

03063 32



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

POSGRADO EN CIENCIA E INGENIERIA DE LA COMPUTACION

GUIA DE TSPiSM USANDO UML

TESIS

**QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS**

**PRESENTA:
PAUL OSWALDO SALDAÑA NAVA**

DIRECTORA DE TESIS: DRA. HANNA OKTABA

MEXICO, D. F.

OCTUBRE 2001



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A la Dra. Hanna Oktaba

Por todas sus enseñanzas, su paciencia, su apoyo y su gran calidad como persona. Le agradezco infinitamente por impulsarme y motivarme a hacer mejor las cosas.

**A la M. en C. Guadalupe Ibargüengoitia,
al Dr. Fernando Gamboa,
al M. en C. Reynaldo Alanis y
al M. en C. Jorge Ortega**

Por el tiempo tan valioso que me han brindado para revisar y mejorar este trabajo. ¡Muchísimas gracias!.

**A Sergio Ramos Chávez, Carmen Dolores Alvarez Sánchez, Fabián de la Cruz,
Leonardo Ramírez Fernández, Alejandro Talavera Rosales, Jaime Ruíz García y
Mario Hernández Mayorga.**

Por su contribución en el diseño, guiones y ejemplos que permitieron hacer del sitio de Internet de este trabajo una realidad.

A Dios

Por darme la oportunidad de concluir otro sueño.

A mi Mama

Por confiar siempre en mi, por sus consejos, sus desvelos y porque siempre esta ahí cuando la necesito.

A mi Papa (q.e.p.d.)

Siempre has estado conmigo, has sido y seguirás siendo un ejemplo a seguir.

A mi abuelita Susana (q.e.p.d.)

Por su cariño, porque hasta el último momento demostró su fortaleza.

A mis abuelitos Juan y Elena

Por apoyarme y quererme. Gracias

A mi hermano Álvaro

Por sus consejos, su apoyo y por ayudarme siempre que se lo pido.

A Alejandra

Por darme el aliento necesario para seguir. Por todo su amor, su paciencia. Este es un sueño de los dos. Gracias por estar siempre conmigo pollito.

A todos mis familiares y amigos

Gracias porque siempre he tenido alguna palabra de aliento, alguna palmada de ánimo o alguna sonrisa de apoyo.

ÍNDICE

Introducción	1
1. Marco Teórico	5
1.1 Team Software Process (TSPi)	6
1.1.1 Principios y objetivos de TSPi	7
1.1.2 Estrategia de TSPi	8
1.1.3 Estructura y formato de TSPi	10
1.1.4 Estrategia de desarrollo en ciclos	11
1.1.5 Fases del Ciclo TSPi	11
1.1.6 Equipo TSPi	14
1.2 Método aplicado para la construcción del modelo de TSPi	15
2. Guía de TSPi usando UML	18
2.1 Descripción de los diagramas	21
2.2 Diagrama General de TSPi	24
2.3 Primer Ciclo	26
2.3.1 Lanzamiento del Primer Ciclo de TSPi	28
2.3.2 Estrategia del Primer Ciclo de TSPi	32
2.3.3 Planeación del Primer Ciclo de TSPi	36
2.3.4 Requerimientos del Primer Ciclo de TSPi	40
2.3.5 Diseño del Primer Ciclo de TSPi	44
2.3.6 Implementación del Primer Ciclo de TSPi	48
2.3.7 Pruebas del Primer Ciclo de TSPi	54
2.3.8 PostMortem del Primer Ciclo de TSPi	57
2.4 Ciclo n	61
2.4.1 Lanzamiento del Ciclo n de TSPi	63
2.4.2 Estrategia del Ciclo n de TSPi	68
2.4.3 Planeación del Ciclo n de TSPi	71
2.4.4 Requerimientos del Ciclo n de TSPi	75
2.4.5 Diseño del Ciclo n de TSPi	79
2.4.6 Implementación del Ciclo n de TSPi	83
2.4.7 Pruebas del Ciclo n de TSPi	89
2.4.8 PostMortem del Ciclo n de TSPi	93
2.5 Realizar Pruebas	97
2.6 Inspecciones	100
3. Conclusiones	103
Bibliografía	107
Anexo A	109
Anexo B	121
Glosario	123

Introducción

En esta sección se presenta el entorno en el cual surge la propuesta para realizar el presente trabajo. Se define su objetivo y se describen brevemente las partes que lo conforman.

En la actualidad, la industria de software requiere de ingenieros de desarrollo bien calificados y preparados. Para que éstos construyan productos de alta calidad, que permitan competir a las empresas, es necesario que posean ciertas cualidades y características.

De acuerdo al Consejo Asesor Industrial de la Universidad Aeronáutica Embry-Riddle [Humphrey, 2000], algunas cualidades que son deseables encontrar en un ingeniero de software son:

- Actitud profesional hacia el trabajo
- Buena comunicación (oral y escrita)
- Habilidad para formar parte de un equipo de trabajo
- Conocimientos de diseño de software y experiencia en el uso de procesos de software
- Conocimiento y buenos fundamentos en cómputo
- Habilidad para aprender nuevas tecnologías

A causa de la alta complejidad y amplitud en los productos de software, los sistemas en su mayoría son desarrollados en equipos de trabajo. Esta cualidad se ha hecho indispensable para cualquier desarrollador de software.

Trabajar en equipo no es tan fácil como podría suponerse. Se necesita un método para saber como trabajar en forma conjunta, para definir el trabajo que debe hacerse y como plantear una estrategia para realizarlo. Para poder mantener una buena relación de equipo es necesario tener metas comunes, estar de acuerdo en el plan de acción y tener un liderazgo apropiado. Se necesita que cada miembro del equipo entienda las virtudes y carencias de los otros miembros, que los apoye y esté dispuesto a pedir ayuda cuando se necesite. Trabajar en equipo no es una habilidad que se adquiere al nacer. Se obtiene a través de la práctica y se mejora día a día con la experiencia.

Para obtener un producto de calidad no solamente es necesario saber trabajar en equipo sino seguir un proceso de software bien definido. Existen diferentes modelos para el desarrollo de software como CMM-SW [CMM, 1995], ISO12207 [ISO/IEC 12207,1995], PSP [Humphrey,1995], ISO15504(SPICE) [ISO/IEC 15504, 1999] y TSPi [Humphrey, 2000]. Sin embargo, todos los modelos anteriores se muestran en una forma textual muy amplia y en idioma inglés, por ejemplo TSPi consta de 447 paginas incluyendo formatos. No poseen ejemplos prácticos que permitan facilitar su comprensión. Para poder implementarlos, las compañías e ingenieros de software tienen que dedicar gran cantidad de tiempo y esfuerzo para asimilar e interpretar estos modelos.

Se realizó la elección de TSPi por ser un proceso de software que se enfoca al trabajo en equipo y por tener una presentación en forma de material didáctico.

El objetivo del presente trabajo es generar una guía gráfica de TSPi (Introduction to Team Software Process) usando diagramas de actividades de UML (Lenguaje Unificado de Modelado) [UML, 1999], que facilite al lector la comprensión de TSPi. La guía contendrá ejemplos prácticos que ilustren el proceso de desarrollo de software definido en TSPi. El método de trabajo para construir la guía se basa en realizar una abstracción del contenido del libro de TSPi, modelando su estructura con diagramas de actividades y organizando la información de tal manera que facilite el aprendizaje y seguimiento del proceso de desarrollo de TSPi.

TSPi es un proceso de desarrollo iterativo de software que incluye un primer ciclo de desarrollo y los ciclos de desarrollo n o subsecuentes. Cada ciclo tiene definidas 8 fases: lanzamiento, estrategia, planeación, requerimientos, diseño, implementación, pruebas y postmortem. Cada fase tiene actividades que deben ser realizadas por determinados roles y productos que se generan a partir de dichas actividades.

El alcance de la guía comprenderá lo siguiente:

- Generación de los siguientes diagramas de actividades:
 - TSPi general.
 - Primer ciclo y ciclo n.
 - Por cada fase de los ciclos primero y n.
 - Pruebas.
 - Inspecciones.
- Descripción de cada una de las actividades definidas en cada fase, definiendo pasos a seguir para concluir las.
- Ejemplos de los productos generados por las actividades para cada fase.

El proceso de TSPi ha sido experimentado en cursos de la licenciatura en Ciencias de la Computación (Facultad de Ciencias 2000-1 y 2000-2) y de la Maestría en Ciencia e Ingeniería de la Computación (2000-2) en la Universidad Nacional Autónoma de México por la Dra. Hanna Oktaba y M. en C. Guadalupe Ibarguengoitia. En particular, el contenido de la guía se basa en la experiencia personal obtenida del curso Ingeniería de Software Orientada a Objetos. Durante este curso se desarrolló un proyecto real para la Coordinación de la Maestría siguiendo el proceso de TSPi.

Este material está dirigido principalmente a:

- Maestros y alumnos de cursos de Ingeniería de Software que quieren aprender las mejores prácticas por vía experimental siguiendo TSPi.
- Administradores y desarrolladores de sistemas de cómputo que busquen un proceso de software concreto y con los elementos necesarios que ayuden a incrementar la calidad.

El trabajo se divide en tres secciones principales: Marco teórico, Guía de TSPi usando UML y Conclusiones. La primera sección introduce y muestra la estructura general de TSPi. También justifica el uso de diagramas de actividades y la forma en que fue creada la guía. La segunda sección presenta la guía de TSPi iniciando con una breve descripción de las partes que conforman cada uno de los diagramas, posteriormente fase a fase explica cada una de las actividades que deben realizarse para poder concluir con éxito. La tercera sección muestra las conclusiones obtenidas del presente proyecto y los trabajos que podrían realizarse a futuro partiendo de la guía.

Capítulo 1

Marco Teórico

En este capítulo se muestra un panorama general de TSPi, sus objetivos y estructura. También se describe la forma en que fue desarrollado el presente trabajo.

1.1 Team Software Process (TSPi)

El contenido de esta sección fue tomado del libro *Introduction to the Team Software Process* [Humphrey, 2000]. Su objetivo es proporcionar un panorama general que introduzca al lector al proceso de trabajo de un equipo de software de acuerdo a TSPi.

TSPi [Humphrey, 2000] es la propuesta de Watts Humphrey para el desarrollo de sistemas de software en equipo. Es una versión reducida de TSP (Team Software Process) para la educación y capacitación de los ingenieros de software. Fue desarrollado en el SEI (Software Engineering Institute) como proceso industrial que contempla desarrollos de hasta 20 ingenieros de software trabajando para proyectos grandes. TSPi contiene los mismos principios que TSP.

1.1.1 Principios y Objetivos de TSPi

El objetivo principal de TSPi es completar con éxito a través de varios ciclos de desarrollo incremental un pequeño proyecto de software con calidad, siguiendo fielmente el proceso y manteniendo durante cada ciclo de desarrollo un equipo eficiente y colaborativo.

TSPi es una guía paso a paso para lograr un proyecto de software en equipo. Enseña como aplicar conocimientos de ingeniería de software y procesos en un ambiente de trabajo en equipo. Define claramente los roles que cada miembro debe desempeñar, así como sus responsabilidades. Nos muestra qué debemos hacer, cómo hacerlo y cuándo hacerlo. Permite practicar y desarrollar una buena actitud de equipo de trabajo.

Para lograr un aprendizaje óptimo de TSPi, se debe de tener experiencia en programación y un conocimiento previo de PSPSM (Personal Software Process) [Humphrey, 1997]. PSP es un método práctico, que fue diseñado para guiar a los ingenieros de software en la realización de un buen trabajo. Enseña como aplicar métodos avanzados en cada una de las tareas diarias. Adicionalmente proporciona métodos para realizar estimados y planeación que puedan ser comparados con el trabajo realizado. Es recomendable trabajar con equipos capacitados en PSP, pues permite un seguimiento más simple y rápido de los procesos de TSPi y se cuenta con una experiencia a detalle de la forma de realizar la planeación de un proyecto.

TSPi proporciona un balance entre proceso, producto y equipo de trabajo. Sus fases y tareas están bien definidas. Contiene todas las formas, guiones y estándares necesarios para poder registrar y seguir el proceso. Nos enseña los procedimientos para iniciar un proyecto, los pasos para poder guiarlo y nos muestra como analizar y reportar los datos obtenidos durante todo el proyecto.

SM PSP es una marca de servicio de Carnegie Mellon University

TSPi está basado en cuatro principios básicos:

- 1. Aprender es más efectivo cuando se sigue un proceso definido y se tiene una rápida retroalimentación.*
- 2. La productividad del equipo es resultado de metas específicas, un ambiente de trabajo de apoyo y con capacidad de entrenamiento y liderazgo.*
- 3. Cuando se ha luchado con problemas del proyecto y se ha tenido una guía que nos aporte soluciones efectivas, uno aprecia los beneficios de una buena práctica de desarrollo.*
- 4. La instrucción es más efectiva cuando se construye sobre un cuerpo de conocimiento previo.*

1.1.2 Estrategia de TSPi

Existen muchas formas para lograr un diseño de un proceso, en TSP existen siete puntos principales, que en conjunto constituyen su estrategia. A continuación se describen brevemente:

1. Proveer un esquema basándose en PSP

El entrenamiento previo en PSP facilita el aprendizaje de TSPi, muchas formas y guiones son muy similares y la disciplina adquirida es esencial al aplicarse en TSPi.

2. Desarrollar productos en varios ciclos

En un curso completo de TSPi se pueden completar alrededor de dos o tres ciclos de desarrollo de un proyecto. Cada uno incluye requerimientos completos, planeación, diseño, implementación y pruebas. En el primer ciclo el objetivo es tener un producto que sirva de base para los siguientes ciclos. En cada ciclo pueden hacerse cambios de roles, un ajuste en el proceso o tomarse la decisión de aumentar la disciplina en la calidad. Después de completar dos o más ciclos ya se sabe con precisión que es lo que está funcionando mejor dentro del equipo.

3. Establecer medidas estándar para rendimiento y calidad

Sin medidas objetivas de un proyecto no podemos hablar de alta calidad en el trabajo. PSP proporciona las medidas que se necesitan para evaluar la calidad del trabajo y enseña como poder interpretarlas y aplicarlas. TSPi enfatiza las metas y métricas que permitan visualizar los beneficios de las medidas de calidad evaluando la planeación y los datos registrados del proyecto.

4. Proporcionar medidas precisas para el equipo y estudiantes

Con los datos registrados en TSPi, se hace transparente el desempeño personal de cada miembro del equipo. Todos los integrantes del equipo conocen cómo han trabajado los demás y esto ayuda a mejorar el esfuerzo del equipo en general.

5. Utilizar evaluaciones de equipo y rol

La idea de realizar evaluaciones es estar mejor informados acerca del rendimiento de otros y del equipo en su totalidad. Si los miembros del equipo están dispuestos a hacer evaluaciones honestas el instructor o gerente podrá estar mejor informado. TSPi cuenta con evaluaciones para el equipo y los roles desempeñados, la idea es evaluar como fue el desempeño de cada rol y como impactó en el rendimiento del equipo.

6. Requiere de disciplina en los procesos

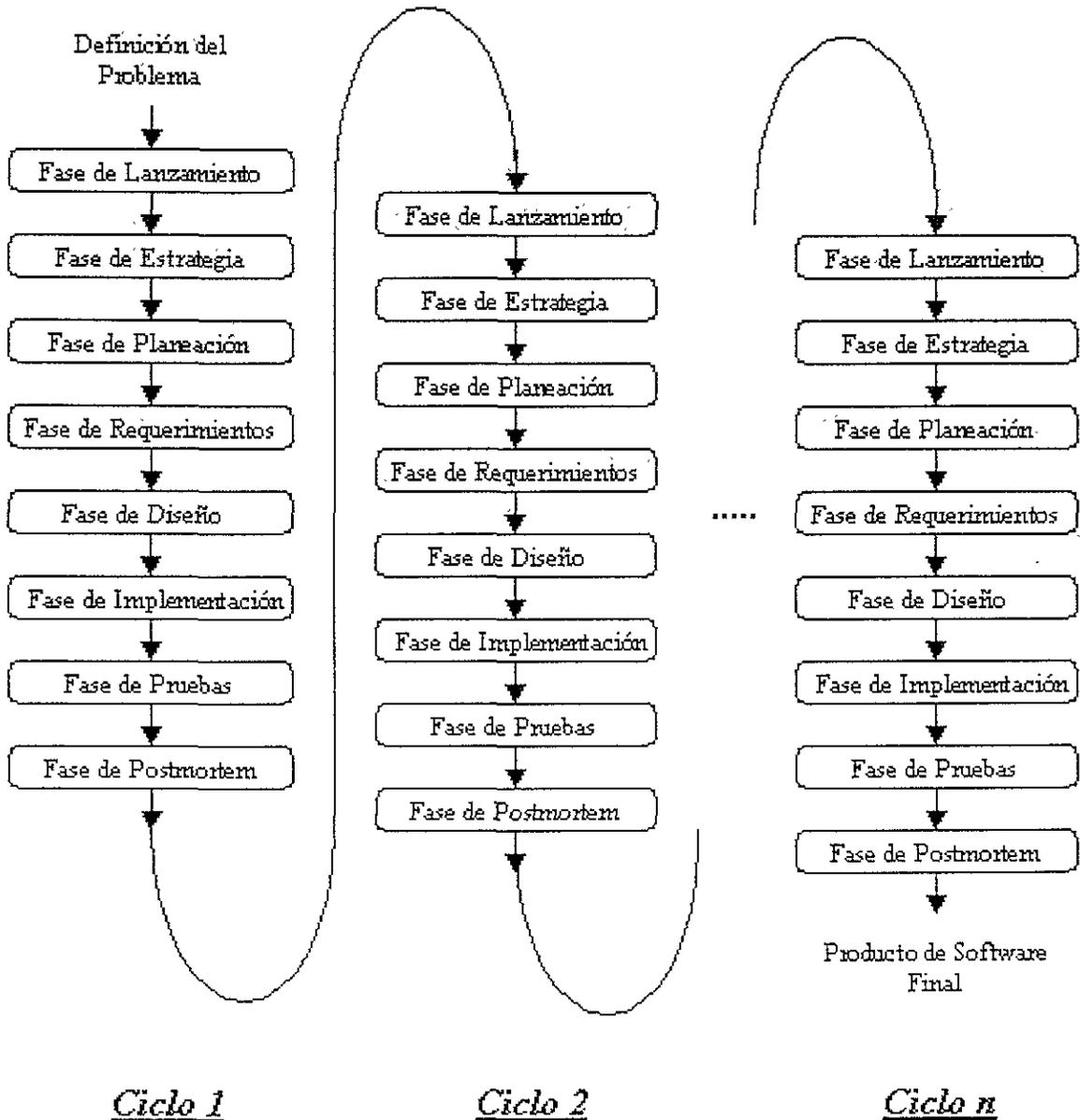
Los ingenieros de software no contamos con modelos de rol que seguir, por lo cual luchamos más para tener un trabajo personal disciplinado. TSPi requiere de alta disciplina en el seguimiento de las tareas dentro del proceso.

7. Provee una guía para resolver los problemas de los equipos de trabajo.

Hasta en el mejor proyecto es común tener problemas al trabajar en equipo, es común que se presenten desacuerdos con los compañeros. Cada miembro del equipo tiene su propia personalidad y esto puede generar problemas en el equipo. Sin embargo con una buena guía y apoyo se puede llegar a ser un miembro efectivo en el equipo.

1.1.3 Estructura y formato de TSPi

La figura siguiente muestra los múltiples ciclos de desarrollo que pueden planearse para obtener un producto final. Cada ciclo contiene las mismas fases y siempre tiene como base al ciclo de desarrollo que lo antecede a excepción del primer ciclo.



1.1.4 Estrategia de desarrollo en ciclos de TPSi

El plan es iniciar con una versión pequeña pero viable del producto. Para decidir el tamaño y contenido de cada ciclo deben tenerse en cuenta los siguientes aspectos:

1. Cada ciclo debe producir una versión que pueda ser probada y que represente un subconjunto del producto final.
2. Cada ciclo debe ser lo suficientemente pequeño como para desarrollar una versión del producto en el tiempo disponible.
3. Los productos de cada ciclo, al combinarse, deben conformar el producto de software final deseado.

1.1.5 Fases del Ciclo TSPi

Las fases en las que se divide el proceso de TSPi se describen brevemente a continuación:

- **Lanzamiento**

Durante esta fase, y siendo el primer ciclo, se realiza una revisión de los objetivos del proyecto. Se da una descripción de la estructura general de TSPi y del contenido. Se forman equipos de estudiantes o ingenieros y a cada uno se le asigna un rol. Finalmente se describen las necesidades del cliente. En los ciclos subsecuentes pueden realizarse intercambios de roles de acuerdo al desempeño o gustos personales.

- **Estrategia**

En esta etapa se crea un diseño conceptual del producto, se establece la estrategia de desarrollo decidiendo que se producirá en cada ciclo. Se realizan estimaciones iniciales acerca del tamaño y el esfuerzo requerido. Se identifican los riesgos.

- **Planeación**

En la planeación se hace un estimado del tamaño de cada parte a ser creada. Se identifican todas las tareas a ser realizadas y se asignan a cada miembro del equipo. Se propone además un plan de calidad que fije parámetros a ser alcanzados.

Con un plan bien detallado se puede trabajar mas eficientemente, pues se conoce con precisión lo que se tiene que hacer y en que momento.

- **Requerimientos**

En esta fase se realiza un análisis de las necesidades del sistema, se efectúan entrevistas con el cliente y se especifican y examinan los requerimientos. Finalmente se desarrolla un plan para realizar las pruebas del sistema.

El proceso de identificación de requerimientos consiste principalmente de hacer preguntas al cliente. Una vez que se ha entendido lo que el usuario requiere se escriben los requerimientos en nuestras propias palabras y se verifica lo escrito junto con el cliente.

La especificación de requerimientos permite entre otras cosas definir el ambiente de sistema operativo, las restricciones, escenarios y procesos. Los requerimientos funcionales se enfocan en entradas, salidas, cálculos matemáticos, casos de uso. Además, se identifican requerimientos de interfaz de usuario, formatos de los archivos, lenguajes, estándares y compatibilidad entre muchos otros.

En esta etapa es necesario realizar un plan de pruebas del sistema que incluya datos de prueba de funcionamiento y un registro de datos esperados y organice la forma y tiempo en el cual será aplicada la prueba.

- **Diseño**

Diseñar es el proceso creativo de decidir cómo construir el producto. Se debe producir una especificación completa y precisa de cómo será nuestro producto. Un diseño completo define las partes principales, describe como interactúan dichas partes y especifica como se unen para producir el producto final.

Esta etapa inicia con la creación de un diseño de alto nivel, donde se especifica y examina cada parte. Posteriormente se realiza un plan de prueba para la integración.

El diseño de alto nivel permite a los ingenieros ver en forma global la interacción de las partes del sistema.

Una forma útil para obtener el diseño es la creación de prototipos de interfaces de usuario, con ello se puede obtener una retroalimentación muy importante en el desarrollo del proyecto.

Un aspecto importante es fijar estándares de diseño que definan: convención de nombres, parámetros, formatos de interfaces y condiciones de error entre otros.

El plan de integración debe tener en cuenta el ensamble e intercomunicación entre clases, se debe especificar la dependencia que cada clase tiene con el resto y organizar la forma de probarlas.

- **Implementación**

En esta fase, cada parte del diseño se traduce a código, se revisa, se compila y prueba.

Los pasos necesarios para el proceso de implementación parten desde la generación del diseño detallado, inspección del diseño detallado, codificación en el lenguaje de programación definido, pruebas unitarias, revisión de la calidad del componente y liberación del mismo.

Es importante antes de iniciar la implementación tener el diseño detallado completo. Dentro del trabajo en equipo se deben administrar y fijar los estándares de nombres, interfaces, mensajes, rutinas reusables, el glosario de nombres. Una parte muy importante que no debe hacerse a un lado es el estándar de codificación. La unificación del código permite que todo el equipo tenga consistencia y ahorra tiempo en la lectura y revisión entre colegas.

• Pruebas

El propósito de esta fase es la verificación del sistema no la corrección del mismo. En forma simultánea se genera la documentación de usuario.

La estrategia para integrar y construir el sistema tiene como propósito asegurarse de que están todas las partes necesarias para ensamblar el sistema y permitir integrarlo y probarlo. Existen varias estrategias que pueden seguirse:

- Big-Bang: Se juntan todas las partes y se prueba todo el sistema en su conjunto.
- Uno a la vez: Se van integrando cada una de las partes de manera incremental y se van probando conforme se ensamblan.
- Agrupamiento: Se identifican los tipos de componentes, clases y se integran de acuerdo a funcionamiento.
- Sistema plano: Se construye integrando todas las partes de alto nivel primero y después se integran las demás partes bajando por capas.

Para administrar los planes de pruebas se deben tener listas de pasos a ejecutar, de materiales requeridos, de resultados producidos y de defectos encontrados. Además hay que tener registrados los objetivos de cada prueba, quién la desarrolla y cuándo se completó la corrección de defectos.

• Postmortem

En esta última fase se realiza un análisis del producto, se escribe un reporte de todo el ciclo, se generan todas las evaluaciones acerca del equipo y finalmente se realiza una presentación del proyecto.

La importancia de esta fase radica en la retroalimentación para cada uno de los miembros del equipo, nos otorga una manera de aprender y mejorar. Aquí es cuando podemos identificar problemas que se presentaron, determinar las causas y proponer medidas para evitarlos. La fase de postmortem es un punto apropiado para identificar mejoras específicas y decidir como incorporar los cambios para alcanzarlas en ciclos posteriores o proyectos a futuro.

1.1.6 Equipo TSPi

TSPi asigna las responsabilidades mediante la identificación de roles.

Los roles de TSPi son:

- Líder del Equipo: Dirige al equipo, se asegura que todos reporten sus datos de los procesos y completen su trabajo tal y como se planeó. Realiza los reportes semanales del avance del equipo.
- Administrador de Desarrollo. Guía al equipo en el diseño y desarrollo del producto.
- Administrador de Planeación. Apoya y guía al equipo en la planeación y seguimiento del trabajo.
- Administrador de Calidad/Proceso. Apoya al equipo en definir sus necesidades acerca del proceso y a establecer y administrar el plan de calidad. Genera estándares para obtener un trabajo uniforme. Modera las inspecciones y revisa cada artefacto generado.
- Administrador de Configuración. Dirige al equipo en el desarrollo de requerimientos de software y ayuda a dar a conocer la tecnología y en las necesidades de apoyo administrativo. Administra el plan de configuración.

1.2 Método aplicado para la construcción del modelo de TSPi

El objetivo de esta sección es mostrar la motivación, el porqué de la elección de diagramas de UML y la forma de trabajo empleada para realizar la guía de TSPi.

La motivación principal para realizar el presente trabajo surge a partir de la materia “Ingeniería de Software Orientada a Objetos” impartida por la Dra. Hanna Oktaba y la M. en C. Guadalupe Ibarguengoitia y que cursé en el semestre 2000-I, como parte del plan de la Maestría en Ciencia e Ingeniería de la Computación. En este curso se desarrolló un sistema de tutores y temas de investigación para la coordinación de la maestría aplicando TSPi. A cada uno de los alumnos nos correspondió desempeñar un rol de los descritos en TSPi. El rol que desempeñé fue administrador de calidad. Desde la visión que esta responsabilidad me daba y a partir de los resultados obtenidos en los dos ciclos en los cuales se desarrolló el proyecto, noté la eficacia, orden y calidad que el proceso ofrece. Para poder aplicar TSPi la única referencia con la que contábamos era el libro de texto de Watts S. Humphrey el cual plantea guiones, formas y párrafos explicativos, sin embargo, el tiempo disponible para realizar las lecturas era muy corto y las horas invertidas en interpretar cada fase nos consumían gran cantidad de calendario del proyecto.

Por esas fechas y a propuesta de la Dra. Hanna Oktaba surge la posibilidad de ver un trabajo de tesis “Modelado de áreas clave de procesos para CMM nivel 2” [Alquicira, 2000] donde a partir de una extensión de los diagramas de actividades de UML se realizaba una explicación sencilla de un nivel de CMM que fue muy bien aceptada por desarrolladores de software.

A partir de la idea mostrada en dicha tesis y la oportunidad de difundir TSPi como un proceso para el desarrollo de software con calidad, surge la idea de realizar una guía que enseñe como seguir TSPi.

El uso de diagramas siempre ha demostrado ser útil en el aprendizaje de cualquier área. En este caso, se eligió representar el proceso de TSPi con diagramas de UML, pues es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software, que en la actualidad ha aumentado su popularidad al ser adoptado por muchos desarrolladores de software.

La guía está desarrollada mediante diagramas de actividades (ver anexo para mas información), cada diagrama de actividad representa una de las fases de TSPi. La presentación se encuentra en forma de hipertexto donde cada actividad dentro del diagrama cuenta con su propia descripción. Adicionalmente, cada artefacto de entrada y salida se enriqueció con ejemplos extraídos del proyecto realizado en el curso. Es necesario aclarar que los ejemplos no son incluidos en la tesis pues son demasiado extensos. Sin embargo pueden ser consultados en la página de Internet <http://kasia.fcencias.unam.mx/TSPi>.

El metodo que se siguió para desarrollar la guía consta de las siguientes partes:

1. Construcción de la guía

- Por cada fase de TSPi correspondiente al primer ciclo o ciclos posteriores se realizó la abstracción del contenido de TSPi y se diseñó un diagrama de actividades.
- Para cada diagrama se identificaron cada uno de los roles que participan dentro de dicha fase y las actividades de las cuales son responsables o comparten responsabilidades. Una labor que se desarrolló en este punto, fue localizar las actividades que pueden realizarse al mismo tiempo.
- En cada actividad que así lo exigiera, se representaron los artefactos de entrada y/o salida ligados a ella.
- Algunas actividades además, tienen identificados guiones y herramientas necesarias para poder llevar a cabo dicha actividad.
- Cada actividad cuenta con una breve descripción indicando su objetivo, los pasos que deben seguirse para realizarla, lista de los artefactos relacionados a ella y precisa los roles responsables dentro de la actividad.
- Para cada artefacto se seleccionó un ejemplo concreto que fue retomado del proyecto realizado para la Coordinación de la Maestría y puede usarse como referencia pero no como una forma estricta de realizar los productos.

2. Verificación de la Guía

- Cada uno de los diagramas, descripciones y ejemplos asociados fueron verificados de una manera periódica durante las reuniones semanales con la directora de tesis, con el fin de asegurar la correspondencia directa con el texto original de TSPi.

3. Publicación de la guía

- Finalmente todo el material fue entregado a Sergio Ramos Chávez, estudiante de la licenciatura en Ciencias de la Computación, quien se ha encargado de diseñar el sitio de Internet, realizar los vínculos y poner a disponibilidad los ejemplos. La versión en español para las formas y guiones de cada una de las fases del proceso de TSPi fueron facilitados por Carmen Alvarez Sánchez estudiante de la Maestría en Ciencia de la Computación.

Las actividades de construcción, verificación y publicación de la guía fueron realizadas de manera incremental e iterativa durante el lapso de seis meses.

La organización de la guía fue generada a partir de un nivel de abstracción mas alto del proceso general de TSPi. El siguiente nivel corresponde a ciclos y fases que se desglosan a su vez en diagramas de actividades atómicas. Se inició describiendo en un primer diagrama el proceso de desarrollo incremental en ciclos de TSPi. A partir de esta descripción, se logró definir las fases correspondientes al primer ciclo de desarrollo, que es el más complicado por el desconocimiento de las labores a realizar, y de los ciclos de desarrollo siguientes que están cimentados en el ciclo de desarrollo que lo antecede.

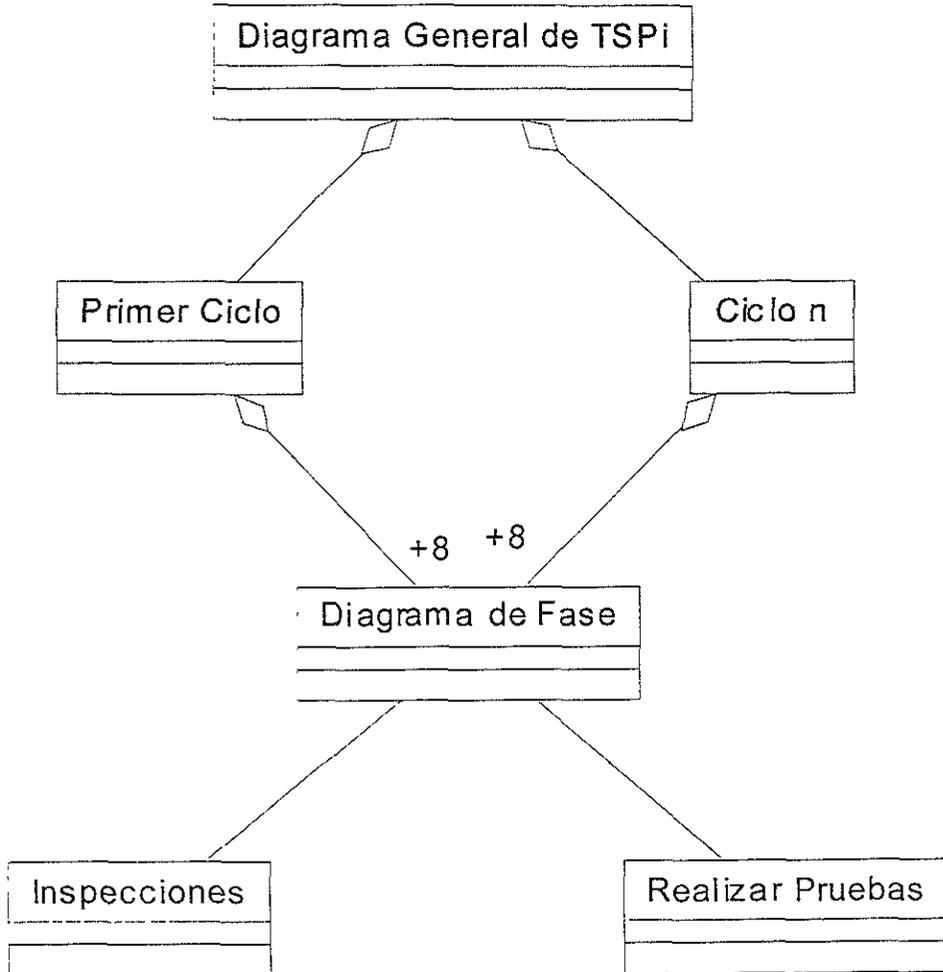
Capítulo 2

Guía de TSPi usando UML

El objetivo de este capítulo es presentar de manera formal la guía de TSPi. La guía describe en primer lugar los objetos que componen a cada uno de los diagramas. Los presenta uno a uno y muestra los iconos que han sido empleados para diferenciarlos unos de otros. En el sitio de Internet correspondiente al material mostrado se hace uso de colores para realzar dicha diferenciación. Si se requiere de una información más detallada de los diagramas de actividades puede dirigirse al anexo A dentro de este trabajo. Si lo que requiere es la descripción de los elementos véase la tabla situada en el anexo B.

Finalmente se muestran los diagramas de actividades de cada una de las fases y las tablas que describen las actividades.

La presentación de éste trabajo fue ideada como una jerarquía de diagramas ligados en forma de hipertexto.



La parte superior de esta estructura es un diagrama al nivel más alto de abstracción que muestra el proceso general de TSPI. Contiene las actividades anidadas correspondientes al primer ciclo de desarrollo y a los n ciclos de desarrollo subsecuentes.

Como siguiente nivel se presentan el conjunto de diagramas y descripciones de actividades correspondientes a las fases del primer ciclo y ciclos siguientes.

La estructura general de la guía es la siguiente:

- Diagrama general de TSPi
 - Diagrama general del primer ciclo
 - Diagrama y tabla de actividades de Lanzamiento
 - Diagrama y tabla de actividades de Estrategia
 - Diagrama y tabla de actividades de Planeación
 - Diagrama y tabla de actividades de Requerimientos
 - Diagrama y tabla de actividades de Diseño
 - Diagrama y tabla de actividades de Implementación
 - Diagrama y tabla de actividades de Pruebas
 - Diagrama y tabla de actividades de PostMortem
 - Diagrama general del ciclo n
 - Diagrama y tabla de actividades de Lanzamiento
 - Diagrama y tabla de actividades de Estrategia
 - Diagrama y tabla de actividades de Planeación
 - Diagrama y tabla de actividades de Requerimientos
 - Diagrama y tabla de actividades de Diseño
 - Diagrama y tabla de actividades de Implementación
 - Diagrama y tabla de actividades de Pruebas
 - Diagrama y tabla de actividades de PostMortem

Al final de la guía se presentan los subdiagramas y tablas de actividades correspondientes a Inspección y Realización de Pruebas que se encuentran a un nivel de abstracción más bajo.

2.1 Descripción de los diagramas

Los principales elementos de los diagramas de actividades que se muestran en la presente guía tienen las siguientes características:

- **Encabezado**

Muestra un número que identifica al diagrama de actividades y un título descriptivo que hace referencia a alguna fase de TSPi.

0	TSPi
---	------

- **Actividad**

Contiene la descripción general de la actividad. Todas las actividades van ligadas a un flujo de trabajo definido por el sentido de las flechas y las barras de sincronización o división (ver anexo A). Las actividades se representan por un rectángulo de esquinas redondeadas.

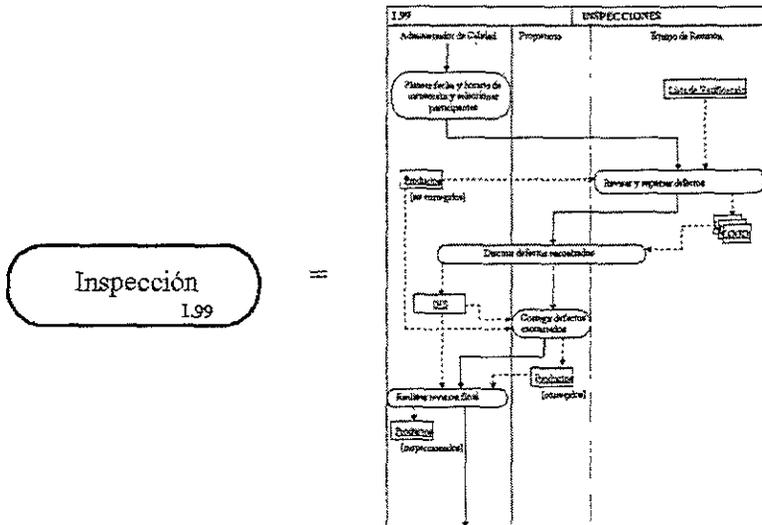


Todas las actividades correspondientes a cada fase están descritas en una tabla que se basa en los guiones de TSPi y donde se muestra: la descripción de la actividad, el propósito, los pasos necesarios para realizarla, se listan los artefactos de entrada y salida si es el caso, finalmente contiene una lista de los roles responsables de realizar dicha actividad.

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definir Diseño de Alto Nivel	Definir las funciones del producto para el primer ciclo	Decidir la estructura del producto Nombrar cada uno de los componentes Producir las especificaciones para cada componente Identificar todas las tareas de diseño que deben ser realizadas	Productos de Requerimientos		Administrador de Desarrollo
Dividir Trabajo	Balancear la carga de trabajo para producir los artefactos de diseño	Identificar todas las tareas necesarias para producir el documento de diseño (SDS) Dividir el trabajo en partes y asignarlas a cada miembro del equipo	Productos de Requerimientos		Administrador de Desarrollo

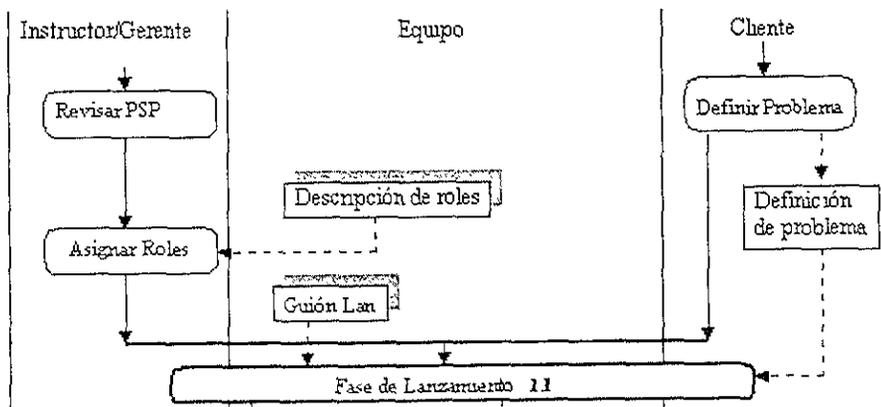
• **Actividades Anidadas**

Este tipo de actividades contienen por sí mismas otro subdiagrama de actividad que debe ser realizado en su totalidad para poder continuar el flujo de trabajo. A diferencia de otras actividades, se simbolizan dibujándose con una línea gruesa, además contienen una numeración que indica el numero de diagrama al que se debe hacer referencia.



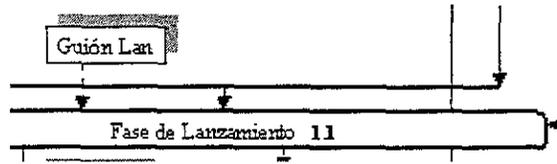
• **Carriles de responsabilidad**

Una actividad tiene algún responsable encargado de realizar su ejecución. En tal caso la actividad queda dentro de los límites del carril del rol responsable. En ocasiones las responsabilidades pueden ser compartidas, en dicho caso la actividad puede rebasar los límites de los carriles y abarcar el espacio correspondiente a los roles responsables. El nombre del responsable se localiza en la parte superior del carril que tiene asignado.



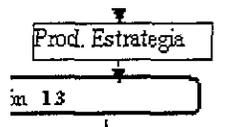
- **Artefacto de Documentos de Apoyo**

Es un documento que es utilizado para definir o guiar una actividad. Haciendo uso de estereotipos e íconos, se representa como un rectángulo sombreado.



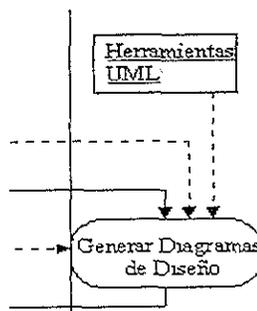
- **Artefacto de Producto de Trabajo**

Es un documento que es necesario como entrada para realizar una actividad o bien son los productos de salida al terminar la ejecución de una actividad. Se representa como un rectángulo sin relleno.



- **Artefacto de herramienta**

Es un objeto que muestra alguna herramienta que puede ser utilizada como apoyo en la realización de una actividad. Se representa como un rectángulo con relleno gris.



La descripción mas detallada de los elementos anteriormente presentados y de algunos otros correspondientes a los diagramas de actividades, se encuentran en el anexo A al final del presente trabajo. En el anexo B se muestra una tabla con los principales elementos de los diagramas.

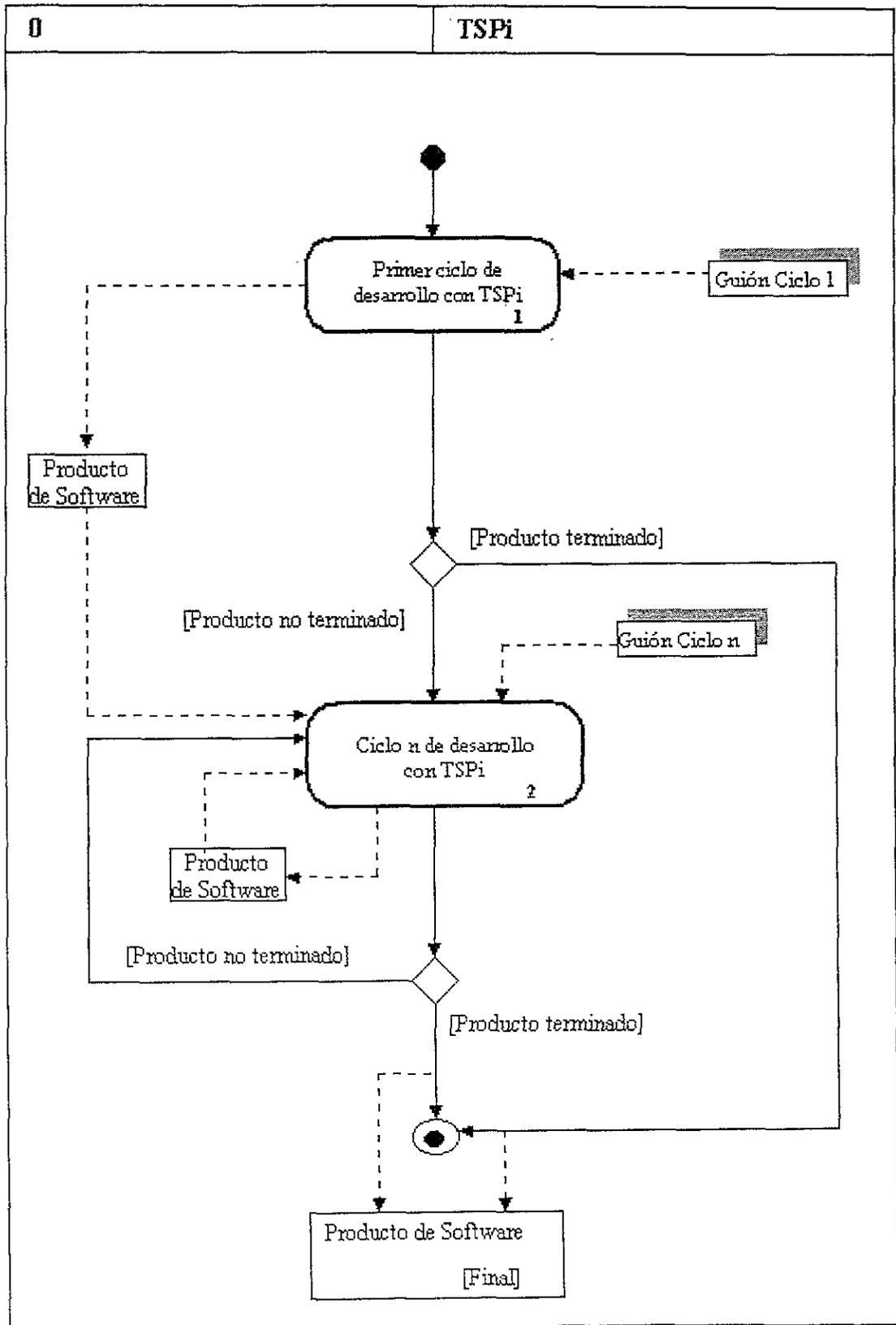
2.2 Diagrama General de TSPi

En esta sección se presenta el diagrama general del proceso de TSPi.

En la figura mostrada en la siguiente página, se encuentra el diagrama de actividad 0 que corresponde al proceso general de TSPi.

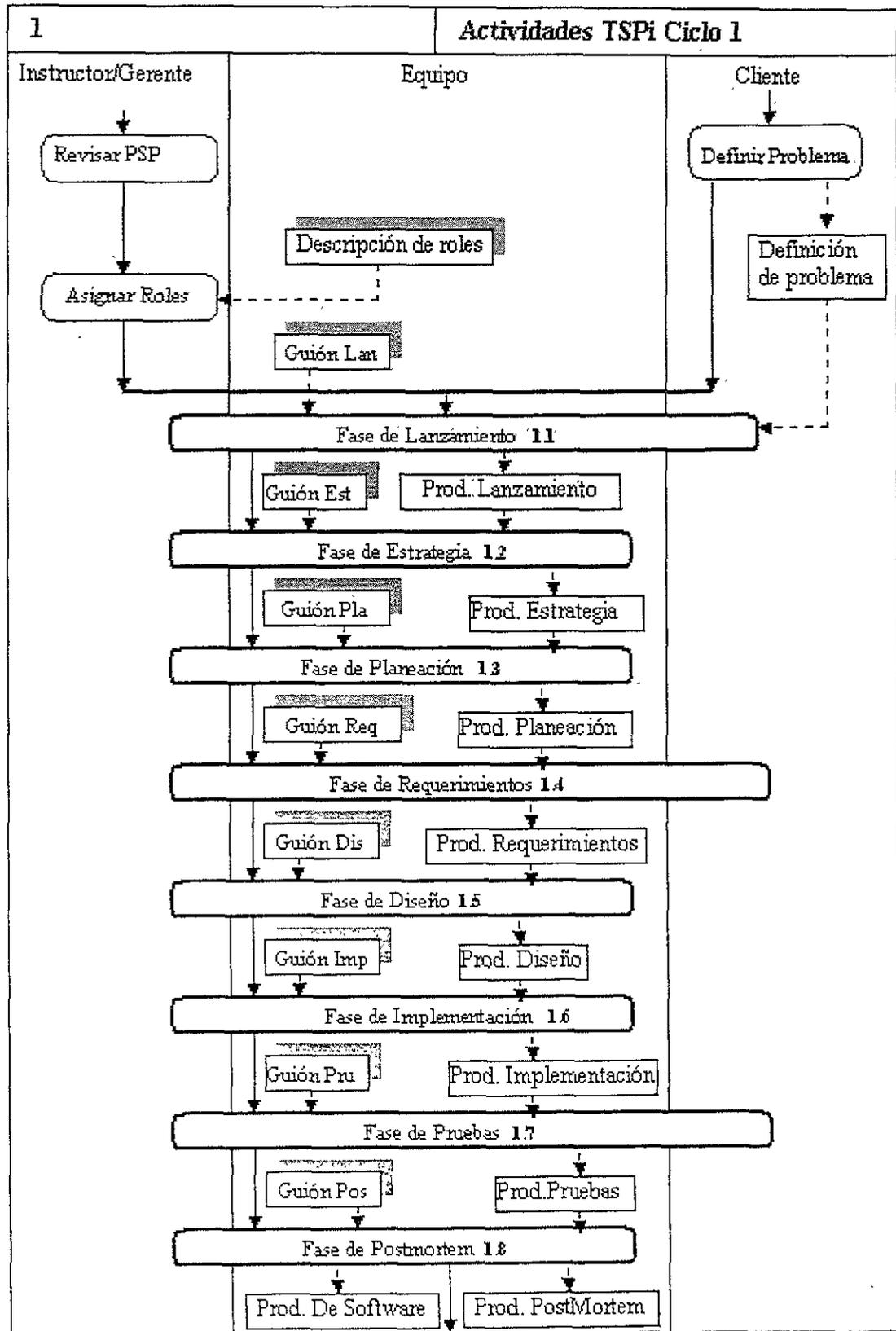
El diagrama comienza a partir de un estado inicial indicado por un círculo relleno. La primera actividad por realizar es *Primer ciclo de desarrollo de TSPi*. Esta actividad es anidada e indica que se desglosa en el diagrama de actividad 1. Para realizar esta actividad se cuenta con un documento de apoyo denominado *Guión ciclo 1*. Al finalizar la actividad se genera un producto de trabajo de salida llamado *Producto de Software*. Siguiendo el sentido de la transición se encuentra una bifurcación en la cual hay dos caminos alternativos por seguir, que dependen de la condición *Producto terminado* o *Producto no terminado*. En el caso que el producto haya sido concluido se llegará al estado final y el producto de software pasará del estado producto terminado al estado final. En caso contrario cuando el producto no ha sido concluido satisfactoriamente, el flujo de la transición indica realizar la actividad *Ciclo n de desarrollo con TSPi*. Esta actividad es anidada y se define en el diagrama de actividades 2. Como se observa en la figura, tiene un producto de entrada denominado *Producto de Software* que proviene del ciclo de desarrollo de TSPi que lo antecede. También tiene un documento de apoyo denominado *Guión Ciclo n* el cual guiara en la realización de la actividad. Una vez que se ha concluido la actividad y siguiendo la transición, tenemos nuevamente una bifurcación que al igual que la anterior depende del resultado de tener un Producto terminado o no. En el primer caso se pasa al estado final generando un *Producto de Software* con estado final. En caso de no contar con un producto terminado regresaríamos a la actividad *Ciclo n de desarrollo con TSPi*. Este ciclo terminará en cuanto se cuente con un Producto de Software terminado.

Cada diagrama de actividades esta basado en un guión tomado de TSPi. En este trabajo no se presentan dichos guiones. Sin embargo, están disponibles en su versión en español en la página de Internet correspondiente a la presente guía <http://kasia.fciencias.unam.mx/TSPi>

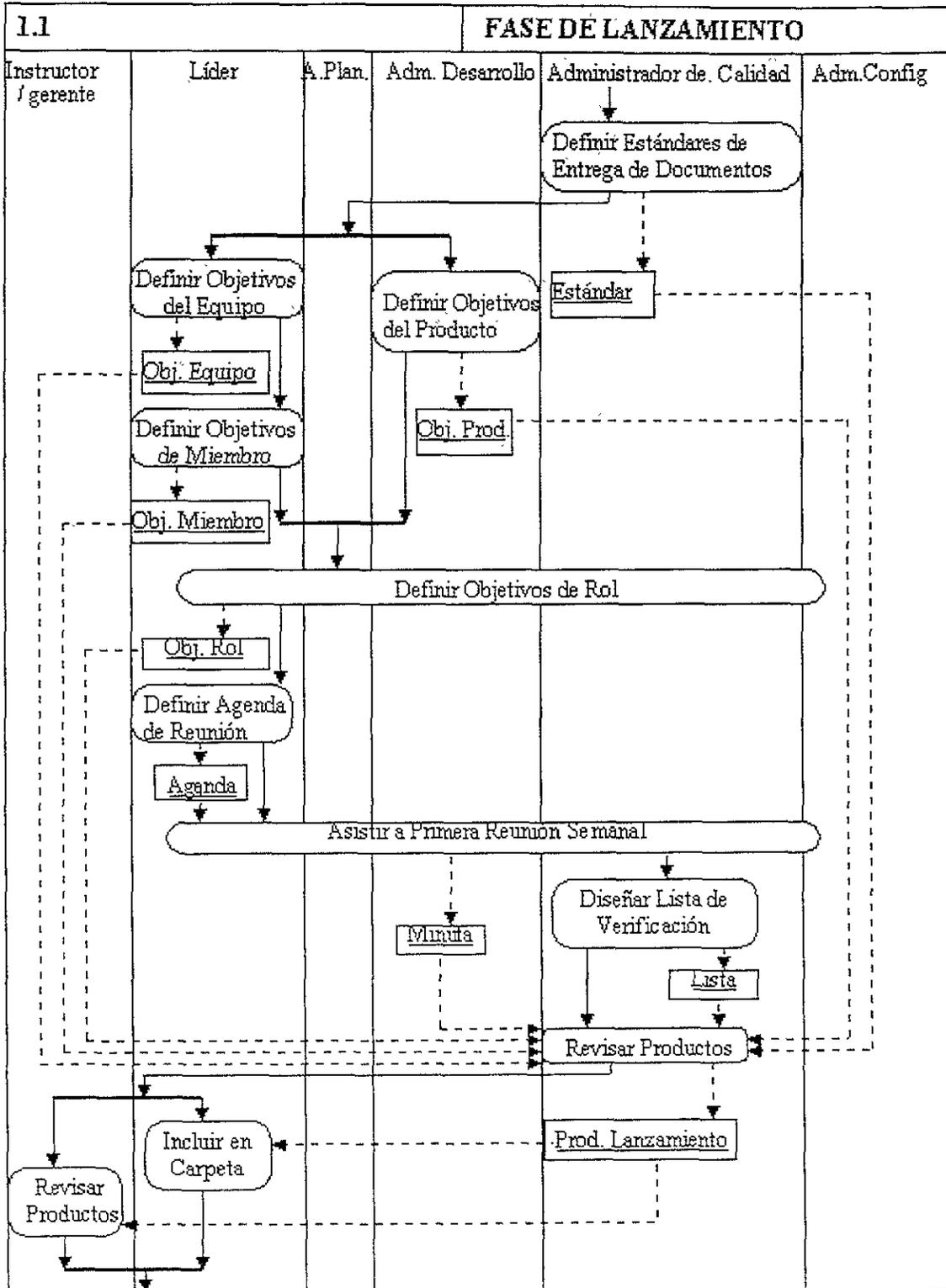


2.3 Primer Ciclo de TSPi

En esta sección se presentan los diagramas y tablas de actividades correspondientes al primer ciclo de desarrollo de TSPi



2.3.1 Lanzamiento del Primer Ciclo de TSPi



Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definir formatos de entrega de documentos	Definir formatos de entrega de documentación	<ul style="list-style-type: none"> - Diseñar y sugerir formas de entrega de documentación - Acordar con el equipo de trabajo los formatos - Entregar copia a cada miembro del equipo 		Estándar de Documentación	Administrador de Calidad
Definir objetivos del equipo	Definir metas reales y alcanzables como equipo Fijar criterios cuantitativos para comparar los resultados alcanzados	<ul style="list-style-type: none"> - Definir los tres objetivos fundamentales propuestos en TSPi y sus métricas <ul style="list-style-type: none"> • Producir un producto de calidad • Poner en funcionamiento un proyecto bien administrado • Finalizar a tiempo - Definir objetivos complementarios que sean importantes en la construcción del producto - Acordar con el equipo todos los objetivos y métricas propuestos - Entregar copia de objetivos a cada miembro del equipo 	Estándar de Documentación	Objetivos del equipo	Líder del equipo
Definir objetivos de miembro del equipo	Definir metas reales y alcanzables como integrante del equipo Fijar métricas cuantitativas para comparar los resultados obtenidos con los objetivos planeados	<ul style="list-style-type: none"> - Definir los cuatro objetivos fundamentales propuestos en TSPi y sus métricas <ul style="list-style-type: none"> • Ser un miembro de equipo cooperativo y efectivo • Realizar un trabajo personal disciplinado • Planear y seguir todo el trabajo personal • Producir un producto de calidad como ingeniero de desarrollo - Definir objetivos 	Estándar de Documentación	Objetivos de miembro del equipo	Líder del equipo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
		complementarios que sean importantes en la construcción del producto - Acordar con el equipo todos los objetivos y métricas propuestos - Entregar copia de objetivos a cada miembro del equipo			
Definir objetivos de rol	Definir metas reales y alcanzables en el rol asignado Fijar métricas cuantitativas para comparar los resultados obtenidos con los objetivos planeados	- Definir los cuatro objetivos fundamentales propuestos en TSPi por rol - Definir objetivos complementarios que sean importantes en la construcción del producto	Estándar de Documentación	Objetivos de rol	Líder del equipo
Definir objetivos del producto	Descripción de objetivos críticos del producto a ser satisfechos, objetivos opcionales y deseables y el criterio para evaluar un producto terminado	- Definir objetivos básicos del producto	Estándar de Documentación	Objetivos del Producto	Administrador de Desarrollo
Definir agenda de reunión semanal	Organizar los temas a tratar durante la reunión semanal	- Definir temas a tratar durante la reunión asignando tiempos para cada uno - Incluir si son necesarios, los temas sugeridos por los miembros del equipo - Distribuir tópicos a cada miembro del equipo	Estándar de Documentación	Agenda de Reunión	Líder del equipo

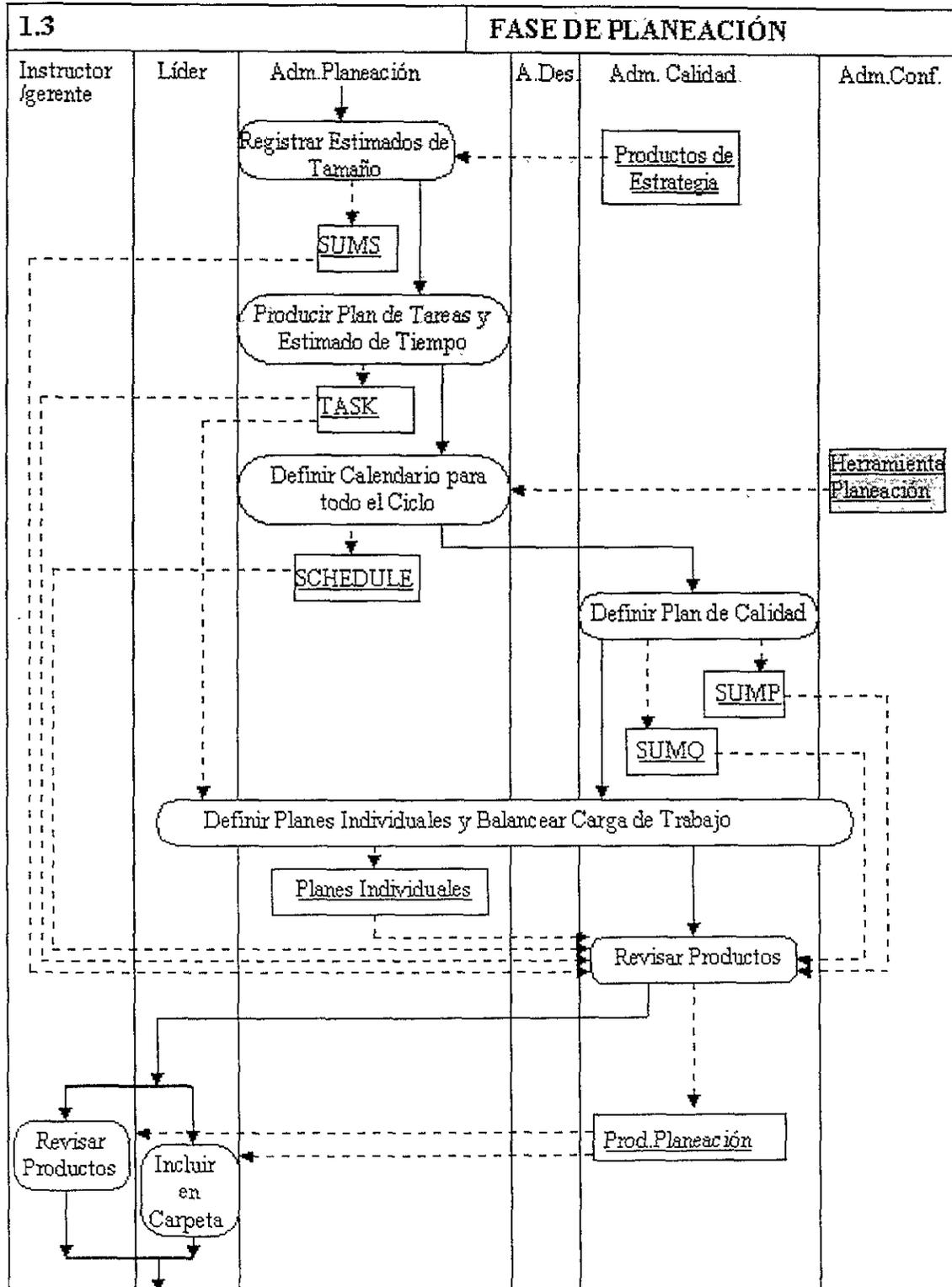
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Reunión a la hora de la reunión	Reunir a los miembros del equipo para discutir y acordar objetivos Actualizar los avances logrados e identificar riesgos y problemas	<ul style="list-style-type: none"> - Seguir la agenda de reunión - Registrar asuntos tratados y acuerdos (Administrador de Calidad) 	Agenda de reunión	Minuta	Todo el equipo de trabajo
Crear Lista de Verificación	Contar con un listado de posibles defectos a localizar en el producto	<ul style="list-style-type: none"> - Realizar una lista con parámetros extraídos del estándar de entrega de documentos - Incluir en la lista los parámetros de ortografía y redacción 	Estándar de documentación	Lista de Verificación	Administrador de Calidad
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de verificación Estándar de documentación Objetivos del equipo Objetivos de miembro del equipo Objetivos de rol Objetivos de producto Minuta	Productos de Lanzamiento: Estándar de documentación Objetivos del equipo Objetivos de Miembro del Equipo Objetivos de Rol Objetivos de Producto Minuta	Administrador de Calidad / Gerente o Instructor
Revisar en el producto	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	Productos de Lanzamiento		Líder del Equipo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Elegir Criterios de Estrategia	Elegir una estrategia de desarrollo para la construcción del producto	<ul style="list-style-type: none"> - Definir el desarrollo del producto en ciclos, el producto final de cada ciclo debe ser de alta calidad, fácil de probar y de mejorar - Identificar componentes principales del producto 	Estándar de Documentación Productos de Lanzamiento	Criterios de Estrategia	Administrador de Desarrollo
Incluir Diseño Conceptual	Identificar las funciones que cada componente del producto debe incluir	<ul style="list-style-type: none"> - Analizar cómo construir el producto a partir de lo que se conoce - Identificar cuáles pueden ser los componentes básicos que formarían al producto - Analizar que funciones en general debe incluir cada componente del producto - Definir el producto en forma modular 	Estándar de Documentación Criterios de Estrategia	Forma STRAT con Diseño Conceptual	Administrador de Desarrollo
Identificar riesgos	Identificar y evaluar los principales riesgos durante todo el proyecto	<ul style="list-style-type: none"> - Cada miembro del equipo analiza la forma STRAT con Diseño Conceptual y sugiere posibles riesgos - Ordenar riesgos de acuerdo a su grado de probabilidad e impacto en el calendario del proyecto - Discutir y buscar formas de evitar cada uno de los riesgos en la reunión semanal - Llenan los resultados en la forma ITL 	Estándar de Documentación Forma STRAT con Diseño Conceptual	Forma ITL	Administrador de Planeación Participantes: Líder del Equipo Administrador de Planeación Administrador de Desarrollo Administrador de Calidad Administrador de Configuración Planeación

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definir estrategia y funcionalidad en cada Ciclo	Definir las funciones a realizar en cada ciclo	<ul style="list-style-type: none"> - Seleccionar del STRAT con Diseño Conceptual un conjunto de funciones para cada ciclo de desarrollo 	Estándar de Documentación Forma STRAT con Diseño Conceptual	Forma STRAT con Diseño Conceptual y Funcionalidad por Ciclos	Administrador de Desarrollo
Reducir tamaños de producto y tiempo	Estimar el tamaño del producto y el calendario de duración necesario para su desarrollo	<ul style="list-style-type: none"> - Realizar un calculo estimado del tamaño (p. Ej. Líneas de Código) para cada función del diseño conceptual, considerar cuantas unidades de tamaño podrían agregarse o modificarse - Realizar un calculo aproximado del número de horas necesarias para cada función de acuerdo al tamaño estimado - Registrar los datos obtenidos en la forma STRAT 	Estándar de Documentación Criterios de Estrategia Forma STRAT con Diseño Conceptual y Funcionalidad por Ciclos	Forma STRAT con Diseño Conceptual, Funcionalidad por Ciclos y Estimaciones de Tamaño y Tiempo	Administrador de Desarrollo
Reducir Plan de Configuración	Realizar un plan que detalle el manejo de versiones y los cambios realizados en los productos generados	<ul style="list-style-type: none"> - Definir a los integrantes de la mesa de control de cambios (MCC) encargados de aceptar o rechazar los mismos - Definir los documentos que serán resguardados - Describir el procedimiento para la solicitud de cambios 	Estándar de Documentación	Plan de Configuración	Administrador de Configuración

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de documentación Criterios de Estrategia Forma STRAT Forma ITL Plan de Configuración	Productos de Estrategia Criterios de Estrategia Forma STRAT Forma ITL Plan de Configuración	Administrador de Calidad / Gerente o Instructor
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Estrategia		Líder del Equipo

2.3.3 Planeación del Primer Ciclo de TSPi

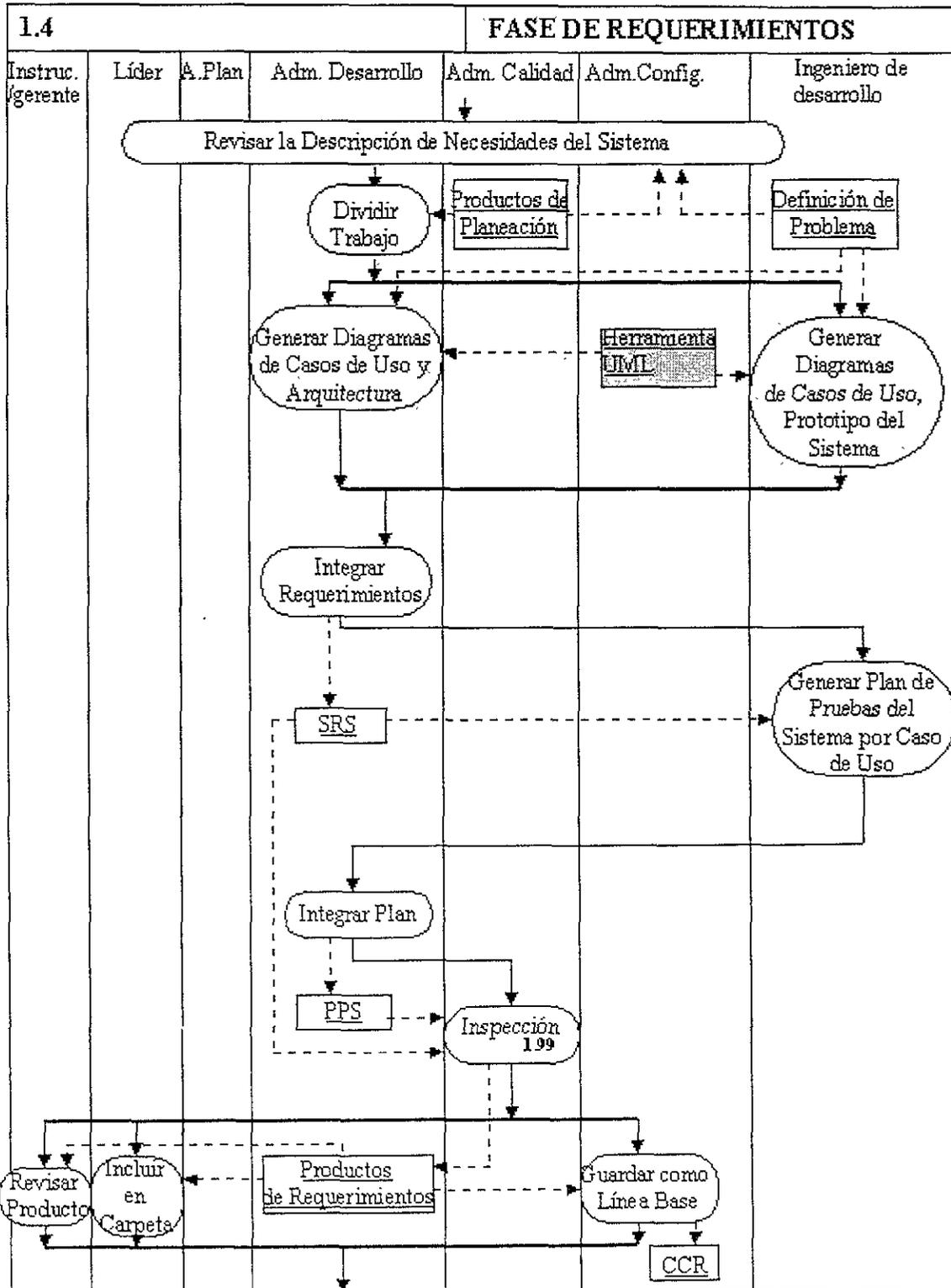


Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Estimar tamaños de las partes del sistema y otros productos a desarrollar	Estimar tamaños de las partes del sistema y otros productos a desarrollar	<ul style="list-style-type: none"> - Identificar los productos que serán producidos <ul style="list-style-type: none"> • Documentos de Requerimientos • Paginas de Diseño de Alto Nivel • Documentación de Usuario • Materiales de Prueba • Otros - Extraer los estimados registrados en la forma STRAT - Registrar todos los datos en la Forma SUMS 	Estándar de Documentación Productos de Estrategia	Forma SUMS	Administrador de Planeación
Definir Plan de áreas y tiempo	Estimar el tiempo para completar cada tarea planeada	<ul style="list-style-type: none"> - Registrar la fase de TSPi en la cual se realizará cada tarea - Indicar el número de ingenieros involucrados por cada tarea - Estimar el tiempo que dedicará cada rol en la tarea correspondiente - Registrar el tiempo total para cada tarea - Guardar todos los datos anteriores en la Forma TASK 	Estándar de Documentación Productos de Estrategia Forma SUMS	Forma TASK	Administrador de Desarrollo
Definir calendario a todo el proyecto	Estimar el número de horas empleadas por semana en el desarrollo del producto	<ul style="list-style-type: none"> - Registrar numero de semana, fechas y totales de tiempo estimados por semana - Guardar los datos en la Forma SCHEDULE 	Estándar de Documentación Forma TASK	Forma SCHEDULE	Administrador de Planeación

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definir Plan de Calidad	Establecer parámetros de calidad y métricas para comparar resultados obtenidos	<ul style="list-style-type: none"> - Calcular valores planeados de tiempo y tamaño de acuerdo a los datos de la forma TASK - Realizar estimado de defectos inyectados y removidos para cada producto y tarea - Registrar los datos en la Forma SUMP - Estimar tazas de defectos y rendimiento para cada fase - Registrar los datos en la Forma SUMQ 	Estándar de Documentación Forma TASK	Forma SUMP Forma SUMQ	Administrador de Calidad
Definir Planes Individuales y balancear la carga de trabajo	Acordar los tiempos y tareas para cada uno de los miembros del equipo de acuerdo a lo planeado	<ul style="list-style-type: none"> - Analizar las horas estimadas para cada tarea y la disponibilidad de horas de cada integrante - Asignar las tareas para cada ingeniero por semana - Entregar copia de las Formas TASK y SCHEDULE a cada miembro del equipo - Analizar las tareas asignadas y balancear si es necesario el trabajo asignado 	Estándar de Documentación Forma TASK Forma SCHEDULE	Planes Individuales	Administrador de Planeación Participantes: Líder del Equipo Administrador de Planeación Administrador de Desarrollo Administrador de Calidad Administrador de Configuración

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de documentación Forma SUMS Forma TASK Forma SCHEDULE Forma SUMP Forma SUMQ Planes Individuales	Productos de Planeación Forma SUMS Forma TASK Forma SCHEDULE Forma SUMP Forma SUMQ Planes Individuales	Administrador de Calidad / Gerente o Instructor
Incluir en el mapa	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	Productos de Planeación		Líder del Equipo

2.3.4 Requerimientos del Primer Ciclo de TSPi

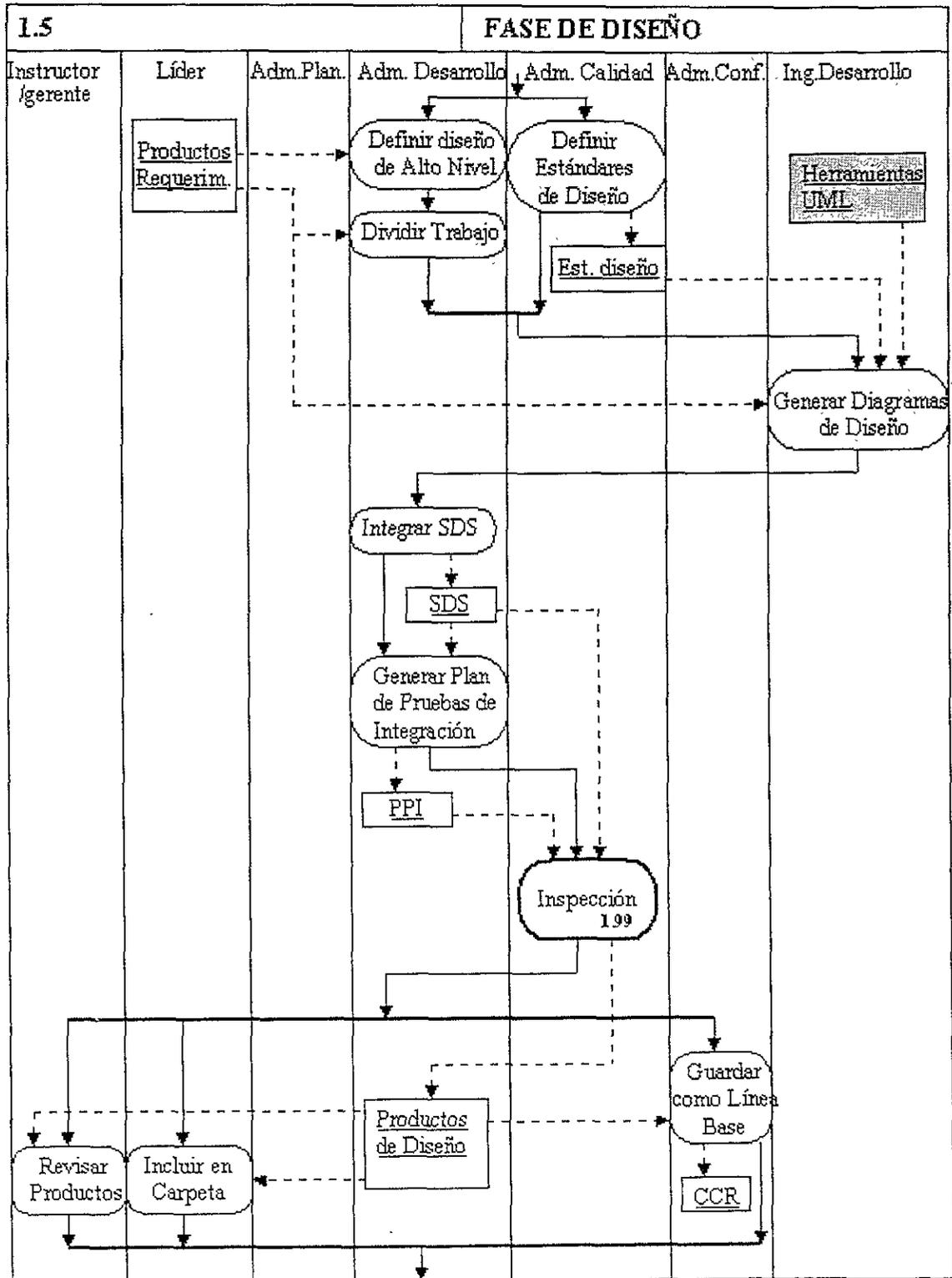


Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Analizar la descripción de las necesidades del Sistema	Clarificar y unificar el entendimiento acerca de las necesidades del sistema entre todos los miembros del equipo y el cliente	<ul style="list-style-type: none"> - Cada ingeniero revisa las necesidades del sistema - Entrevistarse con el cliente para aclarar puntos que podrían ser ambiguos - Identificar claramente las funciones que serán realizadas en el ciclo uno 	Definición de Problema Productos de Planeación		Administrador de Desarrollo Participantes: Líder del Equipo Administrador de Planeación Administrador de Desarrollo Administrador de Calidad Administrador de Configuración Cliente
Dividir el trabajo	Balancear la carga de trabajo para el documento de requerimientos	<ul style="list-style-type: none"> - Identificar todas las tareas necesarias para producir el documento de requerimientos (SRS) - Dividir el trabajo en partes y asignarlas a cada miembro del equipo 	Productos de Planeación		Administrador de Desarrollo
Definir programas Casos de uso y arquitectura	Definir recursos necesarios y escenarios del proyecto en un nivel de abstracción alto	<ul style="list-style-type: none"> - Redactar el propósito del documento de requerimientos (SRS) - Listar todas las necesidades del sistema - Identificar los casos de uso con funcionalidades generales del sistema - Identificar arquitectura general del sistema - Realizar un prototipo que muestre la interfaz de usuario - Indicar las restricciones posibles en el desarrollo 	Estándar de Documentación Definición de Problema		Administrador de Desarrollo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Generar programas y casos de prueba de prototipo del sistema	Generar escenarios e interfaz de usuario por caso de uso	<ul style="list-style-type: none"> - Identificar escenarios con actores - Analizar los flujos normales y extraordinarios por caso de uso 		Estándar de Documentación Definición de Problema	Ingeniero de Desarrollo
Generar requerimientos	Generar el Documento de Requerimientos (SRS)	<ul style="list-style-type: none"> - Integrar el Documento de Requerimientos (SRS) <ul style="list-style-type: none"> • Casos de uso generales • Arquitectura del sistema • Casos de uso por funcionalidad • Prototipo del sistema 	Estándar de Documentación Definición de Problema	SRS	Administrador de Desarrollo
Realizar Plan de Pruebas Sistema Caso de uso	Realizar un plan de prueba por caso de uso	- Documentar la forma en que se probará cada caso de uso del sistema	Estándar de Documentación SRS		Ingeniero de Desarrollo
Generar Plan de Pruebas del Sistema	Producir el Plan de Pruebas del Sistema	- Integrar cada uno de los planes de prueba por funcionalidad producidos por los ingenieros de desarrollo	Estándar de Documentación SRS	PPS (Plan de Pruebas del Sistema)	Administrador de Desarrollo
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de documentación SRS PPS	Productos de Requerimientos SRS PPS CCR	Administrador de Calidad / Gerente o Instructor

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Requerimientos		Líder del Equipo
Guardar en Línea	Resguardar los documentos de requerimientos	- Asignar una versión al documento de requerimientos - Guardarlo en Línea Base - Registrar datos en la forma CCR	Productos de Requerimientos	Forma CCR	Administrador de Configuración

2.3.5 Diseño del Primer Ciclo de TSPi

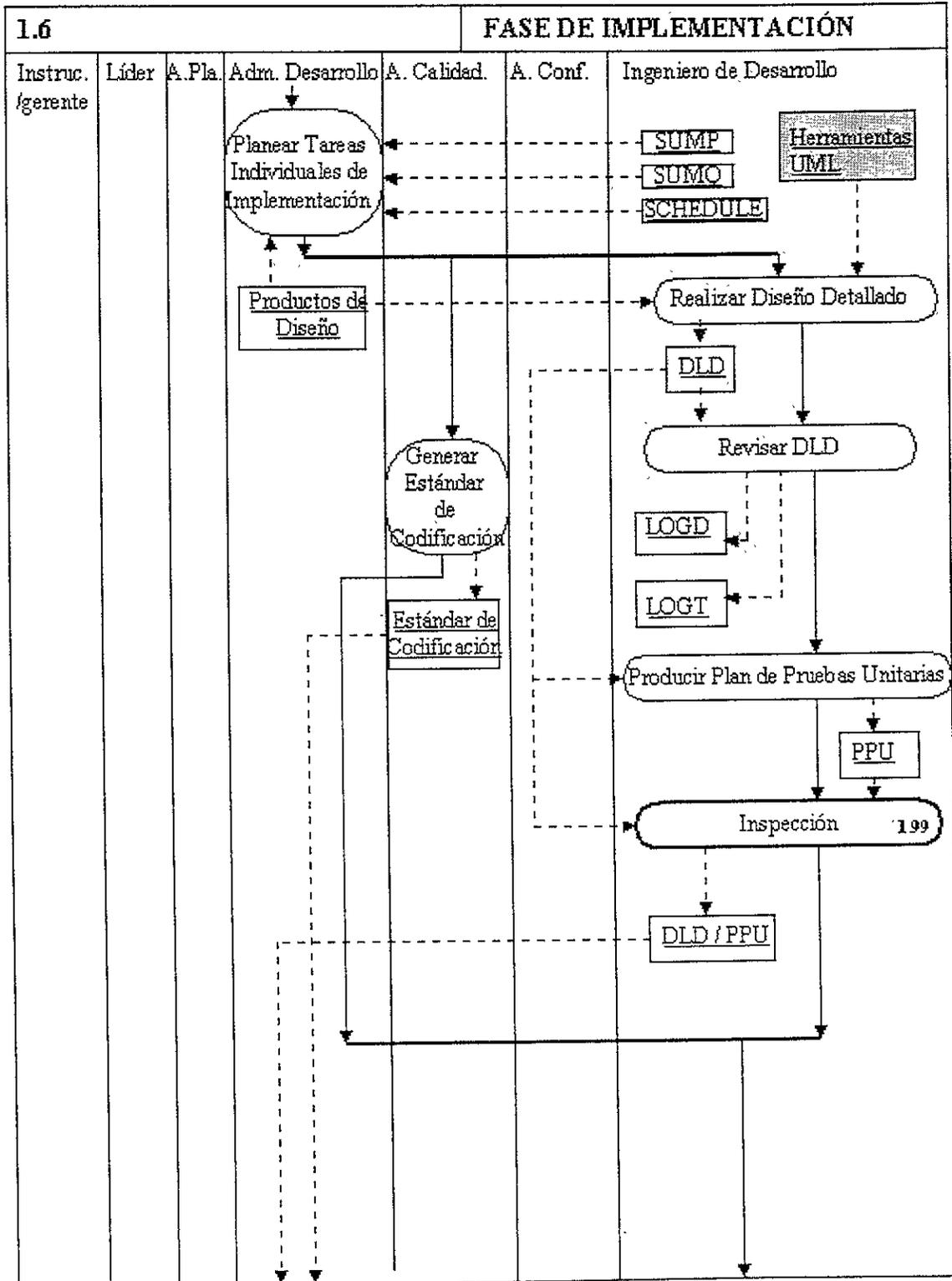


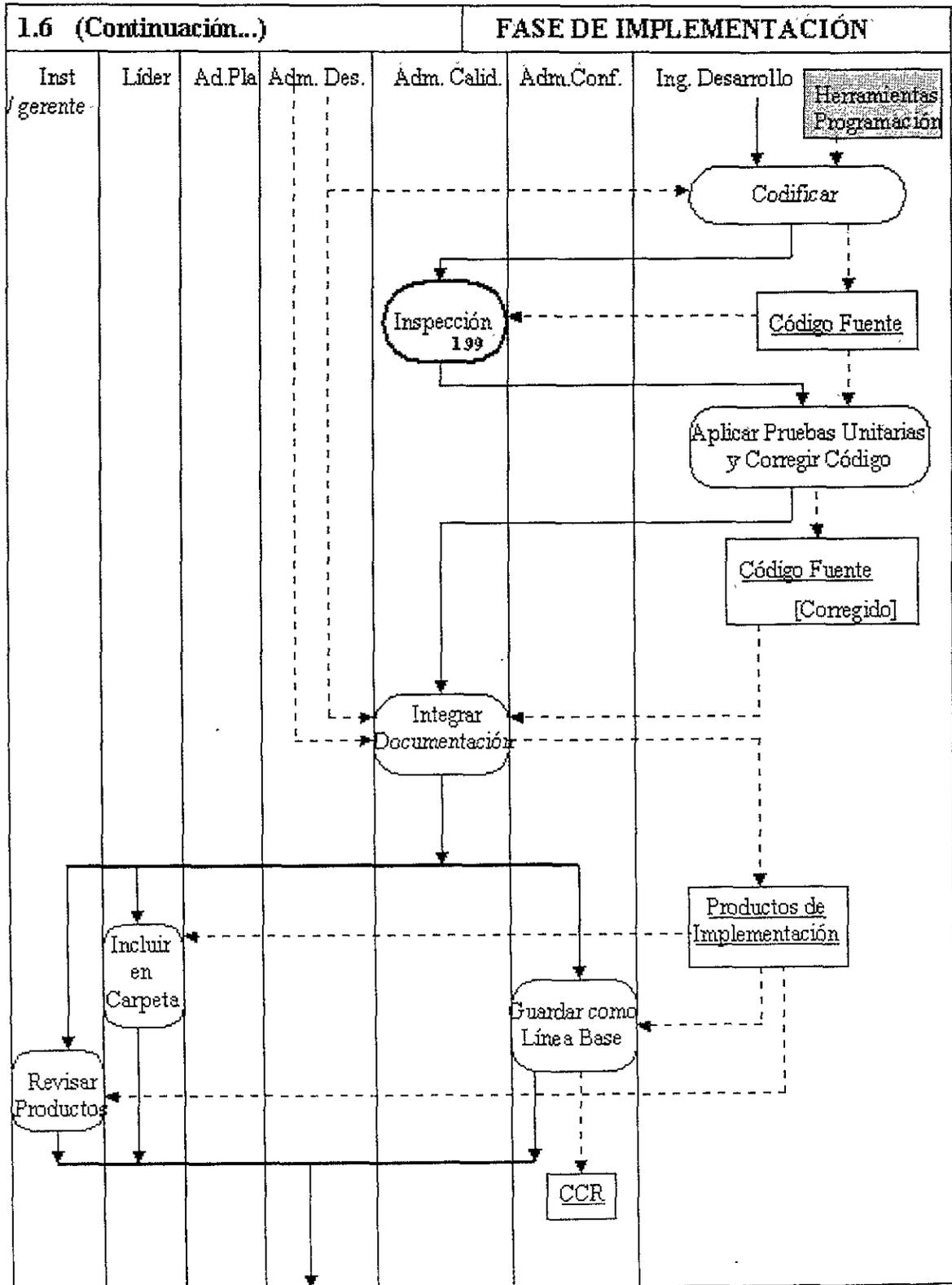
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definir Diseño Alto Nivel	Definir las funciones del producto para el primer ciclo	<ul style="list-style-type: none"> - Decidir la estructura del producto - Nombrar cada uno de los componentes - Producir las especificaciones para cada componente - Identificar todas las tareas de diseño que deben ser realizadas 	Productos de Requerimientos		Administrador de Desarrollo
Dividir Trabajo	Balancear la carga de trabajo para producir los artefactos de diseño	<ul style="list-style-type: none"> - Identificar todas las tareas necesarias para producir el documento de diseño (SDS) - Dividir el trabajo en partes y asignarlas a cada miembro del equipo 	Productos de Requerimientos		Administrador de Desarrollo
Definir estándares de diseño	Fijar los formatos y convenciones a utilizarse durante la producción de los artefactos de diseño	<ul style="list-style-type: none"> - Acordar formatos para la producción de los documentos de diseño - Entregar a cada miembro del equipo una copia del estándar a utilizar 	Estándar de Documentación	Estándar de Diseño	Administrador de Calidad
Definir diagramas de diseño	Determinar el comportamiento dinámico para cada caso de uso	<ul style="list-style-type: none"> - Identificar clases de diseño y realizar diagrama de clases - Realizar diagramas de Interacción por cada caso de Uso - Incluir el flujo normal y excepcional 	Estándar de Documentación Estándar de Diseño Productos de Requerimientos		Ingeniero de Desarrollo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Generar SDS	Generar el Documento de Diseño de Alto Nivel (SDS)	<ul style="list-style-type: none"> - Integrar el Documento de Diseño de Alto Nivel (SDS) <ul style="list-style-type: none"> • Diagramas de clases del diseño • Diagramas de interacción con flujo normal y excepcional • Diagramas de Componentes • Diseño de Base de Datos 		SDS (Diseño de Alto Nivel)	Administrador de Desarrollo
Generar Plan de Pruebas de Integración	Producir el Plan de Pruebas de Integración del Sistema	<ul style="list-style-type: none"> - Identificar la dependencia entre clases - Analizar el orden de integración y la forma de probar cada dependencia 	Estándar de Documentación SDS	PPI (Plan de Pruebas de Integración)	Administrador de Desarrollo
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de Documentación Estándar de Diseño SDS PPI	Productos de Diseño Estándar de Diseño SDS PPI CCR	Administrador de Calidad / Gerente o Instructor

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Diseño		Líder del Equipo
Guardar como línea Base	Resguardar los documento de diseño	- Asignar una versión al documento de diseño Guardarlo en Línea Base - Registrar datos en la forma CCR	Productos de Diseño	Forma CCR	Administrador de Configuración

2.3.6 Implementación del Primer Ciclo de TSPi





Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Realizar Tareas Operacionales de Planeación	Realizar una planeación y asignación de las tareas de implementación	<ul style="list-style-type: none"> - Revisar el trabajo realizado y los estimados registrados en las formas SUMP y SUMQ - Asignar las tareas de acuerdo al componente diseñado por cada ingeniero de desarrollo o bien de acuerdo a sus habilidades e intereses - Cada ingeniero planea su trabajo de implementación de acuerdo a estimados de tamaño y tiempo - Actualizar las formas SCHEDULE, SUMP y SUMQ 	Productos de Diseño Formas SCHEDULE, SUMP y SUMQ	Actualización de Formas SCHEDULE, SUMP y SUMQ	Administrador de Desarrollo
Realizar el Diseño Detallado	Detallar componentes del diseño de alto nivel	<ul style="list-style-type: none"> - Detallar atributos y métodos para las clases de diseño, clases de interfaz, manejo de datos y otras que formen parte del diseño - Detallar diagramas de interacción con flujo normal y excepcional definiendo tipos de parámetros - Definir los diagramas de componentes y paquetes - Definir los tipos de los campos, restricciones y relaciones de la Base de Datos 	Productos de Diseño	DLD (Diseño de Nivel Detallado)	Ingeniero de Desarrollo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Realizar DLD	Realizar una revisión individual del diseño de nivel detallado	<ul style="list-style-type: none"> - Realizar una revisión personal de cada módulo, objeto o clase. Se puede hacer uso de una lista de verificación - Verificar los ciclos, condiciones y caminos lógicos, valores nominales y de límite de variables y parámetros. Se puede realizar un análisis de estados de maquina o usar tablas de seguimiento - Registrar tiempos y defectos encontrados en las formas LOGD y LOGT 	DLD (Diseño de Nivel Detallado)	Forma LOGD Forma LOGT	Ingeniero de Desarrollo
Definir Plan de pruebas unitarias	Definir un plan para verificar el buen funcionamiento de cada componente	<ul style="list-style-type: none"> - Analizar la forma de probar cada atributo y método de clase - Planear una prueba de funcionamiento de la clase 	DLD (Diseño de Nivel Detallado)	PPU (Plan de Pruebas Unitarias)	Ingeniero de Desarrollo
Definir estándar de codificación	Unificar los lineamientos a seguir en la generación del código fuente del proyecto	<ul style="list-style-type: none"> - Acordar entre todos los miembros del equipo los estándares de codificación <p>Nombres de clases, variables, interfaces Uso de sentencias Uso de comentarios Tabulaciones y otros que se consideren necesarios en la uniformidad de código</p>		Estándar de Codificación	Administrador de Calidad

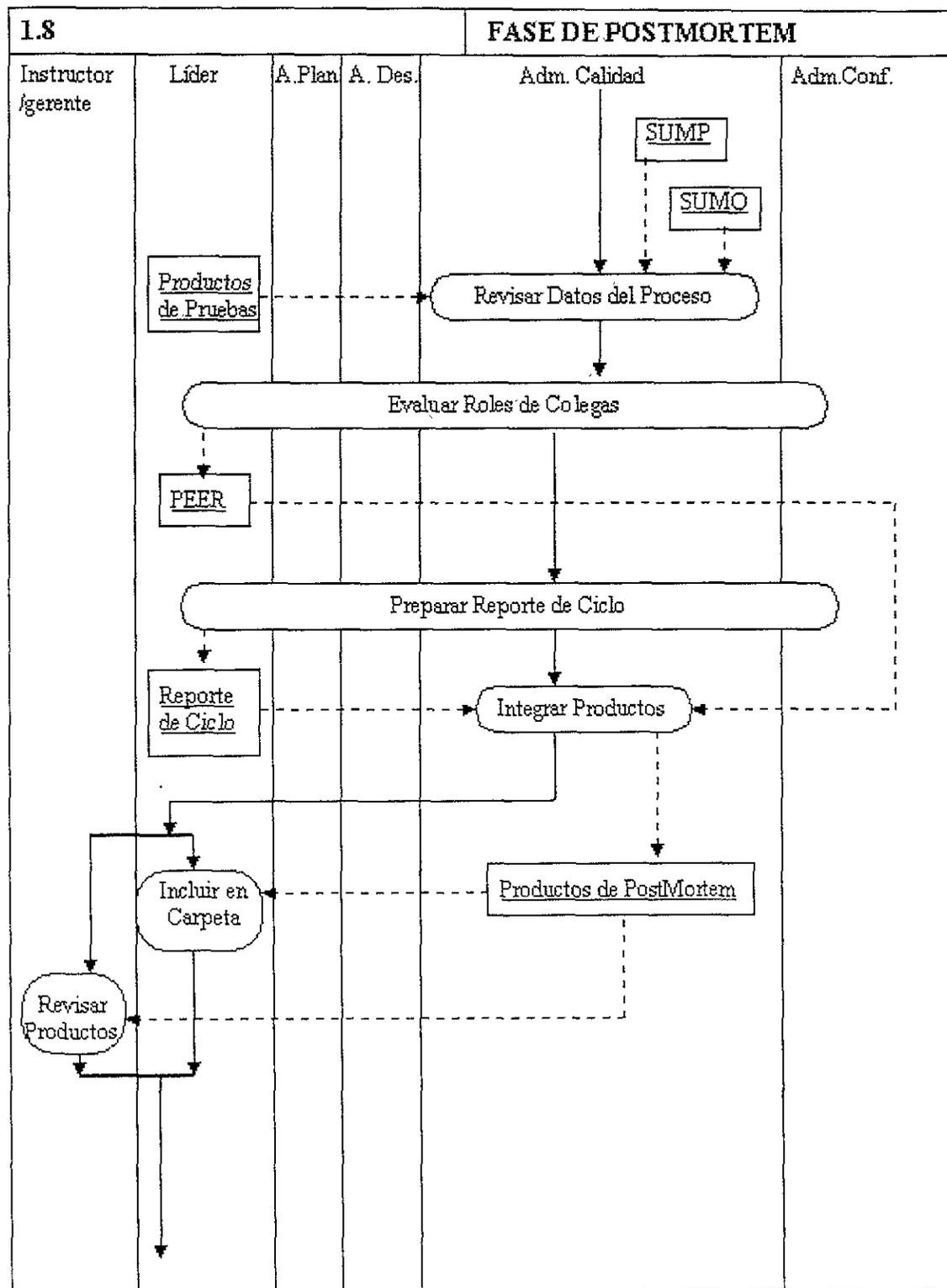
Etapa	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Codificar	Trasladar el diseño de nivel detallado a un lenguaje de programación orientado a objetos	<ul style="list-style-type: none"> - Ingresar el código de acuerdo al esquema del diseño detallado siguiendo el estándar de codificación - Compilar y componer los defectos encontrados hasta eliminarlos por completo - Actualizar las formas LOGD y LOGT 	DLD (Diseño de Nivel Detallado) Estándar de Codificación	Código Fuente Forma LOGD Forma LOGT	Ingeniero de Desarrollo
Realizar Pruebas Unitarias y Corregir Código	Probar eficientemente el código para cada módulo	<ul style="list-style-type: none"> - Ejecutar el plan de pruebas unitarias creado en el diseño detallado - Si es el caso, eliminar defectos encontrados en la codificación 	PPU (Plan de Pruebas Unitarias) Código Fuente	Código Fuente corregido	Ingeniero de Desarrollo
Generación de documentación	Generar el Documento de Implementación	<ul style="list-style-type: none"> - Integrar el Documento de Diseño de Nivel Detallado (DLD) <ul style="list-style-type: none"> • Diagramas de clases del diseño con atributos y especificaciones • Diagramas de interacción con parámetros, atributos y especificaciones • Diagrama de componentes detallado • Diagrama de instalación • Base de Datos normalizada con tipos y restricciones 		DLD (Diseño de Nivel Detallado) PPU (Plan de Pruebas Unitarias) Código	Administrador de Desarrollo

Etapa	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
		<ul style="list-style-type: none"> - Integrar Plan de Pruebas unitarias Integrar el Código de todas las clases 			
Revisar los productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de Documentación Estándar de Diseño Estándar de Codificación DLD PPU Código Fuente	Productos de Implementación Estándar de Codificación DLD PPU Código Fuente CCR	Administrador de Calidad / Gerente o Instructor
en la carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	Productos de Implementación		Líder del Equipo
Guardar como Base	Resguardar los documentos de implementación	<ul style="list-style-type: none"> - Asignar una versión al documento de implementación - Guardarlo en Línea Base - Registrar datos en la forma CCR 	Productos de Implementación	Forma CCR	Administrador de Configuración

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definición de pruebas	Definir las tareas necesarias para realizar las pruebas del producto de software	<p>- Definir todas las tareas para realizar las pruebas del producto de software como son:</p> <ul style="list-style-type: none"> • Desarrollar los procedimientos para la prueba de integración • Desarrollar los procedimientos para la prueba del sistema • Registrar los tiempos y tamaño de las pruebas • Revisar los datos obtenidos • Verificar la corrección de defectos 	Productos de Implementación		Administrador de Desarrollo
Construcción del producto	Verificar que el producto de software este completo en todas sus partes	<p>- Verificar que todas las partes del producto de software estén terminadas y de acuerdo a lo planeado</p> <p>- Construir el producto de software y prepararlo para la prueba de integración</p>	Productos de Implementación		Administrador de Desarrollo Participantes Equipo de Trabajo
Definición de documentación del usuario	Producir la información básica necesaria para que el usuario explote el sistema	<p>- Preparar la estructura general del manual de acuerdo a las necesidades del usuario</p> <p>- Incluir en el manual las secciones de :</p> <ul style="list-style-type: none"> • Cómo iniciar • Explicación de qué es lo que hace el sistema • Glosario de términos • Mensajes de error y procedimientos de solución 	SRS PPS	Documentación del Usuario	Administrador de Calidad

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
		<ul style="list-style-type: none"> • Lista detallada del contenido 			
Activos	Recopilar, organizar y revisar todos los documentos	<ul style="list-style-type: none"> - Reunir cada una de las formas y artefactos resultantes de la realización de las pruebas - Revisar la organización y cumplimiento de estándares del manual de usuario y los artefactos de pruebas 	Lista de Verificación Estándar de Documentación del Usuario Sistema Probado Formas LOGD Formas LOGTEST	Productos de Pruebas Documentación del Usuario Sistema Probado Formas LOGD Formas LOGTEST	Administrador de Calidad
Activos	Revisar el cumplimiento de estándares y la consistencia entre el sistema y la documentación del usuario	<ul style="list-style-type: none"> - Realizar la ejecución del sistema verificando que lo contenido en el manual sea fiel a las pantallas mostradas por el sistema - Realizar las correcciones correspondientes de ser necesario 	Documentación del Usuario Sistema Probado PPS	Productos de Pruebas Documentación del Usuario Sistema Probado Formas LOGD Formas LOGTEST	Administrador de Calidad / Gerente o Instructor
Entrada	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	Productos de Pruebas Sistema Probado Documentación de Usuario Formas LOGTEST Formas LOGD		Líder del Equipo
Trabaja como Base	Resguardar los documentos de pruebas	<ul style="list-style-type: none"> - Asignar una versión al documento de pruebas - Guardarlo en Línea Base - Registrar datos en la forma CCR 	Productos de Pruebas	Forma CCR	Administrador de Configuración

2.3.8 PostMortem del Primer Ciclo de TSPI



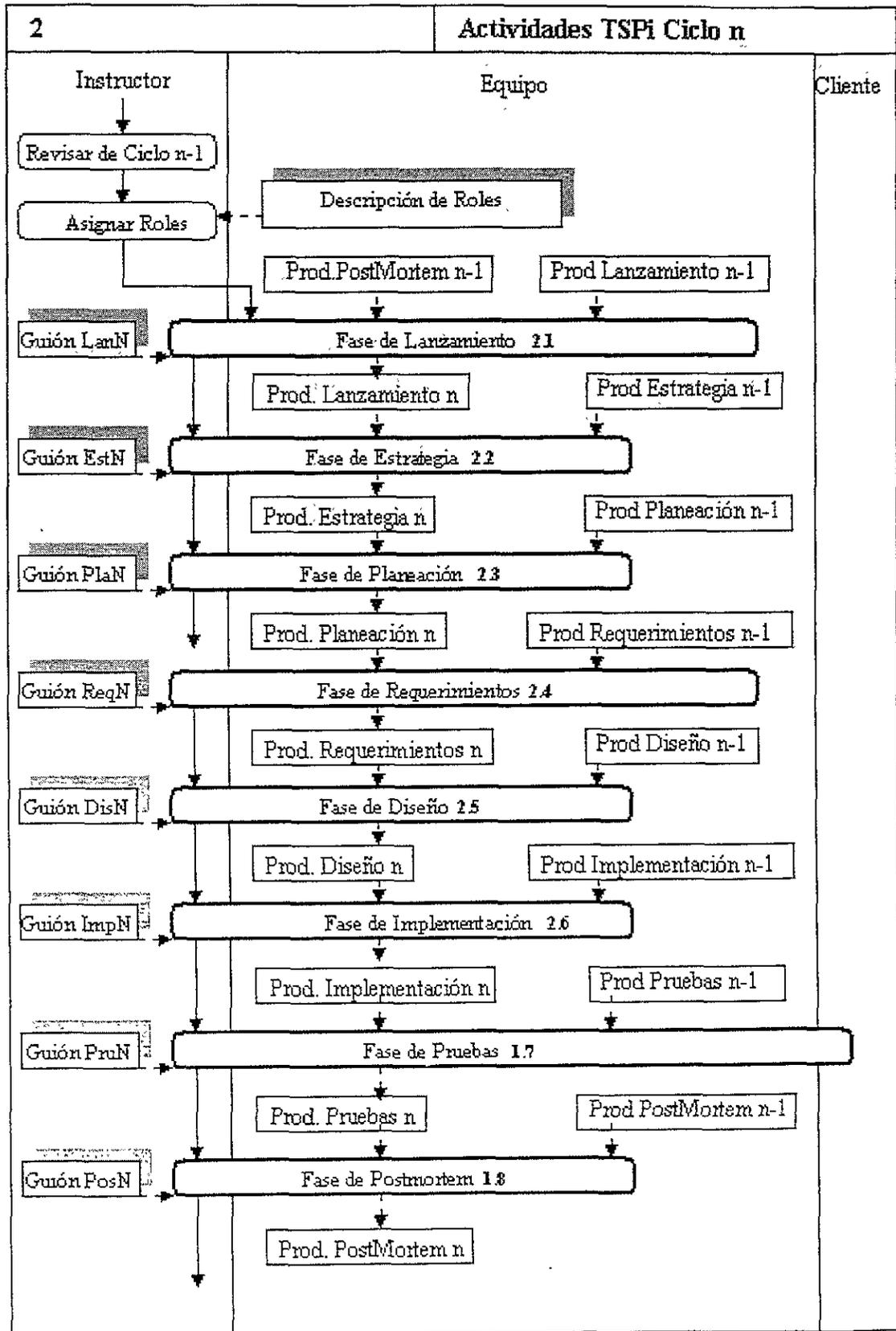
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Comparar los datos del ciclo con los datos planeados y evaluar la calidad del producto	Comparar los datos obtenidos con los datos planeados y evaluar la calidad del producto	<ul style="list-style-type: none"> - Evaluar los siguientes aspectos: <ul style="list-style-type: none"> • Desempeño actual comparado con el plan • Lecciones aprendidas en el ciclo actual • Mejoras que pueden realizarse • Dónde se localizaron problemas y registrarlos para corregirlos en el siguiente ciclo 	Productos de Pruebas Forma SUMP Forma SUMQ		Administrador de Calidad
Evaluar la efectividad en el desempeño de cada uno de los roles	Evaluar la efectividad en el desempeño de cada uno de los roles	<ul style="list-style-type: none"> - Cada miembro del equipo evalúa el desempeño de cada uno de los roles - Registrar para cada rol : <ul style="list-style-type: none"> • Qué funcionó en el proceso • Dónde se encontraron problemas • Formas de mejorar el proceso y nuevas metas para ciclos posteriores - Realizar los reportes de acuerdo a la ayuda y eficacia del mismo - Registrar los datos obtenidos en la forma PEER 		Formas PEER	Líder del Equipo Participantes Todos los miembros del equipo
Evaluar el desempeño personal durante el primer ciclo de desarrollo	Evaluar el desempeño personal durante el primer ciclo de desarrollo	<ul style="list-style-type: none"> - Cada miembro del equipo evalúa su propio desempeño - Revisar qué funcionó durante el ciclo actual y que puede mejorar en ciclos posteriores - Indicar si las mejoras sugeridas fueron eficaces - Evaluar el desempeño del equipo de acuerdo a la perspectiva del rol asignado y de ingeniero de desarrollo 		Reporte de Ciclo	Líder del Equipo Participantes Todos los miembros del equipo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
		- Justificar las conclusiones obtenidas con los datos registrados El contenido del reporte es : <ul style="list-style-type: none"> • Tabla de contenido • Sumario • Reporte de Roles Líder del Equipo Administrador de Desarrollo Administrador de Planeación Administrador de Calidad Administrador de Configuración <ul style="list-style-type: none"> • Reporte de Ingenieros de Desarrollo 			
Registrar productos	Realizar la integración y revisión de productos de PostMortem	- Integrar el reporte de ciclo de desarrollo actual - Revisar conforme a estándares el reporte y formas PEER - Registrar defectos encontrados en formas LOGD - Entregar documentación con defectos a propietario para corrección	Formas PEER Reporte de Ciclo	Productos de PostMortem	Administrador de Calidad
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	- Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección	Lista de Verificación Estándar de Documentación Formas PEER Reporte del Ciclo	Productos de PostMortem Formas PEER Reporte del Ciclo	Administrador de Calidad / Gerente o Instructor

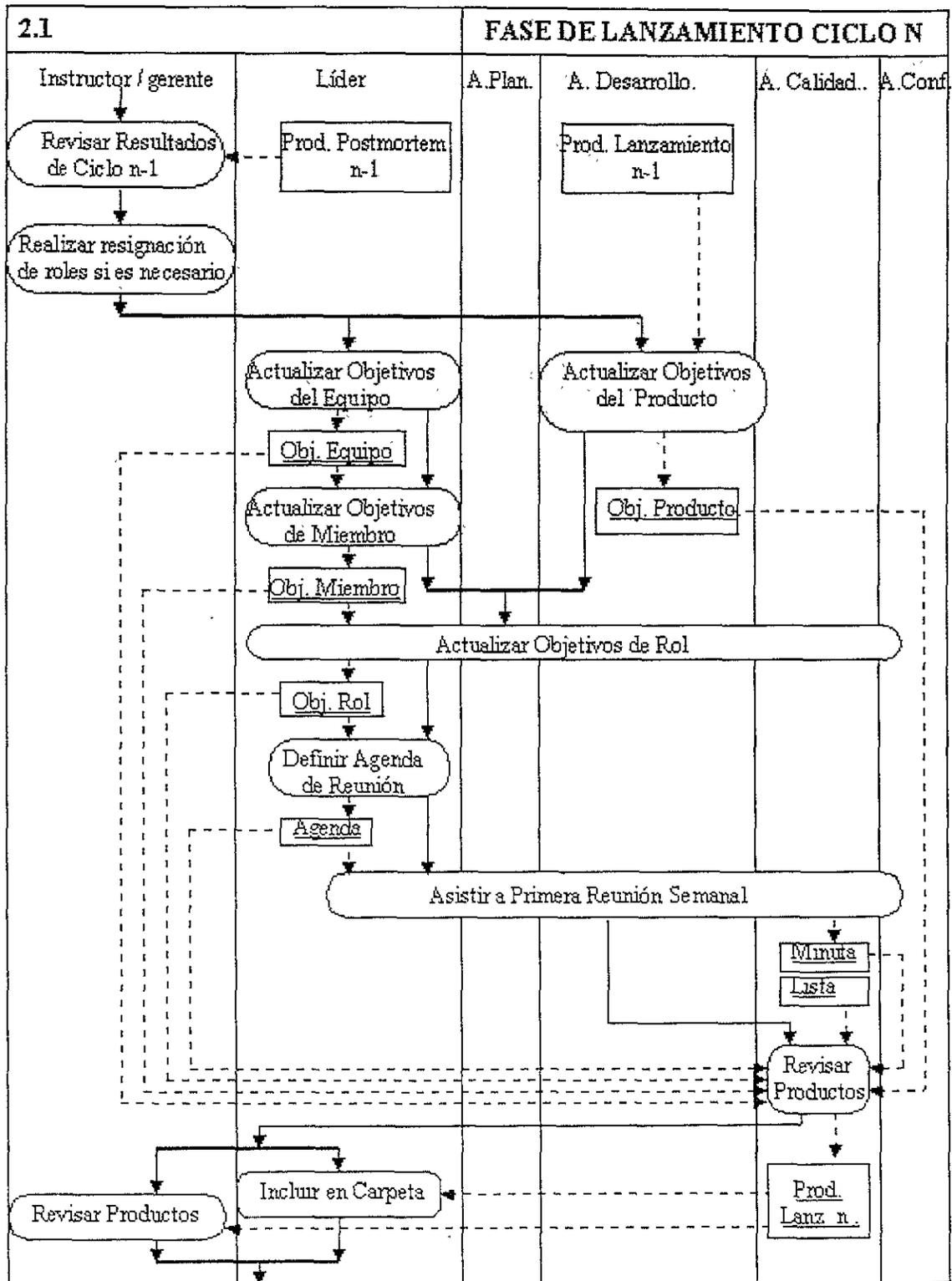
Etapa	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Ir en etapa	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de PostMortem Formas PEER Reporte de Ciclo		Líder del Equipo

2.4 Ciclo n de TSPi

En esta sección se presentan los diagramas y tablas correspondientes a las todas las fases del ciclo n de desarrollo.



2.4.1 Lanzamiento del Ciclo n de TSPi



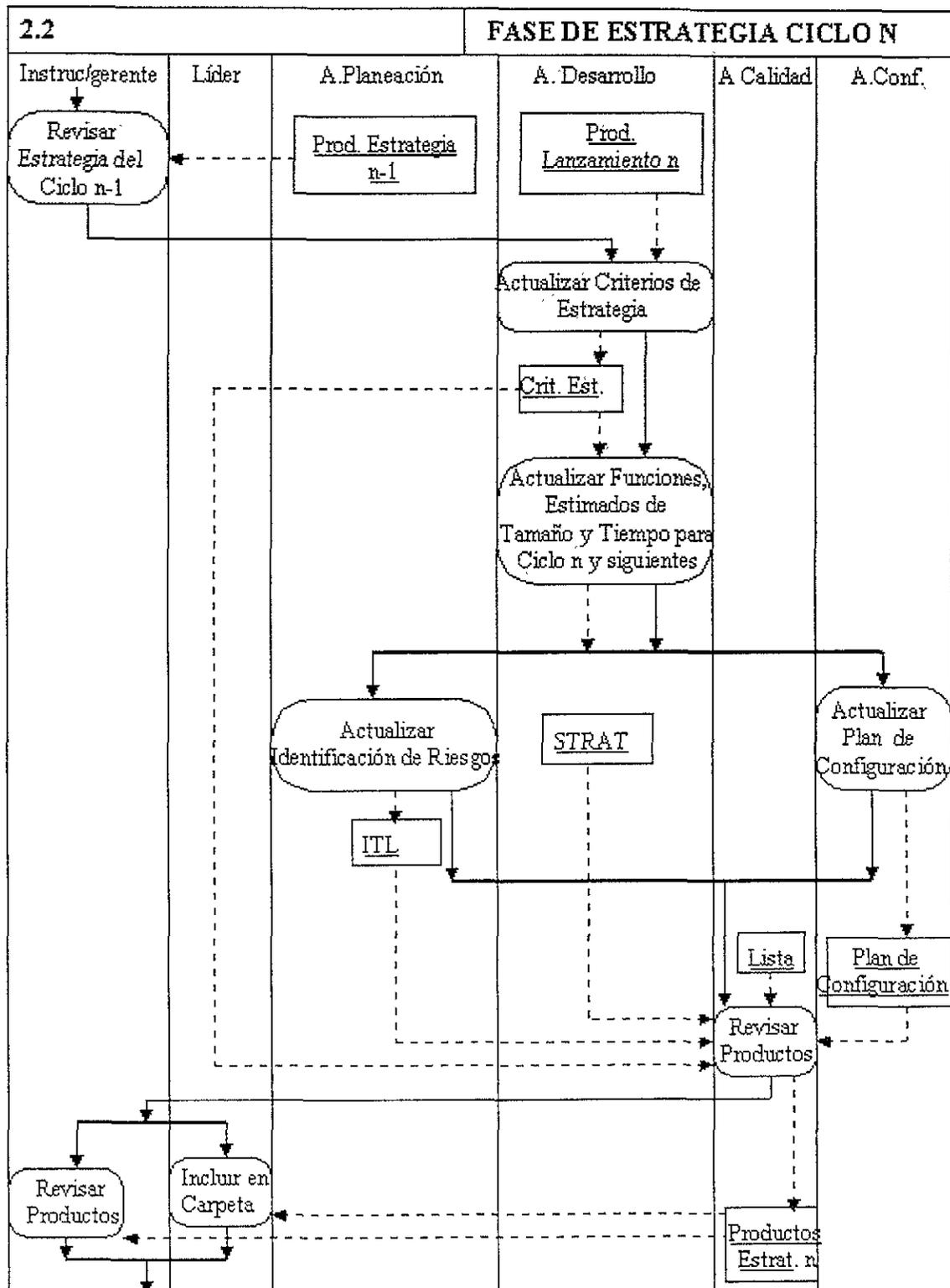
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Comparar resultados del ciclo n-	Comparar resultados obtenidos en la fase de lanzamiento del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de lanzamiento del ciclo anterior - Acordar formas de mejorar el proceso de lanzamiento actual 	Productos de PostMortem de ciclos anteriores Productos de Lanzamiento de ciclos anteriores		Instructor / Gerente
Lanzar (función funcional)	Intercambiar roles en caso de ser necesario	<ul style="list-style-type: none"> - Revisar datos obtenidos de formas PEER - Acordar cambios de Rol 	Productos de PostMortem ciclo anterior		Instructor / Gerente
Definir metas reales y alcanzables como equipo	Definir metas reales y alcanzables como equipo Fijar criterios cuantitativos para comparar los resultados alcanzados	<ul style="list-style-type: none"> - Actualizar los tres objetivos fundamentales propuestos en TSPi y sus métricas <ul style="list-style-type: none"> • Producir un producto de calidad • Poner en funcionamiento un proyecto bien administrado • Finalizar a tiempo - Actualizar objetivos complementarios que sean importantes en la construcción del producto - Acordar con el equipo todos los objetivos y métricas propuestos - Entregar copia de objetivos a cada miembro del equipo 	Estándar de Documentación Productos de Lanzamiento de ciclos anteriores	Objetivos del equipo	Líder del equipo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
<p>Actualizar objetivos miembro equipo</p>	<p>Actualizar metas reales y alcanzables como integrante del equipo Fijar métricas cuantitativas para comparar los resultados obtenidos con los objetivos planeados</p>	<p>- Actualizar los cuatro objetivos fundamentales propuestos en TSPi y sus métricas</p> <ul style="list-style-type: none"> • Ser un miembro de equipo cooperativo y efectivo • Realizar un trabajo personal disciplinado • Planear y seguir todo el trabajo personal • Producir un producto de calidad como ingeniero de desarrollo <p>- Actualizar objetivos complementarios que sean importantes en la construcción del producto - Acordar con el equipo todos los objetivos y métricas propuestos - Entregar copia de objetivos a cada miembro del equipo</p>	<p>Estándar de Documentación Productos de Lanzamiento de ciclos anteriores</p>	<p>Objetivos de miembro del equipo</p>	<p>Líder del equipo</p>
<p>Actualizar objetivos por rol</p>	<p>Actualizar las metas reales y alcanzables en el rol asignado Fijar métricas cuantitativas para comparar los resultados obtenidos con los objetivos planeados</p>	<p>- Actualizar los cuatro objetivos fundamentales propuestos en TSPi por rol - Actualizar los objetivos complementarios que sean importantes en la construcción del producto</p>	<p>Estándar de Documentación Productos de Lanzamiento de ciclos anteriores</p>	<p>Objetivos de rol</p>	<p>Líder del equipo</p> <p>Participantes: Líder del equipo Administrador de Planeación Administrador de Desarrollo Administrador de Calidad Administrador de Configuración</p>

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Analizar objetivos del producto	Descripción de objetivos críticos del producto a ser satisfechos, objetivos opcionales y deseables y el criterio para evaluar un producto terminado	- Definir objetivos básicos del producto para el ciclo actual	Estándar de Documentación Productos de Lanzamiento de ciclos anteriores	Objetivos del Producto	Administrador de Desarrollo
Organizar agenda de reunión semanal	Organizar los temas a tratar durante la reunión semanal	- Definir temas a tratar durante la reunión asignando tiempos para cada uno - Incluir, si son necesarios, los temas sugeridos por los miembros del equipo - Distribuir tópicos a cada miembro del equipo	Estándar de Documentación	Agenda de Reunión	Líder del equipo
Participar a reunión semanal	Reunir a los miembros del equipo para discutir y acordar objetivos Actualizar los avances logrados e identificar riesgos y problemas	- Seguir la agenda de reunión - Registrar asuntos tratados y acuerdos (Administrador de Calidad)	Agenda de reunión	Minuta	Todo el equipo de trabajo
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	- Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección	Lista de Verificación Estándar de documentación Objetivos del equipo Objetivos de Miembro del Equipo Objetivos de Rol Objetivos de Producto Minuta	Productos de Lanzamiento Objetivos del Equipo Objetivos de Miembro del Equipo Objetivos de Rol Objetivos de Producto Minuta	Administrador de Calidad / Gerente o Instructor

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Lanzamiento		Líder del Equipo

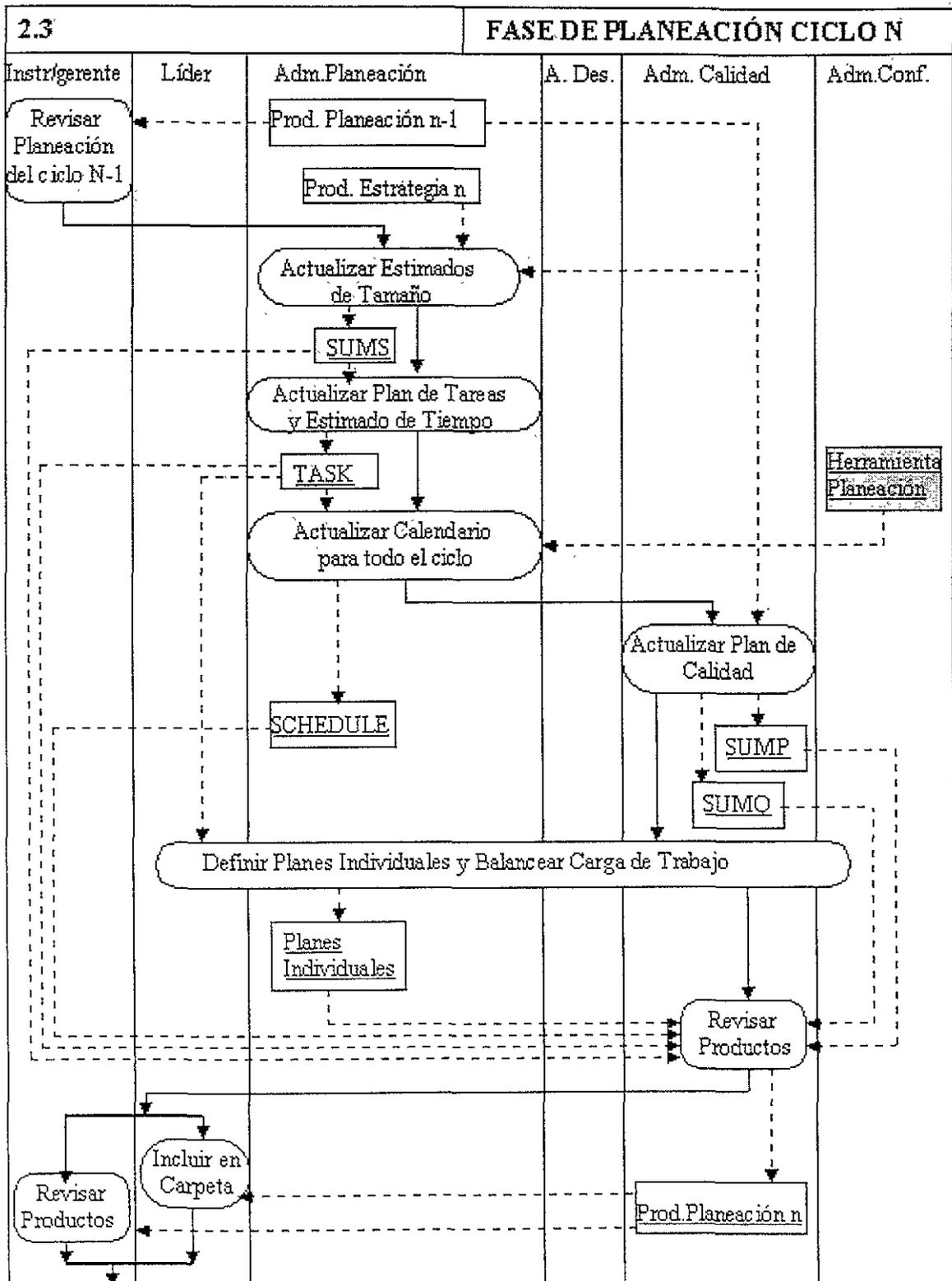
2.4.2 Estrategia del Ciclo n de TSPi



Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Desarrollar estrategia del ciclo n-1	Comparar resultados obtenidos en la fase de estrategia del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de estrategia del ciclo anterior - Acordar formas de mejorar el proceso de estrategia actual 	Productos de Estrategia de ciclos anteriores Productos de Lanzamiento de ciclo actual		Instructor / Gerente
Realizar criterios de estrategia	Elegir una estrategia de desarrollo para la construcción del producto	<ul style="list-style-type: none"> - Identificar componentes principales del producto para ciclo actual - Actualizar los criterios de Estrategia 	Estándar de Documentación Productos de Estrategia de ciclos anteriores Productos de Lanzamiento de ciclo actual	Criterios de Estrategia	Administrador de Desarrollo
Realizar la identificación de riesgos	Identificar y evaluar los principales riesgos durante todo el proyecto	<ul style="list-style-type: none"> - Revisar los riesgos del ciclo anterior y si es el caso añadir nuevos riesgos encontrados - Ordenar riesgos de acuerdo a su grado de probabilidad e impacto en el calendario del proyecto - Discutir y buscar formas de evitar cada uno de los riesgos en la reunión semanal - Guardar los resultados en la forma ITL 	Estándar de Documentación Forma STRAT Productos de Estrategia de ciclos anteriores	Forma ITL	Administrador de Planeación
Realizar cálculos de tamaño y tiempo para el ciclo n y siguientes	Estimar el tamaño del producto y el calendario de duración necesario para su desarrollo	<ul style="list-style-type: none"> - Realizar un calculo estimado del tamaño (p. Ej. Líneas de Código) para cada función del diseño conceptual, considerar cuántas unidades de tamaño podrían agregarse o modificarse - Realizar un calculo aproximado del número de horas necesarias para cada función de acuerdo al 	Estándar de Documentación Productos de Estrategia de ciclos anteriores Criterios de Estrategia	Forma STRAT	Administrador de Desarrollo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
		tamaño estimado - Registrar los datos obtenidos en la forma STRAT			
Actualizar el Plan de Configuración	Realizar un plan que detalle el manejo de versiones y los cambios realizados en los productos generados	- Revisar y realizar las modificaciones pertinentes en la definición del Plan de Configuración	Estándar de Documentación Productos de Estrategia de ciclos anteriores	Plan de Configuración	Administrador de Configuración
Revisar los productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	- Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección	Lista de Verificación Estándar de documentación Criterios de Estrategia Forma STRAT Forma ITL Plan de Configuración	Productos de Estrategia Criterios de Estrategia Forma STRAT Forma ITL Plan de Configuración	Administrador de Calidad / Gerente o Instructor
Organizar la carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Estrategia		Líder del Equipo

2.4.3 Planeación del Ciclo n de TSPI

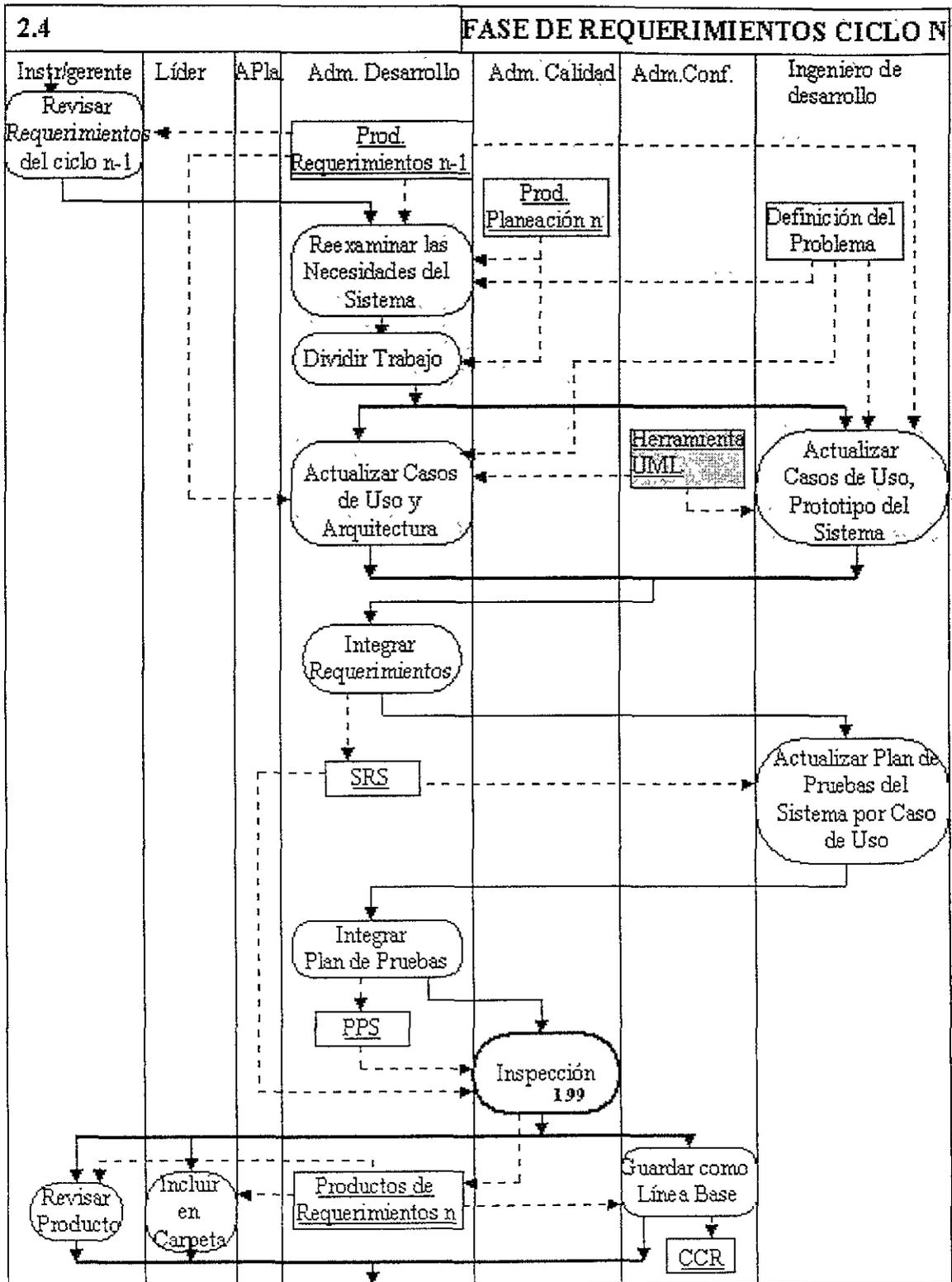


Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Comparar resultados obtenidos en la fase de planeación del ciclo anterior para mejorar el proceso en el ciclo actual	Comparar resultados obtenidos en la fase de planeación del ciclo anterior para mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de planeación del ciclo anterior - Acordar formas de mejorar el proceso de planeación actual 	Productos de Planeación de ciclos anteriores Productos de Estrategia del ciclo actual		Instructor / Gerente
Estimar tamaños de las partes del sistema y otros productos a desarrollar	Estimar tamaños de las partes del sistema y otros productos a desarrollar	<ul style="list-style-type: none"> - Realizar los cambios pertinentes a los estimados de Tamaño - Extraer los estimados registrados en la forma STRAT - Registrar todos los datos en la Forma SUMS 	Estándar de Documentación Productos de Estrategia de ciclo actual Productos de Planeación de ciclos anteriores	Forma SUMS	Administrador de Planeación
Estimar el tiempo para completar cada tarea planeada	Estimar el tiempo para completar cada tarea planeada	<ul style="list-style-type: none"> - Realizar los cambios pertinentes en la planeación de tareas y estimado de tiempo - Guardar todos los datos anteriores en la Forma TASK 	Estándar de Documentación Productos de Estrategia de ciclo actual Productos de Planeación de ciclos anteriores Forma SUMS	Forma TASK	Administrador de Desarrollo
Estimar el número de horas empleadas por semana en el desarrollo del producto	Estimar el número de horas empleadas por semana en el desarrollo del producto	<ul style="list-style-type: none"> - Realizar los cambios pertinentes en el calendario del ciclo - Guardar los datos en la Forma SCHEDULE 	Estándar de Documentación Forma TASK Productos de Planeación de Ciclos anteriores	Forma SCHEDULE	Administrador de Planeación

Realizar de ad	Establecer parámetros de calidad y métricas para comparar resultados obtenidos	<ul style="list-style-type: none"> - Realizar los cambios pertinentes en los valores planeados de tiempo y tamaño de acuerdo a los datos de la forma TASK - Realizar los cambios pertinentes al estimado de defectos inyectados y removidos para cada producto y tarea - Registrar los datos en la Forma SUMP - Realizar los cambios pertinentes a las tazas de defectos y rendimiento para cada fase - Registrar los datos en la Forma SUMQ 	Estándar de Documentación Forma TASK Productos de Planeación de Ciclos anteriores	Forma SUMP Forma SUMQ	Administrador de Calidad
Revisar es individuales	Acordar los tiempos y tareas para cada uno de los miembros del equipo de acuerdo a lo planeado	<ul style="list-style-type: none"> - Analizar las horas estimadas para cada tarea y la disponibilidad de horas de cada integrante - Asignar las tareas para cada ingeniero por semana - Entregar copia de las Formas TASK y SCHEDULE a cada miembro del equipo - Analizar las tareas asignadas y balancear si es necesario el trabajo asignado 	Estándar de Documentación Forma TASK Forma SCHEDULE	Planes Individuales	Administrador de Planeación Participantes: Líder del Equipo Administrador de Planeación Administrador de Desarrollo Administrador de Calidad Administrador de Configuración
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de documentación Forma SUMS Forma TASK Forma SCHEDULE Forma SUMP Forma SUMQ Planes Individuales	Productos de Planeación Forma SUMS Forma TASK Forma SCHEDULE Forma SUMP Forma SUMQ Planes Individuales	Administrador de Calidad / Gerente o Instructor

ir en eta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Planeación		Líder del Equipo
--------------	---	--	-------------------------	--	------------------

2.4.4 Requerimientos del Ciclo n de TSPi

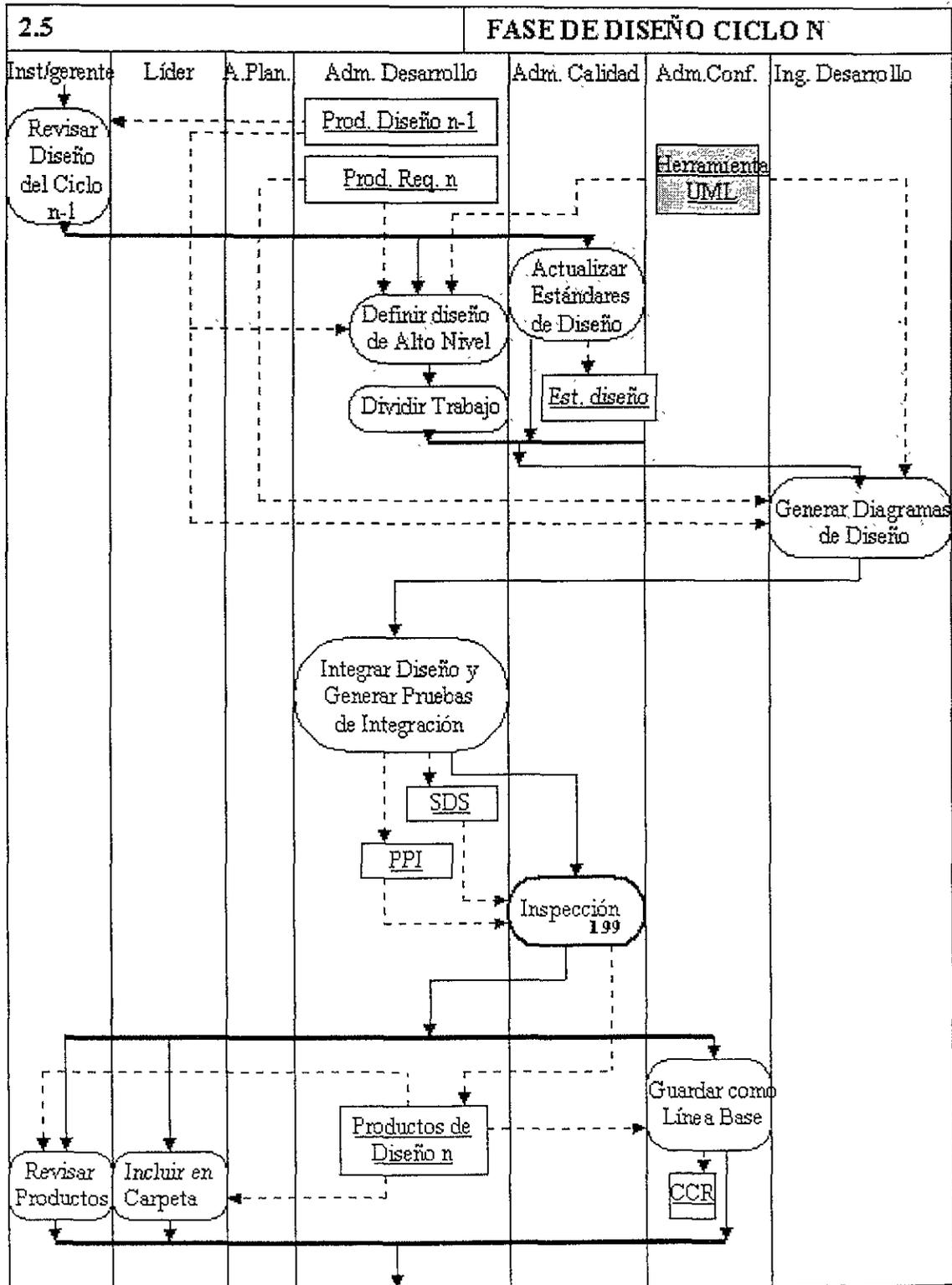


Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Analizar requerimientos del ciclo n-1	Comparar resultados obtenidos en la fase de requerimientos del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de requerimientos del ciclo anterior - Acordar formas de mejorar el proceso de requerimientos actual 	Productos de Requerimientos de ciclos anteriores Productos de Planeación de ciclo actual		Instructor / Gerente
Examinar las necesidades del sistema	Clarificar y unificar el entendimiento acerca de las necesidades del sistema entre todos los miembros del equipo y el cliente para el ciclo actual	<ul style="list-style-type: none"> - Cada ingeniero revisa las necesidades para la nueva versión del sistema - Entrevistarse con el cliente para aclarar puntos que podrían ser ambiguos - Identificar claramente las funciones que serán realizadas en el ciclo actual - Actualizar los datos obtenidos 	Definición de Problema Productos de Requerimientos de ciclos anteriores Productos de Planeación		Administrador de Desarrollo Participantes: Líder del Equipo Administrador de Planeación Administrador de Desarrollo Administrador de Calidad Administrador de Configuración Cliente
Dividir Trabajo	Balancear la carga de trabajo para el documento de requerimientos	<ul style="list-style-type: none"> - Identificar todas las tareas necesarias para producir el documento de requerimientos (SRS) - Dividir el trabajo en partes y asignarlas a cada miembro del equipo 	Productos de Planeación		Administrador de Desarrollo
Actualizar Casos de Uso y Arquitectura	Actualizar recursos necesarios y escenarios del proyecto en un nivel de abstracción alto	<ul style="list-style-type: none"> - Identificar los cambios que serán hechos a los requerimientos <ul style="list-style-type: none"> • Casos de Uso • Arquitectura del Sistema - Actualizar el Documento de Requerimientos 	Estándar de Documentación Definición de Problema Productos de Requerimientos de ciclos anteriores		Administrador de Desarrollo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Actualizar Casos de Uso, Prototipo del Sistema	Actualizar escenarios e interfaz de usuario por caso de uso	<ul style="list-style-type: none"> - Identificar los cambios que serán realizados a los requerimientos <ul style="list-style-type: none"> • Casos de Uso • Prototipo - Actualizar los escenarios de acuerdo a la funcionalidad del ciclo actual 	Estándar de Documentación Definición de Problema Productos de Requerimientos de ciclos anteriores		Ingeniero de Desarrollo
Generar requerimientos	Generar el Documento de requerimientos (SRS)	<ul style="list-style-type: none"> - Integrar el Documento de Requerimientos (SRS) de acuerdo a lo producido por cada Ingeniero de Desarrollo <ul style="list-style-type: none"> • Casos de Uso generales • Arquitectura • Casos de Uso por funcionalidad • Prototipo del Sistema 	Estándar de Documentación	SRS	Administrador de Desarrollo
Actualizar Plan de Pruebas del Sistema por caso de Uso	Realizar un plan de prueba por caso de uso del sistema	<ul style="list-style-type: none"> - Revisar y actualizar el plan de pruebas del sistema 	Estándar de Documentación SRS		Ingeniero de Desarrollo
Integrar Plan de Pruebas del Sistema	Producir el Plan de Pruebas del Sistema	<ul style="list-style-type: none"> - Integrar cada los planes de prueba por función producidos por los ingenieros de desarrollo 	Estándar de Documentación SRS	PPS (Plan de Pruebas del Sistema)	Administrador de Desarrollo
Revisar productos	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	Productos de Requerimientos		Líder del Equipo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Requerimientos		Líder del Equipo
Guardar como en la Base	Resguardar los documentos de requerimientos	<ul style="list-style-type: none"> - Asignar una versión al documento de requerimientos - Guardarlo en Línea Base - Guardar datos en la forma CCR 	Productos de Requerimientos	Forma CCR	Administrador de Configuración

2.4.5 Diseño del Ciclo n de TSPI



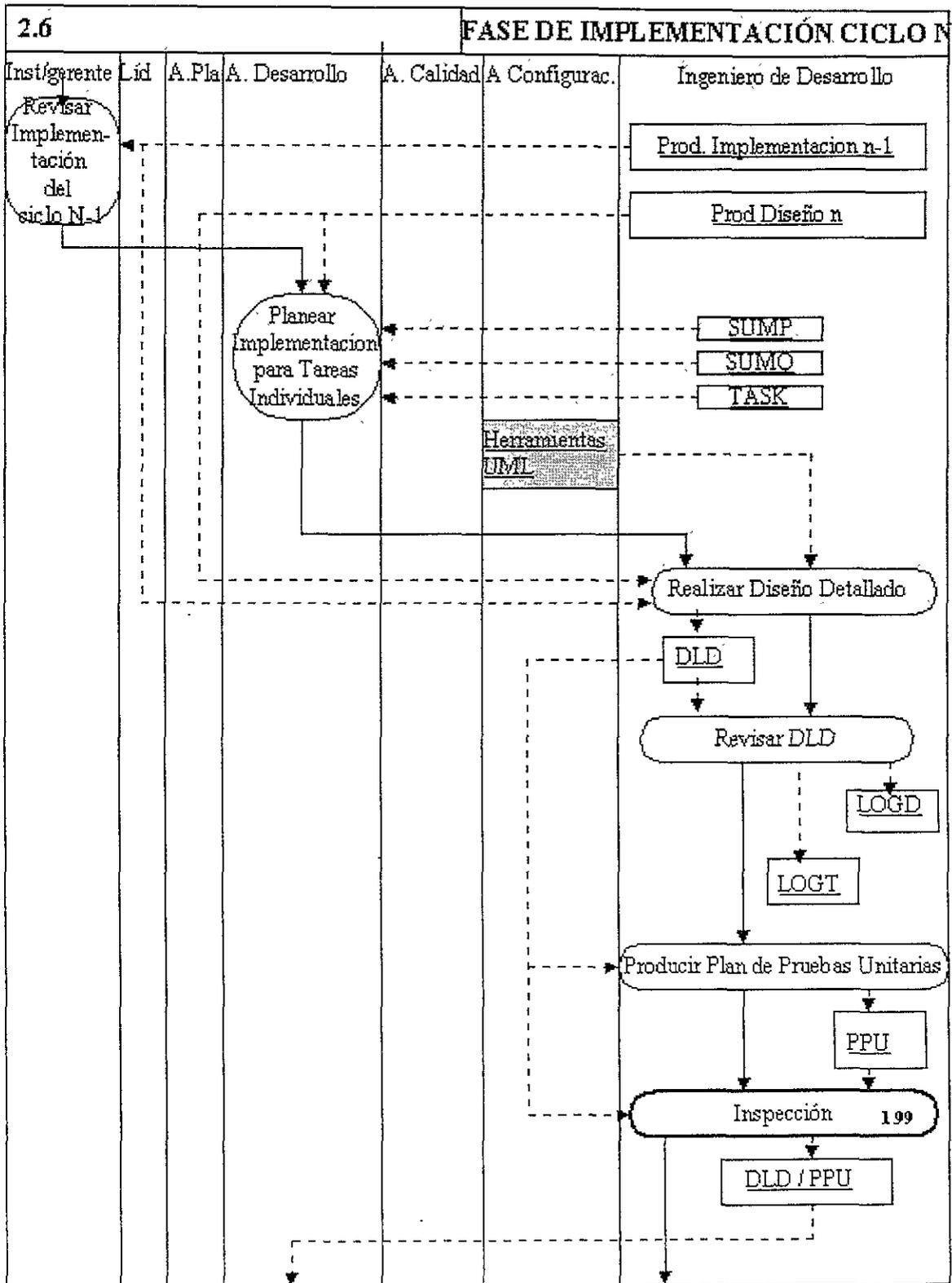
ESTA TESIS NO SALE DE LA BIBLIOTECA

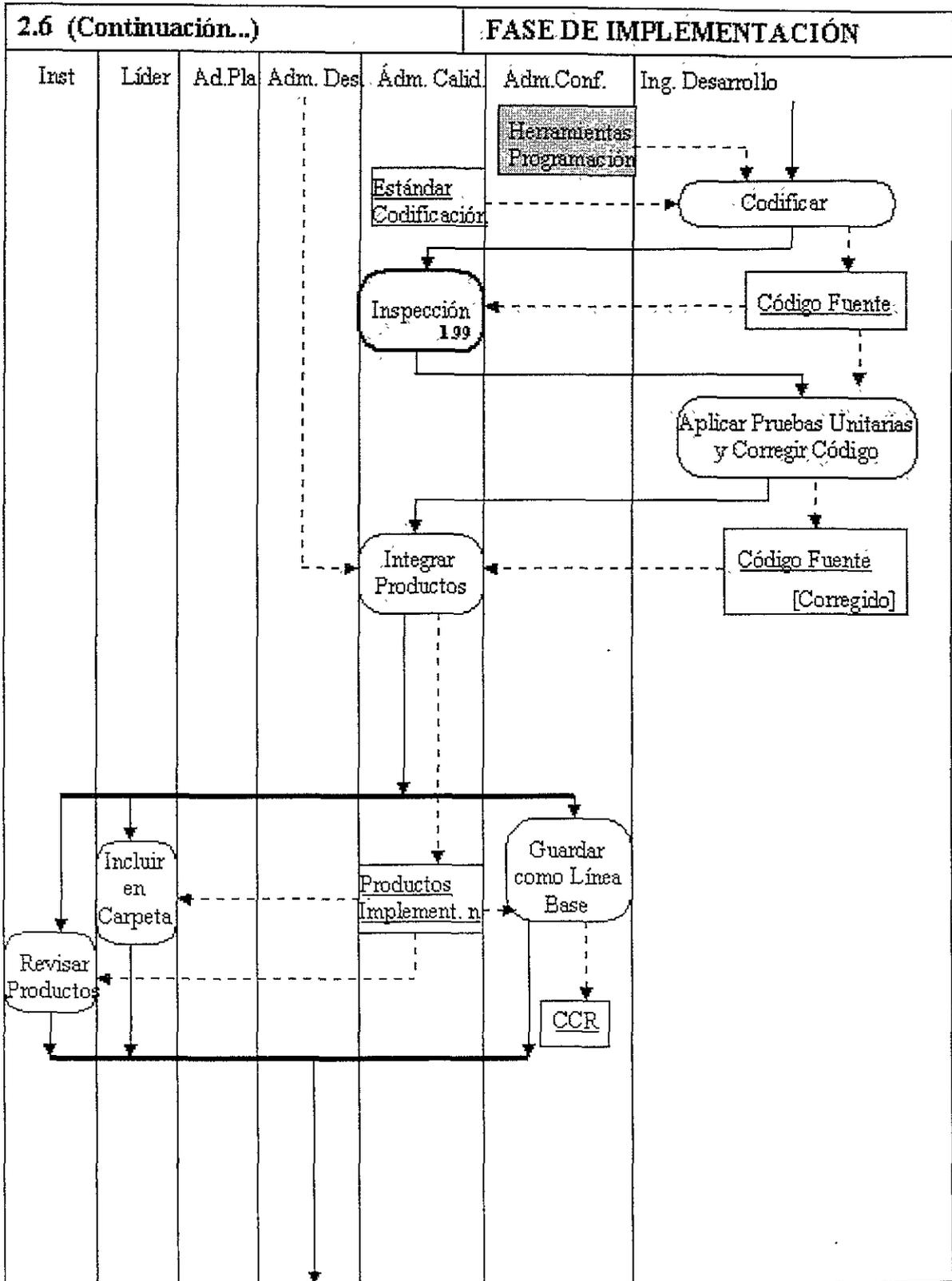
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Revisar el diseño del ciclo n-1	Comparar resultados obtenidos en la fase de diseño del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de diseño del ciclo anterior - Acordar formas de mejorar el proceso de diseño actual 	Productos de Diseño de ciclos anteriores Productos de Requerimientos de ciclo actual		Instructor / Gerente
Analizar el diseño de Nivel	Definir las funciones del producto para el ciclo actual	<ul style="list-style-type: none"> - Analizar y modificar estructura del producto - Nombrar cada uno de los componentes - Producir las especificaciones para cada componente - Identificar todas las tareas de diseño que deben ser realizadas 	Productos de Requerimientos del ciclo actual Productos de Diseño de ciclos anteriores		Administrador de Desarrollo
Dirigir el trabajo	Balancear la carga de trabajo para producir los artefactos de Diseño	<ul style="list-style-type: none"> - Identificar todas las tareas necesarias para producir el documento de diseño (SDS) - Dividir el trabajo en partes y asignarlas a cada miembro del equipo 	Productos de Requerimientos de ciclo actual Productos de Diseño de ciclos anteriores		Administrador de Desarrollo
Analizar estándares de Diseño	Fijar los formatos y convenciones a utilizarse durante la producción de los artefactos de Diseño	- Realizar si es necesario, las modificaciones al estándar de Diseño empleado en ciclos anteriores	Estándar de Documentación	Estándar de Diseño	Administrador de Calidad

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Crear Diagramas de Diseño	Determinar el comportamiento estático y dinámico para cada caso de uso. Modificar si es el caso, los diagramas anteriores y especificar las nuevas funciones para el ciclo actual	<ul style="list-style-type: none"> - Realizar diagramas de clases del diseño - Realizar diagramas de Interacción por cada caso de Uso - Incluir el flujo normal y excepcional 	Estándar de Documentación Estándar de Diseño Productos de Requerimientos de ciclo actual Productos de Diseño de ciclos anteriores		Ingeniero de Desarrollo
Actualizar Diagramas de Integración	Actualizar el Documento de Diseño de Alto Nivel (SDS) y Plan de Pruebas de Integración	<ul style="list-style-type: none"> - Hacer los cambios pertinentes y extender el Documento de Diseño de Alto Nivel (SDS) <ul style="list-style-type: none"> • Diagramas de clases del diseño • Diagramas de interacción con flujo normal y excepcional • Diagramas de Componentes • Diseño de Base de Datos - Actualizar y extender las dependencias entre clases y definir el orden de integración 		SDS (Diseño de Alto Nivel) PPI (Plan de Pruebas de Integración)	Administrador de Desarrollo
Revisar Productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de Documentación Estándar de Diseño SDS PPI	Productos de Diseño Estándar de Diseño SDS PPI CCR	Administrador de Calidad / Gerente o Instructor

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Diseño		Líder del Equipo
Guardar o en Línea Base	Resguardar los documentos de diseño	- Asignar una versión al documento de diseño - Guardarlo en Línea Base - Registrar datos en la forma CCR	Productos de Diseño	Forma CCR	Administrador de Configuración

2.4.6 Implementación del Ciclo n de TSPI





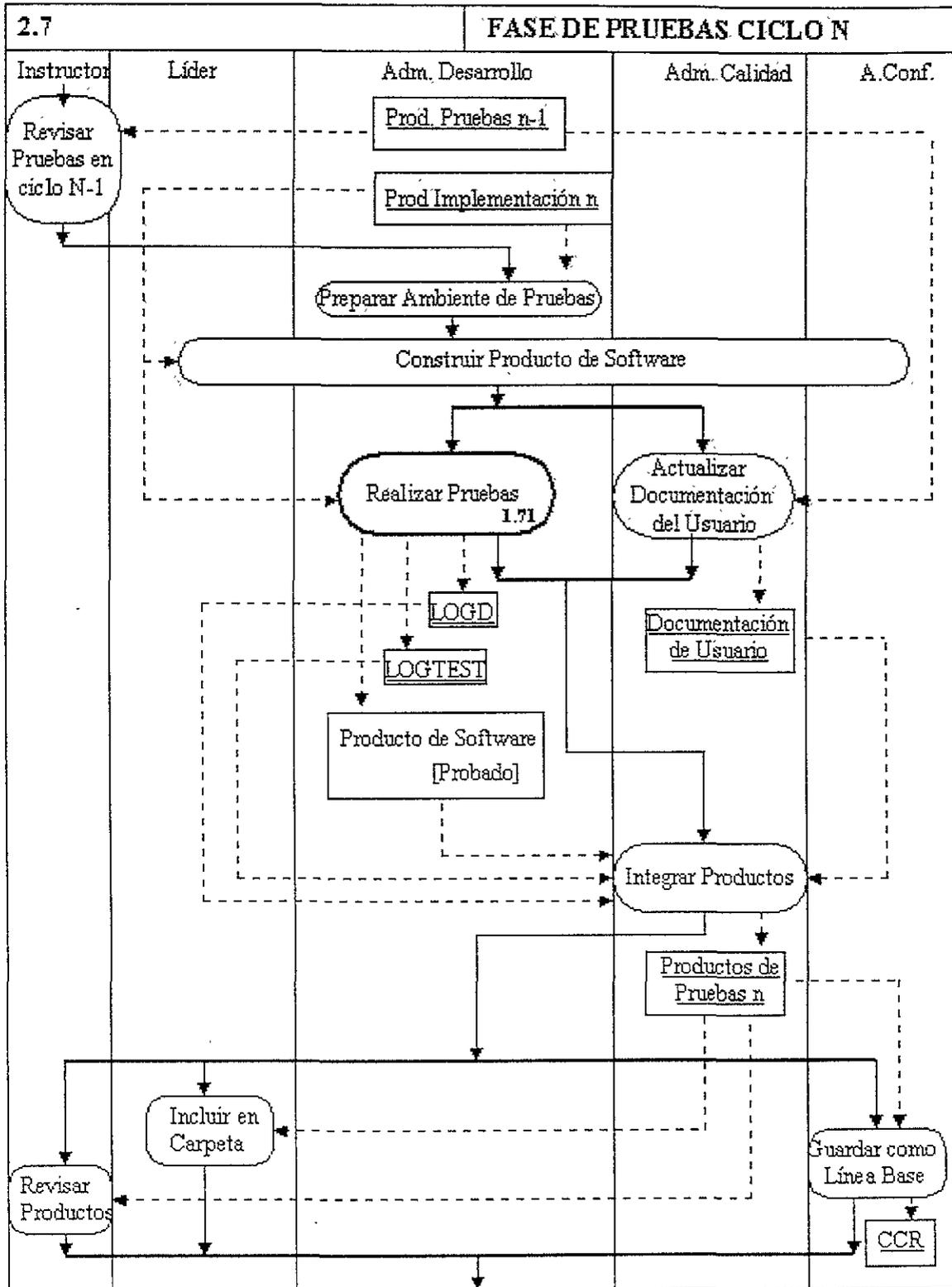
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Revisar implementación del Ciclo n-1	Comparar resultados obtenidos en la fase de implementación del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de implementación del ciclo anterior - Acordar formas de mejorar el proceso de implementación actual 	Productos de Implementación de ciclos anteriores Productos de Diseño de ciclo actual		Instructor / Gerente
Planear Tareas individuales de implementación	Realizar una planeación y asignación de las tareas de implementación Pasos	<ul style="list-style-type: none"> - Revisar el trabajo realizado en las fases de implementación de ciclos anteriores, la fase de diseño actual y los estimados registrados en las formas SUMP y SUMQ - Asignar las tareas de acuerdo al componente diseñado por cada ingeniero de desarrollo o bien de acuerdo a sus habilidades e intereses - Cada ingeniero planea su trabajo de implementación de acuerdo a estimados de tamaño y tiempo - Actualizar las formas SCHEDULE, SUMP y SUMQ 	Productos de Implementación de ciclos anteriores Productos de Diseño de ciclo actual Formas SUMP, SUMQ y SCHEDULE	Actualización de Formas SCHEDULE, SUMP y SUMQ	Administrador de Desarrollo
Actualizar diseño detallado	Detallar componentes del diseño de alto nivel	<ul style="list-style-type: none"> - Analizar la estructura de nivel detallado de ciclos anteriores, realizar los cambios pertinentes y añadir nuevas funcionalidades si es el caso - Detallar atributos y métodos para las clases de diseño, clases de interfaz, manejo de datos y otras que formen 	Productos de implementación del ciclo anterior Productos de Diseño del ciclo actual	DLD (Diseño de Nivel Detallado)	Ingeniero de Desarrollo

		<p>parte del diseño</p> <ul style="list-style-type: none"> - Detallar diagramas de interacción con flujo normal y excepcional definiendo tipos de parámetros - Definir los diagramas de componentes y paquetes- Definir los tipos de los campos, restricciones y relaciones de la Base de Datos 			
Revisar DLD	Realizar una revisión individual del diseño de nivel detallado	<ul style="list-style-type: none"> - Realizar una revisión personal de cada módulo, objeto o clase. Se puede hacer uso de una lista de verificación- Verificar los ciclos, condiciones y caminos lógicos, valores nominales y de limite de variables y parámetros. Se puede realizar un análisis de estados de maquina o usar tablas de seguimiento - Registrar tiempos y defectos encontrados en las formas LOGD y LOGT 	DLD (Diseño de Nivel Detallado)	Forma LOGD Forma LOGT	Ingeniero de Desarrollo
Actualizar Plan Pruebas Unitarias	Definir un plan para verificar el buen funcionamiento de cada componente	<ul style="list-style-type: none"> -Hacer los cambios necesarios al plan de pruebas del sistema del ciclo anterior y añadir las nuevas funcionalidades de ser necesario - Analizar la forma de probar cada atributo y método de clase - Planear una prueba de funcionamiento de la clase 	DLD (Diseño de Nivel Detallado)	PPU (Plan de Pruebas Unitarias)	Ingeniero de Desarrollo

codificar	Trasladar el diseño de nivel detallado a un lenguaje de programación orientado a objetos	<ul style="list-style-type: none"> - Ingresar el código de acuerdo al esquema del diseño detallado siguiendo el estándar de codificación - Compilar y componer los defectos encontrados hasta eliminarlos por completo - Actualizar las formas LOGD y LOGT 	DLD (Diseño de Nivel Detallado) Estándar de Codificación	Código Fuente Forma LOGD Forma LOGT	Ingeniero de Desarrollo
Realizar Pruebas Unitarias y Registrar Código	Probar eficientemente el código para cada módulo	<ul style="list-style-type: none"> - Ejecutar el plan de pruebas unitarias creado en el diseño detallado - Si es el caso, eliminar defectos encontrados en la codificación 	PPU (Plan de Pruebas Unitarias) Código Fuente	Código Fuente corregido	Ingeniero de Desarrollo
Integrar documentación	Generar el Documento de Implementación	<ul style="list-style-type: none"> - Integrar el Documento de Diseño de Nivel Detallado (DLD) <ul style="list-style-type: none"> • Diagramas de clases del diseño con atributos y su especificación • Diagramas de interacción con parámetros, atributos y su especificación • Diagrama de componentes detallado • Diagrama de instalación • Base de Datos normalizada con tipos y restricciones - Integrar Plan de Pruebas unitarias - Integrar el Código de todas las clases 		DLD (Diseño de Nivel Detallado) PPU (Plan de Pruebas Unitarias) Código	Administrador de Desarrollo

Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	<p>Lista de Verificación</p> <p>Estándar de Documentación</p> <p>Estándar de Diseño</p> <p>Estándar de Codificación</p> <p>DLD</p> <p>PPU</p> <p>Código Fuente</p>	<p>Productos de Implementación</p> <p>DLD</p> <p>PPU</p> <p>Código Fuente</p> <p>CCR</p>	<p>Administrador de Calidad / Gerente o Instructor</p>
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	<p>Productos de Implementación</p>		<p>Líder del Equipo</p>
Guardar como línea Base	Resguardar los documentos de implementación	<ul style="list-style-type: none"> - Asignar una versión al documento de implementación - Guardarlo en Línea Base - Registrar datos en la forma CCR 	<p>Productos de Implementación</p>	<p>Forma CCR</p>	<p>Administrador de Configuración</p>

2.4.7 Pruebas del Ciclo n de TSPi

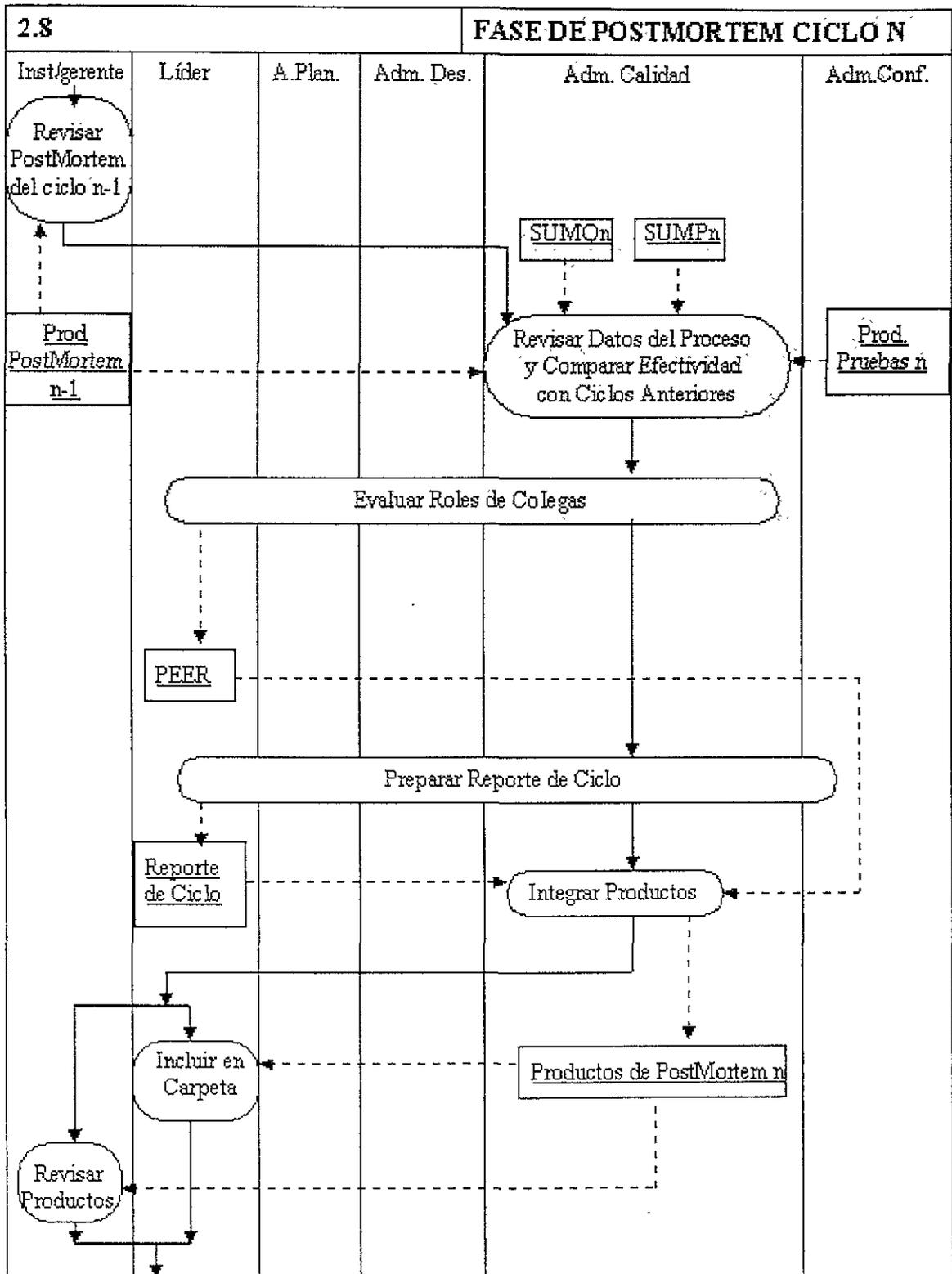


Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Realizar Pruebas del ciclo n-1	Comparar resultados obtenidos en la fase de pruebas del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de pruebas del ciclo anterior - Acordar formas de mejorar el proceso de pruebas actual 	Productos de pruebas de ciclos anteriores Productos de Implementación de ciclo actual		Instructor / Gerente
Realizar Pruebas de integración de software	Definir las tareas necesarias para realizar las pruebas del producto de software	<ul style="list-style-type: none"> - Definir todas las tareas para realizar la construcción del producto de software - Desarrollar los procedimientos para la prueba de integración - Desarrollar los procedimientos para la prueba del sistema - Registrar los tiempos y tamaño de las pruebas - Revisar los datos obtenidos y verificar la corrección de defectos 	Productos de Implementación del ciclo actual		Administrador de Desarrollo
Construir producto	Verificar que el producto de software este completo en todas sus partes	<ul style="list-style-type: none"> - Verificar que todas las partes del producto de software estén terminadas y de acuerdo a lo planeado - Construir el producto de software y prepararlo para la prueba de integración 	Productos de Implementación		Administrador de Desarrollo Participantes Equipo de Trabajo

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Realizar documentación de Usuario	Producir la información básica necesaria para que el usuario explote el sistema	<ul style="list-style-type: none"> - Añadir al manual del usuario las mejoras o nuevas funciones realizadas en el ciclo actual - Preparar la estructura general del manual de acuerdo a las necesidades del usuario - Incluir en el manual las secciones de : <ul style="list-style-type: none"> • Cómo iniciar • Explicación de qué es lo que hace el sistema • Glosario de términos • Mensajes de error y procedimientos de solución • Lista detallada del contenido 	SRS PPS	Documentación del Usuario Productos de Pruebas de ciclos anteriores	Administrador de Calidad
Organizar productos	Recopilar, organizar y revisar todos los documentos	<ul style="list-style-type: none"> - Reunir cada una de las formas y artefactos resultantes de la realización de las pruebas - Revisar la organización y cumplimiento de estándares del manual de usuario y los artefactos de pruebas 	Lista de Verificación Estándar de Documentación del Usuario Sistema Probado Formas LOGD Formas LOGTEST	Productos de Pruebas Documentación del Usuario Sistema Probado Formas LOGD Formas LOGTEST	Administrador de Calidad

Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Desarrollar productos	Revisar el cumplimiento de estándares y la consistencia entre el sistema y la documentación del usuario	<ul style="list-style-type: none"> - Realizar la ejecución del sistema verificando que el contenido en el manual sea fiel a las pantallas mostradas por el sistema - Realizar las correcciones correspondientes de ser necesario 	Documentación del Usuario Sistema Probado PPS	Productos de Pruebas Documentación del Usuario Sistema Probado Formas LOGD Formas LOGTEST	Administrador de Calidad / Gerente o Instructor
Organizar en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	- Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto	Productos de Pruebas Sistema Probado Documentación de Usuario Formas LOGTEST Formas LOGD		Líder del Equipo
Guardar como Base	Resguardar los documentos de pruebas	<ul style="list-style-type: none"> - Asignar una versión al documento de pruebas - Guardarlo en Línea Base - Registrar datos en la forma CCR 	Productos de Pruebas	Forma CCR	Administrador de Configuración

2.4.8 PostMortem del Ciclo n de TSPi



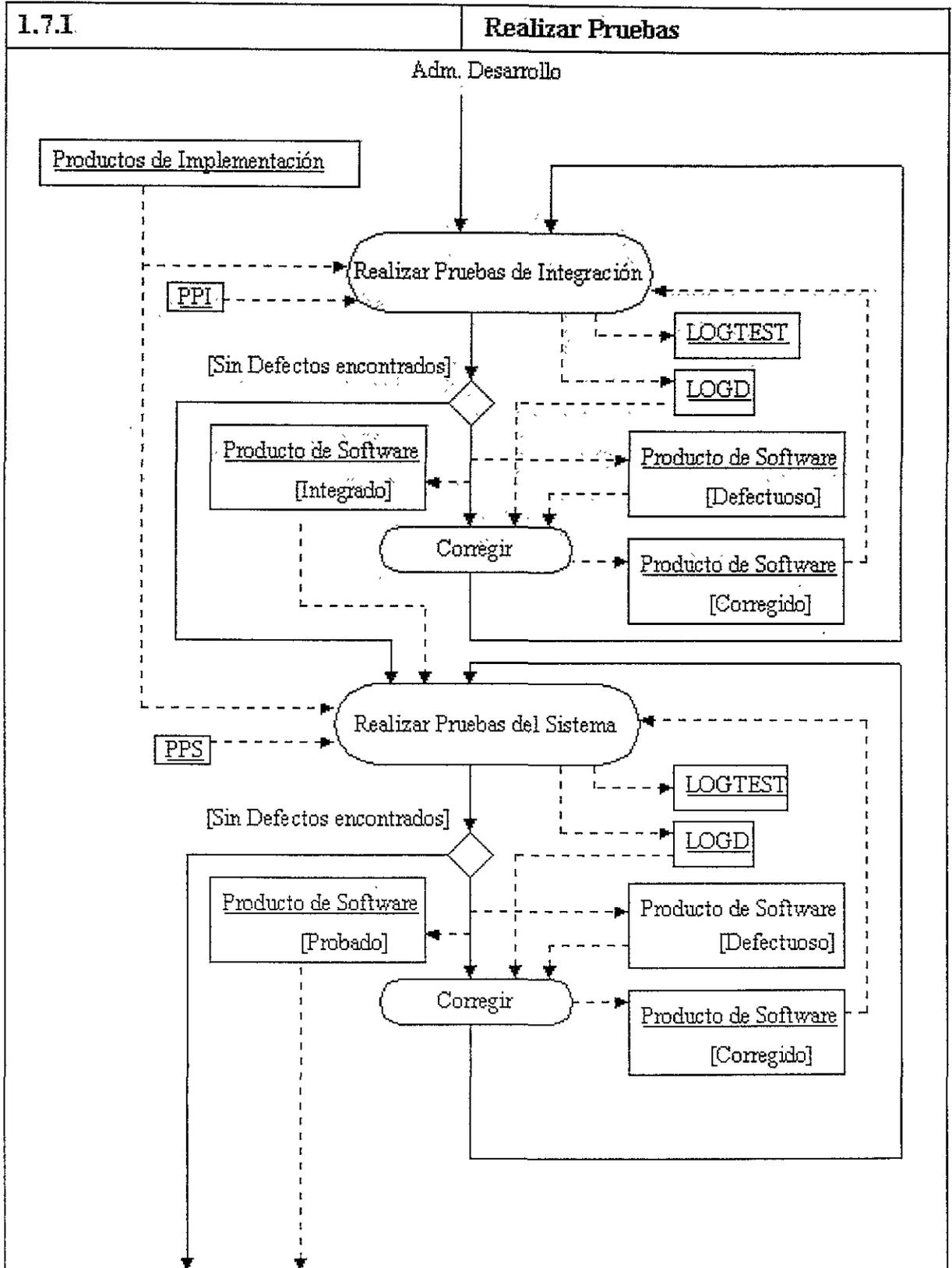
Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Revisar PostMortem del Ciclo n-1	Comparar resultados obtenidos en la fase de postmortem del ciclo anterior y mejorar el proceso en el ciclo actual	<ul style="list-style-type: none"> - Discutir los resultados obtenidos en la fase de postmortem del ciclo anterior - Acordar formas de mejorar el proceso de postmortem actual 	Productos de postmortem de ciclos anteriores		Instructor / Gerente
Revisar datos del proceso y comparar efectividad con Ciclos anteriores	Comparar los datos obtenidos con los datos planeados y evaluar la calidad del producto realizando una comparación con resultados de otros ciclos anteriores de desarrollo	<ul style="list-style-type: none"> - Evaluar los siguientes aspectos <ul style="list-style-type: none"> • Desempeño actual comparado con el plan • Lecciones aprendidas en el ciclo actual • Mejoras que pueden realizarse • Dónde se localizaron problemas y registrarlos para corregirlos en el siguiente ciclo • De acuerdo al ciclo actual verificar que errores han sido recurrentes y donde las mejoras han mostrado su eficacia 	Productos de PostMortem de ciclos anteriores Productos de Pruebas Forma SUMP Forma SUMQ		Administrador de Calidad
Evaluar roles de colegas	Evaluar la efectividad en el desempeño de cada uno de los roles	<ul style="list-style-type: none"> - Cada miembro del equipo evalúa el desempeño de cada uno de los roles - Registrar para cada rol: <ul style="list-style-type: none"> • Qué funcionó en el proceso • Dónde se encontraron problemas • Formas de mejorar el proceso y nuevas metas para ciclos posteriores • Indicar si las mejoras propuestas para este 		Formas PEER	Líder del Equipo Participantes Todos los miembros del equipo

		<p>ciclo fueron eficaces y de no ser así, sugerir modificaciones en el proceso</p> <ul style="list-style-type: none"> - Realizar los reportes de rol de acuerdo a la ayuda y eficacia del mismo - Registrar los datos obtenidos en la forma PEER 			
<p>Comparar el desempeño personal durante el ciclo actual de desarrollo</p>	<p>Reporte de Ciclo</p>	<p>Líder del Equipo Participantes Todos los miembros del equipo</p>	<p>- Cada miembro del equipo evalúa su propio desempeño</p> <p>- Comparar el desempeño del ciclo actual con ciclos anteriores</p> <p>- Revisar qué funciono durante el ciclo actual y que puede mejorar en ciclos posteriores</p> <p>- Indicar si las mejoras sugeridas fueron eficaces</p> <p>- Evaluar el desempeño del equipo de acuerdo a la perspectiva del rol asignado y de ingeniero de desarrollo</p> <p>- Justificar las conclusiones obtenidas con los datos registrados</p> <p>- El contenido del reporte es :</p> <ul style="list-style-type: none"> • Tabla de contenido • Sumario • Reporte de Roles • Líder del Equipo • Administrador de Desarrollo • Administrador de Planeación • Administrador de Calidad • Administrador de Configuración • Reporte de Ingenieros de Desarrollo 		

Entregar productos	Realizar la integración y revisión de productos de PostMortem	<ul style="list-style-type: none"> - Integrar el reporte de ciclo de desarrollo actual - Revisar conforme a estándares el reporte y formas PEER - Registrar defectos encontrados en formas LOGD - Entregar documentación con defectos a propietario para corrección 	Formas PEER Reporte de Ciclo	Productos de PostMortem	Administrador de Calidad
Revisar productos	Revisar el cumplimiento de estándares, detectar y corregir defectos en los productos	<ul style="list-style-type: none"> - Imprimir los productos a revisar - Tomar un inciso de la lista de verificación y dedicarse a su revisión en todo el producto - Resaltar con marcador los defectos encontrados - Realizar el proceso anterior hasta el final de la lista - Entregar al propietario del producto para realizar la corrección 	Lista de Verificación Estándar de Documentación Formas PEER Reporte del Ciclo	Productos de PostMortem Formas PEER Reporte del Ciclo	Administrador de Calidad / Gerente o Instructor
Incluir en carpeta	Contar con una referencia sencilla y práctica de toda la documentación producida en el proyecto. La carpeta puede llevarse en una versión electrónica y/o bibliográfica	<ul style="list-style-type: none"> - Incluir cada producto o subproducto en el espacio correspondiente dentro de la carpeta de proyecto 	Productos de PostMortem Formas PEER Reporte de Ciclo		Líder del Equipo

2.5 Realizar Pruebas

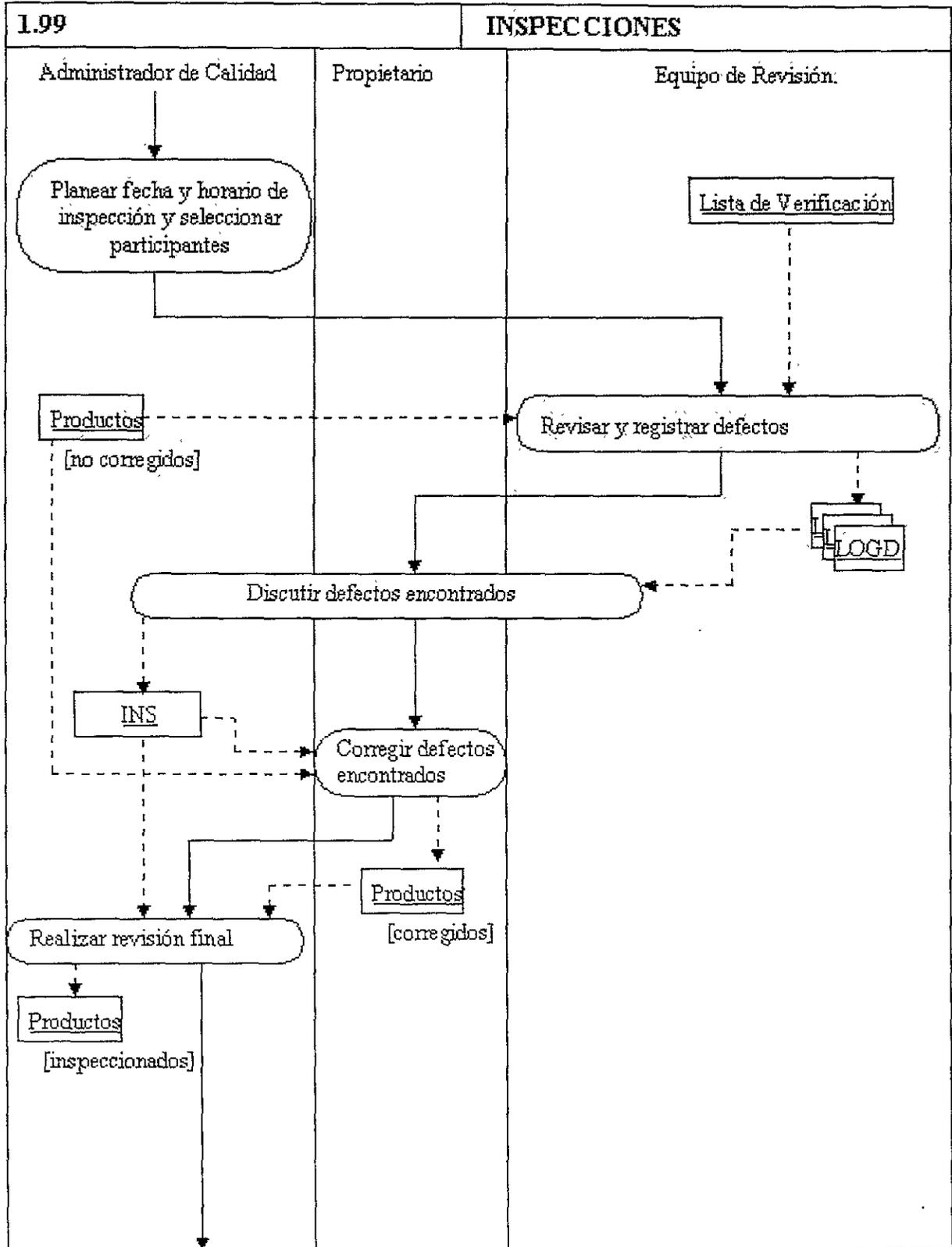
En esta sección se encuentra las actividades correspondientes al diagrama anidado de pruebas. El objetivo de este diagrama es detallar un poco más algunas actividades de pruebas.



Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Realizar pruebas de integración	Realizar las tareas de integración del sistema	<ul style="list-style-type: none"> - Ejecutar el plan de pruebas de integración verificando el buen funcionamiento de las dependencias de clases - Registrar las actividades realizadas en la forma LOGTEST - Registrar los defectos encontrados en la forma LOGD 	Productos de Implementación PPI Software Corregido	Forma LOGD Forma LOGT Producto de Software Integrado sin defectos o Producto de Software Defectuoso	Administrador de Desarrollo
Realizar pruebas del sistema	Probar el producto de software desarrollado en su totalidad	<ul style="list-style-type: none"> -Ejecutar el plan de pruebas del sistema en condiciones normales y excepcionales - Probar la instalación del producto - Registrar las actividades realizadas en la forma LOGTEST - Registrar todos los defectos encontrados en la forma LOGD 	PPS Productos de Implementación Producto de Software Integrado o Software Corregido	Forma LOGTEST Forma LOGD Producto de Software Probado sin defectos o Producto de Software Defectuoso	Administrador de Desarrollo
Registrar	Eliminar defectos encontrados en el producto de software	<ul style="list-style-type: none"> - El propietario del producto depura cada defecto encontrado y registrado en las formas LOGTEST - Compilar el programa y verificar su buen funcionamiento - Entregar la parte correspondiente para realizar una nueva prueba 	Productos de Implementación Producto de Software Defectuoso	Producto de Software Corregido	Administrador de Desarrollo

2.6 Inspecciones

En esta sección se presentan las actividades correspondientes al diagrama de inspecciones. El objetivo al realizar esta sección es mostrar mas a detalle los pasos a seguir para realizar una correcta inspección de software.



Actividad	Propósito	Pasos	Artefactos de Entrada	Artefactos de Salida	Rol Responsable
Definir fecha y horario de reunión y seleccionar participantes	Organizar la revisión del producto y seleccionar inspectores	<ul style="list-style-type: none"> - Seleccionar miembros del equipo que revisaran el producto - Entregar una copia del producto y lista de verificación a cada inspector - Acordar la fecha y hora de reunión 			Administrador de Calidad
Revisar y registrar productos	Realizar una revisión efectiva del producto	<ul style="list-style-type: none"> - Cada inspector revisa el producto en forma individual de acuerdo a la Lista de Verificación - Marca en la impresión del producto los defectos encontrados - Registra cada defecto localizado en la forma LOGD - Registra el tiempo empleado en la revisión 	Producto Impreso Lista de Verificación		Equipo de Revisión
Registrar productos encontrados	Analizar cada defecto encontrado por el equipo de revisión	<ul style="list-style-type: none"> - Registrar de manera secuencial cada defecto encontrado por cada uno de los miembros del equipo de revisión - Analizar con el propietario del producto si el defecto se considera como mayor o menor - Registrar en la Forma INS cada defecto encontrado - La reunión es moderada por el Administrador de Calidad 	Producto a inspeccionar Formas LOGD	Forma INS	Administrador de Calidad Inspectores
Registrar productos encontrados	Depurar el producto de acuerdo a los defectos localizados	<ul style="list-style-type: none"> - Corregir cada defecto registrado en la Forma INS - Entregar Producto depurado a Administrador de Calidad 	Producto Inspeccionado Forma INS	Producto Corregido	Propietario del Producto
Revisión final	Verificar que todos los defectos encontrados fueron corregidos	<ul style="list-style-type: none"> - Realizar una revisión final del producto de acuerdo a los defectos registrados en la Forma INS 	Producto Corregido Forma INS	Producto Inspeccionado	Administrador de Calidad

Capítulo 3

Conclusiones

En este capítulo se muestran los resultados obtenidos de acuerdo al objetivo de la presente guía. Se presentan los primeros datos de encuestas realizadas sobre el aporte de la guía dentro de la enseñanza de TSPi. Adicionalmente se sugieren posibles trabajos a futuro relacionados con el presente trabajo.

La presente tesis cubrió el alcance esperado. Se generaron los diagramas de actividades para la estructura general, primer ciclo, ciclo n y para cada una de las fases correspondientes a dichos ciclos. Se generaron las tablas que describen y muestran los pasos a seguir para cada una de las actividades mostradas en los diagramas. Además se incluyeron ejemplos de productos que fueron tomados de un proyecto real.

Este trabajo contiene los aspectos fundamentales del libro de Watts Humphrey. Cada diagrama, actividad y ejemplo fueron diseñados con base en la experiencia obtenida al desarrollar un proyecto bajo el proceso de TSPi.

La mayor dificultad radicó en la organización de todo el material obtenido y en el desglose de cada actividad en una serie de pasos que permitieran ser comprendidos y seguidos con facilidad.

Todo el trabajo realizado que se muestra en la presente tesis está disponible en español en el sitio de Internet <http://kasia.fciencias.unam.mx/TSPi>. Los ejemplos de formatos que no fueron incluidos en el presente trabajo pueden ser consultados en dicho sitio. La facilidad de contar con una referencia en Internet permite ampliar la difusión de TSPi sobre todo entre la comunidad hispano parlante, en general pretende ser un punto de referencia que permita ampliar la calidad de los productos y procesos relacionados con el desarrollo de software.

Actualmente, la guía está siendo utilizada en su versión electrónica como material de apoyo en los curso Ingeniería de Software semestre 2001-I e Ingeniería de Software Orientada a Objetos semestre 2002-I, que forman parte del plan de estudios de la licenciatura de Ciencias de la Computación de la Facultad de Ciencias y de la Maestría en Ciencia e Ingeniería de la Computación de la U.N.A.M. respectivamente.

Para conocer el aporte que la guía tuvo en la comprensión de TSPi, durante el curso de Ingeniería de Software se desarrolló un cuestionario, en el cual se pedía al alumno que indicara en porcentaje cual había sido el aporte que en su proceso de aprendizaje tuvo la clase, el libro de TSPi y la guía en su versión electrónica. El cuestionario se aplicó a 23 alumnos y los resultados obtenidos son los siguientes:

Alumno	% Clase	% Libro	% Guía
1	40	30	30
2	60	20	20
3	60	20	20
4	60	25	15
5	30	30	40
6	50	15	35
7	50	20	30
8	10	30	60
9	40	40	20

Alumno	% Clase	% Libro	% Guía
10	10	15	75
11	20	20	60
12	20	20	60
13	20	20	60
14	40	20	40
15	30	30	40
16	30	30	40
17	30	35	35
18	50	30	20
19	60	10	30
20	40	30	30
21	50	20	30
22	30	30	40
23	10	30	60
PROMEDIO	36.52	24.78	38.7

Estos son los primeros datos obtenidos del uso de la guía. Como se muestra en la tabla, los promedios finales de aporte de clase y de la guía son muy similares mientras que el uso del libro resultó ser un poco menor. De los 23 alumnos, 12 encontraron en la guía una de las alternativas que más consultaron. Seis alumnos dieron un porcentaje mayor al 50% al uso de la guía. Esto nos indica que la guía de TSPi usando UML es un material de apoyo que puede ser muy útil en el aprendizaje del proceso de Team Software Process. En un principio se diseñó para ser utilizada por personal académico y estudiantes de carreras afines a la computación, sin embargo, la guía no excluye a cualquier desarrollador que quiera conocer procesos de desarrollo de software prácticos y eficientes.

En ningún momento se pretende que la guía desplace el conocimiento que el libro "Introduction to the Team Software Process" de Watts S. Humphrey promueve, sino únicamente servir de apoyo visual y práctico.

Cada uno de los ejemplos mostrados son una referencia de cómo podría realizarse el trabajo. No son normas estrictas que deban ser seguidas.

El trabajo a futuro puede ser enfocado en varios sentidos:

- Ampliar y enriquecer el presente trabajo detallando actividades correspondientes a las fases de diseño, implementación y pruebas.
- Desarrollo de una herramienta de apoyo a la documentación y registro de datos en las formas de TSPi, diseñada para un trabajo en equipo.

- Representarse mediante diagramas de UML diferentes procesos y técnicas como PSP, SPICE, CMM entre otras que son importantes en el desarrollo de sistemas.
- Finalmente puede probarse la eficacia de la guía en comparación con el uso exclusivo del libro de TSPi en algún curso de Ingeniería de Software.

La presente tesis me ha brindado la posibilidad de aprender un excelente proceso de software como lo es TSPi. El esfuerzo que realice es de gran valor para animar a los futuros ingenieros de sistemas a crear productos de software de alta calidad trabajando como miembros activos dentro de un equipo de trabajo.

Bibliografía

[Alquicira, 2000]

Alquicira Esquivel Claudia, "Modelado de áreas clave de procesos para CMM nivel 2", tesis de maestría en ciencia de la computación, UNAM, 2000.

[CMM, 1995]

"The Capability Maturity Model, Guidelines Improving for the Software Process", SEI Series in Software Engineering, Addison Wesley 1995.

[Humphrey, 1997]

Watts S. Humphrey, "Introduction to the Personal Software Process", SEI Series in Software Engineering, Addison Wesley, 1997.

[Humphrey, 2000]

Watts S. Humphrey, "Introduction to the Team Software Process", SEI Series in Software Engineering, Addison Wesley, 2000.

[Fowler, 1999]

Fowler Martin, Scott Kendall, "UML Gota a gota", Addison Wesley, 1999.

[ISO/IEC 12207, 1995]

International Standard ISO/IEC 12207:1995 "Information technology-Software life cycle processes".

[ISO/IEC 15504, 1999]

ISO/IEC 15504 "Software Process Assessment", Technical report, 1999.

[UML, 1999]

Booch Grady, Rumbaugh James and Jacobson Ivan, "The Unified Modeling Language User Guide", Addison Wesley, 1999.

Página de Internet de la Guía de TSPi usando UML

<http://kasia.fciencias.unam.mx/TSPi>

Anexo A

Diagramas de Actividades

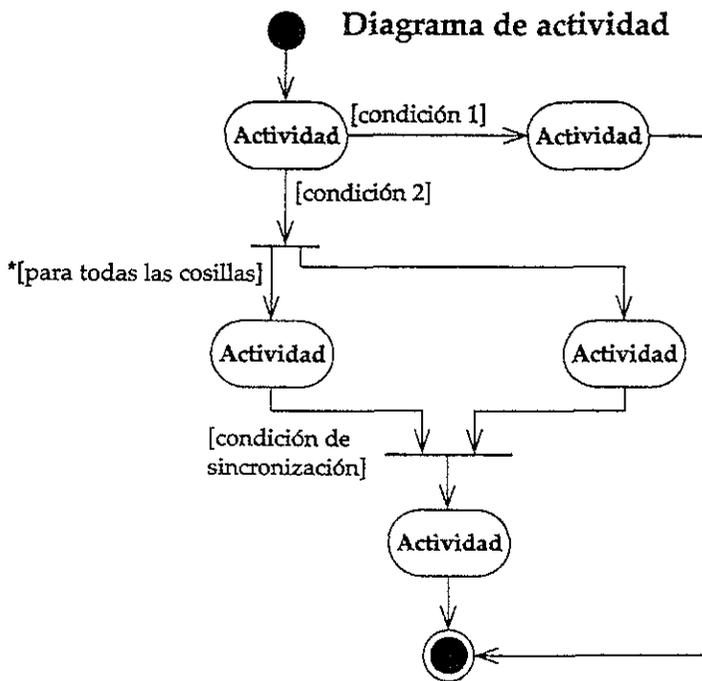
El contenido de esta sección fue tomado del libro “The Unified Modeling Language User Guide” [UML,1999]. Muestra en síntesis los diagramas de actividades, su estructura y componentes.

Un diagrama de actividades es un diagrama que muestra el flujo de control entre actividades a lo largo del tiempo, puede manejar procesos paralelos. Son un caso especial de un diagrama de estados en el cual la mayoría de sus estados son acciones o subactividades y la mayoría de sus transiciones son disparadas al terminarse dicha acción o subactividad.

Con un diagrama de actividades podemos seleccionar el orden en el que se harán las cosas, tenemos una secuenciación. Son de gran utilidad para visualizar, especificar, construir y documentar la dinámica de una sociedad de objetos, o pueden emplearse para modelar el flujo de control de una operación.

Gráficamente un diagrama de actividades es una colección de nodos y arcos.

La estructura general de un diagrama de actividades se muestra en la siguiente figura.

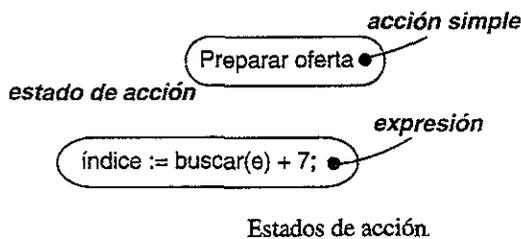


TERMINOS Y CONCEPTOS

ESTADO DE ACCIÓN

Un estado de acción es una ejecución atómica que representa la realización de una acción, por ejemplo evaluar una expresión que fije el valor de un atributo o que regrese un valor, enviar una señal, aplicar una operación sobre un objeto, crearlo o destruirlo. No se pueden descomponer, ni se interrumpe su ejecución y se considera que lleva un tiempo insignificante de ejecución.

Una acción esta compuesta de computaciones atómicas ejecutables que producen un cambio de estado en el sistema o la devolución de un valor.

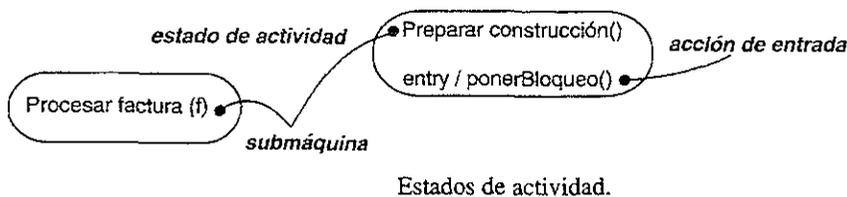


En la figura anterior la acción de preparar oferta es un estado de acción, de la misma manera podemos representar expresiones como asignarle un valor a la variable índice.

ESTADO DE ACTIVIDAD

Los estados de actividad si pueden descomponerse mas, no son atómicos es decir que pueden ser interrumpidos, pueden tener acciones de entrada y salida y especificaciones de submáquinas.

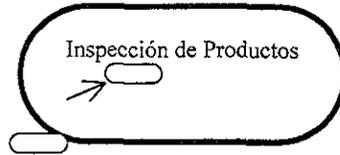
Una actividad es una ejecución no atómica en curso dentro de una máquina de estados.



En la figura anterior observamos que un estado de actividad puede tener acciones de entrada o salida, para el estado de actividad Preparar construcción() se necesita la entrada ponerBloqueo(). Pueden incluirse tambien especificaciones de submaquina.

ESTADO DE SUBACTIVIDAD

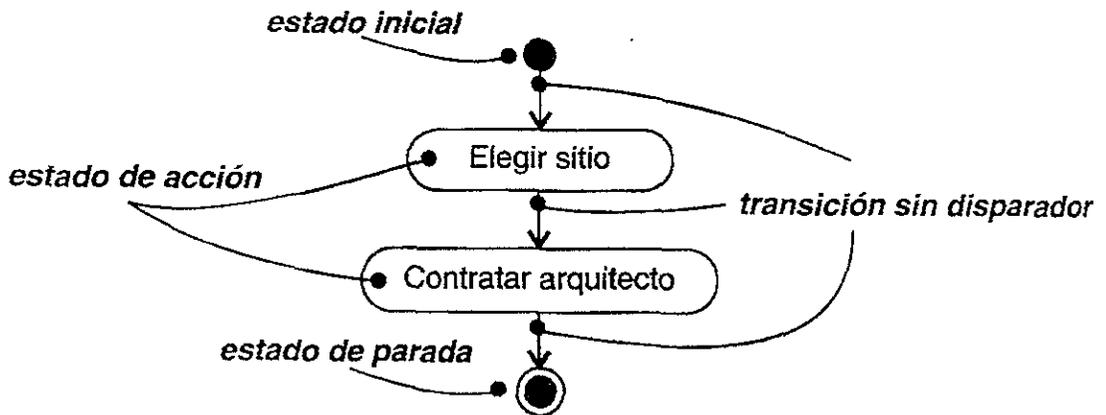
Una subactividad involucra a un diagrama de actividad. Cuando se fija un estado de subactividad entonces un diagrama de actividades se ejecuta. Implica estados de actividad anidados.



En el ejemplo la subactividad Inspección de Productos implica que al llegar a esta actividad se debe realizar un diagrama de actividades correspondiente a la Inspección de Productos, una vez realizadas todas las actividades de dicho diagrama se continua el flujo en la actividad siguiente a la subactividad.

TRANSICIONES

Muestran el camino a seguir de un estado de actividad o acción al siguiente. Se llaman transiciones sin disparadores o de terminación porque el control pasa una vez que se ha finalizado la tarea.

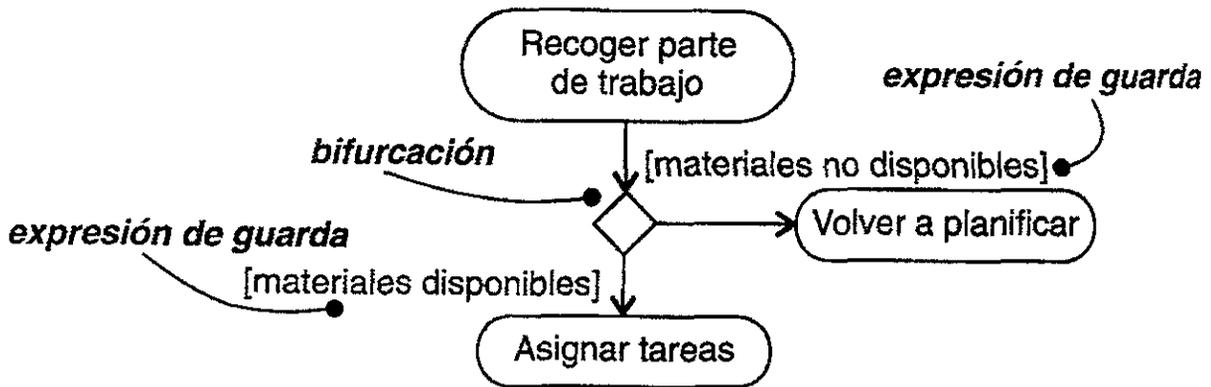


Transiciones sin disparadores.

En la figura podemos apreciar que el flujo de control comienza en un **estado inicial** que se representa con un círculo relleno. Se efectúa la actividad Elegir sitio, una vez finalizada seguimos la transición hasta Contratar arquitecto, una vez que se desarrolló la actividad y siguiendo el camino mostrado por la transición llegamos a un estado de parada o **estado final**, que se representa por medio de un círculo relleno dentro de una circunferencia.

BIFURCACION

Especifica caminos alternativos elegidos según el valor de una expresión booleana que se evalúa solo una vez al entrar a ésta. Una bifurcación puede tener una transición de entrada y dos o más de salida.

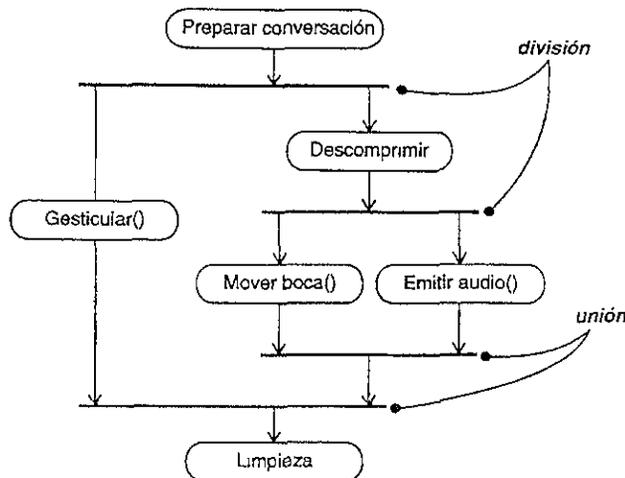


Bifurcación.

DIVISION Y UNION

La división representa la separación de un flujo de control sencillo en dos o más flujos de control concurrentes.

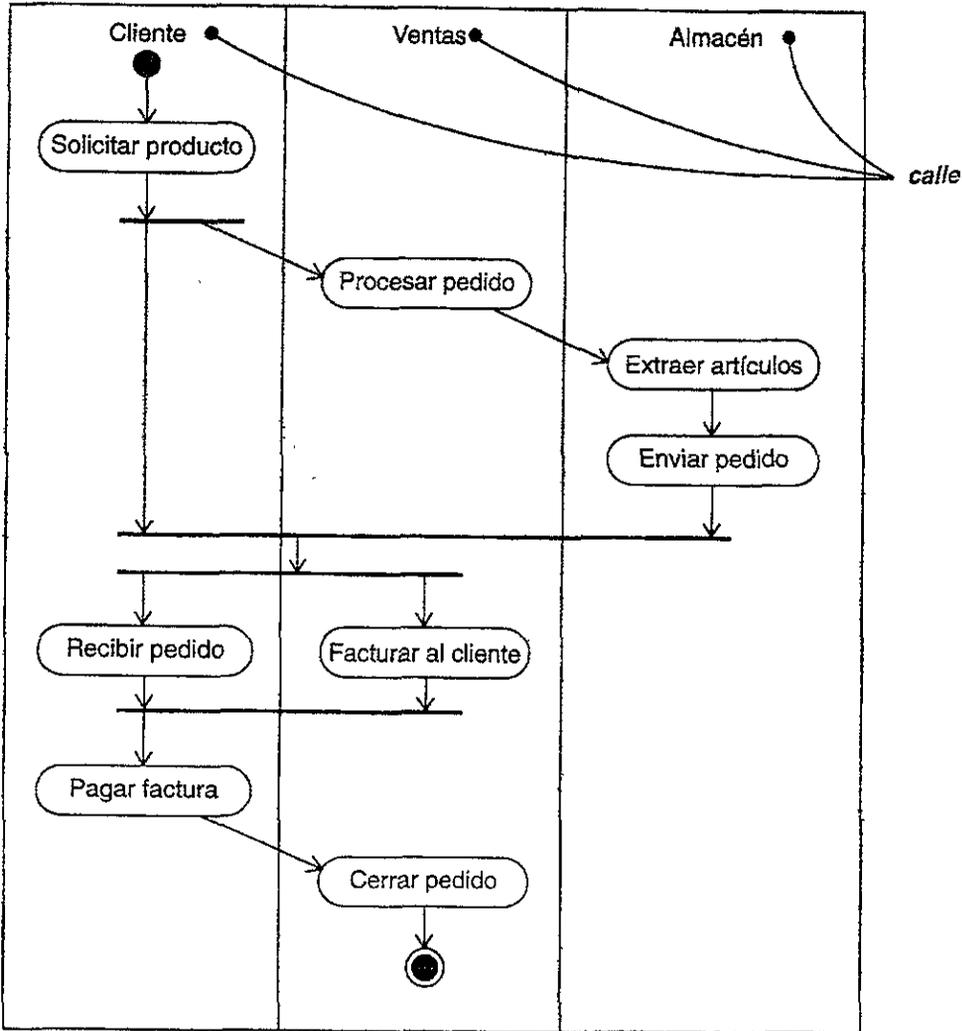
La unión representa la sincronización de dos o más flujos de control concurrentes. En la unión los flujos concurrentes se sincronizan y continúan en un solo flujo de control.



División y unión.

CALLES O CARRILES (Swimlanes)

Permiten dividir los estados de un diagrama de actividad en grupos donde cada uno representa la parte de la organización responsable de esas actividades o subactividades y puede eventualmente ser implementada por uno o más objetos.

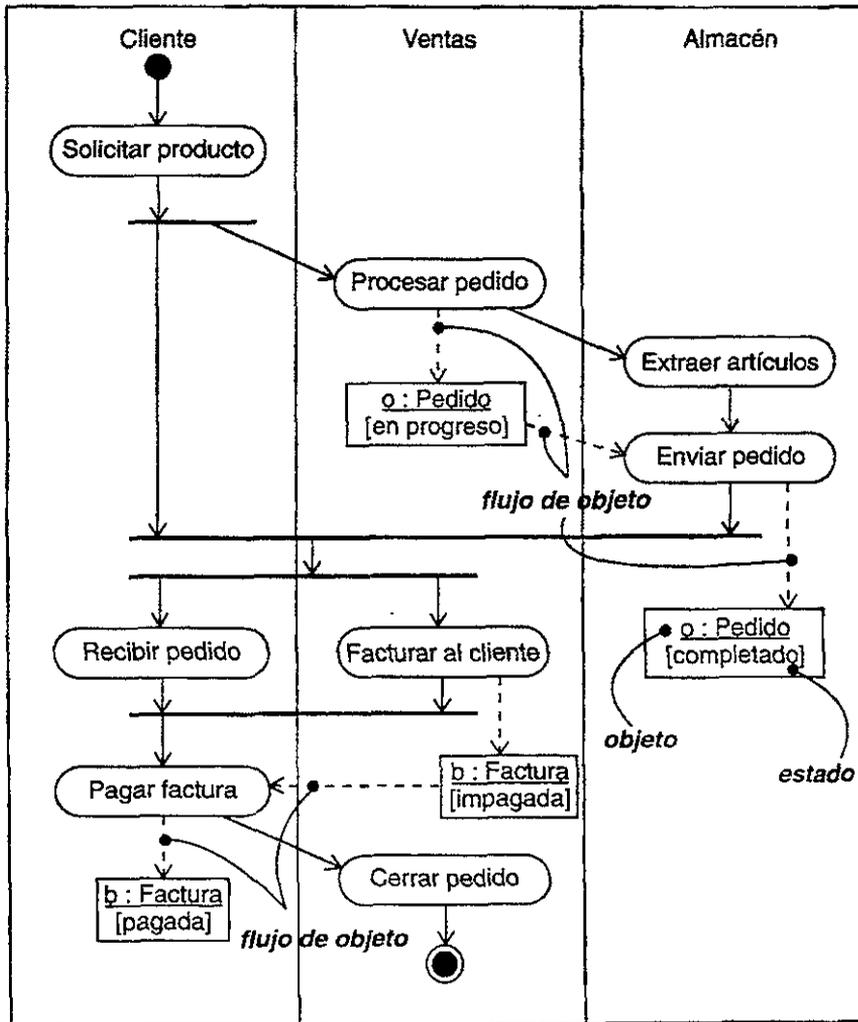


Calles

La figura muestra como la responsabilidad por llevar a cabo las actividades Solicitar producto, Recibir pedido y Pagar factura es del Cliente pues se encuentran dentro de su carril. De manera similar Ventas y Almacén tienen la responsabilidad sobre las actividades asignadas dentro de su carril.

FLUJO DE OBJETOS

En el flujo de control de un diagrama de actividades puede haber involucrados objetos y estos se pueden especificar dentro del diagrama conectados con una dependencia a la actividad o transición que los crea, modifica o destruye.



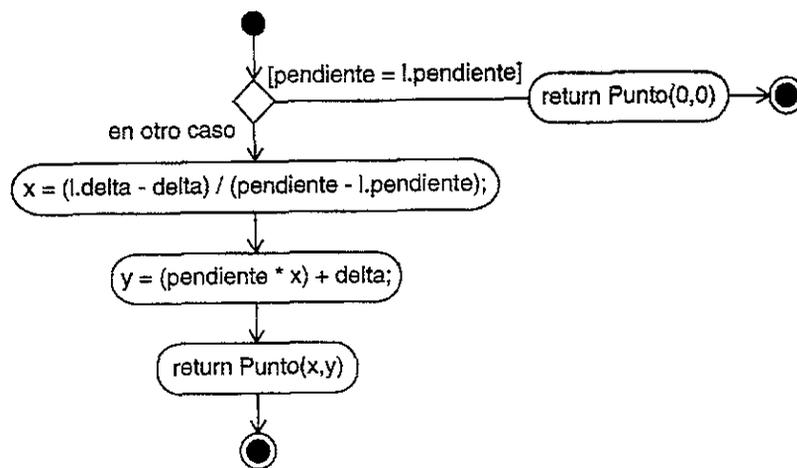
Flujo de objetos

En la figura anterior se refleja la forma en que se lleva a cabo el flujo de objetos. La actividad Procesar pedido tiene como salida un objeto o:Pedido con estado [en progreso], a su vez este objeto es entrada para la actividad Enviar pedido, podemos notar que la salida de esta actividad es el mismo objeto pero con un estado de [completado].

Técnicas para modelar un flujo de trabajo:

- Establecer un centro de interés para el flujo de trabajo.
- Seleccionar los objetos del negocio que tienen las responsabilidades de mas alto nivel en cada parte del trabajo global.
- Identificar las precondiciones del estado de inicio de flujo de trabajo y las poscondiciones del estado final.
- Especificar las acciones y actividades que tienen lugar a lo largo del tiempo y representarlas como estados de actividad o acción.
- Llevar las acciones complicadas o repetidas a estados de actividad y asignar un diagrama de actividades que las expanda.
- Representar las transiciones que conectan los estados de acción y de actividad (flujos secuenciales, bifurcaciones uniones y divisiones).
- Si el flujo involucra objetos importantes hay que representarlos también en el diagrama de actividad, representando su valor y su estado.

2. *Modelar una operación:* Aquí se utilizan como diagramas de flujo para modelar los detalles de la computación.



Modelado de una operación.

Técnicas para modelado de una operación

- Reunir las abstracciones implicadas en la operación incluyendo parámetros de la operación, atributos de clase.
- Identificar las precondiciones en el estado inicial de la operación y las poscondiciones en el estado final y además identificar las invariantes.
- Especificar las actividades y acciones que tienen lugar durante la ejecución y representarlas como estados de actividad y acción.
- Usar bifurcaciones si es necesario especificar caminos alternativos o iteraciones.
- Usar divisiones y uniones cuando sea necesario especificar flujos paralelos de control.

SUGERENCIAS Y CONSEJOS

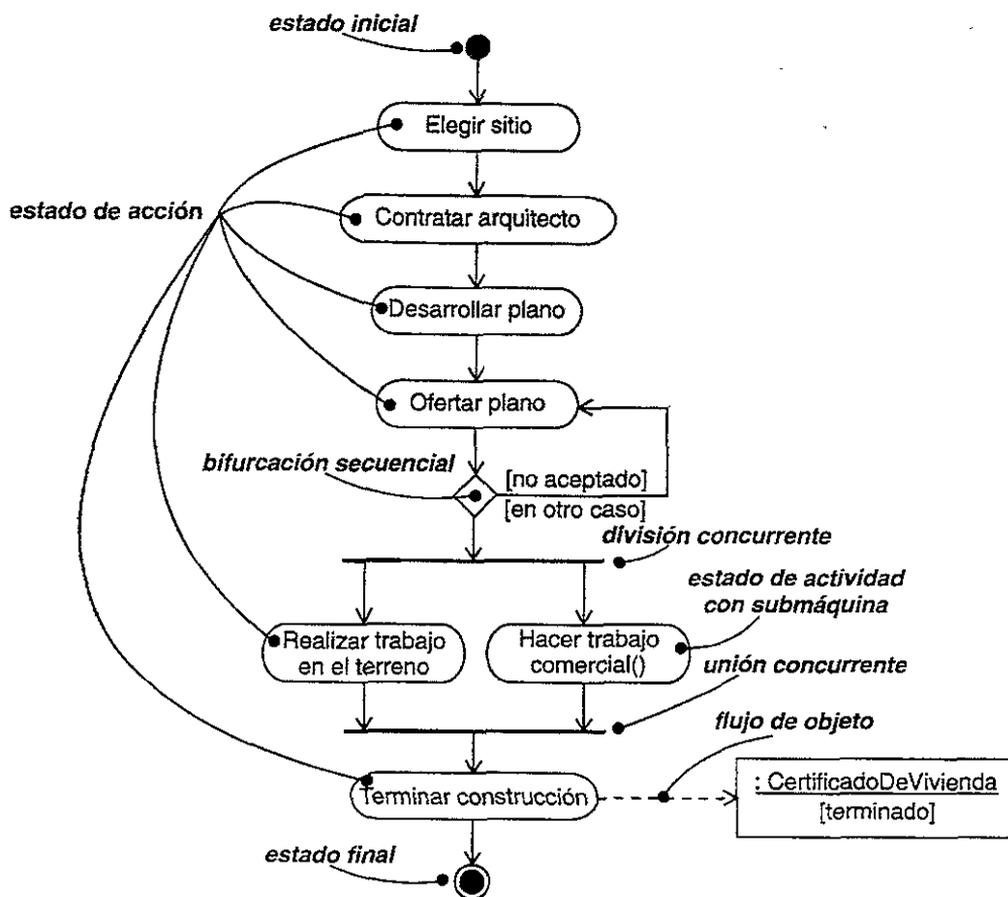
Un diagrama de actividades bien estructurado debe:

- Modelar la dinámica de un sistema.
- Contener solo los elementos esenciales para comprender ese entorno.
- Proporcionar detalles de forma consistente con su nivel de abstracción, solo se muestran los adornos esenciales para una mayor comprensión.
- No ser tan minimalista que no ofrezca información al lector sobre los aspectos importantes de la semántica.

Cuando se dibuje un diagrama de actividades

- Darle un nombre significativo.
- Comenzar por modelar el flujo principal (las bifurcaciones, concurrencia y flujo de objetos se deben considerar como secundarios).
- Distribuir los elementos para minimizar los cruces de líneas.
- Usar notas y colores como apoyo visual para indicar aspectos importantes del diagrama.

EJEMPLOS



Diagramas de actividades.

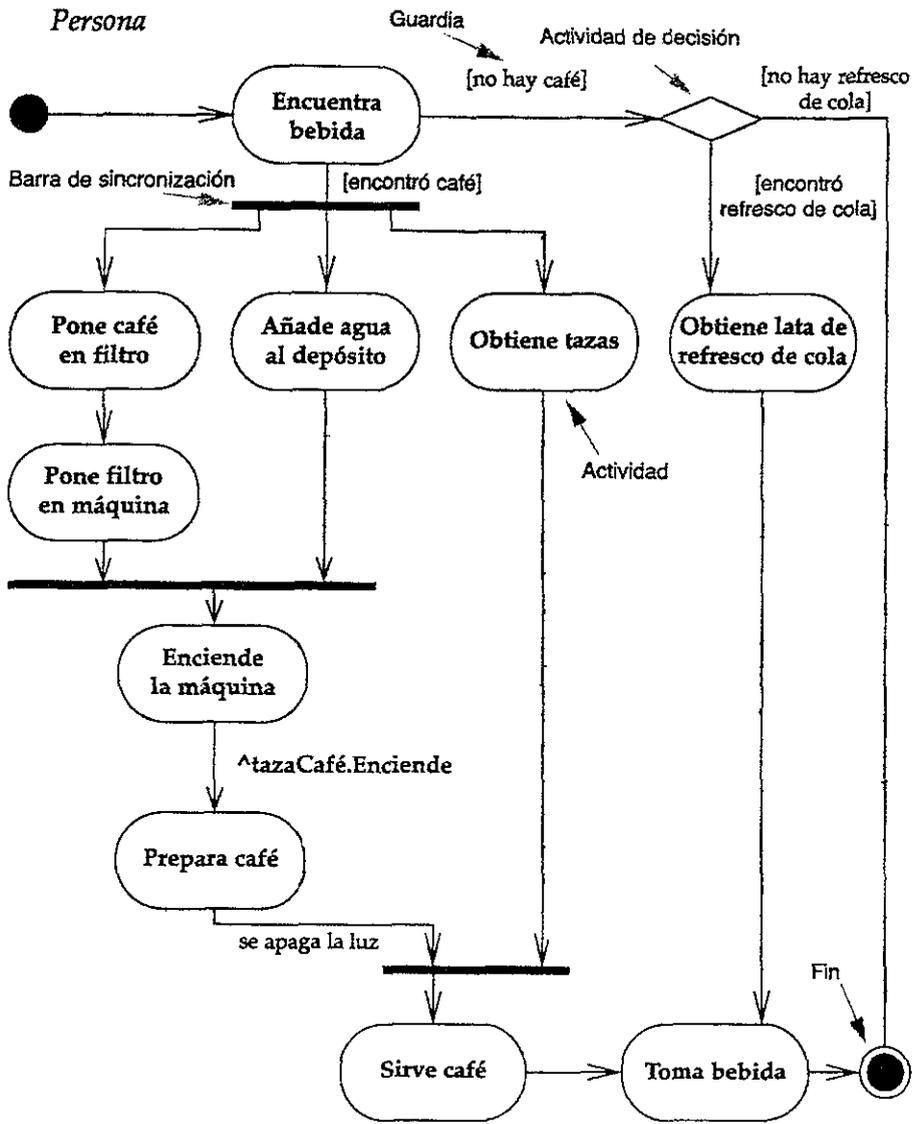
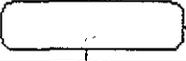
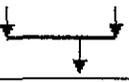
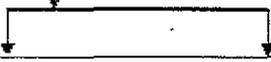
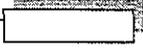
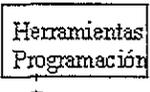


Diagrama de actividades

Anexo B

Tabla de elementos utilizados en diagramas de actividades.

	Estado inicial
	Estado final
	Bifurcación
	Transición
	Flujo de objetos.
	Actividad
	Actividad anidada
	Unión
	División.
	Documento de apoyo
	Herramienta
	Producto de trabajo.

Glosario

Muestra el significado de términos, siglas y abreviaturas principales dentro del presente trabajo.

CCR	Configuration Change Request. Solicitud de cambio a la configuración.
Ciclo de desarrollo	Proceso de desarrollo que construye productos por pasos, cada uno de los cuales produce un subconjunto del producto final.
CMM	Capability Maturity Model. Niveles de madurez de un proceso de software
Defecto	Elemento de requerimientos, diseño o implementación que si no se modifica puede causar un funcionamiento impropio en alguna de las fases siguientes.
Defecto mayor	Problema que cuando se corrige puede cambiar al programa ejecutable.
Defecto menor	Defecto que no se considera mayor
Dis	Diseño
DisN	Diseño del ciclo n
DLD	Detail Level Design. Documento de diseño detallado.
Est	Estrategia
EstN	Estrategia del ciclo n
Guión	Lista de acciones requeridas para completar un proceso o parte de un proceso.
Imp	Implementación
ImpN	Implementación del ciclo n
INS	Inspection. Forma de resumen de inspección.
Inspección	Proceso en el cual varios ingenieros revisan un producto generado por otro ingeniero, con el fin de auxiliar en la búsqueda de defectos.
ITL	Issue Tracking Log. Forma de registro de riesgos.
KLOC	Miles de líneas de código.
Lan	Lanzamiento
LanN	Lanzamiento del ciclo n
Línea Base	Colección de documentos y otros materiales que oficialmente representan el producto en algún momento de tiempo.
LOC	Líneas de Código.
LOGD	Defect Recording Log. Forma de registro de defectos.
LOGT	Time Recording Log. Forma de registro de tiempo
LOGTEST	Test Log. Forma de registro para pruebas.
MCC	Véase Mesa de Control de Cambios.
Mesa de Control de Cambios	Comité de ingenieros que se reúne para revisar y aprobar la línea base de productos y propone cambios
Obj.	Objetivo
PEER	Team and Peer Evaluation. Forma de evaluación de roles entre colegas.
Pla	Planeación
PlaN	Planeación del ciclo n

Plan de Administración de la Configuración	Este plan incluye al menos identificación del plan, procedimientos de control y lista de miembros
Plan de calidad	Rendimiento planeado y actual de calidad de cada parte y ensamble del sistema.
Plan de pruebas	Define las pruebas que serán realizadas, los resultados esperados, los materiales necesarios para realizar las pruebas y el plan para producir dichos materiales.
Pos	PostMortem
PosN	PostMortem del ciclo n
PPI	Plan de pruebas de integración
PPS	Plan de pruebas del sistema
PPU	Plan de pruebas unitarias.
Prod.	Productos
Pru	Pruebas
Prueba de integración	Prueba que verifica que el sistema esta apropiadamente construido, que todas sus partes estén presentes y que funcionen juntas.
Prueba del sistema	Prueba que verifica ampliamente y determina si el sistema funciona o no apropiadamente en los puntos principales.
Prueba unitarias	Prueba de caja blanca que verifica que la estructura del programa es correcta y que funciona apropiadamente en condiciones normales y anormales de los parámetros.
PruN	Pruebas del ciclo n
PSP	Personal Software Process. Proceso de desarrollo de software personal.
Req	Requerimientos
ReqN	Requerimientos del ciclo n
Revisión	En TSPi y PSP las revisiones son realizadas por los ingenieros para localizar defectos en cada uno de los productos que producen personalmente.
Rol	Área de responsabilidad definida para un miembro del equipo.
SCHEDULE	Forma de planeación de calendario.
SDS	Software Design Specification. Documento de especificación de diseño. Generalmente incluye diseño de alto-nivel, arquitectura, definiciones de interface entre otras especificaciones.
SEI	Software Engineering Institute. Instituto de Ingenieros de Software.
SRS	Software Requirement Specification. Documento de especificación de requerimientos. Descripción del producto a desarrollar en palabras propias de los ingenieros. Permite acordar y aclarar con el cliente sus necesidades.
STRAT	Strategy Form. Forma de registro de estrategia.
SUMP	Summary Plan. Resumen de avance de planeación.
SUMQ	Summary Quality. Resumen de datos planeados y obtenidos en calidad de las partes.

SUMS	Summary Size. Sumario de estimados de tamaño.
Tarea	Elemento de trabajo incluido en el plan de desarrollo
TASK	Forma de planeación de tareas.
TSPi	Introducción a Team Software Process.
UML	Unified Modeling Language. Lenguaje unificado de modelado.