

Documento de especificación del Modelo de procesos de desarrollo

Aseguramiento de la Calidad de Software

Versión 2.1

Estatus: Pendiente

Elaboró	Revisó	Autorizó
Nombres: Manuel Antonio Cituk Martínez Jorge Teodoro Dawn Rodríguez Geovanna Fernanda De La Cruz Medina Alma Angélica Ordoñez Sánchez Verónica Marilyn Rivera Manzanero Cargo: Estudiantes	Nombre: Cargo:	Nombre: Cargo:

Índice

Introducción	3
Prólogo	3
Descripción	3
Requerimientos	3
Análisis y diseño	3
Construcción	3
Integración y pruebas	3
Cierre	3
Objetivos	4
Indicadores	4
Metas cuantitativas	4
Entradas	
Plan de desarrollo	4
Salidas	4
Especificación de Requerimientos	4
Análisis y Diseño	5
Registro de rastreo	6
Componente	6
Software	6
Manual de usuario	6
Manual de operación	7
Plan de pruebas del sistema	7
Plan de pruebas de integración	7
Reporte de pruebas de integración	8
Manual de mantenimiento	9
Reporte de actividades	9
Mejoramiento de prácticas	9
Reporte de revisiones	9
Sugerencias de mejora	10
Mediciones	10
Prácticas	10
Roles involucrados y capacitación	10
Actividades	12
Verificación y validación	22
Incorporación a la base de conocimiento	31

Introducción

El propósito de este documento es presentar procesos que brinden el aseguramiento de la calidad del software basado en el Modelo de Procesos para la Industria de Software (MoProSoft).

El cumplimiento de las evaluaciones del proceso establecidas en nuestro modelo, cumplirá con todos los niveles de capacidad de procesos de acuerdo con MoProSoft, teniendo como resultado, el poder determinar si la calidad de su producto es óptima.

Prólogo

Su objetivo principal es incorporar las mejores prácticas en gestión e ingeniería de software. Su incorporación en la industria eventualmente permitirá elevar la capacidad de ofrecer productos y servicios de software con calidad.

Descripción

El proceso de desarrollo y mantenimiento de software se compone de uno o más ciclos de desarrollo. Para llevar a cabo de manera satisfactoria cada uno de estos ciclos, se han propuesto determinadas fases que se deberán cumplir. Dichas fases son:

Requerimientos

Las actividades plasmadas en esta fase permitirán obtener el Documento de Especificación de requerimientos y el Plan de pruebas del sistema para lograr una relación de comunicación amena entre el cliente y el proyecto

Análisis y diseño

Los requerimientos previamente obtenidos permitirán obtener la estructura de los componentes los cuales serán la base de la construcción, la cual permitirá obtener la documentación de toda la fase.

Construcción

Se producen componentes a través de un conjunto de actividades, las cuales toman en cuenta el Análisis y diseño previamente realizados. Cabe destacar que de igual forma se realizan pruebas unitarias y de esta manera se obtienen los componentes probados.

Integración y pruebas

Se integran y prueban los componentes de software a través de un conjunto de actividades con el fin de que el software satisfaga todos los requerimientos. De igual forma se generan los manuales necesarios para obtener un producto de software probado y documentado.

Cierre

Se identifican y documentan las lecciones aprendidas, así como la generación del reporte de mediciones y las posibles mejoras.

Objetivos

O1 Lograr que los productos de salida sean consistentes con los productos de entrada en cada fase de un ciclo de desarrollo mediante las actividades de verificación, validación o prueba.

O2 Sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuros mediante la integración de la *Configuración de Software* del ciclo actual.

O3 Llevar a cabo las actividades de las fases de un ciclo mediante el cumplimiento del *Plan de Desarrollo* actual.

Indicadores

I1 (O1) En cada fase de un ciclo se efectúan todas las actividades de verificación, validación o prueba, así como las correcciones correspondientes.

I2 (O2) La *Configuración de Software* está integrada por los productos generados en el ciclo.

I3 (O3) Las actividades planificadas en cada fase de un ciclo se realizan conforme a lo establecido en el *Plan de Desarrollo*.

Metas cuantitativas

Valor numérico o rango de satisfacción por indicador.

Entradas

Nombre	Descripción
Plan de desarrollo	Se tiene el plan de trabajo completo a través de la descripción del producto, entregables, proceso específico, equipo de trabajo y calendario.

Salidas

Nombre	Descripción
Especificación de requerimientos	<p>Es el producto acerca de la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema.</p> <p>Durante la actividad de captura, recopilamos todos los requisitos de varias fuentes. Durante las actividades de análisis y negociación, analizamos y entendemos esos requisitos. Por último, se debe preparar un documento formal que explique esos requisitos.</p> <p>Este documento contendrá todos los requisitos que se van a imponer en el diseño y verificación del producto con el nivel de detalle correspondiente. También contendrá otra información</p>

	<p>relacionada necesaria para el diseño, verificación y mantenimiento del producto.</p> <p>Descripción de requerimientos</p> <ul style="list-style-type: none"> • Funcionales: Necesidades establecidas que debe satisfacer el software cuando es usado en condiciones específicas. Las funcionalidades deben ser adecuadas, exactas y seguras. • Interfaz con usuario: Definición de aquellas características de la interfaz de usuario que permiten que el software sea fácil de entender, aprender, que genere satisfacción y con el cual el usuario pueda desempeñar su tarea eficientemente. Incluyendo la descripción del prototipo de la interfaz. • Interfaces externas: Definición de las interfaces con otro software o con hardware. Confiabilidad: Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación. • Eficiencia: Especificación del nivel de desempeño del software con respecto al tiempo y a la utilización de recursos. • Mantenimiento: Descripción de los elementos que facilitarán la comprensión y la realización de las modificaciones futuras del software. • Portabilidad: Descripción de las características del software que permitan su transferencia de un ambiente a otro. • Restricciones de diseño y construcción: Necesidades impuestas por el cliente. • Legales y reglamentarios: Necesidades impuestas por leyes, reglamentos, entre otros.
--	---

Nombre	Descripción
Análisis y Diseño	<p>En cuanto a esta sección, se establecen las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos, en esta parte se traducen los requisitos en una representación de software.</p> <p>Conjunto de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación del Análisis y Diseño y Plan de Pruebas de Integración.</p>

Nombre	Descripción
Registro de Rastreo	Se comprueba que exista una exitosa relación entre los requerimientos, diseño y componentes para poder pasar a la fase de pruebas.

Nombre	Descripción
Componente	Unidades de código relacionadas. Recursos desarrollados para un fin concreto, puede formar un entorno funcional requerido..

Nombre	Descripción
Software	Sistema de software que proporciona funciones destinadas a un cliente o usuario. Está conformado por componentes agrupados en subsistemas..

Nombre	Descripción
Manual de Usuario	El manual de usuario de software debe proporcionar una guía clara y detallada sobre cómo usar el software. Debe incluir información sobre la instalación y configuración del software, así como una descripción detallada de las funciones y características del software. Además, debe incluir información sobre cualquier problema conocido y cómo solucionarlo, y también debe proporcionar instrucciones sobre cómo utilizar el software de manera segura y eficiente. Por último, el manual de usuario de software debe ser fácil de leer y entender, y estar actualizado con cualquier información importante y actualizaciones de software relevantes.

Nombre	Descripción
Manual de Operación	El manual de operación debe contener la información necesaria para capacitar a los usuarios y operadores del sistema. Debe contener, pero no limitarse a, lo siguiente:

	<ul style="list-style-type: none"> a) Una introducción b) Cómo usar el sistema c) Cómo preparar la entrada d) Descripciones de entrada de datos e) Descripciones de control de datos f) Cómo ejecutar el sistema g) Distribuciones de producción h) Descripción de los datos de salida e interpretaciones (por ejemplo, mensajes de error) i) Tutoriales y ejercicios de práctica j) Cómo obtener ayuda
--	---

Nombre	Descripción
Plan de Pruebas del Sistema	<p>Introducción: Descripción general del objetivo y alcance del plan de pruebas, incluyendo los criterios de aceptación y las fechas de inicio y finalización previstas.</p> <p>Enfoque de prueba: Descripción de la metodología y estrategia de prueba, incluyendo la división en fases y los métodos de prueba que se utilizarán.</p> <p>Requisitos y especificaciones: Lista de los requisitos y especificaciones del sistema que se deben probar y cómo se abordarán cada uno.</p> <p>Casos de prueba: Detalle de los casos de prueba, incluyendo el objetivo de la prueba, los pasos y las condiciones esperadas.</p> <p>Procedimiento de ejecución: Descripción detallada del procedimiento de ejecución de las pruebas, incluyendo la configuración del sistema y cualquier herramienta o equipo necesario.</p>

Nombre	Descripción
Plan de Pruebas de Integración	<p>El objetivo de este documento es recoger la definición de las pruebas realizadas en múltiples sistemas integrados completos para evaluar su capacidad para comunicarse con éxito entre sí y para cumplir con los requisitos generales especificados de los sistemas integrados.</p> <p>Los aspectos clave de la planificación de pruebas incluyen la coordinación del personal, la disponibilidad de instalaciones y equipos de prueba, la creación y el mantenimiento de toda la documentación relacionada con las pruebas y la planificación de posibles resultados no deseados. Si se mantiene más de una línea de base del software, entonces una consideración importante de planificación es el tiempo y el esfuerzo necesarios para garantizar que el entorno de prueba se establezca en la configuración adecuada.</p>

Nombre	Descripción
Reporte de Pruebas de Integración	<p>El plan está conformado por las siguientes partes:</p> <ul style="list-style-type: none"> • <i>Portada</i>: Nombre del proyecto, nombre del equipo de pruebas, fecha del informe y otros datos relevantes. • <i>Resumen/Abstract</i>: Breve resumen de los resultados de las pruebas de integración, incluyendo el número de casos de prueba ejecutados, el número de errores y fallos identificados y el número de errores y fallos resueltos. • <i>Introducción</i>: Descripción general del objetivo y alcance de las pruebas de integración y una breve descripción de la metodología y estrategia de prueba. • <i>Resultados de las pruebas</i>: Detalle de los resultados de las pruebas de integración, incluyendo el número de casos de prueba ejecutados, el número de errores y fallos identificados y el número de errores y fallos resueltos. • <i>Conclusiones</i>: Análisis de los resultados de las pruebas de integración y conclusiones sobre el cumplimiento de los requisitos y especificaciones. • <i>Recomendaciones</i>: Recomendaciones para solucionar cualquier problema identificado durante las pruebas de integración y mejorar la calidad del software. • <i>Anexos</i>: Cualquier información adicional relevante, incluyendo capturas de pantalla, registros de prueba, documentación técnica, etc.

Nombre	Descripción
Manual de Mantenimiento	<p>El manual de mantenimiento debe de seguir las denominadas “categorías de mantenimiento” las cuales son:</p> <ol style="list-style-type: none"> 1. Mantenimiento correctivo: modificaciones reactivas a un producto software hechas después de la entrega para corregir defectos descubiertos. 2. Mantenimiento adaptativo: modificación de un producto software realizada después de la entrega para permitir que un producto software siga pudiéndose utilizar en un entorno diferente. 3. Mantenimiento perfectivo: modificación de un producto software después de la entrega para mejorar el rendimiento o la mantenibilidad.

Nombre	Descripción
--------	-------------

Reporte de Actividades	Se genera un registro de actividades, siguiendo un estándar, incluyendo una introducción, fechas de inicio y final, descripción de las actividades, así como los responsables de estas actividades. Se registra el tiempo de cada una, así como los defectos encontrados en la verificación, validación o prueba
-------------------------------	--

Nombre	Descripción
Mejoramiento de prácticas	Se realiza un registro con las buenas prácticas realizadas, se toman en cuenta los problemas frecuentes y las experiencias exitosas al solucionar los problemas que se encontraron en el ciclo de desarrollo y mantenimiento.

Nombre	Descripción
Reporte de Mediciones	Registro donde detalla la evaluación y medición de los procesos de software, se tienen las mediciones de los indicadores del proceso de desarrollo y mantenimiento de software. El objetivo principal del reporte es medir la eficacia y la eficiencia de los procesos e identificar áreas que requieren mejora. (Ver Mediciones)

Nombre	Descripción
Sugerencias de Mejora	Reporte que contiene las sugerencias para mejorar el proceso de desarrollo y mantenimiento de software, así como una descripción de las áreas a mejorar, se registran los métodos, herramientas, formatos y estándares que se han empleado

Mediciones

Al final de cada ciclo se genera un reporte del estado de los indicadores del proceso con respecto a las metas cuantitativas definidas, se sugieren las siguientes mediciones:

- **M1 (I1)** Revisar los Reportes de Verificación, Reportes de Validación y/o reportes de pruebas de cada fase para la confirmación de que se han realizado estas actividades y se han incorporado las correcciones.
- **M2 (I2)** Revisar la Configuración de Software para comprobar que los productos que la integran son los mismos que se generaron en el ciclo.
- **M3 (I3)** Comparar el Plan de Desarrollo actual para cada fase con el Reporte de Actividades

Prácticas

Roles involucrados y capacitación

Rol	Abreviatura	Descripción
Responsable de Administración del Proyecto Específico	RAPE	Persona que posee la capacidad de liderazgo y experiencia en la toma de decisiones, buena comunicación, planificación estratégica, gestión de recursos, manejo de personal y desarrollo de software.
Responsable de Desarrollo y Mantenimiento de Software	RDM	Persona que posee conocimiento y experiencia en el desarrollo y mantenimiento de software
Analista	AN	Persona que posee conocimiento y experiencia en la obtención, especificación y análisis de los requerimientos, trabajo en equipo, capacidades para encontrar soluciones, proactividad, creatividad, capacidad de adaptación.
Diseñador de Interfaz de Usuario	DU	Persona con conocimiento en diseño de interfaces de usuario y criterios ergonómicos, dominio de alguna herramienta de prototipado navegable. Interpretación de reportes de analítica digital y entender buenas prácticas en SEO.
Diseñador	DI	Persona con conocimiento y experiencia en el diseño de la estructura de los componentes de software.
Programador	PR	Persona con conocimiento y/o experiencia en la programación, integración, pruebas unitarias y capacidad para trabajar en equipo.
Responsable de Pruebas	RPU	Persona con conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema.
Revisor	RE	Persona con conocimientos en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de software.
Responsable de	RM	Persona con conocimientos en las técnicas de

Manuales		redacción y experiencia en el desarrollo y mantenimiento de software.
Equipo de Trabajo	ET	Persona con conocimiento y experiencia de acuerdo a su rol.
Cliente	CL	Es aquella persona que interpreta el estándar de la especificación de requerimientos.
Usuario	US	Ninguna

Actividades

Rol A1. Realización de la fase de inicio	
ET	A1.1. Los miembros del equipo deberán revisar el Plan de Desarrollo actual a través de reuniones consecutivas, con el fin de lograr un entendimiento común y obtener su compromiso con el proyecto.
RDM	A1.2. Se debe de elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
ET	A1.3 Se debe de comprobar que lo documentado en el reporte de actividades coincida con los objetivos iniciales
A2. Realización de la fase de Requerimientos	
RDM AN	A2.1. Se deberá realizar una distribución de tareas a cada miembro del equipo dependiendo de sus roles y lo anterior, de acuerdo al Plan de Desarrollo actual.
AN CL US DU	<p>A2.2. La especificación de requerimientos se deberá documentar y/o modificar</p> <ul style="list-style-type: none"> → A través de investigaciones realizadas en diferentes fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) se podrán obtener nuevos requerimientos. → El analizar los requerimientos identificados permitirá delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto. → Elaborar o modificar el prototipo de la interfaz con el usuario.

	→ Generar o actualizar la Especificación de Requerimientos.
RE	A2.3. Especificación de Requerimientos. (Verificación)
AN DU	A2.4. Corregir los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Validación y obtener la aprobación de las correcciones.
RPU AN	A2.5. Elaborar o modificar Plan de Pruebas de Sistema.
RE	A2.6 Verificar el Plan de Pruebas de Sistema.
RPU	A2.7 En caso de haber errores en el Plan de Pruebas de Sistema, se deberán corregir con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RM	A2.8 Documentar la versión preliminar del Manual de Usuario o modificar el manual existente.
RE	A2.9 Verificar el Manual de Usuario.
RM	A2.10 Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A2.11. Incorporar Especificación de Requerimientos, Plan de Pruebas de Sistema y Manual de Usuario como líneas base a la Configuración de Software.
RDM	A2.12. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
A3. Realización de la fase de Análisis y Diseño	
RDM AN DI	A3.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo con el Plan de Desarrollo actual.
AN DI DU	<p>A3.2. Documentar o modificar el Análisis y Diseño:</p> <ul style="list-style-type: none"> • Analizar Especificación de Requerimientos para generar descripción de estructura interna del sistema. • Descomponer el sistema en subsistemas y definir interfaces entre ellos. • Describir detalles de la apariencia y comportamiento de la interfaz con base en la Especificación de Requerimientos. • Describir detalles de los componentes para su construcción de manera evidente.

	<ul style="list-style-type: none"> • Generar/actualizar Análisis y Diseño. • Generar/modificar Registro de Rastreo.
RE	A3.3. Verificar el Análisis y Diseño y el Registro de Rastreo.
AN DI DU	A3.4. Corregir los defectos encontrados en el Análisis y Diseño y en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
CL RPU	A3.5. Validar el Análisis y Diseño.
AN DI DU	A3.6. Corregir los defectos encontrados en el Análisis y Diseño con base en el Reporte de Validación y obtener la aprobación de las correcciones.
RPU	A3.7. Elaborar o modificar Plan de Pruebas de Integración.
RE	A3.8. Verificar el Plan de Pruebas de Integración.
RPU	A3.9. Corregir los defectos encontrados en el Plan de Pruebas de Integración con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A3.10. Incorporar Análisis y Diseño, Registro de Rastreo y Plan de Pruebas de Integración como líneas base a la Configuración de Software.
RDM	A3.11. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
A4. Realización de la fase de Construcción	
RDM	<p>A4.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.</p> <p>La gestión de la configuración de software se emplea de acuerdo con el estándar IEEE 828-2012, donde especifica que debe haber una sección sobre asignación de las responsabilidades la cuál define quien es el responsable de realizar cada tarea seleccionada y el rol que tendrá.</p> <p>La fase de construcción contará con documentos de control de acuerdo con el libro “Software Quality Assurance”:</p> <p><i>Documentos previos al proyecto:</i></p> <ul style="list-style-type: none"> - Software development contract - Software development plan

	<p><i>Documentos del ciclo de vida del proyecto:</i></p> <ul style="list-style-type: none"> - Database description - Software test plan - Software test procedure - Software test report - Version description document <p><i>Gestión de calidad:</i></p> <ul style="list-style-type: none"> - Software metrics reports
PR	<p>A4.2. Construir o modificar el(los) Componente(s) de software</p> <ul style="list-style-type: none"> • Implementar o modificar Componente(s) con base a la parte detallada del Análisis y Diseño. <p>La gestión de la configuración de software se emplea de acuerdo con el estándar IEEE 828-2012, este estándar recomienda que se establezcan planes de gestión de la configuración de software para definir los procedimientos y políticas de gestión de la configuración de software. Se deben establecer y mantener una base de datos de gestión de la configuración de software para almacenar información sobre los elementos de configuración de software y su estado.</p> <p>Elementos de configuración aplicables en la realización de la fase de construcción:</p> <p>ITEMS (CI)</p> <ul style="list-style-type: none"> → Código ejecutable → Código fuente → Bases de datos → Lista de programas → Compiladores → Sistemas operativos → Herramientas de programación → Registro de rastreo → Plan de pruebas <p>De igual forma, el estándar recomienda la implementación de procesos de control de versiones y cambios, esto con el fin de garantizar que los cambios se hayan hecho y se tenga un historial de las modificaciones realizadas.</p> <p>Como guía de buenas prácticas para la construcción del código, se utilizará “Python PEP8”, la cuál contiene los lineamientos necesarios para que el código en Python sea legible y consistente.</p> <p>La verificación de una buena construcción del código se validará a través de la plantilla: “Software Code review Template”- Solutions</p>

DevCare, la cual contiene diferentes checklist que ayudarán a mantener la calidad del código.

- Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del Análisis y Diseño.

Las pruebas unitarias se realizarán de manera automática, de acuerdo con el estándar "**Python Unit Test Automation: Practical Techniques for Python Developers and Testers**" el cuál cubre los conceptos básicos de las pruebas unitarias en Python y proporciona información detallada sobre cómo utilizar el módulo unittest y otras herramientas de prueba para automatizar pruebas efectivas en Python. De igual forma, cubre temas avanzados como el desarrollo impulsado por pruebas (TDD), la integración continua y cómo crear informes de prueba útiles. Además, proporciona ejemplos prácticos de cómo escribir pruebas para diferentes tipos de aplicaciones de Python, como aplicaciones web y aplicaciones de ciencia de datos.

- Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos).

La corrección de defectos se llevará a cabo de igual forma de acuerdo con el estándar "Python Unit Test Automation", el cual establece los siguientes pasos:

1. Identificar el defecto: Lo primero que hay que hacer es identificar el defecto que está causando que las pruebas unitarias fallen. Para ello, se puede revisar el mensaje de error que se muestra al ejecutar las pruebas unitarias.
2. Reproducir el defecto: Una vez identificado el defecto, es importante reproducirlo para entender cómo está afectando el código. Para ello, se puede crear un script o función de prueba que ejecute el código y muestre el resultado.
3. Corregir el defecto: Una vez que se ha identificado y reproducido el defecto, hay que corregirlo. Esto puede implicar modificar el código, cambiar la lógica o agregar nuevas funciones. Es importante asegurarse de que la corrección no afecte otras partes del código y que no introduzca nuevos defectos.

	<p>4. Ejecutar pruebas unitarias: Después de corregir el defecto, es importante ejecutar nuevamente las pruebas unitarias para asegurarse de que todas pasen. Si las pruebas unitarias siguen fallando, se puede repetir el proceso de identificar y corregir el defecto.</p> <p>5. Refactorizar el código: Una vez que todas las pruebas unitarias pasan, se puede considerar refactorizar el código para mejorar su calidad y claridad. La refactorización puede implicar la eliminación de código redundante, la creación de funciones más pequeñas y fáciles de entender, y la mejora de la legibilidad del código.</p> <p>6. Ejecutar pruebas unitarias de nuevo: Finalmente, es importante ejecutar las pruebas unitarias nuevamente después de la refactorización para asegurarse de que el código sigue funcionando correctamente y que no se han introducido nuevos defectos.</p> <p>Para verificar que se haya hecho una buena ejecución de las pruebas, se usará una “Test-plan-checklist” la cuál se encuentra construida con base en el estándar IEEE 829-2008.</p>
RE	<p>A4.3. Verificar el Registro de Rastreo.</p> <ul style="list-style-type: none"> ● Actualizar el Registro de Rastreo, incorporando los componentes contruidos o modificados. <p>Para lo anterior, se hará uso de la herramienta “Github”, la cual implementa el rastreo de cambios utilizando Git y sus características de control de versiones, además proporciona herramientas para que los usuarios realicen un seguimiento de los cambios realizados en los proyectos alojados en su plataforma.</p> <p>Para realizar dicho registro, se deberá crear y configurar un repositorio para el equipo de desarrollo, usar las listas de tareas y seguir el path que requiere el registro de rastreo para ser aprobado.</p> <p>El registro de rastro deberá realizarse de acuerdo con la plantilla que propone la herramienta:</p> <p><u>Plantilla de liberación de rastreo:</u> Propuesta para rastrear el progreso para un lanzamiento o para los pasos a completar el día del lanzamiento, la cuál se implementará de la siguiente manera:</p>

	<ul style="list-style-type: none"> <input type="checkbox"/> Repartición de tareas <input type="checkbox"/> Creación de “ramas de trabajo”, de acuerdo con los equipos. <input type="checkbox"/> Pull requests <input type="checkbox"/> Reporte de problemas <input type="checkbox"/> Corrección de defectos <input type="checkbox"/> Aprobación de las correcciones por medio de pull requests <input type="checkbox"/> Liberación del rastreo. <p>El cumplimiento de todas las tareas de cada miembro del equipo, antes de solicitar un pull request, deberá verificarse a través de una checklist.</p>
PR	<p>A4.4. Corregir los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.</p> <p>Los “pull requests” son revisiones que permiten a los colaboradores comentar los cambios propuestos en las solicitudes de extracción, aprobar los cambios o solicitar cambios adicionales antes de fusionar la solicitud de extracción.</p> <p>El equipo de desarrollo deberá enviar pull requests de sus cambios y estar pendientes de la aprobación de los líderes del equipo. Si los líderes detectan defectos, deberán crear una plantilla de Reporte de problemas, github brinda las siguientes opciones al momento de creación de dicha plantilla:</p> <ul style="list-style-type: none"> - Bug report - Feature request - Custom template <p>De esta manera, los líderes rechazan el pull request y solicitan el reporte de problemas.</p> <p>El equipo de desarrollo deberá realizar las correcciones correspondientes y seguir el mismo procedimiento hasta obtener la aprobación total de sus pull request.</p>
RDM	<p>A4.5. Incorporar Componentes y Registro de Rastreo como líneas base a la Configuración de Software.</p> <p>Lo anterior se tendrá que realizar de acuerdo a las especificaciones de nuestro documento de administración de la configuración de software, el cual se basó en el estándar IEEE 828-2012, "Standard</p>

	for Configuration Management in Systems and Software Engineering", esto con el fin de mantener una gestión efectiva de la configuración de software para el Modelo de Procesos de software.
RDM	<p>A4.6. Elaborar el Reporte de Actividades.</p> <p>Se deberán registrar las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. Lo anterior de acuerdo con la Norma ISO 9001: La norma exige que se documenten los resultados de las actividades realizadas, así como los procesos utilizados para llevarlas a cabo.</p> <p>La plantilla a implementar para dicho reporte será: "Informe del estado del proyecto" creada por Microsoft.</p>
A5. Realización de la fase de Integración y Pruebas	
RDM	A5.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
PR RPU	<p>A5.2. Realizar integración y pruebas.</p> <ul style="list-style-type: none"> • Integrar los componentes en subsistemas o en el sistema del Software y aplicar las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración. • Corregir los defectos encontrados, con base en Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (sin defectos). • Actualizar el Registro de Rastreo.
RM	A5.3. Documentar el Manual de Operación o modificar el manual existente.
RE	A5.4. Verificar el Manual de Operación.
RM	A5.5. Corregir los defectos encontrados en el Manual de Operación con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RPU	A5.6. Realizar las pruebas de sistema siguiendo el Plan de Pruebas de Sistema, documentando los resultados en un Reporte de Pruebas de Sistema.
PR	A5.7. Corregir los defectos encontrados en las pruebas de sistema con base en el Reporte de Pruebas de Sistema y obtener la aprobación de las correcciones.
RM	A5.8. Documentar el Manual de Usuario o modificar el existente.

RE	A5.9. Verificar el Manual de Usuario.
RM	A5.10. Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A5.11. Incorporar Software, Reporte de Pruebas de Integración, Registro de Rastreo, Manual de Operación y Manual de Usuario como líneas base a la Configuración de Software.
RDM	A5.12. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
A6. Realización de la fase de Cierre	
RM	A6.1. Documentar el Manual de Mantenimiento o modificar el existente.
RE	A6.2. Verificar el Manual de Mantenimiento.
RM	A6.3. Corregir los defectos encontrados en el Manual de Mantenimiento con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A6.4. Incorporar Manual de Mantenimiento como línea base a la Configuración de Software.
RDM ET	A6.5. Identificar las Lecciones Aprendidas e integrarlas a la Base de Conocimiento. Como ejemplo, se pueden considerar mejores prácticas, experiencias exitosas de manejo de riesgos, problemas recurrentes, entre otras.
RDM ET	<p>A6.6. Generar el Reporte de Mediciones. Considerar los siguientes pasos:</p> <ol style="list-style-type: none"> 1. Identificar los procesos a medir y las actividades serán objeto de medición. 2. Definir las métricas que se utilizarán para medir los procesos, como tiempos de ciclo, calidad del software, entre otras. 3. Analizar los resultados de las mediciones para identificar tendencias, fortalezas y debilidades. 4. Crear el informe de forma detallada que incluya una introducción, descripción de los procesos evaluados, métricas utilizadas, resultados y conclusiones.
RDM ET	A6.7. Generar el Reporte de Sugerencias de Mejora, considerando los siguientes puntos:

	<ol style="list-style-type: none"> 1. Se analizan los procesos del modelo para identificar y determinar qué áreas necesitan mejoras. 2. Evaluar diferentes alternativas para mejorar las áreas identificadas y seleccionar las mejores opciones. 3. Revisar las soluciones propuestas para determinar si son viables y su impacto en los procesos. 4. Establecer una planificación con acciones concretas para implementar las mejoras seleccionadas. 5. Se presenta el informe en un formato adecuado después de haber realizado las revisiones y correcciones necesarias.
RDM	<p>A6.8. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. Se considera lo siguiente:</p> <ol style="list-style-type: none"> 1. Identifica el alcance, es decir, se analiza qué actividades incluir en el informe. 2. Recopila información, datos y documentación sobre las actividades relevantes. 3. Se organiza la información recopilada en una forma que sea lógica y clara. 4. Se hace una revisión y se edita el reporte para asegurarte de que la información sea precisa, clara y concisa. 5. Se presenta el informe en un formato adecuado, ya sea en papel o en formato electrónico.

Verificación y validación

Verificación o validación	Actividad	Producto	Rol	Descripción
Ver1	A2.3	Especificación de Requerimientos	RE	Verificar la claridad de redacción de la Especificación de Requerimientos y su consistencia con la Descripción del Producto y con el estándar de documentación requerido en el Proceso Específico. Adicionalmente revisar que los requerimientos sean completos y no ambiguos o contradictorios. Los defectos encontrados se documentan en un Reporte de Verificación.

				<p>Durante el proceso de validación de requerimientos, se deben llevar a cabo verificaciones sobre requerimientos en el documento de requerimientos [5]. Estas verificaciones comprenden:</p> <ul style="list-style-type: none"> • Validez: No basta con preguntar a un usuario, todos los potenciales usuarios pueden tener puntos de vista distintos y necesitar otros requisitos. • Consistencia: No debe haber contradicciones entre unos requisitos y otros. • Compleitud: Deben estar todos los requisitos. Esto es imposible en un desarrollo iterativo, pero, al menos, deben estar disponibles todos los requisitos de la iteración en curso. • Realismo: Se pueden implementar con la tecnología actual. • Verificabilidad: Tiene que existir alguna forma de comprobar que cada requisito se cumple. <p>Los defectos encontrados se documentan en un Reporte de Verificación</p>
Val1	A2.9	Especificación de Requerimientos	CL US RPU	<p>Validar que la Especificación de Requerimientos cumple con las necesidades y expectativas de las partes interesadas, incluyendo la realización de la prueba de usabilidad de la interfaz del usuario.</p> <p>Requisitos para la Validación de Requisitos:</p> <ul style="list-style-type: none"> • Requisitos Documentos – Esta debe ser una versión completa del documento, no un borrador sin terminar. Formateado y organizado de acuerdo con los estándares organizacionales. • Conocimiento Organizacional – Conocimiento, a menudo implícito, de la organización que puede utilizarse para juzgar el realismo de los requisitos.

				<ul style="list-style-type: none"> ● Estándares organizacionales: Normas locales, por ejemplo, para la organización del documento de requisitos. <p>Salidas de validación de requisitos:</p> <ul style="list-style-type: none"> ● Lista de problemas: Lista de problemas descubiertos en el documento de requisitos. Los defectos encontrados se documentan en un Reporte de Validación ● Acciones Acordadas: Lista de acciones acordadas en respuesta a problemas de requisitos. Algunos problemas pueden tener varias acciones correctivas; algunos problemas pueden no tener acciones asociadas.
Ver2	A2.15	Plan de Pruebas de Sistema	RE	<p>Verificar consistencia del Plan de Pruebas de Sistema con la Especificación de Requerimientos y con el estándar de documentación requerido en el Proceso Especifico. Los defectos encontrados se documentan en un Reporte de Verificación.</p> <p>Las actividades incluidas en un plan de prueba del sistema de software suelen incluir:</p> <p>Desarrollo de la estrategia de prueba: definición del enfoque general para probar el sistema, incluidos los niveles de prueba, los tipos de prueba y los métodos de prueba.</p> <p>Configuración del entorno de prueba: establecimiento de las configuraciones de hardware y software necesarias para la prueba.</p> <p>Preparación de datos de prueba: creación y organización de datos de prueba que se utilizarán durante la prueba.</p> <p>Desarrollo de casos de prueba: crear y documentar casos de prueba detallados para validar la funcionalidad y el rendimiento del sistema.</p>

				<p>Ejecución de pruebas: ejecutar los casos de prueba y documentar los resultados.</p> <p>Análisis de resultados de pruebas: evaluación de los resultados de las pruebas para determinar si el sistema cumple con sus requisitos e identificar cualquier defecto.</p> <p>Seguimiento y resolución de defectos: documentar y rastrear cualquier defecto encontrado durante las pruebas y trabajar con el equipo de desarrollo para resolverlo.</p> <p>Prueba final: realización de pruebas finales para verificar que el sistema se haya probado por completo y que se hayan resuelto todos los defectos.</p> <p>Cierre de prueba: Archivar los artefactos de prueba y documentar los resultados del esfuerzo de prueba.</p>
Ver3	A2.18	Manual de Usuario	RE	Verificar consistencia del Manual de Usuario con la Especificación de Requerimientos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.
Ver4	A3.3	Análisis y Diseño Registro de Rastreo	RE	Verificar claridad de la documentación del Análisis y Diseño, su factibilidad y la consistencia con la Especificación de Requerimientos y con el estándar de documentación requerido en el Proceso Específico. Verificar que el Registro de Rastreo contenga las relaciones adecuadas entre los requerimientos y los elementos de Análisis y Diseño. Los defectos encontrados se documentan en un Reporte de Verificación.
Val2	A3.5	Análisis y Diseño	CL RPU	<p>Validar que el Análisis y Diseño cumple con las necesidades y expectativas acordadas con el cliente. Se distinguen tres etapas distintas:</p> <p>1) Definir objetivos de validación</p>

				<p>2) Definición de la muestra y de la guía de validación:</p> <p>3) Se muestra la solución al usuario y se recibe feedback</p> <p>Los defectos encontrados se documentan en un Reporte de Validación.</p>
Ver5	A3.8	Plan de Pruebas de Integración	RE	<p>Verificar consistencia del Plan de Pruebas de Integración con el Análisis y Diseño y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación</p> <p>Las actividades del plan de pruebas de integración deberán incluir:</p> <ul style="list-style-type: none"> • Identificar los componentes de software que se van a integrar • Definir el entorno de prueba, incluida la configuración de hardware y software • Desarrollar casos de prueba que verifiquen la integración entre componentes • Ejecutar los casos de prueba y documentar los resultados. • Depuración y solución de los problemas detectados durante las pruebas • Evaluación del rendimiento y la fiabilidad general del sistema • Finalizar el plan de pruebas documentando los resultados y las conclusiones.
Ver6	A4.3	Registro de Rastreo	RE	<p>Verificar que el Registro de Rastreo contenga las relaciones adecuadas entre los elementos de Análisis y Diseño y los componentes. Los defectos encontrados se documentan en un Reporte de Verificación.</p> <p>El registro de rastreo deberá tener las siguientes actividades:</p> <p>Registro de los requisitos: vinculación de los requisitos de software con los artefactos de diseño, implementación y prueba para garantizar que se han abordado y cumplido todos los requisitos.</p>

				<p>Gestión de cambios: registro de los cambios realizados en los artefactos de software y seguimiento del impacto de dichos cambios en los artefactos relacionados.</p> <p>Control de versiones: seguimiento de las distintas versiones de los artefactos de software, quién realizó los cambios y cuándo se hicieron.</p> <p>Gestión de la configuración: Gestionar los artefactos que componen un sistema de software y garantizar que sólo se realizan cambios autorizados.</p> <p>Seguimiento de incidencias: registro y seguimiento de problemas, errores e incidencias del software y su resolución.</p> <p>Registro de las pruebas: vincular las pruebas a los requisitos, el diseño y los artefactos de implementación para garantizar que todas las pruebas se realizan y superan.</p> <p>Documentación: creación y mantenimiento de documentación que describa el sistema de software, su arquitectura y sus componentes.</p>
Ver7	A5.4	Manual de Operación	RE	<p>Verificar consistencia del Manual de Operación con el Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.</p> <p>La verificación de un manual de operación de software es un proceso importante para garantizar que la información proporcionada sea clara, precisa y fácil de seguir para los usuarios. Aquí hay algunos pasos que puedes seguir para realizar una verificación efectiva:</p> <ol style="list-style-type: none"> 1. Leer el manual completo y en su totalidad para tener una comprensión general de las instrucciones y procedimientos. 2. Verificar la precisión de la información técnica y los detalles del software.

				<ol style="list-style-type: none"> 3. Asegurarse de que los pasos se describan de manera clara y concisa, sin saltarse ningún detalle importante. 4. Comprobar la consistencia de la información a lo largo del manual y corregir cualquier incongruencia. 5. Probar los procedimientos y seguir las instrucciones para asegurarse de que sean fáciles de seguir y efectivos. 6. Recopilar retroalimentación de otros usuarios y corregir cualquier problema identificado. 7. Revisar y actualizar el manual periódicamente para reflejar cualquier cambio en el software o en su uso. <p>Es importante tener en cuenta que la verificación del manual de operación es un proceso continuo y es necesario realizar revisiones regulares para mantener la precisión y la efectividad de la información proporcionada.</p>
Ver8	A5.9	Manual de Usuario	RE	<p>Verificar consistencia del Manual de Usuario con el sistema de Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.</p> <p>Introducción: Esta sección proporciona una visión general del software, su propósito y el público objetivo.</p> <p>Comenzando: Esta sección proporciona instrucciones paso a paso sobre cómo instalar y configurar el software.</p> <p>Interfaz de usuario: Esta sección describe la interfaz del software y sus diversos componentes, como menús, botones y herramientas.</p> <p>Características y funciones: Esta sección proporciona una descripción detallada de las características del software y cómo usarlas.</p> <p>Guías "cómo hacer": Esta sección proporciona instrucciones paso a paso para realizar tareas específicas con el software.</p>

				<p>Resolución de problemas: Esta sección proporciona soluciones a problemas comunes que los usuarios pueden encontrar al usar el software.</p> <p>Referencia: Esta sección proporciona una referencia completa de las características del software, incluidos detalles sobre configuraciones, opciones y comandos.</p> <p>Glosario: Esta sección proporciona definiciones de términos técnicos utilizados en el software y su documentación.</p>
Ver9	A6.2	Manual de Mantenimiento	RE	<p>Verificar consistencia del Manual de Mantenimiento con la Configuración de Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.</p> <p>La verificación de un manual de mantenimiento de software es un proceso importante que se realiza para asegurarse de que el manual sea preciso, completo y fácil de usar. Aquí hay algunos pasos que se pueden seguir para realizar una verificación efectiva:</p> <ol style="list-style-type: none"> 1. Revisión de contenido: Verificar que el manual incluya todas las instrucciones y procedimientos necesarios para mantener el software de manera efectiva. 2. Coherencia y claridad: Asegurarse de que el manual sea coherente en su estructura y presentación y que los procedimientos se describan claramente. 3. Precisión: Verificar que la información en el manual sea precisa y que no haya errores de gramática o ortografía. 4. Pruebas de usuario: Solicitar a un grupo de usuarios que sigan los procedimientos en el manual y reporten cualquier problema o ambigüedad.

				<p>5. Revisión por expertos: Asignar a un experto en el software para revisar el manual y asegurarse de que toda la información sea precisa y que los procedimientos se describan adecuadamente.</p> <p>Estos son algunos de los pasos que se pueden seguir para realizar una verificación efectiva de un manual de mantenimiento de software. Es importante que se realice esta verificación de forma periódica para asegurarse de que el manual siga siendo preciso y útil para los usuarios.</p>
--	--	--	--	---

Incorporación a la Base de Conocimiento

Producto	Forma de aprobación
Especificación de Requerimientos	Ver1, Val1
Plan de Pruebas de Sistema	Ver2
Manual de Usuario	Ver3
Análisis y Diseño	Ver4, Val2
Registro de Rastreo	Ver4
Plan de Pruebas de Integración	Ver5

Componente(s)	Prueba unitaria exitosa
Registro de Rastreo	Ver6
Software	Prueba de integración exitosa, prueba de sistema exitosa
Manual de Operación	Ver7
Manual de Usuario	Ver8
Manual de Mantenimiento	Ver9
Reporte de Pruebas de Integración	Ninguna
Reporte de Pruebas de Sistema	Ninguna
Reporte(s) de Actividades	Ninguna
Lecciones Aprendidas	Ninguna
Reporte(s) de Verificación	Ninguna
Reporte(s) de Validación	Ninguna