UPN
UNIVERSIDAD PRIVADA DEL NORTE

FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

IMPLEMENTACIÓN DE UN SISTEMA ANTIFRAUDE EN EL PROCESO DE PAGOS DE LA EMPRESA KASHIO PERÚ S.A.C. LIMA, 2026

Trabajo de suficiencia profesional para optar al título profesional de:

Ingeniero de Sistemas Computacionales

Autor:

Juan Carlos Solorzano Zanabria

Asesor:

Mg. Ing. Carlos Eduardo Mendoza Santos

https://orcid.org/0000-0002-7193-9715

Lima - Perú

2026

Informe de Similitud

Juan Carlos SOLORZANO ZANABRIA

COR-F-REC-VAC-05.17_8.docx

REVISION_TESIS
ASESORIA UPN 2026
Asesores

Detalles del documento

Identificador de la entrega
trn:oid:::1:3506331794

Fecha de entrega
13 mar 2026, 5:18 p.m. GMT-5

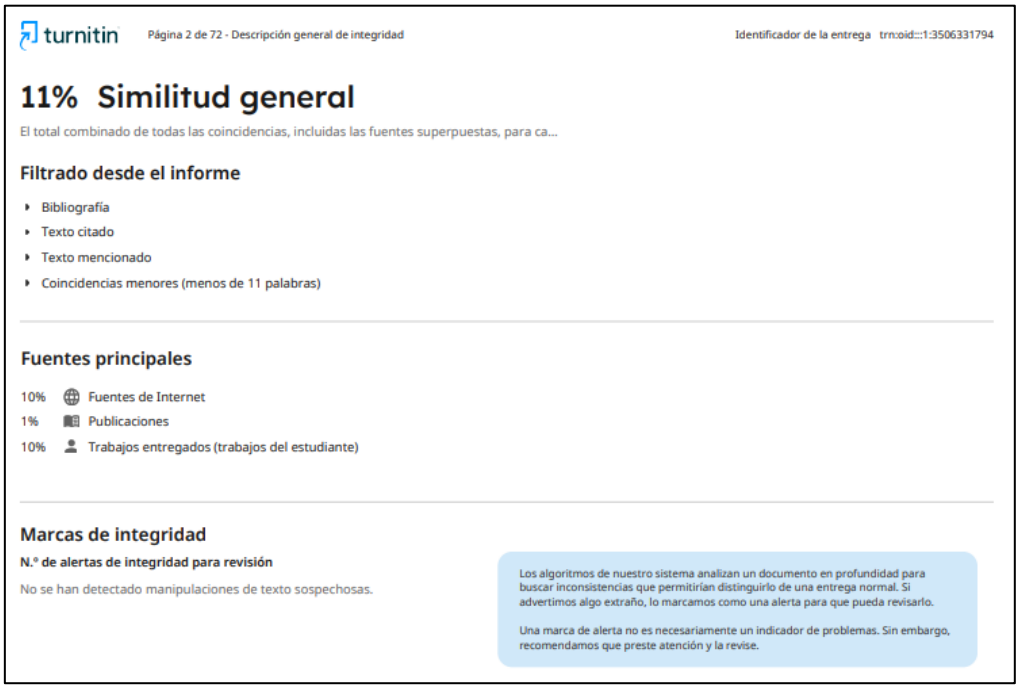
Fecha de descarga
13 mar 2026, 10:22 p.m. GMT-5

Nombre del archivo
COR-F-REC-VAC-05.17_8.docx

Tamaño del archivo
2.6 MB

68 páginas
8809 palabras
53.289 caracteres

turnitin Página 1 de 72 - Portada Identificador de la entrega trn:oid:::1:3506331794



turnitin Página 2 de 72 - Descripción general de integridad Identificador de la entrega trn:oid:::1:3506331794

11% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 11 palabras)

Fuentes principales

10% Fuentes de Internet
1% Publicaciones
10% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.° de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Dedicatoria

A Dios, por ser el guía de mi camino y concederme la sabiduría necesaria para culminar esta etapa profesional.

A mi madre, por su amor incondicional, apoyo y sacrificio a lo largo de mi formación, pues gracias a su entrega he logrado convertirme en la persona que soy. A mi padre, que me guía desde el cielo y cuya memoria sigue siendo mi mayor ejemplo de superación.

A mi esposa, por ser mi apoyo incondicional y caminar a mi lado en cada etapa de este reto, y a mis hijos, quienes son mi motor y la razón de cada uno de mis esfuerzos. A todos ellos les dedico este logro con profunda gratitud.

Agradecimiento

A Dios, por ser mi guía espiritual y darme la fuerza necesaria para ver culminado este sueño que hoy se convierte en realidad.

A mi madre, por su amor incondicional y por ser el pilar fundamental de mi vida; gracias a su sacrificio constante hoy soy quien soy. A mi padre, que me cuida desde el cielo y cuya memoria es el faro que ilumina mi camino.

A mi esposa, por su paciencia eterna y por ser mi compañera incondicional en cada desvelo, y a mis hijos, quienes son mi mayor bendición y la razón por la cual busco superarme cada día.

Asimismo, expreso mi más profundo agradecimiento a mis tías, quienes con su apoyo constante y palabras de aliento me impulsaron a superarme en mi carrera, convirtiéndose en un motivo esencial para alcanzar esta meta profesional.

Tabla de contenido

Índice de Tablas

Índice de Figuras

RESUMEN EJECUTIVO

El presente trabajo de suficiencia profesional describe el diseño e implementación de un Motor de Reglas Antifraude para transacciones PAYIN, desarrollado en la plataforma de pagos digitales Kashio. La experiencia profesional permitió identificar que la organización enfrentaba desafíos en la detección temprana de operaciones sospechosas dentro del flujo de pagos, lo que representaba un riesgo potencial para la seguridad de las transacciones y la confiabilidad del sistema. La ausencia de un mecanismo automatizado para la evaluación de fraude dificultaba la identificación oportuna de comportamientos inusuales y limitaba la capacidad de respuesta ante posibles intentos de fraude.

Con el propósito de mitigar esta problemática, se diseñó e implementó un sistema antifraude capaz de analizar las transacciones en tiempo real mediante un conjunto de reglas configurables por el área de riesgos. El sistema fue desarrollado utilizando tecnologías modernas orientadas a arquitecturas web, empleando Python para el desarrollo del backend, Node.js y TypeScript para la gestión de servicios, y Angular para la construcción de la interfaz de usuario, permitiendo así una arquitectura flexible, escalable y orientada a servicios. Asimismo, se integraron APIs especializadas para la evaluación de transacciones, módulos de administración de reglas y herramientas de monitoreo de operaciones sospechosas.

Entre las principales funcionalidades implementadas se encuentran la evaluación antifraude en línea durante los procesos de creación de deuda y ejecución de pagos, la gestión de reglas antifraude con distintos niveles de criticidad, el registro de transacciones bloqueadas para su posterior revisión y un módulo de autogestión que permite al área de riesgos administrar las reglas del sistema sin intervención directa del equipo técnico.

Adicionalmente, se incorporó un sistema de reportes automáticos y estadísticas que facilita el análisis del comportamiento transaccional y contribuye a mejorar la toma de decisiones estratégicas.

Los resultados del proyecto demostraron una mejora considerable en los mecanismos de control y detección de fraude dentro del flujo transaccional. La automatización de la evaluación permitió reducir los tiempos de análisis de transacciones, mejorar la trazabilidad de las operaciones y fortalecer los mecanismos de seguridad de la plataforma. Asimismo, el sistema permitió detectar y bloquear operaciones sospechosas en tiempo real, reduciendo el riesgo de fraude y optimizando la capacidad de monitoreo del área de riesgos.

En conclusión, la implementación del Motor de Reglas Antifraude contribuyó a fortalecer la seguridad de las transacciones digitales, optimizar los procesos de validación y mejorar la capacidad de respuesta ante posibles escenarios de fraude. Este proyecto permitió aplicar y consolidar competencias profesionales en análisis de sistemas, desarrollo de software, arquitectura de aplicaciones, gestión de bases de datos y seguridad en plataformas de pagos digitales, aportando valor tanto a la organización como al desarrollo profesional del autor.

CAPÍTULO I. INTRODUCCIÓN

1.1 Experiencia profesional

Mi carrera profesional se ha construido sobre una base de constante evolución técnica y estratégica. Inicié mi formación con conocimientos en Redes y Comunicaciones en Cibertec, los cuales sirvieron como cimiento técnico para mis primeros roles en infraestructura y administración de sistemas en Dreams Corporation. Posteriormente, consolidé mi preparación académica obteniendo el grado de Bachiller en Ingeniería de Sistemas por la Universidad Privada del Norte.

A lo largo de mi trayectoria, transité desde el análisis de datos y la gestión de proyectos en empresas como Intelidata, Cinco Millas y Lumingo, hacia una especialización en el ecosistema SAP y la optimización de procesos en el sector Retail con Ripley Perú. Mi experiencia en sistemas de pago se profundizó en SafetyPay como System Engineer, donde lideré el análisis de requerimientos de negocio y la gestión de demanda.

En la etapa más reciente de mi carrera, me he consolidado como Technical Product Owner, especializándome en el diseño y escalabilidad de ecosistemas financieros. Tras liderar soluciones SAP ByDesign para el Banco Santander, asumí mi rol actual en la fintech Kashio. En esta posición, aplico mi experiencia en procesamiento de pagos y mis certificaciones en Scrum para definir la viabilidad técnica de productos complejos. Mi enfoque se centra en transformar requerimientos de negocio en especificaciones técnicas robustas, asegurando integraciones seguras con servicios financieros y garantizando la entrega de soluciones innovadoras en el competitivo sector Fintech.

1.2 Descripción de la empresa

Kashio es una empresa de tecnología financiera (Fintech) de financiación privada fundada en el año 2017. Con sede central en Miami, Florida, la organización se ha consolidado como un referente regional en servicios financieros digitales, operando estratégicamente en Perú, Chile, México y Colombia. La compañía cuenta con un equipo de entre 51 y 200 empleados dedicados a la innovación en sistemas de pago.

Su misión es ayudar a las empresas a adoptar pagos digitales y automatizar sus procesos de cuentas por cobrar y pagar mediante una plataforma en la nube basada en tecnología de banca abierta (Open Banking).

1.3 Cultura Organizacional: Valores y ADN

Los valores organizacionales constituyen el eje fundamental que orienta la cultura corporativa, la toma de decisiones estratégicas y el desarrollo tecnológico dentro de la empresa. En el caso de Kashio, estos principios se encuentran alineados a la naturaleza del sector fintech, donde la confianza, la innovación y la adaptabilidad son factores determinantes para la sostenibilidad del negocio.

De acuerdo con la información publicada en su sitio web oficial, los valores institucionales de Kashio se estructuran en cuatro pilares fundamentales (Kashio, s. f.):

- **Seguridad:** Generación de confianza absoluta mediante procesos robustos y transparentes que garantizan la integridad de cada operación.

- **Simplicidad:** Transformación de lo complejo en soluciones accesibles e intuitivas que eliminan barreras financieras.

- **Disrupción:** Creación de nuevos caminos en la industria financiera a través de la innovación constante y soluciones audaces.

- **Flexibilidad:** Agilidad para adaptarse a las necesidades únicas de cada cliente y realidad del mercado.

- **ADN Kashio (Competencias Clave)**

El equipo cultiva comportamientos orientados a la excelencia, destacando el Cumplimiento normativo, el enfoque en Resultados medibles y la Colaboración sinérgica. Asimismo, promueve una mentalidad Al-Ready para escalar el impacto mediante inteligencia artificial, la Mejora Continua, una visión Cliente Céntrico y el espíritu Kashioness, que define la resiliencia e integridad de sus miembros. (Kashio, s. f.).

1.4 Portafolio de Soluciones y Servicios

Los pagos digitales comprenden un conjunto de servicios que permiten realizar operaciones financieras electrónicas de forma segura y en tiempo real:

- **Payins:** El servicio de Payin permite a comercios y organizaciones recibir pagos digitales de sus clientes mediante diversos métodos electrónicos, tales como transferencias bancarias, billeteras digitales o tarjetas. Este proceso implica la generación de una orden de pago o deuda, la validación del método seleccionado y la confirmación de la transacción en tiempo real (Kashio, s. f.).

- **Payouts:** El servicio de Payout permite la dispersión de fondos desde la plataforma hacia terceros, tales como proveedores, colaboradores o cuentas bancarias registradas. Este proceso puede realizarse de manera individual o masiva, automatizando la ejecución de pagos previamente autorizados (Kashio, s. f.).
- **Gestión de Cuentas:** La gestión de cuentas en plataformas fintech implica la administración automatizada de procesos de recaudación, cobranza y ejecución de pagos. Esta automatización mejora la eficiencia operativa, pero también incrementa la necesidad de controles inteligentes. Entre los principales componentes se encuentran: Automatización de Recaudos, Cobranzas Inteligentes y Pagos Masivos. (Kashio, s. f.).

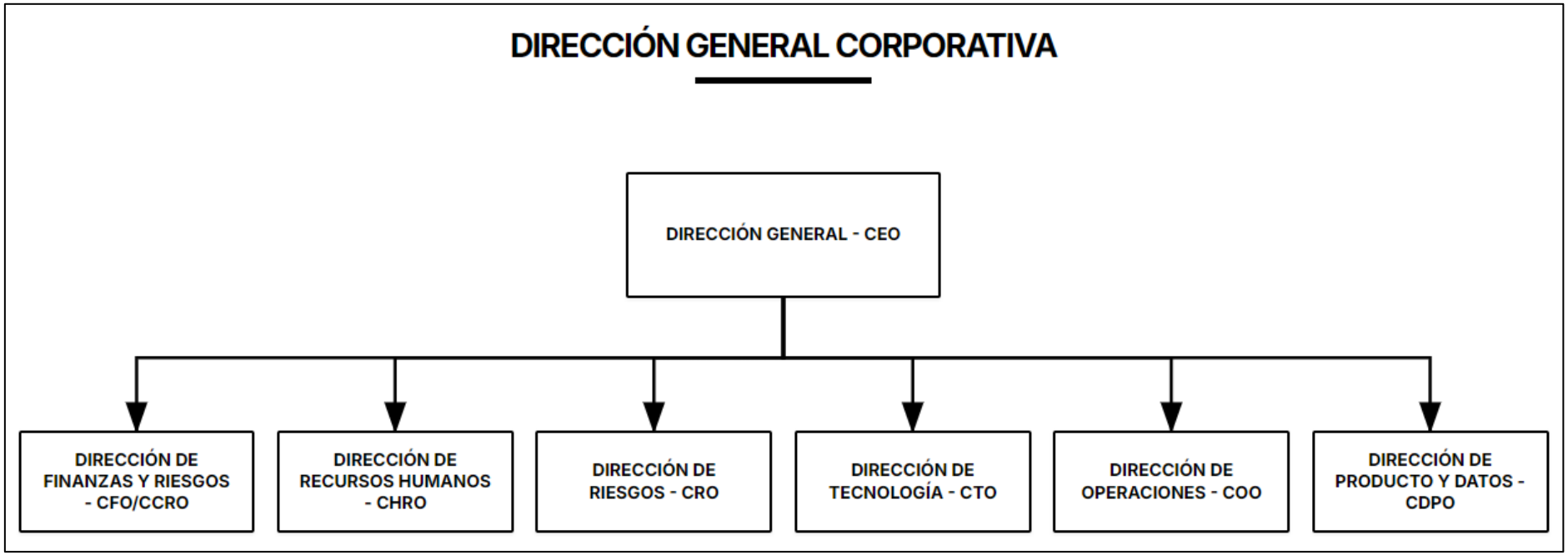
- **Plataforma como Servicio (PaaS):** El modelo Plataforma como Servicio (PaaS) permite ofrecer infraestructura tecnológica que conecta múltiples Prov

- eedores de Servicios de Pago (PSPs) y entidades bancarias dentro de un mismo entorno integrado. Este modelo facilita la integración con diferentes métodos de pago, la estandarización de interfaces mediante APIs, la escalabilidad del procesamiento transaccional y la reducción de complejidad para los comercios afiliados (Kashio, s. f.).

1.5 Organigrama de la Empresa

Figura 1

Organigrama Estructural de Kashio Peru.



Nota: Elaboración propia a partir de información institucional proporcionada por Kashio (comunicación interna, 2026).

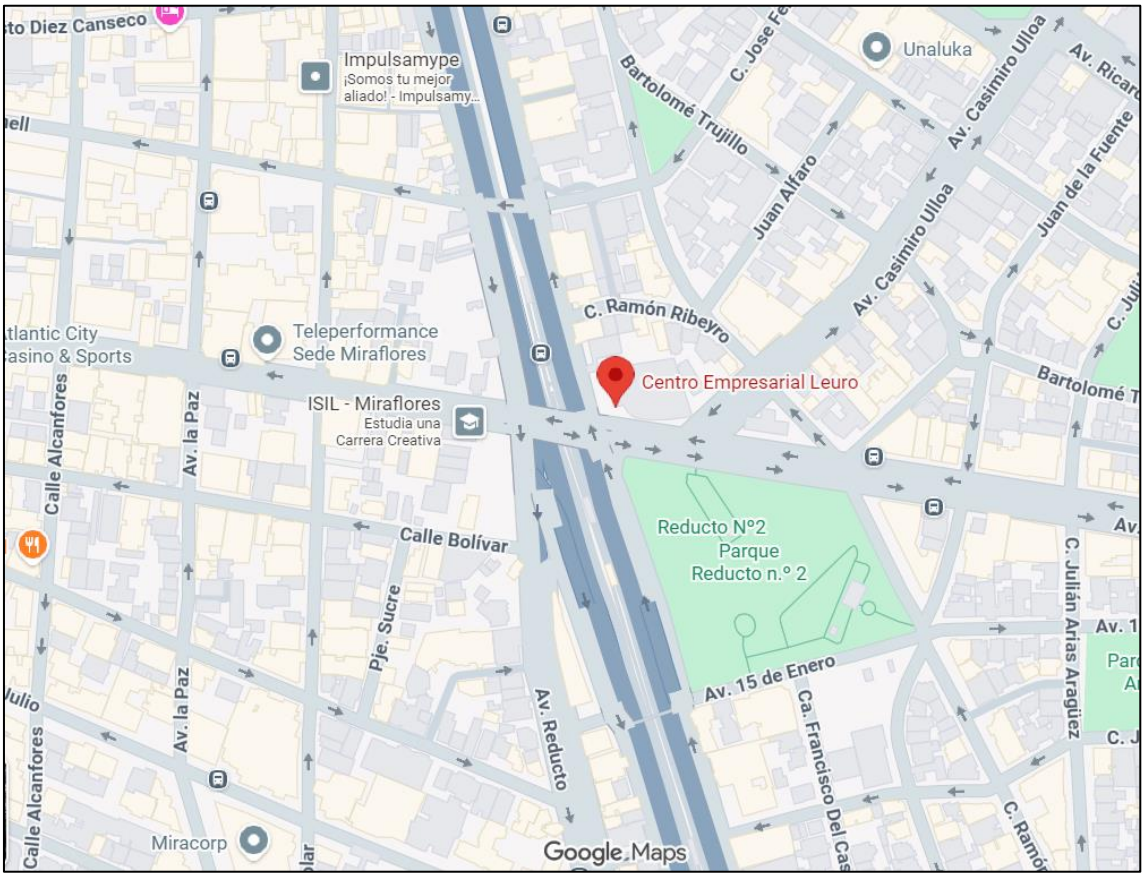
1.6 Ubicación de la Sede en Perú

- **Dirección:** Avenida Paseo de la República N° 5895, Piso 11, Edificio Leuro, Miraflores, Lima 15000, PE.

- **Referencia:**

Figura 2

Ubicación del Centro Empresarial Leuro



Nota. Adaptado de *Google Maps* (s. f.), https://maps.google.com/

CAPÍTULO II. MARCO TEÓRICO

El diseño e implementación del sistema antifraude se sustentó en una arquitectura basada en servicios independientes, lo que permitió separar la lógica de evaluación de riesgo del flujo principal de pagos. Esta estrategia posibilita analizar las transacciones en tiempo real sin afectar la operación del sistema transaccional. La solución se implementó mediante un motor de reglas independiente encargado de analizar operaciones financieras durante la creación y pago de deudas, devolviendo una decisión automática de permitir, revisar o bloquear la transacción.

Para el funcionamiento operativo se implementó un portal administrativo que permite al área de riesgos visualizar transacciones sospechosas y realizar análisis manual cuando el nivel de riesgo lo requiera. Este enfoque combina automatización con supervisión humana, práctica común en sistemas financieros debido a requerimientos regulatorios y de auditoría.

El desarrollo del sistema se realizó utilizando la metodología ágil Scrum, la cual permitió organizar el trabajo en iteraciones cortas denominadas Sprint, facilitando la entrega continua de funcionalidades y la adaptación a cambios en los requerimientos del negocio. Mediante el uso de historias de usuario y backlog priorizado se gestionaron las reglas antifraude, integraciones y funcionalidades del portal administrativo.

El sistema se construyó bajo una arquitectura de microservicios separando el backend de decisiones del frontend de monitoreo. El servicio principal, encargado de la evaluación antifraude, procesa reglas configurables y genera auditoría, mientras que el portal administrativo permite la gestión de alertas y bloqueos.

2.1 Fundamentos teóricos

A. Metodología Scrum

Scrum es una metodología ágil orientada a la gestión de proyectos de software mediante iteraciones cortas que permiten entregar valor de forma progresiva. Este enfoque promueve la colaboración del equipo, la adaptación a cambios y la mejora continua a través de ciclos de trabajo denominados Sprint (Schwaber & Sutherland, 2020). Este enfoque permite entregar valor de manera continua mediante pequeñas entregas funcionales. El proceso se organiza en Product Backlog, Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective. La aplicación de Scrum permitió implementar progresivamente las reglas antifraude e integrar el sistema sin afectar la continuidad operativa, facilitando la adaptación ante nuevos escenarios de riesgo.

B. Python

Python es un lenguaje de programación de alto nivel que se caracteriza por su sintaxis sencilla y su facilidad para desarrollar aplicaciones de manera rápida. Debido a su flexibilidad y amplia disponibilidad de bibliotecas, es utilizado en múltiples áreas del desarrollo de software, incluyendo aplicaciones empresariales y análisis de datos (Van Rossum & Drake, 2009). En el sistema antifraude, Python se utilizó para desarrollar el portal administrativo de monitoreo, facilitando la visualización de transacciones bloqueadas y el análisis operativo por parte del área de riesgos.

C. Angular

Angular es un framework para el desarrollo de aplicaciones web que utiliza un enfoque basado en componentes, permitiendo estructurar la interfaz de usuario en módulos reutilizables. Esta arquitectura facilita la organización del código y la construcción de aplicaciones dinámicas y escalables (Google, 2023). Su modelo de componentes facilita la separación entre presentación y lógica de negocio. En el sistema antifraude, Angular se utilizó para implementar el servicio que interactúa con el motor de reglas, permitiendo gestionar solicitudes y respuestas de evaluación de riesgo dentro del flujo transaccional.

D. API (Interfaz de Programación de Aplicaciones)

Una API permite que diferentes sistemas de software intercambien información mediante interfaces estandarizadas, facilitando la interoperabilidad entre aplicaciones sin necesidad de acceder a su lógica interna, facilitando el intercambio de información sin necesidad de conocer la implementación interna de cada componente (Fielding, 2000). Su principal objetivo es desacoplar aplicaciones, permitiendo que distintos módulos funcionen de manera independiente pero coordinada.

En arquitecturas web modernas, las APIs suelen implementarse bajo el estilo arquitectónico REST (Representational State Transfer), el cual utiliza el protocolo HTTP para realizar operaciones sobre recursos mediante métodos como GET, POST, PUT y DELETE. Este modelo favorece la escalabilidad, simplicidad y compatibilidad entre sistemas heterogéneos (Richardson & Ruby, 2007).

Dentro del sistema desarrollado, la API cumple el rol de intermediario entre el frontend desarrollado en Angular y el backend implementado en Python. A través de ella se gestionan las solicitudes de evaluación, consultas de información y envío de resultados del motor antifraude, garantizando una comunicación segura y estructurada mediante el uso de formatos de intercambio de datos como JSON. Asimismo, la API permite integrar el sistema con plataformas externas sin afectar la lógica interna del sistema.

E. SQL Server

SQL Server es un sistema de administración de bases de datos relacionales desarrollado por Microsoft que permite almacenar y gestionar información estructurada utilizando el lenguaje de consultas Transact-SQL (T-SQL). Este sistema proporciona mecanismos de seguridad, integridad de datos y control de transacciones para aplicaciones empresariales (Microsoft, 2022). Proporciona mecanismos de integridad, seguridad y control transaccional que garantizan la consistencia de los datos. En el sistema antifraude, SQL Server se emplea para almacenar las reglas configurables, catálogos de evaluación y el historial de decisiones del motor de reglas. Esto permite modificar condiciones de evaluación sin alterar el código de la aplicación, garantizando trazabilidad y auditoría de las transacciones analizadas.

Durante el desarrollo e implementación del motor de reglas antifraude se identificaron diversas limitaciones técnicas, operativas y arquitectónicas que influyen en el alcance del sistema y en su comportamiento dentro de entornos productivos.

El motor antifraude se basa en un modelo determinístico de evaluación mediante reglas predefinidas. Este enfoque permite una alta explicabilidad y control del proceso de decisión; sin embargo, limita la capacidad del sistema para detectar patrones de fraude desconocidos o dinámicos. Los ataques novedosos o adaptativos pueden evadir las reglas existentes hasta que estas sean actualizadas manualmente, lo que genera una dependencia directa de la experiencia del analista antifraude.

2.2 Limitaciones

Durante el desarrollo e implementación del motor de reglas antifraude se identificaron diversas limitaciones técnicas, operativas y arquitectónicas que influyen en el alcance del sistema y en su comportamiento dentro de entornos productivos.

En primer lugar, el motor antifraude se basa en un modelo determinístico de evaluación mediante reglas predefinidas. Este enfoque permite una alta explicabilidad y control del proceso de decisión; sin embargo, limita la capacidad del sistema para detectar patrones de fraude desconocidos o dinámicos. Los ataques novedosos o adaptativos pueden evadir las reglas existentes hasta que estas sean actualizadas manualmente, lo que genera una dependencia directa de la experiencia del analista antifraude.

Desde el punto de vista operativo, la gestión de reglas requiere intervención manual por parte de analistas especializados. La creación, ajuste y priorización de reglas depende del conocimiento del comportamiento transaccional y del análisis histórico de fraude, lo que introduce un factor humano susceptible a errores de configuración o retrasos en la respuesta ante nuevas modalidades de fraude.

CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

En julio de 2025 asumí el cargo de Technical Product Owner del equipo de Payin en Kashio Technology, teniendo como principal responsabilidad el liderazgo técnico de iniciativas estratégicas vinculadas a la recepción de pagos digitales. Actualmente continúo desempeñando dicho rol, orientado a la optimización del flujo transaccional, la escalabilidad de la arquitectura tecnológica y la mitigación de riesgos operativos asociados a los pagos electrónicos.

Dentro de las iniciativas más relevantes que lideré se encuentra el proyecto de implementación del Motor de Reglas Antifraude (KFRAUD 2.0), el cual nació como una necesidad estratégica del negocio ante el crecimiento del volumen transaccional y la ausencia de un sistema estructurado de prevención de fraude en el flujo de Payins, particularmente en pagos mediante transferencia bancaria.

La solución fue diseñada bajo una arquitectura de microservicios desacoplados, permitiendo integrar el servicio antifraude dentro del flujo Payins antes de la confirmación final de la transacción. El backend del motor antifraude fue desarrollado en Python, permitiendo gestionar lógica de negocio compleja, ejecución dinámica de reglas y procesamiento eficiente de validaciones en tiempo real. Por su parte, el frontend de monitoreo y gestión antifraude fue desarrollado en Angular, permitiendo una interfaz modular y dinámica orientada al análisis de transacciones en estado de revisión.

El núcleo del sistema se apoyó en el componente Kashio Business Rules Engine (KBRE), encargado de centralizar y ejecutar las reglas antifraude configurables. Esta centralización permitió desacoplar la lógica de validación del flujo principal de pagos, facilitando su evolución y mantenimiento sin impactar la operación productiva.

La información transaccional y los registros de auditoría fueron almacenados en SQL Server, garantizando integridad, trazabilidad y consistencia de datos. El control de versiones se gestionó mediante Git, y los despliegues se realizaron en entornos de desarrollo, staging y producción, asegurando estabilidad y validación progresiva. Durante el proyecto asumí responsabilidades de:

- Definición y priorización de historias de usuario.
- Traducción de requerimientos de negocio a especificaciones técnicas.
- Refinamiento técnico con equipos backend (Python) y frontend (Angular).
- Validación de criterios de aceptación.
- Identificación y mitigación de riesgos técnicos.
- Alineación estratégica con Product Management.

3.1 Acuerdo de Confidencialidad

El presente trabajo de suficiencia profesional se desarrolló para la empresa Kashio Technology, organización fintech especializada en soluciones de pagos digitales (Payins y Payouts). Al manejar información financiera, transaccional y datos sensibles de clientes, así como integraciones con entidades bancarias y Proveedores de Servicios de Pago (PSPs), la empresa mantiene políticas estrictas de seguridad de la información y acuerdos de confidencialidad con todos sus colaboradores, con el fin de proteger la integridad, disponibilidad y confidencialidad de los datos procesados en su plataforma.

El proyecto del Motor de Reglas Antifraude (KFRAUD 2.0), integrado al flujo de Payins específicamente en pagos mediante transferencia bancaria, fue desarrollado respetando dichos lineamientos. Debido a la naturaleza crítica del sistema, no se exponen reglas específicas, parámetros internos de scoring, configuraciones de seguridad ni

credenciales que puedan comprometer la arquitectura productiva o la estrategia antifraude de la organización.

El servicio antifraude fue construido tomando como base el componente Kashio Business Rules Engine (KBRE), el cual centraliza las reglas de validación aplicadas a las transacciones. La solución se desarrolló bajo una arquitectura de microservicios desacoplados, integrados mediante APIs internas, con almacenamiento en base de datos SQL Server y despliegue en entornos diferenciados (desarrollo, staging y producción).

El presente documento describe la experiencia profesional desde un enfoque metodológico, técnico y estratégico, preservando la confidencialidad y la propiedad intelectual de la organización.

3.2 Descripción problemática

El crecimiento sostenido de las transacciones digitales dentro del flujo de Payins en Kashio Technology generó un incremento significativo en la exposición al riesgo transaccional. A medida que aumentaba el volumen de pagos mediante transferencia bancaria, se evidenció que la plataforma no contaba con un sistema antifraude estructurado, centralizado y escalable que permitiera evaluar de manera integral el riesgo asociado a cada operación.

En la etapa inicial del diagnóstico se identificó que las validaciones existentes eran básicas y se encontraban dispersas dentro del flujo transaccional, lo que dificultaba su mantenimiento, trazabilidad y evolución. Estas validaciones no estaban centralizadas en un motor configurable, lo que implicaba que cualquier ajuste requería modificaciones

directas en el código del servicio Payins, incrementando el riesgo técnico y reduciendo la agilidad para adaptarse a nuevos patrones de fraude.

Asimismo, no existía un modelo de estados que permitiera clasificar las transacciones según su nivel de riesgo. Las operaciones se aprobaban o rechazaban de manera directa, sin contemplar un estado intermedio de revisión manual. Esta limitación impedía aplicar controles adicionales en transacciones sospechosas, reduciendo la capacidad de análisis preventivo y aumentando la probabilidad de pérdidas financieras o afectación reputacional.

Otro problema identificado fue la ausencia de trazabilidad detallada sobre las decisiones antifraude. No se contaba con un registro estructurado que permitiera auditar qué reglas se ejecutaron, cuáles se cumplieron o qué variables influyeron en la decisión final. Esta situación dificultaba la generación de reportes, la mejora continua del sistema y el aprendizaje organizacional frente a incidentes de fraude.

Desde una perspectiva tecnológica, el flujo Payins no estaba desacoplado de la lógica de validación. Esto implicaba que el sistema no podía escalar de manera independiente ni evolucionar hacia un modelo más sofisticado de gestión de riesgo. Además, ante el crecimiento proyectado del negocio, la arquitectura existente podía convertirse en un cuello de botella operativo.

En términos estratégicos, la organización enfrentaba el desafío de mantener la experiencia del usuario sin introducir fricción excesiva en el proceso de pago. Implementar controles antifraude más estrictos sin afectar la conversión transaccional representaba un equilibrio crítico entre seguridad y usabilidad.

En consecuencia, la problemática central se definió como: La ausencia de un motor antifraude centralizado, configurable y desacoplado del flujo Payins que permita evaluar en tiempo real el riesgo transaccional, garantizar trazabilidad de decisiones y gestionar de manera eficiente operaciones sospechosas sin afectar la escalabilidad ni la experiencia del usuario.

Esta situación motivó la formulación y ejecución del proyecto KFRAUD 2.0, cuyo propósito fue diseñar e implementar una solución tecnológica robusta que permitiera mitigar los riesgos identificados, fortalecer la arquitectura del sistema y proporcionar a la organización un mecanismo estructurado de prevención y control del fraude digital.

3.3 Objetivo del Proyecto

Diseñar e implementar un Motor de Reglas Antifraude (KFRAUD 2.0) integrado al flujo transaccional de Payins específicamente en pagos mediante transferencia bancaria en la plataforma de Kashio Technology, con el propósito de evaluar en tiempo real el riesgo asociado a cada operación a través de reglas configurables centralizadas en el Kashio Business Rules Engine (KBRE), garantizando trazabilidad, escalabilidad, control operativo y mitigación del riesgo financiero sin afectar la experiencia del usuario.

3.4 Alcance del Proyecto

El alcance del proyecto comprendió el diseño e implementación del Motor de Reglas Antifraude PAYIN (KFRAUD 2.0) para proporcionar una evaluación antifraude en tiempo real dentro del flujo transaccional de Payins, específicamente en pagos mediante transferencia bancaria. El sistema fue integrado en dos puntos críticos del proceso:

1. Creación de deuda (generación de la orden de pago), donde se evalúan variables preliminares asociadas al usuario, comportamiento histórico y características de la operación antes de permitir su registro definitivo.

2. Pago de deuda (confirmación de la transacción), donde se realiza una segunda evaluación antifraude antes de aprobar el procesamiento final del pago.

El proyecto incluyó el desarrollo de un servicio antifraude desacoplado del flujo principal, implementado en Python, responsable de ejecutar reglas configurables en tiempo real y retornar una decisión transaccional. Asimismo, se contempló la integración mediante APIs REST internas, permitiendo la comunicación estructurada entre el servicio Payin y el motor de reglas. Como parte del alcance funcional, se desarrolló:

- Un módulo de gestión de reglas orientado al área de riesgos, permitiendo la configuración, activación y parametrización de validaciones sin necesidad de modificar directamente el código del sistema.

- Un gestor de transacciones bloqueadas o en estado de revisión, implementado mediante una interfaz en Angular, que permite al equipo de riesgos analizar, aprobar o rechazar operaciones sospechosas.

- Las APIs necesarias para la toma de decisiones automáticas y la administración del motor, garantizando trazabilidad y consistencia en cada evaluación realizada.

El sistema fue diseñado para consultar catálogos internos (listas negras, parámetros configurables, históricos de comportamiento u otros registros relevantes) y aplicar reglas clasificadas en cinco niveles de criticidad (N1 a N5), permitiendo una evaluación

escalonada del riesgo. Esta clasificación posibilita diferenciar reglas informativas de reglas bloqueantes, fortaleciendo la precisión en la toma de decisiones.

El alcance incluyó las fases de análisis funcional, diseño arquitectónico, desarrollo backend en Python, desarrollo frontend en Angular, integración con el flujo Payin, pruebas funcionales y técnicas, y despliegue en producción.

3.5 Equipo scrum

Tabla 1

Equipo Scrum

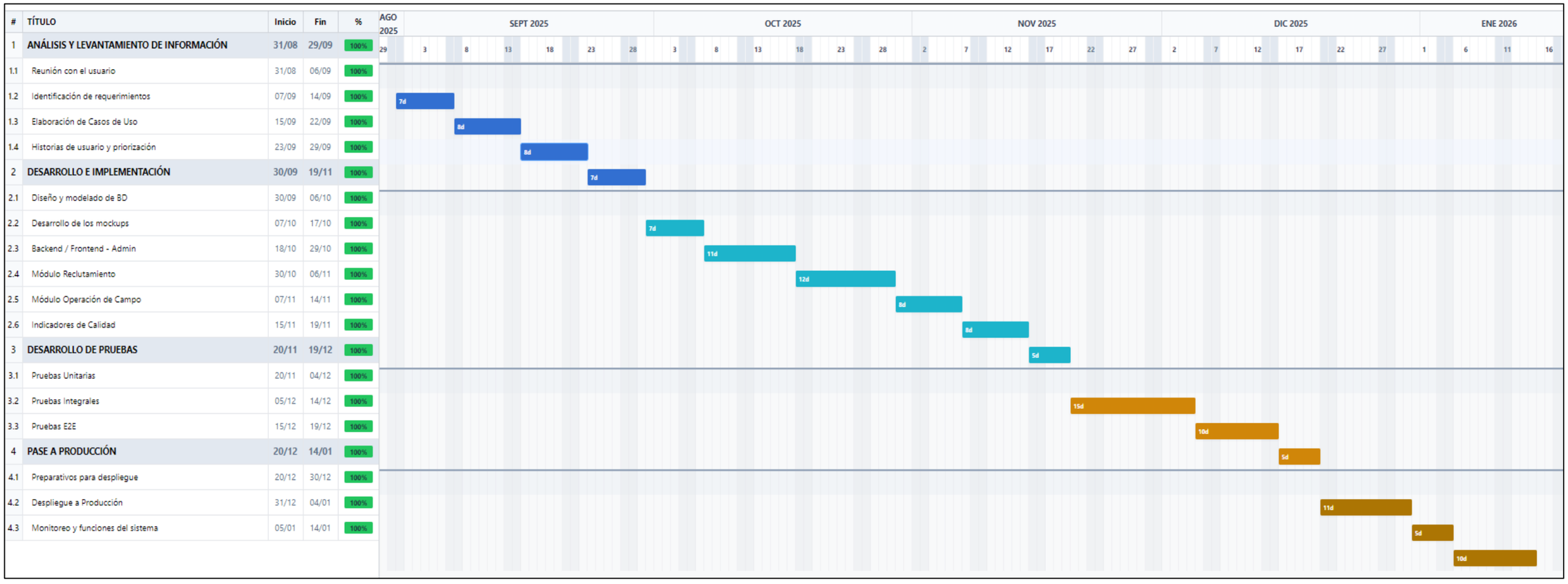
Persona	Cargo	Rol
Juan Carlos Solorzano	Gestor de Proyecto	Product Owner
Karina Limaylla Lujan	Líder de TI	Scrum Master
Jesús Guanga	Líder Técnico	Development team

Nota: Integrantes del equipo scrum, elaborado por el presente autor.

3.6 Cronograma

Figura 3

Cronograma de actividades



#	TÍTULO	Inicio	Fin	%
1	**ANÁLISIS Y LEVANTAMIENTO DE INFORMACIÓN**	31/08	29/09	100%
1.1	Reunión con el usuario	31/08	06/09	100%
1.2	Identificación de requerimientos	07/09	14/09	100%
1.3	Elaboración de Casos de Uso	15/09	22/09	100%
1.4	Historias de usuario y priorización	23/09	29/09	100%
2	**DESARROLLO E IMPLEMENTACIÓN**	30/09	19/11	100%
2.1	Diseño y modelado de BD	30/09	06/10	100%
2.2	Desarrollo de los mockups	07/10	17/10	100%
2.3	Backend / Frontend - Admin	18/10	29/10	100%
2.4	Módulo Reclutamiento	30/10	06/11	100%
2.5	Módulo Operación de Campo	07/11	14/11	100%
2.6	Indicadores de Calidad	15/11	19/11	100%
3	**DESARROLLO DE PRUEBAS**	20/11	19/12	100%
3.1	Pruebas Unitarias	20/11	04/12	100%
3.2	Pruebas Integrales	05/12	14/12	100%
3.3	Pruebas E2E	15/12	19/12	100%
4	**PASE A PRODUCCIÓN**	20/12	14/01	100%
4.1	Preparativos para despliegue	20/12	30/12	100%
4.2	Despliegue a Producción	31/12	04/01	100%
4.3	Monitoreo y funciones del sistema	05/01	14/01	100%

Nota: El cronograma de actividades detalla la planificación temporal del proyecto, estructurado en fases progresivas con dependencias definidas, contemplando análisis, diseño arquitectónico, desarrollo del motor antifraude, integración de servicios, pruebas funcionales y despliegue. Elaborado por el presente autor.

3.7 Requerimientos funcionales

Tabla 2

Requisitos funcionales

Id	Requisito	Descripción
RF-01	Evaluación en creación de deuda	El sistema deberá evaluar antifraude en tiempo real cuando se genere una deuda en el flujo Payin, antes de permitir continuar con el proceso.
RF-02	Evaluación en pago de deuda	El sistema deberá ejecutar una validación antifraude antes de confirmar el pago de la deuda.
RF-03	Niveles de criticidad N1–N5	El motor deberá aplicar reglas clasificadas en cinco niveles de criticidad (N1 a N5), determinando si la operación se aprueba, rechaza o envía a revisión.
RF-04	Gestión de reglas	El sistema deberá permitir crear, editar, activar y desactivar reglas antifraude desde un módulo administrado por el área de riesgos.
RF-05	Gestión de transacciones en revisión	El sistema deberá permitir visualizar y decidir (aprobar o rechazar) las transacciones marcadas como Review.
RF-06	Registro de evaluaciones	El sistema deberá almacenar el detalle de cada evaluación, incluyendo reglas aplicadas y decisión final, para fines de auditoría.
RF-07	APIs de integración	El sistema deberá exponer APIs para evaluar transacciones y consultar su estado dentro del flujo Payin.

Id	Requisito	Descripción
RF-08	Consulta de catálogos	El motor deberá consultar listas internas y parámetros configurables para validar las transacciones.
RF-09	Gestión de estados	El sistema deberá asignar automáticamente el estado Approved, Rejected o Review según el resultado del análisis.
RF-10	Reportes de riesgo	El sistema deberá generar reportes sobre transacciones evaluadas y reglas activadas para el área de riesgos.

Nota: Listado de requerimiento funcionales elaborado por el presente autor.

3.8 Requerimientos no funcionales

Tabla 3

Requisitos no funcionales

Código	Requisito	Descripción
RNF-01	Rendimiento	El sistema deberá responder a la evaluación antifraude en tiempo real sin afectar el flujo de Payin.
RNF-02	Disponibilidad	El sistema deberá estar disponible 24/7 para garantizar la continuidad operativa del servicio Payin.
RNF-03	Seguridad	El sistema deberá proteger la información transaccional medi mecanismos de autenticación y autorización.
RNF-04	Escalabilidad	El sistema deberá soportar incrementos en el volumen de

Código	Requisito	Descripción
		transacciones sin degradar su desempeño.
RNF-05	Trazabilidad	El sistema deberá registrar todas las evaluaciones y decisione para fines de auditoría.
RNF-06	Integridad de da	El sistema deberá garantizar que la información evaluada no s alterada durante el proceso.
RNF-07	Mantenibilidad	El sistema deberá permitir la configuración de reglas sin necesidad de modificar el código fuente.
RNF-08	Compatibilidad	El sistema deberá integrarse con los servicios existentes del ecosistema Payin mediante APIs REST.
RNF-09	Confiabilidad	El sistema deberá asegurar consistencia en la decisión antifra ante solicitudes repetidas.
RNF-10	Auditoría	El sistema deberá permitir la generación de logs y reportes pa control interno y cumplimiento normativo.

Nota: Listado de requerimiento no funcionales elaborado por el presente autor.

3.9 Historias de usuario

Tabla 4

Historias de usuarios

Código	Módulo	Nombre	Descripción
HU-01	Evaluación Payin	Validación al crear deuda	Como sistema Payin quiero validar antifraude al momento de crear una deuda para evitar que se generen órdenes de pago con riesgo potencial desde el inicio del flujo.
HU-02	Evaluación Payin	Validación antes de confirmar pago	Como sistema Payin quiero ejecutar una validación antifraude antes de confirmar el pago para asegurar que solo se procesen transacciones seguras.
HU-03	Gestión de Reglas	Crear regla antifraude	Como analista de riesgos quiero crear nuevas reglas antifraude para responder rápidamente a nuevos patrones de fraude detectados.
HU-04	Gestión de Reglas	Configurar criticidad de regla	Como analista de riesgos quiero definir el nivel de criticidad (N1–N5) de una regla para determinar si la transacción se aprueba, se revisa o se bloquea

Código	Módulo	Nombre	Descripción
			automáticamente.
HU-05	Gestión de Reglas	Activar o desactivar reglas	Como analista de riesgos quiero activar o desactivar reglas sin intervención técnica para ajustar el control según el comportamiento del mercado.
HU-06	Gestión de Transacciones	Revisar transacciones sospechosas	Como analista de riesgos quiero visualizar las transacciones en estado Review para analizarlas y decidir su aprobación o rechazo.
HU-07	Gestión de Transacciones	Decidir manualmente una operación	Como analista de riesgos quiero aprobar o rechazar manualmente una transacción observada para controlar el riesgo sin afectar innecesariamente la operación del cliente.
HU-08	Auditoría	Consultar historial de evaluación	Como auditor interno quiero consultar el historial de evaluaciones antifraude para verificar la trazabilidad y cumplimiento de políticas internas.
HU-09	Integración	Obtener decisión	Como sistema Payin quiero recibir una

Código	Módulo	Nombre	Descripción
		antifraude vía API	respuesta estructurada del motor antifraude para continuar el flujo según la decisión obtenida.
HU-10	Reportes	Monitorear comportamiento de fraude	Como área de riesgos quiero visualizar reportes de reglas activadas y transacciones evaluadas para identificar tendencias y fortalecer la estrategia antifraude.

Nota: Historia de usuario elaborado por el presente autor.

3.10 Ambiente de desarrollo

Tabla 5

Ambiente de desarrollo

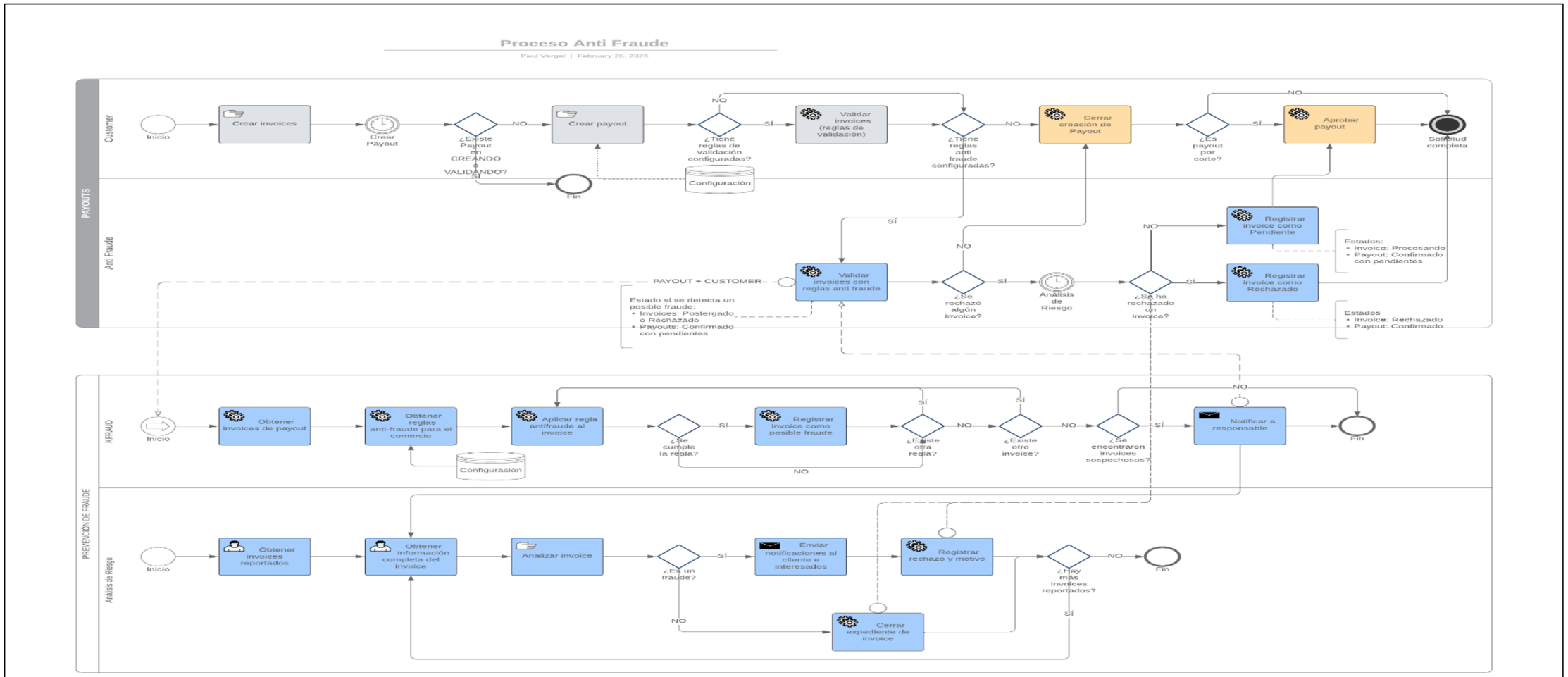
Tipo	Software
Base de datos	SQL SERVER
Lenguaje de programación	PHYTON TYPESCRIPT NODEJS

Nota: Ambientes para el desarrollo elaborado por el presente autor.

3.11 Arquitectura de software

Figura 4

Arquitectura de proceso de fraude

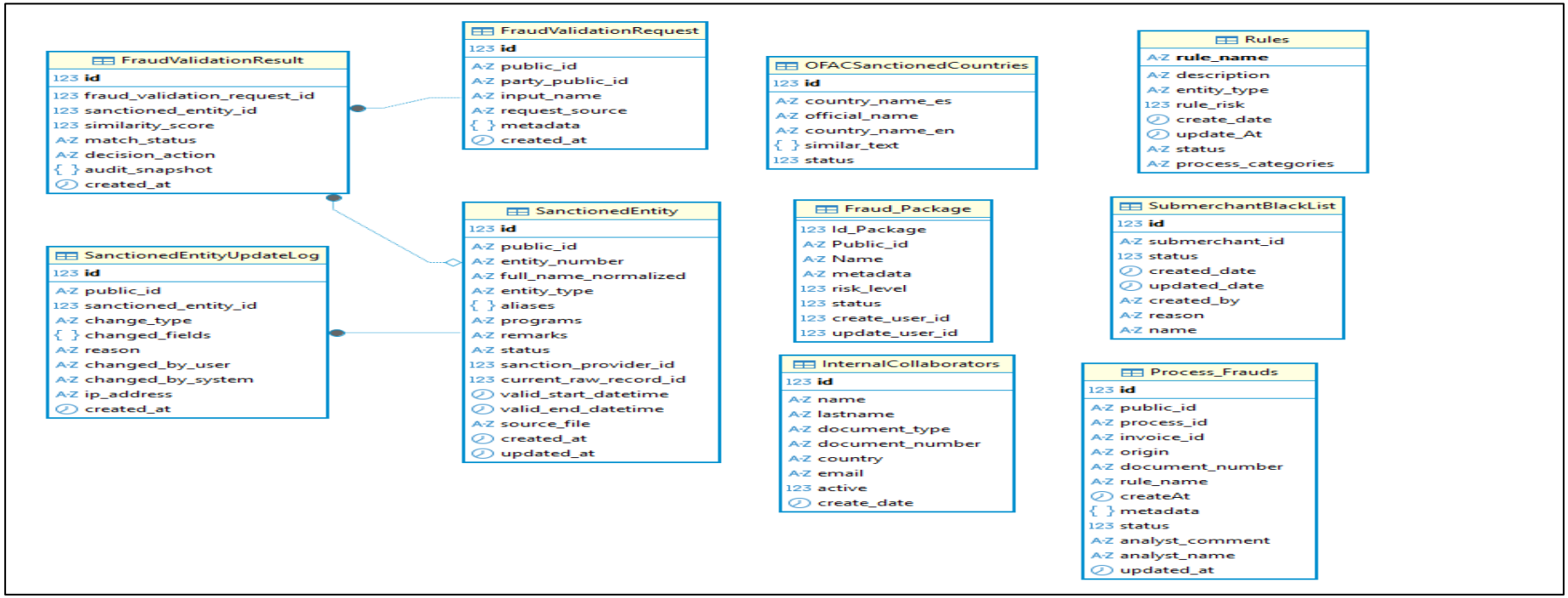


Nota: La arquitectura implementada garantiza separación de responsabilidades, seguridad en la comunicación entre servicios y eficiencia en el procesamiento de transacciones en tiempo real, permitiendo una gestión robusta y escalable del riesgo transaccional, elaborado por el autor presente.

3.12 Diagrama entidad relación

Figura 5

Diagrama del modelo entidad relación



Nota: Diagrama de base de datos elaborado por el autor presente.

Diccionario de datos

El diccionario de datos del módulo **Motor de Reglas Antifraude PAYIN (KFRAUD 2.0)** describe de manera estructurada todas las entidades, atributos y relaciones que conforman la base de datos del sistema. Su finalidad es documentar técnicamente la estructura de almacenamiento de información, garantizando claridad en la definición de cada campo, su tipo de dato, claves primarias y foráneas, así como su propósito dentro del proceso de evaluación antifraude.

Tabla 6

Descripción de las tablas de la base de datos

Solicitudes de validación de fraude ingresadas al sistema.

Esquema	Tabla	Descripción
INT	id	Identificador único de la solicitud
VARCHAR	public_id	Identificador público de la solicitud
VARCHAR	party_public_id	Identificador público de la parte involucrada
VARCHAR	input_name	Nombre ingresado para validación
VARCHAR	request_source	Fuente u origen de la solicitud
JSON	metadata	Metadatos adicionales de la solicitud
TIMESTAMP	created_at	Fecha y hora de creación del registro

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 7

Resultados de la validación de fraude para cada solicitud.

Esquema	Tabla	Descripción
INT	id	Identificador único del resultado
INT	fraud_validation_request_id	FK a FraudValidationRequest
INT	sanctioned_entity_id	FK a SanctionedEntity si hay coincidencia
INT	similarity_score	Puntuación de similitud con entidad sancionada
VARCHAR	match_status	Estado del match (ej. HIT, NO_HIT)
VARCHAR	decision_action	Acción tomada como resultado de la validación
JSON	audit_snapshot	Snapshot del estado al momento de la auditoría
TIMESTAMP	created_at	Fecha y hora de creación del resultado

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 8

Entidades que figuran en listas de sanciones internacionales (OFAC, etc.).

Esquema	Tabla	Descripción
INT	id	Identificador único de la entidad
VARCHAR	public_id	Identificador público de la entidad
VARCHAR	entity_number	Número de entidad en la lista de sanciones
VARCHAR	full_name_normalized	Nombre completo normalizado para búsqueda
VARCHAR	entity_type	Tipo de entidad (persona, empresa, etc.)
JSON	aliases	Lista de alias conocidos de la entidad
VARCHAR	programs	Programas de sanción a los que pertenece
VARCHAR	remarks	Observaciones adicionales
VARCHAR	status	Estado activo/inactivo de la entidad en la lista

INT	sanction_provider_id	Proveedor de la lista de sanciones
INT	current_raw_record_id	Referencia al registro raw actual
TIMESTAMP	valid_start_datetime	Inicio de vigencia del registro
TIMESTAMP	valid_end_datetime	Fin de vigencia del registro
VARCHAR	source_file	Archivo fuente de la entidad
TIMESTAMP	created_at	Fecha de creación del registro
TIMESTAMP	updated_at	Fecha de última actualización

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 9

Lista de países sancionados por OFAC.

Esquema	Tabla	Descripción
INT	Id	Identificador único del país
VARCHAR	country_name_es	Nombre del país en español
VARCHAR	official_name	Nombre oficial del país
VARCHAR	country_name_en	Nombre del país en inglés
JSON	similar_text	Textos similares usados para matching

INT	status	Estado activo/inactivo del registro

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 10

Reglas de negocio utilizadas en la detección de fraude.

Esquema	Tabla	Descripción
VARCHAR	rule_name	Nombre único de la regla (PK)
VARCHAR	description	Descripción de la regla
VARCHAR	entity_type	Tipo de entidad a la que aplica la regla
INT	rule_risk	Nivel de riesgo asociado a la regla
TIMESTAMP	create_date	Fecha de creación de la regla
TIMESTAMP	update_At	Fecha de última actualización
VARCHAR	status	Estado activo/inactivo de la regla
VARCHAR	process_categories	Categorías de proceso donde aplica la regla

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 11

Paquetes de configuración de fraude agrupando reglas y parámetros.

Esquema	Tabla	Descripción
INT	Id_Package	Identificador único del paquete
VARCHAR	Public_id	Identificador público del paquete
VARCHAR	Name	Nombre del paquete
VARCHAR	metadata	Metadatos del paquete
VARCHAR	risk_level	Nivel de riesgo del paquete
INT	status	Estado activo/inactivo del paquete
INT	create_user_id	ID del usuario que creó el paquete
INT	update_user_id	ID del usuario que actualizó el paquete

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 12

Lista negra de submerchants bloqueados por actividad fraudulenta.

Esquema	Tabla	Descripción
INT	id	Identificador único del registro
VARCHAR	submerchant_id	Identificador del submerchant bloqueado

INT	status	Estado del bloqueo (activo/inactivo)
TIMESTAMP	created_date	Fecha de creación del registro
TIMESTAMP	updated_date	Fecha de última actualización
VARCHAR	created_by	Usuario que creó el registro
VARCHAR	reason	Razón del bloqueo
VARCHAR	name	Nombre del submerchant

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 13

Colaboradores internos con acceso al sistema de gestión de fraude.

Esquema	Tabla	Descripción
INT	id	Identificador único del colaborador
VARCHAR	name	Nombre del colaborador
VARCHAR	lastname	Apellido del colaborador
VARCHAR	document_type	Tipo de documento de identidad
VARCHAR	document_number	Número de documento de identidad
VARCHAR	country	País del colaborador
VARCHAR	email	Correo electrónico del colaborador

INT	active	Indicador de estado activo (1) o inactivo (0)
TIMESTAMP	create_date	Fecha de creación del registro

Nota: Diccionario de datos elaborado por el presente autor.

Tabla 14

Registro de procesos de fraude detectados o investigados.

Esquema	**Tabla**	**Descripción**
INT	id	Identificador único del proceso
VARCHAR	public_id	Identificador público del proceso
VARCHAR	process_id	ID del proceso relacionado
VARCHAR	invoice_id	ID de la factura relacionada
VARCHAR	origin	Origen del proceso de fraude
VARCHAR	document_number	Número de documento involucrado
VARCHAR	rule_name	Nombre de la regla que disparó el proceso
TIMESTAMP	createAt	Fecha de creación del proceso
JSON	metadata	Metadatos adicionales del proceso
INT	status	Estado del proceso

VARCHAR	analyst_comment	Comentario del analista que revisó el caso
VARCHAR	analyst_name	Nombre del analista asignado
TIMESTAMP	updated_at	Fecha de última actualización

Nota: Diccionario de datos elaborado por el presente autor.

3.13 Desarrollo de módulos

Figura 6

Fragmento de código de evaluación antifraude – Python (Backend)

```python

from rules_engine import evaluate_rules
from database import save_evaluation

def evaluate_transaction(transaction_data):
    result = evaluate_rules(transaction_data)
    decision = result["decision"]

    save_evaluation(transaction_data["id"], result)

    return {
        "transaction_id": transaction_data["id"],
        "decision": decision,
        "rules_triggered": result["rules"]
    }
```

Nota: En este código en Python se implementa la lógica principal del motor antifraude, conectándose a la capa de reglas para evaluar la transacción y almacenar el resultado en base de datos.

Figura 7

Fragmento de código API Gateway – Node.js con TypeScript

```typescript
// antifraud.controller.ts

import { Request, Response } from 'express';
import axios from 'axios';

export const evaluateTransaction = async (req: Request, res: Response) => {
    const transaction = req.body;

    const response = await axios.post(
        'http://antifraud-service/evaluate',
        transaction
    );

    res.json(response.data);
};
```

Nota: En este código en Node.js con TypeScript se muestra la capa de integración que permite recibir la solicitud desde Payin y reenviarla al microservicio antifraude desarrollado en Python.

Figura 8

Fragmento de código de gestión de reglas – Python

```python
def create_rule(data):
    rule = Rule(
        name=data["name"],
        condition=data["condition"],
        severity=data["severity"],
        active=True
    )
    rule.save()
    return rule
```

Nota: En este código se realiza el registro de nuevas reglas antifraude dentro del sistema, permitiendo su configuración dinámica sin modificar el núcleo del motor.

Figura 9

Fragmento de código Frontend – Angular (TypeScript)

```typescript
// rule.component.ts

createRule() {
  this.http.post('/api/rules', this.ruleForm.value)
    .subscribe(response => {
      console.log('Regla creada correctamente');
    });
}
```

Nota: En este código en Angular se muestra la parte del Frontend que permite visualizar el formulario de creación de reglas y enviar la información al backend mediante una API REST.

Figura 10

Fragmento de código de revisión manual – Python

```python
def review_transaction(evaluation_id, decision, user_id):
    evaluation = Evaluation.get(id=evaluation_id)
    evaluation.decision = decision
    evaluation.reviewed_by = user_id
    evaluation.save()
    return evaluation# Online Python compiler (interpreter) to
# Write Python 3 code in this online editor and run it.
print("Try programiz.pro")
```

Nota: En este código se implementa la funcionalidad que permite al analista de riesgos aprobar o rechazar manualmente una transacción clasificada como Review.

Figura 11

Fragmento de código Frontend – Bandeja de revisión (Angular)

```
// review.component.ts

approveTransaction(id: number) {
  this.http.post(`/api/review/${id}`, { decision: 'Approved' })
    .subscribe(() => {
      alert('Transacción aprobada');
    });
}
```

Nota: En este código en Angular se muestra la interacción del usuario con la bandeja de revisión, permitiendo enviar la decisión manual al backend.

Figura 12

Fragmento de código Frontend – Visualización de reportes (Angular)

```
// review.component.ts
// report.component.ts

loadReport() {
  this.http.get('/api/report')
    .subscribe(data => {
      this.reportData = data;
    });
}
```

Nota: En este código se registra la trazabilidad de las acciones realizadas dentro del sistema, garantizando auditoría y cumplimiento normativo.

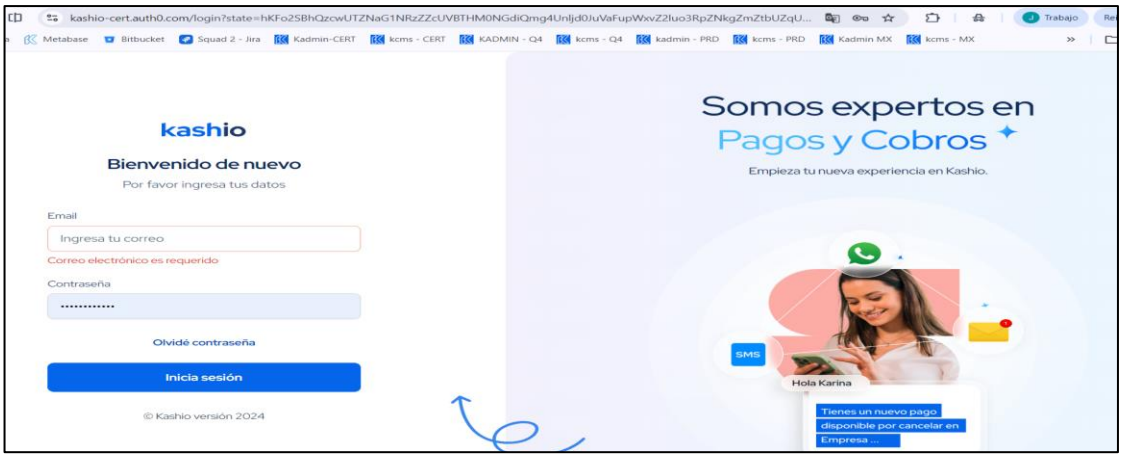
3.14 Funcionalidades del software

El sistema web **Motor de Reglas Antifraude PAYIN (KFRAUD 2.0)** integra una serie de funcionalidades orientadas a fortalecer la prevención, detección y gestión de operaciones sospechosas dentro del flujo transaccional PAYIN.

Estas herramientas permiten centralizar la evaluación de riesgo, automatizar decisiones antifraude, administrar bloqueos y garantizar trazabilidad completa de las operaciones, contribuyendo a una gestión eficiente y segura de las transacciones financieras.

Figura 13

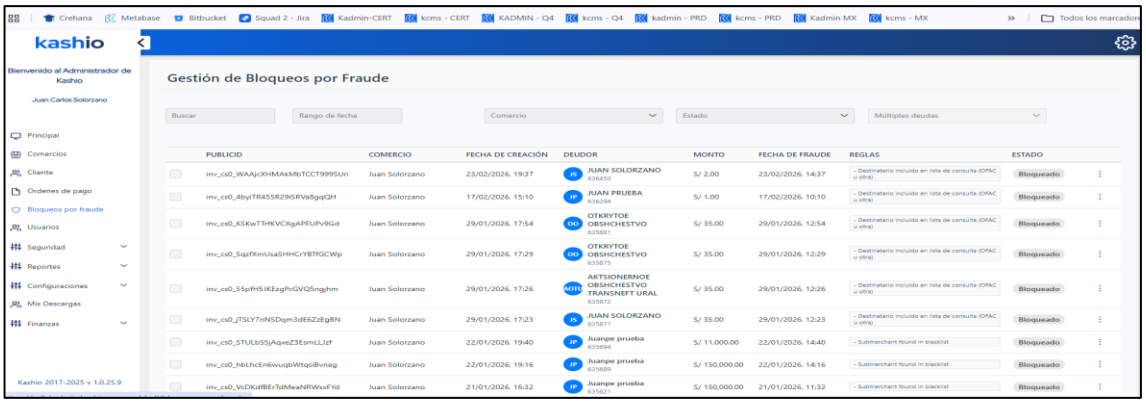
Prototipo de acceso al sistema antifraude.



Nota: El sistema permite el acceso mediante autenticación segura, validando credenciales y asignando permisos según el rol del usuario.

Figura 14

Prototipo de panel principal del módulo antifraude.



Nota: El panel principal muestra un resumen de indicadores clave como transacciones evaluadas, bloqueos por fraude, reglas activas y niveles de criticidad detectados, permitiendo una visión general del estado del sistema.

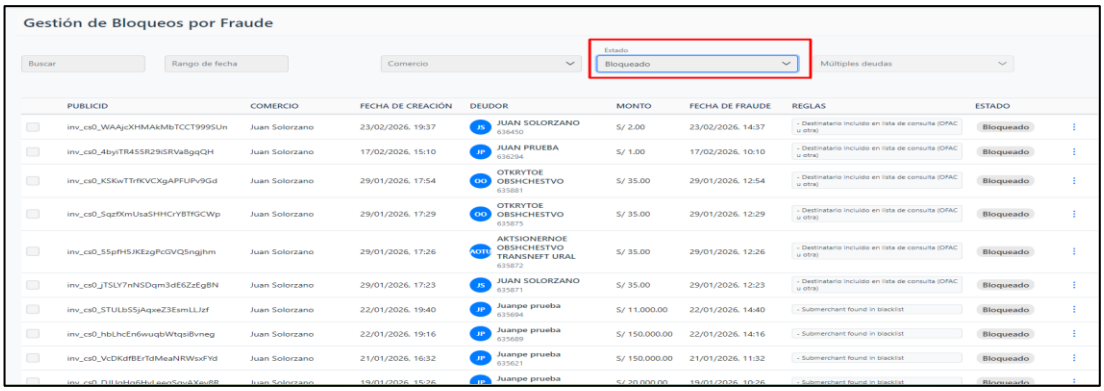
Figura 15

Prototipo de evaluación automática de transacciones.

Nota: Esta funcionalidad permite que el sistema ejecute reglas antifraude en tiempo real durante la creación o pago de una deuda, clasificando la operación como Approved, Review o Blocked.

Figura 16

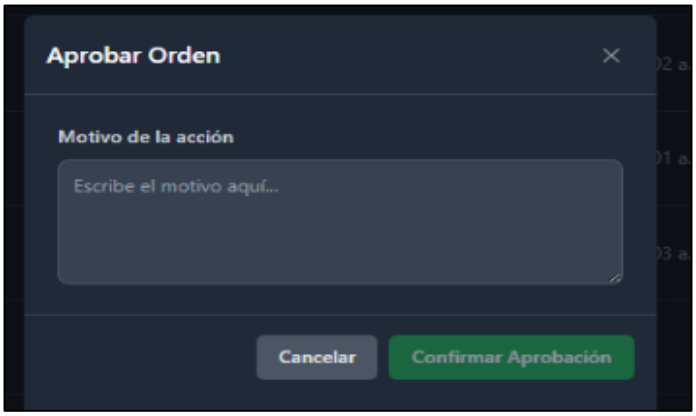
Prototipo de búsqueda de transacciones bloqueadas.



Nota: La interfaz permite visualizar las transacciones bloqueadas, mostrando información como identificador, comercio, deudor, monto, fecha y regla activada.

Figura 17

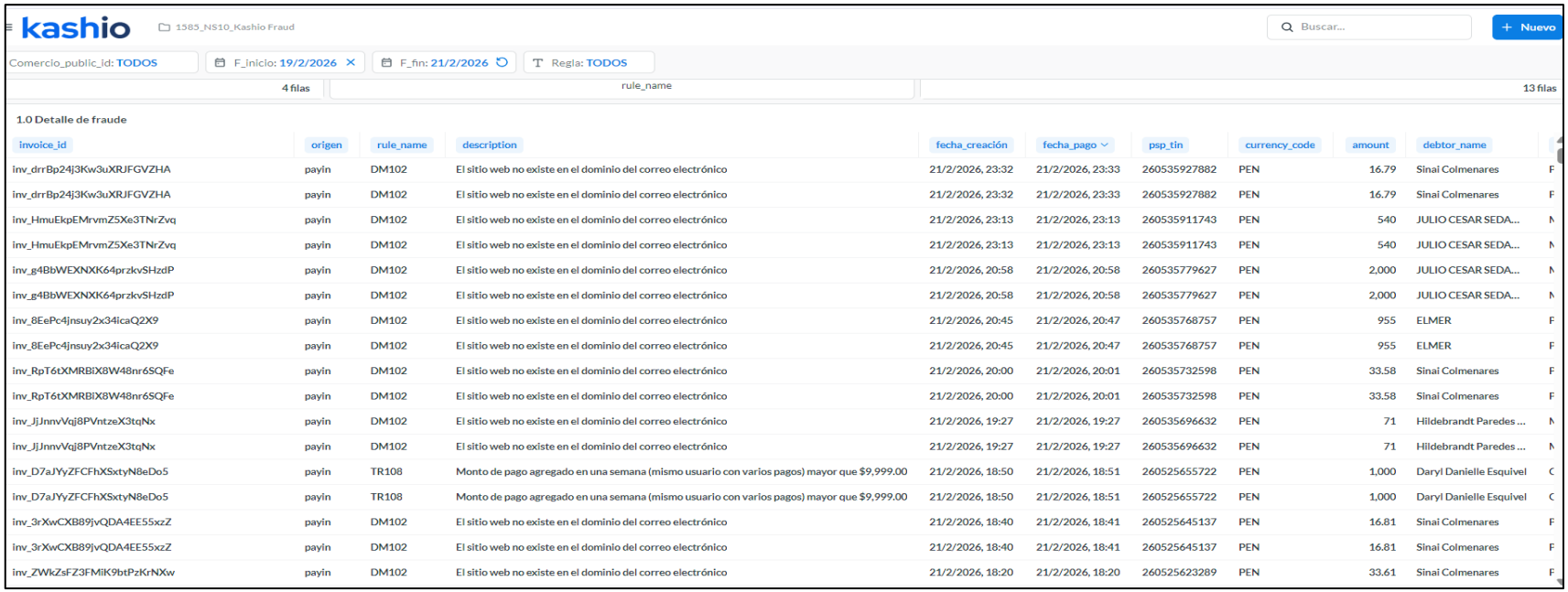
Prototipo de revisión manual de transacciones.



Nota: Cuando una transacción es clasificada como Review, el sistema permite al analista visualizar las reglas disparadas y tomar una decisión manual (aprobar o rechazar).

Figura 18

Prototipo de reporte de transacciones bloqueadas.

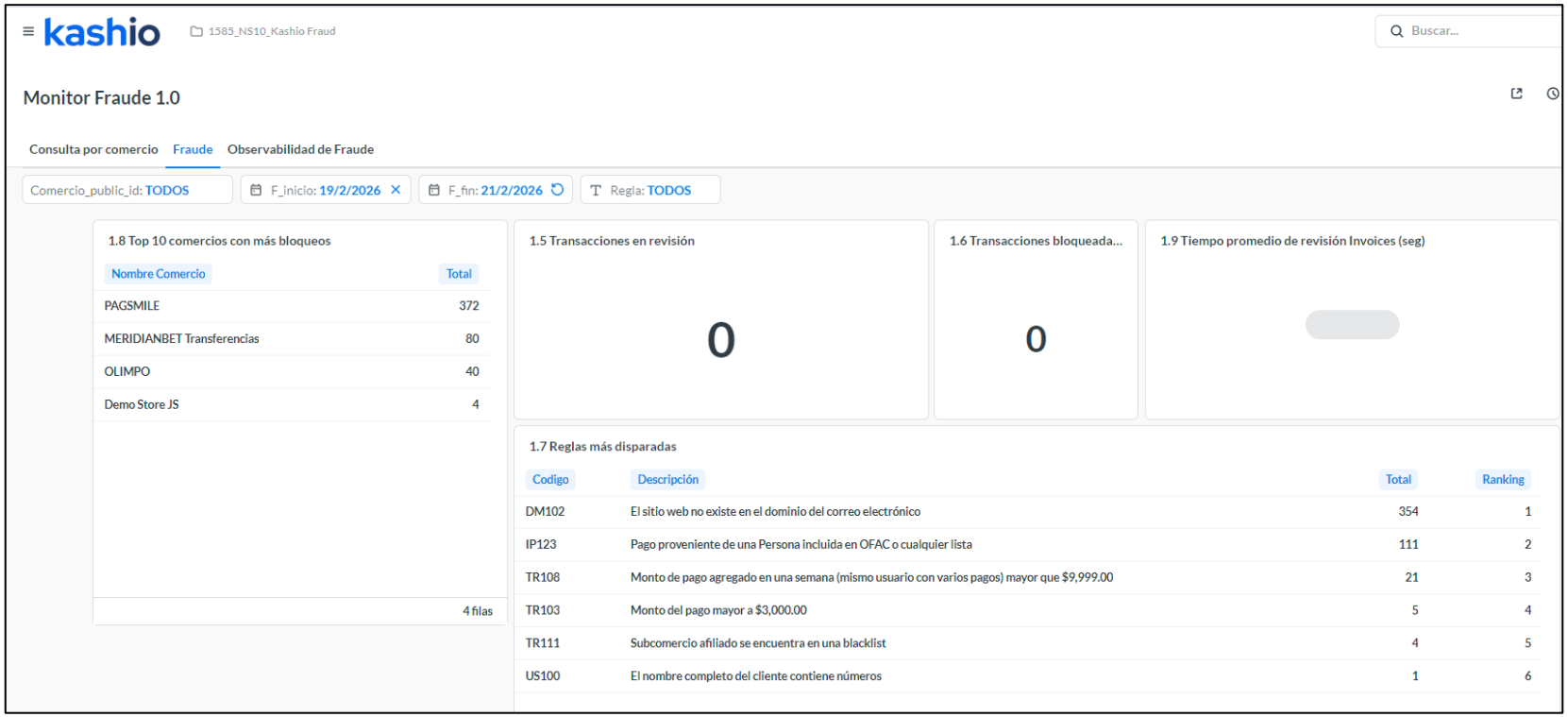


invoice_id	origen	rule_name	description	fecha_creación	fecha_pago	psp_tin	currency_code	amount	debtor_name	
inv_drrBp24j3Kw3uXRJFGVZHA	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 23:32	21/2/2026, 23:33	260535927882	PEN	16.79	Sinai Colmenares	F
inv_drrBp24j3Kw3uXRJFGVZHA	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 23:32	21/2/2026, 23:33	260535927882	PEN	16.79	Sinai Colmenares	F
inv_HmuEkpEMrvmZ5Xe3TNrZvq	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 23:13	21/2/2026, 23:13	260535911743	PEN	540	JULIO CESAR SEDA...	N
inv_HmuEkpEMrvmZ5Xe3TNrZvq	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 23:13	21/2/2026, 23:13	260535911743	PEN	540	JULIO CESAR SEDA...	N
inv_g4BbWEXNXK64przkvSHzdP	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 20:58	21/2/2026, 20:58	260535779627	PEN	2,000	JULIO CESAR SEDA...	N
inv_g4BbWEXNXK64przkvSHzdP	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 20:58	21/2/2026, 20:58	260535779627	PEN	2,000	JULIO CESAR SEDA...	N
inv_8EePc4jnsuy2x34icaQ2X9	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 20:45	21/2/2026, 20:47	260535768757	PEN	955	ELMER	F
inv_8EePc4jnsuy2x34icaQ2X9	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 20:45	21/2/2026, 20:47	260535768757	PEN	955	ELMER	F
inv_RpT6tXMRBlX8W48nr6SQFe	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 20:00	21/2/2026, 20:01	260535732598	PEN	33.58	Sinai Colmenares	F
inv_RpT6tXMRBlX8W48nr6SQFe	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 20:00	21/2/2026, 20:01	260535732598	PEN	33.58	Sinai Colmenares	F
inv_JjJnnvVqj8PVntzeX3tqNx	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 19:27	21/2/2026, 19:27	260535696632	PEN	71	Hildebrandt Paredes ...	N
inv_JjJnnvVqj8PVntzeX3tqNx	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 19:27	21/2/2026, 19:27	260535696632	PEN	71	Hildebrandt Paredes ...	N
inv_D7aJYyZFCFhXSxtyN8eDo5	payin	TR108	Monto de pago agregado en una semana (mismo usuario con varios pagos) mayor que $9,999.00	21/2/2026, 18:50	21/2/2026, 18:51	260525655722	PEN	1,000	Daryl Danielle Esquivel	C
inv_D7aJYyZFCFhXSxtyN8eDo5	payin	TR108	Monto de pago agregado en una semana (mismo usuario con varios pagos) mayor que $9,999.00	21/2/2026, 18:50	21/2/2026, 18:51	260525655722	PEN	1,000	Daryl Danielle Esquivel	C
inv_3rXwCXB89jvQDA4EE55xzZ	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 18:40	21/2/2026, 18:41	260525645137	PEN	16.81	Sinai Colmenares	F
inv_3rXwCXB89jvQDA4EE55xzZ	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 18:40	21/2/2026, 18:41	260525645137	PEN	16.81	Sinai Colmenares	F
inv_ZWkZsFZ3FMiK9btPzKrNXw	payin	DM102	El sitio web no existe en el dominio del correo electrónico	21/2/2026, 18:20	21/2/2026, 18:20	260525623289	PEN	33.61	Sinai Colmenares	F

Nota: Genera reportes consolidados sobre operaciones bloqueadas por período, tipo de regla y nivel de criticidad, apoyando la toma de decisiones estratégicas.

Figura 19

Prototipo de reporte estadístico de riesgo transaccional.



Nota: Permite visualizar tendencias de fraude, reglas más activadas y comportamiento transaccional mediante reportes gráficos y tabulares.

CAPÍTULO IV. RESULTADOS

La implementación del Motor de Reglas Antifraude PAYIN generó mejoras significativas en la gestión del riesgo transaccional y en la eficiencia operativa del área responsable del monitoreo y control de operaciones. La digitalización y automatización de estas validaciones permitió minimizar la exposición al riesgo financiero, optimizar los tiempos de respuesta ante operaciones sospechosas y fortalecer la trazabilidad de cada transacción procesada en el sistema.

El sistema ha potenciado la precisión en la clasificación de operaciones mediante la ejecución de reglas antifraude en tiempo real durante la creación y pago de deudas. Asimismo, la implementación de autenticación segura y control de acceso basado en roles ha reforzado la confidencialidad de la información y la segregación de funciones dentro del área de riesgos.

Del mismo modo, la generación automática de reportes estadísticos y operativos ha mejorado la capacidad de análisis del comportamiento transaccional, permitiendo detectar patrones inusuales, evaluar el impacto de las reglas implementadas y optimizar continuamente la estrategia antifraude.

Resultado de evaluación y respuesta sistema antifraude

Tabla 15

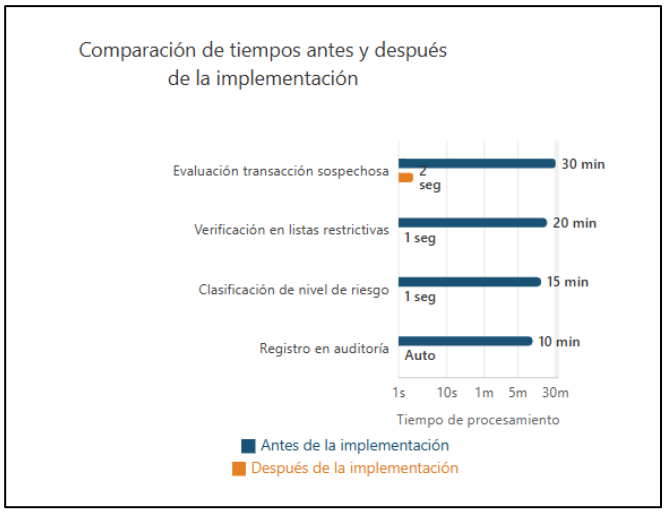
Tiempo de evaluación y respuesta antifraude

Indicador	Antes del sistema	Después del sistema	Mejora
Evaluación manual de transacción sospechosa	30 minutos	2 segundos	Tiempo real
Verificación en listas restrictivas	20 minutos	1 segundo	Tiempo real
Clasificación de nivel de riesgo	15 minutos	1 segundo	Tiempo real
Registro en auditoría	10 minutos	Automático	Tiempo real

Nota. Registro de tipo de evaluación elaborado por el presente autor.

Figura 20

Tiempo de evaluación y respuesta antifraude



Nota. Elaborado por el presente autor.

Análisis e interpretación de la Tabla 15 y la Figura 20

Se observa una mejora significativa en el tiempo de evaluación de transacciones. Antes de la implementación del sistema antifraude, el análisis de operaciones sospechosas se realizaba manualmente, lo cual implicaba revisar información en múltiples fuentes y validar datos de manera independiente, generando demoras considerables.

Con la implementación del Motor de Reglas Antifraude, la evaluación se ejecuta automáticamente en tiempo real mediante reglas configuradas dinámicamente, reduciendo los tiempos de respuesta de minutos a segundos. El gráfico permite visualizar claramente la reducción drástica en los tiempos de análisis, mejorando la eficiencia operativa y permitiendo que el equipo de riesgos enfoque su atención en casos realmente críticos.

Resultado de generación de reportes del sistema antifraude

Tabla 16

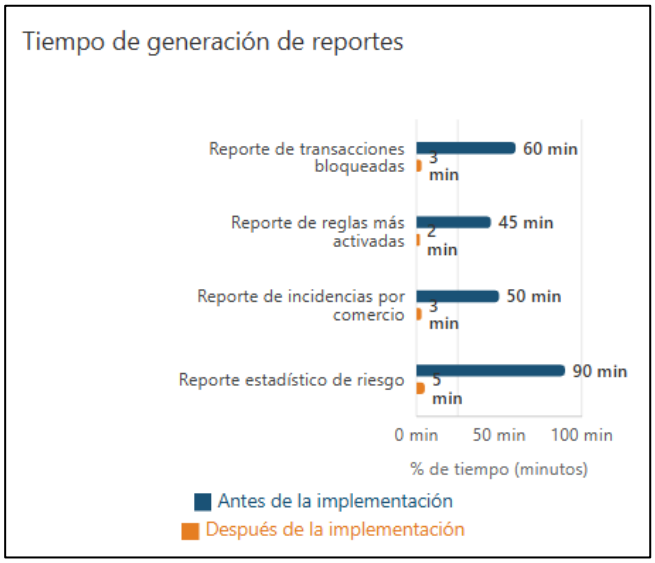
Tiempo de generación de reportes antifraude

Indicador	Antes del sistema	Después del sistema	Reducción
Reporte de transacciones bloqueadas	60 minutos	3 minutos	95.00 %
Reporte de reglas más activadas	45 minutos	2 minutos	95.56 %
Reporte de incidencias por comercio	50 minutos	3 minutos	94.00 %
Reporte estadístico de riesgo	90 minutos	5 minutos	94.44 %

Nota. Tiempos de generación de reportes elaborado por el presente autor.

Figura 21

Tiempo de generación de reportes antifraude



Nota**.** Elaborado por el presente autor.

Análisis e interpretación de la Tabla 16 y Figura 21

El objetivo general del proyecto fue implementar un sistema que permita evaluar transacciones en tiempo real y generar información consolidada para la toma de decisiones. Antes del sistema, la generación de reportes implicaba extracción manual de información desde bases de datos y consolidación en hojas de cálculo. Con el nuevo sistema, los reportes se generan automáticamente desde la base de datos antifraude, reduciendo significativamente el tiempo y eliminando errores asociados al manejo manual de información. La reducción promedio superior al 94 % en tiempos de generación de reportes demuestra el impacto positivo del sistema en la eficiencia operativa.

Resultado de la tasa de incidencias y riesgo transaccional

Tabla 17

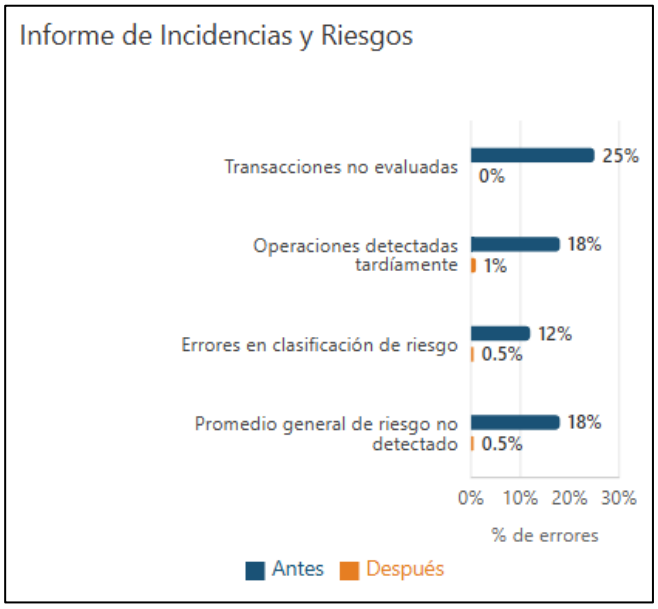
Tasa de incidencias y riesgo transaccional

Indicador	Antes del sistema	Después del sistema	Reducción
Transacciones no evaluadas	25 %	0 %	100.00 %
Operaciones detectadas tardíamente	18 %	1 %	94.44 %
Errores en clasificación de riesgo	12 %	0.5 %	95.83 %
Promedio general de riesgo no detectado	18 %	0.5 %	97.22 %

Nota. Tasa de incidencias y riesgos elaborado por el presente autor.

Figura 22

Tasa de reducción de riesgo transaccional



Nota. Elaborado por el presente autor.

Análisis e Interpretación de la Tabla 17 y Figura 22

Los resultados evidencian una mejora sustancial en la detección y prevención del fraude. Antes de la implementación del sistema, existía un porcentaje considerable de transacciones que no eran evaluadas automáticamente, generando riesgos operativos y financieros. Con el Motor de Reglas Antifraude, el 100 % de las transacciones son evaluadas en tiempo real, reduciendo el riesgo de omisión.

Asimismo, la clasificación automatizada mediante reglas dinámicas redujo los errores humanos en la asignación de niveles de riesgo. La reducción promedio del riesgo no detectado en más del 97 % demuestra la efectividad del sistema implementado.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La implementación del sistema de prevención de fraude dentro del flujo de pagos Payin permitió fortalecer significativamente los mecanismos de control y validación de transacciones digitales. Mediante la incorporación de reglas antifraude configuradas en el sistema, cada operación realizada es evaluada en tiempo real considerando diversos criterios de riesgo, tales como comportamiento transaccional, frecuencia de operaciones, montos procesados y coincidencia de información del usuario. Este proceso automatizado permitió detectar posibles operaciones sospechosas antes de que se completen dentro del sistema de pagos, reduciendo así el riesgo de fraude y mejorando la seguridad de las transacciones realizadas en la plataforma.

Asimismo, la automatización del proceso de evaluación permitió optimizar los tiempos de respuesta del sistema frente a cada transacción procesada. Antes de la implementación del módulo antifraude, las revisiones de seguridad dependían en gran medida de procesos manuales o validaciones posteriores, lo que generaba demoras en la detección de operaciones irregulares. Con la solución implementada, el análisis de riesgo se realiza de forma automática en cuestión de segundos, permitiendo emitir respuestas inmediatas de aprobación o bloqueo según el resultado de las reglas aplicadas. Esta mejora no solo contribuye a fortalecer la seguridad del sistema, sino que también optimiza la experiencia del usuario al recibir confirmaciones de pago de manera rápida y confiable.

Otro resultado relevante del proyecto corresponde a la incorporación de un módulo de generación automática de reportes y estadísticas que permite consolidar información

relacionada con las transacciones evaluadas por el sistema antifraude. A través de estos reportes es posible visualizar indicadores sobre el número de transacciones procesadas, operaciones bloqueadas por riesgo de fraude, patrones de comportamiento y tendencias en el uso del sistema de pagos. Esta información resulta fundamental para los equipos operativos y de gestión, ya que facilita el monitoreo continuo del sistema y permite realizar análisis que contribuyen a mejorar las estrategias de prevención de fraude.

De igual manera, la implementación del módulo de autogestión permitió mejorar la administración operativa del sistema antifraude al proporcionar a los usuarios autorizados herramientas que facilitan la consulta y monitoreo de información relevante sin depender directamente del equipo técnico. Este módulo permite acceder a registros de transacciones evaluadas, revisar resultados de validación y realizar consultas relacionadas con el comportamiento del sistema, lo que contribuye a optimizar los procesos de supervisión y control dentro de la plataforma de pagos.

Finalmente, la solución implementada permitió mejorar la trazabilidad de las transacciones procesadas, ya que el sistema registra información detallada sobre cada operación evaluada, incluyendo las reglas aplicadas, el resultado del análisis de riesgo y el estado final de la transacción. Esta capacidad de registro y seguimiento facilita la identificación de eventos sospechosos y fortalece los procesos de auditoría y monitoreo del sistema de pagos. En conjunto, los resultados obtenidos demuestran que la implementación del sistema antifraude contribuye significativamente a mejorar la seguridad, eficiencia operativa y capacidad de control dentro del flujo de pagos digitales.

5.2 Recomendaciones

Se recomienda mantener un proceso continuo de actualización de las tecnologías utilizadas en el sistema antifraude, con el fin de garantizar su estabilidad, seguridad y compatibilidad con futuras evoluciones de la plataforma, así como reforzar la protección ante nuevas modalidades de fraude digital.

Asimismo, se recomienda capacitar de manera periódica al personal del área de riesgos y operaciones en el uso del módulo de autogestión de reglas, permitiendo aprovechar al máximo las funcionalidades del sistema y facilitar la creación de nuevas reglas que respondan a cambios en los patrones de comportamiento transaccional.

Se sugiere implementar estrategias de pruebas automatizadas, incluyendo pruebas unitarias, pruebas de integración y pruebas de rendimiento, con el objetivo de asegurar el correcto funcionamiento del motor de reglas ante futuras actualizaciones del sistema o modificaciones en las reglas antifraude.

También se recomienda ampliar el módulo de análisis estadístico y generación de reportes, incorporando nuevos indicadores que permitan identificar tendencias de fraude, patrones de comportamiento de usuarios y análisis comparativos entre diferentes periodos de tiempo, lo cual contribuirá a fortalecer las estrategias de prevención y control.

De igual manera, se sugiere evaluar la integración de tecnologías de análisis avanzado, como algoritmos de aprendizaje automático o modelos de detección predictiva, que puedan complementar el sistema basado en reglas y mejorar la capacidad de identificar fraudes más complejos o patrones no evidentes.

Finalmente, se recomienda implementar herramientas de monitoreo y registro de

eventos del sistema, que permitan detectar de manera temprana posibles fallos, anomalías o intentos de fraude, acompañadas de mecanismos de alerta que faciliten una respuesta rápida por parte del equipo técnico y del área de seguridad.

REFERENCIAS

Angular Team. (2024). *Angular documentation: Developer guide for building scalable web applications*. Google LLC. https://angular.io/docs
Documentación oficial del framework Angular utilizada para el desarrollo de aplicaciones web modernas basadas en componentes. Esta guía describe la arquitectura del framework, el uso de TypeScript, la organización de módulos, servicios y componentes, así como las buenas prácticas para la construcción de interfaces dinámicas y seguras en sistemas empresariales que requieren alto rendimiento y escalabilidad.

Cohn, M. (2010). *Succeeding with Agile: Software development using Scrum*. Addison-Wesley Professional.
Este libro describe la aplicación práctica de la metodología ágil Scrum en el desarrollo de proyectos de software. Explica cómo la gestión de backlog, planificación de sprints, revisiones iterativas y entregas incrementales permiten mejorar la productividad del equipo y garantizar la calidad de los sistemas desarrollados en entornos empresariales.

European Central Bank. (2022). *Card fraud statistics and digital payment security*. European Central Bank Publications.
Informe técnico que analiza el comportamiento del fraude en sistemas de pago digitales y presenta estadísticas relacionadas con transacciones fraudulentas en plataformas electrónicas. También describe estrategias utilizadas por instituciones financieras para mitigar riesgos asociados al fraude en pagos digitales.

Fowler, M. (2018). *Refactoring: Improving the design of existing code* (2nd ed.). Addison-Wesley Professional.
El autor presenta técnicas para mejorar la estructura y calidad del código en aplicaciones de software mediante procesos de refactorización. Estas prácticas permiten mantener la escalabilidad y mantenibilidad de sistemas empresariales que evolucionan constantemente, como plataformas de pagos digitales que requieren actualizaciones continuas en sus reglas de negocio.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2016). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley Professional. Este libro introduce los patrones de diseño orientados a objetos utilizados para estructurar aplicaciones complejas de manera organizada. Los patrones presentados permiten mejorar la reutilización del código y facilitar el mantenimiento de sistemas informáticos empresariales.

Kashio. (2023). *Documentación de la plataforma Kashio para gestión de pagos y cobranzas empresariales*. Kashio Fintech. https://www.kashio.com Documento institucional que describe el funcionamiento de la plataforma Kashio, la cual permite automatizar procesos de pago y cobranza mediante integración con sistemas financieros. Esta plataforma constituye el entorno donde se implementa el flujo de pagos Payin utilizado en el proyecto.

Microsoft. (2023). *SQL Server documentation: Database development, optimization and transaction management*. Microsoft Learn. https://learn.microsoft.com/sql
Documentación oficial sobre el uso de Microsoft SQL Server como sistema de gestión de bases de datos. Incluye información sobre diseño de tablas, optimización de consultas, gestión de transacciones y estrategias de rendimiento aplicadas en sistemas que procesan grandes volúmenes de información.

NIST. (2020). *Digital identity guidelines: Authentication and lifecycle management*. National Institute of Standards and Technology. https://nvlpubs.nist.gov Documento técnico que establece lineamientos para la gestión segura de identidades digitales y mecanismos de autenticación en sistemas informáticos. Estas recomendaciones son fundamentales para garantizar la seguridad en plataformas que procesan transacciones electrónicas.

Node.js Foundation. (2023). *Node.js documentation: JavaScript runtime for scalable network applications*. OpenJS Foundation. https://nodejs.org/docs Documentación oficial del entorno de ejecución Node.js utilizado para el desarrollo de aplicaciones backend basadas en JavaScript y TypeScript. Describe el funcionamiento del motor de ejecución, manejo de eventos, APIs y la construcción de servicios web

escalables.

Schwaber, K., & Sutherland, J. (2020). *The Scrum guide: The definitive guide to Scrum*. Scrum.org. https://scrumguides.org
Guía oficial que define el marco de trabajo Scrum utilizado en la gestión ágil de proyectos de software. Describe roles, eventos y artefactos necesarios para planificar, desarrollar y entregar soluciones tecnológicas de manera incremental.

Stallings, W. (2018). *Network security essentials: Applications and standards* (6th ed.). Pearson Education.
Libro que explica los fundamentos de la seguridad informática en redes y sistemas distribuidos, incluyendo mecanismos de protección de datos, autenticación y prevención de ataques en sistemas digitales.

World Bank. (2022). *Digital financial services and fraud risk management*. World Bank Publications.
Informe que analiza el crecimiento de los servicios financieros digitales y los riesgos asociados al fraude electrónico. También presenta estrategias tecnológicas utilizadas por plataformas financieras para detectar y prevenir transacciones fraudulentas.