



Escuela de Administración de Tecnologías de Información

Propuesta de diseño de software de alto nivel para la interoperabilidad de las funcionalidades actuales de Hortiluma

Trabajo Final de Graduación para optar al grado de Licenciatura en Administración de Tecnologías de Información

Proyecto de Graduación

Elaborado por: Luis Sebastian Hernández Venegas

Prof. tutor: Ing. Luis Javier Chavarría Sánchez

Cartago, Costa Rica

II semestre

Noviembre, 2024



Esta obra está sujeta a la licencia **Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional** de Creative Commons. Para ver una copia de esta licencia, visite <https://creativecommons.org/licenses/by-nc-sa/4.0/>

San José, 1 de noviembre de 2024

Señores(as)

Instituto Tecnológico de Costa Rica

Estimados señores(as):

Yo, María Fernanda Sanabria Coto, cédula de identidad 114290780, bachiller en Filología española graduada en la Universidad de Costa Rica, perteneciente a la Asociación Costarricense de Filólogos (ACFIL), carné 225 y al Colegio de Licenciados y Profesores en Letras, Filosofía, Ciencias y Artes de Costa Rica (COLYPRO), código 75402, hago constar que he revisado el documento titulado:

Propuesta de diseño de software de alto nivel para la interoperabilidad de las funcionalidades actuales de Hortiluma

Dicho documento fue elaborado por Luis Sebastián Hernández Venegas, cédula de identidad 118110693, con el fin de optar al grado de Licenciatura en Administración de Tecnologías de Información. He revisado y corregido aspectos tales como construcción de párrafos, vicios del lenguaje trasladados a lo escrito, ortografía, puntuación y otros relacionados con el campo filológico.

Atentamente,

Fernanda S. Coto



MARIA
FERNANDA
SANABRIA
COTO
(FIRMA)

Firmado digitalmente por MARIA FERNANDA SANABRIA COTO (FIRMA)
Nombre de reconocimiento (DN): serialNumber=CPF-01-1429-0780, sn=SANABRIA COTO, givenName=MARIA FERNANDA, c=CR, o=PERSONA FISICA, ou=CIUDADANO, cn=MARIA FERNANDA SANABRIA COTO (FIRMA)
Motivo: Revisión filológica
Ubicación: Costa Rica
Fecha: 2024.11.01 19:17:15 -06'00'

María Fernanda Sanabria Coto

Filóloga

Asociación Costarricense de Filólogos. Carné nro. 225

Colypro. Código 75402

fernanda.sanabria@filologos.cr

Teléfono: +506 6022 9569

Hoja de Aprobación

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESHUELA DE ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN
GRADO ACADÉMICO: LICENCIATURA

Los miembros del Tribunal Examinador de la Escuela de Administración de Tecnologías de Información, recomendamos que el siguiente informe del Trabajo Final de Graduación (TFG) del estudiante Luis Sebastian Hernández Venegas sea aceptado como requisito parcial para obtener el grado académico de Licenciatura de Tecnología de Información.

Ing. Luis Javier Chavarría Sánchez
Profesor Tutor

Benjamín Calvo de León
Lector externo

Agustín Francesa Alfaro
Lector académico

Yarima Sandoval Sánchez
Coordinadora de Trabajo Final de Graduación

Dedicatoria

Este trabajo final de graduación se lo quiero dedicar
a mi familia que me ha apoyado durante todo este tiempo.

A mi profesor tutor, Luis Javier Chavarría Sánchez, que me guio durante el TFG,
y a todas aquellas personas que fueron parte de mi proceso.

Resumen

Hernández Venegas, Luis Sebastian (2024). “Propuesta de diseño de software de alto nivel para la interoperabilidad de las funcionalidades actuales de Hortiluma”. [Trabajo Final de Graduación, Tecnológico de Costa Rica].

Este proyecto se centra en el diseño de una arquitectura de *software* de alto nivel que facilite la interoperabilidad de las diversas funcionalidades de los sistemas de información existentes en Hortiluma. La propuesta busca integrar de manera eficaz los diferentes módulos de la empresa, proporcionando un acceso uniforme y actualizado a la información para mejorar la eficiencia operativa, reducir la duplicación de tareas, minimizar los errores en la gestión de datos y establecer una plataforma más coherente para la toma de decisiones fundamentadas.

La metodología empleada en este estudio es de tipo aplicada, con un enfoque cualitativo y un alcance descriptivo, lo cual permite un análisis detallado de las necesidades de interoperabilidad en un contexto específico. Se recurrió a diversas técnicas para la recolección de datos, entre ellas, entrevistas a los colaboradores, observación directa de los procesos operativos y análisis documental de las funcionalidades y limitaciones actuales de los sistemas. Adicionalmente, se realizaron grupos focales, sesiones de lluvia de ideas para la identificación de mejoras, la definición de historias de usuario específicos y la creación de una matriz de cobertura que garantiza el cumplimiento de los requisitos de interoperabilidad identificados.

Los resultados del estudio evidenciaron que la falta de integración entre los sistemas de Hortiluma ha generado problemas de eficiencia, como duplicación de tareas y errores en los datos, lo cual impacta negativamente tanto a empleados como a clientes y proveedores. La propuesta de diseño presentada en este proyecto se enfoca en solucionar estas deficiencias mediante la creación de una arquitectura que centralice la información, mejore la comunicación entre los sistemas y facilite la actualización y el acceso a los datos.

Finalmente, se recomienda proceder con la implementación de la arquitectura de *software* propuesta para mejorar la gestión de la información y mejorar la competitividad de Hortiluma en el mercado agroindustrial. Esto le permitirá a la empresa enfrentar retos de un entorno cada vez más dinámico y competitivo, adaptándose mejor a las necesidades de sus clientes y mejorando la colaboración interna.

Palabras clave: interoperabilidad, diseño de *software*, integración de sistemas, gestión de datos, eficiencia operativa, arquitectura de *software*, análisis cualitativo.

Abstract

Hernández Venegas, Luis Sebastian (2024). “High-Level Software Design Proposal for the Interoperability of Hortiluma's Current Functionalities.” [Final Graduation Project, Technological Institute of Costa Rica].

This project focuses on designing a high-level software architecture that facilitates the interoperability of various functionalities across Hortiluma's existing information systems. The proposal aims to effectively integrate the company's different modules, providing more consistent and updated access to information to enhance operational efficiency, reduce task duplication, minimize data management errors, and establish a more coherent platform for data-driven decision-making.

The methodology employed in this study is applied, with a qualitative approach and a descriptive scope, allowing for a detailed analysis of interoperability needs within a specific context. Various data collection techniques were utilized, including interviews with staff, direct observation of operational processes, and documentary analysis of current system functionalities and limitations. Additionally, focus groups, brainstorming sessions for identifying improvements, the definition of specific use cases, and the creation of a coverage matrix were carried out to ensure that identified interoperability requirements are met.

The study's findings revealed that the lack of integration between Hortiluma's systems has led to significant efficiency problems, such as task duplication and data errors, negatively impacting employees, customers, and suppliers. The design proposal presented in this project aims to address these deficiencies by creating an architecture that centralizes information, improves communication between systems, and facilitates data updating and access.

It is recommended to proceed with the implementation of the proposed software architecture to improve information management and enhance Hortiluma's competitiveness in the agro-industrial market. This will enable the company to more efficiently navigate the challenges of an increasingly dynamic and competitive environment, better meeting customer needs and improving internal collaboration.

Keywords: interoperability, software design, systems integration, data management, operational efficiency, software architecture, qualitative analysis.

Tabla de contenidos

Tabla de contenidos.....	iv
Índice de tablas.....	x
Índice de figuras.....	xi
1. Introducción.....	1
1.1. Descripción general.....	1
1.2. Antecedentes.....	2
1.2.1. Descripción de la organización.....	2
1.2.2. La organización.....	2
1.2.3. Trabajos similares realizados dentro y fuera de la organización.....	4
1.3. Planteamiento del problema.....	6
1.3.1. Situación problemática.....	6
1.3.2. Justificación del proyecto.....	8
1.3.3. Beneficios esperados o aportes del TFG.....	9
1.4. Objetivos.....	10
1.4.1. Objetivo general.....	10
1.4.2. Objetivos específicos.....	10
1.5. Alcance.....	11
1.5.1. Supuestos del proyecto.....	11
1.6. Entregables del proyecto.....	11
1.6.1. Entregables académicos.....	12
1.6.2. Entregables para la empresa.....	12
1.7. Gestión del proyecto.....	13
1.7.1. Cronograma.....	13
1.7.2. Minutas.....	14
1.7.3. Gestión del cambio.....	14
1.8. Exclusiones del proyecto.....	14
1.9. Limitaciones del proyecto.....	14
2. Marco conceptual.....	15
2.1. Proceso de <i>software</i>	15
2.1.1. Especificación del <i>software</i>	15
2.1.2. Levantamiento de requerimientos.....	15
2.1.3. Tipos de requerimientos.....	16
2.1.4. Arquitectura de <i>software</i> para la interoperabilidad.....	16
2.1.5. Procesamiento de requerimientos y priorización.....	16
2.1.6. Integración de sistemas y verificación de interoperabilidad.....	18

2.1.7.	Atributos de calidad para la interoperabilidad	18
2.2.	Diseño de <i>software</i>	19
2.2.1.	Diseño de los datos	19
2.2.2.	Diseño arquitectónico	19
2.2.3.	Diseño de la interfaz	19
2.2.4.	Diseño de componentes	20
2.2.5.	Diseño del despliegue	20
2.2.6.	Alto nivel	20
2.2.7.	Diseño de alto nivel	20
2.3.	Principios de diseño SOLID.....	20
2.3.1.	Principio de responsabilidad única	21
2.3.2.	Principio de abierto/cerrado	21
2.3.3.	Principio de sustitución de Liskov	21
2.3.4.	Principio de segregación de interfaces.....	21
2.3.5.	Principio de inversión de dependencias	21
2.4.	Patrones de diseño	22
2.4.1.	Patrón <i>Singleton</i>	22
2.4.2.	Patrón <i>Factory Method</i>	22
2.4.3.	Patrón <i>Observer</i>	22
2.4.4.	Patrón <i>Strategy</i>	22
2.4.5.	Patrón <i>Decorator</i>	23
2.4.6.	Patrón <i>Adapter</i>	23
2.4.7.	Patrón <i>Template Method</i>	23
2.4.8.	Patrón <i>Dependency Injection</i>	23
2.4.9.	Patrón <i>Facade</i>	23
2.4.10.	Patrón <i>Command</i>	24
2.4.11.	Patrón <i>Model-View-Controller</i>	24
2.5.	Axiomas fundamentales de diseño	24
2.5.1.	Axioma de separación de responsabilidades (SoC)	24
2.5.2.	Axioma de modularidad.....	24
2.5.3.	Axioma de cohesión.....	25
2.5.4.	Axioma de acoplamiento débil	25
2.5.5.	Axioma de reutilización	25
2.6.	Olores de <i>software</i>	25

2.6.1.	Código duplicado	25
2.6.2.	Funciones largas y clases "Dios"	26
2.6.3.	Dependencias circulares.....	26
2.6.4.	Datos perversos	26
2.6.5.	Envidia de funciones	26
2.6.6.	Clases poco cohesivas.....	26
2.7.	Deuda técnica	27
2.7.1.	Tipos de deuda técnica.....	27
2.7.2.	Impactos de la deuda técnica	27
2.7.3.	Gestión de la deuda técnica.....	27
2.8.	Pensamiento convergente y divergente en el diseño	28
2.8.1.	Pensamiento divergente	28
2.8.2.	Pensamiento convergente.....	28
2.9.	Verificación y validación del <i>software</i> (V&V)	28
2.9.1.	Verificación del <i>software</i>	29
2.9.2.	Validación del <i>software</i>	29
2.9.3.	Matriz de cobertura de requerimientos	29
2.9.4.	Historias de usuario.....	29
2.10.	Evolución del <i>software</i>	30
2.10.1.	Proceso cíclico de evolución.....	30
2.11.	Notación para graficar el diseño.....	30
2.11.1.	Integración entre sistemas de <i>software</i>	31
2.11.2.	Problemas de integración de sistemas.....	31
2.12.	UML	31
2.12.1.	Diagrama de clases.....	31
2.12.2.	Diagrama de secuencia.....	31
2.12.3.	Diagrama de historias de usuario	32
2.12.4.	Diagrama de actividades	32
2.12.5.	Diagrama de estados	32
2.12.6.	Diagrama de componentes	32
2.12.7.	Diagrama de despliegue	32
2.13.	Tipos de atributos	32
2.14.	Funciones	33
2.15.	Visibilidad	33

2.16.	Multiplicidad	33
2.17.	Relaciones entre clases	33
2.18.	Métodos de integración	34
2.19.	SaaS	35
2.19.1.	Arquitectura de microservicios	35
2.19.2.	API	35
2.19.3.	REST	35
2.19.4.	SOAP	36
2.20.	Seguridad en API	36
2.21.	Kyte	36
2.22.	POS	36
2.23.	FACEL	37
2.24.	Migración de datos	37
2.25.	Automatización y estándares de datos	37
2.26.	RPA	37
2.27.	Desactivación progresiva de sistemas	38
2.28.	Visualización de datos y KPI	38
2.29.	<i>Insights</i>	38
3.	Marco metodológico	39
3.1.	Tipo de investigación	39
3.2.	Enfoque de la investigación	39
3.3.	Alcance de la investigación	41
3.4.	Diseño de la investigación	42
3.5.	Fuentes de datos e información	43
3.5.1.	Fuentes primarias	43
3.5.2.	Fuentes secundarias	44
3.6.	Sujetos de investigación	45
3.7.	Variables o categorías de la investigación	46
3.7.1.	Datos redundantes y olores de <i>software</i>	46
3.7.2.	Brechas de interoperabilidad	46
3.7.3.	Herramientas tecnológicas de integración y eliminación de deuda técnica	47
3.7.4.	Simplificación de procesos y eliminación de redundancias	47
3.7.5.	Diseño preliminar de la arquitectura de integración	47
3.7.6.	Reducción de duplicaciones y mejora de interoperabilidad	48
3.7.7.	Historias de usuario y validación de la arquitectura	48

3.7.8.	Cumplimiento de requisitos de interoperabilidad	48
3.8.	Técnicas e instrumentos de recolección de datos	49
3.9.	Matriz de cobertura de las variables vs. el diseño de los instrumentos.....	50
3.10.	Procedimiento metodológico de la investigación.....	51
3.11.	Operacionalización de las variables o categorías	52
3.12.	Matriz de trazabilidad.....	54
4.	Análisis de resultados	56
4.1.	Fase 1: Diagnóstico de la situación actual	56
4.1.1.	Revisión documental.....	57
4.1.2.	Entrevistas.....	58
4.1.3.	Observación directa.....	59
4.1.4.	Análisis de hallazgos de la Fase 1	60
4.2.	Fase 2: Definición del estado meta	61
4.2.1.	Revisión documental.....	62
4.2.2.	Entrevistas.....	63
4.2.3.	Observación directa.....	64
4.2.4.	Grupos focales	65
4.2.5.	Lluvia de ideas	66
4.2.6.	Análisis de los hallazgos de la Fase 2.....	67
4.2.7.	Necesidades del sistema para el diseño de <i>software</i> de alto nivel	68
4.2.8.	Requerimientos del sistema	69
4.2.9.	Análisis de los requerimientos recopilados.....	71
4.3.	Fase 3: Diseño preliminar de la arquitectura.....	77
4.3.1.	Revisión documental.....	77
4.3.2.	Entrevistas.....	79
4.3.3.	Observación directa.....	80
4.3.4.	Grupo focal	81
4.3.5.	Lluvia de ideas	82
4.3.6.	Historias de usuario.....	82
5.	Propuesta de solución	85
5.1.	Fase 3: Diseño preliminar de la arquitectura.....	85
5.1.1.	Diseño de la propuesta	85
5.1.2.	Diseño de <i>software</i>	87
5.2.	Fase 4: Validación del diseño y cumplimiento de requisitos	100
5.2.1.	Revisión documental.....	100

5.2.2.	Pruebas de historias de usuario	102
5.2.3.	Matriz de cobertura de requerimientos	109
5.2.4.	Resultados de la validación.....	110
5.3.	Estudio de costos	111
5.3.1.	Inversión inicial	111
5.3.2.	Costos de la futura implementación.....	111
5.3.3.	Tecnologías	113
5.3.4.	ROI.....	114
5.3.5.	Beneficios esperados.....	114
5.3.6.	Cálculo del ROI de la propuesta de diseño de software de alto nivel.....	116
5.3.7.	Plazo de recuperación de la inversión.....	116
5.3.8.	Conclusión del análisis financiero	117
6.	Conclusiones	118
6.1.	Objetivo específico 1	118
6.2.	Objetivo específico 2.....	119
6.3.	Objetivo específico 3.....	120
6.4.	Objetivo específico 4.....	121
6.5.	Objetivo general	122
7.	Recomendaciones	123
7.1.	Objetivo específico 1	123
7.2.	Objetivo específico 2.....	123
7.3.	Objetivo específico 3.....	124
7.4.	Objetivo específico 4.....	124
7.5.	Objetivo general	124
8.	Referencias.....	125
9.	Apéndices.....	126

Índice de tablas

Tabla 1.....	17
Tabla 2.....	17
Tabla 3.....	18
Tabla 4.....	39
Tabla 5.....	40
Tabla 6.....	40
Tabla 7.....	41
Tabla 8.....	42
Tabla 9.....	43
Tabla 10.....	44
Tabla 11.....	45
Tabla 12.....	46
Tabla 13.....	49
Tabla 14.....	50
Tabla 15.....	51
Tabla 16.....	53
Tabla 17.....	54
Tabla 18.....	56
Tabla 19.....	57
Tabla 20.....	58
Tabla 21.....	59
Tabla 22.....	61
Tabla 23.....	62
Tabla 24.....	63
Tabla 25.....	64
Tabla 26.....	65
Tabla 27.....	66
Tabla 28.....	69
Tabla 29.....	69
Tabla 30.....	70
Tabla 31.....	72
Tabla 32.....	77
Tabla 33.....	77
Tabla 34.....	79
Tabla 35.....	80
Tabla 36.....	83
Tabla 37.....	88
Tabla 38.....	93
Tabla 39.....	97
Tabla 40.....	100
Tabla 41.....	101
Tabla 42.....	109
Tabla 43.....	111
Tabla 44.....	112
Tabla 45.....	112
Tabla 46.....	115

Índice de figuras

Figura 1	3
Figura 2	7
Figura 3	34
Figura 4	51
Figura 5	84
Figura 6	90
Figura 7	91
Figura 8	92
Figura 9	95
Figura 10	96
Figura 11	96
Figura 12	98
Figura 13	99
Figura 14	102
Figura 15	103
Figura 16	104
Figura 17	105
Figura 18	106
Figura 19	107
Figura 20	108

1. Introducción

Este documento presenta la propuesta de diseño de *software* de alto nivel para la interoperabilidad de las funcionalidades actuales de Hortiluma. La empresa enfrenta desafíos críticos debido a la falta de integración entre las funcionalidades de sus diversos sistemas de información, lo que ha provocado la duplicación de tareas, errores en los datos y una disminución de la eficiencia operativa. Estos problemas afectan tanto a empleados como a clientes y proveedores, comprometiendo la competitividad de Hortiluma en un sector agroindustrial donde la eficiencia y precisión son esenciales para mantener una posición sólida en un mercado dinámico.

1.1. Descripción general

Hortiluma ha implementado sus sistemas de información en diferentes momentos, sin una integración efectiva entre sus funcionalidades, lo que ha generado una fragmentación que dificulta la operación coordinada de la empresa. Por lo tanto, este proyecto tiene como objetivo proponer un diseño arquitectónico que permita la interoperabilidad de las funcionalidades de los distintos sistemas de información, mejorando la coherencia de los datos y facilitando una toma de decisiones más precisa y fundamentada. Con la implementación de esta arquitectura, Hortiluma podrá mejorar la gestión de datos, proporcionar una visión unificada y coherente de la información, mejorar la eficiencia operativa y reforzar el control sobre sus operaciones. Estos cambios fortalecerán su capacidad competitiva en el sector agroindustrial, donde la precisión y eficiencia son cruciales para mantener su liderazgo en un mercado altamente dinámico.

El documento está organizado en siete capítulos para asegurar una estructura clara y lógica. El Capítulo 1 constituye la Introducción, que proporciona una visión general del proyecto y del contenido del documento. El Capítulo 2, denominado Marco conceptual, ofrece un sistema coherente de conceptos y definiciones que sustentan el desarrollo del estudio, incluyendo teorías, antecedentes y soluciones similares que han sido analizadas para construir la metodología propuesta. El Capítulo 3, conocido como Marco metodológico, detalla los métodos y procedimientos utilizados para llevar a cabo la investigación, incluyendo el enfoque y diseño de la investigación, así como las técnicas de recolección de datos empleadas. En el Capítulo 4, Análisis de resultados, se presenta un análisis detallado de los datos obtenidos durante la investigación, comparando y contrastando los resultados para formular conclusiones relevantes. Por su parte, el Capítulo 5, titulado Propuesta de solución, expone en detalle la solución planteada para el problema, justificando cómo se cumplen los objetivos específicos y el objetivo general del proyecto. El Capítulo 6, Conclusiones, ofrece un resumen de los descubrimientos y hallazgos más relevantes, reflejando el cumplimiento de los objetivos planteados. Finalmente, el Capítulo 7, Recomendaciones, incluye sugerencias para la implementación de la solución propuesta, así como recomendaciones generales para la futura implementación.

Además, se incluyen Apéndices y las Referencias consultadas, proporcionando un soporte exhaustivo a la información presentada. Estas secciones aseguran que todos los aspectos técnicos y contextuales del proyecto estén bien documentados y sean fácilmente accesibles para el lector.

1.2. Antecedentes

En este apartado, se presentan los aspectos clave relacionados con Hortiluma, la organización donde se desarrolla el presente proyecto. El objetivo es proporcionar un contexto que permita comprender la naturaleza de la empresa y el entorno en el que se lleva a cabo esta propuesta.

1.2.1. Descripción de la organización

A continuación, se ofrece una descripción general de Hortiluma, abarcando aspectos como la misión, visión, valores y el equipo de trabajo.

1.2.2. La organización

Hortiluma es una empresa costarricense fundada en 1992, especializada en la producción agropecuaria sostenible. Con sede principal en San Antonio de Belén, la empresa también cuenta con operaciones ganaderas en la provincia de Limón y cultivos agrícolas en las regiones de Barva y Belén. Hortiluma se dedica a conectar directamente a los productores rurales con los consumidores urbanos, ofreciendo productos frescos de alta calidad, con un firme compromiso con la sostenibilidad y la integridad en toda su cadena de valor (Hortiluma, 2024).

El principal objetivo de Hortiluma es proporcionar alimentos frescos y sostenibles que cumplan con los más altos estándares de calidad. La empresa se distingue por su compromiso con el medio ambiente, implementando prácticas agrícolas responsables que protegen los recursos naturales y promueven el bienestar de las comunidades locales. Hortiluma opera bajo un enfoque integral que le permite satisfacer una amplia gama de necesidades de sus clientes, combinando ventas directas en su punto de venta principal, servicios mayoristas y una plataforma de pedidos en línea que facilita el acceso a sus productos a un público más amplio (Hortiluma, 2024).

Lo que diferencia a Hortiluma de otras empresas del sector es su enfoque holístico en la sostenibilidad, abarcando tanto la producción como la comercialización y distribución de sus productos. La empresa implementa un sistema de distribución que reduce costos y minimiza el impacto ambiental, lo que permite a sus clientes acceder a productos de alta calidad de manera rápida y sostenible. Además, Hortiluma se esfuerza por mantener relaciones sólidas con sus proveedores y clientes, basadas en la confianza, la transparencia y un compromiso mutuo con la sostenibilidad (Hortiluma, 2024).

1.2.2.1. Misión

La misión de Hortiluma es:

“Promover la excelencia en la agroindustria nacional, adoptando y difundiendo prácticas agrícolas y ganaderas responsables, y suministrando productos que representan la frescura y calidad de Costa Rica, con un impacto comunitario y ambiental positivo” (Hortiluma, 2024).

1.2.2.2. Visión

La visión de Hortiluma es:

“Ser pioneros en la producción y distribución de productos agropecuarios sostenibles y de alta calidad, elevando los estándares de la industria en Costa Rica y contribuyendo a un futuro sostenible globalmente” (Hortiluma, 2024).

1.2.2.3. Valores

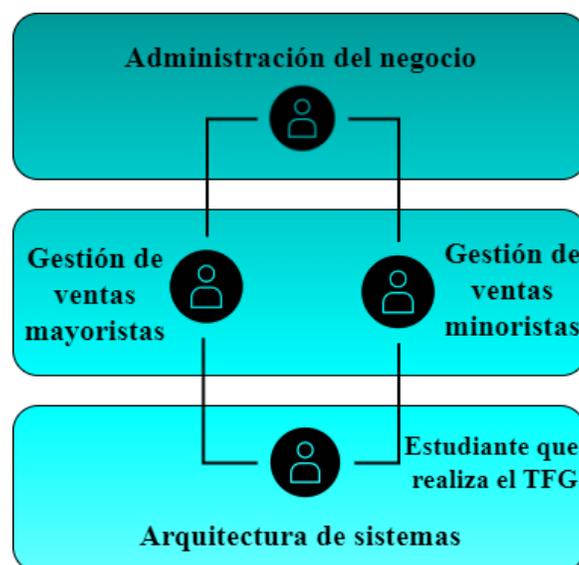
Los valores fundamentales que guían a Hortiluma son:

- “Sostenibilidad: Compromiso con la producción que cuida del planeta.
- Calidad: Mantenimiento de altos estándares en nuestros productos.
- Integridad: Transparencia y honestidad en todas las operaciones.
- Responsabilidad: Aseguramiento de la obtención de artículos acordados.
- Comunidad: Apoyo al bienestar de nuestra localidad, proveedores y clientes” (Hortiluma, 2024).

1.2.2.4. Equipo de trabajo

La mejora de la interoperabilidad de las funcionalidades actuales en Hortiluma es un esfuerzo colaborativo que involucra a miembros clave dentro de la organización. Cada persona aporta su experiencia y conocimientos específicos para asegurar que la solución propuesta cumpla eficazmente con las necesidades de la empresa. En la Figura 1, se muestra el organigrama del proyecto.

Figura 1
Organigrama del proyecto



Nota. Fuente: elaboración propia.

De acuerdo con la estructura del organigrama, a continuación, se especifican las funciones de cada rol:

- Administración del negocio: asume la función de contacto principal, ofreciendo dirección estratégica para alinear la arquitectura de la solución con los objetivos comerciales de Hortiluma.
- Gestión de ventas mayoristas: proporciona perspectivas para mejorar la eficiencia en la gestión y facturación de ventas mayoristas.
- Gestión de ventas minoristas: ofrece retroalimentación directa para la mejora continua del proceso de ventas minoristas, en un punto de venta.
- Arquitectura de sistemas: responsable de conceptualizar y diseñar una arquitectura que unifique las operaciones de Hortiluma, mejorando la interoperabilidad entre las funcionalidades actuales de los sistemas existentes.

Dentro de este organigrama, el rol de la persona arquitecta de sistemas asume una posición central en el desarrollo de la arquitectura tecnológica, trabajando en sinergia con cada sector involucrado, lo cual resulta crucial para el éxito del proyecto.

1.2.3. Trabajos similares realizados dentro y fuera de la organización

En esta sección, se describen los trabajos realizados tanto dentro como fuera de Hortiluma, relacionados con la administración de procesos de negocio en la gestión de la innovación. Estos antecedentes son fundamentales para contextualizar y justificar el enfoque del presente Trabajo Final de Graduación (TFG).

1.2.3.1. Proyectos de la organización

En su compromiso con la mejora continua, Hortiluma ha desarrollado una serie de proyectos que han establecido las bases tecnológicas y operativas esenciales:

Implementación de *E-commerce* con Kyte

Durante la pandemia de COVID-19, Hortiluma lanzó una plataforma de *e-commerce* utilizando Kyte para permitir a sus clientes realizar pedidos a domicilio. Esta iniciativa fue fundamental para mantener la continuidad operativa en un período de restricciones de movilidad. La experiencia adquirida en la gestión de ventas en línea y en la administración de pedidos subrayó la necesidad de un sistema más cohesionado que pudiera gestionar múltiples canales de ventas de manera (Hortiluma, 2024).

Desarrollo de presencia en redes sociales

Hortiluma ha desarrollado una presencia activa en redes sociales, mejorando su alcance y la interacción con los clientes. La gestión de la comunicación digital y el *marketing* directo a través de estos canales ha fortalecido la relación con la comunidad y ha proporcionado información valiosa sobre las preferencias del cliente. Esta experiencia es crucial para el desarrollo futuro de estrategias de ventas integradas y para mejorar la capacidad de respuesta de la empresa a las necesidades del mercado (Hortiluma, 2024).

1.2.3.2. Proyectos externos

Además de las iniciativas internas, la observación y el aprendizaje de prácticas externas ofrecen perspectivas valiosas que pueden aplicarse para mejorar la arquitectura de las funcionalidades de los sistemas actuales en Hortiluma. Se presentan dos casos de estudio que informan este TFG y su aplicación en el contexto agropecuario de la empresa:

Interoperabilidad de sistemas de información en salud

El proyecto de interoperabilidad en sistemas de información en salud, que utiliza estándares como Health Level Seven (HL7), Fast Healthcare Interoperability Resources (FHIR) y Digital Imaging and Communications in Medicine (DICOM), aborda la falta de interoperabilidad estructural y semántica entre diferentes funcionalidades de los sistemas de información médica. Este proyecto facilita el intercambio seguro de datos entre proveedores, pagadores, laboratorios y otros actores en el sector salud. Aunque Hortiluma opera en un sector distinto, enfrenta desafíos similares de fragmentación de datos y falta de interoperabilidad entre las funcionalidades de sus sistemas de información. Al adoptar un enfoque inspirado en el utilizado en el sector salud, Hortiluma podría implementar estándares de comunicación que permitan una integración más efectiva entre sus sistemas. Esto mejoraría la coherencia y la confiabilidad de la información, mejorando los procesos operativos y la toma de decisiones, de manera similar a lo logrado en el proyecto de salud (TATEEDA Global, 2022; BioMed Central, 2020).

Integración e interoperabilidad de sistemas robóticos

Este proyecto, liderado por el National Institute of Standards and Technology (NIST), se centra en la integración de sistemas robóticos en entornos de manufactura. Desarrolla modelos de información, protocolos y herramientas de prueba para permitir a los fabricantes ensamblar sistemas robóticos con capacidades humanas utilizando componentes interoperables. Este enfoque facilita la reconfiguración rápida de los sistemas de producción en respuesta a las dinámicas cambiantes del mercado. Hortiluma enfrenta retos similares en cuanto a la necesidad de integrar y coordinar las funcionalidades de sus sistemas de información para adaptarse a un mercado en constante evolución. Por lo tanto, la adopción de principios de interoperabilidad y flexibilidad, como los implementados en este proyecto, permite a Hortiluma reconfigurar y mejorar las funcionalidades de sus sistemas con mayor agilidad. Esto facilita una respuesta rápida a las demandas del mercado agropecuario, eliminando barreras tecnológicas y promoviendo la adopción de tecnologías avanzadas que incrementen su competitividad (NIST, 2021).

1.3. Planteamiento del problema

Esta sección detalla la problemática actual en Hortiluma, que ha dado origen a la necesidad del presente TFG. Las funcionalidades de los sistemas de información de la empresa, al operar en silos, han generado múltiples desafíos operativos y estratégicos que afectan la eficiencia y la capacidad de adaptarse a un mercado en constante cambio.

1.3.1. Situación problemática

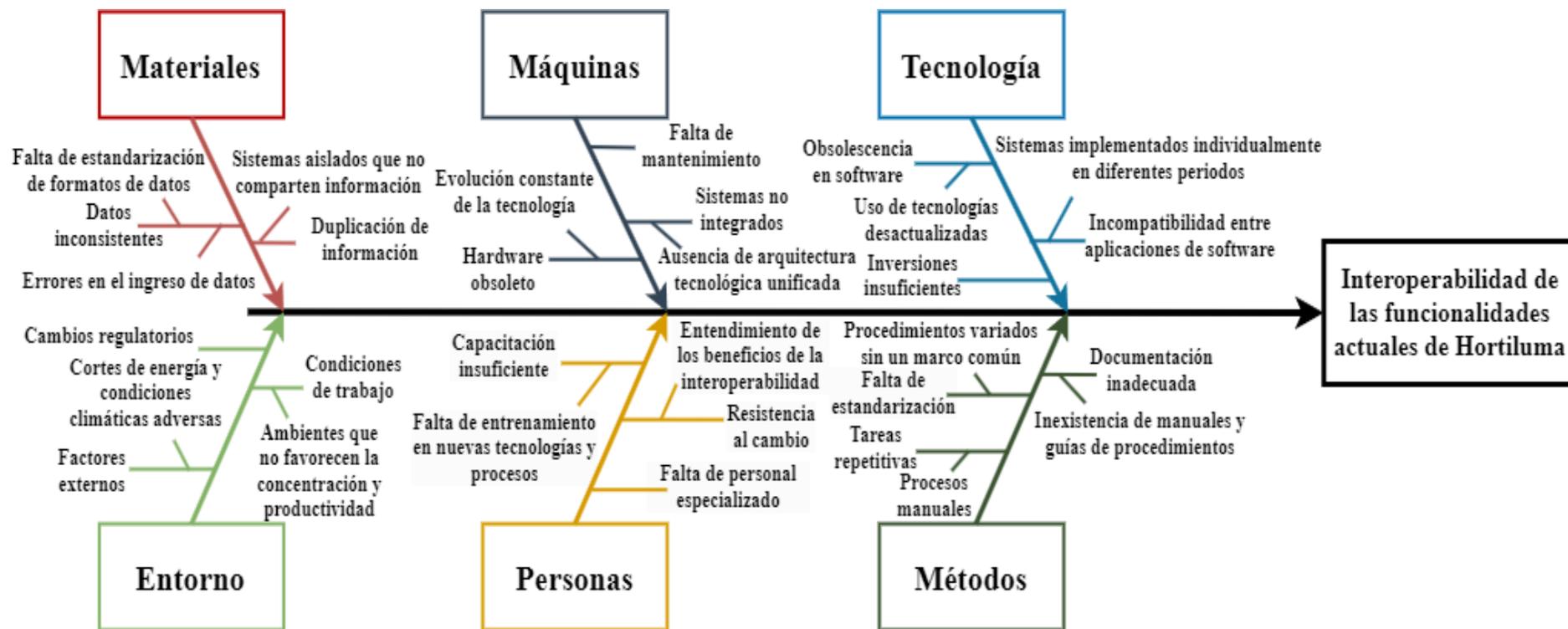
En Hortiluma, los sistemas utilizados actualmente para gestionar las operaciones son rígidos y operan de manera independiente entre sí, lo que ha provocado una fragmentación en las funcionalidades. Esto ha generado problemas importantes que afectan tanto la eficiencia operativa diaria como la capacidad estratégica de la empresa para crecer y mantenerse competitiva. Los principales problemas identificados incluyen:

- Duplicación de tareas: cada sistema requiere la entrada manual y repetitiva de datos para tareas como la facturación y las ventas, lo que resulta in y propenso a errores.
- Inconsistencias en los datos: la falta de un sistema integrado genera discrepancias en la información, lo que puede llevar a decisiones basadas en datos incorrectos o desactualizados.
- Flujo de trabajo in: la fragmentación obliga a los empleados a alternar entre diferentes plataformas para completar tareas simples, lo que extiende innecesariamente los procesos y reduce la productividad.
- Gestión y respuesta estratégica limitada: la rigidez de las funcionalidades de los sistemas existentes dificulta la rápida adaptación a los cambios del mercado y la explotación de nuevas oportunidades comerciales.
- Oportunidades y retos pendientes: la falta de interoperabilidad limita la capacidad de Hortiluma para expandirse a nuevas áreas como el comercio electrónico, restringiendo su potencial para capturar una mayor cuota de mercado.

Esta situación plantea la pregunta central de este TFG: ¿cómo se puede proponer un diseño de *software* de alto nivel para la interoperabilidad de las funcionalidades actuales de los sistemas de Hortiluma que mejore de manera demostrable la eficiencia operativa y asegure la consistencia de datos?

En la Figura 2, se representa la situación problemática mediante un diagrama de Ishikawa:

Figura 2
Diagrama de Ishikawa del problema



Nota. Fuente: Elaboración propia.

1.3.2. Justificación del proyecto

En esta sección, se justifica cómo los conocimientos adquiridos durante la carrera se aplican en este proyecto. La propuesta se enfoca en el diseño de *software* de alto nivel para la interoperabilidad de las funcionalidades actuales de Hortiluma. Este proyecto surge de la necesidad de mejorar los procesos de producción y gestión en la empresa, lo que permite una mejor administración de los recursos, así como un incremento en la productividad y eficiencia operativa.

Se analiza las necesidades específicas de Hortiluma, para recolectar los requerimientos esenciales y crear una arquitectura que integre todas las áreas de la empresa. El objetivo es desarrollar un diseño de *software* que permita una gestión interoperable de tareas como el control de inventarios, la facturación, las ventas y la trazabilidad de los productos.

La ingeniería de requerimientos es una etapa crítica en el desarrollo del *software*, ya que define los diferentes tipos de requerimientos del sistema. Para Hortiluma, se lleva a cabo un análisis exhaustivo de las necesidades del negocio, los requisitos de los usuarios y los requisitos técnicos del producto. Según Wieggers (2013), un correcto levantamiento de requerimientos es fundamental para el éxito del proyecto, asegurando que las soluciones propuestas realmente respondan a las necesidades de la empresa.

El uso de diagramas de Unified Modeling Language (UML) es esencial para modelar, construir y documentar el sistema integral de gestión operativa. Según Paul y Kimmel (2011), de esta forma, se proporciona un lenguaje estándar para visualizar el diseño de *software*, lo que facilita la comunicación entre los desarrolladores y los involucrados, además, garantiza una comprensión clara del sistema.

De acuerdo con Sommerville (2005), el diseño de *software* transforma los requerimientos del usuario en una forma adecuada para la codificación e implementación. Para Hortiluma, esto implica diseñar un sistema que soporte todas las funcionalidades críticas actuales, desde la gestión de inventarios y recursos hasta la logística de distribución, asegurando que el *software* sea robusto, escalable y fácil de mantener.

La arquitectura de *software* para la interoperabilidad define la estructura del sistema como una serie de componentes bien organizados. Para Sommerville (2005), una arquitectura bien definida permite una mejor integración de las diversas funcionalidades de los sistemas actuales y facilita la colaboración entre los equipos de desarrollo. Por su parte, Hortiluma indica que se diseña una arquitectura que soporte la integración de diversas funcionalidades, garantizando una operación fluida del sistema.

La documentación detallada del *software* es crucial para garantizar una implementación y mantenimiento efectivos. Según Sommerville (2005), esta documentación incluye la definición y especificación de los requerimientos, la arquitectura del sistema, el diseño detallado, los procesos y servicios ofrecidos, así como la especificación técnica de las interfaces de programación de aplicaciones (API). Esto permite que los desarrolladores, gerentes y otros involucrados tengan una comprensión clara y detallada del sistema.

1.3.3. Beneficios esperados o aportes del TFG

El proyecto de integración de las funcionalidades actuales de los sistemas en Hortiluma se enfoca en una propuesta de diseño destinada a abordar problemas operativos y estratégicos que impactan la eficiencia y capacidad de adaptación de la empresa. Utilizando principios de ingeniería de software y sistemas de información, como el modelado UML y la arquitectura de sistemas, esta propuesta conceptual establece las bases para optimizar las operaciones. Los beneficios se presentan a continuación, clasificados en directos e indirectos, junto con consideraciones sobre su materialización.

1.3.3.1. Beneficios directos

- Eliminación de la duplicación de tareas: el diseño propuesto centraliza el ingreso de datos, permitiendo que la información se registre una sola vez y se actualice automáticamente en toda la plataforma. Esto disminuye los tiempos asociados con tareas redundantes y los errores causados por la duplicidad de datos en diferentes sistemas.
- Consistencia y precisión en los datos: la centralización garantiza que los datos sean coherentes y estén actualizados en todas las funcionalidades del sistema, proporcionando una base confiable para la toma de decisiones.
- Mejora del flujo de trabajo: al integrar las funcionalidades en una única plataforma, el diseño elimina la necesidad de alternar entre herramientas, simplificando los procesos y optimizando el uso del tiempo de los empleados.

1.3.3.2. Beneficios indirectos

- Flexibilidad y capacidad de respuesta: el diseño permite que los sistemas se adapten a cambios en las necesidades del mercado, como la incorporación de nuevas líneas de productos, sin afectar la operatividad general.
- Facilidad para la expansión hacia nuevos mercados: la estructura centralizada facilita la integración con plataformas de comercio electrónico, aumentando la posibilidad de llegar a nuevos clientes y mercados.
- Reducción de costos operativos a largo plazo: la disminución de errores y tareas duplicadas contribuye a una mejor utilización de los recursos y a una reducción de los costos operativos en el tiempo.

1.3.3.3. Consideraciones sobre la materialización de los beneficios

Los beneficios descritos son resultados esperados basados en el diseño conceptual. Su materialización dependerá de varios factores que exceden el alcance de este proyecto:

- Implementación del sistema: el desarrollo técnico y la puesta en marcha del sistema son fundamentales para que se logren los beneficios planteados.
- Migración y limpieza de datos: la transferencia de los datos existentes, previa depuración y estandarización, será esencial para garantizar la coherencia y funcionalidad del sistema integrado.
- Capacitación de los usuarios: la adopción adecuada del sistema requiere que los usuarios finales comprendan cómo utilizar las herramientas propuestas.
- Validación y ajustes post-implementación: el sistema deberá someterse a pruebas en un entorno real y, de ser necesario, realizar ajustes para que cumpla con los objetivos definidos.

1.4. Objetivos

Los objetivos generales y específicos del TFG se presentan a continuación, formulados según la metodología SMART.

1.4.1. Objetivo general

Proponer un diseño de *software* de alto nivel para la interoperabilidad de las funcionalidades actuales de los sistemas de Hortiluma, en un periodo de dieciséis semanas.

1.4.2. Objetivos específicos

A continuación, se presentan los objetivos específicos del proyecto:

1. Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.
2. Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.
3. Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.
4. Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.

1.5. Alcance

El alcance de este proyecto se limita a la propuesta de diseño conceptual de un sistema integrado para Hortiluma, que busca optimizar las tareas y datos redundantes. Este proyecto no incluye el desarrollo técnico ni la implementación del sistema. A continuación, se detallan los aspectos incluidos y excluidos:

Incluye:

- Diagnóstico de las brechas de interoperabilidad entre sistemas.
- Definición del estado objetivo para la integración de funcionalidades y datos.
- Diseño preliminar y validación de una arquitectura conceptual.

Excluye:

- Construcción técnica y desarrollo del sistema.
- Migración de datos y capacitación de usuarios.
- Validación funcional en entornos reales.

1.5.1. Supuestos del proyecto

Este proyecto se fundamenta en los siguientes supuestos clave:

- Hortiluma proporciona acceso completo y sin restricciones a toda la información y documentación relevante sobre las funcionalidades actuales de los sistemas y los requerimientos del negocio, incluyendo documentación técnica y manuales de usuario.
- Hortiluma facilita la programación de reuniones y entrevistas con los usuarios clave y otros involucrados del proyecto siempre que sea necesario, garantizando que se puedan recoger y validar los requerimientos de manera oportuna.
- Se espera una colaboración total de todos los involucrados durante la definición de requerimientos y las fases de validación del diseño, asegurando un flujo continuo de retroalimentación y apoyo.
- Existe una comunicación asertiva entre todos los involucrados en el proyecto, incluyendo el equipo de desarrollo, los usuarios finales y la gerencia de Hortiluma, para mantener a todas las partes alineadas y actualizadas sobre el progreso y cualquier cambio en el proyecto.
- La contraparte empresarial notifica a todos los involucrados sobre la realización del presente proyecto.

1.6. Entregables del proyecto

Esta sección detalla los entregables claves del proyecto, abarcando tanto los entregables académicos como para la empresa. Estos son fundamentales para el éxito del proyecto, proporcionando una estructura clara y objetivos medibles.

1.6.1. Entregables académicos

A continuación, se describen los entregables académicos del proyecto:

- Capítulo 1. Introducción.
- Capítulo 2. Marco conceptual.
- Capítulo 3. Desarrollo metodológico.
- Capítulo 4. Análisis de resultados.
- Capítulo 5. Propuesta de solución.
- Capítulo 6. Conclusiones.
- Capítulo 7. Recomendaciones.
- Referencias
- Apéndices
- Anexos
- Glosario

1.6.2. Entregables para la empresa

A continuación, se describen los entregables para la empresa que brinda el proyecto:

- Documento que especifica los requerimientos del sistema
 - Diagnóstico detallado del funcionamiento de los sistemas de información actuales de Hortiluma, identificando las brechas de interoperabilidad y las inconsistencias de datos.
 - Requerimientos de integración funcionales y no funcionales, alineados con la norma ISO/IEC/IEEE 29148.

- Documentación del diseño de *software* de alto nivel propuesto, incluyendo arquitecturas y diagramas:
 - Historias de usuario.
 - Componentes.
 - Secuencia.
 - Otros diagramas relevantes para asegurar la interoperabilidad de las funcionalidades actuales de los sistemas de Hortiluma.
- Validaciones costo-beneficio
 - Evaluación detallada del costo-beneficio del diseño de *software* propuesto, asegurando que la solución esté alineada con las necesidades estratégicas y operativas de Hortiluma.
- Medidas de aceptación de la propuesta de diseño
 - Definición de criterios claros para la aceptación del diseño, asegurando que se cumplan todos los requerimientos y se garantice la interoperabilidad entre los sistemas actuales.
- Matriz de cobertura de los requerimientos
 - Matriz que evalúa el grado de cumplimiento de los requerimientos definidos en el diseño, asegurando que cada uno esté cubierto y alineado con los objetivos del proyecto.
 - Comparación de los requerimientos con las funcionalidades incluidas en el diseño de *software* para garantizar una cobertura total.
- Validación por medio de historias de usuario
 - Validación del diseño propuesto mediante historias de usuario específicos, asegurando que el sistema cubra todas las necesidades operativas de Hortiluma.
 - Simulaciones de escenarios basadas en los historias de usuario, garantizando que el diseño propuesto responda adecuadamente a las operaciones diarias de la empresa y cumpla con los objetivos de interoperabilidad.

1.7. Gestión del proyecto

La gestión del proyecto se estructura alrededor de tres artefactos clave para asegurar una ejecución organizada y :

1.7.1. Cronograma

Se presenta un cronograma detallado que cubre todas las fases del proyecto, desde el análisis inicial hasta la evaluación final, garantizando que todas las actividades estén claramente planificadas y definidas temporalmente. Apéndice A. Cronograma de trabajo

1.7.2. Minutas

Se utiliza una plantilla estandarizada para documentar de manera efectiva las reuniones con los involucrados, capturando los objetivos, discusiones y compromisos establecidos. Apéndice B. Plantilla de la minuta

1.7.3. Gestión del cambio

Para gestionar las solicitudes de cambio, se utiliza una plantilla específica que detalla el proceso y los responsables, evaluando el impacto y coordinando la aprobación e implementación de cambios. Apéndice C. Plantilla para la gestión de cambios.

1.8. Exclusiones del proyecto

El alcance de este proyecto excluye específicamente:

- La integración se limita a los sistemas y tecnologías definidos y documentados en los requerimientos iniciales, evitando imprevistos y sobrecargas de trabajo adicionales.
- Las funcionalidades de acceso al catálogo de productos y la realización de pedidos está disponibles únicamente a través de una interfaz web.
- El proyecto se enfoca exclusivamente en la propuesta y diseño de la arquitectura, dejando la implementación para fases posteriores, tras la validación y aprobación del diseño.

1.9. Limitaciones del proyecto

El proyecto enfrenta varias limitaciones que podrían afectar la implementación y operación de la solución. A continuación, se mencionan los posibles factores que podrían influir en el desarrollo del proyecto:

1. La disponibilidad de los involucrados puede generar demoras en la recopilación de información o requerimientos.
2. Requerimientos identificados podrían resultar fuera del alcance definido, lo que podría llevar a ajustes en las expectativas y entregables.
3. Errores o faltantes en la documentación de los sistemas actuales de Hortiluma pueden afectar el avance y la comprensión del contexto del proyecto.
4. La capacidad de Hortiluma para adoptar nuevas tecnologías y ajustar sus procesos operativos es crucial y podría requerir una gestión del cambio efectiva.
5. Cumplir con todas las regulaciones relevantes podría limitar algunas opciones de diseño y funcionalidad.
6. La infraestructura tecnológica actual de Hortiluma podría necesitar mejoras para soportar eficazmente la nueva solución, lo que podría implicar inversiones adicionales en *hardware* o *software*.

2. Marco conceptual

En este capítulo, se presentan los conceptos fundamentales que sustentan el Trabajo Final de Graduación (TFG). Estos son fundamentales para resolver los problemas de duplicación de tareas y desintegración funcional dentro de la empresa. Se enfoca en principios de interoperabilidad, arquitectura de *software* y la integración de sistemas.

2.1. Proceso de *software*

El proceso de *software* es un conjunto estructurado de actividades que guían las diferentes etapas del desarrollo de un sistema. Según Sommerville (2011), este proceso incluye fases clave como la especificación, diseño, desarrollo, verificación, validación y evolución. Estas fases son fundamentales para garantizar que el sistema cumpla con los requisitos establecidos y responda adecuadamente a las necesidades de los usuarios.

El estándar ISO/IEC 12207 establece un marco integral para los procesos de ciclo de vida del *software*, desde la recolección de requisitos hasta el mantenimiento y la evolución del sistema. Este estándar define un enfoque sistemático para desarrollar *software* de calidad, estructurando las actividades en tres categorías principales:

- Procesos primarios: comprenden las actividades relacionadas con la adquisición, suministro, desarrollo, operación y mantenimiento del *software*.
- Procesos de soporte: incluyen la verificación, validación, revisión, auditoría y gestión de la configuración del sistema.
- Procesos organizativos: se centran en la gestión de proyectos, la mejora del proceso y la formación del personal.

2.1.1. Especificación del *software*

La especificación del *software* se refiere a la definición detallada de los requisitos funcionales y no funcionales que debe cumplir el sistema. De acuerdo con IEEE 830, los requisitos deben ser claros, concisos y verificables, lo que garantiza que se pueda evaluar si el sistema cumple con sus objetivos. La especificación es el primer paso en el proceso de desarrollo y tiene como objetivo asegurar que todas las partes interesadas tengan una comprensión común del sistema.

2.1.2. Levantamiento de requerimientos

El levantamiento de requerimientos es el proceso clave para la definición de las necesidades del sistema por parte de los usuarios. Según Sommerville (2005), este proceso incluye la recolección de necesidades funcionales y no funcionales. En el contexto del TFG, se enfoca en los requerimientos de interoperabilidad entre los sistemas actuales de Hortiluma. De acuerdo con la norma ISO/IEC/IEEE 29148, los requerimientos deben ser:

- Necesarios: definen capacidades críticas del sistema y cualquier deficiencia afectará directamente la solución propuesta.

- Sin ambigüedad: los requerimientos deben ser claros y concisos, especialmente al definir la interoperabilidad de las funcionalidades de los diferentes sistemas.
- Verificables: se deben establecer pruebas claras para validar la interoperabilidad de las funcionalidades integradas.

2.1.3. Tipos de requerimientos

La clasificación de los requerimientos es clave para asegurar una correcta integración de las funcionalidades de Hortiluma. Según Wieggers y Beatty (2013), se identifican tres tipos de requerimientos:

- Requerimientos funcionales: describen el comportamiento que se espera del sistema bajo condiciones específicas, como la sincronización de bases de datos y la consolidación de tareas duplicadas.
- Requerimientos no funcionales: incluyen atributos como rendimiento, mantenibilidad y seguridad, necesarios para asegurar que el sistema integrado funcione mente y cumpla con los objetivos de interoperabilidad.
- Requerimientos de interfaz externa: describen las interacciones entre los distintos sistemas que deben comunicarse para compartir datos y funcionalidades, asegurando así una integración entre los módulos actuales.

2.1.4. Arquitectura de *software* para la interoperabilidad

La arquitectura de *software* es esencial para garantizar la interoperabilidad. Por lo que la arquitectura propuesta para Hortiluma debe seguir principios de arquitectura de microservicios permite una integración más flexible y escalable de las funcionalidades. Además, las API juegan un rol fundamental al permitir la comunicación entre los diferentes módulos que antes funcionaban de manera independiente, asegurando así la consistencia de datos y evitando duplicaciones.

2.1.5. Procesamiento de requerimientos y priorización

El análisis y priorización de requerimientos es una parte crítica del desarrollo de *software*. En este TFG, se utiliza la técnica MoSCoW (Lant, 2019) para clasificar los requerimientos en:

- *Must* (deben): requerimientos esenciales para lograr la interoperabilidad.
- *Should* (deberían): requerimientos deseables que mejoran la eficiencia del sistema.
- *Could* (podrían): requerimientos adicionales a implementar si el tiempo lo permite.
- *Won't* (no se incluirán): requerimientos que se consideran fuera del alcance del proyecto.

Para puntuar los requerimientos, se utiliza una combinación de urgencia y valor del negocio, de acuerdo con la metodología de Lant (2019):

- Urgencia: se refiere a la necesidad de implementar un requerimiento de manera completa, rápida y precisa.
- Valor del negocio: representa el valor que tiene un requerimiento para la organización, en términos de impacto estratégico, eficiencia operativa y ventaja competitiva.

Cada requerimiento es puntuado del 1 al 5 en ambos términos (urgencia y valor del negocio), en la Tabla 1 y Tabla 2, siguiendo las guías propuestas por Lant (2019). La priorización final se obtiene multiplicando ambos valores y se posiciona el resultado en la Tabla 3 que utiliza el marco MoSCoW.

Tabla 1
Guía para la puntuación de la urgencia de un requerimiento

Valor	Guía
5	Extremadamente limitado en el tiempo. Nivel extremo de dependencia de otros elementos para completar esta tarea. Sin completarse inmediatamente, tiene valor bajo hacerlo.
4	Altamente limitado en el tiempo. Alto nivel de dependencia de otros elementos en la realización de esta tarea. Importante por requisitos contractuales o del cliente.
3	Moderadamente limitado en el tiempo. Dependencia moderada. Deseable completar en los próximos uno o dos <i>sprints</i> .
2	Mínimamente limitado en el tiempo. Dependencia mínima. La finalización en los siguientes dos o tres <i>sprints</i> es adecuada.
1	Sin limitaciones de tiempo. Sin dependencia. Bajo o ningún impacto si se retrasa.

Nota. Fuente: Elaboración propia.

Tabla 2
Guía para la puntuación del valor del negocio de un requerimiento

Valor	Guía
5	Extremadamente importante para la mayoría o todos los clientes. Impacto extremo en la marca o reputación. Crítico para el éxito del negocio.
4	Importante para clientes. Impacto significativo en la marca o reputación. Ventaja competitiva significativa.
3	Importante para un número moderado de clientes. Impacto significativo moderado en la marca o reputación. Ventaja competitiva importante moderada.
2	Importante solo para unos clientes. Impacto menor en la marca o reputación. Ventaja competitiva menor.
1	Importante solo para unos o ningún cliente. Bajo o ningún impacto en la marca o reputación. Poca o ninguna ventaja competitiva.

Nota. Fuente: Elaboración propia.

Según Lant (2019), la priorización de un requerimiento se calcula multiplicando la urgencia por el valor del negocio. El resultado se clasifica en la Tabla 3 para determinar su prioridad en el marco MoSCoW:

Tabla 3

Valor del Negocio	1	2	3	4	5
5	5 (W)	10 (C)	15 (S)	20 (S)	25 (M)
4	4 (W)	8 (C)	12 (C)	16 (S)	20 (S)
3	3 (W)	6 (C)	9 (C)	12 (C)	15 (S)
2	2 (W)	4 (W)	6 (C)	8 (C)	10 (C)
1	1 (W)	2 (W)	3 (W)	4 (W)	5 (W)

Nota. Fuente: Elaboración propia.

Esta Tabla 3 refleja la priorización de los requerimientos, combinando la urgencia y el valor de negocio. Se puede usar como base para clasificar y priorizar los requerimientos del proyecto, asegurando que los elementos esenciales sean identificados como *Must* y se implementen antes que otros.

2.1.6. Integración de sistemas y verificación de interoperabilidad

La integración de sistemas es uno de los pilares de este TFG. Según Sommerville (2005), la verificación y validación del *software* (v&v) aseguran que el sistema no solo funcione correctamente, sino que cumpla con los objetivos de interoperabilidad planteados. En este contexto, se diseñan pruebas específicas para validar la correcta comunicación entre los sistemas, utilizando técnicas como pruebas de integración de API y pruebas de carga para evaluar el rendimiento del sistema integrado.

2.1.7. Atributos de calidad para la interoperabilidad

Los atributos de calidad son esenciales para asegurar que el sistema sea , escalable y mantenible. De acuerdo con la norma ISO/IEC/IEEE 29148, los atributos de calidad que se consideran son:

- Rendimiento: capacidad del sistema para manejar grandes volúmenes de datos sin afectar la integración.
- Mantenibilidad: facilidad con la que se puede modificar el sistema para adaptarse a nuevas funcionalidades o corregir errores.
- Escalabilidad: capacidad del sistema para integrar nuevos módulos y funcionalidades sin pérdida de rendimiento.

2.2. Diseño de *software*

El diseño de *software* consiste en definir la estructura, componentes, módulos, interfaces y datos de un sistema informático, con el objetivo de cumplir tanto con los requisitos funcionales como no funcionales. Este proceso abarca desde un diseño general de la arquitectura (diseño de alto nivel) hasta los detalles técnicos y específicos. Según Pressman (2005), un buen diseño de *software* debe garantizar la modularidad, flexibilidad y mantenibilidad, permitiendo que el sistema evolucione sin comprometer su rendimiento o capacidad para incorporar nuevas funcionalidades. A continuación, se detallan los componentes del diseño de *software*:

2.2.1. Diseño de los datos

Se centra en la estructura y organización de la información dentro de un sistema. Un modelo de datos compartido es esencial para garantizar la consistencia y coherencia de los datos entre los diferentes módulos del sistema. Para facilitar la interoperabilidad y la integración, se utilizan API y mecanismos de sincronización de datos que aseguran que todos los componentes del sistema trabajen con información actualizada.

2.2.2. Diseño arquitectónico

Define cómo se comunican las diferentes partes de un sistema. Según Pressman (2005), la arquitectura de *software* debe especificar claramente las interfaces que permiten la integración entre los componentes del sistema. Los patrones arquitectónicos como SOAP y arquitectura de microservicios aseguran que los sistemas sean flexibles y escalables, permitiendo modificaciones y expansiones sin afectar al sistema completo. Estos diseños incluyen:

- Dominio de la aplicación: define el ámbito y la naturaleza del *software*, estructurando los requerimientos en dominios específicos del sistema.
- Modelos de requerimientos: utilización de diagramas de flujo de datos y clases de análisis para definir las interacciones entre los componentes.
- Estilos arquitectónicos: los patrones como los microservicios facilitan la escalabilidad y la modificación del sistema sin impactar su integridad global.

2.2.3. Diseño de la interfaz

Describe cómo los usuarios y otros sistemas interactúan con el *software*. Las interfaces deben ser intuitivas y permitir una gestión de los flujos de trabajo. Las API juegan un papel fundamental en el diseño de interfaces, ya que permiten la integración y comunicación fluida entre los diferentes módulos del sistema y con sistemas externos, facilitando así la interoperabilidad.

2.2.4. Diseño de componentes

Se refiere a la segmentación del sistema en partes autónomas que pueden desarrollarse, desplegarse y mantenerse de manera independiente. Aplicando principios como el Principio de responsabilidad única y el Principio de segregación de interfaces, cada componente debe cumplir una función específica, minimizando dependencias y asegurando que los cambios en una parte del sistema no afecten a las demás.

2.2.5. Diseño del despliegue

El despliegue del *software* implica implementar el sistema en un entorno de producción o prueba. Tecnologías como los contenedores y los sistemas de integración continua se utilizan para automatizar y gestionar el despliegue de manera , minimizando interrupciones operativas y facilitando la escalabilidad del sistema. Además, el uso de contenedores como Docker y la gestión de infraestructura mediante plataformas como Kubernetes permiten mejorar la implementación y escalabilidad de los módulos del sistema.

2.2.6. Alto nivel

En el contexto de la ingeniería de *software*, el término alto nivel se refiere a un diseño que opera en un mayor grado de abstracción. Esto significa que se centra en decisiones arquitectónicas generales, como la organización de los componentes y las interacciones entre ellos, en lugar de los detalles técnicos o específicos de implementación. Un diseño de alto nivel define los bloques principales del sistema, como módulos o subsistemas, sus responsabilidades y las interfaces a través de las cuales se comunican. Este tipo de diseño es crucial porque proporciona una base sólida y flexible para el desarrollo futuro, asegurando que las decisiones estratégicas permitan al sistema crecer sin comprometer su estabilidad y mantenibilidad.

2.2.7. Diseño de alto nivel

El diseño de *software* de alto nivel se refiere a una visión más abstracta y estratégica de la arquitectura del sistema. En esta etapa, las decisiones arquitectónicas sobre cómo se organizan y relacionan los componentes principales son prioritarias, sin abordar los detalles técnicos de implementación.

Según Jaiswal (2019), un diseño de alto nivel especifica claramente cómo interactúan los módulos, clases y funciones, proporcionando una base sólida para el desarrollo futuro. Además, facilita la escalabilidad y el mantenimiento del sistema, permitiendo que el *software* se adapte a nuevas necesidades y evoluciones sin generar altos costos.

2.3. Principios de diseño SOLID

Estos principios, formalizados por Robert C. Martin en el año 2000, son reglas fundamentales en el diseño orientado a objetos que mejoran la mantenibilidad, flexibilidad y escalabilidad del *software*. Estos principios permiten crear *software* extensible y fácil de modificar sin comprometer la estabilidad del sistema (Martin, 2003).

2.3.1. Principio de responsabilidad única

El principio de responsabilidad única sostiene que cada clase o módulo debe tener una sola razón para cambiar. Aplicar este principio reduce la complejidad del código y facilita su mantenimiento, ya que el sistema es más sencillo de modificar cuando las responsabilidades están claramente delimitadas. Clases con múltiples responsabilidades pueden generar conflictos de cambios, dificultando la evolución del sistema (Martin, 2003). En los diagramas UML, las clases diseñadas bajo el SRP se muestran con funciones específicas y bien definidas.

2.3.2. Principio de abierto/cerrado

El principio de abierto/cerrado establece que los módulos de *software* deben estar abiertos para extensión, pero cerrados a la modificación. Es decir, las funcionalidades nuevas deben agregarse sin modificar el código existente, protegiendo el comportamiento actual del sistema (Meyer, 1988). El uso de patrón *decorator* y patrón *strategy* permite agregar comportamientos nuevos sin alterar la estructura de clases original, lo cual es visible en los diagramas UML con herencias o interfaces extendidas.

2.3.3. Principio de sustitución de Liskov

El principio de sustitución de liskov establece que los objetos de una subclase deben poder reemplazar a los de su clase base sin alterar la funcionalidad del sistema. Esto asegura que las subclases mantengan el comportamiento de la clase padre y que el sistema funcione de manera coherente (Liskov, 1987). Un diseño UML basado en LSP muestra subclases que no alteran las precondiciones o postcondiciones de los métodos heredados.

2.3.4. Principio de segregación de interfaces

El principio de segregación de interfaces promueve el diseño de interfaces específicas en lugar de interfaces generales que obliguen a los clientes a implementar métodos innecesarios. Esto mejora la cohesión del código y evita sobrecargar las clases con métodos no relacionados. En los diagramas UML, esto se refleja en interfaces pequeñas y bien enfocadas que responden a las necesidades concretas de las clases (Martin, 2003).

2.3.5. Principio de inversión de dependencias

El principio de inversión de dependencias postula que los módulos de alto nivel no deben depender de los módulos de bajo nivel, sino de abstracciones como interfaces o clases abstractas. Esto desacopla los componentes y permite que los módulos sean más flexibles y fáciles de modificar (Martin, 2003). Por su parte, los diagramas UML que cumplen con este principio muestran dependencias hacia interfaces en lugar de dependencias entre clases concretas.

2.4. Patrones de diseño

Los patrones de diseño son soluciones reutilizables que abordan problemas recurrentes en el desarrollo de *software* orientado a objetos. Estos patrones permiten la creación de sistemas flexibles, escalables y mantenibles, minimizando la duplicación de código y facilitando la reducción de la complejidad (Gamma et al., 1994). A continuación, se detallan los patrones de diseño más utilizados, que son clave para este proyecto.

2.4.1. Patrón *Singleton*

El patrón *singleton* garantiza que una clase tenga una única instancia durante el ciclo de vida del sistema, proporcionando un punto global de acceso a esta instancia. Este patrón es particularmente útil en situaciones donde es necesario gestionar recursos compartidos, como configuraciones globales, conexiones de bases de datos o controladores de sistemas (Gamma et al., 1994). Al implementar este patrón, se evita la duplicación innecesaria de objetos, mejorando el uso de recursos.

2.4.2. Patrón *Factory Method*

El patrón *factory method* permite delegar la creación de objetos a sus subclasses, proporcionando flexibilidad en el proceso de instancia de objetos sin especificar explícitamente la clase exacta que se debe instanciar. Este patrón es ideal en sistemas que necesitan crear objetos dinámicamente, mejorando la extensibilidad del *software* sin alterar el código base existente (Gamma et al., 1994).

2.4.3. Patrón *Observer*

El patrón *observer* establece una relación de suscripción entre un sujeto y uno o más observadores, de tal manera que los observadores son notificados automáticamente cuando el sujeto cambia de estado. Este patrón es fundamental para sistemas que manejan eventos o notificaciones, como interfaces de usuario o sistemas de monitoreo, ya que permite una interacción entre componentes sin necesidad de un fuerte acoplamiento entre ellos (Gamma et al., 1994). Al emplearlo, se garantiza una actualización automática de los componentes dependientes, mejorando la modularidad del sistema.

2.4.4. Patrón *Strategy*

El patrón *strategy* define una familia de algoritmos, encapsulando cada uno en una clase separada, lo que permite su intercambio dinámico durante la ejecución sin alterar el código cliente que los utiliza. Es ideal cuando el sistema debe cambiar su comportamiento dependiendo de las condiciones de ejecución. Por ejemplo, este patrón permite seleccionar distintos algoritmos de validación o procesamiento de datos de forma flexible. *Strategy* reduce el acoplamiento y facilita la extensión del sistema añadiendo nuevas estrategias sin modificar la lógica existente (Gamma et al., 1994).

2.4.5. Patrón *Decorator*

El patrón *decorator* permite agregar funcionalidades adicionales a un objeto de manera dinámica, sin modificar su estructura original. Este patrón es ideal cuando se busca aumentar las capacidades de un objeto sin alterar su código base, lo que permite una mayor flexibilidad y adaptabilidad del sistema (Gamma et al., 1994). Esto es especialmente útil en sistemas que requieren agregar comportamientos o propiedades de manera incremental, favoreciendo la reutilización de código y evitando la creación de clases derivadas innecesarias.

2.4.6. Patrón *Adapter*

El patrón *adapter* permite que clases con interfaces incompatibles trabajen juntas al actuar como un intermediario que traduce las interfaces de una clase para que sea compatible con otra. Este patrón es esencial cuando se necesita integrar componentes de *software* existentes en un nuevo sistema sin modificar su estructura, facilitando la interoperabilidad entre módulos que originalmente no estaban diseñados para interactuar (Gamma et al., 1994). Su implementación reduce la necesidad de reescribir o modificar grandes secciones del código original.

2.4.7. Patrón *Template Method*

El patrón *template method* define el esqueleto de un algoritmo en un método base, delegando algunos pasos a las subclasses. Este patrón permite que las subclasses redefinan ciertos pasos de un algoritmo sin cambiar su estructura global. Es útil cuando múltiples clases comparten la estructura general de un algoritmo, pero requieren implementar detalles específicos de manera diferente. Esto favorece la reutilización de código, ya que evita duplicar la lógica común mientras permite variaciones controladas (Gamma et al., 1994).

2.4.8. Patrón *Dependency Injection*

El patrón *dependency injection* facilita la inversión de control en la creación y administración de dependencias. En lugar de que una clase sea responsable de crear sus propias dependencias, permite que estas le sean inyectadas desde el exterior. Esto reduce el acoplamiento entre los componentes, ya que las dependencias pueden ser cambiadas o modificadas fácilmente, lo que es útil para la realización de pruebas unitarias. Este patrón es esencial en sistemas que requieren flexibilidad y escalabilidad, ya que facilita la configuración de los componentes en tiempo de ejecución y mejora la mantenibilidad del código (Martin, 2003).

2.4.9. Patrón *Facade*

El patrón *facade* proporciona una interfaz simplificada para un sistema complejo de clases, módulos o subsistemas. Este patrón es útil cuando se quiere ocultar la complejidad del sistema a los usuarios o clientes externos, ofreciendo un punto de acceso único y más sencillo. Facilita la interacción con sistemas grandes y mejora la separación de responsabilidades, lo que reduce el acoplamiento entre el cliente y los subsistemas internos (Gamma et al., 1994).

2.4.10. Patrón *Command*

El patrón *command* encapsula una solicitud como un objeto, permitiendo parametrizar clientes con operaciones, realizar, deshacer y rehacer acciones, además de soportar operaciones complejas como las transacciones. Es útil en sistemas que requieren operaciones reversibles o en los que se ejecutan acciones en diferentes contextos, como interfaces de usuario con botones de "Deshacer" o sistemas que deben gestionar una serie de operaciones (Gamma et al., 1994).

2.4.11. Patrón *Model-View-Controller*

El patrón *model-view-controller* (MVC) es una arquitectura ampliamente utilizada en el diseño de *software* interactivo y dinámico, especialmente en aplicaciones web. Este patrón separa la lógica de la aplicación en tres componentes principales: el Modelo, que representa los datos y la lógica de negocio; la Vista, encargada de la interfaz de usuario y la presentación de la información; y el Controlador, que actúa como intermediario entre el Modelo y la Vista, manejando las interacciones del usuario y actualizando tanto los datos como la interfaz.

Este patrón es clave para lograr una separación de responsabilidades, lo que facilita el mantenimiento y la escalabilidad del sistema. Además, permite modificar o extender cada uno de los componentes de manera independiente sin afectar al resto del sistema, lo que incrementa la flexibilidad y favorece la reutilización del código.

2.5. Axiomas fundamentales de diseño

Los axiomas fundamentales de diseño son principios rectores que guían la creación de *software* modular, escalable y mantenible. Estos axiomas proporcionan una base conceptual sólida para diseñar sistemas que puedan evolucionar y adaptarse a largo plazo, minimizando la complejidad y garantizando un alto rendimiento.

2.5.1. Axioma de separación de responsabilidades (SoC)

El axioma de separación de responsabilidades (soc) sostiene que cada componente o módulo de un sistema debe cumplir una única responsabilidad. Este principio ayuda a evitar que el código se vuelva intrincado o confuso (lo que se conoce como *spaghetti code*), lo que mejora tanto la mantenibilidad como la escalabilidad del sistema. Un ejemplo típico de SoC es el patrón *model-view-controller*, que segmenta las preocupaciones del sistema en capas separadas (modelo de datos, lógica de negocio y presentación), facilitando su comprensión y modificación sin impacto en otras capas (Pressman, 2009).

2.5.2. Axioma de modularidad

El axioma de modularidad promueve la creación de sistemas compuestos por módulos independientes, cada uno de los cuales tiene una responsabilidad clara y definida. Esto permite que los módulos puedan desarrollarse, probarse y modificarse por separado, reduciendo los riesgos de interferencias entre componentes. La modularidad también facilita la evolución del sistema, ya que los cambios en un módulo no afectan a otros, siempre que se respeten las interfaces públicas establecidas (Yourdon y Constantine, 1979).

2.5.3. Axioma de cohesión

El axioma de cohesión afirma que los elementos dentro de un módulo deben estar altamente relacionados y trabajar juntos para cumplir una única función claramente definida. Una alta cohesión dentro de los módulos facilita el mantenimiento, mejora la claridad del código y asegura que los módulos puedan ser entendidos y modificados con mayor facilidad. Además, un diseño con alta cohesión reduce la probabilidad de que se introduzcan errores al cambiar o actualizar componentes específicos (Pressman, 2009).

2.5.4. Axioma de acoplamiento débil

El axioma de acoplamiento débil sostiene que las interdependencias entre módulos deben ser mínimas. Un bajo acoplamiento significa que los cambios en un módulo tendrán un impacto mínimo o nulo en otros, lo que mejora la flexibilidad y escalabilidad del sistema. Esto se logra utilizando interfaces bien definidas y abstracciones, evitando dependencias directas entre módulos. Un acoplamiento débil es esencial para facilitar el mantenimiento y la expansión del sistema sin causar efectos colaterales significativos (Booch, 1994).

2.5.5. Axioma de reutilización

El axioma de reutilización establece que las soluciones de *software* deben diseñarse con el objetivo de ser reutilizadas en diferentes contextos o proyectos. Un diseño reutilizable permite aprovechar componentes ya probados y mejorados, lo que no solo incrementa la eficiencia del desarrollo, sino que también reduce los costos y el tiempo de entrega. Este axioma fomenta la creación de bibliotecas de componentes genéricos y módulos que pueden integrarse en múltiples sistemas, maximizando el retorno de inversión en desarrollo de *software* (Gamma et al., 1994).

2.6. Olores de *software*

El concepto de “olores de *software*” fue introducido por Kent Beck y popularizado por Martin Fowler para describir indicadores de problemas potenciales en el código que, aunque no generan errores inmediatos, indican que el diseño o la implementación podrían afectar negativamente la mantenibilidad, escalabilidad y flexibilidad del sistema a largo plazo. Estos olores, si no se abordan, tienden a acumularse y aumentan la deuda técnica, lo que hace más difícil y costoso el mantenimiento y la evolución del *software* (Fowler, 1999).

2.6.1. Código duplicado

El código duplicado es uno de los olores de *software* más comunes y perjudiciales. Ocurre cuando fragmentos de código idénticos o similares se repiten en múltiples partes del sistema. Esto incrementa el esfuerzo de mantenimiento, ya que cualquier cambio en el código duplicado debe replicarse en cada una de las ubicaciones, aumentando el riesgo de inconsistencias y errores. Para mitigar este olor, se recomienda refactorizar el código duplicado en métodos reutilizables o emplear patrones de diseño como patrón *template method* o patrón *strategy*, que permiten reutilizar lógicas similares sin duplicar el código (Fowler, 2019; Gamma et al., 1994).

2.6.2. Funciones largas y clases "Dios"

Las funciones largas y clases "dios" son olores que aparecen cuando un método o una clase realiza múltiples tareas o tiene demasiada responsabilidad. Esto dificulta la comprensión del código y su mantenimiento, además de generar una alta dependencia entre distintos módulos. Las clases "Dios" agrupan demasiada funcionalidad, lo que reduce la cohesión y aumenta el acoplamiento entre módulos. Para corregir estos olores, es esencial aplicar el principio de responsabilidad única, refactorizando el código para dividir funciones largas en subfunciones más pequeñas y segmentando las clases grandes en módulos más específicos y cohesionados (Martin, 2003; Fowler, 1999).

2.6.3. Dependencias circulares

Las dependencias circulares ocurren cuando dos o más módulos dependen entre sí, formando un ciclo. Esto complica la evolución del sistema y dificulta su mantenimiento, ya que cualquier cambio en uno de los módulos puede afectar a otros dentro del ciclo. Para eliminar este olor, se recomienda aplicar el principio de inversión de dependencias, desacoplando los módulos mediante el uso de interfaces o clases abstractas, y utilizar el patrón *dependency injection* o patrón *factory method* para gestionar las dependencias sin crear vínculos directos entre módulos (Martin, 2003).

2.6.4. Datos perversos

El olor datos perversos ocurre cuando se abusa de tipos de datos primitivos como enteros, cadenas o booleanos en lugar de crear tipos específicos que encapsulen los comportamientos asociados. Esta práctica reduce la claridad del código y su extensibilidad. Para corregir este olor, se deben crear objetos de valor o clases específicas que encapsulen los comportamientos relacionados con esos datos primitivos, mejorando así la expresividad, la reutilización y la mantenibilidad del código (Fowler, 1999).

2.6.5. Envidia de funciones

La envidia de funciones sucede cuando un método accede con frecuencia a los datos de otra clase para realizar sus cálculos o tareas, lo que indica que la funcionalidad debería residir en la clase que posee esos datos. Este olor se resuelve transfiriendo la funcionalidad a la clase propietaria de los datos, mejorando la axioma de cohesión y reduciendo las dependencias innecesarias entre clases, lo que facilita el mantenimiento del código (Fowler, 1999).

2.6.6. Clases poco cohesivas

Las clases poco cohesivas aparecen cuando una clase depende en exceso de los detalles internos de otra, lo que genera un acoplamiento innecesario. Esto ocurre cuando una clase tiene acceso a los atributos privados o métodos de otra, rompiendo el principio de encapsulación. Para resolver este olor, se deben reducir las interacciones directas entre las clases, utilizando interfaces o métodos públicos que limiten el acceso y respeten la encapsulación, garantizando así una menor dependencia entre las clases (Martin, 2003).

2.7. Deuda técnica

La deuda técnica es un concepto que describe el costo acumulado resultante de decisiones de diseño o implementación que priorizan soluciones rápidas pero subóptimas, con el objetivo de acelerar el desarrollo en el corto plazo. Introducido por Ward Cunningham (1992), este término establece una analogía con las finanzas; al igual que una deuda económica, estas soluciones rápidas representan un "préstamo" que deberá pagarse en el futuro mediante un mayor esfuerzo de mantenimiento y refactorización. Si no se gestiona adecuadamente, la deuda técnica puede aumentar exponencialmente, poniendo en riesgo la sostenibilidad del proyecto, afectando su escalabilidad y dificultando la incorporación de nuevas funcionalidades.

2.7.1. Tipos de deuda técnica

Existen dos tipos principales de deuda técnica:

- Deuda deliberada: este tipo de deuda surge de decisiones conscientes en las que se compromete la calidad del código para cumplir con plazos de entrega o lanzar versiones tempranas del producto. Se asume que la deuda será "pagada" posteriormente, cuando el equipo pueda dedicar tiempo a refactorizar el código y mejorar su calidad.
- Deuda inadvertida: esta deuda ocurre sin intención, debido a la falta de planificación, el conocimiento técnico limitado o decisiones de diseño incorrectas. Este tipo de deuda se descubre con el tiempo, cuando surgen problemas de mantenibilidad y escalabilidad.

2.7.2. Impactos de la deuda técnica

A medida que la deuda técnica se acumula, su impacto puede ser significativo y afectar múltiples aspectos del ciclo de vida del *software*:

- Aumento del esfuerzo de mantenimiento: la acumulación de deuda técnica incrementa tanto el tiempo como los recursos necesarios para mantener el código, elevando los costos a largo plazo.
- Mayor riesgo de errores: la deuda técnica puede introducir inconsistencias en el código, lo que aumenta la probabilidad de errores y fallos en el sistema.
- Pérdida de productividad: los desarrolladores pasan más tiempo tratando de entender y corregir código mal diseñado, lo que reduce la eficiencia del equipo y ralentiza el progreso general del proyecto.

2.7.3. Gestión de la deuda técnica

Gestionar la deuda técnica de manera efectiva es crucial para garantizar la sostenibilidad y el éxito a largo plazo de cualquier proyecto de *software*. Existen diversas estrategias que las organizaciones pueden implementar para controlar y mitigar el impacto de la deuda técnica:

- Priorizar refactorizaciones en áreas críticas: es fundamental identificar las áreas del sistema con mayor deuda técnica y priorizar su refactorización para evitar que la deuda siga acumulándose y afectando a otras partes del sistema.

- Monitorear la deuda técnica de manera periódica: realizar revisiones regulares del código y evaluar su calidad ayuda a identificar la deuda técnica antes de que se vuelva inmanejable.
- Equilibrar la velocidad de entrega con la calidad a largo plazo: es esencial encontrar un balance entre cumplir con plazos de entrega y mantener un nivel adecuado de calidad en el código, evitando la acumulación innecesaria de deuda técnica (Fowler, 2019).

2.8. Pensamiento convergente y divergente en el diseño

El pensamiento divergente y pensamiento convergente son enfoques esenciales en el proceso de diseño de *software*, ya que juntos permiten a los diseñadores generar soluciones innovadoras y seleccionar las más adecuadas para el proyecto. Ambos tipos de pensamiento son complementarios y fundamentales para lograr un diseño más integral y . Mientras el pensamiento divergente abre el campo a nuevas ideas y soluciones; el pensamiento convergente reduce y refina estas opciones, mejorando los recursos y asegurando la implementación de soluciones prácticas y efectivas.

2.8.1. Pensamiento divergente

El pensamiento divergente es un enfoque que promueve la generación de múltiples ideas sin limitaciones. Es particularmente valioso en las etapas iniciales del diseño de *software*, cuando se exploran alternativas para la arquitectura, la integración de sistemas o la funcionalidad. Este enfoque fomenta la creatividad y permite considerar una amplia gama de posibles soluciones antes de establecer una dirección final. Según Guilford (1950), este tipo de pensamiento es clave para la innovación, ya que ayuda a descubrir enfoques no convencionales que pueden aportar valor significativo al proyecto.

2.8.2. Pensamiento convergente

El pensamiento convergente se centra en la evaluación y selección de las soluciones más viables generadas durante la fase divergente. Este enfoque utiliza criterios como la eficiencia técnica, escalabilidad y mantenibilidad para reducir las opciones y asegurar que el diseño final sea óptimo y cumpla con los objetivos del proyecto. Es esencial en las fases finales del diseño, ya que permite tomar decisiones fundamentadas y prácticas para la implementación exitosa del sistema, maximizando el impacto positivo de las ideas generadas durante la fase divergente.

2.9. Verificación y validación del *software* (V&V)

La verificación y validación (V&V) son procesos críticos para garantizar que el *software* cumple con los requisitos especificados y satisface las expectativas de los usuarios. Según IEEE 1012, la verificación asegura que el *software* se ha desarrollado conforme a las especificaciones, mientras que la validación confirma que el sistema satisface los requisitos funcionales y no funcionales establecidos.

Estos procesos se ejecutan a lo largo de todo el ciclo de vida del *software*, desde el análisis de requisitos hasta las pruebas finales, permitiendo identificar y corregir errores en etapas tempranas y reduciendo riesgos futuros. Las técnicas de V&V incluyen revisiones, auditorías y pruebas formales estructuradas. En este TFG, el marco de V&V tiene como principales objetivos:

- Detectar y corregir errores de diseño en etapas tempranas.
- Validar que se cumplen todos los requisitos funcionales y no funcionales, especialmente los relacionados con la interoperabilidad.
- Mitigar riesgos futuros, asegurando que el sistema cumple con los estándares de calidad establecidos.

2.9.1. Verificación del *software*

La verificación se enfoca en garantizar que los productos intermedios (como el diseño y el código) cumplan con las especificaciones y estándares. Este proceso incluye revisiones de código, auditorías y pruebas unitarias, que permiten identificar defectos técnicos antes de la implementación. Según ISO/IEC/IEEE 15288, la verificación es un proceso continuo que se realiza en cada fase del desarrollo.

2.9.2. Validación del *software*

La validación asegura que el *software* satisface las expectativas del usuario final. Las pruebas de validación se centran en evaluar la funcionalidad, el rendimiento y la interoperabilidad del sistema en condiciones reales de operación. Estas pruebas confirman que el sistema cumple plenamente con los requisitos establecidos. En este proyecto, se utilizan dos técnicas principales para la verificación y validación del diseño propuesto:

2.9.3. Matriz de cobertura de requerimientos

Una herramienta clave en V&V es la matriz de cobertura de requerimientos, la cual garantiza que todos los requisitos están reflejados adecuadamente en el diseño y el desarrollo del *software*. Según Sommerville (2011), la trazabilidad de los requisitos es fundamental para asegurar que ningún aspecto importante quede fuera del proceso.

- Objetivo: verificar que todos los requisitos funcionales y no funcionales están cubiertos en el diseño e implementación.
- Método: relacionar cada requisito con su funcionalidad correspondiente en una matriz, lo que facilita una evaluación rápida y precisa de su cobertura y trazabilidad.

2.9.4. Historias de usuario

Las historias de usuario son descripciones breves y enfocadas en el usuario que detallan necesidades o funcionalidades específicas que el sistema debe satisfacer. Según Cohn (2004), una historia de usuario sigue la estructura básica: "Como [rol del usuario], quiero [funcionalidad] para [beneficio o resultado deseado]". Este enfoque proporciona un medio simple pero poderoso para capturar requisitos desde la perspectiva del usuario, promoviendo una mejor comprensión de las necesidades y objetivos del sistema. Las historias de usuario son relevantes en proyectos de software porque facilitan la comunicación entre los interesados, priorizan la experiencia del usuario y contribuyen a garantizar que los desarrollos estén alineados con las metas generales del proyecto. En el contexto de este estudio, su importancia radica en que permiten definir y validar interacciones clave del sistema, especialmente en términos de interoperabilidad y eliminación de redundancias.

2.10. Evolución del *software*

La evolución del *software* es un proceso continuo que permite que los sistemas se ajusten a los cambios en los requisitos o en el entorno operativo. Según ISO/IEC 14764, este proceso debe estar planificado y estructurado para asegurar que el sistema continúe funcionando a lo largo del tiempo, adaptándose a nuevos desafíos tecnológicos y necesidades de los usuarios. La evolución incluye tres actividades principales:

- Identificar y corregir errores que se detectan tras la implementación.
- Modificar el sistema para adaptarlo a nuevos entornos tecnológicos o de *hardware*.
- Incorporar nuevas funcionalidades o mejorar las ya existentes para cumplir con los requisitos cambiantes del negocio.

2.10.1. Proceso cíclico de evolución

Según Sommerville (2005), la evolución del *software* es cíclica y dinámica, ya que responde constantemente a la aparición de nuevos requerimientos, mejoras o cambios inesperados. Este proceso involucra varias etapas:

- Peticiones de cambio: cualquier solicitud de modificación debe ser evaluada cuidadosamente.
- Análisis de impacto: se determina cómo afectará el cambio al sistema y qué ajustes son necesarios.
- Planificación de versiones: los cambios se agrupan en versiones que se liberan según las prioridades establecidas.
- Implementación de cambios: las modificaciones se realizan en el código o en la configuración del sistema.
- Liberación del sistema: tras realizar pruebas exhaustivas, el sistema actualizado se pone a disposición de los usuarios.

2.11. Notación para graficar el diseño

La representación gráfica de la arquitectura de *software* es esencial para comprender la estructura y las interacciones del sistema. Las notaciones gráficas permiten una evaluación clara y temprana de posibles problemas de diseño, mejorando la planificación y asegurando la coherencia entre los distintos componentes del sistema. Las notaciones más relevantes para graficar el diseño de *software* son UML.

UML es una herramienta clave para modelar tanto la estructura estática como el comportamiento dinámico del sistema. A través de diagramas como los de clases, secuencia e historias de usuario, UML ofrece una representación detallada de la complejidad técnica y las relaciones entre los componentes, lo que facilita la detección de inconsistencias y dependencias antes de la implementación. Esto contribuye a que el diseño del sistema esté alineado con los objetivos, asegurando una arquitectura escalable y mantenible (Booch et al., 1999).

2.11.1. Integración entre sistemas de *software*

La integración entre sistemas de *software* es esencial para mejorar la interoperabilidad y eficiencia en cualquier organización. Según Rivas (2006), la integración de sistemas implica conectar y unificar plataformas diversas, ya sea para incorporar aplicaciones heredadas o para adaptar nuevas tecnologías. El objetivo es garantizar que los sistemas funcionen en conjunto sin duplicar funciones ni comprometer la coherencia de los datos.

2.11.2. Problemas de integración de sistemas

Los problemas de integración suelen surgir debido a la autonomía y heterogeneidad de los sistemas involucrados. Según Rivas (2006), la autonomía se refiere al hecho de que los sistemas son diseñados, mantenidos y operados de manera independiente, sin prever una integración futura. Esto da lugar a la heterogeneidad, ya que los sistemas utilizan tecnologías y estándares diferentes, lo que dificulta la comunicación entre ellos. Para superar estos desafíos, es necesario establecer criterios de integración claros que abarquen la infraestructura, los modelos de datos y la gestión de la información. Estos criterios aseguran que la integración sea efectiva, permitiendo que los sistemas se comuniquen y compartan datos de manera fluida.

2.12. UML

El Unified Modeling Language (UML) es un lenguaje estandarizado que permite la representación de sistemas de *software* a través de diagramas que modelan tanto la estructura estática como el comportamiento dinámico. Facilita la comprensión y comunicación del diseño entre equipos de desarrollo y otros *stakeholders*, mejorando la interoperabilidad y la planificación del sistema. Como estándar internacional, es ampliamente utilizado en la ingeniería de *software* para asegurar la correcta documentación y diseño del sistema (Booch, Rumbaugh y Jacobson, 1999).

2.12.1. Diagrama de clases

El diagrama de clases es una herramienta en UML que visualiza la estructura estática del sistema, mostrando las clases, atributos, métodos y las relaciones entre los objetos. Este diagrama permite la identificación de asociaciones, herencias y dependencias entre las distintas clases, proporcionando una visión clara de la organización y composición del sistema (Booch et al., 1999).

2.12.2. Diagrama de secuencia

El diagrama de secuencia es un diagrama de UML que representa el comportamiento dinámico, enfocándose en el intercambio de mensajes entre los objetos a lo largo del tiempo. Este diagrama es útil para visualizar las interacciones y flujos de trabajo entre los componentes, describiendo cómo los objetos se comunican y en qué orden lo hacen (Fowler, 2004).

2.12.3. Diagrama de historias de usuario

El diagrama de historias de usuario en UML detalla las funcionalidades clave del sistema desde la perspectiva de los actores externos, como usuarios o sistemas que interactúan con él. Representa los historias de usuario y las relaciones entre los actores y el sistema, proporcionando una visión clara de los requerimientos funcionales y su interacción (Cockburn, 2001).

2.12.4. Diagrama de actividades

El diagrama de actividades modela los flujos de trabajo o procesos dentro del sistema, representando el flujo de ejecución de una operación o conjunto de tareas. Este diagrama permite la visualización de procesos secuenciales y paralelos, facilitando el análisis de cómo se organizan y controlan las actividades del sistema (Booch et al., 1999).

2.12.5. Diagrama de estados

El diagrama de estados muestra los diferentes estados que puede adoptar un objeto a lo largo de su ciclo de vida y las transiciones que ocurren entre esos. Este diagrama es útil para modelar el comportamiento de los objetos que dependen de eventos o condiciones específicas, ayudando a identificar los posibles estados de los componentes del sistema (Booch et al., 1999).

2.12.6. Diagrama de componentes

El diagrama de componentes representa la organización física del sistema y las interacciones entre los diferentes componentes de *software*. Este diagrama muestra cómo los subsistemas o módulos están organizados e interrelacionados, proporcionando una vista clara de la arquitectura modular del sistema (Booch et al., 1999).

2.12.7. Diagrama de despliegue

El diagrama de despliegue representa la distribución física de los elementos del sistema en nodos de *hardware*, servidores o redes. Este diagrama permite entender cómo los componentes de *software* se implementan y distribuyen en la infraestructura física, facilitando la visualización de las relaciones entre el *software* y el *hardware* (Booch et al., 1999).

2.13. Tipos de atributos

Los atributos de una clase pueden ser de diversos tipos, lo que define la naturaleza de los datos que almacenan. Estos tipos incluyen:

- *Boolean*: representa valores de verdadero o falso. Ejemplos: *True*, *False*.
- *Integer*: números enteros, como 5 o 30.
- *Float*: números decimales con parte fraccionaria, como 2.33 o 1050.50.
- *String*: cadenas de texto, como Nombre o Descripción.

2.14. Funciones

Las funciones definen el comportamiento de las clases, indicando las acciones que pueden realizar. Una función puede retornar un valor o simplemente ejecutar una acción. Estas funciones se definen dentro de las clases para manipular sus atributos y ejecutar las operaciones necesarias, como lo describen Hamilton y Miles (2015). Por ejemplo, una función llamada `calcularPrecio()` podría retornar el precio total de un producto aplicando un descuento.

2.15. Visibilidad

La visibilidad de los atributos y funciones determina si son accesibles desde otras clases o solo desde la clase en la que se definen. Existen varios niveles de visibilidad, que incluyen:

- Pública (*public*): los atributos o funciones son accesibles desde cualquier clase.
- Privada (*private*): solo son accesibles desde la propia clase.
- Protegida (*protected*): son accesibles desde la propia clase y las clases derivadas.

Esto permite un control adecuado sobre la interacción de los componentes de una clase, asegurando encapsulamiento y seguridad en el manejo de datos.

2.16. Multiplicidad

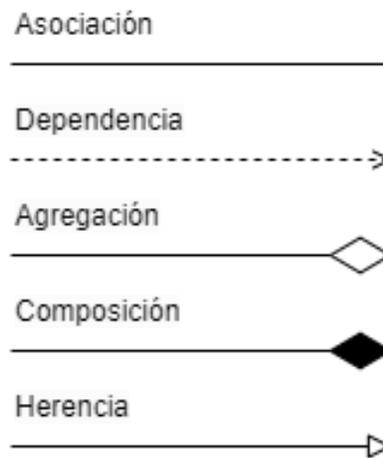
La multiplicidad indica el número de instancias de una clase que pueden estar asociadas a una instancia de otra clase. Es un concepto clave en la relación entre objetos, ya que especifica las posibles cantidades mínimas y máximas de objetos que pueden existir en la relación. Por ejemplo, un cliente puede tener uno o varios pedidos, pero cada pedido pertenece a un solo cliente.

2.17. Relaciones entre clases

Las relaciones entre clases describen las interacciones entre diferentes clases y cómo se conectan. Estas pueden ser de varios tipos:

- Asociación: representa una relación en la que una clase colabora con otra de forma prolongada.
- Dependencia: indica una relación temporal, donde una clase utiliza otra durante un tiempo limitado.
- Agregación: define que una clase comparte referencia con objetos de otra de manera que ambos pueden existir por separado.
- Composición: es una forma más fuerte de agregación, donde la vida de un objeto depende de la existencia de otro.
- Herencia: permite que una clase herede atributos y métodos de otra, facilitando la reutilización y extensión de código.

Figura 3
Notación UML de las relaciones



Nota. Fuente: Elaboración propia.

2.18. Métodos de integración

Existen varios enfoques para la integración de sistemas, cada uno con sus ventajas y desventajas, dependiendo de las necesidades específicas del proyecto. Según Rivas (2006) y Medina (2019), los métodos más comunes son:

- Integración horizontal: se centra en la transmisión de mensajes entre sistemas mediante servicios de transformación y mapeo de datos. Este enfoque permite agregar o modificar componentes sin interrumpir el sistema general, lo que facilita la flexibilidad y la modularidad del sistema.
- Integración vertical: conecta sistemas a través de soluciones funcionales específicas para una tarea concreta. Aunque es rápida y económica, esta integración es menos escalable, ya que no facilita la interoperabilidad completa entre todos los sistemas, lo que puede crear silos de información.
- Integración mediante formato de datos común: este método elimina la necesidad de adaptadores al estandarizar los formatos de datos entre los sistemas. Al utilizar un formato común, se automatiza el flujo de datos entre sistemas, lo que promueve la interoperabilidad y asegura la coherencia en los datos intercambiados.

2.19. SaaS

El Software as a service (SaaS) es un modelo de distribución de *software* en el que las aplicaciones están alojadas y gestionadas en servidores remotos, accesibles a través de la nube. Según ISO/IEC 17788 (2014), SaaS permite a los usuarios acceder a aplicaciones sin necesidad de gestionar la infraestructura subyacente, lo que facilita la escalabilidad, flexibilidad y la centralización de las actualizaciones. Este modelo es ideal para entornos en los que se busca eficiencia operativa y reducción de costos, al permitir que las actualizaciones se gestionen de forma centralizada y las aplicaciones sean accesibles desde cualquier ubicación. El SaaS aporta varios beneficios clave para la gestión de *software* y su evolución, entre ellos:

- Las actualizaciones de *software* se gestionan de manera centralizada, lo que asegura que todos los usuarios trabajen con la versión más reciente y evita problemas de compatibilidad.
- SaaS elimina la necesidad de adquirir y mantener infraestructura física, ofreciendo una solución más económica para las organizaciones.
- Permite ampliar o reducir servicios según las necesidades de los usuarios, adaptándose rápidamente a los cambios en la demanda (Armbrust et al., 2010).
- Las API permiten el acceso a datos entre diferentes sistemas, facilitando la intercomunicación entre funcionalidades diversas dentro de una organización.

2.19.1. Arquitectura de microservicios

La arquitectura de microservicios organiza el *software* en pequeños servicios autónomos que operan de forma independiente, aunque están interconectados a través de API. Según Newman (2015), cada microservicio está diseñado para realizar una función específica y puede evolucionar o escalar sin afectar al resto del sistema. Las características clave de esta arquitectura son:

- Independencia de componentes: los microservicios son autónomos, lo que facilita su despliegue y mantenimiento sin interferir con otros servicios.
- Escalabilidad horizontal: permite agregar más instancias de un servicio para gestionar cargas mayores sin afectar al rendimiento global del sistema (Fowler y Lewis, 2014).

2.19.2. API

Las Application Programming Interface (API) son un conjunto de definiciones y protocolos que permiten la comunicación entre diferentes sistemas o módulos de *software*. Según ISO/IEC 18384 (2016), facilitan la interoperabilidad entre sistemas distribuidos, permitiendo que componentes heterogéneos se integren sin estar profundamente acoplados; son esenciales, ya que facilitan la comunicación entre módulos o servicios, incluso cuando emplean tecnologías diferentes. Los tipos más comunes son:

2.19.3. REST

Un Representational State Transfer (REST) es un estilo arquitectónico que facilita la comunicación entre sistemas a través de métodos HTTP. Fielding (2000) lo define como un enfoque ligero y fácil de implementar, ideal para aplicaciones web distribuidas.

2.19.4. SOAP

Un Simple Object Access Protocol (SOAP) es un protocolo que permite el intercambio de información estructurada entre sistemas, garantizando que los datos se transmitan de manera segura y confiable (Box et al., 2000).

2.20. Seguridad en API

La seguridad en API es fundamental, especialmente en entornos distribuidos o basados en la nube. OAuth 2.0 y OpenID Connect son dos estándares que permiten gestionar de forma segura la autenticación y autorización en aplicaciones conectadas a través de API. Según RFC 6749, OAuth 2.0 proporciona un marco robusto que permite a los usuarios compartir recursos entre aplicaciones sin exponer credenciales sensibles, asegurando el acceso controlado a los datos.

2.21. Kyte

Kyte es una plataforma digital diseñada específicamente para pequeñas y medianas empresas, con el objetivo de simplificar y gestionar las operaciones comerciales en línea. Esta herramienta combina diversas funcionalidades esenciales para facilitar la administración de negocios que tienen presencia digital y ventas a distancia. Entre sus principales características destacan:

- Gestión de inventarios: registro detallado del stock de productos, con actualizaciones que permiten mantener un control preciso de las existencias.
- Ventas y pedidos en línea: creación de catálogos digitales que posibilitan a los clientes realizar pedidos directamente a través de la plataforma.
- Facturación y generación de reportes: emisión automatizada de comprobantes de pago y análisis del rendimiento comercial mediante reportes estadísticos.
- Recepción de pagos: integración de múltiples métodos de pago, como tarjetas de crédito, transferencias bancarias y pagos en efectivo, lo que amplía la accesibilidad y conveniencia para los clientes.

2.22. POS

Un *point of sale* (POS) o sistema de punto de venta, es una solución tecnológica que combina hardware y software para gestionar las operaciones comerciales en tiendas físicas. Este tipo de sistema está diseñado para facilitar la realización de transacciones, optimizar la experiencia del cliente y apoyar la administración diaria de un negocio. Entre sus funcionalidades principales se encuentran:

- Procesamiento de pagos: soporte para diferentes métodos de pago, incluyendo efectivo, tarjetas de crédito y débito, y otros medios digitales.
- Gestión de inventarios: actualización automática del inventario en tiempo real cada vez que se registra una venta, lo que reduce errores y facilita el control.
- Generación de reportes: creación de informes detallados sobre ventas, rendimiento de productos y comportamiento del cliente, útiles para la toma de decisiones estratégicas.

2.23. FACEL

FACEL es un sistema dedicado a la emisión y administración de facturas electrónicas, cumpliendo con los estándares y regulaciones fiscales que exigen la digitalización de documentos tributarios. Su implementación responde a las necesidades de las empresas de modernizar sus procesos administrativos y garantizar la trazabilidad de sus transacciones financieras. Sus principales funcionalidades incluyen:

- Centralización de facturación: Registro y control automatizado de facturas electrónicas en una única plataforma.
- Conformidad fiscal: Garantiza que las facturas emitidas cumplan con los requisitos legales impuestos por las autoridades tributarias.
- Consulta y almacenamiento en línea: permite almacenar y consultar comprobantes electrónicos, facilitando la gestión documental y el acceso a información histórica.

2.24. Migración de datos

La migración de datos es el proceso de trasladar información de un sistema de origen a otro, con el propósito de consolidar, actualizar o modernizar la infraestructura tecnológica de una organización. Este proceso incluye las etapas de extracción, transformación y carga (ETL), asegurando que la información se transfiera de manera precisa y coherente entre sistemas antiguos y nuevos. Además, implica la validación de la integridad de los datos y su adaptación a la estructura del nuevo sistema, lo cual puede incluir la limpieza de datos para eliminar duplicaciones y la normalización para mantener la consistencia. La migración de datos es fundamental para mantener la continuidad operativa, permitiendo integrar tanto datos históricos como actuales en un entorno más . Dependiendo de la criticidad de los datos y las necesidades de la organización, puede llevarse a cabo de manera masiva (*big bang*) o de forma gradual, según Pressman (2005), Kimmel (2008) y Cockburn (2001).

2.25. Automatización y estándares de datos

La automatización mediante API facilita el flujo continuo de información entre sistemas distribuidos, mejorando la eficiencia operativa y eliminando la duplicación de tareas. La estandarización de los formatos de datos, como JSON y XML, asegura que la información se transmita y reciba de manera consistente entre los servicios (Chappell, 2016). Además, el uso de formatos de datos comunes promueve la interoperabilidad entre sistemas sin necesidad de complejos adaptadores, lo que simplifica la integración y reduce la complejidad técnica.

2.26. RPA

La Automatización Robótica de Procesos (RPA) se refiere a la tecnología que permite la automatización de procesos de negocio mediante el uso de *software* o robots que imitan las acciones de los humanos en la interacción con sistemas digitales. Automatiza tareas repetitivas y estructuradas, como la entrada de datos, la gestión de correos electrónicos, la extracción de información de documentos, y la integración de sistemas que no tienen una interfaz directa de integración.

2.27. Desactivación progresiva de sistemas

La desactivación progresiva de sistemas es un enfoque controlado para retirar gradualmente sistemas antiguos de una infraestructura tecnológica, mientras se introduce un sistema nuevo o actualizado. Este proceso permite que los sistemas legados sigan operando en paralelo con la nueva solución durante un período de transición, asegurando la continuidad operativa y la integridad de los datos. La desactivación progresiva es fundamental para minimizar los riesgos de interrupciones en las operaciones diarias de una organización, ya que permite identificar y corregir problemas antes de la desactivación completa de los sistemas antiguos. Además, facilita la transferencia de funciones, la migración de datos y la capacitación del personal en el nuevo sistema, lo que reduce la resistencia al cambio y asegura una integración más fluida de la nueva plataforma tecnológica (Pressman, 2005; Kimmel, 2008).

2.28. Visualización de datos y KPI

La visualización de datos y kpi es el proceso de representar información y métricas clave de forma gráfica para facilitar su interpretación y análisis. Esto incluye el uso de gráficos, tablas, *dashboards* y otros elementos visuales que permiten a los usuarios entender rápidamente grandes volúmenes de datos. Por otro lado, los KPI son métricas específicas que miden el desempeño de procesos y objetivos críticos para una organización. La visualización de KPI es esencial para el seguimiento en tiempo real de la eficiencia operativa y la toma de decisiones estratégicas, ya que proporciona una visión clara y accesible de los indicadores más relevantes para cada rol dentro de la empresa. Una correcta visualización de datos y KPI permite detectar tendencias, identificar áreas de mejora y tomar decisiones basadas en datos de forma ágil, mejorando la gestión de recursos y la efectividad de las acciones organizacionales (Few, 2012; Eckerson, 2011).

2.29. *Insights*

Los *insights* son interpretaciones profundas y valiosas que se obtienen a partir del análisis de datos, observaciones o experiencias, proporcionando una comprensión más clara sobre un fenómeno, tendencia o comportamiento específico. Estos hallazgos permiten identificar patrones ocultos, relaciones significativas y oportunidades de mejora que no serían evidentes con un análisis superficial. En el contexto empresarial, son fundamentales para la toma de decisiones estratégicas, ya que proporcionan una base sólida para ajustar estrategias, mejorar productos o servicios y mejorar procesos internos. La capacidad de generar *insights* depende del uso adecuado de herramientas de análisis de datos, técnicas estadísticas y la experiencia contextual, facilitando una toma de decisiones más informada y precisa.

3. Marco metodológico

Este capítulo describe los elementos metodológicos que guían el desarrollo del Trabajo Final de Graduación (TFG). Se detallan el tipo, el enfoque, el alcance y el diseño de la investigación. También se incluyen las fuentes de datos, los sujetos participantes, las variables clave y los instrumentos empleados para abordar los problemas identificados en los sistemas de Hortiluma.

3.1. Tipo de investigación

Según Hernández-Sampieri et al. (2014): “la investigación es un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un fenómeno o problema” (p. 4). En este proyecto, la investigación está orientada a mejorar la interoperabilidad de las funcionalidades actuales en los sistemas de Hortiluma, eliminando tareas y datos duplicados, tal como se describe en los objetivos. La Tabla 4, presenta los tipos de investigación según Hernández-Sampieri et al. (2014) y su relevancia para este proyecto:

Tabla 4
Tipos de investigación

Tipo de Investigación	Descripción	Relevancia
Investigación básica	Genera conocimientos teóricos sin una aplicación práctica inmediata.	No es adecuada para este proyecto, ya que no aborda la resolución de un problema práctico como la mejora de la interoperabilidad en Hortiluma.
Investigación aplicada	Resuelve problemas prácticos mediante la aplicación de conocimientos previos.	Es la opción adecuada, ya que el proyecto busca mejorar la interoperabilidad y eliminar tareas y datos duplicados en los sistemas de Hortiluma, desarrollando soluciones concretas y medibles.

Nota. Fuente: Elaboración propia.

Se ha seleccionado la investigación aplicada porque el proyecto busca proponer una solución práctica para mejorar la interoperabilidad en los sistemas de Hortiluma, eliminando tareas y datos duplicados, tal como indica el objetivo general. Este enfoque permite el desarrollo de un diseño de *software* de alto nivel alineado con los objetivos específicos, permitiendo la identificación de brechas de interoperabilidad, la simplificación de tareas y datos duplicados, así como la validación del diseño preliminar de la arquitectura de integración.

3.2. Enfoque de la investigación

Según Hernández-Sampieri et al. (2014): “la investigación científica se concibe como un conjunto de procesos sistemáticos y empíricos que se aplican al estudio de un fenómeno; es dinámica, cambiante y evolutiva. Se puede manifestar de tres formas: cuantitativa, cualitativa y mixta” (p. 26). Cada uno se adapta a distintos objetivos y problemas de investigación. En la Tabla 5, se presentan los diferentes enfoques de investigación.

Tabla 5
Enfoques de investigación

Enfoque	Descripción	Relevancia
Cualitativo	Explora fenómenos a través de las percepciones y experiencias de los participantes. Ideal para estudios donde el problema no está completamente definido.	Es flexible y permite identificar áreas no visibles en los datos numéricos, como las interacciones y percepciones de los usuarios.
Cuantitativo	Se centra en la medición numérica de variables para probar hipótesis y generar resultados objetivos.	Proporciona resultados medibles que permiten evaluar el éxito de las soluciones mediante indicadores replicables, ideales para validar los avances del proyecto.
Mixto	Combina los enfoques cualitativo y cuantitativo, proporcionando una visión integral del problema y sus soluciones.	Permite una comprensión profunda a través del análisis cualitativo y la validación objetiva mediante datos cuantitativos.

Nota. Fuente: Elaboración propia.

Este proyecto busca mejorar la interoperabilidad en los sistemas de Hortiluma y eliminar tareas y datos duplicados, lo que requiere explorar las percepciones de los usuarios y validar las soluciones a través de métricas cuantitativas. En la Tabla 6, se detalla el enfoque investigativo seleccionado para cada uno de los objetivos específicos del proyecto:

Tabla 6
Enfoque investigativo por objetivos

Objetivo	Enfoque	Justificación
Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.	Cualitativo	El diagnóstico requiere un enfoque cualitativo para explorar cómo interactúan los sistemas y cómo perciben los usuarios la consistencia de los datos. A su vez, indicadores cuantitativos como la cantidad de tareas y datos duplicados permite evaluar objetivamente la magnitud de las brechas.
Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.	Cualitativo	El enfoque cualitativo captura las expectativas y necesidades de los actores clave mediante entrevistas y opiniones, mientras que el enfoque cuantitativo permite medir la reducción de tareas y datos duplicados, proporcionando una evaluación completa de la simplificación lograda.

Objetivo	Enfoque	Justificación
Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.	Cualitativo	Este objetivo se centra en la interpretación cualitativa de las necesidades y expectativas de los usuarios para diseñar una arquitectura preliminar que cubra las brechas de interoperabilidad. No es necesario un enfoque cuantitativo, ya que la creación del diseño depende de la evaluación de las necesidades de los actores.
Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.	Cualitativo	La validación requiere un enfoque cualitativo para evaluar cómo perciben los usuarios la eficacia del diseño, complementado por un enfoque cuantitativo que mida los avances en la interoperabilidad, la reducción de tareas duplicadas y la mejora de la eficiencia operativa.

Nota. Fuente: Elaboración propia.

El enfoque seleccionado para este proyecto es el mixto, ya que combina los beneficios de los enfoques cualitativo y cuantitativo. Dado que los objetivos incluyen la identificación de brechas de interoperabilidad y la validación del diseño de *software* de alto nivel, es fundamental explorar las percepciones de los usuarios (cualitativo) y medir objetivamente si las soluciones eliminan tareas y datos duplicados (cuantitativo). Este enfoque permite una comprensión completa del problema y garantiza la validación precisa de los resultados a través de análisis detallados y datos cuantificables.

3.3. Alcance de la investigación

Según Hernández-Sampieri et al. (2014), el alcance de una investigación determina la profundidad con la que se abordará un fenómeno o problema. Definir adecuadamente este alcance es clave, ya que establece los límites conceptuales y metodológicos que guiarán el estudio hacia los resultados esperados. En la Tabla 7, se describen los principales tipos de alcance y su relevancia:

Tabla 7
Alcances de investigación

Alcance	Descripción	Relevancia
Exploratorio	Examina problemas poco estudiados, proporcionando un marco inicial cuando hay muchas dudas o el tema no ha sido abordado previamente.	Ideal para estudios preliminares o áreas novedosas donde se busca una aproximación inicial.
Descriptivo	Especifica las características de fenómenos, situaciones o contextos, sin inferir causalidad.	Proporciona una visión detallada del fenómeno o situación. Define claramente las variables, conceptos y métodos de recolección de datos.

Alcance	Descripción	Relevancia
Correlacional	Analiza cómo se asocian las variables, sin indagar en las causas de esas relaciones.	Útil para identificar patrones de relación entre variables, proporcionando una base para estudios más profundos.
Explicativo	Busca entender las causas de un fenómeno, explorando cómo interactúan las variables entre sí.	Ideal para investigaciones que buscan comprender "por qué" y "cómo" ocurren ciertos fenómenos, proporcionando un entendimiento de causas y efectos.

Nota. Fuente: Elaboración propia.

Para este TFG, se ha adoptado un alcance descriptivo, dado que se busca detallar las experiencias, percepciones y necesidades de los usuarios de Hortiluma en relación con la interoperabilidad de sus sistemas. Este enfoque permite obtener una comprensión detallada de la situación actual dentro de la empresa, lo cual es fundamental para desarrollar soluciones efectivas que se ajusten a sus necesidades específicas. Al describir las características y problemas del sistema sin inferir causalidad, el alcance descriptivo se alinea con el enfoque de la investigación seleccionado y los objetivos.

3.4. Diseño de la investigación

De acuerdo con Hernández-Sampieri et al. (2014): “el término diseño se refiere al plan o estrategia concebida para obtener la información necesaria con el fin de responder al planteamiento del problema”. Este plan es esencial para estructurar la investigación, guiar la recolección y el análisis de los datos, así como alcanzar los objetivos. Dado el alcance de la investigación descriptivo de este proyecto, el diseño seleccionado debe permitir describir las experiencias y percepciones de los usuarios de Hortiluma, además de proponer soluciones prácticas que mejoren la interoperabilidad de los sistemas. En la Tabla 8, se presentan los diseños más relevantes según Hernández-Sampieri et al. (2014) para estudios con este enfoque.

Tabla 8
Diseños de investigación

Diseño	Descripción	Relevancia
Fenomenológico	Se centra en comprender las experiencias vividas por los individuos con respecto a un fenómeno, explorando percepciones y significados compartidos.	Aunque útil para captar las percepciones de los usuarios, no es adecuado para este TFG, ya que no aborda la implementación de soluciones prácticas.
Investigación-Acción	Combina la recolección de datos con la implementación de acciones prácticas para resolver problemas específicos. Involucra la colaboración activa de los participantes.	Es el diseño más adecuado para este TFG, ya que permite describir las percepciones de los usuarios, proponer soluciones y evaluar su implementación mediante un proceso iterativo de mejora.

Nota. Fuente: Elaboración propia.

El diseño de investigación-acción ha sido seleccionado para este TFG porque es el más adecuado para cumplir con los objetivos y se ajusta con el

Alcance de la investigación descriptivo. Este diseño permite abordar problemas prácticos en tiempo real, realizar ajustes continuos conforme se identifican nuevas necesidades y obtener resultados. El proceso iterativo garantiza que el TFG avance hacia una mejora constante en la propuesta de diseño para la interoperabilidad de las funcionalidades actuales de los sistemas de Hortiluma.

3.5. Fuentes de datos e información

En este TFG, se emplean fuentes primarias y secundarias para respaldar teórica y empíricamente la propuesta de diseño de *software* para la interoperabilidad de los sistemas de Hortiluma. Según Hernández et al. (2014), las fuentes primarias proporcionan datos de primera mano, fundamentales para ofrecer información original y directa, mientras que las fuentes secundarias proporcionan un contexto más amplio basado en la revisión y análisis de investigaciones previas.

3.5.1. Fuentes primarias

Recopilan información original específicamente para este proyecto. Son esenciales para comprender en profundidad los problemas de interoperabilidad en los sistemas actuales de Hortiluma y para diseñar soluciones prácticas. En la Tabla 9, se presentan las fuentes primarias.

Tabla 9
Fuentes primarias

Fuente	Relevancia
Conversaciones con colaboradores de Hortiluma	Proporcionan información directa sobre los desafíos específicos que enfrenta la empresa en la interoperabilidad de sus sistemas, lo que permite ajustar la propuesta de diseño a las necesidades reales de la organización.
Documentación interna de Hortiluma	Ofrece detalles clave sobre las funcionalidades, procesos y limitaciones de los sistemas actuales, fundamentales para diagnosticar brechas de interoperabilidad y desarrollar soluciones eficaces.
Libro: <i>Ingeniería del Software</i> - Sommerville (7 ^a ed.)	Proporciona principios clave de diseño y desarrollo de <i>software</i> , esenciales para estructurar el diseño arquitectónico del sistema de Hortiluma y asegurar la interoperabilidad entre sus funcionalidades.
Libro: <i>Manual de UML</i> - Kimmel	Facilita el uso de diagramas UML para modelar de manera estructurada la integración de los sistemas actuales de Hortiluma, permitiendo una visualización clara y precisa del diseño propuesto.
Libro: Hernández, R., Fernández, C., y Baptista, P. (2014). <i>Metodología de la investigación</i>	Ofrece una base metodológica sólida para estructurar el TFG, guiando el enfoque metodológico y asegurando la alineación del proyecto con principios probados de investigación científica.

Fuente	Relevancia
Proyectos internos o externos a la organización relacionados	Permiten revisar lecciones aprendidas y mejores prácticas de proyectos anteriores, asegurando que la solución propuesta evite errores previos y maximice la eficacia.

Nota. Fuente: Elaboración propia.

3.5.2. Fuentes secundarias

Proporcionan un contexto más amplio a través de análisis e interpretaciones de investigaciones anteriores. Estas fuentes son cruciales para construir el marco teórico y fundamentar las decisiones de diseño en antecedentes y mejores prácticas. En la Tabla 10 se presentan las fuentes secundarias.

Tabla 10
Fuentes secundarias

Fuente	Relevancia
Repositorio de TFG del Instituto Tecnológico de Costa Rica	Ofrece ejemplos estructurales de trabajos anteriores que abordaron problemas similares, permitiendo analizar soluciones previas y aplicarlas en el diseño del presente TFG.
Sistema de bases de datos suscritas del Instituto Tecnológico de Costa Rica	Proporciona acceso a artículos científicos y libros especializados, fundamentales para respaldar teóricamente el análisis y garantizar que la propuesta de diseño cumpla con las mejores prácticas y normativas internacionales actuales.
Sitios web especializados	Proveen información actualizada sobre tendencias y tecnologías de integración de sistemas, que permiten al investigador mantenerse alineado con las últimas innovaciones y prácticas en el campo de la interoperabilidad.

Nota. Fuente: Elaboración propia.

3.6. Sujetos de investigación

Según Hernández et al. (2014), los sujetos de investigación son los individuos que proporcionan información clave para alcanzar los objetivos del estudio. Su selección debe estar alineada con el tipo de investigación y los propósitos del proyecto. En este TFG, se ha optado por una muestra no probabilística basada en criterios, dado que los sujetos han sido seleccionados en función de su conocimiento y experiencia directa con los sistemas de Hortiluma. Este enfoque asegura que los sujetos seleccionados aporten información relevante para identificar problemas de interoperabilidad y proponer soluciones factibles.

La participación de estos sujetos es fundamental para asegurar que las soluciones propuestas estén alineadas con las necesidades estratégicas y operativas de Hortiluma. Además, su experiencia contribuye a asegurar que las propuestas de mejora y las decisiones relacionadas con la interoperabilidad de los sistemas sean realistas y estén fundamentadas en el contexto operativo de la organización. La Tabla 11 detalla a los sujetos clave involucrados en el proyecto, especificando su rol, años de experiencia y su relevancia para el estudio.

Tabla 11
Sujetos de investigación

Rol	Administración del negocio	Años en el rol	10
Caracterización / Relevancia	Supervisa todas las operaciones comerciales, incluyendo ventas, inventarios y estrategias de <i>marketing</i> . Su visión estratégica global asegura que las mejoras en la interoperabilidad estén alineadas con los objetivos comerciales de Hortiluma.		
Rol	Gestión de ventas minoristas	Años en el rol	10
Caracterización / Relevancia	Administra las transacciones diarias con los clientes y gestiona el inventario de productos vendidos al detalle. Su experiencia diaria en el uso de los sistemas actuales es fundamental para identificar oportunidades de mejora en la eficiencia operativa.		
Rol	Gestión de ventas mayoristas	Años en el rol	10
Caracterización / Relevancia	Responsable de la facturación y las operaciones de ventas mayoristas, colaborando estrechamente con proveedores. Su conocimiento en la gestión de ventas al por mayor es crucial para asegurar que las mejoras faciliten la integración efectiva de las operaciones.		

Nota. Fuente: Elaboración propia.

3.7. Variables o categorías de la investigación

Las variables de investigación son elementos esenciales que permiten abordar las preguntas clave del estudio y asegurar el éxito de las soluciones propuestas. Su identificación adecuada facilita el diseño de instrumentos que capturen datos relevantes y posibiliten un análisis detallado de los resultados. En este TFG, se trabaja con variables independientes y dependientes, alineadas con los objetivos específicos del proyecto. Estas variables se miden a través de indicadores cualitativos y cuantitativos que permiten evaluar el impacto de las soluciones propuestas.

- Variable independiente: es el factor que se manipula o varía en el estudio para observar su impacto en otras variables.
- Variable dependiente: esta variable cambia en respuesta a las modificaciones de la variable independiente.

La Tabla 12 describe las variables clave, con su conceptualización, indicadores y justificación, así como los instrumentos empleados para la recolección de datos.

Tabla 12
Variables de investigación

Objetivo específico	Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.		
Variable independiente	3.7.1. Datos redundantes y olores de <i>software</i>		
Indicadores	Conceptualización	Justificación	Instrumentos
Funcionalidades duplicadas, datos inconsistentes, olores de software detectados	La duplicación de datos y funcionalidades, así como la presencia de otros “olores de software” genera ineficiencias que afectan la interoperabilidad.	Identificar y reducir las redundancias y los olores de software es crucial para mejorar la eficiencia operativa y la comunicación entre los sistemas.	Revisión documental, entrevistas, observación directa.
Objetivo específico	Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.		
Variable dependiente	3.7.2. Brechas de interoperabilidad		
Indicadores	Conceptualización	Justificación	Instrumentos
Tareas duplicadas, inconsistencias de datos, problemas de comunicación entre sistemas	Las brechas de interoperabilidad reflejan las dificultades en la integración fluida entre las funcionalidades, identificadas mediante la falta de coordinación y datos duplicados.	Reducir estas brechas permitirá mejorar la coherencia operativa y la integración de funcionalidades.	Revisión documental, entrevistas, observación directa.

Objetivo específico	Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.		
Variable independiente	3.7.3. Herramientas tecnológicas de integración y eliminación de deuda técnica		
Indicadores	Conceptualización	Justificación	Instrumentos
Grado de integración logrado, redundancias eliminadas, deuda técnica atendida	Las herramientas tecnológicas adecuadas permiten integrar datos, eliminar duplicaciones, resolver errores de software y reducir la deuda técnica acumulada.	Mejorar las herramientas tecnológicas incrementa la eficiencia operativa y reduce los costos asociados a la deuda técnica.	Revisión documental, entrevistas, observación directa, grupo focal, lluvia de ideas.
Objetivo específico	Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.		
Variable dependiente	3.7.4. Simplificación de procesos y eliminación de redundancias		
Indicadores	Conceptualización	Justificación	Instrumentos
Tareas redundantes eliminadas, datos integrados, deuda técnica disminuida	Simplificar los procesos y eliminar redundancias reduce la carga operativa, mejorando la eficiencia y sostenibilidad del sistema.	La eliminación de redundancias y la mejora de procesos minimizan la deuda técnica y mejoran la sostenibilidad del sistema a largo plazo.	Revisión documental, entrevistas, observación directa, grupo focal, lluvia de ideas.
Objetivo específico	Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.		
Variable independiente	3.7.5. Diseño preliminar de la arquitectura de integración		
Indicadores	Conceptualización	Justificación	Instrumentos
Eliminación de duplicaciones, deuda técnica atendida, tecnologías integradas	El diseño preliminar debe eliminar duplicaciones, reducir la deuda técnica, asegurando la interoperabilidad entre sistemas.	Un diseño arquitectónico es fundamental para reducir las redundancias y preparar el sistema para una mayor escalabilidad y eficiencia futura.	Revisión documental, entrevistas, observación directa, grupo focal, lluvia de ideas, pruebas de historias de usuario.

Objetivo específico	Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.		
Variable dependiente	3.7.6. Reducción de duplicaciones y mejora de interoperabilidad		
Indicadores	Conceptualización	Justificación	Instrumentos
Tareas duplicadas eliminadas, consistencia de datos entre sistemas, reducción de “olores de software”.	La reducción de duplicaciones y la mejora en la interoperabilidad asegura un sistema listo para futuras expansiones.	Garantizar la interoperabilidad mejora el uso de recursos y facilita la integración de nuevas funcionalidades.	Revisión documental, entrevistas, observación directa, historias de usuario.
Objetivo específico	Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.		
Variable independiente	3.7.7. Historias de usuario y validación de la arquitectura		
Indicadores	Conceptualización	Justificación	Instrumentos
Número de funcionalidades interoperables, cantidad de historias de usuario validados.	Los historias de usuario verifican que la arquitectura propuesta cubre todas las funcionalidades requeridas y cumple con los requisitos de interoperabilidad.	Validar los historias de usuario garantiza que el sistema cumple con las expectativas del cliente y es viable en el entorno operativo.	Revisión documental, pruebas de historias de usuario.
Objetivo específico	Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.		
Variable dependiente	3.7.8. Cumplimiento de requisitos de interoperabilidad		
Indicadores	Conceptualización	Justificación	Instrumentos
Funcionalidades interoperables, datos consolidados	La interoperabilidad efectiva entre los sistemas es un indicador clave del éxito del diseño arquitectónico, garantizando una integración completa.	Asegurar la interoperabilidad mejora la eficiencia del sistema y facilita su escalabilidad y adaptación futura.	Revisión documental, pruebas de historias de usuario, matriz de cobertura.

Nota. Fuente: Elaboración propia.

3.8. Técnicas e instrumentos de recolección de datos

La recolección de datos, según Hernández-Sampieri y Mendoza (2018), consiste en aplicar instrumentos de medición para obtener información relevante sobre las variables del estudio. Estos datos son esenciales para el análisis y deben alinearse con los objetivos, facilitando la integración de funcionalidades y mejorando la interoperabilidad. En la Tabla 13 se presentan los instrumentos de este estudio.

Tabla 13
Instrumentos de recolección de datos

Instrumento	Descripción	Relevancia	Apéndice
Revisión documental	Técnica que recopila información escrita relevante, como informes técnicos y manuales operativos, para identificar variables y analizar documentos (Machuca et al., 2022).	Identifica áreas de mejora en los sistemas de Hortiluma y mejora la integración tecnológica.	Apéndice E. Plantilla de revisión documental
Entrevistas	Método cualitativo para obtener información a través del diálogo directo con los participantes (Hernández-Sampieri et al., 2014).	Garantizan la uniformidad en la recolección de datos y permiten explorar problemas de interoperabilidad.	Apéndice D. Plantilla de entrevista
Observación directa	Técnica cualitativa que observa interacciones y comportamientos en su entorno natural (Hernández-Sampieri et al., 2014).	Identifica problemas de usabilidad en los sistemas de Hortiluma.	Apéndice F. Plantilla de observación directa
Grupo focal	Técnica que reúne a un grupo pequeño para discutir temas en profundidad y recoger diferentes perspectivas sobre el estudio (Hernández-Sampieri et al., 2014).	Explora necesidades relacionadas con la integración de sistemas de información.	Apéndice G. Plantilla de grupo focal
Lluvia de ideas	Método para la generación de ideas mediante aportes de los participantes, fomentando la creatividad colectiva (Atlassian et al., 2024).	Genera soluciones innovadoras para mejorar la integración de los sistemas.	Apéndice H. Plantilla de lluvia de ideas
Pruebas de historias de usuario	Herramienta para validar el funcionamiento de los sistemas en escenarios reales, asegurando el cumplimiento de los requisitos (Carvajal Segura, 2022).	Garantiza que la integración cumpla con los requisitos planteados.	Apéndice I. Plantilla de historias de usuario
Matriz de cobertura de requerimientos	Herramienta que asegura el cumplimiento de todos los requerimientos definidos en el diseño final del sistema (Chavarría Sánchez, 2023).	Verifica que los aspectos técnicos y operativos estén alineados con las especificaciones del proyecto.	Apéndice K. Plantilla de matriz de cobertura de requerimientos

Nota. Fuente: Elaboración propia.

3.9. Matriz de cobertura de las variables vs. el diseño de los instrumentos

Para asegurar que todas las variables relevantes del proyecto de Hortiluma están adecuadamente cubiertas por los instrumentos de recolección de datos, se presenta la siguiente matriz. Esta herramienta permite verificar la alineación entre las variables identificadas y los instrumentos diseñados, garantizando así una recopilación de datos exhaustiva y orientada hacia el cumplimiento de los objetivos específicos del proyecto. La Tabla 14 presenta la matriz de cobertura de las variables vs. el diseño de los instrumentos.

Tabla 14
Matriz de cobertura de las variables vs. el diseño de los instrumentos

Variable	Revisión documental	Entre vistas	Observación directa	Grupo focal	Lluvia de ideas	Pruebas de historias de usuario	Matriz de cobertura de requerimientos
Datos redundantes y olores de <i>software</i>	X	X	X				
Brechas de interoperabilidad	X	X	X				
Herramientas tecnológicas de integración y eliminación de deuda técnica	X	X	X	X	X		
Simplificación de procesos y eliminación de redundancias	X	X	X	X	X		
Diseño preliminar de la arquitectura de integración	X	X	X	X	X	X	
Reducción de duplicaciones y mejora de interoperabilidad	X	X	X			X	
Historias de usuario y validación de la arquitectura	X					X	
Cumplimiento de requisitos de interoperabilidad	X					X	X

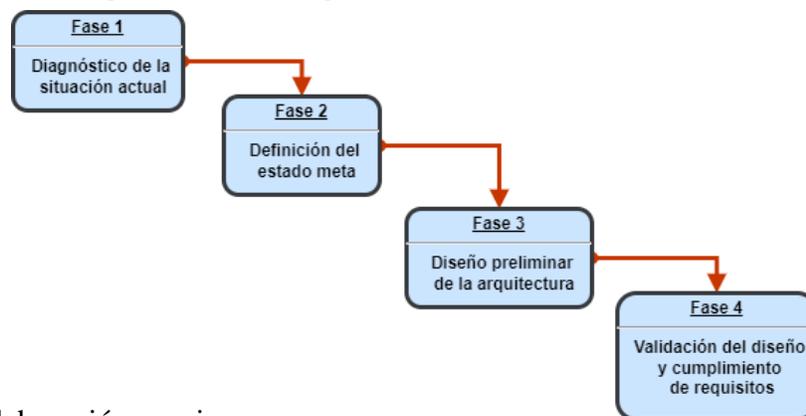
Nota. Fuente: Elaboración propia.

3.10. Procedimiento metodológico de la investigación

El procedimiento metodológico de este TFG sigue las recomendaciones de Hernández, Fernández y Baptista (2014), garantizando la obtención de resultados cuantitativos y cualitativos confiables. El proceso está segmentado en cuatro fases principales, alineadas con los objetivos específicos y las variables de investigación. Cada fase cubre etapas clave del proceso metodológico, desde la definición del problema hasta la validación de las soluciones propuestas, asegurando la rigurosidad en la recolección y análisis de datos. La Figura 4 muestra cómo se relacionan las diferentes fases del procedimiento metodológico, asegurando la coherencia en cada etapa.

- Fase 1: Diagnóstico de la situación actual
- Fase 2: Definición del estado meta
- Fase 3: Diseño preliminar de la arquitectura
- Fase 4: Validación del diseño y cumplimiento de requisitos

Figura 4
Procedimiento metodológico de la investigación



Nota. Fuente: Elaboración propia.

Tabla 15
Procedimiento metodológico de la investigación detallado

Fase	Propósito	Metodología	Apéndice
Fase 1: Diagnóstico de la situación actual	Identificar las brechas de interoperabilidad en los sistemas actuales de Hortiluma, así como problemas operativos relacionados con la duplicación de tareas y redundancia de datos. Incluye la inmersión inicial, el planteamiento del problema y la definición de variables clave del estudio.	- Investigación cualitativa: análisis de documentos técnicos y entrevistas estructuradas con personal clave. - Observación directa: evaluación en tiempo real de procesos operativos para identificar duplicaciones y puntos críticos.	Apéndice E. Plantilla de revisión documental. Apéndice D. Plantilla de entrevista. Apéndice F. Plantilla de observación directa.

Fase	Propósito	Metodología	Apéndice
Fase 2: Definición del estado meta	Establecer un estado meta para los sistemas integrados, enfocado en simplificar procesos y eliminar redundancias. Define el enfoque de la investigación y las metas específicas del diseño.	<ul style="list-style-type: none"> - Metodología participativa: reuniones con grupos focales y lluvias de ideas para definir objetivos compartidos. - Evaluación tecnológica: benchmarking de herramientas disponibles para determinar su viabilidad técnica. - Entrevistas y observación directa: recopilación de información complementaria. 	<p>Apéndice E. Plantilla de revisión documental.</p> <p>Apéndice D. Plantilla de entrevista.</p> <p>Apéndice F. Plantilla de observación directa.</p> <p>Apéndice G. Plantilla de grupo focal.</p> <p>Apéndice H. Plantilla de lluvia de ideas.</p>
Fase 3: Diseño preliminar de la arquitectura	Desarrollar un diseño preliminar que permita la integración de funcionalidades, elimine duplicaciones y permita expansiones futuras. Ajustes iterativos basados en retroalimentación de usuarios.	<ul style="list-style-type: none"> - Diseño iterativo: ajustes continuos con retroalimentación de usuarios clave. - Pruebas de concepto: validación del diseño mediante historias de usuario. - Evaluación técnica: identificación de riesgos técnicos relacionados con escalabilidad y compatibilidad. 	<p>Apéndice E. Plantilla de revisión documental.</p> <p>Apéndice D. Plantilla de entrevista.</p> <p>Apéndice F. Plantilla de observación directa.</p> <p>Apéndice G. Plantilla de grupo focal.</p> <p>Apéndice H. Plantilla de lluvia de ideas.</p> <p>Apéndice I. Plantilla de historias de usuario.</p>
Fase 4: Validación del diseño y cumplimiento de requisitos	Garantizar que el diseño cumple con los requerimientos técnicos y funcionales, mejora la interoperabilidad y está alineado con los objetivos del proyecto.	<ul style="list-style-type: none"> - Validación estructurada: uso de matrices de requerimientos y pruebas de historias de usuario para asegurar cobertura. - Control de calidad: comparación detallada del diseño con los objetivos planteados. 	<p>Apéndice E. Plantilla de revisión documental.</p> <p>Apéndice I. Plantilla de historias de usuario.</p> <p>Apéndice K. Plantilla de matriz de cobertura de requerimientos</p>

Nota. Fuente: Elaboración propia.

3.11. Operacionalización de las variables o categorías

Para clarificar los fundamentos metodológicos de esta investigación, se presenta la Tabla 16

Tabla 16, que muestra cómo se operacionalizan las variables clave en relación con las fases del proyecto y los instrumentos de investigación. Esta tabla facilita la sistematización de la recolección y el análisis de datos, garantizando que cada variable de investigación sea adecuadamente cubierta con los instrumentos y sujetos pertinentes. También establece una relación directa entre las fases del procedimiento metodológico y los objetivos específicos del proyecto.

Tabla 16
Operacionalización de las variables

Fase 1: Diagnóstico de la situación actual		Instrumentos
Objetivo específico	Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.	Apéndice E. Plantilla de revisión documental Apéndice D. Plantilla de entrevista Apéndice F. Plantilla de observación directa
Variables de investigación		Sujetos de investigación
Datos redundantes y olores de <i>software</i> , brechas de interoperabilidad		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas
Fase 2: Definición del estado meta		Instrumentos
Objetivo específico	Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.	Apéndice E. Plantilla de revisión documental Apéndice D. Plantilla de entrevista Apéndice F. Plantilla de observación directa Apéndice G. Plantilla de grupo focal Apéndice H. Plantilla de lluvia de ideas
Variables de investigación		Sujetos de investigación
Herramientas tecnológicas de integración y eliminación de deuda técnica, simplificación de procesos y eliminación de redundancias		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas
Fase 3: Diseño preliminar de la arquitectura		Instrumentos
Objetivo específico	Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.	Apéndice E. Plantilla de revisión documental Apéndice D. Plantilla de entrevista Apéndice F. Plantilla de observación directa Apéndice G. Plantilla de grupo focal Apéndice H. Plantilla de lluvia de ideas Apéndice I. Plantilla de historias de usuario
Variables de investigación		Sujetos de investigación
Diseño preliminar de la arquitectura de integración, reducción de duplicaciones y mejora de interoperabilidad		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas
Fase 4: Validación del diseño y cumplimiento de requisitos		Instrumentos
Objetivo específico	Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.	Apéndice E. Plantilla de revisión documental Apéndice I. Plantilla de historias de usuario Apéndice K. Plantilla de matriz de cobertura de requerimientos
Variables de investigación		Sujetos de investigación
Historias de usuario y validación de la arquitectura, cumplimiento de requisitos de interoperabilidad		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas

Nota. Fuente: Elaboración propia.

3.12. Matriz de trazabilidad

Tabla 17
Matriz de trazabilidad

Objetivo específico	Marco conceptual	Marco metodológico	Análisis de resultados-Propuesta	Conclusiones-Recomendaciones	Apéndices
Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.	Especificación del <i>software</i> Diseño de <i>software</i> Olores de <i>software</i>	Variables o categorías de la investigación Técnicas e instrumentos de recolección de datos Procedimiento metodológico de la investigación	Fase 1: Diagnóstico de la situación actual	Conclusiones Objetivo específico 1 Recomendaciones Objetivo específico 1	Apéndice D. Plantilla de entrevista Apéndice E. Plantilla de revisión documental Apéndice F. Plantilla de observación directa
Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.	Diseño de <i>software</i> Principios de diseño SOLID Patrones de diseño Axiomas fundamentales de diseño Deuda técnica Evolución del <i>software</i> Integración entre sistemas de <i>software</i>	Sujetos de investigación Matriz de cobertura de las variables vs. el diseño de los instrumentos	Fase 2: Definición del estado meta	Conclusiones Objetivo específico 2 Recomendaciones Objetivo específico 2	Apéndice D. Plantilla de entrevista Apéndice E. Plantilla de revisión documental Apéndice F. Plantilla de observación directa Apéndice G. Plantilla de grupo focal Apéndice H. Plantilla de lluvia de ideas Apéndice J. Plantilla de requerimientos

Objetivo específico	Marco conceptual	Marco metodológico	Análisis de resultados-Propuesta	Conclusiones-Recomendaciones	Apéndices
Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.	Diseño de <i>software</i> Deuda técnica Pensamiento convergente y divergente en el diseño UML SaaS	Diseño de la investigación Operacionalización de las variables o categorías	Fase 3: Diseño preliminar de la arquitectura	Conclusiones Objetivo específico 3 Recomendaciones Objetivo específico 3	Apéndice D. Plantilla de entrevista Apéndice E. Plantilla de revisión documental Apéndice F. Plantilla de observación directa Apéndice G. Plantilla de grupo focal Apéndice H. Plantilla de lluvia de ideas Apéndice I. Plantilla de historias de usuario
Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.	Notación para graficar el diseño Automatización y estándares de datos Verificación y validación del <i>software</i> (V&V)	Técnicas e instrumentos de recolección de datos Operacionalización de las variables o categorías	Fase 4: Validación del diseño y cumplimiento de requisitos	Conclusiones Objetivo específico 4 Recomendaciones Objetivo específico 4	Apéndice E. Plantilla de revisión documental Apéndice I. Plantilla de historias de usuario Apéndice K. Plantilla de matriz de cobertura de requerimientos

Nota. Fuente: Elaboración propia.

4. Análisis de resultados

Este capítulo se centra en los resultados obtenidos a través de la aplicación de los instrumentos metodológicos diseñados para esta investigación. Los datos se organizan de acuerdo con las fases del procedimiento metodológico de la investigación y cada hallazgo está respaldado por registros documentales, entrevistas, observaciones directas y análisis de los datos obtenidos de los sistemas de Hortiluma. Las plantillas aplicadas permiten establecer una clara vinculación entre los resultados y las fuentes de información, asegurando una base sólida para la propuesta de solución en el siguiente capítulo.

4.1. Fase 1: Diagnóstico de la situación actual

En la Fase 1 se identificaron problemas clave relacionados con la interoperabilidad, duplicación de tareas y falta de automatización en los sistemas de Hortiluma. Mediante la revisión documental, entrevistas y observaciones directas, se evidenció la redundancia de datos, la falta de integración y la presencia de “olores de *software*”, los cuales impactan negativamente en la operatividad, la precisión de los reportes y la toma de decisiones. Estos hallazgos establecen una base para desarrollar recomendaciones que mejoren la eficiencia, reduzcan errores y mejoren los flujos de trabajo.

Tabla 18
Operacionalización de las variables – Fase 1

Fase 1: Diagnóstico de la situación actual		Instrumentos
Objetivo específico	Realizar un diagnóstico de los sistemas de Hortiluma para la identificación de las brechas de interoperabilidad entre funcionalidades y datos existentes, mediante la evaluación de tareas y datos duplicados.	Apéndice E. Plantilla de revisión documental Apéndice D. Plantilla de entrevista Apéndice F. Plantilla de observación directa
VARIABLES DE INVESTIGACIÓN		SUJETOS DE INVESTIGACIÓN
Datos redundantes y olores de <i>software</i> , brechas de interoperabilidad		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas

Nota. Fuente: Elaboración propia.

4.1.1. Revisión documental

La revisión documental fue fundamental para identificar problemas técnicos en los sistemas de Hortiluma. El análisis de manuales técnicos, reportes de inventarios y documentos internos reveló problemas como la duplicación de datos y la falta de integración, afectando la operatividad. Estos problemas, relacionados con olores de *software* y brechas de interoperabilidad, impactan negativamente en la eficiencia y precisión de los reportes.

Tabla 19
Análisis de fase 1 – Revisiones documentales

Hallazgo	Apéndice	Problema	Impacto	Recomendación
Duplicación de datos entre Kyte y FACEL	Apéndice E.1. Plantilla de revisión documental - RD-001 Apéndice E.2. Plantilla de revisión documental - RD-002	Los empleados deben actualizar manualmente los inventarios en ambos sistemas, lo que genera retrabajo y errores en los reportes de inventarios.	Esto provoca inconsistencias en los datos y afecta la coherencia de los reportes.	Implementar una sincronización automática para evitar la duplicación de tareas y mejorar la precisión de los datos.
Brechas de interoperabilidad entre Kyte, FACEL y POS	Apéndice E.2. Plantilla de revisión documental - RD-002 Apéndice E.3. Plantilla de revisión documental - RD-003	La falta de sincronización entre estos tres sistemas requiere que los empleados ingresen los datos manualmente en cada uno, lo que genera duplicación de tareas y errores en los reportes de ventas e inventarios.	Esto afecta la toma de decisiones y retrasos en la gestión operativa.	Integrar las funcionalidades para permitir una actualización automática en tiempo real y evitar errores manuales.
“Olores de <i>software</i> ” asociados a procesos manuales	Apéndice E.4. Plantilla de revisión documental - RD-004 Apéndice E.5. Plantilla de revisión documental - RD-005	Las tareas manuales entre sistemas provocan duplicación de funciones y son indicativas de ineficiencia operativa.	La falta de automatización genera retrabajo y errores en la consolidación de los datos.	Proponer un diseño arquitectónico que mejore los flujos de trabajo y reduzca las redundancias.

Nota. Fuente: Elaboración propia.

4.1.2. Entrevistas

Las entrevistas ofrecieron información valiosa sobre los problemas operativos en Hortiluma, destacando la duplicación de tareas, las brechas de interoperabilidad y la falta de automatización. Estos problemas afectan la eficiencia y precisión operativa, complicando la toma de decisiones basada en datos fiables.

Tabla 20
Análisis de fase 1 – Entrevistas

Hallazgo	Apéndice	Problema	Impacto	Recomendación
Duplicación de tareas en inventarios y facturación	Apéndice D.3. Plantilla de entrevista - ENT-003	Los empleados ingresan los mismos datos manualmente en tres sistemas (Kyte, FACEL y POS), lo que genera inconsistencias y retrabajo.	Afecta la precisión de inventarios y la eficiencia operativa del negocio.	Centralizar la entrada de datos y automatizar la actualización de inventarios y facturación para mejorar la coherencia de los reportes.
Brechas de interoperabilidad entre sistemas	Apéndice D.2. Plantilla de entrevista - ENT-002	La falta de integración entre Kyte, FACEL y POS genera errores en los reportes de ventas y retrasos en la actualización de inventarios.	Dificulta la gestión de inventarios y la planificación de stock, afectando la satisfacción del cliente.	Integrar las funcionalidades para garantizar la sincronización de los datos en tiempo real.
Falta de automatización en inventarios mayoristas	Apéndice D.1. Plantilla de entrevista - ENT-001	La falta de automatización en los sistemas de inventarios mayoristas provoca discrepancias en la disponibilidad de productos y afecta la planificación de ventas.	Genera errores operativos que retrasan la planificación y aumentan costos.	Automatizar la actualización de inventarios para reflejar la información en tiempo real, mejorando la planificación de ventas.

Nota. Fuente: Elaboración propia.

4.1.3. Observación directa

La observación directa permitió identificar problemas críticos como la duplicación de tareas y la falta de sincronización automática en los sistemas Kyte, FACEL y POS, que generan retrabajo, inconsistencias en reportes de ventas e inventarios, así como retrasos en la toma de decisiones.

Tabla 21

Análisis de fase 1 – Observaciones directas

Hallazgo	Apéndice	Problema	Impacto	Recomendación
Duplicación de tareas en la gestión de inventarios	Apéndice F.1. Plantilla de observación directa - OBS-001 Apéndice F.4. Plantilla de observación directa - OBS-004	Los empleados ingresan manualmente los datos de ventas e inventarios en tres sistemas distintos, lo que genera inconsistencias en los reportes y retrasos.	Aumenta el riesgo de errores operativos y afecta la actualización oportuna de los inventarios.	Implementar una sincronización automática entre las funcionalidades para eliminar la duplicación de tareas y mejorar la coherencia de los datos.
Falta de sincronización automática	Apéndice F.2. Plantilla de observación directa - OBS-002 Apéndice F.3. Plantilla de observación directa - OBS-003	La falta de sincronización automática entre los sistemas provoca errores en la actualización de inventarios y precios, afectando la precisión de los reportes.	Genera ineficiencia operativa y errores en la toma de decisiones.	Automatizar los procesos para permitir una actualización en tiempo real de los inventarios y precios, eliminando los errores manuales.
Duplicación en la facturación	Apéndice F.3. Plantilla de observación directa - OBS-003 Apéndice F.4. Plantilla de observación directa - OBS-004	El proceso de facturación se realiza de manera manual y duplicada en Kyte, FACEL y POS, generando errores en la consolidación de datos y retrasos en la conciliación de ventas.	La falta de integración afecta la precisión de los datos contables y retrasa la conciliación de ventas.	Centralizar el proceso de facturación e integrar los sistemas para evitar duplicaciones y errores.

Nota. Fuente: Elaboración propia.

4.1.4. Análisis de hallazgos de la Fase 1

Durante la Fase 1, se detectaron varios problemas críticos en los sistemas de Hortiluma, los cuales afectan negativamente su operatividad. Estos problemas se relacionan, principalmente, con la falta de interoperabilidad entre los sistemas, la duplicación de tareas y la falta de automatización en procesos clave. En esta sección, se describen los problemas principales identificados, sus impactos en la organización y las áreas que requieren atención urgente para mejorar la eficiencia operativa y la precisión en la toma de decisiones.

1. Duplicación de tareas y procesos manuales:
 - a. Los empleados de Hortiluma deben ingresar los mismos datos en tres sistemas diferentes: Kyte, FACEL y el sistema POS legado. Este proceso genera un alto nivel de retrabajo y errores humanos.
 - b. La redundancia en la entrada de datos no solo consume tiempo, sino que también produce inconsistencias entre los sistemas, afectando la calidad de los reportes generados, lo que impide que la información sea confiable para la toma de decisiones.
2. Falta de integración e interoperabilidad:
 - a. Se evidenció una desconexión entre los sistemas Kyte, FACEL y POS. Al no estar integrados, los datos no se sincronizan automáticamente, lo que genera un flujo de trabajo fragmentado.
 - b. Esta falta de sincronización automatizada resulta en errores de inventarios, retrasos en la actualización de precios y fallos en la facturación. Esto no solo afecta los procesos operativos, sino que también retrasa la respuesta a las necesidades del negocio, especialmente en la gestión de inventarios y ventas.
3. Olores de *software* y procesos redundantes:
 - a. La dependencia de procesos manuales repetitivos es un claro indicio de problemas estructurales en el diseño del *software* actual. Los “olores de *software*” (fallos recurrentes en el diseño del sistema) indican la necesidad de modernización y reestructuración de las plataformas.
 - b. Estos procesos no mejorados aumentan los costos operativos al requerir más recursos humanos y tiempo para completar tareas que podrían automatizarse. Además, el constante retrabajo asociado a estas tareas incrementa el riesgo de errores que afectan la calidad de los datos.
4. Impacto de la falta de automatización:
 - a. La falta de automatización es evidente en la gestión de inventarios, donde las discrepancias entre lo registrado en los sistemas y el inventario real dificultan la planificación de ventas y la toma de decisiones basadas en datos precisos.
 - b. Sin una automatización efectiva, la disponibilidad de productos no se refleja de manera precisa, lo que incrementa los costos debido a un mal manejo de los inventarios y afecta la satisfacción del cliente, que no siempre encuentra productos disponibles cuando los necesita.

Con base en los problemas, se identifican los siguientes impactos relevantes:

- Errores en los reportes: la falta de coherencia entre los datos ingresados en los diferentes sistemas provoca discrepancias en los reportes de ventas, inventarios y facturación, afectando la precisión de la información con la que se toman decisiones críticas.
- Retrasos en la actualización de inventarios: los retrasos en la sincronización de inventarios causan una falta de precisión en la disponibilidad de productos, complicando la planificación de ventas y aumentando el riesgo de desabastecimiento o exceso de stock, lo que, a su vez, afecta los costos operativos.
- Discrepancias en la facturación: las inconsistencias entre los sistemas generan discrepancias en la conciliación contable, lo que conlleva a errores en los reportes financieros. Estos errores complican la gestión financiera del negocio y afectan la toma de decisiones estratégicas.

4.2. Fase 2: Definición del estado meta

En esta fase, se define el estado meta para la integración de los sistemas de ventas, inventarios y facturación de Hortiluma. Se propone diseñar un nuevo sistema desde cero para abordar los problemas actuales relacionados con la deuda técnica, la duplicación de tareas y la falta de interoperabilidad. Este análisis se basa en datos recopilados mediante una revisión documental, entrevistas, observación directa, grupos focales y lluvias de ideas, estas herramientas han sido esenciales para identificar las áreas clave de mejora y desarrollar una propuesta integral que garantice la interoperabilidad y simplificación de los procesos actuales. Pilares clave:

- Automatización de procesos: un diseño automatizado eliminará las redundancias y reducirá la intervención manual, mejorando la eficiencia y precisión.
- Eliminación de la deuda técnica: un sistema completamente nuevo permitirá superar la deuda técnica acumulada y construir una plataforma escalable a largo plazo.
- Mejora en la interoperabilidad: la creación de un sistema integrado garantizará una comunicación fluida entre los componentes, asegurando la coherencia y consistencia de los datos.

Tabla 22
Operacionalización de las variables – Fase 2

Fase 2: Definición del estado meta		Instrumentos
Objetivo específico	Definir el estado meta de las necesidades para la integración de las funcionalidades y datos actuales, medido por la simplificación de tareas y datos.	Apéndice E. Plantilla de revisión documental Apéndice D. Plantilla de entrevista Apéndice F. Plantilla de observación directa Apéndice G. Plantilla de grupo focal Apéndice H. Plantilla de lluvia de ideas
Variables de investigación		Sujetos de investigación
Herramientas tecnológicas de integración y eliminación de deuda técnica, simplificación de procesos y eliminación de redundancias		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas

Nota. Fuente: Elaboración propia.

4.2.1. Revisión documental

La revisión documental sirvió como base para definir el estado meta de la integración de sistemas en Hortiluma. Se evaluaron los sistemas Kyte, FACEL y POS legado para identificar barreras tecnológicas y limitaciones que dificultan la interoperabilidad y automatización de procesos clave, como ventas, inventarios y facturación. A continuación, se presentan los hallazgos clave obtenidos de cada revisión documental, organizados por apéndice:

Tabla 23
Operacionalización de las variables – Fase 2

Hallazgo	Apéndice	Análisis	Recomendación
Módulos en desuso y deuda técnica	Apéndice E.6. Plantilla de revisión documental - RD-006	Se identificaron módulos en desuso y problemas de interoperabilidad causados por la deuda técnica en el POS legado y FACEL.	Eliminar o reemplazar estos módulos obsoletos para simplificar el sistema. Abordar la deuda técnica del POS legado para mejorar la interoperabilidad y reducir costos.
Falta de sincronización y automatización	Apéndice E.7. Plantilla de revisión documental - RD-007	La falta de sincronización automática entre Kyte, FACEL y POS genera duplicación de tareas y errores en los reportes financieros, afectando la eficiencia operativa.	Rediseñar los procesos de sincronización y automatización para evitar duplicaciones y errores, mejorando la coherencia de los datos y la eficiencia operativa.
Actualización manual de datos	Apéndice E.8. Plantilla de revisión documental - RD-008	Las actualizaciones manuales generan errores y costos innecesarios debido a la falta de sincronización en tiempo real entre Kyte y POS.	Realizar una reingeniería de procesos antes de implementar nuevas tecnologías de automatización, garantizando que los procesos estén preparados para la integración.
Incompatibilidad de API	Apéndice E.9. Plantilla de revisión documental - RD-009	La incompatibilidad de API entre Kyte y FACEL, y la falta de una API moderna en POS, limita la integración directa entre sistemas, lo que afecta la coherencia de los datos.	Implementar <i>middleware</i> para conectar los sistemas y superar las barreras de la falta de API modernas, facilitando una integración fluida entre los componentes.
Falta de automatización en inventarios	Apéndice E.10. Plantilla de revisión documental - RD-010	La actualización manual de inventarios genera errores, afectando la precisión de las decisiones operativas y la coherencia de los datos financieros.	Automatizar el proceso de actualización de inventarios para garantizar la coherencia de los datos y mejorar la precisión de los reportes financieros y de operaciones.

Nota. Fuente: Elaboración propia.

4.2.2. Entrevistas

Las entrevistas con el administrador del negocio y los responsables de ventas revelaron que los sistemas actuales de Hortiluma presentan una gran inflexibilidad, lo que limita su capacidad para adaptarse a nuevas soluciones tecnológicas. Se destacó la necesidad de rediseñar desde cero la arquitectura de los sistemas, eliminando las restricciones impuestas por las plataformas actuales (Kyte, FACEL y POS). A continuación, se presentan los hallazgos clave y recomendaciones derivadas de las entrevistas:

Tabla 24
Análisis de fase 2 – Entrevistas

Hallazgo	Apéndice	Análisis	Recomendación
Duplicación de datos y tareas	Apéndice D.4. Plantilla de entrevista - ENT-004	La duplicación de tareas entre los sistemas de ventas e inventarios (Kyte, FACEL y POS) genera errores y aumenta el tiempo de procesamiento.	Implementar un sistema único de entrada de datos o soluciones intermediarias (<i>middleware</i> o API) para reducir la duplicación de tareas y errores humanos.
Limitaciones en la integración actual	Apéndice D.5. Plantilla de entrevista - ENT-005	Las plataformas actuales no permiten una integración fluida ni sincronización en tiempo real entre los sistemas, lo que afecta la operatividad y aumenta los costos.	Rediseñar la arquitectura del sistema utilizando <i>middleware</i> y API para conectar los sistemas y mejorar la interoperabilidad, eliminando las barreras tecnológicas.
Falta de automatización	Apéndice D.6. Plantilla de entrevista - ENT-006	La falta de automatización en la gestión de ventas e inventarios genera redundancias y errores operativos, lo que afecta la eficiencia del negocio.	Implementar tecnologías de Automatización Robótica de Procesos (RPA) para reducir tareas manuales repetitivas y mejorar la eficiencia operativa en procesos críticos.
Deuda técnica y obsolescencia	Apéndice D.5. Plantilla de entrevista - ENT-005	La deuda técnica acumulada en el sistema POS impide la integración, limita su escalabilidad y genera costos elevados de mantenimiento.	Abordar la deuda técnica mediante un rediseño completo, eliminando los sistemas obsoletos y adoptando tecnologías modernas para garantizar la escalabilidad.
Falta de sincronización de datos	Apéndice D.6. Plantilla de entrevista - ENT-006	La sincronización de datos entre Kyte, FACEL y POS es manual, lo que incrementa los errores y ralentiza los procesos de actualización de inventarios y ventas.	Implementar soluciones en la nube o <i>middleware</i> que permitan la sincronización automática en tiempo real entre los sistemas de ventas e inventarios, reduciendo los errores.

Nota. Fuente: Elaboración propia.

4.2.3. Observación directa

La observación directa reveló problemas significativos relacionados con la rigidez de los sistemas actuales y la duplicación de tareas en los procesos manuales de inventarios, ventas y facturación. Además, se constató que la falta de sincronización automática entre los sistemas Kyte, FACEL y POS ralentiza los procesos, provoca errores y genera ineficiencias que impactan tanto en la operatividad interna como en la calidad de los datos. A continuación, se presentan los hallazgos y recomendaciones clave para abordar estos problemas.

Tabla 25
Análisis de fase 2 – Observaciones directas

Hallazgo	Apéndice	Análisis	Recomendación
Falta de sincronización de inventarios en tiempo real	Apéndice F.5. Plantilla de observación directa - OBS-005	Los sistemas Kyte y FACEL no reflejan los niveles de stock en tiempo real, complicando la reposición de productos y afectando la precisión en la toma de decisiones.	Implementar un sistema único automatizado de actualización de inventarios que permita la sincronización en tiempo real, eliminando la verificación manual y mejorando la precisión.
Discrepancias en los reportes de inventarios	Apéndice F.6. Plantilla de observación directa - OBS-00	Un 15% de las actualizaciones de inventarios presenta discrepancias, afectando la planificación de compras y aumentando el riesgo de desabastecimiento.	Automatizar la gestión de inventarios para garantizar la coherencia de los datos y mejorar la planificación de compras, reduciendo el riesgo de desabastecimiento.
Duplicación de tareas manuales	Apéndice F.7. Plantilla de observación directa - OBS-007	Los empleados ingresan manualmente los datos de ventas y facturación en los sistemas Kyte, FACEL y POS de manera independiente, lo que genera inconsistencias y errores.	Implementar un sistema único de entrada de datos automatizado que sincronice los tres sistemas, eliminando la duplicación de tareas manuales y mejorando la precisión de los reportes.
Impacto en el tiempo operativo y costos	Apéndice F.6. Plantilla de observación directa - OBS-006	El 15% del tiempo operativo se invierte en corregir errores manuales y verificar datos entre los sistemas, afectando la eficiencia del personal.	Incorporar soluciones de automatización, como <i>middleware</i> o API, que permitan la sincronización en tiempo real, reduciendo los errores y mejorando la eficiencia operativa.
Riesgo de errores en facturación	Apéndice F.7. Plantilla de observación directa - OBS-007	La falta de sincronización automática entre Kyte, FACEL y POS aumenta la probabilidad de errores en la facturación y la generación de reportes financieros inexactos.	Priorizar el uso de soluciones en la nube o <i>middleware</i> para garantizar la interoperabilidad entre los sistemas, mejorando la precisión en la facturación y los reportes financieros.

Nota. Fuente: Elaboración propia.

4.2.4. Grupos focales

En las sesiones de grupos focales, los responsables de ventas y operaciones destacaron la rigidez de los sistemas actuales y concluyeron que la única solución viable es un rediseño completo del *software*. Este enfoque permitiría eliminar las limitaciones técnicas, asegurar la interoperabilidad total entre los sistemas de ventas, inventarios y facturación, así como mejorar la eficiencia operativa mediante la automatización. Así mismo, los participantes subrayaron la necesidad de utilizar tecnologías avanzadas que ofrezcan escalabilidad y flexibilidad para adaptarse a las necesidades futuras de Hortiluma.

Tabla 26
Análisis de fase 2 – Grupos focales

Hallazgo	Apéndice	Análisis	Recomendación
Rediseño del <i>software</i>	Apéndice G.1. Plantilla de grupo focal - GF-001	Un rediseño completo eliminaría las limitaciones técnicas actuales, garantizando la interoperabilidad total entre Kyte, FACEL y POS, y permitiendo la automatización crítica.	Implementar un nuevo sistema desde cero que facilite la interoperabilidad entre los sistemas y la automatización de procesos clave, como la gestión de inventarios y la facturación.
Automatización y simplificación de procesos	Apéndice G.2. Plantilla de grupo focal - GF-002 Apéndice G.2. Plantilla de grupo focal - GF-002	La automatización de procesos eliminaría las tareas redundantes, mejoraría la precisión de los datos y aumentaría la eficiencia interna, impactando positivamente la experiencia del cliente.	Adoptar herramientas como API de integración y plataformas en la nube para automatizar procesos y reducir la duplicación de tareas, mejorando la coherencia y la eficiencia operativa.
Necesidad de un sistema centralizado	Apéndice G.3. Plantilla de grupo focal - GF-003 Apéndice G.3. Plantilla de grupo focal - GF-003	Un sistema único de entrada de datos centralizado eliminaría la duplicación de tareas y sincronizaría automáticamente los datos entre ventas, inventarios y facturación.	Desarrollar un sistema centralizado que permita la sincronización automática de los datos en tiempo real, mejorando la precisión operativa y reduciendo el tiempo de procesamiento.

Nota. Fuente: Elaboración propia.

4.2.5. Lluvia de ideas

Las sesiones de lluvia de ideas reunieron a los principales actores de ventas minoristas, mayoristas, administración y operaciones, con el objetivo de proponer soluciones para las redundancias e ineficiencias detectadas en la Fase 1. Se priorizó la necesidad de mejorar la interoperabilidad entre los sistemas de ventas, inventarios y facturación, enfocándose en la automatización y la integración tecnológica para mejorar los procesos y reducir la duplicación de tareas.

Tabla 27
Análisis de fase 2 – Lluvia de ideas

Hallazgo	Apéndice	Análisis	Recomendación
Soluciones en la nube y API	Apéndice H.1. Plantilla de lluvia de ideas - BI-001	Se identificó que las soluciones en la nube (SaaS) y las API permiten una integración flexible y escalable entre los sistemas Kyte, FACEL y POS.	Adoptar soluciones en la nube y API para asegurar la integración en tiempo real entre sistemas, eliminando la duplicación de datos
Automatización con RPA	Apéndice H.1. Plantilla de lluvia de ideas - BI-001 Apéndice H.2. Plantilla de lluvia de ideas - BI-002	La automatización mediante RPA es clave para reducir los errores humanos y mejorar la precisión de la información.	Implementar RPA para automatizar procesos repetitivos como la actualización de inventarios y facturación, mejorando la eficiencia operativa.
Sistema centralizado y migración de legados	Apéndice H.2. Plantilla de lluvia de ideas - BI-002 Apéndice H.3. Plantilla de lluvia de ideas - BI-003	Un sistema centralizado de entrada de datos permitirá una mayor coherencia operativa y facilitará la migración gradual de sistemas legados.	Desarrollar un sistema único de entrada de datos y planificar la migración gradual de los sistemas legados (FACEL y POS) a plataformas modernas, asegurando la coherencia de los datos.
Middleware y soluciones híbridas	Apéndice H.3. Plantilla de lluvia de ideas - BI-003	La implementación de <i>middleware</i> facilitará la interoperabilidad sin necesidad de reemplazar completamente los sistemas legados.	Adoptar <i>middleware</i> para integrar los sistemas legados con las nuevas plataformas, minimizando las interrupciones operativas y facilitando una transición fluida.

Nota. Fuente: Elaboración propia.

4.2.6. Análisis de los hallazgos de la Fase 2

La Fase 2 se centró en definir el estado meta para la integración de los sistemas de Hortiluma, con el objetivo de desarrollar un diseño preliminar de una arquitectura interoperable que resuelva los problemas identificados en la Fase 1: deuda técnica, duplicación de tareas y falta de interoperabilidad. A continuación, se presenta el análisis de los hallazgos que fundamentan esta propuesta de diseño:

1. Integración tecnológica y eliminación de deuda técnica: los sistemas actuales (Kyte, FACEL y POS) están limitados por la falta de API modernas, lo que impide la integración y sincronización en tiempo real. Esta deficiencia genera duplicación de tareas, inconsistencias en los datos y errores en los reportes. Para un diseño , se requiere una arquitectura modular que favorezca la interoperabilidad mediante *middleware* y API estructuradas. Esto permitirá conectar los sistemas actuales con plataformas futuras de manera , eliminando problemas de compatibilidad y garantizando una comunicación fluida entre componentes. Además, la deuda técnica acumulada en el sistema POS afecta tanto su escalabilidad como su capacidad de automatización. Un rediseño completo del POS, basado en tecnologías actuales, asegurará que el sistema crezca conforme a las necesidades del negocio, garantizando flexibilidad y fácil integración con otros sistemas.
2. Simplificación de procesos y eliminación de redundancias: los procesos manuales en Kyte, FACEL y POS generan redundancias y retrabajo, afectando la coherencia de los datos y prolongando los tiempos operativos. El nuevo diseño debe centrarse en un sistema centralizado de entrada de datos que sincronice en tiempo real las funciones clave, como ventas, inventarios y facturación. Esta consolidación reducirá la duplicación de tareas y minimizará los errores manuales. Además, la automatización es esencial para procesos críticos como la actualización de inventarios y facturación. Además, incorporar tecnologías como RPA eliminará la intervención manual en tareas repetitivas, mejorando la precisión y mejorando el uso de los recursos.
3. Mejora de la sincronización y consistencia de datos: la actualización manual de datos entre los sistemas actuales incrementa el margen de error y ralentiza las operaciones, lo que impide a Hortiluma reaccionar rápidamente ante cambios. La nueva arquitectura debe priorizar la sincronización automática y en tiempo real entre todos los componentes del sistema, utilizando tecnologías de nube o *middleware*. Esto garantizará que los datos sean coherentes y precisos en todas las fases del proceso, permitiendo tomar decisiones informadas y reduciendo las inconsistencias que afectan tanto a la operación interna como a la experiencia del cliente.
4. Reducción de costos operativos y mejora de la eficiencia: actualmente, gran parte del tiempo operativo se destina a corregir errores manuales y verificar la coherencia de los datos entre sistemas, lo que incrementa los costos operativos. Un diseño interoperable, centrado en la automatización y la integración tecnológica, eliminará muchas de estas ineficiencias. Un sistema bien integrado permitirá que el personal se concentre en tareas estratégicas, reduciendo los costos operativos a largo plazo y mejorando la productividad de la empresa. Además, la reducción de errores, gracias a la automatización y la sincronización en tiempo real, resultará en una operación más competitiva.

4.2.7. Necesidades del sistema para el diseño de *software* de alto nivel

Como se identificó en el planteamiento del problema, Hortiluma enfrenta desafíos debido a la falta de un sistema integral que centralice la gestión de datos y la presentación de información de forma comprensible para los usuarios. Esto incluye la necesidad de contextualizar los datos sobre ventas, inventarios y reportes, permitiendo que los usuarios obtengan información útil para la toma de decisiones. Por lo tanto, la solución propuesta a estas limitaciones es la implementación de una solución interoperable, un sistema que busca centralizar y automatizar la gestión de datos, asegurando una experiencia de usuario fluida, precisa y adaptada a las necesidades de cada perfil de usuario.

El sistema debe ser capaz de manejar grandes volúmenes de datos provenientes de diferentes módulos de operación, integrándolos para ofrecer una vista completa y actualizada de la operativa de Hortiluma. Esto incluye la gestión de productos, inventarios, ventas, generación de facturas, reportes de ventas, así como la integración con plataformas externas para mejorar la interoperabilidad. Las necesidades específicas identificadas para el sistema son las siguientes:

- Centralización de la gestión de productos e inventarios: es crucial tener un control centralizado de los productos disponibles, permitiendo la actualización y consulta de inventarios en tiempo real para garantizar la disponibilidad de stock y la adecuada gestión de recursos.
- Automatización de procesos de ventas y facturación: la automatización en la generación de facturas electrónicas es esencial para agilizar el proceso de ventas, reducir el tiempo de respuesta y minimizar errores humanos en el registro de transacciones.
- Generación de reportes y análisis de datos: el sistema debe facilitar la creación de reportes detallados sobre ventas e inventarios, proporcionando a los usuarios información clave para la toma de decisiones estratégicas y operativas.
- Interoperabilidad con sistemas externos: la necesidad de sincronizar información con plataformas externas, como sistemas de *e-commerce* y contabilidad, permite que el sistema sea flexible y escalable, adaptándose a las diferentes necesidades del negocio.
- Migración de datos desde sistemas anteriores: el sistema debe poder integrar datos históricos provenientes de sistemas legados para asegurar la continuidad de la información y un análisis histórico adecuado.

A partir de los requerimientos y necesidades analizados, se definieron los siguientes historias de usuario que sirven como base para la implementación del sistema. Estos historias de usuario han sido seleccionados considerando su relevancia en la operación diaria y su impacto en la experiencia del usuario final.

Tabla 28
Propuesta de solución – Fase 3 – Historias de usuario

ID	Nombre	Descripción
HU001	Gestión de Productos	Permite la creación, actualización y eliminación de productos en el sistema, asegurando que los datos sean precisos y actualizados.
HU002	Gestión de Inventarios	Controla el stock de productos y genera alertas automáticas cuando se alcanzan niveles bajos, facilitando la gestión de inventarios.
HU003	Realización de Ventas	Registra las transacciones de ventas, actualiza el inventario y guarda los datos de venta para futuros análisis.
HU004	Generación de Facturas	Automatiza la creación de facturas electrónicas detalladas para cada venta, incluyendo la exportación a formatos PDF.
HU005	Gestión de Reportes	Permite la creación de reportes personalizados sobre las ventas e inventarios, proporcionando <i>insights</i> útiles para la planificación estratégica.
HU006	Gestión de Clientes	Administra la información de clientes, permitiendo el registro de nuevos clientes y la actualización de datos existentes.
HU007	Integración con API Externas	Facilita la interoperabilidad del sistema mediante la sincronización de datos con plataformas externas.
HU008	Migración de Datos	Proporciona un proceso seguro para la migración de datos históricos desde sistemas legados.

Nota. Fuente: Elaboración propia.

4.2.8. Requerimientos del sistema

En esta sección, se presentan los requerimientos fundamentales que garantizarán el éxito de la propuesta de diseño. Los requerimientos funcionales detallan las funcionalidades esenciales que el sistema debe proporcionar, mientras que los no funcionales definen las características técnicas que aseguran su rendimiento, seguridad y escalabilidad a largo plazo. La Tabla 29 presenta los detalles de los requerimientos funcionales.

Tabla 29
Requerimientos funcionales

Código	Requerimiento
REQF- 1	El sistema debe integrar la información de inventarios y ventas de forma automática entre todas las plataformas (físicas y digitales), asegurando que los datos se mantengan actualizados y consistentes en tiempo real sin intervención manual.
REQF- 2	El sistema debe ajustar automáticamente los precios en todas las plataformas (mayoristas, minoristas y <i>e-commerce</i>), asegurando que cualquier cambio en las políticas de precios se refleje de manera coherente en todos los canales.
REQF- 3	El sistema debe generar reportes financieros y operativos que reflejen la consolidación de inventarios y ventas, facilitando la trazabilidad y la coherencia en los registros de transacciones.

Código	Requerimiento
REQF- 4	El sistema debe generar órdenes de compra automáticamente basado en niveles de inventario, patrones de ventas y predicciones de demanda, garantizando el abastecimiento sin exceso de inventarios.
REQF- 5	El sistema debe poseer de una arquitectura de microservicios y utilizar APIs para integrar módulos de ventas, inventarios y facturación, permitiendo una comunicación fluida y la fácil incorporación de nuevas funcionalidades.
REQF- 6	El sistema debe permitir la sincronización de inventarios, ventas y precios entre plataformas de comercio electrónico y tiendas físicas para una gestión centralizada y unificada.
REQF- 7	El sistema debe implementar mecanismos que detecten y eliminen automáticamente duplicaciones en los registros de inventarios y ventas, asegurando la consistencia y precisión de la información.
REQF- 8	El sistema debe generar alertas automáticas cuando los inventarios alcancen un umbral mínimo, facilitando la planificación del reabastecimiento.
REQF- 9	El sistema debe monitorear el estado de las órdenes de compra, desde la emisión hasta la recepción de los productos.
REQF- 10	El sistema debe emitir de forma automática las facturas electrónicas de acuerdo con las normativas fiscales locales, reduciendo errores manuales.
REQF- 11	El sistema debe gestionar la administración de clientes y proveedores, permitiendo una vista consolidada de relaciones comerciales y automatizando la comunicación y gestión de pagos.
REQF- 12	El sistema debe permitir la automatización de tareas de marketing como el envío de promociones personalizadas y correos electrónicos automáticos basados en el comportamiento de los clientes.
REQF- 13	El sistema debe permitir la personalización de métricas e indicadores clave (KPIs) para diferentes roles de usuario, mejorando la visualización y análisis de datos.
REQF- 14	El sistema debe mejorar la logística para la distribución de productos, permitiendo la gestión de rutas de entrega entre almacenes y sucursales.

Nota. Fuente: Elaboración propia.

Los requerimientos no funcionales son aquellos que definen las condiciones operativas necesarias para que el sistema cumpla con los estándares de calidad, rendimiento, seguridad y escalabilidad. Estos requisitos son críticos para asegurar que el sistema funcione de manera óptima bajo diversas condiciones y a largo plazo, permitiendo que la solución se adapte a las necesidades cambiantes del negocio. La Tabla 30 detalla los requerimientos no funcionales:

Tabla 30
Requerimientos no funcionales

Código	Requerimiento
REQNF- 1	El sistema debe ser capaz de manejar altos volúmenes de transacciones, manteniendo la sincronización en tiempo real sin afectar el rendimiento, garantizando la eficiencia durante picos de demanda.

Código	Requerimiento
REQNF- 2	El sistema debe implementar autenticación multifactor y cifrado robusto para proteger datos sensibles y cumplir con regulaciones de privacidad, asegurando que solo usuarios autorizados tengan acceso a la información crítica.
REQNF- 3	El sistema debe estar disponible el 99.9% del tiempo, con mecanismos de redundancia para asegurar la continuidad operativa ante fallos y minimizar el tiempo de inactividad.
REQNF- 4	El sistema debe habilitar la capacidad de actualizar el sistema sin interrumpir las operaciones, asegurando una migración fluida que permita realizar mejoras y correcciones sin afectar la actividad diaria.
REQNF- 5	El sistema debe poder escalar horizontalmente en la nube, permitiendo la adición de servidores y la mejorando el manejo de recursos según la demanda.
REQNF- 6	El sistema debe implementar cifrado avanzado para proteger información financiera y crítica, cumpliendo con regulaciones internacionales como GDPR y CCPA, garantizando la protección de la privacidad de los usuarios.
REQNF- 7	El sistema debe contar con mecanismos de respaldo automático y procedimientos de recuperación de datos ante fallos, asegurando la disponibilidad y continuidad operativa del sistema, además de proteger la integridad de la información crítica utilizada en las operaciones diarias.
REQNF- 8	El sistema debe asegurar la compatibilidad con sistemas contables y de auditoría externos, permitiendo la integración fluida con herramientas de gestión empresarial.
REQNF- 9	La interfaz de usuario debe ser intuitiva y adaptable, facilitando la adopción por parte del personal y reduciendo la curva de aprendizaje.
REQNF- 10	El sistema debe implementar un monitoreo avanzado y alertas proactivas para identificar problemas de rendimiento, seguridad o sobrecarga antes de que afecten las operaciones.

Nota. Fuente: Elaboración propia.

4.2.9. Análisis de los requerimientos recopilados

La priorización de los requerimientos definidos en la Tabla 29 y Tabla 30 se lleva a cabo utilizando la técnica MoSCoW, tal como se describe en la sección procesamiento de requerimientos y priorización. Este análisis considera la urgencia, el valor del negocio y la clasificación final de cada requerimiento, asegurando una guía clara para la implementación y el desarrollo del sistema. A continuación, se presentan los criterios para la evaluación y el análisis detallado de los requerimientos:

- **Urgencia:** evalúa la necesidad de implementación inmediata de un requerimiento, basado en su interdependencia con otros elementos del sistema. La escala de 1 a 5 facilita la evaluación del nivel de urgencia, desde extremadamente urgente (5) hasta sin urgencia (1).
- **Valor del negocio:** mide la importancia estratégica de cada requerimiento para la organización. También se califica en una escala de 1 a 5, donde 5 representa un impacto crítico para el éxito del sistema y la satisfacción de los clientes.

- Multiplicación de urgencia y valor: la puntuación final de cada requerimiento se obtiene multiplicando los valores de urgencia y valor del negocio, lo cual permite posicionar el resultado dentro del marco MoSCoW, como se describe, a continuación.
 - *Must* (20-25 puntos): requerimientos esenciales para el funcionamiento básico y éxito del sistema. Su implementación es prioritaria para lograr la interoperabilidad y eficiencia operativa.
 - *Should* (12-16 puntos): mejoras que aumentan la eficiencia del sistema. Estos son los siguientes en prioridad, después de cumplir con los elementos indispensables.
 - *Could* (6-11 puntos): requerimientos adicionales que se implementarán si el tiempo y los recursos lo permiten, aportando mejoras adicionales.
 - *Won't* (1-5 puntos): no aplicable en esta priorización, ya que todos los requerimientos definidos tienen relevancia para el alcance del proyecto.

Tabla 31
Análisis de requerimientos

Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-01	5	5	25	Must
Requerimiento	El sistema debe integrar la información de inventarios y ventas de forma automática entre todas las plataformas (físicas y digitales), asegurando que los datos se mantengan actualizados y consistentes en tiempo real sin intervención manual.			
Justificación	Es esencial para eliminar la duplicación de tareas y garantizar la coherencia de datos entre los sistemas Kyte, FACEL y POS, lo cual es crucial para la toma de decisiones basada en datos confiables.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-02	4	5	20	Must
Requerimiento	El sistema debe ajustar automáticamente los precios en todas las plataformas (mayoristas, minoristas y <i>e-commerce</i>), asegurando que cualquier cambio en las políticas de precios se refleje de manera coherente en todos los canales.			
Justificación	La actualización rápida de precios en todos los canales es clave para la competitividad y para mantener una experiencia de cliente coherente. Esto resuelve problemas identificados de incoherencias de precios.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-03	4	5	20	Must
Requerimiento	El sistema debe generar reportes financieros y operativos que reflejen la consolidación de inventarios y ventas, facilitando la trazabilidad y la coherencia en los registros de transacciones.			
Justificación	Proporciona visibilidad integral del estado financiero y operativo, permitiendo decisiones estratégicas. Es crucial para abordar problemas de reportes inconsistentes detectados.			

Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-04	5	4	20	Must
Requerimiento	El sistema debe generar órdenes de compra automáticamente basado en niveles de inventario, patrones de ventas y predicciones de demanda, garantizando el abastecimiento sin exceso de inventarios.			
Justificación	Mejora la gestión de inventarios y asegura el abastecimiento adecuado, evitando desabastecimientos y excesos de inventario, lo que es esencial para la eficiencia operativa.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-05	4	5	20	Must
Requerimiento	El sistema debe poseer de una arquitectura de microservicios y utilizar APIs para integrar módulos de ventas, inventarios y facturación, permitiendo una comunicación fluida y la fácil incorporación de nuevas funcionalidades.			
Justificación	Facilita la interoperabilidad y la flexibilidad del sistema, abordando la deuda técnica y permitiendo la evolución del sistema hacia una arquitectura más moderna y escalable.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-06	4	4	16	Should
Requerimiento	El sistema debe permitir la sincronización de inventarios, ventas y precios entre plataformas de comercio electrónico y tiendas físicas para una gestión centralizada y unificada.			
Justificación	Mejora la experiencia del cliente y la eficiencia operativa al centralizar la gestión de diferentes canales de ventas, abordando problemas de sincronización y errores de inventario.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-07	4	4	16	Should
Requerimiento	El sistema debe implementar mecanismos que detecten y eliminen automáticamente duplicaciones en los registros de inventarios y ventas, asegurando la consistencia y precisión de la información.			
Justificación	Asegura la consistencia y precisión de la información, reduciendo errores manuales y mejorando la gestión de datos, un problema crítico identificado en la duplicación de tareas.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-08	4	3	12	Should
Requerimiento	El sistema debe generar alertas automáticas cuando los inventarios alcancen un umbral mínimo, facilitando la planificación del reabastecimiento.			
Justificación	Mejora la planificación de reabastecimiento y evita desabastecimientos, abordando las debilidades identificadas en la gestión proactiva de inventarios.			

Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-09	3	4	12	Should
Requerimiento	El sistema debe monitorear en tiempo el estado de las órdenes de compra, desde la emisión hasta la recepción de los productos.			
Justificación	Aumenta la visibilidad y el control sobre el proceso de aprovisionamiento, asegurando tiempos de respuesta s y consistencia en la gestión de órdenes.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-10	4	5	20	Must
Requerimiento	El sistema debe emitir de forma automática las facturas electrónicas de acuerdo con las normativas fiscales locales, reduciendo errores manuales.			
Justificación	Es indispensable para cumplir con las normativas fiscales y mejorar la eficiencia en la facturación, eliminando errores que podrían tener consecuencias legales y financieras.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-11	3	4	12	Should
Requerimiento	El sistema debe gestionar la administración de clientes y proveedores, permitiendo una vista consolidada de relaciones comerciales y automatizando la comunicación y gestión de pagos.			
Justificación	Alineado con la necesidad de unificar la información y mejorar la gestión de relaciones comerciales, asegurando una transición sin pérdida de funcionalidades críticas.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-12	3	3	9	Could
Requerimiento	El sistema debe permitir la automatización de tareas de marketing como el envío de promociones personalizadas y correos electrónicos automáticos basados en el comportamiento de los clientes.			
Justificación	Contribuye a la mejora de la experiencia del cliente y a la retención, pero no es esencial para la primera fase de la implementación del sistema.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-13	3	4	12	Should
Requerimiento	El sistema debe permitir la personalización de métricas e indicadores clave (KPIs) para diferentes roles de usuario, mejorando la visualización y análisis de datos.			
Justificación	Facilita la toma de decisiones informadas por cada departamento, ajustándose a la necesidad de mejorar la visibilidad de información relevante para diferentes áreas.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-14	4	3	12	Should
Requerimiento	El sistema debe mejorar la logística para la distribución de productos, permitiendo la gestión de rutas de entrega entre almacenes y sucursales.			
Justificación	Mejora la eficiencia operativa y reduce costos logísticos, alineándose con la necesidad de mejorar la cadena de suministro y la entrega de productos.			

Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-01	5	5	25	Must
Requerimiento	El sistema debe ser capaz de manejar altos volúmenes de transacciones, manteniendo la sincronización en tiempo real sin afectar el rendimiento, garantizando la eficiencia durante picos de demanda.			
Justificación	Es crucial para mantener la operatividad durante picos de demanda, evitando interrupciones en la sincronización, como se identificó en la necesidad de mejorar la eficiencia operativa.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-02	5	5	25	Must
Requerimiento	El sistema debe implementar autenticación multifactor y cifrado robusto para proteger datos sensibles y cumplir con regulaciones de privacidad, asegurando que solo usuarios autorizados tengan acceso a la información crítica.			
Justificación	Protege la seguridad de la información crítica y cumple con regulaciones de privacidad, abordando la necesidad de fortalecer la seguridad del sistema.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-03	5	5	25	Must
Requerimiento	El sistema debe estar disponible el 99.9% del tiempo, con mecanismos de redundancia para asegurar la continuidad operativa ante fallos y minimizar el tiempo de inactividad.			
Justificación	Garantiza la continuidad operativa del sistema, fundamental para procesos de ventas y facturación, minimizando riesgos de interrupciones.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RF-04	4	4	16	Should
Requerimiento	El sistema debe habilitar la capacidad de actualizar el sistema sin interrumpir las operaciones, asegurando una migración fluida que permita realizar mejoras y correcciones sin afectar la actividad diaria.			
Justificación	Permite el crecimiento y las mejoras continuas sin afectar la operatividad, siendo esencial para mantener la modernización sin comprometer la actividad diaria.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-05	4	5	20	Should
Requerimiento	El sistema debe poder escalar horizontalmente en la nube, permitiendo la adición de servidores y la mejorando el manejo de recursos según la demanda.			
Justificación	Permite adaptarse a cambios en la demanda de manera , facilitando la expansión del sistema y la disminuyendo los costos.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-06	5	5	25	Must
Requerimiento	El sistema debe implementar cifrado avanzado para proteger información financiera y crítica, cumpliendo con regulaciones internacionales como GDPR y CCPA, garantizando la protección de la privacidad de los usuarios.			
Justificación	Es esencial para cumplir con las regulaciones internacionales de privacidad y proteger la información sensible, fortaleciendo la seguridad del sistema.			

Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-07	5	5	25	5
Requerimiento	El sistema debe contar con mecanismos de respaldo automático y procedimientos de recuperación de datos ante fallos, asegurando la disponibilidad y continuidad operativa del sistema, además de proteger la integridad de la información crítica utilizada en las operaciones diarias.			
Justificación	Responde a la necesidad de proteger la continuidad operativa del sistema y los datos críticos en caso de fallos, asegurando la resiliencia del sistema.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-08	3	4	12	Could
Requerimiento	El sistema debe asegurar la compatibilidad con sistemas contables y de auditoría externos, permitiendo la integración fluida con herramientas de gestión empresarial.			
Justificación	Facilita la interoperabilidad con sistemas de terceros, pero no es un requerimiento crítico para la operación inicial del sistema.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-09	3	3	9	Could
Requerimiento	La interfaz de usuario debe ser intuitiva y adaptable, facilitando la adopción por parte del personal y reduciendo la curva de aprendizaje.			
Justificación	Mejora la experiencia del usuario y la productividad, pero no es fundamental para el funcionamiento básico del sistema.			
Código	Urgencia	Valor del negocio	Prioridad	MoSCoW
RNF-10	4	5	20	Should
Requerimiento	El sistema debe implementar un monitoreo avanzado y alertas proactivas para identificar problemas de rendimiento, seguridad o sobrecarga antes de que afecten las operaciones.			
Justificación	Es clave para la detección temprana de problemas, mejorando la estabilidad del sistema y la proactividad en la gestión operativa.			

Nota. Fuente: Elaboración propia.

4.3. Fase 3: Diseño preliminar de la arquitectura

En esta sección, se presenta un análisis preliminar hacia una solución más detallada y viable para abordar las problemáticas de integración de sistemas y automatización de procesos en Hortiluma. Este análisis tiene como objetivo identificar las necesidades empresariales y los requisitos técnicos definidos en fases anteriores, proporcionando una base sólida para el diseño conceptual.

Tabla 32
Operacionalización de las variables – Fase 3

Fase 3: Diseño preliminar de la arquitectura		Instrumentos
Objetivo específico	Elaborar un diseño preliminar de la arquitectura de integración requerida para la cobertura de brechas y el alcance del estado meta, mediante el abandono de tareas y datos duplicados.	Apéndice E. Plantilla de revisión documental Apéndice D. Plantilla de entrevista Apéndice F. Plantilla de observación directa Apéndice G. Plantilla de grupo focal Apéndice H. Plantilla de lluvia de ideas Apéndice I. Plantilla de historias de usuario
Variables de investigación		Sujetos de investigación
Diseño preliminar de la arquitectura de integración, reducción de duplicaciones y mejora de interoperabilidad		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas

Nota. Fuente: Elaboración propia.

4.3.1. Revisión documental

La revisión documental ha sido crucial para identificar las limitaciones técnicas y los problemas de interoperabilidad presentes en los sistemas de Hortiluma. A continuación, se resumen los hallazgos clave y su impacto:

Tabla 33
Propuesta de solución – Fase 3 – Revisión documental

Hallazgo	Apéndice	Descripción	Impacto
Falta de sincronización automática entre sistemas	Apéndice E.11. Plantilla de revisión documental - RD-011 Apéndice E.12. Plantilla de revisión documental - RD-012	Los sistemas Kyte, FACEL y POS no están integrados y requieren actualizaciones manuales. Esto genera inconsistencias en inventarios, precios y reportes financieros.	La falta de sincronización automática afecta la coherencia de los datos, complicando tanto la operatividad diaria como la capacidad de tomar decisiones estratégicas basadas en datos confiables.

Hallazgo	Apéndice	Descripción	Impacto
Duplicación de tareas operativas	Apéndice E.12. Plantilla de revisión documental - RD-012	Los empleados deben ingresar los mismos datos en tres sistemas diferentes, lo que aumenta el riesgo de errores y retrabajo.	La duplicación ralentiza los procesos y genera inconsistencias en los reportes operativos y financieros, consumiendo recursos que podrían ser usados en actividades más estratégicas.
Actualización manual de precios	Apéndice E.13. Plantilla de revisión documental - RD-013	La actualización de precios se realiza de forma manual entre Kyte, FACEL y POS, generando discrepancias entre los precios en las distintas plataformas de ventas.	La inconsistencia en los precios afecta la coherencia operativa, lo que puede generar problemas en la experiencia del cliente y en la rentabilidad de las ventas.
Inconsistencias en los niveles de inventario	Apéndice E.14. Plantilla de revisión documental - RD-014	Se detectaron discrepancias de hasta un 15% en los niveles de inventario entre Kyte y FACEL lo que afecta la planificación de compras y la disponibilidad de productos.	La falta de coherencia en los inventarios puede generar desabastecimiento o exceso de stock, afectando el costo operativo y la experiencia del cliente.
Ausencia de mecanismos de verificación automática	Apéndice E.15. Plantilla de revisión documental - RD-015	No existen herramientas automáticas para verificar la coherencia de los datos entre sistemas, y las auditorías de inventario se realizan manualmente cada trimestre.	La falta de verificación automática genera retrasos en la identificación y corrección de errores, aumentando el riesgo de discrepancias.
Discrepancias en reportes financieros y operativos	Apéndice E.14. Plantilla de revisión documental - RD-014	La falta de integración entre los sistemas provoca inconsistencias en los reportes financieros y operativos.	Los errores en los reportes generan desconfianza en la información, afectando tanto la gestión operativa como la planificación financiera y estratégica.
Retrasos en la auditoría de inventarios	Apéndice E.15. Plantilla de revisión documental - RD-015	La auditoría de inventarios se realiza manualmente cada tres meses, lo que retrasa la detección de errores.	La falta de auditorías frecuentes y automatizadas aumenta el riesgo de que los errores en inventarios afecten las operaciones durante largos períodos de tiempo, dificultando la planificación de compras.

Nota. Fuente: Elaboración propia.

4.3.2. Entrevistas

Las entrevistas realizadas con los responsables de ventas y administración de Hortiluma proporcionaron una visión detallada de los problemas operativos actuales y las expectativas sobre el nuevo sistema. Los principales desafíos identificados están relacionados con la falta de integración, la duplicación de tareas y la necesidad de automatización. Estos desafíos fueron recopilados en tres plantillas que cubren aspectos críticos como la interoperabilidad, la mejora operativa y la deuda técnica. A continuación, se detallan los hallazgos clave:

Tabla 34
Propuesta de solución – Fase 3 – Entrevistas

Hallazgo	Apéndice	Descripción	Impacto
Falta de integración entre sistemas	Apéndice D.7. Plantilla de entrevista - ENT-007	Los entrevistados destacaron que la falta de sincronización entre Kyte, FACEL y POS obliga a los empleados a ingresar los mismos datos en cada sistema, lo que genera duplicaciones y errores.	La falta de integración afecta la coherencia de los datos operativos y financieros, lo que incrementa los errores y retrabajos, reduciendo la eficiencia operativa.
Duplicación de tareas	Apéndice D.8. Plantilla de entrevista - ENT-008	Las tareas duplicadas, como la actualización manual de inventarios en tres sistemas, son una constante en la operación diaria, lo que afecta la precisión de los reportes y los procesos.	La duplicación de tareas consume recursos y tiempo, generando inconsistencias en la planificación de inventarios y errores en los reportes financieros.
Errores en la planificación y operatividad	Apéndice D.9. Plantilla de entrevista - ENT-009	La falta de integración y la duplicación de datos generan constantes errores en los reportes de ventas e inventarios, lo que afecta la toma de decisiones y la planificación de compras.	Los errores en los reportes y la planificación provocan pérdida de tiempo, recursos, y desorganización operativa, especialmente en la gestión de inventarios y ventas.
Impacto en la experiencia del cliente	Apéndice D.7. Plantilla de entrevista - ENT-007	La falta de sincronización en tiempo real afecta la disponibilidad de productos y la coherencia de los precios, lo que genera frustración en los clientes y pérdidas de ventas.	La falta de precisión en la disponibilidad de productos reduce la satisfacción del cliente y provoca pérdidas de ventas potenciales.
Deuda técnica en el sistema POS	Apéndice D.9. Plantilla de entrevista - ENT-009	El sistema POS presenta problemas técnicos que impiden la automatización de tareas clave, lo que dificulta la integración con otros sistemas y limita las capacidades de expansión.	La deuda técnica en el sistema POS retrasa la implementación de mejoras y afecta la capacidad de Hortiluma para automatizar y sincronizar procesos.

Hallazgo	Apéndice	Descripción	Impacto
Necesidad de automatización y sincronización en tiempo real	Apéndice D.8. Plantilla de entrevista - ENT-008	Los responsables de ventas indicaron que la automatización de inventarios y la sincronización en tiempo real entre sistemas reducirían los errores y mejorarían la coherencia operativa.	La automatización y sincronización en tiempo real permitirían precisión en los reportes y una planificación de inventarios, mejorando los recursos operativos.

Nota. Fuente: Elaboración propia.

4.3.3. Observación directa

La observación directa de los procesos operativos de Hortiluma permitió identificar de forma detallada cómo la falta de integración entre los sistemas Kyte, FACEL y POS genera ineficiencias, duplicación de tareas y riesgos operativos. Los empleados ingresan manualmente los datos en los tres sistemas, lo que causa retrasos y errores en la gestión diaria de ventas, inventarios y facturación. La falta de sincronización en tiempo real fue uno de los problemas más evidentes, lo que afecta negativamente la planificación operativa y financiera, así como la coherencia de los reportes generados. En la Tabla 35, se presentan los hallazgos clave, basados en dos plantillas que cubren los aspectos más críticos:

Tabla 35
Propuesta de solución – Fase 3 – Observación directa

Hallazgo	Apéndice	Descripción	Impacto
Duplicación de tareas en la actualización de inventarios y ventas	Apéndice F.8. Plantilla de observación directa - OBS-008	Los empleados deben ingresar manualmente los datos de inventarios y ventas en tres sistemas distintos (Kyte, FACEL y POS).	La duplicación de tareas incrementa la carga de trabajo y el riesgo de errores, afectando la eficiencia y la coherencia de los datos, lo que complica la planificación de compras y ventas.
Falta de sincronización en tiempo real	Apéndice F.9. Plantilla de observación directa - OBS-009	La actualización de inventarios y precios no se realiza en tiempo real, lo que provoca incoherencias en los datos entre los sistemas y retrasos en los reportes operativos y financieros.	La ausencia de sincronización en tiempo real afecta la toma de decisiones basadas en datos precisos, lo que impacta negativamente la gestión de inventarios, ventas y la satisfacción del cliente.
Errores en la conciliación de datos financieros	Apéndice F.8. Plantilla de observación directa - OBS-008	Se observaron discrepancias en los niveles de stock y errores en la actualización de precios y facturas, lo que afecta la conciliación de datos financieros entre los sistemas.	La falta de precisión en los datos financieros y operativos impacta la planificación y control de inventarios, lo que genera problemas en la rentabilidad del negocio y la eficiencia operativa.

Hallazgo	Apéndice	Descripción	Impacto
Ausencia de monitoreo automatizado	Apéndice F.9. Plantilla de observación directa - OBS-009	El monitoreo de inventarios y ventas se realiza manualmente, lo que retrasa la disponibilidad de información precisa para la toma de decisiones.	La falta de monitoreo automatizado incrementa los tiempos de respuesta ante errores o discrepancias, lo que afecta tanto la planificación operativa como la experiencia del cliente.

Nota. Fuente: Elaboración propia.

4.3.4. Grupo focal

El grupo focal fue esencial para validar y ajustar las propuestas desarrolladas en fases previas del proyecto, centrándose en aspectos como la integración de sistemas y la automatización de procesos. Los participantes, entre los que se encontraban el administrador del negocio, responsables de ventas minoristas y mayoristas, compartieron sus experiencias y expectativas para el nuevo sistema. Las discusiones ayudaron a alinear estas expectativas con el diseño preliminar, destacando los puntos clave que necesitaban ser abordados para mejorar la eficiencia operativa. Los principales hallazgos se encuentran detallados en Apéndice G.4. Plantilla de grupo focal - GF-004.

- Los participantes resaltaron que la falta de integración entre los sistemas Kyte, FACEL y POS es uno de los mayores desafíos que enfrentan actualmente. Esto genera inconsistencias en los datos, lo que afecta la planificación de inventarios, ventas y la toma de decisiones estratégicas. Además, acordaron que una solución que permita una integración fluida entre los sistemas es esencial para eliminar errores y mejorar la eficiencia operativa.
- Se enfatizó que la duplicación de tareas es un problema recurrente debido a la falta de interoperabilidad entre los sistemas. Los empleados deben ingresar los mismos datos manualmente en diferentes plataformas, lo que aumenta el riesgo de errores y retrasa los procesos operativos. Así mismo, los participantes concordaron en que la integración de API y *middleware* es fundamental para asegurar un flujo de datos continuo entre los sistemas y eliminar estas tareas repetitivas.
- Los participantes destacaron la importancia de automatizar procesos como la actualización de inventarios, precios y generación de reportes. La introducción de tecnologías como la Automatización Robótica de Procesos (RPA) se consideró crucial para reducir la intervención manual en tareas rutinarias, lo que incrementará la precisión de los datos y mejorará la eficiencia operativa.
- La sincronización en tiempo real fue identificada como un elemento esencial para asegurar la coherencia de los reportes de ventas e inventarios. Además, los participantes señalaron que la falta de sincronización actual provoca discrepancias en los datos y errores en la planificación diaria, lo que afecta tanto la operatividad como la satisfacción del cliente.
- El grupo coincidió en que el sistema debe ser modular y escalable, lo que permitirá integrar nuevos módulos o funcionalidades en el futuro sin comprometer la interoperabilidad ni la eficiencia. Esta flexibilidad será vital para que Hortiluma pueda adaptarse a cambios en sus operaciones a medida que crece o evoluciona su modelo de negocio.

4.3.5. Lluvia de ideas

Las sesiones de lluvia de ideas fueron esenciales para identificar los componentes clave del nuevo sistema, con un enfoque en la integración de sistemas y la eliminación de duplicaciones de tareas operativas. Los participantes, que incluían al administrador del negocio y a los responsables de ventas, discutieron propuestas que se alinearon estrechamente con las fases 1 y 2 del proyecto. A continuación, se presentan los principales puntos abordados, basados en los hallazgos del Apéndice H.4. Plantilla de lluvia de ideas - LLUV-004.

- Se resaltó la necesidad de un sistema modular que permita la integración completa de los sistemas de ventas, inventarios y facturación. Esta arquitectura modular garantizaría la interoperabilidad entre Kyte, FACEL y POS, eliminando la necesidad de ingresar datos manualmente y de forma repetida en cada sistema. Como resultado, se reducirían los errores operativos y se mejoraría la coherencia de la información, facilitando la gestión de datos de manera más precisa y .
- Los participantes coincidieron en la importancia de implementar un sistema que permita la sincronización en tiempo real entre los sistemas. Esta capacidad garantizaría que los datos de inventarios, ventas y precios estén actualizados simultáneamente en todas las plataformas, evitando discrepancias que actualmente afectan la planificación operativa y los reportes financieros. La sincronización en tiempo real también permitiría tomar decisiones más informadas y oportunas.
- Se destacó la relevancia de automatizar tareas repetitivas, como la actualización de inventarios y precios, la generación de reportes y la facturación. La automatización no solo mejoraría la precisión de los datos, sino que también reduciría la carga de trabajo manual y eliminaría la duplicación de tareas, mejorando la eficiencia operativa. Esto liberaría tiempo y recursos que podrían ser dedicados a tareas más estratégicas.
- El diseño del sistema debe ser escalable y adaptable, permitiendo la adición de nuevos módulos y funcionalidades en el futuro sin interrumpir las operaciones actuales. Esta flexibilidad asegurará que el sistema pueda crecer junto con el negocio, permitiendo que Hortiluma se adapte fácilmente a nuevas necesidades y expansiones, manteniendo la eficiencia operativa y la coherencia en todos los sistemas.

4.3.6. Historias de usuario

El modelo de historias de usuario ha sido diseñado para cubrir todos los requerimientos establecidos del proyecto de integración de las funcionalidades de Hortiluma, enfocado en alcanzar los beneficios esperados. La nueva plataforma centralizada busca resolver problemas operativos y estratégicos que actualmente limitan la eficiencia de la empresa, como la duplicación de tareas, la inconsistencia de los datos, la complejidad del flujo de trabajo, la falta de flexibilidad de respuesta y la carencia de capacidades para aprovechar nuevas oportunidades de expansión. Cada historia de usuario ha sido desarrollada para apoyar directamente estos beneficios, contribuyendo a una operativa más escalable.

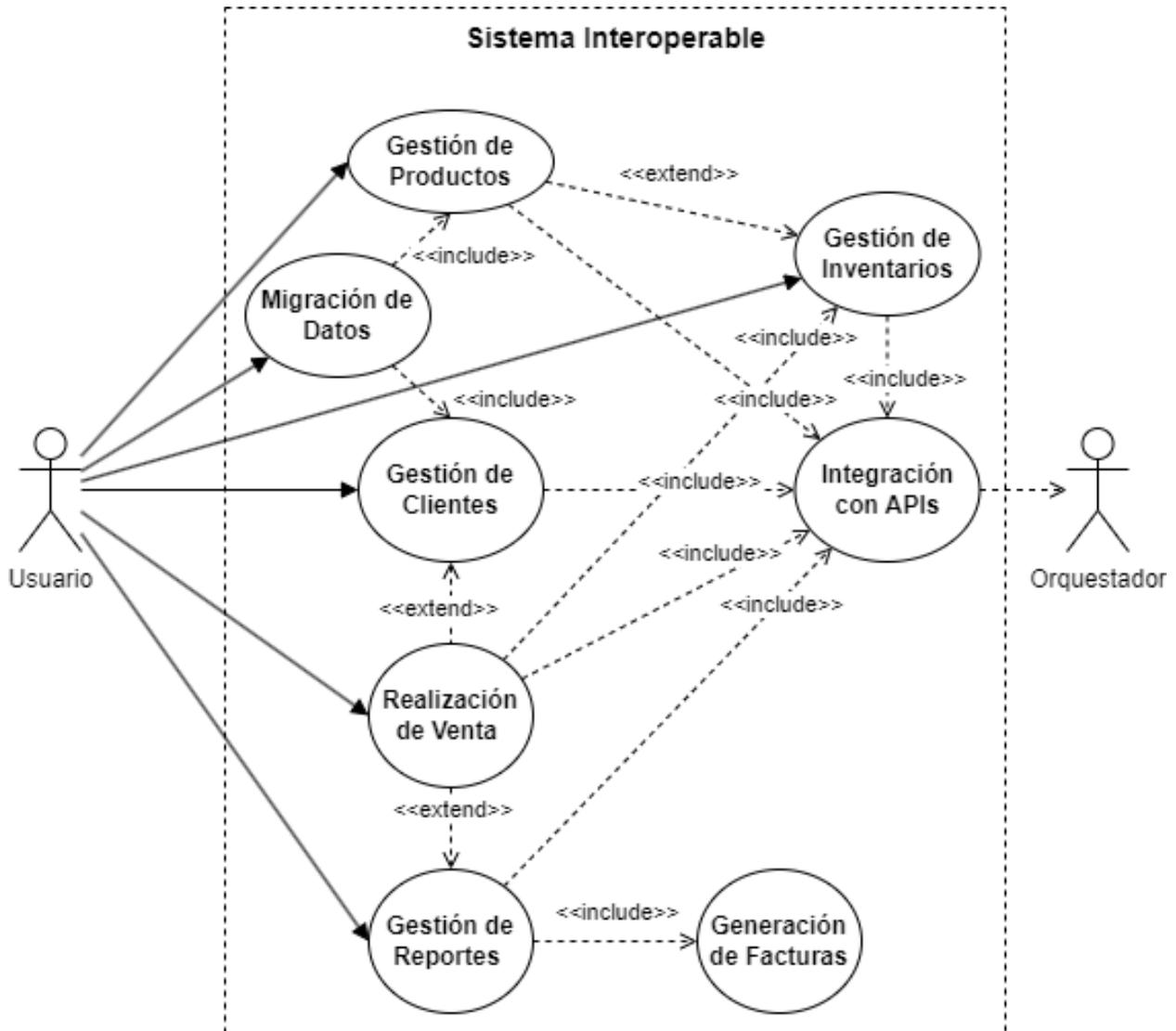
Los historias de usuario se han consolidado a partir del análisis de los requerimientos recopilados, asegurando que cada uno tenga un propósito claro, esté alineado con los objetivos, y cumpla con los requerimientos del sistema. La Tabla 36 especifica su propósito y justificación para garantizar que se abarque la totalidad de los requerimientos.

Tabla 36
Propuesta de solución – Fase 3 – Historias de usuario

Apéndice	Propósito	Justificación
Apéndice I.1. Plantilla de historias de usuario - HU-001	Centralizar la gestión de inventarios y ventas para eliminar la duplicación de tareas.	Mejora la entrada de datos y garantiza la coherencia de la información en tiempo real, facilitando la automatización y la consistencia en todas las plataformas.
Apéndice I.2. Plantilla de historias de usuario - HU-002	Controlar y gestionar el inventario de forma proactiva.	Asegura que los niveles de inventario estén actualizados y automatiza la generación de alertas para evitar desabastecimientos, mejorando la planificación operativa.
Apéndice I.3. Plantilla de historias de usuario - HU-003	Facilitar la realización de ventas y actualización del inventario.	Automatiza la actualización de inventarios tras cada venta, asegurando que los datos estén siempre sincronizados para análisis posteriores y toma de decisiones.
Apéndice I.4. Plantilla de historias de usuario - HU-004	Automatizar la generación de facturas electrónicas.	Simplifica el proceso de facturación, reduce errores manuales y cumple con las normativas fiscales locales, asegurando la precisión y consistencia de los registros.
Apéndice I.5. Plantilla de historias de usuario - HU-005	Proveer reportes detallados y análisis de datos para la toma de decisiones.	Proporciona visibilidad de la información crítica para la planificación estratégica, mejorando la eficiencia en la generación de reportes personalizados.
Apéndice I.6. Plantilla de historias de usuario - HU-006	Gestionar la información de los clientes de forma centralizada.	Asegura la disponibilidad de datos actualizados de los clientes, mejorando la personalización de servicios y facilitando la comunicación con los clientes.
Apéndice I.7. Plantilla de historias de usuario - HU-007	Facilitar la integración con sistemas externos a través de API.	Garantiza la interoperabilidad y escalabilidad de la plataforma, asegurando que el sistema pueda integrarse mente con otras plataformas externas.
Apéndice I.8. Plantilla de historias de usuario - HU-008	Asegurar la migración de datos históricos de manera .	Facilita la integración de datos legados, manteniendo la consistencia y disponibilidad de la información histórica para un análisis completo.

Nota. Fuente: Elaboración propia.

Figura 5
Diagrama de historias de usuario – Modelo general de historias de usuario



Nota. Fuente: Elaboración propia.

5. Propuesta de solución

Este capítulo presenta la solución integral para resolver los problemas detectados en Hortiluma, abordando los desafíos de interoperabilidad, duplicación de tareas y falta de automatización. La propuesta se basa en los resultados obtenidos en las fases de diagnóstico y en la definición del estado meta, con el objetivo de cumplir con los objetivos del Trabajo Final de Graduación (TFG). La propuesta incluye también una evaluación de viabilidad financiera, con un análisis de costos y beneficios para la empresa.

5.1. Fase 3: Diseño preliminar de la arquitectura

Este capítulo presenta una solución detallada y viable para la problemática de integración de sistemas y automatización de procesos en Hortiluma, abordando las necesidades empresariales y los requisitos técnicos identificados. Se propone un diseño de *software* escalable y flexible que promueva la interoperabilidad entre los sistemas Kyte, FACEL y POS, automatizando procesos críticos y mejorando la eficiencia operativa.

5.1.1. Diseño de la propuesta

El diseño de la propuesta para Hortiluma está estructurado para garantizar una integración efectiva de sus funcionalidades clave, siguiendo los principios de interoperabilidad, escalabilidad, modularidad y mantenimiento continuo. Estos principios son esenciales para que la nueva plataforma se adapte a las necesidades actuales y futuras de la empresa, ofreciendo una operación estable y flexible.

La propuesta adopta una metodología enfocada en la integración de sistemas y verificación de interoperabilidad, que responde a los desafíos actuales de Hortiluma, como la duplicación de tareas y la falta de sincronización automática, mediante los siguientes aspectos:

- Eliminación de tareas redundantes: a través de la integración entre sistemas de *software*, se centralizan funciones críticas como la gestión de inventarios y ventas. Esto mejora el uso de recursos y reduce la necesidad de actualizaciones manuales en múltiples sistemas. En este contexto, la historia de usuario Gestión de Productos (Apéndice I.1. Plantilla de historias de usuario - HU-001) asegura que cualquier cambio en los productos y su inventario se sincronice automáticamente en todos los canales de venta.
- Consistencia de datos asegurada: un sistema robusto de monitoreo valida la integridad de los datos antes de que se utilicen en decisiones estratégicas, alineado con los principios de verificación y validación del *software* (v&v). Esto es respaldado por la historia de usuario Monitoreo de Integridad de Datos (Apéndice I.7. Plantilla de historias de usuario - HU-007), que asegura que los datos se mantengan consistentes y libres de errores a través de la verificación en tiempo real.
- Migración de datos : la migración de datos históricos se gestiona mediante *scripts* y *jobs* controlados en lugar de un microservicio continuo. Estos *scripts* garantizan que los datos se transfieran una única vez desde sistemas antiguos a la nueva plataforma. Este proceso es facilitado por la historia de usuario Migración de Datos Históricos (Apéndice I.8. Plantilla de historias de usuario - HU-008), asegurando que la migración se complete sin pérdida de información y sin interrumpir las operaciones.

Se ha optado por una arquitectura de microservicios que soporta un entorno dinámico y escalable. Las ventajas clave de esta arquitectura incluyen:

- Interoperabilidad mejorada: los módulos, como la Gestión Centralizada de Inventarios y Ventas (Apéndice I.1. Plantilla de historias de usuario - HU-001) y la Integración con Plataformas Externas (Apéndice I.6. Plantilla de historias de usuario - HU-006), se comunican a través de un API REST. Esto asegura que las funciones de la plataforma sean accesibles desde diferentes sistemas y dispositivos, facilitando la integración interna y externa.
- Escalabilidad y adaptabilidad: cada microservicio es independiente y escalable, permitiendo que la plataforma evolucione sin comprometer su operatividad. Esto es relevante para la Realización de Ventas (Apéndice I.3. Plantilla de historias de usuario - HU-003), que puede manejar mayores volúmenes de transacciones sin afectar la estabilidad general.
- Modularidad y mantenimiento simplificado: la independencia de cada microservicio facilita actualizaciones y mantenimiento, alineándose con los principios de diseño solid, y permitiendo que el sistema evolucione de forma sostenible.

Cada módulo en el diseño de la propuesta cubre una función crítica para las operaciones de Hortiluma, y las historias de usuario asociadas se detallan para asegurar la claridad en cada proceso:

- Gestión Centralizada de Inventarios y Ventas (Apéndice I.1. Plantilla de historias de usuario - HU-001): este módulo centraliza la administración de inventarios y ventas, eliminando duplicaciones y asegurando la coherencia de los datos en tiempo real en todos los canales de venta. Asegura que el usuario pueda crear, actualizar o eliminar productos en el sistema, reflejando los cambios de manera sincronizada en todas las plataformas de venta. Además, incluye funcionalidades CRUD para consultar detalles de productos, actualizar precios y remover productos cuando sea necesario.
- Gestión de Inventarios Proactiva (Apéndice I.2. Plantilla de historias de usuario - HU-002): este módulo facilita la actualización y monitoreo de inventarios en tiempo real, permitiendo a los usuarios actualizar cantidades y gestionar productos. Se generan alertas automáticas cuando el inventario alcanza niveles mínimos, mejorando la disponibilidad de productos.
- Realización de Ventas (Apéndice I.3. Plantilla de historias de usuario - HU-003): automatiza el proceso de venta, desde la selección de productos hasta la actualización de inventarios y la generación de recibos. La información se sincroniza para reportes posteriores y análisis de tendencias.
- Generación de Facturas Automatizada (Apéndice I.4. Plantilla de historias de usuario - HU-004): facilita la creación y envío de facturas electrónicas, asegurando cumplimiento con normativas fiscales y mejorando la eficiencia en procesos de facturación.
- Generación de Reportes Automatizados (Apéndice I.5. Plantilla de historias de usuario - HU-005): proporciona reportes detallados de ventas e inventarios, ofreciendo herramientas de análisis que apoyan la toma de decisiones estratégicas.
- Integración con Plataformas Externas (Apéndice I.6. Plantilla de historias de usuario - HU-006): a través de APIs, este módulo asegura la interoperabilidad con plataformas externas como sistemas de *e-commerce*, permitiendo que el sistema se adapte a nuevas oportunidades sin realizar modificaciones complejas.

- Monitoreo de Integridad de Datos (Apéndice I.7. Plantilla de historias de usuario - HU-007): proporciona monitoreo en tiempo real para asegurar la consistencia y calidad de los datos, reduciendo riesgos de fallos y manteniendo la confiabilidad de la información en el sistema.
- Migración de Datos Históricos (Apéndice I.8. Plantilla de historias de usuario - HU-008): se utiliza para integrar datos antiguos en la nueva plataforma mediante procesos controlados de migración única. La funcionalidad asegura que la información histórica se preserve y esté disponible para el análisis completo de datos, sin afectar la operativa diaria.

El diseño de la propuesta se fundamenta en varios principios para garantizar la calidad y eficiencia del sistema:

- Automatización de procesos críticos: se automatizan procesos claves como la generación de reportes, actualización de inventarios y migración de datos. Esto minimiza la intervención manual y mejora la eficiencia operativa.
- Cobertura completa de requerimientos: la matriz de cobertura de requerimientos asegura que todos los requerimientos funcionales y no funcionales se cumplan en el diseño, respaldado por los historias de usuario, como gestión de productos, inventarios y ventas.
- Uso de API REST: el uso de un API facilita la integración entre microservicios y con otros sistemas, aportando flexibilidad y escalabilidad para adaptarse a cambios futuros en el entorno de desarrollo.

La solución propuesta, denominada POSIntegrado, prioriza la interoperabilidad, escalabilidad y modularidad mediante un diseño de alto nivel que permite visualizar cómo los componentes del sistema interactúan entre sí y con sistemas externos. Este enfoque asegura que la plataforma cumpla los objetivos del proyecto sin necesidad de incluir vistas detalladas de las capas físicas de la arquitectura, manteniendo un enfoque en el diseño conceptual necesario para la implementación efectiva de la solución.

5.1.2. Diseño de *software*

El diseño de *software* se enfoca en establecer una arquitectura que garantice que el sistema sea modular, escalable, mantenible y seguro, cumpliendo con los requisitos tanto funcionales como no funcionales. Basado en los principios de diseño recomendados por Pressman (2005) y Jaiswal (2019), el proceso se estructura en cinco fases clave que abarcan desde la definición de datos hasta la implementación y pruebas. A continuación, se detalla cada fase.

5.1.2.1. Diseño de datos

El diseño de datos se centra en definir una estructura organizada que permita a la plataforma de Hortiluma procesar y generar información de manera consistente y mejorada. Este diseño es crucial para facilitar la interoperabilidad entre los sistemas existentes y la nueva plataforma, garantizando una gestión adecuada de la información para apoyar tanto las decisiones operativas como las estratégicas.

La organización de los datos sigue los principios de escalabilidad, modularidad y consistencia, atendiendo las siguientes áreas clave: migración de datos, integración de sistemas y verificación de interoperabilidad, así como automatización y estándares de datos. Cada componente de la estructura de datos está diseñado para asegurar que los procesos críticos se realicen de forma automática, minimizando la intervención manual y reduciendo los errores humanos.

La Tabla 37 presenta los elementos de información críticos para el funcionamiento de la nueva plataforma de Hortiluma. Estos datos fueron identificados a partir de los requerimientos recopilados durante la fase de análisis y reflejan las necesidades del negocio para mejorar su operatividad.

Tabla 37

Propuesta de solución – Fase 3 – Datos identificados para POSIntegrado.

Nombre	Objetivo	Característica
Inventario	Representa la información sobre el inventario de productos disponibles en los almacenes de Hortiluma.	Permite actualizaciones en tiempo real, asegurando la consistencia y disponibilidad de productos.
Venta	Registra las transacciones de ventas realizadas en la plataforma, tanto en línea como en tiendas físicas.	Incluye detalles como fecha, productos, cantidades y precios, lo que facilita un análisis detallado.
Cliente	Contiene datos relevantes sobre los clientes, como historial de compras, preferencias y datos de contacto.	Facilita la personalización de promociones y estrategias de fidelización, alineándose con las necesidades de <i>marketing</i> .
Reporte	Genera un análisis detallado de las ventas realizadas en un periodo determinado o del inventario disponible, destacando tendencias y comparativas.	Permite visualización gráfica y exportación en formatos como PDF y Excel, apoyando decisiones estratégicas.
Alertas de inventario	Emite notificaciones automáticas cuando el inventario de un producto cae por debajo de un umbral definido.	Se activan en tiempo real y son personalizables según las necesidades del usuario, mejorando la gestión de inventarios.
API de integración	Facilitan la interoperabilidad entre microservicios y con sistemas externos, como plataformas de <i>e-commerce</i> y sistemas contables.	Basadas en REST, ofrecen una integración flexible y segura, respaldando la escalabilidad de la solución.
Producto	Almacena información detallada de los productos comercializados, incluyendo descripciones, precios y categorías.	Los datos se utilizan para gestionar el inventario y las promociones, facilitando la toma de decisiones en <i>marketing</i> .
Inventario	Representa la información sobre el inventario de productos disponibles en los almacenes de Hortiluma.	Permite actualizaciones en tiempo real, asegurando la consistencia y disponibilidad de productos.

Nota. Fuente: Elaboración propia.

El diseño de datos es fundamental para lograr los objetivos, promoviendo una integración fluida entre sistemas actuales y la nueva plataforma. Esta centralización minimiza tareas redundantes al integrar la gestión de inventarios y ventas, lo cual facilita una operación consistente en todos los módulos del sistema.

El diseño de datos se alinea con la arquitectura de microservicios implementada. Cada microservicio maneja un conjunto específico de datos, asegurando la independencia de componentes y facilitando la escalabilidad autónoma de servicios críticos. Este enfoque modular permite que el sistema responda a cambios y mejoras sin impactar la operación de toda la plataforma, en línea con los principios de diseño solid

- Microservicios para la gestión de inventarios y ventas: permiten actualizar el inventario en tiempo real y se integran con el módulo de alertas automáticas, lo que garantiza una respuesta inmediata ante cambios en el inventario.
- Microservicios para reportes: compilan y analizan datos de ventas e inventarios, generando reportes clave para el análisis de tendencias y la planificación estratégica.

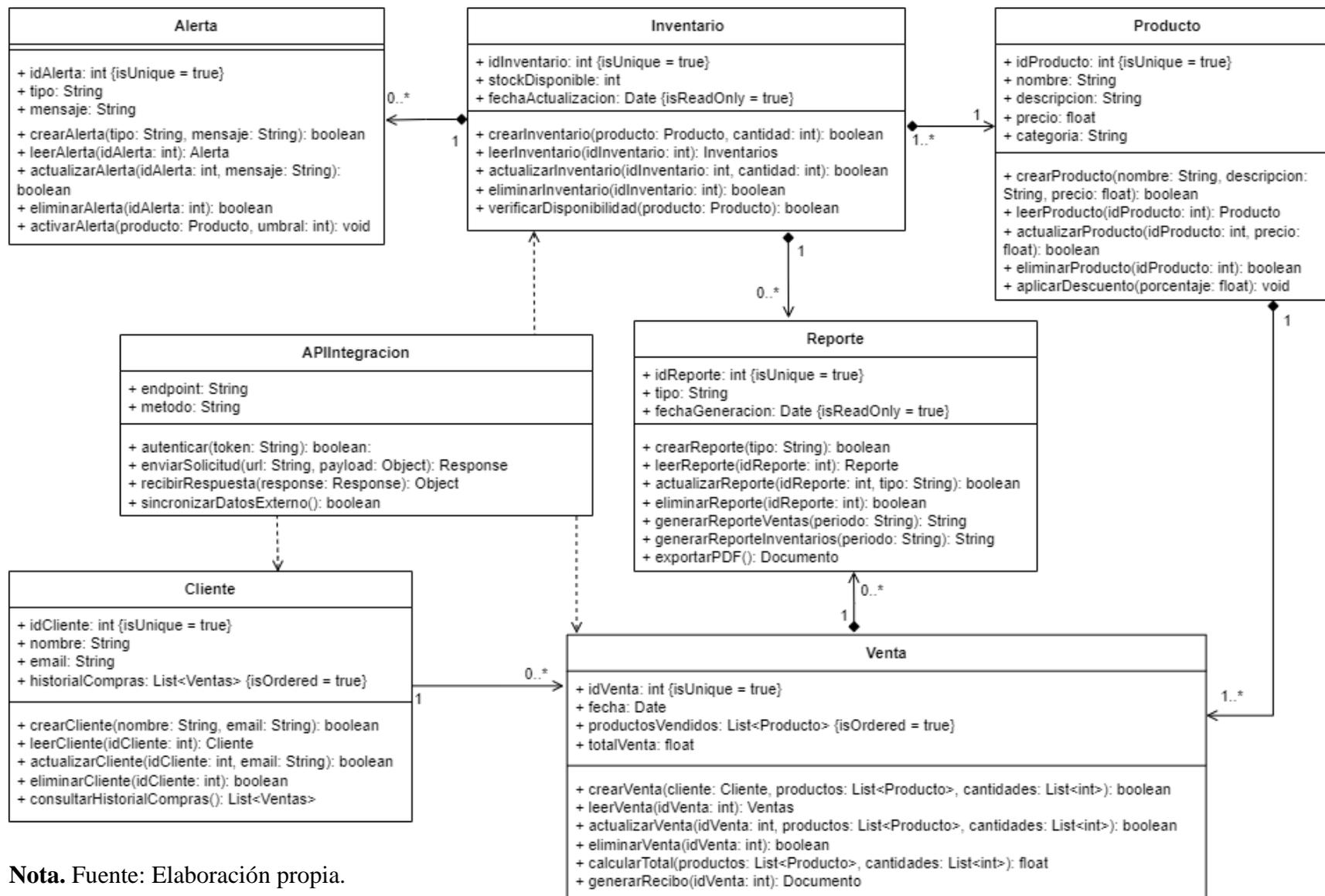
La automatización y estándares de datos es un componente central en el diseño de datos. Los procesos automáticos de actualización de inventarios y generación de reportes minimizan la intervención manual, reduciendo errores y mejorando la eficiencia. La sincronización de datos con sistemas externos a través de un API REST asegura que la información esté actualizada en las diversas plataformas que utiliza Hortiluma.

Para garantizar la calidad y exactitud de los datos, se han implementado procedimientos de verificación y validación del *software* (v&v), que aseguran la precisión de los datos utilizados en reportes y análisis. Este control incluye la validación de datos migrados desde sistemas legados y la verificación de consistencia en las transacciones registradas. Este diseño de datos es esencial para cumplir los objetivos, ya que:

- Facilita una integración fluida entre los sistemas actuales y la nueva plataforma, centralizando la gestión de inventarios y ventas, lo que mejora el flujo de información.
- Permite una toma de decisiones más ágil, gracias a la disponibilidad y consistencia de los datos.
- Asegura la escalabilidad sostenible de la plataforma, permitiendo a Hortiluma adaptarse a las demandas del mercado sin comprometer la calidad de sus servicios.

Este enfoque en la automatización y consistencia de la información prepara a Hortiluma para enfrentar futuros desafíos, contribuyendo al éxito de la plataforma. La estructura de datos asegura que la información esté siempre disponible, precisa y confiable, lo que es clave para la toma de decisiones estratégicas y operativas. En la Figura 6 se incluye un diagrama de clases que ilustra la relación entre los datos detallados en la Tabla 37, aportando una visión clara de cómo se estructuran y relacionan los datos críticos para el sistema.

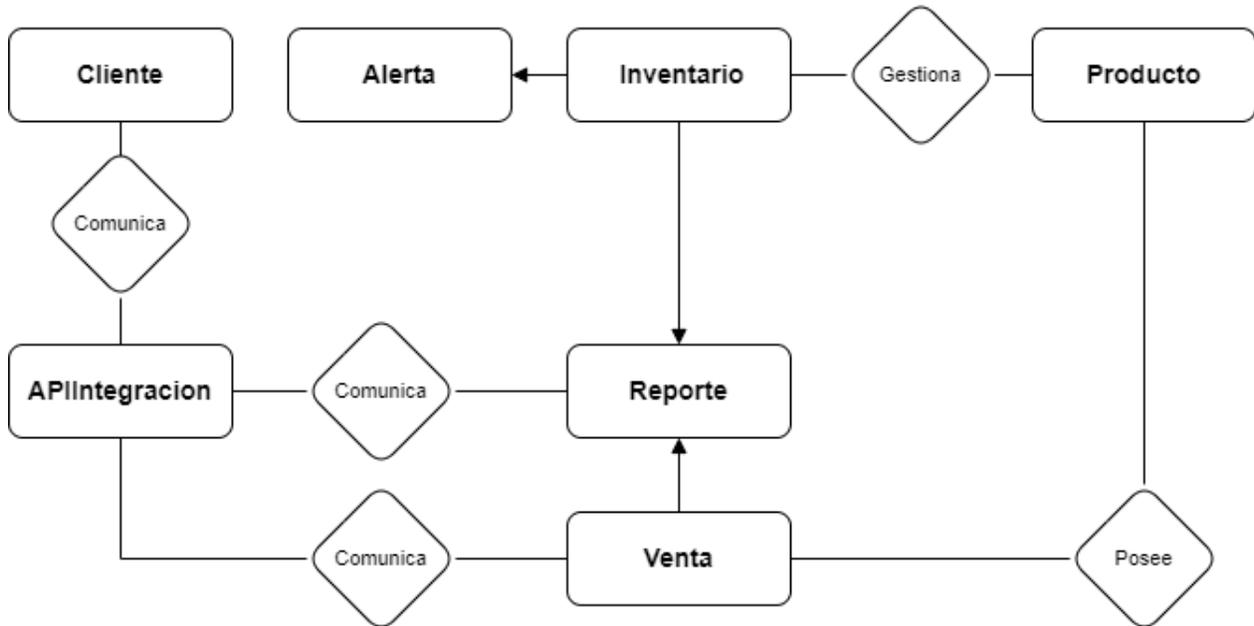
Figura 6
Diagrama de clases – POSIntegrado



Nota. Fuente: Elaboración propia.

A manera de simplificar la comprensión de la Figura 6, en la Figura 7 se muestra el modelo de datos conceptual del sistema POSIntegrado.

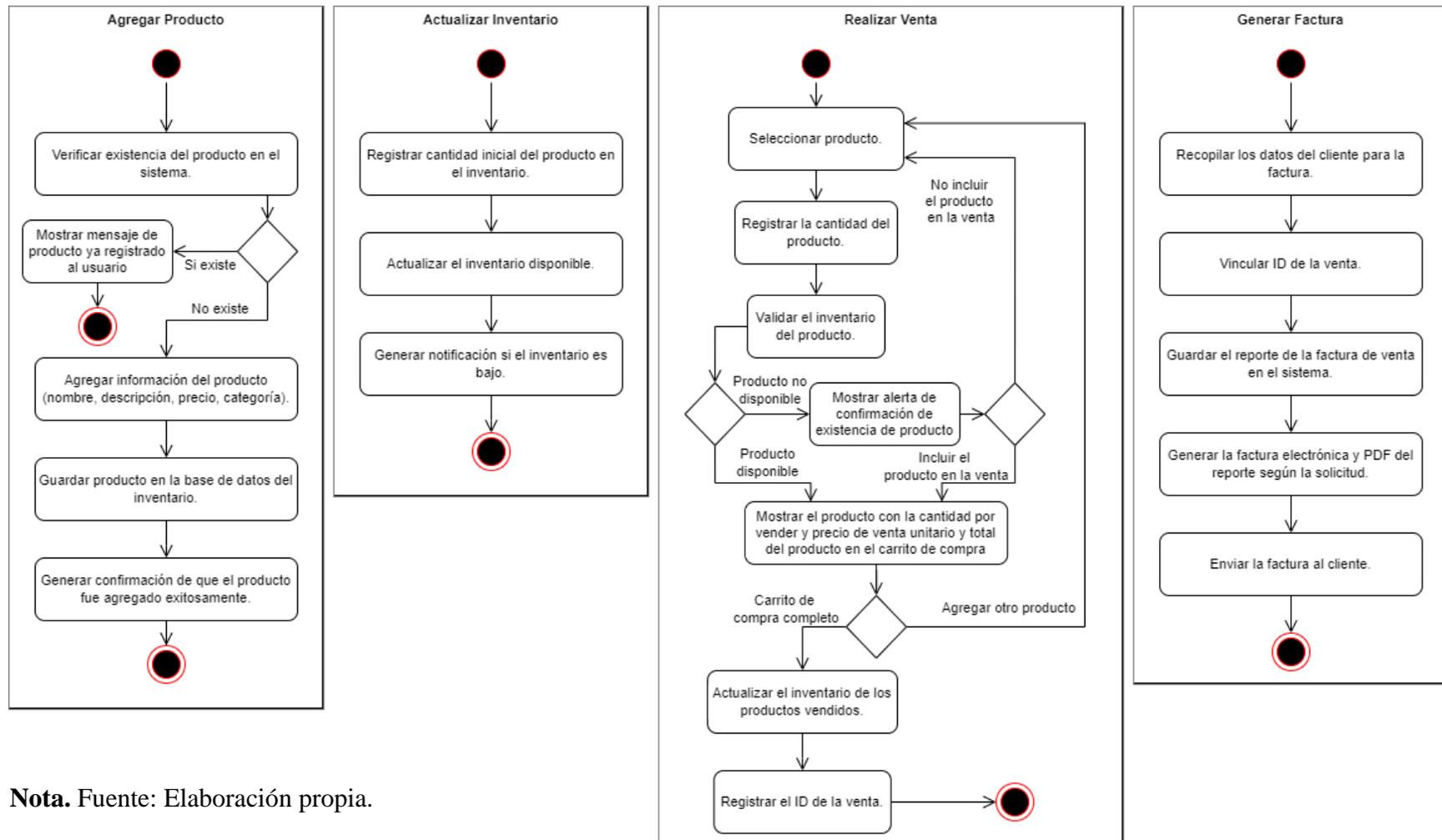
Figura 7
Modelo de datos conceptual – POSIntegrado



Nota. Fuente: Elaboración propia.

La Figura 8 presenta el diagrama de actividad que detalla el proceso completo de integración de los datos y la automatización en el sistema POSIntegrado de Hortiluma, cubriendo los pasos clave para gestionar la adición de un producto hasta la generación de la factura. El flujo abarca desde la verificación y registro del producto en el inventario, la actualización de los niveles de stock, la ejecución de la venta y, finalmente, la creación de un reporte para la factura electrónica.

Figura 8
Diagramas de actividades principales – POSIntegrado



Nota. Fuente: Elaboración propia.

Como se estableció en el planteamiento del problema, Hortiluma enfrenta dificultades debido a la falta de un sistema que permita una gestión integral y la presentación de información de manera comprensible para los usuarios. Esto incluye la necesidad de contextualizar los datos de ventas, inventarios y reportes generados, de acuerdo con el perfil del usuario que accede a la plataforma. Por lo tanto, la solución propuesta busca superar estas limitaciones mediante la implementación de un sistema POSIntegrado que centraliza y automatiza la gestión de datos, asegurando una experiencia fluida y personalizada.

Se han identificado los requerimientos del usuario y los principales historias de usuario de la plataforma. Basándose en estos, se seleccionaron ocho historias de usuario fundamentales para el desarrollo del prototipo del sistema POSIntegrado. La selección se realizó considerando las necesidades de negocio de Hortiluma, priorizando aquellos que pueden representarse visualmente y que tienen un impacto directo en la operación diaria y en la toma de decisiones estratégicas.

Tabla 38
Propuesta de solución – Fase 3 – Datos identificados para POSIntegrado.

ID	Nombre	Por prototipar	Escenario por prototipar
HU001	Gestión de Productos	X	Creación, actualización y eliminación de productos.
HU002	Gestión de Inventarios	X	Actualización de stock, generación de alertas de inventario.
HU003	Realización de Ventas	X	Registro de ventas y actualización de inventario.
HU004	Generación de Facturas	X	Creación de facturas electrónicas y exportación de PDF.
HU005	Gestión de Reportes	X	Creación y visualización de reportes de ventas e inventarios.
HU006	Gestión de Clientes	X	Registro y actualización de información de clientes.
HU007	Integración con API Externas	X	Sincronización de datos con plataformas externas.
HU008	Migración de Datos		Migración de datos históricos desde sistemas anteriores.

Nota. Fuente: Elaboración propia.

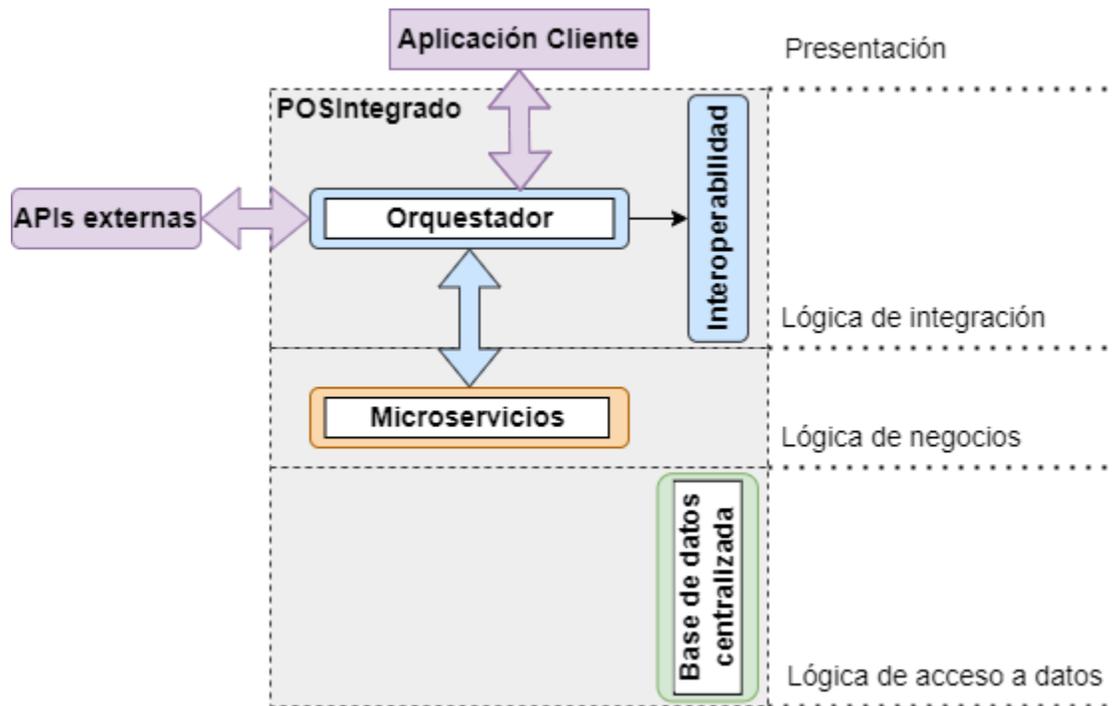
5.1.2.2. Diseño arquitectónico

La arquitectura del sistema POSIntegrado de Hortiluma está diseñada para garantizar la interoperabilidad, escalabilidad y modularidad mediante una estructura basada en microservicios. Cada componente opera de forma autónoma y se comunica a través de un API REST, facilitando una integración sin dependencias rígidas. La Figura 9 muestra esta arquitectura de alto nivel, destacando los componentes clave que soportan las funcionalidades críticas del sistema.

- APIs externas:
 - Las APIs externas permiten que POSIntegrado se comunique con plataformas externas, como sistemas de *e-commerce*, redes sociales y servicios de mensajería.
 - Este componente garantiza la interoperabilidad del sistema con servicios de terceros, posibilitando expansiones futuras sin afectar la operatividad del sistema.
- API REST:
 - Actúa como el punto central de comunicación dentro de POSIntegrado, conectando la aplicación cliente con los microservicios.
 - Facilita la integración tanto interna (entre microservicios) como externa (con APIs externas), permitiendo que el sistema responda a nuevas funcionalidades y requerimientos de manera flexible y segura.
- Aplicación cliente:
 - Proporciona la interfaz de usuario para interactuar con el sistema, facilitando la consulta de inventarios, la generación de reportes y la gestión de ventas.
 - La aplicación cliente se conecta al sistema mediante el API REST, asegurando una comunicación fluida y segura con los diferentes módulos, y ofreciendo una experiencia de usuario intuitiva y mejorada.
- Orquestador:
 - Gestiona el flujo de trabajo entre los diferentes microservicios, asegurando que cada solicitud se dirija al componente correcto.
 - Esta función permite que los microservicios operen de manera independiente mientras se mantienen sincronizados, mejorando el flujo de datos y asegurando que cada microservicio cumpla su rol de forma coordinada.
- Microservicios:
 - Cada microservicio cumple una función específica dentro del sistema, como la gestión de inventarios, facturación y ventas.
 - La modularidad de los microservicios permite que cada uno sea escalable y adaptable, de modo que el sistema en su conjunto pueda responder a cambios en las necesidades del negocio sin afectar su estabilidad.
- Base de datos centralizada:
 - Actúa como el repositorio principal de todos los datos operativos, incluyendo registros de ventas, inventarios y datos históricos.
 - Su integración con los microservicios mediante un API REST asegura un acceso seguro y modularizado a la información, manteniendo la consistencia y disponibilidad de los datos en tiempo real.

La Figura 9 muestra la arquitectura de alto nivel. Esta estructura garantiza que el sistema funcione de manera escalable, manteniendo la consistencia de los datos y adaptándose a las necesidades cambiantes de la empresa.

Figura 9
Diseño de arquitectura de alto nivel



Nota. Fuente: Elaboración propia.

El diseño del sistema POSIntegrado sigue una arquitectura de microservicios, maximizando la flexibilidad y escalabilidad mediante la autonomía y coordinación de cada microservicio. Basado en la referencia de Chal, G. (2007), este enfoque mejora el procesamiento de información y facilita la evolución continua del sistema. Los submódulos de microservicios incluyen:

- **Gestión de Productos:** facilita la creación, actualización y eliminación de productos, manteniendo la información actualizada para la gestión de inventarios y ventas.
- **Gestión de Inventarios:** actualiza en tiempo real los niveles de inventario tras cada venta y genera alertas cuando el stock es bajo, asegurando disponibilidad y planificación.
- **Gestión de Ventas:** coordina y registra las transacciones de venta, ajustando automáticamente los inventarios tras cada operación.
- **Gestión de Clientes:** administra información de clientes, incluyendo datos de contacto e historial de compras, mejorando la personalización en la atención al cliente.
- **Gestión de Reportes:** genera reportes detallados sobre ventas, inventarios y otros datos críticos en formatos gráficos y tabulares, facilitando el análisis estratégico y la toma de decisiones.

La Figura 10 ilustra cada microservicio del sistema POSIntegrado, reforzando una arquitectura robusta que combina la independencia de los microservicios con la capacidad de integración a través de un API REST.

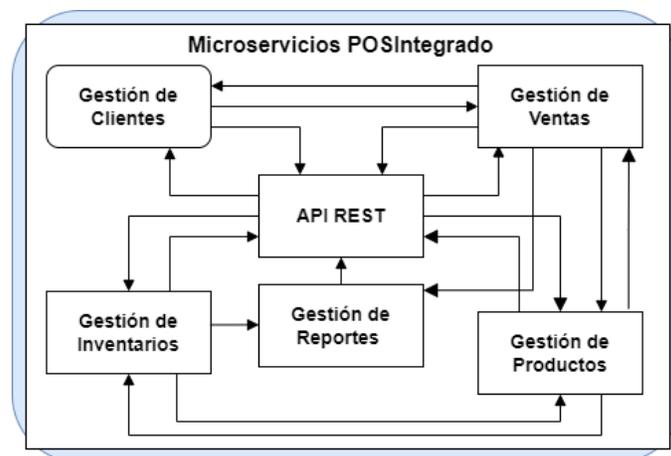
Figura 10
Diseño de arquitectura del módulo de microservicios



Nota. Fuente: Elaboración propia.

La modularidad del diseño permite que cada submódulo de microservicios escale de forma autónoma. Esto significa que las capacidades de cada componente pueden ajustarse según las necesidades del negocio sin afectar la estabilidad del sistema. POSIntegrado es capaz de adaptarse rápidamente a los cambios del entorno de negocios, manteniendo su rendimiento y fiabilidad. En el Módulo Central POSIntegrado, los microservicios actúan como el núcleo de la solución, centralizando y coordinando las principales funcionalidades del sistema de Hortiluma. La Figura 11 muestra la disposición de cada submódulo y su interacción dentro de la arquitectura general.

Figura 11
Diseño de arquitectura del módulo microservicios POSIntegrado



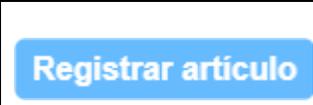
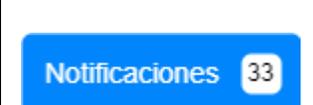
Nota. Fuente: Elaboración propia.

5.1.2.3. Diseño de la interfaz

La interfaz de usuario propuesta para el sistema POSIntegrado de Hortiluma se centra en proporcionar prototipos generales que servirán como base para el diseño detallado en etapas futuras de desarrollo. Estos prototipos están orientados a simplificar las interacciones de los usuarios con el sistema, asegurando que las funcionalidades clave como la gestión de inventarios, la generación de reportes y la sincronización de datos sean accesibles y comprensibles. Aunque el diseño actual es preliminar, sigue principios establecidos de usabilidad para garantizar una experiencia de usuario efectiva.

La Tabla 39 presenta los principales componentes de la interfaz diseñados como prototipos generales para guiar el desarrollo futuro de la aplicación cliente de POSIntegrado. Estos elementos no representan un diseño final, sino un esquema base que puede ser iterado y ajustado en la fase de implementación.

Tabla 39
Componentes

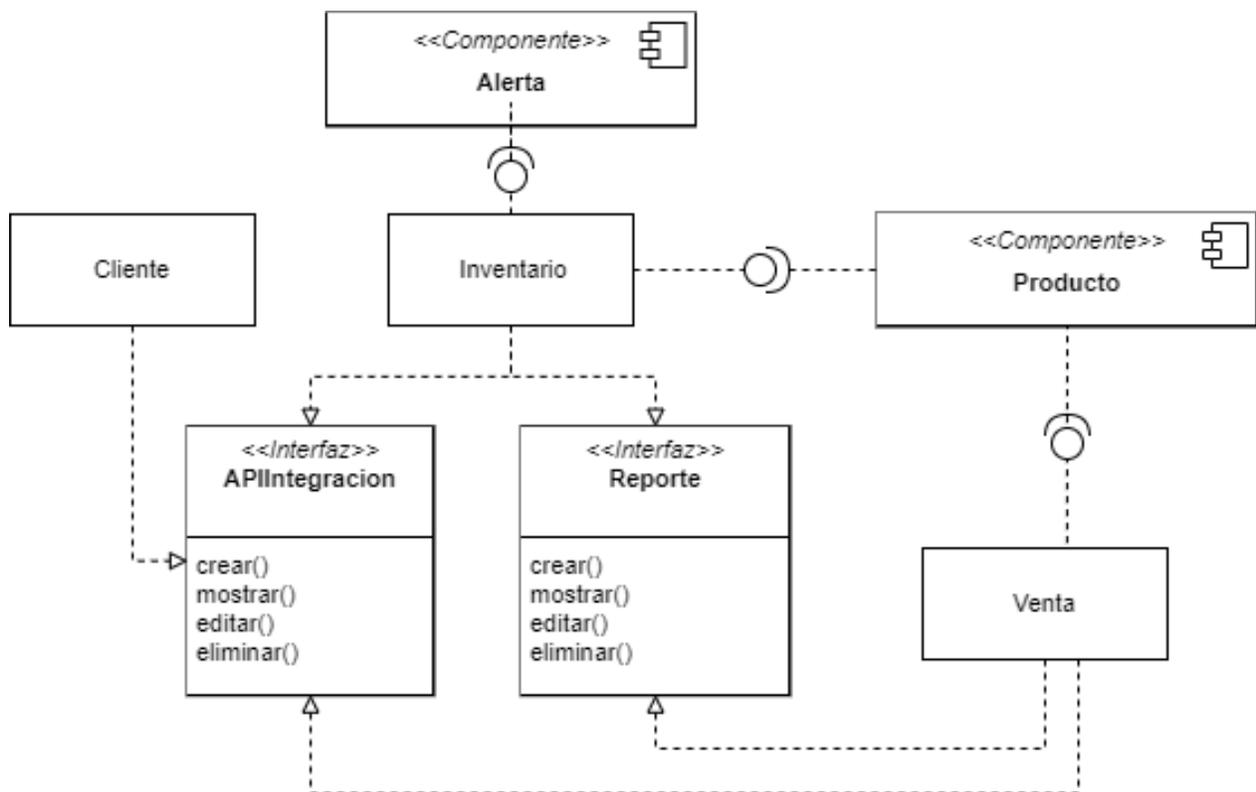
Componente	GUI	Propósito	Argumento de Diseño
Botón primario		Iniciar acciones importantes como generación de facturas, sincronización de inventarios y consultas de reportes.	Según Ruiz Calle (2018), un botón debe ser visualmente destacable, orientado a guiar acciones críticas.
Alertas		Notificar al usuario sobre eventos en tiempo real, como baja de inventarios o errores de sincronización.	De acuerdo con Patternfly, las alertas son ideales para informar sin interrumpir la experiencia de usuario.
Tarjeta		Mostrar resúmenes de información como reportes de ventas o inventarios.	Según Laubheimer (2016), las tarjetas agrupan información relevante para facilitar la consulta rápida.
Producto		Indica los productos del sistema que debe gestionar.	Indica los productos del sistema que debe gestionar.

Nota. Fuente: Elaboración propia.

5.1.2.4. Diseño de componentes

En la Figura 12 se presenta el diseño de componentes del sistema POSIntegrado de Hortiluma, mostrando una estructura modular y detallada que facilita la gestión y la interacción entre los diferentes elementos. Este diseño asegura que cada módulo funcione de manera independiente y escalable, facilitando la integración y el mantenimiento del sistema. A continuación, se detalla cada componente relevante, alineado con los objetivos del proyecto:

Figura 12
Diagrama de componentes



Nota. Fuente: Elaboración propia.

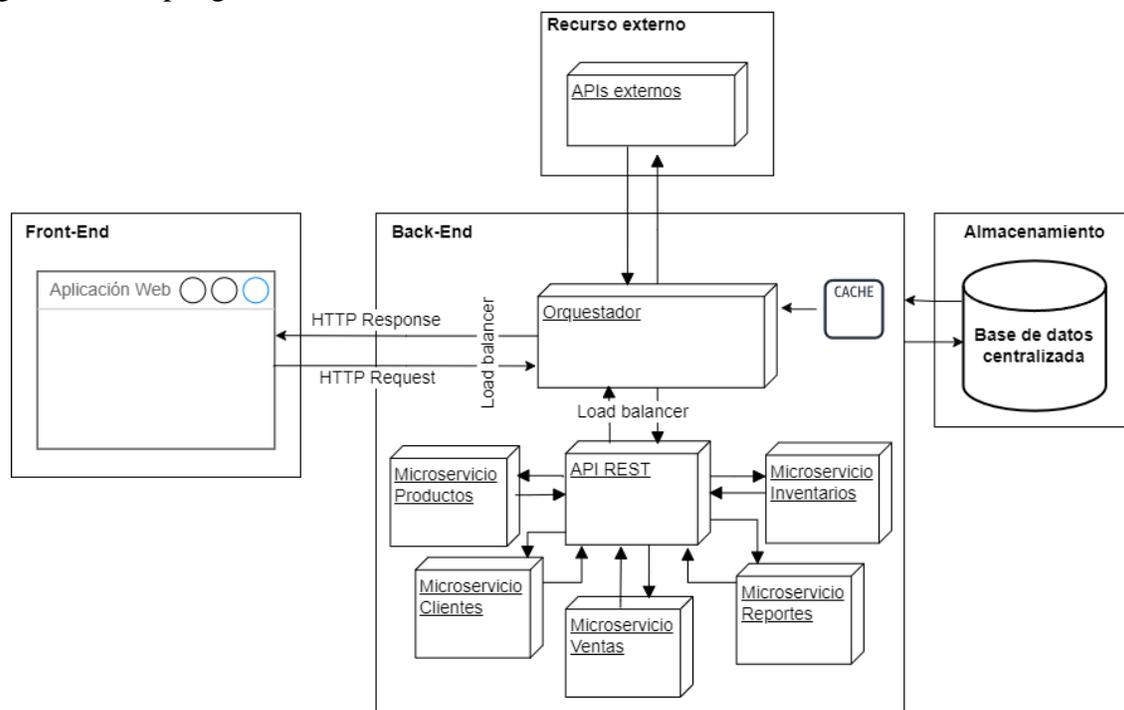
5.1.2.5. Diseño del despliegue

El despliegue del sistema POS Integrado de Hortiluma ha sido diseñado con una arquitectura modular basada en microservicios. Este enfoque permite una organización del *software* que es , escalable y adaptable al entorno de producción. La arquitectura se divide en tres capas principales:

- *Front-End*: esta capa incluye la interfaz de usuario a través de una aplicación web que interactúa con el sistema. Su función es recibir y enviar solicitudes HTTP al *Back-End*, proporcionando una experiencia directa para los usuarios.
- *Back-End*: aquí se alojan los servicios principales del sistema, organizados en varios microservicios especializados (Productos, Clientes, Ventas, Inventarios y Reportes), que están coordinados por un Orquestador. El API REST centraliza las comunicaciones, y un Balanceador de Carga distribuye mente las solicitudes entre los microservicios. Esto asegura un alto rendimiento y una respuesta rápida del sistema.
- Almacenamiento: esta capa gestiona la base de datos centralizada, donde se almacena toda la información relevante. Además, un sistema de cacheo acelera el acceso a los datos, mejorando el rendimiento de las operaciones.

Cada capa cumple un rol específico que contribuye al funcionamiento general del sistema, mejorando tanto la eficiencia operativa como la experiencia del usuario final. La Figura 13 ilustra los componentes clave de este despliegue.

Figura 13
Diagrama de despliegue



Nota. Fuente: Elaboración propia.

5.2. Fase 4: Validación del diseño y cumplimiento de requisitos

La Fase 4 es una etapa crucial dentro del proyecto de rediseño del sistema de Hortiluma, ya que busca asegurar que la nueva arquitectura cumpla con los requisitos técnicos y funcionales previamente establecidos. Se pone un enfoque especial en la interoperabilidad, automatización y eliminación de duplicidades de datos. Esta fase tiene como objetivo principal validar el diseño arquitectónico mediante herramientas que verifiquen el cumplimiento de los requisitos y aseguren que el sistema propuesto funcione correctamente en condiciones reales.

Tabla 40
Operacionalización de las variables – Fase 4

Fase 4: Validación del diseño y cumplimiento de requisitos		Instrumentos
Objetivo específico	Validar el diseño arquitectónico propuesto para la garantía de la interoperabilidad de todas las funcionalidades actuales de Hortiluma, mediante una matriz de cobertura de requerimientos e historias de usuario.	Apéndice E. Plantilla de revisión documental Apéndice I. Plantilla de historias de usuario Apéndice K. Plantilla de matriz de cobertura de requerimientos
Variables de investigación		Sujetos de investigación
Historias de usuario y validación de la arquitectura, cumplimiento de requisitos de interoperabilidad		Administración del negocio, gestión de ventas minoristas, gestión de ventas mayoristas

Nota. Fuente: Elaboración propia.

5.2.1. Revisión documental

La revisión documental ha validado que el diseño del sistema de Hortiluma cumple con los requisitos técnicos y funcionales establecidos en las fases previas del proyecto. Se confirma la implementación correcta de una arquitectura modular basada en microservicios, que asegura escalabilidad y robustez, junto con la integración efectiva de API y *middleware*. La automatización de procesos críticos como la actualización de inventarios y precios también ha sido verificada, asegurando una mayor coherencia de los datos y mejorando los tiempos de procesamiento. Además, el análisis de riesgos ha identificado posibles puntos de fallo y ha propuesto planes de mitigación efectivos. Finalmente, el diseño ha sido validado mediante pruebas reales, lo que garantiza que el sistema está listo para su implementación.

Tabla 41
Fase 4 – Revisión documental

Apéndice	Hallazgo
Apéndice E.16. Plantilla de revisión documental - RD-016	La arquitectura se ha dividido correctamente en microservicios independientes, lo que garantiza la escalabilidad y robustez del sistema. Este diseño modular permite que cada componente funcione de manera independiente, facilitando futuras expansiones sin afectar las operaciones actuales.
Apéndice E.17. Plantilla de revisión documental - RD-017	Las API REST y <i>middleware</i> se han integrado exitosamente con los sistemas existentes (Kyte, FACEL y POS), garantizando una interoperabilidad efectiva entre todos los módulos y plataformas externas. El diseño asegura la correcta comunicación y flujo de datos, eliminando la necesidad de intervención manual.
Apéndice E.18. Plantilla de revisión documental - RD-018	La automatización de los procesos de inventarios y precios ha sido implementada correctamente, funcionando sin intervención manual. Esta implementación mejora la coherencia de los datos, reduciendo errores y tiempos de procesamiento, lo que mejora la eficiencia operativa de Hortiluma.
Apéndice E.19. Plantilla de revisión documental - RD-019	La matriz de cobertura confirma que todos los requisitos funcionales y técnicos han sido cubiertos completamente en el diseño del sistema. Se verifica que no existen brechas en términos de interoperabilidad, automatización y escalabilidad, cumpliendo con los objetivos establecidos.
Apéndice E.20. Plantilla de revisión documental - RD-020	Las pruebas de historias de usuario aseguran el correcto funcionamiento del sistema en condiciones reales. Los escenarios cubren la integración de API, la automatización de procesos y la resiliencia del sistema. Esto garantiza que el sistema responde adecuadamente ante condiciones operativas exigentes.
Apéndice E.21. Plantilla de revisión documental - RD-021	El análisis de riesgos ha identificado posibles puntos de fallo en la interoperabilidad y automatización, además, se han desarrollado planes de mitigación robustos. Estos incluyen medidas preventivas para asegurar que los fallos no afecten la continuidad operativa de la empresa, minimizando el impacto en caso de errores.
Apéndice E.22. Plantilla de revisión documental - RD-022	El diseño final ha sido validado con éxito bajo condiciones operativas reales, cumpliendo con todos los requisitos técnicos y funcionales definidos. El sistema está listo para ser implementado en producción, resolviendo los problemas de duplicidad de datos, falta de automatización y errores en la interoperabilidad detectados previamente.

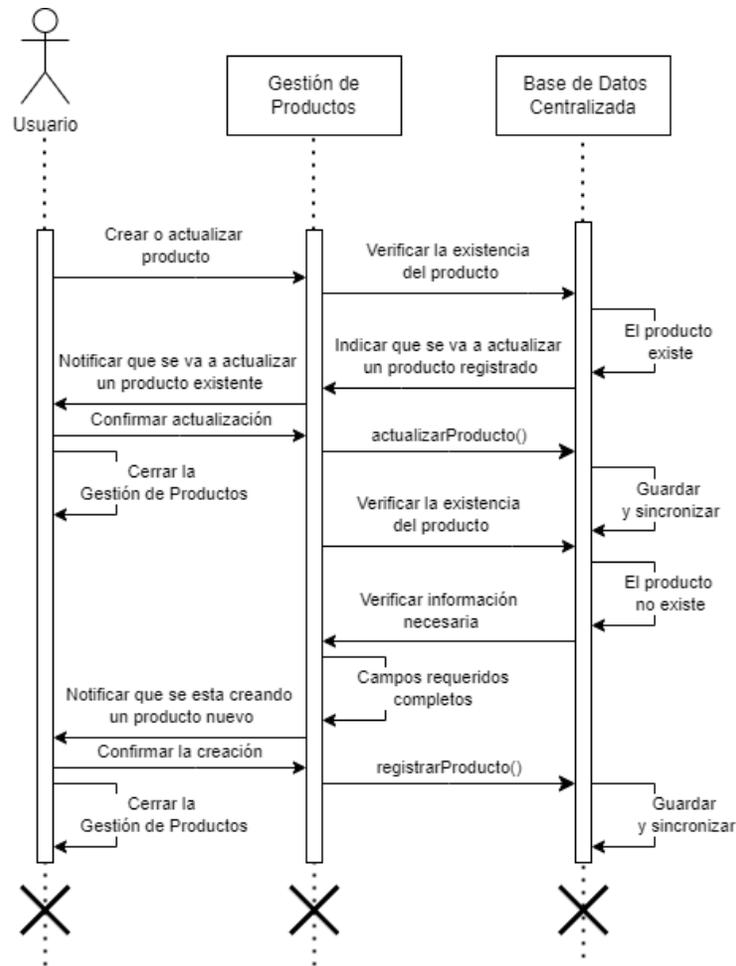
Nota. Fuente: Elaboración propia.

5.2.2. Pruebas de historias de usuario

La validación de los historias de usuario mediante diagramas de secuencia y actividad es esencial para confirmar que el sistema propuesto cumple con los requisitos definidos y opera de manera óptima en situaciones reales. Estos diagramas facilitan la simulación de procesos críticos, como la sincronización de inventarios, la automatización de órdenes de compra, la integración con plataformas externas y la generación de reportes, asegurando que cada funcionalidad se ejecute correctamente dentro del flujo del sistema.

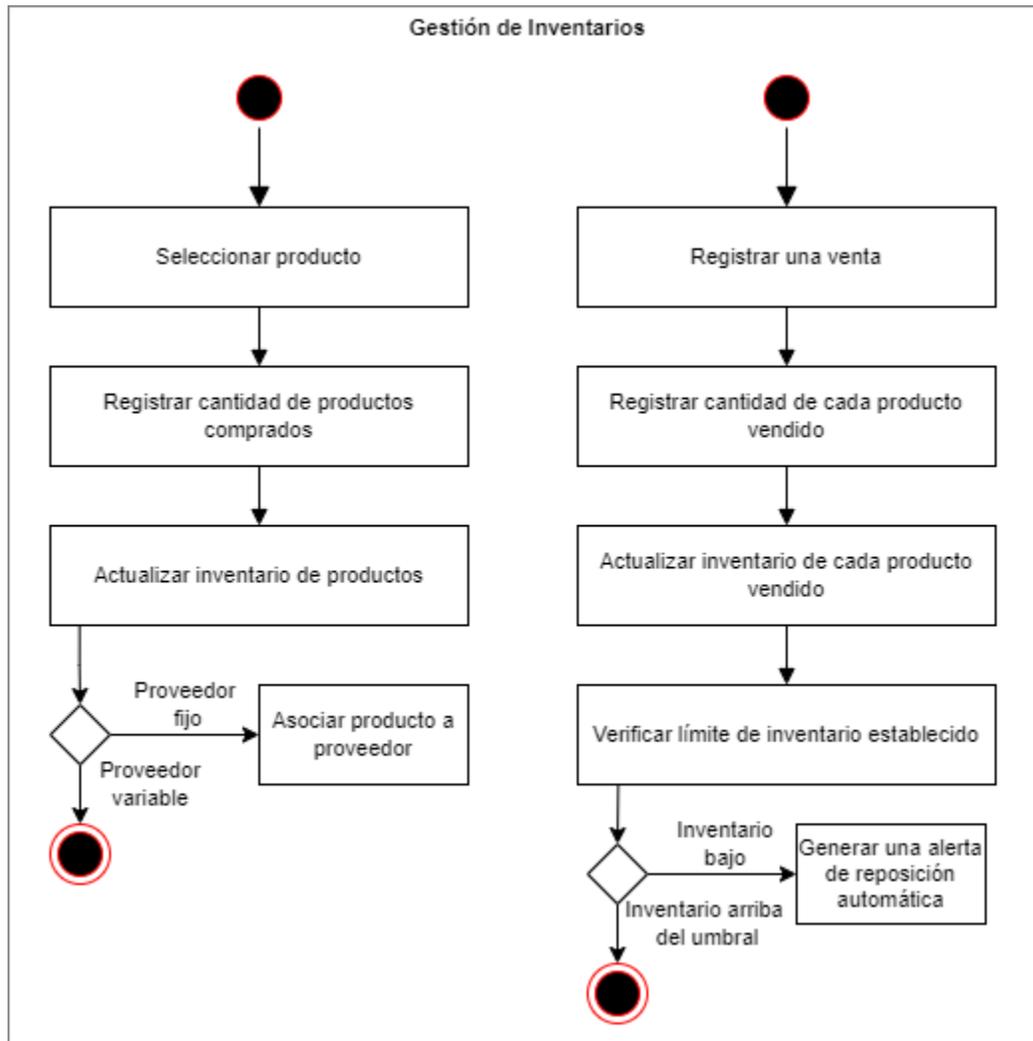
La inclusión de estos diagramas UML permite una representación gráfica precisa de las interacciones y flujos de trabajo entre los diferentes componentes del sistema. Cada diagrama ha sido cuidadosamente seleccionado para ilustrar las interacciones clave, evitando redundancias y resaltando los aspectos más relevantes de cada historia de usuario. Esta metodología no solo proporciona una visión clara de los procesos, sino que también mejora la comprensión y colaboración entre los desarrolladores y otros interesados en el proyecto, garantizando así una validación exhaustiva de la solución propuesta para Hortiluma.

Figura 14
Diagrama de secuencia – Apéndice I.1. Plantilla de historias de usuario - HU-001



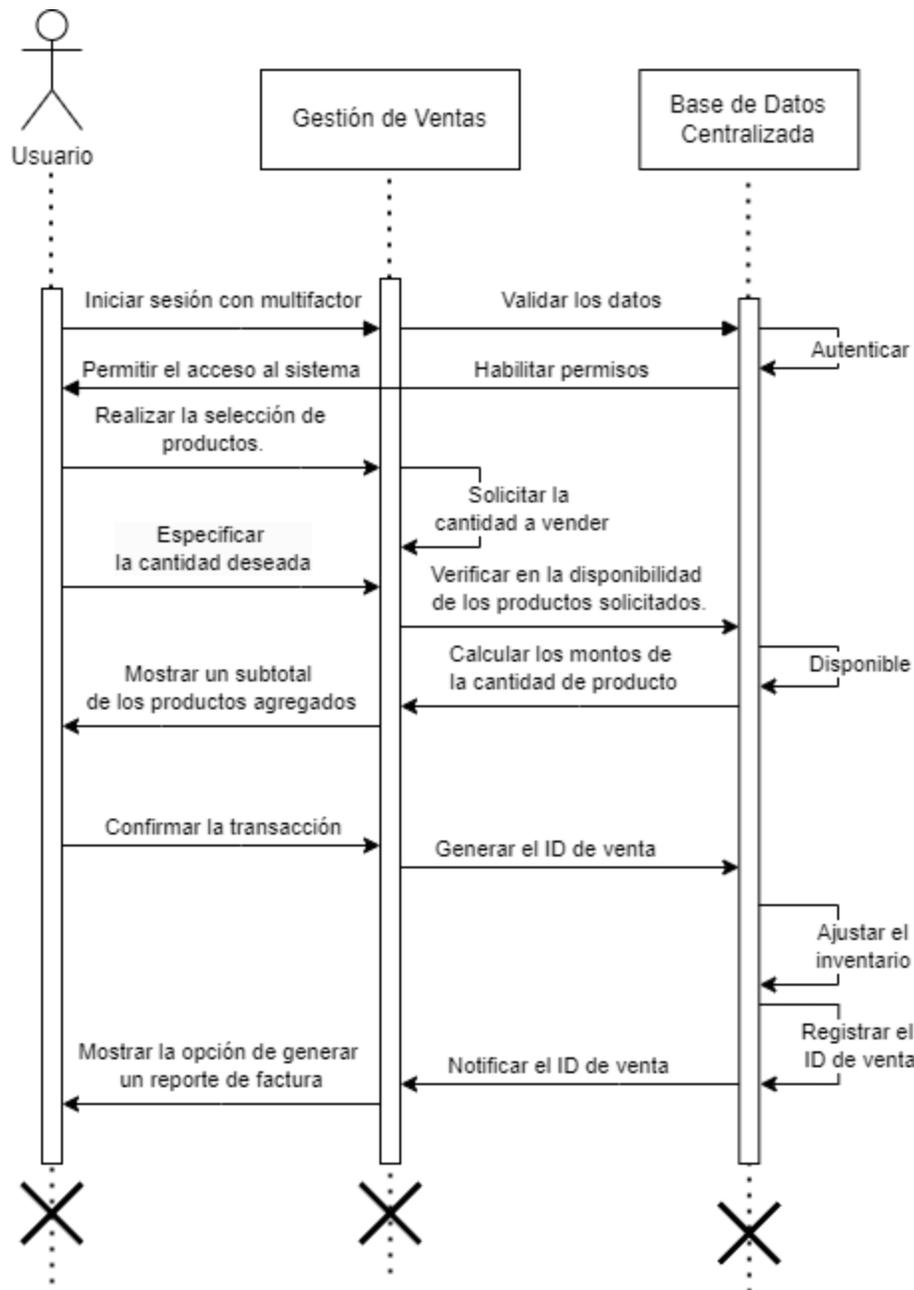
Nota. Fuente: Elaboración propia.

Figura 15
Diagrama de actividad – Apéndice I.2. Plantilla de historias de usuario - HU-002



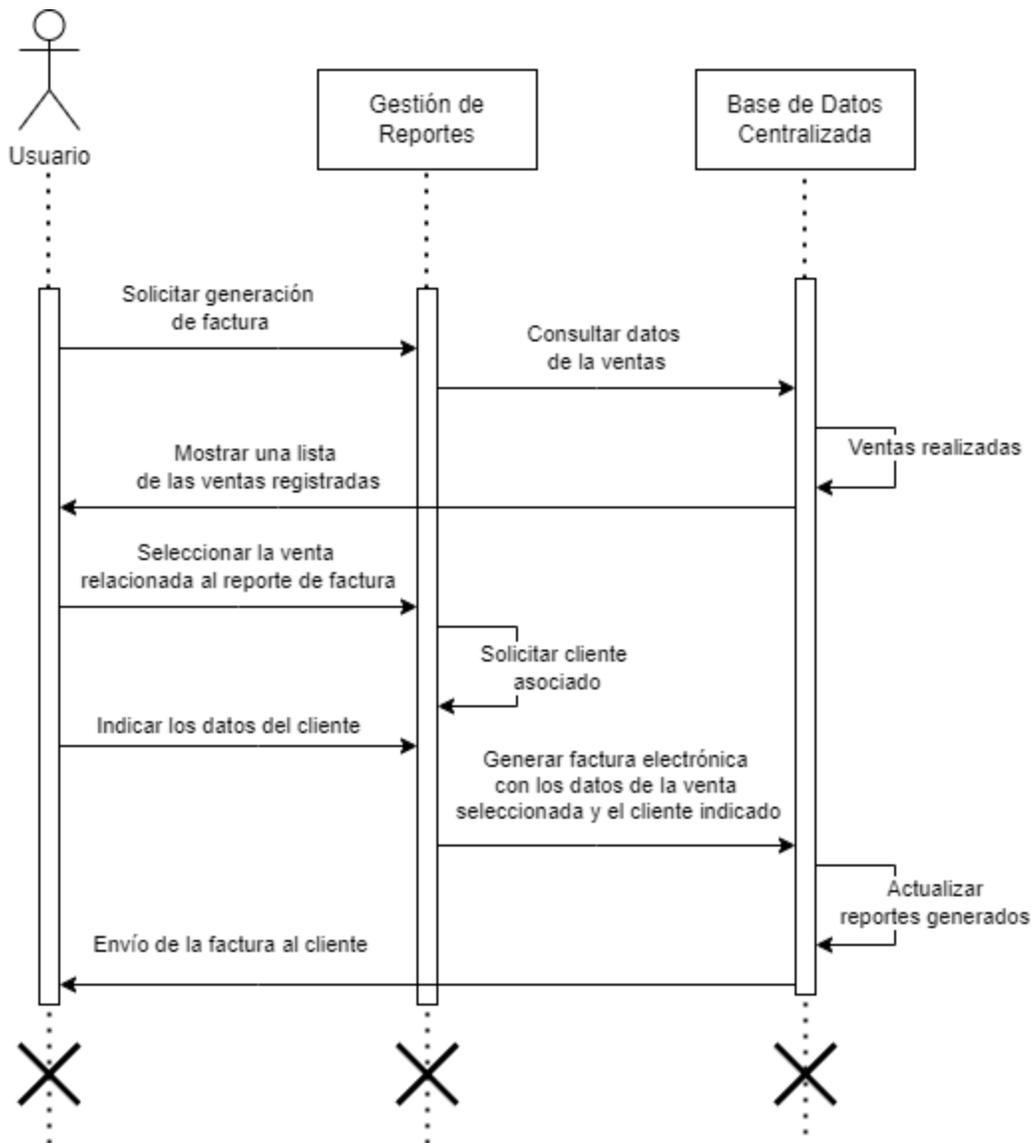
Nota. Fuente: Elaboración propia.

Figura 16
Diagrama de secuencia – Apéndice I.3. Plantilla de historias de usuario - HU-003



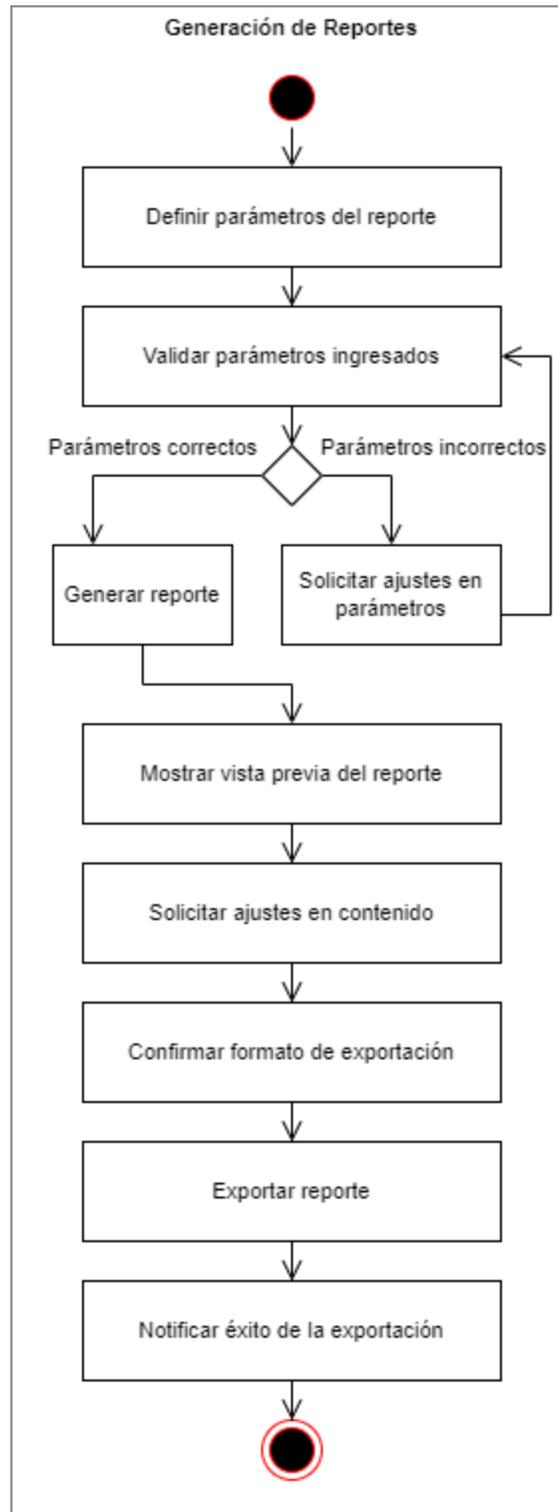
Nota. Fuente: Elaboración propia.

Figura 17
Diagrama de secuencia – Apéndice I.4. Plantilla de historias de usuario - HU-004



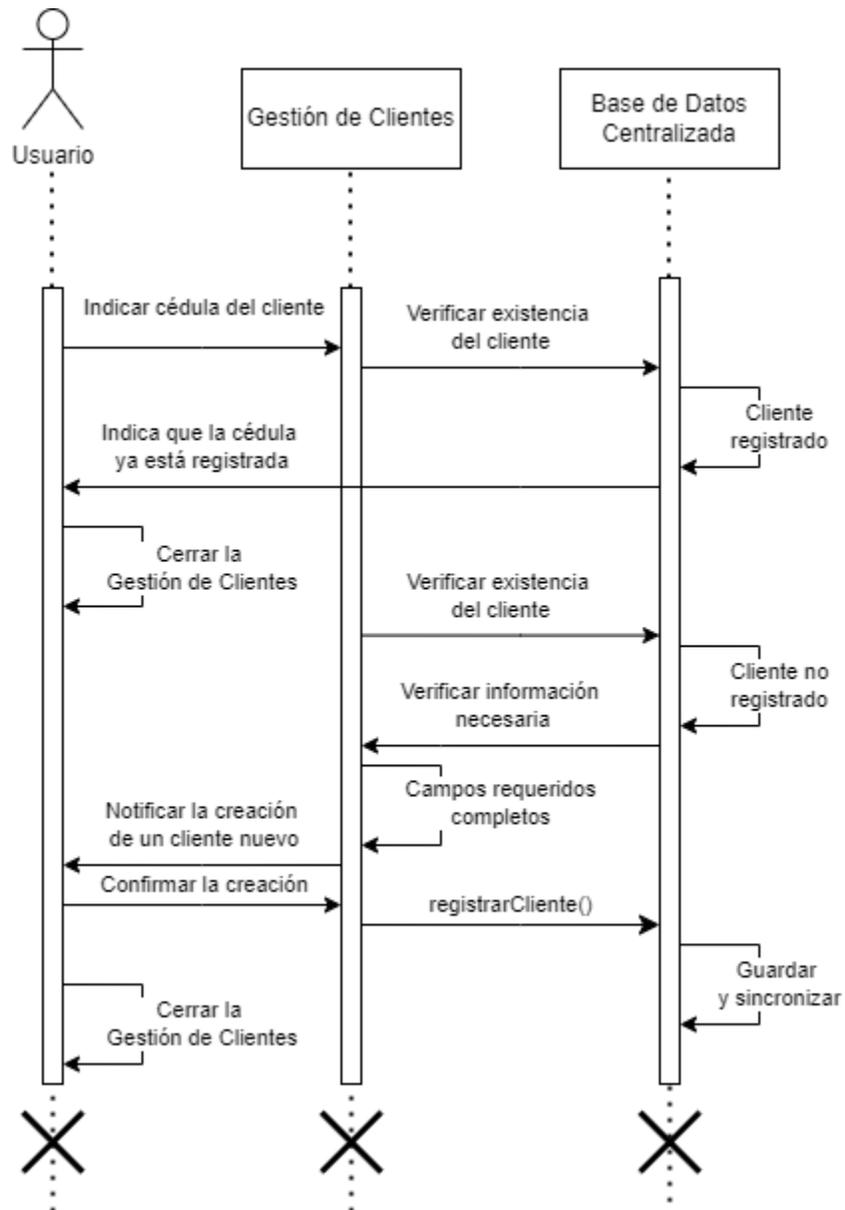
Nota. Fuente: Elaboración propia.

Figura 18
Diagrama de actividad – Apéndice I.5. Plantilla de historias de usuario - HU-005



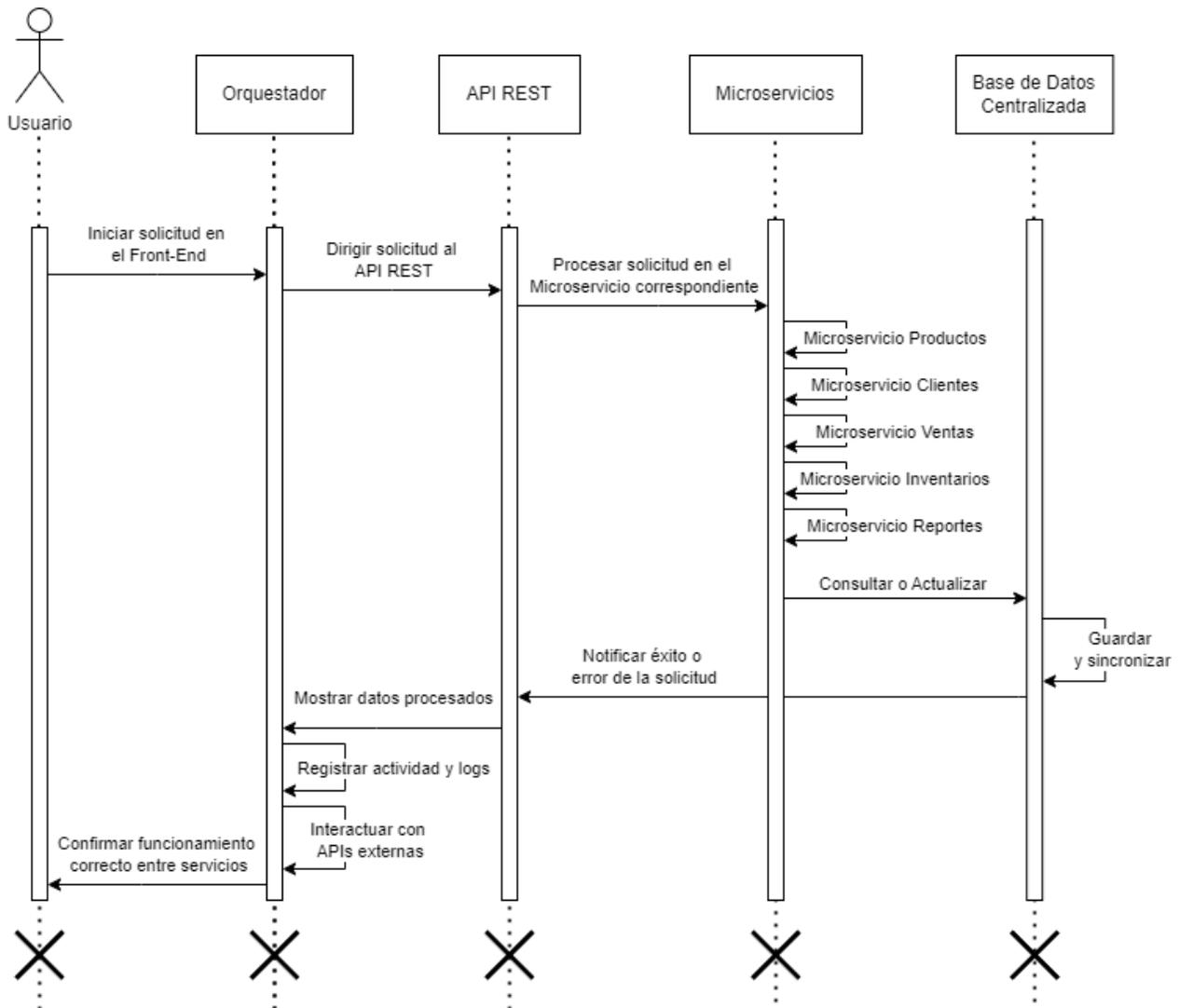
Nota. Fuente: Elaboración propia.

Figura 19
Diagrama de secuencia – Apéndice I.6. Plantilla de historias de usuario - HU-006



Nota. Fuente: Elaboración propia.

Figura 20
Diagrama de secuencia – Apéndice I.7. Plantilla de historias de usuario - HU-007



Nota. Fuente: Elaboración propia.

5.2.3. Matriz de cobertura de requerimientos

La matriz de cobertura de requerimientos es una herramienta esencial para asegurar que todos los requerimientos funcionales (RF) y no funcionales (RNF) han sido abordados y validados a través de los historias de usuario. Este proceso de trazabilidad garantiza que no existan brechas en el diseño del sistema, confirmando que la funcionalidad se ha validado de manera exhaustiva antes de su implementación. Esto es especialmente importante en el contexto de Hortiluma, donde la sincronización de datos y la eficiencia operativa son críticos para el éxito de la solución propuesta.

La matriz, presentada en la Tabla 42 establece una relación directa entre cada historia de usuario probado y los requerimientos que cubre, facilitando la identificación de posibles áreas de mejora y asegurando una cobertura completa de los procesos esenciales. Esta matriz no solo contribuye a la validación del diseño, sino que también permite garantizar que la propuesta de *software* cumple con las necesidades operativas y de negocio identificadas durante el análisis.

Tabla 42
Matriz de cobertura de requerimientos - MDR-001

	HU-001	HU-002	HU-003	HU-004	HU-005	HU-006	HU-007	HU-008
REQF- 1	X	X	X		X	X		
REQF- 2	X					X		
REQF- 3	X	X			X			
REQF- 4		X						
REQF- 5	X	X	X	X	X	X	X	
REQF- 6	X				X			
REQF- 7		X						
REQF- 8		X	X					
REQF- 9			X					
REQF- 10				X			X	
REQF- 11				X		X		
REQF- 12								X
REQF- 13			X		X			
REQF- 14		X			X			X
REQNF- 1	X	X	X					
REQNF- 2				X				
REQNF- 3				X				
REQNF- 4		X	X					X
REQNF- 5					X			X
REQNF- 6	X			X		X		
REQNF- 7		X						
REQNF- 8				X			X	
REQNF- 9	X		X			X	X	
REQNF- 10		X			X			X

Nota. Fuente: Elaboración propia.

5.2.4. Resultados de la validación

La validación del diseño del sistema de Hortiluma ha sido esencial para confirmar que la solución propuesta cumple con los requisitos técnicos y operativos, abordando de manera efectiva los desafíos identificados. A continuación, se presentan los principales hallazgos derivados de la validación a través de pruebas de historias de usuario, la matriz de cobertura de requerimientos y la revisión documental:

1. Las pruebas de historias de usuario y la matriz de cobertura confirman que el sistema satisface plenamente los requisitos funcionales y no funcionales establecidos. Cada funcionalidad crítica, desde la sincronización de datos hasta la automatización de procesos y la integración de plataformas, ha sido implementada y verificada con éxito. Esto garantiza que no existen brechas en el diseño, lo que asegura que el sistema se adapta perfectamente a las necesidades operativas de Hortiluma.
2. Durante la fase de validación, los usuarios finales realizaron pruebas en escenarios reales, lo que permitió confirmar que el sistema responde eficazmente a sus expectativas y mejora de manera significativa los procesos operativos. Esto se traduce en una aceptación positiva del sistema por parte de los usuarios, asegurando que la solución cumple con los objetivos definidos en las fases iniciales del proyecto.
3. La automatización de procesos clave ha generado un impacto positivo en la eficiencia operativa. Las áreas más beneficiadas incluyen:
 - a. El sistema actualiza en tiempo real los niveles de stock y las transacciones de ventas, lo que elimina la necesidad de intervención manual y minimiza los errores asociados a la gestión de inventarios.
 - b. La automatización de la creación de órdenes de compra, según los niveles de stock y las proyecciones de demanda, ha permitido una gestión precisa del inventario, evitando tanto desabastecimientos como acumulación innecesaria de productos.
4. El sistema ha eliminado la duplicidad de tareas en la gestión de inventarios y facturación. Esto ha reducido el tiempo dedicado a tareas repetitivas, permitiendo al equipo centrarse en actividades estratégicas que generan mayor valor para la empresa, como la planificación de ventas y la mejora continua de los procesos.
5. Se ha logrado una integración entre las distintas plataformas de Hortiluma, incluyendo el sistema POS y otras herramientas de venta. Esto ha permitido una sincronización fluida de inventarios, precios y transacciones en tiempo real, tanto en plataformas físicas como digitales, mejorando la coherencia de los datos y facilitando la toma de decisiones operativas.
6. El sistema incorpora un sistema de alertas que se activa ante inconsistencias en la sincronización de datos y cuando los niveles de inventario alcanzan umbrales críticos. Este mecanismo ha demostrado su eficacia al permitir una respuesta rápida y proactiva ante posibles problemas, evitando interrupciones en las operaciones diarias y asegurando la continuidad del negocio.
7. Gracias a la integración y la automatización de procesos, el sistema ha mejorado la estabilidad operativa de Hortiluma. La capacidad de detectar y corregir problemas en tiempo real, junto con la reducción de errores humanos, asegura que las operaciones se mantengan estables y confiables, incluso bajo condiciones de alta demanda.

5.3. Estudio de costos

Este apartado proporciona un análisis detallado de los costos asociados con el proyecto de rediseño del sistema de Hortiluma. Los costos abarcan tanto la implementación del nuevo sistema como los beneficios financieros esperados a partir de la transición de los sistemas actuales (Kyte, FACEL y POS) hacia una solución tecnológica más que mejore las operaciones y mejore la competitividad de la empresa.

5.3.1. Inversión inicial

La inversión inicial abarca los recursos necesarios para el desarrollo del proyecto a lo largo de un periodo de 16 semanas laborables (cuatro meses), considerando una jornada completa de 40 horas semanales. Durante este tiempo se desarrollarán las etapas fundamentales del proyecto, desde la planificación inicial hasta la migración de datos y la entrega del diseño final. La Tabla 43 presenta un análisis detallado del costo asociado al estudiante responsable del desarrollo del proyecto como parte del TFG. Este cálculo se basa en el salario mínimo mensual estimado para profesionales licenciados en Costa Rica.

Tabla 43
Análisis de costos por recurso humano

Puesto	Salario Mensual	Meses Laborales	Costo Total (₡)
Estudiante a cargo de TFG	₡411,623.76	4	₡1,646,495.04
Total Salarios			₡1,646,495.04

Nota. Fuente: Elaboración propia.

Este costo corresponde exclusivamente al recurso humano encargado del TFG y refleja la inversión mínima inicial necesaria para el diseño del sistema propuesto.

5.3.2. Costos de la futura implementación

La futura implementación del sistema requerirá de un equipo de trabajo especializado para ejecutar actividades técnicas específicas, siguiendo un cronograma detallado de cuatro meses. Las tareas incluyen el desarrollo, integración y puesta en producción de los módulos diseñados, así como pruebas de funcionalidad y calidad. A continuación, se describe el cronograma de trabajo y los costos asociados.

El cronograma estimado está estructurado en 16 semanas, distribuidas en fases que abordan desde la planificación y configuración inicial hasta la migración de datos y el despliegue en producción.

Tabla 44
Cronograma estimado de implementación

Recurso Humano	Mes	Enero				Febrero				Marzo				Abril			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Enero	Semana / Actividad																
Administrador de Proyectos (AP)	Planificación, configuración del ambiente de desarrollo, definición de estándares de código y metodología de trabajo.	█	█	█	█												
Desarrollador de Software (DS)	Desarrollo de API REST, servicios principales, sistema de notificaciones, autenticación y pruebas funcionales.					█	█	█	█								
AP y DS	Desarrollo de interfaces, dashboards, vistas, y componentes clave.							█	█	█							
AP y DS	Integración frontend-backend, pruebas de funcionalidad, rendimiento, y usuario.									█	█	█	█				
AP y DS	Migración de datos, despliegue en producción, capacitación, soporte post-implementación, ajustes.													█	█	█	█

Nota. Fuente: Elaboración propia.

El equipo necesario para la futura implementación incluye un administrador de proyectos (AP) y dos desarrolladores de *software* (DS). Los salarios están calculados en función de los mínimos establecidos por el Ministerio de Trabajo y Seguridad Social de Costa Rica en 2024.

Tabla 45
Costos por recurso humano

Puesto	Salario Mensual (₡)	Meses Laborales	Costo Total (₡)
Administrador de Proyectos	₡638,299.51	4	₡2,553,198.04
Desarrollador de Software 1	₡765,985.67	2	₡1,531,971.34
Desarrollador de Software 2	₡765,985.67	2	₡1,531,971.34
Total Salarios			₡5,617,140.72

Nota. Fuente: Elaboración propia.

Los salarios han sido ajustados considerando los requisitos laborales y el periodo de trabajo estimado. Esto asegura que los costos reflejen con precisión la inversión necesaria para los recursos humanos involucrados.

5.3.3. Tecnologías

La implementación del sistema para Hortiluma se llevará a cabo utilizando tecnologías gratuitas y de código abierto, seleccionadas cuidadosamente para garantizar funcionalidad, escalabilidad y bajo costo inicial. Estas tecnologías permitirán a Hortiluma operar de manera eficiente en las fases iniciales del proyecto, con la flexibilidad para migrar a soluciones de pago según las necesidades futuras. A continuación, se detallan las tecnologías óptimas para cada área del desarrollo:

1. Infraestructura en la nube

a. Tecnología:

- i. AWS Free Tier: Incluye cómputo (EC2), almacenamiento (S3) y bases de datos (RDS) gratuitos durante 12 meses, sujeto a límites de uso.
- ii. Google Cloud Platform Free Tier: Ofrece almacenamiento, BigQuery y Firestore gratuitos en entornos de prueba.
- iii. Microsoft Azure Free Account: Servicios gratuitos de bases de datos, almacenamiento y aplicaciones durante 12 meses.

b. Características:

- i. Escalabilidad para manejar volúmenes crecientes de datos y procesamiento.
- ii. Opciones gratuitas ideales para entornos de desarrollo y pruebas.

c. Costo:

- i. Gratuito dentro de los límites definidos por cada proveedor.
- ii. Costos potenciales si se exceden los límites de almacenamiento o procesamiento.

d. Consideraciones:

- i. Supervisar el consumo para evitar sobrecargos.
- ii. Migrar a planes de pago conforme crezcan las necesidades de la empresa.

2. Desarrollo de *middleware* y API

a. Tecnología:

- i. Node.js: Plataforma eficiente para la ejecución de JavaScript en el servidor, ideal para sistemas escalables.
- ii. Express.js: Framework ligero para construir APIs RESTful rápidamente.
- iii. Bases de datos:

1. PostgreSQL: Base de datos relacional gratuita, escalable y de alto rendimiento.
2. MongoDB Community Edition: Solución NoSQL gratuita, eficiente para datos no estructurados.

b. Características:

- i. Escalabilidad para gestionar la integración de módulos clave (inventarios, facturación y ventas).
- ii. Gran soporte comunitario y compatibilidad con tecnologías modernas.

c. Costo:

- i. Gratuito en sus versiones de código abierto.

d. Consideraciones:

- i. Funcionalidades avanzadas como MongoDB Atlas (gestión en la nube) podrían generar costos.

3. Rediseño del sistema POS
 - a. Tecnología:
 - i. React.js: Biblioteca para construir interfaces de usuario interactivas y rápidas.
 - ii. Electron.js: Plataforma para desarrollar aplicaciones de escritorio multiplataforma utilizando tecnologías web.
 - iii. Bootstrap o Material-UI: Frameworks para diseño profesional y adaptable.
 - b. Características:
 - i. Interfaces intuitivas, reutilizables y personalizables para puntos de venta.
 - ii. Compatibilidad con aplicaciones web y de escritorio.
 - iii. Modularidad para agregar funcionalidades en el futuro.
 - c. Costo:
 - i. Gratuito en sus versiones estándar de código abierto.
 - d. Consideraciones:
 - i. Material-UI ofrece una versión "Pro" con componentes adicionales, pero no es necesaria para el desarrollo inicial.

4. Seguridad del sistema:
 - a. Tecnología:
 - i. OAuth 2.0: Protocolo estándar para autorización y autenticación.
 - b. Características:
 - i. Autenticación multifactor y control de acceso basado en roles.
 - ii. Gestión segura de sesiones y usuarios.
 - c. Costo:
 - i. Gratuito en sus versiones de código abierto.

5.3.4. ROI

El Retorno de Inversión (ROI) es una métrica financiera utilizada para medir la rentabilidad de una inversión. Se expresa como un porcentaje que representa el beneficio obtenido en relación con el costo total de la inversión. El ROI se utiliza para evaluar la eficiencia de una inversión y determinar si los beneficios compensan los costos.

$$ROI\% = \frac{(Ganancia\ o\ beneficio\ esperado - Costo\ de\ la\ inversión)}{Costo\ de\ la\ inversión} \times 100$$

Para el cálculo del ROI de la propuesta del sistema interoperable de Hortiluma, se toman en cuenta únicamente los costos de inversión del proyecto y los ahorros generados por la eliminación de los costos asociados a los sistemas Kyte, FACEL y POS. Adicionalmente, se consideran ahorros cuantificables derivados de la mejora en la eficiencia operativa.

5.3.5. Beneficios esperados

Los beneficios calculados incluyen ahorros directos en costos de los sistemas actuales y ahorros derivados de la mejora de procesos operativos. La Tabla 46 detalla los ahorros directos en los costos de los sistemas actuales:

Tabla 46
Costos actuales de Hortiluma

Sistema	Plan	Costo Mensual	Costo Anual
FACEL	PYME 1	€15,000	€180,000
Kyte	Grow	€10,000	€120,000
POS	Anual	€12,500	€150,000
Costo Anual Total			€450,000

Nota. Fuente: Elaboración propia.

Este ahorro refleja el beneficio directo en términos de reducción de gastos operativos relacionados con los sistemas actuales. A medida que se implementa y mejora el sistema propuesto, se espera que otros beneficios económicos indirectos fortalezcan aún más el retorno financiero.

Actualmente, se estima que las tareas manuales relacionadas con la gestión de inventarios, facturación y ventas consumen aproximadamente 10 horas semanales del personal administrativo. Con el nuevo sistema automatizado, estas tareas pueden reducirse en al menos un 70%, liberando aproximadamente 7 horas por semana.

- Horas ahorradas por semana: 7
- Semanas laborales anuales: 52
- Horas ahorradas anuales: $7 \times 52 = 364$ horas
- Costo promedio por hora administrativa: €3,500
- Ahorro anual: $364 \times €3,500 = €1,274,000$

Con un sistema más ágil y automatizado, el personal de ventas podría procesar solicitudes y transacciones un 30% más rápido, permitiendo atender a más clientes o reducir el tiempo necesario para cada transacción. Si el personal de ventas actualmente invierte 20 horas semanales en estas tareas, el tiempo ahorrado sería:

- Horas ahorradas por semana: $20 \times 30\% = 6$
- Semanas laborales anuales: 52
- Horas ahorradas anuales: $6 \times 52 = 312$ horas
- Costo promedio por hora de un vendedor: €4,000
- Ahorro anual: $312 \times €4,000 = €1,248,000$

Los errores en facturación e inventarios generan retrabajo y pérdida de tiempo en el personal administrativo y de ventas. Se estima que el retrabajo consume aproximadamente 5 horas semanales, con un costo de €3,500 por hora. Con el nuevo sistema, los errores se reducirán en al menos un 80%, ahorrando 4 horas por semana. Cálculo del ahorro anual por reducción de errores:

- Horas ahorradas por semana: 4
- Semanas laborales anuales: 52
- Horas ahorradas anuales: $4 \times 52 = 208$ horas
- Costo promedio por hora de correcciones: €3,500
- Ahorro anual: $208 \times €3,500 = €728,000$

5.3.6. Cálculo del ROI de la propuesta de diseño de software de alto nivel

De acuerdo con las secciones beneficios esperados, inversión inicial y costos de la futura implementación, se obtienen las dos variables necesarias para evaluar la rentabilidad de la propuesta:

- Beneficio esperado: ₡450,000 (Ahorro directo) + ₡1,274,000 (Mejora administrativa) + ₡1,248,000 (Productividad en ventas) + ₡728,000 (Reducción de errores) = ₡3,700,000
- Coste de inversión: ₡1,646,495 (Inicial) + ₡5,617,140 (Implementación) = ₡7,263,635

La fórmula utilizada para calcular el ROI es la siguiente:

$$\begin{aligned} ROI\% &= \frac{(\text{Beneficio esperado} - \text{Costo de la inversión})}{\text{Costo de la inversión}} \times 100 \\ ROI\% &= \frac{(3,700,000 - 7,263,635)}{7,263,635} \times 100 \\ ROI\% &= \frac{(3,563,635)}{7,263,635} \times 100 \\ ROI\% &= -49.06\% \end{aligned}$$

El ROI anual calculado es del -49.06%, indicando que los beneficios anuales esperados no superan el costo total de inversión en el primer año.

5.3.7. Plazo de recuperación de la inversión

El plazo de recuperación se calcula dividiendo el costo total de inversión entre los beneficios anuales esperados:

- Beneficios esperados: ₡450,000 (Ahorro directo) + ₡1,274,000 (Mejora administrativa) + ₡1,248,000 (Productividad en ventas) + ₡728,000 (Reducción de errores) = ₡3,700,000
- Coste de inversión: ₡1,646,495 (Inversión inicial) + ₡5,617,140 (Costos de la futura implementación) = ₡7,263,635

La fórmula para calcular el periodo de recuperación es:

$$\begin{aligned} \text{Periodo de recuperación} &= \frac{\text{Coste de inversión}}{\text{Beneficio esperado}} \\ \text{Periodo de recuperación} &= \frac{7,263,635}{3,700,000} \\ \text{Periodo de recuperación} &= 1.96 \text{ años} \end{aligned}$$

El plazo de recuperación estimado es de 1.96 años, tras lo cual el sistema comenzará a generar beneficios netos positivos.

5.3.8. Conclusión del análisis financiero

El análisis financiero del proyecto de rediseño del sistema interoperable de Hortiluma ha identificado una inversión inicial significativa, que no se recupera totalmente en el primer año, según los beneficios proyectados:

- Coste de inversión: ₡7,263,635
- Beneficios esperados: ₡3,700,000
- ROI: -49.06%
- Plazo de recuperación de la inversión: 1.96 años

Sin embargo, el análisis evidencia que:

- El plazo estimado de 1.96 años es adecuado para proyectos tecnológicos de esta magnitud, considerando los beneficios tangibles e intangibles asociados. Este periodo es consistente con el objetivo de transformar las operaciones y mejorar los procesos empresariales de Hortiluma.
- El cálculo del ROI incluye beneficios económicos indirectos. La mejora en la eficiencia operativa, la reducción de errores, y la mejora de procesos administrativos y comerciales resultarán en un impacto positivo a nivel financiero.
- Con un mantenimiento adecuado, la vida útil del sistema puede superar el horizonte de evaluación inicial, prolongando los ahorros operativos e incrementando la rentabilidad general. Este impacto se traduce en:
 - Reducción sostenida de costos operativos.
 - Mayor control sobre procesos administrativos y comerciales.
 - Mejora de la sostenibilidad y competitividad en un mercado dinámico.

Aunque el ROI inicial es negativo, el análisis evidencia que la propuesta es financieramente sólida en el mediano plazo. Con un plazo de recuperación de menos de dos años y beneficios sostenidos a largo plazo, Hortiluma está posicionada para mejorar sus operaciones, su competitividad y fortalecer su capacidad de crecimiento.

Esta inversión no solo moderniza la estructura tecnológica de la empresa, sino que también establece una base sólida para futuras estrategias de innovación y eficiencia operativa.

6. Conclusiones

En este capítulo, se exponen las conclusiones obtenidas a partir del análisis de resultados y la implementación de la propuesta de solución para el diseño de un *software* de alto nivel, orientado a mejorar la interoperabilidad de las funcionalidades actuales de los sistemas de Hortiluma. Las conclusiones presentadas responden a los objetivos específicos del proyecto y están fundamentadas en los datos y evidencias recopiladas durante el proceso de investigación. Cada conclusión refleja el grado de cumplimiento de los objetivos planteados y proporciona una visión clara sobre el impacto de la propuesta en la eficiencia operativa y en la capacidad de integración de los sistemas de la empresa.

6.1. Objetivo específico 1

- La duplicación de tareas entre los sistemas Kyte, FACEL y POS legado representa una carga significativa para el personal administrativo y de ventas, generando ineficiencias operativas y aumentando el riesgo de errores en la gestión de inventarios, facturación y ventas.
- El personal de ventas se beneficiará de una mejora del 30% en la gestión de solicitudes y transacciones, lo que equivale a un ahorro de 312 horas anuales. Esto mejorará la capacidad de atención al cliente y la rapidez en las transacciones.
- La reducción del 80% en los errores de facturación e inventarios disminuirá el tiempo dedicado al retrabajo en 208 horas anuales, mejorando la confiabilidad de los procesos y reduciendo las inconsistencias en la información.
- En conjunto, estas mejoras generarán un ahorro total de 884 horas anuales, distribuidas entre administración, ventas y corrección de errores, reflejando un impacto positivo en la eficiencia operativa y en la capacidad de respuesta de Hortiluma.
- La eliminación de tareas duplicadas y la integración de los sistemas actuales son pasos fundamentales para modernizar las operaciones, mejorar los procesos internos y fortalecer la competitividad de la empresa en el mercado.

6.2. Objetivo específico 2

- La propuesta de un diseño de *software* modular, utilizando *middleware* y API modernas, es esencial para lograr una integración entre los sistemas Kyte, FACEL y POS, superando las limitaciones actuales de interoperabilidad. Este enfoque no solo permite una sincronización en tiempo real, sino que también elimina la duplicación de tareas, reduciendo la inconsistencia de datos y errores en los reportes. La integración tecnológica propuesta responde directamente a la necesidad de superar la deuda técnica acumulada, permitiendo que la infraestructura de Hortiluma crezca de manera flexible y se adapte a futuros cambios.
- La consolidación de una integración de sistemas y verificación de interoperabilidad de entrada de datos permitirá la sincronización automática de inventarios, ventas y facturación, lo cual reducirá el retrabajo y los errores manuales, mejorando la coherencia de los datos en un 35%. La incorporación de tecnologías de automatización como Automatización Robótica de Procesos (RPA) se traduce en una disminución de la intervención manual en tareas repetitivas, aumentando la precisión de los procesos operativos. Esta simplificación es fundamental para agilizar los tiempos de respuesta y mejorar la experiencia del usuario en la gestión de datos.
- La Fase 2: Definición del estado meta prioriza la sincronización automática entre todos los componentes del sistema, garantizando que la información fluya de manera continua y precisa entre ventas, inventarios y facturación. Esto permitirá a Hortiluma tomar decisiones más informadas y reducir las discrepancias de datos que antes afectaban la operación. La sincronización en tiempo real entre plataformas asegura que los datos estén siempre actualizados, lo que se traduce en una mejor planificación de inventarios y respuesta rápida a las demandas del mercado.
- El diseño de *software* no solo simplifica procesos, sino que también reduce los costos operativos relacionados con la corrección de errores manuales y la verificación de la coherencia de datos entre sistemas. Un sistema bien integrado permitirá a Hortiluma reducir la carga operativa, destinando menos tiempo a tareas de corrección y permitiendo al personal enfocarse en actividades estratégicas. Esto resulta en una operación más competitiva, como se describe en el análisis financiero proyectado en la sección necesidades del sistema para el diseño de *software* de alto nivel.

6.3. Objetivo específico 3

- El diseño de componentes basado en una arquitectura modular, ha mejorado la integración y sincronización de datos en tiempo real entre ventas, inventarios y facturación. Esto ha reducido la duplicación de tareas y mejorado la eficiencia operativa, como se establece en el Apéndice I.1. Plantilla de historias de usuario - HU-001.
- La generación de facturas electrónicas basada en el ID de venta y la actualización automática de inventarios han reducido los errores manuales y el tiempo dedicado a tareas repetitivas. Este avance asegura el cumplimiento de las normativas fiscales locales y refuerza la precisión en la toma de decisiones estratégicas, como se detalla en el Apéndice I.3. Plantilla de historias de usuario - HU-003, Apéndice I.4. Plantilla de historias de usuario - HU-004 y Apéndice I.5. Plantilla de historias de usuario - HU-005.
- La integración de la nueva arquitectura con plataformas externas mediante API REST garantiza una interoperabilidad y escalabilidad adaptada a las futuras necesidades de Hortiluma. Este diseño, respaldado por la historia de usuario Apéndice I.7. Plantilla de historias de usuario - HU-007, facilita la comunicación fluida con otras plataformas de *e-commerce* y sistemas contables, permitiendo a la empresa adaptarse rápidamente a nuevas oportunidades de mercado sin necesidad de realizar modificaciones complejas en la infraestructura del sistema.
- La implementación de un sistema de monitoreo para la validación de la integración de sistemas y verificación de interoperabilidad es fundamental para garantizar la consistencia de los datos utilizados en la generación de reportes y en la toma de decisiones. Esto asegura que la información gestionada por el sistema esté siempre alineada con los estándares de calidad esperados y se integre de manera correcta entre todos los componentes del sistema.
- La adopción de una arquitectura de microservicios ha demostrado ser eficaz para mejorar la flexibilidad y escalabilidad del sistema. Cada microservicio puede ampliarse de forma independiente, lo que permite que el sistema crezca conforme a las demandas de la empresa sin comprometer su estabilidad. Este enfoque modular, que incluye la gestión centralizada de inventarios y ventas y la migración de datos históricos, asegura que el sistema pueda evolucionar sin necesidad de rediseños integrales, facilitando un mantenimiento .

6.4. Objetivo específico 4

- La validación del diseño arquitectónico demuestra que la arquitectura basada en microservicios y la integración de API REST han logrado garantizar una interoperabilidad completa entre los sistemas de Hortiluma (Kyte, FACEL y POS). La matriz de cobertura de requerimientos (Tabla 42) confirma que todos los requisitos funcionales y no funcionales han sido abordados, asegurando que no existan brechas en la comunicación y el flujo de datos entre las plataformas.
- La revisión documental y las pruebas de historias de usuario (Apéndice E.18. Plantilla de revisión documental - RD-018 y Apéndice E.20. Plantilla de revisión documental - RD-020) evidencian que la automatización de procesos críticos, como la actualización de inventarios y la generación de órdenes de compra, ha mejorado la coherencia de los datos. Esta automatización minimiza la necesidad de intervención manual, reduciendo los errores humanos y mejorando los tiempos de procesamiento, lo cual se traduce en una mayor eficiencia operativa para Hortiluma.
- La validación de la arquitectura modular a través de la matriz de cobertura y la plantilla de revisión documental (Apéndice E.16. Plantilla de revisión documental - RD-016) destaca la capacidad de la estructura de microservicios para adaptarse a futuras expansiones y demandas del negocio sin comprometer la estabilidad operativa actual. Esta flexibilidad es fundamental para que el sistema pueda evolucionar con las necesidades cambiantes de Hortiluma.
- La evaluación de los historias de usuario (Apéndice I.1. Plantilla de historias de usuario - HU-001 y Apéndice I.3. Plantilla de historias de usuario - HU-003) muestra que la centralización de la gestión de inventarios y la automatización de la sincronización de datos ha permitido eliminar la duplicación de tareas, mejorando la precisión y la confiabilidad de la información registrada. Este logro ha sido clave para mejorar la operativa diaria y la planificación estratégica de la empresa.
- Las pruebas de historias de usuario presentan condiciones reales verificando que el sistema cumple con los estándares de eficiencia esperados, mejorando la estabilidad operativa y garantizando que todas las funcionalidades críticas respondan adecuadamente ante escenarios de alta demanda. Esto permite una aceptación positiva del sistema por parte de los usuarios, facilitando su implementación futura.
- La validación también ha abordado la identificación de riesgos operativos y posibles puntos de fallo, desarrollando planes de mitigación efectivos (Apéndice E.21. Plantilla de revisión documental - RD-021). Estos mecanismos aseguran que la continuidad del negocio no se vea comprometida ante situaciones adversas, garantizando una operación resiliente y confiable para Hortiluma.

6.5. Objetivo general

- El proyecto de diseño de software para Hortiluma logró cumplir de manera integral con el objetivo general de mejorar la interoperabilidad y eficiencia operativa de los sistemas Kyte, FACEL y POS.
- La propuesta, basada en una arquitectura de microservicios y APIs, permitió eliminar la duplicidad de tareas y garantizar la sincronización automática de datos en tiempo real. Esto representa un avance significativo en la integración de procesos clave como la gestión de inventarios, facturación y ventas, mejorando los flujos de trabajo y reduciendo errores operativos.
- El desarrollo del proyecto se llevó a cabo en un periodo de 16 semanas, estructurado en cuatro fases metodológicas: Fase 1: Diagnóstico de la situación actual, Fase 2: Definición del estado meta, Fase 3: Diseño preliminar de la arquitectura, y Fase 4: Validación del diseño y cumplimiento de requisitos. Este enfoque permitió abordar las brechas existentes entre los sistemas actuales, simplificar los procesos operativos y garantizar que el diseño cumpla con las necesidades técnicas y estratégicas de la empresa. Además, el sistema propuesto se diseñó con la flexibilidad necesaria para incorporar nuevas funcionalidades en el futuro sin comprometer su estabilidad, posicionando a Hortiluma en una plataforma tecnológica moderna y escalable.
- La solución propuesta se alinea con las mejores prácticas de diseño de *software*, como se detalla en la sección diseño de *software*. Esto ha permitido no solo asegurar la interoperabilidad y eficiencia del sistema, sino también prepararlo para adaptarse a futuras necesidades de expansión de la empresa. Además, la arquitectura seleccionada facilita la integración de nuevas funcionalidades sin afectar la estabilidad operativa de Hortiluma.
- Desde una perspectiva financiera, el análisis muestra que la propuesta es económicamente viable, con un ROI proyectado del -49.06% y un plazo estimado de recuperación de inversión de 1.96 años. Este resultado valida no solo la sostenibilidad financiera del proyecto, sino también su capacidad para generar beneficios a mediano y largo plazo, reduciendo costos operativos y mejorando la eficiencia administrativa y comercial.

7. Recomendaciones

En este capítulo se presentan recomendaciones orientadas a asegurar la mejora continua del diseño de *software* propuesto para con el fin de mejorar la integración y operatividad de los sistemas. Los hallazgos de este trabajo demuestran la necesidad de implementar la propuesta de diseño de software desarrollada, priorizando la centralización de datos como estrategia clave para reducir la duplicación de tareas y mejorar la precisión operativa en Hortiluma. Las recomendaciones se basan en los hallazgos de los objetivos específicos.

7.1. Objetivo específico 1

- La duplicación de tareas en los sistemas Kyte, FACEL y POS genera ineficiencias, por lo que es esencial implementar un sistema centralizado de entrada de datos que distribuya la información a todas las plataformas. Esto reducirá el tiempo y los errores derivados de la repetición de tareas. La automatización mediante tecnologías de Automatización Robótica de Procesos (RPA) es clave para la actualización de inventarios y generación de reportes, incrementando la precisión y reduciendo costos operativos.
- Crear un plan para la automatización y estándares de datos, asegurando uniformidad en todos los sistemas para mejorar la integración y garantizar una información confiable para la toma de decisiones estratégicas.
- A corto y mediano plazo, se sugiere evaluar el potencial de inteligencia artificial para automatizar procesos de actualización de inventarios, identificando patrones que mejoren el flujo de información entre sistemas.
- Realizar auditorías periódicas para supervisar la integración entre funcionalidades, detectando problemas de duplicación o falta de sincronización y asegurando que las actualizaciones de inventarios y precios sean precisas.

7.2. Objetivo específico 2

- Se recomienda considerar la propuesta de modernización de la arquitectura mediante el uso de API REST y middleware para aumentar la flexibilidad y capacidad de adaptación de los sistemas.
- Extender el uso de APIs y middleware garantizará una infraestructura flexible y preparada para futuras actualizaciones, permitiendo integrar nuevas funciones sin comprometer la estabilidad del *software*.
- Mantener la integración de sistemas y verificación de interoperabilidad en áreas críticas como inventarios y facturación es esencial para evitar desajustes y mejorar la planificación.
- Evaluar continuamente nuevas tecnologías, incluyendo inteligencia artificial, para analizar y predecir demandas de inventarios y facilitar una respuesta ágil a las necesidades del mercado.
- Establecer indicadores de desempeño que midan la reducción de tareas manuales y el impacto de la automatización facilitará la evaluación de mejoras y permitirá ajustes para mejorar la interoperabilidad.

7.3. Objetivo específico 3

- Se recomienda documentar continuamente la arquitectura de microservicios para facilitar la escalabilidad y flexibilidad del sistema.
- Mantener actualizada la documentación del diseño asegura que el equipo de TI comprenda la estructura del sistema, útil para integraciones futuras y para la incorporación de nuevos miembros al equipo.
- Realizar pruebas periódicas de carga y resiliencia para garantizar que el sistema soporte incrementos de demanda. Esto es especialmente importante en caso de expansión de la empresa.
- Implementar versiones preliminares de componentes críticos para realizar ajustes antes de su implementación final, asegurando una integración sin problemas.
- Explorar el uso de inteligencia artificial en análisis predictivo de carga para anticipar demandas y ajustar la infraestructura de manera proactiva.

7.4. Objetivo específico 4

- Se recomienda establecer un sistema de monitoreo que detecte cualquier anomalía en la interoperabilidad, asegurando una operación continua y confiable.
- Implementar monitoreo para la consistencia de datos y calidad del servicio, detectando problemas antes de que afecten la operación.
- Actualizar la matriz de cobertura de requerimientos trimestralmente para reflejar el estado actual y garantizar que los requerimientos del sistema estén completamente integrados.
- Desarrollar planes de mitigación específicos para riesgos potenciales, especialmente en la migración de datos, asegurando estabilidad operativa y continuidad de negocio.
- Evaluar la integración de funcionalidades de inteligencia artificial para monitoreo predictivo, facilitando la identificación proactiva de problemas de interoperabilidad y mejorando la eficiencia del sistema.

7.5. Objetivo general

- Se recomienda implementar la evolución de la arquitectura conforme a la hoja de ruta propuesta en la sección costos de la futura implementación, integrando nuevas tecnologías y sistemas.
- Fomentar un ambiente de mejora continua, capacitando a los equipos en la identificación de oportunidades de mejora y el uso de inteligencia artificial en la gestión de procesos y toma de decisiones.
- Realizar análisis periódicos del ROI para monitorear la viabilidad financiera de la solución implementada, priorizando proyectos con alto impacto económico.
- Mantener la arquitectura flexible para adaptarse a cambios en regulaciones y nuevas demandas de clientes, asegurando que Hortiluma pueda incorporar innovaciones tecnológicas, como inteligencia artificial, de manera efectiva y mantener su competitividad en un mercado dinámico.

8. Referencias

- Bourque, P., & Fairley, R. (2014). *SWEBOK 3.0: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley.
- Heineman, G. T., & Councill, W. T. (2001). *Component-based software engineering: Putting the pieces together*. Addison-Wesley Professional.
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (6ª ed.). McGraw-Hill Education.
- ISO/IEC/IEEE. (2011). *Systems and software engineering - Requirements engineering (ISO/IEC/IEEE 29148)*. International Organization for Standardization.
- NIST. (n.d.). *Robotic systems interoperability and integration*. Recuperado de <https://www.nist.gov/programs-projects/robotic-systems-interoperability-and-integration>
- Paul, K., & Kimmel, P. (2011). *Manual de UML*. McGraw-Hill Education.
- Rivas, L. (2015). *¿Cómo hacer una tesis?* (3ra ed.). IPNEditors. <https://doi.org/10.13140/RG.2.1.3446.6644>
- Sommerville, I. (2011). *Software engineering*. Pearson.
- Tateeda Global. (n.d.). *Interoperability in healthcare IT: System integration guide*. Recuperado de <https://tateeda.com/healthcare-it-system-integration-guide>
- Wieggers, K., & Beatty, J. (2013). *The essential software requirement*. S4Carlisle Publishing Services.

9. Apéndices

Apéndice A. Cronograma de trabajo

Capítulo	Etapa	Semana															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Establecimiento del contexto y definición del problema.	Ajustes anteproyecto	■															
	Desarrollo del capítulo 1		■														
	Entrega capítulo 1			■													
2. Entrevistas con usuarios clave y revisión de documentación existente para especificar los requerimientos de integración.	Entrevistas con usuarios clave			■													
	Revisión de documentación existente			■	■												
	Recopilación de requerimientos			■	■												
	Análisis de requerimientos				■												
	Entrega marco metodológico					■											
	Entrega marco conceptual						■	■									
	Desarrollo del análisis de resultados							■	■								
3. Evaluación y recomendación de tecnologías y plataformas de integración adecuadas	Evaluación de tecnologías y plataformas							■									
	Análisis comparativo							■	■								
	Recomendación de tecnologías							■	■								
4. Desarrollo del diseño preliminar de la arquitectura de integración, detallando todos los componentes y mecanismos necesarios	Diseño preliminar de la arquitectura									■	■						
	Detalle de componentes y módulos									■	■						
	Diagramas de historias de usuario, componentes y secuencia									■	■						
	Plan de implementación y gestión de cambios											■					
5. Validación del diseño a través de pruebas y simulaciones, ajustando el diseño conforme a los resultados obtenidos	Pruebas con usuarios clave											■					
	Simulaciones operativas											■	■				
	Ajustes al diseño basado en resultados de pruebas												■				
6. Documentación completa de los resultados, conclusiones y recomendaciones para la implementación futura.	Entrega conclusiones y recomendaciones													■			
	Entrega documento final																■

Nota. Fuente: Elaboración propia.

Apéndice B. Plantilla de la minuta

Reunión No.	Es un núm. consecutivo para este proyecto	Fecha:	Indicar la fecha exacta de la reunión
Lugar:	Indicar dónde fue la reunión	Hora Inicio/Finalización:	xx:00 am. / yy:00 am
Objetivo de la reunión:			
Participante		Firma	
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
	Debe ser detallado, explícito	Debe ser detallado, explícito	Debe ser detallado, explícito
	Debe ser detallado, explícito	Debe ser detallado, explícito	Debe ser detallado, explícito
	Debe ser detallado, explícito	Debe ser detallado, explícito	Debe ser detallado, explícito
Próxima reunión			
Temas a tratar		Fecha	Convocados
En la próxima reunión		indicar	Nombre de quiénes asistirán a esta próxima reunión.

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_l0QyIoL90sTFsrM/view)

Apéndice C. Plantilla para la gestión de cambios

Solicitud de Cambio			
Datos Generales del Cambio			
N° Cambio			
Solicitante		Fecha de solicitud del cambio	
Responsable de la implementación		Fecha de realización del cambio	
Estado	<input type="checkbox"/> Aprobado	<input type="checkbox"/> En Revisión	<input type="checkbox"/> Rechazado
Detalles del Cambio			
Categoría			
Descripción detallada			
Justificación			
Implicaciones de realizar el cambio			
Impacto	() Alto	() Medio	() Bajo
Prioridad	() Alta	() Media	() Baja
Comentarios/ Observaciones			
Aprobación			
Rol	Nombre	Firma	

Nota. Adoptado de HojaControlCambio, por S. Mora, 2017, Google Sites (<https://drive.google.com/file/d/1jfnTi5TB8DXpnHBpWKe9Mfxqf3dNDny0/view>)

Apéndice D. Plantilla de entrevista

Entrevista		Número de entrevista
Fecha		
Entrevistador		
Aspectos clave		
Participantes		
Rol dentro del proyecto		
Pregunta	Respuesta	
Observaciones		

Nota. Fuente: Elaboración propia.

Apéndice E. Plantilla de revisión documental

Revisión documental		Número de revisión
Fecha		
Documento revisado		
Fuente		
Objetivo de la revisión	Información clave	

Nota. Fuente: Elaboración propia.

Apéndice F. Plantilla de observación directa

Observación directa		Número de observación
Fecha		
Observador		
Lugar		
Aspectos observados		
Descripción de las actividades		
Observaciones clave		

Nota. Fuente: Elaboración propia.

Apéndice G. Plantilla de grupo focal

Grupo focal		Número de sesión
Fecha		
Moderador		
Participantes		
Temas discutidos		
1.		
2.		
Pregunta	Respuesta	
Observaciones		

Nota. Fuente: Elaboración propia.

Apéndice H. Plantilla de lluvia de ideas

Lluvia de ideas		Número de sesión
Fecha		
Facilitador		
Participantes		
Ideas generadas		
1.		
2.		
Conclusiones clave		

Nota. Fuente: Elaboración propia.

Apéndice I. Plantilla de historias de usuario

Nombre de la historia de usuario		HU-00X	
Actores			
Precondiciones			
Postcondiciones			
Escenario principal		Flujo alterno	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Requerimientos cubiertos			

Nota. Fuente: Elaboración propia.

Apéndice J. Plantilla de requerimientos

Código	Descripción	Justificación
REQ-1	...	
REQ-2	...	
REQ-3	...	
REQ-N	...	

Nota. Fuente: Elaboración propia.

Apéndice K. Plantilla de matriz de cobertura de requerimientos

	HU-001	HU-002	HU-003	HU-00N
Requerimiento 1	X			
Requerimiento 2		X		
Requerimiento 3			X	
Requerimiento N				X

Nota. Fuente: Elaboración propia.

Apéndice B.1. Minuta de reunión entre tutor y contraparte

Reunión No.	1	Fecha:	8 de agosto de 2024
Lugar:	Zoom	Hora Inicio/Finalización:	2:00 p.m. – 2:30 p.m.
Objetivo de la reunión:	Confirmar próximas reuniones, revisar los hitos académicos y establecer las evaluaciones por parte de la empresa para el TFG.		
Participante		Firma	
Representante de Hortiluma Gerlyn Hernández Castillo			
Tutor Académico Ing. Luis Javier Chavarría Sánchez			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Confirmación de próximas reuniones	Se revisan y confirman las fechas para las próximas reuniones (semana 10 y 13).	Fechas confirmadas. Próximas reuniones en semana 10 y 13.
2	Presentación de hitos académicos	El estudiante presentó los hitos alcanzados hasta el momento.	Empresa da visto bueno sobre avances académicos.
3	Evaluaciones por parte de la empresa	Se discutió el proceso de evaluación y su importancia en el TFG.	La empresa acordó realizar evaluaciones periódicas.
4	Reglamento del TFG	Se revisó el reglamento para garantizar que todos los involucrados están alineados.	El reglamento fue comprendido y aceptado por todas las partes.
5	Varios	Otros temas menores discutidos.	No hay acuerdos adicionales relevantes.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Segunda evaluación de avances del TFG		Jueves, 26 de septiembre	Representantes de Hortiluma, estudiante, tutor

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.2. Minuta de entrevista - ENT-001

Reunión No.	2	Fecha:	15 de agosto de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	10:00 a.m. – 11:00 a.m.
Objetivo de la reunión:	Evaluar la eficiencia operativa y los problemas de integración de sistemas en el área de ventas mayoristas.		
Participante		Firma	
Responsable de ventas mayoristas Lucrecia Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Impacto de la duplicación de tareas en ventas	La duplicación de tareas en Kyte y FACEL genera inconsistencias en los reportes, afectando decisiones rápidas.	Se acuerda priorizar la integración de sistemas Kyte y FACEL.
2	Falta de automatización en la gestión de inventarios	La falta de automatización provoca discrepancias en inventarios y afecta la planificación de ventas.	Se decide investigar soluciones para automatizar actualizaciones en tiempo real.
3	Retrasos y errores en la consolidación de reportes	La falta de integración causa errores en la consolidación de los reportes.	Se acuerda establecer un plan de acción para resolver estas inconsistencias.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Revisión de los avances en la integración de sistemas		30 de agosto de 2024	Representantes de ventas, equipo de sistemas

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.3. Minuta de entrevista - ENT-002

Reunión No.	3	Fecha:	18 de agosto de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	1:00 p.m. – 2:00 p.m.
Objetivo de la reunión:	Identificar problemas de comunicación entre sistemas y brechas de interoperabilidad en ventas minoristas.		
Participante		Firma	
Responsable de ventas minoristas Rita Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Falta de integración entre sistemas	La falta de integración obliga a realizar tareas duplicadas y genera retrasos en la actualización de inventarios.	Se acuerda realizar un análisis más detallado de las brechas de interoperabilidad.
2	Impacto en los reportes de ventas por la falta de sincronización	Los reportes de ventas son inconsistentes entre sistemas, lo que genera errores que deben corregirse manualmente.	Se establece un plan de revisión de reportes para identificar los errores comunes.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Plan de acción para la mejora de la interoperabilidad		30 de agosto de 2024	Representantes de ventas minoristas, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.4. Minuta de entrevista - ENT-003

Reunión No.	4	Fecha:	20 de agosto de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	11:00 a.m. – 12:00 p.m.
Objetivo de la reunión:	Evaluar los problemas de redundancia funcional y errores operativos en los sistemas de la empresa.		
Participante	Firma		
Representante de Hortiluma Gerlyn Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Problemas de duplicación de datos	Los empleados ingresan los mismos datos en varios sistemas, generando inconsistencias y retrabajo.	Se acuerda implementar un sistema de automatización para reducir la duplicación de tareas.
2	Automatización de procesos para reducir errores	La automatización de inventarios y precios mejoraría la eficiencia operativa, reduciendo los errores manuales.	Se decide investigar herramientas para automatizar estos procesos.
3	Problemas de facturación debido a la falta de integración	Los errores en la conciliación de facturación aumentan debido a la falta de integración entre sistemas.	Se establecerá un equipo para explorar soluciones de integración que mejoren la facturación.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Revisión de propuestas de automatización e integración		5 de septiembre de 2024	Representantes de administración y equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.5. Minuta de entrevista - ENT-004

Reunión No.	5	Fecha:	18 de septiembre de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	10:00 a.m. – 11:00 a.m.
Objetivo de la reunión:	Identificar las redundancias críticas en los procesos de ventas mayoristas y explorar soluciones para simplificarlas.		
Participante		Firma	
Responsable de ventas mayoristas Lucrecia Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Redundancias más críticas en los procesos de ventas e inventarios	La duplicación de datos entre los sistemas Kyte, FACEL y POS es uno de los principales problemas operativos.	Priorizar una solución única de entrada de datos para eliminar la duplicación.
2	Impacto de la simplificación de procesos	Reduciría el tiempo de procesamiento y los errores humanos, mejorando la eficiencia general.	Se acordó explorar middleware o APIs que puedan facilitar la sincronización de datos entre los sistemas.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Revisión de herramientas para la sincronización de datos		1 de octubre de 2024	Responsable de ventas mayoristas, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFP04xQ_10QyIoL90sTFsrM/view)

Apéndice B.6. Minuta de entrevista - ENT-005

Reunión No.	6	Fecha:	20 de septiembre de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	02:00 p.m. – 03:00 p.m.
Objetivo de la reunión:	Explorar las herramientas tecnológicas disponibles para la integración de sistemas y la eliminación de deuda técnica.		
Participante	Firma		
Representante de Hortiluma Gerlyn Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Herramientas tecnológicas para soportar la integración	Las plataformas actuales no soportan adecuadamente la integración; es necesario un rediseño completo.	Iniciar un análisis de alternativas tecnológicas que permitan una reingeniería de los procesos actuales.
2	Retos tecnológicos en la compatibilidad de sistemas	La falta de compatibilidad entre Kyte y FACEL es un obstáculo importante para la integración.	Explorar el uso de middleware o APIs que faciliten la integración entre Kyte, FACEL y POS.
3	Procesos redundantes que afectan la operatividad	La duplicación de tareas en la gestión de inventarios y ventas es un problema clave que genera errores.	Considerar la automatización de procesos para eliminar las tareas manuales redundantes.
4	Soluciones tecnológicas para eliminar redundancias	La implementación de un sistema único de entrada de datos, sincronizado en tiempo real, mejoraría la eficiencia.	Se acordó investigar herramientas RPA para la automatización de procesos repetitivos.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Revisión de propuestas para la automatización de procesos		5 de octubre de 2024	Administrador del negocio, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.7. Minuta de entrevista - ENT-006

Reunión No.	7	Fecha:	22 de septiembre de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	10:00 a.m. – 11:00 a.m.
Objetivo de la reunión:	Evaluar herramientas tecnológicas para la integración de sistemas en ventas minoristas y analizar el impacto de la simplificación de procesos.		
Participante		Firma	
Responsable de ventas minoristas Rita Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Herramientas tecnológicas para la integración	Actualmente no hay herramientas dentro de Kyte y FACEL que faciliten una integración adecuada.	Se acuerda explorar soluciones en la nube y middleware para conectar los sistemas Kyte, FACEL y POS.
2	Impacto de la integración en la gestión de ventas minoristas	La integración de los sistemas eliminaría la necesidad de ingresar datos manualmente en varios sistemas, mejorando la eficiencia.	Priorizar la implementación de soluciones en la nube que permitan una sincronización automática de los datos.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Propuesta de integración de sistemas basada en soluciones en la nube		10 de octubre de 2024	Responsable de ventas minoristas, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.8. Minuta de reunión entre tutor y contraparte

Reunión No.	8	Fecha:	26 de septiembre
Lugar:	Zoom	Hora Inicio/Finalización:	2:00 p.m. – 2:30 p.m.
Objetivo de la reunión:	Revisar el avance del documento académico del TFG y recibir retroalimentación de la contraparte empresarial.		
Participante		Firma	
Representante de Hortiluma Gerlyn Hernández Castillo			
Tutor Académico Ing. Luis Javier Chavarría Sánchez			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Recordatorio de segunda evaluación por la contraparte empresarial	Se recordó la importancia de la segunda evaluación y su alcance.	La evaluación será presentada en la siguiente reunión.
2	Avance del documento académico del TFG	El estudiante presentó los avances en el desarrollo del TFG.	Se recomienda ampliar los detalles en ciertos apartados.
3	Observaciones sobre el desarrollo del TFG	La contraparte empresarial dio su perspectiva sobre el avance.	Se acordaron ajustes en base a las observaciones dadas por Hortiluma.
4	Varios (todos)	Otros temas discutidos.	No hay acuerdos adicionales relevantes.
1	Recordatorio de segunda evaluación por la contraparte empresarial	Se recordó la importancia de la segunda evaluación y su alcance.	La evaluación será presentada en la siguiente reunión.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Tercera revisión del TFG		Jueves, 17 de octubre	Representantes de Hortiluma, estudiante, tutor

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.9. Minuta de entrevista - ENT-007

Reunión No.	9	Fecha:	30 de septiembre de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	10:00 a.m. – 11:00 a.m.
Objetivo de la reunión:	Identificar los principales problemas de interoperabilidad, expectativas del sistema, y los impactos de las duplicaciones en ventas y operatividad.		
Participante		Firma	
Representante de Hortiluma Gerlyn Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Desafíos con el sistema actual	La falta de integración entre Kyte, FACEL y POS genera múltiples problemas operativos, aumentando los errores.	Priorizar la integración de estos sistemas para evitar duplicaciones.
2	Mejora esperada con un sistema integrado y centralizado	Un sistema con sincronización automática en tiempo real mejoraría la eficiencia operativa y reduciría errores.	Explorar soluciones de automatización y sincronización en tiempo real.
3	Impacto de la falta de sincronización en la planificación y experiencia del cliente	Los productos agotados aparecen como disponibles, lo que afecta las ventas y la satisfacción del cliente.	Acelerar la implementación de un sistema que actualice inventarios en tiempo real.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Revisión de propuestas de integración y sincronización		30 de agosto de 2024	Administrador del negocio, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_I0QyIoL90sTFsrM/view)

Apéndice B.10. Minuta de entrevista - ENT-008

Reunión No.	10	Fecha:	1 de octubre de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	01:00 p.m. – 02:00 p.m.
Objetivo de la reunión:	Analizar el impacto de las duplicaciones de tareas y las mejoras esperadas en la planificación y gestión de inventarios.		
Participante		Firma	
Responsable de ventas mayoristas Lucrecia Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Impacto de las tareas duplicadas en eficiencia	Los empleados actualizan manualmente los inventarios en tres sistemas, lo que provoca inconsistencias y errores.	Se acuerda explorar un sistema que sincronice automáticamente las actualizaciones.
2	Expectativas de un sistema integrado	La automatización permitiría gestionar los inventarios con mayor precisión y coherencia entre sucursales.	Priorizar un sistema que sincronice en tiempo real los datos de inventarios y ventas.
3	Problemas en la operatividad diaria debido a la falta de integración	La falta de coherencia entre sistemas mayoristas y minoristas complica la gestión de grandes volúmenes de pedidos.	Se acordó un análisis de los sistemas mayoristas y minoristas para buscar una mejor integración.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Propuesta de automatización para la gestión de inventarios		31 de agosto de 2024	Responsable de ventas mayoristas, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.11. Minuta de entrevista - ENT-009

Reunión No.	11	Fecha:	3 de octubre de 2024
Lugar:	Oficina central de Hortiluma	Hora Inicio/Finalización:	11:00 a.m. – 12:00 p.m.
Objetivo de la reunión:	Evaluar los problemas técnicos de interoperabilidad y analizar soluciones de automatización para reducir errores.		
Participante		Firma	
Representante de Hortiluma Gerlyn Hernández Castillo			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Impacto de las tareas duplicadas en ventas e inventarios	Los empleados actualizan manualmente los inventarios en tres sistemas distintos, lo que genera errores.	Se acuerda explorar herramientas de automatización para reducir la duplicación de tareas.
2	Expectativas de un sistema integrado y automatizado	La automatización permitiría una mejora en la gestión inventarios y coherencia de datos entre sucursales.	Priorizar un sistema que sincronice en tiempo real los datos de inventarios y ventas.
3	Efectos de la falta de integración en la operatividad diaria	Los constantes errores en los inventarios generan pérdidas de tiempo y recursos en las operaciones diarias.	Se decide realizar una auditoría de los sistemas actuales para identificar mejoras posibles.
Próxima reunión			
Temas a tratar		Fecha	Convocados
Propuesta de automatización e integración de sistemas		3 de septiembre de 2024	Administrador del negocio, equipo técnico

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_l0QyIoL90sTFsrM/view)

Apéndice B.12. Minuta de reunión entre tutor y contraparte

Reunión No.	12	Fecha:	17 de octubre
Lugar:	Zoom	Hora Inicio/Finalización:	2:00 p.m. – 2:30 p.m.
Objetivo de la reunión:	Revisar el avance del documento académico del TFG y recibir retroalimentación de la contraparte empresarial.		
Participante		Firma	
Representante de Hortiluma Gerlyn Hernández Castillo			
Tutor Académico Ing. Luis Javier Chavarría Sánchez			
Estudiante Luis Sebastian Hernández Venegas			
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Recordatorio de la tercera evaluación por la contraparte empresarial	Se recordó la importancia de la tercera evaluación y su alcance.	La evaluación será agregada a dropbox una vez se realice.
2	Avance del documento académico del TFG	El estudiante presentó los avances en el desarrollo del TFG.	Se recomienda ajustar detalles en ciertos apartados.
3	Observaciones sobre el desarrollo del TFG	La contraparte empresarial dio su perspectiva sobre el avance.	Se acordaron ajustes en base a las observaciones del tutor.
4	Varios (todos)	Otros temas discutidos.	No hay acuerdos adicionales relevantes.

Nota. Adoptado de Machote de minuta-ejemplo, por S. Mora, 2017, Google Sites (https://drive.google.com/file/d/1b65nGntkFzFPO4xQ_10QyIoL90sTFsrM/view)

Apéndice B.13. Minutas firmadas

En el siguiente enlace se pueden verificar las firmas de todas las minutas firmadas https://drive.google.com/file/d/1RO5KLn5_p7NABwIFvcKsOkHfcsSinAWQ/view?usp=sharing

Apéndice E.1. Plantilla de revisión documental - RD-001

Revisión documental		RD-001
Fecha	12 de agosto de 2024	
Documento revisado	Manual técnico de Kyte, FACEL, POS legado; reportes de inventarios internos	
Fuente	Documentación interna de Hortiluma y sitios web oficiales	
Objetivo de la revisión	Información clave	
Identificar duplicaciones de datos y funcionalidades en la gestión de ventas e inventarios	<ul style="list-style-type: none"> - Se encontró que Kyte tiene duplicación de funcionalidades, lo que genera retrabajo en ventas e inventarios. La actualización manual de inventarios en este sistema provoca errores en los reportes, como registros inconsistentes entre las cantidades de productos vendidos y el inventario disponible. - Se identificaron errores en la actualización de inventarios debido a la falta de integración automática entre FACEL y los demás sistemas. Esto genera inconsistencias en los niveles de inventario que afectan la precisión de los reportes de ventas e inventarios. - La actualización manual de datos entre las ventas registradas y los inventarios en el POS legado es inconsistente, lo que genera errores en los reportes de inventarios, afectando la coherencia de la información entre los sistemas. 	

Nota. Fuente: Elaboración propia.

Apéndice E.2. Plantilla de revisión documental - RD-002

Revisión documental		RD-002
Fecha	15 de agosto de 2024	
Documento revisado	Reportes de integración entre Kyte, FACEL y POS	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Evaluar las brechas de interoperabilidad entre los sistemas actuales	<ul style="list-style-type: none"> - Se detectó que no hay un proceso de sincronización automática entre Kyte y FACEL, lo que provoca que el personal tenga que ingresar manualmente los datos en ambos sistemas. Esta duplicación de trabajo genera errores recurrentes en los reportes de ventas e inventarios, lo que afecta la coherencia de la información. - El POS no está sincronizado con Kyte y FACEL, lo que obliga a realizar actualizaciones manuales, generando una duplicación de tareas y errores operativos. 	

Nota. Fuente: Elaboración propia.

Apéndice E.3. Plantilla de revisión documental - RD-003

Revisión documental		RD-003
Fecha	20 de agosto de 2024	
Documento revisado	Documentos técnicos de integración de sistemas	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Identificar el estado de integración entre Kyte, FACEL y POS	<ul style="list-style-type: none"> - No existe una integración de los sistemas por lo que las actualizaciones se realizan de manera manual, lo que provoca errores en la facturación e inventarios. Estos errores se reflejan en la falta de coherencia en los datos entre los sistemas. - Se detectó una falta de comunicación entre los módulos de ventas y facturación, lo que genera reportes inconsistentes respecto a las cantidades de productos vendidos y el inventario disponible. 	

Nota. Fuente: Elaboración propia.

Apéndice E.4. Plantilla de revisión documental - RD-004

Revisión documental		RD-004
Fecha	25 de agosto de 2024	
Documento revisado	Manual de procesos de ventas y actualizaciones de inventarios	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Identificar problemas operativos derivados de la duplicación de tareas en ventas e inventarios	<ul style="list-style-type: none"> - El personal de ventas tiene que ingresar la información de ventas manualmente en tres sistemas separados (Kyte, FACEL y POS). Esto genera duplicación de datos y errores en los reportes de inventarios, afectando la coherencia entre los sistemas. - El proceso de actualización manual de datos provoca un retraso significativo en la actualización de inventarios y ventas, lo que afecta la precisión de los reportes y la capacidad de tomar decisiones informadas. 	

Nota. Fuente: Elaboración propia.

Apéndice E.5. Plantilla de revisión documental - RD-005

Revisión documental		RD-005
Fecha	28 de agosto de 2024	
Documento revisado	Manual de operaciones de facturación y manejo de inventarios	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Evaluar las inconsistencias entre los reportes de ventas y los inventarios en Kyte, FACEL y POS	<ul style="list-style-type: none"> - Se detectaron discrepancias significativas entre los inventarios registrados en Kyte y FACEL, lo que afecta la precisión de los reportes de ventas. Estas discrepancias se deben a la falta de sincronización automática y a que las actualizaciones deben realizarse de manera manual. - Los reportes de facturación generados en el POS no coinciden con las cantidades de productos vendidas registradas en Kyte y FACEL, lo que genera problemas en la conciliación contable y afecta la coherencia entre los reportes financieros y operativos. 	

Nota. Fuente: Elaboración propia.

Apéndice D.1. Plantilla de entrevista - ENT-001

Entrevista		ENT-001
Fecha	15 de agosto de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Eficiencia operativa y problemas de integración de sistemas	
Participantes		
Rol dentro del proyecto	Responsable de ventas mayoristas	
Pregunta	Respuesta	
¿Qué impacto tiene la duplicación de tareas en las ventas mayoristas?	La duplicación de tareas en la entrada de datos en los sistemas Kyte y FACEL genera inconsistencias en los reportes de ventas, afectando la capacidad para tomar decisiones rápidas.	
¿Cómo afecta la falta de automatización en la gestión de inventarios mayoristas?	La falta de automatización hace que los inventarios no se actualicen en tiempo real, lo que provoca discrepancias y errores en la cantidad de productos disponibles, afectando la planificación de ventas.	
Observaciones		
<ul style="list-style-type: none"> - La falta de integración y sincronización en tiempo real entre Kyte, FACEL y POS obliga al equipo de ventas a realizar tareas duplicadas, lo que provoca errores en la consolidación de reportes y afecta la eficiencia en las ventas mayoristas. - La falta de comunicación entre sistemas provoca retrasos en la actualización de inventarios, lo que complica la planificación de ventas y genera pérdidas de tiempo en las operaciones diarias. 		

Nota. Fuente: Elaboración propia.

Apéndice D.2. Plantilla de entrevista - ENT-002

Entrevista		ENT-002
Fecha	18 de agosto de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Problemas de comunicación entre sistemas, brechas de interoperabilidad	
Participantes		
Rol dentro del proyecto	Responsable de ventas minoristas	
Pregunta	Respuesta	
¿Cómo afecta la falta de integración entre sistemas?	La falta de integración obliga a realizar tareas duplicadas, lo que genera retrasos en la actualización de inventarios y errores en la facturación.	
¿Cómo se ven afectados los reportes de ventas debido a las brechas de interoperabilidad?	Los reportes de ventas son inconsistentes entre los sistemas debido a la falta de sincronización en tiempo real, lo que provoca errores que deben ser corregidos manualmente.	
Observaciones		
<ul style="list-style-type: none"> - La falta de comunicación entre Kyte, FACEL y POS afecta la precisión de los datos operativos, lo que genera errores en la facturación y retrasos en la toma de decisiones. - El personal de ventas tiene que actualizar manualmente la misma información en varios sistemas, lo que aumenta la posibilidad de errores operativos y retrasa la gestión de inventarios. - Debido a la falta de integración, se genera re trabajo significativo para corregir errores en los inventarios y en la facturación. 		

Nota. Fuente: Elaboración propia.

Apéndice D.3. Plantilla de entrevista - ENT-003

Entrevista		ENT-003
Fecha	20 de agosto de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Problemas de redundancia funcional y errores operativos	
Participantes		
Rol dentro del proyecto	Administrador del negocio	
Pregunta	Respuesta	
¿Cuáles son los principales problemas de duplicación de datos?	Los empleados ingresan los mismos datos en varios sistemas (Kyte, FACEL y POS), lo que genera inconsistencias frecuentes en los reportes y retrabajo en los procesos operativos.	
¿Qué procesos podrían beneficiarse de mayor automatización?	La automatización de inventarios y precios reduciría los errores operativos y agilizaría el trabajo diario, ya que los empleados no tendrían que ingresar la misma información en múltiples sistemas.	
¿Cuáles son los principales problemas de facturación que enfrentan debido a la falta de integración de los sistemas?	La falta de integración entre los sistemas genera errores en la conciliación de facturación, lo que implica realizar correcciones manuales constantemente.	
¿Qué áreas del negocio se ven más afectadas por estos problemas de integración?	La facturación y la actualización de inventarios son las áreas más afectadas, ya que la información no se sincroniza automáticamente, lo que provoca inconsistencias en los datos contables.	
¿Cómo afecta la duplicación de tareas en la actualización de inventarios?	Los empleados deben actualizar manualmente los inventarios en Kyte, FACEL y POS, lo que genera discrepancias entre los sistemas y retrasa la planificación logística.	
¿Qué problemas se presentan por la falta de sincronización en tiempo real?	La falta de sincronización automática provoca errores en la cantidad de productos disponibles, lo que afecta tanto la reposición de inventario como las ventas.	
Observaciones		
<ul style="list-style-type: none"> - La entrada manual de datos en Kyte, FACEL y POS genera inconsistencias en facturación e inventarios. - La automatización de inventarios y precios mejoraría la eficiencia operativa. - La falta de integración entre sistemas genera duplicación de procesos, errores contables y retrasos en la actualización de inventarios. - La ausencia de sincronización en tiempo real complica la planificación logística y afecta la operación diaria. 		

Nota. Fuente: Elaboración propia.

Apéndice F.1. Plantilla de observación directa - OBS-001

Observación directa		OBS-001
Fecha	18 de agosto de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Procesos de gestión de inventarios y ventas		
Descripción de las actividades		
Los empleados ingresaban manualmente los datos de ventas e inventarios en los sistemas Kyte, FACEL y POS, lo que generaba inconsistencias y retrabajo al tener que actualizar los datos de forma independiente en cada sistema. La actualización de precios también se realizaba manualmente en cada sistema, lo que provocaba discrepancias en los reportes de inventarios y precios de venta.		
Observaciones clave		
<ul style="list-style-type: none"> - Los empleados deben ingresar la misma información en tres sistemas diferentes (Kyte, FACEL y POS), lo que genera retrabajo y aumenta la posibilidad de errores. - La ausencia de sincronización automática entre los sistemas ralentiza la actualización de inventarios y precios, afectando la coherencia de los datos. - La duplicación de procesos manuales es un indicador de ineficiencia operativa, lo que provoca cuellos de botella en los procesos de ventas e inventarios, afectando la capacidad de tomar decisiones rápidas y precisas. 		

Nota. Fuente: Elaboración propia.

Apéndice F.2. Plantilla de observación directa - OBS-002

Observación directa		OBS-002
Fecha	20 de agosto de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Proceso de actualización de inventarios y ventas		
Descripción de las actividades		
La actualización de inventarios y precios se realizaba de manera manual en los tres sistemas (Kyte, FACEL y POS). Los reportes generados no coincidían entre sistemas, lo que provocaba retrasos en la verificación de inventario y en la facturación.		
Observaciones clave		
<ul style="list-style-type: none"> - La actualización manual de inventarios en tres sistemas diferentes genera inconsistencias operativas y discrepancias en los datos. - La falta de sincronización automática entre los sistemas impide que los inventarios se actualicen en tiempo real, lo que afecta la precisión de los datos y retrasa la toma de decisiones. 		

Nota. Fuente: Elaboración propia.

Apéndice F.3. Plantilla de observación directa - OBS-003

Observación directa		OBS-003
Fecha	22 de agosto de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Proceso de facturación y conciliación de ventas		
Descripción de las actividades		
Se observó que el personal de ventas realiza el proceso de facturación de manera manual para ventas al por mayor en el sistema FACEL, a diferencia de Kyte que genera la factura automática de acuerdo con el pedido solicitado, pero no se encuentra integrado con los otros sistemas, generando discrepancias entre las ventas realizadas y los reportes de facturación generados. El POS no está sincronizado con los otros sistemas, lo que implica que la información de ventas debe actualizarse manualmente en cada sistema. Esto provoca retrasos en la conciliación de reportes y genera errores en los datos contables.		
Observaciones clave		
<ul style="list-style-type: none"> - El proceso de facturación se realiza de manera diferente en Kyte y FACEL, lo que provoca errores en la conciliación de ventas y dificulta la consolidación de datos entre los sistemas. - El POS no se comunica con los otros sistemas, lo que genera una falta de actualización automática de los datos de ventas y facturación, complicando la verificación de la información. - La duplicación de tareas y la falta de integración en el proceso de facturación y ventas son evidencias de un diseño d en los flujos de trabajo y gestión de datos. 		

Nota. Fuente: Elaboración propia.

Apéndice F.4. Plantilla de observación directa - OBS-004

Observación directa		OBS-004
Fecha	25 de agosto de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Proceso de actualización de inventarios y ventas mayoristas		
Descripción de las actividades		
Se observó que la actualización de inventarios para las ventas mayoristas se realiza de manera manual sin ninguno de los sistemas. La falta de sincronización entre los sistemas provoca discrepancias en las cantidades de productos disponibles y genera retrasos en la planificación de reposición de inventario. Los reportes de ventas generados en los tres sistemas no coinciden, lo que complica la conciliación de inventarios y afecta la precisión de los datos operativos.		
Observaciones clave		
- La falta de integración y la necesidad de realizar entradas manuales afectan la precisión de los datos de ventas y generan problemas de conciliación entre los sistemas.		

Nota. Fuente: Elaboración propia.

Apéndice E.6. Plantilla de revisión documental - RD-006

Revisión documental		RD-006
Fecha	10 de septiembre de 2024	
Documento revisado	Informes técnicos de Kyte, FACEL y POS legado	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Identificar herramientas tecnológicas para la integración	<ul style="list-style-type: none"> - Se detectaron módulos en desuso que podrían ser reemplazados o eliminados para simplificar el sistema. - FACEL presenta múltiples incompatibilidades con otros sistemas, creando cuellos de botella en la interoperabilidad. - La deuda técnica acumulada del POS legado y FACEL impide una integración fluida, y requiere intervención urgente para eliminar las rigideces técnicas que afectan la interoperabilidad y aumentan costos. 	

Nota. Fuente: Elaboración propia.

Apéndice E.7. Plantilla de revisión documental - RD-007

Revisión documental		RD-007
Fecha	10 de septiembre de 2024	
Documento revisado	Informes técnicos de Kyte, FACEL y POS legado	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Evaluar las herramientas actuales de automatización e integración	<ul style="list-style-type: none"> - Los sistemas evaluados tienen un diseño rígido que limita la capacidad de implementar automatización y sincronización de datos. - El POS funciona de manera aislada, generando duplicaciones de tareas y errores en la facturación. - Ninguno de los sistemas cuenta con un proceso de automatización adecuado en áreas clave como inventarios y facturación, lo que afecta la eficiencia operativa. 	

Nota. Fuente: Elaboración propia.

Apéndice E.8. Plantilla de revisión documental - RD-008

Revisión documental		RD-008
Fecha	20 de septiembre de 2024	
Documento revisado	Informes técnicos de Kyte, FACEL y POS legado	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Analizar la compatibilidad de los procesos internos con nuevas propuestas tecnológicas	<ul style="list-style-type: none"> - La actualización de datos e inventarios es manual, lo que genera duplicación de tareas, errores y costos innecesarios. - No hay actualización automática ni sincronización en tiempo real entre Kyte y POS, generando discrepancias en los datos. - Los procesos actuales no están listos para nuevas tecnologías de integración, lo que implica una reingeniería antes de implementar soluciones. 	

Nota. Fuente: Elaboración propia.

Apéndice E.9. Plantilla de revisión documental - RD-009

Revisión documental		RD-009
Fecha	25 de septiembre de 2024	
Documento revisado	Reportes internos sobre integraciones entre sistemas	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Identificar fallos en la integración de sistemas anteriores y buscar soluciones viables	<ul style="list-style-type: none"> - Kyte y FACEL no cuentan con APIs compatibles, lo que impide la integración directa, recomendando el uso de middleware como solución. - La falta de APIs moderna en POS impidió su integración con otros sistemas, lo que agravó la duplicación de datos y generó incoherencias en los reportes 	

Nota. Fuente: Elaboración propia.

Apéndice E.10. Plantilla de revisión documental - RD-010

Revisión documental		RD-010
Fecha	30 de septiembre de 2024	
Documento revisado	Informes técnicos de Kyte, FACEL y POS legado	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Evaluar el impacto de la falta de automatización en el sistema de inventarios	<ul style="list-style-type: none"> - Los inventarios se actualizan de forma manual en todos los sistemas, lo que genera redundancias y errores que afectan la precisión de decisiones de compra y ventas. - La falta de sincronización entre inventarios y ventas causa discrepancias en la facturación, lo que complica la gestión financiera del negocio. - La automatización del proceso de actualización de inventarios es fundamental para garantizar la coherencia de datos y mejorar la precisión de los reportes financieros. 	

Nota. Fuente: Elaboración propia.

Apéndice D.4. Plantilla de entrevista - ENT-004

Entrevista		ENT-004
Fecha	18 de septiembre de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Simplificación de procesos y eliminación de redundancias	
Participantes		
Rol dentro del proyecto	Responsable de ventas mayoristas	
Pregunta		Respuesta
¿Cuáles son las redundancias más críticas en los procesos actuales?		La duplicación de datos en los sistemas de ventas e inventarios es uno de los problemas más graves.
¿Qué impacto tendría la simplificación de estos procesos?		La simplificación reduciría el tiempo de procesamiento y los errores humanos, mejorando la eficiencia general del sistema.
Observaciones		
Sin la capacidad de integración directa, la duplicación de datos solo puede resolverse mediante la implementación de un sistema único de entrada de datos, o el uso de soluciones intermediarias como APIs o middleware para sincronizar los sistemas Kyte, FACEL y POS.		

Nota. Fuente: Elaboración propia.

Apéndice D.5. Plantilla de entrevista - ENT-005

Entrevista		ENT-005
Fecha	20 de septiembre de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Herramientas tecnológicas para integración y eliminación de deuda técnica	
Participantes		
Rol dentro del proyecto	Administrador del negocio	
Pregunta	Respuesta	
¿Qué herramientas actuales pueden soportar la integración entre sistemas?	Ninguna de las plataformas actuales soporta adecuadamente la integración; un rediseño completo es necesario.	
¿Cuáles son los mayores retos tecnológicos actuales?	La falta de compatibilidad entre los sistemas Kyte y FACEL impide una integración .	
¿Qué procesos redundantes afectan más la operatividad actual?	La duplicación de tareas en la gestión de inventarios y ventas es uno de los mayores problemas, ya que los datos deben ingresarse manualmente en los tres sistemas (Kyte, FACEL y POS), lo que genera errores en los reportes.	
¿Qué soluciones tecnológicas podrían eliminar estas redundancias?	La implementación de un sistema único de entrada de datos, sincronizado en tiempo real, eliminaría la duplicación de tareas, mejorando la coherencia y eficiencia de los datos. Además, las herramientas de automatización robótica de procesos (RPA) podrían ser útiles para eliminar tareas repetitivas en los procesos de actualización de inventarios y facturación.	
Observaciones		
<p>Será necesario explorar alternativas como middleware para conectar los sistemas Kyte, FACEL y POS, y eliminar la deuda técnica acumulada en POS. Esto permitirá una integración y automatización de procesos.</p> <p>La propuesta debe enfocarse en una reingeniería completa de los procesos, eliminando los sistemas que generan cuellos de botella y adoptando tecnologías más modernas que aseguren la integración y la automatización.</p>		

Nota. Fuente: Elaboración propia.

Apéndice D.6. Plantilla de entrevista - ENT-006

Entrevista		ENT-006
Fecha	22 de septiembre de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Simplificación de procesos y eliminación de redundancias	
Participantes		
Rol dentro del proyecto	Responsable de ventas minoristas	
Pregunta	Respuesta	
¿Qué herramientas tecnológicas actuales podrían facilitar la integración entre sistemas?	Actualmente no hay herramientas tecnológicas dentro de los sistemas Kyte y FACEL que faciliten una integración adecuada. Sin embargo, el uso de APIs o plataformas en la nube podría facilitar la sincronización de datos entre los diferentes módulos.	
¿Cómo impactaría la integración en la gestión de ventas minoristas?	La integración de los sistemas mejoraría la eficiencia operativa, eliminando la necesidad de ingresar datos manualmente en varios sistemas y permitiendo un control más preciso de inventarios y facturación.	
Observaciones		
El enfoque debe estar en la implementación de soluciones en la nube o middleware que permitan la interoperabilidad entre los sistemas Kyte, FACEL y POS, lo cual reduciría los tiempos de procesamiento y errores manuales en la gestión de ventas.		

Nota. Fuente: Elaboración propia.

Apéndice F.5. Plantilla de observación directa - OBS-005

Observación directa		OBS-005
Fecha	22 de septiembre de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Procesos de gestión de inventarios y reposición de inventario		
Descripción de las actividades		
Durante el proceso de reposición de inventarios, se detectó que los sistemas Kyte y FACEL no reflejaban en tiempo real los niveles de inventario, lo que complicaba la toma de decisiones para la reposición de productos. La verificación manual de las existencias retrasaba la gestión de inventarios y afectaba la precisión de las órdenes de compra.		
Observaciones clave		
<ul style="list-style-type: none"> - Las actualizaciones de inventarios no se reflejan en tiempo real, lo que provoca retrasos en la reposición de productos y afecta la disponibilidad en el punto de venta. - Un 15% de las actualizaciones de inventarios presenta discrepancias, afectando la planificación de compras y aumentando el riesgo de desabastecimiento. 		

Nota. Fuente: Elaboración propia.

Apéndice F.6. Plantilla de observación directa - OBS-006

Observación directa		OBS-006
Fecha	25 de septiembre de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Procesos de actualización de inventarios y ventas		
Descripción de las actividades		
Se observó que los empleados registraban manualmente los cambios en los inventarios en FACEL y POS de forma independiente, lo que generaba inconsistencias entre los sistemas y retrasos en la actualización de la información. Estas diferencias afectaban los reportes de inventarios y la planificación de ventas.		
Observaciones clave		
<ul style="list-style-type: none"> - La falta de sincronización automática entre Kyte, FACEL y POS aumenta el riesgo de errores humanos en la gestión de inventarios. - El 15% de las actualizaciones de inventarios presentan discrepancias, lo que afecta la planificación de la reposición de inventario y la toma de decisiones operativas. - Las tareas manuales duplicadas crean ineficiencia y aumentan los costos operativos. 		

Nota. Fuente: Elaboración propia.

Apéndice F.7. Plantilla de observación directa - OBS-007

Observación directa		OBS-007
Fecha	28 de septiembre de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Procesos de actualización de inventarios y ventas.		
Descripción de las actividades		
Durante las actividades de gestión de ventas y facturación, se observó que los empleados ingresaban manualmente los datos en los sistemas Kyte, FACEL y POS de manera independiente. Este proceso provocaba importantes discrepancias en la consolidación de datos, lo que ralentizaba la generación de reportes financieros y operativos.		
Observaciones clave		
<ul style="list-style-type: none"> - Los sistemas no están sincronizados automáticamente, lo que incrementa la probabilidad de errores y retrasa la generación de reportes. - Los empleados invierten un 15% adicional de su tiempo en corregir errores de sincronización y verificar datos. Esto significa que, de cada 100 actualizaciones diarias, aproximadamente 15 deben ser corregidas manualmente. - Estos problemas generan un costo adicional en horas laborales, que impacta los reportes financieros y aumenta la posibilidad de errores en la facturación. 		

Nota. Fuente: Elaboración propia.

Apéndice G.1. Plantilla de grupo focal - GF-001

Grupo focal		GF-001	
Fecha	25 de septiembre de 2024		
Moderador	Arquitecto de sistemas		
Participantes	Gestión de ventas minoristas, gestión de ventas mayoristas		
Temas discutidos			
Integración tecnológica desde cero			
Automatización total			
Eliminación de deuda técnica acumulada			
Pregunta	Respuesta		
¿Cómo un diseño de <i>software</i> nuevo podría mejorar la integración entre sistemas?	Un rediseño completo eliminaría las limitaciones técnicas actuales, permitiendo la interoperabilidad total entre los sistemas Kyte, FACEL y POS. Además, se podría automatizar la mayoría de los procesos clave, como la gestión de inventarios y la facturación, lo que reduciría los errores manuales y mejoraría la eficiencia operativa.		
Observaciones			
La deuda técnica existente en los sistemas actuales impide la integración y la automatización, lo que genera ineficiencias.			
Un nuevo diseño de <i>software</i> garantizaría la escalabilidad y la capacidad de adaptarse a futuros cambios o expansiones del negocio.			
La automatización total de los procesos críticos no solo mejorará la eficiencia interna, sino que reducirá costos operativos y aumentará la precisión de los reportes financieros.			

Nota. Fuente: Elaboración propia.

Apéndice G.2. Plantilla de grupo focal - GF-002

Grupo focal		GF-002	
Fecha	28 de septiembre de 2024		
Moderador	Arquitecto de sistemas		
Participantes	Gestión de ventas minoristas, gestión de ventas mayoristas		
Temas discutidos			
Herramientas tecnológicas Simplificación de procesos Eliminación de redundancias			
Pregunta	Respuesta		
¿Qué herramientas tecnológicas permitirán reducir la duplicación de tareas y mejorar la integración?	Las APIs de integración y el uso de plataformas en la nube se identificaron como las mejores soluciones para sincronizar datos entre los sistemas Kyte, FACEL y POS. Estas tecnologías permiten la comunicación en tiempo real entre sistemas, eliminando la necesidad de ingresar datos manualmente en varias plataformas.		
¿Cómo la simplificación de procesos puede mejorar la experiencia del cliente?	La automatización y la unificación de procesos mejorarán los tiempos de respuesta y mejorarán la precisión en la facturación. Esto, a su vez, incrementará la satisfacción del cliente al garantizar tiempos de entrega más rápidos y evitar errores en los pedidos.		
Observaciones			
La eliminación de tareas redundantes no solo mejorará la eficiencia interna, sino que también reducirá los tiempos de procesamiento y mejorará la experiencia del cliente final. La implementación de soluciones tecnológicas como middleware o plataformas en la nube permitirá una sincronización automática, mejorando la coherencia de los datos y reduciendo los errores humanos.			

Nota. Fuente: Elaboración propia.

Apéndice G.3. Plantilla de grupo focal - GF-003

Grupo focal		GF-003	
Fecha	30 de septiembre de 2024		
Moderador	Arquitecto de sistemas		
Participantes	Gestión de ventas minoristas, gestión de ventas mayoristas		
Temas discutidos			
Necesidad de un sistema centralizado			
Mejora de los flujos de trabajo			
Mejora en la precisión de los datos			
Pregunta	Respuesta		
¿Qué impacto tendría un sistema único de entrada de datos en la eficiencia operativa?	Un sistema único de entrada de datos centralizado eliminaría la duplicación de tareas, permitiendo que las actualizaciones de inventarios, ventas y facturación se sincronicen automáticamente en todos los sistemas. Esto mejoraría la precisión de los datos y reduciría el tiempo de procesamiento.		
¿Cómo se podría mejorar la toma de decisiones con un sistema automatizado?	La automatización en tiempo real de los inventarios y ventas permitiría a los responsables de las decisiones contar con información precisa y actualizada, lo que mejoraría la planificación de ventas, la gestión de inventarios y la facturación.		
Observaciones			
Un sistema centralizado no solo reducirá los errores manuales, sino que también mejorará los flujos de trabajo, mejorando la eficiencia y reduciendo el tiempo de respuesta en todas las áreas de operación.			
El rediseño de los sistemas debe enfocarse en la creación de una plataforma central que permita una visibilidad total de los datos en tiempo real.			

Nota. Fuente: Elaboración propia.

Apéndice H.1. Plantilla de lluvia de ideas - BI-001

Lluvia de ideas		BI-001
Fecha	30 de septiembre de 2024	
Facilitador	Arquitecto de sistemas	
Participantes	Administrador del negocio, responsable de ventas minoristas, responsable de ventas mayoristas	
Ideas generadas		
<p>Propuesta para utilizar soluciones en la nube que faciliten la comunicación entre los sistemas Kyte, FACEL y POS, lo que permitirá una integración flexible y escalable.</p> <p>Recomendación de herramientas de RPA para automatizar procesos repetitivos como la actualización de inventarios, facturación y la entrada de datos.</p> <p>Implementación de un sistema de entrada de datos único, sincronizado en tiempo real con todos los sistemas, para evitar la duplicación de tareas y mejorar la precisión de la información.</p> <p>Se destacó la importancia de las APIs para conectar sistemas, eliminando la duplicación de entradas de datos y mejorando la comunicación entre plataformas.</p>		
Conclusiones clave		
<p>Las soluciones SaaS y las APIs son esenciales para mejorar la integración de los sistemas en Hortiluma.</p> <p>El uso de middleware es necesario para conectar sistemas que no interactúan directamente.</p> <p>La automatización mediante RPA aumentará la eficiencia operativa y reducirá los errores humanos.</p>		

Nota. Fuente: Elaboración propia.

Apéndice H.2. Plantilla de lluvia de ideas - BI-002

Lluvia de ideas		BI-002
Fecha	1 de octubre de 2024	
Facilitador	Arquitecto de sistemas	
Participantes	Administrador del negocio, responsable de ventas minoristas, responsable de ventas mayoristas	
Ideas generadas		
<p>Propuesta para rediseñar los flujos de inventario mediante un sistema centralizado y automatizado que conecte todas las plataformas operativas de Hortiluma. Esto mejorará la visibilidad y el control del inventario en tiempo real.</p> <p>Mejorar la actualización de inventarios en tiempo real para una mayor precisión en los reportes y la planificación de ventas.</p> <p>Utilizar RPA para automatizar la generación de reportes y otras tareas repetitivas, reduciendo los errores humanos y permitiendo que los empleados se enfoquen en actividades estratégicas.</p> <p>Necesidad de migrar gradualmente los sistemas legados como POS y FACEL a plataformas modernas, utilizando soluciones híbridas para facilitar la transición sin interrupciones operativas.</p>		
Conclusiones clave		
<p>La migración gradual de los sistemas legados es esencial para eliminar la deuda técnica y facilitar la integración con nuevas tecnologías.</p> <p>Un sistema centralizado y automatizado mejorará la coherencia de los datos y la eficiencia en la planificación operativa.</p> <p>La automatización con RPA reducirá los errores humanos y mejorará la precisión de los reportes financieros y de inventario.</p>		

Nota. Fuente: Elaboración propia.

Apéndice H.3. Plantilla de lluvia de ideas - BI-003

Lluvia de ideas		BI-003
Fecha	3 de octubre de 2024	
Facilitador	Arquitecto de sistemas	
Participantes	Administrador del negocio, responsable de ventas minoristas, responsable de ventas mayoristas	
Ideas generadas		
<p>Implementación de middleware especializado para permitir la interoperabilidad entre las funcionalidades de los sistemas actuales de Hortiluma, evitando la necesidad de reemplazar completamente los sistemas legados.</p> <p>Se recomendó una estrategia de transición híbrida que combine tecnologías legadas con plataformas modernas, minimizando la interrupción de las operaciones durante la actualización de los sistemas.</p>		
Conclusiones clave		
<p>La adopción de middleware garantizará la interoperabilidad entre sistemas a corto plazo, evitando la duplicación de tareas y los errores en los reportes.</p> <p>El enfoque híbrido permitirá una transición fluida y reducirá el riesgo operativo durante las actualizaciones de sistemas.</p>		

Nota. Fuente: Elaboración propia.

Apéndice E.11. Plantilla de revisión documental - RD-011

Revisión documental		RD-011
Fecha	20 de septiembre de 2024	
Documento revisado	Documentación técnica de Kyte, FACEL, y POS; Reportes internos de inventarios	
Fuente	Documentación interna de Hortiluma y sitios web oficiales	
Objetivo de la revisión	Información clave	
Identificar las limitaciones técnicas y de interoperabilidad de los sistemas actuales.	<ul style="list-style-type: none"> - Kyte no se sincroniza con FACEL ni con el POS, requiriendo actualizaciones manuales que generan errores en inventarios y precios. - FACEL funciona con un catálogo de productos independiente, lo que provoca discrepancias en los precios e inventarios respecto a otros sistemas. - El POS es un sistema aislado, lo que dificulta la integración de los datos de ventas e inventarios, generando reportes inconsistentes. 	

Nota. Fuente: Elaboración propia.

Apéndice E.12. Plantilla de revisión documental - RD-012

Revisión documental		RD-012
Fecha	22 de septiembre de 2024	
Documento revisado	Informes de operaciones internas de inventarios y ventas	
Fuente	Reportes internos de inventarios de Hortiluma	
Objetivo de la revisión	Información clave	
Revisar los flujos de trabajo actuales relacionados con la actualización de inventarios y ventas para identificar redundancias y tareas duplicadas.	<ul style="list-style-type: none"> - Los empleados deben ingresar manualmente los datos en tres sistemas distintos (Kyte, FACEL, POS), lo que genera duplicaciones y discrepancias en los reportes. - Existen inconsistencias entre los niveles de inventario en FACEL y Kyte, lo que afecta la planificación de compras y provoca errores en los reportes financieros. - La falta de un sistema centralizado de inventarios genera retrasos y fallos en la gestión operativa. 	

Nota. Fuente: Elaboración propia.

Apéndice E.13. Plantilla de revisión documental - RD-013

Revisión documental		RD-013
Fecha	25 de septiembre de 2024	
Documento revisado	Documentación técnica de Kyte, FACEL, y POS; Reportes internos de inventarios	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Revisar los procedimientos de actualización de precios para identificar problemas de sincronización entre sistemas.	<ul style="list-style-type: none"> - Los precios se actualizan manualmente en FACEL y Kyte, lo que genera discrepancias significativas en los precios entre los canales de ventas. - La falta de sincronización automática de precios provoca retrasos en la actualización de productos en el inventario y en los sistemas de facturación, afectando la coherencia en las ventas. 	

Nota. Fuente: Elaboración propia.

Apéndice E.14. Plantilla de revisión documental - RD-014

Revisión documental		RD-014
Fecha	30 de septiembre de 2024	
Documento revisado	Reportes internos de inventarios y ventas	
Fuente	Base de datos de inventarios de Hortiluma	
Objetivo de la revisión	Información clave	
Identificar los problemas causados por la falta de sincronización en tiempo real de los inventarios entre sistemas y proponer mejoras en la gestión de inventarios.	<ul style="list-style-type: none"> - Discrepancias de hasta un 15% en los niveles de inventario entre Kyte y FACEL, afectando la planificación de ventas y disponibilidad de productos. - El proceso de sincronización manual provoca retrasos y afecta la coherencia de los reportes financieros y operativos. - No se cuenta con un mecanismo de verificación automática de inventarios en tiempo real para detectar inconsistencias y corregirlas de inmediato. 	

Nota. Fuente: Elaboración propia.

Apéndice E.15. Plantilla de revisión documental - RD-015

Revisión documental		RD-015
Fecha	2 de octubre de 2024	
Documento revisado	Reporte de inventarios	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión	Información clave	
Revisar los procedimientos de auditoría de inventarios para identificar oportunidades de automatización y mejorar la consistencia en la gestión de inventarios.	<ul style="list-style-type: none"> - La auditoría de inventarios es un proceso manual que se realiza trimestralmente, lo que genera retrasos en la identificación de errores en los datos. - No existen controles automatizados para verificar la consistencia de los datos entre Kyte, FACEL, y el POS. - El proceso actual carece de alertas automáticas para detectar y corregir errores de sincronización entre sistemas en tiempo real, lo que aumenta el riesgo de discrepancias. 	

Nota. Fuente: Elaboración propia.

Apéndice D.7. Plantilla de entrevista - ENT-007

Entrevista		ENT-007
Fecha	30 de septiembre de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Expectativas del sistema, problemas actuales de interoperabilidad, eliminación de duplicaciones, impacto en ventas y operatividad	
Participantes		
Rol dentro del proyecto	Administrador del negocio	
Pregunta	Respuesta	
¿Cuáles son los principales desafíos con el sistema actual?	La falta de integración entre Kyte, FACEL y el POS genera múltiples problemas operativos. Los empleados deben ingresar los mismos datos en cada sistema manualmente, lo que incrementa los errores y el tiempo dedicado a tareas duplicadas. Además, la ausencia de sincronización en tiempo real entre los sistemas causa discrepancias en los reportes de inventarios y ventas.	
¿Qué mejoras esperas de un sistema centralizado e integrado?	Un sistema que permita la sincronización automática y en tiempo real de inventarios, ventas y precios reduciría la carga manual y mejoraría la eficiencia operativa. También evitaría errores en los reportes y mejoraría la experiencia del cliente, al asegurar que la disponibilidad de productos refleje con precisión el inventario.	
¿Cómo afecta la falta de sincronización a la experiencia del cliente y la planificación?	La falta de sincronización entre sistemas genera problemas como productos agotados que aparecen disponibles en el sistema, afectando la satisfacción del cliente y provocando pérdidas en ventas potenciales. Además, las discrepancias en inventarios complican la planificación, lo que puede llevar a desabastecimientos o excesos de inventario.	
¿Qué impacto tienen los errores de interoperabilidad en la operatividad diaria?	Los errores en la actualización manual de inventarios, la falta de coherencia entre sistemas y las discrepancias en los reportes afectan negativamente la toma de decisiones, incrementan los costos operativos y limitan la capacidad del equipo técnico para concentrarse en tareas más estratégicas.	
Observaciones		
<p>La automatización de los procesos de actualización de inventarios y ventas es crucial para mejorar la eficiencia y reducir los errores.</p> <p>Se necesita un sistema modular e integrado que permita la fácil interoperabilidad entre Kyte, FACEL y POS, reduciendo la carga manual y eliminando la deuda técnica acumulada en el sistema POS.</p>		

Nota. Fuente: Elaboración propia.

Apéndice D.8. Plantilla de entrevista - ENT-008

Entrevista		ENT-008
Fecha	1 de octubre de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Impacto de las duplicaciones de tareas, mejoras esperadas, planificación y gestión de inventarios	
Participantes		
Rol dentro del proyecto	Responsable de ventas mayoristas	
Pregunta	Respuesta	
¿Cómo afectan las tareas duplicadas a la eficiencia en ventas e inventarios?	La duplicación de tareas afecta principalmente la eficiencia operativa. Los empleados actualizan manualmente los inventarios en tres sistemas distintos (Kyte, FACEL, y POS), lo que genera inconsistencias en los niveles de inventario y provoca errores en los reportes de ventas y planificación de inventarios.	
¿Qué expectativas tienes de un sistema integrado y automatizado?	La sincronización en tiempo real entre los sistemas de ventas e inventarios reduciría los errores en la planificación de pedidos y mejoraría la coherencia de los datos entre sucursales. La automatización permitiría que la gestión de inventarios sea más precisa.	
¿Cómo ha afectado la falta de integración a la operatividad diaria?	La operatividad diaria se ve afectada por las constantes correcciones de errores de inventarios, lo que genera pérdida de tiempo y recursos. Además, la falta de coherencia entre los sistemas mayoristas y minoristas crea problemas en la gestión de grandes volúmenes de pedidos.	
Observaciones		
Un sistema que automatice la actualización de inventarios y sincronice en tiempo real los datos es esencial para evitar errores en la planificación y mejorar la coherencia operativa. La automatización de procesos es fundamental para reducir la duplicación de tareas y mejorar la precisión en los reportes financieros y de ventas.		

Nota. Fuente: Elaboración propia.

Apéndice D.9. Plantilla de entrevista - ENT-009

Entrevista		ENT-009
Fecha	3 de octubre de 2024	
Entrevistador	Arquitecto de sistemas	
Aspectos clave	Problemas técnicos de interoperabilidad, automatización y reducción de errores técnicos	
Participantes		
Rol dentro del proyecto	Administrador del negocio	
Pregunta	Respuesta	
¿Cómo afectan las tareas duplicadas a la eficiencia en ventas e inventarios?	La duplicación de tareas afecta principalmente la eficiencia operativa. Los empleados actualizan manualmente los inventarios en tres sistemas distintos (Kyte, FACEL, y POS), lo que genera inconsistencias en los niveles de inventario y provoca errores en los reportes de ventas y planificación de inventarios.	
¿Qué expectativas tienes de un sistema integrado y automatizado?	La sincronización en tiempo real entre los sistemas de ventas e inventarios reduciría los errores en la planificación de pedidos y mejoraría la coherencia de los datos entre sucursales. La automatización permitiría que la gestión de inventarios sea más precisa.	
¿Cómo ha afectado la falta de integración a la operatividad diaria?	La operatividad diaria se ve afectada por las constantes correcciones de errores de inventarios, lo que genera pérdida de tiempo y recursos. Además, la falta de coherencia entre los sistemas mayoristas y minoristas crea problemas en la gestión de grandes volúmenes de pedidos.	
Observaciones		
Un sistema que automatice la actualización de inventarios y sincronice en tiempo real los datos es esencial para evitar errores en la planificación y mejorar la coherencia operativa. La automatización de procesos es fundamental para reducir la duplicación de tareas y mejorar la precisión en los reportes financieros y de ventas.		

Nota. Fuente: Elaboración propia.

Apéndice F.8. Plantilla de observación directa - OBS-008

Observación directa		OBS-008
Fecha	15 de agosto de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Gestión de inventarios, ventas, facturación y monitoreo de información.		
Descripción de las actividades		
<p>Los empleados deben ingresar manualmente los datos de inventarios, ventas y facturación en tres sistemas distintos (Kyte, FACEL, y POS), lo que provoca duplicaciones de tareas y retrasos en la disponibilidad de información. Este proceso genera inconsistencias en los reportes de inventarios y ventas, afecta la precisión de la facturación, y provoca errores frecuentes en la conciliación de datos financieros. Durante las observaciones, se identificaron discrepancias en los niveles de inventario y errores en la actualización de precios y facturas, lo que impacta negativamente en la planificación operativa y financiera.</p>		
Observaciones clave		
<p>La falta de integración entre los sistemas genera un alto nivel de duplicación de tareas, lo que afecta la coherencia de los datos entre plataformas y provoca errores operativos. La actualización manual de inventarios y precios, y la conciliación de facturas sin automatización, afecta tanto los reportes de ventas como la gestión de inventarios y la planificación de compras.</p> <p>La falta de sincronización en tiempo real complica la toma de decisiones basadas en datos precisos, afectando la operatividad del negocio y la satisfacción del cliente.</p>		

Nota. Fuente: Elaboración propia.

Apéndice F.9. Plantilla de observación directa - OBS-009

Observación directa		OBS-009
Fecha	18 de agosto de 2024	
Observador	Arquitecto de sistemas	
Lugar	Oficinas centrales de Hortiluma	
Aspectos observados		
Actualización de inventarios, flujos de trabajo en ventas y monitoreo de reportes.		
Descripción de las actividades		
Se observó que los empleados de Hortiluma gestionan manualmente la actualización de inventarios, ventas y facturación en tres sistemas distintos (Kyte, FACEL, y POS), lo que genera inconsistencias entre los datos. La actualización de precios y niveles de inventario no se realiza en tiempo real, lo que provoca retrasos y errores en los reportes de inventarios y ventas. Además, el monitoreo de estas actividades carece de un sistema automatizado, lo que retrasa la disponibilidad de información precisa para la toma de decisiones.		
Observaciones clave		
La duplicación de tareas es evidente en la actualización manual de inventarios, lo que genera incoherencias en los reportes de ventas y facturación. La falta de sincronización en tiempo real entre sistemas afecta la precisión de los reportes financieros y operativos, lo que impacta directamente en la gestión de inventarios y la experiencia del cliente. Se requiere un sistema de monitoreo automatizado que permita obtener información en tiempo real y reduzca los errores operativos.		

Nota. Fuente: Elaboración propia.

Apéndice G.4. Plantilla de grupo focal - GF-004

Grupo focal		GF-004
Fecha	25 de agosto de 2024	
Moderador	Arquitecto de sistemas	
Participantes	Administrador del negocio, responsable de ventas minoristas, responsable de ventas mayoristas	
Temas discutidos		
<p>Se discutió la falta de integración entre Kyte, FACEL, y POS, y cómo esto afecta la coherencia de los datos, generando errores en la planificación de inventarios y ventas. La interoperabilidad entre estos sistemas es crítica para eliminar la duplicación de tareas manuales y asegurar que los datos fluyan sin problemas entre plataformas. La automatización se identificó como un componente clave para reducir la intervención manual en tareas repetitivas como la actualización de inventarios, precios, y la generación de reportes. Los participantes destacaron la importancia de implementar tecnologías como RPA para automatizar estas operaciones, aumentando la eficiencia operativa. La falta de sincronización en tiempo real provoca errores en los reportes de ventas e inventarios, afectando la planificación diaria y la toma de decisiones. Un sistema que permita la sincronización automática y en tiempo real de datos entre los tres sistemas (Kyte, FACEL, POS) es esencial para mejorar la precisión y evitar errores humanos. El grupo enfatizó que el sistema debe ser modular y escalable, permitiendo la integración de nuevos módulos o funcionalidades en el futuro sin comprometer la interoperabilidad o la eficiencia operativa.</p>		
Conclusiones clave		
<ul style="list-style-type: none"> - Los participantes coincidieron en que un sistema modular que sincronice en tiempo real los datos de ventas e inventarios es esencial para mejorar la coherencia de los reportes y la eficiencia operativa. - La duplicación de tareas y la falta de interoperabilidad entre los sistemas deben resolverse mediante la integración de APIs y middleware, y un sistema que automatice los procesos críticos. - El diseño debe ser escalable, garantizando que la solución no solo resuelva los problemas actuales, sino que esté preparada para futuras expansiones o cambios en los procesos operativos. 		

Nota. Fuente: Elaboración propia.

Apéndice H.4. Plantilla de lluvia de ideas - LLUV-004

Lluvia de ideas		LLUV-004
Fecha	1 de septiembre de 2024	
Facilitador	Arquitecto de sistemas	
Participantes	Administrador del negocio, responsable de ventas minoristas, responsable de ventas mayoristas	
Ideas generadas		
<p>Se discutió la necesidad de un sistema modular que permita la integración de los sistemas de ventas, inventarios y facturación. Esta arquitectura modular facilitaría la interoperabilidad entre Kyte, FACEL, y POS, eliminando la necesidad de múltiples entradas de datos y minimizando los errores operativos.</p> <p>Los participantes señalaron que es crucial implementar la sincronización en tiempo real entre los sistemas para garantizar la coherencia de los datos. Esto evitaría problemas de actualización de inventarios y precios, y mejoraría la precisión en los reportes financieros y operativos.</p> <p>Se destacó la importancia de automatizar las tareas repetitivas, como la actualización de inventarios y precios, la generación de reportes y la facturación. La automatización reduciría la carga de trabajo manual, eliminando la duplicación de tareas y mejorando la eficiencia.</p> <p>La estructura propuesta debe ser escalable, permitiendo la adición de nuevos módulos y funcionalidades en el futuro sin afectar la operatividad actual. Esto garantizaría que el sistema se pueda adaptar a las necesidades futuras del negocio sin interrupciones.</p>		
Conclusiones clave		
<ul style="list-style-type: none"> - El nuevo sistema debe enfocarse en eliminar las tareas duplicadas a través de la integración de sistemas y la sincronización en tiempo real. - La automatización de procesos es fundamental para mejorar la eficiencia y reducir errores operativos. - El sistema debe ser escalable y flexible para permitir futuras expansiones y garantizar una operación fluida a medida que el negocio crece. 		

Nota. Fuente: Elaboración propia.

Apéndice I.1. Plantilla de historias de usuario - HU-001

Gestión de Productos		HU-001	
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema. El sistema debe contar con una base de datos de productos inicializada. 		
Postcondiciones	<ul style="list-style-type: none"> Los productos se crean, actualizan o eliminan, y la base de datos de inventarios se actualiza en tiempo real. Los cambios en los productos se reflejan automáticamente en todos los canales de venta. 		
Escenario principal		Flujo alternativo	
El usuario selecciona la opción de gestión de productos. Se registra la información del producto (nombre, descripción, precio, categoría). El sistema verifica si el producto ya existe. Si no existe, se guarda en la base de datos y se sincroniza en tiempo real con todas las plataformas. Se confirma la creación del producto.		Si el producto ya existe, el sistema ofrece la opción de actualizar la información o eliminarlo.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Añadir un nuevo producto al sistema, reflejando el cambio en todos los canales.	Consultar los detalles de un producto existente y su disponibilidad en tiempo real.	Modificar la información de un producto, ajustando el precio automáticamente en todas las plataformas.	Remover un producto de la base de datos y actualizar el inventario en todos los canales.
Requerimientos cubiertos	REQF- 1, REQF- 2, REQF- 5, REQF- 6, REQNF- 1, REQNF- 6, REQNF- 9		

Nota. Fuente: Elaboración propia.

Apéndice I.2. Plantilla de historias de usuario - HU-002

Gestión de Inventarios			HU-002
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> El usuario debe tener permisos para actualizar el inventario. La base de datos de inventarios debe estar operativa. 		
Postcondiciones	<ul style="list-style-type: none"> El inventario se actualiza automáticamente y los cambios son sincronizados con las plataformas. Se generan alertas automáticas cuando el inventario alcanza umbrales mínimos. 		
Escenario principal		Flujo alterno	
El usuario accede a la gestión de inventarios. Registra la cantidad inicial o actualiza el inventario de un producto. El sistema guarda los cambios en la base de datos de inventarios. Si el inventario es bajo, se genera una alerta automática para la compra.		Si el inventario no puede actualizarse, el sistema muestra un mensaje de error y solicita corregir la entrada.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Añadir un nuevo registro de inventario.	Consultar el estado de inventarios y el historial de cambios.	Modificar la cantidad de productos en inventario, generando alertas si es necesario.	Remover un registro de inventario obsoleto.
Requerimientos cubiertos	REQF- 3, REQF- 4, REQF- 5, REQF- 7, REQF- 8, REQNF- 2, REQNF- 7, REQNF- 10		

Nota. Fuente: Elaboración propia.

Apéndice I.3. Plantilla de historias de usuario - HU-003

Realización de Venta		HU-003	
Actores	Usuario (Administración del negocio, Dirección)		
Precondiciones	<ul style="list-style-type: none"> El cliente debe estar registrado en el sistema. El inventario debe estar actualizado con la disponibilidad de productos. 		
Postcondiciones	<ul style="list-style-type: none"> La venta queda registrada, y el inventario de los productos vendidos se reduce automáticamente. Se actualiza la información de ventas en tiempo real para todos los reportes. 		
Escenario principal		Flujo alterno	
El usuario selecciona los productos que el cliente desea comprar. Se ingresa la cantidad de cada producto. El sistema verifica la disponibilidad de los productos. Si hay suficiente inventario, la venta se confirma y el inventario se actualiza. Se genera un recibo para el cliente.		Si no hay suficiente inventario, el sistema sugiere cantidades disponibles o productos alternativos.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Registrar una nueva venta y actualizar el inventario automáticamente.	Consultar ventas anteriores y detalles de transacciones.	Modificar detalles de una venta antes de finalizarla.	Cancelar una venta registrada y restaurar el inventario del inventario.
Requerimientos cubiertos	REQF- 5, REQF- 8, REQF- 9, REQF- 13, REQNF- 1, REQNF- 4, REQNF- 9		

Nota. Fuente: Elaboración propia.

Apéndice I.4. Plantilla de historias de usuario - HU-004

Generación de Facturas			HU-004
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> • Debe existir una venta registrada para generar una factura. • El sistema debe estar configurado para emitir facturas electrónicas. 		
Postcondiciones	<ul style="list-style-type: none"> • La factura se genera y se envía al cliente en formato PDF. • Los registros de facturación se actualizan en el sistema para auditoría. 		
Escenario principal		Flujo alternativo	
El usuario accede al módulo de facturación. Se selecciona la venta para la cual se desea generar la factura. El sistema compila los datos de la venta y crea un reporte detallado. Se genera un archivo PDF de la factura y se envía al cliente.		Si el cliente desea una modificación, se ajustan los detalles antes de generar la factura.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Generar una nueva factura para una venta.	Consultar facturas emitidas y su estado.	Modificar detalles antes de la generación de la factura.	Anular una factura si la venta es cancelada.
Requerimientos cubiertos	REQF- 5, REQF- 10, REQF- 11, REQNF- 3, REQNF- 6, REQNF- 8		

Nota. Fuente: Elaboración propia.

Apéndice I.5. Plantilla de historias de usuario - HU-005

Gestión de Reportes			HU-005
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> • Deben existir datos de ventas e inventarios para generar reportes. • El usuario debe tener permisos para generar reportes. 		
Postcondiciones	<ul style="list-style-type: none"> • El reporte se genera y se almacena para futuras consultas. • Los reportes se pueden exportar en formatos PDF y Excel. 		
Escenario principal		Flujo alterno	
El usuario selecciona el tipo de reporte a generar (ventas, inventarios, etc.). Se definen los parámetros de tiempo y detalle. El sistema genera el reporte en formato gráfico y tabular. El reporte puede exportarse a PDF o Excel.		Si los parámetros de tiempo son incorrectos, el sistema solicita ajustes.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Generar un nuevo reporte.	Visualizar reportes generados anteriormente.	Cambiar los criterios de un reporte existente.	Borrar un reporte innecesario.
Requerimientos cubiertos	REQF- 3, REQF- 14, REQNF- 5, REQNF- 10		

Nota. Fuente: Elaboración propia.

Apéndice I.6. Plantilla de historias de usuario - HU-006

Gestión de Clientes			HU-006
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> El cliente debe proporcionar sus datos para ser registrado. El usuario debe tener permisos para la gestión de clientes. 		
Postcondiciones	<ul style="list-style-type: none"> El cliente queda registrado o actualizado en el sistema, y los datos se reflejan en los módulos de ventas y facturación. Se actualiza la información de contacto y preferencias de manera consistente en todas las plataformas. 		
Escenario principal		Flujo alterno	
El usuario registra los datos personales del cliente. Se verifica que el cliente no exista previamente. Si no existe, el sistema guarda los datos. Se genera una confirmación del registro.		Si el cliente ya existe, el sistema ofrece la opción de actualizar los datos.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Registrar un nuevo cliente.	Consultar información de un cliente.	Modificar los datos de contacto de un cliente.	Eliminar un registro de cliente y sus datos asociados.
Requerimientos cubiertos	REQF- 2, REQF- 5, REQF- 6, REQF- 11, REQNF- 6, REQNF- 9		

Nota. Fuente: Elaboración propia.

Apéndice I.7. Plantilla de historias de usuario - HU-007

Integración con API Externas		HU-007	
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> El sistema debe estar configurado para la conexión con APIs externas. Las credenciales de acceso a la API deben ser válidas y estar actualizadas. 		
Postcondiciones	<ul style="list-style-type: none"> Los datos se sincronizan correctamente con las plataformas externas, manteniendo la integridad de la información en todos los sistemas. Se genera un registro detallado de la actividad de sincronización para auditoría. 		
Escenario principal		Flujo alterno	
El usuario selecciona la plataforma con la cual desea sincronizar datos. Se configuran los parámetros de la API (token, URL, etc.). El sistema envía y recibe datos desde la API. Se genera un reporte de sincronización exitosa.		Si la sincronización falla, el sistema muestra un mensaje de error y sugiere correcciones.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Configurar una nueva integración.	Consultar registros de sincronización y logs de errores.	Modificar los parámetros de integración.	Eliminar una integración con una API obsoleta o que ya no es necesaria.
Requerimientos cubiertos	REQF- 4, REQF- 5, REQF- 10, REQNF- 2, REQNF- 9		

Nota. Fuente: Elaboración propia.

Apéndice I.8. Plantilla de historias de usuario - HU-008

Migración de Datos			HU-008
Actores	Usuario		
Precondiciones	<ul style="list-style-type: none"> Los datos legados deben estar preparados para la migración en formatos compatibles. El usuario debe tener los permisos necesarios para realizar la migración. 		
Postcondiciones	<ul style="list-style-type: none"> Los datos antiguos se integran al sistema sin pérdida de información, manteniendo la integridad de los registros históricos. Se genera un reporte detallado de la migración, incluyendo posibles errores y registros migrados con éxito. 		
Escenario principal		Flujo alterno	
El usuario selecciona la opción de migración de datos. Se cargan los archivos o bases de datos antiguos. El sistema verifica la integridad de los datos y su compatibilidad con el nuevo sistema. Se completan los procesos de migración y se valida la consistencia de la información.		Si hay errores en los datos, el sistema muestra un log detallado para corrección y permite volver a intentar la carga.	
Escenarios de CRUD			
Crear	Leer	Actualizar	Eliminar
Iniciar un nuevo proceso de migración de datos.	Consultar el estado y log de los procesos de migración realizados.	Reintentar procesos de migración tras corrección de errores.	Descartar datos migrados erróneamente o anular una migración incompleta.
Requerimientos cubiertos	REQF- 12, REQF- 14, REQNF- 4, REQNF- 5, REQNF- 10		

Nota. Fuente: Elaboración propia.

Apéndice E.16. Plantilla de revisión documental - RD-016

Revisión documental		RD-016
Fecha	15 de octubre de 2024	
Documento revisado	Especificación técnica de la arquitectura de integración	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Validar la correcta implementación de los principios de arquitectura de microservicios y modularidad		Confirmación de que la arquitectura se ha dividido en microservicios independientes que garantizan escalabilidad y robustez.

Nota. Fuente: Elaboración propia.

Apéndice E.17. Plantilla de revisión documental - RD-017

Revisión documental		RD-014
Fecha	16 de octubre de 2024	
Documento revisado	Diseño preliminar de APIs y middleware	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Validar que las APIs y middlewares implementados garanticen la interoperabilidad entre los sistemas de Hortiluma		Las APIs REST se integran correctamente con los sistemas existentes, asegurando la correcta comunicación entre módulos y plataformas externas.

Nota. Fuente: Elaboración propia.

Apéndice E.18. Plantilla de revisión documental - RD-018

Revisión documental		RD-018
Fecha	17 de octubre de 2024	
Documento revisado	Historias de usuario de automatización de procesos	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Confirmar que los procesos automatizados de actualización de inventarios y precios funcionan correctamente		La automatización de los procesos de inventarios y precios ha sido implementada con éxito y se ejecuta sin intervención manual.

Nota. Fuente: Elaboración propia.

Apéndice E.19. Plantilla de revisión documental - RD-019

Revisión documental		RD-019
Fecha	18 de octubre de 2024	
Documento revisado	Matriz de cobertura de requisitos	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Validar que todos los requisitos funcionales y técnicos han sido cubiertos en el diseño de la solución propuesta		Se ha verificado que todos los requisitos de interoperabilidad, automatización y escalabilidad han sido correctamente implementados.

Nota. Fuente: Elaboración propia.

Apéndice E.20. Plantilla de revisión documental - RD-020

Revisión documental		RD-017
Fecha	19 de octubre de 2024	
Documento revisado	Plan de pruebas de historias de usuario	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Validar que los historias de usuario incluidos en las pruebas aseguran el correcto funcionamiento del sistema en situaciones reales		Las pruebas realizadas incluyen escenarios clave para la validación del sistema en condiciones reales, tales como la integración de APIs, automatización y resiliencia del sistema.

Nota. Fuente: Elaboración propia.

Apéndice E.21. Plantilla de revisión documental - RD-021

Revisión documental		RD-021
Fecha	20 de octubre de 2024	
Documento revisado	Análisis de riesgos y plan de mitigación	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Validar que el análisis de riesgos ha considerado todos los posibles puntos de fallo y que existen planes de mitigación robustos		Los riesgos relacionados con la interoperabilidad y la automatización están bien documentados, y existen medidas claras para minimizar impactos en caso de fallo.

Nota. Fuente: Elaboración propia.

Apéndice E.22. Plantilla de revisión documental - RD-022

Revisión documental		RD-022
Fecha	21 de octubre de 2024	
Documento revisado	Informe final de validación de diseño	
Fuente	Documentación interna de Hortiluma	
Objetivo de la revisión		Información clave
Confirmar que el diseño final validado cumple con todos los requisitos y está listo para la implementación en condiciones reales		El diseño final ha sido validado en condiciones reales y cumple con los requisitos técnicos y funcionales definidos en las fases anteriores.

Nota. Fuente: Elaboración propia.