Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y Naturales y Agrimensura

Trabajo Final de la Maestría en Tecnologías de la Información

Proceso para el mantenimiento del software. Propuesta para una PyME de servicios de la Región NEA.

Autora: Lic. Luciana Marcela Aguirre

Directora: Dra. Sonia Itatí Mariño

Año 2024

Dedicatoria

"A mis amigos y seres queridos, quienes han estado a mi lado en los momentos buenos y difíciles, brindándome ánimo y alegría.

A mis profesores y mentores, por su guía experta, estímulo intelectual y confianza en mis capacidades.

A todas las personas que una u otra manera han contribuido a este logro, su influencia no ha pasado desapercibida.

Este trabajo está dedicado a todos ustedes, con profundo agradecimiento y afecto."

Resumen

El enfoque principal de este trabajo consiste en diseñar la documentación de un proceso de mantenimiento de software adaptado a las necesidades específicas de las pequeñas y medianas empresas (PyMEs) del sector transporte en la región NEA. En este contexto, se resalta la importancia crucial de mantener el software funcional y actualizado en un entorno empresarial dinámico y complejo.

La documentación del proceso de mantenimiento de software se ha desarrollado siguiendo las directrices de **Mantelasoft** y la norma **ISO/IEC 12207**, aplicando además la metodología ágil **Scrum**. El mantenimiento del software abarca ajustes, modificaciones de código, mejoras en el diseño y corrección de errores una vez que el sistema está en producción. Estos cambios deben adaptarse a las evaluaciones continuas del entorno del software y a los requisitos en constante evolución.

En resumen, la implementación de este plan de mantenimiento establece una base sólida para futuras mejoras y optimizaciones en la gestión de la tecnología de la información empresarial. Esto contribuye al éxito continuo de la empresa en un entorno competitivo, garantizando la calidad y eficiencia de sus sistemas de software.

Palabras clave: Ingeniería del Software, mantenimiento del software, procesos, empresas.

Abstract

The main focus of the work consists of designing the documentation of a software maintenance process adapted to the specific needs of small and medium-sized enterprises (SMEs) in the transportation sector in the NEA region. In this context, the crucial importance of keeping software functional and up-to-date in a dynamic and complex business environment is highlighted.

The documentation of the software maintenance process has been developed following the Mantelasoft guide and the ISO/IEC 12207 standard, and the agile Scrum methodology has been applied. Software maintenance encompasses adjustments, code modifications, design improvements, and bug fixes once the system is in production. These changes must adapt to evaluations in the software environment and changing requirements.

In summary, the implementation of this maintenance plan establishes a solid foundation for future improvements and optimizations in the management of business information technology. This contributes to the company's continued success in a competitive business environment, ensuring the quality and efficiency of its software systems.

Key words: Software Engineering, Software Maintenance, Processes, Companies

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido de manera significativa a la realización de este trabajo de investigación.

En primer lugar, deseo agradecer a mi directora de tesis Sonia Mariño, por su orientación experta, apoyo constante y dedicación incansable a lo largo de este proceso. Sus consejos sabios fueron fundamentales para dar forma a este trabajo y expandir mis horizontes académicos.

Mi gratitud se extiende a colegas, compañeros de trabajos, quienes compartieron ideas, recursos y experiencias, creando un ambiente de colaboración y aprendizaje enriquecedor.

No puedo pasar por alto el apoyo incondicional de mi padre y amigos, quienes estuvieron a mi lado durante los momentos desafiantes y celebraron mis logros con entusiasmo. Su amor, compresión y aliento fueron el motor que me impulsó a seguir adelante.

Finalmente, agradezco a todas las personas que de alguna manera contribuyeron a la realización de este trabajo, su aporte ha sido invaluable.

Este logro no habría sido posible sin el respaldo y la colaboración de todos ustedes. A cada uno de ustedes, mi más profundo agradecimiento.

Índice de Contenido

Contenido

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1 Introducción

La actividad empresarial y su forma de administración o gestión han evolucionado desde la revolución industrial con el crecimiento de los servicios. El desarrollo tecnológico y científico, así como el mercado en su sentido más amplio, incluyendo el transporte y las comunicaciones, han determinado numerosos de estos cambios, poniendo énfasis en la competitividad y sostenibilidad vinculadas a su contexto operacional y de mercado. La función del mantenimiento debe garantizar el buen desarrollo de los procesos que soportan la eficacia de las funciones de los sistemas.

El mantenimiento de software es una actividad crítica para las empresas del Sector y Servicios Informáticos (SSI), ya que estas dependen del software para automatizar sus áreas de trabajo. Uno de los principales desafíos en esta etapa es el costo, que frecuentemente se origina por aspectos de mantenibilidad no previstos o no considerados durante el desarrollo del producto software [1].

En este Trabajo Final de Maestría (TFM), se llevó a cabo un análisis exhaustivo de la situación actual de los sistemas informáticos disponibles en una empresa mediana, con el fin de mejorar el mantenimiento correctivo, una estrategia clave para la mejora continua. Este análisis permitió desarrollar un proceso estructurado basado en la metodología Mantelasoft, que está orientada específicamente a la optimización del mantenimiento de software en las pequeñas y medianas empresas (PyMEs) del sector transporte. Como parte del trabajo, se diseñó una plantilla para documentar el mantenimiento correctivo. Para validar este enfoque, se ha seleccionado el sistema informático “Sistema Control de Tráfico” (SCT), un ejemplo representativo que refleja los retos y oportunidades inherentes al mantenimiento correctivo en un entorno empresarial.

Este análisis sirvió como base para profundizar en el mantenimiento correctivo como estrategia clave para la mejora continua. En consecuencia, se ha establecido un proceso y se ha diseñado una plantilla destinada a documentar el mantenimiento correctivo. Para validar este enfoque, se ha escogido el sistema informático “Sistema Control de Tráfico” (SCT).

1.2 Planteamiento del problema

El mantenimiento del software abarca una serie de actividades esenciales que responden a preguntas clave como qué, cuándo, cómo y por qué deben llevarse a cabo [2]. Este proceso

es fundamental para asegurar que el software permanezca funcional, actualizado y adaptado a su entorno específico.

En este contexto, se propone diseñar un proceso de mantenimiento basado en la guía **Mantelasoft** y desarrollar una plantilla para documentar el mantenimiento correctivo, especialmente adaptada a las necesidades de las pequeñas y medianas empresas (PyMEs) del sector servicios en la región NEA.

Para las PyMEs que utilizan herramientas de código abierto, el costo del mantenimiento es relativamente menor [3]. Sin embargo, es crucial contar con un proceso que respalde de manera eficiente este mantenimiento, sobre todo en un entorno dinámico y complejo, caracterizado por la alta rotación del personal en el área de sistemas.

La motivación para este **Trabajo Final de Maestría** surge de la experiencia laboral de la maestranda en el área de Tecnologías de la Información y Comunicación (TIC) en una PyME de servicios de la región NEA. Durante su desempeño, identificó varios problemas clave:

- La empresa cuenta con diversos sistemas informáticos desarrollados tanto por empresas externas como por su propio equipo de desarrollo tecnológico.
- La falta de documentación actualizada en algunos sistemas genera incertidumbre en su uso, lo que incrementa el riesgo de ejecutarlos sin un respaldo adecuado. Esta carencia de claridad puede provocar demoras en la resolución de problemas o la implementación de mejoras, debido a la falta de información necesaria para abordarlos de manera eficiente y oportuna.
- La ausencia de documentación actualizada para los desarrolladores incrementa las horas hombre necesarias para reanalizar los sistemas durante su mantenimiento. La información desactualizada, incompleta o difícil de entender crea dificultades, especialmente si el desarrollador principal no está disponible o si el equipo carece de conocimiento sobre el sistema, complicando así la solución de los problemas identificados.

En este contexto, contar con una documentación de software clara, actualizada y accesible resulta fundamental. La falta de esta documentación puede derivar en una serie de desafíos y problemas:

- **Dificultad para comprender el sistema**: La falta de documentación actualizada dificulta que los desarrolladores comprendan aspectos críticos del sistema, como sus componentes, flujos de datos y reglas de negocio, lo que puede generar malentendidos y errores en la implementación de nuevas funcionalidades o en la corrección de fallos.

- **Aumento en el tiempo de desarrollo**: Sin una documentación clara, los desarrolladores necesitan más tiempo para investigar y entender el código existente, lo que ralentiza el proceso de desarrollo.
- **Mayor riesgo de introducir errores**: La falta de documentación puede aumentar el riesgo de introducir errores, ya que los desarrolladores pueden malinterpretar o desconocer ciertos aspectos del sistema, afectando su funcionamiento.
- **Dificultades en el mantenimiento y actualización del sistema**: Sin documentación actualizada, resulta más complicado realizar cambios, actualizaciones o mantenimiento, lo que puede llevar a que el sistema quede obsoleto o difícil de mantener a largo plazo.
- **Problemas de colaboración**: La ausencia de una documentación clara también puede afectar la colaboración entre desarrolladores y equipos, dificultando la comunicación y el intercambio de conocimientos sobre el sistema.

En resumen, la falta de documentación actualizada genera numerosos desafíos que impactan negativamente en la eficiencia, calidad y mantenibilidad del sistema. Por ello, es crucial mantener la documentación al día y accesible para los desarrolladores y demás usuarios involucrados en el proyecto.

1.3 Objetivos

Objetivo general

- Diseñar un proceso para gestionar el mantenimiento del software, centrándose en el área de TIC de una PyME de la Región NEA.

Para el logro del objetivo general, se definieron los siguientes objetivos específicos:

- Identificar estándares y metodologías vinculadas con la mantenibilidad del software.
- Formular un proceso para el mantenimiento del software, en particular la documentación, basado en Mantelasoft para una PyME de servicios regional.
- Validar la propuesta en el área TIC de la PyME regional dedicada a servicios de transporte regional.

1.4 Alcances

El presente Trabajo Final de Maestría se enfoca en el diseño de un proceso de mantenimiento del software y en la creación de una plantilla para documentar el mantenimiento correctivo.

Se adapta contemplando algunos elementos de la guía Mantelasoft y será implementado en el área TIC, siendo el SCT definido para su validación.
El objetivo de esta propuesta es mejorar el tiempo de respuesta a los usuarios y reducir o evitar interrupciones en el ingreso, las modificaciones y la gestión utilizando el SCT.

1.5 Estructura del Trabajo Final de Maestría

En el capítulo 1 se presenta la introducción y los objetivos del Trabajo Final de Maestría, detallando el alcance y contexto de este trabajo. Esta sección se complementa con los siguientes capítulos.
El capítulo 2 aborda el Marco Teórico, resumiendo el conocimiento disciplinar en el que se centra este TFM. Este capítulo incluye una síntesis del mantenimiento del software, sus diversos tipos y las metodologías identificadas para llevarlo a cabo. Se destaca la construcción de la Ingeniería del Software (IS) como disciplina clave en la gestión de la información, lo cual respalda el cumplimiento del objetivo específico 1.
En el capítulo 3 se presenta el Marco Metodológico seguido para alcanzar los objetivos planteados. Se incluyen los instrumentos diseñados y aplicados, como las encuestas y entrevistas elaboradas para investigar y conocer la opinión de los usuarios.
El capítulo 4 describe la solución propuesta, la cual se fundamenta en el estudio realizado y se relaciona con la situación contextual utilizando el marco metodológico establecido. En este capítulo se define el proceso para el mantenimiento correctivo del software, además de abordar su validación.
En el capítulo 5 se expresan las conclusiones en relación a los productos resultantes de este TFM, específicamente el marco metodológico que sustenta el procedimiento diseñado y validado.
En resumen, los productos generados en este TFM incluyen la propuesta metodológica y el proceso de mantenimiento correctivo de software diseñado para una PyME del sector servicios de transporte.

Capítulo 2

Estado de la Cuestión

Este capítulo presenta un resumen de los conceptos fundamentales que sustentan la definición del proceso propuesto, con un enfoque destacado en la agilidad, en la gestión de proyectos y el mantenimiento del software. Este último es el núcleo central del proceso desarrollado. Asimismo, se destaca el análisis FODA como una herramienta esencial para evaluar las condiciones actuales de la organización. Este análisis permite identificar fortalezas, debilidades, oportunidades y amenazas que influyen en el desarrollo y mantenimiento del software, facilitando la creación de estrategias personalizadas que optimizan el uso de los recursos internos y alinean el plan de mantenimiento con las necesidades reales de la empresa y su entorno competitivo [2].

En el contexto de las PyMEs, particularmente aquellas dedicadas a la prestación de servicios, el mantenimiento de software presenta un desafío considerable debido a los recursos limitados disponibles. Por lo tanto, la propuesta se centra en la creación de un plan de mantenimiento correctivo que aborde los problemas principales identificados por los usuarios del sistema. Este plan también incluye el diseño de una plantilla específica para documentar las actividades de mantenimiento correctivo, lo que facilitará su implementación y seguimiento.

La relación entre la Ingeniería del Software (IS) y el mantenimiento de software es estrecha y crucial. La IS abarca todas las fases de la producción de software, desde la especificación inicial del sistema hasta su implementación y posterior mantenimiento. Dentro de esta disciplina, el mantenimiento de software es una de las áreas fundamentales, definida en la guía SWEBOK [3] como el proceso de modificar un sistema tras su entrega, con el fin de corregir errores, mejorar su rendimiento o adaptarlo a un entorno cambiante. Este enfoque subraya la importancia de contar con un proceso bien definido y adaptado a las necesidades de la organización para garantizar la continuidad y la eficiencia operativa del sistema a lo largo del tiempo.

2.1 Procesos de la Ingeniería del Software

El concepto Ingeniería del Software se propuso originalmente en 1968 durante una conferencia destinada a abordar la denominada “crisis del software”. En aquel momento, quedó claro que los enfoques individuales para el desarrollo de programas no eran adecuados para abordar los desafíos planteados por los sistemas de software grandes y complejos. Estos

sistemas carecían de fiabilidad, excedían los presupuestos previstos y su distribución sufría retrasos.

A lo largo de las décadas de 1970 y 1980, se desarrollaron una variedad de nuevas técnicas y métodos de ingeniería de software, incluyendo la programación estructurada, la ocultación de información y el desarrollo orientado a objetos. También se perfeccionaron herramientas y notaciones estándar que se aplicaron de manera amplia en la industria [4].

Los procesos de software proveen un enfoque sistemático para el desarrollo y evolución del software. Existen procesos específicos que se centran en garantizar atributos de calidad del producto, como ser la mantenibilidad, seguridad o usabilidad [5]. La guía SWEBOK identifica 15 áreas del conocimiento relacionadas con la ingeniería del software, las cuales abarcan diversos aspectos del desarrollo y mantenimiento del software [5] [8].

- requisitos de software,
- diseño de software,
- construcción de software,
- pruebas de software,
- mantenimiento del software,
- gestión de la configuración de software,
- gestión de Ingeniería de software,
- proceso de Ingeniería de software,
- modelos y métodos de Ingeniería de software,
- calidad del software,
- práctica profesional de la Ingeniería de software,
- economía de Ingeniería de software,
- fundamentos de computación,
- fundamentos matemáticos,
- fundamentos de Ingeniería.

El Trabajo Final de Maestría, desde la perspectiva de la Ingeniería del Software, se enfoca en el área del mantenimiento del software, siguiendo la definición establecida por la guía SWEBOK. En esta guía, el proceso de software se define como una serie de actividades y tareas interrelacionadas que transforman productos de trabajo en entradas y en salidas. Estos procesos facilitan la comunicación y coordinación de las personas involucradas en la gestión del proyecto, el desarrollo de software y apoyando la mejora continua, asegurando así la calidad del producto [5].

En Sommerville [6] se define un modelo de proceso de software como “Una representación simplificada de un proceso de software, vista desde una perspectiva especifica”.
Los modelos son herramientas fundamentales que pueden orientar el proceso de mantenimiento del software [6]. La aplicación de modelos de proceso de software puede mejorar considerablemente la eficiencia y efectividad del mantenimiento del software, asegurando que el software continúe satisfaciendo las demandas del usuario a lo largo del tiempo [7].
Entre los estándares más utilizados para los procesos en PyMEs dedicadas al desarrollo de software existen varios estándares relacionados con los procesos para empresas de servicios [9].
Estos estándares proporcionan orientación y mejores prácticas para los procesos en empresas de servicios, lo que permite que optimicen sus operaciones, aumentar la satisfacción del usuario y alcanzar sus objetivos estratégicos.

2.2 Mantenimiento del software

El mantenimiento del software implica realizar cambios o mejoras a un sistema una vez que ha sido entregado o puesto en producción. Estos cambios pueden incluir desde simples modificaciones de código hasta mejoras en el diseño, corrección de errores de especificación o la incorporación de nuevos requisitos [10].
El objetivo principal del mantenimiento es adaptar el sistema a medida que el entorno y los requisitos del usuario evolucionan [11]. Con el tiempo, el mantenimiento de software ha experimentado una evolución significativa, adoptando nuevas políticas e ideologías para responder a las necesidades cambiantes de las empresas. A medida que los sistemas de software se vuelven más complejos, técnicas como la refactorización del código se emplean para optimizar el software, haciéndolo más legible y comprensible para otros desarrolladores. Cualquier trabajo realizado para modificar el software después de su puesta en marcha se considera mantenimiento.
El mantenimiento de software abarca una amplia gama de actividades, desde la corrección de errores hasta la mejora de capacidades y la eliminación u optimización de funciones. Dado que el cambio es inevitable, es esencial desarrollar mecanismos para evaluar, controlar y modificar el software. En este sentido, existen tres tipos principales de mantenimiento de software [10]:

- **Reparación de fallas**
- **Adaptación ambiental**

- **Adición de funcionalidad**

En el diseño del TFM, se optó por enfocarse en la reparación de fallas, ya que los sistemas requieren modificaciones para prolongar su vida útil o mejorar sus características. Esto incluye la corrección de errores, la adaptación a nuevos entornos tecnológicos y la incorporación de nuevas funcionalidades.

El mantenimiento del software se lleva a cabo de manera continua desde su implementación inicial hasta su eventual desuso. Existen diferentes tipos de mantenimiento, cada uno con un enfoque específico:

- **Mantenimiento preventivo**: Consiste en revisar regularmente el software para identificar posibles problemas que podrían surgir en el futuro.
- **Mantenimiento predictivo**: Evalúa el flujo de ejecución del programa para anticipar cuándo ocurrirá una falla, lo que permite realizar ajustes antes de que se presenten problemas.
- **Mantenimiento correctivo**: Corrige los defectos detectados en el software que provocan un comportamiento no deseado. Estas fallas pueden ser de procesamiento, rendimiento, programación, seguridad o estabilidad.
- **Mantenimiento adaptativo**: Se realizan modificaciones en el software cuando es necesario cambiar el entorno en el que opera (sistema operativo, plataforma de hardware, navegador web, etc.) para mantener su funcionalidad.
- **Mantenimiento evolutivo**: La adaptación es obligatoria en este caso, ya que sin ella el software quedaría obsoleto. Un ejemplo común es el ajuste de aplicaciones web cuando cambia la versión de un navegador.
- **Mantenimiento perfectivo**: Se añaden nuevas funcionalidades o características a petición del usuario, adaptando el software a nuevas necesidades. Este tipo de mantenimiento prolonga la vida útil del software y puede evitar la necesidad de una migración costosa a una nueva aplicación.

La guía SWEBOK define el mantenimiento del software como el conjunto de actividades necesarias para brindar soporte rentable al software, tanto antes como después de su entrega [12]. Las actividades previas incluyen la planificación para las operaciones futuras y la capacidad de mantenimiento.

La creciente demanda de sistemas con una vida útil prolongada destaca la importancia de la mantenibilidad del software, que se refiere a la facilidad con la que se puede modificar, mejorar y adaptar un producto [13]. La mantenibilidad está influenciada por varios principios de diseño y arquitectura [14]. Entre los factores que influyen se encuentran [12]:

- **Estabilidad**
- **Facilidad de cambio**
- **Facilidad de análisis**
- **Facilidad de prueba**

Además, otros factores incluyen la modularidad, la documentación, la simplicidad, la trazabilidad, y la consistencia. Estos factores, junto con el uso de un lenguaje de programación estandarizado y la validación y pruebas adecuadas, son esenciales para garantizar la calidad y mantenibilidad del software.

En el Trabajo Final de Maestría, se consideró la documentación como un factor clave para la mantenibilidad del software. Una documentación clara y completa facilita el entendimiento y uso del sistema por parte de los usuarios, y es esencial para el propio proceso de mantenimiento, al proporcionar un registro de los cambios y mejoras realizadas.

Entre los atributos clave de la mantenibilidad destacan:

- **Acoplamiento**: Un alto nivel de acoplamiento entre componentes del sistema aumenta la complejidad del diseño y, por lo tanto, la dificultad de realizar pruebas. Minimizar el acoplamiento es esencial para mejorar la calidad del software.
- **Cohesión**: Una alta cohesión entre los componentes de un sistema mejora su capacidad de adaptación y modificación, especialmente en sistemas modulares.
- **Documentación**: La documentación clara de los diseños y procesos facilita el análisis, la adaptación y el mantenimiento del software.
- **Estandarización**: Seguir estándares de programación definidos mejora la legibilidad y mantenibilidad del código, lo que facilita su análisis y modificación.
- **Facilidad de lectura**: Un código fuente legible contribuye a mejorar la mantenibilidad, portabilidad y reutilización del software.
- **Trazabilidad**: Es fundamental para analizar el impacto de los cambios, la comprensión del sistema y la capacidad de modificación. La falta de trazabilidad reduce la capacidad para implementar cambios con eficacia.

El desarrollo del TFM se centró en la documentación como atributo de la mantenibilidad, tomando como referencia normas, metodologías y guías asociadas al mantenimiento del software, como:

- La norma **ISO/IEC 12207**, que especifica que el proceso de mantenimiento se activa cuando el código o la documentación sufren modificaciones debido a errores o necesidades de mejora [15].

- La guía **SWEBOK**, que trata el mantenimiento del software como una actividad esencial para proporcionar soporte rentable [16].
- El modelo de referencia **Mantus**, que establece subcaracterísticas de mantenibilidad basadas en la norma **ISO/IEC 25010** [1].
- La metodología **Mantema**, que hace que el mantenimiento sea un proceso controlable y medible, con actividades estructuradas en tareas específicas [15].
- La guía metodológica **Mantelasoft**, diseñada para hacer el proceso de desarrollo de software más práctico, económico y eficiente, centrada en la retroalimentación y ajustes en versiones posteriores [17]

2.3 Agilidad en gestión de proyectos

El Trabajo Final de Maestría propone un proceso destinado a asegurar la mantenibilidad del software empleado en una empresa de servicios, con el fin de garantizar una respuesta rápida y flexible que satisfaga las necesidades de los usuarios.

Dado que las PyMEs deben responder de manera eficiente y efectiva a los diversos perfiles de usuarios que interactúan con el sistema, se ha seleccionado la metodología ágil scrum para gestión de proyectos.

La adaptación de la agilidad y la documentación representa un desafío común en el desarrollo de software y otros proyectos ágiles. Aunque las metodologías ágiles, como Scrum, priorizan la entrega rápida y continua de software funcional sobre la documentación extensa, reconocen la importancia crítica de una documentación adecuada para el éxito a largo plazo del proyecto.

Uno de los principios fundamentales del manifiesto Ágil es “Software funcional sobre documentación exhaustiva” [8]. La documentación se produce de manera iterativa y en consonancias con el ciclo de vida del desarrollo de software lo que permite una respuesta rápida ante los cambios. La gestión ágil de la documentación se integra con el proyecto, evolucionando en sincronía y creando un entorno propicio para la adaptabilidad y el aprendizaje continuo.

En resumen, las metodologías ágiles fomentan una documentación flexible, adaptable y que evoluciona junto con el desarrollo del software. El término “ágil” en el desarrollo de software surgió en febrero de 2001, durante una reunión en Utah, EE.UU., con el objetivo de proporcionar una alternativa que permita a los equipos desarrollar software rápidamente y adaptarse a los cambios que puedan surgir durante el proyecto.

El desarrollo de software ágil se centra en metodologías incrementales que fortalecen los procesos de comunicación y fomentan la colaboración entre equipos multidisciplinarios [9]. Los enfoques ágiles están alineados con los principios y valores declarados en el manifiesto ágil, enfatizan la responsabilidad individual durante el desarrollo y la actitud positiva hacia los cambios en los requerimientos. Uno de los enfoques ágiles más ampliamente aceptados y utilizados en la actualidad es Scrum, que se define como un marco de trabajo que permite abordar problemas complejos adaptativos y entregar productos de máximo valor de manera productiva y creativa [10].

Al implementar metodologías ágiles en PyMES, es fundamental superar desafíos como la falta de compresión y capacitación en la metodología, así como el compromiso limitado de los usuarios. Se recomienda que las organizaciones y equipos de trabajo se familiaricen por completo con la metodología ágil, realizando las modificaciones necesarias para adaptarse a las características específicas del entorno [11]. Esto facilita la mantenibilidad del software, permitiendo modificaciones, mejoras y adaptaciones según sea necesario [12].

Scrum, elegido como metodología en el Trabajo Final de Maestría, dado que es una adaptación de un estudio sobre nuevas prácticas de producción al proceso de desarrollo de software [11]. Este marco de trabajo proporciona reglas y tareas específicas donde cada miembro del equipo, quienes deben producir entregables en cada una de las iteraciones (Sprint). A diario, el equipo lleva a cabo reuniones donde cada miembro inspecciona el trabajo de los demás, permitiendo realizar las adaptaciones necesarias. Además, se comunican los impedimentos encontrados y se actualiza el estado de la lista de tareas del proyecto de software, asegurando así su correcta elaboración [15] [13]. Esto da como resultado que Scrum sea un enfoque liviano, muy fácil de aprender, pero muy difícil de dominar [17]. El marco de trabajo Scrum está conformado principalmente por los equipos y sus roles, los eventos, los artefactos y sus reglas asociadas. Cada componente dentro del marco de trabajo sirve para un propósito específico y es esencial para el éxito de Scrum y para su uso [19].

La guía Scrum proporciona una descripción detallada del marco de trabajo, el cual comprende eventos principales como el: *Sprint, Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective*. Además, define los roles fundamentales del *Scrum Master, Product Owner y Development Team*, junto con los *key artifacts* como el *Product Backlog, Sprint Backlog e Increase.* Scrum fomenta la formación de equipos autoorganizados, promoviendo la co-ubicación de todos los miembros del equipo y facilitando la comunicación directa entre todas las disciplinas involucradas en el proyecto.

Este enfoque que proporciona una serie de reglas y tareas específicas que deben llevarse a cabo en cada iteración de un proyecto de software para garantizar su correcta ejecución. Su simplicidad radica en reglas claras que minimizan la subjetividad. Sin embargo, esta misma rigidez puede limitar la capacidad de adaptación en ciertos contextos [10].

Dentro del marco de Scrum, una fase esencial para el mantenimiento del software es la Sprint Retrospective (Retrospectiva del Sprint). Esta etapa se lleva a cabo al finalizar cada iteración o Sprint, y su objetivo es que el equipo reflexione sobre los aspectos que funcionaron bien, aquellos que no lo hicieron tanto, y qué mejoras se pueden implementar en el siguiente Sprint.

En el contexto del mantenimiento de software, la retrospectiva es fundamental, ya que permite identificar áreas de mejora tanto en el proceso de desarrollo como en la calidad del producto. Durante esta fase, el equipo evalúa la eficacia de las prácticas utilizadas, la calidad del código desarrollado, y aborda problemas relacionados con la comunicación o la colaboración. En este sentido, resulta especialmente relevante revisar cómo se implementaron las metodologías teóricas ISO/IEC 12207, Scrum y Mantelasoft y qué ajustes son necesarios para maximizar su efectividad.

Profundizar en la Sprint Retrospective permite identificar patrones de errores recurrentes, debilidades en la infraestructura de desarrollo o en las herramientas empleadas. A través de esta reflexión, se establecen acciones correctivas para abordar dichos problemas en los próximos Sprints, contribuyendo significativamente a mejorar tanto la calidad del software como la optimización del proceso de desarrollo. Este enfoque asegura que las metodologías teóricas estén alineadas con los resultados obtenidos, fortaleciendo la conexión entre la teoría y la práctica.

A continuación, se detallan las fortalezas y áreas de mejora de cada metodología empleada en el proceso de mantenimiento, basándose en los resultados obtenidos:

- **ISO/IEC 12207**:
    - **Fortalezas**: Esta norma proporcionó una estructura clara y bien definida para cada fase del ciclo de vida del mantenimiento, lo que facilitó la trazabilidad y garantizó la calidad de las intervenciones realizadas.
    - **Áreas de mejora**: En algunas etapas del proceso, la documentación exigida por ISO/IEC 12207 generó una carga administrativa mayor de lo previsto. Para mejorar la eficiencia, sería recomendable ajustar el nivel de detalle en ciertas secciones, sin comprometer la rigurosidad del seguimiento.

- **Scrum**:
    - **Fortalezas**: La metodología ágil Scrum, con su organización en Sprints, permitió una rápida adaptación a los cambios y una priorización efectiva de las tareas críticas. Esto mejoró considerablemente la flexibilidad y la capacidad de respuesta en la gestión del mantenimiento.
    - **Áreas de mejora**: En ocasiones, las reuniones diarias (Daily Scrum) se extendían más de lo necesario, lo que afectaba la productividad. Un ajuste sugerido sería limitar su duración estrictamente a 15 minutos, enfocándose únicamente en los aspectos más relevantes para mantener el flujo eficiente.
- **Mantelasoft**:
    - **Fortalezas**: Las plantillas de documentación ofrecidas por Mantelasoft proporcionaron una estructura clara y organizada para registrar todos los cambios, asegurando que cada intervención fuera trazable y comprensible para futuros análisis.
    - **Áreas de mejora**: La rigidez de las plantillas dificultó en algunos casos la adaptación a cambios rápidos en el software. Un ajuste recomendado sería flexibilizar el formato para permitir modificaciones más ágiles, manteniendo al mismo tiempo la coherencia documental.

Durante la revisión, se concluyó que la implementación de ISO/IEC 12207 y Scrum permitió cumplir con los principios teóricos expuestos en el marco conceptual. Sin embargo, se identificaron áreas donde la teoría podría ajustarse para reflejar mejor la realidad práctica de la PyME, como la necesidad de equilibrar el nivel de detalle en la documentación.

A partir del análisis realizado en la revisión, se definieron los siguientes ajustes

- Reducir la carga administrativa asociada a la documentación de **ISO/IEC 12207** en ciertas fases del mantenimiento, optimizando el uso del tiempo.
- Mantener las reuniones de **Daily Scrum** estrictamente dentro de los 15 minutos, para mejorar la eficiencia en la coordinación.
- Flexibilizar las plantillas de **Mantelasoft** para permitir ajustes rápidos en la documentación sin perder trazabilidad.

Estos ajustes se implementarán en el próximo ciclo de mantenimiento correctivo, con el objetivo de asegurar que las metodologías teóricas estén completamente alineadas con las soluciones propuestas y que el proceso de mantenimiento continúe mejorando su eficiencia y efectividad.

La fase de revisión, al permitir un análisis crítico sobre el uso de las metodologías y la implementación de los ajustes necesarios, asegura que la propuesta se mantenga sólida y adaptable. Esto fortalece la conexión entre la teoría y la práctica, consolidando el plan de mantenimiento y garantizando su sostenibilidad a largo plazo.

2.4 Trabajos relacionados

A continuación, se mencionan algunos antecedentes relacionados con la propuesta de TFM, contemplando el mantenimiento del software, que abarcan desde la identificación y corrección de errores hasta la actualización y mejora continua del software.

En referencia a la implementación del mantenimiento en las organizaciones, se menciona a [4] quienes relevan el abordaje desde la Ingeniería del Software, en particular, el tratamiento de sus temas desde la guía SWEBOK, y cómo son representadas algunas de estas áreas en los Trabajos Finales de Graduación de estudiantes de la carrera Licenciatura en Sistemas de Información en el ciclo lectivo 2016.

En [14] se establece que ejecutar un plan de mantenimiento de software a medida, mejorará la calidad del producto final, ya que se combinan herramientas de software, métodos y técnicas.

Otro abordaje [15], describe el proceso para administrar y ejecutar actividades de mantenimiento del software, que puede complicarse a medida que los sistemas crecen o se producen disminución o cambios de personal. En este sentido, se coincide con que el "mantenimiento del software lleva la mayor porción de la vida de un sistema" [13]. En particular el mantenimiento permite la actividad de modificar el software conservando su integridad [16].

En [17] se expone una vista integral de diferentes atributos de mantenibilidad obtenidos a partir de la literatura y propone una clasificación de los mismos considerando: "(i) las sub-características de mantenibilidad de ISO/IEC 25010 sobre las que influye, y (ii) el flujo de trabajo del desarrollo de software de RUP (Rational Unified Process) en los que se presenta".

En una revisión sistemática de mejora de procesos software (SPI) en micro, pequeñas y medianas empresas, para conocer los esfuerzos llevados a cabo en PyMEs desarrolladoras de software relacionados con la mejora de sus procesos [17]. En el artículo se da conocer las mejoras que se obtuvieron en las empresas con respecto a procesos software, para contar con una visión completa de la situación actual de la empresa.

En [15] se expone la evaluación y mejora de los estándares de mantenimiento de software y la introducción de un modelo de madurez propuesto para las actividades diarias de

mantenimiento de software denominado Modelo de madurez de mantenimiento de software (SMmm). Éste trata las actividades únicas de mantenimiento de software manteniendo una estructura similar a al modelo de madurez CMMI. Este modelo se basa en la experiencia de los practicantes, los estándares internacionales y la literatura sobre mantenimiento de software, presentándose el propósito, alcance, fundamento, arquitectura del modelo y su validación inicial [9].

En [9] se propone la guía de mantenimiento de software Mantelasoft. La misma permite a las empresas evaluar el estado del mantenimiento de sus desarrollos sin incrementar la demanda del tiempo requerido para ajustarla a sus necesidades. Se basa en la metodología Mantema y la norma ISO/IEC 12207. La guía contribuyó a la mejora del proceso de mantenimiento de software en Mipymes que se encuentran en la ciudad de Pereira.

En [16] se propone una Aproximación a la Mejora de la Metodología Ágil Mantema para la Adopción del Modelo de Calidad ISO/IEC 15504-ISO/IEC 12207 en PyMEs. El artículo presenta una mejora realizada a la metodología Ágil Mantema. Dicha mejora se realizó comparándola con las normas ISO/IEC 15504 y ISO/IEC 12207 y de este análisis se determinaron los puntos débiles de Ágil Mantema y se describieron las actividades y tareas propuestas. Adicionalmente, se implementó la nueva propuesta en una PyME de México, dando resultados favorables [9].

El INTI desarrolló un software para reducir costos y mejorar la gestión en PyMEs [18]. Este software de Gestión Integral de Mantenimiento (GIM) es una herramienta informática pensada para digitalizar los procesos y mejorar la gestión de las empresas.

Según los datos relevados, la PyME logró incrementar el tiempo medio entre falla de uno de los principales equipos en un 93%; reducir el tiempo de máquina parada en un 60% y mejorar su confiabilidad en un 94%. A su vez, se pueden mencionar mejoras en la organización de las tareas de mantenimiento y, en función de que se cuenta con mayor tiempo disponible, en la calidad de los trabajos, la motivación del personal, la digitalización y ordenamiento de la información, entre otros aspectos. En síntesis, las mejoras impactaron en la línea de producción optimizando el tiempo de respuesta, la imagen hacia terceros y disminuyendo costos [18].

En [19] se trata el desarrollo de un plan de gestión de mantenimiento de software para el departamento de sistemas de la universidad politécnica salesiana basado en la norma ISO/IEC 14764:2006, en este trabajo se realiza un plan de mantenimiento que será usada en aquellos productos que se mantendrán un tiempo lo más largo posible dentro de la

Universidad. La aplicación proporcionará una quía para realizar el mantenimiento a las aplicaciones desarrollada. Proponen aplicar el mantenimiento a programas de ordenador, código, datos y documentación de administración y más adelante en el tiempo a productos de software que sean creados durante el desarrollo de un nuevo software.

En [9] se procesó para la evaluación y mantenimiento de software en la PyMES y los departamentos de desarrollo de software en la ciudad de Pereira (Mantelasoft), se propuso una guía denominada "Mantelasoft", enfocada en el proceso de evolución y mantenimiento de software para las MiPymes y departamentos de desarrollo de software. Esta guía corresponde a una combinación entre el proceso Ágil Mantema y los procesos de apoyo planteados por la norma ISO/IEC 12207:2008.

En *Generando productos software mantenibles desde el proceso de desarrollo: El modelo de referencia MANTu*S [1], se propuso un modelo de referencia para la inclusión de subcaracterísticas de mantenibilidad al producto software durante el proceso de desarrollo, este fue elaborado a partir de: (i) la identificación de los atributos de software que influyen en la mantenibilidad del producto, (ii) la clasificación de los atributos identificados de acuerdo a las subcaracterísticas de mantenibilidad y a los procesos en los que se presentan, (iii) el análisis de como potenciar cada subcaracterística, (iv) la estructuración del modelo de referencia y (v) la evaluación del modelo de referencia realizada por medio de un caso de estudio. Se observa que la mantenibilidad de software es una característica de calidad que se debe considerar desde etapas tempranas del ciclo de vida del producto y durante todo el proceso de desarrollo de software con el fin de disminuir los costos de desarrollo y obtener un producto altamente mantenible, es por esto que las prácticas base definidas para los procesos del modelo de referencia son sencillas, esto con el fin de no aumentar el esfuerzo durante el desarrollo; sin embargo, es importante resaltar que estas deben considerarse durante el proceso de desarrollo y no en etapas posteriores del ciclo de vida del producto. De la aplicación del estudio de caso se puede concluir que la inclusión de las prácticas base del modelo de referencia al proceso de construcción permite potenciar la mantenibilidad del producto software, ya que la capacidad para ser analizado y modificado del producto aumenta considerablemente. Los resultados obtenidos en el estudio de caso aplicado demuestran que la utilización de prácticas base del proceso de construcción del modelo de referencia MANTuS permite potenciar la mantenibilidad del producto software.

En [16] se enfoca en la Educación Superior ofrece un vasto dominio para el diseño y desarrollo de soluciones tecnologías orientadas a la gestión de la información. Se presenta una experiencia fundada en conceptos teóricos de la evolución del software, en particular su

mantenimiento, actividad que permite modificar el mismo conservando su integridad. la propuesta se resume una práctica atinente a la evolución del software centrada en el mantenimiento perfectivo aplicado en un sistema gestor de contenidos de código abierto.
En [20] el artículo se centra en la implementación de un repositorio de documentos digitales administrativos, se seleccionaron cuestionarios y encuestas como técnicas e instrumentos de recolección de información. El objetivo que orientó la creación de estos instrumentos se basó en la necesidad de comprender los documentos físicos y digitales, su preservación, así como los recursos disponibles que se incorporan en este proceso de relevancia administrativa y financiera mensual.

2.5 Otros recursos para la definición del proceso

En esta sección se resumen otros recursos seleccionados para definir el proceso y sintetizan otros recursos seleccionados para la definición del proceso propuesto en el marco del Trabajo Final de Maestría.

2.5.1. Matriz FODA

La Matriz FODA implica la evaluación de las fortalezas, debilidades, oportunidades y amenazas de una empresa. Su propósito principal es identificar estrategias para capitalizar las oportunidades externas, mitigar las amenazas, fortalecer y proteger los puntos fuertes de la compañía, y abordar las debilidades [29] [30] [31], resumidas de manera concisa en el Anexo II. Por lo tanto, en el contexto del TFM, el objetivo de llevar a cabo un análisis FODA es crear, reforzar o perfeccionar un modelo de negocio específico de la empresa, especialmente en el área de desarrollo, de manera que optimice, adapte o integre de manera más efectiva sus recursos y capacidades con las demandas del entorno en el que opera.

2.5.2 Protección de los datos personales

La realización de la encuesta se llevó a cabo teniendo en cuenta lo establecido en la Ley 25.326 [21], sancionada el 4 de octubre de 2000 y parcialmente promulgada el 30 octubre de 2000
Según el Artículo 1° de esta ley [21], su objetivo es la protección integral de los datos personales registrados en archivos, registros, bancos de datos u otros medios técnicos de procesamiento de datos, ya sean públicos o privados, destinados a proporcionar información, con el fin de salvaguardar el derecho al honor y a la intimidad de las personas. Además,

busca garantizar el acceso a la información que se registre sobre ellas, en conformidad con lo establecido en el artículo 43, párrafo tercero de la Constitución Nacional.

Capítulo 3

Marco metodológico

Para alcanzar los objetivos propuestos en este TFM, se ha diseñado un proceso que permitirá a una PyME de servicios mantener sus sistemas de manera eficaz y eficiente, ajustándose tanto a sus necesidades específicas como a los requisitos particulares que puedan surgir. Esta propuesta se fundamenta en los siguientes aspectos clave:

- **Implementación en el corto plazo:** Dada la importancia crítica del sistema para la operación de la PyME, se ha establecido un plazo máximo de 60 días para la implementación del mantenimiento correctivo. Este plazo garantiza una respuesta rápida y eficiente a las demandas del sistema, minimizando el impacto en las actividades diarias de la empresa.
- **Resolución de problemas críticos:** Se han identificado una serie de problemas que afectan el funcionamiento del sistema, tales como la imposibilidad de acceso, fallos en la conexión a ciertas bases de datos y caídas ocasionadas por el alto tráfico de internet. La resolución de estos problemas será prioritaria para asegurar la estabilidad y continuidad del sistema, evitando interrupciones en el servicio.
- **Búsqueda y adaptación de soluciones a medida:** A medida que se presentan problemas específicos, se han desarrollado soluciones personalizadas con el objetivo de mejorar la experiencia del usuario. Este enfoque garantiza que las soluciones no solo resuelvan los problemas existentes, sino que también estén adaptadas a las características únicas del sistema y las necesidades de los usuarios, permitiendo una mayor flexibilidad y efectividad en la operación del software.

Además, se consideraron las siguientes etapas, las cuales se detallan a continuación:

Fase 1. Revisión sistemática de la literatura.

Fase 2. Recopilación de información del sector TI, específicamente del área correspondiente a la PyME objeto de estudio en la propuesta.

Fase 3. Profundización en el conocimiento de metodologías, procesos y normas o estándares que respalden la propuesta

Fase 3.1 Selección del estándar, en este caso se optó por ISO/IEC 12207

Fase 3.2 Elección y adaptación de la metodología ágil como scrum y seguimiento de directrices establecidas por Mantelasoft

Fase 3.3 Selección de la metodología de mantenimiento de software, en este caso el mantenimiento correctivo.

Fase 4. Elaboración del proceso

Fase 5. Validación del proceso

Fase 6. Análisis de los resultados

Fase 7. Redacción del informe final

Cada una de estas fases se detalla en la Fig. 1, han sido fundamentales para el desarrollo y la exitosa conclusión del TFM.

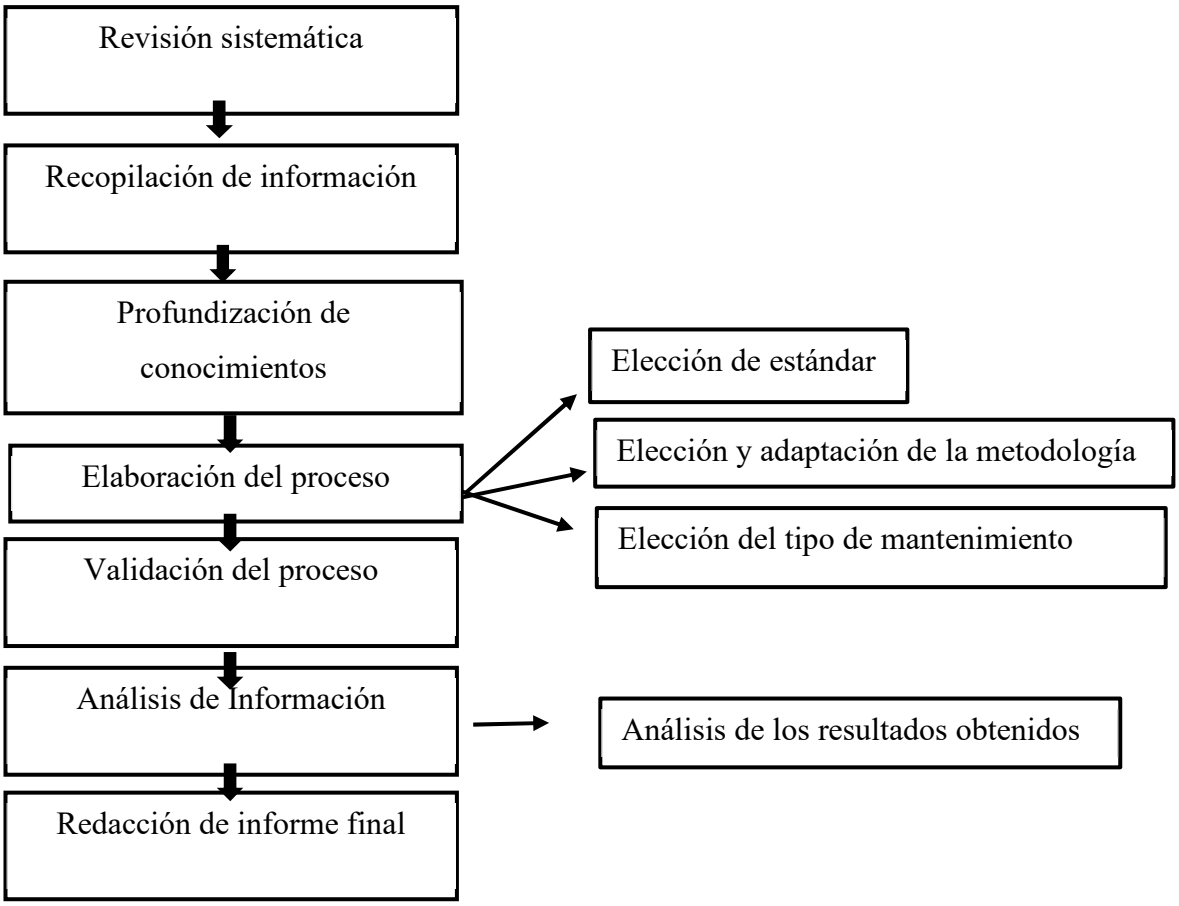


Fig.1 Etapas de elaboración del TFM

Fase 1. Revisión sistemática de la literatura

Para definir el alcance del TFM, se llevó a cabo una Revisión Sistemática de la Literatura (RSL). A lo largo de este estudio, se enriqueció la bibliografía existente. La RSL proporciona un medio confiable, riguroso y auditable para evaluar e interpretar las investigaciones disponibles sobre un tema específico, en este caso, el proceso de mantenimiento del software. Se realizaron búsquedas y selecciones bibliográficas con el objetivo de recopilar información relevante.

Durante más de una década, las RSL han sido ampliamente utilizadas en el campo de la Ingeniería de Software para proporcionar resúmenes significativos de evidencia sobre una variedad de temas [22].

La búsqueda de artículos se limitó al período comprendido entre los años 2016 y 2022. La información recopilada se utilizó para la elaboración de Capitulo 1 y se incluye como Anexo 1.

Fase 2. Recopilación de información del área de la PyME objeto de la propuesta.

Para planear y tomar decisiones con respecto a la realización del proceso de mantenimiento del software, es necesario conocer el estado actual del sistema que se tomará como caso de estudio, se procedió al diseño e implementación de encuestas ver Anexo 3. Con la finalidad de delimitar el objeto de estudios con fines de elaboración de la propuesta y su posterior validación, el proceso de mantenimiento propuesto se centra en el Sistema Control de Tráfico.

Las preguntas incluidas para la elaboración de la encuesta sobre el mantenimiento de software fueron adaptadas siguiendo [9]:

- Generalidades del mantenimiento.
- Planificación del mantenimiento.
- Estimación de costos.
- Proceso de mantenimiento.
- Personal de mantenimiento.
- Técnicas y herramientas de mantenimiento.
- Evolución o mejora del producto.

Una vez obtenido el resultado de la encuesta se procedió a analizar la información recolectada y así a identificar qué tipo de mantenimiento ayudará a mejorar el funcionamiento del sistema.

Siguiendo lo expuesto en [9], se diseñó una encuesta entregada al personal administrativo que maneja el sistema, para determinar el funcionamiento del mismo, la información disponible y documentación (física o digital) que se reciben y producen. La misma contaba con preguntas abiertas para recabar las expresiones de los participantes y con preguntas cerradas para seleccionar opciones de una lista de posibles respuestas. Ver Anexo 4

Se realizó un análisis de los datos recolectados mediante encuestas, utilizando herramientas TIC para su procesamiento. A partir de esta información, se elaboró una matriz FODA

(Fortalezas, Oportunidades, Debilidades y Amenazas), cuyo objetivo fue analizar la situación actual de la empresa. Esta herramienta permitió identificar áreas clave para facilitar la toma de decisiones estratégicas, proporcionando una visión clara sobre los aspectos que requieren atención y aquellos que ofrecen potencial para mejorar el rendimiento de la empresa.

El enfoque mencionado se ha aplicado en el caso de estudio, esto llevo a los siguientes resultados

- **Fortalezas:**
  - **Experiencia del equipo:** se evaluó la experiencia y habilidades técnicas del equipo de mantenimiento de software.
  - **Herramientas y tecnologías:** Se Analizó las herramientas y tecnologías utilizadas en el proceso de mantenimiento y su efectividad.
  - **Historial de éxito:** se revisó proyectos anteriores de mantenimiento para identificar éxitos y mejores prácticas.
- **Oportunidades:**
  - **Actualización tecnológica:** Se identificó oportunidades para adoptar nuevas tecnologías que mejoren la eficiencia del sistema.
- **Debilidades:**
  - **Falta de documentación:** Falta de documentación o información detallada sobre el software y sus componentes.
  - **Falta de recursos:** El equipo de mantenimiento no cuenta con los recursos adecuados en términos de personal, tiempo y presupuesto.
- **Amenazas:**
  - **Avance tecnológico rápido:** El rápido avance tecnológico puede afectar la relevancia de las habilidades actuales de los sistemas implementados.

Fase 3. Profundización de conocimientos en torno a metodologías, procesos y normas o estándares que sustentan la propuesta

El diseño de un proceso eficiente de mantenimiento correctivo de software requiere una selección estratégica de metodologías y estándares que aseguren tanto la efectividad operativa como la adaptabilidad a las características específicas de las PyMEs. En este contexto, la elección metodológica se centra en una combinación de enfoques ágiles, como Scrum, junto con el uso de la guía Mantelasoft y el marco normativo definido por ISO/IEC 12207.

1. **Elección de metodología**

   Para enfrentar el desafío del mantenimiento correctivo en el software, se ha optado por una cuidadosa selección y adaptación de metodología ágiles. Scrum se destaca por su capacidad de gestionar proyectos con alta flexibilidad y su enfoque en la mejora continua a través de ciclos iterativos (Sprints). Sin embargo, dada la necesidad de un marco más estructurado y específico para el mantenimiento de software, se ha complementado con Mantelafost, una guía diseñada específicamente para optimizar el ciclo de vida del mantenimiento de software en PyMEs. Esta elección también se fundamenta en los principios establecidos por la norma ISO/IEC 12207, que proporciona un marco integral para gestionar el ciclo de vida del software, desde su concepción y especificación inicial hasta su operación y mantenimiento.

   La combinación de Scrum y Mantelasoft permite un equilibrio entre agilidad y estructura, garantizando que las actividades de mantenimiento sean dinámicas y adaptativas, pero al mismo tiempo, guiadas por un proceso robusto y estandarizado que minimice errores y optimice los tiempos de respuesta.

2. **Elección del tipo de mantenimiento de software**

   Es este caso de estudio, se optó por el mantenimiento correctivo como enfoque principal, ya que es el tipo de mantenimiento que mejor se adapta a las necesidades operativas de las PyMEs del sector transporte. El mantenimiento correctivo se enfoca en la identificación y corrección de errores que surgen durante la operación del software, asegurando que las aplicaciones sigan funcionando de manera óptima en entornos cambiantes y dinámicos.

3. **Adaptabilidad a PyMEs**

   Una de las principales razones por las cuales Mantelasoft se considera la metodologia más adecuada para este proyecto es su flexibilidad y simplicidad en la gestión del ciclo de vida del software. Dado que las PyMEs suelen operar como recursos limitados, tanto en términos de personal como de infraestructura tecnológica, Mantelasoft permite una implementación eficaz sin requerir grandes inversiones ni equipos altamente especializados. Esta característica es crucial, ya que ofrece a las PyMEs una solución ajustada a sus necesidades, optimizando la relación entre recursos y resultados.

4. **Flexibilidad en entornos dinámicos**

   Otra ventaja clave de Mantelasoft es su capacidad de adaptarse rápidamente a los cambios en el entorno operativo. Las PyMEs del sector transporte operan en un mercado dinámico, con requerimientos de software que pueden variar según las circunstancias o los cambios regulatorios. Mantelasoft permite realizar ajustes rápidos en las actividades de mantenimiento, facilitando una respuesta ágil ante nuevas exigencias o la aparición de fallas imprevistas. Esta capacidad de adaptación minimiza las interrupciones en el servicio, mejorando la continuidad operativa.

5. **Simplicidad de la implementación**

   Finalmente, la simplicidad en la implementación es una de las grandes fortalezas de Mantelasoft. Esta guía ha sido diseñada para ser fácil de aplicar en empresas con equipos de IT pequeños, ya que su enfoque se centra en procesos simplificados y manejables, que no requieren una gran inversión en tiempo o recursos. Esto lo convierte en una opción ideal para PyMEs que buscan mejorar su gestión de mantenimiento sin comprometer otros aspectos de su operación.

Fase 4. Elaboración del proceso: esta se enfoca en definir y planificar los procesos necesarios para llevar a cabo las actividades de mantenimiento de manera efectiva, asegurando que se sigan prácticas y estándares adecuados durante el proceso.

Fase 5. Validación del proceso: esta fase se centra en garantizar que los cambios realizados en el software mantenido sean apropiados y efectivos. Esto ayuda a mantener la calidad y la integridad del software a lo largo del tiempo.

Fase 6. Análisis de los resultados: es una etapa crítica que se centra en evaluar y comprender los resultados obtenidos durante las actividades de mantenimiento. Proporciona información valiosa que puede utilizarse para mejorar continuamente la calidad y eficacia del software y del proceso de mantenimiento

Fase 7. Redacción del informe final: esta fase implica documentar y comunicar los hallazgos, resultados y lecciones aprendidas durante todo el proceso de mantenimiento del software. Proporciona una visión completa y transparente del trabajo realizado y sirve como base para la mejora continua del proceso de mantenimiento el software.

A continuación, se presenta una relación clara entre los conceptos clave del marco teórico y su aplicación directa en las acciones y metodologías utilizadas en la propuesta de mantenimiento de software. Este enfoque asegura que las teorías seleccionadas no solo guíen

la implementación, sino que validen su eficacia en un entorno de PyMEs del sector de transporte:

Metodologías aplicadas en el mantenimiento de software

- ISO/IEC 12207: Este estándar internacional, que abarca todo el ciclo de vida del software, se aplica en todas las fases del mantenimiento, con especial énfasis en la aplicación y documentación de cada intervención en el software.
    - Vinculación: Durante cada fase del mantenimiento, se siguen estrictamente los procesos y actividades definidos por ISO/IEC 12207, desde la identificación del problema hasta su resolución y la documentación correspondiente. Esto garantiza una estructura sólida y trazable en cada acción realizada.
- Scrum: El marco ágil se utiliza para gestionar el mantenimiento a través de ciclos iterativos conocidos como Sprints.
    - Vinculación: Las actividades de mantenimiento se planifican y ejecutan en Sprints, con roles claramente definidos como el Scrum Master y el Product Owner. Estos roles aseguran la correcta priorización de las tareas y la entrega continua de mejoras, manteniendo la agilidad y flexibilidad del proceso.
- Mantelasoft: Este enfoque proporciona una guía sólida para la documentación y trazabilidad de todo el proceso de mantenimiento.
    - Vinculación: Se emplea Mantelasoft para mantener actualizada las plantillas de documentación, asegurando que cualquier intervención quede registrada de manera adecuada, lo que facilita futuras intervenciones y mejora la trazabilidad del mantenimiento.

Aplicación en la PyME

1. ISO/IEC 12207 en la planificación del mantenimiento correctivo:
    a. Ante la detección de la falla en el módulo diagrama de servicio, se sigue el proceso detallado por IOS/IEC 12207 para analizar el impacto y documentar cada paso de la corrección.
    b. Se documenta el análisis del error, la corrección aplicada y las pruebas de validación, asegurando la trazabilidad del proceso y facilitando su repetición en futuros incidentes.
2. Scrum en el proceso de mantenimiento:

a. Un equipo de mantenimiento organiza sus tareas en Sprints para abordar errores en el sistema de control de tráfico.
b. Durante cada Sprint, se priorizan los errores más críticos. Al final de cada ciclo, se realiza una Sprint Review, donde el Product Owner valida las mejoras implementadas, asegurando la entrega continua de valor.

3. Mantelasoft en la documentación del proceso:
    a. Después de cada Sprint, se actualizan las plantillas de Mantelasoft para reflejar los cambios realizados en el software.
    b. El equipo documenta las modificaciones en las configuraciones del sistema, lo que garantiza que todos los miembros del equipo puedan comprender y referenciar los cambios de manera clara y efectiva.

La *fig. 2* ilustra la interacción de las metodologías ISO/IEC 12207, Scrum y Mantelasoft en el proceso de mantenimiento de software, destacando los roles clave de Scrum (Scrum Master, Product Owner), la estructura de gestión conforme a ISO/IEC 12207 y la documentación organizada mediante Mantelasoft.

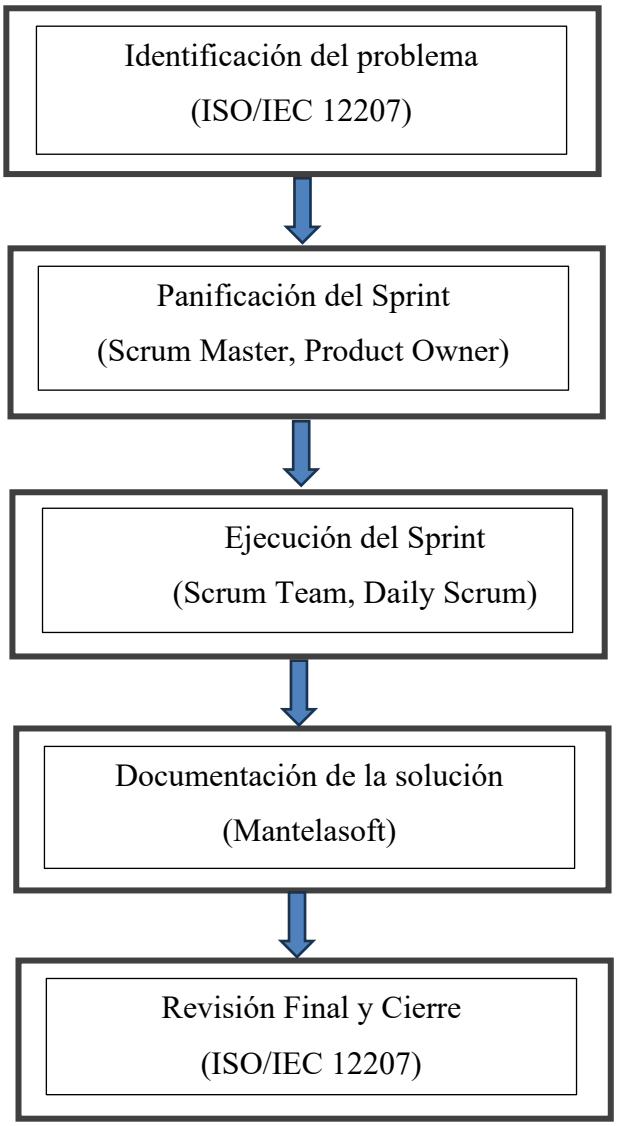


Fig. 2 Iteración de las metodologías

Capítulo 4

Solución propuesta

4.1 Introducción

En este capítulo se expone la Solución o resultados relativos al estado actual del objeto de estudio, que consiste en el desarrollo de un proceso para el sistema de una PyME denominado como Sistema Control de Tráfico. Tras realizar el relevamiento de datos y siguiendo los principios del mantenimiento del software, se ha determinado que la opción más adecuada para el caso de estudio es la implementación de un proceso de mantenimiento correctivo de software, así como el diseño de una plantilla para documentar dicho mantenimiento. En resumen, la propuesta se enfoca en abordar de manera eficaz los problemas y fallos identificados durante la utilización del software.

> "El mantenimiento se realiza cuando un software presenta fallas, entonces se procede a reparar y restablecer la misma para solucionar el daño para que así el software siga funcionando adecuadamente. Este mantenimiento se debe realizar en el menor tiempo posible, apenas la falla esa identificada para evitar que se incremente. Se debe determinar la causa y solucionarla" [23].

Para simplificar el desarrollo del proceso de mantenimiento, se ha decidido utilizar una adaptación de Mantelasoft como la herramienta principal de gestión de los sistemas existentes. Esta herramienta se emplea en la fase de planificación del mantenimiento y en el proceso mismo del mantenimiento. Además, se han utilizado Sprint para definir las actividades de los equipos de trabajos. Cabe destacar que esta herramienta se alinea con el estándar establecido por la norma del modelo de procesos de mantenimiento de software ISO/IEC 12207 [24].

La propuesta diseñada se caracteriza por su enfoque centrado en la implementación rápida y efectiva, priorizando la resolución de problemas puntuales dentro de un plazo máximo de 60 días. Este enfoque se basa en abordar problemas específicos que afectan la operatoria del sistema. Además, se destaca la búsqueda activa y adaptación de soluciones específicas a medida que surgen nuevos inconvenientes, con el objetivo de mejorar la experiencia del usuario al operar con el sistema. Este enfoque proactivo y orientado a la acción garantiza una respuesta ágil y eficiente a los desafíos que puedan surgir durante el uso del sistema.

En términos generales, el proceso de mantenimiento se concibe como el conjunto de operaciones necesarias para asegurar la integridad y funcionalidad del software [25]. En la

sección 4.2 se presenta la propuesta de mantenimiento de software Sistema Control de Tráfico (SCT), mientras que en la sección 4.3 se detallan las validaciones realizadas.

4.2 Propuesta plan de mantenimiento de software SCT

Se propone un plan de mantenimiento para el **Sistema de Control de Tráfico (SCT)**, que tiene como objetivo asegurar el funcionamiento óptimo del software y su alineación con las necesidades organizacionales. Este plan define las acciones necesarias para mantener el sistema operativo, documentar los procesos y gestionar los cambios de manera eficiente.

Tras definir los objetivos del plan, se asignan responsabilidades a los miembros del equipo, garantizando que cada uno asuma tareas acordes con sus habilidades y experiencia. Además, se establecen fechas límite para la finalización de las tareas de mantenimiento, lo que permite un seguimiento claro y controlado del progreso.

El equipo de desarrollo trabaja en estrecha colaboración con las partes interesadas para definir los objetivos y el alcance del mantenimiento. Posteriormente, se identifican las tareas necesarias para implementar las modificaciones, incluyendo la programación y pruebas requeridas para garantizar el correcto funcionamiento del sistema.

El proceso comienza con la planificación, en la que se asignan los responsables, se adquiere conocimiento detallado del software a mantener, se preparan los entornos de pruebas y se definen los procesos según las distintas solicitudes de los usuarios. En esta fase, se realiza una evaluación inicial de las solicitudes recibidas, diferenciando entre problemas urgentes y no urgentes.

- **Problemas urgentes**: Estos se gestionan inmediatamente, sin pasar por la fase de planificación. El equipo analiza las causas, ejecuta las correcciones necesarias y realiza las pruebas correspondientes para validar la solución en el menor tiempo posible.
- **Problemas no urgentes**: Estos son planificables y se incluyen en el flujo regular del mantenimiento. El equipo realiza un análisis detallado de las solicitudes y selecciona la opción más adecuada para su resolución, utilizando la fase de planificación para priorizar las tareas y asignar recursos eficientemente.

En este sentido, la propuesta combina las mejores prácticas de Mantelasoft, enfocándose en la gestión integral del ciclo de vida del software, y de Scrum, garantizando una respuesta ágil y una planificación efectiva. Además, la norma ISO/IEC 12207 proporciona un marco

que asegura la calidad y trazabilidad de todas las actividades relacionadas con el mantenimiento del software.

“El proceso de mantenimiento contiene las actividades y tareas realizadas por el mantenedor. Este proceso se activa cuando en el software surge modificaciones en el código y la documentación asociada, debido a un problema o a la necesidad de mejora o adaptación”. [19]

Para llevar a cabo el mantenimiento del software, es necesarios seguir una serie de pasos. En primer lugar, se realizar una evaluación del software para identificar los problemas y áreas de mejora. Lugo, se planifica el mantenimiento, definiendo objetivos, tareas, recursos y un cronograma para las actividades de mantenimiento. La *Fig. 3* resume el ciclo de mantenimiento.

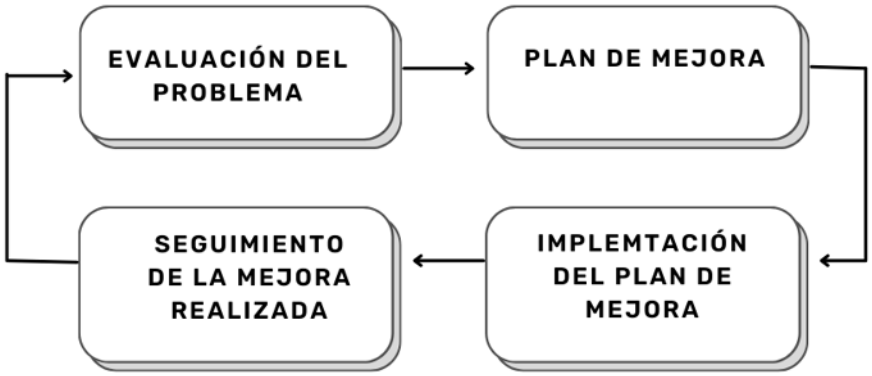


Fig.3 Ciclo para el mantenimiento.

El diagrama de flujo se empleó como una herramienta que facilitó la presentación de un proceso para el mantenimiento correctivo del software. Su utilización radica en la representación visual del flujo de actividades dentro de dicho proceso, lo cual permite identificar las tareas necesarias y la secuencia en la que deben llevarse a cabo. Una vez planificado el mantenimiento, se procede a ejecutar las tareas correspondientes, las cuales pueden abarcar desde la corrección de errores y la mejora del rendimiento, hasta la implementación de nuevas funcionalidades *Fig. 4*. En la misma se puede ver cómo los errores urgentes se gestionan de inmediato, sin pasar por la fase de planificación, mientras que los problemas no urgentes siguen el proceso habitual.

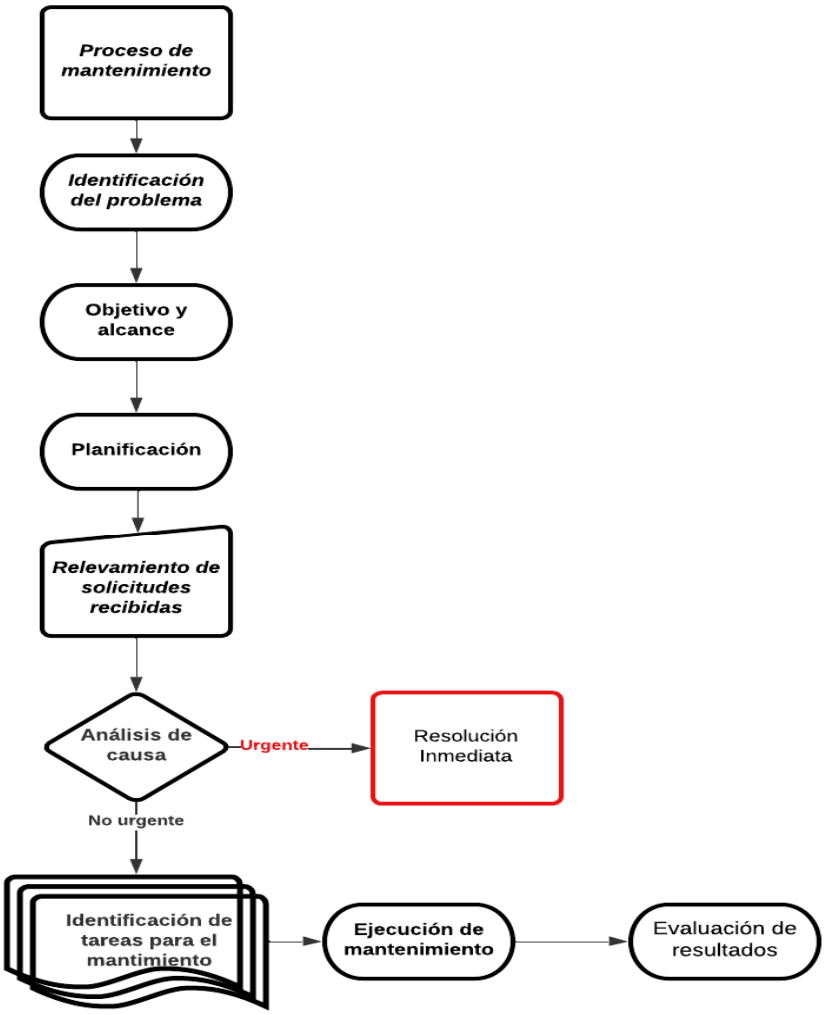


Fig. 4 Propuesta plan de mantenimiento de software SCT

En la definición de la propuesta se utilizó mantenimiento correctivo urgente con elementos claves de Mantelasoft, los mismos son: evaluación inicial del sistema, planificación del mantenimiento, respaldo de datos y configuraciones, actualización del software, optimización del rendimiento, prueba y verificación, capacitación y documentación con la finalidad de lograr una solución eficiente y eficaz [9]. En el punto 4.3 se propone utilizar plantillas basados en la norma ISO/IEC 12207 con la finalidad de garantizar una gestión adecuada del mantenimiento del sistema.

Para la implementación del plan de mantenimiento, se examinaron los principios y técnicas propuestos por Mantelasoft, los cuales se alinean con los requisitos establecidos en la norma ISO/IEC 12207 en términos de actividades, roles y documentos. Se llevó a cabo un análisis de ambas propuestas, identificando similitudes y diferencias.

Al aplicar los conceptos de actividades, roles y documentos al contexto del mantenimiento de software en una empresa de transporte, se puede ofrecer una contribución valiosa. Por ejemplo, en lo que respecta a las actividades, sería beneficioso considerar una planificación de mantenimiento diseñada específicamente para abordar las necesidades del sector, con las actualizaciones de software para el o los sistemas que fueran necesarios. En cuanto a los equipos de trabajo a cada rol, se podrían definir responsabilidades específicas para el personal encargado del mantenimiento, tales como el desarrollador, y el coordinador. En relación a los documentos, sería oportuno desarrollar una documentación detalladas de los procesos de mantenimiento adaptados a las particularidades del sistema, lo que incluía protocolos de seguridad para salvaguardar los datos sensibles de los usuarios y asegurar la integridad de las operaciones.

Para llevar a cabo el proceso, se combinaron los principios de Mantelasoft con los estándares de la norma ISO/IEC 12207, además de incorporar conceptos de Scrum.

En particular, de Scrum, se adaptó la formación de equipos que operaron conforme a una lista de tareas priorizadas y plazos de entrega establecidos, fragmentados en ciclos conocidos como Sprints. Para integrar a los equipos en el diseño de la solución se consideraron

1- **Definir el equipo**: En esta etapa se identifican los diferentes roles dentro del equipo Scrum. Por lo tanto, se propone diseñar un proceso basado en Scrum, destacando roles clave como el Scrum Master, el Product Owner y los desarrolladores. A cada miembro del equipo se le asignan responsabilidades claras en relación con la creación y gestión de la documentación para el mantenimiento correctivo.
2- **Establecer un orden de tareas**: Priorizar las tareas según su importancia y urgencia para el proceso de mantenimiento.
3- **Planificar los Sprints:** Organizar reuniones de planificación de Sprints donde el equipo Scrum selecciona las tareas de la lista para el siguiente Sprint. Establece objetivos claros para cada Sprint, incluyendo la producción de documentación necesaria para el mantenimiento correctivo.
4- **Definir reuniones diarias:** Fomenta el desarrollo interactivo de la documentación, con entregas parciales al final de cada Sprint. Realizar revisiones periódicas con el equipo Scrum y otras partes interesadas para obtener retroalimentación y ajustar la documentación según sea necesario.
5- **Definir la entrega y mejora continua:** Al final de cada Sprint, entregar la documentación completada y revisada como parte del del producto final del

mantenimiento correctivo. Utilizar la retroalimentación recibida durante las revisiones para mejorar continuamente la calidad y relevancia de la documentación.

6- Realizar un seguimiento regular de los resultados para identificar áreas de mejora y optimización en el proceso de elaboración de documentación

Al integrar los equipos en la solución para la elaboración de la documentación del mantenimiento correctivo del software, se puede mejorar la eficiencia, la colaboración y la calidad de la documentación producida, la que a su vez contribuye a lograr un mantenimiento más efectivo y transparente.

A continuación, se describen los pasos llevados a cabo:

1- Se elaboró la visión del proceso de mantenimiento.
2- Se designó al Scrum Master, quien lideraría el equipo de trabajo, y se identificaron los distintos equipos involucrados en el proceso de mantenimiento.
3- Se formaron los equipos de trabajos.
    a. **Sprint 1:** Revisión y Análisis
        i. **Objetivo:** Revisar y analizar los informes de errores y solicitudes de cambio para identificar las correcciones necesarias.
        ii. **Tareas:**
            1. Revisar los informes de errores y solicitudes de cambio.
            2. Priorizar los problemas según su impacto y urgencia.
            3. Realizar un análisis detallado de los problemas identificados.
        iii. **Duración:** 1 semana
    b. **Sprint 2:** Implementación de correcciones
        i. **Objetivo:** Implementar las correcciones necesarias para abordar los problemas identificados en el sprint 1.
        ii. **Tareas:**
            1. Desarrollar y probar las soluciones para los errores identificados.
            2. Realizar pruebas de integración para garantizar que las correcciones no introduzcan nuevos problemas.
            3. Documentar los cambios realizados.
        iii. **Duración:** 2 semanas
    c. **Sprint 3**: Actualización de Documentación
        i. **Objetivo:** Actualizar la documentación del software para reflejar los cambios realizados durante el sprint 2.

ii. Tareas:

1. Revisar y actualizar manuales de usuarios, guías de instalación y documentación técnica.
2. Añadir descripciones detalladas de las nuevas funcionalidades o cambios introducidos.
3. Verificar la coherencia y precisión de la documentación actualizada.

iii. **Duración**: 1 semana

d. **Sprint 4:** Pruebas y Validación Final

i. **Objetivo:** Realizar pruebas exhaustivas para validar las correcciones implementadas y la documentación actualizada.

ii. Tareas:

1. Ejecutar pruebas de regresión para asegurar que las correcciones no hayan introducido nuevos problemas.
2. Validar la documentación actualizada con usuarios finales.
3. Preparar la versión final del software y la documentación para su lanzamiento.

iii. **Duración:** 1 semana

e. **Sprint 5**: Revisión y Retrospectiva

i. **Objetivo:** Revisar el proceso del sprint y realizar una retrospectiva para identificar áreas de mejora.

ii. Tareas:

1. Revisar los resultados obtenidos durante el sprint.
2. Identificar puntos fuertes y áreas de mejora en el proceso de mantenimiento del software.
3. Planificar acciones correctivas y mejoras para futuros Sprints.

iii. **Duración:** 1 semana.

4- Finalizada la planificación, se procedió con la implementación del mantenimiento.

La norma ISO/IEC 12207, Mantelasoft y Scrum son tres enfoques diferentes y con similitudes en su orientación para el desarrollo de software y la gestión de proyectos, Se detallan las similitudes localizadas para su aplicación en el TFM:

- **Proceso y fases definidas:** Cada uno de estos enfoques define procesos y fases para el desarrollo de software. Por ejemplo, la norma ISO/IEC 12207 describe procesos

como la gestión de la configuración, el aseguramiento de la calidad y la gestión de proyectos. Mantelafost y Scrum también tienen procesos y fases definidos para la gestión de proyectos y el desarrollo de software.

- **Enfoque en la mejora continua:** Estos enfoques promueven la mejora continua en el desarrollo de software y la gestión de proyectos. Esto se logra mediante la retroalimentación, la revisión de procesos y la adaptación a los cambios.
- **Colaboración y trabajo en equipo:** Tanto Mantelasoft como Scrum enfatizan la colaboración y el trabajo en equipo entre los miembros del equipo de desarrollo. Esto se refleja en prácticas como las reuniones diarias en Scrum y el enfoque en la comunicación efectiva en Mantelasoft.
- **Flexibilidad y adaptabilidad:** Scrum y Mantelasoft son enfoques ágiles que valoran la flexibilidad y la adaptabilidad ente los cambios en los requisitos del proyecto. La norma ISO/IEC 12207 también proporciona estructuras que permiten la adaptación a cambios, aunque su enfoque es más general y no está específicamente diseñado como un marco ágil.

Gestión de la documentación

El enfoque central de este trabajo radica en el desarrollo de un proceso integral para el mantenimiento del software, poniendo especial énfasis en la creación y gestión de la documentación asociada. Este aspecto es crucial, ya que la documentación juega un rol fundamental en asegurar la comprensión, el seguimiento y la evolución eficiente de los sistemas de información. Una adecuada gestión de la documentación no solo actúa como un registro detallado de los requisitos, diseño, implementación y pruebas, sino que también facilita la gestión de proyectos y previene problemas recurrentes en el mantenimiento, dificultades de colaboración entre equipos, y malentendidos por parte de los usuarios finales [35].

La relevancia de este proceso reside en su capacidad para mejorar tanto la eficiencia como la calidad del mantenimiento de software dentro de la organización. Una documentación exhaustiva y bien gestionada sirve como una guía clara para asegurar un mantenimiento adecuado y la evolución continua del software a lo largo del tiempo. Además, facilita la comprensión de los sistemas de información críticos de la empresa, promoviendo una gestión más efectiva de los recursos disponibles.

Uno de los elementos clave en el mantenimiento de software es, sin duda, la documentación, la cual garantiza la calidad y sostenibilidad de los sistemas informáticos. En este trabajo, se

incorpora la documentación como un componente esencial para preservar la calidad del software y asegurar la eficacia del proceso de mantenimiento, siguiendo las directrices de Mantelasoft y aplicando las mejores prácticas de esta metodología.

Para garantizar un manejo eficiente de la documentación generada durante el mantenimiento del **Sistema de Control de Tráfico (SCT)**, se implementarán dos herramientas clave:

1. **Google Drive**:
    - **Plataforma colaborativa**: Se utilizará Google Drive para la creación y edición de documentos en tiempo real, facilitando una colaboración eficiente entre los miembros del equipo.
    - **Almacenamiento en la nube**: Los documentos se almacenarán en Google Drive, lo que garantizará su accesibilidad remota y permitirá el trabajo desde cualquier ubicación.
    - **Control de versiones**: La plataforma permitirá mantener un historial de versiones, facilitando la recuperación de documentos previos cuando sea necesario.
2. **Servidor \SRVFS**:
    - **Repositorio local**: El servidor \SRVFS se utilizará como repositorio interno para almacenar copias de seguridad de documentos críticos y versiones finales de los informes.
    - **Seguridad y control**: Este sistema proporcionará un mayor control sobre los permisos de acceso, asegurando que solo usuarios autorizados puedan modificar los documentos.
    - **Backup y redundancia**: Se establecerán copias de seguridad periódicas desde Google Drive hacia \SRVFS para garantizar la preservación de documentos importantes.

Con estas herramientas, se asegurará que la documentación esté siempre actualizada, accesible y segura, contribuyendo a la mejora continua del proceso de mantenimiento.

En resumen, este trabajo tiene como objetivo desarrollar un proceso integral de mantenimiento de software, con un enfoque principal en la gestión de la documentación, con el fin de mejorar tanto la eficacia como la calidad del mantenimiento en pequeñas y medianas empresas (PyMEs). La investigación se fundamenta en la metodología Mantelasoft, reconocida por su eficacia en la gestión de proyectos de software, lo que proporcionará una base sólida y confiable para la implementación de este proceso.

4.3 Propuesta de proceso

En este apartado, se describe el proceso utilizado para recopilar información precisa y estructurada sobre el mantenimiento del sistema de control de tráfico utilizado por una PyME de la región. Para ello, se propone una plantilla para documentar el mantenimiento correctivo, basada en la norma **ISO/IEC 12207** Plantilla 1.

La plantilla incluye los siguientes elementos clave:

- Definición de los requerimientos del sistema.
- Requerimientos específicos para la documentación.
- Correcta documentación del mantenimiento realizado.

Para la futura elaboración de la documentación, se deben seguir las siguientes directrices de formato:

1. **Encabezado**: Cada página del documento incluirá:
    - **Logo de la empresa**.
    - **Nombre del documento**: el título correspondiente.
    - **Versión**: Comienza con la versión 01 y se incrementa a medida que se realicen cambios.
    - **Fecha**: En formato (DD-MM-AAAA), que indica cuándo se aprueba la versión actual.
    - **Paginación**: Estructurada como "Página X de Y".
2. **Pie de página**: En la primera página, se incluirán los siguientes apartados (excepto en formatos donde no sea necesario):
    - **Elaborado por**: Nombre y firma de la persona encargada de elaborar el documento.
    - **Revisado por**: Nombre y firma de la(s) persona(s) encargada(s) de revisar el documento.
    - **Aprobado por**: Nombre y firma de la(s) persona(s) encargada(s) de aprobar el documento.

El proceso de documentación sigue los lineamientos de la norma ISO/IEC 12207, complementado con los principios de Scrum y Mantelasoft. Estos enfoques permiten gestionar el ciclo de vida del software de manera eficiente, asegurando la actualización continua de la documentación y la trazabilidad de los cambios.

La implementación del plan de mantenimiento también considerará los roles y actividades clave basados en Scrum, priorizando tareas en función de su urgencia e impacto. Se organizarán reuniones de planificación de Sprints, donde el equipo asigna las tareas correspondientes y define los objetivos de cada Sprint, con entregas parciales de documentación al final de cada iteración.

De esta forma, se asegura que tanto los problemas urgentes como los no urgentes se gestionen de manera eficiente, siguiendo un proceso claro que combina la flexibilidad de Scrum con la rigurosidad de ISO/IEC 12207.

Plantilla 1. Documentación de Mantenimiento Correctivo

	DOCUMENTACIÓN	VERSION: XX
	Nombre del procedimiento	FECHA: dd/mm/aaaa
		PAGINA: 1 de xx

Versión xx
Elaborado por: xxx
Revisado por: Persona 1 / Gerente del área de sistemas: Persona 2
Fecha:

1 OBJETIVO

El objetivo deberá contener una explicación del propósito que se pretende cumplir con el procedimiento; su elaboración se ajustara a los lineamientos que se describen a continuación.

- Especificar con claridad la finalidad que pretende el documento.
- La redacción será clara, concreta y directa.

El objetivo deberá ser lo más concreto posible, y su redacción clara y en párrafo breves, además, la primera parte de su contenido deberá expresar "que se hace"; la segunda, "para que se hace"

2 ALCANCE

Describe el ámbito de aplicación del procedimiento, es decir, a que áreas involucra, puesto y actividades, así como qué no aplica.

Responsabilidades y funciones de las personas/cargos incluidos:

Aquí se debe indicar quien es el responsable de la elaboración, emisión, control, vigilancia del proceso, así como también, quien es el responsable de la revisión y aprobación del mismo.

Descripción del funcionamiento sistema para proporcionar mantenimiento

Nombre del sistema: XXXX

Descripción del sistema: Aquí se procede hacer una breve descripción del funcionamiento del sistema al que se le realizará el mantenimiento

Responsable del mantenimiento del software: XXXX

Objetivo del mantenimiento: Se describe que se pretende llegar con el mantenimiento elegido.

Proceso de mantenimiento XX siguiendo la norma ISO/IEC 12207

3 **IDENTIFICACIÓN DE PROBLEMAS:** Se describe cual es el problema a resolver.

4 **ANÁLISIS DEL PROBLEMA Y EVALUACIÓN DE IMPACTO:** Una vez identificado el problema se procede hacer un análisis del mismo para poder identificar las causas y posibles soluciones.

5 **PRIORIZACIÓN DE PROBLEMAS:** Una vez analizado los problemas se dividen los mismos en prioridades para poder revolverlos según el grado de criticidad.

6 **DOCUMENTACIÓN DE PROBLEMAS**: Se debe documentar por cada error detectado las posibles soluciones realizadas.

7 **DISEÑO DE SOLUCIONES:** se debe diseñar las soluciones para poder resolver los errores detectados lo mismos pueden ser parcial o permanente.

8 **IMPLEMENTACIÓN DE SOLUCIONES:** Una vez que se tenga detectados las soluciones se debe proceder a ser implementados.

9 **PRUEBAS DE ACEPTACIÓN:** Se debe hacer participar al usuario para identificar si se solucionó el inconveniente o no.

10 **MONITOREO Y EVALUACIÓN:** Se realiza un seguimiento para controlar el comportamiento luego de la implementación y uso de las modificaciones realizadas.

11 **CIERRE DEL PROCESO:** Fecha de cierre. Evaluación de la eficacia de la solución: Análisis sobre la efectividad y la continuidad operativa del sistema después de la corrección. Firma de conformidad: responsable Técnico, validación del usuario, aprobación final.

12 **OBSERVACIONES FINALES:** Resumen de resultados. Recomendaciones: Propuestas para fortalecer y evitar recurrencias del problema.

Elaborado por:	Firma	Fecha
Revisado por:		
Aprobado por:		

El proceso de mantenimiento correctivo diseñado en el TFM, basado en la norma ISO/IEC 12207 y la metodología Mantelasoft, se representa en la *Fig. 5*. Cada cuadro en el diagrama representa una fase clave del proceso de mantenimiento correctivo. Desde la detección del problema hasta el cierre del proceso, cada etapa involucra diferentes áreas del equipo técnico, como Desarrollo, Infraestructura y Pruebas, las cuales desempeñan un papel crucial en la resolución del problema y la validación del resultado final.

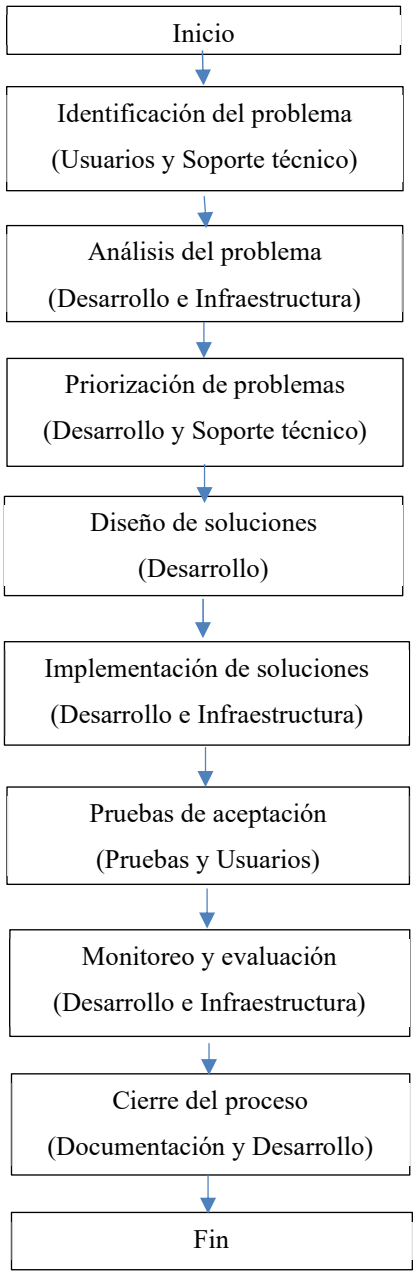


Fig. 5 Diagrama de flujo – Fases del mantenimiento correctivo

Elaboración propia

El flujo asegura que cada área involucrada cumpla con sus responsabilidades durante el ciclo de mantenimiento, garantizando así que los problemas sean abordados eficazmente y que las soluciones sean implementadas y validadas correctamente

4.4 Gestión de Cambios y Permisos de Acceso

Para asegurar la trazabilidad y control en la documentación, el sistema registrará cada cambio realizado, generando una nueva versión que incluirá detalles como el autor de la modificación y la fecha en que se efectuó. Este proceso de control de versiones es esencial para mantener un historial preciso y actualizado, permitiendo una referencia clara en futuras consultas y auditorías.

Permisos de acceso según el rol

Dado que la documentación es clave en los procesos de mantenimiento, resulta fundamental establecer permisos de acceso diferenciados según el rol de cada miembro del equipo. A continuación, se describen los niveles de acceso recomendados:

- **Desarrolladores**: Acceso completo a la documentación técnica, código fuente, registros de errores y solicitudes de cambio.
- **Personal de Mantenimiento**: Acceso a la documentación operativa y manuales de usuario, que les permita realizar sus actividades sin comprometer documentos críticos.
- **Coordinadores**: Acceso a informes, plantillas de mantenimiento correctivo y capacidad para revisar o aprobar cambios en la documentación.

Servidor Compartido con Control de Acceso

Se recomienda la implementación de un servidor compartido en la infraestructura de la PyME para alojar toda la documentación relacionada con el mantenimiento de software. Este servidor deberá estar configurado con políticas estrictas de control de acceso y auditoría, garantizando así la seguridad y disponibilidad de la información. La estructura propuesta para la organización de documentos es la siguiente:

- **Carpetas organizadas por tipo de documento**: Facilitará el acceso rápido y ordenado a la información. Las carpetas se estructurarán de la siguiente manera:
    - **Mantenimiento Preventivo**: Guías y procedimientos de mantenimiento preventivo.

- **Mantenimiento Correctivo**: Informes de errores y plantillas de solución alineadas con el estándar documentado.
- **Manual del Sistema Control de Tráfico (SCT)**: Instrucciones detalladas para el uso y mantenimiento operativo del sistema SCT.

Control de Acceso Basado en Roles

Para garantizar la integridad de los documentos y el cumplimiento de los protocolos de mantenimiento, se aplicarán permisos de acceso según el rol de cada usuario:

- **Lectura y escritura**: Permitida exclusivamente a desarrolladores senior y al Scrum Master, quienes podrán modificar documentos críticos como plantillas de mantenimiento y documentación técnica.
- **Solo lectura**: Personal de mantenimiento y otros miembros del equipo tendrán acceso a consultar manuales, plantillas y guías sin la posibilidad de editarlas.

Este esquema de control de acceso protege la integridad de los documentos, un aspecto esencial en los procesos de mantenimiento de software.

Copias de Seguridad Periódicas

Para prevenir la pérdida de información, se deberá implementar un sistema automatizado de copias de seguridad periódicas.

Tecnologías Recomendadas

- **Servidor local**: Permite el acceso seguro y controlado a archivos dentro de la red local.
- **FTP Seguro (SFTP)**: Protocolo recomendado para la transferencia de archivos, proporcionando un acceso seguro y restringido a los documentos alojados en el servidor.

4.5 Validación de la propuesta

En esta sección, se valida la propuesta de mantenimiento correctivo implementada en una PyME del sector de servicios. El objetivo principal es abordar los problemas identificados a través de una estrategia planificada, garantizando la efectividad del mantenimiento y la satisfacción del usuario final.

Evaluación de problemas

El primer paso en el proceso de validación fue la evaluación de los errores reportados por los usuarios. Escuchar atentamente sus quejas y observaciones permitió identificar los módulos específicos del sistema que requerían atención. Esto no solo mejoró la experiencia del usuario, sino que también proporcionó una visión clara de las áreas críticas dentro del Sistema de Control de Tráfico (SCT).

Una vez recopilados los informes de errores, se procedió a investigar sus causas subyacentes. Para esto, se realizaron pruebas exhaustivas que permitieron determinar si los fallos eran consecuencia de errores en el código, problemas de compatibilidad o deficiencias en la interfaz de usuario. Este análisis profundo permitió formular soluciones efectivas y establecer acciones correctivas para evitar la recurrencia de los problemas en el futuro.

Soluciones planteadas por el equipo

El equipo técnico, en coordinación con los usuarios finales, se enfocó en corregir los problemas identificados y garantizar la eficacia de las soluciones propuestas. El proceso de solución siguió los siguientes pasos clave:

1. **Actualización del sistema**: Se llevaron a cabo mejoras en la infraestructura del sistema, que incluyeron la adición de recursos a los servidores y la implementación de parches de software para optimizar el rendimiento.
2. **Corrección de errores en el código fuente**: Los problemas identificados fueron corregidos directamente en el código. Las pruebas posteriores aseguraron que los errores no se reprodujeran ni generaran nuevos fallos en el sistema.
3. **Implementación de mejoras en el tiempo de respuesta**: Se optimizaron los tiempos de carga y el acceso a los módulos del SCT, mejorando la experiencia del usuario al interactuar con el sistema.
4. **Pruebas con usuarios**: Los usuarios probaron las soluciones implementadas, lo que permitió confirmar que estas abordaban adecuadamente las necesidades planteadas. Su retroalimentación fue fundamental para validar el éxito de las correcciones realizadas.

Proceso de trabajo del equipo

Para ejecutar este proceso de manera eficiente, se sugirió trabajar en fases, como ser [26], lo que permitió una mejor organización y un seguimiento detallado de cada tarea:

- **Planeación**: Esta fase se centró en la solución de los problemas reportados, tanto de hardware como de software. Se lograron resultados inmediatos al reducir los tiempos de inactividad del sistema, mejorando la eficiencia operativa de la PyME.
- **Reclutamiento del equipo**: Se definieron los roles y responsabilidades de los miembros del equipo de trabajo, asegurando la colaboración efectiva entre todas las partes involucradas en el proceso de mantenimiento.
- **Identificación de acciones de mejora**: Con base en la metodología Mantelasoft, se evaluaron las fases del mantenimiento, identificando áreas de mejora en el proceso y estableciendo prioridades según la urgencia de los problemas reportados.
- **Análisis y reporte de resultados**: Se documentaron todas las etapas del mantenimiento, proporcionando una visión integral del estado del sistema y de las mejoras implementadas.

Problemas detectados y soluciones implementadas

Durante la ejecución del mantenimiento correctivo, se identificaron diversos problemas, los cuales fueron priorizados según su impacto en el sistema. Entre los principales problemas detectados se encuentran:

- **Saturación del servidor**: Esto ocasionaba caídas en el sistema y desconexiones para los usuarios. Se implementaron reinicios automáticos fuera de los horarios operativos para evitar la saturación causada por consultas pendientes.
- **Tiempos de respuesta lentos**: Al acceder a ciertos módulos, el sistema presentaba demoras. Para resolver esto, se añadieron más recursos al servidor y se realizaron ajustes en la configuración del sistema.
- **Errores en la base de datos**: Se detectaron problemas en los archivos **tempdb**, que no estaban configurados correctamente. Como solución, se implementó una configuración automática de crecimiento de la base de datos.

Documentación y trazabilidad

Una parte crucial del proceso fue la documentación exhaustiva de cada corrección realizada. Se estableció un sistema de registro que permite mantener una trazabilidad detallada de las intervenciones, garantizando que cualquier problema futuro pueda ser abordado de manera eficiente. La Plantilla 2 muestra el diseño de la plantilla utilizada para documentar el mantenimiento correctivo, siguiendo los lineamientos establecidos en la sección 4.3.

Plantilla 2. Plantilla para documentar el mantenimiento correctivo

	DOCUMENTACIÓN	VERSION: 1.0
	REGISTRO DE MANTENIMIENTO CORRECTIVO	FECHA: 10-07-2023
	PROPUESTA PARA UNA PYME DE SERVICIOS DE LA REGIÓN NEA	PAGINA: 4 de 14

SCT
Fecha:
Versión 1.0
Responsables: Equipo de Desarrollo e Infraestructura
Revisado por: Luciana Marcela Aguirre
Revisado por: Sonia Mariño / Gerente del área de sistemas

1 OBJETIVO

Documentar el proceso de mantenimiento correctivo para registrar fallas, aplicar correcciones y garantizar la trazabilidad del sistema SCT, asegurando un proceso de corrección efectivo según las prácticas de ISO/IEC 12207 y Mantelasoft.

2 ALCANCE

Este documento se aplica a todos los módulos del SCT en una PyME de transporte de la región NEA, abordando problemas que afectan la operación y calidad del servicio.

3 IDENTIFICACIÓN DEL PROBLEMA

Área afectada: Módulo de inicio de sesión / base de datos / servidores

Descripción del problema: Usuarios reportan problemas como errores en inicio de sesión, caída de conexión con base de datos y desconexiones causadas por tráfico elevado en internet. Estos errores afectan la continuidad de las operaciones del SCT.

Fuente de información: Correos electrónicos enviados por los usuarios, capturas de pantalla y enlaces de acceso al sistema.

4 ANALISIS DEL PROBLEMA Y EVALUACIÓN DE IMPACTO

Causa raíz: los errores se deben a la saturación de servidores, falta de optimización en el rendimiento y configuraciones incorrectas en archivos de base de datos como tempdb.

Procedimiento de análisis: se realizaron pruebas exhaustivas en las áreas afectadas para identificar si los fallos provenían de errores en código, problemas de compatibilidad o deficiencias en la interfaz de usuario.

Evaluación del impacto: Afecta la disponibilidad del sistema, ralentiza la respuesta y causa de interrupciones en el despacho de servicios.

Análisis de incidentes relacionados: documentar problemas similares registrados anteriormente y sus soluciones.

Trazabilidad de dependencias: Descripción de otros módulos o sistemas que podrían verse afectados directa o indirectamente.

5 PRIRIZACIÓN DE PROBLEMAS

Criterio: Los problemas fueron clasificados según la criticidad y el impacto en la operación del SCT.

Resultados: Los problemas con mayor criticidad (saturación de servidores y errores en la base de datos) se atendieron de forma prioritaria.

6 DOCUMENTACIÓN DE PROBLEMAS

Objetivo: Mantener un registro detallado de problemas y decisiones, facilitando la trazabilidad de las intervenciones.

Medio de documentación: Se utiliza una plantilla estandarizada para registrar cada problema y su respectiva solución.

7 DISEÑO DE SOLUCIONES

Soluciones identificadas:

2- Actualización del sistema: Mejora de infraestructura y adición de recursos en servidores.
3- Corrección de errores en código: Se realizaron ajustes en el código fuente para evitar los errores identificados.
4- Optimización de tiempo de respuesta: Ajustes en la configuración para mejorar los tiempos de carga.

8 IMPLEMENTACIÓN DE SOLUCIONES

Proceso: Las soluciones se implementaron en periodos de bajo uso del sistema. En casos críticos, se reiniciaron los servidores para evitar interrupciones mayores.

Seguimiento de proceso: Cada cambio se realizó asegurando su efectividad mediante pruebas exhaustivas antes de ser implementado en producción.

9 PRUEBAS DE ACEPTACIÓN

Objetivo: Confirmar la solución con el usuario, garantizando que el problema se ha resuelto sin generar otros inconvenientes.

Resultados: Las pruebas fueron exitosas, permitiendo al usuario retomar las operaciones con normalidad.

10 MONITOREO Y EVALUACIÓN

Actividad: Supervisión continua para asegurar la efectividad de las soluciones adoptadas.

Mecanismo: se monitorearon informes por correo electrónico, observando una disminución de errores a medida que se implementaron correcciones.

11 CIERRE DEL PROCESO

Condiciones de cierre: Cada problema fue documentado y solucionado de acuerdo con las necesidades del sistema.

Documentación final: Se elaboró un informe de cada intervención, detallando los problemas resueltos y las mejoras implementadas.

12 OBSERVACIONES FINALES

Observaciones: La implementación de este proceso de mantenimiento correctivo permitió mejorar la estabilidad y el tiempo de respuesta del SCT, beneficiando a los usuarios finales y reduciendo las interrupciones operativas.

Conclusión: se logró una respuesta ágil y efectiva, alineada con los lineamientos de la norma ISO/IEC 12207, garantizando una documentación adecuada y trazabilidad.		
Elaborado por:	Firma	Fecha
Revisado por:		
Aprobado por:		

En la Tabla 1, se observan las actividades realizadas para informar sobre la detección de un error, la actividad que se está llevando a cabo, el responsable encargado de resolver esa actividad y el registro correspondiente, con el fin de dejar documentado casa solución de cada actividad.

Tabla 1. Pasos de procesos de documentación

NRO	ACTIVIDAD	RESPONSABLE	REGISTRO
1	El usuario que identifique la necesidad de crear, modificar o eliminar un documento del Sistema Control de Tráfico (SCT) debe informar al responsable del proceso correspondiente a través del formato se Solicitud de Elaboración, modificación y eliminación de documentos.	Usuario del sistema	Formulario para solicitud de elaboración, modificación y eliminación de documentos
2	El responsable del proceso revisa la solicitud utilizando el formulario de solicitud de elaboración, modificación y eliminación de documentos. Posteriormente, determina su viabilidad y comunica su decisión al solicitante a través del mismo formulario, en la sección de observaciones. Si la solicitud es viable, se procede con la actividad 3; de lo contrario, se concluye el proceso.	Responsables de los procesos	Formato de solicitud de elaboración,
3	El responsable del proceso designa a un analista para elaborar o modificar el documento requerido, y a un desarrollador para llevar a cabo la modificación correspondiente.	Responsable de los procesos	Correo, etc.
4	El analista y el desarrollador asignados elaboran la propuesta del documento según los procesos establecidos y la entregan al líder del proyecto. Este líder revisa el contenido de la documentación, y también se solicita al desarrollador que verifique el cumplimiento de las disposiciones establecidas para su elaboración.	Analista y desarrollador	No aplica
5	El líder del proyecto examina el documento para garantizar que su contenido sea apropiado y cumpla con los lineamientos de la empresa. Si se identifica alguna discrepancia entre lo establecido en el proceso y la realidad, o si los criterios de elaboración del documento no se cumplen, se devuelve el documento con las observaciones correspondientes para realizar los ajustes necesarios. Luego, se repite la actividad número 4.	Líder de proyecto	No aplica

6	Una vez que los responsables de la revisión han firmado el documento, se procede a entregarlo al responsable del área correspondiente para su aprobación.	Responsable del área	No aplica
7	El responsable del área aprueba el documento y lo entrega al líder del proyecto	Líder de proyecto	No aplica
8	El líder del proyecto registra los documentos aprobados en el formato de Documentos internos y procede a realizar las actualizaciones necesarias junto con el desarrollador asignado. Cada modificación del documento se registra en la tabla de control de cambios correspondiente.	Líder de proyecto	Documento interno

En el Trabajo Final de Maestría (TFM), se desarrolló un proceso de mantenimiento correctivo urgente, el cual se aplica en situaciones donde los fallos detectados tienen un alto impacto en la operatividad del sistema, requiriendo una solución inmediata. Los errores clasificados como urgentes suelen afectar áreas críticas de la empresa, como la disponibilidad del sistema y la calidad del servicio prestado. Para determinar la criticidad de un problema y su necesidad de corrección urgente, se utilizaron los siguientes criterios:

1. **Impacto en las operaciones**: Si el fallo afecta directamente las funciones críticas del sistema, tales como el control de tráfico o la capacidad de procesar solicitudes esenciales.
2. **Gravedad del fallo**: Problemas que imposibilitan el funcionamiento del sistema o que pueden causar pérdidas de datos significativas.
3. **Tiempo de respuesta**: Errores que, si no se abordan rápidamente, pueden provocar tiempos prolongados de inactividad y afectar al despacho de los buses.
4. **Seguridad del sistema**: Fallos que comprometen la seguridad o integridad de los datos, representando un riesgo para la empresa o los usuarios.

El mantenimiento correctivo urgente implica actuar rápidamente desde la detección del problema hasta su solución, sin pasar por el ciclo de planificación típico de otros tipos de mantenimiento. A continuación, se detallan las fases del proceso según se describen en la Tabla 2:

Tabla 2. Mantenimiento correctivo urgente

Etapas	Tarea
Generalidades del mantenimiento	- Se analizaron los reportes diarios de los reclamos reportados por los usuarios. - Se preparó la propuesta inicial para la ejecución del mantenimiento correctivo.
Planificación	- Se planificó un cronograma de actividades y se definieron horarios para las implementaciones y las correcciones sin interrumpir el flujo de trabajo. - Se verificó el estado del sistema y se chequeó el código fuente para identificar los errores. - Se asignaron y agregaron los recursos necesarios para garantizar la efectividad de la solución.
Proceso de mantenimiento	- Se verificaron los procesos clave operados por el SCT, como las bases de datos y servidores vinculados. - Se ejecutaron las correcciones necesarias para solucionar los problemas detectados en el menor tiempo posible.
Evolución o mejora del producto	- Se evaluó el impacto de las correcciones, observando una mejora del sistema cuando no se recibieron nuevos reclamos de los usuarios. - Se continuó monitoreando el sistema para asegurar la estabilidad de las soluciones implementadas.

Proceso detallado:

1. **Generalidades del Mantenimiento**: En esta etapa inicial, se analizaron los reportes diarios enviados por los usuarios y se identificaron los problemas críticos que requerían atención urgente. Con esta información, se elaboró una propuesta para realizar el mantenimiento correctivo, asegurando que los recursos necesarios estuvieran disponibles para una resolución rápida.
2. **Planificación**: Aunque el proceso de mantenimiento correctivo urgente no sigue un ciclo de planificación extenso, se realizó una planificación rápida para definir el momento adecuado para ejecutar las correcciones, minimizando el impacto en las operaciones del negocio. Además, se asignaron los recursos necesarios para resolver el problema de manera eficiente, lo que incluyó agregar recursos a los servidores si era necesario.
3. **Proceso de Mantenimiento**: Durante la fase de ejecución, se verificaron los procesos asociados con el Sistema de Control de Tráfico (SCT), incluyendo las bases de datos y servidores. Una vez identificadas las fallas, se aplicaron las correcciones

necesarias. Este proceso garantizó que los problemas se resolvieran rápidamente para evitar tiempos de inactividad prolongados.

4. **Evolución o Mejora del Producto**: Tras la implementación de las soluciones, se realizó un monitoreo continuo para asegurarse de que el sistema funcionara correctamente. Se observó una mejora notable cuando no se recibieron más reclamos de los usuarios, lo que indicó que las correcciones habían sido efectivas. El monitoreo continuó para garantizar la estabilidad del sistema a largo plazo.

Cada una de estas fases permitió priorizar las acciones necesarias para resolver el problema en el menor tiempo posible. La clasificación de los problemas como urgentes y la rápida intervención aseguraron que el mantenimiento correctivo urgente lograra reducir el mínimo el impacto en las operaciones de la empresa, preservando la calidad del servicio y la seguridad del sistema.

Para el mantenimiento correctivo no urgente: se reservaron las mejoras que, aunque no impactan de manera inmediata en el funcionamiento del sistema, resultan esenciales para su óptimo desempeño, como se detalla en la Tabla 3.

Tabla 3. Mantenimiento correctivo no urgente

Etapas	Tarea
Generalidades del mantenimiento	- Se analizaron los reportes diarios de los reclamos reportados - Se preparó la propuesta para la realización del mantenimiento
Planificación	- Se planeó un cronograma de actividades, horario para una de las implementaciones - Se analizó el código del sistema para identificar en que módulos se registraban los errores reportados. - Se estimó, en caso de ser necesario, la incorporación de recursos.
Proceso de mantenimiento	- Se documentaron las posibles soluciones - Se seleccionaron las acciones necesarias para realizar las modificaciones que se han decidido utilizar. - Se verificó y validó el funcionamiento del sistema luego de las modificaciones realizadas tanto con la monitorización del comportamiento como con los reclamos de los usuarios.
Evolución o mejora del producto	- Se visualizó una mejora del sistema, evidencia de la carencia de correos electrónicos con reclamos de los usuarios.

Mantenimiento adaptativo: son las modificaciones o actualizaciones llevadas a cabo con el equipo para adaptar al buen funcionamiento del sistema, como se representa en la Tabla 4.

La identificación de los distintos tipos de mantenimiento ha posibilitado el establecimiento de un control del rendimiento del servidor y las bases de datos que son cruciales para el funcionamiento del sistema. Además, se ha implementado un seguimiento de los correos electrónicos con los informes de los usuarios. Esta validación se realizó durante el período comprendido entre mayo y noviembre 2023.

En la Tabla 5, se observa que la implementación del mantenimiento del software durante los meses de mayo a noviembre 2023 ha resultado en una reducción significativa de la inactividad del SCT. Esta mejora se traduce en un ahorro considerable en términos de costos asociados a la pérdida de productividad y posible pérdida de información.

Durante este periodo, se nota una notable optimización en el rendimiento del sistema, con una disminución en los tiempos de inactividad y una mejora general en la experiencia del usuario.

Tabla 4. Mantenimiento adaptativo

Etapas	Tarea
Generalidades del mantenimiento	- Se analizaron los reportes diarios de los reclamos reportados por los usuarios. - Se elaboró la propuesta inicial del plan de mantenimiento.
Planificación	- Se desarrolló un cronograma de actividades y horarios para las implementaciones. - Se verificó el estado general del sistema y se evaluó el código fuente para detectar problemas. - Se estimó la necesidad de recursos adicionales para ejecutar las correcciones. - Se definieron las soluciones a implementar durante la fase de mantenimiento.
Proceso de mantenimiento	- Se realizaron copias de seguridad de las bases de datos antes de proceder. - Se implementaron las modificaciones y soluciones seleccionadas. - Se verificó y validó el correcto funcionamiento del sistema después de aplicar las correcciones. - Se monitoreó el sistema para evaluar el comportamiento y respuesta del sistema ante las modificaciones.
Evolución o mejora del producto	- Se observó una mejora en el sistema reflejada en la ausencia de nuevos reclamos por parte de los usuarios.

	- Se continuó monitoreando el sistema para asegurar que las mejoras implementadas solucionaran los problemas reportados.

Tabla 5. Comparativa sin mantenimiento y con mantenimiento

	Costos	**Tiempos de inactividad**
Antes – Previo a implementar plan de mantenimiento	70%	15 hs semanal
Después - Posterior a implementar plan de mantenimiento	30%	1 hs semanal

1. Tiempo de respuesta antes y después del mantenimiento

Mejora en el tiempo de respuesta:

- **Cálculo**: Mejora (%)

$$\text{Mejora (\%)} = \left(\frac{\textbf{Tiempo antes} - \textbf{Tiempo despues}}{\text{Tiempo antes}} \right) \text{x } 100$$

$$\left(\frac{15\ hs - 1\ hs}{15 hs}\right) \text{x } 100 = 0.93 \Rightarrow 93\%$$

2. Reducción en el número de errores o caídas del sistema

En el análisis del estado actual del sistema, además de la realización de evaluación del funcionamiento de cada uno de los módulos a mejorar y la identificación de las fallas más comunes, como demoras en realización de modificaciones de algún servicio o tiempos de respuesta lentos al realizar consultas o al iniciar sesión, también es importante establecer un plan de mantenimiento preventivo para evitar futuros problemas recurrentes. Este plan incluye el control del funcionamiento de las bases de datos, la supervisión de la capacidad de memoria disponibles en el servidor y, de ser necesario, la programación de reinicios para evitar la saturación del sistema.

Tabla 6. Reducción en el número de errores o caídas del sistema

	Tiempos de inactividad
Antes – Previo a implementar plan de mantenimiento	10 errores/mes
Después - Posterior a implementar plan de mantenimiento	2 errores/mes

Antes del mantenimiento:

- **Número de errores mensuales**: 30 errores/mes

Después del mantenimiento:

- **Número de errores mensuales**: 15 errores/mes

Reducción en el número de errores:

- **Cálculo**: Reducción (%)
- Reducción (%) = ($\frac{\textbf{Errores antes} - \textbf{Errores despues}}{\text{Errores antes}}$) x 100

- Antes del mantenimiento: 30 errores/mes
- Después del mantenimiento: 15 errores/mes
- Reducción:
- ($\frac{30\ hs - 15\ hs}{30 hs}$) x 100 = 0.5 => 50%

Resultados obtenidos

Tras la implementación de las soluciones, se realizó un seguimiento riguroso para garantizar que los problemas no volvieran a ocurrir y que el sistema operara de manera óptima. Los usuarios reportaron una notable mejora en el rendimiento del SCT, con tiempos de respuesta significativamente reducidos y menos interrupciones del servicio.

Se logró una reducción en los tiempos de inactividad del sistema, lo que contribuyó a un aumento en la productividad y una mejora en la experiencia del usuario. Estos resultados

validan la efectividad de las soluciones implementadas y confirman que el plan de mantenimiento correctivo propuesto es adecuado para las necesidades de la PyME.

Para mejorar la gestión de consultas e interacciones con cada módulo del sistema, se realizó mantenimiento en las distintas bases y servidores utilizados por el sistema, con el fin de mejorar el tiempo de respuesta y evitar caídas que impidieran su uso a los usuarios.
El seguimiento del sistema se lleva a cabo diariamente, principalmente a través del control realizado por el usuario final, quien reporta los inconvenientes por correo electrónico adjuntando capturas correspondientes de los casos problemáticos. Además, se realiza un monitoreo constante por parte del área de infraestructura del rendimiento de las bases de datos y de los servidores *Fig. 6*

Fig. 6. Propuesta de mantenimiento preventivo

Se procede a resumir las mejoras realizadas implementado el plan de mantenimiento. A continuación, se presenta una descripción concisa del contenido de cada columna de la Tabla 7.

- **Columna Descripción:** Esta columna proporciona una breve descripción del problema o la necesidad que se aborda en la cada fila de la tabla. Aquí se detalla el aspecto específico del sistema que requiere atención o mejora.
- **Columna soluciones previas al mantenimiento:** En esta columna se detallan las soluciones o acciones que se han implementado previamente para abordar el problema o la necesidad descrita. Estas soluciones pueden haber sido temporales, parciales o provisionales.
- **Columna soluciones después del mantenimiento:** Aquí se describen las soluciones o acciones que se implementarán como parte del plan de mantenimiento para abordar de manera más completa y efectiva el problema o las necesidades identificadas. Estas soluciones pueden incluir cambios en el software, ajustes en la configuración, actualizaciones de hardware, entre otras medidas.

Tabla 7. Mejora con el plan de mantenimiento

	Descripción	**Solución previa al mantenimiento**	**Solución después del mantenimiento**
1	El sistema se cae frecuentemente	Se reiniciaba manualmente el sistema	Implementación de parches y actualizaciones en el sistema
2	Los datos se perdían al guardar	Se guardaba el archivo de forma manual y se copiaba a otro lugar como backup	Implementación de un sistema de respaldo automático y revisión de permisos de usuario en el sistema
3	Los tiempos de respuestas eran lentos	Se cerraban todos los programas y se reiniciaba el equipo	Optimización del sistema y actualización de drivers de hardware
4	El sistema genera errores aleatorios	Se ignoraban los errores y se seguía trabajando	Revisión del código del software para corregir errores y mejorar la estabilidad
5	Los usuarios advertían mensajes de error al realizar ciertas acciones	Se solicitaba que el usuario intentara ingresar nuevamente o se contactará al soporte técnico	Corrección de errores en el código y mejora de la calidad del software.
6	El sistema se bloqueaba al acceder a ciertas funciones	Se reiniciaba el sistema para poder acceder a la función deseada	Corrección de errores en el código y optimización de las funciones críticas

Tabla 8. Comparativa de Indicadores Antes y Después del Mantenimiento

Indicador	Antes del Mantenimiento	Después del Mantenimiento	Mejora (%)
Tiempo de respuesta promedio	5 segundos	2 segundos	60%
Número de errores mensuales	30 errores	15 errores	50%
Disponibilidad del sistema	95%	99%	4.21%
Correos de soporte mensuales	20 correos	5 correos	75%

Impacto de la implementación

Tras la implementación de las fases de mantenimiento correctivo siguiendo estas metodologías, se logró optimizar significativamente la gestión del mantenimiento de software en la PyME. Este enfoque integrado, que combina metodologías ágiles con estándares internacionales y una documentación estructurada, ha demostrado ser eficiente, flexible y fácil de gestionar, ofreciendo una solución sostenible para el mantenimiento del software.

1. ISO/IEC 12207:
    - Aplicación: La norma ISO/IEC 12207 proporciona un marco detallado para estructurar cada intervención de mantenimiento, desde la identificación del problema hasta su solución. Su implementación en este proyecto permitió:
        - Crear un proceso documentado y repetible para cada intervención de mantenimiento.
        - Garantizar la calidad y trazabilidad en cada cambio realizado, facilitando la verificación y validación del código.
        - Asegurar que cada intervención tuviera un registro claro y accesible para futuras correcciones.

- Conclusión: La teoría sobre ISO/IEC 12207 se validó en la práctica, mejorando la eficiencia y calidad del mantenimiento del software.

2. Scrum:
    - Aplicación: El marco Scrum permitió abordar el mantenimiento de manera ágil, organizando el trabajo en ciclos iterativos (Sprints) y asegurando la flexibilidad en la ejecución de las tareas. Con su implementación, se logró:
        - Mayor adaptabilidad ante cambios, al permitir la priorización de tareas urgentes en cada Sprint.
        - Mejorar la comunicación y coordinación del equipo mediante reuniones diarias (Daily Scrum) y revisiones periódicas (Sprint Reviews).
        - Agilizar la toma de decisiones y la implementación de soluciones, garantizando la entrega continua de mejoras.
    - Conclusión: Scrum, como marco ágil, demostró ser efectivo para gestionar el mantenimiento del software de forma dinámica, facilitando una mayor adaptabilidad y eficiencia en la gestión de tareas.
3. Mantelasoft:
    - Aplicación: Mantelasoft fue clave para mantener actualizada la documentación del proceso de mantenimiento, lo que aseguró una trazabilidad completa de cada acción realizada. Su uso permitió:
        - Establecer una documentación clara y precisa de cada intervención, facilitando la revisión de las acciones realizadas.
        - Estandarizar los reportes, mejorando la comprensión y referencia futura.
        - Mantener un registro completo y accesible de cada modificación en el sistema.
    - Conclusión: La teoría sobre la importancia de una documentación estructurada fue validada, ya que Mantelasoft mejoró considerablemente la organización y control del proceso de mantenimiento.

La implementación de las metodologías ISO/IEC 12207, Scrum y Mantelasoft proporcionó una estructura sólida y flexible para el proceso de mantenimiento correctivo. Este enfoque no solo garantizó la eficiencia operativa, sino que también validó la relevancia de los principios teóricos expuestos en el marco conceptual. Los resultados obtenidos confirmaron que estas metodologías no solo son aplicables en teoría, sino que, integradas, ofrecen una

solución eficiente, trazable y escalable para la gestión del mantenimiento de software en PyMEs del sector transporte.

4.5 Aplicación del modelo en otros sectores

Aplicación del modelo de mantenimiento en estaciones de servicio de la región NEA

El modelo de mantenimiento propuesto puede ser adaptado para su uso en estaciones de servicio dentro de la región NEA. Este tipo de empresas, aunque no pertenecen directamente al sector del transporte, comparte algunas características similares que permiten aplicar los principios generales del modelo.

A continuación, se describen los ajustes específicos que se deben realizar para garantizar una implementación eficaz en este contexto.

1. Mantenimiento de equipos de combustible: Las estaciones de servicio dependen de equipos clave, como surtidores de combustible, tanques de almacenamiento y sistemas de filtrado. El modelo debe incluir una categoría específica para el mantenimiento preventivo y correctivo de estos equipos, documentando las verificaciones y calibraciones periódicas, así como los ciclos de mantenimiento establecidos según las normativas locales.

2. Estructura operativa simplificada: En comparación con las PyMEs del transporte, las estaciones de servicio suelen tener estructuras organizativas más reducidas. Esto implica una simplificación de roles en el equipo, donde el gerente o encargado puede asumir las responsabilidades del Scrum Master y el Product Owner, coordinando las tareas de mantenimiento y asegurando la documentación adecuada para los procesos correctivos.

3. Cumplimiento de normativas específicas: El modelo debe ajustarse para asegurar el cumplimiento de normativas de seguridad y ambientales. Las estaciones de servicio están sujetas a regulaciones estrictas sobre la gestión de combustibles y residuos peligrosos. Por lo tanto, es fundamental integrar procesos de inspección y control de acuerdo con las normativas vigentes.

4. Gestión de inventarios de combustibles: El manejo preciso de inventarios es esencial en una estación de servicio. Se deben incluir módulos de gestión de inventarios y control automático de existencias, con procedimientos claros para la supervisión de stock, calibración de tanques y detección de posibles pérdidas.

5. Automatización de sistemas y tecnologías: Las estaciones de servicio modernas dependen de tecnologías automatizadas, como sistemas de pago electrónico y sensores inteligentes. El modelo debe incluir procedimientos de mantenimiento para estos sistemas, así como la

documentación adecuada para el mantenimiento preventivo y correctivo del software y hardware involucrado.

Implementación práctica del modelo en estaciones de servicio

Para implementar este modelo ajustado, se seguirían los siguientes pasos:

1. Identificación de equipos críticos: Realizar un inventario de los equipos clave de la estación de servicio, priorizando aquellos que requieren mantenimiento preventivo.
2. Asignación de roles y responsabilidades: Definir roles específicos para el personal de la estación, asegurando que el gerente o encargado asuma un papel de liderazgo en la coordinación del mantenimiento.
3. Ciclo de mantenimiento adaptado: Establecer un ciclo de mantenimiento regular adaptado a las necesidades operativas y a las normativas locales, con revisiones periódicas de los equipos.
4. Cumplimiento normativo: Implementar un sistema de auditoría y documentación que asegure el cumplimiento de todas las normativas locales de seguridad y gestión de residuos.

La adaptación del modelo propuesto a estaciones de servicio en la región NEA es completamente factible con los ajustes descritos. Estos cambios permiten que el modelo aborde eficazmente las necesidades específicas de este sector, mejorando la eficiencia operativa, garantizando el cumplimiento normativo y asegurando la continuidad del servicio. Este enfoque integral podría ser implementado para validar la propuesta en un entorno real, reforzando la aplicabilidad práctica del modelo en otros sectores.

Capítulo 5

Conclusiones

5.1 Conclusiones

La realización de este Trabajo Final de Maestría permitió una profundización significativa en el campo de la Ingeniería del Software, con un enfoque particular en el Mantenimiento del Software, una de las áreas clave de esta disciplina. A través de la aplicación de enfoques teóricos, metodológicos y empíricos, se desarrolló una plantilla destinada a documentar y gestionar el mantenimiento correctivo de software, la cual fue validada mediante un estudio de caso en el área TIC de una PyME de servicios ubicada en la región NEA.

El objetivo general del trabajo se cumplió satisfactoriamente mediante la identificación, análisis y selección de estándares y metodologías relevantes, como Mantelasoft y Scrum, aplicadas de manera específica al proceso de mantenimiento del software. La metodología elegida permitió integrar un enfoque centrado en la documentación, un aspecto fundamental dado que uno de los objetivos principales de Scrum es entregar productos que satisfagan los requisitos del usuario en incrementos pequeños y frecuentes.

El análisis exhaustivo del proceso de implementación del mantenimiento del software reveló que la estrategia adoptada mejoró considerablemente el rendimiento y la fiabilidad del Sistema de Control de Tráfico (SCT). Los resultados demuestran un impacto positivo en la productividad y la eficiencia operativa de la PyME, evidenciado por la reducción significativa de la inactividad del sistema entre los meses de mayo y noviembre de 2023, así como una notable disminución en los costos derivados de la pérdida de productividad.

El seguimiento continuo de los requerimientos enviados al área de sistemas también validó los avances obtenidos, con una disminución gradual en la cantidad de reclamos de los usuarios, hasta prácticamente su inexistencia durante el mismo periodo. Esto reafirma el éxito del plan de mantenimiento implementado, que no solo optimizó el funcionamiento del sistema, sino también la experiencia del usuario.

El mantenimiento del software es un pilar fundamental para asegurar el correcto funcionamiento y la longevidad de los sistemas, y este trabajo demuestra que la elección de utilizar los principios de Mantelasoft, diseñados específicamente para PyMEs, fue una decisión acertada. Este enfoque permitió estructurar un proceso eficiente y adaptable, alineado con las necesidades de una pequeña y mediana empresa, donde los recursos limitados exigen una gestión estratégica del mantenimiento.

Al comparar los principios de Mantelasoft con los establecidos en la norma ISO/IEC 12207, se logró desarrollar un enfoque integral que adapta estas metodologías a las particularidades

del caso de estudio. Esta integración ha sido clave para crear una solución robusta que responda a los desafíos operativos de la empresa de transporte seleccionada.

En conclusión, la implementación del plan de mantenimiento del software, junto con el diseño de una plantilla específica para documentar el mantenimiento correctivo, ha sentado una base sólida para la gestión futura de la tecnología en esta PyME. Al continuar aplicando estos principios y prácticas recomendadas, se espera seguir mejorando la eficiencia y la confiabilidad de los sistemas, contribuyendo directamente al éxito a largo plazo de la empresa en un entorno empresarial cada vez más competitivo.

5.2 Líneas de trabajo futuro y recomendaciones para la mejora del mantenimiento correctivo de software

Entre las líneas de trabajo futuro en el ámbito del mantenimiento correctivo de software, se propone la validación de la plantilla diseñada para monitorear los distintos tipos de mantenimiento. Esta plantilla ha sido plasmada en el "Documento de proceso de mantenimiento del software" y en el "Documento de posibles soluciones" (Anexo 4), cubriendo tanto aspectos disciplinarios como aplicativos.

Aspectos disciplinares

Desde una perspectiva disciplinar, se recomienda explorar y desarrollar nuevas metodologías y enfoques destinados a incrementar la eficiencia y efectividad del mantenimiento correctivo. Este esfuerzo podría incluir investigaciones sobre técnicas avanzadas de detección y corrección de errores, así como el diseño de procesos más ágiles y flexibles, que se adapten mejor a las cambiantes demandas del mercado y a las tecnologías emergentes.

Además, resulta fundamental profundizar en el estudio de la gestión de riesgos y planificación estratégica del mantenimiento correctivo. El objetivo sería identificar y mitigar posibles obstáculos antes de que se conviertan en problemas significativos, minimizando así el impacto negativo en los sistemas.

Aplicación de nuevas herramientas y tecnologías

En el ámbito práctico, se debe continuar la evolución de herramientas y tecnologías utilizadas en el mantenimiento correctivo. Esto incluye el desarrollo de sistemas avanzados de detección automática de errores, el diseño de interfaces de usuario más intuitivas para la

gestión de problemas, y la implementación de técnicas de análisis de datos que permitan identificar patrones y tendencias en los problemas reparados.

Finalmente, se hace hincapié en la importancia de mejorar la colaboración y comunicación entre los equipos de desarrollo, los usuarios finales y otros actores involucrados. Una mejor coordinación permitirá responder de manera más ágil y eficaz a los problemas identificados.

Recomendaciones para la adaptación del modelo en otras PyMEs

El modelo propuesto ha demostrado su efectividad en el contexto de una PyME del sector transporte, pero es importante considerar su aplicabilidad en otros sectores. A continuación, se ofrecen algunas recomendaciones para futuras implementaciones en PyMEs de distintos sectores:

1. **Ampliación del modelo para PyMEs de diversos sectores**: Se debe explorar la adaptación del modelo para sectores como el comercio minorista, manufactura o servicios tecnológicos. Cada sector presenta desafíos específicos en términos de equipos, estructura organizativa y procesos operativos. Futuras investigaciones deberían centrarse en ajustar el modelo a estas particularidades.
2. **Personalización del ciclo de mantenimiento**: Es recomendable adaptar los ciclos de mantenimiento preventivo y correctivo según las características y necesidades de los equipos críticos de cada empresa. Un enfoque flexible permitirá ajustar la frecuencia y profundidad de las actividades de mantenimiento, garantizando su adaptabilidad a una variedad de procesos.
3. **Escalabilidad del modelo**: Una de las claves para futuras investigaciones es estudiar la escalabilidad del modelo en empresas de distintos tamaños y complejidades. Las PyMEs más pequeñas podrían beneficiarse de una simplificación del modelo, mientras que las más grandes requerirán una mayor especialización de los roles dentro del equipo de mantenimiento.
4. **Incorporación de tecnologías emergentes**: Tecnologías como inteligencia artificial (IA) pueden revolucionar la gestión del mantenimiento, ofreciendo soluciones predictivas y automatizadas que reduzcan la carga operativa y mejoren la eficiencia. Futuras implementaciones deben investigar cómo integrar estas tecnologías en el modelo de mantenimiento.
5. **Adaptación a diferentes normativas**: Las PyMEs de sectores como la salud, alimentación o manufactura están sujetas a normativas específicas de calidad y seguridad. El modelo debe ajustarse a los requisitos normativos de cada sector, lo que garantiza su cumplimiento y mejora la seguridad y la calidad de los servicios.

6. **Aplicación de metodologías ágiles en diversos sectores**: Se recomienda investigar cómo las metodologías ágiles, como Scrum, pueden ser adaptadas a sectores distintos al transporte. Esto incluye estudiar la viabilidad de la asignación de roles, la estructura del equipo y la frecuencia de revisiones, que puede requerir ajustes según la naturaleza y tamaño de la empresa.

En resumen, las líneas de trabajo futuro en el mantenimiento correctivo de software abarcan tanto aspectos disciplinarios como aplicativos. El enfoque en la innovación continua y la mejora de procesos y herramientas utilizados será clave para enfrentar los desafíos del futuro. Además, las recomendaciones prácticas planteadas para la aplicación del modelo en PyMEs de distintos sectores ofrecen una guía para futuras investigaciones e implementaciones, ampliando el impacto del modelo propuesto.

Glosario

Definiciones

Mantenimiento correctivo: La modificación no programada realizada para mantener temporalmente funcionando un sistema, teniendo pendiente realizar un mantenimiento correctivo.

Mantenibilidad: la capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a los cambios en el entorno y en requisitos y especificaciones funcionales.

Mantenedor: Una organización o persona que realiza actividades de mantenimiento.

Mejora de mantenimiento: Una mejora de mantenimiento es un cambio en el software que comprende un nuevo requisito.

Plan de mantenimiento: documento en el que especifica las prácticas de mantenimiento, los recursos y la secuencia de las actividades pertinentes para dar mantenimiento a un producto de software.

Solicitud de modificación (MR): término genérico que se utiliza para identificar los cambios propuestos a un producto de software,

Informar un problema (PR): término que se utiliza para identificar y describir los problemas detectados en un producto de software.

Mantenimiento de software: El mantenimiento del software es el conjunto de actividades necesarias para proporcionar un soporte eficiente a un sistema de software.

Migración de software: Es el proceso de transición de un producto de software desde un antiguo a un nuevo entorno operativo.

Baja de un software: Es el proceso de eliminación de un producto de software en funcionamiento una vez que el producto ha llegado al final de su vida útil.

Transición del software: una secuencia controlada y coordinada de acciones en el que el desarrollo del software pasa de la organización ejecutante desarrollo inicial del software a la organización de realizar el mantenimiento del software.

Referencias

[1] J. E. Martínez, A. F. Gómez, and F. J. Pino, “Generando productos software mantenibles desde el procesode desarrollo: El modelo de referencia MANTuS,” *Ingeniare*, vol. 24, no. 3, pp. 420–434, Jul. 2016, doi: 10.4067/S0718-33052016000300007.

[2] O. Salim, L. Salgado, G. Rivera, J. Ignacio, and R. García, “Análisis Foda Sobre El Uso De La Inteligencia Competitiva En Pequeñas Empresas De La Industria Del Vestido Swot Analysis of the Use of Competitive Intelligence in Small Enterprisesin the Clothing Industry,” *Rev. Científica "Visión Futur.*, vol. 21, no. 1, pp. 78–99, 2017, [Online]. Available: https://www.redalyc.org/pdf/3579/357951171003.pdf.

[3] S. I. Mariño, P. L. Alfonzo, S. I. Mariño, and P. L. Alfonzo, “Las áreas de conocimiento SWEBOK en producciones de graduación. Un estudio de la disciplina Informática en la Facultad de Ciencias Exactas y Naturales y Agrimensura (FaCENA) de la Universidad Nacional del Nordeste (UNNE), Argentina,” *E-Ciencias la Inf.*, vol. 9, no. 2, pp. 103–120, Jun. 2019, doi: 10.15517/ECI.V9I2.35553.

[4] I. SOMMERVILLE, *Ingenieria del Software*, Séptima ed. 2004.

[5] J. I. Rocca Temesio, “Procesos de software orientados a la usabilidad,” Universidad ORT Uruguay Facultad de Ingeniería Procesos, 2019.

[6] L. de la C. Delgado Olivera, L. M. Díaz Alonso, L. de la C. Delgado Olivera, and L. M. Díaz Alonso, “Modelos de Desarrollo de Software,” *Rev. Cuba. Ciencias Informáticas*, vol. 15, no. 1, pp. 37–51, 2021, [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000100037&lng=es&nrm=iso&tlng=es.

[7] L. E. Rioseco Norambuena, “Análisis y evaluación del proceso de desarrollo de software para el área de consultoría de la Empresa PowerData Chile,” 2016, [Online]. Available: http://repositorio.uchile.cl/handle/2250/144486.

[8] “Documentación ágil: metodología y mejores prácticas.” https://document360.com/blog/agile-documentation/ (accessed Feb. 23, 2024).

[9] L. M. Suárez, A. F. Montoya, and J. I. Vélez, “MANTELASOFT: una nueva guía para Mipymes desarrolladoras de mantenimiento de software,” *Entre Cienc. e Ing.*, vol. 13, no. 25, pp. 95–99, 2019, doi: 10.31908/19098367.4019.

[10] J. Gaete, R. Villarroel, I. Figueroa, H. Cornide-Reyes, and R. Muñoz, “Enfoque de aplicación ágil con Scrum, Lean y Kanban Agile application approach with Scrum, Lean and Kanban,” *Rev. Chil. Ing.*, vol. 29, no. 1, pp. 141–157, 2021.

[11] F. Flores-Cerna, V.-M. Sanhueza-Salazar, H.-M. Valdés-González, and L. Reyes-Bozo, “Metodologías ágiles: un análisis de los desafíos organizacionales para su implementación,” *Rev. Científica*, vol. 43, no. 1, pp. 38–49, 2021, doi: 10.14483/23448350.18332.

[12] J. Adones Farfán and V. Vega-Zepeda, “Mantenibilidad del Software. Consideraciones para su especificación y validación,” *Ingeniare. Rev. Chil. Ing.*, vol. 28, no. 4, pp. 654–667, Dec. 2020, doi: 10.4067/S0718-33052020000400654.

[13] E. Maida and J. Pacienzia, “Metodologías de desarrollo de software,” pp. 12–13, 2015, [Online]. Available: http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf.

[14] Universidad Nacional del Sur, “Guía para la documentación de proyectos de software.” 2017, [Online]. Available: https://cs.uns.edu.ar/~ldm/mypage/data/oc/info/guia_para_la_documentacion_de_proyectos_de_software.pdf.

[15] “1219-1992 - IEEE Standard for Software Maintenance - IEEE Standard.” https://ieeexplore.ieee.org/document/257623?denied= (accessed May 30, 2020).

[16] A. E. G. Codutti, S. I. Mariño, and P. L. Alfonzo, “Gestión de la información en Educación Superior: Una experiencia de evolución del software con sistemas gestores de contenidos,” *IJERI Int. J. Educ. Res. Innov. - Rev. Int. Investig. e Innovación Educ.*, vol. 0, no. 8, pp. 293–304, 2017.

[17] J. D. Erazo, A. S. Florez, and F. J. Pino, “Analysis and classification of software maintainability attributes: a comparative review from the state of the art,” *Entre Ciencia e Ingenieria vol.10 no.19 Pereira June 2016*. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1909-83672016000100006 (accessed Nov. 09, 2020).

[18] “El INTI desarrolló un software para reducir costos y mejorar la gestión en PyMEs | Argentina.gob.ar.” https://www.argentina.gob.ar/noticias/el-inti-desarrollo-un-software-para-reducir-costos-y-mejorar-la-gestion-en-pymes (accessed Aug. 05, 2022).

[19] V. A. Herrera, “‘Desarrollo de un plan de gestión de mantenimiento de Software para el Departamento de Sistemas de la Universidad de la Politécnica Salesiana basado en la Norma ISO/IEC 14764:2006,’” Universidad Politécnica Salesiana sede: Cuenca, 2015.

[20] M. E. Sanchez and S. I. Mariño, “Implementación de un repositorio para apoyo a la gestión administrativa,” *Palabra Clave (La Plata)*, vol. 10, no. 2, p. e130, 2021, doi: 10.24215/18539912e130.

[21] H. Sigurdsson Houghton, B., McNutt, S., Rymer, H. y Stix, J and F. M. Wedge, “El Senado y Cámara de Diputados de la Nación Argentina reunidos en Congreso, etc. sancionan con fuerza de Ley: Ley de Protección de los Datos Personales,” *Encycl. volcanoes.*, vol. 3, p. 662, 2000.

[22] C. Wohlin, E. Mendes, K. R. Felizardo, and M. Kalinowski, “Guidelines for the search strategy to update systematic literature reviews in software engineering,” *Inf. Softw. Technol.*, vol. 127, no. January, p. 106366, 2020, doi: 10.1016/j.infsof.2020.106366.

[23] M. C. Colorado, “Plan y presupuesto integral de mantenimiento para la secretaría de movilidad de la alcadía de medellín,” 2023.

[24] M. C. Nathali Yomira, “Caracterización de metodologías de métricas de software para la optimización del desarrollo web.,” 2018.

[25] D. S. Fábrega, “Propuesta de procedimiento para el mantenimiento de software,” Universidad de las Ciencias Informáticas Tesis, 2009.

[26] C. Orozco, C. Pardo, K. Zuñiga, and S.-C. Certuche, “Proceso para fomentar y apoyar la adopción de DevOps en PyMEs de software,” *Rev. Científica*, vol. 45, no. 3, pp. 422–437, 2022, doi: 10.14483/23448350.19644.

[27] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, 2007.

[28] J. D. Erazo, A. S. Florez, and F. J. Pino, “Análisis y clasificación de atributos de mantenibilidad del software: una revisión comparativa desde el estado del arte,” *Entre Cienc. e Ing.*, vol. 10, no. 19, pp. 40–49, 2016, Accessed: Feb. 12, 2022. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1909-83672016000100006&lng=en&nrm=iso&tlng=es.

[29] O. Chiguano, A. Paola, Q.-E. Vega, A. R. Karina, and Q. -Ecuador, “Importancia del análisis FODA para la elaboración de estrategias en organizaciones americanas, una revisión de la última década.”

[30] A. P. C. Oña and R. K. A. Vega, “Importancia Del Analisis Foda Para Elaboracion De Estrategias,” p. 10, 2018.

[31] A. O. Sarli, R. Ruth, P. O. González, and S. Inés, “Análisis FODA. Una herramienta necesaria Swot analysis, a necessary tool.”

[32] M. A. Hitt, R. D. Ireland, and R. E. Hoskisson, *Administración Estratégica: Competitividad y Globalización. Conceptos y Casos*. 2008.

[33] L. Manosalvas, B. Burbano, G. Peñafiel, and J. Acurio, "Formulación de estrategias para el desarrollo empresarial de la constructora Emanuel en el cantón La Maná," 2020. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202020000400045.

[34] A. Díaz Concepción, A. del Castillo Serpa, and L. Villar Ledo, "Instrumento para evaluar el estado de la gestión de mantenimiento en plantas de bioproductos: Un caso de estudio," *Ingeniare*, vol. 25, no. 2, pp. 306–313, 2017, doi: 10.4067/S0718-33052017000200306.

[35] M. E. Sánchez, S. I. Mariño, and U. Nacional, "Una propuesta para la administración central de liquidaciones en la Universidad Methodology for the collection and analysis of information . A proposal for the central administration of settlements at the," vol. 2215.

[36] M. E. Sanchez and S. I. Mariño, "Implementación de un repositorio para apoyo a la gestión administrativa," *Palabra Clave (La Plata)*, vol. 10, no. 2, p. e130, 2021, doi: 10.24215/18539912e130.

.

Anexo1 Revisión de la literatura

En este Anexo se presenta la revisión de la literatura elaborada con la finalidad de identificar los estándares y las metodologías vinculadas con la mantenibilidad del software.

Formulación de la pregunta de investigación.

Los criterios de inclusión: metodología del estudio, participantes, intervenciones, comparaciones a estudiar y medidas de resultado. Estas características marcarán el protocolo de estudio y su correcta definición facilitará el resto del proceso.

Búsqueda de estudios en la literatura científica a través de una estrategia de búsqueda que cumpla con los requisitos que nos proponemos, con la lectura del título o el abstract y /o revisando el artículo completo seleccionamos aquellos que reúnen nuestros criterios de selección. Estos estudios constituirán nuestra revisión, de ellos se extraerán los datos necesarios y se evaluarán tanto cualitativa como cuantitativamente.

En los que exista homogeneidad entre los estudios incluidos, y al menos dos de ellos presenten datos razonablemente combinables, se realizará un análisis cuantitativo denominado “metaanálisis”, generalmente mediante la ayuda de programas estadísticos informatizados que facilitan este trabajo, y que permiten visualizar los resultados gráficamente en los denominados forest plot.

Interpretar los resultados obtenidos y extraer las correspondientes conclusiones.

Etapa 1 – **Planificación** de la revisión

El objetivo de esta etapa es definir el protocolo a llevar adelante para adecuada revisión sistemática de la literatura sobre el “Procedimiento para el mantenimiento del Software”, que consta de seis pasos que se detallan a continuación:

Objetivo

Preguntas de investigación

La cadena de búsqueda y filtros por repositorio

Fuentes de investigación

Criterios de inclusión y exclusión

En esta etapa se define el protocolo que se aplicará a los efectos de llevar a cabo una adecuada revisión sistemática de la literatura sobre el "Procedimiento para el mantenimiento del Software", que consta de seis pasos que se detallan a continuación:
Esta Revisión Sistemática de la literatura (RSL) se realiza con el objetivo de sintetizar la literatura existente sobre los procedimientos de mantenimiento de software. El conocimiento extraído de la RSL será la base para el procedimiento del mantenimiento con el fin de mejorar el desarrollo del software.
Se inicia por una selección de preguntas de investigación (PI), siguiendo con la etapa de selección de las fuentes de datos, luego con la definición del procedimiento de selección, y por último la especificación de estrategias de validación de los datos.

Preguntas de **Investigación**

Al iniciar el proceso se establece como objetivo de esta investigación Identificar trabajos relacionados con el procedimiento para el mantenimiento del software con el fin de mejorar el desarrollo de los sistemas
A partir del mismo se formula la siguiente pregunta de investigación (PI):
PI1: ¿Qué tipo de mantenimiento se utiliza?
La cual derivó a las siguientes sub-preguntas (SP):
SP1. ¿Qué metodologías existen?

PI2: Identificar los atributos que influyen en el mantenimiento del software
Seguido a esto se determinan un conjunto de palabras claves y así poder adaptarlas a las cadenas de búsqueda en los repositorios digitales seleccionados para tal fin.
Palabras claves: *Mantenibilidad, Metodología de mantenibilidad, mantenibilidad del software*
Keywords: *Maintainability, Maintainability methodology, Software Maintainability*
Se decidió utilizar palabras claves en inglés por la relevancia del tema y dado que la mayor parte de la literatura se puede encontrar en ese idioma.

Etapa 2 - **Selección** de l**as** f**uentes** de d**atos**

En esta etapa se procede a explorar las diferentes bibliotecas y base de datos científicas, entre las cuales se han seleccionado las que pueden visualizarse en la tabla Nº1 junto a su cadena de búsqueda:

Tabla 1 *Fuentes de datos y cadenas de búsqueda*

Fuentes de datos	Cadena de búsqueda
IEEE Xplorer	("mantenibilidad " OR " metodología de mantenibilidad " OR "mantenibilidad del software") AND ("Maintainability " OR "Maintainability methodology " OR "Software Maintainability")
ACM Portal	("mantenibilidad " OR " metodología de mantenibilidad " OR "mantenibilidad del software") OR ("Maintainability " OR "Maintainability methodology " OR "Software Maintainability")
ScienceDirect	("mantenibilidad " OR " metodología de mantenibilidad " OR "mantenibilidad del software") OR ("Maintainability " OR "Maintainability methodology " OR "Software Maintainability")
SciELO	("mantenibilidad " OR " metodología de mantenibilidad " OR "mantenibilidad del software") OR ("Maintainability " OR "Maintainability methodology " OR "Software Maintainability")

Procedimiento de selección

Al iniciar el proceso se aplicó las cadenas de búsquedas a las distintas bibliotecas digitales, las cuales arrojaron un número significativo de resultados.

Siguiendo los lineamientos planteados por Kitchenham y Charters[27], se aplicaron los siguientes criterios de inclusión para ir definiendo los artículos que formarán parte de nuestro lote de publicaciones:

IC1: Trabajos publicados en el período 2016-2022.

IC2: La publicación se encuentra en revistas, conferencias o workshops.

IC3: El título o resumen de la publicación describe Procedimiento para el mantenimiento del Software.

Seguido a esto se removieron los artículos duplicados que pudiera existir.

El siguiente paso fue aplicar los siguientes criterios de exclusión y así determinar los artículos que no van a formar parte de nuestra selección:

EC1: La publicación es una revisión sistemática o estado del arte respecto a los conceptos claves.

EC2: La publicación no debe ser un tutorial.

EC3: El idioma de la publicación es diferente al inglés o español.

Por último, se definieron las preguntas de validaciones para determinar los artículos primarios para su estudio

VC1: ¿Tiene propuestas para el mantenimiento del software?

VC2: ¿Propone procesos para el mantenimiento del software?

VC3: ¿Describe otros métodos para la realización de mantenibilidad del software?

Etapa 3. Reporte de los resultados

Los resultados de los estudios primarios se exponen según las preguntas de investigación planteados en la página 8.

¿Qué tipo de mantenimiento se utiliza? (PI1)

¿Qué metodologías existen? (SP1)

¿Qué atributos que influyen en el mantenimiento del software? (PI2)

2.4 Resultados

En la Tabla 2 se observa el proceso de selección de artículos dirigido a la determinación de los estudios primarios:

Tabla 2 *Proceso de selección de estudios primarios*

Fuentes	Encontrados	Analizados (*)	Cumplen ICs	No cumplen ECs	Primarios
IEEE Xplorer	3117	25	12	13	4
ACM Portal	602	6	4	2	3
Springer Link	1	1	0	1	0
Sedici	301	5	0	5	0
Scielo	4	4	2	2	2
ScienceDirect	379	10	1	9	0
Total	4404	51	19	32	9

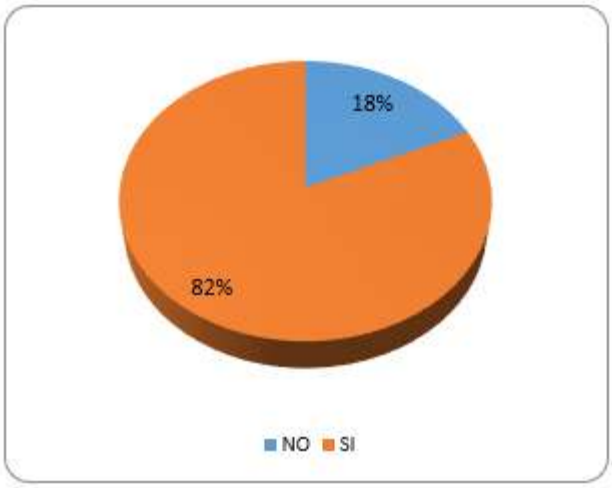
(*) se tomó como umbral 50 artículos

Resultado de artículos que se seleccionaron

Sugiero incluir la interpretación, las columnas de tabla en español. Incluir grafica o tabla…

Análisis de los resultados

Se procedió a realizar de analizar la información que se obtuvo y descartar los que no tenían relación con el tema del trabajo.



PI1: ¿Qué tipo de mantenimiento se utiliza?

Existen métricas que pueden ayudar a los desarrolladores a medir y analizar objetivamente el nivel de mantenibilidad de un proyecto. La mayoría de estas métricas implican un análisis automatizado del código

La mantenibilidad del producto de software es fundamental para lograr la calidad del producto de software. Para mantener el software útil el mayor tiempo posible, la predicción de la mantenibilidad del producto de software (SPMP) se ha convertido en un esfuerzo importante.

SP1: Existen varias metodologías utilizadas en el mantenimiento del software, cada una con enfoques y técnicas específicas. Algunas de las metodologías más comunes incluyen: Mantenimiento Correctivo, mantenimiento preventivo, mantenimiento adaptativo, mantenimiento evolutivo, mantenimiento perfectivo,

PI2: Identificar los atributos que influyen en el mantenimiento del software

En [28] se definen los atributos que influyen en el mantenimiento del software:

Acoplamiento: Se afirma que el acoplamiento en los sistemas software tiene un fuerte impacto negativo en la calidad de software y por lo tanto se debe mantener al mínimo durante la etapa de diseño. En [28] menciona que el acoplamiento debe ser considerado durante el diseño, porque se afirma que es una de las características importantes que brinda eficiencia al diseño en la orientación a objetos.

Cohesión: se define como una característica importante que ayuda a la eficiencia del diseño. En [28] se ve a la cohesión como un atributo de diseño más que de código y un atributo que puede ser usado para predecir las propiedades de implementación como "facilidad de depuración, facilidad de mantenimiento y facilidad de modificación". En [28] se afirma que cuanto menor es la cantidad de acoplamiento y la complejidad de los componentes y más alta sea la cohesión más fácil será la capacidad para ser analizado, la estabilidad y la capacidad para ser probado del producto software. En [39] indican que se ha demostrado que las métricas de la cohesión son buenos predictores de la capacidad para ser probado, al encontrar una correlación clara entre ellas.

Documentación: en [28] afirma que los documentos de diseño son una fuente de información importante, especialmente cuando los sistemas entran en la fase de mantenimiento. En [28] se indica que la documentación contribuye a capacidad para ser analizado, la capacidad para ser modificado y la reusabilidad del software. Es importante tener documentación clara de los requerimientos y especificaciones.

Estandarización: en [28] ya que este atributo implica tener un conjunto de estándares de programación definidos se evidencia que es importante considerarlo durante la etapa de implementación, además este puede mejorar la facilidad de lectura del código fuente, la facilidad de entendimiento del mismo y con esto la capacidad para ser analizado y modificado.

Facilidad de lectura: en [28] se afirma que debido a que la facilidad de lectura puede afectar la calidad del software, los programadores deben preocuparse por ella. En [40] se dice que se debe inspeccionar la facilidad de lectura del código fuente para asegurar la mantenibilidad, portabilidad y reusabilidad del software. La facilidad de lectura afecta la capacidad para ser analizado del software, ya que simplifica el trabajo de identificar las modificaciones que se requieren hacer al sistema., mejora la capacidad para ser analizado y también la capacidad para ser cambiado del mismo.

Trazabilidad: en [28] se define la trazabilidad como una característica que se debe tener en cuenta en la especificación de los requisitos del software e indican que esta área permanece como un problema ampliamente reportado debido a que no hay un análisis de las fuentes de requerimientos utilizadas en los desarrollos de software. También se señala que un diseño es considerado de mayor calidad siempre que se mantenga la trazabilidad con el código. Como el diseño representa una abstracción de la implementación, se espera que las clases en el diseño estén representadas en el código.

Anexo 2. Otros recursos para la definición del proceso

El Anexo 2 resume otros recursos utilizados para la definición del proceso propuesto en el Trabajo Final de Maestría.

Análisis FODA

"El análisis FODA se utiliza desde épocas pasadas presentando puntos a favor y puntos en contra. El mismo que ha sido estudiado desde varias perspectivas facilitando a las organizaciones adaptar esta metodología de manera sencilla y fácil como una herramienta de apoyo para el desarrollo de estrategias. Así mismo se puede mencionar que constituye una herramienta de fácil uso, simplificando en muchos casos el análisis de factores que son de importancia en cuanto a la elaboración de estrategias." [39]

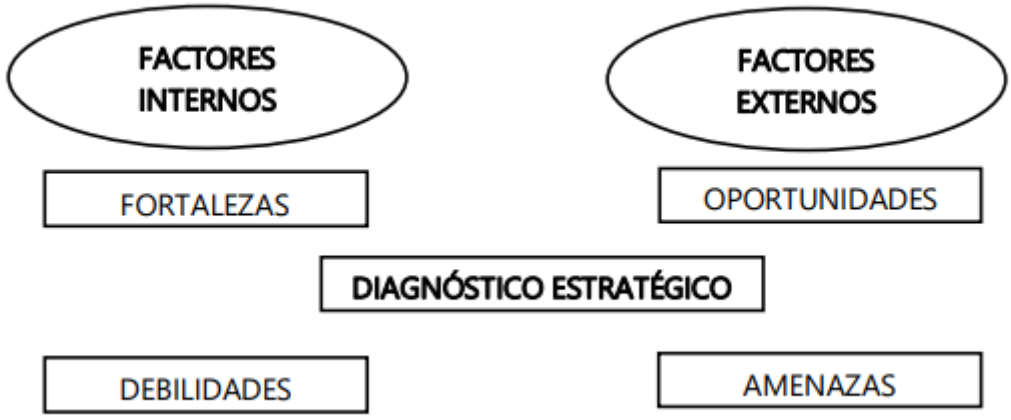


Fig. 1: Factores del análisis FODA [30]

FODA, siglas que provienen del acrónimo en inglés SWOT (strenghts, weaknesses, opportunities, threats); en español, aluden a fortalezas, oportunidades, debilidades y amenazas [31], análisis FODA consiste en realizar una evaluación de los factores fuertes y débiles que, en su conjunto, diagnostican la situación interna de una organización, así como su evaluación externa, es decir, las oportunidades y amenazas. También es una herramienta que puede considerarse sencilla y que permite obtener una perspectiva general de la situación estratégica de una organización determinada. Se dice que el análisis FODA estima el efecto que una estrategia tiene para lograr un equilibrio o ajuste entre la capacidad interna de la organización y su situación externa, esto es, las oportunidades y amenazas.

Los resultados del análisis FODA son datos muy valiosos para elegir las estrategias de la organiza. [32]

El análisis FODA es una herramienta que permite conformar un cuadro de la situación actual del objeto de estudio (persona, empresa u organización, etc.) accediendo de esta manera a un diagnóstico preciso que permite, en función de ello, tomar decisiones acordes con los objetivos y políticas formulados. La relevancia de emplear una matriz de análisis FODA es que posibilita la búsqueda y el análisis metodológico de todas las variables que intervienen en el ámbito de estudio, con el fin de tener más y mejor información al momento de tomar decisiones [31].

El análisis FODA es muy utilizada en casi todas las empresas porque les proporciona una mejor idea de cómo crear sus estrategias y así ser una organización exitosa porque este contribuye a que la empresa pueda poner en práctica sus mejores estrategias que puedan hacer que pueda crecer dentro del mercado, de igual manera ser más competitivas, ya que a través de esta se pueden mejorar aquellas debilidades que tiene la empresa, así como poder prepararse para hacerle frente aquellas amenazas que pueda tener, donde se tomaran muy en cuenta las fortalezas y oportunidades para hacer de estas una herramienta que ayuden a que la empresa pueda funcionar de la mejor manera y cumplir con metas y objetivos planteados [33].

Anexo 3. Encuesta

Una encuesta es un método empírico que se utiliza para recopilar información de personas para describir, comparar o explicar su conocimiento, sus actitudes o su comportamiento.
La encuesta ayuda a evaluar la calidad y funcionamiento del software cuyo objetivo final es evaluar un listado de preguntas que permitirá obtener una información concreta sobre las percepciones de los clientes [34]
Los parámetros para el diseño de una encuesta son:

1- Sencilla, concreta y concisa para obtener y tabular la información.
2- El objetivo es obtener información sobre el software a los desarrolladores y miembros de la PyME.

“Las encuestas son entrevistas con un gran número de personas utilizando un cuestionario prediseñado. Según el mencionado autor, el método de encuesta incluye un cuestionario estructurado que se da a los encuestados y que está diseñado para obtener información específica” [35]. El proceso de encuestas se define por una serie de actividades bien establecidas [36] y consta de las siguientes partes:

- Establecer los objetivos de la encuesta.
- Diseñar la encuesta.
- Desarrollar los cuestionarios.
- Evaluar y validar los cuestionarios.
- Obtener los datos de la encuesta.
- Reportar los resultados.

Modelo de encuesta realizada

Encuesta para evaluación de la criticidad del Sistema de Control de Tráfico

Objetivo: Conocer la experiencia y percepción de los usuarios del sistema de control de tráfico para determinar la criticidad del sistema y las necesidades de mantenimiento correctivo y documentación.

Sección 1: Información General del Usuario

1. **Cargo o rol:**
2. **Área o departamento:**
3. **Años de experiencia utilizando el sistema:**
4. **Frecuencia de uso del sistema:**
    - Diario
    - Semanal
    - Mensual
    - Rara vez

Sección 2: Importancia del Sistema

5. **¿Cuán importante es el sistema de control de tráfico en su trabajo diario?**
    - Vital: no puedo realizar mi trabajo sin él.
    - Muy importante: afecta gran parte de mis tareas.
    - Moderado: me ayuda, pero puedo trabajar parcialmente sin él.

- Bajo: solo lo utilizo ocasionalmente.

6. **¿Cómo le afecta una falla en el sistema durante su jornada laboral?**
   - Me impide realizar mis tareas por completo.
   - Dificulta mi trabajo, pero puedo continuar en menor medida.
   - Genera inconvenientes menores, pero no afecta significativamente.
   - No afecta mi trabajo.
7. **¿Qué áreas o aspectos de su trabajo dependen directamente de este sistema?**
   - Supervisión de tráfico
   - Seguridad
   - Planificación
   - Administración
   - Otros (especificar): _____

Sección 3: Experiencia con Fallas en el Sistema

8. **¿Con qué frecuencia experimenta problemas técnicos en el sistema?**
   - Diario
   - Semanal
   - Mensual
   - Trimestral
   - Rara vez o nunca
9. **¿Qué tipos de problemas experimenta con mayor frecuencia?**
   - Fallas en la conexión
   - Errores en la visualización de datos
   - Caídas del sistema
   - Problemas de configuración
   - Otros (especificar): _____
10. **Cuando ocurre un problema en el sistema, ¿qué impacto tiene en sus tareas?**
    - Muy alto: no puedo continuar.
    - Alto: necesito soporte para seguir.
    - Moderado: puedo continuar parcialmente.
    - Bajo: afecta mínimamente.

11. **¿Cuál es su nivel de preocupación en cuanto a la seguridad en caso de fallas en el sistema?**
    - Muy alto: afecta la seguridad de personas o activos.
    - Alto: puede comprometer la seguridad en situaciones críticas.
    - Moderado: tiene cierto impacto, pero no es grave.
    - Bajo: no afecta la seguridad.

Sección 4: Necesidad de Mantenimiento y Documentación

12. **¿Con qué frecuencia considera que deberían realizarse actualizaciones o mantenimientos en el sistema?**
    - Mensual
    - Trimestral
    - Semestral
    - Anual
    - Solo cuando sea necesario
13. **¿Recibe el soporte necesario cuando el sistema tiene fallas?**
    - Siempre y de forma inmediata
    - Casi siempre, pero tarda en solucionarse
    - A veces, pero demora mucho tiempo
    - Rara vez o nunca recibo soporte
14. **¿Le resulta fácil acceder a la información o documentación sobre el sistema?**
    - Sí, está bien documentado y es fácil de entender
    - Sí, pero la documentación es limitada o confusa
    - No, la documentación es insuficiente
    - No tengo acceso a ninguna documentación
15. **En caso de problemas técnicos, ¿le ayuda la documentación actual para resolverlos?**
    - Sí, es muy útil
    - Útil, pero necesita más detalles
    - Poco útil, se necesita más información
    - No es útil o no existe documentación
16. **¿Qué tan útil considera que sería mejorar la documentación del sistema para su trabajo diario?**

- Muy útil
- Útil
- Poco útil
- No útil

Sección 5: Comentarios Adicionales

17. **¿Qué sugerencias tiene para mejorar el sistema y su documentación?**

Gracias por su participación. Su experiencia y opinión son esenciales para mejorar el mantenimiento y documentación del sistema de control de tráfico.

Anexo 4. Documentaciones posibles de soluciones

El Anexo 4 presenta la documentación de las posibles soluciones que se utilizaron para el proceso propuesto en el Trabajo Final de Maestría.

Problema: Los usuarios reportan que el sistema se bloquea al intentar realizar ciertas operaciones.

Posible solución 1: Identificación y corrección de errores de programación en el código que causan el bloqueo del sistema.

Pasos a seguir: Identificación y corrección de errores de programación en el código que causan

1. Realizar un análisis exhaustivo del código relacionado con las operaciones reportadas.
2. Identificar y corregir cualquier error de programación que pueda causar el bloqueo deal sistema.
3. Realizar pruebas exhaustivas para verificar que el problema se haya solucionado y que no se hayan introducido nuevos errores.

Posible solución 2: Optimización del rendimiento del sistema para mejorar su estabilidad.

Pasos a seguir:

1. Identificar áreas del sistema que puedan estar contribuyendo al bloqueo debido a problemas de rendimiento.
2. Realizar ajustes en el código y en la configuración del sistema para mejorar su rendimiento.
3. Realizar pruebas de carga para evaluar el impacto de las mejoras en el rendimiento y verificar que el problema se haya resuelto.

Estas documentaciones proporcionan una descripción clara del problema, así como posibles soluciones y los pasos a seguir para implementarlas. Esto asegura que el equipo de desarrollo tenga una guía clara sobre cómo abordar y solucionar el problema de manera efectiva.