

UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC
FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS



Tesis

Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024

Presentado por:

Mario Choquetaipe Asto

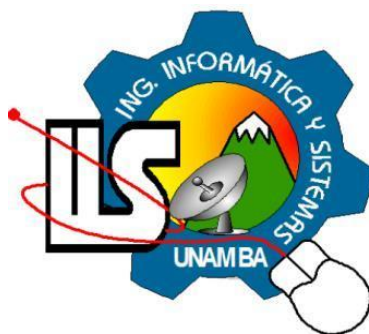
Para optar el título de Ingeniero Informático y Sistemas

Abancay, Perú

2026



UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC
FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS



TESIS

Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024

Presentado por **Mario Choquetaipe Asto**, para optar el título de Ingeniero Informático y Sistemas

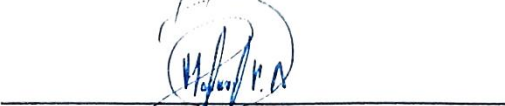
Sustentado y aprobado el 17 de diciembre del 2025 ante el jurado evaluador:

Presidente:



Dr. Ezech Ordoñez Ramos

Primer miembro:



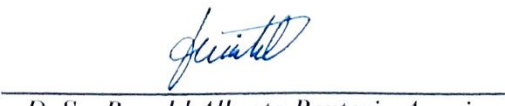
Mtro. Makleny Peralta Ascue

Segundo miembro:



Mag. Kevin Arnold Arias Figueroa

Asesor:



D. Sc. Ronald Alberto Renteria Ayquipa

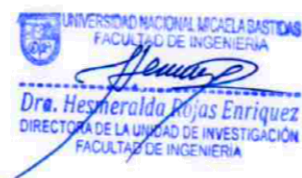
"Año de la recuperación y consolidación de la economía peruana"

CONSTANCIA DE ORIGINALIDAD N° 302-2025

La Universidad Nacional Micaela Bastidas de Apurímac, a través de la Unidad de Investigación de la Facultad de Ingeniería declara que, la Tesis intitulada: **Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de 10 proceso de venta en empresas usuarias de "Rumisof-pos", 2024**, presentada por el Bach. **Mario Choquetaipe Asto**, para optar el Título de **Ingeniero Informática y Sistemas**; ha sido sometido a un mecanismo de evaluación y verificación de similitud, a través del Software Turnitin, siendo el índice de similitud ACEPTABLE de **(12%)** por lo que, cumple con los criterios de originalidad establecidos por la Universidad.

Abancay, 12 de diciembre del 2025

Atentamente,



UNIVERSIDAD NACIONAL MICAELA BASTIDAS
FACULTAD DE INGENIERÍA
Dra. Hesméralda Rojas Enriquez
DIRECTORA DE LA UNIDAD DE INVESTIGACIÓN
FACULTAD DE INGENIERÍA

C. c.
Archivo
REG. N° 1015

Agradecimiento

A la Universidad Nacional Micaela Bastidas de Apurímac, por ofrecerme las herramientas y conocimientos que hoy me hacen un profesional. A todos mis docentes, por su paciencia y dedicación.

*Un agradecimiento enorme a la vida, por enseñarme que con esfuerzo y perseverancia se puede lograr todo. Esta tesis es el resultado de años de sacrificio, de noches sin dormir y de días en los que tenía que trabajar y estudiar al mismo tiempo. Ha sido complicado, pero hoy puedo decir con orgullo: ¡**LO LOGRÉ!***

A todos los que creyeron en mí, a los que no lo hicieron (que también me dieron fuerzas para seguir), y especialmente a mí mismo, por nunca rendirme.

¡Gracias, vida! ¡Este triunfo también es tuyo!

Mario Choquetaipe Asto



Dedicatoria

A mis padres, Asencia Asto Rojas y Pascual Choquetaipe Huamani, quienes desde el cielo me guían con su amor y bendición. A mi pareja, Sofía Yucra Rea, y a mi pequeña bebé, Yarely Evalyz, que son mi mayor motivación para seguir adelante cada día. A mis hermanas, Gregoria, Gumercinda, Rodecindo, Eulalia, Eva, Juana, Gregoria y Milagros, por ser ese apoyo incondicional en los momentos difíciles.

Quiero dedicar este logro sobre todo a Gumercinda, quien fue como una madre para mí, dándome su apoyo, buenos consejos y esa fortaleza que necesitaba en cada paso de este camino. Sin ella, este sueño no hubiera sido posible.



Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024

Línea de investigación: Ingeniería Informática, Industria y Sociedad

Esta publicación está bajo una Licencia Creative Commons



ÍNDICE

	Pág.
INTRODUCCIÓN	1
RESUMEN	2
ABSTRACT	3
CAPÍTULO I	4
PLANTEAMIENTO DEL PROBLEMA	4
1.1 Descripción del problema	4
1.2 Enunciado del problema	6
1.2.1 Problema general	6
1.2.2 Problemas específicos	6
1.3 Justificación de la investigación	6
1.4 Ubicación y contextualización	7
CAPÍTULO II	8
OBJETIVOS E HIPÓTESIS	8
2.1 Objetivos de la investigación	8
2.1.1 Objetivo general	8
2.1.2 Objetivos específicos	8
2.2 Hipótesis de la investigación	8
2.2.1 Hipótesis general	8
2.2.2 Hipótesis específicas	8
2.3 Operacionalización de variables	9
CAPÍTULO III	10
MARCO TEÓRICO REFERENCIAL	10
3.1 Antecedentes	10
3.1.1 A nivel internacional	10
3.1.2 A nivel nacional	13
3.2 Marco teórico	17
3.2.1 Sistema de información de servicios compartidos multiplataforma	17
3.2.2 Proceso de venta	23

3.3	Marco conceptual	26
CAPÍTULO IV		34
METODOLOGÍA		34
4.1	Tipo y nivel de investigación	34
4.1.1	Tipo de investigación	34
4.1.2	Nivel de investigación	34
4.2	Diseño de investigación	34
4.3	Población y muestra	35
4.3.1	Población	35
4.3.2	Muestra	35
4.4	Procedimiento	35
4.4.1	Fase de definición y análisis	35
4.4.2	Fase de diseño conceptual y técnico	36
4.4.3	Fase de desarrollo y construcción iterativa	37
4.4.4	Fase de pruebas, optimización y despliegue	38
4.4.5	Fase de evaluación y consolidación final	39
4.5	Técnicas e instrumentos	39
4.6	Análisis estadístico	41
CAPÍTULO V		42
RESULTADOS Y DISCUSIÓN		42
5.1	Análisis de resultados	42
5.1.1	Descripción del sistema	48
5.2	Contrastación de hipótesis	108
5.3	Discusión	113
CAPÍTULO VI		115
CONCLUSIONES Y RECOMENDACIONES		115
6.1	Conclusiones	115
6.2	Recomendaciones	116
REFERENCIAS BIBLIOGRÁFICAS		118
ANEXOS		128



ÍNDICE DE TABLAS

	Pág.
Tabla 1 — Ubicación de empresas usuarias de Rumisof-pos	7
Tabla 2 — Operacionalización de variables	9
Tabla 3 — Ficha de recolección de datos de la dimensión seguridad	40
Tabla 4 — Análisis de confiabilidad	40
Tabla 5 — Ficha de recolección de datos de la posprueba	41
Tabla 6 — Tabla de frecuencias de la dimensión usabilidad	42
Tabla 7 — Tabla resumen de la ejecución de las pruebas de seguridad	44
Tabla 8 — Requisitos funcionales	48
Tabla 9 — Requisitos no funcionales	49
Tabla 10 — Casos de uso principales	50
Tabla 11 — Historias de usuario	50
Tabla 12 — Especificaciones técnicas de la BD	67
Tabla 13 — Resumen tablas principales de la BD	68
Tabla 14 — Fases de implementación de la facturación electrónica	70
Tabla 15 — Justificación de la metodología	103
Tabla 16 — Roles y responsabilidades	105
Tabla 17 — Lenguajes de programación	106
Tabla 18 — Frameworks de desarrollo	106
Tabla 19 — Herramientas de BD	107
Tabla 20 — Herramientas de infraestructura y despliegue	107
Tabla 21 — Herramientas de productividad y desarrollo	107
Tabla 22 — Pruebas de normalidad	108
Tabla 23 — Prueba de rangos con signo de Wilcoxon HE1	109
Tabla 24 — Prueba de rangos con signo de Wilcoxon HE2	110
Tabla 25 — Prueba de rangos con signo de Wilcoxon HG	112
Tabla 26 — Matriz de consistencia	129
Tabla 27 — Matriz de datos de la dimensión usabilidad	136
Tabla 28 — Matriz de datos de la dimensión seguridad	137



ÍNDICE DE FIGURAS

	Pág.
Figura 1 — Problemática general del proceso de venta	6
Figura 2 — Pasos generales del proceso de venta	24
Figura 3 — Categorías de la dimensión usabilidad	43
Figura 4 — Ejecución de pruebas de seguridad bajo OWASP	43
Figura 5 — Evolución de las ejecuciones	45
Figura 6 — Tiempos de proceso de órdenes antes y después	46
Figura 7 — Tiempos de proceso de facturación antes y después	47
Figura 8 — Tiempos de proceso de venta antes y después	47
Figura 9 — Diagrama de secuencia del proceso de venta en módulo restaurante	54
Figura 10 — Diagrama de secuencia del proceso de venta en módulo minimarket	56
Figura 11 — Diagrama de secuencia del proceso de <i>check-in</i> en módulo hotel	58
Figura 12 — Diagrama de secuencia del proceso de facturación electrónica	59
Figura 13 — Diagrama de flujo y componentes del sistema	62
Figura 14 — Diagrama de arquitectura general del sistema	65
Figura 15 — Arquitectura interna del módulo Core	66
Figura 16 — Arquitectura interna del módulo de facturación	66
Figura 17 — Arquitectura interna del módulo de orquestación	67
Figura 18 — Modelo E-R de la base de datos del sistema	69
Figura 19 — XML firmado y constancia de recepción (CDR) de boleta electrónica	75
Figura 20 — Proceso de facturación	78
Figura 21 — Módulo de configuración del sistema según tipo de negocio	78
Figura 22 — Interfaz de login del sistema según perfil de usuario	79
Figura 23 — Interfaz de ingreso de contraseña	79
Figura 24 — Interfaz del módulo en modo hotel	80
Figura 25 — Interfaz de administración del modo restaurant	80
Figura 26 — Interfaz del modo hotel	81
Figura 27 — Interfaz del modo minimarket	81
Figura 28 — Tablas que conforman específicamente la facturación electrónica	82

Figura 29 — Diagrama del patrón BLoC	99
Figura 30 — Ejecución del sistema multiplataforma	102
Figura 31 — Tablero digital de flujo de trabajo	104
Figura 32 — Tablero digital de flujo de trabajo con tareas	104
Figura 33 — Instrumento para medir la usabilidad	130
Figura 34 — Instrumento para la prepueba de la variable dependiente	131

INTRODUCCIÓN

En la actualidad, muchas empresas enfrentan dificultades en la eficiencia de sus procesos de venta, especialmente en la gestión de órdenes, generación de facturas y seguimiento de ventas. Estas limitaciones generan demoras, errores y pérdida de oportunidades comerciales, afectando directamente la productividad y competitividad de las organizaciones.

Esta problemática motivó el desarrollo de una solución tecnológica que permita optimizar dichos procesos. El tema de esta investigación aborda el desarrollo de un sistema de información de servicios compartidos multiplataforma denominado “Rumisof-pos” orientado a mejorar el tiempo de proceso de venta en empresas.

La importancia de esta investigación radica en la necesidad de contar con soluciones tecnológicas que permitan superar dichas limitaciones. Los sistemas de información desempeñan un papel crucial en la mejora de la productividad empresarial, la generación de documentos, reportes y la automatización de procesos operativos. En sectores como servicios y retail (comercio minorista), las soluciones tecnológicas se han convertido en herramientas fundamentales para optimizar la ejecución de modelos de negocio.

La investigación fue de tipo aplicada, con nivel explicativo y diseño preexperimental. La población estuvo conformada por 47 empresas usuarias del sistema Rumisof-pos, entre estas tenemos empresas de servicios y de retail; y se trabajó con la totalidad de dicha población como muestra.

La estructura del informe final se organizó de la siguiente manera: el capítulo I presenta la descripción y enunciado del problema, así como la justificación. El capítulo II expone los objetivos, hipótesis y operacionalización. El capítulo III desarrolla el marco teórico. El capítulo IV detalla la metodología de investigación. El capítulo V analiza los resultados, verifica las hipótesis y discute los hallazgos. Finalmente, el capítulo VI presenta las conclusiones y recomendaciones, seguido de las referencias bibliográficas y anexos correspondientes.

RESUMEN

La presente tesis tuvo por objetivo la implementación de un sistema de información de servicios compartidos multiplataforma para la mejora del proceso de venta, este sistema implementado se denominó Rumisof-pos. La metodología empleada fue de tipo aplicada, nivel explicativo, la muestra de estudio consistió en 47 empresas y se utilizó el estadístico no paramétrico de Wilcoxon para determinar la diferencia de medias en el análisis estadístico inferencial. Los resultados evidencian que el 100% de empresas evaluó al sistema con usabilidad aceptable, sobre la variable proceso de venta, se evidenció disminución en los tiempos después de la implementación del sistema, esta reducción fue de 67.5% en los tiempos de proceso de órdenes, reducción del 86.1% en los tiempos de proceso de facturación y 76.8% de reducción en los tiempos de proceso de venta. Las pruebas de hipótesis fueron con estadística no paramétrica, la cual evidenció con un p-valor igual a 0 en todos los casos, que los tiempos de proceso de órdenes, de generación de facturas y de proceso de ventas antes de la implementación del sistema son mayores a los tiempos después de la implementación del sistema. Se concluyó que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de órdenes, el tiempo de generación de facturas y por lo tanto el tiempo de proceso de venta en empresas usuarias del sistema.

Palabras clave: *proceso de venta, usabilidad, seguridad, proceso de órdenes, proceso de facturación.*



ABSTRACT

The objective of this thesis was the implementation of a multiplatform shared services information system to improve the sales process. This implemented system was named Rumisofpos. The methodology employed was applied and explanatory. The study sample consisted of 47 companies, and the Wilcoxon non-parametric test was used to determine the difference in means for the inferential statistical analysis. The results show that 100% of the companies evaluated the system as having acceptable usability. Regarding the sales process variable, a decrease in processing times was observed after the system's implementation: a 67.5% reduction in order processing times, an 86.1% reduction in billing processing times, and a 76.8% reduction in sales processing times. Hypothesis testing was conducted using non-parametric statistics, which showed a p-value of 0 in all cases, indicating that order processing, invoice generation, and sales processing times before system implementation were longer than those after implementation. It was concluded that developing and implementing a multiplatform shared services information system improves order processing time, invoice generation time, and therefore sales processing time for companies using the system.

Keywords: *sales process, usability, security, order processing, invoicing process.*



CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del problema

La eficiencia en el proceso de venta es un factor crítico para la competitividad, especialmente en sectores de alta rotación como el expendio de alimentos y bebidas. Este proceso se divide en dos componentes clave: la gestión de órdenes y la facturación. Las demoras en cualquiera de estas etapas afectan directamente la experiencia del cliente, la productividad operativa y la rentabilidad del negocio.

A nivel global, diversos estudios y datos de organizaciones reconocidas demuestran que los tiempos prolongados en los procesos de venta afectan negativamente a las empresas, tanto en su desempeño financiero como en la satisfacción y lealtad de los clientes. Así también, se destaca la importancia de la tecnología en la optimización de estos procesos.

Retrasos en los procesos de venta, especialmente en entornos digitales, disminuyen significativamente las tasas de conversión y las ventas. Los clientes son especialmente sensibles a las demoras en etapas críticas como el checkout, lo que puede llevar a la pérdida de ventas potenciales (Gallino, Karacaoglu y Moreno 2023; Niemi et al. 2020; Akturk, Mallipeddi y Jia 2022). Del mismo modo, los tiempos de espera elevados generan insatisfacción, reducen la calificación de la experiencia del cliente y afectan la lealtad (Akturk, Mallipeddi y Jia 2022; Dr. Thiruchelvi et al. 2025). En cuanto al impacto financiero, se ha cuantificado que retrasos de hasta dos semanas en la entrega o en el proceso pueden reducir las ventas hasta en un 10%, y retrasos mayores pueden causar pérdidas de hasta el 40% del valor de ventas, afectando directamente la rentabilidad (Niemi et al. 2020).

Estudios empíricos concluyen que la estandarización y automatización de procesos de venta, con la adopción de tecnologías (CRM, IA, automatización), reducen significativamente los tiempos y mejoran la eficiencia y satisfacción del cliente (Success Ugbaja et al. 2024; Rodriguez y Boyer 2020; Parodi, Montañez y Chavez Ugaz 2024). Un estudio realizado por Singh et. al (2023), analizó el impacto de la integración de sistemas omnichannel y la digitalización de procesos en el retail, estos logran una reducción promedio del

25% en el tiempo de procesamiento de pedidos y una mejora significativa en la precisión de la facturación, lo que se traduce en una mayor satisfacción del cliente y eficiencia operativa.

En Latinoamérica, la digitalización presenta desafíos y oportunidades claras en las Micro, Pequeñas y Medianas Empresas (MIPYMES). Un informe de la Comisión Económica para América Latina y el Caribe (CEPAL) de 2022 sobre la digitalización empresarial post-pandemia indica que, si bien la adopción de tecnologías de facturación electrónica ha avanzado, todavía existe una brecha significativa en la integración de sistemas de front-office (toma de pedidos) con los de back-office (facturación y stock). El informe destaca que la falta de integración es un factor clave detrás del 20% de las ineficiencias operativas reportadas por las MIPYMES encuestadas en el sector servicios, afectando la velocidad y la trazabilidad de la transacción completa (CEPAL 2022).

A nivel nacional, el estudio de Koc Collazos y Pizán Salcedo (2024), publicado por la Universidad Peruana de Ciencias Aplicadas, identificó que las demoras en la atención de pedidos en empresas manufactureras peruanas se deben a la centralización de decisiones y sobrecarga de funciones. Estas demoras afectan la eficiencia operativa y la fidelización del cliente, y se proponen soluciones basadas en gestión por procesos. Si bien el estudio se centra en manufactura, su principio de la centralización como causa de la lentitud es aplicable a la gestión de caja y facturación en servicios.

El presente estudio se centra en empresas usuarias del sistema "Rumisof-pos" ubicadas en diversas regiones del Perú. Se ha identificado que, durante los horarios de mayor afluencia, el tiempo promedio de atención por pedido supera los 5 minutos, y el tiempo de emisión de factura puede extenderse hasta 5 minutos adicionales, generando acumulación de clientes y pérdida de ventas potenciales. Esta evidencia empírica ha sido recopilada mediante observación directa en establecimientos afiliados a "Rumisof-pos", entrevistas con encargados de caja y análisis de reportes de ventas.

En la Figura 1 se sintetiza la problemática identificada en los distintos tipos de negocio (restaurante, minimarket y hotel), mostrando cómo el problema general impacta de manera específica en el cliente/huésped, el cajero/administrador y el mozo/recepcionista/vendedor.

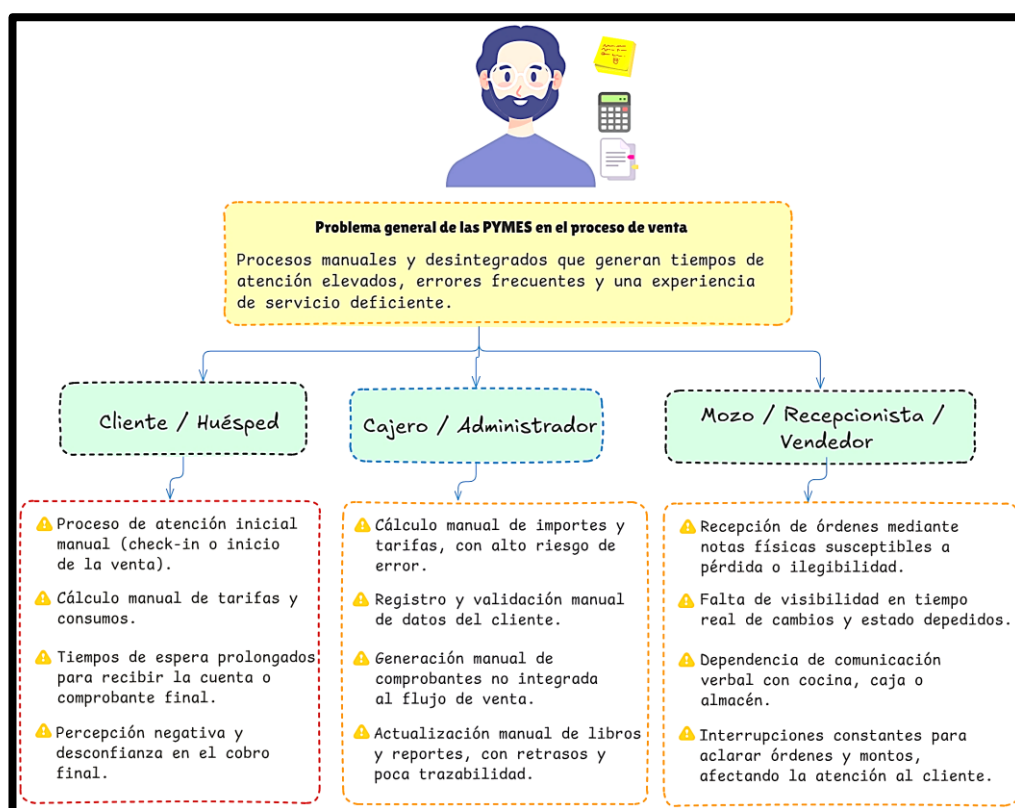


Figura 1 — Problemática general del proceso de venta

1.2 Enunciado del problema

1.2.1 Problema general

¿En qué medida el sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024?

1.2.2 Problemas específicos

- ¿En qué medida el sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024?
- ¿En qué medida el sistema de información de servicios compartidos multiplataforma mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024?

1.3 Justificación de la investigación

Esta investigación cobra relevancia por su contribución al fortalecimiento de la competitividad y sostenibilidad de las MIPYMES peruanas mediante la optimización del proceso de venta, específicamente en las etapas de gestión de órdenes y facturación. Estas actividades, al ser parte esencial de la interacción con el cliente, representan una oportunidad

estratégica para mejorar la experiencia de compra, consolidar la fidelización y elevar los estándares de atención en el sector.

La investigación propuesta busca aportar soluciones concretas y escalables para optimizar estos procesos mediante el análisis de variables clave como el tiempo de toma de pedidos, tiempo de emisión de facturas y cumplimiento de requisitos SUNAT. Los resultados permitirán fortalecer la competitividad de las MIPYMES, mejorar la experiencia del cliente y contribuir a la formalización digital del comercio en el Perú.

Los beneficios incluyen el fortalecimiento de la imagen institucional de las empresas, el incremento en la satisfacción del cliente, la reducción de errores administrativos y la mejora en la gestión tributaria. Además, al enfocarse en empresas usuarias del sistema Rumisof-pos, la investigación contribuirá al desarrollo de herramientas adaptadas al contexto nacional, con la posibilidad de ser aplicadas a otros sectores.

Entonces, este estudio se justifica por su aporte al desarrollo económico, la innovación operativa y la mejora continua en el ecosistema empresarial peruano, alineándose con los objetivos de formalización, digitalización y calidad en los servicios comerciales.

1.4 Ubicación y contextualización

La ejecución del proyecto de investigación se realizó en las empresas usuarias que utilizan el sistema de servicios compartidos multiplataforma denominado “Rumisof-pos”, donde se realizó la observación y aplicación del sistema.

Estas empresas están distribuidas geográficamente en diferentes regiones del país, la Tabla 1 muestra esta situación.

Tabla 1 — Ubicación de empresas usuarias de Rumisof-pos

Departamento	Frecuencia	%
Loreto	17	36.17%
Apurímac	6	12.77%
Lima	6	12.77%
Ucayali	4	8.51%
Arequipa	3	6.38%
Tacna	3	6.38%
Callao	2	4.26%
Cajamarca	2	4.26%
Ica	1	2.13%
Huancavelica	1	2.13%
Amazonas	1	2.13%
Cusco	1	2.13%

CAPÍTULO II

OBJETIVOS E HIPÓTESIS

2.1 Objetivos de la investigación

2.1.1 Objetivo general

Desarrollar e implementar un sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.

2.1.2 Objetivos específicos

- Desarrollar e implementar un sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024.
- Desarrollar e implementar un sistema de información de servicios compartido multiplataforma para mejorar el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.

2.2 Hipótesis de la investigación

2.2.1 Hipótesis general

Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.

2.2.2 Hipótesis específicas

- Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024.



- Al desarrollar e implementar un sistema de información de servicios compartido multiplataforma mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.

2.3 Operacionalización de variables

Definición de variables:

- **Variable independiente:** Sistema de información de servicios compartidos multiplataforma
- **Variable dependiente:** Proceso de venta

Tabla 2 — Operacionalización de variables

Variable	Dimensión	Indicadores	Índices/Escala
Sistema de información de servicios compartidos multiplataforma (Laudon y Laudon 2012; Tech-Target 2024; Triangle 2023)	Usabilidad (Chen, Sockel y Falk 2009)	– Eficacia – Eficiencia – Satisfacción (Brooke 1995; UXables 2024; WeAreTesters 2025)	Totalmente en desacuerdo En desacuerdo Neutro De acuerdo Totalmente de acuerdo
	Seguridad (Alexei y Alexei 2023; Kotrikadze 2023)	– Autenticación – Inyección – Validación (Ezenwoye 2024)	Fail Pass
Proceso de venta (Ongallo Chaclón 2012)	Proceso de órdenes (Bertone 2020)	Tiempo de proceso órdenes (CargoSapiens 2025)	Minutos
	Proceso de facturación (González Jiménez y Novoa Vargas 2024)	Tiempo de proceso facturación (Duemint 2024)	Minutos

CAPÍTULO III

MARCO TEÓRICO REFERENCIAL

3.1 Antecedentes

3.1.1 A nivel internacional

- a) El artículo de Romero, La Paz y García (2024) titulado “Impacto de las herramientas tecnológicas en las ventas de empresas chilenas”, que tiene por objetivo evaluar el impacto de las herramientas tecnológicas en las ventas de las empresas chilenas, para lo cual analizaron la encuesta Acceso y Uso de Tecnología de Información y Comunicación en Empresas. Las variables que afectan las ventas las determinaron realizando un análisis descriptivo y regresión lineal, encontrando que aumentos en variables de especialistas TIC, software ERP, redes sociales, banda ancha, sitios web, dispositivos electrónicos y Big Data generan aumentos en las ventas de las empresas chilenas, siendo los sistemas ERP y las TIC los que tienen los coeficientes de regresión más altos en el modelo. Por lo que recomiendan que las empresas adopten herramientas digitales, especialmente sistemas ERP y contratación de profesionales en TIC.
- b) Otro estudio es el de Acosta, Garcia y Pacheco (2024), titulado “FAST a Digital Sales Transformation Software for the Real Estate Industry” cuyo objetivo fue implementar un sistema web para mejorar la gestión y eficiencia del proceso de ventas en una empresa inmobiliaria. La metodología usada para el desarrollo e implementación del sistema web fue la metodología Extreme Programming (XP), con fases de planificación, diseño, codificación y pruebas; los autores midieron los tiempos de búsqueda y registro antes y después. Los resultados más importantes fueron la reducción del 64% en el tiempo de búsqueda de información y 62% en el registro de ventas; aumento del 72% en el desempeño de tareas diarias. Concluyeron que el sistema de información optimizó significativamente el proceso de ventas y la administración de clientes. Y las recomendaciones propuestas fueron adoptar sistemas similares en otras empresas y explorar otras metodologías de desarrollo para validar resultados.

- c) El estudio de Poljak, Dević y Glavaš (2024), denominado “Enhancing Sales Efficiency with the Use of Business Intelligence and Analytics in a Public Company”, tuvo por finalidad analizar si la aplicación de sistemas de inteligencia de negocios (BI) en la red de ventas incrementa la eficiencia de los trabajadores, para lo cual aplicaron como metodología el análisis de datos de ventas y eficiencia de trabajadores durante cinco años, antes y después de implementar el sistema BI. Los resultados fueron un aumento significativo en la realización de ventas por trabajador y por cliente tras la implementación del sistema. Las conclusiones fueron que los sistemas de BI mejoran la eficiencia de ventas, aunque se recomienda controlar factores externos en futuros estudios. Las recomendaciones realizadas fueron realizar estudios experimentales y considerar factores como motivación, calidad del producto y condiciones económicas.
- d) Smith (2024) realizó un estudio titulado “The Impact of Technology on Sales Performance in B2B Companies”, el cual tuvo como objetivo explorar el impacto de la tecnología (IA, big data, automatización) en el desempeño de ventas B2B. Aplicó una metodología de revisión de casos de empresas que implementaron tecnologías avanzadas en ventas. Sus resultados mostraron una mejora en la eficiencia, personalización y productividad de ventas; mejor toma de decisiones basada en datos. Concluyó que la tecnología transforma los procesos de venta y mejora el rendimiento, aunque existen barreras de adopción. Las recomendaciones propuestas fueron superar la resistencia al cambio y facilitar la integración tecnológica.
- e) Parvin y Ramyar (2024) en su estudio denominado “Role of Sales and Marketing Information Systems in Companies Sales Rate Increase” plantearon como objetivo investigar el papel de los sistemas de información de ventas y marketing en el incremento de las tasas de venta empresariales. El tipo de estudio fue empírico cuantitativo, de análisis correlacional y predictivo, aplicaron encuestas y análisis estadístico multivariado en empresas de diversos sectores industriales. Los resultados arrojaron una correlación positiva significativa en los sistemas de información y la tasa de ventas, en promedio un 28% de mejora en eficiencia de procesos de ventas, reducción del 45% en tiempo de respuesta a clientes e incremento del 22% en tasa de conversión de leads. Concluyeron que los sistemas de información de ventas son catalizadores críticos del crecimiento y que la integración de datos mejora significativamente la toma de decisiones.



- f) Abaddi (2025) en su estudio “Analysis and improvement of sales management through CRM and simulation utilization”, que tuvo por objetivo analizar y mejorar la gestión de ventas mediante la utilización de sistemas CRM y técnicas de simulación para optimizar procesos comerciales. Realizaron una investigación aplicada con simulación, en este estudio cuantitativo aplicaron técnicas de implementación de CRM, simulación de procesos y análisis estadístico. Los resultados más importantes fueron: mejora del 42% en eficiencia de gestión de leads, reducción del 35% en tiempo de ciclo de ventas, incremento del 28% en tasas de conversión y optimización del 30% en asignación de recursos de ventas. Concluyendo que la integración CRM-simulación proporciona insights valiosos para optimización y que la simulación permite identificar cuellos de botella en procesos de venta.
- g) Velica Yoyellia, Kuncoro y Yandra Arkeman (2024) en su artículo “Desain sistem informasi rantai pasok terintegrasi berbasis enterprise resource planning” se propusieron diseñar un sistema de información integrado de cadena de suministro basado en Enterprise Resource Planning (ERP) para optimizar procesos de ventas. Utilizaron un diseño de sistemas con enfoque ERP. Los resultados obtenidos fueron la integración exitosa de módulos de ventas, inventario y finanzas, reducción del 40% en tiempo de procesamiento de órdenes, mejora del 35% en precisión de información de inventario y automatización del 80% de procesos manuales de cadena de suministro. Las conclusiones más relevantes fueron que los sistemas ERP integrados mejoran significativamente la eficiencia operacional y que la integración de cadena de suministro optimiza los procesos de venta.
- h) Fagundes et al. (2024) en su estudio “Alavancagem das vendas por cross-selling e up-selling por meio de soluções digitais” se propusieron analizar cómo las soluciones digitales pueden potenciar las ventas mediante estrategias de cross-selling y up-selling en entornos comerciales. Fue una investigación aplicada con enfoque cuantitativo, que hizo un análisis de implementación de soluciones digitales. Los resultados fueron incremento del 34% en ventas por cross-selling mediante soluciones digitales, mejora del 28% en up-selling utilizando algoritmos de recomendación, reducción del 25% en tiempo de identificación de oportunidades de venta y aumento del 40% en valor promedio de transacción.

3.1.2 A nivel nacional

- a) Agurto Orellana y Flores Meca (2024) realizaron la tesis denominada “Implementación de un sistema CRM para mejorar la gestión de ventas en una empresa importadora en Lima, 2023”, en la cual se propusieron mejorar la gestión de ventas con la implementación de un sistema CRM. El estudio fue pre experimental con pretest y posttest, la muestra fue de 45 clientes de la empresa importadora, a estos se les aplicaron cuestionarios de encuesta. Los resultados principales indican que la implementación del sistema CRM mejora la gestión de ventas en una empresa importadora, pasando de niveles medios a niveles altos.
- b) Arrivasplata Rojas (2024), en su estudio titulado “Implementación de un Sistema Web para mejorar la Gestión de Ventas en la Empresa Sumac Chasca Perú S.A.C” se propuso proporcionar una solución implementando un sistema web para mejorar la gestión de ventas de la empresa. El estudio fue cuantitativo, de diseño no experimental. Los resultados destacan una relación directa entre el sistema y la gestión de ventas con p-valor de 0.0078, y una mejora en el servicio al cliente. El estudio concluye que la implementación del sistema web mejora la gestión de ventas.
- c) Anchiraico Garcia (2022) en su tesis “Implementación de sistema web multiplataforma con inteligencia de negocios para mejorar la gestión de ventas en la farmacia Fiorella - La Victoria, 2021”, tuvo por objeto comprobar que la implementación de un sistema web multiplataforma con inteligencia de negocios mejora el control y gestión de las ventas. La investigación fue del tipo experimental, de nivel explicativo, la muestra fue constituida por los 30 trabajadores de la farmacia a quienes se la aplicarán encuestas. Como resultados importantes se observa con índice de relación de 0.87 que las variables sistema web con inteligencia de negocios se relaciona con la gestión de ventas.
- d) Rojas Lorenzo (2021) en su tesis “Sistema informático bajo plataforma web para la mejora del proceso de ventas de la empresa 4J y M SAC”, tuvo como objetivo implementar un software informático web para la mejora de proceso de ventas. Se desarrolló el sistema usando la programación extrema XP, Ubuntu, PostgreSQL y PHP para la interfaz. Como conclusión menciona que el sistema, ha satisfecho todas las expectativas del usuario, resolviéndose el problema de la facturación, reserva de pedidos, cálculos de impuestos, ranking de ventas, facturación electrónica, guías de remisión para el traslado de productos,



etc. Todas estas herramientas implementadas en el sistema, han vuelto más competitiva a la empresa, incrementándose sus ventas y mejorando su posicionamiento. El sistema ha sido implementado eficiente y eficazmente, gracias a los conocimientos adquiridos de desarrollo de sistemas informáticos.

- e) Melgarejo Chavez y Muñoz Murga (2024) en su tesis “Desarrollo de un sistema web para optimizar el proceso de ventas, área de comercialización de minerales, Ingenia Soft, Trujillo 2024”, tuvo como objetivo implementar un sistema web para optimizar el proceso de ventas. El sistema web fue desarrollado utilizando PHP y el framework Yii2, apoyado por una base de datos gestionada en PostgreSQL, usando la metodología SCRUM como herramienta de desarrollo ágil, el sistema permitió automatizar procesos claves como el registro de ventas, el seguimiento y control del estado de la venta, la emisión de documentos de venta, mejorando significativamente la precisión y rapidez en la operación del área comercial. Entre los principales resultados, se destaca la reducción de errores operacionales y una mayor capacidad para analizar los datos de ventas para tomar decisiones estratégicas, estos avances mejoraron la calidad del proceso de digitalización en la empresa, culminando en la obtención del certificado ISO 9001. La implementación de este sistema marca un avance significativo hacia la transformación digital de la empresa, contribuyendo a su competitividad y sostenibilidad en un contexto empresarial que presenta crecientes demandas.
- f) La investigación de Arredondo (2023) Implementación de un sistema web para el proceso de ventas en la empresa Lconst Contratistas Generales E.I.R.L, el cual tuvo por objetivo evaluar el impacto de la implementación del sistema web para el proceso de ventas de la empresa, en los indicadores cierre de ventas y gastos de ventas. De tipo aplicada, nivel explicativo y diseño preexperimental, realizando un muestreo no probabilístico por conveniencia se tomaron todos los comprobantes de pago documentados desde julio del 2020 a mayo del 2021. Los resultados permiten concluir que no hay evidencia suficiente para demostrar que la implementación del software aumenta la tasa de cierre de ventas, así como concluir que el software reduce la tasa de gastos de ventas. Por lo que la implementación del sistema web mejoró el proceso de venta en la empresa en un 20%.

- g) En el trabajo de López y Ruíz (2021), denominado “Sistema Web basado en la Metodología XP para la Gestión de Pedidos en el Restaurante Náutico de Trujillo”, el objetivo fue mejorar la gestión de pedidos en el restaurant mediante un sistema web. La muestra fue de 30 procesos de gestión de pedidos, el instrumento fue la ficha de observación. Los resultados encontrados fueron el decremento del tiempo de registro de pedidos del cliente en un 35.48%, la reducción del tiempo de búsqueda de información de pedidos del cliente en un 68.58%, también se redujo el tiempo de emisión de reporte de pedidos del cliente en un 62.26% y por último el nivel de satisfacción del cliente se aumentó en un 63.3%.
- h) En la investigación de Serrano (2023) denominada Sistema web para la mejora del proceso de atención al cliente en el restaurant Los Ángeles, El Tambo – Huancayo, 2023, el fin fue implementar un sistema web para el proceso de atención al cliente. De tipo aplicada, nivel explicativo, la población fueron los clientes del restaurant, las técnicas fueron las fichas de observación y la encuesta. Los resultados fueron que el 64.83% de los clientes estuvieron satisfechos por la atención al cliente antes de desarrollar el sistema web, mientras que, luego de implementado el sistema web, el 85.68% de los clientes presentan satisfacción en el proceso de atención al cliente. Como conclusión se obtuvo que la implementación de un sistema web incide significativamente en la mejora de los tiempos de espera en la atención al cliente, donde se obtuvo un valor de T-Student de -5.694 y para p-valor de $0.000 < 0.05$; lo que indicó que se reduce el tiempo de espera en la atención al cliente.
- i) En la investigación de Principe y Pariasca (2018) que lleva como título Desarrollo e implementación de un sistema web para la gestión de ventas de la empresa Factory Solutions, el objetivo fue determinar en qué medida la implementación de un sistema web mejora el proceso de gestión de ventas de la empresa Factory Solutions SAC. Este estudio fue aplicado, explicativo, la muestra fue de 30 flujos de venta. Sobre los resultados permiten concluir que todos los indicadores mejoraron después de la implementación del sistema web, es decir el tiempo empleado en la elaboración del comprobante de venta mejoró en 54%, el tiempo empleado en la generación de estados de cuenta en 52%, el tiempo empleado en la generación de reporte de deudas por cobrar en 52% y el tiempo empleado en el envío de notificación de pago en 63%. Con lo cual se cumple el

objetivo general que era determinar la medida en que mejora el proceso de gestión de ventas de la empresa con el sistema web.

- j) El estudio de Huaman y Huayanca (2017) titulado Desarrollo e implementación de un sistema de información para mejorar los procesos de compras y ventas en la empresa Humaju, tuvo por objetivo desarrollar e implementar un sistema de información con la metodología Proceso Unificado Ágil (AUP) para mejorar los procesos de compras y ventas en dicha empresa. La investigación fue aplicada de nivel explicativo, diseño pre experimental, se usaron las técnicas de observación y encuesta para recoger los datos, la muestra estuvo conformada por 30 procesos de compras y ventas. Los resultados permitieron comprobar que la implementación del sistema de información usando la metodología Proceso Unificado Ágil (AUP) disminuyó: el tiempo para elaborar orden y comprobante de pedido, el tiempo de elaboración de orden de compra de productos faltantes, el tiempo para emitir cotización al cliente, el tiempo en la elaboración de Kardex valorizado y el tiempo para generar pedido a proveedores.
- k) Vargas (2017) en su investigación titulada Sistema web para el proceso de venta en la empresa Calzatec E.I.R.L, tuvo como finalidad implementar un sistema web para mejorar el proceso de venta de la empresa. El tipo de investigación fue aplicada, de nivel experimental, se hizo un pretest y postest de los indicadores pedidos por cliente y porcentaje de fidelidad, obteniendo mejoras en ambos indicadores, el promedio de pedidos por cliente aumentó en 0.09 y el porcentaje de fidelidad aumentó en 9.25%. Con lo que se concluyó que el sistema web mejoró el proceso de venta en la empresa Calzatec.
- l) La investigación de Carrillo (2017), denominada Implementación de un sistema de Información para mejorar la gestión de los procesos de compra, venta y almacén de Productos deportivos en la tienda Casa de Deportes Rojitas E.I.R.Ltda. - Chimbote; 2014, tuvo como objetivo implementar un sistema de información para la mejora de la gestión de procesos de compra, venta y almacén en la mencionada empresa. Su diseño fue no experimental, tipo documental y descriptiva, la muestra fue de 16 trabajadores. Los resultados reflejaron que el 75% refirió que la tienda necesita la implementación del sistema de información, así también el 75% mencionó que es beneficioso contar con el sistema, el 100% indica que existe pérdida de tiempo en la realización de reportes, así como un 100% declaró que los datos en la administración están desactualizados.

- m) Además, está Chugnas (2014) en su tesis Sistema de gestión de ventas para reducir el tiempo de atención al cliente y aumentar el margen de utilidad en la empresa “Corporación Dijol SRL” en el año 2012, que planteó como objetivo diseñar e implementar un sistema de gestión de ventas para reducir el tiempo de atención al cliente y aumentar el margen de utilidad en dicha empresa. El estudio fue explicativo, pre-experimental. Los datos recolectados permitieron concluir que mediante el desarrollo e implementación del sistema de gestión de ventas se redujo el tiempo de atención al cliente en un 30% y aumentó el margen de utilidad en un 10% en la empresa Corporación DIJOL SRL.
- n) Otro estudio es el de Ludeña y Rodríguez (2013) cuyo título es Sistema de información Web-Móvil para mejorar la gestión de ventas de entradas de cine en la ciudad de Trujillo, cuyo objetivo general fue desarrollar un sistema de información web-móvil para la mejora de la gestión de ventas de entradas de cine en Trujillo. La investigación fue de tipo aplicada, explicativa y preexperimental. En base a los resultados se concluyó que el tiempo promedio de compra de entradas de cine es menor con el sistema propuesto, además que el tiempo promedio de realizar reportes en las ventas de entradas de cine es menor con el sistema propuesto, también se comprobó que el nivel de satisfacción del cliente es mayor con el sistema propuesto y el nivel de satisfacción del personal del área de ventas es mayor con el sistema propuesto.

3.2 Marco teórico

3.2.1 Sistema de información de servicios compartidos multiplataforma

Para definir esta variable compuesta, primero se definen las partes que la componen:

a) *Sistema de información*

Un sistema de información se define como un conjunto de servicios que crean un flujo de trabajo de información, donde la gestión de la información y/o el conocimiento es parte de sus funcionalidades (Saleh y Abel 2015). Un sistema de información es una colección de muchos conjuntos de datos que garantizan el cumplimiento exitoso de un objetivo comercial.



Así mismo, Laudon y Laudon (2012) indican que un sistema de información es un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización.

b) *Servicio de software*

En el desarrollo de software, los servicios son unidades de funcionalidad pequeñas y flexibles que pueden ser implementadas de manera autónoma. En este contexto, los "servicios" representan diversas funciones de una aplicación en entornos complejos como los basados en la web y las empresariales (Pradhan y Dash 2020). Estos servicios son conocidos también como los microservicios.

En una arquitectura de microservicios, las funciones de una aplicación se descomponen en numerosos componentes más diminutos, cada uno dedicado a tareas específicas. Estos componentes o servicios son de granularidad fina y generalmente cuentan con pilas tecnológicas, métodos de gestión de datos y bases de datos independientes. Estos pueden interconectarse con otros servicios de la aplicación mediante API REST, sistemas de mensajería y transmisión de datos (Pahl y Jamshidi 2016).

La arquitectura de microservicios se presenta como un enfoque eficaz para la construcción de aplicaciones. Debido a su baja dependencia y distribución, incluso si un servicio presenta algún problema, ello no afectará al conjunto del sistema, a diferencia de enfoques más tradicionales. El acoplamiento débil contribuye a reducir las complejidades y las dependencias en una aplicación. Esto permite que los equipos de desarrollo aceleren la creación de nuevos componentes de la aplicación y satisfagan las crecientes necesidades del negocio.

c) *Servicios compartidos*

Los servicios compartidos son aquellos que son usados por varias áreas de una organización. En el desarrollo de software se refiere al uso de estos servicios por diferentes funciones, rutinas en la arquitectura u otros componentes. Según TechTarget (2024), los servicios compartidos constituyen un modelo de entrega que las organizaciones utilizan para consolidar funciones empresariales similares en una única unidad que apoya a toda la empresa, operando como su propia entidad y de manera similar a un proveedor externo.

Un servicio compartido proporciona ciertos servicios de tiempo de ejecución para varias aplicaciones o servicios al usuario por parte de varias aplicaciones. Los servicios compartidos se crean en el modelo de arquitectura de múltiples usuarios en el que un único despliegue puede dar servicio a varios usuarios o aplicaciones cliente (Soalheira y Timbrell 2014).

d) *Software multiplataforma*

El software multiplataforma, también conocido como cross-platform software, se refiere a aplicaciones o programas que pueden operar en varios sistemas operativos o dispositivos, comúnmente denominados plataformas. Cuando una aplicación es compatible con múltiples plataformas, los usuarios tienen la flexibilidad de utilizar el software en una amplia variedad de dispositivos y computadoras (Triangle 2023).

La ventaja principal del software multiplataforma es su capacidad para funcionar de manera consistente en diferentes sistemas operativos incluso en dispositivos móviles. Esta uniformidad en la experiencia del usuario reduce la curva de aprendizaje, lo que significa que los usuarios pueden ser más productivos y utilizar software con el que están familiarizados sin importar el sistema operativo que utilicen (Cantão et al. 2023).

Existen muchos marcos de desarrollo para crear aplicaciones multiplataforma, un ejemplo de ellos es Flutter; al utilizar estos marcos, los desarrolladores deben realizar el proceso de diseño y codificación solo una vez y luego implementarlos en cualquier lugar seleccionando personalizaciones específicas del objetivo para crear la aplicación final en múltiples sistemas operativos.

e) *Sistema de información de servicios compartidos multiplataforma*

Haciendo una integración de las definiciones anteriores, un Sistema de Información de Servicios Compartidos Multiplataforma es una plataforma tecnológica centralizada que consolida y estandariza funciones empresariales comunes (como recursos humanos, finanzas, tecnologías de información o gestión de pedidos y facturación), operando como un proveedor de servicios interno para múltiples unidades de negocio dentro de una organización, y que es accesible y funcional a través de múltiples plataformas tecnológicas (web, móvil, escritorio), facilitando la interoperabilidad, la eficiencia operativa, la toma de decisiones y



la gobernanza compartida mediante la recolección, procesamiento, almacenamiento y distribución de información en tiempo real (Laudon y Laudon 2012; TechTarget 2024; Triangle 2023).

Este tipo de sistema permite a las organizaciones:

- Conectar múltiples organizaciones, proveedores y servicios a una única plataforma.
- Simplificar el intercambio de datos entre stakeholders.
- Mejorar la interoperabilidad a largo plazo.
- Consolidar, centralizar y estandarizar funciones corporativas de back-office en una fuente compartida para todas las unidades de negocio.
- Proporcionar acceso desde cualquier dispositivo y plataforma, garantizando flexibilidad y accesibilidad.

f) Dimensiones

Las dimensiones de la variable Sistema de información de servicios compartidos multiplataforma son usabilidad y seguridad. Las fuentes más sólidas para establecer las dimensiones de calidad de un software son las normas internacionales ISO/IEC 25010, las cuales definen, entre otras características, tanto la usabilidad como la seguridad.

Además, diversos estudios recientes abordan cómo la usabilidad y la seguridad son dimensiones fundamentales y a menudo interdependientes en los sistemas de información. Estos estudios indican que existe una relación inversa frecuente entre usabilidad y seguridad: aumentar la seguridad puede dificultar la usabilidad, y viceversa. Sin embargo, la tendencia actual es buscar un equilibrio, ya que sistemas poco usables tienden a ser menos seguros en la práctica debido a errores humanos u omisión de controles (Agboola, Adegede y Jacob 2024; Zapata, Moreno y Fernandez-Medina 2013).

La investigación actual respalda que la usabilidad y la seguridad deben ser consideradas de manera conjunta y desde etapas tempranas en el desarrollo de sistemas de información, utilizando metodologías que permitan evaluar y balancear ambas dimensiones para lograr software efectivo y seguro.

i. Usabilidad

La usabilidad es un concepto multifacético que se refiere a la facilidad con la que los usuarios pueden interactuar con un sistema o producto para alcanzar sus objetivos de forma eficaz, eficiente y satisfactoria. La Organización Internacional de Normalización (ISO) define la usabilidad como el grado en que un producto puede ser utilizado por usuarios específicos para alcanzar objetivos específicos en un contexto específico (Chen, Sockel y Falk 2009).

El cuestionario SUS (System Usability Scale) es el instrumento estandarizado más usado para evaluar la usabilidad percibida de software en diversos escenarios. Este cuestionario consta de 10 ítems cuyas respuestas están en escala Likert, proporciona un resultado de 0 a 100. Es independiente de la tecnología, aplicable a software, hardware, sitios web y apps móviles. Ha sido traducido y validado en diferentes idiomas, incluido el español, manteniendo su fiabilidad y validez (Lewis 2018; Sevilla-Gonzalez et al. 2020).

La usabilidad puede dividirse en indicadores derivados del cuestionario SUS (System Usability Scale) de Brooke (1986). Este instrumento no solo evalúa la percepción global de la usabilidad, sino que sus ítems permiten agrupar indicadores operativos ampliamente reconocidos en estudios de UX y sistemas de información.

Los indicadores de usabilidad que se desprenden de SUS son: Eficacia, eficiencia y satisfacción del usuario que fueron descritos en (Brooke 1995). Además, UXables (2024) y WeAreTesters (2025) describen esta agrupación interpretativa como ejes derivados del diseño del SUS.

- **Eficacia:** mide la capacidad del usuario para alcanzar sus objetivos dentro del sistema, reflejando la facilidad de uso y la claridad de la interfaz. Está directamente vinculada a la efectividad en las tareas evaluadas por los ítems 3 (“el sistema era fácil de usar”), 5 (“las funciones estaban bien integradas”) y 7 (“es fácil aprender a usar el sistema”) (uiFromMars 2020).
- **Eficiencia:** refleja el esfuerzo cognitivo y temporal necesario para cumplir tareas, analizado mediante los ítems 2 (“el sistema es innecesariamente complejo”), 6 (“hay demasiada inconsistencia”), 8 (“el sistema es incómodo de usar”) y 10 (“necesité aprender muchas cosas antes de usar el sistema”) (UXables 2024).



- **Satisfacción:** estima el grado de agrado y confianza del usuario en la interacción con el sistema. Se asocia con los ítems 1 (“me gustaría usar el sistema con frecuencia”) y 9 (“me sentí seguro usando el sistema”) (UXables 2024).

ii. Seguridad

La seguridad de la información protege los datos y sistemas frente a amenazas cada vez más sofisticadas. Su objetivo principal es garantizar la confidencialidad, integridad y disponibilidad de la información, tanto en organizaciones como en servicios digitales y entornos personales. Seguridad de la información se centra en proteger la información en cualquier formato, mientras que la ciberseguridad abarca todos los activos en el ciberespacio, incluyendo dispositivos, redes y procesos. La ciberseguridad es un término más amplio que incluye la seguridad de la información, pero también aspectos físicos y de infraestructura (Alexei y Alexei 2023; Kotrikadze 2023).

La seguridad de los sitios web es un tema fundamental en la era digital, ya que los ataques y amenazas evolucionan constantemente. Proteger la información, garantizar la privacidad de los usuarios y mantener la confianza en los servicios en línea requiere una combinación de tecnologías, buenas prácticas y educación tanto para desarrolladores como para usuarios.

Los sitios web enfrentan amenazas como phishing, robo de contraseñas, ataques a través de vulnerabilidades en el código, y riesgos asociados al almacenamiento en la nube y la gestión de contraseñas débiles. La seguridad en dispositivos móviles es especialmente crítica, ya que los usuarios suelen prestar menos atención a las configuraciones de seguridad, lo que facilita ataques como phishing y XSS (Chen et al. 2025).

Diversos estudios y marcos teóricos abordan la seguridad dividiéndola en dimensiones o indicadores como autenticación, inyección y validación, especialmente en el contexto de aplicaciones web, IoT y sistemas distribuidos.

Un enfoque destacado propone una arquitectura flexible basada en patrones de diseño para mitigar debilidades relacionadas con inyección, autenticación y validación. Este modelo utiliza plantillas de seguridad extensibles que asignan estrategias concretas para cada dimensión: autenticación de usuarios, verificación contra inyecciones (como SQLi y XSS) y validación de datos de



entrada. El sistema mapea debilidades específicas a casos de uso y emplea patrones arquitectónicos y de comportamiento para mitigarlas, mostrando cómo cada indicador puede abordarse de manera independiente y complementario (Ezenwoye 2024). Por lo cual la literatura científica respalda la división de la seguridad en las dimensiones de autenticación, inyección y validación, tanto a nivel teórico como práctico, y existen arquitecturas y marcos que abordan explícitamente cada una para fortalecer la seguridad integral de sistemas.

- **Autenticación:** En el mundo de la seguridad informática, la autenticación es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse. El remitente siendo autenticado puede ser una persona que usa un computador, un computador por sí mismo o un programa del computador. En una web de confianza, 'autenticación' es un modo de asegurar que los usuarios son quien ellos 'dicen' ser - que el usuario que intenta realizar funciones en un sistema es de hecho el usuario que tiene la autorización para hacerlo (Cohaila Lara 2020).
- **Inyección:** La inyección de código, especialmente inyección SQL (SQLi) y Cross-Site Scripting (XSS), es una de las vulnerabilidades más críticas que afecta aplicaciones web. Un atacante puede manipular entradas de usuario no validadas para ejecutar comandos no autorizados en la base de datos o scripts maliciosos en el navegador del usuario (Puerta Corredor, Barrera Lasso y Garnica Estrada 2025).
- **Validación:** En el desarrollo de software, la validación consiste en verificar que los datos de entrada y salida cumplen con las reglas definidas por el sistema, con el fin de prevenir vulnerabilidades como inyecciones, corrupción de datos o ejecución no autorizada de código. Su correcta implementación es una práctica esencial dentro de las fases de desarrollo seguro para garantizar la integridad y confiabilidad del sistema (Khan et al. 2022).

3.2.2 Proceso de venta

Bertone (2020) define el proceso de venta como una serie de actividades organizadas y relacionadas que son necesarias para el logro de los objetivos trazados por la gerencia comercial. Desde una perspectiva más integral, Kotler y Armstrong (2012)



sostienen que el proceso de ventas es la serie de pasos que debe seguir un representante de ventas desde la investigación del mercado y del producto hasta el cierre del negocio y más allá. Este proceso operacionaliza el concepto de marketing al convertir estrategias comerciales en transacciones efectivas que generan valor tanto para el cliente como para la empresa.

El concepto más pertinente para ese estudio es el de Ongallo Chaclón (2012), el cual indica que el proceso de venta es una serie de pasos organizados y estratégicos que una empresa o un vendedor de forma individual sigue para convertir a un cliente potencial en un cliente que realiza una compra. Estos pasos están diseñados para guiar al vendedor a lo largo de la interacción con el cliente, desde el primer contacto hasta el cierre de la venta.

El proceso de ventas varía de acuerdo a la industria. En este caso, el proceso de venta en un modelo de negocio multiempresa que tiene servicios compartidos, tiene los siguientes pasos generales:

- Recepción o recojo de información: el cliente elige los productos/servicios que comprará y los hace conocer al personal encargado de ventas.
- Transferencia de órdenes al sistema de información: El encargado de ventas registra los datos de la venta en el sistema de información.
- Entrega de órdenes al cliente: El encargado de ventas entrega los productos y/o servicios al cliente.
- Pago y facturación de venta: El encargado de ventas registra los datos de pago y genera el comprobante de pago de la venta.

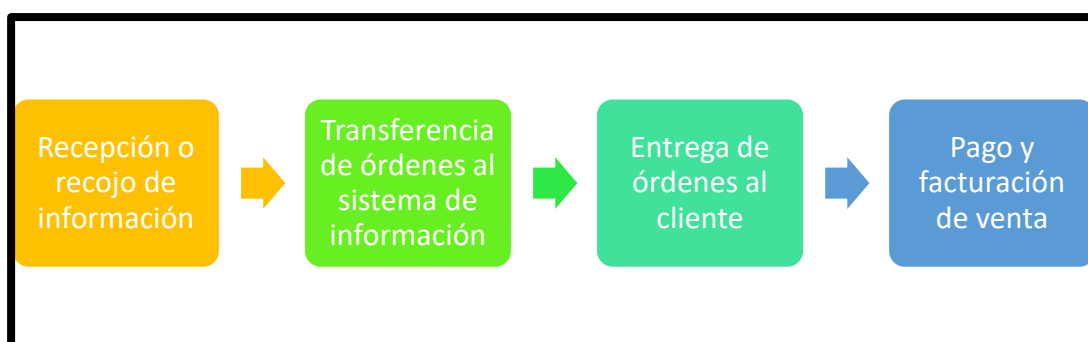


Figura 2 — Pasos generales del proceso de venta

a) Dimensiones

La variable Proceso de venta se divide en las dimensiones proceso de órdenes y proceso de facturación. Estas dimensiones constituyen etapas consecutivas e interdependientes del ciclo comercial completo y agrupan los pasos del proceso venta mostrados en la Figura 2. La eficiencia en el procesamiento de órdenes determina la rapidez con que se puede facturar, y la calidad de la facturación impacta en los tiempos de cobro y la satisfacción del cliente.

Bertone (2020) identifica que el proceso de gestión de pedidos y el proceso de gestión de cobranzas son dos de los procesos fundamentales del área de ventas que toda organización debe gestionar adecuadamente. La correcta ejecución de estos procesos permite:

- Reducir el tiempo del ciclo de ventas
- Mejorar la experiencia del cliente
- Optimizar el flujo de caja
- Incrementar la eficiencia operativa
- Fortalecer la competitividad empresarial

i. Proceso de órdenes

El proceso de órdenes (o gestión de pedidos) abarca el conjunto de actividades administrativas y operativas necesarias para tramitar, aprobar y cumplir las solicitudes de compra de los clientes. Este proceso inicia con la recepción de la orden de compra del cliente y concluye con la preparación del pedido para su expedición (Bertone 2020; Appvizer 2024).

Componentes del proceso de órdenes (RajaPack 2024):

- Recepción y registro de la orden de compra
- Verificación de disponibilidad de inventario
- Validación de datos y condiciones comerciales
- Preparación del pedido (picking y packing)
- Confirmación y programación de entrega

El indicador para la dimensión proceso de órdenes es el Tiempo de proceso de órdenes. El tiempo de procesamiento de órdenes (también conocido como



tiempo del ciclo de pedido u Order Cycle Time - OCT) es un indicador clave de rendimiento que mide el intervalo de tiempo transcurrido desde que se recibe un pedido hasta que el producto es entregado al cliente (CargoSapiens 2025).

ii. Proceso de facturación

El proceso de facturación constituye la etapa administrativa mediante la cual una orden de venta confirmada se convierte en un comprobante fiscal formal que documenta la transacción comercial. Este proceso genera el documento que respalda oficialmente la venta y establece la obligación de pago del cliente (González Jiménez y Novoa Vargas 2024).

Componentes del proceso de facturación (Appvizer 2024):

- Generación y verificación de datos de la orden de compra
- Emisión de la factura con información completa (cliente, productos, precios, impuestos)
- Envío de la factura al cliente
- Registro contable
- Seguimiento de pagos
- Archivo de documentación

El indicador para la dimensión proceso de facturación es el Tiempo de proceso de facturación. El cual mide el lapso que transcurre desde la entrega del producto o servicio hasta la emisión formal de la factura al cliente. Es un indicador importante para la gestión del flujo de caja, ya que retrasos en la emisión impactan directamente en los tiempos de cobro y la liquidez de la empresa (Duemint 2024).

3.3 Marco conceptual

- **Android.** Sistema operativo de código abierto diseñado específicamente para dispositivos móviles con pantalla táctil como smartphones y tabletas, basado en el kernel de Linux. Desarrollado inicialmente por Android Inc. y posteriormente adquirido por Google, Android se ha convertido en el sistema operativo móvil más utilizado globalmente, destacándose por su capacidad de personalización, amplio soporte comunitario



y diversidad de dispositivos hardware que lo implementan a través de la Open Handset Alliance (Ukpabi y Ibrahim 2024).

- **API (Application Programming Interface).** Es una Interfaz de Programación de Aplicaciones que permite la comunicación entre diferentes componentes de software. La falta de una API en el backoffice antiguo era un problema, ya que, por ejemplo, si se decidía desarrollar una aplicación móvil, el código del servidor no estaba preparado para comunicarse con ella debido a que la lógica del servidor renderizaba directamente las vistas y no existía una abstracción de dicha lógica (Goncalves Costa 2017).
- **Aplicaciones multiplataforma.** El desarrollo multiplataforma trata de optimizar la relación costo/beneficio entre las aplicaciones nativas y el desarrollo para cada plataforma, utilizando la misma base de código para las diferentes plataformas. Tienen como ventaja la reducción de tiempos y costos de desarrollo, así como características y rendimiento similares a las nativas. Pero a pesar de esto, sus desventajas son que el rendimiento e interfaces de usuario pueden afectar la experiencia de usuario. Estas aplicaciones multiplataforma se clasifican en: aplicaciones web móviles, híbridas, interpretadas y generadas por compilación cruzada (Delía 2017).
- **Balanceador de carga en sistemas distribuidos modernos.** El balanceo de carga (Load Balancing) en los sistemas distribuidos es la estrategia esencial para la redistribución metódica y eficiente de la carga de trabajo entre los procesadores o nodos. Su objetivo principal es maximizar la utilización de recursos y el rendimiento global del sistema, asegurando que ningún nodo individual se sature mientras otros permanecen inactivos. Esta funcionalidad es clave para mejorar la escalabilidad, la confiabilidad y la tolerancia a fallos, minimizando los tiempos de respuesta ante una demanda creciente (Ali y Khan 2012).
- **Cliente.** Desde la perspectiva del servicio este concepto tiene diferentes significados: i) Cliente es aquél que pertenece al público objetivo. ii) Cliente es el comprador potencial. iii) Cliente es el comprador eventual. iv) Cliente es el comprador habitual (Paz 2005).
- **Dart.** Lenguaje de programación diseñado para el desarrollo cliente, como para aplicaciones web y móviles. Es una herramienta muy poderosa para crear aplicaciones web y móviles debido a su estabilidad y facilidad de aprendizaje (Urathal, Vinodhini y Sasikala 2021).



- **Facturación.** Es el proceso de generar un documento mercantil que contenga toda la información de una operación de compraventa. Como: fecha de devengado de la operación, cantidad de productos o servicios entregados, precio unitario e importe total, descuentos e impuestos. La importancia de la factura es que es justificante fiscal y garantiza que se ha entregado el producto o servicio (Carrasco y Núñez 2023).
- **Facturación electrónica.** La facturación electrónica es un proceso de generación, envío y almacenamiento de facturas y otros documentos relacionados con transacciones comerciales en formato digital o electrónico (Jamdaade y Champaneri 2015). En lugar de utilizar papel impreso, las empresas y organizaciones generan facturas electrónicas en archivos digitales, generalmente en formatos como PDF, XML u otros estándares electrónicos. Estos documentos electrónicos contienen la misma información que una factura impresa, como detalles de la transacción, los productos o servicios vendidos, el precio, los impuestos, el nombre del cliente y del proveedor, entre otros datos. En muchos países, la facturación electrónica se ha convertido en un requisito legal y fiscal. Las autoridades tributarias suelen establecer estándares y regulaciones específicas para la generación y el intercambio de facturas electrónicas, con el fin de garantizar la integridad de los datos y facilitar la supervisión fiscal (Dzyublyuk 2020).
- **Flutter.** SDK (Software Development Kit) de código abierto desarrollado y mantenido por Google, dedicado al desarrollo de aplicaciones multiplataforma de alto rendimiento (Android, iOS, Web y Desktop) desde una única base de código. Flutter se distingue por no utilizar web views o widgets OEM del dispositivo, sino que utiliza su propio motor de renderizado (Skia). El uso de compilación AOT (Ahead-of-Time) permite que el código sea compilado directamente a código ARM nativo, lo que le otorga un rendimiento comparable al de las aplicaciones desarrolladas de forma nativa (Tashildar et al. 2020).
- **HTTPS** (HTTP over TLS). Es la versión segura del protocolo HTTP, a la que se agregó cifrado en el año 2000. Esta evolución empoderó a las empresas con una comunicación mucho más segura con sus clientes, ya que permite a los clientes verificar la autenticidad del sitio web a través de certificados. Al cifrar todas las comunicaciones, los ataques de tipo "man-in-the-middle" se vuelven casi imposibles (Goncalves Costa 2017).
- **iOS.** Sistema operativo móvil propietario desarrollado por Apple Inc. para todos sus dispositivos móviles, incluyendo iPhone, iPad y iPod touch. Basado tecnológicamente en Mac OS, iOS representa el segundo sistema operativo móvil más popular a nivel



global después de Android, caracterizándose por su interfaz de usuario altamente intuitiva, estricto control de aplicaciones a través de la App Store y optimización perfecta con el hardware Apple, aunque con limitadas opciones de personalización en comparación con alternativas de código abierto (Ukpabi y Ibrahim 2024).

- **Java.** Lenguaje de programación orientado a objetos diseñado para gestionar la complejidad del software. Proporciona un conjunto de bibliotecas y especificaciones que permiten el diseño y despliegue de aplicaciones multiplataforma, siendo utilizado en diversos dominios como sistemas empresariales y servidores web (Gherardi, Brugali y Comotti 2013).
- **JavaScript.** Es un lenguaje de programación de alto nivel, interpretado, caracterizado por su naturaleza dinámica y su tipado débil (o no tipado). Su núcleo se define a través del estándar internacional ECMAScript (ECMA-262). Es conocido como el lenguaje de programación de la Web y, a pesar de su origen en el frontend, su expansión a través de entornos como Node.js lo ha convertido en un lenguaje full-stack capaz de gestionar tanto la lógica del lado del cliente como la del lado del servidor (Martins de Pinho 2017).
- **JWT (JSON Web Tokens).** representan un estándar de código abierto para la creación de tokens de acceso que se utilizan para la autenticación y la transferencia segura de información entre partes como objetos JSON. Los JWT son compactos, seguros y permiten a los servicios (Service API) verificar la identidad y los permisos del cliente de forma descentralizada y sin necesidad de consultar repetidamente una base de datos central. Esto resulta crucial para el diseño de arquitecturas stateless (sin estado) y para asegurar la comunicación en sistemas de microservicios, como el utilizado para las APIs de Rumisof-POS (Jones, Bradley y Sakimura 2015).
- **Kanban.** El método Kanban se ha consolidado como la adición más reciente al área de investigación del desarrollo de software ágil y Lean. Derivado de los sistemas de producción Just-In-Time, Kanban es un mecanismo de control de flujo pull-driven (impulsado por la demanda) que guía a los equipos a visualizar el flujo de trabajo, limitar el trabajo en curso (WIP) y medir el tiempo de ciclo en cada etapa. Este enfoque es fundamental para el diseño de sistemas modernos como Rumisof-POS, ya que los beneficios reportados incluyen la mejora del tiempo de entrega (lead time), el aumento de la calidad del software, la mejora en la comunicación y la coordinación, y



la disminución de los defectos reportados por el cliente. En última instancia, su adopción es clave para mantener un flujo de desarrollo constante y un ritmo sostenible (Anderson 2010).

- **Laravel.** Es un framework de desarrollo web basado en PHP que implementa el patrón arquitectónico Modelo-Vista-Controlador (MVC). Su diseño enfatiza la productividad mediante una sintaxis expresiva y la inclusión de herramientas avanzadas para la gestión de bases de datos (Eloquent ORM) y el manejo de peticiones (Routing y Middleware). Estas características lo consolidan como una plataforma robusta y modular para el desarrollo de aplicaciones web complejas y APIs RESTful (Subecz 2021).
- **Microsoft Windows.** Es un sistema operativo para estaciones de trabajo domésticas y empresariales desarrollado por Microsoft, caracterizado por ser la interfaz gráfica de usuario original que rápidamente dominó la industria de las computadoras personales. Windows es actualmente el sistema operativo más utilizado a nivel mundial, habiendo evolucionado significativamente desde sus primeras versiones hasta convertirse en una plataforma muy amigable para el usuario y compatible con la mayoría de dispositivos y softwares disponibles en el mercado (Ukpabi y Ibrahim 2024).
- **MIPYMES.** La Micro, Pequeña y Mediana Empresa (MIPYME) se define como una unidad económica constituida bajo cualquier forma legal para desarrollar actividades de extracción, transformación, producción, o comercialización de bienes o servicios (Congreso de la República del Perú 2003). Su estudio académico, sin embargo, requiere una perspectiva multidisciplinaria debido a la notable heterogeneidad en los enfoques, que incluyen dimensiones sociales, administrativas y económicas (Alarcón Osuna, Ocampo Figueroa y Guzmán Anaya 2024). La delimitación rigurosa se establece mediante el cumplimiento de criterios concurrentes que combinan el número de trabajadores y el volumen de ventas anuales (INEI 2013). En la mayoría de las legislaciones, esta clasificación segmenta a la Microempresa (ej. hasta 10 trabajadores) y la Pequeña Empresa (ej. hasta 100 trabajadores), siendo su tamaño un factor crucial que determina su acceso al sistema financiero formal y su vulnerabilidad estructural (Ministerio de la Producción 2024).
- **Node.js.** Es un entorno de ejecución de JavaScript de código abierto, basado en el motor V8 de Google, que permite el desarrollo de aplicaciones del lado del servidor (backend). Su diseño se fundamenta en un modelo event-driven (orientado a eventos)



e I/O no bloqueante (non-blocking), lo cual le permite ser rápido y confiable para aplicaciones que manejan cargas pesadas de red y archivos (Shah y Soomro 2017).

- **Orden.** Orden de pedido o de compra es el acuerdo en el que ambas partes (vendedor y cliente) se ponen de acuerdo en los términos básicos de la compra: los productos que se comprarán, la cantidad de cada uno y el precio pactado (Carrasco y Núñez 2023).
- **OWASP** (Open Web Application Security Project). Es una comunidad abierta y global dedicada a mejorar la seguridad del software, siendo su lista OWASP Top 10 la referencia estándar de la industria para identificar los riesgos de seguridad más críticos en aplicaciones web. Adoptar las directrices de OWASP permite al equipo de desarrollo mitigar las vulnerabilidades de software desde las fases iniciales del diseño y la codificación, garantizando que el sistema Rumisof-POS cumpla con los más altos estándares de protección de datos y defensa contra ciberataques, tales como la inyección de código o la ruptura de control de acceso (OWASP Foundation 2021).
- **Patrón Bloc.** El patrón BLoC, que significa "Business Logic Component," fue concebido por Paolo Soares y Cong Hu de Google y se dio a conocer en la Dart Conference 2018. Este patrón permite reutilizar código entre sus aplicaciones móviles, desarrolladas con Flutter y Dart, y sus aplicaciones web, que utilizaban Angular Dart. BloC actúa como un intermediario entre las vistas y el modelo de la aplicación, desempeñando un rol similar al de un presentador en el patrón MVP o un modelo de vista (view model) en el contexto del MVVM. La definición del modelo puede variar ampliamente según el tamaño y la arquitectura de la aplicación, pudiendo abarcar desde la capa de datos en una arquitectura simplificada hasta el dominio o el hexágono en estructuras como la Clean Architecture o Hexagonal Architecture.
- **Proceso de venta.** La venta no es solo el acto en el que se compra un producto o servicio. Esta pueda ser venta al comprador final si se realiza en lugares de venta al detalle o puede ser venta de empresa a empresa si el fabricante vende al mayorista o intermediarios. El proceso de venta abarca varias etapas: i) establecer una asociación con el cliente y generación de confianza. ii) reconocer las necesidades del comprador (comprador final o empresa). iii) Elegir la diferenciación de nuestro producto o servicio. iv) Socializar nuestra ventaja competitiva al comprador. v) Establecer servicios posventa que generen relaciones duraderas con el comprador (Publicaciones Vértice 2008).

- **Servicios compartidos.** Un servicio compartido proporciona ciertos servicios de tiempo de ejecución para varias aplicaciones o servicios al usuario por parte de varias aplicaciones. Los servicios compartidos se crean en el modelo de arquitectura de múltiples usuarios en el que un único despliegue puede dar servicio a varios usuarios o aplicaciones cliente (Soalheira y Timbrell 2014).
- **SOAP** (Simple Object Access Protocol). Es un protocolo ligero que es independiente de la plataforma, del transporte y del sistema operativo, todo porque está construido utilizando sistemas probados en el tiempo como el protocolo HTTP y el marcado de texto en XML (Suda 2003).
- **Swift.** Lenguaje de programación compilado para aplicaciones de iOS, watchOS, tvOS, macOS y Linux, creado para facilitar la creación de código de manera más cómoda y segura, así como para hacer más fácil el aprendizaje de la programación. Fue presentado por primera vez en la Worldwide Developers Conference de Apple en 2014 y se caracteriza por ofrecer una sintaxis más simple, código más corto y programas más seguros (Fojtik 2020).
- **Ubuntu.** Es una distribución del sistema operativo GNU/Linux, de código abierto y completamente gratuito. Desarrollado a partir de Debian en 2004 por Canonical Ltd., opera bajo la Licencia Pública General de GNU (GPL) y emplea un kernel monolítico. Está disponible en versiones para escritorio y servidor, compatible con arquitecturas Intel, AMD y ARM, y utiliza APT para la gestión de paquetes. (Gkrekos et al. 2019)
- **Web Scraping.** También conocido como extracción o recolección web, es una técnica para extraer datos de la web y guardarlos en un sistema de archivos o base de datos para su posterior recuperación o análisis. Así, el web scraping es ampliamente reconocido como una técnica eficiente y poderosa para recopilar grandes datos y adaptarse a una variedad de escenarios (Zhao 2017).
- **XML** (Extensible Markup Language). El Lenguaje de Marcado Extensible (XML) constituye un estándar fundamental para la estructuración, almacenamiento e intercambio de datos entre sistemas heterogéneos, especialmente en aplicaciones distribuidas y de comercio electrónico. Su diseño, basado en la simplicidad y la universalidad, permite definir estructuras de datos personalizadas y facilita la interoperabilidad entre diferentes plataformas y lenguajes de programación, siendo la base de estándares de comunicación como SOAP y, de manera crucial en entornos tributarios,



del formato UBL (Universal Business Language) para la facturación electrónica (W3C 2013).



CAPÍTULO IV

METODOLOGÍA

4.1 Tipo y nivel de investigación

4.1.1 Tipo de investigación

La investigación fue aplicada, este es un enfoque de investigación científica que se centra en la resolución de problemas prácticos y la aplicación de los resultados de la investigación en situaciones del mundo real. A diferencia de la investigación pura o básica, cuyo objetivo principal es ampliar el conocimiento teórico sin necesariamente tener una aplicación inmediata, la investigación aplicada tiene un enfoque más orientado a la acción y se concentra en la utilidad y relevancia de los hallazgos (Esteban Nieto 2018).

4.1.2 Nivel de investigación

El nivel de investigación fue explicativo, ya que estuvo dirigido a responder por las causas de los eventos y fenómenos, enfocándose en explicar por qué ocurre el fenómeno y bajo qué condiciones se manifiesta la relación entre las variables (Hernández-Sampieri y Mendoza 2018).

4.2 Diseño de investigación

Un diseño preexperimental es un tipo de diseño de investigación en la que solo se tiene un grupo, dentro de los preexperimentos existe el tipo de diseño preprueba/posprueba con un solo grupo (Hernández-Sampieri y Mendoza 2018). En este diseño a un único grupo se le aplica una medición antes del estímulo, luego se interviene con la variable independiente (tratamiento) y finalmente se realiza una medida posterior de la variable dependiente.

Este diseño se puede expresar gráficamente de la siguiente manera:



donde



G	Grupo único (muestra)
M_1	Medida previa al tratamiento (preprueba)
X	Tratamiento o estímulo (Sistema de información)
M_2	Medida posterior al tratamiento (posprueba)

4.3 Población y muestra

4.3.1 Población

La población de estudio se refiere a un conjunto definido, limitado y accesible de casos que sirve como punto de referencia para la selección de la muestra, siguiendo criterios predefinidos. (Arias-Gómez, Villasís-Keever y Miranda Novales 2016).

La población de estudio es la cantidad total de empresas usuarias del sistema de servicios compartidos multiplataforma “Rumisof-pos”, que asciende a 47 empresas.

4.3.2 Muestra

Es un subconjunto representativo de una población más amplia que se selecciona con el propósito de realizar investigaciones y obtener conclusiones que puedan aplicarse a la población en su conjunto. En lugar de estudiar o recopilar datos de toda la población, que puede ser grande o incluso infinita en algunos casos, se selecciona una muestra más manejable y práctica para analizar y sacar inferencias.

En el caso de esta investigación, la muestra fue la totalidad de la población es decir se hizo un muestreo censal en el que se tomaron las 47 empresas usuarias del sistema de servicios compartidos multiplataforma “Rumisof-pos”.

4.4 Procedimiento

El presente proyecto de investigación, que culmina en la construcción de una solución de software multiplataforma, se ejecutó mediante un procedimiento fásico, iterativo y sistémico. Dicho procedimiento fue gestionado bajo un marco de trabajo ágil adaptado, que permitió la flexibilidad en el flujo de trabajo y aseguró la trazabilidad completa desde la conceptualización hasta el despliegue final.

La ejecución del proyecto se articuló en cinco fases macro, cada una con objetivos específicos y entregables definidos.

4.4.1 Fase de definición y análisis

Esta fase inicial tuvo como objetivo establecer la justificación teórica y los límites funcionales precisos de la solución de software.



a) *Análisis de literatura y estado del arte*

1. Revisión arquitectónica: Estudio comparativo de patrones de arquitectura de software, validando la idoneidad de un enfoque de microservicios e identificando las mejores prácticas para la construcción de API.
2. Fundamentación tecnológica: Análisis de la curva de aprendizaje, madurez, rendimiento y soporte de los frameworks seleccionados (Laravel, Flutter, No-de.js) para justificar su inclusión en el ecosistema.
3. Marco regulatorio: Investigación exhaustiva sobre la normativa fiscal y los estándares técnicos (como UBL y el manejo del CDR) que rigen la emisión y recepción de la Factura Electrónica.

b) *Ingeniería y documentación de requisitos:*

1. Levantamiento funcional: Recolección y clasificación de los requisitos esenciales que definen el comportamiento de la aplicación (ej. la secuencia lógica para la emisión de un comprobante, la gestión de inventario).
2. Definición no funcional: Establecimiento de métricas críticas de calidad, rendimiento y seguridad (ej. latencia de la API en X milisegundos, concurrencia máxima, nivel de cifrado de las transacciones).
3. Instrumentación de requisitos: Documentación formal de los requisitos en la herramienta de gestión de conocimiento (Notion), incluyendo criterios de aceptación para cada ítem.

4.4.2 Fase de diseño conceptual y técnico

Con base en los requisitos aprobados, esta fase se centró en la conceptualización de la solución y la planificación de su construcción.

a) *Conceptualización de la arquitectura de software:*

1. Diseño de capas: Definición de la segmentación de la lógica de negocio, especificando los roles de la API principal (construida con Laravel) y los microservicios especializados de alto tráfico o asíncronos (implementados con Node.js).
2. Definición de protocolos: Establecimiento de la comunicación cliente-servidor mediante API REST para transacciones internas y la integración obligatoria con servicios externos (entidades tributarias) a través del protocolo SOAP para el intercambio de documentos UBL.



b) Modelamiento de la base de datos:

1. Diseño lógico (MER): Creación del Modelo Entidad-Relación (MER) para abstraer la estructura de datos requerida (clientes, productos, comprobantes, etc.).
2. Diseño físico y rendimiento: Normalización del esquema de la base de datos y diseño de índices estratégicos para optimizar el rendimiento de las consultas, especialmente aquellas relacionadas con grandes volúmenes de transacciones históricas.

c) Diseño de la interfaz y experiencia de usuario (UX/UI):

1. Wireframing: Elaboración de bocetos de bajo nivel (utilizando Xcalidraw) para definir la estructura y la navegación de la aplicación móvil.
2. Diseño de Interfaz: Creación de un sistema de diseño consistente, adaptado al enfoque de widgets de Flutter, que garantice la coherencia visual y la responsividad en plataformas iOS, Android y Web.

4.4.3 Fase de desarrollo y construcción iterativa

Esta fase comprendió la codificación, gestión de versiones y la construcción de los componentes definidos en la fase anterior, ejecutándose de manera continua e iterativa.

a) Configuración del entorno de desarrollo y control de versiones:

1. Gestión del repositorio: Creación de la estructura del repositorio central y definición de la estrategia de branching utilizando Git. Se seleccionó el cliente gráfico Fork para estandarizar las operaciones complejas de gestión del historial y las fusiones (merges).
2. Instrumentación local: Preparación de los entornos de desarrollo (IDE: VS Code) y la herramienta universal de administración de bases de datos (Dbeaver) para el equipo de desarrollo.

b) Desarrollo del back-end y servicios:

1. Implementación de la API core: Codificación de la lógica de negocio principal y las capas de servicio utilizando el framework Laravel (PHP).
2. Desarrollo de microservicios: Creación de los servicios asíncronos y de alto rendimiento utilizando Node.js y JavaScript, enfocándose en tareas de notificación o procesamiento en segundo plano.



3. Conector de facturación: Desarrollo del cliente SOAP especializado para generar, firmar y enviar documentos UBL a la entidad fiscal, y para procesar la CDR.

c) Desarrollo multiplataforma (Front-End):

1. Codificación con Dart/Flutter: Implementación de las interfaces de usuario mediante el lenguaje Dart y el framework Flutter, asegurando la reutilización del código en las plataformas objetivo (iOS, Android, Web).
2. Gestión de estado: Aplicación de un patrón robusto de gestión de estado para manejar la complejidad de las interacciones de usuario y la sincronización de datos con el back-end.

d) Integración continua de componentes:

1. Pruebas de conectividad: Verificación de la serialización y deserialización de datos (JSON) entre el front-end (Flutter) y el back-end (Laravel/Node.js).
2. Pruebas de carga de módulos: Asegurar que los componentes nativos (si aplica, con Swift en iOS) se comuniquen correctamente con la capa Dart.

4.4.4 Fase de pruebas, optimización y despliegue

Esta fase se dedicó a la validación de la calidad, el rendimiento y la implementación del sistema en un entorno operativo real.

a) Validación integral de la calidad (QA):

1. Pruebas unitarias: Ejecución de pruebas automatizadas sobre las clases y funciones de la lógica de negocio (Laravel y Node.js) para garantizar el funcionamiento aislado de los componentes.
2. Pruebas de integración: Verificación del flujo de datos completo, desde la interfaz de Flutter hasta la base de datos y el servicio de facturación electrónica (incluyendo la correcta recepción del CDR).
3. Pruebas de aceptación de usuario (UAT): Sesiones estructuradas con usuarios finales para validar que el sistema cumpla con los requisitos funcionales de negocio en un entorno simulado o de staging.

b) Despliegue y puesta en producción:

1. Configuración del servidor: Preparación y endurecimiento del servidor de producción utilizando Ubuntu.



2. Orquestación de servicios: Despliegue de los servicios Node.js mediante el administrador de procesos PM2 para asegurar la estabilidad, el balanceo de carga y la persistencia de las aplicaciones.
3. Liberación de aplicaciones: Compilación de las aplicaciones nativas para Android e iOS y su distribución correspondiente.

4.4.5 Fase de evaluación y consolidación final

La fase de cierre se enfocó en la medición del impacto y el cumplimiento de los objetivos planteados en la investigación.

a) Análisis cuantitativo y cualitativo de resultados:

1. Medición de rendimiento: Evaluación de las métricas de latencia de la API, el tiempo de ciclo del proceso de facturación y la eficiencia de la aplicación móvil.
2. Cumplimiento de requisitos: Verificación formal de que la solución desarrollada satisface todos los requisitos funcionales y no funcionales definidos anteriormente.

4.5 Técnicas e instrumentos

- Para la variable Sistema de información de servicios compartidos multiplataforma, se midieron sus dimensiones usabilidad y seguridad.
 - La usabilidad se midió con la técnica de la encuesta y aplicando el instrumento Cuestionario SUS (System Usability Scale), el cual es un cuestionario validado y ampliamente utilizado para medir este atributo (Brooke 1986). Este cuestionario tiene 10 ítems o enunciados predefinidos, las respuestas son en escala Likert de cinco niveles. De los 10 enunciados, los impares son positivos y los pares negativos. El puntaje final tiene un rango de 0 a 100, se calcula de la siguiente manera: $((\text{Suma de impares} - 5) + (25 - \text{Suma de pares})) * 2.5$. Este cuestionario fue auto-administrado, se envió en formato digital a los responsables de cada empresa para su aplicación. Se muestra el formulario en el Anexo B.
 - Para la seguridad, se aplicó una batería de pruebas de seguridad al sistema de información alineados con prácticas recomendadas por OWASP (Open Worldwide Application Security Project) utilizando herramientas automatizadas y manuales las cuales analizaron el sistema y mostraron los resultados. Por lo que se usó la

técnica de análisis documental junto con el instrumento de ficha de recolección de datos que se muestra en la Tabla 3.

Tabla 3 — Ficha de recolección de datos de la dimensión seguridad

Ejecución	Fecha	Total Pruebas	PASS	FAIL	% Éxito

- Debido a que el estudio es de diseño preexperimental, la variable dependiente Proceso de venta, se midió en dos oportunidades, antes y después del uso del Sistema de información de servicios compartidos multiplataforma.
 - Para la preprueba o medición antes de la implementación del sistema de información se utilizó la técnica de la encuesta con su instrumento cuestionario que es un conjunto de preguntas que miden una o más variables (Hernández-Sampieri y Mendoza 2018). Este cuestionario fue un formulario web autoadministrado, cuyo enlace se envió a los encargados de las empresas. Este cuestionario se muestra en el Anexo C. La validación de este instrumento se realizó por juicio de expertos, el cual indicó que el cuestionario es Aplicable. Se adjuntan los formatos de validación en el Anexo D. Para demostrar la confiabilidad del instrumento se realizó una prueba piloto a 10 individuos, y se calculó el coeficiente de confiabilidad de alfa de Cronbach el cual arrojó el valor de 1.00 como se observa en la Tabla 4, este valor al ser mayor a 0.9 indica que el cuestionario es Excelente, de acuerdo a los criterios de interpretación del coeficiente de (George y Mallery 2003).

Tabla 4 — Análisis de confiabilidad

Alfa de Cronbach	N de elementos
1.00	2

- Para la posprueba, al tener ya implementado el sistema de información, se programó el software para que realice las mediciones y las almacene, por lo que la técnica empleada fue el análisis documental con el instrumento ficha de recolección de datos que se muestra en la Tabla 5. El análisis documental radica en explorar datos almacenados en documentos ya existentes, como bases de datos, actas, informes entre otros (Lifeder 2021).

Tabla 5 — Ficha de recolección de datos de la posprueba

Nº	EMPRESA (Razón Social)	RUC	Orden (tiempo en min)	Facturación (tiempo en min)	Venta (tiempo en min)

4.6 Análisis estadístico

Una vez recogidos los datos, estos se constituyeron mediante hojas de cálculo en la matriz de datos. A continuación, se utilizó un software estadístico para realizar el análisis estadístico. Primero se realizó el análisis estadístico descriptivo de las variables y dimensiones mediante tablas de frecuencias y gráficos. Finalmente se ejecutó el análisis estadístico inferencial para comprobar las hipótesis, para lo cual se realizó la prueba de normalidad de Shapiro-Wilk para determinar el tipo de análisis paramétrico o no paramétrico, y en base a esto se aplicó la prueba de comparación de medias debido a la naturaleza de las hipótesis. Esta comparación se realizó con el estadístico no paramétrico de Wilcoxon.

El estadístico no paramétrico de Wilcoxon tiene dos fórmulas principales, dependiendo del tipo de comparación que se realice. Para esta investigación aplicamos la Prueba de Rangos con Signo de Wilcoxon (T), la cual se utiliza para muestras relacionadas o pareadas (ej. una misma muestra medida "antes" y "después"). Evalúa si existe una diferencia significativa en la mediana de las diferencias entre los pares de datos.

El estadístico de prueba es T (a veces se usa W), que se define simplemente como el menor valor entre la suma de los rangos positivos (R_+) y la suma de los rangos negativos (R_-).

$$T = \min(R_+, R_-)$$

donde

R_+ es la suma de los rangos de las diferencias positivas.

R_- es la suma de los rangos de las diferencias negativas.

CAPÍTULO V

RESULTADOS Y DISCUSIÓN

En este capítulo se presenta el análisis estadístico descriptivo de las variables de estudio y a continuación se realiza el análisis estadístico inferencial de comprobación de hipótesis.

5.1 Análisis de resultados

En cuanto al análisis descriptivo de la variable independiente Sistema de información de servicios compartidos multiplataforma, esta variable se conforma de dos dimensiones: usabilidad y seguridad.

Sobre la dimensión usabilidad, se aplicó el cuestionario SUS, luego de tabular los datos se calculó el puntaje total por cada empresa en el rango de 0 a 100, estos puntajes se categorizaron para su interpretación en: Aceptable [100-70], marginal <70-50] e inaceptable <50-0] (Sauro 2018). Esta matriz de datos se muestra a detalle en el Anexo E. A continuación, se presenta la tabla de frecuencia de las puntuaciones categorizadas obtenidas por cada empresa en el cuestionario SUS.

Tabla 6 — Tabla de frecuencias de la dimensión usabilidad

Categoría	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Aceptable	47	100,0	100,0	100,0

La Tabla 6 muestra que el 100% de las empresas evaluó el sistema con usabilidad aceptable, las categorías marginal e inaceptable no tuvieron frecuencia alguna por lo que no se muestran en la tabla, lo mismo refleja la Figura 3 a continuación.



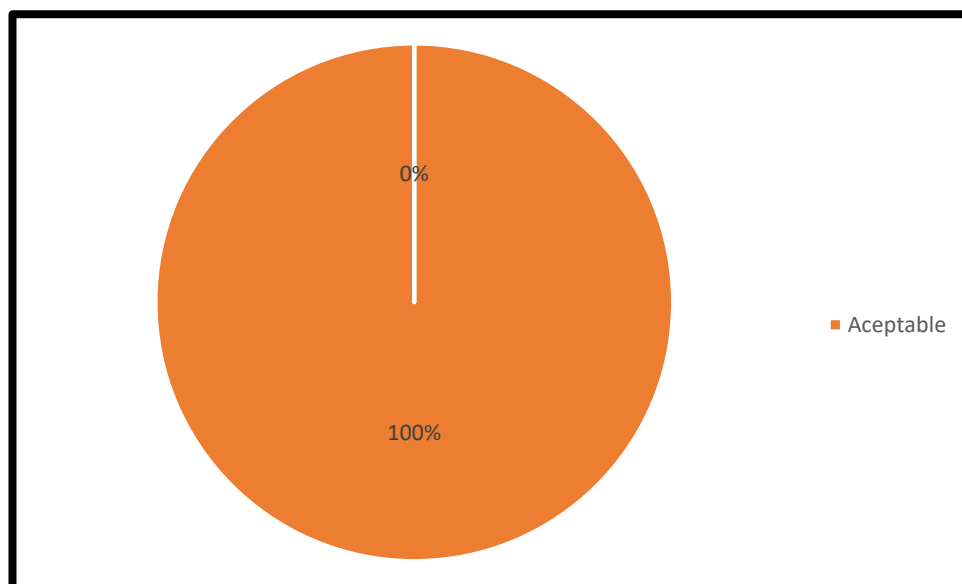


Figura 3 — Categorías de la dimensión usabilidad

Con respecto a la dimensión seguridad, se realizaron pruebas al sistema alineadas de acuerdo a OWASP, ver Figura 4, se realizaron varias ejecuciones en diferentes fechas, cada ejecución consistió en 16 pruebas de diferentes categorías.

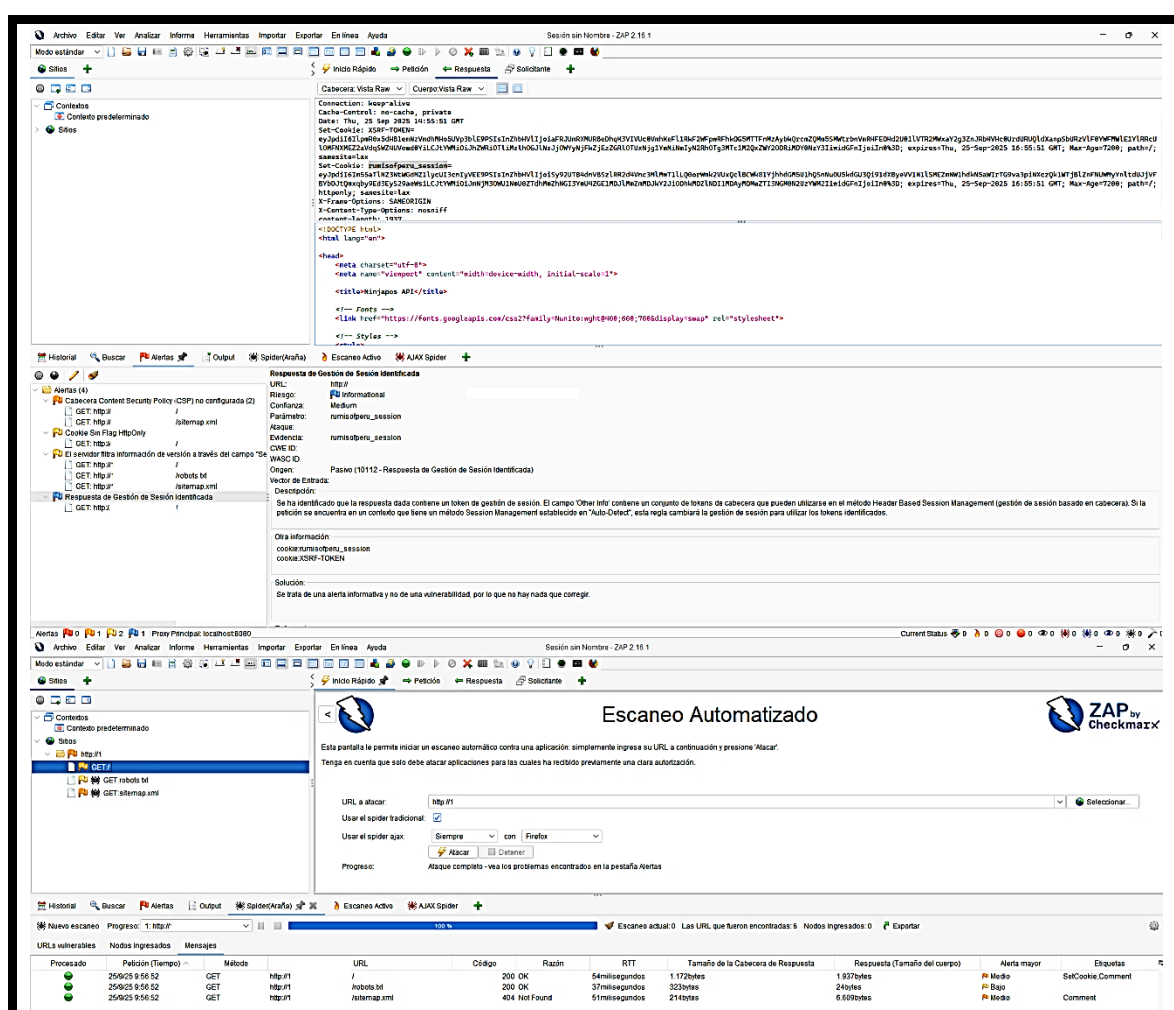


Figura 4 — Ejecución de pruebas de seguridad bajo OWASP

En cada ejecución los resultados para cada prueba fueron FAIL (fallido) o PASS (aprobado). En las primeras ejecuciones la mayoría de pruebas resultaron FAIL, se corrigieron y se volvieron a realizar ejecuciones. En total se realizaron 15 ejecuciones hasta que todas las pruebas se estabilizaron en PASS, en el Anexo F se muestra los datos obtenidos en las ejecuciones de las pruebas. A continuación, en la Tabla 7 se muestra un resumen de las ejecuciones.

Tabla 7 — Tabla resumen de la ejecución de las pruebas de seguridad

Ejecución	Total Pruebas	PASS	FAIL	% Éxito
Ejecución 1	16	4	12	25%
Ejecución 2	16	3	13	19%
Ejecución 3	16	2	14	13%
Ejecución 4	16	6	10	38%
Ejecución 5	16	5	11	31%
Ejecución 6	16	12	4	75%
Ejecución 7	16	5	11	31%
Ejecución 8	16	8	8	50%
Ejecución 9	16	8	8	50%
Ejecución 10	16	13	3	81%
Ejecución 11	16	15	1	94%
Ejecución 12	16	11	5	69%
Ejecución 13	16	16	0	100%
Ejecución 14	16	16	0	100%
Ejecución 15	16	16	0	100%

Como se mencionó, se observa que el porcentaje de éxito en las primeras ejecuciones fue bajo, se hicieron las correcciones y este porcentaje fue subiendo hasta alcanzar el 100%. La evolución de las pruebas FAIL y PASS se muestran también en la Figura 5.

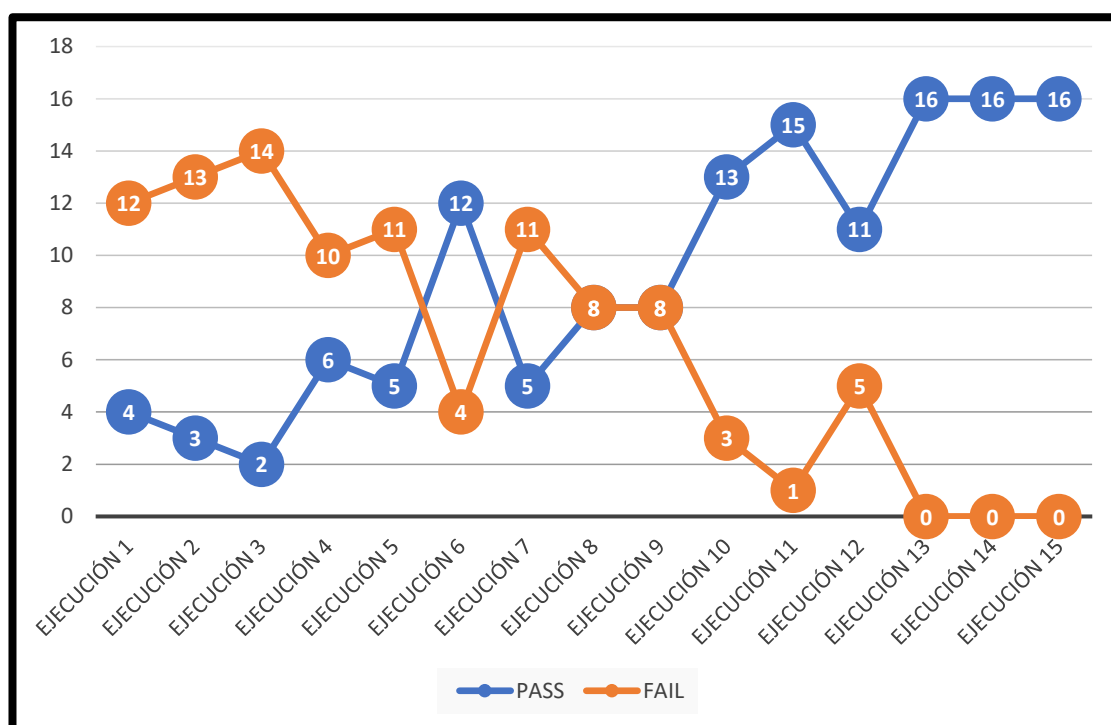


Figura 5 — Evolución de las ejecuciones

Sobre la variable dependiente Proceso de venta, esta tiene dos dimensiones: proceso de órdenes y proceso de facturación.

Sobre la dimensión proceso de órdenes, esta se midió en dos oportunidades antes y después de la implementación del sistema como se observa en el Anexo G. Aquí se observa que el tiempo del proceso de órdenes antes de la implementación del sistema tuvo un promedio de 5.88 minutos, mientras que después de la implementación el promedio fue de 1.91 minutos. Lo cual representa una disminución en el tiempo de órdenes en un 67.5%.

En la Figura 6 se muestra la representación de los tiempos del proceso de órdenes antes y después de la implementación del Sistema de información de servicios compartidos multiplataforma. En esta figura se puede observar que los tiempos de proceso de órdenes antes fueron superiores a los tiempos de proceso de órdenes después de la aplicación del Sistema de información de servicios compartidos multiplataforma.

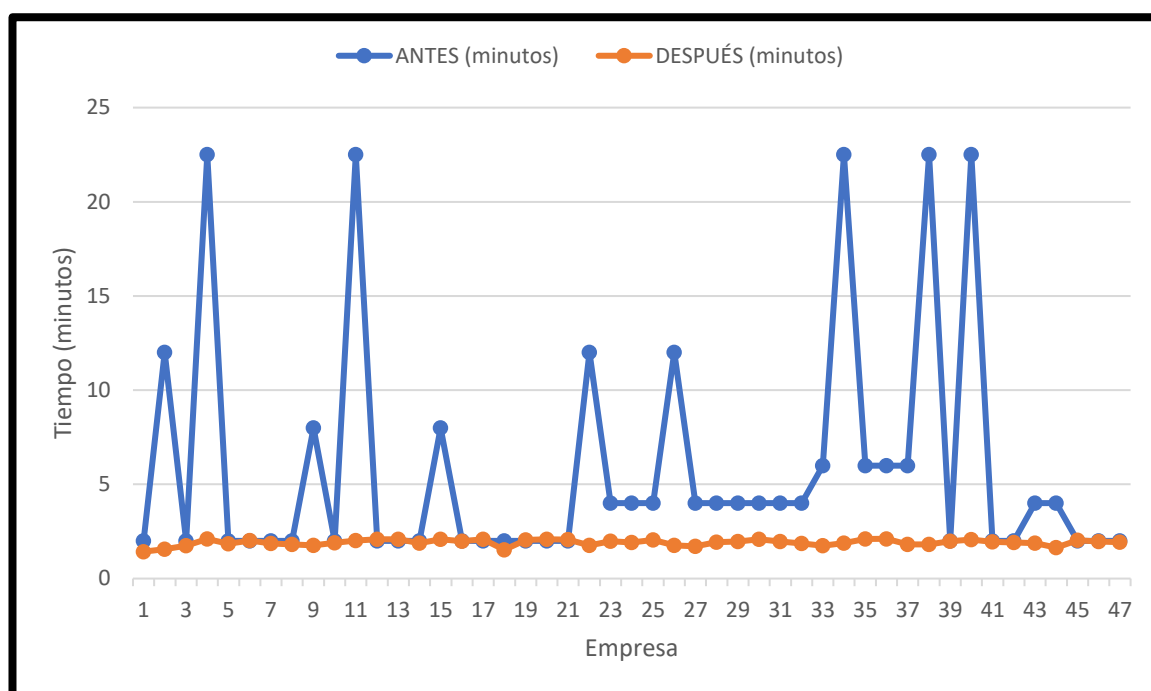


Figura 6 — Tiempos de proceso de órdenes antes y después

Sobre la dimensión proceso de facturación, esta se midió en dos oportunidades antes y después de la implementación del sistema como se observa en el Anexo G, el promedio de los tiempos del proceso de facturación recogidos a las empresas usuarias antes de la implementación del sistema fue de 5.88 minutos, y después de la implementación el promedio fue de 0.82 minutos. Lo cual representa una disminución en el tiempo de facturación de 86.1%.

A continuación, se muestra en la Figura 7 la representación de los tiempos del proceso de facturación antes y después del Sistema de información de servicios compartidos multiplataforma. En esta figura se puede observar que los tiempos de proceso de facturación antes fueron superiores a los tiempos de proceso de facturación después de la aplicación del Sistema de información de servicios compartidos multiplataforma.

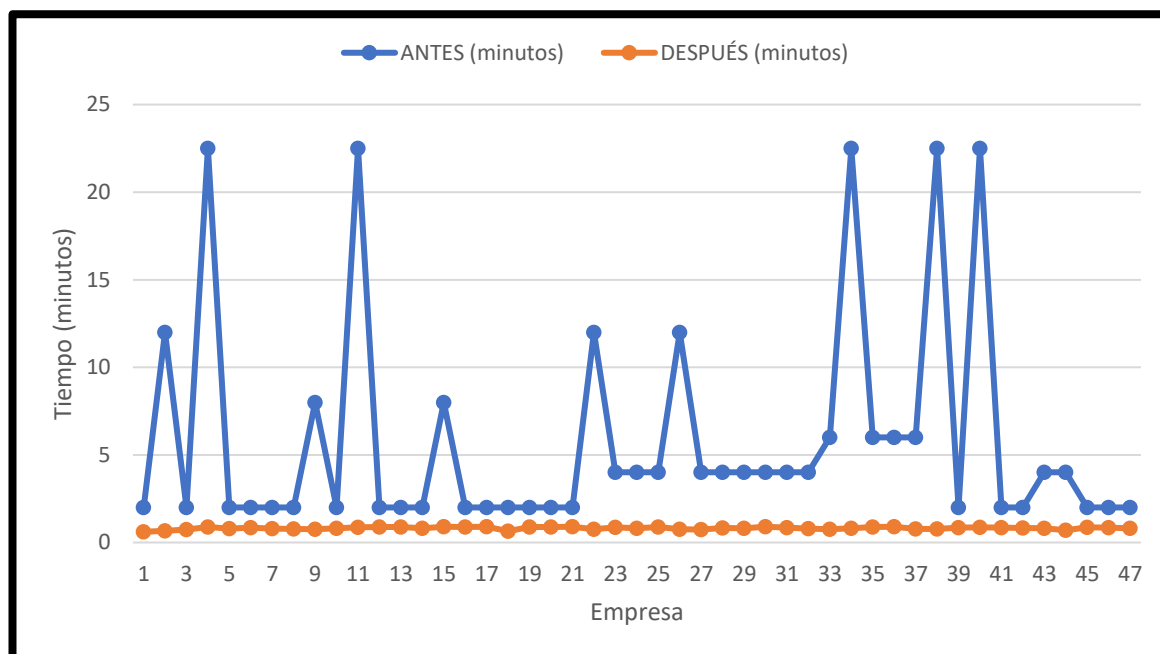


Figura 7 — Tiempos de proceso de facturación antes y después

Sobre la variable Proceso de venta, de la cual se tienen los datos del pretest y postest en el Anexo G, se muestra en la Figura 8 la representación de los tiempos del proceso de venta antes y después del Sistema de información de servicios compartidos multiplataforma. En esta figura se puede observar que los tiempos de proceso de venta antes fueron superiores a los tiempos de proceso de venta después de la aplicación del Sistema de información de servicios compartidos multiplataforma.

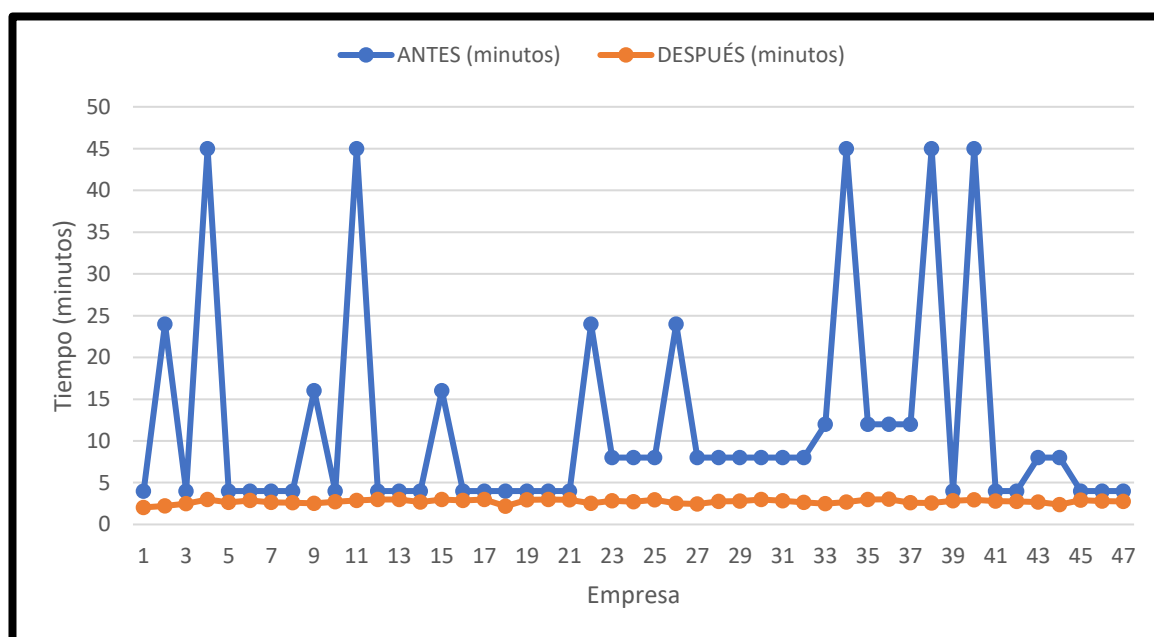


Figura 8 — Tiempos de proceso de venta antes y después

El Anexo G, muestra que el tiempo promedio de la variable proceso de venta antes de la implementación del sistema fue 11.8 minutos y el promedio después de la implementación del sistema fue de 2.73 minutos. Esto representa una reducción del 76.8% en los tiempos del proceso de venta.

5.1.1 Descripción del sistema

a) *Análisis del sistema*

i. *Especificación de requisitos del sistema*

La especificación de requisitos se centró en la necesidad de un sistema de servicios compartidos multiplataforma capaz de mejorar el tiempo de proceso de venta al adaptarse a múltiples rubros de negocio. Estos requisitos guiaron la arquitectura de desarrollo de un solo código base para tres modos operativos (minimarket, restaurante, hotel).

ii. *Requisitos funcionales (RF)*

Los requisitos funcionales Tabla 8, detallan el comportamiento específico y las funcionalidades que el sistema debe ejecutar.

Tabla 8 — Requisitos funcionales

Código	Descripción	Justificación estratégica
RF01	Gestión transaccional POS unificada: El sistema debe registrar ventas, devoluciones y arqueos de caja, asegurando la integridad de datos a través de los tres modos de operación (minimarket, restaurante, hotel).	Asegura la eficiencia del proceso de venta en todos los rubros, utilizando una lógica de <i>backend</i> centralizada (servicios compartidos).
RF02	Adaptación automática de interfaz (Modos): El <i>frontend</i> debe cambiar dinámicamente su interfaz (mesas, código de barras, registro de huéspedes) al seleccionar el modo de negocio (restaurante, minimarket, hotel).	Es el requisito clave que soporta el concepto de servicios compartidos multiplataforma, optimizando el tiempo de venta al proveer la UI exacta para cada rubro.
RF03	Emisión de documentos electrónicos: El sistema debe generar documentos fiscales conforme a la normativa, independientemente del modo de venta utilizado.	Garantiza el cumplimiento legal y reduce el tiempo de post-venta, centralizando la lógica de facturación electrónica.
RF04	Gestión de servicios compartidos (API central): Las operaciones de <i>stock</i> , clientes y reportes deben consumir una API central, compartida por los tres modos de interfaz.	Fundamento de los servicios compartidos. Evita la duplicación de lógica y acelera la actualización y mantenimiento del sistema.

RF05	Control de stock y costos multi-modo: El sistema debe gestionar entradas, salidas y transferencias de productos, aplicando descuentos o costos específicos según el modo activo (ej. ingredientes en restaurante, ítems unitarios en minimarket).	Provee información de gestión en tiempo real, crucial para la toma de decisiones y la eficiencia operativa.
------	---	---

iii. Requisitos no funcionales (RNF)

Los requisitos no funcionales definen los criterios de calidad que el sistema debe satisfacer, esenciales para la escalabilidad y la confianza del usuario Tabla 9.

Tabla 9 — Requisitos no funcionales

Código	Descripción	Justificación Estratégica
RNF01	Rendimiento de transacción crítica: El tiempo de respuesta de la API central (registro de venta/pedido) no debe exceder los 400 ms bajo concurrencia, con un énfasis en el modo restaurante (mayor concurrencia).	Objetivo primario de la tesis: Mejorar el tiempo de proceso de venta. Un <i>backend</i> rápido es indispensable.
RNF02	Multiplataforma nativa: El sistema debe operar con un rendimiento nativo en Android, Windows y Web a partir de una única base de código Flutter.	Fundamento de la tesis: Asegurar el desarrollo multiplataforma para abarcar la infraestructura existente de las 47 empresas.
RNF03	Escalabilidad de infraestructura: El <i>backend</i> debe ser capaz de soportar la adición de futuros modos de negocio (ej. Farmacia) sin reescribir la lógica de servicios compartidos.	Asegura la viabilidad a largo plazo y justifica la elección de arquitecturas desacopladas (APIs).
RNF04	Consistencia y sincronización: Utilizar WebSocket para garantizar la sincronización en tiempo real (ej. estado de mesas en restaurante o actualizaciones de <i>stock</i>).	Mejora la eficiencia operativa al reducir errores por información desactualizada, impactando positivamente el tiempo de proceso de venta.

iv. Casos de uso

Luego del análisis, se presentan los casos de uso principales del sistema.

Tabla 10 — Casos de uso principales

ID	Nombre del caso de Uso	Actor principal	Frecuencia	Tecnologías	Módulo
CU-001	Procesamiento automático de facturación electrónica	Sistema (Scheduler)	Diaria 23:00	Node.js, Bull Queue, Greenter, MySQL	api_ninja_node
CU-002	Generación automática de reportes de ventas	Sistema (Scheduler)	Diaria 00:00	Node.js, Puppeteer, ExcelJS, WebSocket	api_ninja_node
CU-003	Respaldo automático de bases de datos	Sistema (Scheduler)	Diaria 01:00	Node.js, mysqldump, mongodump, cron	api_ninja_node
CU-004	Gestión de inventario en tiempo real	Usuario (Vendedor/Admin)	Según necesidad	Flutter, Node.js, WebSocket, MySQL	rumi-sof_ui
CU-005	Proceso de venta completo	Usuario (Vendedor)	Por cada venta	Flutter, Node.js, esc_pos_printer, MySQL	rumi-sof_ui
CU-006	Consulta de estado de comprobantes SUNAT	Usuario (Contador/Admin)	Según necesidad	Flutter, PHP/Symfony, Greenter, SUNAT	api_facturacion_rs
CU-007	Generación de notas de crédito	Usuario (Administrador)	Según necesidad	Flutter, Laravel, PHP/Symfony, Greenter	api_client_rs
CU-008	Migración automática de datos	Sistema (Scheduler)	Semanal Dom 02:00	Node.js, MySQL, MongoDB, cron	api_ninja_node
CU-009	Gestión multi-empresa	Usuario (Super Admin)	Según necesidad	Flutter, Laravel, PHP/Symfony, Node.js	api_client_rs
CU-010	Generación de reportes personalizados	Usuario (Gerente/Admin)	Según necesidad	Flutter, Node.js, Puppeteer, ExcelJS	rumi-sof_ui

v. Historias de usuario

Tabla 11 — Historias de usuario

ID	Como	Quiero	Para	Prioridad	Estado	Módulo
HU-001	Administrador	Que el sistema envíe automáticamente las facturas a SUNAT	No tener que hacerlo manualmente	Crítica	Completada	api_ninja_node

HU-002	Gerente	Recibir reportes diarios de ventas automáticamente	Tomar decisiones informadas	Alta	Completada	api_ninja_no de
HU-003	Administrador	Que el sistema haga respaldos automáticos	Garantizar la seguridad de los datos	Crítica	Completada	api_ninja_no de
HU-004	Usuario	Acceder a la aplicación desde cualquier dispositivo	Trabajar con flexibilidad	Alta	Completada	rumi-sof_ui
HU-005	Vendedor	Generar facturas rápidamente	Atender más clientes en menos tiempo	Alta	Completada	rumi-sof_ui
HU-006	Contador	Consultar el estado de los comprobantes enviados	Verificar cumplimiento tributario	Media	Completada	api_facturacion_rs
HU-007	Gerente	Ver reportes de inventario en tiempo real	Controlar stock y evitar faltantes	Alta	Completada	rumi-sof_ui
HU-008	Administrador	Configurar múltiples empresas en el sistema	Manejar varios clientes	Media	Completada	api_client_rs
HU-009	Vendedor	Imprimir tickets de venta automáticamente	Agilizar el proceso de venta	Alta	Completada	rumi-sof_ui
HU-010	Administrador	Gestionar inventario desde la aplicación móvil	Controlar stock desde cualquier lugar	Alta	Completada	rumi-sof_ui
HU-011	Contador	Generar notas de crédito para anular facturas	Cumplir con normativas tributarias	Alta	Completada	api_client_rs
HU-012	Gerente	Ver reportes de pagos en tiempo real	Controlar flujo de caja	Alta	Completada	rumi-sof_ui
HU-013	Administrador	Sincronizar datos entre múltiples sucursales	Mantener consistencia de información	Alta	Completada	api_ninja_no de
HU-014	Vendedor	Consultar productos por código de barras	Agilizar búsqueda de productos	Media	Completada	rumi-sof_ui
HU-015	Administrador	Exportar reportes a Excel y PDF	Compartir información con stakeholders	Media	Completada	rumi-sof_ui
HU-016	Gerente	Recibir notificaciones de stock bajo	Evitar faltantes de productos	Alta	Completada	api_ninja_no de

HU-017	Contador	Consultar historial de comprobantes enviados	Auditoría y control tributario	Media	Completada	api_facturacion_rs
HU-018	Administrador	Configurar diferentes tipos de documentos	Adaptar a diferentes tipos de negocio	Media	Completada	api_client_rs
HU-019	Vendedor	Procesar ventas sin conexión a internet	Continuar operaciones durante cortes	Alta	Completada	rumisof_ui
HU-020	Gerente	Ver dashboard con métricas en tiempo real	Monitorear rendimiento del negocio	Alta	Completada	rumisof_ui

vi. Diagramas de secuencia de los procesos optimizados con el sistema

Proceso optimizado en el módulo restaurantes

La implementación del Sistema Rumisof-POS redefine el proceso operativo de venta, resolviendo de manera estructural los cuellos de botella del modelo manual. La solución se basa en la digitalización, la automatización y la centralización de datos, logrando una reducción del 86% en el tiempo total del ciclo (Figura 9).

Las optimizaciones críticas se resumen en:

1. Digitalización inicial y toma de órdenes: El mozo inicia el proceso mediante la selección digital de mesas en el sistema, actualizando su estado en tiempo real (30 segundos), eliminando las anotaciones manuales. La toma de pedidos se realiza mediante la selección electrónica de productos con capacidad de personalizar atributos específicos (picante, término, etc.). El sistema registra y almacena la orden con todos sus detalles en la base de datos, reduciendo el tiempo de toma de órdenes de 5 minutos a 2 minutos y erradicando el riesgo de errores de transcripción.
2. Distribución inteligente y simultánea: La distribución de la orden se transforma con la impresión inteligente por áreas. El sistema procesa automáticamente la orden y envía simultáneamente comandos específicos a cada estación de preparación (cocina, bar, etc.), reduciendo el tiempo de distribución física de 10 minutos a 30 segundos. Esta funcionalidad elimina la dependencia del mozo para la comunicación física, asegurando que cada área reciba únicamente su porción correspondiente.

3. Generación automatizada de precuenta: El sistema Rumisof-POS automatiza la generación de la precuenta. Al solicitarla, el sistema la calcula instantáneamente basándose en los datos centralizados, eliminando por completo los cálculos manuales y disminuyendo el tiempo de 8 minutos a 30 segundos.
4. Identificación rápida y eliminación de la digitación manual: El proceso de pago se optimiza mediante la identificación automática de la orden (30 segundos) a través del código de mesa. El cajero accede instantáneamente a la información completa de la orden en el sistema, eliminando la necesidad de buscar notas y la digitación manual de datos de clientes y comprobantes.

Como resultado de estas integraciones, el tiempo total de la fase 1 del proceso de venta se reduce drásticamente de 40 minutos a 5.5 minutos. Esta eficiencia no solo elimina los errores humanos recurrentes, sino que también mejora significativamente la experiencia del personal y la rotación de mesas, estableciendo un proceso de gestión de pedidos robusto y escalable.



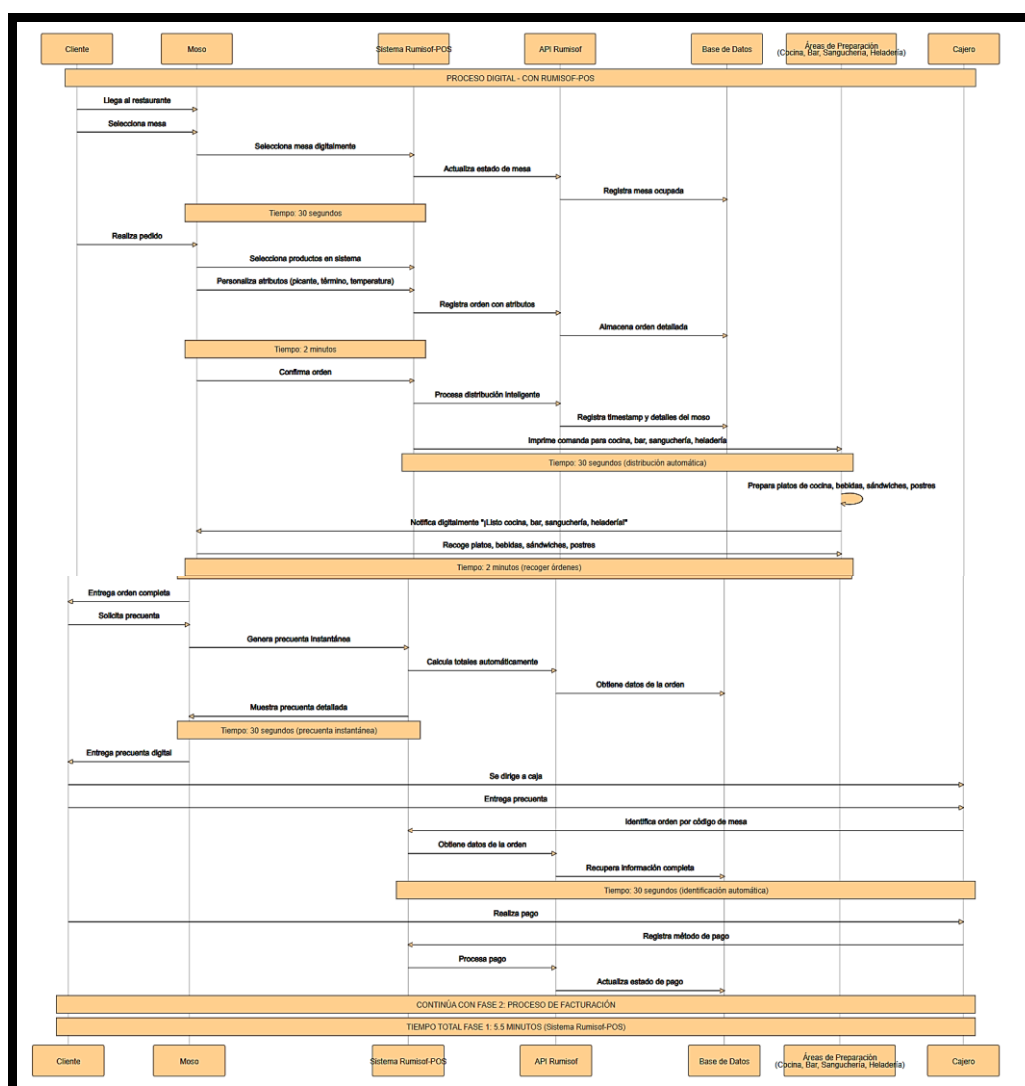


Figura 9 — Diagrama de secuencia del proceso de venta en módulo restaurante

Proceso optimizado en el módulo minimarket

La Figura 10 ilustra la transformación del proceso de venta en el módulo minimarket mediante la implementación del Rumisof-POS, resolviendo los cuellos de botella de la manera manual. La solución se basa en la automatización del registro, los cálculos y la identificación, logrando una reducción del tiempo total de 18 minutos a 3 minutos (una optimización del 83%).

Las innovaciones clave son:

1. Escaneo automático y consulta instantánea (5 segundos/producto): El cajero inicia la venta utilizando un scanner físico que envía el código de barras al sistema. El sistema consulta inmediatamente la base de datos y muestra la descripción, precio unitario y stock, reduciendo el tiempo de

verificación de productos de 5 minutos a un proceso de escaneo casi instantáneo (5 segundos por producto). Esto elimina el riesgo de errores de tarificación manual.

2. Cálculo de precios totalmente automatizado: El sistema Rumisof-POS se encarga de calcular automáticamente subtotales, aplicar el IGV y determinar el total general (30 segundos). Esta automatización elimina completamente la necesidad de cálculos manuales, disminuyendo el tiempo de 3 minutos a 30 segundos y erradicando los errores aritméticos.
3. Búsqueda de cliente automatizada e integrada (10 segundos): La identificación de clientes para comprobantes se optimiza mediante la búsqueda automática por RUC/DNI en la base de datos. El sistema accede y muestra la información del cliente, reduciendo el tiempo de búsqueda de 2 minutos a solo 10 segundos. Esto agiliza la atención y elimina los errores de digitación de datos.

La integración de estas funcionalidades digitales permite que la Fase 1 del proceso de venta se complete en un tiempo total de 3 minutos. Esta eficiencia mejora significativamente el flujo de caja, la experiencia del cliente y la precisión de los registros de inventario.

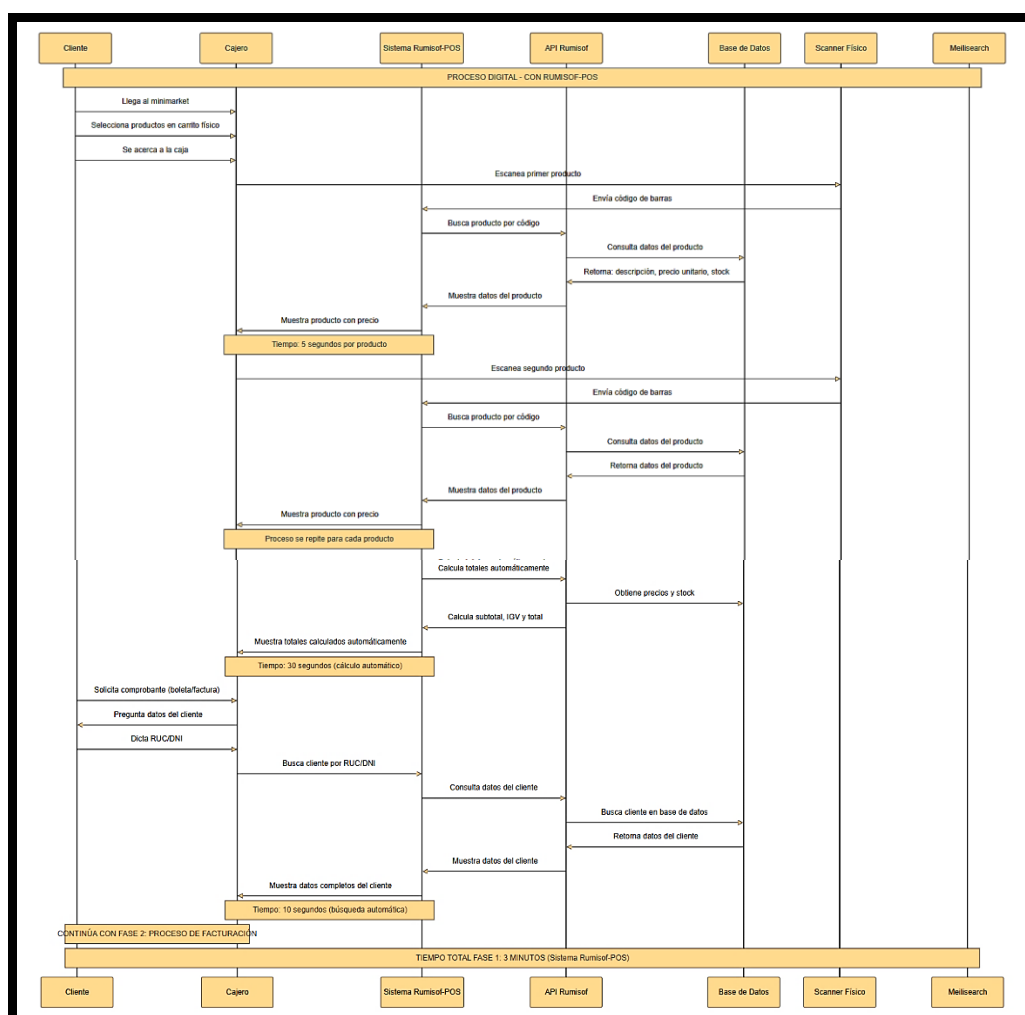


Figura 10 — Diagrama de secuencia del proceso de venta en módulo minimarket

Proceso optimizado en el módulo hotel

La Figura 11 detalla la implementación del Rumisof-POS en el módulo hotel, logrando un *check-in* completamente digital. La automatización elimina los 25 minutos del proceso manual y lo reduce a solo 2.5 minutos (una eficiencia del 90%), enfocándose en la velocidad y la precisión de los datos.

Las optimizaciones críticas se resumen en:

1. Visualización y selección automática de habitación (30 segundos): El sistema muestra al recepcionista la disponibilidad de habitaciones en tiempo real. La selección reduce el tiempo de cálculo y asignación manual de 5 minutos a 30 segundos, actualizando automáticamente el estado de ocupación.
2. Verificación instantánea y registro de huésped (30 segundos): El sistema automatiza la verificación de edad y la consulta de la base de datos de

clientes frecuentes por DNI/RUC. Esto agiliza el registro, reduciendo el proceso de 8 minutos a solo 30 segundos y eliminando los errores de transcripción de datos.

3. Registro de pago y cálculo de estancia automatizado (30 segundos): El sistema registra el pago y aplica automáticamente las reglas de estancia y tarifas, calculando la hora de salida sin intervención manual. Este paso reduce la tarea de 10 minutos a 30 segundos, eliminando cálculos propensos a errores.
4. Generación inmediata de comprobantes: El sistema genera y almacena automáticamente el JSON del comprobante electrónico, permitiendo su entrega inmediata al huésped. Esto elimina la generación manual y el registro en libros.

La integración de estas funcionalidades digitales permite que la fase 1 de *check-in* se complete en un tiempo total de 2.5 minutos, transformando la experiencia del huésped y garantizando registros exactos de la ocupación hotelera.

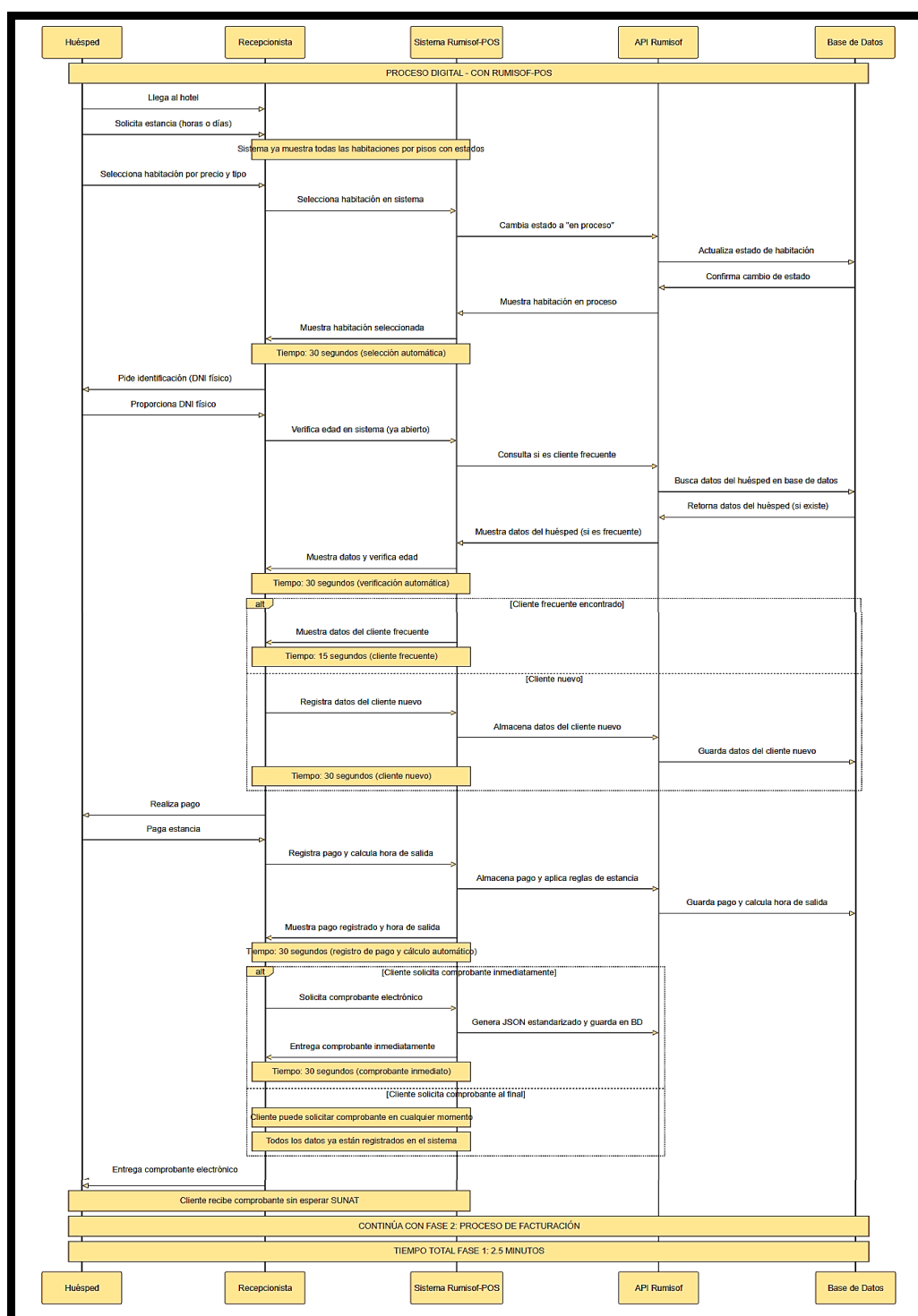


Figura 11 — Diagrama de secuencia del proceso de *check-in* en módulo hotel

Proceso optimizado de facturación electrónica

La Figura 12 detalla la implementación del proceso de facturación electrónica dentro del sistema Rumisof-POS, una funcionalidad transversal que resuelve el alto riesgo fiscal y el cuello de botella administrativo del proceso manual. La solución digital reduce el tiempo total de emisión y validación de 16 minutos a solo 1 minuto, logrando una optimización del 94%.

El proceso se basa en dos pilares: la inmediatez en el punto de venta y la validación asíncrona:

1. Emisión inmediata del comprobante (30 segundos): Tras la transacción, el sistema automáticamente envía los datos a la API, que genera un JSON estandarizado y lo almacena en la base de datos. Esta acción permite la entrega inmediata del comprobante electrónico al cliente (30 segundos), eliminando la espera por la validación de SUNAT y el riesgo de errores de digitación.
2. Validación asíncrona y conformidad tributaria: El proceso de validación con SUNAT se ejecuta automáticamente en segundo plano (par), sin requerir intervención del cajero/recepcionista. El API Task (Scheduler) evalúa los comprobantes pendientes, envía el documento a la API Facturación, la cual genera el XML UBL 2.1, realiza la firma digital y lo remite a SUNAT. La constancia de recepción (CDR) es almacenada automáticamente en la base de datos para su actualización.

Esta arquitectura asíncrona garantiza que la inmediatez del servicio no esté comprometida por los tiempos de respuesta de la SUNAT. El sistema Rumi-sof-POS asegura la conformidad tributaria mediante la validación automática, eliminando los errores humanos y la carga administrativa en el punto de venta.

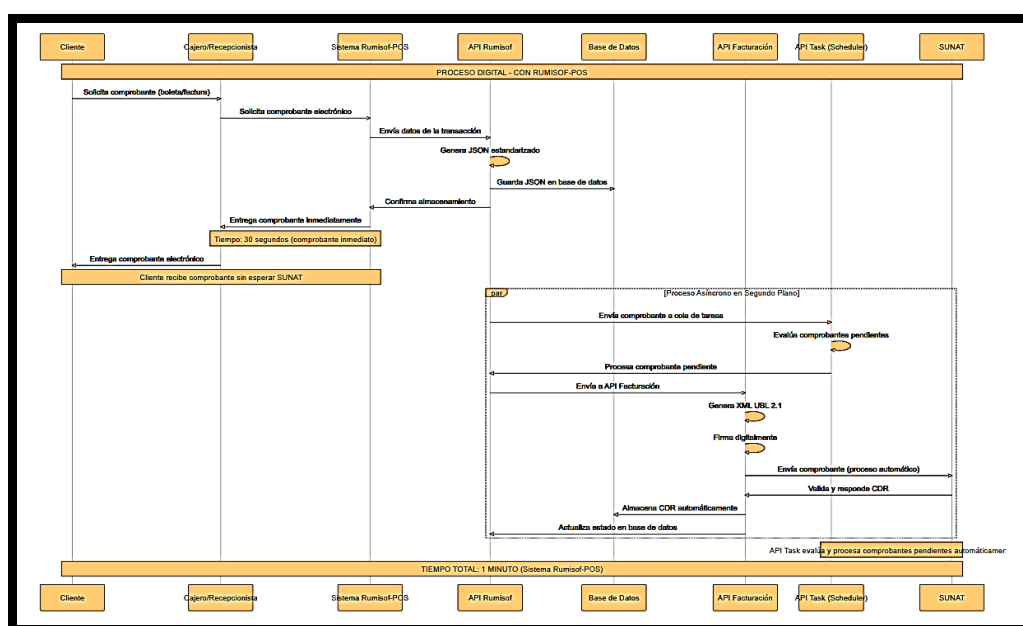


Figura 12 — Diagrama de secuencia del proceso de facturación electrónica

b) Metáfora del sistema

Para comprender mejor la arquitectura del "Sistema de información de servicios compartidos multiplataforma", se establece una metáfora con un ecosistema de servicios organizado que atiende a diversos tipos de clientes mediante una estructura eficiente y especializada, orientada a optimizar los tiempos de proceso en las operaciones de venta.

Componentes de la metáfora

Clientes multiplataforma: Establecimientos comerciales diversificados

Los establecimientos representados en el sistema - minimarket, restaurante, hotel y farmacia - corresponden a empresas usuarias de Rumisof-POS que requieren agilizar sus procesos de venta. Cada negocio posee características operativas particulares, pero comparten la necesidad de ejecutar transacciones rápidas y eficientes.

Servidor Ubuntu: Infraestructura central

El servidor Ubuntu constituye la base tecnológica que soporta toda la operación, funcionando como la columna vertebral del sistema que permite la integración y coordinación de todos los componentes.

Balanceador de carga Nginx: Coordinador de solicitudes

Este componente opera como distribuidor inteligente de cargas de trabajo, optimizando el flujo de transacciones mediante:

- Distribución equilibrada de solicitudes entre módulos disponibles
- Direccionamiento estratégico hacia servicios apropiados
- Prevención de congestiones en puntos críticos
- Garantía de respuestas oportunas

Módulos Api-rumisof: Unidades especializadas de procesamiento

Los múltiples módulos API-RUMISOF representan capacidades de procesamiento especializado que funcionan de manera paralela, ofreciendo:

- Atención simultánea de múltiples transacciones
- Especialización funcional por tipo de operación
- Escalabilidad adaptable según demanda operativa

- Redundancia operacional para continuidad del servicio
- Módulos especializados: Soporte funcional

Api-task: Gestor de procesos programados coordina operaciones automatizadas y tareas recurrentes en segundo plano.

Api-facturación: Administrador documental gestiona la emisión y control de comprobantes electrónicos, preparándolos para su validación oficial.

SUNAT: Entidad validadora oficial

SUNAT funciona como autoridad tributaria validadora que:

- Recibe comprobantes electrónicos para validación legal
- Verifica cumplimiento de requisitos tributarios
- Proporciona respuestas oficiales sobre estados documentales
- Emite constancias y códigos de autorización

Flujo operativo del sistema

1. El proceso de venta sigue una secuencia integrada:
2. Recepción de solicitudes mediante balanceador de carga
3. Direccionamiento a módulos especializados de procesamiento
4. Ejecución de transacciones comerciales
5. Generación de comprobantes electrónicos
6. Transmisión a SUNAT para validación oficial
7. Recepción y registro de respuestas
8. Finalización del ciclo transaccional

Ventajas de la arquitectura propuesta

La estructura implementada ofrece:

- Optimización de tiempos de respuesta transaccional
- Escalabilidad adaptativa según demanda operativa
- Especialización funcional para eficiencia procesal
- Resiliencia operacional mediante redundancia

- Cumplimiento normativo automatizado
- Comunicación eficiente con entidades regulatorias
- Validación inmediata de comprobantes electrónicos

Conclusión metafórica

La metáfora presentada demuestra cómo el sistema multiplataforma organiza sus componentes de manera coordinada y especializada, logrando el objetivo de reducir significativamente los tiempos de proceso en operaciones de venta, mientras mantiene comunicación efectiva con SUNAT para la validación oficial de comprobantes electrónicos.

En la Figura 13 se muestra la representación gráfica de la infraestructura multiplataforma. La imagen detalla el flujo de datos desde los puntos de venta de las empresas usuarias, pasando por la capa de balanceo y distribución (Nginx), hasta los servicios de aplicación (APIs) alojados en el servidor Ubuntu. Se enfatiza la especialización de los módulos (Procesamiento, Tareas Programadas y Facturación) y la integración crítica con el servicio tributario (SUNAT) para la validación legal de transacciones, fundamentales para reducir el tiempo total del proceso de venta.

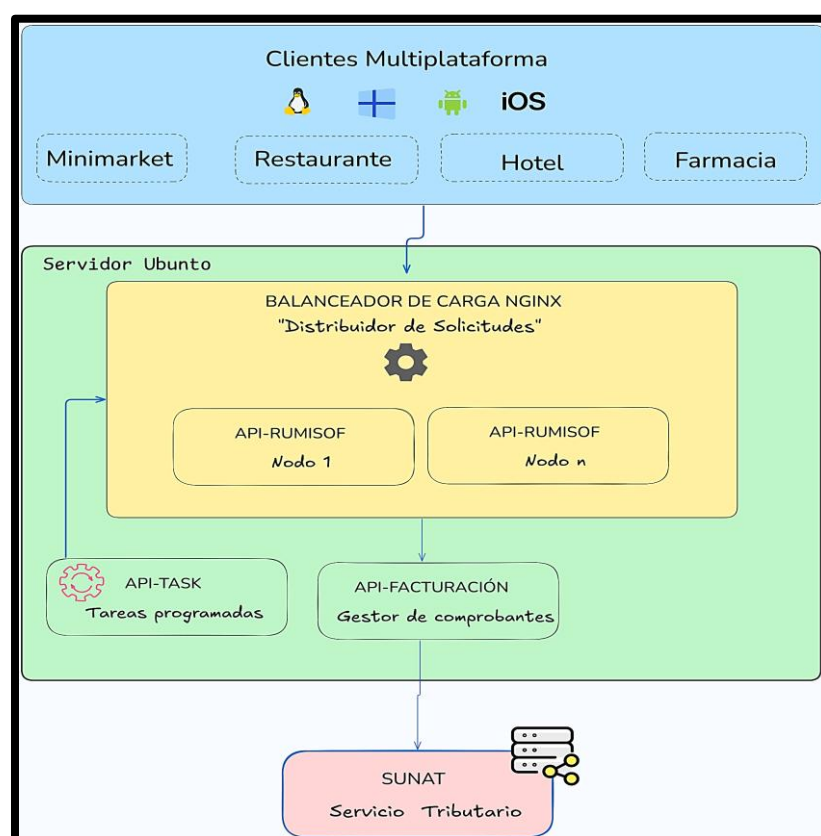


Figura 13 — Diagrama de flujo y componentes del sistema

c) *Arquitectura del sistema*

El diseño del sistema se fundamenta en una arquitectura distribuida basada en microservicios, complementada con estrategias avanzadas de persistencia polígota y multi-Tenancy. Esta decisión arquitectónica responde a la necesidad crítica de desacoplar las responsabilidades funcionales del negocio gestión comercial, facturación electrónica y procesamiento asíncrono para permitir una escalabilidad granular, un despliegue independiente y una alta tolerancia a fallos, superando las limitaciones de escalabilidad vertical inherentes a los sistemas monolíticos tradicionales.

El sistema se rige por los principios de alta cohesión y bajo acoplamiento, donde cada microservicio encapsula un dominio de negocio específico (bounded context) y expone sus funcionalidades únicamente a través de interfaces estandarizadas, ocultando sus detalles de implementación interna.

Fundamentación del estilo arquitectónico

La arquitectura adopta un enfoque de programación polígota, seleccionando la pila tecnológica más eficiente para cada tipo de carga de trabajo, optimizando así el consumo de recursos y el rendimiento global:

1. Procesamiento transaccional (Laravel/PHP): Implementado en el módulo `api_client_rs`. Se seleccionó por su robustez en el manejo de estructuras de datos relacionales complejas y su ecosistema maduro (Eloquent ORM) para la gestión de reglas de negocio críticas.
2. Procesamiento normativo y XML (Symfony/PHP): Utilizado en `api_facturacion_rs` debido a su estricto apego a estándares empresariales y su capacidad superior para la manipulación de esquemas XML/SOAP requeridos por la normativa SUNAT UBL 2.1.
3. Procesamiento asíncrono (Node.js/Express): Implementado en `api_ninja_node` para aprovechar el modelo de I/O no bloqueante basado en eventos (Event Loop), ideal para la orquestación de tareas concurrentes y monitoreo en tiempo real sin penalizar el rendimiento del CPU.
4. Interfaz de usuario (Flutter/Dart): Permite la compilación nativa multiplataforma (ARM/x86), garantizando un rendimiento en tiempo de ejecución superior a las soluciones híbridas y una experiencia de usuario consistente en entornos móviles y de escritorio.



Arquitectura de despliegue e infraestructura

La topología física del sistema se orquesta mediante un servidor centralizado que implementa los patrones de API Gateway y balanceo de carga, actuando como intermediario entre los clientes externos y la red interna de microservicios.

Componentes de infraestructura

- API Gateway (NGINX): Funciona como el único punto de entrada al sistema (Reverse Proxy). Es responsable de la terminación SSL/TLS, el enrutamiento de peticiones basado en URIs y la distribución inteligente de la carga de tráfico hacia las instancias disponibles de los microservicios.
- Cluster de escalabilidad: El servicio crítico de ventas (api_client_rs) se despliega en múltiples nodos (réplicas) para soportar la concurrencia de usuarios. El balanceador distribuye las peticiones utilizando un algoritmo de Least Connections, asegurando la disponibilidad del servicio ante picos de demanda.
- Estrategia de persistencia distribuida:
 - Cluster MySQL (Database per Tenant): Implementa el aislamiento físico y lógico de los datos. Cada empresa cliente opera sobre una base de datos independiente, lo que garantiza la seguridad de la información y evita el problema del "vecino ruidoso" (Noisy Neighbor), donde un cliente con alto tráfico afecta el rendimiento de otros.
 - Motor de búsqueda (Meilisearch): Almacén de datos NoSQL optimizado para lecturas. Se implementa un patrón CQRS simplificado (Command Query Responsibility Segregation), donde las escrituras se dirigen a la base de datos relacional (MySQL) para garantizar integridad ACID, mientras que las lecturas intensivas se sirven desde Meilisearch, logrando tiempos de respuesta sub-milisegundo.

Diagrama de arquitectura general

La Figura 14, ilustra la interacción entre los componentes, el flujo de datos y la segregación de responsabilidades en la infraestructura del servidor.



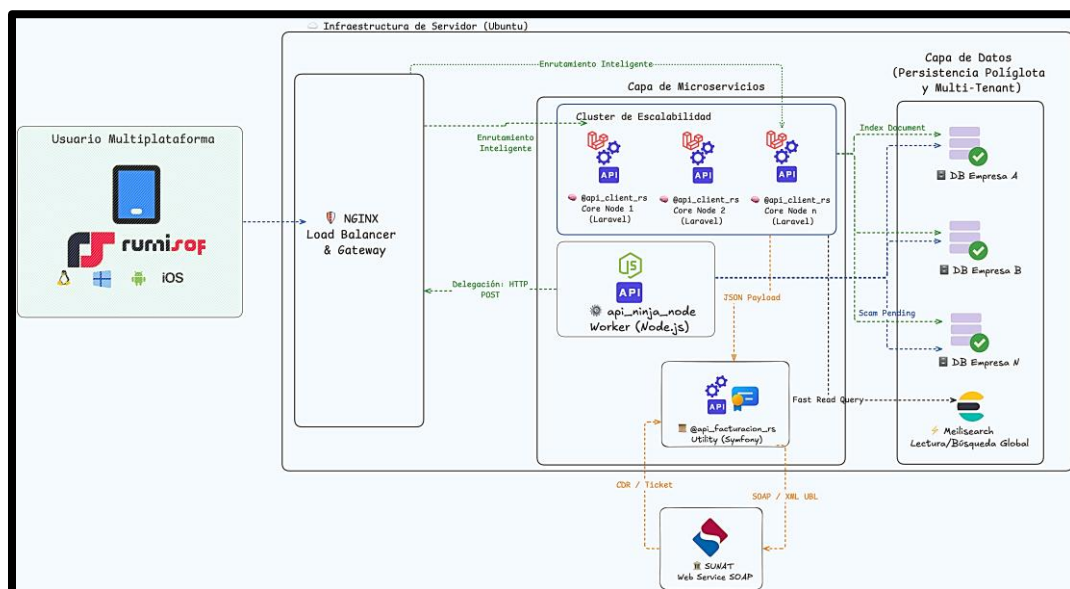


Figura 14 — Diagrama de arquitectura general del sistema

Diseño detallado de componentes

A continuación, se describe la arquitectura interna, los patrones de diseño aplicados y la estructura lógica de cada uno de los módulos que componen el sistema.

Módulo core de negocio (api_client_rs)

Este microservicio constituye el núcleo transaccional del sistema. Su diseño se basa en el patrón arquitectónico MVC (Modelo-Vista-Controlador), enriquecido con una arquitectura de capas de servicios y middleware especializado para la resolución de la multitenencia.

Patrones y decisiones de diseño:

1. **Middleware de resolución de Tenant:** Se implementó un componente de middleware personalizado (ConexionDinamica) que intercepta cada solicitud HTTP entrante. Este componente extrae el identificador del inquilino (Tenant ID) y reconfigura la conexión a la base de datos en tiempo de ejecución. Esto permite que una única instancia de código sirva a múltiples empresas manteniendo un aislamiento estricto de datos.
2. **Patrón Repository:** Se utiliza para abstraer la lógica de acceso a datos, permitiendo que la lógica de negocio en los controladores permanezca agnóstica respecto a la implementación de la persistencia, facilitando futuras migraciones o pruebas unitarias.

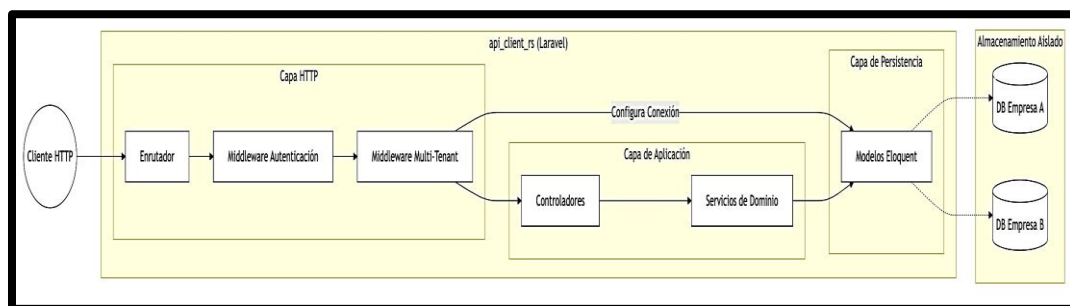


Figura 15 — Arquitectura interna del módulo Core

Módulo de facturación (api_facturacion_rs)

Este módulo opera como un servicio utilitario stateless (sin estado), diseñado específicamente para encapsular la complejidad de la facturación electrónica. Su arquitectura implementa patrones de diseño comportamentales y creacionales para manejar la variabilidad de los documentos tributarios.

Patrones y decisiones de diseño:

1. Strategy Pattern (Patrón estrategia): Dada la diversidad de documentos (facturas, boletas, notas de crédito), se implementó el patrón Strategy. Esto permite encapsular el algoritmo de generación de XML de cada documento en una clase separada, haciendo que sean intercambiables en tiempo de ejecución según la solicitud del cliente.
2. Factory Pattern (Patrón fábrica): Utilizado para la instanciación dinámica de la configuración de firma digital (SeeFactory). Dependiendo del RUC de la empresa emisora, la fábrica construye el objeto de configuración con los certificados digitales correspondientes, asegurando la validez de la firma.

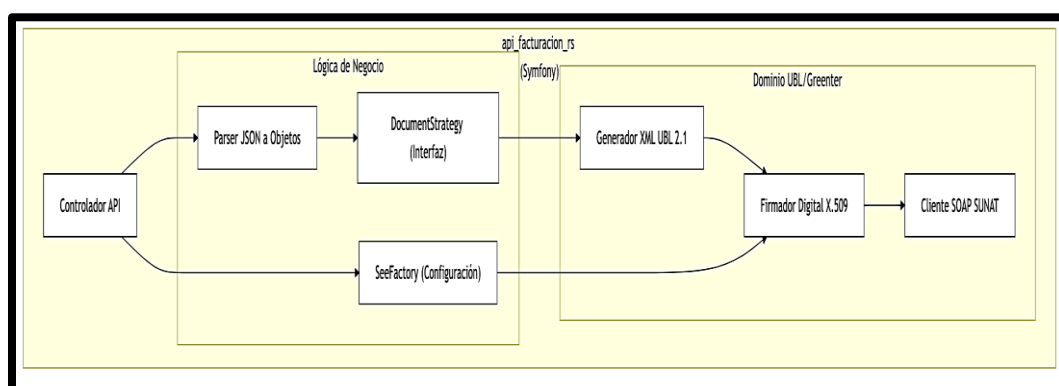


Figura 16 — Arquitectura interna del módulo de facturación

Cliente multiplataforma (rumisof_ui)

La aplicación cliente está construida sobre Flutter utilizando una arquitectura en capas (Layered Architecture) combinada con el patrón de gestión de estado BLoC (Business Logic Component).

Componentes arquitectónicos:

1. Capa de presentación (UI): Compuesta por Widgets pasivos que renderizan el estado actual y capturan eventos del usuario.
2. Capa de lógica de negocio (BLoC): Actúa como puente entre la UI y los datos, recibiendo eventos y emitiendo nuevos estados tras procesar la lógica.
3. Capa de datos: Implementa el patrón Repositorio para decidir la fuente de los datos (local o remota), permitiendo el funcionamiento offline-first.

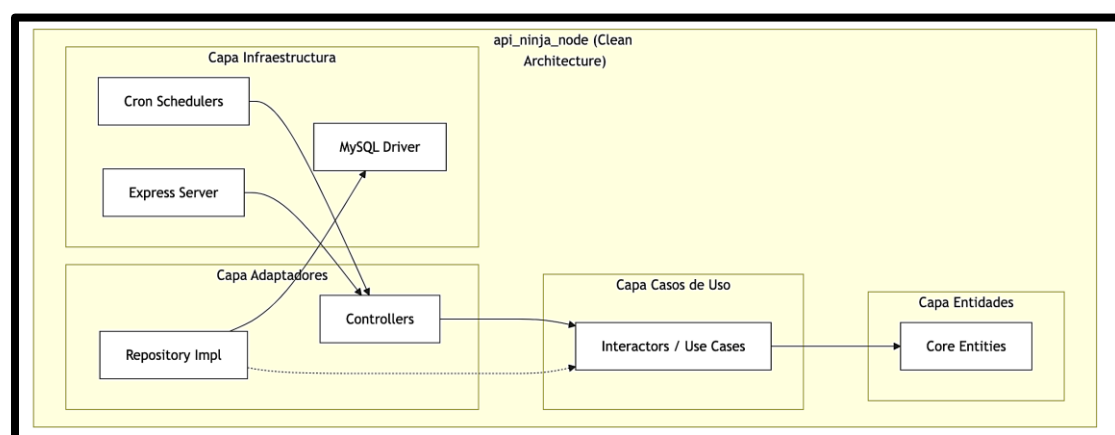


Figura 17 — Arquitectura interna del módulo de orquestación

d) Diseño de la base de datos

La base de datos implementada utiliza MariaDB 10.6.22 como motor de base de datos, siguiendo los principios de normalización relacional y garantizando la integridad de datos mediante constraints y foreign keys.

Tabla 12 — Especificaciones técnicas de la BD

Característica	Especificación
Sistema gestor de bases de datos (DBMS)	MariaDB 10.6.22
Juego de caracteres (Charset)	utf8mb3
Colación (Collation)	utf8mb3_general_ci
Motor de almacenamiento (Storage Engine)	InnoDB
Modelo de datos	Relacional
Nivel de normalización aplicado	Tercera Forma Normal (3NF)

La Figura 18 presenta la base de datos " Rumisof-POS ", que constituye el núcleo del sistema de información de servicios compartidos multiplataforma Rumisof-POS. Esta arquitectura de base de datos unificada está diseñada específicamente para soportar múltiples modelos de negocio de manera dinámica, permitiendo la operación simultánea o alternada entre minimarkets, restaurantes y hoteles mediante un mismo sistema centralizado. La flexibilidad del esquema relacional permite la adaptación en tiempo real entre los diferentes modos operativos, gestionando de forma integrada inventarios, transacciones, clientes y servicios específicos de cada rubro, lo que convierte al sistema en una solución versátil y escalable para empresas con operaciones diversificadas en el sector.

En la Tabla 13 se hace un resumen de las tablas principales del sistema.

Tabla 13 — Resumen tablas principales de la BD

Tabla	Total campos	Campos FK	Campos PK	Descripción
ticket	21	6	1	Tickets de venta principales
ticketlines	17	2	1	Líneas de detalle de tickets
product	22	6	1	Catálogo de productos
invoice	18	4	1	Facturas electrónicas
customer	17	0	1	Base de clientes
user	13	2	1	Usuarios del sistema
stock	6	2	1	Stock de productos
stockdiary	13	5	1	Historial de movimientos de stock
category	4	0	1	Categorías de productos
tax	6	0	1	Impuestos del sistema
location	15	0	1	Ubicaciones/sucursales
role	4	0	1	Roles de usuario
typepayment	5	0	1	Tipos de pago
closedcash	6	1	1	Sesiones de caja
stockdiary-reason	3	0	1	Razones de movimiento de stock

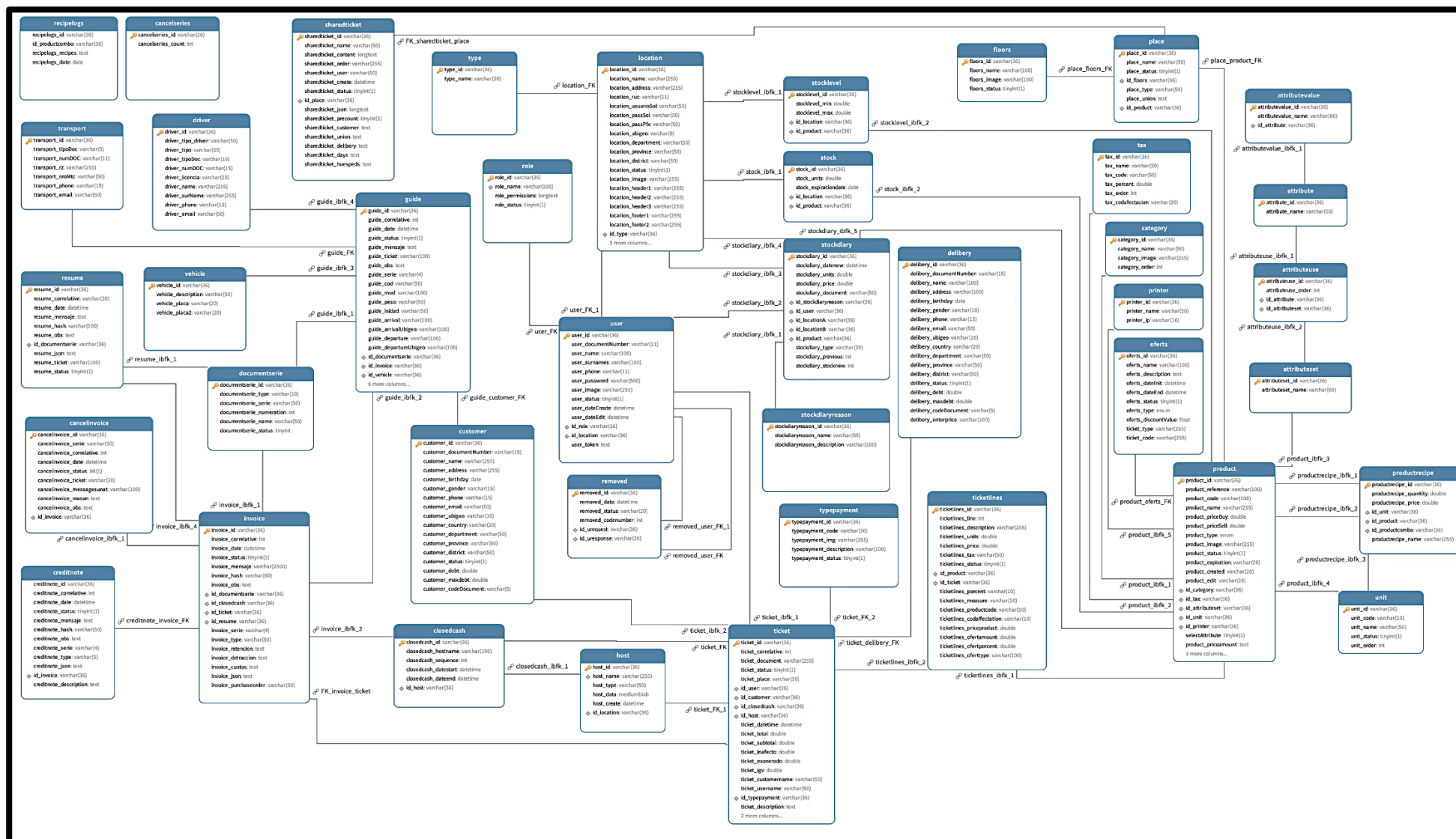


Figura 18 — Modelo E-R de la base de datos del sistema

e) Características del sistema

i. Facturación electrónica

Para cumplir con los requisitos de validación de documentos electrónicos ante la Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT), se diseñó e implementó un servicio especializado de facturación electrónica. Este módulo constituye el núcleo del sistema desarrollado, siendo responsable de gestionar integralmente el flujo completo de emisión de comprobantes electrónicos.

El servicio fue implementado bajo un enfoque de procesamiento por etapas, que comprende desde la generación estructurada del documento electrónico hasta su transmisión segura a los servidores de SUNAT. Cada etapa del proceso fue meticulosamente implementada siguiendo los lineamientos, especificaciones técnicas y protocolos establecidos en el Manual del Programador proporcionado por SUNAT, garantizando así la conformidad normativa y la interoperabilidad con los sistemas de la entidad tributaria.

Arquitectura del servicio

La implementación se estructuró en cuatro fases principales, como se observa en la Tabla 14.

Tabla 14 — Fases de implementación de la facturación electrónica

Fases	Tareas
Transformación de datos a estructura UBL	Conversión de datos desde formato JSON a XML UBL 2.1 Validación de esquemas y reglas de negocio Aplicación de templates según tipo de comprobante
Proceso de firma digital	Canonicalización XML Generación de hash mediante SHA-1 Firma con certificado digital X.509 Inserción de elementos de firma en el XML
Preparación para transmisión	Compresión del archivo XML firmado a formato ZIP Aplicación de naming conventions según SUNAT Validación final de integridad del paquete
Comunicación con WebServices SUNAT	Implementación de clientes SOAP con seguridad WS-Security Gestión de modos síncrono y asíncrono Procesamiento de respuestas y constancias

Base técnica y cumplimiento normativo

El desarrollo se sustentó en la documentación oficial de SUNAT, específicamente en:

- Manual del programador
- Esquemas XML UBL 2.1
- Especificaciones de WebServices
- Protocolos de seguridad y comunicación

Esta fundamentación técnica garantiza que el servicio genere documentos electrónicos válidos y aceptables por SUNAT, cumpliendo con todos los requisitos establecidos para la facturación electrónica en el Perú.

El siguiente código representa un ejemplo completo y válido de un archivo XML para una Boleta de Venta Electrónica, elaborado bajo el estándar UBL 2.1 y que cumple con todos los requisitos técnicos establecidos por SUNAT:

```
<?xml version="1.0" encoding="utf-8"?>
<Invoice xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2"
  xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:ext="urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2">
  <ext:UBLExtensions>
    <ext:UBLExtension>
      <ext:ExtensionContent>
        <ds:Signature Id="NinjaSign">
          <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
            <ds:Reference URI="">
              <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
              </ds:Transforms>
              <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
              <ds:DigestValue>+Lck19GCOJtkzWFGGq0REHQq1E0=</ds:DigestValue>
            </ds:Reference>
          </ds:SignedInfo>
          <ds:SignatureValue>BZL4JTaG-TQqQ+hZilNN++ONyiEg+2vdLoh9g4YCDCntPbkCg==</ds:SignatureValue>
        </ds:Signature>
      </ext:ExtensionContent>
    </ext:UBLExtension>
  </ext:UBLExtensions>
</Invoice>
```



```

<ds:X509Data>
    <ds:X509Certificate>MIIJYzCCB05DSUEXMzAx-
BgNVBAsMKklzc3VlZCBieSBBQyBDQU1FUkZJUK1BIFB-
FUsOaIxjySzw==</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</ext:ExtensionContent>
</ext:UBLExtension>
</ext:UBLExtensions>
<cbc:UBLVersionID>2.1</cbc:UBLVersionID>
<cbc:CustomizationID>2.0</cbc:CustomizationID>
<cbc:ID>BA01-1801</cbc:ID>
<cbc:IssueDate>2025-09-05</cbc:IssueDate>
<cbc:IssueTime>21:37:10</cbc:IssueTime>
<cbc:InvoiceTypeCode listID="0101">03</cbc:InvoiceTypeCode>
  <cbc:Note languageLocaleID="1000"><![CDATA[CATORCE CON 50/100
SOLES]]></cbc:Note>
<cbc:DocumentCurrencyCode>PEN</cbc:DocumentCurrencyCode>
<cac:Signature>
  <cbc:ID>20556693766</cbc:ID>
  <cac:SignatoryParty>
    <cac:PartyIdentification>
      <cbc:ID>20556693766</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name><![CDATA[BOTICA OKLIN]]></cbc:Name>
    </cac:PartyName>
  </cac:SignatoryParty>
  <cac:DigitalSignatureAttachment>
    <cac:ExternalReference>
      <cbc:URI>#Ninja-SIGN</cbc:URI>
    </cac:ExternalReference>
  </cac:DigitalSignatureAttachment>
</cac:Signature>
<cac:AccountingSupplierParty>
  <cac:Party>
    <cac:PartyIdentification>
      <cbc:ID schemeID="6">20556693766</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name><![CDATA[BOTICA OKLIN]]></cbc:Name>
    </cac:PartyName>
    <cac:PartyLegalEntity>
      <cbc:RegistrationName><![CDATA[BOTICA OKLIN]]></cbc:RegistrationName>
      <cac:RegistrationAddress>
        <cbc:ID>150101</cbc:ID>
        <cbc:AddressTypeCode>0000</cbc:AddressTypeCode>
        <cbc:CitySubdivisionName>-</cbc:CitySubdivisionName>
        <cbc:CityName>LIMA</cbc:CityName>
        <cbc:CountrySubentity>LIMA</cbc:CountrySubentity>
        <cbc:District>SAN LUIS</cbc:District>
      </cac:RegistrationAddress>
    </cac:PartyLegalEntity>
  </cac:Party>
</cac:AccountingSupplierParty>

```



```

        <cac:AddressLine>
          <cbc:Line><![CDATA[AV. NICOLAS AYLLON NRO. 1352 INT. H LIMA -
LIMA - SAN LUIS]]></cbc:Line>
        </cac:AddressLine>
        <cac:Country>
          <cbc:IdentificationCode>PE</cbc:IdentificationCode>
        </cac:Country>
        </cac:RegistrationAddress>
      </cac:PartyLegalEntity>
    </cac:Party>
  </cac:AccountingSupplierParty>
  <cac:AccountingCustomerParty>
    <cac:Party>
      <cac:PartyIdentification>
        <cbc:ID schemeID="1">00000000</cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyLegalEntity>
        <cbc:RegistrationName><![CDATA[USUARIOS VARIOS]]></cbc:Registration-
Name>
        <cac:RegistrationAddress>
          <cbc:ID>-</cbc:ID>
          <cac:AddressLine>
            <cbc:Line><![CDATA[-]]></cbc:Line>
          </cac:AddressLine>
          <cac:Country>
            <cbc:IdentificationCode>PE</cbc:IdentificationCode>
          </cac:Country>
          </cac:RegistrationAddress>
        </cac:PartyLegalEntity>
      </cac:Party>
    </cac:AccountingCustomerParty>
    <cac:PaymentTerms>
      <cbc:ID>FormaPago</cbc:ID>
      <cbc:PaymentMeansID>Contado</cbc:PaymentMeansID>
    </cac:PaymentTerms>
    <cac:TaxTotal>
      <cbc:TaxAmount currencyID="PEN">2.21</cbc:TaxAmount>
      <cac:TaxSubtotal>
        <cbc:TaxableAmount currencyID="PEN">12.29</cbc:TaxableAmount>
        <cbc:TaxAmount currencyID="PEN">2.21</cbc:TaxAmount>
        <cac:TaxCategory>
          <cac:TaxScheme>
            <cbc:ID>1000</cbc:ID>
            <cbc:Name>IGV</cbc:Name>
            <cbc:TaxTypeCode>VAT</cbc:TaxTypeCode>
          </cac:TaxScheme>
        </cac:TaxCategory>
      </cac:TaxSubtotal>
    </cac:TaxTotal>
    <cac:LegalMonetaryTotal>
      <cbc:LineExtensionAmount currencyID="PEN">12.29</cbc:LineExtensionAmount>
      <cbc:TaxInclusiveAmount currencyID="PEN">14.50</cbc:TaxInclusiveAmount>
    </cac:LegalMonetaryTotal>
  </cac:AccountingCustomerParty>
</cac:Party>

```



```

    <cbc:PayableAmount currencyID="PEN">14.50</cbc:PayableAmount>
  </cac:LegalMonetaryTotal>
  <cac:InvoiceLine>
    <cbc:ID>1</cbc:ID>
    <cbc:InvoicedQuantity unitCode="NIU">1</cbc:InvoicedQuantity>
    <cbc:LineExtensionAmount currencyID="PEN">0.85</cbc:LineExtensionAmount>
    <cac:PricingReference>
      <cac:AlternativeConditionPrice>
        <cbc:PriceAmount currencyID="PEN">1.00</cbc:PriceAmount>
        <cbc:PriceTypeCode>01</cbc:PriceTypeCode>
      </cac:AlternativeConditionPrice>
    </cac:PricingReference>
    <cac:TaxTotal>
      <cbc:TaxAmount currencyID="PEN">0.15</cbc:TaxAmount>
      <cac:TaxSubtotal>
        <cbc:TaxableAmount currencyID="PEN">0.85</cbc:TaxableAmount>
        <cbc:TaxAmount currencyID="PEN">0.15</cbc:TaxAmount>
        <cac:TaxCategory>
          <cbc:Percent>18</cbc:Percent>
          <cbc:TaxExemptionReasonCode>18</cbc:TaxExemptionReasonCode>
          <cac:TaxScheme>
            <cbc:ID>9996</cbc:ID>
            <cbc:Name>GRA</cbc:Name>
            <cbc:TaxTypeCode>FRE</cbc:TaxTypeCode>
          </cac:TaxScheme>
        </cac:TaxCategory>
      </cac:TaxSubtotal>
    </cac:TaxTotal>
    <cac:Item>
      <cbc:Description><![CDATA[SAN CARLOS AGUA]]></cbc:Description>
      <cac:SellersItemIdentification>
        <cbc:ID>COD</cbc:ID>
      </cac:SellersItemIdentification>
    </cac:Item>
    <cac:Price>
      <cbc:PriceAmount currencyID="PEN">0.8474576271</cbc:PriceAmount>
    </cac:Price>
  </cac:InvoiceLine>
</Invoice>

```

A continuación, se muestra el conjunto de archivos generados en el proceso optimizado de emisión electrónica: el archivo JSON con los datos de la transacción, el archivo XML en formato UBL 2.1 que se envía a SUNAT, y el archivo de respuesta R-*.xml (CDR) que confirma la recepción y validez del comprobante por parte de la administración tributaria.

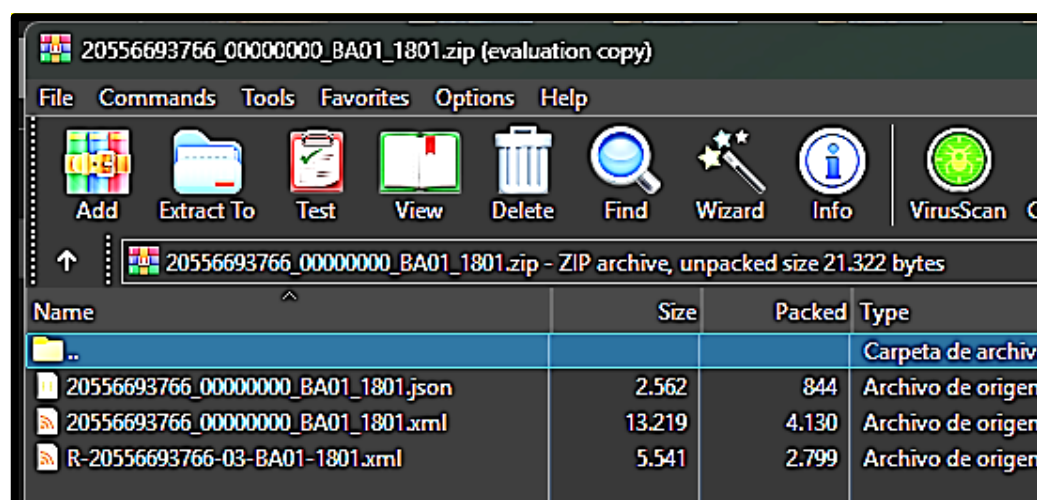


Figura 19 — XML firmado y constancia de recepción (CDR) de boleta electrónica

ii. Servicios compartidos

El sistema Rumisof-POS está diseñado para manejar múltiples empresas dentro de una misma plataforma mediante un único servicio de facturación electrónica compartido. La flexibilidad del sistema se basa en su capacidad para gestionar de manera eficiente los datos de clientes, facturación, series documentarias, y cancelaciones, manteniendo la integridad de la información para cada empresa. Esto se logra mediante una estructura de tablas bien organizada, donde cada entidad empresarial puede personalizar su proceso de venta y sus series documentarias.

La flexibilidad del sistema se logra compartiendo servicios, a través de tablas clave que permiten gestionar la facturación electrónica para varias empresas con un solo sistema, facilitando la automatización y reducción de errores en el proceso de venta. A continuación, se detallan las funciones principales de las tablas más relevantes:

– Tabla customers

La tabla customers permite almacenar la información de cada cliente, independiente del negocio que lo maneje. Esto garantiza que, aunque varias empresas utilicen el sistema, los datos de los clientes sean accesibles y puedan ser gestionados de manera centralizada.

Campos clave como `customer_documentnumber`, `customer_name`, y `customer_address` permiten la identificación única de los clientes, necesarios para la emisión de facturas electrónicas.

– Tabla documentseries

La tabla documentseries permite manejar las diferentes series documentarias que cada empresa puede utilizar para emitir sus facturas, boletas, y otros documentos tributarios. Esto permite que cada negocio tenga su propia numeración de facturas y boletas, asegurando el cumplimiento normativo y permitiendo la personalización según el tipo de documento fiscal requerido por SUNAT.

– Tabla invoices

La tabla invoices es el núcleo del sistema de facturación, almacenando cada factura emitida por cualquier empresa registrada en la plataforma.

Con campos como invoice_correlative, invoice_date, y invoice_status, el sistema asegura que cada transacción se registre de manera ordenada, permitiendo a cada negocio gestionar de forma independiente sus documentos tributarios sin interferir con otras empresas.

– Tabla cancelinvoices

La tabla cancelinvoices permite registrar las anulaciones o cancelaciones de facturas, garantizando que las empresas puedan gestionar las devoluciones y cancelaciones de manera formal. Esto es crucial para la flexibilidad del sistema, ya que cada empresa puede manejar la cancelación de sus propios documentos sin afectar a las otras empresas que utilizan el mismo sistema.

– Tabla tickets y ticketlines

Tickets y ticketlines permiten la gestión detallada de las ventas realizadas en puntos de venta, integrando múltiples métodos de pago y líneas de productos. La relación entre tickets y customers facilita la personalización del proceso de venta, permitiendo que se asocien ventas específicas a clientes individuales, lo cual es útil para las estrategias de fidelización y servicio post-venta.

– Tabla guides

La tabla guides gestiona la emisión de guías de remisión, permitiendo a las empresas llevar control sobre el transporte de bienes, lo cual es crucial para empresas que requieren transporte de productos. Este módulo es especialmente útil para sectores como el hotelero y de retail que necesiten documentar

la entrega de bienes, facilitando la interoperabilidad con los sistemas de logística.

– Tabla resumes

Resumes permite la generación de resúmenes de boletas de venta, una función esencial para la integración con SUNAT, garantizando que los documentos emitidos cumplan con la normativa de facturación electrónica. Esto asegura que el sistema cumpla con las normativas tributarias y sea capaz de integrarse con servicios gubernamentales para la validación de documentos.

El uso de un sistema multiempresa como Rumisof-POS permite a las empresas reducir significativamente el tiempo de procesamiento de ventas, al centralizar todas las operaciones relacionadas con la emisión de facturas y boletas en un solo servicio de facturación electrónica. Esto reduce la duplicación de procesos, mejora la eficiencia administrativa y garantiza el cumplimiento normativo. Además, la automatización del manejo de documentos tributarios, incluyendo las anulaciones y guías de remisión, agiliza el ciclo de ventas y mejora la experiencia tanto para los usuarios como para los clientes.

Dentro de las ventajas de un servicio compartido tenemos:

- Centralización de datos: Permite la administración de múltiples empresas dentro de un mismo entorno, reduciendo costos de infraestructura y simplificando la gestión.
- Interoperabilidad con SUNAT: Cumplimiento automático de la normativa de facturación electrónica y la posibilidad de gestionar diferentes tipos de documentos tributarios según el giro del negocio.
- Escalabilidad: La estructura de tablas permite escalar el sistema para agregar nuevas empresas sin modificar la estructura básica, lo que favorece la expansión del servicio.

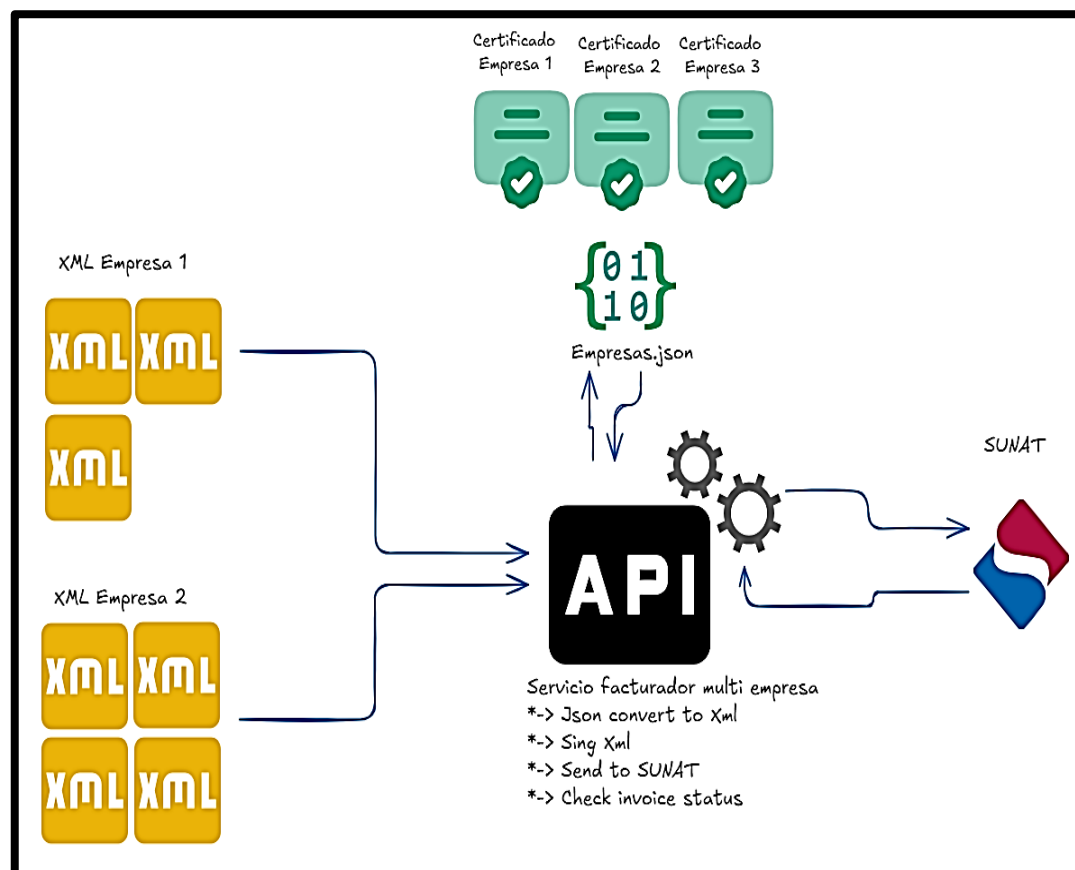


Figura 20 — Proceso de facturación

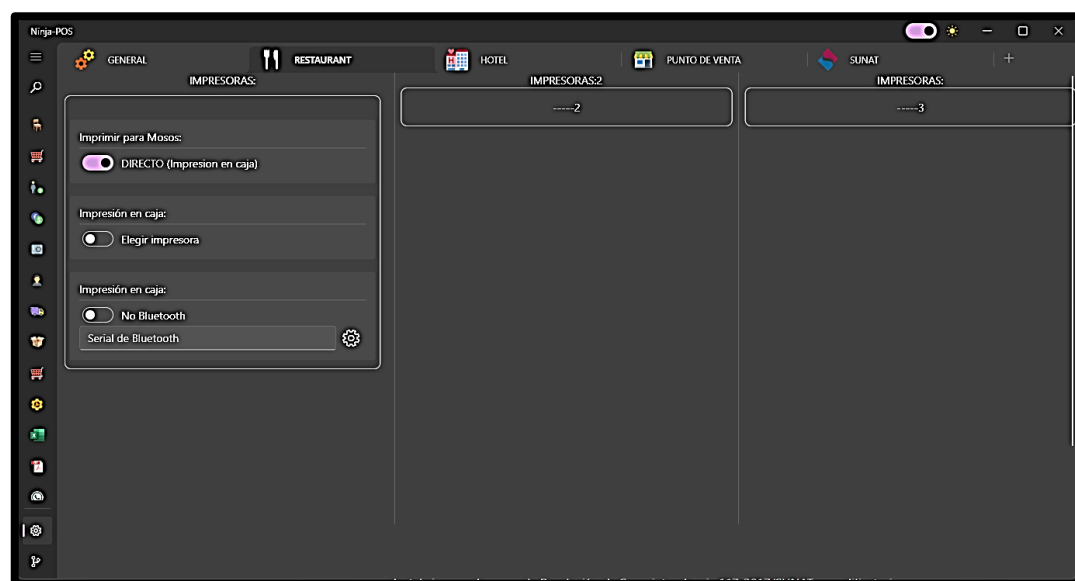


Figura 21 — Módulo de configuración del sistema según tipo de negocio

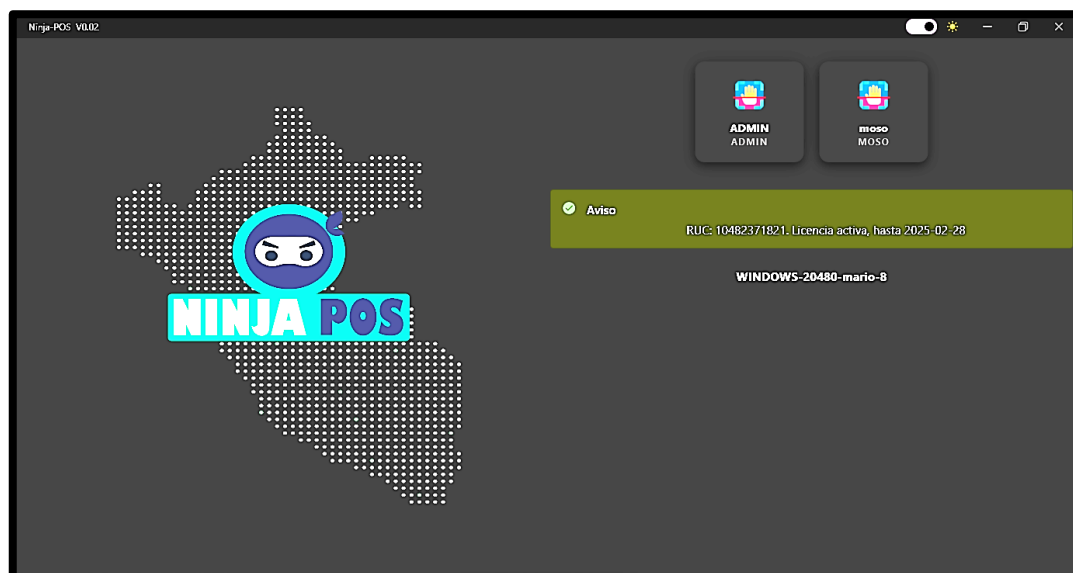


Figura 22 — Interfaz de login del sistema según perfil de usuario

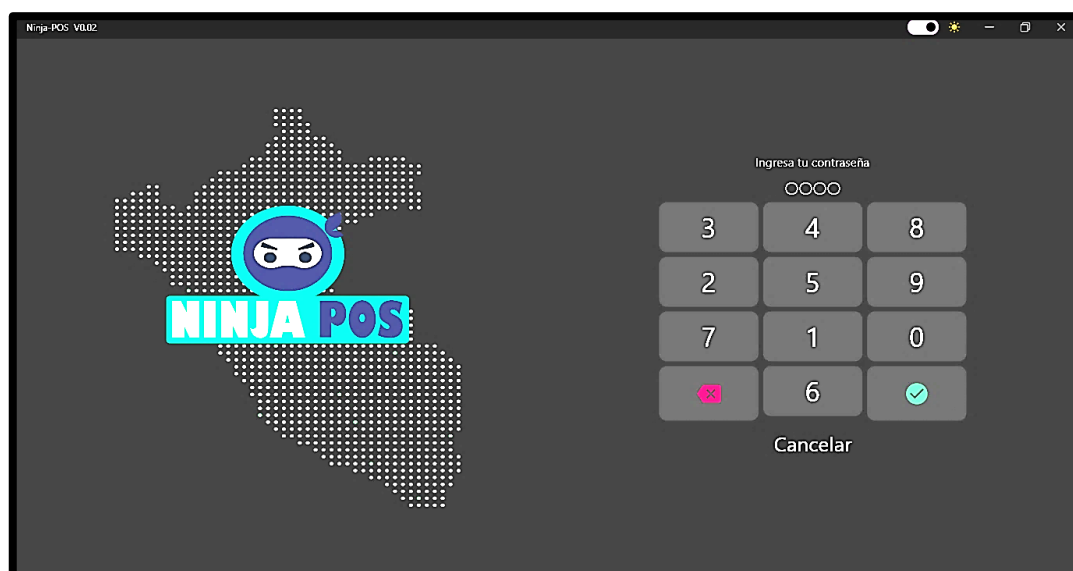


Figura 23 — Interfaz de ingreso de contraseña

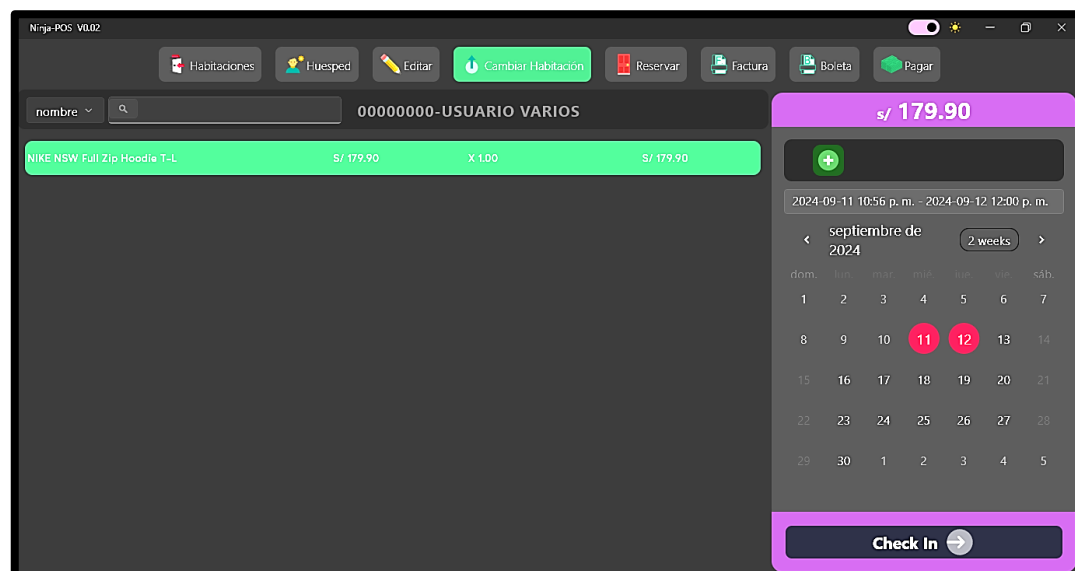


Figura 24 — Interfaz del módulo en modo hotel

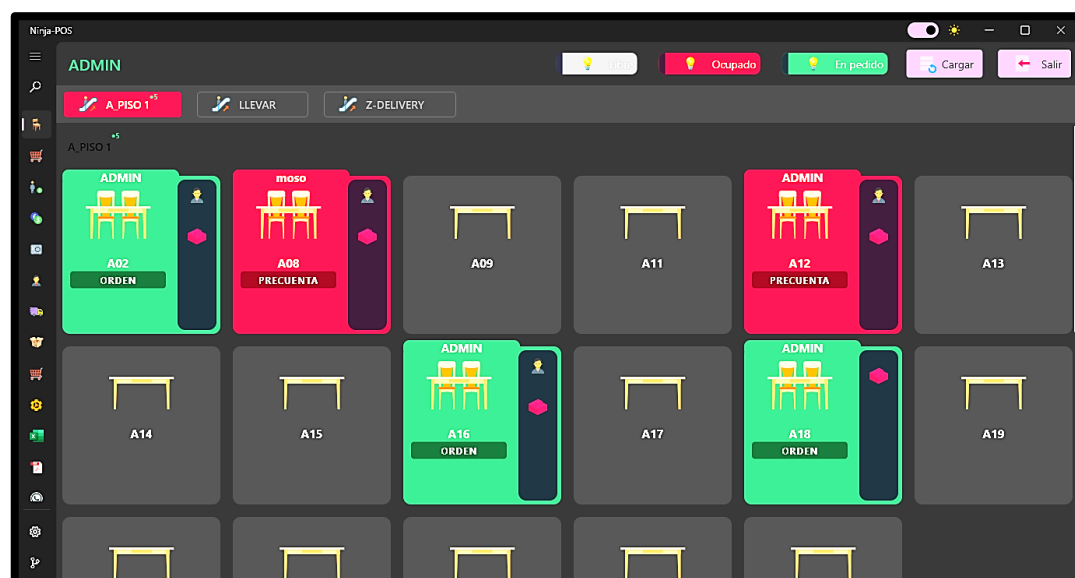


Figura 25 — Interfaz de administración del modo restaurant

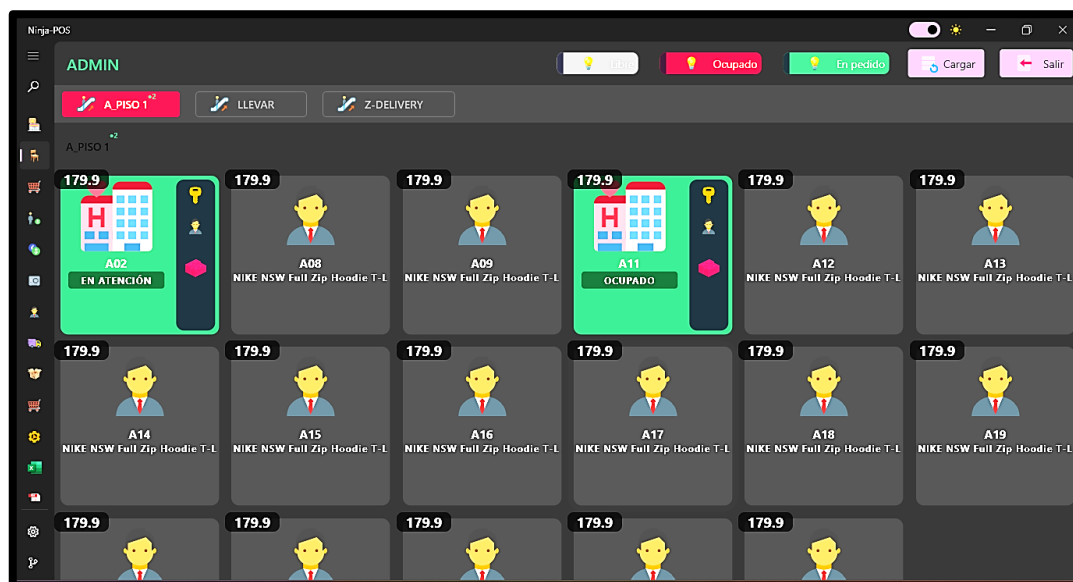


Figura 26 — Interfaz del modo hotel



Figura 27 — Interfaz del modo minimarket

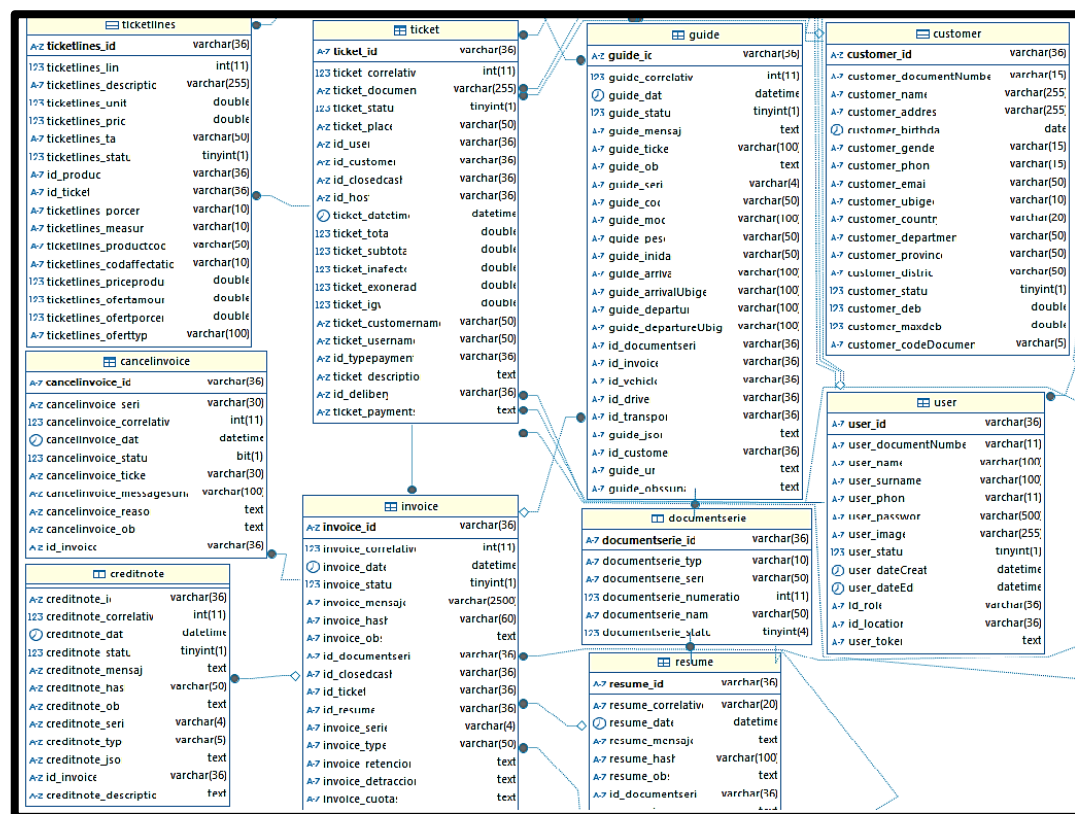


Figura 28 — Tablas que conforman específicamente la facturación electrónica

El sistema de facturación multiempresa fue diseñado para adaptarse a diferentes tipos de negocios, incluidos restaurantes, hoteles, minimarkets, mayoristas de productos de minimarkets, negocios de comida rápida, ferreterías y lubricentros. Para lograr esta adaptación, se realizaron modificaciones en los módulos del sistema, permitiendo que funcione de manera óptima en cada contexto. A continuación, se describen las adaptaciones realizadas para cada tipo de negocio y las funcionalidades compartidas que garantizan la versatilidad del sistema:

Funcionalidades comunes entre todos los negocios

Todos los tipos de negocios mencionados comparten ciertas funcionalidades claves que hacen que el sistema Rumisof-pos sea adaptable y eficiente. Estas funcionalidades incluyen:

- Búsqueda de clientes por DNI/RUC: Esta característica permite registrar y buscar fácilmente a los clientes, optimizando el proceso de facturación y evitando errores al capturar datos.

- Integración de múltiples métodos de pago: El sistema acepta diferentes formas de pago, como Yape, Plin, Visa, efectivo, entre otros, ofreciendo flexibilidad tanto al cliente como al negocio.
- Gestión de precios y stock: La capacidad de gestionar los precios de los productos y controlar el inventario es esencial en todos los tipos de negocio. El sistema permite actualizar precios en tiempo real y mantener el control de los niveles de stock.
- Facturación electrónica: El sistema permite la generación de facturas electrónicas, independientemente del tipo de negocio, cumpliendo con las normativas tributarias vigentes en el país.
- Historial de transacciones: Todos los tipos de negocios tienen acceso a un registro de ventas y transacciones, lo que permite generar informes y realizar auditorías de forma sencilla.

Modificaciones específicas por tipo de negocio

Restaurantes

- Módulo de secciones y mesas: Se implementó un sistema para la gestión de mesas y secciones (por ejemplo, terraza, salón, bar), permitiendo asignar estados a cada mesa (disponible, ocupada, en espera).
- Corte de cuenta: Los usuarios pueden solicitar cortes de cuenta parciales o totales, lo que es útil para mesas compartidas o clientes que desean dividir la cuenta.
- Mover y unir mesas: La flexibilidad para mover y unir mesas facilita la administración del espacio según el flujo de clientes.
- Orden en tiempo real: El sistema permite enviar órdenes a la cocina o barra en tiempo real, utilizando web sockets, lo que agiliza la preparación de los pedidos.
- Delivery: Se añadió un módulo para registrar pedidos por delivery, con la posibilidad de asignar repartidores y gestionar el proceso de entrega.

Hoteles

- Secciones como pisos y habitaciones: El sistema reutiliza la estructura de secciones del restaurante, pero adaptada para el manejo de pisos y habitaciones en un hotel. Las "mesas" del restaurante se transforman en habitaciones con su respectivo estado (ocupada, reservada, disponible).
- Reservas, check-in y check-out: Se añadieron funciones específicas para gestionar las reservas de habitaciones, así como el proceso de check-in y check-out.
- Registro de huéspedes: El sistema permite registrar a los huéspedes y mantener un listado actualizado, lo cual es esencial para la facturación y la gestión hotelera.
- Cambio de habitación: Si un cliente desea cambiar de habitación, el sistema permite gestionar este cambio sin afectar la facturación.

Minimarkets y mayoristas de productos de minimarket

- Búsqueda rápida de productos: Se optimizó el sistema para manejar grandes volúmenes de productos, permitiendo búsquedas rápidas por nombre, código o categoría.
- Precios mayoristas y minoristas: Los mayoristas de productos pueden manejar múltiples niveles de precios (al por mayor y al por menor), permitiendo ajustar los precios automáticamente según la cantidad de compra.
- Gestión eficiente de stock: Dado que manejan un gran volumen de productos, el sistema tiene funciones avanzadas para gestionar el inventario, emitir alertas de bajo stock y generar órdenes de compra automáticamente.

Negocios de comida rápida

- Pedido rápido y ágil: El sistema fue adaptado para registrar los pedidos de manera rápida y eficaz, priorizando la velocidad en la atención y la facturación, dada la naturaleza del negocio.
- Integración con plataformas de delivery: Se incluyó una funcionalidad para integrar el sistema con plataformas de delivery, lo que facilita la gestión de los pedidos externos.

Ferreterías y lubricentros

- Gestión de productos especializados: Se adaptaron módulos para manejar productos más específicos como materiales de construcción, herramientas o lubricantes.
- Facturación de servicios: Además de la venta de productos, el sistema permite facturar servicios adicionales (por ejemplo, en lubricentros, los cambios de aceite).
- Control de inventarios complejos: Dado el volumen de productos y las especificidades del sector, el sistema incluye funciones avanzadas para gestionar el inventario, con categorías detalladas para facilitar la búsqueda y administración de productos.

iii. API

A continuación, se describe detalladamente la implementación del API REST en Laravel, desarrollado para el sistema Rumisof-POS. Este sistema es utilizado para la gestión de productos, usuarios, mesas, habitaciones, facturación electrónica, envío de comprobantes y generación de reportes, optimizando el tiempo de venta en empresas de servicio como restaurantes, hoteles, minimarkets y más.

El sistema de información implementado es multiplataforma, basado en tecnologías web modernas como Laravel y ofrece servicios RESTful para facilitar la integración y gestión de procesos en las empresas de servicio. Incluye rutas, funciones y middleware implementado en la versión 1 del API.

Estructura general del API

- Versión del API: v1
- Tecnología base: Laravel
- Middleware:

conexion: Asegura la correcta conexión antes de procesar solicitudes.

jwt: Autentica las solicitudes a través de tokens JWT.

throttle: 1500,1: Limita las solicitudes a 1500 por minuto para evitar abusos.



Rutas del API

A continuación, se detalla cada una de las rutas implementadas en el API, organizadas por funcionalidad.

Gestión de autenticación y token

- Obtener token de usuario

GET /token/getToken

Obtiene un token de autenticación para el usuario.

- Validar token

GET /tokenvalidate/getToken

Valida la vigencia del token de autenticación.

Gestión de correo electrónico

- Obtener correos

POST /v1/getMail

Devuelve los correos recibidos en el sistema.

- Enviar correos

POST /v1/sendMail

Permite el envío de correos electrónicos desde el sistema.

Gestión de usuarios

- Obtener usuarios

POST /v1/user/getUsers

Devuelve una lista de los usuarios del sistema.

- Obtener todos los usuarios

POST /v1/user/getUsersAll

Devuelve una lista completa de todos los usuarios.

- Crear un nuevo usuario

POST /v1/user/createUser

Crea un nuevo usuario en el sistema.



- Actualizar usuario

POST /v1/user/updateUser/{user_id}

Actualiza la información de un usuario existente basado en su user_id.

- Eliminar usuario

POST /v1/user/deleteUser/{user_id}

Elimina un usuario específico del sistema.

Gestión de productos

- Obtener un producto

POST /v1/product/getProduct

Devuelve los detalles de un producto basado en su código o referencia.

- Crear un producto

POST /v1/product/createProduct

Registra un nuevo producto en la base de datos.

- Eliminar un producto

POST /v1/product/deleteProduct

Elimina un producto específico del sistema.

- Carga masiva de productos desde un archivo CSV

POST /v1/product/saveCSV

Permite la carga masiva de productos desde un CSV, optimizando el manejo de inventarios.

- Obtener atributos, categorías, impuestos, unidad y configuración de impresión

POST /v1/product/getCategoryAttributeTaxUnitPrint

Devuelve la configuración de atributos, categorías, impuestos y unidades de medida.

- Obtener unidades de medida

POST /v1/product/getUnit

Devuelve las unidades de medida registradas en el sistema.

Gestión de categorías

- Obtener categorías de productos

POST /v1/product/getCategory

Devuelve una lista de categorías de productos disponibles en el sistema.

Gestión de mesas y habitaciones

- Obtener pisos (habitaciones)

POST /v1/floors/getFloors

Devuelve los pisos y habitaciones de un hotel o establecimiento.

- Obtener mesas de un piso

POST /v1/floors_places/getFloors_Places

Devuelve las mesas asociadas a un piso del establecimiento.

- Obtener mesas ocupadas

POST /v1/floors_places/getFloors_PlacesFill

Devuelve las mesas que actualmente están ocupadas.

- Mover una mesa

POST /v1/floors_places/sendMovePlace

Cambia la ubicación de una mesa en el sistema.

- Obtener líneas de pedido de una mesa

POST /v1/floors_places/placeGetLines

Devuelve las líneas de pedido asociadas a una mesa específica.

Gestión de clientes

- Crear un cliente

POST /v1/customer/createCustomer

Registra un nuevo cliente en el sistema.



- Obtener lista de clientes

POST /v1/customer/getCustomers

Devuelve la lista de clientes registrados en el sistema.

- Actualizar información de un cliente

POST /v1/customer/updateCustomer

Actualiza los datos de un cliente existente.

Gestión de facturación electrónica

- Enviar factura a la SUNAT

POST /v1/invoice/sendInvoiceSUNAT

Envía una factura electrónica a la SUNAT para su validación y registro.

- Generar archivo XML de una factura

POST /v1/invoice/getXML

Genera y devuelve el archivo XML de una factura electrónica.

- Obtener estado de una factura enviada

POST /v1/invoice/getStatusFacturaSunat

Consulta el estado de una factura que ha sido enviada a la SUNAT.

- Generar resumen diario de facturas

POST /v1/invoice/createResumen

Crea un resumen diario de facturas electrónicas para su envío a la SUNAT.

- Obtener PDF de una factura en base64

POST /v1/invoice/getPdfBase64

Genera y devuelve el PDF de una factura en formato base64.

- Obtener correlativos de facturación

POST /v1/correlative/InvoiceFactura

Devuelve el correlativo actual para las facturas generadas.



Movimientos de caja

- Obtener estado de la caja cerrada

POST /v1/closedcash/getcash

Devuelve el estado de la caja cerrada actualmente.

- Crear una nueva caja

POST /v1/closedcash/createcash/{host_id}

Crea una nueva caja asociada a un host_id.

- Registrar nuevo movimiento de caja

POST /v1/closedcash/newcash

Registra un nuevo movimiento de caja.

- Obtener detalles de la caja

POST /v1/closedcash/detail

Devuelve un reporte detallado del estado actual de la caja.

- Generar reporte completo de caja

POST /v1/closedcash/reportCashAll

Genera un reporte de todas las transacciones registradas en la caja.

Gestión de tickets

- Crear ticket (comprobante de venta)

POST /v1/ticket/createTicket

Registra un nuevo ticket o comprobante de venta.

- Registrar movimiento de efectivo en un ticket

POST /v1/ticket/cashMovement

Registra un movimiento de efectivo asociado a un ticket.

- Obtener lista de movimientos de efectivo

POST /v1/ticket/cashMovementList

Devuelve una lista de todos los movimientos de efectivo registrados.

Gestión de delivery

- Crear registro de delivery

POST /v1/delivery/createDelivery

Crea un nuevo registro de entrega o delivery.

- Obtener lista de entregas

POST /v1/delivery/getDelivery

Devuelve una lista de todas las entregas registradas en el sistema.

- Actualizar registro de delivery

POST /v1/delivery/updateDelivery

Actualiza los datos de un registro de entrega existente.

Consultas de RUC y DNI

- Obtener datos por DNI

POST /v1/Document/Dni

Consulta y devuelve los datos de un cliente basado en su número de DNI.

- Obtener datos por RUC

POST /v1/Document/Ruc

Consulta y devuelve los datos de un cliente basado en su número de RUC.

Gestión de impresoras

- Obtener impresoras disponibles

POST /v1/printer/getPrinters

Devuelve una lista de impresoras disponibles en el sistema.

Otros servicios

- Crear nueva base de datos

POST /v1/rsDB



Crea una nueva base de datos en el sistema.

- Configurar tipo de aplicación

POST /v1/config/typeApp

Establece o consulta la configuración del tipo de aplicación.

iv. Web scraping

Se implementó un módulo de búsqueda automática de clientes mediante web scraping en la página: <https://e-consultaruc.sunat.gob.pe/cl-ti-itmr-consruc/FrameCriterioBusquedaWeb.jsp>

El objetivo principal de esta implementación es agilizar la búsqueda de clientes, ahorrar tiempo en el registro de datos y reducir errores humanos, lo que a su vez mejora la eficiencia del sistema y permite ahorrar costos relacionados con procesos manuales.

El sistema fue implementado utilizando Node.js con las bibliotecas Puppeteer para la automatización del navegador y Cheerio para el análisis del contenido de las páginas web.

El código de Web scraping para RUC (Node.js), se realizó en la página de la SUNAT utilizando Puppeteer para extraer datos de un RUC:

```

1  const puppeteer = require('puppeteer');
2  const fs = require('fs');
3
4  (async () => {
5    const browser = await puppeteer.launch({
6      headless: false,
7      slowMo: 100,
8      args: ['--no-sandbox', '--disable-dev-shm-usage']
9    });
10
11    const page = await browser.newPage();
12
13    // Navegar a la página
14    await page.goto('https://e-consultaruc.sunat.gob.pe/cl-ti-itmr-consruc/FrameCriterioBusquedaWeb.jsp', { waitUntil: 'networkidle0' });
15
16    // Esperar un poco
17    await new Promise(resolve => setTimeout(resolve, 2000));
18
19    // Ingresar el RUC y hacer clic en el botón
20    await page.type('#txtRuc', '10477931508');
21    await page.waitForSelector('#btnAceptar');
22    await page.click('#btnAceptar');
23
24    // Esperar a que la página cargue después del clic
25    await page.waitForNavigation({ waitUntil: 'networkidle0' });
26
27    // Extraer los datos de la página
28    const data = await page.evaluate(() => {
29      const resultado = {};
30
31      // Función para obtener texto basado en label
32      const getText = (label) => {
33        const element = Array.from(document.querySelectorAll('.list-group-item')).find(el => el.innerText.includes(label));
34        if (element) {
35          const valueElement = element.querySelector('p.list-group-item-text, h4 + div, h4.list-group-item-heading + div, .col-sm-7 h4.list-group-item-heading, .col-sm-7 p.list-group-item-text');
36          return valueElement ? valueElement.innerText.trim() : null;
37        }
38        return null;
39      };
40    });

```

```

1
2 // Extrayendo datos de la página
3 resultado["Número de RUC"] = getText('Número de RUC:');
4 resultado["Tipo Contribuyente"] = getText('Tipo Contribuyente:');
5 resultado["Tipo de Documento"] = getText('Tipo de Documento:');
6 resultado["Nombre Comercial"] = getText('Nombre Comercial:');
7 resultado["Fecha de Inscripción"] = getText('Fecha de Inscripción:');
8 resultado["Fecha de Inicio de Actividades"] = getText('Fecha de Inicio de Actividades:');
9 resultado["Estado del Contribuyente"] = getText('Estado del Contribuyente:');
10 resultado["Condición del Contribuyente"] = getText('Condición del Contribuyente:');
11 resultado["Domicilio Fiscal"] = getText('Domicilio Fiscal:');
12 resultado["Sistema Emisión de Comprobante"] = getText('Sistema Emisión de Comprobante:');
13 resultado["Actividad Comercio Exterior"] = getText('Actividad Comercio Exterior:');
14 resultado["Sistema Contabilidad"] = getText('Sistema Contabilidad:');
15
16 // Extraer información específica de tablas
17 const getTableData = (label) => {
18   const element = Array.from(document.querySelectorAll('.list-group-item')).find(el => el.innerText.includes(label));
19   if (element) {
20     const rows = Array.from(element.querySelectorAll('table tbody tr td'));
21     return rows.map(row => row.innerText.trim()).join(', ');
22   }
23   return null;
24 };
25
26 resultado["Actividad(es) Económica(s)"] = getTableData('Actividad(es) Económica(s):');
27 resultado["Comprobantes de Pago c/aut. de impresión"] = getTableData('Comprobantes de Pago c/aut. de impresión (F. 806 u 816):');
28 resultado["Sistema de Emisión Electrónica"] = getTableData('Sistema de Emisión Electrónica:');
29 resultado["Comprobantes Electrónicos"] = getText('Comprobantes Electrónicos:');
30 resultado["Afiliado al PLE desde"] = getText('Afiliado al PLE desde:');
31 resultado["Padrones"] = getTableData('Padrones:');
32
33 return resultado;
34 });
35
36 // Guardar los datos en un archivo JSON
37 fs.writeFileSync('resultado.json', JSON.stringify(data, null, 2));
38
39 // Tomar una captura de pantalla
40 await page.screenshot({ path: 'resultado.png' });
41
42 // Cerrar el navegador
43 await browser.close();
44
45 console.log('Datos guardados en resultado.json');
46 });
47

```

El scraping se utiliza para extraer información pública de la web de la SUNAT, como el número de RUC, razón social, estado y condición del contribuyente, y el domicilio fiscal. Estos datos se almacenan temporalmente en el sistema para su uso en la facturación y registro de clientes, pero no se distribuyen ni se almacenan permanentemente sin autorización.

Mejoras implementadas

- Eficiencia y ahorro de tiempo: El sistema automatiza la búsqueda de clientes, lo que reduce el tiempo de registro manual de información, especialmente en empresas con gran volumen de transacciones diarias.
- Ahorro de dinero: Al reducir los procesos manuales y los errores en la entrada de datos, el sistema ayuda a reducir costos operativos, optimizando el tiempo de los empleados y evitando multas por errores en la facturación.
- Validaciones previas: Se implementaron validaciones en el formato de DNI y RUC, asegurando que solo se realicen búsquedas con datos válidos.
- Control de errores: Se gestionan excepciones para manejar situaciones como páginas no disponibles o errores en la búsqueda, garantizando que el sistema siga funcionando en caso de fallos.

- Integración multiplataforma: Al estar implementado en Node.js, el sistema se integra fácilmente con otros módulos del sistema multiplataforma en Flutter, facilitando su instalación y funcionamiento en cualquier equipo, ya sea de escritorio o móvil.

Cumplimiento normativo

- La información utilizada en el sistema, como el RUC y otros datos tributarios, se extrae de la página web de la SUNAT, que ofrece acceso público a un padrón resumido de todos los contribuyentes. Este acceso público es parte del servicio que la SUNAT brinda para facilitar la consulta de información sobre contribuyentes y empresas, como el estado de su RUC, la razón social, y el domicilio fiscal.
- Es importante señalar que, si bien la página incluye un CAPTCHA, parece estar destinado principalmente a evitar consultas masivas no controladas, más que a restringir el acceso individual a esta información pública. Hasta la fecha, no se ha encontrado ningún apartado explícito de términos y condiciones en la página de consulta, lo que refuerza la interpretación de que el acceso a esta información está destinado al público en general.
- Sin embargo, eludir el CAPTCHA con el fin de automatizar estas consultas puede estar en contra de los principios éticos, aunque no esté claramente regulado en los términos del sitio web. Por lo tanto, para evitar cualquier conflicto, se recomienda siempre buscar métodos alternativos para la automatización de consultas, como el uso de APIs oficiales, en caso de que se ofrezcan.
- Finalmente, el sistema respeta la Ley de Protección de Datos Personales en Perú (Ley N° 29733), utilizando los datos extraídos de manera legal y exclusivamente para los fines legítimos de la facturación electrónica, asegurando además que estos datos estén protegidos mediante medidas de seguridad adecuadas.

v. *ETL*

El sistema implementado permite realizar una carga masiva de productos en la base de datos, facilitando la importación de inventarios desde otros sistemas o desde archivos CSV (Excel). A continuación, se destacan las funcionalidades más importantes del código:



Lectura del archivo CSV:

- El sistema utiliza la biblioteca csv-parser para leer archivos CSV. En este caso, se cargan inventarios desde un archivo llamado INVENTARIO_BO-TICA_ABANCAY.csv.
- Los datos se procesan de manera eficiente para su posterior inserción en la base de datos.
- El formato del CSV está delimitado por un separador específico (;).

Manejo de categorías:

- El sistema busca si la categoría del producto ya existe en la base de datos mediante el modelo Category.
- Si la categoría no existe, se crea una nueva entrada en la tabla de categorías, lo que asegura que los productos siempre estén correctamente categorizados.

Registro y actualización de productos:

- Para cada producto del CSV, el sistema verifica si ya existe un producto con el mismo código de barras (product_code) y nombre (product_name).
- Si el producto ya existe, se actualizan los precios de compra y venta.
- Si el producto no existe, se registra uno nuevo con toda la información relevante, como referencia, precios, stock, categoría, impuestos, y fechas de expiración.

Gestión del stock:

- Se verifica si el producto ya tiene un stock registrado en una ubicación específica (en este caso, con un ID de ubicación predefinido: id_location).
- Si el producto ya existe en esa ubicación, se actualizan las unidades de stock sumando las nuevas unidades del archivo CSV.
- Si no existe, se crea un nuevo registro de stock, asegurando que cada producto tenga una cantidad correcta en la ubicación especificada.

Diario de stock

- Para cada cambio de stock, el sistema registra una entrada en el StockDiary. Esto permite mantener un historial detallado de cada movimiento



de stock (entradas, salidas, ajustes), incluyendo la cantidad, el precio de compra y otros detalles importantes.

- Cada movimiento se registra con un identificador único (stockdiary_id), el motivo del movimiento (CE), el usuario que realizó la operación (id_user), y la ubicación de almacenamiento.

Contadores de productos registrados y actualizados:

- El sistema lleva un seguimiento de cuántos productos fueron registrados por primera vez y cuántos fueron actualizados en la base de datos, lo que permite tener una visión clara de las acciones tomadas durante la carga masiva.

Integración con Sequelize:

- El código utiliza Sequelize, un ORM (Object-Relational Mapping) para manejar las consultas e interacciones con la base de datos de manera eficiente.
- Las interacciones con las tablas de productos, categorías, stock y el diario de stock están bien estructuradas y permiten mantener integridad en los datos durante el proceso de carga.

```
1  /* Librerías */
2  ...
3
4  /* Función para importar datos */
5  async function importData(rows, counters) {
6    /* Buscar o crear categoría */
7    const category = await Category.findOne({ where: { category_name: rows[0].category } })
8    || await Category.create({ category_id: uuidv4(), category_name: rows[0].category });
9
10   /* Registrar o actualizar producto */
11   const productInstance = await Product.findOne({ where: { product_code: rows[0].codigo_barra } });
12   const product = productInstance ?
13     await productInstance.update({ product_priceBuy: rows[0].pricebuy, product_priceSell: rows[0].venta })
14     : await Product.create({ product_id: uuidv4(), product_code: rows[0].codigo_barra, ... });
15   counters[productInstance ? 'updated' : 'registered']++;
16
17   /* Manejo de stock */
18   const stockInstance = await Stock.findOne({ where: { id_product: product.product_id } });
19   stockInstance ?
20     await stockInstance.update({ stock_units: stockInstance.stock_units + parseInt(rows[0].stock, 10) })
21     : await Stock.create({ stock_id: uuidv4(), stock_units: rows[0].stock, id_product: product.product_id });
22
23   /* Registrar en StockDiary */
24   await StockDiary.create({ stockdiary_id: uuidv4(), stockdiary_units: rows[0].stock, ... });
25 }
26
27 /* Función para leer CSV */
28 async function importCSV() { ... }
29
30 /* Ejecutar la importación */
31 async function run() {
32   const csvData = await importCSV();
33   const counters = { updated: 0, registered: 0 };
34
35   for (const row of csvData) await importData([row], counters);
36
37   console.log(`Actualizados: ${counters.updated}, Registrados: ${counters.registered}`);
38 }
39
40 run();
41
```

vi. Sistema de WebSocket en tiempo real

El sistema implementa un sistema único de comunicación en tiempo real que permite la sincronización inmediata de datos entre todos los dispositivos conectados, proporcionando una experiencia de usuario fluida y actualizada.

Características específicas implementadas:

- Sincronización instantánea: Actualización inmediata de datos entre todos los dispositivos
- Comunicación bidireccional: Intercambio de información en tiempo real entre dispositivos
- Gestión de conexiones: Sistema robusto de manejo de conexiones y reconexión automática
- Optimización de ancho de banda: Transmisión eficiente de datos para minimizar el uso de recursos

vii. Sistema de adaptación automática por tipo de negocio

Un aspecto distintivo del sistema 'Rumisof-pos' es su capacidad única de adaptación automática que detecta el tipo de negocio del cliente y reconfigura completamente la interfaz de usuario, funcionalidades y flujos de trabajo según las necesidades específicas de cada sector. Esta característica permite que el sistema 'Rumisof-pos' sirva eficientemente a restaurantes, hoteles, minimarkets, ferreterías y otros tipos de negocios sin necesidad de desarrollos personalizados.

Características específicas implementadas en Rumisof-pos:

- Detección automática del tipo de negocio: El sistema 'Rumisof-pos' identifica automáticamente el sector del cliente durante la configuración inicial
- Reconfiguración dinámica de la interfaz: La aplicación 'Rumisof-pos' cambia completamente su apariencia y funcionalidades según el tipo de negocio
- Flujos de trabajo adaptativos: Los procesos de venta en 'Rumisof-pos' se adaptan automáticamente a las necesidades específicas de cada sector

- Módulos específicos por industria: Implementación de funcionalidades especializadas para cada tipo de negocio en el sistema 'Rumisof-pos'

viii. *Manejador de estados*

El desarrollo del frontend para el sistema Rumisof-pos se implementó utilizando Flutter, con el patrón BLoC (Business Logic Component) para manejar los estados de los widgets de manera eficiente. El patrón BLoC es ideal para separar la lógica de negocios del código de interfaz, promoviendo la reutilización del código y mejorando la escalabilidad de la aplicación.

Esta sección documenta en detalle cómo se ha implementado el patrón BLoC en la arquitectura de la aplicación, describiendo su estructura, manejo de eventos, cambios de estado, y la lógica para realizar operaciones específicas como la gestión de productos, impuestos, cálculo de totales, y la interacción con diferentes servicios REST.

Estructura del BloC

La arquitectura del patrón BLoC incluye tres componentes principales:

- Eventos (BlocAdminEvent): Los eventos son disparadores que provocan cambios en el estado. Por ejemplo, un evento para agregar un producto a la lista o para aplicar un pago.
- Estados (BlocAdminState): Los estados representan los diferentes escenarios o valores de la interfaz en un momento específico. Se define el estado inicial y los sucesivos estados resultantes de los eventos.
- Bloc (BlocAdminBloc): Este es el núcleo del patrón BLoC, que responde a los eventos y emite nuevos estados. Utiliza streams para manejar de manera eficiente los cambios de estado.

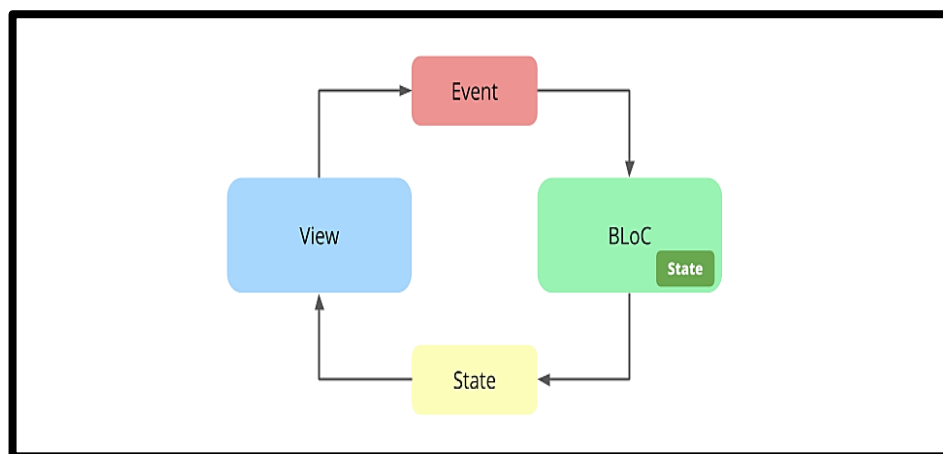


Figura 29 — Diagrama del patrón BLoC

La imagen muestra un diagrama del patrón BLoC (Business Logic Component), un patrón arquitectónico comúnmente utilizado en el desarrollo de aplicaciones Flutter.

Definición de eventos

Los eventos son acciones que pueden ser disparadas desde la interfaz de usuario o el backend y que modifican el estado de la aplicación. Aquí algunos eventos clave:

- GetCategoryByProduct: Este evento recupera las categorías de productos desde el servidor y actualiza el estado con la lista de productos y métodos de pago.
- AddProductList: Agrega un producto a la lista de productos seleccionados. Este evento también realiza cálculos de los impuestos, subtotales, y totales.
- ItemPlus: Incrementa la cantidad de un producto específico en la lista.
- ItemSubtract: Decrementa la cantidad de un producto específico, y si llega a cero, lo elimina de la lista.
- CreateSharedticket: Crea un ticket compartido para las mesas o habitaciones en el sistema.

Definición de estados

El estado es el resultado de los eventos disparados y contiene la información que se muestra en la interfaz de usuario. El estado inicial contiene

los valores predeterminados de los widgets. Algunos estados importantes son:

- productListForSearch: Contiene la lista de productos disponibles para búsqueda.
- globalTotal: Guarda el total calculado de todos los productos seleccionados.
- globalImpuesto: Calcula el impuesto total de los productos según su categoría tributaria.
- productItem: Es la lista de productos seleccionados que se muestran en la pantalla.

Lógica de la aplicación

La lógica principal se maneja dentro del método `mapEventToState`, el cual recibe eventos y genera un nuevo estado basado en las acciones realizadas. A continuación, se describen algunas de las operaciones clave:

Agregar productos a la lista

Cuando se agrega un producto a la lista, el evento `AddProductList` es disparado. Este evento genera un nuevo estado que actualiza la lista de productos seleccionados, realiza cálculos de impuestos, subtotales y totales, y los refleja en la interfaz de usuario.

```
1  if (event is AddProductList) {
2    String hash = uuid.v4();
3    double amount = (event.product.productUnit == null) ? 1 : event.product.productUnit!;
4
5    state.productItem!.add(Product(
6      productId: event.product.productId,
7      productIdItem: hash,
8      productDescription: event.product.productDescription,
9      productPriceSell: event.product.productPriceSell!,
10     productPriceTotal: event.product.productPriceSell! * amount,
11     productUnit: amount,
12   ));
13
14   // Recalcular totales
15   double total = 0;
16   for (var element in state.productItem!) {
17     total += element.productPriceTotal!;
18   }
19
20   yield state.copyWith(
21     productItem: state.productItem,
22     globalTotal: total,
23   );
24 }
```

Actualización de impuestos

La función tax maneja la lógica de cálculo de los impuestos según la categoría del producto (IGV, exonerado, inafecto).

```

1 Map<String, dynamic> tax(BlocAdminState state, Produc element, String operador) {
2   final map = <String, dynamic>{};
3   final priceSellAll = element.productPriceSell! * element.productUnit!;
4
5   switch (element.tax![0].taxName!) {
6     case "IGV|IMPUESTO IGV":
7       final taxPercent = element.tax![0].taxPercent!;
8       final divisor = 1 + taxPercent;
9       final taxAmount = priceSellAll - (priceSellAll / divisor);
10
11       map["impuesto"] = (operador == "+") ? state.globalImpuesto! + taxAmount : state.globalImpuesto! - taxAmount;
12       break;
13
14     case "EXO|EXONERADO":
15       map["exonerado"] = (operador == "+") ? state.globalExanorado! + priceSellAll : state.globalExanorado! - priceSellAll;
16       break;
17
18     case "INA|INAFECTO":
19       map["inafecto"] = (operador == "+") ? state.globalInafecto! + priceSellAll : state.globalInafecto! - priceSellAll;
20       break;
21   }
22
23   map.forEach((key, value) {
24     map[key] = value.abs();
25   });
26
27   return map;
28 }
29

```

Manejo de productos en tickets compartidos

Cuando se crea o actualiza un ticket compartido, se dispara el evento CreateSharedticket, que gestiona la creación del ticket y lo sincroniza con el backend.

```

1 if (event is CreateSharedticket) {
2   _map.clear();
3
4   _map = await _api.createSharedticket(
5     idTable: event.idTable,
6     idShare: event.idShared,
7   );
8
9   yield state.copyWith(
10     idSharedTicket: _map["data"]["responseData"]["sharedticket_id"],
11     idTable: event.idTable,
12     tableName: event.tableName
13   );
14 }
15

```

ix. Sistema multiplataforma

Un aspecto clave del sistema 'Rumisof-pos' es que ha sido optimizado para ser multiplataforma mediante el uso de la tecnología Dart a través del framework Flutter. Esto permite que el sistema pueda ser instalado y ejecutado en



cualquier equipo, independientemente de su sistema operativo, ya sea Windows, iOS, Android, Linux o en entornos web.

Esta implementación garantiza que se pueda acceder al sistema desde una amplia variedad de dispositivos, como computadoras de escritorio, tabletas o teléfonos móviles, manteniendo una experiencia de usuario fluida en todas las plataformas, como se observa en la Figura 30. Además, Flutter permite que el sistema mantenga un único código base para todas las plataformas, lo que facilita el mantenimiento y las actualizaciones, asegurando un rendimiento nativo en cada dispositivo.

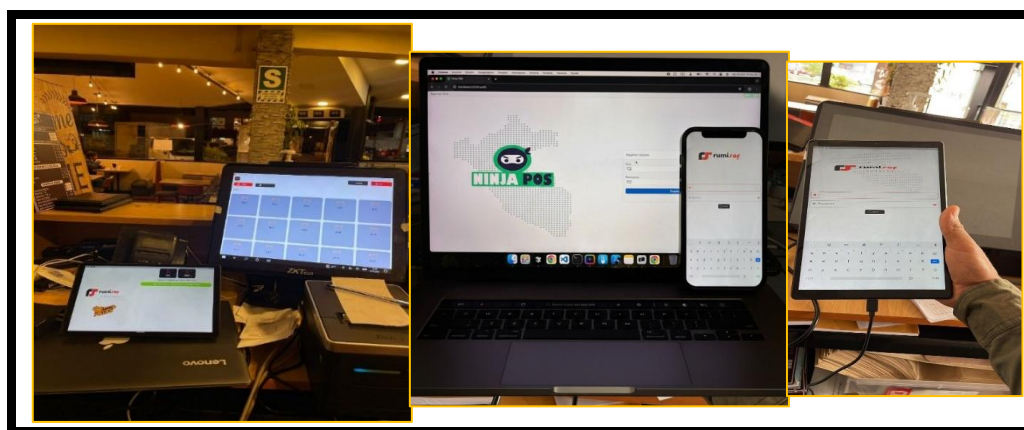


Figura 30 — Ejecución del sistema multiplataforma

f) Metodología de gestión y estructura del proyecto

Enfoque metodológico de desarrollo

Para la gestión eficiente del desarrollo del producto de software un sistema complejo que combina tecnologías de back-end (Laravel, Node.js) y una capa de presentación multiplataforma (Flutter) se adoptó el marco de trabajo ágil Kanban, una metodología ágil que fue elegida específicamente para maximizar la eficiencia del desarrollador único y gestionar el alto nivel de variabilidad introducido por los tres modos de negocio y el feedback de 47 empresas. Esta elección metodológica se justifica plenamente en la naturaleza del proyecto y la estructura del equipo de desarrollo.

Justificación de la elección

Kanban, formalizado por David J. Anderson (2010), se define como un sistema de gestión del flujo de trabajo que prioriza la eficiencia, la visualización de las

tareas y la mejora continua del proceso. La selección de Kanban sobre otras metodologías ágiles (como Scrum) se basó en los siguientes criterios específicos del proyecto:

Tabla 15 — Justificación de la metodología

Criterio	Justificación en el proyecto
Naturalidad individual del desarrollo	El proyecto fue ejecutado por un único desarrollador. Kanban se adapta perfectamente a este contexto al reemplazar las ceremonias de equipo (como el <i>Daily Scrum</i> y la planificación de <i>Sprints</i> rígidos) con auto-gestión y seguimiento continuo del flujo.
Flujo de trabajo variable	El proyecto exigió flexibilidad, especialmente en la integración con servicios externos (API SOAP/Factura Electrónica), donde las dependencias y los tiempos de respuesta de terceros son impredecibles. Kanban permite adaptar la priorización de tareas sin romper un <i>Sprint</i> .
Énfasis en el flujo y el enfoque	Dada la complejidad del ecosistema tecnológico, fue crucial evitar el <i>multitasking</i> . Kanban, mediante la limitación del Trabajo en Progreso (WIP), forzó la concentración en la finalización de un módulo (ej. el Conector SOAP) antes de iniciar otro (ej. la interfaz de Flutter).
Trazabilidad y visibilidad	La tesis requería una trazabilidad exhaustiva del desarrollo. La visualización del tablero Kanban (en Notion) sirvió como evidencia directa del progreso y la medición de métricas clave (<i>Cycle Time</i>).

Adaptación de Kanban para el desarrollo individual

Para optimizar el rendimiento del desarrollo en un contexto de investigación de tesis, el marco Kanban fue adaptado y simplificado en sus prácticas fundamentales:

a) Visualización del flujo de valor

Se implementó un tablero digital (utilizando Notion) con las siguientes columnas, representando el ciclo de vida de cada requisito funcional y no funcional:

1. **Backlog:** Requisitos validados y priorizados, listos para ser iniciados.
2. **Por Iniciar:** Tareas asignadas al ciclo de trabajo actual.
3. **En desarrollo (WIP 1):** Tareas en codificación activa. **Límite WIP = 1**
4. **En prueba/QA (WIP 1):** Tareas terminadas en código, pero bajo revisión de integración y validación funcional. **Límite WIP = 1**
5. **Listo para despliegue:** Tareas que han superado las pruebas (QA) y están pendientes de subida al entorno de *staging* o producción.
6. **Hecho (Done):** Tareas cerradas y validadas en el entorno objetivo.

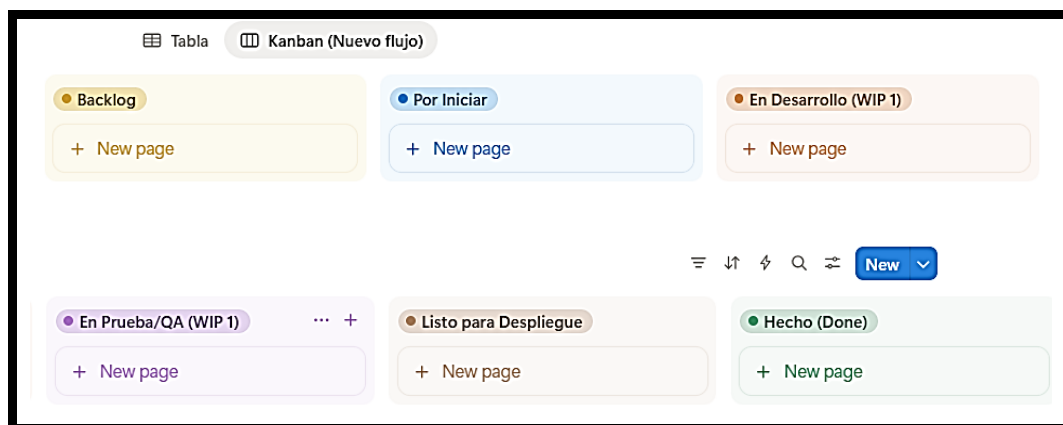


Figura 31 — Tablero digital de flujo de trabajo



Figura 32 — Tablero digital de flujo de trabajo con tareas

b) Gestión del flujo y métricas

La métrica principal de gestión fue el tiempo de ciclo (Cycle Time), medido desde el momento en que una tarea pasaba de "Por Iniciar" hasta que llegaba a "Hecho". La optimización continua de esta métrica fue la principal palanca para la mejora del proceso.

c) Prácticas de retroalimentación y mejora

Las ceremonias tradicionales de equipo fueron sustituidas por prácticas de auto-reflexión y gestión de backlog adaptadas:

1. **Daily stand-up individual (5 minutos diarios):** Una revisión personal de las actividades del día anterior, la planificación del día actual y la identificación de cualquier bloqueo o impedimento (*impediments*).

2. **Revisión semanal de backlog (30 minutos):** Una sesión estructurada para refinar los requisitos futuros, asegurar que las tareas de la Fase D (Pruebas) y Fase C (Desarrollo) estuvieran correctamente desglosadas, y reajustar las prioridades en función de los hallazgos técnicos.
3. **Retrospectiva quincenal (20 minutos):** Análisis reflexivo del flujo de trabajo de las últimas dos semanas. El foco fue identificar cuellos de botella (ej. tiempo excesivo en la columna "En Prueba") y aplicar ajustes incrementales al proceso (ej. mejorar la disciplina de pruebas unitarias).

Esta implementación de Kanban aseguró que el desarrollo, aunque realizado de forma individual, mantuviera un enfoque riguroso en la finalización de los elementos de valor, adaptándose a las exigencias técnicas y temporales del proyecto de tesis.

Roles consolidados y responsabilidades integrales del tesista

Dada la naturaleza del proyecto, el tesista, Mario Choquetaipe Asto, asumió todos los roles críticos. Esta estructura de responsabilidad integral fue esencial para mantener la coherencia arquitectónica del sistema de servicios compartidos.

Tabla 16 — Roles y responsabilidades

Rol asumido	Responsabilidad principal detallada
Arquitecto de software y product owner (PO)	Decisión clave: Establecer la arquitectura de servicios compartidos y la lógica del sistema de adaptación automática. Priorizar las funcionalidades que impactan el tiempo de proceso de venta según el <i>feedback</i> de las 47 empresas.
Desarrollador full-stack (multiplataforma)	Implementación de la lógica del <i>backend</i> (API central) y desarrollo del <i>frontend</i> multiplataforma (Flutter), asegurando que el código único sea funcional en los tres modos operativos.
Especialista en QA y soporte	Ejecución de pruebas de rendimiento (RNF01) y funcionalidad de modo (RF02). Validación de que el sistema cumple con la meta de reducir los tiempos de transacción.
Administrador de infraestructura (DevOps)	Configuración de Nginx y PM2 para garantizar la alta disponibilidad (RNF04) de la API central, crítica para un sistema de servicios compartidos.

g) Stack tecnológico y herramientas de desarrollo

El desarrollo del sistema Rumisof-POS se sustentó en un stack tecnológico robusto y contemporáneo, seleccionado estratégicamente para garantizar la escalabilidad, la seguridad y la capacidad multiplataforma de la solución. Este conjunto de tecnologías se clasifica en las siguientes categorías.



Lenguajes de programación

La arquitectura del sistema Rumisof-POS opera sobre una variedad de lenguajes, cada uno asignado a un rol específico para maximizar el rendimiento en diferentes entornos (cliente, servidor, tareas de background).

Tabla 17 — Lenguajes de programación

Lenguaje	Descripción	Aplicación específica
Dart	Lenguaje tipado estáticamente de alto rendimiento, optimizado para la construcción de interfaces de usuario (UI) fluidas.	Desarrollo de la interfaz de usuario <code>rumisof_ui</code> a través del <i>framework</i> Flutter.
PHP	Lenguaje maduro y altamente compatible con entornos web, seleccionado por su amplia base de <i>frameworks</i> robustos y facilidad de despliegue.	Construcción de las APIs principales: <code>api_client_rs</code> y <code>api_facturacion_rs</code> .
Node.js	Entorno de ejecución de JavaScript asíncrono y orientado a eventos, ideal para operaciones de I/O intensivas.	Desarrollo de <code>api_ninja_node</code> , enfocado en tareas de automatización y schedulers (procesos programados).
Java / Kotlin / Swift	Seleccionados para la integración nativa y el desarrollo de componentes específicos de alto rendimiento en los sistemas operativos correspondientes.	Desarrollo de módulos nativos para plataformas Android (Kotlin/Java) y iOS (Swift).

Frameworks de desarrollo

La selección de frameworks priorizó la velocidad de desarrollo y la funcionalidad multiplataforma.

Tabla 18 — Frameworks de desarrollo

Framework	Descripción	Componente asociado
Flutter	Framework Multiplataforma líder, elegido para permitir el despliegue de una única base de código en iOS, Android, Web, Linux y Windows, reduciendo el <i>time-to-market</i> y los costos de mantenimiento.	Aplicación de usuario <code>rumisof_ui</code> .
Laravel	Framework MVC de PHP, reconocido por su sintaxis elegante, seguridad integrada y provisión de herramientas para desarrollo rápido de APIs.	Servidor API principal: <code>api_client_rs</code> .

Gestión de datos

Se implementó un enfoque de gestión de datos dual para optimizar la confiabilidad transaccional y la velocidad de búsqueda.



Tabla 19 — Herramientas de BD

Tecnología	Tipo	Función principal en Rumisof-POS
MySQL	Sistema de gestión de bases de datos relacional (SGBDR).	Almacenamiento principal de todos los datos transaccionales (ventas, inventario, usuarios) debido a su confiabilidad y soporte ACID.
Meilisearch	Motor de búsqueda de código abierto.	Indexación y motor de búsqueda de baja latencia para optimizar consultas rápidas de productos y clientes en la interfaz de usuario.

Infraestructura y despliegue

La infraestructura se configuró buscando estabilidad operativa, alto rendimiento y gestión eficiente de los procesos.

Tabla 20 — Herramientas de infraestructura y despliegue

Herramienta / SO	Función estratégica	Rol en el despliegue
Ubuntu Server	Distribución Linux de grado empresarial.	Sistema operativo base para alojar todos los servicios de <i>backend</i> .
Nginx	Servidor web y proxy inverso de alto rendimiento.	Manejo de solicitudes web, balanceo de cargas y distribución de tráfico a los diferentes servicios API.
PM2	Gestor de procesos avanzado para entornos Node.js.	Mantenimiento de la continuidad operativa de las aplicaciones Node.js (<i>api_ninja_node</i>) y gestión de <i>logs</i> .

Herramientas de productividad y desarrollo

Estas herramientas facilitaron el desarrollo, las pruebas, la documentación y la colaboración del equipo.

Tabla 21 — Herramientas de productividad y desarrollo

Herramienta	Categoría	Propósito en el ciclo de desarrollo
Visual Studio Code	Editor de código (IDE).	Entorno de Desarrollo Integrado (IDE) principal, seleccionado por su extensibilidad y soporte nativo para los lenguajes utilizados.
Postman	Testing de APIs.	Herramienta estándar para la validación funcional y no funcional de los <i>endpoints</i> REST y para generar documentación automatizada.
DBeaver	Cliente de bases de datos.	Administración, monitorización y gestión de esquemas de bases de datos relacionales (MySQL).
Notion	Gestión de proyectos y documentación.	Espacio de trabajo centralizado para la documentación técnica, la gestión de tareas (<i>backlog</i>) y la creación de la Wiki del proyecto.

Excalidraw	Diseño y arquitectura visual.	Creación rápida de diagramas de arquitectura y flujos de trabajo durante las fases de diseño y planificación.
------------	-------------------------------	---

5.2 Contrastación de hipótesis

Antes de la comprobación de hipótesis se presentan las pruebas de normalidad.

Para las investigaciones antes – después, en las que se miden repetidamente a los sujetos de una misma muestra, se genera una nueva variable con la diferencia del antes y después y se prueba la normalidad en esta nueva variable, y no de manera individual en cada variable.

Tabla 22 — Pruebas de normalidad

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tiempo de proceso de orden diferencia	,645	47	,000
Tiempo de generación de factura diferencia	,637	47	,000
Tiempo de proceso de venta diferencia	,641	47	,000

La muestra es menor que 50, por lo que se analiza el estadístico de Shapiro-Wilk para establecer la normalidad. La regla de decisión indica que si p-valor es menor al nivel de significancia entonces no se puede aceptar la hipótesis nula, en caso contrario se acepta la hipótesis nula.

Como se observa en la Tabla 22, la Sig. (Significancia asintótica bilateral) o p-valor es 0 para todos los casos y el nivel de significancia es 0.05, por lo que $p\text{-valor} < \alpha$. Entonces no se puede aceptar la hipótesis nula y se acepta la hipótesis alterna que indica que los datos no tienen una distribución normal para las tres variables diferencia.

En vista que ninguna variable cumple el supuesto de normalidad, se realizará un análisis inferencial no paramétrico para la diferencia de medias, aplicando el estadístico de Wilcoxon para la comprobación de las hipótesis.

Hipótesis específica 1

Al desarrollar e implementar un sistema de información de servicios compartidos multi-plataforma mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisofpos”, 2024.

1. Hipótesis estadísticas



H_0 : El tiempo de proceso de órdenes antes del desarrollo e implementación del sistema de información es menor o igual al tiempo de proceso de órdenes después del desarrollo e implementación del sistema de información.

H_1 : El tiempo de proceso de órdenes antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de órdenes después del desarrollo e implementación del sistema de información.

2. Nivel de significancia

Se definió el nivel de significancia en 5%, es decir $\alpha = 0.05$

3. Estadístico de prueba

Debido a la naturaleza de la hipótesis, se pretende comparar las medias de ambas mediciones (antes y después), además no cumplen las pruebas de normalidad. Por lo que se utiliza el estadístico no paramétrico de Wilcoxon, este análisis se realizó en el software estadístico SPSS Statistics.

Se realizó la comparación de medias de las variables tiempo de proceso de órdenes antes y tiempo de proceso de órdenes después del desarrollo e implementación del sistema de información de servicios compartidos multiplataforma.

Tabla 23 — Prueba de rangos con signo de Wilcoxon HE1

	Z	Sig. asintótica(bilateral)
Tiempo de proceso de orden después - Tiempo de proceso de orden antes	-5,228 ^b	,000

b. Se basa en rangos positivos.

4. Lectura del p-valor

Como se observa en la Tabla 23, el p-valor es 0, pero este p-valor es para un estudio bilateral (dos colas), pero en nuestro caso la hipótesis es unilateral (una cola) por lo que debemos dividir el valor anterior entre 2. Así, para nuestra prueba el p-valor es igual a 0.

El cálculo del p-valor nos permite afirmar que: con una probabilidad de error del 0% el tiempo de proceso de órdenes antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de órdenes después del desarrollo e implementación del sistema de información.

5. Decisión estadística

Como $p - \text{valor} < \alpha$, existe evidencia estadística para no aceptar la hipótesis nula (H_0) con un nivel de significancia del 5%, por lo tanto, se acepta la hipótesis alterna (H_1).

Esto es, que el tiempo de proceso de órdenes antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de órdenes después del desarrollo e implementación del sistema de información.

Lo cual comprueba la hipótesis específica 1 que dice: Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024.

Hipótesis específica 2

Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.

1. Hipótesis estadísticas

H_0 : El tiempo de la generación de facturas antes del desarrollo e implementación del sistema de información es menor o igual al tiempo de la generación de facturas después del desarrollo e implementación del sistema de información.

H_1 : El tiempo de la generación de facturas antes del desarrollo e implementación del sistema de información es mayor al tiempo de la generación de facturas después del desarrollo e implementación del sistema de información.

2. Nivel de significancia

Se definió el nivel de significancia en 5%, es decir $\alpha = 0.05$

3. Estadístico de prueba

Debido a la naturaleza de la hipótesis, se pretende comparar las medias de ambas mediciones (antes y después), además no cumplen las pruebas de normalidad. Por lo que se utiliza el estadístico no paramétrico de Wilcoxon, este análisis se realizó en el software estadístico SPSS Statistics.

Se realizó la comparación de medias de las variables tiempo de la generación de facturas antes y tiempo de la generación de facturas después del desarrollo e implementación del sistema de información de servicios compartidos multiplataforma.

Tabla 24 — Prueba de rangos con signo de Wilcoxon HE2

	Z	Sig. asintótica(bilateral)
Tiempo de generación de factura después - Tiempo de generación de factura antes	-5,969 ^b	,000

b. Se basa en rangos positivos.

4. Lectura del p-valor

Como se observa en la Tabla 24, el p-valor es 0, pero este p-valor es para un estudio bilateral (dos colas), pero en nuestro caso la hipótesis es unilateral (una cola) por lo que debemos dividir el valor anterior entre 2. Así, para nuestra prueba el p-valor es igual a 0.

El cálculo del p-valor nos permite afirmar que: con una probabilidad de error del 0% el tiempo de la generación de facturas antes del desarrollo e implementación del sistema de información es mayor al tiempo de la generación de facturas después del desarrollo e implementación del sistema de información.

5. Decisión estadística

Como $p - valor < \alpha$, existe evidencia estadística para no aceptar la hipótesis nula (H_0) con un nivel de significancia del 5%, por lo tanto, se acepta la hipótesis alterna (H_1).

Esto es, que el tiempo de la generación de facturas antes del desarrollo e implementación del sistema de información es mayor al tiempo de la generación de facturas después del desarrollo e implementación del sistema de información.

Lo cual comprueba la hipótesis específica 2 que dice: Al desarrollar e implementar un sistema de información de servicios compartido multiplataforma se mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.

Hipótesis general

Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.

1. Hipótesis estadísticas

H_0 : El tiempo de proceso de venta antes del desarrollo e implementación del sistema de información es menor o igual al tiempo de proceso de venta después del desarrollo e implementación del sistema de información.

H_1 : El tiempo de proceso de venta antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de venta después del desarrollo e implementación del sistema de información.

2. Nivel de significancia



Se definió el nivel de significancia en 5%, es decir $\alpha = 0.05$

3. Estadístico de prueba

Debido a la naturaleza de la hipótesis, se pretende comparar las medias de ambas mediciones (antes y después), además no cumplen las pruebas de normalidad. Por lo que se utiliza el estadístico no paramétrico de Wilcoxon, este análisis se realizó en el software estadístico SPSS Statistics.

Se realizó la comparación de medias de las variables tiempo de proceso de venta antes y tiempo de proceso de venta después del desarrollo e implementación del sistema de información de servicios compartidos multiplataforma.

Tabla 25 — Prueba de rangos con signo de Wilcoxon HG

	Z	Sig. asintótica(bilateral)
Tiempo de proceso de venta después - Tiempo de proceso de venta antes	-5,968 ^b	,000

b. Se basa en rangos positivos.

4. Lectura del p-valor

Como se observa en la Tabla 25, el p-valor es 0, pero este p-valor es para un estudio bilateral (dos colas), pero en nuestro caso la hipótesis es unilateral (una cola) por lo que debemos dividir el valor anterior entre 2. Así, para nuestra prueba el p-valor es igual a 0.

El cálculo del p-valor nos permite afirmar que: con una probabilidad de error del 0% el tiempo de proceso de venta antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de venta después del desarrollo e implementación del sistema de información.

5. Decisión estadística

Como $p - valor < \alpha$, existe evidencia estadística para no aceptar la hipótesis nula (H_0) con un nivel de significancia del 5%, por lo tanto, se acepta la hipótesis alterna (H_1).

Esto es, que el tiempo de proceso de venta antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de venta después del desarrollo e implementación del sistema de información.

Lo cual comprueba la hipótesis general que dice: Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.

5.3 Discusión

En cuanto al objetivo general, que pretende desarrollar e implementar un sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024; en base a los resultados se aceptó la hipótesis alterna, esto es que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.

Estos resultados muestran semejanzas al ser comparados con lo investigado por (Principe y Pariasca 2018), los cuales desarrollaron un sistema web que logró mejorar el proceso de gestión de ventas en la empresa Factory Solutions. Asimismo, el estudio de (Huaman y Huayanca 2017) determinó la disminución de tiempos en los procesos de compra y venta en la empresa Humaju al implementar un sistema de información con metodología AUP. También en el estudio de (Ludeña y Rodríguez 2013) se concluyó que el tiempo promedio de compra de entradas de cine en la ciudad de Trujillo es menor con el sistema de información web-móvil propuesto. Chugnas (2014) también concluyó que mediante la implementación del sistema de gestión de ventas se redujo el tiempo de atención al cliente en un 30%. Además, la investigación de Arredondo (2023) determinó que luego de la implementación de un sistema web, el proceso de venta en la empresa Lconst Contratistas Generales mejoró en un 20%.

En cambio, los resultados encontrados en esta investigación muestran diferencias con lo encontrado por (Romero, La Paz y García 2024) quienes determinaron que ante aumentos en especialistas TIC y software ERP generan aumentos en las ventas de empresas chilenas. (Vargas 2017) implementó un sistema web y se determinó que el promedio de pedidos por cliente aumentó en 0.09 y el porcentaje de fidelidad aumentó en 9.25%. Y Carrillo (2017), que tras implementar un sistema de información para mejorar la gestión de compra, venta y almacén concluyó que la percepción de los trabajadores sobre el sistema fue positiva. Los estudios mencionados en este párrafo difieren a los hallazgos de este estudio debido a que tienen objetivos diferentes.

Sobre el objetivo específico 1, que pretende desarrollar e implementar un sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024; en base a los resultados se aceptó la hipótesis alterna, esto es que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024.



Los estudios que reafirman lo encontrado en esta investigación son Chugnas (2014) que concluyó que al desarrollar un sistema de gestión de ventas se redujo el tiempo de atención al cliente en 30%. (Ludeña y Rodríguez 2013) también concluyen que al desarrollar el sistema de información web-móvil se redujo el tiempo promedio de compra de entradas al cine. (Huaman y Huayanca 2017) demostraron en su estudio que la implementación del sistema de información usando la metodología AUP disminuyó el tiempo para elaborar orden y comprobante de pedido.

El objetivo específico 2, que pretende desarrollar e implementar un sistema de información de servicios compartido multiplataforma para mejorar el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024; en base a los resultados se aceptó la hipótesis alterna, esto es que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.

En el mismo sentido tenemos a (Principe y Pariasca 2018) que en su investigación concluyeron que el tiempo empleado en la elaboración del comprobante de venta mejoró en 54% con la implementación de un sistema web. (Huaman y Huayanca 2017) determinaron que la implementación del sistema de información usando la metodología AUP disminuyó el tiempo para elaborar orden y comprobante de pedido.



CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El desarrollo e implementación del sistema de información de servicios compartidos multiplataforma "Rumisof-pos" permitió demostrar empíricamente su impacto en la optimización de los procesos de venta en las empresas usuarias. Los resultados obtenidos, respaldados por el análisis estadístico ($p\text{-valor} = 0$), conducen a las siguientes conclusiones:

- En cuanto al tiempo total del proceso de venta, se determinó que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024. Al realizar un análisis descriptivo de los datos medidos antes y después de la implementación del sistema, se encontró que el promedio del tiempo del proceso de venta antes de la implementación fue de 11.8 minutos, frente a 2.73 minutos que fue el promedio del tiempo después de la implementación del sistema, lo que constituye una reducción de 76.8%. El cálculo del $p\text{-valor}$ nos permite afirmar que con una probabilidad de error del 0% el tiempo de proceso de ventas antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de ventas después del desarrollo e implementación del sistema de información. Esto se explica debido a que los sistemas de información automatizan y hacen más eficiente el uso de los recursos. Por lo que se debe implementar este tipo de sistemas de información para la gestión de ventas en las empresas.
- En cuanto al proceso de órdenes, se determinó que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024. El promedio del tiempo de proceso de órdenes antes del sistema fue de 5.88 minutos y el promedio después del sistema fue de 1.91 minutos. Lo que representa una disminución del 67.5% del tiempo en esta dimensión. El cálculo del $p\text{-valor}$ nos permite afirmar que



con una probabilidad de error del 0% el tiempo de proceso de órdenes antes del desarrollo e implementación del sistema de información es mayor al tiempo de proceso de órdenes después del desarrollo e implementación del sistema de información.

- En cuanto al proceso de facturación, se determinó que al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma se mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024. La reducción del tiempo aquí fue del 86.1%, constituye la mejora más significativa del estudio, se atribuye a la automatización completa del proceso (validación de datos, generación del XML, firma digital, envío a SUNAT y obtención del CDR). El cálculo del p-valor nos permite afirmar que con una probabilidad de error del 0% el tiempo de la generación de facturas antes del desarrollo e implementación del sistema de información es mayor al tiempo de la generación de facturas después del desarrollo e implementación del sistema de información.

6.2 Recomendaciones

Se recomienda a futuros investigadores y desarrolladores ampliar las funcionalidades del sistema actual, integrando nuevos módulos como la gestión de inventario, análisis predictivo usando IA y/o API pública para conectar con otras plataformas como Rappi o Uber Eats.

Explorar su adaptación a otros rubros con alta demanda de servicios compartidos, como logística o turismo cultural.

Incorporar estándares WCAG 2.1 en futuras actualizaciones, asegurando compatibilidad con lectores de pantalla para usuarios con discapacidad visual.

El estudio no analizó el costo-beneficio a largo plazo ni la curva de aprendizaje del personal, aspectos críticos para la sostenibilidad del sistema en microempresas con recursos limitados.

Además, a partir de los hallazgos obtenidos en la presente investigación, se plantean las siguientes recomendaciones estratégicas orientadas a potenciar el impacto del sistema Rumisof-pos en el contexto empresarial peruano:

- Para el sector restaurantes, se recomienda implementar un módulo de cartas digitales con código QR que permita a los clientes visualizar el menú, realizar pedidos directamente desde su dispositivo móvil y efectuar el pago mediante integración con Yape

o Plin. Esta funcionalidad reduciría significativamente los tiempos de espera y optimizaría la labor del personal durante horas pico. Adicionalmente, sería estratégico desarrollar una API de conexión bidireccional con plataformas de delivery como Rappi y Uber Eats, permitiendo la sincronización automática de pedidos y la actualización en tiempo real del inventario, un aspecto crítico para restaurantes peruanos que operan en múltiples canales simultáneamente.

- En el contexto hotelero peruano, caracterizado por un creciente flujo turístico, se sugiere incorporar un sistema de check-in/check-out digital. Los huéspedes podrían realizar estos trámites de forma anticipada desde su celular, recibiendo una llave digital (QR) para acceder a su habitación. Integrar este módulo con una pasarela de pagos local que acepte soles y dólares facilitaría la gestión para el turista extranjero. Asimismo, conectar el sistema con plataformas de reservas como Booking.com o Airbnb automatizaría la actualización de la disponibilidad, reduciendo errores de sobre-reserva, un problema común en el sector.
- Para minimarkets y bodegas, que constituyen el comercio minorista más extendido en el Perú, se recomienda el desarrollo de un módulo de fidelización digital. Este sistema registraría automáticamente las compras de los clientes recurrentes, otorgando puntos canjeables por productos. La integración con pasarelas de pago locales agilizaría las transacciones, mientras que una funcionalidad de "pedidos recurrentes" para abarrotes permitiría a los clientes programar entregas periódicas, fomentando su lealtad.
- Para asegurar la adopción efectiva y sostenibilidad del sistema, es crucial complementar el desarrollo tecnológico con un programa de capacitación avalado por cámaras de comercio locales, enfocado en cerrar la brecha de competencias digitales identificada en los propietarios de MIPYMES. Adicionalmente, se recomienda a futuros investigadores analizar el impacto del sistema en la formalización de negocios y realizar un estudio de costo-beneficio que considere la curva de aprendizaje del personal, aspectos críticos para la viabilidad a largo plazo en microempresas con recursos limitados. Finalmente, se sugiere explorar la adaptación del sistema a otros rubros con alta demanda de servicios compartidos en el Perú, como la logística para el transporte de mercancías o la gestión de operaciones en el creciente sector del turismo cultural.

REFERENCIAS BIBLIOGRÁFICAS

- ABADDI, S., 2025. Analysis and improvement of sales management through CRM and simulation utilisation. *Applied Operations and Analytics*, vol. 1, no. 1, ISSN 2996-6892. DOI 10.1080/29966892.2024.2434267.
- ACOSTA, V., GARCIA, L. y PACHECO, A., 2024. FAST a Digital Sales Transformation Software for the Real Estate Industry. *F1000Research*, vol. 13, pp. 264. ISSN 2046-1402. DOI 10.12688/f1000research.145185.1.
- AGBOOLA, T., ADEGEDE, J. y JACOB, J., 2024. Balancing Usability and Security in Secure System Design: A Comprehensive Study on Principles, Implementation, and Impact on Usability. *International Journal of Computing Sciences Research*, vol. 8, pp. 2995-3009. ISSN 25460552. DOI 10.25147/ijcsr.2017.001.1.199.
- AGURTO ORELLANA, L.A. y FLORES MECA, J.F., 2024. *Implementación de un sistema CRM para mejorar la gestión de ventas en una empresa importadora en Lima, 2023* [en línea]. Tesis de Pregrado. Lima: Universidad Tecnológica del Perú. [consulta: 18 octubre 2025]. Disponible en: <https://hdl.handle.net/20.500.12867/8694>.
- AKTURK, M.S., MALLIPEDDI, R.R. y JIA, X., 2022. Estimating impacts of logistics processes on online customer ratings: Consequences of providing technology-enabled order tracking data to customers. *Journal of Operations Management*, vol. 68, no. 6-7, pp. 775-811. ISSN 0272-6963. DOI 10.1002/joom.1204.
- ALARCÓN OSUNA, M.A., OCAMPO FIGUEROA, L.E. y GUZMÁN ANAYA, L., 2024. Presente y futuro de la micro, pequeña y mediana empresa desde una perspectiva holística. *Carta Económica Regional*, vol. 0, no. 134, pp. 8-22. ISSN 01877674. DOI 10.32870/cer.v0i134.7917.
- ALEXEI, Arina y ALEXEI, Anatolie, 2023. THE DIFFERENCE BETWEEN CYBER SECURITY VS INFORMATION SECURITY. *Journal of Engineering Science*, vol. 29, no. 4, pp. 72-83. ISSN 25873474. DOI 10.52326/jes.utm.2022.29(4).08.
- ALI, Md.F. y KHAN, R.Z., 2012. The Study On Load Balancing Strategies In Distributed Computing System. *International Journal of Computer Science & Engineering Survey*, vol. 3, no. 2, pp. 19-30. ISSN 09763252. DOI 10.5121/ijcses.2012.3203.
- ANCHIRAICO GARCIA, R., 2022. *Implementación de sistema web multiplataforma con inteligencia de negocios para mejorar la gestión de ventas en la farmacia Fiorella* -



- La Victoria*, 2021 [en línea]. Tesis de Pregrado. Lima: Universidad Tecnológica del Perú. [consulta: 18 octubre 2025]. Disponible en: <https://hdl.handle.net/20.500.12867/5955>.
- ANDERSON, D.J., 2010. *Kanban : successful evolutionary change in your technology business*. S.l.: Blue Hole Press. ISBN 0984521402.
- APPVIZER, 2024. Descubre los 8 pasos fundamentales del proceso de facturación. *Appvizer* [en línea]. [consulta: 24 octubre 2025]. Disponible en: <https://www.appvizer.es/revista/contabilidad-finanzas/facturacion/proceso-de-facturacion>.
- ARIAS-GÓMEZ, J., VILLASÍS-KEEVER, M.Á. y MIRANDA NOVALES, M.G., 2016. El protocolo de investigación III: la población de estudio. *Revista Alergia México*, vol. 63, no. 2, pp. 201-206.
- ARREDONDO, R., 2023. *Implementación de un sistema web para el proceso de ventas en la empresa Lconst Contratistas Generales E.I.R.L* [en línea]. Tesis de Pregrado. Lima: Universidad Autónoma del Perú. Disponible en: <https://hdl.handle.net/20.500.13067/2722>.
- ARRIVASPLATA ROJAS, J.G., 2024. *Implementación de un Sistema Web para mejorar la Gestión de Ventas en la Empresa Sumac Chasca Perú S.A.C* [en línea]. Tesis de Pregrado. Lima: Universidad Tecnológica del Perú. [consulta: 18 octubre 2025]. Disponible en: <https://hdl.handle.net/20.500.12867/9536>.
- BERTONE, P.R., 2020. *La venta como un proceso: diseñando modelos de gestión de ventas* [en línea]. Tesis de Maestría. Córdoba: Universidad Católica de Córdoba. [consulta: 23 octubre 2025]. Disponible en: <http://pa.bibdigital.ucc.edu.ar/id/eprint/1990>.
- BROOKE, J., 1986. System Usability Scale (SUS). . London:
- BROOKE, J., 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* [en línea], vol. 189, [consulta: 21 octubre 2025]. Disponible en: https://digital.ahrq.gov/sites/default/files/docs/survey/systemusabilityscale%2528sus%2529_comp%255B1%255D.pdf.
- CANTÃO, R.F., RIBEIRO-OLIVEIRA, J.P., SILVA, E.A.A. da, SANTOS, A.R. dos, DE FARIA, R.Q. y SARTORI, M.M.P., 2023. POMONA: a multiplatform software for modeling seed physiology. *Frontiers in Plant Science*, vol. 14, no. July, pp. 1-13. ISSN 1664462X. DOI 10.3389/fpls.2023.1151911.



- CARGOSAPIENS, 2025. KPIs logísticos: qué son, principales indicadores y cómo aplicarlos en tu empresa. *CargoSapiens* [en línea]. [consulta: 24 octubre 2025]. Disponible en: <https://cargosapiens.com/es/blog/kpis-logistica/>.
- CARRASCO, S. y NÚÑEZ, M., 2023. *Organización de procesos de venta*. Santiago de Compostela: Ediciones Paraninfo. ISBN 978-84-283-5945-0.
- CARRILLO, J., 2017. *Implementación de un sistema de Información para mejorar la gestión de los procesos de compra, venta y almacén de Productos deportivos en la tienda Casa de Deportes Rojitas E.I.R.Ltda. - Chimbote; 2014* [en línea]. Tesis de Pregrado. Chimbote: Universidad Los Ángeles de Chimbote. Disponible en: <https://hdl.handle.net/20.500.13032/1459>.
- CEPAL, 2022. La Digitalización Empresarial en América Latina: Oportunidades y Desafíos para la Recuperación Post-Pandemia. . Santiago de Chile:
- CHEN, K., SOCKEL, H. y FALK, L.K., 2009. Usability in Mobile Computing and Commerce. *Encyclopedia of Multimedia Technology and Networking, Second Edition*. S.l.: IGI Global, pp. 1480-1486.
- CHEN, Yuanchao, LI, Yuwei, LU, Y., PAN, Z., CHEN, Yuan, JI, S., CHEN, Yu, LI, Yang y SHEN, Y., 2025. Understanding the Security Risks of Websites Using Cloud Storage for Direct User File Uploads. *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2677-2692. ISSN 1556-6013. DOI 10.1109/TIFS.2025.3544082.
- CHUGNAS, J., 2014. *Sistema de gestión de ventas para reducir el tiempo de atención al cliente y aumentar el margen de utilidad en la empresa "Corporación Dijol SRL" en el año 2012* [en línea]. Tesis de Pregrado. Lima: Universidad Privada del Norte. Disponible en: <https://hdl.handle.net/11537/3164>.
- COHAILA LARA, C.M., 2020. *Análisis comparativo de los paradigmas de la programación, orientada a objetos y programación orientada a aspectos en la construcción del módulo de autenticación del software de la biblioteca de la Liga Departamental de Fútbol de Tacna - 2015*. Tesis de Pregrado. Tacna: Universidad Nacional Jorge Basadre Grohmann.
- CONGRESO DE LA REPÚBLICA DEL PERÚ, 2003. *Ley N° 28015. Ley de Promoción y Formalización de la Micro y Pequeña Empresa*. 2003. Lima: <https://www.sunat.gob.pe/orientacion/mypes/normas/ley-28015.pdf>. Ley N° 28015.
- DELÍA, L., 2017. *Desarrollo de Aplicaciones Móviles Multiplataforma*. S.l.: Universidad Nacional de la Plata.



- DR. THIRUCHELVI, SUDHIKSHA S, VARUNI TAMILMARAI, AASIA PARVEEN SHABBIR, DEEBADARSENE SUNDARAM y PRASHEETHA BALAMURALI, 2025. Impact of waiting time on customer satisfaction in quick commerce: The moderating role of compensation strategy. *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 568-575. ISSN 25828185. DOI 10.30574/ijjsra.2025.15.2.1371.
- DUEMINT, 2024. Ciclo de vida de una factura: KPIs para cada etapa del proceso de cobranza. *Duemint* [en línea]. [consulta: 24 octubre 2025]. Disponible en: <https://www.duemint.com/blog/ciclo-de-vida-de-una-factura-kpis-para-cada-etapa-del-proceso-de-cobranza>.
- DZYUBLYUK, O., 2020. Essential aspects of electronic money in the conditions of digitalization of banking business. *Herald of Economics*, vol. 2, no. 96, pp. 76-92. DOI <https://doi.org/10.35774/visnyk2020.02.076>.
- ESTEBAN NIETO, N.T., 2018. Tipos de investigación. . Lima:
- EZENWOYE, O., 2024. Mitigating Injection-Based Weaknesses: A Flexible Template Architecture. *The Journal of Object Technology*, vol. 23, no. 1, pp. 1:1. ISSN 1660-1769. DOI 10.5381/jot.2024.23.1.a3.
- FAGUNDES, P.L., PEREIRA, G.M., TRENTA, L.R., MILAN, G.S., BORCHARDT, M. y SOUZA, M. de, 2024. Alavancagem das vendas por cross-selling e up-selling por meio de soluções digitais. *Revista Produção Online*, vol. 24, no. 2, pp. 5241. ISSN 1676-1901. DOI 10.14488/1676-1901.v24i2.5241.
- FOJTIK, R., 2020. Swift a New Programming Language for Development and Education. *igital Science 2019. DSIC 2019. Advances in Intelligent Systems and Computing*. S.l.: Springer, pp. 284-295. vol. 1114.
- GALLINO, S., KARACAOGLU, N. y MORENO, A., 2023. Need for Speed: The Impact of In-Process Delays on Customer Behavior in Online Retail. *Operations Research*, vol. 71, no. 3, pp. 876-894. ISSN 0030-364X. DOI 10.1287/opre.2022.2262.
- GEORGE, D. y MALLERY, P., 2003. *SPSS for Windows step by step: A simple guide and reference*. 4. Boston: Allyn & Bacon.
- GHERARDI, L., BRUGALI, D. y COMOTTI, D., 2013. A Java vs. C++ performance evaluation: a 3D modeling benchmark. *BRICS* [en línea], [consulta: 25 octubre 2025]. Disponible en: <https://best-of-robotics.org/pages/publications/gherardi12java.pdf>.



- GKREKOS, I., CHATZOPOULOS, A., PAPOUTSIDAKIS, M. y KONG LEE, W., 2019. USES AND APPLICATIONS OF UBUNTU: A TECHNICAL GUIDE. *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 5, pp. 444-454. ISSN 2455-2143.
- GONCALVES COSTA, D., 2017. *Desenvolvimento de módulos para plataforma Intranet especializada em contexto universitario*. Tesis doctoral. Coimbra: Universidade de Coimbra (Portugal).
- GONZÁLEZ JIMÉNEZ, S.M. y NOVOA VARGAS, R.A., 2024. Reestructuración del modelo de facturación en salud para una IPS de servicios domiciliarios en Colombia. *Revista CIFE: Lecturas de Economía Social*, vol. 25, no. 43, ISSN 2248-4914. DOI 10.15332/22484914.9676.
- HERNÁNDEZ-SAMPIERI, R. y MENDOZA, C., 2018. *Metodología de la Investigación. Las rutas cuantitativa cualitativa y mixta*. S.l.: McGraw Hill. ISBN 978-1-4562-6096-5.
- HUAMAN, J. y HUAYANCA, C., 2017. *Desarrollo e implementación de un sistema de información para mejorar los procesos de compras y ventas en la empresa Humaju*. S.l.: Universidad Autónoma del Perú.
- INEI, 2013. Micro, Pequeñas y Medianas empresas concentran más del 20% de las ventas.
- JAMDAADE, K. y CHAMPANERI, H., 2015. A Review: Secured Electronic Payment Gateway. *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 3, pp. 70-72.
- JONES, M., BRADLEY, J. y SAKIMURA, N., 2015. JSON Web Token (JWT). *IETF* [en línea]. [consulta: 25 octubre 2025]. Disponible en: <https://www.rfc-editor.org/rfc/rfc7519>.
- KHAN, R.A., KHAN, S.U., KHAN, H.U. y ILYAS, M., 2022. Systematic Literature Review on Security Risks and its Practices in Secure Software Development. *IEEE Access*, vol. 10, pp. 5456-5481. ISSN 2169-3536. DOI 10.1109/ACCESS.2022.3140181.
- KOC COLLAZOS, C. y PIZÁN SALCEDO, M., 2024. *Implementación de un modelo de gestión para reducir retrasos en los pedidos de la empresa Fuyao Group Perú S.A.C*. S.l.: Universidad Peruana de Ciencias Aplicadas.



- KOTLER, P. y ARMSTRONG, G., 2012. *Marketing*. 14. Mexico: PEARSON EDUCACIÓN.
- KOTRIKADZE, G., 2023. INFORMATION SECURITY AND CYBER SECURITY. *Georgian Scientists*, vol. 5, no. 1, pp. 322-333. ISSN 26679760. DOI 10.52340/gs.2023.05.01.28.
- LAUDON, K. y LAUDON, J., 2012. *Sistemas de Información Gerencial*. 12. México: PEARSON EDUCACIÓN. ISBN 978-607-32-0949-6.
- LEWIS, J.R., 2018. The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, vol. 34, no. 7, pp. 577-590. ISSN 1044-7318. DOI 10.1080/10447318.2018.1455307.
- LIFEDER, 2021. 7 Técnicas e Instrumentos para la Recolección de Datos. [en línea]. [consulta: 18 marzo 2025]. Disponible en: <https://www.lifeder.com/tecnicas-instrumentos-recoleccion-datos/>.
- LÓPEZ, A. y RUÍZ, V., 2021. *Sistema Web basado en la Metodología XP para la Gestión de Pedidos en el Restaurante Náutico de Trujillo*. S.l.: Universidad César Vallejo.
- LUDEÑA, J. y RODRIGUEZ, A., 2013. *Sistema de información Web-Móvil para mejorar la gestión de ventas de entradas de cine en la ciudad de Trujillo*. S.l.: Universidad Privada del Norte.
- MARTINS DE PINHO, D., 2017. *ECMAScript 6 : Entre de Cabeça No Futuro Do JavaScript*. S.l.: Casa do Código. ISBN 9788555192593.
- MELGAREJO CHAVEZ, E.I. y MUÑOZ MURGA, D.J., 2024. *Desarrollo de un sistema web para optimizar el proceso de ventas, área de comercialización de minerales, Ingeniería Soft, Trujillo 2024* [en línea]. Trabajo de suficiencia profesional. Lima: Universidad Privada del Norte. [consulta: 18 octubre 2025]. Disponible en: <https://hdl.handle.net/11537/41992>.
- MINISTERIO DE LA PRODUCCIÓN, 2024. LAS MIPYME EN CIFRAS 2023. . Lima:
- NIEMI, T., HAMERI, A.-P., KOLESNYK, P. y APPELQVIST, P., 2020. What is the value of delivering on time? *Journal of Advances in Management Research*, vol. 17, no. 4, pp. 473-503. ISSN 0972-7981. DOI 10.1108/JAMR-12-2019-0218.
- ONGALLO CHACLÓN, C., 2012. El proceso de venta. *El libro de la venta directa*. Madrid: Editorial Díaz de Santos, pp. 63-70. ISBN 978-84-9969-446-7.



- OWASP FOUNDATION, 2021. OWASP Top 10. *OWASP Top 10:2021* [en línea]. [consulta: 25 octubre 2025]. Disponible en: <https://owasp.org/Top10/>.
- PAHL, C. y JAMSHIDI, P., 2016. Microservices: A Systematic Mapping Study. *Proceedings of the 6th International Conference on Cloud Computing and Services Science*, vol. 1, pp. 137-146. DOI <https://doi.org/10.5220/0005785501370146>.
- PARODI, A., MONTAÑEZ, S. y CHAVEZ UGAZ, R., 2024. Optimization and Standardization of the Sales Process in a Service Sector Company through Lean Tools. *Proceedings of the International Conference on Industrial Engineering and Operations Management*. Michigan, USA: IEOM Society International, ISBN 979-8-3507-1734-1. DOI 10.46254/AN14.20240171.
- PARVIN, M. y RAMYAR, F., 2024. *Role of Sales and Marketing Information Systems in Companies' Sales Rate Increase*. 3 octubre 2024. Auburn: Auburn University.
- PAZ, R., 2005. *Servicio al cliente. La comunicación y la calidad del servicio en la atención al cliente*. Vigo: Ideaspropias Editorial. ISBN 978-84-96578-12-8.
- POLJAK, J., DEVIĆ, I. y GLAVAŠ, J., 2024. ENHANCING SALES EFFICIENCY WITH THE USE OF BUSINESS INTELLIGENCE AND ANALYTICS IN A PUBLIC COMPANY. *Poslovna izvrsnost - Business excellence*, vol. 18, no. 1, pp. 39-55. ISSN 18463355. DOI 10.22598/pi-be/2024.18.1.39.
- PRADHAN, R. y DASH, A., 2020. *An Overview of Microservices*. Singapore: Springer. vol. 601.
- PRINCIPE, I. y PARIASCA, G., 2018. *Desarrollo e implementación de un sistema web para la gestión de ventas de la empresa Factory Solutions*. S.l.: Universidad Autónoma del Perú.
- PUBLICACIONES VÉRTICE, 2008. *El proceso de venta*. Málaga: Editorial Vértice. ISBN 978-84-92533-17-6.
- PUERTA CORREDOR, A.L., BARRERA LASSO, M.A. y GARNICA ESTRADA, E., 2025. Identificación de vulnerabilidades basados en programación: buenas prácticas de programación para sistemas informáticos más seguros. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, vol. 12, no. 23, pp. 121-137. ISSN 2357-3716. DOI 10.21017/rimci.1132.
- RAJAPACK, 2024. Las 5+1 fases del proceso de gestión de pedidos (y cómo optimizarlas). *RajaPack* [en línea]. [consulta: 24 octubre 2025]. Disponible en: <https://www.raja-pack.es/blog-es/fases-proceso-gestion-pedidos>.



- RODRIGUEZ, M. y BOYER, S., 2020. The impact of mobile customer relationship management (mCRM) on sales collaboration and sales performance. *Journal of Marketing Analytics*, vol. 8, no. 3, pp. 137-148. ISSN 2050-3318. DOI 10.1057/s41270-020-00087-3.
- ROJAS LORENZO, J.N., 2021. *Sistema informático bajo plataforma web para la mejora del proceso de ventas de la empresa 4J y M SAC* [en línea]. Trabajo de suficiencia profesional. Lima: Universidad Privada del Norte. [consulta: 18 octubre 2025]. Disponible en: <https://hdl.handle.net/11537/25516>.
- ROMERO, J.A., LA PAZ, A. y GARCÍA, M.J., 2024. Impacto de las herramientas tecnológicas en las ventas de empresas chilenas. *Información tecnológica*, vol. 35, no. 2, pp. 11-22. ISSN 0718-0764. DOI 10.4067/s0718-07642024000200011.
- SALEH, M. y ABEL, M.-H., 2015. Information Systems: Towards a System of Information Systems. *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* [en línea]. S.l.: SCITEPRESS - Science and Technology Publications, pp. 193-200. vol. 3. ISBN 978-989-758-158-8. DOI 10.5220/0005596101930200. Disponible en: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005596101930200>.
- SAURO, J., 2018. 5 Ways to Interpret a SUS Score. *Measuringu* [en línea]. [consulta: 22 junio 2025]. Disponible en: <https://measuringu.com/interpret-sus-score/>.
- SERRANO, F., 2023. *Sistema web para la mejora del proceso de atención al cliente en el restaurant Los Angeles, El Tambo – Huancayo, 2023*. S.l.: Universidad Nacional del Centro del Perú.
- SEVILLA-GONZALEZ, M.D.R., MORENO LOAEZA, L., LAZARO-CARRERA, L.S., BOURGUET RAMIREZ, B., VÁZQUEZ RODRÍGUEZ, A., PERALTA-PEDRERO, M.L. y ALMEDA-VALDES, P., 2020. Spanish Version of the System Usability Scale for the Assessment of Electronic Tools: Development and Validation. *JMIR Human Factors*, vol. 7, no. 4, pp. e21161. ISSN 2292-9495. DOI 10.2196/21161.
- SHAH, H. y SOOMRO, T.R., 2017. Node.js Challenges in Implementation. *Global Journal of Computer Science and Technology: E Network, Web & Security*, vol. 17, no. 2,



- SINGH, S., KUMAR, R. y GUPTA, A., 2023. Impact of Omnichannel Integration on Order Fulfillment Efficiency and Customer Satisfaction in Retail. *Journal of Retailing and Consumer Services*, vol. 71,
- SMITH, J.D., 2024. The Impact of Technology on Sales Performance in B2B Companies. *Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023*, vol. 3, no. 1, pp. 86-102. ISSN 3006-4023. DOI 10.60087/jaigs.vol03.issue01.p102.
- SOALHEIRA, J. y TIMBRELL, G., 2014. What is shared services? *Advanced Series in Management*, vol. 13, pp. 67-84. ISSN 18776361. DOI 10.1108/S1877-636120140000013004.
- SUBECZ, Z., 2021. WEB-DEVELOPMENT WITH LARAVEL FRAMEWORK. *Gradus*, vol. 8, no. 1, pp. 211-218. ISSN 2064-8014.
- SUCCESS UGBAJA, U., STELLA NWABEKEE, U., OSEREMEN OWOBURU, W. y ANTHONY ABIEBA, O., 2024. The Impact of AI and Business Process Automation on Sales Efficiency and Customer Relationship Management (CRM) Performance. *International Journal of Advanced Multidisciplinary Research and Studies*, vol. 4, no. 6, pp. 1829-1841. ISSN 2583049X. DOI 10.62225/2583049X.2024.4.6.4156.
- SUDA, B., 2003. *SOAP Web Services* [en línea]. Tesis de Maestría. Edinburgh: University of Edinburgh. [consulta: 25 octubre 2025]. Disponible en: <https://suda.co.uk/publications/MSc/brian.suda.thesis.pdf>.
- TASHILDAR, A., SHAH, N., GALA, R., GIRI, T. y CHAVHAN, P., 2020. APPLICATION DEVELOPMENT USING FLUTTER. *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, no. 8,
- TECHTARGET, 2024. What are shared services? *TechTarget* [en línea]. [consulta: 24 octubre 2025]. Disponible en: <https://www.techtarget.com/searchcio/definition/Shared-services>.
- TRIANGLE, 2023. What is cross platform software? [en línea]. Barcelona: [consulta: 19 octubre 2025]. Disponible en: <https://www.triangle.es/en/news/what-is-cross-platform-software/>.
- UIFROMMARS, 2020. Medir la usabilidad con el Sistema de Escalas de Usabilidad (SUS). .
- UKPABI, K.H. y IBRAHIM, A.M., 2024. EXPLORING OPERATING SYSTEM DIVERSITY: A COMPARATIVE ANALYSIS OF WINDOWS, MAC OS, ANDROID AND IOS. *Systematic and Modern Science Research (JSMSR)* [en línea], vol.



- 5, no. 9, pp. 23-40. [consulta: 25 octubre 2025]. Disponible en: <https://berkeleypublications.com/bjsmsr/article/view/264>.
- URATHAL, U., VINODHINI, P. y SASIKALA, V., 2021. AN INTERPRETATION OF DART PROGRAMMING LANGUAGE. *Dogo Rangsang Research Journal*, vol. 11, no. 10, pp. 144-149.
- UXABLES, 2024. Medir con el sistema de escala de usabilidad (SUS). .
- VARGAS, J., 2017. *Sistema web para el proceso de venta en la empresa Calzatec E.I.R.L.* S.l.: Universidad César Vallejo.
- VELICA YOYELLIA, KUNCORO, D.K.R. y YANDRA ARKEMAN, 2024. DESAIN SISTEM INFORMASI RANTAI PASOK TERINTEGRASI BERBASIS ENTERPRISE RESOURCE PLANNING ODOO PENJUALAN. *Industry Xplore*, vol. 9, no. 2, pp. 446-455. ISSN 2580-5479. DOI 10.36805/teknikindustri.v9i2.8031.
- W3C, 2013. Extensible Markup Language (XML) 1.0 (Fifth Edition). *W3C Recommendation* [en línea]. [consulta: 25 octubre 2025]. Disponible en: <https://www.w3.org/TR/xml/>.
- WEARETESTERS, 2025. Escala de usabilidad SUS. ¿Cómo usarla para mejorar tu UX? .
- ZAPATA, L., MORENO, A.M. y FERNANDEZ-MEDINA, E., 2013. Is Usability an Obstacle for Information Systems Security? *Proceedings of the 10th International Workshop on Security in Information Systems*. S.l.: SciTePress - Science and Technology Publications, pp. 53-65. ISBN 978-989-8565-64-8. DOI 10.5220/0004602300530065.
- ZHAO, B., 2017. Web Scraping. *Encyclopedia of Big Data* [en línea]. Cham: Springer International Publishing, pp. 1-3. Disponible en: http://link.springer.com/10.1007/978-3-319-32001-4_483-1.



ANEXOS



Anexo A

Tabla 26 — Matriz de consistencia

Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024

Problemas	Objetivos	Hipótesis	Variables	Dimensiones	Indicadores	Metodología
¿En qué medida el sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024?	Desarrollar e implementar un sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.	Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de venta en empresas usuarias de “Rumisof-pos”, 2024.	Variable independiente Sistema de información de servicios compartidos multiplataforma	Usabilidad	Eficacia	Tipo Aplicada Nivel Explicativo Diseño Preexperimental
					Eficiencia	
					Satisfacción	
				Seguridad	Autenticación	
					Inyección	
					Validación	
Problemas específicos	Objetivos específicos	Hipótesis específicas	Variable dependiente			Población 47 empresas usuarias del sistema de servicios compartidos multiplataforma “Rumisof-pos” Muestra Se tomó la totalidad de la población, 47 empresas. Estadístico de investigación Análisis no paramétrico para la diferencia de medias Wilcoxon
¿En qué medida el sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024? ¿En qué medida el sistema de información de servicios compartidos multiplataforma mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024?	- Desarrollar e implementar un sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024. - Desarrollar e implementar un sistema de información de servicios compartido multiplataforma para mejorar el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.	- Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de proceso de órdenes en empresas usuarias de “Rumisof-pos”, 2024. - Al desarrollar e implementar un sistema de información de servicios compartidos multiplataforma mejora el tiempo de la generación de facturas en empresas usuarias de “Rumisof-pos”, 2024.	Proceso de venta	- Proceso de órdenes - Proceso de facturación	- Tiempo de proceso de órdenes - Tiempo de proceso de facturación	

Anexo B

Cuestionario de Usabilidad

Nombre de la Empresa:

FERRETERIA LA SOLUCIÓN

RUC:

10477931508

⚠ Debes responder todas las preguntas antes de enviar.

Pregunta	1. Totalmente en desacuerdo	2. En desacuerdo	3. Neutro	4. De acuerdo	5. Totalmente de acuerdo
1. Creo que me gustaría utilizar este sistema con frecuencia	☐	☐	☐	☐	☐
2. Encontré el sistema innecesariamente complejo	☐	☐	☐	☐	☐
3. Pensé que el sistema era fácil de usar	☐	☐	☐	☐	☐
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema	☐	☐	☐	☐	☐
5. Encontré que las diversas funciones de este sistema estaban bien integradas	☐	☐	☐	☐	☐
6. Pensé que había demasiada inconsistencia en este sistema	☐	☐	☐	☐	☐
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente	☐	☐	☐	☐	☐
8. Encontré el sistema muy complicado de usar	☐	☐	☐	☐	☐
9. Me sentí muy seguro usando el sistema	☐	☐	☐	☐	☐
10. Necesitaba aprender muchas cosas antes de empezar con este sistema	☐	☐	☐	☐	☐

Enviar Respuestas

Figura 33 — Instrumento para medir la usabilidad

Anexo C

Situación Antes de Implementar Rumisof-POS

¿Qué sistema utilizaba su empresa antes de implementar Rumisof-POS?

- ☐ Sistema propio
- ☐ Sistema de escritorio
- ☒ Sin sistema (procesos manuales)
- ☐ Otro

¿En cuántos minutos, en promedio, tomaba el proceso de venta y/o pedido de orden desde que el cliente solicitaba el producto/servicio hasta que se completaba la transacción?

4 minutos

¿En cuántos minutos, en promedio, tomaba la emisión del comprobante electrónico (factura/boleta)?

2 minutos

¿Cuáles eran las principales dificultades que enfrentaba en el proceso de venta? (Seleccione todas las que apliquen)

- ☐ Sistema lento
- ☐ Falta de soporte técnico
- ☐ Problemas de compatibilidad con otros sistemas operativos
- ☒ Necesidad de realizar procesos manuales

Anterior

Siguiente

Figura 34 — Instrumento para la prepueba de la variable dependiente

Anexo D

Formatos de validación por juicio de expertos

FICHA DE VALIDEZ DEL INSTRUMENTO

I. DATOS GENERALES

- 1.1. Apellidos y nombres del experto: González Ruíz, Manuel Francisco
- 1.2. Grado académico y especialidad: Ingeniero de Sistemas
- 1.3. Título de la investigación: Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de "Rumisof-pos", 2024
- 1.4. Autor del instrumento: Mario Choquetaipe Asto
- 1.5. Nombre del instrumento: "Cuestionario de medición pretest de variable dependiente Proceso de venta"
- 1.6. Objetivo del Instrumento: Medir la variable Proceso de venta antes de la implementación del sistema de información Rumisof-pos
- 1.7. Variable de Estudio: Proceso de Venta
- 1.8. Dimensiones de la Variable: Proceso de órdenes y Proceso de facturación

Instrucciones: A continuación, se le solicita evaluar cada ítem del cuestionario según los siguientes tres criterios: Pertinencia, Claridad y Coherencia, escribiendo el número (1, 2 o 3) que mejor represente su opinión en la casilla correspondiente de la matriz.

Escala de valoración		
1	2	3
No cumple	Cumple parcialmente	Cumple totalmente

Nº	DIMENSIONES / ítems	Pertinencia ¹	Claridad ²	Coherencia ³	Observaciones
	Variable: Proceso de venta	(1-3)	(1-3)	(1-3)	
	DIMENSION: Ninguna es de tipo contextual				
1.	¿Qué sistema utilizaba su empresa antes de implementar Rumisof-POS?	3	3	3	
	DIMENSION: Proceso de órdenes				
2.	¿En cuántos minutos, en promedio, tomaba el proceso de venta y/o pedido de orden desde que el cliente solicitaba el producto/servicio hasta que se completaba la transacción?	3	3	3	
	DIMENSION: Proceso de facturación				
3.	¿En cuántos minutos, en promedio, tomaba la emisión del comprobante electrónico (factura/boleto)?	3	3	3	
	DIMENSION: Ninguna es de tipo contextual				
4.	¿Cuáles eran las principales dificultades que enfrentaba en el proceso de venta? (Seleccione todas las que apliquen)	3	3	3	

Observaciones (precisar si hay Suficiencia⁴): Sí, existe suficiencia

Opinión de aplicabilidad: Aplicable [x] Aplicable después de corregir [] No aplicable []

Grado, Apellidos y Nombres del juez validador:

Ingeniero de Sistemas, González Ruíz Manuel Francisco. CIP 126865.

Especialidad del juez validador:

Fecha: 03 de diciembre del 2024.

Firma del experto informante

DNI: 40704575

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Claridad: El ítem está redactado de forma clara, precisa y sin ambigüedades, facilitando su comprensión.

³Coherencia: El ítem tiene una relación lógica y consistente con la dimensión y la variable de estudio.

⁴Suficiencia: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

FICHA DE VALIDEZ DEL INSTRUMENTO

I. DATOS GENERALES

- 1.1. Apellidos y nombres del experto: Juan Carlos Tello Freitas
- 1.2. Grado académico y especialidad: Ingeniero de Sistemas
- 1.3. Título de la investigación: Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de "Rumisof-pos", 2024
- 1.4. Autor del instrumento: Mario Choquetaipe Asto
- 1.5. Nombre del instrumento: "Cuestionario de medición pretest de variable dependiente Proceso de venta"
- 1.6. Objetivo del Instrumento: Medir la variable Proceso de venta antes de la implementación del sistema de información Rumisof-pos
- 1.7. Variable de Estudio: Proceso de Venta
- 1.8. Dimensiones de la Variable: Proceso de órdenes y Proceso de facturación

Instrucciones: A continuación, se le solicita evaluar cada ítem del cuestionario según los siguientes tres criterios: Pertinencia, Claridad y Coherencia, escribiendo el número (1, 2 o 3) que mejor represente su opinión en la casilla correspondiente de la matriz.

Escala de valoración		
1	2	3
No cumple	Cumple parcialmente	Cumple totalmente

N°	DIMENSIONES / ítems	Pertinencia ¹ (1-3)	Claridad ² (1-3)	Coherencia ³ (1-3)	Observaciones
	Variable: Proceso de venta				
	DIMENSION: Ninguna es de tipo contextual				
1.	¿Qué sistema utilizaba su empresa antes de implementar Rumisof-POS?	2	3	3	
	DIMENSION: Proceso de órdenes				
2.	¿En cuántos minutos, en promedio, tomaba el proceso de venta y/o pedido de orden desde que el cliente solicitaba el producto/servicio hasta que se completaba la transacción?	3	2	3	
	DIMENSION: Proceso de facturación				
3.	¿En cuántos minutos, en promedio, tomaba la emisión del comprobante electrónico (factura/boleta)?	3	2	3	
	DIMENSION: Ninguna es de tipo contextual				
4.	¿Cuáles eran las principales dificultades que enfrentaba en el proceso de venta? (Seleccione todas las que apliquen)	3	3	3	

Observaciones (precisar si hay Suficiencia⁴):

Si es suficiente

Opinión de aplicabilidad: Aplicable [x] Aplicable después de corregir [] No aplicable []

Grado, Apellidos y Nombres del juez validador: Juan Carlos Tello Freitas

Especialidad del juez validador: Ingeniero de Sistemas

Fecha: 02 de diciembre del 2024.

Firma del experto informante

DNI: 40692491

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Claridad: El ítem está redactado de forma clara, precisa y sin ambigüedades, facilitando su comprensión.

³Coherencia: El ítem tiene una relación lógica y consistente con la dimensión y la variable de estudio.

⁴Suficiencia: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

FICHA DE VALIDEZ DEL INSTRUMENTO

I. DATOS GENERALES

- 1.1. Apellidos y nombres del experto: Juan Carlos Cosme Lopez
- 1.2. Grado académico y especialidad: Título de Ing. de sistemas
- 1.3. Título de la investigación: Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de "Rumisof-pos", 2024
- 1.4. Autor del instrumento: Mario Choquetaipe Asto
- 1.5. Nombre del instrumento: "Cuestionario de medición pretest de variable dependiente Proceso de venta"
- 1.6. Objetivo del Instrumento: Medir la variable Proceso de venta antes de la implementación del sistema de información Rumisof-pos
- 1.7. Variable de Estudio: Proceso de Venta
- 1.8. Dimensiones de la Variable: Proceso de órdenes y Proceso de facturación

Instrucciones: A continuación, se le solicita evaluar cada ítem del cuestionario según los siguientes tres criterios: Pertinencia, Claridad y Coherencia, escribiendo el número (1, 2 o 3) que mejor represente su opinión en la casilla correspondiente de la matriz.

Escala de valoración		
1	2	3
No cumple	Cumple parcialmente	Cumple totalmente

Nº	DIMENSIONES / ítems	Pertinencia ¹	Claridad ²	Coherencia ³	Observaciones
	Variable: Proceso de venta	(1-3)	(1-3)	(1-3)	
	DIMENSION: Ninguna es de tipo contextual				
1.	¿Qué sistema utilizaba su empresa antes de implementar Rumisof-POS?	3	3	3	
	DIMENSION: Proceso de órdenes				
2.	¿En cuántos minutos, en promedio, tomaba el proceso de venta y/o pedido de orden desde que el cliente solicitaba el producto/servicio hasta que se completaba la transacción?	3	3	3	
	DIMENSION: Proceso de facturación				
3.	¿En cuántos minutos, en promedio, tomaba la emisión del comprobante electrónico (factura/boleto)?	3	3	3	
	DIMENSION: Ninguna es de tipo contextual				
4.	¿Cuáles eran las principales dificultades que enfrentaba en el proceso de venta? (Seleccione todas las que apliquen)	2	3	3	

Observaciones (precisar si hay Suficiencia⁴): __Si existe suficiencia en los

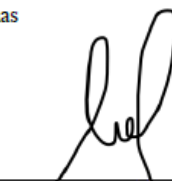
ítems _____

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Grado, Apellidos y Nombres del juez validador: Juan Carlos Cosme Lopez

Especialidad del juez validador: Ing. De Sistemas

Fecha: 03 de diciembre del 2024.



 Firma del experto informante

DNI:

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Claridad: El ítem está redactado de forma clara, precisa y sin ambigüedades, facilitando su comprensión.

³Coherencia: El ítem tiene una relación lógica y consistente con la dimensión y la variable de estudio.

⁴Suficiencia: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

FICHA DE VALIDEZ DEL INSTRUMENTO**I. DATOS GENERALES**

- 1.1. Apellidos y nombres del experto: Segundo Atilano Vigo Contreras
- 1.2. Grado académico y especialidad: Ing. De Sistemas, con especialización en mejora de procesos en red local y cloud(AWS).
- 1.3. Título de la investigación: Sistema de información de servicios compartidos multiplataforma para mejorar el tiempo de proceso de venta en empresas usuarias de "Rumisof-pos", 2024
- 1.4. Autor del instrumento: Mario Choquetaipe Asto
- 1.5. Nombre del instrumento: "Cuestionario de medición pretest de variable dependiente Proceso de venta"
- 1.6. Objetivo del Instrumento: Medir la variable Proceso de venta antes de la implementación del sistema de información Rumisof-pos
- 1.7. Variable de Estudio: Proceso de Venta
- 1.8. Dimensiones de la Variable: Proceso de órdenes y Proceso de facturación

Instrucciones: A continuación, se le solicita evaluar cada ítem del cuestionario según los siguientes tres criterios: Pertinencia, Claridad y Coherencia, escribiendo el número (1, 2 o 3) que mejor represente su opinión en la casilla correspondiente de la matriz.

Escala de valoración		
1	2	3
No cumple	Cumple parcialmente	Cumple totalmente

Nº	DIMENSIONES / ítems	Pertinencia ¹ (1-3)	Claridad ² (1-3)	Coherencia ³ (1-3)	Observaciones
	Variable: Proceso de venta				
	DIMENSIÓN: Ninguna es de tipo contextual				
1.	¿Qué sistema utilizaba su empresa antes de implementar Rumisof-POS?	3	3	3	
	DIMENSIÓN: Proceso de órdenes				
2.	¿En cuántos minutos, en promedio, tomaba el proceso de venta y/o pedido de orden desde que el cliente solicitaba el producto/servicio hasta que se completaba la transacción?	3	3	3	
	DIMENSIÓN: Proceso de facturación				
3.	¿En cuántos minutos, en promedio, tomaba la emisión del comprobante electrónico (factura/boleta)?	3	3	3	
	DIMENSIÓN: Ninguna es de tipo contextual				
4.	¿Cuáles eran las principales dificultades que enfrentaba en el proceso de venta? (Seleccione todas las que apliquen)	3	3	3	

Observaciones (precisar si hay Suficiencia⁴): ____ Si hay

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Grado, Apellidos y Nombres del juez validador: Segundo Atilano Vigo Contreras

Especialidad del juez validador: Especialización en mejora de procesos en red local y cloud (AWS)

Fecha: 04 de diciembre del 2024.

Firma del experto informante

DNI: 08161326

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Claridad: El ítem está redactado de forma clara, precisa y sin ambigüedades, facilitando su comprensión.

³Coherencia: El ítem tiene una relación lógica y consistente con la dimensión y la variable de estudio.

⁴Suficiencia: Se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

Anexo E

Tabla 27 — Matriz de datos de la dimensión usabilidad

#	Razón Social	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Puntaje	Categoría	
1	MANGAL MANCILLA VALENCIA SONIA YSABEL	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
2	BOTICA VENDABAL	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
3	CAFEERIA FORMARIA LINCE	5	1	4	1	5	1	5	1	5	3	92.5	1	Aceptable
4	FERETERIA LA SOLUCION	5	2	5	1	5	2	5	1	3	2	87.5	1	Aceptable
5	CONTRERAS SOLORZANO PAVEL	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
6	EL CANAZTON	5	1	4	1	5	1	5	3	5	2	90	1	Aceptable
7	LUBRICENTRO	4	1	5	1	5	1	5	1	5	3	92.5	1	Aceptable
8	OKLIN LAVANDERIA E.I.R.L.	3	1	3	1	5	1	5	1	5	1	90	1	Aceptable
9	PASTOR CACERES PAULINO- PIÑATERIA	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
10	POLLO LOKO INVERSIONES EMPRESA INDIVID	5	1	5	2	5	1	3	1	5	2	90	1	Aceptable
11	VALARAM	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
12	LOPEZ ANTICONA RONAL NILSON	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
13	BALDEON DE LA TORRE JHOBERT	5	1	5	1	4	3	5	1	5	2	90	1	Aceptable
14	ZUMBAO_COMAS	5	1	5	1	5	1	5	1	3	1	95	1	Aceptable
15	MEZA CAMPOS RONALD DIEGO	5	1	5	1	5	2	5	2	5	1	95	1	Aceptable
16	CARBAJAL LLOMERO JOEL JOSIAS	5	1	5	3	4	1	4	1	3	1	85	1	Aceptable
17	DIAZ CENAS JESUS JAVIER	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
18	WOKA	5	1	5	1	3	1	5	1	5	1	95	1	Aceptable
19	RODRIGUEZ TINCOPA MITCHEL ANTONIO	5	1	5	1	5	1	4	1	4	1	95	1	Aceptable
20	PEREZ POMATAY GEORGE PAUL	5	1	5	1	4	1	4	3	5	2	87.5	1	Aceptable
21	TORRES ROSILLO FRANCESCA DANAE	3	1	5	2	5	1	5	1	3	1	87.5	1	Aceptable
22	ROJAS ROJAS EDSON EMMANUEL	3	1	5	1	5	1	5	1	3	1	90	1	Aceptable
23	ALVAREZ NARVAEZ SANTOS PEDRO	5	1	5	3	5	1	4	1	5	1	92.5	1	Aceptable
24	AGROSERVICIOS AYNÍ S.A.C.	5	3	5	3	5	1	5	1	5	3	85	1	Aceptable
25	CENTRO DE INFORMÁTICA Y SISTEMA ON LIN	5	1	5	1	5	3	4	1	5	1	92.5	1	Aceptable
26	DON LUCHO NUTRICION ANIMAL E.I.R.L.	5	1	5	1	5	1	5	2	5	1	97.5	1	Aceptable
27	INDUSTRIAS ELIAL P&O S.A.C.	4	2	5	1	5	1	5	1	5	1	95	1	Aceptable
28	TRANSPORTE & MAQUINARIA JULITA S.A.C.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
29	MIJASEB E.I.R.L.	5	2	5	1	5	1	5	1	5	1	97.5	1	Aceptable
30	CHILENO E HIJOS E.I.R.L.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
31	MILKICA S.A.C.	3	1	5	1	5	2	5	1	4	1	90	1	Aceptable
32	ELECTRIC THIALYM E.I.R.L.	5	3	3	1	5	2	5	1	5	1	87.5	1	Aceptable
33	FENIX MOBILIARIO S.A.C.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
34	CAPSIDRY S.A.C.	5	1	5	1	3	2	5	1	3	1	87.5	1	Aceptable
35	MUNDO MEDIA NETWORK S.A.C.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
36	CONDUCCION DE ENERGIA	4	2	5	2	4	1	5	1	5	1	90	1	Aceptable
37	AMAZON PROJECT INGENIERIA Y ARQUITECTU	3	3	5	1	5	1	5	1	5	1	90	1	Aceptable
38	GOLDEN DOOR S.A.C.	5	1	5	1	5	2	5	1	3	2	90	1	Aceptable
39	CONSORCIO SUPERVISOR HOSPITALARIO HUA	5	2	5	1	5	1	5	1	4	1	95	1	Aceptable
40	INVERSIONES HACHIN & TRANSPORTE E.I.R.L.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
41	CONSTRUCTORA E INMOBILIARIA SP S.A.C.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
42	ELIZABETH SALON Y EVENTOS E.I.R.L.	5	1	4	1	4	1	5	1	3	1	90	1	Aceptable
43	FRENOS & SERVICIOS ROY EIRL	5	3	5	1	5	1	5	1	3	1	90	1	Aceptable
44	IMPORTADORA IMOVA S.A.C.	5	1	5	1	5	1	5	1	5	1	100	1	Aceptable
45	FERTOR GROUP E.I.R.L.	5	1	5	2	3	3	5	1	5	1	87.5	1	Aceptable
46	ARROCES DE LA TIA LINA S.A.C.	5	1	5	1	5	3	5	1	5	1	95	1	Aceptable
47	J & M INVERSIONES KRISMAR MOTORS E.I.R.L	5	1	5	1	5	1	5	1	3	1	95	1	Aceptable

Anexo F

Tabla 28 — Matriz de datos de la dimensión seguridad

Categoría/Prueba	Ejec. 1 01/04	Ejec. 2 02/04	Ejec. 3 03/04	Ejec. 4 04/04	Ejec. 5 05/04	Ejec. 6 06/04	Ejec. 7 07/04	Ejec. 8 08/04	Ejec. 9 09/04	Ejec. 10 10/04	Ejec. 11 11/04	Ejec. 12 12/04	Ejec. 13 13/04	Ejec. 14 14/04	Ejec. 15 15/04
AUTENTICACIÓN															
OWASP-API1: Acceso sin credenciales	FAIL	FAIL	PASS	PASS	FAIL	PASS	PASS	PASS	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
OWASP-API2: Token JWT manipulado	FAIL	FAIL	FAIL	PASS	FAIL	PASS	PASS	PASS	FAIL	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API4: Fuerza bruta en login	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	FAIL	PASS	FAIL	PASS	PASS	PASS	PASS
OWASP-API6: Token expirado	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API1: Credenciales por defecto	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS
INYECCIÓN															
OWASP-API8: SQL Injection en parámetros	FAIL	PASS	FAIL	PASS	PASS	PASS	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API8: NoSQL Injection en body	FAIL	FAIL	FAIL	PASS	PASS	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API9: Inyección de comandos OS	PASS	FAIL	PASS	FAIL	PASS	PASS	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API4: XXE en XML inputs	PASS	FAIL	FAIL	FAIL	FAIL	PASS	PASS	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
OWASP-API8: Inyección LDAP	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API8: Inyección de plantillas	PASS	PASS	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS
VALIDACIÓN															
OWASP-API7: XSS en campos de entrada	FAIL	FAIL	FAIL	PASS	FAIL	PASS	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API5: CSRF token missing	PASS	FAIL	FAIL	PASS	PASS	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API10: Tamaño payload excesivo	FAIL	FAIL	FAIL	FAIL	PASS	PASS	FAIL	PASS	FAIL	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API8: Tipo de archivo malicioso	FAIL	PASS	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
OWASP-API7: Bypass de validación del lado c	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	FAIL	PASS	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS

Anexo G

Tabla 29 — Matriz de datos de la variable Proceso de venta: Pretest y postest

EMPRESA (Razón Social)	RUC	ANTES (minutos)			DESPUÉS (minutos)		
		Orden	Facturación	Venta	Orden	Facturación	Venta
MANGAL _ MANCILLA VALENCIA SONIA YSABEL	10400379012	2	2	4	1.43	0.61	2.04
BOTICA VENDABAL	10470738397	12	12	24	1.55	0.67	2.22
CAFEERIA FORMARIA LINCÉ	10484987101	2	2	4	1.75	0.74	2.49
FERETERIA LA SOLUCION	10082996911	22.5	22.5	45	2.1	0.89	2.99
CONTRERAS SOLORZANO PAVEL	10101122188	2	2	4	1.84	0.8	2.64
EL CANAZTON	10430131724	2	2	4	2.01	0.85	2.86
LUBRICENTRO	10805607233	2	2	4	1.86	0.8	2.66
OKLIN LAVANDERIA E.I.R.L.	20556693766	2	2	4	1.81	0.78	2.59
PASTOR CACERES PAULINO- PIÑATERIA	10480569119	8	8	16	1.77	0.75	2.52
POLLO LOKO INVERSIONES E.I.R.L	20608549961	2	2	4	1.9	0.81	2.71
VALARAM	20603266693	22.5	22.5	45	2.01	0.86	2.87
LOPEZ ANTICONA RONAL NILSON	10721754516	2	2	4	2.09	0.88	2.97
BALDEON DE LA TORRE JHOBERT	10722320757	2	2	4	2.09	0.89	2.98
ZUMBAO COMAS	10415207846	2	2	4	1.88	0.81	2.69
MEZA CAMPOS RONALD DIEGO	10726592174	8	8	16	2.09	0.9	2.99
CARBAJAL LLOMERO JOEL JOSIAS	10737432829	2	2	4	1.99	0.88	2.87
DIAZ CENAS JESUS JAVIER	10739303848	2	2	4	2.09	0.9	2.99
WOKA	20608215264	2	2	4	1.53	0.65	2.18
RODRIGUEZ TINCOPA MITCHEL ANTONIO	10746957357	2	2	4	2.05	0.88	2.93
PEREZ POMATAY GEORGE PAUL	10747347960	2	2	4	2.08	0.89	2.97
TORRES ROSILLO FRANCESCA DANAE	10755359209	2	2	4	2.06	0.9	2.96
ROJAS ROJAS EDSON EMMANUEL	10761916217	12	12	24	1.77	0.76	2.53
ALVAREZ NARVAEZ SANTOS PEDRO	10806350899	4	4	8	1.99	0.86	2.85
AGROSERVICIOS AYNÍ S.A.C.	20613179799	4	4	8	1.91	0.81	2.72
CENTRO DE INFORMÁTICA Y SISTEMA ON LINE EN TARAPOTO S.A.C	20613179888	4	4	8	2.05	0.89	2.94
DON LUCHO NUTRICION ANIMAL E.I.R.L.	20613180100	12	12	24	1.76	0.75	2.51
INDUSTRIAS ELIAL P& O S.A.C.	20613180215	4	4	8	1.71	0.73	2.44
TRANSPORTE & MAQUINARIA JULITA S.A.C.	20613180355	4	4	8	1.94	0.83	2.77
MIJASEB E.I.R.L.	20613180398	4	4	8	1.96	0.82	2.78
CHILENO E HIJOS E.I.R.L.	20613180495	4	4	8	2.08	0.9	2.98
MILKICA S.A.C.	20613180541	4	4	8	1.97	0.85	2.82
ELECTRIC THIALYM E.I.R.L.	20613180576	4	4	8	1.86	0.79	2.65
FENIX MOBILIARIO S.A.C.	20613180606	6	6	12	1.75	0.75	2.5
CAPSIDRY S.A.C.	20613180967	22.5	22.5	45	1.88	0.81	2.69
MUNDO MEDIA NETWORK S.A.C.	20613180975	6	6	12	2.1	0.89	2.99
CONDUCCION DE ENERGIA	20613180983	6	6	12	2.11	0.9	3.01
AMAZON PROJECT INGENIERIA Y ARQUITECTURA S.A.C.	20613180991	6	6	12	1.82	0.78	2.6
GOLDEN DOOR S.A.C.	20613181017	22.5	22.5	45	1.81	0.77	2.58
CONSORCIO SUPERVISOR HOSPITALARIO HUARAZ	20613181033	2	2	4	1.98	0.84	2.82
INVERSIONES HACHIN & TRANSPORTE E.I.R.L.	20613181050	22.5	22.5	45	2.07	0.86	2.93
CONSTRUCTORA E INMOBILIARIA SP S.A.C.	20613181068	2	2	4	1.95	0.84	2.79
ELIZABETH SALON Y EVENTOS E.I.R.L.	20613181076	2	2	4	1.92	0.83	2.75
FRENOS & SERVICIOS ROY EIRL	20613181084	4	4	8	1.88	0.81	2.69
IMPORTADORA IMOVA S.A.C.	20613181092	4	4	8	1.65	0.71	2.36
FERTOR GROUP E.I.R.L.	20613181114	2	2	4	2.04	0.87	2.91
ARROCES DE LA TIA LINA S.A.C.	20613181149	2	2	4	1.96	0.85	2.81
J & M INVERSIONES KRISMAR MOTORS E.I.R.L.	20613181165	2	2	4	1.93	0.82	2.75
PROMEDIOS		5.88	5.88	11.77	1.91	0.82	2.73