

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

METODOLOGÍA HÍBRIDA PARA ESTANDARIZAR EL DESARROLLO DE APLICACIONES WEB, APLICANDO NORMAS ISO/IEC 25000 EN LA EMPRESA ELÉCTRICA AMBATO S.A. (EEASA).

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información.

ÁREA: Bases de datos y sistemas informáticos

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y sistemas de control

AUTOR: Bryan Santiago Jurado Acurio

TUTOR: Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

Ambato - Ecuador

agosto - 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: METODOLOGÍA HÍBRIDA PARA ESTANDARIZAR EL DESARROLLO DE APLICACIONES WEB, APLICANDO NORMAS ISO/IEC 25000 EN LA EMPRESA ELÉCTRICA AMBATO S.A. (EEASA), desarrollado bajo la modalidad Proyecto de Investigación por el señor Bryan Santiago Jurado Acurio, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.



Firmado electrónicamente por:
FRANKLIN OSWALDO
MAYORGA MAYORGA

Ing. Franklin Oswaldo Mayorga Mayorga Mg.
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: METODOLOGÍA HÍBRIDA PARA ESTANDARIZAR EL DESARROLLO DE APLICACIONES WEB, APLICANDO NORMAS ISO/IEC 25000 EN LA EMPRESA ELÉCTRICA AMBATO S.A. (EEASA) es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

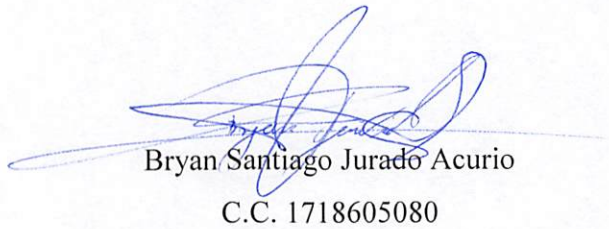
Ambato, agosto 2024.

Bryan Santiago Jurado Acurio
C.C. 1718605080
AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2024.



Bryan Santiago Jurado Acurio

C.C. 1718605080

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Bryan Santiago Jurado Acurio, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado METODOLOGÍA HÍBRIDA PARA ESTANDARIZAR EL DESARROLLO DE APLICACIONES WEB, APLICANDO NORMAS ISO/IEC 2500 EN LA EMPRESA ELÉCTRICA AMBATO S.A. (EEASA), nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.

Firmado electrónicamente por:
ELSA PILAR URRUTIA URRUTIA

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
DAVID OMAR GUEVARA AULESTIA

Ing. David Omar Guevara Aulestia, Mg.
PROFESOR CALIFICADOR

Firmado electrónicamente por:
RUBEN EDUARDO NOGALES PORTERO

Ing. Rubén Eduardo Nogales Portero, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Dedico el presente proyecto de investigación a mis padres Jorge Jurado y Martha Acurio, que han sido mi apoyo en todo momento, ellos son el pilar fundamental de mi vida, ya que han sido mi guía inculcándome sus principios, valores y de lo importante que es la perseverancia y empeño.

Bryan Santiago Jurado Acurio

AGRADECIMIENTO

En primer lugar, agradezco a Dios por concederme salud y vida durante esta etapa de mi vida, y por ayudarme a alcanzar una meta más.

Agradezco a mis padres por ser siempre incondicionales conmigo, por brindarme su sabiduría, valores y creer en mi durante todo este proceso.

A mis tíos por todo el amor y comprensión que me brindaron diariamente y ayudarme a crecer como persona.

A mis amigos, que a través de los años los he llegado a considerar como hermanos, y que han estado presentes en los momentos de alegría y aún más en los momentos difíciles con sus consejos y ayuda.

A mis tutores Ing. Franklin Mayorga, Ing. Diego Cadme, quienes que con su guía y ayuda se realizó el presente proyecto de investigación.

Bryan Santiago Jurado Acurio

ÍNDICE GENERAL DE CONTENIDOS

PORTADA ..... **i**

APROBACIÓN DEL TUTOR ..... **ii**

AUTORÍA ..... **iii**

DERECHOS DE AUTOR ..... **iv**

APROBACIÓN DEL TRIBUNAL DE GRADO ..... **v**

DEDICATORIA ..... **vi**

AGRADECIMIENTO ..... **vii**

ÍNDICE GENERAL DE CONTENIDOS ..... **viii**

ÍNDICE DE TABLAS ..... **xi**

ÍNDICE DE FIGURAS ..... **xiii**

ÍNDICE DE ANEXOS ..... **xv**

RESUMEN EJECUTIVO ..... **xvii**

ABSTRACT ..... **xviii**

CAPÍTULO I. MARCO TEÓRICO **1**

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 2

1.3 Fundamentación teórica 7

1.3.1 Clasificación y tipos de metodologías de desarrollo de software 7

1.3.2 Análisis de requerimientos de procesos 14

1.3.3 Requerimientos estratégicos 14

1.3.4 Requerimientos técnicos 15

1.3.5 Requerimientos de seguridad 15

1.3.6 Requerimientos de usabilidad 15

1.3.7 Modelos que establecen etapas durante un proyecto 16

1.3.8 Normas ISO/IEC 25000 17

1.3.9 Estándar IEEE 1074 22

1.4 Objetivos 25

1.4.1 Objetivo general 25

1.4.2 Objetivos específicos 25

CAPÍTULO II. METODOLOGÍA 26

2.1 Materiales 26

2.2 Métodos 29

2.2.1 Modalidad de la investigación 29

2.2.2 Población y muestra 29

2.2.3 Recolección de información 30

2.2.4 Procesamiento y análisis de datos 49

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 51

3.1 Análisis y discusión de resultados 51

3.1.1 Metodologías de desarrollo de aplicaciones 51

3.1.2 Estructura de las Normas ISO/IEC 25000 66

3.1.3 Estructura del Estándar IEEE 1074 79

3.2 Desarrollo de la propuesta 89

3.2.1 Diagnóstico empresarial 89

3.2.2 Metodología propuesta 90

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 115

4.1 Conclusiones 115

4.2 Recomendaciones 116

REFERENCIAS BIBLIOGRÁFICAS 118

ANEXOS 122

ÍNDICE DE TABLAS

Tabla 1. Características de la ISO/IEC 25010 18

Tabla 2. Grupos de Procesos del Estándar IEEE 1074 24

Tabla 3. Ficha de Observación 26

Tabla 4. Población 30

Tabla 5. Ficha de Observación del Departamento DRI 31

Tabla 6. Ficha de Observación del Departamento de Distribución 32

Tabla 7. Ficha de Observación del Departamento Comercial 33

Tabla 8. Encuesta Personal Informático Alfa de Cronbach 35

Tabla 9. Análisis de Metodologías Tradicionales 52

Tabla 10. Análisis de Metodologías Ágiles 57

Tabla 11. Estrategias de Trabajo Aplicada al Desarrollo Web 65

Tabla 12. Características y Sub-Características de las Normas ISO/IEC 25000 66

Tabla 13. Porcentaje de Ponderación de las Características a Evaluar 68

Tabla 14. Rangos de Calificación 68

Tabla 15. Porcentajes de Ponderación en Sub-Características 69

Tabla 16. Módulo de Adecuación Funcional 72

Tabla 17. Módulo de Fiabilidad 73

Tabla 18. Módulo de Eficiencia de Desempeño 74

Tabla 19. Módulo de Seguridad 75

Tabla 20. Módulo de Usabilidad 76

Tabla 21. Secciones Lógicas del Estándar IEEE 1074 79

Tabla 22. Estructura del Estándar IEEE 1074..... 81

Tabla 23. Requisitos Funcionales y No Funcionales 89

Tabla 24. Roles Dentro del Equipo de Desarrollo 92

Tabla 25. Estructuración de Ciclos de las Actividades de la IEEE 1074..... 94

Tabla 26. Formulario de Iteración de Usuario 101

Tabla 27. Primera Iteración de Usuario para Petición de un Nuevo Rol 110

Tabla 28. Segunda Iteración de Usuario para Agregar un Reporte..... 111

Tabla 29. Rangos de calificación 113

Tabla 30. Detalle de Fechas del Proyecto SISGERH-PASANTIAS 114

ÍNDICE DE FIGURAS

Figura 1. Metodología Prototyping 8

Figura 2. Normas ISO/IEC 25000 17

Figura 3. Características de la ISO/IEC 25020 20

Figura 4. Evaluación de calidad del producto de software 21

Figura 5. Uso de una Metodología 36

Figura 6. Importancia de actividades en el desarrollo de software 37

Figura 7. Problemas que se presentan en el desarrollo de un proyecto 38

Figura 8. Correcta recolección de requerimientos 39

Figura 9. Importancia del análisis y documentación en desarrollo de proyecto. 40

Figura 10. Revisiones constantes 41

Figura 11. Entregas en el tiempo establecido 42

Figura 12. Volver a fase de codificación al detectar deficiencias 43

Figura 13. Correcciones durante la etapa de codificación 44

Figura 14. Realizar evaluación de calidad de software 45

Figura 15. Normas ISO/IEC 25000 46

Figura 16. Conocimiento sobre las normas ISO/IEC 25000 47

Figura 17. Utilizar una metodología para superar inconvenientes 48

Figura 18. Enfoque de las Metodologías Tradicionales 51

Figura 19. Enfoque de las Metodologías Ágiles 56

Figura 20. Diagrama de Roles y Privilegios 99

Figura 21. Diagrama de Procesos 99

Figura 22. Esquema de Bases de Datos Relacional 100

Figura 23. Esquema del Ciclo de Vida de la Metodología Híbrida 105

Figura 24. Diseño de la Base de Datos de SISGERH-PASANTIAS..... 107

Figura 25. Diagrama para la Asignación de Roles..... 107

Figura 26. Diagrama de Procesos para Registro y Actualización de Información .. 108

Figura 27. Diseño de la Interfaz de Administración 108

Figura 28. Diseño de la Interfaz de Reportes..... 109

Figura 29. Módulo de Registro de Estudiantes 109

Figura 30. Módulo de Reportes..... 110

Figura 31. Registro de Estudiantes..... 111

Figura 32. Reconocimiento Económico por Pasantías..... 112

ÍNDICE DE ANEXOS

Anexo A. Diagrama de Procesos de Desarrollo con Inconvenientes 122

Anexo B. Formulario de Iniciación del Proyecto 122

Anexo C. Formulario de Encuesta 125

Anexo D. Formulario de Ficha de Observación 126

Anexo E. Cronograma de Actividades 127

Anexo F. Formulario de Exploración de Conceptos 129

Anexo G. Formulario para Aprobación de Inicio del Proyecto 131

Anexo H. Formulario de Requerimientos de Software e Implementación 132

Anexo I. Formulario de Evaluación e Instalación 134

Anexo J. Formulario de Pruebas 137

Anexo K. Formulario para el Pase a Producción 138

Anexo L. Formulario de Instalación y Mantenimiento 139

Anexo M. Desarrollo de Documentación y Capacitación 141

Anexo N. Aplicación de Ficha de Observación 142

Anexo O. Aplicación del Formulario Iniciación del Proyecto 143

Anexo P. Aplicación del Formulario Exploración de Conceptos 146

Anexo Q. Aplicación del Formulario para Aprobación del Inicio del Proyecto 148

Anexo R. Aplicación del formulario de Requerimientos de Software e Implementación 149

Anexo S. Aplicación del Formulario de Evaluación e Instalación 151

Anexo T. Aplicación del Formulario de Pruebas Módulo de Registro y Actualización 154

Anexo U. Aplicación del Formulario de Pruebas Módulo de Reportes..... 155

RESUMEN EJECUTIVO

El presente proyecto tiene como objetivo el desarrollo de una metodología híbrida para estandarizar el desarrollo de aplicaciones web en la EEASA, el proyecto surgió al no tener la documentación adecuada con la cual aplicar buenas prácticas, métodos, técnicas o estrategias para el desarrollo de software, lo cual ha ocasionado inconvenientes durante el desarrollo de un proyecto y en el manejo del ciclo de vida.

Para el cumplimiento de esta propuesta se hizo uso del estándar IEEE 1074 adaptándolo a la estructura organizacional de la empresa y a los requisitos obtenidos a través de las técnicas de recolección y análisis de información, con el cual se establece el estándar del ciclo de vida durante el desarrollo de una aplicación web, posteriormente se realizó un estudio basado en conocer y analizar procesos, herramientas y técnicas de trabajo de metodologías tradicionales y ágiles, de entre las cuales se identificó a las más representativas que son: Modelo en Cascada, Espiral, Proceso Racional Unificado (Rational Unified Process, RUP, en inglés), Método de Desarrollo de Sistemas Dinámicos (Dynamic Systems Development Method, DSDM, en inglés), SCRUM, Exteme Programming (XP) y Crystal. Después haber realizado un análisis cualitativo de las metodologías se evidenció que las metodologías ágiles se adaptan al desarrollo de aplicaciones web y a los requisitos planteados por la empresa.

Al trabajar con el estándar IEEE 1074, se consiguió plantear una metodología híbrida con un ciclo de vida que contiene grupos de actividades ordenadas y que se las puede asignar un cronograma claro y detallado, el cual el equipo de desarrollo puede seguir paso a paso, y con esto evitar el uso excesivo de tiempo durante el desarrollo de un proyecto o inconsistencias entre los requerimientos y objetivos planteados al momento de realizar la entrega del producto al departamento, sección o área requirente, y también al aplicar una evaluación de calidad del software con la familia de normas ISO/IEC 25000, que se encuentra dentro de las actividades que conforman la metodología, el equipo de desarrollo puede obtener un informe con el cual se comprueba que el producto que se entrega cumple con normas de calidad.

Palabras clave: Metodologías, tradicionales, ágiles, normas ISO/IEC 25000, estándar IEEE 1074, calidad, ciclo de vida.

ABSTRACT

This project aims to develop a hybrid methodology to standardize the development of web applications in EEASA, the project arose from not having adequate documentation with which to apply best practices, methods, techniques or strategies for software development, which has caused problems during the development of a project and in the management of the life cycle.

For the fulfillment of this proposal, the IEEE 1074 standard was used, adapting it to the organizational structure of the company and to the requirements obtained through the techniques of the collection and analysis of the information, with which the standard of the life cycle is established during the development of a web application, then a study was conducted based on knowing and analyzing processes, tools and work techniques of traditional and agile methodologies, among which the most representative ones were identified, which are; Waterfall Model, Spiral, Rational Unified Process (RUP), Dynamic Systems Development Method (DSDM), SCRUM, Extreme Programming (XP) and Crystal. After conducting a qualitative analysis of the methodologies, it became evident that the agile methodologies are adapted to the development of web applications and to the requirements set forth by the company.

By working with the IEEE 1074 standard, it was possible to propose a hybrid methodology with a life cycle that contains groups of ordered activities that can be assigned a clear and detailed schedule, which the development team can follow step by step, and thus avoid excessive use of time during the development of a project or inconsistencies between the requirements and objectives set at the time of delivery of the product to the department, also by applying a software quality evaluation with the ISO/IEC 25000 family of standards, which is part of the activities that make up the methodology, the development team can obtain a report that verifies that the delivered product complies with quality standards.

Keywords: Methodologies, traditional, agile, ISO/IEC 25000 standards, IEEE 1074 standards, quality, life cycle.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

METODOLOGÍA HÍBRIDA PARA ESTANDARIZAR EL DESARROLLO DE APLICACIONES WEB, APLICANDO NORMAS ISO/IEC 25000 EN LA EMPRESA ELÉCTRICA AMBATO S.A. (EEASA).

1.1.1 Planteamiento del problema

El desarrollo de software es una actividad diferente de toda la industria construida desde la revolución industrial. Las prácticas comunes en otras áreas de la ingeniería son ineficientes para el área informática, ya que un proyecto de software necesita un control empírico sobre el proceso, para esto la Ingeniería de Software es una ciencia en la que se aplican principios de computación y matemática para alcanzar soluciones con una mejor relación entre el coste y el beneficio para el problema [1].

Al principio el software eran programas pequeños, conforme la capacidad computacional fue mejorando, la complejidad del software desarrollado creció, este fue el inicio para empezar a utilizar técnicas, métodos, modelos y estrategias, que con el tiempo se les conoce como metodologías de desarrollo de software [1].

En la actualidad, las metodologías ágiles se usan con más frecuencia a nivel mundial en el desarrollo de software, estas metodologías se enfocan en el trabajo en equipo, la colaboración con el cliente y la entrega continua de software funcional y de alta calidad. Existen metodologías específicas que se utilizan como son Scrum, Kanban, Lean, Extreme Programming (XP) y también hacen uso de metodologías tradicionales como es el Modelo en Cascada y Espiral, todo depende de los requerimientos y el tipo de organización que se encuentra desarrollando el software [2].

Sin embargo, debido a la creciente demanda de software dentro de las organizaciones en diferentes países, ha provocado que el uso de las metodologías tradicionales y ágiles por separado dejen de ser altamente efectivas al momento de su aplicación en un proyecto de desarrollo, por lo que se ha optado en proponer metodologías híbridas, en las que se toma las herramientas, eventos, procesos, o estrategias de dos o tres metodologías, las cuales

se ajustan al proyecto que se va a desarrollar o a la estructura organizacional de la empresa, de esta forma teniendo como resultado un desarrollo versátil y limpio [2], [3].

Al ser una tendencia tanto en las empresas pequeñas, medianas y grandes dentro del estado ecuatoriano, optar por una metodología ágil para el desarrollo de software, una empresa como la EEASA, considerada una empresa grande, podría optar por utilizar metodologías ágiles para estandarizar el desarrollo de software, o una metodología híbrida en la que se trabaje con diferentes características que se adapten a las necesidades del proyecto.

Teniendo en cuenta de que el software desarrollado en la EEASA, va a ser utilizado por personal de la empresa y clientes, los cuales son ajenos no solo al proceso del desarrollo sino también del proceso organizacional de la Empresa, se podría obtener un apoyo considerable, al realizar una evaluación de calidad del software basándose en la norma ISO/IEC 25000.

1.2 Antecedentes investigativos

Después de haber realizado la investigación bibliográfica, se pudo encontrar trabajos de investigación y artículos de revistas académicas realizados en diferentes Universidades, que serán utilizados como base para la investigación: Según Hernán Naranjo y Erick Gamboa [4] en su trabajo de investigación establece que:

- La metodología Extreme Programming XP, se enfoca en el proceso de comunicación continuo, entre el cliente y el equipo de desarrollo, y así se puede simplificar el desarrollo de tareas y permite hacer frente a los cambios, por esta razón es utilizada en proyectos en los que existe un alto riesgo técnico.

- Scrum es una metodología diferente ya que está diseñada para proyectos en los que existe una alta posibilidad de tener cambios en los requerimientos iniciales, para esto cuenta con una estrategia llamada Sprints en las que se realiza entregas del software al cliente, para verificar su funcionalidad y proponer cambios en el caso de que existan.

Según Edison Álvarez y Blanca Lisintuña [5] en su trabajo de investigación se encuentra definido el concepto de lo que es la metodología RUP, se puede concluir que:

- El Proceso Unificado de Desarrollo (RUP, en inglés), se caracteriza principalmente por ser iterativo e incremental, y también es la metodología estándar más utilizada para el análisis, implementación y documentación de un proyecto.

- La metodología RUP, tiene como objetivo la asignación de tareas y responsabilidades de forma disciplinada e implementar buenas prácticas en Ingeniería de Software como:

    - Desarrollo iterativo.

    - Administración de requisitos.

    - Uso de arquitectura basada en componentes.

Según Alan Alva y John Reyes [6] dentro de su artículo de revista académica, se pudo determinar que:

- Al trabajar con la metodología Scrum se puede realizar un análisis de requerimientos exitoso, de una manera rápida e interactiva.

- Con el apoyo de XP se pudo hacer uso de sus estrategias dentro del equipo de trabajo durante el desarrollo de software, ya que es una metodología ágil basada en la simplicidad, comunicación, retroalimentación y valor.

- El contar con un sistema de ventas es de gran aporte para reducir de forma significativa el tiempo de los procesos que se llevan a cabo como por ejemplo el registro de un servicio o para elaborar un cierre semanal.

Según Erika Yépex y Kleber Armijos [7] dentro de su proyecto de investigación donde se aplica la metodología Kanban, se tiene una conclusión y recomendación con respecto a lo que es la aplicación de metodologías ágiles en el desarrollo de software:

- Una vez realizada la investigación, se puede concluir que la metodología ágil Kanban, permite ejecutar procesos con mayor rapidez y obtener un sistema de mejor calidad en poco tiempo, a diferencia del uso de metodologías tradicionales, ya que tienen un alto grado de flexibilidad y adaptabilidad.

- Para la aplicación de la metodología Kanban dentro de un proyecto de desarrollo de software, se recomienda tener en consideración los conocimientos y recursos necesarios, además de haberse realizado anteriormente un análisis de requerimientos y decidir la metodología que se va a utilizar.

- En el caso de una institución el tener un sistema informático para el control y gestión de los procesos que se deben llevar a cabo, es de gran importancia también realizar actualizaciones sobre el mismo, para ello se presenta un proyecto de desarrollo de software para la obtención de un nuevo módulo que se integra al sistema ya existente.

Según Oscar Capuñay y Juan Antón [8] dentro de su artículo de revista académica, una vez realizada la investigación sobre la metodología Scrum se concluye que:

- La metodología ágil Scrum influye significativamente, en el tiempo de entrega de los proyectos, aunque igualmente se presentan retrasos en algunas de las entregas, el promedio de tiempo es mucho menor frente al tiempo presentado en proyectos realizados sin Scrum.

- Scrum es una metodología ágil que ha logrado mejorar el rendimiento o calificación final de los proyectos de desarrollo de software.

Según Marcos Carrasco, William Ocampo, Luis Ulloa y Jon Azcona [9] en su artículo de revista académica, sobre la combinación de metodologías ágiles Scrum y XP en el desarrollo de software se pudo llegar a las siguientes conclusiones:

- Al haber utilizado historias de usuario como instrumento de recolección de datos, se logró recabar toda la información necesaria para el desarrollo de software, y junto con la frecuente comunicación con el Product Owner (Dueño del Producto), se generó la retroalimentación necesaria para resolver cualquier error.

- La combinación de la metodología XP y el marco de trabajo Scrum, han sido de gran ayuda durante el proceso de desarrollo de software. Los procesos y el ciclo de vida ideal que propone XP junto con los eventos y estrategias de Scrum aportaron para que el desarrollo sea versátil y limpio.

- Durante el desarrollo de software es importante aplicar los aspectos 13 básicos de la Ingeniería de Software, ya que esta ciencia se interesa por todas las etapas de la producción de software, desde las etapas más tempraneras hasta el mantenimiento que es una etapa que se aplica después de haber implementado el software.

Según Edwin Bautista Villegas [10] en su artículo de revista académica, al haber realizado la investigación sobre las metodologías ágiles XP y Scrum, empleadas para el desarrollo de páginas web se obtienen las siguientes conclusiones:

- Mientras que la metodología Scrum es la más idónea para la gestión de cambios en los requisitos durante el desarrollo de un proyecto, siendo capaz de adaptarse a los mismos de la manera más óptima, en cambio la metodología XP es la más idónea para desarrollar aplicaciones Web en el menor tiempo posible y con la funcionalidad necesaria.

- Las organizaciones presentan una alta necesidad de gestionar procesos de forma más eficiente, para esto el desarrollo de aplicaciones Web se ha considerado como un apoyo, siempre y cuando el desarrollo del proyecto se lo realice aplicando metodologías que se ajusten a los requerimientos y al final obtener un producto de calidad.

Según Margarita Ramírez, María del Consuelo Soto, Hilda Moreno, Esperanza Rojas, Nora del Carmen Millán y Ricardo Cisneros [11] dentro de la investigación realizada en su artículo académico sobre la metodología Scrum en el desarrollo de software, se tiene que:

- El uso de Scrum permitió realizar una distribución adecuada de responsabilidades y tareas entre los miembros del equipo, dar seguimiento puntual a los avances de las metas, y poder realizar los ajustes necesarios.

Según Erika Reina y Susana Patiño [12] en su artículo de revista académica, donde se realizó una investigación sobre cómo aplicar la norma ISO/IEC 25000, en la evaluación de calidad del software, se concluye que:

- Al realizar la evaluación de calidad del software y al haber aplicado las métricas propuestas en el marco de trabajo especificado en el estándar ISO/IEC 25000, se puede evaluar si las características y las subcaracterísticas cumplen con la calidad propuesta sobre el software.

Según Eder Figueroa, Gilberto Carrión, Manuel Chero, Andrés Figueroa y Walter Figueroa [13] en su artículo de revista académica, sobre un modelo de calidad basado en las normas ISO/IEC 25000, se obtuvo la siguiente información:

- Para medir y mejorar la calidad de una plataforma se debe tener en cuenta los siguientes atributos: estandarización, flexibilidad, funcionalidad, interactividad, accesibilidad, usabilidad, escalabilidad, y para lograr evaluar cada uno de esos atributos se debe hacer uso de una o más características en la documentación presentada sobre la familia de normas ISO/IEC 25000.

1.3 Fundamentación teórica

1.3.1 Clasificación y tipos de metodologías de desarrollo de software

Debido a que durante el paso del tiempo hubo una mayor demanda de software, se tenía necesidad de estructurar metodologías que se adapten al tipo de proyecto que se va a realizar, ya que durante el análisis de requerimientos es lo que se demandaba. Debido a esto empezaron a aparecer las denominadas metodologías tradicionales [14], [15].

a. Metodologías tradicionales

Las metodologías tradicionales tienen la principal característica de ser mucho más rígidas y se las debe trabajar con un equipo de desarrollo bien estructurado. Generalmente las metodologías tradicionales son utilizadas por organizaciones gubernamentales [14], [16].

- ***Metodología en Cascada***

La metodología en cascada definida por Winston Royce, sugiere un enfoque sistemático y secuencial para el desarrollo de software. Tiene más disciplina y se basa en el análisis, diseño, pruebas y mantenimiento.

Esta metodología consiste en un proceso secuencial y lineal para la gestión de proyectos, esto la hace una metodología estricta ya que consiste en varias fases separadas y ninguna fase comienza hasta que se haya concluido con la fase anterior, y la finalización de cada fase es definitiva, esto quiere decir que no permite regresar a una fase anterior [14].

- ***Metodología Espiral***

A pesar de que el modelo en cascada, prevalecía y era el más utilizado, Barry W Boehm fue quien propuso el desarrollo en espiral, que a diferencia del modelo “Code and Fix” y el modelo en cascada, el modelo en espiral se basa en la identificación y solución de riesgos.

El enfoque se centra en los posibles factores que pueden causar incertidumbres para el software o para todo el proyecto, en el supuesto caso de existir los riesgos y si es que pueden ser controlados de una manera rentable, no habría nada que impida la finalización exitosa del proyecto.

Para minimizar los riesgos se realizan prototipos, simulaciones, pruebas de referencia o entrevistas con los usuarios. Para cierto tipo de riesgos también se realiza una revisión de todo el procedimiento y estructurado de manera diferente, de ser necesario también puede intervenir la gerencia o management en cada fase del proyecto [14].

- ***Proceso Unificado Racional (RUP)***

En junio de 1998 se presenta el Proceso Unificado Racional, siendo un método de desarrollo iterativo, que especifica principalmente, la constitución del equipo y las escalas de tiempo, así como número de modelos de documentos.

RUP tiene un enfoque práctico que describe buenas prácticas de ingeniería de software, que se recomienda para su uso en el desarrollo de sistemas [14].

- ***Prototipo (Prototyping, en inglés)***

El prototipado es una técnica que permite realizar y materializar diversas ideas de soluciones propuestas en un proyecto de diseño o rediseño de productos y servicios. Se puede aplicar el prototipado como parte del proceso de innovación y diseño [17].

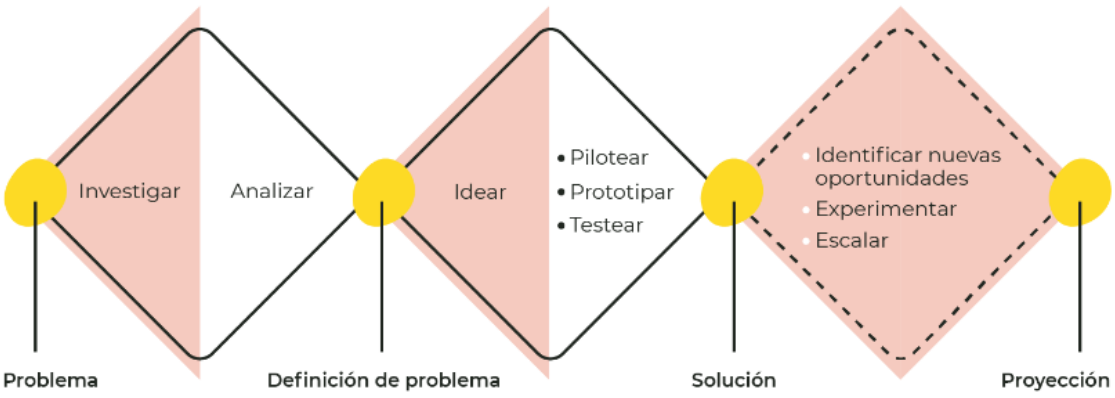


Figura 1. Metodología Prototyping

Fuente: FREED [18]

La metodología de prototipado está relacionada con la mejora continua y el ciclo de Deming que consiste en un proceso iterativo enfocado en diseñar, implementar, medir y ajustar un plan. En el caso de un prototipo se puede aplicar 5 etapas en el diseño e implementación de una idea [17].

Etapas para diseñar e implementar un prototipo:

- Definir los requerimientos y variables, conocer cuál es el propósito del prototipo, y que elementos se requieren.

- Definir las herramientas para el diseño, las cuales pueden ser herramientas que requieran o no conocimientos sobre diseño o código.

- Diseñar la idea del prototipo, y analizar el tipo de prototipo más adecuado tomando en cuenta factores como diseño, ergonomía, formas, dimensiones, todos aquellos que resulten en un diseño cómodo para el usuario.

- Testeo del prototipo, dar a conocer el prototipo a los usuarios, en esta etapa se debe observar y registra el desempeño del prototipo y permitir al usuario aportar ideas.

- Analizar los resultados y aprendizaje, a partir del desempeño del prototipo, se debe perfeccionar el concepto de solución.

Al tener la disponibilidad de las metodologías tradicionales, ya se podía realizar un desarrollo de software más estructurado, pero al existir una alta demanda del software mucho más moderno y de alta calidad, se empieza a diseñar y estructurar metodologías modernas, llamadas metodologías ágiles.

b. Metodologías Ágiles

El uso de prototipos rápidos, procedimientos y normas evolucionó al uso del concepto de la Metodología del Proceso de la Ingeniería de Software, destacando la utilización de las metodologías ligeras y completas. Es así como se introduce el manejo de las Metodologías de Desarrollo Ágil, contemplando un conjunto de métodos, que se basan en el desarrollo iterativo e incremental, teniendo presente los cambios y respondiendo a estos mediante la colaboración de los desarrolladores [14].

- ***Metodología Extreme Programming (XP)***

La metodología XP fue fundad por Ken Beck, en la que se empezó por identificar qué era lo difícil y lo simple al momento de programar. Se encontró que la clave del éxito es potenciar las relaciones interpersonales durante el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores [14], [15].

Al ser una metodología fundamentada en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas, ha contribuido como método a la generación efectiva de software. También promueve una adecuada práctica para proyectos con requisitos imprecisos y muy cambiantes [15], [16].

- ***Metodología Scrum***

Scrum está enfocada en ser adaptable, rápido, iterativo, eficaz y flexible, de esta forma dar valor a un proyecto de una manera ágil y rápida. Scrum permite mejorar la comunicación y crea un entorno de responsabilidad común entre los participantes del proyecto, con esto lograr una tasa de avance de forma constante [15], [16].

El desarrollo de Scrum se fragmenta en pequeños períodos de tiempo de trabajo, llamados Sprints. Con la estrategia de los sprints, se permite un alto ritmo de trabajo, para esto también es necesario que los equipos de trabajo sean altamente cualificados y con las autorizaciones suficientes para tomar decisiones [14], [15].

- ***Metodología Kanban***

La metodología Kanban es sencilla, se puede actualizar y los equipos de trabajo la pueden asumir sin problema, al ser un método visual en el que se utiliza tarjetas en las que se definen tareas y el flujo de trabajo, de esta forma es mucho más fácil evaluar el estado en el que se encuentra el proyecto y si es factible añadir más tareas [15].

Para que esta metodología se diferencie de las demás tiene características como; visualización de todas las tareas de tal forma que se encuentran expuestas en un tablero y se pueden identificar rápidamente, también es un sistema de mejora para el desarrollo de proyectos basado en los objetivos que se quieren conseguir, prioriza las actividades dentro

del desarrollo del tal manera que logra reducir o eliminar lo que es secundario del proyecto y provee la capacidad para dar una respuesta efectiva a tareas imprevistas [15].

- ***Desarrollo Rápido de Aplicaciones (RAD, en inglés)***

Este método fue diseñado para responder a la necesidad de entregar sistemas rápidamente. El método tiene una lista de tareas y una estructura de trabajo diseñada para la rapidez, comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering).

Tradicionalmente el desarrollo rápido de aplicaciones tendría tendencia a englobar también la usabilidad, utilidad y rapidez de ejecución [14].

- ***Crystal de Cockburn***

El método Crystal fue desarrollado por Alistair Cockburn, Quien decidió no centrarse en estrategias de desarrollo paso a paso, sino desarrollar la colaboración y la comunicación en equipo, algunas de las características de Crystal es que; el proyecto debe ser flexible, debe ser con enfoques adaptativos, es decir, no tiene herramientas fijas ya que pueden cambiar para satisfacer las necesidades específicas del equipo y es ultraligera esto implica que, no requiere mucha documentación [19], [20].

Entre las propiedades del método Crystal tenemos las siguientes:

- Entregas frecuentes, permite entregar regularmente los productos y el código de prueba a usuarios reales.

- Mejora reflexiva, esto quiere decir que los equipos siempre pueden implementar mejoras en sus prácticas futuras, dado que siempre hay áreas en las que se puede mejorar el producto.

- Comunicación osmótica, al tener los equipos en la misma fase física es muy importante ya que permite que la información fluya entre los miembros de un equipo.

- Seguridad personal, no existen malas sugerencias en un equipo de Crystal, los miembros del equipo deben sentirse seguros para discutir ideas abiertamente.

- Enfoque, cada miembro del equipo debe saber exactamente lo que tiene que hacer, lo que les permite enfocar su atención, esto aumenta la interacción del equipo y trabaja hacia un mismo objetivo.

- Fácil acceso a usuarios expertos, mejora la comunicación del equipo con los usuarios y se recibe comentarios regulares de usuarios reales.

- Herramientas técnicas, el equipo de desarrollo de software debe disponer de herramientas técnicas muy específicas que se utilizarán durante las pruebas, esto le permite al equipo identificar cualquier error en menos tiempo.

- ***DSDM***

El Método de Desarrollo de Sistemas Dinámicos (DSDM o Dynamic Systems Development Method, en inglés), es un método ágil que incorpora todo el ciclo de vida de un proyecto, la piedra angular de la filosofía del DSDM es la alineación del proyecto con los objetivos estratégicos de la organización [21].

El objetivo de un proyecto DSDM es satisfacer las necesidades de la empresa y ofrecer beneficios comerciales reales, además de que es un enfoque ágil que permite permanecer flexible y cumplir con los cambiantes requerimientos del negocios [22], [23].

Existen seis fases principales en el marco del DSDM:

- Pre – proyecto, durante esta fase la organización se asegura de que el proyecto esté alineado con las metas estratégicas y tenga un objetivo claramente definido.

- Viabilidad, durante la fase de factibilidad, la organización se asegura de que el proyecto sea técnicamente viable y rentable.

- Cimientos, son los fundamentos con los que se establece una comprensión de alto nivel de la justificación del proyecto. Se define el alcance del proyecto y se llega a un acuerdo sobre cómo, por quién, cuándo y dónde se llevará a cabo el proyecto.

- Desarrollo evolutivo, la solución del proyecto es desarrollada en detalle por el equipo de desarrollo de soluciones.

- Despliegue, esta fase consiste de tres actividades principales: ensamblar, revisar y desplegar. La solución puede desplegarse en su totalidad o por partes.

- Post – proyecto, en esta fase la organización evalúa si los beneficios esperados del proyecto han sido entregados.

- ***Design Sprint***

Design Sprint es una metodología de desarrollo de producto eficiente que permite construir y probar un prototipo en tan solo cinco fases, es considerada como una metodología ágil y es una nueva forma de probar ideas y acelerar su puesta en marcha, para esto se introdujeron algunas características de otras metodologías ágiles para obtener varias mejoras y seguir optimizando procesos [24].

Su objetivo principal es seguir un proceso de creación muy rápido y efectivo, a partir de una serie de pasos bien estructurados, con el fin de validar ideas de negocio descubrir sus fortalezas y dificultades o retos que presenta, en el menor tiempo posible.

La metodología Design Sprint se divide en cinco etapas clave:

- Comprensión y planificación, en esta etapa se debe enfocar en comprender el proyecto en cuestión y definir el objetivo que se debe cumplir, acotar y poner límites será clave para cumplir con la planificación.

- Boceto, En esta etapa el equipo dedicará su tiempo a generar una serie de bocetos o borradores que puedan dar solución al problema planteado. El objetivo será genera ideas diferentes para atacar el problema desde varios enfoques y encontrar soluciones innovadoras.

- Decisión, El equipo deberá contar con una serie de ideas y soluciones que deberían permitir alcanzar el objetivo que se ha planteado, analizar y decidir cuál de las soluciones tendrán mayores posibilidades de tener éxito.

- Prototipo, en esta etapa se deberá construir la idea seleccionada, diseñar un prototipo de alta calidad para convertir la idea en algo tangible y funcional.

- Prueba, esta última etapa el equipo estará dedicado a la prueba del prototipo con los usuarios, el equipo recoge los comentarios de los usuarios y valora si el prototipo construido soluciona el problema inicial. Para finalizar se realizará una valoración final, en la que se determinará si se ha cumplido con el objetivo.

1.3.2 Análisis de requerimientos de procesos

En el momento en el que se empieza a trabajar sobre el desarrollo de programas, surgieron métodos que permitían llevar a cabo un proyecto, con un producto de software calificado como bueno, pero las metodologías aplicadas eran simples, los métodos estaban únicamente desarrollados hacia los procesos [14].

El modelo de procesos fue predominante, pero existía inconformidad entre los usuarios finales, por la misma razón de que el modelo no tenía en consideración los datos sino solo procesos, resultó ser un modelo poco eficiente [14].

Entonces el software fue muy criticado, por su usabilidad y por la forma en la que se debía ejecutar algunas tareas, por lo que la solución a los problemas presentados fue realizar un proceso de documentación de requerimientos junto con las personas interesadas en el software y posteriormente realizar un análisis de los mismos, esto se conoció como una de las primeras etapas durante el desarrollo de un proyecto [14].

1.3.3 Requerimientos estratégicos

Los requerimientos estratégicos se pueden deducir que son los requerimientos funcionales y los no funcionales.

Los requerimientos funcionales describen acciones específicas que el ingeniero de software debe ser capaz de realizar durante el desarrollo de software. Los requerimientos funcionales suelen dividirse en reglas de negocio y casos de uso, eso quiere decir que las reglas de negocio son declaraciones de alto nivel que definen lo que un sistema debe hacer, mientras que los casos de uso son descripciones detalladas de cómo debe funcionar un sistema [17].

Los requerimientos no funcionales describen características específicas que el software debe poseer durante el desarrollo de la aplicación, y se divide en categorías de: rendimiento, seguridad y calidad [17].

1.3.4 Requerimientos técnicos

Los requerimientos técnicos se habrán definido en gran parte en los requerimientos funcionales y no funcionales, pero en el momento en el que el ingeniero de software decida iniciar con el proyecto, es necesario tener en cuenta que se deben discutir requerimientos más técnicos que tienen que ver con las tecnologías a usarse, las cuales pueden estar ya disponibles o plantear introducir nuevas prácticas o tecnologías [25]. Algunos de los requisitos más comunes se incluyen:

- Las características y funcionalidades deseadas en el producto.
- Plataformas para desarrollar aplicaciones, por ejemplo, iOS, Android y web.
- Especificaciones de diseño, la interacción del usuario con el software.
- Funcionalidad de back-end: integración de API y bases de datos.
- Planificación de actividades y plazo de finalización.

1.3.5 Requerimientos de seguridad

Los requerimientos de seguridad especifican las medidas que se deben tomar al momento de desarrollar o gestionar un sistema para proteger los datos del acceso no autorizado. En la mayoría de los casos los requerimientos de seguridad especifican el nivel de protección, como confidencial o de alto secreto. Esto implica que se debe realizar una definición de roles para autenticación y cifrado [26].

1.3.6 Requerimientos de usabilidad

Los requerimientos de usabilidad se dividen en rendimiento y calidad, con esto se puede medir si la solución que se ha desarrollado cumple con los objetivos planteados al principio con los cuales se pretende solventar los problemas que se han analizado.

- Requerimientos de calidad, especifican el nivel de calidad que debe cumplir un sistema, en ciertos casos se pueden especificar métodos que se van a utilizar para medir la calidad. Para obtener los resultados de la densidad de defectos o la satisfacción del usuario se cuenta con cuatro medidas de calidad: conformidad, usabilidad, confiablidad y mantenibilidad [25].

- Requerimientos de rendimiento, suelen dividirse en dos categorías: tiempo de respuesta y rendimiento. El tiempo de respuesta es la cantidad de tiempo que tarda el sistema en responder a la solicitud de un usuario, mientras que el rendimiento es el número de solicitudes que el sistema puede manejar. Estas medidas suelen ser más críticas para los sistemas interactivos y las aplicaciones web, en las que los usuario esperan respuestas inmediatas [26].

1.3.7 Modelos que establecen etapas durante un proyecto

Al ya tener la primera etapa que es la recolección, documentación y análisis de requerimientos, como continuación de la solución y un consenso para la construcción centralizada de software, empezó a aparecer el ciclo de vida de desarrollo de software [14].

Es así, como la ingeniería de software se comienza a implantar y a valerse de una serie de modelos que establecen y muestran las diferentes etapas y estados por los que pasa un producto de software, a estos modelos se les denomina “Modelos de ciclo de vida del software” [14].

Dentro de los Modelos de ciclo de vida del software se encuentran evidenciadas las siguientes etapas; concepción inicial, desarrollo, puesta en marcha, mantenimiento, y en algunos casos la retirada del producto [14].

1.3.8 Normas ISO/IEC 25000

La familia de normas ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation, en inglés), es un conjunto de normas que sirve para evaluar la calidad del software, y son el resultado de la evolución de normas anteriores, principalmente de las normas ISO/IEC 9126 y de forma complementaria las normas ISO/IEC 9126, por lo que se obtienen cinco divisiones [27], [28].



Figura 2. Normas ISO/IEC 25000

Fuente: ISO 25000 [28]

a. ISO/IEC 2500n – División de Gestión de Calidad

En la división de gestión de calidad se definen modelos, términos y definiciones para los siguientes estándares que los conforman:

- ***ISO/IEC 25000 – Guía SQuaRE:***

Establece la terminología y los modelos de referencia en la arquitectura SQuaRE.

- ***ISO/IEC 25001 – Planificación y Gestión:***

Establece los requisitos para gestionar en la evaluación y especificación de los requisitos del software.

b. ISO/IEC 2501n – División de Modelo de Calidad

En esta división se presenta modelos de calidad donde interviene calidad interna, externa y el uso del producto del software, y se encuentra conformada por:

- ***ISO/IEC 25010***

Modelo de calidad para el producto y la calidad en operación, presenta características y sub-características para evaluar el software. Determina la calidad interna y externa de un producto de software, y está compuesta por ocho características. [29], [28].

Tabla 1. Características de la ISO/IEC 25010

Fuente: ISO 25000 [27]

Factor	Descripción	Criterios
Adecuación funcional	Capacidad del producto para proporcionar funciones que satisfacen los requerimientos declarados.	Completitud funcional Corrección funcional Pertinencia funcional
Eficiencia de desempeño	Grado de utilización de recursos bajo determinadas condiciones.	Comportamiento temporal Utilización de recursos Capacidad
Compatibilidad	Capacidad de intercambio de información que comparte el mismo hardware.	Coexistencia Interoperabilidad
Usabilidad	Capacidad para que el producto sea entendible, aprendido, usado y resultar atractivo para el usuario.	Capacidad de aprendizaje Protección contra errores de usuario

		Estética de interfaz de usuario Accesibilidad
Fiabilidad	Capacidad del sistema para realizar sus funciones cuando se use bajo condiciones y en un periodo determinado.	Madurez Disponibilidad Tolerancia a fallos
Seguridad	Capacidad para la protección de datos.	Confidencialidad Integridad Responsabilidad Autenticidad
Mantenibilidad	Capacidad del software para la evolución y modificación debido a las necesidades.	Modularidad Reusabilidad
Portabilidad	Capacidad del software para ser transferido de forma efectiva a otro entorno hardware y software.	Adaptabilidad Capacidad para ser instalado

- ***ISO/IEC 25012***

Modelo de calidad para los datos, precisamente para aquellos que están alojados de manera estructurada y que forman parte de un sistema [29], [28].

c. ISO/IEC 2502n – División de Medición de Calidad

En la división de medición de calidad se incluye un modelo de referencia de calidad del software, con mediciones internas, externas y en uso.

- ***ISO/IEC 25020 – Guía y Modelo de Referencia para la medición***

Proporciona una explicación introductoria a los elementos de medida de calidad, crea guías para los usuarios.

- ***ISO/IEC 25021 – Elementos de Medida de Calidad***

Define y especifica métricas base y derivadas que puede ser utilizadas en el proceso de desarrollo del software.

- ***ISO/IEC 25022 – Medición de Calidad en Uso***

Especifica las métricas que se utilizaran para la medición en el uso del software.

- ***ISO/IEC 25023 - Medición de la Calidad del Producto Software y Sistemas***

Especifica la medición de la calidad del software.

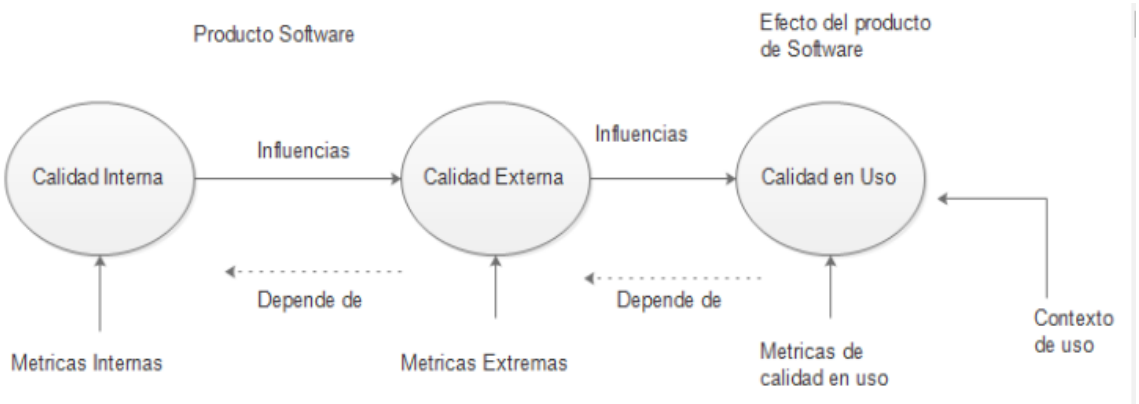


Figura 3. Características de la ISO/IEC 25020

Fuente: ISO 25000 [27]

- ***ISO/IEC 25024 – Medición de la Calidad de Datos***

Describe métricas para la medición de datos.

d. ISO/IEC 2503n – División de Requisitos de Calidad

Esta división especifica los requisitos de calidad que pueden ser utilizados en la entrada en el proceso de evaluación.

- ***ISO/IEC 25030 – Requisitos de Calidad***

Este estándar permite especificar los requisitos de calidad del software.

e. ISO/IEC 2504n – División de Evaluación de Calidad

Las normas que pertenecen a la división de evaluación de calidad son las que aportan con requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto de software, y se encuentra conformado por:

- ***ISO/IEC 25040 – Guía y Modelo de Referencia de la Evaluación***

Establece la referencia para la evaluación, que tiene en cuenta las entradas, restricciones y recursos en el proceso de evaluación, para obtener las salidas.

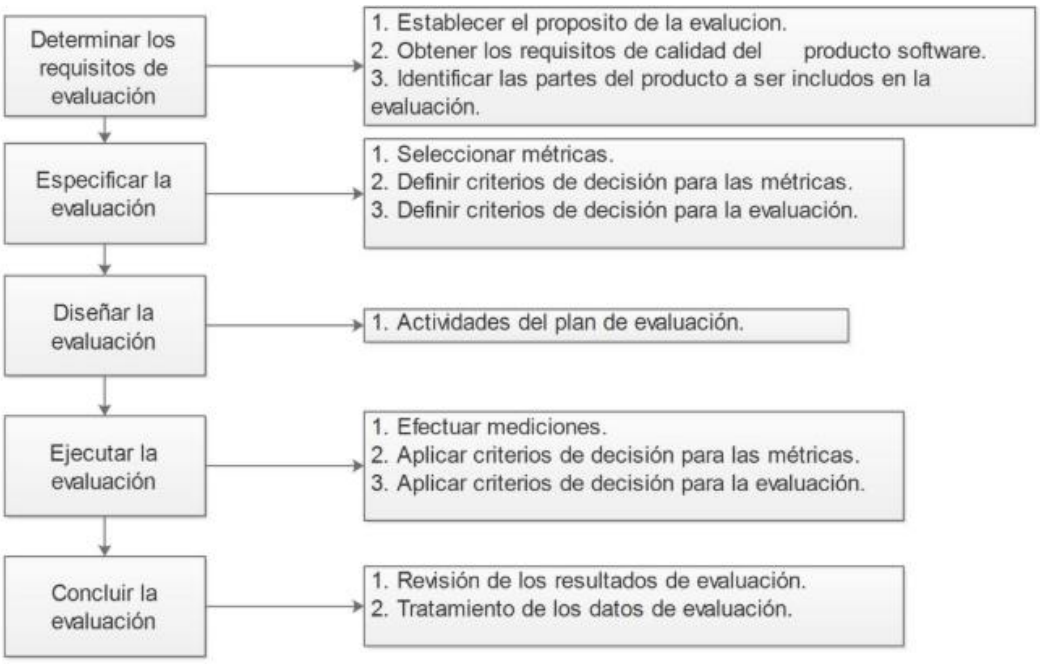


Figura 4. Evaluación de calidad del producto de software

Fuente: ISO 25000 [27]

- ***ISO/IEC 25041 – Guía de Evaluación para Desarrolladores***

Pondera requisitos y recomendaciones para la práctica de evaluación del producto software desde el punto de vista tanto del desarrollador.

- ***ISO/IEC 25042 – Módulos de Evaluación***

Consiste en establecer la documentación de los módulos de evaluación.

- ***ISO/IEC 25045 – Módulo de Evaluación para Recuperabilidad***

Define un módulo para la evaluación de los sistemas que ejecutan transacciones concurrentes.

1.3.9 Estándar IEEE 1074

El estándar IEEE 1074, fue desarrollado por IEEE en 1991, para determinar los procesos de vida del software, describe el conjunto de actividades y procesos obligatorios para el desarrollo y mantenimiento del software [30], [31]. Los procesos se dividen en 6 secciones lógicas o grupos de procesos, de esta forma se tiene 17 grupos con un total de 65 actividades:

a. Proceso de Modelo del Ciclo de Vida del Software

El proceso de modelo del ciclo de vida del software es el primer grupo, el cual proporciona actividades que se necesitan para identificar los modelos del ciclo de vida del software.

b. Proceso de Gestión del Proyecto

El segundo proceso que es, el proceso de gestión del proyecto, propone un conjunto de procesos de iniciación, supervisión y control del proyecto a lo largo del ciclo de vida del software, teniendo en cuenta los siguientes grupos que forman parte de este proceso:

- Iniciación del proyecto
- Planificación del proyecto
- Monitoreo y control del proyecto

c. Procesos Orientados al Desarrollo

Este grupo se centra principalmente en los procesos para el desarrollo del software y está compuesto por tres procesos de los cuales cada uno se complementa con grupos específicos para el desarrollo de sus actividades:

- ***De Pre-Desarrollo***
  - Exploración de conceptos
  - Asignación al sistema
  - Importación al software
- ***De Desarrollo***
  - Requerimientos de software
  - Diseño
  - Implementación
- ***De Post-Desarrollo***
  - Evaluación
  - Instalación
  - Operación y soporte
  - Mantenimiento
  - Retiro (dar de baja)

d. Procesos Integrales

En este último grupo de procesos integrales o de soporte, se encuentran aquellos procesos que se necesitan para completar con éxito las actividades y que contiene los siguientes grupos:

- Gestión de configuración del software
- Desarrollo de documentación
- Capacitación

Tabla 2. Grupos de Procesos del Estándar IEEE 1074

Grupos de Procesos	
Modelamiento del ciclo de vida	Selección de un ciclo de vida
Gestión del proyecto	Iniciación del proyecto Planificación del proyecto Monitoreo y control del proyecto
Pre-Desarrollo	Exploración de conceptos Asignación al sistema
Desarrollo	Requerimientos de software Diseño Implementación
Pos-Desarrollo	Evaluación Instalación Operación y soporte Mantenimiento

	Retiro (dar de baja)
Procesos Integrales	Gestión de configuración de software Desarrollo de documentación Capacitación

1.4 Objetivos

1.4.1 Objetivo general

Proponer una metodología híbrida para estandarizar el proceso de desarrollo de aplicaciones Web en la Empresa Eléctrica Ambato S.A. (EEASA), aplicando las normas ISO/IEC 25000

1.4.2 Objetivos específicos

- Identificar los requerimientos estratégicos, técnicos, de seguridad y usabilidad para el desarrollo de aplicaciones web de la EEASA.
- Investigar metodologías tradicionales y ágiles de desarrollo de software, para analizar las mejores técnicas, herramientas o estrategias, para integrarlas en una metodología híbrida.
- Plantear una metodología híbrida que se acople al desarrollo de aplicaciones web, aplicando las norma ISO/IEC 25000.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para recolectar la información del presente proyecto de investigación se aplicó una encuesta dirigida al personal informática, y una ficha de observación, para obtener información sobre las dificultades que se presentan al realizar un proyecto de desarrollo web en la EEASA.

Ficha de observación aplicada al personal informático.

Tabla 3. Ficha de Observación

Indicadores	**Siempre**	**Casi Siempre**	**A veces**	**Nunca**	**Observaciones**
Se realiza una correcta recolección y análisis de requerimientos.					
Se obtiene un modelamiento y diseño de base de datos adecuado.					
Se tiene un objetivo claro de lo que se quiere lograr con la aplicación web.					
Se elabora un efectivo desarrollo de la aplicación web (codificación) en el tiempo indicado.					
Realiza entregas constantes sobre el avance del proyecto durante el desarrollo del mismo.					
Durante las revisiones del desarrollo del proyecto se realiza una retroalimentación, para verificar que el proyecto continúa enfocado a los requerimientos y al objetivo principal.					
Se realiza cambios en los requerimientos iniciales del proyecto durante el desarrollo de la aplicación web (codificación, pruebas).					
Conclusiones:					

Encuesta aplicada al personal informático.

1. **¿Utiliza usted una metodología de desarrollo de software, en el diseño y desarrollo de soluciones informáticas de tipo web, en la EEASA?**

    - Si
    - No

2. **Al iniciar el desarrollo de una aplicación web, ¿Para usted cuál de las siguientes actividades tiene mayor importancia?**

    - Modelamiento y diseño de la Base de Datos.
    - Recolección y análisis de requerimientos.
    - Diagramas de flujo para establecer actividades en la Aplicación Web.
    - Desarrollo de la Aplicación Web (Codificación).

3. **¿Cuál es el mayor problema que usted ha tenido al momento de realizar proyectos de desarrollo de aplicaciones web?**

    - Descubrir falencias en la aplicación web al realizar pruebas.
    - Uso excesivo del tiempo al no tener tareas planificadas.
    - El proyecto podría quedar inconcluso, al no haber planteado un objetivo claro.
    - Realizar cambios imprevistos en los requerimientos iniciales del proyecto.

4. **¿Se realiza una correcta recolección de requerimientos, enfocada al objetivo principal de los proyectos de desarrollo de aplicaciones web?**

    - Si
    - No

5. **¿Para usted qué grado de importancia tiene el análisis y la documentación de los proyectos de desarrollo de aplicaciones web?**

    - Muy importante
    - Importante
    - Poco importante
    - No es importante

6. **¿Considera usted que se debe realizar revisiones constantes durante la fase de codificación, en los proyectos de desarrollo de aplicaciones web?**

    - Totalmente de acuerdo
    - De acuerdo
    - En desacuerdo
    - Totalmente en desacuerdo

7. **¿Durante la etapa de desarrollo de aplicaciones web se realizan entregas según el cronograma establecido para su revisión y/o pruebas?**

    - Si
    - No

8. **¿Cuándo considera usted óptimo volver a la fase de codificación en el caso de detectar deficiencias en la aplicación web durante la fase de pruebas?**

    - Siempre
    - Casi siempre
    - A veces
    - Nunca

9. **¿Está usted de acuerdo en que las correcciones de las deficiencias detectadas en la aplicación web, se las debe realizar durante las revisiones en la etapa de codificación?**

    - Totalmente de acuerdo
    - De acuerdo
    - En desacuerdo
    - Totalmente en desacuerdo

10. **¿Considera usted que es importante realizar una evaluación de calidad de software, durante la fase de pruebas?**

    - Si
    - No

11. **¿Está usted familiarizado con las Normas ISO/IEC 25000?**

    - Si
    - No

12. **¿En qué nivel considera usted que se encuentra su conocimiento sobre las normas ISO/IEC 25000?**

    - Muy Alto
    - Alto
    - Medio
    - Ninguno

13. **¿Estaría usted de acuerdo con utilizar una metodología que le ayude a superar los inconvenientes que se presentan al realizar proyectos de desarrollo de aplicaciones web en la EEASA?**

    - Si
    - No

2.2 Métodos

2.2.1 Modalidad de la investigación

La presente investigación se contextualizó en la modalidad de investigación de campo, bibliográfica-documental.

- Investigación de campo

  La investigación fue de campo debido a que el investigador analiza la problemática en el lugar donde se produce, que es en la EEASA, y se obtendrá los datos necesarios mediante entrevistas en las cuales se establecen causas y consecuencias del problema, y también datos que brinden posibles soluciones al problema que se estudia.

- Investigación bibliográfica-documental

  La investigación es bibliográfica-documental debido a que se fundamenta en la recolección de datos a través de artículos, revistas y tesis, las cuales tengan un enfoque similar al problema y se utilizarán para la construcción de la fundamentación teórica.

2.2.2 Población y muestra

La encuesta que se va a aplicar para la obtención de información y posteriormente realizar el análisis de requisitos, será para el personal que se encuentra en la sección de informática de cada departamento, y que se encuentran a cargo del desarrollo de proyectos de software, que son considerados como la población con la que se va a trabajar como se muestra en la Tabla 4.

Tabla 4. Población

Población	Número	Porcentaje
Director del departamento de desarrollo de software.	1	8.34%
Personal encargado del desarrollo de proyectos de software.	11	91.66%
Total	12	100%

La población en la cual se centrará el trabajo investigativo no sobrepasa de cien elementos, en consecuencia, no se realizarán muestras representativas y se trabajará con la totalidad de la población.

2.2.3 Recolección de información

Los siguientes resultados de la ficha de observación, son el resultado de haberla aplicado en 3 diferentes departamentos; Departamento de Recursos Industriales (DRI), Distribución y Comercial donde se encuentra el personal informático.

Tabla 5. Ficha de Observación del Departamento DRI

Indicadores	**Siempre**	**Casi Siempre**	**A veces**	**Nunca**	**Observaciones**
Se realiza una correcta recolección y análisis de requerimientos.		**X**			
Se obtiene un modelamiento y diseño de base de datos adecuado.		**X**			
Se tiene un objetivo claro de lo que se quiere lograr con la aplicación web.			**X**		Al no realizar una correcta recolección de requerimientos, no se puede establecer un objetivo claro para la aplicación web.
Se elabora un efectivo desarrollo de la aplicación web (codificación) en el tiempo indicado.				**X**	Si no se realiza la codificación en el tiempo indicado, eso quiere decir que no se han establecido tareas, para realizarlas en un tiempo específico.
Realiza entregas constantes sobre el avance del proyecto durante el desarrollo del mismo.			**X**		Al no tener un tiempo específico para el desarrollo de tareas, entonces no se va a realizar entregas constantes del mismo para su revisión.
Durante las revisiones del desarrollo del proyecto se realiza una retroalimentación, para verificar que el proyecto continúa enfocado a los requerimientos y al objetivo principal.		**X**			
Se realiza cambios en los requerimientos iniciales del proyecto durante el desarrollo de la aplicación web (codificación, pruebas).		**X**			
Conclusiones: Al no tener una forma organizada de documentar el proceso del desarrollo de un proyecto, se tiene inconvenientes en la realización de tareas y en el tiempo que se toma para realizar las mismas.					

Tabla 6. Ficha de Observación del Departamento de Distribución

Indicadores	**Siempre**	**Casi Siempre**	**A veces**	**Nunca**	**Observaciones**
Se realiza una correcta recolección y análisis de requerimientos.		**X**			
Se obtiene un modelamiento y diseño de base de datos adecuado.			**X**		Al no tener claro los requerimientos iniciales de un proyecto, el modelamiento de la base de datos, podría llegar a ser deficiente.
Se tiene un objetivo claro de lo que se quiere lograr con la aplicación web.	**X**				
Se elabora un efectivo desarrollo de la aplicación web (codificación) en el tiempo indicado.		**X**			
Realiza entregas constantes sobre el avance del proyecto durante el desarrollo del mismo.		**X**			
Durante las revisiones del desarrollo del proyecto se realiza una retroalimentación, para verificar que el proyecto continúa enfocado a los requerimientos y al objetivo principal.		**X**			
Se realiza cambios en los requerimientos iniciales del proyecto durante el desarrollo de la aplicación web (codificación, pruebas).		**X**			
Conclusiones: Al no realizar una recolección y análisis de requerimientos adecuada, de obtiene un diseño de BD ineficiente y al no ser detectado a tiempo, puede causar retrasos en el desarrollo de un proyecto.					

Tabla 7. Ficha de Observación del Departamento Comercial

Indicadores	**Siempre**	**Casi Siempre**	**A veces**	**Nunca**	**Observaciones**
Se realiza una correcta recolección y análisis de requerimientos.		**X**			
Se obtiene un modelamiento y diseño de base de datos adecuado.			**X**		
Se tiene un objetivo claro de lo que se quiere lograr con la aplicación web.	**X**				
Se elabora un efectivo desarrollo de la aplicación web (codificación) en el tiempo indicado.		**X**			
Realiza entregas constantes sobre el avance del proyecto durante el desarrollo del mismo.		**X**			
Durante las revisiones del desarrollo del proyecto se realiza una retroalimentación, para verificar que el proyecto continúa enfocado a los requerimientos y al objetivo principal.		**X**			
Se realiza cambios en los requerimientos iniciales del proyecto durante el desarrollo de la aplicación web (codificación, pruebas).	**X**				Al iniciar un proyecto y no establecer un objetivo claro de lo que se va a obtener, entonces los requerimientos iniciales no serán los correctos y durante el desarrollo del proyecto se tendrá que volver para corregirlos.
Conclusiones: Es importante realizar una retroalimentación para verificar que el desarrollo del proyecto vaya de acuerdo al objetivo principal y a los requerimientos, pero si es necesario regresar a realizar cambios en los requerimientos iniciales del proyecto, entonces se tiene una estructuración de proyecto deficiente y esto provoca pérdida de tiempo o que el proyecto quede inconcluso.					

Validación del instrumento

Para la validación de la encuesta que se realizó al personal informático y ver la fidelidad de la misma se utiliza el coeficiente de Alfa de Cronbach.

$$\propto = \frac{k}{k-1}\ (1 - \frac{\sum S_i^2}{S_t^2})$$

Ecuación 1. Alfa de Cronbach

Donde:

K = Número de ítems del instrumento

$\sum \boldsymbol{S_i^2}$ = Sumatoria de la varianza de los ítems

$\boldsymbol{S_t^2}$ = Varianza total del instrumento

Para la investigación:

K = 11

$\sum \boldsymbol{S_i^2}$ = 8,55

$\boldsymbol{S_t^2}$ = 32,35

$$\propto = \frac{11}{11-1}\left(1 - \frac{8{,}55}{32{,}35}\right)$$

$$\propto = 0{,}81$$

Ecuación 2. Resultado del Coeficiente de Alfa de Cronbach

Tabla 8. Encuesta Personal Informático Alfa de Cronbach

Pregunta	**Varianza**	**Varianza del instrumento**
1	0,25	32,35
2	0,24	
3	0,25	
4	0,69	
5	0,42	
6	0,22	
7	0,22	
8	0,24	
9	0,14	
10	0,25	
11	0,00	
Total	2,92	
K	11	
∝:	0,81	

Los siguientes resultados que se muestran forman parte de la encuesta realizada al personal encargado del desarrollo de proyectos de software.

Pregunta 1: ¿Utiliza usted una metodología de desarrollo de software, en el diseño y desarrollo de soluciones informáticas de tipo web, en la EEASA?

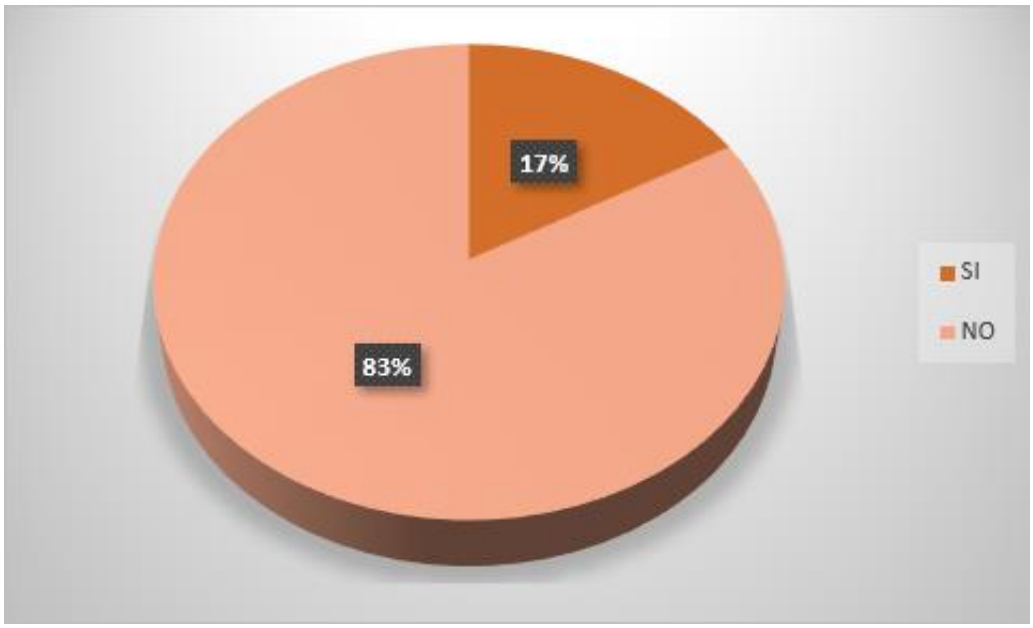


Figura 5. Uso de una Metodología

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 5, el 83% de los encuestados no utilizan una metodología para el desarrollo de proyectos de tipo web, y el 17% que sí, se deduce que no se utiliza una metodología que se ajuste a los proyectos que se realizan y de esa forma no se puede estandarizar el desarrollo de proyectos.

Pregunta 2: Al iniciar el desarrollo de una aplicación web, ¿Para usted cuál de las siguientes actividades tiene mayor importancia?

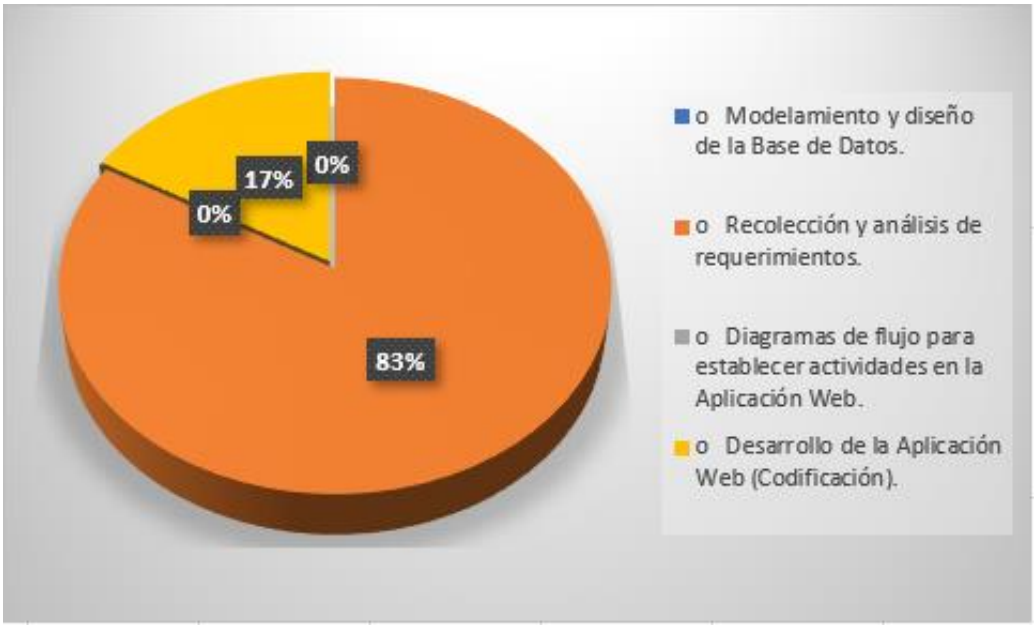


Figura 6. Importancia de actividades en el desarrollo de software

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 6, el 83% de los encuestados señalaron que la actividad que tiene mayor importancia es recolección y análisis de requerimientos, mientras que el 17% indicó que el desarrollo de la aplicación (Codificación) tiene mayor importancia, descartando así el diseño de la base de datos y la realización de diagramas de flujo como actividades de importancia, se deduce que no existe un orden de actividades estandarizado el cual seguir para el desarrollo de un proyecto y la documentación del mismo.

Pregunta 3: ¿Cuál es el mayor problema que usted ha tenido al momento de realizar proyectos de desarrollo de aplicaciones web?

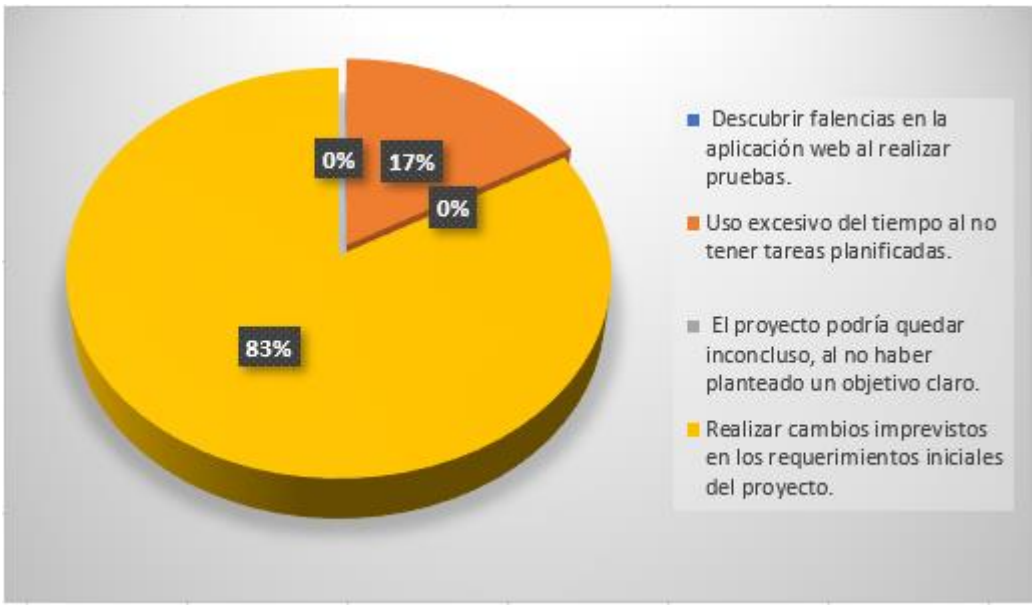


Figura 7. Problemas que se presentan en el desarrollo de un proyecto

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 7, el 83% de los encuestados señalaron que el mayor inconveniente que se presenta al realizar el desarrollo de una aplicación web es, realizar cambios imprevistos en los requerimientos iniciales del proyecto, el 17% señaló que se hace uso excesivo de tiempo al no tener tareas planificadas, sin que se haya señalado las opciones de, descubrir falencias al realizar pruebas o que el proyecto podría quedar inconcluso como mayor inconveniente, podrían llegar a ser consecuencias de los problemas que si tienen mayor impacto al no tener una metodología documentada con la cual guiarse y establecer tareas con un objetivo alcanzable, y un tiempo estándar para cada una de ellas.

Pregunta 4: ¿Se realiza una correcta recolección de requerimientos, enfocada al objetivo principal de los proyectos de desarrollo de aplicaciones web?

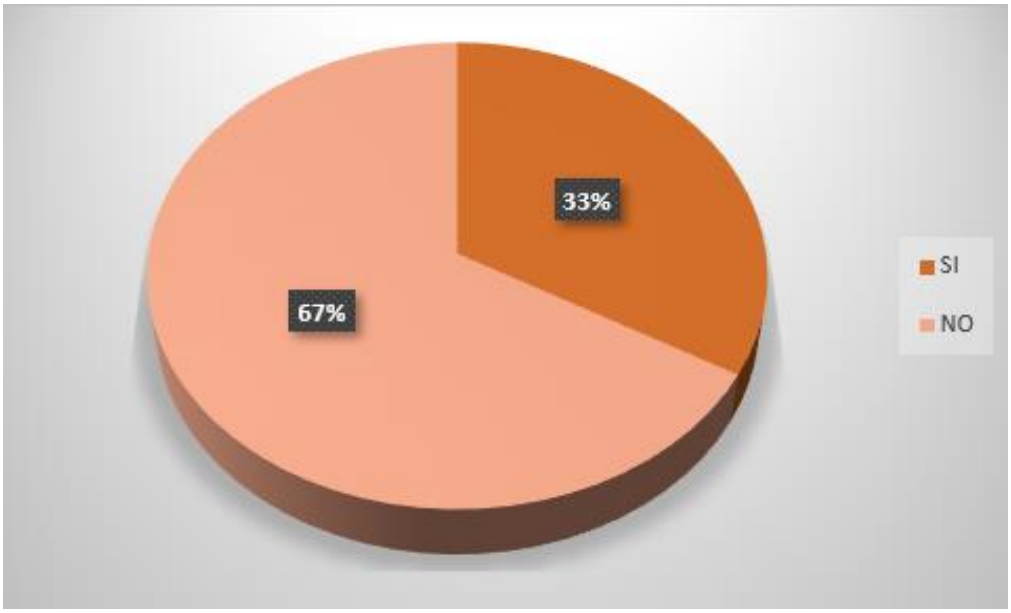


Figura 8. Correcta recolección de requerimientos

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 8, el 67% de los encuestados señalaron que no se realiza una correcta recolección de requerimientos, y el 33% de los informáticos señala que si, entonces se deduce que, al no tener una metodología estandarizada, no se puede cumplir con una recolección y análisis de requerimientos eficiente para posteriormente dar inició al desarrollo del proyecto.

Pregunta 5: ¿Para usted qué grado de importancia tiene el análisis y la documentación de los proyectos de desarrollo de aplicaciones web?

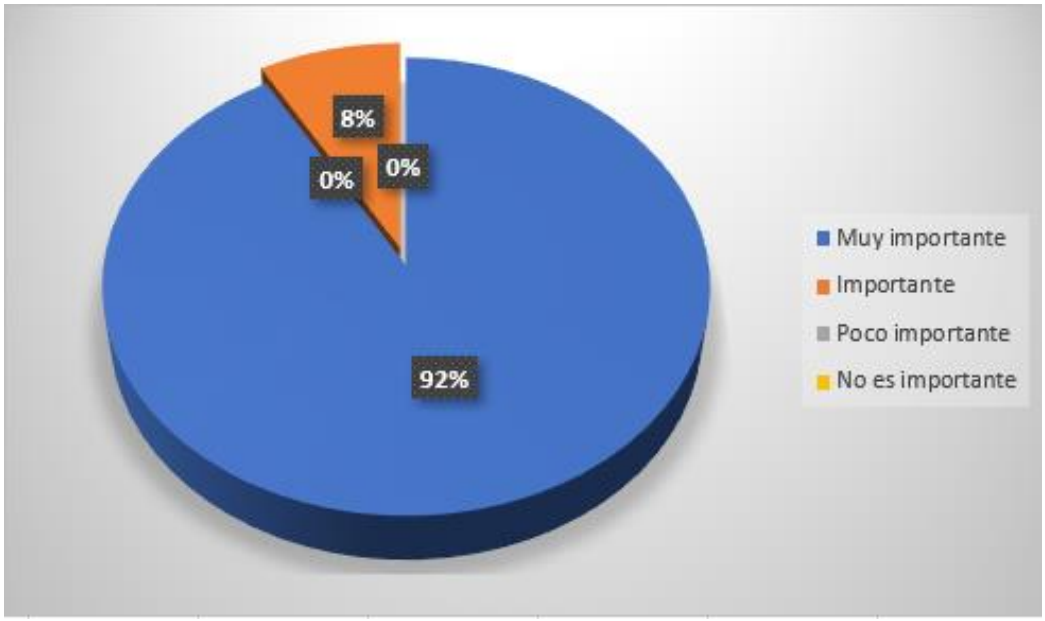


Figura 9. Importancia del análisis y documentación en desarrollo de proyecto.

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 9, el 92% de los encuestados señalaron que es muy importante el análisis y documentación en el desarrollo de aplicaciones web, mientras que el 8% señaló que es importante, se deduce que para el personal encargado del desarrollo de proyectos de software es necesario realizar el análisis y documentación dentro de los proyectos que se realizan.

Pregunta 6: ¿Considera usted que se debe realizar revisiones constantes durante la fase de codificación, en los proyectos de desarrollo de aplicaciones web?

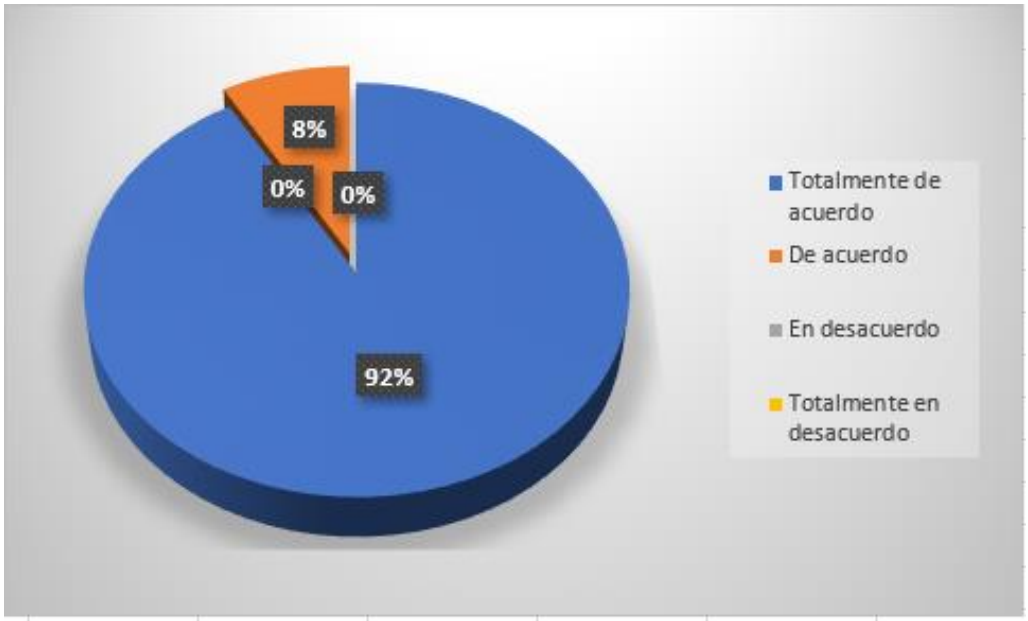


Figura 10. Revisiones constantes

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 10, el 92% de los encuestados señalaron en que están totalmente de acuerdo en realizar revisiones constantes durante la fase de codificación en el desarrollo de una aplicación web, mientras que el 8% señaló que están de acuerdo, se deduce que es necesario realizar revisiones constantes, por lo cual se debe disponer de un cronograma detallado con tiempos establecidos para revisiones.

Pregunta 7: ¿Durante la etapa de desarrollo de aplicaciones web se realizan entregas según el cronograma establecido para su revisión y/o pruebas?

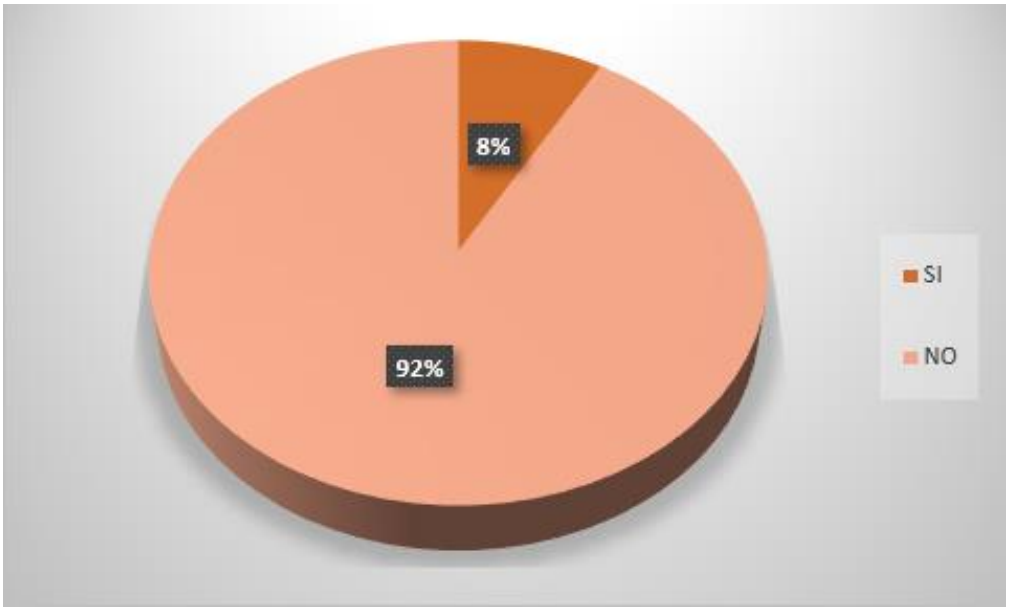


Figura 11. Entregas en el tiempo establecido

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 11, el 92% de los encuestados señalaron en que no se realizan entregas según el cronograma establecido durante el desarrollo de aplicaciones web, y el 8% señaló que sí, se deduce que a pesar de no disponer de una metodología que establezca un cronograma claro, tratan de realizar entregas según un tiempo estimado, pero al disponer de un cronograma detallado no pueden realizar entregas sin retrasos porque se plantean tareas considerablemente largas y que no están bien organizadas.

Pregunta 8: ¿Cuándo considera usted óptimo volver a la fase de codificación en el caso de detectar deficiencias en la aplicación web durante la fase de pruebas?

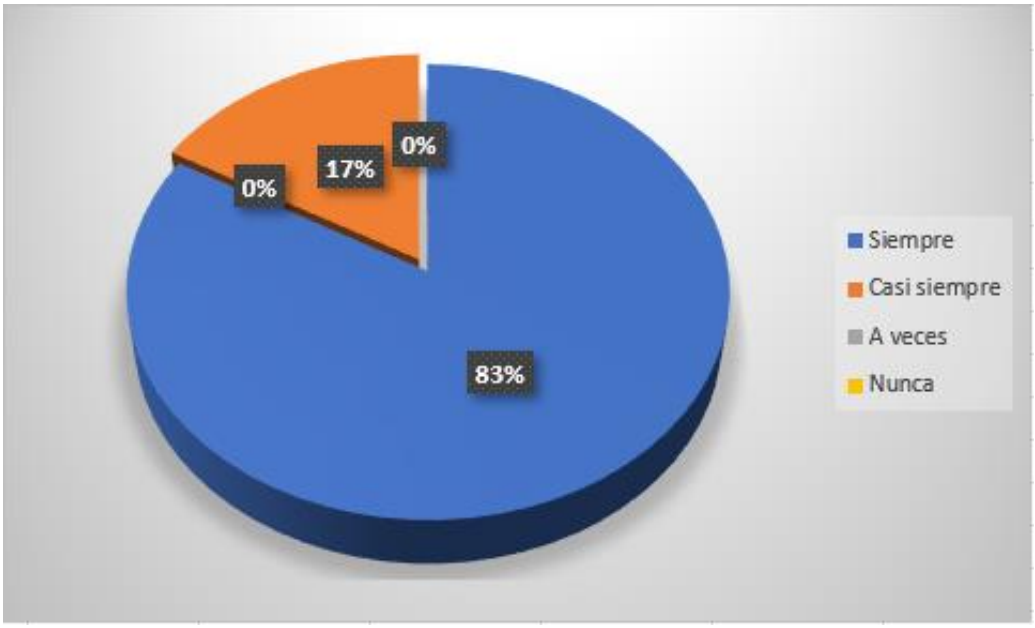


Figura 12. Volver a fase de codificación al detectar deficiencias

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 12, el 83% de los encuestados señalaron que se considera, siempre volver a la fase de codificación al momento de detectar deficiencias en la aplicación web durante la fase de pruebas, en cambio el 17% considera que casi siempre, se deduce es necesario regresar a la fase de codificación en el caso de detectar deficiencias en la aplicación web durante las pruebas, por lo tanto se requiere de un cronograma dentro de una metodología que les ayude con una organización eficiente para la realización de las actividades.

Pregunta 9: ¿Está usted de acuerdo en que las correcciones de las deficiencias detectadas en la aplicación web, se las debe realizar durante las revisiones en la etapa de codificación?

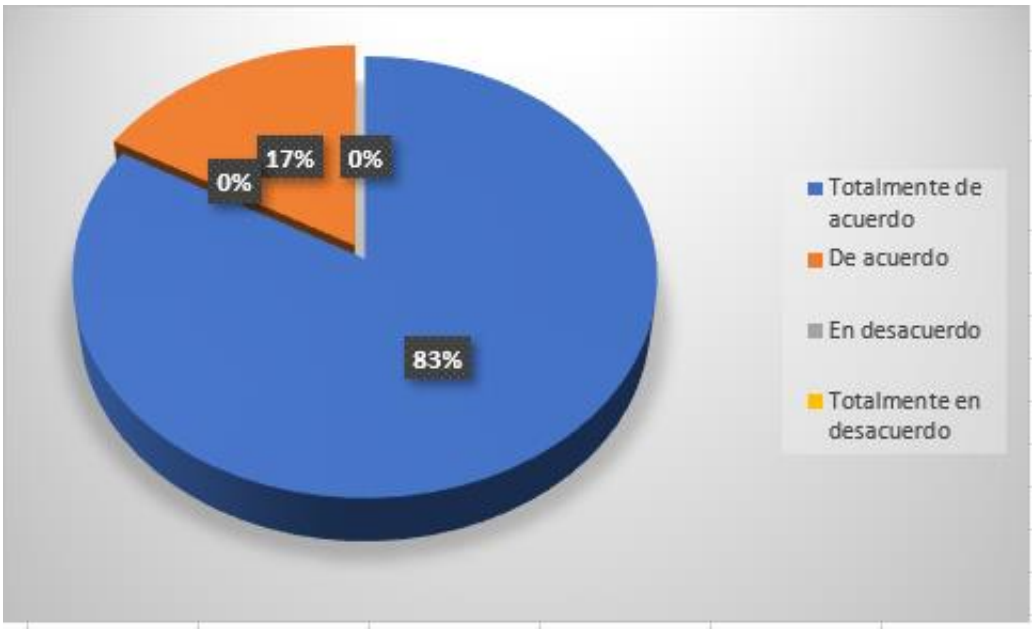


Figura 13. Correcciones durante la etapa de codificación

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 13, el 83% de los encuestados señalaron en que están totalmente de acuerdo en que las correcciones de las deficiencias detectadas en la aplicación web, se las debe realizar durante las revisiones de la etapa de codificación, mientras que el 17% están de acuerdo, se deduce que es de importancia las revisiones, para posteriormente implementar las correcciones indicadas, durante el desarrollo de una aplicación web (Codificación).

Pregunta 10: ¿Considera usted que es importante realizar una evaluación de calidad de software, durante la fase de pruebas?

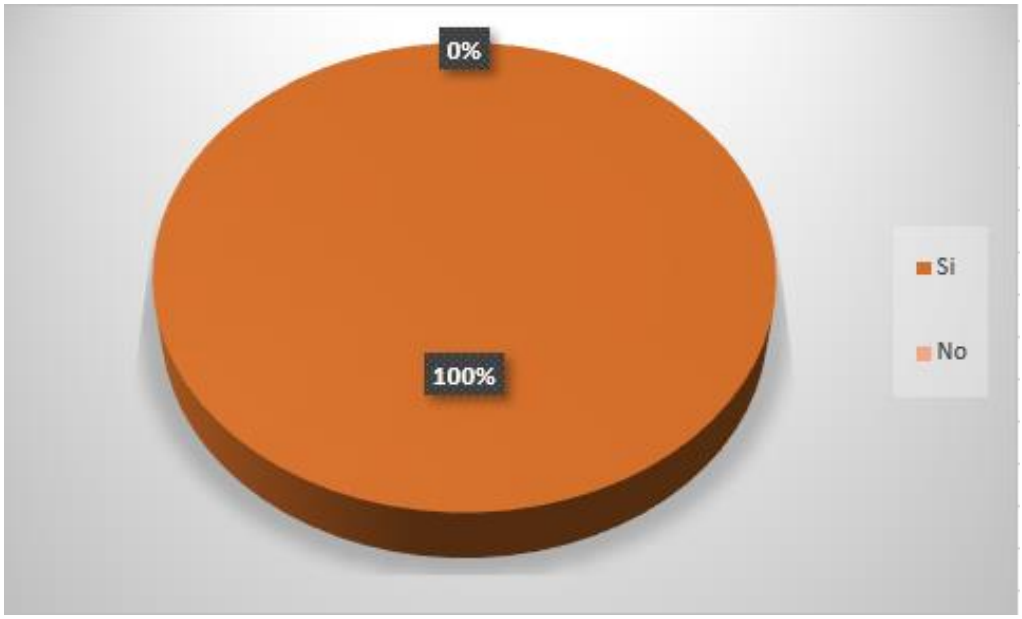


Figura 14. Realizar evaluación de calidad de software

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 14, el 100% de los encuestados señalaron en que, si se debe hacer una evaluación de calidad de software, se puede deducir que, es necesario realizar la evaluación de calidad para que el resultado sean aplicaciones de utilidad para el usuario y de esta forma pueda realizar sus tareas sin inconvenientes.

Pregunta 11: ¿Está usted familiarizado con las Normas ISO/IEC 25000?

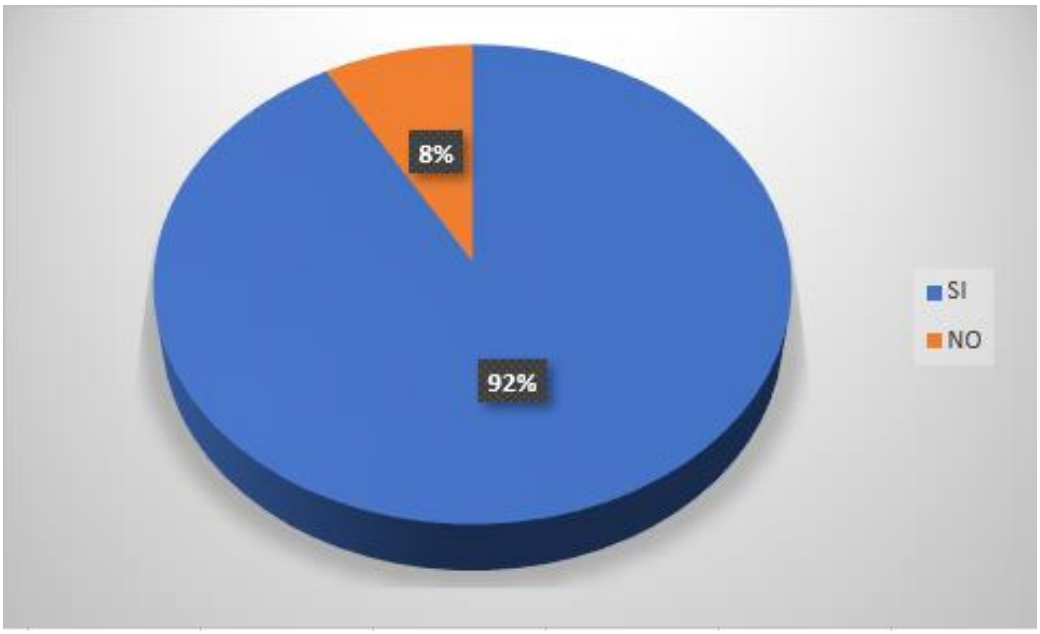


Figura 15. Normas ISO/IEC 25000

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 15, el 92% de los encuestados están familiarizados con las normas ISO/IEC 25000, mientras que el 8% no están familiarizados, por lo tanto, se deduce que al ser la mayoría están familiarizados con las normas y se puede realizar una evaluación de las aplicaciones web que se desarrollan en la EEASA.

Pregunta 12: ¿En qué nivel considera usted que se encuentra su conocimiento sobre las normas ISO/IEC 25000?

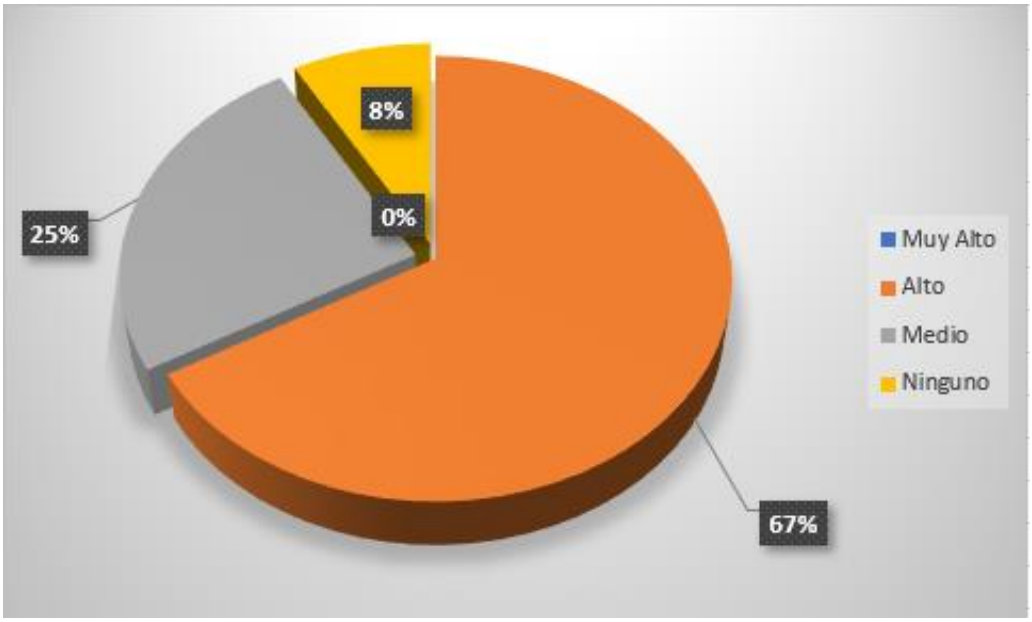


Figura 16. Conocimiento sobre las normas ISO/IEC 25000

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 16, el 67% de los encuestados tienen un conocimiento alto sobre las normas ISO/IEC 25000, el 25% tienen un conocimiento medio, y el 8% no están familiarizados con sobre las normas de calidad, entonces se deduce que se puede realizar una evaluación de calidad de software efectiva, ya que la mayoría posee conocimiento sobre las normas de calidad.

Pregunta 13: ¿Estaría usted de acuerdo con utilizar una metodología que le ayude a superar los inconvenientes que se presentan al realizar proyectos de desarrollo de aplicaciones web en la EEASA?

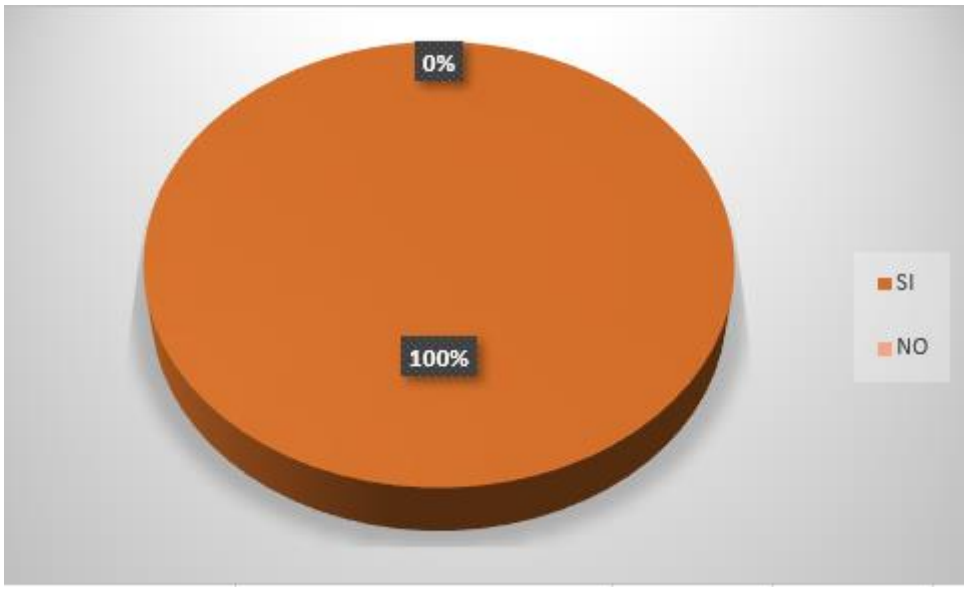


Figura 17. Utilizar una metodología para superar inconvenientes

Fuente: Encuesta aplicada

Análisis e interpretación de resultados

Según los resultados que se muestran en la Figura 17, el 100% de los encuestados están de acuerdo con utilizar una metodología que les ayude a superar los inconvenientes que se presentan al realizar proyectos de desarrollo de aplicaciones web.

2.2.4 Procesamiento y análisis de datos

De acuerdo con los materiales de recolección de información aplicadas al personal informático de la EEASA se demostró lo siguiente:

- Al iniciar un proyecto de desarrollo de aplicaciones web, y no disponer de una metodología no se puede establecer actividades de forma organizada, por lo tanto, dificulta realizar de forma efectiva una recolección y análisis de requerimientos, y al continuar con el proyecto se pueden presentar inconvenientes en el diseño y modelamiento de la base de datos o durante el desarrollo de la aplicación (Codificación), como se muestra en el diagrama de procesos del Anexo A.

- La forma en la que se desarrolla actualmente las aplicaciones web no es eficiente, ya que toman mucho más tiempo del que se tiene planificado o en algunos casos quedan inconclusos por no satisfacer los requerimientos iniciales ya que casi siempre están sujetos a cambios, esto se lo puede observar en el Anexo A.

- No se dispone de una metodología que se pueda utilizar de apoyo para establecer actividades de forma ordenada y documentación del proyecto.

- Se tiene inconvenientes para realizar en la entrega del proyecto en el tiempo establecido, ya que se presentan imprevistos con los desarrolladores o con los materiales que se utilizan y no se dispone de un cronograma que contemple los posibles problemas y se tenga un tiempo previsto para solucionarlos.

- El producto que en este caso es la aplicación web no resulta de mucha utilidad para el usuario, ya que, al no aplicar las normas para la evaluación de calidad de software, no se puede conocer si la aplicación es eficiente y si es de ayuda para las actividades que debe realizar el usuario final.

- El personal encargado del desarrollo de proyectos de software está familiarizado con las normas ISO/IEC 25000, por lo que se puede agregar la evaluación de calidad del software como una actividad documentada dentro de la metodología híbrida.

- Al haber evidenciado los inconvenientes que presentan al desarrollar una aplicación web, todo el personal encargado del desarrollo de proyectos de software, están de acuerdo con utilizar una metodología que estandarice el desarrollo de aplicaciones web, en la EEASA.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados

3.1.1 Metodologías de desarrollo de aplicaciones

Al momento de realizar el desarrollo de software, existen algunas metodologías de las cuales se puede disponer para realizar las tareas de desarrollo de forma organizada, para esto se debe analizar las metodologías tradicionales y ágiles.

a. Metodologías tradicionales

Las metodologías tradicionales a raíz de que se crearon para dar solución a la crisis del software, su enfoque principal es el cumplimiento de una serie de actividades para marcar el inicio y fin de un proyecto de desarrollo de software, esto quiere decir que es un procedimiento lineal en el cual se cumple con el análisis de requerimientos, desarrollo del proyecto, pruebas e implementación del sistema, cada una de estas tareas se suelen desarrollar una vez, como se muestra en la Figura 18.

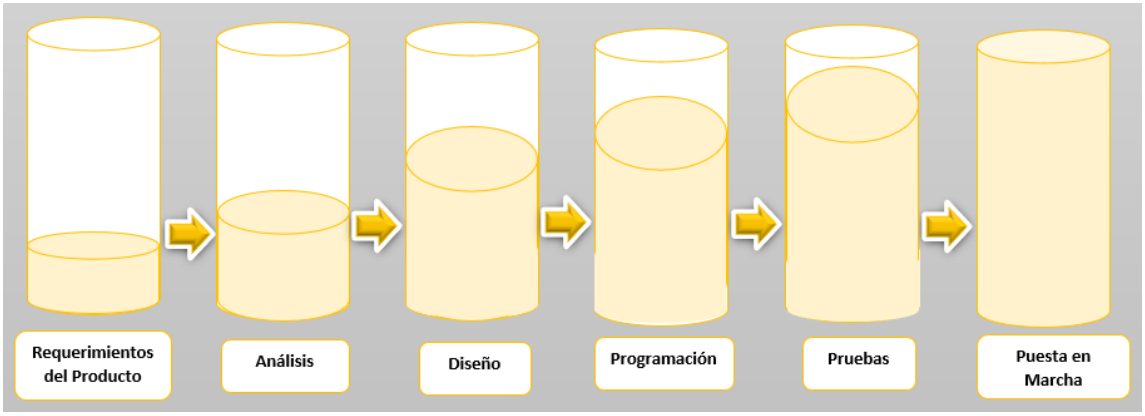


Figura 18. Enfoque de las Metodologías Tradicionales

Para integrar en una metodología híbrida las técnicas de trabajo o estrategias de las diferentes metodologías tradicionales, se debe realizar una investigación y análisis de las mismas, como se muestra en la Tabla 9.

Tabla 9. Análisis de Metodologías Tradicionales

Elaborado por: Investigador

Metodología	Enfoque	Gestión de trabajo	Flexibilidad	Organización	Técnicas de trabajo
Cascada [14]	Es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto y al contrario que en los modelos interactivos, cada una de las fases se ejecutan tan solo una vez.	El desarrollador deberá adaptarse a las 7 fases que presenta el modelo en cascada, que son: requisitos del sistema, requisitos de software, análisis, diseño, implementación, prueba y servicio.	A pesar de que el modelo se lo puede dividir en 5 fases, que son: análisis, diseño, implementación, verificación, y mantenimiento. El modelo es muy poco flexible ya que cada una de estas fases se basa en la anterior y es de una sola iteración.	La persona o el equipo encargado del desarrollo de software deberá planificar y prever un conjunto de actividades con el cual se pueda cumplir cada una de las fases que componen el modelo.	La técnica que propone el modelo en cascada es tomar los objetivos de mayor alcance o de alto nivel y dividirlos en subobjetivos independientes a los cuales se los puede cuantificar de tal forma que los interesados puedan asignarle pesos y de esa forma trabajar con los subobjetivos empresariales propuestos.

Espiral [15]	El modelo en espiral describe el ciclo de vida de un software por medio de espirales que se repiten hasta que se pueda entregar el producto terminado, la clave de este modelo es la minimización de los riesgos en el desarrollo de software.	El modelo en espiral se caracteriza por trabajar con los siguientes ciclos: objetivo y alternativa, análisis y evaluación de riesgos, desarrollo, pruebas y planificación del siguiente ciclo.	El modelo en espiral al tener un enfoque diferente en el que se realiza un análisis de riesgos y se trabaja con algunas iteraciones, entonces es más flexible y permite al desarrollador realizar cambios en una etapa posterior.	El desarrollador al tener que realizar un análisis de riesgos, puede llegar a tener una mejor planificación de las actividades y mantener mayor comunicación con el cliente durante el desarrollo del proyecto.	El modelo de ciclo de vida espiral consta de 4 fases: planificación, análisis de riesgos, implementación y evaluación, se debe pasar repetidamente por cada una de las fases de esta forma se obtienen iteraciones y cada iteración debe mostrar el avance del proyecto.
Proceso Unificado Racional [16]	Proporciona un acercamiento disciplinado a la asignación de tareas	Este modelo en esencial ayuda a la mejora de la productividad del	El modelo RUP, no ofrece un entorno de trabajo muy flexible ya que cada uno de los	Cada miembro del equipo organizará sus actividades de acuerdo al cronograma específico propuesto para	Dentro de las actividades para cumplir con el proceso de trabajo que se

	y responsabilidades en una organización de desarrollo.	equipo desarrollador, reduciendo la complejidad de los procesos y el diseño de los mismos, ya que permite que cada miembro tenga tareas asignadas específicas, para cumplir con las cinco siguientes fases: fase de inicio, fase de elaboración, fase de construcción y fase de transición.	miembros del equipo desarrollador tiene tareas específicas, las cuales se deben cumplir en tiempos asignados y predecibles para cumplir con cada fase dentro del desarrollo.	cumplir con las fases de trabajo.	presenta en las fases del modelo RUP, principalmente se encuentra la gestión de requisitos, utilizar arquitecturas basadas en componentes y modelar visualmente el software, para esto se hace uso del UML (Unified Modeling Languaje, en inglés).
Prototipo [17]	La metodología Prototipo permite realizar y materializar diversas ideas de soluciones	Este modelo basado en el ciclo Deming, consiste en un proceso iterativo, que consiste en diseñar,	A pesar de que el modelo de prototipado está enfocado al desarrollo de aplicaciones web, no resulta ser muy flexible	Dentro del equipo de desarrollo, cada uno de los miembros pueden exponer posibles soluciones al problema planteado y de	La metodología de prototipado se centra principalmente en cumplir de la forma más visual y tangible con el

	propuestas en un proyecto, y está relacionado con la mejora continua.	implementar, mediar y ajustar un plan, en el que se aplican cinco fases: definir los requerimientos y variables, definir herramientas de diseño y testeo, diseñar el prototipo y análisis de resultados.	para el desarrollo con un equipo de trabajo porque al igual que otras metodologías tradicionales se debe cumplir con las fases de la metodología de forma lineal, y en la fase final si los resultados no son los esperados se puede desechar el producto o verse en la necesidad de volver a empezar.	esta forma realizar un prototipo adecuado.	diseño que hayan propuesto los interesados, para esto se debe hacer uso de herramientas de prototipado, con el propósito de diseñar y realizar testeo de las aplicaciones que se quieren desarrollar.

b. Metodologías ágiles

Las metodologías ágiles se presentan a partir de la necesidad de entregar proyectos de formas más eficientes, y esto se logra a que dentro de su estructura se tienen ciclos de trabajo cortos, con entregables cada 2 o 4 semanas dependiendo del tamaño del proyecto, con revisiones y retroalimentación continúas, esto quiere decir que se cumple con el ciclo de vida de un proyecto de desarrollo de software pero a diferencia de las metodologías tradicionales, en las metodologías ágiles cada bloque de trabajo se lo realiza con iteraciones y no una única vez, como se muestra en la Figura 19, esto también permite que se acoplen a proyectos de desarrollo pequeños, medianos y grandes.

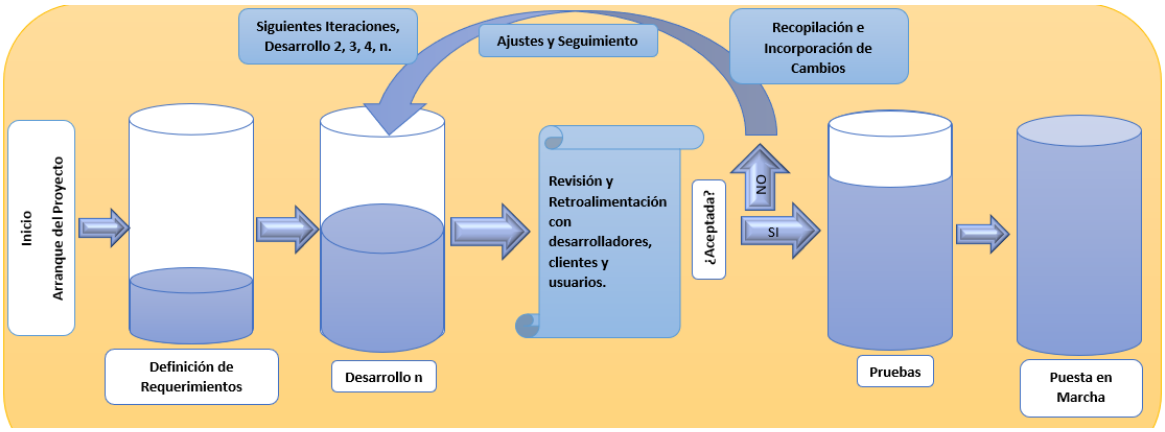


Figura 19. Enfoque de las Metodologías Ágiles

Al disponer de varias metodologías ágiles, las cuales pueden ser aplicadas para el desarrollo web, al igual que las metodologías tradicionales se debe realizar una investigación y análisis de las técnicas de trabajo de cada una de ellas, como se muestra en la Tabla 10.

Tabla 10. Análisis de Metodologías Ágiles

Elaborado por: Investigador

Metodología	Enfoque	Gestión de trabajo	Flexibilidad	Organización	Técnicas de Trabajo
Programación Extrema (XP o Extreme Programming, en inglés) [14]	El objetivo de este modelo es el desarrollo y gestión de proyectos con eficacia, flexibilidad y control, es una metodología basada en la comunicación y retroalimentación.	En la gestión de trabajo se caracteriza porque se lo realiza mediante un proceso iterativo y define las prácticas y roles del equipo, teniendo en cuenta 4 variables para cualquier proyecto, estas son: coste, tiempo, calidad y alcance.	Para este modelo se trabaja con una planificación flexible y abierta, en el cual se debe mantener una interacción continua con el cliente y los desarrolladores esto permite que el desarrollo sea iterativo e incremental aportando pequeñas mejoras.	Existe una lista predeterminada de funciones para cada miembro del equipo, pero se trabaja con flexibilidad para lograr un ritmo de trabajo sostenido y efectivo.	La metodología XP consta de 5 fases: planificación, diseño, codificación, pruebas y lanzamiento, en cada una de las fases esta metodología se basa en una estrecha interacción con los clientes, pruebas constantes y ciclos de desarrollo cortos, para esto se debe tener un ritmo de trabajo sostenible por lo que todos los días deben

					tener una carga equitativa de trabajo, así la productividad y efectividad del modelo se cumplen. Una forma de tener claro el cronograma de actividades es utilizando tableros Kanban.
Scrum [15]	La metodología Scrum es un proceso para llevar a cabo un conjunto de tareas de forma regular con el objetivo principal de trabajar de manera	Con este modelo se pretende resolver el inconveniente de que el proyecto no se está entregando a tiempo a un determinado cliente, porque permite identificar y solucionar deficiencias de forma	A pesar de que Scrum demanda disciplina para cumplir con los objetivos de manera colaborativa, también ofrece flexibilidad para la resolución de problemas,	Al tener un cronograma de tareas organizadas, se puede trabajar de forma colaborativa no solo con el equipo de desarrollo sino también con el cliente, todo esto por el hecho de que se	Para el cumplimiento de las fases que componen el modelo Scrum, se debe trabajar de forma colaborativa y fomentar el trabajo en equipo y es necesario realizar entregas regulares y parciales del producto

	colaborativa, para fomentar el trabajo en equipo.	sistemática. Esto se puede cumplir al ejecutar las siguientes cuatro fases: planificación, ejecución de sprints, control y monitorización de las actividades, revisión y adaptación.	adaptación y experimentación.	trabaja con sprints y daily scrum.	final, por lo que el modelo es indicado para proyectos con requisitos cambiantes y la flexibilidad es indispensable para el equipo de desarrollo.
RAD [16]	RAD se caracteriza por ser un modelo en el que los módulos de aplicación individuales se desarrollan en paralelo y se ensamblan en un	Ya que existe una mayor comunicación con el usuario final, para un intercambio de ideas y revisiones de los avances para cumplir con las actividades establecidas por calendario, se tratan las siguientes fases: modelado de gestión, modelado de datos,	Esta metodología se la puede considerar como flexible ya que se le puede añadir "Timeboxing", esto quiere decir que, las funciones secundarias pueden llegar a ser eliminadas para cumplir con un cronograma, pero esto puede conducir a una	Dentro de la organización se puede hacer uso de un calendario grupal en el que cada miembro del equipo de desarrollo tiene un mismo tiempo determinado para realizar la tarea o tareas que se le ha asignado para luego	RAD tiene una forma de desarrollo de proyectos iterativo, incremental y de obtener feedback de manera constante lo que asegura que el producto evolucione en armonía con las necesidades cambiantes y las

	producto terminado.	modelado de procesos, generación de la aplicación, y pruebas de entrega.	desventaja que es el de funciones reducidas dentro de la aplicación.	realizar las revisiones de los avances.	expectativas del usuario.
Crystal [19], [20]	El enfoque de la metodología Crystal es que está diseñada para ser liviana y flexible, ya que en lugar de tratar de hacer que un proyecto se ajuste a un conjunto de reglas, Crystal adapta el método al proyecto.	La gestión de trabajo con esta metodología se simplifica ya que se adapta al tipo de proyecto, se trabaja con ciclos de entrega diferentes, estos pueden ser semanales o mensuales, y la comunicación de progreso y revisiones puede hacerse con reuniones diarias o con herramientas digitales para la gestión de tareas y	Este modelo se lo puede considerar flexible, ya que se adapta al tipo de proyecto y al número de persona con las que se va a trabajar en el equipo de desarrollo.	Para la organización del equipo de desarrollo se puede adaptar la metodología Kanban con un tablero digital para el seguimiento de tareas o la combinación de tableros físicos y digitales para la gestión de tareas.	El modelo crystal se basa en las siguientes fases: puesta en escena, revisiones, monitoreo, paralelismo y flujo, para el cumplimiento de estas cuatro fases el desarrollo debe ser iterativo e incremental, con la participación activa del usuario, por lo que cada incremento tiene iteraciones que incluye actividades que deben ser monitoreadas

		comunicación entre equipos.			a partir de las diferentes entregas.
DSDM [21], [22]	El enfoque principal de este modelo es la alineación del proyecto con los objetivos estratégico de la organización, con eso se puede medir la factibilidad y que existan fundamentos sólidos antes de que se inicie el proyecto.	DSDM es una metodología que permite permanecer flexible y cumplir con los cambiantes requerimientos de negocios, para esto se tiene seis fases: pre-proyecto, viabilidad, cimientos, desarrollo evolutivo, despliegue y post-proyecto.	Esta metodología brinda flexibilidad ya que consta de los siguientes principios: concentrarse en la necesidad del negocio, trabajo colaborativo en equipos, construir gradualmente desde cimientos firmes, se desarrolla de forma iterativa y la comunicación es de forma continua.	Para la organización del equipo de desarrollo, se debe tener en cuenta que el trabajo es incremental, priorizando los requisitos del proyecto, esto quiere decir que los proyectos a largo plazo se entregan puntualmente mediante la entrega de cada incremento.	La principal técnica en la que se basa el modelo DSDM es que antes de dediquen recursos significativos al proyecto, se debe tener una sólida comprensión de los requisitos del proyecto y la solución propuesta, después de cada Timebox las prioridades y la viabilidad del proyecto son reevaluados, esto fomenta la comunicación continua y clara durante las

					reuniones constantes para la retroalimentación.
Design Sprint [24]	Es una metodología de duración breve, pero intensa, que se usa para dar respuesta a problemas de negocios, sobre todo al inicio de un nuevo producto o servicio, y parala retroalimentación de los usuarios.	La característica de este modelo es que consiste en un proceso estricto, con un equipo pequeño para cubrir todo el trabajo, el cual debe estar enfocado en el producto que será utilizado por los usuarios y obtener retroalimentación. Para esto es necesario integrar las siguientes fases: comprensión, idear, decidir, realizar el prototipo y testear.	Esta metodología se puede considerar que es muy poco flexible ya que dentro de su gestión de trabajo existe un proceso estricto de actividades a cumplir en un corto plazo de tiempo, para que al final se presente un producto al usuario.	Para la organización del equipo de desarrollo, se necesita la comprensión de que serán sesiones de trabajo intensas, y deben estar comprometidos a trabajar en un horario convenido de seis a ocho horas.	La metodología design sprint propone un ritmo de trabajo intenso para el cumplimiento de cada una de las fases, para esto es necesario disponer de un equipo integrado por personas con un elevado conocimiento de desarrollo de software, y después asignar tareas específicas a cada uno de los miembros del equipo, las cuales se deben cumplir en un

					tiempo limitado ya que se debe cumplir con un proceso de cinco días y cada uno con una carga de trabajo de seis a ocho horas, y finalizar la validación que es el testeo con usuarios.

c. Características del desarrollo de aplicaciones web

En el desarrollo de las aplicaciones web, existen características las cuales están para ser cumplidas por el equipo de desarrollo durante el ciclo de vida del proyecto, muchas de ellas se encuentran contempladas dentro de las metodologías de desarrollo de software, como se muestra en la Tabla 11, para que sean llevadas a cabo a través de tareas, algunas de ellas consideradas como requerimientos funcionales y no funcionales.

Al iniciar un proyecto de desarrollo de aplicación web, es de importancia realizar un análisis inicial el cual sirva como una guía para la empresa o para el cliente, y se tenga el conocimiento del impacto que podría llegar a tener, para el cumplimiento de esta etapa se realizan los siguientes análisis: análisis de factibilidad y viabilidad, análisis de riesgos y posteriormente recolección y análisis de requerimientos [32], [33].

Para continuar con la siguiente etapa del desarrollo de una aplicación web, una vez hecho el análisis se puede dar inicio al diseño de la arquitectura del software, al encontrarse en esta etapa se debe contemplar las siguientes actividades [33], [34]:

- Diseño y modelamientos de la base de datos
- Diseño de la interfaz y experiencia de usuario
- Diagrama de procesos y asignación de roles.

En las siguientes etapas del desarrollo de una aplicación web, se encuentra la codificación en la cual las siguientes características tienen un papel muy importante ya que son en las que se trabaja para cumplir con los requerimientos funcionales y no funcionales:

- Adecuación funcional
- Desempeño
- Usabilidad y seguridad,

Posteriormente se tiene la etapa de pruebas en las que el cliente o usuario final tendrá interacción con el software y con ello obtener retroalimentación y observaciones para continuar con el desarrollo del proyecto [35].

Tabla 11. Estrategias de Trabajo Aplicada al Desarrollo Web

Desarrollo de Aplicaciones Web		
Características	Metodologías Tradicionales	Metodologías Ágiles
Recolección y análisis de requerimientos.	X	X
Análisis de riesgos.	X	X
Análisis de factibilidad y viabilidad.	X	X
Desarrollo de proyecto (Codificación) iterativos e incremental.		X
Ciclos de desarrollo (Codificación) cortos e interacción constante con los usuarios y clientes.		X
Flexibilidad, para el equipo de desarrollo.		X
Revisiones cada intervalo de tiempo para retroalimentación.		X
Adaptarse a requerimientos cambiantes durante el desarrollo del proyecto.		X
Pruebas e implementación de la aplicación de acuerdo a los objetivos planteados.	X	X
Pocos roles, más genéricos y flexibles		X

La arquitectura se va definiendo y mejorando a lo largo del proyecto		X

3.1.2 Estructura de las Normas ISO/IEC 25000

Las normas ISO/IEC 25000 son una familia de normas las cuales se complementan para realizar una evaluación de calidad de software efectiva, en la cual se evalúa calidad externa, interna y calidad de uso, para completar las tareas que se involucran en el proceso de evaluación el evaluador tendrá que contemplar las actividades que forman parte de cada una de las divisiones de las normas ISO 25000, en las cuales se definen características, sub-caracteríticas, atributos, métricas y el porcentaje de ponderación que tendrá cada una de ellas.

a. División para gestión de la calidad

Para la gestión de la calidad, se establecen requisitos los cuales son parte de las características y sub-características que componen las normas ISO/IEC 25000, como se muestra en la Tabla 12.

Tabla 12. Características y Sub-Características de las Normas ISO/IEC 25000
Fuente: ISO 25000 [28]

Normas ISO/IEC 25000	
Características	**Sub-Características**
Adecuación Funcional	Idoneidad
	Corrección
Fiabilidad	Madurez

	Tolerancia a Fallos
Eficiencia de Desempeño	Comportamiento Temporal
	Utilización de Recursos
Seguridad	Confidencialidad
	Integridad
Usabilidad	Inteligibilidad
	Aprendizaje
	Protección de Errores de Usuario
	Operatividad

b. División para el modelo de calidad

En la división para el modelo de calidad, se establecen características con las cuales se pretende evaluar calidad interna, calidad externa y calidad de uso del producto de software, para esto a cada una de las características se les ha asignado pesos o una ponderación diferente, según su nivel de importancia basándose en los requisitos de la empresa.

La valoración de cada una de las características ha sido asignada por el equipo de desarrollo según el grado de importancia, con el principio de que al final de la evaluación se obtendrá una calificación final sobre 100, en la que todas las características se complementan para obtener dicha calificación, como se muestra en la Tabla 13.

Tabla 13. Porcentaje de Ponderación de las Características a Evaluar

Característica	Porcentaje de Ponderación	
Adecuación funcional	20	
Fiabilidad	25	
Eficiencia de desempeño	20	
Seguridad	25	
Usabilidad	10	
	Total	100

c. División para la medición de la calidad

Al tener el porcentaje de ponderación de cada una de las características que van a ser evaluadas, como se muestra en la Tabla 13, en la división para la medición de la calidad se tiene rangos de calificación, los cuales han sido establecidos por el evaluador y que aplica a los niveles de puntuación y grados de satisfacción propuestos para la norma ISO/IEC 25040, como se muestra en la Tabla 14, para reflejar el resultado de la evaluación de forma cuantificable.

Tabla 14. Rangos de Calificación

Rangos de Calificación	**Resultado de Evaluación**
90 - 100	Excelente

80,00 – 89,99	Muy Bueno
69,01 - 79,99	Aceptable
50 - 69,00	Malo
0 – 49,99	Deficiente

d. División para los requisitos de calidad

Para el cumplimiento de la división para los requisitos de calidad, se debe definir de forma detallada los atributos que se deben tomar en cuenta para realizar una correcta evaluación de calidad, como se muestra en la Tabla 15, en la que cada uno de los atributos completarán un porcentaje de la ponderación de las sub-características que comprenden la calidad interna, calidad externa y calidad de uso.

Tabla 15. Porcentajes de Ponderación en Sub-Características

Característica	**Sub-Características**	**Atributo**	**Porcentaje de Ponderación**
Adecuación funcional	Idoneidad	Cobertura	10
		Completitud funcional	
	Corrección	Precisión y exactitud computacional	10
		Coherencia	

Fiabilidad	Madurez	Presencia de errores	15
		Prevención de fallas contra errores internos	
	Tolerancia a fallos	Continuidad en la operación	10
Eficiencia de desempeño	Comportamiento temporal	Capacidad de responder a las necesidades en un tiempo de retorno aceptable	8
	Utilización de recursos	Capacidad de optimizar el uso de memoria	12
		Capacidad para manejar concurrencia	
		Capacidad para optimizar las comunicaciones	
Seguridad	Confidencialidad	Capacidad de control acceso a información privada sin permisos	15
		Seguridad de contraseña	
	Integridad	Integridad del sistema	10
Usabilidad	Inteligibilidad	Capacidad para proveer entradas y salidas entendibles	3

		Capacidad para brindar claridad al usuario	
	Aprendizaje	Capacidad para ser aprendido fácilmente por el usuario	2
		Documentación adecuada	
	Protección de Errores de usuario	Presencia de entradas válidas	2
	Operatividad	Capacidad para ser operado y recordado por el usuario con facilidad	3
		Capacidad para reducir procedimientos de operación	
		Presencia de valores por defecto	
		Presencia de mensajes claros de usuario	

e. División para la evaluación de calidad

Para complementar a la familia de normas ISO/IEC 25000, se tiene la división para la evaluación de calidad, en la que se muestra a cada una de las características junto con sus atributos y métricas a evaluarse, sirve como una guía de referencia y modelo de evaluación, y cuenta con módulos diferentes, cada módulo se describe en las siguientes tablas; Tabla 16, Tabla 17, Tabla 18, Tabla 19 y Tabla 20.

Tabla 16. Módulo de Adecuación Funcional

Adecuación Funcional			
Sub-Características	Atributo	Métrica	Aplicación
Idoneidad	Cobertura	Porcentaje de requerimientos cumplidos	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
	Completitud funcional	Consistencia de la implementación funcional	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
Corrección	Precisión y exactitud computacional	Prueba de Exactitud	Pruebas del software realizadas por parte del equipo evaluador.
	Coherencia		

Tabla 17. Módulo de Fiabilidad

Fiabilidad			
Sub-Característica	Atributo	Métrica	Aplicación
Madurez	Presencia de errores	Densidad de fallas contra casos de prueba	Pruebas del software realizadas por parte del equipo evaluador.
	Prevención de fallas contra errores internos	Validación de entradas	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
		Prevención de fallas contra errores internos	
Tolerancia a fallos	Continuidad en la operación	Capacidad de continuidad en la operación	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.

Tabla 18. Módulo de Eficiencia de Desempeño

Eficiencia de Desempeño			
Sub-Característica	Atributo	Métrica	Aplicación
Comportamiento temporal	Capacidad de responder a las necesidades en un tiempo de retorno aceptable	Tiempo de respuesta	Pruebas del software realizadas por parte del equipo evaluador.
Utilización de recursos	Capacidad de optimizar el uso de memoria	Prueba de optimización de uso de memoria	Pruebas del software realizadas por parte del equipo evaluador.
	Capacidad para manejar concurrencia	Capacidad de manejo de concurrencia	Pruebas del software realizadas por parte del equipo evaluador.
	Capacidad para optimizar las comunicaciones	Capacidad de optimizar comunicaciones y uso de dispositivos I/O	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.

Tabla 19. Módulo de Seguridad

Seguridad			
Sub-Característica	Atributo	Métrica	Aplicación
Confidencialidad	Capacidad de control acceso a información privada sin permisos	Capacidad de control de acceso a información sin permisos	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
	Seguridad de contraseña	Coherencia de contraseñas entre interfaces	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
Integridad	Integridad del sistema	Pruebas de asignación de roles y privilegios de usuario	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.

Tabla 20. Módulo de Usabilidad

Usabilidad			
Sub-Característica	Atributo	Métrica	Aplicación
Inteligibilidad	Capacidad para proveer entradas y salidas entendibles	Porcentaje de capacidad para proveer entradas y salidas entendibles	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
	Capacidad para brindar claridad al usuario	Porcentaje de capacidad para brindar claridad al usuario	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
Aprendizaje	Capacidad para ser aprendido fácilmente por el usuario	Porcentaje de capacidad para ser aprendido fácilmente por el usuario	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.

	Documentación adecuada	Porcentaje de documentación adecuada	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
Protección de Errores de usuario	Presencia de entradas válidas	Verificación de entradas válidas.	Contar de número de ítems de entrada que son validados y compararlos con el número de ítems que necesitarían ser validados
Operatividad	Capacidad para ser operado y recordado por el usuario con facilidad	Porcentaje de la capacidad para ser operado y recordado por el usuario con facilidad.	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
	Capacidad para reducir procedimientos de operación	Porcentaje de reducción del procedimiento de operación	Cuestionario a partir de una lista de chequeo, realizada por el

			equipo evaluador.
	Presencia de valores por defecto	Porcentaje de presencia de valores por defecto.	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.
	Presencia de mensajes claros de usuario	Porcentaje de presencia de mensajes claros de usuario.	Cuestionario a partir de una lista de chequeo, realizada por el equipo evaluador.

3.1.3 Estructura del Estándar IEEE 1074

El estándar IEEE 1074 se utiliza para establecer el inicio y fin en el desarrollo de un proyecto de software, el cual consta de seis secciones lógicas, y cada una se complementa con grupos de actividades, como se muestra en la Tabla 21, los cuales son una guía para que el equipo de desarrollo las realice paso a paso [30].

Tabla 21. Secciones Lógicas del Estándar IEEE 1074

Sección	**Grupos**
Modelamiento del Ciclo de Vida	Elección del modelo del ciclo de vida.
Gestión del Proyecto	Iniciación del proyecto
	Planificación del proyecto
	Monitoreo y control del proyecto
Pre-Desarrollo	Exploración de conceptos
	Asignación del sistema
Desarrollo	Requerimientos de software
	Diseño
	Implementación
Post-Desarrollo	Instalación

	Operación y soporte
	Mantenimiento
Procesos Integrales	Evaluación
	Gestión de configuración de software
	Desarrollo de documentación
	Capacitación

El conjunto de procesos que se describen en la IEEE 1074 es un estándar para el modelo del ciclo de vida del software, que ayuda a describir y establecer un grupo de actividades de forma organizada y así cumplir con cada uno de las secciones que la componen. Un equipo de desarrollo de software deberá elegir una metodología que se adapte al tipo de proyecto que se va a realizar y a los recursos disponibles, y cada metodología tendrá un estándar de ciclo de vida, de esta forma se asegura de que se cumpla con las actividades necesarias para que un proyecto tenga un inicio y fin [31].

Tabla 22. Estructura del Estándar IEEE 1074

Sección	**Grupos**	**Actividades**	**Descripción de Actividades**
Modelamiento del ciclo de vida	Selección de un ciclo de vida	Ciclo de vida XP	El ciclo de vida de XP, se adapta a proyectos en los que los requerimientos son cambiantes, también satisface la necesidad de que exista mayor interacción con el usuario, fomenta el trabajo en equipo y resulta ser flexible para el equipo de desarrollo ya que al tener n iteraciones durante la realización del proyecto se puede realizar avances de forma incremental, con cada una de las revisiones que se realizan con los usuarios se permite registrar observaciones y posteriormente realizar las correcciones pertinentes y finalmente obtener

			el producto de software que satisface los requerimientos.
Gestión del proyecto	Iniciación del proyecto	Realización de estimaciones	- Estimaciones del proyecto - Asignaciones de los recursos - Métodos de recolección y análisis de datos
		Asignación de los recursos del proyecto	
		Definición de métricas	
	Planificación del proyecto	Planificación de revisiones	- Información del plan de revisión - Información del plan de evaluación - Información del plan de documentación
		Planificación de las evaluaciones de calidad	
		Planificación de la documentación	

	Monitoreo y control del proyecto	Gestión de riesgos	- Información del informe de la gestión de riesgos - Información del informe de análisis
Pre-Desarrollo	Exploración de conceptos	Identificación de necesidades	- Información preliminar de la necesidad - Enfoques posibles - Limitaciones y ventajas
		Formulación de soluciones potenciales	
		Realización de estudios de viabilidad	
	Asignación al sistema	Análisis de las funciones	- Descripción funcional del sistema - Requisitos no funcionales del hardware y humanos del sistema
		Descomposición de los requisitos del sistema	

			- Requisitos funcionales del software del sistema
Desarrollo	Requerimientos de software	Definición y desarrollo de los requisitos del software	- Requisitos preliminares del software
		Definición de los requisitos de las interfaces	- Requisitos de la interfaz del software
		Integración de los requisitos del software	
	Diseño	Realización del diseño arquitectónico	- Diseño arquitectónico del software
		Diseño de la base de datos	- Diseño de la base de datos
		Diseño de la interfaz	- Diseño de la interfaz

			- Diseño detallado del software
	Implementación	Creación de código ejecutable	- Código fuente - Código ejecutable - Base de datos - Documentación operativa - Registros de operaciones - Anomalías - Recomendaciones de mejora del software - Informe de problemas de mejora - Informe de problemas de corrección
		Creación de la documentación operativa	
		Revisión y retroalimentación con desarrolladores, clientes y usuarios	
		Recopilación e incorporación de cambios	
		Ajustes y seguimiento	
		Reaplicación del ciclo de vida	

Post-Desarrollo	Evaluación	Realización de revisiones	Todas las actividades dentro del proceso de pruebas se realizan con las especificaciones de la estructura de las normas ISO/IEC 25000.
		Desarrollo de procedimientos de prueba	
		Ejecución de la prueba	
		Presentación del informe de los resultados de la evaluación	
	Instalación	Distribución del software	- Información del informe de instalación - Aceptación del cliente
	Operación y soporte	Operación del sistema	- Registros de operaciones - Anomalías - Registro de peticiones de soporte
		Prestación de asistencia y consultas técnicas	

	Mantenimiento	Identificación de las necesidades de mejora del software	- Recomendaciones de mejora del software - Informe de problemas de mejora - Recomendaciones de mantenimiento
Procesos Integrales	Gestión de configuración de software	Realización del control de la configuración	- Identificación de la configuración
	Desarrollo de documentación	Implementación de la documentación	- Documentación
		Producción y distribución de la documentación	

	Capacitación	Desarrollo de los materiales de formación	- Manual de capacitación (Manual de usuario o manual técnico) - Materiales de capacitación - Presentaciones preparadas
		Implementación del programa de formación	

3.2 Desarrollo de la propuesta

3.2.1 Diagnóstico empresarial

Al haber realizado el análisis de los resultados que se obtuvieron a través de los instrumentos de recolección de la información que se aplicaron, se puede realizar un diagnóstico empresarial y definir requisitos funcionales y no funcionales, como se observa en la Tabla 23, con los cuales satisfacer las necesidades que se han planteado en el objetivo principal.

Tabla 23. Requisitos Funcionales y No Funcionales

Requisitos funcionales	Requisitos estratégicos	Análisis de viabilidad y factibilidad del proyecto
		Análisis de riesgos
	Requisitos técnicos	Requerimientos funcionales del software
		Base de Datos (Modelamiento y diseño de la base de datos)
		Diagrama de proceso
		Historias de usuario
		Documentación
Requisitos no funcionales	Requisitos de seguridad	Definición de roles y privilegios
		Vulnerabilidades

	Requisitos de usabilidad	Pruebas y evaluación de calidad aplicando las normas ISO/IEC 25000

3.2.2 Metodología propuesta

La metodología que se propone es una metodología híbrida para estandarizar el desarrollo de aplicaciones web en el EEASA, la cual ha sido creada a partir de las necesidades que presenta la empresa, en esta metodología se encuentran algunas prácticas existentes dentro de metodologías tradicionales y ágiles, por lo cual se le considera una metodología híbrida entre lo tradicional y lo ágil.

a. Características de la metodología

La metodología que se propone para el desarrollo de proyectos de software, tiene las siguientes características:

- Está diseñada para proyectos de desarrollo web.
- Proyectos que se desarrollan en un lapso de tiempo de seis meses a dos años.
- Equipos de desarrollo conformados de dos a cuatro integrantes, sin tener en cuenta al cliente y los usuarios.

b. Estrategias de trabajo aplicadas al desarrollo web

- ***Metodologías tradicionales***

  - Espiral

    La clave de este modelo es la minimización de inconvenientes durante el desarrollo, a través de la gestión de riesgos, se tomó esta característica para la metodología con la diferencia de que no se realizará el estudio y análisis de riesgos en cada iteración sino solo una vez para que la metodología sea más ligera.

- Proceso Unificado Racional (RUP)

  En este modelo principalmente se encuentra la gestión de requisitos, para lo cual se hace uso de arquitecturas basadas en componentes y modelar visualmente el software, para esto se hace uso del UML (Unified Modeling Languaje, en inglés).

- ***Metodologías ágiles***

  - Extreme Programming (XP)

    De este modelo se tomó el ciclo de vida, ya que es flexible y se puede trabajar por iteraciones e incremental durante el desarrollo, además de que se centra en la interacción con el usuario.

    En cada iteración se hace uso de historias de usuario en las que se puede registrar cambios, mejoras, nuevas funcionalidades y posteriormente integrarlas durante el desarrollo en la iteración n.

  - Scrum

    Este modelo se caracteriza por llevar a cabo un conjunto de tareas de forma regular con el objetivo principal de trabajar de manera colaborativa, para fomentar el trabajo en equipo, lo cual permite la asignación de roles más genéricos y flexibles, en lugar de asignar un rol junto con todas sus tareas a una sola persona.

  - Crystal

    Este modelo está diseñado para ser liviano y flexible, ya que en lugar de tratar de hacer que un proyecto se ajuste a un conjunto de reglas, Crystal adapta el método al proyecto, y para la organización del equipo de desarrollo se puede adaptar la metodología Kanban para el seguimiento de tareas, lo cual se puede realizar con un cronograma de actividades.

- DSDM (Dynamic System Development Method)

  El enfoque principal de este modelo es que se debe tener una sólida comprensión de los requisitos del proyecto y la solución propuesta, con lo que se puede medir la viabilidad y factibilidad, para que existan fundamentos sólidos antes de que se inicie el proyecto.

c. Roles de la metodología

En esta metodología se plantea que dentro del equipo de desarrollo deben existir los siguientes roles, que se muestran en la Tabla 24:

Tabla 24. Roles Dentro del Equipo de Desarrollo

Roles	**Descripción**
Cliente Departamento, Sección o Área Requirente	Especifica los requerimientos
Líder del proyecto Jefe de Área Informática del Departamento Requirente	Es el integrante del equipo que tiene mayor relación con el cliente y usuarios, además de ser el intermediario entre el cliente y administrador, por lo que deberá unir ideas, personas, recursos y realizar la toma de decisiones.
Administrador del proyecto Jefe de Área Informática del Departamento Requirente	El administrador se encarga de coordinar a los programadores, al revisor y al documentador, al igual que organizar las reuniones necesarias para analizar requerimientos, realizar el diseño, el plan del proyecto y las pruebas.

Programador	Son los integrantes del equipo encargados de codificar el diseño.
Revisor Líder de Desarrollo de la EEASA	El revisor debe verificar que se realizan las actividades de manera adecuada en cada fase del proceso de desarrollo de software.
Documentador	La función del documentador es generar documentos que respalden y documenten lo que se va generando a lo largo del desarrollo del proyecto.
Evaluador Líder de Infraestructura de la EEASA Líder de Desarrollo de la EEASA Comunicación Institucional	El evaluador es la persona encargada de aplicar las normas ISO/IEC 25000, para la evaluación de calidad del software.
Usuarios finales	Los usuarios finales serán las personas que interactúan con el software.

d. Estructuración de ciclos

Tabla 25. Estructuración de Ciclos de las Actividades de la IEEE 1074

Sección	Grupos	Roles
Gestión del Proyecto	Iniciación del Proyecto	Líder de proyecto Administrador de proyecto Documentador
	Planificación del proyecto	
	Monitoreo y control del proyecto	
Pre-Desarrollo	Exploración de Conceptos	Líder de proyecto Administrador de proyecto Documentador
	Asignación del sistema	
Desarrollo	Requerimientos de software	Programador Líder de proyecto Administrador de proyecto Documentador
	Diseño	
	Implementación	
Post-Desarrollo	Evaluación	Evaluador Líder de proyecto Administrador de proyecto Programador Documentador
	Instalación (Pase a producción)	
	Operación y soporte	
	Mantenimiento	

Procesos Integrales	Gestión de configuración de software	Líder de proyecto Administrador de proyecto Programador Documentador
	Desarrollo de documentación	
	Capacitación	

e. Proceso de desarrollo

Para dar inicio a la metodología según el estándar IEEE 1074, se tiene la sección del modelamiento del ciclo de vida, con el cual definirá el inicio, el desarrollo y el fin del proyecto, para esto se utilizará el modelo de ciclo de vida de la metodología XP que es la que se adapta a los requisitos y estructura organizacional de la empresa, dándole así también un enfoque de desarrollo ágil, que es lo más óptimo para el desarrollo de aplicaciones web.

Una vez que se ha definido el ciclo de vida, se debe continuar con el resto de secciones que se encuentran en el estándar IEEE 1074, y realizar las actividades planteadas para cumplir con el ciclo de vida del desarrollo de un proyecto.

- ***Sección Gestión del Proyecto***

En esta sección se propone un conjunto de procesos de iniciación, supervisión y control del proyecto a lo largo del ciclo de vida del software, para ello se tiene los siguientes grupos en los que se tiene actividades a llevarse a cabo:

- Iniciación del proyecto
    - Estimaciones del proyecto
    - Asignaciones de los recursos

      Para todas estas actividades se tendrá un formulario principal el cual es una guía con un orden de pasos a seguir, el formulario tiene el nombre de Iniciación del Proyecto y se lo puede revisar en el Anexo B.

- Métodos de recolección y análisis de datos

  En la actividad de métodos y recolección de análisis de datos, se tienen formularios de encuesta o ficha de observación según lo amerite el departamento, sección o área requirente, estos formularios se encuentran en el Anexo C y Anexo D.

- Planificación del proyecto

  - Información del plan de revisión
  - Información del plan de evaluación
  - Información del plan de documentación

    Para complementar el grupo de actividades de planeación, se tiene que establecer un cronograma de actividades, el cual se encuentra en el Anexo E.

- Monitoreo y control del proyecto

  - Información del informe de la gestión de riesgos
  - Información del informe de análisis de riesgos

    Para llevar a cabo el análisis y gestión de riesgos se tiene el apartado Gestión de Riesgos que es una guía para el equipo de desarrollo y se encuentra en el formulario principal Iniciación del Proyecto y se lo puede revisar en el Anexo B.

Al terminar el formulario principal Iniciación del Proyecto, junto con el cronograma de actividades y las técnicas de recolección de la información, será el primer entregable que se deba presentar al encargado de revisiones, rol que se habrá definido dentro del formulario Iniciación del Proyecto.

- ***Sección de Pre-Desarrollo***

Es esta sección se encuentran grupos de actividades, con los cuales se tiene que identificar la necesidad por la que se va a realizar el desarrollo la aplicación web, y las posibles soluciones, también un estudio de viabilidad del proyecto y establecer requisitos no funcionales de hardware y requisitos funcionales del software.

- Exploración de conceptos
    - Información preliminar de la necesidad
    - Enfoques posibles
    - Limitaciones y ventajas (Estudio de viabilidad y factibilidad)
- Asignación del sistema
    - Descripción funcional del sistema
    - Requisitos no funcionales del hardware y humanos del sistema
    - Requisitos funcionales del software del sistema

Para el cumplimiento de las actividades se tiene un segundo formulario principal con el nombre de Exploración de Conceptos, que se encuentra en el Anexo F, el cual se deberá seguir paso a paso para continuar con la siguiente sección, una vez terminado el formulario será el segundo entregable que se deba presentar al revisor.

Al haber presentado al revisor el formulario de Exploración de Conceptos lo siguiente, según la estructura organizacional de la empresa, será presentar al comité informático los formularios de Iniciación del Proyecto y Exploración de Conceptos, para la aprobación de la iniciación del proyecto, para esto también se utilizará con el formulario de Aprobación para Inicio de Proyecto, que está en el Anexo G.

- ***Sección de Desarrollo***

En esta sección se encuentran actividades que se deben llevar acabo después de que haya sido aprobado el inicio del proyecto, esto quiere decir que el formulario de Aprobación para Inicio de Proyecto debe estar firmado por los miembros del comité informático, según la estructura organizacional de la empresa.

Para dar inicio a la sección de Desarrollo se debe empezar por obtener requerimientos por parte del departamento, área o sección requirente, para después continuar con el diseño de base de datos, diseño de software e interfaz, y lo siguiente es la implementación grupo dentro del cual se realiza desarrollo el código fuente con el que se pretende satisfacer los requerimientos planteados.

- Requerimientos de software
    - Requisitos preliminares del software
    - Requisitos de la interfaz del software

      Para la obtención de requerimientos de software, para lo cual se tiene un tercer formulario con el nombre de Requerimientos de Software e Implementación, el cual se lo puede revisar en el Anexo H.

- Diseño
    - Diseño arquitectónico del software
    - Diseño de la base de datos
    - Diseño de la interfaz
    - Diseño detallado del software

      En el diseño arquitectónico del software, diseño de base de datos e interfaz, se debe realizar un diagrama en el que se muestre los roles existentes y un segundo diagrama de procesos, a estos diagramas se los conoce como diagramas UML, como se muestra en la Figura 20 y Figura 21, los cuales se los puede desarrollar con el software StarUML y el modelamiento de una base de datos relacional.

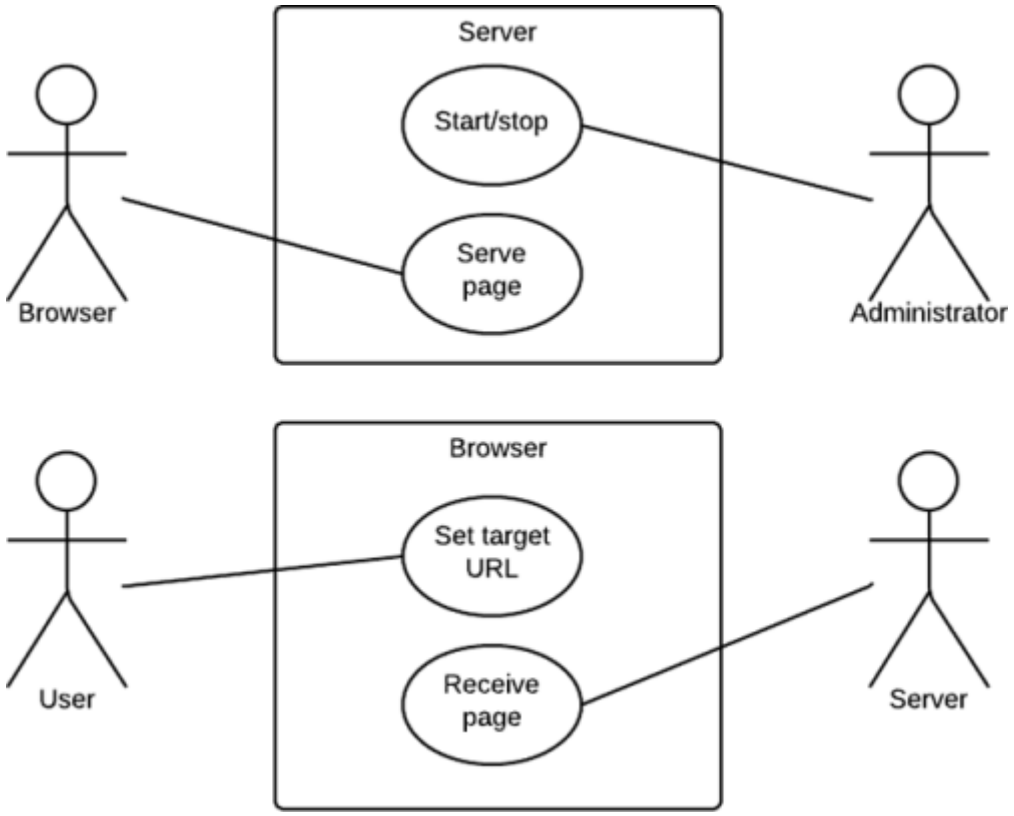


Figura 20. Diagrama de Roles y Privilegios

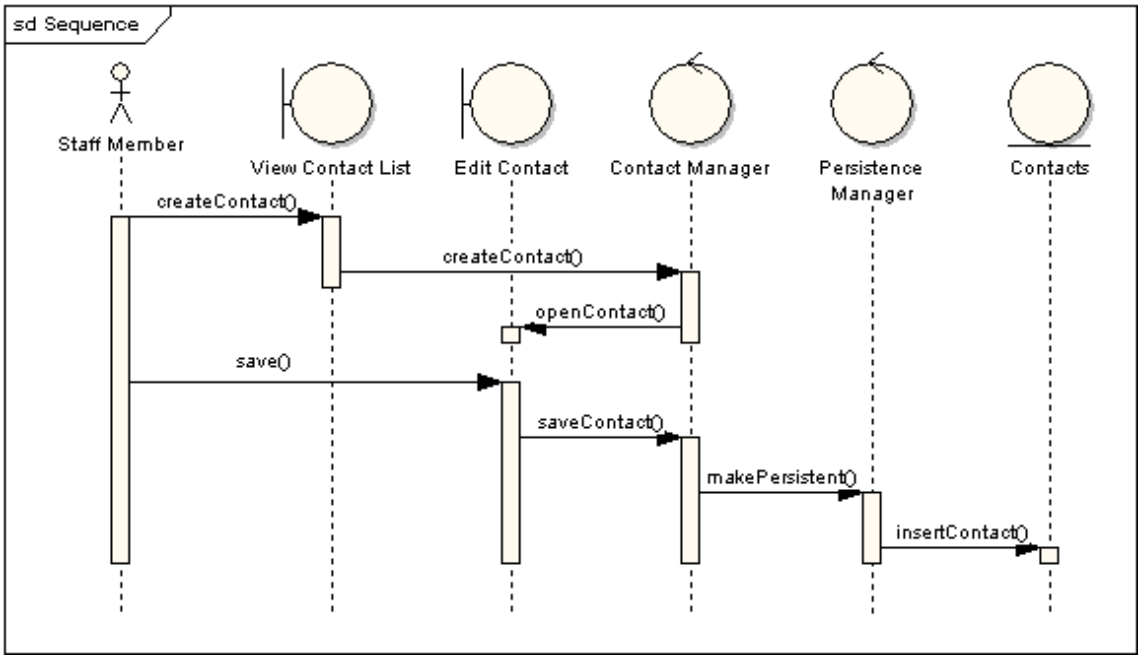


Figura 21. Diagrama de Procesos

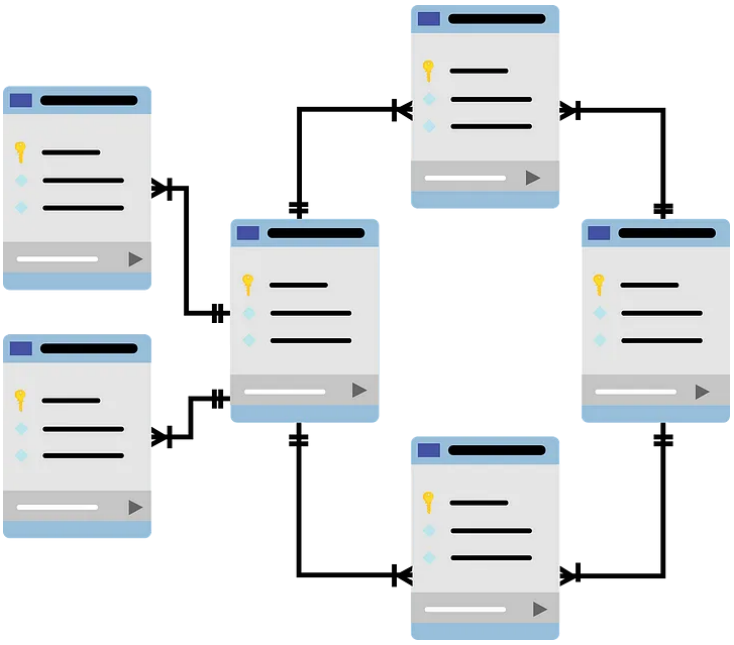


Figura 22. Esquema de Bases de Datos Relacional

- Implementación
    - Código fuente
    - Código ejecutable
    - Base de datos
    - Documentación operativa
    - Registros de operaciones
    - Anomalías
    - Recomendaciones de mejora del software
    - Informe de problemas de mejora
    - Informe de problemas de corrección

Durante el desarrollo de la aplicación web (Codificación), se debe realizar reuniones con clientes y usuarios para la retroalimentación, registro de operaciones, anomalías, recomendaciones e informes de mejora o correcciones del software, para esto de ser necesario se utiliza un formulario de nombre Iteración de Usuario en el que se registrará las peticiones de mejora o corrección, como se muestra en la Tabla 26.

Tabla 26. Formulario de Iteración de Usuario

Iteración de Usuario		
Fecha		
Nombre del Proyecto		
Número 1	Usuario Cliente	
Nombre de Historia	Agregar un módulo de administración	
Prioridad en Negocio: Alta	Riesgo de Desarrollo: Baja	
Programador Responsable: José Pérez		
Descripción: Se desea agregar un módulo de administración ya que es necesario para el requirente del proyecto		
Validación: El jefe de área del departamento requirente, necesita el acceso a más módulos y funciones del sistema para el cumplimiento de sus actividades		
Aprobación de Iteración		
Rol	Nombre	Firma
Líder de Proyecto		

Al finalizar con la primera iteración de la sección de desarrollo se deberá presentar los siguientes documentos al revisor asignado; el formulario de Requerimientos de Software e Implementación, diagrama de procesos, diagrama de roles y privilegios e iteraciones de usuario.

- ***Sección de Post-Desarrollo***

En esta sección se realizarán pruebas de software y evaluación de calidad de software, para posteriormente llevar el producto, la aplicación web, a producción, en la cual de igual forma se realizará registro de operaciones, de anomalías y el mantenimiento para llevar acabo las recomendaciones de mejora del software.

- Evaluación de Calidad
    - Revisión de funcionalidad del software
    - Desarrollo de procedimientos de pruebas de calidad
    - Ejecución de las pruebas de calidad del software
    - Presentación del informe de los resultados de la evaluación de calidad

Para dar inicio a la evaluación de calidad y cumplir con todas las actividades que la componen se utilizará las especificaciones de la estructura de las normas ISO/IEC 25000, para lo que se tiene un formulario con el nombre de Evaluación e Instalación que está en el Anexo I, el cual contiene de forma detallada las características y métricas que se van evaluar, con su respectivo porcentaje de ponderación según se había establecido con el equipo de desarrollo.

Durante la evaluación también se deben realizar pruebas de software las cuales las aplicarán el equipo de desarrollo, para lo cual se utilizará el formulario con el nombre de Formulario de Pruebas, en el que se registrará el desempeño del software, y que se encuentra en el Anexo J.

Una vez que se ha realizado la evaluación de calidad del software, y haya sido aprobado, entonces se volverá al comité informático, para presentar el siguiente formulario con el nombre de Formulario Para el Pase a Producción, que está en el Anexo K, ya que la estructura organizacional de la empresa así lo indica.

- Instalación
    - Pase a producción
    - Información del informe de instalación
    - Aceptación del cliente

    Para el cumplimiento de la actividad de instalación, se deberá utilizar la segunda parte del formulario de Evaluación e Instalación, el cual se encuentra en el Anexo L.

- Operación y Soporte
    - Registros de operaciones
    - Registro de peticiones de soporte

- Mantenimiento
    - Recomendaciones de mejora del software
    - Informe de problemas de mejora
    - Recomendaciones de mantenimiento

      Para el cumplimiento de las actividades de operación y mantenimiento, en el cual se registrará operaciones, peticiones de mejora y recomendaciones de mantenimiento, se utilizará el apartado de Mantenimiento que se encuentra en la segunda parte del formulario de Evaluación e Instalación, el cual se encuentra en el Anexo L, y de ser necesario también se puede hacer uso del formulario de nombre Iteración de Usuario en el que se registrará las peticiones de mejora o corrección que se encuentra en la Tabla 26.

- ***Sección de Procesos Integrales***

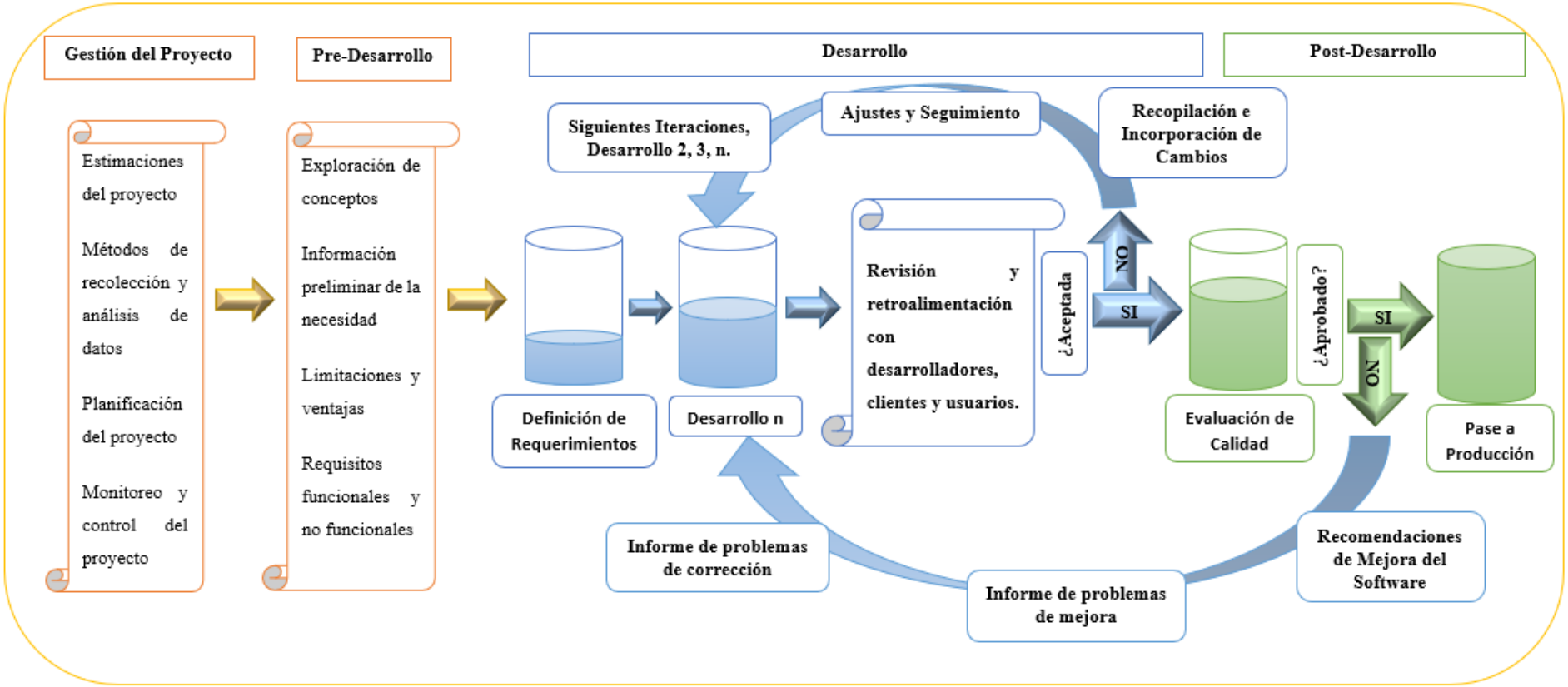
En esta sección se encuentran las actividades finales, así como señala el estándar IEEE 1074 para la aplicación del ciclo de vida del desarrollo de un proyecto, por esta razón los grupos de actividades indican identificación y gestión de configuración, documentación y capacitación para el departamento, sección o área requirentes los cuales serán los usuarios finales de la aplicación web. Para el cumplimiento de las siguientes actividades se tiene el formulario con el nombre Desarrollo de Documentación y Capacitación, el cual se encuentran en el Anexo M.

- Gestión de configuración de software
    - Identificación de la configuración
- Desarrollo de Documentación
    - Documentación
- Capacitación
    - Manual de capacitación (Manual de usuario o manual técnico)

- Materiales de capacitación
- Presentaciones preparadas

f. Esquema

Figura 23. Esquema del Ciclo de Vida de la Metodología Híbrida



g. Validación de la metodología híbrida

Para la validación de la metodología híbrida, se va a realizar el desarrollo de la página web con el nombre SISGERH-PASANTIAS, en el cual se debe realizar el registro de los estudiantes que ingresan a realizar prácticas o pasantías en la EEASA.

- ***Sección de gestión del proyecto***

Para cumplir con las actividades que se encuentran en esta sección se aplicó la ficha de observación, con la que se obtiene información del inconveniente que se presenta al realizar la gestión de información de los pasantes, como se muestra en el Anexo N1.

Posteriormente se aplicó el formulario con el nombre Iniciación del Proyecto, como se muestra en el Anexo O, con el que se establece objetivos, asignación de recursos, roles dentro del desarrollo de la aplicación y el estudio de la gestión de riesgos.

- ***Sección de pre-desarrollo***

Es esta sección se identificó la necesidad por la que se va a realizar el desarrollo la aplicación web, también se realizó un estudio de viabilidad del proyecto y establecer requisitos no funcionales de hardware y requisitos funcionales del software, como se muestra en el Anexo P.

Una vez que se ha concluido con la sección de gestión del proyecto y la sección de pre-desarrollo, se debe continuar con el proceso de aprobación de inicio del proyecto, según la estructura organizacional de la empresa, para lo cual se utilizó el formulario con el nombre Aprobación de Inicio del Proyecto, como se muestra en el Anexo Q.

- ***Sección de desarrollo***

Para dar inicio a la sección de Desarrollo se debe empezar por obtener requerimientos por parte del departamento, como se muestra en el Anexo R, para después continuar con el diseño de base de datos, diseño de software e interfaz.

Diseño

Una vez que se obtiene los requerimientos funcionales por parte del departamento requirente, se debe iniciar con el diseño de la base de datos, como se muestra, en la Figura 24.

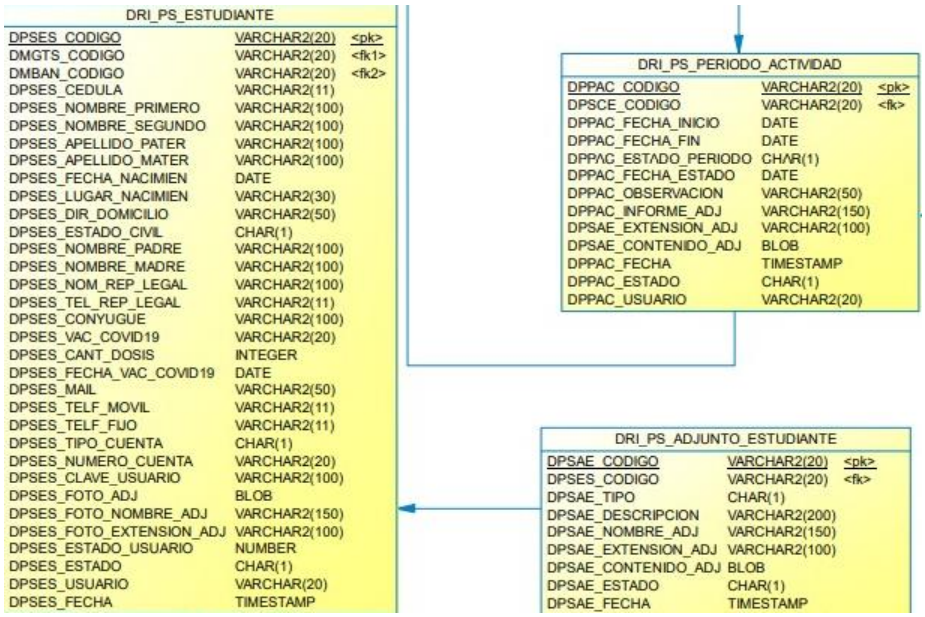


Figura 24. Diseño de la Base de Datos de SISGERH-PASANTIAS

En el diseño arquitectónico del software, se debe realizar un diagrama en el que se muestre los roles existentes y un segundo diagrama de procesos, como se muestra en la Figura 25 y Figura 26.

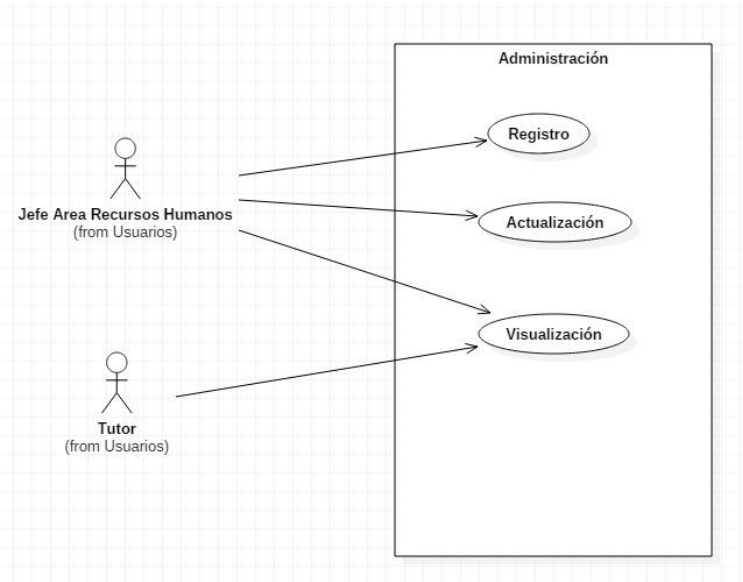


Figura 25. Diagrama para la Asignación de Roles

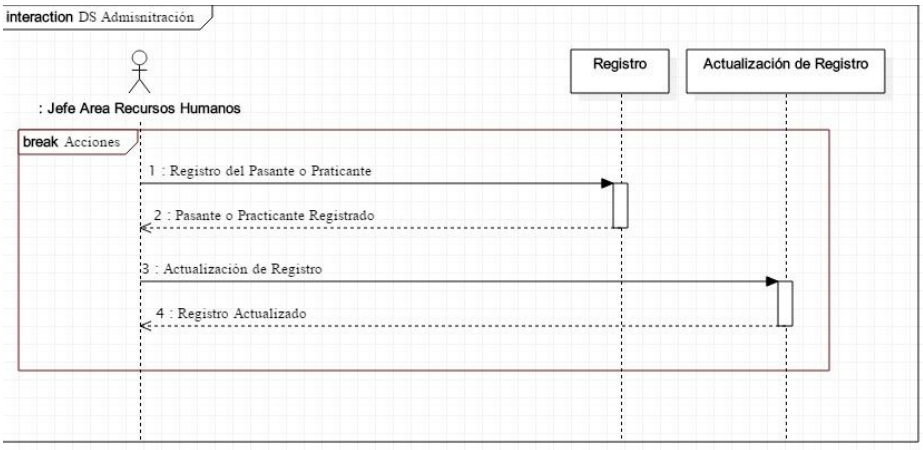


Figura 26. Diagrama de Procesos para Registro y Actualización de Información

Antes de continuar con el desarrollo de la aplicación web (Codificación), también se debe realizar un previo diseño de la interfaz de usuario que tendrán los módulos de administración y reportes, como se muestra en la Figura 27 y Figura 28.

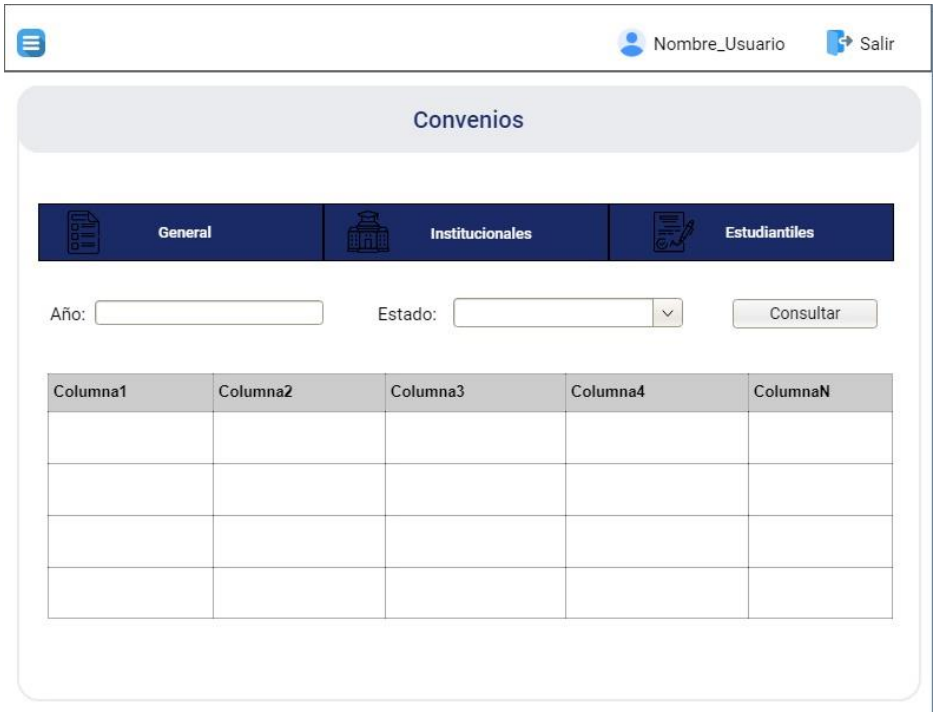


Figura 27. Diseño de la Interfaz de Administración

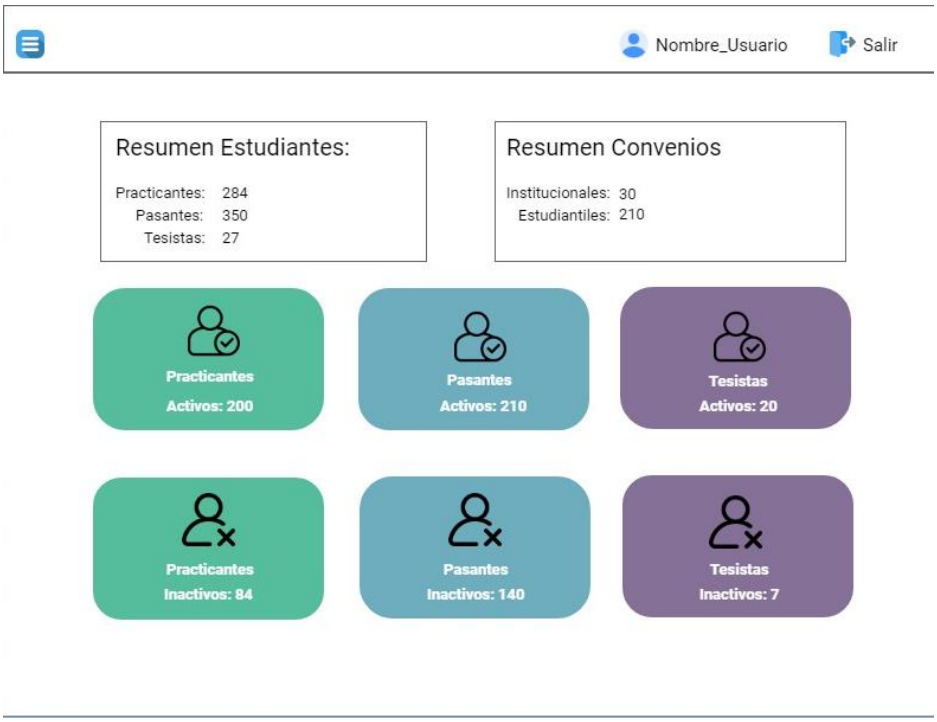


Figura 28. Diseño de la Interfaz de Reportes

Para continuar con las fases que se propone en la metodología, el siguiente paso es la implementación grupo dentro del cual se realiza desarrollo el código fuente con el que se pretende satisfacer los requerimientos planteados, obteniendo así los módulos de registro de estudiantes y de reportes, como se muestra en la Figura 29 y la Figura 30.

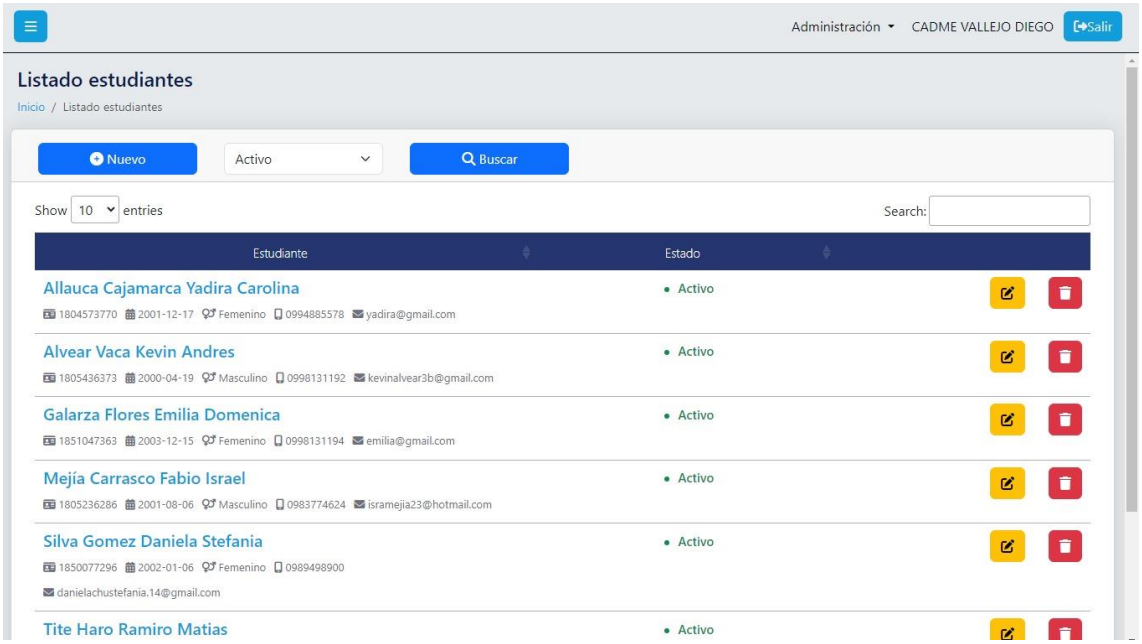


Figura 29. Módulo de Registro de Estudiantes

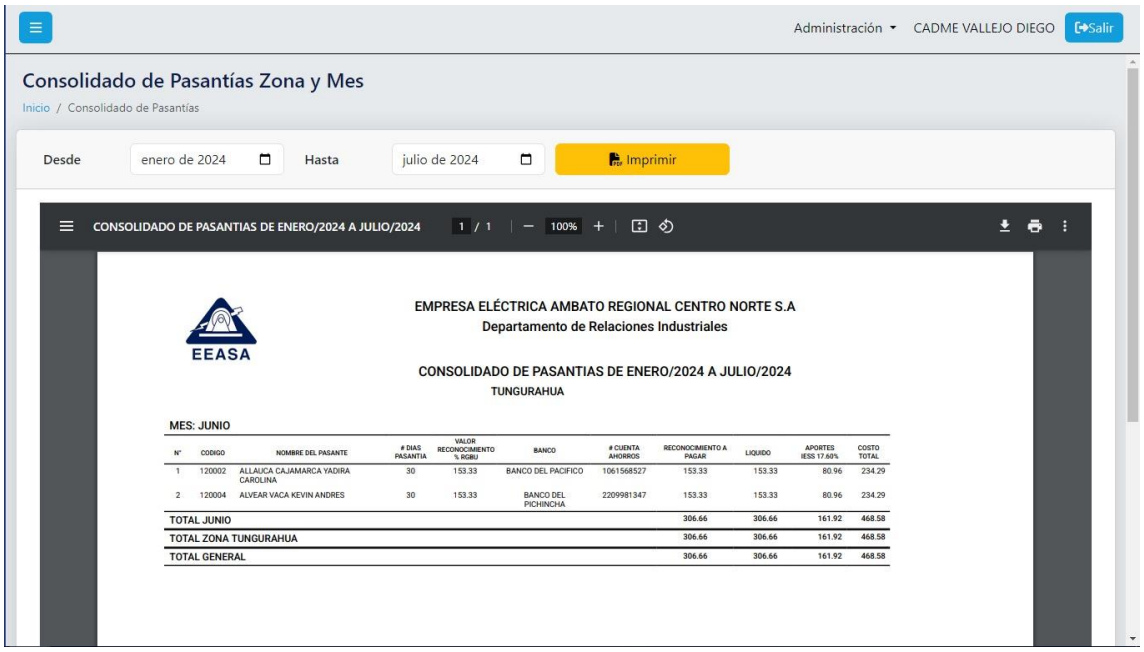


Figura 30. Módulo de Reportes

Una vez que se ha realizado el desarrollo de la aplicación web, para el cumplimiento de los requerimientos, se debe realizar reuniones con clientes y usuarios para la retroalimentación, y en caso de ser necesario se utiliza un formulario de nombre Iteración de Usuario en el que se registrará las peticiones de mejora o corrección, como se muestra en la Tabla 27 y la Tabla 28.

Tabla 27. Primera Iteración de Usuario para Petición de un Nuevo Rol

Iteración de Usuario		
Fecha	28-07-2024	
Nombre del Proyecto	SISGERH-PASANTIAS	
Número 1	Usuario: Cliente	
Nombre de Historia	Agregar el rol de pasante	
Prioridad en Negocio: Alta	Riesgo de Desarrollo: Baja	
Programador Responsable: Bryan Jurado		
Descripción: Se desea agregar el rol de pasante ya que es necesario para que el estudiante pueda llevar un registro de sus actividades.		
Validación: El jefe de área del departamento requirente, necesita que el tutor del pasante tenga acceso a las actividades del pasante para su revisión y validación.		
Aprobación de Iteración		
Rol	Nombre	Firma
Líder de Proyecto	Ing. Diego Cadme	APROBADO

Tabla 28. Segunda Iteración de Usuario para Agregar un Reporte

Iteración de Usuario		
Fecha	28-07-2024	
Nombre del Proyecto	SISGERH-PASANTIAS	
Número 1	Usuario: Cliente	
Nombre de Historia	Agregar un reporte	
Prioridad en Negocio: Alta	Riesgo de Desarrollo: Baja	
Programador Responsable: Bryan Jurado		
Descripción: Se desea agregar un reporte de reconocimiento económico por pasantías.		
Validación: El jefe de área del departamento requirente, necesita conocer el valor total a pagar por el reconocimiento económico de pasantías.		
Aprobación de Iteración		
Rol	Nombre	Firma
Líder de Proyecto	Ing. Diego Cadme	APROBADO

Al finalizar con el desarrollo de la aplicación web (Codificación) y haber realizado las revisiones con desarrolladores, clientes y usuarios se tiene los módulos con los que se cumple los requerimientos iniciales como se ve en la Figura 31 y la Figura 32.

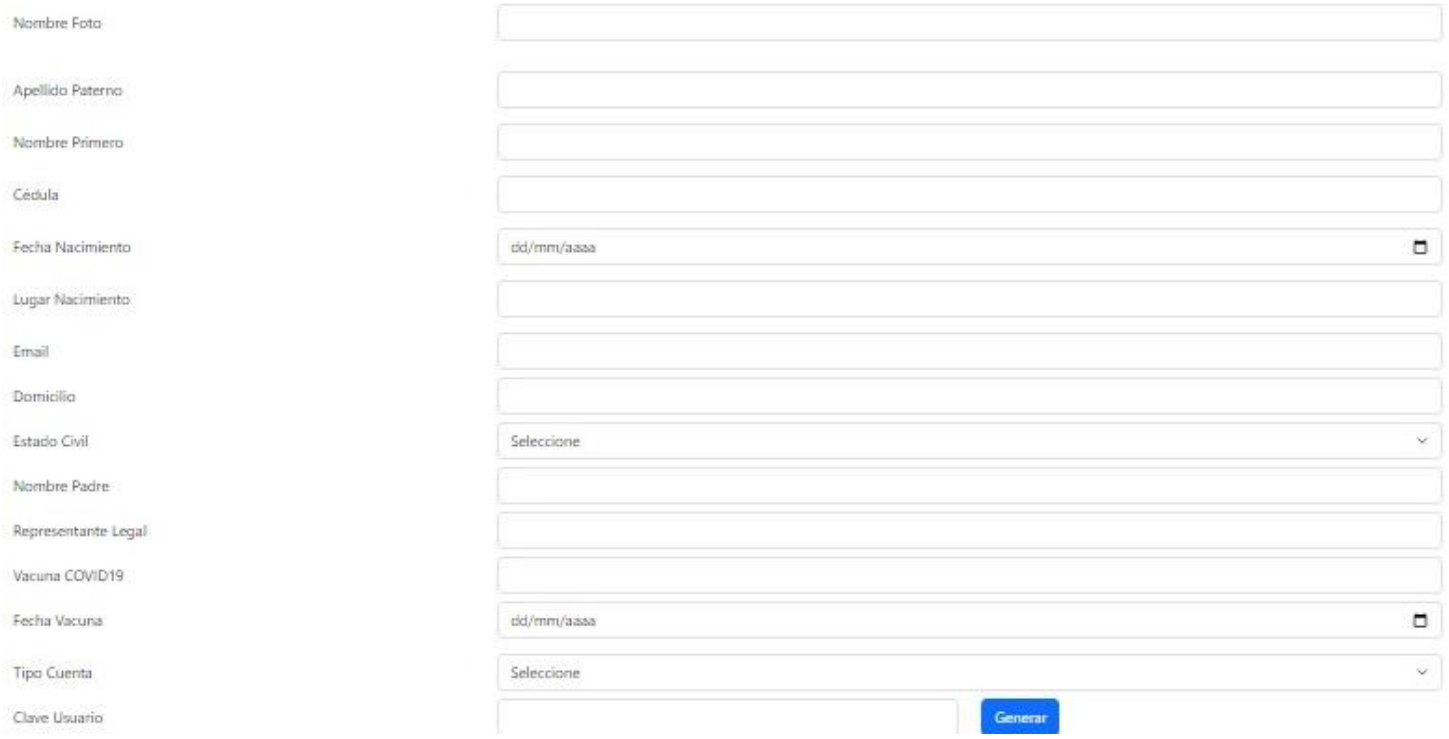


Figura 31. Registro de Estudiantes

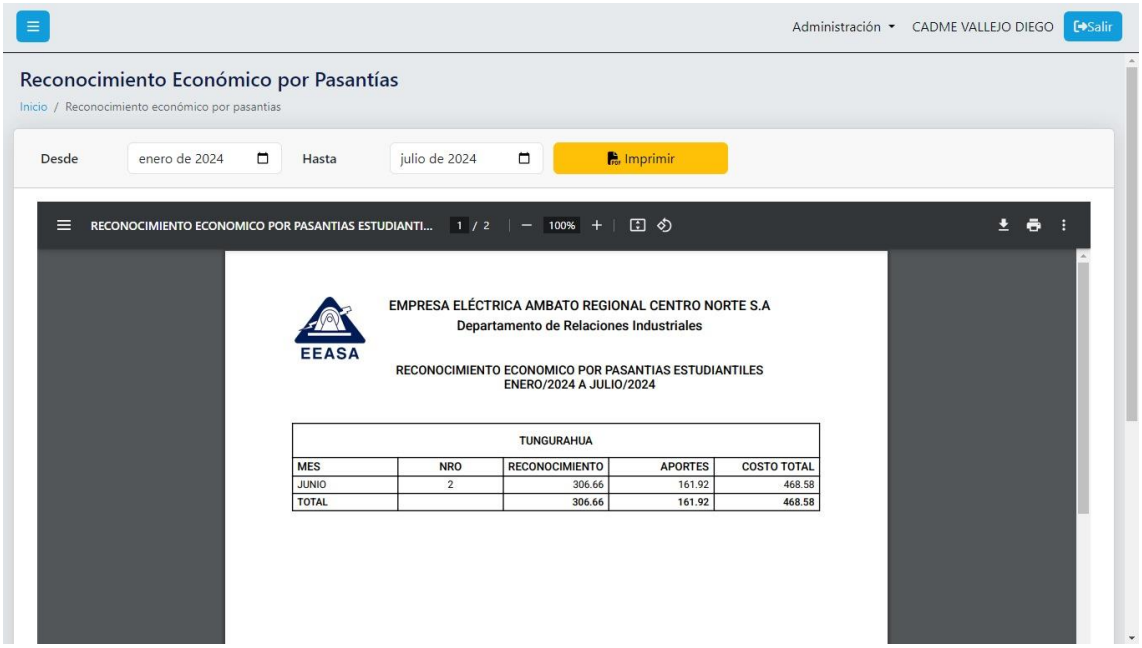


Figura 32. Reconocimiento Económico por Pasantías

- ***Sección de post-desarrollo***

En la siguiente fase al terminar con el desarrollo de software se realizó pruebas y evaluación de calidad de software, como se ve en el Anexo S, Anexo T y en el Anexo U.

Una vez que se ha realizado la evaluación de calidad se registró la calificación sobre 100, según se estableció en los porcentajes de ponderación de las características de las normas ISO/IEC 25000, con lo que se obtuvo 89.75/100, como se muestra al final del Anexo S

Los rangos de calificación, que han sido establecidos por el evaluador y que aplica a los niveles de puntuación y grados de satisfacción propuestos para la norma ISO/IEC 2504, como se muestra en la Tabla 29, son con los que se debe trabajar para conocer de forma descriptiva el resultado de la evaluación que al haber obtenido un 89.75/100 el software es muy bueno.

Tabla 29. Rangos de calificación

Rangos de Calificación	**Resultado de Evaluación**
90 - 100	Excelente
80,00 – 89,99	Muy Bueno
69,01 - 79,99	Aceptable
50 - 69,00	Malo
0 – 49,99	Deficiente

Una vez que se ha realizado el desarrollo de la aplicación web SISGERH-PASANTIAS, durante la sección de gestión del proyecto se planteó un tiempo estimado de 7 a 8 días para su realización, como se muestra en el formulario con el nombre Iniciación del Proyecto en el Anexo O. Al haber seguido paso a paso las fases de la metodología propuesta junto a las técnicas de trabajo establecidas (Formularios), se terminó el desarrollo del proyecto en un intervalo de tiempo de 5 días, como se muestra en la Tabla 30.

Tabla 30. Detalle de Fechas del Proyecto SISGERH-PASANTIAS

Proyecto SISGERH-PASANTIAS		
Fechas	Sección	Formularios
25-07-2024	Sección de gestión del proyecto	Ficha de observación. Formulario iniciación del proyecto.
	Sección de pre-desarrollo	Formulario exploración de conceptos. Formulario para aprobación del inicio del proyecto.
26-07-2024 27-07-2024 28-07-2024	Sección de desarrollo	Formulario de requerimientos de software e implementación. Formulario de iteración de usuario.
29-07-2024	Sección de post-desarrollo	Formulario de Evaluación e Instalación. Formulario de pruebas.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Durante el desarrollo del proyecto de investigación, se llegó a las siguientes conclusiones:

- Es importante identificar el método correcto para la recolección de información, durante la obtención de requisitos funcionales y no funcionales, de esta forma se puede clasificar y ordenar correctamente los requerimientos estratégicos, técnicos, de seguridad y usabilidad para el desarrollo de aplicaciones web.

- Para el desarrollo de software existen algunas metodologías tradicionales y ágiles disponibles, para obtener un desarrollo de software efectivo con alguna de las metodologías es importante conocer el equipo de desarrollo del que se dispone, de los recursos y tiempo para adaptarse a las técnicas y estrategias que proponen las metodologías, como es el caso de una metodología híbrida se puede adaptar al equipo de desarrollo y a la estructura organizacional con la que se trabaja tomando las herramientas de metodologías tradicionales y ágiles.

- Al ser una metodología híbrida diseñada específicamente para el desarrollo de aplicaciones web en la EEASA, las metodologías ágiles como; XP, Scrum, Crystal y DSDM, aportaron sus características, técnicas de trabajo y estrategias, y también se dispone de las metodologías tradicionales, las cuales son; Espiral y RUP, con las cuales se pudo complementar las tareas a realizarse en las fases del desarrollo para el cumplimiento de los requisitos empresariales.

- Con el estándar IEEE 1074, se puede trabajar el ciclo de vida y se dispone de actividades que se encuentran distribuidas por secciones para el cumplimiento del inicio, el desarrollo y el final de un proyecto de desarrollo de software.

- Al realizar una evaluación de calidad del software, se tiene que tomar en cuenta algunos enfoques como es la calidad interna, calidad externa y calidad de uso, para cubrir todos estos parámetros se debe hacer uso de la familia de normas ISO/IEC 25000, la cual consta de características, sub-características y métricas para llevar a cabo una correcta evaluación de calidad del software.

- Al plantear una metodología híbrida que se acople al desarrollo de aplicaciones web en la empresa, es de utilidad para el equipo de desarrollo ya que se combina el ciclo de vida de desarrollo ágil con actividades de metodologías tradicionales como la gestión de riesgos, estudios y análisis de viabilidad del proyecto, y un trabajo centrado en la recolección de requerimientos e interacción con el usuario, los cuales ayudan a reducir los inconvenientes y deficiencias durante el desarrollo del proyecto y al adjuntar una evaluación de calidad del software a las actividades dentro del ciclo de vida, el equipo de desarrollo puede llevar acabo un informe que demuestra que el software cumple con normas de calidad.

- Al haber realizado la validación de la metodología propuesta con el desarrollo de la aplicación web con el nombre SISGERH-PASANTIAS, en la que se tenía previsto un estimado de 7 a 8 días, con la ayuda de una metodología como guía documentada la cual sirve de apoyo para que el equipo de desarrollo realice las actividades que se encuentran en cada una de las fases del ciclo de vida planteado, se terminó en 5 días, demostrando que los inconvenientes como la deficiente recolección de requerimientos y el uso excesivo de tiempo durante el desarrollo de la aplicación web, han sido reducidos al haber realizado la gestión de riesgos, estudios y análisis de viabilidad antes de dar inicio al proyecto, además de haber realizado un trabajo centrado en la recolección de requerimientos e interacción con el usuario.

4.2 Recomendaciones

- Se recomienda utilizar las técnicas de recolección de información que se adapten al equipo de desarrollo, departamento o estructura organizacional de la empresa de la que se pretende obtener información sobre los inconvenientes que presenta y con ello establecer técnicas de trabajo que cumplan con los requerimientos que se han planteado.

- Las metodologías tradicionales y ágiles son utilizadas para el desarrollo de software, siempre y cuando se tome en cuenta que las metodologías tradicionales tienen un ciclo de vida lineal y muy poco flexible lo que se puede aplicar en una empresa con una estructura organizacional rígida y con un equipo de desarrollo con experiencia, y en cambio las metodologías ágiles se pueden aplicar en un

entorno mucho más flexible y en el que para el equipo de desarrollo sea importante un desarrollo iterativo e incremental y en el que se priorice interactuar con el cliente/usuario.

- Se debe tener en cuenta que el estándar IEEE 1074 es un modelo guía para el cumplimiento del ciclo de vida dentro de una metodología, por lo que se deben tomar las actividades y secciones que se consideren indispensables teniendo como base los requisitos funcionales y no funcionales planteados por la estructura organizacional de la empresa o el equipo de desarrollo.

- En el caso de aplicar las normas ISO/IEC 25000, se debe hacer un análisis y estudio de las características, sub-caracteríticas y métricas que la componen, porque de no ser el caso realizar una evaluación de calidad del software puede llegar a tomar demasiado tiempo o también ser deficiente, esto se puede evitar al realizar el análisis junto con el equipo de desarrollo y establecer un porcentaje de ponderación a cada una de las características y de esta forma definir qué características tienen mayor importancia para cumplir con los requisitos planteados por el equipo de desarrollo.

- Al proponer una metodología híbrida para estandarizar el desarrollo de aplicaciones web dentro de una empresa, se debe conocer la estructura organizacional de la misma y obtener un claro planteamiento del problema a través inconvenientes que se han presentado a lo largo del desarrollo de un proyecto, para esto es importante una adecuada aplicación de instrumentos de recolección y análisis de información, con esto se puede obtener un metodología con actividades que se adaptan al equipo de desarrollo y que ayude a reducir los inconvenientes durante el desarrollo de un proyecto.

REFERENCIAS BIBLIOGRÁFICAS

[1] L. de la C. Delgado Olivera y L. M. Díaz Alonso, «Modelos de Desarrollo de Software», *Rev. Cuba. Cienc. Informáticas*, vol. 15, n.º 1, pp. 37-51, mar. 2021.

[2] L. P. Dallos Bustos, D. S. Moncada Castaño, D. C. Ariza Hernández, y V. J. Franco Vargas, «Análisis comparativo entre metodologías ágiles y tradicionales para la gerencia de proyectos», Bachelor Thesis, Universidad EAN, 2020. Accedido: 3 de agosto de 2023. [En línea]. Disponible en:

https://repository.universidadean.edu.co/handle/10882/9559

[3] M. A. Santos-Romero y N. E. Escudero-López, «Propuesta de Metodología Híbrida y Base de Documentación para el Desarrollo de Software Actual», *Concienc. Tecnológica*, n.º 60, 2020, Accedido: 3 de agosto de 2023. [En línea]. Disponible en: https://www.redalyc.org/journal/944/94465715002/html/

[4] E. D. Gamboa Teneta, «Prototipo de un chatbot para compras online utilizando bot framework», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 2019. Accedido: 7 de abril de 2024. [En línea]. Disponible en: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/30105

[5] B. P. Lisintuña Poaquiza, «Sistema de gestión para la optimizacion del desarrollo de prácticas de los estudiantes en los laboratorios en la Facultad de Ciencia e Ingeniería en Alimentos», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera Ingeniería en Sistemas Computacionales e Informáticos, 2019. Accedido: 7 de abril de 2024. [En línea]. Disponible en: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/29784

[6] A. B. Alva Salcedo y J. B. Reyes Laynes, «Desarrollo e implementación de un sistema de ventas basado en la metodología Scrum y XP para el proceso de ventas de servicio de la empresa Emsoir», *AUTONOMA*, mar. 2019, Accedido: 7 de abril de 2024. [En línea]. Disponible en: http://repositorio.autonoma.edu.pe/handle/20.500.13067/702

[7] Y. L. Erika Dayana y A. G. Kleber Fabián, «Aplicación de la metodología kanban en el desarrollo del software para generación, validación y actualización de reactivos, integrado al sistema informático de control académico UNACH», bachelorThesis, Universidad Nacional de Chimborazo,2020, 2020. Accedido: 7 de abril de 2024. [En línea]. Disponible en: http://dspace.unach.edu.ec/handle/51000/6457

[8] O. E. Capuñay Uceda y J. M. Antón Perez, «Influencia de SCRUM en los plazos de entrega y rendimiento en los proyectos de las asignaturas de Desarrollo de Software», *Rev. Iberoam. Tecnol. En Educ. Educ. En Tecnol.*, n.º 29, pp. 36-42, sep. 2021.

[9] M. K. C. Gonzaga, W. J. O. Pazos, L. J. U. Meneses, y J. A. Esteban, «METODOLOGÍA HÍBRIDA DE DESARROLLO DE SOFTWARE COMBINANDO XP Y SCRUM», *Mikarimin Rev. Científica Multidiscip.*, vol. 5, n.º 2, Art. n.º 2, ago. 2019.

[10] E. Bautista-Villegas, «Metodologías agiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel», *Rev. Amaz. Digit.*, vol. 1, n.º 1, Art. n.º 1, ene. 2022, doi: 10.55873/rad.v1i1.168.

[11] M. R. 1 Ramírez, M. del C. S. 1 Soto, H. B. R. 1 Moreno, E. M. 1 Rojas, N. del C. O. 1 Millán, y R. F. R. U. A. de B. C. Cisneros, «Metodología SCRUM y desarrollo de Repositorio Digital», pp. 1062-1072, 2019.

[12] E. Reina y S. Patiño, «Evaluación de la calidad en uso de un sistema web/ móvil de control de asistencia a clases de docentes y estudiantes aplicando la norma ISO/IEC 25000 SQuaRe», *RISTI - Rev. Iber. Sist. E Tecnol. Inf.*, vol. E19, p. 108, abr. 2019.

[13] E. Figueroa, G. Carrión Barco, M. J. Sánchez Chero, A. H. Figueroa Alvarado, y W. A. Figueroa Neyra, «Modelo de calidad para plataformas e-learning basado en normas ISO/IEC 25000», *Rev. Univ. Zulia*, vol. 12, n.º 32, pp. 330-343, 2021.

[14] J. P. Zumba y C. A. L. Arreaga, «Evolución de las metodologías y modelos utilizados en el desarrollo de software.», *INNOVA Res. J.*, vol. 3, n.º 10, Art. n.º 10, oct. 2018, doi: 10.33890/innova.v3.n10.2018.651.

[15] L. I. Torres Valencia, «Estudio comparativo entre metodologías tradicionales y metodologías ágiles aplicadas a proyectos IT en entorno industrial.», 2020, Accedido: 7 de abril de 2024. [En línea]. Disponible en: https://academica-e.unavarra.es/xmlui/handle/2454/38990

[16] S. M. Velásquez, J. D. V. Montoya, M. E. G. Adasme, E. J. R. Zapata, A. A. Pino, y S. L. Marín, «Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software», *Rev. CINTEX*, vol. 24, n.º 2, Art. n.º 2, dic. 2019, doi: 10.33131/24222208.334.

[17] E. G. Maida y J. Pacienzia, «Metodologías de desarrollo de software», 2015, Accedido: 22 de abril de 2024. [En línea]. Disponible en: https://repositorio.uca.edu.ar/handle/123456789/522

[18] «Prototipo: qué es y para qué sirve», FREED TOOLS. Accedido: 22 de abril de 2024. [En línea]. Disponible en: https://freed.tools/blogs/ux-cx/prototipo

[19] J. Martinez Ruiz, M. E. Sandoval Parson, y K. Y. Tellez Rosales, «Desarrollo e implementación de sistema de facturación en Miscelánea “El Encanto” SYSFACT-MEN, usando la metodología ágil Crystal Clear», other, Universiad Nacional Autónoma de Nicaragua, Managua, 2014. Accedido: 22 de abril de 2024. [En línea]. Disponible en: https://repositorio.unan.edu.ni/6091/

[20] A. N. Cadavid, J. D. F. Martínez, y J. M. Vélez, «Revisión de metodologías ágiles para el desarrollo de software», *PROSPECTIVA*, vol. 11, n.º 2, pp. 30-39, 2013.

[21] L. M. Egas Robalino, «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», masterThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Maestría en Gerencia de Sistemas de

Información, 2020. Accedido: 23 de abril de 2024. [En línea]. Disponible en: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/30567

[22] P. K. Yumiseba Sanunga y C. R. Gutiérrez Mancheno, «Desarrollo de un sistema informático para automatizar el registro de pacientes atendidos por el servicio de emergencia del Hospital Pediátrico Alfonso Villagómez aplicando el framework Django.», bachelorThesis, Escuela Superior Politécnica de Chimborazo, 2019. Accedido: 23 de abril de 2024. [En línea]. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/11766

[23] D. L. Raigoza Valencia, «Desarrollo de una aplicación web para la gestión del montaje de red GPON físico y lógico del proveedor de servicios de internet Rapidred, mediante el framework Django.», mar. 2022, Accedido: 23 de abril de 2024. [En línea]. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/20146

[24] J. A. D. Prado, «Qué es la metodología Design Sprint y sus fases», UXABLES | Blog. Accedido: 23 de abril de 2024. [En línea]. Disponible en: https://www.uxables.com/diseno-ux-ui/que-es-la-metodologia-design-sprint-y-sus-fases/

[25] U. Continental, «Análisis y Requerimientos de Software: manual autoformativo interactivo», *Univ. Cont.*, 2017, Accedido: 23 de abril de 2024. [En línea]. Disponible en: https://repositorio.continental.edu.pe/handle/20.500.12394/4281

[26] L. Guerrero y J. Jaime, «Implementación de un sistema de información web para la gestión y visibilidad del grupo Guane.», feb. 2020, Accedido: 23 de abril de 2024. [En línea]. Disponible en: http://repository.unad.edu.co/handle/10596/31841

[27] M. V. Tarco Caisa y P. R. Calapucha Grefa, «Análisis del Desempeño entre MONGODB Y COUCHDB Utilizando Norma ISO/IEC 25000. Caso Práctico: Aplicación Web de Gestión de Documentos CONAGOPARE Chimborazo», bachelorThesis, Universidad Nacional de Chimborazo, 2019, 2019. Accedido: 17 de abril de 2024. [En línea]. Disponible en: http://dspace.unach.edu.ec/handle/51000/5569

[28] «NORMAS ISO 25000». Accedido: 21 de abril de 2024. [En línea]. Disponible en: https://iso25000.com/index.php/normas-iso-25000

[29] B. F. Paguay Duchi, «Desarrollo de un sistema de gestión de quirófanos para el Hospital Universitario Andino aplicando graphql», bachelorThesis, Riobamba, Universidad Nacional de Chimborazo, 2021. Accedido: 17 de abril de 2024. [En línea]. Disponible en: http://dspace.unach.edu.ec/handle/51000/7552

[30] C. N. Orna Jijón, «Modelo de calidad para la etapa de planificación de Proyectos Medianos de Software.», bachelorThesis, LATACUNGA / ESPE / 2013, 2013. Accedido: 21 de abril de 2024. [En línea]. Disponible en: http://repositorio.espe.edu.ec/jspui/handle/21000/6902

[31] M. D. Panizzi, A. Hossian, y R. García Martínez, «Implantación de sistemas: estudio comparativo e identificación de vacancias en metodologías usuales», presentado en XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)., 2016.

Accedido: 22 de abril de 2024. [En línea]. Disponible en: http://sedici.unlp.edu.ar/handle/10915/56646

[32] F. T. Capuz Pazmiño, «Desarrollo de una aplicación web para el control de permisos de funcionamiento del cuerpo de bomberos del cantón Tisaleo utilizando Java EE Y PostgreSQL.», bachelorThesis, Escuela Superior Politécnica de Chimborazo, 2020. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/13939

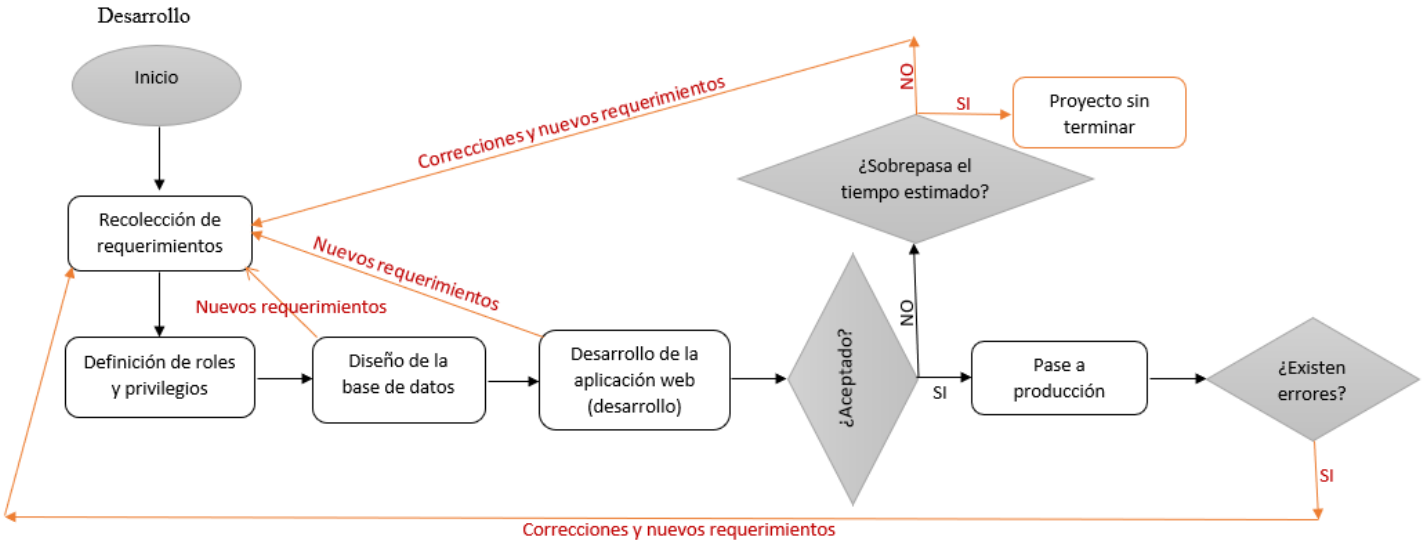
[33] M. R. Poma Velastegui, «Análisis metodológico del diseño UX/UI en el sitio web e-commerce Almacenes Juan Eljuri.», dic. 2022, Accedido: 20 de mayo de 2024. [En línea]. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/20539

[34] B. Soler y J. Pedro, «Diseño y desarrollo web. Análisis de casos», Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València, 2015. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://riunet.upv.es/handle/10251/49757

[35] C. Mateu, *Desarrollo de aplicaciones web*. Catalunya: Fundación para la Universitat Oberta de Catalunya, 2004., 2004. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: http://libros.metabiblioteca.org/handle/001/591

ANEXOS

Anexo A. Diagrama de Procesos de Desarrollo con Inconvenientes



Anexo B. Formulario de Iniciación del Proyecto

Tabla B1. Formulario de Iniciación del Proyecto

Fecha			
Nombre de Proyecto			
Monto USD incluido IVA			
Plazo ejecución		Plazo de Garantía o soporte	

I. Iniciación del Proyecto

(En esta sección se definen los aspectos generales e introductorios para la implementación del proyecto, así como los actores o participantes del mismo)

1. Resumen del Proyecto

(Realizar una pequeña introducción de las características del proyecto a implementarse, que se va adquirir, por qué y cuáles serían los principales beneficiados)

2. Antecedentes

(Definir algún proceso, recurso o sistema informático similar en la EEASA para ser tomado como referencia para atender a las necesidades del área solicitante y del mismo modo sirva como base para el nuevo proyecto a implementarse)

3. Definición de Métricas

(Definir algún método de recolección y análisis de información, por ejemplo; encuesta o ficha de observación según el departamento, sección o área requirente lo amerite y planteamiento del problema)

4. Objetivo General

(Definir los objetivos del proyecto a implementarse)

5. Objetivos Específicos

(Definir los objetivos específicos del proyecto a implementarse)

6. Justificación

(Definir la utilidad e importancia del proyecto considerando de igual manera la factibilidad técnica, operativa y económica)

7. Alcance

(Definir el alcance del proyecto en base a los objetivos, presupuesto referencial, limitantes y factibilidad del proyecto)

8. Actores Clave del Proyecto

(Definir a los distintos participantes o líderes involucrados en el desarrollo del proyecto de software)

Roles	Actores	Nombre	Cargo/Función
Cliente	Responsable informático del área solicitante		
Líder de Proyecto (Responsable informático del área solicitante)	Jefe de Área Informática del Departamento Requirente		
Administrador del Proyecto	Jefe de Área Informática del Departamento Requirente		
Programador	Integrantes del equipo encargados de codificar el diseño.		
Revisor	Líder de Desarrollo de la EEASA, Comunicación institucional.		
Documentador	Encargado de generar documentos que respalden y documenten lo que se va generando a lo largo del desarrollo del proyecto.		
Evaluador	Líder de Infraestructura de la EEASA Líder de Desarrollo de la EEASA Comunicación Institucional		

9. Gestión de Riesgos

a. Identificar los Riesgos

(Definir los riesgos o amenazas potenciales en relación con la gestión de proyectos de software y evaluar su importancia)

b. Analizar los Posibles Problemas de los Riesgos

(Una vez que el equipo identifique posibles riesgos, se deben responder las siguientes preguntas ¿Qué tan probable es que ocurran estos riesgos? Y si ocurren, ¿cuáles serán las ramificaciones?)

c. Clasificar y Priorizar los riesgos

(Priorizar los riesgos identificados del proyecto de software para centrarse en los de mayor impacto)

d. Desarrollar un plan de riesgos

(Cree un de plan de acción para la mitigación de riesgos)

e. Monitorear Continuamente las Amenazas de los Riesgos

(Mantener comunicación clara entre el equipo y las partes interesadas es vital cuando se trata de supervisar continuamente las posibles amenazas)

Formulario de Gestión de Riesgos	
Fecha	
Nombre del Proyecto	
Nombre del Encargado	

Identificar los Posibles Riesgos Para el Proyecto	Posibles Consecuencias o Impacto sobre el Proyecto	Alto	Medio	Bajo
Desconocimiento de la tecnología base del proyecto	Uso de tiempo extra para la capacitación del equipo de desarrollo			
Continuos cambios en el entorno organizacional	Dificultad para mantener la continuidad del desarrollo del proyecto			
Alta rotación del personal	Desconocimiento o poco entendimiento del software que se está desarrollando, por parte de nuevo integrantes del equipo de desarrollo			
Los objetivos del proyecto no son realistas	Objetivos que no se pueden cumplir por limitaciones de tiempo, de software, de hardware o de conocimiento			
Falta de claridad en los requerimientos o alta variación de los requerimientos.	Requerimientos mal planteados y que no cumplen con las necesidades que se habían planteado por parte del cliente/usuario			

Descripción del Plan de Riesgo	
Nº	Plan de Riesgo
1	Considerar dentro de la planeación del proyecto, el tiempo necesario para la capacitación del equipo de desarrollo en el caso de que sea necesario.
2	Priorizar el desarrollo del software, para el cumplimiento del mismo a pesar de que haya cambios en el personal o en el equipo de desarrollo.
3	Mantener la revisión continua por parte de la persona encargada de revisiones, para asegurarse de que existe una documentación adecuada en cada fase.
4	Revisión de los objetivos planteados por parte del líder de desarrollo, clientes/usuario y resto del equipo de desarrollo para mantener la integridad de los requerimientos y que todos los involucrados estén de acuerdo que se pueden realizar.
5	Realizar una recolección y análisis de requerimientos adecuados desde el inicio del proyecto, con reuniones y revisiones constantes con el cliente/usuario.

Anexo C. Formulario de Encuesta

Tabla C1. Formulario de Encuesta

Formulario de Encuesta		
(En este formulario se va a utilizar preguntas para dar a conocer el por qué se realiza el desarrollo del proyecto, y justificar la necesidad de la aplicación web)		
Fecha		
Nombre del Proyecto		
Nombre del Encargado		
Nº	Pregunta	Opciones
1	¿Está usted de acuerdo con que una aplicación web le ayudaría a mejorar el desempeño de sus actividades en el ámbito laboral?	- Totalmente de acuerdo - Medianamente de acuerdo - En desacuerdo - Totalmente en desacuerdo
2	¿Qué tan importante cree usted que es el uso de una aplicación web, para mejorar la administración de la información?	- Muy importante - Importante - Poco importante - No es importante
3	¿Está usted de acuerdo con que, una aplicación web sería de utilidad para obtener información de forma eficiente y exacta para la toma de decisiones en el área, departamento o sección?	- Totalmente de acuerdo - Medianamente de acuerdo - En desacuerdo - Totalmente en desacuerdo
4	¿Qué tan importante cree usted que es el uso de una aplicación web para automatizar el registro de actividades y obtención de informes que se realicen en el área, departamento o sección?	- Muy importante - Importante - Poco importante - No es importante
5	¿Está usted de acuerdo con utilizar una aplicación web para mantener un registro y control adecuado sobre las actividades que se realizan en el área, departamento o sección?	- Totalmente de acuerdo - Medianamente de acuerdo - En desacuerdo - Totalmente en desacuerdo
Procesamiento y Análisis de Resultados		
Pregunta Nº	Análisis de los Resultados	
1	Descripción del análisis de resultados de la pregunta 1	

Anexo D. Formulario de Ficha de Observación

Tabla D1. Formulario de Ficha de Observación

Formulario de Ficha de Observación					
(En este formulario se va a utilizar indicadores que se valorarán con escalas para dar a conocer el por qué y qué tan necesario es el desarrollo de la aplicación web)					
Fecha					
Nombre del Proyecto					
Nombre del Encargado					
Indicadores	Si		No		Observaciones
Se tienen claros los procesos que se quieren automatizar con el desarrollo de una aplicación web.					
Indicadores	Siempre	Casi Siempre	A veces	Nunca	Observaciones
Los procesos que se realizan actualmente son óptimos para cumplir con las actividades del departamento, área o sección.					
La forma en la que se registra la información es la adecuada.					
Los procesos para la gestión de la información son eficientes.					
Actualmente se puede obtener información oportuna y confiable a través de reportes requeridos por el departamento, área o sección.					
Conclusiones					

Anexo E. Cronograma de Actividades

Tabla E1. Cronograma de Actividades

Cronograma de Actividades																
Fecha																
Nombre del Proyecto																
Nombre del Encargado																
Nombre del Sistema	Mes 1				Mes 2				Mes 3				Mes n...			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Gestión del proyecto																
Estimaciones del proyecto																
Asignaciones de los recursos																
Métodos de recolección y análisis de datos																
Información del plan de revisión																
Información del plan de evaluación																
Información del plan de documentación																
Información del informe de la gestión de riesgos																
Información del informe de análisis																
Pre-Desarrollo																
Información preliminar de la necesidad																
Enfoques posibles																
Limitaciones y ventajas																
Descripción funcional del sistema																
Requisitos no funcionales del hardware y humanos del sistema																
Requisitos funcionales del software del sistema																
Desarrollo																
Requisitos preliminares del software																
Requisitos de la interfaz del software																
Diseño arquitectónico del software																
Diseño de la base de datos																
Diseño de la interfaz																
Diseño detallado del software																
Código fuente																
Código ejecutable																
Base de datos																
Documentación operativa																
Registros de operaciones																
Anomalías																
Recomendaciones de mejora del software																
Informe de problemas del software																

Informe de problemas de corrección																
Post-Desarrollo																
Realización de revisiones																
Desarrollo de procedimientos de prueba																
Ejecución de la prueba																
Presentación del informe de los resultados de la evaluación																
Distribución del software																
Operación del sistema																
Prestación de asistencia y consultas técnicas																
Identificación de las necesidades de mejora del software																
Procesos Integrales																
Identificación de la configuración																
Documentación																
Manual de capacitación																
Materiales de capacitación																
Presentaciones preparadas																

Anexo F. Formulario de Exploración de Conceptos

Tabla F1. Formulario de Exploración de Conceptos

II. Exploración de Conceptos	
(En esta sección se definen la información preliminar de la necesidad, enfoques, limitaciones, ventajas, requerimientos funcionales y no funcionales.)	
Fecha	
Nombre del Proyecto	
Nombre del Encargado	

1. Identificación de Necesidades
(Información preliminar del inconveniente, del cual surge la necesidad del sistema que presenta el requirente)

2. Formulación de Soluciones Potenciales
(Definir enfoques de posibles soluciones al inconveniente planteado por el requirente del sistema)

3. Estudios de Viabilidad
(Se definen ventajas y posibles limitaciones, que se presentan para el desarrollo del proyecto)
a. Técnico
(Se comprueba cuáles son los requisitos tecnológicos necesarios para llevar a cabo el proyecto, y se deben responder las siguientes preguntas; ¿Tiene la empresa el software necesario?, ¿Tiene la empresa el hardware necesario?, ¿Se tiene ya las habilidades necesarias para llevar a cabo el proyecto o hace falta formación?)

b. Económico
(Se estudia la rentabilidad del proyecto, principalmente en el caso de necesitar nuevo software o hardware, y también en el caso de necesitar apoyo de terceros)

c. Legal
(Se analiza si existe alguna barrera legal para la implementación del proyecto, ¿Infringe algún acuerdo que se haya firmado con otras partes (Incluidos acuerdos de confidencialidad) ?, ¿Infringe los derechos de propiedad intelectual de alguna otra organización?)

d. Operacional
(Se realiza el estudio para dar a conocer si el proyecto se puede realizar definiendo los procesos y requisitos funcionales)

e. Planificación
(Se debe planificar las fases del proyecto, ¿Se han definido las fechas límites del proyecto?, ¿Son realistas los plazos?, ¿Será el proyecto demasiado intensivo?)

4. Requisitos No Funcionales del Hardware
(Especificar los requisitos de hardware necesarios para la implementación y uso del sistema en el área solicitante) • Ejem. (Espacio en disco requerido 100 mb) • Ejem2. (Memoria RAM mínima 2 gb)

5. Requisitos Funcionales del software del sistema

(Especificar los requisitos de software necesarios para la implementación y uso del sistema en el área solicitante)
Ejem. (Navegador web Google Chrome)

6. Información Técnica del Sistema

Versión	(Especificar la versión del sistema)		
Plataforma	**Aplicación**		
	Tipo		
	Cliente - Servidor		
	Escritorio		
	Web	X	
	Móvil		
	Sistema Operativo		
	Linux	X	
	MS. Windows		
	Servidor de Aplicaciones	Oracle WebLogic 12C en Clúster (12.2.1.4.0) versión de java (build 1.8.0_191-b12)	
	Servidor Web		
	Bases de Datos	Oracle 19C (19.10.0)	
	Herramientas de Programación		
	Herramientas: (Especificar software necesario para el desarrollo e implementación del sistema) • Nom_herramienta1 (Ejem. IDE de desarrollo Visual Studio Code) • Nom_herramienta2		
	Seguridades		
		Seguro	Inseguro
	Nivel de Base de datos (BD)	X	
	Nivel de acceso a la aplicación	X	
	Nivel de funcionalidades del sistema	X	

Anexo G. Formulario para Aprobación de Inicio del Proyecto

Tabla G1. Formulario para Aprobación de Inicio del Proyecto

Formulario para Aprobación de Inicio del Proyecto		
Fecha		
Nombre del Proyecto		
Para la aprobación del proyecto se deben presentar los siguientes formularios: - Formulario de iniciación del proyecto - Formulario de exploración de conceptos		
Firmas de los Encargados de Aprobación		
Cargo	Nombre	Firma
Administrador del proyecto		
Líder de Desarrollo (Responsable informático del área solicitante)		
Líder de Infraestructura		
Coordinador Informático		
Comunicación Institucional		

Anexo H. Formulario de Requerimientos de Software e Implementación

Tabla H1. Formulario de Requerimientos de Software e Implementación

III. Requerimientos de Software e Implementación

(En esta sección se definen los requerimientos de la funcionalidad del software, interfaz del software, diseño de la arquitectura del software y diseño de la base de datos)

1. Definición y Desarrollo de los Requisitos del Software

(Definición de requisitos funcionales del software)

Fecha	
Nombre del Proyecto	
Módulo	(Especificar el módulo) Ejem. **Administración**
Objetivo	
Alcance	(Definir el alcance de los requisitos funcionales el módulo presente a partir del objetivo)

Detalle de Requerimientos		
Nº	Nombre	Descripción
1		
2		
3		
...		

Listado de Anexos		
Nº	Nombre	Descripción
1		
2		
3		

2. Diseño de la Interfaz

(Se define el diseño detallado del software)

Cuadros del Módulo del Sistema		
Cod. Módulo	Nombre Módulo	Observaciones
1	Nombre_mod_1	Observacion_mod_1
2	Nombre_mod_2	Observacion_mod_2

3. Diseño de la Base de Datos

(Se definen roles y privilegios)

Detalle de Roles y Privilegios de Usuario							
Resumen del Privilegio Asignado en el Sistema	Rol Asignado Dentro del Sistema						
	Ejem. Administrador	Ejem. Tutor	Ejem. Pasante	Ejem. Invitado	"Rol 1"	"Rol 2"	"Rol n"
Ejem. Creación de nuevos registros de pasantes	X	X					

Ejem. Creación de nuevos registros de actividades de pasante	X	X	X				
(Privilegio 1)							
(Privilegio 2)							
(Privilegio 3)							
(Privilegio n…)							

4. Implementación

(Se define la documentación que se va a realizar durante el desarrollo del proyecto, también se realizan revisiones y retroalimentación con desarrolladores, clientes y usuarios, recopilación e incorporación de cambios, ajustes y seguimiento, reaplicación del ciclo de vida)

Anexo I. Formulario de Evaluación e Instalación

Tabla I1. Formulario de Evaluación e Instalación

IV. Evaluación e Instalación	
(En esta sección se define la evaluación de calidad y distribución del software)	
Fecha	
Nombre del Proyecto	
Nombre del Encargado	

1. Cuestionario de Evaluación de Calidad del Software Calificación/100					
Adecuación Funcional 20pts					
Atributo	Métrica	Cumple 5pts	Medianamente Cumple 3pts	No Cumple 1pts	Total
Cobertura	Porcentaje de requerimientos cumplidos				
Completitud Funcional	Consistencia de la implementación funcional				
Precisión y exactitud computacional	Prueba de exactitud				
Coherencia	Prueba de exactitud				
				Total	

Fiabilidad 25pts					
Atributo	Métrica	Cumple 5pts	Medianamente Cumple 3pts	No Cumple 1pts	Total
Presencia de errores	Densidad de fallas contra casos de prueba				
Prevención de falla contra errores internos	Validación de entradas				
Prevención de falla contra errores internos	Prevención de fallas contra errores internos				
Atributo	Métrica	Cumple 10pts	Medianamente Cumple 6pts	No Cumple 2pts	Total
Continuidad de operación	Capacidad de continuidad en la operación				
				Total	

Eficiencia de Desempeño 20pts					
Atributo	Métrica	Cumple 8pts	Medianamente Cumple 5pts	No Cumple 2pts	Total
Capacidad de responder a las necesidades en un	Tiempo de respuesta				

tiempo de retorno aceptable					
Atributo	Métrica	Cumple 4pts	Medianamente Cumple 2pts	No Cumple 1pts	Total
Capacidad de optimizar el uso de memoria	Prueba de optimización de uso de memoria				
Capacidad para manejar concurrencia	Capacidad de manejo de concurrencia				
Capacidad para optimizar las comunicaciones	Capacidad de optimizar comunicaciones y uso de dispositivos I/O				
				Total	

Seguridad 25pts					
Atributo	Métrica	Cumple 7.5pts	Medianamente Cumple 4.5pts	No Cumple 1.5pts	Total
Capacidad de control acceso a información privada sin permisos	Capacidad de control de acceso a información sin permisos				
Seguridad de contraseña	Coherencia de contraseñas entre interfaces				
Atributo	Métrica	Cumple 10pts	Medianamente Cumple 6pts	No Cumple 2pts	Total
Integridad del sistema	Pruebas de asignación de roles y privilegios de usuario				
				Total	

Usabilidad 10pts					
Atributo	Métrica	Cumple 1.5pts	Medianamente Cumple 1pts	No Cumple 0.5pts	Total
Capacidad para proveer entradas y salidas entendibles	Porcentaje de capacidad para proveer entradas y salidas entendibles				
Capacidad para brindar claridad al usuario	Porcentaje de capacidad para brindad claridad al usuario				
Atributo	Métrica	Cumple 1pts	Medianamente Cumple 0.6pts	No Cumple 0.2pts	Total
Capacidad para ser aprendido fácilmente por el usuario	Porcentaje de capacidad para ser aprendido fácilmente por el usuario				
Documentación adecuada	Porcentaje de documentación adecuada				

Atributo	Métrica	Cumple 2pts	Medianamente Cumple 1.5pts	No Cumple 1pts	Total
Presencia de entradas válidas	Verificación de entradas válidas.				
Atributo	Métrica	Cumple 0.75pts	Medianamente Cumple 0.5pts	No Cumple 0.25pts	Total
Capacidad para ser operado y recordado por el usuario con facilidad	Porcentaje de la capacidad para ser operado y recordado por el usuario con facilidad.				
Capacidad para reducir procedimientos de operación	Porcentaje de reducción del procedimiento de operación				
Presencia de valores por defecto	Porcentaje de presencia de valores por defecto				
Presencia de mensajes claros de usuario	Porcentaje de presencia de mensajes claros de usuario.				
				Total	
Suma Total del Resultado de Evaluación:					

Anexo J. Formulario de Pruebas

Tabla J1. Formulario de Pruebas

Formulario de Pruebas						
Fecha	(Fecha que en la que se está realizando la prueba)					
Nombre del Proyecto	(Indicar el nombre del proyecto)					
Cod. Caso de Prueba	(Código de caso de uso, ejemplo: "CP001")					
Módulo Puesto a Prueba	(Especificar el módulo que fue puesto a prueba)					
Descripción	(Especificar la función que se va a poner a prueba)					
Responsable a Cargo	(Especificar el nombre del encargado a realizar la prueba)					
Requerimientos para la Prueba	(Que acción se realiza antes de ejecutar la opción a probar)					
Nº Actividad	Descripción	Datos Prueba	Resultado Esperado	Resultado Obtenido	Estado (cumple/no cumple)	Observación
1	Descripción de la actividad a realizar (Ejemplo: Ingresar al link del sistema:)	Datos que se deben tener en cuenta para la realización de la prueba	Resultado que se espera obtener (Ejemplo: ingresar al sistema)	Resultado real de la prueba (Ejemplo: Error de credenciales)	(Colocar "cumple" o "No Cumple" según corresponda)	Indicar las observaciones que se encontraron al realizar la actividad (Ejemplo: Tiempos de espera muy prolongados)
2	(Ejem2. Registro de nuevo pasante en el sistema)	(Ejem2. Uso de las variables: "nombre", "apellido", "cedula", "Institución"	(Ejem2. Registro realizado correctamente)	(Ejem2. Registro realizado correctamente)	(Ejem2. Cumple)	(Ejem2. El resultado del registro del nuevo pasante se evidencia sin tiempos de espera en el sistema)
3						
...	...	...	...	...	...	...
Resultados de la Prueba						
Defectos y desviaciones encontradas						
(Especificar los distintos defectos y desviaciones encontradas en el caso de que hayan existido, caso contrario colocar: "No fueron encontrado defectos o desviaciones durante el proceso de prueba") En la prueba se evidenciaron los siguientes defectos: • Resultado 1 • Resultado 2						

Anexo K. Formulario para el Pase a Producción

Tabla k1. Formulario para el Pase a Producción

Formulario Para el Pase a Producción		
Para realizar el pase a producción de deben presentar los siguientes formularios: - Formulario creación del proyecto - Formulario Roles y Privilegios - Formulario de Requerimientos funcionales - Manual de usuario o manual técnico - Diagrama de Base de Datos - Código Fuente - Formularios de Pruebas Informe de Resultado de la Evaluación de Calidad		
Firmas Para el Pase a Producción		
	Nombre	Firma
Administrador del proyecto		
Responsable informático del área solicitante		
Líder de Desarrollo		
Líder de Infraestructura		
Coordinador Informático		
Comunicación Institucional		

Anexo L. Formulario de Instalación y Mantenimiento

Tabla L1. Formulario de Instalación y Mantenimiento

2. Instalación
(Se define el registro de información del informe de instalación)

Fecha		
Nombre del Proyecto		
Datos del Sistema		
Características Generales del Sistema		
Alcance	Uso interno	X
	Uso externo	
	Uso de credenciales de acceso	
	Si	X
	No	
	Usuarios de aplicación	

3. Operación y Soporte
(Se define registros de operaciones y registros de peticiones de soporte)

Información Técnica del Sistema			
Versión	(Especificar la versión del sistema)		
Plataforma	Aplicación		
	Tipo		
	Cliente – Servidor		
	Escritorio		
	Web		
	Móvil		X
	Sistema Operativo		
	Linux		
	MS. Windows		
	Bases de Datos		Oracle 19C (19.10.0)
	Herramientas de Programación		
	Herramientas: (Especificar software necesario para el desarrollo e implementación del sistema) • Nom_herramienta1 (Ejem. IDE de desarrollo Visual Studio Code) • Nom_herramienta2		
	Framework (Front-End)	Angular 14.0.0	
	Back-end	Java API REST	
	Lenguaje de programación a utilizar en Back-end	Java 1.8	
Soporte	**Seguridades**		
		Seguro	**Inseguro**
	Nivel de la Base de Datos (BD)	X	

	Nivel de acceso a la aplicación	X	
	Nivel de funcionalidades del sistema	X	

4. Mantenimiento

(Se definen recomendaciones de mejora del software, informe de problemas de mejora y recomendaciones de mantenimiento)

Formulario para el informe de recomendaciones de mejora		
Fecha		
Nombre del Proyecto		
Nombre del Módulo	Problema Sujeto a Corrección o Mejora	Recomendación
(Especificar el nombre del módulo)	Descripción del problema (Ejm: El módulo no genera el reporte indicado en su descripción)	Describir una recomendación (Ejm: Describir correctamente el módulo o corregir el reporte con los parámetros sugeridos en el formato)
…	,,,	,,,

Anexo M. Desarrollo de Documentación y Capacitación

Tabla M1. Desarrollo de Documentación y Capacitación

V. Desarrollo de Documentación y Capacitación
(En esta sección se definen gestión de configuración de software, desarrollo de documentación y materiales de capacitación)

1. Desarrollo de Documentación y Gestión de Configuración
(Se definen documentos y entregables a producción)

Documentos y Entregables a Producción	
Requerimientos Funcionales	
Formulario de Requerimientos funcionales	X
Sistema	
Manual de usuario o manual técnico	X

2. Capacitación
(Se define los materiales de capacitación para el personal (cliente/usuarios), para esto se prepara presentaciones y se hace uso del manual técnico o manual de usuario)

Anexo N. Aplicación de Ficha de Observación

Tabla N1. Aplicación de Ficha de Observación

Formulario de Ficha de Observación					
(En este formulario se va a utilizar indicadores que se valorarán con escalas para dar a conocer el por qué y qué tan necesario es el desarrollo de la aplicación web)					
Fecha	25-07-2024				
Nombre del Proyecto	SISGERH-PASANTIAS				
Nombre del Encargado	Ing. Diego Cadme				
Indicadores	Si		No		Observaciones
Se tienen claros los procesos que se quieren automatizar con el desarrollo de una aplicación web.	X				
Indicadores	Siempre	Casi Siempre	A veces	Nunca	Observaciones
Los procesos que se realizan actualmente son óptimos para cumplir con las actividades del departamento, área o sección.			X		
La forma en la que se registra la información es la adecuada.			X		
Los procesos para la gestión de la información son eficientes.			X		
Actualmente se puede obtener información oportuna y confiable a través de reportes requeridos por el departamento, área o sección.			X		
Conclusiones					
Al no disponer de una aplicación web, con la que se gestione el registro de estudiantes que ingresan a la empresa a realizar prácticas o pasantías, todos los registros se realizan manualmente lo cual dificulta una gestión eficiente de la información.					

Anexo O. Aplicación del Formulario Iniciación del Proyecto

Tabla O.1 Aplicación del Formulario Iniciación del Proyecto

Fecha	25-07-2024		
Nombre de Proyecto	SISGERH-PASANTIAS		
Monto USD incluido IVA	1200 $		
Plazo ejecución	7 - 8 días	Plazo de Garantía o soporte	

VI. Iniciación del Proyecto

(En esta sección se definen los aspectos generales e introductorios para la implementación del proyecto, así como los actores o participantes del mismo)

10. Resumen del Proyecto

Realizar una aplicación web para la gestión de información de estudiantes que ingresan a realizar prácticas o pasantías en la EEASA.

11. Antecedentes

Los procesos principales para el uso de la aplicación web son el registro de estudiantes, y la obtención de reportes.

12. Definición de Métricas

Se utilizará la ficha de observación.

13. Objetivo General

Realizar una aplicación web que permita realizar la gestión de información de practicantes y pasantes de la EEASA.

14. Objetivos Específicos

- Realizar registro y actualización de información de practicantes y pasantes.
- Obtener reportes sobre la información de los pasantes.

15. Justificación

Actualmente se realiza el registro de practicantes y pasantes manualmente y eso dificulta la obtención de información, al momento de necesitar informes, reportes, o actualización de información de los mismos, por lo que se requiere automatizar todos los procesos.

16. Alcance

La aplicación web SISGERH-PASANTIAS será para administrar la información necesaria de los pasantes o practicantes, de todos los departamentos de la EEASA.

17. Actores Clave del Proyecto

(Definir a los distintos participantes o líderes involucrados en el desarrollo del proyecto de software)

Roles	Actores	Nombre	Cargo/Función
Cliente	Responsable informático del área solicitante	Ing. Yoconda Poveda	Jefe de área del DRI.

Líder de Proyecto (Responsable informático del área solicitante)	Jefe de Área Informática del Departamento Requirente	Ing. Mercedes Yánez	Jefe del área informática del DRI.
Administrador del Proyecto	Jefe de Área Informática del Departamento Requirente	Ing. Mercedes Yánez	Jefe del área informática del DRI.
Programador	Integrantes del equipo encargados de codificar el diseño.	Bryan Jurado	
Revisor	Líder de Desarrollo de la EEASA, Comunicación institucional.	Ing. Diego Cadme Ing. Mercedes Yánez	Líder de desarrollo del área informática del DRI. Jefe del área informática del DRI.
Documentador	Encargado de generar documentos que respalden y documenten lo que se va generando a lo largo del desarrollo del proyecto.	Bryan Jurado	
Evaluador	Líder de Infraestructura de la EEASA Líder de Desarrollo de la EEASA Comunicación Institucional	Ing. Diego Cadme Bryan Jurado	Líder de desarrollo del área informática del DRI.

18. Gestión de Riesgos

f. Identificar los Riesgos

(Definir los riesgos o amenazas potenciales en relación con la gestión de proyectos de software y evaluar su importancia)

g. Analizar los Posibles Problemas de los Riesgos

(Una vez que el equipo identifique posibles riesgos, se deben responder las siguientes preguntas ¿Qué tan probable es que ocurran estos riesgos? Y si ocurren, ¿cuáles serán las ramificaciones?)

h. Clasificar y Priorizar los riesgos

(Priorizar los riesgos identificados del proyecto de software para centrarse en los de mayor impacto)

i. Desarrollar un plan de riesgos

(Cree un de plan de acción para la mitigación de riesgos)

j. Monitorear Continuamente las Amenazas de los Riesgos

(Mantener comunicación clara entre el equipo y las partes interesadas es vital cuando se trata de supervisar continuamente las posibles amenazas)

Formulario de Gestión de Riesgos				
Fecha	25-07-2024			
Nombre del Proyecto	SISGERH-PASANTIAS			
Nombre del Encargado	Ing. Diego Cadme, Bryan Jurado			
Identificar los Posibles Riesgos Para el Proyecto	Posibles Consecuencias o Impacto sobre el Proyecto	Alto	Medio	Bajo
Desconocimiento de la tecnología base del proyecto	Uso de tiempo extra para la capacitación del equipo de desarrollo			X

Continuos cambios en el entorno organizacional	Dificultad para mantener la continuidad del desarrollo del proyecto			X
Alta rotación del personal	Desconocimiento o poco entendimiento del software que se está desarrollando, por parte de nuevo integrantes del equipo de desarrollo			X
Los objetivos del proyecto no son realistas	Objetivos que no se pueden cumplir por limitaciones de tiempo, de software, de hardware o de conocimiento			X
Falta de claridad en los requerimientos o alta variación de los requerimientos.	Requerimientos mal planteados y que no cumplen con las necesidades que se habían planteado por parte del cliente/usuario			X

Descripción del Plan de Riesgo	
Nº	Plan de Riesgo
1	Considerar dentro de la planeación del proyecto, el tiempo necesario para la capacitación del equipo de desarrollo en el caso de que sea necesario.
2	Priorizar el desarrollo del software, para el cumplimiento del mismo a pesar de que haya cambios en el personal o en el equipo de desarrollo.
3	Mantener la revisión continua por parte de la persona encargada de revisiones, para asegurarse de que existe una documentación adecuada en cada fase.
4	Revisión de los objetivos planteados por parte del líder de desarrollo, clientes/usuario y resto del equipo de desarrollo para mantener la integridad de los requerimientos y que todos los involucrados estén de acuerdo que se pueden realizar.
5	Realizar una recolección y análisis de requerimientos adecuados desde el inicio del proyecto, con reuniones y revisiones constantes con el cliente/usuario.

Anexo P. Aplicación del Formulario Exploración de Conceptos

Tabla P.1 Aplicación del Formulario de Exploración de Conceptos

VII. Exploración de Conceptos	
(En esta sección se definen la información preliminar de la necesidad, enfoques, limitaciones, ventajas, requerimientos funcionales y no funcionales.)	
Fecha	25-07-2024
Nombre del Proyecto	SISGERH-PASANTIAS
Nombre del Encargado	Ing. Diego Cadme, Bryan Jurado

7. Identificación de Necesidades

- Inconvenientes al realizar el registro y actualización de información de practicantes y pasantes.
- Dificultad para obtener reportes e informes, que son realizados por practicantes y pasantes.
- Deficiente obtención de informes para realizar el pago a pasantes mensualmente.

8. Formulación de Soluciones Potenciales

- Realizar una aplicación web con la que se lleve el registro y actualización de información de pasantes.
- Agregar un módulo de reportes para la obtención de informes de practicantes y pasantes.

9. Estudios de Viabilidad

(Se definen ventajas y posibles limitaciones, que se presentan para el desarrollo del proyecto)

f. Técnico

¿Tiene la empresa el software necesario?

Si, se cuenta con el software.

¿Tiene la empresa el hardware necesario?

Si, la empresa tiene los recursos de hardware necesarios.

¿Se tiene ya las habilidades necesarias para llevar a cabo el proyecto o hace falta formación?

El desarrollador y el encargado del proyecto, ya tienen las habilidades necesarias para llevar acabo el proyecto.

g. Económico

El proyecto es rentable, ya que la empresa cuenta con los recursos necesarios.

h. Legal

¿Infringe algún acuerdo que se haya firmado con otras partes (Incluidos acuerdos de confidencialidad)?

No infringe acuerdos firmados.

¿Infringe los derechos de propiedad intelectual de alguna otra organización?

No infringe derechos de propiedad intelectual.

i. Operacional

El desarrollador puede llevar acabo los procesos necesario establecidos dentro de la aplicación web.

j. Planificación

¿Se han definido las fechas límites del proyecto?

Si se han establecido fechas límite.

¿Son realistas los plazos?

Si se puede cumplir con los objetivos en el plazo establecido.

¿Será el proyecto demasiado intensivo?
El proyecto será ligeramente intenso de acuerdo a las horas de trabajo y el plazo establecido.

10. Requisitos No Funcionales del Hardware

- Espacio en disco requerido 250Gb
- Memoria RAM mínima 8 Gb

11. Requisitos Funcionales del software del sistema

- Navegador web Google Chrome
- StarUML
- Oracle 19c
- NetBeans - Java
- Visual Studio Code

12. Información Técnica del Sistema

Versión	(Especificar la versión del sistema)		
Plataforma	Aplicación		
	Tipo		
	Cliente - Servidor		
	Escritorio		
	Web	X	
	Móvil		
	Sistema Operativo		
	Linux		
	MS. Windows	X	
	Servidor de Aplicaciones	Oracle WebLogic 12C en Clúster (12.2.1.4.0) versión de java (build 1.8.0_191-b12)	
	Servidor Web		
	Bases de Datos	Oracle 19C (19.10.0)	
	Herramientas de Programación		
	Herramientas: • IDE de desarrollo Visual Studio Code		
	Seguridades		
		Seguro	Inseguro
	Nivel de Base de datos (BD)	X	
	Nivel de acceso a la aplicación	X	
	Nivel de funcionalidades del sistema	X	

Anexo Q. Aplicación del Formulario para Aprobación del Inicio del Proyecto

Tabla Q.1 Aplicación del Formulario para Aprobación del Inicio del Proyecto

Formulario para Aprobación de Inicio del Proyecto		
Fecha	25-07-2024	
Nombre del Proyecto	SISGERH-PASANTIAS	
Para la aprobación del proyecto se deben presentar los siguientes formularios: - Formulario de iniciación del proyecto - Ficha de Observación - Formulario de exploración de conceptos		
Firmas de los Encargados de Aprobación		
Cargo	Nombre	Firma
Administrador del proyecto	Ing. Diego Cadme	APROBADO
Líder de Desarrollo (Responsable informático del área solicitante)	Ing, Diego Cadme	APROBADO
Líder de Infraestructura	Ing. Ivan Vargas	APROBADO
Coordinador Informático	Ing. Mercedes Yánez	APROBADO
Comunicación Institucional	Ing. Carmiña Porras	APROBADO

Anexo R. Aplicación del formulario de Requerimientos de Software e Implementación

Tabla R.1 Aplicación del formulario de Requerimientos de Software e implementación

VIII. Requerimientos de Software e Implementación
(En esta sección se definen los requerimientos de la funcionalidad del software, interfaz del software, diseño de la arquitectura del software y diseño de la base de datos)

5. Definición y Desarrollo de los Requisitos del Software
La aplicación web, debe permitir al usuario realizar el registro y actualización de la información de practicantes y pasantes. Es necesario un módulo de reportes parala obtención de informes y reportes.

Fecha	26-07-2024
Nombre del Proyecto	SISGERH-PASANTIAS
Módulo	Registro de pasantes y practicantes
Objetivo	Realizar la gestión de la información de pasantes y practicantes.
Alcance	La aplicación web deberá ser utilizada por todos los departamentos de la EEASA, para la gestión de información de pasantes.

Detalle de Requerimientos		
Nº	Nombre	Descripción
1	Gestión de información	Registro y actualización de información
2	Reportes	Obtención de informes de pasantes y practicantes, y generación de reportes sobre los pagos mensuales a pasantes.

6. Diseño de la Interfaz
(Se define el diseño detallado del software)

Cuadros del Módulo del Sistema		
Cod. Módulo	Nombre Módulo	Observaciones
1	Módulo de administración	El administrador deberá tener acceso a todas las funciones de registro y actualización de información.
2	Módulo de reportes	Los reportes se generan para tutores de los practicantes y pasantes.

7. Diseño de la Base de Datos
(Se definen roles y privilegios)

Detalle de Roles y Privilegios de Usuario			
Resumen del Privilegio Asignado en el Sistema	M Rol Asignado Dentro del Sistema		
	Administrador	Tutor	Pasante
Creación de nuevos registros de pasantes	X	X	
Actualización de información de pasantes	X	X	

Generación de informes	X	X	X
Generación de reportes	X	X	

8. Implementación

(Se define la documentación que se va a realizar durante el desarrollo del proyecto, también se realizan revisiones y retroalimentación con desarrolladores, clientes y usuarios, recopilación e incorporación de cambios, ajustes y seguimiento, reaplicación del ciclo de vida)

Anexo S. Aplicación del Formulario de Evaluación e Instalación

Tabla S.1 Aplicación del Formulario de Evaluación e Instalación

IX. Evaluación e Instalación	
(En esta sección se define la evaluación de calidad y distribución del software)	
Fecha	29-07-2024
Nombre del Proyecto	SISGERH-PASANTIAS
Nombre del Encargado	Ing, Diego Cadme, Bryan Jurado

5. Cuestionario de Evaluación de Calidad del Software Calificación/100					
Adecuación Funcional 20pts					
Atributo	Métrica	Cumple 5pts	Medianamente Cumple 3pts	No Cumple 1pts	Total
Cobertura	Porcentaje de requerimientos cumplidos	X			5
Completitud Funcional	Consistencia de la implementación funcional	X			5
Precisión y exactitud computacional	Prueba de exactitud		X		3
Coherencia	Prueba de exactitud		X		3
				Total	16

Fiabilidad 25pts					
Atributo	Métrica	Cumple 5pts	Medianamente Cumple 3pts	No Cumple 1pts	Total
Presencia de errores	Densidad de fallas contra casos de prueba	X			5
Prevención de falla contra errores internos	Validación de entradas	X			5
Prevención de falla contra errores internos	Prevención de fallas contra errores internos		X		3
Atributo	Métrica	Cumple 10pts	Medianamente Cumple 6pts	No Cumple 2pts	Total
Continuidad de operación	Capacidad de continuidad en la operación	X			10
				Total	23

Eficiencia de Desempeño 20pts					
Atributo	Métrica	Cumple 8pts	Medianamente Cumple 5pts	No Cumple 2pts	Total
Capacidad de responder a las necesidades en un	Tiempo de respuesta	X			8

tiempo de retorno aceptable					
Atributo	Métrica	Cumple 4pts	Medianamente Cumple 2pts	No Cumple 1pts	Total
Capacidad de optimizar el uso de memoria	Prueba de optimización de uso de memoria	X			4
Capacidad para manejar concurrencia	Capacidad de manejo de concurrencia	X			4
Capacidad para optimizar las comunicaciones	Capacidad de optimizar comunicaciones y uso de dispositivos I/O	X			4
				Total	20

Seguridad 25pts					
Atributo	Métrica	Cumple 7.5pts	Medianamente Cumple 4.5pts	No Cumple 1.5pts	Total
Capacidad de control acceso a información privada sin permisos	Capacidad de control de acceso a información sin permisos	X			7.5
Seguridad de contraseña	Coherencia de contraseñas entre interfaces	X			7.5
Atributo	Métrica	Cumple 10pts	Medianamente Cumple 6pts	No Cumple 2pts	Total
Integridad del sistema	Pruebas de asignación de roles y privilegios de usuario		X		6
				Total	21

Usabilidad 10pts					
Atributo	Métrica	Cumple 1.5pts	Medianamente Cumple 1pts	No Cumple 0.5pts	Total
Capacidad para proveer entradas y salidas entendibles	Porcentaje de capacidad para proveer entradas y salidas entendibles	X			1.5
Capacidad para brindar claridad al usuario	Porcentaje de capacidad para brindad claridad al usuario	X			1.5
Atributo	Métrica	Cumple 1pts	Medianamente Cumple 0.6pts	No Cumple 0.2pts	Total
Capacidad para ser aprendido fácilmente por el usuario	Porcentaje de capacidad para ser aprendido fácilmente por el usuario	X			1
Documentación adecuada	Porcentaje de documentación adecuada	X			1

Atributo	Métrica	Cumple 2pts	Medianamente Cumple 1.5pts	No Cumple 1pts	Total
Presencia de entradas válidas	Verificación de entradas válidas.	X			2
Atributo	Métrica	Cumple 0.75pts	Medianamente Cumple 0.5pts	No Cumple 0.25pts	Total
Capacidad para ser operado y recordado por el usuario con facilidad	Porcentaje de la capacidad para ser operado y recordado por el usuario con facilidad.	X			0.75
Capacidad para reducir procedimientos de operación	Porcentaje de reducción del procedimiento de operación	X			0.75
Presencia de valores por defecto	Porcentaje de presencia de valores por defecto		X		0.5
Presencia de mensajes claros de usuario	Porcentaje de presencia de mensajes claros de usuario.	X			0.75
				Total	9.75
Suma Total del Resultado de Evaluación:	89.75/100				

Anexo T. Aplicación del Formulario de Pruebas Módulo de Registro y Actualización

Tabla T.1 Aplicación del Formulario de Pruebas Módulo de Registro y Actualización

Formulario de Pruebas						
Fecha	29-07-2024					
Nombre del Proyecto	SISGERH-PASANTIAS					
Cod. Caso de Prueba	CP001					
Módulo Puesto a Prueba	Registro de estudiantes					
Descripción	Registro de nuevos estudiantes Actualización de información de los estudiantes					
Responsable a Cargo	Ing. Diego Cadme, Bryan Jurado					
Requerimientos para la Prueba	Registrar un nuevo estudiante Actualizar la información del estudiante					
Nº Actividad	Descripción	Datos Prueba	Resultado Esperado	Resultado Obtenido	Estado (cumple/no cumple)	Observación
1	Registro de la información de un nuevo estudiante.	Se debe registrar la información del estudiante en la base de datos satisfactoriamente.	Registro de un nuevo estudiante.	Se registró la información del nuevo estudiante.	Cumple	Revisión de controles para el registro de la información.
2	Actualización de la información del pasante.	Actualización de la información del estudiante.	Actualización de la información del estudiante.	La información se actualizó correctamente.	Cumple	
Resultados de la Prueba						
Defectos y desviaciones encontradas						
No se evidenciaron defectos de corrección.						

Anexo U. Aplicación del Formulario de Pruebas Módulo de Reportes

Tabla U.1. Aplicación del Formulario de Pruebas Módulo de Reportes

Formulario de Pruebas						
Fecha	29-07-2024					
Nombre del Proyecto	SISGERH-PASANTIAS					
Cod. Caso de Prueba	CP001					
Módulo Puesto a Prueba	Obtención de reportes					
Descripción	Reporte de Consolidad de Zona y Mes Reporte de Reconocimiento Económico por Pasantías					
Responsable a Cargo	Ing. Diego Cadme, Bryan Jurado					
Requerimientos para la Prueba	Obtención del Reporte de Consolidad de Zona y Mes Obtención de Reporte de Reconocimiento Económico por Pasantías					
Nº Actividad	Descripción	Datos Prueba	Resultado Esperado	Resultado Obtenido	Estado (cumple/no cumple)	Observación
1	Obtener el Reporte de Consolidad de Zona y Mes	Se debe obtener la información del pago a pasantes por zona y mes.	Obtener el reporte de consolidado por zona y mes.	Se obtuvo el reporte de consolidado por zona y mes.	Cumple	Revisión de formato en el reporte.
2	Obtener el Reporte de Reconocimiento Económico por Pasantías	Se debe obtener la información del pago total a pasantes.	Obtener el reporte de reconocimiento económico.	Se obtuvo el reporte de reconocimiento económico por pasantías.	Cumple	
Resultados de la Prueba						
Defectos y desviaciones encontradas						
No se evidenciaron defectos de corrección.						