



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**“REPOSITORIO DE PAQUETES DE DESPLIEGUE DE
PROCESOS BAJO UNA ARQUITECTURA
ORIENTADA A SERVICIOS”**

T E S I S

QUE PARA OBTENER EL GRADO DE :

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

CARLOS RICARDO CRUZ MENDOZA

DIRECTOR DE TESIS:

DRA. HANNA JADWIGA OKTABA

México, D.F.

2009



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.

Agradecimientos

Esta tesis de maestría, si bien ha requerido de esfuerzo y mucha dedicación por parte del autor y su director de tesis, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que a continuación citaré y muchas de las cuales han sido un soporte muy fuerte en momentos en los cuales necesitaba apoyo y guía.

Primero y antes que nada, dar gracias a mis Padres, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente; a ellos agradezco de todo corazón su amor, cariño y comprensión. A ellos los llevo siempre conmigo.

Agradecer hoy y siempre a mi familia ya que sin su apoyo y esfuerzo mis estudios no hubiesen sido posibles. En especial a ellos les debo el ánimo, apoyo, alegría y la fortaleza necesaria para seguir adelante.

Agradezco a la Dra. Hanna Oktaba por haber confiado en mi persona, por la paciencia y por la dirección de este trabajo. A la Mtra. Guadalupe Ibarguengoitia por los consejos, el apoyo y el ánimo que me brindó. Al Mtro. Gustavo Márquez por sus comentarios y enseñanzas, a la Mtra. Claudia Alquicira por la atenta lectura de este trabajo y, por último pero no menos importante, al Dr. Fernando Gamboa por sus comentarios en todo el proceso de elaboración de la Tesis.

A mi novia, por ser la persona que ha compartido el mayor tiempo a mi lado, porque en su compañía las cosas malas se convierten en buenas, la tristeza se transforma en alegría y la soledad no existe.

En general quisiera agradecer a todas y cada una de las personas que han vivido conmigo la realización de esta tesis de maestría, con sus altos y bajos y que no necesito nombrar porque tanto ellas como yo sabemos que desde lo más profundo de mi corazón les agradezco el haberme brindado todo el apoyo, colaboración, ánimo y sobre todo cariño y amistad.

Tabla de contenido

1 Introducción

- 1.1 Antecedentes y Motivación
- 1.2 Planteamiento del problema
- 1.3 Objetivos
- 1.4 Propuesta de Solución
- 1.5 Recorrido por el resto de la tesis

2 Marco de Referencia

- 2.1 Glosario
- 2.2 Antecedentes de los estándares y modelos de desarrollo de procesos de software en VSEs .
 - 2.2.1 Modelo de Procesos COMPETISOFT
 - 2.2.2 Norma ISO/IEC 29110
 - 2.2.3 Paquetes de despliegue
- 2.3 Paquetes de despliegue para el Análisis de Requerimientos basado en la norma ISO/IEC 29110 y el Modelo de Mejora COMPETISOFT.
- 2.4 Repositorios
- 2.5 Arquitectura Orientada a Servicios .
 - 2.5.1 Servicios Web
- 2.6 Aprendizaje en Línea y Sistemas Administradores de Aprendizaje

3 Desarrollo del sistema de repositorio de paquetes de despliegue para procesos.

- 3.1 Estandarización de los Paquetes de Despliegue
- 3.2 Desarrollo del repositorio HEPALE! - Repo Admin
- 3.3 Especificación de Requerimientos

3.4 Análisis y Diseño del repositorio

3.4.1 Arquitectura del Repositorio

3.5 Construcción del Repositorio

3.6 Pruebas

3.7 Uso del Repositorio

4 Conclusiones y trabajo a futuro .

Referencias Bibliográficas.

Apéndice A. Especificación de Requerimientos

Apéndice B. Análisis y Diseño

Apéndice C. Construcción.

Apéndice D. Pruebas .

Apéndice E. Deployment Package Software Requirements Analysis Basic Profile

Apéndice F. Paquete de Despliegue para la Mejora de Procesos COMPETISOFT

REPOSITORIO DE PAQUETES DE DESPLIEGUE DE PROCESOS BAJO UNA ARQUITECTURA
ORIENTADA A SERVICIOS

Carlos Ricardo Cruz Mendoza

Tutor de Tesis: Dra. Hanna Jadwiga Oktaba

RESUMEN

UN PAQUETE DE DESPLIEGUE ES UN PRODUCTO DESARROLLADO DENTRO DE LA NORMA ISO/IEC PDTR 29110-5-1 SOFTWARE ENGINEERING - LIFECYCLE PROFILES FOR VERY SMALL ENTITIES (VSEs) CON EL FIN DE FACILITAR LA IMPLEMENTACIÓN DE UN PROCESO O PRÁCTICA EN ALGÚN MODELO DE MEJORA DE PROCESOS PARA EMPRESAS PEQUEÑAS Y MEDIANAS. EL SIGUIENTE TRABAJO DE TESIS PRESENTA LA CREACIÓN DE UN REPOSITORIO REALIZANDO CAMBIOS NECESARIOS EN LA ESTRUCTURA DE LOS PAQUETES DE DESPLIEGUE TOMANDO EN CUENTA DISTINTOS ESTÁNDARES Y HERRAMIENTAS USADAS EN LA CREACIÓN, REPRESENTACIÓN, INTERACCIÓN Y MANIPULACIÓN DE OBJETOS DE APRENDIZAJE Y E-LEARNING. EL REPOSITORIO ES CONSTRUIDO BAJO UNA ARQUITECTURA ORIENTADA A SERVICIOS (SOA) IMPLEMENTANDO SERVICIOS WEB PARA LA MANIPULACIÓN, BÚSQUEDA Y ALMACENAMIENTO DE LOS PAQUETES DE DESPLIEGUE UTILIZANDO ESTÁNDARES PARA EL MANEJO DE SEGURIDAD, BÚSQUEDA Y ALMACENAMIENTO. EL REPOSITORIO FACILITARÁ EL INTERCAMBIO DE CONTENIDOS ENTRE DIFERENTES SISTEMAS DE ADMINISTRACIÓN DE APRENDIZAJE ADEMÁS DE FACILITAR LA PUBLICACIÓN DE ESTOS.

1 Introducción

1.1 Antecedentes y Motivación.

Las organizaciones pequeñas y medianas o grupos de trabajo con menos de 50 empleados, son fundamentales para el crecimiento de muchas economías. En los Estados Unidos, Brasil, Canadá, China, India, Finlandia, Irlanda, Hungría y muchos otros países representan hasta el 85% de todas las empresas desarrolladoras de software [1] [2] [3].

La gran mayoría de estas empresas guían sus procesos¹ de acuerdo a las habilidades técnicas y capacidades tecnológicas de los individuos que la componen, Sin embargo para crecer y mejorar, estas empresas deben adoptar procesos y modelos de Ingeniería de Software.

Organizaciones internacionales como la Organización Internacional de Estandarización (ISO) y el Instituto de Ingeniería de Software (SEI) han desarrollado estándares y modelos dirigidos a empresas o grupos de trabajo mayores. Como consecuencia de esto, las organizaciones pequeñas, medianas y grupos de trabajo se enfrentaban a grandes obstáculos a la hora de intentar adoptar estos estándares y modelos. Dichos modelos no están orientados a satisfacer las necesidades específicas de estos grupos ya que resultan excesivos en términos de esfuerzo y costo. Para fortalecer, crecer y mejorar las organizaciones pequeñas, medianas o grupos de trabajo necesitan prácticas eficientes de Ingeniería de Software adaptadas a su tamaño y tipo de negocio [43].

En un esfuerzo para ayudar a las organizaciones pequeñas, medianas o grupos de trabajo a adoptar procesos, se crearon modelos nacionales (*MoProSoft* [15] [38]) e iberoamericanos (*COMPETISOFT* [4][5]). A nivel internacional destaca el proyecto de la Organización Internacional para la Estandarización a través del Grupo de Trabajo WG24 del Sub-Comité SC7 (Ingeniería de software) del comité técnico de Tecnologías de la Información JTC1, de crea la norma internacional *ISO/IEC 29110*

¹ Un proceso puede ser definido como un conjunto de actividades que se necesitan para transformar las necesidades de un cliente en sistema de software.

Software Engineering - Lifecycle Profiles for VSEs [6]. Este grupo de trabajo decide atacar el problema de la adopción y entendimiento de procesos en organizaciones, empresas y grupos de trabajo de menos de 25 personas (Very Small Entities – VSEs por sus siglas en inglés), a través del desarrollo de perfiles (subconjuntos de estándares existentes y guías para la implementación de estos perfiles).

Para las guías de ISO/IEC 29110 se crea el concepto de *paquete de despliegue* [7][8], el cual explica con mayor detalle las tareas necesarias para llevar a cabo correctamente distintas prácticas de los procesos de Ingeniería de Software. Estos paquetes constan de una serie de puntos, criterios y tareas necesarias para cumplir con un proceso en algún modelo, es decir, funcionan para describir elementos de procesos tanto de COMPETISOFT, MoProSoft, ISO/IEC 29110-5-1 o cualquier otro modelo de procesos de software, ayudando a las VSE en la adopción de procesos. Los elementos típicos de un paquete de despliegue son:

- Descripción de los procesos (actividades, tareas, roles y productos).
- Plantillas y Ejemplos de Plantillas.
- Listas de Verificación.
- Herramientas.
- Mapeos a otros estándares y modelos.

Estos paquetes de despliegue son el punto central y materia de trabajo a lo largo de esta tesis. Ejemplos de estos paquetes de despliegue aplicados a los procesos de análisis de requerimientos, diseño, construcción y prueba de software, pueden encontrarse en [10].

1.2 Planteamiento del problema

Debido a que el concepto de paquete de despliegue es reciente, existen distintos problemas asociados a esto, tal es el caso de:

- Dispersión del material relacionado con los paquetes de despliegue.
- Falta de difusión, distribución y divulgación de los paquetes de despliegue.

- Necesidad de herramientas que faciliten la administración, almacenamiento y divulgación de los paquetes de despliegue.

1.3 Objetivos

Con el objetivo de solucionar los problemas planteados anteriormente se definieron los siguientes objetivos:

- Encontrar la representación de paquetes de despliegue que facilite su difusión, distribución y aprendizaje a través de herramientas en línea.
- Crear un repositorio capaz de almacenar, administrar y distribuir los paquetes de despliegue.

1.4 Propuesta de Solución

Con el fin de cumplir estos objetivos se propone como primer paso la recopilación de los paquetes de despliegue creados por la Escuela Superior de Tecnología de Quebec, tomando como caso de estudio dos de estos paquetes. El primer paquete de despliegue que se tomará como caso de estudio está basado en las actividades de análisis y administración de requerimiento de la futura norma ISO/IEC 29110-5-1 y, como segundo caso de estudio, se utilizará el paquete creado por la Universidad de Castilla-La Mancha el cual está basado en el modelo de mejora de procesos de COMPETISOFT.

A los paquetes de despliegue del caso de estudio, se les realizarán adecuaciones estructurales, utilizando estándares internacionales de aprendizaje en línea [11] (e-learning, por sus siglas en inglés), con la finalidad de facilitar su manipulación y almacenamiento en un repositorio “hecho a la medida”.

Una vez realizadas las modificaciones estructurales a los paquetes de despliegue, se propone crear un repositorio que se llamará: HEPALÉ! - Repo Admin (**H**erramienta **P**ara **A**doptar **L**os **E**stándares – Administrador del Repositorio) el cual facilitará la administración, almacenamiento y distribución de los paquetes de despliegue. En el desarrollo de éste repositorio se utilizará como metodología de trabajo y su documentación la tesis titulada “Mejora de Procesos de Software basada en plantillas” [38]. Esta metodología tiene como objetivo generar una guía para la implementación de procesos a través de la definición de plantillas que agilizan los procesos de administración del proyecto, análisis de requerimientos, diseño, construcción, implementación y pruebas.

La figura 1.0 resume los diferentes pasos en la propuesta de solución.



Fig. 1.0 Pasos de la propuesta de solución.

- El paso número 1 consiste en la recopilación de distintos paquetes de despliegue creados; en particular se tomarán para el desarrollo del resto de la tesis dos de estos paquetes:
 - Deployment Package – Software Requirements Analysis (en colaboración con Claude Laporte, profesor de la Escuela Superior de Tecnología, Quebec, Canadá y editor del WG24)
 - Paquete de Despliegue para el Modelo de Mejora de Procesos COMPETISOFT (en colaboración con la Universidad de Castilla-La Mancha, Ciudad Real, España).
- El paso número 2 consiste en realizar las adecuaciones estructurales en los paquetes de despliegue utilizando herramientas que faciliten su difusión, divulgación y distribución.
- El paso número 3 consiste en la creación de un repositorio y un sistema administrador llamado HEPALE! - Repo Admin, capaz de utilizar los recursos ofrecidos por el repositorio. El desarrollo del repositorio se documentará utilizando las plantillas de la tesis de Miguel Trujillo "Mejora de Procesos de Software basada en plantillas" [38].

1.5 Recorrido por el resto de la tesis

Este trabajo de tesis consta de 2 capítulos, 6 apéndices, conclusiones, trabajo a futuro y las referencias bibliográficas. En el primer capítulo se presenta un marco de referencia que sienta las bases y el contexto de los diferentes elementos utilizados en esta tesis. Dentro del marco de referencia se explican temas básicos de ingeniería de software, antecedentes y situación actual de los estándares y modelos de desarrollo de procesos de software en VSEs. Se ahondará en los modelos de procesos *COMPETISOFT* e *ISO/IEC 29110-5-1 Software Engineering – Lifecycle Profiles for VSEs*; se ampliarán los conceptos de paquete de despliegue, arquitectura orientada a servicios (SOA), servicios web y aprendizaje en línea. Todos estos conceptos son necesarios para comprender el trabajo realizado.

En el capítulo 2 se explican los cambios estructurales necesarios en los paquetes de despliegue, para que puedan ser almacenados y distribuidos a través del repositorio tomando en cuenta, estándares, herramientas y temas relacionados con el aprendizaje en línea. Así mismo, se explica la metodología de plantillas seguida en el desarrollo del repositorio.

En los apéndices A, B, C, D se detalla la utilización de estas plantillas con los datos necesarios para especificar el desarrollo de HEPALE! Repo - Admin.

En los apéndices E y F se muestran los datos de los dos paquetes de despliegue tomados como caso de estudio. Finalmente en la última parte de esta tesis se mencionarán las conclusiones obtenidas, así como el trabajo a futuro.

2 Marco de Referencia

En este capítulo, se presentan a detalle las bases, conceptos y contexto de los distintos elementos asociados en este trabajo de tesis. Como punto de partida se presenta un glosario con definiciones básicas; posteriormente se explican los antecedentes de los estándares y modelos de desarrollo de procesos de software en VSEs. Una vez explicada la situación actual de estos modelos y estándares, se realiza una selección de los modelos utilizados; los cuales son: COMPETISOFT debido a su carácter internacional y la futura norma ISO/IEC 29110 debido a su carácter internacional y enfocado a VSEs. De esta norma se tomará especialmente el concepto de paquete de despliegue explicando su definición y utilidad a lo largo del trabajo. A manera de ejemplo se detallará el contenido de dos paquetes de despliegue tomados como caso de estudio, ejemplificando su utilidad y funcionamiento. Estos paquetes de despliegue serán la materia prima de almacenamiento del repositorio. Finalmente, se detallara los conceptos de: Arquitectura Orientada a Servicios, Servicios Web y Estándares de Aprendizaje en Línea; todos estos, conceptos necesarios para entender la implementación del repositorio y los cambios realizados en la estructura de los paquetes de despliegue.

2.1 Glosario

A continuación se presentan algunas definiciones básicas para la comprensión de este trabajo de tesis.

- **Proceso.**- Conjunto de actividades relacionadas entre sí, llevadas a cabo a través de roles y por elementos automatizados, que utilizando recursos/productos/insumos de entrada producen un producto/recurso/sistema de software como salida.
- **Actividad.**- Conjunto de diferentes tareas de un proceso.
- **Tarea.**- Es parte de una serie de acciones (por ejemplo, pasos) que ejecutan/resuelven un trabajo o un problema.
- **Sub tarea.**- Cuando una tarea es compleja puede dividirse en otras.
- **Rol.**- Es responsable por un conjunto de actividades de uno o más procesos. Un rol puede ser asumido por una o más personas de tiempo parcial o completo.

- **Producto.**- Pieza de información o entregable que puede ser producido por una o más tareas (por ejemplo: un documento, código)
- **Modelos de Desarrollo de Procesos de Software.**- Paradigmas o filosofías de desarrollo en los cuales se puede apoyar para la realización/implementación de software.

2.2 Antecedentes de los estándares y modelos de desarrollo de procesos de software en VSEs

Países como la India, Irlanda y Singapur han sido exitosos en desarrollar su industria de software como motor de crecimiento económico. En México también han comenzado a desarrollar la industria del software como base para poder crecer económicamente. Factores como ubicación geográfica (México tiene en el mismo uso horario que el mayor mercado de software del mundo, EUA), tratados comerciales y afinidad con la cultura de negocios occidental, han facilitado la creación de VSEs (90% de las industrias desarrolladoras de software en México son pequeñas y medianas empresas [39]), pero el problema surge cuando estas empresas deben de consolidarse y sustentarse en el mercado. No hace mucho tiempo atrás la producción de software era realizada de una manera “artesanal” por no decir; desordenada y caótica, lo cual provocaba la constante presencia de proyectos fallidos. Teniendo en cuenta esto las organizaciones y empresas se dieron cuenta de la necesidad de mejorar la calidad de los productos a través de estándares y modelos de Ingeniería de Software.

Diversos esfuerzos se han realizado para apoyar a las VSE en su crecimiento, desarrollo y consolidación. Actualmente existe una fuerte competencia entre las distintas VSEs para obtener un proyecto o trabajo y en ocasiones el factor de éxito entre el obtener o no un proyecto se basa en la capacidad de la VSE de poder desarrollar un proyecto de manera satisfactoria.

En México se han realizado esfuerzos considerables para que las pequeñas y medianas empresas puedan consolidarse en el mercado laboral. Uno de los esfuerzos más importantes recae en el programa PROSOFT [14]. Este programa financiado por la Secretaría de Economía y con apoyo de la Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de la Información (CANIETI) y la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS), plantea el fomento a la industria y el mercado de las Tecnologías de la Información (TI) a través de apoyos financieros para que empresas puedan adoptar modelos, procesos y estándares de desarrollo de software, ya que la calidad de un producto es directamente proporcional a la del proceso con el que es generado[15].

Actualmente existen distintas opciones relacionadas con estándares y modelos a utilizar. Ejemplos de estos modelos son: ISO 9001-2000[40], ISO/IEC 15504[41], CMMI[42], MoProSoft, COMPETISOFT e ISO/IEC 29110-5-10. Estos modelos y estándares tienen el objetivo apoyar la implementación de una forma de trabajo dentro de las empresas que les permita entender, controlar, comunicar, mejorar, predecir y certificar el trabajo realizado, todo esto a través de procesos, actividades, evaluaciones; tareas que conducen a la creación de productos (software, documentos, materiales).

La pregunta que surge es ¿Cuál de estos modelos se tiene que implementar en una VSE? Ciertamente no existe una sola respuesta a esta pregunta porque, para poder contestarla se requiere saber exactamente qué necesita una organización y qué no necesita.

	¿Es específico para el desarrollo y mantenimiento de software?	¿Ofrece reconocimiento a nivel internacional?	¿Está dirigido a VSEs?	¿Es fácil de entender?	¿Es fácil de aplicar en una VSE?
CMMI	X	X	-	-	-
ISO 15504	X	X	-	-	-
ISO 9000:2000	X	X	-	-	-
MoProSoft	X	-	X	X	X
COMPETISOFT	X	X	X	X	X
ISO/IEC 29110	X	X	X	X	X

Tabla 1.0. Marco Comparativo entre distintos modelos de referencia del proceso de software.

La tabla 1.0 muestra una comparativa entre los distintos modelos/estándares y las características deseadas para el objeto de estudio de esta tesis.

Una vez obtenidas las características deseadas por los distintos modelos se optó por tomar como objeto de estudio los modelos COMPETISOFT e ISO/IEC 29110-5-1 ya que además de cumplir con todas las características deseadas ambos modelos están enfocados a pequeñas y

medianas empresas y no son simplemente marcos de referencia de certificación. En la siguiente parte se explicará a detalle el modelo de procesos COMPETISOFT.

2.2.1 Modelo de Procesos COMPETISOFT

El modelo de Procesos COMPETISOFT fue un proyecto financiado por el Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED) como un esfuerzo para incrementar el nivel de competitividad de las PyMEs (Pequeñas y Medianas Empresas) iberoamericanas productoras de tecnologías de la información, a través de la creación de un marco metodológico que pueda ser utilizado como un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica. El proyecto fue dirigido por Mario Piattini (España) y Hanna Oktaba, (México), e involucró a más de 100 investigadores de países como España, México, Colombia, Brasil, Argentina, Uruguay, Costa Rica, Perú, entre otros [16].

Los objetivos que buscó el proyecto COMPETISOFT son:

- Desarrollo de un Marco Metodológico común que pueda ser ajustado a las PyMEs Iberoamericanas.
- Difundir la cultura de la mejora de procesos en Iberoamérica.
- Establecer un mecanismo común y mutuamente reconocido de evaluación y certificación de la industria del software en Iberoamérica.

El Marco metodológico de COMPETISOFT, está constituido por un Modelo de Procesos de Referencia, un Modelo de Mejora y un Método de Evaluación. Este marco toma como base a modelos y mejores prácticas ya existentes, los cuales han sido mejorados y adaptados con las experiencias de los investigadores, empresas pequeñas, medianas y unidades gubernamentales, usando como marco general para la evaluación a la norma internacional ISO/IEC 15504: Performing an Assessment[17].

La figura 2.0 Muestra una visión general del proyecto COMPETISOFT.

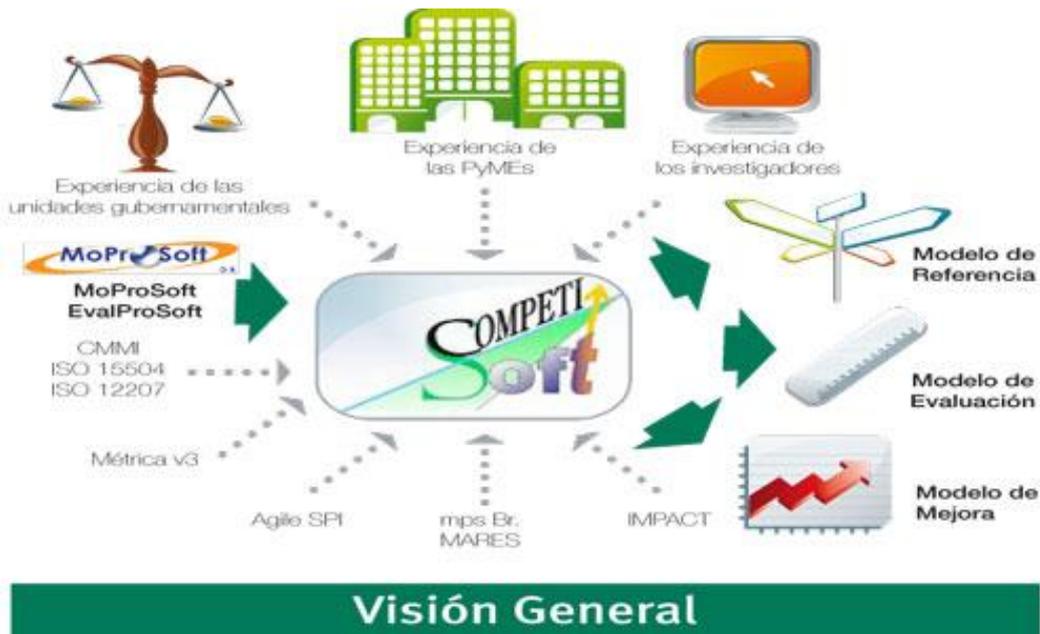


Figura 2.0 Visión General del proyecto COMPETISOFT

El modelo de referencia de COMPETISOFT, al estar basado en MoProSoft establece tres categorías que agrupan procesos de acuerdo a la estructura típica de una organización: Alta Dirección, Gerencia y Operación. Cada una de estas categorías tiene un objetivo dentro del modelo de referencia, en particular la categoría de Alta Dirección es la encargada de la Gestión de Negocio, la Categoría de Gerencia es la encargada de la Gestión de Procesos, Gestión de Cartera de Proyectos y la Gestión de Recursos (humanos, bienes, servicios, infraestructura y conocimiento), mientras que la categoría de Operación es la encargada de la Administración del Proyecto, Desarrollo y Mantenimiento de Software, teniendo en cuenta que las distintas Categorías se comunican y retroalimentan entre ellas. La Figura 2.1 muestra el modelo de procesos o referencia mencionado.



Modelo de Procesos

Figura 2.1 Modelo de Procesos o Referencia de COMPETISOFT

El modelo de mejora de procesos guía y gestiona la implementación de proyectos en el contexto de las VSEs.

Para guiar a las pequeñas organizaciones software en la conducción y gestión de los proyectos de mejora, el modelo de mejora de COMPETISOFT define un conjunto de componentes (ver figura 2.2), los cuales se describen a continuación:

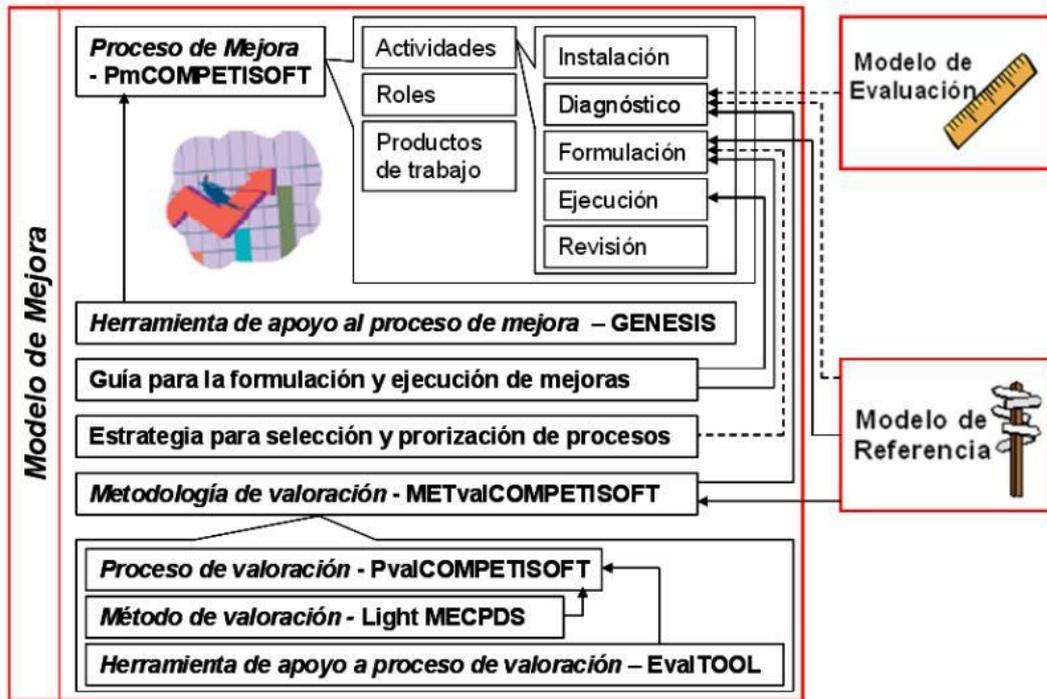


Figura 2.2. Componentes del modelo de mejora de COMPETISOFT.

- Un proceso para guiar la mejora continua de procesos denominado PmCOMPETISOFT, y una herramienta software, denominada GENESIS, para apoyar al responsable de conducir la mejora en la implementación de las actividades descritas por este proceso.
- Una metodología para la valoración de procesos (evaluación interna de procesos) denominada METvalCOMPETISOFT, la cual está compuesta por: (i) un proceso para guiar la valoración de procesos, (ii) un método para la valoración de procesos, (iii) un conjunto de medidas para estimar el rendimiento y la capacidad de los procesos, y (iv) un conjunto de herramientas de soporte a la valoración de procesos software.

- Una guía para formular y ejecutar mejoras utilizando Scrum, con el fin de aprovechar la sinergia de la valoración de procesos y encausarla hacia la iniciación de la realización efectiva de mejoras al interior de la pequeña organización software.
- Una estrategia para la selección y priorización de procesos [18].

Una vez explicado los antecedentes y componentes del proyecto COMPETISOFT; en la siguiente sección se explicarán a detalle los elementos de la futura norma ISO/IEC 29110.

2.2.2 Norma ISO/IEC 29110

Debido a la necesidad de normas orientadas a VSEs se han identificado varias iniciativas nacionales e internacionales para la creación de normas que mejoren la calidad de los procesos, de carácter más reducido y orientado expresamente a la VSEs. Entre las iniciativas más conocidas se pueden destacar el ESSI (European Software and System Initiative) en la Unión Europea, los modelos MoProSoft y EvalProSoft en México, el modelo ITMARK del ESI (Instituto Europeo del Software) y el proyecto COMPETISOFT para Iberoamérica. No obstante estas iniciativas si bien suponen un importante paso en la creación de modelos para VSEs adolecen de su carácter regional. En esta línea y para apoyar a las VSEs, ISO (en especial el Joint Technical Committee 1, JTC 1, encargado de la estandarización a nivel internacional en el campo de Tecnología de la Información) ha creado dentro del sub-comité SC7 el grupo de trabajo WG24 [19] formado por investigadores de alto nivel y expertos en el área. El propósito de este grupo es:

- La creación de estándares para la mejora de procesos de software, estableciendo un marco común para describir perfiles (conjunto de procesos para ciertos contextos de negocio), guías (implementaciones de los perfiles) y métodos de evaluación, enfocados a las necesidades de las VSEs [20].

Si bien el trabajo del grupo continúa, ya existen un conjunto de borradores técnicos que son la base del futuro estándar ISO para la mejora de procesos en VSEs: el ISO/IEC 29110 para VSEs. El cual, orgullosamente está basado en la norma mexicana NXM-I-059-NYCE (MoProSoft) y COMPETISOFT.

2.2.2.1 Estructura Estándar ISO/IEC 29110

Los documentos que componen la norma ISO/IEC 29110 son los siguientes:

- **Visión General**

Este informe incluye los conceptos clave para comprender y utilizar los demás documentos de la norma.

- **Perfiles**

Los perfiles se definen con el propósito de empaquetar referencias a/y/o partes de otros documentos de manera formal, con el fin de adaptarlos a las necesidades y características de las VSE.

- **Marco de Trabajo y Taxonomía.**

Especifica los elementos comunes a todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles de la norma.

- **Especificación – de Perfil para las VSE.**

Define la composición de un perfil, sus relaciones con el subconjunto normativo de estándares usados, y las referencias a documentos de entrada.

Un ejemplo de una especificación de perfil es el documento 29110-4-1 (Especificación - Perfil Básico). Su objetivo es identificar los elementos de administración de proyectos y desarrollo de software, adecuados a las necesidades de las VSEs, como un subconjunto de procesos y sus resultados (outcomes) de ISO/IEC 12207.

- **Guías.**

Las guías contienen instrucciones que indican cómo se deben implementar los procesos de un perfil y como deben ser evaluados.

- **Guías de evaluación**

Se describe el proceso necesario a seguir para realizar una evaluación que determine las capacidades de proceso y madurez de la organización.

- **Guía de Ingeniería y Gestión – para las VSE.**

Las guías de ingeniería y gestión proporcionan orientación sobre la implementación y uso de un perfil. Para cada perfil, existe una guía de este tipo.

Un ejemplo de guía de ingeniería y gestión es el documento 29110-5-1. Esta guía ayuda en implementar los procesos definidos en el Perfil Básico de 29110-4-1; se compone de dos procesos: Project Management (Administración de Proyecto) y Software Implementation (Implementación, Desarrollo y Mantenimiento de Software).

La figura 3.0 Muestra la estructura de los documentos ISO/IEC 29110.

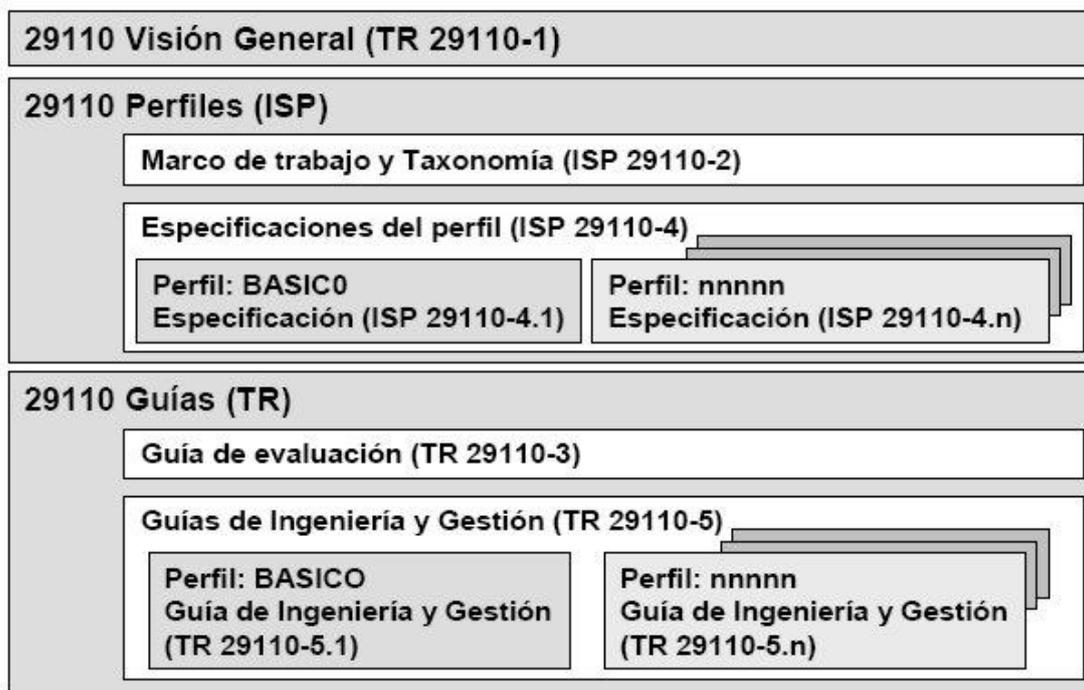


Figura 3.0. Documentos ISO/IEC 29110

2.2.3 Paquetes de despliegue

Uno de los aportes que se tienen dentro de la norma ISO/IEC 29110-5-1 es la definición del concepto **paquete de despliegue (deployment package)**.

Un paquete de despliegue es una serie de productos desarrollados para facilitar la aplicación de un conjunto de prácticas/procesos en una VSE. Además, los paquetes de despliegue son creados con el propósito de ayudar a que una VSE pueda implementar una práctica/ proceso de algún modelo o norma, sin la necesidad de tener que implementar toda la norma o modelo al mismo tiempo. Tampoco importa que el proceso a implementar sea de ISO/IEC 29110, ISO/IEC 29110-5-1, COMPETISOFT, MoProSoft ó cualquier otro modelo o norma.

Los elementos típicos de un paquete de despliegue se encuentran listados en la tabla 2.0.

1.- Nombre del Paquete de Despliegue	
a. Descripción Técnica	
i.	Propósito del documento
ii.	Relación con el modelo
b. Definiciones clave	

c. Introducción
d. Descripción detallada de los Procesos, Actividades, Roles, Productos
i. Descripción Detallada de los Procesos, Actividades y Tareas
ii. Roles
iii. Descripción de los Productos
e. Plantillas
f. Ejemplos de Plantillas
g. Listas de Verificación
h. Herramientas
i. Referencias a otros estándares
i. Referencias bibliográficas
j. Formas de Evaluación

Tabla 2.0 Elementos Típicos de un paquete de despliegue.

A continuación se explican a detalle cada uno de estos puntos:

1. Nombre/Título del paquete. Debe ser descriptivo
 - a. Descripción Técnica.
 - i. Propósito del documento. Se explica el concepto de paquete de despliegue, la finalidad del paquete de despliegue, la descripción de los elementos típicos de un paquete de despliegue (descripción del proceso, actividades, entradas, salidas, roles, la referencia a las normas y modelos, guías, plantillas, material y la lista de herramientas).
 - ii. Relaciones con el modelo. Se mapea el paquete de despliegue con un proceso de algún modelo o estándar de implementación de procesos de software.

- b. Definiciones clave. En este punto se explican los conceptos necesarios para comprender el proceso que se intenta describir en el paquete de despliegue.
- c. Introducción. Se presenta un pequeño resumen del tema a tratar en el paquete de despliegue.
- d. Descripción detallada de los Procesos, Actividades, Roles, Productos. En este punto se realiza una especificación de los procesos, actividades, roles y productos que deben ser cubiertos por la norma o modelo.
 - i. Descripción Detallada de los Procesos, Actividades y Tareas. Se explica la estructura de los procesos, actividades y tareas con su respectivo objetivo, explicación del objetivo, descripción y productos de entrada o productos de salida, además de notas y diagramas que ayuden a clarificar el tema.
 - ii. Roles. Se detalla la lista de roles, abreviaciones y descripción de las competencias requeridas.
 - iii. Descripción de los Productos. Se detalla lista de los productos necesarios en el proceso, descripciones y propiedades adicionales.
- e. Plantillas. Producto que proporciona una guía o esqueleto de los elementos necesarios para cumplir con los procesos, actividades, tareas, roles del paquete de despliegue.
- f. Ejemplos de Plantillas. Plantillas aplicadas a un proyecto o proceso.
- g. Listas de Verificación. Se usa para determinar los puntos que se deben cumplir del proceso.
- h. Herramientas. Se detalla el software (de pago o libre) que facilite la(s) tarea(s)/actividad(es) descritas en el paquete de despliegue.

- i. Referencia a otros estándares. Se proveen ejemplos en donde se relacionan/mapean las tareas, actividades y procesos que se realizan en el paquete de despliegue con otros modelos o estándares ISO/IEC.
- i. Referencias Bibliográficas, utilizadas en el paquete de despliegue.
- j. Formas de Evaluación. Cuestionarios que permiten mejorar el contenido del paquete de despliegue.

En la siguiente sección se explicará los antecedentes, utilidad y uso de dos paquetes de despliegue que se tomarán como caso de estudio.

2.3 Paquetes de despliegue para el Análisis de Requerimientos basado en la norma ISO/IEC 29110 y el Modelo de Mejora COMPETISOFT.

Bajo la coordinación de la Escuela Superior de Tecnología (Quebec, Canadá), se crearon una serie de paquetes de despliegue cuyo objetivo es ayudar a la implementación de los procesos para el Perfil Básico de la norma ISO/IEC 29110-5-1. Estos paquetes de despliegue describen los procesos de:

- Análisis de requerimientos.
- Arquitectura y diseño detallado.
- Implementación y pruebas unitarias.
- Pruebas del Software.
- Entrega del Producto.
- Control de Versiones.
- Administración del Proyecto.
- Verificación y Validación.
- Auto-Evaluación.

- Selección y Realización de proyectos piloto.

Cabe destacar que el paquete de despliegue para el manejo de análisis de requerimientos y un paquete de despliegue para el modelo de mejora de COMPETISOFT fueron creados en colaboración conjunta entre la Escuela Superior de Tecnología (Claude Y. Laporte), la Universidad de Castilla-La Mancha, España (Mario Piattini, Fco. J. Pino) y la Universidad Nacional Autónoma de México (Hanna Oktaba, Anna Vázquez, Carlos Ricardo Cruz, Miguel E. Morales), sentando las bases de los primeros esfuerzos realizados en la creación de paquetes de despliegue.

El primero de los paquetes de despliegue fue utilizado como prototipo, ayudando a definir y redefinir puntos de la especificación de los futuros paquetes de despliegue. Este paquete fue ligado a la parte de análisis de requerimientos del Perfil Básico de la futura norma ISO/IEC 29110 Parte 5-1: Guía de Ingeniería y Gestión para el Perfil Básico, guiando a las VSEs a través de las actividades, tareas y productos necesarios para cumplir con la fase de análisis de requerimientos de la norma.

El segundo paquete creado está ligado al proceso de mejora de COMPETISOFT. Este paquete tiene como finalidad proveer a las VSEs de una guía capaz de ajustarse a las necesidades de la empresa con el fin de establecer los elementos necesarios para dirigir y administrar la mejora de procesos en una pequeña organización desarrolladora de software, y lograr institucionalizar la cultura de la mejora continua al interior de la organización. En este paquete de despliegue se especificaron las actividades PvalCOMPETISOFT y PfemCOMPETISOFT descritas anteriormente, además de un detallado de las tareas, productos, roles, plantillas, ejemplos de plantillas y herramientas necesarias en el proceso de mejora de COMPETISOFT. Ambos paquetes de despliegue pueden consultarse en los apéndices E y F.

Estos dos paquetes de despliegue serán la materia prima de almacenamiento del repositorio.

En las siguientes secciones se introducen los conceptos de: Repositorio, Arquitectura Orientada a Servicios, Servicios Web y Estándares de Aprendizaje en Línea. Todos estos conceptos son necesarios para entender la implementación del repositorio y los cambios realizados en la estructura de los paquetes de despliegue.

2.4 Repositorios

Un repositorio es un sistema de software que almacena objetos digitales tales como: tesis, libros, imágenes, artículos, reportes técnicos, objetos de aprendizaje¹ y sus metadatos, además

¹ Un objeto de aprendizaje [12] es una estructura o archivo que contiene un objetivo general, objetivos específicos, una actividad de aprendizaje, metadatos y mecanismos de evaluación, digitalizados en archivos HTML y javascript.

de proporcionar algún tipo de interfaz de búsqueda de los mismos, o bien para la interacción con otros sistemas [13] [24]. Debido a la variedad de los objetos digitales, no es posible hacer una distinción clara entre los tipos de repositorios existentes, por lo que se mencionarán los repositorios que abarcan la mayoría de los objetos digitales:

- Repositorios Institucionales. Los cuales pueden definirse como: “conjunto de servicios que ofrece la universidad a los miembros de su comunidad para la dirección y distribución de materiales digitales creados por la institución y los miembros de la comunidad” [25]. El ejemplo más común de este tipo de repositorio es Dspace@MIT [26], el cual es un repositorio institucional construido para guardar, compartir y buscar objetos digitales de una universidad y/o organización.
- Repositorios de Objetos de Aprendizaje. Un repositorio de objetos de aprendizaje es un conjunto de servicios ofrecidos por una organización con el objetivo de administrar, difundir y facilitar el acceso a los objetos de aprendizaje para potenciar su reutilización. Un ejemplo de un repositorio de este tipo es CAREO (Campus Alberta Repository of Educational Objects) [27], el cual es un repositorio que provee de las herramientas necesarias para la búsqueda de objetos de aprendizaje en otros repositorios.

Estos tipos de repositorios en ocasiones no hacen una separación clara entre el manejo de los objetos de aprendizaje y otros objetos digitales, además de que no aprovechan las ventajas que los objetos de aprendizaje pueden ofrecer, es por este motivo que se decidió implementar un repositorio personalizado de objetos de aprendizaje, que explote las características únicas de estos objetos, utilizando una arquitectura y modalidad que facilite su enseñanza, distribución, compartición y búsqueda.

2.5 Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios (SOA por sus siglas en inglés) es definida de acuerdo con el W3C (World Wide Web Consortium) como un conjunto de componentes que pueden ser invocados, cuyas descripciones de interfaces se pueden publicar y descubrir. [44] Más allá de esta definición, la arquitectura orientada a servicios es un paradigma para organizar y utilizar

servicios distribuidos que pueden estar controlados bajo diferentes propietarios e implementados bajo diferentes tecnologías.

La arquitectura orientada a servicios define la base para que un conjunto de servicios independientes puedan colaborar entre sí dando lugar a procesos de negocio más complejos [21]. La figura 4.0 Muestra un ejemplo de interacción entre servicios.

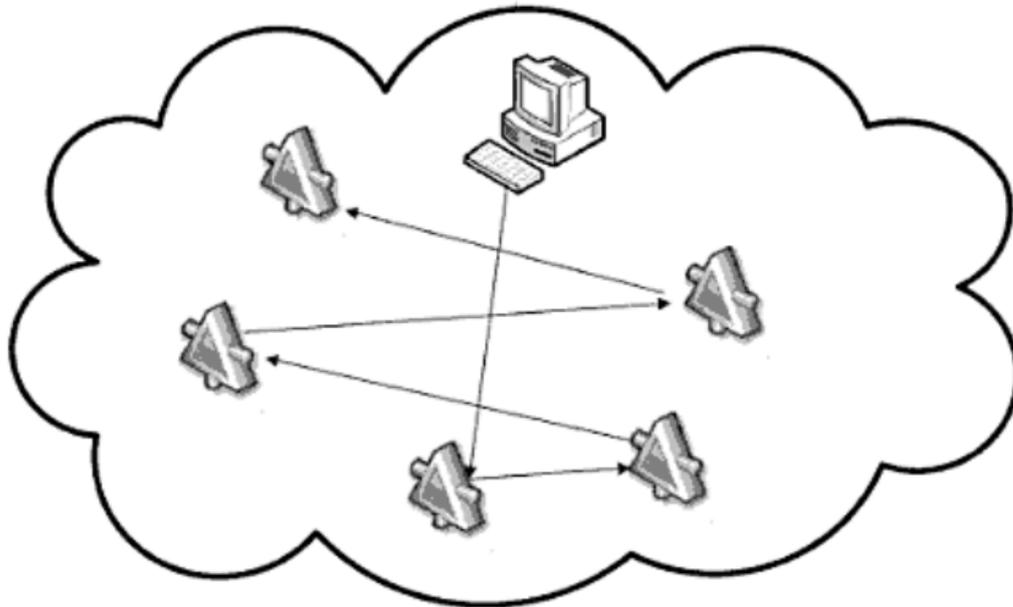


Figura 4.0. Ejemplo de Interacción entre servicios.

Es importante mencionar que SOA como su nombre lo indica es una arquitectura de software, por lo que no está ligado a ninguna tecnología, lo que permite que distintas tecnologías puedan comunicarse entre ellas facilitando la interoperabilidad de servicios.

SOA establece la comunicación entre sistemas de manera distinta, ve las aplicaciones como servicios autónomos donde se definen fronteras explícitas y se asume la heterogeneidad para que colaboraren plataformas de programación diferentes. Para que esta interoperabilidad se logre; los servicios deben usar un mismo lenguaje (eXtensible Markup Language, XML) y los mismos esquemas o protocolos (Simple Object Access Protocol, SOAP).

El objetivo de una arquitectura SOA es proveer de transparencia al momento de utilizar un servicio, es decir, facilita la posibilidad de utilizar un determinado servicio que se encuentre en cualquier lugar, sin la necesidad de tener que modificar ningún servicio.

En el nivel más alto de la arquitectura orientada a servicio podemos encontrar tres componentes:

- El servicio: cada servicio tiene una funcionalidad a la que pueden acceder el resto de servicios y clientes; pueden participar cualquier número de servicios.

- El archivo descriptor: el archivo descriptor tiene información sobre los servicios y la funcionalidad de estos. Y también tiene la información de cómo se puede acceder a cada servicio.
- El cliente: usa el archivo descriptor para localizar servicios y poder usar su funcionalidad. Un cliente puede ser otro servicio que quiere acceder o utilizar la funcionalidad que aportan otros servicios.

Y también podemos encontrar tres relaciones entre los componentes:

- Localización de servicios: clientes potenciales de los servicios localizan los servicios por medio del archivo descriptor. El archivo descriptor aporta a los clientes la información sobre cómo encontrar un servicio.
- Publicación de servicios: un componente publica un servicio, haciéndole disponible a los clientes a través del archivo descriptor.
- La comunicación entre los servicios y el cliente: el cliente hace peticiones al servicio a través un protocolo en común. El servicio recoge la petición del cliente y da una respuesta con el mismo protocolo. La figura 4.1 muestra las relaciones entre los componentes de SOA.

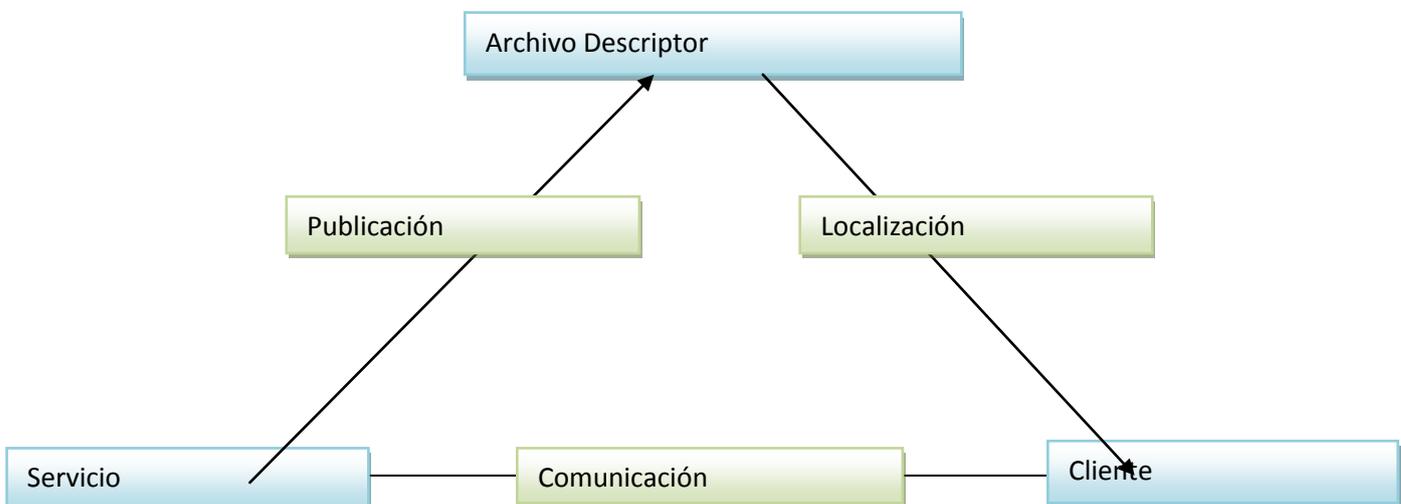


Figura 4.1 Componentes de SOA

Normalmente el archivo descriptor recibe el nombre de UDDI (Universal Description, Discovery and Integration)[22], la comunicación entre los componentes se realiza mediante SOAP y para describir un servicio se utiliza el lenguaje WSDL (Web Service Description Language).

El concepto de SOA no es nuevo (fue descrito por primera vez en [23]). Lo que realmente es nuevo es la implementación de Servicios Web.

2.5.1 Servicios Web

Un Servicio Web es un servicio que se comunica con los clientes a través de un conjunto estándar de protocolos y tecnologías. Estos estándares están implementados en los productos y paquetes de desarrollo de algunos de los principales proveedores de software del mercado (SUN, Microsoft); lo que hace a los servicios web la principal opción para construcción de arquitecturas SOA.

Para diseñar la arquitectura del repositorio de paquetes de despliegue se eligió la arquitectura orientada a servicios web por las siguientes razones:

- **Interoperabilidad:** la arquitectura SOA es débilmente acoplada y los clientes y servicios se comuniquen independientemente de la plataforma en que residan. Los protocolos de comunicación con Servicios Web son independientes de la plataforma, lenguaje de codificación y sistema operativo por lo que facilitan la comunicación entre los involucrados. En particular en este trabajo SOA permitirá el intercambio de contenido entre el repositorio y las distintas aplicaciones que hacen uso de ellos (Sistemas Web y Sistemas Administradores de Aprendizaje), sin importar el lenguaje de programación en el que se encuentren desarrollados.
- **Reutilización:** Los servicios del repositorio pueden ser expuestos como servicios web y en base a estos definir nuevos servicios que cubran nuevas funcionalidades.
- **Reducción de complejidad:** Ya que al estar basado en estándares se facilita la implementación del repositorio.

2.6 Aprendizaje en Línea y Sistemas Administradores de

Aprendizaje

Como se ha mencionado anteriormente, COMPETISOFT y la próxima norma ISO/IEC 29110 son modelos relativamente nuevos, por lo que es necesario difundir el conocimiento acerca de los mismos en las VSEs. Para poder abarcar un gran número de VSEs o interesados, en este trabajo se propone al aprendizaje en línea como una herramienta que ayude con este objetivo.

El aprendizaje en línea (e-learning) es un conjunto de tecnologías de la información y elementos pedagógicos que facilitan la enseñanza y formación a distancia [28].

El aprendizaje en línea ocupa el concepto de objeto de aprendizaje como unidad o medio necesario para que los capacitados (llámense alumnos, empleados, estudiantes), aprendan de manera guiada y autodidáctica. En particular las VSE encuentran útil la implantación de aprendizaje en línea debido a que:

- Facilita la enseñanza a distancia y acceso a los contenidos. Ya que al no tener un lugar necesario para que un maestro capacite a los empleados, el empleado puede capacitarse desde cualquier lugar; como puede ser su propio hogar, trabajo o desde donde tenga acceso al contenido (por ejemplo a través de dispositivos móviles).
- Flexibilidad de horarios. El empleado al no tener un horario fijo, puede capacitarse en el momento que lo requiera.
- Minimización de costos. Se evitan gastos de residencia, transporte, tanto para el capacitador como el empleado.
- Reutilización de contenidos. Ya que al no ser necesario un capacitador cualquier experto en el área puede crear contenidos que pueden ser utilizados dentro de las VSEs.

Pero al igual que ventajas el aprendizaje en línea presenta problemas que son necesarios conocer para evaluar el uso del aprendizaje en línea como una estrategia adecuada para la adopción en VSEs. Entre las desventajas podemos encontrar:

- Falta de Motivación. Un empleado con hábitos pobres puede encontrar el aprendizaje en línea soso y aburrido, ya que el aprendizaje en línea requiere de una mayor disciplina de estudio.
- Calidad de Contenidos. Dado que cualquier persona con el debido entrenamiento es capaz de generar objetos de aprendizaje existen muchos contenidos que pueden no tener la calidad requerida en el proceso de aprendizaje.
- Brecha Digital. Esto debido a que el aprendizaje en línea requiere de cierto equipo tecnológico, por lo que ciertos grupos sociales pueden no tener acceso a este tipo de tecnología.

Como se mencionó antes, el aprendizaje en línea es un conjunto de tecnologías de la información, una de estas tecnologías son los llamados sistemas administradores de aprendizaje (Learning Management Systems, LMS, por sus siglas en inglés).

Estos LMS integran servicios como foros, wikis, actividades, evaluaciones, chats con la finalidad de proveer una interacción dinámica durante el proceso de enseñanza, logrando instruir a un gran número de capacitados al mismo tiempo y de manera asíncrona; ejemplos de estos sistemas administradores de aprendizaje son Moodle [29] y Blackboard [30].

Existen muchos esfuerzos a nivel mundial por crear estándares en las tecnologías asociadas con e-learning; los principales tres promotores de estándares que han desarrollado especificaciones para objetos y diseños de aprendizaje son:

- ADL (Advanced Distributed Learning). Financiada por el Departamento de Defensa de los Estados Unidos, es la encargada de desarrollar especificaciones y estándares que permitan la reutilización e interoperabilidad de contenidos educativos. El estándar propuesto por ADL se llama modelo de Referencia de Objetos de Contenido Compartido (SCORM [31] por sus siglas en inglés), el cual combina varias especificaciones basados en e-learning e implementados a través de HTML y javascript con el objetivo de hacer dinámica la iteración entre los sistemas administradores de aprendizaje y el capacitado.
- IEEE (Institute of Electrical and Electronics Engineers). El comité de estándares de tecnologías del aprendizaje de la organización IEEE, también ha desarrollado

recomendaciones y estándares técnicos que recalcan las mejores prácticas en el uso de e-learning.

- IMS. El consorcio global para el aprendizaje IMS, reúne a vendedores y fabricantes de hardware y software, instituciones educativas, editoriales, agencias gubernamentales, proveedores de contenidos multimedia y otros consorcios de menor tamaño; es actualmente la iniciativa más activa en el desarrollo de especificaciones y estándares en este dominio. Los estándares más importantes de IMS son: IMS Meta Data, que plantea recursos para el procesamiento de datos de aprendizaje; IMS Content Packaging (Paquetes de Contenido)[32], que describe la manera de digitalizar y empaquetar contenidos, IMS QTI, que describe formas de intercambio de preguntas y evaluaciones.

Teniendo en mente estos estándares de e-learning; en este trabajo se proponen adecuaciones estructurales a los paquetes de despliegue a través del uso de SCORM e IMS Content Packaging. IMS Content Packaging facilitará la tarea de digitalizar y empaquetar los paquetes de despliegue permitiendo su intercambio, divulgación y distribución en el repositorio, mientras que SCORM añadirá contenido dinámico para la enseñanza y representación de los paquetes de despliegue. En el siguiente capítulo se explicará a detalle estos cambios en los paquetes de despliegue.

3 Desarrollo del sistema de repositorio de paquetes de despliegue para procesos.

En este capítulo, como primer punto se presenta a detalle los cambios estructurales en los paquetes de despliegue a través de una herramienta especializada en la implementación de estándares de e-learning. Posteriormente se explicará la metodología utilizada en el desarrollo de un repositorio hecho a la medida llamado: HEPALE! Repo Admin. Esta metodología está basada en la utilización de plantillas, permitiéndonos agilizar los procesos de análisis de requerimientos, diseño, arquitectura, construcción y pruebas. Una vez definida la metodología y los cambios necesarios en la estructura de los paquetes de despliegue, se explicará el desarrollo del sistema a través del uso de plantillas que especifican los requerimientos, análisis, diseño, construcción y pruebas de HEPALE! Repo Admin. Finalmente se detalla la integración y uso del repositorio con un sistema administrador de aprendizaje a través del proyecto HEPALE!

3.1 Estandarización de los Paquetes de Despliegue

Como se mencionó anteriormente, se necesitaba que los paquetes de despliegue y sus metadatos fueran almacenados en el repositorio. La solución a este problema fue la adecuación de los paquetes de despliegue a un estándar internacional que pudiera mejorar su divulgación, distribución y reutilización. La solución planteada a este problema consiste en la adecuación de los paquetes de despliegue a *paquetes de contenidos*. Un paquete de contenido es usado en e-learning como forma estándar para el almacenamiento de objetos de aprendizaje y sus metadatos, permitiendo exportar contenidos de un sistema administrador de aprendizaje a otro conservando al mismo tiempo la estructura y forma de visualizar los objetos de aprendizaje.

Estos paquetes de contenido constan de 2 elementos principales:

- Un archivo o varios archivos XML que describen el contenido estructural, metadatos y objetos de aprendizaje dentro del paquete. Generalmente este archivo es llamado Manifiesto.
- Objetos de Aprendizaje (OAs), los cuales pueden ser archivos de texto (explicaciones descontextualizadas, resúmenes, lecturas, definiciones, imágenes, HTML). Se pueden

agrupar varios OAs para construir otros y en algunos casos los OAs pueden ser lanzados como parte de una experiencia de aprendizaje.

El primer paso para la creación de estos paquetes de contenidos fue la creación de OAs basados en los puntos o temas definidos dentro de los paquetes de despliegue utilizando la herramienta eXe ¹ [16]. Esta herramienta facilita la conversión de contenido estructurado a Objetos de Aprendizaje. Debido a que los paquetes de despliegue son contenidos o temas estructurados, el utilizar eXe facilitó la tarea de convertir cada contenido, punto o tema del paquete de despliegue a un Objeto de Aprendizaje. Por ejemplo: el punto del paquete de despliegue “Nombre del Paquete de Despliegue” siempre será mapeado junto con su contenido al Objeto de Aprendizaje “*inicio.html*”, el punto del paquete de despliegue “Descripción Técnica” siempre será mapeado junto con su contenido al Objeto de Aprendizaje “*descripciontecnic.html*” y así sucesivamente con cada uno de los puntos definidos por la tabla de contenido del paquete de despliegue. Al realizar de esta manera el mapeo aseguramos que toda la información contenida en los paquetes de despliegue pueda ser mapeada en OAs. Además al realizar de esta manera el mapeo se asegura que el mismo proceso pueda ser repetido por cada uno de los paquetes de despliegue existentes.

Para la creación de los OAs necesarios en un paquete de contenido, se realizó un mapeo de los puntos establecidos por la tabla de contenido de los paquetes de despliegue y OAs. El resultado de este mapeo es presentado en la tabla 3.0.

Tabla de Contenido de los Paquetes de Despliegue	Objeto de Aprendizaje Creado
1.- Nombre del Paquete de Despliegue	inicio.html
a. Descripción Técnica	descripciontecnic.html
i. Propósito del documento	propositodeldocumento.html
ii. Relaciones entre los modelos	relacionentremodelos.html
b. Definiciones clave	definicionesclave.html
c. Introducción	introducción.html
d. Descripción detallada de los Procesos, Actividades, Roles, Productos	descripciondetalladadelos_p_a_t_r_p.html

¹ eXe (e.learning XHTML) es un entorno de software libre para la creación y edición de contenido multimedia basado en web diseñado para ayudar a académicos, en la creación de guías didácticas, objetos de aprendizaje y recursos de e-learning sin necesidad de tener conocimientos técnicos de HTML, XML o en tareas complicadas relacionadas con e-learning.

i.	Descripción Detallada de los Procesos, Actividades y Tareas	descripciondetalladadelos_p _a_t.html
ii.	Roles	roles.html
iii.	Descripción de los Productos	descripcionproductos.html
e.	Plantillas	plantillas.html
f.	Ejemplos de Plantillas	ejemplosdeplantillas.html
g.	Listas de Verificación	listasdeverificacion.html
h.	Herramientas	herramientas.html
i.	Referencias a otros estándares	referenciasaotrostandares.html
j.	Referencias bibliográficas	referenciasbibliograficas.html
k.	Formas de Evaluación	formasdeevaluacion.html

Tabla 3.0 Mapeo entre la tabla de contenido de los paquetes de despliegue y objetos de aprendizaje.

Una vez realizado este mapeo se utilizó la herramienta multimedia para crear los objetos de aprendizaje y metadatos. En la figura 7.0 se muestra una captura de pantalla de la herramienta eXe mostrando el contenido del OA *inicio.html*.

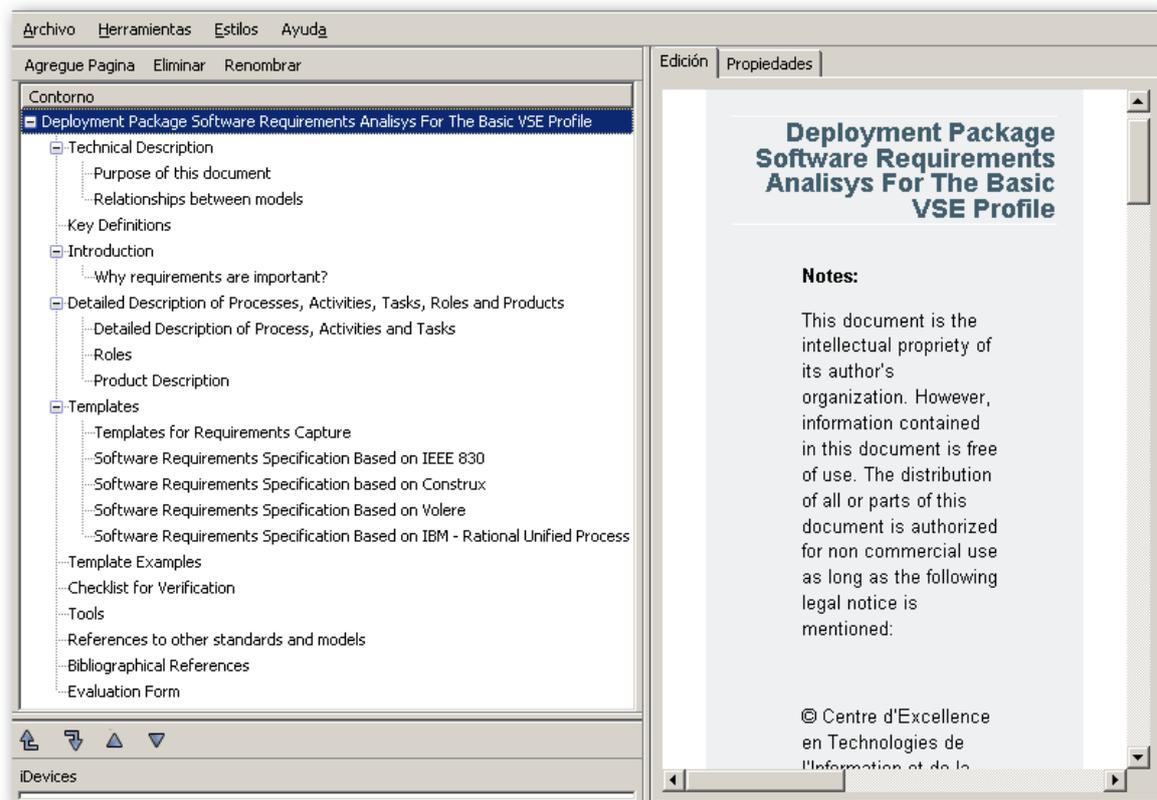


Figura 7.0. Captura de pantalla del OA *inicio.html* en eXe

Una vez digitalizados los OAs (páginas HTML e imágenes), la herramienta multimedia crea automáticamente un archivo *manifiesto* el cual describe el contenido estructural, organización y metadatos de los OAs. Este archivo contiene:

- Declaración de los metadatos. Se describen los metadatos asociados a los OAs.
- Declaración de la organización. Se crea una guía didáctica en forma jerárquica para mostrar el orden en el cual se presentarán los OAs.
- Declaración de los OAs (recursos).

Una vez creados todos estos componentes la herramienta empaqueta todos los OAs y archivos de configuración en un único archivo con las especificaciones necesarias para la creación del paquete de contenido. El cual es considerado como la unidad básica que se almacenará en el repositorio.

Las características principales por las cuales se optó por utilizar eXe como herramienta de contenido multimedia son:

- Intuitivo y fácil de usar.
- Creación automática de los recursos y OAs.

- No se requiere de conocimiento técnico especializado.
- Utiliza la funcionalidad what you see is what you got (wysiwyg, por sus siglas en inglés). Haciendo posible ver la apariencia del contenido en todo momento.
- Extensa documentación.
- Decremento del tiempo de creación de contenidos al automatiza el proceso de la creación de los recursos, archivos de configuración y empaquetado de los paquetes de contenidos.

3.2 Desarrollo del repositorio HEPALE! - Repo Admin

En el desarrollo de éste proyecto se contó con la participación de distintas personas; por lo que fue necesario contar con un modelo de desarrollo de software. El modelo elegido como forma de trabajo es titulado “Mejora de Procesos de Software basada en plantillas” [38], este trabajo tiene como objetivo generar una guía para la implementación de procesos en las VSEs a través de la definición de plantillas. En este trabajo se define una plantilla como un conjunto de unidades atómicas necesarias para cubrir un grupo de procesos descrito en el modelo. Estas unidades atómicas, se definen como elementos que contienen los aspectos necesarios para cubrir una entidad/actividad del proceso presente en el modelo. Estas plantillas son ordenadas mediante dos grupos: plantillas generales y plantillas del proceso. Dentro de las plantillas generales se tienen las siguientes:

- Administración del proyecto
- Desarrollo de software
- Mantenimiento del software.

Para definir las plantillas correspondientes a la administración del proyecto, se siguieron las prácticas establecidas por PMBOK [29]. Para definir las plantillas del desarrollo de software, se siguieron las prácticas establecidas por RUP [30]. Para desarrollar la plantilla de mantenimiento de software, se siguieron las prácticas establecidas por Agil Mantema [45].

Dentro de las plantillas generales para el desarrollo de software se tiene las siguientes plantillas del proceso:

- Especificación de requerimientos
- Diseño
- Análisis
- Construcción

- Pruebas

Una de las ventajas de la estructura estas plantillas, es que al estar compuestas por unidades atómicas, permiten la trazabilidad de los elementos y vuelven flexible a la plantilla, permitiendo agregar o quitar unidades atómicas a la plantilla según las necesidades del desarrollo.

Con lo anterior expuesto, para el desarrollo del repositorio y del sistema administrador se adaptaron y utilizaron las platillas que definen la toma de requerimientos, diseño, análisis, construcción y pruebas.

3.3 Especificación de Requerimientos

Una vez digitalizados los paquetes de despliegue en paquetes de contenido y definida la metodología de desarrollo se procedió a crear el sistema.

El propósito de este sistema es la creación de un repositorio que ofrezca distintos Paquetes de Contenidos (IMS Content Packaging). Estos Paquetes de Contenidos almacenan Objetos de Aprendizaje de los paquetes de despliegue de los modelos de procesos ISO/IEC 20110-5-1 y COMPETISOFT.

Los Objetos de Aprendizaje y sus metadatos serán almacenados utilizando la especificación de Paquetes de Contenidos y los datos representados mediante SCORM, con el fin de mantener un repositorio que sea compatible con distintos Sistemas de Administración de Aprendizaje (LMS por sus siglas en inglés).

Además este repositorio deberá contar con Servicios Web (Web Services) capaces de administrar estos paquetes, es decir, contar con las funcionalidades necesarias para poder dar de alta, borrar, consultar y cambiar los Paquetes de Contenidos.

Como requerimientos no funcionales. El repositorio permite realizar búsquedas sobre los metadatos de los paquetes de despliegue basándose en el estándar Dublin Core [33]. Dublin Core es un estándar para el manejo de metadatos utilizado por repositorios de objetos de aprendizaje para la localización e indexación de la información concerniente al objeto y así lograr búsquedas precisas y un manejo de contenido conveniente.

A través del uso de seguridad en los servicios web se controla el acceso a los contenidos tanto por usuarios, implementadores del modelo y sistemas administradores de aprendizaje.

El repositorio puede ser utilizado por 3 tipos de clientes diferentes: usuarios, implementadores del modelo (los cuales tienen la función de administrar y crear los contenidos del repositorio) y sistemas administradores de aprendizaje. Para cada uno de los clientes se manejan roles y privilegios de acceso.

Cualquier usuario y sistema administrador de aprendizaje autenticado ante el repositorio puede hacer uso del repositorio para realizar búsquedas basadas en los metadatos de los paquetes de contenidos además de búsquedas sencillas por autor o nombre del paquete de contenido. La diferencia principal entre un usuario y un sistema administrador de aprendizaje, radica en la forma de consultar los servicios web, ya que, mientras un usuario necesita de una interfaz intermedia para la búsqueda de información, un sistema administrador de aprendizaje hace uso directo de los servicios web disponibles. La Figura 5.0 muestra el diagrama general de casos de uso del repositorio.

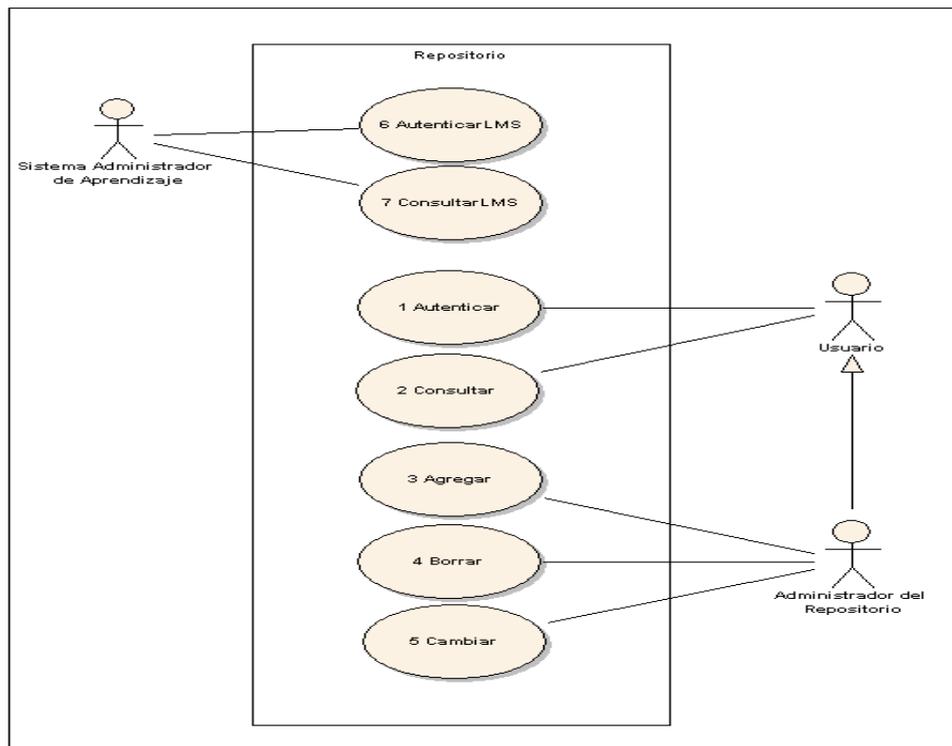


Figura 5.0 Diagrama General de Casos de Uso del Repositorio

En el diagrama general de casos de uso se muestra la funcionalidad de cada uno de los servicios web implementados:

Autenticar: Se asignan roles y permisos de acceso para los usuarios e implementadores del modelo.

Agregar: Añade un nuevo paquete de despliegue al repositorio llenando los metadatos necesarios para su posterior búsqueda.

Cambiar: Se localiza un paquete de contenido con el fin de cambiar los metadatos asociados a este.

Consultar: Se busca un paquete de contenido almacenado en el repositorio de dos maneras, una búsqueda sencilla en la cual se busca por autor, título o nombre del paquete de despliegue o por una búsqueda compleja en la cual se busca por cada uno de los campos definidos en el estándar Dublin Core.

Borrar: Se busca un paquete de contenido con la finalidad de eliminarlo borrando todas las relaciones existentes con otros paquetes de contenido.

AutenticarLMS: Asignar un rol y privilegios de acceso al LMS de una manera optima sin necesidad de una interfaz intermedia.

ConsultarLMS: Al igual que el servicio web para la consulta de los usuarios e implementadores del modelo se puede consultar los paquetes de contenidos por una búsqueda sencilla y avanzada, la diferencia principal radica en que no se necesita una interfaz intermedia para la realización de la búsqueda.

El implementador del modelo es el único usuario capaz de administrar el repositorio adicionando, borrando y cambiando los paquetes de contenidos. El implementador del modelo es el encargado de revisar el contenido de los paquetes de contenidos manteniendo el repositorio con contenidos coherentes y revisados.

El repositorio además cuenta con un sistema web de administración llamado el cual hace uso de los servicios creados para el manejo de los paquetes de contenido (ver figura 6.0). En las secciones posteriores se detallarán la funcionalidad y arquitectura del repositorio.

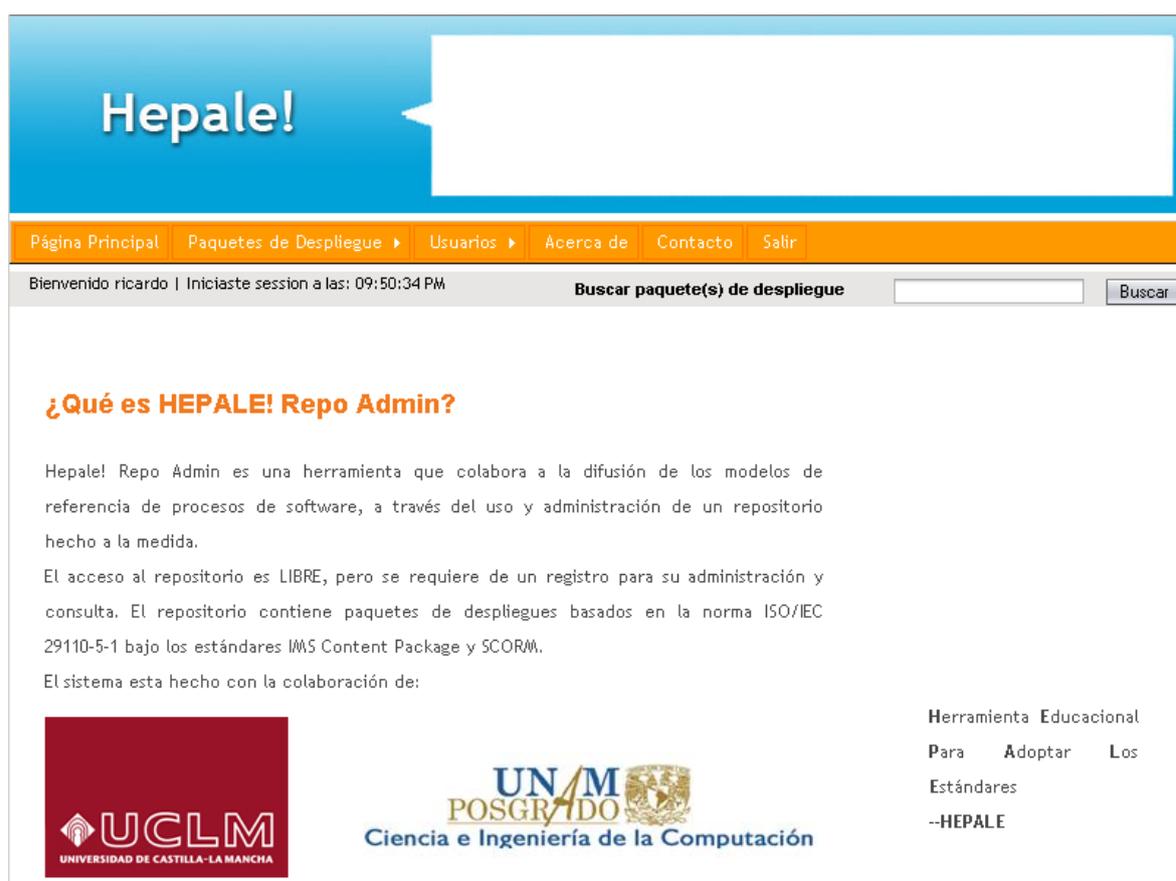


Figura 6.0 Sistema Web para la Administración del Repositorio

Una descripción en detalle de la especificación de requerimientos del repositorio puede consultarse en el apéndice A.

3.4 Análisis y Diseño del repositorio

Una vez establecidas las consideraciones generales y especificaciones de los requerimientos del repositorio y del sistema administrador, se procedió con la fase de análisis y diseño. Una descripción en detalle del análisis y diseño del repositorio puede consultarse en el apéndice B, por lo que tan solo se hará una descripción general del diseño y arquitectura del repositorio de paquetes de despliegue.

3.4.1 Arquitectura del Repositorio

El repositorio es construido bajo una Arquitectura Orientada a Servicios implementando Servicios Web. Los Servicios Web tienen la finalidad de proporcionar funcionalidades para la manipulación y almacenamiento de los paquetes de contenido y sus metadatos, los cuales permitirán la interoperabilidad entre distintos sistemas administradores de aprendizaje, automatizando procesos y facilitando la publicación y distribución de los paquetes en diferentes fuentes o aplicaciones.

La arquitectura utilizada en la creación del repositorio está basada en SOA (ver figura 8.0). Este tipo de arquitectura permite que diferentes aplicaciones intercambien datos a través de los servicios web y protocolos de comunicación (SOAP). En particular en este proyecto SOA permite el intercambio de paquetes de contenido entre el repositorio y las diferentes aplicaciones que hacen uso de ellos (Sistemas Web y LMS).

Existen varias razones por las que utilizó SOA.

- Reutilización. Ya que los mismos servicios web pueden ser reutilizados para crear nuevos.
- Interoperabilidad. Esto con el fin de que los servicios web ofrecidos puedan ser utilizados por distintos sistemas sin importar el lenguaje en el que fueron creados.
- Escalabilidad. Como los servicios SOA están débilmente acoplados, las aplicaciones que usan estos esos servicios escalan fácilmente. Esto debido a que existe muy poca dependencia entre las aplicaciones clientes y los servicios que usan.
- Flexibilidad. Es otra de las características que proporciona el acoplamiento débil entre los servicios. Cualquier cambio en la implementación de uno de ellos no afectaría al resto siempre que se mantenga la interfaz.

- Herramientas para la creación de servicios web y SOA existentes.

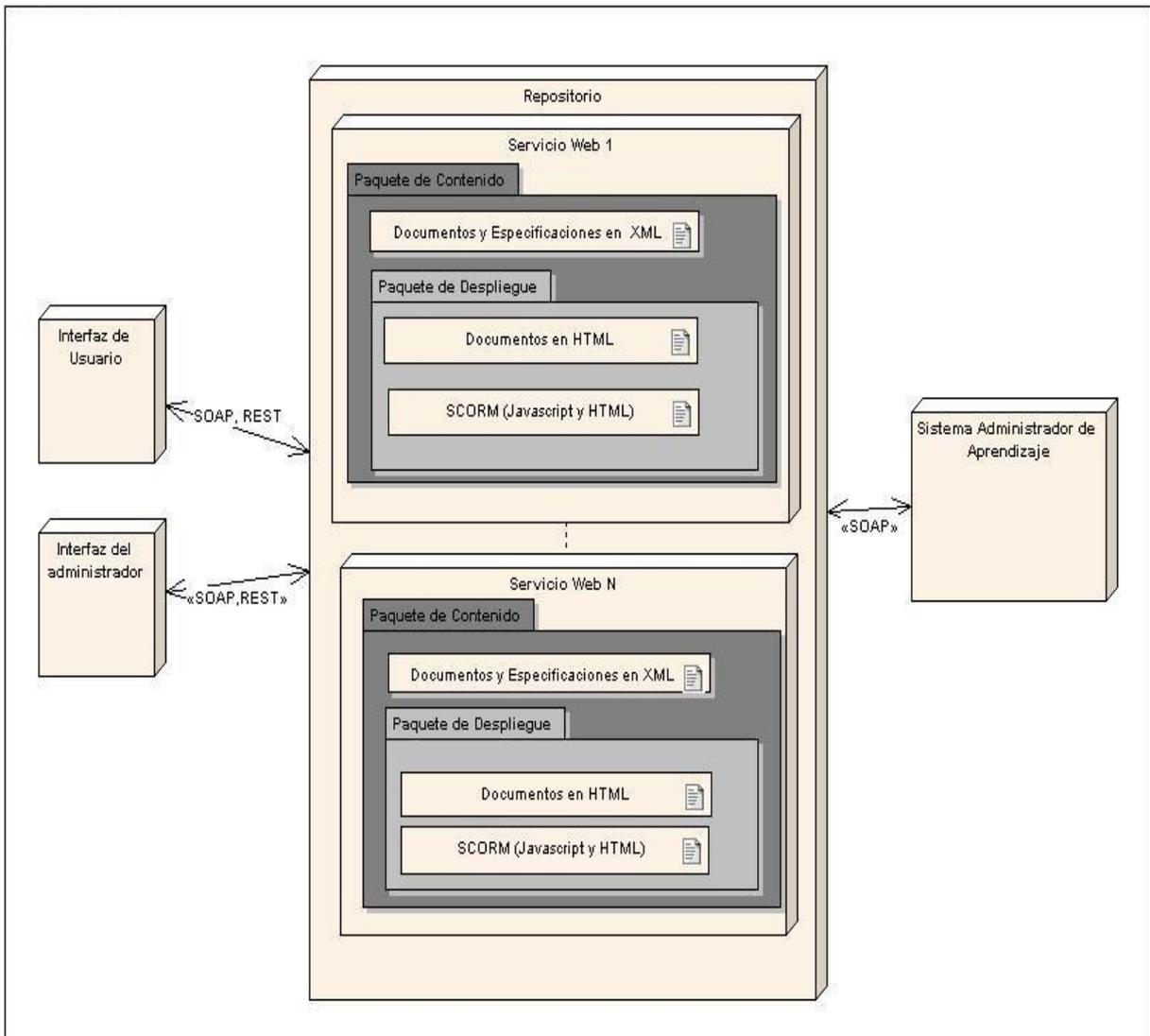


Figura 8.0 Arquitectura del Repositorio

Como se puede observar en la imagen anterior el repositorio tiene distintos servicios web, cada uno de ellos administra los distintos paquetes de despliegue almacenados en el repositorio bajo el estándar IMS Content Package (Paquetes de Contenido). Estos paquetes de contenido contienen archivos XML y páginas HTML permitiendo exponer su contenido de una manera estandarizada a través de los servicios web. Finalmente los servicios web se comunicarán con las distintas aplicaciones a través del protocolo de comunicación SOAP descrito anteriormente.

3.5 Construcción del Repositorio

Una vez definido el análisis, diseño y arquitectura, se crearon los servicios web y componentes necesarios para el manejo y manipulación de los paquetes de contenido. En el apéndice C detalla el proceso de construcción. Para la construcción de estos componentes y servicios se utilizaron las siguientes tecnologías:

- J2EE. Como marco de programación para desarrollar y ejecutar las funcionalidades que requieren los servicios web.
- WSIT. Utilizado en la creación de la arquitectura SOA e implementación de los servicios web; debido a las facilidades en cuanto a la implementación de los servicios web, además de la extensa documentación que puede ser encontrada en libros e internet.
- TopLink. Como manejador de persistencia de datos (acceso y manejo de bases de datos).
- Java Server Faces. Como marco de desarrollo para la creación del Sistema Web encargado de administrar el repositorio.
- Glassfish. Como servidor de aplicaciones.

Una vez creados los servicios web y componentes del repositorio se creó un sistema web, el cual funge como la interfaz gráfica del repositorio, además de encargarse de administrar los usuarios y paquetes de despliegue mediante el uso de los servicios web del repositorio.

Mediante el sistema, tanto los usuarios como los implementadores del modelo pueden consultar los distintos paquetes de despliegue del repositorio; ya sea mediante una consulta sencilla (se busca a través de los campos del identificador del paquete de despliegue, título, tema y autor) ó a través de una búsqueda avanzada (utilizando cualquiera de los campos definidos por Dublin Core). Las figuras 9.0, 9.1 y 9.2 muestran el uso de estas funcionalidades.



Administrar Paquetes de Despliegue

Buscar Paquete de Despliegue

[Búsqueda Avanzada](#)

[Mostrar todos los paquetes de despliegue](#)

Figura 9.0 Consulta sencilla de los paquetes de despliegue



Búsqueda Avanzada de Paquetes de Despliegue

Campo de Búsqueda	Palabra o Frase	Operador
<input type="text" value="Idpaquetedespliegue"/>	<input type="text"/>	<input type="text" value="AND"/>
<input type="text" value="Titulo"/>	<input type="text"/>	<input type="text" value="AND"/>
<input type="text" value="Autor"/>	<input type="text"/>	

Figura 9.1 Búsqueda avanzada de los paquetes de despliegue

Listar Paquetes de Despliegue

Elemento 1..5 de 7 [Restantes 2](#) [Crear nuevo Paquete de Despliegue](#)

Id	Titulo	Tema	Autor	Identificador	
2	Deployment Package Software Requirements Analysis For the Basic Profile	Requirements Analysis	Cruz Mendoza Carlos Ricardo	PaqueteScorm.zip	Detalles Editar Borrar
26	Paquete de Despliegue del Modelo de Mejora de Proceso de Proceso COMPETISOFT	modelo de mejora de procesos MOPROSOFT	Carlos Ricardo Cruz, Hanna Oktaba	IMSCP_DP_Modelo_de_Mejora_de_Procesos_COMPETISOFT.zip	Detalles Editar Borrar

Figura 9.2 Listado de los paquetes de despliegue

Otra funcionalidad de HEPAL! Repo Admin es el manejo de los metadatos de los paquetes de despliegue, las figuras 9.3 y 9.4 muestran la creación y edición de los paquetes de despliegue y sus metadatos.

Crear un nuevo paquete de despliegue

Titulo:	<input type="text"/>
Tema:	<input type="text"/>
Autor:	<input type="text"/>
Descripcion:	<input type="text"/>
Fuente:	<input type="text"/>
Lenguaje:	<input type="text"/>
Relacion:	<input type="text"/>
Covertura:	<input type="text"/>
Editor:	<input type="text"/>
Colaboradores:	<input type="text"/>
Derechos:	<input type="text"/>
Fecha:	<input type="text"/>
Tipo:	<input type="text"/>
Formato:	<input type="text"/>
Archivo:	<input type="text"/> <input type="button" value="Examinar..."/>

[Crear Nuevo paquete de Despliegue](#)

Figura 9.3 Creación de un nuevo paquete de despliegue

Editar paquete de despliegue

Id:	26
Título:	<input type="text" value="Paquete de Despliegue"/>
Tema:	<input type="text" value="modelo de mejora de p"/>
Autor:	<input type="text" value="Carlos Ricardo Cruz, H"/>
Descripción:	<input type="text"/>
Fuente:	<input type="text"/>
Lenguaje:	<input type="text"/>
Relación:	<input type="text"/>
Covertura:	<input type="text"/>
Editor:	<input type="text"/>
Colaboradores:	<input type="text"/>
Derechos:	<input type="text"/>
Fecha:	<input type="text"/>
Tipo:	<input type="text"/>
Formato:	<input type="text"/>

[Guardar Cambios](#)

Fig. 9.4 Edición de los metadatos de un paquete de despliegue

3.6 Pruebas

Con la finalidad de comprobar que el sistema realice las tareas indicadas en la especificación del problema, se realizaron pruebas controladas.

En el apéndice D se muestra el documento con los resultados obtenidos de las pruebas controladas. En este documento se describen las actividades que se realizaron en el repositorio, especificando los datos de entrada, salidas esperadas, salidas obtenidas, errores encontrados y posibles soluciones. Cabe mencionar que estas pruebas fueron realizadas tanto por el desarrollador del sistema como por un grupo de usuarios a los cuales se les indicó las tareas que se tenían que realizar.

La segunda prueba consistió en la utilización de los servicios web creados mediante la integración con un sistema tercero.

Las pruebas que se realizaron fueron las siguientes:

- Uso del servicio web para la consulta básica de los paquetes de despliegue.
- Uso del servicio web para la consulta avanzada de los paquetes de despliegue.

En la siguiente sección se muestra con detalle la integración y uso del repositorio con un sistema administrador de aprendizaje.

3.7 Uso del Repositorio

Según [35] [36] [37] un factor de éxito para la mejora de procesos en VSEs es la capacitación de los empleados en los procesos expuestos por el modelo o norma de desarrollo de procesos de software. Lo que permite a los empleados tener conocimiento sobre las buenas prácticas necesarias al momento de incorporarlas en el proceso de la organización. Se considera como propuesta para apoyar este factor de éxito el uso combinado del repositorio y una herramienta educacional a través del proyecto HEPALE! (Herramienta Educacional Para la Adopción de Estándares) [46].

HEPALE! aumenta la difusión y la distribución de los modelos y estándares a través del uso de los paquetes de despliegue. En la figura 10.0 se muestra el flujo de los paquetes de despliegue a través del proyecto HEPALE!

La figura 10.0 Muestra el diagrama general del proyecto HEPALE!

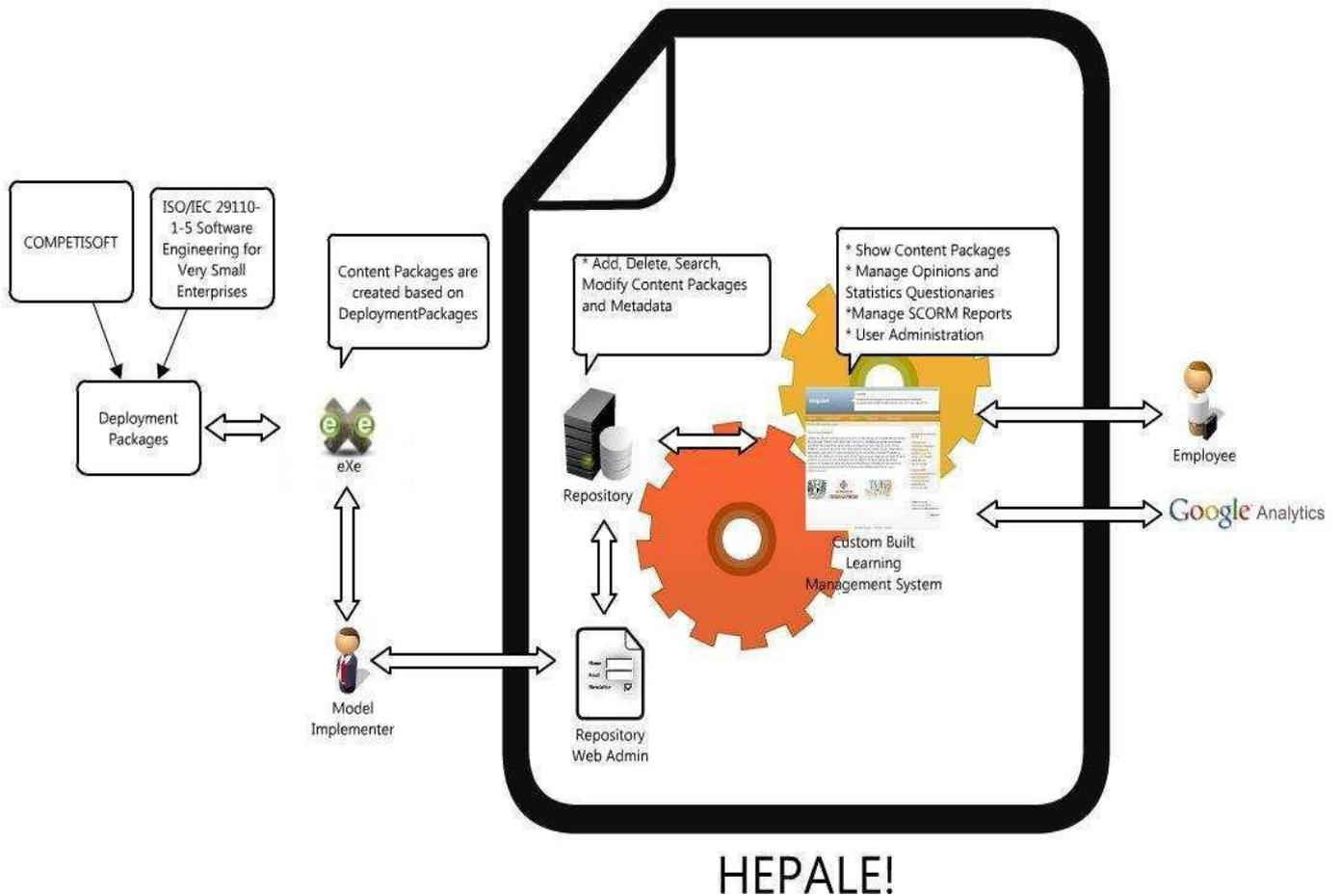


Figura 10.0 Flujo de un paquete de despliegue a través del proyecto HEPALE!

Como se puede observar en la imagen el primer paso es la creación de los paquetes de despliegue, una vez creados el implementador del modelo a través de la herramienta eXe digitaliza los paquetes de despliegue que serán almacenados en el repositorio. Una vez almacenados estos paquetes un sistema administrador de aprendizaje “hecho a la medida” se encarga de mostrar el contenido de estos paquetes, manejar estadísticas y opiniones y en general de todo el proceso de enseñanza a los empleados de las VSEs.

El proyecto HEPALE! es una herramienta creada recientemente que incorpora los estándares necesarios para utilizar los paquetes de despliegue, proveer un ambiente educativo que permitan a los empleados de las VSEs comprender los modelos o normas de desarrollo de procesos de software, y a los instructores, obtener información que permita mejorar la actividad instructiva.

Actualmente HEPALE! hace uso del repositorio mediante los servicios web explicados en este trabajo. Es importante ubicar que HEPALE! es una herramienta prototipo que no está diseñada para brindar todas las funcionalidades de los sistemas administradores de aprendizaje. En vez de esto, HEPALE! aprovecha la funcionalidad del repositorio con el fin de fomentar el

aprendizaje de los modelos de procesos de software y obtener información útil para la mejora de estos.

4 Conclusiones y trabajo a futuro

Según los objetivos de este trabajo:

- Encontrar la representación de paquetes de despliegue que facilite su difusión, distribución y aprendizaje a través de herramientas en línea.
- Crear un repositorio capaz de almacenar, administrar y distribuir los paquetes de despliegue.

Se puede concluir que:

- Se encontró una representación de los paquetes de despliegue capaz de facilitar su distribución, difusión y aprendizaje, mediante la utilización de una herramienta llamada eXe y utilizando estándares de e-learning.
- Se creó un repositorio capaz de almacenar, administrar y distribuir los distintos paquetes de despliegue.

Como trabajo a futuro hay tres tareas pendientes, como primera tarea se pretende crear una federación de repositorios de objetos de aprendizaje ó en su defecto, ingresar a una federación, con el fin de recopilar y reutilizar el material relacionado, proporcionando un sitio en el cual se pueda tener material actualizado y disponible para la creación de futuras aplicaciones.

En la segunda actividad se pretenden realizar mejoras al repositorio, una de ellas la mejora en la autenticación del repositorio mediante SSO (single sign-on), en la cual los clientes se identificarán ante un proveedor de identidad que permitirá el acceso a los distintos repositorios que se encuentren en la federación de repositorios, además de mejoras en el indexado y búsqueda de información; agregando funcionalidades al repositorio.

La tercera tarea es la más importante de todas ya que a corto plazo se pretenden crear nuevos paquetes de despliegue, con el fin de ser aplicados en una iniciativa de mejora de procesos dentro de una VSE, analizando la utilidad, y esfuerzo de de mejora.

Referencias Bibliográficas.

- [1] Richardson Ita, Gresse von Wangenheim Christiane, ***Why Are Small Software Organizations Different?***. IEEE Software, January/February 2007. Vol. (40). Pp.18-20.
- [2] ***Software Industry Statics for 1991-2005***, Enterprise Ireland, 2006; www.nsd.ie/htm/ssii/stat.htm, 15/01/2009.
- [3] Qualidade no Setor de Software Brasileiro-Pesquisa 2001, ***Quality in the Brazilian Software Sector-Survey 2001***, Ministry of Science and Technology (MTC), 2001; <http://ctjovem.mct.gov.br/index.php/content/view/34854.htm> (Texto en portugués), 15/01/2009.
- [4] Oktaba, H.; García, F., Piattini, M., Pino, F., Alquicira, C., Ruíz, F. ***Software Process Improvement: The COMPETISOFT Project***. October, 2007. Vol. 40(10), pp. 21-28. IEEE Computer.
- [5] ***COMPETISOFT***, <http://alarcos.infcr.uclm.es/Competisoft/>, 15/01/2009.
- [6] ***ISO/IEC CD TR 29110-5-1 Software Engineering - Lifecycle Profiles for Very Small Enterprises (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.
- [7] Claude Y. Laporte; Simon Alexandre, Alain Renault, ***Developing International Standards for Very Small Enterprises***. IEEE Computer 41(3): 98-101 (2008).
- [8] 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.
- [9] ***Personal Page of Claude Laporte***, <http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html>, 20/08/2009.
- [10] ***Deployment Packages***, <http://profs.logti.etsmtl.ca/claporte/English/VSE/VSE-packages.html>, 12/10/2009.
- [11] ***Learning Technology Standards Committee***, Institute of Electrical and Electronics Engineers IEEE, 2002, p.45.
- [12] Beck, Robert J., ***What Are Learning Objects?***, Center for International Education, University of Wisconsin-Milwaukee, http://www.uwm.edu/Dept/CIE/AOP/LO_what.html, 29/04/2008.

[13] Gertrudis L. María, Hernández Y., Beleño C., Pernalet D., Miguel V., Montaña N., *Un repositorio basado en Servicios Web para el Sistema Generados de Ambientes de Aprendizaje AMBAR*, Universidad Central de Venezuela.

[14] **PROSOFT**, <http://www.economia.gob.mx/?P=1128>, 26/10/2009.

[15] **MoProSoft: modelo de procesos de software hecho en México**, revista en línea Enter@te, <http://www.enterate.unam.mx/Articulos/2006/marzo/moprosoft.htm>, 27/10/2009.

[16] **Software Guru – COMPETISOFT**, <http://www.sg.com.mx/content/view/818/>, 28/10/2009.

[17] Oktaba, H., Piattini Mario., Pino, F., Orozco Julia y Alquicira Claudia, *COMPETISOFT Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos*, editorial RA-MA, 2008.

[18] 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.

[19] *Lecciones aprendidas del desarrollo de la norma internacional ISO/IEC 29110 basada en Moprosoft*, <http://www.slideshare.net/RevistaSG/lecciones-aprendidas-del-desarrollo-de-la-norma-internacional-isoiec-29110-basada-en-moprosoft>, 20/10/2009.

[20] Ma Carmen García, Javier Garzás, Mario Piattini, *La mejora de procesos en pequeñas empresas y la ISO/IEC 29110*, 20/10/2009, http://www.kybeleconsulting.com/downloads/MCGarcia_MejoraProcesos_ISO29110.pdf.

[21] De la Torre, César, Gonzáles, César. *Arquitectura SOA con tecnología MicroSoft*, editorial Krasis Press, 2008.

[22] *Universal Description, Discovery and Integration (UDDI)*: <http://www.uddi.org>, <http://uddi.microsoft.com/> y <http://www-3.ibm.com/services/uddi/>.

[23] SSA *Research Note* SPA-401-068, *Service Oriented' Architectures, Part 1*, 12/04/1996.

[24] Sicilia, M.A., *Estándares en e-learning. En Buenas Prácticas de E-Learning. UDIMA. ANCED*. Capítulo 16, 2007.

[25] Josep Vives (coord.), Ramon Alberch, Javier Álvarez, Aurora Cuevas, Ignasi Labastida, Glòria Munilla, Dora Pérez, M. Dolors Portús, Jordi Prats, Anna Rovira, Toni Rius, Laura Solanilla i Josep Vives, *Digitalización del patrimonio: archivos, bibliotecas y museos en la red*, Ed. Uoc, Junio 2009.

[26] **Dspace@MIT**, <http://dspace.mit.edu/>, 20/10/2009.

- [27] *Campus Alberta Repository of Educational Objects, CAREO*, <http://www.ucalgary.ca/commons/careo/repository.htm>, 20/10/2009
- [28] Learning Technology Standards Committee, Institute of Electrical and Electronics Engineers IEEE, 2002, p.45.
- [29] *Moodle*, <http://moodle.org>, 20/10/2009.
- [30] *Blackboard*, <http://www.blackboard.com/>, 20/10/2009.
- [31] *SCORM*, <http://www.adlnet.gov/scorm/>, 12/03/2009.
- [32] *IMS Content Packaging*, <http://www.imsproject.org/content/packaging/>, 12/03/2009.
- [33] *Dublin Core*, <http://dublincore.org/>, 20/10/2009
- [34] *Elearning XHTML Editor, eXe*, <http://sourceforge.net/apps/trac/exe/wiki>, 21/10/2009.
- [35] Pino, Francisco. García Felix y Piattini Mario. *Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas*. *Revista Española de Innovación, Calidad e Ingeniería de Software*, Vol 2, No. 1, 2006, pp. 6-23.
- [36] Niazi, M., D. Wilson and D. Zowghi. *Implementing Software Process Improvement Initiatives. An empirical Study*, 2006. Product-Focused Software Process Improvement, *7th International Conference, PROFES 2006*, Amsterdam. The Netherlands, Springer, pp. 92-99.
- [37] Conradi, R. and A. Fuggeta, *Improving Software Process Improvement. IEEE Software*, 2002. Vol 19(4), July/August pp 92-99.
- [38] Morales Trujillo E. Miguel, *“Mejora de Procesos de Software basada en plantillas”*, Tesis de Maestría, Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México.
- [39] CONDUSEF. *La Secretaría de Economía y las MIPYME*. Gobierno de México. http://www.condusef.gob.mx/index.php?option=com_content&view=article&id=758&Itemid=80, 17-12-2008.
- [40] International Organization for Standardization, *ISO 9001:2000*, http://www.iso.org/iso/management_standards.htm, 29/11/2009.
- [41] International Organization for Standardization, *ISO/IEC 15504*, <http://www.isospice.com/>, 29/11/2009.
- [42] *Capability Maturity Model Integration (CMMI)*, Software Engineering Institute, Carnegie Mellon, <http://www.sei.cmu.edu/cmmi/start/>, 29/10/2009.

[43] 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.

[44] World Wide Web Consortium, *SOAP*, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, 20/03/2010

[45] Macario Polo, Mario Piattini, Francisco Ruiz, Coral 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, Page 76, 1999.

[46] Morgado Castillo Mauricio, “*Herramienta Educativa Para La Adopción De Procesos*”, Tesis de Maestría, Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México.

Apéndice A. Especificación de Requerimientos



Índice Apéndice A.

1. Introducción	3
1.1 Propósito	3
1.2 Alcance	3
1.3 Definiciones, Acrónimos y Abreviaturas	3
1.4 Referencias	4
2. Descripción general	4
3. Requisitos No Funcionales	4
3.1 Usabilidad	4
3.1.1 <RU 1>	4
3.2 Confiabilidad	4
3.2.1 <RC 1>	4
3.3 Eficiencia	4
3.3.1 <RE1 Tiempos de respuesta>	4
3.3.2 <RE1 Capacidad>	4
3.4 Mantenimiento	5
3.4.1 <RM1 Documentación>	5
3.5 Portabilidad	5
3.5.1 <RP1 Portabilidad del repositorio>	5
3.5.2 <RP2 Portabilidad de las herramientas>	5
3.6 Restricciones de diseño y construcción	5
3.6.1 <RDC1 Restricción de diseño y construcción del repositorio>	5
3.6.2 <RDC2 Restricción de diseño y construcción de los paquetes de contenido>	5
3.7 Reusabilidad	5
3.8 Interfaces	5
3.8.1 Interfaces de Usuario	5
3.8.2 Interfaces de Hardware	5
3.8.3 Interfaces de Software	5
3.8.4 Interfaces de Comunicación	6
3.9 Licenciamiento	6
3.10 Avisos Legales, Derechos de autor y otros	6
4. Requisitos Funcionales	7
4.1 Identificación de los casos de uso	7
4.2 Priorización de los casos de uso	8
4.3 Detallado de los casos de uso	8
4.3.1 <CU Autenticar 01>	8
4.3.1.1 Descripción breve del caso de uso	8
4.3.1.2 Flujo de eventos	8
4.3.1.2.1 Flujo básico	8
4.3.1.2.2 Flujo alternativo	9
4.3.1.3 Precondiciones	9
4.3.1.4 Poscondiciones	9
4.3.2 <CU Consultar 02>	9

4.3.2.1 Descripción breve del caso de uso	9
4.3.2.2 Flujo de eventos	10
4.3.2.2.1 Flujo básico.....	10
4.3.2.2.2 Flujo alternativo.....	12
4.3.2.3 Precondiciones.....	12
4.3.2.4 Poscondiciones	12
4.3.3 <CU Agregar 03>	12
4.3.3.1 Descripción breve del caso de uso.....	12
4.3.3.2 Flujo de eventos.....	12
4.3.3.2.1 Flujo básico.....	12
4.3.3.2.2 Flujo alternativo.....	14
4.3.3.3 Precondiciones.....	14
4.3.3.4 Poscondiciones	14
4.3.4 <CU Borrar 04>	15
4.3.4.1 Descripción breve del caso de uso.....	15
4.3.4.2 Flujo de eventos.....	15
4.3.4.2.1 Flujo básico.....	15
4.3.4.2.2 Flujo alternativo.....	18
4.3.4.3 Precondiciones.....	18
4.3.4.4 Poscondiciones	18
4.3.5 <CU Cambiar 05>	18
4.3.5.1 Descripción breve del caso de uso.....	18
4.3.5.2 Flujo de eventos.....	18
4.3.5.2.1 Flujo básico.....	18
4.3.5.2.2 Flujo alternativo.....	20
4.3.5.3 Precondiciones.....	20
4.3.5.4 Poscondiciones	20
4.3.6 <CU AutenticarLMS 06>	21
4.3.6.1 Descripción breve del caso de uso.....	21
4.3.6.1.1 Flujo básico.....	21
4.3.6.1.2 Flujo alternativo.....	21
4.3.6.2 Precondiciones.....	21
4.3.6.3 Poscondiciones	22
4.3.7 <CU Consultar 02>	22
4.3.7.1 Descripción breve del caso de uso.....	22
4.3.7.2 Flujo de eventos.....	22
4.3.7.2.1 Flujo básico.....	22
4.4 Prototipo de la interfaz del usuario	23
4.4.1 <Pantalla 1>	23
5. Justificación de los requerimientos de software	23
6. Información de apoyo	23

1.Introducción

Se desea realizar un repositorio de paquetes de despliegue para los modelos de procesos ISO/IEC 29110-5-1 y COMPETISOFT tomando en cuenta distintos estándares para la creación, representación, interacción y manipulación de Objetos de Aprendizaje.

1.1Propósito

El propósito del sistema es la creación de un repositorio que ofrezca distintos Paquetes de Contenidos (IMS Content Packaging). Estos Paquetes de Contenidos almacenan Objetos de Aprendizaje de los paquetes de despliegue de los modelos de procesos ISO/IEC 20110-5-1 y COMPETISOFT.

Los Objetos de Aprendizaje y sus metadatos serán almacenados utilizando la especificación de Paquetes de Contenidos y los datos representados mediante SCORM, con el fin de mantener un repositorio que sea compatible con distintos Sistemas de Administración de Aprendizaje (LMS por sus siglas en inglés).

Se deberán implementar Servicios Web (Web Services) para poder dar de alta, borrar, consultar y cambiar los Paquetes de Contenidos.

1.2Alcance

El sistema ofrecerá una única funcionalidad la cual consiste en ofrecer los Paquetes de Contenidos a través de servicios web para darlos de alta, borrarlos, cambiarlos y consultarlos.

Los servicios web para dar de alta, borrar y cambiar los Paquetes de Contenidos solamente podrán ser utilizados por el administrador del repositorio, mientras que para poder consultar los Paquetes de Contenidos los usuarios se tendrán que autenticar ante el sistema.

1.3Definiciones, Acrónimos y Abreviaturas

Término/Acrónimo/Abreviatura	Definición
ISO/IEC 29110-5-1	Norma Internacional para la Mejora de Procesos de Software.
COMPETISOFT	Modelo de Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos.
Objeto de Aprendizaje	Cualquier recurso digital que pueda ser reutilizado para apoyar la enseñanza.
Paquete de Despliegue	Guías y materiales para seguir correctamente y con buenas prácticas un proceso de algún modelo de desarrollo de software.
Paquete de Contenido	Archivo comprimido en el cual se guarda objetos de aprendizaje y metadatos asociados a estos.
SCORM	Es una colección de estándares y especificaciones basados en e-learning implementados a través de HTML y javascript.
E-learning	Concepto de educación a distancia en el que se integra el uso de las tecnologías de la información y otros elementos didácticos para la formación, capacitación y enseñanza de los usuarios o estudiantes en línea.
SOA	Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a la capa de procesos de negocio.

Servicios Web	Conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
Sistema de Administración de Aprendizaje	Programa empelado para distribuir, administrar y controlar el proceso de enseñanza.

1.4 Referencias

COMPETISOFT, <http://alarcos.inf-cr.uclm.es/Competisoft/>, 15/01/2009

ISO/IEC 29110-5-1, ISO/IEC CD TR 29110-5-1 Software Engineering - Lifecycle Profiles for Very Small Enterprises (VSE) -- Part 5-1: Management and Engineering Guide -- Basic Profile, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51160, 15/01/2009

Learning Object, Learning Technology Standards Committee, Institute of Electrical and Electronics Engineers IEEE, 2002, p.45

IMS Content Packaging, <http://www.imsproject.org/content/packaging/>, 12/03/2009

SCORM, <http://www.adlnet.gov/scorm/>, 12/03/2009

Learning Management Systems, http://en.wikipedia.org/wiki/Learning_management_system, 12/03/2009

Newcomer, Eric; Lomow, Greg (2005). *Understanding SOA with Web Services*. Addison Wesley. ISBN 0-321-18086-0.

2. Descripción general

El repositorio contará con paquetes de contenidos los cuales podrán ser dados de alta, baja y consultarlos. Se espera que cualquier sistema de administración de aprendizaje pueda utilizar estos paquetes de contenidos. Los paquetes de contenidos deberán utilizar estándares de e-learning tal es el caso de SCORM. La licencia, instalación y criterios de calidad del sistema se definirá en una futura revisión.

3. Requisitos No Funcionales

3.1 Usabilidad

3.1.1 <RU 1>

3.2 Confiabilidad

3.2.1 <RC 1>

3.3 Eficiencia

3.3.1 <RE1 Tiempos de respuesta>

El repositorio deberá contar con un tiempo de respuesta a peticiones menor a 3 segundos. El tiempo promedio de respuesta deberá ser menor a un segundo.

3.3.2 <RE1 Capacidad>

El repositorio solamente podrá atender a 10 personas a la vez.

3.4 Mantenimiento

3.4.1 <RM1 Documentación>

El sistema deberá contar con documentación en cada clase y método, además de una especificación amplia de los estándares y algoritmos utilizados en la construcción del repositorio.

3.5 Portabilidad

3.5.1 <RP1 Portabilidad del repositorio>

El sistema deberá ser utilizado en distintos sistemas operativos (Windows, Linux, MacOS).

3.5.2 <RP2 Portabilidad de las herramientas>

Las herramientas utilizadas deben poderse utilizar indistintamente entre distintos sistemas operativos (Windows, Linux, MacOS).

3.6 Restricciones de diseño y construcción

3.6.1 <RDC1 Restricción de diseño y construcción del repositorio>

El repositorio deberá ser construido bajo una Arquitectura Orientada a Servicios (SOA) utilizando el framework de desarrollo GRAILS más el plugin XFIRE, los cuales permitan la interoperabilidad entre los diferentes LMS, automatizando procesos de negocio y facilitando la publicación de la información en diferentes fuentes.

3.6.2 <RDC2 Restricción de diseño y construcción de los paquetes de contenido>

Mediante la utilización de la herramienta eXe (eLearning XHTML editor) se deberán crear los distintos Paquetes de Contenido facilitando la implementación de los distintos estándares y especificaciones utilizados en SCORM.

3.7 Reusabilidad

3.8 Interfaces

3.8.1 Interfaces de Usuario

3.8.2 Interfaces de Hardware

3.8.3 Interfaces de Software

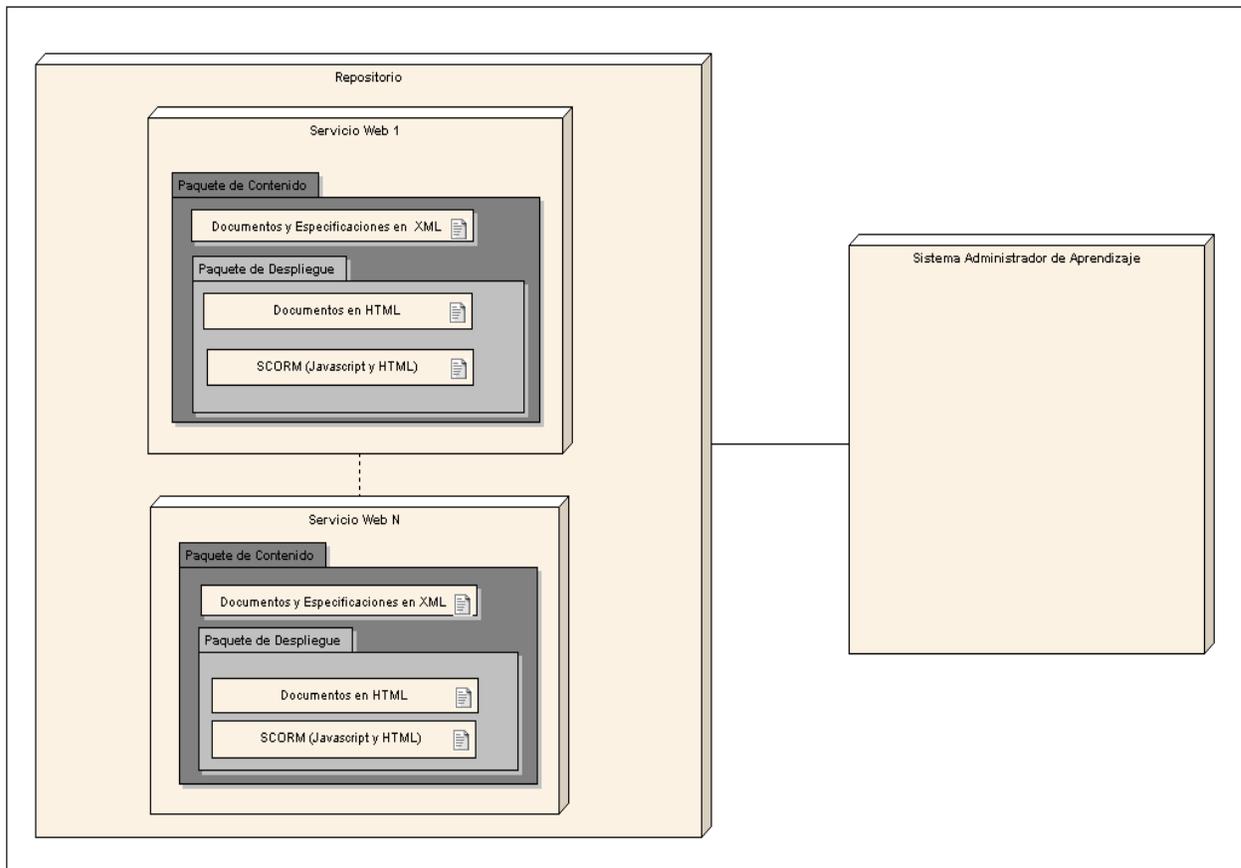


Fig. 3.8.3.1 Diagrama de despliegue del sistema

En la figura 3.8.3.1 se muestran los componentes principales del repositorio.

El repositorio constará de servicios web los cuales tendrán asociados paquetes de contenidos. Estos paquetes de contenidos serán archivos comprimidos que contienen archivos en XML, documentos HTML y javascript, los cuales implementarán los distintos estándares necesarios para que puedan ser utilizados por el sistema administrador de aprendizaje.

3.8.4 Interfaces de Comunicación

3.9 Licenciamiento

3.10 Avisos Legales, Derechos de autor y otros

4.Requisitos Funcionales

4.1Identificación de los casos de uso

En la Fig. 4.1 se puede observar el caso de uso general del sistema.

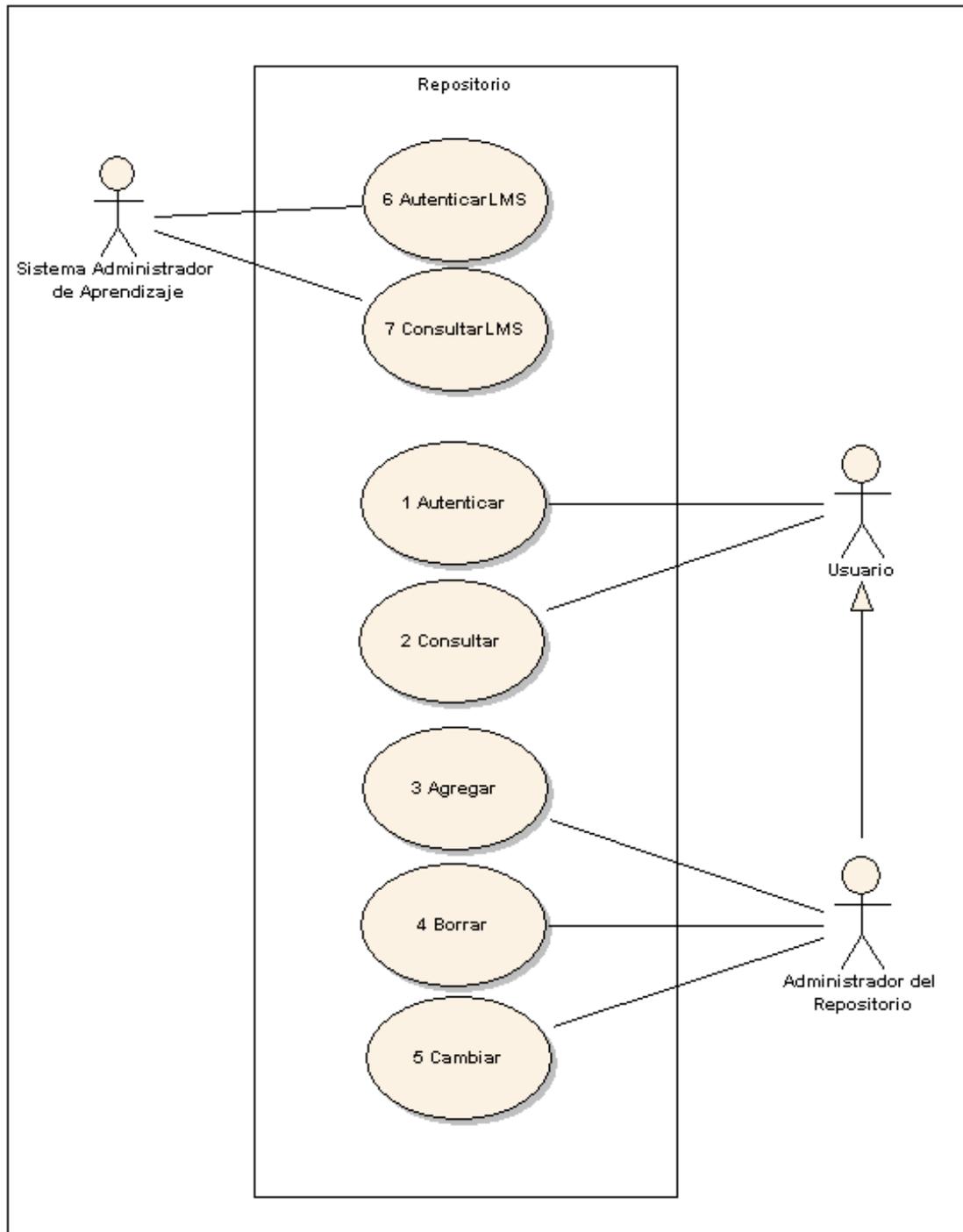


Fig. 4.1 Diagrama General de Casos de Uso

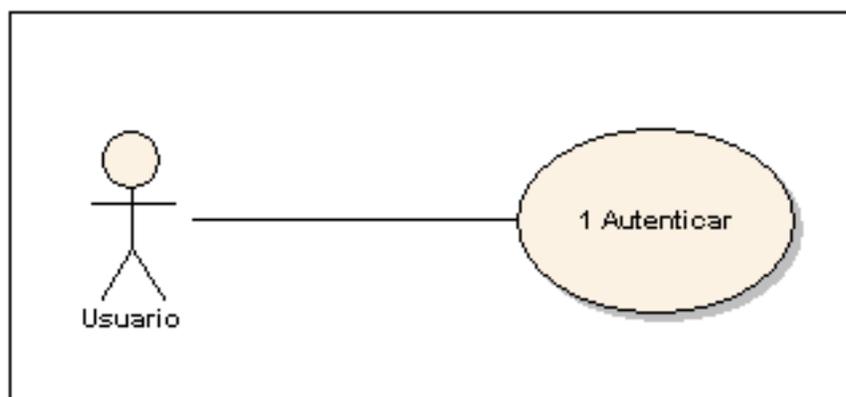
Los casos de uso para agregar, borrar y cambiar los Paquetes de Contenidos solamente podrán ser utilizados por el administrador del repositorio, mientras que para poder consultar los Paquetes de Contenidos los sistemas administradores de aprendizaje tendrán que autenticarse ante el repositorio

4.2 Priorización de los casos de uso

Caso de Uso	Prioridad
AutenticarLMS	Alta
ConsultarLMS	Alta
Autenticar	Alta
Consultar	Alta
Agregar	Media
Borrar	Media
Cambiar	Baja

4.3 Detallado de los casos de uso

4.3.1 <CU Autenticar 01>



4.3.1.1 Descripción breve del caso de uso

Se comprueba el certificado de seguridad, nombre de usuario y contraseña con el cual se autenticaran tanto el administrador del sistema como los usuarios ante el repositorio.

4.3.1.2 Flujo de eventos

4.3.1.2.1 Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El usuario ingresa su nombre de usuario y contraseña. (REQ01)	2	El repositorio comprueba la información ingresada y busca por el nombre de usuario y contraseña ingresado por el usuario. El repositorio comprueba el certificado de seguridad para comprobar si es correcto. El repositorio obtiene el rol asociado al usuario.	E1,E2,E3

		3	El repositorio queda configurado para proporcionar las futuras peticiones de acuerdo al rol asociado.	
--	--	---	---	--

ID	TIPO	ACCION
E1	La información es incorrecta.	El repositorio manda un mensaje al cliente indicando que verifique la información proporcionada ya que no se pudo identificar al usuario.
E2	El usuario no existe en el sistema	El repositorio indica que verifique la información proporcionada ya que no se pudo identificar al usuario.
E3	El certificado de seguridad no es correcto.	El repositorio indica al usuario que su certificado de seguridad no es correcto.

REQ01		
DATO	DESCRIPCIÓN	REQUERIDO
Nombre de usuario	Nombre único que identifica a un usuario del sistema	S
Contraseña	Clave alfanumerica generada a partir de algún algoritmo de cifrado (simétrico o asimétrico), no puede ser obtenido su valor a partir de la clave generada.	S

4.3.1.2.2Flujo alternativo

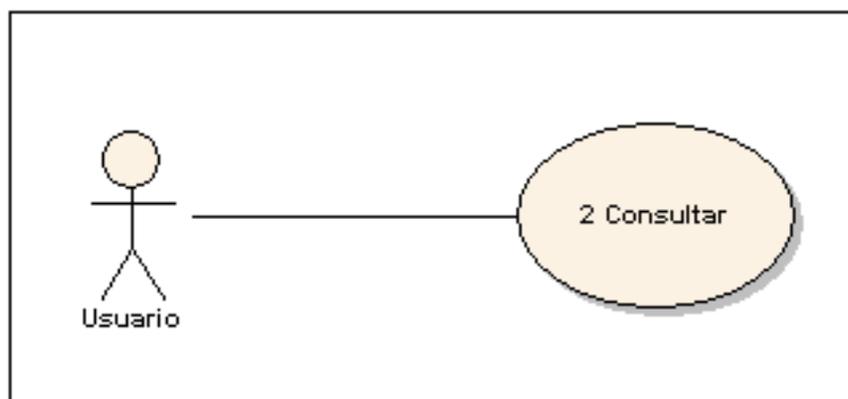
4.3.1.3Precondiciones

El usuario deberá estar dado de alta previamente por el administrador del repositorio.

4.3.1.4Poscondiciones

El usuario o sistema podrá realizar más acciones en el repositorio de acuerdo al rol asignado.

4.3.2<CU Consultar 02>



4.3.2.1Descripción breve del caso de uso

Tanto el usuario como el administrador del sistema tiene la misma interfaz para la consulta de información del repositorio, esta consulta se realiza a través del nombre/título del paquete de contenido ó a través de los

metadatos definidos en el estándar Dublin Core.

4.3.2.2 Flujo de eventos

4.3.2.2.1 Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El usuario selecciona la opción “Consultas”	2	El sistema muestra una página con las opciones para consultar un paquete de contenido.	
3	El usuario introduce los datos necesarios para la consulta. (REQ01)			
4	El usuario envía su consulta	5	El sistema valida los datos	E1
		6	Se regresa el paquete de contenido que concuerde con la búsqueda solicitada	E2

ID	TIPO	ACCION
E1	No se selecciona ningún campo.	El sistema no regresa ningún dato.
E2	Ningún paquete de contenido concuerda con los datos introducidos en la búsqueda.	El sistema regresa un mensaje informando al usuario que su búsqueda no arroja ningún resultado.

REQ01		
DATO	DESCRIPCIÓN	REQUERIDO
Título o Nombre	Es el nombre dado al recurso, habitualmente por el autor. Etiqueta: DC.Title	S
Claves	Los tópicos del recurso. Típicamente, se expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales. Etiqueta: DC.Subject	S
Descripción	Una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual. Etiqueta: DC.Description	N
Fuente	Secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. Etiqueta: DC.Source	N
Lenguaje	Lengua/s del contenido intelectual del recurso.	N

	Etiqueta: DC.Language	
Relación	Es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos. Etiqueta: DC.Relation	N
Cobertura	Es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado (que ya lo encontramos en el elemento Fecha). Etiqueta: DC.Coverage	N
Autor o Creador	La persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales. Etiqueta: DC.Creator	S
Editor	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual Etiqueta: DC.Publisher	N
Otros Colaboradores	Una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creador. (Por ejemplo: editor, ilustrador y traductor). Etiqueta: DC.Contributor	N
Derechos	Son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso. Etiqueta: DC.Rights	N
Fecha	Una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha. DC.Date	N
Tipo del Recurso	La categoría del recurso. Por ejemplo, página personal, romance, poema, diccionario, etc. Etiqueta: DC.Type	N
Formato	Es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso. Etiqueta: DC.Format	N

Identificador del Recurso	Secuencia de caracteres utilizados para identificar unívocamente un recurso. Ejemplos para recursos en línea pueden ser URLs i URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN ("International Standard Book Number"). Etiqueta: DC.Identifier	N
----------------------------------	--	---

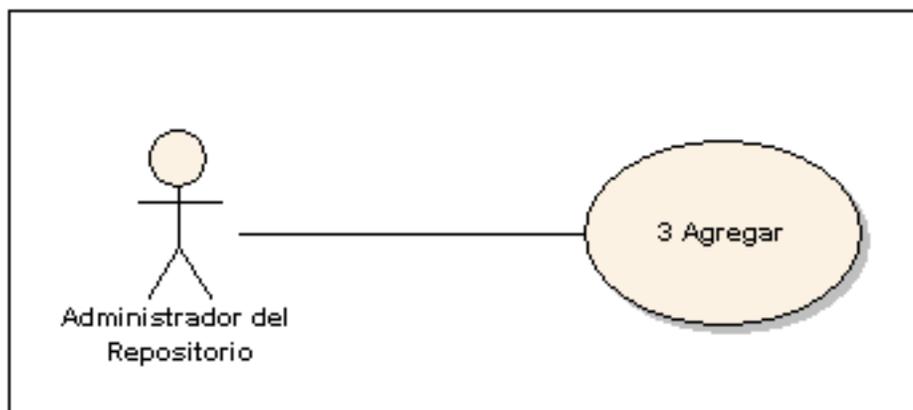
4.3.2.2Flujo alternativo

4.3.2.3Precondiciones

El usuario o sistema deberá estar dado de alta previamente por el administrador del repositorio.

4.3.2.4Poscondiciones

4.3.3<CU Agregar 03>



4.3.3.1Descripción breve del caso de uso

El administrador del sistema agregara un paquete de contenido previamente creado con los metadatos correspondientes definidos en el estándar Dublin Core.

4.3.3.2Flujo de eventos

4.3.3.2.1Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El administrador del repositorio selecciona la opción “Agregar”	2	El sistema muestra una página con las opciones para dar de alta un paquete de contenido.	
3	El administrador introduce los datos necesarios para el alta del paquete de contenido. (REQ01)			
4	El usuario envía su petición.	5	El sistema valida los datos	E1
		6	Se regresa un mensaje indicando que el paquete de contenido con sus metadatos fue agregado correctamente.	E2

ID	TIPO	ACCION
E1	No se selecciona ningún campo.	El sistema no agrega ningún paquete de contenido y manda un mensaje indicando los datos necesarios por llenar.
E2	Existe un error en la comunicación con el repositorio	Se muestra un mensaje de error indicando que no fue posible comunicarse con el repositorio.

REQ01		
DATO	DESCRIPCIÓN	REQUERIDO
Dirección del paquete de contenido	Ruta en la maquina local hacia el paquete de contenido.	S
Título o Nombre	Es el nombre dado al recurso, habitualmente por el autor. Etiqueta: DC.Title	S
Claves	Los tópicos del recurso. Típicamente, se expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales. Etiqueta: DC.Subject	S
Descripción	Una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual. Etiqueta: DC.Description	N
Fuente	Secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. Etiqueta: DC.Source	N
Lenguaje	Lengua/s del contenido intelectual del recurso. Etiqueta: DC.Language	N
Relación	Es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos. Etiqueta: DC.Relation	N
Cobertura	Es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado (que ya lo encontramos en el elemento Fecha). Etiqueta: DC.Coverage	N
Autor o Creador	La persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de	S

	recursos visuales. Etiqueta: DC.Creator	
Editor	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual Etiqueta: DC.Publisher	N
Otros Colaboradores	Una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creador. (Por ejemplo: editor, ilustrador y traductor). Etiqueta: DC.Contributor	N
Derechos	Son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso. Etiqueta: DC.Rights	N
Fecha	Una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha. DC.Date	N
Tipo del Recurso	La categoría del recurso. Por ejemplo, página personal, romance, poema, diccionario, etc. Etiqueta: DC.Type	N
Formato	Es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso. Etiqueta: DC.Format	N
Identificador del Recurso	Secuencia de caracteres utilizados para identificar unívocamente un recurso. Ejemplos para recursos en línea pueden ser URLs i URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN ("International Standard Book Number"). Etiqueta: DC.Identifier	N

4.3.3.2Flujo alternativo

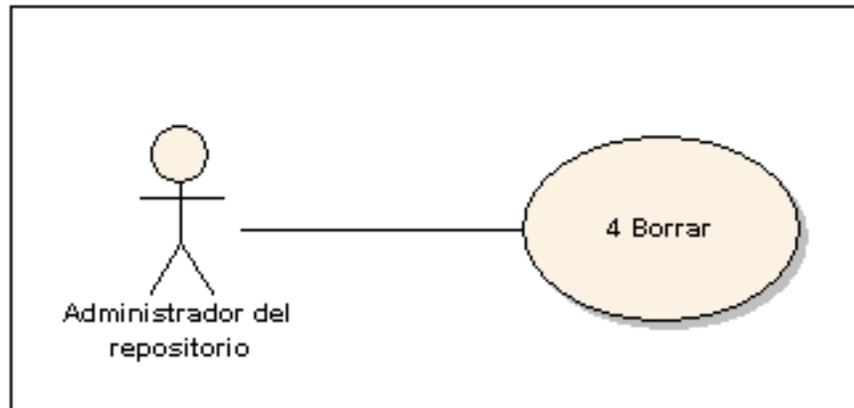
4.3.3.3Precondiciones

El usuario o sistema deberá estar dado de alta previamente por el administrador del repositorio.

4.3.3.4Poscondiciones

El repositorio tendrá disponible un nuevo paquete de contenido.

4.3.4<CU Borrar 04>



4.3.4.1 Descripción breve del caso de uso

El administrador del sistema borrará un paquete de contenido previamente creado con los metadatos correspondientes definidos en el estándar Dublin Core.

4.3.4.2 Flujo de eventos

4.3.4.2.1 Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El administrador del repositorio selecciona la opción "Borrar"	2	El sistema muestra una página con las opciones para dar de alta un paquete de contenido.	
3	El administrador realiza una consulta para encontrar el paquete(s) a borrar(REQ01)			
4	El administrador envía su petición.	5	El sistema valida los datos	E1
		6	Se regresan un resumen con los datos del o los paquetes a borrar. (REQ02)	E2,E3
5	El usuario selecciona el paquete(s) a borrar.			
		7	El sistema muestra un mensaje indicando si el paquete fue borrado.	E3

ID	TIPO	ACCION
E1	No se selecciona ningún campo.	El sistema no realiza ninguna consulta y manda un mensaje indicando los datos necesarios por llenar.
E2	No se encontró ningún paquete de contenido con el criterio indicado.	El sistema manda un mensaje para indicar que no se encontró ningún paquete.
E3	.Existe un error en la comunicación con el repositorio	Se muestra un mensaje de error indicando que no fue posible comunicarse con el repositorio.

REQ01

DATO	DESCRIPCIÓN	REQUERIDO
Título o Nombre	Es el nombre dado al recurso, habitualmente por el autor. Etiqueta: DC.Title	S
Claves	Los tópicos del recurso. Típicamente, se expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales. Etiqueta: DC.Subject	S
Descripción	Una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual. Etiqueta: DC.Description	N
Fuente	Secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. Etiqueta: DC.Source	N
Lenguaje	Lengua/s del contenido intelectual del recurso. Etiqueta: DC.Language	N
Relación	Es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos. Etiqueta: DC.Relation	N
Cobertura	Es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado (que ya lo encontramos en el elemento Fecha). Etiqueta: DC.Coverage	N
Autor o Creador	La persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales. Etiqueta: DC.Creator	S
Editor	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual Etiqueta: DC.Publisher	N
Otros Colaboradores	Una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creador. (Por ejemplo: editor, ilustrador y traductor). Etiqueta: DC.Contributor	N

Derechos	Son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso. Etiqueta: DC.Rights	N
Fecha	Una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha. DC.Date	N
Tipo del Recurso	La categoría del recurso. Por ejemplo, página personal, romance, poema, diccionario, etc. Etiqueta: DC.Type	N
Formato	Es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso. Etiqueta: DC.Format	N
Identificador del Recurso	Secuencia de caracteres utilizados para identificar unívocamente un recurso. Ejemplos para recursos en línea pueden ser URLs i URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN ("International Standard Book Number"). Etiqueta: DC.Identifier	N

REQ02		
DATO	DESCRIPCIÓN	REQUERIDO
Título o Nombre	Es el nombre dado al recurso, habitualmente por el autor. Etiqueta: DC.Title	S
Claves	Los tópicos del recurso. Típicamente, se expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales. Etiqueta: DC.Subject	S
Descripción	Una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual. Etiqueta: DC.Description	N
Autor o Creador	La persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales. Etiqueta: DC.Creator	S

4.3.4.2 Flujo alternativo

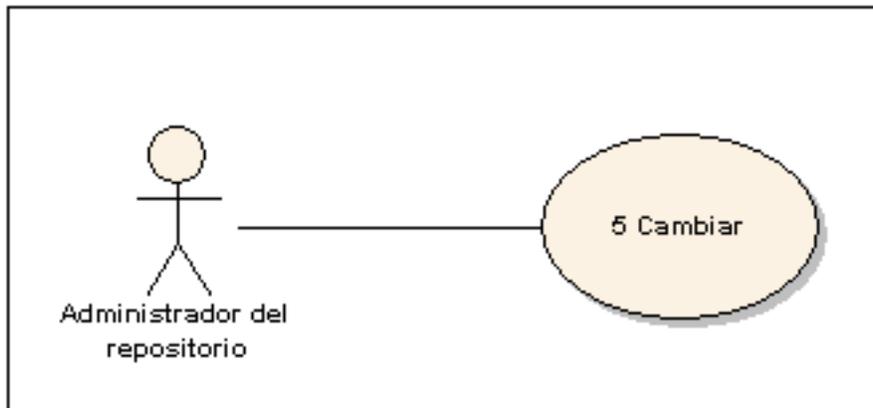
4.3.4.3 Precondiciones

El usuario o sistema deberá estar dado de alta previamente por el administrador del repositorio.

4.3.4.4 Poscondiciones

El repositorio tendrá uno o varios paquetes de contenidos menos.

4.3.5 <CU Cambiar 05>



4.3.5.1 Descripción breve del caso de uso

El administrador del sistema cambiará algún metadato de un paquete de contenido agregado previamente.

4.3.5.2 Flujo de eventos

4.3.5.2.1 Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El administrador del repositorio selecciona la opción "Cambiar"	2	El sistema muestra una página con las opciones para cambiar los metadatos de un paquete de contenido.	
3	El administrador realiza una consulta para encontrar el paquete(s) a cambiar(REQ01)			
4	El administrador envía su petición.	5	El sistema valida los datos	E1
		6	Se regresan los datos del paquete al cual se le quieren cambiar los metadatos. (REQ01)	E2,E3
5	El usuario selecciona el paquete(s) a cambiar.			
		7	El sistema muestra un mensaje indicando si el paquete fue cambiado y un resumen con los datos cambiados.	E3

ID	TIPO	ACCION
----	------	--------

E1	No se selecciona ningún campo.	El sistema no realiza ninguna consulta y manda un mensaje indicando los datos necesarios por llenar.
E2	No se encontró ningún paquete de contenido con el criterio indicado.	El sistema manda un mensaje para indicar que no se encontró ningún paquete.
E3	Existe un error en la comunicación con el repositorio	Se muestra un mensaje de error indicando que no fue posible comunicarse con el repositorio.

REQ01		
DATO	DESCRIPCIÓN	REQUERIDO
Título o Nombre	Es el nombre dado al recurso, habitualmente por el autor. Etiqueta: DC.Title	S
Claves	Los tópicos del recurso. Típicamente, se expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales. Etiqueta: DC.Subject	S
Descripción	Una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual. Etiqueta: DC.Description	N
Fuente	Secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. Etiqueta: DC.Source	N
Lenguaje	Lengua/s del contenido intelectual del recurso. Etiqueta: DC.Language	N
Relación	Es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos. Etiqueta: DC.Relation	N
Cobertura	Es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado (que ya lo encontramos en el elemento Fecha). Etiqueta: DC.Coverage	N
Autor o Creador	La persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales. Etiqueta: DC.Creator	S
Editor	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual	N

	Etiqueta: DC.Publisher	
Otros Colaboradores	Una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creador. (Por ejemplo: editor, ilustrador y traductor). Etiqueta: DC.Contributor	N
Derechos	Son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso. Etiqueta: DC.Rights	N
Fecha	Una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha. DC.Date	N
Tipo del Recurso	La categoría del recurso. Por ejemplo, página personal, romance, poema, diccionario, etc. Etiqueta: DC.Type	N
Formato	Es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso. Etiqueta: DC.Format	N
Identificador del Recurso	Secuencia de caracteres utilizados para identificar unívocamente un recurso. Ejemplos para recursos en línea pueden ser URLs i URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN ("International Standard Book Number"). Etiqueta: DC.Identifier	N

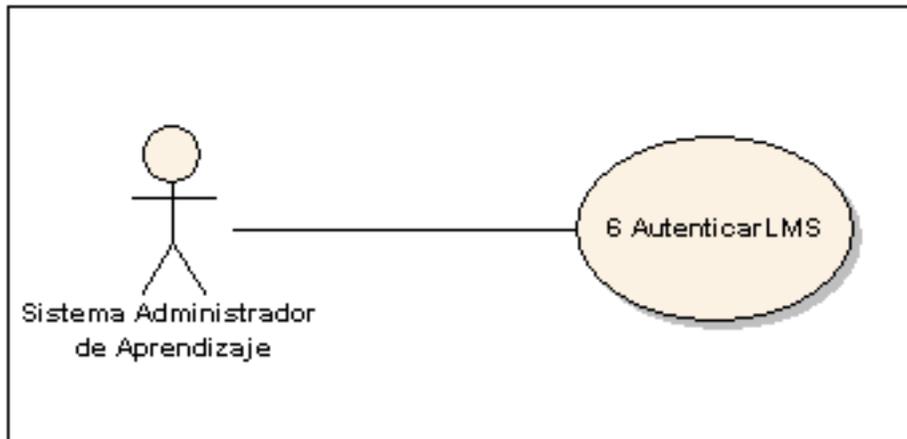
4.3.5.2 Flujo alternativo

4.3.5.3 Precondiciones

El usuario o sistema deberá estar dado de alta previamente por el administrador del repositorio.

4.3.5.4 Poscondiciones

4.3.6<CU AutenticarLMS 06>



4.3.6.1 Descripción breve del caso de uso

Se comprueba el certificado de seguridad con el cual se autenticaran el sistema administrador de aprendizaje ante el repositorio.

4.3.6.1.1 Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El sistema administrador de aprendizaje ingresa a la opción de autenticación ante el repositorio. (REQ01)	2	El repositorio comprueba el certificado de seguridad.	E1
		3	El repositorio queda configurado para proporcionar las futuras peticiones de consulta.	

ID	TIPO	ACCION
E1	El certificado de seguridad no es correcto.	El repositorio indica al sistema administrador de aprendizaje que su certificado de seguridad no es correcto.

REQ01		
DATO	DESCRIPCIÓN	REQUERIDO
Certificado de seguridad.	Cuando se utilizan certificados de seguridad, se guarda uno tanto en el cliente como en el repositorio para poder validar la solicitud.	S

4.3.6.1.2 Flujo alternativo

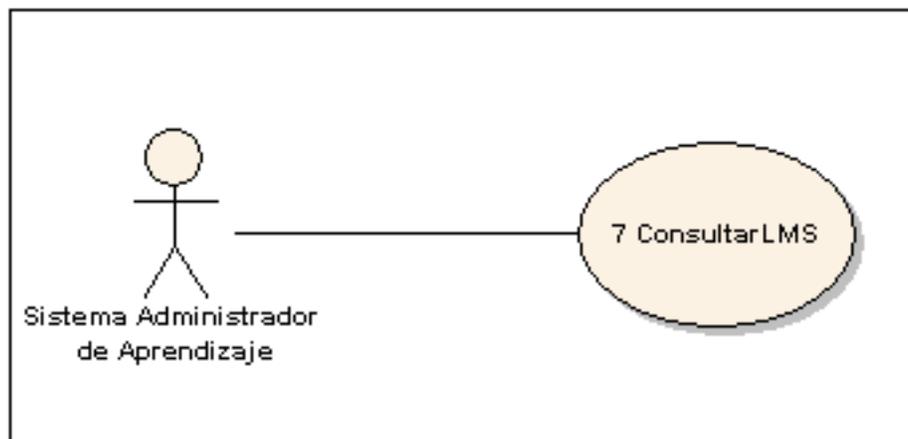
4.3.6.2 Precondiciones

Los certificados de seguridad del cliente y del repositorio deben estar creados.

4.3.6.3 Poscondiciones

El sistema administrador de aprendizaje podrá realizar consultas al repositorio.

4.3.7 <CU Consultar 02>



4.3.7.1 Descripción breve del caso de uso

El sistema administrador de aprendizaje enviará una consulta basada en los metadatos de Dublin Core con el fin de obtener los paquetes de contenido.

4.3.7.2 Flujo de eventos

4.3.7.2.1 Flujo básico

ACTOR		SISTEMA		
PASO	ACCIÓN	PASO	ACCIÓN	EXCEPCIÓN
1	El sistema administrador de aprendizaje envía los campos con los criterios de búsqueda (REQ1).		.	
		2	El sistema valida los datos	E1
		3	Se regresa el paquete(s) de contenido(s) que concuerde con la búsqueda solicitada.	E2

ID	TIPO	ACCION
E1	La cadena no tiene el formato correcto.	El repositorio regresa un mensaje de error informando el problema.
E2	Ningún paquete de contenido concuerda con los datos introducidos en la búsqueda.	El sistema regresa un mensaje informando al usuario que su búsqueda no arrojó ningún resultado.

REQ01		
DATO	DESCRIPCIÓN	REQUERIDO
Cadena de Búsqueda.	Los campos debe tener formato alfanumérico y debe tener la siguiente forma:	S

	<p>(String Titulo, String Clave, String Descripcion, String Fuente, String Lenguaje, String Relación, String Cobertura, String Autor, String Editor, String Colaboradores, String Derechos, Date Fecha, String TipoRecurso, String Formato, String IdentificadorRecurso)</p> <p>Los unicos parámetros obligatorios son:</p> <p>Título, Clave, Autor/Creador.</p>	
--	--	--

4.4 Prototipo de la interfaz del usuario

4.4.1 <Pantalla 1>

5. Justificación de los requerimientos de software

6. Información de apoyo

Apéndice B. Análisis y Diseño



Índice del Apéndice B

1.	Introducción	2
1.1	Propósito	2
1.2	Alcance	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
1.4	Referencias	3
2.	Representación de la Arquitectura	4
2.1	Metas de la arquitectura	4
2.2	Restricciones de la arquitectura	5
3.	Modelo del Análisis	6
4.	Clases del Análisis	7
4.1	Identificación de las clases	7
4.1.1	Clases del Modelo	7
4.1.2	Clases de la Vista	8
4.1.3	Clases de control	9
5.	Realización de Caso de Uso - Análisis	9
5.1	Diagramas de interacción	10
5.1.1	Diagrama Secuencia para el Caso de Uso Autenticar.	10
5.1.2	Diagrama Secuencia para el Caso de Uso Consultar Usuario	11
5.1.3	Diagrama Secuencia para el Caso de Uso de Agregar	12
5.1.4	Diagrama Secuencia para el Caso de Uso de AutenticarLMS	13
6.	Justificación de las decisiones del Análisis	13



Análisis

1. Introducción

En este documento se describen con mayor detalle las necesidades y especificaciones del cliente desde el punto de vista conceptual. En este documento se especifica la arquitectura del sistema, diagramas y las distintas clases necesarias para el desarrollo.

1.1 Propósito

El propósito del documento es ofrecer al equipo de trabajo, la información necesaria para poder comprender los detalles de los requerimientos planteados. El lenguaje utilizado en este documento es técnico y debe ser interpretado por el equipo de trabajo.

1.2 Alcance

El alcance del documento pretende cubrir con un detalle técnico los requerimientos expuestos en el documento de análisis de requerimiento.

1.3 Definiciones, Acrónimos y Abreviaturas

Término/Acrónimo/Abreviatura	Definición
ISO/IEC 29110-5-1	Norma Internacional para la Mejora de Procesos de Software.
COMPETISOFT	Modelo de Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos.
Objeto de Aprendizaje	Cualquier recurso digital que pueda ser reutilizado para apoyar la enseñanza.
Paquete de Despliegue	Guías y materiales para seguir correctamente y con buenas prácticas un proceso de algún modelo de desarrollo de software.
Paquete de Contenido	Archivo comprimido en el cual se guarda objetos de aprendizaje y metadatos asociados a estos.
SCORM	Es una colección de estándares y especificaciones basados en e-learning implementados a través de HTML y javascript.
E-learning	Concepto de educación a distancia en el que se integra el uso de las tecnologías de la información y otros elementos didácticos para la formación, capacitación y enseñanza de los usuarios o estudiantes en línea.
SOA	Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a la capa de procesos de negocio.
Servicios Web	Conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
Sistema de Administración de Aprendizaje	Programa empelado para distribuir, administrar y controlar el proceso de enseñanza.



SOAP	Este protocolo es un protocolo basado en XML para intercambiar información estructurada en un entorno descentralizado y distribuido. Los Servicios Web utilizan normalmente el protocolo SOAP sólo para las comunicaciones.
XML	Lenguaje de Marcado Extensivo (por sus siglas en inglés). Es un metalenguaje de propósito general para la creación de lenguajes para diferentes necesidades. Es un lenguaje de marcas porque al igual que HTML tiene una estructura que permite la creación de elementos para definir los datos y estructura del nuevo lenguaje.
Model View Controles MVC	Modelo, Vista, Controlador, patrón de diseño que guía la arquitectura del sistema.
Petición	Acción que se genera desde el navegador (Firefox, Opera, Internet Explorer) cuando el cliente da click en un componente del sistema.
Respuesta	Página resultante que contiene la información solicitada desde el navegador del cliente.
Cliente	Persona que usa el sistema desde una computadora con acceso a Internet.

1.4 Referencias

COMPETISOFT, <http://alarcos.inf-cr.uclm.es/Competisoft/>, 15/01/2009

ISO/IEC 29110-5-1, ISO/IEC CD TR 29110-5-1 Software Engineering - Lifecycle Profiles for Very Small Enterprises (VSE) -- Part 5-1: Management and Engineering Guide -- Basic Profile, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51160, 15/01/2009

Learning Object, Learning Technology Standards Committee, Institute of Electrical and Electronics Engineers IEEE, 2002, p.45

IMS Content Packaging, <http://www.imsproject.org/content/packaging/>, 12/03/2009

SCORM, <http://www.adlnet.gov/scorm/>, 12/03/2009

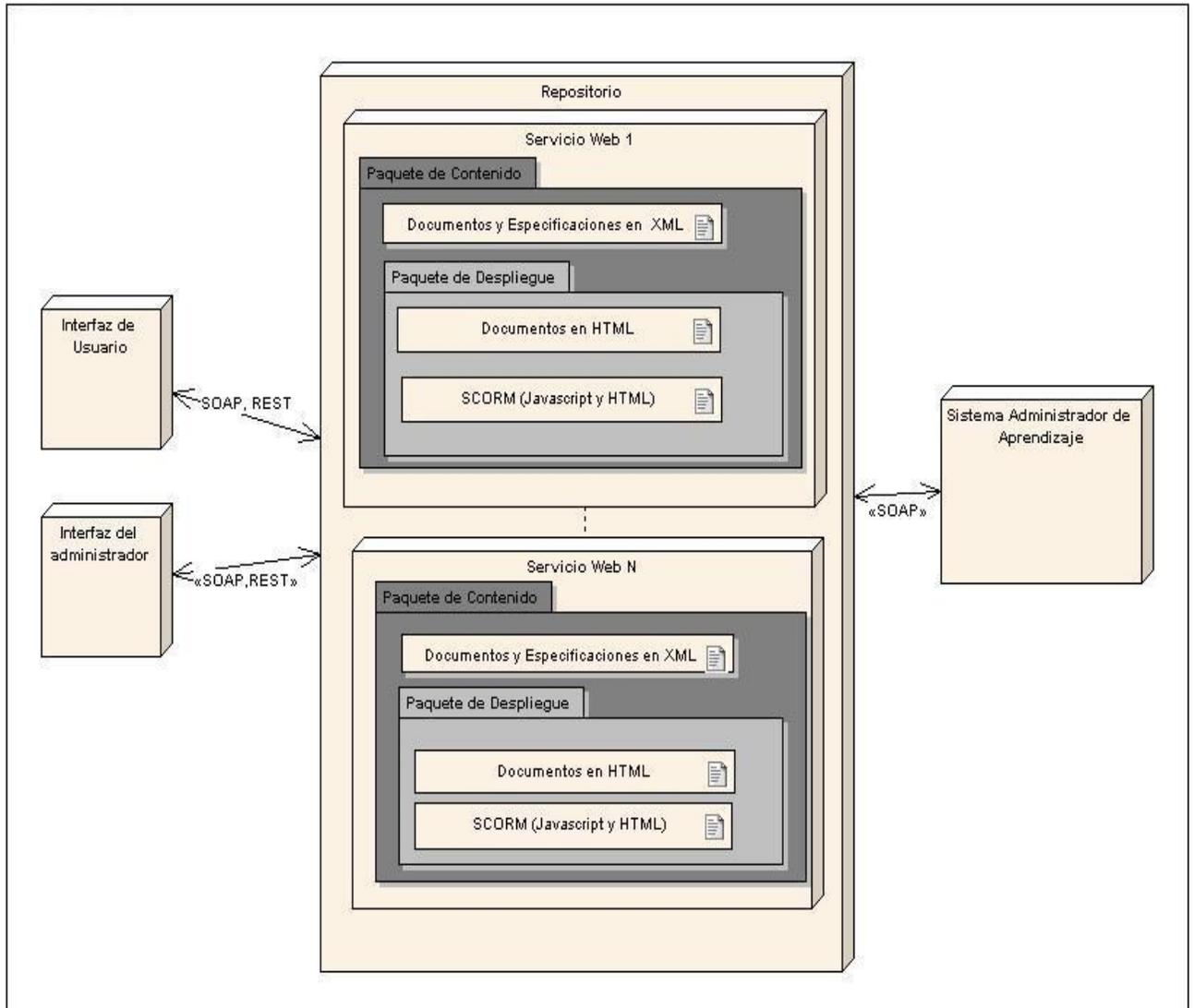
Learning Management Systems, http://en.wikipedia.org/wiki/Learning_management_system, 12/03/2009

Model View Controller MVC, http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, 12/03/2009

Newcomer, Eric; Lomow, Greg (2005). *Understanding SOA with Web Services*. Addison Wesley. ISBN 0-321-18086-0.



2. Representación de la Arquitectura



2.1 Metas de la arquitectura

El repositorio será construido bajo una Arquitectura Orientada a Servicios (SOA) implementando Servicios Web (Web Services). Los Servicios Web servirán para proporcionar servicios para la manipulación y almacenamiento de los paquetes de contenido, los cuales permitirán la interoperabilidad entre los diferentes LMS, automatizando procesos y facilitando la publicación de la información en diferentes fuentes o aplicaciones.

Existen varias razones por las que se debe seguir SOA.

- Reutilización. Ya que los mismos servicios web pueden ser reutilizados para crear nuevos.
- Interoperabilidad. Esto con el fin de que los servicios web ofrecidos puedan ser utilizados por distintos sistemas sin importar el lenguaje en el que fueron creados.
- Escalabilidad. Como los servicios SOA están débilmente acoplados, las aplicaciones que usan estos servicios escalan fácilmente. Esto debido a que existe muy poca dependencia entre las aplicaciones clientes y los servicios que usan.



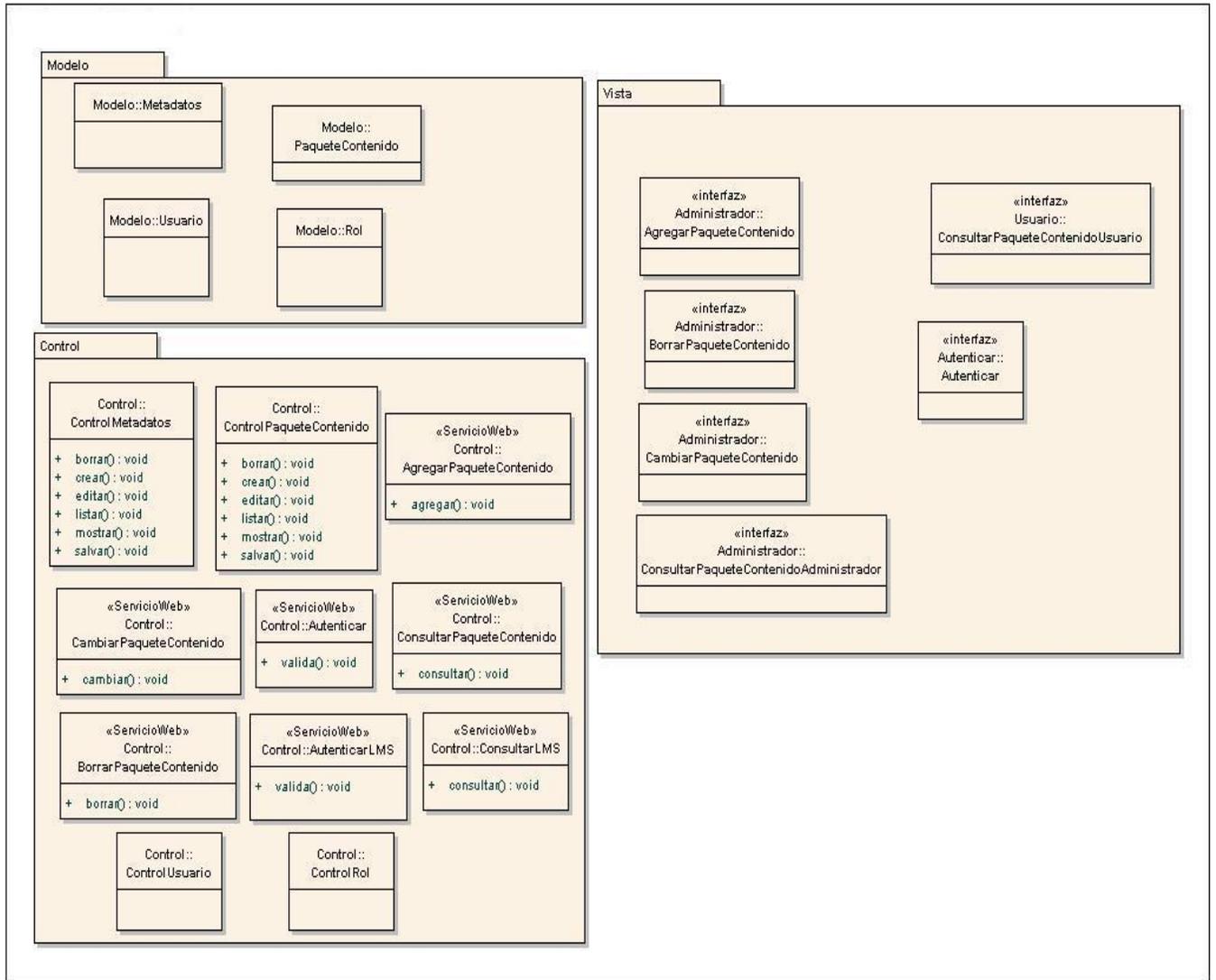
- Flexibilidad. Es otra de las características que proporciona el acoplamiento débil entre los servicios. Cualquier cambio en la implementación de uno de ellos no afectaría al resto siempre que se mantenga la interfaz.
- Herramientas para la creación de servicios web y SOA existentes.

2.2 Restricciones de la arquitectura

El sistema deberá ser capaz de ofrecer los servicios web tanto a sistemas administradores de aprendizaje, usuarios y administradores, a través de SOA.



3. Modelo del Análisis

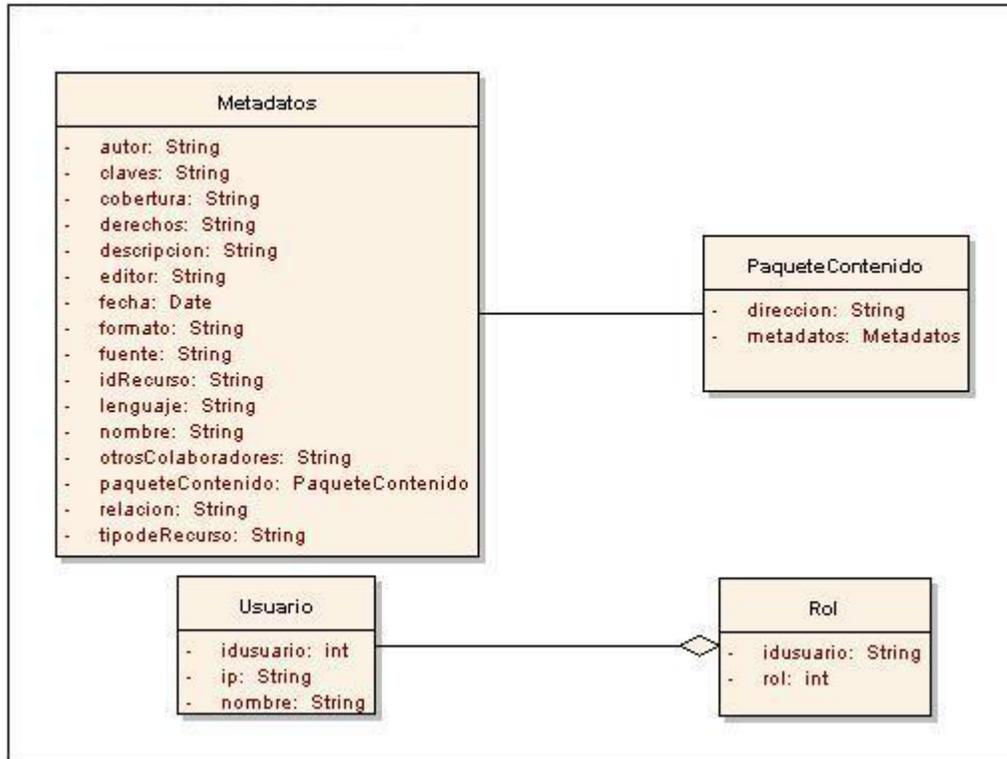




4. Clases del Análisis

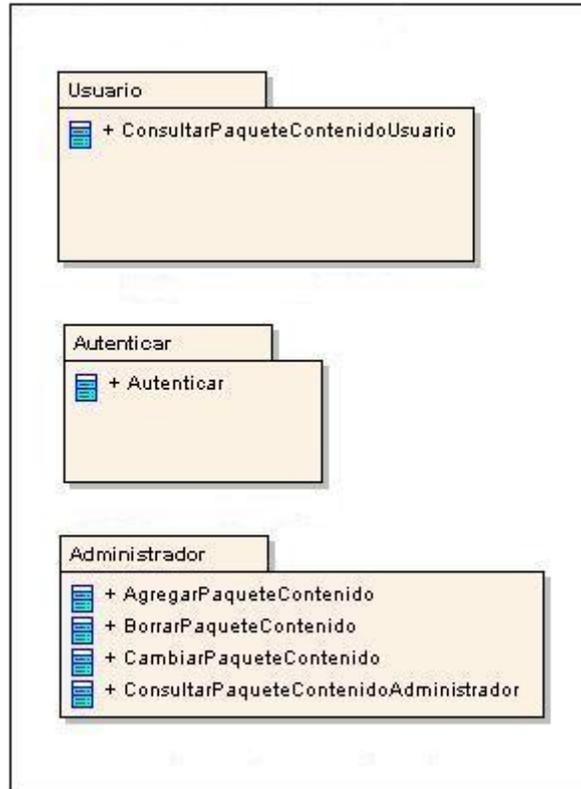
4.1 Identificación de las clases

4.1.1 Clases del Modelo



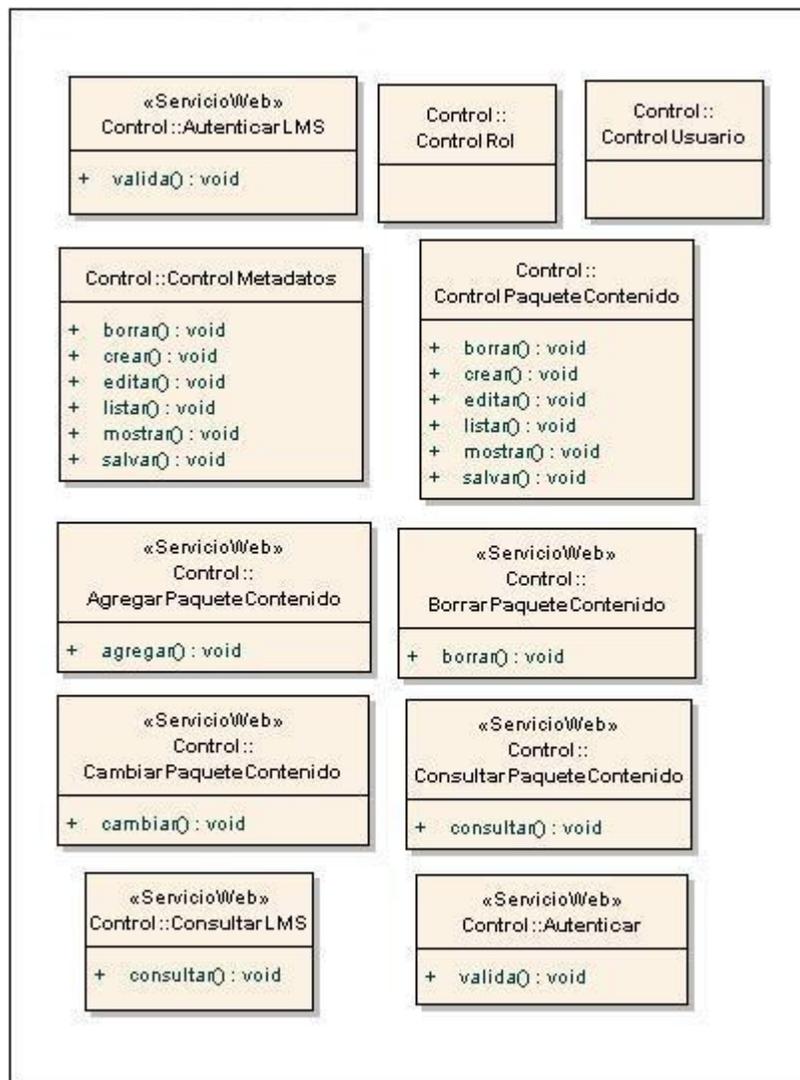


4.1.2 Clases de la Vista





4.1.3 Clases de control



5. Realización de Caso de Uso - Análisis

En la siguiente tabla se especifican los casos de usos cuyo diagrama de secuencia muestra el comportamiento relevante del sistema.

Número	Nombre	Detalle
1	Autenticar	<i>Si</i>
2	Consultar	<i>Si</i>
3	Agregar	<i>Si</i>

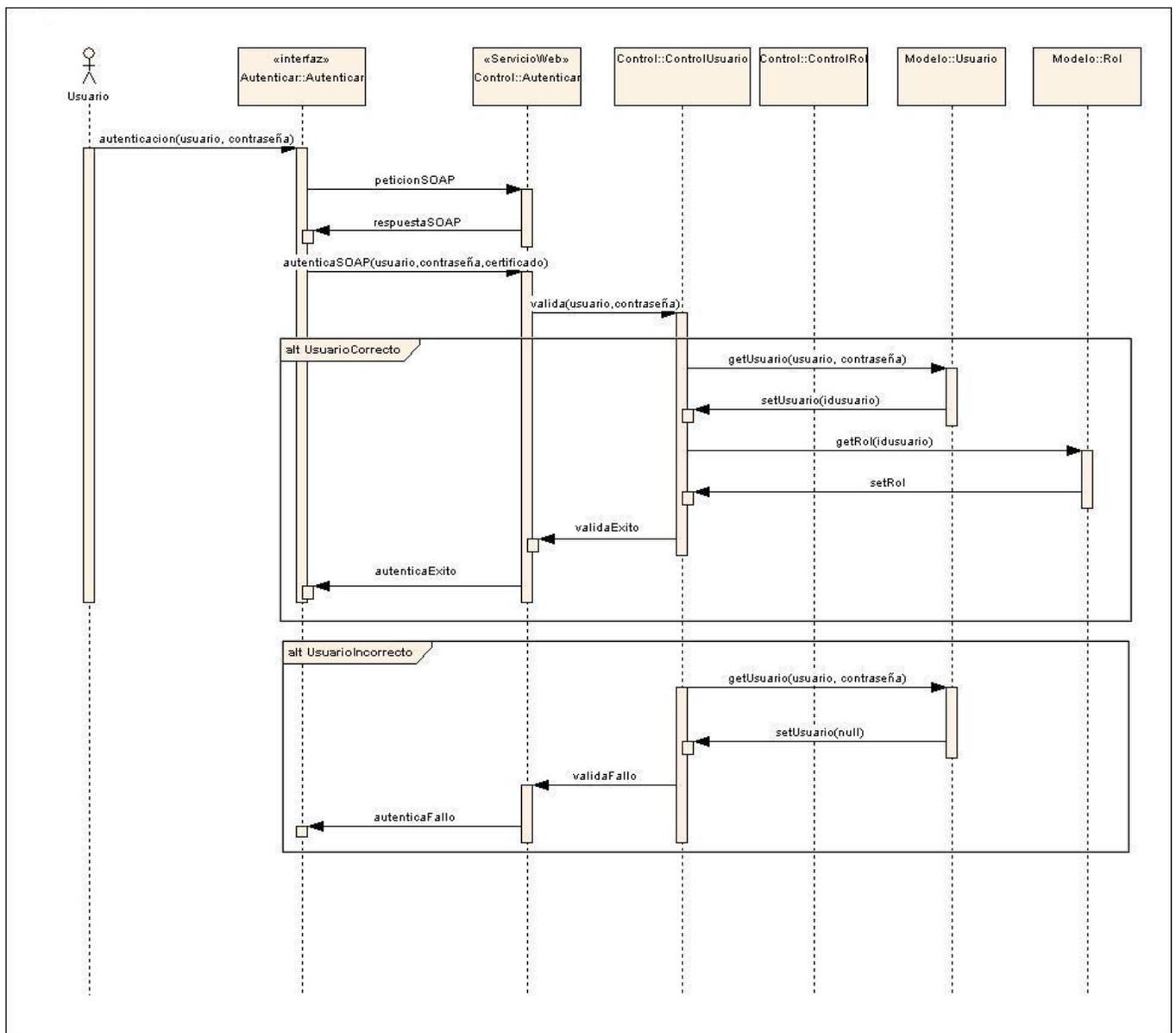


4	Borrar	No
5	Cambiar	No
6	AutenticarLMS	<i>Si</i>
7	ConsultarLMS	No

Figura 5.1. Casos de uso que serán especificados mas adelante.

5.1 Diagramas de interacción

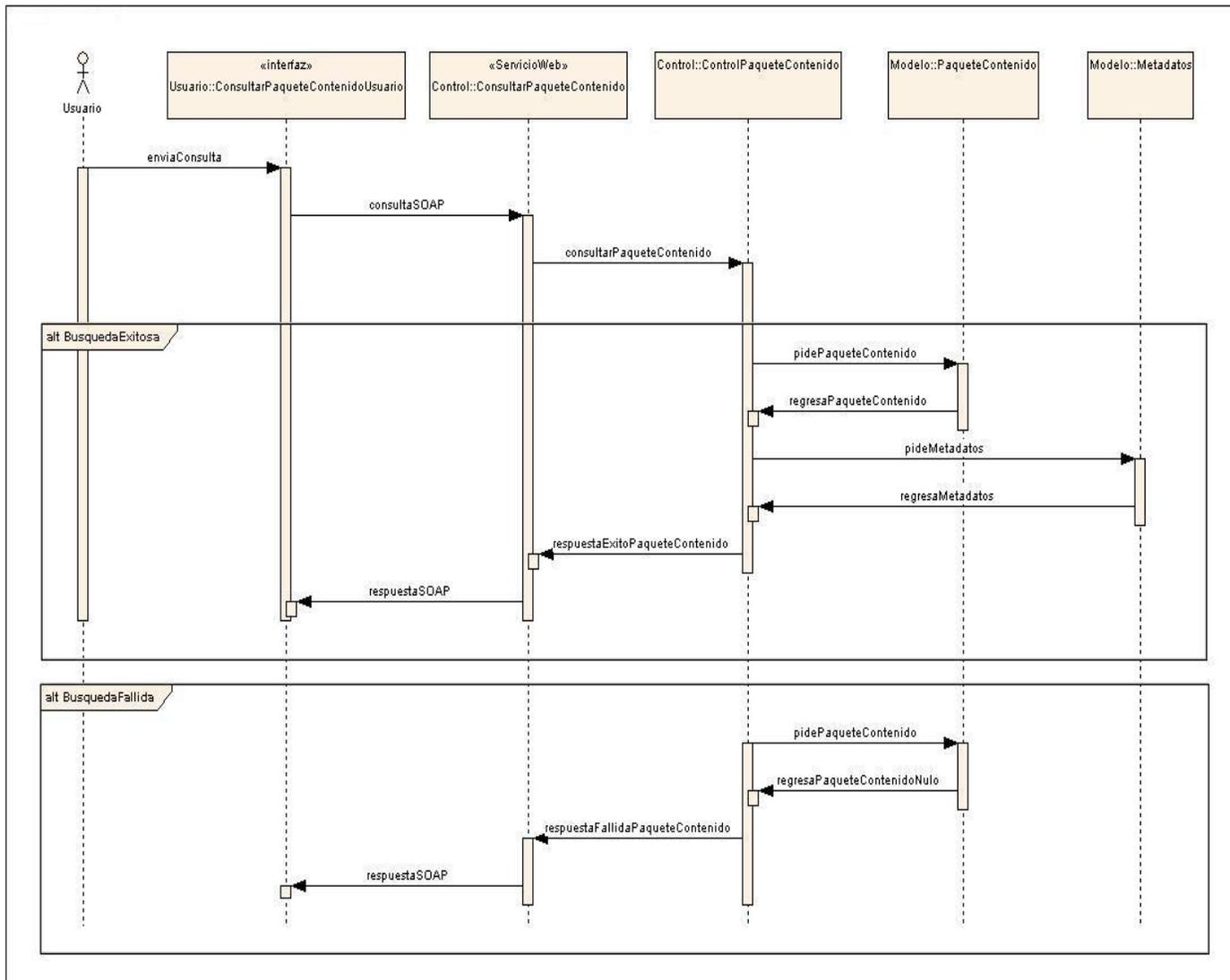
5.1.1 Diagrama Secuencia para el Caso de Uso Autenticar.





En este caso de uso se tiene una interfaz llamada **Autenticar** la cual será capaz de comunicarse con el servicio web **Autenticar** mediante el uso de SOAP y SAML (para el manejo de los certificados y seguridad). La forma de autenticación se realiza cuando la interfaz de autenticación realiza una petición del archivo de descripción del servicio web (**peticionSOAP**), el servicio web responde y envía el archivo WSDL solicitado (**respuestaSOAP**), el cual contiene todos los datos necesarios para que se pueda procesar la petición SOAP (seguridad, certificados, usuario, contraseña, operaciones), una vez hecho esto la interfaz es capaz de mandar la petición SOAP con todos los datos necesarios para autenticarse ante el repositorio (mandar el usuario, contraseña y certificados de autenticidad).

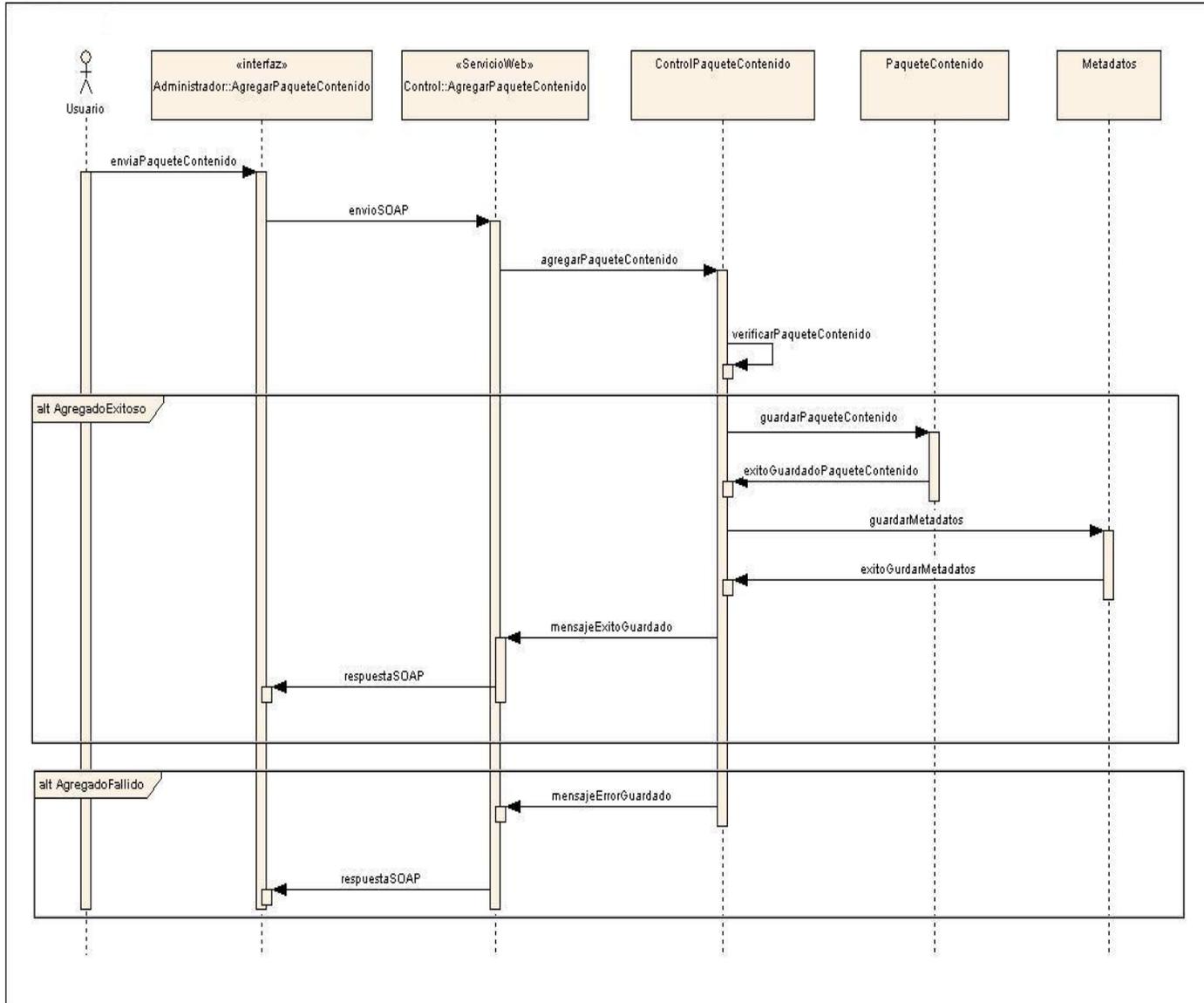
5.1.2 Diagrama Secuencia para el Caso de Uso Consultar Usuario



En este diagrama se toma en consideración que el usuario ya se autentico ante el repositorio. El caso de uso para las consultas del administrador es análogo.



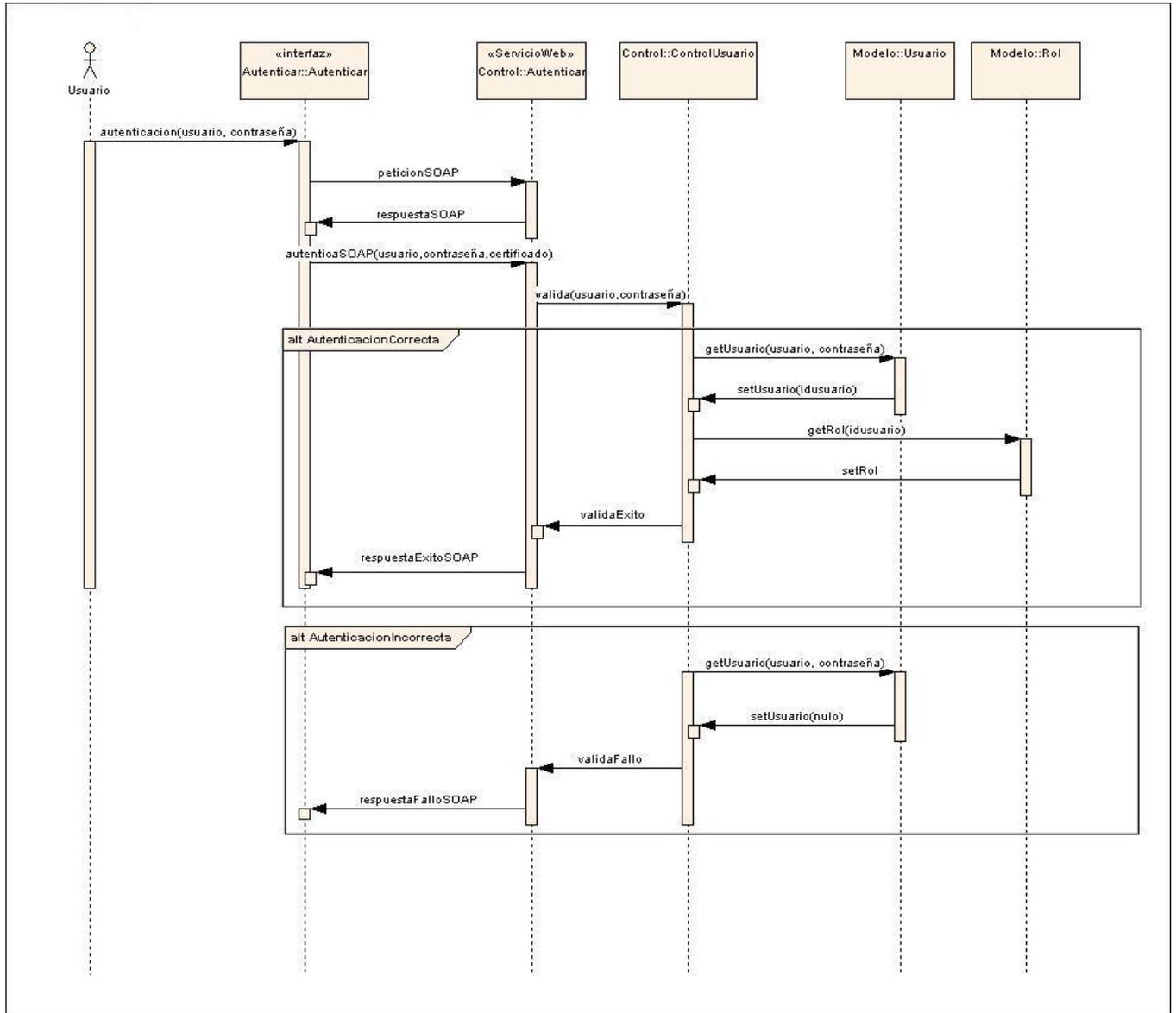
5.1.3 Diagrama Secuencia para el Caso de Uso de Agregar



En este diagrama se toma en consideración que el usuario ya se autentico ante el repositorio.



5.1.4 Diagrama Secuencia para el Caso de Uso de AutenticarLMS



6. Justificación de las decisiones del Análisis

Se optará utilizar WSIT/Metro para la creación de los servicios web debido a las facilidades en cuanto a la implementación de los servicios, además de la extensa documentación que puede ser encontrada en internet.

En cuanto a la seguridad de los servicios web se revisaron 3 propuestas:

- Seguridad de los servicios web mediante WSIT Style Security.
- Seguridad de los servicios web mediante XWSS 2.0 Style Security Configuration.



- Seguridad de los servicios web mediante JAXWS Handler's y XWSS 2.0

De estas tres opciones se optó por utilizar la seguridad en servicios web mediante WSIT Style Security debido a que es la propuesta más nueva de Sun Microsystems en cuanto a interoperabilidad además de que mejora el desempeño respecto a los otros dos modelos de seguridad.

Apéndice C. Construcción



Apéndice C. Construcción

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	2
3.2	A modificar	4
4.	Justificación de las decisiones de la Construcción	4
5.	Información de apoyo	5



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

Este documento está dirigido a los desarrolladores de software facilitando el proceso de implementación describiendo las actividades que el repositorio debe desarrollar.

1.2 Alcance

Este documento muestra los componentes a implementar en la fase de construcción.

1.3 Definiciones, Acrónimos y Abreviaturas

En la siguiente tabla se proveen definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento.

Término	Definición
Componente	Un elemento de software que integra una serie de funcionalidades (métodos).

1.4 Referencias

2. Descripción general

En este documento se explican los componentes a implementar,, posibles cambios realizados, justificaciones en las decisiones de la construcción e información de apoyo

3. Componentes

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.

3.1 A implementar

A continuación se muestra el listado de los componentes necesarios en la construcción del sistema.

Componente Autenticar.

- Interfaz Autenticar



- ServicioWeb Autenticar
- Clase ControlAutenticar
- Clase ControlUsuario
- Clase ControlRol
- Clase ModeloUsuario
- Clase ModeloRol

Componente ConsultarPaqueteContenido.

- Interfaz ConsultarPaqueteContenido
- ServicioWeb ConsultarPaqueteContenido
- Clase ConsultarPaqueteContenido
- Clase ControlPaqueteContenido
- Clase ModeloPaqueteContenido
- Clase ModeloMetadatos

Componente AgregarPaqueteContenido.

- Interfaz AgregarPaqueteContenido
- ServicioWeb AgregarPaqueteContenido
- Clase AgregarPaqueteContenido
- Clase ControlPaqueteContenido
- Clase ModeloPaqueteContenido
- Clase ModeloMetadatos

Componente CambiarPaqueteContenido.

- Interfaz CambiarPaqueteContenido
- ServicioWeb CambiarPaqueteContenido
- Clase CambiarPaqueteContenido
- Clase ControlPaqueteContenido
- Clase ModeloPaqueteContenido
- Clase ModeloMetadatos

Componente BorrarPaqueteContenido.

- Interfaz BorrarPaqueteContenido
- ServicioWeb BorrarPaqueteContenido
- Clase BorrarPaqueteContenido
- Clase ControlPaqueteContenido
- Clase ModeloPaqueteContenido
- Clase ModeloMetadatos

Componente ConsultarUsuario.

- Interfaz ConsultarUsuario
- ServicioWeb Usuario
- Clase Usuario
- Clase ControlPaqueteContenido
- Clase ModeloPaqueteContenido
- Clase ModeloMetadatos



Componente AgregarUsuario.

- Interfaz AgregarUsuario
- ServicioWeb AgregarUsuario
- Clase AgregarUsuario
- Clase ControlUsuario
- Clase ControlRol
- Clase ModeloUsuario
- Clase ModeloRol

Componente CambiarUsuario.

- Interfaz CambiarUsuario
- ServicioWeb CambiarUsuario
- Clase CambiarUsuario
- Clase ControlUsuario
- Clase ControlRol
- Clase ModeloUsuario
- Clase ModeloRol

Componente BorrarUsuario.

- Interfaz BorrarUsuario
- ServicioWeb BorrarUsuario
- Clase BorrarUsuario
- Clase ControlUsuario
- Clase ControlRol
- Clase ModeloUsuario
- Clase ModeloRol

3.2 A modificar

4. Justificación de las decisiones de la Construcción

Para la construcción de estos componentes y servicios se utilizaron las siguientes tecnologías:

- J2EE. Como framework de programación para desarrollar y ejecutar las funcionalidades que requieren los servicios web.
- WSIT. Utilizado en la creación de la arquitectura SOA e implementación de los servicios web; debido a las facilidades en cuanto a la implementación de los servicios web, además de la extensa documentación que puede ser encontrada en libros e internet.



- TopLink. Como manejador de persistencia de datos (acceso y manejo de bases de datos).
- Java Server Faces. Como framework para la creación del Sistema Web encargado de administrar el repositorio.
- Glassfish. Como servidor de aplicaciones.

5. Información de apoyo

Apéndice D. Pruebas



Apéndice D. Pruebas

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	6
5.1	Tipos de pruebas	6
5.1.1	Pruebas de integridad de la Base de Datos y los Datos	7
5.1.2	Pruebas de funcionalidad	7
5.1.3	Pruebas de negocio	7
5.1.4	Pruebas de interfaz de usuario	7
5.1.5	Pruebas de desempeño	7
5.1.6	Pruebas de carga de trabajo	7
5.1.7	Pruebas de stress	7
5.1.8	Pruebas de volumen	7
5.1.9	Pruebas de seguridad y control de acceso	7
5.1.10	Pruebas de tolerancia y recuperación a fallos	7
5.1.11	Pruebas de configuración	7
5.1.12	Pruebas de Instalación	7
5.1.13	Pruebas de Aceptación	7
6.	Herramientas	7
7.	Recursos	7
8.	Justificación de las decisiones de las Pruebas	7
9.	Información de apoyo	7



Pruebas

1. Introducción

Este 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

Este documento está dirigido a los desarrolladores de software facilitando el proceso de control de pruebas describiendo las actividades que se desean realizar en el repositorio, especificando los datos de entrada, salidas esperadas, salidas obtenidas, errores encontrados y posibles soluciones.

1.2 Background

La primera parte de las pruebas se realizará a través de una serie de tareas que deben ser realizadas tanto por el desarrollador del sistema como por un grupo de usuarios con la finalidad de encontrar posibles errores (pueden ser de diseño o programación) y soluciones a estos.

La segunda prueba consistirá en la utilización de los servicios web del repositorio mediante la integración con un sistema tercero.

Las pruebas que se realizarán son las siguientes:

- Uso del servicio web para la consulta básica de los paquetes de despliegue.
- Uso del servicio web para la consulta avanzada de los paquetes de despliegue.

1.3 Alcance

Las pruebas que se realizarán son pruebas de integridad y de desempeño para determinar la funcionalidad y mejora del repositorio. Las pruebas que se realizarán son las siguientes:

- Búsqueda por consulta sencilla desde la pantalla principal
- Búsqueda por consulta sencilla desde la pantalla de administración del los paquetes de despliegue
- Búsqueda avanzada
- Creación de un paquete de despliegue
- Modificación de un paquete de despliegue
- Borrado de los paquetes de despliegue
- Detallado de los paquetes de despliegue

Cada una de estas pruebas tendrá distintos datos de entrada.

1.4 Definiciones, Acrónimos y Abreviaturas

En la siguiente tabla se proveen definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento.



Término	Definición
Pruebas de Integridad	Pruebas necesarias para encontrar defectos de software o de diseño.
Pruebas de Desempeño	Pruebas necesarias para solucionar los requerimientos no funcionales. Cabe mencionar que estas pruebas deben ser realizadas después de realizar las pruebas de integridad.

1.5 Referencias

2. Descripción general

3. Requerimientos para las pruebas

4. Reporte de pruebas del sistema

4.1 Casos de prueba

4.1.1 <ABCC Paquetes de despliegue - Búsqueda por consulta sencilla desde la pantalla principal>

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Búsqueda sencilla	No se especifica una entrada	Lista todos los paquetes de despliegue encontrados en el repositorio	La esperada
Búsqueda sencilla	cadena = "Deployment Package"	Muestra los paquetes de despliegue encontrados en el repositorio que concuerden con el criterio de búsqueda	La esperada
Casos de prueba con datos inválidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos

4.1.2 < ABCC Paquetes de despliegue - Búsqueda por consulta sencilla desde la pantalla de administración de los paquetes de despliegue >

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Búsqueda Sencilla	No se especifica una entrada	Lista todos los paquetes de despliegue encontrados en el repositorio	La esperada
Búsqueda Sencilla	cadena = "Deployment Package"	Muestra los paquetes de despliegue encontrados en el repositorio que concuerden con	La esperada



		el criterio de búsqueda	
Casos de prueba con datos inválidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos

4.1.3 < ABCC Paquetes de despliegue - Búsqueda por consulta avanzada >

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Búsqueda Avanzada	No se especifica una entrada	Lista todos los paquetes de despliegue encontrados en el repositorio	La esperada
Búsqueda Avanzada	Idapquetedespliegue = '2' ó título = 'título'	Muestra los paquetes de despliegue encontrados en el repositorio que concuerden con el criterio de búsqueda	La esperada
Búsqueda Avanzada	El usuario especifica los campos a buscar	Muestra los paquetes de despliegue encontrados en el repositorio que concuerden con el criterio de búsqueda	La esperada
Casos de prueba con datos inválidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos

4.1.4 <ABCC Paquetes de Despliegue – Agregar un paquete de despliegue >

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Creación de un paquete de despliegue	Se especifican los campos de: Título: Tema: Autor: Descripción: Fuente: Lenguaje: Relación: Cobertura: Editor: Colaboradores: Derechos: Fecha: Tipo: Formato: Archivo:	Se crea con éxito el paquete de despliegue	La esperada
Creación de un paquete de despliegue	Se especifican los campos de: Descripción: Fuente: Lenguaje: Relación: Cobertura: Editor: Colaboradores: Derechos: Fecha: Tipo: Formato: Archivo:	Muestra un mensaje de error especificando que es necesario llenar los campos de titulo, tema, autor.	La esperada
Casos de prueba con datos inválidos			



Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Creación de un paquete de despliegue	Se especifican los campos de: Título: Tema: Autor: Descripción; con un tamaño de más de 500 caracteres	El sistema muestra un mensaje de error.	El sistema falla cuando la cadena en alguno de los campos de búsqueda es muy larga

4.1.5 <ABCC Paquetes de Despliegue – Modificar un paquete de despliegue>

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Modificación de un paquete de despliegue	Se especifican los campos de: Título: Tema: Autor: Descripción: Fuente: Lenguaje: Relación: Cobertura: Editor: Colaboradores: Derechos: Fecha: Tipo: Formato: Archivo:	Se crea con éxito el paquete de despliegue	La esperada
Modificación de un paquete de despliegue	Se especifican los campos de: Descripción: Fuente: Lenguaje: Relación: Cobertura: Editor: Colaboradores: Derechos: Fecha: Tipo: Formato: Archivo:	Muestra un mensaje de error especificando que es necesario llenar los campos de titulo, tema, autor.	La esperada
Casos de prueba con datos inválidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Modificación de un paquete de despliegue	Se especifican los campos de: Título: Tema: Autor: Descripción; con un tamaño de más de 500 caracteres	El sistema muestra un mensaje de error.	El sistema falla cuando la cadena en alguno de los campos de búsqueda es muy larga

4.1.6 <ABCC Paquetes de Despliegue – Borrar un paquete de despliegue>

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Borrar un paquete de despliegue	No se especifica una entrada	Se borra el paquete de despliegue seleccionado	La esperada
Casos de prueba con datos inválidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos



4.1.7 <ABCC Paquetes de Despliegue – Detallar un paquete de despliegue>

Casos de prueba con datos válidos			
Caso de Uso	Entradas	Resultados Esperados	Resultados Obtenidos
Detallado de un paquete de despliegue	No se especifica una entrada	Se muestra un resumen con el detallado de los paquetes de despliegue	La esperada
Casos de prueba con datos inválidos			
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.]

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

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]

Apéndice E. Deployment Package Software Requirements

Analysis Basic Profile

Deployment Package

Software Requirements Analysis

Basic Profile

Notes:

This document is the intellectual propriety of its author’s organization. However, information contained in this document is free of use. The distribution of all or parts of this document is authorized for non commercial use as long as the following legal notice is mentioned:

© Centre d’Excellence en Technologies de l’Information et de la Communication and École de Technologie Supérieure

Commercial use of this document is strictly forbidden. This document is distributed in order to enhance exchange of technical and scientific information.

This material is furnished on an “as-is” basis. The author(s) make(s) no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material.

The processes described in this Deployment Package are not intended to preclude or discourage the use of additional processes that Very Small Enterprises may find useful.

Authors	S. ALEXANDRE, – Centre d’Excellence en Technologies de l’Information et de la Communication (CETIC), (Belgium) C. Y. LAPORTE, École de Technologie Supérieure (ETS), (Canada) Ricardo Cruz Mendoza – Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México. Hanna Oktaba – Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México.
Editors	C. Y. LAPORTE, École de Technologie Supérieure ANA VAZQUEZ – 5th level, (México)

Creation date	7 August 2007
Last update	24 March 2010
Status	Draft
Version	1.2

Version 1.2

Versions

Date	Version	Auteur	Modification
07/08/2007	0.1	S. ALEXANDRE	Document creation
20/08/2007	0.2	C.Y. LAPORTE	Comments on document structure
1/10/2007	0.3	S. ALEXANDRE	Implementation of comment and finalization of V1.0
1/10/2007	0.4	S. ALEXANDRE	Comments after review
8/10/2007	0.5	S. ALEXANDRE	Update and completion of section 3.1
14/10/2007	0.6	S. ALEXANDRE – C.Y. LAPORTE	Update and revision of section 3.1
19/10/2007	0.7	S. ALEXANDRE	Update of section 3.1 and 3.2
29/10/2007	0.8	S. ALEXANDRE	Replacement of 'practice' by 'activity' to comply with OGF SPEM terminology.
2/11/2007	0.9	S. ALEXANDRE	Alignment of the content with ISO12207:2008.
27/11/2007	0.10	S. ALEXANDRE	Update of graphical representation of steps.
1/12/2007	1.0	S. ALEXANDRE	Final version – ready for final review
21/01/2008	1.1	S. ALEXANDRE	Update of coverage matrix
7/07/2009	1.2	C.Y. LAPORTE	Modification to the copyright clause definitions and acronyms, and implementation of new DP template.

Abbreviations/Acronyms

Abre./Acro.	Definition
DP	Deployment Package - a set of artefacts developed to facilitate the implementation of a set of practices, of the selected framework, in a Very Small Entity.
VSE	Very Small Entity – an enterprise, organization, department or project having up to 25 people.
VSEs	Very Small Entities

Table of Contents

1. Technical Description	5
<i>Purpose of this document.....</i>	5
<i>Why Requirements Management is Important ?</i>	5
Main Causes of Success.....	6
2. Definitions.....	8
Generic Terms	8
Specific Terms	8
3. Relationships with ISO/IEC 29110.....	10
4. Description of Processes, Activities, Tasks, Steps, Roles and Products .	12
Tasks	12
Requirements identification.....	12
Requirements refinement and analysis	13
Requirements verification & validation.....	14
Requirements change management	15
Roles.....	17
Artefacts.....	17
5. Template	18
6. Example of Lifecycle.....	23
Example 1 of Requirement Practices Lifecycle	23
Example 2 of Requirement Practices Lifecycle	24
7. Checklist.....	25
Requirement checklist	25
8. Tool.....	26
Traceability Tool	26
9. References to Other Standards and Models	28
ISO 9001 Reference Matrix	28
ISO/IEC 12207 Reference Matrix.....	29
CMMI Reference Matrix	29
10. References	30
11. Evaluation Form	31

1. Technical Description

Purpose of this document

This Deployment Package (DP) supports the Basic Profile as defined in ISO/IEC 29110 Part 5-1: Management and Engineering Guide. A DP is a set of artefacts developed to facilitate the implementation of a set of practices in a Very Small Entity (VSE). A DP is not a process reference model (i.e. it is not prescriptive). The elements of a typical DP are: description of processes, activities, tasks, roles and products, template, checklist, example, reference and reference to standards and models, and tools.

In IT projects, it is critical to define as unambiguously as possible the customer requirements to ensure a common comprehension of requirements between the stakeholders, and to guarantee that requirements evolution is handled as part of the project.

Requirements analysis process includes the production and the maintenance of Software Requirements Specifications on the basis of customer demands and changes in these demands. Software Requirements Specifications will then constitute the basis for cost estimate, planning, implementation and tracking of activities throughout the project.

Requirements Management is one of the principal parameters for process stabilization and successful repeatability.

The content of this document is entirely *informative*.

This document has been produced by CETIC (Centre of Excellence in Information and Communication Technologies – www.cetic.be), CRPHT (Public Research Centre Henri Tudor's – www.tudor.lu) and ETS (Ecole de Technologie Supérieure - www.etsmtl.ca) beyond their official participation to ISO JTC1/SC7/WG24.

Why Requirements Management is Important ?

Several studies clearly underlined the importance of requirement management in software engineering. Among these studies, the Chaos Report published by the Standish Group (www.standishgroup.com/) from 1994 the Standish Group analyzed thousands of IT projects all over the world.

The Standish Group categorized IT projects into three resolution categories:

- *Successful*: The project is completed on time and on budget, with all features and functions originally specified.
- *Challenged*: The project is completed and operational, but over-budget, over the time estimate, and with fewer features and functions than initially specified.
- *Failed*: The project is cancelled before completion or is never implemented.

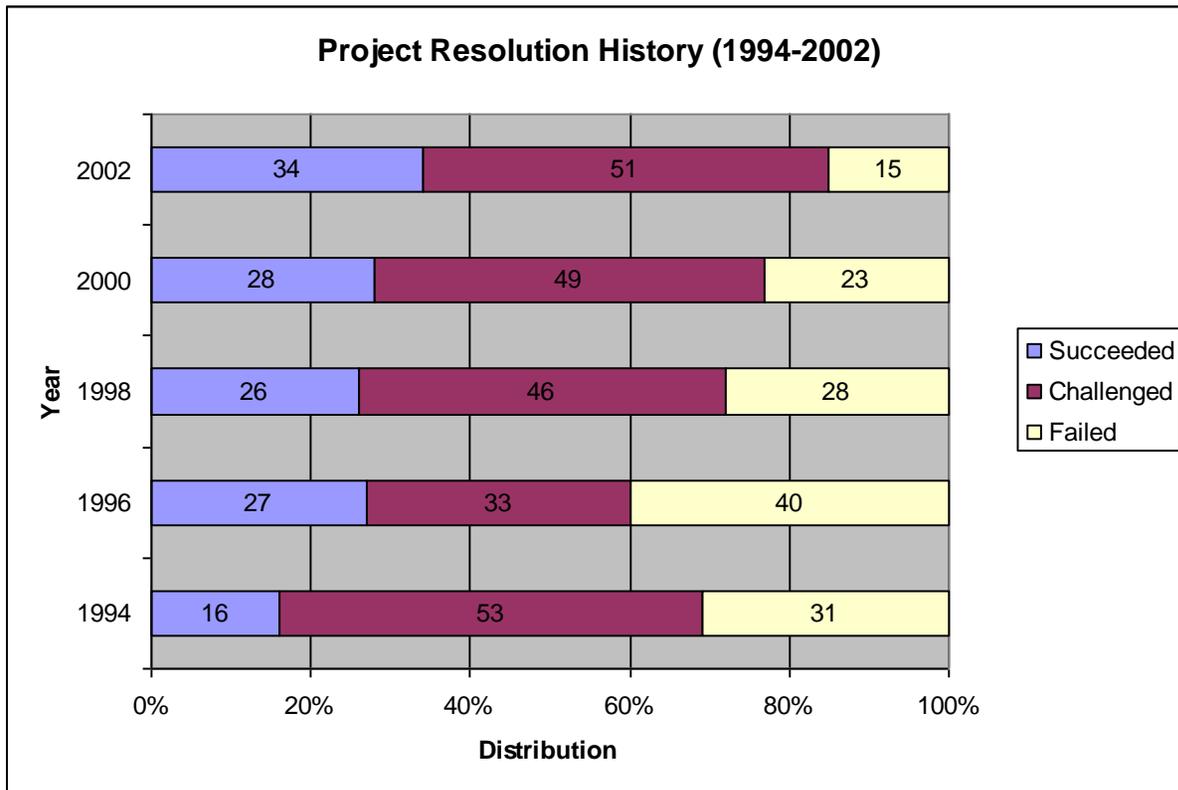


Figure 1 Project resolution history

Main Causes of Success

According to the Standish Group, the main causes of **SUCCESS** are:

- *User Involvement*
- Executive Support
- Clear Business Objectives
- Experienced Project Manager
- Small Milestones
- *Firms Basic Requirements*

Standish Group experts underlined the importance of user involvement and the good management and analysis of their requirements.

Figure 2 representing data from a real company¹ shows that close to 50% of the defects are produced during the requirements phase.

¹ Selby, P., Selby, R.W., Measurement-Driven Systems Engineering Using Six Sigma Techniques to Improve Software Defect Detection, Proceedings of 17th International Symposium, INCOSE, June 2007, San Diego.

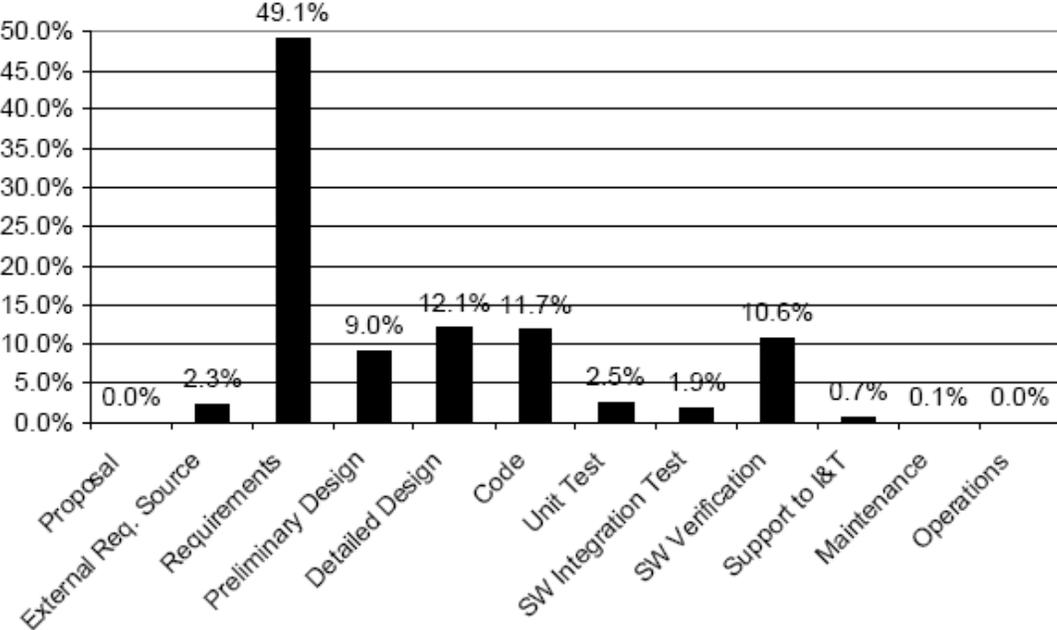


Figure 2 Origins of software defects

2. Definitions

In this section, the reader will find two sets of definitions. The first set defines the terms used in all Deployment Packages, i.e. generic terms. The second set of terms used in this Deployment package, i.e. specific terms.

Generic Terms

Process: set of interrelated or interacting activities which transform inputs into outputs [ISO/IEC 12207].

Activity: a set of cohesive tasks of a process [ISO/IEC 12207].

Task: required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process [ISO/IEC 12207].

Sub-Task: When a task is complex, it is divided into sub-tasks.

Step: In a deployment package, a task is decomposed in a sequence of steps.

Role: a defined function to be performed by a project team member, such as testing, filing, inspecting, coding. [ISO/IEC 24765]

Product: piece of information or deliverable that can be produced (not mandatory) by one or several tasks. (e. g. *design document, source code*).

Artefact: information, which is not listed in ISO/IEC 29110 Part 5, but can help a VSE during the execution of a project.

Specific Terms

Requirement: 1. a statement that identifies what a product or process must accomplish to produce required behaviour and/or results. *IEEE 1220-2005 IEEE Standard for the Application and Management of the Systems Engineering Process*. 3.1.16. 2. a system or software requirement that specifies a function that a system/software system or system/software component must be capable of performing. *ISO/IEC 24765, Systems and Software Engineering Vocabulary*. 3. a requirement that specifies a function that a system or system component must be able to perform. [ISO/IEC24765]

Requirements analysis: The process of studying user needs to arrive at a definition of system, hardware, or software requirements. [ISO/IEC 24765]

Requirements document: a document containing any combination of recommendations, requirements or regulations to be met by a software package. [ISO/IEC 24765]

Version 1.2

Requirements phase: the period of time in the software life cycle during which the requirements for a software product are defined and documented. [ISO/IEC 24765]

Software Requirements Specifications: The SRS is a specification for a particular software product, program, or set of programs that performs certain functions in a specific environment. The SRS may be written by one or more representatives of the supplier, one or more representatives of the customer, or by both. [IEEE830-98]

The SRS document contains both functional and non functional requirements.

The SRS can be materialized in a word document but it can also be managed in a database or in an Excel file.

Non Functional Requirement: a software requirement that describes not what the software will do but how the software will do it. ISO/IEC 24765, Systems and Software Engineering Vocabulary. Syn. design constraints, non-functional requirement. See also: functional requirement. NOTE for example, software performance requirements, software external interface requirements, software design constraints, and software quality attributes. Non functional requirements are sometimes difficult to test, so they are usually evaluated subjectively. [ISO/IEC24765]

Prototype: an experimental model, either functional or non functional, of the system or part of the system. *IEEE 1233, 1998 Edition (R2002) IEEE Guide for Developing System Requirements Specifications.* 3.12. **2.** a preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system. *ISO/IEC 24765, Systems and Software Engineering Vocabulary.* **3.** model or preliminary implementation of a piece of software suitable for the evaluation of system design, performance or production potential, or for the better understanding of the software requirements. *ISO/IEC 15910:1999 Information technology -- Software user documentation process.* 4.41. [ISO/IEC24765]

Traceable. 1. having components whose origin can be determined. [ISO/IEC24765]

Traceability matrix. 1. a matrix that records the relationship between two or more products of the development process. [ISO/IEC24765]

3. Relationships with ISO/IEC 29110

This deployment package covers the activities related to requirements analysis of the ISO Technical Report ISO/IEC 29110 Part 5-1 for Very Small Entities (VSEs) – Basic Profile [ISO/IEC29110].

In this section, the reader will find a list of applicable Project Management (PM) and Software Implementation (SI) process, activities, tasks and roles that are directly related to this topic.

- **Process: Software Implementation**
- **Activity: SI.2 Software requirements analysis**
- **Tasks and Roles:**

Task List	Role
SI.2.1 Assign tasks to the Work Team members in accordance with their role, based on the current <i>Project Plan</i> .	TL, WT
SI.2.2 Document or update the <i>Requirements Specification</i> . Identify and consult information sources (customer, users, previous systems, documents, etc.) in order to get new requirements. Analyze the identified requirements to determinate the scope and feasibility. Generate or update the <i>Requirements Specification</i> .	AN, CUS
SI.2.3 Verification of the <i>Requirements Specification</i> . Verify the correctness and testability of the <i>Requirements Specification</i> and its consistency with the <i>Product Description</i> . Additionally, review that requirements are complete, unambiguous and not contradictory. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes were needed, initiate a <i>Change Request</i> .	AN
SI.2.4 Validation of the <i>Requirements Specification</i> Validate that <i>Requirements Specification</i> satisfies needs and agreed upon expectations, including the user interface usability. The results found are documented in a <i>Validation Results</i> and corrections are made until the document is approved by CUS.	CUS, AN
SI.2.5 Document the preliminary version of the <i>Software User Documentation</i> or update the present manual. (optional)	AN
SI.2.6 Verification of the <i>Software User Documentation</i> Verify consistency of the <i>Software User Documentation</i> with the <i>Requirement Specification</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes	AN

Version 1.2

Task List	Role
were needed, initiate a <i>Change Request</i> . (optional)	
SI.2.7 Incorporate the <i>Requirements Specification</i> , and <i>*Software User Documentation</i> to the <i>Software Configuration</i> in the baseline. *(optional)	TL

4. Description of Processes, Activities, Tasks, Steps, Roles and Products

Process: Software Implementation

Activity: SI.2 Software requirements analysis

Task List	Role
SI.2.1 Assign tasks to the Work Team members in accordance with their role, based on the current <i>Project Plan</i> .	TL, WT
SI.2.2 Document or update the <i>Requirements Specification</i> .	AN, CUS
SI.2.3 Verification of the <i>Requirements Specification</i> .	AN
SI.2.4 Validation of the <i>Requirements Specification</i>	CUS, AN
SI.2.5 Document the preliminary version of the <i>Software User Documentation</i> or update the present manual. (optional)	AN
SI.2.6 Verification of the <i>Software User Documentation</i>	AN
SI.2.7 Incorporate the <i>Requirements Specification</i> , and <i>*Software User Documentation</i> to the <i>Software Configuration</i> in the baseline. *(optional)	TL

Tasks

Requirements identification

Objectives:	The objective of this activity is to clearly define the scope of the project and identify the key requirements of the system.
Rationale:	It is important to clearly define the project scope (boundaries) and to identify key functionalities of the future system with the customer to avoid problems like forgotten key functionalities or requirements creep.
Roles:	Project Manager Analyst
Artefacts:	Use Cases – scenarios Requirements Document
Steps:	1. Collect information about the application domain (e.g. finance, medical) 2. Identify project's scope 3. Identify and capture requirements 4. Structure and prioritize requirements

Step Description:	<p>Step 1. Collect information about the domain:</p> <p>During this Step, analyst captures the key concepts of the business domain of the customer. The customer assists the analyst by giving him all the information (existing documentation or explanation) that will facilitate this understanding.</p> <p>Key concepts are listed in a glossary section in the <i>Software Requirements Specification Document outline</i> document.</p> <p>Step 2. Identify project’s scope</p> <p>Software analyst, helped by the person in charge of the contractual aspects of the project (sales manager) clearly identifies main functionalities that are included in the project scope.</p> <p><i>Tips:</i> Identifying functionalities that are OUT of scope is also very valuable to clarify differences of understanding with your customers.</p> <p>Step 3. Identify and capture requirements</p> <p>Having in mind key concepts related to the customer business domain, analyst can start requirements identification. None of the situations in IT projects are identical. In some cases, most of the requirements are already identified in a document (call for tender in case of fixed priced projects). However, in most of the cases, key requirements are just (orally) mentioned by the customer.</p> <p>Analyst must identify and list the key requirements of the system to be built. During this Step, analyst should not start detailing identified requirements. The main goal is to gain a comprehensive view of the system requirements.</p> <p>Step 4. Structure and prioritize requirements:</p> <p>Using requirements identified in the previous Step, the analyst has to organise and structure identified requirements accordingly (e.g. by business processes or by system functions).</p> <p>A priority must be identified by the customer for system key functionalities. Priorities can be stated like</p> <ul style="list-style-type: none"> • ‘High’ - a functionality that <i>shall be</i> implemented • ‘Medium’ - a functionality that <i>should be</i> implemented • ‘Low’ - a functionality that <i>could be</i> implemented <p>The output of this Step is a list of requirements that are organized in the <i>Requirements Document</i>.</p>
--------------------------	---

Requirements refinement and analysis

Objectives:	The objective of this Step is to detail and analyse all the requirements identified.

Version 1.2

Rationale:	It is important to go through identified requirements in order to detect requirements that seem easy to implement but hiding a business complexity that will cause problems in the project.
Roles:	Analyst
	Customer
	Developer
Artefacts:	Use Cases – scenarios
	Requirements Document
	Software prototype
Steps:	1. Detail requirements
	2. Produce a prototype
Step Description:	<p>Step 1. Detail requirements:</p> <p>The analyst goes through a set of identified requirements and performs a more detailed analysis.</p> <p>Business complexity can be implicit for some requirements. This must be clarified at this point before any implementation.</p> <p>The analyst will interact with customer representatives in order to clarify ergonomic question, if relevant (e.g. if a Graphical User Interface must be developed).</p> <p>This Steps results in a new version of the <i>Software Requirements Specification Document</i>.</p> <p>Step 2. Produce a prototype</p> <p>Producing a prototype can facilitate requirement understanding from all project participants (i.e. customer side and development team side). A prototype may only implement some of the functionalities.</p>

Requirements verification & validation

Objectives:	Verify requirements and obtain validation from the customer or his representative.
Rationale:	In order to avoid constant fundamental changes in the requirements, it is important to ask for the requirement validation from the customer.
Roles:	Analyst
	Customer
	Project Manager
	Developer
Artefacts:	Requirements Document
	Software prototype

Version 1.2

Steps:	1. Clarify fuzzy requirements (verification)
	2. Review software requirements specification
	3. Validate requirements
Description:	<p>Step 1. Clarify fuzzy requirements: Review requirements in order to detect requirements that are not clear enough (customer or software developer could understand it differently). Theses criteria can be used to perform this review:</p> <ul style="list-style-type: none"> • Clear (avoid ambiguous requirements) • Unique (i.e. avoid two requirements stating the same thing) • Feasible (according project allocated resources) • Testable <p>Step 2. Review software requirements specification: During this Step requirements are roughly reviewed with the customer in order to be sure that requirements are:</p> <ul style="list-style-type: none"> • Complete • Correct <p><i>Tips:</i> This Step can be iteratively done by reviewing a given subset of requirements. Software engineers must be involved in order to identify technical dependencies between requirements (i.e. Requirement A must be implemented before requirement B due to implementation reason.)</p> <p>Step 3. Validate requirements: Obtain from your customer an approval of the requirements (or of a given subset if you are using an iterative lifecycle).</p>

Requirements change management

Objectives:	To manage requirements change according with a process agreed upon with the customer.
Rationale:	Requirements change is a permanent feature of most of the IT projects. Change management must be planned and agreed upon with the customer on the project.
Roles:	Analyst
	Project Manager
	Customer

Version 1.2

Artefacts:	Requirements Document
Steps:	1. Track changes to requirements
	2. Analyze impact of changes
	3. Identify changes that are out of the project scope
	4. Prioritize changes
Description:	<p>Step 1. Track changes to requirements This Step aims to collect and manage in a central repository (can be an excel sheet or any other database) any changes that are formulated against requirements. This encompasses changes to existing requirements but also new or deleted requirements.</p> <p>Step 2. Analyze impact of changes Identify the impact on the project schedule and cost of each of the requested changes.</p> <p>Step 3. Identify changes that are out of the project scope Analyst helped by the person in charge of the contractual aspects of the project (sales manager) identifies changes that are out of the project scope. Changes that could impact the project budget should be discussed with the customer.</p> <p>Step 4. Prioritize changes During this Step, the project manager must obtain from the customer a prioritization of the identified changes in order to adapt the project planning.</p>

Version 1.2

Roles

Role	Definition
Analyst	Person in charge within the development team to gather, analyze and manage the requirements related to the software to be developed.
Customer	Person in charge within the customer side to transfer and validate requirements to the development team. It can be the customer or any representative.
Developer	Person in charge of the development of the software.
Project Manager	Person in charge of managing the project (cost, schedule, tasks, contract,...)

Table 1 Definitions of Roles**Artefacts**

Artefacts	Definition
Use Cases – scenarios	Description of a sequence of interactions between the user and the future systems. Uses cases can be written as prescribed by UML but can also be text scenarios.
Requirements Document	Document in which all identified requirements are centralized. See <i>Software Requirements Specification</i> definition in section 0.
Software prototype	Working piece of software produced during the early phases in order to demonstrate/validate a functionality of a system.

Table 2 Definitions of Artefacts

Version 1.2

5. Template

The templates provided with this deployment package should be customized for your project.

SRS Template Table of Content –Basic List of Requirements

To be used in an Excel sheet structured, for example, as:

<i>ID</i>	<i>Requirement</i>	<i>Description</i>	<i>Priority</i>

SRS Template Table of Content –Adapted from IEEE 830

1. Introduction

- 1.1 Purpose
- 1.2 Document conventions
- 1.3 Intended audience
- 1.4 Additional information
- 1.5 Contact information/SRS team members
- 1.6 References

2. Overall Description

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User classes and characteristics
- 2.4 Operating environment
- 2.5 User environment
- 2.6 Design/implementation constraints
- 2.7 Assumptions and dependencies

3. External Interface Requirements

- 3.1 User interfaces
- 3.2 Hardware interfaces
- 3.3 Software interfaces
- 3.4 Communication protocols and interfaces

4. System Features

- 4.1 System feature A
 - 4.1.1 Description and priority
 - 4.1.2 Action/result
 - 4.1.3 Functional requirements

Version 1.2

4.2 System feature B

5. Other Non functional Requirements

5.1 Performance requirements

5.2 Safety requirements

5.3 Security requirements

5.4 Software quality attributes

5.5 Project documentation

5.6 User documentation

6. Other Requirements

Appendix A: Terminology/Glossary/Definitions list

Appendix B: To be determined

SRS Template Table of Content -Construx²

1 INTRODUCTION	1
1.1 PURPOSE	1
1.2 SCOPE	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	1
1.4 REFERENCES	1
1.5 OVERVIEW	1
2 OVERALL DESCRIPTION	2
2.1 PRODUCT PERSPECTIVE.....	2
2.1.1 System Interfaces.....	2
2.1.2 User Interfaces	2
2.1.3 Hardware Interfaces.....	2
2.1.4 Software Interfaces.....	2
2.1.5 Communications Interfaces.....	2
2.1.6 Memory Constraints	3
2.1.7 Operations.....	3
2.1.8 Site Adaptation Requirements	3
2.2 PRODUCT FUNCTIONS	3
2.3 USER CHARACTERISTICS.....	3
2.4 CONSTRAINTS	3
2.5 ASSUMPTIONS AND DEPENDENCIES.....	4
2.6 APPORTIONING OF REQUIREMENTS	4
3 SPECIFIC REQUIREMENTS	5
3.1 EXTERNAL INTERFACE REQUIREMENTS	5
3.1.1 User Interfaces	6
3.1.2 Hardware Interfaces.....	6
3.1.3 Software Interfaces.....	6
3.1.4 Communications Interfaces.....	6
3.2 SOFTWARE PRODUCT FEATURES	6
3.2.1 Feature 1	6
Purpose	6
Stimulus/Response Sequence.....	6
Associated Functional Requirements	6
3.3 PERFORMANCE REQUIREMENTS	6
3.4 DESIGN CONSTRAINTS	7
3.5 SOFTWARE SYSTEM ATTRIBUTES	7
3.5.1 Reliability	7
3.5.2 Availability	7
3.5.3 Security.....	7
3.5.4 Maintainability	7
3.6 LOGICAL DATABASE REQUIREMENTS	7
3.7 OTHER REQUIREMENTS	8

² www.construx.com

SRS Template Table of Content – Volere³

Contents

Project Drivers

1. The Purpose of the Project
2. The Client, the Customer, and Other Stakeholders
3. Users of the Product

Project Constraints

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

Functional Requirements

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

Nonfunctional Requirements

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

Project Issues

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

³ <http://atlsysguild.com/GuildSite/Robs/Template.html>

Requirement #: Unique id **Requirement Type:** **Event/use case #'s:** List of events / use cases that need this requirement

Description: A one sentence statement of the intention of the requirement

Rationale: A justification of the requirement

Originator: The person who raised this requirement

Fit Criterion: A measurement of the requirement such that it is possible to test if the solution matches the original requirement

Customer Satisfaction: **Customer Dissatisfaction:** **Conflicts:** Other requirements that cannot be implemented if this one is

Priority: A rating of the customer value

Supporting Materials: ——— Pointer to documents that illustrate and explain this requirement

History: Creation, changes, deletions, etc.

Degree of stakeholder happiness if this requirement is successfully implemented.
Scale from 1 = uninterested to 5 = extremely pleased.

Measure of stakeholder unhappiness if this requirement is not part of the final product.
Scale from 1 = hardly matters to 5 = extremely displeased.

Volere
Copyright © Atlantic Systems Guild

SRS

Template Table of Content – Volere Requirement Shell

6. Example of Lifecycle

Disclaimer: This section provides some graphical representation of example of requirement practices lifecycle. These examples are provided to help the reader to implement his own requirement lifecycle fitting his IT project's context and constraints.

Example 1 of Requirement Practices Lifecycle

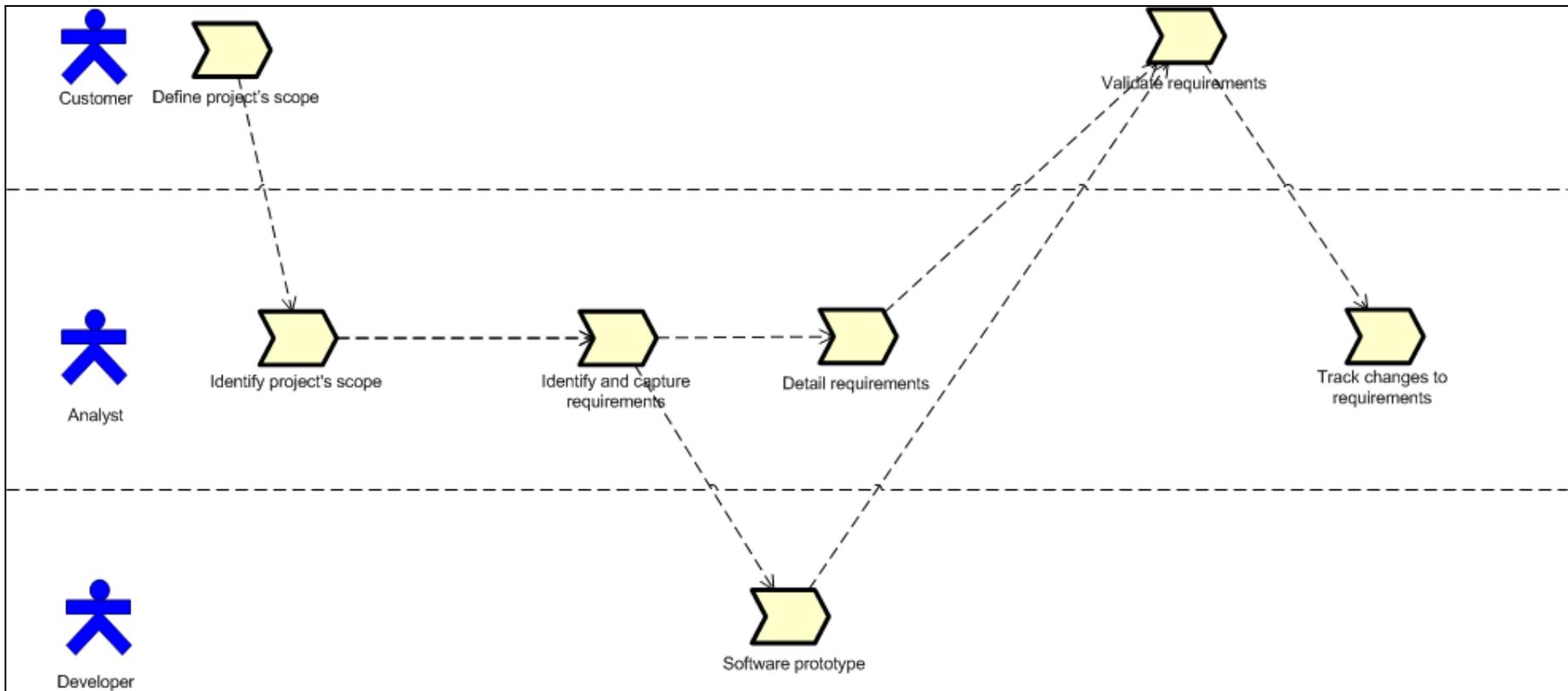


Figure 3 Example 1 of Requirement Practices Lifecycle

Example 2 of Requirement Practices Lifecycle

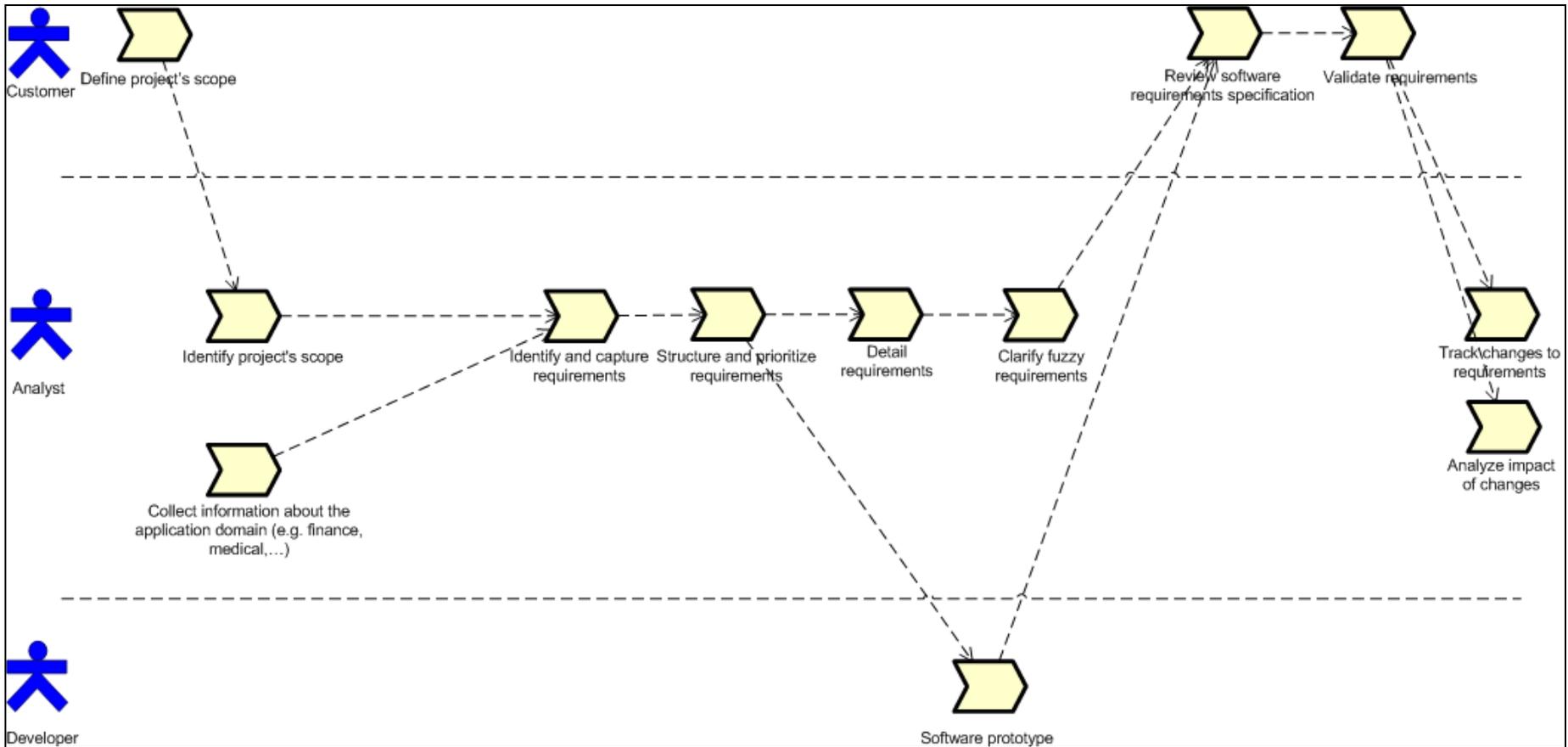


Figure 4 Example 2 of Requirement Practices Lifecycle

Version 1.2

7. Checklist

Requirement checklist

This Requirement checklist is based on [Constr07]

RS 1 Testable	All requirements are verifiable (objectively)
RS 2 Complete	Are the requirements complete?
RS 3 Traceable	All requirements must be traceable to a systems specification, contractual/proposal clause.
RS 4 Correct	Requirements must be correct (i.e. reflect exactly customer's requirements)
RS 5 Unique	Requirements must be stated only once
RS 6 Elementary	Requirements must be broken into their most elementary form
RS 8 High Level	Requirement must be stated in terms of final need, not perceived means (solutions)
RS 9 Quality	Quality attributes have been defined.
RS 10 Unambiguous	
RS 11 Hardware	Hardware environment is completely defined.
RS 12 Solid	Requirements are a solid base for design

8. Tool

Traceability Tool

- Objectives:
 - To maintain the linkage from the source of each requirement through its decomposition to implementation and test (verification).
 - To ensure that all requirements are addressed and that only what is required is developed.
 - Useful when conducting impact assessments of requirements, design or other configured item changes.

Traceability Matrix									
Date (yy-mm-dd): _____									
Title of project: _____									
Name (Print)			Signature				Date (yy-mm-dd)		
Verified by: _____			_____				_____		
Approved by: _____			_____				_____		
Identification Number	Text of the need	Text of the requirement	Verification method	Title or ID of Use Case	Title or ID of Code Module	Title or ID of test Procedure	Verification Date	Name of person who performed the verification	Result of verification

Instructions	
The above table should be created in a spreadsheet or database such that it may be easily sorted by each column to achieve bi-directional traceability between columns. The unique identifiers for items should be assigned in a hierarchical outline form such that the lower level (i.e. more detailed) items can be traced to higher items.	
Unique Requirement Identification (ID)	The Unique Requirement ID / System Requirement Statement where the requirement is referenced, and/or the unique identification (ID) for decomposed requirements
Requirement Description	Enter the description of the requirement (e.g., Change Request description).
Design Reference	Enter the paragraph number where the CR is referenced in the design documentation
Module / Configured Item Reference	Enter the unique identifier of the software module or configured item where the design is realized.
Release Reference	Enter the release/build version number where the requirement is fulfilled
Test Script Name/Step Number Reference	Enter the test script name/step number where the requirement is referenced (e.g., Step 1)

Version 1.2

Guideline

Requirements traceability should:

- Ensure traceability for each level of decomposition performed on the project. In particular:
 - Ensure that every lower level requirement can be traced to a higher level requirement or original source
 - Ensure that every design, implementation, and test element can be traced to a requirement
 - Ensure that every requirement is represented in design and implementation
 - Ensure that every requirement is represented in testing/verification
- Ensure that traceability is used in conducting impact assessments of requirements changes on project plans, activities and work products
- Be maintained and updated as changes occur.
- Be consulted during the preparation of Impact Assessments for every proposed change to the project
- Be planned for, since maintaining the links/references is a labor intensive process that should be tracked/monitored and should be assigned to a project team member
- Be maintained as an electronic document

9. References to Other Standards and Models

This section provides references of this deployment package to ISO/IEC Standards and to the Capability Maturity Model IntegrationSM version 1.2 of the Software Engineering Institute (CMMI^{®4}).

Notes:

- This section is provided for information purpose only.
- Only tasks covered by this Deployment Package are listed in each table.
- The tables use the following convention:
 - Full Coverage = F
 - Partial Coverage = P
 - No Coverage = N

Note: Coverage matrices are not completed, they are provided as an example to illustrate how to complete them.

ISO 9001 Reference Matrix

Title of the Task and Step	Coverage F/P	Clause of ISO 9001	Comments
Requirements identification Step 1- Collect information about the application domain	F	7.2.1 Determination of requirements related to the product b) requirements not stated by the customer but necessary for specified or intended use, where known,	
Requirements identification Step 2 - Identify project's scope	F	7.2.1 Determination of requirements related to the product The organization shall determine a) requirements specified by the customer, including the requirements for delivery and post-delivery activities,	

SM CMM Integration is a service mark of Carnegie Mellon University.

[®] Capability Maturity Model, CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

ISO/IEC 12207 Reference Matrix

Title of the Task and Step	Coverage F/P	Clause of ISO/IEC 12207	Comments
Requirements identification Step 1 - Collect information about the application domain			
Requirements identification Step 2 - Identify project's scope	P	7.1.2 Software Requirements Analysis Process 7.1.2.2 Outcomes	

CMMI Reference Matrix

Title of the Task and Step	Coverage F/P	Objective/ Practice of CMMI V1.2	Comments
Requirements identification Step 1 - Collect information about the application domain	P		
Requirements identification Step 2 - Identify project's scope	P	SG 1 Develop Customer Requirements	

10. References

Key	Reference
[ISO/IEC 29110]	Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1: Management and Engineering Guide - Basic VSE Profile
[OWPL-EN]	Renault A., Habra N., Alexandre S., Deprez J.-C., OWPL. Software Process Improvement for VSE, SME and low maturity enterprises. Version 1.2.2, FUNDP-CETIC, 2000. (http://www.cetic.be/internal393.html)
[IEEE830-98]	IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.
[ISO/IEC12119]	ISO/IEC 12119:1994 Information technology – Software packages -- Quality requirements and testing.
[ISO/IEC12207]	ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes.
[ISO/IEC24765]	ISO/IEC 24765, Systems and Software Engineering Vocabulary
[ConstSoft02]	Construx Software – Checklist for Software Requirements Specifications, 2002.
[SELB07]	Selby, P., Selby, R.W., Measurement-Driven Systems Engineering Using Six Sigma Techniques to Improve Software Defect Detection, Proceedings of 17 th International Symposium, INCOSE, June 2007, San Diego.
[STAN02]	Standish Group – Chaos report 2002.
[SPEM05]	Software Process Engineering Metamodel Specification, OMG, 2005.
[VOLE07]	Volere, Requirements Resources - http://www.volere.co.uk

Version 1.2

11. Evaluation Form

Deployment Package – Software Requirements Analysis – Version 1.2

Your feedback will allow us to improve this package, your comments and suggestions are welcomed

1. How satisfied are you with the **CONTENT** of this deployment package?

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

2. The sequence in which the topics are discussed, are logical and easy to follow?

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

3. How satisfied were you with the **APPEARANCE/FORMAT** of this deployment package?

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

4. Have any unnecessary topics been included? (please describe)

5. What missing topic would you like to see in this package? (please describe)

- Proposed topic:
- Rationale for new topic

6. Any error in this deployment package?

- Please indicate:
 - Description of error :
 - Location of error (section #, figure #, table #) :

7. Other feedback or comments:

8. Would you recommend this Deployment package to a colleague from another VSE?

Definitely Probably Not Sure Probably Not Definitely Not

Optional

- Name: _____
- e-mail address : _____

Email this form to : simon.alexandre@cetic.be or: claudio.y.laporte@etsmtl.ca or Avumex2003@yahoo.com.mx

**Apéndice F. Paquete de Despliegue para la Mejora de
Procesos COMPETISOFT**

Paquete de Despliegue para la Mejora de Procesos COMPETISOFT

Notas:

Este documento es propiedad intelectual del autor y su afiliación. Sin embargo, la información contenida en este documento es de libre uso. La distribución de la totalidad o partes de este documento no está autorizada para uso comercial, siempre y cuando se mencione el siguiente aviso legal:

© Instituto de Tecnologías de la Información y Sistemas, Universidad de Castilla-La Mancha. Posgrado en Ciencia e Ingeniería de la Computación. Universidad Nacional Autónoma de México.

El uso comercial de este documento está estrictamente prohibido. Este documento se distribuye con el fin de mejorar el intercambio de información técnica y científica.

El autor (es) no ofrece garantías de ningún tipo, ya sea expresa o implícita, en cuanto a cualquier asunto, incluyendo, la garantía con el propósito de comercialización, exclusividad, o resultados obtenidos a partir de la utilización del material.

Los procesos descritos en este paquete de despliegue no están destinados a impedir o desalentar la utilización de otros procesos que las VSEs puedan encontrar útiles.

Autores, Afiliación, País	Ricardo Cruz Mendoza – Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México. Hanna Oktaba – Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México. Francisco J. Pino – IDIS Grupo de Investigación en Electrónica y Telecomunicaciones, Facultad de Ingeniería, Colombia. Mario Piattini – Alarcos Grupo de Investigación, Instituto de Tecnologías de la Información y Sistemas, Universidad de Castilla-La Mancha, España.
Aseguramiento de Calidad	Miguel Morales Trujillo – Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México.
Fecha de Creación	26 Mayo 2009
Ultima Actualización	26 Mayo 2009
Estatus	Borrador
Versión	0.1

Versiones

Fecha	Versión	Autor	Modificación
26/Mayo/2009	0.1	Ricardo Cruz Mendoza, Hanna Oktaba, Francisco J. Pino, Mario Piattini.	Creación del Documento

Abreviaciones/Acrónimos

Abre./Acro.	Definición
VSE	Very Small Entities por sus siglas en inglés – empresas, organizaciones, departamentos o proyectos que cuenten con menos de 25 empleados.

Tabla de Contenidos

Descripción Técnica.....	5
<i>Propósito del Documento</i>	<i>5</i>
Relación con COMPETISOFT.....	6
Definiciones Clave	10
Introducción.....	11
<i>Propósitos de la mejora de procesos utilizando PmCOMPETISOFT.....</i>	<i>12</i>
Descripción detallada de los procesos, actividades, tareas, roles y productos	14
<i>Descripción detallada de los procesos, actividades y tareas.</i>	<i>15</i>
<i>Roles</i>	<i>29</i>
<i>Descripción de los Productos</i>	<i>30</i>
Plantillas	33
Ejemplos de plantillas	34
Listas de verificación.....	35
Herramientas	36
Referencia a otros estándares y modelos de referencia de software	37
Referencias Bibliográficas	38
Forma de Evaluación	39

Descripción Técnica

Propósito del Documento

El propósito de este documento, titulado: “Paquete de Despliegue para la Mejora de Procesos COMPETISOFT” es proveer a las VSEs de una guía capaz de ajustarse a las necesidades de la empresa con el fin de establecer los elementos necesarios para guiar y gestionar la mejora de procesos en una pequeña organización software, y lograr institucionalizar la cultura de la mejora continua al interior de la organización.

Un paquete de despliegue es un conjunto de productos desarrollados para facilitar la implementación de un conjunto de prácticas en algún modelo de referencia de procesos, en una VSE. La descripción de las tareas en un paquete de despliegue, tienen como propósito facilitar la implementación de un proceso en un modelo de referencia como puede ser ISO/IEC 29110-5-1 o COMPETISOFT. Los elementos típicos de un paquete de despliegue son:

- Descripción de los procesos (actividades, tareas, roles y productos).
- Plantillas y Ejemplos de Plantillas.
- Listas de Verificación.
- Herramientas.
- Mapeos a otros estándares y modelos.

Este documento apoya el Modelo de Mejora de COMPETISOFT. Se tiene que tener en cuenta que este documento no intenta ser un sustituto de un modelo de referencia de procesos de software, sino un apoyo para comprender y facilitar la implementación de un proceso, en el caso de este documento se pretende apoyar la implementación de un Modelo de Mejora de Procesos basándose en COMPETISOFT en una VSE.

Relación con COMPETISOFT

Este paquete de despliegue cubre las actividades relacionadas con el Proceso de Mejora de COMPETISOFT.

En esta sección el lector encontrará una breve descripción de las actividades, tareas y roles relacionados con la mejora de procesos. Estos tópicos se detallarán en una sección posterior.

- **Actividad:** INSTALACIÓN DEL CICLO
- **Tareas:**

Roles¹	Tareas²
GGM	A1.1 Empezar Instalación.
GGM	A1.2. Identificar las necesidades del negocio y los requisitos que conducen la mejora.
RPM y GGM	A1.3. Construir una Propuesta de Mejora (PT01_PM) del proceso de software.
GGM	A1.4. Verificar la Propuesta de Mejora.
GGM	A1.5. Corregir los defectos encontrados en la Propuesta de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A1.6. Validar la Propuesta de Mejora.
GGM	A1.7. Corregir los defectos encontrados en la Propuesta de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.
GGM	A1.8. Obtener la aprobación de la propuesta de mejora y los recursos iniciales.
RMP y GGM	A1.9. Adecuar la propuesta de mejora de acuerdo a los recursos aprobados, las restricciones de la organización y las recomendaciones de la gerencia.
RMP y GGM	A1.10. Lanzar el ciclo.

¹ Los Roles serán definidos en una sección posterior.

² Las Tareas serán descritas en una sección posterior.

Versión 0.1

- **Actividad:** DIAGNÓSTICO DE PROCESOS
- **Tareas:**

Roles	Tareas
EV	A2.1. Valorar.
GMP	A2.2. Priorizar.
GMP	A2.3. Visualizar el mapa de procesos que tiene la organización.
GMP	A2.4. Construir un Plan General de Mejora.
RMP	A2.5. Verificar el Plan General de Mejora.
GGM	A2.6. Corregir los defectos encontrados en el Plan General de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A2.7. Validar el Plan General de Mejora.
GGM	A2.8. Corregir los defectos encontrados en el Plan General de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.
GMP	A2.9. Capacitación en mejora de procesos.
GGM	A2.10. Comunicar el Plan General de Mejora.

- **Actividad (Opcional):** PvalCOMPETISOFT
- **Tareas:**

Roles	Tareas
EV, RMP	A1. Planificación de la Valoración
EV, RP	A2. Ejecución de la Valoración
EV, RMP	A3. Generación y Socialización de Resultados
RMP	A4. Priorización de Procesos
RMP	A5. Planificación Preliminar de Mejoras

- **Actividad:** FORMULACIÓN DE MEJORAS
- **Tareas:**

Roles	Tareas
GMP	A3.1. A partir del Plan General de Mejora, escoger los casos de mejora que estén relacionados con un proceso a mejorar.
GMP y RP	A3.2. Construir el Plan de Implementación de Mejora, a partir de la formulación del caso de mejora.
RMP	A3.3. Verificar el Plan de Implementación de Mejora.
GGM	A3.4. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RMP	A3.5. Validar el Plan de Implementación de Mejora.
GGM	A3.6. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.

- **Actividad (Opcional):** PfemCOMPETISOFT
- **Tareas:**

Roles	Tareas
GGM	Planificación de la Iteración
GMP	Diseño del Caso de Mejora
	Iteración de Mejora
RMP	Presentación del Proceso Mejorado
RMP	Presentación de la Próxima Iteración

- **Actividad:** EJECUCIÓN DE MEJORAS
- **Tareas:**

Roles	Tareas
GMP y RP	A4.1. Ejecutar el caso de mejora formulado en la actividad anterior, descrito en el Plan de Implementación de Mejora.
GMP	A4.2. Documentar las iteraciones de mejora y actualizar el Plan de Implementación de Mejora.
RMP	A4.3. Verificar el Plan de Implementación de Mejora.
GGM	A4.4. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A4.5. Validar el Plan de Implementación de Mejora.
GGM	A4.6. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.

- **Actividad:** REVISIÓN DEL CICLO
- **Tareas:**

Roles	Tareas
GGM y GMP	A5.1. Realizar realimentación.
GMP	A5.2. Crear/Actualizar Base de Conocimiento.
GGM y RMP	A5.3. Analizar impacto de mejora.
GGM y RMP	A5.4. Analizar el compromiso.
GGM y RMP	A5.5. Preparación del siguiente ciclo de mejora.
GGM	A5.6. Verificar el Reporte de Mejora.
GGM	A5.7. Corregir los defectos encontrados en el Reporte de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A5.8. Validar el Reporte de Mejora.
GGM	A5.9. Corregir los defectos encontrados en el Reporte de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.

Definiciones Clave

COMPETISOFT [Hanna et al., 2007]. Es un proyecto financiado por CYTED, programa internacional de cooperación científica y tecnológica multilateral, de ámbito iberoamericano que tiene como propósito incrementar el nivel de competitividad de las VSEs iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, llegue a ser la base sobre la que se pueda establecer un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica.

PmCOMPETISOFT [MMPCOMPETISOFT]: El Proceso de mejora continua de procesos de COMPETISOFT tiene como propósito mejorar los procesos de la organización en función de sus objetivos de negocio, así como ayudar a conducir la mejora de procesos software enfocada en las pequeñas organizaciones a través de la definición de una guía para implementar paso a paso la mejora de procesos.

PvaCOMPETISOFT [MMPCOMPETISOFT]: El proceso para la valoración de procesos de COMPETISOFT tiene como propósito generar información confiable con la cual identificar de manera general las fortalezas, debilidades y riesgos de los procesos software de la organización, esta información debe ser la base para tomar las decisiones relacionadas con el ciclo de mejora.

PfemCOMPETISOFT [MMPCOMPETISOFT]: El proceso de formulación y ejecución de mejoras de COMPETISOFT, es un proceso para ayudar a las pequeñas empresas a formular y ejecutar las oportunidades de mejora encontradas en la actividad de diagnóstico de procesos. Para ello, sigue la filosofía del método ágil de gestión de procesos Scrum. El objetivo es aprovechar la sinergia de la valoración de procesos y encausarla hacia la iniciación de la realización efectiva de mejoras al interior de la pequeña organización software.

SPI: Software Process Improvement, por sus siglas en inglés. Una Mejora de Procesos Software es una serie de acciones adoptadas para identificar, analizar y mejorar los procesos existentes dentro de una organización para hacer frente a nuevas metas y objetivos. Estas acciones a menudo siguen una metodología o estrategia específica para obtener mejores resultados.

Introducción

Actualmente hay pocos trabajos relacionados con modelos que guían y gestionan la mejora de procesos de software (SPI por sus siglas en inglés) para pequeñas organizaciones software (VSEs por sus siglas en inglés) y esto ha conducido a que la utilización de un modelo de implementación de mejora por parte de las pequeñas organizaciones sea baja [Pino et al., 2008b].

Esto supone un gran inconveniente porque un modelo de este tipo constituye la guía necesaria para articular todas las demás actividades relacionadas con la mejora de procesos y por supuesto dinamizar todos los demás modelos involucrados.

Teniendo en cuenta este aspecto, se crea el Modelo de Mejora de COMPETISOFT, el cual guía y gestiona la implementación de proyectos en el contexto de las VSEs.

Para guiar a las pequeñas organizaciones software en la conducción y gestión de los proyectos de mejora, el modelo de mejora de COMPETISOFT define un conjunto de componentes (ver Figura 1.0), los cuales se describen a continuación:

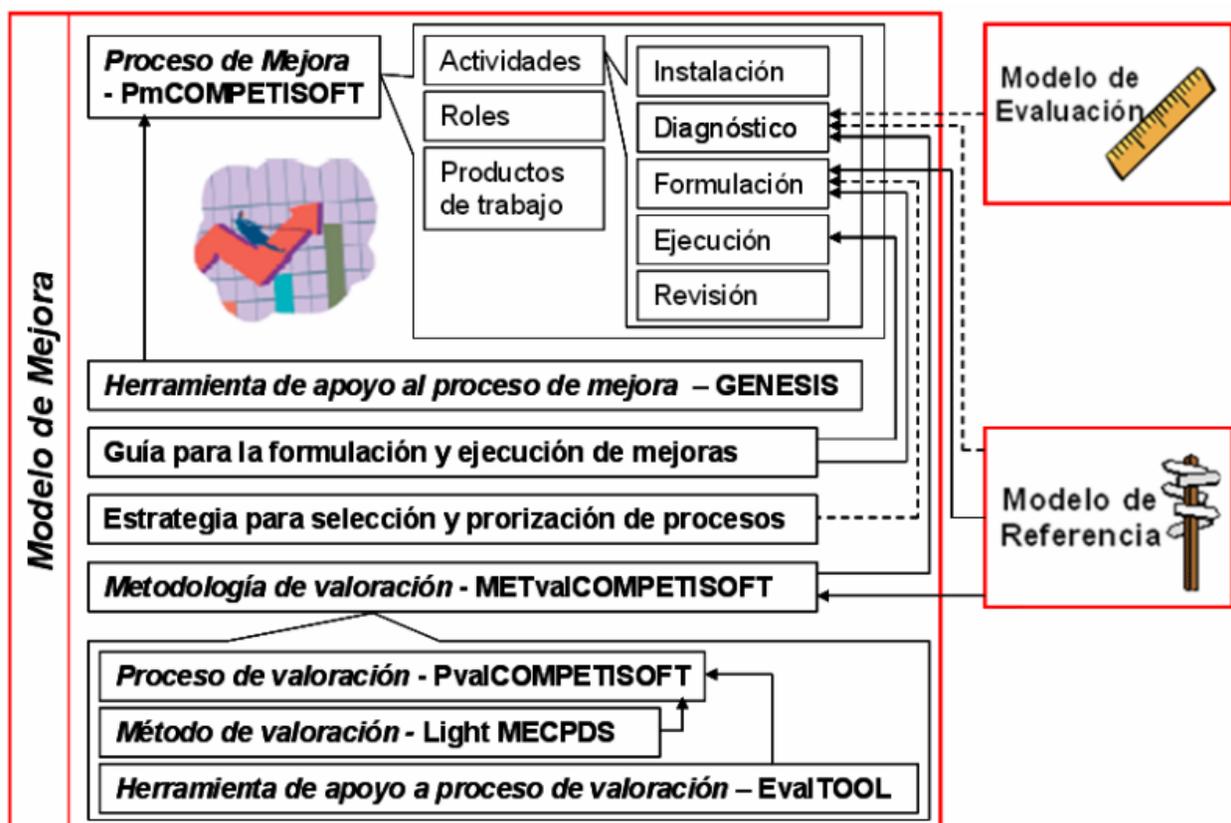


Figura 1.0 Componentes del modelo de mejora de COMPETISOFT.

- Un proceso para guiar la mejora continua de procesos denominado PmCOMPETISOFT, y una herramienta software, denominada GENESIS, para apoyar al responsable de conducir la mejora en la implementación de las actividades descritas por este proceso.

- Una metodología para la valoración de procesos (evaluación interna de procesos) denominada METvalCOMPETISOFT, la cual está compuesta por: (i) un proceso para guiar la valoración de procesos, (ii) un método para la valoración de procesos, (iii) un conjunto de medidas para estimar el rendimiento y la capacidad de los procesos, y (iv) un conjunto de herramientas de soporte a la valoración de procesos software.
- Una guía para formular y ejecutar mejoras utilizando Scrum, con el fin de aprovechar la sinergia de la valoración de procesos y encausarla hacia la iniciación de la realización efectiva de mejoras al interior de la pequeña organización software.
- Una estrategia para la selección y priorización de procesos. [Hanna et al.]

Dado que un paquete de despliegue sintetiza partes fundamentales del proceso de un modelo de mejora se optó por utilizar las partes esenciales de PmCOMPETISOFT, METvalCOMPETISOFT (PvalCOMPETISOFT) y PfemCOMPETISOFT como marco de referencia para la creación del mismo.

Propósitos de la mejora de procesos utilizando PmCOMPETISOFT.

Dentro de PmCompetisoft se han planteado los siguientes propósitos:

- Satisfacer las necesidades de mejora de procesos de la organización, a través de la entrega temprana y continua de mejoras significativas de los procesos de desarrollo de software –PDS-, es la principal prioridad. Obtener mejoras de los PDS a corto plazo lo cual implica seguir un ciclo de vida iterativo e incremental.
- Diagnosticar continuamente los procesos de la organización, ya que es difícil definir los requisitos de mejora de procesos totalmente estables por parte de la organización, en especial cuando empiezan a trabajar con SPI. Analizar, priorizar y llevar a cabo estos nuevos requisitos de mejora de procesos en la medida que sea factible.
- Establecer una colaboración efectiva entre los diferentes actores involucrados en el ciclo SPI. Un ciclo de mejora debe basarse en la colaboración entre todas las personas involucradas y relacionadas con este. Debe buscarse los mecanismos humanos y tecnológicos para facilitar la comunicación eficiente y efectiva dentro de un equipo de mejora.
- Construir proyectos individuales, grupales y organizacionales en torno a individuos motivados hacia la mejora de procesos. Brindarles la oportunidad, el respaldo y la confianza que necesitan para que realicen las tareas de manera autónoma. PmCOMPETISOFT promueve la conformación efectiva de los grupos y se preocupa por la calidad del trabajo humano a realizar.

Además PmCOMPETISOFT para el éxito del ciclo de mejora promueve:

- El desarrollo sostenido del ciclo de mejora, a través del trabajo continuo. La capacidad del proceso y el desempeño promedio de los proyectos de desarrollo de software que debe ser la forma primaria y liviana de medir la mejora del progreso.
- Una infraestructura técnica y de gestión, adecuada para soportar la mejora de procesos. PmCOMPETISOFT promueve la conformación de una infraestructura organizacional dinámica, basada en objetivos y no en estrategias de control.

- El aprendizaje continuo como una disciplina clave. El objetivo de esta disciplina es que permita conocer el trabajo, reflexionar acerca de este y ajustarlo a través de iteraciones cortas y concisas.

Descripción detallada de los procesos, actividades, tareas, roles y productos

En esta sección el lector encontrará una serie de actividades, tareas, roles y productos que están directamente relacionados con la mejora de procesos.

Los siguientes elementos se describen brevemente:

- **Proceso:** conjunto de actividades interrelacionadas.
- **Actividad:** Un conjunto de tareas de un proceso.
- **Tarea:** Un conjunto de acciones o pasos con la finalidad de cumplir un producto, trabajo o problema.
- **Roles:** Roles Claves³ involucrados en la tarea (*ej. responsable del proceso, evaluador, responsable de la mejora de procesos*)
- **Producto:** Pieza de información entregable que puede o no ser producida por una o más tareas (*ej. tutoriales, documentos, código fuente*).

Nota: No existen reglas específicas para el formato de un producto (*ej. El manejo de los documentos puede ser a través de un documento de word o en una herramienta basada en web*).

Nota: El esfuerzo requerido en cada paso puede8 variar en función del tamaño de la empresa, tamaño del proyecto, horas/persona, persona/días.

³ Es importante mencionar que una sola persona puede ejercer diferentes roles, especialmente en VSEs.

Descripción detallada de los procesos, actividades y tareas.

Proceso: Modelo de Mejora Continua de Procesos Software

El modelo de mejora tiene como propósito mejorar los procesos de la organización en función de sus objetivos de negocio, así como ayudar a conducir la mejora de procesos software enfocada en las pequeñas organizaciones a través de la definición de una guía para implementar paso a paso la mejora de procesos. En la Figura 2.0 se muestra la estructura de desglose de trabajo del proceso de mejora continua.

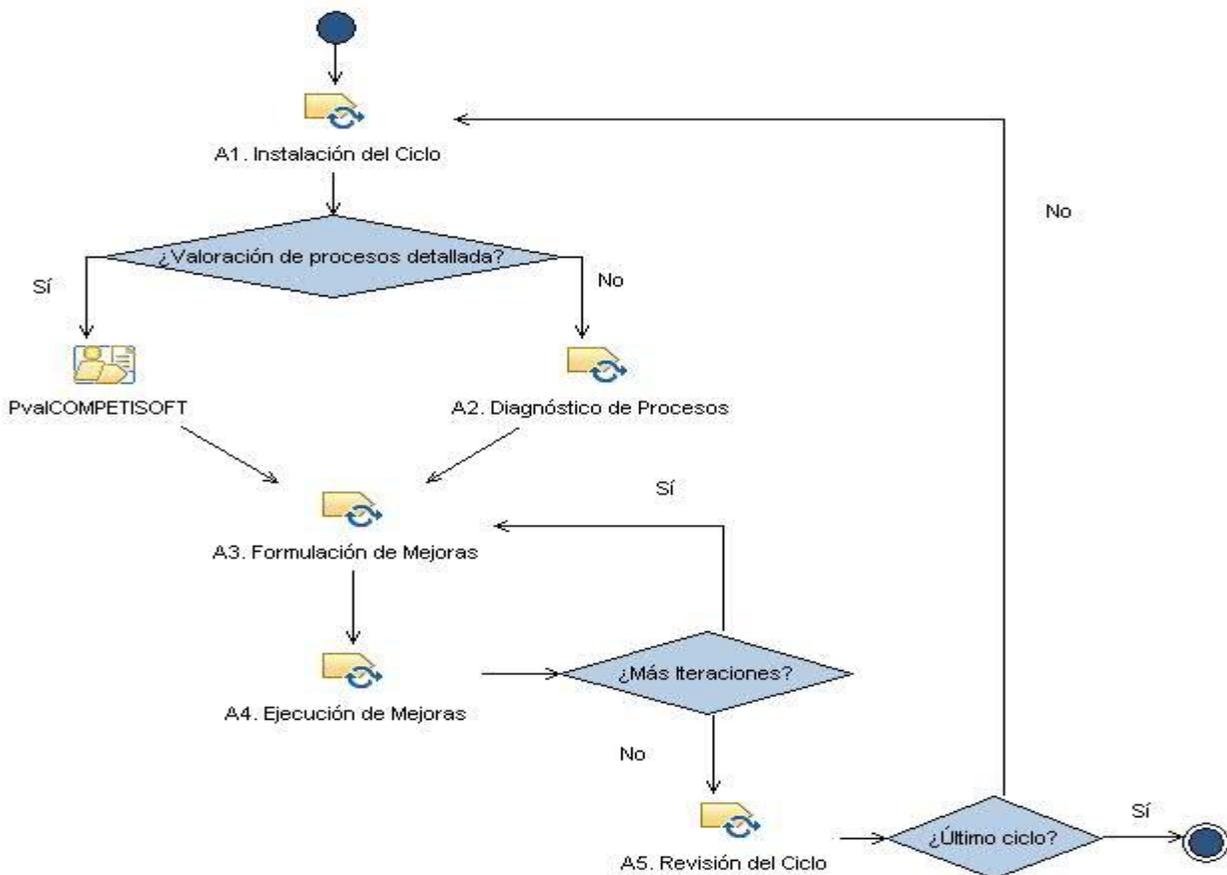


Figura 2.0 Desglose de trabajo del proceso de mejora

El proceso de mejora continua de procesos se compone de uno o más ciclos de mejora⁴. Cada ciclo de mejora consta de 5 actividades: Instalación del ciclo, Diagnóstico de procesos, Formulación de mejoras, Ejecución de mejoras, Revisión del ciclo y dos actividades opcionales (PvalCOMPETISOFT y PfemCOMPETISOFT). En el siguiente subtema se detallará cada una de las actividades de trabajo con sus respectivas tareas.

⁴ Un ciclo de mejora es un conjunto de actividades conducentes a introducir mejoras de procesos software al interior de la organización.

- **Actividad:** *INSTALACIÓN DEL CICLO.* Se crea ó actualiza una *Propuesta de Mejora* alineada con la planeación estratégica de la organización plasmada en el *Plan Estratégico*. Esta propuesta guía a la organización a través de cada una de las fases siguientes del ciclo de mejora. La propuesta debe ser aprobada y divulgada para garantizar, de esta manera, la asignación de los recursos necesarios y el compromiso de los involucrados.
- **Entradas:** *Plan Estratégico de la Organización* que evidencie: Sensibilización del grupo directivo por la mejora de procesos de la organización
- **Salidas:** *Propuesta de Mejora.* Confirmación de GGMM, GMP y el RMP, Compromiso de la organización con la propuesta de Mejora.
- **Tareas:**

Roles	Detallado de las Tareas
GGM	A1.1 Empezar Instalación. El Grupo Directivo designa un Responsable de Mejora de Procesos quien organiza un equipo de trabajo para presentar una propuesta de mejora.
GGM	A1.2. Identificar las necesidades del negocio y los requisitos que conducen la mejora. Teniendo en cuenta el Plan Estratégico de la organización: <ul style="list-style-type: none"> • Identificar las necesidades y objetivos del negocio. • Definir los objetivos generales de mejora alineados con el Plan Estratégico. • Definir el(los) modelo(s) referente(s) para la valoración y mejora de los procesos. • Definir los roles de acuerdo al proceso de mejora a seguir.
RPM y GGM	A1.3. Construir una Propuesta de Mejora (PT01_PM) del proceso de software. <ul style="list-style-type: none"> • Construir una propuesta preliminar donde se desarrolle una estrategia para el desarrollo de los objetivos de mejora planteados. • Estimar tiempo, recursos y costo inicial conforme a los datos de ciclos de mejora anteriores.
GGM	A1.4. Verificar la Propuesta de Mejora.
GGM	A1.5. Corregir los defectos encontrados en la Propuesta de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A1.6. Validar la Propuesta de Mejora.

GGM	A1.7. Corregir los defectos encontrados en la Propuesta de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.
GGM	A1.8. Obtener la aprobación de la propuesta de mejora y los recursos iniciales. Presentar la propuesta de mejora a la gerencia y obtener su aprobación.
RMP y GGM	A1.9. Adecuar la propuesta de mejora de acuerdo a los recursos aprobados, las restricciones de la organización y las recomendaciones de la gerencia.
RMP y GGM	A1.10. Lanzar el ciclo. <ul style="list-style-type: none"> • Sensibilizar a toda la organización acerca del proyecto de mejora. • Informar de las actividades y pasos subsecuentes que se deben seguir con el propósito de comenzar el ciclo de mejora de procesos.

- **Actividad:** *DIAGNÓSTICO DE PROCESOS*. Se realiza la actividad de evaluación interna de procesos para conocer el estado general de los procesos de la organización y analizar los resultados con el objetivo de establecer las oportunidades de mejora de un proceso (las cuales se pueden agrupar en casos de mejora⁵) y su prioridad de mejora, lo cual queda consignado en el *Informe de Valoración*. Se realiza una planificación preliminar y general del ciclo de mejora, determinando una ruta de ejecución de la mejora a través de las iteraciones con el objetivo de generar una versión preliminar del *Plan General de Mejora*.
- **Entradas:** *Propuesta de Mejora*. Información del personal interno de la empresa. Procesos actuales de la organización.
- **Salidas:** *Plan General de Mejora, Informe de Valoración, Instrumentos de Recolección de Información, Plan de Valoración, Plan Preliminar de Mejora*.
- **Tareas:**

Roles	Detallado de las Tareas
EV	A2.1. Valorar. Identificar mediante una valoración ligera de procesos de software de la organización su perfil de capacidad con base en el modelo de procesos de referencia y el método de valoración seleccionado. Registrar la información de los resultados obtenidos en el Informe de Diagnóstico.

⁵ El caso de mejora está compuesto de una o más oportunidades de mejora, las cuales se agrupan en función de los objetivos del ciclo de mejora y del plan estratégico de la organización. Un caso de mejora es un contenedor que es utilizado para gestionar la complejidad de las oportunidades de mejora.

GMP	<p>A2.2. Priorizar.</p> <ul style="list-style-type: none"> • Analizar los resultados de la valoración e identificar las oportunidades de mejora (casos de mejora) a realizarse. • Priorizar los casos de mejora de acuerdo a los objetivos de la organización.
GMP	A2.3. Visualizar el mapa de procesos que tiene la organización.
GMP	<p>A2.4. Construir un Plan General de Mejora.</p> <p>Determinar una ruta preliminar de ejecución de los casos de mejora a través de iteraciones de mejora de acuerdo a la priorización previa que se haya realizado.</p>
RMP	A2.5. Verificar el Plan General de Mejora.
GGM	A2.6. Corregir los defectos encontrados en el Plan General de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A2.7. Validar el Plan General de Mejora.
GGM	A2.8. Corregir los defectos encontrados en el Plan General de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.
GMP	<p>A2.9. Capacitación en mejora de procesos.</p> <p>Planear y realizar jornadas de capacitación en la organización relacionadas con temas que involucra la mejora de procesos en la organización.</p>
GGM	<p>A2.10. Comunicar el Plan General de Mejora.</p> <ul style="list-style-type: none"> • Contribuir a que la organización entera esté informada acerca de todo el ciclo de mejora y de la manera como éste se ha planificado. • Realizar la estrategia y material de divulgación. El material puede ser una serie de informes o comunicados, correo interno, carteleras, pizarras, reuniones o en la página principal de una Intranet, donde se comunique el trabajo que se ha planificado realizar durante el ciclo de mejora.

- **Actividad (Opcional):** PvalCOMPETISOFT. Proceso que tiene como propósito generar información confiable con la cual identificar de manera general las fortalezas, debilidades y riesgos de los procesos software de la organización, esta información debe ser la base para tomar las decisiones relacionadas con el ciclo de mejora. El proceso para la valoración de procesos software consta de 5 actividades: Planificación de la valoración, Ejecución de la valoración, Generación de resultados, Priorización de procesos y Planificación preliminar de mejoras. En la Figura 3.0 se muestra el desglose de actividades de PvalCOMPETISOFT.

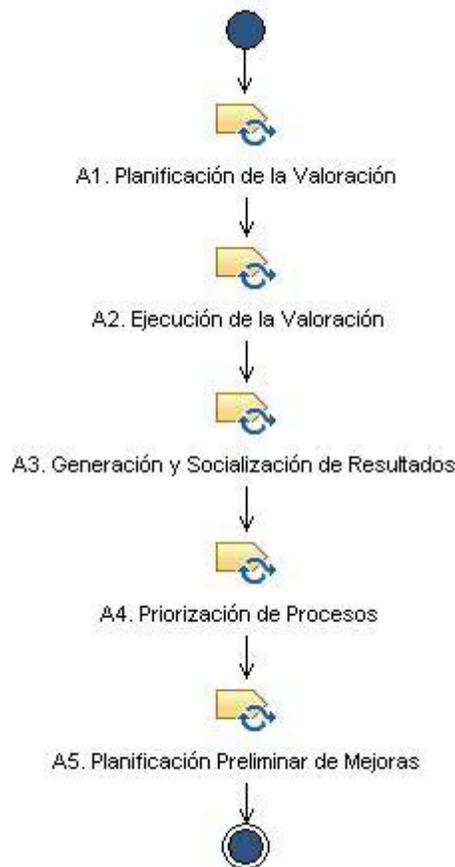


Figura 3.0 Desglose de actividades de PvalCOMPETISOFT

- **Entradas:** *Propuesta de Mejora, Informe de Valoración, Instrumentos de Recolección de Información, Instrumentos de Recolección de Información Completados, Oportunidades de Mejora Priorizadas.*
- **Salidas:** *Instrumentos de Recolección de Información, Instrumentos de Recolección de Información Completados, Plan de Valoración, Informe de Valoración, Oportunidades de Mejora Priorizadas, Plan Preliminar de Mejora.*

• **Tareas:**

Roles	Detallado de las Tareas
EV, RMP	<p>A1. Planificación de la Valoración.</p> <p>A partir de la <i>Propuesta de Mejora</i> que contiene: el objetivo general de mejora, el proceso de mejora continua, el alcance, las metas globales y los recursos del ciclo de mejora actual, se desarrolla y documenta un plan de valoración. El Evaluador y el Responsable de mejora de procesos describen los objetivos y el alcance de la valoración, el instrumento de recolección de información a aplicar en la organización, el método de valoración utilizado para determinar la capacidad de los procesos, el proyecto a evaluar y un cronograma de la valoración (involucra fecha, actividades, tiempo estimado para realizar cada actividad y roles involucrados, entre otros).</p> <p>Sub-actividades :</p> <ul style="list-style-type: none"> • A1.1. Revisar la Propuesta de Mejora. • A1.2. Definir los objetivos de la valoración. • A1.3. Definir el alcance de la valoración. • A1.4. Definir el calendario para la valoración. • A1.5. Definir el método de evaluación utilizado para determinar la capacidad de los procesos. • A1.6. Crear los instrumentos para recolectar la información de la organización. • A1.7. Definir el proyecto de valoración. • A1.8. Crear el Plan de Valoración.
EV, RP	<p>A2. Ejecución de la Valoración.</p> <p>Teniendo en cuenta el alcance definido en la actividad anterior el Evaluador recolecta y valida sistemáticamente los datos necesarios para la valoración de procesos mediante la aplicación de técnicas de recolección de evidencias como la entrevista y encuesta, siguiendo los instrumentos de recolección de información definidos o creados para tal fin. Para validar la información el Evaluador aplica la técnica de recolección de información de manera separada al Responsable del proceso a valorar (con el cual se revisa la documentación de los procesos de la organización) y a una persona involucrada en la ejecución de este proceso.</p> <p>Sub-actividades :</p> <ul style="list-style-type: none"> • A2.1. Aplicar los instrumentos para recolectar la información de los procesos. • A2.2. Validar la información recolectada.
EV, RMP	<p>A3. Generación y Socialización de Resultados</p> <p>El Evaluador analiza y sintetiza los datos obtenidos con los</p>

	<p>instrumentos de recolección de información para determinar el estado actual de los procesos mediante el hallazgo de sus debilidades, riesgos y fortalezas. Por cada proceso valorado se presenta información relacionada con su diagrama de actividades, los hallazgos, los productos de trabajo, así como las oportunidades de mejora. Finalmente el Evaluador asigna un valor a los atributos de proceso para establecer un perfil del nivel de capacidad de los procesos valorados en la organización. El Evaluador entrega al Responsable de mejora de procesos el producto de trabajo <i>Informe de valoración</i> con los resultados de la valoración de los procesos de la organización para su validación y refinamiento.</p> <p>Sub-actividades :</p> <ul style="list-style-type: none"> • A3.1. Analizar y sintetizar los datos obtenidos con los instrumentos de recolección de información. • A3.2. Determinar la información sobre diagramas de valoración, los productos de trabajo y los descubrimientos para cada proceso valorado. • A3.3. Determinar los riesgos, las debilidades y las fortalezas de los procesos valorados. • A3.4. Determinar oportunidades de mejora para cada proceso valorado. • A3.4. Asignar valor a los atributos de los procesos para establecer el perfil del nivel de capacidad de los procesos valorados. • A3.6. Crear el Informe de Valoración.
RMP	<p>A4. Priorización de Procesos.</p> <p>El Responsable de mejora de procesos con la información presentada en el Informe de valoración prioriza las oportunidades de mejora encontrados de los procesos valorados y las puede organizar en casos de mejora, dependiendo de su complejidad. La priorización se puede hacer teniendo en cuenta diferentes enfoques, como por ejemplo: (i) basado en mejorar los procesos básicos y fundamentales para el adecuado funcionamiento de una pequeña organización como son los procesos de desarrollo de software y administración de proyectos, (ii) basado en mejorar procesos que den soporte a los objetivos del plan estratégico de la organización, (iii) basado en establecer la mejora continua de procesos siguiendo el enfoque de la gestión de procesos.</p> <p>Sub-actividades:</p> <ul style="list-style-type: none"> • A4.1. Analizar los objetivos de negocio de la organización. • A4.2. Analizar los objetivos de mejora del ciclo de mejora. • A4.3. Determinar la prioridad de las oportunidades de mejora teniendo en cuenta la información de los pasos anteriores.
RMP	<p>A5. Planificación Preliminar de Mejoras.</p> <p>El Responsable de mejora de procesos registra y planea las estrategias a seguir para abordar las oportunidades de mejora</p>

	<p>presentadas en el Informe de valoración con las cuales se guía las diferentes iteraciones de mejora que se llevarán a cabo dentro de la organización. El Responsable de mejora de procesos propone el número de iteraciones del actual ciclo de mejora y un borrador de la planificación general de éstas, además define el plan de medición de procesos, el plan de capacitación y el de manejo de riesgos del ciclo de mejora. El Responsable de mejora de procesos genera con los entregables de las tareas realizadas el producto de trabajo <i>Plan preliminar de mejora</i>.</p> <p>Sub-actividades:</p> <ul style="list-style-type: none"> • A5.1. Analizar y agrupar las oportunidades de mejora en casos de mejora. • A5.2. Planificar y documentar estrategias para implementar los casos de mejora mediante iteraciones. • A5.3. Describir el número de iteraciones del ciclo de mejora actual y realizar una planificación general de cada iteración. • A5.4. Crear el plan de medición de procesos, el plan de formación y el plan de gestión de riesgos para el ciclo de mejora.
--	---

- **Actividad:** FORMULACIÓN DE MEJORAS. Se planifica la iteración actual del ciclo de mejora y define la estrategia a seguir para mejorar el proceso seleccionado. De la primera iteración se obtiene una medida del esfuerzo de conducir esta iniciativa de mejora. Esta información se utiliza como base para la estimación del esfuerzo, costo, tiempo, recursos, entre otros, que demandarán las demás iteraciones del ciclo de mejora. La información relacionada con esta actividad y el aprendizaje adquirido en cada una de las iteraciones se registran en el *Plan de Implementación de Mejora*. Esta actividad puede realizarse una o varias veces en un ciclo de mejora.
- **Entradas:** *Plan General de Mejora*.
- **Salidas:** *Plan de Implementación de Mejora, Plan de Implementación de Mejora (actualizado), Plan General de Mejora (actualizado)*.
- **Tareas:**

Roles	Tareas
GMP	A3.1. A partir del Plan General de Mejora, escoger los casos de mejora que estén relacionados con un proceso a mejorar.
GMP y RP	<p>A3.2. Construir el Plan de Implementación de Mejora, a partir de la formulación del caso de mejora.</p> <ul style="list-style-type: none"> • Planificar, analizar y diseñar la mejora del proceso establecido para la iteración actual de mejora. • Utilizar información de iteraciones anteriores para la

	estimación del esfuerzo, costo y tiempo que demandará ésta nueva iteración del ciclo de mejora. <ul style="list-style-type: none"> • Documentar toda la información obtenida de las actividades anteriores.
RMP	A3.3. Verificar el Plan de Implementación de Mejora.
GGM	A3.4. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RMP	A3.5. Validar el Plan de Implementación de Mejora.
GGM	A3.6. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.

- **Actividad (Opcional):** PfmCOMPETISOFT Es un proceso para ayudar a las pequeñas empresas a formular y ejecutar las oportunidades de mejora encontradas en la actividad de diagnóstico de procesos. Para ello, sigue la filosofía del método ágil de gestión de procesos Scrum. El objetivo es aprovechar la sinergia de la valoración de procesos y encausarla hacia la iniciación de la realización efectiva de mejoras al interior de la pequeña organización software. En la figura 4.0 se muestra el desglose de trabajo de PfmCOMPETISOFT.

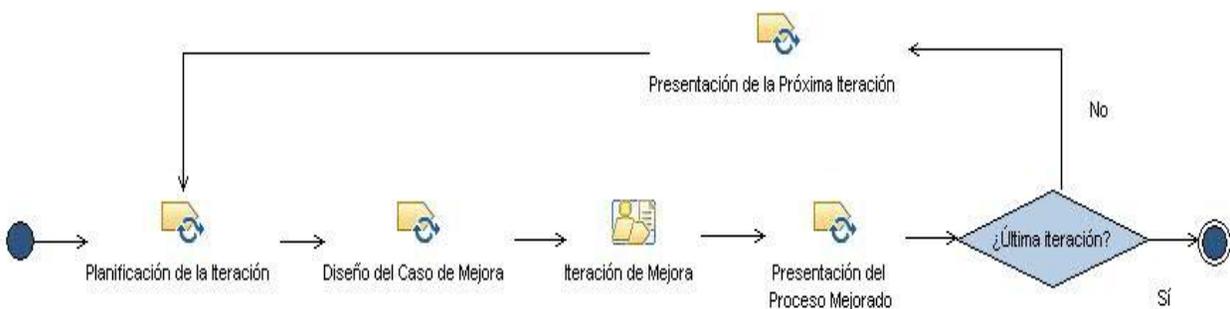


Figura 4.0 Desglose de las actividades de trabajo de PfmCOMPETISOFT

- **Entradas:** Oportunidades de Mejora Priorizadas.
- **Salidas:** Plan de Implementación de Mejoras, Informe de Iteración de Mejora.

- **Tareas:**

Roles	Tareas
GGM	<p>Planificación de la Iteración.</p> <p>Se describen los objetivos de la iteración y se revisan las oportunidades de mejora asociadas con los casos de mejora descritos en el <i>Plan General de Mejora</i>.</p>
GMP	<p>Diseño del Caso de Mejora.</p> <p>Se identifican las estrategias mediante las que satisfacer las oportunidades de mejora. Se definen las tareas asociadas a las estrategias para llevar a cabo la mejora. Se realiza una estimación (en horas o días) de esas tareas. Se definen las tareas y se asignan a los miembros del equipo de trabajo de la organización. Se llega a un acuerdo con el equipo de trabajo sobre el desarrollo de la iteración de mejora.</p>
	<p>Iteración de Mejora.</p> <p>Teniendo en cuenta el concepto de <i>sprint</i> de SCRUM, una iteración de mejora se considera un conjunto de actividades de mejora conducidas sobre un periodo predefinido (normalmente de una a cuatro semanas), el cual se utiliza para implementar y monitorizar las oportunidades de mejora. La mejora de procesos es una actividad que se realiza continuamente complementando las actividades de desarrollo de software. En una iteración debería realizarse una o dos reuniones cada semana, dependiendo del progreso y las necesidades de la iteración de mejora. En esas reuniones se ponen en común las lecciones aprendidas y los problemas encontrados, así como los resultados más importantes obtenidos en la iteración. Estas reuniones permiten que todo el equipo conozca el desarrollo de la iteración actual y lo tenga en cuenta para la iteración siguiente.</p>
RMP	<p>Presentación del Proceso Mejorado.</p> <p>Se presenta el proceso mejorado y se determina la manera en que debe institucionalizarse en toda la organización.</p>
RMP	<p>Presentación de la Próxima Iteración.</p> <p>Se anuncia la nueva iteración de mejora. El trabajo llevado a cabo en las tareas de la iteración se registra en el <i>Informe de Iteración de Mejora</i>.</p>

En la figura 5.0 se muestra un resumen con las actividades, roles y productos involucrados en las fases del proceso de SCRUM aplicado en PfmCOMPETISOFT.

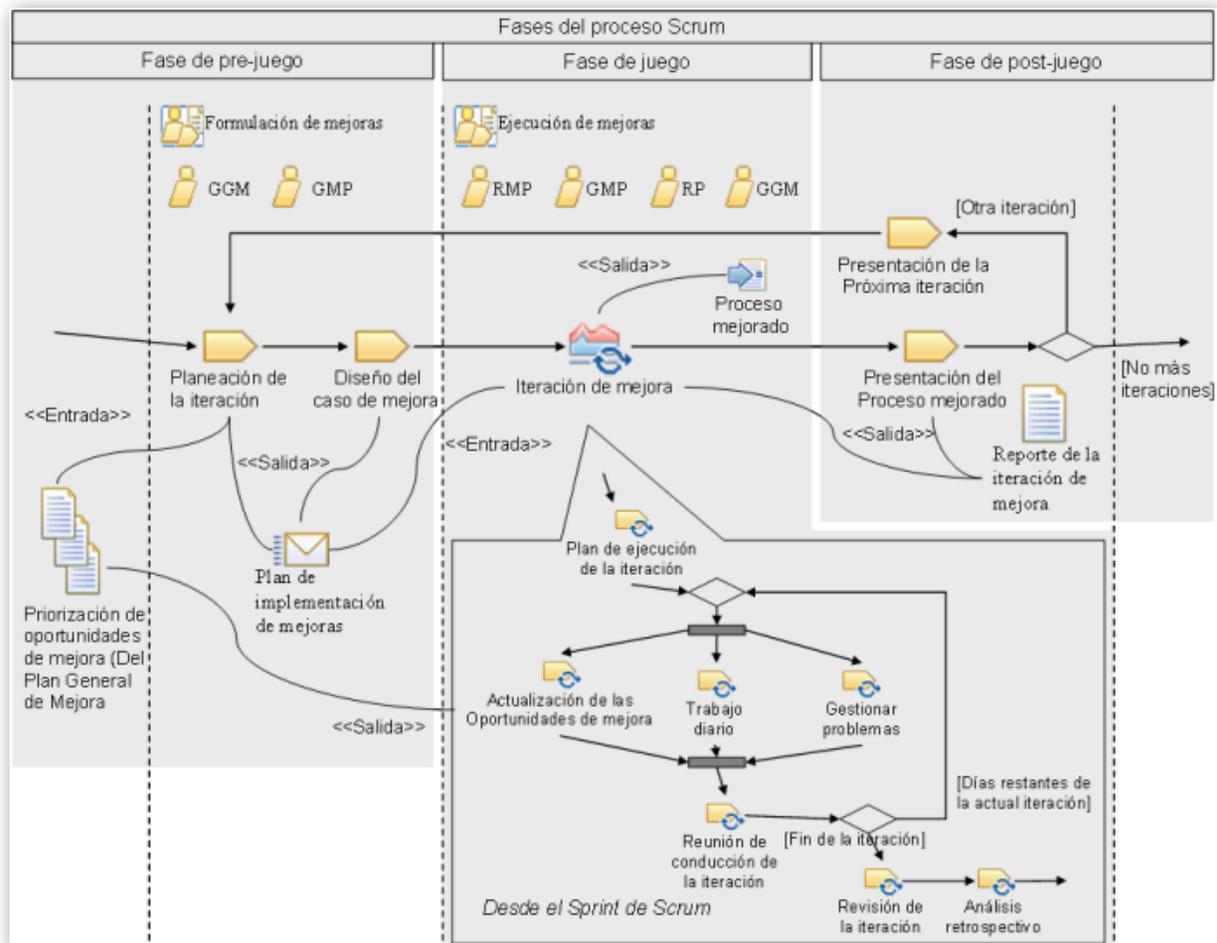


Figura 5.0 Fases de PfemCOMPETISOFT

- **Actividad:** EJECUCIÓN DE MEJORAS. Se gestionan y ejecutan los casos de mejora correspondientes a la iteración actual de acuerdo con los planes establecidos. Si la planificación de la iteración se ha desarrollado satisfactoriamente se aceptan e institucionalizan los nuevos procesos en la organización. La información relacionada con esta actividad se registra en el *Plan de Implementación de Mejora*. Esta actividad puede realizarse una o varias veces en un ciclo de mejora.
- **Entradas:** *Plan de Implementación de Mejora*.
- **Salidas:** *Informe de Iteración de Mejora, Plan General de Mejora (actualizado), Proceso Mejorado*.

• **Tareas:**

Roles	Tareas
GMP y RP	<p>A4.1. Ejecutar el caso de mejora formulado en la actividad anterior, descrito en el Plan de Implementación de Mejora.</p> <ul style="list-style-type: none"> • Ejecutar el plan que contiene la formulación del caso de mejora. • Aplicar en la organización el caso de mejora planeado y diseñado. • Registrar las mediciones de la actividad de mejora, establecidas para estimación del esfuerzo futuro tanto del proceso a mejorar como del ciclo de mejora en sí. • Evaluar las mejoras introducidas al proceso por el caso de mejora. • Medir y analizar el impacto en el proceso antes y después de ser aplicadas las mejoras al proceso.
GMP	<p>A4.2. Documentar las iteraciones de mejora y actualizar el Plan de Implementación de Mejora.</p> <ul style="list-style-type: none"> • Documentar las lecciones aprendidas en la iteración de mejora, como por ejemplo lecciones aprendidas, recursos utilizados, y demás información relevante para los siguientes casos de mejora que serán realizados en la siguiente fase a través de nuevas iteraciones de mejora. • Actualizar el Plan de Implementación de la Mejora con los resultados obtenidos de las iteraciones.
RMP	A4.3. Verificar el Plan de Implementación de Mejora.
GGM	A4.4. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A4.5. Validar el Plan de Implementación de Mejora.
GGM	A4.6. Corregir los defectos encontrados en el Plan de Implementación de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.

- **Actividad:** REVISIÓN DEL CICLO. Se corrigen o ajustan todos los elementos relacionados con la ejecución de cada una de las iteraciones de mejora. Se aplica nuevamente la evaluación interna de los procesos para cuantificar las mejoras que se han realizado. Al final se hace un análisis post-mortem del trabajo realizado en todo el ciclo de mejora. Se lleva a cabo una realimentación del ciclo de mejora realizado antes de volver a comenzar la fase de instalación de un nuevo ciclo. Las medidas desarrolladas para evaluar el cumplimiento de los objetivos, los procesos mejorados y las lecciones aprendidas se registran en el *Reporte de Mejora*. Si se trata del último ciclo de mejora se realiza un análisis general de todo el proyecto de mejora lo cuál queda también registrado en el *Reporte de Mejora*.
- **Entradas:** *Informe de Iteración de Mejora, Plan de Implementación de Mejora, Plan General de Mejora.*
- **Salidas:** *Plan de Mejora, Plan de Mejora (actualizado).* Base de conocimiento creada/actualizada. Recomendaciones para el siguiente ciclo de mejora. Lecciones aprendidas.
- **Tareas:**

Roles	Tareas
GGM y GMP	A5.1. Realizar realimentación. Recolectar, estudiar y evaluar todo tipo de experiencia o lección aprendida útil a futuros ciclos de mejora y así preparar mejor a las personas involucradas en posteriores ciclos
GMP	A5.2. Crear/Actualizar Base de Conocimiento. Poner a disponibilidad de la organización proyectos de mejora desarrollados.
GGM y RMP	A5.3. Analizar impacto de mejora. Desarrollar estudios comparativos en los cuales se pueda determinar si la mejora realizada ha disminuido o aumentado la eficiencia de los procesos de la empresa, si es el primer caso se deben estudiar y generar soluciones o estrategias que permitan equilibrar la balanza en relación a cuánto se desea sacrificar por mejorar los procesos.
GGM y RMP	A5.4. Analizar el compromiso. Evaluar el compromiso de la alta dirección y asegurarse de que el compromiso e interés de la gerencia todavía se mantiene para la financiación del próximo ciclo de mejora.
GGM y RMP	A5.5. Preparación del siguiente ciclo de mejora. Crear nuevos retos, estrategias, enfoques de mejora y caminos a seguir de acuerdo a la experiencia y objetivos alcanzados, que permitan actualizar y crear una propuesta de mejora más madura en la fase de Instalación del próximo ciclo, basándose en la experiencia y Lecciones Aprendidas.

GGM	A5.6. Verificar el Reporte de Mejora.
GGM	A5.7. Corregir los defectos encontrados en el Reporte de Mejora con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
GGM	A5.8. Validar el Reporte de Mejora.
GGM	A5.9. Corregir los defectos encontrados en el Reporte de Mejora con base en el Reporte de Validación y obtener la aprobación de las correcciones.

Roles

A continuación se muestra una tabla con la lista de roles, abreviaciones y descripción de las competencias requeridas.

	Roles	Abreviación	Competencia
1.	Responsable de Mejora de Procesos	RMP	Capacidad de liderazgo y de gestión. Conocimientos en mejora de procesos software. Capaz de diseñar procesos, guiar la implantación, hacer control de calidad al programa de mejora y entrenar al personal en los nuevos procesos definidos en el ciclo de mejora.
2.	Grupo de Mejora de Procesos	GMP	Conocimiento para planear e implantar la mejora de procesos específicos en la organización. Este grupo coordina, planea y ejecuta el ciclo de mejora de toda la organización.
3.	Grupo de Gestión de Mejora	GGM	Conocimiento para administrar y gestionar el ciclo de mejora. Constituido por un representante del Grupo Directivo (GD), un representante del Grupo de Mejora de Procesos (GMP) y el Responsable de Mejora de Procesos (RMP)
4.	Responsable de Proceso	RP	Conocimiento del proceso del cual es responsable. Es un administrador de conocimientos asociados al proceso, a su diagnóstico y mejora.
5.	Evaluador	EV	Conocimiento de la metodología y aplicación de la evaluación, valoración de procesos, así como análisis de los datos recogidos.

Descripción de los Productos

A continuación se presenta una lista de los productos necesarios en el proceso, descripciones y propiedades adicionales.

	Nombre	Descripción	Propiedades Adicionales
1.	Plan Estratégico de la Organización	<p>Contiene los siguientes apartados:</p> <ul style="list-style-type: none"> • Misión: Razón de ser de la organización. • Visión: Posición deseada de la organización en el mercado. • Valores: Cualidades y virtudes que se comparten entre los miembros de la organización y se desean mantener. • Análisis del entorno organizacional: Estudio de fortalezas y debilidades internas a la organización, por lo que es posible actuar directamente sobre ellas. Oportunidades y amenazas que son externas. • Objetivos: Resultados a buscar para cumplir con la Misión y Visión. • Estrategias: Forma de lograr los objetivos. • Procesos Requeridos: Identificación de los procesos con su propósito y objetivos para llevar a cabo las estrategias. • Cartera de Proyectos: Conjunto de proyectos externos e internos u oportunidades de proyectos. • Estructura de la Organización: Definición de áreas y responsabilidades de la organización requerida para llevar a cabo las estrategias. • Estrategia de Recursos: Definición, planificación y asignación de recursos en la organización para el cumplimiento de las estrategias, considerando los elementos de la Base de Conocimiento necesarios para el almacenamiento y consulta de la información generada en la organización. • Presupuesto: Gastos e ingresos esperados para un periodo determinado. • Periodicidad de Valoración: Definición de los periodos para realizar las revisiones de valoración y mejora. • Plan de Comunicación con el Cliente: Definición de los mecanismos para establecer los canales de comunicación con los clientes. 	

	Nombre	Descripción	Propiedades Adicionales
2.	Propuesta de Mejora	Documento que proporciona una introducción e iniciación a un ciclo de mejora, y contiene: objetivos de mejora generales, el proceso de mejora continua, alcance y metas globales del ciclo de mejora (modelo de referencia de procesos y método de evaluación) y recursos.	
3.	Plan General de Mejora	Documento constituido por dos partes: el <i>Informe de Valoración</i> y el <i>Plan Preliminar de Mejora</i> . El <i>Informe de Valoración</i> recopila: estado, análisis y priorización de los procesos. El <i>Plan Preliminar de Mejora</i> define: número de iteraciones, planeación general, plan de mediciones, plan de capacitación, plan de manejo de riesgos y cronograma.	Documento constituido por: <ul style="list-style-type: none"> • <i>Informe de Valoración</i>. • <i>Plan Preliminar de Mejora</i>.
4.	Informe de Valoración	Recopila la información de la planificación y ejecución de la valoración. Además, recopila el estado de los procesos con respecto a la capacidad al iniciar un ciclo de mejora, así como las oportunidades de mejora de los procesos valorados. Es un documento que describe los responsables y participantes en la valoración, la planificación y ejecución de la valoración, los hallazgos y el nivel de capacidad de los procesos.	
5.	Instrumentos de recolección de información	Este documento indica los instrumentos utilizados para recoger información, como, por ejemplo, entrevistas o encuestas.	
6.	Plan de Valoración	En este producto se recoge la información de planificación de la valoración.	
7.	Plan Preliminar de Mejora	Recopila las directrices de mejora para el trabajo a realizar en las fases siguientes al diagnóstico de procesos. Es un documento que describe el número de iteraciones del ciclo de mejora, la planificación general, el plan de mediciones, el plan de capacitación, el plan de manejo de riesgos y cronograma de las diferentes iteraciones del ciclo de mejora.	
8.	Oportunidades de Mejora Priorizadas	Resultado de priorizar las oportunidades de mejora descritas en el <i>Informe de Valoración</i> .	Prioriza las oportunidades de mejora del: <ul style="list-style-type: none"> • <i>Informe de Valoración</i>.
9.	Plan Implementación de Mejora	Documento que define las prácticas y actividades a seguir para crear, ejecutar e institucionalizar los casos de mejora; y contiene: la planeación específica	

	Nombre	Descripción	Propiedades Adicionales
		de la iteración, el registro de la ejecución de la mejora, las evaluaciones de los casos de mejora, la documentación de los procesos y el plan de aceptación e institucionalización.	
10.	Informe de Iteración de Mejora	Documento de cierre de una iteración de mejora, el cual contiene: procesos mejorados, evaluación inicial y final del proceso, esfuerzo realizado, logros obtenidos, lecciones aprendidas y recomendaciones para la iteración siguiente.	
11.	Oportunidades de Mejora Priorizadas	Resultado de priorizar las oportunidades de mejora descritas en el <i>Informe de Valoración</i> .	
12.	Proceso Mejorado	Documento en donde se asienta el resultado de la realización de las actividades de mejora, uno o varios procesos incrementan su nivel de capacidad.	
13.	Reporte de Mejora	Documento que cierra una iteración o ciclo de mejora; y contiene: procesos mejorados, evaluación inicial, evaluación final, esfuerzo involucrado, logros alcanzados, lecciones aprendidas, revisión postmortem y recomendaciones de ajuste al proceso de mejora. Este documento está constituido por los diferentes <i>Informes de Iteración de Mejora</i> .	

Plantillas

Ejemplos de plantillas

Listas de verificación

Herramientas

Referencia a otros estándares y modelos de referencia de software

Referencias Bibliográficas

Llave	Referencia
[Hanna et al., 2007]	Oktaba, H.; García, F., Piattini, M., Pino, F., Alquicira, C., Ruíz, F. "Software Process Improvement: The COMPETISOFT Project". <i>IEEE Computer</i> , October, 2007. Vol. 40(10), pp. 21-28.
[MMPCOMPETISOFT]	COMPETISOFT Modelo de Mejora de Procesos para pequeñas empresas, http://alarcos.inf-cr.uclm.es/Competisoft/framework/ , 28/05/06.
[Pino et al. 2008b]	Pino, F., F. García y Mario Piattini, (2008b). <i>Priorización de procesos como apoyo a la mejora de procesos en pequeñas organizaciones de software</i> . XXXIII Conferencia Latinoamericana de Informática, CLEI 2007, San José, Costa Rica, pp. 77.
[COMPETISOFT]	Oktaba, H., Piattini Mario., Pino, F., Orozco Julia y Alquicira Claudia, "COMPETISOFT Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos", editorial RA-MA, 2008.
[ISO/IEC12207]	ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes.
[ISO/IEC24765]	ISO/IEC 24765, Systems and Software Engineering Vocabulary

Forma de Evaluación

Paquete de despliegue – Modelo de Mejora de Procesos – Versión 0.1

Su opinión nos permitirá mejorar este paquete de despliegue, sus comentarios y sugerencias son bienvenidas

1. Está satisfecho con el contenido de este paquete de implementación?

Muy Satisfecho *Satisfecho* *Ni satisfecho ni insatisfecho* *Insatisfecho* *Muy Insatisfecho*

2. La secuencia y lógica en la que los temas se discuten fue fácil de seguir?

Muy Satisfecho *Satisfecho* *Ni satisfecho ni insatisfecho* *Insatisfecho* *Muy Insatisfecho*

3. ¿Qué tan satisfecho está con la APARIENCIA/FORMATO de este paquete de despliegue?

Muy Satisfecho *Satisfecho* *Ni satisfecho ni insatisfecho* *Insatisfecho* *Muy Insatisfecho*

4. Se tienen tópicos innecesarios? (por favor descríbalos)

5. ¿Qué tópico cree que le hace falta a este paquete de despliegue? (por favor descríbalos)

- Tópico propuesto:
- Justificación del nuevo tópico

6. Encontró errores en este paquete de despliegue?

- Por favor indique:
 - Descripción del error :
 - Localización del error (sección #, figura #, tabla #) :

7. Otras sugerencias o comentarios:

8. ¿Recomendaría usted este paquete de despliegue a otro colega/empleador?

Definitivamente *Probablemente* *No estoy seguro* *Probablemente no* *Definitivamente no*

Opcional

- Nombre: _____
- Dirección electrónica: _____