



Escuela de Administración de Tecnologías de Información

**Propuesta de modernización de arquitectura del software Gestemed en Key
Tech**

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

Elaborado por:

Dayana Alexandra Hernández Hernández

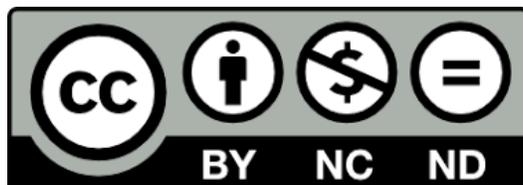
Prof. Tutor:

Máster. Luis Javier Chavarría Sánchez

Cartago, Costa Rica

II Semestre

Julio, 2024.



Esta obra está bajo una Licencia Creative Commons ReconocimientoNoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0). Para consultar una copia de esta licencia, ingrese al siguiente enlace:

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

Hoja de Aprobación

Posterior a la aprobación de su defensa Usted deberá recopilar las firmas y adjuntar la hoja de aprobación en esta sección del documento final.

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN

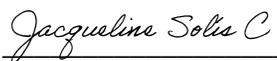
GRADO ACADÉMICO: LICENCIATURA

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

Máster. Luis Javier Chavarría Sánchez
Profesor tutor



Mónica Ureña Cordero
Lectora externa



Jacqueline Solís Céspedes
Lectora académica

Yarima Sandoval Sánchez
Coordinadora de Trabajo Final de Graduación

Resumen

El proyecto de Trabajo Final de Graduación consiste en la propuesta de modernización del software Gestemed para la gestión de equipos médicos. Este software, desarrollado en 2007, ha quedado obsoleto debido a su incompatibilidad con tecnologías actuales. El objetivo principal de este proyecto es proponer una actualización integral de su arquitectura, alineada con las mejores prácticas de la industria y estándares tecnológicos emergentes. Para ello, se realizó un análisis FODAL (Fortalezas, Oportunidades, Debilidades, Amenazas, Limitaciones) del sistema actual, se definieron los requerimientos funcionales y no funcionales, y se seleccionaron las tecnologías más adecuadas para su implementación futura.

Palabras clave: Modernización de software, arquitectura de software, gestión de equipos médicos, FODAL, tecnologías emergentes. este es el resumen

Abstract

The Final Graduation Project focuses on the proposal to modernize the Gestemed software for managing medical equipment. This software, developed in 2007, has become obsolete due to its incompatibility with current technologies. The main objective of this project is to propose a comprehensive upgrade of its architecture, aligned with industry best practices and emerging technology standards. To achieve this, a FODAL (Strengths, Opportunities, Weaknesses, Threats, Limitations) analysis of the current system was conducted, functional and non-functional requirements were defined, and the most suitable technologies were selected for future implementation.

Keywords: *Software modernization, software architecture, medical equipment management, FODAL, emerging technologies.*

Tabla de Contenidos

Hoja de Aprobación	3
Resumen.....	4
Abstract	5
1. Introducción.....	1
1.1. Descripción General.....	2
1.2. Antecedentes	3
1.2.1. Descripción de la organización.....	3
Misión.....	3
Visión.....	3
Valores.....	3
1.2.1.1. Equipo de trabajo	4
1.2.2. Trabajos similares realizados dentro y fuera de la organización	5
1.2.2.1. Interno	5
1.2.2.2. Externo	6
1.3. Planteamiento del problema.....	7
1.3.1. Situación problemática.....	7
1.3.2. Justificación del proyecto	12
1.3.3. Beneficios esperados o aportes del Trabajo Final de Graduación	13
1.3.3.1. Beneficios directos.....	13
1.3.3.2. Beneficios indirectos.....	13
1.4. Objetivos del Trabajo Final de Graduación	15
1.4.1. Objetivo General.....	15
1.4.2. Objetivos específicos.....	15
1.5. Alcance	16
1.6. Entregables.....	17
1.7. Supuestos	17
1.8. Limitaciones.....	18
1.9. Exclusiones	18

2.	Marco conceptual	19
2.1.	Requerimientos de software.....	19
2.1.1.	Definición de requerimiento	19
2.1.2.	Tipos de requerimientos.....	19
2.1.3.	Problemas comunes relacionados con los requerimientos.....	21
2.1.4.	Desarrollo de requerimientos.....	21
2.1.5.	Priorización de requerimientos	24
2.1.6.	Gestión de requerimientos	25
2.2.	Arquitectura de software.....	27
2.2.1.	Definición de arquitectura de software.....	27
2.2.2.	Principios fundamentales de la arquitectura de software.....	27
2.2.3.	Características arquitectónicas.....	28
2.2.4.	Patrones arquitectónicos	28
2.2.5.	Toma de decisiones arquitectónicas y compromisos	29
2.2.6.	Evaluación de riesgos arquitectónicos	29
2.2.7.	Componentes en arquitectura de software	29
2.2.8.	Atributos de calidad de una arquitectura de software.....	32
2.2.9.	Patrones de integración de arquitectura de software.....	34
2.3.	Modernización de software.....	39
2.3.1.	Definición de modernización de software	39
2.3.2.	Importancia de las buenas prácticas.....	39
2.3.3.	Proceso de Modernización	40
2.4.	Estándares	41
2.5.	Gestión del Cambio.....	43
2.6.	Planificación del soporte técnico	44
3.	Marco metodológico.....	46
3.1.	Tipo de investigación.....	46
3.2.	Enfoque de la investigación	47
3.3.	Diseño de la investigación	48
3.4.	Tipos de Fuentes de Información.....	49
3.4.1.	Fuentes primarias	49
3.3.2.	Fuentes secundarias	50

3.5.	Sujetos de Investigación	51
3.6.	VARIABLES DE INVESTIGACIÓN	53
3.7.	Técnicas e instrumentos de recolección de datos	58
3.8.	Procedimiento metodológico de la investigación	60
3.8.1.	Fase 1: Revisión documental y análisis del contexto actual	60
3.8.2.	Fase 2: Elicitación y análisis de requerimientos	60
3.8.3.	Fase 3: Análisis de alternativas tecnológicas	60
3.8.4.	Fase 4: Diseño de la propuesta de modernización	61
3.8.5.	Fase 5: Validación y refinamiento de la propuesta	61
3.8.6.	Fase 6: Documentación y entrega final	61
3.9.	Operacionalización de las variables o categorías	62
3.10.	Tabla resumen del procedimiento metodológico de la investigación	64
4.	Análisis de Resultados	65
4.1.	Fase 1: Revisión documental y análisis contextual	66
4.1.1.	Resultados guía de entrevista	66
4.1.2.	Resultados guía de satisfacción	68
4.1.3.	Resultados de <i>benchmarking</i>	70
4.1.4.	Resultados del PESTEL	72
4.1.5.	FODAL	73
4.2.	Fase 2: Elicitación y análisis de requerimientos	82
4.2.1.	Resultados guía para taller de elicitación de requerimientos	82
4.2.2.	Requerimientos	84
4.2.3.	Priorización de requerimientos	90
4.3.	Fase 3: Análisis de alternativas tecnológicas	97
4.3.1.	Resultados de la revisión documental	97
4.3.2.	Principios fundamentales de la arquitectura de software	99
4.3.2.1.	Componentes críticos de la arquitectura de software	101
5.	Propuesta de solución	103
5.1.	Fase 4: Diseño de la propuesta de modernización	103
5.1.1.	Estructura de la propuesta de modernización	103
Diagrama de Arquitectura Propuesto para Gestemed		107
5.1.2.	Compatibilidad con los requerimientos del FODAL	113

5.1.2.1.	Resultados del mapa de alineación de requerimientos con el FODAL	113
5.1.3.	Uso de buenas prácticas de la industria	115
5.1.3.1.	Resultados Lista de verificación de buenas prácticas	115
5.2.	Análisis de la viabilidad de la propuesta.....	117
6.	Conclusiones.....	118
7.	Recomendaciones	120
8.	Referencias	123
9.	Apéndices	127
A.	Guía para taller de elicitación de requerimientos	127
B.	Guía de entrevista	131
C.	Respuestas de entrevista	131
D.	Guía de satisfacción	131
E.	Respuestas de la guía de satisfacción.....	131
F.	Lista de verificación de buenas prácticas.....	131
G.	Revisión documental.....	131
H.	<i>Benchmarking</i>	131
I.	Mapa de alineación de requerimientos	132
J.	PESTEL	135
K.	Informe viabilidad.....	136
	Descripción del proyecto.....	136
	Análisis costo-beneficio	136
	Riesgos y mitigación	137
10.	Anexos	139
10.1.	Anexo I Carta de aprobación filológica	139
10.2.	Anexo II Carta de aprobación de minutas de reunión de TFG.....	140
10.3.	Anexo III Plantilla de minutas	141
10.4.	Anexo IV Minuta uno con contraparte.....	142
10.5.	Anexo V Minuta dos con contraparte	143
10.6.	Anexo VI Minuta tres con contraparte.....	144
10.7.	Anexo VII Salarios mínimos según MTSS.....	145
	146	
10.8.	Anexo VIII Primera evaluación de la contraparte.....	147

10.9. Anexo IX Segunda evaluación de la contraparte 147
10.10. Anexo X Tercera evaluación de la contraparte 147

Índice de Figuras

Figura 1 4
Figura 2 8
Figura 3 10
Figura 4 107

Índice de Tablas

Tabla 1	5
Tabla 2	17
Tabla 3	49
Tabla 4	50
Tabla 5	51
Tabla 6	53
Tabla 7	55
Tabla 8	56
Tabla 9	57
Tabla 10	58
Tabla 11	62
Tabla 12	64
Tabla 13	73
Tabla 14	74
Tabla 15	76
Tabla 16	78
Tabla 17	80
Tabla 18	84
Tabla 19	88
Tabla 20	90
Tabla 21	93
Tabla 22	95
Tabla 23	96
Tabla 24	99
Tabla 25	101

1. Introducción

Este capítulo presenta la propuesta de modernización de la arquitectura del software Gestemed en Key Tech, abordando diversos aspectos fundamentales para su desarrollo. Inicia con una descripción general, que proporciona un panorama del proyecto y sus objetivos principales. Posteriormente, se detallan los antecedentes, además, se incluye una descripción de la organización con su misión, visión, valores y una explicación del equipo de trabajo involucrado. Por último, se examinan trabajos similares realizados dentro y fuera de la organización, tanto a nivel interno como externo, ofreciendo un contexto para la iniciativa actual.

El capítulo continúa con el planteamiento del problema, donde se analiza la situación problemática que motiva la modernización del software. Se justifica la necesidad del proyecto y se describen los beneficios esperados o aportes, tanto directos como indirectos, que se anticipan como resultado del trabajo. Seguidamente, se establecen los objetivos del Trabajo Final de Graduación (TFG), tanto el objetivo general como los objetivos específicos que guiarán el desarrollo del proyecto.

Se describe detalladamente el alcance del proyecto, que define los límites y expectativas del trabajo a realizar, excluyendo la implementación del software en esta fase. Finalmente, se enumeran los entregables del proyecto, los supuestos bajo los cuales se desarrollará y las limitaciones que podrían afectar su ejecución. Este capítulo proporciona una guía integral para la planificación y futura modernización del software Gestemed, asegurando que esté alineada con las necesidades operativas y estratégicas de Key Tech.

1.1. Descripción General

La modernización y actualización de software en el ámbito empresarial es un tema relevante en la actualidad, especialmente en organizaciones que buscan mejorar su eficiencia operativa, satisfacer las demandas del mercado y fortalecer su competitividad. En este contexto, el presente trabajo se enfoca en la planificación detallada de la modernización del software Gestemed, con el objetivo de desarrollar un plan exhaustivo que guíe la mejora de la gestión de equipos médicos, aumente la satisfacción del cliente y fortalezca la competitividad de la empresa en el mercado.

El software Gestemed, desarrollado en 2007, ha demostrado ser una herramienta valiosa para la gestión de equipos médicos. Sin embargo, su arquitectura desactualizada y su incompatibilidad con tecnologías modernas han generado limitaciones significativas en su funcionalidad y eficiencia. Esta situación ha motivado la necesidad de una modernización integral del software, que permita aprovechar las innovaciones tecnológicas actuales y mejorar la experiencia de los usuarios.

En este sentido, el presente trabajo constituye un plan de modernización integral del software Gestemed en la empresa Key Tech, con el objetivo de desarrollar un plan detallado que guíe la mejora de la gestión de equipos médicos, aumente la satisfacción del cliente y fortalezca la competitividad de la empresa en el mercado. El proceso de modernización abarca varias etapas críticas, incluyendo la evaluación del estado actual del software, la investigación de mejores prácticas y estándares de la industria, la recopilación y análisis de requerimientos, la selección de tecnologías emergentes y actuales, y el diseño de la nueva arquitectura del software. A través de este enfoque integral, se busca abordar los desafíos actuales de obsolescencia del software Gestemed y aprovechar las oportunidades futuras de crecimiento y competitividad en el mercado.

1.2. Antecedentes

1.2.1. Descripción de la organización

Key Tech es una empresa líder en la prestación de servicios tecnológicos de vanguardia, especializada en la transformación digital impulsada por la inteligencia artificial (IA). Fundada en 2023, Key Tech se ha establecido como un socio estratégico esencial para empresas que buscan mantenerse competitivas en la era digital mediante el uso de tecnologías innovadoras y prácticas de la industria. (N. Morales, comunicación personal, 2024).

El enfoque de la organización se centra en proporcionar soluciones tecnológicas personalizadas que optimicen procesos, mejoren la eficiencia y fomenten el crecimiento sostenible de los clientes. La oferta abarca una amplia gama de servicios, incluyendo análisis y mejora de procesos, gestión de proyectos, ciberseguridad, calidad de software, implementación de mejores prácticas, gestión de redes sociales, desarrollo de branding, inteligencia de negocios, y desarrollo de software.

Misión

“Impulsar el potencial de la IA y las tecnologías digitales para mejorar la vida de las personas y las empresas con calidad y productividad.” (Key Tech, 2024)

Visión

“Ser el socio tecnológico de confianza para las empresas de la región, ayudándoles a alcanzar sus objetivos de negocio mediante el uso de tecnologías de punta.” (Key Tech, 2024)

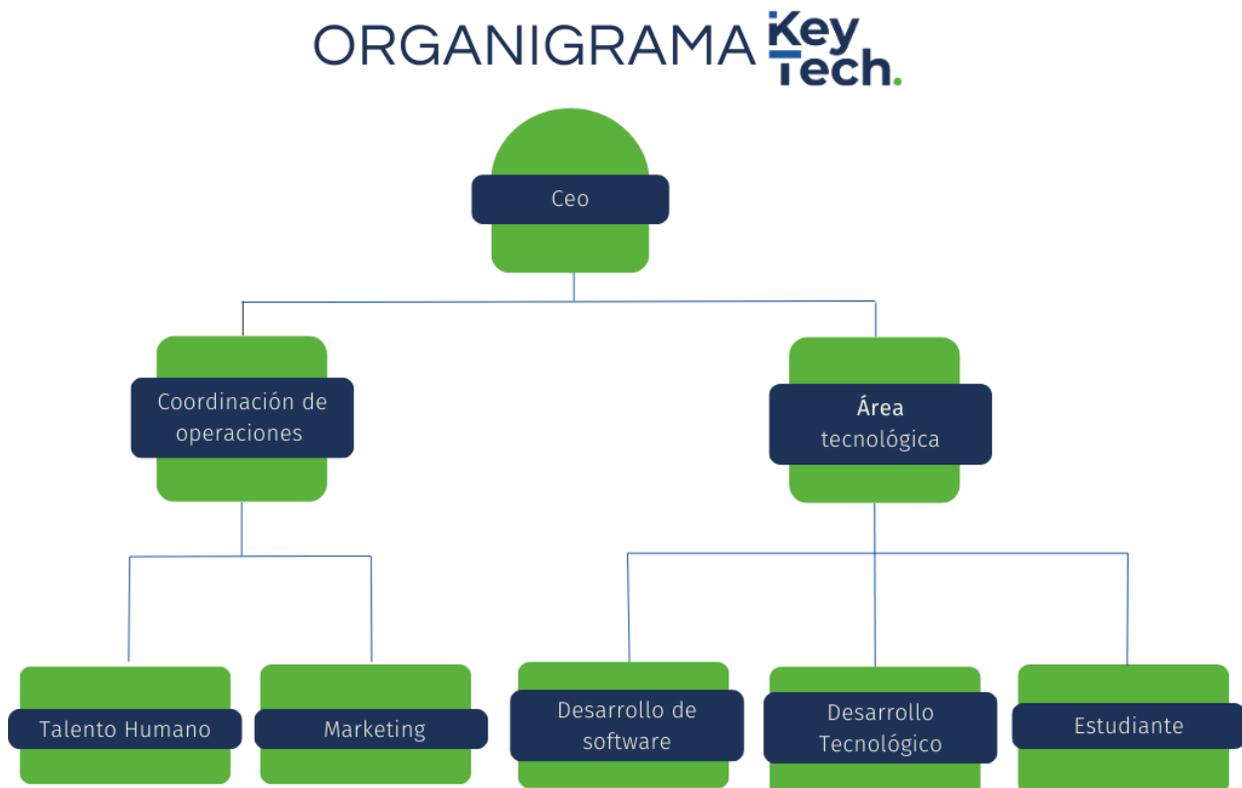
Valores

- “Excelencia.
- Innovación.
- Trabajo en equipo.
- Compromiso con el cliente.
- Transparencia.” (Key Tech, 2024)

1.2.1.1. Equipo de trabajo

En la Figura 1, se presenta el organigrama de la empresa Key Tech y equipo de trabajo involucrado en el desarrollo del proyecto.

Figura 1
Organigrama Key Tech.



Nota. Creación propia (2024).
Basado en información brindada por Key Tech.

En la Tabla 1 se describe el rol de cada miembro en la empresa y su participación específica en el TFG.

Tabla 1
Equipo de trabajo

Puesto laboral	Descripción	Rol ante TFG
Líder tecnológico	Responsable de dirigir y supervisar los proyectos tecnológicos, asegurando la alineación con los objetivos estratégicos de la empresa y gestionando el equipo de desarrollo.	Supervisar el TFG, asegurar la alineación con los objetivos de negocio, y proporcionar orientación técnica.
Desarrolladores	Encargado de la programación y desarrollo de software, implementación de soluciones técnicas, y soporte continuo en la empresa.	Apoyar en el diseño y tecnicismos del proyecto.
Estudiante	Estudiante responsable de elaborar la propuesta para la modernización de la arquitectura del software Gestemed.	Responsable de desarrollar la propuesta de modernización de arquitectura del software Gestemed.

Nota. Creación propia (2024).

Basado en información brindada por Key Tech.

1.2.2. Trabajos similares realizados dentro y fuera de la organización

1.2.2.1. Interno

Este proyecto marca un hito significativo para Key Tech, ya que es la primera vez que la empresa posee una iniciativa de un plan de actualización integral. Aunque Key Tech tiene una sólida trayectoria en la implementación de soluciones tecnológicas innovadoras, no se había abordado un proyecto de este tipo. Esta nueva iniciativa surge como una respuesta a la creciente demanda de soluciones tecnológicas avanzadas y representa una oportunidad única para la empresa de explorar nuevas capacidades y expandir su oferta de servicios.

1.2.2.2. Externo

- Propuesta de metodología para la priorización de entregables y asignación de recurso humano de Core Data Solutions en Intel Costa Rica: Este proyecto es relevante porque aborda la organización y gestión eficiente de los recursos humanos y la priorización de entregables. En la modernización del software Gestemed, también se requiere un enfoque preciso para priorizar qué aspectos del sistema se deben mejorar primero y cómo asignar de manera óptima el personal técnico. Aplicar una metodología similar garantiza que se distribuyan los esfuerzos de manera eficiente, evitando sobrecargas de trabajo o retrasos.
- Propuesta de mejora de los procesos de admisión y matrícula del Instituto Tecnológico de Costa Rica utilizando una metodología BPM: Este proyecto está centrado en la mejora de procesos utilizando BPM, lo cual es directamente aplicable a la modernización del software Gestemed, ya que esta modernización implica no solo un cambio técnico, sino también una redefinición de procesos dentro de Key Tech. Al integrar BPM, es posible optimizar cómo se gestionan los equipos médicos y cómo interactúan los usuarios con el sistema.

1.3. Planteamiento del problema

En esta sección se desarrolla el planteamiento del problema que aborda las causas fundamentales de la obsolescencia de Gestemed.

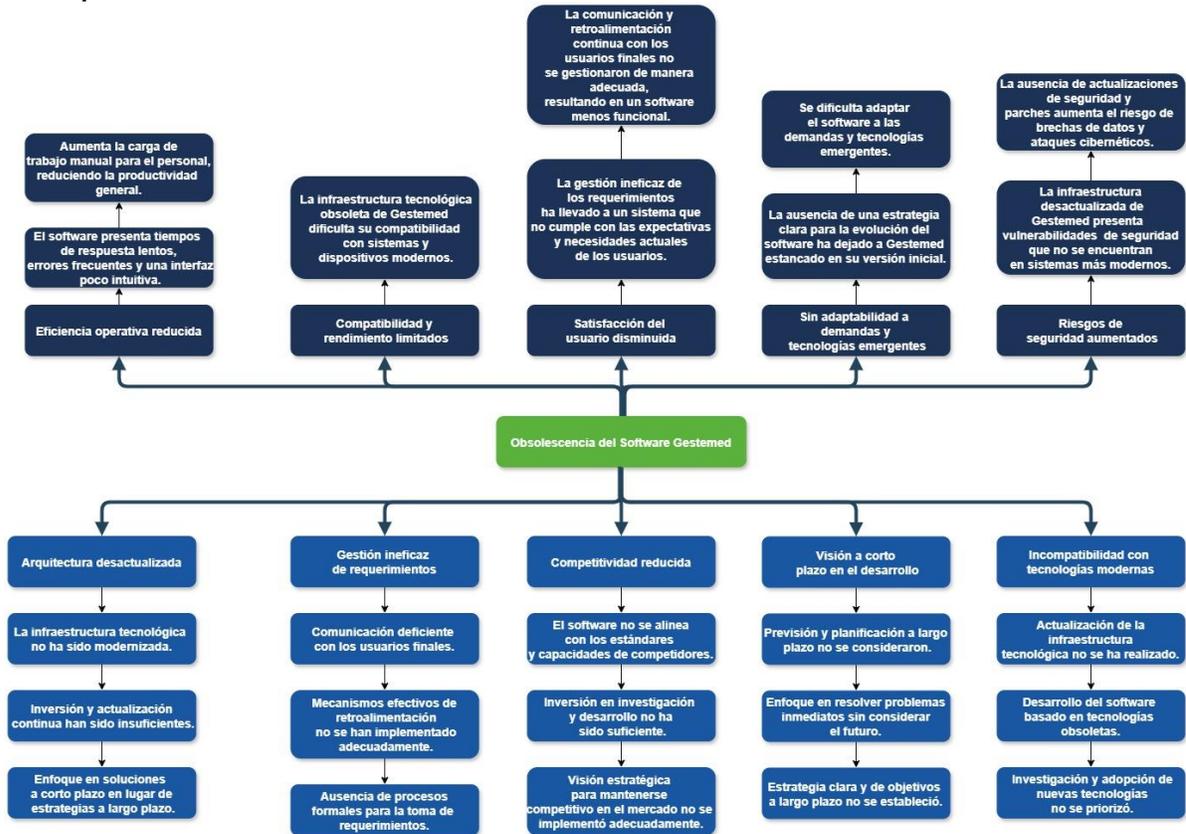
1.3.1. Situación problemática

Key Tech enfrenta actualmente un desafío significativo con el software Gestemed, desarrollado en 2007. Gestemed es un sistema de gestión de equipos médicos que asigna un código único a cada equipo recibido. Utiliza identificadores de radiofrecuencia y códigos QR para abrir expedientes y gestionar registros tanto preventivos como correctivos. Este sistema permite un seguimiento detallado del equipo médico desde su recepción hasta su mantenimiento y eventual reemplazo.

La obsolescencia de Gestemed se debe a su arquitectura desactualizada, que es incompatible con las tecnologías modernas. Esta obsolescencia no solo afecta la eficiencia operativa y la competitividad, de la misma manera limita la capacidad para adoptar nuevas tecnologías y responder adecuadamente a las demandas del mercado. Además, la gestión ineficaz en la toma de requerimientos y la visión a corto plazo en su desarrollo inicial han contribuido a su desactualización. Esto ha resultado en un software que no cumple con las expectativas actuales de los usuarios ni con el aprovechamiento de las innovaciones tecnológicas disponibles, lo que crea una barrera para el crecimiento y la adaptación en un entorno competitivo y en constante evolución.

Para entender la magnitud de este problema y justificar la necesidad de un proyecto de actualización integral, se ha elaborado la Figura 2, un árbol del problema que se divide en dos secciones principales: causas y efectos. Este enfoque permite visualizar claramente cómo las diferentes dimensiones del problema se interrelacionan y contribuyen a los desafíos actuales de la empresa.

Figura 2
Árbol del problema



Nota. Creación propia (2024).

La primera sección del árbol del problema se enfoca en las causas que han llevado a la obsolescencia de Gestemed, la infraestructura tecnológica de Gestemed no ha sido modernizada, lo que afecta su compatibilidad y rendimiento con las tecnologías actuales. Este problema se debe a la inversión y actualización insuficientes, así como a un enfoque a corto plazo en lugar de estrategias a largo plazo. Además, la comunicación deficiente con los usuarios finales ha resultado en un sistema que no satisface las expectativas y necesidades de sus usuarios. Los mecanismos efectivos de retroalimentación no se han implementado adecuadamente y los procesos formales para la toma de requerimientos no están bien estructurados.

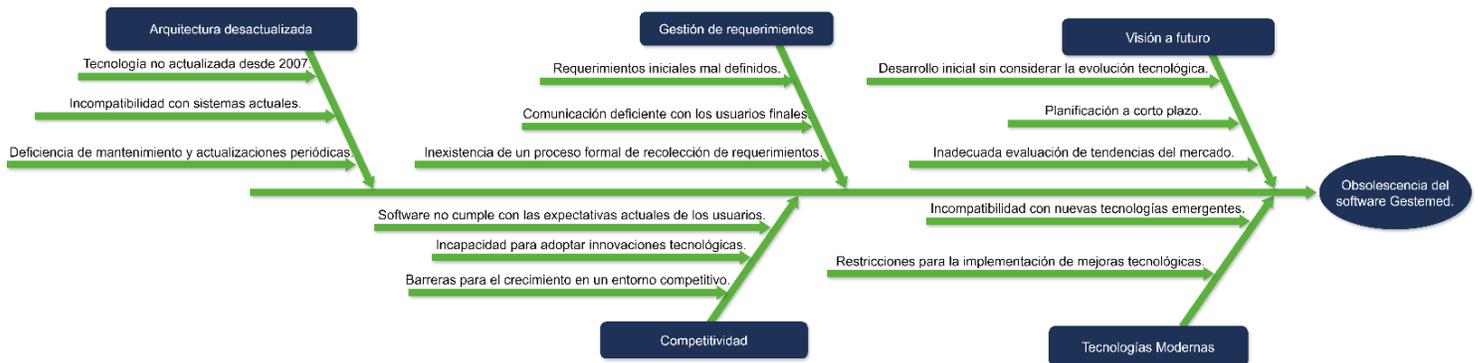
La previsión y planificación a largo plazo no se consideraron, lo que ha llevado a que el software no esté preparado para adaptarse a las demandas y tecnologías emergentes. No se estableció una estrategia clara y de objetivos a largo plazo, dejando a Gestemed estancado en su versión inicial. De la misma manera, la actualización de la infraestructura tecnológica no se ha realizado y el desarrollo del software se basó en tecnologías obsoletas. La investigación y adopción de nuevas tecnologías no se priorizó. El software no se alinea con los estándares y capacidades de los competidores. La inversión en investigación y desarrollo ha sido insuficiente, y la visión estratégica para mantenerse competitivo en el mercado no se implementó adecuadamente.

La segunda sección del árbol del problema se enfoca en los efectos que la obsolescencia de Gestemed ha generado, el software presenta tiempos de respuesta lentos, errores frecuentes y una interfaz poco intuitiva. Esto incrementa la carga de trabajo manual para el personal, reduciendo la productividad general. La infraestructura tecnológica obsoleta de Gestemed dificulta su compatibilidad con sistemas y dispositivos modernos. La gestión ineficaz de los requerimientos ha llevado a un sistema que no cumple con las expectativas y necesidades actuales de los usuarios. La comunicación y retroalimentación continua con los usuarios finales no se gestionaron adecuadamente, resultando en un software menos funcional.

La ausencia de una estrategia clara para la evolución del software ha dejado a Gestemed estancado en su versión inicial, dificultando su adaptación a nuevas demandas y tecnologías emergentes. La inversión continua en la mantención de un sistema obsoleto desvía recursos que deberían utilizarse para desarrollar nuevas funcionalidades. Además, la incapacidad de ofrecer un software competitivo reduce las oportunidades de negocio. La infraestructura desactualizada de Gestemed presenta vulnerabilidades de seguridad que no se encuentran en sistemas más modernos. La ausencia de actualizaciones de seguridad aumenta el riesgo de brechas de datos y ataques cibernéticos, poniendo en riesgo la integridad y confidencialidad de los datos gestionados por el software.

El Diagrama de Ishikawa, también conocido como diagrama de espina de pescado, es una herramienta visual que ayuda a identificar las posibles causas de un problema específico. La Figura 3 permite visualizar las causas principales que contribuyen al problema actual, así como los efectos directos que estas causas tienen en la empresa.

Figura 3
Diagrama de Ishikawa



Nota. Creación propia (2024)

Las principales categorías identificadas como responsables de esta obsolescencia son la arquitectura desactualizada, la gestión de requerimientos, la visión a futuro, la competitividad y las tecnologías modernas. En primer lugar, la arquitectura desactualizada del software Gestemed señala que la infraestructura tecnológica no ha sido modernizada, lo cual afecta su compatibilidad y rendimiento con las tecnologías actuales. Esta obsolescencia en la base tecnológica impide que el software funcione eficientemente y aproveche las innovaciones disponibles.

En segundo lugar, la gestión de requerimientos presenta problemas relacionados con la recopilación y análisis de las necesidades del software. Esto se debe principalmente a una comunicación deficiente con los usuarios finales, lo que resulta en un sistema que no satisface las expectativas y necesidades de sus usuarios. La falta de retroalimentación y ajuste continuo ha llevado a que el software se vuelva menos relevante y funcional con el tiempo. La visión a futuro es otra causa crítica de la obsolescencia. Esta categoría apunta a la ausencia de previsión y planificación a largo plazo, lo cual ha llevado a que el software no esté preparado para adaptarse a las demandas y tecnologías emergentes. Sin una estrategia clara para la evolución del software, Gestemed ha quedado estancado en su versión inicial. Por último, la incompatibilidad del software con las tecnologías modernas impide la integración de nuevas herramientas y funcionalidades que mejoran su desempeño y utilidad. La adopción de tecnologías emergentes es crucial para mantener la relevancia y eficiencia del software, pero la obsolescencia de Gestemed dificulta esta integración.

La obsolescencia del software Gestemed ha generado múltiples desafíos para Key Tech. Para entender la magnitud del problema y justificar la necesidad de un proyecto de actualización integral, es crucial examinar en detalle los impactos negativos actuales. A continuación, se presentan los principales efectos adversos de la infraestructura desactualizada de Gestemed:

- El software, al no estar preparado para tecnologías modernas, presenta tiempos de respuesta lentos, errores frecuentes y una interfaz poco intuitiva para los usuarios. Esto se traduce en una mayor carga de trabajo manual para el personal, incrementando el tiempo dedicado a tareas repetitivas y reduciendo la productividad general.
- La infraestructura tecnológica obsoleta de Gestemed dificulta su compatibilidad con sistemas y dispositivos modernos.
- La gestión ineficaz de los requerimientos ha llevado a la implementación de un sistema que no cumple con las expectativas y necesidades actuales de los usuarios. La falta de comunicación y retroalimentación continua con los usuarios finales ha resultado en un software que se ha vuelto irrelevante y menos funcional con el tiempo. Esto no solo disminuye la satisfacción del usuario, sino que también aumenta la probabilidad de errores y fallos operativos.
- La ausencia de una estrategia clara para la evolución del software ha dejado a Gestemed estancado en su versión inicial. Sin una planificación a largo plazo, dificulta adaptar el software a las demandas y tecnologías emergentes, lo cual es crucial en un entorno tecnológico en constante cambio.
- La obsolescencia del software también conlleva un alto costo de oportunidad. La inversión continua en la mantención de un sistema obsoleto desvía recursos que podrían utilizarse para desarrollar nuevas funcionalidades. Además, la incapacidad de ofrecer un software competitivo reduce las oportunidades de negocio y limita el crecimiento.
- La infraestructura desactualizada de Gestemed presenta vulnerabilidades de seguridad que no se encuentran en sistemas más modernos. Estas vulnerabilidades al ser explotadas por actores malintencionados ponen en riesgo la integridad y confidencialidad de los datos gestionados por el software. La falta de actualizaciones de seguridad y parches aumenta el riesgo de brechas de datos y ataques cibernéticos, lo que tiene consecuencias graves.

La suma de estos impactos negativos justifica la necesidad urgente de actualizar el software Gestemed. Este proyecto no solo mejorará la eficiencia operativa y la satisfacción del cliente, sino que también fortalecerá la competitividad de Key Tech en el mercado, permitiendo a la empresa adaptarse a las tecnologías emergentes y aprovechar nuevas oportunidades de negocio.

1.3.2. Justificación del proyecto

La modernización de la arquitectura del software Gestemed es una necesidad crítica para asegurar que el sistema continúe siendo una herramienta eficaz y confiable para sus usuarios. A medida que las tecnologías avanzan y las demandas de los usuarios se vuelven más complejas, mantener un software con una infraestructura desactualizada conlleva varios problemas. Estos incluyen una disminución en el rendimiento, dificultades en el mantenimiento y actualizaciones, y vulnerabilidades de seguridad que pueden exponer el sistema a riesgos innecesarios.

El software Gestemed ha sido fundamental en la gestión de procesos y operaciones clave, pero su arquitectura actual está limitada por tecnologías que ya no son consideradas para las exigencias modernas. Estas limitaciones no solo afectan la capacidad del software para operar de manera eficiente en el presente, sino que también restringen su capacidad para integrarse con nuevas tecnologías, escalar con el crecimiento de la organización, y adaptarse a futuros cambios en los requisitos de los usuarios.

La modernización permite al software aprovechar al máximo las tecnologías actuales, lo que resulta en un sistema más ágil, flexible y robusto. Esto incluye la posibilidad de mejorar la experiencia del usuario con interfaces más intuitivas, aumentar la capacidad de procesamiento y almacenamiento para manejar mayores volúmenes de datos, y mejorar la seguridad cibernética con mecanismos avanzados de protección contra amenazas. Además, un sistema modernizado es más fácil de mantener y actualizar, lo que reduce significativamente los costos y esfuerzos asociados con el mantenimiento de un software anticuado.

Realizar este proyecto es esencial para asegurar que el software Gestemed siga siendo una herramienta viable y eficiente. La modernización no solo mejora la funcionalidad actual del sistema, también lo prepara para enfrentar otro tipo de exigencias. Un software que no evoluciona junto con las necesidades de sus usuarios y las tecnologías emergentes corre el riesgo de volverse obsoleto, lo que compromete a la capacidad de la organización para cumplir con sus objetivos operativos.

Este proyecto es importante porque permitirá al software Gestemed continuar ofreciendo un alto nivel de servicio a sus usuarios, sin los problemas asociados con las limitaciones tecnológicas actuales. Además, al anticipar y abordar estos problemas antes de que se conviertan en críticos, se evitan interrupciones costosas y se asegura una operación continua y eficiente. En definitiva, modernizar la arquitectura del software es una inversión en la sostenibilidad y el futuro del sistema, garantizando que permanezca alineado con las mejores prácticas tecnológicas y preparado para responder a los desafíos y oportunidades del futuro.

1.3.3. Beneficios esperados o aportes del Trabajo Final de Graduación

En este apartado se detallan los beneficios directos e indirectos esperados de la actualización integral del software Gestemed en Key Tech. En este caso, los beneficios directos e indirectos de la actualización del software Gestemed que traerá a Key Tech van más allá de la mejora del producto en sí, impactando en la competitividad, la imagen de la empresa, la satisfacción del cliente y otros aspectos clave que contribuyen al éxito y crecimiento de la organización.

1.3.3.1. Beneficios directos

- Al finalizar el TFG, se cuenta con un plan de implementación detallado que incorpora una arquitectura de software modernizada y alineada con las tecnologías actuales. Este plan incluye la migración hacia una infraestructura más robusta, junto con la integración de nuevas funcionalidades que mejoran significativamente el rendimiento y la usabilidad del software. Estas mejoras contemplan una interfaz de usuario más intuitiva y eficiente, que facilita el uso del software por parte de los operadores, así como la adopción de tecnologías de vanguardia que permiten al sistema adaptarse a innovaciones y necesidades.
- El proyecto proporciona un diagnóstico exhaustivo de las necesidades actuales de los usuarios y de las expectativas que tienen sobre el software Gestemed. Este análisis detallado permite identificar con precisión las especificaciones técnicas y funcionales necesarias para que la solución propuesta cumpla con los estándares modernos. Además, se abordan aspectos críticos como la escalabilidad, la seguridad y la facilidad de mantenimiento, asegurando que el software no solo se modernice, sino que también sea capaz de evolucionar junto con los cambios en el entorno tecnológico y operativo.

1.3.3.2. Beneficios indirectos

- Aunque Key Tech no está directamente involucrada en la implementación, el plan de modernización del software Gestemed y las mejoras proyectadas en la gestión de equipos médicos tienen un impacto positivo en la satisfacción de los clientes de la empresa. Al ofrecer una solución más robusta y eficiente, los clientes experimentan un software que responde mejor a sus necesidades, con menos problemas técnicos y una mayor facilidad de uso. Esto, a su vez, refuerza la confianza de los clientes en las capacidades tecnológicas de la empresa, mejorando su percepción y fidelidad a largo plazo.
- La actualización del software basada en el plan de implementación desarrollado durante el TFG abre nuevas oportunidades de negocio para Key Tech. Al contar con un producto más competitivo y tecnológicamente avanzado, la empresa está en una posición más fuerte para expandirse hacia nuevos mercados y captar a nuevos clientes que buscan soluciones de vanguardia. Esto no solo permite a Key Tech diversificar su

base de clientes, también consolidar su presencia en el mercado, aprovechando la ventaja competitiva que representa contar con un software actualizado y alineado con las últimas tendencias tecnológicas.

1.4. Objetivos del Trabajo Final de Graduación

1.4.1. Objetivo General

Proponer la modernización de la arquitectura del software Gestemed en Key Tech, para el favorecimiento de su compatibilidad con tecnologías modernas y satisfacción de las necesidades actuales de los usuarios, durante el segundo semestre del 2024.

1.4.2 Objetivos específicos

1. Analizar la situación actual de la arquitectura de software Gestemed, para el diagnóstico técnico mediante Fortalezas, Oportunidades, Debilidades, Amenazas, Limitaciones (FODAL).
2. Establecer el estado meta de la modernización de la arquitectura para atender los requerimientos funcionales y no funcionales que sean identificados mediante el FODAL.
3. Valorar las mejores prácticas de la industria sobre arquitectura de software para identificar los componentes a considerar en la modernización.
4. Construir una propuesta de modernización de la arquitectura del software para satisfacer todas las mejoras identificadas en el FODAL y los requerimientos elicitados.

1.5. Alcance

El alcance del proyecto incluye varias fases clave orientadas a la modernización de la arquitectura del software Gestemed. En primer lugar, se lleva a cabo una evaluación exhaustiva de la situación actual mediante un análisis FODAL. Este análisis permite identificar los aspectos técnicos más relevantes del sistema y las áreas que necesitan mejoras, estableciendo una base sólida para la modernización de la arquitectura, acorde a las exigencias actuales de la industria y las necesidades operativas del sistema.

Posteriormente, se procede a la definición de los requerimientos funcionales y no funcionales necesarios para la modernización. Estos requerimientos serán formulados en función de las necesidades detectadas tanto en los usuarios finales como en la organización, asegurando que el nuevo diseño del sistema cumpla con los objetivos y sea capaz de soportar las demandas tecnológicas actuales. Además, se establecen criterios de rendimiento, usabilidad y seguridad, lo que permite que el nuevo sistema no solo cumpla con los requisitos funcionales, sino que también se mejore la eficiencia y la experiencia de usuario.

Como siguiente paso, se evalúa la arquitectura actual del sistema y se investigan alternativas de arquitecturas de referencia, poniendo énfasis en la identificación y adopción de buenas prácticas de la industria. Estas buenas prácticas sirven de guía para la selección de las tecnologías más adecuadas y modernas, que garanticen eficiencia, escalabilidad y sostenibilidad a largo plazo. Se consideran las tendencias tecnológicas emergentes y los estándares internacionales reconocidos en el sector, con el fin de asegurar que las decisiones tecnológicas estén a la vanguardia y se adapten fácilmente a las demandas cambiantes del mercado.

Finalmente, se elabora una propuesta de modernización basada en las tecnologías investigadas, este enfoque garantiza que la modernización sea escalable, sostenible y alineada con los objetivos estratégicos de la empresa. Esta propuesta incluirá una descripción técnica de las tecnologías seleccionadas, especificando cómo se integrarán en la arquitectura actual del sistema y los cambios necesarios para su implementación exitosa. además, se proveen a la empresa diversas opciones tecnológicas adaptadas a sus necesidades específicas, considerando las tendencias emergentes en la industria. Esto permitirá a Key Tech trazar un camino claro y estructurado hacia la modernización, facilitando la toma de decisiones informadas sobre qué tecnologías adoptar y cómo integrarlas de manera eficiente en su infraestructura. Además, la propuesta considerará un plan detallado que abarcará los recursos financieros y humanos requeridos, así como las fases del proceso de modernización. Se incluyen estrategias específicas para mitigar riesgos y gestionar el cambio dentro de la organización. La propuesta también toma en cuenta aspectos críticos como la compatibilidad con los sistemas existentes, la escalabilidad para expansiones y el cumplimiento de normativas regulatorias y de seguridad de la información.

1.6. Entregables

En Tabla 2 se especifican los entregables según los objetivos específicos.

Tabla 2

Entregables del producto

Objetivo	Entregables asociados
Objetivo específico 1	<ul style="list-style-type: none">Documento con los resultados del análisis FODAL de la arquitectura de software actual. Este documento identifica las áreas críticas que necesitan mejoras y sirve como base para la modernización.
Objetivo específico 2	<ul style="list-style-type: none">Documento que define los requerimientos funcionales y no funcionales necesarios para la modernización del sistema. Este entregable contempla las necesidades de los usuarios finales y la organización.
Objetivo específico 3	<ul style="list-style-type: none">Documento con las buenas prácticas de la industria que se adoptarán y una justificación de la selección de tecnologías modernas y adecuadas.
Objetivo específico 4	<ul style="list-style-type: none">Propuesta que incluye una descripción técnica completa de las tecnologías seleccionadas, los cambios requeridos en la arquitectura actual. Este entregable incluye las fases de modernización, estrategias de mitigación de riesgos y gestión del cambio.

Nota. Creación propia (2024).

1.7. Supuestos

En la sección de supuestos del proyecto se detallan los factores o elementos que se asumen como ciertos para la realización del proyecto. En este proyecto son:

- Se asume que se contará con los recursos humanos, financieros y tecnológicos necesarios para llevar a cabo el proyecto, incluyendo personal capacitado en las tecnologías seleccionadas.
- Se supone que los usuarios de Gestemed colaborarán activamente en la recopilación de requerimientos y en la validación de las soluciones propuestas, asegurando que el nuevo software responda a sus necesidades.
- Se asume que se tendrá acceso a toda la información relevante sobre el sistema actual de Gestemed, incluyendo su arquitectura, código fuente, bases de datos y documentación asociada.

1.8. Limitaciones

En la sección de restricciones del proyecto se detallan los factores o elementos que limitan de alguna manera la realización del proyecto, excluyendo el tiempo y el costo. Estas limitaciones afectan la planificación y ejecución del proyecto y es importante tenerlas en cuenta desde el inicio.

- El proyecto tiene un plazo definido de 6 meses para su ejecución, donde existe la posibilidad de limitar la profundidad de algunos análisis y la implementación de ciertas funcionalidades.
- La arquitectura y el código fuente de Gestemed si son complejos y poco documentados, dificultaría la comprensión completa de su funcionamiento y la identificación de áreas de mejora.

1.9. Exclusiones

- El trabajo se enfoca principalmente en proporcionar una descripción textual detallada y en realizar un análisis conceptual exhaustivo de las necesidades, objetivos y propuestas del proyecto, dejando la representación gráfica como un aspecto secundario o complementario. Esto es común en proyectos en los que el propósito inicial es establecer una base sólida de conocimientos y justificar las decisiones tomadas antes de pasar a etapas más técnicas.
- En un entorno empresarial dinámico y en constante evolución, es fundamental que las soluciones tecnológicas propuestas sean lo suficientemente flexibles para adaptarse a las circunstancias específicas de la organización. En el caso de este proyecto, ofrecer varias opciones en lugar de una única propuesta permite que la empresa elija la solución que mejor se alinee con su contexto actual, considerando factores como su capacidad técnica, recursos financieros, prioridades estratégicas y objetivos a largo plazo.

2. Marco conceptual

2.1. Requerimientos de software

Los requerimientos de software son la base fundamental sobre la que se construye cualquier sistema de software. Se definen como las necesidades y expectativas documentadas que el sistema debe cumplir para satisfacer a los usuarios y a las partes interesadas (Wiegiers & Beatty, 2013, p. 5). Estos abarcan una amplia gama de especificaciones, desde funcionalidades técnicas precisas hasta características no funcionales, como la seguridad y la usabilidad.

2.1.1. Definición de requerimiento

Un requerimiento se refiere a cualquier función, capacidad o característica que un sistema debe poseer para cumplir con los objetivos de negocio, satisfacer las necesidades del usuario, o cumplir con las restricciones técnicas o legales del entorno en el que opera. Wiegiers y Beatty (2013) describen el requerimiento de software como una declaración de lo que debe hacer el sistema, sin especificar cómo lo hará (p. 6). Esto implica que los requerimientos son el puente entre las necesidades del usuario y la implementación técnica del sistema.

2.1.2. Tipos de requerimientos

2.1.2.1. Requerimientos funcionales

Los requerimientos funcionales describen lo que el sistema debe hacer para cumplir con las expectativas del usuario y los objetivos del negocio. Específicamente, los requerimientos funcionales se centran en las acciones que el sistema debe ser capaz de realizar y las interacciones que los usuarios tendrán con el mismo. Estos incluyen características como la entrada de datos, procesamiento, cálculos y la generación de salidas específicas.

Wiegiers y Beatty (2013) explican que los requerimientos funcionales son esenciales porque describen las capacidades que el sistema debe proporcionar para cumplir con los requerimientos del usuario y los de negocio (p. 261). Estos requerimientos suelen redactarse en un formato claro y estructurado, y se representan mediante casos de uso, diagramas de flujo de datos, o descripciones en lenguaje natural.

Ejemplo de requerimientos funcionales:

- El sistema debe permitir a los usuarios registrar sus datos personales a través de un formulario en línea.
- El sistema debe validar que los campos obligatorios en el formulario de registro estén completos antes de permitir el envío de la información.
- El sistema debe calcular automáticamente los impuestos basados en la ubicación geográfica del usuario.

- El sistema debe generar informes financieros mensuales que puedan ser exportados en formato PDF o Excel.

Estos ejemplos muestran cómo los requerimientos funcionales detallan las tareas específicas que el software debe realizar, ayudando a definir las interacciones entre los usuarios y el sistema.

2.1.2.2. Requerimientos no funcionales

Mientras que los requerimientos funcionales se centran en las acciones que el sistema debe realizar, los requerimientos no funcionales definen cómo el sistema debe comportarse y cómo debe cumplir con sus funciones. Estos describen atributos de calidad del sistema y aspectos relacionados con el rendimiento, la seguridad, la usabilidad, la capacidad de mantenimiento, la disponibilidad, y otros factores importantes que determinan el éxito general del software.

Wieggers y Beatty (2013) destacan que los requerimientos no funcionales son críticos para garantizar que el sistema sea eficiente, seguro y fácil de usar, pero a menudo se les presta menos atención que a los funcionales (p. 263).

Categorías comunes de requerimientos no funcionales:

- **Rendimiento:** Define las expectativas sobre la rapidez con la que el sistema debe responder a las acciones del usuario o manejar grandes volúmenes de datos.
 - **Ejemplo:** El sistema debe procesar al menos 1000 transacciones por segundo con un tiempo de respuesta no mayor a 2 segundos.
- **Escalabilidad:** Establece los requisitos para que el sistema crezca en capacidad y pueda manejar un mayor volumen de datos o usuarios.
 - **Ejemplo:** El sistema debe ser capaz de manejar el incremento de usuarios concurrentes desde 100 hasta 10,000 sin degradar su rendimiento.
- **Seguridad:** Incluye especificaciones sobre el manejo seguro de los datos, autenticación, autorización y protección contra vulnerabilidades.
 - **Ejemplo:** El sistema debe encriptar todos los datos sensibles, como contraseñas y números de tarjetas de crédito, utilizando el estándar de cifrado AES-256.
- **Disponibilidad:** Establece los niveles de tiempo en los que el sistema debe estar operativo y accesible para los usuarios.
 - **Ejemplo:** El sistema debe estar disponible el 99.9% del tiempo durante el horario laboral (de 8:00 a.m. a 6:00 p.m.), excepto en tiempos programados de mantenimiento.
- **Usabilidad:** Define lo intuitiva y fácil que debe ser la interfaz del sistema para los usuarios finales.

- **Ejemplo:** Los usuarios deben ser capaces de completar una transacción en menos de 3 minutos con un máximo de 5 clics.
- **Compatibilidad:** Describe los requerimientos del sistema en cuanto a interoperabilidad con otros sistemas, plataformas o dispositivos.
 - **Ejemplo:** El sistema debe ser compatible con los navegadores web más utilizados (Chrome, Firefox, Edge, Safari) y debe funcionar tanto en dispositivos móviles como en computadoras de escritorio.

2.1.3. Problemas comunes relacionados con los requerimientos

Una adecuada gestión de los requerimientos es esencial para el éxito de cualquier proyecto de software. Wiegers y Beatty (2013) destacan que un proceso sólido de requerimientos reduce el riesgo de fallos en el proyecto, mejora la comunicación entre las partes interesadas y el equipo de desarrollo, y asegura que el producto final cumpla con las expectativas (p. 22). Cuando los requerimientos están mal definidos o se manejan de manera inadecuada, es común que ocurran problemas como la falta de participación de los usuarios, requisitos ambiguos, o cambios no controlados en el alcance del proyecto (Wiegers & Beatty, 2013, p. 19).

Wiegers y Beatty (2013) mencionan varios problemas recurrentes que los proyectos de software enfrentan cuando no existe un proceso de requerimientos adecuado:

1. **Falta de participación de los usuarios:** Si los usuarios finales no están involucrados desde el principio, los desarrolladores pueden no entender completamente las necesidades del sistema, lo que lleva a un producto que no satisface a los usuarios (p. 20).
2. **Requerimientos ambiguos:** Los requerimientos escritos de manera vaga o confusa pueden interpretarse de diferentes maneras por los desarrolladores, resultando en un sistema que no cumple con lo esperado (p. 21).
3. **Creep de requerimientos:** El *scope creep* ocurre cuando los requerimientos aumentan durante el desarrollo sin una gestión adecuada de los cambios. Esto conduce a retrasos, sobrecostos y un sistema sobrecargado de funciones no planificadas (p. 20).

2.1.4. Desarrollo de requerimientos

El proceso de desarrollo de requerimientos implica no solo la definición de estos, sino su gestión a lo largo del ciclo de vida del proyecto. Esto implica mantener los requerimientos actualizados, manejar los cambios y asegurar que todas las partes interesadas comprendan y acepten las modificaciones (Wiegers & Beatty, 2013, p. 457).

El libro *Software Requirements* se enfoca en el desarrollo de los requerimientos, abarcando desde la identificación de las necesidades del negocio hasta la documentación y validación de estos. Esta parte proporciona un marco de trabajo detallado y técnicas prácticas para asegurar que los requerimientos reflejen de manera precisa lo que los usuarios y las partes interesadas necesitan del sistema.

El primer paso en el desarrollo de requerimientos es comprender las necesidades del negocio. Los requerimientos de negocio son las metas de alto nivel que describen los beneficios que el sistema debe aportar a la organización (Wieggers & Beatty, 2013, p. 78). Estos requerimientos definen el "por qué" del sistema, es decir, el motivo detrás de su desarrollo. Se trata de asegurarse de que el proyecto esté alineado con las metas estratégicas de la organización.

Ejemplo de requerimientos de negocio:

- Reducir los costos de adquisición en un 25% durante el primer año de operación del sistema.
- Mejorar la eficiencia de los empleados reduciendo el tiempo de procesamiento de solicitudes en un 30%.

Después de definir los objetivos del negocio, se deben establecer los requerimientos del usuario, que describen lo que los usuarios necesitan para cumplir con sus tareas. Este capítulo subraya la importancia de involucrar a los usuarios en las primeras etapas del proyecto para asegurar que el sistema sea útil y fácil de usar. (Wieggers & Beatty, 2013, p. 143).

Los casos de uso y las historias de usuario son dos herramientas que se utilizan para documentar los requerimientos del usuario. Un caso de uso describe cómo un usuario, denominado actor, interactúa con el sistema para lograr un objetivo específico. Un caso de uso es una secuencia detallada de acciones, que representa la interacción entre el actor y el sistema, y abarca todos los posibles escenarios, incluidos los caminos alternativos y las excepciones. Los casos de uso son una herramienta poderosa para describir los requerimientos funcionales de un sistema. Según Wieggers y Beatty (2013), un caso de uso proporciona una descripción estructurada y paso a paso de cómo un actor utiliza el sistema para lograr un objetivo, lo que ayuda a definir claramente los requerimientos funcionales desde la perspectiva del usuario (p. 143).

Las historias de usuario son una técnica más ágil y simple para capturar los requerimientos desde la perspectiva del usuario. A diferencia de los casos de uso, que son más detallados y formales, las historias de usuario son breves descripciones escritas en lenguaje natural que indican lo que el usuario quiere lograr y por qué es importante. Son comunes en entornos de desarrollo ágil.

Una historia de usuario suele seguir una estructura sencilla, a menudo conocida como el formato: Como [usuario], quiero [objetivo] para [motivo]. Esto asegura que la historia esté claramente centrada en el valor que la funcionalidad proporcionará al usuario final. Según Wieggers y Beatty (2013), las historias de usuario permiten a los equipos capturar rápidamente las necesidades del usuario, lo que es útil en proyectos ágiles donde los requerimientos pueden evolucionar durante el desarrollo (p. 147). Las historias son intencionalmente breves, lo que facilita su creación y modificación rápida, y se profundizan en detalle a medida que se acerca el momento de implementarlas.

Los casos de uso describen interacciones específicas entre los usuarios y el sistema, mientras que las historias de usuario son descripciones más simples y breves de las necesidades de los usuarios.

El modelado de requerimientos es una técnica útil para visualizar las interacciones, los procesos y las estructuras de datos del sistema. Los diagramas de flujo de datos, los diagramas de entidad-relación y los mapas de diálogo son ejemplos de modelos que pueden complementar la descripción textual de los requerimientos (Wieggers & Beatty, 2013, p. 187).

Los diagramas de flujo de datos (DFD) son una técnica visual utilizada para modelar el flujo de información dentro de un sistema. Representan cómo los datos se mueven a través de las diferentes partes del sistema, incluyendo cómo los datos entran y salen, dónde se almacenan y cómo se procesan. Los DFD se utilizan para identificar los procesos principales del sistema y las interacciones entre estos procesos y las entidades externas. Según Wieggers y Beatty (2013), los DFD son útiles para describir cómo los datos fluyen a través de un sistema sin entrar en detalles sobre la implementación. Estos diagramas permiten a los equipos de desarrollo entender cómo la información se mueve entre los distintos componentes y actores del sistema (p. 187).

Los diagramas de entidad-relación (DER) son una técnica visual utilizada para modelar las relaciones entre las entidades (tablas o conjuntos de datos) en una base de datos. Un DER muestra cómo las entidades (objetos del mundo real) se relacionan entre sí dentro de un sistema. Esta técnica es especialmente útil durante la fase de diseño de bases de datos, ya que ayuda a los desarrolladores a estructurar y organizar los datos de manera efectiva. Según Wieggers y Beatty (2013), un DER es útil para capturar la estructura lógica de los datos, mostrando las relaciones entre diferentes conjuntos de datos y ayudando a definir cómo se organizarán y conectarán las tablas en una base de datos relacional (p. 193).

Los mapas de diálogo son una técnica que se utiliza para modelar la interacción entre un usuario y el sistema a través de interfaces gráficas. Un mapa de diálogo representa los distintos estados o pantallas que el usuario verá y cómo puede navegar entre ellos. Son útiles para diseñar interfaces de usuario intuitivas, al permitir a los diseñadores visualizar cómo los usuarios interactuarán con el sistema de manera lógica y fluida. Según Wieggers y Beatty (2013), los mapas de diálogo permiten a los equipos de diseño comprender cómo fluyen las interacciones entre pantallas y ayudan a identificar posibles problemas de usabilidad antes de que el sistema se implemente (p. 198).

2.1.5. Priorización de requerimientos

No todos los requerimientos son igualmente importantes, y es esencial que el equipo de desarrollo se enfoque en los que aporten mayor valor al negocio (Wiegiers & Beatty, 2013, p. 314). Se sugieren varias técnicas de priorización, como la matriz MoSCoW (Must, Should, Could, Won't), que ayuda a clasificar los requerimientos en diferentes niveles de prioridad.

2.1.5.1. MoSCoW

- **Must have - Debe tener:** Son los requerimientos críticos para el éxito del proyecto. Sin estos requerimientos, el sistema o producto no podrá ser considerado funcional. Los *must have* representan las características esenciales que son imprescindibles para el producto final (Wiegiers & Beatty, 2013, p. 315).
- **Should have - Debería tener:** Son los requerimientos importantes, pero no críticos. Aunque son muy deseables, el sistema funcionaría sin ellos. Su falta no comprometerá la funcionalidad principal, aunque la experiencia del usuario podría ser menos eficiente (Wiegiers & Beatty, 2013, p. 316).
- **Could have - Podría tener:** Son los requerimientos deseables, pero no esenciales. Son considerados extras que se implementarán solo si hay tiempo o recursos suficientes, sin comprometer otros requerimientos más prioritarios (Wiegiers & Beatty, 2013, p. 316).
- **Won't have - No tendrá por ahora:** Son los requerimientos que se han acordado que no se implementarán en la versión actual del sistema, pero que pueden ser considerados para versiones futuras. No se incluyen debido a limitaciones de tiempo, presupuesto o alcance (Wiegiers & Beatty, 2013, p. 317).

2.1.5.2. Modelo Kano

El Modelo Kano clasifica los requerimientos en función de cómo impactan la satisfacción del cliente:

- **Requerimientos básicos:** Características esperadas que, si no están presentes, causan insatisfacción (Wiegiers & Beatty, 2013, p. 318).
- **Requerimientos de desempeño:** Características que mejoran la satisfacción cuando se implementan de manera óptima (Wiegiers & Beatty, 2013, p. 319).
- **Requerimientos atractivos:** Características que no se esperan pero que, si están presentes, generan gran satisfacción (Wiegiers & Beatty, 2013, p. 320).

2.1.5.3. Método de puntos ponderados

En este método, se asignan pesos o valores a los requerimientos basados en criterios como valor para el negocio, costo, complejidad técnica, y riesgo (Wiegiers & Beatty, 2013, p. 321). Se suman los valores asignados para obtener una puntuación total, lo que permite priorizar los requerimientos con base en su impacto global.

2.1.5.4. Método 100 dólares

En este enfoque, las partes interesadas reciben 100 dólares ficticios para asignar a diferentes requerimientos. Los que reciban más asignación de recursos son priorizados más alto (Wiegers & Beatty, 2013, p. 322).

2.1.5.5. Método MoM (*Minimally Marketable Feature*)

Esta técnica se enfoca en identificar el conjunto mínimo de características necesarias para que el producto sea viable comercialmente. Se priorizan los requerimientos que permiten lanzar una versión funcional del producto lo antes posible (Wiegers & Beatty, 2013, p. 323).

2.1.5.6. Método de priorización basada en el Retorno de la Inversión (ROI)

En este método, se priorizan los requerimientos en función de su impacto en el ROI. Los requerimientos que ofrecen el mayor valor por menor costo son los que reciben la mayor prioridad (Wiegers & Beatty, 2013, p. 324).

2.1.5.7. Análisis de Riesgo-Beneficio

Este enfoque combina la evaluación del riesgo con el beneficio que aporta cada requerimiento. Aquellos con alto beneficio y bajo riesgo se priorizan más alto, mientras que los de alto riesgo y bajo beneficio se relegan (Wiegers & Beatty, 2013, p. 325).

2.1.6. Gestión de requerimientos

La gestión de requerimientos se enfoca en proporcionar prácticas, herramientas y técnicas para garantizar que los requerimientos sean precisos, actualizados y comprendidos a lo largo de todo el ciclo de vida del proyecto. La gestión de los requerimientos es fundamental para evitar malentendidos, retrasos y sobrecostos, ya que permite mantener los requerimientos bajo control.

Las principales prácticas de gestión de requerimientos que son esenciales para mantener un proyecto bien organizado y alineado con los objetivos del negocio. Las actividades clave incluyen:

- **Control de versiones:** Implica el mantenimiento de versiones claras y actualizadas de los requerimientos. Cada cambio o modificación debe ser registrado en una nueva versión para asegurar la trazabilidad del desarrollo y evitar confusiones (Wiegers & Beatty, 2013, p. 457).
- **Control de cambios:** Dado que los requerimientos tienden a evolucionar, el control de cambios es esencial para gestionar las modificaciones de manera formal, evaluando el impacto en el proyecto antes de su implementación (Wiegers & Beatty, 2013, p. 462).
- **Seguimiento del estado de los requerimientos:** El seguimiento del estado de los requerimientos permite monitorizar su evolución a lo largo del ciclo de vida, desde la propuesta inicial hasta su implementación y validación final (Wiegers & Beatty, 2013, p. 467).

- **Trazabilidad de los requerimientos:** La trazabilidad permite conectar cada requerimiento con sus respectivos elementos de diseño, código, pruebas, y otros artefactos del proyecto, garantizando que no se pierda ningún requerimiento durante el proceso de desarrollo (Wiegiers & Beatty, 2013, p. 470).

Una gestión de cambios efectiva es crucial para evitar el *scope creep* (incremento descontrolado del alcance) y asegurar que los cambios propuestos se manejen de manera controlada (Wiegiers & Beatty, 2013, p. 472), para esto se debe tomar en cuenta:

- **Política de control de cambios:** Define cómo se proponen, revisan y aprueban los cambios, proporcionando una estructura para evaluar su impacto (Wiegiers & Beatty, 2013, p. 473).
- **Junta de control de cambios (CCB):** Es un grupo encargado de revisar y decidir sobre los cambios propuestos en los requerimientos. Esto asegura que solo se implementen los cambios que aportan verdadero valor al proyecto (Wiegiers & Beatty, 2013, p. 475).

La trazabilidad de los requerimientos es el proceso de conectar los requerimientos con otros elementos clave del sistema, como el diseño, el código y las pruebas. Esta trazabilidad asegura que todos los requerimientos sean implementados y verificados correctamente (Wiegiers & Beatty, 2013, p. 485).

2.2. Arquitectura de software

La arquitectura de software es un elemento crítico en el desarrollo de sistemas complejos. Representa tanto la estructura como las decisiones técnicas que permiten a un sistema cumplir con sus requisitos funcionales y no funcionales (Bass, Kazman, & Clements, 2013). La arquitectura abarca desde la organización de los componentes de software hasta las interacciones entre ellos, proporcionando una visión clara del sistema y facilitando su desarrollo, mantenimiento y evolución.

2.2.1. Definición de arquitectura de software

La arquitectura de software es definida como el conjunto de estructuras necesarias para razonar sobre un sistema, que comprende elementos de software, relaciones entre ellos y sus propiedades (Bass et al., 2013). Estas estructuras permiten entender cómo los diferentes componentes del sistema interactúan y se integran para cumplir con los objetivos de negocio y los requisitos técnicos. La arquitectura también define los límites de cada componente y las interfaces necesarias para que los elementos del sistema puedan comunicarse de manera efectiva.

2.2.2. Principios fundamentales de la arquitectura de software

La arquitectura de software sigue varios principios clave que aseguran que el sistema sea robusto, escalable y mantenible:

- **Modularidad:** La modularidad es un principio esencial en la arquitectura, que organiza el sistema en componentes o módulos independientes. Esto facilita la escalabilidad y el mantenimiento, ya que los cambios en un módulo no afectan a los demás (Richards & Ford, 2020). Los módulos se desarrollan, despliegan y modifican de manera independiente, lo que permite una mayor flexibilidad a lo largo del ciclo de vida del sistema.
- **Abstracción:** La abstracción ayuda a los arquitectos a enfocarse en las interacciones de alto nivel entre los componentes del sistema, sin preocuparse por los detalles técnicos de implementación. Esto es crucial para gestionar la complejidad de los sistemas modernos (Bass et al., 2013).
- **Desacoplamiento:** El principio de desacoplamiento implica que los componentes del sistema estén lo suficientemente aislados entre sí para minimizar las dependencias. Esto permite que cada componente evolucione sin afectar el funcionamiento del resto del sistema (Richards & Ford, 2020).
- **Escalabilidad:** La capacidad del sistema para manejar un crecimiento en la carga de trabajo, ya sea a través de la adición de más recursos o la optimización de los existentes. Este principio es especialmente importante en arquitecturas basadas en servicios o microservicios (Bass et al., 2013).

2.2.3. Características arquitectónicas

Las características arquitectónicas definen los atributos de calidad que un sistema debe cumplir. Estas características se dividen en tres categorías principales: (Richards & Ford, 2020)

- **Operacionales:** Estas características están relacionadas con el comportamiento del sistema en producción. Incluyen:
 - **Rendimiento:** La capacidad del sistema para responder en tiempos adecuados.
 - **Disponibilidad:** La capacidad del sistema para estar disponible en todo momento.
 - **Escalabilidad:** La habilidad del sistema para manejar una creciente cantidad de usuarios o transacciones sin degradar su rendimiento.
- **Estructurales:** Se refieren a cómo está organizado el sistema y su capacidad de ser modificado:
 - **Modularidad:** Organización del sistema en componentes independientes.
 - **Flexibilidad:** Capacidad para adaptarse a cambios en los requisitos sin una reestructuración completa.
- **Transversales:** Son aquellas características que afectan a todo el sistema, como la seguridad y la mantenibilidad. Estas características son críticas para asegurar que el sistema pueda soportar cambios, garantizar la integridad de los datos y protegerse contra ataques.

2.2.4. Patrones arquitectónicos

Los patrones arquitectónicos ofrecen soluciones probadas a problemas comunes en el diseño de sistemas. Algunos de los patrones más importantes incluyen:

- **Arquitectura en capas:** Se organiza el sistema en capas, donde cada capa tiene una responsabilidad específica. Este patrón mejora la separación de preocupaciones y facilita el mantenimiento del sistema (Richards & Ford, 2020).
- **Microservicios:** Los microservicios dividen el sistema en pequeños servicios autónomos que pueden desarrollarse, desplegarse y escalarse de forma independiente. Esto mejora la flexibilidad y la capacidad de respuesta del sistema ante cambios en los requisitos (Bass et al., 2013).
- **Event-Driven Architecture:** Utiliza eventos para comunicar cambios entre componentes, lo que lo hace adecuado para sistemas distribuidos que manejan grandes volúmenes de datos en tiempo real (Richards & Ford, 2020).

2.2.5. Toma de decisiones arquitectónicas y compromisos

La arquitectura de software implica la toma de decisiones críticas que influyen en el desarrollo y la operación del sistema. Estas decisiones incluyen la elección de tecnologías, estrategias de despliegue y la estructura del sistema. Cada decisión implica compromisos o trade-offs, ya que mejorar una característica, como la seguridad, puede afectar negativamente a otras, como el rendimiento (Richards & Ford, 2020). Por ejemplo, un sistema con altos niveles de seguridad puede experimentar tiempos de respuesta más largos debido a la encriptación de datos y otras medidas de protección.

2.2.6. Evaluación de riesgos arquitectónicos

La evaluación de riesgos es fundamental en la gestión de la arquitectura de software. Richards y Ford (2020) recomiendan el uso de matrices de riesgos que permiten identificar y gestionar riesgos potenciales a lo largo del ciclo de vida del sistema. La creación de funciones de aptitud arquitectónica ayuda a monitorear de manera automatizada si el sistema sigue cumpliendo con los criterios arquitectónicos establecidos, como la disponibilidad o el tiempo de respuesta.

2.2.7. Componentes en arquitectura de software

Los componentes juegan un rol esencial en la arquitectura de software, especialmente en los enfoques modernos como la arquitectura orientada a servicios (SOA) y los microservicios. A lo largo del tiempo, las arquitecturas han evolucionado hacia estructuras más modulares y desacopladas, donde los componentes permiten construir sistemas complejos mediante la combinación de elementos más simples y reutilizables.

- **Definición de componentes en arquitectura de software**

En la arquitectura de software, un componente es una unidad funcional autónoma que encapsula una parte del sistema y ofrece servicios a través de interfaces bien definidas. Este concepto promueve el desarrollo modular, donde cada componente puede ser diseñado, implementado, probado y mantenido de manera independiente del resto del sistema. Bass, Clements y Kazman (2003) definen un componente como "un elemento de software con interfaces especificadas que puede ser reemplazado y utilizado en diferentes contextos sin modificar su implementación interna".

- **Evolución histórica de los componentes en arquitectura de software**

El uso de componentes en arquitectura de software no es nuevo; se remonta a los primeros intentos de modularización en el desarrollo de sistemas complejos. Los inicios se pueden rastrear hasta la programación estructurada, donde se promovía la descomposición de sistemas en funciones o módulos que, si bien eran reutilizables, no estaban lo suficientemente desacoplados como para ser considerados componentes en el sentido moderno.

El verdadero auge de los componentes llegó con la programación orientada a objetos (POO), en la cual los objetos encapsulaban tanto datos como comportamiento. Sin embargo, el concepto de componente va más allá de la simple encapsulación de datos y métodos. Como lo plantea

Szyperski (2002), un componente debe ser visto como una unidad autónoma que no solo encapsula funcionalidad, sino que también expone interfaces claramente definidas para interactuar con otros componentes de manera interoperable.

Durante los años 90, el desarrollo de plataformas como *Common Object Request Broker Architecture* (CORBA) y *Distributed Component Object Model* (DCOM) introdujo la idea de componentes distribuidos. Estas tecnologías buscaban permitir la comunicación y coordinación de componentes distribuidos en redes, aunque sufrían de complejidades inherentes a la interoperabilidad y al acoplamiento rígido entre sistemas heterogéneos (Orfali, Harkey, & Edwards, 1996).

- **Arquitectura basada en componentes (CBA) y su aplicación**

La CBA es un enfoque de diseño en el cual los sistemas se construyen ensamblando componentes que pueden ser desarrollados de manera independiente y luego integrados a través de interfaces definidas. La CBA se basa en dos principios fundamentales:

- **Composición:** El sistema completo se construye a partir de componentes que interactúan a través de interfaces bien definidas.
- **Desacoplamiento:** Los componentes deben estar lo suficientemente desacoplados para que puedan ser modificados, reemplazados o actualizados sin necesidad de realizar cambios en otros componentes o en la infraestructura general del sistema.

Este enfoque tiene su fundamento en la teoría de sistemas y en la modularidad promovida por Parnas (1972), quien destacó la importancia de dividir los sistemas en módulos independientes para reducir la complejidad y mejorar la mantenibilidad. La CBA amplía este concepto al considerar los componentes no solo como módulos internos, sino como unidades autónomas que pueden ser desplegadas de manera independiente y reutilizadas en otros sistemas o aplicaciones. (Heineman & Council, 2001)

- **Arquitectura orientada a servicios (SOA) y componentes**

La SOA es otro paradigma que ha influido en la evolución del uso de componentes. En SOA, los servicios son tratados como componentes que ofrecen funcionalidades bien definidas y que están disponibles para otras aplicaciones o servicios a través de una red. Estos componentes se comunican usando protocolos estándar, como SOAP o REST, permitiendo que sistemas heterogéneos interactúen de manera más flexible y dinámica.

En SOA, los componentes no solo encapsulan lógica de negocio, sino que también están diseñados para ser desplegables y orquestados en tiempo de ejecución. Esto agrega una capa de complejidad adicional en cuanto a la gestión de dependencias y la coordinación de servicios distribuidos. Sin embargo, SOA fue criticada por introducir alto acoplamiento entre servicios debido a las dependencias implícitas y los contratos estrictos entre los mismos. (Erl, 2005)

- **Microservicios: una evolución de los componentes distribuidos**

Una evolución clave de SOA es la arquitectura de microservicios, en la cual los componentes (microservicios) son aún más pequeños y están completamente desacoplados, operando como aplicaciones autónomas que se comunican a través de interfaces ligeras como HTTP. Cada microservicio gestiona su propio ciclo de vida, desde el almacenamiento de datos hasta la lógica de negocio, lo que reduce el acoplamiento entre componentes y aumenta la resiliencia del sistema.

Los microservicios aportan varias ventajas al concepto de componentes:

- **Despliegue independiente:** Cada componente puede ser actualizado, escalado o reemplazado sin afectar a los demás.
- **Aislamiento de fallos:** Si un microservicio falla, el resto del sistema puede seguir funcionando, lo que mejora la disponibilidad general del sistema (Newman, 2015).
- **Escalabilidad individual:** Los microservicios permiten escalar partes del sistema de manera granular, ajustando los recursos a las necesidades específicas de cada componente.

Sin embargo, los microservicios también introducen desafíos, como la gestión de la comunicación entre servicios, la necesidad de manejar transacciones distribuidas y la orquestación de servicios para que funcionen correctamente en conjunto. Estos desafíos han sido abordados con tecnologías como Docker y Kubernetes, que permiten la gestión eficiente de microservicios en contenedores y la orquestación de componentes en entornos distribuidos. (Burns & Oppenheimer, 2017)

- **Desafíos en el uso de componentes**

El uso de componentes en arquitectura de software ofrece claras ventajas, pero también presenta varios desafíos:

- **Interoperabilidad y estandarización:** A medida que los sistemas se vuelven más complejos y distribuidos, asegurar que los componentes puedan comunicarse de manera efectiva puede ser problemático, especialmente si los componentes están escritos en diferentes lenguajes de programación o corren en plataformas distintas.
- **Gestión de dependencias:** Las dependencias entre componentes pueden generar problemas de acoplamiento y versiones. Esto es especialmente complicado en sistemas distribuidos, donde la actualización de un componente puede romper la compatibilidad con otros servicios (Clements et al., 2010).
- **Coordinación y orquestación:** En arquitecturas basadas en microservicios, la orquestación y coordinación de servicios se vuelve crítica para asegurar que los flujos de trabajo de negocio funcionen correctamente. Tecnologías como los sistemas de orquestación y los *event-driven architectures* (EDAs) son esenciales para manejar estos escenarios (Bass et al., 2003).
- **Seguridad:** A medida que los componentes interactúan a través de redes, se debe garantizar que las interfaces expuestas estén protegidas contra accesos no autorizados o

ataques maliciosos. Este desafío se agrava en arquitecturas distribuidas donde los puntos de entrada de los servicios son más numerosos (Newman, 2015).

2.2.8. Atributos de calidad de una arquitectura de software

Los atributos de calidad de una arquitectura de software, también conocidos como atributos no funcionales, son características clave que determinan el éxito de un sistema, ya que van más allá de la funcionalidad básica para abarcar aspectos como el rendimiento, la escalabilidad y la mantenibilidad. Estos atributos permiten evaluar cómo una arquitectura responde a las necesidades del negocio, los usuarios y los desarrolladores. A continuación, se detallan los principales atributos de calidad, su importancia y cómo impactan en el diseño arquitectónico.

Rendimiento (*Performance*)

El rendimiento se refiere a la capacidad del sistema para manejar solicitudes de manera eficiente, tanto en términos de velocidad de respuesta (latencia) como de capacidad para procesar múltiples solicitudes simultáneas (*throughput*). Por ejemplo, un sistema de comercio electrónico durante el Black Friday debe poder procesar miles de transacciones por segundo sin experimentar retrasos. Técnicas como la optimización de consultas de base de datos o el uso de caching son fundamentales para mejorar el rendimiento. Bass, Clements y Kazman (2003) señalan que un diseño arquitectónico eficiente puede mejorar significativamente el rendimiento general del sistema.

Escalabilidad (*Scalability*)

La escalabilidad es la capacidad del sistema para crecer y manejar un aumento en la carga sin sacrificar el rendimiento. Un ejemplo sería un servicio de *streaming* de video que debe adaptarse a un creciente número de usuarios. En estos casos, se suele recurrir a la escalabilidad horizontal, agregando más instancias de servidores. Fowler (2017) explica que las arquitecturas basadas en microservicios permiten escalar eficientemente los servicios específicos que necesitan más recursos.

Disponibilidad (*Availability*)

La disponibilidad mide cuánto tiempo un sistema permanece operativo y accesible. Esto es crucial en sistemas críticos como los de banca en línea, donde el tiempo de inactividad puede afectar gravemente a los usuarios y las operaciones comerciales. Para garantizar la alta disponibilidad, se utilizan técnicas como la redundancia de servidores y el balanceo de carga. Bass et al. (2003) subrayan la importancia de minimizar el tiempo de inactividad mediante diseños que soporten la redundancia y la recuperación rápida ante fallos.

Mantenibilidad (*Maintainability*)

La mantenibilidad se refiere a la facilidad con la que se puede modificar el sistema para corregir errores, actualizar características o adaptarse a nuevos requisitos. Un ejemplo común es la capacidad de una aplicación móvil para recibir actualizaciones frecuentes sin afectar su

funcionamiento. Clements et al. (2010) afirman que una arquitectura bien diseñada, basada en módulos independientes, facilita la mantenibilidad al permitir cambios localizados sin afectar al resto del sistema.

Seguridad (*Security*)

La seguridad es un atributo crítico que asegura que los datos y las operaciones dentro de un sistema estén protegidos contra accesos no autorizados o ataques maliciosos. Un ejemplo típico sería un sistema bancario que emplea autenticación multifactor y cifrado para proteger la información financiera de los usuarios. Erl (2005) destaca que la seguridad debe estar integrada desde las primeras etapas del diseño arquitectónico para garantizar que el sistema sea robusto contra posibles amenazas.

Flexibilidad (*Flexibility*)

La flexibilidad permite que un sistema se adapte a cambios en los requisitos o en el entorno sin una reestructuración significativa. Un sistema de gestión de inventarios, por ejemplo, debe poder ajustarse rápidamente a nuevas reglas o procesos de negocio. Heineman y Councill (2001) mencionan que los componentes reutilizables y desacoplados son clave para garantizar que una arquitectura sea lo suficientemente flexible para adaptarse a cambios futuros.

Portabilidad (*Portability*)

La portabilidad se refiere a la capacidad de un sistema para ser ejecutado en diferentes entornos sin necesidad de realizar grandes cambios. Por ejemplo, una aplicación que funcione tanto en servidores locales como en la nube debe estar diseñada para ser portátil. Burns y Oppenheimer (2017) destacan que el uso de contenedores y tecnologías como Docker facilitan la portabilidad, ya que permiten ejecutar la misma aplicación en diferentes infraestructuras.

Usabilidad (*Usability*)

La usabilidad se refiere a la facilidad con la que los usuarios pueden interactuar con el sistema. Aunque suele asociarse más con el diseño de interfaces, la arquitectura también influye en la usabilidad al garantizar que el sistema sea rápido y accesible. Un ejemplo sería una aplicación móvil que permita a los usuarios completar tareas con un mínimo de pasos y tiempos de carga reducidos. Nielsen y Budiu (2013) destacan que la usabilidad es crucial para la satisfacción del usuario y debe ser una prioridad en cualquier sistema.

Reusabilidad (*Reusability*)

La reusabilidad se refiere a la capacidad de reutilizar componentes o módulos de un sistema en diferentes proyectos o contextos. Esto es especialmente útil en el desarrollo de software empresarial, donde un módulo de facturación puede ser reutilizado en diferentes aplicaciones. Heineman y Councill (2001) señalan que el diseño modular, donde los componentes están claramente definidos y desacoplados, facilita la reusabilidad y reduce el tiempo de desarrollo en nuevos

2.2.9. Patrones de integración de arquitectura de software

Los patrones de integración de arquitectura de software juegan un papel crucial en la creación de sistemas que necesitan interoperar eficientemente, sobre todo en un entorno donde los sistemas son heterogéneos y distribuidos. Estos patrones no solo facilitan la comunicación entre distintos componentes, sino que también promueven la escalabilidad, mantenibilidad y flexibilidad del sistema. A continuación, se amplían algunos de los patrones más comunes y se exploran en mayor detalle sus características, ventajas, desventajas y ejemplos.

- **Mensajería Asíncrona (*Asynchronous Messaging*)**

La mensajería asíncrona es un patrón ampliamente utilizado en sistemas distribuidos para evitar la dependencia directa entre los emisores y receptores. En lugar de que el emisor espere una respuesta inmediata, los mensajes se envían a un intermediario, como una cola de mensajes, donde el receptor los recoge cuando está disponible. Esto garantiza que los sistemas puedan continuar funcionando sin interrupciones, incluso si hay problemas de conectividad o latencia en la red.

Características:

- **Comunicación desacoplada:** Los sistemas emisores y receptores no necesitan estar activos al mismo tiempo.
- **Tolerancia a fallos:** Si un servicio se cae, los mensajes se almacenan en la cola y se procesan cuando el servicio vuelva a estar disponible.
- **Escalabilidad:** Permite que los sistemas gestionen cargas de trabajo más grandes sin congestión.

Ventajas:

- Mejora la resiliencia del sistema.
- Incrementa la capacidad de manejar múltiples tareas simultáneamente.
- Facilita la comunicación entre servicios distribuidos en diferentes entornos.

Desventajas:

- La latencia puede ser mayor en comparación con una comunicación síncrona.
- La gestión de los mensajes perdidos o duplicados puede ser más compleja.

Ejemplo:

En un sistema de comercio electrónico, cuando un cliente realiza un pedido, el sistema de *frontend* puede enviar un mensaje a una cola de procesamiento. Los servicios de inventario y facturación recogerán estos mensajes cuando estén disponibles para procesarlos de manera independiente, sin necesidad de una comunicación síncrona con el *frontend* (Hohpe & Woolf, 2004).

- **Orquestación de Servicios (*Service Orchestration*)**

La orquestación de servicios centraliza el control de los flujos de trabajo y coordina la interacción entre varios servicios para cumplir con procesos de negocio complejos. El orquestador se encarga de invocar a los diferentes servicios en un orden específico y asegura que cada tarea se complete antes de pasar a la siguiente.

Características:

- **Control centralizado:** La lógica de negocio se maneja en un componente único que define cómo deben interactuar los servicios.
- **Procesos secuenciales o paralelos:** El orquestador puede definir si las interacciones deben ser secuenciales o ejecutarse en paralelo.
- **Monitorización y manejo de errores:** Es fácil de supervisar y detectar dónde ocurre un fallo dentro del flujo.

Ventajas:

- Proporciona claridad en el flujo de procesos.
- Facilita la implementación de transacciones distribuidas.
- Simplifica la gestión de errores y el control de las dependencias.

Desventajas:

- Puede crear un punto único de fallo si el orquestador tiene problemas.
- Introduce acoplamiento entre los servicios y el orquestador, lo que puede dificultar la evolución independiente de los servicios.

Ejemplo:

En un sistema de reservas de hoteles, el orquestador coordina el flujo completo: verifica la disponibilidad de habitaciones, procesa el pago, y genera una confirmación de reserva. Cada uno de estos servicios se invoca de manera secuencial y su éxito es monitoreado para garantizar que todo el proceso se complete correctamente (Bass, Clements, & Kazman, 2003).

- **Coreografía de servicios (*Service Choreography*)**

A diferencia de la orquestación, la coreografía de servicios no depende de un controlador central para gestionar la interacción entre los servicios. En su lugar, cada servicio está programado para reaccionar a ciertos eventos de forma autónoma, lo que permite una interacción más distribuida y descentralizada.

Características:

- **Descentralización:** No existe un orquestador central; los servicios son autónomos y saben cómo reaccionar a los eventos que suceden en el sistema.
- **Eventos asincrónicos:** Los servicios reaccionan a eventos o mensajes y, en función de estos, ejecutan las acciones necesarias.
- **Desacoplamiento:** Los servicios están más desacoplados, lo que permite una mayor independencia.

Ventajas:

- Mayor flexibilidad y desacoplamiento entre los servicios.
- Facilita la escalabilidad y la evolución independiente de los servicios.
- No depende de un único componente, lo que reduce el riesgo de puntos de fallo centralizados.

Desventajas:

- Puede ser más difícil de supervisar, ya que no hay un controlador central.
- La gestión de errores y transacciones distribuidas es más compleja.

Ejemplo:

En una plataforma de comercio electrónico, cuando un cliente finaliza un pedido, se emite un evento "Pedido realizado". Los servicios de inventario, logística y facturación escuchan este evento y actúan de manera autónoma para cumplir con sus tareas. Ningún servicio coordina la acción, pero todos están programados para actuar en función de los eventos que reciben (Pautasso, Zimmermann, & Leymann, 2008).

4. Punto a punto (*Point-to-Point*)

El patrón punto a punto establece una conexión directa entre dos sistemas que deben interactuar sin intermediarios. Este patrón es común en situaciones simples donde la integración directa es más rápida y menos compleja que otras soluciones. Sin embargo, a medida que aumenta el número de sistemas integrados, este enfoque puede volverse difícil de manejar.

Características:

- Conexión directa entre emisores y receptores.
- Comunicación síncrona o asíncrona.
- Generalmente utilizado para integraciones simples.

Ventajas:

- Bajo *overhead*: La comunicación es directa y sencilla.
- Rápido de implementar para necesidades puntuales.

Desventajas:

- Acoplamiento alto entre los sistemas, lo que dificulta su escalabilidad.
- A medida que crece el número de conexiones, se vuelve difícil de gestionar y mantener.

Ejemplo:

Un sistema de facturación puede comunicarse directamente con un sistema ERP para enviar detalles de las transacciones. Si bien esto funciona bien para una integración simple, podría no escalar adecuadamente si se añaden más sistemas que deban interactuar (Richards & Ford, 2020).

- **Canal de Mensajes (*Message Channel*)**

El canal de mensajes es un patrón que utiliza un intermediario, como un *message broker*, para facilitar la comunicación entre emisores y receptores. Los mensajes se envían a un canal específico, donde el sistema receptor los recoge cuando está disponible, desacoplando completamente a los sistemas.

Características:

- Desacopla a los sistemas emisores y receptores.
- Comunicación basada en mensajes, que pueden ser procesados cuando los sistemas receptores están listos.
- Puede soportar múltiples emisores y receptores.

Ventajas:

- Desacoplamiento total de los sistemas.

- Aumenta la tolerancia a fallos, ya que los mensajes no se pierden si un receptor está temporalmente inactivo.
- Fácil escalabilidad para manejar más mensajes y receptores.

Desventajas:

- Introduce complejidad adicional en la infraestructura, ya que requiere el mantenimiento de un intermediario.
- Requiere manejar duplicación o pérdida de mensajes, lo que puede ser complicado.

Ejemplo:

Un sistema de procesamiento de pedidos puede utilizar un canal de mensajes para enviar confirmaciones de pedidos a los servicios de facturación, logística e inventario. Cada servicio recogerá el mensaje del canal cuando esté listo para procesarlo, lo que permite que cada componente funcione de manera independiente (Hohpe & Woolf, 2004).

- **Patrón de Adaptador (*Adapter Pattern*)**

El patrón de adaptador es esencial para integrar sistemas que tienen interfaces incompatibles. Este patrón permite que un sistema o componente antiguo (legacy) se conecte a una nueva arquitectura sin tener que modificar el código fuente del sistema original.

Características:

- Traduce interfaces incompatibles entre sistemas.
- Permite que los sistemas antiguos funcionen con nuevas tecnologías.
- Puede usarse para conectar sistemas que usan diferentes protocolos.

Ventajas:

- Facilita la reutilización de sistemas antiguos o con tecnología obsoleta.
- Evita costosos rediseños de sistemas legacy.

Desventajas:

- Puede introducir sobrecarga en términos de rendimiento.
- Los adaptadores pueden necesitar actualizaciones frecuentes si los sistemas evolucionan.

Ejemplo:

Un sistema financiero heredado que utiliza SOAP puede integrarse con una API moderna basada en REST mediante un adaptador que convierte las solicitudes y respuestas entre los dos protocolos (Gamma, Helm, Johnson, & Vlissides, 1995).

2.3. Modernización de software

2.3.1. Definición de modernización de software

La modernización del software es el proceso de actualizar, mejorar y adaptar los sistemas de software existentes para que cumplan con los requerimientos actuales y futuros de la empresa o del mercado. Este proceso implica la revisión y transformación de la arquitectura de software para mejorar aspectos clave, como la escalabilidad, la seguridad, la eficiencia y la capacidad de integrar nuevas tecnologías. Según Richards y Ford (2020), la modernización de un sistema también incluye la adopción de buenas prácticas de la industria, la evaluación continua de las decisiones arquitectónicas y la capacidad de adaptar el sistema a los cambios en las necesidades del negocio y el entorno tecnológico (p. 16).

En otras palabras, la modernización del software involucra actividades como migrar a la nube, implementar nuevas arquitecturas (como microservicios), mejorar la eficiencia de los procesos de desarrollo e incorporar nuevas funcionalidades que respondan a las necesidades cambiantes de los usuarios.

2.3.2. Importancia de las buenas prácticas

Las buenas prácticas en la arquitectura de software son esenciales para garantizar que un sistema sea escalable, mantenible y adaptable a las necesidades cambiantes del negocio. Una de las prácticas recomendadas por Richards y Ford (2020) es la adopción de un enfoque de "arquitectura iterativa". Esto permite realizar ajustes en la arquitectura a medida que el proyecto avanza y el entorno cambia. Asimismo, los autores señalan que el uso de patrones arquitectónicos probados, como la separación de responsabilidades y el encapsulamiento de la lógica de negocio, contribuye a mejorar la calidad y la sostenibilidad del software. (Richards & Ford, 2020, pp. 38-46).

Otra buena práctica es la definición de características arquitectónicas (“-ilities”), como la escalabilidad, la seguridad, y la eficiencia. Estas características deben ser extraídas tanto de los requisitos funcionales como de los no funcionales, y se recomienda involucrar a los equipos de desarrollo en la definición de estas características para asegurar su alineación con los objetivos de la empresa. (Richards & Ford, 2020, pp. 65-67).

La adopción de enfoques como la arquitectura evolutiva, el uso de pruebas automatizadas y la implementación de herramientas de automatización, forman parte del conjunto de buenas prácticas que apoyan la modernización de la arquitectura y garantizan su éxito en el largo plazo (Richards & Ford, 2020, pp. 16-17).

2.3.3. Proceso de modernización

El proceso de modernización de la arquitectura de software implica varias fases clave que se deben seguir para lograr un cambio efectivo:

- **Evaluación de la arquitectura actual:** Esta fase incluye un análisis profundo de la situación actual para identificar las fortalezas, debilidades y puntos críticos del sistema. Según Richards y Ford (2020), es fundamental realizar un análisis exhaustivo de la estructura actual y de los problemas que enfrenta, lo que permitirá priorizar los cambios y definir una estrategia de modernización adecuada (pp. 4-5).
- **Definición de requisitos:** La modernización debe basarse en una clara comprensión de los requisitos funcionales y no funcionales del sistema. Richards y Ford (2020) sugieren extraer estos requisitos a partir de las necesidades del negocio y del análisis de los procesos actuales, permitiendo que la nueva arquitectura esté diseñada para satisfacer tanto las demandas técnicas como las expectativas de los usuarios (pp. 65-67).
- **Investigación y selección de tecnologías:** En esta etapa, se investigan las alternativas tecnológicas disponibles que mejor se adapten al nuevo diseño. La selección debe considerar criterios de rendimiento, escalabilidad, compatibilidad con los sistemas existentes y seguridad. Richards y Ford (2020) recomiendan optar por soluciones tecnológicas que permitan una fácil integración y que sigan las tendencias actuales de la industria para asegurar la sostenibilidad del sistema a largo plazo (pp. 267-271).
- **Implementación de cambios:** Una vez seleccionadas las tecnologías y definida la nueva arquitectura, se procede a la implementación de los cambios. Richards y Ford (2020) subrayan la importancia de seguir un enfoque iterativo y basado en incrementos, lo que permitirá introducir cambios gradualmente, minimizar riesgos y evaluar el impacto de cada cambio (pp. 245-263).

Ingeniería y evolución arquitectónica

En la modernización del software, la arquitectura no solo se concibe como una estructura estática, sino como un conjunto de decisiones y principios que deben adaptarse con el tiempo. La ingeniería y evolución arquitectónica implican un proceso continuo de evaluación y ajuste de la arquitectura para abordar los cambios en los requisitos y las condiciones del entorno.

Richards y Ford (2020) describen este enfoque como "arquitectura evolutiva", donde la arquitectura se desarrolla y ajusta conforme cambian las necesidades del negocio y la tecnología (p. 16). Para soportar esta evolución, los autores proponen el uso de "fitness functions" o funciones de aptitud arquitectónica. Estas funciones permiten evaluar y medir continuamente el cumplimiento de las características arquitectónicas deseadas, como el rendimiento y la seguridad, garantizando que la arquitectura mantenga su integridad a medida que se introduce el cambio (pp. 83-85).

La arquitectura evolutiva también se basa en la colaboración continua entre arquitectos y desarrolladores, así como en la implementación de prácticas como la automatización de pruebas, la integración y la implementación continuas. Esto permite una rápida detección de problemas y

facilita el ajuste oportuno de la arquitectura a medida que el sistema crece y se adapta a las nuevas demandas (Richards & Ford, 2020, pp. 16-17).

2.4. Estándares

La modernización del software requiere el uso de estándares, marcos y mejores prácticas que garanticen seguridad, calidad y alineación con los objetivos del negocio. A continuación, se detallan los componentes clave de cada uno de los estándares y marcos que contribuyen al proceso de modernización:

2.4.1. Organización Internacional de Normalización (ISO)

2.4.1.1. ISO/IEC 27001 (Seguridad de la información)

Esta norma no solo se centra en la implementación de un Sistema de Gestión de Seguridad de la Información (SGSI), sino también en el establecimiento de un enfoque sistemático para gestionar la seguridad de la información sensible. Involucra varios procesos, como la evaluación de riesgos, el tratamiento de riesgos mediante controles específicos, la auditoría interna y la mejora continua. Además, ISO/IEC 27001 se relaciona con otras normas de la familia 27000, lo que ayuda a la organización a establecer una estructura integral de seguridad de la información (ISO/IEC, 2013).

2.4.1.2. ISO/IEC 27002 (Controles de seguridad)

Esta norma ofrece una guía detallada sobre la selección, implementación y gestión de controles de seguridad, como controles criptográficos, seguridad física, seguridad en las comunicaciones y la gestión de incidentes. Los controles se agrupan en 14 dominios que cubren aspectos como políticas de seguridad, organización de la seguridad de la información, seguridad en recursos humanos y gestión de activos, lo que permite implementar medidas de seguridad adecuadas a las características de la organización y del software a modernizar (ISO/IEC, 2013).

2.4.1.3. ISO/IEC 25010 (Calidad del software)

El modelo de calidad definido en esta norma se compone de ocho características clave: funcionalidad, eficiencia de rendimiento, compatibilidad, usabilidad, confiabilidad, seguridad, mantenibilidad y portabilidad. Estas características se subdividen en subcaracterísticas específicas que proporcionan un marco completo para evaluar la calidad del software. Por ejemplo, la característica de seguridad se desglosa en confidencialidad, integridad, no repudio, autenticación y autorización, facilitando una evaluación integral del software modernizado (ISO/IEC, 2011).

2.4.1.4. ISO/IEC 12207 (Procesos del ciclo de vida del software)

Esta norma abarca todo el ciclo de vida del software, desde la conceptualización hasta el retiro del sistema. Define procesos principales, procesos de apoyo y procesos organizacionales. Por ejemplo, incluye procesos de desarrollo, implementación, mantenimiento, y gestión de la configuración. Durante la modernización, ISO/IEC 12207 sirve como una guía para asegurar que

cada etapa del ciclo de vida siga prácticas estructuradas y efectivas, contribuyendo a la calidad y sostenibilidad del software (ISO/IEC, 2008).

2.4.1.5. ISO/IEC 29110 (Desarrollo de software para pequeñas organizaciones)

Esta norma se diseñó para satisfacer las necesidades de las pequeñas y medianas empresas de software (PYMES). Contempla perfiles de ciclo de vida básicos y avanzados que proporcionan un enfoque simplificado para la gestión de proyectos, el desarrollo y la entrega de software. Incluye procesos específicos como la gestión de proyectos, implementación, pruebas y mantenimiento, adaptando las mejores prácticas de desarrollo de software a la escala y capacidad de las organizaciones más pequeñas (ISO/IEC, 2011).

2.4.2. The Open Group Architecture Framework (TOGAF)

El TOGAF 9.2 incluye el enfoque *Business-Driven* para la arquitectura empresarial, lo que significa que toda la estructura de la arquitectura debe alinearse con los objetivos estratégicos del negocio. TOGAF proporciona una forma estandarizada de modelar, documentar y gestionar los artefactos arquitectónicos, asegurando que todas las partes interesadas compartan una visión común de la arquitectura. Este enfoque es crucial para la modernización, ya que permite definir cómo deben ser los sistemas futuros y alinear las decisiones tecnológicas con la estrategia organizacional (The Open Group, 2018).

Architecture Development Method (ADM) el ADM es un proceso iterativo que consta de ocho fases, desde la "Preliminar" (donde se establece el alcance y las políticas de la arquitectura) hasta la "Gestión de Cambios" (que garantiza que la arquitectura se mantenga alineada con los objetivos a lo largo del tiempo). En la modernización del software, este método guía las etapas de análisis, diseño, implementación y evaluación, permitiendo ajustar la arquitectura conforme a las necesidades cambiantes del entorno (The Open Group, 2018).

El *Content Framework* incluye una serie de herramientas como catálogos (lista de aplicaciones y sistemas actuales y futuros), matrices (relaciones entre componentes) y diagramas (flujos de procesos, sistemas, redes). Esto facilita la representación visual de la arquitectura y permite comunicar las transiciones previstas en la modernización a todas las partes interesadas, asegurando un entendimiento compartido del proceso y sus objetivos (The Open Group, 2018).

2.4.3. Open Web Application Security Project (OWASP)

2.4.3.1. OWASP Top Ten

Esta lista no solo identifica las amenazas más críticas para las aplicaciones web, sino que también proporciona ejemplos concretos y medidas de mitigación para cada amenaza. Por ejemplo, para la amenaza inyección de código (como SQL Injection), OWASP recomienda la validación de entradas y el uso de consultas parametrizadas. Esta lista es esencial para cualquier proceso de modernización, ya que la seguridad debe ser un aspecto prioritario en el desarrollo de nuevas aplicaciones o en la actualización de las existentes. (OWASP, 2021).

2.4.3.2. OWASP Software Assurance Maturity Model (SAMM)

SAMM evalúa la madurez de las prácticas de seguridad en el desarrollo de software y proporciona un marco de mejora continua. SAMM se divide en cinco áreas de negocio: gobierno, diseño, implementación, verificación y operaciones, que cubren todas las fases del desarrollo seguro. Durante la modernización, SAMM permite evaluar los procesos actuales de desarrollo seguro, identificar áreas de mejora y definir un plan para alcanzar un nivel óptimo de madurez (OWASP, 2020).

2.4.3.3. OWASP Application Security Verification Standard (ASVS):

ASVS proporciona una lista de requisitos de seguridad organizados en tres niveles, desde requisitos básicos hasta avanzados, que permiten verificar la seguridad de una aplicación de manera estructurada. Este estándar incluye verificaciones específicas para la autenticación, manejo de sesiones, almacenamiento de datos, entre otros, facilitando la integración de la seguridad en las fases de desarrollo y pruebas durante la modernización. (OWASP, 2019)

2.4.3.4. OWASP Mobile Application Security Verification Standard (MASVS)

MASVS proporciona una lista detallada de requisitos de seguridad para aplicaciones móviles, desde la seguridad del entorno de desarrollo hasta la protección de datos y la autenticación. Esta norma es especialmente útil si la modernización implica el desarrollo de aplicaciones móviles, ya que garantiza que se implementen prácticas de seguridad específicas para este tipo de aplicaciones (OWASP, 2019).

2.5. Gestión del Cambio

La gestión del cambio es un proceso estructurado para guiar a individuos, equipos y organizaciones en la transición desde su estado actual hacia un estado deseado, asegurando la adopción efectiva de nuevas prácticas o tecnologías. Su relevancia es crucial en proyectos de modernización tecnológica, ya que los cambios en infraestructura, software o procesos generan desafíos organizacionales y humanos (Kotter, 2012).

2.5.1. Definición y objetivos

La gestión del cambio busca reducir la resistencia al cambio, asegurar la alineación de todos los niveles organizacionales y maximizar el retorno de inversión en proyectos innovadores. Según Hiatt (2006), el éxito de un cambio depende de tres pilares:

- **Liderazgo efectivo:** Los líderes deben comunicar claramente los beneficios del cambio y establecer objetivos alcanzables.
- **Compromiso de los empleados:** La participación de los usuarios es esencial para garantizar la adopción del nuevo sistema.
- **Sistemas de soporte:** Las herramientas y recursos deben estar disponibles para facilitar la transición.

2.5.2. Metodologías aplicables

- **Modelo de Kotter (2012):**
 1. Crear un sentido de urgencia.
 2. Formar una coalición de liderazgo.
 3. Desarrollar y comunicar una visión clara del cambio.
 4. Asegurar victorias rápidas para mantener el impulso.
- **Modelo ADKAR (Hiatt, 2006):**
 1. Conciencia del cambio.
 2. Deseo de participar.
 3. Conocimiento para implementar.
 4. Habilidad para ejecutar el cambio.
 5. Refuerzo para mantener los resultados.

2.6. Planificación del soporte técnico

La planificación del soporte técnico es una actividad estratégica que asegura la continuidad operacional de sistemas tecnológicos mediante el mantenimiento, diagnóstico y resolución eficiente de problemas técnicos. En proyectos de modernización de software, un plan sólido de soporte técnico es esencial para garantizar el rendimiento del sistema y la satisfacción del usuario final (Chaffey & White, 2011).

2.6.1. Objetivos principales

- **Minimizar interrupciones:** Reducir el impacto de fallas técnicas en la operación diaria.
- **Garantizar el rendimiento:** Asegurar que el software funcione de manera óptima bajo diferentes condiciones de uso.
- **Proveer continuidad:** Facilitar la escalabilidad y actualización del sistema según sea necesario.

2.6.2. Componentes clave

- **Diagnóstico y mantenimiento preventivo:**
 1. Implementar herramientas de monitoreo para identificar posibles fallos antes de que ocurran (Behrendt et al., 2016).
 2. Programar actualizaciones y parches regulares.

- **Capacitación técnica:**
 1. Formar a los equipos internos sobre el uso y resolución básica de problemas del sistema.
 2. Garantizar que los usuarios finales puedan reportar incidencias de forma eficiente.
- **Gestión de incidentes:**
 1. Definir procesos claros para atender y resolver fallos críticos.
 2. Establecer acuerdos de nivel de servicio (SLA, por sus siglas en inglés) para tiempos de respuesta y solución.
- **Infraestructura de soporte:**
 1. Sistemas de tickets para la gestión de solicitudes.
 2. Bases de conocimiento para la resolución de problemas frecuentes.

3. Marco metodológico

3.1. Tipo de investigación

La investigación se clasifica en varios tipos según su propósito, basado en esto se describen brevemente tres tipos fundamentales de investigación:

- **Investigación exploratoria:** Este tipo de investigación tiene como objetivo principal familiarizarse con un tema o problema que es poco conocido o que no ha sido previamente estudiado de manera exhaustiva. Según Hernández, Sampieri y Fernández (2014), la investigación exploratoria es útil para generar nuevas ideas, identificar variables clave, y establecer las bases para investigaciones futuras. Este tipo de investigación es fundamental en las primeras etapas de un proyecto, ya que permite al investigador adentrarse en áreas novedosas y formular preguntas más precisas.
- **Investigación descriptiva:** La investigación descriptiva, como indican Hernández, Sampieri y Fernández (2014), tiene como propósito principal describir las características de un fenómeno o una población. Este tipo de investigación se centra en detallar ¿qué es? en lugar de ¿por qué es? Se utiliza comúnmente en estudios donde se requiere un análisis detallado de las variables, pero sin entrar en la búsqueda de causas o efectos. La investigación descriptiva es útil para establecer perfiles, identificar patrones o tendencias, y proporcionar una visión completa de la situación actual.
- **Investigación explicativa:** La investigación explicativa va un paso más allá al intentar comprender las causas y relaciones entre las variables estudiadas. Este enfoque, según Hernández, Sampieri y Fernández (2014), busca responder preguntas del tipo ¿por qué? y ¿cómo? en relación con el fenómeno en estudio. Es un tipo de investigación más compleja que no solo describe un fenómeno, sino que también explora sus causas, efectos y las interacciones entre las variables involucradas. La investigación explicativa es fundamental para desarrollar teorías y modelos que pueden ser aplicados en situaciones reales.

El enfoque seleccionado para el proyecto en Key Tech es **la investigación explicativa**. Este tipo de investigación es particularmente adecuado debido a la naturaleza del problema que se enfrenta: la obsolescencia del software Gestemed. La elección de este enfoque se fundamenta en la necesidad de ir más allá de la simple descripción de la situación actual del software; se requiere una comprensión profunda de las causas subyacentes que han llevado a su obsolescencia y cómo estas causas afectan la eficiencia operativa y la competitividad de la empresa.

3.2. Enfoque de la investigación

Según Daniel Cauas (2015), la investigación cualitativa se caracteriza por el uso predominante o exclusivo de información cualitativa, enfocándose en lograr descripciones detalladas de los fenómenos estudiados. Este tipo de investigación suele enfatizar la aplicación práctica de los resultados obtenidos, siendo comúnmente utilizada en contextos donde se busca una comprensión profunda de las experiencias, significados y procesos subyacentes. Entre los métodos cualitativos más destacados se encuentran la investigación participativa, la investigación-acción, la investigación-acción participativa, la investigación etnográfica y el estudio de casos.

Por otro lado, Daniel Cauas (2015) indica que la investigación cuantitativa se distingue por su enfoque en la recolección y análisis de datos cuantitativos o cuantificables, es decir, aquellos que son medibles y que son expresados numéricamente. Este tipo de investigación es ampliamente utilizada para identificar patrones, establecer relaciones entre variables y generalizar resultados a poblaciones más amplias. Algunos de los métodos cuantitativos más utilizados incluyen los diseños experimentales, los cuasi-experimentales y las investigaciones basadas en encuestas sociales, siendo esta última una de las más empleadas en estudios cuantitativos.

De la misma manera los métodos mixtos, también conocidos como híbridos, se definen como un enfoque de investigación que combina tanto técnicas cualitativas como cuantitativas para la recolección y análisis de datos. Según Hernández et al. (2018), este enfoque “representa un conjunto de procesos sistemáticos, empíricos y críticos de investigación e implica la recolección y el análisis de datos tanto cuantitativos como cualitativos, así como su integración y discusión conjunta, para realizar inferencias producto de toda la información recabada (denominadas meta inferencias) y lograr un mayor entendimiento del fenómeno bajo estudio” (p. 10). De esta manera, el enfoque mixto busca ofrecer una comprensión más amplia y profunda del fenómeno investigado al combinar lo mejor de ambos paradigmas.

En este proyecto, se utiliza un enfoque de **investigación cualitativa**. La investigación cualitativa se centra en comprender fenómenos complejos desde una perspectiva holística, buscando captar las experiencias, percepciones y significados de los participantes. Este enfoque es adecuado para este proyecto, ya que se busca comprender en profundidad las necesidades, expectativas y desafíos.

3.3. Diseño de la investigación

Para seleccionar un diseño de investigación cualitativa que permitan una comprensión profunda de las experiencias y percepciones de los involucrados en el proceso de modernización del software Gestemed se explican a continuación:

- **Diseño de teoría fundamentada:** Se utiliza para desarrollar teorías directamente basadas en los datos obtenidos del campo. Este enfoque sigue un proceso iterativo de recolección de datos y análisis, donde los investigadores codifican los datos en categorías hasta que emergen teorías que explican el fenómeno estudiado. La teoría fundamentada es útil para explorar áreas de investigación con poca teoría previa y se emplea frecuentemente en estudios sobre procesos sociales o comportamientos complejos (Hernández-Sampieri & Mendoza, 2024, p. 526).
- **Diseños etnográficos:** Son ideales para estudiar las culturas, costumbres y comportamientos dentro de un grupo específico. El investigador se sumerge en la comunidad o el entorno que está investigando, observando y participando en las actividades diarias. El propósito es obtener una comprensión profunda de las dinámicas sociales y los significados culturales desde la perspectiva de los propios miembros de la comunidad (Hernández-Sampieri & Mendoza, 2024, p. 537).
- **Diseños narrativos:** Se centran en recopilar y analizar historias de vida o experiencias personales. Este enfoque permite comprender cómo las personas interpretan sus experiencias a lo largo del tiempo, dándole un valor importante al significado subjetivo que los individuos asignan a los eventos de sus vidas. Los estudios narrativos suelen usarse en investigaciones educativas, psicológicas y de salud (Hernández-Sampieri & Mendoza, 2024, p. 542).
- **Diseños fenomenológicos:** El propósito de este enfoque es comprender la esencia de las experiencias vividas por las personas en relación con un fenómeno determinado. En este tipo de investigación, se profundiza en las experiencias subjetivas de los participantes, buscando encontrar los elementos comunes y únicos que conforman su vivencia. Es particularmente útil para explorar emociones, percepciones y reacciones ante situaciones específicas (Hernández-Sampieri & Mendoza, 2024, p. 548).
- **Investigación-acción:** Este diseño implica una colaboración directa entre el investigador y los participantes para resolver un problema práctico. A menudo, se lleva a cabo en entornos educativos, de trabajo o comunitarios, donde las acciones y estrategias de mejora se implementan y evalúan continuamente. El ciclo de diagnóstico, intervención y evaluación permite una mejora progresiva de las condiciones o procesos estudiados (Hernández-Sampieri & Mendoza, 2024, p. 552).

El diseño de la investigación se basa en un enfoque de **investigación-acción** participativa, donde se busca la colaboración estrecha entre los investigadores y los usuarios de Key Tech. Esta colaboración activa permitirá identificar y abordar problemas en tiempo real, así como diseñar soluciones que se ajusten a las necesidades específicas de la organización.

3.4. Tipos de Fuentes de Información

Las fuentes primarias según lo indica Morales (2021) son un tipo fundamental de fuente de información, ya que proporcionan datos nuevos y originales, resultantes de una investigación o trabajo intelectual. Estas fuentes contienen información que no ha sido alterada, interpretada o analizada por otros autores, es decir, son testimonios directos del propio autor o investigador. En otras palabras, la información que proviene de una fuente primaria se mantiene intacta desde su creación, sin sufrir modificaciones ni reinterpretaciones.

Por otro lado, las fuentes secundarias representan otro tipo de fuente de información. Estas fuentes ofrecen datos que han sido organizados, elaborados o analizados por terceros. A menudo, las fuentes secundarias consisten en traducciones, resúmenes o reorganizaciones de la información obtenida de fuentes primarias. En este sentido, las fuentes secundarias proporcionan una visión ampliada o interpretada de los resultados presentados en las fuentes primarias, facilitando su comprensión y contextualización. (Morales, 2021)

3.4.1. Fuentes primarias

Una fuente primaria es un documento, registro o artefacto que proporciona información directa o evidencia original sobre un evento, objeto, persona o trabajo. En la Tabla 3 se listan las fuentes primarias.

Tabla 3

Fuentes primarias

Documento	Detalle
Documentación interna	Análisis de los documentos sobre el software como descripción del software, documentación técnica, listado de requerimientos iniciales del software.
Entrevistas con colaboradores de Key Tech	Entrevistas con el líder tecnológico y desarrollador para obtener información detallada sobre el software.

Nota. Creación propia (2024).

3.3.2. Fuentes secundarias

Una fuente secundaria es aquella que proporciona información y análisis derivados o basados en fuentes primarias. En la Tabla 4 se listan las fuentes secundarias.

Tabla 4

Fuentes secundarias

Documento	Detalle
ISO/IEC 12207	Esta norma proporciona un marco para el ciclo de vida del software, incluyendo procesos relacionados con la adquisición, suministro, desarrollo, operación, mantenimiento y eliminación del software. Siguiendo esta norma, se garantizará una gestión eficiente y efectiva de todo el ciclo de vida del software.
ISO/IEC 25010	Esta normativa proporciona un marco para la calidad del producto de software, incluyendo características como la funcionalidad, la fiabilidad, la usabilidad, la eficiencia y la mantenibilidad. Siguiendo esta normativa, se garantizará que el nuevo software Gestemed cumpla con altos estándares de calidad en todas sus áreas.
ISO/IEC 27001	Esta norma establece los requisitos para un sistema de gestión de seguridad de la información (SGSI). Dado que la seguridad de la información es crucial en el contexto de la gestión de datos médicos, seguir esta norma asegurará que el software modernizado cumpla con los estándares de seguridad necesarios.
<i>Software Requirements</i> de Karl Wieggers y Joy Beatty.	Este libro cubre una variedad de aspectos esenciales para la definición y gestión de requerimientos de software. Entre los temas tratados se incluyen: <ul style="list-style-type: none"> • Técnicas para la elicitación de requerimientos. • Análisis y especificación de requerimientos. • Validación y gestión de requerimientos. Herramientas y mejores prácticas para la documentación de requerimientos.

Nota. Creación propia (2024)

3.5. Sujetos de Investigación

Un sujeto de investigación es una persona o grupo de personas que participan en un estudio y proporcionan datos relevantes para el análisis e interpretación de la investigación. Estos sujetos son seleccionados en función de criterios específicos relacionados con el objetivo del estudio, y su participación es crucial para obtener resultados significativos y válidos. Según Hernández, 2014, los sujetos de investigación son las unidades de análisis a las que se les aplican los instrumentos de recolección de datos, y su selección debe responder a los criterios de representatividad y relevancia para el fenómeno estudiado. Los sujetos de investigación están listados en Tabla 5.

Tabla 5

Sujetos de investigación

Rol del sujeto	Años de experiencia	Caracterización del sujeto	Importancia
CEO	2 años	Responsable de supervisar todos los aspectos de la tecnología de la información dentro de Key Tech. Sus funciones incluyen la planificación estratégica de TI, la gestión de recursos tecnológicos, la implementación de políticas de seguridad y la supervisión del equipo de TI.	Este sujeto es crucial para la investigación debido a su visión estratégica y comprensión profunda de las necesidades tecnológicas y los objetivos empresariales de Key Tech. Su experiencia permitirá evaluar cómo la actualización del software Gestemed se alinea con la estrategia general de TI de la empresa.
Desarrollador	2 años	El desarrollador es responsable del diseño, desarrollo, mantenimiento y optimización del software Gestemed. Sus funciones incluyen la codificación, integración de nuevas funcionalidades, corrección de errores y actualización del sistema, de acuerdo con los requerimientos de los usuarios y las necesidades tecnológicas.	El desarrollador es clave para la investigación debido a su conocimiento técnico profundo del sistema actual. Su experiencia en la implementación de soluciones tecnológicas permitirá evaluar las limitaciones del software existente y proponer mejoras técnicas viables durante el proceso de modernización. Además, el desarrollador garantiza que las tecnologías seleccionadas sean adecuadas para su integración y futura implementación.

Rol del sujeto	Años de experiencia	Caracterización del sujeto	Importancia
Líder tecnológico	2 años	Responsable de la visión tecnológica a largo plazo de Key Tech. Sus funciones incluyen la identificación de nuevas tecnologías y tendencias, la definición de la arquitectura tecnológica y la guía estratégica para la implementación de soluciones tecnológicas.	Este sujeto es clave para la investigación debido a su rol en la dirección de la innovación y la adaptación tecnológica dentro de la empresa. Su perspectiva ayudará a asegurar que el plan de actualización de Gestemed esté alineado con las futuras tendencias tecnológicas y la estrategia a largo plazo de Key Tech.
Usuarios actuales /finales	1 a 10 años	Los cuatro usuarios finales son quienes interactúan directamente con el sistema Gestemed para realizar sus tareas operativas cotidianas. Estos usuarios pueden ocupar diversos roles dentro de la organización, desde empleados administrativos hasta personal técnico o de atención al cliente. Sus responsabilidades incluyen el uso del software para registrar datos, generar reportes, gestionar información de clientes o realizar otros procesos necesarios para el funcionamiento diario de la organización.	Los usuarios finales son fundamentales para la investigación, ya que proporcionan información directa sobre la usabilidad, eficiencia y limitaciones del sistema actual. Su experiencia y feedback son esenciales para identificar áreas de mejora en la interfaz de usuario, la funcionalidad y la eficiencia operativa. Al considerar sus necesidades y sugerencias, la modernización del sistema podrá alinearse mejor con los objetivos de mejorar la experiencia del usuario y aumentar la productividad organizacional.

Nota. Creación propia (2024)

3.6. Variables de Investigación

Las variables de investigación en este proyecto son esenciales para abordar de manera estructurada y efectiva el proceso de modernización del software Gestemed. Estas variables incluyen las fortalezas, oportunidades, debilidades y amenazas de la arquitectura actual del software, así como los requerimientos funcionales y no funcionales necesarios para su actualización. Además, se consideran variables críticas como la compatibilidad tecnológica y la viabilidad técnica de las alternativas propuestas. La correcta operacionalización de estas variables permitirá obtener datos precisos y relevantes, facilitando un análisis profundo que guiará la toma de decisiones en cada fase del proyecto.

En las siguientes cuatro tablas se desarrollan las variables de la investigación.

Tabla 6
Variables de investigación primer objetivo

Objetivo		Analizar la situación actual de la arquitectura de software Gestemed, para el diagnóstico técnico mediante Fortalezas, Oportunidades, Debilidades, Amenazas, Limitaciones (FODAL)		
Nombre de la variable	Tipo	Indicador	Detalle	Definición instrumental
Fortalezas	Cualitativa.	Número de aspectos positivos identificados en la arquitectura actual de Gestemed.	Identificación de características actuales que añaden valor al software.	Mediante entrevistas con el desarrollador y usuarios actuales, se recopilarán y analizarán las características que se perciben como fortalezas dentro de la arquitectura de Gestemed.
Oportunidades	Cualitativa/ Cuantitativa.	Número de posibles mejoras o adaptaciones que posiblemente sean implementadas.	Identificación de tecnologías emergentes o prácticas se integran en la modernización.	A través de un análisis de tendencias tecnológicas y <i>benchmarking</i> con sistemas similares, se identifican oportunidades de mejora en la arquitectura que pueden aprovecharse para fortalecer el software.
Debilidades	Cualitativa.	Número de aspectos negativos identificados en la arquitectura actual de Gestemed.	Identificación de limitaciones y problemas actuales en el software.	Mediante encuestas de satisfacción de usuarios para determinar las debilidades actuales de la arquitectura del software.

Nombre de la variable	Tipo	Indicador	Detalle	Definición instrumental
Amenazas	Cualitativa/ Cuantitativa.	Factores externos que impactan negativamente en el proyecto de modernización.	Identificación de riesgos externos.	Realizar un análisis Político, Económico, Social, Tecnológico, Ecológico, Legal (PESTEL) para identificar amenazas que podrían afectar el éxito de la modernización del software Gestemed.
Limitaciones	Cualitativa.	Número y tipo de restricciones internas que afectan la capacidad de modernización.	Restricciones internas como limitaciones de recursos financieros, técnicos o humanos	A través de entrevistas con el líder tecnológico y un análisis de recursos disponibles, se identificarán las principales limitaciones que obstaculizan la modernización de Gestemed.
Capacidades de integración existentes	Cualitativa	Número y tipo de capacidades de integración presentes en la arquitectura actual	Evaluación de las capacidades actuales, como APIs existentes, conexiones punto a punto, y el uso de bases de datos compartidas para determinar su eficiencia y efectividad.	Evaluar mediante entrevistas al desarrollador.
Interfaces de integración requeridas	Cualitativa/ Cuantitativa	Número y compatibilidad de las interfaces necesarias para la integración	Especificación de las interfaces necesarias, como protocolos de comunicación conectores de bases de datos, y mensajería que se requieren	Evaluar mediante entrevistas

Nota. Creación propia (2024)

Tabla 7

Variables de investigación segundo objetivo

Objetivo Establecer el estado meta de la modernización de la arquitectura para atender los requerimientos funcionales y no funcionales que sean identificados mediante el FODAL.				
Nombre de la variable	Tipo	Indicador	Detalle	Definición instrumental
Requerimientos funcionales	Cualitativa/ Cuantitativa.	Número de funciones requeridas y su nivel de prioridad.	Características específicas del software que deben ser implementadas para cumplir con las necesidades de los usuarios.	Recopilación de los requerimientos funcionales mediante talleres con el líder tecnológico, entrevistas con usuarios actuales/finales.
Requerimientos no funcionales	Cualitativa/ Cuantitativa.	Indicadores de rendimiento, usabilidad, seguridad, y escalabilidad.	Atributos del software que determinan cómo debe funcionar, incluyendo aspectos como rendimiento, seguridad y mantenibilidad.	Identificación de los requerimientos no funcionales mediante revisiones de documentación técnica.
Prioridad de requerimientos	Cuantitativa.	Clasificación de los requerimientos según su importancia y urgencia.	Establecimiento de niveles de prioridad para la implementación de los requerimientos funcionales y no funcionales.	Se empleará la técnica <i>Must have, Should have, Could have, Won't have</i> (MoSCoW) en sesiones de trabajo con el líder tecnológico para priorizar los requerimientos según su impacto en la modernización del software.

Nota. Creación propia (2024)

Tabla 8

Variables de investigación tercer objetivo

Objetivo		Valorar las mejores prácticas de la industria sobre arquitectura de software para identificar los componentes a considerar en la modernización.		
Nombre de la variable	Tipo	Indicador	Detalle	Definición instrumental
Buenas prácticas de la industria identificadas	Cualitativa.	Número de prácticas relevantes recopiladas.	Revisión de las buenas prácticas reconocidas a nivel internacional, por ejemplo: ISO, OWASP, TOGAF.	Se identifican mediante revisión bibliográfica y análisis de estándares internacionales de arquitectura de software.
Componentes críticos de la arquitectura	Cualitativa.	Número y tipo de componentes clave.	Componentes fundamentales para modernizar, como capas, módulos, servicios y bases de datos.	Se define a través de la evaluación de la arquitectura actual y su comparación con modelos de referencia y estándares.

Nota. Creación propia (2024).

Tabla 9

Variables de investigación cuarto objetivo

Objetivo Construir una propuesta de modernización de la arquitectura del software para satisfacer todas las mejoras identificadas en el FODAL y los requerimientos elicitados.				
Nombre de la variable	Tipo	Indicador	Detalle	Definición instrumental
Estructura de la propuesta de modernización	Cualitativa.	Descripción de los componentes arquitectónicos.	Detalle de la nueva estructura propuesta, tecnologías seleccionadas y componentes clave.	Documento técnico que incluye un diseño de arquitectura basado en las mejoras identificadas y los requerimientos.
Compatibilidad con los requerimientos del FODAL	Cuantitativa	Porcentaje de mejoras propuestas.	Nivel de correspondencia entre las mejoras del FODAL y las propuestas de modernización.	Se utiliza un mapa de alineación entre las mejoras identificadas en el FODAL y la propuesta técnica.
Requerimientos funcionales y no funcionales	Cualitativa	Inclusión de todos los requerimientos en la propuesta	Evaluación de cómo la propuesta aborda los requerimientos funcionales y no funcionales.	Documento que detalla cómo cada requerimiento se satisface dentro de la propuesta de modernización.
Uso de buenas prácticas de la industria	Cuantitativa	Porcentaje de adopción de buenas prácticas	Cumplimiento con estándares de arquitectura y tendencias de la industria.	Se aplica una lista de verificación con las mejores prácticas de la industria.

Nota. Creación propia (2024)

3.7. Técnicas e instrumentos de recolección de datos

Para la recolección de datos en el proyecto de modernización de la arquitectura del software Gestemed, se emplearán tanto técnicas cualitativas como cuantitativas. La elección de estas técnicas está orientada a obtener una comprensión integral y detallada de las necesidades, problemas y oportunidades presentes en la arquitectura actual del software, así como a identificar los requisitos para su modernización.

Tabla 10

Técnicas de recolección de datos

Técnicas de recolección de datos	
Entrevistas	Se realizarán entrevistas con los colaboradores de Key Tech, incluyendo al líder tecnológico, desarrolladores y usuarios del sistema Gestemed. Estas entrevistas permitirán recopilar información detallada sobre la arquitectura actual del software, así como sobre los requerimientos funcionales y no funcionales que se necesitan para la modernización.
Grupos focales	Se organizarán grupos focales con usuarios finales del software para entender sus experiencias con el sistema actual, sus principales desafíos y sus expectativas para la modernización. Los grupos focales ayudarán a identificar áreas específicas de mejora y a priorizar funcionalidades.
Análisis de documentación	Se realizará una revisión de la documentación interna del software Gestemed, incluyendo informes técnicos, código fuente, diagramas de arquitectura y registros de mantenimiento. Esta técnica permitirá entender la estructura técnica del software y sus limitaciones actuales.
Análisis FODAL	Se utilizará esta técnica para evaluar la situación actual del software Gestemed y detectar áreas de oportunidad para la modernización. Este análisis se basará en la información obtenida de las entrevistas, los grupos focales y la revisión documental.
Talleres de elicitación de requerimientos	Se organizarán talleres colaborativos con el líder tecnológico y usuarios finales para definir los requerimientos funcionales y no funcionales del sistema modernizado.

Nota. Creación propia (2024)

Tabla 11

Instrumentos de recolección de datos

Instrumentos de recolección de datos	
Guía de entrevista	<p>Este instrumento contiene preguntas abiertas y cerradas que se utilizaron durante las entrevistas con los colaboradores clave de Key Tech. Las preguntas se diseñaron para obtener información sobre la situación actual del software y las expectativas para su modernización.</p> <p>Este instrumento se encuentra en la sección Apéndice B. Guía de entrevista</p>
<p>Enlace a la entrevista con respuestas</p> Guía de satisfacción	<p>Se desarrolló un cuestionario dirigido a los usuarios finales del sistema para evaluar su nivel de satisfacción con la interfaz, funcionalidad y rendimiento del software actual.</p> <p>Este instrumento se encuentra en la sección Apéndice B. Guía de entrevista Apéndice D. Enlace a la entrevista con respuestas</p> <p>Guía de satisfacción</p>
Guía para taller de elicitación de requerimientos	<p>La Guía para taller de elicitación de requerimientos se utilizó como un instrumento de recolección de datos porque permite recopilar de manera sistemática y estructurada la información necesaria sobre las necesidades, expectativas y características que los usuarios actuales/finales y requieren para el software.</p> <p>Este instrumento se encuentra en la sección Apéndice A. Guía para taller de elicitación de requerimientos</p>
Lista de verificación de buenas prácticas	<p>Este instrumento se aplicó para evaluar la adherencia del sistema modernizado a las buenas prácticas internacionales en arquitectura de software, seguridad y escalabilidad.</p> <p>Este instrumento se encuentra en la sección Apéndice D. Lista de verificación de buenas prácticas</p>
PESTEL	<p>Herramienta de análisis que evalúa factores Políticos, Económicos, Sociales, Tecnológicos, Ambientales y Legales.</p> <p>Este instrumento se encuentra en la sección Apéndice H. PESTEL</p>
Revisión documental	<p>Proceso de evaluación y análisis de documentos, informes previos y estándares relevantes para el proyecto.</p> <p>Este instrumento se encuentra en la sección Apéndice E. Revisión documental</p>
Benchmarking	<p>Proceso de comparación sistemática con otras organizaciones o proyectos que son reconocidos como líderes en la industria.</p> <p>Este instrumento se encuentra en la sección Apéndice F. <i>Benchmarking</i></p>

Nota. Creación propia (2024)

3.8. Procedimiento metodológico de la investigación

El procedimiento metodológico de la investigación para la modernización de la arquitectura del software Gestemed se organizó en varias fases clave, cada una diseñada para abordar aspectos específicos del proyecto de manera sistemática y rigurosa.

3.8.1. Fase 1: Revisión documental y análisis del contexto actual

En esta fase se busca comprender en detalle la situación actual del software Gestemed, incluyendo su arquitectura y funcionamiento. Se realizó una revisión exhaustiva de toda la documentación interna disponible, como informes técnicos, diagramas de arquitectura y registros de mantenimiento. El objetivo fue identificar los aspectos clave que necesitan mejorarse o modernizarse. Paralelamente, se aplicó el análisis FODAL para evaluar de manera integral el estado actual del software, permitiendo identificar tanto los aspectos positivos que se deben conservar, como aquellos que requieren atención prioritaria. Esta fase establece una base sólida para guiar las decisiones durante el proceso de modernización.

3.8.2. Fase 2: Elicitación y análisis de requerimientos

Esta fase se enfocó en identificar los requerimientos funcionales y no funcionales necesarios para la modernización del software Gestemed. Para ello, se realizaron entrevistas semiestructuradas con el líder tecnológico y el desarrollador para recopilar información técnica relevante. Además, se organizaron grupos focales con usuarios finales del sistema, con el fin de captar sus experiencias y sugerencias de mejora. A través de talleres colaborativos, se elicitaban los requerimientos que garantizarán que el software modernizado cumpla con las expectativas de los usuarios y las necesidades operativas. Esta fase asegura que las necesidades de todos los interesados sean escuchadas y debidamente integradas en la propuesta.

3.8.3. Fase 3: Análisis de alternativas tecnológicas

Una vez definidos los requerimientos, en esta fase se realizó un análisis de las alternativas tecnológicas más adecuadas para la modernización de Gestemed. Se investigaron arquitecturas de referencia modernas, como microservicios y SOA, comparándolas con la arquitectura actual del software. Se revisaron también las mejores prácticas de la industria, basadas en estándares como ISO/IEC 12207, 25010 y 27001, para garantizar que la solución propuesta cumpliera con los más altos estándares de calidad, seguridad y eficiencia. El análisis incluyó la evaluación de tecnologías emergentes, asegurando que la solución seleccionada fuera escalable, compatible con las infraestructuras actuales y adaptable a futuros cambios tecnológicos.

3.8.4. Fase 4: Diseño de la propuesta de modernización

En esta fase, se diseñó una propuesta técnica que integrara los requerimientos y las tecnologías seleccionadas durante las fases anteriores. El diseño de la nueva arquitectura del software se detalla especificando los componentes clave, como capas, módulos, servicios y bases de datos. Se elabora un plan de modernización que incluye las fases del proceso, los cambios que se deben implementar en la infraestructura tecnológica, y las estrategias para mitigar riesgos y gestionar el cambio dentro de la organización. Esta propuesta constituye una guía técnica clara para la futura implementación del sistema, asegurando que se cubran todos los aspectos críticos identificados durante el análisis.

3.8.5. Fase 5: Validación y refinamiento de la propuesta

La propuesta de modernización desarrollada en la fase anterior es presentada a Key Tech. En esta fase, se recibe *feedback* a través de entrevistas y sesiones de validación, donde se discutieron los componentes del diseño propuesto. Los comentarios y observaciones obtenidos fueron analizados para ajustar y mejorar la propuesta inicial, asegurando que sea viable y estuviese alineada con las expectativas de los usuarios y las capacidades tecnológicas de la organización. Este proceso iterativo permitió afinar la propuesta antes de la documentación final.

3.8.6. Fase 6: Documentación y entrega final

En la última fase, se documentó de manera exhaustiva todo el proceso realizado, desde la evaluación inicial hasta la propuesta final de modernización. El informe incluyó el análisis de la situación actual, los requerimientos elicitados, la evaluación de tecnologías y el diseño arquitectónico detallado del nuevo sistema. Se presenta la propuesta final de modernización a Key Tech y se entregan todos los documentos técnicos que servirán de guía para la futura implementación. Este informe final es el resultado consolidado del proceso investigativo, proporcionando a la empresa una hoja de ruta clara y detallada para la modernización del software Gestemed.

3.9. Operacionalización de las variables o categorías

En este apartado, se detallan las variables clave que se consideran en la investigación para la modernización de la arquitectura del software Gestemed en Key Tech.

Tabla 11

Matriz de operacionalización de variables

Matriz de operacionalización de variables				
Objetivo específico	Variable	Fase de investigación	Instrumentos utilizados	Sujetos de investigación relacionados
Analizar la situación actual de la arquitectura de software Gestemed para el diagnóstico técnico mediante FODAL	Fortalezas	Fase 1: Revisión documental y análisis contextual.	Guía de entrevista Enlace a la entrevista con respuestas Guía de satisfacción	Líder tecnológico, usuario final.
	Oportunidades		Guía de entrevista Enlace a la entrevista con respuestas Guía de satisfacción Benchmarking	Líder tecnológico.
	Debilidades		Enlace a la entrevista con respuestas Guía de satisfacción	Usuario final.
	Amenazas		Guía de entrevista PESTEL	Líder tecnológico.
	Limitaciones		Guía de entrevista	Líder tecnológico.
	Capacidades de integración existentes		Guía de entrevista	Líder tecnológico
	Interfaces de integración requeridas		Guía de entrevista	Líder tecnológico
Establecer el estado meta de la modernización de la arquitectura para atender los requerimientos funcionales y no funcionales identificados mediante FODAL	Requerimientos funcionales	Fase 2: Elicitación y análisis de requerimientos.	Guía de entrevista Revisión documental	Desarrollador, líder tecnológico, usuarios finales.
	Requerimientos no funcionales		Guía de entrevista Revisión documental	Líder tecnológico, desarrollador, usuarios finales.
	Prioridad de requerimientos		Guía para taller de elicitación de requerimientos	Líder tecnológico, desarrollador.

Propuesta de modernización de arquitectura del software Gestemed en Key Tech

Objetivo específico	Variable	Fase de investigación	Instrumentos utilizados	Sujetos de investigación relacionados
Valorar las mejores prácticas de la industria sobre arquitectura de software para identificar los componentes a considerar en la modernización	Buenas prácticas de la industria	Fase 3: Análisis de alternativas tecnológicas.	Revisión documental	Líder tecnológico, desarrollador.
	Componentes críticos de la arquitectura			
Construir una propuesta de modernización de la arquitectura del software para satisfacer todas las mejoras identificadas en el FODAL y los requerimientos elicitados.	Estructura de la propuesta de modernización	Fase 4: Diseño de la propuesta de modernización.	Guía de entrevista Enlace a la entrevista con respuestas Guía de satisfacción Revisión documental Benchmarking PESTEL	Líder tecnológico, desarrollador.
	Compatibilidad con los requerimientos del FODAL		Mapa de alineación de requerimientos.	
	Uso de buenas prácticas de la industria		Lista de verificación de buenas prácticas.	

Nota. Creación propia (2024)

3.10. Tabla resumen del procedimiento metodológico de la investigación

La Tabla 12 de resumen del procedimiento metodológico de la investigación ofrece una visión consolidada de los pasos clave y enfoques adoptados durante el desarrollo del estudio.

Tabla 12

Tabla resumen del procedimiento metodológico de la investigación

Tabla resumen del procedimiento metodológico de la investigación			
Fase	Descripción	Actividades claves	Resultados esperados
Revisión documental y análisis	Evaluar la situación actual del software Gestemed, su arquitectura y funcionamiento.	Revisión de documentación técnica y operativa.	FODAL.
Elicitación y análisis de requerimientos	Identificar los requerimientos funcionales y no funcionales para la modernización del software.	Realización de entrevistas, organización de grupos focales con usuarios, talleres de elicitación de requerimientos.	Lista detallada de requerimientos funcionales y no funcionales, priorización de los requerimientos.
Análisis de alternativas tecnológicas	Evaluar alternativas de arquitectura de software y seleccionar tecnologías modernas.	Investigación de arquitecturas de referencia, revisión de buenas prácticas y normativas internacionales.	Selección de tecnologías y arquitecturas adecuadas para la modernización, lista de buenas prácticas adoptadas.
Diseño de la propuesta de modernización	Crear una propuesta técnica para la modernización del software Gestemed.	Diseño de la nueva arquitectura del software, elaboración del plan de modernización, definición de componentes clave.	Propuesta técnica detallada, que incluye componentes, fases y tecnologías seleccionadas, plan de implementación.
Validación y refinamiento de la propuesta	Validar la propuesta con los principales interesados y ajustar según el <i>feedback</i> recibido.	Presentación de la propuesta a los interesados, recopilación de <i>feedback</i> mediante entrevistas y sesiones de validación. .	Propuesta ajustada y validada, integración de mejoras basadas en el <i>feedback</i> recibido.
Documentación y entrega final	Documentar y presentar los resultados y la propuesta final. .	Elaboración del informe final, presentación de los resultados a Key Tech, documentación técnica del proceso.	Informe final detallado, propuesta técnica completa para la modernización del software. Informe de viabilidad del proyecto.

Nota. Creación propia (2024)

4. Análisis de Resultados

Este capítulo presenta los resultados obtenidos a partir de las actividades realizadas y los instrumentos utilizados, según lo descrito en el Marco metodológico. Además, se comparan estos resultados con la información del Marco conceptual, donde se definieron los conceptos que abordan el problema planteado.

El análisis de los resultados sigue la metodología establecida en el marco metodológico, organizada según las fases identificadas en la sección de procedimiento metodológico de la investigación. Para este capítulo, se utilizan tres de las cinco fases establecidas: Fase 1: Revisión documental y análisis, Fase 2: Elicitación y análisis de requerimientos y Fase 3: Análisis de alternativas tecnológicas.

Cada resultado de las actividades detalla las técnicas empleadas, los instrumentos utilizados, los datos recopilados, los resultados obtenidos y su respectivo análisis.

4.1. Fase 1: Revisión documental y análisis contextual

En esta fase se realizó una entrevista con el líder tecnológico y usuarios, se aplicó un *Benchmarking*, se aplicó una Guía de satisfacción con el fin de determinar la situación actual del software. Gracias a los distintos instrumentos aplicados anteriormente se obtiene el FODAL, esto instrumentos se encuentran en la sección de los apéndices: Apéndice B. Guía de entrevista, Apéndice D. Guía de satisfacción, Apéndice H. *Benchmarking*, Apéndice K. PESTEL.

4.1.1. Resultados guía de entrevista

En el apéndice C. Respuestas de entrevista con el líder tecnológico de Key Tech pone en evidencia varias áreas clave que requieren atención para optimizar el funcionamiento del sistema y mejorar la experiencia del usuario. Uno de los puntos más importantes que se destaca es que el software cumple con sus funciones básicas de manera adecuada, especialmente en lo que respecta al registro de equipos médicos, una funcionalidad que ha sido señalada como uno de los aspectos más útiles y eficientes del sistema. Sin embargo, a pesar de este éxito en las funcionalidades fundamentales, el software enfrenta una serie de limitaciones que afectan tanto su rendimiento como su competitividad en el mercado actual.

En términos de estabilidad, Gestemed presenta graves problemas que impactan de manera negativa el trabajo diario. El sistema experimenta frecuentes periodos extensos de carga y colapsos de memoria, lo que hace que los usuarios enfrenten interrupciones constantes en su flujo de trabajo. Esto ha sido calificado de manera muy negativa por el líder tecnológico, quien menciona que la estabilidad es uno de los puntos más débiles del software, con una puntuación de 1/5. Estos problemas de rendimiento hacen que sea difícil para los usuarios realizar tareas complejas o personalizadas, lo que también afecta la percepción de la eficiencia y fiabilidad del sistema.

Otro aspecto crítico es la falta de integración del software con otras plataformas o sistemas que ya existen en la empresa. En la actualidad, Gestemed no cuenta con APIs públicas o privadas ni soporta protocolos de comunicación modernos como REST o SOAP, lo que impide que pueda interactuar con otros sistemas importantes como ERPs o herramientas de análisis de datos como Power BI. Esto representa una limitación significativa, ya que, en el entorno tecnológico actual, la integración es fundamental para la eficiencia operativa y la capacidad de generar valor a partir de la interoperabilidad de distintos sistemas. Esta falta de integración reduce la competitividad del software en el mercado, haciéndolo menos atractivo para empresas que buscan soluciones conectadas y dinámicas.

Además de la falta de integración, el líder tecnológico también señala que el software tiene dificultades en la funcionalidad de rastreo de equipos médicos. Si bien el registro de los equipos funciona bien, el rastreo de estos ha sido descrito como ineficiente y obsoleto. El líder tecnológico sugiere que el software debe aprovechar tecnologías emergentes como GPS y códigos QR para mejorar esta funcionalidad. La implementación de estas tecnologías no solo resolvería los problemas actuales, sino que también proporcionaría al software una ventaja competitiva

significativa en un sector que cada vez más valora el uso de herramientas tecnológicas avanzadas para mejorar la trazabilidad y control de activos.

Otro desafío importante es la falta de capacitación para los usuarios del sistema. El líder tecnológico menciona que no ha recibido suficiente formación sobre cómo utilizar el software de manera eficiente, lo que sugiere que el sistema presenta una curva de aprendizaje pronunciada. Además, señala que es necesario mejorar primero la documentación del software para que los usuarios puedan entender mejor sus funcionalidades y optimizar su uso. Esto resalta la importancia de tener un plan de soporte y documentación sólido, que permita a los usuarios maximizar el valor del software y reducir las frustraciones relacionadas con la falta de conocimiento o capacitación adecuada.

Por otro lado, a pesar de las limitaciones actuales, el líder tecnológico identifica varias oportunidades de mejora para el software. Entre ellas, se destaca la posibilidad de integrar tecnologías emergentes como Inteligencia Artificial (IA) y Big Data, que podrían optimizar los procesos y ofrecer análisis avanzados de los datos generados por el sistema. Estas tecnologías emergentes no solo mejorarían la funcionalidad del software, sino que también lo posicionarían a la vanguardia del mercado, ofreciendo un producto más competitivo y alineado con las tendencias actuales. Además, el líder tecnológico menciona la posibilidad de integrar el software con plataformas en la nube como Amazon AWS, lo que abriría nuevas oportunidades de escalabilidad y flexibilidad para la implementación de soluciones tecnológicas avanzadas.

Finalmente, uno de los puntos negativos que resalta el líder tecnológico es la seguridad de los datos en Gestemed. Aunque el software presenta limitaciones importantes en otros aspectos, la seguridad ha sido valorada con una calificación de 2/5. Esto indica que el sistema ofrece un nivel adecuado de protección para la información crítica, lo que es crucial para mantener la confianza de los usuarios y asegurar el cumplimiento de normativas de protección de datos.

4.1.2. Resultados guía de satisfacción

El análisis detallado según el apéndice E. Respuestas de la guía de satisfacción sobre el uso del software Gestemed revela una serie de áreas críticas que requieren atención para mejorar su funcionamiento y la satisfacción de los usuarios. Uno de los puntos más destacados es la percepción general de la facilidad de uso del sistema, la cual ha sido calificada como limitada. Los usuarios han manifestado dificultades al navegar y utilizar la interfaz, lo que se refleja en una baja puntuación en términos de usabilidad. Este problema afecta de manera particular a los usuarios menos experimentados en tecnología, quienes encuentran el software poco intuitivo. Esta situación indica la necesidad de realizar ajustes importantes en el diseño de la interfaz, simplificando su estructura y mejorando la accesibilidad para distintos perfiles de usuarios. La modernización de la interfaz no solo mejoraría la experiencia de los usuarios actuales, sino que también facilitaría la adopción por parte de nuevos usuarios.

A pesar de los desafíos en cuanto a usabilidad, los usuarios han otorgado una calificación moderadamente positiva a la capacidad del sistema para satisfacer las necesidades diarias. Esto sugiere que, aunque el software enfrenta problemas técnicos, las funcionalidades básicas resultan útiles y cumplen con los requisitos operativos. Sin embargo, la percepción positiva no elimina el hecho de que el sistema es percibido como lento en su rendimiento, especialmente cuando se trata de acceder a los datos. La baja calificación en la velocidad de respuesta señala que el tiempo necesario para realizar tareas críticas es mayor de lo esperado. Este problema se agrava por los errores frecuentes que experimentan, lo que genera desconfianza en la fiabilidad del sistema. Por lo tanto, optimizar el rendimiento del software para garantizar una mayor rapidez en la ejecución de tareas se convierte en una prioridad.

Uno de los aspectos más alarmantes es la ausencia de soporte técnico adecuado. Los usuarios han calificado este aspecto con la puntuación más baja posible, mencionando que no hay acceso a soporte por parte del fabricante original del software. Esta falta de soporte es alarmante, ya que los usuarios no tienen a quién acudir cuando enfrentan problemas graves, lo que afecta la continuidad operativa. El acceso a un soporte técnico de calidad es fundamental para resolver problemas con rapidez y garantizar que el sistema funcione sin interrupciones. La situación actual sugiere que la empresa debe considerar seriamente la posibilidad de buscar nuevos proveedores de soporte o desarrollar capacidades internas para atender las necesidades técnicas de los usuarios. La creación de un equipo interno de soporte técnico no solo permitiría una respuesta más ágil, sino que también ofrecería la posibilidad de personalizar el servicio según las características específicas del sistema y la organización.

Otro aspecto relevante es la falta de actualizaciones y mejoras en el software. Los usuarios sienten que el sistema está estancado y no se adapta a las necesidades cambiantes del entorno tecnológico. Esto ha generado una demanda por nuevas funcionalidades que permitan gestionar de manera más eficiente los datos obtenidos y ofrecer información más detallada sobre el uso del sistema. La implementación de actualizaciones regulares y el desarrollo de nuevas funcionalidades no solo mejoraría la experiencia del usuario, sino que también alinearía el sistema con las tendencias tecnológicas actuales, ofreciendo un producto más competitivo y funcional. Además,

la falta de integración con otros sistemas utilizados en la organización es otro aspecto que limita la optimización de procesos. Actualmente, el software opera de manera aislada, lo que impide una gestión más integral de la información y procesos operativos. Integrar Gestemed con otros sistemas podría permitir un flujo de trabajo más eficiente y un mayor aprovechamiento de los recursos tecnológicos disponibles en la organización.

A pesar de los problemas identificados, uno de los pocos aspectos positivos que los usuarios resaltan es la calidad de la documentación del software. Los manuales y guías han sido bien valorados, lo que sugiere que, a pesar de las deficiencias del sistema, los recursos de soporte documental son claros, completos y útiles para resolver dudas. Esto destaca la importancia de mantener y mejorar la documentación del software, ya que es una herramienta que puede compensar parcialmente las carencias en otras áreas, como el soporte técnico.

Finalmente, el análisis revela la necesidad de abordar de manera prioritaria los problemas de conectividad y estabilidad del software. Los usuarios han reportado problemas de estabilidad que resultan en caídas frecuentes del sistema, lo que afecta gravemente su capacidad para completar tareas sin interrupciones. Estas caídas no solo generan frustración, sino que también tienen un impacto directo en la productividad y confianza de los usuarios en el sistema. Resolver los problemas de estabilidad debe ser una prioridad inmediata para garantizar una operación fluida y reducir los tiempos de inactividad.

4.1.3. Resultados de *benchmarking*

El análisis comparativo de Gestemed frente a otras soluciones (Apéndice H. *Benchmarking*) de gestión de mantenimiento y activos, como Maintenance Connection, Infor EAM, eMaint CMMS, Hippo CMMS, UpKeep, EZOfficeInventory y Asset Panda, revela importantes fortalezas y debilidades en cada criterio evaluado, proporcionando una visión clara de cómo se posiciona Gestemed en el mercado de software de mantenimiento y gestión de activos.

En cuanto a funcionalidades, Gestemed cubre aspectos básicos como la gestión de equipos médicos, mantenimiento preventivo y órdenes de trabajo. Sin embargo, se destaca su limitación en la integración con tecnologías modernas, lo que le resta competitividad. Por otro lado, Maintenance Connection e Infor EAM ofrecen soluciones más robustas, con análisis predictivo y gestión avanzada de activos, mientras que otros competidores, como UpKeep y Asset Panda, incluyen características como el rastreo de activos mediante códigos QR y funcionalidades móviles, que incrementan la flexibilidad y eficiencia operativa.

La usabilidad de Gestemed es otro de sus puntos débiles, ya que su interfaz desactualizada dificulta la experiencia del usuario. En comparación, competidores como Maintenance Connection, Hippo CMMS, y eMaint CMMS ofrecen interfaces intuitivas y fáciles de usar, lo que mejora significativamente la interacción con el software. Soluciones como UpKeep y Asset Panda se destacan especialmente por ofrecer una experiencia de usuario óptima a través de dispositivos móviles, una característica clave en un entorno de trabajo cada vez más móvil.

En términos de compatibilidad tecnológica, Gestemed presenta grandes desafíos debido a su arquitectura desactualizada, lo que limita su capacidad para integrarse con tecnologías emergentes. En contraste, Infor EAM y UpKeep son altamente compatibles con sistemas y aplicaciones modernas, lo que les da una ventaja competitiva significativa. EZOfficeInventory y Asset Panda también destacan por su capacidad de integración con dispositivos móviles y sistemas empresariales, lo que les permite ser más versátiles en distintos entornos operativos.

La seguridad de Gestemed es otro aspecto que requiere atención, ya que su nivel de protección de datos es básico y presenta deficiencias en encriptación y autenticación. Por otro lado, Infor EAM y UpKeep cumplen con estándares avanzados como ISO/IEC 27001, lo que asegura una protección robusta de los datos sensibles. La seguridad es crucial en la gestión de activos y mantenimiento, especialmente en sectores que manejan información crítica, lo que coloca a Gestemed en desventaja en este aspecto.

En cuanto al costo, Gestemed tiene un precio moderado, aunque la necesidad de invertir en su modernización podría incrementar los costos a largo plazo. En comparación, UpKeep y eMaint CMMS ofrecen modelos de suscripción flexibles y asequibles, lo que los hace atractivos para pequeñas y medianas empresas. Infor EAM presenta costos más elevados, pero justificados por las funcionalidades avanzadas y el soporte integral que ofrece, lo que lo convierte en una opción preferida para grandes empresas.

El soporte técnico de Gestemed es limitado y se caracteriza por tiempos de respuesta lentos, lo que afecta negativamente la experiencia del usuario. Esto contrasta con soluciones como Maintenance Connection e Infor EAM, que ofrecen soporte técnico disponible 24/7 y atención integral en varios idiomas. Un servicio de soporte técnico eficiente es esencial para resolver problemas rápidamente y garantizar la continuidad operativa, lo que coloca a Gestemed en una situación desventajosa frente a sus competidores.

En términos de escalabilidad, Gestemed también enfrenta limitaciones debido a su arquitectura actual, lo que impide su crecimiento y adaptación a las necesidades cambiantes de las empresas. Por otro lado, soluciones como Maintenance Connection y UpKeep son altamente escalables, lo que les permite ajustarse fácilmente a empresas de diferentes tamaños. La capacidad de escalar un sistema es fundamental para empresas en expansión, ya que necesitan soluciones que crezcan con ellas y se adapten a sus nuevas demandas.

Finalmente, la experiencia de usuario (UX) en Gestemed requiere mejoras significativas. Su interfaz y diseño no cumplen con las expectativas modernas, lo que dificulta su adopción y uso eficiente. En comparación, la mayoría de los competidores, como Maintenance Connection, Hippo CMMS, y UpKeep, ofrecen una experiencia de usuario fluida, intuitiva y diseñada para maximizar la eficiencia. Además, la capacidad de personalización y el uso de dispositivos móviles en Asset Panda y EZOfficeInventory mejoran considerablemente la experiencia del usuario.

4.1.4. Resultados del PESTEL

El análisis apéndice J. PESTEL aplicado a la modernización de Gestemed revela varios factores clave que deben considerarse para asegurar el éxito del proceso. En el ámbito político, el software debe cumplir con estrictas normativas y regulaciones del sector salud, especialmente en la protección de datos sensibles de los pacientes. La modernización debe adherirse a regulaciones como el Reglamento General de Protección de Datos (GDPR) y otras leyes de privacidad, garantizando que la información esté protegida adecuadamente. Además, es fundamental revisar los contratos vigentes para asegurarse de que las modificaciones sean compatibles con las nuevas tecnologías y las políticas de privacidad existentes.

Desde el punto de vista económico, es esencial evaluar los costos asociados a la modernización, incluyendo posibles inversiones adicionales. El sistema actual genera ineficiencias que pueden traducirse en pérdidas económicas, por lo que modernizar el software no solo mejorará la competitividad, sino que también aumentará la productividad y satisfacción de los usuarios. Mantener la competitividad en el mercado requiere ofrecer soluciones actualizadas que respondan a las demandas cambiantes de los clientes, lo que a largo plazo puede generar retornos económicos significativos.

En el factor social, la aceptación por parte de los usuarios es crucial. Los técnicos y administrativos que interactúan diariamente con el software deben adaptarse fácilmente a las nuevas versiones. Para lograr esto, la interfaz de usuario debe ser más intuitiva y amigable, alineándose con las expectativas actuales. La creciente demanda de soluciones tecnológicas modernas en el entorno empresarial y de salud obliga a que el software evolucione para cumplir con las expectativas de los usuarios, mejorando su experiencia y evitando resistencia al cambio.

En el aspecto tecnológico, Gestemed enfrenta desafíos significativos debido a su arquitectura desactualizada, lo que limita su eficiencia y capacidad de integración con tecnologías emergentes. La modernización debe enfocarse en integrar nuevas tecnologías como la nube, la inteligencia artificial y herramientas de análisis de datos, que mejorarán el rendimiento y la capacidad del software para gestionar grandes volúmenes de información. También es crucial garantizar que el sistema sea compatible con otros dispositivos y sistemas utilizados en el sector salud, lo que aumentaría su funcionalidad y relevancia en el mercado.

El factor ecológico sugiere que es importante adoptar soluciones que reduzcan el consumo energético del sistema y optimicen su infraestructura tecnológica para minimizar el impacto ambiental. En un entorno donde la sostenibilidad es cada vez más valorada, se deben implementar prácticas de eficiencia energética y asegurarse de que cualquier equipo obsoleto se elimine de manera segura y sostenible. Además de los beneficios ambientales, estas medidas también pueden generar ahorros económicos a largo plazo.

Finalmente, el aspecto legal es fundamental, ya que Gestemed debe cumplir con todas las leyes de protección de datos, como la GDPR y la Ley de Protección de Datos Personales, asegurando que la información de los pacientes esté protegida. La modernización debe evitar infringir derechos de propiedad intelectual o licencias de software existentes, y además cumplir con estándares de seguridad como la ISO/IEC 27001, que garantizan la protección de la información sensible.

4.1.5. FODAL

El análisis FODAL es una herramienta estratégica fundamental que permite evaluar de manera integral el estado actual de un sistema, proyecto o proceso. En el contexto del proyecto de modernización de la arquitectura del software Gestemed en KeyTech, el FODAL ofrece un marco estructurado para diagnosticar los factores críticos que influyen en el desempeño actual del sistema. A través del análisis de la fase uno, es posible identificar las fortalezas que respaldan el proceso de modernización, las oportunidades de mejora dentro del entorno tecnológico y de mercado, así como las debilidades y amenazas que podrían dificultar el avance del proyecto. Además, la inclusión de limitaciones permite delinear los aspectos que deben ser gestionados para evitar impactos negativos en los resultados esperados.

Tabla 13
Fortalezas

FODAL
Fortalezas
<ul style="list-style-type: none"> • El registro y control de equipos médicos es una fortaleza ya que el software es efectivo en la asignación de códigos únicos y en la gestión de los equipos médicos a lo largo de su ciclo de vida, utilizando identificadores de radiofrecuencia y códigos QR. Esto permite un seguimiento detallado del mantenimiento preventivo y correctivo, optimizando la trazabilidad. • La base de datos histórica consolidada es una fortaleza ya que el sistema ha acumulado datos significativos a lo largo de los años, convirtiéndose en un activo valioso para el análisis y toma de decisiones sobre el rendimiento, estado y vida útil de los equipos médicos. • La capacidad de almacenamiento y recuperación de datos es una fortaleza ya que Gestemed posee una estructura de datos sólida que facilita la recuperación y el análisis de la información, permitiendo generar informes históricos y operativos de forma eficiente. • El sistema cuenta con mecanismos de autenticación y autorización por roles, proporcionando un nivel de control de acceso que reduce el riesgo de manipulación no autorizada de los datos. • El software permite la personalización de ciertos registros y ajustes en la configuración de equipos médicos, lo que facilita la adaptación a distintas necesidades operativas. • Gestemed tiene años de experiencia en el sector de la gestión de equipos médicos, lo que le permite estar alineado con los procesos y requisitos específicos de esta industria. • Gestemed ha demostrado la capacidad de manejar y procesar grandes volúmenes de información, lo cual es fundamental para el seguimiento de múltiples equipos médicos en tiempo real. • La funcionalidad de seguimiento y control de equipos médicos facilita la gestión de inventarios, asegurando que los equipos estén disponibles y en buen estado cuando se necesiten. • Gestemed se adapta a los procesos internos de gestión y mantenimiento de equipos médicos, permitiendo su uso en diferentes contextos operativos y estructuras organizacionales. • La trazabilidad y el registro detallado de actividades y eventos en el software facilitan la realización de auditorías internas y externas, asegurando la conformidad y el control de los procesos.

Nota. Creación propia (2024)

Tabla 14

Oportunidades

FODAL

Oportunidades

- Dado que Gestemed cuenta con una base de datos histórica consolidada de equipos médicos existe una oportunidad para integrar herramientas de inteligencia artificial y análisis de datos predictivos que anticipen posibles fallos y programen mantenimientos preventivos. Esto no solo reducirá el tiempo de inactividad de los equipos, sino que también optimizará la vida útil de los mismos, lo que representa un valor agregado importante para los usuarios del software.
- La posibilidad de conectar Gestemed con sistemas hospitalarios y de gestión empresarial (ERP) ofrece una oportunidad para centralizar la información y mejorar la eficiencia operativa. Al integrarse con estos sistemas, Gestemed puede ofrecer una visión integral de la gestión de equipos médicos en el contexto de la operación hospitalaria, permitiendo un mejor seguimiento de recursos, la planificación de servicios y la optimización de los flujos de trabajo.
- La modernización de la experiencia del usuario y usabilidad es una oportunidad ya que a la hora de rediseñar la interfaz de usuario para hacerla más intuitiva y amigable, basada en la retroalimentación recibida, permitirá que el software sea más accesible, especialmente para usuarios con poca experiencia en tecnología. Al mejorar la usabilidad, Gestemed puede incrementar su adopción y uso efectivo, lo que reducirá errores y tiempos de capacitación, incrementando la satisfacción del usuario y su eficiencia en las tareas diarias.
- Existe la oportunidad de expandir la funcionalidad del software mediante el desarrollo de nuevos módulos, como la gestión de inventarios, gestión de pedidos de reparación, informes de auditoría, y análisis de tendencias en el uso de equipos médicos. Esto permitirá que Gestemed se convierta en una solución integral que cubra más aspectos del proceso de gestión de equipos médicos, brindando un mayor valor a los usuarios y aumentando su competitividad en el mercado.
- La migración a la nube mejorará la accesibilidad y flexibilidad del software, permitiendo a los usuarios acceder al sistema desde cualquier lugar y dispositivo. Además, la nube ofrece escalabilidad y potencia de procesamiento adicional, que puede ayudar a manejar grandes volúmenes de datos y usuarios de manera más eficiente, al mismo tiempo que reduce la necesidad de inversiones en infraestructura local.
- La adopción de tecnologías como el Internet de las cosas (IoT) permitirá el monitoreo en tiempo real de los equipos médicos, mientras que la inteligencia artificial (IA) puede mejorar la toma de decisiones basada en datos históricos y tendencias. La utilización de Big Data permitirá realizar análisis complejos sobre el rendimiento y utilización de los equipos, proporcionando información valiosa para la optimización de procesos.
- Desarrollar e implementar APIs permitirá que el software se integre con otras soluciones tecnológicas del mercado, facilitando la interoperabilidad con sistemas de terceros. Esto no solo

ampliará la funcionalidad del software, sino que también lo convertirá en una opción más atractiva para las organizaciones que buscan soluciones integrales y que necesitan conectarse a diferentes plataformas y herramientas.

- La capacidad de generar reportes y paneles interactivos en tiempo real permitirá a los usuarios monitorear el estado de los equipos médicos, identificar problemas potenciales y tomar decisiones informadas de manera ágil. Esto potenciará la toma de decisiones estratégicas y operativas, mejorando la eficiencia y efectividad de la gestión.
- Las características y funcionalidades de Gestemed tienen el potencial de adaptarse y expandirse a otros sectores más allá del área médica, como la gestión de activos en la industria de manufactura, logística, o servicios públicos. Esto presenta una oportunidad para diversificar y ampliar la base de clientes, lo que incrementará el alcance y la rentabilidad del software.
- Establecer alianzas con proveedores de tecnología, empresas de consultoría o instituciones educativas permitirá acelerar la implementación de nuevas funcionalidades y tecnologías, así como expandir la presencia del software en el mercado. Estas alianzas pueden facilitar el acceso a recursos, conocimientos especializados y canales de distribución, impulsando el crecimiento de Gestemed.
- Aprovechar la oportunidad de certificar el software bajo estándares internacionales de calidad y seguridad (ISO, HL7, etc.) aumentará su confiabilidad y aceptación en el mercado, especialmente en sectores que requieren altos niveles de cumplimiento normativo. Esto permitirá acceder a clientes que buscan soluciones certificadas y confiables.
- La implementación de programas de capacitación más efectivos y la mejora del soporte técnico incrementarán la satisfacción de los usuarios y su capacidad para utilizar el software de manera óptima. Esto podría incluir la creación de tutoriales interactivos, videos explicativos y sesiones de capacitación en línea que faciliten la adopción del software por parte de nuevos clientes y mejoren la experiencia de los actuales.

Nota. Creación propia (2024)

Tabla 15
Debilidades

FODAL
Debilidades
<ul style="list-style-type: none">• El diseño monolítico actual del software significa que todos los componentes y funcionalidades están fuertemente acoplados en un solo sistema. Esto genera grandes dificultades para realizar cambios, actualizaciones o correcciones sin afectar a todo el sistema. Cualquier modificación, por mínima que sea, requiere un despliegue completo del software, lo que aumenta los tiempos de mantenimiento y la posibilidad de introducir nuevos errores, limitando la capacidad de responder a las necesidades cambiantes del negocio o del mercado.• El software opera principalmente mediante procesamiento por lotes (batch), lo que significa que la información no se actualiza de forma inmediata. Esto limita la capacidad de reaccionar ante eventos críticos que requieren atención inmediata, como el seguimiento del estado de equipos médicos en tiempo real. En entornos donde la disponibilidad y precisión de la información son esenciales, esta debilidad impacta directamente en la eficiencia y la toma de decisiones.• La interfaz actual no es intuitiva ni amigable para el usuario, lo que crea una experiencia de uso compleja y difícil, especialmente para aquellos con poca experiencia en tecnología. Esto genera una curva de aprendizaje prolongada, incrementa la probabilidad de errores y reduce la eficiencia en el uso del software. Además, la falta de una interfaz moderna afecta la percepción de los usuarios sobre la calidad del software, haciéndolo parecer desactualizado.• La infraestructura tecnológica del software está desactualizada y no permite la fácil incorporación de tecnologías emergentes, como inteligencia artificial, IoT, o análisis avanzado de datos. Esto limita la capacidad del software para mantenerse competitivo y adaptarse a las demandas actuales de los usuarios, reduciendo su relevancia en un mercado que evoluciona rápidamente.• Gestemed no permite adaptaciones específicas a los requisitos de cada organización o cliente. Esto limita su capacidad para atender las necesidades particulares de diferentes usuarios y sectores, haciendo que las organizaciones busquen otras soluciones que ofrezcan una mayor personalización y adaptación a sus procesos.• El software no cuenta con APIs públicas o privadas que permitan su integración con otros sistemas, lo que dificulta la conexión con herramientas de análisis de datos, sistemas de gestión hospitalaria, y otras plataformas que podrían agregar valor. Esta falta de interoperabilidad limita la capacidad de Gestemed para convertirse en una solución integral en un entorno donde la conectividad y la integración son cruciales.• Gestemed no ofrece herramientas de monitoreo proactivo para la detección y resolución de problemas, lo que significa que los usuarios solo se dan cuenta de los fallos cuando estos ya han afectado la operación. Esto aumenta el tiempo de inactividad y reduce la eficiencia del sistema, afectando la confianza de los usuarios en la estabilidad y fiabilidad del software.

- software está construido sobre tecnologías que han quedado desactualizadas, lo que dificulta la contratación de personal con las habilidades necesarias para su mantenimiento y desarrollo. Esta dependencia incrementa los costos y tiempos asociados con la implementación de mejoras y la resolución de problemas, ya que se requiere personal especializado para trabajar con tecnologías antiguas.
- Los usuarios no cuentan con un soporte técnico eficiente ni con actualizaciones regulares del software, lo que genera frustración y reduce la productividad cuando surgen problemas o fallos. La ausencia de un soporte técnico adecuado también afecta la percepción del software como una solución confiable y actualizada.
- A medida que se incrementa la cantidad de datos y usuarios, el rendimiento del software disminuye significativamente. Esto es un problema grave para organizaciones que necesitan expandir su uso, ya que Gestemed no está diseñado para manejar un crecimiento escalable sin experimentar problemas de rendimiento, como tiempos de respuesta prolongados y caídas del sistema.
- El software no está adaptado para responder rápidamente a los cambios en regulaciones, tecnologías o demandas de los usuarios. Esto afecta la capacidad de la organización para mantenerse actualizada y cumplir con estándares del sector de la salud, especialmente en un entorno altamente regulado y exigente.
- La falta de herramientas que permitan la visibilidad y trazabilidad de los procesos internos dificulta el seguimiento y la optimización de los flujos de trabajo. Esto limita la capacidad de identificar áreas de mejora y ajustar el uso del software para adaptarse a las necesidades operativas de los usuarios.
- Aunque Gestemed cuenta con capacidades básicas de generación de informes, estas no son lo suficientemente avanzadas para satisfacer las necesidades de análisis de las organizaciones. La ausencia de paneles interactivos y reportes en tiempo real limita la capacidad de los usuarios para obtener *insights* detallados y tomar decisiones informadas.
- El software no cuenta con un mecanismo que permita a los usuarios proporcionar retroalimentación de forma estructurada sobre los problemas o áreas de mejora. Esto impide la identificación de oportunidades para optimizar la experiencia del usuario y mejora continua del sistema.
- No se cuenta con programas de capacitación efectivos ni con recursos de formación actualizados, lo que afecta la adopción del software y el aprovechamiento de todas sus funcionalidades. Esto también contribuye a que los usuarios experimenten dificultades en su uso, lo que disminuye la eficiencia y la satisfacción general.

Nota. Creación propia (2024)

Tabla 16
Amenazas

FODAL
Amenazas
<ul style="list-style-type: none"> • El mercado actual cuenta con numerosas soluciones de software como servicio (SaaS) que ofrecen funcionalidades similares a Gestemed, pero con tecnologías más avanzadas, arquitectura moderna y modelos de precios más flexibles. Estos competidores brindan ventajas como actualizaciones automáticas, escalabilidad y una experiencia de usuario superior. Si Gestemed no se moderniza y se adapta rápidamente, corre el riesgo de ser reemplazado por estas alternativas más ágiles y tecnológicamente avanzadas, perdiendo su posición en el mercado. • Las regulaciones en el sector de la salud evolucionan constantemente, y existen normativas estrictas relacionadas con la protección de datos, privacidad y seguridad (por ejemplo, GDPR, HIPAA). El software Gestemed, en su estado actual, podría no cumplir con futuras actualizaciones de estas regulaciones, lo que pone en riesgo su viabilidad y su aceptación en el mercado. El incumplimiento de estos estándares puede llevar a sanciones legales, pérdida de clientes, o la necesidad de realizar costosas modificaciones de emergencia para adaptarse a los nuevos requisitos. • A medida que la tecnología evoluciona, las amenazas cibernéticas también se vuelven más sofisticadas. La arquitectura obsoleta de Gestemed y la falta de actualizaciones regulares lo hacen susceptible a ciberataques, lo que pone en riesgo la confidencialidad, integridad y disponibilidad de los datos de los equipos médicos. Un incidente de seguridad podría tener consecuencias devastadoras, como la pérdida de datos, daño a la reputación y la confianza de los clientes, e incluso responsabilidades legales si se compromete información sensible. • Los usuarios esperan que el software sea cada vez más intuitivo, fácil de usar y capaz de integrarse con otras soluciones tecnológicas. Las expectativas también incluyen la capacidad de acceder al software desde cualquier lugar y dispositivo, utilizando interfaces modernas y procesos simplificados. Si Gestemed no evoluciona para satisfacer estas expectativas, los usuarios podrían optar por soluciones que ofrezcan una experiencia más amigable, actualizada y funcional, lo que afectaría negativamente su cuota de mercado. • La rápida evolución de tecnologías como IA, IoT, <i>big data</i>, y la computación en la nube representan una amenaza si Gestemed no logra adaptarse a estas tendencias. Los competidores que adopten estas tecnologías ganarán una ventaja significativa, ofreciendo soluciones más inteligentes, conectadas y eficientes. El no integrar estas tecnologías en el software podría resultar en la pérdida de competitividad y relevancia en un mercado que valora la innovación y la capacidad de adaptación. • Los clientes del sector de la salud y otras industrias están adoptando modelos de negocio más ágiles y flexibles, y requieren software que pueda adaptarse a estos cambios. Si Gestemed no puede ajustarse a las nuevas formas de trabajo y no permite la personalización ni la integración

con otros sistemas, los clientes podrían migrar a otras soluciones que se adapten mejor a sus necesidades operativas y estratégicas.

- El uso de tecnologías y arquitecturas obsoletas hace que Gestemed sea cada vez más difícil y costoso de mantener y actualizar. Esto representa una amenaza en términos de soporte técnico, ya que podría volverse insostenible a medida que el personal capacitado en estas tecnologías se vuelve escaso. La dependencia de un sistema desactualizado limita la capacidad de reaccionar a cambios en el mercado y amenaza la viabilidad a largo plazo del software.
- Existe un alto riesgo de que quede obsoleto, especialmente porque las organizaciones buscan soluciones que les permitan aprovechar las últimas tecnologías y mantenerse competitivas. La obsolescencia tecnológica implica no solo la pérdida de clientes actuales, sino también la incapacidad de atraer nuevos clientes que buscan soluciones de vanguardia.
- Las organizaciones de salud y otras industrias pueden enfrentarse a recortes presupuestarios y la necesidad de maximizar sus inversiones en tecnología. Los clientes podrían buscar soluciones más económicas o que ofrezcan un mayor retorno de inversión, lo que podría llevarlos a preferir software con modelos de precios más flexibles o que ofrezcan mayor funcionalidad por un costo similar o menor.
- El surgimiento de competidores especializados que ofrecen soluciones altamente adaptadas a nichos específicos de la gestión de equipos médicos representa una amenaza para Gestemed. Estos competidores pueden ofrecer funcionalidades y características personalizadas que satisfacen de manera más precisa las necesidades de ciertos segmentos de mercado, lo que podría hacer que Gestemed pierda clientes que buscan soluciones más específicas.
- Si Gestemed no logra diferenciarse claramente de otras soluciones del mercado, corre el riesgo de ser percibido como un software genérico y poco atractivo. La falta de características únicas y de valor añadido podría llevar a los clientes a elegir otras opciones que ofrezcan más beneficios o que estén más alineadas con sus necesidades y objetivos.
- El mercado de software de gestión de equipos médicos está en constante crecimiento y cada vez más saturado. Con la aparición continua de nuevas soluciones, existe el riesgo de que Gestemed se pierda en un mercado altamente competitivo si no logra mantenerse relevante y adaptarse a las tendencias y demandas del sector.
- Los problemas actuales de estabilidad, rendimiento y usabilidad de Gestemed pueden afectar la percepción de los clientes y su confianza en el producto. En un mercado donde las opiniones y experiencias de los usuarios se comparten rápidamente, una percepción negativa puede disuadir a nuevos clientes y motivar a los actuales a buscar alternativas.

Nota. Creación propia (2024)

Tabla 17
Limitaciones

FODAL
Limitaciones
<ul style="list-style-type: none">• La estructura del código de Gestemed es compleja y está desactualizada, lo que dificulta la comprensión, modificación y optimización del software. Esta complejidad incrementa el tiempo y esfuerzo necesarios para implementar nuevas funcionalidades, corregir errores y mejorar el sistema. Además, el código no está adecuadamente documentado, lo que añade un nivel adicional de dificultad para cualquier equipo que desee trabajar en su mantenimiento o modernización.• El proceso de modernización de Gestemed requiere una inversión significativa en recursos tecnológicos, personal capacitado y tiempo para llevar a cabo las mejoras necesarias. Sin embargo, las restricciones presupuestarias de la organización limitan la capacidad de implementar todas las mejoras de manera simultánea, lo que puede retrasar el proyecto y reducir la posibilidad de abordar todos los problemas críticos de manera efectiva. Esta limitación también podría llevar a la necesidad de priorizar ciertas mejoras sobre otras, dejando aspectos importantes sin atender.• El proceso de modernización y mejora del software requiere profesionales con conocimientos avanzados en tecnologías modernas, arquitectura de software y desarrollo de sistemas. Sin embargo, encontrar y contratar personal capacitado para trabajar con tecnologías obsoletas y estructuras de código complejas resulta difícil. La escasez de profesionales especializados implica que la modernización puede llevar más tiempo y resultar más costosa, afectando la calidad y el ritmo del proceso.• Gestemed depende de tecnologías y herramientas que han quedado desactualizadas y que no cuentan con soporte o actualizaciones por parte de sus fabricantes. Esta dependencia limita la capacidad de integrar el software con sistemas modernos y dificulta la implementación de mejoras, ya que muchas de las tecnologías utilizadas ya no son compatibles con las soluciones actuales. Además, cualquier falla en estas tecnologías obsoletas puede poner en riesgo la operatividad del sistema.• La falta de documentación técnica completa y actualizada del software complica la comprensión de la estructura, el diseño y la lógica del sistema. Esto representa un obstáculo significativo para los desarrolladores y técnicos que intentan implementar mejoras o solucionar problemas, ya que no disponen de una guía clara sobre cómo funciona el sistema o cómo se relacionan sus componentes. La falta de documentación también dificulta la transferencia de conocimiento a nuevos miembros del equipo.• El proyecto de modernización no cuenta con un equipo especializado que trabaje de manera exclusiva en esta tarea, lo que significa que el personal asignado tiene que dividir su tiempo entre la modernización y otras responsabilidades operativas. Esta falta de enfoque y dedicación afecta

la velocidad y eficiencia con la que se pueden realizar las mejoras y podría provocar retrasos o inconsistencias en la ejecución del proyecto.

- La arquitectura actual de Gestemed no permite la integración con otros sistemas o plataformas mediante APIs, lo que limita la posibilidad de conectarse con soluciones externas, como sistemas de gestión hospitalaria, herramientas de análisis de datos, o plataformas de administración de activos. Esta limitación reduce la capacidad del software para adaptarse a las necesidades de los clientes que requieren interoperabilidad con otras soluciones tecnológicas.
- La estructura del software no está diseñada para escalar de manera efectiva a medida que crece el número de usuarios, la cantidad de datos o las funcionalidades requeridas. Esta limitación implica que, a medida que más organizaciones o departamentos adopten el sistema, la eficiencia y el rendimiento pueden verse afectados. La falta de escalabilidad representa un riesgo para el futuro del software, ya que no puede adaptarse al crecimiento de los clientes o a nuevas demandas operativas.
- La modernización del software implica un cambio significativo en la forma de trabajar para los usuarios y el equipo de desarrollo. Es posible que algunos miembros de la organización se resistan a adoptar nuevas tecnologías o métodos de trabajo, lo que puede ralentizar el proceso de modernización y dificultar la implementación de nuevas funcionalidades. Esta resistencia también puede provocar que se mantengan prácticas obsoletas o ineficientes, lo que afecta la capacidad de adaptación del software.
- La infraestructura actual en la que se ejecuta Gestemed puede no ser adecuada para soportar nuevas tecnologías o actualizaciones que se planean implementar. Esto incluye servidores, bases de datos y otros componentes tecnológicos que podrían no tener la capacidad de manejar el aumento en la carga de trabajo o las nuevas funcionalidades. La necesidad de actualizar o reemplazar esta infraestructura representa un costo adicional y un desafío logístico.
- El avance del proyecto de modernización puede depender de decisiones y aprobaciones de partes interesadas externas, como proveedores, clientes o autoridades regulatorias. Esta dependencia puede retrasar la implementación de mejoras y cambios, ya que cualquier retraso o desacuerdo puede impactar negativamente en el cronograma y en el alcance del proyecto.
- La falta de programas de capacitación estructurados y recursos de formación dificulta la preparación de los usuarios y el personal técnico para trabajar con el sistema modernizado. Esto no solo afecta la adopción del software por parte de los usuarios finales, sino que también impide que el equipo de desarrollo adquiera las habilidades necesarias para mantener y mejorar el sistema en el futuro.
- Cualquier cambio en el software debe cumplir con las regulaciones y normativas del sector de la salud y la tecnología de la información. La necesidad de garantizar el cumplimiento puede limitar las opciones disponibles para la modernización y requerir procesos adicionales de revisión y aprobación, lo que aumenta la complejidad y el tiempo necesario para implementar las mejoras.

Nota. Creación propia (2024)

4.2. Fase 2: Elicitación y análisis de requerimientos

En esta fase se realizó una entrevista (Apéndice C. Respuestas de entrevista) con el líder tecnológico, además una revisión documental. Gracias a los distintos instrumentos aplicados anteriormente se obtiene una lista de requerimientos tanto funcionales como no funcionales y su debida priorización. Gracias a los distintos instrumentos aplicados anteriormente se obtiene la lista de requerimientos y su debida priorización, esto instrumentos se encuentran en la sección de los apéndices: Apéndice B. Guía de entrevista, Apéndice G. Revisión documental, Apéndice A. Guía para taller de elicitación de requerimientos.

4.2.1. Resultados guía para taller de elicitación de requerimientos

El apéndice A. Guía para taller de elicitación de requerimientos para la modernización del software Gestemed reveló varios puntos clave que necesitan atención para mejorar la funcionalidad y la experiencia del usuario. En primer lugar, las funcionalidades actuales que se consideran imprescindibles son el registro de equipos médicos, el rastreo de los equipos y la consulta de atributos técnicos, que son fundamentales para la gestión diaria de activos médicos. Sin embargo, se identificaron algunas funcionalidades que casi no se utilizan, como la gestión avanzada de inventarios y las opciones detalladas de reportes, lo que sugiere que estas pueden requerir simplificación o rediseño. Por otro lado, los participantes expresaron interés en nuevas funcionalidades, como el rastreo por GPS, la integración de códigos QR para el registro rápido y una herramienta de análisis predictivo basada en inteligencia artificial, lo cual optimizaría el mantenimiento preventivo.

En cuanto a la interfaz de usuario, el consenso fue que, aunque funcional, la interfaz está desactualizada y no es lo suficientemente intuitiva. Los usuarios sugirieron la necesidad de un diseño más moderno y accesible, especialmente para usuarios con menos habilidades técnicas. También se mencionó que más opciones de personalización mejorarían la experiencia del usuario, permitiendo que cada perfil ajuste la interfaz según sus necesidades específicas.

El rendimiento del sistema fue otro de los puntos críticos abordados. Actualmente, el software es lento en la ejecución de tareas críticas, especialmente cuando se manejan grandes volúmenes de datos, lo que afecta significativamente la eficiencia operativa. Además, se observó que a medida que se incrementa la cantidad de datos, el rendimiento disminuye considerablemente, lo que plantea preocupaciones sobre la capacidad del sistema para escalar y manejar el crecimiento de la organización a largo plazo.

La seguridad del sistema fue evaluada como adecuada, pero no cumple con los estándares más avanzados que se requieren para proteger datos sensibles, particularmente en el sector salud. Los participantes indicaron que es necesario mejorar la encriptación de datos y adoptar estándares como ISO/IEC 27001 para garantizar una protección más robusta de la información. Este aspecto es esencial para asegurar la confianza de los usuarios y el cumplimiento con regulaciones de privacidad.

En términos de compatibilidad, el software enfrenta limitaciones al no ser completamente compatible con tecnologías emergentes ni con otros sistemas de gestión de la organización. Se mencionaron problemas de incompatibilidad con dispositivos móviles y dificultades en la integración de datos, lo que limita la capacidad del sistema para interoperar con otras soluciones tecnológicas, lo cual es fundamental en un entorno empresarial moderno.

El mantenimiento y las actualizaciones del sistema también fueron discutidos. Los participantes sugirieron que el software debería recibir actualizaciones de manera trimestral para mantener su rendimiento y capacidad de integración con nuevas tecnologías. Sin embargo, también se destacó que las actualizaciones anteriores han causado inestabilidad temporal, lo que ha afectado la continuidad de las operaciones, lo que subraya la importancia de mejorar el proceso de actualización para evitar interrupciones.

El soporte técnico fue evaluado de manera negativa, ya que los usuarios experimentaron retrasos significativos en la resolución de problemas, con tiempos de respuesta lentos que han afectado la eficiencia general. Esto resalta la necesidad de mejorar la calidad del soporte técnico para garantizar una experiencia más satisfactoria y oportuna.

En cuanto a la productividad, aunque el software contribuye en algunas áreas, los problemas de integración y el bajo rendimiento han limitado su impacto positivo en la mejora de la productividad general. Se identificaron varios procesos repetitivos, como la gestión de mantenimiento preventivo y la generación de reportes, que podrían beneficiarse de la automatización para mejorar la eficiencia operativa y ahorrar tiempo.

Por último, en términos de escalabilidad, los participantes expresaron preocupaciones sobre la capacidad del sistema para manejar el crecimiento de la organización. El software no es lo suficientemente escalable para adaptarse a nuevas tecnologías o gestionar grandes volúmenes de datos, lo que podría convertirse en una barrera para el crecimiento futuro. Además, la falta de flexibilidad para adaptarse a nuevos requerimientos también fue señalada, ya que la arquitectura actual dificulta la personalización y la incorporación de nuevas funcionalidades.

4.2.2. Requerimientos

La definición de requerimientos y su priorización son pasos esenciales en cualquier proceso de modernización tecnológica, ya que establecen las bases para el éxito del proyecto al identificar y jerarquizar las necesidades del sistema.

Tabla 18

Lista de requerimientos funcionales

Lista de requerimientos funcionales	
ID	Requerimiento
RF-001	El sistema permite el registro de nuevos equipos médicos, incluyendo detalles como nombre, marca, modelo, número de serie, ubicación, fecha de adquisición y estado actual.
RF-002	El sistema envía recordatorios automáticos de mantenimiento preventivo según la programación establecida para cada equipo médico.
RF-003	El sistema permite registrar y gestionar la información completa de los proveedores, incluyendo sus capacidades, contratos, datos de contacto y calificaciones.
RF-004	El sistema almacena y organiza manuales, certificaciones y documentación de cumplimiento para cada equipo, asegurando que sean accesibles y descargables para usuarios autorizados.
RF-005	El sistema programa y gestiona las calibraciones de los equipos médicos, enviando alertas automáticas cuando se falten 3 días la fecha de calibración.
RF-006	El sistema proporciona control de acceso basado en roles, permitiendo la creación de perfiles de usuario con diferentes niveles de permisos para proteger la información sensible.
RF-007	El sistema genera reportes detallados y personalizables sobre el uso, mantenimiento, y estado de los equipos médicos, con opciones de exportación en formatos como PDF y Excel.
RF-008	El sistema permite la integración y sincronización de datos con otros sistemas hospitalarios como EDUS y el ERP del usuario, asegurando la interoperabilidad.
RF-009	El sistema debe proporcionar notificaciones en tiempo real sobre cambios de estado, eventos de mantenimiento, o requisitos de documentación pendientes relacionados con los equipos médicos.
RF-010	El sistema incluye un módulo de búsqueda avanzada que permite a los usuarios encontrar equipos médicos, proveedores, y documentos por múltiples criterios, como nombre, fecha de registro, o estado.
RF-011	El sistema permite la asignación de equipos a diferentes áreas o departamentos del hospital, con la posibilidad de transferirlos según sea necesario, manteniendo un historial de dichas transferencias.
RF-012	El sistema asegura la encriptación de toda la información almacenada, tanto en reposo como en tránsito, cumpliendo con los estándares de seguridad y privacidad aplicables.
RF-013	El sistema proporciona autenticación de dos factores para usuarios que manejan información sensible, garantizando un acceso seguro y confiable.

ID	Requerimiento
RF-014	El sistema realiza copias de seguridad automáticas de la base de datos de forma diaria, asegurando la recuperación en caso de fallos.
RF-015	El sistema permite la asignación y seguimiento de órdenes de compra para equipos médicos, incluyendo la opción de vincular dichas órdenes con proveedores y contratos previamente registrados.
RF-016	El sistema debe manejar hasta 1,000 usuarios simultáneos sin experimentar degradación en su rendimiento.
RF-017	El sistema muestra la disponibilidad de los equipos en tiempo real, indicando si están en uso, mantenimiento, calibración, o disponibles para uso inmediato.
RF-018	El sistema permite la generación de alertas y notificaciones automáticas a los usuarios responsables cuando se detecten equipos con fallos o requerimientos de mantenimiento urgente.
RF-019	El sistema ofrece funcionalidades para la gestión del ciclo de vida de los equipos, desde su adquisición hasta su desecho, registrando todos los eventos asociados.
RF-020	El sistema garantiza la accesibilidad a usuarios con diferentes niveles de habilidades técnicas, ofreciendo una interfaz de usuario intuitiva y soporte en línea.
RF-021	El sistema permite a los usuarios realizar solicitudes de reparación de equipos médicos, proporcionando detalles del problema, la fecha de la solicitud y el nivel de urgencia.
RF-022	El sistema mantiene un historial detallado de todas las intervenciones realizadas en cada equipo, incluyendo mantenimientos, reparaciones, calibraciones, y cambios de piezas.
RF-023	El sistema permite la generación automática de informes mensuales sobre el estado y rendimiento de los equipos médicos, destacando los equipos que requieren atención inmediata.
RF-024	El sistema facilita la gestión de contratos de mantenimiento con terceros, permitiendo el registro de fechas de inicio y vencimiento, tipo de cobertura, y costo asociado.
RF-025	El sistema proporciona un tablero de control (<i>dashboard</i>) que muestra indicadores clave de rendimiento (KPIs) de los equipos, como tiempo promedio entre fallos, costos de mantenimiento, y porcentaje de disponibilidad
RF-026	El sistema ofrece una funcionalidad para la evaluación y calificación de proveedores, basada en criterios como calidad de servicio, tiempo de respuesta, y cumplimiento de contratos.
RF-027	El sistema genera alertas automáticas cuando los contratos de mantenimiento o calibración están próximos a vencer, permitiendo a los usuarios renovar o buscar alternativas.
RF-028	El sistema permite la creación y asignación de tareas a técnicos de mantenimiento, con seguimiento del progreso y tiempos de ejecución de cada tarea.
RF-029	El sistema asegura que los equipos retirados de servicio sean etiquetados correctamente como 'En reparación', 'Para calibración', o 'Fuera de servicio', y se excluyen de la lista de equipos disponibles.

ID	Requerimiento
RF-030	El sistema permite la importación y exportación de datos en formatos como CSV y Excel, facilitando la actualización masiva de información y el análisis externo.
RF-031	El sistema verifica y registra automáticamente el cumplimiento de las normativas aplicables para cada equipo, proporcionando advertencias si algún equipo no cumple con los requisitos.
RF-032	El sistema debe ofrecer un módulo de capacitación y soporte para los usuarios, con tutoriales, manuales y videos explicativos sobre el uso del sistema.
RF-033	El sistema garantiza que los datos ingresados sean validados para evitar registros incompletos o incorrectos, asegurando la integridad de la información almacenada.
RF-034	El sistema permite la programación de auditorías internas de los equipos médicos, con la capacidad de generar informes detallados sobre hallazgos y recomendaciones.
RF-035	El sistema facilita la generación de códigos QR para cada equipo, permitiendo a los usuarios acceder rápidamente a su información mediante dispositivos móviles.
RF-036	El sistema debe enviar alertas automáticas a los responsables cuando se detecten anomalías en los registros de mantenimiento o calibración de los equipos.
RF-037	El sistema incluye la funcionalidad de seguimiento de piezas de repuesto, permitiendo la gestión de inventarios y el control de su uso en reparaciones o mantenimientos.
RF-038	El sistema permite la creación y configuración de flujos de trabajo personalizados para procesos de mantenimiento, reparación y calibración, adaptándose a las necesidades del hospital.
RF-039	El sistema debe ser compatible con dispositivos móviles y tabletas, permitiendo a los usuarios acceder y actualizar la información en tiempo real desde cualquier ubicación.
RF-040	El sistema genera reportes de eficiencia del personal técnico, incluyendo métricas como tiempo promedio de reparación, número de intervenciones completadas y nivel de satisfacción de los usuarios.
RF-041	El sistema proporciona un módulo para la planificación y seguimiento de inventarios, permitiendo la verificación de existencias y la generación de alertas cuando el stock de repuestos esté bajo.
RF-042	El sistema ofrece la funcionalidad de búsqueda avanzada para localizar documentos específicos relacionados con equipos, como manuales o certificaciones, utilizando filtros por categoría, fecha o palabras clave.
RF-043	El sistema facilita la generación de informes de costos de mantenimiento por equipo, departamento o proveedor, permitiendo la identificación de áreas de mayor gasto.
RF-044	El sistema permite la configuración de diferentes idiomas en la interfaz, asegurando su uso por personal de distintas regiones o con diferentes preferencias lingüísticas.
RF-045	El sistema verifica que los equipos médicos hayan recibido el mantenimiento y calibración requeridos antes de autorizarlos para su uso, asegurando la seguridad de los pacientes.
RF-046	El sistema registra y analiza las causas principales de fallos de los equipos, generando recomendaciones de prevención y mejora.

ID	Requerimiento
RF-047	El sistema permite la visualización y gestión de la trazabilidad completa de cada equipo médico, desde su adquisición hasta su desecho o reemplazo.
RF-048	El sistema debe incluir un módulo de gestión de garantías para los equipos médicos, permitiendo el seguimiento de fechas de expiración y condiciones de cobertura.
RF-049	El sistema asegura que las copias de seguridad se realicen de forma incremental para optimizar el espacio de almacenamiento y minimizar tiempos de recuperación.
RF-050	El sistema debe ser compatible con múltiples sistemas operativos y bases de datos para asegurar su implementación en diferentes entornos.
RF-051	El sistema debe ofrecer opciones de respaldo y recuperación de datos, garantizando la protección y disponibilidad de la información en caso de fallos.

Nota. Creación propia (2024)

Tabla 19

Lista de requerimientos no funcionales

Lista de requerimientos no funcionales	
ID	Requerimiento
RNF-001	El sistema debe tener una interfaz de usuario intuitiva y amigable, permitiendo que los usuarios realicen sus tareas principales en un máximo de tres clics, reduciendo la curva de aprendizaje.
RNF-002	El sistema debe ser accesible para usuarios con diferentes niveles de habilidades técnicas, proporcionando guías y tutoriales integrados para facilitar su uso.
RNF-003	El sistema debe proporcionar mensajes de error claros y comprensibles, acompañados de sugerencias para resolver los problemas detectados, asegurando una experiencia de usuario fluida.
RNF-004	El sistema debe garantizar una disponibilidad del 99.9%, permitiendo un acceso constante y minimizando el tiempo de inactividad.
RNF-005	El sistema debe incluir un mecanismo de recuperación automática en caso de fallos, asegurando que la operación se reanude en un máximo de 5 minutos después de un fallo.
RNF-006	El sistema debe realizar copias de seguridad incrementales diarias y copias completas semanalmente para asegurar la protección de los datos.
RNF-007	El sistema debe ser capaz de manejar al menos 1,000 usuarios simultáneos sin experimentar una degradación en el rendimiento.
RNF-008	El tiempo de respuesta para las operaciones comunes del sistema no debe superar los 2 segundos bajo condiciones normales de carga.
RNF-009	El sistema debe ser capaz de procesar y mostrar reportes que contengan hasta 10,000 registros en un máximo de 5 segundos.
RNF-010	El sistema debe implementar autenticación de dos factores para garantizar un nivel de seguridad adecuado en el acceso de usuarios.
RNF-011	Toda la información almacenada y transmitida por el sistema debe estar encriptada utilizando protocolos de seguridad como AES-256 y TLS 1.3 para proteger la confidencialidad de los datos.
RNF-012	El sistema debe registrar y monitorear todas las actividades de los usuarios, generando un log de auditoría que se almacene durante un periodo de al menos 12 meses.
RNF-013	El sistema debe incluir mecanismos de prevención contra ataques comunes, como inyecciones SQL, ataques Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF)
RNF-014	El sistema debe estar diseñado de manera modular, permitiendo que los componentes se actualicen o modifiquen sin afectar el funcionamiento de otras partes del sistema.
RNF-015	El sistema debe permitir la implementación de actualizaciones y parches de seguridad sin necesidad de interrumpir las operaciones en curso.

ID	Requerimiento
RNF-016	El código fuente del sistema debe cumplir con los estándares de codificación definidos por la industria (por ejemplo, PEP 8 para Python) y debe incluir comentarios que expliquen la lógica de las funcionalidades principales.
RNF-017	El sistema debe ser compatible con navegadores web modernos (Chrome, Firefox, Edge, Safari) y versiones móviles para asegurar un acceso flexible y multiplataforma.
RNF-018	La arquitectura del sistema debe ser escalable, permitiendo la adición de nuevas funcionalidades o el aumento de la capacidad de usuarios sin requerir rediseños significativos.
RNF-019	El sistema debe implementar un mecanismo de alertas que notifique automáticamente a los administradores cuando se detecten intentos de acceso no autorizados o posibles brechas de seguridad.
RNF-020	El sistema debe permitir la configuración de permisos y roles de acceso de manera granular, asegurando que cada usuario acceda únicamente a la información y funciones que le corresponden.
RNF-021	Los tiempos de carga de las páginas y módulos del sistema no deben exceder los 3 segundos incluso en condiciones de alta demanda, asegurando una experiencia de usuario ágil.
RNF-022	El sistema debe ser compatible con protocolos de integración estándar, como REST y SOAP, para facilitar la interoperabilidad con otros sistemas hospitalarios y de gestión.
RNF-023	Las actualizaciones y cambios en el sistema deben registrarse automáticamente en un historial de versiones, permitiendo un seguimiento y reversión en caso de errores o fallos.
RNF-024	El sistema debe ser capaz de realizar análisis de rendimiento y generar reportes sobre su uso y funcionamiento, facilitando la identificación de posibles áreas de mejora.
RNF-025	El sistema debe cumplir con normativas internacionales de seguridad y privacidad de datos, como la ISO/IEC 27001 y el Reglamento General de Protección de Datos (GDPR), asegurando su conformidad legal.
RNF-026	Los componentes del sistema deben ser reutilizables y seguir principios de diseño orientado a objetos para facilitar la mantenibilidad y futuras expansiones.
RNF-027	El sistema debe ofrecer soporte técnico y asistencia en línea a través de un centro de ayuda, con respuestas a preguntas frecuentes y chat en tiempo real para resolver problemas de los usuarios.
RNF-028	La instalación y configuración inicial del sistema deben completarse en menos de 2 horas, asegurando un proceso de implementación eficiente para el cliente.

Nota. Creación propia (2024)

4.2.3. Priorización de requerimientos

El apéndice A. Guía para taller de elicitación de requerimientos para la modernización del software Gestemed reveló la priorización de los requerimientos.

Tabla 20

Priorización de requerimientos MUST HAVE

Priorización de requerimientos <i>must have</i>	
ID	Requerimiento
RF-001	El sistema permite el registro de nuevos equipos médicos, incluyendo detalles como nombre, marca, modelo, número de serie, ubicación, fecha de adquisición y estado actual.
RF-002	El sistema envía recordatorios automáticos de mantenimiento preventivo según la programación establecida para cada equipo médico.
RF-003	El sistema permite registrar y gestionar la información completa de los proveedores, incluyendo sus capacidades, contratos, datos de contacto y calificaciones.
RF-004	El sistema almacena y organiza manuales, certificaciones y documentación de cumplimiento para cada equipo, asegurando que sean accesibles y descargables para usuarios autorizados.
RF-006	El sistema proporciona control de acceso basado en roles, permitiendo la creación de perfiles de usuario con diferentes niveles de permisos para proteger la información sensible.
RF-008	El sistema permite la integración y sincronización de datos con otros sistemas hospitalarios como EDUS y el ERP del usuario, asegurando la interoperabilidad.
RF-012	El sistema asegura la encriptación de toda la información almacenada, tanto en reposo como en tránsito, cumpliendo con los estándares de seguridad y privacidad aplicables.
RF-013	El sistema proporciona autenticación de dos factores para usuarios que manejan información sensible, garantizando un acceso seguro y confiable.
RF-014	El sistema realiza copias de seguridad automáticas de la base de datos de forma diaria, asegurando la recuperación en caso de fallos.
RF-016	El sistema debe manejar hasta 1,000 usuarios simultáneos sin experimentar degradación en su rendimiento.
RF-017	El sistema muestra la disponibilidad de los equipos en tiempo real, indicando si están en uso, mantenimiento, calibración, o disponibles para uso inmediato.
RF-019	El sistema ofrece funcionalidades para la gestión del ciclo de vida de los equipos, desde su adquisición hasta su desecho, registrando todos los eventos asociados.
RF-031	El sistema verifica y registra automáticamente el cumplimiento de las normativas aplicables para cada equipo, proporcionando advertencias si algún equipo no cumple con los requisitos.
RF-050	El sistema debe ser compatible con múltiples sistemas operativos y bases de datos para asegurar su implementación en diferentes entornos.

ID	Requerimiento
RF-051	El sistema debe ofrecer opciones de respaldo y recuperación de datos, garantizando la protección y disponibilidad de la información en caso de fallos.
RF-041	El sistema proporciona un módulo para la planificación y seguimiento de inventarios, permitiendo la verificación de existencias y la generación de alertas cuando el stock de repuestos esté bajo.
RF-047	El sistema permite la visualización y gestión de la trazabilidad completa de cada equipo médico, desde su adquisición hasta su desecho o reemplazo.
RF-029	El sistema asegura que los equipos retirados de servicio sean etiquetados correctamente como 'En reparación', 'Para calibración', o 'Fuera de servicio', y se excluyen de la lista de equipos disponibles.
RNF-004	El sistema debe garantizar una disponibilidad del 99.9%, permitiendo un acceso constante y minimizando el tiempo de inactividad.
RNF-005	El sistema debe incluir un mecanismo de recuperación automática en caso de fallos, asegurando que la operación se reanude en un máximo de 5 minutos después de un fallo.
RNF-006	El sistema debe realizar copias de seguridad incrementales diarias y copias completas semanalmente para asegurar la protección de los datos.
RNF-007	El sistema debe ser capaz de manejar al menos 1,000 usuarios simultáneos sin experimentar una degradación en el rendimiento.
RNF-008	El tiempo de respuesta para las operaciones comunes del sistema no debe superar los 2 segundos bajo condiciones normales de carga.
RNF-010	El sistema debe implementar autenticación de dos factores para garantizar un nivel de seguridad adecuado en el acceso de usuarios.
RNF-011	Toda la información almacenada y transmitida por el sistema debe estar encriptada utilizando protocolos de seguridad como AES-256 y TLS 1.3 para proteger la confidencialidad de los datos.
RNF-012	El sistema debe registrar y monitorear todas las actividades de los usuarios, generando un log de auditoría que se almacene durante un periodo de al menos 12 meses.
RNF-013	El sistema debe incluir mecanismos de prevención contra ataques comunes, como inyecciones SQL, ataques Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF).
RNF-014	El sistema debe estar diseñado de manera modular, permitiendo que los componentes se actualicen o modifiquen sin afectar el funcionamiento de otras partes del sistema.

ID	Requerimiento
RNF-015	El sistema debe permitir la implementación de actualizaciones y parches de seguridad sin necesidad de interrumpir las operaciones en curso.
RNF-025	El sistema debe cumplir con normativas internacionales de seguridad y privacidad de datos, como la ISO/IEC 27001 y el Reglamento General de Protección de Datos (GDPR), asegurando su conformidad legal.

Nota. Creación propia (2024)

Tabla 21

Priorización de requerimientos Should have

Priorización de requerimientos <i>should have</i>	
ID	Requerimiento
RF-005	El sistema programa y gestiona las calibraciones de los equipos médicos, enviando alertas automáticas cuando falten 3 días para la fecha de calibración.
RF-007	El sistema genera reportes detallados y personalizables sobre el uso, mantenimiento, y estado de los equipos médicos, con opciones de exportación en formatos como PDF y Excel.
RF-009	El sistema debe proporcionar notificaciones en tiempo real sobre cambios de estado, eventos de mantenimiento, o requisitos de documentación pendientes relacionados con los equipos médicos.
RF-011	El sistema permite la asignación de equipos a diferentes áreas o departamentos del hospital, con la posibilidad de transferirlos según sea necesario, manteniendo un historial de dichas transferencias.
RF-015	El sistema permite la asignación y seguimiento de órdenes de compra para equipos médicos, incluyendo la opción de vincular dichas órdenes con proveedores y contratos previamente registrados.
RF-018	El sistema permite la generación de alertas y notificaciones automáticas a los usuarios responsables cuando se detecten equipos con fallos o requerimientos de mantenimiento urgente.
RF-021	El sistema permite a los usuarios realizar solicitudes de reparación de equipos médicos, proporcionando detalles del problema, la fecha de la solicitud y el nivel de urgencia.
RF-023	El sistema permite la generación automática de informes mensuales sobre el estado y rendimiento de los equipos médicos, destacando los equipos que requieren atención inmediata.
RF-024	El sistema facilita la gestión de contratos de mantenimiento con terceros, permitiendo el registro de fechas de inicio y vencimiento, tipo de cobertura, y costo asociado.
RF-026	El sistema ofrece una funcionalidad para la evaluación y calificación de proveedores, basada en criterios como calidad de servicio, tiempo de respuesta, y cumplimiento de contratos.
RF-048	El sistema debe incluir un módulo de gestión de garantías para los equipos médicos, permitiendo el seguimiento de fechas de expiración y condiciones de cobertura.
RF-043	El sistema facilita la generación de informes de costos de mantenimiento por equipo, departamento o proveedor, permitiendo la identificación de áreas de mayor gasto.
RNF-001	El sistema debe tener una interfaz de usuario intuitiva y amigable, permitiendo que los usuarios realicen sus tareas principales en un máximo de tres clics, reduciendo la curva de aprendizaje.
RNF-002	El sistema debe ser accesible para usuarios con diferentes niveles de habilidades técnicas, proporcionando guías y tutoriales integrados para facilitar su uso.

ID	Requerimiento
RNF-003	El sistema debe proporcionar mensajes de error claros y comprensibles, acompañados de sugerencias para resolver los problemas detectados, asegurando una experiencia de usuario fluida.
RNF-009	El sistema debe ser capaz de procesar y mostrar reportes que contengan hasta 10,000 registros en un máximo de 5 segundos.
RNF-017	El sistema debe ser compatible con navegadores web modernos (Chrome, Firefox, Edge, Safari) y versiones móviles para asegurar un acceso flexible y multiplataforma.
RNF-018	La arquitectura del sistema debe ser escalable, permitiendo la adición de nuevas funcionalidades o el aumento de la capacidad de usuarios sin requerir rediseños significativos.
RNF-019	El sistema debe implementar un mecanismo de alertas que notifique automáticamente a los administradores cuando se detecten intentos de acceso no autorizados o posibles brechas de seguridad.
RNF-020	El sistema debe permitir la configuración de permisos y roles de acceso de manera granular, asegurando que cada usuario acceda únicamente a la información y funciones que le corresponden.
RNF-021	Los tiempos de carga de las páginas y módulos del sistema no deben exceder los 3 segundos incluso en condiciones de alta demanda, asegurando una experiencia de usuario ágil.
RNF-022	El sistema debe ser compatible con protocolos de integración estándar, como REST y SOAP, para facilitar la interoperabilidad con otros sistemas hospitalarios y de gestión.
RNF-023	Las actualizaciones y cambios en el sistema deben registrarse automáticamente en un historial de versiones, permitiendo un seguimiento y reversión en caso de errores o fallos.

Nota. Creación propia (2024)

Tabla 22

Priorización de requerimientos could have

Priorización de requerimientos <i>could have</i>	
ID	Requerimiento
RF-010	El sistema incluye un módulo de búsqueda avanzada que permite a los usuarios encontrar equipos médicos, proveedores, y documentos por múltiples criterios, como nombre, fecha de registro, o estado.
RF-020	El sistema garantiza la accesibilidad a usuarios con diferentes niveles de habilidades técnicas, ofreciendo una interfaz de usuario intuitiva y soporte en línea.
RF-025	El sistema proporciona un tablero de control (dashboard) que muestra indicadores clave de rendimiento (KPIs) de los equipos, como tiempo promedio entre fallos, costos de mantenimiento, y porcentaje de disponibilidad.
RF-027	El sistema genera alertas automáticas cuando los contratos de mantenimiento o calibración están próximos a vencer, permitiendo a los usuarios renovar o buscar alternativas.
RF-028	El sistema permite la creación y asignación de tareas a técnicos de mantenimiento, con seguimiento del progreso y tiempos de ejecución de cada tarea.
RF-030	El sistema permite la importación y exportación de datos en formatos como CSV y Excel, facilitando la actualización masiva de información y el análisis externo.
RF-032	El sistema debe ofrecer un módulo de capacitación y soporte para los usuarios, con tutoriales, manuales y videos explicativos sobre el uso del sistema.
RF-033	El sistema garantiza que los datos ingresados sean validados para evitar registros incompletos o incorrectos, asegurando la integridad de la información almacenada.
RF-034	El sistema permite la programación de auditorías internas de los equipos médicos, con la capacidad de generar informes detallados sobre hallazgos y recomendaciones.
RF-035	El sistema facilita la generación de códigos QR para cada equipo, permitiendo a los usuarios acceder rápidamente a su información mediante dispositivos móviles.
RF-044	El sistema permite la configuración de diferentes idiomas en la interfaz, asegurando su uso por personal de distintas regiones o con diferentes preferencias lingüísticas.
RNF-016	El código fuente del sistema debe cumplir con los estándares de codificación definidos por la industria (por ejemplo, PEP 8 para Python) y debe incluir comentarios que expliquen la lógica de las funcionalidades principales.
RNF-024	El sistema debe ser capaz de realizar análisis de rendimiento y generar reportes sobre su uso y funcionamiento, facilitando la identificación de posibles áreas de mejora.
RNF-026	Los componentes del sistema deben ser reutilizables y seguir principios de diseño orientado a objetos para facilitar la mantenibilidad y futuras expansiones.
RNF-027	El sistema debe ofrecer soporte técnico y asistencia en línea a través de un centro de ayuda, con respuestas a preguntas frecuentes y chat en tiempo real para resolver problemas de los usuarios.

ID	Requerimiento
RNF-028	La instalación y configuración inicial del sistema deben completarse en menos de 2 horas, asegurando un proceso de implementación eficiente para el cliente.

Nota. Creación propia (2024)

Tabla 23
Priorización de requerimientos won't have

Priorización de requerimientos won't have	
ID	Requerimiento
RF-036	El sistema debe enviar alertas automáticas a los responsables cuando se detecten anomalías en los registros de mantenimiento o calibración de los equipos.
RF-037	El sistema incluye la funcionalidad de seguimiento de piezas de repuesto, permitiendo la gestión de inventarios y el control de su uso en reparaciones o mantenimientos.
RF-038	El sistema permite la creación y configuración de flujos de trabajo personalizados para procesos de mantenimiento, reparación y calibración, adaptándose a las necesidades del hospital.
RF-039	El sistema debe ser compatible con dispositivos móviles y tabletas, permitiendo a los usuarios acceder y actualizar la información en tiempo real desde cualquier ubicación.
RF-040	El sistema genera reportes de eficiencia del personal técnico, incluyendo métricas como tiempo promedio de reparación, número de intervenciones completadas y nivel de satisfacción de los usuarios.
RF-042	El sistema ofrece la funcionalidad de búsqueda avanzada para localizar documentos específicos relacionados con equipos, como manuales o certificaciones, utilizando filtros por categoría, fecha o palabras clave.
RF-045	El sistema verifica que los equipos médicos hayan recibido el mantenimiento y calibración requeridos antes de autorizarlos para su uso, asegurando la seguridad de los pacientes.
RF-046	El sistema registra y analiza las causas principales de fallos de los equipos, generando recomendaciones de prevención y mejora.
RF-049	El sistema asegura que las copias de seguridad se realicen de forma incremental para optimizar el espacio de almacenamiento y minimizar tiempos de recuperación.

Nota. Creación propia (2024)

4.3. Fase 3: Análisis de alternativas tecnológicas

En esta fase del proyecto, se llevó a cabo una Apéndice G. Revisión documental con el fin de identificar y seleccionar las soluciones más adecuadas para la modernización de la arquitectura del software Gestemed. A lo largo de esta etapa, se evaluaron diversas arquitecturas de referencia, buenas prácticas de la industria, y las tendencias tecnológicas emergentes, asegurando que la solución propuesta sea escalable, eficiente y sostenible a largo plazo. Gracias a los distintos instrumentos aplicados anteriormente se obtiene el análisis de alternativas tecnológicas, estos instrumentos se encuentran en la sección de los apéndices: Apéndice G. Revisión documental.

4.3.1. Resultados de la revisión documental

El Apéndice G. Revisión documental revela aspectos clave que impactan directamente en el éxito del proyecto de modernización de la arquitectura de software Gestemed. En cuanto a la seguridad, los documentos de Shostack (2014) y OWASP (2021) destacan la necesidad de integrar la seguridad desde las primeras fases del diseño, un enfoque conocido como "Security by Design". Shostack propone el modelado de amenazas como una herramienta crucial para identificar y mitigar vulnerabilidades potenciales antes de que representen un riesgo para el sistema. OWASP refuerza estas prácticas al proporcionar un marco claro para evitar las vulnerabilidades más comunes en aplicaciones web, como la inyección de código y la fuga de datos sensibles. Implementar medidas como la sanitización de entradas y la autenticación robusta es esencial para proteger el sistema. La adopción de estas prácticas mejora significativamente la seguridad del software durante todo su ciclo de vida.

La resiliencia y escalabilidad del sistema se abordaron mediante los patrones arquitectónicos recomendados por Newman (2019), quien propone el uso de patrones como Circuit Breaker y Retry. Estos patrones permiten que el sistema continúe operando incluso ante fallos transitorios, evitando que los errores afecten al sistema completo. Esto es esencial para garantizar que el software Gestemed no solo sea escalable, sino también capaz de recuperarse automáticamente de problemas sin interrumpir las operaciones comerciales. La implementación de estos patrones resulta fundamental para asegurar que la modernización del sistema proporcione una infraestructura tecnológica sólida y adaptable a largo plazo.

En cuanto a la estandarización y gobernanza, los marcos de referencia ISO/IEC 42010:2011 y TOGAF desempeñan un papel crucial. ISO/IEC 42010:2011 establece directrices claras para la correcta documentación de la arquitectura, asegurando que sea comprensible y mantenible para todos los interesados. Esto facilita la gestión y permite una toma de decisiones más informada durante el desarrollo y mantenimiento del software. Por otro lado, TOGAF promueve la alineación de la arquitectura de software con los objetivos estratégicos de la organización, garantizando que el diseño arquitectónico apoye los fines empresariales de Key Tech. Ambos marcos proporcionan la estructura necesaria para desarrollar un sistema coherente y alineado con las mejores prácticas de la industria.

La documentación adecuada de la arquitectura también es esencial para asegurar la sostenibilidad del proyecto. El estándar ISO/IEC 42010:2011 ofrece un enfoque sistemático para crear vistas arquitectónicas que representen los intereses de todas las partes involucradas, lo que facilita la comprensión y gestión de la arquitectura del software. Este enfoque mejora la transparencia y asegura que las decisiones sobre el sistema se tomen de manera informada y precisa, ayudando a mantener la claridad a lo largo del ciclo de vida del proyecto.

Finalmente, la revisión de documentos como los de Shostack (2014), Newman (2019) y OWASP (2021) revela la importancia de seguir mejores prácticas en el diseño arquitectónico. Estas fuentes refuerzan la necesidad de un diseño que integre seguridad, resiliencia y escalabilidad, todo debidamente documentado y alineado con estándares internacionales. Siguiendo estas recomendaciones, el proyecto de modernización del software Gestemed asegurará una arquitectura robusta y adaptable a las necesidades de la organización.

4.3.2. Principios fundamentales de la arquitectura de software

De la Apéndice G. Revisión documental se detallan los principios fundamentales de la arquitectura de software.

Tabla 24

Principios fundamentales de la arquitectura de software

Principios fundamentales de la arquitectura de software	
Principio	Detalle
Modularidad	El sistema incluye un módulo de búsqueda avanzada que permite a los usuarios encontrar equipos médicos, proveedores, y documentos por múltiples criterios, como nombre, fecha de registro, o estado.
Escalabilidad horizontal y vertical	El sistema garantiza la accesibilidad a usuarios con diferentes niveles de habilidades técnicas, ofreciendo una interfaz de usuario intuitiva y soporte en línea.
Seguridad	<p>La seguridad debe ser una consideración primordial desde las primeras etapas de diseño de la arquitectura de software. Esto implica realizar <i>Threat Modeling</i> (modelado de amenazas) para identificar y abordar posibles vulnerabilidades antes de que se conviertan en problemas. Es fundamental aplicar prácticas como la validación y sanitización de entradas, el uso de autenticación y autorización robustas, la encriptación de datos tanto en tránsito como en reposo, y la adopción de técnicas de defensa en profundidad (Shostack, 2014). La implementación de controles de acceso basados en roles y la protección de datos sensibles son esenciales para prevenir brechas de seguridad y proteger la integridad del sistema.</p> <p>La arquitectura debe incluir herramientas y procesos de monitoreo que detecten posibles amenazas y vulnerabilidades en tiempo real. La implementación de auditorías regulares, pruebas de penetración y la adopción de un enfoque proactivo ante la seguridad garantizan la detección temprana de posibles fallos o ataques. Esto permite corregir rápidamente las debilidades y mantener la seguridad del sistema.</p> <p>OWASP (Open Web Application Security Project) proporciona un conjunto de buenas prácticas de seguridad y un listado de las 10 principales amenazas de seguridad para aplicaciones web, conocidas como "OWASP Top 10". Incorporar estas prácticas en el diseño y desarrollo de la arquitectura ayuda a mitigar vulnerabilidades como la inyección de código, la fuga de datos sensibles y la autenticación incorrecta. Esto se alinea con el principio de "Security by Design", asegurando que la seguridad esté integrada desde el inicio del desarrollo (OWASP, 2021).</p>
Rendimiento	El rendimiento es un aspecto crítico de la arquitectura que impacta la experiencia del usuario y la eficiencia operativa del sistema. Para optimizar el rendimiento, es fundamental identificar y minimizar los cuellos de botella y latencias en el diseño de la arquitectura. Esto puede incluir la implementación de técnicas de "caching" para reducir el acceso repetido a datos, el uso de balanceo de carga para distribuir la carga entre múltiples servidores y la adopción de bases de datos NoSQL para manejar grandes volúmenes de datos en tiempo real. (Avgeriou & Zdun, 2021)

Principio	Detalle
Disponibilidad y resiliencia	La arquitectura debe ser resiliente y capaz de resistir fallos imprevistos. Esto implica diseñar componentes que sean independientes y redundantes para evitar un punto único de fallo (SPOF). La implementación de patrones de diseño como "Circuit Breaker" y "Retry" permite que el sistema responda de manera adecuada a fallos transitorios, evitando que una falla en un componente afecte al sistema completo (Newman, 2019).
Portabilidad	El sistema proporciona un tablero de control (dashboard) que muestra indicadores clave de rendimiento (KPIs) de los equipos, como tiempo promedio entre fallos, costos de mantenimiento, y porcentaje de disponibilidad.
Flexibilidad y extensibilidad	El sistema genera alertas automáticas cuando los contratos de mantenimiento o calibración están próximos a vencer, permitiendo a los usuarios renovar o buscar alternativas.
Compatibilidad	El sistema permite la creación y asignación de tareas a técnicos de mantenimiento, con seguimiento del progreso y tiempos de ejecución de cada tarea.
Observabilidad	El sistema permite la importación y exportación de datos en formatos como CSV y Excel, facilitando la actualización masiva de información y el análisis externo.
Principio de KISS	El sistema debe ofrecer un módulo de capacitación y soporte para los usuarios, con tutoriales, manuales y videos explicativos sobre el uso del sistema.
Cumplimiento de Estándares (ISO/IEC 42010:2011)	Un principio fundamental es garantizar que la arquitectura de software cumpla con estándares internacionales reconocidos, como la ISO/IEC 42010:2011, que establece directrices sobre cómo describir y documentar una arquitectura de software. Esto incluye la definición de vistas arquitectónicas, preocupaciones de los interesados, y cómo los diferentes elementos de la arquitectura interactúan. Adoptar este estándar asegura que la arquitectura sea comprensible y gestionable para todas las partes interesadas (ISO, 2011).
Estandarización y Gobernanza (TOGAF)	TOGAF es un marco de referencia que proporciona una metodología para diseñar, planificar, implementar y gobernar una arquitectura de software empresarial. Uno de los principios fundamentales que fomenta TOGAF es la "alineación estratégica", asegurando que la arquitectura de software esté alineada con los objetivos y estrategias del negocio. Esto implica el uso de un enfoque sistemático para el desarrollo de la arquitectura, lo que garantiza que sea coherente, reutilizable y capaz de adaptarse a los cambios en el entorno empresarial (The Open Group, 2018).

Nota. Creación propia (2024)

4.3.2.1. Componentes críticos de la arquitectura de software

Del Apéndice G. Revisión documental se detallan los componentes críticos de la arquitectura de software.

Tabla 25

Componentes críticos de la arquitectura de software

Componentes críticos de la arquitectura de software	
Componente	Detalle
Capa de presentación (Front-End)	Debe ser lo suficientemente flexible para adaptarse a diferentes dispositivos y plataformas (responsive design). Además, debe incorporar prácticas de accesibilidad para asegurar que sea utilizable por personas con discapacidades (W3C, 2021). Las tecnologías modernas como React, Angular y Vue.js permiten construir interfaces de usuario más dinámicas y eficientes, y la arquitectura debe considerar la experiencia del usuario (UX) como un componente crítico.
Capa de negocio (Back-End)	Esta capa maneja la lógica de la aplicación y debe ser diseñada para ser escalable y segura. Los principios de "Domain-Driven Design" (DDD) son útiles para modelar la lógica del negocio de manera que refleje las reglas y procesos reales del dominio (Evans, 2003). La separación de las preocupaciones y la adopción de patrones como "Service Layer" o "Transaction Script" ayudan a mantener la lógica del negocio limpia y organizada.
Capa de datos	La capa de datos no solo incluye la base de datos principal, sino también la gestión de caché, colas de mensajes, y servicios de búsqueda. La arquitectura debe considerar el uso de bases de datos relacionales (SQL) y NoSQL, dependiendo de los requisitos del sistema (Date, 2003). También es importante considerar la seguridad de los datos (encriptación en reposo y en tránsito), la normalización/desnormalización de la base de datos y los mecanismos de respaldo y recuperación para garantizar la disponibilidad y consistencia de los datos.
APIs y servicios	Las APIs y servicios deben ser diseñados utilizando estándares como REST o GraphQL, lo que facilita la comunicación entre diferentes componentes y con sistemas externos (Fielding, 2000). Además, es fundamental aplicar principios de diseño como "API First" y "OpenAPI Specification" para asegurar que las APIs sean consistentes, documentadas y fáciles de usar por otros desarrolladores y servicios.
Middleware	Este componente facilita la comunicación entre diferentes capas y sistemas, actuando como un intermediario que garantiza que los datos se transmitan correctamente y en un formato adecuado. El middleware puede incluir sistemas de mensajería como RabbitMQ o Apache Kafka, que permiten la comunicación asíncrona entre servicios y componentes del sistema, lo cual es crítico para la construcción de sistemas distribuidos y escalables (Di Francesco et al., 2019).

Componente	Detalle
Gestión de configuración y control de versiones	Se debe contar con una infraestructura de gestión de la configuración que asegure que los cambios en la arquitectura sean controlados y documentados. Herramientas como Git y sistemas de CI/CD (Jenkins, GitLab CI/CD) son fundamentales para garantizar que las versiones del software sean rastreables y que el proceso de despliegue sea consistente y automatizado (Loeliger & McCullough, 2012).
Componentes de seguridad	La arquitectura debe incorporar mecanismos como autenticación, autorización, encriptación y auditoría. Herramientas como OAuth2, JWT y soluciones de Identity and Access Management (IAM) se utilizan para gestionar la autenticación y autorización de usuarios y servicios (Shostack, 2014).
Capa de seguridad (OWASP ASVS)	La seguridad debe ser un componente fundamental de cualquier arquitectura de software. OWASP proporciona el "Application Security Verification Standard" (ASVS), que es un marco que define los requisitos de seguridad para aplicaciones web, incluyendo autenticación, autorización, gestión de sesiones, y protección de datos. Incluir ASVS en la capa de seguridad asegura que el sistema cumple con los estándares de seguridad más rigurosos y protege contra amenazas comunes (OWASP, 2019).
Control y Documentación (ISO/IEC 42010:2011)	La documentación de la arquitectura debe ser detallada y seguir las directrices de ISO/IEC 42010:2011 para garantizar que todos los componentes, interacciones y dependencias estén claramente definidos. Esto incluye la creación de vistas arquitectónicas que describen los diferentes aspectos del sistema, como la vista lógica, la vista de implementación y la vista de despliegue, permitiendo que los interesados comprendan cómo está estructurado el sistema (ISO, 2011).
Gobernanza y desarrollo de la arquitectura (TOGAF ADM)	El " <i>Architecture Development Method</i> " (ADM) de TOGAF es un proceso iterativo que guía el desarrollo y la implementación de la arquitectura, asegurando que se aborden todas las fases del ciclo de vida, desde la planificación inicial hasta la implementación y mantenimiento. Este enfoque garantiza que cada componente de la arquitectura esté alineado con los objetivos del negocio y las mejores prácticas de la industria, y que se mantenga una gobernanza eficaz durante todo el proceso (The Open Group, 2018).
Evaluación de calidad y mantenibilidad (ISO/IEC 25010:2011)	El estándar ISO/IEC 25010 define las características de calidad que debe tener un producto de software, como la mantenibilidad, portabilidad, eficiencia y seguridad. Evaluar la arquitectura de software contra estos atributos asegura que los componentes críticos sean de alta calidad, facilitando el mantenimiento, la actualización y la escalabilidad del sistema a lo largo del tiempo (ISO, 2011).

Nota. Creación propia (2024)

5. Propuesta de solución

5.1. Fase 4: Diseño de la propuesta de modernización

Gracias a los distintos instrumentos aplicados anteriormente se obtiene la propuesta de modernización, estos instrumentos se encuentran en la sección de los apéndices: Apéndice C. Respuestas de entrevista, Apéndice E. Respuestas de la guía de satisfacción, Apéndice H. *Benchmarking*, Apéndice K. PESTEL, Apéndice I. Mapa de alineación de requerimientos, Apéndice F. Lista de verificación de buenas prácticas.

5.1.1. Estructura de la propuesta de modernización

La arquitectura propuesta para la modernización del software Gestemed se basa en principios de modularidad, escalabilidad, flexibilidad y seguridad. La transición hacia una arquitectura basada en microservicios y el uso de tecnologías modernas permitirá que el sistema sea más fácil de mantener, escalar y adaptar a nuevas exigencias. La estructura de la nueva arquitectura se organizará en capas bien definidas, cada una con roles específicos.

Principios de la nueva arquitectura

El rediseño de la arquitectura de Gestemed está impulsado por los siguientes principios:

- Modularidad: Descomponer el sistema en componentes y módulos independientes que puedan actualizarse o reemplazarse sin afectar a otros componentes.
- Escalabilidad Horizontal y Vertical: Asegurar que el sistema pueda aumentar su capacidad tanto añadiendo más servidores (horizontal) como mejorando los recursos existentes (vertical).
- Desacoplamiento de Componentes: Cada servicio funcionará de manera independiente, lo que reduce las interdependencias y facilita su mantenimiento.
- Seguridad Integral: Implementar seguridad desde el diseño hasta la ejecución, protegiendo los datos y asegurando el acceso controlado.
- Interoperabilidad: La arquitectura permitirá integrar fácilmente otros sistemas, tecnologías emergentes (como IoT) y soluciones de terceros mediante APIs.

Estructura de la arquitectura propuesta

La arquitectura estará organizada en capas bien definidas, donde cada capa es responsable de un conjunto específico de funcionalidades y servicios. Estas capas incluyen la capa de presentación, la capa de lógica de negocio, la capa de integración y la capa de persistencia. A continuación, se detalla cada una de ellas:

Capa de Presentación

La capa de presentación es la interfaz que interactúa directamente con los usuarios, permitiendo acceder a las funcionalidades de Gestemed desde diferentes dispositivos (escritorio,

tabletas y móviles). Esta capa incluye tanto la interfaz de usuario (UI) como la lógica de presentación que maneja las interacciones del usuario.

Tecnología sugerida:

- **React.js o Vue.js:** Para desarrollar aplicaciones web rápidas, escalables y de fácil mantenimiento.
- **React Native o Flutter:** Para aplicaciones móviles, permitiendo desarrollar aplicaciones multiplataforma (iOS y Android) con una base de código única.

Características Clave:

- **Interfaz de UI:** Rediseño completo de la UI para hacerla más intuitiva y eficiente. Se garantizará que la experiencia del usuario (UX) sea mejorada significativamente, con acceso fácil a todas las funciones críticas del sistema.
- **Responsividad:** La interfaz debe ser completamente responsive, adaptándose a diversos dispositivos y resoluciones de pantalla.
- **Capacidad de visualización de datos en tiempo real:** Monitoreo del estado de los equipos médicos y notificaciones instantáneas.
- **Accesibilidad:** Garantizar que el diseño de la UI sea inclusivo y accesible para usuarios con discapacidades (cumplimiento de WCAG).

Capa de servicios y lógica de negocio

Esta capa es el núcleo funcional de Gestemed, donde se implementa la lógica de negocio del sistema. Los microservicios serán responsables de manejar todas las funcionalidades clave como la gestión de equipos médicos, las alertas de mantenimiento y la gestión de usuarios.

Tecnología sugerida:

- **Spring Boot (Java) o Node.js (JavaScript)** para el desarrollo de microservicios.

Componentes principales:

- **Gestión de equipos médicos:** Controla todo el ciclo de vida de los equipos médicos, desde su ingreso, mantenimiento, hasta su eventual desecho. Cada equipo se gestiona mediante identificadores únicos (códigos QR o RFID).
- **Módulo de alertas y notificaciones:** Microservicio dedicado a generar y enviar alertas automáticas a los usuarios sobre mantenimientos programados o reemplazos necesarios. Este módulo puede comunicarse a través de correos electrónicos, mensajes SMS o aplicaciones móviles.
- **Autenticación y autorización:** Implementación de OAuth 2.0 para gestionar el acceso de los usuarios a los diferentes servicios del sistema de manera segura.
- **Gestión de mantenimiento preventivo y correctivo:** Servicios para programar y ejecutar mantenimientos preventivos, así como la generación de tickets para acciones correctivas cuando se detecten fallos en los equipos.

- **API Gateway:** Se implementará un *API Gateway* que gestionará las solicitudes que provienen de la capa de presentación, distribuyéndolas a los microservicios adecuados. Este *gateway* también se encargará de funciones como autenticación, balanceo de carga y limitación de tráfico.

Capa de Integración

La capa de integración conecta los distintos componentes del sistema, permitiendo la comunicación y la orquestación de los microservicios y su interacción con sistemas externos.

Tecnología sugerida:

- **RabbitMQ o Apache Kafka** para la mensajería asíncrona entre servicios.
- **GraphQL** para consultas eficientes de datos a través de las APIs.

Características clave:

- **Mensajería asíncrona:** Garantiza que los servicios se comuniquen de manera eficiente y no bloqueante, facilitando la escalabilidad y el desacoplamiento.
- **API RESTful y GraphQL:** Implementación de APIs RESTful y GraphQL para garantizar una comunicación eficiente entre la capa de presentación y la capa de servicios. Esto permitirá a los usuarios y otros sistemas acceder a los datos de manera rápida y precisa.
- **Interoperabilidad con Dispositivos IoT:** Posibilidad de conectar dispositivos médicos inteligentes (a través de estándares como FHIR o HL7) que envían datos en tiempo real al sistema para su monitoreo.

Capa de Persistencia y Bases de Datos

La capa de persistencia manejará el almacenamiento y recuperación de datos, optimizando el acceso a grandes volúmenes de información generados por la gestión de equipos médicos y las interacciones del sistema.

Tecnología sugerida:

- **MongoDB (NoSQL)** para almacenar grandes volúmenes de datos no estructurados, como registros de mantenimiento y datos históricos de los equipos.
- **PostgreSQL (SQL)** para manejar datos relacionales y transaccionales, como información de usuarios y permisos.

Características clave:

- **Escalabilidad horizontal:** Implementación de *sharding* en MongoDB para distribuir los datos de manera eficiente entre múltiples nodos.
- **Alta disponibilidad:** Uso de mecanismos de replicación en PostgreSQL y MongoDB para asegurar la disponibilidad de los datos en caso de fallos.

- **Seguridad de datos:** Cifrado de datos en reposo y en tránsito utilizando AES-256, asegurando la protección de la información sensible, como detalles de los equipos médicos y usuarios.
- **Índices y consultas optimizadas:** Configuración de índices eficientes para consultas rápidas y optimización de tiempos de respuesta.

Seguridad Integral

La seguridad será una prioridad en todas las capas de la arquitectura, garantizando que los datos y servicios sean seguros tanto en tránsito como en reposo.

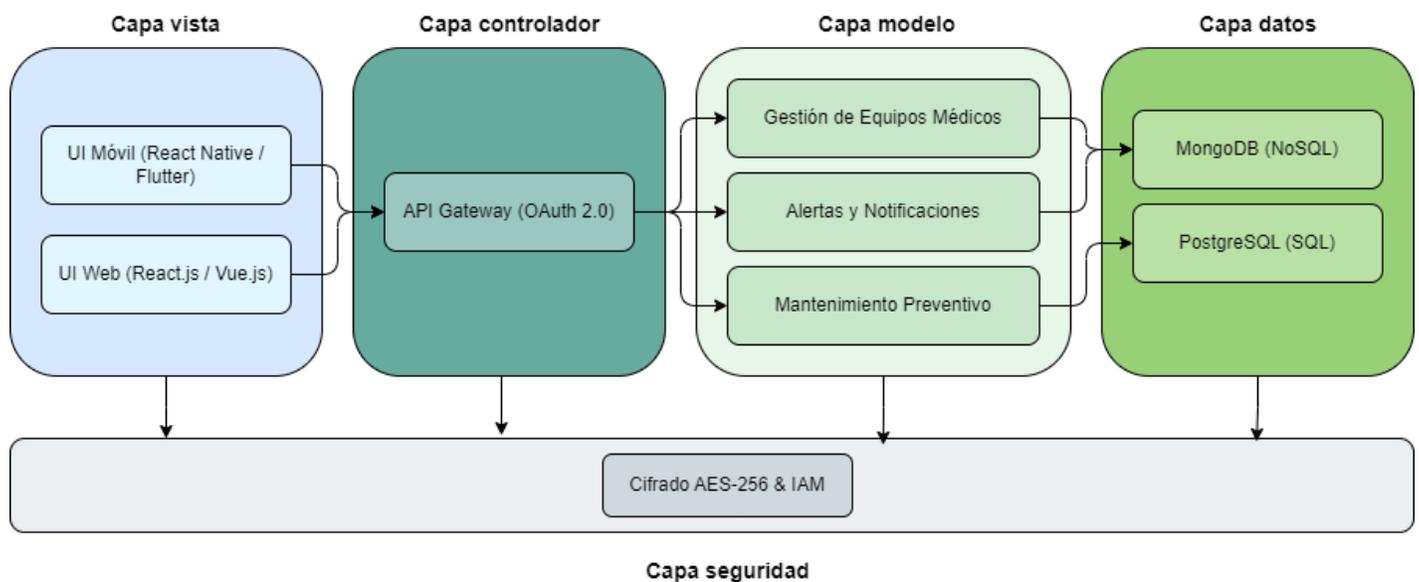
- **Cifrado de datos:** Implementación de cifrado mediante AES-256 para proteger los datos sensibles (como datos médicos y personales).
- **Autenticación y autorización segura:** Uso de OAuth 2.0 para la gestión de permisos, con soporte para autenticación multifactor (MFA) en funciones críticas.
- **Trazabilidad y auditoría:** Registro detallado de todas las actividades realizadas en el sistema, permitiendo auditorías de seguridad y trazabilidad completa de las operaciones.

Diagrama de Arquitectura Propuesto para Gestemed

Este diagrama muestra una arquitectura moderna y eficiente basada en microservicios, organizada en capas lógicas que facilitan la escalabilidad, el desacoplamiento y la seguridad de los componentes. Las diferentes capas se comunican entre sí de manera fluida, asegurando que las solicitudes del usuario sean manejadas de manera eficiente y segura.

Cada componente tiene responsabilidades bien definidas, lo que permite la actualización y mantenimiento de cada parte sin afectar al sistema completo, en línea con los principios de modularidad y escalabilidad que se proponen para la modernización del sistema Gestemed.

Figura 4
Diagrama de arquitectura



Nota. Creación propia (2024)

Capa de Vista

Esta capa representa los puntos de interacción del usuario final con el sistema, ya sea desde aplicaciones web o móviles.

- **UI Web (React.js/Vue.js):** Esta es la interfaz de usuario para dispositivos web, desarrollada con tecnologías modernas como React.js o Vue.js, lo que proporciona una experiencia rápida y escalable.
- **UI Móvil (React Native/Flutter):** Aquí se gestionan las aplicaciones móviles para iOS y Android, utilizando una base de código común con React Native o Flutter. Esto permite un desarrollo multiplataforma con un único esfuerzo.
- **Interacción:** Las interfaces web y móviles envían solicitudes al API Gateway en la capa de Controlador. Esto se representa con flechas que indican la dirección de las solicitudes desde las interfaces hacia el API Gateway.

Capa de Controlador (API Gateway)

Esta capa actúa como el intermediario que gestiona todas las solicitudes entrantes de los usuarios y las distribuye a los microservicios correspondientes.

- **API Gateway (OAuth 2.0):**
 - **Autenticación:** El API Gateway se encarga de autenticar a los usuarios mediante **OAuth 2.0** para asegurarse de que solo usuarios autorizados accedan a los servicios.
 - **Balanceo de Carga:** También distribuye las solicitudes de manera eficiente entre los diferentes microservicios, asegurando que las cargas de trabajo se manejen sin saturar los servicios.
 - **Distribución:** Recibe las peticiones desde la capa de vista y las redirige a los microservicios en la capa de modelo según el tipo de solicitud (ej., consultas de equipos médicos, alertas, mantenimiento).

Capa de Modelo (Microservicios)

Esta es la capa donde reside la lógica de negocio. Cada funcionalidad clave del sistema está desglosada en microservicios independientes que se comunican entre sí de forma desacoplada.

- **Gestión de Equipos Médicos:** Microservicio encargado de controlar el ciclo de vida de los equipos médicos, desde su ingreso hasta el mantenimiento y el descarte.
- **Alertas y Notificaciones:** Este microservicio es responsable de generar alertas automáticas, como las notificaciones de mantenimiento o reemplazo de equipos. Las notificaciones pueden ser enviadas por correo electrónico, SMS o mostradas en las interfaces de usuario.
- **Mantenimiento Preventivo/Correctivo:** Se encarga de la planificación y ejecución del mantenimiento preventivo, así como la generación de tickets para mantenimiento correctivo cuando se detectan fallos en los equipos.
- **Interacción:** El **API Gateway** distribuye las solicitudes a estos microservicios específicos, dependiendo de la acción solicitada. Los microservicios también pueden interactuar entre sí de manera asíncrona usando tecnologías como RabbitMQ o Apache Kafka (aunque no están explícitamente representados en el diagrama, este flujo de comunicación ocurre en segundo plano).

Capa datos (Persistencia de datos)

Aquí se manejan todos los datos que el sistema necesita almacenar y recuperar, dividiendo la persistencia entre bases de datos relacionales y NoSQL.

- **MongoDB (NoSQL):** Esta base de datos almacena grandes volúmenes de datos no estructurados, como registros de mantenimiento, datos históricos de equipos, y cualquier otra información que no requiera una estructura rígida.
- **PostgreSQL (SQL):** Maneja los datos transaccionales y relacionales, como la información de los usuarios, roles, permisos y auditoría.
- **Interacción:** Los microservicios de la capa de modelo interactúan con estas bases de datos cuando necesitan almacenar o recuperar información (ej., guardar el estado de un equipo o consultar registros de mantenimiento).

Capa de Seguridad

La capa de seguridad abarca todas las capas anteriores, asegurando que los datos y las operaciones se realicen de forma segura.

- **Cifrado AES-256:** Todos los datos, tanto en tránsito como en reposo, están protegidos mediante cifrado **AES-256**, lo que garantiza que la información sensible esté segura.
- **IAM (Identity and Access Management):** Se implementa la gestión de identidades y accesos, lo que asegura que solo los usuarios autorizados puedan acceder a ciertas funciones del sistema, especialmente cuando se trata de operaciones críticas, como la gestión de equipos médicos o datos sensibles.

Flujo de Datos y Conexiones

1. **Interacción Usuario:** Un usuario interactúa con la UI Web o UI Móvil. Realiza acciones como consultar equipos médicos o registrar un mantenimiento preventivo.
2. **Solicitud al API Gateway:** La interfaz de usuario envía la solicitud al API Gateway (autenticado mediante OAuth 2.0).
3. **Redirección a Microservicios:** El API Gateway distribuye la solicitud al microservicio adecuado (por ejemplo, el microservicio de gestión de mantenimiento).
4. **Consulta o Escritura en la Base de Datos:** Si el microservicio necesita interactuar con los datos, envía la consulta a MongoDB o PostgreSQL para almacenar o recuperar información relevante.
5. **Respuesta al Usuario:** La respuesta de la operación se envía desde el microservicio al API Gateway, que la redirige de vuelta a la Capa de Vista para ser presentada al usuario.

Plan de desarrollo y mantenimiento

Esta sección presenta un plan integral que detalla los pasos necesarios para llevar a cabo la actualización del sistema, desde el análisis inicial hasta la implementación y mantenimiento.

El enfoque propuesto se basa en metodologías ágiles y mejores prácticas de la industria, permitiendo una transición eficiente y segura hacia una arquitectura moderna, escalable y sostenible. Asimismo, se contemplan estrategias para mitigar riesgos, optimizar recursos y asegurar la continuidad operativa durante todo el proceso de modernización.

La estructura de este plan abarca ocho fases principales: diagnóstico, definición de requerimientos, diseño arquitectónico, validación, planificación de la implementación, desarrollo iterativo, capacitación y, finalmente, mantenimiento continuo. Cada etapa está diseñada para abordar de manera sistemática los retos identificados en el análisis FODAL y para maximizar el impacto positivo de la modernización tanto en la experiencia del usuario como en la competitividad de la empresa.

Fase 1: Diagnóstico y análisis de situación actual

Revisión documental y análisis contextual:

- Revisar la documentación existente de Gestemed.
- Realizar un análisis FODAL para identificar fortalezas, oportunidades, debilidades y amenazas.
- Evaluar el estado tecnológico y las necesidades de los usuarios.

Entregables:

- Documento del análisis FODAL.

Fase 2: Definición de requerimientos

Elicitación y priorización:

1. Realizar talleres y entrevistas con usuarios finales y stakeholders.
2. Documentar requerimientos funcionales y no funcionales según estándares como MoSCoW.

Entregables:

1. Lista priorizada de requerimientos funcionales y no funcionales.

Fase 3: Diseño de arquitectura

Evaluación de alternativas tecnológicas:

- Analizar arquitecturas modernas como microservicios, SOA o capas.
- Evaluar tecnologías actuales y estándares relevantes.

Propuesta de modernización:

- Diseñar una arquitectura alineada con las prácticas de modularidad, escalabilidad y seguridad.
- Crear diagramas técnicos detallados.

Entregables:

- Propuesta arquitectónica validada.
- Mapas de alineación con requerimientos.

Fase 4: Validación y refinamiento

Validación de Propuesta:

- Presentar la propuesta a los *stakeholders* y obtener retroalimentación.
- Refinar el diseño con base en comentarios y pruebas iniciales.

Entregables:

- Versión refinada de la arquitectura y requisitos alineados.

Fase 5: Planificación de implementación

Definir estrategias de migración:

- Elaborar un plan detallado para migrar de la arquitectura actual a la nueva.
- Planificar la mitigación de riesgos y estrategias de cambio organizacional.

Estimación de recursos:

- Determinar los recursos financieros, tecnológicos y humanos necesarios.
- Planificar la capacitación del personal.

Entregables:

- Cronograma detallado del proyecto.
- Plan de gestión de recursos y riesgos.

Fase 6: Implementación y pruebas

Desarrollo iterativo:

- Implementar la arquitectura en etapas utilizando metodologías ágiles (Scrum, Kanban).
- Realizar pruebas continuas para garantizar la calidad.

Entregables:

- Módulos funcionales progresivamente desplegados.
- Reportes de pruebas y ajustes.

Fase 7: Capacitación y lanzamiento

Capacitación a Usuarios:

- Desarrollar manuales y talleres de capacitación para los usuarios finales.
- Brindar soporte técnico post-lanzamiento.

Lanzamiento gradual:

- Implementar un despliegue controlado para minimizar riesgos.

Entregables:

- Sistema completamente funcional y documentado.
- Manuales y material de capacitación.

Fase 8: Mantenimiento y mejora continua

Monitoreo Continuo:

- Establecer KPIs para medir rendimiento y usabilidad.
- Implementar mecanismos para capturar retroalimentación de usuarios.

Mantenimiento Preventivo y Correctivo:

- Realizar actualizaciones de seguridad y mejoras funcionales.
- Documentar cambios y mantener la trazabilidad.

Entregables:

- Plan de mantenimiento con cronogramas y responsabilidades.
- Registro de incidencias y actualizaciones.

5.1.2. Compatibilidad con los requerimientos del FODAL

5.1.2.1. Resultados del mapa de alineación de requerimientos con el FODAL

El análisis de resultados basado en el mapa de alineación de requerimientos con el FODAL refleja una estrategia integral que aborda de manera efectiva las necesidades y desafíos del sistema Gestemed en Key Tech. Los requerimientos identificados están alineados con las fortalezas, oportunidades, debilidades, amenazas y limitaciones, lo que garantiza que el proyecto de modernización esté bien encaminado para lograr sus objetivos.

En cuanto a las fortalezas, el sistema ya cuenta con capacidades sólidas, como la identificación de equipos médicos mediante tecnologías QR y RFID, el control de acceso por roles y la generación de reportes detallados. Los requerimientos relacionados refuerzan estas áreas clave, asegurando la continuidad y mejorando la seguridad. Por ejemplo, los requerimientos RF-001 y RF-035 garantizan la compatibilidad continua con las tecnologías de identificación, mientras que el RNF-020 mejora la seguridad del sistema mediante la implementación de autenticación de dos factores. Estas mejoras aseguran que Gestemed mantenga sus puntos fuertes, lo que es crucial para su éxito operativo.

Respecto a las oportunidades, el análisis FODAL identifica varias áreas donde el sistema puede expandirse y mejorar. La integración de tecnologías emergentes como la inteligencia artificial y el análisis predictivo de mantenimiento (RNF-024) ofrece un gran potencial para mejorar la eficiencia operativa. Además, la compatibilidad con dispositivos móviles y la integración con sistemas hospitalarios (RF-008, RNF-017) son oportunidades clave que se abordan en los requerimientos. Estos cambios permitirán que el sistema se expanda hacia nuevos mercados y mejore la flexibilidad del personal al proporcionar acceso remoto. En resumen, el sistema está preparado para aprovechar estas oportunidades, lo que incrementará su competitividad y funcionalidad.

Por otro lado, las debilidades del sistema, como su arquitectura desactualizada y la baja escalabilidad, son aspectos críticos que están siendo abordados en la modernización. Los requerimientos que promueven la transición hacia una arquitectura de microservicios y la mejora de la escalabilidad (RNF-018, RNF-007) son fundamentales para resolver estos problemas. Estas mejoras permitirán que el sistema sea más flexible y capaz de manejar un mayor volumen de usuarios y datos. Además, la simplificación del código fuente y la mejora de la documentación técnica (RNF-016) también abordan la complejidad que ha dificultado el mantenimiento y la evolución del sistema. Con estos requerimientos, las debilidades del sistema se están mitigando de manera efectiva.

En cuanto a las amenazas, los riesgos de vulnerabilidades de seguridad y la posibilidad de quedarse obsoleto frente a competidores más avanzados son desafíos importantes que se han identificado. Los requerimientos que implementan medidas de seguridad avanzadas y adoptan nuevas tecnologías emergentes (RNF-011, RNF-013) garantizan que el sistema esté protegido contra amenazas externas y preparado para enfrentar la competencia. Por ejemplo, el RNF-011 introduce estándares de seguridad avanzados y actualizaciones regulares, mientras que el RNF-

013 asegura la adopción de tecnologías emergentes como la inteligencia artificial, lo que fortalecerá la posición competitiva de Gestemed.

Finalmente, las limitaciones del proyecto, como el plazo limitado de seis meses y la complejidad del código fuente, también están siendo gestionadas adecuadamente. Los requerimientos que priorizan los cambios críticos y simplifican el código (RNF-028, RNF-016) garantizan que el proyecto se realice dentro del plazo estipulado, sin sacrificar la calidad ni la funcionalidad. Además, la implementación por fases asegura que los aspectos más críticos se aborden primero, permitiendo una transición controlada hacia un sistema modernizado.

5.1.3. Uso de buenas prácticas de la industria

5.1.3.1. Resultados Lista de verificación de buenas prácticas

La nueva arquitectura propuesta para Gestemed, basada en microservicios, cumple ampliamente con los requisitos de modularidad, flexibilidad y desacoplamiento de componentes. La separación de responsabilidades entre las diferentes capas (presentación, lógica de negocio, integración y persistencia) está bien definida, lo que facilita el mantenimiento y la escalabilidad del sistema. Además, esta estructura modular asegura que los cambios o actualizaciones en una parte del sistema no afecten a otras, lo que resulta en un diseño robusto y flexible.

Otra fortaleza clave es la compatibilidad de la arquitectura con la infraestructura tecnológica actual, como el uso de contenedores y entornos virtualizados (e.g., Docker y Kubernetes), que permiten un despliegue ágil y eficiente del sistema. Sin embargo, aunque la arquitectura está claramente delineada, la documentación aún está incompleta. Esto representa un riesgo para la correcta implementación y adopción del sistema por parte del equipo de desarrollo, por lo que es crucial priorizar la creación de una documentación clara y detallada antes de continuar con las siguientes fases del proyecto.

En el ámbito de la seguridad, el sistema cumple con todos los requisitos necesarios para garantizar la protección de los datos y la integridad del sistema. Se han incluido medidas de seguridad avanzadas, como la autenticación multifactor (MFA) para usuarios y administradores, así como el uso de protocolos seguros como HTTPS y TLS para la transmisión de datos. Además, se ha planificado la encriptación de datos en reposo y en tránsito mediante AES-256, lo que asegura la confidencialidad de la información, tanto durante el almacenamiento como en su transmisión.

Otro punto fuerte es la inclusión de controles de acceso basados en roles (RBAC), lo que permite gestionar de manera segura los permisos de los usuarios. Asimismo, se realizarán pruebas de penetración y auditorías de seguridad periódicas, garantizando la identificación proactiva de vulnerabilidades. Sin embargo, aunque se planifica el monitoreo de actividades sospechosas utilizando herramientas como AWS CloudWatch, es importante asegurar que este monitoreo se configure adecuadamente para proporcionar alertas tempranas y evitar posibles brechas de seguridad.

La arquitectura está diseñada para manejar un aumento significativo en la carga de trabajo sin afectar el rendimiento, gracias a su capacidad de escalar horizontal y verticalmente. El uso de microservicios asegura que cada componente del sistema pueda escalar de manera independiente, lo que es esencial para gestionar eficientemente la creciente demanda de usuarios y equipos médicos. Además, el sistema está equipado con un balanceo de carga efectivo, lo que distribuye el tráfico entre los diferentes servicios y asegura la alta disponibilidad del sistema.

Un aspecto positivo adicional es la inclusión de pruebas de estrés y carga planificadas para validar la capacidad del sistema de manejar grandes volúmenes de usuarios y transacciones. Esto ayudará a identificar posibles cuellos de botella antes del despliegue en producción y permitirá

ajustar los recursos según sea necesario para asegurar que el sistema funcione de manera eficiente bajo condiciones de alta demanda.

El diseño de la arquitectura también contempla la optimización del rendimiento del sistema. Se planifica la implementación de técnicas de caching para reducir los tiempos de respuesta y mejorar la latencia, lo que es esencial para ofrecer una experiencia de usuario rápida y eficiente. Asimismo, se realizarán pruebas de rendimiento bajo diferentes cargas de trabajo, lo que permitirá ajustar el sistema y detectar cualquier problema de latencia o degradación antes de que impacte a los usuarios finales.

Además, se ha identificado la necesidad de optimizar las consultas a las bases de datos, implementando índices y otros mecanismos que aseguren una respuesta rápida incluso cuando se manejen grandes volúmenes de datos. La constante monitorización de los tiempos de respuesta y la latencia también será una parte clave del mantenimiento continuo del sistema.

La mantenibilidad del sistema aún presenta algunas áreas de mejora. Aunque se está trabajando en la documentación, aún no está completamente actualizada, lo que puede afectar la capacidad del equipo para realizar mantenimientos y actualizaciones de manera eficiente. Además, no se ha elaborado un manual de mantenimiento que detalle los procedimientos para realizar reparaciones y actualizaciones, lo cual es fundamental para asegurar la continuidad operativa.

En términos de control del código, el uso de sistemas de control de versiones como Git es una buena práctica que se está implementando, lo que facilitará la colaboración entre los desarrolladores y permitirá un seguimiento claro de los cambios en el código. Sin embargo, se deben definir mejor los procesos de gestión de incidentes para asegurar una rápida resolución de problemas, y se recomienda que se formalicen estos procesos antes de pasar a la fase de despliegue en producción.

El enfoque en la calidad del software es sólido, con la implementación de pruebas automatizadas para validar la funcionalidad y realizar pruebas de regresión tras cada cambio significativo en el sistema. Además, el sistema contará con pruebas unitarias, de integración y de extremo a extremo, lo que garantiza que todas las partes del sistema se prueben adecuadamente antes del despliegue. Este enfoque proactivo asegura que cualquier defecto o problema se identifique rápidamente, evitando posibles fallos en producción.

También se llevará a cabo un seguimiento continuo de las métricas de calidad, como la cobertura de pruebas y los defectos encontrados, lo que permitirá un ciclo de mejora continua en el desarrollo del software. Este enfoque es esencial para mantener la alta calidad del sistema a lo largo del tiempo y asegurar que cumpla con los requisitos de rendimiento, seguridad y escalabilidad.

5.2. Análisis de la viabilidad de la propuesta

El análisis de viabilidad del proyecto de modernización de la arquitectura del software Gestemed muestra resultados positivos y viables a largo plazo. Los costos estimados ascienden a ₡95,750,000 para un periodo de 6 meses. Estos costos incluyen tanto los gastos directos como los de desarrollo de software, consultoría, licencias y pruebas, así como los costos indirectos asociados a la administración del proyecto y la pérdida temporal de productividad. A pesar de que la inversión inicial es considerable, es necesario destacar que se trata de una modernización crítica para un sistema que ha quedado obsoleto y requiere adaptación a tecnologías más actuales.

Por otro lado, los beneficios anuales estimados alcanzan ₡55,000,000. Este monto se deriva de la reducción del 30% en tareas manuales, la disminución de errores y fallos, así como el incremento del 10% en ventas. Estos beneficios no solo mejorarán la eficiencia operativa y reducirán costos, sino que también abrirán nuevas oportunidades de negocio para la empresa, lo que refuerza la viabilidad del proyecto desde una perspectiva financiera.

El retorno sobre la inversión (ROI) calculado es del 57.43%. Este porcentaje indica que el proyecto generará un retorno considerable en relación con los costos iniciales, lo que valida la rentabilidad de la modernización. Aunque el ROI no llega al 100%, un 57.43% es un valor positivo, ya que refleja un retorno importante en el primer año de beneficios. En este sentido, se puede afirmar que la modernización del sistema tendrá un impacto acumulativo positivo a largo plazo.

Finalmente, el período de recuperación de la inversión es de 1.74 años. Este resultado es favorable, ya que significa que la empresa recuperará la inversión en menos de dos años. A partir del mes siete, los beneficios comenzarán a generar un retorno significativo, lo que garantiza la sostenibilidad del proyecto en el mediano plazo.

La modernización del software Gestemed es una inversión estratégica que permitirá a Key Tech mejorar su eficiencia operativa, reducir costos y aumentar su competitividad. Aunque el proyecto implica costos significativos de ₡95,750,000, los beneficios anuales de ₡55,000,000, junto con un retorno de la inversión del 57.43% y un periodo de recuperación de 1.74 años, hacen que esta modernización sea financieramente viable y crucial para el crecimiento sostenible de la empresa.

6. Conclusiones

El capítulo ofrece una síntesis de los hallazgos más relevantes obtenidos a lo largo del desarrollo del proyecto de modernización del software Gestemed. A través de un análisis detallado de los resultados alcanzados, se presenta una evaluación del cumplimiento de los objetivos propuestos y se destaca el impacto positivo que tendrá la implementación de la nueva arquitectura en Key Tech. En este capítulo se muestran las conclusiones detalladas por objetivos.

Objetivo General: Proponer la modernización de la arquitectura del software Gestemed en Key Tech, para el favorecimiento de su compatibilidad con tecnologías modernas y satisfacción de las necesidades actuales de los usuarios.

- El proyecto cumplió con el objetivo de proponer una modernización integral de la arquitectura del software Gestemed, alineando el diseño con las necesidades actuales y futuras de los usuarios.
- La propuesta presentada abordó problemas de compatibilidad tecnológica y define una estructura que facilita la evolución del sistema frente a demandas tecnológicas actuales.
- Se integraron componentes y estándares que permiten la interoperabilidad y sostenibilidad del software en el entorno actual.

Objetivo 1: Analizar la situación actual de la arquitectura de software Gestemed para el diagnóstico técnico mediante Fortalezas, Oportunidades, Debilidades, Amenazas y Limitaciones (FODAL).

- Se identificaron las principales debilidades y áreas críticas del software Gestemed mediante el análisis FODAL, destacando problemas en la escalabilidad y la compatibilidad tecnológica. Este diagnóstico ha servido de base para establecer las acciones de modernización necesarias, como se describe en la sección 4.2.1.
- El análisis FODAL identificó las principales debilidades del software Gestemed, como problemas relacionados con la escalabilidad y la incompatibilidad con tecnologías actuales.
- Las fortalezas del sistema actual se utilizaron como base para estructurar mejoras que optimicen su desempeño operativo.
- Las amenazas y limitaciones detectadas en el análisis sirvieron para definir áreas críticas a priorizar en la propuesta de modernización.

Objetivo 2: Establecer el estado meta de la modernización de la arquitectura para atender los requerimientos funcionales y no funcionales identificados mediante el FODAL.

- Se lograron definir los requerimientos funcionales y no funcionales del sistema a partir de las necesidades de los usuarios y la organización. Esto ha permitido diseñar una arquitectura que satisface tanto las demandas actuales como futuras, tal como se indica en la sección 4.2.2.

- Se definieron los requerimientos funcionales del sistema, considerando las necesidades específicas de los usuarios y la organización.
- Los requerimientos no funcionales establecieron lineamientos claros relacionados con rendimiento, seguridad y escalabilidad.
- La definición de un estado meta permitió estructurar una arquitectura que responde a las necesidades inmediatas del software y se adapta a posibles escenarios futuros.
- La planificación de los requerimientos identificados facilita la alineación del sistema con los estándares operativos y tecnológicos requeridos.

Objetivo 3: Valorar las mejores prácticas de la industria sobre arquitectura de software para identificar los componentes a considerar en la modernización.

- Se implementaron siete de las diez buenas prácticas identificadas en el análisis de la industria, mejorando la eficiencia del sistema y alineándolo con las tendencias tecnológicas más recientes, como se detalla en la sección 5.1.3.
- Se analizaron diez prácticas de la industria, de las cuales siete fueron incorporadas en el diseño de la propuesta, atendiendo a criterios de eficiencia y sostenibilidad.
- La integración de patrones arquitectónicos como microservicios permitió segmentar el sistema en componentes modulares e independientes.
- Las prácticas seleccionadas aseguran una arquitectura que cumple con estándares reconocidos en el sector tecnológico.
- Se establecieron criterios para seleccionar tecnologías que optimicen la implementación y el mantenimiento del sistema.

Objetivo 4: Construir una propuesta de modernización de la arquitectura del software para satisfacer todas las mejoras identificadas en el FODAL y los requerimientos elicitados.

- Se elaboró una propuesta técnica de modernización que incluye la adopción de tecnologías, la cual garantiza la sostenibilidad a largo plazo del sistema y su compatibilidad actualizaciones tecnológicas, como se presenta en la sección 5.2.
- Se elaboró una propuesta técnica que detalla las tecnologías seleccionadas y los cambios requeridos en la arquitectura del sistema.
- La propuesta incluye estrategias para mitigar riesgos asociados a la implementación y garantizar la continuidad operativa durante el proceso de modernización.
- La arquitectura propuesta está diseñada para integrar nuevas tecnologías y soportar actualizaciones de forma progresiva.
- El plan desarrollado considera la interoperabilidad del software con otros sistemas y su capacidad de adaptación a escenarios operativos diversos.

7. Recomendaciones

Con base al análisis de resultados y la propuesta de modernización presentada, se han identificado una serie de recomendaciones orientadas a garantizar el éxito del proyecto, minimizar riesgos durante la implementación y asegurar la sostenibilidad del sistema a largo plazo. Estas recomendaciones están dirigidas tanto a la fase de ejecución del proyecto como a los procesos organizacionales que rodean el uso del software.

Se recomienda implementar esta actualización en un período máximo de 6 meses. Esto permitirá evitar incrementos en los costos asociados y mantener el proyecto dentro de los límites presupuestarios definidos. Un cronograma ajustado y claramente estructurado será esencial para cumplir este objetivo.

Se sugiere lo siguiente complementar esta información con gráficos estadísticos para cumplir completamente con los criterios de visualización efectiva de datos. Incorporar gráficos que resuman los resultados de manera visual e intuitiva, como diagramas de barras, de pastel o infografías interactivas. Esto reforzará los argumentos presentados y facilitará la comprensión por parte de los stakeholders.

Es imprescindible desarrollar una documentación técnica exhaustiva que incluya especificaciones del diseño, arquitectura y procesos del sistema actualizado. Esto permitirá transferir conocimientos clave a nuevos equipos y facilitará los mantenimientos futuros, minimizando riesgos operativos y reduciendo el tiempo de resolución de problemas.

Se recomienda considerar arquitecturas de tipo *Platform as a Service* (PaaS) y soluciones en la nube como parte de la propuesta. Estas tecnologías pueden ofrecer escalabilidad, flexibilidad y reducción de costos operativos a largo plazo. Además, estas opciones están alineadas con las tendencias modernas en tecnología y permitirán integrar nuevas capacidades de manera más eficiente.

Se recomienda generar diagramas de la arquitectura propuesta, es fundamental elaborar diagramas comparativos entre las diferentes opciones evaluadas durante el proyecto, mostrando ventajas y desventajas. Diagramas detallados de la arquitectura final seleccionada, incluyendo diagramas de despliegue, de componentes y de flujo de datos. Estos diagramas facilitarán la comprensión de las decisiones tomadas y servirán como referencia visual para los equipos técnicos.

Es fundamental que la implementación de la modernización del software se lleve a cabo de manera gradual y controlada. Se recomienda adoptar una estrategia de implementación faseada que permita migrar componentes críticos del sistema en etapas, comenzando por aquellos que tienen un mayor impacto en la operación diaria, como la gestión de equipos médicos y el control de datos. Esta estrategia reducirá el riesgo de interrupciones operativas y permitirá un ajuste progresivo a las nuevas tecnologías. Además, se aconseja realizar pilotos en áreas específicas de la empresa antes de la implementación completa, para identificar y corregir posibles errores.

La modernización del software Gestemed incluye la mejora de la seguridad del sistema, pero se recomienda ir más allá y establecer un enfoque proactivo de ciberseguridad. Esto implica la implementación de auditorías regulares de seguridad, actualizaciones automáticas de parches de seguridad y la adopción de mecanismos avanzados de cifrado de datos sensibles. Dado que la obsolescencia tecnológica del software anterior representaba un riesgo significativo en términos de vulnerabilidades de seguridad, es imperativo asegurar que la nueva versión cumpla con los estándares más altos de protección de la información. También se recomienda la integración de sistemas de monitoreo de amenazas en tiempo real para detectar y mitigar riesgos cibernéticos de manera temprana.

La capacitación del personal es clave para asegurar una adopción exitosa del nuevo sistema. Se sugiere desarrollar un plan integral de gestión del cambio, que incluya no solo sesiones de formación técnica para el uso del software, sino también talleres enfocados en la adaptación organizacional y el manejo del cambio tecnológico. La participación de los usuarios finales y el equipo de soporte técnico es esencial desde las fases iniciales del proyecto, asegurando que sus preocupaciones y necesidades sean atendidas. Adicionalmente, se recomienda crear un equipo de soporte interno que brinde asistencia personalizada durante los primeros meses de uso del sistema modernizado, lo que reducirá la curva de aprendizaje y aumentará la confianza del personal en la nueva herramienta.

La modernización de la arquitectura de software Gestemed ofrece una oportunidad para revisar y optimizar los procesos operativos de la organización. Se recomienda que Key Tech aproveche esta modernización no solo para actualizar el software, sino también para rediseñar flujos de trabajo y mejorar la eficiencia operativa. Automatizar tareas repetitivas, como la gestión de inventarios o el seguimiento de equipos médicos, reducirá la carga manual sobre el personal y permitirá que los recursos se utilicen de manera más eficiente. Además, es conveniente alinear los nuevos procesos operativos con las funcionalidades avanzadas del software, lo que asegurará un mejor aprovechamiento de las capacidades técnicas del sistema.

Una vez implementado el nuevo software, es crucial establecer un plan de monitoreo continuo para garantizar su correcto funcionamiento y anticiparse a posibles problemas. Se sugiere que Key Tech implemente herramientas de monitoreo del rendimiento del sistema que permitan detectar cuellos de botella, caídas de servicio o errores en tiempo real. Asimismo, se debe priorizar un enfoque de mantenimiento preventivo, realizando revisiones periódicas para detectar áreas que puedan requerir ajustes o mejoras antes de que afecten el rendimiento general del sistema. La creación de un cronograma de mantenimiento preventivo asegurará la longevidad del software y reducirá costos asociados a reparaciones no planificadas.

Es importante que Key Tech implemente un mecanismo de retroalimentación continua para evaluar la satisfacción de los usuarios con el sistema modernizado. Esto permitirá identificar áreas de mejora en tiempo real y garantizar que el software se mantenga alineado con las necesidades cambiantes de los usuarios y de la organización. Se recomienda realizar encuestas de satisfacción periódicas, así como la creación de un canal de comunicación directa con el equipo de desarrollo, lo que facilitará la solución rápida de problemas y la implementación de mejoras continuas.

La implementación de una arquitectura modernizada implica una serie de riesgos asociados, como la interrupción temporal del servicio o dificultades en la integración con sistemas preexistentes. Por lo tanto, se recomienda establecer un plan de gestión de riesgos que incluya la identificación de riesgos potenciales, su análisis y la elaboración de estrategias de mitigación. Contar con un equipo que supervise constantemente los avances de la implementación y gestione los riesgos de manera proactiva es crucial para evitar contratiempos que afecten el desempeño general del proyecto.

Con la introducción de nuevas tecnologías en la modernización del software, será necesario que el equipo de soporte técnico también se modernice y amplíe sus capacidades. Es recomendable que Key Tech invierta en la capacitación continua del equipo de soporte, asegurando que tengan los conocimientos necesarios para gestionar eficientemente las nuevas tecnologías implementadas. Además, se sugiere la implementación de un sistema de tickets y seguimiento para la resolución de problemas técnicos, garantizando tiempos de respuesta rápidos y efectivos.

Se sugiere incluir el uso de herramientas de IA para optimizar y agilizar los procesos operativos. La implementación de IA permitirá analizar grandes volúmenes de datos en tiempo real, lo que facilitará la toma de decisiones informadas. Además, la IA puede mejorar la automatización en áreas como la detección de anomalías en el sistema, el mantenimiento predictivo y el análisis de satisfacción del usuario, incrementando la eficiencia y reduciendo errores humanos.

Finalmente, la modernización del software Gestemed no solo debe enfocarse en mejorar la eficiencia operativa, sino también en potenciar el modelo de negocio de Key Tech. Se recomienda que, tras la implementación de la modernización, la empresa revise sus procesos de negocio y explore nuevas oportunidades de mercado. La modernización del software permitirá a Key Tech ofrecer un servicio más competitivo, por lo que es crucial alinear las capacidades del sistema con una estrategia de expansión que maximice los beneficios comerciales.

8. Referencias

- Asset Panda. (2022). Software de gestión de activos. <https://www.assetpanda.com>
- Avgeriou, P., & Zdun, U. (2021). Software architecture: Foundations, theory, and practice. Springer.
- Barroso, L. A., Clidaras, J., & Hölzle, U. (2018). The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Morgan & Claypool.
- Bass, L., Clements, P., & Kazman, R. (2003). Software architecture in practice (2nd ed.). Addison-Wesley.
- Behrendt, M., Govindarajan, S., & Graf, G. (2016). Technology in business. Wiley.
- Boehm, B. W., & Turner, R. (2003). Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley.
- Brooks, F. P. (1995). The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley.
- Brown, A. W., & Booch, G. (2020). Software Architecture: A Comprehensive Framework and Guide for Practitioners. Addison-Wesley Professional.
- Burns, B., & Oppenheimer, D. (2017). Kubernetes: Up and running. O'Reilly Media.
- Cauas, D. (2015). Definición de las variables, enfoque y tipo de investigación. Bogotá: biblioteca electrónica de la universidad Nacional de Colombia, 2, 1-11
- Chaffey, D., & White, G. (2011). Business Information Management. Pearson Education.
- Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., & Stafford, J. (2010). Documenting software architectures: Views and beyond (2nd ed.). Addison-Wesley.
- Di Francesco, P., Lago, P., & Malavolta, I. (2019). Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption. Journal of Systems and Software, 150, 250-268.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of Business Process Management (2nd ed.). Springer.
- eMaint CMMS. (2023). Computerized Maintenance Management System. <https://www.emaint.com>
- Erl, T. (2005). Service-oriented architecture: Concepts, technology, and design. Prentice Hall.
- Evans, E. (2003). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley Professional.

- EZOfficeInventory. (2022). Asset tracking software. Recuperado de <https://www.ezofficeinventory.com>
- Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).
- Fowler, M. (2003). Patterns of Enterprise Application Architecture. Addison-Wesley.
- Fowler, M. (2019). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley.
- Francisco Coll Morales. (2021, February 17). Fuente primaria. Economipedia.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design patterns: Elements of reusable object-oriented software. Addison-Wesley.
- Gartner. (2023). Trends in Modernizing Enterprise Software: A Gartner Report. Gartner Research.
- Heineman, G. T., & Councill, W. T. (2001). Component-based software engineering: Putting the pieces together. Addison-Wesley.
- Hernández, R., Baptista, P., & Fernández, C. (2014). Metodología de la investigación (6a ed.). McGraw-Hill.
- Hernández-Sampieri, R. (2024). *Metodología de la investigación* (pp. 522-561).
- Hiatt, J. (2006). ADKAR: A model for change in business, government, and our community. Prosci.
- Hippo CMMS. (2022). Software de gestión de mantenimiento preventivo. Recuperado de <https://www.hippocmms.com>
- Hohpe, G., & Woolf, B. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley.
- <https://economipedia.com/definiciones/fuente-primaria.html>
- Infor EAM. (2023). Enterprise Asset Management Software. Recuperado de <https://www.infor.com>
- ISO. (2011). ISO/IEC 25010:2011 Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. <https://www.iso.org/standard/35733.html>
- ISO. (2011). ISO/IEC 42010:2011 Systems and software engineering — Architecture description. <https://www.iso.org/standard/50508.html>
- ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. International Organization for Standardization.

- ISO/IEC. (2008). ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes.
- ISO/IEC. (2011). ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
- ISO/IEC. (2011). ISO/IEC 29110-4-1:2011 Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 4-1: Software engineering basic profile group.
- ISO/IEC. (2013). ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements.
- KeyTech. (2024). Propuesta de modernización de arquitectura del software Gestemed en Key Tech. Informe Interno.
- Kitchenham, B., & Charters, S. (2019). Guidelines for performing Systematic Literature Reviews in Software Engineering. Keele University and Durham University Joint Report.
- Kotter, J. P. (2012). Leading change. Harvard Business Review Press.
- Leffingwell, D. (2018). SAFe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises. Addison-Wesley Professional.
- Loeliger, J., & McCullough, M. (2012). Version control with Git: powerful tools and techniques for collaborative software development. O'Reilly Media.
- Maintenance Connection. (2022). Mantenimiento basado en la nube. <https://www.maintenanceconnection.com>
- Newman, S. (2015). Building microservices: Designing fine-grained systems. O'Reilly Media.
- Newman, S. (2019). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.
- Nielsen, J., & Budiu, R. (2013). Mobile usability. New Riders.
- Oracle Business Process Management. (2018). Oracle.com. <https://www.oracle.com/middleware/Technologies/bpm.html>
- Orfali, R., Harkey, D., & Edwards, J. (1996). The essential distributed objects survival guide. John Wiley & Sons.
- OWASP. (2019). Application Security Verification Standard (ASVS) Version 4.0.3.
- OWASP. (2019). OWASP Application Security Verification Standard (ASVS) 4.0.1. <https://owasp.org/www-project-application-security-verification-standard/>
- OWASP. (2020). Software Assurance Maturity Model (SAMM) Version 2.0.

- OWASP. (2021). OWASP Top Ten 2021.
- OWASP. (2021). OWASP Top Ten. <https://owasp.org/www-project-top-ten/>
- Papazoglou, M. P. (2012). Service-Oriented Computing: Concepts, Characteristics, and Directions. Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE'03).
- Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. Communications of the ACM, 15(12), 1053-1058.
- Parnas, D. L. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the ACM, 15(12), 1053-1058.
- Pautasso, C., Zimmermann, O., & Leymann, F. (2008). Restful web services vs. "big" web services: Making the right architectural decision. In Proceedings of the 17th international conference on World Wide Web (pp. 805-814). ACM.
- Prosci. (2021). Change Management Methodology. Prosci.
- Quienes somos. (2024). ¿Quienes somos? KeyTech-Solutions.com. <https://www.keyTech-solutions.com/quienessomos>
- Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media.
- Richardson, C., & Smith, F. (2021). *Microservices Patterns: With examples in Java*. Manning Publications.
- Roberts, M. (2018). *Serverless architectures on AWS: With examples using AWS Lambda*. Manning Publications.
- Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.
- Szyperski, C. (2002). *Component software: Beyond object-oriented programming* (2nd ed.). Addison-W
- The Open Group. (2018). TOGAF® Version 9.2 - A Standard of The Open Group. The Open Group.
- The Open Group. (2018). TOGAF® Version 9.2. Van Haren Publishing.
- UpKeep. (2023). Maintenance management software. Recuperado de <https://www.onupkeep.com>
- W3C (World Wide Web Consortium). (2021). Web Content Accessibility Guidelines (WCAG) 2.1. Recuperado de <https://www.w3.org/TR/WCAG21/>
- Wiegers, K. E., & Beatty, J. (2013). *Software requirements* (3ra ed.). Microsoft Press.

9. Apéndices

A. Guía para taller de elicitación de requerimientos

Guía para taller de elicitación de requerimientos

Objetivo: Identificar los requerimientos funcionales y no funcionales necesarios para la modernización del software Gestemed.

Secciones:

1. Introducción:

- Explicación del propósito del taller y la importancia de la participación activa.

2. Dinámicas colaborativas:

- **Priorización de funcionalidades:** Usar la técnica MoSCoW para priorizar los requerimientos.

3. Preguntas para discusión:

Funcionalidad:

- **¿Qué funcionalidades actuales son imprescindibles para su trabajo diario?**

Las funcionalidades clave que utilizamos a diario son el registro de equipos médicos, el rastreo de los equipos y la consulta de atributos técnicos. Estas son esenciales para el seguimiento y gestión de los activos médicos que utilizamos.

- **¿Cuáles son las funcionalidades que casi nunca utiliza?**

Funcionalidades como la gestión avanzada de inventarios o las opciones detalladas de reportes no son utilizadas frecuentemente debido a que no están alineadas con nuestras necesidades actuales.

- **¿Qué funcionalidades adicionales le gustaría tener?**

Me gustaría contar con una funcionalidad de rastreo por GPS más eficiente y la posibilidad de integrar tecnologías como códigos QR para el registro rápido de equipos. Además, una herramienta de análisis predictivo basada en inteligencia artificial sería valiosa para optimizar el mantenimiento preventivo.

- **¿El sistema permite realizar todas las tareas operativas necesarias? Si no, ¿qué le falta?**

Actualmente, el sistema no cubre todas las tareas operativas necesarias. Le falta mayor integración con otros sistemas de gestión y la capacidad de manejar datos en tiempo real para optimizar la toma de decisiones.

Interfaz de usuario:

- **¿Considera que la interfaz es amigable e intuitiva? ¿Qué cambiaría?**

La interfaz es funcional, pero está desactualizada y no es lo suficientemente intuitiva. Cambiaría el diseño para hacerlo más moderno y fácil de navegar, sobre todo para los usuarios menos técnicos.

- **¿Le gustaría tener más opciones de personalización en la interfaz?**

Definitivamente, más opciones de personalización permitirían a los usuarios ajustar la interfaz a sus necesidades específicas, lo cual mejoraría la experiencia general de uso.

Rendimiento:

- **¿Qué tan rápido responde el sistema en tareas críticas?**

En tareas críticas, el sistema es lento, especialmente cuando se accede a grandes volúmenes de datos. Esto afecta la eficiencia operativa.

- **¿Ha notado una disminución en el rendimiento del sistema a medida que se manejan más datos?**

Sí, hemos notado que el rendimiento del sistema disminuye significativamente a medida que el volumen de datos crece. Esto representa un riesgo para la escalabilidad futura.

Seguridad:

- **¿Cómo evalúa la seguridad del sistema para proteger datos sensibles?**

La seguridad actual del sistema es adecuada, pero no cumple con los estándares avanzados necesarios para proteger completamente los datos sensibles en el sector salud.

- **¿Cree que es necesario implementar mejoras en la protección de datos?**

Sí, es necesario mejorar la encriptación de datos y adoptar estándares como ISO/IEC 27001 para asegurar una mejor protección de la información sensible.

Compatibilidad:

- **¿El software es compatible con los dispositivos y sistemas que utiliza regularmente?**

El software tiene problemas de compatibilidad con tecnologías emergentes y con otros sistemas de gestión de la organización, lo que limita nuestra capacidad operativa.

- **¿Qué problemas de compatibilidad ha experimentado?**

Hemos experimentado incompatibilidad con dispositivos móviles y con algunos sistemas de integración de datos, lo que dificulta la interoperabilidad.

Mantenimiento y actualizaciones:

- **¿Con qué frecuencia cree que el sistema debería recibir actualizaciones?**

El sistema debería recibir actualizaciones trimestrales para mantener su rendimiento y mejorar su capacidad de integración con nuevas tecnologías.

- **¿Ha tenido problemas con las actualizaciones del sistema?**

Sí, en ocasiones las actualizaciones han generado inestabilidad temporal, lo que ha afectado la continuidad de las operaciones.

Soporte técnico:

- **¿El soporte técnico ha sido efectivo en resolver los problemas con el software?**

El soporte técnico no ha sido eficaz. Hemos experimentado retrasos significativos en la resolución de problemas críticos.

- **¿Cómo calificaría los tiempos de respuesta del soporte técnico?**

Los tiempos de respuesta son lentos, lo que ha afectado la eficiencia en la resolución de problemas.

Productividad:

- **¿El software contribuye a aumentar su productividad en el trabajo?**

Aunque el software contribuye a ciertas áreas, la falta de integración y los problemas de rendimiento disminuyen su capacidad para mejorar la productividad en su totalidad.

- **¿Existen funciones repetitivas o procesos manuales que podrían ser automatizados?**

Sí, varios procesos como la gestión de mantenimiento preventivo y la generación de reportes podrían ser automatizados para ahorrar tiempo y mejorar la eficiencia.

Escalabilidad:

- **¿Cree que el sistema puede manejar eficientemente el crecimiento de la organización?**

No, el sistema actual no puede manejar eficientemente el crecimiento de la organización, ya que no es lo suficientemente escalable para integrarse con nuevas tecnologías y gestionar grandes volúmenes de datos.

- **¿Qué tan flexible es el sistema para adaptarse a nuevos requerimientos?**

El sistema es poco flexible para adaptarse a nuevos requerimientos, ya que su arquitectura actual dificulta las personalizaciones y la incorporación de funcionalidades adicionales.

4. **Cierre:**

- Resumen de los puntos clave discutidos.
- Agradecimiento por la participación y explicación de los próximos pasos en el proceso de modernización.

B. Guía de entrevista

[Enlace a la entrevista](#)

C. Respuestas de entrevista

[Enlace a la entrevista con respuestas](#)

D. Guía de satisfacción

[Enlace a la guía de satisfacción](#)

E. Respuestas de la guía de satisfacción

[Enlace a la entrevista con respuestas](#)

F. Lista de verificación de buenas prácticas

[Enlace a la lista de verificación de buenas prácticas](#)

G. Revisión documental

[Enlace a la revisión documental](#)

H. Benchmarking

[Enlace a la matriz de *Benchmarking*](#)

I. Mapa de alineación de requerimientos

Elemento	Descripción	Requerimiento Asociado (ID)
Fortalezas	Capacidad de identificación de equipos médicos mediante QR y RFID.	- Mantener compatibilidad con QR y RFID (RF-001, RF-035).
	Base de datos consolidada de equipos médicos.	- Mejorar consultas de base de datos para búsqueda avanzada (RF-010, RF-004).
	Control de acceso por roles y autenticación.	- Implementar control de acceso basado en roles y seguridad avanzada (RF-006, RF-013, RNF-020).
	Generación de reportes detallados y personalizables.	- Mejorar la generación de reportes en varios formatos (RF-007, RF-023, RNF-021).
	Capacidad para gestionar expedientes de mantenimiento.	- Optimizar el seguimiento de mantenimiento preventivo y correctivo (RF-002, RF-005).
	Establecer niveles de seguridad para usuarios.	- Implementar autenticación de dos factores y encriptación avanzada (RNF-009, RNF-019).
Oportunidades	Integrar tecnologías de IA y análisis predictivo para mantenimiento.	- Implementar mantenimiento predictivo utilizando IA (RNF-024, RNF-038).
	Expansión a interfaces móviles y dispositivos de terceros.	- Compatibilidad con dispositivos móviles para acceso remoto (RF-008, RF-050, RNF-017).
	Generación de reportes automatizados sobre uso y mantenimiento.	- Implementar generación de informes automáticos (RF-023, RF-007).
	Integración con sistemas hospitalarios y ERP.	- Sincronización de datos con sistemas hospitalarios y ERP (RF-008, RF-050).
	Optimización del almacenamiento de datos en la nube.	- Implementar almacenamiento en la nube para datos de mantenimiento (RNF-031, RNF-022).

Elemento	Descripción	Requerimiento Asociado (ID)
Debilidades	Arquitectura desactualizada que limita la integración.	- Modernizar hacia microservicios y una arquitectura modular (RNF-018, RNF-026).
	Complejidad en la gestión de contratos de mantenimiento.	- Mejorar la gestión de contratos con terceros y su monitoreo (RF-024, RF-048).
	Baja escalabilidad y lentitud en respuesta del sistema.	- Escalabilidad para hasta 1,000 usuarios simultáneos (RNF-007, RNF-008).
	Proceso de actualización manual y propenso a errores.	- Implementar actualizaciones automáticas y un mecanismo de rollback (RNF-023, RNF-015).
	Falta de retroalimentación en la toma de requerimientos.	- Implementar un mecanismo de retroalimentación para los usuarios del sistema (RF-027, RF-051).
	Dificultad para incorporar nuevas tecnologías debido a la arquitectura actual.	- Rediseñar la arquitectura para facilitar la integración de nuevas tecnologías (RNF-016, RNF-032).
	Documentación técnica insuficiente.	- Mejorar la documentación técnica y el código fuente (RNF-029, RNF-016).
Amenazas	Vulnerabilidades de seguridad por falta de actualizaciones.	- Implementar estándares de seguridad avanzados y actualizaciones regulares (RNF-011, RNF-025, RNF-034).
	Competidores que adoptan tecnologías emergentes más rápidamente.	- Implementar tecnologías emergentes como inteligencia artificial y análisis avanzado de datos (RNF-013, RNF-038).
	Riesgo de quedarse obsoleto frente a competidores que implementan actualizaciones más frecuentes.	- Adaptar el sistema a estándares internacionales de compatibilidad y escalabilidad (RNF-026, RF-045).
	Falta de integración con sistemas modernos y de terceros.	- Integrar con sistemas modernos y tecnologías emergentes (RF-052, RNF-030, RNF-033).

Elemento	Descripción	Requerimiento Asociado (ID)
Limitaciones	Plazo limitado para implementación del proyecto (6 meses).	- Priorización de los cambios críticos y fases de mejoras continuas (RNF-028, RF-015).
	Complejidad del código fuente que dificulta el mantenimiento.	- Simplificación del código fuente y documentación siguiendo buenas prácticas (RNF-016, RNF-029).
	Presupuesto limitado para las fases iniciales de modernización.	- Optimización de costos mediante la adopción de tecnologías rentables (RNF-036, RNF-012).
	Requisitos de compatibilidad con normativas regulatorias.	- Cumplir con regulaciones locales e internacionales, incluyendo normativas de seguridad de datos (RNF-035, RF-041).
	Falta de automatización en los procesos de actualización y mantenimiento.	- Implementar automatización en los procesos de mantenimiento y actualizaciones (RNF-040, RNF-023).
	Infraestructura tecnológica limitada para soportar nuevas capacidades.	- Mejorar la infraestructura tecnológica para soportar las nuevas capacidades y escalabilidad del sistema (RNF-031, RNF-042).

J. PESTEL

Factor	Descripción
<i>Político</i>	El software debe cumplir con las normativas y regulaciones del sector salud, como la protección de datos sensibles de los pacientes. Se debe asegurar que la modernización cumpla con las leyes y políticas de privacidad como el Reglamento General de Protección de Datos (GDPR) o la Ley de Protección de Datos Personales. Es fundamental evaluar los contratos existentes para asegurar la viabilidad de las modificaciones y compatibilidad con las nuevas tecnologías.
<i>Económico</i>	Se debe evaluar los costos asociados a la modernización y la posible necesidad de inversiones adicionales. Considerar los ingresos perdidos debido a la ineficiencia del sistema actual y los beneficios económicos que la modernización aportaría. Mejorar el software para mantener la competitividad en el mercado, ofreciendo soluciones eficientes y modernas que se adapten a las demandas de los clientes.
<i>Social</i>	Los usuarios finales, como técnicos y administrativos, deben aceptar y adaptarse a la nueva versión del software. La interfaz debe ser amigable y adaptarse a las expectativas actuales para mejorar la satisfacción del usuario. Existe una creciente demanda por soluciones tecnológicas modernas, por lo que el software debe evolucionar para cumplir con las expectativas de los usuarios.
<i>Tecnológico</i>	Gestemed presenta una arquitectura desactualizada que limita su eficiencia y compatibilidad con tecnologías modernas. La modernización debe considerar la integración con tecnologías como la nube, inteligencia artificial, y herramientas de análisis de datos. Es crucial asegurar que Gestemed sea compatible con otros sistemas y dispositivos utilizados en el sector de la salud.
<i>Ecológico</i>	Implementar soluciones tecnológicas que reduzcan el consumo energético del sistema. Considerar prácticas para la eliminación segura y sostenible de equipos obsoletos, si aplica. Evaluar si la infraestructura tecnológica puede ser optimizada para reducir su huella ambiental.
<i>Legal</i>	Asegurar que el software cumple con leyes como la GDPR y la Ley de Protección de Datos Personales en el manejo de la información de los pacientes. Verificar que la modernización no infrinja derechos de propiedad intelectual, licencias de software, o patentes existentes. Asegurar el cumplimiento de normativas como ISO/IEC 27001 para garantizar la seguridad del sistema.

K. Informe viabilidad

Descripción del proyecto

El proyecto busca modernizar la arquitectura del software Gestemed, un sistema utilizado para la gestión de equipos médicos, que ha quedado obsoleto debido a su arquitectura anticuada y falta de compatibilidad con tecnologías actuales. La modernización se centrará en mejorar la eficiencia operativa, aumentar la satisfacción de los usuarios y abrir nuevas oportunidades de negocio.

Análisis costo-beneficio

A continuación, se desglosan los costos y beneficios del proyecto, considerando que los principales recursos serán utilizados durante este tiempo.

Costos estimados

Costos directos:

Equipo de desarrolladores: El salario promedio de un desarrollador en Costa Rica es de ₡24,000,000 anuales, lo que equivale a ₡12,000,000 por 6 meses. Para 5 desarrolladores:

$$5 \times 12\,000\,000 = 60\,000\,000$$

Consultores tecnológicos: Se estima un costo de ₡11,000,000 por los servicios de consultoría tecnológica durante los 6 meses de implementación.

Licencias de software y herramientas de desarrollo: El costo de las licencias para el periodo de 6 meses será ₡5,500,000.

Pruebas y validación del software: Las pruebas para garantizar la calidad del software durante los 6 meses costarán ₡8,250,000.

Capacitación del personal: El costo de capacitar al personal en el uso del software será de ₡2,750,000.

Costos indirectos:

Gastos administrativos y de gestión: Se destinan ₡2,750,000 para cubrir los gastos administrativos y de gestión del proyecto en este periodo.

Pérdida de productividad temporal: Durante la capacitación y migración al nuevo sistema, se estima una pérdida de productividad de ₡5,500,000.

Total de costos estimados para 6 meses:

$$60\,000\,000 + 11\,000\,000 + 5\,500\,000 + 8\,250\,000 + 2\,750\,000 + 2\,750\,000 \\ = \text{€}95\,750\,000$$

Beneficios estimados (a partir del mes 7)

Los beneficios del proyecto comenzarán a percibirse inmediatamente después de los 6 meses de implementación, mejorando la eficiencia operativa, reduciendo errores y aumentando las oportunidades de negocio.

1. **Incremento en la eficiencia operativa:** La modernización reducirá en un 30% las tareas manuales, generando un ahorro de €16,500,000 al año.
2. **Reducción de errores y fallos:** La estabilidad mejorada del sistema reducirá los costos por errores en aproximadamente €11,000,000 anuales.
3. **Aumento de oportunidades de negocio:** Con el software modernizado, se espera un incremento del 10% en las ventas anuales, lo que generará €27,500,000 adicionales al año.

Total de beneficios anuales estimados:

$$16\,500\,000 + 11\,000\,000 + 27\,500\,000 = 55\,000\,000$$

Cálculos de retorno financiero

Retorno de la inversión (ROI): El retorno de la inversión se calcula comparando los beneficios anuales con el costo total del proyecto implementado en 6 meses:

$$ROI = (55\,000\,000 / 95\,750\,000) \times 100 = 57.43\%$$

Período de recuperación de la inversión (*Payback Period*): El período de recuperación es el tiempo que tomará recuperar la inversión inicial con los beneficios generados a partir del mes 7:

$$\text{Período de recuperación} = 95,750,000 / 55,000,000 = 1.74 \text{ años}$$

Riesgos y mitigación

El proyecto de modernización del software Gestemed presenta ciertos riesgos que deben ser gestionados adecuadamente para asegurar su éxito. A continuación, se identifican algunos de los riesgos más relevantes y las estrategias de mitigación correspondientes.

Riesgos técnicos:

1. **Problemas durante la migración:** Durante la modernización del software, podría haber dificultades en la migración de datos o en la integración con otros sistemas.

- **Mitigación:** Se realizarán pruebas exhaustivas de migración en entornos de prueba para asegurar que los datos se transfieran correctamente y sin pérdida de información.
2. **Incompatibilidad con sistemas actuales:** La actualización del software podría generar incompatibilidades con los sistemas que Key Tech ya utiliza.
- **Mitigación:** Se garantizará que el software modernizado sea compatible con las plataformas existentes mediante el uso de estándares de interoperabilidad.

Riesgos financieros:

1. **Beneficios retrasados:** Los beneficios del proyecto podrían no materializarse inmediatamente después de la implementación.
- **Mitigación:** Se establecerán métricas claras de éxito y se monitorizarán los indicadores clave de rendimiento (KPIs) desde el mes 7 en adelante.

10. Anexos

10.1. Anexo I Carta de aprobación filológica

INSTITUTO TECNOLÓGICO DE COSTA RICA (TEC)
Escuela de Administración de Tecnologías de Información

Estimados Señores:

Por este medio hago constar que he revisado el Trabajo Final de Graduación titulado: **“Propuesta de modernización de arquitectura del software Gestemed en Key Tech”**, presentado por la postulante: Dayana Alexandra Hernández Hernández, para optar al grado de Licenciatura en Administración de Tecnologías de la Información, en el cual he corregido los errores de redacción, gramática, estilo, puntuación y punto de vista del discurso científico.

Así mismo hago constar que el estudio es portador de los siguientes valores:

- Alto rigor científico en toda la investigación.
- El proyecto hace un aporte capital a la empresa Key Tech, fortaleciendo su competitividad en el mercado con una tecnología emergente que le abre las puertas a nuevas oportunidades de negocio.

Dado en San Isidro de El General, el 02 de noviembre del 2024, a solicitud de los interesados.

Cordialmente:

ERIC
GONZALEZ
CONDE
(FIRMA)



Firmado digitalmente
por ERIC GONZALEZ
CONDE (FIRMA)
Fecha: 2024.11.02
15:46:34 -06'00'

Eric González Conde
Lic. En Filología Universidad Central, Marta Abreu, Cuba.
Lector Externo del Consejo Editor de la editorial de la EUNED
Su obra se ha publicado en seis países de dos continentes
Carné 1855. Cédula 119200332703

10.2. Anexo II Carta de aprobación de minutas de reunión de TFG

Cartago, Costa Rica.

02 de noviembre 2024.

Asunto: Aceptación de minutas de TFG

Por esto medio yo, Dayana Alexandra Hernández Hernández, carné universitario 2019390563, estudiante de la carrera Administración de Tecnología de Información, solicito la aprobación de las minutas realizadas durante el TFG durante el segundo semestre 2024.

Esta aprobación incluye las siguientes minutas:

- Minuta N° 1: 08/08/2024
- Minuta N° 2: 26/09/2024
- Minuta N° 3: 17/10/2024
- Minuta N° 4: 29/07/2024
- Minuta N° 5: 17/08/2024



Dayana Alexandra

Hernández Hernández,

TEC | Tecnológico
de Costa Rica

Digitally signed by LUIS
JAVIER CHAVARRIA
SANCHEZ (FIRMA)
Date: 2024.11.02
14:39:01 -06'00'

Máster. Luis Javier

Chavarría Sánchez

10.3. Anexo III Plantilla de minutas

Minuta N° ##			
Objetivo			
Medio		Fecha Hora	
Temas tratados			
•			
Participantes	•		

10.4. Anexo IV Minuta uno con contraparte

Minuta N° 01			
Objetivo	Presentación de la parte académica y empresarial		
Medio	Zoom	Fecha	08/08/2024
		Hora	4:00 pm
Temas tratados			
<ul style="list-style-type: none">• Confirmación de próximas reuniones• Presentación de hitos académicos• Evaluaciones por parte de la Empresa• Reglamento del Trabajo Final de Graduación• Varios			
Participantes	<ul style="list-style-type: none">• Néstor Rodríguez• Alexandra Hernández.• Luis Chavarría.		

10.5. Anexo V Minuta dos con contraparte

Minuta N° 02			
Objetivo	Reunión mensual de la parte académica y empresarial		
Medio	Zoom	Fecha	26/09/2024
		Hora	4:00 pm
Temas tratados			
<ul style="list-style-type: none">• Confirmación de próxima reunión• Evaluaciones por parte de la empresa• Temas sobre la empresa• Varios			
Participantes	<ul style="list-style-type: none">• Néstor Rodríguez• Alexandra Hernández.• Luis Chavarría.		

10.6. Anexo VI Minuta tres con contraparte

Minuta N° 03			
Objetivo	Reunión mensual de la parte académica y empresarial		
Medio	Zoom	Fecha	16/10/2024
		Hora	4:00 pm
Temas tratados			
<ul style="list-style-type: none">• Recordatorio tera evaluación por parte de la contraparte empresarial.• Sobre el avance del documento académico de TFG.• Observaciones o comentarios sobre el desarrollo del TFG.• Varios.			
Participantes	<ul style="list-style-type: none">• Néstor Rodríguez• Alexandra Hernández.• Luis Chavarría.		

10.7. Anexo VII Salarios mínimos según MTSS

Acomodador de parqueos (no chófer)	TONC	¢11,953,65	Contador privado *	TMED	¢422.798,93	Limpiador de piscinas (sin químicos)	TONC	¢11,953,65
Acompañante en buseta escolar	TONC	¢11,953,65	Contador privado *	DES	¢562.756,90	Limpiador de tanques sépticos	TOC	¢13.448,72
Adiestrador de animales	TOC	¢13.448,72	Contador privado *	Bach.	¢638.299,51	Llantero	TOSC	¢12.998,72
Agente de aduanas	TES	¢24.231,19	Contador privado*	Lic.	¢765.985,67	Locutor de radioemisora	TOE	¢15.613,91
Agente de seguridad*	TOSCG	¢389.961,60	Copiloto (Primer Oficial de Aviación)	TES	¢24.231,19	Locutor de televisión	TES	¢24.231,19
Agente de seguridad custodio de valores*	TOCG	¢403.461,60	Cortador de piezas tela (patrones)	TOC	¢13.448,72	Luminotécnico de televisión	TES	¢24.231,19
Agente de ventas *	TOCG	¢403.461,60	Cosedor piezas (prendas a máquina)	TOC	¢13.448,72	Maestro de obras (Construcción)	TOE	¢15.613,91
Albañil	TOC	¢13.448,72	Costurera (Modista)	TOE	¢15.613,91	Manicurista; Maquillador	TOC	¢13.448,72
Alistador automotriz (lijador)	TOSC	¢12.998,72	Counter (Vendedor de pasajes) *	TOCG	¢403.461,60	Masajista	TOC	¢13.448,72
Analista de crédito *	TOCG	¢403.461,60	Demostrador (display)	TONC	¢11.953,65	Mantenimiento correctivo de cómputo	TOE	¢15.613,91
Animador de eventos	TOC	¢13.448,72	Demostrador-vendedor	TOSC	¢12.998,72	Mantenimiento de edificios	TOC	¢13.448,72
Aplanchador con equipo de vapor	TOC	¢13.448,72	Dependiente	TOSC	¢12.998,72	Mantenimiento preventivo de cómputo	TOC	¢13.448,72
Aserrador (usa sierra de motor)	TOC	¢13.448,72	Despachador de vuelo	TES	¢24.231,19	Mecánico de calderas (Mantenimiento)	TOE	¢15.613,91
Asistente de abogacía *	TOEG	¢463.248,99	Diagramador en artes gráficas	TOE	¢15.613,91	Mecánico general	TOC	¢13.448,72
Asistente domicilio	TOE	¢15.613,91	Digitador	TOC	¢13.448,72	Mecánico precisión	TOE	¢15.613,91
(Cuidados especiales)			Ebanista	TOE	¢15.613,91	Mecánico máquinas de coser industrial	TOE	¢15.613,91
Auxiliar agente de aduanas	TOE	¢15.613,91	Educador aspirante sin título *	TOEG	¢463.248,99	Mensajero *	TONCG	¢358.609,50
Auxiliar de contabilidad *	TOCG	¢403.461,60	Electricista	TOC	¢13.448,72	Misceláneo *	TONCG	¢358.609,50
Auxiliar dental	TOE	¢15.613,91	Electromecánico	TOE	¢15.613,91	Misceláneo hogares de la tercera edad	TONC	¢11.953,65
Ayudante de cocina	TOSC	¢12.998,72	Empacador, etiquetador (manual)	TONC	¢11.953,65	Monitoreador de cámaras de video*	TOSCG	¢389.961,60
Ayudante en mecánica general	TOSC	¢12.998,72	Empleado de despacho	TOSC	¢12.998,72	Montacarguista	TOSC	¢12.998,72
Ayudante de operario, construcción	TOSC	¢12.998,72	Encuestador *	TOSCG	¢389.961,60	Mucama	TONC	¢11.953,65
Baqueano	TOSC	¢12.998,72	Enderezador automotriz	TOC	¢13.448,72	Niñera, excepto en el hogar del niño	TONC	¢11.953,65
Barbero	TOC	¢13.448,72	Entrenador de fútbol (Primera y segunda división)	TOE	¢15.613,91	Niñera en el hogar del niño (Trabajo doméstico) *		¢246.624,40
Barista	TOC	¢13.448,72	Escaneador inventarios (hand held)*	TOSCG	¢389.961,60	Oficinista (General) *	TOSCG	¢389.961,60
Bodeguero (Encargado) *	TOSCG	¢389.961,60	Esteticista	TOE	¢15.613,91	Operador de aplanador	TOSC	¢12.998,72
Bodeguero (Peón) *	TONCG	¢358.609,50	Estibador por kilo frutas y vegetales		¢0,0821	Operador de "araña" (Serigrafía)	TOC	¢13.448,72
Cajero *	TOCG	¢403.461,60	Estibador por movimiento		¢433,07	Operador de cabina de radioemisora	TOE	¢15.613,91
Camarero	TONC	¢11,953,65	Estibador por tonelada		¢101,56	Operador de caldera	TOC	¢13.448,72
Camarógrafo de prensa	TES	¢24.231,19	Estilista	TOC	¢13.448,72	Operador de excavadora	TOE	¢15.613,91
Cantante	TOC	¢13.448,72	Florista	TOC	¢13.448,72	Operador de grúa estacionaria	TOE	¢15.613,91
Cantinero	TOSC	¢12.998,72	Fontanero	TOC	¢13.448,72	Operador de maquinaria pesada	TOC	¢13.448,72
Capitán de embarcación	TOE	¢15.613,91	Fotocopiador (Centro fotocopiado)	TOSC	¢12.998,72	Operador de radio-taxi	TOC	¢13.448,72
Cargador cilindros gas y extintores	TOSC	¢12.998,72	Fotógrafo de prensa	TOE	¢15.613,91	Operario en construcción	TOC	¢13.448,72
Carnicero empleado de despacho	TOSC	¢12.998,72	Fresador (Metalmecánica)	TOE	¢15.613,91	Ordeñador a mano	TONC	¢11,953,65
Carnicero destazador	TOC	¢13.448,72	Fumigador en casas y edificios	TOSC	¢12.998,72	Ordeñador con máquina	TOC	¢13.448,72
Carpintero	TOC	¢13.448,72	Futbolista Primera División	TOE	¢15.613,91	Panadero	TOC	¢13.448,72
Catador	TOE	¢15.613,91	Futbolista Segunda División	TOC	¢13.448,72	Parrillero	TOSC	¢12.998,72
Cerrajero	TOC	¢13.448,72	Gestor de redes sociales (Community Manager, diseña artes finales)	TOE	¢15.613,91	Pastelero	TOC	¢13.448,72
Chapulinerio	TOC	¢13.448,72	Gestor de redes sociales (Community manager, monitorea y da respuesta).*	TOSCG	¢389.961,60	Pelador de camarón	TONC	¢11,953,65
Chef	TOE	¢15.613,91	Gestor redes sociales (Community manager, lleva datos y los interpreta).	TOCG	¢403.461,60	Peón agrícola (labores manuales)	TONC	¢11,953,65
Chequeador de buses	TONC	¢11,953,65	Gondolero	TONC	¢11,953,65	Peón agrícola (motoguadaña y similares)	TOSC	¢12.998,72
Cobrador *	TOSCG	¢389.961,60	Guía de turismo	TOC	¢13.448,72	Peón de carga y descarga	TONC	¢11,953,65
Cocinero	TOC	¢13.448,72	Hojalatero	TOC	¢13.448,72	Peón de construcción	TONC	¢11,953,65
Coctelero (Bartender o Barwoman)	TOC	¢13.448,72	Instalador vidrios (corta, pone marcos)	TOC	¢13.448,72	Peón de jardín (labores manuales)	TONC	¢11,953,65
Conductor ambulancia (socorrismo)	TOC	¢13.448,72	Instructor de bailes	TOC	¢13.448,72	Peón de jardín (motoguadaña y similares)	TOSC	¢12.998,72
Conductor de bus (no cobra)	TOC	¢13.448,72	Instructor de gimnasio (sin título)	TOC	¢13.448,72	Peón de embarcación	TONC	¢11,953,65
Conductor de bus (cobrador)	TOE	¢15.613,91	Jardinero (diseña jardines)	TOC	¢13.448,72	Pilero (lavador de platos)	TONC	¢11,953,65
Conductor de tráiler	TOE	¢15.613,91	Joyero	TOC	¢13.448,72	Pintor automotriz	TOE	¢15.613,91
Conductor de vehículo liviano	TOSC	¢12.998,72	Lavador y encerador de carros	TONC	¢11,953,65	Pintor de brocha gorda	TOC	¢13.448,72
Conductor de vehículo pesado	TOC	¢13.448,72				Pistero (dispensa combustibles)	TOSC	¢12.998,72
Conductor microbús (menos 11 pasaj.)	TOSC	¢12.998,72				Pistero-cobrador (responsable dinero)*	TOCG	¢403.461,60
Conserje *	TONCG	¢358.609,50				Pizzero (arma y homea pizza)	TOSC	¢12.998,72

Propuesta de modernización de arquitectura del software Gestemed en Key Tech

Pizzero (cocinero)	TOC	¢13.448,72
Procesador inventarios (hand hell)	TOC	¢13.448,72
Programador computación (sin título)	TOE	¢15.613,91
Programador en radioemisoras	TOE	¢15.613,91
Proveedor *	TOCG	¢403.461,60
Recepcionista*	TOSCG	¢389.961,60
Recolectores de basura y reciclaje	TONC	¢11.953,65
Recolector de café por cajuela		¢1.138,14
Recolector de coyol	TONC	¢11.953,65
Relojero	TOC	¢13.448,72
Repostero	TOC	¢13.448,72
Sabanero	TONC	¢11.953,65
Sacristán	TONC	¢11.953,65
Seleccionador manual de residuos	TONC	¢11.953,65
Salonero	TONC	¢11.953,65
Sastre (Prendas a la medida)	TOE	¢15.613,91
Secretaria *	TOCG	¢403.461,60
Secretaria *	TMED	¢422.798,93
Secretaria *	DES	¢562.756,90
Secretaria *	Bach	¢638.299,51
Secretaria *	Lic.	¢765.985,67
Serigrafista (realiza diseño)	TOE	¢15.613,91
Serigrafista (Estampa diseños)	TOC	¢13.448,72
Soldador (Soldaduras Especiales)	TOE	¢15.613,91
Soldador en general	TOC	¢13.448,72
Talador (usa sierra de motor)	TOC	¢13.448,72
Tapicero	TOC	¢13.448,72
Tatuador	TOC	¢13.448,72
Taxista	TOC	¢13.448,72
Técnico en aire acondicionado	TOC	¢13.448,72
Técnico en aparatos ortopédicos	TES	¢24.231,19
Técnico en lentes de contacto	TES	¢24.231,19
Técnico refrig. doméstica/industrial	TES	¢24.231,19
Técnico máq. coser ind. especiales	TES	¢24.231,19
Técnico reparación audio y video	TES	¢24.231,19
Telefonista *	TOSCG	¢389.961,60
Tornero en metal	TOE	¢15.613,91
Trabajo doméstico *		¢246.624,40
Tractorista (Oruga o Llanta)	TOC	¢13.448,72
Vagonetero	TOC	¢13.448,72
Verdulero	TOSC	¢12.998,72
Volantero	TONC	¢11.953,65
Zapatero	TOC	¢13.448,72

Estos salarios contienen, en relación con los salarios mínimos del Decreto N° 43849-MTSS, un incremento del **1,83 %** para todas las categorías del Decreto de Salarios Mínimos, excepto para el servicio doméstico, al que se le otorga un **2,33962%** adicional (**resolución CNS-RG-2-2019**). Asimismo, se aplica un incremento adicional a las siguientes categorías salariales incluidas en la resolución **CNS-RG-6-2020**: **TOSCG** un **0,3986390%**, **TOC** un **0,3955514%** y el **TOEG** un **0,5562880%**, aumentos que se aplican posterior al general.

La lista de salarios mínimos se clasifica con base en los Perfiles Ocupacionales, documento aprobado por el Consejo Nacional de Salarios (Resolución Administrativa 03-2000). Esta lista es una guía ilustrativa, elaborada de conformidad con esos perfiles, contiene algunas ocupaciones seleccionadas por el Departamento de Salarios Mínimos. Las ocupaciones aquí incluidas se basan en las tareas típicas conocidas, por lo que un puesto determinado podría tener una clasificación distinta según sus características y responsabilidades específicas.

CONSULTAS DE SALARIOS

 consulta.salarios@mtss.go.cr
 salario.minimo@mtss.go.cr

2256-2221, 2233-0347, 2222-2168

CONSULTAS LABORALES

Llamada gratuita: 800-TRABAJO (800-872-2256)

Chat institucional: www.mtss.go.cr

En Costa Rica, de acuerdo con la Constitución política, el salario será siempre igual para trabajo igual en idénticas condiciones de eficiencia. Además, no se pueden establecer diferencias por consideración de edad, sexo nacionalidad o etnia.



GOBIERNO
DE COSTA RICA

DEPARTAMENTO DE
SALARIOS MÍNIMOS

LISTA DE SALARIOS MÍNIMOS SECTOR PRIVADO AÑO 2024

Según Decreto N°44293-MTSS, publicado en la Gaceta N°232, Alcance N°250 del 14 de diciembre del 2023. Rige 01 de enero del 2024

SIGLAS Y SALARIOS MÍNIMOS

TONC	Trabajador en Ocupación No Calificada	¢ 11.953,65
TOSC	Trabajador en Ocupación Semicalficada	¢ 12.998,72
TOC	Trabajador en Ocupación Calificada	¢ 13.448,72
TOE	Trabajador en Ocupación Especializada	¢ 15.613,91
TES	Trabajador de Especialización Superior	¢ 24.231,19
TONCG	Trabajador en Ocupación No Calificada (Genérico)	¢ 358.609,50
TOSCG	Trabajador en Ocupación Semicalficada (Genérico)	¢ 389.961,60
TOCG	Trabajador en Ocupación Calificada (Genérico)	¢ 403.461,60
TMED	Técnico Medio en Educación Diversificada	¢ 422.798,93
TOEG	Trabajador en Ocupación Especializada (Genérico)	¢ 463.248,99
TEdS	Técnico de Educación Superior	¢ 521.053,17
DES	Diplomado de Educación Superior	¢ 562.756,90
Bach.	Bachiller Universitario	¢ 638.299,51
Lic.	Licenciado Universitario	¢ 765.985,67

***Salario Mínimo Mensual.**

El Salario Mínimo que no tiene ninguna indicación (*), está por jornada ordinaria

Para mayor información y debido a que se han hecho circular algunas listas alteradas, se sugiere consultar personalmente en el Departamento de Salarios Mínimos en Barrio Tournon, Edificio Centro Comercial Tournon, Ministerio de Trabajo y Seguridad Social, primer piso.

Esta lista está disponible en:
www.mtss.go.cr

DOCUMENTO GRATUITO
Prohibida su reproducción y venta

10.8. Anexo VIII Primera evaluación de la contraparte

[Enlace primera evaluación](#)

10.9. Anexo IX Segunda evaluación de la contraparte

[Enlace segunda evaluación](#)

10.10. Anexo X Tercera evaluación de la contraparte

[Enlace tercera evaluación](#)