

RESOLUCIÓN DE CONSEJO DE COMISIÓN ORGANIZADORA

N.º 161-2025-CCO-UNAJ

Juliaca, 26 de febrero de 2025.

VISTOS:

La Carta N° 064-2025-OTI-UNAJ/HVC, de fecha 14 de febrero de 2025; Acuerdo N.º 148-2025-SO-CCO-UNAJ de Sesión Ordinaria de Consejo de Comisión Organizadora de fecha 19 de febrero de 2024; Informe N° 044-2025-OPP/UPM-UNAJ, de fecha 26 de febrero de 2025; y,

CONSIDERANDO:

Que, conforme a lo dispuesto por el Art. 18°, 4to párrafo de la Constitución Política del Estado, cada Universidad es autónoma en su régimen normativo de gobierno, académico, administrativo y económico. Las Universidades se rigen por sus propios estatutos en el marco de la constitución y las leyes;

Que, la Ley Universitaria Ley N° 30220, en su Art. 8°, establece que el Estado reconoce la autonomía Universitaria, la autonomía inherente a las Universidades se ejerce de conformidad con lo establecido en la Constitución, la presente Ley y demás normativa aplicable, esta autonomía se manifiesta en los siguientes regímenes: Normativo, de Gobierno, Académico, Administrativo y Económico;

Que, el segundo párrafo del Art. 29° de la norma en marras establece que la Comisión Organizadora tiene a su cargo la aprobación del estatuto, reglamento y documentos de gestión académica y administrativa de la universidad formulados en los instrumentos de planeamiento, así como su conducción y dirección hasta que se constituyan los órganos de gobierno que, de acuerdo a la presente Ley le correspondan;

Que, mediante Carta N° 064-2025-OTI-UNAJ/HVC, de fecha 14 de febrero de 2025, el Jefe de la Oficina de Tecnologías de la Información, solicita la aprobación mediante acto resolutivo de la metodología Denominada "DESARROLLO Y MANTENIMIENTO DE SOFTWARE" según lo estipula la norma NTP-ISO/IEC 12207, que permitirá a la universidad estandarizar y proporcionar un enfoque sistemático para el desarrollo, operación y mantenimiento del software;

Que, el objetivo de la metodología "DESARROLLO Y MANTENIMIENTO DE SOFTWARE", es gestionar la implementación de una metodología de desarrollo y mantenimiento de software basado en la NTP-ISO/IEC 12207:2016, en la Universidad Nacional de Juliaca;

Que, el Pleno del Consejo de Comisión Organizadora de la UNAJ, en su Sesión Ordinaria de fecha 19 de febrero de 2025, mediante Acuerdo N° 148-2025-SO-CCO-UNAJ, acordó POR UNANIMIDAD: APROBAR la metodología denominada "DESARROLLO Y MANTENIMIENTO DE SOFTWARE", de la Universidad Nacional de Juliaca, previo informe técnico favorable de la Unidad de Planeamiento y Modernización;

Que, mediante Informe N° 044-2025-OPP/UPM-UNAJ, de fecha 26 de febrero de 2025, el Jefe de la Unidad de Planeamiento y Modernización, de la revisión de la metodología de "DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207", concluye que esta metodología, recoge los lineamientos generales de las normas técnicas y se alinea indirectamente a la acción estratégica institucional AEI.04.05 PROGRAMA DE GOBIERNO DIGITAL IMPLEMENTADO PARA LA COMUNIDAD UNIVERSITARIA del Plan Estratégico Institucional 2025-2030 de la Universidad Nacional de Juliaca;

En uso de las facultades y atribuciones conferidas por el Art. 18°, de la Constitución Política del Perú, la Ley N° 30220, Resolución Viceministerial N° 244-2021-MINEDU, modificado mediante Resolución Viceministerial N° 055-2022-MINEDU, modificado mediante Resolución Viceministerial N° 053-2023-MINEDU y el Estatuto de la UNAJ;

SE RESUELVE:

Artículo Primero.- APROBAR la metodología de "DESARROLLO Y MANTENIMIENTO DE SOFTWARE" de la Universidad Nacional de Juliaca, el mismo que como anexo forma parte del presente acto resolutivo a folios sesenta y uno (61).

Artículo Segundo.- NOTIFICAR el presente acto resolutivo a la Oficina de Tecnologías de la Información de la Universidad Nacional de Juliaca, a efecto de que pueda realizar la difusión de lo aprobado en el artículo primero del presente acto resolutivo a toda la comunidad universitaria.

Artículo Tercero.- DISPONER que la Oficina de Tecnologías de la Información, publique el presente acto resolutivo en el portal web de la institución (www.unaj.edu.pe).

Regístrese, comuníquese y cúmplase.



Abg. JOSÉ LUIS PACHECO CÁCERES
SECRETARIO GENERAL

DISTRIBUCIÓN:
VP Acad.,
VP de Invest.
DGA
OTI
Arch. /2025

KVCQ



DR. EDELFRE FLORES VELÁSQUEZ
PRESIDENTE
COMISIÓN ORGANIZADORA

I-71-0052-2025





UNIVERSIDAD NACIONAL DE JULIACA



Metodología

DESARROLLO Y MANTENIMIENTO SOFTWARE

NTP-ISO/IEC 12207

Oficina de Tecnologías de la
Información-OTI

Presidencia de Comisión Organizadora



Firmado digitalmente por:
VARGAS CANCINO Henry FAU
20448261272 soft
Motivo: Soy el autor del
documento
Fecha: 14/02/2025 12:36:01-0500

INDICE

INTRODUCCIÓN	5
1. OBJETIVO	6
2. ALCANCE	6
3. BASE LEGAL	6
4. NORMA TÉCNICA PERUANA NTP-ISO/IEC 12207	6
4.1. Categorías de los procesos del ciclo de vida	6
PROCESOS DE CONTEXTO DEL SISTEMA	7
PROCESOS ESPECÍFICOS DEL SOFTWARE	8
5. METODOLOGÍA DE DESARROLLO DE SOFTWARE	9
5.1. Scrum	9
5.1.1. Principios de Scrum	10
5.1.2. Aspectos de Scrum	11
5.1.3. Fases y Procesos de Scrum	17
5.1.4. Eventos de Scrum	21
5.1.5. Artefactos de Scrum	23
6. METODOLOGÍA DE MANTENIMIENTO DE SOFTWARE	25
6.1. Proceso de Solicitud y Aprobación de Cambios	25
6.2. Planificación y Programación de Mantenimientos	26
6.3. Implementación de Cambios y Actualizaciones	26
6.4. Despliegue y Soporte Post-Mantenimiento	27
7. CALIDAD DEL PRODUCTO DE SOFTWARE	28
7.1. Requisitos y Evaluación de Calidad de Sistemas y Software (SQuaRE)	28
ANEXOS	34
FLUJOGRAMA	58



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 3 de 61

INDICE DE FIGURAS

Figura 1: Flujo de Scrum	10
Figura 2: Principio de Scrum:.....	10
Figura 3: Organización Scrum	13
Figura 4: Justificación de negocio y el ciclo de vida del proyecto	14
Figura 5: Ciclo de Planificar, Hacer, Verificar y Actuar PDCA en Scrum	14
Figura 6: Ejemplo del Proceso de aprobación de cambios.....	15
Figura 7: Integración del cambio en Scrum	16
Figura 8: Fase de Inicio	18
Figura 9: Fase de planificación y estimación	18
Figura 10: Fase de Implementación	19
Figura 11: Scrumboard	19
Figura 12: Fase de revisión y retrospectiva	20
Figura 13: Fase de lanzamiento	20
Figura 14: Modelo de calidad del producto	29
Figura 15: Proyecto con Scrum.....	58
Figura 16: Ciclo de vida de software.....	59
Figura 17: Mantenimiento de software.....	60
Figura 18: Software en producción	61

INDICE DE TABLAS

Tabla 1: Minimizar riesgos por medio de Scrum	17
Tabla 2: Procesos fundamentales de Scrum.....	17
Tabla 3:Procesos adicionales de Scrum.....	21



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 4 de 61

APROBACIONES

Elaborado por:	Revisado por:	Aprobado por:
Ingº Cristian YAGUNO CANCINO	Ingº Henry VARGAS CANCINO	Dr. Edelfré FLORES VELÁSQUEZ
ASISTENTE DE LA OFICINA DE TECNOLOGIAS DE LA INFORMACIÓN	JEFE DE LA OFICINA DE TECNOLOGIAS DE LA INFORMACIÓN	PRESIDENTE DE LA COMISIÓN ORGANIZADORA



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 5 de 61

INTRODUCCIÓN

El DESARROLLO Y MANTENIMIENTO DE SOFTWARE son fundamentales en el ámbito académico y profesional, siendo crucial garantizar la calidad y eficiencia en su creación y gestión. La **Norma Técnica Peruana NTP-ISO/IEC 12207** proporciona un marco estructurado para gestionar el CICLO DE VIDA DEL SOFTWARE, desde su desarrollo hasta su mantenimiento. Esta norma establece procesos y directrices que permiten asegurar la calidad y el control en los proyectos informáticos.

Por otro lado, la metodología Scrum, dentro del enfoque ágil, se destaca por su flexibilidad, iteración y colaboración, permitiendo responder rápidamente a los cambios y necesidades del usuario. Su aplicación en proyectos de software favorece la entrega continua de valor y mejora la comunicación dentro de los equipos.

El propósito de esta metodología es mejorar el proceso de desarrollo, mantenimiento y adquisición de software basado en la NTP ISO/IEC 1207:2016 en la Universidad Nacional de Juliaca, con el fin de establecer una **METODOLOGÍA HÍBRIDA QUE OPTIMICE EL DESARROLLO Y MANTENIMIENTO DE SOFTWARE** en la institución. La combinación de ambas metodologías puede mejorar la calidad de los proyectos, facilitar la adaptabilidad a cambios y fomentar el uso de buenas prácticas en ingeniería de software.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 6 de 61

1. OBJETIVO

Gestionar la implementación de una metodología de desarrollo y mantenimiento de software basado en la NTP-ISO/IEC 12207:2016 en la Universidad Nacional de Juliaca.

2. ALCANCE

Este documento aborda la implementación de una metodología de desarrollo y mantenimiento de software basado en la NTP-ISO/IEC 12207:2016 en la Universidad Nacional de Juliaca.

3. BASE LEGAL

- Resolución Ministerial N°. 041-2017-PCM, aprueba el uso obligatorio de la Norma Técnica Peruana “NTP-ISO/IEC 12207:2016 Ingeniería de Software y Sistemas. Procesos del ciclo de vida del software. 3° Edición”, en todas las entidades integrantes de Sistema Nacional de Informática.
- Resolución Ministerial N° 004-2016-PCM, aprueba el uso obligatorio de la Norma Técnica Peruana “NTP ISO/IEC 27001:2014 Tecnología de la Información. Técnicas de Seguridad. Sistemas de Gestión de Seguridad de la Información, Requisito 2da Edición”, en todas las integrantes del Sistemas Nacional de Informática.
- ISO/IEC/IEEE 15288-2015 International Standard - Systems and software engineering - System life cycle processes.
- ISO/IEC 25010-2011 International Standard - Systems and software engineering- Systems and software Quality Requirements and Evaluation (SQuARE)- System and Software quality models.

4. NORMA TÉCNICA PERUANA NTP-ISO/IEC 12207

Esta norma establece un marco de trabajo común de los procesos de ciclo de vida del software, con terminología bien definida.

4.1. Categorías de los procesos del ciclo de vida

Esta norma agrupa las actividades que se pueden ejecutar durante el ciclo de vida de un sistema software en siete procesos. Estos procesos se dividen en diversas categorías, y cada categoría incluye un conjunto de actividades



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 7 de 61

específicas. A continuación, te proporciono un desglose más detallado de los procesos específicos según las categorías mencionadas:

PROCESOS DE CONTEXTO DEL SISTEMA

a. Procesos de Contratación

Estos procesos definen actividades necesarias para establecer un acuerdo entre dos organizaciones.

- Procesos de Adquisición.
- Proceso de Suministro.

b. Procesos Organizacionales de Habilitación de Proyectos

Administran la capacidad de la organización para adquirir y suministrar productos o servicios a través de la iniciación, soporte y control de los proyectos. Los procesos Organizacionales de habilitación de proyecto constan con los siguientes procesos:

- Proceso de Gestión del Modelo de Ciclo de Vida.
- Proceso de Gestión de la Infraestructura.
- Proceso de Gestión del Portafolio del Proyecto.
- Proceso de Gestión de los Recursos Humanos.
- Proceso de Gestión de la Calidad.

c. Procesos de Proyecto

Los principios relacionados con estos procesos se pueden aplicar en cualquier área de gestión de una organización. Existen dos categorías de Procesos de Proyecto.

Los Procesos de Gestión de Proyectos se utilizan para planificar, para evaluar el logro real y el progreso frente a los planes y para controlar la ejecución del proyecto hasta cumplimiento:

- Proceso de Planificación del Proyecto.
- Proceso de Evaluación y Control del Proyecto.

Los Procesos de Soporte del Proyecto dan soporte a los objetivos especializados de la gestión:

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 8 de 61

- Proceso de Gestión de Decisiones.
- Proceso de Gestión del Riesgo.
- Proceso de Gestión de la Configuración.
- Proceso de gestión de la Información.
- Proceso de Medición.

d. Procesos Técnicos

Definen las actividades que habilitan las funciones organizacionales del proyecto con el fin de optimizar los beneficios y reducir los riesgos que presentan debido a las decisiones y las acciones técnicas. Los procesos Técnicos constan de los siguientes procesos:

- Proceso de Definición de Requisitos de las Partes Interesadas.
- Proceso de Análisis de Requisitos del Sistema.
- Proceso de Diseño Arquitectura del Sistema.
- Proceso de Implementación.
- Proceso de Integración del Sistema.
- Proceso de Pruebas de Calificación del Sistema.
- Proceso de Instalación de Software.
- Proceso de Soporte de la Aceptación del Software.
- Proceso de Operación del Software.
- Proceso de Mantenimiento del Software.
- Proceso de Retiro de Software.

PROCESOS ESPECÍFICOS DEL SOFTWARE

e. Procesos de Implementación del software

Se utilizan para producir un elemento del sistema especificado implementado en el software. Estos procesos transforman comportamiento especificado, interfaces y restricciones de implementación en acciones de software de implementación, que dan como resultado un elemento del sistema que satisface los requisitos derivados de los requisitos del sistema.

- Proceso de Implementación del Software.
- Proceso de Análisis de Requisitos del Software.

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 9 de 61

- Proceso de Diseño Arquitectural del Software.
- Proceso de Diseño Detallado del Software.
- Proceso de Construcción del Software.
- Proceso de Integración del Software.
- Proceso de Pruebas de Calificación del Software.

f. Procesos de Soporte del Software.

Proporcionan un conjunto específico y enfocado de actividades para realizar un proceso de software especializado. Ayuda al proceso de implementación de Software como una parte integral con un propósito diferente, contribuyendo al éxito y a la calidad del proyecto de software.

- Proceso de Gestión de la Documentación del Software.
- Proceso de Gestión de la Configuración del Software.
- Proceso de Aseguramiento de la Calidad del Software.
- Proceso de Verificación del Software.
- Proceso de Validación del Software.
- Proceso de Revisión del Software.
- Proceso de Auditoria del Software.
- Proceso de Resolución de Problemas del Software.

g. Procesos de Reutilización del Software

Consta de tres procesos que dan soporte a la capacidad de una organización para reutilizar los elementos de software a través de los límites del proyecto.

- Proceso de Ingeniería de Dominio.
- Proceso de Gestión de Programas de Reutilización.
- Proceso de Gestión de Activos de Reutilización.

5. METODOLOGÍA DE DESARROLLO DE SOFTWARE

5.1. Scrum

Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos. Es un framework adaptable, iterativo, rápido, flexible y



eficaz, diseñado para ofrecer un valor considerable en forma rápida a lo largo del proyecto.

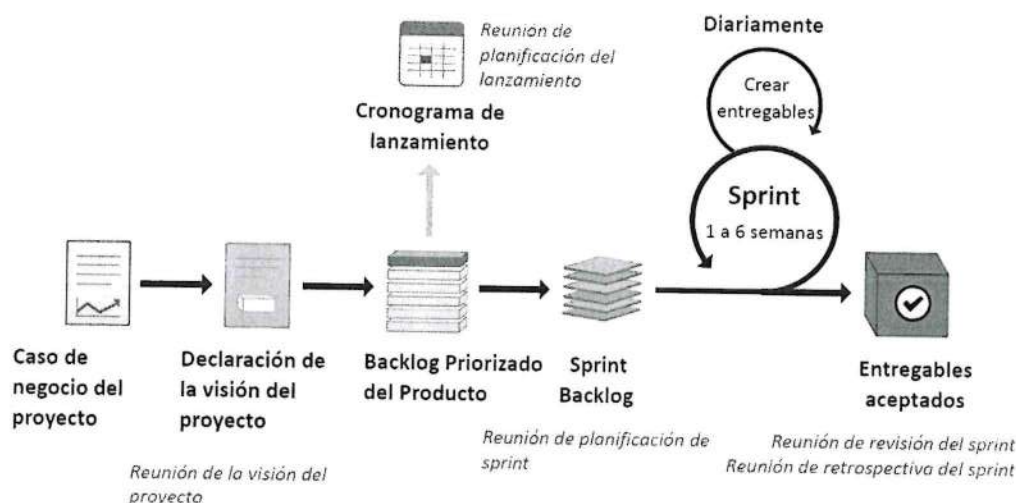


Figura 1: Flujo de Scrum

5.1.1. Principios de Scrum

Los principios de Scrum se pueden aplicar a cualquier tipo de proyecto en cualquier organización y deben cumplirse a fin de garantizar la aplicación efectiva del framework de Scrum. Los seis principios de Scrum que se presentan son los siguientes:

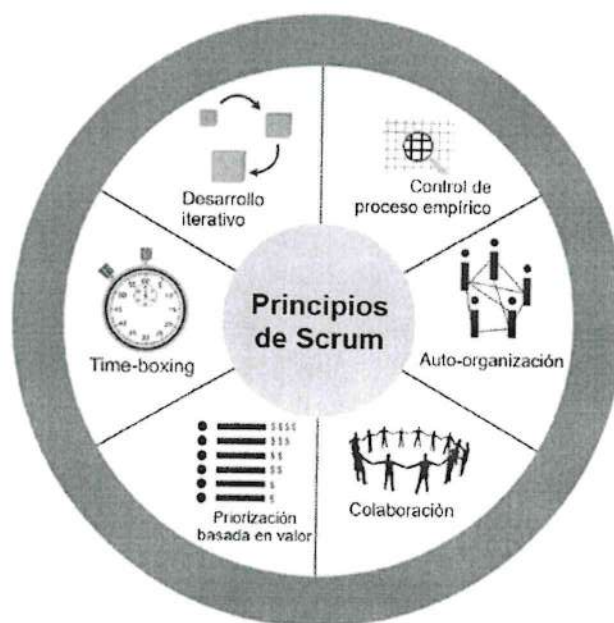


Figura 2: Principio de Scrum:

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 11 de 61

- **Control del proceso empírico:** Este principio enfatiza la filosofía central de Scrum con base a las tres ideas principales de transparencia, inspección y adaptación.
- **Auto - organización:** Este principio se enfoca en los trabajadores de hoy en día, que entregan un valor considerablemente mayor cuando se auto-organizan, lo cual resulta en equipos que poseen un gran sentido de compromiso y responsabilidad; a su vez, esto produce un ambiente innovador y creativo que es más propicio para el crecimiento.
- **Colaboración:** Este principio se centra en las tres dimensiones básicas relacionadas con el trabajo colaborativo: conocimiento, articulación y apropiación. También fomenta la gestión de proyectos como un proceso de creación de valor compartido con equipos que trabajan e interactúan conjuntamente para ofrecer el mayor valor.
- **Priorización basada en valor:** Este principio pone de relieve el enfoque de Scrum para ofrecer el máximo valor de negocio, desde el principio del proyecto hasta su conclusión.
- **Time-boxing:** Este principio describe cómo el tiempo se considera una restricción limitante en Scrum, y cómo este se utiliza para ayudar a manejar eficazmente la planificación y ejecución del proyecto. Los elementos del time boxing en Scrum incluyen sprints, Daily Standups, reuniones de planificación del sprint y reuniones de revisión del sprint.
- **Desarrollo iterativo:** Este principio define el desarrollo iterativo y hace énfasis en cómo gestionar mejor los cambios y crear productos que satisfagan las necesidades del cliente. También delinea las responsabilidades del Product Owner y las de la organización relacionadas con el desarrollo iterativo.

5.1.2. Aspectos de Scrum

Los aspectos de Scrum deben ser abordados y administrados a lo largo de un proyecto Scrum en la Universidad Nacional de Juliaca.

a. Organización

Los roles de Scrum se dividen en dos amplias categorías:



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 12 de 61

Roles Centrales: Son Aquellos que se requieren obligadamente para crear el producto o servicio del proyecto.

- **Product Owner:**

- Responsable de lograr el máximo valor empresarial para el proyecto.
- Responsable de la articulación de requisitos del cliente.
- Mantiene la justificación del negocio para el proyecto.
- Representa la voz del cliente.

- **Scrum Master:**

- Asegura que el Equipo Scrum cuente con un ambiente propicio para completar el proyecto con éxito.
- Guía, facilita y enseña las prácticas de Scrum a todos los involucrados en el proyecto.
- Elimina los impedimentos que pueda tener el equipo.
- Se asegura de que se estén siguiendo los procesos de Scrum.

- **Equipo Scrum:**

- Responsables de entender los requisitos especificados por el Product Owner.
- Crea los entregables del proyecto.

Roles no centrales: Son los que no son necesariamente obligatorios para el proyecto Scrum, y estos pueden incluir a miembros de los equipos que estén interesados en el proyecto.

- **Stakeholder(s):**

- Es un término colectivo que incluye a clientes, usuarios y patrocinadores.
- Interactúan frecuentemente con el equipo principal de Scrum, e influyen en el proyecto a lo largo de su desarrollo.
- Lo más importante es que el proyecto produzca beneficios colaborativos para los stakeholders.

- **Scrum Guidance Body (SGB):**

- Es un rol opcional, que generalmente consiste en un conjunto de documentos y/o un grupo de expertos que normalmente están



involucrados en la definición de los objetivos relacionados con la calidad, las regulaciones gubernamentales, la seguridad y otros parámetros claves de la organización.

- El SGB guía el trabajo llevado a cabo por el Product Owner, el Scrum Master y el Equipo Scrum.

- **Vendedores:**

- Los vendedores incluyendo a individuos u organizaciones externas, ofrecen productos y/o servicios que no están dentro de las competencias centrales de la organización del proyecto.

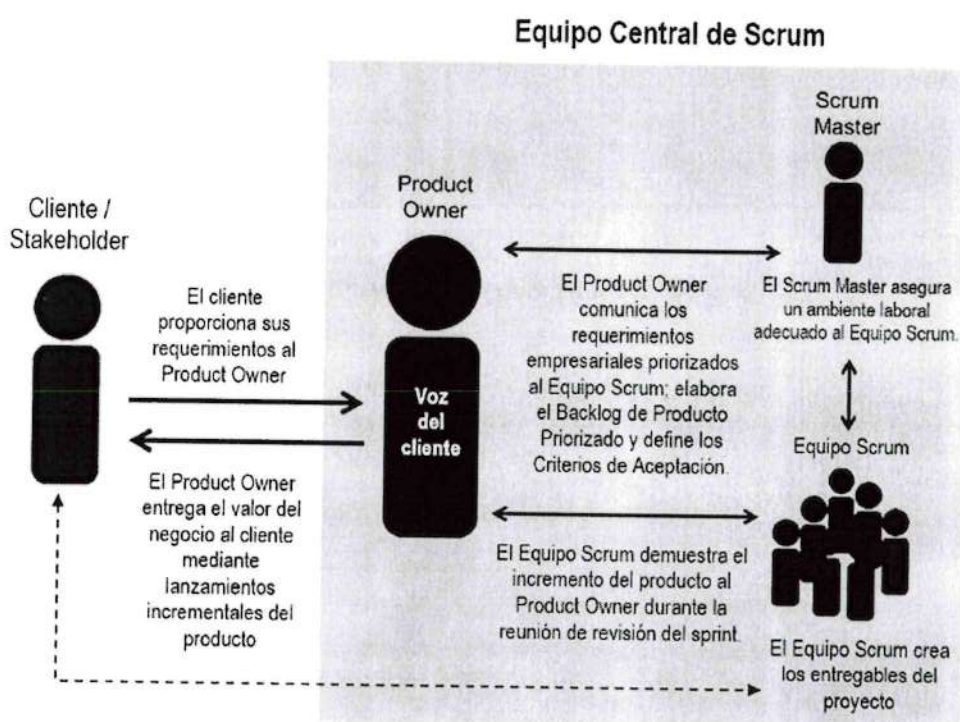


Figura 3: Organización Scrum

b. Justificación del Negocio

En Scrum, la justificación del negocio se basa en el concepto de entrega impulsada por el valor (Valuedriven Delivery). Es importante señalar que, si bien el Product Owner es el responsable principal de la justificación del negocio, otros miembros del equipo también contribuyen considerablemente.



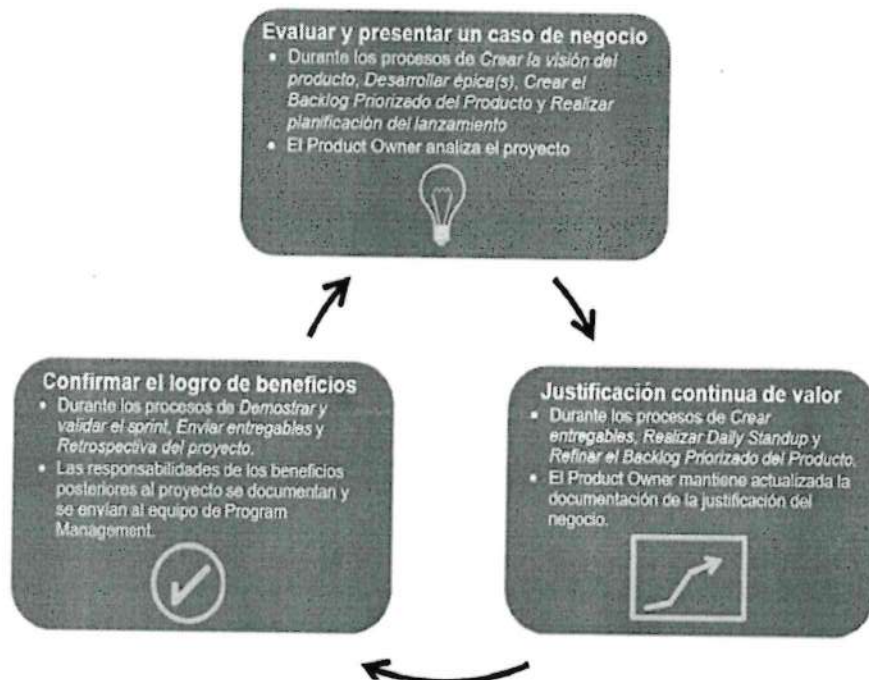


Figura 4: Justificación de negocio y el ciclo de vida del proyecto

c. Calidad

En Scrum, la calidad se define como la capacidad con la que cuenta el producto o los entregables para cumplir con los criterios de aceptación y de alcanzar el valor de negocio que el cliente espera.

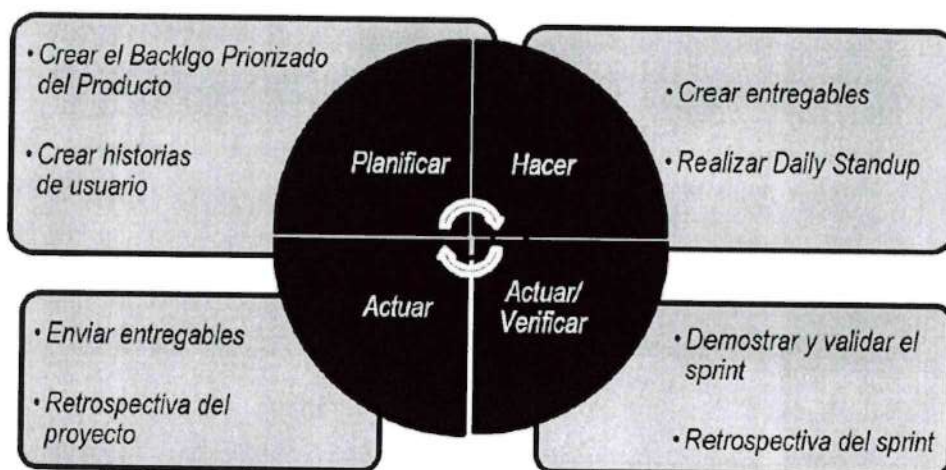


Figura 5: Ciclo de Planificar, Hacer, Verificar y Actuar PDCA en Scrum

d. Cambio

Cada proyecto, independientemente del método o framework que se utilice, está expuesto a cambios.

Un principio fundamental de Scrum es su reconocimiento de que:

- Los stakeholders (clientes, usuarios y patrocinadores) cambian de opinión acerca de lo que quieren y lo que necesitan durante un proyecto (a esto se le conoce en ocasiones como: “requisitos volátiles” o requirements churn).
- Que es muy difícil, si no es que imposible, que los stakeholders definan todos los requisitos al inicio del proyecto.

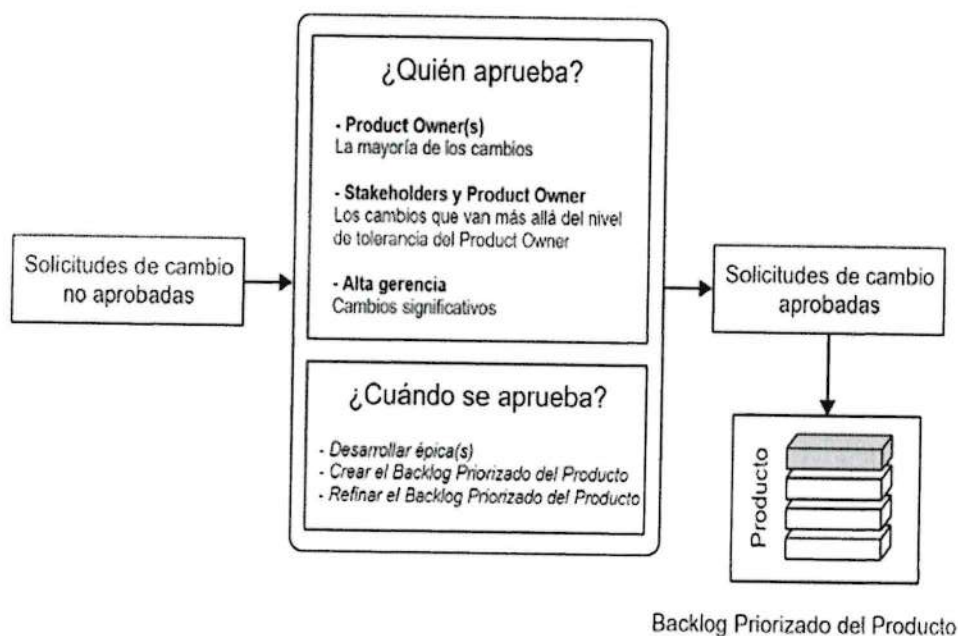


Figura 6: Ejemplo del Proceso de aprobación de cambios



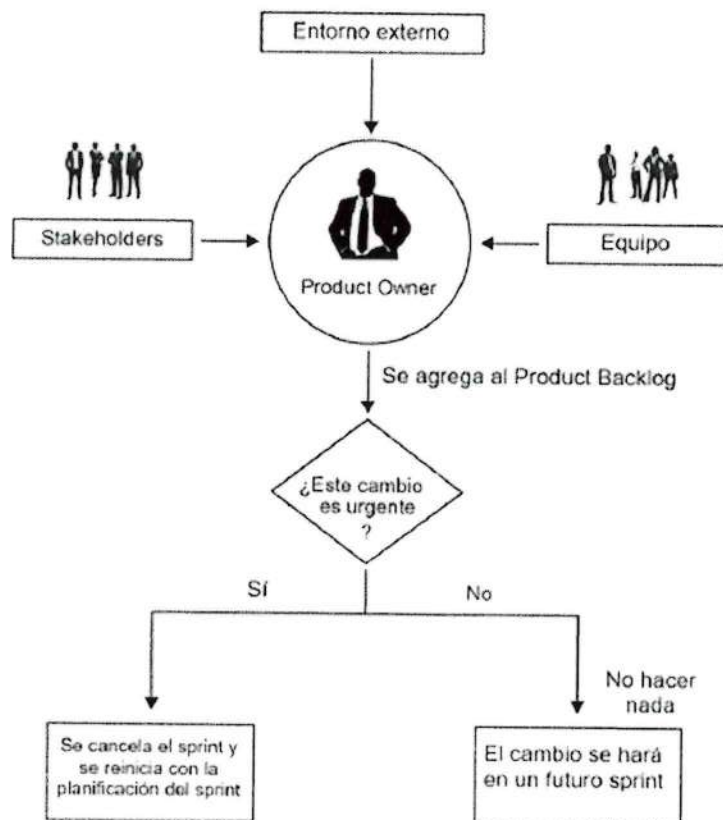


Figura 7: Integración del cambio en Scrum

e. Riesgo

El riesgo se define como un evento incierto o serie de eventos que pueden afectar los objetivos de un proyecto y pueden contribuir a su éxito o fracaso.

Procedimiento para la gestión de riesgos:

1. **Identificación de riesgos:** Utilizar diversas técnicas para identificar todo los riesgos potenciales.
2. **Evaluación de riesgos:** Evaluar y estimar los riesgos identificados.
3. **Priorización de riesgos:** Dar prioridad al riesgo que habrá de incluirse en el Backlog priorizado del Producto.
4. **Mitigación de riesgos:** Desarrollar de una estrategia adecuada para hacer frente a un riesgo.
5. **Comunicación de riesgos:** Comunicar a los stakeholders apropiados los resultados de los primeros cuatro pasos de la gestión de riesgos y determinar su percepción respecto a eventos inciertos.



Minimizar riesgos por medio de Scrum				
La flexibilidad reduce el riesgo relacionado al entorno empresarial.	La retroalimentación constante reduce el riesgo relacionado a las expectativas.	La propiedad del equipo reduce la estimación de riesgo.	La transparencia reduce el riesgo de no detección.	La entrega iterativa reduce el riesgo de inversión.

Tabla 1: Minimizar riesgos por medio de Scrum

5.1.3. Fases y Procesos de Scrum

Los procesos de Scrum abordan las actividades específicas y el flujo de un proyecto de Scrum. En total hay 19 procesos fundamentales de Scrum que aplican a todos los proyectos.

FASE	PROCESOS FUNDAMENTALES DE SCRUM
Inicio	1. Crear la visión del proyecto. 2. Identificar al Scrum Master y Stakeholder(s). 3. Formar Equipos Scrum. 4. Desarrollar épicas(s) 5. Crear el Blacklog priorizado del producto. 6. Realizar la planificación de lanzamiento.
Planificación y estimación	7. Crear historias de usuario. 8. Estimar historias de usuario. 9. Comprometer historias de usuario. 10. Identificar tareas. 11. Estimar tareas. 12. Crear el Sprint Backlog.
Implementación	13. Crear entregables. 14. Realizar Daily Standup. 15. Refinar el Backlog Priorizado del Producto.
Revisión y retrospectiva	16. Demostrar y validar el sprint. 17. Retrospectiva del Sprint.
Lanzamiento	18. Enviar entregables. 19. Retrospectiva del proyecto.

Tabla 2: Procesos fundamentales de Scrum



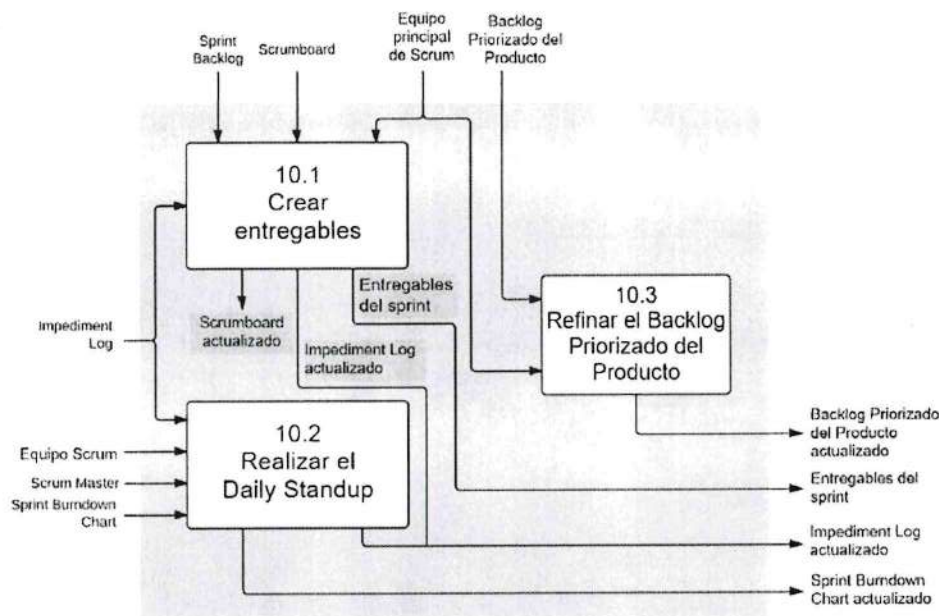


Figura 10: Fase de Implementación

En Scrum, la transparencia proviene de las herramientas visiblemente abiertas tales como el Scrumboard, donde se muestra el avance del equipo. El equipo utiliza un Scrumboard para planificar y dar seguimiento al progreso durante cada sprint. El tablero contiene cuatro columnas para indicar el progreso de las tareas estimadas para el sprint: una columna “por hacer” para las tareas que aún no se inician; una columna “en progreso” para las tareas que ya iniciaron, pero no se han concluido; una columna “en prueba” para las tareas concluidas pero que están en proceso de evaluación y una columna de “terminado” para las tareas que se han concluido y evaluado satisfactoriamente. Al principio de cada sprint, todas las tareas se colocan en la columna “por hacer” y avanzan según su progreso.

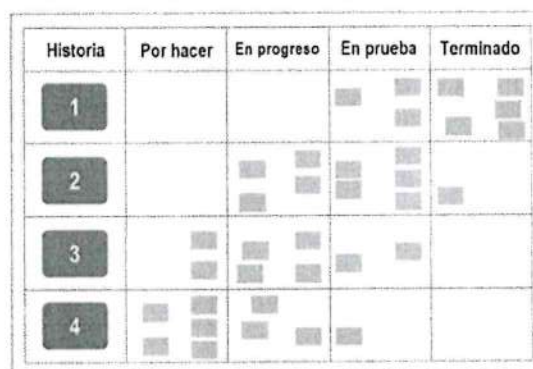


Figura 11: Scrumboard

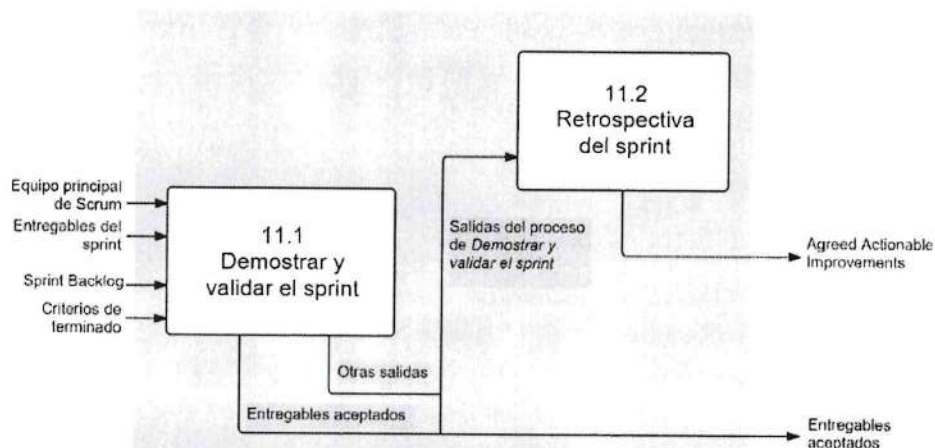


Figura 12: Fase de revisión y retrospectiva

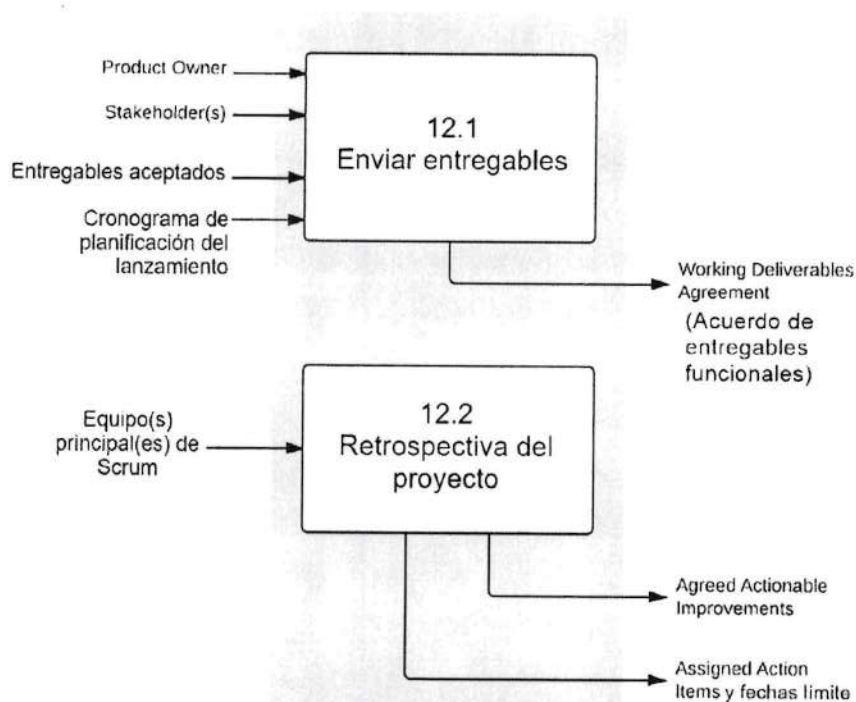


Figura 13: Fase de lanzamiento

Para proyectos Scrum a grande escala y en las empresas que requieren de la coordinación entre múltiples equipos, existen 8 procesos adicionales de Scrum.

Aplicabilidad	Procesos adicionales de Scrum
Scrum para grandes proyectos	1. Crear componentes de grandes proyectos. 2. Realizar y coordinar sprints. 3. Preparar el lanzamiento de grandes proyectos.

Aplicabilidad	Procesos adicionales de Scrum
Scrum para la empresa	1. Crear componentes de programa o portafolio. 2. Revisar y actualizar el Scrum Guidance Body. 3. Crear y refinar el backlog del programa o portafolio 4. Coordinar los componentes del programa o portafolio. 5. Retrospectiva de los lanzamientos del programa o portafolio.

Tabla 3: Procesos adicionales de Scrum

5.1.4. Eventos de Scrum

El Sprint: Son el corazón de Scrum, donde las ideas se convierten en valor. Son eventos de duración de un mes o menos para crear consistencia. Un nuevo Sprint comienza inmediatamente después de la conclusión de Sprint anterior.

Todo el trabajo necesario para lograr el Objetivo del Producto, incluido la Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective, ocurre dentro de los Sprints.

Durante el Sprint:

- No se realizan cambios que pongan en peligro el Objetivo del Sprint;
- La calidad no disminuye;
- El Product Backlog se refina según sea necesario; y,
- El alcance se puede aclarar y renegociar con el Product Owner a medida que se aprende más.

Sprint Planning: Inicia el Sprint al establecer el trabajo que se realizará para el Sprint. El Scrum Team crea este plan resultante mediante trabajo colaborativo.

El Product Owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Product Backlog y cómo se relacionan con el Objetivo del Producto. El Scrum Team también puede invitar a otras personas a asistir a la Sprint Planning para brindar asesoramiento.

La Sprint Planning aborda los siguientes temas:



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 22 de 61

- **Tema 01: ¿Por qué es valioso este Sprint?**

El Product Owner propone cómo el producto podría Incrementar su valor y utilidad en el Sprint actual. Luego, todo el Scrum Team colabora para definir un Objetivo del Sprint que comunica por qué el Sprint es valioso para los interesados. El Objetivo del Sprint debe completarse antes de que termine la Sprint Planning.

- **Tema 02: ¿Qué se puede hacer en este Sprint?**

A través de una conversación con el Product Owner, los Developers seleccionan elementos del Product Backlog para incluirlos en el Sprint actual. El Scrum Team puede refinar estos elementos durante este proceso, lo que aumenta la comprensión y la confianza.

Seleccionar cuánto se puede completar dentro de un Sprint puede ser un desafío. Sin embargo, cuanto más sepan los Developers sobre su desempeño pasado, su capacidad actual y su Definición de Terminado, más confiados estarán en sus pronósticos para el Sprint.

- **Tema 03: ¿Cómo se realizará el trabajo elegido?**

Para cada elemento del Product Backlog seleccionado, los Developers planifican el trabajo necesario para crear un Increment que cumpla con la Definición de Terminado. A menudo, esto se hace descomponiendo los elementos del Product Backlog en elementos de trabajo más pequeños de un día o menos. La forma de hacerlo queda a criterio exclusivo de los Developers. Nadie más les dice cómo convertir los elementos del Product Backlog en Increments de valor.

Daily Scrum: Es inspeccionar el progreso hacia el objetivo del sprint y adapta el Sprint Backlog según sea necesario, ajustando el trabajo planificado entrante.

La Daily Scrum es un evento de 15 minutos para los developers de Scrum Team. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del Sprint. Si el Product Owner o Scrum



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 23 de 61

Master están trabajando activamente en elementos del Sprint Backlog, participan como Developers.

Sprint Review: El propósito de la Sprint Review es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El Scrum Team presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el Objetivo del Producto.

La Sprint Review es el penúltimo evento del Sprint y tiene un límite de tiempo de máximo cuatro horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

Sprint Retrospective: El propósito de la Sprint Retrospective es planificar formas de aumentar la calidad y la efectividad.

La Sprint Retrospective concluye el Sprint. Tiene un tiempo limitado a máximo tres horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

5.1.5. Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor que los mismos están diseñados para maximizar la transparencia de la información clave. Por lo tanto, todas las personas que los inspeccionan tienen la misma base de adaptación.

- **Product Backlog:** Es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente de trabajo realizado por el Scrum Team.

Los elementos del Product Backlog que el Scrum Team puede dar por terminados dentro de un Sprint se consideran preparados para ser seleccionados en un evento de Sprint Planning. Suelen adquirir este grado de transparencia tras las actividades de refinamiento. El refinamiento del Product Backlog es el acto de dividir y definir aún más los elementos del Product Backlog en elementos más pequeños y precisos. Esta es una actividad continua para agregar detalles, como una



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 24 de 61

descripción, orden y tamaño. Los atributos suelen variar según el ámbito del trabajo.

El Objetivo del Producto es el objetivo a largo plazo del Scrum Team. Ellos deben cumplir (o abandonar) un objetivo antes de asumir el siguiente objetivo.

- **Sprint Backlog:** Se compone del Objetivo del Sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan de acción para entregar el Increment (cómo).

El Sprint Backlog es un plan realizado por y para los Developers. Es una imagen muy visible y en tiempo real del trabajo que los Developers planean realizar durante el Sprint para lograr el Objetivo del Sprint. En consecuencia, el Sprint Backlog se actualiza a lo largo del Sprint a medida que se aprende más. Debe tener suficientes detalles para que puedan inspeccionar su progreso en la Daily Scrum.

El Objetivo del Sprint se crea durante el evento Sprint Planning y se agrega al Sprint Backlog. Mientras los Developers trabajan durante el Sprint, tienen en mente el Objetivo del Sprint. Si el trabajo resulta ser diferente de lo que esperaban, colaboran con el Product Owner para negociar el alcance del Sprint Backlog dentro del Sprint sin afectar el Objetivo del Sprint.

- **Increment:** Es un peldaño concreto hacia el Objetivo del Producto. Cada Increment se suma a todos los Increments anteriores y se verifica minuciosamente, lo que garantiza que todos los Increments funcionen juntos. Para proporcionar valor, el Increment debe ser utilizable.

Se pueden crear múltiples Increments dentro de un Sprint. La suma de los Increments se presenta en la Sprint Review apoyando así el empirismo. Sin embargo, se puede entregar un Increment a los interesados antes del final del Sprint. La Sprint Review nunca debe considerarse una puerta para liberar valor.

La Definición de Terminado crea transparencia al brindar a todos un entendimiento compartido de qué trabajo se completó como parte del Increment. Si un elemento del Product Backlog no cumple con la



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 25 de 61

Definición de Terminado, no se puede publicar ni presentar en la Sprint Review. En su lugar, vuelve al Product Backlog para su consideración futura.

6. METODOLOGÍA DE MANTENIMIENTO DE SOFTWARE

6.1. Proceso de Solicitud y Aprobación de Cambios

El proceso de solicitud y aprobación de cambios establece el procedimiento para manejar las solicitudes de modificaciones, mejoras o correcciones del software. Esta es la fase donde se detectan las necesidades de cambio y se formaliza el proceso de modificación. El proceso incluye:

1. Recepción de Solicitudes de Cambio:

- Las solicitudes pueden ser generadas por los usuarios, el equipo de soporte, o a través de auditorías internas. Estas solicitudes incluyen defectos, mejoras, adaptaciones o actualizaciones.
- Las solicitudes deben ser documentadas de manera clara, detallada y con una descripción precisa del cambio solicitado.

2. Evaluación del Cambio:

- El equipo de mantenimiento analiza la solicitud para determinar el impacto del cambio (técnico, funcional, y de negocio), su viabilidad, urgencia y costo.
- Se evalúa si el cambio es necesario, si afectará otros módulos o sistemas y si se ajusta a las prioridades y los objetivos del proyecto.

3. Aprobación o Rechazo:

- Las solicitudes de cambio deben ser aprobadas por las partes interesadas relevantes (por ejemplo, el gerente de proyectos, el cliente, etc.).
- Las decisiones deben quedar documentadas, y si un cambio es rechazado, también se deben comunicar las razones.
- Este proceso asegura que solo los cambios validados y necesarios sean implementados en el sistema, evitando modificaciones innecesarias o riesgosas.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 26 de 61

6.2. Planificación y Programación de Mantenimientos

La planificación y programación del mantenimiento son fundamentales para garantizar que los cambios y actualizaciones se realicen de manera ordenada, minimizando el impacto en el sistema y los usuarios. Esta etapa incluye:

1. Definición de Tareas de Mantenimiento:

- Con base en las solicitudes aprobadas, se elaboran los planes de trabajo específicos, que detallan las tareas a realizar (por ejemplo, corrección de errores, actualizaciones, mejoras, etc.).
- Se determina la complejidad y el alcance de cada tarea.

2. Estimación de Recursos y Tiempo:

- Se deben estimar los recursos humanos, técnicos y financieros necesarios para llevar a cabo las actividades de mantenimiento.
- Se establecen los tiempos estimados para cada tarea y se definen los plazos generales para la ejecución del mantenimiento.

3. Programación de Tareas:

- Las tareas deben ser programadas teniendo en cuenta la disponibilidad de recursos, las prioridades de los cambios y las ventanas de mantenimiento (si es necesario, para minimizar la interrupción de los servicios).
- Se debe considerar la programación de actividades de prueba, validación y despliegue.

4. Comunicación y Aprobación del Plan:

- El plan debe ser comunicado a todas las partes interesadas (usuarios, equipos de desarrollo, soporte, etc.) y debe ser aprobado antes de proceder con la implementación.

La planificación efectiva es crucial para la eficiencia del proceso de mantenimiento y para asegurar que los cambios se realicen en el momento adecuado y sin generar riesgos innecesarios.

6.3. Implementación de Cambios y Actualizaciones

La implementación de cambios y actualizaciones se refiere a la ejecución de las tareas de mantenimiento planificadas. En esta fase, se realizan las modificaciones en el software y se verifica su impacto en el sistema.

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 27 de 61

1. Desarrollo de Cambios:

- Los cambios pueden incluir correcciones de errores, nuevas funcionalidades o ajustes en la configuración del software.
- Los desarrolladores implementan estos cambios según las especificaciones aprobadas.

2. Pruebas de Validación:

- Antes de desplegar los cambios, se deben realizar pruebas para asegurarse de que los cambios no afectan negativamente al sistema y cumplen con los requisitos.
- Las pruebas incluyen pruebas unitarias, de integración y de aceptación (según corresponda).
- Las pruebas deben realizarse en un entorno controlado que simule el entorno de producción.

3. Revisión de Calidad y Documentación:

- Los cambios deben ser revisados por el equipo de calidad para asegurar que se ajusten a los estándares definidos.
- Se debe actualizar la documentación técnica y de usuario del software para reflejar los cambios realizados.

4. Aprobación Final:

- Una vez completadas las pruebas y validaciones, los cambios son aprobados para su despliegue en producción.
- Esta etapa es clave para garantizar que los cambios no introduzcan nuevos errores y que el software mantenga su estabilidad y funcionalidad.

6.4. Despliegue y Soporte Post-Mantenimiento

Una vez que los cambios han sido implementados y validados, se realiza el despliegue del software actualizado en el entorno de producción. Además, se ofrece soporte para resolver cualquier inconveniente post-mantenimiento.

1. Despliegue Controlado:

- El despliegue de los cambios debe realizarse en un entorno controlado, con procedimientos que minimicen el riesgo de interrupción.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 28 de 61

- Si es posible, se debe realizar un despliegue gradual para verificar que el sistema funcione correctamente antes de una implementación masiva.

2. Monitoreo Post-Despliegue:

- Después del despliegue, el equipo de mantenimiento debe monitorear el sistema para detectar problemas inesperados y corregirlos rápidamente.
- El monitoreo incluye el rendimiento del sistema, la seguridad y la satisfacción de los usuarios.

3. Soporte Post-Mantenimiento:

- Se debe ofrecer soporte a los usuarios para resolver cualquier problema relacionado con los cambios implementados.
- También se puede proporcionar formación adicional si los cambios incluyen nuevas funcionalidades.

4. Cierre del Mantenimiento:

- Una vez que el sistema esté funcionando correctamente y los usuarios estén satisfechos con el resultado, se cierra el ciclo de mantenimiento correspondiente.
- Se documentan las lecciones aprendidas y los cambios realizados para futuras referencias.

7. CALIDAD DEL PRODUCTO DE SOFTWARE

7.1. Requisitos y Evaluación de Calidad de Sistemas y Software (SQuaRE)

ISO/IEC 25010:2011 es un estándar que proporciona pautas y recomendaciones para evaluar la calidad de los productos de software. Define un conjunto de características de calidad y sub-características que son esenciales para evaluar la calidad general de un producto de software.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por 8 características principales que tienen que ser evaluados al medir la calidad de software o sistema, se muestran en la siguiente Figura N° 9:



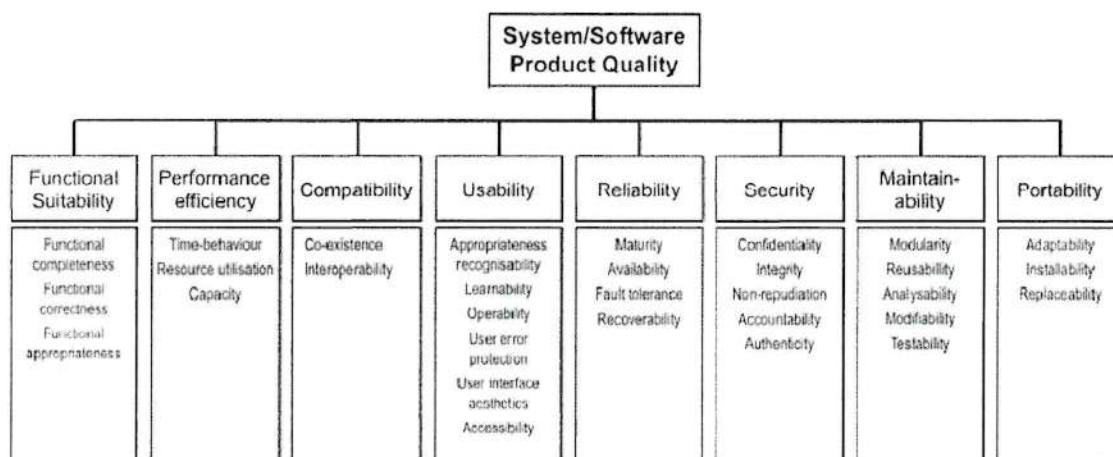


Figura 14: Modelo de calidad del producto

1. Adecuación Funcional (Functional Suitability)

Esta característica se refiere a la capacidad del software para cumplir con los requisitos funcionales definidos, proporcionando las funcionalidades adecuadas.

Sub-características:

- **Completitud de las funcionalidades (Functional Completeness):**
La medida en que el software proporciona todas las funcionalidades requeridas por el usuario y por los requisitos del sistema.
- **Exactitud (Functional Correctness):** Se refiere al grado en que el software cumple con las especificaciones y genera los resultados correctos.
- **Conformidad (Functional Appropriateness):** Evaluación de si las funcionalidades implementadas son apropiadas para las tareas que se deben realizar, es decir, si la solución es adecuada a las necesidades del usuario.

2. Eficiencia en el Rendimiento (Performance Efficiency)

Evalúa el uso eficiente de los recursos del sistema, asegurando que el software funcione correctamente bajo diferentes condiciones de carga y uso.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 30 de 61

Sub-características:

- **Comportamiento temporal (Time Behavior):** El tiempo de respuesta, la latencia y el rendimiento del sistema en general. Se mide la rapidez con la que el sistema responde a las acciones del usuario o realiza una tarea.
- **Uso de recursos (Resource Utilization):** La cantidad de recursos (memoria, CPU, red, etc.) que utiliza el sistema mientras realiza sus funciones.
- **Capacidad (Capacity):** La habilidad del sistema para manejar grandes volúmenes de datos o usuarios, sin deteriorar su rendimiento.

3. Compatibilidad (Compatibility)

La compatibilidad mide la capacidad del software para funcionar con otros sistemas o plataformas y su interacción sin conflictos.

Sub-características:

- **Coexistencia (Co-existence):** La capacidad del software para operar simultáneamente con otros sistemas sin que interfiera en su rendimiento ni afecte su funcionamiento.
- **Interoperabilidad (Interoperability):** La capacidad de interactuar con otros sistemas y software, compartir datos e intercambiar funcionalidades sin problemas.

4. Usabilidad (Usability)

Se refiere a la facilidad con la que los usuarios pueden aprender a utilizar el software, interactuar con él y lograr sus objetivos de forma efectiva.

Sub-características:

- **Facilidad de aprendizaje (Learnability):** La facilidad con que un nuevo usuario puede aprender a utilizar el sistema.
- **Eficiencia de uso (Efficiency of Use):** La capacidad del sistema para permitir que los usuarios realicen tareas con eficacia, en el menor tiempo posible.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 31 de 61

- Satisfacción del usuario (User Satisfaction): El grado en que el sistema satisface las expectativas y necesidades del usuario, proporcionando una experiencia positiva.
- Facilidad de operación (Operability): La facilidad con la que un usuario puede operar y manejar el sistema sin cometer errores.
- Accesibilidad (Accessibility): Qué tan accesible es el software para personas con discapacidades, asegurando que todos los usuarios puedan utilizarlo independientemente de sus habilidades.

5. Fiabilidad (Reliability)

La fiabilidad mide la capacidad del software para mantener un nivel de desempeño estable y correcto durante un período determinado bajo condiciones predefinidas.

Sub-características:

- **Madurez (Maturity):** El grado en que el sistema ha sido probado y utilizado, y la probabilidad de que falle en condiciones normales de operación.
- **Tolerancia a fallos (Fault Tolerance):** La capacidad del sistema para continuar funcionando incluso si ocurren fallos o errores en el sistema.
- **Recuperación (Recoverability):** La capacidad del software para recuperar su funcionamiento normal después de que se haya producido un fallo o error.
- **Disponibilidad (Availability):** La capacidad del software para estar operativo y accesible en todo momento, sin interrupciones prolongadas.

6. Seguridad (Security)

La seguridad abarca las medidas implementadas para proteger el sistema y los datos de accesos no autorizados, corrupción o mal uso.

Sub-características:

- **Confidencialidad (Confidentiality):** Asegura que solo las personas o sistemas autorizados puedan acceder a la información del software.

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 32 de 61

- **Integridad (Integrity):** La capacidad de garantizar que los datos y las operaciones no se alteren de manera no autorizada, asegurando la exactitud y consistencia de la información.
- **Disponibilidad (Availability):** Se refiere a la disponibilidad de los recursos y datos para los usuarios autorizados cuando lo necesiten, sin que existan interrupciones o fallos.
- **No repudio (Non-repudiation):** Asegura que no se pueda negar la autoría o la realización de ciertas acciones (como la modificación de datos o el acceso a sistemas).
- **Control de acceso (Access Control):** Mecanismos para restringir el acceso a los datos y funciones a usuarios o sistemas no autorizados, utilizando autenticación, autorización y auditoría.

7. **Mantenibilidad (Maintainability)**

La mantenibilidad evalúa qué tan fácil es realizar modificaciones en el sistema para corregir defectos, mejorar el rendimiento o adaptarlo a nuevas necesidades.

Sub-características:

- **Modificabilidad (Modifiability):** La facilidad con la que se pueden realizar cambios en el software sin causar efectos adversos o introducir errores nuevos.
- **Analizabilidad (Analyzability):** La facilidad con la que se puede analizar y diagnosticar problemas en el software.
- **Testabilidad (Testability):** La facilidad con la que se pueden realizar pruebas para verificar el funcionamiento correcto del software.
- **Estabilidad (Stability):** La capacidad del sistema para mantenerse operativo y sin problemas durante las modificaciones y las actualizaciones.
- **Facilidad de instalación (Installability):** La facilidad con la que el software puede instalarse y configurarse en un nuevo entorno.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 33 de 61

8. Portabilidad (Portability)

La portabilidad mide la capacidad del software para ser transferido o adaptado a diferentes entornos o plataformas.

Sub-características:

- **Adaptabilidad (Adaptability):** La capacidad del software para adaptarse a diferentes entornos, configuraciones o plataformas con mínima intervención.
- **Instalabilidad (Installability):** La facilidad con la que el software puede ser instalado en un nuevo entorno, lo que incluye la instalación, configuración y puesta en marcha sin problemas.
- **Compatibilidad con plataformas (Compatibility with Platforms):** La capacidad del software para funcionar en diferentes plataformas, sistemas operativos o arquitecturas de hardware sin requerir modificaciones sustanciales.



	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 34 de 61

ANEXOS

ANEXO 01: REQUERIMIENTOS FUNCIONALES



	ELICITACIÓN DE REQUERIMIENTOS UNIVERSIDAD NACIONAL DE JULIACA	OFICINA DE TECNOLOGÍAS DE LA INFORMACIÓN
Artefacto: [NOMBRE DE SISTEMA ABREVIADO]_REQF_ESP_V2.0 Fase de Inicio: Lista de requisitos especificados del sistema NOMBRESISTEMA	[NOMBRE DE SISTEMA ABREVIADO]	Cliente: [Nombre de la Oficina] Fecha: 02/01/2025

[NOMBRE DE SISTEMA ABREVIADO]
 [Nombre Completo del Sistema]

**LISTA DE REQUERIMIENTOS FUNCIONALES DEL
 SISTEMA [NOMBRE DE SISTEMA ABREVIADO]**
 [NOMBRE DE
 SISTEMA ABREVIADO]_REQF_ESP_V2.0

Historial de Revisiones

Fecha	Versión	Descripción	Autor
02/01/2025	2.0	Edición del Documento de Lista de requisitos funcionales del sistema.	Equipo [NOMBRE DE SISTEMA ABREVIADO]



Cliente: [Nombre de Cliente]
Fecha: 02/01/2025

[illegible]

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 37 de 61

ANEXO 02: ANALISIS DE REQUERIMIENTOS

	ELICITACIÓN DE REQUERIMIENTOS UNIVERSIDAD NACIONAL DE JULIACA	OFICINA DE TECNOLOGÍAS DE LA INFORMACIÓN
Artefacto: [NOMBRE DE SISTEMA ABREVIADO]_CU_V2.0 Fase de Inicio: Lista de requisitos especificados del sistema NOMBRESISTEMA	[NOMBRE DE SISTEMA ABREVIADO]	Cliente: [Nombre de la Oficina] Fecha: 02/01/2025

[NOMBRE DE SISTEMA ABREVIADO]

[Nombre Completo del Sistema]

ANÁLISIS DE REQUERIMIENTOS

[NOMBRE DE SISTEMA ABREVIADO]_REQF_ESP_V2.0

Historial de Revisiones

Fecha	Versión	Descripción	Autor
02/01/2025	2.0	Edición del Documento de Lista de requisitos funcionales del sistema.	Equipo [NOMBRE DE SISTEMA ABREVIADO]



Artefacto: [NOMBRE DE SISTEMA] [NOMBRE DE SISTEMA] Cliente: [Nombre de Oficina/otro]
ABREVIADO_REQ_ESP_V2.0 Fecha: 02/01/2025
Ciclo de Negocio: Elucidación de requerimientos ABREVIADO]

Tabla de Contenido

1. INTRODUCCIÓN	2
1.1. Propósito	2
1.2. Definiciones, acrónimos y abreviaturas	2
2. DIAGRAMA DE MODELO CONCEPTUAL	2
3. DIAGRAMA DE ACTIVIDADES	3
4. CASOS DE USO DE SISTEMA	3
4.1. Identificar de actores	3
4.2. Diagrama de Casos de Uso	4
4.3. Especificaciones de casos de uso	5



Artefacto: [NOMBRE DE SISTEMA]
ABREVIADO: REQ_ESP_V2.0
Ciclo de Negocio: Elitación de requerimientos

[NOMBRE DE SISTEMA
ABREVIADO]

Cliente: [Nombre de Oficina/Ente]
Fecha: 02/01/2025

1. INTRODUCCIÓN

1.1. Propósito

[El proceso de Trámite Documentario tiene como principal objetivo lograr atender los requerimientos de los usuarios, ya sean directores, docentes, padres de familia, personas naturales o jurídicas, superando las expectativas en tiempo, costo y calidad de la información en los trámites que se solicita, además de brindar orientación y atención de calidad al usuario. El proceso de Trámite Documentario compromete a todas las oficinas que conforman en la Universidad Nacional de Juliaca.]

1.2. Definiciones, acrónimos y abreviaturas

STD: Sistema de tramite documentario

CU: Casos de uso

DA-STD01: Sistema de trámite documentario diagrama de actividades 1

2. DIAGRAMA DE MODELO CONCEPTUAL

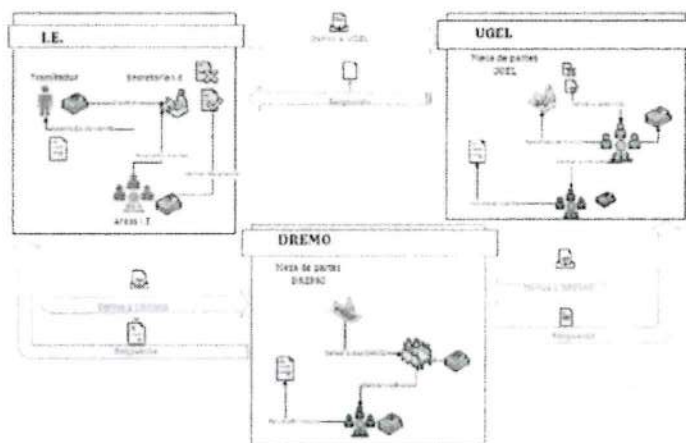


Figura N° 1: Modelo conceptual Sistema de tramite documentario



Artefacto: [NOMBRE DE SISTEMA] [NOMBRE DE SISTEMA] Cliente: [Nombre de Oficina / Divi]
ABREVIADO)_REQ_ESP_V2.0 ABREVIADO] Fecha: 02-01-2025
Ciclo de Negocio: Elitacion de requerimientos

3. DIAGRAMA DE ACTIVIDADES

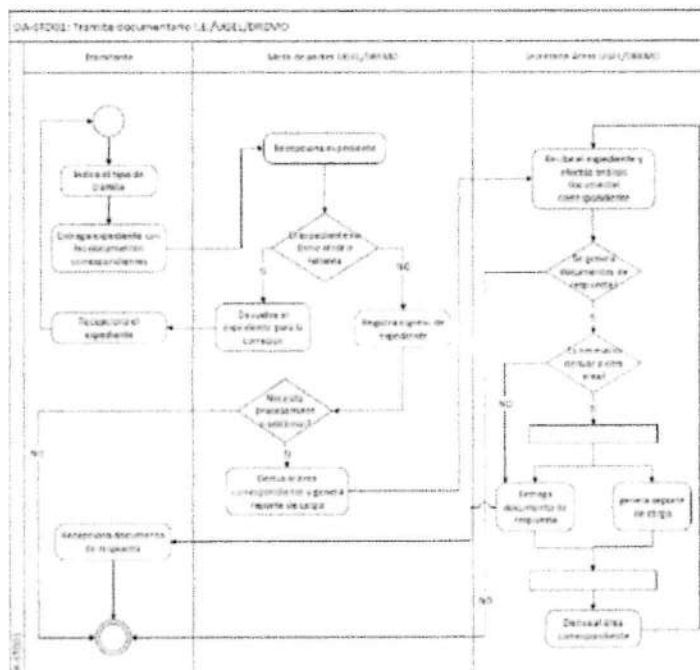


Figura N° 2: Diagrama de actividades tramite

4. CASOS DE USO DE SISTEMA

4.1. Identificar de actores

ACT-STD01	Administrador STD
Versión	1.0 (21-01-2025)
Autores	Equipo [NOMBRE DE SISTEMA ABREVIADO]
Fuentes	Stakeholder
Descripción	Es la persona que hará la configuración inicial del módulo, registrando tipos de trámites de la institución y sus requisitos, además podrá generar diversos tipos de documentos.
Comentarios	



Artefacto: [NOMBRE DE SISTEMA]
ABREVIADO]_REQ_ESP_V2.0
Ciclo de Negocio: Elitación de requerimientos

[NOMBRE DE SISTEMA
ABREVIADO]

Cliente: [Nombre de Cliente]
Fecha: 02/11/2025

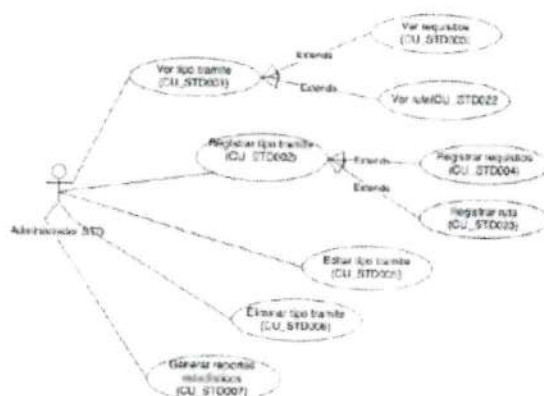
ACT - [NOMBRE ABREVIADO]02	[NOMBRE DE ACTOR]
Versión	
Autores	
Fuentes	
Descripción	
Comentarios	

ACT - [NOMBRE ABREVIADO]03	[NOMBRE DE ACTOR]
Versión	
Autores	
Fuentes	
Descripción	
Comentarios	

4.2. Diagrama de Casos de Uso

4.2.1. Administrador

DCU_STD01: Administrador



Artefacto: [NOMBRE DE SISTEMA] [NOMBRE DE SISTEMA] Cliente: [Nombre de Oficina: etc]
ABREVIADO_REQF_ESP_V2.0 ABREVIADO] Fecha: 02/01/2025
Ciclo de Negocio: Elitación de requerimientos

4.3. Especificaciones de casos de uso

4.3.1. CU-STD001 Ver tipo de trámite

CU-STD001	Ver tipo de trámite	
Versión	1.0 (21-01-2025)	
Autores	Equipo SIGESMED	
Actores	Administrador STD	
Fuentes	stakeholder	
Dependencias	Ninguno	
Requisitos funcionales	ERF1STD06	
Descripción	Este caso de uso describe los pasos para visualizar un tipo de trámite en el sistema.	
Precondición	El administrador está logueado en el sistema	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa al menú principal y selecciona la opción "Lista tramites"
	2	El sistema muestra la lista de trámites de la institución con los datos (Código de trámite, nombre, descripción, costo, duración, requisitos, ruta).
	3	El usuario filtra la búsqueda por tipo, código o nombre del trámite y presiona "buscar".
	4	El sistema actualiza la lista de trámites según el filtro realizado.
Secuencia Alternativa 1:	Paso	Acción
	1	
Postcondición	El usuario tiene la posibilidad de agregar nuevos trámites o editar los requisitos o ruta de los trámites registrados.	
Excepciones		
Importancia	Vital	
Urgencia	Inmediata	
Estado	Pendiente de verificación	
Estabilidad	Alta	
Comentarios	Ninguno	



Artefacto: [NOMBRE DE SISTEMA] [NOMBRE DE SISTEMA] Cliente: [Nombre de Cliente]
ABREVIADO_REQ_ESP_V2.0 ABREVIADO] Fecha: 02/01/2025
Ciclo de Negocio: Ejecución de Requerimientos

4.3.2. CU-[NOMBRE ABREVIADO]002 [Nombre de Caso de Uso]

CU-[NOMBRE ABREVIADO]002	[Nombre de Caso de Uso]	
Versión		
Autores		
Actores		
Fuentes		
Dependencias		
Requisitos funcionales		
Descripción		
Precondición		
Secuencia Normal	Paso	Acción
	1	
	2	
	3	
Secuencia Alternativa 1:	Paso	Acción
	1	
Postcondición		
Excepciones		
Importancia		
Urgencia		
Estado		
Estabilidad		
Comentarios		

	METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE NTP-ISO/IEC 12207	Código:MDMSW-01
		Versión: 1.0
		Página 44 de 61

ANEXO 03: DISEÑO DE MOCKUPS

	ELICITACIÓN DE REQUERIMIENTOS UNIVERSIDAD NACIONAL DE JULIACA	OFICINA DE TECNOLOGÍAS DE LA INFORMACIÓN
Artefacto: [NOMBRE DE SISTEMA ABREVIADO]_CU_V2.0 Fase de inicio: Lista de requisitos especificados del sistema NOMBRESISTEMA	[NOMBRE DE SISTEMA ABREVIADO]	Cliente: [Nombre de la Oficina] Fecha: 02/01/2025

[NOMBRE DE SISTEMA ABREVIADO]

[Nombre Completo del Sistema]

DISEÑO DE MOCKUPS

[NOMBRE DE SISTEMA ABREVIADO]_DISEÑO _V2.0

Historial de Revisiones

Fecha	Versión	Descripción	Autor
02/01/2025	2.0	Elaboración del documento de diseño de Mockups del Sistema Trámite Documentario	Equipo [NOMBRE DE SISTEMA ABREVIADO]

Artefacto: [NOMBRE DE SISTEMA] [NOMBRE DE SISTEMA] Cliente: [Nombre de Oficina/Pro]

ABREVIADO]_REQ_LSP_V2.0 ABREVIADO] Fecha: 02/01/2025

Ciclo de Negocio: Elaboración de requerimientos

Tabla de Contenido

1. INTRODUCCIÓN	2
1.1. Propósito	2
1.2. Definiciones, acrónimos y abreviaturas	2
2. STD: SISTEMA DE TRÁMITE DOCUMENTARIO	2
2.1. Lista de Trámites	2



Artefacto: [NOMBRE DE SISTEMA] [NOMBRE DE SISTEMA] Cliente: [Nombre de Cliente]
 ABREVIADO]_REQ_ESP_V2.0 Fecha: 02/11/2025
 Ciclo de Negocio: 1. Etapa de requerimientos

ABREVIADO]

1. INTRODUCCIÓN

1.1. Propósito

[El propósito de este documento es mostrar los mockups diseñados para el submódulo Sistema de trámite documentario del módulo administrativo de acuerdo a las especificaciones de casos de uso realizados. En cada especificación se da a conocer el requisito especificado, una descripción acerca del funcionamiento de los mockups y cómo se relacionan entre ellos, y las figuras correspondientes a los mockups realizados.]

1.2. Definiciones, acrónimos y abreviaturas

MOCKUP: Modelo a escala o tamaño real de un diseño de ventanas de un sistema.

STD: Sistema de trámite documentario.

DI: Diseño de Interfaz

CU: Caso de uso

2. STD: SISTEMA DE TRÁMITE DOCUMENTARIO

2.1. Lista de Trámites

CASO DE USO	CU-STD001 Ver tipo de trámite	REQUISITOS ESPECIFICADOS	ERF1-STD06
MOCKUP	STD_DI201. Ver Tipos de Trámite	USUARIOS	Responsable
Descripción	El usuario ingresa a la opción "Sistema de trámite documentario" del módulo administrativo, donde se muestra la lista de trámites registrados (código, nombre, descripción, costo, requisitos, rutas y la opción de editar) según la UNAJ, a la que corresponda.		

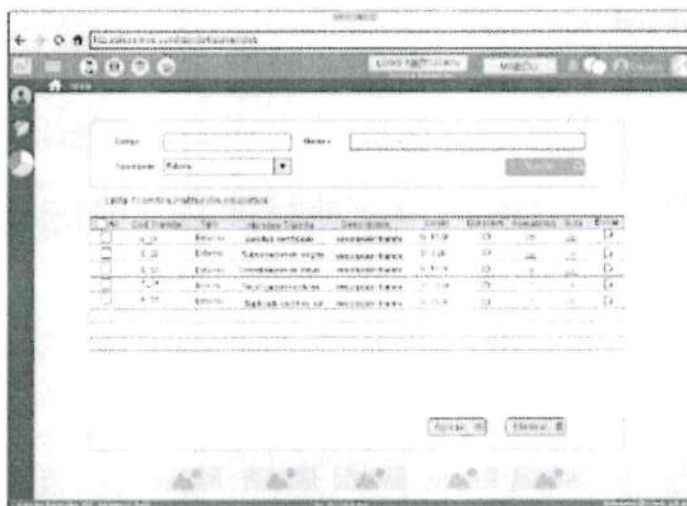


Figura N° 1: Ver tipo de trámite

ANEXO 04: FORMATO PILA PRODUCTO (PRODUCT BACKLOG)

PRODUCT BACKLOG				
ITEM	ESTIMACIÓN	DESCRIPCION	PRIORIDAD	REALIZADO POR
1	400	Logueo de Usuario	ALTA	
2	400	Registro de Usuario	ALTA	
3	400	Añadir/Modificar datos como Postulante	ALTA	
4	400	Añadir/Modificar/Eliminar estudios de grado del postulante	ALTA	
5	400	Añadir/Modificar/Eliminar estudios especializados	ALTA	
6	400	Añadir/Modificar/Eliminar EXPERIENCIA LABORAL	ALTA	
7	400	Añadir/Modificar/Eliminar REFERENCIAS PERSONALES	ALTA	
8	400	Visualizar Pantalla de bienvenido POSTULANTE	ALTA	
9	400	Añadir/Modificar datos como EMPRESA	ALTA	
10	400	Mantenimiento de OFERTAS LABORALES	ALTA	
11	400	Seguimiento de OFERTAS LABORALES POR PARTE DE LA EMPRESA	ALTA	
12	400	Seguimiento de OFERTAS LABORALES POR PARTE DE LOS POSTULANTES	ALTA	
13	200	Pantalla de Inicio	NORMAL	

ANEXO 05: FORMATO SPRINT BACKLOG (PILA SPRINT)

SPRINT BACKLOG					
ID	DURACION	SPRINT	HISTORIAL	REQUISITO	TAREA
ID1	7 Dias	1	H1	REQ002	Logueo de Usuario
			H2	REQ003	Registro de Usuario
			H3	REQ004	Añadir/Modificar datos como Postulante
			H4	REQ005	Añadir/Modificar/Eliminar estudios de grado del postulante
ID2	5 Dias	2	H5	REQ006	Añadir/Modificar/Eliminar estudios especializados
			H6	REQ007	Añadir/Modificar/Eliminar EXPERIENCIA LABORAL
			H7	REQ008	Añadir/Modificar/Eliminar REFERENCIAS PERSONALES
ID3	7 Dias	3	H8	REQ009	Visualizar Pantalla de bienvenido POSTULANTE
			H9	REQ010	Añadir/Modificar datos como EMPRESA
			H10	REQ011	Mantenimiento de OFERTAS LABORALES
ID4	6 Dias	4	H11	REQ012	Seguimiento de OFERTAS LABORALES POR PARTE DE LA EMPRESA
			H12	REQ013	Seguimiento de OFERTAS LABORALES POR PARTE DE LOS POSTULANTES

ANEXO 06: FORMATO DE CUADRO DE HISTORIAL SPRINT

Sprint No.						1	
Referencia Historial							
Fecha Inicio							
Fecha Final							
						Tareas Pendientes	
						Días Pendientes	
Historial	Orden de Priorizado	Descripción			Responsable	Duración Días	Estado
H1	400	Logueo de Usuario					1
H2	400	Registro de Usuario					2
H3	400	Añadir/Modificar datos como Postulante					2
H4	400	Añadir/Modificar/Eliminar estudios de grado del postulante					2



ANEXO 07: FORMATO PARA DOCUMENTAR USO DE SCRUM

Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

Proyecto

Descripción de la metodología de trabajo (scrum)

Historial de Revisiones

Fecha	Versión	Descripción	Autor
20/11/2011	1.0	Primera versión con los apartados y contenidos básicos	



Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

Tabla de contenido

1. Introducción.....	3
1.1. Propósito de este documento.....	3
1.2. Alcance.....	3
2. Descripción General de la Metodología.....	3
2.1. Fundamentación.....	3
2.2. Valores de trabajo.....	3
3. Personas y roles del proyecto.....	4
4. Artefactos.....	4
4.1. Pila de producto.....	4
4.2. Pila del sprint.....	5
4.3. Sprint.....	6
4.4. Incremento.....	6
4.5. Gráfica de producto (Burn Up).....	6
4.6. Gráfica de avance (Burn Down).....	8
4.7. Reunión de inicio de sprint.....	9
4.8. Reunión técnica diaria.....	9
4.9. Reunión de cierre de sprint y entrega del incremento.....	10



Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

Descripción de la metodología de trabajo

1. Introducción

Este documento describe la implementación de la metodología de trabajo scrum en la empresa para la gestión del desarrollo el proyecto.

Incluye junto con la descripción de este ciclo de vida iterativo e incremental para el proyecto, los artefactos o documentos con los que se gestionan las tareas de adquisición y suministro: requisitos, monitorización y seguimiento del avance, así como las responsabilidades y compromisos de los participantes en el proyecto.

1.1. Propósito de este documento

Facilitar la información de referencia necesaria a las personas implicadas en el desarrollo del sistema

1.2. Alcance

Personas y procedimientos implicados en el desarrollo del sistema

2. Descripción General de la Metodología

2.1. Fundamentación

Las principales razones del uso de un ciclo de desarrollo iterativo e incremental de tipo scrum para la ejecución de este proyecto son:

- Sistema modular. Las características del sistema permiten desarrollar una base funcional mínima y sobre ella ir incrementando las funcionalidades o modificando el comportamiento o apariencia de las ya implementadas.
- Entregas frecuentes y continuas al cliente de los módulos terminados, de forma que puede disponer de una funcionalidad básica en un tiempo mínimo y a partir de ahí un incremento y mejora continua del sistema.
- Previsible inestabilidad de requisitos.
 - o Es posible que el sistema incorpore más funcionalidades de las inicialmente identificadas.
 - o Es posible que durante la ejecución del proyecto se altere el orden en el que se desean recibir los módulos o historias de usuario terminadas.
 - o Para el cliente resulta difícil precisar cuál será la dimensión completa del sistema, y su crecimiento puede continuarse en el tiempo suspenderse o detenerse.

2.2. Valores de trabajo

Los valores que deben ser practicados por todos los miembros involucrados en el desarrollo y que hacen posible que la metodología Scrum tenga éxito son:

- Autonomía del equipo
- Respeto en el equipo
- Responsabilidad y auto-disciplina
- Foco en la tarea



Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

- Información transparencia y visibilidad.

3. Personas y roles del proyecto

Persona	Contacto	Rol

4. Artefactos

- **Documentos**
 - Pila de producto o Product Backlog
 - Pila de sprint o Sprint Backlog

- **Sprint**

- **Incremento**

Gráficas para registro y seguimiento del avance.

- Gráfica de producto o Burn Up
- Gráfica de avance o Burn Down.

Comunicación y reporting directo.

- Reunión de inicio de sprint
- Reunión técnica diaria
- Reunión de cierre de sprint y entrega del incremento

4.1. Pila de producto

Es el equivalente a los requisitos del sistema o del usuario (Con-Ops) en esta metodología.

El gestor de producto de su correcta gestión, durante todo el proyecto. El gestor de producto puede recabar las consultas y asesoramiento que pueda necesitar para su redacción y gestión durante el proyecto al Scrum Manager de este proyecto.

Responsabilidades del gestor de producto

- Registro en la lista de pila del producto de las historias de usuario que definen el sistema.
- Mantenimiento actualizado de la pila del producto en todo momento durante la ejecución del proyecto.
 - Orden en el que desea quiere recibir terminada cada historia de usuario.
 - Incorporación / eliminación /modificaciones de las historias o de su

Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

orden de prioridad.

- Disponibilidad: [Indicar modo: si mantiene directamente la pizarra o intranet o medios de comunicación, o envía las modificaciones al Scrum Manager para su actualización, u otro modo empleado en la empresa...]

Responsabilidades del Scrum Manager

- Supervisión de la pila de producto, y comunicación con el gestor del producto para pedirle aclaración de las dudas que pueda tener, o asesorarle para la subsanación de las deficiencias que observe.

Responsabilidades del equipo técnico

- Conocimiento y comprensión actualizado de la pila del producto.
- Resolución de dudas o comunicación de sugerencias con [indicar según se haya implementado en la empresa (gestor de producto / scrum manager / ...)]

Responsabilidades del resto de implicados

- Conocimiento y comprensión actualizado de la pila del producto.
- Resolución de dudas o comunicación de sugerencias con [indicar según se haya implementado en la empresa (gestor de producto / scrum manager / ...)]

Notas: Si lo necesita, el gestor de producto puede solicitar asesoría al Scrum Manager del proyecto o personal técnico del equipo para conocer la estimación temprana de las historias de usuario cuyo tamaño aproximado le presenten dudas.

1	Muy alta	Plataforma tecnológica	30	AR
2	Muy alta	Interfaz usuario	40	LR
3	Muy alta	Entorno de desarrollo del sistema	40	LR
4	Alta	Elaboración de la pila de producto	60	AR
5	Alta	Test	900	XX

4.2. Pila del sprint

Es el documento de registro de los requisitos detallados o tareas que va a desarrollar el equipo técnico en la iteración (actual o que está preparándose para comenzar)

Responsabilidades del gestor de producto

- Presencia en las reuniones en las que el equipo elabora la pila del sprint.
- Resolución de dudas sobre las historias de usuario que se descomponen en la pila del sprint.

Responsabilidades del Scrum Manager

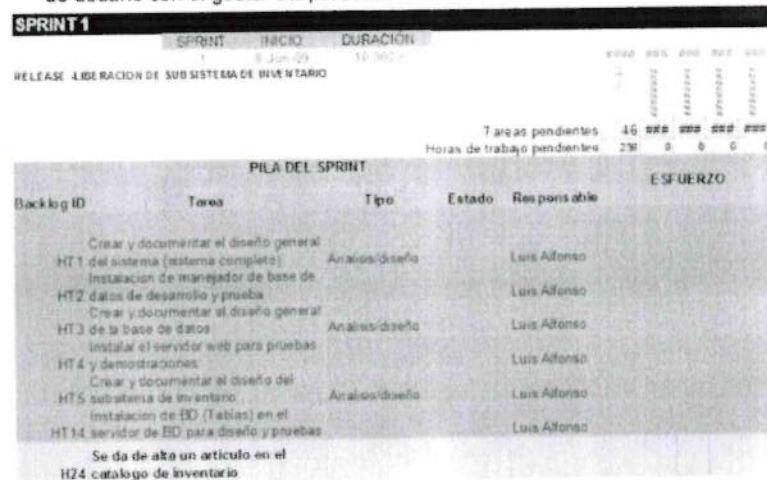
- Supervisión y asesoría en la elaboración de la pila de la pila del sprint.

Responsabilidades del equipo técnico

- Elaboración de la pila del sprint.
- Resolución de dudas o comunicación de sugerencias sobre las historias

Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

de usuario con el gestor del producto.



4.3. Sprint

Cada una de las iteraciones del ciclo de vida iterativo Scrum. La duración de cada sprint [indicar según las características de la implementación: es de xx días laborables / inferior a xx días y se determina al inicio del mismo / otras posibilidades]

4.4. Incremento

Parte o subsistema que se produce en un sprint y se entrega al gestor del producto completamente terminada y operativa.

4.5. Gráfica de producto (Burn Up)

Representación gráfica del plan de producto previsto por el gestor de producto. Es una gráfica que representa los temas o epics del sistema en el orden que se desean, y el tiempo en el que se prevé su ejecución.

Responsabilidades del gestor de producto

- Confección.
- Mantenimiento actualizado en todo momento durante la ejecución del proyecto.
 - o Orden en el que desea disponer de los temas o "epics" del sistema, e hitos del producto (versiones).
 - o Incorporación / eliminación / modificaciones de los temas, de su orden de prioridad, estimaciones o hitos.
 - o Disponibilidad: [Indicar modo: si mantiene directamente la pizarra o intranet o medios de comunicación, o envía las modificaciones al Scrum Manager para su actualización, u otro modo empleado]

Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

en la empresa...]

Responsabilidades del Scrum Manager

- Supervisión del gráfico de producto, y comunicación con el gestor del producto para pedirle aclaración de las dudas que pueda tener, o asesorarle para la subsanación de las deficiencias que observe.

Responsabilidades del equipo técnico

- Conocimiento y comprensión actualizado del plan del producto.
- Resolución de dudas o comunicación de sugerencias con [indicar según se haya implementado en la empresa (gestor de producto / scrum manager / ...)]

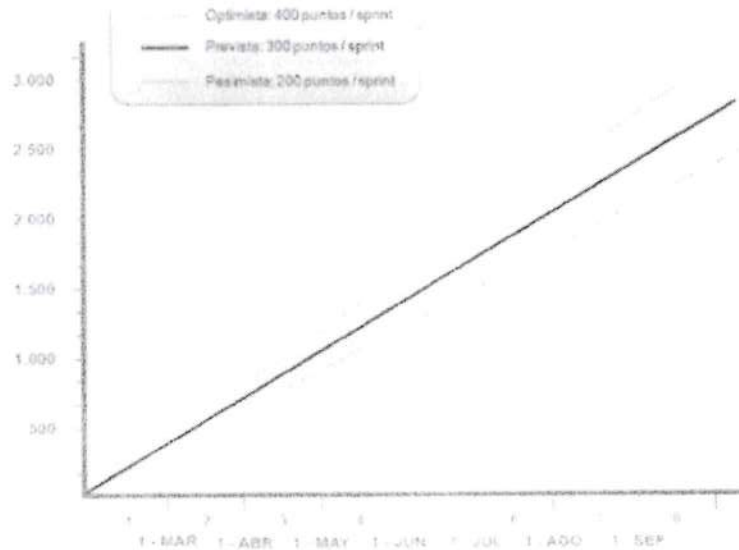
Responsabilidades del resto de implicados

- Conocimiento y comprensión actualizado del plan de producto.
- Resolución de dudas o comunicación de sugerencias con [indicar según se haya implementado en la empresa (gestor de producto / scrum manager / ...)]

Id	Temas	Trabajo	Criterio de validación	Estimación
1	Tema A 1.0	150	Ut enim ipsam dolor sit amet	650 PUNTOS
2	Tema B 1.0	250	consectetur adipiscing elit	
3	Tema C 1.0	250	Aliquam vehicula dui quisque titor	
4	Tema D 1.0	350	Paraphrased Lorem	
5	Tema A 1.1	250	Propositoribus	1.700 PUNTOS
6	Tema D 1.1	950	peratibus et magnis dis partur	
7	Tema E 1.0	150	Quisque volutpat ante sit amet velit	2.550 PUNTOS
8	Tema B 1.1	900	Cras sed ipsum pellentesque	
9	Tema C 1.1	150	vestibulum vari diam sed pede	Versión 1.2
10	Tema E 1.1	200	Suspendisse ut aliquam felis et turpis	
11	Tema F 1.0	TBD	Nam enim eros et lorem vitae pulvinar	
12	Tema A 1.2	TBD	Suspendisse ut pulvinar in nec magna	
13	Tema C 1.2	TBD	Nam enim eros et lorem vitae pulvinar	
14	Tema F 1.1	TBD	Morbi ante felis congue	



Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	



La línea de velocidad proyecta sobre el eje X la fecha o sprint en el que previsiblemente se completarán las versiones representadas en el eje Y.

4.6. Gráfica de avance (Burn Down)

Gráfico que muestra el estado de avance del trabajo del sprint en curso.

Responsabilidades del gestor de producto

- Sin responsabilidades específicas, más allá de mantenerse regularmente informado del avance del sprint y disponible para atender decisiones para la resolución de opciones en sprints sobrevalorados o infravalorados (la gráfica de avance predice una entrega anterior o posterior a la fecha prevista)

Responsabilidades del Scrum Manager

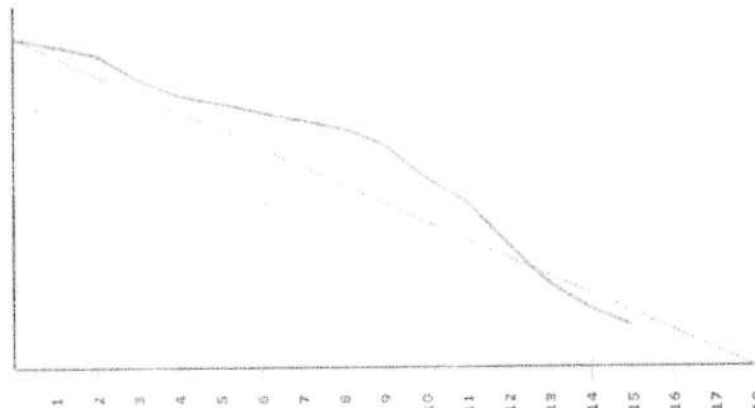
- Supervisión de la actualización diaria por parte del equipo.

Responsabilidades del equipo técnico

- Actualización diaria del gráfico de avance.



Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	



4.7. Reunión de inicio de sprint

Reunión para determinar las funcionalidades o historias de usuario que se van a incluir en el próximo incremento.

Responsabilidades del gestor de producto

- Asistencia a la reunión.
- Exposición y explicación de las historias que necesita para la próxima iteración y posibles restricciones de fechas que pudiera tener.

Responsabilidades del Scrum Manager

- Moderación de la reunión

Responsabilidades del equipo técnico

- Confección de la pila del sprint.
- Auto-asignación del trabajo.

4.8. Reunión técnica diaria

Puesta en común diaria del equipo con presencia del Coordinador del proyecto o Scrum Manager de duración máxima de 10 minutos.

Responsabilidades del Scrum Manager

- Supervisión de la reunión y anotación de las necesidades o impedimentos que pueda detectar el equipo.
- Gestión para la solución de las necesidades o impedimentos detectados por el equipo.

Responsabilidades del equipo técnico

- Comunicación individual del trabajo realizado el día anterior y el previsto para día actual.
- Actualización individual del trabajo pendiente.
- Actualización del gráfico de avance [o artefacto de seguimiento de sprint empleado] para reflejar el estado de avance.
- Notificación de necesidades o impedimentos previstos u ocurridos para realizar las tareas asignadas.



Desarrollo del Sistema	Versión:
Planificación	Fecha:
Descripción de la metodología de trabajo	

4.9. Reunión de cierre de sprint y entrega del incremento

Reunión para probar y entregar el incremento al gestor del producto.

Características.

- Prácticas: sobre el producto terminado, no sobre simulaciones o imágenes).
- De tiempo acotado máximo de 2 horas.

Responsabilidades del gestor de producto

- Asistencia a la reunión.
- Recepción del producto o presentación de reparos.

Responsabilidades del Scrum Manager

- Moderación de la reunión

Responsabilidades del equipo técnico

- Presentación del incremento.



FLUJOGRAMA

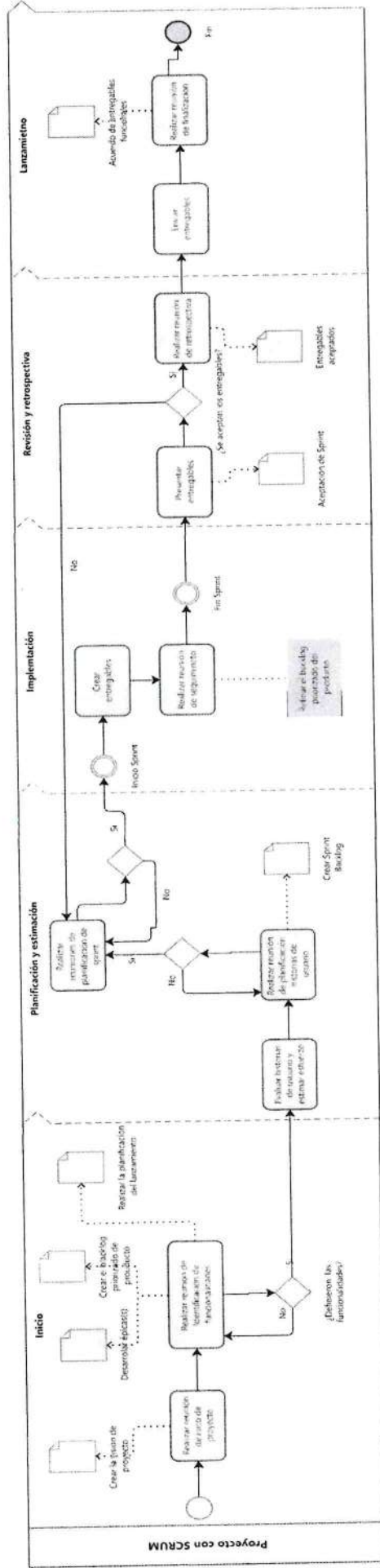


Figura 15: Proyecto con Scrum



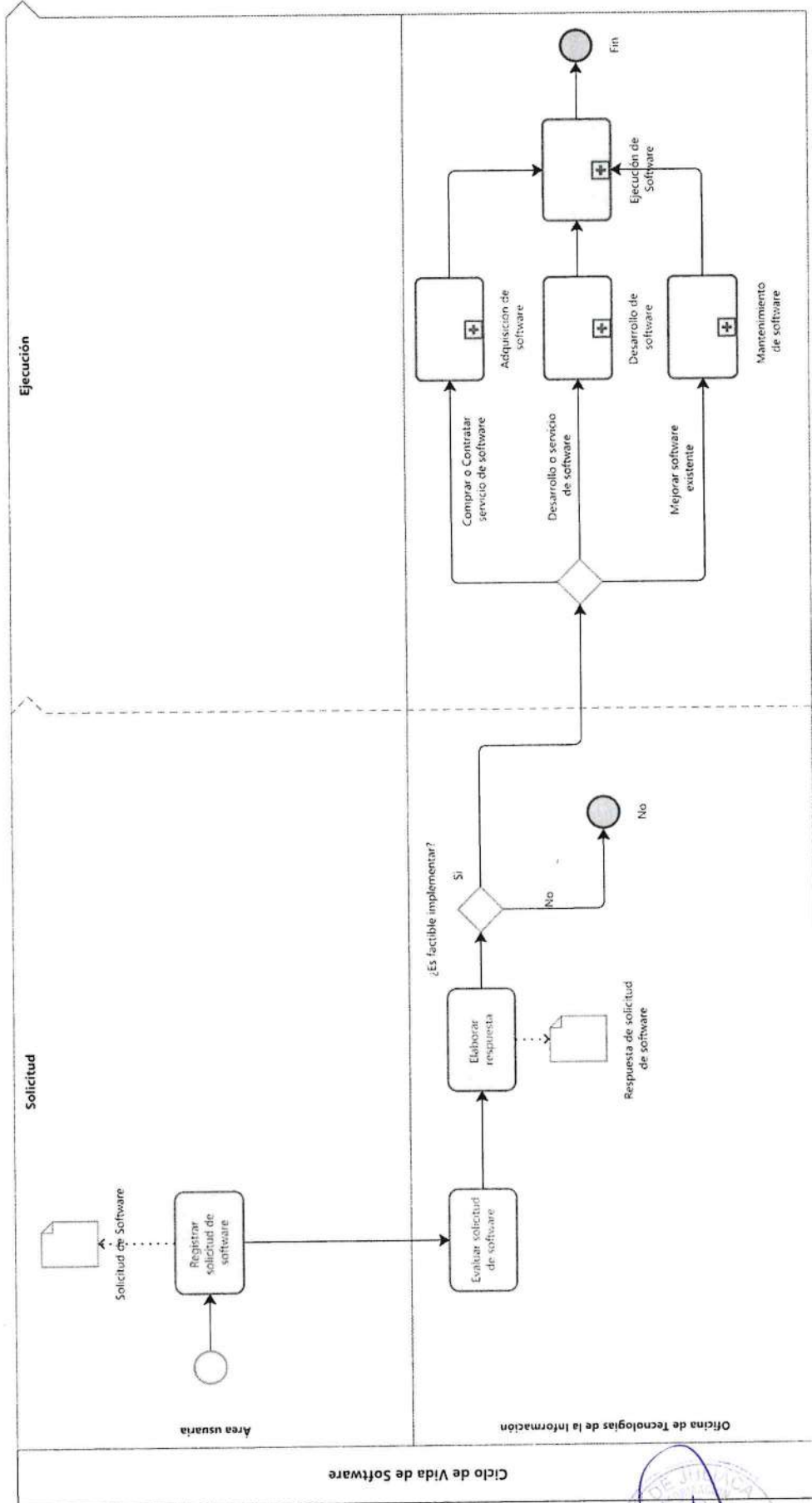


Figura 16: Ciclo de vida de software

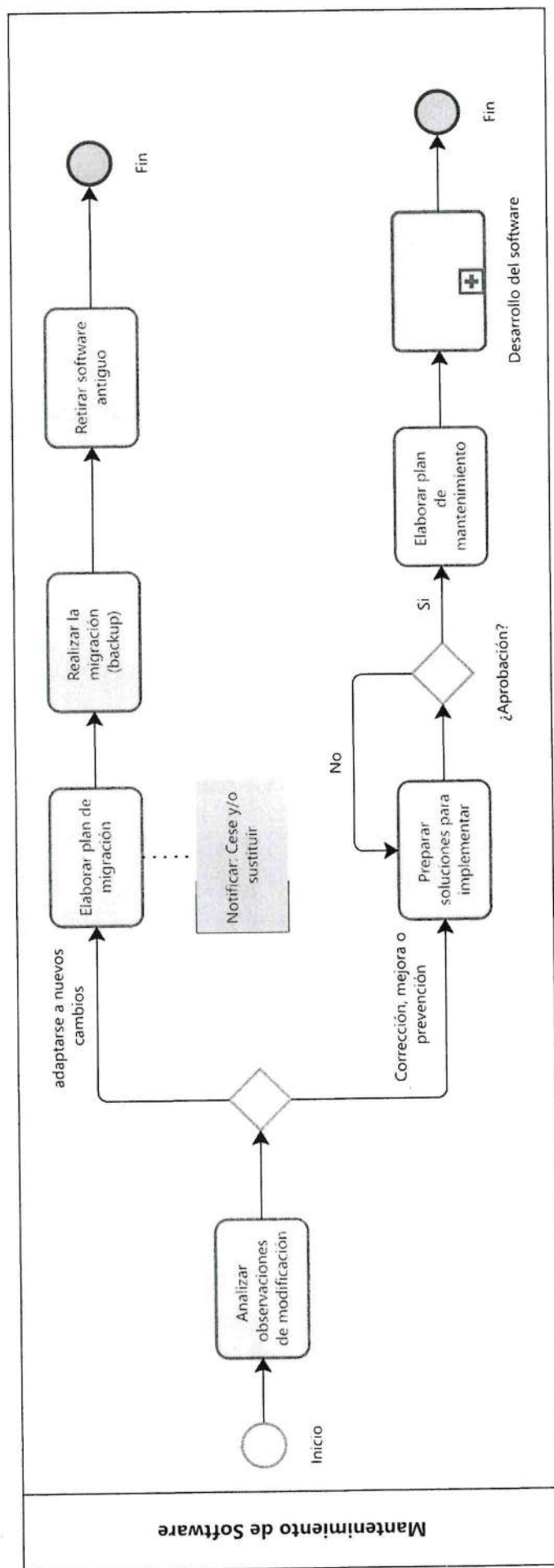


Figura 17: Mantenimiento de software



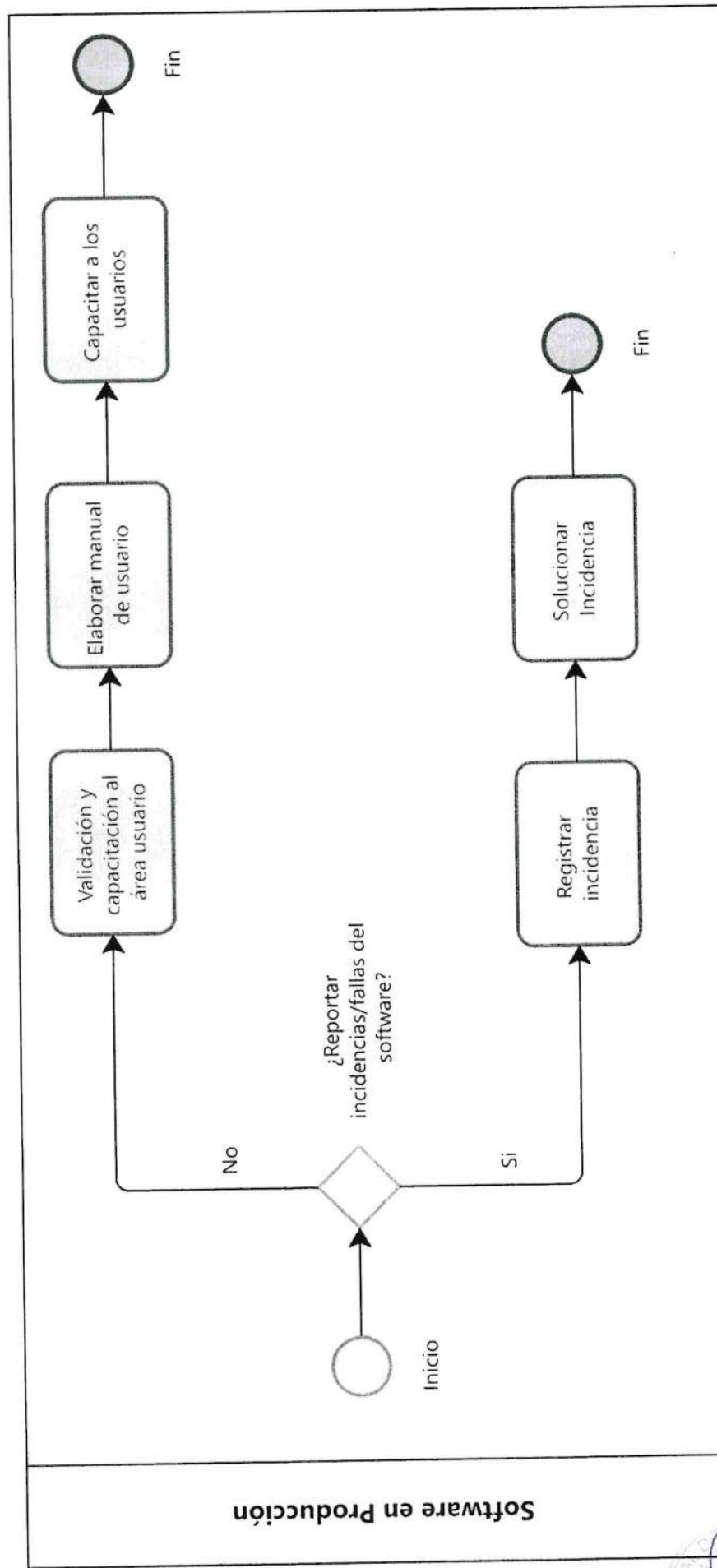


Figura 18: Software en producción