



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**EL PROCESO DE DESARROLLO Y MANTENIMIENTO
DE SOFTWARE PROPUESTO POR COMPETISOFT DE
ACUERDO AL PROCESO UNIFICADO**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

MIGUEL EHÉCATL MORALES TRUJILLO

**DIRECTORA DE TESIS:
M. EN C. GUADALUPE ELENA IBARGÜENGOITIA GONZÁLEZ**

MÉXICO, D.F.

2010.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

Agradecimientos

Agradezco sinceramente a mi directora de tesis, la M. en C. Guadalupe Ibargüengoitia González, por su excelente asesoramiento, por el apoyo y tiempo que me dedicó, y sobre todo por el constante estímulo y confianza que siempre depositó en mí.

A Francisco Pino que a lo largo de mi estancia de investigación en Ciudad Real me guió en la realización de éste trabajo de una manera muy provechosa, y que además me ayudó a adaptarme y a sentirme como en casa.

Al jurado integrado por la Dra. Hanna Oktaba, la MSC. Gloria Quintanilla y los Drs. Javier García García y Fernando Gamboa Rodríguez por sus valiosos comentarios y sugerencias que no hicieron más que mejorar este trabajo.

Dedicatoria

A mi esposa Полина y a mi hijo Миша.

A mis padres.

A mis hermanas.

ÍNDICE

ÍNDICE.....	5
Introducción.....	7
0.1 Objetivo.....	7
0.2 Antecedentes.....	7
0.3 Estructura del trabajo.....	8
CAPÍTULO 1 Conceptos Básicos.....	9
1.1 Qué es un proceso.....	9
1.2 Qué son los modelos de referencia de procesos.....	10
1.3 Características de las PyMEs.....	14
1.4 Obstáculos para que las PyMEs adopten Modelos de Referencia de Procesos..	14
CAPÍTULO 2 Modelos de Procesos de Software para PyMEs.....	19
2.1 COMPETISOFT.....	19
2.2 Administración de Proyecto.....	21
2.3 Desarrollo de Software.....	23
2.4 Mantenimiento de Software.....	27
2.5 PmCOMPETISOFT.....	29
2.6 ISO/IEC 29110.....	30
CAPÍTULO 3 Adopción de Procesos de Software Basada en Plantillas.....	33
3.1 Adoptando un modelo de referencia de procesos a través de plantillas.....	33
3.2 Estructura de las plantillas.....	35
3.3 Diagrama de Unidades atómicas.....	37
CAPÍTULO 4 Plantillas de Administración del Proyecto.....	39
4.1 Plan del Proyecto.....	41
4.2 Documento de Aceptación.....	43
4.3 Matriz de Trazabilidad.....	44
4.4 Reporte de Seguimiento.....	45
CAPÍTULO 5 Plantillas de Desarrollo de Software.....	47
5.1 Especificación de Requerimientos.....	48
5.2 Análisis.....	49
5.3 Diseño.....	51
5.4 Construcción.....	53
5.5 Integración.....	55
5.6 Pruebas.....	56
5.7 Cierre.....	57
CAPÍTULO 6 Plantillas de Mantenimientos de Software.....	59
6.1 Petición de Modificación.....	59
6.2 Intervención.....	60
CAPÍTULO 7 Plantillas de PmCOMPETISOFT.....	63
7.1 Propuesta de Mejora.....	64
7.2 Plan General de Mejora.....	64
7.3 Plan de Implementación de Mejora.....	65
7.4 Reporte de Mejora.....	66
CAPÍTULO 8 Resultados Obtenidos.....	67
8.1 Guía para el uso de las plantillas.....	67



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

8.2 Difusión de las plantillas a través de E-learning	69
8.3 Adaptación de las Unidades Atómicas a otros modelos.....	70
8.4 Artículos publicados.....	74
Conclusiones y Trabajo Futuro	75
Contribuciones.....	76
Trabajo Futuro	77
Apéndice A Plantillas de Administración de Proyectos.....	79
Apéndice B Plantillas de Desarrollo de Software	97
Apéndice C Plantillas de Mantenimiento de Software.....	131
Apéndice D Plantillas de PmCOMPETISOFT.....	133
Apéndice E Diagramas de Unidades Atómicas.....	141
Apéndice F Constancia de participación como autor en el ENC'09 (Congreso Internacional Mexicano de Ciencias de la Computación 2009).....	151
Referencias	153

Introducción

0.1 Objetivo

El primer paso que efectúa una PyME al decidir implantar un modelo de procesos de software, es elegir un modelo de referencia adecuado a su escenario y posibilidades, que posteriormente deberá ajustar a sus necesidades específicas. Consideramos que el modelo de referencia puede ser adaptado a las necesidades de la PyME de una manera más sencilla mediante la utilización de plantillas, las cuales pueden permitir la integración adecuada de las actividades propuestas en él y aquéllas que la empresa ya venía realizando, resultando en la disminución de la brecha entre la definición del modelo elegido y su adopción.

El objetivo de este trabajo es presentar una alternativa centrada en plantillas de productos, basadas en técnicas descritas en el Proceso Unificado [Jacobson], RUP [Kruchten] y el Cuerpo de Conocimiento de la Administración de Proyectos (PMBok por sus siglas en inglés) [PMBok], que pretenden facilitar la adopción de los procesos de los modelos de referencia para PyMEs, definidos en COMPETISOFT y la futura norma ISO/IEC 29110. El propósito es incorporar los procesos a las PyMEs, con el apoyo de plantillas de productos que sean manejables, fáciles y entendibles, tomando en cuenta las limitantes intrínsecas de las pequeñas empresas, que se mencionan en la sección 1.4.

0.2 Antecedentes

Existen algunos modelos de referencia disponibles para PyMEs, como son COMPETISOFT [Oktaba 2007] o ISO/IEC PDTR 29110 [ISO/IEC 29110], en estos modelos se definen grupos de procesos, y éstos a su vez agrupan actividades que tienen que realizar las organizaciones para aplicarlos. Sin embargo, estos modelos se limitan a describir las actividades de cada grupo de procesos y los productos que se generan sin indicar técnicas y pasos concretos para realizarlos.

COMPETISOFT surge con el objetivo de incrementar el nivel de competitividad de las pequeñas empresas de software Iberoamericanas, difundiendo un marco común que pudiese ajustarse a las necesidades específicas de cada una de ellas, con el fin de establecer un mecanismo de evaluación y certificación dirigido a las PyMEs.

Buscando que el modelo sea el que se adapte a la organización, en COMPETISOFT se integran de manera consistente elementos básicos e indispensables para llevar a cabo una implantación adecuada del marco metodológico. El modelo está dividido en tres capas, Alta Dirección, Gerencia y Operación, esta estructura acumula los principios básicos que componen a una PyME. Por una parte la Alta Dirección es la encargada de la planificación estratégica y la revisión continua del todo en la organización; la Gerencia es la encargada de proveer los recursos, procesos y proyectos, además de vigilar el cumplimiento de los objetivos de la organización; y como tercer bloque se tiene la capa de Operación, donde se concentran los procesos de Desarrollo y Mantenimiento de Software así como la Administración del Proyecto.

La capa de Operación busca que cada uno de los proyectos de desarrollo de software que tiene la organización cumpla con los compromisos establecidos en tiempo, costo y calidad [Oktaba 2008]. Los procesos que establecen las prácticas requeridas para terminar



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

el desarrollo de software en tiempo y costo son los comprendidos en la Administración del Proyecto. Mientras tanto, en el Desarrollo de Software se establecen las actividades necesarias para cumplir con los requisitos y características del software y eliminar defectos en fases tempranas a través de las verificaciones y validaciones, asegurando criterios de calidad. Por último en el Mantenimiento de Software debe realizarse después de que se ha liberado el producto. Debido a la importancia crítica que tiene la capa de Operación para la organización, este trabajo se concentra en esos grupos de procesos para la realización de una guía de implantación basada en plantillas.

Una plantilla será la encargada de guiar cada una de las actividades descritas en un modelo de referencia. Una plantilla es un documento genérico que contendrá una descripción detallada de alguna técnica que espera ser suficientemente clara para poder llevar a cabo las actividades. Directamente, se establece una metodología para regir su estructura y desarrollo. En este caso las plantillas están basadas en las técnicas del Proceso Unificado para desarrollos orientados a objetos y PMBoK, ya que son estándares aplicables a cualquier desarrollo de software.

La elección del Proceso Unificado como regidor de las plantillas se debe en gran parte a que, al igual que COMPETISOFT, propone un desarrollo iterativo e incremental, es altamente adaptable a las necesidades específicas de cada organización y motiva el cumplimiento de los criterios de calidad. Agregado a esto, la aceptación que tiene como estándar entre la industria para desarrollos Orientados a Objetos.

Por su parte, PMBoK, es un estándar definido para la Administración de Proyectos, dividido en nueve áreas del conocimiento, en donde se engloban las mejores prácticas y se estandariza la información referente a la gestión de proyectos de manera consistente y aplicando ese conocimiento a la ingeniería de software.

A lo largo del trabajo se presentarán las características y el desarrollo de las plantillas, que pretenden ser una guía explícita de cómo concretar la realización de las actividades en el desarrollo de software orientado a objetos.

0.3 Estructura del trabajo

El trabajo está estructurado de la siguiente manera, en el capítulo 1 se presentan los conceptos básicos utilizados a lo largo del desarrollo. En el capítulo 2 se habla de los Procesos de Software. En el capítulo 3 se desarrolla la propuesta de la guía para el desarrollo de software basada en plantillas. Los capítulos 4, 5 y 6 presentan el desarrollo detallado de las plantillas de Administración del Proyecto, Desarrollo de Software y Mantenimiento de Software respectivamente, en estos capítulos se anexan ejemplos de uso de las plantillas así como sus características particulares. En el capítulo 7 se incluye el desarrollo de las plantillas para guiar un Proceso de Mejora. El capítulo 8 recopila los resultados obtenidos y por último se presentan las conclusiones y el trabajo futuro.

CAPÍTULO 1 Conceptos Básicos

Las organizaciones que se catalogan como pequeñas empresas (PyMEs) dentro de la industria del software, son fundamentales para muchas economías, en países como Estados Unidos, Brasil, Canadá, China, India, Finlandia, Irlanda y Hungría las PyMEs representan cerca del 85% de todas las organizaciones dedicadas al desarrollo de software [MTC] [Enterprise Ireland]. Por ello resulta ser la industria de software una actividad económica relevante en una gran mayoría de países. Sin embargo las PyMEs enfrentan varios obstáculos en su acontecer diario, lo que complica la mejora e incremento de sus capacidades de competencia al intentar adoptar algún modelo de referencia de procesos.

1.1 Qué es un proceso

Un proceso puede ser definido como un conjunto de actividades que se necesitan para transformar las necesidades de un cliente en un sistema de software, por otra parte un proceso es una red de actividades vinculadas ordenadamente las cuales se llevan a cabo repetidamente y que utilizan recursos e información para transformar insumos en productos, abarcando desde el inicio del proceso hasta la satisfacción de las necesidades del cliente [Ljungberg]. Trasladando esta definición a la Ingeniería de Software podemos decir que un proceso de software es un *conjunto de actividades que se necesitan para transformar las necesidades de un cliente en un sistema de software*.

De acuerdo a las definiciones anteriores es claro que para la ejecución de un proceso es necesaria la inclusión de elementos que permitan llevarlo a cabo, en la Ingeniería de Software estos elementos están bien definidos y son los Roles, las Actividades, las Tareas y los Productos. En la Figura 1.1 se muestran las relaciones existentes entre estos elementos.

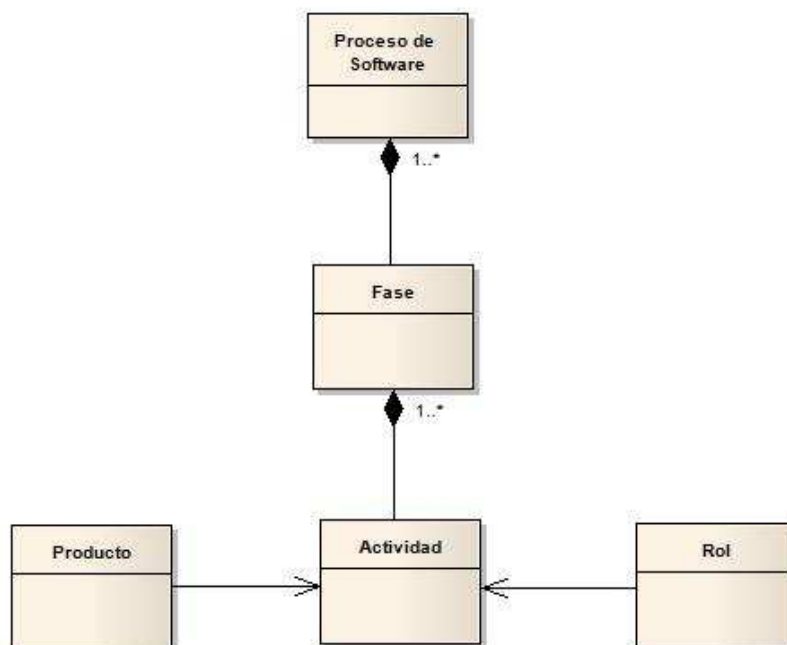


Figura 1.1 Proceso de Software [Oktaba 1998].



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

Además de definir a cada uno de los elementos que componen a un proceso de software, se deben mencionar las relaciones existentes entre ellos. Un Proceso de Software está compuesto por fases, pudiendo ser una o más, las fases de cada proceso se definen a través de objetivos que permitan establecer de manera clara el comienzo y fin de cada una.

Las actividades son los elementos que componen a cada una de las fases, cada actividad es descompuesta en pasos para indicar su secuencia, tiene asociados productos de salida o entrada además de un rol responsable de ejecutarla. El objetivo de cada actividad es transformar los productos de entrada en productos de salida, cada producto tiene un nombre, se establece en qué actividad se crea y cuáles se utiliza.

Los roles son el conjunto de actividades que debe realizar cada persona, según el papel que juega en un momento determinado del proceso, para definir los roles es necesario especificar el nombre, sus responsabilidades y las habilidades necesarias para desempeñar las actividades.

Los procesos pueden ser relacionados y conjuntados en grupos de procesos. Estos grupos son utilizados para la definición de modelos de referencia de procesos que buscan incrementar la madurez de la empresa y su capacidad.

1.2 Qué son los modelos de referencia de procesos

La gran mayoría de las empresas desarrolladoras de software guían sus procesos de acuerdo a las habilidades técnicas y capacidades tecnológicas de los individuos que la componen, sin embargo el enfocarse en éstos aspectos dificulta el desarrollo de capacidades como la organización del trabajo en equipo. Por ello es recomendable que las empresas se guíen de alguna manera a través del proceso. Como respuesta a esto surgen los modelos de referencia, los cuáles pretenden ser el estándar que le dé forma a los procesos que ejecuta la empresa.

El documentar el proceso conlleva establecer mecanismos para guiarlo y servir de referencia para la empresa, al encontrarse definido el proceso se está en posibilidad de mejorarlo y administrarlo, como consecuencia lleva a la empresa a un nivel de madurez mayor. Hasta este punto, resalta la necesidad de definir un proceso o grupos de procesos para construir software, ya sea siguiendo experiencias exitosas, adaptando las mejores prácticas tomadas de procesos ya probados o las más recurrentes, adoptando un modelo de referencia de procesos reconocido como estándar internacional.

1.2.1 Estándares internacionales

Un estándar es un documento, establecido por consenso y aprobado por una institución reconocida, que provee, para uso común y repetido, reglas, directrices o características para actividades o sus resultados, encaminados a la consecución de un grado óptimo de orden en un contexto determinado, un estándar debe basarse en resultados consolidados de la ciencia, tecnología y experiencia, encaminado a la promoción y difusión de beneficios óptimos para la comunidad [ISO/IEC 29110]. Por estándar internacional se entenderá a toda norma que es adoptada por una organización internacional de normalización y puesto a disposición del público.

Organizaciones internacionales como la Organización Internacional de Estandarización (ISO) y el Instituto de Ingeniería de Software (SEI) han desarrollado estándares dirigidos a empresas desarrolladoras de un tamaño mayor a las PyMEs, es por ello que las pequeñas empresas se enfrentaban a grandes obstáculos a la hora de intentar adoptar estos estándares. Dichos modelos no están orientados a satisfacer las necesidades específicas de las PyMEs, ya que resultan excesivos en términos de esfuerzo y costo considerando el entorno en el que se desenvuelven las pequeñas organizaciones. Para fortalecer este tipo de organizaciones se necesitan prácticas eficientes de Ingeniería del Software adaptadas a su tamaño y tipo de negocio [Pino 2006].

La existencia de normas internacionales como la ISO/IEC 12207 [ISO 2008] o la ISO/IEC 15504 [ISO 2004] de ISO o el estándar CMMI [SEI] no logra cubrir los aspectos requeridos por todas las organizaciones, en particular a las PyMEs les es complejo adquirir este tipo de modelos dirigidos a organizaciones de mucho mayor tamaño.

En un esfuerzo efectuado por la ISO la llevó a conformar el grupo de trabajo SC7-WG24 con el objetivo de establecer un modelo de referencia específicamente para VSEs (Very Small Entities), las VSEs son organizaciones cuya principal característica es que cuentan con 25 o menos empleados, en nuestro caso PyMEs. Como base, el WG24 ha tomado en cuenta a la norma mexicana MoProSoft [Oktaba 2005] para el desarrollo de la norma ISO/IEC 29110. La futura norma está compuesta por perfiles con el objetivo de facilitar a la organización la adopción del modelo de referencia, un perfil es definido como un conjunto de procesos para ayudar a aplicar la norma [Oktaba 2008].

1.2.2 Estándares para desarrollo de software

Entre los estándares más utilizados se encuentran el CMMI, y las ISO/IEC 12207 y 15504.

La ISO/IEC12207, que establece un ciclo de vida para el software, esta norma abarca procesos y actividades que se aplican a lo largo de todo el proceso de desarrollo, comenzando con la definición de requerimientos del software, para la creación o modificación del software, hasta la terminación de éste o su retirada, ver Figura 1.2. El objetivo principal de la ISO/IEC 12207 es proporcionar una estructura común para el desarrollo de software proveyendo procesos bien definidos a la organización [ISO 2004].

Proceso de Acuerdo	Proceso de Proyecto	Procesos Técnicos	Procesos de Implementación	Proceso de Soporte	Reutilización del Software
<ul style="list-style-type: none"> •Adquisición •Suministro 	<ul style="list-style-type: none"> •Planificación del proyecto •Evaluación y Control del proyecto •Administración de la Decisión •Administración de Riesgos •Administración de la Configuración •Administración de la Información •Medición 	<ul style="list-style-type: none"> •Especificación de Requerimientos •Análisis de Requerimientos •Diseño de la Arquitectura •Implementación •Comprobación de Requerimientos •Integración del Software •Aceptación del Software •Operación del Software •Mantenimiento del Software •Retirada del Software 	<ul style="list-style-type: none"> •Implementación del software •Análisis de Requerimientos •Diseño de la arquitectura •Diseño detallado •Construcción del software •Integración del software •Comprobación de Requerimientos 	<ul style="list-style-type: none"> •Administración de la Documentación •Administración de la Configuración •Aseguramiento de la calidad •Verificación del software •Validación del software •Revisión del software •Auditoría del software •Resolución de problemas del software 	<ul style="list-style-type: none"> •Ingeniería del Dominio •Administración de programas de Reutilización •Administración de Recursos Reutilizables
Procesos Organizacionales					
<ul style="list-style-type: none"> •Administración del Ciclo de Vida •Administración de la Infraestructura •Administración de la cartera de Proyectos •Administración de Recursos Humanos •Administración de la Calidad 					

Figura 1.2 Modelo de procesos ISO/IEC 12207:2008

Por otra parte la ISO/IEC 15504 es una norma internacional para establecer y mejorar la capacidad y madurez de los procesos de las organizaciones en la adquisición, el suministro, el desarrollo, la evolución y el soporte de productos y servicios. Esta norma proporciona un marco de trabajo para la evaluación del proceso y establece los requisitos mínimos para realizar una evaluación que asegure la consistencia de las valoraciones obtenidas [ISO 2004]. En la Figura 1.3 se muestran las partes que componen a la norma ISO/IEC 15504.

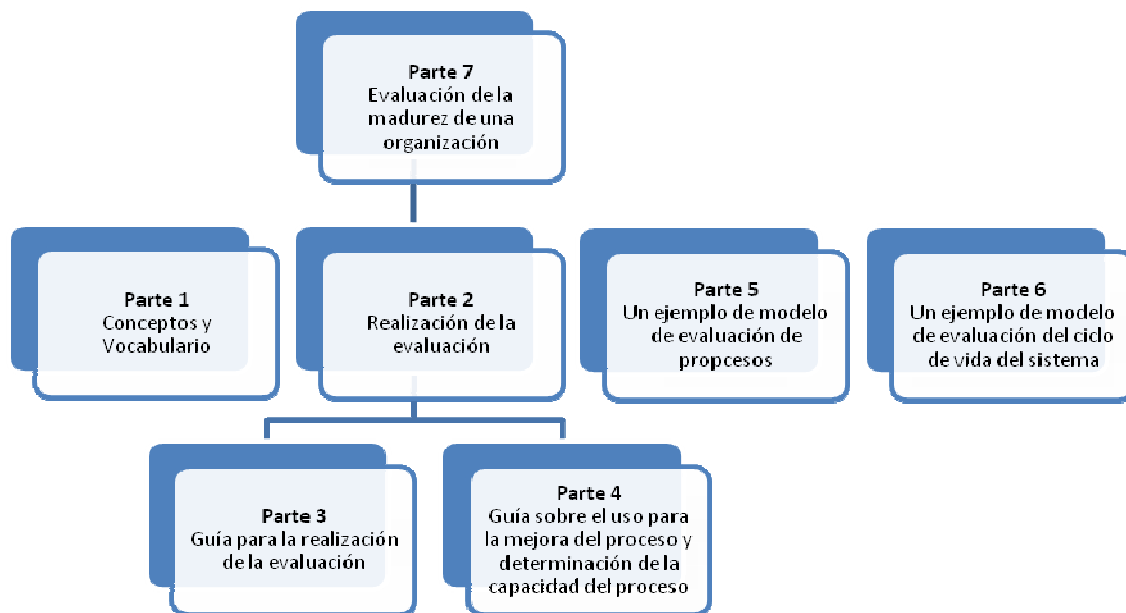


Figura 1.3 Partes de la norma ISO/IEC 15504

Agrupar los procesos y los evaluar en una escala de 6 niveles de acuerdo a la capacidad y madurez alcanzada:

- Nivel 0: Incompleto
- Nivel 1: Realizado
- Nivel 2: Gestionado
- Nivel 3: Establecido
- Nivel 4: Predecible
- Nivel 5: En optimización

El CMMI (Capability Maturity Model Integrated) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. De acuerdo con el SEI, el CMMI ayuda a *integrar las funciones de organización tradicionalmente separadas, establecer metas de mejora de procesos y prioridades, proporcionar una guía para procesos de calidad, y proveer un punto de referencia para evaluar los procesos actuales* [SEI].

En la Figura 1.4 se muestran los estándares más utilizados por PyMEs.

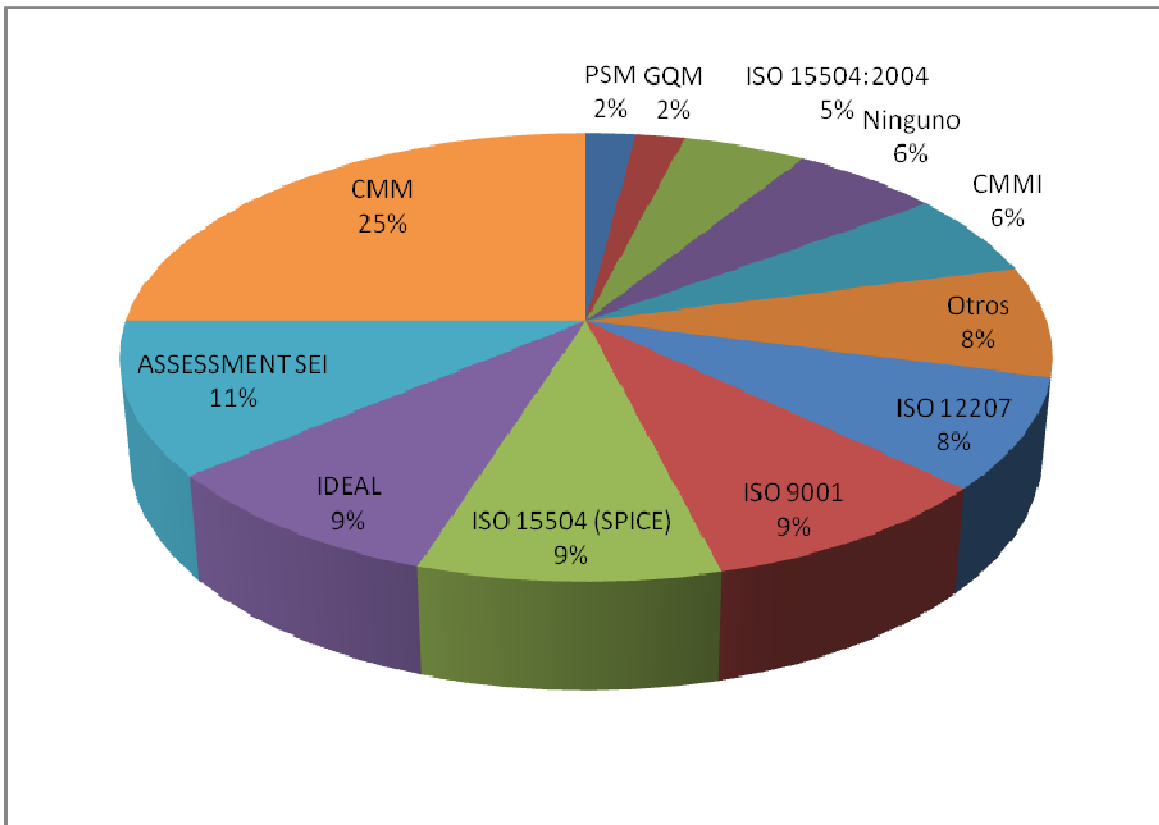


Figura 1.4 Estándares utilizados por las PyMEs

1.3 Características de las PyMEs

Las PyMEs son catalogadas de esa manera de acuerdo al número de empleados con que cuenta la organización, en un rango que varía desde los 10 hasta los 50 individuos. Las pequeñas empresas se caracterizan por centrarse en las habilidades técnicas de sus integrantes para conseguir sus metas, entre las principales particularidades de las PyMEs se pueden mencionar las siguientes:

- Tienen metas y modelos de negocio específicos
- Nicho de Mercado específico
- Recursos económicos limitados
- Buena capacidad de respuesta y flexibilidad como ventaja competitiva
- Administración informal
- Falta de personal que resulta de las limitaciones de recursos
- Falta de capacitación regular de los empleados

1.4 Obstáculos para que las PyMEs adopten Modelos de Referencia de Procesos

De acuerdo a los resultados recopilados por el WG24 se obtuvieron características más específicas como obstáculos para las PyMEs al intentar adoptar un modelo de referencia:

- Consciencia de la importancia de las normas y estándares
- Falta de conocimiento y dificultad en la aplicación de las normas debido a su complejidad
- Falta de conocimiento acerca de las metodologías y normas
- Falta de información suficiente para hacer uso efectivo de las normas disponibles
- Falta de lenguaje y comprensión de la terminología de la norma ISO [ISO/IEC 29110].
- Falta de recursos humanos con las capacidades requeridas.
- Factores económicos limitados.

1.4.1 Situación de la mejora de procesos en PyMEs

A pesar de los obstáculos que enfrentan las PyMEs al intentar adoptar un modelo, representan un porcentaje considerable del total de organizaciones que se encuentran involucradas en esfuerzos de mejora de sus procesos. El 80% de las organizaciones involucradas en esfuerzos de mejora son micro y pequeñas empresas. Sin embargo, a corto plazo, la mayoría de los casos no condujeron a una certificación o evaluación formal por parte de los organismos rectores, ISO o SEI [Oktaba 2008]. Se puede notar en la Figura 1.5 la distribución por país de los esfuerzos realizados.

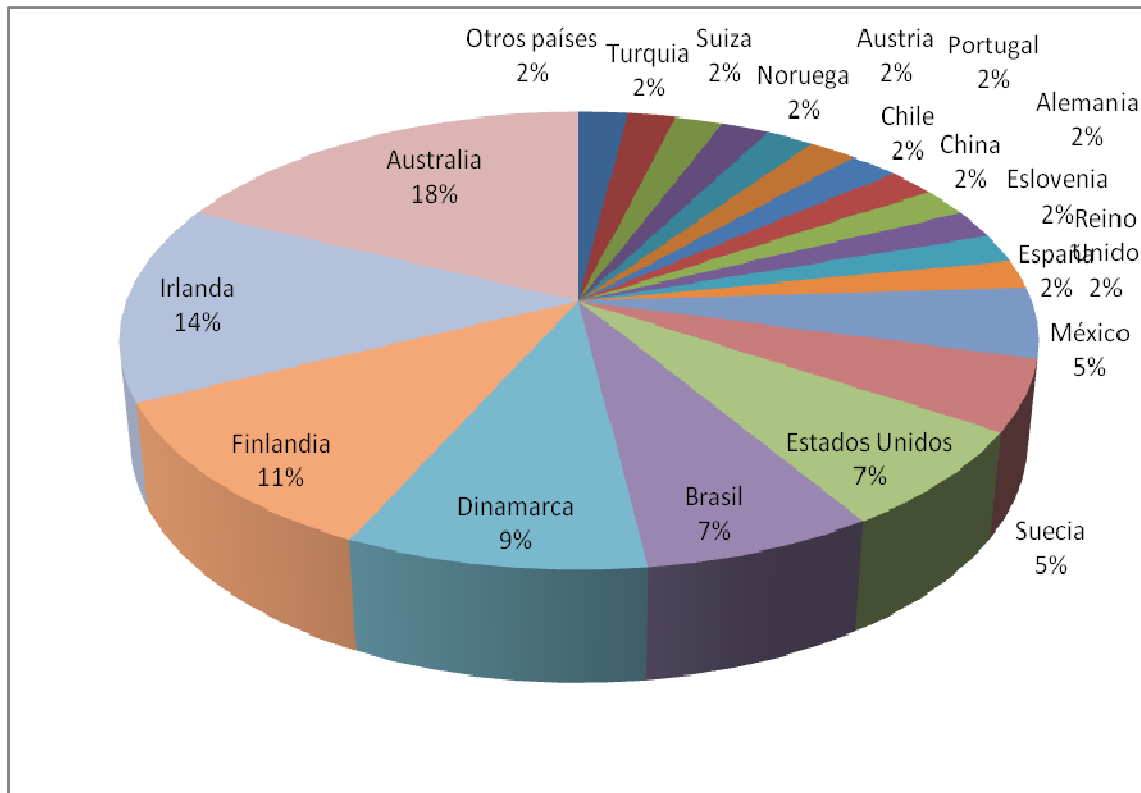


Figura 1.5 Distribución de los esfuerzos de mejora de procesos por país

1.4.2 Factores de éxito en la adopción de modelos de referencia en PyMEs

Entre los factores de éxito identificados por [Oktaba 2008] para que una PyME adopte satisfactoriamente un modelo de referencia y logre una mejora en sus procesos se pueden listar los siguientes:

- Estabilidad de la PyME
- Conseguir un rápido retorno de la inversión
- Minimizar la resistencia al cambio
- Involucrar a todos los miembros de la organización
- Evaluar frecuentemente
- Establecer un mecanismo de comunicación eficiente
- Llevar a cabo actividades de medición
- Comprometer a los directivos de la empresa
- Lograr la asesoría de un experto

Principalmente en el listado se puede notar que son cuestiones de compromiso por parte de la PyME. Contar con el apoyo de los principales miembros de la organización e incluir a cada uno de los demás miembros resulta fundamental para evitar truncar el proceso antes de obtener la mejora.

Además, el obtener un rápido retorno de la inversión (ROI) incrementa la motivación de la organización a continuar con el proceso, aquí adquiere importancia el hecho de tomar mediciones para poder cuantificar la evolución de la organización.

El ROI si bien es un indicador esencial en el área económica, no se utiliza eficientemente en el campo de los proyectos software [Garzás], en [Boehm] se resalta la falta de buenos modelos de ROI para software, lo que hace difícil la valoración de las inversiones.

El análisis de ROI más conocido es el de *costo-beneficio*, que tiene como objetivo proporcionar una medida de los costos en que se incurre en la realización de un proyecto y comparar dicha previsión de costos con los beneficios esperados de la realización de dicho proyecto. En general los costos suelen ser cuantificables y estimables en unidades económicas, no así los beneficios, los cuales pueden ser tangibles o intangibles [Garzás].

Es por esta razón que establecer un número preciso que defina el ROI se vuelve una tarea compleja, debido a la cantidad de variables que intervienen y afectan a cada PyME. Los beneficios que se toman en cuenta principalmente serán aquellos procesos mejorados después de implementar la adopción de un modelo.

Entre los procesos con mayor incidencia de mejora de acuerdo a [Oktaba 2008] está el de Administración de Proyectos, Documentación, Administración y Control de Cambios y Obtención de Requerimientos, ver Figura 1.6.

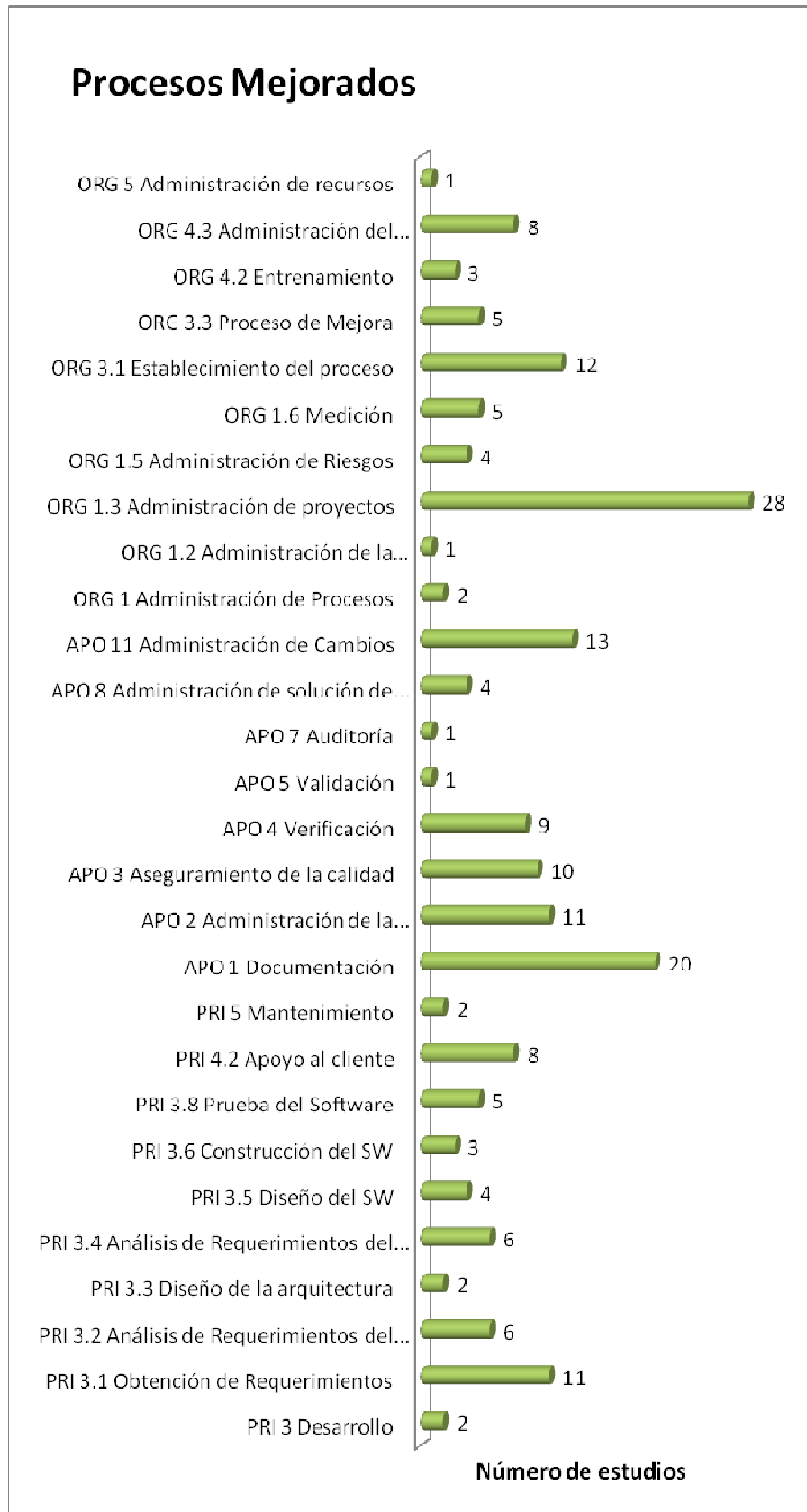


Figura 1.6 Procesos mejorados en la PyMEs

CAPÍTULO 2 Modelos de Procesos de Software para PyMEs

Actualmente, en un esfuerzo para que las PyMEs eviten el obstáculo de tener que adaptarse a modelos que sobrepasan sus capacidades y se inicien en un proceso de mejora, se han desarrollado modelos de referencia de procesos dirigidos específicamente a ellas como son COMPETISOFT e ISO/IEC 29110. COMPETISOFT es un modelo dirigido a PyMEs Iberoamericanas, mientras que la futura norma ISO/IEC 29110 tendrá un alcance internacional.

2.1 COMPETISOFT

Para hablar de COMPETISOFT tenemos que mencionar primero a MoProSoft. MoProSoft nace como parte del Programa para el Desarrollo de la Industria del Software (PROSOFT), iniciativa de la Secretaría de Economía. Una de las líneas estratégicas de este programa era "*Alcanzar niveles internacionales en capacidad de procesos*", como primera instancia se estudió la posibilidad de adoptar modelos de referencia existentes como CMMI o la norma ISO 15504. El análisis arrojó que dichos modelos eran excesivos para la mayoría de las empresas de desarrollo de software en México, por lo que únicamente fueron tomados como guías para el desarrollo de un modelo que considerara plenamente las características de la industria mexicana.

En el año 2005 se publicó el Modelo de Procesos para la Industria de Software o MoProSoft como norma mexicana NMX-059/01-NYCE-2005 [NYCE].

En ese mismo año, el Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED) comenzó la financiación del proyecto COMPETISOFT, cuyo propósito era incrementar el nivel de competitividad de las PyMEs Iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas [Orozco] incrementara sus capacidades y procesos lo cual impactara directamente en la calidad de su producto y por consiguiente la hiciera competitiva.

El modelo de referencia de procesos COMPETISOFT está dirigido a empresas o áreas internas de empresas dedicadas al desarrollo y mantenimiento de software, en particular aquellas catalogadas como PyMEs. COMPETISOFT buscó adaptarse desde su creación a las necesidades de las PyMEs de Iberoamérica, de acuerdo al contexto socio-económico en el que están inmersas.

COMPETISOFT está orientado a ser un modelo de referencia de procesos para una certificación o evaluación, contribuyendo a mejorar los resultados de la organización que lo aplica. No requiere de una estructura organizacional compleja, ya que el modelo abstrae tres elementos suficientes para definirla, Alta Dirección, Gerencia y Operación.

Los principales tres objetivos de COMPETISOFT fueron desde un comienzo los siguientes:

- Desarrollar un marco metodológico común iberoamericano.
- Difundir la cultura de mejora de procesos mediante la formación de investigadores, docentes y profesionales.



Universidad Nacional
Autónoma de México



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 esta 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.

- Incidir en los diferentes organismos de normalización y certificación de los países iberoamericanos, para que asuman que los principios metodológicos, objeto de este proyecto puedan ser la base para establecer un mecanismo común y mutuamente reconocido de evaluación y certificación de la industria del software iberoamericana. [Oktaba 2007]

Finalmente, COMPETISOFT está conformado por tres modelos, ver Figura 2.1:

Modelo de Referencia: establece tres categorías que agrupan procesos de acuerdo a la estructura típica de una organización: Alta Dirección, Gerencia y Operación. Está basado en MoProSoft.

Modelo de Evaluación: establece un mecanismo para la evolución de procesos definiendo un conjunto de medidas sobre los mismos. Está basado en el método de evaluación EvalProSoft y la ISO/IEC 15504-2.

Modelo de Mejora: establece los elementos necesarios para conducir una mejora de procesos en una PyME de forma ágil. Está basado en AgileSPI [Hurtado].

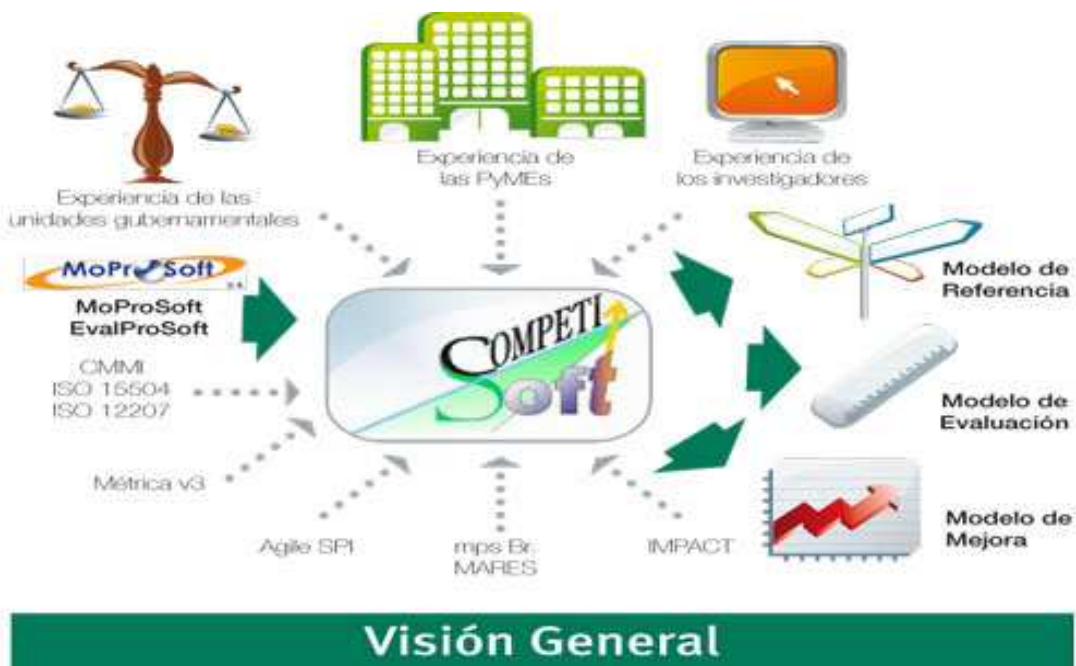


Figura 2.1 Proyecto COMPETISOFT [Orozco].

2.1.1 Categorías de procesos

Una categoría en el modelo se define como un conjunto de procesos que abordan una misma área general de actividad dentro de una organización. Conformado por las categorías de Alta Dirección, Gerencia y Operación, el modelo busca adaptarse a cualquier estructura organizacional existente en la PyME. En la Figura 2.2 se muestran las categorías del modelo de referencia de COMPETISOFT.



Figura 2.2 Modelo De Procesos [Orozco].

Categoría de Alta Dirección: Incluye las prácticas relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos.

Categoría de Gerencia: Incluye las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados a la Categoría de Alta Dirección.

Categoría de Operación: Incluye las prácticas de los proyectos de desarrollo y de mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados.

Profundizaremos en la Categoría de Operación, ya que es el objetivo principal del trabajo. Como ya se mencionó, los procesos incluidos en esta categoría son los encargados de llevar a cabo los proyectos señalados en la categoría de Gerencia, y está compuesta por tres procesos: Administración del Proyecto, Desarrollo de Software y Mantenimiento de Software.

2.2 Administración de Proyecto

La Administración del proyecto busca establecer los lineamientos para proveer de una planeación que cumpla con los objetivos en el tiempo, costo y alcance determinados, resultando en un producto con el nivel de calidad más alto posible. Para ello, se consideran como sus principales fases al Inicio, la Planeación, la Ejecución, el Control y el Cierre.

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único, con temporal se refiere a que existe un inicio y fin establecidos.

que delimitan al esfuerzo, el final se alcanza cuando todos los objetivos que dieron origen al proyecto han sido alcanzados, cuando la imposibilidad de ser alcanzados es clara, cuando la necesidad del proyecto ya no exista o éste sea cancelado.

Una característica de los proyectos es que la elaboración del producto, servicio o resultado único es gradual que comienza con la definición del alcance, para posteriormente realizar una construcción de cada una de las especificaciones definidas. Los proyectos son una forma de organizar actividades y tareas que no pueden ser incluidas dentro de los límites operativos normales de la organización [PMBok].

Se define a la Administración de Proyectos como la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer sus requisitos. La administración de proyectos se logra mediante la aplicación e integración de las fases anteriormente mencionadas: Inicio, Planeación, Ejecución, Control y Cierre. En la Figura 2.3 se muestra la relación entre estas fases.

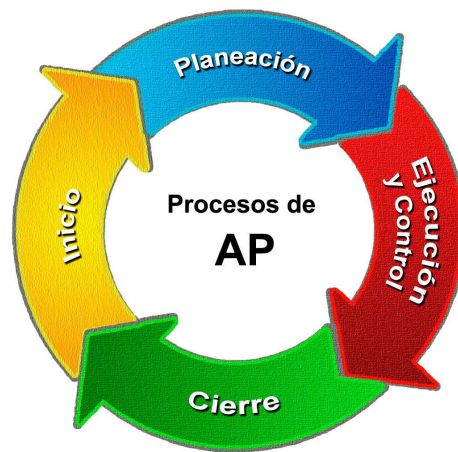


Figura 2.3 Procesos de la Administración de Proyectos

2.2.1 PMBoK

El Cuerpo de Conocimiento de la Administración de Proyectos o PMBoK, está dividido por áreas del conocimiento, cada uno de los cinco procesos incluyen actividades que pertenecen a alguna de éstas nueve áreas del conocimiento.

- Alcance
- Adquisiciones
- Calidad
- Comunicación
- Costo
- Integración
- Recursos Humanos
- Riesgo
- Tiempo

En el área de *Alcance* se administran y definen los productos y actividades, además de definir su seguimiento, que consiste en verificar y controlar el alcance. El área de *Adquisiciones* contempla la planeación de compras y adquisiciones, contrataciones, manejo de proveedores, selección y solicitud de vendedores, administración y cierre de contratos.

La *Calidad* está encargada de la planeación, aseguramiento y control de calidad. En el área de *Comunicación* se encarga de definir la manera en que se va a distribuir la información del proyecto y gestionar a los interesados. El *Costo*, estimación y control de costos, además de la preparación del presupuesto. La *Integración* es el área encargada de la constitución y definición del proyecto y las áreas involucradas, desde dirigir y administrar la ejecución del proyecto hasta que se realiza el cierre administrativo.

El área de *Recursos Humanos* planificación, adquisición, desarrollo y administración de los equipos del proyecto. El área encargada de administrar el *Riesgo*, los identifica, planifica su administración y la respuesta a éstos, les da seguimiento y control, además de realizar un análisis cuantitativo y cualitativo. El *Tiempo* está encargado de definir las actividades interrelacionadas y cronogramas, estimar recursos y duración para cada una de las actividades.

2.3 Desarrollo de Software

El propósito del proceso de Desarrollo de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software cumpliendo con especificado en la definición de requerimientos.

El desarrollo del sistema de software puede ser dividido en varios ciclos, de acuerdo a los requerimientos del mismo. Repitiendo cada una de las fases por cada ciclo, se obtiene un desarrollo iterativo e incremental, este tipo de desarrollo permite al cliente y al equipo de trabajo darse cuenta del avance del trabajo mediante revisiones a los entregables intermedios, además de vigilar la consistencia de lo especificado como requerimientos.

El ciclo de vida está claramente establecido, es un desarrollo cíclico, cada ciclo está compuesto por fases bien definidas y cada fase es descompuesta en una serie de actividades, éstas últimas a su vez, dependiendo de su complejidad, se pueden dividir en tareas, asignadas a algún miembro del equipo de trabajo. En la Figura 2.4 se muestra el desarrollo en ciclos y las fases componentes de cada uno.

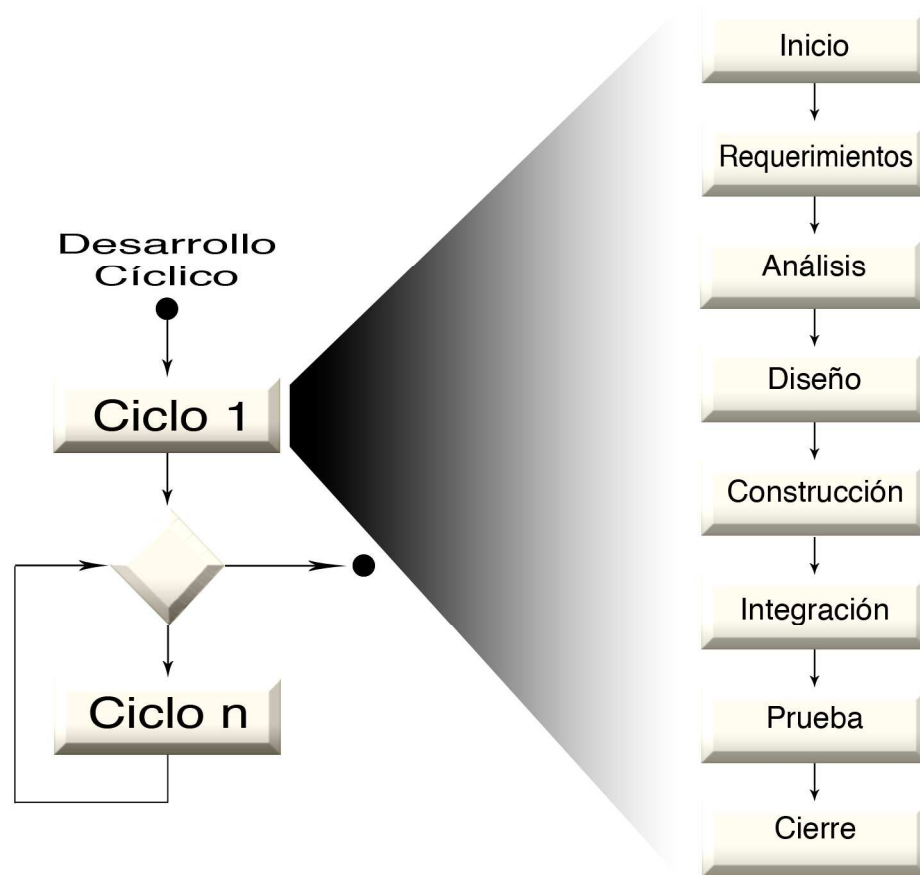


Figura 2.4 Desarrollo cíclico de acuerdo a COMPETISOFT

2.3.1 Proceso Unificado

El Proceso Unificado es un proceso de desarrollo de software, de acuerdo a sus creadores *es más que un simple proceso de software; es un marco de referencia genérico que puede ser especializado para una gran variedad de sistemas de software, para diferentes aplicaciones, diferentes tipos de organizaciones, diferentes niveles de competencia y proyectos de diferentes tamaños* [Jacobson].

El Proceso Unificado es un desarrollo basado en componentes, esto quiere decir que la construcción del sistema de software será en base a componentes de software conectados por interfaces bien definidas. El Proceso Unificado se puede definir en tres conceptos:

- Guiado por casos de uso
- Centrado en la arquitectura
- Iterativo e incremental

Por *Guiado por casos de uso* se refiere a que el sistema es originado debido a la exigencia de un usuario para cubrir sus necesidades, por lo tanto cada elemento que sea generado a lo largo del proceso será en respuesta para solventar alguna de estas necesidades. Los casos de uso son un medio sistemático e intuitivo que permite capturar los requerimientos funcionales del usuario, por otro lado posibilitan ligar cada uno de los flujos de trabajo propuestos. En consecuencia, son los casos de uso los que inician el desarrollo y transformaciones de los mismos las que van aconteciendo durante cada fase del desarrollo, por lo que los casos de uso resultan ser un mecanismo que apoya la trazabilidad a través de cada modelo del Proceso Unificado, manteniendo su integridad y consistencia.

La arquitectura involucra aspectos del comportamiento estático y dinámico del sistema, es influenciada por varios factores como lo son la plataforma, los componentes reutilizables, consideraciones de despliegue y sobretodo los requerimientos no funcionales [Jacobson]. La arquitectura es lo que le da forma al sistema completando aquellos aspectos que por sí solos los casos de uso no especifican, es por ello que el Proceso Unificado está *Centrado en la arquitectura*.

Por último, el ser *Iterativo e incremental*, permite identificar y especificar aquellos casos de uso que serán implementados en cada ciclo o iteración, obteniendo un producto funcional intermedio a partir del cual se irán anexando funcionalidades en incrementos graduales. Como consecuencia al tener un proceso iterativo reduce el costo de los riesgos que podrían ocurrir si se construye el sistema en un único incremento, en caso de que se requiera repetir la iteración el esfuerzo necesario es menor.

El Proceso Unificado está compuesto por modelos relacionados entre sí, ver Figura 2.5:

- **Modelo de Casos de Uso:** contiene a todos los casos de uso y a las relaciones con los usuarios
- **Modelo del Análisis:** el cual tiene dos propósitos, refinar y detallar los casos de uso y proveer un conjunto de objetos que provean de comportamiento al sistema.
- **Modelo del Diseño:** en él se define la estructura estática del sistema en subsistemas, clases e interfaces, además de definir el comportamiento dinámico como colaboración entre los mismos.
- **Modelo de Implementación:** es donde se realiza el mapeo entre las clases y los componentes identificados para comenzar con su construcción.
- **Modelo de Despliegue:** aquí se definen los nodos físicos del sistema y el mapeo entre los componentes de cada uno de esos nodos.
- **Modelo de Pruebas:** describe los casos de prueba para verificar y validar cada uno de los casos de uso.

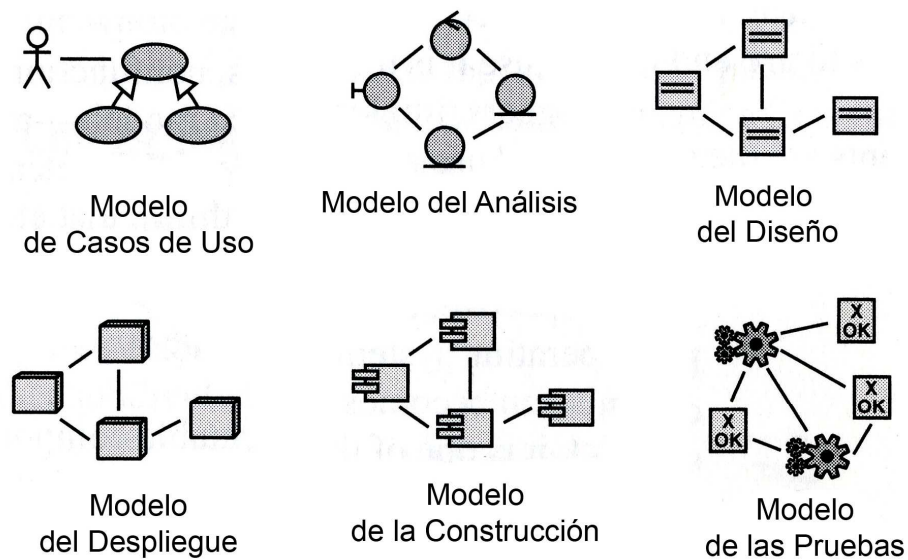


Figura 2.5 Modelos del Proceso Unificado

Para completar al Proceso Unificado se identifican elementos fundamentales, conocidos como *las cuatro Ps*, refiriéndose realmente a cinco elementos, ver Figura 2.6:

- **Personas:** se refiere a los arquitectos, desarrolladores, analistas, probadores, stakeholders, etc.
- **Proyecto:** se refiere al elemento organizacional a través del cual el desarrollo de software es controlado, como salida del proyecto se obtiene un producto de software.
- **Producto:** artefactos que son construidos a lo largo del proyecto.
- **Proceso:** es la definición del conjunto completo de actividades necesarias para transformar una serie de requerimientos del usuario en un producto. Un Proceso es una plantilla para crear un proyecto.
- **Herramientas:** son el software que es usado para automatizar actividades definidas en el proceso.

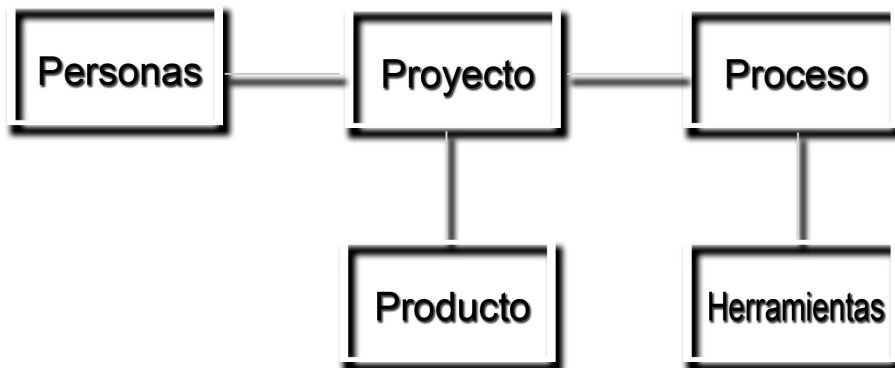


Figura 2.6 Las cuatro Ps

Debido a que el Proceso Unificado es un estándar genérico para desarrollos orientados a objetos que de ninguna manera se contrapone al proceso de Desarrollo de Software propuesto por COMPETISOFT, se decidió guiar Desarrollo de Software desde una perspectiva orientada a objetos, mediante el Proceso Unificado.

2.4 Mantenimiento de Software

El proceso de Mantenimiento de Software es el encargado de atender peticiones de modificación, corrección o incremento de funcionalidades a un sistema de software existente. Su objetivo es llevar a cabo dichas intervenciones cumpliendo con las necesidades del cliente sin que el sistema de software pierda consistencia.

Dentro de este proceso se consideran mecanismos para captar, analizar y dar seguimiento a las peticiones recibidas. El proceso de mantenimiento puede verse como el conjunto de todas las operaciones que es necesario realizar sobre el software para implementar las modificaciones solicitadas. Sin embargo, para dotar a este conjunto de operaciones de una base metodológica, es preciso definir con anterioridad el propio proceso de mantenimiento, detallando qué debe realizarse, cuándo, cómo y por quién, de tal manera que cada intervención de mantenimiento que se lleve a cabo, sea conforme a un proceso de mantenimiento predefinido [Polo].

Por esta razón se utiliza la versión ágil de MANTEMA para detallar el proceso de Mantenimiento de Software de COMPETISOFT.

2.4.1 Agil_MANTEMA

Agil_MANTEMA [Pino 2008] es una metodología ágil para el mantenimiento de software dirigida a pequeñas organizaciones. Tiene el objetivo de proveer a un nivel de detalle adecuado lo que debe realizarse antes, durante y después de una intervención de mantenimiento. Está basada en MANTEMA [Polo] y SCRUM [SCRUM].

La estructura del ciclo de vida de la metodología de Agil_MANTEMA está definida en cuatro fases, la *Definición del Proceso de Mantenimiento*, *Registro y Análisis de la Petición*, *Ejecución de la Intervención* y la *Migración y retirada del software*. A su vez Agil_MANTEMA define tres niveles de servicio: *Básico*, *Intermedio* y *Avanzado*. Los cruces entre los tipos de mantenimiento y los niveles de servicio se observa en la Tabla 2.1.

	Nivel Básico	Nivel Intermedio	Nivel Avanzado
Tipos de Mantenimiento	<ul style="list-style-type: none"> • Correctivo Urgente 	<ul style="list-style-type: none"> • Correctivo Urgente • Correctivo No Urgente • Perfectivo 	<ul style="list-style-type: none"> • Correctivo Urgente • Correctivo No Urgente • Perfectivo • Adaptativo • Preventivo
Interfaz fundamental	<ul style="list-style-type: none"> • Soporte al cliente • Administración de resolución de problemas 	<ul style="list-style-type: none"> • Soporte al cliente • Administración de resolución de problemas • Administración de la Configuración • Aseguramiento de la Calidad 	<ul style="list-style-type: none"> • Soporte al cliente • Administración de resolución de problemas • Administración de la Configuración • Aseguramiento de la Calidad • Administración de cambio de requerimientos • Administración de Proyectos

Tabla 2.1 Niveles de servicio de Agil_MANTEMA

Existen cinco tipos de mantenimientos en la metodología, organizados en las categorías de Planificable y No planificable. Con esta división se pretende lograr una mejor administración.

- **Mantenimiento No Planificable:**
 - **Correctivo urgente (Nivel Básico):** Es aquél que se da en situaciones en que existe un error en el producto software que bloquea la aplicación o el proceso de funcionamiento de la empresa, que debe ser resuelto a la brevedad.
- **Mantenimiento Planificable:**
 - **Correctivo no urgente (Nivel Intermedio):** Se produce cuando existe un error en el producto software que no es crítico, pero que tal vez impida el funcionamiento de la aplicación o el normal funcionamiento de la empresa en un periodo de tiempo relativamente corto.
 - **Perfectivo (Nivel Intermedio):** Se ocupa de añadir al software en explotación nuevas características o funcionalidades, habitualmente solicitadas por el cliente.
 - **Adaptativo (Nivel Avanzado):** Se aplica cuando el software en explotación va a cambiarse para que continúe funcionando correctamente en un entorno cambiante.
 - **Preventivo (Nivel Avanzado):** Es aplicado cuando se desea mejorar las características internas de un producto software buscando que en un futuro el esfuerzo de mantenimiento sea menor [Pino 2008].

Como participantes en el proceso de Mantenimiento de Software, Agil_MANTEMA identifica a tres entidades:

- **Ciente:** Es la organización propietaria, por lo tanto es quien recibe el servicio de mantenimiento.
 - Propietario del Producto: Representa a todos los interesados en el producto final. El propietario del producto por lo general formula peticiones de modificación del tipo perfectivo o adaptativo.
- **Usuario:** Es quien utiliza el software. Propone las peticiones de modificación correctivas, urgentes o no urgentes, y perfectivas.
- **Mantenedor:** Es quien realiza la modificación del software.
 - **Administrador de Peticiones:** Es quien acepta o rechaza las peticiones de modificación y decide el tipo de mantenimiento que corresponde.
 - **Responsable de Mantenimiento:** Es el responsable del mantenimiento, prepara el proceso y establece las normas y procedimientos necesarios para aplicar la metodología.
 - **Equipo de Mantenimiento:** Es el grupo de personas que implementa las peticiones de mantenimiento.

2.5 PmCOMPETISOFT

Para completar el marco metodológico, además del Modelo de Referencia, COMPETISOFT cuenta también con el Modelo de Evaluación y el Modelo de Mejora. El Modelo de Mejora llamado PmCOMPETISOFT es un proceso ligero que guía la ejecución de un ciclo de mejora en la PyME, está basado en AgileSPI [Hurtado].

El propósito de PmCOMPETISOFT es mejorar el conjunto de procesos de la organización en función de sus objetivos de negocio. Como objetivos principales tiene:

- Lograr la mejora de procesos de manera disciplinada mediante el cumplimiento y realización sistemática de las actividades y productos de trabajo propuestos.
- Definir objetivos y metas para el ciclo de mejora con base en los objetivos del negocio de la organización descritos en su plan estratégico.
- Evaluar los resultados de cada ciclo de mejora respecto a las mejoras introducidas en los procesos. También monitorear y supervisar el ciclo de mejora evaluando frecuentemente su eficiencia en la organización.
- Identificar nuevas estrategias para mejorar el proceso y las lecciones aprendidas, con el fin de aprender continuamente el proceso y mejorarlo con la experiencia adquirida por la gente que participa en el ciclo de mejora.

PmCOMPETISOFT es un proceso iterativo e incremental. Cada *iteración de mejora* se lleva a cabo dentro de un *ciclo de mejora* con el fin de implementar *casos de mejora*. Una *iteración de mejora* es un incremento pequeño y concreto que permite adelantar la implementación y administración de un *caso de mejora* de manera independiente.

Un *caso de mejora* agrupa oportunidades de mejora concretas que pueden ser ejecutadas en corto plazo con el fin de entregar de manera continua mejoras significativas de los procesos de la organización. El propósito de dividir la mejora en casos de mejora es obtener resultados rápidos de mejora [Oktaba 2008]. Para completar el proceso de mejora se incluyen cinco roles: el Responsable de Mejora de Procesos, el Grupo de Mejora de Procesos, Grupo de Gestión de Mejora, Responsable de Proceso y Evaluador.

En resumen PmCOMPETISOFT establece los elementos necesarios para guiar y administrar una mejora de procesos en una PyME, ocupando pocos recursos y buscando obtener resultados visibles a corto plazo.

2.6 ISO/IEC 29110

La familia de documentos de la ISO/IEC 29110 está compuesta de varios documentos cada uno con distintos propósitos, agrupados en Perfiles y Guías. Por una parte los Perfiles tienen como objetivo *empaquetar referencias a y/o partes de otros documentos de manera formal, con el fin de adaptarlos a las necesidades y características de las PyMEs* [García]. Cada perfil está compuesto por los siguientes dos elementos:

- **Marco de trabajo y taxonomía (TR29110-2):** Especifica los elementos comunes a todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles ISO/IEC 29110.
- **Especificaciones de perfil (TR29110-4):** Proporciona la composición definitiva de un perfil, los enlaces normativos al subconjunto normativo de estándares usados en el perfil, y los enlaces informativos (referencias) a documentos de “entrada”. Para cada perfil existe un documento de este tipo.

Por otra parte las Guías contienen *directrices de aplicación sobre cómo realizar los procesos para alcanzar los niveles de madurez*. Se desarrollan para la implantación de los procesos y para la evaluación de los mismos [García]. Al igual que los Perfiles, las Guías tienen dos componentes:

- **Guías de evaluación (TR29110-3):** describen el proceso a seguir para realizar una evaluación que determine las capacidades de proceso y la madurez organizacional.
- **Guías de ingeniería de gestión (TR29110-5):** proporcionan orientación sobre la implementación y uso de un perfil. Para cada perfil, existe una guía de este tipo.

Para completar a la familia de documentos se tiene a la Visión General (TR 29110-1) que en la cual se incluyen los conceptos principales para comprender y utilizar correctamente los demás documentos de ISO/IEC 29110, ver Figura 2.7.

Por el momento la ISO ha desarrollado este conjunto de Reportes Técnicos (TR), los cuales serán la base para la futura norma ISO/IEC 29110.

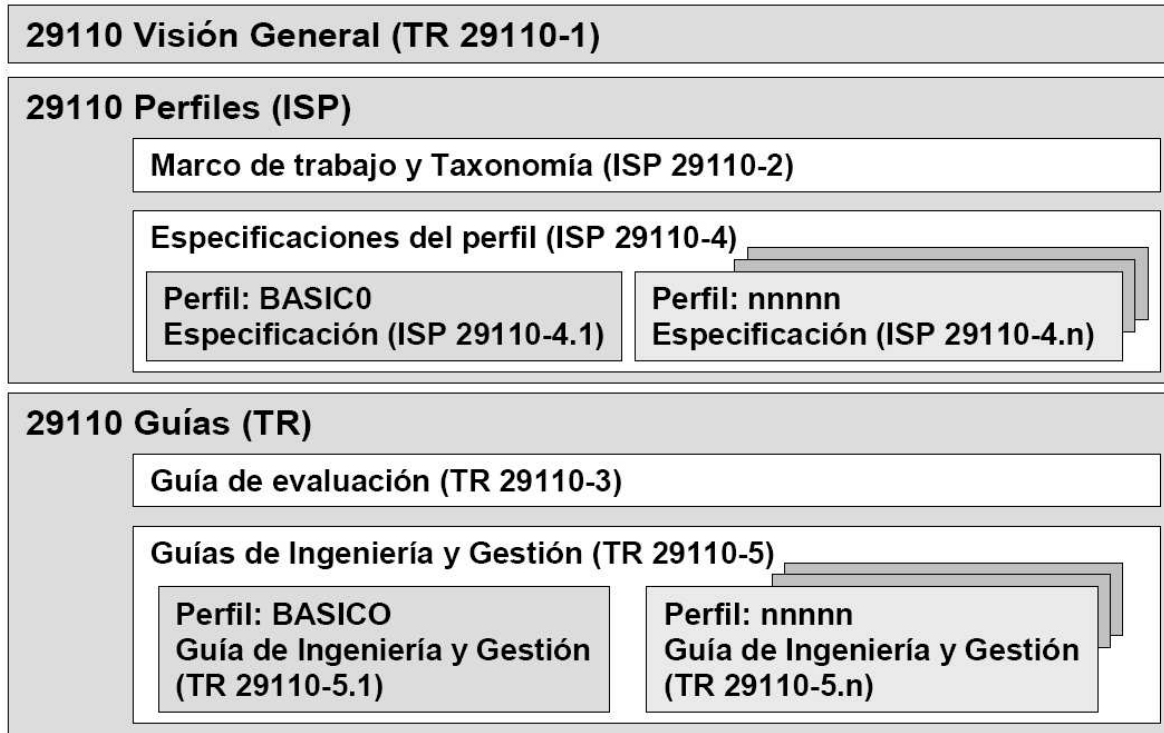


Figura 2.7 Familia de documentos de la ISO/IEC 29110

El objetivo de la futura norma es proveer de un modelo de referencia a las PyMEs especialmente hecho para ellas, haciendo al contenido de la norma más accesible para las organizaciones, adaptándolo a su tamaño y necesidades del negocio.

CAPÍTULO 3 Adopción de Procesos de Software Basada en Plantillas

Nuestra propuesta es encuadrar al marco metodológico de COMPETISOFT en un desarrollo que se base en plantillas, en las que se plasmarán las guías para realizar las actividades para generar los productos estandarizados, que cumplan con los requerimientos adecuados para incrementar el nivel de madurez de la organización y su capacidad de procesos.

Para que las PyMEs obtengan una mejora continua, es necesario que adopten modelos de referencia de mejora de procesos. Sin embargo es muy frecuente que las PyMEs consideren que la implantación de estos modelos es demasiado costosa, requiere demasiado tiempo y que sobretodo piensan que dichos modelos están dirigidos a organizaciones grandes, lo que resulta un obstáculo para adoptarlos en las PyMEs [Richardson] [Laporte 2008a].

Modelos como COMPETISOFT e ISO/IEC 29110 simplifican las actividades requeridas para llevar a cabo la mejora. Sin embargo, estos modelos en general no describen cómo realizar dichas actividades, dejando a la organización a un paso de concretar la adopción de procesos de software.

El paso de ejecutar las actividades descritas no es trivial, ya que los modelos de referencia expresan el qué hacer pero no el cómo. En este punto la organización enfrenta la problemática de definir cómo realiza las actividades y generar un producto. Una estrategia para guiar la implementación de procesos de software según los modelos de referencia de procesos [Kong], dentro de una organización es el uso de plantillas para los productos que guíen en la ejecución de las actividades. Como un apoyo a esta estrategia, el presente trabajo exhibe la propuesta de una guía basada en plantillas.

El uso de las plantillas ayuda a reducir el desconocimiento que tienen las organizaciones de los modelos, ya que representan los resultados de las actividades que deben realizar, indicando el cómo, y disminuyen la necesidad de recurrir a otro material de apoyo o fuente de conocimiento. Es decir, la plantilla actuará como guía explícita y receptáculo de los resultados de cada actividad.

3.1 Adoptando un modelo de referencia de procesos a través de plantillas

Una plantilla la podemos definir como un documento genérico que sirve como punto de partida para la creación de un nuevo documento. Es decir, una plantilla es una forma o patrón utilizado como guía para realizar algo [Diccionario Informático].

La utilización de plantillas como guías explícitas de la mejora de procesos espera lograr un aporte a la organización, al ser una herramienta [Word Template Index] [Artifact Overview RUP] que disminuirá la brecha entre la organización y el uso de los modelos de referencia. En particular este trabajo presentará el desarrollo de plantillas siguiendo el modelo de referencia COMPETISOFT, enfocado a la capa de Operación compuesta por la Administración de Proyectos, Desarrollo de Software y Mantenimiento de Software.

El desarrollo de las plantillas comenzó con los procesos de Administración del Proyecto, posteriormente con los de Desarrollo y Mantenimiento. Para cada uno de los



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

procesos establecidos se identificaron aquellas actividades que generaban un producto y aquellas que no lo hacían.

Para aquellas actividades que tienen un producto asociado, la plantilla respectiva ofrece una técnica para la realización de la actividad en un desarrollo orientado a objetos. Cada una de las técnicas elegidas fue cuidadosamente seleccionada para que éstas satisficieran los requerimientos definidos en el modelo de referencia. Cuidando siempre que su aplicación quedara bajo el alcance y recursos de la PyME, sin que esto evitara que se alcanzara una cobertura satisfactoria de cada actividad.

Por ejemplo, para la Especificación de Requerimientos se propone realizarla con la técnica del *Diagrama de casos de uso*, para la creación del Prototipo de interfaz de usuario se propone un *taller de requerimientos* y para el Plan de pruebas de integración se propone hacerlo por medio de un *grafo de dependencias*.

Las técnicas se recopilaron del PMBoK, para la Administración del Proyecto, del Proceso Unificado y RUP, para el Desarrollo de Software y de Agil_MANTEMA para el Mantenimiento de Software orientado a objetos, ver Figura 3.1. En cuanto a la Administración del Proyecto [Charbonneau] presenta un mapeo conjunto entre RUP y el PMBoK. A través del mapeo, se hacen notar las similitudes y diferencias que existen entre ellos. Fundamentalmente, el Proceso Unificado es un estándar para desarrollos de software orientado a objetos iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura, abarcando las mejores prácticas dentro de un contexto de desarrollo de software, mientras que el PMBoK contiene mejores prácticas genéricas y aplicables a la administración de proyectos en cualquier dominio de los nuevos procesos de negocio en una empresa. Concluyendo que no hay contradicciones entre los dos estándares por lo que las prácticas del RUP y las de PMBoK pueden coexistir sin anomalías. Lo que nos permite establecer una traza entre PMBoK, RUP, el Proceso Unificado y COMPETISOFT.

Por otro lado, las principales razones por las que se eligió al Proceso Unificado en la parte de Desarrollo y Mantenimiento de Software se deben a que estos procesos están dirigidos por casos de uso y centrados en la arquitectura. Además de acuerdo al estudio realizado en [Cánepa] se concluye que el nivel de cobertura entre estos dos procesos es altamente alcanzado de acuerdo a la calificación de ISO/IEC 15504 [van Loon]. La norma ISO/IEC 15504 provee un modelo para la evaluación de procesos. Mediante un conjunto de requerimientos mínimos que deben cumplir los procesos se determina el nivel de cobertura.

Sin embargo, se encontró que existían productos propuestos en COMPETISOFT que RUP no describe, por ejemplo el *Reporte de Pruebas de Integración* para el que COMPETISOFT destina una fase completa del Desarrollo, por lo que en casos similares se optó por utilizar otros estándares de la industria como base para adaptarlos a los objetivos de este trabajo.

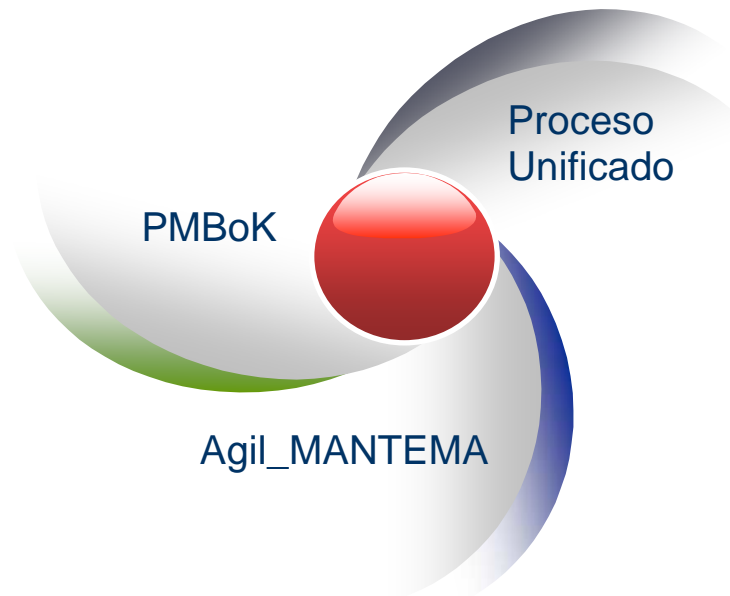


Figura 3.1 Adopción de Procesos de Software basada en Plantillas

3.2 Estructura de las plantillas

La característica principal de las plantillas es su estructura modular, compuesta por *unidades atómicas* maleables a entidades del proceso, lo que permite adaptar no sólo a COMPETISOFT, sino como se mostrará más adelante, a otros modelos tales como la futura norma ISO/IEC 29110. Cada unidad atómica del proceso contiene los elementos necesarios para llevar a cabo la actividad asociada. Las actividades son las marcadas en el modelo de referencia de COMPETISOFT, en específico de la capa de Operación del modelo.

3.2.1 Unidades Atómicas

Una plantilla es un conjunto de *unidades atómicas* [Morales] necesarias para cubrir un proceso o grupo de procesos descrito en el modelo de referencia. Formalmente, definimos a una unidad atómica como el elemento que contiene todos los aspectos necesarios para cubrir una entidad de proceso. Por entidad de proceso entendemos como aquella entidad que se encuentra definida en el modelo de referencia ya sea como una actividad especificada, *Verificación y Validación* por ejemplo, o como un componente de algún producto requerido como salida, por ejemplo el *Plan de Comunicación*.

Las unidades atómicas fueron identificadas a partir de lo descrito en el modelo de referencia, cada una de ellas está encargada de proveer al desarrollador una técnica explícita para los requerimientos particulares de cada entidad del proceso. Para la construcción del contenido de las unidades atómicas fue necesario elegir un mecanismo para plasmar los resultados de ejecutar las actividades, el mecanismo elegido fue presentar técnicas ampliamente aceptadas por la industria del software.

Particularmente, la elección de los componentes de cada *unidad atómica* estuvo dirigida por los requerimientos de COMPETISOFT siempre vigilando que su diligencia fuese sencilla y autocontenida en la plantilla.

Se pueden diferenciar dos tipos de unidades atómicas: las generales, que son comunes a todas las plantillas y están enfocadas a describir aspectos como la *Introducción*, *Descripción general del documento*, *Justificación de decisiones* e *Información de apoyo*; y las propias del proceso que son las que contendrán todo aquello que generen las actividades descritas en el modelo de referencia.

Cada entidad identificada es tratada como una unidad atómica en la plantilla, y cada unidad atómica contiene una descripción de la técnica propuesta para generar su contenido. Con esto se provee una descripción con más detalle de la manera de generar el producto para que cumpla con lo requerido por las actividades del proceso.

La descomposición en unidades atómicas facilita la trazabilidad de los elementos a lo largo del ciclo de vida del software, consecuencia directa del ciclo de vida del Proceso Unificado, y a las dependencias existentes entre cada uno de los modelos que lo componen.

Ejemplo de esto es la traza entre los requisitos solicitados por el cliente que son modelados mediante los casos de uso, para posteriormente ser especificados en un *Modelo del Análisis*, realizados en el *Modelo del Diseño*, implementados en el *Modelo de Implementación* y finalmente verificados en el *Modelo de Pruebas*. Todos estos modelos están relacionados y en conjunto representan al sistema como un todo. Elementos en un modelo son trazables, hacia atrás o hacia adelante, a otro modelo del propio Proceso Unificado, siendo rastreables estas correspondencias en cualquier instante del desarrollo.

Entre las principales características de las plantillas está la modularidad con la que cuentan las entidades del proceso que componen su estructura, ya que permiten adecuarlas a varios modelos de referencia. La modularidad de las plantillas se logra mediante la inclusión o exclusión de unidades atómicas de acuerdo a las necesidades del modelo y del desarrollador, ocasionando una alta adaptabilidad de las plantillas a cualquier desarrollo orientado a objetos.

En la Figura 3.2 se muestra la vista genérica de una plantilla y la estructura de una unidad atómica.



Estructura de las unidades atómicas

Nombre de la Plantilla

1. Nombre de la unidad atómica

[Descripción de la técnica propuesta para generar el contenido que cubra a la entidad del proceso]

1.1 Elementos de la unidad atómica

[Sub-unidades necesarias para completar lo requerido por la entidad del proceso]

1.1.1 <Tablas propuestas>

Nombre de la tabla		

Figura 3.2 Estructura de una unidad atómica

3.3 Diagrama de Unidades atómicas

Para mostrar de una manera gráfica la aplicación de las plantillas durante la ejecución del proceso, se procedió a realizar *Diagramas de Unidades Atómicas* [Morales] en los que se representan tres elementos principalmente: roles involucrados, listado de actividades descritas en el modelo y unidades atómicas de la plantilla que son utilizadas.

Cada diagrama de unidades atómicas cuenta con un diagrama de estados que representa la secuencia de actividades establecidas en el modelo. Para aquellos estados que representan a una actividad que tiene un producto de salida asociado, en el diagrama de unidades atómicas se establece la relación entre el estado y la unidad atómica de la plantilla que refleja la generación del producto.

Existen dos elementos que pueden ser asociados a un estado y que no representan una unidad atómica de la plantilla en uso, uno de ellos son las *Verificaciones y Validaciones* que son representadas por un óvalo con las siglas V&V y que su realización depende totalmente del equipo de trabajo. Por otro lado el segundo elemento son aquellas actividades cuya ejecución requiere del uso de una unidad atómica que no pertenece a la plantilla en uso, ejemplo de esto es la actualización de la Matriz de Trazabilidad, estos casos son representados con el ícono utilizado para representar a una plantilla externa. Por último, se establece el rol responsable de proveer de contenido a cada unidad atómica mediante carriles.

En la Figura 3.3 presentamos como ejemplo un diagrama de unidades atómicas genérico. En la parte izquierda de la figura, se muestra el grafo de las actividades descritas en el modelo de referencia. Por medio de carriles se diferencian los roles involucrados (columnas) en cada una de las actividades. Dentro de cada carril se señala la unidad atómica de la plantilla mediante la cual se refleja la realización de la tarea o actividad del proceso correspondiente.

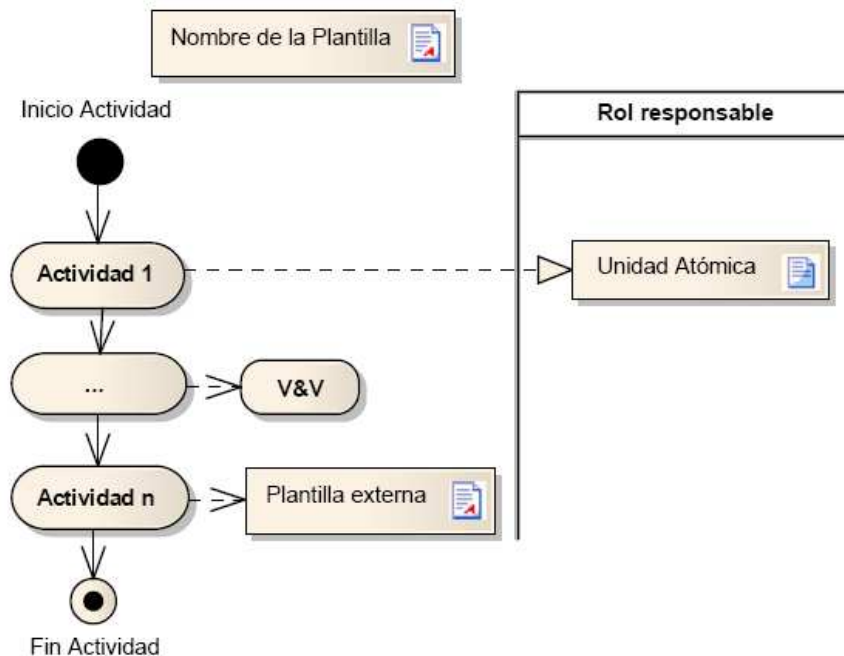


Figura 3.3 Diagrama de unidades atómicas

CAPÍTULO 4 Plantillas de Administración del Proyecto

El primer proceso concierne a la Administración del Proyecto (AP), y es el encargado de establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos del proyecto en tiempo y costo.

Las actividades de Administración de Proyectos, generaron un total de cuatro plantillas, tres de ellas son productos que se generan varias veces a lo largo del proceso:

- la *Matriz de Trazabilidad*,
- el *Documento de Aceptación*
- el *Reporte de Seguimiento* y
- el *Plan del Proyecto*

Por esta razón se presenta una estandarización de esos documentos en la que se incluyen los elementos específicos de cada actividad que, de acuerdo a PMBoK se consideran buenas prácticas.

La cuarta plantilla, *Plan del Proyecto*, describe la información relativa al proyecto en el que se incluyen y documentan técnicas de diferentes áreas del conocimiento del PMBoK.

El llenado de las plantillas de Administración de Proyectos, buscó ser lo más intuitivo posible, debido a que son productos que están presentes varias veces a lo largo del proceso, ver Figura 4.1.

16. Estrategia de control de versiones

[Método para administrar la configuración del sistema, se debe incluir un mecanismo para solicitar cambios así como la estructura y reglas de uso del repositorio.]

Estrategia del control de versiones															
Identificación de la configuración	<i>[establecer lo que será almacenado en el repositorio, suministrar la línea base para clasificar productos y documentos]</i>														
Estado de la configuración	<i>[información de los productos que están en el repositorio, listando los productos almacenados, su ubicación y el estado en que se encuentran]</i>														
Auditor	<i>[individuo quién verificará la existencia y estado de los productos que se establecieron generar. El auditor deberá corroborar cada auditoría que se ha cumplido con la ejecución de los productos]</i>														
Frecuencia de auditorías	<i>[establecer el plazo en que se efectuarán las auditorías]</i>														
Cambios solicitados	<table border="1"> <thead> <tr> <th>Nombre del producto</th> <th>Descripción</th> <th>Impacto</th> <th>Solicitante</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>					Nombre del producto	Descripción	Impacto	Solicitante	Estado					
	Nombre del producto	Descripción	Impacto	Solicitante	Estado										
<i>[listado de los cambios solicitados a lo largo del proceso]</i>															
Políticas	<i>[acuerdos para el uso del repositorio, establecer quién puede almacenar, modificar y acceder al repositorio]</i>														

Figura 4.1 Fragmento de una unidad atómica del Plan del Proyecto



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

En COMPETISOFT la Administración de Proyectos se define como *aquel proceso que se compone de una serie de actividades, tiene una duración determinada, se realiza con recursos humanos y técnicos limitados, con un costo asignado y cuyo fin es cumplir los objetivos marcados por el cliente en tiempo, costo y forma* [Oktaba 2007]. Para este grupo de procesos se definen cuatro objetivos fundamentales:

- O1 Lograr los *Objetivos* del proyecto en tiempo y costo mediante la coordinación y el manejo de los recursos del mismo.
- O2 Mantener informado al Cliente mediante la realización de reuniones de avance del proyecto.
- O3 Atender las *Solicitudes de Cambio al desarrollo* o *Peticiones de Modificación* para el mantenimiento mediante la recepción y análisis de las mismas.
- O4 Establecer y mantener la integridad de los productos de trabajo del proyecto y hacerlos disponibles al Equipo de Trabajo.

Todo proyecto debe pasar por las fases de Inicio, Planeación, Ejecución, Control y Cierre, cada una de estas fases se detalla de la siguiente manera, ver Figura 4.2:

- **Fase de Inicio:** tiene como finalidad definir y autorizar el proyecto o una fase del mismo.
- **Fase de Planeación:** define y refina los objetivos, planifica el curso de acción requerido para lograr los objetivos y el alcance pretendido por el proyecto.
- **Fase de Ejecución:** integra a personas y otros recursos para llevar a cabo el *Plan del Proyecto*.
- **Fase de Control:** mide y supervisa regularmente el avance, a fin de identificar las variaciones respecto al *Plan del Proyecto*, de tal forma que se tomen medidas correctivas cuando sea necesario para cumplir con los objetivos del proyecto.
- **Fase de Cierre:** formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto o una fase del mismo.

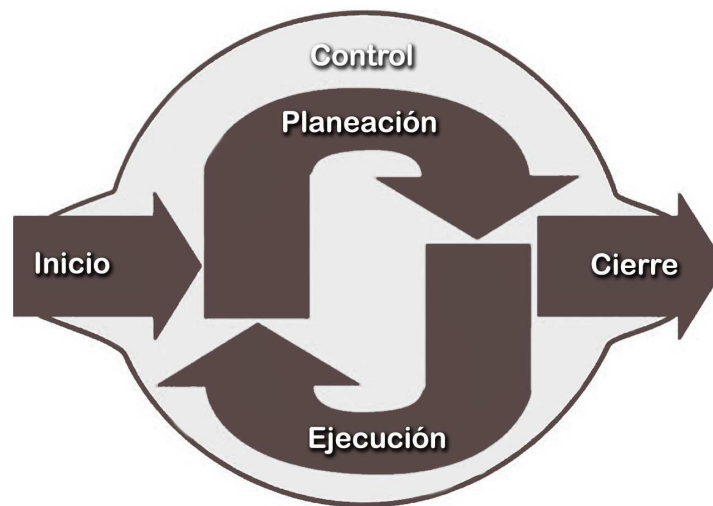


Figura 4.2 Fases de la Administración de Proyectos

Los productos requeridos por el grupo de procesos de Administración del Proyecto, se engloban en las plantillas de *Plan del Proyecto* que es generado durante la fase de Planeación y puede sufrir modificaciones durante la Ejecución. Además las plantillas de *Documento de Aceptación*, *Matriz de Trazabilidad* y la de *Reporte de Seguimiento* que pueden ser utilizadas durante la fase de Control.

4.1 Plan del Proyecto

De acuerdo a COMPETISOFT, el Plan del Proyecto es el producto más importante de la Administración del Proyecto, es el documento formal usado como guía para la ejecución y control del proyecto. En él se concentran elementos como la Descripción del Producto y tipo de proyecto, el Alcance, los Objetivos, Entregables, Ciclos y Actividades, el Tiempo y Costo Estimado y el Equipo de Trabajo.

Comenzando por la unidad atómica de la *Descripción del producto* en donde se deben establecer puntos como la Perspectiva del producto, las Funciones del Producto, Supuestos del Producto. Además de aspectos como Licenciamiento, Instalación y Criterios de Calidad.

La descripción del *Alcance* es en donde se establecerán los límites del proyecto. En los *Objetivos* se clarificarán aquellas metas que se desean alcance el proyecto por una parte y el producto por otra. Los Objetivos del Proyecto son aquéllos que se desean alcanzar en conjunto, aquellas metas que de antemano se sabe que de no trabajar en equipo difícilmente lograrían. Estos objetivos proporcionan una base de lanzamiento para el proyecto.

Por *Objetivos del Producto* se consideran todas aquellas metas que debe cumplir el producto terminado. Van más encaminados a cuestiones de funcionalidad del sistema desde el punto de vista del usuario final, cliente y por supuesto de los realizadores.

La unidad atómica *Entregables* contiene un listado de todos los productos a ser entregados a lo largo del proyecto, además de definir un *Protocolo de Entrega*, que debe incluir un mecanismo aceptado para la entrega de productos descritos en *Entregables*. En *Restricciones* se establecen las restricciones y limitaciones presentes para la realización del proyecto.

Para definir el *Proceso Específico* es necesario determinar a los grupos de actividades que se implementarán a lo largo del proyecto. Ligado a esta elección, se encuentra la definición de los *Ciclos y Actividades*, en donde se establecerá el número de ciclos necesarios para llevar a cabo todas y cada una de las actividades identificadas en el modelo, en este caso, el contenido del *Proceso Específico y Actividades* todo está claramente establecido en el modelo de referencia de COMPETISOFT, por lo que únicamente se debe definir el número de ciclos necesarios para llevar a cabo el desarrollo.

En cuanto al *Tiempo Estimado*, de acuerdo a las actividades identificadas se debe establecer su duración y las dependencias entre éstas y el esfuerzo necesario para completar cada una de ellas, con esa información se podrá dar un valor estimado de tiempo.

El *Plan de Adquisiciones y Capacitación*, en COMPETISOFT es considerado como un documento aparte al *Plan del Proyecto*, en esta propuesta se decidió anexarlo al documento del *Plan del Proyecto* con la finalidad de incluir todo lo relacionado con la planeación en un solo documento, además de que está altamente ligado con aspectos de costo, tiempo y alcance. Esta unidad atómica está compuesta por tres *subplantillas* para las

Adquisiciones, las Contrataciones y las Capacitaciones.

El *Equipo de Trabajo* es un listado que incluye a cada uno de los miembros participantes en el proyecto. En el *Calendario* se establecen gráficamente las dependencias entre actividades, su duración y fechas de inicio y término.

El *Costo* es donde se establece el valor monetario resultante de la realización del proyecto.

En *Riesgos* se identifican y documentan los posibles riesgos que pudieran impactar el desarrollo normal del proyecto, aspectos como probabilidad de ocurrencia, impacto del riesgo, plan de contingencia y estado del riesgo.

La unidad atómica de *Estrategia de Control de Versiones* define un mecanismo para administrar la configuración del sistema, se debe incluir un mecanismo para solicitar cambios así como la estructura y reglas de uso del repositorio. Además de considerar políticas de uso y auditorías.

Por último el *Plan de Comunicación* es en donde se mencionan y describen los mecanismos de comunicación válidos entre los involucrados a lo largo del proyecto. Así como definir el formato para los comunicados, su frecuencia, finalidad, plazo y periodos de validez.

A continuación se resume en un listado a las unidades atómicas que se construyeron de acuerdo a lo establecido en el PMBoK:

- Introducción
- Descripción del producto
- Alcance
- Objetivos
- Entregables
- Restricciones
- Proceso específico
- Protocolo de entrega
- Ciclos y actividades
- Tiempo estimado
- Plan de adquisiciones y capacitación
- Equipo de trabajo
- Calendario
- Costo
- Riesgos
- Estrategia de control de versiones
- Plan de comunicación
- Justificación del plan del proyecto
- Información de apoyo

En la Figura 4.3, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Plan del Proyecto.

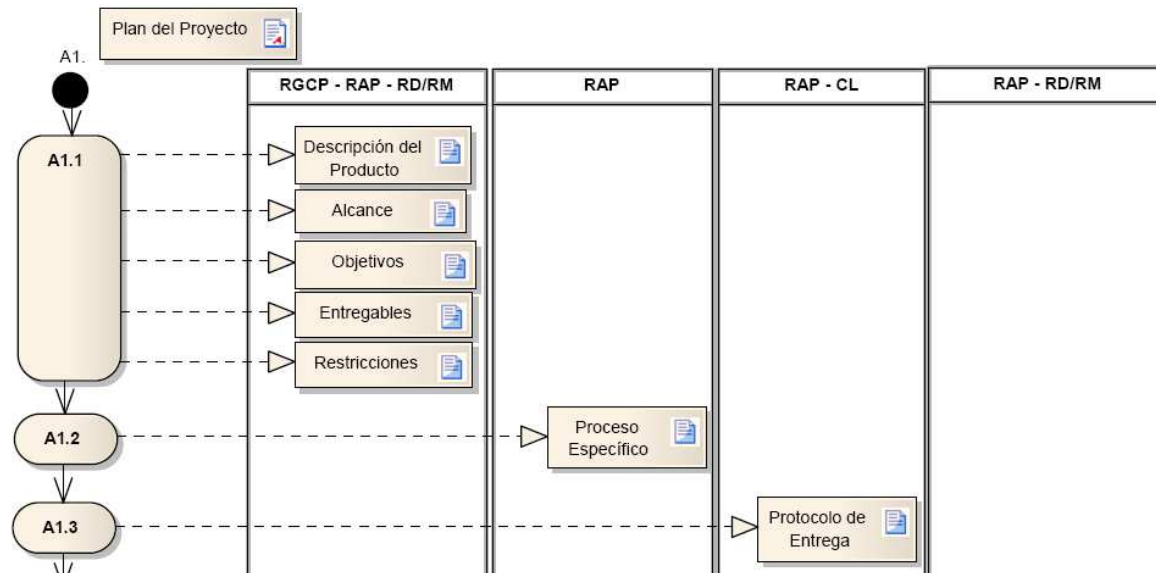


Figura 4.3 Fragmento del Diagrama de Unidades Atómicas del Plan del Proyecto

4.2 Documento de Aceptación

El formato establecido para el *Documento de Aceptación* varía de acuerdo con lo visto en el *Plan del Proyecto*, esto es consecuencia directa de la cantidad de veces que es generado cada uno a lo largo del proceso. Como el *Plan del Proyecto* es generado en una sola ocasión y únicamente sufre modificaciones, se propone una estructura más rígida, mientras que el *Documento de Aceptación* aparece en más de una ocasión por lo que su estructura y llenado buscan ser puntuales y breves para facilitar su diligencia.

Como unidades atómicas de esta plantilla tenemos las siguientes:

- Nombre del documento/proyecto
- Descripción
- Elaborado por
- Última revisión
- Versión
- Válido
- Cambios sobre el documento
- Estado
- Autorización

4.3 Matriz de Trazabilidad

La *Matriz de Trazabilidad* es un documento que se genera una sola ocasión pero que es altamente modificable. Para su llenado es necesario que cada elemento tenga un identificador y nombre únicos con el objeto de facilitar su identificación y mantener una consistencia a lo largo del desarrollo.

La trazabilidad será impuesta en dos direcciones, hacia atrás y hacia adelante, la trazabilidad hacia atrás identifica al elemento que origina la necesidad del elemento actual, y hacia adelante implica qué elemento cubrirá las necesidades originadas por el elemento actual.

Esta plantilla se basa principalmente en aspectos del Proceso Unificado para establecer la trazabilidad entre los componentes. En primera instancia se identificaron todos los elementos que de acuerdo al Proceso Unificado son trazables, posteriormente se definió una estructura para poder establecer la trazabilidad de los elementos hacia atrás y hacia delante de manera clara y sin ambigüedades, ver Figura 4.4.

Las unidades atómicas de esta plantilla son:

- Descripción General
- Matriz de Trazabilidad
- Información de Apoyo



AP Matriz de Trazabilidad

Matriz de Trazabilidad

1. Descripción general

[La matriz de trazabilidad resulta de establecer asociaciones entre los requisitos establecidos en la ERS con los productos que se generan a través del proceso para satisfacerlos.]

2. Matriz de trazabilidad

[Matriz propuesta para establecer las relaciones entre un requerimiento establecido por el cliente y un caso de uso abstraído por el equipo de trabajo.]

Identificador	Elemento	Trazabilidad hacia adelante
<i>[Identificador del elemento.]</i>	<i>[Descripción del requerimiento establecido por el cliente.]</i>	<i>[Identificador del elemento que satisface el requerimiento.]</i>

[Matriz propuesta para establecer las relaciones entre los casos de uso y el plan de pruebas del caso de uso]

Figura 4.4 Fragmento de la Matriz de Trazabilidad

4.4 Reporte de Seguimiento

El *Reporte de Seguimiento* debe realizarse y presentarse periódicamente para tener un control adecuado del proyecto. Las unidades atómicas de esta plantilla son:

- Introducción
- Descripción General
- Reporte de Actividades
- Reporte de Mediciones y Sugerencias de Mejora
- Lecciones Aprendidas
- Información de Apoyo

La unidad atómica fundamental de esta plantilla es la de *Reporte de Actividades*, compuesta por cuatro secciones:

Actividades, que es donde se listan todas las actividades realizadas por el equipo de trabajo durante el periodo de tiempo reportado, se debe incluir para cada actividad reportada, ver Figura 4.5:

- Tiempo estimado y real que tomó realizarla
- Fecha estimada y real en que se desarrolló
- Tamaño y Estado del producto obtenido si es que dicha actividad genera alguno

Cambios, en ella se realiza el control de cambios efectuados a lo largo del periodo que reporta. Este seguimiento facilita el manejo del repositorio, así que los cambios reportados deben ser consistentes con el estado de la configuración. En esta sección, por cada cambio se registra:

- Nombre del producto
- Descripción del cambio
- Impacto ocasionado si se realiza
- Solicitante y
- Estado del cambio

Riesgos, es la tercera sección, en ella se registran los riesgos nuevos identificados o aquéllos que ocurrieron durante el periodo de tiempo reportado. Se debe proveer la descripción del riesgo, plan de contingencia, el impacto del riesgo y su estado.

Por último se debe agregar el *Resumen* de condensar las mediciones tomadas en el periodo de tiempo que abarca el seguimiento, para ello se proponen tablas que incluyan la información más relevante de una manera breve y clara, principalmente Productos, Cambios y Riesgos.

Para completar la plantilla, se proveen dos unidades atómicas más:

- *Reporte de Mediciones y Sugerencias de Mejora*
- *Lecciones Aprendidas*

En las *Lecciones Aprendidas* estarán englobadas las lecciones experimentadas por el equipo tanto individual como colectivamente. Aquellas prácticas que el equipo considera que beneficiaron el desempeño o que resultaron en una experiencia exitosa.

El *Reporte de Mediciones y Sugerencias de Mejora* incluye el cálculo de los minutos invertidos en cada fase así como el tiempo total del ciclo, el tamaño de los productos generados en cada fase y el total de productos elaborados. Otro aspecto importante que se reporta es el control y registro de los defectos encontrados, si éstos han sido corregidos o no y el estado del producto.

Las *Sugerencias de Mejora* identifican las mejores prácticas en los rubros del equipo y el proceso. Los problemas recurrentes tanto en el funcionamiento del equipo como en el proceso. Las experiencias exitosas y las reflexiones sobre lo que habrá que mejorar tanto al proceso como al equipo.

2. Descripción general

[En esta sección se describen de manera general los factores que afectaron el proceso así como información relevante de lo ocurrido.]

2.1 Periodo

[Indicar de manera clara el periodo de tiempo que abarca este reporte. De dd-mm-aa a dd-mm-aa.]

3. Reporte de actividades

[Esta unidad atómica debe realizarse y presentarse periódicamente, es recomendable que sea de manera semanal.]

3.1 Actividades

[Listado de todas las actividades realizadas por el equipo durante el periodo de tiempo reportado.]

Actividad	Tiempo			Fecha		Tamaño del producto	Estado del producto
	Estimado	Real	Desviación	Estimada	Real		
<i>[nombre de la actividad]</i>	0:00	0:00	0.00%	<i>[dd-mm-aa]</i>	<i>[dd-mm-aa]</i>	0	<i>[Terminada/En proceso/Retrasada]</i>
...							
Totales	00:00	00:00	0.00%			0 y 0 LOC	

Figura 4.5 Fragmento del Reporte de Seguimiento

CAPÍTULO 5 Plantillas de Desarrollo de Software

El proceso de Desarrollo de Software (DSW) propuesto en el modelo de referencia de COMPETISOFT establece un ciclo de vida de software iterativo e incremental, que se llevará a cabo con técnicas y estándares orientados a objetos. El desarrollo es dividido en fases, ver Figura 5.1, el descomponer en ciclos de desarrollo permite identificar riesgos en fases tempranas, incluir en las primeras etapas los requerimientos más importantes del sistema, permitiendo un refinamiento de los mismos en ciclos posteriores, validando la implementación con el cliente. Además como meta en cada ciclo se tiene un producto funcional, lo que implica proveer al cliente una muestra del resultado final del producto, con ello se incrementa la participación del cliente y sobre todo permite que el cliente se dé cuenta de lo que si quiere y de lo que no, de lo que falta y lo que excede sus necesidades.

La decisión de cuántos ciclos de desarrollo serán necesarios estará dirigida por los requerimientos identificados. Es al término de cada uno donde el desarrollo iterativo e incremental permite una visualización clara al equipo de trabajo y al cliente de lo que se ha hecho y se hará.



Figura 5.1 Fases del ciclo de vida del software

Los productos requeridos por el grupo de procesos de Desarrollo de Software, se engloban en una plantilla por cada fase propuesta por COMPETISOFT. Siendo éstas:

- Especificación de Requerimientos
- Análisis
- Diseño
- Construcción
- Integración
- Pruebas
- Cierre

Que corresponden cada una a los flujos de trabajo establecidos en el Proceso Unificado.



Universidad Nacional
Autónoma de México



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 esta 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.

5.1 Especificación de Requerimientos

En la Especificación de Requerimientos es en la cual se incluye la unidad atómica específica para modelar los requerimientos mediante la técnica de casos de uso, tal como lo indica el Proceso Unificado, siendo la parte más importante del documento. A partir de la identificación, captura y priorización de cada requerimiento se comienza con la trazabilidad a lo largo del proceso según lo propone el Proceso Unificado.

Dentro del mismo documento se deben incluir el *Plan de prueba del sistema*, que no es considerado en el Proceso Unificado pero si por el modelo de referencia, y el *Prototipo de la interfaz de usuario* para cubrir cada caso de uso requerido.

Las unidades atómicas que completan la plantilla de Especificación de Requerimientos son *Requerimientos no funcionales* que contiene todas las exigencias de software no funcionales, a un nivel de detalle lo suficientemente claro para que los miembros del equipo de desarrollo puedan satisfacerlos. Además se enumeran aquellos requerimientos más comunes con una descripción de los mismos, para facilitar la identificación por parte de los desarrolladores. La usabilidad, confiabilidad, eficiencia, mantenimiento, restricciones de diseño y construcción, reusabilidad, interfaces y licenciamiento.

Como ya se mencionó, el componente principal de la plantilla es *Requerimientos funcionales*, en donde se identifican, priorizan y detallan los casos de uso, mismos que son trazables a lo largo de todo el proceso de desarrollo, ver Figura 5.2.

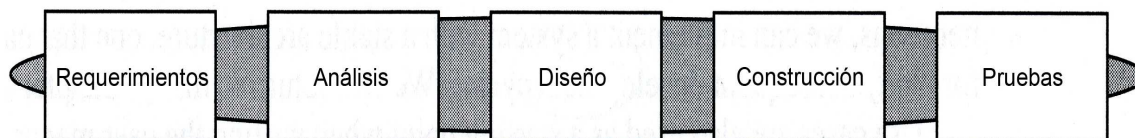


Figura 5.2 Trazabilidad de los casos de uso

El *Plan de pruebas del sistema* está relacionado estrictamente con los casos de uso identificados, de acuerdo con lo establecido en el Proceso Unificado. Se agregan atributos como son la definición de requerimientos y estrategia de las pruebas, herramientas y recursos necesarios para llevarlas a cabo y los casos de prueba que cubran datos válidos, inválidos y las reglas del negocio. Resumen de unidades atómicas creadas para la plantilla *Especificación de Requerimientos* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Requisitos No Funcionales
- Requisitos Funcionales
- Plan de Pruebas del Sistema
- Justificación de los requerimientos del software
- Información de apoyo

En la Figura 5.3, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Especificación de Requerimientos.

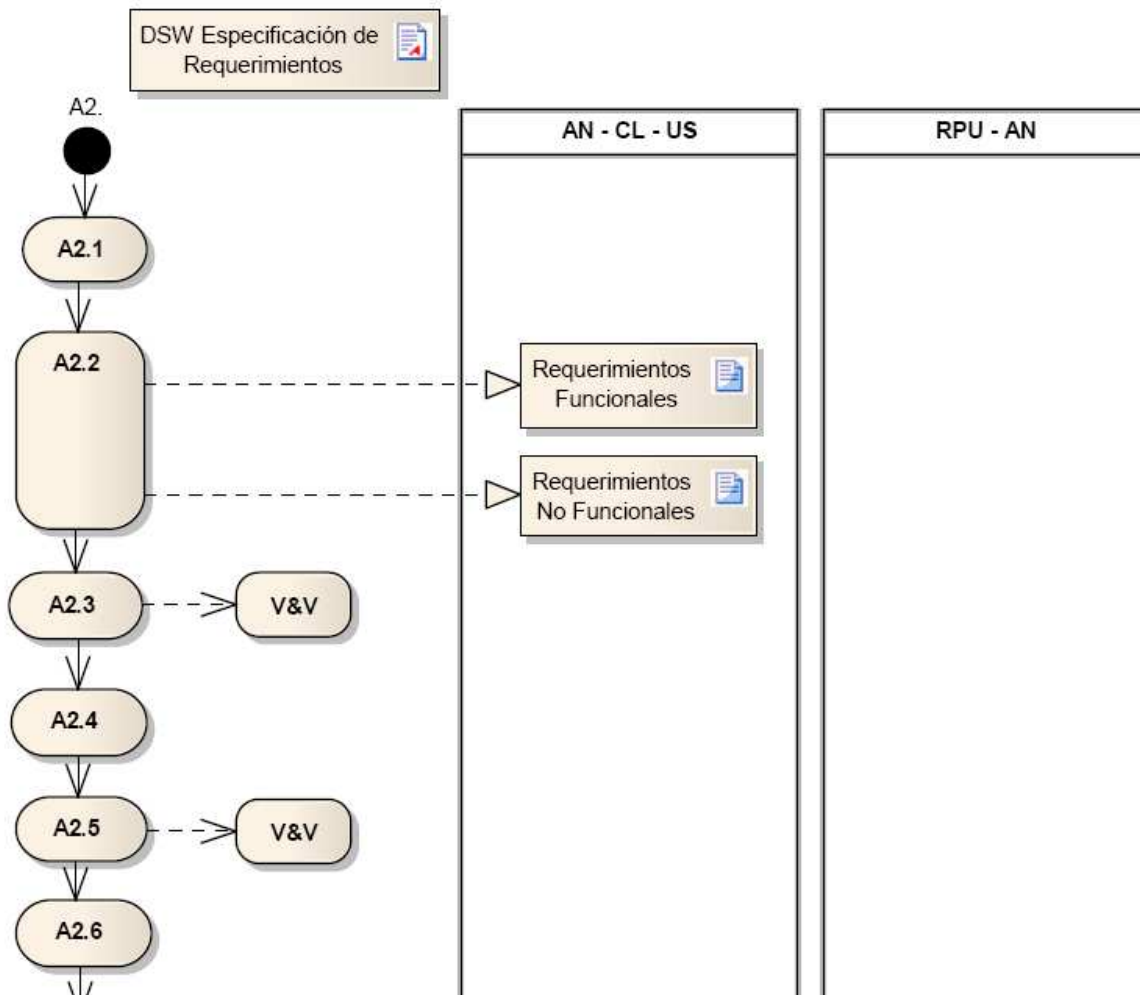


Figura 5.3 Fragmento del Diagrama de Unidades Atómicas de la Especificación de Requerimientos

5.2 Análisis

El Análisis y el Diseño son consideradas en COMPETISOFT y el Proceso Unificado como dos fases, mientras que de acuerdo a RUP son consideradas como una sola, a pesar de ello son claramente diferenciables las actividades pertenecientes a cada una de las dos fases. Mientras que en la fase de Análisis lo que se busca es describir los requerimientos y objetivos de software que tienen algún impacto significativo en la arquitectura, en el Diseño se ofrece un panorama de la arquitectura del sistema, con la intención de captar y transmitir las decisiones importantes de arquitectura tomadas para satisfacer los aspectos identificados en el Análisis.

Como se puede notar, la definición y representación de la arquitectura del sistema toma importancia debido a que el Proceso Unificado propone un desarrollo centrado en la arquitectura. En la plantilla de Análisis la unidad atómica de *Representación de la arquitectura* contiene información acerca de sus metas y restricciones. Por una parte en las metas se deben englobar los objetivos esperados que tienen que ver con la arquitectura del sistema, este punto debe contener los suficientes puntos de apoyo para sostener una

discusión, y posteriormente durante la fase de Diseño una elección de la arquitectura a utilizar, mientras que en las restricciones se deben incluir aquellas limitaciones que enfrente el sistema e impacten en la arquitectura del mismo.

La plantilla de Análisis requiere de un modelado a un nivel adecuado, esto se obtiene realizando un *Modelo del análisis* orientado a objetos en el que se represente al conjunto de paquetes de más alto nivel identificados a partir de los requerimientos obtenidos en la Especificación de Requerimientos, representando las abstracciones en subsistemas o capas del sistema. Posteriormente en el desarrollo orientado a objetos se identificarán las *Clases del análisis* necesarias para que el sistema cumpla con los requerimientos establecidos, para finalmente obtener el comportamiento estático del sistema y los tipos de relaciones existentes entre las clases.

Por último, para completar las unidades atómicas específicas de la plantilla de Análisis, se requiere de un modelado dinámico del sistema mismo que es posible plasmar con la unidad atómica *Realización de caso de uso – Análisis*, en la que se requieren los diagramas de Interacción y de Navegación.

Resumen de unidades atómicas creadas para la plantilla *Análisis* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Representación de la Arquitectura
- Modelo del Análisis
- Clases del Análisis
- Realización de Caso de Uso-Análisis
- Justificación de las decisiones del Análisis
- Información de apoyo

En la Figura 5.4, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Análisis.

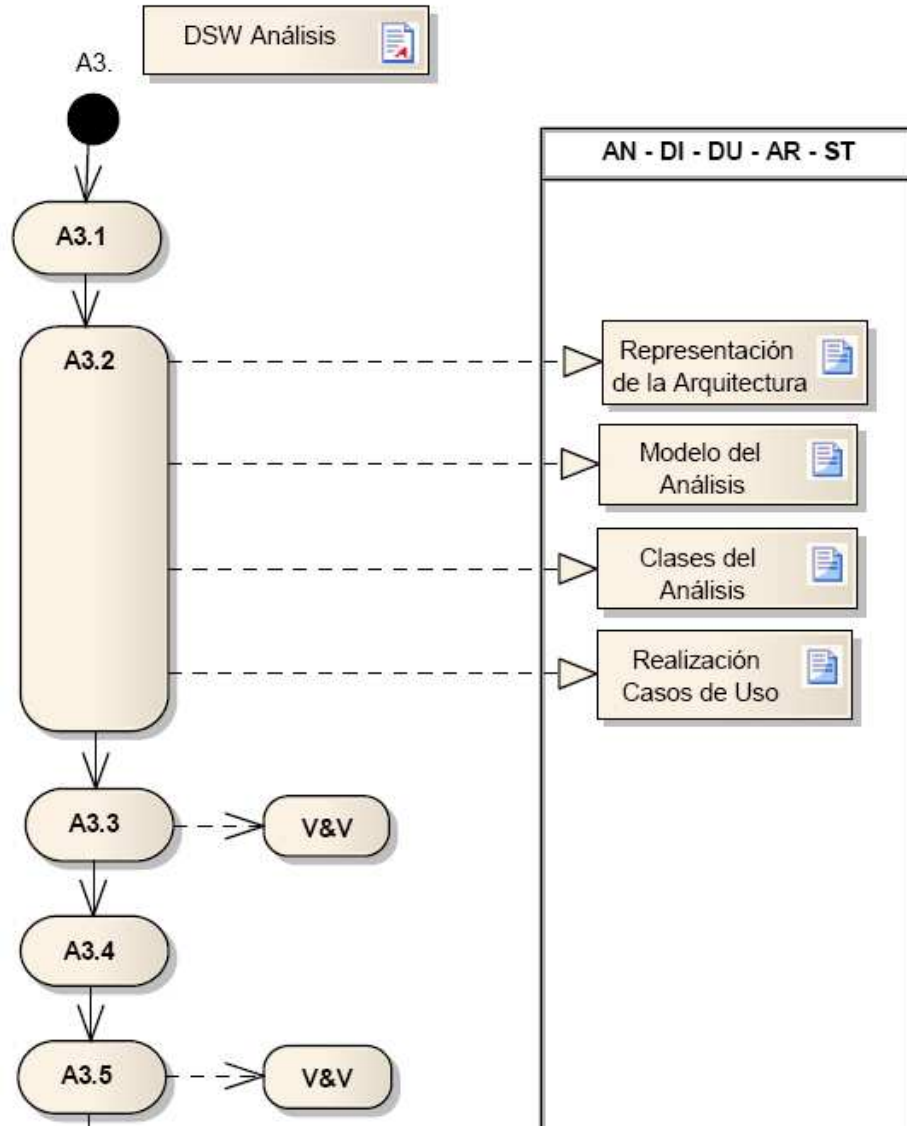


Figura 5.4 Fragmento del Diagrama de Unidades Atómicas del Análisis

5.3 Diseño

La fase de Diseño es en donde se da forma al sistema y se define una arquitectura que cubra los requerimientos, tanto funcionales como no funcionales. De acuerdo al Proceso Unificado, en el flujo del diseño se adquiere un entendimiento más profundo de los requerimientos no funcionales, las restricciones de los lenguajes de programación, reutilización de componentes y sistemas operativos, entre otras cosas [Jacobson]. El Diseño es desarrollado al final de la etapa de Elaboración y al inicio de la Construcción, por lo que resulta el puente que contribuye a darle solidez y estabilidad al paso hacia la construcción.

La unidad atómica *Representación de la Arquitectura*, es en donde se define aquella arquitectura que cumpla y cubra con las metas y restricciones establecidas en el Análisis. La *Vista Lógica*, se compone de la identificación de los paquetes del diseño,

descomponiendo el modelo por capas y asignándoles cierta jerarquía, posteriormente se identifican las clases del diseño, junto con sus atributos y métodos, agrupándolas en cada uno de los paquetes previamente identificados. Finalmente se obtiene un *Diagrama de Paquetes de la Arquitectura* en el que es posible identificar los aspectos mencionados anteriormente.

En cuanto a las *Vistas del Proceso, de Despliegue y de Datos*, la primera describe la descomposición del sistema en procesos y grupos de procesos. Los organiza de acuerdo a las relaciones de interacción que existen entre ellos. Mientras que la *Vista de Despliegue* debe describir la configuración de la red física del software para su ejecución, indica los nodos, además de las interconexiones entre éstos, que son necesarios para el funcionamiento correcto del software. Debe incluirse el *Diagrama de Despliegue* para clarificar la distribución de la arquitectura.

En la *Vista de Datos*, se provee una descripción del mecanismo de almacenamiento de los datos, así como sus supuestos y restricciones para poder realizar en fases posteriores la construcción de la base de datos de ser necesaria. Como complemento, en la sección 8.3.2, se presenta a manera de ejemplo la generación de unidades atómicas específicas para desarrollos centrados en bases de datos, de acuerdo a las prácticas y guías de bases de datos propuestas en [García-García].

Para el *Plan de Pruebas de Integración*, su unidad atómica correspondiente está guiada por los puntos que marca el Proceso Unificado para realizar “pruebas no contempladas”, por lo que resulta esta unidad atómica en una adaptación de éstas enfocadas en la integración del sistema, fundamental para la fase de Integración propuesta en COMPETISOFT.

Primeramente se debe proveer una *Descripción del tipo de prueba*, y de cómo se planea realizar la integración. Se incluye un *Listado de los subsistemas* a integrar, así como los *Requerimientos* para realizar las pruebas y la *Estrategia* elegida. En la *Estrategia* se debe incluir un Objetivo, Técnica, un Criterio de Terminación y Consideraciones especiales de las pruebas.

Posteriormente se debe proveer un listado de las Herramientas y Recursos necesarios, y para finalizar se deben incluir los casos de prueba para cada iteración o incremento de las pruebas.

Resumen de unidades atómicas creadas para la plantilla *Diseño* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Representación de la Arquitectura
- Vista Lógica
- Vista del Proceso
- Vista de Despliegue
- Vista de Datos
- Plan de Pruebas de Integración
- Justificación de las decisiones del Diseño
- Información de apoyo

En la Figura 5.5, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Diseño.

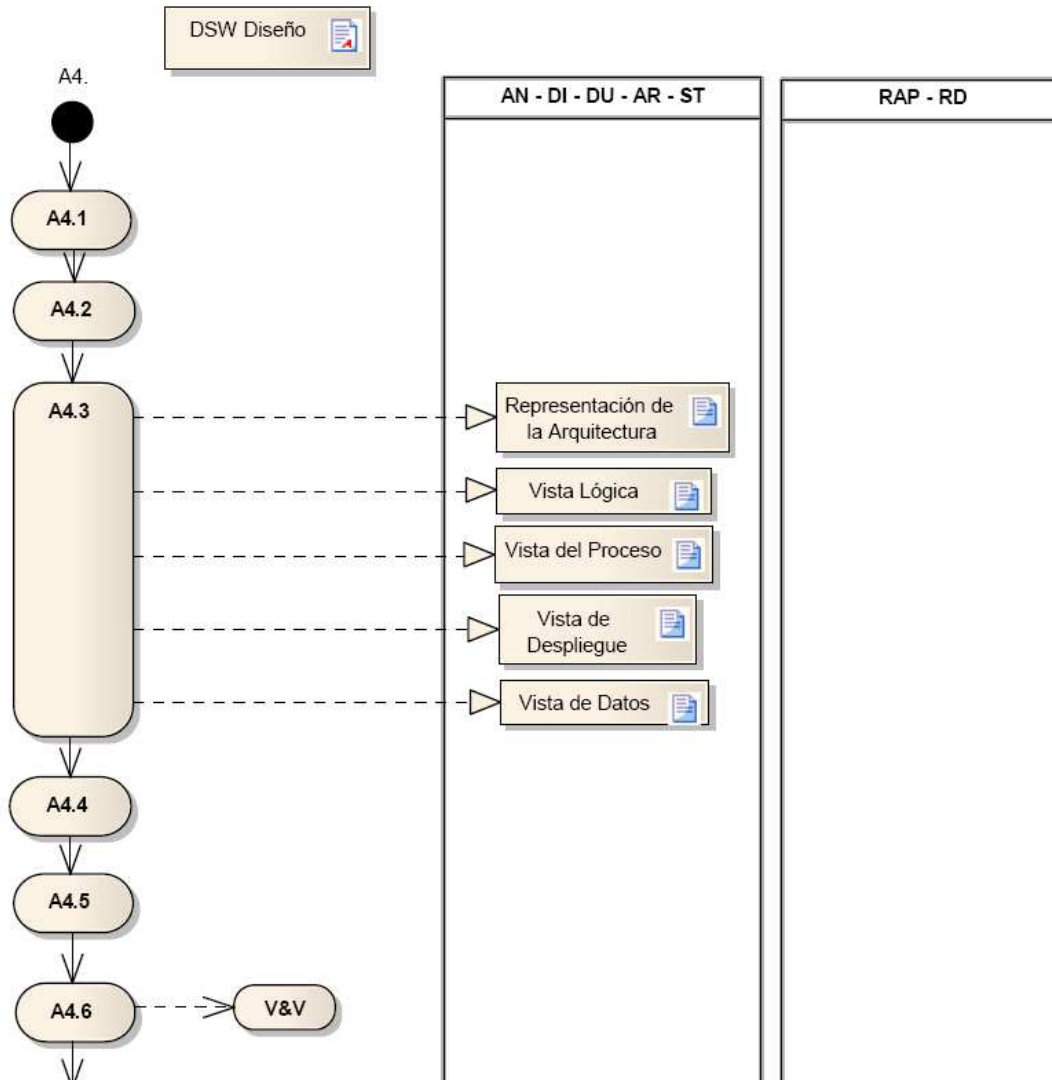


Figura 5.5 Fragmento del Diagrama de Unidades Atómicas del Diseño

5.4 Construcción

La fase de *Construcción* en COMPETISOFT, o el flujo de trabajo de *Implementación* de acuerdo al Proceso Unificado según [Jacobson], cuando se comienza la implementación del sistema en términos de componentes, como lo son, código fuente, scripts, binarios y ejecutables entre otras cosas.

Los *Componentes* son elementos fundamentales de la fase de Construcción. Un componente se puede definir como los paquetes físicos de los elementos modelados [UML]. Cada componente tiene una traza a algún elemento del modelo que implementa, es por ello que la unidad atómica principal de la plantilla asociada a esta fase es la llamada de igual forma *Componentes*, esta parte contiene el listado de los componentes que se necesitan implementar totalmente, es decir, aquellos componentes que resultan nuevos objetivos esperados. Además de incluir un listado de los componentes que se necesitan modificar, es decir, aquellos componentes existentes en la configuración y que requieren de modificaciones para su posterior integración al sistema.

La unidad atómica de *Plan de Pruebas Unitarias*, al ser un desarrollo orientado a objetos, presenta una estructura que servirá como patrón para establecer los casos de prueba de cada uno de los componentes. Se consideran dos tipos de pruebas, por una parte las Pruebas de Caja Blanca que se efectuarán sobre cada una de las clases a un bajo nivel, es decir se verificará cada método de las clases, qué tipo de dato recibe y regresa, así como las excepciones cachadas.

Por otro lado las pruebas de Caja Negra, a un nivel más alto, sólo se preocupan por verificar las entradas y salidas del sistema y comparando los valores esperados contra los recibidos.

Esta unidad atómica también considera una *Descripción de las pruebas*, los Requerimientos de las pruebas donde se debe incluir el listado de los requerimientos necesarios para obtener un ambiente que sea adecuado para realizar las pruebas y considerarlas válidas. En la *Estrategia de las Pruebas* se incluye el objetivo, técnicas, criterios de terminación y consideraciones especiales. Para completarla se especifican también a las *Herramientas, Recursos y Casos de Prueba*.

En esta misma fase se tienen que efectuar las pruebas y reportar los resultados, es por ello que la plantilla también incluye la unidad atómica de *Reporte de Pruebas Unitarias*, en ella se deben reportar los defectos encontrados y los valores recibidos al ejecutar cada caso de prueba.

Resumen de unidades atómicas creadas para la plantilla *Construcción* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Componentes
- Plan de Pruebas Unitarias
- Reporte de Pruebas Unitarias
- Justificación de las decisiones de la Construcción
- Información de apoyo

En la Figura 5.6, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Construcción.

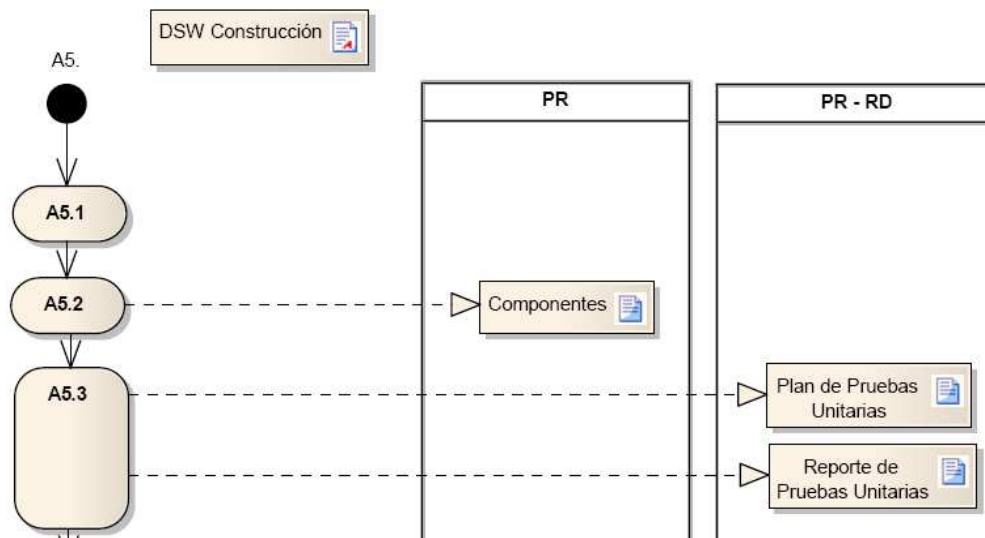


Figura 5.6 Fragmento del Diagrama de Unidades Atómicas de la Construcción

5.5 Integración

Durante la fase de Integración propuesta en el modelo de referencia se solicita efectuar la integración de los módulos o componentes que previamente han sido verificados mediante la ejecución de las pruebas unitarias. Es por ello que la plantilla dedicada para cubrir las actividades de esta fase está encargada de reportar los resultados de efectuar el *Plan de Pruebas de Integración*.

La unidad atómica *Reporte de Pruebas de Integración* está conformada únicamente por el reporte de cada iteración o incremento de la integración de los componentes y el reporte de los defectos encontrados.

Se utilizan incrementos o iteraciones debido a que es lo propuesto por el Proceso Unificado como técnica para elaborar este plan de pruebas.

Resumen de unidades atómicas creadas para la plantilla *Integración* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Reporte de Pruebas de Integración
- Justificación de las decisiones de la Integración
- Información de apoyo

En la Figura 5.7, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Integración.

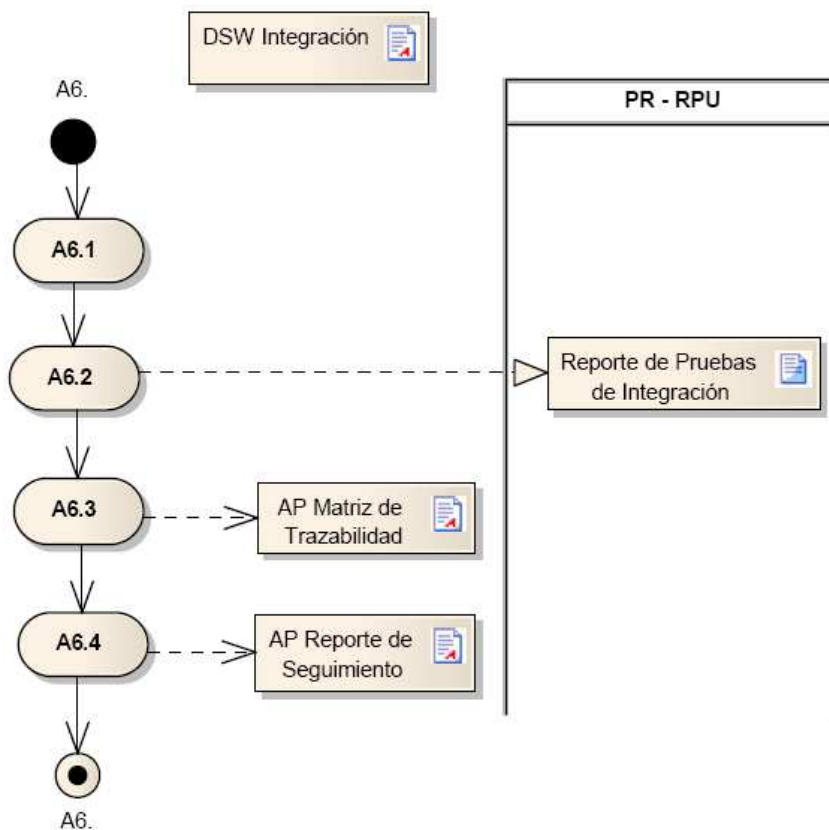


Figura 5.7 Diagrama de Unidades Atómicas de la Integración

5.6 Pruebas

La fase de Pruebas es en donde se realizará lo especificado en el *Plan de pruebas del sistema* establecido en la fase de Especificación de Requerimientos, y aquellas no contempladas en el desarrollo orientado a objetos pero que son requeridas.

La unidad atómica *Requerimientos de las Pruebas* es en donde se especifican los requerimientos generales de las pruebas. Además de las *Herramientas y Recursos* necesarios.

En la *Estrategia de las Pruebas* primeramente se presenta una tabla para establecer los Objetivos, Técnica, Criterio de Terminación, Consideraciones especiales, Casos de Prueba y Responsable para cada uno de los tipos de prueba no considerados. También se incluye un listado con una amplia variedad de pruebas propuestas por RUP y que pudieran ser necesarias para completar el desarrollo orientado a objetos.

Resumen de unidades atómicas creadas para la plantilla *Integración* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Requerimientos de las Pruebas
- Estrategia de las Pruebas
- Herramientas
- Recursos
- Justificación de las decisiones de las Pruebas
- Información de apoyo

En la Figura 5.8, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Pruebas.

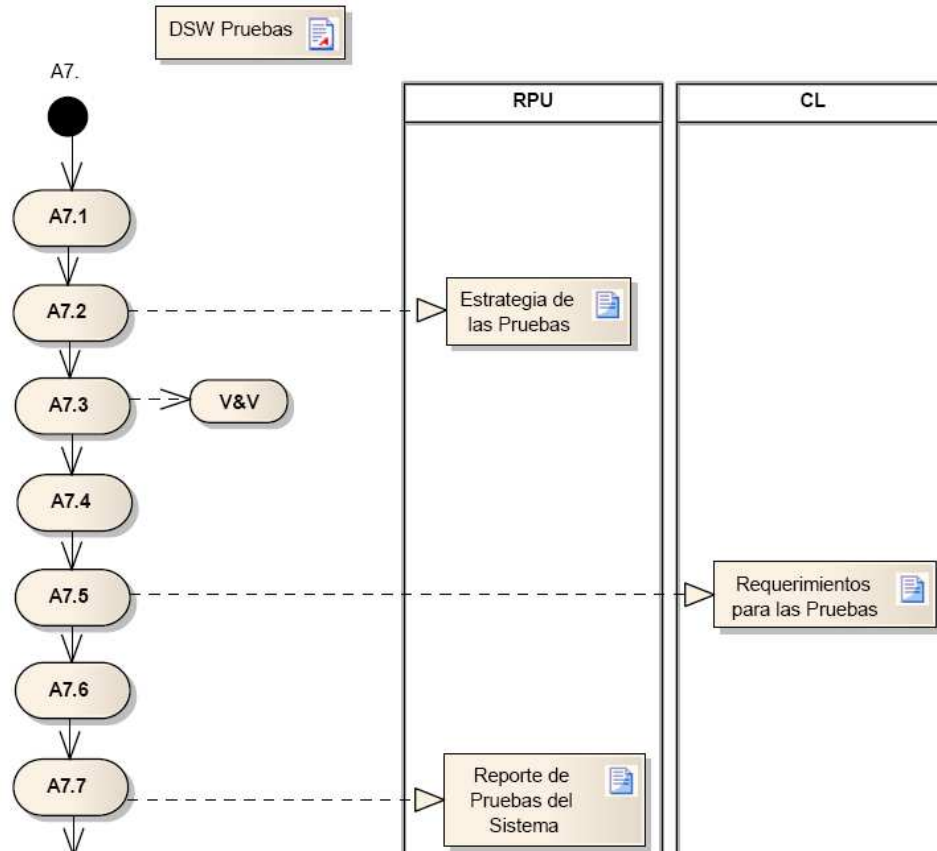


Figura 5.8 Fragmento del Diagrama de Unidades Atómicas de la Pruebas

5.7 Cierre

Durante la fase de Cierre se realiza el cierre administrativo del ciclo, además, se efectúa una recopilación de los seguimientos y mediciones que a lo largo del ciclo de desarrollo orientado a objetos se obtuvieron.

Entre las modificaciones hechas en este trabajo, en orden de incrementar la utilidad y facilitar la adopción del modelo, se decidió incluir hasta esta fase la generación de los Manuales de Usuario, Mantenimiento y Operación. El cambio no impacta de manera grave a lo establecido en el modelo de referencia, ya que la entrega de estos documentos únicamente se retrasa una fase, pasa de la fase de Pruebas al Cierre.

Además de las unidades atómicas de *Manual de Usuario*, *Manual de Mantenimiento* y *Manual de Operación*, que es en donde se incluyen los manuales, está la unidad de Compendio de Seguimientos, en la cual se deben concentrar todas las mediciones realizadas durante el ciclo de desarrollo.

Resumen de unidades atómicas creadas para la plantilla *Integración* correspondiente a la fase homónima en COMPETISOFT:

- Introducción
- Descripción General
- Manual de Usuario
- Manual de Mantenimiento
- Manual de Operación

- Compendio de Seguimientos
- Justificación de las decisiones del Cierre
- Información de apoyo

En la Figura 5.9, se muestra un fragmento del Diagrama de Unidades Atómicas asociado a la plantilla de Cierre.

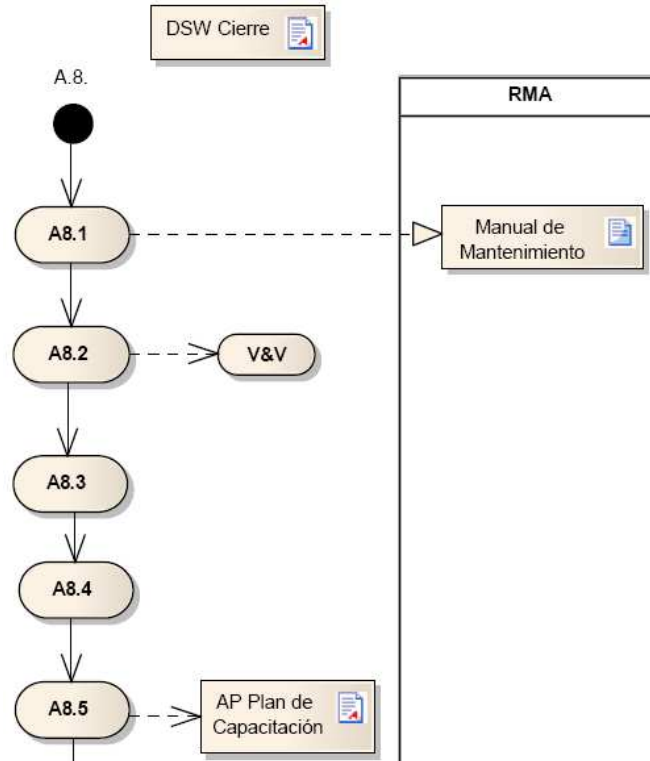


Figura 5.9 Fragmento del Diagrama de Unidades Atómicas del Cierre

CAPÍTULO 6 Plantillas de Mantenimientos de Software

El Mantenimiento de Software (MSW) propuesto en COMPETISOFT es un proceso ágil basado en Agil_MANTEMA [Pino 2008], ver Figura 6.1. Para este proceso se abstraeron dos elementos necesarios para llevarlo a cabo, la plantilla de *Petición de Modificación* en la que se solicita la participación del equipo de mantenimiento, y la plantilla de *Intervención*, en la cual se describe en detalle lo necesario para cumplir con la petición de modificación.

La estructura de estas plantillas, al igual que algunas de las plantillas de Administración del Proyecto, pretende que sean de fácil diligencia, debido a que, por ejemplo, la *Petición de Modificación* puede ser diligenciada por el usuario del producto de software y no por un miembro de la organización desarrolladora. Se observa que cada elemento de estas plantillas busca ser fácil de abstraer y que la información que refleje sea de utilidad para el equipo de mantenimiento. Las guías explícitas de cada unidad atómica son preguntas directas, al poseedor de la plantilla, cuya respuesta debe ser inmediata, clara y suficiente.

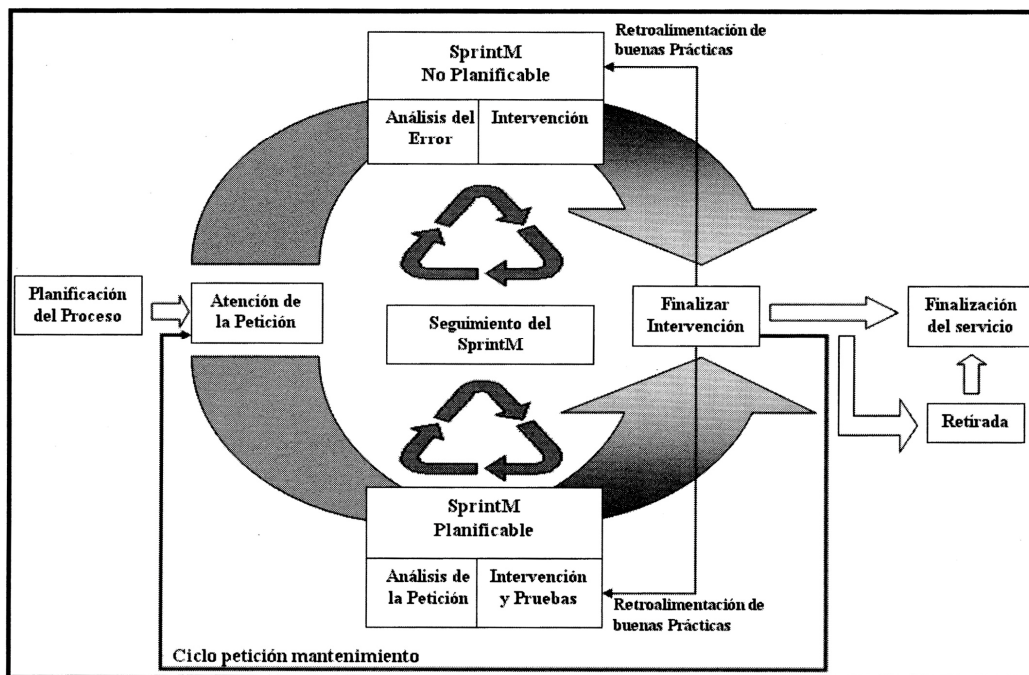


Figura 6.1 Proceso de Mantenimiento de Software propuesto en Agil_MANTEMA

6.1 Petición de Modificación

La plantilla de *Petición de Modificación* contiene diferentes rubros abstraídos de la metodología de Agil_MANTEMA. En esta metodología se establecen cinco tipos de mantenimientos:



Universidad Nacional
Autónoma de México



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 esta 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.

- Correctivo urgente
- Correctivo no urgente
- Perfectivo
- Adaptativo
- Preventivo

Después de analizar cada una de las características de los cinco tipos de mantenimientos se logró conjuntar en una sola plantilla lo necesario para que el equipo de mantenimiento realice una intervención adecuada con la información requerida. En el caso de los mantenimientos Perfectivo, Adaptativo y Preventivo la diferencia fundamental es que la petición no se origina a partir de un error, existe la posibilidad de responder a este tipo de peticiones de dos maneras, como un mantenimiento que rectifique alguna funcionalidad existente lo que nos lleva a poder tratarlo como un mantenimiento no urgente, o por otra parte, que la petición sea tratada como un incremento de funcionalidades en el sistema, por lo que cabe la posibilidad de considerarlo una nueva iteración del proceso de desarrollo.

En cuanto a las unidades atómicas de la plantilla tenemos a la *Prioridad* que determina si el servicio será urgente o no urgente, en la *Descripción del error* se establece una serie de preguntas con el objetivo de clarificar e identificar las acciones que dispararon el error. El *Entorno de ocurrencia* establece los parámetros existentes al momento de la ocurrencia, esta información es prioritaria para que el equipo de mantenimiento pueda recrear el error y estudiarlo de manera detenida.

Los *Aspectos afectados* son aquellas entidades que resultan perturbadas por el error o el mantenimiento. El *Diagnóstico y Posibles soluciones* contiene propuestas para solucionar el error, así como el diagnóstico de la ocurrencia describiendo la naturaleza del mismo. En *Información de Apoyo* sirve para complemento de la información no requerida en los rubros anteriores pero que puede resultar útil para ejecutar la intervención.

Por último se debe incluir al solicitante y a las personas que autorizan la petición. El listado de las unidades atómicas de la plantilla es:

- Prioridad
- Descripción del error
- Entorno de ocurrencia
- Aspectos afectados
- Solicitado por
- Diagnóstico y posibles soluciones
- Información de apoyo
- Autorización

6.2 Intervención

La *Intervención* comienza al recibir la *Petición de Modificación*, en ella se debe definir el procedimiento para darle solución. Es el equipo de mantenimiento el que realmente define el tipo de intervención que se efectuará, en la plantilla se tiene que definir el equipo de mantenimiento responsable de atender la petición, se establecen las características para recrear el entorno del sistema, los *Componentes involucrados* de los cuales se identificarán los *Componentes a corregir*. Además se deben fijar las *Restricciones*

y *Premisas* a tomar en cuenta para realizar la modificación y así definir claramente las *Acciones correctivas*.

Al concluir con esta parte se deben ejecutar diferentes tipos de prueba para mantener la consistencia del sistema, como lo son las *Pruebas Unitarias*, *Pruebas de Integración* y *Pruebas de Regresión*. Además se debe definir el *Plan de Retirada* de los componentes que ya no son útiles.

El listado completo de las unidades atómicas de la plantilla de *Intervención* es el siguiente:

- Tipo de intervención
- Equipo de mantenimiento
- Recreación del entorno
- Componentes involucrados
- Premisas y restricciones
- Componentes a corregir
- Acciones correctivas
- Pruebas unitarias
- Pruebas de integración
- Pruebas de regresión
- Plan de retirada
- Información de apoyo
- Autorización

En la Figura 6.2 se presenta un segmento de la plantilla de *Intervención*.

Intervención

Tipo de intervención	<i>[Planificable / No planificable]</i>
Equipo de mantenimiento	<i>[En este apartado se deberá incluir el listado de las personas y los roles encargados de dar atención a esta intervención]</i>
Recreación del entorno	<i>[En este apartado se debe incluir un listado de las características presentes en el entorno del sistema en el momento en que se presentó el error.]</i>
Componentes involucrados	<i>[En esta sección se deberán listar los aspectos que se ven afectados por el defecto descrito o la ocurrencia del error.]</i>
Premisas y restricciones	<i>[Listado de supuestos a tomar en cuenta para realizar la modificación, las restricciones delimitan las acciones.]</i>
Componentes a corregir	<i>[Listado de los componentes que serán modificados durante el mantenimiento.]</i>
Acciones correctivas	<i>[Listado de las medidas que se ejecutarán, elegidas a partir del diagnóstico, como soluciones a la petición de modificación.]</i>

Figura 6.2 Fragmento de una plantilla de Mantenimiento de Software

CAPÍTULO 7 Plantillas de PmCOMPETISOFT

La mejora de procesos de software (SPI por sus siglas en inglés), es un esfuerzo controlado y guiado que tiene como objetivo mejorar el conjunto de procesos de una empresa desarrolladora de software, en particular su capacidad y su madurez. Toda iniciativa para implantar una Mejora de Procesos de Software en una organización debe ser guiada por un marco metodológico que presente las actividades necesarias para obtener esa mejora dentro de la organización.

PmCOMPETISOFT es el modelo de mejora de COMPETISOFT. Tiene como propósito mejorar el conjunto de procesos de una organización en función de sus objetivos de negocio, a través de una serie de pasos bien establecidos. PmCOMPETISOFT es un proceso iterativo e incremental que consta de cinco fases por cada ciclo de mejora, ver Figura 7.1:

Instalación del ciclo: en esta fase se crea una propuesta de mejora basada en las necesidades del negocio, la cual guiará a la organización a través de cada una de las actividades, además de definir claramente los objetivos de la mejora, los cuales son establecidos a partir de las necesidades de la organización.

Diagnóstico de procesos: en esta fase se realizan valoraciones a los procesos de software existentes para poder identificar con mayor claridad las posibles *oportunidades de mejora*. Como resultado de la valoración de los procesos se definen los *casos de mejora* para posteriormente priorizarlos.

Formulación de mejoras: en esta fase se priorizan los *casos de mejora* y se planifican las iteraciones necesarias para realizarlos.

Ejecución de mejoras: en esta fase se ejecutan los casos de mejora, se genera un documento donde se registra la ejecución de los pilotos de prueba la evaluación de la mejora por la introducción de nuevos procesos y por el perfeccionamiento de los procesos ya existentes. En caso de éxito se desarrollan planes de aceptación e institucionalización de los nuevos procesos.

Revisión del ciclo: en esta fase se realiza un análisis de cierre recopilando las lecciones aprendidas y métricas recopiladas.

Existen cuatro productos de trabajo en PmCOMPETISOFT, la *Propuesta de Mejora*, el *Plan General de Mejora*, el *Plan de Implementación de Mejora* y el *Reporte de Mejora*, para cada uno de estos productos se creó una plantilla siguiendo lo definido en [Pino 2009].



Figura 7.1 Fases de PmCOMPETISOFT



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

7.1 Propuesta de Mejora

La plantilla de la *Propuesta de Mejora* está compuesta por las unidades atómicas de *Ciclo de Mejora* en donde se identifica a la mejora con un nombre. La de *Objetivos y alcance del ciclo de mejora* define las necesidades del negocio, los objetivos de mejora generales, el alcance del ciclo, el modelo de procesos a seguir y el método de evaluación elegido. En la unidad atómica de *Recursos del ciclo de mejora* se establece al asesor encargado de guiar la mejora, y se propone al posible evaluador, el responsable de la mejora en la organización, el listado de integrantes del Grupo de Mejora y del Grupo de Gestión de Mejora

Complementan a esta plantilla las unidades atómicas de:

- Ciclo de mejora
- Objetivos y alcance del ciclo de mejora
- Recursos del ciclo de mejora
- Otra información relevante
- Aprobación del Plan Preliminar de Mejora
- Sugerencias de mejora

7.2 Plan General de Mejora

El *Plan General de Mejora* está conformado por las unidades atómicas de *Ciclo de Mejora*, en donde se concretan y fijan los integrantes de los Grupos de Mejora de Procesos, Gestión de Mejora y al Responsable de Mejora de Procesos.

El *Nivel de capacidad actual y esperado de los procesos a mejorar en este Ciclo* es en donde se listan los Procesos a mejorar, además de registrar el Nivel de capacidad actual de cada uno de los procesos listados y el Nivel de capacidad esperado al finalizar el proceso de mejora. En la unidad atómica de *Número Iteración del Ciclo de Mejora* se propone una tabla en la que se priorizan los procesos a mejorar, la iteración en la que se va a mejorar, la duración de el esfuerzo de mejora y la fecha de inicio de la mejora, ver Figura 7.2.

La unidad atómica de *Planeación General* incluye un calendario de actividades en el que se indicará el tiempo estimado de la mejora y cada una de las actividades a realizar. Además se presentan unidades para el *Plan de manejo de Riesgos*, el *Plan de Capacitación* y el *Plan de Mediciones*.

Complementan a esta plantilla las unidades atómicas de:

- Ciclo de Mejora
- Nivel de capacidad actual y esperado de los procesos a mejorar en este Ciclo
- Número Iteración del Ciclo de Mejora
- Planeación General
- Plan de manejo de Riesgos
- Plan de Capacitación
- Plan de Mediciones

Nivel de capacidad actual y esperado de los procesos a mejorar en este Ciclo			
Proceso a mejorar	Nivel de Capacidad Actual	Nivel de Capacidad Esperado	
[Listado de los procesos priorizados a mejorar]	[Nivel de capacidad actual del proceso, de 0 a 4]	[Nivel de capacidad esperado del proceso, de 1 a 5]	
Número Iteración del Ciclo de Mejora			
Número de iteraciones del ciclo de mejora	[número de iteraciones de mejora]		
[Tabla en la que se muestra la planeación de las iteraciones]			
Iteración	Proceso	Duración	Desde
[número de iteración]	[Proceso a mejorar]	[Duración planeada de la mejora del proceso]	[Fecha de inicio de la mejora del proceso]

Figura 7.2 Fragmento de la plantilla del Plan General de Mejora

7.3 Plan de Implementación de Mejora

El *Plan de Implementación de Mejora* se realiza por cada iteración establecida en el *Plan General de Mejora*, en la plantilla tenemos a la unidad atómica de *Iteración* en donde se indica el caso de mejora y la iteración, la *Descripción del proceso* incluye un desglose de las actividades y productos del proceso a mejorar. La Estimación para el caso de mejora requiere la asignación de responsables así como los recursos asignados para cada actividad.

Los *Resultados destacables de la iteración* y las *Incidencias* son elementos requeridos, las unidades atómicas de esta plantilla son las siguientes:

- Ciclo de Mejora
- Iteración
- Descripción del Proceso
- Objetivos de la iteración
- Estimación para el caso de mejora
- Planeación de la iteración
- Resultados destacables de la iteración
- Incidencias
- Sugerencia para la planeación de la siguiente iteración
- Plan de Mediciones
- Otra información relevante
- Aprobación de la Implementación de Mejora
- Sugerencias de mejora a esta plantilla

7.4 Reporte de Mejora

En el *Reporte de Mejora* se registran las mediciones efectuadas por los evaluadores, debe existir una comparación entre las evaluaciones iniciales y finales, comparando los valores obtenidos y analizando qué procesos alcanzaron las expectativas de mejora, qué esfuerzo total se involucró y bajo qué modelo de evaluación se llevó a cabo la comparación, todos estos elementos son plasmados en la unidad atómica de *Evaluación de la mejora*, ver Figura 7.3.

Se deben de listar el *Registro de mejores prácticas* con los logros alcanzados, las lecciones aprendidas, las propuestas de mejora y las recomendaciones de ajuste. Completan la plantilla las siguientes unidades atómicas:

- Ciclo de Mejora
- Evaluación de la mejora
- Registro de mejores prácticas
- Otra información relevante
- Aprobación del Reporte de Mejora
- Sugerencias de mejora a esta plantilla

Ciclo de Mejora				
<i>Nombre de la empresa</i>		<i>[nombre de la empresa que realizará el proceso de mejora]</i>		
<i>Nombre del proyecto de mejora</i>		<i>[nombre que identifica al proceso de mejora]</i>		
Evaluación de la mejora				
<i>Evaluador</i>		<i>[Nombre del encargado de llevar a cabo la evaluación]</i>		
<i>Evaluación inicial</i>	<i>[Valores resultantes de la evaluación inicial de capacidades]</i>			
	Evaluación inicial			
	Atributos del proceso	A. 1.	...	A. n.
	Valoración			
Nivel de capacidad				
<i>Evaluación final</i>	<i>[Valores resultantes de la evaluación final de capacidades]</i>			
	Evaluación final			
	Atributos del proceso	A. 1.	...	A. n.
	Valoración			
Nivel de capacidad				

Figura 7.3 Fragmento de la plantilla Reporte de Mejora

CAPÍTULO 8 Resultados Obtenidos

Como resultado de este trabajo, se obtuvieron cuatro conjuntos de plantillas propuestas para los procesos de Administración de Proyectos, Desarrollo de Software, Mantenimiento de Software y Proceso de Mejora, todas ellas desarrolladas de acuerdo a lo establecido en COMPETISOFT. Estas plantillas están completas y listas para aplicarse.

8.1 Guía para el uso de las plantillas

Antes de comenzar a utilizar las plantillas es necesario dar un primer paso al definir el modelo de referencia que regirá al conjunto de procesos de la organización. El segundo paso será seleccionar la tecnología de trabajo encargada de ejecutar los procesos.

Si se ha elegido como marco de referencia de procesos a COMPETISOFT y se decidió seguir un desarrollo de software orientado a objetos, entonces las plantillas propuestas en este trabajo pueden ser utilizadas.

La organización que desee implementar o mejorar los procesos, debe seguir las plantillas de acuerdo a dos aspectos fundamentales, lo descrito en el modelo de referencia y el orden descrito en los *Diagramas de Unidades Atómicas* propuestos en este trabajo.

No se debe perder de vista que en primera instancia el objetivo es implementar el modelo en la organización, por lo que el regidor de los procesos debe ser sin discusión el mismo modelo, en este caso COMPETISOFT. Para lograr este primer objetivo bastará seguir el desarrollo iterativo e incremental definido por las actividades de cada proceso del modelo.

En COMPETISOFT el orden de las actividades está claramente descrito, por lo que el qué hacer ya está definido. Cuando la organización llega a preguntarse el cómo hacerlo es el momento en el que las plantillas toman una participación directa en el proceso de adopción.

Para cada plantilla existe un *Diagrama de Unidades Atómicas* que guía su uso. Cada actividad establecida en el modelo de referencia está representada dentro del diagrama correspondiente a la fase a la que pertenece la actividad. En caso de que la realización de la actividad genere algún producto, en el diagrama existirá una relación entre ésta y la unidad atómica que contiene la técnica que define el cómo realizar la actividad.

Además de que cada unidad atómica está contenida dentro de carriles que definen al rol responsable de llevarla a cabo.

Para proveer un ajuste mayor de las plantillas a la organización, el reacomodo, inclusión o exclusión de las unidades atómicas resulta primordial, el solo hecho de incluir o excluir a las unidades atómicas resulta sencillo debido a cómo fueron concebidas y generadas, sin embargo la importancia recae en el análisis que efectúa la organización para establecer con qué unidades atómicas trabajará. Esta decisión está fuera del alcance de este trabajo ya que recae totalmente en la organización. Para finalizar la recomendación es utilizar a las plantillas con todas sus unidades atómicas, si es el primer contacto con ellas, y posteriormente establecer mecanismos para su modificación y ajuste.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

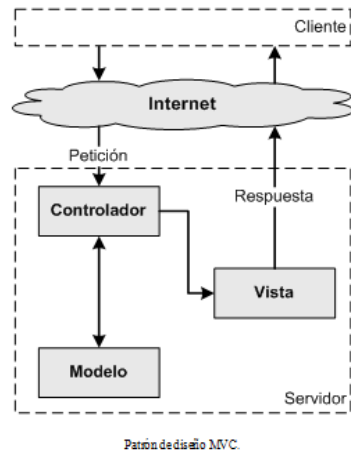
Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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 continuación a manera de ejemplo, en la Figura 8.1, se muestra el documento resultante de utilizar la plantilla de la fase de Análisis en el desarrollo de HEPALE!, se presenta el resultado de aplicar las técnicas propuestas para la unidad atómica de *Representación de la Arquitectura*, donde se presenta el diagrama, metas y restricciones impuestas a la arquitectura del sistema.



2. Representación de la Arquitectura



2.1 Metas de la arquitectura

La arquitectura deberá ser modular, para poder separar cada una de las características del modelo, en este sentido se debe de proveer una arquitectura que separe la lógica del sistema, de la capa de presentación y el acceso a datos.

Esto brindará al sistema una claridad para el flujo de los datos y al mismo tiempo para su mantenimiento y mejora, en este sentido se utilizará el patrón de diseño arquitectónico Modelo Vista Controlador (MVC), este patrón separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC es ampliamente utilizado en aplicaciones para la web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador

Figura 8.1 Uso de la Plantilla Análisis en del Proyecto HEPALE!

Finalmente, ha surgido el interés de aplicar las plantillas por parte de algunas instituciones, por ejemplo:

- En el desarrollo del sistema *HemoSist*, realizado por alumnos del curso de *Ingeniería de Software Orientada a Objetos* del Posgrado en Ciencia e Ingeniería de la Computación de la UNAM, para el Instituto Nacional de Cardiología.
- En el desarrollo de un sistema administrador del aprendizaje y un repositorio desarrollado también por alumnos del Posgrado.
- En el de desarrollo de software para el control de equipo de medición a distancia realizado por alumnos de la Universidad Autónoma del Estado de México.
- En el Laboratorio Nacional de Informática Avanzada (LANIA), con el fin de generar aplicaciones en las líneas de investigación en las que trabaja.

8.2 Difusión de las plantillas a través de E-learning

La propuesta presentada en este trabajo de tesis forma parte de un proyecto de mayor tamaño llamado HEPALE! (Herramienta Educativa Para la Adopción de Los Estándares) [Cruz 2009a]. HEPALE! es un proyecto compuesto por tres esfuerzos con el objetivo de atacar la falta de conocimiento y diseminación de los modelos de referencia y de las herramientas disponibles para ponerlos en práctica.

El primer paso para la consecución de HEPALE! fue la creación de un conjunto de paquetes de despliegue [Laporte 2008b] para ISO/IEC 29110 y COMPETISOFT. Dentro de estos paquetes de despliegue se incluyen las plantillas, desarrolladas a partir de este trabajo de tesis, para la implementación de estos dos modelos de referencia.

El segundo paso fue el desarrollo de un repositorio hecho a la medida [Cruz 2009b] que cuenta con los cambios necesarios en su estructura para la manipulación de Objetos de Aprendizaje e E-learning. El repositorio fue construido bajo una arquitectura orientada a servicios implementando servicios web para la manipulación, búsqueda y almacenamiento de paquetes de despliegue. El repositorio facilita el intercambio de contenidos entre diferentes sistemas administradores del aprendizaje, lo que facilita su publicación y difusión.

Finalmente se creó un sistema administrador del aprendizaje a la medida [Morgado] que permitiera una adecuada conjunción del repositorio, los paquetes de despliegue y las plantillas, ver Figura 8.2.

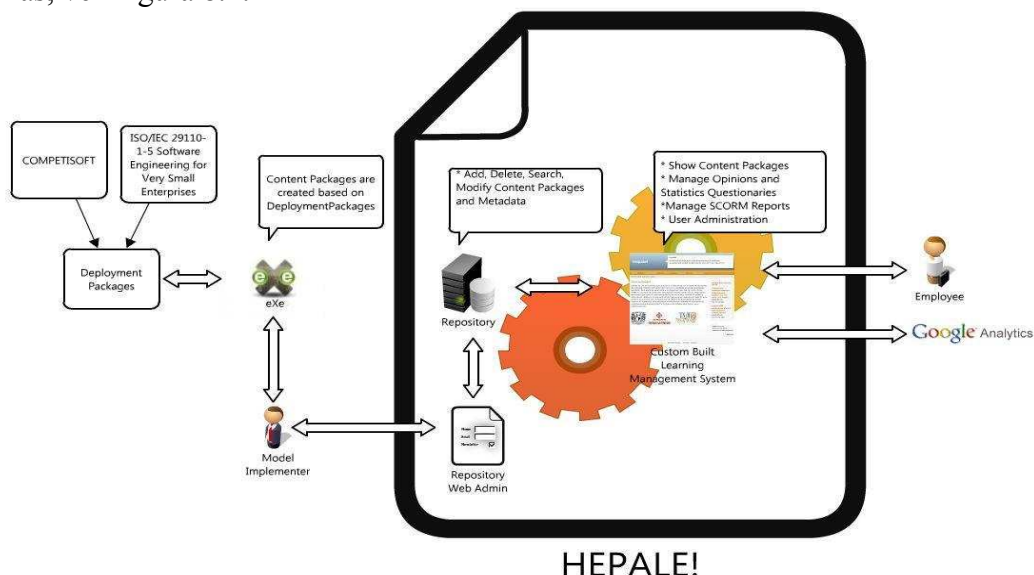


Figura 8.2 Diagrama general del Proyecto HEPALE!

En su totalidad, HEPALE! es una herramienta que apoya la adopción y mejora de procesos, utilizando al e-learning como medio de difusión del conocimiento de los modelos de referencia de procesos. Al mismo tiempo HEPALE! recaba información estadística útil que puede ser analizada con el fin de mejorar la calidad de la oferta educativa y contribuir con la difusión de los modelos de referencia a nivel internacional.

Actualmente HEPALE! se encuentra a punto de comenzar con los casos de estudio, el primero de ellos tendrá lugar en el Cluster de PyMEs de la Industria del Software del estado de Guerrero.

8.3 Adaptación de las Unidades Atómicas a otros modelos

Las unidades atómicas fueron diseñadas con el objetivo de cubrir una entidad descrita en algún proceso y el contenido de las mismas está basado en cuerpos de conocimiento los suficientemente sólidos y aceptados en la Ingeniería de Software, por lo que al contar con un número suficiente de unidades atómicas, es posible elegir aquéllas que cubran los procesos de algún otro modelo de referencia.

A modo de ejemplo se presenta el caso de la adaptación de las unidades atómicas al paquete de despliegue *Software Requirements Specification* (Especificación de Requerimientos del Software) de la futura norma ISO/IEC 29110, en su perfil básico. Y el caso de la adaptación de una unidad atómica para incluir desarrollos que requieren una alta especialización en el manejo de datos.

8.3.1 Adaptación a ISO/IEC 29110-5-1

La futura norma ISO/IEC 29110 tiene como principal objetivo proveer de un modelo de referencia a las PyMEs especialmente hecho para ellas, buscando hacer el contenido de la norma más asequible para las organizaciones, adaptándolo a su tamaño y sus necesidades de negocio.

La descripción de la ISO/IEC 29110, en su perfil básico 5-1 incluye paquetes de despliegue con el objetivo de hacerla accesible y reducir el esfuerzo de implementación por parte de la organización. Un paquete de despliegue se define como *un conjunto de productos desarrollados con el objetivo de facilitar y acelerar la implementación de un conjunto de prácticas de un modelo de referencia en una PyME* [Laporte 2008a].

Mediante la implementación de un paquete de despliegue, una PyME puede ver un paso concreto para lograr la cobertura del Proceso, descrito en las partes 4 y 5. Los paquetes de despliegue están diseñados de tal manera que un PyME puede aplicar su contenido, sin tener que aplicar el marco completo al mismo tiempo, ver Tabla 8.1.

Contenido Típico de un Paquete de Despliegue
1. Descripción Técnica
Propósito del documento
Por qué este tema es importante?
2. Definiciones
3. Relaciones con la ISO/IEC 29110
4. Visión General de los Procesos, Actividades, Tareas, Roles y Productos
5. Descripción de los Procesos, Actividades, Tareas, Roles y Productos
Descripción de los Roles
Descripción de los Productos
Descripción de los Artefactos
6. Plantillas
7. Ejemplos
8. Listas de Verificación
9. Herramientas
10. Referencias a otros Estándares o Modelos
11. Referencias
12. Forma de Evaluación

Tabla 8.1 Contenido típico de un paquete de despliegue

Cabe resaltar que en la estructura de un paquete de despliegue, de acuerdo a su objetivo, es fundamental la presencia de *Plantillas* y *Ejemplos*, puntos 6 y 7 respectivamente. Aprovechando la modularidad de las plantillas, se eligió un conjunto de unidades atómicas para adaptarlas a la ISO/IEC 29110-5-1, con el fin de construir plantillas que formen parte de los paquetes de despliegue que propone esta futura norma.

En la Figura 8.3 se muestra el contenido resultante de conjuntar las unidades atómicas adecuadas para completar el paquete de despliegue de *Software Requirements Specification* de ISO/IEC 29110-5-1. La facilidad con la que se ha podido adaptar a este paquete de despliegue es un ejemplo de las virtudes de la definición de las plantillas a través de unidades atómicas. Esta modularidad es un aspecto que busca hacer flexible la implantación de los procesos en la organización.

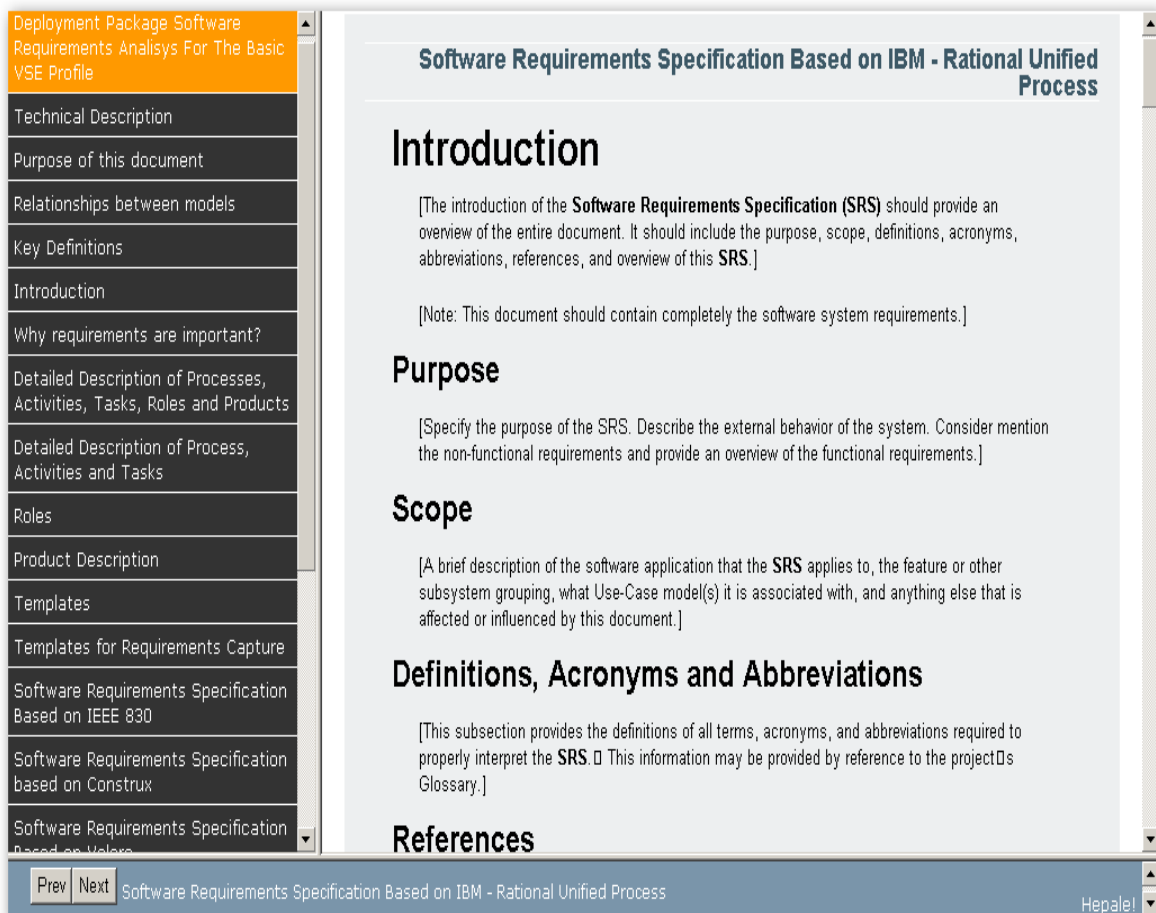


Figura 8.3 Paquete de despliegue para la ISO/IEC 29110-5-1

La construcción del paquete de despliegue *Software Requirements Specification* fue resultado de un trabajo conjunto con la Universidad de Quebec (Canadá), la Universidad de Castilla-La Mancha (España) y la Universidad Nacional Autónoma de México, sentando las bases para la construcción de los restantes paquetes de despliegue de la ISO/IEC 29110-5-1.

8.3.2 Componentes de Base de Datos para MoProSoft

Para desarrollos que requieren de una alta especialización en el manejo de datos, en [García-García] se definen prácticas y guías de bases de datos con el objetivo de integrarlas a la estructura de MoProSoft. En estas guías se define una propuesta para que, durante el Desarrollo y Mantenimiento de Software, la base de datos cumpla con los estándares de calidad establecidos.

Los elementos definidos en estas prácticas y guías tienen que ver con la inclusión de roles, actividades y productos específicos para bases de datos. Esta inclusión no contradice o excluye aspectos considerados previamente en el modelo de referencia. Sin embargo es necesario prestar atención a la trazabilidad entre los nuevos elementos.

Aprovechando la afinidad entre los modelos de referencia, COMPETISOFT y MoProSoft, a manera de ejemplo se construyó la unidad atómica necesaria para cubrir los requerimientos de bases de datos de la fase de Análisis y Diseño *extendida* planteada en [García-García].

Reutilizando la unidad atómica de *Vista de Datos*, perteneciente a la plantilla de Diseño, ésta fue enriquecida con las guías y prácticas descritas para bases de datos. Resultando en una unidad atómica robusta que satisface lo requerido por la *fase extendida* y por consecuencia en las fases análogas de COMPETISOFT y MoProSoft.

El elemento existente de *Descripción del Contenido* fue modificado internamente, por otro lado el elemento de *Diagrama de la Base de Datos*, hipotéticamente, será reacomodado y colocado en la fase de *Especificación de Requerimientos extendida*. Posteriormente se incluyeron los demás elementos solicitados en la *fase extendida*. A continuación se presentan brevemente los elementos de la unidad atómica *Vista de Datos* modificada:

- **Descripción del Contenido:** en este elemento se describe brevemente el contenido de la unidad atómica.
- **Objetivos del Modelo de la Base de Datos:** en este elemento se definen los puntos que se generarán y serán trazables a la fase de Construcción.
- **Desarrollo del Modelo de la Base de Datos:** aquí se deben diseñar los esquemas del Modelo de la Base de Datos que servirán como base para la fase de Construcción.
 - **Esquema Lógico Estándar**
 - **Esquema Lógico Específico**
 - **Esquema Físico**
 - **Diccionario de Datos**
- **Glosario de Términos:** en este elemento se relacionan los términos propios del contexto y su significado. Este glosario sirve como vocabulario de referencia.

Cabe resaltar que para esta adaptación no se incluyeron a dos elementos descritos en la *fase extendida* debido a que los mismos ya se encuentran definidos con anterioridad o cuentan con un equivalente en esta propuesta.

En la Figura 8.4 se muestra un fragmento de la unidad atómica resultante.

7.3.3 Esquema Físico

[En esta sección se debe diseñar el Esquema Físico de la base de datos. Para llevar a cabo este proceso se debe utilizar el ELEsp como base, considerando los siguientes elementos:

Diseño de Bloques: este diseño sólo se puede realizar si el SMBD lo permite, definiendo el espacio libre de cada bloque, el porcentaje de utilización y el número de bloques asignados a cada tabla.

Diseño de la Organización: este diseño sirve para definir como realizar la organización de los archivos para cada tabla e indica la forma de acceso a los datos almacenados.

Utilización de Índices: con el objetivo de optimizar los tiempos de respuesta a consultas.

Redundancia de Datos: se debe minimizar la redundancia para evitar inconsistencias de los datos.]

7.3.4 Diccionario de Datos

[En esta sección se deben establecer para cada dato los siguientes metadatos]

Nombre	Tipo de Dato	Tamaño	Tipo de Llave	Descripción
[Nombre del campo]	[Especificar el tipo de dato que almacenará el campo]	[Especificar el número máximo de caracteres que almacenará el campo]	[Primaria / Foránea / No es llave]	[Breve descripción de la función del campo y lo que almacena]

Figura 8.4 Fragmento de la unidad atómica modificada

El Diagrama de Unidades Atómicas asociado a esta extensión de la fase sufre una ligera modificación. Debido a que en [García-García] se propone la inclusión de dos nuevos roles, Diseñador y Administrador de la Base de Datos, que no están presentes en el modelo de COMPETISOFT ni en MoProSoft, el diagrama deberá considerar que la unidad atómica de *Vista de Datos* es utilizada, en este caso, por el Diseñador de la Base de Datos (DBD). Ver Figura 8.5.

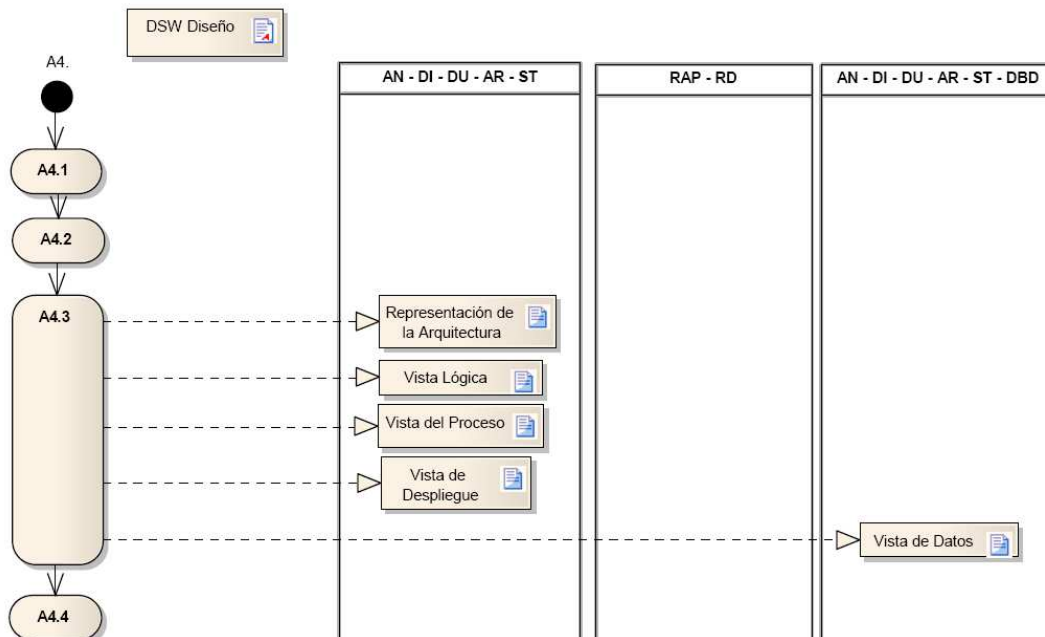


Figura 8.5 Fragmento del Diagrama de Unidades Atómicas modificado

8.4 Artículos publicados

Como resultado del trabajo desarrollado en colaboración con la Universidad de Castilla-La Mancha y la Universidad Nacional Autónoma de México, se logró la publicación de dos artículos:

- *Supporting the Software Process Improvement in Very Small Entities through E-learning: the HEPALE! Project* [Cruz 2009a], presentado en el Congreso Internacional Mexicano de Ciencias de la Computación (ENC'09).
- *Apoyando la Mejora de Procesos de Software a través de Plantillas* [Morales], presentado en el Simposio de Ingeniería de Software (SIS'09).

El primero de ellos se enfoca en el desarrollo del Proyecto HEPALE! cuyo objetivo es disminuir la falta de conocimiento y disseminación de los modelos de referencia y de las herramientas disponibles para ponerlos en práctica.

El segundo es resultado directo de la propuesta para la adopción de procesos de software en PyMEs desarrollada en este trabajo de tesis.

Conclusiones y Trabajo Futuro

Al menos tres cuartas partes de las empresas de desarrollo de software a nivel mundial son PyMEs, esto es, cuentan con 25 o menos empleados, por lo que les es difícil adoptar los grandes modelos de referencia para mejorar la calidad de sus productos y volverse competitivas en un mundo globalizado.

Con esta motivación, el trabajo se inició estableciendo como objetivo el presentar una alternativa que facilitara la adopción de los procesos establecidos por modelos de referencia para PyMEs, en particular COMPETISOFT y la futura norma ISO/IEC 29110.

Se construyó una guía centrada en un conjunto de plantillas de productos, basadas en técnicas descritas en el Proceso Unificado, RUP y PMBoK, para apoyar la adopción de los modelos con el fin de proveer una mejora tomando en cuenta la realidad y las necesidades de este tipo de organizaciones.

A lo largo de este escrito se presentaron las plantillas que componen la guía para implementar el modelo de referencia de COMPETISOFT. Se construyeron cuatro conjuntos de plantillas que podemos diferenciar en dos estilos bien definidos. Por una parte las plantillas utilizadas para la fase de *Control* de la Administración de Proyectos, para el Mantenimiento de Software y para el Proceso de Mejora de COMPETISOFT, son plantillas de fácil diligencia y autocontenidas, buscando proveer de agilidad y sencillez a los distintos procesos sin perder de vista la formalidad de los mismos.

Por otro lado las plantillas de Desarrollo de Software y la correspondiente a la fase de *Planeación* de la Administración de Proyectos, tienen una estructura enfocada en cubrir todos los requerimientos establecidos en el modelo de referencia, por lo que el objetivo principal fue que mantuvieran un grado considerable de consistencia y formalidad para su uso y ejecución.

Dicha guía tiene la característica de ser flexible debido a la estructura modular de las plantillas, por lo que resulta fácilmente adaptable a las necesidades de la organización. La flexibilidad con la que cuentan las plantillas se logró mediante la introducción del concepto de *unidades atómicas* como base para construirlas, ya que permiten adecuarlas a varios modelos de referencia.

La modularidad tiene como consecuencia una alta adaptabilidad de las plantillas a cualquier desarrollo orientado a objetos, además, al proveer técnicas de estándares de la industria altamente aceptados y reconocidos, ocasiona que el efecto de adaptabilidad se incremente.

Asimismo, esta propuesta se está aplicando en distintos entornos, por lo que las plantillas han sido distribuidas a varios grupos de trabajo para su utilización:

- En el desarrollo del sistema administrador del aprendizaje y del repositorio *a la medida*, componentes del Proyecto HEPALE!
- En el desarrollo del sistema *HemoSist*, mismo que se encuentra en proceso de realización por alumnos del curso de *Ingeniería de Software Orientada a Objetos* del Posgrado en Ciencia e Ingeniería de la Computación de la UNAM, para el Instituto Nacional de Cardiología.
- En el proyecto para la creación del *Laboratorio Remoto de Mediciones en Plataforma Web para Intranet*, de la Universidad Autónoma del Estado de México, Centro Universitario Ecatepec.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

- En el Laboratorio Nacional de Informática Avanzada (LANIA), con el fin de generar aplicaciones en las líneas de investigación en las que trabaja.

Al momento de finalizar este trabajo se han recopilado una serie de comentarios iniciales que nos permiten identificar algunos beneficios consecuencia del uso de estas plantillas:

- Las plantillas facilitan y muestran el camino a seguir al realizar las actividades de un proceso según un modelo de referencia.
- Las plantillas integran los aspectos necesarios y solicitados por el modelo de referencia, por lo que no es posible pasar por alto ninguno.
- La aplicación de las plantillas reduce el tiempo de implantación de cada fase del proceso.
- En las plantillas se puede concentrar todo lo que resulta de realizar las actividades del modelo, por lo que al final se obtiene la documentación integral, completa y estandarizada sin incongruencias ni contradicciones.
- El uso de *unidades atómicas* facilita la trazabilidad de los elementos, además provee de una flexibilidad notable a la plantilla, ya que la decisión de usar o no cada *unidad atómica* incrementa su utilidad para el desarrollador.
- La manera en que están formuladas las *unidades atómicas*, en especial las de Administración de Proyectos y de Mantenimiento de Software, acorta el tiempo y simplifica su diligencia, además facilita su aplicación a varios modelos de referencia.

Por cuestiones de tiempo en este trabajo no se recopilaron la totalidad de los comentarios de los casos de estudio y sus resultados.

Contribuciones

La principal contribución de esta tesis es presentar y promover una alternativa flexible y estandarizada para la implementación de buenas prácticas de Ingeniería de Software para las PyMEs. La introducción de las *unidades atómicas* como base para la adaptación de cualquier modelo de referencia a una PyME, resultan ser una herramienta más para reducir la brecha entre las PyMEs y los modelos.

Por otra parte, la aportación de decidir encapsular entidades del proceso en *unidades atómicas* hace posible su utilización y adaptación a otros modelos de referencia, como lo fue la conformación del primer paquete de despliegue *Software Requirements Specification* de la futura norma ISO/IEC 29110.

Otra contribución, es la consecución del Proyecto HEPALE! como el comienzo de una herramienta educativa que apoya la adopción de modelos de referencia en PyMEs.

Además, se ha iniciado una colaboración con el Laboratorio Nacional de Informática Avanzada para continuar con el estudio de esta propuesta de adopción basada en plantillas.

Finalmente, se logró la publicación de dos artículos íntimamente relacionados con este trabajo:

- *Supporting the Software Process Improvement in Very Small Entities through E-learning: the HEPALE! Project*, presentado en el Congreso Internacional Mexicano de Ciencias de la Computación (ENC'09).
- *Apoyando la Mejora de Procesos de Software a través de Plantillas*, presentado en el Simposio de Ingeniería de Software (SIS'09).

Trabajo Futuro

Como trabajo futuro se identificaron los siguientes puntos a desarrollar en base a este trabajo:

- Integrar las plantillas a todo el trabajo previo que existe sobre COMPETISOFT, para hacerlas accesibles y disponibles para usarse por las organizaciones que lo deseen.
- Obtener un mayor número de aplicaciones de las plantillas para evaluar, refinar y mejorar su contenido.
- Iniciar una aplicación industrial de las plantillas.

Apéndice A Plantillas de Administración de Proyectos



AP Plan del Proyecto

Índice

1.	Introducción	2
1.1	Definiciones, Acrónimos y Abreviaturas	2
1.2	Referencias	2
2.	Descripción del producto	2
2.1	Perspectiva del producto	2
2.2	Funciones del producto	2
2.3	Supuestos del producto	2
2.4	Licenciamiento	2
2.5	Instalación	2
2.6	Criterios de calidad	2
3.	Alcance	3
4.	Objetivos	3
4.1	Objetivos del proyecto	3
4.2	Objetivos del producto	3
5.	Entregables	3
6.	Restricciones	3
7.	Proceso específico	4
8.	Protocolo de entrega	4
9.	Ciclos y actividades	4
10.	Tiempo estimado	4
11.	Plan de adquisiciones y capacitación	5
12.	Equipo de trabajo	6
13.	Calendario	6
14.	Costo	7
15.	Riesgos	7
16.	Estrategia de control de versiones	8
17.	Plan de comunicación	9
18.	Justificación del plan del proyecto	10
19.	Información de apoyo	10

Fase:

Responsable:

A.1.1 Plantilla de Plan del Proyecto



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.



Plan del Proyecto

1. Introducción

[La introducción del Plan del Proyecto ofrece una visión general de todo el documento.]

1.1 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.2 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del Plan del Proyecto. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción del producto

2.1 Perspectiva del producto

[cuál es el propósito del proyecto.]

2.2 Funciones del producto

[qué funciones cumplirá el producto]

2.3 Supuestos del producto

[el producto dependerá de ...]

[el producto contará con ...]

2.4 Licenciamiento

[propiedad intelectual del sistema quedará a favor de ...]

2.5 Instalación

[la instalación del sistema correrá a cargo de ...]

2.6 Criterios de calidad

[presentación del código]

[entregas de documentación]

[pruebas]

[desempeño]

[lenguajes de programación.]

Fase:

Responsable:

A.1.2 Plantilla de Plan del Proyecto



3. Alcance

[Descripción del alcance del sistema, descripción de la aplicación, características, asociaciones existentes y cualquier aspecto que se vea afectado por el desarrollo de éste.]

4. Objetivos

4.1 Objetivos del proyecto

[Los objetivos del proyecto son aquéllos que se desean alcanzar en conjunto, aquellas metas que de antemano se sabe que de no trabajar en equipo difícilmente lograrían. Estos objetivos proporcionan una base de lanzamiento para el proyecto, que de ser cumplidos satisfactoriamente darán como resultado un equipo de trabajo y un cliente satisfechos con lo realizado a lo largo del proyecto.]

4.2 Objetivos del producto

[Los objetivos del producto, se consideran todas aquellas metas que debe cumplir el producto terminado. Van más encaminados a cuestiones de funcionalidad del sistema desde el punto de vista del usuario final, cliente y por supuesto de los realizadores.]

5. Entregables

[Listado de los productos a ser entregados a lo largo del proyecto.]

Entregables	
Producto	Tipo
<i>[Nombre de la fase]</i>	
<i>[nombre del producto]</i>	<i>[tipo del producto]</i>
<i>[Nombre de la fase]</i>	

6. Restricciones

[Listado de las restricciones presentes para la realización del proyecto.]

Restricciones	
Nombre	<i>[nombre de la restricción]</i>
Descripción	<i>[breve descripción de la restricción]</i>
Origen	<i>[interna / externa]</i>
Tipo	<i>[monetaria / temporal / personal / institucional]</i>
Limitan a	<i>[acciones a seguir si el riesgo ocurre]</i>

Fase:

Responsable:

A.1.3 Plantilla de Plan del Proyecto



7. Proceso específico

[Descripción del proceso, en este caso debe incluirse un vínculo a COMPETISOFT.]

8. Protocolo de entrega

[Mecanismo aceptado para la entrega de productos descritos en la sección de Entregables.]

Protocolo de entrega	
Responsable	<i>[miembro del equipo encargado de realizar la entrega]</i>
Destinatario	<i>[nombre de la persona a la que se debe entregar]</i>
Fecha	<i>dd/mm/aaaa</i>
Duración máxima	<i>[tiempo máximo estimado para la presentación]</i>
Lugar	<i>[lugar donde se realizará la entrega]</i>
Medio de comunicación	<i>[presencial / vía electrónica]</i>
Elementos entregados	<i>[listado de elementos a entregar en la sesión]</i>

9. Ciclos y actividades

[De acuerdo al proceso específico para el desarrollo del proyecto y a las actividades identificadas para la consecución del mismo, será necesario establecer el número de ciclos necesarios para llevar a cabo todas y cada una de las actividades. Esto estará fuertemente relacionado con la capacidad de la empresa y el proceso específico elegido.]

10. Tiempo estimado

[El primer paso será identificar las actividades que deben ser realizadas para producir un entregable, posteriormente se deberán establecer las dependencias entre las actividades, por último se deberá estimar la cantidad de periodos laborables, horas hombre o días necesarios para completar la actividad. RUP cuenta con una lista de actividades tipo.]

Fase:

Responsable:

A.1.4 Plantilla de Plan del Proyecto



11. Plan de adquisiciones y capacitación

[Listado de las adquisiciones, contrataciones y capacitaciones necesarias a lo largo del proyecto.]

Adquisiciones	
Nombre	<i>[nombre del elemento]</i>
Descripción	<i>[breve descripción del elemento]</i>
Fecha de solicitud	<i>dd/mm/aaaa</i>
Fecha de adquisición	<i>dd/mm/aaaa</i>
Tipo de adquisición	<i>[compra definitiva o renta]</i>
Solicitante	<i>[persona física que lo solicita]</i>
Resultado esperado	<i>[breve descripción de lo que se espera al adquirir este elemento]</i>

Contrataciones	
Nombre	<i>[nombre del contrato]</i>
Descripción	<i>[breve descripción del contrato]</i>
Tipo de contrato	<i>[De precio fijo o Por tiempo y materiales]</i>
Inicio del contrato	<i>dd/mm/aaaa</i>
Término del contrato	<i>dd/mm/aaaa</i>
Autorizado a cerrar el contrato	<i>[persona física autorizada a dar por cerrado el contrato]</i>
Resultado esperado	<i>[breve descripción de lo que se espera obtener mediante esta contratación]</i>

Fase:

Responsable:

A.1.5 Plantilla de Plan del Proyecto



Capacitaciones	
Nombre del curso	<i>[nombre del curso]</i>
Descripción	<i>[breve descripción del curso]</i>
Tipo	<i>[presencial / en línea]</i>
Horas totales	<i>[número de horas de duración del curso]</i>
Responsable	<i>[persona física autorizada a dar por cerrado el contrato]</i>
Número de participantes	<i>[cupo de participantes en este curso]</i>
Fecha de inicio	<i>dd/mm/aaaa</i>
Fecha de término	<i>dd/mm/aaaa</i>
Resultado esperado	<i>[breve descripción de lo que se espera obtener mediante este curso]</i>

12. Equipo de trabajo

[Listado de los miembros del equipo de trabajo, incluyendo información de contacto.]

Equipo de Trabajo					
Rol	Nombre	Apellidos	Teléfono	Dirección	Correo electrónico

13. Calendario

[Calendario de la duración del proyecto, puede establecerse mediante un diagrama de Gantt.]

Calendario						
Identificador	Descripción de la actividad	Unidades de calendario	Tiempo estimado del proyecto			
			Periodo 1			Periodo n

Fase:

Responsable:

A.1.6 Plantilla de Plan del Proyecto



14. Costo

[Valor monetario resultante de la realización del proyecto.]

Costo													
Categoría	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12	Total
Sueldos													
Hardware													
Subcontratos													
Mantenimiento													
Capacitación													
Comunicaciones													
Administración													
...													
Total													

15. Riesgos

[Listado de los riesgos identificados que podrían afectar el desarrollo del proyecto en caso de ocurrencia.]

Riesgos	
Nombre	<i>[nombre del riesgo]</i>
Descripción	<i>[breve descripción del riesgo]</i>
Probabilidad de ocurrencia	<i>[alta / media / baja]</i>
Impacto del riesgo	<i>[alto / medio / bajo]</i>
Plan de contingencia	<i>[acciones a seguir si el riesgo ocurre]</i>
Estado del riesgo	<i>[identificado / controlado / mitigado / ignorado]</i>

Fase:

A.1.7 Plantilla de Plan del Proyecto

Responsable:



16. Estrategia de control de versiones

[Método para administrar la configuración del sistema, se debe incluir un mecanismo para solicitar cambios así como la estructura y reglas de uso del repositorio.]

Estrategia del control de versiones											
Identificación de la configuración	<i>[establecer lo que será almacenado en el repositorio, suministrar la línea base para clasificar productos y documentos]</i>										
Estado de la configuración	<i>[información de los productos que están en el repositorio, listando los productos almacenados, su ubicación y el estado en que se encuentran]</i>										
Auditor	<i>[individuo encargado de verificar la existencia y estado de los productos que se estableció generar. El auditor deberá corroborar en cada auditoría que se ha cumplido con la ejecución de los productos]</i>										
Frecuencia de auditorías	<i>[establecer el plazo en que se efectuarán las auditorías]</i>										
Cambios solicitados	<table border="1"> <thead> <tr> <th>Nombre del producto</th> <th>Descripción</th> <th>Impacto</th> <th>Solicitante</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p><i>[listado de los cambios solicitados a lo largo del proceso]</i></p>	Nombre del producto	Descripción	Impacto	Solicitante	Estado					
Nombre del producto	Descripción	Impacto	Solicitante	Estado							
Políticas	<i>[acuerdos para el uso del repositorio, establecer quién puede almacenar, modificar y acceder al repositorio]</i>										

Forma de Solicitud de Cambio

Proyecto:

Fecha:

Información del elemento

Nombre:

Dirección de respaldo:

Información del cambio

Descripción del cambio

Beneficio del cambio

Impacto del cambio

Estado del cambio

Fecha de dictamen:

APROBADO

RECHAZADO

Fase:

Responsable:

A.1.8 Plantilla de Plan del Proyecto



Nombre y firma del dictaminador:

Repositorio	
Clasificación	<i>[distribuida / centralizada]</i>
Herramienta a utilizar	<i>[CVS / Subversion]</i>
Ubicación	<i>[dirección de donde se encuentra el repositorio]</i>
Estructura	<i>[descripción de la estructura elegida para la organización del repositorio, árbol de directorios]</i>

17. Plan de comunicación

[En esta sección se deben mencionar y describir los mecanismos de comunicación válidos entre los involucrados a lo largo del proyecto.]

Plan de Comunicación	
Medios de distribución aceptados	<i>[listado de medios de comunicación aceptados]</i> <i>Juntas</i> <i>E-mail</i> <i>V-mail</i> <i>Uno a uno</i> <i>Teléfono</i> <i>Fax</i> <i>Cartas y memorandums</i> <i>Video conferencias</i> <i>Juntas virtuales</i> <i>Chat</i>
Elementos involucrados	<i>[listado de disciplinas, departamentos y especialidades involucradas]</i>
Necesidades de información interna	<i>[acuerdo para la comunicación entre los miembros del proyecto]</i>
Necesidades de información externa	<i>[acuerdo para la comunicación entre el equipo de trabajo y medios externos]</i>

Fase:

Responsable:

A.1.9 Plantilla de Plan del Proyecto



Comunicado	
Elemento de comunicación	<i>[información que será distribuida]</i>
Destinatarios	<i>[persona o grupos que recibirán la información]</i>
Finalidad	<i>[motivo por el que es distribuida la información]</i>
Frecuencia	<i>[intervalo de tiempo en el que se redistribuirá la información]</i>
Plazo	<i>dd/mm/aaaa al dd/mm/aaaa</i>
Medio	<i>[medio de distribución de la información]</i>
Responsable de la comunicación	<i>[individuo encargado de asegurarse que la información ha sido transmitida]</i>
Cadena de mando	<i>[persona o grupo inmediato superior encargado de resolver polémicas respecto a esta información]</i>

18. Justificación del plan del proyecto

[Esta sección debe contener las justificaciones principales del por qué se incluyeron los aspectos definidos anteriormente.]

19. Información de apoyo

[La información de apoyo para el Plan del Proyecto. Incluye:

Tabla de contenidos

Índice

Apéndices.]



Documento de Aceptación

Nombre del documento/proyecto	<i>[nombre del documento/proyecto]</i>		
Descripción	<i>[Nombre corto del documento, incluido el nombre corto del proyecto y una breve descripción del contenido del documento]</i>		
Elaborado por	<i>[nombre de la persona que elaboró el documento]</i>		
Última revisión	<i>[dd/mm/aaaa]</i>		
Versión	<i>[0.0]</i>		
Válido	Desde: <i>[dd/mm/aaaa]</i>	Hasta: <i>[dd/mm/aaaa]</i>	
Cambios sobre el documento	<i>[ninguno]</i>	Estado	<input type="checkbox"/> Borrador <input type="checkbox"/> Definitivo
Autorización	Revisado por Responsable de Administración del Proyecto <i>[nombre]</i>	Aceptado por Cliente <i>[nombre]</i>	

Fase:

Responsable:

A.2 Plantilla de Documento de Aceptación



Matriz de Trazabilidad

1. Descripción general

[La matriz de trazabilidad resulta de establecer asociaciones entre los requisitos establecidos en la ERS con los productos que se generan a través del proceso para satisfacerlos.]

2. Matriz de trazabilidad

[Matriz propuesta para establecer las relaciones entre un requerimiento establecido por el cliente y un caso de uso abstraído por el equipo de trabajo.]

Identificador	Elemento	Trazabilidad hacia adelante
<i>[Identificador del elemento.]</i>	<i>[Descripción del requerimiento establecido por el cliente.]</i>	<i>[Identificador del elemento que satisface el requerimiento.]</i>

[Matriz propuesta para establecer las relaciones entre los casos de uso y el plan de pruebas del caso de uso]

[Matriz propuesta para establecer las relaciones entre los casos de uso y los diagramas de realización]

[Matriz propuesta para establecer las relaciones entre los diagramas de realización y los métodos que requieren]

[Matriz propuesta para establecer las relaciones entre los casos de uso y las clases del análisis]

[Matriz propuesta para establecer las relaciones entre los casos de uso y los paquetes del diseño]

[Matriz propuesta para establecer las relaciones entre los paquetes del diseño y plan de pruebas de integración]

[Matriz propuesta para establecer las relaciones entre las clases del diseño y los componentes construidos]

[Matriz propuesta para establecer las relaciones entre los componentes y el plan de pruebas unitarias]

Identificador	Elemento	Trazabilidad hacia atrás	Trazabilidad hacia adelante
<i>[Identificador del elemento.]</i>	<i>[Nombre del caso de uso abstraído por el equipo de trabajo.]</i>	<i>[Identificador del elemento que origina el requerimiento.]</i>	<i>[Identificador del caso de prueba del sistema que verifica el caso de uso.]</i>
			<i>[Identificador del diagrama que muestra la</i>

Fase:

Responsable:

A.3.1 Plantilla de Matriz de Trazabilidad



AP Matriz de Trazabilidad

			<i>realización del caso de uso.]</i>
			<i>[Identificador del diagrama que muestra la navegación del caso de uso.]</i>
			<i>[Identificador de la clase del análisis que cubre la realización del caso de uso.]</i>
	<i>[Nombre del diagrama de realización.]</i>	<i>[Identificador del caso de uso que representa.]</i>	<i>[Identificador de las clases necesarias para la realización.]</i>
	<i>[Nombre del diagrama de navegación.]</i>	<i>[Identificador del caso de uso que representa.]</i>	<i>[Identificador de las pantallas de la interfaz de usuario.]</i>
	<i>[Nombre de la clase del análisis.]</i>	<i>[Identificador del diagrama de realización que la requiere.]</i>	<i>[Identificador del caso de prueba unitaria que verifica la clase.]</i>
			<i>[Identificador del paquete al que pertenece la clase.]</i>
	<i>[Nombre del paquete del diseño.]</i>	<i>[Identificador de las clases o requerimientos que lo generan.]</i>	<i>[Identificador del caso de prueba que verifica la integración del paquete.]</i>
	<i>[Nombre de la pantalla de interfaz de usuario.]</i>	<i>[Identificador del diagrama de navegación que la genera.]</i>	
	<i>[Nombre de los casos de prueba]</i>	<i>[Identificador del elemento que verifican.]</i>	

3. Información de apoyo

[La información de apoyo hace que la Matriz de Trazabilidad sea más fácil de usar. Incluye:

Tabla de contenidos

Apéndices.]

Fase:

Responsable:

A.3.2 Plantilla de Matriz de Trazabilidad



Índice

1.	Introducción	2
1.1	Definiciones, Acrónimos y Abreviaturas	2
1.2	Referencias	2
2.	Descripción general	2
2.1	Periodo	2
3.	Reporte de actividades	2
3.1	Actividades	2
3.2	Cambios	3
3.3	Riesgos	3
3.4	Resumen	3
4.	Reporte de mediciones y Sugerencias de mejora	4
4.1	Reporte de mediciones	4
4.2	Sugerencias de mejora	4
5.	Lecciones aprendidas	5
6.	Información de apoyo	5

Fase:

Responsable:

A.4.1 Plantilla de Reporte de Seguimiento



Reporte de Seguimiento

1. Introducción

[La introducción del Reporte de Seguimiento (RS) ofrece una visión general de todo el documento, a quién va dirigido y su objetivo]

1.1 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

1.2 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del Reporte de Seguimiento.]

2. Descripción general

[En esta sección se describen de manera general los factores que afectaron el proceso así como información relevante de lo ocurrido.]

2.1 Periodo

[Indicar de manera clara el periodo de tiempo que abarca este reporte. De dd-mm-aa a dd-mm-aa.]

3. Reporte de actividades

[Esta unidad atómica debe realizarse y presentarse periódicamente, es recomendable que sea de manera semanal.]

3.1 Actividades

[Listado de todas las actividades realizadas por el equipo durante el periodo de tiempo reportado.]

Actividad	Tiempo			Fecha		Tamaño del producto	Estado del producto
	Estimado	Real	Desviación	Estimada	Real		
<i>[nombre de la actividad]</i>	0:00	0:00	0.00%	<i>[dd-mm-aa]</i>	<i>[dd-mm-aa]</i>	0	<i>[Terminada/En proceso/Retrasada]</i>
...							
Totales	00:00	00:00	0.00%			0 y 0 LOC	

Fase:

Responsable:

A.4.2 Plantilla de Reporte de Seguimiento



3.2 Cambios

[El control de cambios efectuados a lo largo del periodo que reporta el seguimiento facilita el manejo del repositorio. Los cambios reportados aquí deben ser consistentes con el estado de la configuración.]

Nombre del producto	Descripción del cambio	Impacto del cambio	Solicitante	Estado del cambio
<i>[nombre del producto]</i>	<i>[Se solicita modificar ...]</i>	<i>[Alto/Medio/Bajo]</i>	<i>[miembro 1]</i>	<i>[En Proceso / Efectuado]</i>
...				

3.3 Riesgos

[Aquí se indicarán los posibles inconvenientes a los que el equipo se enfrentó. Los riesgos identificados deben ser registrados de manera clara, una breve oración en infinitivo bastaría para ejemplificar dicho riesgo.]

Además se debe establecer el plan que se siguió como respuesta a la ocurrencia, el impacto que tuvo en el desarrollo y el estado actual del riesgo.]

Descripción del riesgo	Plan de contingencia	Impacto del riesgo	Estado del riesgo
<i>[nombre del riesgo]</i>	...	<i>[Alto/Medio/Bajo]</i>	<i>[Identificado/Controlado/Mitigado]</i>
...			

3.4 Resumen

[Es conveniente resumir las mediciones tomadas en el periodo de tiempo que abarca el seguimiento, para ello se proponen tablas que incluyan la información más relevante de una manera breve y clara.]

Productos	
A tiempo:	0
Retrasados:	0
Adelantados:	0
Postergados:	0

Cambios	
Solicitados:	0
Rechazados:	0
Aceptados:	0
Efectuados:	0

Riesgos	
Identificados:	0
Resueltos:	0
Ignorados:	0

Fase:

A.4.3 Plantilla de Reporte de Seguimiento

Responsable:



4. Reporte de mediciones y Sugerencias de mejora

[Esta unidad atómica debe realizarse y presentarse periódicamente, es recomendable que sea únicamente al final de cada ciclo de desarrollo.]

4.1 Reporte de mediciones

[El informe de mediciones incluye el cálculo de los minutos invertidos en cada fase así como el tiempo total del ciclo, el tamaño de los productos generados en cada fase así como el total de productos elaborados.]

Por otra parte se llevará el control y registro de los defectos encontrados, elaborando una tabla listando todos los productos generados a lo largo del ciclo y divididos por fase en la que indique con exactitud cuántos defectos se encontraron en cada uno de los documentos o productos mencionados. Posteriormente encontrará el índice de errores por cada producto, es decir tomará en cuenta el número de defectos encontrados en un producto entre el tamaño de éste. Un número pequeño indicará que hubo pocos defectos y que el producto fue de buena calidad. Por el contrario, un número grande señala que el producto no es de buena calidad y deberá ser elaborado nuevamente.]

Este documento consta de dos partes, la primera resulta muy útil para la planeación de ciclos posteriores ya que se ha creado una idea de los tiempos empleados, y la segunda es un control que indica la calidad del producto resultante.]

Nombre del producto	Tamaño del producto	Defectos		Estado del producto
		Encontrados	Corregidos	
<i>[nombre del producto]</i>	0	0	0	<i>[Terminado/En proceso/Retrasado]</i>
...				
Totales	0 y 0 LOC	0	0	

4.2 Sugerencias de mejora

[Se identifican las mejores prácticas en los rubros del equipo y el proceso. Los problemas recurrentes tanto en el funcionamiento del equipo como en el proceso. Las experiencias exitosas también se registrarán. En cuanto a las sugerencias de mejora, se reflexiona sobre lo que habrá que mejorar tanto al proceso como al equipo.]

Fase:

A.4.4 Plantilla de Reporte de Seguimiento

Responsable:



5. Lecciones aprendidas

[En el documento de Lecciones Aprendidas, estarán englobadas las lecciones experimentadas por cada miembro del equipo tanto individual como colectivamente. Se recomienda analizar aquellas prácticas que el equipo considera que beneficiaron el desempeño o que resultaron en una experiencia exitosa, identificar los problemas más recurrentes y a éstos proponerles soluciones o sugerencias para minimizarlos o evitarlos.

Problemas a los que se enfrentaron

Riesgos importantes no considerados

Productos reusables

Recomendaciones para proyectos similares.]

6. Información de apoyo

[La información de apoyo hace que el RS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices.]

Apéndice B Plantillas de Desarrollo de Software



DSW Especificación de Requerimientos

Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Alcance	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
1.4	Referencias	2
2.	Descripción general	2
3.	Requerimientos No Funcionales	3
3.1	Usabilidad	3
3.2	Confiabilidad	3
3.3	Eficiencia	3
3.4	Mantenimiento	4
3.5	Portabilidad	4
3.6	Restricciones de diseño y construcción	4
3.7	Reusabilidad	4
3.8	Interfaces	4
3.9	Licenciamiento	4
3.10	Avisos Legales, Derechos de autor y otros	4
4.	Requerimientos Funcionales	5
4.1	Identificación de los casos de uso	5
4.2	Priorización de los casos de uso	5
4.3	Detallado de los casos de uso	5
4.4	Prototipo de la interfaz del usuario	7
5.	Plan de pruebas del sistema	7
5.1	Descripción	7
5.2	Requerimientos de las pruebas	7
5.3	Estrategia de las pruebas	7
5.4	Herramientas	8
5.5	Recursos	8
5.6	Casos de prueba	8
6.	Justificación de los requerimientos de software	8
7.	Información de apoyo	9

Fase:

B.1.1 Plantilla de Especificación de Requerimientos

Responsable:



Especificación de Requerimientos

1. Introducción

[La introducción de la Especificación de Requerimientos de Software (ERS) ofrece una visión general de todo el documento ERS. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la ERS.]

[Nota: La ERS captura completamente los requerimientos de software para el sistema, o una porción del sistema.]

1.1 Propósito

[Especifica el propósito de la ERS. Aquí se describe el comportamiento externo del sistema. También se deben mencionar los requerimientos no funcionales del sistema y proveer una descripción de los requerimientos funcionales.]

1.2 Alcance

[Breve descripción del alcance del sistema, descripción de la aplicación, características, asociaciones existentes y cualquier aspecto que se vea afectado por el desarrollo de éste.]

1.3 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.4 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido de la ERS. Debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción general

[En esta sección se describen de manera general los factores que afectan al producto y sus requerimientos. Aquí no se especifican aún los requerimientos, en cambio, se proveen los antecedentes a esos requerimientos que serán definidos y detallados más adelante en este mismo documento, con el objetivo de hacerlos más entendibles.]

- *Perspectiva del producto*
- *Funciones del producto*
- *Supuestos*
- *Licenciamiento*
- *Instalación*
- *Criterios de calidad*

Fase:

Responsable:

B.1.2 Plantilla de Especificación de Requerimientos



3. Requerimientos No Funcionales

[Esta sección contiene todos los requerimientos de software no funcionales, a un nivel de detalle lo suficientemente claro para que los miembros del equipo de desarrollo puedan satisfacerlos.]

3.1 Usabilidad

[En esta sección se incluyen los requerimientos que afectan la usabilidad, por ejemplo:

Especificar el tiempo de entrenamiento necesario para que un usuario normal se vuelva productivo al utilizar el sistema en ciertas operaciones

Especificar los tiempos que toma la realización de una tarea típica, pudiéndose basar en mediciones obtenidas al realizar tareas similares en otros sistemas con los que el usuario está familiarizado

Especificar si se va a guiar el desarrollo de acuerdo a algún estándar común de usabilidad]

3.1.1 <Requerimiento de usabilidad 1>

[Descripción del requerimiento.]

3.2 Confiabilidad

[Requerimientos de confiabilidad que el sistema deberá cumplir.

Disponibilidad— especificar el porcentaje de tiempo disponible, horas de uso, etc.

Tiempo promedio entre fallas (MTBF) — usualmente especificado en horas.

Tiempo promedio para repararla (MTTR) — cuánto tiempo estará el sistema fuera de operación después de ocurrir una falla?

Precisión — especificar la precisión que es requerida por las salidas del sistema.

Proporción de defectos — usualmente expresada en términos del número de errores entre cien líneas de código (bugs/KLOC) o errores entre puntos de función (bugs/function-point).

3.2.1 <Requerimiento de confiabilidad 1>

[Descripción del requerimiento.]

3.3 Eficiencia

[El desempeño del sistema será descrito en esta sección, incluyendo los tiempos promedio y máximo de respuesta específicos aplicables a cada caso de uso.

Tiempo de respuesta para una transacción

Rendimiento, transacciones por segundo

Capacidad, número de transacciones soportadas por el sistema en un instante dado

Utilización de recursos, cuánta memoria, disco, etc.

3.3.1 <Requerimiento de eficiencia 1>

[Descripción del requerimiento.]

Fase:

Responsable:

B.1.3 Plantilla de Especificación de Requerimientos



3.4 Mantenimiento

[En esta sección se indican los requerimientos de soporte y mantenimiento del sistema.]

3.4.1 <Requerimiento de mantenimiento 1>

[Descripción del requerimiento.]

3.5 Portabilidad

[Descripción de las características del software que permitirían su transferencia de un lugar a otro.]

3.6 Restricciones de diseño y construcción

[Aquí se indicarán las restricciones de diseño y construcción del sistema. Las restricciones de diseño son decisiones a las que se debe adherir el sistema, por ejemplo, lenguajes de programación, uso de herramientas de desarrollo específicas, componentes comprados, etc.]

3.6.1 <Restricción de diseño y construcción 1>

[Descripción del requerimiento.]

3.7 Reusabilidad

[En esta sección se describen los componentes que serán comprados o rehusados, que están sujetos algún licenciamiento o restricciones de uso.]

3.8 Interfaces

[En esta sección se definirán las interfaces que serán soportadas por la aplicación. Se deberán especificar de manera adecuada los protocolos de comunicación, puertos y direcciones lógicas, además de el cómo se desenvolverá contra estos requerimientos.]

3.8.1 Interfaces de Usuario

[Describe la interfaz de usuario a ser implementada, criterios ergonómicos.]

3.8.2 Interfaces de Hardware

[En esta sección se define cualquier interfaz de hardware que será soportada por el software, incluyendo la estructura lógica, direcciones lógicas, comportamiento esperado, etc.]

3.8.3 Interfaces de Software

[Aquí se describen las interfaces de software a otros componentes del sistema de software. This section describes software interfaces to other components of the software system. Éstos pueden ser componentes comprados, rehusados de otras aplicaciones o algunos otros desarrollados y fuera del alcance de la ERS pero que interactuará con el sistema.]

3.8.4 Interfaces de Comunicación

[Describe todas las interfaces de comunicación a otros sistemas o dispositivo, como redes LAN, dispositivos remotos, etc.]

3.9 Licenciamiento

[Aquí se definen todos los requerimientos de licenciamientos u otras restricciones.]

3.10 Avisos Legales, Derechos de autor y otros

[En esta sección se describe cualquier aviso legal, garantías, derechos de autor, patentes, marcas registradas y cuestiones de cumplimiento respecto al software.]

Fase:

Responsable:

B.1.4 Plantilla de Especificación de Requerimientos



4. Requerimientos Funcionales

4.1 Identificación de los casos de uso

[En esta sección se debe incluir una primera definición de los casos de uso identificados en el sistema, es decir una visión general de los casos de uso o que son aplicables al desarrollo. Se debe incluir una lista de los nombres de los actores y de cada caso de uso, junto con un diagrama que identifique las relaciones entre ellos. En esta sección se debe incluir una breve descripción de los supuestos y premisas existentes identificadas en los casos de uso.]

4.2 Priorización de los casos de uso

[De acuerdo a los requerimientos identificados en el punto anterior, se deben organizar y estructurar de acuerdo a las necesidades del negocio o a las funciones del sistema. Posteriormente se debe incluir una priorización de los casos de uso en Alta, Media o Baja, obteniendo finalmente un listado de los requerimientos priorizados.]

Identificador	Caso de Uso	Prioridad
<i>[identificador del caso de uso]</i>	<i>[Nombre o identificador del caso de uso]</i>	<i>[Alta / Media / Baja]</i>

4.3 Detallado de los casos de uso

[Esta sección debe incluir en primera instancia al actor, quien es un individuo ajeno al sistema que dispara el inicio de un caso de uso. Se debe proveer una descripción del caso de uso, en donde se escribirán los pasos que sigue el actor y el sistema para llevar a cabo el caso de uso. Las precondiciones de un caso de uso, son aquellas acciones que se dan por hecho en el instante inmediato anterior en el que el actor realice la acción, mientras que las poscondiciones son aquellos hechos que se esperan al concluir con la secuencia de pasos del caso de uso.]

Es necesario hacer notar que existe la posibilidad de que el flujo del caso de uso no sea normal y que ocurran excepciones o un flujo esperado pero no deseado, a esto se le llamará flujo excepcional y tiene que ver con el manejo de errores externos durante la ejecución del caso de uso.]

4.3.1 <Nombre e identificador del Caso de uso 1>

4.3.1.1 Descripción breve del caso de uso

[La descripción debe expresar brevemente el papel y el propósito del caso de uso, así como el actor involucrado. Un solo párrafo debería ser suficiente para esta descripción, en caso contrario debería considerarse estudiar si ese caso de uso puede subdividirse en más casos de uso de una complejidad menor.]

4.3.1.2 Flujo de eventos

4.3.1.2.1 Flujo básico

[Un caso de uso siempre comienza cuando el actor realiza una acción que lo dispara. El detallado del flujo de eventos, debe describir las acciones que realizan el actor y el sistema. Puede considerarse como una forma de diálogo entre el actor y el sistema.]

Fase:

B.1.5 Plantilla de Especificación de Requerimientos

Responsable:



El caso de uso debe describir lo que sucede dentro del sistema, pero no cómo ni por qué. Si se intercambia información, sea específico acerca de lo que se pasa de ida y vuelta. Por ejemplo, no es muy esclarecedor decir que el actor introduce información de los clientes. Es mejor decir, el actor introduce el nombre del cliente y la dirección.]

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1		2		-
3				

4.3.1.2.1.1 Listado de Excepciones

ID	TIPO	ACCIÓN
E1	<i>[tipo genérico de la excepción]</i>	<i>[acción efectuada al ocurrir la excepción]</i>
...		

4.3.1.2.1.2 Listado de descripción de Datos Requeridos

DATO	DESCRIPCIÓN	REQUERIDO
D1	<i>[descripción del dato requerido en el flujo del caso de uso]</i>	<i>[Si / No]</i>
...		

4.3.1.2.2 Flujo alternativo

[Cada flujo alternativo representa un comportamiento provocado por la ocurrencia de una excepción producida en el flujo principal. Cuando un flujo alternativo finaliza, los acontecimientos del flujo principal se reanudan a menos que se indique lo contrario.]

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
FA1		FA2		-
FA3				

Fase:

Responsable:

B.1.6 Plantilla de Especificación de Requerimientos



4.3.1.3 Precondiciones

[Una precondición para un caso de uso es el estado del sistema que debe estar presente antes de un caso de uso que se esté realizando.]

4.3.1.4 Poscondiciones

[Una poscondición de un caso de uso es una lista de posibles estados en el que se encuentre el sistema en uso inmediatamente después de que un caso de uso haya terminado.]

4.4 Prototipo de la interfaz del usuario

[El prototipo de la interfaz del usuario es el esqueleto de la interfaz que se presentará a los usuarios. Es conveniente asociar subconjuntos de pantallas con un caso de uso, esto permitirá ubicar y reconocer excepciones que se podrían presentar y cómo manejarlas, así como proponer una organización de los elementos de cada pantalla que permitan un uso más eficiente al usuario.

Debiendo tener especial atención en las siguientes coincidencias: los nombres y acciones de las pantallas y botones diseñados en el prototipo con lo descrito en los casos de uso.

Mediante un taller de requerimientos se puede construir un primer acercamiento del prototipo, este taller consiste en una reunión del equipo de trabajo encargado de la construcción de la interfaz y el cliente. En esta reunión se invita al cliente a tener una participación activa en el esbozo del prototipo, esto se puede lograr si el diseño se hace en vivo y en el lugar, por ejemplo utilizando un lienzo en el que el mismo cliente pueda plasmar sus ideas y el equipo de trabajo pueda proponer y evaluar varias alternativas.]

4.4.1 <Pantalla 1>

[Pantalla.]

5. Plan de pruebas del sistema

5.1 Descripción

[La prueba debe centrarse en los requerimientos que fueron rastreados directamente de los casos de uso. El objetivo de estas pruebas es verificar los datos de aceptación, procesamiento y recuperación, y la aplicación adecuada de las reglas de negocio.]

5.2 Requerimientos de las pruebas

[Aquí se debe incluir el listado de los requerimientos necesarios para obtener un ambiente que sea adecuado para realizar las pruebas y considerar a éstas válidas.]

5.3 Estrategia de las pruebas

[Al ser este tipo de pruebas basadas en técnicas de caja negra, es decir, verificar mediante una interacción con la aplicación a través de la interfaz gráfica de usuario (GUI) la salida o resultados. La estrategia primordialmente utilizará métodos para verificar los casos de uso con datos válidos, inválidos e incorrectos, además de tomar en cuenta casos de prueba que cumplan las reglas del negocio.]

5.3.1 Objetivo

[Garantizar funcionalidad, incluyendo la navegación, entrada de datos, procesamiento y recuperación y comportamiento.]

Fase:

Responsable:

B.1.7 Plantilla de Especificación de Requerimientos



5.3.2 Técnica

[Ejecutar cada caso de uso, flujo del caso de uso o de la función, utilizando datos válidos y no válidos, para verificar lo siguiente:

- *Los resultados esperados se producen cuando se utilizan datos válidos.*
- *El error o mensajes de advertencia ocurre al ingresar datos no válidos.*
- *Cada regla de negocio se aplique correctamente.]*

5.3.3 Criterio de terminación

[Todas las pruebas previstas se han ejecutado y todos los defectos identificados han sido registrados.]

5.3.4 Consideraciones especiales

[Identificar o describir los temas o asuntos (internos o externos) que afectan a la aplicación y ejecución de las pruebas.]

5.4 Herramientas

[Listado de las herramientas empleadas para cada grupo de casos de prueba.]

5.5 Recursos

[Esta sección presenta los recursos necesarios para llevar a cabo las pruebas, es un listado de las principales responsabilidades de cada recurso y el conocimiento y habilidades necesarias para completar las pruebas por parte de cada elemento.]

5.6 Casos de prueba

5.6.1 <Nombre del caso de uso 1>

Casos de prueba con datos válidos		
Caso de Uso	Entradas	Resultados Esperados
Casos de prueba con datos inválidos		
Caso de Uso	Entradas	Resultados Esperados
Casos de prueba par las reglas del negocio		
Caso de Uso	Entradas	Resultados Esperados

6. Justificación de los requerimientos de software

[Esta sección debe contener las justificaciones del porque se incluyeron los requerimientos definidos anteriormente.

Este planteamiento es especialmente importante en casos como:

Más de una alternativa puede satisfacer algunos de los requerimientos asignados al sistema.

Dos o más de los requerimientos del sistema se oponen.

El sistema debe interactuar con los sistemas ya existentes.]

Fase:

B.1.8 Plantilla de Especificación de Requerimientos

Responsable:



7. Información de apoyo

[La información de apoyo hace que la ERS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices

Estos pueden incluir la utilización de los casos de uso o prototipos de la interfaz de usuario.]



Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Alcance	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
1.4	Referencias	2
2.	Descripción general	2
3.	Representación de la Arquitectura	2
3.1	Metas de la arquitectura	2
3.2	Restricciones de la arquitectura	3
4.	Modelo del Análisis	3
5.	Clases del Análisis	3
5.1	Identificación de las clases	3
5.1.1	Clases de Interfaz o del paquete 1	3
5.1.2	Clases de entidad o del paquete 2	3
5.1.3	Clases de control o del paquete 3	3
6.	Realización de Caso de Uso - Análisis	3
6.1	Diagramas de interacción	4
6.2	Diagramas de navegación	4
7.	Justificación de las decisiones del Análisis	4
8.	Información de apoyo	4

Fase:

Responsable:

B.2.1 Plantilla de Análisis



Análisis

1. Introducción

[La introducción del Documento del Análisis de Software (DAS) ofrece una visión general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la fase de Análisis.]

1.1 Propósito

[Especifica el propósito del DAS. Se describe brevemente la estructura del documento, primordialmente se deberá indicar a quién va dirigido este documento así como la manera en que se espera sea interpretado el contenido de éste.]

1.2 Alcance

[Breve descripción del alcance del DAS, indicando qué es influenciado o afectado con el contenido de este documento.]

1.3 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.4 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del DAS. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción general

[Esta sección debe describir lo que el resto del DAS contiene y explica la forma en que el documento está organizado.]

3. Representación de la Arquitectura

[Esta sección describe los requisitos de software y los objetivos que tienen algún impacto significativo en la arquitectura, por ejemplo, la seguridad, privacidad, portabilidad, distribución y reutilización.]

3.1 Metas de la arquitectura

[Esta sección contiene los objetivos esperados del sistema que tienen que ver con la arquitectura del sistema, este punto debe contener los suficientes puntos de apoyo para sostener una discusión, y una posterior elección, de la arquitectura a utilizar durante la fase de Diseño.]

Fase:

B.2.2 Plantilla de Análisis

Responsable:



3.2 Restricciones de la arquitectura

[Esta sección contiene todas las restricciones a las que se enfrenta el desarrollo del sistema y que impactan en la arquitectura.]

4. Modelo del Análisis

[Esta sección contiene al conjunto de paquetes de más alto nivel identificados a partir de los requerimientos presentados en la ERS. Se representarán las abstracciones en subsistemas o capas del sistema.]

[Nota: si la decisión de la arquitectura ha sido tomada previamente por el cliente o existe una alternativa óptima cuya elección es irrevocable, este punto puede omitirse y especificarse hasta el DDS.]

5. Clases del Análisis

5.1 Identificación de las clases

[Esta sección contendrá los diagramas que describen el comportamiento estático del sistema y los tipos de relaciones existentes entre las clases. Se generará un diagrama de clases por cada paquete que contenga el sistema.]

Para generar el documento, es conveniente como primer paso identificar qué clases será necesario crear para que el sistema cumpla con los requerimientos establecidos, esto se realizará revisando cada caso de uso identificando qué necesidades cubre cada uno de éstos y con ello decidir qué clases cumplirán con ese trabajo y situarlas en el paquete adecuado.]

5.1.1 Clases de Interfaz o del paquete 1

[.]

5.1.2 Clases de entidad o del paquete 2

[.]

5.1.3 Clases de control o del paquete 3

[.]

6. Realización de Caso de Uso - Análisis

[Cada caso de uso deberá ser representado con un diagrama de secuencia, éstos muestran cómo se comportan los objetos entre ellos a través del tiempo. Representan el comportamiento dinámico de los casos de uso. Deberá tomarse en cuenta que por cada caso de uso deben generarse al menos dos diagramas de secuencia ejemplificando los flujos normales y aquellos que manejan excepciones, flujos excepcionales. Además se deberá identificar cada una de las clases participantes en el caso de uso para que se representen en el diagrama de secuencia. Estas clases deberán aparecer en alguno de los diagramas de clases.]

Para mostrar la navegación en el sistema se usarán diagramas de estado. Se construye un diagrama para indicar cómo navega cada tipo de usuario a través de las interfaces del sistema. Cada diagrama tendrá un estado inicial que se identificará con una circunferencia negra que rodea un punto negro, los estados serán representados por un rectángulo redondeado con el nombre de la



DSW Análisis

interfaz en la que se encuentra el usuario al realizar alguna acción, las acciones estarán determinadas en el documento por flechas con un estado origen, el evento que origina el cambio a un estado destino. Es necesario indicar aquellos eventos que generarían un error y manejarlos de acuerdo a sus necesidades, por ejemplo enviando al usuario a un estado donde pueda recuperarse para continuar sin necesidad de reiniciar todo el proceso.]

6.1 Diagramas de interacción

[Diagramas de secuencia.]

6.2 Diagramas de navegación

[Diagramas de estado.]

7. Justificación de las decisiones del Análisis

[Esta sección debe contener las justificaciones de las decisiones tomadas que resultaron del Análisis de los requerimientos del sistema definidos anteriormente.

Este planteamiento es especialmente importante en casos como:

Más de una alternativa puede satisfacer algunos de los requisitos asignados al sistema.

Dos o más de los asignados a los requisitos del sistema se oponen.

Existen restricciones de diseño.

La arquitectura del sistema o parte de ella se define por el cliente.

El sistema debe interactuar con los sistemas ya existentes.]

8. Información de apoyo

[La información de apoyo hace que el DAS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices]

Fase:

B.2.4 Plantilla de Análisis

Responsable:



Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Alcance	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
1.4	Referencias	2
2.	Descripción general	2
3.	Representación de la Arquitectura	2
3.1	Descripción de la arquitectura	2
4.	Vista Lógica	3
4.1	Descripción	3
4.2	Paquetes de la arquitectura	3
4.2.1	<Nombre del Paquete 1>	3
4.3	Clases del Diseño	3
4.3.1	<Diagrama de clases detallado>	3
5.	Vista del Proceso	3
5.1	Descripción	3
6.	Vista de Despliegue	3
6.1	Descripción	3
6.2	Modelo de despliegue	3
7.	Vista de Datos	3
7.1	Descripción	3
7.1.1	<Diagrama de la Base de Datos>	3
8.	Plan de Pruebas de Integración	4
8.1	Descripción	4
8.2	Subsistemas	4
8.3	Requerimientos para la integración	4
8.4	Estrategia de Integración	4
8.4.1	Objetivo	4
8.4.2	Técnica	4
8.4.3	Criterio de terminación	4
8.4.4	Consideraciones especiales	4
8.5	Herramientas	5
8.6	Recursos	5
8.7	Casos de prueba	5
8.7.1	<Iteración 1>	5
9.	Justificación de las decisiones del Diseño	5
10.	Información de apoyo	5

Fase:

Responsable:

B.3.1 Plantilla de Diseño



Diseño

1. Introducción

[La introducción del Documento de Diseño del Software (DDS) ofrece una visión general de todo el documento DDS. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la DDS.]

1.1 Propósito

[Especifica el propósito de la DDS. Este documento ofrece un panorama de la arquitectura del sistema, utilizando un número de diferentes puntos de vista arquitectónico para representar diferentes aspectos del sistema. Se tiene la intención de captar y transmitir las decisiones importantes de arquitectura que se han hecho en el sistema.]

1.2 Alcance

[Una breve descripción de la Arquitectura de Software, las decisiones del ambiente de implementación y lo que se ve afectado o influenciado por este documento.]

1.3 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.4 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido de la DDS. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción general

[Esta sección debe describir lo que el resto del DDS contiene y explica la forma en que el documento está organizado.]

3. Representación de la Arquitectura

[Esta sección describe la arquitectura del software para el sistema actual, y la forma en que estará representada. Enumera las vistas que son necesarias, y explica los tipos de modelos de elementos que contiene cada una de ellas.]

3.1 Descripción de la arquitectura

[Recoge las condiciones específicas que pueden aplicarse: el diseño y la estrategia de aplicación, el ambiente de desarrollo y su versión, las herramientas de diagramación para UML, si se requiere un manejador de bases de datos, cuál se usará y en qué versión, si la aplicación es para web, se define el servidor de aplicaciones, herramientas para automatizar las pruebas unitarias, etc.]



4. Vista Lógica

4.1 Descripción

[Esta sección describe de manera general la descomposición del modelo mediante jerarquía de paquetes.]

4.2 Paquetes de la arquitectura

[Por cada paquete se debe incluir un nombre, una breve descripción y un diagrama con las clases más importantes que contiene.]

4.2.1 <Nombre del Paquete 1>

4.2.1.1 Descripción breve del paquete

[La descripción debe expresar brevemente el propósito del paquete.]

4.3 Clases del Diseño

[Para cada clase del paquete se debe incluir un detallado de las clases identificadas, una descripción de las responsabilidades de la clase como sus atributos y operaciones.]

4.3.1 <Diagrama de clases detallado>

5. Vista del Proceso

5.1 Descripción

[Esta sección describe la descomposición del sistema en procesos y grupos de procesos. Los organiza de acuerdo a las relaciones de interacción que existen entre ellos. Describe los principales medios de comunicación utilizados en dichas interacciones.]

6. Vista de Despliegue

6.1 Descripción

[En esta sección se debe describir la configuración de la red física del software para su ejecución. Se deberán indicar los nodos (computadoras, routers, etc.) que son necesarios para el funcionamiento correcto del software, además de las interconexiones entre los nodos (bus, LAN, etc.). Idealmente se debe incluir el mecanismo o protocolo de comunicación entre los nodos.]

6.2 Modelo de despliegue

[Distribución de la arquitectura.]

7. Vista de Datos

7.1 Descripción

[Debe incluirse una descripción del mecanismo de almacenamiento de datos del sistema. Esta sección es opcional, si hay poca o ninguna persistente de datos, o la traducción entre el modelo de diseño y el modelo de datos es trivial.]

7.1.1 <Diagrama de la Base de Datos>

Fase:

B.3.3 Plantilla de Diseño

Responsable:



8. Plan de Pruebas de Integración

8.1 Descripción

[Contiene una breve descripción de cómo se planea realizar la integración del sistema. Se establece el orden de los componentes que se irán integrando. La recomendación es tomar en cuenta la arquitectura y la dependencia estática, es decir, identificar qué paquetes dependen de otros para funcionar correctamente.]

8.2 Subsistemas

[Listado de los subsistemas que se integrarán en la iteración, este listado debe estar establecido de acuerdo al orden de integración.]

8.3 Requerimientos para la integración

[Aquí se debe incluir el listado de los requerimientos necesarios para obtener un ambiente que sea adecuado para realizar la integración.]

8.4 Estrategia de Integración

[La integración será dividida en una serie de incrementos, cada incremento resulta en un producto funcional, a dicho producto debe de estar relacionado a un conjunto de casos de prueba de integración. Para cada paso de la integración se debe especificar cómo se construye y los criterios para su evaluación.]

8.4.1 Objetivo

[Detectar fallas de interacción entre las distintas unidades que componen al sistema, asegurando una funcionalidad conjunta.]

8.4.2 Técnica

[Establecer claramente el orden de integración de cada uno de los componentes, esto debe basarse en la dependencia de paquetes definida en la arquitectura construyendo un grafo de dependencias entre los sistemas, este grafo definirá un orden que ayudará a definir el alcance de cada incremento. El grafo se construye asociando a cada subsistema un nodo, y colocando una arista dirigida entre aquellos subsistemas con dependencia entre sí. La integración comenzará por aquellos nodos que no tienen aristas incidentes en ellos. Además se deberán incluir aspectos de:

- *Construcción*

Construir scripts e instrucciones de como debe integrarse el sistema

- *Evaluación y prueba*

Los criterios de evaluación, son una descripción de las cualidades que debe cumplir el producto integrado al final de cada incremento.

Instrucciones de instalación y configuración para ejecutar y probar la integración.]

8.4.3 Criterio de terminación

[Todas las pruebas previstas se han ejecutado y todos los defectos identificados han sido registrados. Cuando los casos de prueba, los procedimientos de prueba, scripts de prueba hayan arrojado los resultados esperados.]

8.4.4 Consideraciones especiales

[Identificar o describir los temas o asuntos (internos o externos) que afectan a la aplicación y ejecución de las pruebas.]

Fase:

Responsable:

B.3.4 Plantilla de Diseño



8.5 Herramientas

[Listado de las herramientas empleadas para la integración.]

8.6 Recursos

[Esta sección presenta los recursos necesarios para llevar a cabo la integración y las pruebas, es un listado de las principales responsabilidades de cada recurso y el conocimiento y habilidades necesarias para completar las pruebas por parte de cada elemento.]

8.7 Casos de prueba

8.7.1 <Iteración 1>

9. Justificación de las decisiones del Diseño

[Esta sección debe contener las justificaciones del por qué se tomaron las decisiones importantes del diseño del sistema definidas anteriormente.

Este planteamiento es especialmente importante en casos como:

Más de una alternativa puede satisfacer algunos de las necesidades del sistema.

Existen restricciones de diseño.

La arquitectura del sistema o parte de ella la define el cliente.

El sistema debe interactuar con sistemas ya existentes.]

10. Información de apoyo

[La información de apoyo hace que la DDS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices]



Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Alcance	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
1.4	Referencias	2
2.	Descripción general	2
3.	Componentes	2
3.1	A implementar	3
3.2	A modificar	3
4.	Plan de pruebas unitarias	3
4.1	Descripción	3
4.2	Requerimientos de las pruebas	3
4.3	Estrategia de las pruebas	3
4.3.1	Objetivo	3
4.3.2	Técnica	3
4.3.3	Criterio de terminación	3
4.3.4	Consideraciones especiales	4
4.4	Herramientas	4
4.5	Recursos	4
4.6	Casos de prueba	4
4.6.1	Pruebas de Caja Blanca	4
4.6.2	Pruebas de Caja Negra	4
5.	Reporte de pruebas unitarias	5
5.1	Pruebas de Caja Blanca	5
5.1.1	<Nombre de la clase>	5
5.1.2	<Descripción del defecto encontrado>	5
5.2	Pruebas de Caja Negra	5
5.2.1	<Nombre de la clase>	5
5.2.2	<Descripción del defecto encontrado>	5
6.	Justificación de las decisiones de la Construcción	5
7.	Información de apoyo	5

Fase:

B.4.1 Plantilla de Construcción

Responsable:



Construcción

1. Introducción

[La introducción del Documento de Construcción del Software (DCS) ofrece una visión general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la fase de Construcción.]

1.1 Propósito

[Especifica el propósito del DCS. Se describe brevemente la estructura del documento, primordialmente se deberá indicar a quién va dirigido este documento así como la manera en que se espera sea interpretado su contenido.]

1.2 Alcance

[Breve descripción del alcance del DCS, indicando qué es influenciado o afectado con el contenido de este documento.]

1.3 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.4 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del DCS. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción general

[Esta sección debe describir lo que el resto del DCS contiene y explica la forma en que el documento está organizado.]

3. Componentes

[Esta sección debe contener el listado de los componentes necesarios para la construcción del sistema.]

El grueso de la producción de esta fase es el código para las clases, dependiendo del lenguaje de programación señalado en la especificación de la implementación. El equipo establecerá un estándar para los archivos que contengan el código, pudiendo ser éste el propuesto por en el estándar del lenguaje o alguno ideado por los miembros del equipo. Dicho formato deberá facilitar la identificación de las clases y paquetes mediante la forma de nombrarlos y la distribución del código dentro del archivo.

Fase:

B.4.2 Plantilla de Construcción

Responsable:



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.



3.1 A implementar

[Esta sección contiene el listado de los componentes que se necesitan implementar totalmente, es decir, aquellos componentes que resultan nuevos y necesarios para cumplir con los requerimientos establecidos.]

3.2 A modificar

[Esta sección contiene el listado de los componentes que se necesitan modificar, es decir, aquellos componentes existentes en la configuración y que requieren de modificaciones para su posterior integración al sistema.

Es necesario incluir en el listado la versión y ubicación de los componentes a modificar.

4. Plan de pruebas unitarias

4.1 Descripción

[Con el plan de pruebas unitarias se busca validar la correcta ejecución de cada una de las instrucciones que fueron codificadas en cada una de las clases.]

4.2 Requerimientos de las pruebas

[Aquí se debe incluir el listado de los requerimientos necesarios para obtener un ambiente que sea adecuado para realizar las pruebas y considerarlas válidas.]

4.3 Estrategia de las pruebas

[Existen varios tipos de pruebas unitarias, la Estructural, basada en criterios de cobertura, tales como sentencias, decisiones, condiciones, bucles, caminos, etc.

Y la Funcional, basada en que para una entrada dada, se obtenga la salida esperada o establecida.]

4.3.1 Objetivo

[Identificar defectos relacionados a la lógica e implementación de cada una de las diferentes unidades que constituyen el sistema, aplicando estrategias funcionales o estructurales.]

4.3.2 Técnica

[En este caso se propone utilizar dos tipos de pruebas: pruebas de caja blanca que corresponden a aquellas que prueban la correcta ejecución de cada una de las instrucciones codificadas en las clases. Para determinar cuántos casos de prueba son necesarios es conveniente señalar todas las posibles trayectorias que un actor determinado tenga opción de seguir durante la ejecución del programa, tomando en cuenta los operadores condicionales que se encuentren en el código.

Y por otro lado, las pruebas de caja negra, que tomarán como punto de partida los posibles valores de entrada asignados por el usuario y los valores esperados de salida del sistema, validando los resultados esperados con los obtenidos.]

4.3.3 Criterio de terminación

[Todas las pruebas previstas se han ejecutado y todos los defectos identificados han sido registrados.]

Fase:

B.4.3 Plantilla de Construcción

Responsable:



4.3.4 Consideraciones especiales

[Identificar o describir los temas o asuntos (internos o externos) que afectan a la aplicación y ejecución de las pruebas.]

4.4 Herramientas

[Listado de las herramientas empleadas para cada grupo de casos de prueba.]

4.5 Recursos

[Esta sección presenta los recursos necesarios para llevar a cabo las pruebas, es un listado de las principales responsabilidades de cada recurso y el conocimiento y habilidades necesarias para completar las pruebas de cada elemento.]

4.6 Casos de prueba

4.6.1 Pruebas de Caja Blanca

4.6.1.1 <Nombre de la clase>

4.6.1.2 <Descripción de la clase>

Método	Recibe	Regresa	Excepción
<Nombre del método>	<tipo de dato>: <descripción del dato>	<tipo de dato>: <descripción del dato>	<descripción de la excepción>

4.6.2 Pruebas de Caja Negra

4.6.2.1 <Nombre de la clase>

4.6.2.2 <Descripción de la clase>

Método	Recibe	Esperado
<Nombre del método>	<valor de los datos recibidos>	<valor de los datos esperados>

Fase:

Responsable:

B.4.4 Plantilla de Construcción



5. Reporte de pruebas unitarias

[En el reporte de pruebas unitarias se registrarán aquellos casos de prueba que registraron algún defecto, en particular todos aquellos que no regresaron los valores esperados]

5.1 Pruebas de Caja Blanca

5.1.1 <Nombre de la clase>

Método	Recibió	Regresó	Defecto
<Nombre del método>	<tipo de dato>: <descripción del dato>	<tipo de dato>: <descripción del dato>	<defecto encontrado>

5.1.2 <Descripción del defecto encontrado>

5.2 Pruebas de Caja Negra

5.2.1 <Nombre de la clase>

Método	Recibió	Esperado	Regresó
<Nombre del método>	<valor de los datos recibidos>	<valor de los datos esperados>	<valor de los datos obtenidos>

5.2.2 <Descripción del defecto encontrado>

6. Justificación de las decisiones de la Construcción

[Esta sección debe contener las justificaciones de las decisiones tomadas durante la Construcción.]

7. Información de apoyo

[La información de apoyo hace que el DCS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices]

Fase:

B.4.5 Plantilla de Construcción

Responsable:



Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Background	2
1.3	Alcance	2
1.4	Definiciones, Acrónimos y Abreviaturas	2
1.5	Referencias	2
2.	Descripción general	3
3.	Reporte de Pruebas de Integración	3
3.1.1	<Iteración de Integración 1>	3
3.1.2	<Descripción de los defectos encontrados>	3
4.	Justificación de las decisiones de la Integración	3
5.	Información de apoyo	3



Integración

1. Introducción

[Se deben reportar los resultados obtenidos al final de cada serie de incrementos durante la integración. Cada incremento está definido en el plan de pruebas de integración, y éste resulta en un producto funcional, por lo que en ese punto se debe especificar cuáles de los criterios de evaluación fueron alcanzados.]

1.1 Propósito

[Especifica el propósito del Documento de Integración del Software (DIS). Se describe brevemente la estructura del documento, primordialmente se deberá indicar a quién va dirigido este documento así como la manera en que se espera sea interpretado el contenido.]

1.2 Background

[Proveer una breve descripción de las metas y objetivos de las pruebas de integración descritas en el Documento de Diseño del Software DDS. Incluir información como las principales funciones y características.]

1.3 Alcance

[Breve descripción del alcance del DIS, indicando qué o cómo es influenciado o afectado con el contenido de este documento.]

Listar las hipótesis formuladas durante el desarrollo de este documento que pueden tener repercusiones en el desarrollo o aplicación de las pruebas.

Listar los riesgos o contingencias que puedan afectar el desarrollo o aplicación de las pruebas.

Listar las limitaciones que pueden afectar el desarrollo o aplicación de las pruebas.]

1.4 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.5 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del DIS. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

Este rubro cobra importancia debido a que los planes de prueba no se encuentran en este documento.]

Fase:

Responsable:

B.5.2 Plantilla de Integración



2. Descripción general

[Esta sección debe describir lo que el resto del DIS contiene y explica la forma en que el documento está organizado.]

3. Reporte de Pruebas de Integración

[Esta sección debe contener el reporte de las pruebas efectuadas de acuerdo al Plan de Pruebas de Integración.]

3.1.1 <Iteración de Integración 1>

3.1.2 <Descripción de los defectos encontrados>

4. Justificación de las decisiones de la Integración

[Esta sección debe contener las justificaciones de las decisiones tomadas durante la Integración.]

5. Información de apoyo

[La información de apoyo hace que el DIS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices]



Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Background	2
1.3	Alcance	2
1.4	Definiciones, Acrónimos y Abreviaturas	2
1.5	Referencias	3
2.	Descripción general	3
3.	Requerimientos para las pruebas	3
4.	Reporte de pruebas del sistema	3
4.1	Casos de prueba	3
4.1.1	<Nombre del caso de uso 1>	3
5.	Estrategia de las pruebas	3
5.1	Tipos de pruebas	4
5.1.1	Pruebas de integridad de la Base de Datos y los Datos	4
5.1.2	Pruebas de funcionalidad	4
5.1.3	Pruebas de negocio	4
5.1.4	Pruebas de interfaz de usuario	4
5.1.5	Pruebas de desempeño	4
5.1.6	Pruebas de carga de trabajo	4
5.1.7	Pruebas de stress	4
5.1.8	Pruebas de volumen	4
5.1.9	Pruebas de seguridad y control de acceso	4
5.1.10	Pruebas de tolerancia y recuperación a fallos	4
5.1.11	Pruebas de configuración	4
5.1.12	Pruebas de Instalación	4
5.1.13	Pruebas de Aceptación	4
6.	Herramientas	5
7.	Recursos	5
8.	Justificación de las decisiones de las Pruebas	5
9.	Información de apoyo	5

Fase:

B.6.1 Plantilla de Integración

Responsable:



Pruebas

1. Introducción

[La introducción del Documento de Pruebas del Software (DPS) ofrece una visión general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la fase de Pruebas.]

1.1 Propósito

[Especifica el propósito del DPS. Se describe brevemente la estructura del documento, primordialmente se deberá indicar a quién va dirigido este documento así como la manera en que se espera sea interpretado el contenido de éste.]

1.2 Background

[Proveer una breve descripción de las metas y objetivos de las pruebas. Incluir información como las principales funciones y características.

Identificar información relativa al proyecto o a los componentes de software que serán probados.

- *Listar los requerimientos de alto nivel para las pruebas.*
- *Recomendar y describir estrategias de prueba para ser empleadas.*
- *Identificar los recursos requeridos y proveer un estimado del esfuerzo para las pruebas.*
- *Listar los entregables resultantes de la realización de las pruebas]*

1.3 Alcance

[Breve descripción del alcance del DPS, indicando qué es influenciado o afectado con el contenido de este documento.

Describir las etapas de las pruebas y los tipos de pruebas que serán realizadas, por ejemplo, de Desempeño o de Integridad.

Proveer un listado de los objetivos generales de las pruebas, en caso de ser necesario, incluir un listado de los componentes o tipos de prueba que no serán efectuadas.

Listar las hipótesis formuladas durante el desarrollo de este documento que pueden tener repercusiones en el diseño, desarrollo o aplicación de las pruebas.

Listar los riesgos o contingencias que puedan afectar el diseño, desarrollo o aplicación de las pruebas.

Listar las limitaciones que pueden afectar el diseño, desarrollo o aplicación de las pruebas.]

1.4 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

Fase:

Responsable:

B.6.2 Plantilla de Integración



1.5 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del DPS. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

Este rubro cobra importancia debido a que los planes de prueba no se encuentran en este documento.]

2. Descripción general

[Esta sección debe describir lo que el resto del DPS contiene y explica la forma en que el documento está organizado.]

3. Requerimientos para las pruebas

[Listado de los requerimientos generales para las pruebas, es decir, se debe describir el qué de las pruebas.]

4. Reporte de pruebas del sistema

[En esta sección se reportarán los resultados de aplicar los casos de prueba establecidos en el Plan de Pruebas del Sistema. Aquellas pruebas que arrojen resultados no esperados tendrán que ser reportadas como defectos encontrados]

4.1 Casos de prueba

4.1.1 <Nombre del caso de uso 1>

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Casos de prueba con datos inválidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Casos de prueba par las reglas del negocio			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos

5. Estrategia de las pruebas

[La estrategia de las pruebas presenta el enfoque recomendado para alcanzar los objetivos de éstas. Aquí se describe el cómo de las pruebas.]

Por cada prueba se debe proveer una descripción de cómo será planeada y efectuada.

En caso de que no se vaya a ejecutar alguna prueba, esto debe ser justificado adecuadamente.

Las principales consideraciones para la estrategia de pruebas son las técnicas que se utilizarán y el criterio para saber cuándo se ha completado la prueba.]

Fase:

Responsable:

B.6.3 Plantilla de Integración



5.1 Tipos de pruebas

[En caso de que el desarrollo necesite de más pruebas, se sugiere utilizar la siguiente tabla para definir las.]

Objetivo de la prueba	<i>[Metas del tipo de prueba a ejecutar.]</i>
Técnica de la prueba	<i>[Métodos para llevar a cabo las pruebas.]</i>
Criterio de terminación	<i>[Marca impuesta que indicará el término de la prueba.]</i>
Consideraciones especiales	<i>[Aspectos que deben tomarse en cuenta para la ejecución de las pruebas.]</i>
Casos de prueba	<i>[Listado de los casos de prueba.]</i>
Responsable	<i>[Miembro del equipo responsable de ejecutar las pruebas.]</i>

5.1.1 Pruebas de integridad de la Base de Datos y los Datos

5.1.2 Pruebas de funcionalidad

5.1.3 Pruebas de negocio

5.1.4 Pruebas de interfaz de usuario

5.1.5 Pruebas de desempeño

5.1.6 Pruebas de carga de trabajo

5.1.7 Pruebas de stress

5.1.8 Pruebas de volumen

5.1.9 Pruebas de seguridad y control de acceso

5.1.10 Pruebas de tolerancia y recuperación a fallos

5.1.11 Pruebas de configuración

5.1.12 Pruebas de Instalación

5.1.13 Pruebas de aceptación

Fase:

Responsable:

B.6.4 Plantilla de Integración



6. Herramientas

[Listado de las herramientas empleadas para cada grupo de pruebas.]

7. Recursos

[Esta sección presenta los recursos necesarios para llevar a cabo las pruebas, es un listado de las principales responsabilidades de cada recurso y el conocimiento y habilidades necesarias para completar las pruebas por parte de cada elemento.]

8. Justificación de las decisiones de las Pruebas

[Esta sección debe contener las justificaciones de las decisiones tomadas durante las Pruebas.]

9. Información de apoyo

[La información de apoyo hace que el DPS sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices]



Índice

1.	Introducción	2
1.1	Propósito	2
1.2	Alcance	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
1.4	Referencias	2
2.	Descripción general	2
3.	Manual de usuario	2
4.	Manual de mantenimiento	3
5.	Manual de operación	3
6.	Compendio de Seguimientos	3
7.	Justificación de las decisiones del Cierre	3
8.	Información de apoyo	3



Cierre

1. Introducción

[La introducción del Documento de Cierre (DC) ofrece una visión general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la fase de Cierre.]

1.1 Propósito

[Especifica el propósito del DC. Se describe brevemente la estructura del documento, primordialmente se deberá indicar a quién va dirigido este documento así como la manera en que se espera sea interpretado el contenido de éste.]

1.2 Alcance

[Breve descripción del alcance del DC, indicando qué es influenciado o afectado con el contenido de este documento.]

1.3 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
<i>[Término, acrónimo o abreviatura]</i>	<i>[Definición del término]</i>

1.4 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del DC. Éste debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción general

[Esta sección debe describir lo que el resto del DC contiene y explica la forma en que el documento está organizado.]

3. Manual de usuario

[La importancia de este documento para el usuario es prioritaria, será a lo primero que se dirija al ocurrir alguna duda mientras utiliza el sistema. Deberá estar escrito en un lenguaje que pueda entender el usuario, evitar que sea ambiguo y ser una guía para cualquier tarea realizada en el sistema.

En cuanto a la organización del manual, deberá contener un índice del contenido en el que se distingan de manera clara los temas tratados. Deberá estar organizado acorde a las necesidades del usuario, esto es, que el usuario pueda identificar en dónde se encuentra su duda o problema.

Es aconsejable que el manual indique paso a paso cómo realizar las tareas, para identificar cada una de éstas. El título de cada sección indicará de manera precisa qué funcionalidad llevará a

Fase:

Responsable:

B.7.2 Plantilla de Cierre



DSW Cierre

cabo, y comenzará la descripción del proceso desde la primera interfaz con que el usuario tenga contacto, describiendo cada objeto que el usuario observa en la pantalla e indicando su utilidad y cómo llevarla a cabo. Se podrá incluir en el manual la imagen de cada interfaz para un mejor entendimiento para el cliente. Cabe mencionar la opción de generar no sólo un manual físico, sino también un manual digital al que el usuario pueda acceder de manera rápida y mientras usa el sistema. Este manual puede considerar situaciones comunes que se le presentan al usuario y cuya solución no requiere de una extensa explicación.]

4. Manual de mantenimiento

[Este manual incluye la documentación generada en las fases de Especificación de Requerimientos, Diseño, Construcción, Integración y Pruebas separados de acuerdo a los ciclos a que pertenecen.

Este manual claramente estará compuesto por un lenguaje técnico y los estándares que el equipo haya impuesto al inicio del proceso de desarrollo, es fundamentalmente para uso exclusivo de los Ingenieros de Desarrollo del proyecto o en su defecto para Ingenieros encargados de darle mantenimiento o escalar al sistema.]

5. Manual de operación

[El manual de operación deberá describir los pasos necesarios para instalar el sistema, los requerimientos mínimos de hardware y software con sus versiones necesarias en las computadoras donde residirá el sistema.

Se incluirán los datos de contacto de los desarrolladores del software en caso de que el usuario final tenga problemas con la instalación del sistema. El proceso de instalación se puede incluir en el Manual de Usuario de ser considerado necesario.]

6. Compendio de Seguimientos

[De ser necesario se deberá generar un compendio de los seguimientos, resumiendo los rubros importantes y mostrando las sumas totales de las métricas tomadas a lo largo del proyecto. Deberá ser lo más conciso y proveer la información.]

7. Justificación de las decisiones del Cierre

[Esta sección debe contener las justificaciones de las decisiones tomadas durante el Cierre.]

8. Información de apoyo

[La información de apoyo hace que el Cierre sea más fácil de usar. Incluye:

Tabla de contenidos

Índice

Apéndices]

Fase:

B.7.3 Plantilla de Cierre

Responsable:

Apéndice C Plantillas de Mantenimiento de Software



MSW Petición de Modificación

Petición de Modificación

Prioridad	<i>[Normal / Urgente]</i>	
Descripción del error/adaptación	<p><i>[En este apartado se deberán responder las preguntas del estilo:</i></p> <p><i>¿A quién se le presentó el error?</i></p> <p><i>¿Cuándo sucedió el error?</i></p> <p><i>¿Cómo ocurrió el error?</i></p> <p><i>Dichas preguntas deberán ser respondidas por el usuario del sistema a quien se le presentó el error. El objetivo de esta sección es establecer las acciones que dispararon el error.]</i></p>	
Entorno de ocurrencia	<i>[En este apartado se debe incluir un listado de las características presentes en el entorno del sistema en el momento en que se presentó el error.]</i>	
Aspectos afectados	<i>[En esta sección se deberán listar los aspectos que se ven afectados por el defecto descrito o la ocurrencia del error.]</i>	
Solicitado por	<i>[nombre de la persona que realiza la petición, en caso de ser distinta al usuario al que le ocurrió será necesario incluir el nombre de éste]</i>	
Diagnóstico y posibles soluciones	<i>[En esta sección se deben proponer posibles soluciones al error encontrado, así como el diagnóstico del error, es decir la naturaleza del error mediante la observación de las señales o causantes de éste.]</i>	
Información de apoyo	<i>[La información de apoyo complementa aspectos no incluidos en esta plantilla y que se considera oportuno incluir.]</i>	
Autorización	<p>Revisado por</p> <p>Responsable del Mantenimiento del Software</p> <p><i>[nombre]</i></p>	<p>Aceptado por</p> <p>Cliente</p> <p><i>[nombre]</i></p>

Fase:

Responsable:

C.1 Plantilla de Petición de Modificación



Intervención

Tipo de intervención	<i>[Planificable / No planificable]</i>	
Equipo mantenimiento de	<i>[En este apartado se deberá incluir el listado de las personas y los roles encargados de dar atención a esta intervención]</i>	
Recreación entorno del	<i>[En este apartado se debe incluir un listado de las características presentes en el entorno del sistema en el momento en que se presentó el error.]</i>	
Componentes involucrados	<i>[En esta sección se deberán listar los aspectos que se ven afectados por el defecto descrito o la ocurrencia del error.]</i>	
Premisas restricciones y	<i>[Listado de supuestos a tomar en cuenta para realizar la modificación, las restricciones delimitan las acciones.]</i>	
Componentes corregir a	<i>[Listado de los componentes que serán modificados durante el mantenimiento. Es necesario tener en cuenta la trazabilidad de los componentes para poder prever el alcance de las modificaciones en el sistema.]</i>	
Acciones correctivas	<i>[Listado de las medidas que se ejecutarán, elegidas a partir del diagnóstico, como soluciones a la petición de modificación.]</i>	
Pruebas unitarias	<i>[Las pruebas unitarias buscan validar la correcta ejecución de cada una de las instrucciones que fueron codificadas en cada una de las clases modificadas.]</i>	
Pruebas integración de	<i>[En caso de que el mantenimiento sea planificable. Se establece el orden de los componentes que se irán integrando. Identificando qué paquetes modificados dependen de otros para funcionar correctamente.]</i>	
Pruebas de regresión	<i>[En caso de que el mantenimiento sea planificable. Las pruebas de regresión tienen como objetivo detectar los posibles errores introducidos como resultado de la intervención.]</i>	
Plan de retirada	<i>[Se definen la fecha y los pasos para llevar a cabo la retirada del software.]</i>	
Información apoyo de	<i>[La información de apoyo complementa aspectos no incluidos en esta plantilla y que se considera oportuno incluir.]</i>	

Autorización	Revisado por	Aceptado por
	Responsable del Mantenimiento del Software <i>[nombre]</i>	Cliente <i>[nombre]</i>

C.2 Plantilla de Intervención

Apéndice D Plantillas de PmCOMPETISOFT



PM Propuestade Mejora

Propuesta de Mejora

Ciclo de Mejora	
Nombre de la empresa	[nombre de la empresa que realizará el proceso de mejora]
Nombre del proyecto de mejora	[nombre que identifica al proceso de mejora]
Objetivos y alcance del ciclo de mejora	
Necesidades del negocio	[Listado de aquellas necesidades de la organización que son candidatas a sufrir una mejora]
Objetivos de mejora generales	[Listado de metas que se pretenden alcanzar al obtener la mejora]
Alcance del ciclo	[Aspectos que se ven afectados a través del proceso de mejora]
Proceso de Mejora Continua de Procesos	[Nombre del modelo de referencia que guiará la mejora]
Modelo de procesos	[Nombre del modelo de procesos de software que sigue la organización]
Método de evaluación	[Nombre del método de evaluación que calificará los resultados de la mejora]
Recursos del ciclo de mejora	
Recursos del ciclo de mejora	[Listado de recursos necesarios para completar el ciclo de mejora, principalmente de costo y tiempo]
Asesor de mejora del Proyecto COMPETISOFT	[Nombre del experto que guiará la mejora]
Responsable de Mejora de Procesos de la organización	[Nombre de la persona encargada de la mejora dentro de la organización]
Grupo de Mejora de Procesos	[Listado de los integrantes del grupo que coordina, planea y ejecuta la mejora en la organización]
Grupo de Gestión de Mejora	[Listado de los integrantes del grupo que gestionará la mejora, un representante del grupo directivo, otro del grupo de mejora de procesos y el responsable de la mejora del proceso]
Responsable de Mejora de Procesos	[Individuo que diseñará y guiará la implantación de la mejora, además de capacitar y entrenar al personal]
Evaluador	[Nombre del individuo encargado de evaluar los resultados de la mejora]
Otra información relevante	
[Registre aquí otra información que considere relevante]	
Aprobación del Plan Preliminar de Mejora	
Líder de Mejora de la Empresa [nombre]	Responsable de COMPETISOFT [nombre]
Sugerencias de mejora a esta plantilla	
[Registre aquí sugerencias de mejora para esta plantilla.]	

D.1.1 Plantilla de Propuesta de Mejora



Plan General de Mejora

Ciclo de Mejora			
Nombre de la empresa	<i>[nombre de la empresa que realizará el proceso de mejora]</i>		
Nombre del proyecto de mejora	<i>[nombre que identifica al proceso de mejora]</i>		
Grupo de Mejora de Procesos	<i>[Listado de los integrantes del grupo que coordina, planea y ejecuta la mejora en la organización]</i>		
Grupo de Gestión de Mejora	<i>[Listado de los integrantes del grupo que gestionará la mejora, un representante del grupo directivo, otro del grupo de mejora de procesos y el responsable de la mejora del proceso]</i>		
Responsable de Mejora de Procesos	<i>[Individuo que diseñará y guiará la implantación de la mejora, además de capacitar y entrenar al personal]</i>		
Nivel de capacidad actual y esperado de los procesos a mejorar en este Ciclo			
<i>Proceso a mejorar</i>	<i>Nivel de Capacidad Actual</i>	<i>Nivel de Capacidad Esperado</i>	
<i>[Listado de los procesos priorizados a mejorar]</i>	<i>[Nivel de capacidad actual del proceso, de 0 a 4]</i>	<i>[Nivel de capacidad esperado del proceso, de 1 a 5]</i>	
Número Iteración del Ciclo de Mejora			
Número de iteraciones del ciclo de mejora	<i>[número de iteraciones de mejora]</i>		
<i>[Tabla en la que se muestra la planeación de las iteraciones]</i>			
Iteración	Proceso	Duración	Desde
<i>[número de iteración]</i>	<i>[Proceso a mejorar]</i>	<i>[Duración planeada de la mejora del proceso]</i>	<i>[Fecha de inicio de la mejora del proceso]</i>

Fase:

D.2.1 Plantilla de Plan General de Mejora

Responsable:



Planeación General

[Planeación general de las actividades a realizar para cada iteración del ciclo de mejora. Aquí se deberá estimar el esfuerzo en tiempo necesario para completar la mejora.]

Calendario						
Identificador	Descripción de la actividad	Unidades de calendario	Tiempo estimado de la mejora			
			Periodo 1			Periodo n

Actividades	Semanas															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Instalación del ciclo																
Diagnóstico de los procesos																
Formulación de la iteración 1																
Mejora proceso de la iteración 1																
Evaluación mejoras iteración 1																
Formulación de la iteración 2																
Mejora proceso de la iteración 2																
Evaluación mejoras iteración 2																
Revisión del ciclo de mejora																

Actividad realizada
 Actividad planeada y por realizar

Plan de manejo de riesgos

Riesgos	
Nombre	<i>[nombre del riesgo]</i>
Descripción	<i>[breve descripción del riesgo]</i>
Probabilidad de ocurrencia	<i>[alta / media / baja]</i>
Impacto del riesgo	<i>[alto / medio / bajo]</i>
Plan de contingencia	<i>[acciones a seguir si el riesgo ocurre]</i>
Estado del riesgo	<i>[identificado / controlado / mitigado / ignorado]</i>

Fase:

D.2.2 Plantilla de Plan General de Mejora

Responsable:



Plan de capacitación	
Capacitaciones	
Nombre del curso	<i>[nombre del curso]</i>
Descripción	<i>[breve descripción del curso]</i>
Tipo	<i>[presencial / en línea]</i>
Horas totales	<i>[número de horas de duración del curso]</i>
Responsable	<i>[persona física autorizada a dar por cerrado el contrato]</i>
Número de participantes	<i>[cupo de participantes en este curso]</i>
Fecha de inicio	<i>[dd/mm/aaaa]</i>
Fecha de término	<i>[dd/mm/aaaa]</i>
Resultado esperado	<i>[breve descripción de lo que se espera obtener mediante este curso]</i>
Plan de Mediciones	
<i>[Aquí se deberán registrar las mediciones de los procesos, pudiendo ser éstas:</i>	
<ol style="list-style-type: none"> 1. <i>Los procesos a mejorar. Basado en obtener el nivel de capacidad de los procesos de la organización.</i> 2. <i>El proceso de mejora utilizado. Midiendo el esfuerzo de realizar este proceso.]</i> 	
Otra información relevante	
<i>[Registre aquí otra información que considere relevante]</i>	
Aprobación del Plan Preliminar de Mejora	
Líder de Mejora de la Empresa <i>[nombre]</i>	Responsable de COMPETISOFT <i>[nombre]</i>
Sugerencias de mejora a esta plantilla	
<i>[Registre aquí sugerencias de mejora para esta plantilla.]</i>	

Fase:

D.2.3 Plantilla de Plan General de Mejora

Responsable:



Plan de Implementación de Mejora

Ciclo de Mejora					
Nombre de la empresa		<i>[nombre de la empresa que realizará el proceso de mejora]</i>			
Nombre del proyecto de mejora		<i>[nombre que identifica al proceso de mejora]</i>			
Grupo de Mejora de Procesos		<i>[Listado de los integrantes del grupo que coordina, planea y ejecuta la mejora en la organización]</i>			
Grupo de Gestión de Mejora		<i>[Listado de los integrantes del grupo que gestionará la mejora, un representante del grupo directivo, otro del grupo de mejora de procesos y el responsable de la mejora del proceso]</i>			
Responsable de Mejora de Procesos		<i>[Individuo que diseñará y guiará la implantación de la mejora, además de capacitar y entrenar al personal]</i>			
Iteración					
Identificador		<i>[identificador de la mejora]</i>			
Caso de mejora		<i>[Nombre del caso de mejora]</i>			
Descripción del Proceso					
<i>[descripción del procesos a mejorar, se deberán incluir las actividades y productos del proceso que se desea mejorar]</i>					
Objetivos de la iteración					
<i>[listado de los objetivos que se desean alcanzar al finalizar la iteración]</i>					
Estimación para el caso de mejora					
Fecha	Actividad	Nombre ó Rol de las personas involucradas	Horario	Tiempo Asesor (minutos)	Tiempo Empresa (minutos)
Total separado					
Total consolidado					

Fase:

D.3.1 Plantilla de Plan de Implementación de Mejora

Responsable:



Planeación de la iteración					
<i>[Estrategia de la iteración y cronograma]</i>					
			Semanas		
No	Rol Responsable	Actividades	1	...	n
<i>[número de identificación de la actividad]</i>	<i>[persona encargada de velar por la realización de la actividad]</i>	<i>[listado de las actividades a llevar a cabo durante la iteración]</i>			
Resultados destacables de la iteración					
<i>Evaluación de los casos de mejora desarrollados</i>	<i>[evaluaciones de los casos de mejora llevados a cabo, para verificar su eficiencia]</i>				
<i>Logros Alcanzados durante la mejora</i>	<i>[listado de los logros alcanzados a través de la mejora]</i>				
<i>Lecciones aprendidas.</i>	<i>[listado de las lecciones aprendidas durante el desarrollo de la iteración]</i>				
Incidencias					
<i>Desviación respecto a la planificación</i>	<i>[registro de la desviación en tiempo o recursos respecto a la planeación]</i>				
<i>Clave del éxito o fracaso</i>	<i>[situaciones identificadas como determinantes para el éxito o fracaso de la iteración]</i>				
<i>Dificultades encontradas</i>	<i>[listado de las dificultades enfrentadas a lo largo de la iteración]</i>				
<i>Solución propuesta a cada una de las dificultades anteriores</i>	<i>[listado de soluciones propuestas para enfrentar a las dificultades presentadas]</i>				
Sugerencia para la planeación de la siguiente iteración					
<i>[sugerencias a tomar en cuenta para la siguiente iteración]</i>					
Plan de Mediciones					
<i>[Aquí se deberán registrar las mediciones de los procesos, pudiendo ser éstas:</i>					
1. <i>Los procesos a mejorar. Basado en obtener el nivel de capacidad de los procesos de la organización.</i>					
2. <i>El proceso de mejora utilizado. Midiendo el esfuerzo de realizar este proceso.]</i>					
Otra información relevante					
<i>[Registre aquí otra información que considere relevante]</i>					
Aprobación de la Implementación de Mejora					
Líder de Mejora de la Empresa <i>[nombre]</i>			Responsable de COMPETISOFT <i>[nombre]</i>		
Sugerencias de mejora a esta plantilla					
<i>[Registre aquí sugerencias de mejora para esta plantilla.]</i>					

D.3.2 Plantilla de Plan de Implementación de Mejora



Reporte de Mejora

Ciclo de Mejora																	
Nombre de la empresa	<i>[nombre de la empresa que realizará el proceso de mejora]</i>																
Nombre del proyecto de mejora	<i>[nombre que identifica al proceso de mejora]</i>																
Evaluación de la mejora																	
Evaluador	<i>[Nombre del encargado de llevar a cabo la evaluación]</i>																
Evaluación inicial	<p><i>[Valores resultantes de la evaluación inicial de capacidades]</i></p> <table border="1"> <thead> <tr> <th colspan="4">Evaluación inicial</th> </tr> <tr> <th>Atributos del proceso</th> <th>A. 1.</th> <th>...</th> <th>A. n.</th> </tr> </thead> <tbody> <tr> <td>Valoración</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Nivel de capacidad</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Evaluación inicial				Atributos del proceso	A. 1.	...	A. n.	Valoración				Nivel de capacidad			
Evaluación inicial																	
Atributos del proceso	A. 1.	...	A. n.														
Valoración																	
Nivel de capacidad																	
Evaluación final	<p><i>[Valores resultantes de la evaluación final de capacidades]</i></p> <table border="1"> <thead> <tr> <th colspan="4">Evaluación final</th> </tr> <tr> <th>Atributos del proceso</th> <th>A. 1.</th> <th>...</th> <th>A. n.</th> </tr> </thead> <tbody> <tr> <td>Valoración</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Nivel de capacidad</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Evaluación final				Atributos del proceso	A. 1.	...	A. n.	Valoración				Nivel de capacidad			
Evaluación final																	
Atributos del proceso	A. 1.	...	A. n.														
Valoración																	
Nivel de capacidad																	
Procesos mejorados	<i>[Listado de procesos cuya evaluación arrojó una mejora]</i>																
Esfuerzo involucrado	<p><i>[Esfuerzo total que fue necesario para llevar a cabo cada actividad de la mejora]</i></p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Actividad</th> <th>Nombre ó Rol de las personas involucradas</th> <th>Tiempo</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td colspan="3" style="text-align: right;"><i>Total separado</i></td> <td></td> </tr> <tr> <td colspan="3" style="text-align: right;"><i>Total Consolidado</i></td> <td></td> </tr> </tbody> </table>	Fecha	Actividad	Nombre ó Rol de las personas involucradas	Tiempo					<i>Total separado</i>				<i>Total Consolidado</i>			
Fecha	Actividad	Nombre ó Rol de las personas involucradas	Tiempo														
<i>Total separado</i>																	
<i>Total Consolidado</i>																	
Modelo de procesos	<i>[Nombre del modelo de procesos tomado como referencia]</i>																
Método de evaluación	<i>[Descripción del método de evaluación llevado a cabo]</i>																

Fase:

Responsable:

D.4.1 Plantilla de Reporte de Mejora



PM Reporte de Mejora

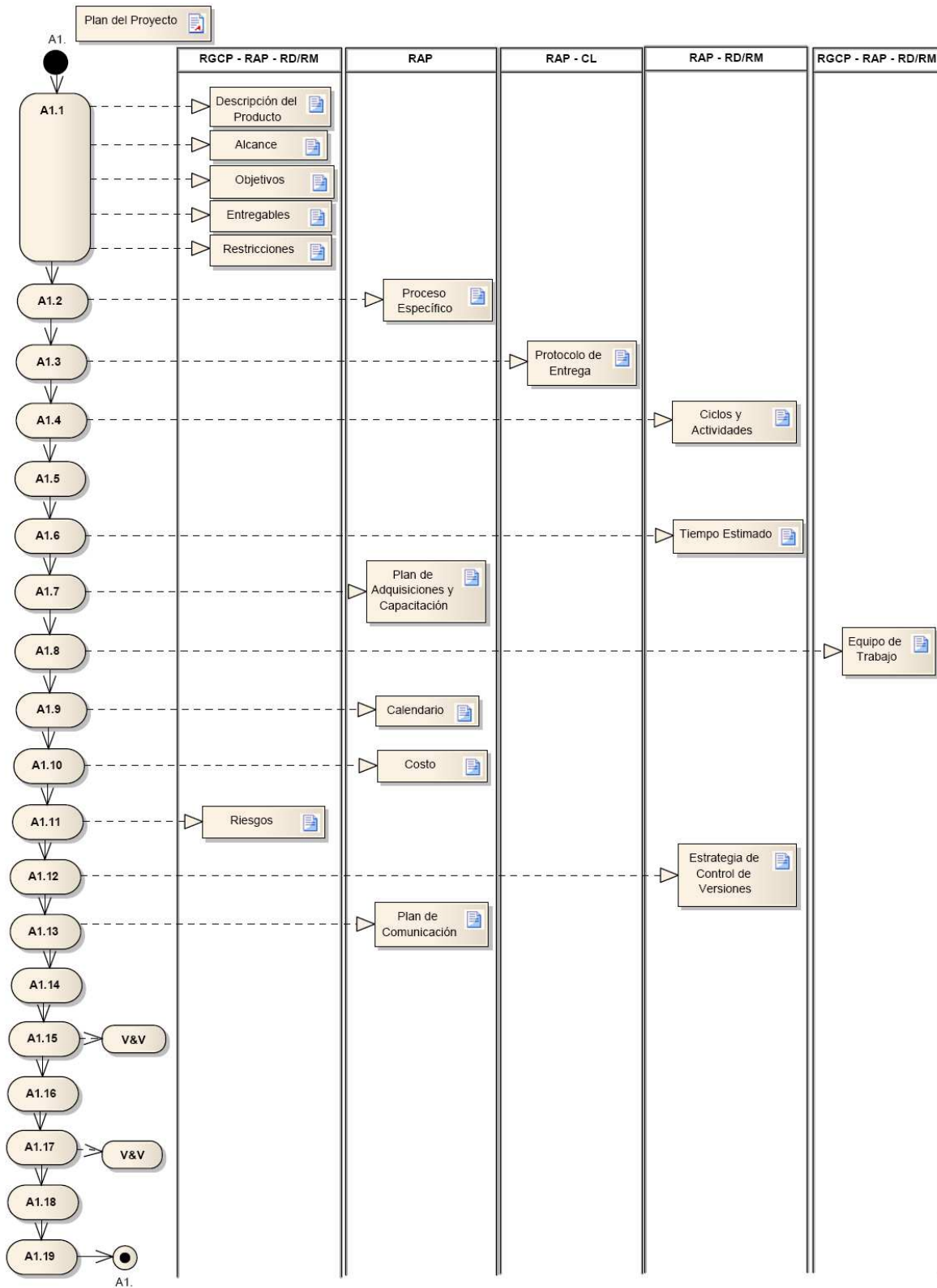
Registro de mejores prácticas	
Logros alcanzados	<i>[Listado de los logros alcanzados al obtener la mejora]</i>
Lecciones aprendidas	<i>[Listado de las lecciones aprendidas durante la mejora]</i>
Propuestas de mejora	<i>[Listado de propuestas a tomar en cuenta para mejorar un futuro ciclo de mejora]</i>
Recomendaciones de ajuste al proceso de mejora	<i>[Listado de recomendaciones para ajustar un siguiente ciclo de mejora]</i>
Otra información relevante	
<i>[Registre aquí otra información que considere relevante]</i>	
Aprobación del Reporte de Mejora	
Líder de Mejora de la Empresa <i>[nombre]</i>	Responsable de COMPETISOFT <i>[nombre]</i>
Sugerencias de mejora a esta plantilla	
<i>[Registre aquí sugerencias de mejora para esta plantilla.]</i>	

Fase:

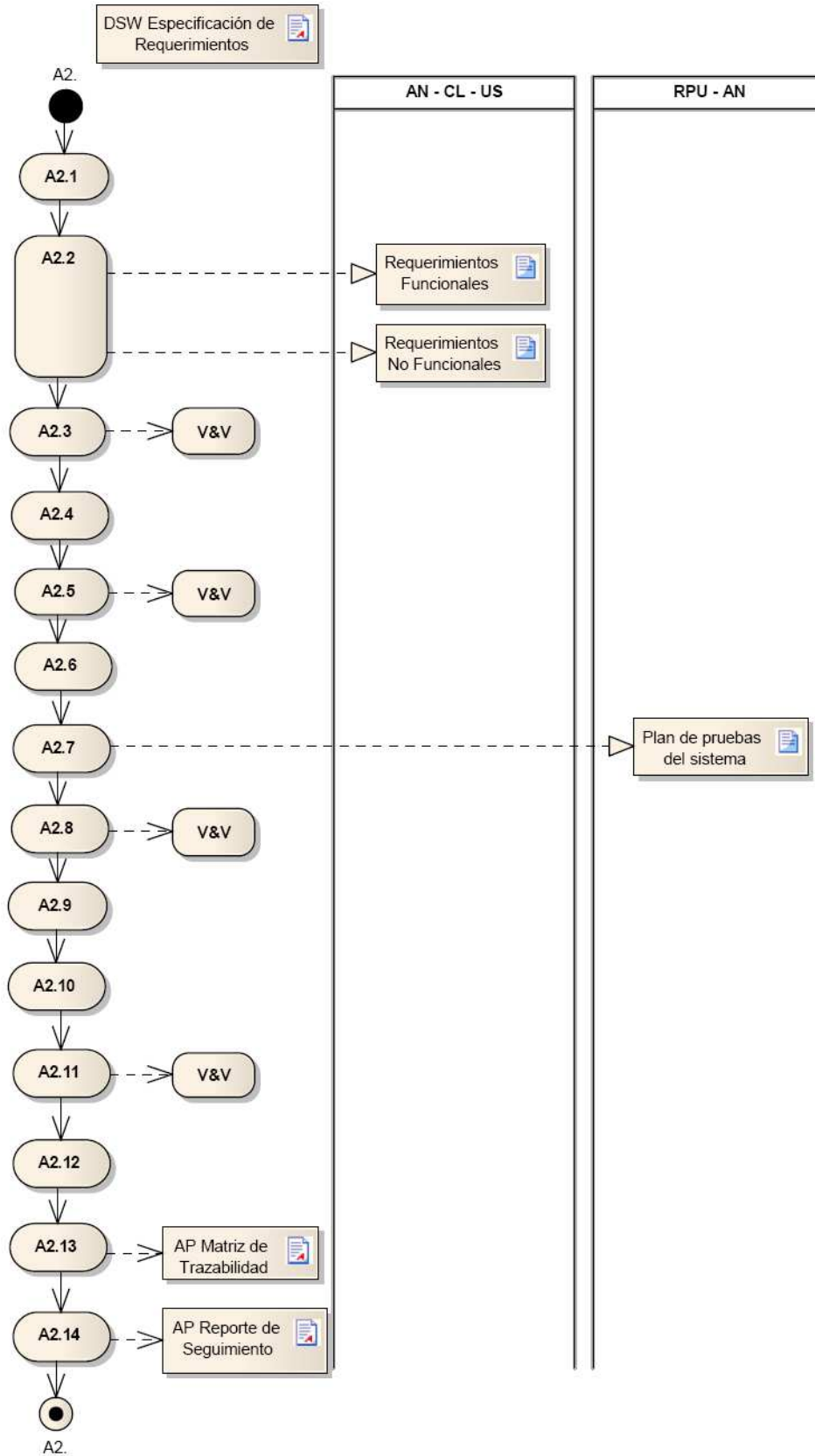
D.4.2 Plantilla de Reporte de Mejora

Responsable:

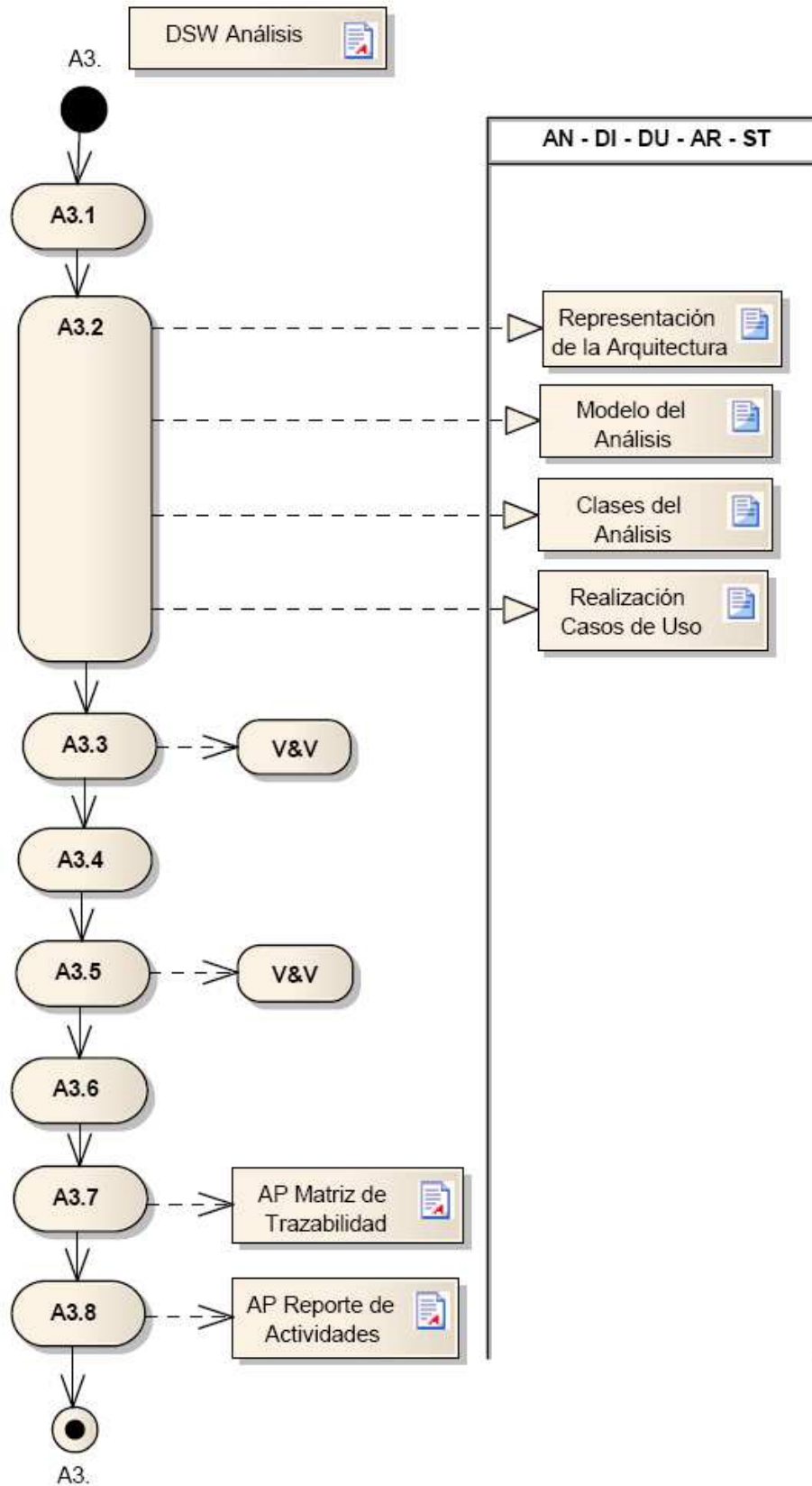
Apéndice E Diagramas de Unidades Atómicas



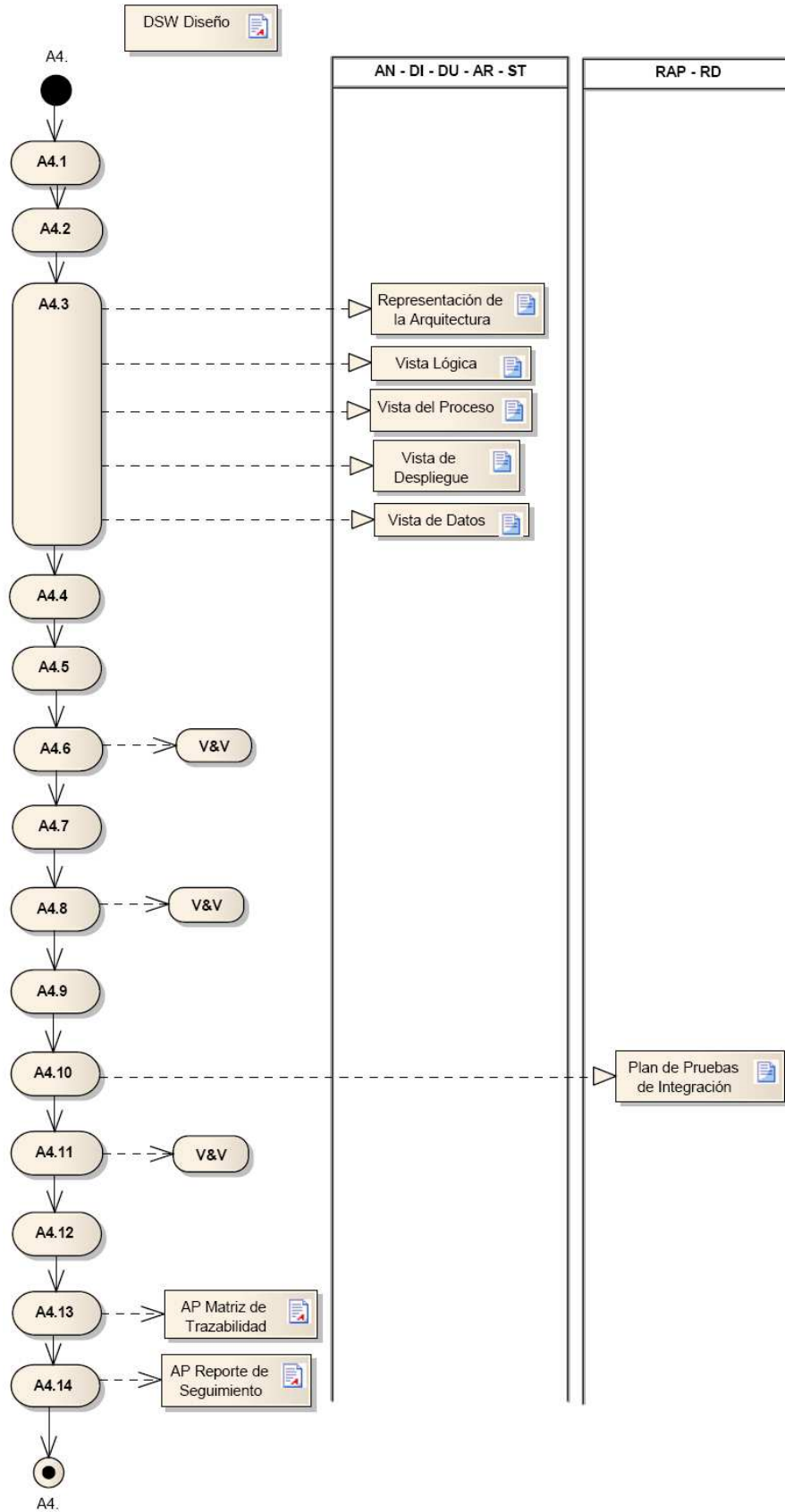
E.1 Diagrama de Unidades Atómicas correspondiente a la plantilla de Plan del Proyecto



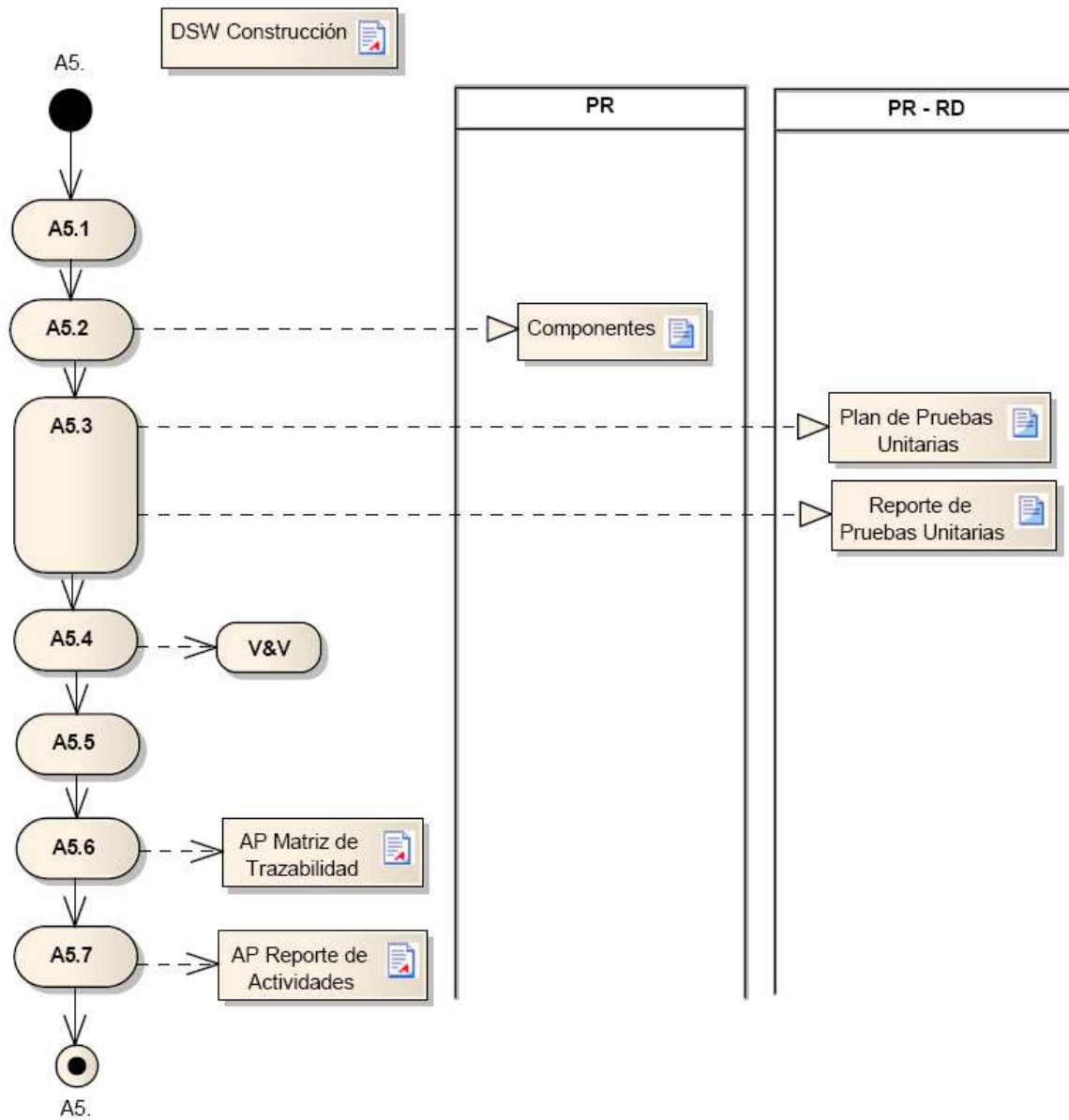
E.2 Diagrama de Unidades Atómicas correspondiente a la plantilla de Especificación de Requerimientos



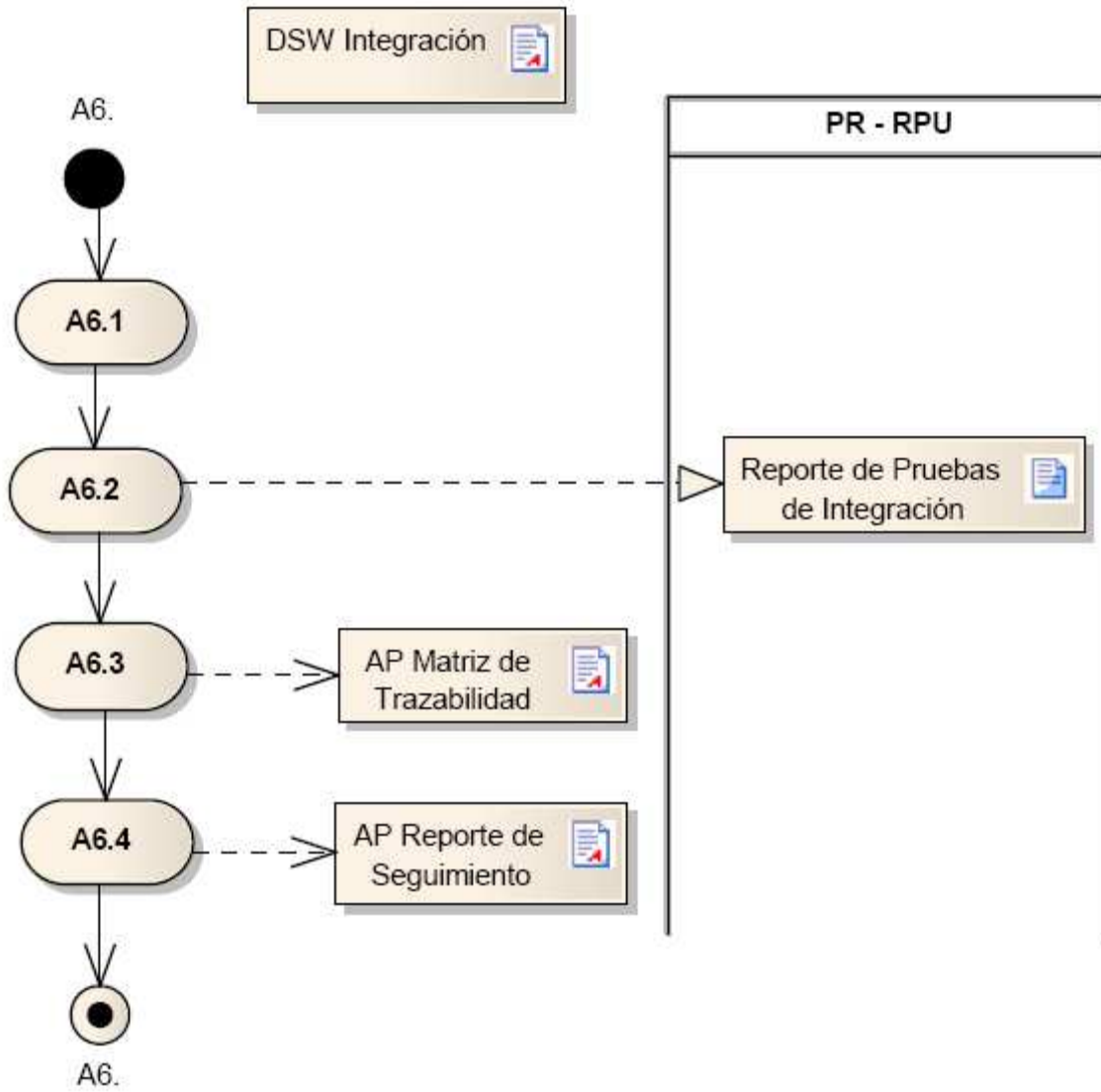
E.3 Diagrama de Unidades Atómicas correspondiente a la plantilla de Análisis



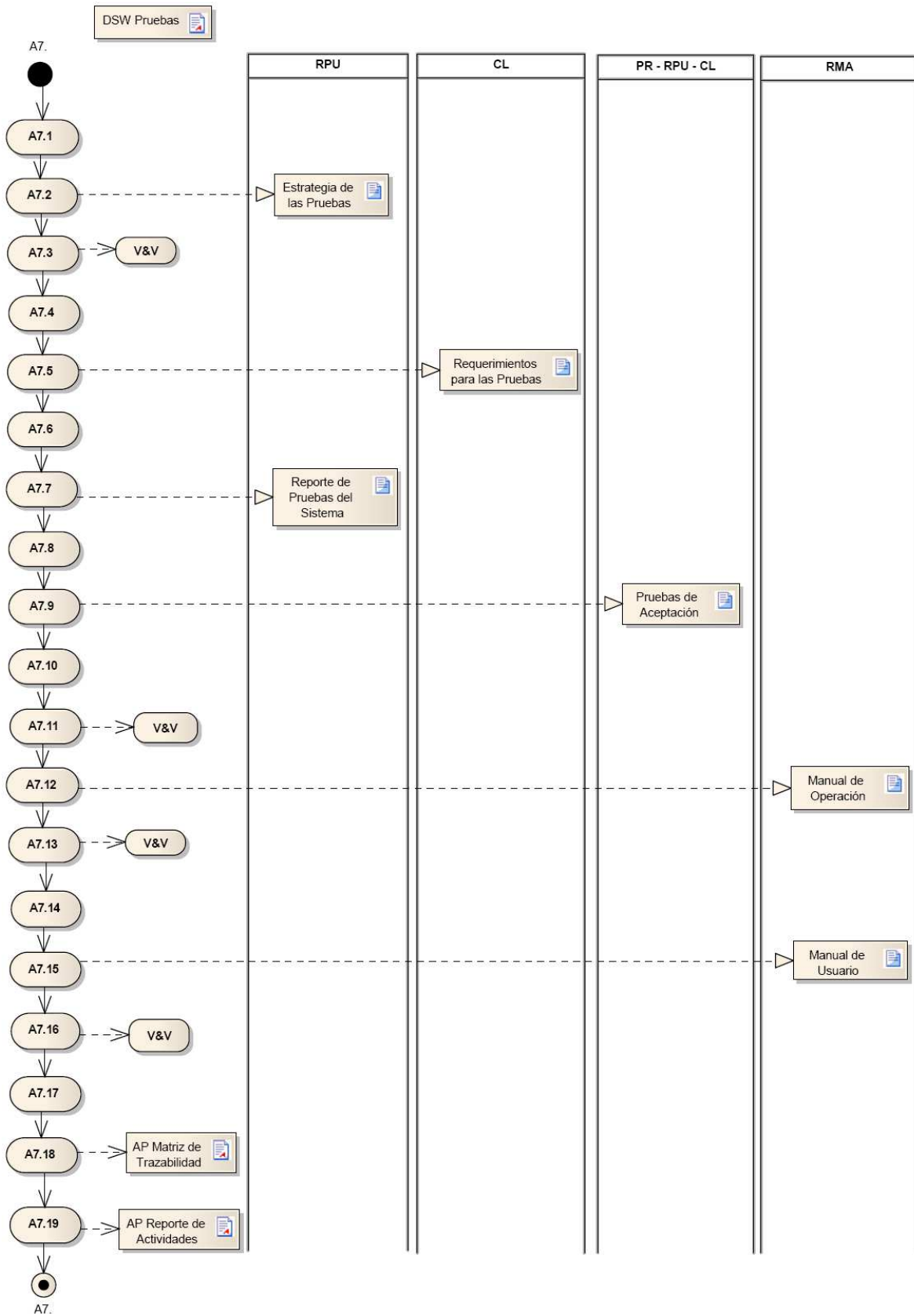
E.4 Diagrama de Unidades Atómicas correspondiente a la plantilla de Diseño



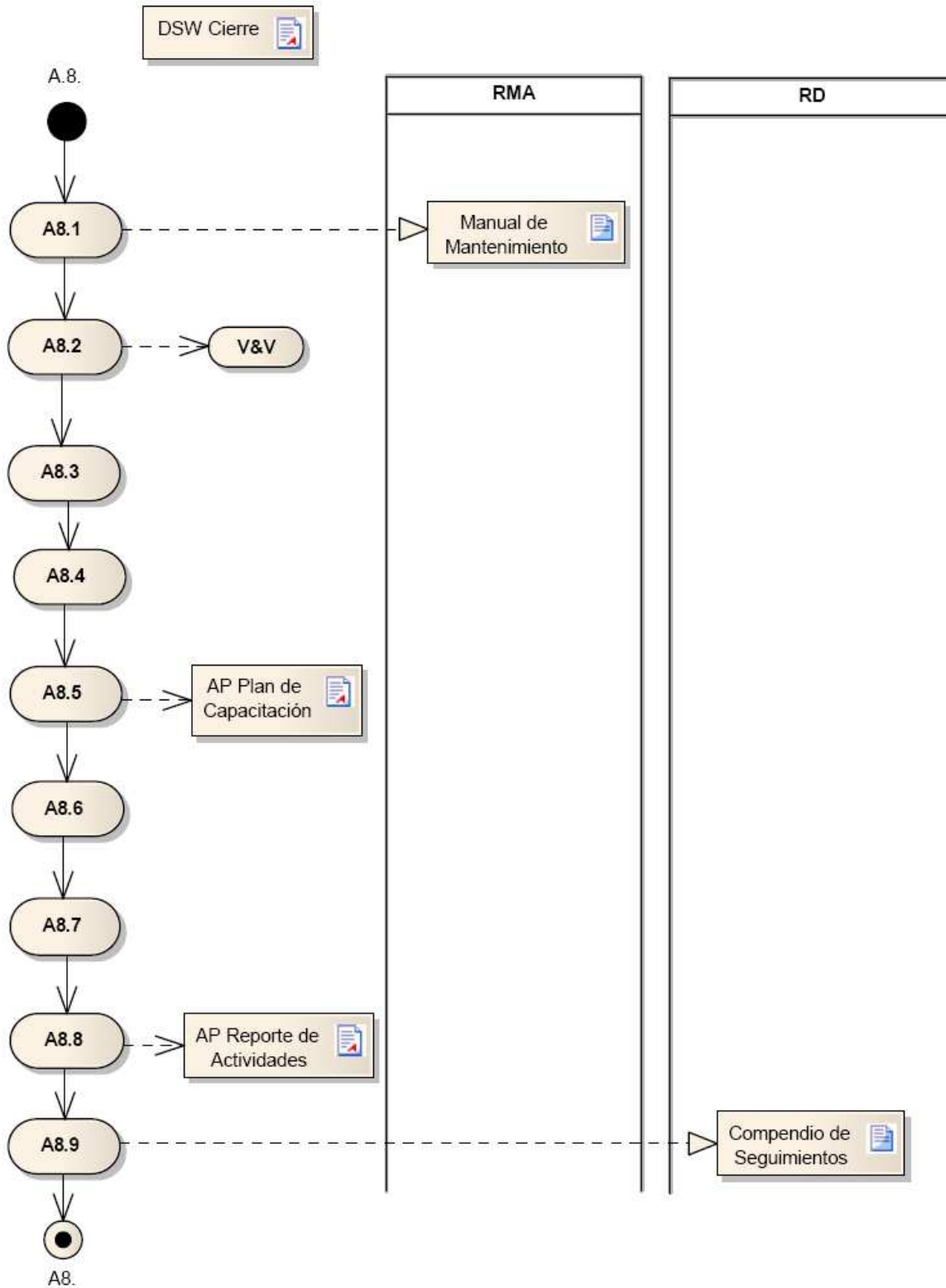
E.5 Diagrama de Unidades Atómicas correspondiente a la plantilla de Construcción



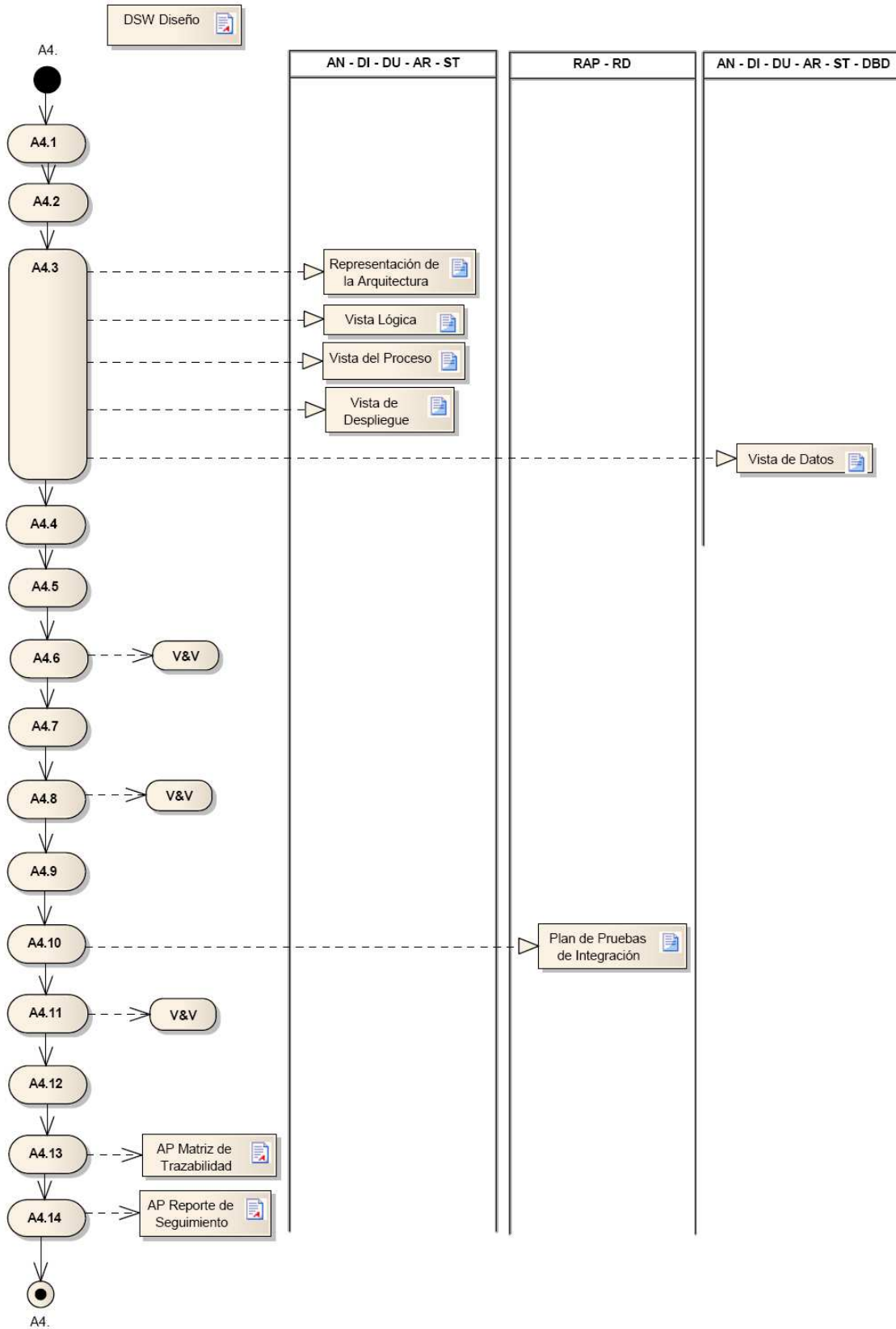
E.6 Diagrama de Unidades Atómicas correspondiente a la plantilla de Integración



E.7 Diagrama de Unidades Atómicas correspondiente a la plantilla de Pruebas

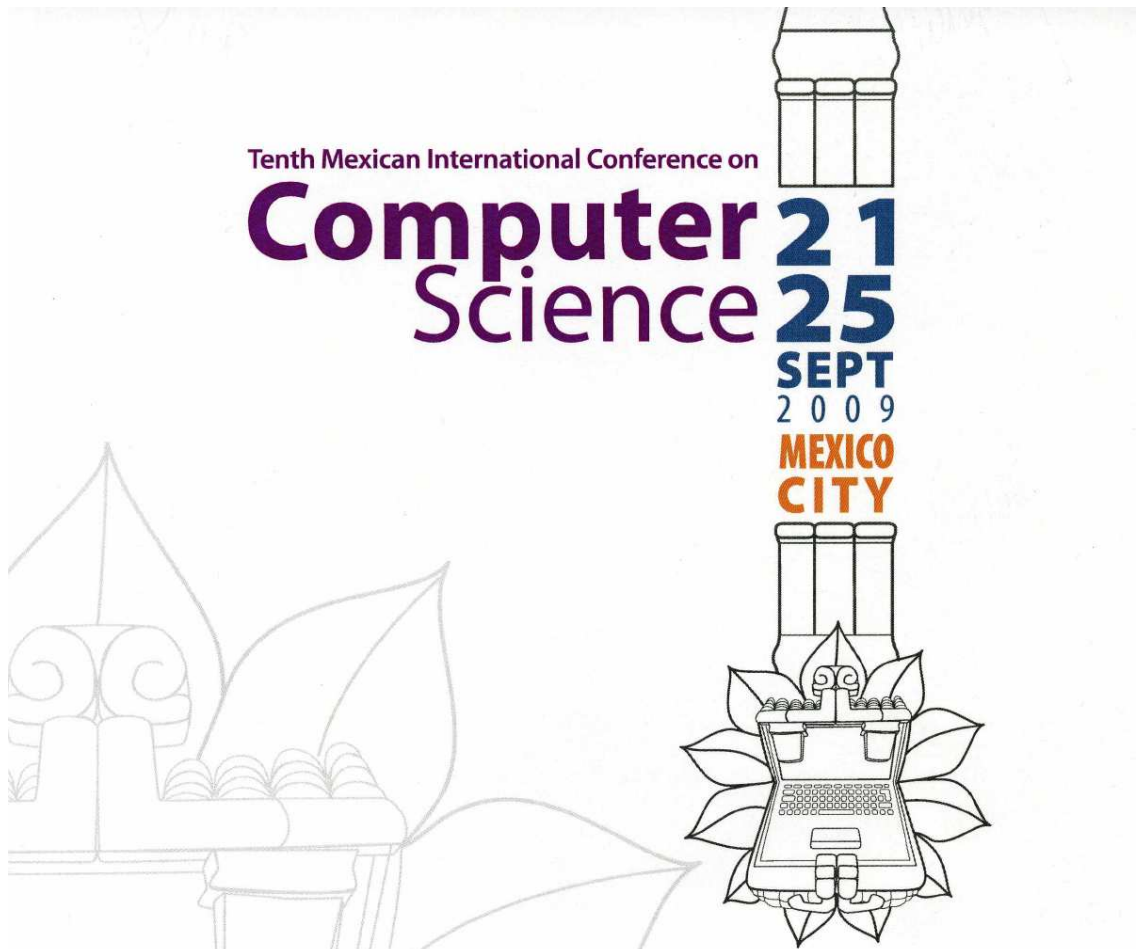


E.8 Diagrama de Unidades Atómicas correspondiente a la plantilla de Cierre




E.9 Diagrama de Unidades Atómicas correspondiente a la plantilla Diseño modificada

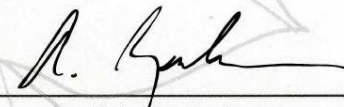
Apéndice F Constancia de participación como autor en el ENC'09 (Congreso Internacional Mexicano de Ciencias de la Computación 2009)

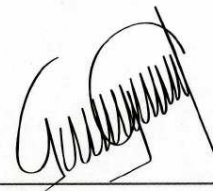


Tenth Mexican International Conference on
Computer Science 21
25
SEPT
2009
MEXICO
CITY

MIGUEL MORALES
participated in ENC 2009 as
AUTHOR


Dr. Rafael Lozano Espinosa
Organization Committee President


Dr. Alejandro Buchmann
Honorary President


Dra. Genoveva Vargas-Solar
SMCC President

F.1 Constancia de participación como autor en el ENC'09

Referencias

- [Artifact Overview RUP]
Rational Unified Process: Artifact Overview;
http://www.ts.mah.se/RUP/RationalUnifiedProcess/process/artifact/ovu_arts.htm, 01/05/2009
- [Boehm]
Boehm, B., Huang, L., Jain, A., Madachy, R. "The ROI of Software Dependability: The iDAVE Model". IEEE Software p. 54-61 (2004).
- [Cánepa]
Cánepa K., Dávila, A., "Mapeo de los Procesos de RUP respecto a MoProSoft", JIISIC 08, pp. 139-146
- [Charbonneau]
Charbonneau, S. "A mapping between RUP and the PMBok";
<http://www.ibm.com/developerworks/rational/library/4721.html#author>, 01/05/2009
- [Cruz 2009a]
Cruz, R., Morales, M., Morgado, M., Oktaba, H., Ibarguengoitia, G., Pino, F., Piattini, M. "Supporting the Software Process Improvement in Very Small Entities through E-learning: the HEPALE! Project". ENC 09, por publicarse.
- [Cruz 2009b]
Cruz, R., Oktaba, H., Pino, F., Piattini, M. "Repositorio de paquetes de despliegue de procesos bajo una arquitectura orientada a servicios SOA". SIS 09, por publicarse.
- [Enterprise Ireland]
Enterprise Ireland, Software Industry Statics for 1991-2005, 2006; www.nsd.ie/htm/ssii/stat.htm, 15/01/2009.
- [García]
García, C., Garzás, J., Piattini, M. "La mejora de procesos en pequeñas empresas y la ISO/IEC 29110"; Kybele Consulting, www.kybeleconsulting.com, 01/11/09
- [García-García]
García-García, J., Rodríguez-Hernández, J., Oktaba, H., Piattini, M., Ordoñez, C. "Prácticas y guías de Bases de Datos para MoProSoft". SIS 08, 2008.
- [Garzás]
Garzás, J., Cabrero, D. El valor y el retorno de la inversión de las TSI. Ra-Ma, 2007.
- [Hurtado]
Hurtado, J., Pardo, C., Fernández, L., Vidal, J., "Una Estrategia Ágil de Evaluación del Proceso Software: Agile SPI - Process Assessment Method". JIISIC 2007. pp. 349-358.
- [ISO 2004]
ISO/IEC 15504, Information technology - Process assesment. International Organization for Standarization, 2004
- [ISO 2008]
ISO/IEC 12207:2008, Systems and software engineering - Software life cycle processes. International Organization for Standarization, 2008
- [ISO/IEC 29110]
ISO/IEC 29110 Draft version, BER-005 29110 TR VSEP Overview WD2 15CL
- [ISO/IEC CD TR 29110-5-1]
ISO/IEC CD TR 29110-5-1 Software Engineering - Lifecycle Profiles for Very Small Entities (VSE) - Part 5-1: Management and Engineering Guide - Basic Profile;
http://www.iso.org/iso/iso_catalogue/catalogue_tc/cataloguedetail.htm?csnumber=51160, 15/01/2009.
- [Jacobson]
Jacobson I., Booch G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, Pearson, 2000
- [Kong]
Kong, B., Luo, X., Jiang, Y., Mao, M. "Template-based software process improvement", Journal of Information and Computational Science 1 (2), pp. 175-180
- [Kruchten]
Kruchten P., The Rational Unified Process. An Introduction. 2nd edition. Addison Wesley. 2000."



Universidad Nacional
Autónoma de México



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 esta 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.

- [Laporte 2008a]
Laporte, C. Y., Alexandre, S., O'Connor, R., "A software engineering lifecycle standard for very small enterprises", in R. V. O'Connor et al. (Eds.): EuroSPI 2008, CCIS 16, pp. 129-141
- [Laporte 2008b]
Laporte, C., Alexandre, S., Renault, A. "Developing International Standards for Very Small Entities", IEEE Computer 41(3): 98-101 (2008).
- [Ljungberg]
Ljungberg, A., "Process Measurement". 1994 - International Journal of Physical Distribution & Logistics Management Vol.32 No4, pp 254-287
- [Morales]
Morales, M., Ibarguengoitia, G., Pino, F., Piattini, M. "Apoyando la Mejora de Procesos de Software a través de Plantillas". SIS 09, por publicarse.
- [Morgado]
Morgado, M. Oktaba, H., Pino, F., Piattini, M. "Herramienta educacional para apoyar la adopción de procesos". SIS 09, por publicarse.
- [MTC]
Ministry of Science and Technology (MTC), Qualidade no Setor de Software Brasileiro-Pesquisa 2001, Quality in the Brazilian Software Sector-Survey 2001, 2001; <http://ctjovem.mct.gov.br/index.php/content/view/34854.htm> (Texto en portugués) 15/01/2009.
- [NYCE]
Modelo de Procesos para la Industria del Software - MoProSoft - versión 1.3. NMX-059/03-NYCE-2005. Ciudad de México, Organismo nacional de normalización y evaluación de la conformidad - NYCE.
- [Oktaba 1998]
Oktaba H., Ibarguengoitia G. "Software Processes Modeled with Objects: Static View", Computación y Sistemas, Iberoamerican Journal of Computer Science, CIC-IPN, México, 1, 4 (1998), pp.228-238.
- [Oktaba 2005]
Oktaba, H., MoProSoft: A Software Process Model for Small Enterprises. Proceedings of International Research Workshop for Process Improvement in Small Settings, 19-20 octubre, Pittsburgh, EUA, Special Report CMU/SEI-2006-SR-001, pp. 21-28
- [Oktaba 2007]
Oktaba, H.; García, F., Piattini, M., Pino, F., Alquicira, C., Ruíz, F. "Software Process Improvement: The COMPETISOFT Project". IEEE Computer, October, 2007. Vol. 40(10), pp. 21-28.
- [Oktaba 2008]
Oktaba, H., Piattini, M., Pino, F., Orozco, M. Julia., Alquicira, C., COMPETISOFT: Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos, Ra-Ma, 2008
- [Orozco]
Orozco, M. J., "Mejora de Procesos de Software en Iberoamérica", Software Guru, Febrero/Marzo 2009
- [Pino 2006]
Pino, F., Garcia, F., Piattini, M. "Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas. Revista Española de Innovación". Calidad e Ingeniería del Software (REICIS). Vol. 2(1) Abril 2006 pp. 6-23.
- [Pino 2008]
Pino, F., F., Triñanes, G., García, F. y Piattini, M. "Agil_MANTEMA: Una metodología de mantenimiento de software para pequeñas organizaciones". JISBD 08, pp.171-182
- [Pino 2009]
Pino, F., Hurtado, J., Vidal, J., García, F., Piattini, M. "A process for driving Software Process Improvement in small organizations". ICSP 2009. 342-353.
- [PMBok]
A Guide to the Project Management Body of Knowledge, Third Edition, Project Management Institute, EUA, 2004

[Polo]

Polo, M., M. Piattini, F. Ruiz, and C. Calero. "MANTEMA: A Software Maintenance Methodology Based on the ISO/IEC 12207 Standard". Proceedings of the 4th IEEE International Symposium and Forum on Software Engineering Standards 1999. pp. 76-81.

[Richardson]

Richardson Ita, Gresse von Wangenheim Christiane, "Why Are Small Software Organizations Different?", IEEE Software, January/February 2007. Vol. (40). pp.18-20.

[SCRUM]

SCRUM; <http://www.scrumalliance.org/>, 01/11/09

[SEI]

SEI, CMMI for Systems Engineering/Software Engineering, version 1.1. Pittsburgh, Software Engineering Institute (SEI). <Http://www.sei.cmu.edu/cmmi/> 01/11/09

[UML]

The Unified Modeling Language for Object-Oriented Development, Documentation set, ver 1.1, Rational Software Corp., September 1997

[van Loon]

van Loon, H. "Process Assessment and ISO 15504", Springer, 2007

[Word Template Index]

Word Template Index; <http://homepages.fh-giessen.de/~hg14540/SQM/projekt/rup/wordtmpl/index.htm>