



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

TITULO DEL TRABAJO DE TITULACIÓN

**PROPUESTA DE UN MARCO DE TRABAJO PARA LA GESTIÓN DE
PROYECTOS DE DESARROLLO DE SOFTWARE EN LA EMPRESA
EDINA S.A.**

AUTOR

Ing. Carlos Alfredo Banguera Díaz

TRABAJO DE TITULACIÓN

**Previo a la obtención del grado académico en
MAGISTER EN TECNOLOGÍAS DE LA INFORMACIÓN**

TUTOR

Lsi. Daniel Iván Quirumbay Yagual, Msc

Santa Elena, Ecuador

Año 2023



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
TRIBUNAL DE SUSTENTACIÓN**

**ING. ALICIA ANDRADE VERA, MGTR
COORDINADORA DEL PROGRAMA**

**LSI. DANIEL QUIRUMBAY, MSC
TUTOR**

**ING. CARLOS CASTILLO, MSC
DOCENTE ESPECIALISTA**

**ING. MARJORIE CORONEL, MSC
DOCENTE ESPECIALISTA**

**AB. MARÍA RIVERA, MGTR
SECRETARIA GENERAL
UPSE**



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por BANGUERA DÍAZ CARLOS ALFREDO, como requerimiento para la obtención del título de Magister en Tecnologías de la Información.

TUTOR



Lsi. Daniel Iván Quirumbay Yagual, Msc

Santa Elena, a los 15 días del mes de agosto del año 2023



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

DECLARACIÓN DE RESPONSABILIDAD

Yo, BANGUERA DÍAZ CARLOS ALFREDO

DECLARO QUE:

El trabajo de Titulación, Propuesta de un marco de trabajo para la gestión de proyectos de desarrollo de software en la empresa Edina S.A. previo a la obtención del título en Magister en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, a los 15 días del mes de agosto del año 2023

EL AUTOR

BANGUERA DÍAZ CARLOS ALFREDO



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE CIENCIAS DE LA INGENIERÍA
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado “Propuesta de un marco de trabajo para la gestión de proyectos de desarrollo de software en la empresa Edina S.A.”, presentado por el estudiante, BANGUERA DÍAZ CARLOS ALFREDO fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 9%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

	Banguera CarlosFinalTIC	9% Similitud	9% Tercera versión actualizada de similitud antiplagio
Nombre del documento: Banguera CarlosFinalTIC.docx ID del documento: 299A2501060-84270222782097-282501060151 Tamaño del documento original: 52472 KB	Depositar: DANIEL IVÁN QUIRUMBAY YAGUAL Fecha de depósito: 14/05/2025 Tipo de carga: 012-500 Fecha de fin de análisis: 14/05/2025	Número de palabras: 15.250 Número de caracteres: 122.672	0% Industria asociada

TUTOR



**DANIEL IVÁN
QUIRUMBAY YAGUAL**

Lsi. Daniel Iván Quirumbay Yagual, Msc



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

AUTORIZACIÓN

Yo, **BANGUERA DÍAZ CARLOS ALFREDO**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 15 días del mes de agosto del año 2023

EL AUTOR

BANGUERA DÍAZ CARLOS ALFREDO

AGRADECIMIENTO

El agradecimiento muy sincero a toda mi familia por su gran apoyo y por siempre estar presentes en todos los proyectos de mi vida.

Carlos, Banguera Díaz

DEDICATORIA

Este trabajo va dedicado a toda mi familia, que siempre me ha inspirado a seguir adelante pese a las dificultades que se han presentado en mi camino a lo largo de mi carrera académica.

Carlos, Banguera Díaz

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN.....	I
TRIBUNAL DE SUSTENTACIÓN.....	II
CERTIFICACIÓN.....	III
DECLARACIÓN DE RESPONSABILIDAD.....	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
RESUMEN	XIV
ABSTRACT.....	XV
INTRODUCCIÓN.....	1
1.1. PLANTEAMIENTO DEL PROBLEMA	2
1.2. IDEA DE DEFENDER.....	3
1.3. ALCANCE.....	4
1.4. OBJETIVOS	5
1.4.1. Objetivo General.....	5
1.4.2. Objetivos Específicos	5
1.5. JUSTIFICACIÓN	5
2. CONTEXTO Y ESTADO DEL ARTE	7
2.1. ESTADO DEL ARTE	7
2.1.1. A nivel internacional.....	7
2.1.2. A nivel nacional	9

2.1.3. A nivel regional	11
3. DESARROLLO	12
3.1. MARCO TEÓRICO	12
3.1.1. Marcos de trabajo y arquitectura de software	12
3.1.2. Metodologías de desarrollo de software	14
3.1.2.1. Metodologías Tradicionales.....	14
3.1.2.2. Metodologías Ágiles.....	15
3.1.3. Metodologías ágiles más representativas.....	17
3.1.3.1. Metodología XP.....	18
3.1.3.2. Metodología Scrum.....	18
3.1.3.3. Metodología Kanban.....	20
3.2. MARCO CONCEPTUAL	21
3.3. METODOLOGÍA DE INVESTIGACIÓN	24
3.3.1. Recolección de datos	25
3.3.2. Análisis de Recolección de datos.....	26
3.4. METODOLOGÍA DE DESARROLLO	27
4. DESARROLLO ESPECÍFICO DE LA CONTRIBUCIÓN.....	29
4.1. TABLAS COMPARATIVAS	29
4.2. RESULTADOS	35
5. CONCLUSIONES	43
6. RECOMENDACIONES.....	45
BIBLIOGRAFÍA	47
ANEXOS	52

ÍNDICE DE TABLAS

Tabla 1. Cuadro comparativo metodologías tradicionales vs. ágiles.....	29
Tabla 2. Marcos de trabajo tradicionales y ágiles más representativos	32
Tabla 3. Cuadro comparativo de las tres metodologías ágiles más representativas	34

ÍNDICE DE GRÁFICOS

Gráfico 1. Ciclo de vida Scrum	36
Gráfico 2. Ciclo de desarrollo de proyecto en la Empresa Edina S.A.	37
Gráfico 3. Proceso de desarrollo de proyectos usando Scrum.....	38
Gráfico 4. Tareas del Proyecto “Desarrollo de nueva opción para agregar servicios en el Sistema de Clientes”	39
Gráfico 5. Tiempo de ejecución del Proyecto “Desarrollo de nueva opción para agregar servicios en el Sistema de Clientes”	40
Gráfico 6. Tareas del Proyecto “Desarrollo de nueva opción para agregar productos en el Sistema de Clientes”	41
Gráfico 7. Tiempo de ejecución del Proyecto “Desarrollo de nueva opción para agregar productos en el Sistema de Clientes”	42

ÍNDICE DE ANEXOS

Anexo 1: Observación para recolección de información.....	52
Anexo 2: Encuesta para recolección de información.....	53
Anexo 3: Diagrama de aceptación de propuesta de desarrollo.....	56
Anexo 4: Diagrama de marco de trabajo para el desarrollo de software.....	57

RESUMEN

Este proyecto tiene como finalidad establecer un marco de trabajo adecuado de acuerdo con las mejores prácticas del mercado de desarrollo de software con la finalidad de cumplir los parámetros de calidad, entrega y presupuestos establecidos en cada uno de los proyectos, por lo que se decide proponer el uso y aplicación del marco de trabajo Scrum para la gestión de proyectos de desarrollo de software en la empresa Edina S.A. de la ciudad de Guayaquil, para que con esta metodología seleccionada, se garantice que siempre se encuentre la excelencia en proyectos de este tipo, apoyado en la mejora de las habilidades y capacidades del equipo, teniendo la eficiencia en la entrega del producto y brindando siempre oportunidades de aprendizaje en el corto plazo, gracias a que Scrum es una herramienta muy poderosa y se debe entender que su éxito depende de la dedicación y compromiso de todos los miembros.

Palabras claves: Software, Metodología, Scrum, Framework.

ABSTRACT

The purpose of this project is to establish an appropriate framework in accordance with the best practices of the software development market in order to meet the quality, delivery and budget parameters established in each of the projects, so it is decided to propose the use and application of the Scrum framework for the management of software development projects in the company Edina S.A. of the city of Guayaquil, so that with this selected methodology, it is guaranteed that excellence is always found in projects of this type, supported by the improvement of the skills and capabilities of the team, having efficiency in the delivery of the product and always providing learning opportunities in the short term, thanks to the fact that Scrum is a very powerful tool and it must be understood that its success depends on the dedication and commitment of all members.

Keywords: Software, Methodology, Scrum, Framework.

INTRODUCCIÓN

Los proyectos de software, como cualquier otro tipo de proyecto, son complejos, así que necesitan un conjunto de patrones que permiten la gestión y ejecución. Las metodologías de desarrollo de software nacen de aplicar el conocimiento de otras disciplinas. Recientemente, están partiendo de cada concepto y características, específico de proyectos de TI y actualmente existen muchas metodologías de desarrollo de software, producto de una gran cantidad de investigaciones de expertos sobre este tema. Sin embargo, no existe una que se aplica a todos los proyectos de TI, por lo tanto, los planificadores de proyectos eligen entre muchas opciones.

Los proyectos de software están orientados a satisfacer ciertas necesidades de sistematización, automatizar procesos, mejorar la productividad de la empresa y, de manera general, obtener mejores resultados financieros y operativos en la institución a la cual está dirigida. La empresa Edina S.A. es una empresa que originalmente estaba enfocada a la elaboración de directorios telefónicos, venta de publicidad, mercadeo, entre otras cosas. Con el pasar del tiempo la empresa ha ido incorporando la tecnología en su “Core” de negocio y las tareas mencionadas en el párrafo anterior se hacen de manera digital cambiando de forma drástica las operaciones institucionales. La empresa cuenta con un departamento de Tecnología de la información y comunicación integrados por profesionales que están orientados a dar soporte a todas las operaciones de la institución y, además, implementar proyectos tecnológicos para captar nuevos clientes y mejorar los procesos operativos.

El proyecto está orientado a analizar diferentes modelos, marcos de trabajo, metodologías de desarrollo de software para identificar las mejores prácticas de la industria para ajustar un modelo o marco de trabajo a la empresa que permita estandarizar el proceso entre los participantes de cada proyecto para que, de esta manera, se pueda asegurar la entrega del proyecto en el tiempo adecuado cumpliendo los parámetros de calidad y el presupuesto planificado.

El primer capítulo pone en evidencia la problemática existente, se determinan los objetivos y la justificación con el fin de clarificar el contexto sobre el cual se va a desarrollar esta investigación. En el segundo capítulo se da a conocer la metodología de investigación y de desarrollo, además de especificar claramente la propuesta de solución de trabajo.

1.1. PLANTEAMIENTO DEL PROBLEMA

El desarrollo de software ha evolucionado desde las primeras prácticas de desarrollo, pasando por el modelo de procesos, la conceptualización de los modelos del ciclo de vida del software, la propuesta de los modelos de la ingeniería del software, hasta desembocar en la metodología misma de los procesos de la ingeniería de software actuales (Egas & Játiva, 2014). En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez que ha terminado ésta (Cadavid, Fernández, & Morales, 2013).

Edina S.A. es una empresa del sector de editores de guías de publicidad con oficinas en la ciudad de Guayaquil, provincia del Guayas, que cuenta con sucursales en otras ciudades del Ecuador, como son Quito y Cuenca. En las oficinas de la ciudad de Guayaquil, cuentan con su propio departamento de Sistemas el cual desarrolla los proyectos de Software de los clientes internos y externos de la empresa. Este departamento no tiene una metodología aplicada a sus proyectos y estos son desarrollados sin ningún control de las diferentes etapas necesarias para cumplir con éxito y entregar un buen producto al usuario final. Lo que usan es una metodología tradicional, pero basada en los conocimientos previos de cada uno de los desarrolladores sin un marco específico al cual deban regirse todos los miembros del equipo.

Se aplican varias entrevistas al equipo del departamento de sistemas y además de la técnica de observación ([ver Anexos](#)) y se determina que en el desarrollo de software no se realiza ninguna gestión de proyectos y se aplican marcos de trabajos y metodologías de forma empírica, por lo que carece de estándares de programación, lineamientos o procedimientos. Por esta forma de trabajo, las fechas de finalización previstas no se cumplen, los tiempos de desarrollo se amplían con nuevas especificaciones y requisitos, el producto entregado contiene demasiados errores cuando ya está en funcionamiento y es por esta razón que los proyectos fracasan por no satisfacer las necesidades de los usuarios.

En Ecuador, las empresas de software adoptan modelos, estándares y certificaciones para lograr competitividad internacional. Estas aplicaciones deben contar con estructuras organizacionales apropiadas, sistemas de gestión de calidad y acreditación internacional (Palacios & Merchán, 2014). Si necesitamos ser más competitivos en este tipo de mercado, lo idóneo es adoptar una metodología actual que haya sido usada en empresas similares y en las cuales se haya comprobado que sus resultados al usarla superen a los que ya se habían obtenido con la que utilizaban antes de aplicar la nueva propuesta.

El presente trabajo plantea identificar políticas y prácticas apropiadas para el campo de desarrollo de software y aprovecharlas en un enfoque o marco que aborde las necesidades de desarrollo. Para ello, es necesario definir estándares para los proyectos de desarrollo de software, especificar estilos de codificación, establecer técnicas de planificación y seguimiento, conseguir productos sin errores a la hora de ser entregados al cliente y puestos en producción sin inconvenientes futuros.

1.2. IDEA DE DEFENDER

Estandarizando el proceso de desarrollo de software con la aplicación de un marco de trabajo se obtendrán beneficios y logros importantes para la empresa Edina S.A. Dicho marco buscará establecer prácticas coherentes y consistentes en todas las etapas que dure el proceso de implementación del nuevo proyecto, garantizando que todas las personas

involucradas trabajen de manera uniforme y sigan las mismas prácticas. Esto asegura que los resultados sean predecibles y confiables mejorando la calidad del producto, gracias a estas nuevas habilidades bien definidas, reduciendo la probabilidad de errores y defectos en el producto final, optimizando la calidad del software entregado.

1.3. ALCANCE

La investigación se establece para el departamento de desarrollo de software de la empresa Edina S.A., de la ciudad de Guayaquil, provincia del Guayas. El proyecto brindará un desarrollo teórico para proponer un marco de trabajo ágil que se ajuste a las necesidades que se presentan en el departamento de desarrollo de software, teniendo en cuenta que los métodos ágiles son muy flexibles y se pueden adaptar a diferentes tipos de proyectos y entornos organizacionales. Además, un proyecto ágil puede cambiar con el tiempo a medida que surgen nuevos desafíos y se adquieren nuevos conocimientos durante el desarrollo del producto.

Esta metodología ágil analizada, deberá ser seleccionada de acuerdo a las necesidades de la empresa, basados en estudios previos similares, que nos brinden una mejor ayuda en la elección que dependerá de factores como el tamaño y la complejidad del proyecto, el grado de certeza de los requisitos, la experiencia del equipo y la idoneidad de la organización, teniendo en cuenta que no existe un método flexible que se adapte a todas las situaciones, por lo que es importante adaptar las necesidades y características específicas de los proyectos y del equipo.

Para la realización del presente proyecto se ha considerado realizar una estructura de desglose de trabajo basada en fases según lo expuesto en la guía de fundamentos para la dirección de proyectos (Institute, 2017).

- **Fase de reconocimiento:** Abordar toda la indagación preliminar, trabajar sobre la revisión bibliográfica e identificación del proceso actual para el desarrollo de software.

- **Fase de revisión:** Analizar los marcos de trabajos más utilizados que podrían servir como base para aplicar en el departamento de desarrollo fundamentado en la revisión bibliográfica.
- **Fase de estudio observacional:** Selección del marco de trabajo adecuado y aplicarlo en el sector objetivo y realizar un análisis comparativo entre lo que se tenía con la aplicación del marco de trabajo seleccionado.
- **Fase de construcción de la propuesta:** De acuerdo con los datos obtenidos en el análisis de resultados, se elaborarán los lineamientos metodológicos.

1.4. OBJETIVOS

1.4.1. Objetivo General

Establecer un marco de trabajo adecuado de acuerdo con las mejores prácticas del mercado de desarrollo de software con la finalidad de cumplir los parámetros de calidad, entrega y presupuestos establecidos en cada proyecto.

1.4.2. Objetivos Específicos

1. Analizar los marcos de trabajo de desarrollo de software para la identificación de su importancia dentro de la arquitectura de sistemas.
2. Caracterizar las diferentes metodologías de desarrollo de software que existen dentro de la ingeniería en sistemas a fin de compararlas, identificando ventajas y desventajas de cada una.
3. Analizar las metodologías ágiles más utilizadas para el desarrollo de software bajo marcos de trabajo.
4. Aplicar un marco de trabajo para el desarrollo de proyectos de software orientado a mejorar los resultados obtenidos en la empresa Edina, S.A.

1.5. JUSTIFICACIÓN

Como lo indica Torres Samuel “las Empresas Desarrolladoras de Software (EDS) llevan a cabo procesos que deben integrarse para hacer uso eficiente de los recursos empleados y culminar satisfactoriamente el software de acuerdo con los requerimientos del cliente”

(Torres Samuel, 2013). Para la implementación de nuevos productos (soluciones de software), el departamento de TICS orienta su trabajo a la elaboración y ejecución de proyectos para cumplir con las necesidades de los clientes (internos o externos).

Los proyectos que se ejecutan en la institución pueden enfrentarse a un sin número de riesgos asociados en cada una de las fases lo que podría causar retraso en el tiempo de entrega o presentar fallas en el resultado final o que el presupuesto sea superado. Los directores o gestores de proyectos deben garantizar la calidad del producto, que se entregue en el tiempo establecido y dentro del presupuesto estimado.

A pesar de los continuos avances en las tecnologías y metodologías de desarrollo de software, aún representa un desafío para las organizaciones dedicadas a esta actividad, debido a que conlleva las dificultades propias de un proyecto de ingeniería, incluyendo los particulares retos inherentes a la construcción de un producto de software como lo son los asociados a la complejidad de su implementación, la gestión de requisitos, los desafíos tecnológicos y la dificultad para asegurar su calidad (Yépez, Primera, & Torres, 2013).

El proceso de desarrollo de software requiere capacidades claves para asegurar la calidad del producto, y que a su vez representan cualidades de madurez de dicho proceso. En este sentido, la naturaleza intangible del software dificulta la visualización y medición del avance en los procesos de desarrollo, siendo vital una gestión eficiente del tiempo, costo y recursos (Yépez, Primera, & Torres, 2013).

Se realizará levantamiento de información para identificar los pasos que siguen actualmente los involucrados en el proceso de desarrollo y definir los aspectos a considerar para ajustar una metodología acorde a las necesidades de la institución. Al finalizar el proyecto se tendrá un marco de trabajo establecido el cual seguirá cada uno de los participantes que permitirá estandarizar el proceso y mejorar los resultados de los proyectos implementados.

Este trabajo demostrará que la falta del uso de modelos, marcos de trabajo, metodologías de desarrollo de software para la gestión de las etapas de dicho software, incide de manera directa en las actividades, ocasionando retrasos, productos finales con muchas fallas y errores, equipo de trabajo desmotivado por el aumento de la carga horaria que se impone para poder cumplir las metas y un excedente en el presupuesto por todos los puntos mencionados anteriormente.

Este proyecto se orienta en los siguientes ejes del Plan de Creación de Oportunidades 2021 – 2025 (Secretaría Nacional de Planificación, 2021):

Eje: Económico.

Objetivo: Impulsar un sistema económico con reglas claras que fomente el comercio exterior, turismo, atracción de inversiones y modernización del sistema financiero nacional (Plan de Creación de Oportunidades 2021 – 2025).

Política: Promover un adecuado entorno de negocios que permita la atracción de inversiones y las asociaciones público-privadas.

2. CONTEXTO Y ESTADO DEL ARTE

2.1. ESTADO DEL ARTE

2.1.1. A nivel internacional

En Perú, Alvarado & Arriola (2020) realizaron una investigación titulada “Marco de trabajo para la gestión de proyectos de desarrollo de software de la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruiz Gallo - Lambayeque 2016”, la cual formula un marco de trabajo basado en la quinta edición de la guía del PMBOK para la gestión de proyectos de software elaborados bajo Scrum en la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruiz Gallo. El estudio busca alcanzar un mejor control de cada uno de los proyectos que se desarrollan

por la mencionada facultad, proponiendo una adaptación de los procesos que se sugieren en la guía del PMBOK aplicados a la gestión de proyectos de software basados en el marco de trabajo Scrum. Los resultados mostraron la importancia de lograr un replanteamiento de los procesos de inicio, ejecución y cierre, obteniendo una alta probabilidad de éxito no solo en la creación, sino también en la puesta en marcha de proyectos desarrollados por la facultad. El estudio aporta las ventajas de trabajar con una metodología Scrum de desarrollo de software que permita controlar mejor los proyectos que se desarrollan y poder replantear los procesos que se vienen ejecutando.

En Argentina, Dousdebes (2017), realizó un proyecto de grado titulado “Marco de Trabajo para el desarrollo de Software” el cual resalta la importancia de utilizar metodologías de desarrollo de software tanto ágiles como tradicionales, dependiendo el caso. El principal objetivo de la investigación es proporcionar un marco de trabajo balanceado utilizando técnicas ágiles y tradicionales. Dicho marco integra los actores del desarrollo de un proyecto de software, permitiendo escalabilidad, y aportando la probabilidad de poder manejar equipos distribuidos. El marco de trabajo que se pretende es uno que sea eficiente dentro del complejo entorno moderno, tratando de abarcar una gran diversidad de proyectos en los cuales se pueda ser efectivo; con el objeto de optimizar la definición del proyecto y la previsibilidad de la entrega. La investigación aporta formas de enfrentar las etapas del desarrollo de software; especificando artefactos relacionados con el manejo de tareas y procedimientos; suministrando lineamientos generales sobre los cuales se debe actuar.

En México, Matla (2014) realizó una investigación titulada “Desarrollo de software guiado por la Norma ISO/IEC 29110 y Scrum SIDEP V.2.0.”, cuyo alcance es analizar los fallos que presenta el sistema Sidep v.1.0 que utiliza la División de Estudios de Posgrado de la Facultad de Medicina y proponer soluciones. El proyecto se guio por metodologías ágiles y tradicionales, fundamentado principalmente en SCRUM y la Norma ISO/IEC 29110 para las metodologías correspondientes; dando origen a un proceso de desarrollo de software que involucra habilidades de las diferentes metodologías que permita la generación de soluciones de la División. El estudio se

justifica por el hecho de que Sidep v.1.0 presentó fallos desde su liberación relacionados con la integridad y consistencia de datos que generaba pérdida y recaptura de información no almacenada y la manipulación de la base de datos utilizando sentencias SQL. El estudio aporta un entendimiento de las diversas metodologías que existen para el desarrollo de software y las que serían más adecuadas para el problema detectado en la División objeto de estudio.

2.1.2. A nivel nacional

En Ecuador, López (2021) realizó una tesis de maestría titulada “Propuesta metodológica de desarrollo ágil de software con énfasis en la seguridad” (López, 2021), con el objetivo de diseñar una metodología de desarrollo ágil de software que añada el atributo de seguridad, a través de técnicas de seguridad informáticas que permitan disminuir las vulnerabilidades de los sistemas frente a ataques informáticos. La investigación compara las diferentes metodologías ágiles de desarrollo de software, diseña la metodología y valida la propuesta metodológica. Para la obtención de información se realizó una revisión sistemática de Literatura SLR y se planteó un estudio de caso; aplicando entrevistas al equipo de desarrollo de la empresa SOLNUS, ubicada en la ciudad de Loja. Para la validación de la información se utilizó la técnica de chi cuadrado, la cual ayudó a comprobar que la propuesta podía minimizar las vulnerabilidades ante ataques tanto internos como externos. El estudio aporta información teórica sobre metodologías ágiles de desarrollo de software y marca un precedente de cómo éstas pueden emplearse en empresas ecuatorianas.

Por su parte, Egas (2019) presenta una tesis de maestría titulada “Análisis de metodologías ágiles para la estandarización de procesos de gestión de proyectos de software en el departamento de desarrollo de sistemas de la Dirección de Tecnologías de Información y Comunicaciones de la Fuerza Aérea Ecuatoriana”, con el propósito de desarrollar una propuesta para la estandarización de procesos de gestión de proyectos de software en el mencionado departamento (Egas L. , 2019). Establece un procedimiento para la determinación de la metodología que mejor se adapta al Departamento, iniciando con el conocimiento de la orientación de desarrollo de software tradicional o ágil, que,

mediante la aplicación de una evaluación cuantitativa, verificando del cumplimiento de los principios ágiles, encontró la existencia de esta agilidad. Luego de esto, se efectuó un estudio de las metodologías más representativas de desarrollo ágil y posteriormente, mediante la aplicación de un framework del autor Iacovelli, se seleccionó la metodología SCRUM. Luego, se siguió con la adaptación de dicha metodología, determinándose el nivel de estandarización de procesos alcanzado. Para la verificación de la propuesta, se tomó un caso de estudio: el proyecto “Sistema Automatizado de Importación y Exportación (SAIMEX)”, en el cual se obtuvo efectos positivos en lo que respecta a la planificación, velocidad, sinergia del equipo de trabajo, calidad del producto y satisfacción del cliente. Para la validación de la propuesta, se utilizaron evaluaciones cuantitativas aplicadas al director de la Dirección de Tecnologías de la Información y Comunicaciones, al equipo de trabajo del Departamento Desarrollo de Sistemas y a expertos externos, cuyo resultado fue la aprobación de la misma. El estudio aporta información sobre las metodologías ágiles más representativas y los beneficios de escoger la más adecuada a las necesidades de la empresa o institución.

Por último, en Quito Cárdenas (2018) realizó el estudio “Mejoramiento de los procesos de desarrollo de software para la empresa HSB SOFTECUADOR C.A.”, con el objetivo de mejorar los procesos de desarrollo de software de la empresa HSB SOFTECUADOR CA (Cárdenas, 2018), para que pueda generar sus productos eficaz y eficientemente. Se inicia a partir del análisis situacional de la empresa para identificar los aspectos negativos y positivos como impedimento o apoyo para la ejecución de nuevas metodologías. Posterior a ello, se consideraron teóricamente los fundamentos de desarrollo de software y las metodologías que pueden ser utilizadas de acuerdo con el tamaño y necesidades de la empresa; sirviendo de base para generar la propuesta de mejora de los procesos de la empresa con énfasis en metodologías ágiles de desarrollo de software (Cárdenas, 2018). Luego se determinó la situación actual de la empresa, utilizando diagramas de flujo y listas de verificación, a fin de identificar las brechas existentes entre la metodología actual y la metodología ágil propuesta. Finalmente, se cuenta con el plan de implementación en el que se determina si la propuesta es ideal para la empresa. El estudio aporta diferencias entre las metodologías tradicionales de cascada y las metodologías modernas ágiles, así como los beneficios que cada una de ellas reportan.

2.1.3. A nivel regional

En Guayaquil, Magallanes (2021) realizó una tesis denominada “Análisis de las metodologías ágiles y su incidencia en el desarrollo de software” (Magallanes, 2021); con el propósito determinar cuál es la metodología ágil más adecuada para el desarrollo de software, puesto que se ha evidenciado que la falencia en el uso de metodologías tradicionales principalmente se debe al uso extensivo de procesos y documentos, por lo que frecuentemente las especificaciones declaradas por el cliente no se cumplen, generando insatisfacción al cliente. Se analizaron tres modelos ágiles: Scrum, Kanban y Extreme Programming, mediante la recopilación de información, obteniendo características relevantes sobre cada una; luego se realizó un análisis de estudio de casos sobre la implementación de los mencionados modelos ágiles. El estudio aporta herramientas teóricas de comparación sobre las distintas metodologías, permitiendo precisar cuál de las metodologías es la más adecuada para las empresas.

Adicionalmente, Caregua (2021) realizó una tesis denominada “Implementación de metodología ágil en la gestión de proyectos en una fábrica de software en la ciudad de Guayaquil”, la cual estuvo motivada por quejas que los clientes han presentado debido a atrasos y falta de comunicación, generando la necesidad del empleo de nuevas prácticas que contribuyan a la mejora de la gestión de proyectos en la fábrica. El presente trabajo propone la implementación de una metodología ágil que ayude en el conocimiento del estado de las actividades, a fin de ejecutar acciones correctivas o preventivas durante el proyecto. Se analizan tres metodologías ágiles de forma comparativa, para luego seleccionar una de ellas y aplicarla en un proyecto piloto. La investigación aporta herramientas teóricas sobre diversos tipos de metodologías ágiles, así como las ventajas que estas tienen sobre las tradicionales, en especial para la elaboración de proyectos.

Por otra parte, Coque (2017) realizó un trabajo de maestría en Guayaquil titulado “Modelo de adopción de metodologías ágiles en mi PYME desarrolladora de software” (Coque, 2017). El trabajo hace referencia a que las metodologías ágiles vienen teniendo gran aceptación en el área de sistemas, específicamente en el desarrollo de software. El objetivo de este trabajo es crear un modelo de adopción de metodologías ágiles,

fundamentado en teorías y modelos de adopción de tecnologías; analizando una serie de factores que constituyen dichos modelos. Adicionalmente, se describen afirmaciones para cada factor que compone el modelo propuesto, y se realiza una evaluación del mismo, el cual cuenta con la participación de desarrolladores de software. Los resultados arrojan una fuerte interrelación entre los factores que se proponen, así como una alta consistencia de las afirmaciones. El estudio aporta información teórica y aplicada sobre las metodologías ágiles de desarrollo de software, especificando sus características y ventajas en microempresas y en pequeñas y medianas empresas ecuatorianas.

3. DESARROLLO

3.1. MARCO TEÓRICO

3.1.1. Marcos de trabajo y arquitectura de software

La arquitectura de software describe las partes que conforman un sistema y la interacción que existe entre ellas, así como también, los patrones utilizados para su composición y las restricciones impuestas al uso de esos patrones (Martínez, 2015). Por su parte, la aplicación de marcos de trabajo favorece la evaluación objetiva de los procesos y de los productos, disminuyendo el esfuerzo que se dedica a la corrección de fallas o defectos mediante la detección de estos en la fase más cercana de donde nacen (Marin et al, 2018). Ello permite inferir que la arquitectura de software se desarrolla a partir de marcos de trabajo.

En tal sentido, los marcos de trabajo constituyen un conjunto de clases e interfaces que hacen posible el encapsulamiento de un diseño que pudiera ser utilizado para ciertos problemas y sistemas que se relacionan. Dentro de los marcos de trabajo se especifica la forma en que interactúan sus instancias. Por lo tanto, un marco de trabajo es un diseño reutilizable de un sistema de manera total o parcial. La principal ventaja de la utilización de marcos de trabajo es la reducción de costos y de complejidad y la reutilización del software que se está desarrollando (Fayad & Schmidt, 1997).

De acuerdo con Martínez (2015), en un marco de trabajo pueden ser definidos dos niveles de abstracción: la arquitectura, por un lado, y la implementación del marco por el otro. Otra denominación que se le ha dado a los marcos de trabajo es arquitectura de software específica de un dominio. De modo que, se pasa de una descripción de arquitectura a un marco de trabajo en el momento en que tal arquitectura se utiliza para un sistema de dominio específico. Con base en ello, un marco de trabajo puede ser definido también como un conjunto de interfaces y clases cooperativas que arman un diseño reutilizable para un software específico. Para el diseño del marco de trabajo se utilizan clases abstractas, definiendo sus responsabilidades, normalmente a través del uso de interfaces. En síntesis, un marco de trabajo es una estructura definida de soporte, a partir de la cual se pueden planificar, organizar y desarrollar proyectos de software.

En general, los marcos de trabajo incluyen un soporte para desarrollo de programas, unas bibliotecas de componentes, la posibilidad de incluir lenguaje scripting, un software de desarrollo y unión de diferentes componentes del proyecto de desarrollo de software. Además, una de las ventajas de construir el software a través de marcos de trabajo es que facilitan el desarrollo del mismo debido a que proporcionan códigos que no tendrían que reescribirse; evitan los detalles de bajo nivel ayudando a la identificación de los requerimientos de software; minimizan los tiempos que se emplean en el desarrollo; y construyen una arquitectura con características consistente entre aplicaciones (Martínez, 2015).

Por otra parte, los marcos de trabajo pueden desarrollarse haciendo uso de patrones de diseño, es decir de una arquitectura que brinda soluciones a problemas específicos dentro de la arquitectura general de un sistema. Tales patrones también sirven para documentar el diseño que se está realizando. (Gamma et al, 1995) Además, los marcos de trabajo se utilizan para el desarrollo eficaz de un software que permite la reusabilidad del código fuente, mayor productividad y flexibilidad, mejor grado de estandarización, homogenización, más organización del trabajo de todos los miembros involucrados en el proyecto y agilización en el desarrollo del software (Anglada & Garófalo, 2013).

3.1.2. Metodologías de desarrollo de software

Como ya se explicó en el apartado anterior, las arquitecturas en el desarrollo de software cada día cobran más importancia sobre todo por su simplicidad en la implementación; siendo estas capaces de priorizar requisitos, comunicación efectiva con el cliente y cooperación entre los distintos desarrolladores. Teniendo esto en mente, para el desarrollo de software existe una gran cantidad de metodologías que inciden de distintas maneras y dimensiones durante el proceso; estas van desde metodologías tradicionales iniciales y predictivas, hasta metodologías ágiles actuales, las cuales se han impuesto favorablemente en el desarrollo y ejecución de proyectos de software.

Las metodologías de desarrollo de software han logrado la mejora significativa del producto de software a través de una serie de fases o procesos que han sido calificados como efectivos, y que promueven la calidad del producto final. En el desarrollo de sistemas o software informáticos se hace necesaria la administración, la planificación, el seguimiento, el control del grupo que participa en el desarrollo del trabajo, así como los procesos que recopilan y analizan los requisitos del sistema. (Molina & Zea, 2017). Pero para que sean considerados como ágiles no debe quedar de lado la comunicación con el cliente y la cooperación entre los desarrolladores.

De acuerdo con Fowler (2005) las metodologías ágiles y las tradicionales difieren en dos aspectos fundamentales; el primero es que las ágiles son adaptativas y no predictivas, en cambio las tradicionales sí son de predictivas; el segundo es que las metodologías ágiles se orientan hacia las personas, pero las tradicionales van orientadas a los procesos. Para conocer más de las mencionadas metodologías de desarrollo de software, a continuación se explican los aspectos más resaltantes de cada una de ellas.

3.1.2.1. Metodologías Tradicionales.

Las metodologías tradicionales tienen como propósito imponer disciplina a los procesos de desarrollo de software, volviéndolo eficiente y predecible; para ello se soportan en un proceso bien detallado basado en la planeación. Uno de los problemas primordiales de

este enfoque es la cantidad de actividades por hacer para seguir una metodología estricta, lo cual retrasa la etapa de ejecución o desarrollo del producto (Fowler, 2005). En las metodologías tradicionales es concebido un solo proyecto, con una estructura bien definida y de grandes dimensiones. Se caracteriza por seguir un proceso secuencial que va en una sola dirección y no tiene marcha atrás; además su proceso no cambia con facilidad pues es rígido; los requerimientos se acuerdan desde el inicio para todo el proyecto. Esta metodología demanda plazos grandes de planeación y muy poca comunicación con el cliente (Khurana & Sohal, 2011).

En tal sentido, las metodologías tradicionales de desarrollo de software se orientan principalmente en la planeación, documentación y alta cantidad de requerimientos, previo a la fase de análisis y diseño. Siempre inician el desarrollo de un proyecto con un proceso que es bien riguroso y estricto, controlado por muchas reglas, con metodología rígida y resistencia al cambio; con lo cual tratan de asegurar resultados de calidad y oportunos.

3.1.2.2. Metodologías Ágiles.

En el mundo del desarrollo de software, el concepto de “Ágil” está bien extendido. Se trata de una metodología de gestión de proyectos en la que el trabajo es adaptado dependiendo a las expectativas del usuario y a las necesidades del cliente, consiguiendo inmediatez y flexibilidad al momento de adaptar el proyecto de desarrollo de software. Además de ello, el método trata de realizar transformaciones de los entornos de trabajo para lograr que los profesionales se mantengan motivados, pero también comprometidos en la realización de sus labores (Digital Talent Agency, 2018).

De acuerdo con Canos & Letelier (2003), las metodologías ágiles son adaptativas, permitiendo ejecutar proyectos de desarrollo de software que se adapten a los cambios, como una importante oportunidad de mejoras para el sistema e incremento de la satisfacción del cliente. En este tipo de metodologías, se consideran los cambios como un aspecto inherente al proceso de desarrollo, logrando con ello una mejor adaptación al entorno, una maximización de la inversión, una reducción de los costos, entre otras cosas;

bien sea para cambiar parte de sus funciones, añadir nuevas, o para adaptarlo a un nuevo dominio de aplicación. Las metodologías ágiles son una solución simplificada de desarrollo de software, que no renuncia a las prácticas implementadas para el aseguramiento de la calidad del producto.

De modo que, para la ejecución exitosa de las metodologías ágiles, es de suma importancia tener una comprensión de los requisitos del cliente, a fin de extraer al máximo sus deseos y así poder entregar un producto de calidad lo más cercano a la realidad del usuario. Estas metodologías, al ser flexibles y colaborativas, se ajustan con facilidad a los cambios en requisitos, y permiten entregas constantes al cliente, así como una retroalimentación. Esto significa que el producto y los procesos pueden ser mejorados de manera frecuente, pudiendo realizar, además, entregas continuas en plazos breves; lo cual permite que el cliente pueda verificar en el sitio el desarrollo del proyecto y entender su funcionalidad (Navarro et al, 2016).

En las metodologías ágiles del desarrollo de software, se trata de sacrificar el tiempo que se invierte en documentación innecesaria, para utilizarlo en la fase de desarrollo del producto; intentando siempre no generar documentación superflua que no se necesitará durante el proceso. Esto significa que solo debe generarse documentación primordial y necesaria para la toma de decisiones durante el proceso de desarrollo. En definitiva, más que la cantidad de documentación, lo importante es desarrollar un software completo, estable y de calidad (Digital Talent Agency, 2018).

Por otra parte, según Kerievsky (2004), el principal objetivo de utilizar metodologías ágiles es la minimización de riesgos. Para poder lograr esto los desarrollos que se van realizando se deben ir liberando en periodos de tiempo cortos. De manera tal que, en cada interacción, se añadan pocas funcionalidades y se haga más énfasis en la calidad del código. Usualmente, en este tipo de metodología se emplean pruebas unitarias, técnicas de refactorización e integración continua para control, seguimiento y garantía de calidad del producto en desarrollo.

En síntesis, esta metodología surge como respuesta a las necesidades que las metodologías tradicionales no eran capaces de cubrir. Algunos de los problemas que buscan solucionar las metodologías ágiles son el de los retrasos que existían en las decisiones y el de la planificación adaptativa. Cabe acotar que, aunque estas metodologías iniciaron debido a las necesidades encontradas a la hora de gestionar proyectos de sistemas, muchas empresas ya han empezado a utilizar el enfoque ágil de muchas otras formas, aplicando la metodología en sus procesos de creación, logística, desarrollo, entre otros; con la finalidad de convertirse en empresas más funcionales y eficaces, adaptándose a las nuevas expectativas y necesidades de los usuarios y al mercado. Además de lo anterior, en la metodología ágil los equipos de proyectos son multidisciplinarios, organizándose de acuerdo a las necesidades de cada uno, aunado al hecho de que cada etapa del proyecto se ejecuta por un equipo diferente, donde la comunicación es un aspecto esencial.

3.1.3. Metodologías ágiles más representativas

Las metodologías ágiles según Molina & Zea (2017), son técnicas empleadas para la gestión de proyectos de desarrollo de sistemas que surgieron como contraposición a las metodologías clásicas de gestión. Los principios de este tipo de metodologías se agrupan en cuatro valores, a saber, el recurso humano y su interacción, son más importantes que los procesos y herramientas; software que trabaja por encima de la exhaustiva documentación; colaboración y trabajo con el cliente, por encima de los tratos; y la flexibilidad y adaptación al cambio, por encima de la estricta planeación inicial (Molina & Zea, 2017).

Actualmente, las metodologías ágiles han ganado importancia en el área de la ingeniería en sistemas y desarrollo de software, debido a que han demostrado ser más eficientes en entornos de incertidumbre y constantes cambios, en comparación con las tradicionales. La definición de ágil va enfocada hacia el hecho de que los requisitos y soluciones van evolucionando con el paso del tiempo, de acuerdo a las necesidades del proyecto y de los usuarios finales. Teniendo esto en cuenta, y, conociendo ya las principales diferencias entre marcos de trabajo tradicionales y ágiles, abordados con anterioridad, se procede a

estudiar las metodologías ágiles más representativas, a saber, XP, Scrum y Kanban; desde una perspectiva teórica.

3.1.3.1. Metodología XP

La metodología de Programación Extrema conocida comúnmente como XP, se fundamenta en principios ágiles; y consiste en reuniones para realizar una lista de tareas que se llevan a cabo y tienen su culminación en la próxima iteración que por lo general no dura más de un mes. En esta metodología se involucra tanto al propietario del producto como al equipo de desarrolladores (Beck, 2000). Como complemento, puede decirse que la metodología XP busca el manejo de ciertos requerimientos de alto nivel y la identificación de un usuario experto, fundado en pruebas de aceptación de usuario, en pruebas unitarias de código, y en el desarrollo dirigido de pruebas (Carnaghan, 2014; Lee, 2006).

En este tipo de metodología de Programación Extrema, los desarrolladores muestran poco interés hacia las necesidades de los usuarios finales, presentándose falta de comprensión y una mala interpretación; ello conduce a la incapacidad de conexión con los requerimientos de los usuarios finales o clientes. Esto genera problemas de comunicación en áreas como: la necesidad o requerimiento de usabilidad ágil y el diseño fundamentado en escenarios para el extremo (Lee, 2006).

En síntesis, la programación extrema (XP) es la metodología ágil más representativa y utilizada del desarrollo de software, cuyo objetivo principal es el proceso ágil usado en organizaciones grandes. XP utiliza un enfoque orientado a objetos y engloba reglas y prácticas que se desarrollan en el contexto de cuatro actividades estructurales, como lo son: la planeación, el diseño, la codificación y las pruebas.

3.1.3.2. Metodología Scrum

Scrum es un término aplicable al deporte, inspirado en la jugada en formación del rugby; su primera referencia se registra en el año 1986. El enfoque incremental Scrum tiene como fundamento la teoría de control de procesos y sus equipos trabajan en iteraciones, siendo

autogestionados y multifuncionales. Esta metodología define tres roles, el primero es el “Scrum master”, que es el líder del equipo; el segundo se denomina “dueño del producto” representado por los interesados que velan siempre por la maximización del valor del entregable; y el tercero es “el equipo de desarrollo” quienes son los responsables de transformar los requerimientos del cliente en iteraciones del producto que sean funcionales (Cadavid et al. 2013).

Scrum surge como una alternativa para atender proyectos complejos y cambiantes, donde se definen equipos de trabajo, estrategias y roles para controlar el proyecto, coordinar las actividades y dar seguimiento a los avances (Kniberg, 2011). Es normal que al inicio de un proyecto queden lagunas y dudas por resolver relacionadas con las especificaciones del proyecto, y que surjan problemas que afectan la operación, y que conlleven a la toma de decisiones sobre la marcha, no obstante, esto puede atenderse ya que los proyectos ágiles como el scrum, están creados para adaptarse a las necesidades continuas (Scrum Manager, 2014).

En este mismo orden de ideas, Díaz y Del Dago (2008) definen la metodología Scrum como una agrupación de procesos utilizados para gestionar proyectos de sistemas, que permiten tener como punto central la entrega de valor para el cliente y la potencialidad del equipo con la finalidad de alcanzar eficiencia máxima, todo ello dentro del esquema de mejora continua. Con esta metodología, se busca alcanzar la satisfacción del cliente o usuario final, ya que es capaz de adaptarse a los cambios.

Por su parte, Deemer et al (2009), explica que la metodología scrum se encuentra estructurada por ciclos de trabajo llamados Sprints; los cuales representan iteraciones que se dan de manera consecutiva, hasta un máximo de cuatro semanas. Cuando inicia un Sprint, el equipo scrum escoge los requisitos del cliente, comprometiéndose a terminar los elementos. Luego, durante el Sprint no se deben cambiar los elementos elegidos inicialmente. Después, al final del Sprint, todo el equipo multidisciplinar revisa el proyecto junto con los interesados, con la finalidad de que el equipo reciba observaciones

para el siguiente Sprint. En este mismo orden de ideas, cada sprint corresponde a los siguientes elementos: reunión de planificación, revisión diaria, trabajo de desarrollo, revisión de sprint y retrospectiva de sprint. El término también se refiere a "Artefactos de Scrum", que son los subproductos de las actividades metodológicas que brindan dirección y transparencia al equipo, tales como: Product Backlog, Sprint Backlog, Monitoreo de Progreso e Incremento (DEEMER Pete, 2009).

Por último, es importante mencionar que el marco de trabajo Scrum es adecuado cuando se trata de desarrollar productos complejos, altamente adaptativos, de máximo valor y muy creativos. Aplica a equipos auto-organizados y bien disciplinados donde los productos a elaborar son representados en una lista de producto gestionada por el cliente, a partir de allí, los equipos Scrum seleccionan tareas que deben realizar en un periodo de tiempo ya mencionado conocido como "Sprint", durante ese tiempo se realizan ciclos diarios de trabajo; una vez evaluados los avances, los planes y los problemas, el trabajo es entregado, produciéndose luego una retroalimentación por parte del cliente.

3.1.3.3. Metodología Kanban

Kanban es un término japonés empleado por Taiichi Onho (Toyota), para hacer referencia al sistema de visualización utilizado en los procesos de desarrollo, el cual coordina, en una cadena de montaje, las entregas oportunas de cada parte evitando almacenamiento de producto innecesario y sobreproducción. Esta metodología basada en un enfoque ágil, se refiere a técnicas de representación visual de información que contribuyen a la mejora de la eficiencia de las tareas ejecutadas en un proyecto. Kanban se basa en las siguientes reglas: visualización del flujo de trabajo, determinación del límite de trabajo que se está ejecutando y medición del tiempo de término de una tarea (Kupiainen et al, 2015).

Kanban efectúa el rol de enfoque ágil, pero a su vez de herramienta, siendo su objetivo principal cumplir un conjunto de reglas previamente definidas, tales reglas según Sugimori et al (1977) son: visualizar el flujo de trabajo, limitar el trabajo en proceso, definir políticas bien explicadas y medir el flujo de trabajo hacia una mejora continua.

Como posee características propias de una herramienta, es fácil de implementar, lo único que se necesita es entender bien su funcionamiento y estar seguros de que todos los miembros que conforman el equipo de trabajo lo apliquen de manera correcta.

Kanban se presenta como una metodología con enfoque ágil que además de establecer propiedades, también brinda una herramienta a través de la cual se puede mejorar a gran escala la visualización del flujo de trabajo en cada uno de los proyectos. Ello permite que la metodología Kanban, al igual que la de Scrum, sea bastante sencilla en lo que respecta a su implementación y a la integración con otros enfoques. No obstante, un posible problema en su implantación radica en la producción de cuellos de botella durante su aplicación, generando deficiencia para su desarrollo (Gaete et al, 2021).

El principal problema de Kanban está asociado con el WIP o trabajo en proceso, pues al ser aplicado como lo plantea Kanban, se pueden generar cuellos de botella durante el desarrollo. Por ejemplo, en un tablero de Kanban, a cada una de las columnas se le asigna un número que simboliza la máxima cantidad de tareas que forman dicha columna; si la columna se llena, no podrán recibirse nuevas tareas hasta que alguna de las tareas en ejecución finalice (Matthies, 2018; Kupiainen et al, 2015). Esto significaría un gran problema para el proyecto debido a que, si una de las columnas de la tabla se llena, la anterior no podrá entregar sus tareas finalizadas, generando tareas inactivas e ineficientes para el proyecto.

3.2. MARCO CONCEPTUAL

Software: “Son instrucciones que proporcionan las características, funciones y desempeño buscado cuando se ejecutan; también se definen como estructuras de datos que permiten que un programa funcione correctamente con información, e información descriptiva en papel y formularios virtuales que describen el funcionamiento y uso de un programa.” (Pressman, Ingeniería del software. Un enfoque práctico., 2010).

Software de computadora: “Es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquier medio electrónico” (Pressman, Ingeniería del software. Un enfoque práctico., 2010).

Software de sistema: “Se llama Software de Sistema o Software de Base al conjunto de programas que sirven para interactuar con el sistema, confiriendo control sobre el hardware, además de dar soporte a otros programas que administran los recursos y proporcionan una interfaz de uso” (Maida & Pacienza, 2015).

Software de programación: “Es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación, de una manera práctica” (Maida & Pacienza, 2015).

Software de aplicación: “Son los programas diseñados para o por los usuarios para facilitar la realización de tareas específicas en la computadora, como pueden ser las aplicaciones ofimáticas u otros tipos de software especializados” (Maida & Pacienza, 2015).

Ingeniería de software: "La ingeniería del software surge a partir de las ingenierías de sistemas y de hardware, y considera tres elementos claves que son: los métodos, las herramientas y los procedimientos que facilitan el control del proceso de desarrollo de software y brinda a los desarrolladores las bases de la calidad de una forma productiva" (Pressman, Ingeniería del Software: un enfoque práctico, 1990).

“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza” (Sommerville, 2005).

Metodología de desarrollo de software: “Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información” (Hughes, Fisher, & Mc Daniel, 2010). “Versión ampliada del ciclo de vida completo del desarrollo de sistemas, que incluyen tareas o pasos para cada fase, funciones desempeñadas en cada tarea, productos resultantes, normas de calidad y técnicas de desarrollo que se utilizan en cada tarea” (Whitten, Bentley, & Barlow, 2003).

Metodologías tradicionales: “Centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto” (Maida & Pacienza, 2015).

Metodologías ágiles: “Son métodos de desarrollo de software en los que las necesidades y soluciones evolucionan a través de una colaboración estrecha entre equipos multidisciplinarios. Se caracterizan por enfatizar la comunicación frente a la documentación, por el desarrollo evolutivo y por su flexibilidad” (Bioul, Escobar, Alvarez, Nardin, & Aparicio, 2010).

Manifiesto ágil: Principios y valores presentados que establecen un marco definiendo características generales de roles y personas, la forma de afrontar el trabajo y la manera en la cual se manejan los artefactos (Dousdebes, 2017).

Scrum: “Es un marco de trabajo ágil destinado a equipos de desarrollo pequeños (menos de 9 personas), enfocado en proyectos complejos con requerimientos aparentemente ambiguos, donde sea necesaria la comunicación constante con los stakeholders, con el

objetivo de proporcionar entregas de software funcional entre periodos cortos y fijos, llamados sprints” (Dousdebes, 2017).

3.3. METODOLOGÍA DE INVESTIGACIÓN

La investigación se establece para la empresa Edina S.A., de la ciudad de Guayaquil, provincia del Guayas. Es un grupo de visionarios empresarios fundada en 1973 y se constituye en la primera empresa nacional que produce las Guías Telefónicas para todo el País con personal netamente ecuatoriano. Son consideradas como las Páginas Amarillas del Ecuador y cuenta con un departamento de desarrollo de software.

(Sánchez, 1999, pág. 57) En las ciencias sociales, los estudios descriptivos se proponen realizar esencialmente una medición precisa de una o más variables en alguna población definida o en una muestra de dicha población. La descripción es, pues un discurso que evidencia y significa el ser de una realidad a través de sus partes, sus rasgos estructurales, sus cualidades, sus propiedades, sus caracteres estructurales o sus circunstancias.

En este sentido, se dice que la investigación descriptiva viene a ser un proceso inicial y preparatorio de una investigación, pues en la medida que el fenómeno a estudiar forma un sistema complejo y muy amplio, la misma nos permite acotarlo, ordenarlo, caracterizarlo y clasificarlo, es decir hacer una descripción del fenómeno lo más precisa y exacta que sea posible (Tinto Arandes, 2013).

En las investigaciones correlacionales, podemos decir que también estamos relacionando variables y no solo describiendo. En el caso de todas las variables se realiza tan solo una recogida de información para medirlas, pero en ningún caso se las manipula (Argibay, 2009). Los métodos inductivos están generalmente asociados con la investigación cualitativa mientras que el método deductivo está asociado frecuentemente con la investigación cuantitativa (Pita Fernández, 2002).

En este trabajo se usará una metodología de investigación cuantitativa, porque al usar un método deductivo, este va de la mano con este tipo de investigación, ya que contribuye a la formulación de los problemas investigativos gracias a la elaboración de los aspectos teóricos bibliográficos.

3.3.1. Recolección de datos

El grupo poblacional involucrado para la recolección de datos y poder realizar de esta manera levantamiento de información necesaria, está centrado en el departamento de Sistemas que cuenta con el gerente de sistemas, jefe de proyectos y tres desarrolladores. La observación es una herramienta fundamental en el proceso de investigación; sirve para que el investigador obtenga un número mayor de datos válidos que luego serán analizados. Gran parte del acervo de conocimientos que constituye la ciencia ha sido lograda mediante la observación (Díaz, 2011). También se realiza la entrevista, que presenta dos características básicas que la distinguen del resto de los métodos de captura de información: recoge información proporcionada verbalmente o por escrito por un informante mediante un cuestionario estructurado y utiliza muestras de la población objeto de estudio (Danissa, 2020).

La observación puede proporcionar datos más precisos y confiables porque el observador puede registrar objetivamente lo que sucede sin la interpretación subjetiva de un tercero, por lo tanto, puede capturar el comportamiento real y no solo las percepciones o los recuerdos que las personas puedan tener sobre esos comportamientos. Una vez obtenida valiosa información gracias a esta herramienta, se buscó asentarla en un medio escrito con una entrevista, en la cual se realizó un cuestionario con preguntas enfocadas y dirigidas en los puntos que se pudieron obtener en el primer instrumento aplicado, sabiendo que la entrevista es una valiosa técnica que puede proporcionar varios beneficios, especialmente cuando se trata de obtener información directa de las personas involucradas para profundizar en información detallada y de interés que servirán en la comprensión más completa del tema. La entrevista se realizará con preguntas abiertas y cerradas. Se utiliza el tipo de preguntas cerradas porque se busca resaltar los problemas

ya encontrados en la observación y con este tipo de preguntas se logra respuestas específicas, gracias a que se colocan opciones limitadas de respuesta a elegir.

3.3.2. Análisis de Recolección de datos

Con la observación que se realiza durante tres meses, gracias a que el investigador de este trabajo es parte del equipo, se pudo obtener de primera mano, que existe un descontento y desmotivación en todo el equipo de desarrollo por el aumento de su carga horaria que les asignan, muchas veces hasta los fines de semana para poder cumplir con los tiempos de entrega y que, sin embargo, el producto es entregado en tiempos no indicados. En ocasiones se presentan nuevos requisitos no planteados al inicio que son aceptados por el gerente de sistemas, el cual los acepta para poder disminuir un poco el problema causado por no entregar a tiempo el producto.

Esta desmotivación cauda en el recurso humano, afecta en su rendimiento y su capacidad para realizar su trabajo, lo que ocasionará directamente problemas al proyecto, dando lugar de esta manera a distintos errores que aparecen en producción cuando el software ya está en funcionamiento. Estos errores deben ser corregidos sobre la marcha y luego puestos en producción nuevamente sin tener ningún escenario de pruebas para los mismos. Estos inconvenientes no sólo afectan al departamento que desarrolla y entrega, también afecta en el trabajo diario del departamento que usa el producto terminado, si es que fue un proyecto interno o a la empresa cliente que compra el sistema que se desarrolló.

Por medio de la entrevista se pudo obtener información relevante con respecto a que en la empresa Edina S.A., en su departamento de sistemas no se emplea un marco de trabajo para los proyectos de desarrollo de software y que los procesos que se usan en la actualidad no cubren las necesidades de la empresa y no son los adecuados para con estos trabajos de desarrollo. Además, el equipo de desarrollo en un 100% no está a gusto con la forma de trabajo con la que se lleva a cabo el desarrollo de software e insisten en que la causa por la que se entregan los productos fuera del tiempo es por la falta de una metodología que gestione las diferentes etapas de un proyecto de desarrollo de software

y opinan que con la aplicación de este marco dejarán de aparecer escenarios de error una vez que el producto esté en producción.

También se conoce gracias a la entrevista que el 100% de los miembros del equipo de desarrollo considera que con una estandarización para este tipo de proyectos mejoraría el desenvolvimiento de la empresa y estarían dispuestos a utilizar la metodología de un marco de trabajo ágil para que esta sea la guía y se puedan realizar todas las etapas correspondientes con el tiempo suficiente y sin errores en su ejecución. Al preguntárseles con respecto al conocimiento del marco de trabajo Scrum y si estarían dispuestos a usarlo, todos contestaron de forma positiva pudiendo de esta manera entender que no son inflexibles al cambio y mucho menos si este viene con muchas mejoras para lo que realizan en su trabajo diario.

Por otra parte, Scrum es un enfoque ágil que permite a los equipos adaptarse rápidamente a los cambios en los requisitos y las necesidades del cliente. La capacidad de ajustar el enfoque en cada tarea permite responder a la retroalimentación y a los cambios del mercado de manera más efectiva logrando así una mayor satisfacción del cliente, garantizando una mejora continua a través del aprendizaje de sus propias experiencias y buscando constantemente diversas formas de mejorar su desempeño.

3.4. METODOLOGÍA DE DESARROLLO

Para la realización del presente proyecto se ha considerado realizar una estructura de desglose de trabajo basada en fases según lo expuesto en la guía de fundamentos para la dirección de proyectos (Institute, 2017). Esta guía es una referencia fundamental para la gestión de proyectos y se utiliza ampliamente, proporcionando un marco y estándares para la dirección de proyectos.

- **Fase de reconocimiento:**

Como primera fase, se realizará una indagación preliminar basada en encuestas y observaciones directas con los implicados en el proceso de desarrollo de software de la empresa. Ese grupo es el que ayudará a conocer cómo se realiza actualmente el desarrollo

y cuáles son los efectos que causa el no tener una metodología o marco de trabajo para su desarrollo. En base a esto, se buscará tres proyectos parecidos, similares que se hayan investigado previamente, en el que se indique a base de resultados obtenidos, cuál es el mejor marco de trabajo para la gestión de etapas para una empresa que realiza sus desarrollos de manera interna para sus propios procesos y de manera externa para sus s que son pequeñas empresas.

- **Fase de revisión:**

En esta segunda fase, fundamentada en una exhaustiva investigación bibliográfica, se encontrará y aplicará dos marcos de trabajo de entre todos los consultados, para poder aplicarlos a la metodología de desarrollo de software de la empresa y poder analizar los resultados entre estas, para con esto poder realizar el cuadro comparativo de la siguiente etapa. Estos dos marcos deberán ser seleccionados, uno de entre los marcos de trabajo tradicionales y el otro de los marcos de trabajo ágiles. Estas dos selecciones basadas en bibliografía investigada, en la cual se exprese claramente cuál es el marco de trabajo indicado para el tipo de empresa que tenemos en este proyecto de investigación.

- **Fase de estudio observacional:**

Selección del marco de trabajo adecuado y aplicarlo en el sector objetivo y realizar un análisis comparativo entre lo que se tenía con la aplicación del marco de trabajo seleccionado. Al aplicarlo en el departamento de sistemas e indicando cómo se trabaja con el marco que resulte seleccionado de la etapa anterior, se obtendrá una comparación más detallada de lo que es un trabajo de desarrollo sin un marco a seguir con la diferencia de cuando se aplica un marco indicado y adecuado para el tipo de empresa que se tiene en este estudio.

- **Fase de construcción de la propuesta:**

De acuerdo con los datos obtenidos en el análisis de resultados, se elaborarán los lineamientos metodológicos, para poder de esta manera sacar conclusiones debido a que ya se tendrán las mejoras en cuanto a la gestión de las etapas del desarrollo. Se podrá recomendar sobre las ventajas que dejará el uso y aplicación de este marco en la empresa y este trabajo debe servir como base para todas aquellas empresas en las cuales estén buscando y hayan decidido incluir un marco para la gestión de sus desarrollos de software.

4. DESARROLLO ESPECÍFICO DE LA CONTRIBUCIÓN

4.1. TABLAS COMPARATIVAS

En este punto, se presenta una tabla comparativa de los marcos de trabajo existentes para el desarrollo de software, considerando que existen dos grandes tipos: las metodologías tradicionales y las ágiles. Las metodologías tradicionales son conocidas también como rígidas, pues establecen una disciplina de trabajo durante el proceso de desarrollo del software. Se caracterizan por definir los requisitos en la etapa inicial del proyecto de ingeniería de software; son poco flexibles y no permiten la realización de cambios y funcionan aplicando un enfoque de tipo lineal en el que las etapas del desarrollo del software deben complementarse de manera secuencial.

Por el contrario, las metodologías ágiles son estrategias integrales que impulsan a las empresas a gestionar sus proyectos de manera rápida y flexible. Son metodologías que ayudan en el desarrollo de proyectos más adaptados a las necesidades del cliente. Con este tipo de metodología los proyectos se dividen en partes, permitiendo la adaptación sobre la marcha, y la complementación de etapas en poco tiempo; de tal manera que, en vez de planificar y diseñar el proyecto por adelantado, se va definiendo en la medida que se desarrolla cada etapa. Al finalizar la ejecución de las tareas, cada uno de los miembros participantes hace la entrega de los avances, recibiendo devoluciones e iniciando otra vez el proceso; lo que conlleva a la implementación de los cambios. A continuación, se presenta la *Tabla 1*, la cual contiene las principales características de las metodologías tradicionales y ágiles, que permiten compararlas, establecer sus diferencias, ventajas y desventajas de cada una de ellas. La comparación se establece en función a los cambios del entorno, procesos de trabajo, documentación, roles y artefactos, responsabilidades y roles, clientes, contratos, dominio, influencia, jerarquía, entre otros aspectos.

Tabla 1. Cuadro comparativo metodologías tradicionales vs. ágiles

Aspecto Característico	Tradicionales	Agiles
Metodología	Metodología rígida	Metodología flexible
Arquitectura	Metodología centrada en la arquitectura	Metodología con menos énfasis en la arquitectura

Planeación	Requieren de un plan detallado desde que inicia el proyecto	Se va planeando gradualmente a medida que se avanza en el proyecto
Requisitos	Los requisitos son detallados desde el inicio del proyecto y no pueden cambiar	Los requisitos son cambiantes
Planificación del trabajo	Proceso de trabajo planificado y controlado con muchas reglas	Proceso de trabajo no planificado y con pocas reglas
Funcionalidades innecesarias	Permite la solicitud de funcionalidades innecesarias	El enfoque continuo no permite la inclusión de funcionalidades innecesarias
Contratos	Trabajan con contratos tradicionales	Requieren de contratos ágiles
Cambios	Alta resistencia al cambio. Hacer un cambio requiere de un proceso formal y rígido	Dispuestas para el cambio. Un cambio puede darse en cualquier momento del proyecto
Tiempo	Se conoce el tiempo de entrega del producto	Existe incertidumbre respecto al tiempo de entrega del producto

Costo	El costo del proyecto está previamente definido	Existe incertidumbre respecto al costo del proyecto
Documentación	Se da una atención exhaustiva a la documentación	Solo se genera la documentación de valor al cliente y al proyecto
Cliente	El cliente participa en reuniones de apoyo para el desarrollo del producto, es decir, interactúa con el equipo de trabajo	El cliente se involucra directamente en el desarrollo del producto, es decir, es un miembro del equipo de trabajo
Tipo de proyecto	Metodología eficiente para proyectos restrictivos e ineficaz para proyectos modernos	Metodología poco eficiente para proyectos restrictivos y eficaz para proyectos modernos
Riesgos	Los riesgos son asumidos por el proveedor	Existe voluntad del cliente para compartir riesgos
Valor	Se valora más el proceso	Se valora más al individuo y sus interacciones
Entregables	Se producen entregables que requieren mucho tiempo para su elaboración	Se producen entregables en tiempos cortos y con alta calidad
Plantillas y Artefactos	Innumerables plantillas y artefactos para cumplir con el proceso	Pocas plantillas y artefactos para el proceso

Roles	Empleo de muchos roles	Empleo de pocos roles
Ciclos de entrega	Pocos	Muchos
Burocracia	Existencia de una mayor burocracia	Existencia de una menor burocracia
Estructura jerárquica	Jerarquía con estructura vertical	Jerarquía con estructura horizontal

Fuente: Autor con base en (Dousdebes, 2017), (Maida & Pacienza, 2015) y (Navarro, Fernández, & Morales, 2013)

Conociendo ya las principales diferencias entre marcos de trabajo tradicionales y ágiles, se procede a comparar algunos de estos, escogiendo los más representativos. Para las metodologías tradicionales: Cascada, Espiral y Proceso Unificado, las cuales son las más mencionados por los diferentes autores que las definen; y para las ágiles: Scrum, XP y Kanban; estas últimas son las más utilizadas y todas están guiadas por un patrón establecido por el Manifiesto Ágil.

Tabla 2. Marcos de trabajo tradicionales y ágiles más representativos

Tradicionales	Ágiles
<p>Cascada</p> <p>Es un marco de trabajo tradicional que define un conjunto de fases por las cuales transita el proyecto de desarrollo de software, para poder llegar a término de manera exitosa. Es un proceso que está dirigido por un plan; en el que se plantea y programan todas las actividades del proceso de antemano.</p>	<p>Programación Extrema XP</p> <p>Es la metodología más utilizada del desarrollo de software ágil, que tiene por objetivo el proceso ágil usado en organizaciones grandes. La programación extrema (XP) utiliza un enfoque orientado a objetos y engloba reglas y prácticas que se desarrollan en el contexto de cuatro actividades estructurales, como lo son: planeación, diseño, codificación y pruebas.</p>

Espiral

Es un marco de trabajo tradicional de desarrollo de software centrado en hacer entregas de manera secuencial del software considerando un análisis del riesgo para la realización de la nueva iteración. Es un modelo evolutivo e iterativa para hacer prototipos con aspectos controlados y sistémicos de la metodología cascada. Tiene la característica de hacer un rápido desarrollo de versiones cada vez más completas. Este modelo comienza con la especificación de los requerimientos del cliente, avanzando mediante la planeación, modelado, construcción y despliegue.

Scrum

Scrum es un marco de trabajo ágil propuesto para equipos de desarrollo pequeños, enfocado en proyectos complejos, donde es necesaria la comunicación constante con los stakeholders, cuyo objetivo es proporcionar entregas de software funcional entre periodos cortos y fijos, denominados sprints.

Proceso unificado

Es una metodología tradicional de desarrollo de software incremental e iterativa, basada en componentes e interfaces. El proceso unificado es un proceso del software impulsado por el caso de uso y centrado en la arquitectura. Existen dos implementaciones de este marco de trabajo, a saber, Proceso unificado, Open Unified Process (UOP) y Rational Unified Process (RUP) de IBM.

Kanban

Es una metodología ágil diseñada para disparar trabajo cuando existe capacidad para procesarlo. Se implementa a través de tableros Kanban y permite a los equipos visualizar sus flujos y carga de trabajo; los cuales se muestran en un proyecto en forma de tablero organizado por columnas, donde cada columna representa una etapa.

Fuente: Autor con base en (Dousdebes, 2017), (Maida & Pacienza, 2015) y (Pressman, 2010)

Con base en la información presentada en las *Tablas 1 y 2*, será preciso para esta investigación la elección de una metodología flexible, que acepte cambios y se adapte al entorno actual; de modo que para efecto del presente estudio se trabajará con una metodología ágil. Es importante mencionar que existe una diversidad de metodologías ágiles que han venido apareciendo y se han utilizado en el desarrollo de software. Algunas de las más representativas son: Scrum, XP, Kanban; de modo que, para poder realizar la mejor elección de la metodología ágil, es necesario conocer las características particulares de cada una de ellas, las cuales se pueden visualizar en la *Tabla 3*.

Tabla 3. Cuadro comparativo de las tres metodologías ágiles más representativas

Aspecto	XP	Scrum	Kanban
Roles	Tracker, Customer, Programmer, Coach, Manager, Tester	Stakeholders, Product Owner, Scrum Master, Equipo	Service request manager, Service delivery manager
Artefactos	Tarjetas de historias de usuario, tarjetas de tareas para descargas de documentos, código, pruebas unitarias y de integración, pruebas de aceptación	Reserva de Producto, Reserva de Sprint, Estados	Puntos de control, Tipos de ítems de trabajo, Tarjetas de ítems de trabajo, Secuencia de actividades, Tablero, Limitación del WIP, Optimización del flujo
Reuniones	Reuniones diarias	Planificación de Sprint, reunión diaria, revisión de Sprint, Retrospectiva de Sprint	Reunión de reabastecimiento, Kanban diario, reunión de prestación de servicios, reunión de planificación de entregas, revisión de estrategias, revisión de

			operaciones, revisión de riesgos
Actividades / Fases	Planeación, Diseño, Codificación, Pruebas, Lanzamiento	Preparación, juego, post-juego	Visualización del trabajo y las fases del ciclo de producción, determinación del límite de trabajo en curso, medición del tiempo en completar una tarea
Ventajas	Eficiencia en el proceso de pruebas y planificación, Tasa de error pequeña, programación organizada, Fomento de comunicación entre desarrolladores y clientes.	Claridad en los objetivos, Flexibilidad, Mayor control de imprevistos, Equipos altamente productivos, Predictibilidad, Fácil implementación	Generación de valor, Equipo de trabajo organizado, Cumplimiento en los Tiempos de entrega, eficiencia en la asignación de tareas, Calidad del producto final, Flexibilidad, Contribuye a la mejora de manera colaborativa

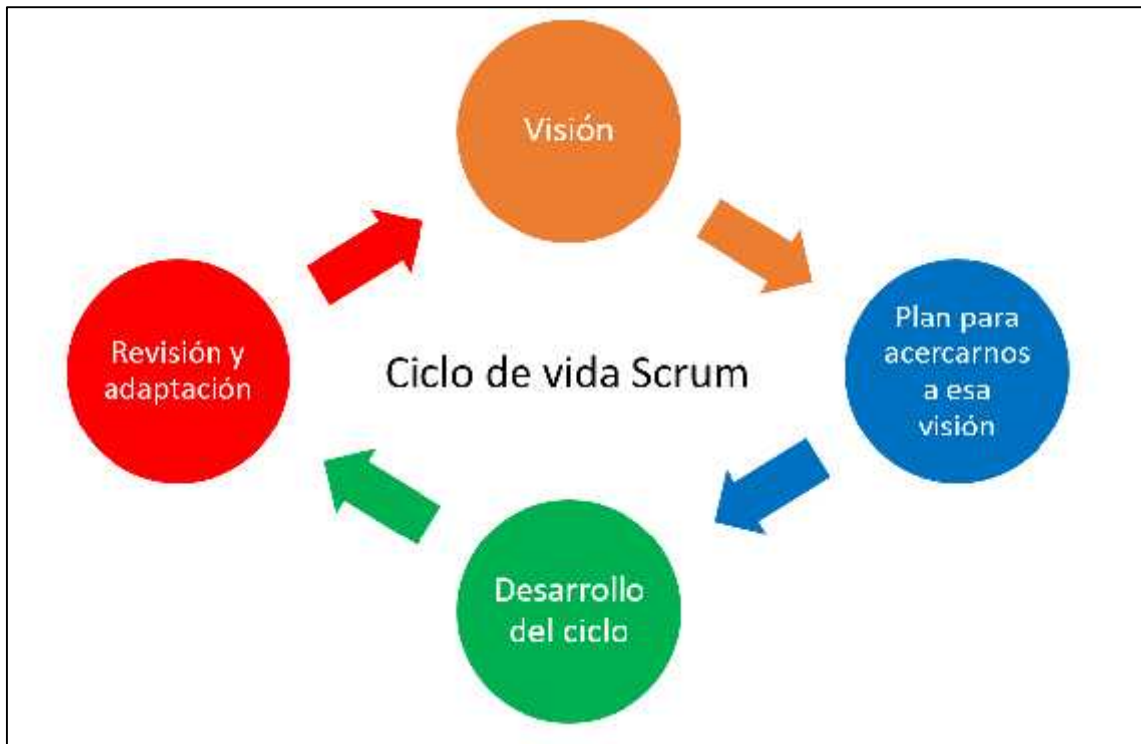
Fuente: Autor con base en (Dousdebes, 2017), (Maida & Pacienza, 2015) y (Pressman, 2010)

4.2. RESULTADOS

Scrum es considerada como marco para la gestión de proyectos a nivel internacional, ayudando a mejorar el retorno sobre la inversión de los mismos; dado a que es un modelo iterativo, rápido, flexible y eficaz, creando un ambiente colaborativo y responsable que propende a la mejora continua (García, 2021). A continuación, el ciclo de vida de un proyecto de software desarrollado en la Empresa Edina ya con la aplicación de la

metodología del marco de trabajo Scrum, en el cual ya se cuenta con planeación y tiempo a través de Sprint.

Gráfico 1. Ciclo de vida Scrum



Fuente: (Sutherland, 2020).

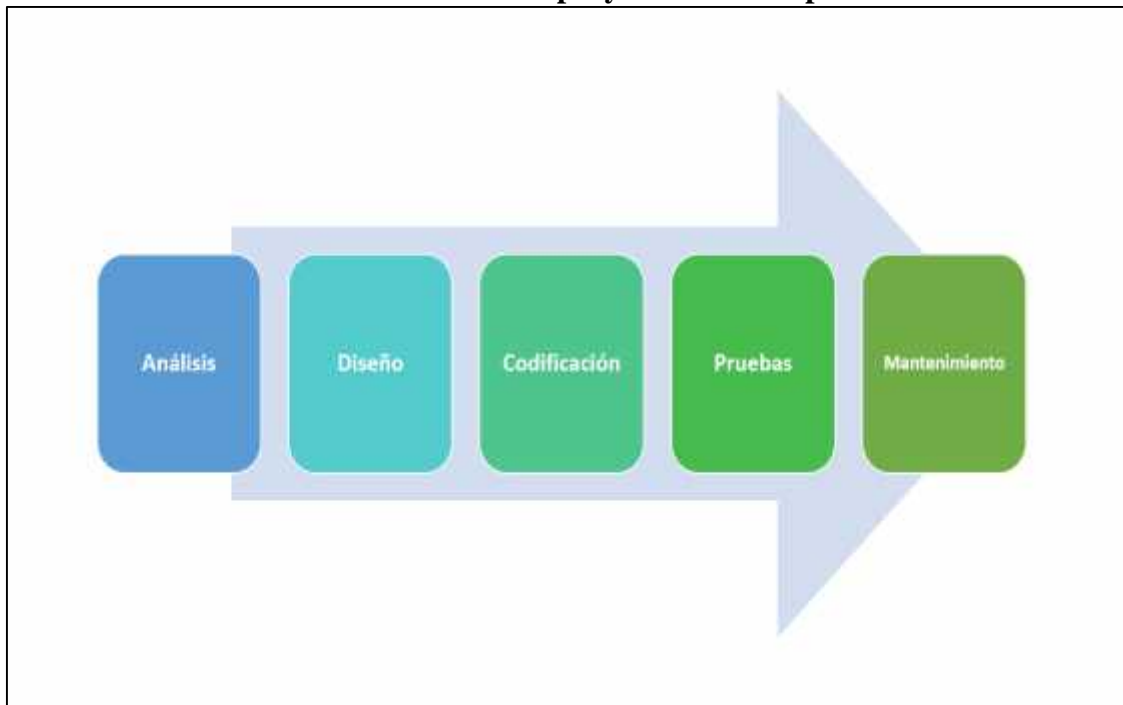
Con este ciclo de vida de Scrum, separados en Sprint, se puede dividir el tiempo de trabajo en períodos más cortos, haciendo que una tarea muy grande, se convierta en varias tareas pequeñas. Se mejoró además en tiempos de entrega ya que cada Sprint tuvo una duración de dos semanas y si existen cambios, esta metodología permite adaptarse a estos. El Sprint es vital para lograr una entrega iterativa e incremental de valor, garantizar la transparencia y la rendición de cuentas, facilitar la mejora continua y mantener el enfoque del equipo en los resultados medibles y concretos. Al seguir el ciclo de Sprints, el equipo de desarrollo puede adaptarse rápidamente a los cambios y ofrecer valor de manera constante.

Desde el sólo hecho de ya contar con una planeación desde el inicio del proyecto, ya es una mejora porque en la empresa se desarrollaba sin los tiempos establecidos para las fases requeridas y necesarias para que un desarrollo de software sea muy bien realizado. Como resultados tenemos la mejora en planeación, tiempos de ejecución de las tareas y

entregas de los diferentes productos que se van obteniendo en las diferentes etapas del marco de trabajo Scrum.

En mejoras del desarrollo como tal, los resultados también son muy visibles y verificables. En el gráfico que se muestra a continuación, se explica cómo se lo hacía antes de aplicar el marco de trabajo Scrum, en el cual en la última etapa de mantenimiento se tenía un producto terminado con muchos errores que eran corregidos en producción. Esto causaba mucha carga horaria posterior al pase a producción para el equipo de desarrollo y un descontento en el usuario final que usaba el sistema:

Gráfico 2. Ciclo de desarrollo de proyecto en la Empresa Edina S.A.

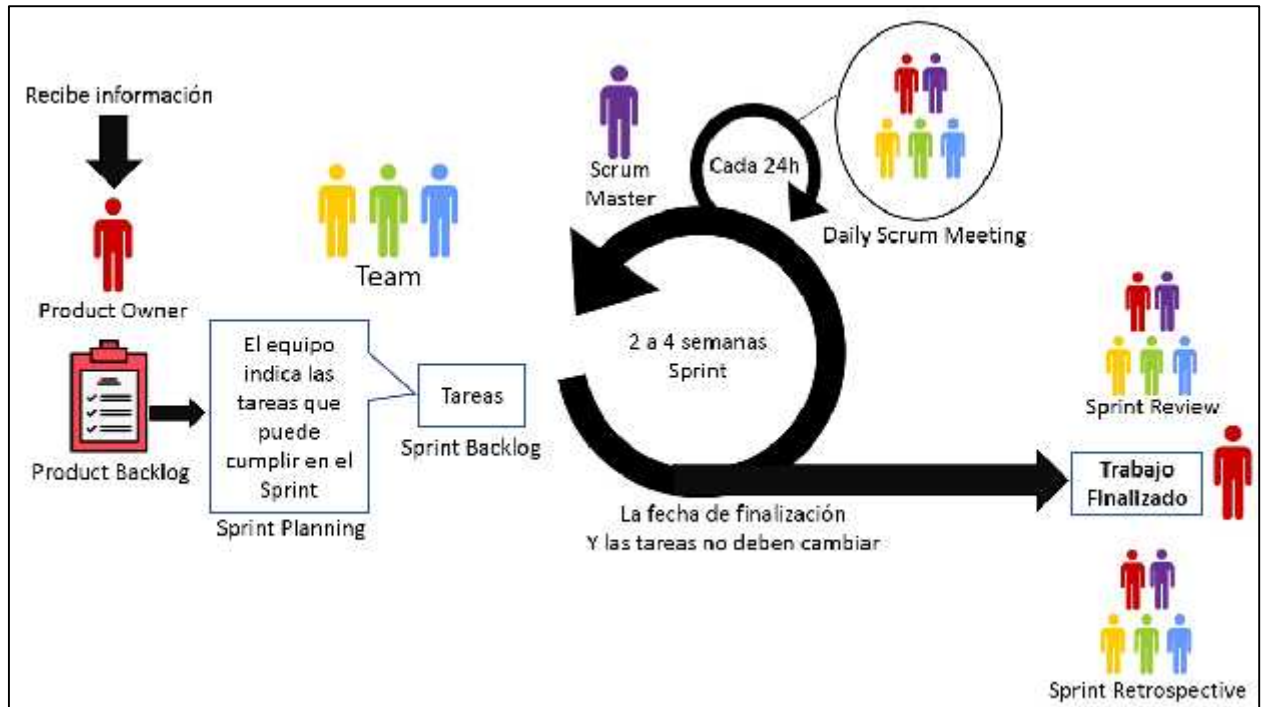


Fuente: Autor.

Para poder implementar el marco de trabajo Scrum se necesita una sala para las reuniones, un computador en el cual el Scrum Máster llevará el control de los diferentes Sprint, las personas idóneas para el equipo desarrollo, un Product Owner, un Scrum Máster y toda la infraestructura necesaria para que estos desarrolladores puedan realizar sus tareas sin inconvenientes. Esta infraestructura mencionada, ya era parte de la empresa y no fue necesario que se haga alguna adquisición, evitando así la inversión monetaria que podría

ser un pretexto para no utilizar la nueva metodología a aplicar. El proceso del marco de trabajo Scrum trabaja de la siguiente manera:

Gráfico 3. Proceso de desarrollo de proyectos usando Scrum



Fuente: (DEEMER Pete, 2009).

Se le pide al departamento de Sistemas, realizar una actualización en el menú de su SGC (Sistema de Gestión de Cliente) para agregar una opción nueva con la cual este no cuenta, aprovechando que este pedido fue antes de la implementación de Scrum, se acepta y documenta el desarrollo y así poder tener un análisis real de las etapas del proyecto y todas las falencias que se presentarían en este. Como se puede observar en el siguiente gráfico, las distintas tareas planteadas (por los mismos desarrolladores), se extienden demasiado en tiempos para su ejecución. Estas tareas, no tuvieron una etapa suficiente de análisis y es por esto que se logra identificar que desde la planeación existen fallas al no existir el tiempo necesario para dicho análisis y no tener el espacio suficiente para el diseño y codificación:

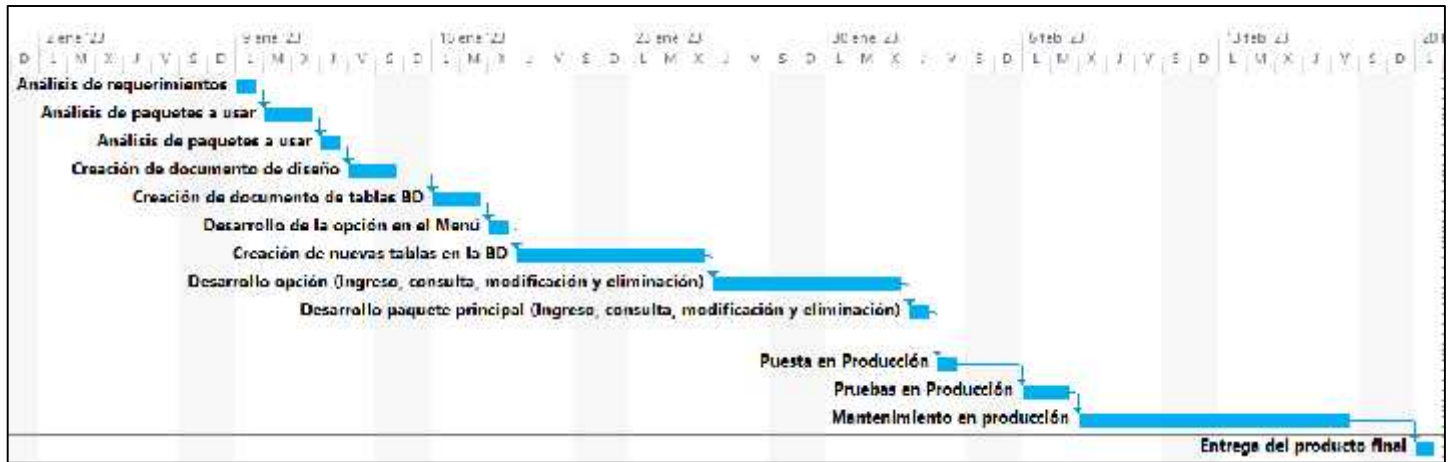
Gráfico 4. Tareas del Proyecto “Desarrollo de nueva opción para agregar servicios en el Sistema de Clientes”

Nombre de tarea	Duración	Comienzo	Fin
Análisis de requerimientos	1 día	lun 9/1/23	lun 9/1/23
Análisis de paquetes a usar	2 días	mar 10/1/23	mié 11/1/23
Análisis de paquetes a usar	1 día	jue 12/1/23	jue 12/1/23
Creación de documento de diseño de pantallas	2 días	vie 13/1/23	sáb 14/1/23
Creación de documento de nuevas tablas en la BD	2 días	lun 16/1/23	mar 17/1/23
Desarrollo de la opción en el Menú	1 día	mié 18/1/23	mié 18/1/23
Creación de nuevas tablas en la BD	5 días	jue 19/1/23	mié 25/1/23
Desarrollo de pantalla principal de la opción Servicios (Ingreso, consulta, modificación y eliminación)	5 días	jue 26/1/23	mié 1/2/23
Desarrollo del paquete principal (Ingreso, consulta, modificación y eliminación)	1 día	jue 2/2/23	jue 2/2/23
Puesta en Producción	1 día	vie 3/2/23	vie 3/2/23
Pruebas en Producción	2 días	lun 6/2/23	mar 7/2/23
Mantenimiento a los errores presentados en producción	8 días	mié 8/2/23	vie 17/2/23
Entrega del producto final a usuarios	1 día	lun 20/2/23	lun 20/2/23

Fuente: Autor.

En el cronograma que se muestra a continuación, se notan las tareas con sus tiempos demasiado amplios por el motivo que no se hizo bien el análisis y al realizar la puesta en producción del proyecto, se encuentran inconvenientes todo esto porque las pruebas se realizan en producción y todo error encontrado debe ser resuelto con el software ya en uso por los usuarios finales, que son los que informan de los diferentes problemas que van encontrando:

Gráfico 5. Tiempo de ejecución del Proyecto “Desarrollo de nueva opción para agregar servicios en el Sistema de Clientes”



Fuente: Autor.

Dando, así como resultado, un software que luego de ocho días de estar en producción, resolviendo sus errores día a día, causando inconvenientes en el trabajo diario de usuarios finales y de desarrolladores que deben arreglar todo en el menor tiempo posible. La falta de motivación entre los desarrolladores de software debido a la carga de trabajo excesiva es un problema común en la industria de la tecnología. El agotamiento por trabajar largas horas puede conducir a una mala salud física y mental. Los desarrolladores necesitan tiempo para descansar y recuperarse para mantener altos niveles de productividad y creatividad. Si los desarrolladores trabajan muchas horas seguidas, la calidad de su trabajo puede verse afectada. La fatiga y el estrés pueden conducir a errores y malas decisiones que afectan directamente su trabajo.

Por el lado del usuario final, causar problemas en su trabajo diario debido a un software que contiene errores o fallas, puede tener varios impactos negativos en la productividad y la satisfacción del mismo. Algunas formas en las que los errores en el software pueden afectar incluyen pérdida de tiempo y eficiencia, enfrentando retrasos en sus tareas diarias cuando el software no funciona correctamente. También se logra desmotivación por el lado de estos usuarios, afectando su actitud hacia el trabajo y su disposición para utilizar el software.

Ahora, se le pide al departamento de Sistemas, realizar una actualización en el menú de su SGC (Sistema de Gestión de Cliente) para agregar una opción nueva muy similar a la analizada en los gráficos anteriores, a diferencia de que este desarrollo se lo va a realizar con la implementación de Scrum, para documentar el desarrollo y así poder tener un análisis real de las etapas del proyecto y todas las mejoras que se presentarían en este. Como se puede observar en el siguiente *Gráfico 6*, las distintas tareas cuentan con el tiempo necesario para su ejecución. Teniendo desde el inicio el Sprint Planning, el cual lleva todo el análisis necesario que no se tenía en proyectos en los cuales no se aplicaba el marco y cumpliendo de una manera más eficiente con las tareas destinadas para realizar el diseño y codificación:

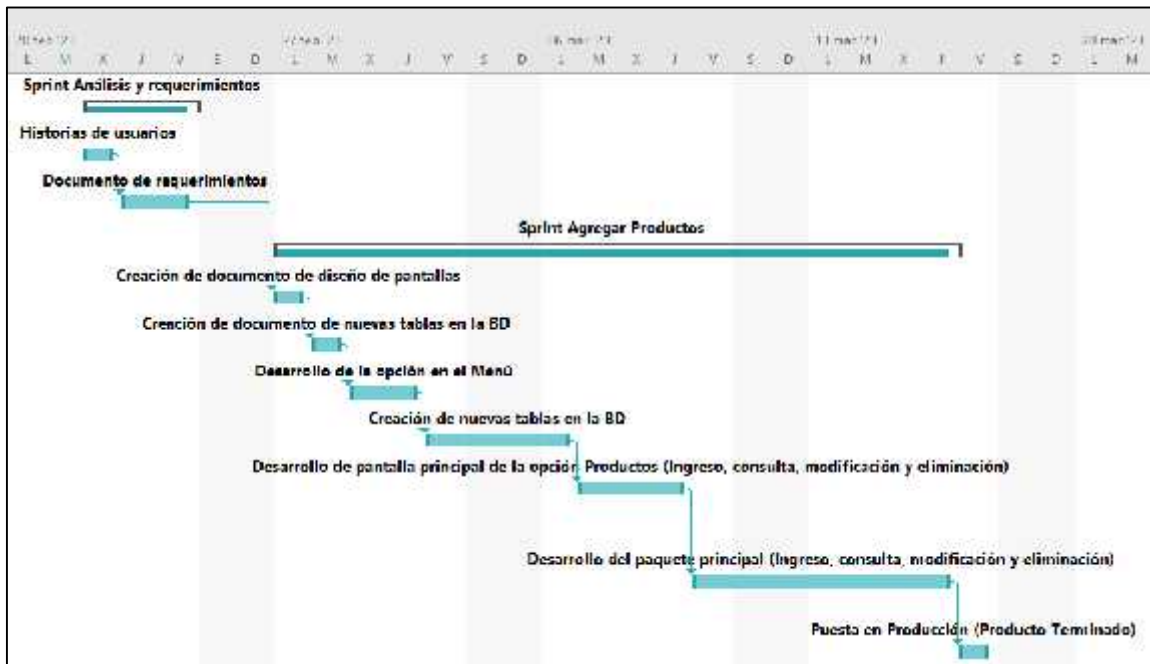
Gráfico 6. Tareas del Proyecto “Desarrollo de nueva opción para agregar productos en el Sistema de Clientes”

Nombre de la tarea	Duración	Comienzo	Fin
▣ Sprint Análisis y requerimientos	3,25 días	mié 22/02/23	vie 24/02/23
Historias de usuarios	1 día	mié 22/02/23	mié 22/02/23
Documento de requerimientos	2 días	jue 23/02/23	vie 24/02/23
▣ Sprint Agregar Productos	14,25 días	lun 27/02/23	jue 16/03/23
Creación de documento de diseño de pantallas	1 día	lun 27/02/23	lun 27/02/23
Creación de documento de nuevas tablas en la BD	1 día	mar 28/02/23	mar 28/02/23
Desarrollo de la opción en el Menú	2 días	mié 01/03/23	jue 02/03/23
Creación de nuevas tablas en la BD	2 días	vie 03/03/23	lun 06/03/23
Desarrollo de pantalla principal de la opción Productos (Ingreso, consulta, modificación y eliminación)	3 días	mar 07/03/23	jue 09/03/23
Desarrollo del paquete principal (Ingreso, consulta, modificación y eliminación)	5 días	vie 10/03/23	jue 16/03/23
Puesta en Producción (Producto Terminado)	1 día	vie 17/03/23	vie 17/03/23

Fuente: Autor.

Vemos así de esta manera, notables cambios y mejoras. Tiempo suficiente para realizar todas las tareas planeadas y un producto terminado sin errores que no afecte el trabajo diario de los usuarios finales del software puesto en producción, así como también permite que los desarrolladores cumplan con sus tiempos de labores normales sin excedente en sus cargas horarias:

Gráfico 7. Tiempo de ejecución del Proyecto “Desarrollo de nueva opción para agregar productos en el Sistema de Clientes”



Fuente: Autor.

Es importante recordar que una implementación exitosa de Scrum no garantiza automáticamente una mejora significativa en el tiempo de entrega. La colaboración, la comunicación efectiva y la mejora continua son esenciales para el éxito de Scrum y permiten una entrega más rápida y eficiente. Además, Scrum puede no ser adecuado para todos los proyectos o equipos, por lo que la metodología debe adaptarse a las necesidades y características específicas de cada proyecto.

5. CONCLUSIONES

Como ya se analizó en los apartados anteriores, existen muchas formas de hacerle frente al desarrollo de software, y cada una de las metodologías enfoca el trabajo desde visiones diferentes. No obstante, a pesar de las diferencias, algunos de ellas tienen una apariencia similar, indistintamente de pertenecer al marco de trabajo tradicional o ágil, tal es el caso, por ejemplo, de las iteraciones que están presentes en metodologías como la espiral, la Rational Unified Process y la Scrum.

Sea cual sea el marco de trabajo para desarrollar el software, todas las metodologías son realmente útiles, además que cada una de ellas involucra un objetivo, alcance, tiempo, costo y personas específicas, que componen el proyecto. De modo que identificar las ventajas y desventajas, a veces representa un trabajo difícil, sobre todo determinar las características de cada método de desarrollo en contraste con el resto. Es por ello que, con las tablas comparativas de los marcos de trabajo, ha resultado un poco más fácil identificar el más apropiado para la empresa objeto de estudio de la presente investigación, a saber, las metodologías ágiles.

Como ya se ha mencionado, la metodología ágil debe basarse en el Manifiesto ágil, como, por ejemplo, XP, Scrum y Kanban, modelos iterativos, que reportan ventajas referidas a un trabajo incremental y de calidad; que toman muy en cuenta al cliente, y permiten realizar cambios importantes en el camino, aun cuando las mejoras no se hayan planificado de antemano. Además, son metodologías más baratas, es decir, implican el uso de menos tiempo y recursos, logrando los mismos resultados, incluso mejores, comparados con las metodologías tradicionales.

Ahora bien, entre las tres metodologías ágiles analizadas en el cuadro comparativo, la que se acerca más a las necesidades de la empresa objeto de estudio es la Scrum, por tener ventajas tales como claridad en la definición de objetivos, flexibilidad, control de imprevistos, uso de equipos altamente productivos, predictibilidad y por ser de fácil implementación. Además, dentro de los roles entra en juego un Scrum Master, quien es

el máximo responsable del desarrollo de un proyecto de tipo ágil, entre sus funciones está el liderazgo de equipos, el control de calidad del trabajo y la organización de los equipos. Sobre esto, el propio investigador de este estudio cuenta con la certificación Scrum Master, quien, a su vez, forma parte del Departamento de Sistemas; por lo que esto no representaría un problema a la hora de ejecutar la metodología de desarrollo de software en la empresa objeto de estudio.

Con la implementación de Scrum en los proyectos de desarrollo como bien lo podemos ver en el *Gráfico 5*, que se necesitó treinta y un días laborables para poder realizar la entrega del producto a los usuarios y en el *Gráfico 7*, nos damos cuenta de resultados muy favorables en un proyecto similar donde se necesitaron once días laborables, teniendo con esto una reducción muy considerable de tiempo de veinte días siendo esto un 35,48% de reducción.

También podemos resaltar de que esta mejora es debido a que existe menos presión para los miembros del equipo de desarrollo, los cuales ya no deben estar analizando mientras programan, sino más bien, ya tienen una etapa de análisis donde participaron todos y sacaron las principales tareas desde el inicio del proyecto, incluyendo al Product Owner que siempre participa en todas las etapas del proyecto.

En cuanto a la entrega del producto, se eliminan los errores presentados ya en producción, teniendo que resolver todos los escenarios ocho días laborables y ahora reducir eso un 100%, gracias a que, con Scrum, se detectan los errores en sus etapas previas a la puesta en producción del producto. Con esto podemos realizar una proyección de que antes de Scrum se realizaban proyectos y se entregaban en un año 10 productos de mala calidad y después de Scrum se entregarán cerca de 24 productos, sin errores, sin más cargas horarias, sin presiones y con usuarios felices porque ya no se detendrá su trabajo debido a que se tenía que realizar un pase a producción para que corrija uno o más escenarios de error.

6. RECOMENDACIONES

El uso de las metodologías ágiles, permiten potenciar las ventajas competitivas de las organizaciones y/o proyectos en donde se implementan, ya que permite afrontar de manera eficaz las posibles variaciones que se presenten, gracias a la mejora de la comunicación entre los involucrados del proyecto, lo que a su vez genera una mayor comprensión de los requisitos y reducción de los reprocesos que afectan las condiciones de calidad de un proyecto: alcance, tiempo y costo (García, 2021).

Es importante fomentar la colaboración y la comunicación en Scrum, pudiendo así obtener la participación activa de todos los miembros del equipo y siempre buscando una comunicación abierta y transparente. Es crucial establecer un entorno propicio para la colaboración, donde los miembros del equipo se sientan cómodos compartiendo ideas, resolviendo problemas y tomando decisiones de manera conjunta. Las historias de usuario son consideradas la base para el desarrollo del proyecto, estas deben ser pequeñas y bien definidas para así evidenciar claramente los criterios de adaptación, los interesados son aquellos que no hacen parte del equipo Scrum y que son consumidores del producto en construcción o que pueden influir en él; ayudan a descubrir, desarrollar, lanzar y promover el producto (García, 2021).

Gracias a la implementación de Scrum que es el marco de trabajo elegido para el desarrollo de proyectos en la empresa Edina, se ha podido conseguir la utilización de una herramienta que nos sirve de mucha ayuda en el análisis, planeación, desarrollo y puesta en producción de los desarrollos de Software, brindando de esta manera un trabajo más organizado, evidenciando para todos los miembros del equipo y para los usuarios finales del producto, las diferentes mejoras que se mencionaron en los apartados en el cual se analizó la implementación de dicho marco de trabajo y del por qué se elegiría a este.

El Manual de Scrum de Ken Schwaber y Jeff Sutherland (creadores de Scrum) establece que el Scrum Team debe ser lo suficientemente pequeño para seguir siendo ágil y lo suficientemente grande como para completar un trabajo significativo dentro de un Sprint,

generalmente de diez personas o menos (Sutherland, 2020). Con esto se recomienda a la empresa que aumente su número de desarrolladores de tres personas a seis como mínimo, sabiendo que es importante que el equipo tenga las diversas habilidades necesarias en todos los aspectos del desarrollo como son diseño, programación, documentación y pruebas.

Se seguirá usando este marco de trabajo para garantizar que siempre se encuentre la excelencia en el desarrollo de proyectos de software, apoyado en la mejora de las habilidades y capacidades del equipo, garantizando la eficiencia en la entrega del producto y brindando siempre oportunidades de aprendizaje y mejora en el corto plazo. Scrum es una herramienta muy poderosa y se debe entender que su éxito depende de la dedicación y compromiso de todos los miembros del equipo.

Se puede considerar a futuro integrar pruebas automatizadas en su proceso Scrum para ayudar con la entrega continua y rápida de nuevas funciones. Las prácticas se pueden integrar para automatizar la implementación, las pruebas y la supervisión del software. Al permitir pruebas rápidas y repetibles, ayudará a garantizar la calidad del software y acortará el ciclo de desarrollo, incluyendo varios tipos de pruebas automatizadas como son las pruebas unitarias, pruebas de integración y pruebas de aceptación automatizadas.

Para obtener una imagen más clara del rendimiento del equipo y el progreso del proyecto, también se puede pensar en incorporar métricas y análisis en el proceso Scrum, tales como la productividad del equipo, la calidad del código, la satisfacción del cliente y otros indicadores cruciales para comprender y tomar decisiones acertadas. En cuanto a la gestión de riesgos a medida que se desarrolla el proyecto, es importante tenerla en cuenta, estableciendo estrategias para mitigar los riesgos potenciales después de identificarlos y así disminuir los efectos adversos en el proyecto, realizando evaluaciones de riesgo periódicas y tomando las medidas preventivas que sean necesarias.

BIBLIOGRAFÍA

- Alvarado, P., & Arriola, C. (2020). *Marco de trabajo para la gestión de proyectos de desarrollo de software de la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruiz Gallo - Lambayeque 2016*. Universidad Nacional Pedro Ruiz Gallo, Lambayeque, Perú.
- Anglada, R., & Garófalo, A. (Abril-Junio de 2013). Marco de trabajo para el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura en GNU/Linux . *Revista Cubana de Ciencias Informáticas* , 7(2).
- Argibay, J. C. (2009). Muestra en investigación cuantitativa. *Subjetividad y Procesos Cognitivos*.
- Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley professional.
- Bioul, G., Escobar, F., Alvarez, M., Nardin, A., & Aparicio, E. (2010). Metodologías Ágiles , análisis de su implementación y nuevas propuestas. (págs. 597–606). Argentina: XVI Congreso Argentino de Ciencias de la Computación.
- Cadavid, A., Fernández, D., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Revista Prospectiva*.
- Canos, J., & Letelier, P. (2003). *Metodologías Ágiles en el Desarrollo de Software*. España: JISBD.
- Cárdenas, C. (2018). *Mejoramiento de los procesos de desarrollo de software para la empresa HSB SOFTECUADOR C.A*. Tesis, Pontificia Universidad Católica del Ecuador. , Quito, Ecuador.
- Caregua, L. (2021). *Implementación de metodología ágil en la gestión de proyectos en una fábrica de software en la ciudad de Guayaquil*. Tesis de Maestría, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.
- Carnaghan, I. (2014). *An expeditionary learning approach to effective curriculum mapping formalizing the process by exploring a user-centered framework* . Tesis Doctoral, Retrieved from University of Baltimore, ProQuest, UMI Dissertations Publishing.

- Coque, S. (2017). *Modelo de adopción de metodologías ágiles en mi PYME desarrolladora de software*. Tesis de maestría, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.
- Danissa. (2020). TÉCNICAS DE RECOPIACION DE INFORMACIÓN. *Club de ensayos*.
- DEEMER Pete, B. G. (2009). Información Básica de Scrum the Scrum Primer Version 1.1. *Scrum Training Institute*.
- Díaz, L. (2011). *La Observación*. Ciudad de México: Facultad Psicología, UNAM.
- Digital Talent Agency. (2018). *Modelo de gestión de proyectos*. (D. T. Group, Ed.)
- Dousdebes, J. (2017). *Marco de Trabajo para el desarrollo de Software*. Proyecto de Grado, Repositorio Universidad Católica de Salta, Argentina.
- Egas, L. (2019). *Análisis de Metodologías ágiles para la estandarización de procesos de gestión de proyectos de software en el departamento de desarrollo de sistemas de la dirección de tecnologías de información y comunicaciones de la Fuerza Aérea Ecuatoriana*. Tesis de Maestría, Universidad Técnica de Ambato, Ambato, Ecuador.
- Egas, L., & Játiva, J. (2014). IMPLEMENTACIÓN DE UNA METODOLOGÍA DE DESARROLLO ÁGIL PARA EL ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA SOFTWARE Y APLICACIÓN DE UN CASO DE ESTUDIO PRÁCTICO, EN EL DEPARTAMENTO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES DE LA ESCUELA TÉCNICA DE LA. *IMPLEMENTACIÓN DE UNA METODOLOGÍA DE DESARROLLO ÁGIL PARA EL ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA SOFTWARE Y APLICACIÓN DE UN CASO DE ESTUDIO PRÁCTICO, EN EL DEPARTAMENTO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES DE LA ESCUELA TÉCNICA DE LA*. Latacunga, Ecuador.
- Fayad, M., & Schmidt, D. (1997). Object-oriented applications frameworks. *Communications of the ACM*, 40(10), 32–38.

- Fowler, M. (2005). *The new methodology* . Obtenido de <https://martinfowler.com/articles/newMethodology.html>
- Gaete, J., Villarroé, R., Figueroa, I., Cornide, H., & Muñoz, R. (2021). Enfoque de aplicación ágil con Serum, Lean y Kanban. *Ingeniare. Revista chilena de ingeniería*, 29(1), 141-157.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company.
- Garcés, L., & Egas, L. (2015). Evolución de las Metodologías de Desarrollo de la Ingeniería. *Revista Científica y*.
- García, J. (2021). Diseño de una Propuesta de Aplicación de Scrum en la Ejecución de Proyectos de Infraestructura y Dotación de Espacios Lúdicos en el Municipio de Vista Hermosa Departamento del Meta. *Diseño de una Propuesta de Aplicación de Scrum en la Ejecución de Proyectos de Infraestructura y Dotación de Espacios Lúdicos en el Municipio de Vista Hermosa Departamento del Meta*. Bucaramanga, Colombia.
- Hughes, P., Fisher, P., & Mc Daniel, J. (2010). *System development life cycle models and methodologies*. Canadian Society for International Health Certificate Course in Health Information Systems.
- Institute, P. M. (2017). *La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK) / Project Management Institute*. Project Management Institute, Inc.
- Kerievsky, J. (2004). *Refactoring to Patterns*. Addison-Wesley.
- Khurana, H., & Sohal, J. (2011). Agile: The necessitate of contemporary software developers. *International Journal of Engineering Science & Technology*, 3(2), 1031-1039.
- Kniberg, H. (2007). *Scrum y XP desde las trincheras*. EUA: C4Media Inc.
- Kupiainen, E., Mantyla, M., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development A systematic literature review of industrial studies". *Information and Software Technology*, 62, 143-163.

- Lee, J. (2006). Embracing agile development of usable software systems. In CHI'06 Extended Abstracts on Human Factors in Computing Systems. En *In CHI'06 Extended Abstracts on Human Factors in Computing Systems* (págs. 1767-1770). doi:<https://doi.org/10.1145/1125451.1125784>
- López, G. (2021). *Propuesta metodológica de desarrollo ágil de software con énfasis en la seguridad*. Tesis de Maestría, Universidad Técnica de Macaha.
- Magallanes, J. (2021). *Análisis de las metodologías ágiles y su incidencia en el desarrollo de software*. Tesis, Universidad de Guayaquil, Guayaquil, Ecuador.
- Maida, E., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Tesis, Universidad Católica Argentina.
- Marante, M. (2009). Planificación y seguimiento en proyectos de desarrollo y mantenimiento de software dirigido por gestión de tiempos. España: Universidad Politécnica de Valencia.
- Marin, A., Trujillo, Y., & Buedo, D. (2018). Marco de Trabajo para gestionar actividades de calidad. *Revista Cubana de Ciencias Informáticas*, 12(2), 74-88.
- Martínez, I. (2015). *Marco de trabajo para la generación de software para la gestión de sistemas de energía solar*. Tesis Doctoral, Universidad de Málaga.
- Matla, E. (2014). *Desarrollo de software guiado por la Norma ISO/IEC 29110 y Scrum SIDEP V.2.0*. Tesis, Universidad Nacional Autónoma de México, México.
- Matthies, C. (2018). Scrum2kanban: integrating kanban and scrum in a university software engineering capstone course . En *In Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials* (págs. 48-55).
- Molina, J., & Zea, M. (2017). Metodologías de desarrollo en Aplicaciones Web. *Revista de Postgrado FaCE-UC*.
- Navarro, A., Fernández, J., & Morales, J. (2013). (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30-39.
- Navarro, M., Moreno, M., Aranda, J., Parra, L., & Rueda, J. (2016). Integración de Metodologías Ágiles y Arquitecturas de Software en el desarrollo de Sistemas

- de Información. En R. d. (RedUNCI), *Memorias del XVIII Workshop de Investigadores en Ciencias de la Computación*. Argentina.
- Palacios, A., & Merchán, V. (2014). *Guía de Fundamentos de Desarrollo de Proyectos de Software y Metodologías Ágiles*. Quito.
- Pita Fernández, S. P. (2002). Investigación cuantitativa y cualitativa. *fisterra*, 1.
- Planificación, S. N. (2021). *Plan de Creación de Oportunidades 2021 – 2025*. Ecuador.
- Pressman, R. (1990). *Ingeniería del Software: un enfoque práctico* (Segunda ed.). McGraw-Hill.
- Pressman, R. (2010). *Ingeniería del software. Un enfoque práctico*. (Séptima ed.). México: McGraw-Hill.
- Scrum Manager. (2014). *Gestión de proyectos Scrum Manager. Versión 2.5*. . EUA: Iubaris Info 4 Media S.L.
- Senn. (1992). *Análisis y Diseño de Sistemas de*. México: Mc Graw Hill.
- Sommerville, I. (2005). *Ingeniería del software* (Séptima ed.). España: Pearson Education S.A.
- Sugimori, Y., Kusunoki, K., Cho, F., & Uchikawa, S. (1977). Toyota production system and Kanban system materialization of just-in time and respect-for-human system. *The International Journal of Production Research*, 15(6), 553-564.
- Sutherland, K. S. (2020). *La Guía de Scrum*.
- Torres Samuel, M. &. (2013). Mejoras al proceso de planificación de proyectos de. *Compendium*.
- Whitten, J., Bentley, L., & Barlow, V. (2003). *Análisis y Diseño de Sistemas de Información* (Tercera ed.). Madrid: McGraw-Hill.
- Yépez, W., Primera, C., & Torres, M. (2013). Mejoras al proceso de planificación de proyectos de software usando el Modelo de Madurez de Capacidad Integrado (CMMI). *Compendium*, 22-47.

ANEXOS

Anexo 1: Observación para recolección de información



MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN OBSERVACIÓN PARA RECOLECCIÓN DE INFORMACIÓN

El siguiente formato de Observación forma parte del trabajo de investigación obtener el título de Máster en TIC's la que permitirá la recolección de datos para determinar el proceso de desarrollo de software.

Nombre de la empresa observada: Edina S.A.	
Periodo sujeto a revisión: 3 meses	
Tipo de observación: Directa	
Descripción de la observación: Teniendo en cuenta y respetando que la información y los recursos informáticos son activos que deben ser protegidos del acceso no autorizado, la manipulación y la destrucción en las empresas, se observará el modo en el que se desarrolla un producto de software en el departamento de desarrollo de sistemas de la empresa Edina S.A.	
Causa: El departamento no cuenta con un marco de trabajo estandarizado para la gestión de las etapas de sus desarrollos de software.	
Efectos: <ol style="list-style-type: none">1. La organización entrega productos de software en tiempos no indicados.2. Los productos de software tienen escenarios de error ya en producción.3. Se presenta un aumento del presupuesto por el desfase en entregas.4. Se presenta más carga horaria para el equipo de desarrollo para que pueda cumplir con los requisitos iniciales.5. Se presentan nuevos requisitos no planteados al inicio que son aceptados por el Gerente de Sistemas, el cual los acepta para poder disminuir un poco el problema causado por no entregar a tiempo el producto y por los distintos errores que aparecen en producción.6. Descontento y desmotivación en todo el equipo de desarrollo.	
Recomendaciones: La inclusión de marco de trabajo que permita gestionar de una mejor manera las etapas del desarrollo de software.	
Firma Gerente de Proyectos	Firma Observador

Anexo 2: Entrevista para recolección de información



MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN ENTREVISTA PARA RECOLECCIÓN DE INFORMACIÓN

La siguiente entrevista forma parte del trabajo de investigación para obtener el título de Máster en TIC's la que permitirá la recolección de datos para realizar el levantamiento de información relevante que nos sirva como base para la realización de este proyecto.

Fecha: _____

Nombre:			
Cargo:		Antigüedad:	

¿Se siente a gusto con la forma de trabajo con la que se lleva a cabo el desarrollo de software de la empresa Edina S.A.?

¿Por qué cree que se entregan proyectos fuera del tiempo estimado?

¿Cuál es la causa por la cual aparecen escenarios de error una vez que el producto está en producción?

¿Qué grado de complejidad tienen los proyectos de desarrollo que realizan en la empresa Edina S.A.?

ALTO

MEDIO

BAJO

Dentro de tu equipo de trabajo. ¿En qué etapa del proceso de desarrollo participas?

- | | |
|--------------------------------------|--------------------------|
| 1. Definición de requerimientos | <input type="checkbox"/> |
| 2. Diseño de software | <input type="checkbox"/> |
| 3. Implementación y desarrollo | <input type="checkbox"/> |
| 4. Integración y pruebas del sistema | <input type="checkbox"/> |
| 5. Operación y mantenimiento | <input type="checkbox"/> |
| 6. Administración del proyecto | <input type="checkbox"/> |
| 7. Otro | <input type="checkbox"/> |

¿El departamento de desarrollo de software de la empresa Edina S.A. emplea un marco de trabajo, para el desarrollo de software?

SI NO A VECES

¿Qué marco de trabajo emplea la empresa para el desarrollo de software?

CASCADA SCRUM OTRO Especifique.....

¿El marco de trabajo utilizado por el departamento de sistemas cubre con necesidades de la empresa en cuanto al desarrollo de software?

SI NO A VECES

¿Los procedimientos y métodos que utiliza Edina S.A. son los adecuados para el desarrollo de software?

SI NO DESCONOZCO

¿Considera usted que existe una guía de desarrollo de software acorde a las necesidades del departamento de desarrollo de software?

SI NO DESCONOZCO

De las siguientes metodologías de desarrollo. ¿Con cuáles ha trabajado?

- | | |
|------------------------------|--------------------------|
| 1. Modelo en cascada | <input type="checkbox"/> |
| 2. Modelo en espiral | <input type="checkbox"/> |
| 3. Modelo en V | <input type="checkbox"/> |
| 4. Scrum | <input type="checkbox"/> |
| 5. Kanban | <input type="checkbox"/> |
| 6. Programación extrema (XP) | <input type="checkbox"/> |
| 7. Otra..... | <input type="checkbox"/> |

¿Estaría usted dispuesto a utilizar otros marcos de trabajos para el desarrollo de software en la empresa?

SI

NO

¿Estaría de acuerdo que se haga un estudio de trabajos para estandarizar el desarrollo de software en el departamento?

SI

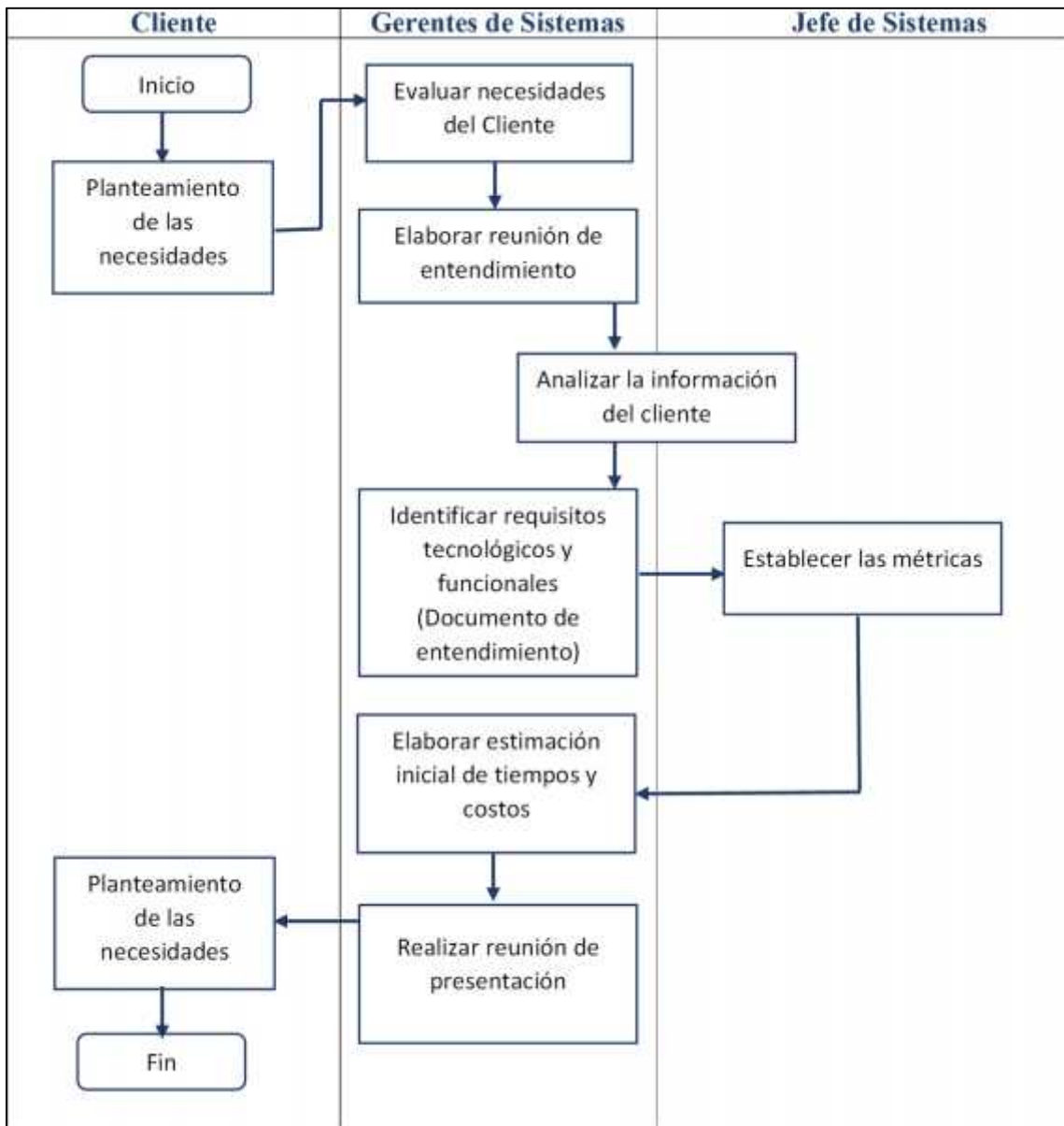
NO

¿Considera usted que con una estandarización para el desarrollo de software mejoraría el desenvolvimiento de la empresa?

SI

NO

Anexo 3: Diagrama de aceptación de propuesta de desarrollo



Anexo 4: Diagrama de marco de trabajo para el desarrollo de software

