



PERÚ

Ministerio de Trabajo
y Promoción del Empleo

"Decenio de la Igualdad de oportunidades para mujeres y hombres"
"Año de la lucha contra la Corrupción y la Impunidad"

RESOLUCIÓN DE DIRECCIÓN EJECUTIVA
N° 098 – 2019-MTPE/3/24.2/JOVENES PRODUCTIVOS/DE

Lima, 05 DIC. 2019

VISTOS:

El Informe N° 339-2019/JOVENES PRODUCTIVOS/DE/UGP/AEI, emitido por el Jefe del Área de Estadística e Informática, Informe N° 307-2019/JOVENES PRODUCTIVOS/DE/UGP, emitido por el Área de Planificación, Presupuesto, Monitoreo y Evaluación, Memorando N° 422-2019-JÓVENES PRODUCTIVOS/DE/UGAL, emitido por la Unidad Gerencial de Asesoría Legal, el Informe N° 415-2019/JOVENES PRODUCTIVOS/DE/UGP/AEI, emitido por el Jefe del Área de Estadística e Informática y el Informe N° 195-2019-JÓVENES PRODUCTIVOS/DE/UGAL, emitido por la Unidad Gerencial de Asesoría Legal, y;

CONSIDERANDO:

Que, mediante Decreto Supremo N° 013-2011-TR, modificado por Decreto Supremo N° 004-2012-TR y N° 004-2015-TR, se creó el Programa Nacional de Empleo Juvenil "Jóvenes Productivos", en el ámbito del Ministerio de Trabajo y Promoción del Empleo, con la finalidad de insertar a los jóvenes de 15 a 29 años de edad, de escasos recursos económicos y/o en situación de vulnerabilidad, de las familias más pobres de la población, rural y urbana, a través de su capacitación y fomento en el acceso al mercado de trabajo formal. Está a cargo de un Director Ejecutivo, quien es designado mediante Resolución Ministerial, quien es designado mediante Resolución Ministerial del Ministerio de Trabajo y Promoción del Empleo, a propuesta del Viceministro de Promoción del Empleo y Capacitación Laboral;

Que, la Ley N° 27658, Ley Marco de Modernización de la Gestión del Estado, modificado por Decreto Legislativo N° 1446, tiene como objeto establecer los principios y la base legal para iniciar el proceso de modernización de la gestión del Estado, en todas sus instituciones e instancias.

Que, mediante Decreto Legislativo N° 1412, se aprobó la Ley de Gobierno Digital, con el objeto de establecer el marco de gobernanza del gobierno digital para la adecuada gestión de la identidad digital, servicios digitales, arquitectura digital, interoperabilidad, seguridad digital y datos, así como el régimen jurídico aplicable al uso transversal de tecnologías digitales en la digitalización de procesos y prestación de servicios digitales por parte de las entidades de la Administración Pública en los tres niveles del gobierno.

Que, mediante la Resolución Ministerial N° 041-2017-PCM, se 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, 3ª Edición" en las entidades integrantes del Sistema Nacional de Informática.

Que, el Manual de Operaciones del Programa Nacional de Empleo Juvenil "Jóvenes Productivos" aprobada mediante la Resolución Ministerial N° 179-2012-TR y modificada mediante Resolución Ministerial N° 215-2014-TR, señala en los incisos i) del artículo 11, que la Dirección Ejecutiva tiene entre otras funciones, el aprobar directivas, reglamentos, instrumentos y



PERÚ

Ministerio de Trabajo
y Promoción del Empleo

"Decenio de la Igualdad de oportunidades para mujeres y hombres"
"Año de la lucha contra la Corrupción y la Impunidad"

procedimientos de carácter técnico operativo del Programa y además, de acuerdo al inciso h) que tiene la facultad de expedir resoluciones en asuntos de su competencia;

Que, el proyecto de Directiva sobre la "Metodología para el Desarrollo y Mantenimiento de Software del Programa Nacional de Empleo Juvenil Jóvenes Productivos", tiene como objetivo establecer los procedimientos, actividades y tareas, a nivel de gestión que garanticen obtener el producto software en un marco de tiempo, alcance y en un nivel adecuado de calidad;

Que, mediante el Informe N° 307-2019/JOVENES PRODUCTIVOS/DE/UGP, la Unidad Gerencial de Planificación, Presupuesto, Monitoreo y Evaluación, señaló que el proyecto de la Directiva se encuentra conforme a las normativas vigentes;

Que, asimismo, la Unidad Gerencial de Asesoría Legal emite opinión legal sobre la viabilidad de la Directiva, indicando que se encuentra conforme a la normativa vigente;

Que, por las consideraciones expuestas, resulta pertinente aprobar la directiva denominada "Metodología para el Desarrollo y Mantenimiento de Software del Programa Nacional de Empleo Juvenil Jóvenes Productivos";

Con los visaciones de las Unidades Gerenciales de Planificación, Presupuesto, Monitoreo y Evaluación y la Unidad Gerencial de Asesoría Legal, y;

De conformidad con la Ley N° 27658, Ley Marco de Modernización de la Gestión del Estado, modificado por el Decreto Legislativo N° 1446, el Decreto Legislativo N° 1412, que aprueba la Ley de Gobierno Digital, la Resolución Ministerial N° 041-2017-PCM y la Norma Técnica Peruana NTP-ISO/IEC 12207-2016-Ingeniería de Software y Sistemas. Procesos del Ciclo de vida del Software 3ª Edición y la Resolución Directoral N° 013-2016-INACAL/DN, que aprueba, entre otras, la Norma Técnica Peruana NTP-ISO/IEC 12207:2016 – Ingeniería de Software y Sistemas. Procesos del ciclo de vida del Software 3ª Edición.

SE RESUELVE:

Artículo 1°.- Aprobar la Directiva N° 006-2019/MTPE/3/24.2/JOVENES PRODUCTIVOS/DE, denominada "Metodología para el Desarrollo y Mantenimiento de Software del Programa Nacional de Empleo Juvenil Jóvenes Productivos, la misma que como anexo forma parte integrante de la presente resolución.

Artículo 2°.- Notificar el contenido de la presente resolución al Jefe del Área de Estadística e Informática.

Artículo 3°.- ENCARGAR al Área de Estadística e Informática de la Unidad Gerencial de Planificación, Presupuesto, Monitoreo y Evaluación, la publicación de la Resolución en el portal institucional (www.jovenesproductivos.gob.pe) del Programa Nacional de Empleo Juvenil "Jóvenes Productivos".

REGÍSTRESE Y COMUNÍQUESE.


ROXANÁ NOLI CHAVEZ
Directora Ejecutiva
Programa Nacional de Empleo Juvenil
"Jóvenes Productivos"

www.trabajo.gob.pe

Av. Salaverry N° 655
Jesús María

DIRECTIVA N° 006 - 2019-MTPE-JOVENES PRODUCTIVOS/DE
"METODOLOGIA PARA EL DESARROLLO Y MANTENIMIENTO DE SOFTWARE DEL
PROGRAMA NACIONAL DE EMPLEO JUVENIL JOVENES PRODUCTIVOS"

I. FINALIDAD

Lograr y asegurar un proceso uniforme para optimizar la calidad de los sistemas de información del Programa Nacional de Empleo Juvenil "Jóvenes Productivos".

II. OBJETIVO

Establecer los procedimientos, actividades y tareas, a nivel de gestión, que garanticen obtener el producto software en un marco de tiempo, alcance y en un nivel adecuado de calidad.

III. BASE LEGAL

- 3.1** Ley N° 27658, Ley Marco de Modernización de la Gestión del Estado, modificado por Decreto Legislativo N° 1446.
- 3.2** Decreto Legislativo N° 1412, que aprueba la Ley de Gobierno Digital.
- 3.3** Decreto Supremo N° 013-2011-TR, modificado por los Decretos Supremos N° 004-2012-TR y 004-2015-TR, que crea el Programa Nacional de Empleo Juvenil "Jóvenes Productivos".
- 3.4** Resolución Ministerial N° 179-2012-TR, que aprueba el Manual de Operaciones del Programa Nacional de Empleo Juvenil "Jóvenes a la obra", modificado con Resolución Ministerial N° 234-2014-TR.
- 3.5** Resolución Ministerial N° 041-2017-PCM que 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. 3a Edición.
- 3.6** Resolución Directoral N° 013-2016-INACAL/DN, que aprueba, entre otras, la Norma Técnica Peruana NTP-ISO/IEC 12207:2016 – Ingeniería de Software y Sistemas. Procesos del ciclo de vida del software. 3a Edición.

IV. ALCANCE

Las disposiciones descritas en la presente Directiva son de cumplimiento obligatorio para todos los que participen o intervengan en los proyectos de software internos o generados por terceros.

V. ABREVIATURAS

- **AEI:** Área de Estadística e Informática.
- **PNEJP:** Programa Nacional de Empleo Juvenil "Jóvenes Productivos".
- **UGP:** Unidad Gerencial de Planificación, Presupuesto, Monitoreo y Evaluación.
- **UML:** Lenguaje Unificado de Modelado.
- **UO:** Unidad Orgánica.

DIRECTIVA N° 006 - 2019-MTPE-JOVENES PRODUCTIVOS/DE

**“METODOLOGIA PARA EL DESARROLLO Y MANTENIMIENTO
DE SOFTWARE DEL PROGRAMA NACIONAL DE EMPLEO
JUVENIL JOVENES PRODUCTIVOS”**



VI. DEFINICIONES DE TERMINOS

Para los efectos de la presente directiva se entenderá por:

- 6.1 Artefacto.** - Documentos y/o diagramas que se generan durante el proceso de gestión del ciclo de vida del software.
- 6.2 Ciclo de desarrollo de software.** - Tiempo entre el inicio del desarrollo del software y la entrega del mismo.
- 6.3 Entregable.** - Producto de software, informe y/o servicio
- 6.4 Equipo.** - Se refiere al equipo de colaboradores designados en la elaboración del Software, también llamados *Team* o Equipo Área de Estadística e Informática.
- 6.5 Gestión de Proyectos.** - Es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para cumplir con los requisitos del mismo.
- 6.6 Historias de usuario.** - Representación de un requisito de software escrito en una o dos frases utilizando un lenguaje sencillo que describan dicho requisito.
- 6.7 Lenguaje Unificado de Modelado.** - Lenguaje de propósito general que combina notaciones provenientes desde: Modelado orientado a objetos, Modelado de Datos, Modelado de Componentes y Modelado de flujos de trabajo.
- 6.8 Línea base.** - Denominación que se da al conjunto de documentos de planificación del proyecto sea de, alcance, tiempo o costo.
- 6.9 Marco de proceso común.** - Es un proceso genérico que define las actividades requeridas para ejecutar un proyecto software. Puede ser adaptado para un proyecto específico.
- 6.10 Metodología de ingeniería del software.** - Los métodos de ingeniería del software proporcionan el "cómo" para la construcción del software. Abarcan las tareas técnicas requeridas para realizar y documentar el análisis de requisitos, el diseño, la construcción de programas, las pruebas y el mantenimiento.
- 6.11 Modelos de proceso ágiles.** - Los modelos de proceso ágiles son enfoques de desarrollo donde los requisitos del cliente se cumplen temprano en el ciclo de vida de desarrollo del software a través de continuas entregas de software. En estos modelos, los cambios en los requisitos son bienvenidos.
- 6.12 Modelos de proceso de software.** - Un modelo de proceso de software es una estrategia abstracta para la construcción y mantenimiento de un producto software. Ayuda a los jefes de proyecto en la planificación y ejecución de un proyecto. También se le llama modelo de ciclo de vida o paradigma de ingeniería del software.
- 6.13 Modelo de proceso incremental.** - El modelo de proceso es un enfoque en que el producto se entrega al cliente de manera incremental sobre un periodo de tiempo planificado.
- 6.14 Proceso.** - Es un conjunto de actividades o eventos (coordinados u organizados) que se realizan o suceden (alternativamente o simultáneamente) bajo ciertas circunstancias en un determinado lapso de tiempo.
- 6.15 Productos de trabajo.** - Elementos que se crean durante el curso del proceso software, tales como documentos, software y manuales.
- 6.16 Product Backlog.** - Listado de requerimientos que debe cubrir el producto software, que van de la mano con sus historias de usuarios respectiva.
- 6.17 PMBOK.** - *Project Management Body of Knowledge* o Guía de los Fundamentos para la dirección de Proyectos.

- 6.18 Proyecto.** - Es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.
- 6.19 Requerimiento.** - Un requerimiento del software constituye una "capacidad o condición que deberá ser alcanzada por el producto software".
- 6.20 Rol.** - Rol estándar propuesto para ejecutar según corresponda diversos procesos durante un Ciclo de Vida de Software.
- 6.21 RUP.** - *Rational Unified Process* o en español Proceso Unificado de Rational es un proceso de desarrollo de software, el cual constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas.
- 6.22 Scrum.** - Es un conjunto de buenas prácticas con el objeto de trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto.
- 6.23 Scrum Master.** - Su trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *Scrum Master* no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El *Scrum Master* se asegura de que el proceso Scrum se utiliza como es debido haciendo que las reglas se cumplan.
- 6.24 Sistemas de información.** - Conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo. Dichos elementos formarán parte de alguna de las siguientes categorías:

- Personas
- Datos
- Actividades o técnicas de trabajo
- Recursos materiales en general (generalmente recursos informáticos y de comunicación, aunque no necesariamente)

- 6.25 Sprint.** - Al usar la metodología SCRUM, se planifican entregables en un determinado número de semanas (en promedio 3 semanas), los cuales se le denominan *Sprint* o iteración.

- 6.26 Sprint Backlog.** - Es el listado de tareas a realizar por iteración o Sprint.

- 6.27 Tarea.** - Ítem de un Sprint Backlog, a ser desarrollado por un miembro del equipo del Área de Estadística e Informática.

VII. RESPONSABILIDAD

- 7.1** El área de Estadística e Informática y los que se encuentren involucrados en los productos de software internos o generados por terceros son responsables de velar por el estricto cumplimiento de lo dispuesto en la presente Directiva.
- 7.2** El área de Estadística e Informática brindará opinión técnica para aquellos procesos realizados por terceros.
- 7.3** El área de Estadística e Informática, se encargará del desarrollo y mantenimiento de los sistemas de información elaborados en el PNEJJP.
- 7.4** El área de Estadística e Informática, se encargará de mantener actualizada la presente Directiva de acuerdo a los cambios que se produzcan en el sector, regulaciones externas, por mejora continua y cuando las nuevas tendencias tecnológicas lo exijan.

VIII. MECANICA OPERATIVA

8.1 DE LAS FASES DEL PROCESO

8.1.1 Fase de Requerimientos. - Se inicia con la recepción del Memorando, el cual deberá especificar las necesidades de cada Unidad Orgánica del Programa.

Posteriormente, se coordinará las reuniones con los usuarios y el equipo de desarrollo, a fin de identificar y priorizar el *Product Backlog* que aportará con la planificación del proyecto.

N°	Proceso NTP	Artefactos RUP	Actividades SCRUM
1.	Proceso de Definición de Requisitos de las Partes Interesadas.	Especificación de Requerimientos (Alcance)	- Identificación de involucrados - Formación del equipo - Creación de la lista priorizada del producto (Product Backlog)

8.1.2 Fase de Análisis y Diseño. - En esta fase se ejecuta el trabajo planificado de los entregables de ingeniería de software que figuran en el Cronograma del Proyecto. Se realiza la especificación de los requerimientos que facilitará el mecanismo apropiado para comprender lo que el usuario necesita.

N°	Proceso	Artefactos RUP	Actividades SCRUM
1.	Proceso de Análisis de Requisitos del Software.	Diseño y desarrollo de casos de uso, o flujos de casos de uso	Elaborar, aprobar, estimar y asignar las historias de usuarios.
2.	Proceso de Diseño detallado del Software.	Documento de Diseño conteniendo los diagramas UML: Clases, Entidad-Relación (o modelo de base de datos) e interfaz de usuario, como mínimo.	

8.1.3 Fase de Implementación (Desarrollo). - Se tiene como objetivo la construcción del código fuente y los componentes del software, así también como los objetos de base de datos según el análisis y diseño realizado en la fase anterior.

N°	Proceso	Artefactos RUP	Actividades SCRUM
1.	Proceso de Construcción del Software.	Código fuente	- Crear entregables y realizar el seguimiento del trabajo (<i>Scrumboard</i>) - Llevar a cabo la reunión diaria de estado (<i>Conduct daily standup</i>) - Actualizar lista de requerimientos (Seguimiento al Product Backlog)

8.2 DESCRIPCIÓN Y DESARROLLO DE LOS PROCESOS

8.2.1 FASE DE REQUIRIMENTOS

- a) **Definición de Requisitos de las partes interesadas.** - Su propósito es definir los requisitos para un sistema o software que pueda proveer los servicios que los usuarios y otras partes interesadas necesiten en un ambiente definido.

En el decurso del proceso, se identificará a las partes interesadas, sus necesidades y expectativas, para posteriormente realizar el análisis y transformación en un conjunto común priorizado de requisitos (*Product Backlog*) que expresará la interacción prevista y el cual será documentado a través de una lista de Especificaciones de Requerimientos.

Las acciones realizadas por el personal que participe y/o intervenga en los proyectos de software, deben considerar lo siguiente:

- o **Actividades del proceso**

Código y nombre	1. Recepcionar Memorando de solicitud a UGP.	
Descripción	Mediante un documento de tipo "Memorando", toda Unidad Orgánica del PNEJP, solicitará el desarrollo o implementación de mejoras de software, indicando los detalles de la necesidad, objetivos y requisitos de alto nivel.	
Rol / Responsable	- o Gerente de la Unidad Orgánica - Emisor - o Gerente de la Unidad Gerencial de Planificación, Presupuesto y Monitoreo - Receptor	
Tareas	Paso	Descripción
	1	El gerente de la UO o jefe del área usuaria, elaborará el memorando y derivará mediante el sistema de trámite documentario a la UGP, quien a su vez aprobará y derivará el documento a AEI.
	2	El jefe del AEI se comunicará con el gerente de la UO para agendar una reunión.
Comentarios	Es importante tener en cuenta el valor del proyecto al PNEJP y consecuentemente al logro de sus objetivos y determinar si el proyecto está alineado con las Directivas y normativas vigentes. Decidir si se aprueba o desestima el inicio formal del proyecto.	
Entradas	Memorando	
Salidas		

Código y nombre	2. Identificar involucrados.	
Descripción	El jefe de la AEI identificará durante la reunión, las personas involucradas de la UO solicitante y AEI asignando los siguientes roles: - <u>Scrum master</u>: Persona que lidera al equipo guiándolo para que cumpla los procesos de la metodología. - <u>Product Owner</u>: Representante del área usuaria y es responsable de trasladar la visión del proyecto al equipo y priorizar el <i>Product Backlog</i>.	
Rol / Responsable	☐ Jefe del Área de estadística e Informática – Receptor.	
Tareas	Paso	Descripción
	1	Entre los jefes de las áreas usuarias y AEI se definirá la lista de involucrados con sus roles dentro del proyecto.
	2	Elaborar un acta de reunión, el cual deberá especificarse la información de los participantes y sus roles.
Comentarios	Se recomienda explicar el presente marco metodológico durante la reunión con el objetivo de alinear a las personas de las diferentes Unidades del PNEJP con la dinámica de trabajo que se realizará, indicando sus responsabilidades y solicitando el compromiso de todos. Durante la misma reunión se puede realizar la elaboración y priorización del <i>Product Backlog</i> (Según lo amerite la complejidad del proyecto).	
Entradas	Memorando	
Salidas		

Código y nombre	3. Formar el equipo.	
Descripción	Se identifica el equipo de profesionales con los conocimientos técnicos para desarrollar el proyecto de manera conjunta, llevando a cabo la implementación de cada sprint.	
Rol / Responsable	☐ Scrum Master ☐ Product Owner	
Tareas	Paso	Descripción
	1	Evaluar la experiencia y carga de trabajo del equipo de desarrolladores del AEI y área usuaria.
	2	Realizar una reunión de <i>Kick-off</i> con el equipo designado para transmitir la visión del proyecto, el alcance, revisar el backlog y establecer los acuerdos iniciales.
Comentarios	Es importante que el equipo completo comprenda la esencia del proyecto que están iniciando. Se recomienda elaborar una presentación con la información del proyecto y sus entregables.	
Entradas	Memorando	

Salidas		
Código y nombre	4. Elaborar el Product Backlog o Lista de requerimientos.	
Descripción	Se deberá revisar las principales características del producto del Software que representan las funcionalidades obligatorias, alternativas y opcionales. Se asignará el valor que suponen y el coste estimado para su ejecución, sus iteraciones, plazos y riesgos identificados con sus respectivas acciones de mitigación y contingencia.	
Rol / Responsable	Product Owner	
Tareas	Paso	Descripción
	1	Identificar los objetivos/requisitos detallando la información de tipo, y riesgos identificados.
	2	Realizar la valoración de riesgos y las acciones de mitigación y contingencia indicando la persona responsable por cada uno.
	3	Elaborar la lista de sprints y la estimación de la ejecución de cada uno según la capacidad productiva del equipo <i>scrum</i> .
Comentarios		
Entradas	Memorando	
Salidas	Product Backlog (lista de requerimientos no priorizada)	

Código y nombre	5. Priorizar la lista de requerimientos.	
Descripción	Se debe realizar un ordenamiento de la lista de requerimientos basada en el valor que genera para el área usuaria, las dependencias con otras Unidades orgánicas y las metas del PNEJJP.	
Rol / Responsable	○ Product Owner ○ Equipo scrum	
Tareas	Paso	Descripción
	1	Definir los criterios para la estimación del valor de los requerimientos.
	2	Estimar el esfuerzo de desarrollo de cada requerimiento.
	3	Realizar la planificación de los sprints.
Comentarios		
Entradas	Product Backlog	
Salidas	– Product Backlog ordenado y priorizado. – Especificaciones de requerimientos (o Documento de alcance).	

○ Diagrama de fase de requerimiento. -

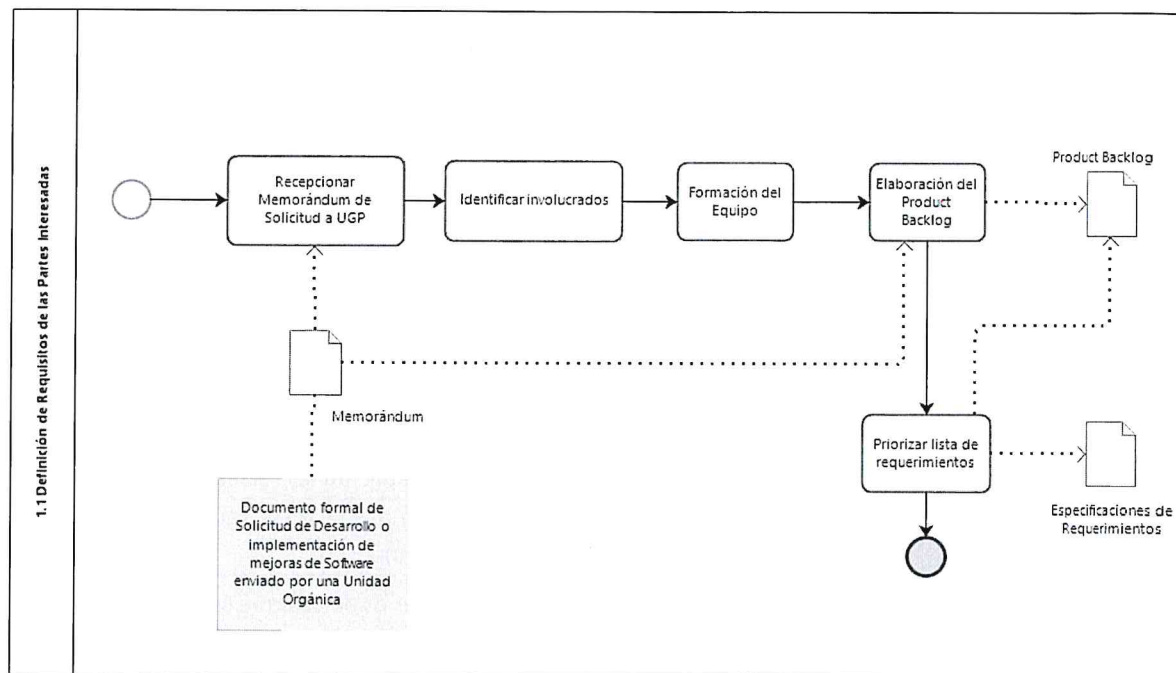


ILUSTRACIÓN 1: ACTIVIDADES DEL PROCESO DE DEFINICIÓN DE REQUISITO DE LAS PARTES INTERESADAS - FASE DE REQUERIMIENTOS

8.2.2 FASE DE ANALISIS Y DISEÑO. - Fase que comprende los subprocesos de *a) Análisis de Requisitos del Software* y *b) Diseño detallado del software*.

a) Análisis de Requisitos del Software. - Tiene como propósito, asignar los requisitos a los elementos de software del sistema determinando las especificaciones técnicas relacionadas, principalmente con las siguientes características:

- Especificaciones funcionales
- Interfaces externas
- Especificaciones de seguridad
- Requisitos de documentación

Adicionalmente, se evaluará los requisitos del software teniendo en cuenta los criterios de trazabilidad, consistencia con respecto a los requisitos del usuario y factibilidad de diseño, construcción, pruebas, operación y mantenimiento.

La información deberá registrarse en un documento de Especificaciones de Casos de Uso conteniendo la descripción de las acciones con un nivel apropiado de detalle (evitando el lenguaje técnico), los actores involucrados y sus respectivos diagramas.

Las acciones realizadas por el personal que participe y/o intervenga en los proyectos de software, deben considerar lo siguiente:

○ Actividades del proceso:

Código y nombre	1. Redactar las Historias de Usuarios.	
Descripción	Por cada requerimiento identificado, se elaborará una Historia de Usuario, que son pequeñas descripciones de los requerimientos de un usuario indicando la funcionalidad, el resultado esperado, los beneficios para su Unidad y los criterios de aceptación.	
Rol / Responsable	○ Product Owner	
Tareas	Paso	Descripción
	1	Identificar el usuario que utilizará la funcionalidad a desarrollarse.
	2	Detallar su necesidad y problemas actuales para entender sus expectativas reales.
	3	Especificar qué producto se requiere (Resultado de la implementación).
Tareas	4	Definir los criterios de aceptación.
	Es deseable que las historias de usuario sean escritas por el usuario, en una frase corta, sin embargo, debe describir el requerimiento del usuario de forma explícita.	
Entradas	Especificaciones de requerimientos. (Documento de Alcance)	
Salidas	Historias de usuarios	

Código y nombre	2. Identificar los Casos de Uso del Sistema.	
Descripción	Definir uno o más escenarios que indican una secuencia de interacciones que se desarrollarán entre el sistema con los usuarios finales (actores) o con otro sistema para conseguir un objetivo específico.	
Rol / Responsable	○ Product Owner ○ Usuario ○ Equipo scrum	
Tareas	Paso	Descripción
	1	Revisar por orden de priorización, los requerimientos aprobados y asociarlos con una o más historia de usuarios.
	2	Identificar los escenarios y sus actores basados en la relación y comportamiento de los requerimientos y la funcionalidad descrita por los usuarios.
	3	Codificar y nombrar cada escenario o caso de uso en una lista priorizada, según la especificación de requerimientos.
Comentarios		
Entradas	– Especificaciones de requerimientos (Documento de Alcance) – Historias de usuarios	
Salidas	Lista de Casos de Usos priorizada	

Código y nombre	3. Realizar la especificación de los Casos de Uso identificados.	
Descripción	Proporcionar los siguientes detalles textuales por cada caso de uso identificado: ○ <u>Nombre</u>: Expresa el resultado objetivo y observable del caso de uso. ○ <u>Descripción</u>: Describe brevemente el rol y objetivo del caso de uso. ○ <u>Flujo de eventos</u>: Describe el comportamiento del sistema. Si se intercambia información, el caso de uso debe ser específico sobre lo que ha transmitido ○ <u>Flujo básico</u>: Describe el comportamiento ideal y principal del sistema. ○ <u>Flujos alternativos</u>: Describe las excepciones o desviaciones del flujo básico. ○ <u>Requisitos especiales</u>: Requisitos no funcionales (legales, de calidad, compatibilidad, rendimiento, usabilidad, por ejemplo) que son específicos de un caso de uso no especificados en el texto del flujo de eventos. ○ <u>Condiciones previas</u>: Estado del sistema que debe estar presente antes del inicio del caso de uso. ○ <u>Condiciones posteriores</u>: Una lista de posibles estados del sistema después de finalizar el caso de uso. ○ <u>Puntos de ampliación</u>: Punto del flujo de eventos de caso de uso en el que se hace referencia a otro caso de uso.	
Rol / Responsable	Equipo Scrum.	
Tareas	Paso	Descripción
	1	Describir el comportamiento de cada caso de uso con un nivel adecuado de detalle según los campos indicados en la descripción de la presente actividad.
	2	Describir las interacciones con actores y otros casos de uso.
	3	Diagramar el escenario utilizando UML (Diagrama de casos de uso).
Comentarios	Se recomienda: ○ Ser cuidadoso al usar estructuras condicionales en la descripción del caso de uso, ya que los usuarios podrían no estar familiarizados con este tipo de estructuras, especialmente si son complejas. ○ Intentar que todos los casos de uso estén descritos al mismo nivel de detalle. ○ En los diagramas de casos de uso, evitar que se crucen las líneas que unen los actores a los casos de uso.	
Entradas	Lista de Casos de Usos priorizada	
Salidas	Documento de Casos de Uso propuesto	



Código y nombre	4. Aprobar el Documento de Casos de Uso del Sistema.	
Descripción	Verificar y validar el documento de casos de uso elaborado con el Product Owner del área usuaria.	
Rol / Responsable	○ Product Owner ○ Scrum Master	
Tareas	Paso	Descripción
	1	Revisar los escenarios documentados para comprobar que reflejan correctamente los requisitos especificados.
	2	Revisar los escenarios documentados para comprobar que son consistentes, factibles y se pueden probar.
Comentarios		
Entradas	– Especificaciones de requerimientos. (Documento de Alcance) – Documento de casos de uso propuesto.	
Salidas	Documento de casos de uso aprobado	

○ Diagrama de fase de Proceso de Análisis de Requisitos del Software:

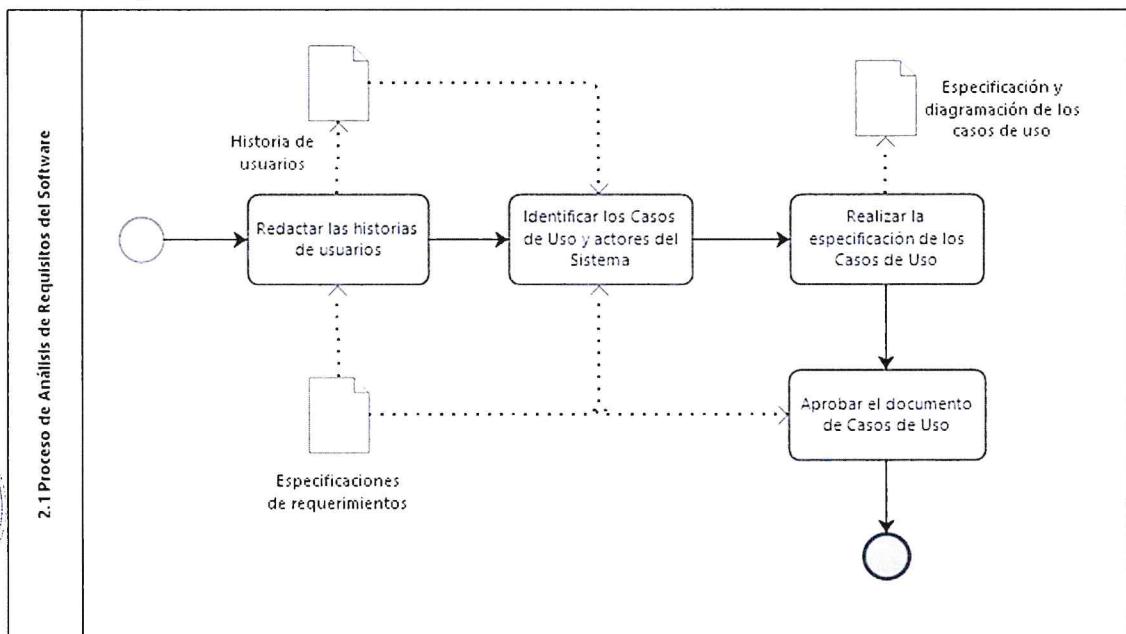


ILUSTRACIÓN 1: ACTIVIDADES DEL PROCESO DE ANÁLISIS DE REQUISITOS DEL SOFTWARE

Nota: Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementará. Simplemente muestran lo que el actor hace o debe hacer para realizar una operación.

- b) Diseño detallado del Software.** - Tiene como propósito proveer una arquitectura para el proyecto de software que permita verificar los requisitos y el diseño del mismo. Dicho sistema de información debe encontrarse detallado para su posterior codificación y prueba, con el fin de lograr los siguientes resultados:

- Descripción y detalle de cada componente del software a desarrollarse.
- Definición de las interfaces externas e internas de cada componente.
- Consistencia en los requisitos y documentos de casos de uso aprobado.

Para transformar los **casos de uso** en un diseño o modelo lógico del sistema a implementar, se debe considerar las siguientes actividades:

- Definir el modelo de Clases (Diagrama UML de clases)
- Definir el modelo relacional o modelo de base de datos (Diagrama Entidad-Relación)
- Desarrollar el prototipo de las interfaces o pantallas principales del Sistema.

Por la complejidad del sistema o necesidad de detallar los componentes del software, se debe evaluar hacer uso de los siguientes diagramas para complementar la documentación de diseño:

- Diagrama de Secuencia y/o de Colaboración
- Diagrama de Estados
- Diagrama de Actividades
- Diagrama de Clases de Implementación
- Diagrama de Componentes o Despliegue

Asimismo, las acciones a integrarse por el personal que participe y/o intervenga en los proyectos de software, deben considerar lo siguiente:

○ **Actividades del proceso:**

Código y nombre	1. Identificar principales interfaces de usuario	
Descripción	Listar las interfaces de usuario más importantes del proyecto (en función de los Casos de Uso del Negocio priorizados).	
Rol / Responsable	Equipo Scrum.	
Tareas	Paso	Descripción
	1	Revisar cada caso de Uso e identificar las interacciones de los usuarios con el sistema.
	2	Proponer la interfaz que corresponda.
Comentarios		
Entradas	Documento de Casos de uso aprobado.	
Salidas	Lista de interfaces.	

Código y nombre	2. Definir la navegación y usabilidad	
Descripción	Planificar la articulación o recorrido entre las interfaces identificadas que permitan orientar a los usuarios en los recorridos al momento de usar el software.	
Rol / Responsable	Equipo Scrum.	
Tareas	Paso	Descripción



	1	Definir los contenidos de las interfaces y sus dependencias.
	2	Organizar los contenidos por secciones.
	3	Elaborar el árbol jerárquico de contenidos.
Comentarios	Esto se puede lograr mediante la elaboración de un mapa de navegación, que son esquemas de organización de la información de una estructura web.	
Entradas	Lista de interfaces.	
Salidas	Árbol de contenidos o Mapa de sitio.	

Código y nombre	3. Realizar el prototipado	
Descripción	Diseñar las interfaces identificadas dibujando las maquetas de los formularios y sus elementos gráficos que permitirá conocer la visión de los programadores que las implementarán. Utilizar herramientas de prototipado de aplicaciones de software que permita crear <i>wireframes</i> (croquis o bosquejo) estáticos o interactivos que simulan la experiencia de uso y navegación del usuario real.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Dibujar los bosquejos de los formularios y sus elementos según los estándares y línea gráfica del PNEJJP.
	2	Simular la navegación mediante enlaces entre los formularios.
	3	Registrar en el documento de diseño.
Comentarios	El principal objetivo de una interfaz de usuario es que este pueda comunicar información a través de ella hacia el sistema, por lo que, se recomienda que las interfaces se diseñen para un fácil uso e intuitivas para el usuario que la utiliza.	
Entradas	○ Lista de interfaces ○ Árbol de contenidos o Mapa de sitio	
Salidas	Documento de Diseño propuesto actualizado.	

Código y nombre	4. Validar el prototipo no funcional	
Descripción	Realizar la presentación a los usuarios de los formularios diseñados con el objetivo de transmitir las ideas y visión del equipo de desarrollo; y retroalimentarlo con las observaciones de los usuarios, con la intención de actualizar el diseño que satisfagan sus necesidades y expectativas.	
Rol / Responsable	Equipo Scrum Usuarios	
Tareas	Paso	Descripción
	1	Realizar la presentación de cada interfaz, sus dependencias y las funcionalidades que realizará.

	2	Obtener y registrar el feedback.
	3	Identificar oportunidades de mejora.
	4	Actualizar el prototipo, el documento del diseño y repetir esta actividad hasta obtener la validación del usuario.
Comentarios	La validación del diseño con el usuario exige una reunión previa entre el equipo scrum para repasar la presentación, escribiendo algunas notas en relación a las siguientes fases: - Empatizar: Asociar las interfaces a las historias de usuarios y Casos de uso. - Definir: Asociar las ideas del usuario con los diseños propuestos de cada interfaz. - Idear: Lista de ideas propuestas por el equipo de desarrollo.	
Entradas	Documento de Diseño propuesto.	
Salidas	Documento de Diseño validado.	

Código y nombre	5. Aprobar el diseño de las interfaces de usuarios.	
Descripción	Aprobar formalmente el Documento de Diseño propuesto inicial, conteniendo las interfaces gráficas validadas por los usuarios.	
Rol / Responsable	Product Owner Scrum Master	
Tareas	Paso	Descripción
	1	Realizar la presentación del prototipo no funcional.
	2	Firmar la aceptación.
Comentarios		
Entradas	Documento de Diseño validado.	
Salidas	Documento de Diseño actualizado y aprobado.	

Código y nombre	6. Identificar los componentes del sistema.	
Descripción	Determinar el conjunto de componentes del sistema que serán desarrollados, organizarlos en paquetes o subsistemas e identificar sus dependencias.	
Rol / Responsable	Equipo scrum Scrum Master	
Tareas	1	Reconocer todos los componentes que conformarán el sistema.
	2	Organizarlos en paquetes o subsistemas.
	3	Según la arquitectura del programa, determinar la lista de componentes que contemplarán las capas (<i>layers</i>).

Comentarios	El enfoque iterativo e incremental de la presente metodología permite identificar componentes de forma progresiva y decidir cuáles desarrollar, cuáles reutilizar y cuáles comprar. Es importante conocer las dependencias de cada componente, ya que pueden ayudar a determinar qué componentes se desarrollarán de forma concurrente y cuáles de forma secuencial. De igual forma, es importante realizar un diseño modular y desacoplado, que puedan probarse de forma individual e integrarse gradualmente para conformar el sistema completo.
Entradas	Documento de Casos de Uso aprobado Documento de Diseño (Interfaces) aprobado
Salidas	Documento de Diseño actualizado

Código y nombre	7. Definir los modelos de clases y E-R (Entidad-Relación).	
Descripción	Utilizando el lenguaje UML, se construirá y documentarán los componentes del sistema elaborando los siguientes diagramas como mínimo: <u>Diagrama de Clase:</u> Describir los componentes utilizando los siguientes campos: - Nombre de la clase. - Atributos - propiedades. - Operaciones o métodos. Indicar también sus relaciones, estas pueden ser: - Asociación - Agregación - Composición <u>Diagrama de Entidad-Relación:</u> Describir las entidades u objetos de datos mediante un modelo relacional que permita posteriormente la construcción de los objetos de base de datos (tablas, campos, procedimientos almacenados, etc.). - Nombre del objeto (tabla) - Campos o atributos indicando el tipo de datos; Numérico, texto, fecha, etc. - Métodos o procedimientos.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Revisar los detalles de las especificaciones de los Casos de uso.
	2	Identificar las clases, sus métodos y atributos.
	3	Elaborar el diagrama de clases para cada Caso de Uso.
	4	Revisar los Casos de uso y las Interfaces diseñadas e identificar las entidades y objetos de datos, así también como sus relaciones.
	5	Actualizar el Documento de Diseño.

Comentarios	Para poder reconocer las clases y entidades del sistema en los detalles de las especificaciones de los casos de uso, se sugiere utilizar la siguiente técnica: a) Los sustantivos son candidatos a clases y entidades de datos (tablas). b) Los adjetivos son candidatos a propiedades de la clase o campos de una tabla. c) Los verbos son candidatos a métodos o procedimientos.
Entradas	○ Documento de Diseño (Interfaces) aprobado. ○ Documento de Casos de Uso aprobados.
Salidas	Documento de Diseño actualizado.

Código y nombre	8. Validar el modelado lógico del Sistema.	
Descripción	Realizar la revisión del diseño o modelo lógico elaborado para determinar: - Es correcto, consistente y con trazabilidad hacia los requisitos. - Cumple con los estándares y especificaciones. - Implementa la secuencia correcta de eventos, entradas, salidas e interfaces.	
Rol / Responsable	Equipo Scrum Scrum Master	
Tareas	Paso	Descripción
	1	Realizar la presentación de los diagramas elaborados.
	2	Obtener y registrar el feedback del equipo.
	3	Identificar oportunidades de mejora.
	4	Actualizar el modelo lógico y repetir esta actividad hasta obtener la validación del equipo.
Comentarios		
Entradas	Documento de Diseño propuesto	
Salidas	Documento de Diseño validado	

Código y nombre	9. Aprobar el diseño de las interfaces de usuarios.	
Descripción	Aprobar el Documento de Diseño validado por el equipo scrum, conteniendo las interfaces gráficas y el modelo lógico del Sistema que se desarrollará.	
Rol / Responsable	Equipo Scrum Scrum Master	
Tareas	Paso	Descripción
	1	Realizar la aprobación.
	2	Comunicar el inicio de las fases iterativas de Desarrollo, pruebas y despliegue a los involucrados del proyecto.
Comentarios	El enfoque iterativo e incremental de la presente metodología permite realizar las mejoras al diseño e interfaces durante la ejecución de los <i>Sprints</i> planificados.	
Entradas	Documento de Diseño actualizado	
Salidas	Documento de Diseño aprobado	

o Diagrama de fase de Proceso de diseño del detalle del Software:

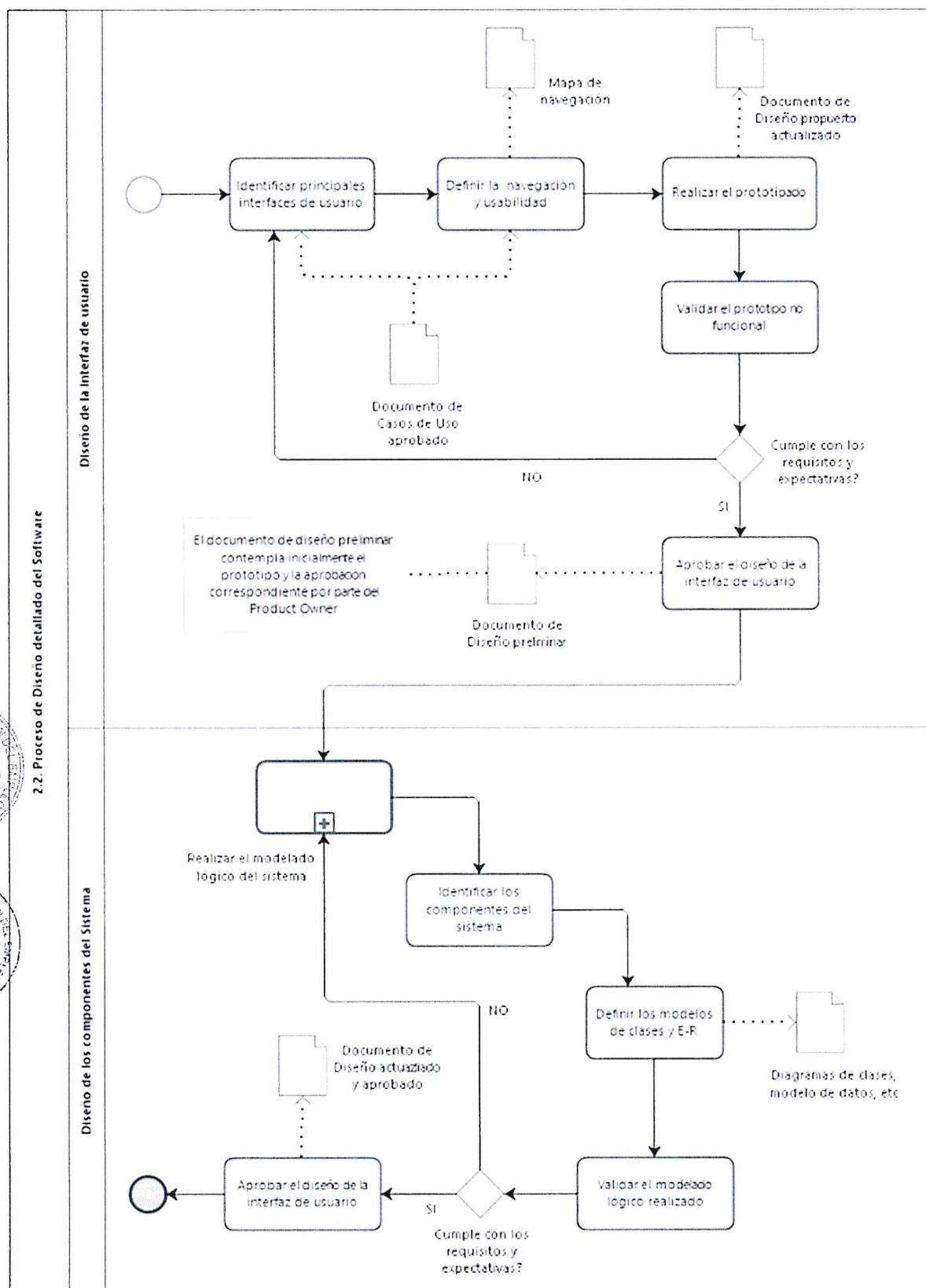


ILUSTRACIÓN 2: ACTIVIDADES DEL PROCESO DE DISEÑO DEL DETALLE DEL SOFTWARE

8.2.3 FASE DE IMPLEMENTACIÓN. - Fase que comprende el subproceso *Construcción del Software*.

- a) **Construcción del Software.** - Tiene como objetivo producir unidades del software ejecutables y base de datos que reflejen de manera correcta el diseño o modelo lógico del software.

El equipo scrum iniciará sus labores de desarrollo del código fuente y los objetos de base de datos según la planificación del *Product Backlog (Sprints)* realizando las reuniones diarias de estado (*Daily standup o Daily scrum meeting*) con el fin de resolver problemas específicos, compartir la visión general de lo realizado y pendiente, recolectar información de atrasos de actividades y comprometer a los miembros del equipo.

El Scrum master es responsable de resolver los impedimentos que se planteen en la reunión de estado o, de no poder hacerlo, asegurarse que alguien más en el equipo resuelva el inconveniente rápidamente.

Las acciones realizadas por el personal que participe y/o intervenga en los proyectos de software, deben considerar lo siguiente:

- **Actividades del proceso**

Código y nombre	1. Preparar condiciones para la implementación	
Descripción	Disponer de las herramientas para el desarrollo de las unidades de software y base de datos para dar soporte a la codificación y pruebas unitarias. Basado en las buenas prácticas para ambientes de desarrollo de software, se deberá: ○ Contar con los accesos a los servidores compartidos. ○ Incluir repositorios para respaldo y control de versiones. ○ Incluir réplicas de los componentes para las interoperaciones que deba contemplar el software. ○ Utilizar nombres de dominio diferentes para los ambientes de producción, pruebas y desarrollo. ○ Tener instalado el mismo software (y versión) del software requerido en los ambientes de producción y pruebas.	
Rol / Responsable	Equipo Scrum Scrum Master	
Tareas	Paso	Descripción
	1	Determinar las necesidades del software
	2	Instalar los requerimientos del software y componentes de interoperabilidad necesarios.
	3	Validar los accesos, conectividad y uso compartido a los equipos de desarrollo.
	4	Preparar el ambiente e interfaces de programación (IDE) para la codificación del software y base de datos.

Comentarios	Cada vez adquiere una mayor importancia la definición de procedimientos adecuados para administrar los ambientes de desarrollo de software, considerando que el PNEJP, en la actualidad, demanda la ejecución de múltiples proyectos simultáneos, con tiempos de puesta en producción exigentes, lo que conlleva a muchos desarrolladores de software, tanto internos como externos a la organización, compartiendo los mismos ambientes de desarrollo. Se sugiere, además, contar con herramientas para generar documentación automáticamente.
Entradas	○ <i>Product Backlog</i> priorizado ○ Documento de Casos de uso aprobado ○ Documento de Diseño aprobado
Salidas	Ambientes de desarrollo instalados

Código y nombre	2. Implementar los componentes de la unidad de Software	
Descripción	Programar cada componente de la unidad de software contemplada en el sprint planificado siguiendo los estándares de nomenclatura: ○ Implementación de las interfaces aprobadas y su funcionalidad. ○ Codificación de las bibliotecas de clase en las capas de la arquitectura diseñada (lógica de negocio, acceso a datos, etc.) ○ Programar las funciones y métodos para el correcto uso de los recursos de memoria, almacenamiento y procesamiento.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Realizar la programación de los componentes.
Comentarios	Se recomiendan las siguientes prácticas de programación de software, evitando: ○ El uso de variables globales ○ El uso de sentencias <i>goto</i>, <i>break</i> y <i>continue</i> ○ Funciones y procedimientos demasiado largas ○ Copiar y duplicar código	
Entradas	Documento de Diseño aprobado	
Salidas	Código fuente	

Código y nombre	3. Implementar los objetos de Base de datos
Descripción	Programar los objetos de base de datos descritos en los diagramas de Entidad-Relación del Documento de diseño (Modelo de datos): ○ Tablas y sus atributos. ○ Consultas ○ Procedimientos y operaciones ○ Informes
Rol / Responsable	Equipo Scrum

Tareas	Paso	Descripción
	1	Realizar la programación de los objetos de Base de Datos asignando el tipo de dato.
	2	Establecer las relaciones entre tablas.
	3	Configurar y optimizar los procedimientos, funciones y consultas.
Comentarios		
Entradas		Documento de Diseño aprobado
Salidas		Código fuente

Código y nombre	4. Realizar pruebas unitarias	
Descripción	Comprobar el correcto funcionamiento de una unidad de código del software escribiendo (automatizando) los casos de prueba para cada función no trivial o método en el módulo, de forma que cada caso sea independiente del resto. El objetivo de esta actividad es aislar cada parte del software e identificar los errores de manera individual proporcionando las siguientes ventajas: ○ Fomentar el cambio para mejorar su estructura ○ Simplificar la integración posterior. ○ Documentar el código. ○ Desacoplar las interfaces y sus implementaciones. ○ Identificar fácilmente los errores.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Determinar las funciones no triviales (Unidad del Código con Estructura de Complejidad Alta).
	2	Programar las pruebas a las funciones identificadas.
	3	Verificar que la implementación cubra los resultados esperados.
Comentarios	4	Descubrir los errores de lógica, configuración, programación.
Entradas		Documento de Casos de uso aprobados Documento de Diseño aprobado Código fuente
Salidas		Código fuente probado

Código y nombre	5. Identificar los componentes a integrar	
Descripción	Determinar las dependencias de los componentes de software con el objetivo de identificar las estructuras de programación y los tipos de datos que serán integrados para asegurar el cumplimiento de las funcionalidades esperadas.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción



PERÚ

Ministerio de Trabajo
y Promoción del Empleo

	1	Distinguir las propiedades y métodos de los componentes relacionados en el Documento de Diseño.
	2	Elaborar una lista de componentes a integrar.
Comentarios	Se recomienda determinar el orden de la implementación y combinación de las clases.	
Entradas	Documento de Diseño aprobado	
Salidas	Lista de componentes a integrar ordenados por dependencia.	

Código y nombre	6. Implementar la integración	
Descripción	Esta actividad de desarrollo y programación permitirá combinar incrementalmente los componentes de software implementados. La integración se puede realizar: a) Entre componentes del mismo subsistema b) Entre subsistemas de un sistema completo	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Realizar la programación de la integración de los componentes de software.
Comentarios		
Entradas	Documento de Diseño aprobado	
Salidas	Código fuente	

Código y nombre	7. Realizar las pruebas de integración	
Descripción	Comprobar el correcto funcionamiento de las unidades de software integradas en conjunto. El resultado garantizará que el proyecto de software pueda avanzar hacia la siguiente iteración o <i>Sprint</i> .	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Determinar las comunicaciones funcionales entre los componentes integrado.
	2	Realizar las pruebas funcionales.
	3	Verificar que la integración cubra los resultados esperados.
Comentarios	4	Descubrir los errores de lógica, configuración y programación.
	Se centra principalmente en probar la comunicación entre los componentes y sus comunicaciones ya sea hardware o software. Durante este proceso en el cual se verifican los distintos tipos de integración, el equipo scrum deberá ensamblar los módulos independientes, para ir formando el software progresivamente (iteración).	
	Documento de Casos de uso aprobados Documento de Diseño aprobado Código fuente	
	Código fuente probado	





PERÚ

Ministerio de Trabajo
y Promoción del Empleo

Código y nombre	8. Entregar unidad de Software	
Descripción	Realizar el pase o instalación de las unidades de software en los ambientes de pruebas para su presentación a los usuarios y el <i>Product Owner</i> .	
Rol / Responsable	Equipo Scrum Scrum Master	
Tareas	Paso	Descripción
	1	Determinar las necesidades y requerimientos del software.
	2	Instalar la unidad de software.
	3	Validar el acceso a los enlaces de pruebas y la visualización de interfaces del software.
Comentarios	El ambiente de pruebas debe ser homólogo al ambiente de producción.	
Entradas	Código fuente	
Salidas		



○ Diagrama de proceso de Construcción del Software

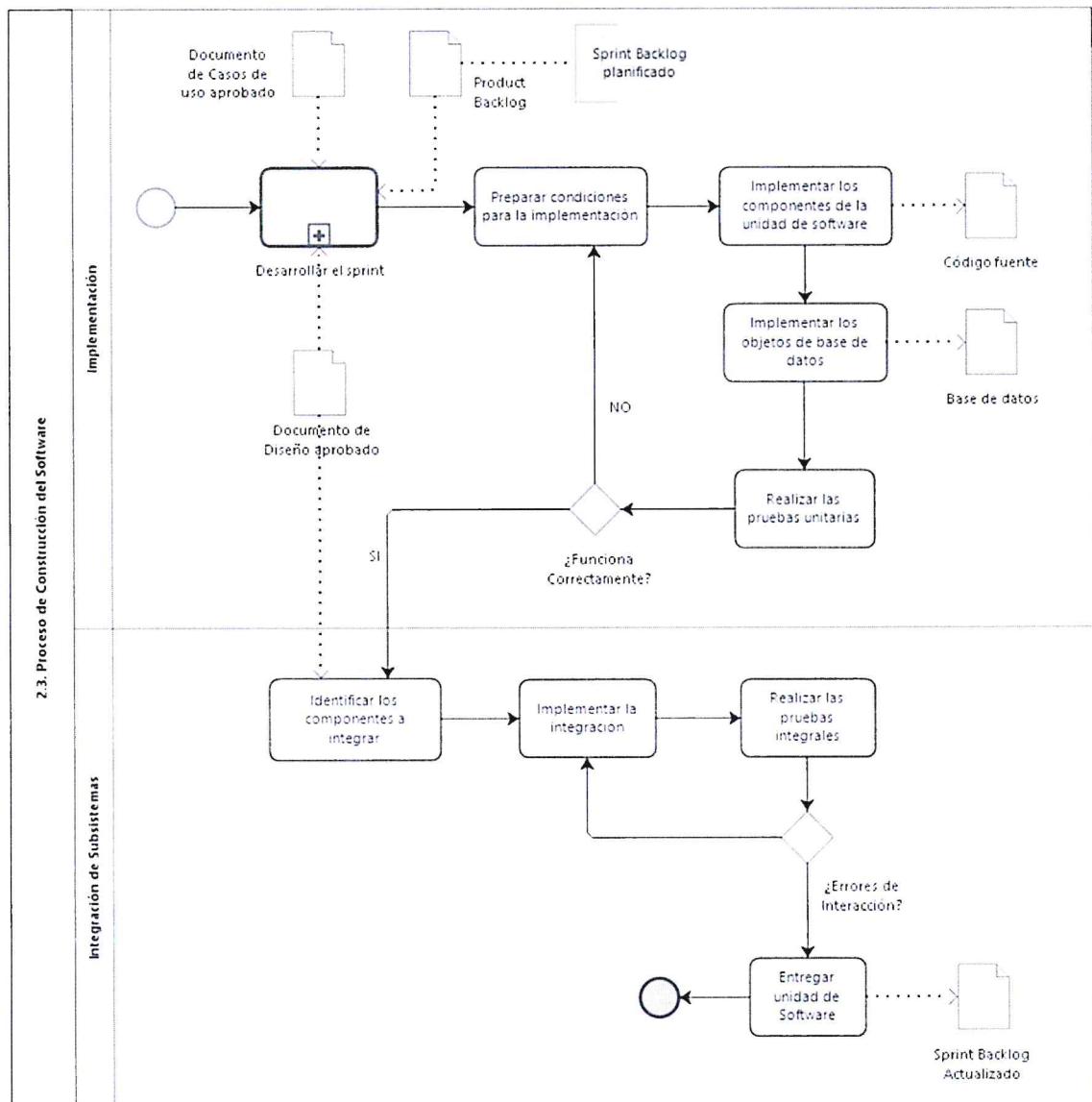


ILUSTRACIÓN 3: ACTIVIDADES DEL PROCESO DE CONSTRUCCIÓN DEL SOFTWARE

8.2.4 FASE DE DESPLIEGUE O TRANSICIÓN. - Fase que comprende los subprocesos de a) *Instalación del Software* y b) *Soporte de la aceptación del Software*.

- a) **Instalación del Software.** – Comprende la instalación del producto de software que satisfaga los requisitos acordados con los usuarios.

Las acciones realizadas por el personal que participe y/o intervenga en los proyectos de software, deben considerar lo siguiente:

○ **Actividades del proceso**

Código y nombre	1. Generar producto de software	
Descripción	Proporcionar el paquete de software formado por los programas ejecutables, sus dependencias y requisitos funcionales, bibliotecas de clases y otros tipos de archivos (imágenes, localizaciones, elementos de diseño, etc.) que serán instalados.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Identificar dependencias y requisitos preliminares de funcionamiento del software.
	2	Compilar los archivos ejecutables.
	3	Elaborar los scripts de base de datos.
	4	Generar el paquete de software.
Comentarios	Existen herramientas que ayudan a crear paquetes de instalación de software. Se recomienda evaluar su adquisición y uso.	
Entradas	Código fuente y Objetos de base de datos	
Salidas	Paquete de software conteniendo: - Archivos ejecutables - Requisitos de software - Scripts de base de datos - Imágenes y hojas de estilos. - Archivos de recursos y localización. - Otros	

Código y nombre	2. Elaborar nota de pase a producción	
Descripción	Registrar los elementos que conforman el paquete de software consignando el número de versión y los pasos y/o procedimientos en orden para su instalación.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Describir el motivo del pase y la funcionalidad que representa.
	2	Registrar la información de elementos.
	3	Registrar los pasos de instalación.
Comentarios	Es importante llevar una bitácora de pases a producción indicando la persona del equipo scrum responsable, fecha y hora de instalación, elementos que fueron desplegados y motivo o funcionalidad que afecta.	
Entradas	Paquete de software	
Salidas	Nota de pase a producción	

Código y nombre	3. Desplegar el software	
Descripción	Realizar una copia de respaldo e instalar el paquete de software según el procedimiento descrito en las Notas de pase a producción.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Realizar un <i>backup</i> de los elementos de software que serán actualizados o reemplazados (según procedimientos del área de soporte).
	2	Ejecutar los procedimientos de instalación indicados en la nota de pase a producción.
	3	Registrar el evento en una lista o bitácora de pase a producción.
	4	Comunicar al resto del equipo scrum.
Comentarios		
Entradas	Nota de pase a producción	
Salidas	Software desplegado	

Código y nombre	4. Realizar prueba operacional	
Descripción	Realizar las pruebas de funcionamiento del software en el ambiente de producción final.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Verificar la operatividad del software.
	2	Verificar la funcionalidad instalada.
	3	Registrar las incidencias (si aplica).
	4	Comunicar los resultados.
Comentarios		
Entradas	Software desplegado	
Salidas	Resultados de la prueba operacional	

Código y nombre	5. Revertir el pase o <i>Rollback</i>	
Descripción	Reestablecer el software mediante operaciones de reversión de versionamiento de su repositorio al estado anterior o mediante el despliegue de su copia de respaldo.	
Rol / Responsable	Equipo Scrum	
Tareas	Paso	Descripción
	1	Realizar la reversión del software.
	2	Comunicar el restablecimiento.
Comentarios		
Entradas	Resultados de la prueba operacional	
Salidas	Software restablecido	

Código y nombre	6. Entregar software al usuario	
Descripción	Notificar al Product Owner y usuarios de la Unidad Orgánica del PNEJJP, la puesta en marcha del software en los ambientes de producción. Durante esta notificación (formal o informal), se deberá indicar los enlaces de acceso al software y solicitar la reunión para iniciar el proceso de aceptación formal del software.	
Rol / Responsable	Equipo Scrum Product Owner Scrum master	
Tareas	Paso	Descripción
	1	Elaborar la comunicación.
	2	Enviar la comunicación a todas las personas involucradas del proyecto.
Comentarios	Esta actividad también puede considerarse como parte de la fase de transición, por lo tanto, en la siguiente versión de la presente metodología se deberá evaluar pertenecer a dicha fase.	
Entradas	Software operando en el ambiente productivo	
Salidas	Notificación del pase a producción.	

○ Diagrama de instalación del Software

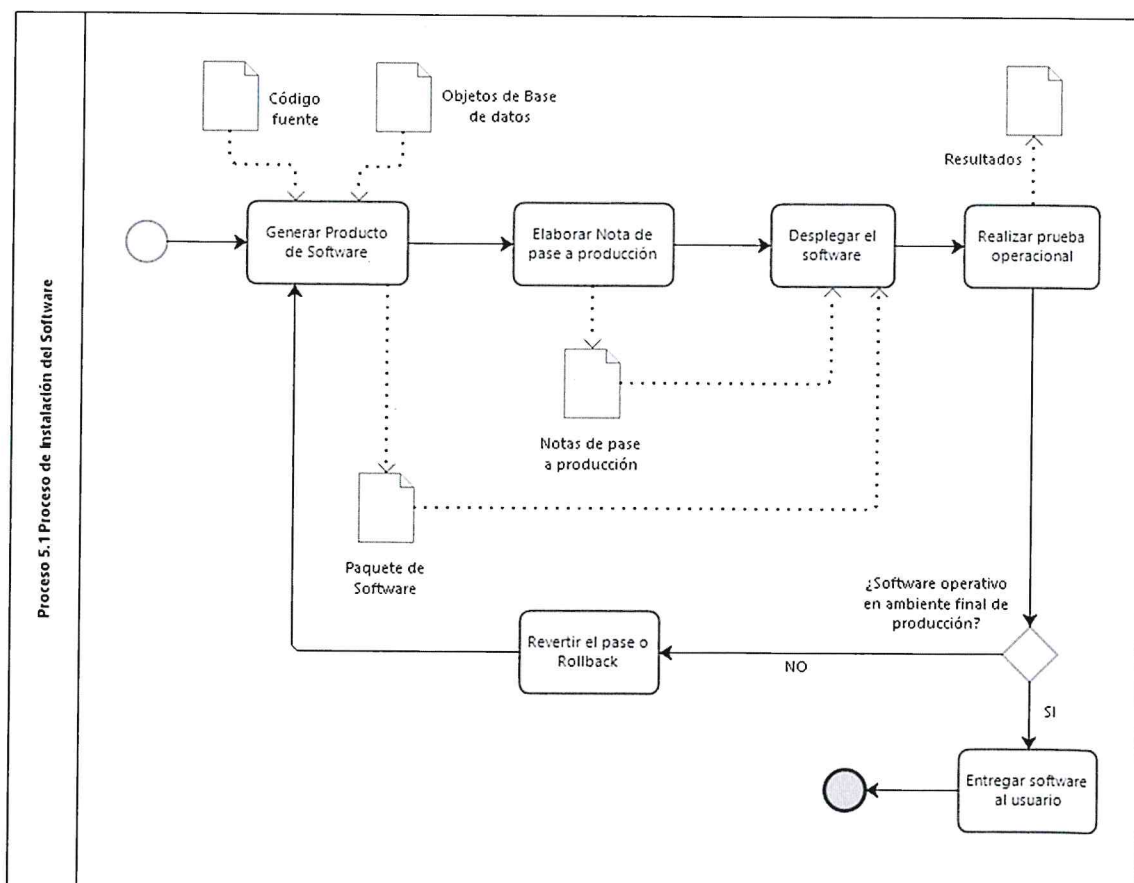


ILUSTRACIÓN 4: ACTIVIDADES DE INSTALACIÓN DEL SOFTWARE



PERÚ

Ministerio de Trabajo
y Promoción del Empleo**8.2.5 Soporte de la aceptación del Software. -**

El propósito del Proceso de Soporte de la Aceptación del Software es asistir al usuario final para obtener la aprobación formal del software y la confianza en que el producto satisface sus requisitos.

Las acciones realizadas por el personal que participe y/o intervenga en los proyectos de software, deben considerar lo siguiente:

○ **Actividades del proceso:**

Código y nombre	1. Documentar la funcionalidad del software	
Descripción	Elaborar o actualizar el manual de usuario progresivamente una vez que el <i>sprint</i> ha sido ejecutado, validado y la unidad de software programada e integrada está operando en los ambientes de producción del PNEJJP.	
Rol / Responsable	Equipo Scrum Scrum master	
Tareas	Paso	Descripción
	1	Elaborar o actualizar el manual de usuario.
	2	Incorporar la documentación en el repositorio del proyecto.
	3	Notificar al <i>Product Owner</i> y coordinar la reunión para realizar la revisión y aprobar formalmente el software.
Comentarios	Se recomienda centralizar la documentación en el sistema de gestión de proyectos de informática del AEI llamado <i>OpenProject</i> .	
Entradas	Software operando en el ambiente productivo	
Salidas	Manual de usuario	

Código y nombre	2. Apoyar la aceptación del software	
Descripción	Realizar la revisión del producto de software y su documentación (manual de usuario) considerando los resultados de las pruebas de usuario para obtener la aprobación formal del <i>Product Owner</i> en representación de la Unidad Orgánica del PNEJJP.	
Rol / Responsable	Scrum master Product Owner Usuarios	
Tareas	Paso	Descripción
	1	Comprobar que el producto de software está disponible.
	2	Revisar y validar el manual del usuario.
	3	Elaborar el acta de aceptación.
	4	Proceder a firmar el acta por parte del <i>Product Owner</i> y el <i>Scrum master</i> .
Comentarios		

Entradas	Documento de resultado de las pruebas de usuario. Software operando en el ambiente productivo. Manual de usuario.
Salidas	Acta de aceptación

Código y nombre	3. Capacitar a los usuarios	
Descripción	Proporcionar el entrenamiento inicial y continuo a los usuarios nuevos del software.	
Rol / Responsable	Equipo scrum Product owner Usuarios	
Tareas	Paso	Descripción
	1	Identificar los usuarios a capacitar.
	2	Programar los días de capacitación.
	3	Comunicar el cronograma y la lista de participantes.
	4	Realizar la capacitación
Comentarios		
Entradas	Documento de resultado de las pruebas de usuario. Software operando en el ambiente productivo. Manual de usuario.	
Salidas	Acta de aceptación	

Diagrama de proceso de soporte de la aceptación del Software:

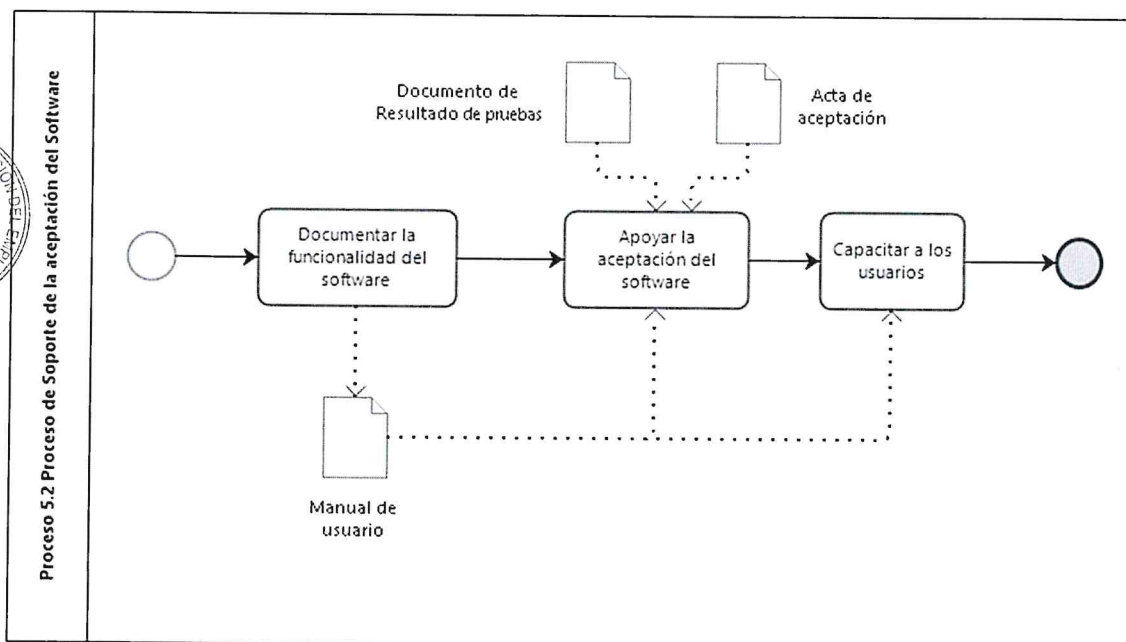


ILUSTRACIÓN 5: ACTIVIDADES DEL PROCESO DE SOPORTE DE LA ACEPTACIÓN DEL SOFTWARE

8.3 Del mantenimiento del Software. - El proceso de mantenimiento se realizará en atención a los nuevos requerimientos (parte de requerimientos), ya sea para actualizaciones y adecuaciones de un producto software.

La definición de nuevos requerimientos sobre un producto software que se encuentre en producción, se dará por nuevos requerimientos, planteados por los colaboradores del PNEJJP, en el marco de optimización de procesos de negocio o específicamente en la decisión de retirar el producto software.

Se debe considerar cuando se requieran ejecutar mantenimientos correctivos o evolutivos (mejoras del software), estos deberán ser tratados como un proyecto de desarrollo de baja complejidad y, en este caso, solo se requerirán los procesos descritos en la sección 8.4 Esquema de la Metodología.

El proceso de mantenimiento debe asegurar la trazabilidad de los requerimientos desde su concepción hasta su puesta en producción o retirada de producción, y las fases para el Mantenimiento de Software serán las mismas que se aplican al Desarrollo del Software:

- Requerimientos (de mantenimiento)
- Análisis y diseño
- Desarrollo
- Pruebas
- Transición



8.4 Esquema de metodología. - La metodología descrita es aplicable en el desarrollo, operación y mantenimiento de productos del software independientemente que sea desarrollados por el equipo interno (equipo de desarrollo de sistemas del AEI) o externamente (proveedor) al PNEJJP.



Los proyectos que ejecutará el Área de Estadística e Informática, serán definidos bajo los criterios de tiempo, riesgos y costos planificados, los cuales se clasificarán de acuerdo a la complejidad “baja”, “media” o “alta”, con la finalidad de evaluar en qué proceso, tareas y artefactos propuestos serán necesarios incluirlos durante su ejecución.

IX. DISPOSICIONES COMPLEMENTARIAS Y FINALES

- 
- 9.1** El Área de Estadística e Informática será responsable de verificar y supervisar el estricto cumplimiento de cada una de las etapas del proceso de desarrollo de sistemas de información descritas en la presente Directiva.
 - 9.2** Todo proyecto de desarrollo, mantenimiento y/o adquisición de software, deberá ser canalizado por el Área de Estadística e Informática.
 - 9.3** Las disposiciones de la presente Directiva están basadas en la Norma Técnica Peruana NTP-ISO/IEC 12207:2016 – Ingeniería de Software y Sistemas. Procesos del ciclo de vida del software. 3a Edición.

X. ANEXOS



Formatos de documentos del proceso de ciclo de vida de Software.