



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 una solución tecnológica para el procesamiento en streaming de pagos masivos en una entidad financiera

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Katty Susana GUTIERREZ VILLANUEVA

ASESOR

Mg. Víctor Hugo BUSTAMANTE OLIVERA

Lima, Perú

2026



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

Gutierrez, K. (2026). *Implementación de una solución tecnológica para el procesamiento en streaming de pagos masivos en una entidad financiera*. [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	Katty Susana Gutierrez Villanueva
Tipo de documento de identidad	DNI
Número de documento de identidad	72199822
URL de ORCID	https://orcid.org/0000-0001-7766-7848
Datos de asesor	
Nombres y apellidos	Víctor Hugo Bustamante Olivera
Tipo de documento de identidad	DNI
Número de documento de identidad	25655590
URL de ORCID	https://orcid.org/0000-0001-8805-9629
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cesar Alberto Molina Neyra
Tipo de documento	DNI
Número de documento de identidad	40553679
Miembro del jurado 1	
Nombres y apellidos	John Ledgard Trujillo Trejo
Tipo de documento	DNI
Número de documento de identidad	06187585
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	2025
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 15:00 horas del día 31 de marzo del año 2026, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Molina Neyra Cesar Alberto (Presidente), Mg. Trujillo Trejo John Ledgard (Miembro) y el Mg. Bustamante Olivera Víctor Hugo (Miembro Asesor), usando la plataforma Meet (<http://meet.google.com/wcc-jfha-tms>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“IMPLEMENTACIÓN DE UNA SOLUCIÓN TECNOLÓGICA PARA EL PROCESAMIENTO EN STREAMING DE PAGOS MASIVOS EN UNA ENTIDAD FINANCIERA”**, por la Bachiller **Gutierrez Villanueva Katty Susana**; para obtener el Título Profesional de Ingeniero de Sistemas.

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

La 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 habiendo efectuado la calificación correspondiente por los miembros del Jurado, la Bachiller obtuvo la nota de **19 (DIECINUEVE)**.

A continuación el Presidente de Jurados el Mg. Molina Neyra Cesar Alberto, declara a la Bachiller **Ingeniero de Sistemas**.

Siendo las 15:55 horas, se levantó la sesión.

Presidente

Mg. Molina Neyra Cesar Alberto

Miembro

Mg. Trujillo Trejo John Ledgard

Miembro Asesor

Mg. Bustamante Olivera Víctor Hugo

CERTIFICADO DE SIMILITUD

Yo, Victor Hugo Bustamante Olivera en mi condición de asesor acreditado mediante Resolución Decanal N° 001268-2025-D-FISI/UNMSM y OFICIO N° 006-PTTSP-FISI-2025 del Trabajo de Suficiencia Profesional, cuyo título es Implementación de una solución tecnológica para el procesamiento en streaming de pagos masivos en una entidad financiera, presentado por la bachiller Gutierrez Villanueva Katty Susana para optar el Título Profesional de Ingeniera 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 9% 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 **TITULO PROFESIONAL DE INGENIERA DE SISTEMAS.**

Lima, 4 de junio de 2026



Firma del Asesor:

Nombres y Apellidos del asesor: Victor Hugo Bustamante Olivera

DNI: 25655590

FICHA CATALOGRÁFICA

**IMPLEMENTACIÓN DE UNA SOLUCIÓN TECNOLÓGICA PARA
EL PROCESAMIENTO EN STREAMING DE PAGOS MASIVOS EN
UNA ENTIDAD FINANCIERA**

AUTOR: GUTIERREZ VILLANUEVA, KATTY SUSANA

ASESOR: BUSTAMANTE OLIVERA, VÍCTOR HUGO

LIMA – PERÚ, 2025

TÍTULO PROFESIONAL: Título Profesional de Ingeniero de Sistemas

ÁREA / PROGRAMA / LÍNEA DE INVESTIGACIÓN:
**Ingenierías / Industria, Innovación e Infraestructura / Ingeniería de
Software**

**PREGRADO: Universidad Nacional Mayor de San Marcos – Facultad de
Ingeniería de Sistemas e Informática – Escuela Profesional de Ingeniería
de Sistemas**

Formato 21 x 29.7 cm

Páginas: viii, 83

DEDICATORIA

Dedico este trabajo a mis padres, Isaac y Susana, y
a mi hermano Ruben, por su apoyo incondicional y
por acompañarme siempre en mi crecimiento
personal y profesional.

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

**Implementación de una solución tecnológica para el procesamiento en streaming de pagos
masivos en una entidad financiera**

Autor: Gutierrez Villanueva, Katty Susana

Asesor: Bustamante Olivera, Víctor Hugo

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

Fecha: Enero de 2026

RESUMEN

El presente trabajo de suficiencia profesional aborda la implementación de una solución tecnológica para el procesamiento en streaming de pagos masivos de nóminas entre cuentas del mismo banco. El problema central identificado es la obsolescencia del sistema de pagos masivos vigente en el área de Banca Empresa de la entidad, el cual se ha convertido en una barrera para la innovación digital. A ello se suman la limitada disponibilidad horaria del servicio, las restricciones de escalabilidad y la baja reutilización de componentes, factores que en conjunto comprometen la competitividad estratégica de la institución. Entre los objetivos de la propuesta se encuentran elevar los niveles de servicio del producto, reducir la dependencia del mainframe bancario y habilitar un procesamiento casi en tiempo real, sentando bases sólidas de escalabilidad. El alcance se centró en la exposición de servicios backend, destinados a abastecer los distintos canales de atención, sin incorporar desarrollos de interfaz. El proyecto fue concebido y desarrollado bajo el enfoque ágil, lo que permitió una adecuada priorización del backlog, flexibilidad ante los cambios y mejoras continuas tanto en la calidad del producto como en la eficiencia del equipo de trabajo. Como resultado, se implementó una solución alineada con las necesidades de los clientes externos e internos, que ofrece mayor disponibilidad operativa, amplía la capacidad de procesamiento de pagos por planilla, proporciona una trazabilidad más clara y oportuna de las operaciones, y refuerza las validaciones de seguridad, límites transaccionales y controles de fraude. Asimismo, se logró la modularización de las funcionalidades de pago de nóminas, habilitando APIs reutilizables para su consumo desde los distintos canales de la entidad.

Palabras claves: procesamiento en streaming, pagos masivos de nóminas, sector bancario, trazabilidad de operaciones, tecnologías legacy

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

Implementation of a technological solution for real-time processing of bulk payments at a financial institution.

Author: Gutierrez Villanueva, Katty Susana

Advisor: Bustamante Olivera, Víctor Hugo

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

Date: January 2026

ABSTRACT

This professional proficiency project addresses the implementation of a technological solution for the streaming processing of mass payroll payments between accounts at the same bank. The central problem identified is the obsolescence of the current mass payment system in the entity's Corporate Banking area, which has become a barrier to digital innovation. Added to this are the limited availability of the service, scalability restrictions, and low component reuse, factors that together compromise the institution's strategic competitiveness. The objectives of the proposal include raising the product's service levels, reducing dependence on the banking mainframe, and enabling near real-time processing, laying a solid foundation for scalability. The scope focused on exposing backend services designed to supply the various service channels, without incorporating interface developments. The project was conceived and developed using an agile approach, which allowed for adequate prioritization of the backlog, flexibility in the face of changes, and continuous improvements in both product quality and team efficiency. As a result, a solution was implemented that was aligned with the needs of external and internal customers, offering greater operational availability, expanding payroll payment processing capacity, providing clearer and more timely traceability of operations, and strengthening security validations, transaction limits, and fraud controls. Likewise, the payroll payment functionalities were modularized, enabling reusable APIs for consumption from the entity's various channels.

Keywords: streaming processing, mass payroll payments, banking sector, transaction traceability, legacy technologies

Tabla de contenido

INTRODUCCIÓN.....	1
CAPÍTULO I - TRAYECTORIA PROFESIONAL.....	3
CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA.....	7
2.1. Empresa - Actividad que realiza	7
2.2. Visión	7
2.3. Misión.....	7
2.4. Organización de la empresa	8
2.5. Área, cargo y funciones desempeñadas	8
2.6. Experiencia profesional realizada en la organización	9
CAPÍTULO III - ACTIVIDADES DESARROLLADAS	11
3.1. Situación problemática	11
3.1.1. Definición del problema	12
3.2. Solución	13
3.2.1. Objetivos	13
3.2.2. Alcance	14
3.2.3. Etapas y Metodología.....	15
3.2.4. Fundamentos utilizados.....	18
3.2.5. Implementación de las áreas, procesos, sistemas y buenas prácticas ²⁹	
3.3. Evaluación	68
3.3.1. Evaluación Económica / Evaluación Costo-Beneficio	68
3.3.2. Interpretación del VAN y del TIR.....	72
CAPÍTULO IV - REFLEXIÓN CRÍTICA DE LA EXPERIENCIA.....	74
CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES.....	76
5.1. Conclusiones	76
5.2. Recomendaciones	77
5.3. Fuentes de información	78
5.4. Glosario	80
ANEXOS.....	81

ÍNDICE DE FIGURAS

Figura 1	8
Figura 2	16
Figura 3	19
Figura 4	23
Figura 5	30
Figura 6	32
Figura 7	34
Figura 8	36
Figura 9	37
Figura 10	38
Figura 11	41
Figura 12	45
Figura 13	48
Figura 14	50
Figura 15	54
Figura 16	55
Figura 17	59
Figura 18	63
Figura 19	64
Figura 20	65
Figura 21	65
Figura 22	66
Figura 23	66
Figura 24	67
Figura 25	69
Figura 26	70
Figura 27	70
Figura 28	71
Figura 29	73
Figura 30	73

ÍNDICE DE TABLAS

Tabla 1.....	3
Tabla 2.....	5
Tabla 3.....	5
Tabla 4.....	6
Tabla 5.....	29
Tabla 6.....	42
Tabla 7.....	43
Tabla 8.....	44
Tabla 9.....	47
Tabla 10.....	48
Tabla 11.....	49
Tabla 12.....	50
Tabla 13.....	51
Tabla 14.....	52
Tabla 15.....	54
Tabla 16.....	57
Tabla 17.....	58
Tabla 18.....	59
Tabla 19.....	60
Tabla 20.....	61
Tabla 21.....	62
Tabla 22.....	63

INTRODUCCIÓN

En plena era de la digitalización, con fuertes tendencias tecnológicas, como la IA generativa, analítica en tiempo real y las redes 5G, el sistema financiero atraviesa una profunda transformación. Este cambio es más notable en la banca tradicional, que viene equilibrando sus prioridades para responder a las nuevas exigencias del entorno. Según KPMG (2025), en su más reciente estudio del sector bancario, las inversiones clave de TI se proyectan a la transformación digital, modernización del core, IA Generativa, modernización de datos, ciberseguridad y modernización de pagos.

En este contexto, el presente trabajo aborda una de las problemáticas relacionadas a la transformación digital y la modernización de pagos dentro de la entidad bancaria en estudio, la obsolescencia del sistema de pagos masivos de la Banca Empresa. Esta situación se refleja en la afectación de los niveles de servicio del producto, ausencia de un servicio de trazabilidad y una limitada capacidad de escalamiento, impactando directamente en la satisfacción de las necesidades de los clientes y en la competitividad de la entidad frente al resto del sector.

El objetivo principal del proyecto es implementar una solución tecnológica para el procesamiento en streaming de pagos masivos de nóminas entre cuentas del mismo banco, dejando atrás el enfoque monolítico de los sistemas legacy. Con ello se busca lograr la ejecución en near-real-time de procesos, disponer de trazabilidad y monitorización E2E de operaciones e incrementar la escalabilidad del producto mediante la modularización de sus funcionalidades. Asimismo, el proyecto se enfoca en la exposición de servicios backend, con el fin de abastecer a los distintos canales de cara al cliente.

La ideación y el refinamiento de la solución se realizaron siguiendo la metodología Lean Inception, mientras que la etapa de desarrollo se llevó a cabo bajo el marco de trabajo Scrum, lo que permitió realizar entregas incrementales con aporte de valor. El área de Banca Empresa asumió el rol de Product Owner, definiendo la priorización del backlog y los criterios de aceptación, mientras que el equipo Scrum ejecutó la implementación con el seguimiento correspondiente a través de las ceremonias ágiles. De esta manera, se logró cumplir con las estimaciones planteadas y mantener una adecuada alineación con las necesidades de negocio, las cuales fueron refinadas de forma

continua sprint a sprint.

Esta implementación ha mejorado la experiencia del cliente, ofreciéndole mayor disponibilidad para la realización de operaciones. Asimismo, incrementó la confianza de los usuarios al brindar información transparente y actualizada sobre los estados de los pagos, y al fortalecer las validaciones de seguridad, los límites operacionales y los lineamientos antifraude. Con ello, se marca un punto de partida para la mejora progresiva de los demás servicios de pagos masivos de Banca Empresa y para el escalamiento del producto a todos sus canales digitales, incluidos los modelos de Open Banking.

A continuación, se detalla la estructura del trabajo de suficiencia profesional:

En el CAPÍTULO I - TRAYECTORIA PROFESIONAL, se presenta un resumen de la experiencia laboral, formación académica, formación complementaria, así como estudios de idioma cursados por la autora.

En el CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA, se proporciona información general de la entidad financiera en estudio; además, se describen las funciones desempeñadas por la autora en el cargo ocupado y la experiencia profesional adquirida en su paso por la institución.

En el CAPÍTULO III - ACTIVIDADES DESARROLLADAS, se define la situación problemática y la solución propuesta, se detallan los aspectos principales de la implementación realizada y se presenta la evaluación de los resultados obtenidos.

En el CAPÍTULO IV - REFLEXIÓN CRÍTICA DE LA EXPERIENCIA, se brinda una reflexión crítica en base a la experiencia, resaltando los aprendizajes clave y los desafíos enfrentados.

Finalmente, en el CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES, la autora formula las conclusiones en función a los objetivos planteados y se proponen recomendaciones para futuros desarrollos.

CAPÍTULO I - TRAYECTORIA PROFESIONAL

La autora del presente trabajo de suficiencia profesional cuenta con un grado académico de Bachiller en Ingeniería de Sistemas, con más de 5 años de experiencia en proyectos de empresas de diferentes rubros. Especializada en el análisis, diseño, desarrollo y mantenimiento de sistemas de información.

Actualmente desarrolla sus labores en una entidad financiera liderando proyectos multidisciplinarios enfocados en la automatización de procesos de la banca empresa.

Anteriormente la autora trabajó desempeñando diversas funciones para instituciones de diferentes sectores. Los detalles se describen en la Tabla 1:

Tabla 1

Experiencia Laboral

Empresa/Periodo	Descripción de actividades realizadas
Entidad Bancaria (Octubre 2022 – Actualidad)	Cargo: Software Solutions Development Associate I Actividades: <i>Proyecto: Release 4.0 Enterprise Payments</i> - Análisis, refinamiento y priorización de entregables. - Coordinar, dirigir y supervisar la construcción, integración y despliegue de la nueva versión del backend de los flujos de pagos de la banca empresa. - Asegurar el correcto funcionamiento de los flujos productivos comprometidos en la actualización del software. Cargo: Software Solutions Development Analyst II Actividades: <i>Proyecto: Procesamiento en Streaming de Pagos Masivos de Nóminas y Proveedores en Banca Empresa</i> - Implementación de servicios bancarios transaccionales con Java y Spring Framework. - Implementación de jobs batch para el procesamiento en lotes con Java y Spring Batch. - Creación de eventos para el procesamiento de flujos de datos en tiempo real mediante Data Bus Streams. - Implementación de servicios de transferencia de datos desde múltiples fuentes en la nube. - Implementación de componentes web frontend con Lit. - Creación y configuración de buckets de

	almacenamiento en la nube para la gestión de objetos lógicos. - Programación y administración de mallas de ejecución de jobs en Dimensions CM y BMC Control-M. - Elaboración de modelos de bases de datos operacionales en Oracle Database y Elasticsearch. - Integración con herramientas DevOps: Jira, Bitbucket (GRaaS), JFrog Artifactory, Jenkins, SonarQube. - Ejecución de pruebas funcionales en entornos de validación, preproducción y producción. - Coordinación y ejecución de despliegues productivos de componentes backend. - Gestión de incidencias y requerimientos de cambios en BMC Helix. - Diagnóstico y solución de incidencias en producción.
Stefanini IT Solutions Perú SAC (Enero 2022 – Septiembre 2022)	Cargo: Analista Desarrolladora Actividades: <i>Proyecto: Alertas y Órdenes Fx (Foreign Exchange) de Tipo de Cambio en Banca Empresa – Entidad Bancaria</i> - Integración con el motor Fx para la cotización en tiempo real de divisas. - Implementación del módulo de alertas automáticas, basado en variaciones del tipo de cambio. - Implementación de la funcionalidad de simulación y liquidación de órdenes de compra y venta de divisas, tanto de forma automática como manual. - Implementación de módulos frontend con JSP y servicios backend con Java. - Despliegue de componentes frontend y backend, integrados con Dimensions CM e IBM WebSphere Application Server. - Gestión de incidencias y requerimientos de cambios en BMC Helix. <i>Otras funciones:</i> - Ejecución de pruebas funcionales y soporte en la resolución de incidencias en Banca Empresa.
Superintendencia Nacional de los Registros Públicos (Octubre 2020 – Noviembre 2021)	Cargo: Practicante Preprofesional de Desarrollo de Sistemas Actividades: <i>Proyecto: Sistema de Gestión Documental</i> - Implementación del backend con Spring Boot, utilizando microservicios y WebSocket. - Implementación del frontend basado en componentes con Angular. - Implementación del módulo de Firma Digital mediante 4identity.

- Integración con IBM Content Manager para la gestión de archivos.
- Conexión con servicios SOAP de la Plataforma Nacional de Interoperabilidad de la PCM.
- Documentación de servicios con Swagger.

Otras funciones:

- Elaboración de manuales de usuario para la Plataforma de Servicios Institucionales (PSI).
- Ejecución de pruebas de esfuerzo con JMeter en el SIGESAR (Sistema de Título Archivado).
- Automatización de pruebas funcionales con Selenium en el SPRL (Sistema de Publicidad Registral en Línea).

Nota. Elaboración Propia.

La formación académica de la autora se detalla en la Tabla 2:

Tabla 2

Formación Académica

Periodo	Formación Académica
2017 - 2021	Grado Académico de Bachiller en Ingeniería de Sistemas Escuela Académico Profesional de Ingeniería de Sistemas Facultad de Ingeniería de Sistemas e Informática Universidad Nacional Mayor de San Marcos

Nota. Elaboración Propia.

La formación complementaria de la autora se detalla en la Tabla 3:

Tabla 3

Formación Complementaria

Institución	Curso
Banking Campus	Cybersecurity Essentials
SCRUMstudy	Scrum Fundamentals Certified
CertiProf	Scrum Foundation Professional Certificate (SFPC)
Microsoft Partner	Microsoft Azure Fundamentals

Nota. Elaboración Propia.

Los estudios de idioma realizados por la autora se detallan en la Tabla 4:

Tabla 4

Idiomas

Periodo	Idioma	Nivel	Institución
2022 - 2024	Inglés	Avanzado	Instituto Británico

Nota. Elaboración Propia.

CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1. Empresa - Actividad que realiza

La entidad financiera objeto de este trabajo opera en el rubro comercial en el Perú hace varias décadas, teniendo como enfoque tanto la banca retail como la banca empresa. Además, posee unidades estratégicas de negocio claramente definidas, tales como Banca Personas, Banca PYME, Banca Corporativa y de Inversión, Tarjetas y Medios de Pago, Recursos de Mercado y Canales Digitales.

Dentro de los principales productos ofrecidos por la empresa figuran las cuentas de ahorros, cuentas corrientes, tarjetas de crédito y débito, préstamos personales e hipotecarios, así como depósitos a plazo. En el rubro corporativo destacan productos como líneas de crédito, financiamiento comercial y leasing. En servicios, abarcan asesoría financiera, plataformas digitales, seguros y gestión patrimonial.

Desde su fundación la entidad ha evolucionado constantemente, modernizando su estructura, ampliando su cobertura e impulsando su estrategia operacional, reforzando su presencia en todo el país. Asimismo, en los últimos años ha apostado por la innovación y consolidación a nivel tecnológico, mejorando sus canales digitales y optimizando los procesos transaccionales.

2.2. Visión

Ser el banco más confiable de todos los peruanos, reconocido por transformar la experiencia financiera de nuestros clientes con innovación y servicio de excelencia.

2.3. Misión

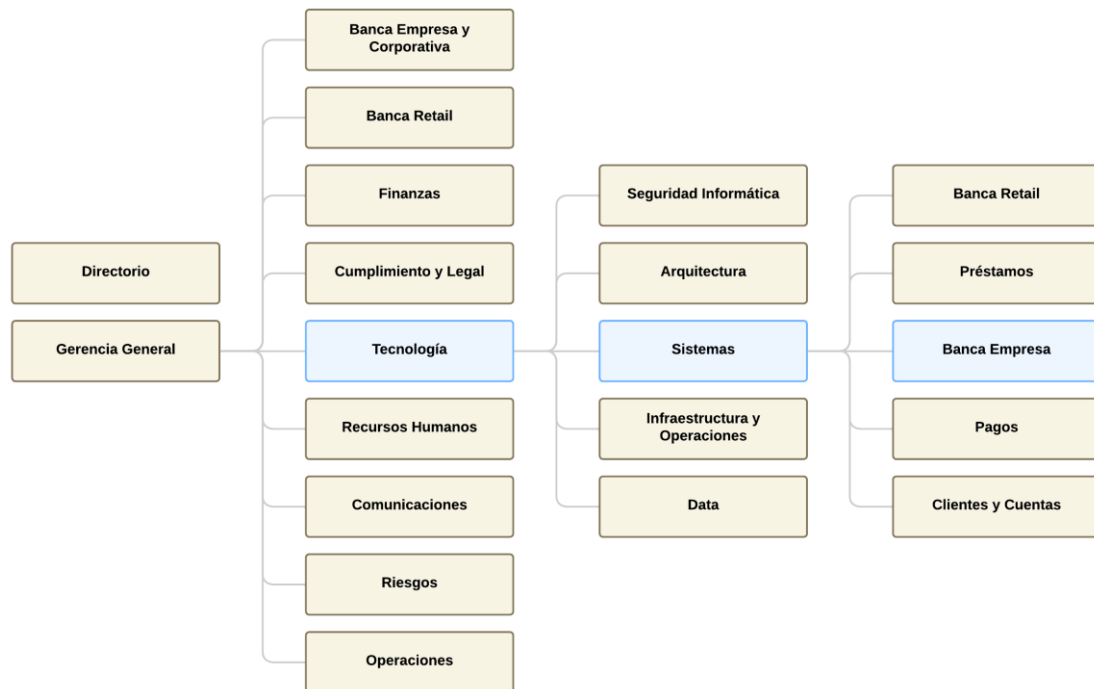
Brindar soluciones financieras innovadoras, seguras y accesibles a nuestros clientes, que les permita alcanzar sus metas personales y empresariales.

2.4. Organización de la empresa

En la siguiente figura podemos observar el organigrama de la entidad financiera donde se realizó el trabajo de suficiencia profesional.

Figura 1

Organigrama de la entidad financiera



Nota. Elaboración adaptada del organigrama de la entidad financiera.

2.5. Área, cargo y funciones desempeñadas

La autora del presente trabajo se desempeñó como Software Engineer en la subárea de Sistemas abarcando soluciones de la línea de Banca Empresa dentro de la entidad financiera.

Las principales funciones incluyeron:

- **Participar en el Refinamiento de Proyectos:** Acompañar al Scrum Master, Tech Leader y Product Owners en la definición del alcance del proyecto; estimar el esfuerzo para la atención de los MVPs y facilitar las reuniones de refinamiento del

proyecto.

- **Participar en el Onboarding de los Proyectos:** Identificar dependencias que puedan generarse al inicio del proyecto; participar en la planificación de actividades y definir los compromisos por cada MVP a desarrollar.
- **Desarrollar los Proyectos:** Elaborar la documentación funcional y técnica requerida para el desarrollo del proyecto; construir componentes de software bajo los estándares definidos; preparar, ejecutar y elaborar la documentación de los casos de pruebas de acuerdo con los requisitos funcionales y técnicos del proyecto y; brindar el soporte para el pase a producción de los componentes desarrollados.
- **Dar soporte post producción a los desarrollos desplegados:** Asegurar el correcto funcionamiento del end to end de la solución implementada y coordinar el plan de traspaso de las soluciones implementadas al equipo de Soporte Operativo.
- **Asesorar y dar soporte técnico al equipo:** Brindar asesoría al equipo en limitantes técnicas para el desarrollo del proyecto y brindar asesoría en nuevas tecnologías o en soluciones implementadas en proyectos previos.

2.6. Experiencia profesional realizada en la organización

Durante su experiencia profesional como Software Engineer en la entidad financiera, la autora del presente trabajo participó activamente en proyectos orientados a optimizar los productos y canales de Banca Empresa. Este rol le permitió consolidar sus conocimientos técnicos y, al mismo tiempo, ampliar su comprensión del funcionamiento y las particularidades de negocio del sector bancario. Asimismo, afianzó sus competencias en gestión y liderazgo, contribuyendo al trabajo coordinado de equipos multidisciplinarios.

A continuación, se detallan los principales proyectos de Banca Empresa Digital en los que la autora participó:

- Actualización y mantenimiento de la solución de Pagos Masivos e Individuales, Trazabilidad y Monitorización de Operaciones.

- Incorporación de la funcionalidad de Pagos Interbancarios Inmediatos en el módulo de Pagos Masivos de Nóminas y Proveedores.
- Incorporación de la funcionalidad de Pagos Interbancarios Por Horarios en el módulo de Pagos Masivos de Nóminas y Proveedores.
- Implementación de la nueva solución de Pagos Masivos y Trazabilidad de Operaciones de Proveedores.
- Implementación de la nueva solución de Pagos Masivos y Trazabilidad de Operaciones de Nóminas.
- Implementación de Alertas y Órdenes Fx (Foreign Exchange) de Tipo de Cambio.

CAPÍTULO III - ACTIVIDADES DESARROLLADAS

3.1. Situación problemática

Actualmente, la industria de la banca digital está creciendo a ritmos acelerados, permitiendo tanto a las personas naturales como jurídicas acceder a una amplia gama de servicios financieros. Esta evolución tecnológica también implica retos significativos para el sector bancario tales como interacción con sistemas legacy, riesgos operacionales por innovación tecnológica, cumplimientos regulatorios, así como gestión del cambio en la cultura organizacional.

En diversas regiones del mundo, muchos bancos tradicionales siguen operando con sistemas obsoletos, haciendo que la migración a soluciones modernas sea una tarea compleja y requiera una mayor inversión monetaria. Sin embargo, la falta de adaptación incrementará las ineficiencias y limitará la capacidad para responder a las cambiantes demandas de los clientes. (Eloshvili, 2025)

El estudio de Accenture (2025) sobre las principales tendencias en la industria bancaria, señala que las entidades financieras actuales se encuentran profundamente apegados a sus sistemas heredados y a sus formas de codificación, lo cual ha retrasado la migración de sus principales plataformas, desarrolladas en su mayoría en lenguajes obsoletos como COBOL, hacia soluciones en la nube. Con pocos especialistas capaces de comprender y modificar estas líneas de código, los servicios bancarios se vuelven más rígidos y frágiles, encareciendo su mantenibilidad.

En el ámbito nacional, la Superintendencia de Banca, Seguros y AFP (SBS, 2025), en su Informe de Estabilidad del Sistema Financiero expone que la transformación digital ha impulsado la eficiencia e innovación en los servicios financieros; no obstante, también ha intensificado la dependencia de infraestructuras digitales y proveedores tecnológicos externos. Como consecuencia, las amenazas cibernéticas y las fallas tecnológicas se han convertido en riesgos relevantes con potencial alcance sistémico.

Esta problemática no es ajena a la institución objeto de estudio. En particular, se presentan ineficiencias relevantes en el servicio de pagos masivos de nóminas, uno de sus productos clave enfocados a la banca empresa. Este servicio es el encargado de procesar hasta 2000 pagos en una sola planilla ingresada al sistema, habilitando a los clientes

empresa a ejecutar el desembolso de haberes a sus colaboradores de manera agrupada.

No obstante, desde años anteriores este producto ha experimentado episodios de inestabilidad, especialmente en horas pico y al cierre de mes. Además, los clientes deben ajustarse a ventanas horarias para operar y a un procesamiento diferido de las planillas ingresadas. El límite de pagos por lote resulta insuficiente para organizaciones de gran escala debido a su elevado número de empleados. A esto se suma la falta de trazabilidad y una monitorización de los pagos de extremo a extremo (E2E), factores que dificultan el control operativo y la atención oportuna de incidencias.

Una de las principales limitaciones es la tecnología legacy, desarrollos en COBOL sobre un mainframe de IBM, sobre la cual se realizan las operaciones. Aunque esta plataforma destaca por su capacidad de procesar grandes volúmenes de información, el enfoque batch limita el tratamiento en tiempo real de los pagos. Asimismo, la arquitectura monolítica del servicio, la cual no expone APIs, restringe la reutilización de componentes y por ende la escalabilidad del producto.

3.1.1. Definición del problema

A continuación, se presenta la formulación del problema general y de los problemas específicos.

3.1.1.1. Problema General

La entidad financiera cuenta con un sistema para la ejecución de pagos masivos de nóminas que depende de manera crítica de una plataforma mainframe. Su arquitectura monolítica y su elevado costo de mantenimiento se han convertido en una barrera para la innovación digital. A ello se suman la limitada disponibilidad horaria, las restricciones de escalabilidad y la baja reutilización de componentes, factores que comprometen la competitividad estratégica de la institución.

3.1.1.2. Problemas Específicos

- **Ejecución diferida de procesos:** El ingreso de planillas al sistema solo está habilitado de 8:00 a 16:00 y su procesamiento se realiza mediante jobs batch en

ventanas horarias específicas. Esto limita la posibilidad de procesar operaciones en línea y atender requerimientos en tiempo real.

- **Escalabilidad limitada del producto:** El sistema opera sobre una plataforma mainframe con arquitectura monolítica que impide distribuir la carga en múltiples instancias. Al no ser posible escalar horizontalmente, el crecimiento depende únicamente del escalamiento vertical, una opción costosa y limitada. Esto reduce la capacidad para procesar mayores volúmenes de pagos en una sola planilla y aumenta el riesgo de saturación o fallas en momentos de alta demanda.
- **Baja reutilización de componentes:** La falta de modularización limita el aprovechamiento de funcionalidades ya existentes y dificulta su exposición como API, lo que reduce la eficiencia al desarrollar mejoras y restringe la flexibilidad para incorporar nuevas capacidades.
- **Ausencia de trazabilidad y monitorización de operaciones:** El sistema no registra el detalle de todas las etapas por las que atraviesa una planilla, lo que impide brindarle al cliente un estado real de sus pagos y dificulta el monitoreo de operaciones de lado del equipo de atención ante incidencias.

3.2. Solución

3.2.1. Objetivos

A continuación, se presenta la formulación del objetivo general y de los objetivos específicos.

3.2.1.1. Objetivo General

Implementar una solución tecnológica para el procesamiento en streaming de pagos masivos de nóminas, que mejore los niveles de servicio del producto y reduzca la dependencia del mainframe bancario. Asimismo, que habilite la liquidación en línea de planillas, permita incrementar el volumen de pagos a procesar y se base en servicios reutilizables.

3.2.1.2. *Objetivos Específicos*

- **Realizar la ejecución en near-real-time de procesos:** Habilitar disponibilidad 24x7 para el ingreso de planillas al sistema. Así como realizar el procesamiento del débito al cliente y el abono a los beneficiarios de los pagos mediante transacciones en línea de manera inmediata.
- **Incrementar la escalabilidad del producto:** Mejorar el rendimiento de la plataforma para manejar planillas de hasta 7 000 pagos, asegurando un servicio estable y escalable ante mayores volúmenes de transacciones.
- **Adoptar un enfoque basado en servicios reutilizables:** Modularizar las funcionalidades del sistema para favorecer la reutilización de componentes y habilitar la operación del cliente desde distintos canales digitales de la entidad.
- **Disponer de trazabilidad y monitorización E2E de operaciones:** Registrar y disponibilizar la información detallada de todas las etapas de la liquidación de operaciones, en función de que tanto los usuarios externos como internos puedan conocer, en todo momento y de forma online, los estados de las planillas y pagos.

3.2.2. *Alcance*

A continuación, se presenta la formulación del alcance organizacional, funcional y técnico.

3.2.2.1. *Alcance Organizacional*

El proyecto se centra en el producto digital de pagos masivos de nóminas dirigido a la base de clientes empresa de todos los segmentos (micro, pequeña, mediana y corporativa) de la entidad financiera. La implementación está a cargo del área de Tecnología, en función a los requerimientos definidos por el área sponsor del proyecto, Banca Empresa y Corporativa. Quedan excluidos de este alcance otros productos y servicios de pagos, tanto de esta misma unidad como de Banca Retail.

3.2.2.2. Alcance Funcional

Dentro de los requerimientos definidos por el área de Negocio, el proyecto abarca la implementación de los servicios de incorporación y liquidación de nóminas entre cuentas del mismo banco, así como la consulta del historial de planillas por cliente, el detalle de cada planilla y el listado y el detalle de sus pagos, con la información necesaria para realizar la trazabilidad del lado de los usuarios. Además, los servicios se diseñan para ser reutilizables desde cualquier canal digital de Banca Empresa.

3.2.2.3. Alcance Técnico

El alcance técnico se limita a la exposición de servicios backend, con el objetivo de abastecer a los distintos canales de cara al cliente. Incluye endpoints para la incorporación y liquidación de nóminas intrabanco, así como para la consulta del historial y detalle de planillas, y el listado y detalle de pagos con su trazabilidad. Esta capa contempla autenticación y autorización acordes a las políticas de la entidad, incorpora validaciones de límites transaccionales y controles antifraude. Quedan fuera de este alcance los desarrollos frontend.

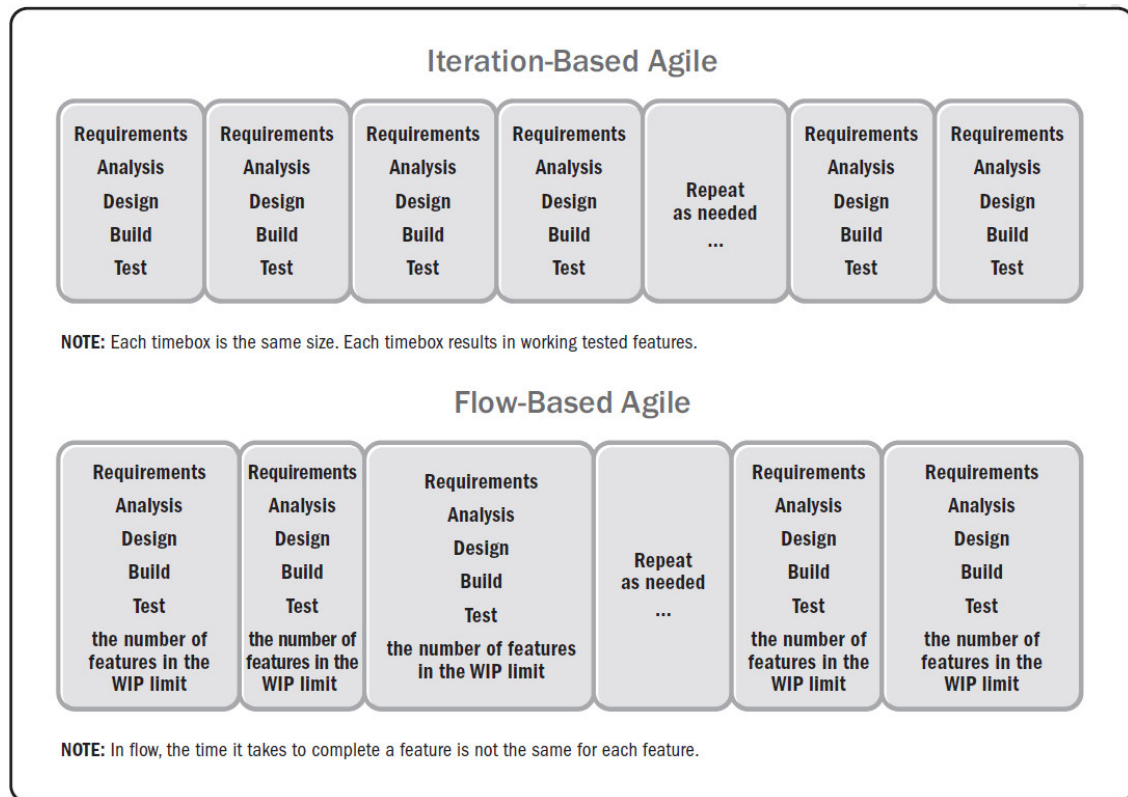
3.2.3. Etapas y Metodología

El proyecto se concibió y desarrolló bajo el enfoque ágil, idóneo para gestionar eficazmente tecnologías disruptivas y mantener la satisfacción del cliente como principal prioridad. En este entorno, donde se espera que los requisitos cambien, la naturaleza iterativa e incremental del enfoque permite perfeccionar las funcionalidades del producto y realizar entregas tempranas. (Project Management Institute & Agile Alliance, 2017)

En la siguiente figura se presenta el ciclo de vida ágil, con dos formas posibles de lograr una entrega incremental, con el fin de ajustar el proyecto a las necesidades del cliente y facilitar su adaptación cuando sea requerido. Para el presente proyecto se eligió el enfoque ágil basado en flujos, dado que la forma de trabajo considera que cada entregable responde a casuísticas distintas y presenta tiempos de finalización heterogéneos.

Figura 2

Ciclos de vida ágiles basados en iteraciones y basados en flujos



Nota. Fuente: Project Management Institute, & Agile Alliance. (2017). *Agile practice guide* (p. 24). Newtown Square, USA: Project Management Institute.

A continuación, se detallan las etapas que se siguieron:

3.2.3.1. ***Ideación y Refinamiento***

En esta etapa se realiza la definición temprana y efectiva de la solución funcional y de los objetivos del proyecto, siguiendo la metodología ágil Lean Inception. Dicha metodología ayuda a determinar un Producto Mínimo Viable (MVP) y producir un backlog inicial priorizado que habilita el inicio de la ejecución ágil. (Caroli, 2019). A la par, se elabora la solución técnica inicial en la que se establece la arquitectura propuesta y el modelo de datos base.

3.2.3.2. ***Planificación, Ejecución y Despliegue***

Con los MVPs definidos, el backlog inicial priorizado y la solución técnica base establecidos, se procede a la planificación, ejecución y despliegue de los entregables siguiendo el marco de trabajo Scrum. Este framework, de enfoque iterativo e

incremental, ayuda a los equipos a generar valor a través de soluciones adaptativas para problemas complejos. (Schwaber & Sutherland, 2020)

Schwaber y Sutherland (2020) definen al Scrum Team como la unidad fundamental de este marco de trabajo. Se trata de un equipo multifuncional y autogestionado, sin subequipos ni jerarquías, cuyos integrantes se alinean en torno al Objetivo del Producto.

Participantes del Scrum Team:

- **Developers:** Encargados de crear, en cada Sprint, cualquier aspecto utilizable de un Increment.
- **Product Owner:** Responsable de maximizar el valor del producto resultante del trabajo del Scrum Team y de realizar la gestión efectiva del Product Backlog.
- **Scrum Master:** Responsable de lograr la efectividad del Scrum Team; además, de promover y facilitar la adopción de Scrum dentro del equipo y de la organización en general.

Asimismo, se definen los siguientes eventos en los cuales se inspeccionan y adaptan los artefactos Scrum; estos establecen regularidad y reducen la necesidad de reuniones no previstas en el marco de trabajo.

- **El Sprint:** Evento de duración de un mes o menos donde se realiza el trabajo necesario para avanzar hacia el Objetivo del Producto.
- **Sprint Planning:** Sesión inicial del Sprint, en donde se establece el trabajo que se realizará en ese periodo y se discuten los ítems más importantes del Product Backlog y como interactúan con el Objetivo del Producto.
- **Daily Scrum:** Breve reunión diaria que tiene como propósito inspeccionar el avance hacia el Objetivo del Sprint.
- **Sprint Review:** Evento orientado a revisar el resultado del Sprint, discutir el progreso hacia el Objetivo del Producto y estudiar futuros cambios en el Product Backlog.

- **Sprint Retrospective:** Espacio en el que se analiza lo que funcionó, los problemas encontrados y sus causas, y se planifican mejoras para aumentar la calidad y la efectividad.

Por último, se establecen los artefactos de Scrum, que representan trabajo y/o valor, además, cada uno de ellos tiene un compromiso sobre el cual se puede medir el progreso.

- **Product Backlog:** Fuente de trabajo del Scrum Team, contiene el listado de todo lo necesario para mejorar el producto. Su compromiso es el Objetivo del Producto, definido como la meta a larga plazo del equipo.
- **Sprint Backlog:** Conjunto de elementos seleccionados del Product Backlog, acompañado de un plan de acción elaborado por los Developers en donde se define lo que se realizará durante el Sprint. Su compromiso es el Objetivo del Sprint, propósito del Sprint medido en base a sus resultados.
- **Increment:** Avance concreto hacia el Objetivo del Producto, establecido como una entrega que proporciona valor y es utilizable. Su compromiso es la Definición de Terminado, que marca las medidas de calidad requeridas por el producto.

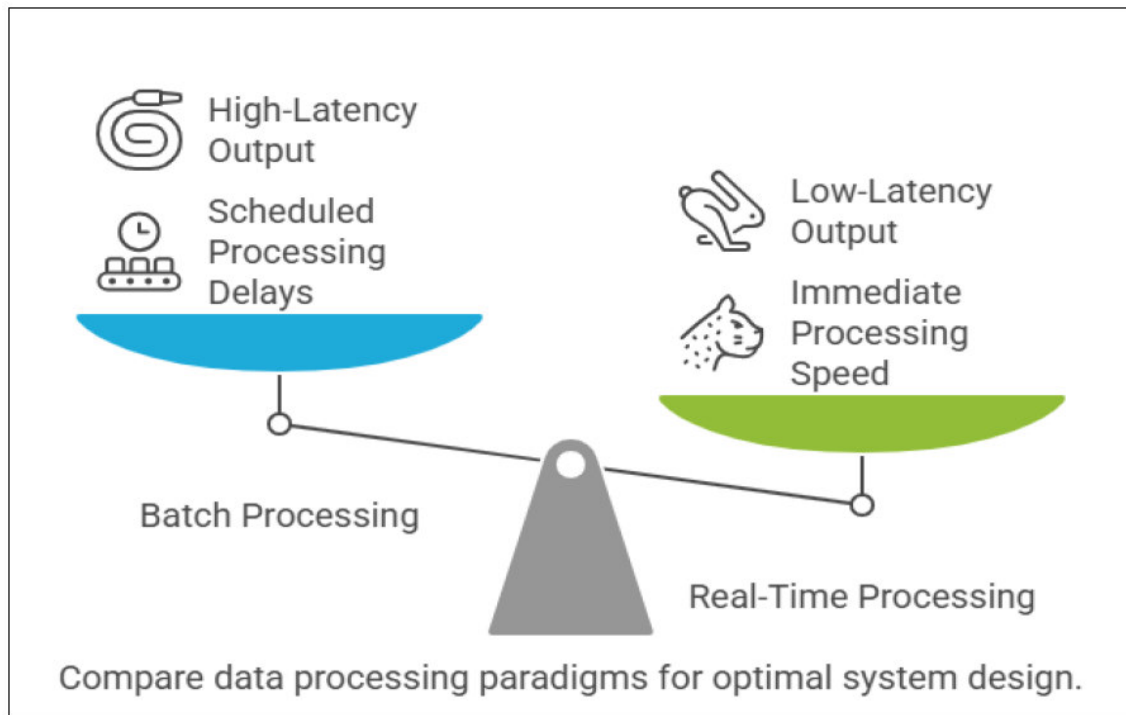
3.2.4. Fundamentos utilizados

3.2.4.1. Procesamiento Batch vs Procesamiento en Streaming

Los sistemas tradicionales de procesamiento por lotes, que recopilan y procesan información a través de jobs ejecutados en horarios programados, han sido durante años un componente central de la infraestructura de datos empresarial. Estos sistemas destacan por su eficiencia al manejar volúmenes masivos de información. Sin embargo, la transformación digital en diversas industrias ha impulsado nuevas demandas de procesamiento. En sectores como servicios financieros, comercio electrónico, sanidad y fabricación, las organizaciones necesitan cada vez más la capacidad de procesar los datos y generar respuestas inmediatas a medida que estos van ingresando a los sistemas. (Tatipamula, 2025)

Figura 3

Procesamiento Batch vs Procesamiento Real-Time



Nota. Fuente: Tatipamula, S. (2025). Real-Time vs. Batch Data Processing: When speed matters. *World Journal of Advanced Research and Reviews*, 26(01), 1612–1631. <https://doi.org/10.30574/wjarr.2025.26.1.1213> (p. 1613).

Procesamiento Batch

El procesamiento en lotes consiste en recolectar datos de un periodo de tiempo y luego procesarlo mediante jobs de gran tamaño ejecutados en horarios definidos. Para ello, los datos atraviesan etapas claramente establecidas, orientadas a maximizar la eficiencia y asegurar un procesamiento completo. Primero se realiza la recolección de la información, la cual se obtiene de distintas fuentes y se consolida en una plataforma de almacenamiento. Luego, se aplican procesos ETL (Extracción, Transformación y Carga) que llevan a cabo actividades de depuración y normalización para asegurar la consistencia de los datos. Finalmente, estas ejecuciones se programan en función de los requerimientos de negocio y la disponibilidad de los recursos. (Tatipamula, 2025)

Procesamiento en Streaming

En un entorno de procesamiento real-time, el flujo de datos sigue un pipeline distinto al que se utiliza en arquitecturas batch. El proceso comienza con la ingesta

continua de datos desde múltiples orígenes, como logs de aplicaciones, interacciones de usuarios, dispositivos IoT y sistemas transaccionales. A diferencia de los enfoques basados en procesar datasets completos, estos sistemas utilizan mecanismos de particionamiento para distribuir la carga entre los recursos de procesamiento. Luego, cada evento es procesado de inmediato, ejecutando transformaciones, enriquecimientos y funciones analíticas. Al final del pipeline se generan salidas disponibles en cuestión de milisegundos, las cuales pueden expresarse como alertas, actualizaciones en real-time dashboards o respuestas dirigidas a sistemas automatizados de toma de decisiones, entre otros. (Tatipamula, 2025)

Según un estudio de Badman y Kosinski (s.f.), las principales diferencias entre ambos enfoques radican en:

- **Modelo de procesamiento:** El procesamiento por lotes consolida y analiza la información en lotes en intervalos fijos, mientras que el procesamiento en streaming trata la información conforme va llegando. Por ello, los sistemas en streaming pueden actuar de forma inmediata, en contraste con los sistemas por lotes, que operan de manera periódica.
- **Necesidades de infraestructura:** Los sistemas batch se apoyan en herramientas tradicionales de almacenamiento y análisis, mientras que un sistema en streaming requiere infraestructura especializada y plataformas diseñadas para gestionar flujos de datos en tiempo real.
- **Requisitos de rendimiento:** En el procesamiento por lotes es posible optimizar el uso de los recursos durante las ejecuciones programadas, mientras que el procesamiento en streaming requiere sistemas tolerantes a fallas con baja latencia. Es decir, los sistemas de transmisión deben procesar los datos en tiempo real sin retrasos, incluso ante volúmenes elevados de información o la presencia de incidentes.

3.2.4.2. *Arquitectura Event-Driven (EDA)*

La arquitectura basada en eventos (EDA) es un enfoque de diseño de software orientado a la publicación, captura, procesamiento y almacenamiento de eventos. Este modelo permite que los equipos detecten eventos del sistema (es decir, cualquier cambio o acción que ocurra dentro de él) y respondan ante ellos en tiempo real o casi en tiempo

real. En una EDA, los emisores de eventos (por ejemplo, microservicios, APIs y dispositivos IoT) envían notificaciones hacia los consumidores, los cuales activan rutinas de procesamiento específicas en tiempo real. (IBM, s. f.)

Una de las ventajas principales de una arquitectura basada en eventos es el desacoplamiento entre los componentes frontend y backend, lo que hace posible que los sistemas compartan información sin necesidad de conocerse. Los emisores pueden publicar eventos sin saber qué consumidor los procesará, y los consumidores pueden recibirlos sin realizar solicitudes a los emisores. Lo que quiere decir que las EDA facilitan que los sistemas operen de manera independiente y gestionen eventos de forma asincrónica. (IBM, s. f.)

Definición de evento

La transmisión de eventos se centra en el flujo continuo, secuencial y en tiempo real de registros de datos conocidos como eventos, estos son estructuras que representan cualquier ocurrencia o cambio dentro de un sistema o su entorno. Por ejemplo, pueden reflejar acciones como cuando un usuario agrega un producto al carrito en un sitio de comercio electrónico, solicita un restablecimiento de contraseña o se produce un cambio de estado en una aplicación. En otras palabras, un evento equivale a un punto de datos del sistema, y un flujo o transmisión de datos corresponde a la entrega ininterrumpida de dichos eventos. (IBM, s. f.)

Por lo general, cada evento incluye una clave que permite identificar el evento o la entidad asociada, un conjunto de datos que contiene la información principal, una marca de tiempo que indica cuándo ocurrió o fue registrado y, en algunos casos, metadatos vinculados a la fuente de datos, la versión del esquema u otro atributo. (IBM, s. f.)

En muchos casos, los eventos se relacionan con operaciones del negocio o con el recorrido de navegación del usuario y, usualmente, desencadenan una acción posterior, un proceso o incluso una nueva secuencia de eventos. (IBM, s. f.)

Componentes clave de la arquitectura basada en eventos

Además de los eventos, las EDA se apoyan en tres componentes principales para transportar los datos de eventos a lo largo de la arquitectura. (IBM, s. f.)

- **Productores (o emisores) de eventos:** Son las fuentes de eventos que generan u envían los eventos hacia el resto del sistema. Un productor podría ser una interfaz de usuario, un sensor, un servicio u otro sistema capaz de detectar o emitir un cambio de estado relevante para otros componentes.
- **Consumidores de eventos (o suscriptores):** Son los responsables de ejecutar el procesamiento de eventos en una EDA. Se mantienen a la escucha de los canales de eventos y reaccionan cuando se publica un evento de interés. Esa reacción puede adoptar distintas formas, como actualizar una base de datos, iniciar un proceso posterior o registrar información.
- **Enrutadores de eventos (u orquestadores de eventos):** Actúan como intermediarios que desacoplan a los productores de eventos del frontend y a los consumidores del backend. Funcionan como el canal por el cual se entregan los eventos, permitiendo su transmisión sin una conexión directa entre las partes. Estos enrutadores pueden basarse en componentes convencionales y utilizarse en diversos mecanismos de mensajería, tales como colas de mensajes, modelos de publicación/suscripción, evento streams, entre otros.

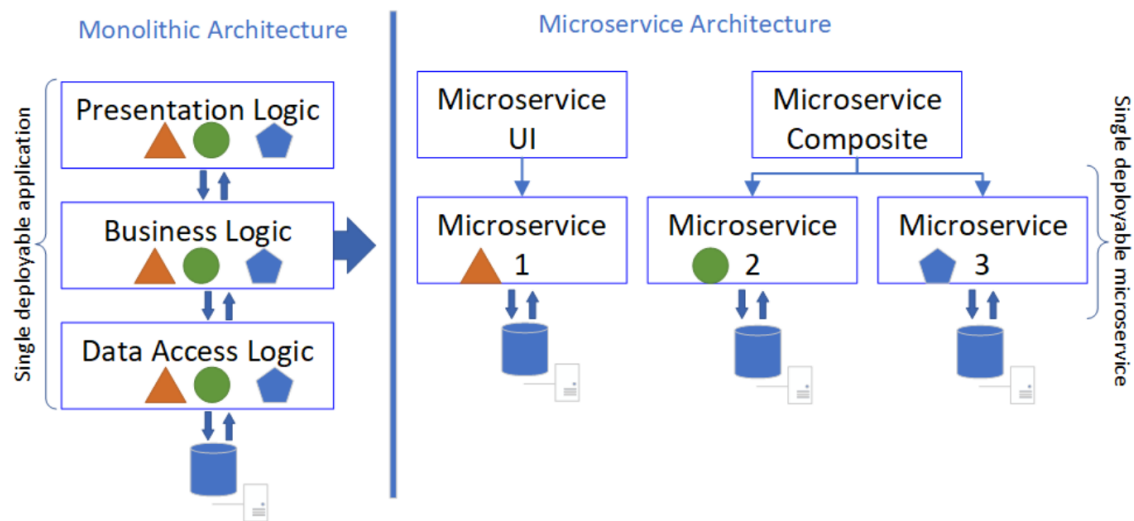
3.2.4.3. *Arquitectura Monolítica vs Arquitectura de Microservicios*

La arquitectura monolítica y la arquitectura de microservicios representan dos maneras diferentes de diseñar y desarrollar sistemas de software. Optar por una u otra depende de múltiples variables, por lo que no puede afirmarse de forma general que una sea mejor que la otra. Cada enfoque presenta ventajas y limitaciones, y la elección debe alinearse con las necesidades y condiciones particulares del proyecto. (Velepucha & Flores, 2023)

En la siguiente figura se presenta una comparación entre el diseño de una arquitectura monolítica y una arquitectura basada en microservicios.

Figura 4

Arquitectura Monolítica vs Arquitectura de Microservicios



Nota. Fuente: Velepucha, V., & Flores, P. (2023). A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges. *IEEE, 11*, 88339 – 88358. <https://doi.org/10.1109/ACCESS.2023.3305687> (p. 88343).

Arquitectura Monolítica

Una arquitectura monolítica se define por contar con un único ejecutable; aunque pueda estar compuesta internamente por varios módulos, estos no se ejecutan de manera independiente. Si bien la solución puede estructurarse en capas lógicas, el despliegue se realiza de forma conjunta, de modo que toda la funcionalidad se entrega como un solo bloque y se ejecuta como un único proceso. La principal limitación de este enfoque es que, conforme aumentan las funcionalidades del negocio y, con ello, la complejidad del sistema, su mantenimiento tiende a volverse insostenible. Asimismo, resulta complejo incorporar tecnologías más recientes cuando se trabaja sobre una arquitectura rígida y estandarizada, orientada al uso de una única tecnología. En este contexto, se observan restricciones asociadas a la escalabilidad, la mantenibilidad y la flexibilidad para implementar cambios, además de posibles afectaciones en disponibilidad, confiabilidad y tiempos de desarrollo. (Velepucha & Flores, 2023)

Arquitectura de Microservicios

Un microservicio es un componente de una aplicación que puede ejecutarse de manera independiente, a diferencia de una aplicación monolítica. Cada microservicio se

implementa para cubrir una funcionalidad específica del negocio y no se estructura por capas lógicas como ocurre en el monolito; en consecuencia, suele caracterizarse por un acoplamiento flexible y una alta cohesión funcional. Al tratarse de servicios pequeños, pueden desarrollarse de forma autónoma y operarse de manera comparable a como se gestiona una aplicación monolítica. (Velepucha & Flores, 2023)

La adopción de microservicios suele asociarse con la facilidad de desarrollo al delimitar responsabilidades por funcionalidad, con la posibilidad de escalar mediante contenedores y con la conveniencia de desplegarlos en entornos de cloud computing. No obstante, también deben considerarse desafíos vinculados a su gestión y monitoreo. En escenarios de migración desde un monolito, el objetivo central es descomponer la aplicación en servicios pequeños, autónomos e independientes. En línea con ello, las tendencias actuales destacan la migración de aplicaciones monolíticas hacia microservicios y su despliegue en cloud computing, así como la automatización de actividades de compilación y despliegue en soluciones migradas. (Velepucha & Flores, 2023)

3.2.4.4. *Core bancario en mainframe: COBOL, procesamiento batch (JCL) y procesamiento transaccional (CICS)*

Los mainframes son plataformas de procesamiento de gran tamaño y alta capacidad que, desde inicios de la década de 1950, han ocupado un rol central en los sistemas de grandes organizaciones. En sus primeros años, IBM los desarrolló para administrar diversos tipos de información y entregar potencia de cómputo a los entornos empresarial, administrativo y financiero. Estos sistemas están orientados a procesar en paralelo grandes volúmenes de datos y suelen sobresalir por su confiabilidad, escalabilidad y seguridad, incluso frente a computadoras personales y algunos entornos de servidores. Aunque hoy existen alternativas basadas en computación distribuida y soluciones cloud, los mainframes siguen siendo relevantes cuando se necesita operación continua y un alto volumen de transacciones. (Gill, 2019)

En este contexto, el lenguaje COBOL ha sido clave, ya que desde su origen en los años cincuenta hasta la actualidad se sigue utilizando para abordar operaciones a gran escala típicas de los mercados financieros. Se emplea con frecuencia en procesamiento batch por su capacidad para manejar grandes conjuntos de datos con rapidez,

especialmente cuando se requieren transformaciones sobre esos datos. En aplicaciones bancarias, por ejemplo, COBOL se usa para procesar transacciones, actualizar saldos y generar reportes. Además, su sintaxis, basada en designaciones simples, suele asociarse con código mantenible. (Gill, 2019)

Para el procesamiento batch en mainframe, JCL (Job Control Language) también cumple un rol importante. JCL funciona como un lenguaje de scripting que permite definir, programar y administrar jobs batch dentro del mainframe. Su aporte principal es especificar cómo se ejecutan estos jobs de forma automatizada: qué programas corren, en qué horarios y en qué secuencia, con el fin de mantener el orden correcto del procesamiento. En ese sentido, un control y planificación adecuados de los jobs es determinante para manejar altos volúmenes transaccionales, ya que ayuda a optimizar el uso de recursos y a evitar sobrecargas en ciclos de trabajo intensivo. (Gill, 2019)

Por otro lado, el procesamiento transaccional es un estilo de computación orientado a soportar aplicaciones interactivas. En este enfoque, el trabajo se divide en operaciones individuales e indivisibles, llamadas transacciones. Esto contrasta con el procesamiento batch, donde uno o más programas procesan una serie de registros (un lote) con poca o ninguna acción por parte del usuario u operador. (Rayns et al., 2011)

Un sistema de procesamiento transaccional permite gestionar el procesamiento concurrente de transacciones, habilitar el intercambio de datos, garantizar la integridad de la información y administrar la priorización de la ejecución. Cuando una transacción inicia su procesamiento, CICS (Customer Information Control System) ejecuta un programa asociado; dicho programa puede transferir el control a otros programas durante la ejecución, lo que permite construir aplicaciones modulares compuestas por múltiples programas CICS. En un sistema CICS pueden ejecutarse simultáneamente muchas instancias de una transacción, y cada instancia en ejecución se conoce como una task (tarea). Durante el tiempo en que una task está activa, tiene uso exclusivo de cada recurso de datos que modifica, garantizando la propiedad de aislamiento de la transacción. (Rayns et al., 2011)

3.2.4.5. Cloud Computing

El cloud computing consiste en acceder bajo demanda, a través de Internet y con un esquema de pago por uso, a recursos informáticos como servidores físicos o virtuales,

almacenamiento de datos, capacidades de red, herramientas para el desarrollo de aplicaciones, software y plataformas analíticas con IA, entre otros. Este modelo ofrece mayor flexibilidad y escalabilidad en comparación con la infraestructura tradicional local. De esta manera, es posible obtener la capacidad de cómputo o el almacenamiento necesarios sin tener que adquirir ni administrar directamente el hardware físico. (Susnjara & Smalley, s.f.)

En la actualidad, el cloud computing se ha vuelto indispensable en los entornos empresariales, precisamente por esa flexibilidad y escalabilidad superiores frente a la infraestructura on-premise. Entre sus aplicaciones se incluyen el soporte al teletrabajo al facilitar el acceso a datos y aplicaciones desde cualquier lugar, la habilitación de un marco para una atención al cliente omnicanal sin fricciones y la provisión de la potencia de cómputo y otros recursos requeridos para aprovechar tecnologías de vanguardia, como la IA generativa y la computación cuántica. (Susnjara & Smalley, s.f.)

En comparación con la TI local tradicional, donde la empresa debe adquirir y mantener centros de datos y servidores propios para disponer de potencia de cómputo, almacenamiento y otros recursos, el cloud computing aporta beneficios relevantes. Entre ellos, destaca la rentabilidad, ya que reduce el gasto y el esfuerzo asociados a comprar, instalar, configurar y administrar infraestructura en las instalaciones, y permite pagar únicamente por los recursos utilizados. También favorece una mayor velocidad y agilidad, al posibilitar el uso de aplicaciones empresariales en minutos en lugar de esperar semanas o meses por aprovisionamiento e instalación. Asimismo, ofrece escalabilidad mediante elasticidad y autoservicio, permitiendo ampliar o reducir capacidad según variaciones de demanda. Finalmente, mejora el valor estratégico, al facilitar la adopción de tecnologías e innovaciones recientes para fortalecer la competitividad en distintos sectores, incluyendo el uso de soluciones con IA y el soporte a la colaboración y monitoreo en tiempo real en procesos de negocio. (Susnjara & Smalley, s.f.)

Servicios de computación en la nube

La infraestructura como servicio (IaaS), la plataforma como servicio (PaaS), el software como servicio (SaaS) y la computación sin servidor (serverless) se consideran los modelos “como servicio” más habituales dentro de las plataformas de nube. En organizaciones de gran escala, la mayoría de los equipos de desarrollo suele adoptar una

combinación de estos cuatro enfoques. (Susnjara & Smalley, s.f.)

- **IaaS (infraestructura como servicio):** Ofrece acceso bajo demanda, a través de internet y bajo un esquema de pago por uso, a recursos informáticos básicos como servidores físicos o virtuales, redes y almacenamiento. Este modelo permite ajustar la capacidad según las necesidades, lo que disminuye la dependencia de grandes inversiones iniciales y de infraestructura propia u on-premise.
- **PaaS (Plataforma como servicio):** Brinda a los desarrolladores una plataforma bajo demanda para ejecutar, desarrollar y gestionar aplicaciones sin asumir el costo y la complejidad de mantener esa plataforma en las instalaciones; en este modelo, el proveedor de nube aloja en su centro de datos la infraestructura y la pila necesaria (como servidores, redes, almacenamiento, sistema operativo, middleware y bases de datos), de modo que los desarrolladores solo seleccionan los recursos y entornos que requieren para crear, probar, implementar, mantener, actualizar y escalar sus aplicaciones.
- **SaaS (Software como servicio):** También llamado software o aplicaciones en la nube, corresponde a software de aplicación interactivo alojado en la nube al que los usuarios acceden mediante un navegador web, un cliente de escritorio dedicado o una API integrada con un sistema operativo de escritorio o móvil. Por lo general, los proveedores lo ofrecen bajo un esquema de suscripción mensual o anual, aunque también puede comercializarse con precios de pago por uso.
- **Serverless (Computación sin servidor):** Es un modelo de nube en el que el proveedor asume las tareas de administración de la infraestructura backend, como aprovisionamiento, escalado, programación y aplicación de parches, lo que permite que los desarrolladores se enfoquen en el código y la lógica de negocio. En este enfoque, el código se ejecuta únicamente cuando es solicitado y la infraestructura se escala automáticamente según la cantidad de solicitudes. Además, el cliente paga solo por los recursos consumidos durante la ejecución, sin costos por capacidad inactiva.

Tipos de cloud computing

Susnjara y Smalley (s. f.) presentan los siguientes cuatro tipos principales de nube.

- **Nube Pública:** Es un modelo de computación en la nube en el que un proveedor pone a disposición de los usuarios, a través de Internet público, distintos recursos informáticos, como aplicaciones SaaS, máquinas virtuales (VM), hardware, infraestructuras completas de nivel empresarial y plataformas de desarrollo. El acceso a estos recursos puede ser gratuito o estar sujeto a esquemas de precios por suscripción o pago por uso.
- **Nube Privada:** Es un entorno en el que la infraestructura y los recursos de cómputo de la nube se asignan exclusivamente a un único cliente. Este modelo combina beneficios propios de la computación en la nube, como elasticidad, escalabilidad y facilidad de provisión de servicio, con un mayor control de acceso, seguridad y capacidad de personalización de los recursos, en línea con lo que se busca en infraestructura on-premises.
- **Nube Híbrida:** Es una combinación de nube pública, nube privada y entornos on-premises. En términos prácticos, integra estos tres entornos en una infraestructura flexible que permite ejecutar las aplicaciones y cargas de trabajo de la organización.
- **Multicloud:** Se refiere al uso de dos o más nubes pertenecientes a dos o más proveedores distintos. Un entorno multinube puede ser tan sencillo como utilizar el SaaS de correo electrónico de un proveedor y el SaaS de edición de imágenes de otro; sin embargo, en el contexto empresarial normalmente implica consumir múltiples servicios en la nube, incluyendo SaaS, PaaS e IaaS, de dos o más proveedores.

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

Para la implementación del presente proyecto se siguió el enfoque ágil, combinando la metodología Lean Inception durante las fases de Ideación y Refinamiento, y el marco de trabajo Scrum para las etapas de Planificación, Ejecución y Despliegue. A continuación, se describen las fases consideradas en el desarrollo.

3.2.5.1. Ideación y refinamiento

Durante esta etapa se realizó el levantamiento de la información junto con las partes interesadas, con el fin de lograr una visión alineada de los objetivos, las estrategias y el alcance del proyecto. Esto permitió definir los entregables y establecer un backlog inicial de funcionalidades. Asimismo, se elaboró el documento de diseño de solución, en el cual se especificó la solución funcional y técnica.

En la siguiente tabla se detallan las actividades realizadas, los participantes involucrados y los entregables obtenidos.

Tabla 5

Listado de Actividades Ideación y Refinamiento

Actividad	Participantes	Entregable	Duración
Ideación	Sponsor Product Owner	Business Canvas	10 días hábiles
Refinamiento	Sponsor Product Owner Solution Architect Scrum Master / Tech Lead	Documento de Diseño de Solución Listado priorizado de entregables MVPs y EMCs Roadmap Alto Nivel	20 días hábiles

Nota. Elaboración Propia.

Siguiendo los lineamientos de la metodología Lean Inception, se inició con la etapa de Ideación, durante la cual el Sponsor y el Product Owner pudieron establecer la visión del producto, aclarar las características del producto, identificar los tipos de usuario y sus necesidades, y analizar los puntos clave del problema. Este proceso permitió clasificar y priorizar las principales funcionalidades, estableciendo finalmente el Business

Canvas, donde se definió los elementos esenciales de la ruta inicial del producto.

Figura 5

Business Canvas

1. ¿CUÁL ES LA NECESIDAD?		5. SITUACIÓN ACTUAL	
Brindar a nuestros clientes una experiencia digital renovada para la gestión de sus órdenes de pago, disminuyendo la intervención manual y facilitando la realización y trazabilidad de los pagos desde cualquier lugar, en cualquier momento y a través de los distintos canales del Banco.		La plataforma actual presenta limitaciones que afectan la eficiencia operativa y la experiencia de nuestros clientes. Su estructura dificulta la incorporación de mejoras, no permite una gestión oportuna y transparente de las órdenes de pago y, en los últimos meses, ha generado incidencias que impactan la continuidad del servicio.	
2. PROPUESTA DE VALOR	3. SEGMENTO DE CLIENTES	7. ACTIVIDADES CLAVE	
Esta iniciativa busca habilitar el procesamiento de órdenes de pago en near real time, garantizar una trazabilidad E2E y ofrecer una plataforma con alta disponibilidad y capacidad de escalamiento.	Clientes empresa que gestionan altos volúmenes de pagos, como nóminas y pagos a proveedores.	Diseño, desarrollo y mantenimiento de la plataforma de pagos. Gestión de integración con sistemas internos (canales, core bancario, contabilidad). Gestión de integración con sistemas de clientes (APIs, pruebas, certificaciones). Actualizaciones de seguridad y cumplimiento con regulaciones.	
4. ENTREGABLES ESPERADOS POR EL USUARIO		6. CANALES	8. RECURSOS CLAVE
Módulo para la incorporación de transferencias masivas de nóminas mismo banco. Módulo para el procesamiento near real time de transferencias masivas de nóminas mismo banco. Módulo de seguimiento y trazabilidad de operaciones de nóminas.		Canal Web Canal Movil APIs	Plataforma tecnológica de procesamiento en streaming. Equipo de desarrollo y arquitectura. Equipo de operaciones y monitoreo 24/7.
9. INDICADORES CLAVE DE DESEMPEÑO			
KPI1: Monto total de comisiones por transacciones procesadas vía API. KPI2: Volumen de transacciones procesadas. KPI3: Número de clientes nuevos activos en la plataforma.			

Nota. Elaboración Propia.

Con base en la información proporcionada por el equipo de Negocio en el Business Canvas, se continuó con la etapa de Refinamiento. Durante esta fase se tuvieron sesiones conjuntas entre los equipos de Negocio e Ingeniería con el propósito de concretar la solución requerida para atender la necesidad identificada. A partir de este trabajo colaborativo, el Solution Architect elaboró el Documento de Diseño de Solución Inicial, en el que se definieron dos componentes principales: la solución funcional y la solución técnica. Ambos aspectos se describen a continuación.

Solución Funcional

Como parte de la solución funcional propuesta, se identificaron los principales usuarios que interactúan con la plataforma y participan en su operación. Los actores que se definieron fueron los siguientes:

- **Clientes Empresa:** Responsables de contratar y utilizar los servicios ofrecidos por la plataforma.
- **Empleados:** Incluyen al equipo de atención y soporte operativo, encargado de brindar apoyo a los clientes empresa durante el uso de la plataforma, así como a los administradores, responsables del funcionamiento general y de la parametrización de la plataforma.

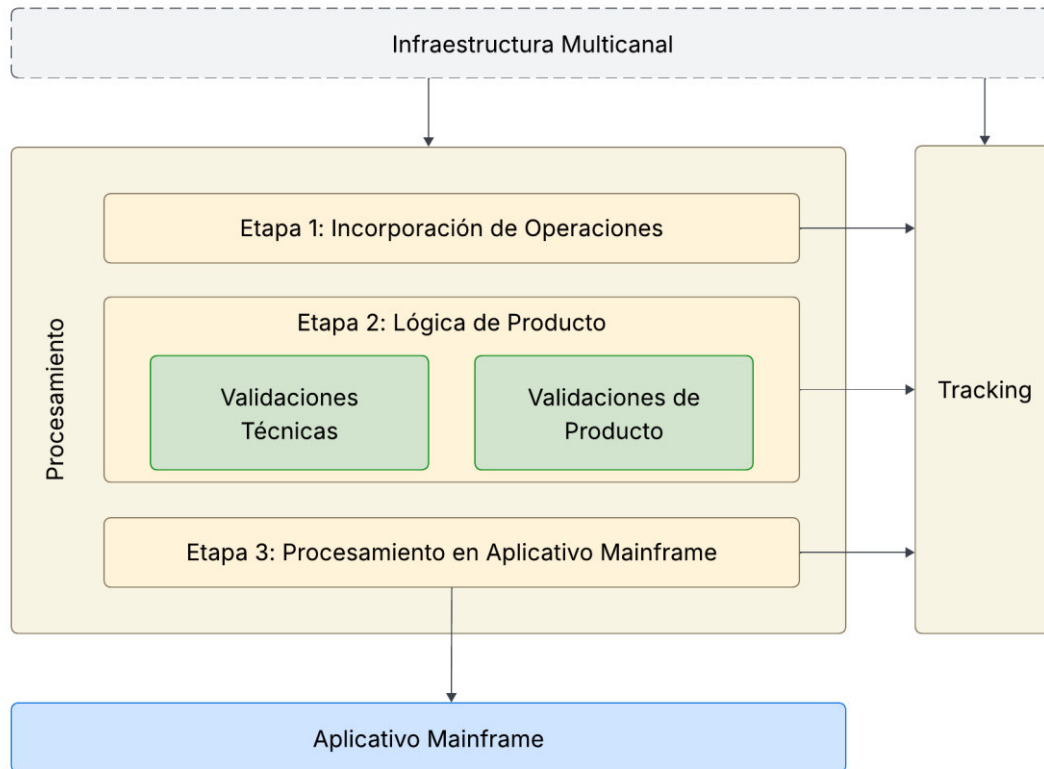
Asimismo, se definió el flujo funcional del proceso, considerando que el procesamiento de pagos se lleva a cabo en dos pasos principales: creación y liquidación. En el apartado siguiente se describen dichas etapas.

- **Autenticación de usuario:** El usuario accede a los canales del banco y se autentica utilizando sus credenciales.
- **Autorización de la planilla de pagos:** Durante la etapa de preproceso se realizan las validaciones necesarias para asegurar que el usuario cuenta con los permisos y el perfil requeridos para ejecutar el servicio, así como para completar el proceso de firma correspondiente.
- **Incorporación de la planilla de pagos:** Una vez que ha pasado las validaciones del preproceso, se realizan una serie de acciones para asegurar la correcta recepción y almacenamiento de la planilla, dejándola lista para su posterior tratamiento.
- **Validaciones sobre la planilla de pagos:** Tras realizar el almacenamiento adecuado de la planilla, se realizan validaciones adicionales con el objetivo de minimizar rechazos en las etapas posteriores de imputación, tales como verificación de límites transaccionales y controles de fraude.
- **Liquidación de la planilla de pagos:** Finalizadas las validaciones, se procede con la liquidación de la planilla, ejecutando los pagos en el aplicativo mainframe.

Se definieron los siguientes bloques funcionales que cubren la necesidad planteada de negocio: Canales, Procesamiento, Aplicativo Mainframe y Tracking, tal como se muestra en la Figura 6.

Figura 6

Diagrama de Solución Funcional



Nota. Elaboración Propia.

La capa de Canales, también denominada capa de distribución, es responsable de ofrecer a los distintos usuarios de la plataforma los puntos de entrada al Banco para su interacción. Esta capa estará conformada por los canales web, móvil y APIs.

La capa de Procesamiento es responsable del tratamiento de los pagos masivos enviados por el cliente, se subdivide en tres principales etapas:

- **Etapa 1 Incorporación de Operaciones:** Durante esta etapa, la planilla es recepcionada a través del servicio junto con la información general de los pagos, la cual es enriquecida con las condiciones específicas del cliente según su parametrización. Posteriormente, se genera un identificador único para la planilla y un identificador individual para cada operación, lo que permite asegurar la trazabilidad del proceso. Las operaciones recibidas se almacenan en las bases de datos operacional e informacional y, finalmente, se responde al cliente confirmando la correcta recepción y almacenamiento de la planilla.

- **Etapa 2 Lógica de Producto:** Una vez que la información ha sido incorporada en las bases de datos y se ha completado el proceso de firma, se inicia la etapa de preparación para la liquidación. En esta fase, y en función del volumen de pagos, la información es particionada en bloques con el fin de facilitar su procesamiento. A partir de estas particiones se realizan validaciones técnicas y de producto sobre los pagos, donde las primeras verifican la estructura y consistencia de la información, y las segundas aseguran, desde el punto de vista de negocio, que las órdenes puedan ser ejecutadas, incluyendo controles de límites operacionales y fraude. Los resultados de las validaciones, tanto de pagos exitosos como rechazados, se almacenan con el estado correspondiente, manteniendo coherencia entre el estado de la planilla y el de los pagos que lo componen.
- **Etapa 3 Procesamiento en Aplicativo Mainframe:** Esta etapa se encarga de seleccionar y agrupar los pagos que han superado las validaciones de la etapa previa, para ser enviados al Aplicativo Mainframe, responsable de la liquidación y contabilización de las operaciones. El tamaño de los grupos es parametrizable y se define en función de la capacidad del aplicativo para gestionar las operaciones de manera unitaria. El resultado del envío se registra en la base de datos con el estado correspondiente, reflejando la situación final de las operaciones y asegurando que todos los pagos de la planilla hayan sido procesadas. En caso de presentarse operaciones rechazadas con posterioridad al cargo, estas se registran para permitir la ejecución del proceso posterior de devolución.

La capa del Aplicativo Mainframe es la encargada de realizar la liquidación final de la planilla, la cual es recibida en bloques de pagos y procesada de manera online. En esta etapa se ejecuta un único cargo por el total de la planilla, conocido como cargo agrupado, y posteriormente se realizan los abonos correspondientes conforme van siendo procesados los bloques de pagos. Asimismo, se actualizan los estados finales de cada operación y, en caso de existir pagos rechazados, estos se agrupan para ejecutar la devolución del importe correspondiente en un proceso posterior, una vez concluido el procesamiento de todos los pagos.

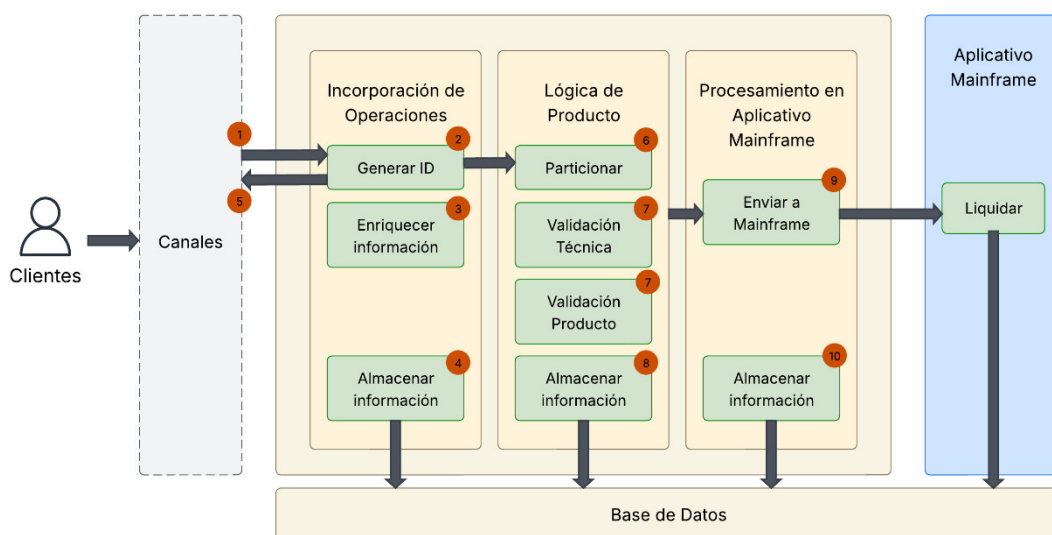
La capa de Tracking permite realizar el seguimiento end to end de cada planilla que ingresa a la plataforma, utilizando los identificadores asignados en la etapa de incorporación. De esta forma, tanto los clientes como los empleados del banco encargados

de dar atención a los clientes podrán conocer, en todo momento y de forma online, el estado de cada uno de los pagos.

El siguiente diagrama de secuencia presenta el flujo de interacción entre los distintos componentes de la solución, durante el procesamiento de una planilla de pagos. Este diagrama permite visualizar cómo se coordinan las diferentes capas del sistema a lo largo de las etapas de incorporación, validación, procesamiento y liquidación de las operaciones.

Figura 7

Diagrama de Secuencia de Solución Funcional



Nota. Elaboración Propia.

Detalle del módulo funcional:

1. El cliente envía su planilla de pagos a través de los canales del banco.
2. Se genera un ID único por planilla y por cada uno de los pagos.
3. Se enriquece la información enviada con las condiciones específicas del cliente.
4. Se almacena la información en la base de datos operacional e informacional.
5. Se regresa como respuesta al cliente el ID de la planilla y de cada uno de los pagos.
6. Se particiona el bloque de pagos recibido en bloques más pequeños que permitan

paralelizar y optimizar el proceso.

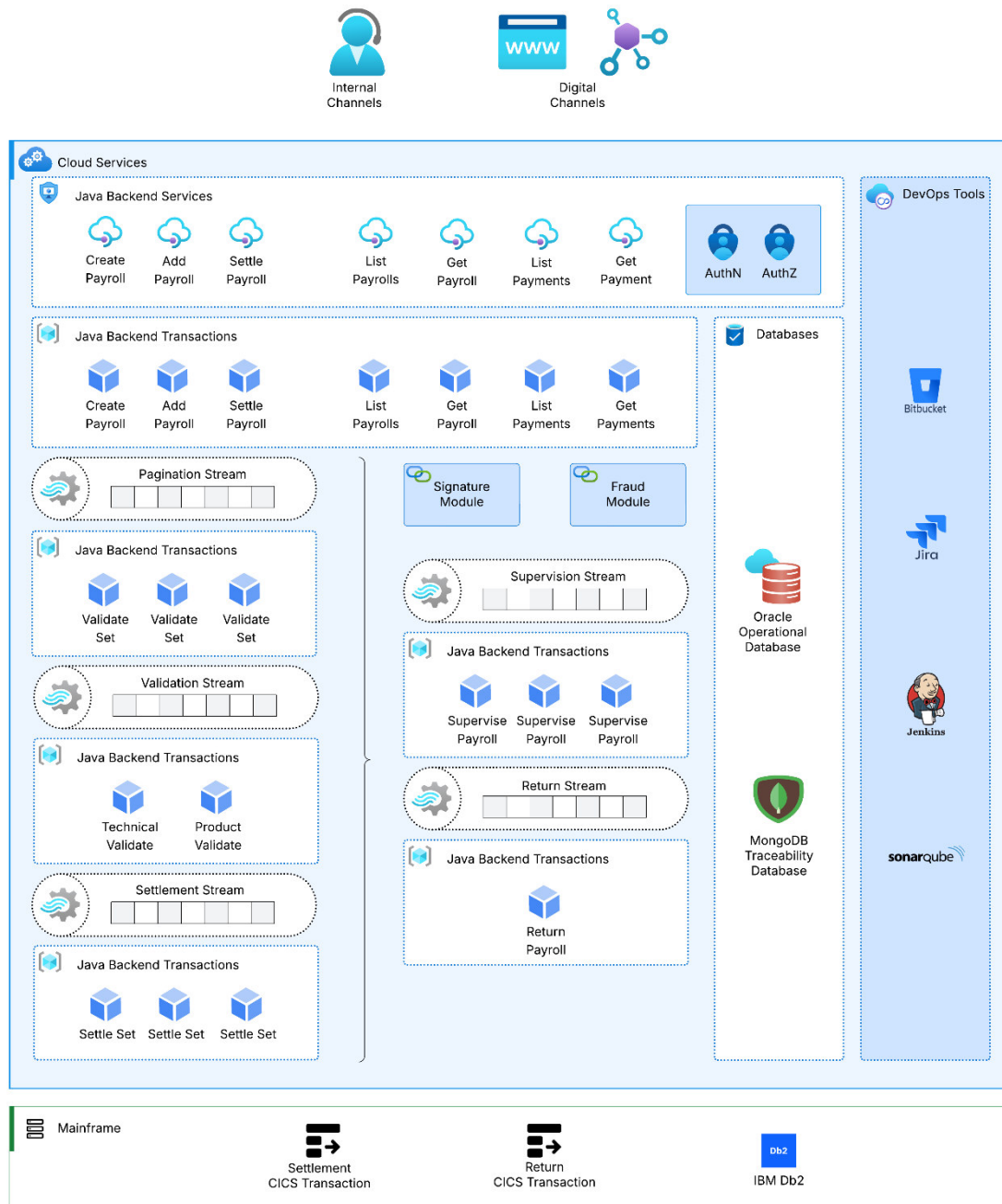
7. Se realiza las validaciones técnicas y de producto sobre los bloques de información particionados.
8. Se almacena la información en la base de datos operacional e informacional.
9. Se invoca al Aplicativo Mainframe para hacer la liquidación de la planilla de pagos.
10. Se almacena la información en la base de datos operacional e informacional.

Solución Técnica

Una vez definida la solución a nivel funcional, se estableció la arquitectura técnica a nivel backend de acuerdo con el alcance del proyecto. En esta etapa se definió la infraestructura base sobre la cual se ejecutarán los servicios y se identificaron los principales componentes, como la capa de servicios, la capa de transacciones, la capa de datos y los flujos de eventos, necesarios para asegurar el correcto procesamiento y la trazabilidad de las planillas. Asimismo, se planteó la reutilización de la capa de seguridad ya existente dentro de los servicios bancarios para los procesos de autenticación y autorización, así como de los módulos de firma y fraude ya disponibles en el catálogo del banco.

Figura 8

Diagrama de Solución Técnica



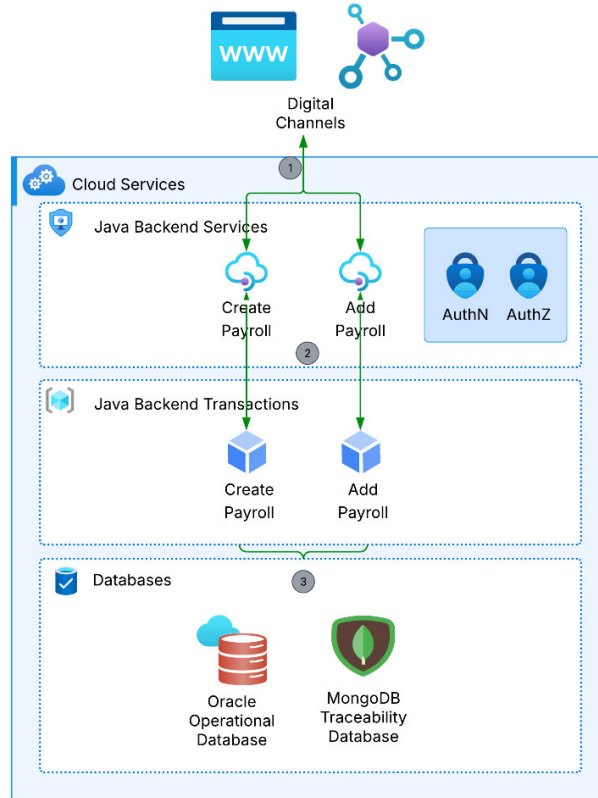
Nota. Elaboración Propia.

Para separar claramente las funcionalidades de incorporación, liquidación en streaming y trazabilidad, así como los componentes técnicos involucrados, se elaboraron diagramas de secuencia independientes para cada uno de estos flujos.

- Incorporación de Operaciones

Figura 9

Diagrama de Secuencia de Incorporación de Operaciones



Nota. Elaboración Propia.

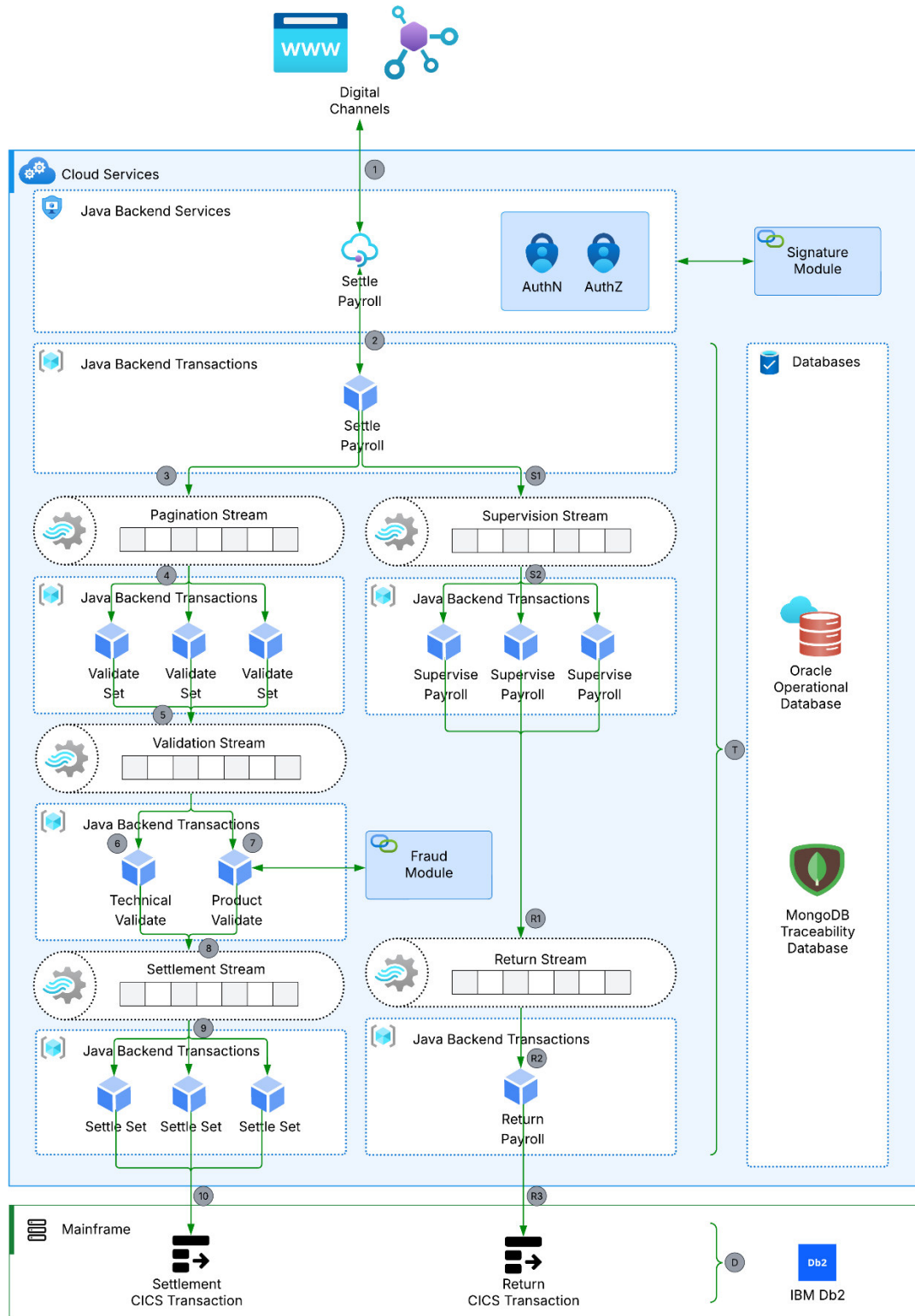
A continuación, se detallan los pasos para incorporar (crear y añadir) una planilla enviada por el Cliente, independientemente del canal de entrada.

1. Desde el canal, se invoca al servicio backend, donde se ejecuta una etapa de preproceso en la que se realizan las validaciones de seguridad necesarias para la ejecución del servicio, tales como la autenticación y autorización.
2. Una vez superadas las prevalidaciones a nivel de servicio, se ejecuta la transacción backend encargada de realizar las validaciones de los datos de entrada, así como de generar el identificador único para la planilla y para cada uno de los pagos.
3. Como último paso, la información procesada se almacena en la base de datos operacional Oracle y en la base de datos informacional MongoDB.

- Liquidación en Streaming de Operaciones

Figura 10

Diagrama de Secuencia de Liquidación en Streaming de Operaciones



Nota. Elaboración Propia.

A continuación, se detallan los pasos para liquidar una planilla enviada por el Cliente, independientemente del canal de entrada.

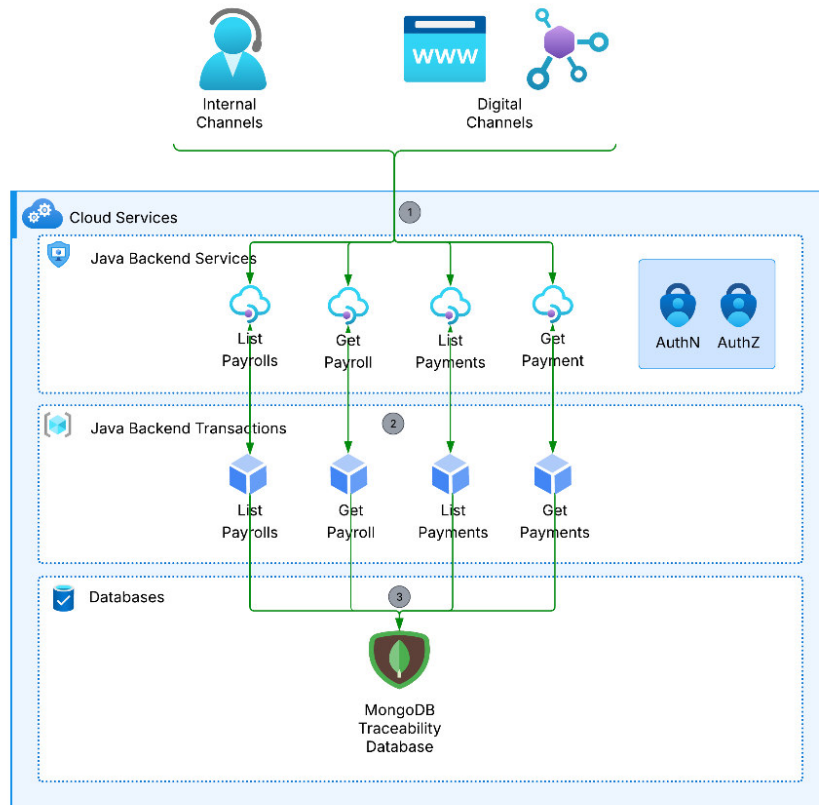
1. Desde el canal se invoca el servicio backend, enviando el identificador de la planilla y la cuenta ordenante a liquidar. En este paso se ejecuta un preproceso en el que se realizan las validaciones de seguridad necesarias para la ejecución del servicio, como la autenticación y la autorización, y se lleva a cabo la firma de la planilla utilizando los mecanismos ya establecidos en el banco.
2. Una vez superadas las prevalidaciones a nivel de servicio, se ejecuta la transacción backend responsable de obtener y procesar la información de la planilla necesaria para el flujo de paginación.
3. Se invoca al evento de paginación, encargado de mapear la información recepcionada y dividir la planilla en subconjuntos de 10 pagos.
4. Cada subconjunto se procesa de manera independiente y ejecuta la transacción backend responsable de obtener y procesar la información del bloque necesaria para el flujo de validaciones.
5. Se invoca al evento de validaciones, encargado de dirigir los subconjuntos hacia las dos transacciones principales de validación: validaciones técnicas y validaciones de producto.
6. Cada subconjunto se procesa de manera independiente y ejecuta la transacción backend responsable de realizar las validaciones técnicas de estructura y formato de los datos por pago. Si algún pago no cumple con las condiciones establecidas, es rechazado y se continúa con el procesamiento de los demás pagos.
7. Cada subconjunto se procesa de manera independiente y ejecuta la transacción backend responsable de realizar las validaciones de producto, como la verificación de límites transaccionales de la cuenta ordenante y los controles antifraude. Si algún pago no cumple con las condiciones establecidas, es rechazado y se continúa con el procesamiento de los demás pagos.
8. Se invoca al evento de liquidación, encargado de reagrupar en subconjuntos de 10 pagos únicamente aquellos que han superado las etapas de validación.

9. Cada nuevo subconjunto es procesado de manera independiente y ejecuta la transacción backend encargada de preparar y transformar los pagos en el formato requerido para su envío a la transacción de liquidación en el mainframe.
10. Se invoca a la transacción de liquidación en el mainframe, encargada de ejecutar las validaciones finales sobre las cuentas y realizar el cargo agrupado del importe total, así como los abonos correspondientes a cada cuenta destino.
- S1. El evento del supervisor es invocado inicialmente por la primera transacción backend del flujo y, a partir de ese momento, se mantiene activo realizando invocaciones periódicas a la transacción backend del supervisor.
- S2. A través de la transacción backend del supervisor se monitorea de forma continua el estado de los pagos de la planilla, manteniéndose activa mediante reintentos hasta que todos los pagos alcanzan un estado final. Una vez que la planilla ha sido procesada en su totalidad, la ejecución del supervisor finaliza.
- R1. El evento de retorno se activa una vez la transacción backend del supervisor ha finalizado su ejecución, invocando una única vez, para toda la planilla, a la transacción backend de devoluciones.
- R2. A través de la transacción backend de retorno se determina si existen pagos rechazados; de ser así, se continúa con el flujo de invocación al mainframe, y en caso contrario se da por finalizado todo el proceso de streaming.
- R3. Se invoca a la transacción de devolución en el mainframe, encargada de calcular el monto total a devolver a partir de los pagos rechazados y de realizar el abono de devolución correspondiente en la cuenta ordenante.
- T. Todas las transacciones backend mantienen comunicación constante con la base de datos operacional Oracle y la base de datos informacional MongoDB, con el fin de actualizar los estados tanto de los pagos como de la planilla.
- D. Las transacciones del mainframe almacenan y consultan información exclusivamente relacionada con la transacción monetaria, utilizando la base de datos operacional DB2.

- Trazabilidad de Operaciones

Figura 11

Diagrama de Secuencia de Trazabilidad de Operaciones



Nota. Elaboración Propia.

A continuación, se detallan los pasos para consultar la trazabilidad de una planilla o de los pagos enviados por el Cliente, independientemente del canal de entrada.

1. Desde el canal, se invoca al servicio backend, donde se ejecuta una etapa de preproceso en la que se realizan las validaciones de seguridad necesarias para la ejecución del servicio, tales como la autenticación y autorización.
2. Una vez superadas las prevalidaciones a nivel de servicio, se ejecuta la transacción backend encargada de construir las consultas necesarias para obtener la información de la planilla o de los pagos a partir de los filtros de entrada.
3. Como último paso, se realiza la consulta de la trazabilidad de la planilla o de los pagos del cliente en la base de datos informacional MongoDB.

3.2.5.2. Planificación, Ejecución y Despliegue

Durante esta etapa se llevó a cabo el desarrollo integral de la solución, abarcando desde el diseño de los componentes hasta su despliegue en producción, siguiendo el marco de trabajo Scrum y priorizando las funcionalidades definidas por negocio en la etapa de Ideación y Refinamiento. En esta fase se trabajaron los siguientes MVPs:

- MVP1: Servicio de incorporación de transferencias masivas de nóminas (mismo banco)
- MVP2: Servicio de procesamiento near-real-time de transferencias masivas de nóminas (mismo banco)
- MVP3: Servicio de seguimiento y trazabilidad de transferencias masivas de nóminas

Para cada uno de los MVPs definidos, se detallan a continuación su objetivo, alcance y los criterios de aceptación considerados para su implementación.

Tabla 6

MVP 1: Servicio de incorporación de transferencias masivas de nóminas (mismo banco)

Objetivo	Habilitar la incorporación de planillas de nóminas para operaciones entre cuentas del mismo banco, asegurando su recepción y registro en las bases de datos operacional e informacional.
Alcance	Incluye: - Recepción de la nómina independientemente del canal. - Registro de la nómina y sus pagos con un identificador único. - Persistencia de datos mínimos necesarios para habilitar las siguientes etapas del flujo. - Incorporación de planillas únicamente en moneda PEN y USD. No Incluye: - Procesamiento en near-real-time de los pagos. - Ejecución del cargo y abono (transferencias). - Otros productos/tipos de pagos fuera de nómina mismo banco.
Criterios de aceptación	- Al recibir una nómina, el servicio retorna un identificador único de planilla (idPlanilla). - El servicio permite incorporar planillas de hasta 7 000 pagos (sin error por capacidad), manteniendo consistencia en el registro.

-
- La nómina y sus pagos quedan registrados en la base de datos operacional e informacional, asociados al idPlanilla.
-

Nota. Elaboración Propia.

Tabla 7

MVP 2: Servicio de procesamiento near-real-time de transferencias masivas de nóminas (mismo banco)

Objetivo	Procesar y ejecutar en near-real-time las transferencias de nóminas entre cuentas del mismo banco, garantizando consistencia de estados, validaciones técnicas y de producto, y manejo de errores.
Alcance	Incluye: - Procesamiento de la nómina independientemente del canal. - Procesamiento de la información incorporada en el MVP1 (planilla y pagos registrados). - Validaciones de negocio definidas para el producto (límites transaccionales y controles de fraude). - Ejecución del cargo y abono de los pagos (transferencias mismo banco). - Actualización de estados de planilla y pagos durante el procesamiento. No Incluye: - Modificación/edición de la información registrada en la etapa de incorporación. - Validaciones de negocio distintas a las definidas para nómina mismo banco. - Otros productos/tipos de pagos fuera de nómina mismo banco.
Criterios de aceptación	- Dado un idPlanilla y la cuenta ordenante, el servicio inicia el procesamiento en streaming de los pagos. - Antes de procesar los pagos, el servicio valida la firma digital de la planilla; si la validación falla, la planilla se rechaza y no se ejecuta ninguna transferencia asociada. - El servicio procesa en near-real-time planillas de hasta 7 000 pagos, manteniendo un desempeño estable (sin degradación significativa) y garantizando la integridad y consistencia de la información.

Nota. Elaboración Propia.

Tabla 8

MVP 3: Servicio de seguimiento y trazabilidad de transferencias masivas de nóminas

Objetivo	Permitir la consulta de la información general de planillas y pagos de un cliente, así como el estado actual y la trazabilidad de estados por los que ha transitado, incluyendo errores asociados.
Alcance	Incluye: - Servicios de consulta disponibles para todos los canales habilitados. - Detalle/Listado de planillas ingresadas por cliente y detalle/listado de pagos por planilla. - Filtros de búsqueda: identificador del cliente, identificador de planilla, identificador de pago, rango de fechas y estados. - Consulta de historial de estados de planilla y pagos, incluyendo registro de errores/motivos cuando aplique. No Incluye: - Filtros adicionales a los definidos. - Edición/modificación de planillas o pagos.
Criterios de aceptación	- Dado un conjunto de filtros soportados, el servicio retorna la información correspondiente de planillas y/o pagos, según aplique. - Para cada planilla y pago consultado, se retorna el estado actual y el historial de estados (trazabilidad E2E), incluyendo fecha/hora y motivo de error cuando exista. - La información consultada se obtiene desde la base de datos informacional y refleja el último estado registrado del proceso (consistencia con lo registrado en el flujo).

Nota. Elaboración Propia.

A partir de los MVPs priorizados, se definió un roadmap a alto nivel, considerando los lineamientos de desarrollo de la entidad basados en quarters, los cuales comprenden un periodo de seis sprints de dos semanas de duración, además de un sprint 0 inicial de una semana.

Esto se desglosa en features con entregables definidos, los cuales se desarrollan a lo largo del quarter, y a su vez en historias de usuario de distintos tipos, tales como: habilitadoras [EN], funcionales [FUN], de integración [INT], pruebas funcionales [QA] y pase a producción [REL], que se ejecutan sprint a sprint.

Asimismo, la estimación de recursos para proyectos se realiza por quarter en la etapa previa al inicio del Sprint 0, momento en el cual el Scrum Master define el staffing

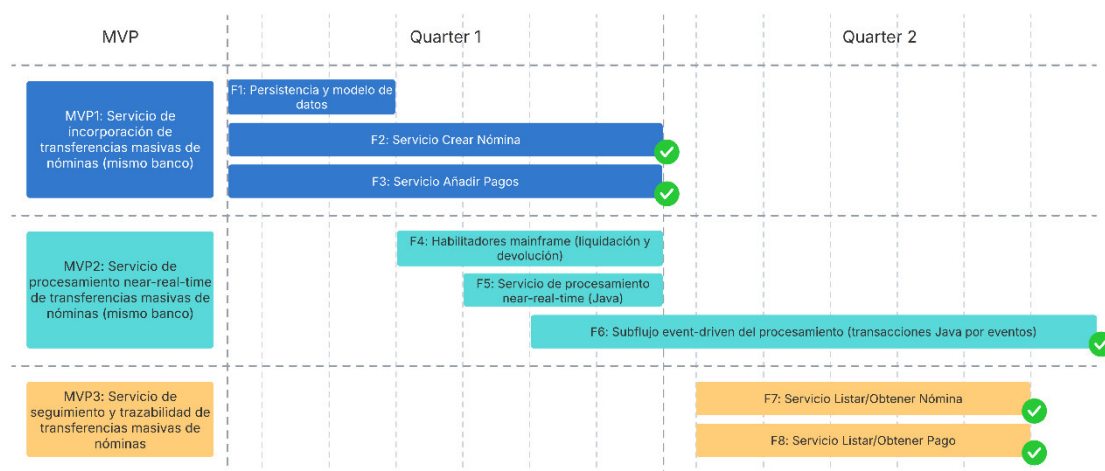
del equipo Scrum.

En base a ello, para este proyecto, que contempla tres MVPs, se estableció una duración de dos quarters. En el primer quarter se desarrolló el MVP1 y parte del MVP2, mientras que en el segundo se dio continuidad al MVP2 y se desarrolló el MVP3.

A continuación, se presenta el roadmap a alto nivel definido para el proyecto.

Figura 12

Roadmap a alto nivel del proyecto



Nota. Elaboración Propia.

Planificación Quarter 1 (Q1)

Durante el primer quarter del proyecto se tuvo como objetivo desarrollar y desplegar en producción el MVP1, así como iniciar el desarrollo del MVP2. Para ello se definieron los siguientes features por MVP a trabajar durante el Q1:

MVP1. Servicio de incorporación de transferencias masivas de nóminas (mismo banco)

- F1: Persistencia y modelo de datos
- F2: Servicio Crear Nómina
- F3: Servicio Añadir Pagos

MVP2: Servicio de procesamiento near-real-time de transferencias masivas de nóminas (mismo banco)

- F4: Habilitadores mainframe (liquidación y devolución)
- F5: Servicio de procesamiento near-real-time (Java)
- F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)

En cuanto al staffing de los recursos humanos, el Scrum Master definió los perfiles necesarios para la ejecución de los features establecidos para el Q1, considerando que cada recurso se asigna durante toda la duración del quarter.

- 1.0 Scrum Master / Tech Lead
- 3.0 Backend Java Developers
- 1.0 COBOL Developer
- 0.5 Backend Event / Stream Developer
- 0.5 QA
- 0.3 Solution Architect
- 0.3 Data Architect
- 0.3 Backend Java Architect
- 0.3 Event / Stream Architect

En cuanto a los demás recursos tecnológicos, estos no se incluyen dentro de la planificación, ya que se reutilizarán los recursos de infraestructura ya existentes en el banco, lo que no implica la adquisición de nuevas licencias, equipos u otros activos.

Ejecución Quarter 1 (Q1)

- Q1 – Sprint 0

Objetivo del Sprint: Definir el modelo conceptual/lógico de las persistencias de datos y el diseño inicial de los servicios backend de creación de nóminas y adición de pagos.

Tabla 9

Q1 – Sprint 0

Feature	Historia de Usuario (HU)	Entregable
F1: Persistencia y modelo de datos	F1-HU01 [EN]: Persistencia en BD Operacional (Oracle)	Modelo conceptual/lógico Oracle
	F1-HU02 [EN]: Persistencia en BD Informacional (MongoDB)	Modelo conceptual/lógico MongoDB
F2: Servicio Crear Nómina	F2-HU01 [FUN]: Servicio y transacción backend Java - Crear Nómina	Diseño de servicio backend - Crear Nómina
F3: Servicio Añadir Pagos	F3-HU01 [FUN]: Servicio y transacción backend Java - Añadir Pagos	Diseño de servicio backend - Añadir Pagos

Nota. Elaboración Propia.

Durante el Sprint 0 se dio inicio a las features asociadas el MVP1, comenzando con la definición de los modelos de datos y el diseño inicial de los servicios que serán consumidos por los canales. Debido a la corta duración de este sprint, no fue posible iniciar el diseño de las transacciones backend; sin embargo, se logró un avance significativo en el diseño lógico de las capas de persistencia.

- Q1 – Sprint 1

Objetivo del Sprint: Establecer el modelo de datos final de las bases de datos operacional e informacional, así como el diseño final de los servicios y transacciones backend para la creación de nóminas y la adición de pagos, dejando la solución lista para iniciar la construcción de los componentes.

Tabla 10

Q1 – Sprint 1

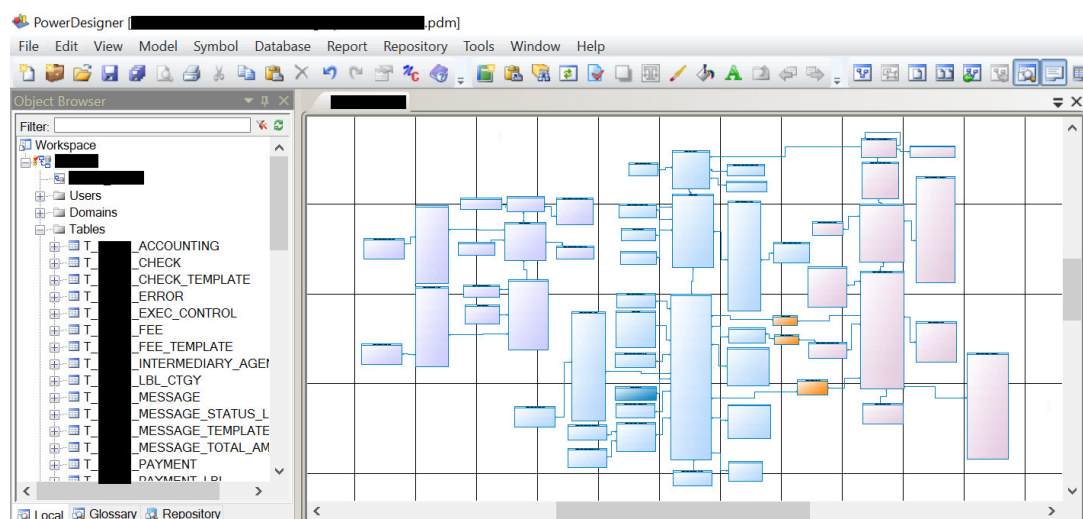
Feature	Historia de Usuario (HU)	Entregable
F1: Persistencia y modelo de datos	F1-HU01 [EN]: Persistencia en BD Operacional (Oracle)	Modelo físico final Oracle
	F1-HU02 [EN]: Persistencia en BD Informacional (MongoDB)	Modelo final MongoDB
F2: Servicio Crear Nómina	F2-HU01 [FUN]: Servicio y transacción backend Java - Crear Nómina	Diseño de transacción backend - Crear Nómina
F3: Servicio Añadir Pagos	F3-HU01 [FUN]: Servicio y transacción backend Java - Añadir Pagos	Diseño de transacción backend - Añadir Pagos

Nota. Elaboración Propia.

Durante el Sprint 1 se finalizó el diseño de los modelos de datos, los cuales fueron enviados para aprobación por parte del equipo de arquitectura de datos. Asimismo, se culminó el diseño de los servicios y transacciones correspondientes a los flujos de creación de nóminas y adición de pagos. Con ello se dio por concluida la etapa de diseño del MVP1, quedando abierta la posibilidad de realizar adecuaciones durante la etapa de construcción.

Figura 13

Modelo BD Operacional Oracle



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

- Q1 – Sprint 2

Objetivo del Sprint: Realizar el despliegue de la capa de persistencia en ambientes previos (Oracle y MongoDB) y completar la construcción de los servicios y transacciones backend de creación de nóminas y adición de pagos, dejando los componentes listos para su integración.

Tabla 11

Q1 – Sprint 2

Feature	Historia de Usuario (HU)	Entregable
F1: Persistencia y modelo de datos	F1-HU01 [EN]: Persistencia en BD Operacional (Oracle)	Scripts (DDL/DML) + despliegue en ambientes previos
	F1-HU02 [EN]: Persistencia en BD Informacional (MongoDB)	Scripts/artefactos de despliegue + despliegue en ambientes previos.
F2: Servicio Crear Nómina	F2-HU01 [FUN]: Servicio y transacción backend Java - Crear Nómina	Servicio y transacción backend implementado + pruebas unitarias
F3: Servicio Añadir Pagos	F3-HU01 [FUN]: Servicio y transacción backend Java - Añadir Pagos	Servicio y transacción backend implementado + pruebas unitarias

Nota. Elaboración Propia.

Durante el Sprint 2 se desplegaron las bases de datos en los ambientes previos, dando por finalizada la feature habilitadora de persistencia y modelo de datos. Asimismo, se concluyó la construcción de los servicios y transacciones correspondientes a los flujos de incorporación de nóminas, añadiendo algunos ajustes respecto al diseño inicial para cubrir completamente la necesidad y asegurar un almacenamiento consistentemente de la información en la base de datos.

Figura 14

Script de creación de índice BD Informativa MongoDB

```
#!/bin/bash
#
# Script generado desde PWD para creación de INDICE
#
# HELP:
#
# WARNING:
# Si el índice ya existe y se especifica el flag "-d", será borrado con todos sus datos
#
# VARIABLES:
HOSTNAME=$1
USER=$2
PASSWORD=$3
INDICE=$4
shards=$5
replicas=$6
seconds=$7
ANYO=$8

ALIAS=

# Llamada CURL:
curl -X PUT -H'Content-Type: application/json' -u $USER:$PASSWORD 'https://'$HOSTNAME/'$INDICE' -d '{
  "settings": {
    "number_of_shards": '$shards',
    "number_of_replicas": '$replicas',
    "refresh_interval" : "'$seconds'",
    "analysis": {
      "analyzer": {
        "analyzer" : {
          "tokenizer": "standard",
          "filter" : ["lowercase","asciifolding"]
        }
      }
    },
    "index.search.slowlog.threshold.query.warn": "200ms",
    "index.search.slowlog.threshold.query.trace": "100ms",
    "index.search.slowlog.threshold.fetch.warn": "200ms",
    "index.search.slowlog.threshold.fetch.trace": "100ms",
  },
}
```

Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

- Q1 – Sprint 3

Objetivo del Sprint: Integrar los componentes backend desarrollados (servicio, transacción y base de datos) para habilitar el flujo operativo de creación de nóminas y adición de pagos, y definir el diseño de la transacción CICS de liquidación como habilitador para las etapas posteriores.

Tabla 12

Q1 – Sprint 3

Feature	Historia de Usuario (HU)	Entregable
F2: Servicio Crear Nómina	F2-HU02 [INT]: Integración de componentes (Servicio + Transacción + BD)	Componentes integrados del flujo (Servicio + Transacción + BD)
F3: Servicio Añadir Pagos	F3-HU02 [INT]: Integración de componentes (Servicio + Transacción + BD)	Componentes integrados del flujo (Servicio + Transacción + BD)

F4: Habilitadores mainframe (liquidación y devolución)	F4-HU01 [EN]: Transacción CICS de Liquidación	Diseño de transacción mainframe - Liquidación
--	---	---

Nota. Elaboración Propia.

Durante el Sprint 3 se realizó la integración de los distintos componentes backend desarrollados, con el objetivo de completar los flujos de incorporación E2E como parte del MVP1. Por otro lado, una vez definido y desplegado el modelo de datos, se inició la implementación del MVP2, comenzando con el diseño de la transacción CICS de liquidación como parte de la feature habilitadora de implementación de componentes en el mainframe.

- Q1 – Sprint 4

Objetivo del Sprint: Ejecutar las pruebas técnicas y de integración finales de los servicios de incorporación de nóminas. Asimismo, culminar con la implementación de la transacción CICS de liquidación y el diseño de la transacción CICS de devolución.

Tabla 13

Q1 – Sprint 4

Feature	Historia de Usuario (HU)	Entregable
F2: Servicio Crear Nómina	F2-HU03 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración
	F2-HU04 [REF]: Refinamiento y adecuaciones	Ajustes incorporados sobre hallazgos (bugs/mejoras)
	F2-HU05 [QA]: Pruebas funcionales del servicio	Test Plan aprobado + casos de prueba definidos
F3: Servicio Añadir Pagos	F3-HU03 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración
	F3-HU04 [REF]: Refinamiento y adecuaciones	Ajustes incorporados sobre hallazgos (bugs/mejoras)
	F3-HU05 [QA]: Pruebas	Test Plan aprobado + casos de

	funcionales del servicio	prueba definidos
F4: Habilitadores mainframe (liquidación y devolución)	F4-HU01 [EN]: Transacción CICs de Liquidación	Transacción mainframe implementada + pruebas unitarias
	F4-HU02 [EN]: Transacción CICs de Devolución	Diseño de transacción mainframe - Devolución
F5: Servicio de procesamiento near-real-time (Java)	F5-HU01 [FUN]: Servicio y transacción backend Java - Liquidar Nómina	Diseño de servicio y transacción backend - Liquidar Nómina

Nota. Elaboración Propia.

Durante el Sprint 4 se realizaron las pruebas técnicas y de integración de los servicios de incorporación de nóminas, identificándose algunos bugs que fueron corregidos de manera oportuna con el fin de minimizar errores en las pruebas funcionales. Asimismo, se inició la preparación del plan de pruebas del MVP1, como paso previo a la etapa de gestión de pase a producción. Por otro lado, se culminó la implementación de la versión inicial de la transacción CICS de liquidación, la cual fue refinada posteriormente durante su integración con los demás componentes backend.

- Q1 – Sprint 5

Objetivo del Sprint: Ejecutar las pruebas funcionales de los servicios de incorporación de nóminas y culminar con la implementación de la transacción CICS de devolución, así como del servicio y la transacción backend asociados a la liquidación de nóminas.

Tabla 14

Q1 – Sprint 5

Feature	Historia de Usuario (HU)	Entregable
F2: Servicio Crear Nómina	F2-HU05 [QA]: Pruebas funcionales del servicio	Ejecución de casos + evidencias + reporte de resultados
	F2-HU06 [REL]: Pase a producción de componentes	Checklist / Documentación de pase a producción

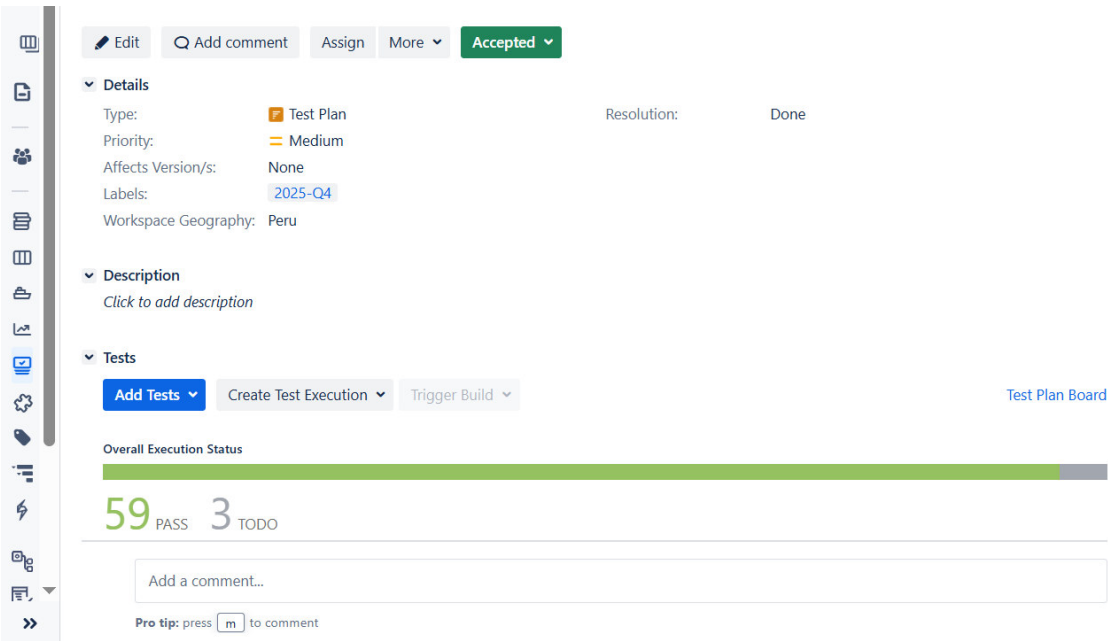
F3: Servicio Añadir Pagos	F3-HU05 [QA]: Pruebas funcionales del servicio	Ejecución de casos + evidencias + reporte de resultados
	F3-HU06 [REL]: Pase a producción de componentes	Checklist / Documentación de pase a producción
F4: Habilitadores mainframe (liquidación y devolución)	F4-HU02 [EN]: Transacción CICs de Devolución	Transacción mainframe implementada + pruebas unitarias
F5: Servicio de procesamiento near-real-time (Java)	F5-HU01 [FUN]: Servicio y transacción backend Java - Liquidar Nómina	Servicio y transacción backend implementado + pruebas unitarias
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU01 [FUN]: Transacciones backend Java - Paginación / Validaciones / Liquidación (flujo por eventos)	Diseño de transacciones backend - Flujo por eventos
	F6-HU03 [FUN]: Stream/Eventos - Paginación / Validaciones / Liquidación	Diseño de streams y suscriptores

Nota. Elaboración Propia.

Durante el Sprint 5 se realizaron las pruebas funcionales del MVP1 durante las cuales el equipo de QA identificó algunos bugs que fueron corregidos oportunamente por el equipo de desarrollo, sin impactar el avance de las pruebas. Además, se culminó la implementación de la transacción CICS de devolución, así como del servicio y transacción backend de liquidación de pagos inicial invocado por el canal. Estos componentes fueron desarrollados de manera independiente y, hasta ese momento, solo habían sido validados mediante pruebas unitarias, sin haberse integrado aún entre sí.

Figura 15

Registro, planificación y seguimiento del Test Plan en Jira



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

- Q1 – Sprint 6

Objetivo del Sprint: Desplegar a producción todos los componentes backend del flujo de incorporación de nóminas y finalizar las pruebas técnicas de las transacciones CICS de liquidación y devolución.

Tabla 15

Q1 – Sprint 6

Feature	Historia de Usuario (HU)	Entregable
F2: Servicio Crear Nómina	F2-HU06 [REL]: Pase a producción de componentes	Despliegue productivo
F3: Servicio Añadir Pagos	F3-HU06 [REL]: Pase a producción de componentes	Despliegue productivo
F4: Habilitadores mainframe (liquidación y devolución)	F4-HU03 [INT]: Pruebas técnicas de transacción	Evidencias / reporte de pruebas técnicas
F5: Servicio de procesamiento	F5-HU02 [INT]: Integración de componentes (Servicio +	Componentes integrados del flujo (Servicio + Transacción)

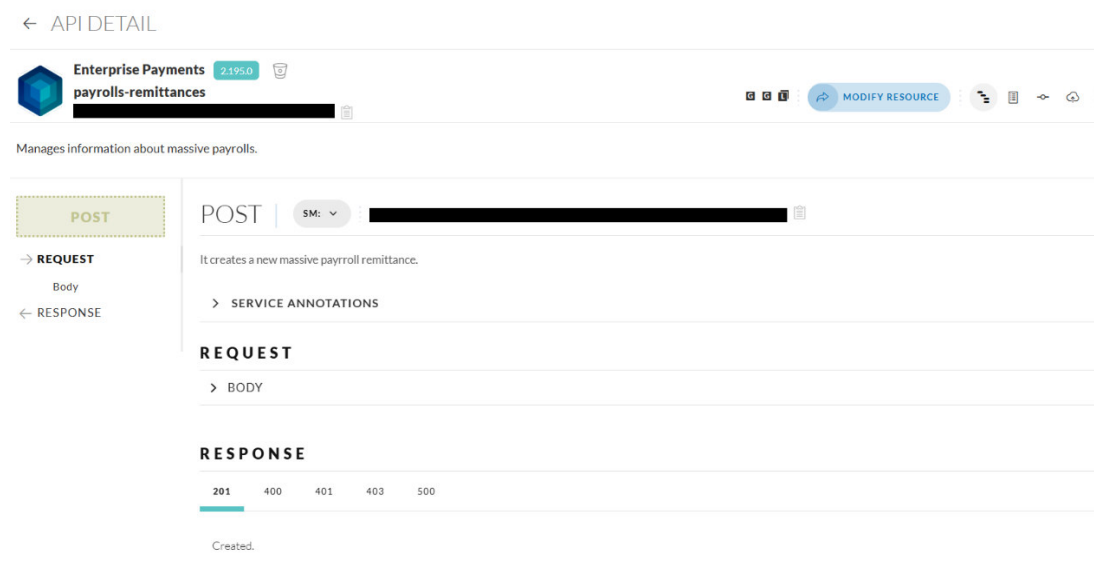
near-real-time (Java)	Transacción)	
	F5-HU03 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU01 [FUN]: Transacciones backend Java - Paginación / Validaciones / Liquidación (flujo por eventos)	Transacciones backend implementadas + pruebas unitarias
	F6-HU03 [FUN]: Stream/Eventos - Paginación / Validaciones / Liquidación	Scripts/artefactos de creación de streams y suscriptores + despliegue en ambientes previos

Nota. Elaboración Propia.

Durante el último sprint del Q1 se logró desplegar a producción los servicios de incorporación de nóminas, logrando finalizar de esta manera el MVP1. Por otro lado, también se logró integrar y realizar las pruebas técnicas de las transacciones CICS y de los servicios y transacciones backend java del flujo de liquidación y procesamiento de pagos, cerrando de esta manera los dos primeros features del MVP2.

Figura 16

APIs de los servicios de Incorporación de Nóminas



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

Planificación Quarter 2 (Q2)

Durante el segundo quarter del proyecto se tuvo como objetivo culminar el desarrollo de los MVP2 y MVP3, llevando ambos a producción y dando por finalizado el proyecto. Para ello, se definieron los siguientes features por MVP a trabajar durante el Q2:

MVP2: Servicio de procesamiento near-real-time de transferencias masivas de nóminas (mismo banco)

- F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)

MVP3: Servicio de seguimiento y trazabilidad de transferencias masivas de nóminas

- F7: Servicio Listar/Obtener Nómina
- F8: Servicio Listar/Obtener Pago

En cuanto al staffing de los recursos humanos, se presentaron variaciones respecto a lo definido en el Q1, ya que algunas features requirieron ajustes en el nivel de esfuerzo de determinados perfiles. Con base en ello, se asignó el siguiente personal para la ejecución del Q2:

- 1.0 Scrum Master / Tech Lead
- 3.0 Backend Java Developers
- 0.5 COBOL Developer
- 1.0 Backend Event / Stream Developer
- 1.0 QA
- 0.3 Solution Architect
- 0.3 Backend Java Architect
- 0.3 Event / Stream Architect

En cuanto a los demás recursos tecnológicos, al igual que en el Q1, estos no se incluyen dentro de la planificación, ya que se reutilizarán los recursos de infraestructura ya existentes en el banco, lo que no implica la adquisición de nuevas licencias, equipos u otros activos.

Ejecución Quarter 2 (Q2)

- Q2 – Sprint 0

Objetivo del Sprint: Definir el diseño de las transacciones backend y de los eventos asociados a los flujos de Supervisor y Retorno.

Tabla 16

Q2 – Sprint 0

Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU02 [FUN]: Transacciones backend Java - Supervisor y Retorno (flujo por eventos)	Diseño de transacciones backend - Flujo por eventos
	F6-HU04 [FUN]: Stream/Eventos - Supervisor y Retorno	Diseño de streams y suscriptores

Nota. Elaboración Propia.

Durante el Sprint 0 se continuó con el desarrollo del MVP2, concentrándose principalmente en la definición del diseño de las transacciones backend y de los eventos de Supervisor y Retorno. No fue posible iniciar la ejecución del MVP3 debido a la corta duración de este sprint.

- Q2 – Sprint 1

Objetivo del Sprint: Culminar el desarrollo de las transacciones backend y de los eventos asociados a los flujos de Supervisor y Retorno, así como iniciar el diseño de los servicios y transacciones backend del flujo de trazabilidad.

Tabla 17

Q2 – Sprint 1

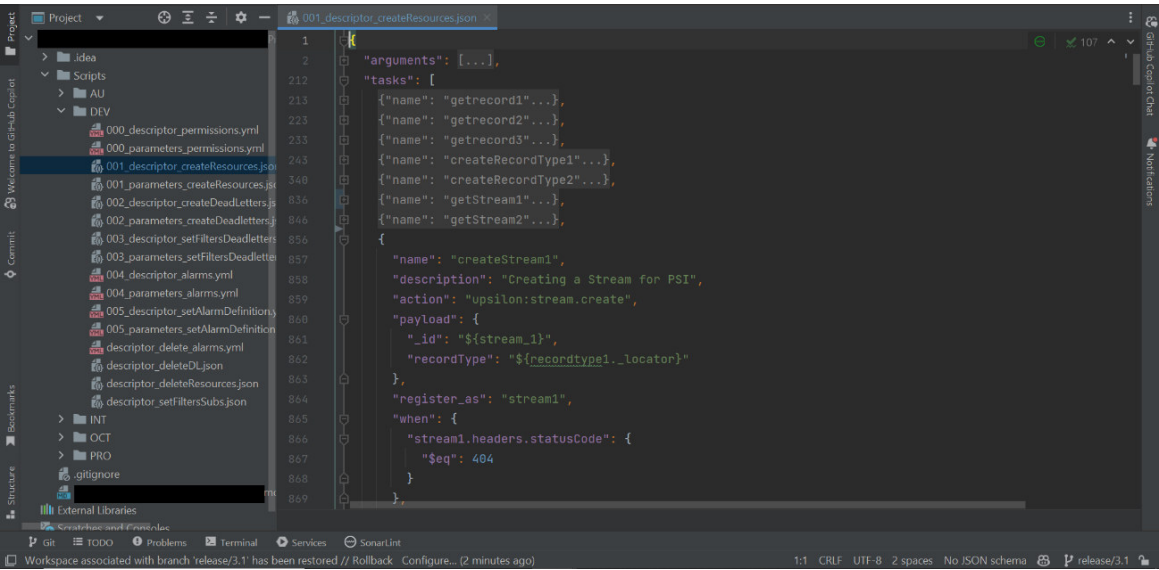
Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU02 [FUN]: Transacciones backend Java - Supervisor y Retorno (flujo por eventos)	Transacciones backend implementadas + pruebas unitarias
	F6-HU04 [FUN]: Stream/Eventos - Supervisor y Retorno	Scripts/artefactos de creación de streams y suscriptores + despliegue en ambientes previos
F7: Servicio Listar/Obtener Nómina	F7-HU01 [FUN]: Servicio y transacción backend Java - Listar/Obtener Nómina	Diseño de servicio y transacción backend - Listar/Obtener Nómina
F8: Servicio Listar/Obtener Pago	F8-HU01 [FUN]: Servicio y transacción backend Java - Listar/Obtener Pago	Diseño de servicio y transacción backend - Listar/Obtener Pago

Nota. Elaboración Propia.

Durante el Sprint 1 se finalizó el desarrollo de las transacciones y eventos del flujo de streaming; sin embargo, debido a algunas incompatibilidades entre los componentes, no fue posible realizar pruebas integrales, quedando pendientes algunas adecuaciones adicionales para asegurar la correcta ejecución del flujo completo. En cuanto al MVP3 de trazabilidad, se dio inicio al diseño de los servicios y transacciones backend, considerando los filtros y la información definidos dentro del alcance.

Figura 17

Script de creación y configuración de streams



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

- Q2 – Sprint 2

Objetivo del Sprint: Adecuar y homologar los componentes del flujo de streaming, así como construir los servicios y transacciones backend del flujo de trazabilidad.

Tabla 18

Q2 – Sprint 2

Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU05 [INT]: Integración de componentes (Servicio + Transacción + Eventos/Mainframe)	Componentes integrados del flujo (Servicio + Transacción + Eventos/Mainframe)
F7: Servicio Listar/Obtener Nómina	F7-HU01 [FUN]: Servicio y transacción backend Java - Listar/Obtener Nómina	Servicio y transacción backend implementado + pruebas unitarias
	F7-HU04 [QA]: Pruebas funcionales del servicio	Test Plan aprobado + casos de prueba definidos
F8: Servicio Listar/Obtener Pago	F8-HU01 [FUN]: Servicio y transacción backend Java - Listar/Obtener Pago	Servicio y transacción backend implementado + pruebas unitarias

F8-HU04 [QA]: Pruebas funcionales del servicio	Test Plan aprobado + casos de prueba definidos
--	--

Nota. Elaboración Propia.

Durante el Sprint 2 se realizaron las adecuaciones necesarias para integrar los componentes del flujo de liquidación en streaming, quedando pendiente la ejecución de pruebas técnicas más exhaustivas para asegurar su correcto funcionamiento. Por otro lado, en el flujo de trazabilidad se culminó la construcción de los servicios de consulta y se avanzó con la elaboración del test plan, con el objetivo de evitar el traslape de actividades de prueba con el flujo de liquidación.

- Q2 – Sprint 3

Objetivo del Sprint: Realizar pruebas técnicas y de integración tanto del servicio de liquidación como de los servicios de trazabilidad.

Tabla 19

Q2 – Sprint 3

Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU06 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración
	F6-HU07 [REF]: Refinamiento y adecuaciones	Ajustes incorporados sobre hallazgos (bugs/mejoras)
	F6-HU08 [QA]: Pruebas funcionales del servicio	Test Plan aprobado + casos de prueba definidos
F7: Servicio Listar/Obtener Nómina	F7-HU02 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración
	F7-HU03 [REF]: Refinamiento y adecuaciones	Ajustes incorporados sobre hallazgos (bugs/mejoras)
F8: Servicio Listar/Obtener Pago	F8-HU02 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración

F8-HU03 [REF]: Refinamiento y adecuaciones	Ajustes incorporados sobre hallazgos (bugs/mejoras)
--	---

Nota. Elaboración Propia.

Durante el Sprint 3 se realizaron las pruebas técnicas de los servicios de liquidación y de trazabilidad. En el caso del servicio de liquidación, se identificaron algunos bugs que no pudieron ser resueltos de manera inmediata debido a la complejidad del flujo y a la necesidad de un mayor análisis, lo que impidió culminar las pruebas de dicho flujo en este sprint. Por otro lado, en los servicios de trazabilidad, si bien se detectaron incidencias durante las pruebas, estas fueron corregidas oportunamente, permitiendo completar satisfactoriamente las pruebas técnicas correspondientes.

- Q2 – Sprint 4

Objetivo del Sprint: Culminar las adecuaciones y las pruebas técnicas del flujo de liquidación, y realizar las pruebas funcionales de los servicios de trazabilidad.

Tabla 20

Q2 – Sprint 4

Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU06 [INT]: Pruebas técnicas y de integración de servicio	Evidencias / reporte de pruebas técnicas e integración
	F6-HU07 [REF]: Refinamiento y adecuaciones	Ajustes incorporados sobre hallazgos (bugs/mejoras)
F7: Servicio Listar/Obtener Nómina	F7-HU04 [QA]: Pruebas funcionales del servicio	Ejecución de casos + evidencias + reporte de resultados
	F7-HU05 [REL]: Pase a producción de componentes	Checklist / Documentación de pase a producción
F8: Servicio Listar/Obtener Pago	F8-HU04 [QA]: Pruebas funcionales del servicio	Ejecución de casos + evidencias + reporte de resultados
	F8-HU05 [REL]: Pase a producción de componentes	Checklist / Documentación de pase a producción

Nota. Elaboración Propia.

Durante el Sprint 4 se realizaron las correcciones de las incidencias del flujo de liquidación identificadas en el sprint anterior, lo que implicó algunos ajustes en el diseño de las transacciones. No obstante, se logró finalizar satisfactoriamente las pruebas técnicas de dicho flujo. Asimismo, se ejecutaron las pruebas funcionales del MVP3 de trazabilidad y se completó la documentación necesaria para gestionar el pase a producción de este flujo.

- Q2 – Sprint 5

Objetivo del Sprint: Desplegar en producción los servicios del flujo de trazabilidad e iniciar las pruebas funcionales del flujo de liquidación, así como las gestiones necesarias para su pase a producción.

Tabla 21

Q2 – Sprint 5

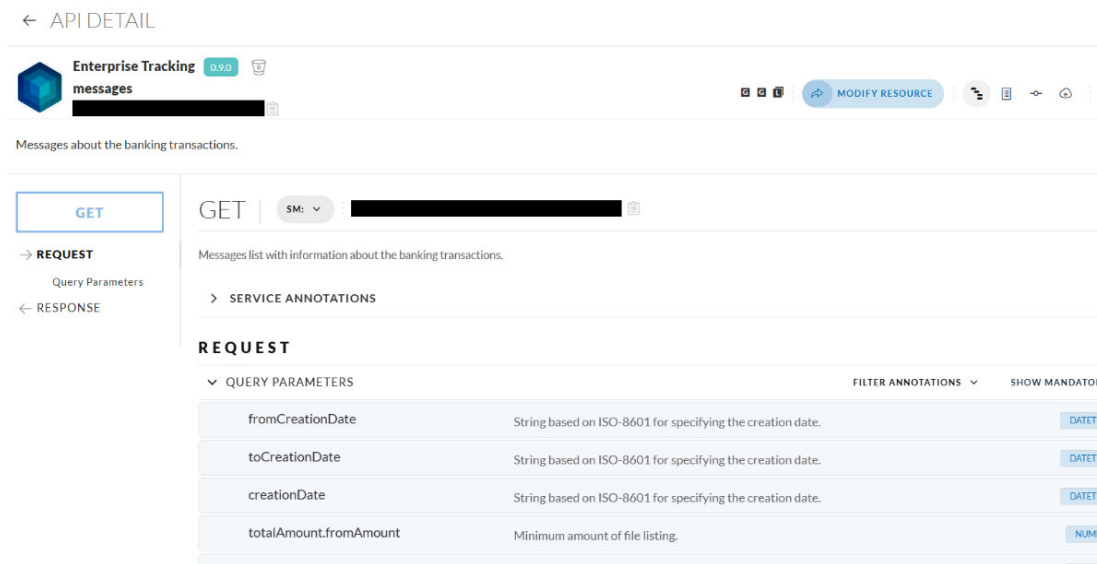
Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU08 [QA]: Pruebas funcionales del servicio	Ejecución de casos + evidencias + reporte de resultados
	F6-HU09 [REL]: Pase a producción de componentes	Checklist / Documentación de pase a producción
F7: Servicio Listar/Obtener Nómina	F7-HU05 [REL]: Pase a producción de componentes	Despliegue productivo
F8: Servicio Listar/Obtener Pago	F8-HU05 [REL]: Pase a producción de componentes	Despliegue productivo

Nota. Elaboración Propia.

Durante el Sprint 5, una vez finalizadas las pruebas funcionales de los flujos de trazabilidad y completada la documentación requerida, se logró el pase a producción de los servicios consultivos de trazabilidad, dando por concluida la ejecución del MVP3. Por el lado del MVP2, tras resolver las incidencias detectadas durante las pruebas técnicas, se inició la fase de pruebas funcionales y la elaboración de la documentación de pase a producción; sin embargo, se presentaron inconvenientes relacionados con el reporte de contabilidad, lo que requirió realizar ajustes en las transacciones CICS y ocasionó un

retraso en el cierre de las pruebas funcionales.

Figura 18
APIs de los servicios de Trazabilidad



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

- **Q2 – Sprint 6**

Objetivo del Sprint: Desplegar en producción los componentes del flujo de liquidación.

Tabla 22
Q2 – Sprint 6

Feature	Historia de Usuario (HU)	Entregable
F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU08 [QA]: Pruebas funcionales del servicio	Ejecución de casos + evidencias + reporte de resultados
	F6-HU09 [REL]: Pase a producción de componentes	Despliegue productivo

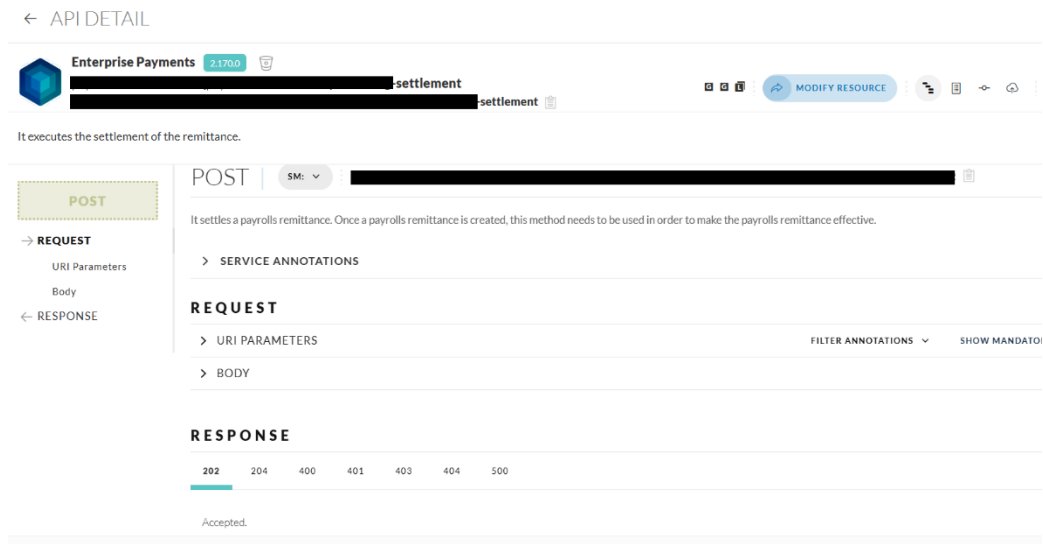
Nota. Elaboración Propia.

Durante el último sprint del proyecto se concentraron los esfuerzos en culminar las pruebas funcionales del flujo de liquidación, las cuales se habían visto retrasadas por las incidencias detectadas y eran un requisito para el pase a producción. Estas pruebas se completaron con éxito, permitiendo finalmente el despliegue en producción del último

flujo pendiente del proyecto.

Figura 19

APIs de los servicios de Liquidación de Nóminas



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

En el Anexo 1 se presenta un resumen detallado de los MVPs trabajados, incluyendo sus respectivos features, historias de usuario, sprints de ejecución y entregables, consolidando la información descrita a lo largo de las secciones anteriores.

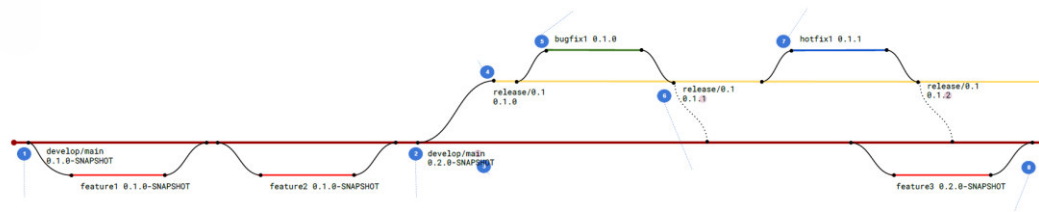
Flujo de Trabajo de Integración y Entrega Continua de Software (CI/CD)

La entidad cuenta con un flujo de integración y entrega continua (CI/CD) para la implementación de componentes de software, el cual fue utilizado en este proyecto con el objetivo de facilitar el desarrollo, asegurar la calidad de software y permitir la detección temprana de errores. A continuación, se describen los pasos que conforman este flujo de trabajo:

1. Se crea el recurso de software desde la infraestructura cloud del banco, proceso que se encuentra automatizado para generar de manera predeterminada el repositorio correspondiente y la rama develop.
2. Una vez generado el recurso, se crean las ramas de trabajo correspondientes dentro del repositorio, las cuales varían según el tipo de cambio a realizar, como feature, bugfix o hotfix.

Figura 20

Modelo de gestión de ramas de trabajo

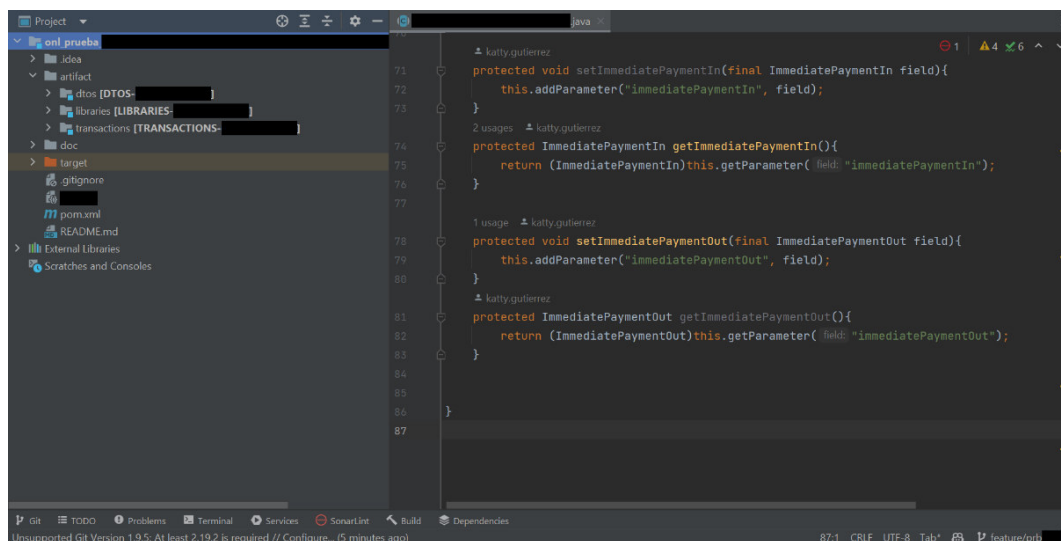


Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

3. El desarrollo del componente y la construcción de las pruebas unitarias se realizan de manera local por parte del desarrollador.

Figura 21

Modelo de estructura de repositorio para componentes backend



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

4. Posteriormente, los cambios se envían al repositorio remoto, el cual se encuentra integrado con un pipeline de Jenkins. Este pipeline ejecuta una serie de etapas que incluyen la compilación y generación del artefacto, la ejecución de pruebas unitarias y el análisis de calidad del código mediante SonarQube, considerando los estándares definidos por la entidad. Todas las etapas deben completarse de manera satisfactoria.

Figura 22

Pipeline de integración continua en Jenkins

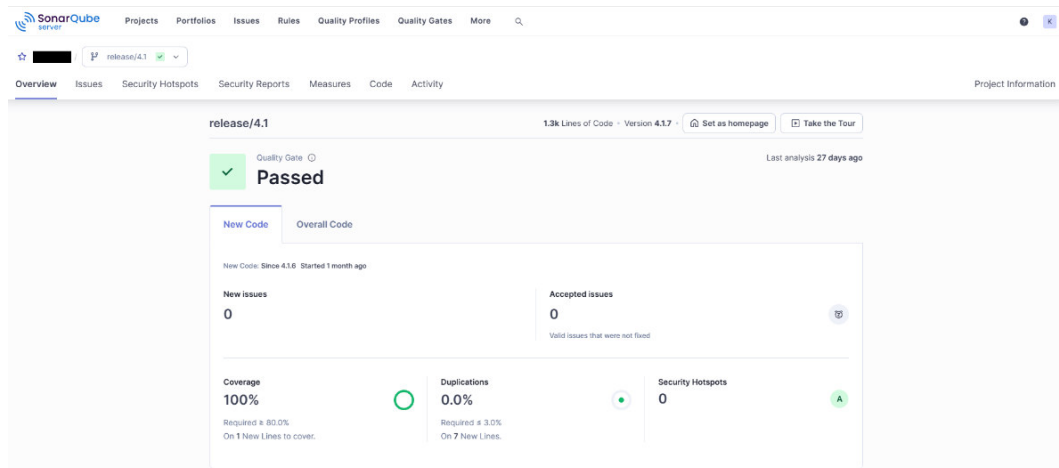
Stage View



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

Figura 23

Análisis de calidad de código mediante SonarQube



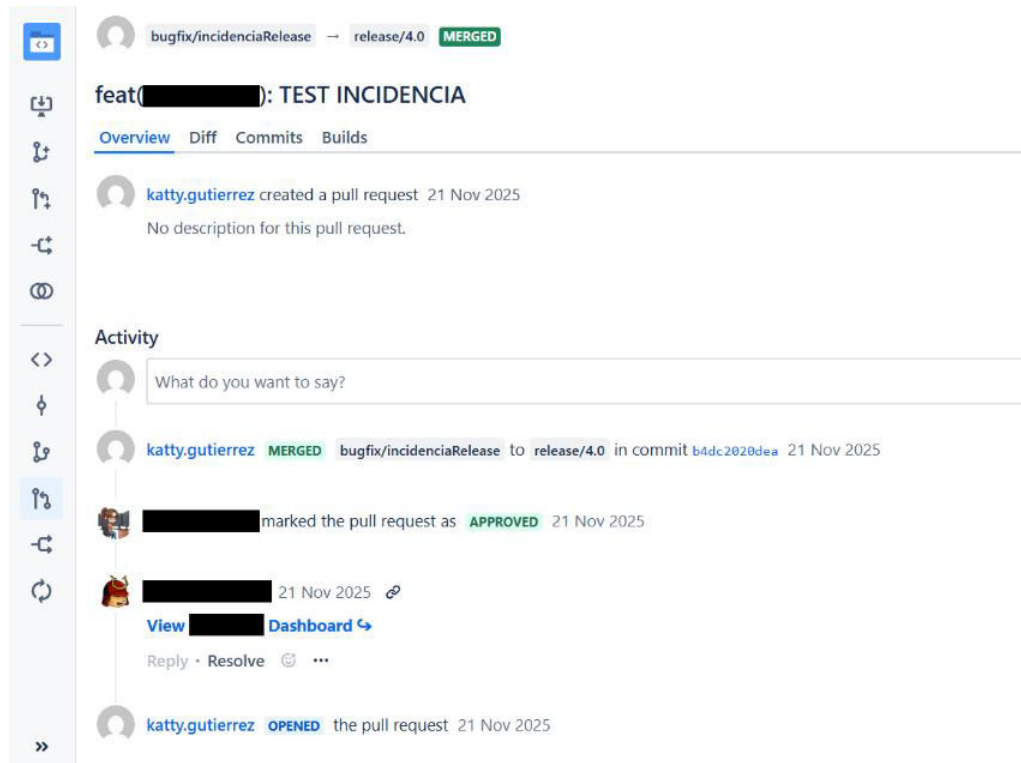
Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

- Si el pipeline de una rama feature, bugfix o hotfix finaliza correctamente, el componente puede desplegarse desde la consola cloud de la entidad hacia el ambiente de Integración, donde el desarrollador realiza las pruebas técnicas y de integración apuntando al endpoint correspondiente.
- Para desplegar en el ambiente de Calidad, es necesario contar con una versión generada desde una rama release. No es posible desplegar directamente desde una rama de trabajo, por lo que se debe crear una pull request desde la rama feature, bugfix o hotfix hacia develop o hacia una rama release existente, según corresponda. Esta solicitud pasa por un proceso de aprobación a cargo de los Release Managers del producto. Una vez aprobado el código y realizado el merge,

el pipeline de Jenkins se ejecuta nuevamente y, si finaliza correctamente, se genera la versión de despliegue.

Figura 24

Proceso de integración de cambios a través de pull request



Nota. Recorte de captura de pantalla; adaptada y anonimizada para fines académicos.

7. Con la versión de despliegue disponible, se procede al despliegue en el ambiente de Calidad, donde el equipo de QA ejecuta las pruebas funcionales apuntando al endpoint de dicho ambiente.
8. Finalmente, el despliegue en el ambiente de Producción se gestiona también desde la consola cloud, pero requiere una programación previa en una fecha determinada y la aprobación del Release Manager y del equipo de gestión de cambios de la entidad. Para ello, debe completarse toda la documentación requerida y haberse finalizado satisfactoriamente las pruebas funcionales.

3.3. Evaluación

3.3.1. Evaluación Económica / Evaluación Costo-Beneficio

Para la evaluación económica del proyecto se realizó un análisis integral orientado a contrastar la inversión requerida con el valor generado para la entidad. Este análisis considera tanto los costos asociados a la implementación de la solución como los beneficios derivados de su puesta en operación, permitiendo evaluar su impacto económico y operativo. Asimismo, se analizan los beneficios obtenidos, diferenciando aquellos de carácter cuantitativo y cualitativo.

3.3.1.1. Costos del proyecto

El costo total del proyecto se determinó considerando el costo del personal interno asignado durante la ejecución, los costos indirectos asociados a infraestructura, herramientas y servicios de soporte, así como los costos operativos posteriores a la puesta en producción. No se incluyeron los costos correspondientes a la etapa de Ideación y Refinamiento, dado que esta fase se abordó mediante apoyos puntuales y no requirió una dedicación continua.

Costo del personal interno

El costo del personal interno se estimó en función de la dedicación de los perfiles asignados durante la etapa de ejecución, valorizando el esfuerzo según su equivalencia de tiempo completo (FTE). La estimación se presenta por quarter, considerando cada quarter como un periodo de tres meses; de este modo, el costo por quarter se calculó a partir del costo mensual referencial de cada perfil, multiplicado por tres meses y ajustado según el FTE promedio asignado.

Figura 25

Costo del personal interno (S/)

Perfil	Dedicación (FTE)	Costo mensual (S/)	Costo por quarter (S/) (= mensual×3)	Costo total (S/) (= FTE×costo Q)
Q1				
Product Owner	0.5	S/.8,000	S/.24,000	S/.12,000
Scrum / Tech Lead	1.0	S/.8,000	S/.24,000	S/.24,000
Backend Java Developer	3.0	S/.5,000	S/.15,000	S/.45,000
COBOL Developer	1.0	S/.6,000	S/.18,000	S/.18,000
Backend Event / Stream Developer	0.5	S/.6,000	S/.18,000	S/.9,000
QA	0.5	S/.3,500	S/.10,500	S/.5,250
Solution Architect	0.3	S/.9,000	S/.27,000	S/.8,100
Data Architect	0.3	S/.8,000	S/.24,000	S/.7,200
Backend Java Architect	0.3	S/.8,000	S/.24,000	S/.7,200
Event / Stream Architect	0.3	S/.9,000	S/.27,000	S/.8,100
Subtotal Q1				S/.143,850
Q2				
Product Owner	0.5	S/.8,000	S/.24,000	S/.12,000
Scrum / Tech Lead	1.0	S/.8,000	S/.24,000	S/.24,000
Backend Java Developer	3.0	S/.5,000	S/.15,000	S/.45,000
COBOL Developer	0.5	S/.6,000	S/.18,000	S/.9,000
Backend Event / Stream Developer	1.0	S/.6,000	S/.18,000	S/.18,000
QA	1.0	S/.3,500	S/.10,500	S/.10,500
Solution Architect	0.3	S/.9,000	S/.27,000	S/.8,100
Backend Java Architect	0.3	S/.8,000	S/.24,000	S/.7,200
Event / Stream Architect	0.3	S/.9,000	S/.27,000	S/.8,100
Subtotal Q2				S/.141,900
Total del proyecto				S/.285,750

Nota. Elaboración Propia.

Costos indirectos asignados a infraestructura, herramientas y servicios de soporte

Si bien durante la ejecución del proyecto no se realizaron adquisiciones adicionales de infraestructura ni compra de nuevas licencias específicas, para efectos de la evaluación económica se consideraron los costos indirectos asociados al uso de recursos corporativos necesarios para implementar la solución. En este contexto, infraestructura hace referencia a los recursos físicos, uso de instalaciones y pago de servicios; herramientas corresponde a las licencias de software y plataformas corporativas utilizadas, y servicios de soporte se refiere al soporte de TI provisto por terceros. El cálculo de estos costos indirectos se realizó de manera proporcional al esfuerzo del proyecto, utilizando como referencia los FTE asignados por quarter.

Figura 26

Costos indirectos asignados a infraestructura, herramientas y servicios de soporte (S/)

Concepto	FTE del quarter	Tarifa mensual por FTE (S/)	Costo por quarter (S/) (= mensual×3)	Costo total (S/) (= FTE×costo Q)
Q1				
Depreciación de laptops	7.7	S/.208	S/.624	S/.4,805
Licencias de software y herramientas corporativas	7.7	S/.200	S/.600	S/.4,620
Infraestructura de oficina e instalaciones (internet/servicios)	7.7	S/.300	S/.900	S/.6,930
Soporte TI (consultora)	7.7	S/.150	S/.450	S/.3,465
Subtotal Q1				S/.19,820
Q2				
Depreciación de laptops	7.9	S/.208	S/.624	S/.4,930
Licencias de software y herramientas corporativas	7.9	S/.200	S/.600	S/.4,740
Infraestructura de oficina e instalaciones (internet/servicios)	7.9	S/.300	S/.900	S/.7,110
Soporte TI (consultora)	7.9	S/.150	S/.450	S/.3,555
Subtotal Q2				S/.20,335
Total del proyecto				S/.40,154

Nota. Elaboración Propia.

Costos operativos post-producción

Este costo se estima de forma anual y está asociado al esfuerzo requerido para asegurar la continuidad del producto en producción. La estimación considera la asignación de un perfil de soporte, valorizado según su costo mensual referencial y con una dedicación (FTE) que puede variar año a año en función del crecimiento del volumen operativo, e incluye actividades de monitoreo, atención de incidentes y gestión de despliegues. Asimismo, se incluyen los costos indirectos asignados a infraestructura, herramientas y servicios de soporte, los cuales se calcularon de manera proporcional, siguiendo el mismo criterio de asignación de FTE por año.

Figura 27

Proyección de costos operativos post-producción a 5 años (S/)

Concepto	Año 1	Año 2	Año 3	Año 4	Año 5
Costo anual de mantenimiento/operación	-36000	-53000	-53000	-71000	-71000

Nota. Elaboración Propia.

3.3.1.2. Beneficios del proyecto

Los beneficios del proyecto se clasificaron en cuantitativos y cualitativos. Los beneficios cuantitativos se incorporaron en el flujo de caja para el cálculo del VAN y la TIR, mientras que los beneficios cualitativos se analizaron como impactos relevantes del

proyecto que no se reflejaron directamente en términos monetarios. Esta clasificación permitió evaluar de manera integral tanto el retorno económico de la solución como los aportes obtenidos a nivel operativo y tecnológico.

Beneficios cuantitativos

Los beneficios cuantitativos del proyecto correspondieron a los ingresos obtenidos a partir de la apertura del canal de Open Banking para el producto, lo que permitió exponer el flujo de pagos a través de APIs, así como a los ahorros generados por la migración de los componentes legacy. En este contexto, se consideraron los siguientes beneficios:

- **Beneficio por uso de API (Open Banking):** Comisiones generadas por el procesamiento de transacciones a través de las APIs expuestas en el canal de Open Banking, calculados en función del volumen transaccional proyectado para cada año.
- **Beneficio por suscripción (Open Banking):** Ingresos recurrentes provenientes del cobro de una suscripción fija por el acceso y uso de los servicios del producto a través del canal de Open Banking.
- **Ahorro por desuso/migración de sistema obsoleto (costos evitados):** Ahorros asociados a la reducción del uso del sistema obsoleto, reflejados en menores costos de operación y mantenimiento.
- **Ahorro por reducción de incidencias/caídas (pérdidas evitadas):** Ahorros asociados a la disminución de los riesgos operativos y a la mejora de la estabilidad del servicio, representando las pérdidas evitadas por interrupciones en la operación.

Figura 28

Proyección de beneficios cuantitativos a 5 años (S/)

Beneficio	Año 1	Año 2	Año 3	Año 4	Año 5
Beneficio por uso de API	36000	90000	216000	360000	600000
Beneficio por suscripción	720000	1200000	2160000	3600000	6000000
Ahorro por desuso/migración del legacy	72000	72000	108000	108000	108000
Ahorro por reducción de incidencias/caídas	30000	30000	48000	48000	48000
Total beneficios cuantitativos	858000	1392000	2532000	4116000	6756000

Nota. Elaboración Propia.

Beneficios cualitativos

Los beneficios cualitativos del proyecto se manifestaron en mejoras relevantes sobre el producto y en la incorporación de capacidades técnicas y funcionales, las cuales se detallan a continuación:

- **Ejecución near-real-time:** Procesamiento en línea de las transacciones, con disponibilidad continua del producto durante las 24 horas del día.
- **Trazabilidad end to end:** Seguimiento de las transacciones, permitiendo al cliente visualizar en tiempo real el estado de su planilla y facilitando el monitoreo de las operaciones por parte del equipo de atención al cliente.
- **Escalabilidad del producto:** Capacidad de procesar planillas de hasta 7,000 pagos y de soportar incrementos en el volumen de transacciones, manteniendo un desempeño estable del servicio.
- **Reutilización de servicios:** Servicios backend reutilizables para nuevas integraciones, acelerando futuras iniciativas.

Estos beneficios, si bien no pudieron ser cuantificados en esta evaluación, contribuyen en el mediano y largo plazo a la fidelización de los clientes, a la atracción de nuevos usuarios y al fortalecimiento de la competitividad del banco dentro del sector.

3.3.2. Interpretación del VAN y del TIR

Con los costos y beneficios estimados, se elaboró el flujo de caja del proyecto considerando una proyección de cinco años. En el Año 0 se incorporó la inversión asociada al costo de implementación, correspondiente al personal interno y a los costos indirectos asignados a infraestructura, herramientas y servicios de soporte. En los años posteriores se consideraron los costos operativos post-producción y los beneficios monetizables derivados del canal de Open Banking y del ahorro por la migración de los componentes legacy.

Figura 29

Flujo de caja del proyecto para evaluación económica (S/)

Concepto	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Costos						
Costo total del proyecto	-325904					
Costo anual de mantenimiento/operación	0	-36000	-53000	-53000	-71000	-71000
Beneficios						
Beneficio por uso de API	0	36000	90000	216000	360000	600000
Beneficio por suscripción	0	720000	1200000	2160000	3600000	6000000
Ahorro por desuso/migración del legacy	0	72000	72000	108000	108000	108000
Ahorro por reducción de incidencias/caídas	0	30000	30000	48000	48000	48000
Flujo Neto	-325904	822000	1339000	2479000	4045000	6685000

Nota. Elaboración Propia.

A partir del flujo neto anual resultante, se realizó el cálculo del VAN y la TIR, cuyos resultados se presentan a continuación.

Figura 30

Indicadores de evaluación económica

Indicador	Valor
Horizonte de evaluación	5 años
Tasa de descuento	10%
VAN (S/)	10,304,138
TIR	318%

Nota. Elaboración Propia.

El VAN obtenido es positivo (S/ 10,304,138), lo cual indica que, al descontar los flujos futuros a una tasa del 10%, el proyecto genera valor neto durante los años proyectados. En otras palabras, los beneficios monetizables proyectados superan tanto los costos del proyecto como su costo de oportunidad asociado a la tasa de descuento.

Por su parte, la TIR alcanzada (318%) es significativamente superior a la tasa de descuento utilizada (10%), lo que confirma que, bajo los supuestos planteados, el proyecto presenta una rentabilidad mayor al rendimiento mínimo exigido y, en consecuencia, resulta financieramente viable.

CAPÍTULO IV - REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

Durante la ejecución del proyecto, la autora participó como Backend Software Engineer a lo largo de todo el ciclo de implementación, desde la etapa de Refinamiento hasta el despliegue en producción. Lo que le permitió entender el proyecto en su totalidad, no solo desde el desarrollo técnico, sino también desde la coordinación con áreas transversales y con negocio. En este proceso, la autora fortaleció sus habilidades técnicas y blandas, y amplió su conocimiento sobre el negocio de Banca Empresa y los flujos de trabajo típicos de proyectos en el sector bancario.

A nivel técnico, la implementación de la solución para el procesamiento en streaming de pagos masivos representó un reto importante por tratarse de un flujo crítico que involucra transacciones monetarias. En este contexto, una falla podía generar impactos operativos y contables, además de riesgos reputacionales para la entidad. Por ello, se requirió mucho cuidado en el diseño de escenarios de error y en el manejo de casuísticas especiales, lo que permitió a la autora reforzar su criterio técnico en arquitectura y desarrollo backend orientado a streaming, comprendiendo la relevancia de diseñar soluciones tolerantes a fallos y preparadas para condiciones reales de operación.

Una de las lecciones aprendidas más relevantes fue comprender la función que aún cumple el mainframe en el entorno bancario. Pese a las iniciativas de modernización, el mainframe continúa siendo clave en el procesamiento de operaciones críticas por su robustez. Esto le permitió a la autora tener una visión más realista sobre los procesos de migración tecnológica en banca: la transición hacia cloud suele ser progresiva, con convivencia entre componentes legacy y soluciones modernas, manteniendo aún un uso controlado de transacciones en mainframe.

Por otro lado, el proyecto necesitó coordinación con diversos equipos (Arquitectura, Gestión de Cambios, responsables de plataformas y entornos, entre otros), lo que requirió una comunicación constante para coordinar actividades y resolver bloqueos. En ese proceso, la autora fortaleció su comunicación efectiva y su capacidad de negociación al definir alcances y prioridades con distintas áreas. Además, el trabajo colaborativo, la gestión del tiempo y la resolución de incidencias fueron clave para cumplir los objetivos definidos por negocio, considerando que los bloqueos por dependencias externas podían retrasar pruebas o despliegues.

Respecto a la forma de trabajo, el equipo aplicó una metodología ágil combinada, Lean Inception en la etapa de ideación y refinamiento, y Scrum en la etapa de ejecución. Lean Inception permitió alinear la visión inicial del producto y trasladarla a un backlog de entregables de acuerdo con las necesidades primordiales de negocio, mientras que Scrum facilitó la organización, priorización y el seguimiento continuo del avance. No obstante, debido a la complejidad técnica de la solución, no siempre fue posible realizar entregas productivas en cada sprint, ya que varios entregables correspondían a habilitadores necesarios para alcanzar el resultado final. Esto reforzó en la autora la importancia de definir de forma temprana criterios de “Terminado” realistas en proyectos complejos con múltiples integraciones.

Entre las principales oportunidades de mejora, se identificó la definición tardía de algunos criterios de aceptación relacionados con el manejo de errores y devoluciones, lo que generó ajustes de diseño y retrabajo en etapas avanzadas. Asimismo, las dependencias con otros equipos para despliegues en entornos de pruebas y los problemas de estabilidad de dichos entornos extendieron los tiempos de validación, evidenciando la necesidad de considerar en la planificación factores como accesos, despliegues y estabilidad de ambientes.

Finalmente, la autora considera que este proyecto consolidó sus conocimientos sobre los flujos de pagos en Banca Empresa, permitió comprender la convivencia entre sistemas legacy y soluciones cloud, y amplió su visión sobre los servicios bancarios. Asimismo, al tratarse de una primera migración por fases, el proyecto sirvió como base y referencia para iniciativas posteriores orientadas a migrar otros productos de pagos masivos. En conjunto, la experiencia fortaleció el perfil técnico de la autora y su capacidad de colaboración interdisciplinaria.

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Al finalizar el proyecto, se implementó una solución tecnológica para el procesamiento en streaming de pagos masivos de nóminas, logrando reducir la dependencia del mainframe bancario y utilizándolo solo para la ejecución de la transacción monetaria. De este modo, la lógica de negocio y el procesamiento operativo se trasladaron a la nube, lo que permitió disminuir restricciones técnicas y riesgos operativos asociados al sistema legacy, mejorar la continuidad del servicio y contribuir a la reducción de costos derivados de la operación sobre la plataforma mainframe.

- Se habilitó la ejecución near-real-time de la liquidación de planillas, permitiendo la recepción 24x7 y eliminando las ventanas horarias limitadas de liquidación. Asimismo, se logró que el débito al cliente y el abono a los beneficiarios se ejecuten de forma inmediata mediante transacciones en línea.
- Se mejoró la escalabilidad del producto, permitiendo procesar planillas de hasta 7000 pagos manteniendo la estabilidad del servicio, incluso en periodos de alta demanda como quincena y fin de mes.
- Se logró modularizar las funcionalidades de pago de nóminas, habilitando APIs reutilizables para su consumo desde diferentes canales de la entidad. Con ello, se pudo habilitar el modelo de Open Banking para este producto. Además, esta arquitectura facilita la reutilización futura de los servicios para otro tipo de pagos masivos, como pago a proveedores.
- Se incorporó la funcionalidad de trazabilidad y seguimiento en tiempo real, permitiendo al cliente consultar el estado de su planilla y de cada pago, y que el equipo de atención al cliente pueda agilizar la gestión de incidencias. Esto contribuye a mejorar la experiencia del usuario y fortalecer la fidelización de clientes.

5.2. Recomendaciones

- Se sugiere reutilizar los servicios desarrollados para habilitar el pago en streaming de otros productos de pagos masivos, tales como pago a proveedores y pago de CTS, considerando tanto pagos mismo banco como interbancarios, con el fin de aprovechar la arquitectura ya construida y reducir tiempos de implementación.
- Se propone incorporar validaciones previas a la invocación de la transacción en mainframe, como el chequeo del estado de las cuentas y el saldo disponible de la cuenta ordenante, con el objetivo de seguir reduciendo el consumo de recursos del mainframe y prevenir rechazos en etapas posteriores.
- Se recomienda mantener el marco de trabajo Scrum en futuras implementaciones, ya que facilita el seguimiento continuo de las actividades mediante las ceremonias ágiles y ayuda a mantener el foco en los objetivos. Para proyectos complejos con varios componentes e integraciones, se sugiere además socializar con negocio las entregas habilitadoras, de modo que quede constancia del avance hacia el entregable final.
- Al momento de estimar entregas, se sugiere considerar holguras amplias para pruebas, debido a la dependencia con otros equipos para accesos, despliegues y estabilidad de entornos. Incluir estas estimaciones desde la planificación ayuda a evitar retrasos no previstos y a acordar fechas más realistas con negocio desde el inicio.
- Finalmente, se recomienda definir desde la etapa de Refinamiento un catálogo de casuísticas de error que involucren devoluciones de dinero, sobre todo en flujos monetarios sensibles que pueden generar descuadres contables. Esto reduce el retrabajo durante la construcción y las pruebas, y ayuda a asegurar un diseño más claro desde los primeros sprints.

5.3. Fuentes de información

Accenture. (2025). *Banking: The future is back: Trends shaping the industry in 2025 and beyond* [Informe en PDF]. Recuperado de <https://www.accenture.com/content/dam/accenture/final/industry/banking/document/Accenture-Banking-Top-10-Trends-2025-Report.pdf#zoom=40>

Badman, A., & Kosinski, M. (s. f.). *¿Qué es la transmisión de datos?. IBM*. Recuperado de <https://www.ibm.com/mx-es/think/topics/streaming-data>

Caroli, P. (2019). *Lean Inception: Creando conversaciones hacia un producto exitoso* (P. Trejo, Trad.). Porto Alegre, Brasil: Editora Caroli. (Trabajo original publicado en 2018)

Eloshvili, R. (2025, abril 18). Tech-Driven Digital Banking: Innovations And Challenges Ahead. *Forbes*. Recuperado de <https://www.forbes.com/councils/forbestechcouncil/2025/04/18/tech-driven-digital-banking-innovations-and-challenges-ahead/>

Gill, A. (2019). Building Scalable Batch Processing Systems for Financial Transactions Using Mainframe. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 10(12), 92–109. <https://doi.org/10.61841/turcomat.v10i12.14954>

IBM. (s. f.). *¿Qué es la arquitectura basada en eventos?.* Recuperado de <https://www.ibm.com/mx-es/think/topics/event-driven-architecture>

KPMG. (2025). *2025 Banking Survey: Technology*. Recuperado de <https://kpmg.com/us/en/articles/2025/kpmg-banking-survey-2025.html>

Project Management Institute, & Agile Alliance. (2017). *Agile practice guide*. Newtown Square, USA: Project Management Institute.

Rayns, C., Bertram, S., Bogner, G., Carlin, C., Clark, A., Ferrell, A., ... Woerner, E. (2011). *CICS Transaction Server from Start to Finish*. IBM Corp.

Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum: La Guía Definitiva de Scrum: Las Reglas del Juego* [Guía en PDF]. Recuperado de

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Superintendencia de Banca, Seguros y AFP. (2025). *Informe de Estabilidad del Sistema Financiero* [Informe en PDF]. Recuperado de <https://www.sbs.gob.pe/Portals/0/IESF-2025-1A.pdf>

Susnjara, S., & Smalley, I. (s. f.). *¿Qué es cloud computing?. IBM*. Recuperado de <https://www.ibm.com/mx-es/think/topics/cloud-computing>

Tatipamula, S. (2025). Real-Time vs. Batch Data Processing: When speed matters. *World Journal of Advanced Research and Reviews*, 26(01), 1612–1631. <https://doi.org/10.30574/wjarr.2025.26.1.1213>

Velepucha, V., & Flores, P. (2023). A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges. *IEEE*, 11, 88339 – 88358. <https://doi.org/10.1109/ACCESS.2023.3305687>

5.4. Glosario

- **MVP (Producto Mínimo Viable):** Es la primera versión funcional de un producto que incorpora únicamente las características esenciales para generar valor temprano y validar la solución con el cliente con la menor inversión posible.
- **EMC (Entregable en Manos del Cliente):** Corresponde a un entregable que ya se encuentra disponible para el cliente, es decir, una pieza de valor implementada y puesta a disposición del usuario final para su uso, validación o adopción.
- **Backlog:** Es el listado priorizado de trabajo pendiente del producto o proyecto, que agrupa requerimientos, mejoras y correcciones, y que se actualiza y refina de manera continua según el valor y las necesidades del negocio.
- **E2E (End to End):** Se refiere a la ejecución completa de un flujo de principio a fin, en términos de trazabilidad, implica poder rastrear la información o el proceso a lo largo de todo su ciclo de vida.
- **Legacy:** Hace referencia a sistemas, aplicaciones o tecnologías heredadas que continúan operando en una organización porque aún cumplen su propósito, aunque suelen presentar limitaciones por antigüedad, como poca flexibilidad para evolucionar y mayor complejidad de mantenimiento.
- **DB2:** Es un sistema de gestión de bases de datos de IBM, ampliamente utilizado en entornos empresariales y plataformas mainframe.
- **DevOps:** Es un enfoque de trabajo que combina prácticas, cultura y herramientas para alinear a los equipos de desarrollo y operaciones, con el objetivo de mejorar la colaboración y acelerar la entrega de software con estabilidad y calidad.
- **CI/CD (Integración Continua y Entrega/Despliegue Continuo):** Es un conjunto de prácticas que automatiza la integración, pruebas y despliegue de código, permitiendo liberar versiones con mayor frecuencia y menor riesgo.
- **Omnicanalidad:** Es una estrategia que busca conectar y coordinar todos los canales de atención e interacción con el cliente para ofrecer una experiencia consistente e integrada, permitiendo continuidad sin interrupciones.

ANEXOS

Anexo 1: Resumen de MVPs, features, historias de usuario, sprints y entregables

MVP	Features	Historias de Usuario	Sprints	Entregables
MVP1: Servicio de incorporación de transferencias masivas de nóminas (mismo banco)	F1: Persistencia y modelo de datos	F1-HU01 [EN]: Persistencia en BD Operacional (Oracle)	Q1-SP0	Modelo conceptual/lógico Oracle
			Q1-SP1	Modelo físico final Oracle
			Q1-SP2	Scripts (DDL/DML) + despliegue en ambientes previos
		F1-HU02 [EN]: Persistencia en BD Informacional (MongoDB)	Q1-SP0	Modelo conceptual/lógico MongoDB
			Q1-SP1	Modelo final MongoDB
			Q1-SP2	Scripts/artefactos de despliegue + despliegue en ambientes previos.
	F2: Servicio Crear Nómina	F2-HU01 [FUN]: Servicio y transacción backend Java - Crear Nómina	Q1-SP0	Diseño de servicio backend - Crear Nómina
			Q1-SP1	Diseño de transacción backend - Crear Nómina
			Q1-SP2	Servicio y transacción backend implementado + pruebas unitarias
		F2-HU02 [INT]: Integración de componentes (Servicio + Transacción + BD)	Q1-SP3	Componentes integrados del flujo (Servicio + Transacción + BD)
		F2-HU03 [INT]: Pruebas técnicas y de integración de servicio	Q1-SP4	Evidencias / reporte de pruebas técnicas e integración
		F2-HU04 [REF]: Refinamiento y adecuaciones	Q1-SP4 / SP5	Ajustes incorporados sobre hallazgos (bugs/mejoras)
		F2-HU05 [QA]: Pruebas funcionales del servicio	Q1-SP4	Test Plan aprobado + casos de prueba definidos
		F2-HU06 [REL]: Pase a producción de componentes	Q1-SP5	Ejecución de casos + evidencias + reporte de resultados
			Q1-SP5	Checklist / Documentación de pase a producción
			Q1-SP6	Despliegue productivo
	F3: Servicio Añadir Pagos	F3-HU01 [FUN]: Servicio y transacción backend Java - Añadir Pagos	Q1-SP0	Diseño de servicio backend - Añadir Pagos
			Q1-SP1	Diseño de transacción backend - Añadir Pagos
			Q1-SP2	Servicio y transacción backend implementado + pruebas unitarias
		F3-HU02 [INT]: Integración de componentes (Servicio + Transacción + BD)	Q1-SP3	Componentes integrados del flujo (Servicio + Transacción + BD)
		F3-HU03 [INT]: Pruebas técnicas y de integración de servicio	Q1-SP4	Evidencias / reporte de pruebas técnicas e integración
		F3-HU04 [REF]: Refinamiento y adecuaciones	Q1-SP4 / SP5	Ajustes incorporados sobre hallazgos (bugs/mejoras)
		F3-HU05 [QA]: Pruebas funcionales del servicio	Q1-SP4	Test Plan aprobado + casos de prueba definidos
		F3-HU06 [REL]: Pase a producción de componentes	Q1-SP5	Ejecución de casos + evidencias + reporte de resultados
			Q1-SP5	Checklist / Documentación de pase a producción
			Q1-SP6	Despliegue productivo

MVP	Features	Historias de Usuario	Sprints	Entregables
MVP2: Servicio de procesamiento near-real-time de transferencias masivas de nóminas (mismo banco)	F4: Habilitadores mainframe (liquidación y devolución)	F4-HU01 [EN]: Transacción CICs de Liquidación	Q1-SP3	Diseño de transacción mainframe - Liquidación
			Q1-SP4	Transacción mainframe implementada + pruebas unitarias
		F4-HU02 [EN]: Transacción CICs de Devolución	Q1-SP4	Diseño de transacción mainframe - Devolución
			Q1-SP5	Transacción mainframe implementada + pruebas unitarias
		F4-HU03 [INT]: Pruebas técnicas de transacción	Q1-SP6	Evidencias / reporte de pruebas técnicas
	F5: Servicio de procesamiento near-real-time (Java)	F5-HU01 [FUN]: Servicio y transacción backend Java - Liquidar Nómina	Q1-SP4	Diseño de servicio y transacción backend - Liquidar Nómina
			Q1-SP5	Servicio y transacción backend implementado + pruebas unitarias
		F5-HU02 [INT]: Integración de componentes (Servicio + Transacción)	Q1-SP6	Componentes integrados del flujo (Servicio + Transacción)
		F5-HU03 [INT]: Pruebas técnicas y de integración de servicio	Q1-SP6	Evidencias / reporte de pruebas técnicas e integración
	F6: Subflujo event-driven del procesamiento (transacciones Java por eventos)	F6-HU01 [FUN]: Transacciones backend Java - Paginación / Validaciones / Liquidación (flujo por eventos)	Q1-SP5	Diseño de transacciones backend - Flujo por eventos
			Q1-SP6	Transacciones backend implementadas + pruebas unitarias
		F6-HU02 [FUN]: Transacciones backend Java - Supervisor y Retorno (flujo por eventos)	Q2-SP0	Diseño de transacciones backend - Flujo por eventos
			Q2-SP1	Transacciones backend implementadas + pruebas unitarias
		F6-HU03 [FUN]: Stream/Eventos - Paginación / Validaciones / Liquidación	Q1-SP5	Diseño de streams y suscriptores
			Q1-SP6	Scripts/artefactos de creación de streams y suscriptores + despliegue en ambientes previos
		F6-HU04 [FUN]: Stream/Eventos - Supervisor y Retorno	Q2-SP0	Diseño de streams y suscriptores
			Q2-SP1	Scripts/artefactos de creación de streams y suscriptores + despliegue en ambientes previos
		F6-HU05 [INT]: Integración de componentes (Servicio + Transacción + Eventos/Mainframe)	Q2-SP2	Componentes integrados del flujo (Servicio + Transacción + Eventos/Mainframe)
		F6-HU06 [INT]: Pruebas técnicas y de integración de servicio	Q2-SP3 / SP4	Evidencias / reporte de pruebas técnicas e integración
		F6-HU07 [REF]: Refinamiento y adecuaciones	Q2-SP3 / SP4	Ajustes incorporados sobre hallazgos (bugs/mejoras)
		F6-HU08 [QA]: Pruebas funcionales del servicio	Q2-SP3	Test Plan aprobado + casos de prueba definidos
			Q2-SP5 / SP6	Ejecución de casos + evidencias + reporte de resultados
		F6-HU09 [REL]: Pase a producción de componentes	Q2-SP5	Checklist / Documentación de pase a producción
			Q2-SP6	Despliegue productivo

MVP	Features	Historias de Usuario	Sprints	Entregables
MVP3: Servicio de seguimiento y trazabilidad de transferencias masivas de nóminas	F7: Servicio Listar/Obtener Nómina	F7-HU01 [FUN]: Servicio y transacción backend Java - Listar/Obtener Nómina	Q2-SP1	Diseño de servicio y transacción backend - Listar/Obtener Nómina
			Q2-SP2	Servicio y transacción backend implementado + pruebas unitarias
		F7-HU02 [INT]: Pruebas técnicas y de integración de servicio	Q2-SP3	Evidencias / reporte de pruebas técnicas e integración
		F7-HU03 [REF]: Refinamiento y adecuaciones	Q2-SP3	Ajustes incorporados sobre hallazgos (bugs/mejoras)
		F7-HU04 [QA]: Pruebas funcionales del servicio	Q2-SP2	Test Plan aprobado + casos de prueba definidos
			Q2-SP4	Ejecución de casos + evidencias + reporte de resultados
	F8: Servicio Listar/Obtener Pago	F7-HU05 [REL]: Pase a producción de componentes	Q2-SP4	Checklist / Documentación de pase a producción
			Q2-SP5	Despliegue productivo
		F8-HU01 [FUN]: Servicio y transacción backend Java - Listar/Obtener Pago	Q2-SP1	Diseño de servicio y transacción backend - Listar/Obtener Pago
			Q2-SP2	Servicio y transacción backend implementado + pruebas unitarias
		F8-HU02 [INT]: Pruebas técnicas y de integración de servicio	Q2-SP3	Evidencias / reporte de pruebas técnicas e integración
		F8-HU03 [REF]: Refinamiento y adecuaciones	Q2-SP3	Ajustes incorporados sobre hallazgos (bugs/mejoras)
		F8-HU04 [QA]: Pruebas funcionales del servicio	Q2-SP2	Test Plan aprobado + casos de prueba definidos
			Q2-SP4	Ejecución de casos + evidencias + reporte de resultados
		F8-HU05 [REL]: Pase a producción de componentes	Q2-SP4	Checklist / Documentación de pase a producción
			Q2-SP5	Despliegue productivo