

-- PLAN DE CALIDAD --

Sistema de Software para la gestión de Procesos de Ingeniería de Requerimientos

Presentado por:

Dario Andres Ramos

Juan Vergara

Jorge Daniel Rincones F.

Joan Jiménez

Universidad Popular del Cesar.

Valledupar – Cesar

2023

Historial de Revisiones

Fecha	Versión	Descripción	Autor
18-Sept-23	1.0	Creación del documento	Jorge Daniel Rincones, Dario Andres Ramos,Juan Vergara, Joan Jiménez
20-Sept-23	1.1	Revisión de información adicional.	Jorge Daniel Rincones, Dario Andres Ramos,Juan Vergara, Joan Jiménez
21-Sept-23	1.2	Gestión de planificación.	Jorge Daniel Rincones, Dario Andres Ramos,Juan Vergara, Joan Jiménez

ÍNDICE

1. INTRODUCCIÓN	4
1.1 <i>Propósito</i>	4
1.2 <i>Alcance</i>	4
1.3 <i>Glosario de Términos</i>	5
1.4 <i>Acrónimos</i>	6
2. REQUERIMIENTOS	7
2.1 <i>Requerimientos funcionales detallados</i>	9
2.2 <i>Requerimientos no funcionales detallados</i>	11
3. MODELO DE DESARROLLO	13
3.1 <i>Actividades del proceso de desarrollo</i>	14
3.2 <i>Productos de Trabajo</i>	15
3.2.1 <i>Definición de los atributos de calidad</i>	18
3.2.2 <i>Puntos de revisión</i>	19
4. GESTION	21
4.1 <i>Organización y Responsabilidades</i>	21
4.2 <i>Riesgos</i>	23
4.2.1 <i>Riesgos del producto</i>	23
4.2.2 <i>Riesgo del proyecto</i>	23
4.2.3 <i>Evaluación de los riesgos</i>	24
5. GESTIÓN DE CALIDAD	25
5.1 <i>Objetivo de calidad</i>	25
5.2 <i>Procesos de calidad</i>	25
5.3 <i>Medición y métricas de calidad</i>	26
5.4 <i>Plan de prueba</i>	26
5.5 <i>Criterios generales de calidad</i>	28
5.6 <i>Seguimiento y mejora continua</i>	29
6. APROBACIÓN	30
6.1 <i>Planificación</i>	30
7. BIBLIOGRAFÍA	31
8. ANEXOS	32
10.1 <i>Plan de Proyecto</i>	32
10.2 <i>Plan de Riesgos</i>	33
10.3 <i>Especificación de Requerimientos</i>	34
10.4 <i>Especificación del Sistema (Solución Propuesta)</i>	35
10.5 <i>Especificación Funcional</i>	36
10.6 <i>Plan de Pruebas</i>	37
10.7 <i>Especificación de Diseño de Sistema</i>	38
10.8 <i>Especificación de diseño de soporte</i>	39
10.10 <i>Informe de Pruebas (Testing)</i>	40
10.11 <i>Manual de Usuario</i>	42

1. INTRODUCCIÓN

La creciente demanda de la industria de software por soluciones eficientes para la ingeniería de requerimientos ha generado la necesidad de desarrollar un sistema de software innovador y de alta calidad. El proyecto tiene como objetivo principal crear una herramienta que permita automatizar los procesos de ingeniería de requerimientos y optimizar la gestión de requisitos, contribuyendo así a la mejora de la calidad y la eficiencia en el desarrollo de software.

1.1. Propósito

El propósito de este plan de calidad es definir las estrategias y actividades relacionadas con el aseguramiento de la calidad a lo largo de todo el ciclo de vida del proyecto. Estas actividades tienen como objetivo garantizar que el sistema de software desarrollado cumpla con los estándares de calidad establecidos y satisfaga las necesidades del cliente y los usuarios finales. Además, entregar guías para la ejecución de las actividades de SQA, definir los estándares, los procedimientos y las convenciones que serán utilizados durante estas actividades y establecer las herramientas, técnicas y metodologías que soportan las prácticas de SQA.

Por lo tanto, el plan de SQA está dirigido al jefe de proyecto, los desarrolladores y al grupo de SQA, responsable de la elaboración, actualización y monitoreo del plan.

1.2. Alcance

Este plan de calidad abarca todas las fases del ciclo de vida del proyecto, desde la planificación y especificación de requerimientos hasta la implementación, pruebas, entrega y operación del sistema de software. Se aplica a todos los miembros del equipo de desarrollo y a las partes interesadas involucradas en el proyecto.

El objetivo del Plan de Calidad es comunicar el ámbito, recursos, y herramientas a los gestores del software y personal técnico, además de entregar a la administración una visibilidad adecuada del proceso utilizado y los productos construidos durante el proyecto mediante acciones planificadas y sistemáticas que aseguren la calidad de los procesos y productos.

1.3. Glosario de Términos

Para lograr un mejor entendimiento de los términos técnicos que se utilizan en el presente Plan de SQA, se mencionan a continuación los significados de los siguientes términos:

- **Aseguramiento de la Calidad del Software (SQA)** – El propósito de SQA es entregar a la administración una visibilidad adecuada del proceso utilizado y los productos contruidos mediante acciones planificadas y sistemáticas que aseguren la calidad de dichos procesos y productos.
- **Auditoría** – Evaluación independiente de los productos de trabajo y de un conjunto de procesos de software para asegurar la adherencia con las especificaciones, los estándares, procedimientos y otros acuerdos.
- **Gestión de la Configuración del Software (SCM)**– El propósito de SCM es establecer y mantener la integridad de los productos a través de todo el ciclo de vida del software, para así proveer un adecuado control de los cambios en los diversos ítems de configuración.
- **Revisión** – Metodología definida, estructurada y disciplinada para la detección e identificación de defectos en los productos de trabajo durante el ciclo de vida del software.
- **Prueba (*Testing*)** – Actividad que evalúa los atributos y la capacidad de un programa o sistema para determinar si se cumple con los resultados definidos.
- **Requisitos:** Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal.
- **Requerimientos:** Se presenta a continuación la definición existente en el glosario de la IEEE de lo que es un “Requerimiento”:

1. “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”. (Std 610.12-1900, IEEE: 62)

2. “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”. (Std 610.12-1900, IEEE: 62) También, Ian Sommerville presenta una definición acerca de lo que es un “Requerimiento”:

3. “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste”. (Sommerville, 2005: 108)

- **Sistema de Información:** Es un conjunto ordenado de recursos económicos, humanos, técnicos, datos y procedimientos, que interactúan entre sí y al ser ejecutados

apropiadamente, proporcionan la información requerida para apoyar la toma de decisiones y facilitar el control en la organización.

- **Norma ISO/IEC/IEEE 29148:** Esta norma establece los procesos y actividades para la ingeniería de requerimientos en proyectos de software y sistemas. Define los conceptos, términos y definiciones relacionados con la ingeniería de requerimientos, y establece los requisitos para la documentación y gestión de los requerimientos.
- **Norma ISO/IEC 12207:** Esta norma proporciona una guía para la gestión del ciclo de vida del software, incluyendo los procesos de ingeniería de requerimientos. Proporciona un marco de trabajo para el desarrollo de software y sistemas, y establece los procesos y actividades necesarias para la ingeniería de requerimientos en diferentes fases del ciclo de vida del proyecto.
- **Norma ISO/IEC 15288:** Esta norma proporciona una guía para la gestión del ciclo de vida de los sistemas, incluyendo los procesos de ingeniería de requerimientos. Proporciona un marco de trabajo para el desarrollo de sistemas, y establece los procesos y actividades necesarias para la ingeniería de requerimientos en diferentes fases del ciclo de vida del proyecto.
- **Estándar IEEE 830:** Este estándar establece los requisitos para la documentación de los requerimientos de software, incluyendo la estructura y el contenido de los documentos de especificación de requerimientos. Proporciona pautas para la definición de los requerimientos, y establece la terminología y el formato para la documentación de los mismos.
- **ISO/IEC 24766:** Tiene como objetivo proporcionar un marco de trabajo para la evaluación de la calidad de los requisitos de software, con el fin de mejorar la calidad del software y reducir los costos y riesgos asociados con el desarrollo de software defectuoso. La norma establece una serie de requisitos que los requisitos de software deben cumplir para ser considerados de alta calidad.
- **ISO 1233:** El objetivo de la norma es proporcionar una terminología común y estandarizada para la ingeniería de software y sistemas. La norma establece una serie de términos y definiciones que se utilizan en la ingeniería de software y sistemas, con el fin de mejorar la comunicación entre los diferentes actores que participan en el desarrollo de software y sistemas.

1.4. Acrónimos

Acrónimo	Significado
SQA	<i>Software Quality Assurance</i> , Aseguramiento de la Calidad del Software
SCM	<i>Software Configuration Management</i> , Gestión de Configuración del Software
WBS	<i>Work Breakdown Structure</i>

2. REQUERIMIENTO Y FUNCIONALIDADES

Anexo Plan de proyecto: [Plan de Proyecto](#)

Un requerimiento es un aspecto del producto requerido o deseado por el cliente. Los Requerimientos Funcionales cubren las funciones y operaciones a realizar para proporcionar un sistema que operará de acuerdo a las necesidades del usuario. Al elaborar una lista completa de las percepciones de los usuarios respecto a sus requerimientos, se definen las funciones que tendrán que ser realizadas por el sistema a desarrollar. En cambio, un Requerimiento No Funcional indica cómo se deben hacer todas las actividades de desarrollo para obtener un producto con la mayor calidad posible ya que ésta puede hacer la diferencia entre el éxito o fracaso de una aplicación.

Requerimiento Épicos	Prioridad
El aplicativo debe permitir al analista de negocios permitir la gestión de los requerimientos, así como, el registro, consultas, modificación y eliminación de los mismos.	9
El aplicativo debe permitir al gerente de proyectos y analista de negocio, la gestión de proyectos (fec, fina, desc,nombre), así como, el registro, consultas, modificación y eliminación de los mismos.	10
El aplicativo debe permitir al analista de negocio y cliente realizar la gestión de la documentación del proceso de ingeniería de requerimientos, debe permitir la generación de documentos.	8
El aplicativo debe permitir al gerente la gestión de usuarios, así como, el registro, consultas, modificación, eliminación y asignación de roles de los mismos.	7

El aplicativo debe permitir el trabajo colaborativo entre los miembros del equipo de desarrollo, gerente, analista de negocio y cliente mediante comentarios o notas dentro de los requerimientos, permitiendo la revisión y aprobación de los requerimientos.	6
El aplicativo debe permitir al equipo de desarrollo y analista de negocio el control de versiones de los requerimientos, de manera que se puedan describir y rastrear los cambios realizados y quién los realizó.	5
El aplicativo debe permitir al analista de negocio, cliente y equipo de desarrollo la priorización de los requerimientos (escala definida de 1 a 5) en función de su importancia y de los objetivos del proyecto.	4
El aplicativo debe permitir al equipo de desarrollo, gerente, analista de negocio y cliente el seguimiento de los requerimientos, permitiendo conocer su estado actual y su progreso en el tiempo.	3
El aplicativo debe permitir al gerente de proyecto la generación de informes y documentar el análisis sobre los requerimientos, permitiendo conocer el estado del proyecto y su evolución en el tiempo.	2

2.1. Requerimientos funcionales detallados:

Requerimiento Épico 1: Gestión de Requerimientos

Requerimientos Detallados:

- El sistema debe permitir a los analistas de negocios crear nuevos requisitos especificando detalles como nombre, descripción y prioridad.
- Los analistas de negocios deben poder consultar y buscar requisitos de manera eficiente utilizando filtros y palabras clave.
- Los requisitos deben poder modificarse y actualizarse con un registro de cambios.
- Se debe implementar una papelera de reciclaje para recuperar requisitos eliminados.

Requerimiento Épico 2: Gestión de Proyectos

Requerimientos Detallados:

- El sistema debe permitir a los gerentes de proyectos crear proyectos con información detallada como fechas de inicio y finalización, descripción y nombre.
- Los gerentes de proyectos deben poder consultar y buscar proyectos en función de diferentes criterios.
- Se debe permitir la modificación y actualización de detalles del proyecto, como fechas y descripción.
- Se debe implementar la posibilidad de eliminar proyectos, con la opción de archivarlos para referencia futura.

Requerimiento Épico 3: Gestión de Documentación

Requerimientos Detallados:

- Los usuarios deben poder crear documentos de ingeniería de requerimientos y agregar requisitos a ellos.
- Los documentos deben poder exportarse en diferentes formatos, como PDF o Word.
- Se debe permitir la revisión y edición colaborativa de documentos entre analistas de negocios y clientes.
- Los documentos deben poder vincularse a proyectos específicos para una gestión organizada.

Requerimiento Épico 4: Gestión de Usuarios

Requerimientos Detallados:

- El sistema debe permitir el registro de usuarios con información personal y de contacto.
- Debe ser posible asignar roles y permisos específicos a cada usuario.
- Los usuarios deben poder buscar y consultar la información de otros usuarios.

- Se debe permitir la modificación y actualización de datos de usuario.
- Debe ser posible eliminar usuarios y reasignar sus responsabilidades si es necesario.

Requerimiento Épico 5: Colaboración y Comunicación

Requerimientos Detallados:

- Se deben habilitar comentarios y notas asociados a los requisitos para facilitar la colaboración entre el equipo de desarrollo, el gerente, el analista de negocios y el cliente.
- Debe existir un sistema de revisión y aprobación de requisitos mediante interacciones en tiempo real.
- Se deben proporcionar herramientas de comunicación integradas, como notificaciones y mensajes directos.

Requerimiento Épico 6: Control de Versiones

Requerimientos Detallados:

- El sistema debe permitir el registro y seguimiento de cambios en los requisitos.
- Debe registrarse una descripción detallada de los cambios realizados y quién los realizó.
- Debe existir un historial de versiones para rastrear la evolución de los requisitos.

Requerimiento Épico 7: Priorización de Requisitos

Requerimientos Detallados:

- Los usuarios deben poder clasificar los requisitos en función de su importancia utilizando una escala definida de 1 a 5.
- Debe ser posible visualizar la prioridad de los requisitos de manera clara y sencilla.

Requerimiento Épico 8: Seguimiento de Requisitos

Requerimientos Detallados:

- El sistema debe permitir el seguimiento del estado actual y el progreso de los requisitos a lo largo del tiempo.
- Debe proporcionar información detallada sobre el estado de los requisitos y su cumplimiento.

Requerimiento Épico 9: Generación de Informes

Requerimientos Detallados:

- Debe ser posible generar informes que incluyan un resumen ejecutivo, estado de los requisitos, progreso, riesgos, incidencias y recomendaciones.
- Se deben proporcionar gráficos visuales para ilustrar el estado y progreso de los requisitos y análisis.

Requerimiento Épico 10: Auditoría de Métricas e Informes

Requerimientos Detallados:

- Se debe llevar a cabo una evaluación de métricas de calidad del software.
- Debe existir un proceso de auditoría de informes y documentación sobre el análisis de requisitos.

2.2. Requerimientos no funcionales:

Requerimiento Épico 1: Gestión de Requerimientos

Requerimientos No Funcionales Detallados:

- Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de usar para que los analistas de negocios puedan realizar las tareas de gestión de requerimientos de manera eficiente.
- Desempeño: El sistema debe ser capaz de gestionar grandes volúmenes de requerimientos sin retrasos significativos.
- Disponibilidad: El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad planificado mínimo.
- Seguridad: Los datos de requerimientos deben estar protegidos mediante medidas de seguridad, como autenticación y cifrado.

Requerimiento Épico 2: Gestión de Proyectos

Requerimientos No Funcionales Detallados:

- Desempeño: El sistema debe admitir múltiples proyectos concurrentes sin degradación del rendimiento.
- Escalabilidad: Debe ser posible ampliar la capacidad del sistema para manejar un número creciente de proyectos.
- Seguridad: La información de los proyectos debe estar protegida contra accesos no autorizados o filtraciones.

Requerimiento Épico 3: Gestión de Documentación

Requerimientos No Funcionales Detallados:

- **Compatibilidad:** El sistema debe ser compatible con múltiples formatos de documentos, como PDF y Word.
- **Rendimiento:** La generación de documentos debe ser rápida y eficiente.
- **Colaboración en tiempo real:** Varios usuarios deben poder editar un documento simultáneamente.

Requerimiento Épico 4: Gestión de Usuarios

Requerimientos No Funcionales Detallados:

- **Seguridad:** Los datos de usuario deben estar protegidos mediante autenticación y autorización adecuadas.
- **Escalabilidad:** El sistema debe ser escalable para acomodar un crecimiento en el número de usuarios registrados.
- **Rendimiento:** Las operaciones relacionadas con usuarios, como la asignación de roles, deben realizarse de manera eficiente.

Requerimiento Épico 5: Colaboración y Comunicación

Requerimientos No Funcionales Detallados:

- **Comunicación en tiempo real:** Las notificaciones y mensajes deben entregarse en tiempo real.
- **Escalabilidad:** El sistema debe ser escalable para soportar un alto volumen de interacciones en tiempo real.

Requerimiento Épico 6: Control de Versiones

Requerimientos No Funcionales Detallados:

- **Registro de Cambios:** Deben registrarse todos los cambios en los requisitos de manera clara y completa.
- **Auditoría:** Se debe llevar un registro de auditoría detallado de las modificaciones realizadas.

Requerimiento Épico 7: Priorización de Requisitos

Requerimientos No Funcionales Detallados:

- **Eficiencia en la Priorización:** La plataforma debe permitir una priorización rápida y eficiente de los requisitos por parte de los usuarios.

Requerimiento Épico 8: Seguimiento de Requisitos

Requerimientos No Funcionales Detallados:

- **Actualización en Tiempo Real:** Los cambios en el estado y el progreso de los requisitos deben reflejarse en tiempo real.

Requerimiento Épico 9: Generación de Informes

Requerimientos No Funcionales Detallados:

- **Generación Rápida:** La generación de informes debe ser rápida y no debe causar retrasos significativos en el sistema.
- **Escalabilidad de Informes:** El sistema debe manejar informes cada vez más complejos a medida que crece la cantidad de datos.

Requerimiento Épico 10: Auditoría de Métricas e Informes

Requerimientos No Funcionales Detallados:

- **Precisión de Métricas:** Las métricas y los informes generados deben ser precisos y confiables.
- **Integridad de la Auditoría:** Los registros de auditoría deben ser inmutables y resistentes a la manipulación.

2.3 FUNCIONALIDAD

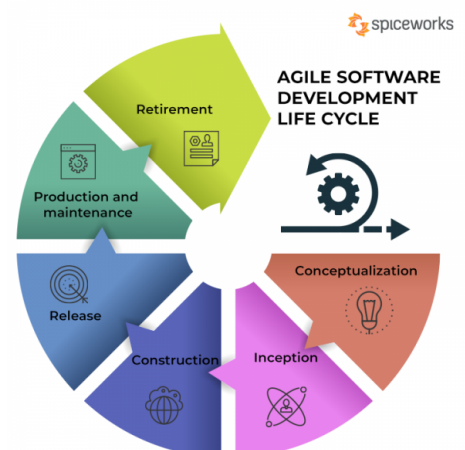
- **Gestión de Requerimientos:** Esta función permite la creación, consulta, modificación y eliminación de requerimientos. Para calcular los Puntos de Función, debes identificar las transacciones específicas dentro de esta función, como "Crear Requisito," "Consultar Requisito," "Modificar Requisito," y "Eliminar Requisito."
- **Gestión de Proyectos:** Esta función incluye la creación, consulta, modificación y eliminación de proyectos. Identifica las transacciones relacionadas, como "Crear Proyecto," "Consultar Proyecto," "Modificar Proyecto," y "Eliminar Proyecto."
- **Gestión de Documentación:** Esta función involucra la creación de documentos de ingeniería de requerimientos y la generación de documentos. Identifica las transacciones, como "Crear Documento," "Generar Documento," y "Editar Documento."

- **Gestión de Usuarios:** Esto implica la creación, consulta, modificación y eliminación de usuarios. Identifica transacciones como "Registrar Usuario," "Consultar Usuario," "Modificar Usuario," y "Eliminar Usuario."
- **Colaboración y Comunicación:** Esta función permite la comunicación en tiempo real y la colaboración en los requerimientos. Identifica transacciones como "Agregar Comentario," "Revisar Requisito," y "Aprobar Requisito."
- **Control de versiones:** Esta función se encarga del control de versiones de los requerimientos. Identifica transacciones como "Registrar Cambio," "Consultar Historial," y "Rastrear Cambios."
- **Priorización de Requisitos:** Esta función permite la priorización de requerimientos. Identifica transacciones como "Priorizar Requisito" y "Consultar Prioridades."
- **Seguimiento de Requisitos:** Esta función implica el seguimiento y la consulta de estado y progreso de los requerimientos. Identifica transacciones como "Consultar Estado" y "Consultar Progreso."
- **Generación de Informes:** Esta función permite generar informes sobre el estado y el progreso de los requerimientos. Identifica transacciones como "Generar Informe" y "Consultar Informe."
- **Auditoría de Métricas e Informes:** Esta función involucra la evaluación de métricas y la auditoría de informes. Identifica transacciones como "Evaluar Métricas" y "Auditar Informe."

3. MODELO DE DESARROLLO

Se optó por la estrategia de desarrollo “Modelo Incremental-ágil”, el cual aplica secuencias iterativa mientras progresa el tiempo en el calendario. Cada secuencia produce un “incremento” del software. La elección de la estrategia seleccionada se debe a que la intención es entregar el software en partes pequeñas, pero utilizables, llamadas “incrementos”, es decir, cada “incremento” se construye sobre aquél que ya ha sido entregado.

En resumen, el modelo de desarrollo ágil, con un marco de trabajo en particular Scrum, es una elección sólida para el proyecto de automatización de procesos de ingeniería de requerimientos debido a su flexibilidad, adaptabilidad y enfoque en la satisfacción del cliente. Permite a los equipos



gestionar los cambios de requisitos de manera eficiente y garantizar la entrega de un producto de alta calidad que cumpla con las necesidades del cliente.

3.1 Actividades del proceso de desarrollo

Para el modelo de desarrollo ágil, en particular utilizando la metodología Scrum, las fases se organizan en ciclos de desarrollo iterativos e incrementales llamados "Sprints". Cada Sprint tiene una duración fija, típicamente de 2 a 4 semanas, y pasa por las siguientes fases:

- **Planificación del Sprint:** En esta fase, el equipo de desarrollo se reúne para seleccionar un conjunto de elementos del Backlog del Producto (requisitos) que se abordarán durante el próximo Sprint. Estos elementos se transfieren al "Backlog del Sprint", que es el conjunto de tareas específicas que se llevarán a cabo en el Sprint.
- **Desarrollo y Codificación:** Durante el Sprint, el equipo trabaja en la implementación de las tareas asignadas. Se siguen las prácticas de desarrollo ágil, como la colaboración constante, la revisión de código y la automatización de pruebas.
- **Revisión y Demostración:** Al finalizar el Sprint, se realiza una reunión de revisión del Sprint en la que se muestran las funcionalidades completadas al cliente o al Product Owner. Esta revisión permite al cliente ver el progreso y proporcionar retroalimentación.
- **Retrospectiva del Sprint:** Después de la revisión del Sprint, se lleva a cabo una retrospectiva en la que el equipo analiza lo que funcionó bien y lo que se puede mejorar en el proceso. Se identifican oportunidades de mejora y se ajusta el enfoque para futuros Sprints.
- **Entrega de Incremento:** Al final de cada Sprint, se debe entregar un incremento del software que incluye las funcionalidades completadas y probadas durante ese Sprint. Esto garantiza que haya una versión funcional del software disponible después de cada iteración.
- **Actualización del Backlog del Producto:** Durante la retrospectiva y la revisión del Sprint, el Product Owner y el equipo pueden ajustar el Backlog del Producto en función de la retroalimentación del cliente y las necesidades cambiantes del proyecto.
- **Inicio del Siguiente Sprint:** Una vez completado un Sprint, se inicia un nuevo ciclo con una nueva planificación del Sprint. El proceso se repite hasta que se logre el producto final deseado.

Estas fases se repiten de manera iterativa e incremental a lo largo del proyecto, lo que permite una entrega temprana de funcionalidades y la adaptación a cambios en los requisitos. La flexibilidad y la retroalimentación constante son características clave de este enfoque ágil.

3.2 Productos de Trabajo

A continuación, se definen los productos de trabajo que deberían entregarse dentro del desarrollo del proyecto:

- **Plan de Proyecto**

Para contar con una forma de monitorear y controlar lo que se está llevando a cabo dentro del desarrollo del proyecto. El Plan de Proyecto proporciona un repositorio central que contiene la información de planificación e implementación requerida para ejecutar el plan de proyecto entero. Provee un resumen y una integración de todos los planes contenidos en el proyecto y de todos los proyectos contenidos en el programa. Posibilita el monitoreo del progreso del proyecto de manera consistente con el plan.

El Plan de Proyecto inicial refleja la información que está disponible en la fase de análisis. Al tener información más detallada, según avanza el proyecto, el plan de proyecto es actualizado de acuerdo con esto. La cobertura del Plan de Proyecto es el proyecto completo, pero en cualquier punto del tiempo, normalmente contendrá actividades detalladas solamente para la fase actual y la fase siguiente.

- **Plan de Riesgos**

El Plan de Riesgos establece las posibles situaciones en las que el proyecto podría verse afectado. Para ello, se realiza un Análisis de Riesgos, a través de la cual se establecerán las acciones a tomar en caso que se concreten dichas situaciones. Básicamente, el Plan de Riesgos debe contemplar la: definición de los riesgos, la posibilidad de que se concrete cada uno de los riesgos detectados, definir qué tan grande sería el impacto de cada riesgo identificado en el proyecto, definir e indicar cuales son los eventos indicadores de que el riesgo se ha concretado, y definir el plan de contingencia para cada uno de ellos (Ver página 36).

- **Especificación de Requerimientos**

La Especificación de Requerimientos proporciona un repositorio central que contiene la información actualizada de cada uno de los requerimientos detectados, como el número secuencial que identifica en forma única al requerimiento, tipo de requerimiento, identificación de requerimiento original (cuando es reclasificado), estado del requerimiento, persona que lo planteó, categoría del requerimiento, nombre del requerimiento, fecha en que se da por aprobado el requerimiento por parte del cliente, nombre de la persona que da por

aprobado el requerimiento. Todo esto, permitirá una administración correcta de los requerimientos del proyecto.

Por otra parte, la Especificación de Requerimientos refleja las necesidades de negocio, los requerimientos necesarios para resolver aquellas necesidades, y objetivos del cliente que buscan una solución. Este es el primer documento que se genera en un proyecto y debe ser conciso, claro y completo; de modo tal que sea la base para las siguientes etapas. Por otro lado, debe ser escrito en un lenguaje de alto nivel, en términos del negocio. Debiera ser capaz de ser producido en un tiempo limitado, y no debería contener detalles específicos a menos que sean considerados requisitos para el negocio. La Especificación de Requerimientos forma la base para el desarrollo de la Solución Propuesta, pero al mismo tiempo difieren, pues ésta refleja las necesidades y objetivos de negocio traducidos en requerimientos para una solución. La Solución Propuesta detalla especificaciones para una solución específica seleccionada a partir de un número de alternativas posibles.

- **Plan de pruebas**

El Plan de Pruebas es un documento estrechamente alineado con la Especificación Funcional, el cual describe las pruebas que serán llevadas a cabo para demostrar al cliente que la solución satisface los requerimientos y que la solución cumple con el uso que se pretende en el ambiente operacional del cliente (Verificación y Validación).

- **Especificación de Diseño de Sistema**

La Especificación de Diseño del Sistema define la Arquitectura de la Solución e identifica todos los componentes del sistema. Establece las interfaces y la especificación a alto nivel de cada componente. Para cada componente se establece: la descripción de la componente, diagrama de flujo de datos de la componente, interfaces de la componente, estructura de datos de la componente, diseño funcional de la componente, criterios de diseño y restricciones de la componente, especificación de test de la componente e integración, y ambiente de la componente.

Por otro lado, también especifica un nivel básico de implementación y los test de componentes e integración que se utilizan en la fase de implementación.

- **Plan de Aseguramiento de Calidad SQA**

El Plan de QA define todas las actividades de aseguramiento de calidad que se harán durante el proyecto. La importancia de este plan reside en contar con un documento formal con instrucciones explícitas acerca de la forma de llevar a cabo cada una de las actividades previamente planificadas

y de esta forma poder controlar cada una de las variables que inciden en el correcto desarrollo del producto.

El Proyecto, debe tener definidas ciertas actividades para cumplir con sus estándares de calidad, entre ellas realizar revisiones e inspecciones dentro del proyecto, llevar a cabo testing de los

módulos desarrollados, entre otras. Estas actividades, serán realizadas durante todo el proceso de desarrollo del software para asegurar que este cumpla con los criterios de calidad impuestos.

Algunas de las etapas a seguir son las de llevar a cabo controles sobre la documentación del software, códigos fuentes, manuales e informes de requerimientos, mantener toda la documentación con respecto a los cambios efectuados durante el desarrollo, llevar a cabo procedimientos que permitan asegurar los ajustes de los estándares de desarrollo de software, mecanismos para realizar mediciones de manejo de información e identificación de procesos, entre otros.

Por otra parte, el Plan de QA, entrega todos los procedimientos y estándares que se llevarán a cabo durante el desarrollo del proyecto, así como los formularios y checklist correspondientes. Se entrega junto con el Plan de Proyecto.

- **Informe de pruebas (testing)**

Los resultados de estas pruebas ayudarán a comprobar el “buen” funcionamiento del software una vez integrados sus componentes.

- **Manual de usuario**

Explica el comportamiento del sistema desde el punto de vista funcional de la aplicación. Para ello, se basa en el documento de Especificación Funcional (ver anexo Manual de Usuario).

3.2.1 Definición de los atributos de calidad

Con una perspectiva orientada hacia la Revisión del Producto se tienen los siguientes factores:

- | | |
|-------------------|----------------------------------|
| 1º. Mantenimiento | ¿Se pueden corregir los errores? |
| 2º. Flexibilidad | ¿Pueden cambiarlo? |
| 3º. Prueba | ¿Puede ser probado? |

Tomando una perspectiva desde la Operación del Producto se tienen los siguientes factores:

- | | |
|----------------------|---------------------------------------|
| 4º. Corrección | ¿Hace lo que se le pidió? |
| 5º. Confiabilidad | ¿Es confiable todo el tiempo? |
| 6º. Facilidad de uso | ¿Está diseñado para ser usado? |
| 7º. Integridad | ¿Es seguro (inviolable)? |
| 8º. Eficiencia | ¿Podrá ejecutarse lo mejor que pueda? |

Con una perspectiva de la Transición del Producto se tienen los siguientes factores:

- | | |
|------------------------|--------------------------------------|
| 9º. Portabilidad | ¿Podrá usarse en otra máquina? |
| 10º. Reusabilidad | ¿Podrá rehusar algo del software? |
| 11º. Interoperatividad | ¿Podrá interactuar con otro sistema? |

Otros atributos definidos son los siguientes:

- | | |
|------------------|--|
| 12°. Completitud | ¿Se encuentran todas las funciones y restricciones en el sistema? |
| 13°. Claridad | Son suficientemente claros los conceptos e ideas que se intentan expresar? |

3.2.2 Puntos de revisión (hitos)

Se identifican como puntos de revisión aquellos que permiten validar y controlar las tareas realizadas dentro de cada etapa del ciclo de desarrollo y por cada cambio producido en mantención. Debe ser utilizado por la unidad de SQA durante la planificación para verificar el correcto establecimiento de los hitos de calidad.

- **Planificación**
 - Revisión y aprobación del plan de SQA.
 - Revisión y aprobación del plan de proyecto.
 - Revisión y aprobación del plan de riesgos.
 - Reporte del estado y los resultados de las actividades de SQA.
- **Especificación de requerimientos (Definición)**
 - Revisión y aprobación de la especificación de requerimientos.
 - Reporte del estado y los resultados de las actividades de SQA
- **Análisis**
 - Revisión y Aprobación de la Especificación del Sistema (Solución Propuesta).
 - Reporte del estado y los resultados de las actividades de SQA.

· **Diseño**

- Revisión y aprobación de la Especificación de Diseño de Sistema.
- Revisión y aprobación de la Especificación Funcional del Sistema.
- Revisión y aprobación de la Especificación de Diseño de Soporte del Sistema.
- Revisión y aprobación del Plan de Pruebas del sistema
- Reporte del estado y los resultados de las actividades de SQA.

· **Implementación**

- Revisión y aprobación de los casos de prueba.
- Revisión y aprobación de la especificación de los procedimientos de prueba.
- Revisión y aprobación del código y su documentación.
- Revisión y aprobación de los resultados de la prueba de unidad, integración, y sistema
- Reporte del estado y los resultados de las actividades de SQA.
- Reporte del estado y los resultados de las actividades de SQA.
- Revisión y aprobación del Manual de Usuario.
- Revisión y Aprobación del Manual de Instalación del Sistema

· **Instalación (Aceptación y entrega)**

- Revisión y aprobación el software y su documentación.
- Reporte del estado y los resultados de las actividades de SQA.

· **Operación (Mantenición)**

- Revisión y aprobación de cada cambio producido durante la mantención en su especificación, diseño, implementación y prueba.
- Revisión y aprobación de la documentación asociada a los cambios.
- Revisión y aprobación de la nueva versión del software y de su documentación.
- Reporte del estado y los resultados de las actividades de SQA.

4. GESTION

4.1. Organización y Responsabilidades

La gestión del proyecto seguirá las siguientes estrategias y procedimientos organizacionales:

- **Asignación de Roles:** Cada miembro del equipo tendrá roles y responsabilidades claramente definidos, incluyendo un gerente de proyecto, analistas de negocios, desarrolladores y especialistas en calidad.
- **Comunicación Regular:** Se llevarán a cabo reuniones de seguimiento periódicas para revisar el progreso del proyecto, resolver problemas y garantizar una comunicación efectiva entre los miembros del equipo.
- **Documentación:** Se mantendrá una documentación completa y actualizada de los procesos, requisitos y decisiones del proyecto.

1.4 Números de Personas y Partes Involucradas

El proyecto involucra a las siguientes personas y partes interesadas:

- **Gerente del Proyecto** (1 persona): El gerente del proyecto es responsable de planificar, ejecutar y supervisar todas las actividades relacionadas con el proyecto. Esto incluye la gestión de recursos, la comunicación con las partes interesadas y el cumplimiento de los plazos y objetivos del proyecto.
- **Representantes del Cliente de la Industria de Software** (2 personas): Estos representantes del cliente son las personas que actúan como enlace entre tu equipo de desarrollo y la empresa cliente. Su papel implica comunicar las necesidades y expectativas del cliente, proporcionar retroalimentación durante el desarrollo y asegurarse de que el producto final cumpla con los requisitos del cliente.
- **Analista de Negocios:** desempeña un papel crucial en el proyecto al actuar como un puente entre los objetivos comerciales del cliente y el equipo de desarrollo. Sus responsabilidades incluyen la identificación y documentación de los requisitos comerciales, la realización de análisis de mercado, la definición de casos de uso y la creación de modelos de negocio.
- **Usuarios Finales** (varios, dependiendo del alcance del proyecto): Los usuarios finales son las personas que utilizarán el sistema una vez que esté completo. Su número puede variar según la escala del proyecto y la cantidad de usuarios previstos. Es importante identificar y comprender las necesidades y expectativas de los usuarios finales para diseñar un sistema que satisfaga sus requerimientos.

- **Equipo de Desarrollo** (varios miembros, dependiendo del alcance del proyecto): El equipo de desarrollo está compuesto por profesionales encargados de diseñar, programar y poner en marcha el sistema de software. Este equipo puede incluir roles como desarrolladores de software, diseñadores de interfaz de usuario, arquitectos de software y otros especialistas en desarrollo de software. El número de miembros del equipo dependerá de la complejidad y el alcance del proyecto.
- **Equipo de Analistas de Pruebas** (varios miembros, dependiendo del alcance del proyecto): El equipo de analistas de pruebas se encarga de asegurar la calidad del software desarrollado. Sus responsabilidades incluyen la creación de casos de prueba, la ejecución de pruebas, el informe de errores y problemas, y la verificación de que el software cumpla con los requisitos establecidos. El número de miembros de este equipo también dependerá de la magnitud del proyecto y la necesidad de pruebas exhaustivas.
- **Equipo de Arquitectura** (varios miembros, dependiendo de la complejidad del proyecto): El equipo de arquitectura se encarga de diseñar la estructura técnica del sistema de software. Sus responsabilidades incluyen la definición de la arquitectura del software, la selección de tecnologías y herramientas adecuadas, la creación de diagramas y planos arquitectónicos, y la supervisión de la implementación para garantizar la coherencia y la escalabilidad del sistema. Este equipo puede incluir arquitectos de software, ingenieros de sistemas y otros expertos técnicos.
- **Equipo Administrativo** (varios miembros): El equipo administrativo proporciona el soporte administrativo necesario para el proyecto. Este equipo puede incluir administradores de proyectos, asistentes administrativos y personal de recursos humanos, entre otros. Su función es gestionar la documentación, la logística, las comunicaciones internas y la coordinación general del proyecto.
- **Manager de Entregas** (1 persona): El Manager de Entregas es responsable de supervisar el proceso de entrega del proyecto. Sus tareas incluyen la planificación de entregas, la gestión de recursos y la garantía de que los hitos y los plazos se cumplan de manera eficiente. Trabajando en colaboración con el gerente del proyecto, el Manager de Entregas se asegura de que los entregables se entreguen a tiempo y cumplan con los estándares de calidad.

4.2. Riesgo del Proyecto

Los riesgos potenciales del proyecto incluyen:

4.2.1. Riesgos del Producto:

- **Seguridad de Datos:** Existe un riesgo de que los datos personales de los usuarios se vean comprometidos si no se implementan medidas de seguridad adecuadas.
- **Compatibilidad de Dispositivos:** Puede haber desafíos en la garantía de que la aplicación funcione correctamente en una variedad de dispositivos móviles y sistemas operativos.
- **Estabilidad de la Aplicación:** Riesgo de que la aplicación tenga fallas o se bloquee en ciertos escenarios, lo que afectaría la experiencia del usuario.
- **Integración con Servicios Externos:** Si la aplicación depende de servicios externos (por ejemplo, servicios de autenticación), hay un riesgo de que problemas en estos servicios afectan la funcionalidad de la aplicación.
- **Cumplimiento Normativo:** Si la aplicación maneja datos sensibles o está relacionada con la educación, podría haber riesgos legales y de cumplimiento normativo si no se siguen las regulaciones adecuadas.

4.2.2. Riesgos del Proyecto:

- **Retrasos en el desarrollo:** Los retrasos en la entrega de componentes o funcionalidades del proyecto pueden impactar el tiempo disponible para las pruebas.
- **Cambios en los requisitos:** Los cambios frecuentes en los requisitos del proyecto pueden dificultar la planificación y ejecución de las pruebas.
- **Recursos Insuficientes:** Si el equipo de desarrollo o pruebas no cuenta con recursos suficientes, podría haber retrasos o una calidad de prueba inferior.
- **Comunicación Ineficaz:** La falta de comunicación efectiva entre los miembros del equipo y los stakeholders podría llevar a malentendidos y problemas en la ejecución de las pruebas.
- **Cambios en el Alcance:** Cambios significativos en el alcance del proyecto pueden llevar a la necesidad de ajustar la planificación de pruebas.
- **Requisitos Incompletos o Ambiguos:** La presencia de requisitos incompletos o ambiguos puede dar lugar a pruebas incompletas o ineficaces.

4.2.3. Evaluación de riesgos:

Escala: 1-5, donde 1 es bajo y 5 es muy alto

Nivel de riesgo = Impacto x Probabilidad

Riesgos del Producto				
RIESGO	IMPACTO	PROBABILIDAD	RIESGO CALCULADO	PLAN DE ACCIÓN
Seguridad de Datos	4	2	8	Integración con seguridad interna.
Compatibilidad de Dispositivos	2	4	8	
Estabilidad de la Aplicación	2	2	4	Plan reacción a errores.
Integración con Servicios Externos	4	1	4	Plan reacción a caída de servicios externos.

Riesgos del Proyecto				
RIESGO	IMPACTO	PROBABILIDAD	RIESGO CALCULADO	PLAN DE ACCIÓN
Retrasos en el Desarrollo	4	2	8	Reunión de revisión de plan de trabajo y recursos en equipo de desarrollo.
Cambios en los Requisitos	4	1	4	Revisión de planeación y alcance, integración de cambios.
Recursos Insuficientes	4	1	4	Revisión presupuestal. Revisión planeación.
Comunicación Ineficaz	1	1	1	Reuniones de comunicación asertiva.
Cambios en el Alcance	4	1	4	Revisión de planeación y alcance, integración de cambios.
Requisitos Incompletos o Ambiguos	4	1	4	Revisión de planeación y alcance, integración de cambios.

3. GESTIÓN DE CALIDAD.

5.1. Objetivos de Calidad

Los objetivos específicos de calidad que se deben lograr con el sistema de software son los siguientes:

- Garantizar la captura precisa y completa de requerimientos.
- Asegurar la trazabilidad y rastreabilidad de los requisitos a lo largo del ciclo de vida del proyecto.
- Facilitar la colaboración entre los miembros del equipo y las partes interesadas mediante una plataforma centralizada y herramientas de comunicación.
- Minimizar errores y defectos en el proceso de ingeniería de requerimientos mediante revisiones, validaciones y verificaciones periódicas.
- Cumplir con los estándares y mejores prácticas de la industria de desarrollo de software, incluyendo la normativa ISO/IEC 29148:2018 sobre ingeniería de requisitos.

5.2. Procesos de Calidad

Los procesos y actividades que se llevarán a cabo para garantizar la calidad del proyecto incluyen:

- Revisión continua de los requisitos para garantizar su precisión y completitud.
- Verificación y validación de requisitos mediante técnicas como la revisión por pares y la verificación de consistencia.
- Uso de herramientas de control de versiones para rastrear cambios en los requisitos.
- Pruebas exhaustivas para asegurar que el software cumple con los requisitos y está libre de defectos.
- Gestión de cambios controlada para evaluar y aprobar cambios en los requisitos.
- Auditorías de métricas e informes para evaluar el progreso y la calidad del proyecto.
- Comunicación continua con el cliente y las partes interesadas para mantenerlos informados sobre el estado del proyecto.

5.3 Medición y Métricas de Calidad

Las métricas que se utilizarán para evaluar la calidad del sistema de software incluyen:

- Tasa de precisión de requisitos capturados.
- Trazabilidad de requisitos a través del ciclo de vida del proyecto.
- Número de errores y defectos identificados durante las pruebas.
- Cumplimiento de los criterios de aceptación establecidos.
- Satisfacción del cliente medida a través de encuestas y retroalimentación.
- Tiempo de respuesta del sistema y rendimiento en términos de eficiencia y velocidad de procesamiento.

5.4. Plan de Pruebas

Anexo Plan de pruebas: [Anexo Plan de Pruebas](#)

El plan de pruebas incluirá:

- Pruebas unitarias para validar componentes individuales del software.
- Pruebas de integración para asegurar que los módulos funcionen correctamente juntos.
- Pruebas de aceptación con la participación del cliente para confirmar que se cumplen los requisitos.
- Pruebas de rendimiento y carga para evaluar el rendimiento del sistema.
- Documentación detallada de casos de prueba, criterios de aceptación y resultados esperados.

Estrategia de plan de pruebas: Estrategia sugerida

Objetivo de la Estrategia de Pruebas: Garantizar que la aplicación cumple con los requisitos funcionales especificados y que se entrega sin defectos críticos.

Alcance de las Pruebas: Las pruebas se centrarán en las funciones y características relacionadas anteriormente, estas se basarán dependiendo el marco de trabajo empleado.

Tipos de Pruebas:

1. **Pruebas de Funcionalidad:** Verificar que todas las funciones y características específicas funcionen según los criterios de aceptación.
2. **Pruebas de Seguridad:** Evaluar la seguridad de los datos y la protección de la información.
3. **Pruebas de Usabilidad:** Evaluar la facilidad de uso de la aplicación desde la perspectiva del usuario final.

4. **Pruebas de Performance:** Evaluar el rendimiento de la aplicación, especialmente en lo que respecta al tiempo de respuesta y la capacidad de manejar múltiples usuarios concurrentes.

Etapas de las Pruebas:

1. **Pruebas de Unidad (Realizadas por los desarrolladores):** Asegurarse de que cada componente individual de la aplicación funcione correctamente antes de la integración.
2. **Pruebas de Integración:** Probar la interacción entre diferentes componentes y módulos de la aplicación.
3. **Pruebas de Sistema:** Evaluar la funcionalidad del sistema en su conjunto.
4. **Pruebas de Aceptación del Usuario (UAT):** Trabajar con los stakeholders para validar que la aplicación cumple con sus necesidades y expectativas.

Entorno de Pruebas:

- Se utilizará un entorno de pruebas separado del entorno de producción para evitar la interferencia con usuarios reales.

Criterios de Éxito:

- Todas las historias de usuario relacionadas deben aprobar las pruebas funcionales.
- Todas las historias de usuario relacionadas deben ser revisadas por las pruebas especializadas.
- No se deben encontrar defectos críticos que impidan el funcionamiento de las funciones esenciales del rol del docente.

Herramientas y Recursos:

- Se utilizarán herramientas de automatización de pruebas si es posible y beneficioso para acelerar el proceso de prueba. (IntelliJ aplicando screenplay, jmeter)
- El equipo de pruebas contará con dispositivos y sistemas operativos representativos para pruebas manuales.

Plan de Pruebas:

- Se creará un plan de pruebas detallado que incluya casos de prueba específicos para cada requisito.
- Se asignarán responsabilidades claras para la ejecución de pruebas y la documentación de resultados.

Plan de Contingencia:

- Se preparará un plan de contingencia para abordar cualquier defecto crítico que se encuentre durante las pruebas.
- Se establecerán procedimientos para la comunicación y resolución de problemas de manera eficiente.

5.5 Criterios Generales de Calidad

PLANEADA: La calidad planeada se compone de lo siguiente:	FUNCIONALIDAD: Toda la funcionalidad del sistema de software.	NECESARIA: La calidad necesaria se compone de lo siguiente:	FUNCIONALIDAD: Toda la funcionalidad necesaria para un MVP.
	REQUERIMIENTO: Todos los requerimientos del sistema de software.		REQUERIMIENTO: Toda la funcionalidad necesaria para un MVP.
Criterio aceptación	80% Funcional.	Criterio aceptación	100% Funcional.

Requerimiento Épico 1: Gestión de Requerimientos

- **Calidad Necesaria:** Creación de requisitos con nombre, descripción y prioridad. Consulta y búsqueda de requisitos.
- **Calidad Planeada:** Modificación con registro de cambios, implementación de papelera de reciclaje.

Requerimiento Épico 2: Gestión de Proyectos

- **Calidad Necesaria:** Creación de proyectos con detalles esenciales (fechas, descripción, nombre). Consulta y búsqueda de proyectos.
- **Calidad Planeada:** Modificación de detalles del proyecto, archivado de proyectos eliminados.

Requerimiento Épico 3: Gestión de Documentación

- **Calidad Necesaria:** Creación de documentos de ingeniería, vinculación de requisitos a documentos.
- **Calidad Planeada:** Exportación en diferentes formatos, revisión y edición colaborativa, vinculación con proyectos.

Requerimiento Épico 4: Gestión de Usuarios

- **Calidad Necesaria:** Registro de usuarios, asignación de roles y permisos, consulta de información de usuarios.
- **Calidad Planeada:** Modificación y actualización de datos, eliminación y reasignación de responsabilidades.

Requerimiento Épico 5: Colaboración y Comunicación

- **Calidad Necesaria:** Comentarios y notas en requisitos, revisión y aprobación de requisitos.
- **Calidad Planeada:** Herramientas de comunicación integradas.

Requerimiento Épico 6: Control de Versiones

- **Calidad Necesaria:** Registro de cambios en requisitos, descripción de cambios.
- **Calidad Planeada:** Historial de versiones, rastreo de evolución de requisitos.

Requerimiento Épico 7: Priorización de Requisitos

- **Calidad Necesaria:** Clasificación de requisitos en función de su importancia.
- **Calidad Planeada:** Visualización clara y sencilla de la prioridad.

Requerimiento Épico 8: Seguimiento de Requisitos

- **Calidad Necesaria:** Seguimiento del estado y progreso de requisitos.
- **Calidad Planeada:** Información detallada sobre el cumplimiento de los requisitos.

Requerimiento Épico 9: Generación de Informes

- **Calidad Necesaria:** Generación de informes con resumen ejecutivo, estado de requisitos y progreso.
- **Calidad Planeada:** Gráficos visuales, análisis de riesgos, incidencias y recomendaciones.

Requerimiento Épico 10: Auditoría de Métricas e Informes

- **Calidad Necesaria:** Evaluación de métricas de calidad del software.
- **Calidad Planeada:** Proceso de auditoría de informes y documentación

Para alcanzar una calidad necesaria para un MVP (Mínimo Producto Viable), podemos seleccionar los requerimientos esenciales que son críticos para el funcionamiento básico del sistema. En este caso, los requerimientos necesarios para un MVP serían los siguientes:

- El aplicativo debe permitir al analista de negocios permitir la gestión de los requerimientos, así como, el registro, consultas, modificación y eliminación de los mismos.
- El aplicativo debe permitir al gerente de proyectos y analista de negocio, la gestión de proyectos (fec, fina, desc, nombre), así como, el registro, consultas, modificación y eliminación de los mismos.
- El aplicativo debe permitir al analista de negocio y cliente realizar la gestión de la documentación del proceso de ingeniería de requerimientos, debe permitir la generación de documentos.
- El aplicativo debe permitir al gerente la gestión de usuarios, así como, el registro, consultas, modificación, eliminación y asignación de roles de los mismos.
- El aplicativo debe permitir el trabajo colaborativo entre los miembros del equipo de desarrollo, gerente, analista de negocio y cliente mediante comentarios o notas dentro de los requerimientos, permitiendo la revisión y aprobación de los requerimientos.

Estos cinco requerimientos son esenciales para la funcionalidad básica del sistema y permitirán crear un MVP funcional que permita a los usuarios realizar la gestión de requerimientos y proyectos, así como colaborar en la documentación y revisión de requerimientos de manera efectiva. A partir de este MVP, se pueden agregar características adicionales en versiones posteriores del sistema para alcanzar la calidad planificada completa del proyecto.

5.6. Seguimiento y Mejora Continua

La calidad se monitorea de manera continua a lo largo del proyecto mediante revisiones regulares, pruebas y auditorías. Se llevarán a cabo reuniones de retroalimentación con el cliente y se documentará lecciones aprendidas. Las mejoras continuas se implementarán según sea necesario.

6. APROBACIÓN

Este plan de calidad será revisado y aprobado por los stakeholders y miembros clave del equipo antes de su implementación.

Con este plan de calidad completo, se establecen las bases para garantizar que el proyecto de desarrollo del sistema de software para la automatización de procesos de ingeniería de requerimientos cumpla con los estándares de calidad y los objetivos establecidos.

JORGE D. RINCONES

JUAN VERGARA

DARIO A. RAMOS

JOAN JIMENEZ

7 BIBLIOGRAFÍA

Norma ISO/IEC/IEEE 29148 Systems and software engineering — Life cycle processes — Requirements engineering

[ISO - ISO/IEC/IEEE 29148:2011 - Ingeniería de sistemas y software — Procesos de ciclo de vida — Ingeniería de requisitos](#)

<https://www.iso.org/standard/72081.html>

Norma ISO/IEC 12207 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes

[ISO - ISO/IEC/IEEE 12207:2017 - Ingeniería de sistemas y software — Procesos del ciclo de vida del software](#)

<https://www.iso.org/standard/63712.html>

ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes

[ISO - ISO/IEC 15288:2008 - Ingeniería de sistemas y software — Procesos del ciclo de vida del sistema](#)

<https://www.iso.org/standard/46403.html>

Estándar IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications

[830-1993 - Práctica recomendada IEEE para especificaciones de requisitos de software | Estándar IEEE | IEEE Xplore](#)

<https://ieeexplore.ieee.org/document/720574>

ISO/IEC TR 24766:2009, Information technology — Systems and software engineering — Guide for requirements engineering tool capabilities

[ISO - ISO/IEC TR 24766:2009 - Tecnología de la información — Ingeniería de sistemas y software — Guía para las capacidades de las herramientas de ingeniería de requisitos](#)

<https://www.iso.org/standard/45100.html>

IEEE Std 1233, IEEE Guide for Developing System Requirements Specifications

[1233-1998 - Guía IEEE para desarrollar especificaciones de requisitos del sistema | Estándar IEEE | IEEE Xplore](#)

<https://ieeexplore.ieee.org/document/720573>