proyecto se han estimado considerando el monto promedio de los microcréditos, y en base a la información de la plataforma de Glassdoor para los cálculos del costo de implementación.
- La implementación del sistema backend involucra un costo, para que esta inversión sea rentable; debemos considerar el escenario conservador o optimista con VAN es positivo y la TIR es alta por ello se puede considerar como una buena inversión.

- El lanzamiento de la plataforma a producción representó un importante impulso para la bancarización de muchos peruanos.
- Aunque no dispongo de evidencia directa de que el proyecto haya recuperado su inversión, se han realizado cálculos estimados utilizando métodos financieros estándar para proporcionar una aproximación razonable sobre la rentabilidad del proyecto

5.2. Recomendaciones

A partir de la valiosa experiencia derivada de la implementación del proyecto.

- Se puede utilizar este aprendizaje para probar nuevos modelos de negocio similar al de los socios conductores.
- El proceso de solicitud que es digital se puede alimentar otras herramientas como modelos de score crediticio.
- Se puede hacer un seguimiento de las métricas del nivel de satisfacción.
- Se debería revisar las solicitudes realizadas en el tercer periodo, para determinar cuál es la razón de las pocas colocaciones.
- Se puede utilizar herramientas de análisis de comportamiento en el frontend de la plataforma, para poder obtener información importante que se puede contrastar con lo registrado en la base de datos.

5.3. Fuentes de información

Apache Software Foundation. (n.d.). Apache Maven. <https://maven.apache.org/>

Atlassian. (n.d.). ¿Qué es Jira Software? <https://www.atlassian.com/es/software/jira/guides/getting-started/introduction#what-is-jira-software>

Atlassian. (n.d.). Acerca de Confluence Obtenido de <https://www.atlassian.com/es/software/confluence/resources/guides/get-started/overview#about-confluence>

Bitbucket. (n.d.). Introducción a Bitbucket. Obtenido de <https://bitbucket.org/product/es/guides/getting-started/overview#a-brief-overview-of-bitbucket>

Credicorp. (2024). Estudio de Índice de Inclusión Financiera 2024. Credicorp. Obtenido de <https://grupocredicorp.com/indice-inclusion-financiera/IIF/Ebook-Indice-de-Inclusion-Financiera-de-Credicorp-2024.pdf>

Hernández G. (2024). Uso de taxis por aplicativo aumenta en Perú: ¿Cuánto dinero mueve este antiguo oficio apoyado en la tecnología? Obtenido de <https://rpp.pe/economia/economia/uso-de-taxis-por-aplicativo-aumenta-en-el-peru-cuanto-dinero-mueve-este-antiguo-oficio-apoyado-en-la-tecnologia-noticia-1587251?ref=rpp>

InDrive. (2024). InDrive lanza InDrive Money, ofreciendo servicios financieros digitales accesibles, convenientes y justos. InDrive. <https://in-drive.another.co/indrive-lanza-indrive-money-ofreciendo-servicios-financieros-digitales-accesibles-convenientes-y-justos>

Instituto Peruano de Economía. (2023). *Impacto de las plataformas digitales en la economía peruana: Informe final*. Informe elaborado para Sociedad de Comercio Exterior del Perú (COMEX). <https://www.ipe.org.pe/portal/wp-content/uploads/2023/08/Estudio-Impacto-economico-de-las-plataformas-digitales.pdf>

JUnit Team. (n.d.). JUnit 5 user guide. <https://junit.org/junit5/docs/current/user-guide/>

Git. (n.d.). Git - el sistema de control de versiones distribuido. <https://git-scm.com/>

Lombok. (n.d.). Project Lombok. <https://projectlombok.org/>

López, S. P., & Urquía, G. E. (2023). Mind the Gap in Financial Inclusion! Microcredit Institutions fieldwork in Peru. <https://revistas.um.es/rcsar/article/view/432671/334701>

López, S. P., & Urquía, G. E. (2023). Mind the Gap in Financial Inclusion! Microcredit Institutions fieldwork in Peru. <https://revistas.um.es/rcsar/article/view/432671/334701>

Microsoft. (n.d.). Visual Studio Code. <https://visualstudio.microsoft.com/es/#vscode-section>

Microsoft. (2024). Overview of Azure SQL Database as a service (PaaS). <https://learn.microsoft.com/es-es/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>

Microsoft. (2024). ¿Qué es Azure Logic Apps? <https://learn.microsoft.com/es-es/azure/logic-apps/logic-apps-overview>

Microsoft. (2023). Descripción general de Azure Blob Storage. <https://learn.microsoft.com/es-es/azure/storage/blobs/storage-blobs-overview>

Microsoft. (2023). Información general sobre Azure Functions Obtenido de <https://learn.microsoft.com/es-es/azure/azure-functions/functions-overview?pivot=programming-language-java>

Microsoft. (2023). ¿Qué es Azure? Obtenido de <https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/get-started/what-is-azure>

Microsoft. (2024). Introducción a Azure App Services. <https://learn.microsoft.com/es-es/azure/app-service/overview>

Mockito. (n.d.). Mockito documentation. <https://site.mockito.org/>

Mulili, F. M. (2024). Effects of digital banking channels on customer experience within microfinance institutions in Kenya [Strathmore University]. <http://hdl.handle.net/11071/15499>

Oracle. (n.d.). Java SE 8 Documentation. <https://docs.oracle.com/javase/8/docs/>

PlantUML. (n.d.). PlantUML. <https://plantuml.com/es/>

Swagger. (n.d.). About Swagger. https://swagger.io/docs/specification/v3_0/about/

SendGrid. (n.d.). SendGrid services. <https://sendgrid.com/en-us/marketing/sendgrid-services-es>

Slack. (n.d.). ¿Qué es Slack? Obtenido de <https://slack.com/intl/es-es/help/articles/115004071768-%C2%BFQu%C3%A9-es-Slack>

Schwaber, K., & Sutherland, J. (2020). Guía de Scrum (2020). <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Scrum.org. (2021). *Guía de Kanban* [PDF]. <https://scrumorg-website-prod.s3.amazonaws.com/drupal/2021-03/2021-Kanban-Guide-Spanish-European.pdf>

SonarSource. (2020). SonarQube. <https://docs.sonarsource.com/sonarqube/9.9/#what-is-sonarcloud>

Spring. (n.d.). Spring Boot. <https://spring.io/projects/spring-boot>

Spring. (n.d.). Spring Data. <https://spring.io/projects/spring-data>

5.4. Glosario

- API: Interfaz de Programación de Aplicaciones, por sus siglas en inglés, es un conjunto de instrucciones y herramientas que permiten que diferentes servicios se comuniquen entre sí. Facilitando el intercambio de datos de manera organizada y estándar.
- Azure: Es una plataforma de servicios en la nube de Microsoft que ofrece una amplia gama de herramientas y servicios para crear, implementar y gestionar aplicaciones y servicios.
- Bitbucket: Herramienta online que permite almacenar el código desarrollado en el proyecto.
- Glassdoor: Es una plataforma en línea donde los empleados y exempleados de las empresas comparten de forma anónima sus opiniones sobre evaluaciones, cultura laboral, salarios, beneficios y entrevistas.
- Patrón AAA: Estructura que debe tener una prueba unitaria considerando las 3 A por las siglas en ingles de Arrange (Preparar), Act (Ejecutar) y Assert (Verificar)
- Pipeline: Es un archivo por lo general en formato yml que contiene pasos automatizados para ejecutar alguna tarea como despliegue, análisis de código estático.
- POC: Prueba de concepto por sus siglas en ingles proof of concept, es un experimento que tiene como objetivo validar si alguna solución es viable técnicamente, identificando limitantes en caso los hubiera.
- Shift Left: Es una práctica en el desarrollo de software donde las pruebas y revisión de calidad se realiza lo más temprano posible ya no se espera al llegar a la certificación. Con ello se ahorra tiempo y se garantiza la calidad del código.
- Socio: Es una persona cuyo servicio como conductor es prestado utilizando la aplicación de taxi aplicativo.

ANEXOS

Anexo 1: Certificaciones

Figura 48

Certificación del autor en Azure Fundamentals



Nota: Obtenido de Azure

Figura 49

Certificación del autor en Java Architect.



Nota: Obtenido de CJAVA

Anexo 2: Cálculo de Beneficios Esperados

Se realiza el cálculo del beneficio esperado considerando el monto promedio a pagar por un microcrédito y el número de colocaciones por periodo.

Figura 50

Cálculo de beneficios esperados.

	Monto de microcrédito:	S/ 700.00	
	Monto a pagar x el microcrédito:	S/ 831.46	
Periodo	Cantidad de colocaciones	Monto promedio a pagar por préstamo	Beneficio Esperado
2019	1,110	831.46	922920.60
2020	805	831.46	669325.30
2021	679	831.46	564561.34

Nota: Fuente de elaboración propia

Anexo 3: Cálculo de VPC y VPB en Excel

En el capítulo V, se realiza el cálculo del valor presente costos y valor presente beneficios. Utilizando la fórmula VNA en Excel.

Figura 51

Cálculo de VPC y VPB.

Año	Beneficios	Costos	Tasa	
1	922,920.60	438,740.94	0.30	
2	669,325.30	438,740.94		
3	564,561.34	438,740.94		
Valor Presente de Beneficios (VPB):		Valor Presente de Costos (VPC):		
'=VNA(TASA; RANGOBENEFICIOS)		'=VNA(TASA; RANGOCOSTOS)		
S/ 1,362,958.60		S/ 796,803.07		

Nota: Fuente de elaboración propia

Anexo 3: Cálculo del índice Beneficio/Costo

En el capítulo V, se calcula el índice de beneficio costo en Excel con los valores obtenidos de VPC y VPB del Anexo 2

Figura 52

Índice de beneficio/costo.

Be					
Valor Presente de Beneficios (VPB):			Valor Presente de Costos (VPC):		
S/ 1,362,958.60			S/ 796,803.07		
		Evaluación Beneficio/Costo			
		B/C(VPB/VPC)	1.71		

Nota: Fuente de elaboración propia

Anexo 4: Cálculo del VAN

En el capítulo V, se calcula el VAN, considerando los escenarios posibles pesimista, conservador y optimista.

Figura 53

Cálculo del VAN.

Inversión inicial	Valor	Tasa	Año	Beneficios	Costos	Flujo Neto
Escenario pesimista	1,000,000.00	0.3	1	922,920.60	431,995.74	490,924.86
Escenario conservador	431,995.74	0.3	2	669,325.30	431,995.74	237,329.56
Escenario optimista	100,000.00	0.3	3	564,561.34	431,995.74	132,565.60

VAN PESIMISTA	VAN CONSERVADOR	VAN OPTIMISTA
=VNA(TASA;FLUJOCAJA)-INVERSIONINICIAL	=VNA(TASA;FLUJOCAJA)-INVERSIONINICIAL	=VNA(TASA;FLUJOCAJA)-INVERSIONINICIAL
-\$/ 139,179.98	S/ 146,409.83	S/ 478,405.57

Nota: Fuente de elaboración propia

Anexo 5: Cálculo de la TIR

En el capítulo V, se calcula la TIR, considerando los escenarios posibles pesimista, conservador y optimista; utilizando la fórmula TIR de Excel.

Figura 54

Cálculo de la TIR

TIR PESIMISTA				TIR CONSERVADOR				TIR OPTIMISTA	
Año	Flujo de caja Neto			Año	Flujo de caja Neto			Año	Flujo de caja Neto
0	-1000000.00			0	-431995.74			0	-100000.00
1	490924.86			1	490924.86			1	490924.86
2	237329.56			2	237329.56			2	237329.56
3	132565.60			3	132565.60			3	132565.60
TIR PESIMISTA	-9%			TIR CONSERVADOR	60%			TIR CONSERVADOR	439%

Nota: Fuente de elaboración propia