



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

DIAGNÓSTICO DE LA ORGANIZACIÓN Y PLANIFICACIÓN DEL “SISTEMA DE APOYO A UN PROCESO DE MEJORA, GENESIS”

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

ANDRES FLORES SANZ

**DIRECTORA DE LA TESIS: M. EN C. MARÍA GUADALUPE ELENA IBAR-
GÜENGOITIA GONZÁLEZ**

MÉXICO, D.F.

2009.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Gracias por el apoyo, soporte y confianza depositados en esta tesis a: Lupita Ibargüengoitia, Hanna Oktaba, Francisco Pino, Mario Piattini, Goyi y compañeros del grupo Alarcos de la Universidad Castilla-La Mancha.

Gracias a mis padres por estar conmigo y apoyarme durante mi maestría.

Gracias a mi compañera en este proyecto Maricela, por olvidar nuestras diferencias y usarlas a favor de esta tesis.

Gracias a todas las instituciones y proyectos que de forma directa o indirecta aportaron a este proyecto (CONACYT, DGEP, COMPETISOFT).

Gracias al posgrado y a aquellas personas que con su trabajo permiten que tengamos la oportunidad de materializar nuestras ideas.

Maricela Hernández Durán
Febrero 2009

Resumen

En los últimos años, la aplicación de iniciativas de mejora de procesos es utilizada por las PyMES's dedicadas al desarrollo de software, con esto pretenden incrementar la calidad del software desarrollado y así aumentar su competitividad ante la industria del software.

Debemos tomar en cuenta que los recursos humanos asignados por una PyME para la conducción de mejoras de procesos serán mínimos. Considerando la importancia de la disponibilidad del recurso humano para la mejora de procesos, es conveniente el uso de herramientas de software que brinden soporte directo al proceso de mejora.

Teniendo en cuenta esta perspectiva, en esta tesis se presentan parte de los módulos que integran el sistema de apoyo a un proceso de mejora: GENESIS. Estos módulos tienen como principal objetivo, el ofrecer soporte a la planificación del proceso de mejora y al diagnóstico de los procesos de la organización. Los otros módulos que complementan GENESIS son presentados en la tesis: "GENESIS, Sistema de apoyo a un proceso de mejora, gestión y revisión" por Maricela Hernández Durán.

Se tomo el proceso de mejora PmCompetiSoft como modelo de mejora guía para la planificación de las implementaciones de mejora.

En GENESIS se puede cargar cualquier modelo de referencia de procesos (CMMI, MoProSoft, CompetiSoft, etc.) que este modelado en SPEM mediante EPFComposer para la implementación de mejoras en los procesos de una organización.

Los módulos: Inicialización, Diagnóstico y Correspondencia presentados en esta tesis fueron finalmente integrados a GENESIS por lo que con esta versión se brinda soporte la implementación de mejoras en pequeñas organizaciones dedicadas el desarrollo de software.

Índice General

| | |
|---|-----------|
| AGRADECIMIENTOS | 3 |
| RESUMEN | 5 |
| ÍNDICE GENERAL | 7 |
| ÍNDICE DE FIGURAS | 9 |
| ÍNDICE DE TABLAS | 11 |
| INTRODUCCIÓN..... | 15 |
| CAPÍTULO 1..... | 21 |
| MEJORA DE PROCESOS DE SOFTWARE | 21 |
| 1.1 <i>BREVE INTRODUCCIÓN A LA MEJORA DE PROCESOS DE SOFTWARE</i> | 23 |
| 1.2 <i>EL USO DE HERRAMIENTAS DE APOYO A PROCESOS DE SOFTWARE</i> | 25 |
| 1.3 <i>MODELOS DE PROCESOS DE SOFTWARE</i> | 27 |
| 1.3.1 MOPROSOFT Y EL PROYECTO COMPETISOFT | 28 |
| 1.3.2 MODELO PARA LA IMPLEMENTACIÓN DE MEJORAS DE PROCESOS PMCOMPETISOFT | 33 |
| 1.3.2.1 DESCRIPCIÓN GENERAL..... | 33 |
| 1.3.2.2 ACTIVIDADES DEL PROCESO PMCOMPETISOFT | 36 |
| 1.3.2.3 ITERACIONES EN PMCOMPETISOFT | 38 |
| 1.3.2.4 EL PROCESO DE MEJORA PMCOMPETISOFT | 39 |
| CAPÍTULO 2..... | 43 |
| APLICACIONES RICAS DE INTERNET..... | 43 |
| 2.1 <i>LA EXPERIENCIA DEL USUARIO</i> | 44 |
| 2.2 <i>EVOLUCIÓN DE HERRAMIENTAS Y PATRONES</i> | 44 |
| 2.3 <i>DESARROLLO DE APLICACIONES RIA</i> | 46 |
| CAPÍTULO 3..... | 51 |
| GENESIS, SISTEMA DE APOYO A UN PROCESO DE MEJORA..... | 51 |
| 3.1 <i>ANTECEDENTES</i> | 53 |
| 3.2 <i>PLANTEAMIENTO GENERAL DEL PROBLEMA</i> | 56 |
| 3.3 <i>PROPUESTA DE SOLUCIÓN</i> | 57 |
| 3.4 <i>DESCRIPCIÓN GENERAL DE GENESIS</i> | 59 |
| 3.5 <i>COMPONENTES O MÓDULOS DE GENESIS</i> | 62 |
| 3.5.1 DIAGNOSTICO DE LA ORGANIZACIÓN Y PLANIFICACIÓN EN GENESIS | 65 |
| 3.5.1.1 DIAGNOSTICO Y PLANIFICACIÓN EN PMCOMPETISOFT | 66 |
| 3.5.2 GESTIÓN Y REVISIÓN DE LAS MEJORAS USANDO GENESIS | 69 |
| CAPÍTULO 4..... | 73 |
| ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE LOS MÓDULOS..... | 73 |
| 4.1 <i>EL PROCESO DE DESARROLLO</i> | 75 |
| 4.2 <i>FUNCIONALIDAD DE LOS MÓDULOS DESARROLLADOS</i> | 77 |
| 4.3 <i>ARQUITECTURA Y DISEÑO DE LOS MÓDULOS</i> | 85 |
| 4.3.1 DISEÑO DE LA BASE DE DATOS | 89 |
| 4.4 <i>CONSTRUCCIÓN Y PRUEBAS</i> | 94 |
| 4.5 <i>EPFCOMPOSER, OBTENCIÓN DE DATOS DEL MODELO DE REFERENCIA DE PROCESOS</i> | 94 |

| | | |
|---------------------------------------|--|------------|
| 4.5.1 | MODELADO DE PROCESOS | 94 |
| 4.5.2 | EPF COMPOSER | 95 |
| 4.5.3 | CARGA DE MODELOS DE REFERENCIA DE PROCESOS..... | 95 |
| CAPÍTULO 5..... | 99 | |
| UN PASEO POR GENESIS | 99 | |
| 5.1 | <i>REGISTRO DE RECURSOS Y USUARIOS.....</i> | <i>101</i> |
| 5.2 | <i>PLANIFICACIÓN DE MEJORAS</i> | <i>109</i> |
| 5.3 | <i>APLICACIÓN DE UNA VALORACIÓN RÁPIDA</i> | <i>118</i> |
| 5.4 | <i>CREACIÓN DE CASOS DE MEJORA.....</i> | <i>120</i> |
| CONCLUSIONES..... | 125 | |
| <i>RESULTADOS Y CONCLUSIONES.....</i> | <i>127</i> | |
| <i>CONTRIBUCIONES.....</i> | <i>128</i> | |
| <i>LIMITACIONES</i> | <i>128</i> | |
| <i>TRABAJO FUTURO.....</i> | <i>129</i> | |
| BIBLIOGRAFÍA | 133 | |

Índice de figuras

| | |
|--|-----|
| Figura 1: Herramientas de apoyo al proceso de software. _____ | 25 |
| Figura 2: El Proyecto COMPETISOFT: Diferentes iniciativas de varios países de América Latina se estudiaron para COMPETISOFT. _____ | 31 |
| Figura 3: Método de evaluación COMPETISOFT. Basado en EvalProSoft, el método define un conjunto de medidas para estimar la capacidad y el rendimiento de los procesos de software. _____ | 33 |
| Figura 4: Proceso de Mejora Continua PmCompetiSoft: Actividades, Iteraciones, Ciclos. _____ | 40 |
| Figura 5: Módulos de GENESIS y su interacción con las actividades del proceso PmCompetiSoft _____ | 62 |
| Figura 6: Flujo de trabajo para la fase de instalación de un ciclo de mejora en PmCompetiSoft _____ | 67 |
| Figura 7: Flujo de trabajo para la fase de diagnóstico de procesos en PmCompetiSoft _____ | 68 |
| Figura 8: Flujo de trabajo para la fase de formulación de mejoras en PmCompetiSoft _____ | 68 |
| Figura 9: Flujo de trabajo para la fase de ejecución de mejoras en PmCompetiSoft _____ | 69 |
| Figura 10: Flujo de trabajo para la fase revisión del ciclo de proceso de mejora PmCompetiSoft _____ | 69 |
| Figura 11: Resultados obtenidos durante el ciclo de vida del Proceso Unificado _____ | 76 |
| Figura 12: Diagrama General de casos de Uso. _____ | 81 |
| Figura 13: Casos de Uso de este proyecto. _____ | 82 |
| Figura 14: Diagrama del Caso de Uso: CU-01.1 _____ | 84 |
| Figura 15: Pantalla Datos de Usuarios. _____ | 84 |
| Figura 16: Diagrama de estado para el CU-01.1 Alta de usuarios. _____ | 85 |
| Figura 17: Diagrama de secuencia para el CU-01.1 Alta de usuario. _____ | 86 |
| Figura 18: Forma posible de distribución de los componentes de GENESIS. _____ | 88 |
| Figura 19: Diagrama de paquetes _____ | 89 |
| Figura 20: Tablas relacionadas al ambiente. _____ | 90 |
| Figura 21: Tablas relacionadas con el caso de mejora. _____ | 90 |
| Figura 22: Tablas relacionadas con los recursos. _____ | 91 |
| Figura 23: Tablas relacionadas con la implementación de mejora. _____ | 92 |
| Figura 24: Tablas relacionadas con la oportunidad de mejora. _____ | 93 |
| Figura 25: Pantalla para captura de Datos de Recurso Humano. _____ | 101 |
| Figura 26: Pantalla para la captura de Datos de Recurso Equipo. _____ | 102 |
| Figura 27: Pantalla para la captura de Datos de recurso espacio colaborativo. _____ | 103 |

| | |
|--|-----|
| Figura 28: Pantalla de Recursos del Sistema. | 104 |
| Figura 29: Pantalla de Administración de Recursos (Implementación). | 105 |
| Figura 30: Pantalla Editor de Consumibles. | 106 |
| Figura 31: Pantalla para los Datos de Recurso Dinero. | 107 |
| Figura 32: Pantalla para captura de Datos de Usuario. | 108 |
| Figura 33: Pantalla para la captura de los Datos de Ambiente. | 110 |
| Figura 34: Pantalla de la Biblioteca de Procesos Propios. | 110 |
| Figura 35: Pantalla Datos de Implementación y Navegador de Implementaciones. | 111 |
| Figura 36: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 1). | 112 |
| Figura 37: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 2). | 112 |
| Figura 38: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 3). | 113 |
| Figura 39: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 4). | 114 |
| Figura 40: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 5). | 115 |
| Figura 41: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 6). | 115 |
| Figura 42: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 7). | 116 |
| Figura 43: Contenido de la Implementación. | 117 |
| Figura 44: Contenido del Ciclo. | 117 |
| Figura 45: Contenido de la Iteración. | 118 |
| Figura 46: Contenido de la fase Formulación de Mejora. | 118 |
| Figura 47: Contenido de la fase Ejecución de Mejoras. | 118 |
| Figura 48: Pantalla del Cuestionario de Valoración. | 119 |
| Figura 49: Pantalla para captura de Datos de Valoración. | 120 |
| Figura 50: Pantalla Editor de Casos de mejora (creación del caso). | 121 |
| Figura 51: Pantalla Editor de Casos de mejora (cierre del caso). | 122 |

Índice de Tablas

| | |
|--|----|
| Tabla 1: Roles de PmCompetiSoft _____ | 34 |
| Tabla 2: Productos de Trabajo de PmCompetiSoft _____ | 36 |
| Tabla 3: Algunas tecnologías para el desarrollo de RIA _____ | 46 |
| Tabla 4: Resultados del Análisis realizado al proceso de mejora PmCompetiSoft. _____ | 77 |

Introducción

Las micro y pequeñas empresas (MyPE's)¹ dedicadas al desarrollo de software presentan en su mayoría escasa repetitividad en sus procesos (si es que cuentan con procesos formalmente establecidos), lo cual es consecuencia de su escasa madurez y se refleja en la calidad de sus productos generados. Aplicar una iniciativa de mejora de procesos es una estrategia actualmente utilizada por las MyPE's para desarrollar software con niveles aceptables de calidad y aumentar su competitividad ante la industria de software.

Es importante resaltar que los modelos de procesos más utilizados por la industria fueron diseñados para grandes empresas (1) y que no es factible que estos modelos sean adoptados e implementados por una MyPE (2). Estos modelos resultan costosos y agobiantes por la cantidad de recursos que se debe destinar (3) (4) y por la documentación que debe ser generada durante una iniciativa de mejora (5).

Tanto las iniciativas de mejora como las herramientas se han tenido que ir adaptando a las necesidades de las MyPE's, lo cual ha motivado la aparición de diferentes propuestas enfocadas a la mejora de procesos acordes a las necesidades de este tipo de empresas.

De acuerdo a lo anterior, surge la siguiente cuestión, ¿Cómo puede una MyPE iniciar un proyecto de mejora y concluirlo con éxito? Es una pregunta interesante y que no es fácil de resolver. Un factor que puede ayudar a una MyPE a conducir con éxito una iniciativa de mejora de procesos, es ofrecer apoyo tecnológico, mediante herramientas de software que permitan automatizar, guiar, aplicar y gestionar modelos y/o métodos de procesos, evaluación ó mejora orientados a MyPE's.

Debemos tomar en cuenta que los recursos humanos asignados por una MyPE a la conducción de las iniciativas de mejora serán mínimos y además, en la mayoría de los casos estas personas deberán continuar con sus tareas propias dentro de la organización (6).

Considerando la importancia de la disponibilidad del recurso humano para la mejora de procesos, es conveniente utilizar herramientas de software que den soporte a este proceso de mejora, con el fin de reducir la carga

¹ Empresas que pueden tener desde 1 hasta 50 empleados.

cognitiva de las personas involucradas en la iniciativa de mejora y gestionar adecuadamente la carga administrativa asociada con su aplicación.

Otro factor importante a considerar es la importancia de guiar a una MyPE durante el proceso de mejora, lo cual puede resolverse con un asistente software que guíe paso a paso al responsable de las mejoras, indicando los pasos a seguir durante la aplicación y gestión de dicha iniciativa.

Además, una iniciativa de mejora involucra actividades específicas, como administración del proceso de mejora, administración del conocimiento y trazabilidad de las mejoras realizadas, entre otras, que no son soportadas por herramientas tradicionales de administración de proyectos.

Teniendo en cuenta la perspectiva descrita anteriormente, en esta tesis se presentan los 3 módulos base que integran el sistema de apoyo a un proceso de mejora llamado GENESIS, sistema que fue diseñado para apoyar a los responsables de la conducción de las mejoras dentro de una organización mediante la automatización de tareas realizadas durante este proceso. Este sistema ofrece soporte al proceso de mejora PmCompetiSoft, el cual forma parte del framework de mejora de COMPETISOFT.

Este trabajo recibió el apoyo de expertos en PmCompetiSoft durante una estancia de investigación realizada en la Universidad de Castilla La Mancha, en Ciudad Real, España y ha sido financiado por los proyectos COMPETISOFT (506AC0287) del programa CYTED, MECENAS (PBI06-0024) e INGENIO (PAC08-0154-9262) de la Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia y ESFINGE (TIN2006-15175-C05-05) del Ministerio de Ciencia y Tecnología (España), por el Proyecto Institucional de Apoyo a los Estudios de Posgrado PAEP, UNAM y por el Programa de Becas Mixtas del CONACYT.

GENESIS está integrado en total por 6 módulos. 3 de los módulos que integran el sistema son presentados en la tesis "GENESIS, sistema de apoyo a un proceso de mejora: gestión y revisión" por Maricela Hernández Durán. Dichos módulos contienen soluciones referentes a la gestión del proceso de mejora y a la revisión de los datos generados durante las fases de proceso de mejora.

Por otro lado, el propósito de esta tesis es presentar las soluciones destinadas a la planificación y organización del proceso de mejora. De esta

forma, en esta tesis se presentan los módulos: Inicialización, Diagnostico y Correspondencia que integran una parte del sistema GENESIS.

El contenido de esta tesis está distribuido en 5 capítulos los cuales se describen brevemente a continuación:

En el primer capítulo se presenta un panorama general de la mejora de procesos de software, una vista general del proyecto COMPETISOFT y de su framework de mejora PmCompetiSoft, enfocándonos principalmente en la estructura del proceso de mejora PmCompetiSoft.

En el capítulo 2 se presentan conceptos que giran alrededor de las Aplicaciones Ricas de Internet y las tecnologías que permiten el desarrollo de aplicaciones de este tipo.

El capítulo 3 describe los antecedentes y el planteamiento general del problema, a partir de los cuales se presenta una propuesta de solución. Incluye también la descripción general y las características de los 6 módulos que integran el sistema de apoyo a un proceso de mejora GENESIS.

En el capítulo 4 se detalla la forma en que se desarrolló la propuesta de solución, los requerimientos de funcionalidad deseados para gestionar y revisar el proceso de mejora PmCompetiSoft, la arquitectura de los módulos y el diseño de la base de datos. Se encuentra también en este capítulo la descripción de la forma en que se obtienen los datos del modelo de referencia de procesos modelado en SPEM (7).

El capítulo 5 contiene un vistazo a la aplicación y la interfaz usada para el alta de recursos y usuarios, la realización de la planificación de mejoras, una valoración rápida y la creación de casos de mejora.

Finalmente se presentan los resultados, conclusiones, contribuciones, limitaciones y trabajo futuro de esta tesis.

Capítulo 1

MEJORA DE PROCESOS DE SOFTWARE

La mejora del proceso de software (Software Process Improvement - SPI) es el concepto que agrupa las iniciativas de mejora de calidad en aquellas organizaciones que desarrollan o utilizan intensivamente software. La adopción de prácticas de calidad durante el desarrollo de software puede conseguir un proceso de desarrollo de software sin fallos.

1.1 BREVE INTRODUCCIÓN A LA MEJORA DE PROCESOS DE SOFTWARE

La mejora de procesos está basada en el supuesto de que la calidad del proceso de desarrollo es crítica para la calidad del producto, de esta forma, se identifican cuatro factores que afectan la calidad de un producto de software (8):

- La tecnología de desarrollo.
- La calidad del proceso.
- La calidad del personal.
- El costo, tiempo y duración.

Dependiendo del tamaño del proyecto es que influirán estos factores, pero el determinante principal de la calidad del producto es el proceso de software. Es importante que organizaciones o equipos pequeños cuenten con una buena tecnología de desarrollo ya que el equipo no puede dedicar mucho tiempo a procedimientos administrativos tediosos, por lo que el uso de herramientas afectan fuertemente su productividad.

Los procesos de software pueden catalogarse dependiendo del grado de formalización, de los tipos de productos desarrollados y del tamaño de la organización, entre otros, pero básicamente pueden catalogarse en 4 clases (8):

- Procesos informales: En estos procesos no existe un modelo de procesos definido de forma estricta. Los procesos informales pueden utilizar procedimientos formales, pero los procedimientos a utilizar son definidos por el equipo de desarrollo.
- Procesos gestionados: El proceso de desarrollo es dirigido por un modelo de procesos, este modelo define los procedimientos, su agenda y las relaciones entre los procedimientos.

- Procesos metodológicos: En estos son utilizados algún o algunos métodos de desarrollo definidos, y son beneficiados con el uso de herramientas CASE².
- Procesos de mejora: son procesos que tienen inherentemente el objetivo de mejora. Existe un presupuesto específico para estos procesos de mejora, y de procedimientos para introducir tales mejoras, como parte de estas mejoras se introducen mediciones cuantitativas del proceso.

La calidad de un producto de software desarrollado parece estar determinada principalmente por la calidad del proceso de desarrollo utilizado. Por lo tanto las organizaciones de software necesitan mejorar sus procesos de desarrollo para mejorar sus productos de software.

La mejora de procesos implica entender los procesos existentes dentro de una organización y realizar modificaciones en los mismos para lograr una mejor calidad de los productos o bien reducir costos y tiempos.

Los procesos de software tienen atributos y características, dependiendo de la complejidad y magnitud del proceso. Su estructura puede estar conformada por varios niveles de abstracción, en los que los subprocesos de más bajo nivel se descomponen finalmente en actividades o tareas. No es posible realizar mejoras de procesos que optimicen todos los atributos del proceso en forma simultánea.

Es importante notar que la mejora de procesos no solo es adoptar métodos o herramientas particulares o utilizar un modelo de referencia de procesos en lugar de otro. La mejora de procesos debe verse como una actividad específica de la organización ó como parte de la misma. La mejora de procesos es una actividad cíclica, a largo plazo y continua, en la que se introducen nuevos procesos, el entorno del negocio cambia y los procesos por sí mismos evolucionan.

² Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre muchas otras.

1.2 EL USO DE HERRAMIENTAS DE APOYO A PROCESOS DE SOFTWARE

La mayoría de los procesos de software tienen ahora el apoyo de herramientas CASE ó de herramientas especializadas que brindan soporte automático o semiautomático por lo menos en alguna fase del proceso, por lo que son procesos con soporte.

La efectividad del uso de herramientas de apoyo en un proceso de software dependerá del tipo de herramienta y de la clasificación del proceso utilizado. (Ver Figura 1).

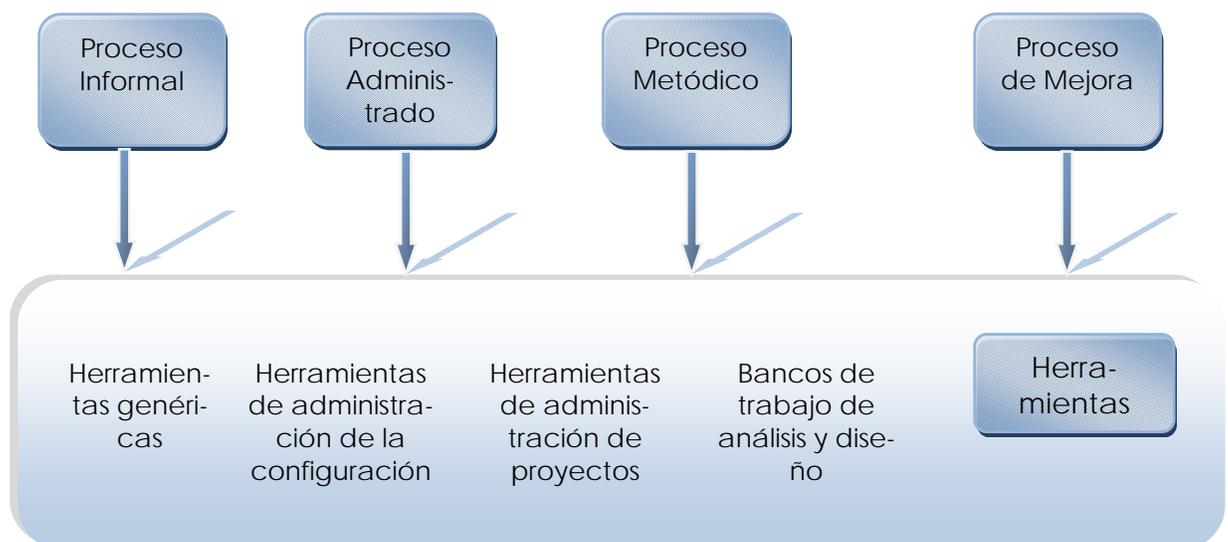


Figura 1: Herramientas de apoyo al proceso de software.

Las herramientas utilizadas podrían clasificarse ya sea por su funcionalidad, las fases del ciclo de vida del desarrollo que cubren, o por la arquitectura de las aplicaciones que producen, con lo que contamos con una gran gama de herramientas que de acuerdo a su funcionalidad apoyan ya sea en una o en varias fases de un proceso de software.

Una herramienta de apoyo a procesos de software puede contar con alguna(as) de las siguientes funcionalidades:

- Modelado de negocio.
- Modelado y administración de procesos.
- Planificación y administración de proyectos.
- Análisis de riesgos.
- Seguimiento de requerimientos.
- Desarrollo y diseño de interfaz.
- Generación de prototipos.
- Análisis y diseño de software.
- Control de métricas y gestión.
- Producción de documentación.
- Control de calidad.
- Codificación de cuarta generación.
- Mantenimiento de software.
- Gestión de configuración de software.
- Integración y comprobación.
- Análisis estático o dinámico.
- Reingeniería o ingeniería inversa.

El uso de estas herramientas busca principalmente automatizar tareas, mejorar la gestión y dominio sobre el proyecto, minimizar costos (tiempo y recursos), mejorar la productividad durante las distintas fases del desarrollo de software, facilitar la utilización de metodologías y mejorar la calidad del software desarrollado.

No es fácil la tarea de elegir una herramienta de apoyo, ya que más que catalogarlas como mejor o peor una respecto a otra, hay que tomar en cuenta el propósito y funcionalidad de la herramienta de apoyo, pero

sobre todo el tipo de proceso al que se dará soporte con dicha herramienta.

1.3 MODELOS DE PROCESOS DE SOFTWARE

El Software Engineering Institute (SEI) organismo de la Universidad Carnegie Mellon, fue establecido para mejorar las capacidades de la industria de software en los Estados Unidos de América. A mediados de los años ochenta, el SEI inicio un estudio de las formas de evaluar las capacidades de los proveedores de software.

Como resultado de estos estudios se creó el Modelo de Madurez de la Capacidad de Software SEI (9) (CMM), el cual ha influido considerablemente a que la mejora de procesos sea considerada por la comunidad de ingeniería del software. Posteriormente el SEI creó un nuevo programa, en el cual desarrollo un Modelo de Capacidad Integrado (CMMI) (10).

La Organización Internacional para la Estandarización (ISO, International Organization for Standardization) es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.

Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional. ISO desarrollo las normas: ISO 12207:2002 (11) Information Technology / Software Life Cycle Processes, que es el estándar para los procesos de ciclo de vida del software de la organización ISO; ISO 15504:1998, ISO/IEC 15504 (12) el cual es el Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software, además de la norma ISO 9001:2000 (13) que especifica los requisitos para un sistema de gestión de la calidad, ésta norma también puede ser aplicada al desarrollo de software.

Estos modelos fueron diseñados y creados para grandes organizaciones de software (14) (15), y hay una tendencia generalizada por parte de la comunidad de ingeniería de software a resaltar que los programas de mejora de procesos de software (SPI, Software Process Improvement), no son exitosos con pequeñas organizaciones, debido a la estructura de este tipo de

organizaciones y al costo que implica la adopción de algunos de estos modelos.

A partir de los años noventa la comunidad de ingeniería del software incremento su interés hacia la mejora de procesos, y sobre todo hacia la mejora de procesos enfocados a PyMe's. Esto se ve reflejado por la aparición de nuevas iniciativas relacionadas con SPI enfocadas a MyPE's y PyME's (Pequeñas y medianas organizaciones). Dentro de las nuevas propuestas podemos mencionar: MoProSoft (16), IDEAL (17), MR-MPS (18), RAMALA (19), MESOPYME (20), Agile-SPI (21), IMPACT (22), COMPETISOFT (23).

1.3.1 MOPROSOFT Y EL PROYECTO COMPETISOFT

MoProSoft es la denominación del "Modelo de Procesos para la Industria del Software" desarrollado por la Asociación Mexicana para la Calidad en Ingeniería del Software (AMCIS) a través de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM) por encargo de la Secretaría de Economía.

La Secretaría de Economía, dentro del Programa para el Desarrollo de la Industria del Software (PROSOFT) (24) , que forma parte del Plan Nacional de Desarrollo 2001-2006, identificó como modelos referentes para la calidad en el desarrollo y mantenimiento de software a CMMI e ISO 15504, pero al analizarlos para su inclusión en el plan, el comité designado consideró que son demasiado formalistas o pesados para la mayoría de las empresas mexicanas. Por lo que se decidió la elaboración de un modelo adecuado a las necesidades de las empresas mexicanas, pero que se basaría en los modelos evaluados previamente.

Se propuso que la norma estuviera dividida en tres partes:

- Modelo de procesos (¿Qué procesos?).
- Modelo de capacidades de procesos (¿Qué evaluar?).
- Método de evaluación (¿Cómo evaluar?).

La primera versión se terminó de elaborar en diciembre de 2002, en la cual se desarrolló el Modelo de Procesos para la Industria de Software (MoProSoft). En Mayo de 2003, fue terminada la versión 1.1, y el 15 de Agosto de 2005 fue publicada en el diario oficial de la federación (25) la declaratoria de la norma Mexicana NMX-059/01-NYCE-2005 referente a MoProSoft.

MoProSoft identifica los procesos empleados por las pequeñas y medianas organizaciones de software y los clasifica en tres categorías:

- Categoría de alta dirección (DIR): Aborda las prácticas de la alta dirección relativas a la gestión del negocio. Proporciona alineación a los procesos de la categoría de gerencia (GER) y se retroalimenta de la información que éstos generan. Dentro de esta categoría se encuentra el proceso de Gestión de negocio.
- Categoría de Gerencia (GER): Aborda las prácticas de gestión de procesos, proyectos y recursos en función de las alineaciones establecidas a través de los procesos de alta dirección (DIR). Proporciona los elementos para el funcionamiento de los procesos de la siguiente categoría (Operación), recibe y evalúa la información que generan, y comunica los resultados a los procesos de alta dirección. En esta categoría están los procesos: Gestión de procesos, Gestión de proyectos y Gestión de recursos.
- Categoría de Operación (OPE): Aborda las prácticas para los proyectos de desarrollo y mantenimiento de software. Los procesos de esta categoría realizan las actividades de acuerdo con los elementos proporcionados por los de gerencia, y remite a ésta la información y los productos generados. En esta categoría están los procesos de Administración de proyectos específicos y Desarrollo y mantenimiento de software.

Además establece y emplea un patrón para definir cada proceso. El patrón de procesos es una agrupación esquemática de los elementos que configuran un proceso. Está formado por tres partes: Definición general del proceso, Prácticas y Guías de ajuste.

En el año 2005, varios investigadores y profesionales Iberoamericanos, reconocieron la importancia de contar con un marco de mejora y certificación para las pequeñas organizaciones de software. Por lo que propusieron COMPETISOFT al Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED).

Para desarrollar el proyecto COMPETISOFT, se estudiaron diferentes iniciativas de América Latina, como MoProSoft, el Modelo de Mejora de procesos brasileño MPS.BR y también el proceso ágil para la mejora de procesos de software Agile SPI. El Ministerio español de Administraciones Públicas también consideró Métrica v3, ya que tiene un enfoque encaminado a mejorar los procesos de software (*Ver Figura 2*).

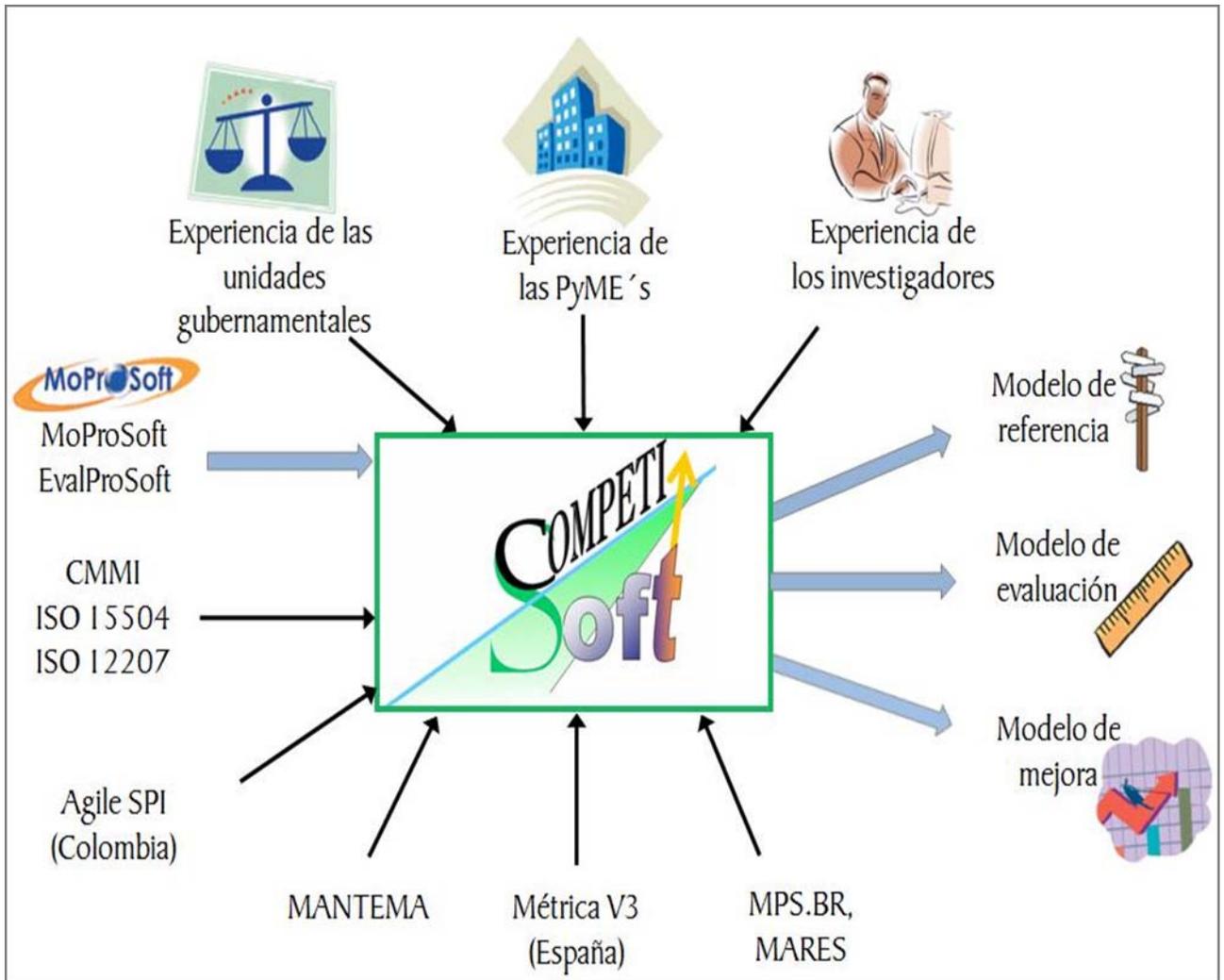


Figura 2: El Proyecto COMPETISOFT: Diferentes iniciativas de varios países de América Latina se estudiaron para COMPETISOFT.

COMPETISOFT define un *Modelo de Referencia de Procesos*, un *Método de Evaluación*, y define también un *Modelo de Mejora*.

El *Modelo de Referencia de Procesos* de COMPETISOFT está basado en el definido por MoProSoft. Tiene tres categorías de procesos: Alta Dirección, Gerencia y Operación que reflejan la estructura de una organización:

- La categoría de Alta Dirección contiene el proceso de Gestión de Negocio.
- La categoría de Gerencia está integrada por los procesos de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos. Éste último

está constituido por los subprocesos de Gestión de Recursos Humanos, Gestión de Bienes, Servicios e Infraestructura y Gestión de Conocimiento.

- La categoría de Operación está integrada por los procesos de Administración de Proyectos Específicos y de Desarrollo y Mantenimiento de Software.

Contiene también un patrón de procesos el cual es un esquema de elementos que servirá para la documentación de los procesos. Éste patrón está constituido por tres partes: Definición general del proceso, prácticas y guías de ajuste.

El *Método de Evaluación* define una guía para la ejecución de la actividad de evaluación. El proceso de evaluación es un elemento integrador de los componentes de COMPETISOFT, este proceso guía las actividades de evaluación a lo largo de todo el proyecto de mejora. Esta guía permite abordar la evaluación de los procesos de la organización de dos formas:

- Evaluación rápida: valoración de procesos internos con fines de diagnóstico inicial.
- Evaluación profunda: mejoramiento y verificación de mejoras antes de pasar a una entidad certificadora.

El *Modelo de Mejora* de COMPETISOFT está basado en Agile SPI (21), formando parte del marco metodológico de COMPETISOFT (Junto con el *Método de Evaluación* y el *Modelo de Referencia de Procesos*) (Ver figura 3).

Este modelo de mejora define un proceso para guiar la mejora continua de procesos denominado PmCompetiSoft. Este proceso cumple con el patrón de procesos establecido por el proyecto COMPETISOFT.

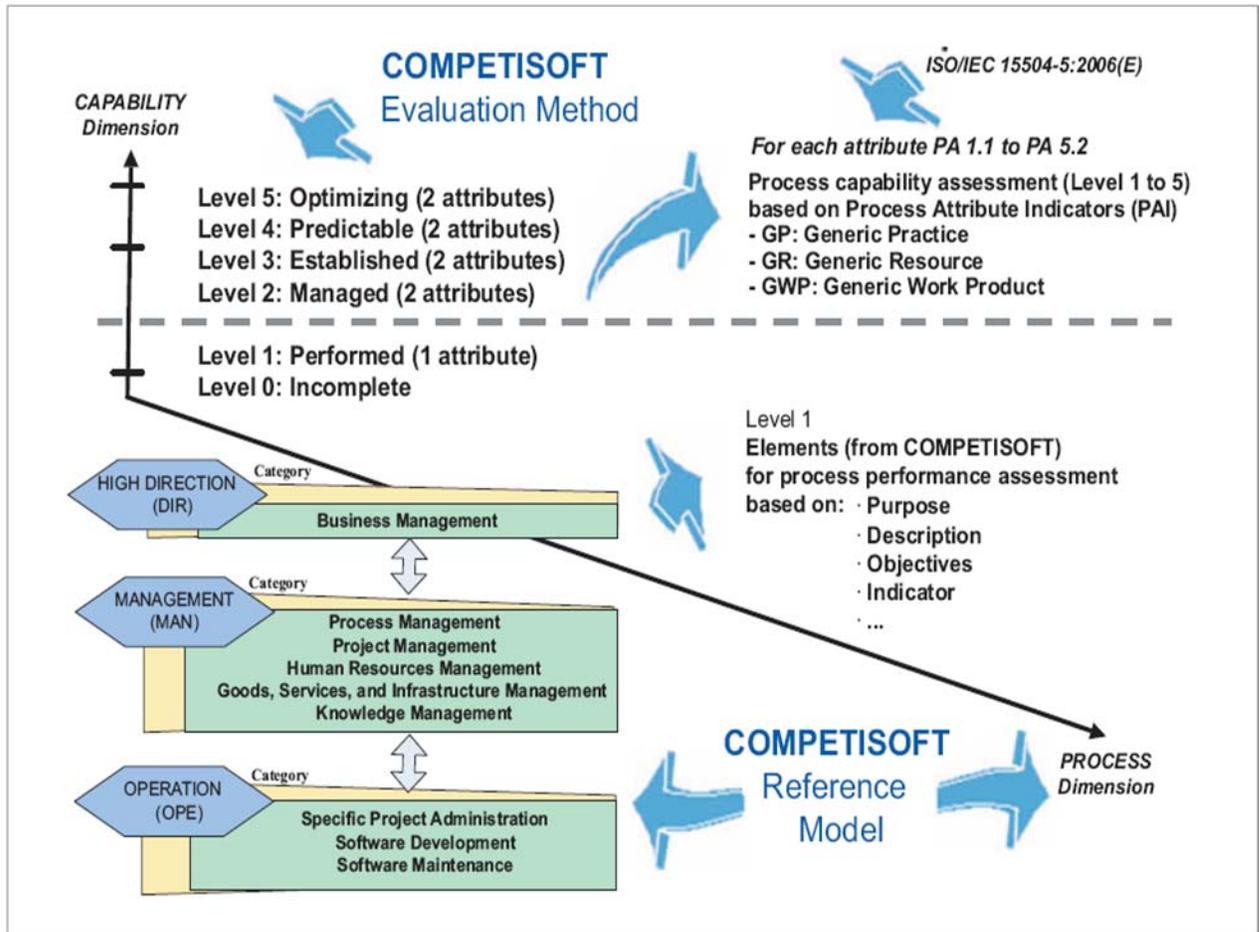


Figura 3: Método de evaluación COMPETISOFT. Basado en EvalProSoft, el método define un conjunto de medidas para estimar la capacidad y el rendimiento de los procesos de software.

1.3.2 MODELO PARA LA IMPLEMENTACIÓN DE MEJORAS DE PROCESOS PMCOMPETISOFT

1.3.2.1 DESCRIPCIÓN GENERAL

PmCompetiSoft es un proceso ligero de mejora de procesos de software que guía la ejecución de un ciclo de mejora de procesos de software en pequeñas y medianas organizaciones. Es un proceso liviano y sigue un enfoque iterativo e incremental (26) (27).

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 de software enfocada en las pequeñas organizaciones a través

de la definición de una guía para implementar paso a paso la mejora de procesos (28).

Los objetivos de PmCompetiSoft definidos son:

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

Este proceso define 5 roles involucrados los cuales se muestran en la siguiente tabla:

Tabla 1: Roles de PmCompetiSoft

| NOMBRE | ROL | COMPETENCIAS |
|---------------|--|---|
| RMP | <i>Responsable de Mejora de Procesos</i> | Capacidad de liderazgo y de gestión. Conocimientos en mejora de procesos software. Capaz de diseñar procesos, guiar la implementación y entrenar al personal en los nuevos procesos definidos en el programa de mejora. |
| GMP | <i>Grupo de Mejora de Procesos</i> | Conocimiento para planear e implantar la mejora de procesos específicos en la organización. Este grupo coordina, planea y ejecuta el proyecto SPI de toda la organización. |
| GGM | <i>Grupo de Gestión de Mejora</i> | Conocimiento para administrar y gestionar el proyecto de mejora. Constituido por un representante del grupo directivo, un representante del grupo de mejora de procesos y el responsable de mejora de procesos. |
| RP | <i>Responsable de Proceso</i> | Conocimiento del proceso del cual es responsable. Es un administrador de conocimientos asociados al proceso, a su diagnóstico y mejora. |

| | | |
|----|------------------|--|
| EV | <i>Evaluador</i> | Conocimiento en la metodología y aplicación de evaluación de procesos. |
|----|------------------|--|

Este proceso define también productos de trabajo, para lo cual se cuenta con una plantilla auto-contenida. En cada producto de trabajo se registra el esfuerzo de realizar las tareas asociadas con cada actividad y relacionadas con dichos productos:

Tabla 2: Productos de Trabajo de PmCompetiSoft

| NOMBRE DEL PRODUCTO | DESCRIPCIÓN |
|--|--|
| <i>Propuesta de Mejora.</i> | 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. |
| <i>Plan General de Mejora.</i> | Documento constituido por dos partes: el Informe de Valoración y el Plan Preliminar de Mejora. El Informe de Valoración recopila: estado, análisis y priorización de los procesos. El Plan Preliminar de Mejora define: número de iteraciones, planeación general, plan de mediciones, plan de capacitación, plan de manejo de riesgos y cronograma. |
| <i>Plan de Implementación de Mejora.</i> | Documento que define las prácticas y actividades a seguir para crear, ejecutar e institucionalizar los casos de mejora, y contiene: la planeación específica 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. |
| <i>Reporte de mejora.</i> | Documento que cierra una iteración ó ciclo de mejora; y contiene: procesos mejorados, evaluación inicial, evaluación final, esfuerzo involucrado, logros alcanzados, lecciones aprendidas, revisión post-mortem y recomendaciones de ajuste al proceso de mejora. |

Con el fin de orientar a la persona responsable de la mejora del proceso en una pequeña organización, se incluye también una “Guía del consultor”, en donde se indica cómo utilizar el proceso PmCompetiSoft para iniciar y establecer ciclos de mejora.

1.3.2.2 ACTIVIDADES DEL PROCESO PMCOMPETISOFT

El proceso de mejora continua de procesos se compone de uno o más ciclos de mejora y cada ciclo de mejora consta de 5 Fases o Actividades, las cuales se definen a continuación:

- **Instalación del Ciclo**, esta es la actividad de partida para el ciclo de mejora. En esta actividad se crea una propuesta de mejora basada en las necesidades del negocio, la cual guiará a la organización a través de cada una de las actividades siguientes.

Esta propuesta debe ser aprobada por la alta gerencia para garantizar la asignación de los recursos necesarios involucrados en el ciclo de mejora. Se definen los objetivos de mejora, los cuales son establecidos a partir de las necesidades de la organización.

También se debe estructurar una infraestructura de gestión, la cual describe la manera en la cual se organizan las personas comprometidas dentro del ciclo de mejora. Esta infraestructura organiza el ciclo de mejora teniendo en cuenta al Responsable de Mejora de Procesos (RMP), al Grupo de Mejora de Procesos (GMP), al Grupo de Gestión de Mejora (GGM), al Responsable de Proceso (RP) y al Evaluador (EV).

- **Diagnóstico de Procesos**, en esta actividad ya se ha iniciado un ciclo de mejora. Se realizan actividades de valoración de procesos software para saber cuál es el estado general de los procesos de la empresa y así obtener Oportunidades de Mejora³.

Además, se realiza un análisis de los resultados de la valoración con el fin de establecer la prioridad de los Casos de Mejora⁴. Este análisis sirve para crear el plan general de mejora.

- **Formulación de Mejoras**, en ésta actividad se toman los casos de mejora de mayor prioridad, según los resultados arrojados por la valoración de procesos hecha en la actividad de diagnóstico y se realiza la planificación de una primera (ó nueva) Iteración de Mejora⁵.

El objetivo en un principio es realizar una medida del esfuerzo que sirva de base para la estimación del esfuerzo que tomará llevar a cabo el resto del ciclo de mejora. Se realizan las planificaciones correspondientes a las diferentes iteraciones que pueden resultar con cada uno de los casos de mejora definidos.

- **Ejecución de Mejoras**, en esta actividad se ejecuta los casos de mejora, basados en la estimación hecha en la fase anterior, y se gestiona

³ Una Oportunidad de Mejora es una desviación o implementación insuficiente de una práctica de Ingeniería de Software en la organización.

⁴ Un Caso de Mejora permite gestionar la complejidad de las oportunidades de mejora encontradas en el proyecto de mejora que se ejecuta en la organización. Un caso de mejora agrupa oportunidades de mejora concretas de acuerdo a los objetivos de mejora de la organización.

⁵ Una Iteración de Mejora es un incremento pequeño y concreto de mejora que permite adelantar el desarrollo y gestión de un Caso de Mejora de manera independiente.

todo el esfuerzo involucrado en éstos. Se genera un documento donde se registra la ejecución de los pilotos de prueba, la evaluación de la mejora por la introducción de los nuevos procesos y por el perfeccionamiento de los procesos ya existentes.

Si los planes piloto se han desarrollado satisfactoriamente se definen planes de aceptación e institucionalización de los nuevos procesos en la organización.

- **Revisión del Ciclo**, en esta actividad se hace un análisis post-mortem del ciclo de mejora ejecutado antes de volver a comenzar la fase de inicio de un nuevo ciclo. En esta fase todas las lecciones aprendidas y las métricas desarrolladas para medir el cumplimiento de los objetivos sirven como base de conocimiento o fuente de información para las personas involucradas en el siguiente ciclo de mejora.

Con toda la información recolectada se debe evaluar el trabajo realizado y se deben corregir o ajustar todos los elementos relacionados con la ejecución del ciclo de mejora, como la infraestructura establecida, los métodos utilizados y los canales de comunicación, entre otros.

1.3.2.3 ITERACIONES EN PMCOMPETISOFT

Las Iteraciones de Mejora de PmCompetiSoft involucran a las Actividades de **Diagnóstico de Procesos, Formulación de Mejoras y Ejecución de Mejoras**. Las Iteraciones de Mejora constan de 3 momentos:

- La pre-mejora: que hace referencia a actividades de diagnóstico y planeación.
- La mejora: que hace referencia a la evaluación profunda y localizada, análisis, diseño e implantación de procesos.
- La post-mejora: que hace referencia a la gestión de la configuración de los procesos, la revisión de la iteración y los logros de mejora alcanzados.

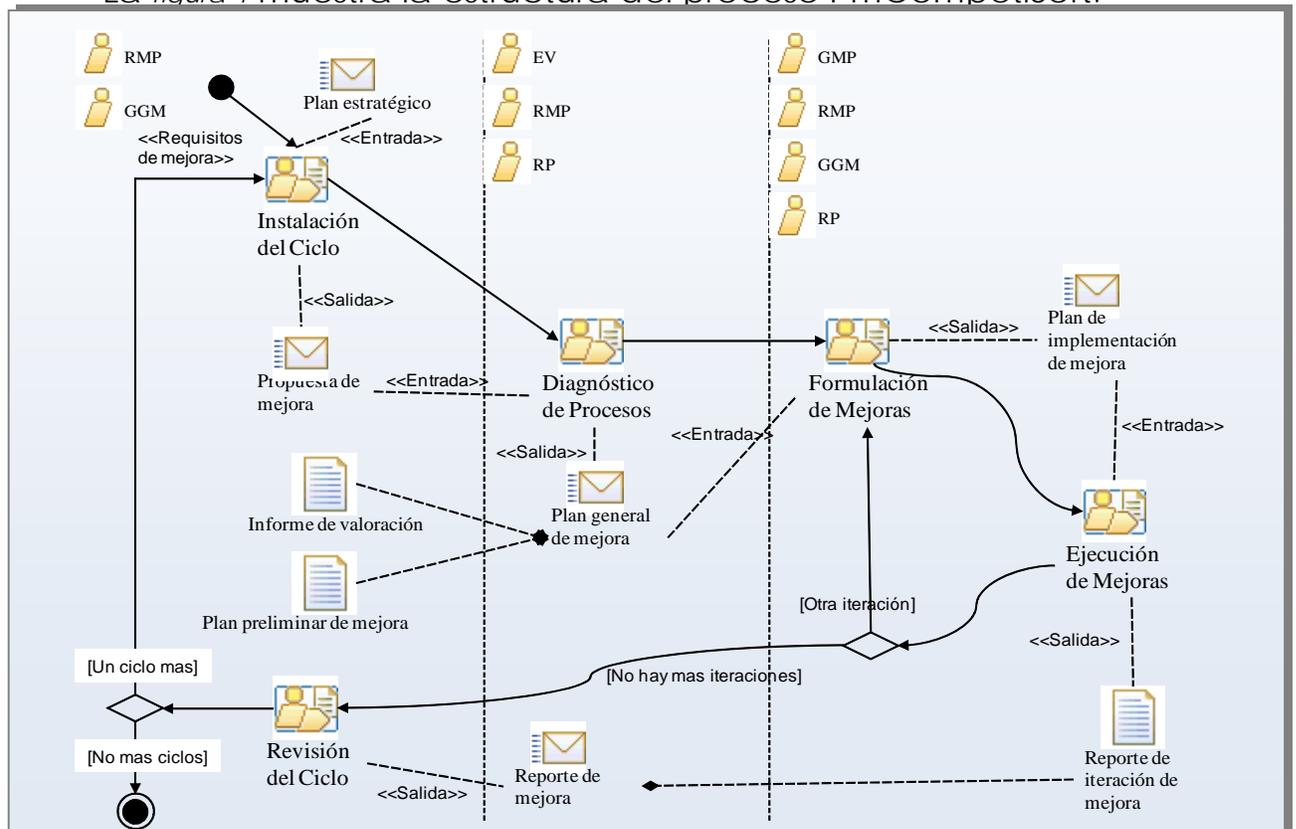
Si la infraestructura de la organización lo permite y los Casos de Mejora son independientes, se pueden ejecutar Iteraciones de Mejora ya sea en serie o en paralelo. Las distintas iteraciones de mejora se realizan dependiendo de la priorización que se haya hecho a los Casos de Mejora.

1.3.2.4 EL PROCESO DE MEJORA PMCOMPETISOFT

PmCompetiSoft es un proceso iterativo e incremental organizado a través de Iteraciones de Mejora que pueden abarcar uno o más Casos de Mejora, ejecutando todo esto dentro de un Ciclo de Mejora global.

Gracias a esta forma de ejecución, mantiene la filosofía de resultados rápidos de mejora, que permite que las mejoras sean visibles desde las actividades tempranas, en la medida en que las Iteraciones de Mejora terminen dependiendo de los criterios de priorización que la organización haya definido desde el principio del Ciclo de Mejora.

La figura 4 muestra la estructura del proceso PmCompetiSoft.



Capítulo 2

APLICACIONES RICAS DE INTERNET

Día a día surgen propuestas de tecnologías innovadoras que buscan mejorar la experiencia del usuario ofreciendo un mejoramiento importante en la interfaz de usuario. Con esto, se pretende hacer más sencillo el uso de las aplicaciones web o de escritorio.

2.1

2.1 LA EXPERIENCIA DEL USUARIO

En todo Proyecto de Software se debe tener en cuenta la experiencia que vivirá el usuario final, ya que en parte de ello depende que el usuario logre realizar las tareas para las cuales fué creado un sistema de software, o bien, que el usuario realice sus tareas de manera simple, tal que pueda aprovechar todas las funcionalidades brindadas por un sistema de software.

Desde el enfoque del negocio, el proporcionar a los clientes una experiencia superior puede ser clave para que una empresa tenga éxito, además, el gran auge de Internet ha ido sustituyendo las interacciones personales, lo cual puede ser en determinado momento ventajoso, por ejemplo para reducir costos y tiempos.

En respuesta al interés de fomentar mejores experiencias a los usuarios finales, surge una nueva categoría de aplicaciones de Internet: las Aplicaciones Ricas de Internet (RIA, Internet Rich Applications) (29).

Estas aplicaciones combinan la facilidad de respuesta y la interactividad de las aplicaciones de escritorio, con el amplio alcance y la facilidad de distribución de las aplicaciones web. Las RIA's impulsan un mayor rendimiento de las inversiones porque simplifican y mejoran la interacción del usuario de las aplicaciones web. Y posibilitan aplicaciones que promueven una experiencia de usuario más dinámica, más interactiva y más sensible (30).

Algunos autores consideran este tipo de aplicaciones como una tecnología que forma parte de la web 2.0.

2.2 EVOLUCIÓN DE HERRAMIENTAS Y PATRONES

Cuando hablamos de las RIA's web, es indispensable hacer la comparación entre las aplicaciones web tradicionales (HTML) y las RIA's web.

Por un lado, las aplicaciones web tradicionales basan todo el procesamiento de datos en el servidor, utilizando al cliente (Navegador) solo para mostrar la información generada, teniendo que redibujar todo el contenido

de la página entre una interacción y otra, aunque solo sea por un cambio mínimo.

Mientras que en las RIA's solo es necesario cargar la interfaz una sola vez haciendo pedidos específicos de información al servidor sin la necesidad de recargar nada, solo mostrar la respuesta del servidor reduciendo así el tiempo de espera del usuario y teniendo la posibilidad de retroalimentar al usuario para que esté enterado de que está pasando en la aplicación en cada momento.

Otro beneficio de las RIA's sobre las aplicaciones web tradicionales es sin duda las mejoras en la Interfaz gráfica del usuario, ya que en las RIA's podemos agregar, eliminar y modificar elementos sin tener que recargar la vista actual, con lo que se pueden ofrecer interfaces más atractivas y sobre todo interactivas.

Si bien la propuesta de las RIA's es bastante atractiva, se busca que las organizaciones puedan integrar estas nuevas tecnologías con su infraestructura y procesos existentes. Para facilitar la adopción de nuevas tecnologías y promover el rehúso, las nuevas tecnologías deben cumplir como mínimo con los siguientes requisitos:

- Proporcionar un lenguaje de programación conocido. Los desarrolladores se han ido familiarizando con lenguajes orientados a objetos (como Java y C#) para el desarrollo de la lógica del negocio y el desarrollo del sistema central. También están familiarizados con lenguajes basados en etiquetas (como JSP, ASP y CFML) para el desarrollo de la interfaz de usuario. Es primordial que las nuevas tecnologías surjan en base a estos lenguajes existentes para poder aprovechar las habilidades que ya se tienen y asegurar una fácil adopción a bajo costo.
- Aprovechar la infraestructura existente. Las organizaciones invierten mucho en tecnología de servidor de aplicaciones. El uso y la continuidad de esa infraestructura es también un requisito para la mayoría de las organizaciones al adoptar una tecnología nueva.
- Adoptar protocolos de normas y API. Uno de los resultados positivos de la web es la adopción de una amplia gama de normas que abarcan toda una variedad de tecnologías. Esto incluye, entre otras, normas

establecidas en la industria, tales como HTML/HTTP(S), XML, SOAP/servicios web, CSS, SVG, así como J2EE y .NET.

- Continuar con el uso de herramientas existentes. Un elemento clave para los desarrolladores que están adoptando soluciones de niveles de presentación es asegurar que puedan usar editores existentes o Entornos de Desarrollo Integrado (IDE, Integrated Development Environment) para escribir las aplicaciones. Entornos como Eclipse, Borland JBuilder, JetBrains IntelliJ IDEA, Microsoft Visual Studio.
- Herramientas eficientes y productivas. Las herramientas de productividad le ayudan a los desarrolladores a trabajar con eficiencia con las nuevas tecnologías. También pueden ayudar en el aprendizaje de nuevos lenguajes, adoptar nuevas estructuras y aprender los montajes de la nueva tecnología, adicionalmente, puede promover la adopción de mejores prácticas.
- Soporte para patrones de diseño. El desarrollo de software se vale de patrones de diseño en la arquitectura y creación de las aplicaciones. En los últimos años, los patrones como el Modelo Vista-Controlador (MVC) han venido forjándose un lugar dominante en el desarrollo de las aplicaciones con J2EE ó .NET. Por lo que es importante que las soluciones de niveles de presentación se puedan ajustar dentro de esa arquitectura basada en patrones.

2.3 DESARROLLO DE APLICACIONES RIA

Existen actualmente diversas tecnologías que permiten el desarrollo de RIA's, tanto para aplicaciones web como para aplicaciones de escritorio, algunas más populares que otras, además tenemos opciones de software libre:

Tabla 3: Algunas tecnologías para el desarrollo de RIA

| Tecnología | Descripción |
|------------|---|
| AJAX | Creado en 2005, AJAX, es un conjunto de técnicas ya existentes que hacen posible la creación de aplicaciones web que se ejecutan en el navegador de los usuarios, sin que sea necesario en todo momento una comunicación con el servidor. |

| | |
|-----------------------------------|--|
| | <p>El funcionamiento de AJAX consiste en que al enviarse la página al navegador desde el servidor web, se incluyen diversos ficheros de información que permiten interactuar dinámicamente con la información mostrada desde el navegador. Uno de los servicios más populares que utilizan esta tecnología es Google, con Gmail y Google Reader.</p> |
| <p>OpenLaszlo (31)</p> | <p>Es una plataforma código abierto para el desarrollo y distribución de RIA. Ofrece un entorno integrado para el desarrollo de aplicaciones RIA que genera el código Flash o DHTML necesario para la ejecución en el navegador del usuario.</p> <p>Las aplicaciones de Laszlo pueden ser desplegadas como tradicionales Java Servlets, que se compilan y regresan al navegador de manera dinámica. Este método requiere que en el servidor web se ejecute el Servidor OpenLaszlo.</p> |
| <p>JavaFX (32)</p> | <p>Es una familia de productos y tecnologías de Sun Microsystems, anunciados en la conferencia de desarrolladores JavaOne en Mayo de 2007. Los productos JavaFX se han pensado para ser usados en la creación de RIA's. Actualmente JavaFX se compone de JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX planeados.</p> <p>Permite a los desarrolladores construir RIA's de alto impacto de forma sencilla y rápida, que combinan gráficos de dos y tres dimensiones, audio, vídeo de alta fidelidad y animaciones, aprovechando la solidez y funcionalidad de la plataforma existente de Java. En julio de 2008 Sun Microsystems lanzó una versión preliminar del kit de desarrollo de JavaFX.</p> |
| <p>Microsoft Silverlight (33)</p> | <p>Es un complemento para navegadores de Internet que agrega nuevas funciones a Windows Presentation como la reproducción de vídeos, gráficos, animaciones y otros elementos. El objetivo es crear una aplicación que haga las mismas cosas que flash player, es decir, Microsoft Silverlight es como un flash player. Silverlight conserva un modo de gráficos de sistema, similar al del WPF e integra en un solo complemento multimedia, gráficos de computadora, animaciones e interactividad.</p> <p>La base de su programación es XAML y el acceso a los objetos</p> |

| | |
|--------------------------|---|
| | <p>esta dado por JavaScript. El XAML puede ser usado para marcar los gráficos vectoriales y las animaciones. Microsoft Expression Blend es la herramienta de Microsoft que se utiliza para crear las animaciones en Silverlight.</p> |
| <p>Adobe® Flex® (34)</p> | <p>Es un término que agrupa una serie de tecnologías publicadas desde Marzo de 2004 por Macromedia para dar soporte al despliegue y desarrollo de Aplicaciones Enriquecidas de Internet, basadas en su plataforma propietaria Flash.</p> <p>Flex fue inicialmente liberado como una aplicación de la J2EE o biblioteca de etiquetas JSP que compilara el lenguaje de marcas Flex (MXML) y ejecutara mediante ActionScript aplicaciones Flash (archivos SWF binarios).</p> <p>Versiones posteriores de Flex soportan la creación de archivos estáticos que son compilados, y que pueden ser distribuidos en línea sin la necesidad de tener una licencia de servidor. El objetivo de Flex es permitir a los desarrolladores de aplicaciones web construir rápida y fácilmente Aplicaciones de Internet Ricas. Vistas en un modelo multi-capa, las aplicaciones Flex pertenecen al nivel de presentación (también llamado: interfaz).</p> <p>Flex 2 cambio el modelo de licencias para abrir la puerta a una versión libre de esta tecnología denominada "Flex Framework". Flex Builder 2 está basado en el entorno de desarrollo Eclipse. Los servicios orientados a empresas siguieron estando disponibles para aquellos que necesitan características avanzadas, tales como las pruebas automáticas. Flex 2 introdujo el uso de una nueva versión del lenguajes de scripts ActionScript, ActionScript 3, que requiere reproductor Flash 9 o posterior para su funcionamiento.</p> <p>Flex 3 fue inicialmente puesto en libertad el 25 de febrero de 2008. Se centra en 4 áreas:</p> <ul style="list-style-type: none"> • Diseñador / Desarrollador de flujo de trabajo: por ejemplo, el aumento de las limitaciones, Flex Component Kit de Flash CS3 • Trabajar con datos: por ejemplo, la lista de efectos • Adobe AIR Applications: apoyo a características en el marco de Adobe AIR • Evolución de Plataforma: por ejemplo, la persistencia de la |

| | |
|------------------------|--|
| | <p>memoria caché, el rendimiento del compilador</p> |
| <p>Adobe® Air (35)</p> | <p>Es un entorno de ejecución multiplataforma para la construcción de aplicaciones RIA utilizando Adobe Flash, Adobe Flex, HTML y Ajax, las cuales pueden usarse como aplicaciones de escritorio.</p> <p>Las aplicaciones AIR pueden funcionar sin conexión a internet y, a continuación, activar aún más la funcionalidad de carga de datos o cuando la conexión esté disponible.</p> |

Capítulo 3

GENESIS, SISTEMA DE APOYO A UN PROCESO DE MEJORA

Cada vez son más notorios los esfuerzos en el campo de la Mejora de Procesos de Software aplicados a las MyPE's. Aplicar una iniciativa de mejora de procesos es una estrategia actualmente utilizada por este tipo de empresas para desarrollar software con niveles aceptables de calidad. Un factor que puede ayudar a las MyPE's a conducir con éxito una iniciativa de mejora de procesos, es ofrecer apoyo tecnológico mediante herramientas software que les permita guiar, aplicar y gestionar modelos y/o métodos de procesos, evaluación ó mejora.

GENESIS es una herramienta que ofrece soporte al proceso de mejora PmCompetiSoft, apoyando a los responsables de la conducción de la mejora de procesos en MyPE's durante la implementación de estas actividades al interior de una organización.

3.1

3.1 ANTECEDENTES

El interés de la comunidad de ingeniería del software hacia la mejora de procesos de software ha tomado gran fuerza durante los últimos años. La mejora de procesos se ha convertido en uno de los objetivos estratégicos de las organizaciones dedicadas al desarrollo de software, ya que requieren ofrecer mejores resultados con menores costos y así poder ser competitivos en el mercado de software.

Poco a poco surgieron diversas iniciativas de mejora de procesos de software como: CMM (9), CMMI (10), ISO/IEC 12207 (11), ISO 15554 (36) (12), ISO 9001:2000 (13) entre otros; los cuales fueron creados pensando en grandes empresas (14) (15). En la mayoría de estos modelos se requiere una gran inversión económica, de tiempo y de recursos para poder adoptar sus prácticas (3) (4).

Diversos estudios han mostrado que la gran mayoría de las empresas dedicadas al desarrollo de software tienen muy pocos empleados y sin embargo desarrollan productos significativos (37) (38) (39). Gracias a esto ha surgido la necesidad de adaptar las propuestas de mejora para poder aplicarlos a pequeñas y medianas organizaciones de software, para las cuales es importante consumir poco tiempo, pocos recursos y además de contar con modelos de referencia de procesos no tan rigurosos que puedan adaptarse al tamaño y necesidades de la organización.

A partir de estas necesidades se han elaborado diversas propuestas de mejoras de procesos orientados a pequeñas organizaciones de software, de las cuales destacan: COMPETISOFT (23), MOPROSOFT (16), MR-MPS (18), RAMALA (19).

Estas propuestas en su mayoría, adaptan alguno de los modelos anteriormente presentados, con lo que pretenden reducir costos y tiempo durante la adopción de sus prácticas en pequeñas organizaciones, además de que se muestran flexibles ante las necesidades de negocio de este tipo de organizaciones.

La forma más utilizada por las pequeñas organizaciones para conducir la mejora es el uso del modelo IDEAL (17) o adaptaciones del mismo; aun-

que también hay organizaciones que utilizan IMPACT (22) o MESOPYME (20) como guía para llevar a cabo las actividades relacionadas con la mejora (40).

El marco metodológico desarrollado en el contexto del proyecto COMPETISOFT además de contar con un modelo de referencia y con un modelo de evaluación, define también un modelo de mejora el cual está basado en Agile SPI (21). Este modelo define un proceso para guiar la mejora continua de procesos, denominado PmCompetiSoft (26).

Al ejecutar un proceso de mejora en una organización pequeña, es importante tener en cuenta ciertos factores que pueden ser determinantes para terminar con éxito la adopción de mejoras: la documentación generada; la carga cognitiva; la disposición de pocos recursos dentro de la organización; la realización de tareas repetitivas.

Para ayudar a minimizar el impacto de los factores mencionados, se deben proporcionar a las organizaciones pequeñas herramientas de software que le permitan gestionar la información generada, realizar tareas repetitivas y que además guíen a la organización durante la adopción de mejoras.

En el mercado del software podemos encontrar múltiples herramientas, tanto comerciales como de software libre, que ofrecen a una organización soporte durante la planeación, gestión y control de cualquier tipo proyectos.

Tomando en cuenta la importancia de utilizar un modelo de mejora que ofrezca la guía necesaria al momento de adoptar mejoras, existen también en el mercado herramientas de software que además de ofrecer funcionalidades propias de la gestión de proyectos, ofrecen soporte para alguno de los modelos de mejora, reduciendo la carga cognitiva de los recursos involucrados y realizando tareas más específicas propias del proceso de mejora.

MKS Integrity Suite (41), apoya iniciativas de mejora de procesos de software y el cumplimiento de certificaciones como CMM, CMMI. Proporciona seguimiento y control automatizado, permitiendo a la organización rastrear los orígenes y el detalle de los cambios de las aplicaciones críticas del ne-

gocio y del proceso documentado, cuenta con una arquitectura flexible orientada al proceso que permite a las empresas construir procesos repetibles y flujos de trabajo ajustados a CMM.

TargetProcess (42), es una suite de gestión de proyectos y seguimiento de fallos de software, ofrece soporte en los procesos ágiles de desarrollo, centrándose en la planificación del proyecto y seguimiento de fallos, apoya a todos los procesos iterativos y promueve el uso de una planificación al estilo Extreme Programming (43) y otros procesos ágiles.

SIMPLe (44) es una herramienta de software que está orientada a dar soporte al jefe de proyecto de calidad en la gestión de un proyecto de mejora de procesos. Proporciona conceptos básicos de CMMI y propone un modelo para gestión del proyecto basado en IDEAL. Contiene información de los modelos y material descriptivo de ayuda. La herramienta SIMPLe brinda soporte para la documentación de procesos organizacionales, relacionándolos con el modelo CMMI. Cuenta con una opción para activar áreas de proceso para el ciclo, establecer responsable del área de procesos y generar la creación automática de actividades.

3.2 PLANTEAMIENTO GENERAL DEL PROBLEMA

Existen pocos trabajos relacionados con modelos de mejora que guíen la adopción de mejoras en para organizaciones pequeñas de software, y esto conduce a que la utilización de un modelo de implementación de mejora sea baja (40).

Además se cuentan con pocas opciones de herramientas de software que soporten directamente un modelo para guiar la mejora de procesos.

Cabe resaltar la importancia de contar con la utilización de un modelo para guiar la mejora, ya que ayuda a planificar y gestionar todas las actividades relacionadas con la mejora de procesos y así poder concluir con éxito la adopción de un modelo de referencia de procesos o continuar con la mejora continua de los procesos dentro de la organización.

A pesar de que en el mercado se ofrecen algunas herramientas de software orientadas a modelos de mejora como IDEAL, estas herramientas solo brindan soporte a modelos de referencia de procesos como CMM o CMMI y a algunos procesos ágiles.

Recordemos que estos modelos fueron creados pensando en grandes organizaciones, surge entonces la necesidad de proveer a las pequeñas organizaciones herramientas de software que además de soportar modelos de procesos como CMMI, también brinden soporte a modelos orientados a pequeñas organizaciones, es decir, que soporten modelos como MOPROSOFT ó COMPETISOFT.

Hay que resaltar también la necesidad de proveer soporte a modelos de mejora orientados a pequeñas organizaciones, como por ejemplo: PROCESSUS (45), MESOPyME ó PmCompetiSoft, ya que estos modelos fueron diseñados a partir de las necesidades y características propias de este tipo de organizaciones.

Además, tomando en cuenta la necesidad de consumir la menor cantidad de recursos durante la adopción de mejoras, es importante guiar en cada momento del proceso al o los responsables de las mejoras, los cuales, sin necesidad de ser expertos en mejora de procesos de software, pue-

dan llevar a cabo la gestión de la mejora continua mediante la utilización de un asistente, o mediante la consulta de la ayuda brindada por una herramienta de software.

3.3 PROPUESTA DE SOLUCIÓN

En esta tesis se propone el diseño y construcción de 3 módulos que integran el núcleo del *sistema de apoyo a un proceso de mejora "GENESIS"*. Las soluciones ofrecidas por estos módulos están orientadas a la planificación de una implementación de mejora y al diagnóstico de los procesos que la organización desea mejorar.

La estructura de la implementación de mejora, así como las actividades a las cuales darán soporte los módulos propuestos serán acorde al proceso de mejora PmCompetiSoft. Es decir, que una implementación de mejora planificada con GENESIS podrá contener n -ciclos de mejora. Cada ciclo de mejora se ejecutará en 5 fases o actividades.

Las oportunidades de mejora obtenidas durante la fase de diagnóstico de la organización podrán agruparse en casos de mejora. Los casos de mejora podrán ser asignados a iteraciones de mejora las cuales se ejecutan dentro de ciclo de mejora. Y se podrá ejecutar dentro de un ciclo de mejora una o más iteraciones de mejora ya sea de forma secuencial o de forma paralela.

Aunque son pocos los recursos disponibles en una pequeña organización, es necesario poder administrar dichos recursos con la finalidad de poder distribuir las tareas del proceso de mejora entre recursos humanos, y por otro lado se optimice el uso de los recursos materiales durante la ejecución del proceso de mejora. Por lo tanto, en los módulos propuestos se ofrece al Responsable de Mejora de Procesos herramientas que le permitan la administración tanto de recursos humanos como de recursos materiales.

GENESIS se propone como un sistema independiente del modelo de referencia de procesos a utilizar durante la implementación de mejoras. Es así que cualquier modelo de referencia de procesos (como CCMI, CompetiSoft, MoProSoft) que este modelado en SPEM (7) a través de la herramienta EPFComposer (46) podrá ser gestionado por el sistema GENESIS.

Uno de los módulos propuestos ofrece una forma de valoración de procesos, el cual permitirá tomar y valorar las actividades de los procesos del modelo de referencia de procesos elegido para la adopción de mejoras. A partir de esta valoración se podrán obtener las oportunidades de mejora de los procesos.

Toda la información generada durante la ejecución de los ciclos de mejora, así como la información de los recursos queda disponible para su análisis por parte de los módulos propuestos en la tesis complementaria a esta tesis (47).

3.4 DESCRIPCIÓN GENERAL DE GENESIS

Al ejecutar un proceso de mejora, ya sea durante la implantación de un modelo de referencia de procesos o bien, durante la mejora continua de los procesos, surgen diversas necesidades. Estas necesidades pueden ser resueltas y/o se puede minimizar su impacto sobre la organización, a través de la automatización de las tareas de los procesos.

GENESIS es un sistema que permite la automatización del proceso de mejora (PmCompetiSoft). Por lo tanto, se da soporte a cada una de las 5 actividades que conforman PmCompetiSoft: Instalación del Ciclo, Diagnóstico de Procesos, Formulación de Mejoras, Ejecución de Mejoras y Revisión del Ciclo.

GENESIS es independiente del modelo de referencia de procesos, por lo que se pueden administrar las oportunidades de mejora de cualquier modelo de referencia de procesos (CMMI, MoProSoft, CompetiSoft), por lo tanto una organización puede elegir el modelo que más se adapte a sus necesidades de negocio. Para que un modelo de referencia de procesos pueda ser soportado por GENESIS, solo bastará con tener modelados los procesos del modelo elegido en SPEM (7) mediante de EPFComposer (46).

Ya se cuenta con el modelo de referencia de procesos COMPETISOFT modelado en EPFComposer, así que por el momento GENESIS solo puede ser usada para la implementación de este modelo. Posteriormente, si más modelos de referencia de procesos son modelados en EPFComposer, podrán ser utilizados para su implantación con GENESIS.

GENESIS permitirá a una organización crear distintos ambientes de trabajo, y en cada ambiente definir el modelo de referencia de procesos y el nivel de madurez que la organización desea alcanzar, así como el o los procesos que se deseen implementar o mejorar.

La forma en que GENESIS permite al Responsable de Mejora de Procesos (RMP) obtener las oportunidades de mejora, es a través de la realización de valoraciones rápidas. Una valoración rápida se realiza a través de la aplicación de un cuestionario que toma las actividades de los procesos a mejorar. Este cuestionario permitirá indicar el nivel de cumplimiento de las

prácticas requeridas por el modelo de referencia de procesos, registrar las referencias de la evidencia física que avala dicho cumplimiento y registrar los comentarios de las evidencias.

Una vez que el RMP cuenta con las oportunidades de mejora, puede priorizar las oportunidades de mejora obtenidas y comenzar con la planificación de los casos de mejora.

Uno de los objetivos de este sistema es aumentar la productividad y reducir la carga cognitiva del RMP y de los involucrados en las mejoras, es así que GENESIS cuenta con un asistente, el cual indica en orden cada una de las tareas que se deben realizar durante el proceso de mejora y solo continúa con la ejecución del proceso si las actividades o tareas requeridas son completadas.

Permite también, administrar y gestionar la información generada durante el proceso de mejora. La información generada estará disponible durante un ciclo de mejora ó durante ciclos de mejora posteriores, permitiendo así el análisis de ciclos anteriores, lo cual es importante durante la toma de decisiones en la creación de nuevos ciclos de mejora.

Además el RMP tendrá disponible para consulta la información correspondiente al modelo de referencia de procesos relacionada con las oportunidades de mejora en ejecución (actividades, roles, productos de entrada, salida, internos, etc.).

Ofrece además trazabilidad de las oportunidades de mejora, con lo que se podrá obtener el origen de una oportunidad de mejora y su progreso durante las iteraciones de mejora en los cuales se haya gestionado dicha oportunidad.

Toda la información correspondiente a los ciclos de mejora, estará disponible para la generación e impresión de reportes o documentos de trabajo correspondientes al proceso de mejora (PmCompetiSoft).

Con la finalidad de ofrecer una mejor experiencia al usuario, GENESIS es una Aplicación Rica de Internet (RIA), por lo que además de ser una herramienta web, la interfaz de usuario ofrecida es dinámica y pretende ser gráficamente atractiva para el usuario. GENESIS es una herramienta web

colaborativa, por lo que se permite la distribución o delegación de las responsabilidades en la ejecución de mejoras.

3.5 COMPONENTES O MÓDULOS DE GENESIS

GENESIS se compone básicamente de 6 módulos. Estos módulos brindan soporte durante distintos momentos de la ejecución de cada una de las actividades del proceso de mejora PmCompetiSoft. En la *figura 5* se muestra de forma general la interacción de los módulos de génesis con las actividades de PmCompetiSoft:

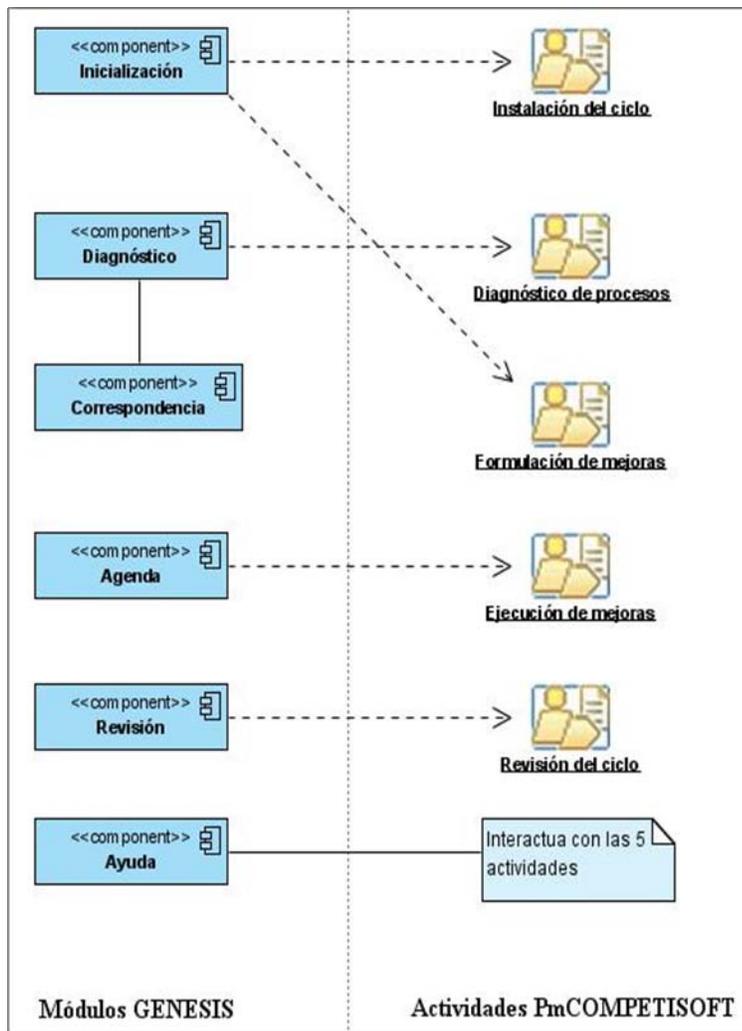


Figura 5: Módulos de GENESIS y su interacción con las actividades del proceso PmCompetiSoft

A continuación se describen brevemente las principales funcionalidades ofrecidas por cada módulo que integra GENESIS:

- *Inicialización*: en este módulo el RMP puede establecer el modelo de referencia de procesos a usar para la implementación de mejoras, el ni-

vel de capacidad que desea alcanzar la organización y el conjunto de procesos a mejorar.

A demás de los procesos requeridos por el modelo de referencia se pueden incluir procesos propios de la organización⁶, que se desean gestionar para su mejora con el uso de GENESIS. Estos procesos también deberán estar modelados en SPEM mediante EPFComposer para que puedan ser soportados por el sistema.

Este módulo permite planificar la implementación de mejoras, asignando tiempos y recursos a ciclos, iteraciones y casos de mejora. El RMP puede crear casos de mejora a partir de las oportunidades de mejora obtenidos en el módulo de *Diagnóstico* y posteriormente asignarlos a iteraciones de mejora para su ejecución. Los casos de mejora pueden priorizarse para su ejecución de acuerdo a las necesidades de la organización.

- *Diagnóstico*: este módulo permite al RMP obtener las oportunidades de mejora de los procesos que fueron seleccionados en el módulo de *Iniciación* para su mejora, las oportunidades de mejora se obtienen por el momento a partir de la realización de valoraciones rápidas, en las cuales se aplica un cuestionario, y de acuerdo al resultado obtenido se obtendrán las oportunidades de mejora de los procesos evaluados.
- *Correspondencia*: Después de realizar una valoración rápida, la organización contará con una relación de las oportunidades de mejora encontradas en los procesos evaluados contra la evidencia existente por parte de la organización (documentación o practicas realizadas para cumplir con las actividades de los procesos).

El RMP puede hacer modificaciones de las evidencias solo durante la ejecución del caso de mejora al cual pertenece la oportunidad de mejora. Una vez que el caso de mejora sea cerrado, la información histórica por caso de mejora de las evidencias está disponible para consulta.

⁶ Procesos propios: son procesos creados e implementados por la organización, que debido a su estructura o propósito, no pueden ser mapeados con ninguno de los proceso pertenecientes al modelo de referencia de procesos (CMMI, MoProSoft, CompetiSoft, etc.) elegido para la implementación de mejoras.

- *Agenda*: este módulo permite gestionar la ejecución de los ciclos de mejora. Se divide en 2 sub-módulos los cuales se describen a continuación:
 - *Agenda de eventos*: permite al RMP administrar eventos relacionados con las actividades del proceso de mejora PmCompetiSoft; a los cuales se les puede asociar la siguiente información: fecha de inicio, fecha de término, responsable(s), participantes, oportunidad de mejora y/o caso de mejora a la que pertenece, éxito/fracaso, esfuerzo estimado, esfuerzo real, lecciones aprendidas, problemas y soluciones. Los eventos se pueden categorizar según su tipo: valoración, capacitación, actividad, tarea, hito o revisión.
 - *Agenda de productos*: permite al RMP programar y administrar fechas de entregas de productos de trabajo y/o documentación generada por la organización de acuerdo a lo requerido por el modelo de referencia de procesos elegido, solo se pueden agregar productos correspondientes al nivel de capacidades previamente definido por el RMP y a los procesos elegidos en el módulo de *Inicialización*.

A estos productos se les puede asociar la siguiente información: fecha de alta, fecha de entrega, responsable, revisor(es), oportunidad de mejora y/o caso de mejora que cubre, éxito/fracaso, esfuerzo estimado, esfuerzo real, lecciones aprendidas, problemas y soluciones.

- *Revisión*: este módulo permite al RPM consultar el estado actual de la ejecución del proceso de mejora, muestra la información organizada por ciclo, iteración, caso u oportunidad de mejora. Además permite generar los reportes o productos de trabajo del proceso de mejora PmCompetiSoft. Para la impresión de estos productos se utilizan las plantillas propuestas por PmCompetiSoft, las cuales están disponibles en GENESIS.

El RMP dispone de algunos gráficos con la información de ciclos, iteraciones, casos u oportunidades de mejora. Estos gráficos dan una vista rápida y dinámica del progreso de las mejoras en ejecución. GENESIS permite la generación de información sobre el origen, desarrollo y

estado actual de cada oportunidad de mejora en ejecución, esto se realiza por medio de la trazabilidad de las mejoras.

También se pueden generar estadísticas del desempeño de los casos de mejora en donde se podrán obtener esfuerzos estimados y reales, recursos y tiempos. Permite también el uso de las estadísticas generadas en ciclos de mejora anteriores para hacer estimados en la creación de nuevos ciclos de mejora.

- *Ayuda*: este módulo ofrece una guía de ejecución del proceso de mejora PmCompetiSoft, indica paso a paso las actividades a realizar durante el proceso de mejora e indica las funcionalidades disponibles en GENESIS para poder apoyar dichas actividades. Desde este módulo está disponible el Modelo de Referencia de Procesos para fines de consulta.

Con la finalidad de no disminuir las funcionalidades que ofrece GENESIS, se decidió dividir el desarrollo del sistema en 2 partes, y cada una de estas partes se desarrolló en 2 proyectos de tesis, que se han realizado en paralelo, con lo que se puede contar con el sistema completo desde su primer versión.

La parte desarrollada en esta tesis, conforma el núcleo del sistema y se encarga principalmente del diagnóstico de la organización para la obtención de mejoras de la planificación y estructuración de las actividades del proceso de mejora, así como de la administración de los recursos involucrados en la implementación de las mejoras.

La parte, desarrollada en la tesis "GENESIS, Sistema de apoyo a un proceso de mejora: gestión y revisión" (47), está enfocada a la gestión del proceso de mejora, al análisis y revisión de la información generada y a proveer ayuda a los responsables de las mejoras.

En los siguientes apartados se describen en forma muy general los objetivos y el propósito de cada proyecto de tesis.

3.5.1 DIAGNOSTICO DE LA ORGANIZACIÓN Y PLANIFICACIÓN EN GENESIS

La parte de GENESIS que comprende los módulos de *Inicialización*, *Diagnóstico* y *Correspondencia* corresponden al trabajo realizado en este proyecto de tesis. Como se mencionó anteriormente, estos módulos están enfocados principalmente a la planificación del proceso de mejora y al diagnóstico de la organización para la ejecución de mejoras.

Se proponen medios y soluciones viables para un RMP que requiera administrar un proceso de mejora y propone una estructura de actividades para la ejecución de un proceso de mejora basado en PmCompetiSoft.

Básicamente pretende cumplir con los siguientes objetivos generales:

- Permitir la administración de una mejora de procesos de un modelo de referencia de procesos elegido por la organización de acuerdo a sus necesidades de negocio.
- Ofrecer un medio de control para la planificación y estructuración de implementaciones de mejora dentro de una organización.
- Ofrecer funcionalidades útiles para la formulación de mejoras realizando actividades alineadas al proceso de mejora PmCompetiSoft.
- Ofrecer un medio de control para el diagnóstico de una organización, permitiendo identificar las oportunidades de mejora de los procesos.
- Ofrecer un medio para registrar la correspondencia existente entre las actividades que una organización realiza en sus procesos, contra las actividades requeridas por un modelo de referencia de procesos.
- Fomentar el trabajo colaborativo dentro de una organización, permitiendo la administración de recursos y la delegación de responsabilidades a diferentes roles del equipo de trabajo.

Con estos objetivos se dará soporte principalmente a las primeras 3 fases o actividades del proceso de mejora de PmCompetiSoft: *Instalación del Ciclo*, *Diagnóstico de Procesos* y *Formulación de Mejoras*.

3.5.1.1 DIAGNOSTICO Y PLANIFICACIÓN EN PMCOMPETISOFT

La fase de *Instalación del Ciclo* del proceso de mejora PmCompetiSoft es la actividad de partida para el ciclo de mejora, en este punto se estructura

una infraestructura de gestión, en la cual se describe la organización, distribución y delegación de responsabilidades de los recursos que serán destinados para la ejecución de las mejoras. Estas actividades de administración de recursos y estructuración de la implementación de mejora podrán realizarse desde el módulo de inicialización. En la figura 6 se muestra el flujo de trabajo para la fase de instalación del ciclo de mejora.

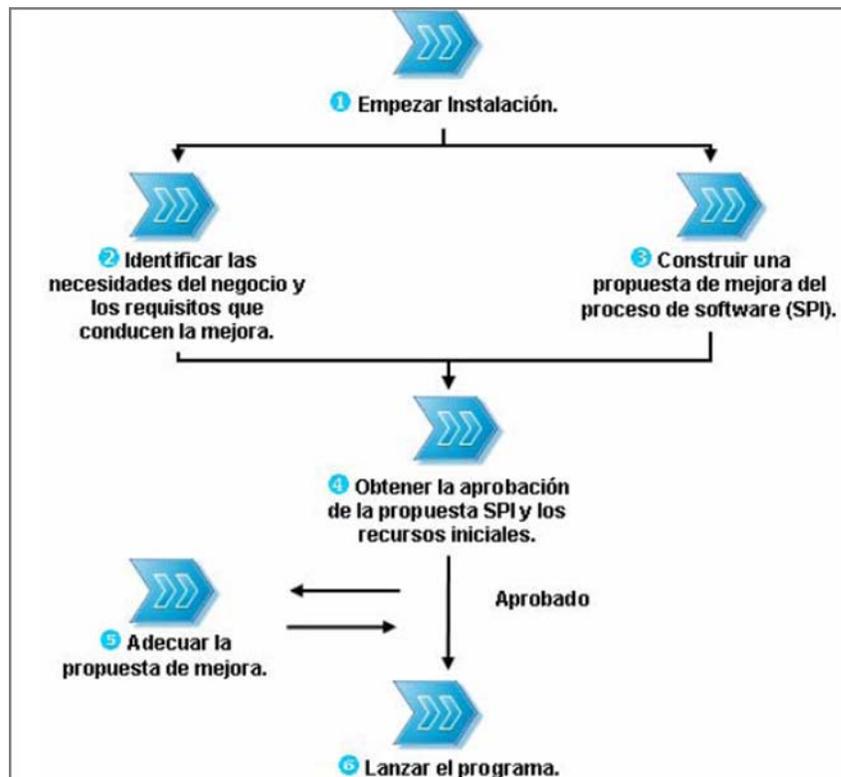


Figura 6: Flujo de trabajo para la fase de instalación de un ciclo de mejora en PmCompetiSoft

La fase siguiente es el *Diagnóstico de procesos*, en esta actividad el ciclo de mejora ya ha iniciado y se requiere realizar actividades de valoración con el fin de reconocer el estado actual de los procesos de la organización. A partir de los resultados de la valoración se obtienen oportunidades de mejora y se establecen los casos de mejora priorizados. En la figura 7 se muestra el flujo de trabajo para esta fase.

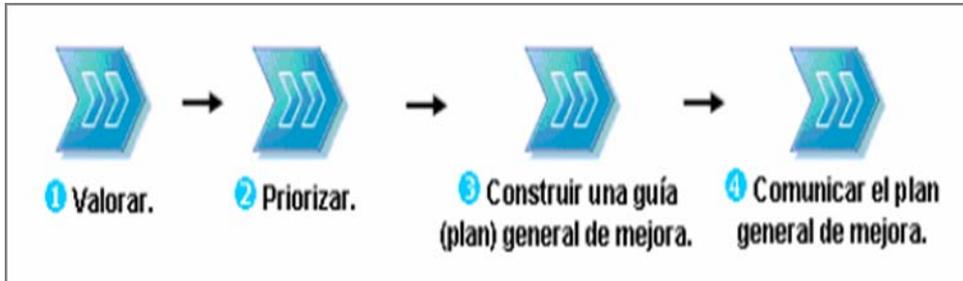


Figura 7: Flujo de trabajo para la fase de diagnóstico de procesos en PmCompetiSoft

El diagnóstico de los procesos podrá realizarse en el módulo de diagnóstico propuesto en esta tesis, el cual tomara las actividades de los procesos, y se podrá asignar a estas actividades un porcentaje de cumplimiento. Ya con las oportunidades de mejora se podrán administrar y priorizar casos de mejora.

Una vez que se cuenta con los casos de mejora priorizados, se planifican las iteraciones de mejora, a las cuales se les asigna uno o más casos de mejora para su ejecución. Esto se realiza en la fase de formulación de mejoras. La figura 8 muestra el flujo de trabajo para la fase de *Formulación de mejoras*.

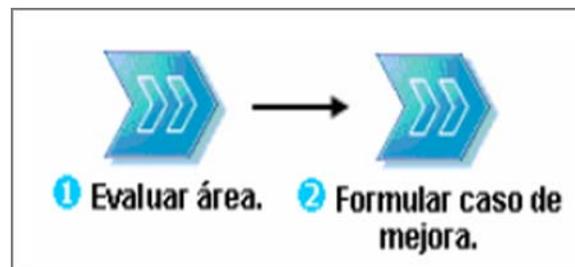


Figura 8: Flujo de trabajo para la fase de formulación de mejoras en PmCompetiSoft

En la siguiente fase: *Ejecución de mejoras* se procede a ejecutar las iteraciones de mejora planeadas en el ciclo de mejora anterior. Se pueden ejecutar una o más iteraciones de mejora, ya sea de forma secuencial o en paralelo. La figura 9 muestra el flujo de trabajo para la fase de *Ejecución de mejoras*.

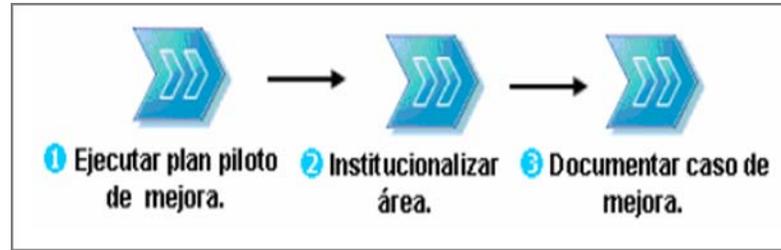


Figura 9: Flujo de trabajo para la fase de ejecución de mejoras en PmCompetiSoft

En el módulo de correspondencia quedará disponible para consulta la información relacionada con las evidencias encontradas para las oportunidades de mejora.

Finalmente una vez concluida la ejecución de las mejoras se continúa con la fase de *Revisión del ciclo*, en esta fase se da cierre a los casos mejora y se registra la información necesaria para realizar un análisis post-mortem. La figura 10 muestra el flujo de trabajo para la fase: *Revisión del ciclo*.



Figura 10: Flujo de trabajo para la fase revisión del ciclo de proceso de mejora PmCompetiSoft

Terminando la ejecución del ciclo de mejora, en el módulo de *Iniciación* se podrá dar cierre a casos, iteraciones, ciclos y/o implementaciones de mejora, registrando la información asociada, que estará disponible para el análisis de las mejoras implementadas.

3.5.2 GESTIÓN Y REVISIÓN DE LAS MEJORAS USANDO GENESIS

La segunda parte de GENESIS comprende los módulos de: *Agenda, Revisión y Ayuda*, los cuales fueron desarrollados en la tesis: "GENESIS, sistema de apoyo a un proceso de mejora: gestión y revisión" (47). La funcionalidad de dichos módulos está orientada principalmente a la gestión y revisión del proceso de mejora PmCompetiSoft.

Con el desarrollo de estos módulos se pretende cumplir con los objetivos generales siguientes:

- Ofrecer al RMP un medio de apoyo para la gestión de los eventos asociados a con la ejecución del proceso de mejora en una organización.
- Ofrecer al RMP un medio de apoyo para la gestión de productos de trabajo requeridos por el modelo de referencia de procesos elegido por la organización para la adopción de mejoras.
- Ofrecer un medio de control al RMP con el cual pueda administrar la información generada durante la ejecución de mejoras de procesos (lecciones aprendidas, problemas, soluciones, esfuerzos estimados y reales, recursos participantes, progreso de las oportunidades de mejora, entre otros).
- Permitir al RMP generar reportes o informes del estado del proceso de mejora durante y posteriormente a la ejecución de mejoras de procesos.
- Ofrecer al RMP trazabilidad de las oportunidades de mejora, con lo que tendrá disponible el origen de dichas oportunidades (en el modelo de referencia de procesos), y podrá revisar su progreso durante y posteriormente a su ejecución.
- Ofrecer ayuda al RMP durante la ejecución de un proceso de mejora, indicando las actividades a realizar en el proceso de mejora PmCompetiSoft y a la par mostrar las funcionalidades de GENESIS disponibles para la realización de dichas oportunidades.

De esta forma se pretende dar soporte sobre todo a las fases de *Ejecución de Mejoras* y *Revisión del Ciclo* del proceso de mejora PmCompetiSoft

Capítulo 4

ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE LOS MÓDULOS



4.1 EL PROCESO DE DESARROLLO

El proceso de desarrollo utilizado para este proyecto de tesis, fue el Proceso Unificado (48), el cual es un proceso de desarrollo de software configurable. Este proceso puede adaptarse en proyectos de distintos tamaños y/o complejidades.

El Proceso Unificado guía a los equipos a cargo de los proyectos para administrar de un modo controlado el desarrollo iterativo, mientras se realiza un balance entre los requerimientos, el tiempo y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de la arquitectura lo antes posible, para implementar y probar el sistema de acuerdo a los requerimientos y a la arquitectura. El Proceso Unificado describe qué entregables se deben producir y cómo se deben producir.

Las características principales del Proceso Unificado son:

- Es iterativo e incremental.
- Está centrado en la arquitectura.
- Es guiado por casos de uso.
- Permite la confrontación de riesgos.

Los conceptos principales del Proceso Unificado son:

- Fase e iteraciones, ¿Cuándo se hace?
- Flujos de trabajo de procesos (actividades y pasos), ¿Qué se está haciendo?
- Artefactos (modelos, reportes, documentos), ¿Qué se produjo?
- Roles, ¿Quién lo hace?

El desarrollo de los módulos que se presentan en esta tesis tuvo un ciclo de vida acorde al flujo de trabajo del Proceso Unificado. Este ciclo de vida se integró de 3 iteraciones y en cada iteración se obtuvieron los siguientes resultados relevantes:

Cada ciclo del proceso de desarrollo incluyó 4 fases:

- Fase de inicio: la cual describe en su flujo de trabajo la administración del proyecto, la definición de requerimientos y el análisis.
- Fase de Elaboración: esta fase contiene en su flujo de trabajo el análisis, la administración de la configuración y el diseño.
- Fase de construcción: aquí se realizan las actividades de implementación y pruebas.
- Fase de transición: en la cual se revisaron los productos generados en el ciclo.

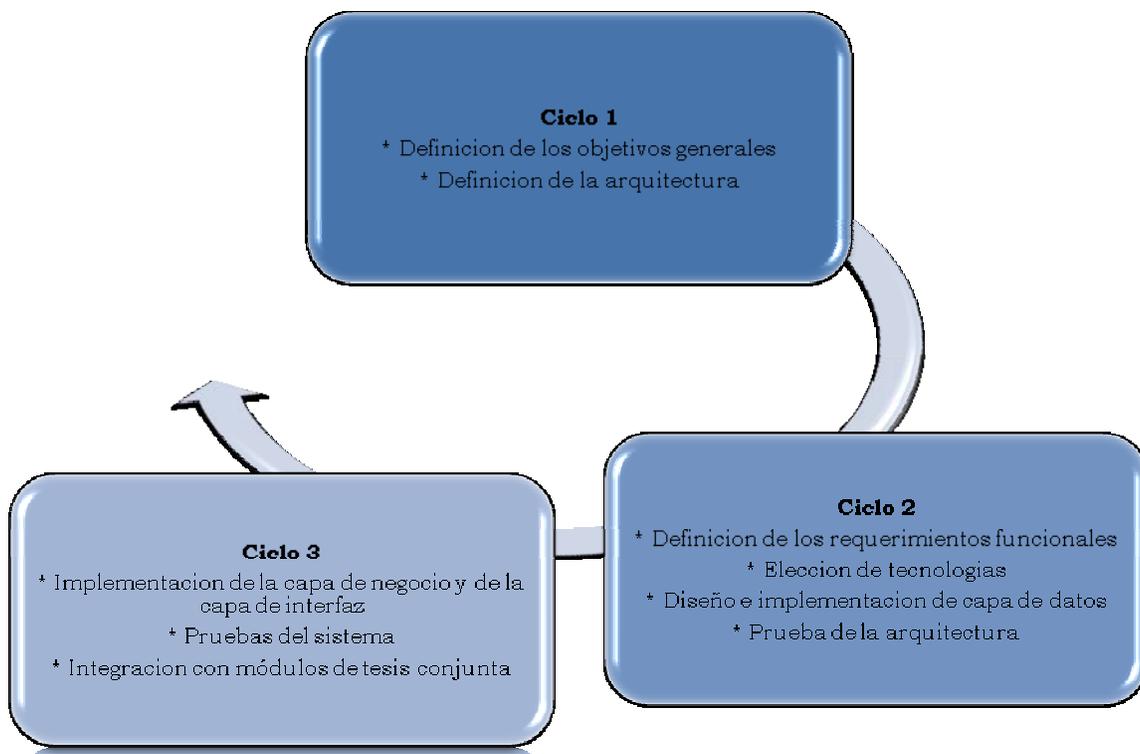


Figura 11: Resultados obtenidos durante el ciclo de vida del Proceso Unificado

4.2 FUNCIONALIDAD DE LOS MÓDULOS DESARROLLADOS

Los módulos desarrollados en este proyecto de tesis tienen como objetivo automatizar algunas de las tareas a realizar durante las primeras fases de la implementación de mejoras en una organización. Estos módulos finalmente serán integrados a GENESIS, con lo cual se obtendrá un sistema que apoyará directamente a un proceso de mejora.

Estos módulos se diseñaron para apoyar al responsable de las mejoras durante la planificación del proceso de mejora, en donde se necesitan definir tiempos para las fases que integran el proceso, asignar recursos disponibles en la organización para la ejecución de las mejoras y organizar en sí la implementación de las mejoras.

El levantamiento de requerimientos del sistema se realizó básicamente en 2 fases, en la primera se realizó una entrevista a un consultor, el cual tiene amplia experiencia implementando mejoras en pequeñas organizaciones de TI mexicanas, en esta fase se determinaron requerimientos que formaron la base de los módulos a desarrollar.

En la segunda fase, los requerimientos fueron centrados al proceso de mejora, se eligió como proceso base a ofrecer en la herramienta el proceso de mejora PmCompetiSoft, por lo tanto se trabajó conjuntamente con un experto en mejoras de procesos de software de la Universidad de Castilla la Mancha para revisar las actividades del proceso e identificar las necesidades primarias a automatizar.

El resultado de este análisis arrojó las siguientes características deseadas para el sistema, las cuales se presentan agrupadas por cada fase del proceso de mejora PmCompetiSoft:

Tabla 4: Resultados del Análisis realizado al proceso de mejora PmCompetiSoft.

| Fase | Resultados |
|------------------------------|--|
| <i>Instalación del ciclo</i> | <p>Debe permitir la creación de usuarios con diferentes niveles de acceso al sistema, con el fin de permitir la delegación de responsabilidades.</p> <p>Debe permitir la gestión de la infraestructura de la mejora administrando los recursos para las mejoras, el costo de los mismos, las posibles relaciones entre ellos y entre los elementos de la planeación.</p> |

| | |
|--------------------------------|--|
| | <p>Debe permitir definir el modelo de referencia de procesos a utilizar para las implementaciones de mejora, a través del modelado con la herramienta EPF-Composer.</p> <p>Debe permitir elegir los procesos a mejorar del modelo de referencia de procesos seleccionado.</p> <p>Debe ofrecer un medio o método de evaluación rápido que solo muestre la información respectiva al nivel y a los procesos previamente seleccionados del modelo de referencia de procesos elegido.</p> <p>Debe permitir la planificación de implementaciones de mejora en tiempo, recursos y costos para hacer estimaciones iniciales.</p> <p>Debe permitir usar la planificación y estimación para conformar la propuesta de mejora.</p> <p>Debe permitir la difusión del estado actual de las actividades y pasos, reflejando la información instantáneamente a los usuarios del sistema.</p> |
| <i>Diagnóstico del proceso</i> | <p>Debe poder identificar las oportunidades a mejorar por medio de la aplicación de valoraciones ligeras a los procesos de la organización elegidos para mejorar.</p> <p>Debe permitir la creación de casos de mejora con oportunidades de mejora provenientes de la evaluación.</p> <p>Debe permitir la priorización de los casos de mejora para definir el orden de ejecución.</p> <p>Debe permitir la priorización de las oportunidades de mejora dentro de un caso para definir el orden de ejecución.</p> <p>Debe permitir la difusión del orden de ejecución de la planeación a los usuarios del sistema.</p> |
| <i>Formulación de mejoras</i> | <p>Debe permitir la planificación de las iteraciones de mejora definiendo el tiempo de duración y su costo.</p> |
| <i>Ejecución de mejoras</i> | <p>Cada que termine un caso de mejora se debe permitir aplicar una evaluación del caso de mejora así al terminar la iteración se tendrá la evaluación actualizada de las mejoras ejecutadas durante la iteración.</p> <p>Debe permitir el almacenamiento de la información asociada a las mejoras, como los recursos involucrados, los costos.</p> |
| <i>Revisión del ciclo</i> | <p>Permitir la recolección de la información generada al cierre del ciclo.</p> |

A partir de los resultados obtenidos se procedió a definir las funcionalidades deseadas para el sistema, quedando agrupadas dichas funcionalidades en 3 módulos: *Inicialización, Diagnóstico y Correspondencia*.

A continuación se describen las características deseadas para cada módulo:

Módulo Inicialización:

- Permitir al RMP la creación de Ambientes de trabajo.
- Permitir la carga de modelos de referencia de procesos modelados en SPEM a mediante la herramienta EPFComposer.
- Permitir la administración de procesos propios de la organización en formato XML.
- Permitir al RMP la selección del modelo de referencia de procesos y el nivel de cumplimiento que desea alcanzar la organización.
- Permitir al RMP la selección del conjunto de procesos a gestionar.
- Permitir al RMP la creación de implementaciones en los ambientes de trabajo.
- Permitir al RMP la administración de usuarios del sistema (RMP, RCM).
- Permitir al RMP la administración de recursos, tanto recursos humanos como recursos materiales.
- Registrar información asociada a los recursos como: nombre, responsable, costo, fecha de asignación, entre otros.
- Registrar la información de presupuestos asignados a una implementación.
- Registrar información asociada a la implementación como la planeación de tiempos y recursos.

- Registrar información asociada a los ciclos de mejora permitiendo la planeación de sus fases e iteraciones y recursos asignados.
- Permitir la importación de recursos de implementaciones de mejora anteriores.

Módulo Diagnóstico:

- Permitir la aplicación de una evaluación rápida a los procesos previamente seleccionados.
- Obtener oportunidades de mejora de los procesos evaluados.
- Registrar la evidencia física, porcentaje de cumplimiento y comentarios a las oportunidades de mejora evaluadas.
- Permitir la agrupación de las oportunidades que tengan menos del 100% de cumplimiento y asignar estas oportunidades agrupadas a casos de mejora.
- Permitir la administración de los casos de mejora.
- Permitir la priorización tanto de las oportunidades de mejora como de los casos de mejora para su ejecución.

Módulo de Correspondencia:

- Administrar la información de la evidencia física y los comentarios de las oportunidades de mejora evaluadas.

Con estas características se identificaron como usuarios del sistema los siguientes roles: Responsable de Mejora de Procesos (RMP) y Responsable de Caso de Mejora (RCM). Posteriormente se definió el diagrama general de casos de uso para el sistema (Ver Figura 12).

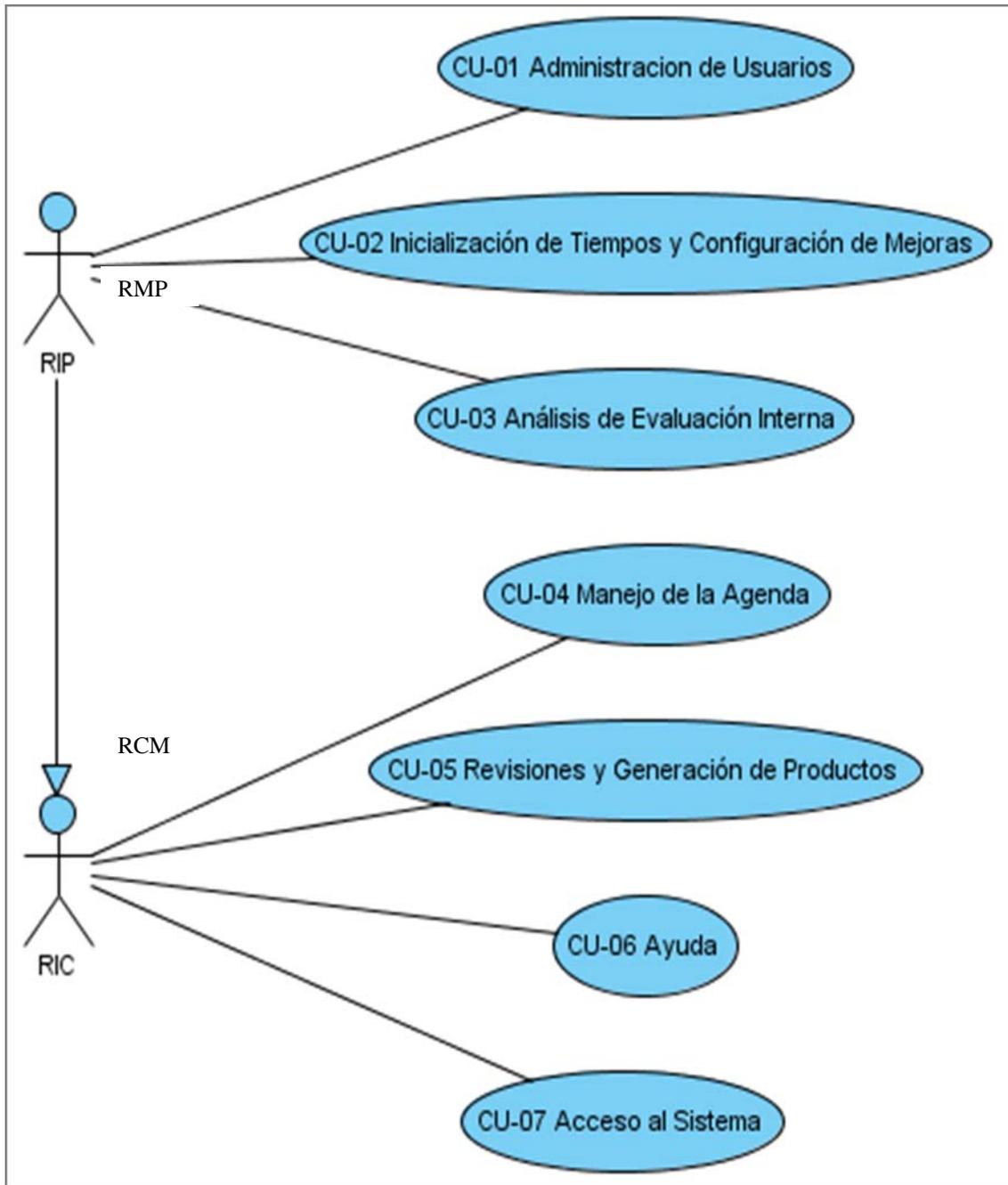


Figura 12: Diagrama General de casos de Uso.

Se identificaron las funcionalidades del sistema y se definió que los casos de uso desarrollados en este proyecto de tesis son exclusivos del RMP así que se creó un diagrama de casos de uso de este proyecto de tesis (Ver

Figura 13). Por lo tanto, solo el RMP tiene acceso a los módulos de *Iniciación, Diagnóstico y correspondencia* ya que es un usuario de mayor nivel y responsabilidad.

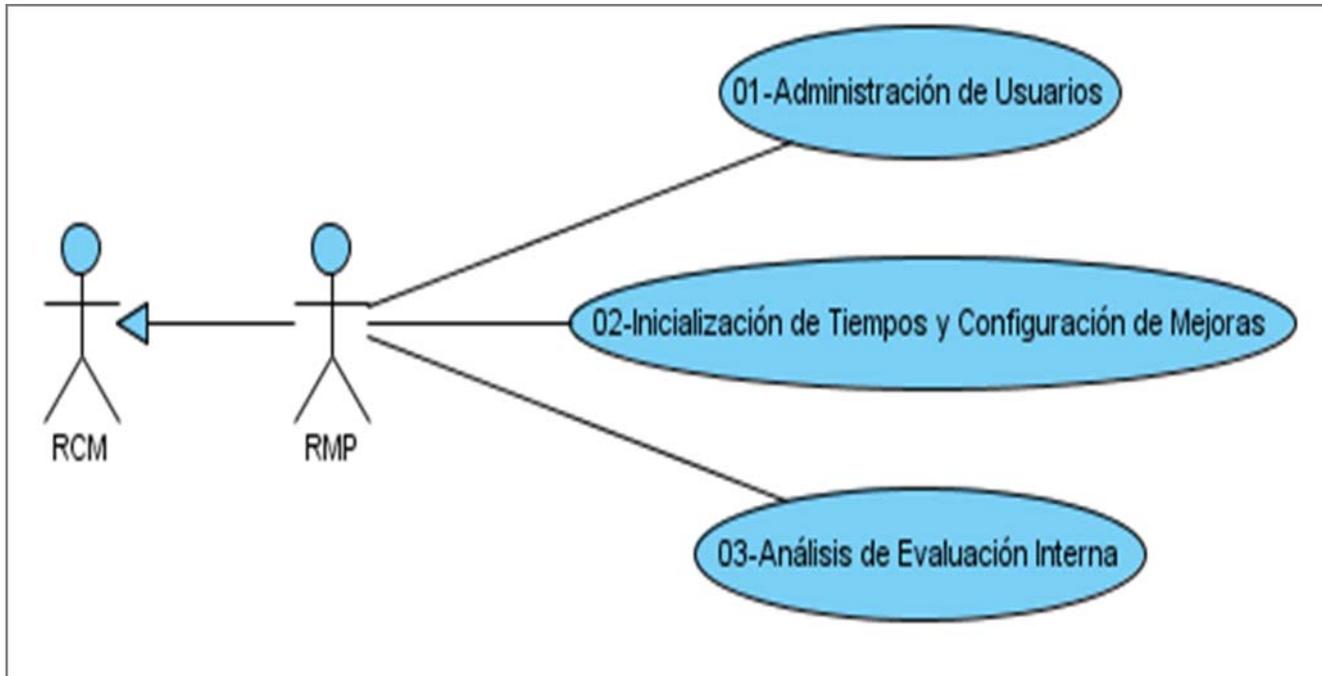


Figura 13: Casos de Uso de este proyecto.

Para describir la información de entrada y salida de cada caso de uso, se realizó el detalle de casos de uso, en esta parte se comenzó con la identificación de las entidades, relaciones y atributos necesarios. A continuación se muestra un ejemplo de detalle de caso de uso para el CU-01.1 Alta de Usuarios:

CU-01.1 Alta de Usuarios

Proyecto: **GENESIS**

Fecha: **19/02/2008**

Autor: **Andres Flores Sanz**

Grupo: **UNAM**

| | Nivel |
|---|----------------------|
| X | Resumen muy general |
| | Resumen |
| | Actividad de Usuario |
| | Detalle |

| | Alcance |
|---|----------------------------|
| X | Organización (caja negra) |
| | Organización (caja blanca) |
| | Módulo |
| | Método |

Descripción Breve

Permite al *RMP* ingresar los datos de un usuario nuevo al *sistema*.

Actores

| | |
|---------------------|---|
| Actor Principal | Responsable de Mejora de Procesos (<i>RMP</i>). |
| Actores Secundarios | |

Eventos que lo inician

| | |
|---|---|
| 1 | El <i>RMP</i> seleccionó la opción <i>Nuevo</i> en la pantalla de <i>Administración de Usuarios</i> . |
|---|---|

Flujo de eventos primario

| | |
|---|--|
| 1 | El <i>sistema</i> despliega la pantalla de <i>Datos de Usuarios</i> vacía. |
| 2 | El <i>RMP</i> llena la información del usuario nuevo. |
| 3 | El <i>RMP</i> selecciona la opción <i>Guardar</i> . |
| 4 | El <i>sistema</i> almacena la información del usuario nuevo. |
| 5 | El <i>sistema</i> cierra la pantalla de <i>Datos de Usuarios</i> . |
| 6 | Fin del caso. |

Flujo de eventos alternativos

| | |
|-----|--|
| 2.1 | EL <i>RMP</i> selecciona la opción <i>Cerrar</i> . |
| 2.2 | Se continúa en el punto 5 del flujo primario. |
| 3.1 | EL <i>RMP</i> selecciona la opción <i>Cerrar</i> . |
| 4.1 | Si existe alguna excepción en los datos del usuario nuevo el <i>sistema</i> despliega la información referente a la excepción. |
| 4.2 | El <i>RMP</i> selecciona la opción <i>Aceptar</i> en la excepción. |
| 4.3 | Se continúa en el punto 2 del flujo primario. |

Precondiciones

| | |
|---|--|
| 1 | El <i>RMP</i> inició sesión previamente. |
|---|--|

Pos condiciones

| | |
|---|---|
| 1 | Los datos del nuevo usuario quedan almacenados en el <i>sistema</i> . |
|---|---|

Documentación Adjunta

| | |
|---|--------------------------|
| 1 | Diagrama de Caso de Uso. |
| 1 | Plan de Pruebas. |
| 1 | Prototipo de Interfaz. |



Figura 14: Diagrama del Caso de Uso: CU-01.1

Plan de pruebas del CU-01.1

| Entrada | Esperado |
|--|---|
| Seleccionar la opción <i>Guardar</i> . | Cierre de la pantalla de Datos de Usuario, el usuario nuevo queda registrado en el <i>sistema</i> . |
| Seleccionar la opción <i>Cerrar</i> . | Cierre de la pantalla de Datos de Usuario. |
| De existir una excepción. | El <i>sistema</i> despliega información de la excepción, el usuario no se guarda hasta corregir la excepción. |

Prototipo de Interfaz para el CU-01.1

Figura 15: Pantalla Datos de Usuarios.

4.3 ARQUITECTURA Y DISEÑO DE LOS MÓDULOS

Durante la fase de diseño del sistema se realizaron algunos diagramas de estado para identificar el flujo de pantallas. Se elaboraron diagramas de secuencia con la finalidad de identificar los objetos y clases de cada escenario de los casos de uso, y posteriormente de modelar la interacción entre los objetos. Con estos diagramas se validaron los casos de uso. La

Figura 16 muestra un ejemplo del diagrama de estado y la Figura 17 muestra un ejemplo de un diagrama de secuencia del CU-01.1 Alta de Usuarios.

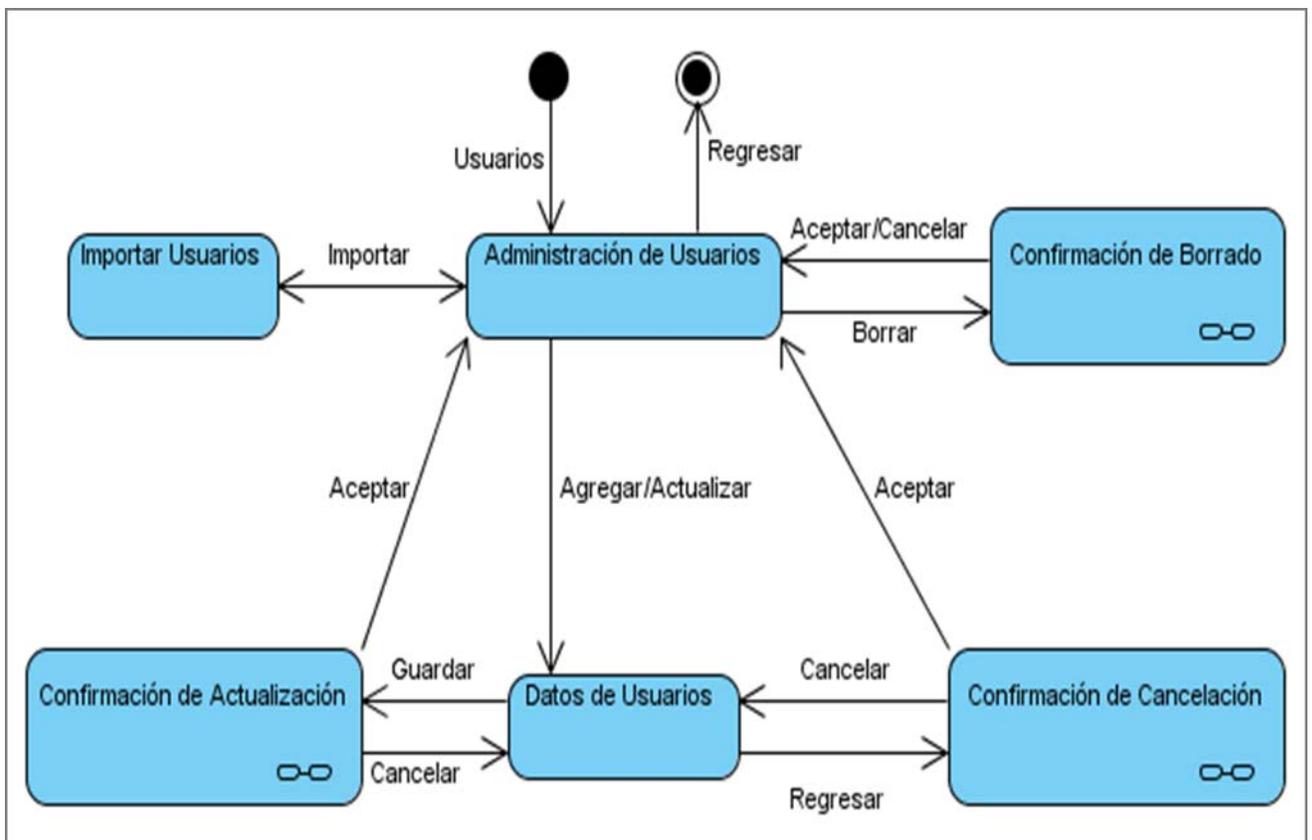


Figura 16: Diagrama de estado para el CU-01.1 Alta de usuarios.

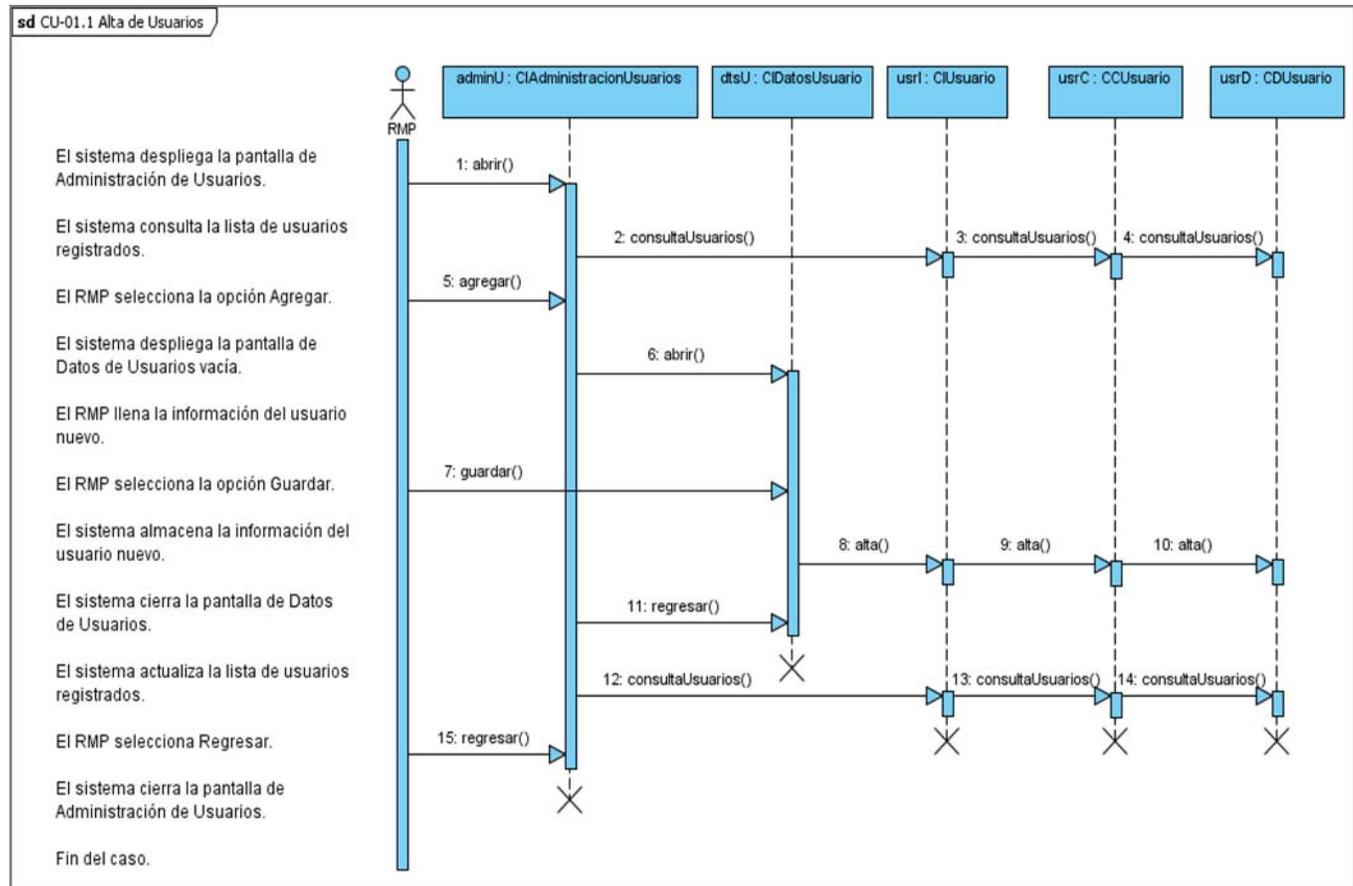


Figura 17: Diagrama de secuencia para el CU-01.1 Alta de usuario.

Para facilitar el mantenimiento futuro de la herramienta, se aplicó la arquitectura de tres capas:

1. La capa de interfaz contiene toda la lógica para recibir las peticiones del cliente y es la encargada de la presentación de los datos gráficamente. Con la finalidad de producir una herramienta rica gráficamente, se decidió usar tecnología para crear Aplicaciones de Internet Ricas (Rich Internet Applications, RIA), por este motivo la capa de interfaz se divide en 2 subcapas: La primera subcapa está desarrollada en Flex 3.0. y la segunda subcapa está desarrollada en JAVA JEE con implementación de JSP.
2. La capa de control contiene todas las clases relacionadas con el comportamiento de la herramienta, por lo que representa su funcio-

nalidad conformando el núcleo de la lógica del sistema. Esta capa está desarrollada en JAVA JEE.

3. La capa de datos se encarga de la persistencia de la información manejada por la herramienta. Esta capa está conformada por una base de datos activa manejada por el gestor de bases de datos Oracle. La base de datos es activa ya que incluye el uso de disparadores (triggers) para la creación o actualización de vistas (views), estas vistas contienen información general de la implementación en un instante de tiempo dado.

Las tres capas pueden residir en una misma computadora, pero si es necesario, cada capa puede residir en diferentes computadoras, distribuyendo de esta manera las cargas producidas por estrés, logrando dar soporte a un mayor número de usuarios. La

Figura 18 muestra una de las posibles formas de distribución de los componentes de GENESIS.

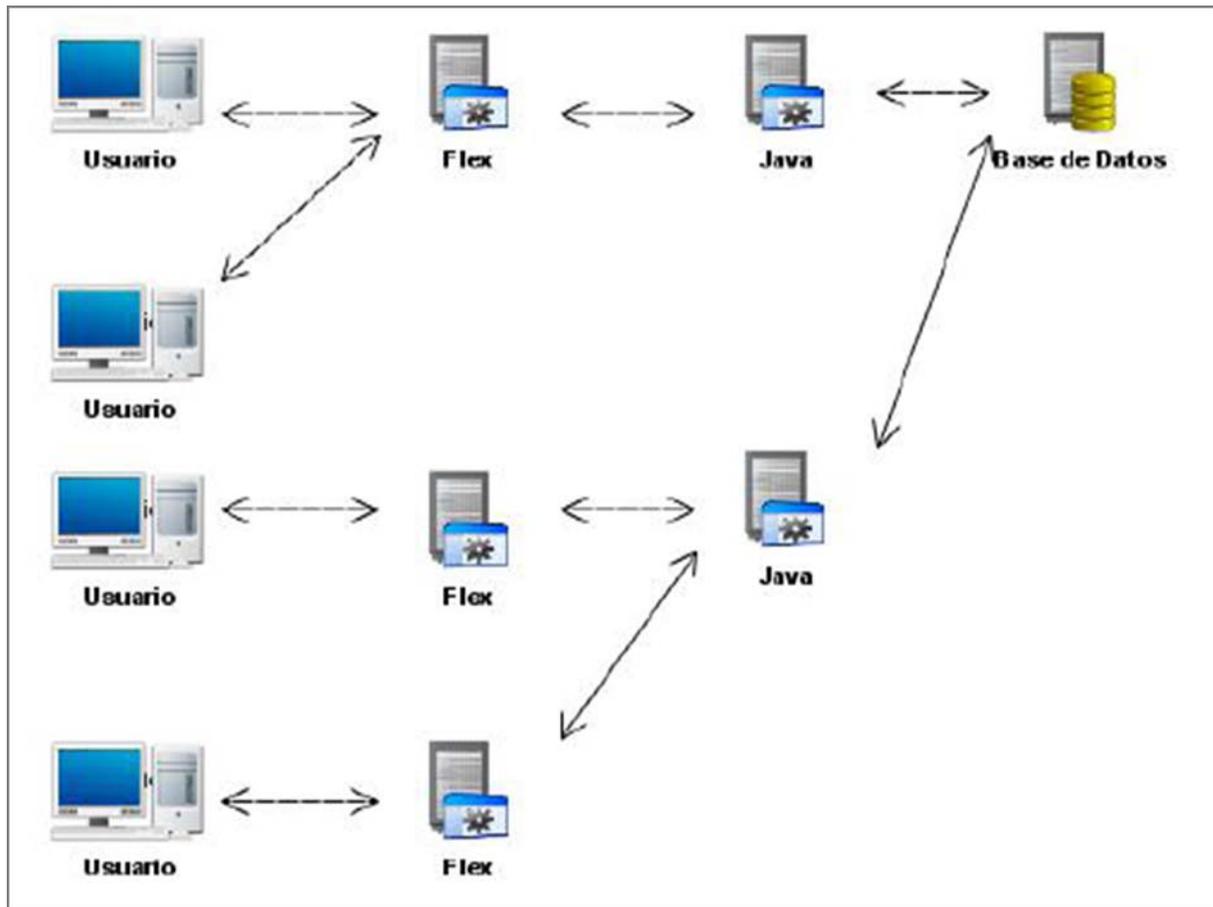


Figura 18: Forma posible de distribución de los componentes de GENESIS.

En el desarrollo de GENESIS se aplican diversos patrones de diseño para asegurar la mejor constitución del sistema, entre los patrones utilizados destacan:

- Conexión a bases de datos, Objeto de Acceso a Datos (Data Access Object, DAO) que sirve para desacoplar las clases de datos de la implementación, del método de almacenamiento persistente.
- Objetos de Valores (Value Objects, VO) permite crear un objeto para encapsular los valores de una entidad (normalmente entidades de la capa de datos) para su fácil transferencia entre capas.
- Decorador (Decorator) que sirve para reconocer y almacenar diseños que el usuario crea con elementos del sistema.
- Fabricación de elementos abstractos (Abstract Factory) que permite controlar funcionalidades comunes de elementos.

La distribución de las capas está asociada al paquete en que se desarrollaron las clases, la

Figura 19 muestra el diagrama de distribución de los paquetes del sistema.

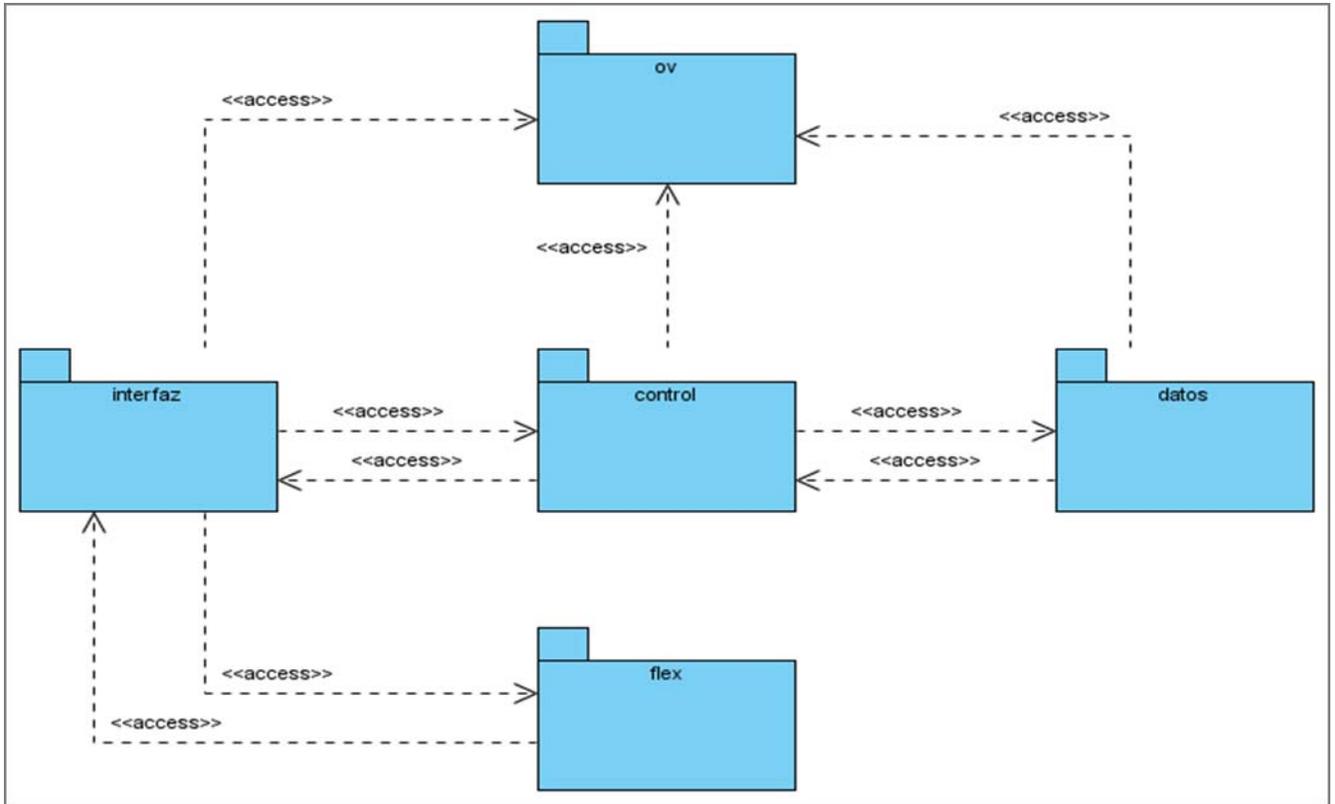


Figura 19: Diagrama de paquetes

4.3.1 DISEÑO DE LA BASE DE DATOS

Se utilizó Oracle® como manejador de base de datos, el sistema es capaz, de comprobar la integridad, borrar y crear la base de datos si es necesario y de manera automática. La base de datos usada en génesis está diseñada usando el modelo entidad-relación y está estructurada para almacenar históricamente información relevante para el conocimiento de la organización a continuación se muestran diagramas de la base de datos diseñada para ser usada en este sistema.

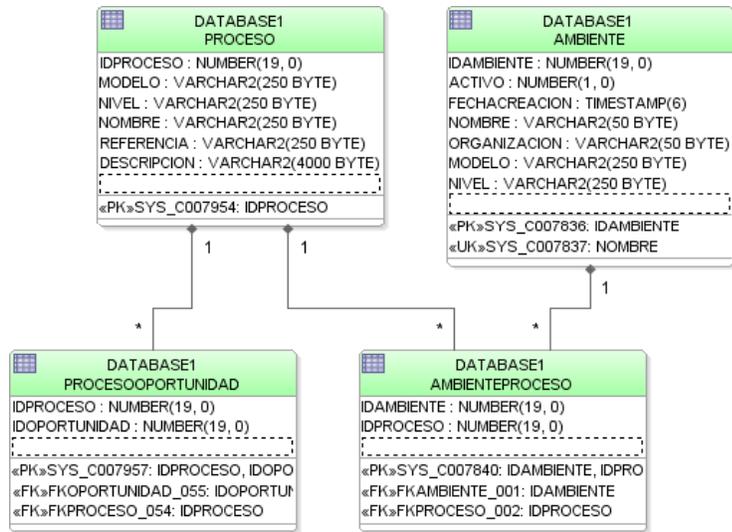


Figura 20: Tablas relacionadas al ambiente.

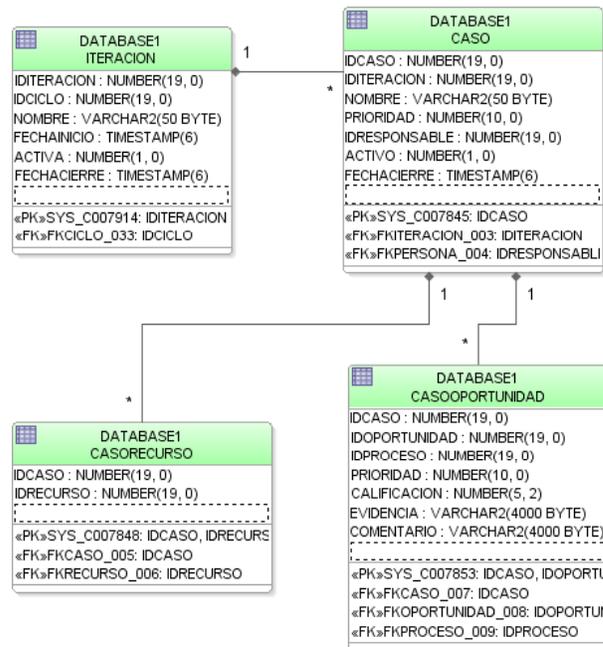


Figura 21: Tablas relacionadas con el caso de mejora.

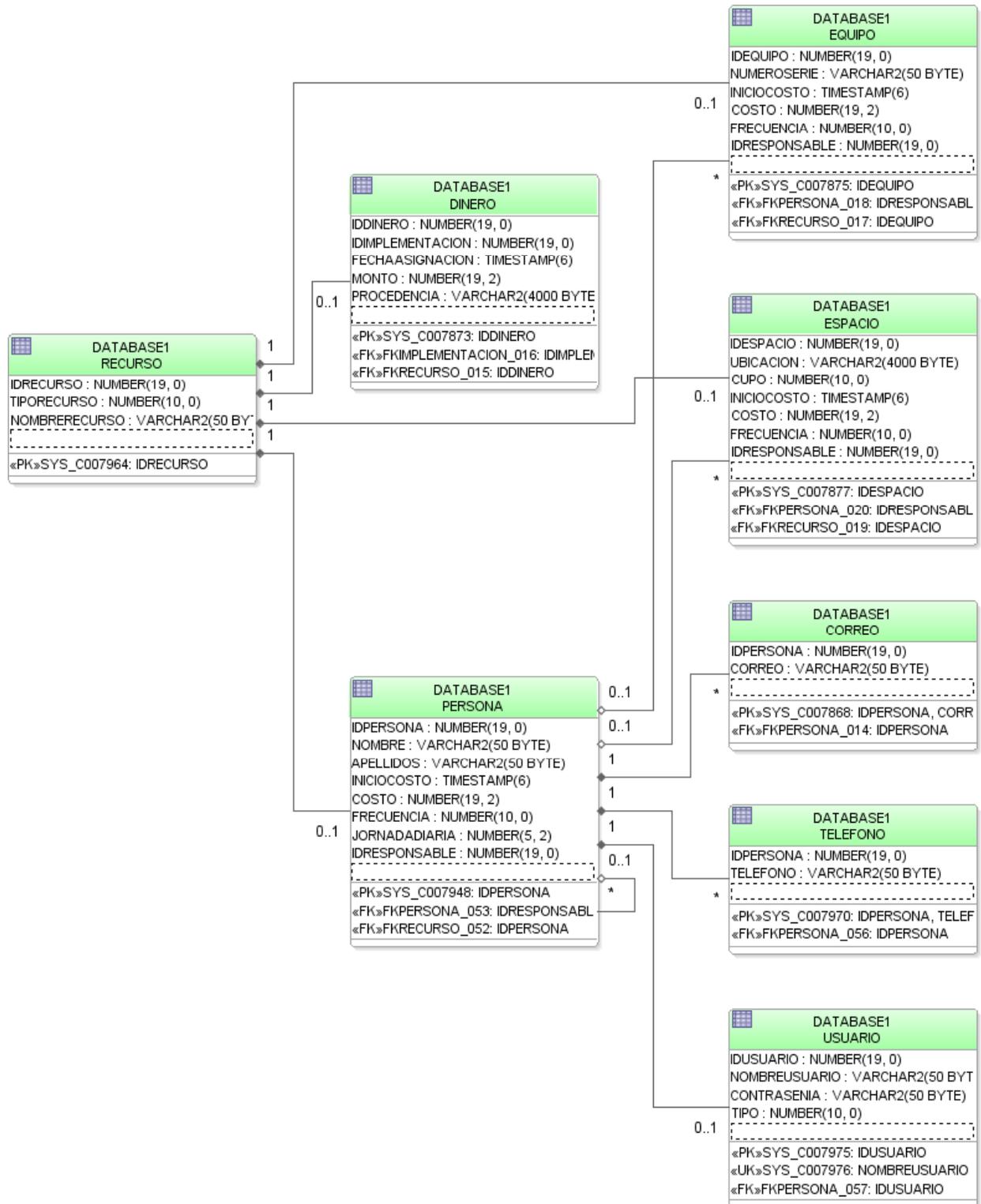


Figura 22: Tablas relacionadas con los recursos.

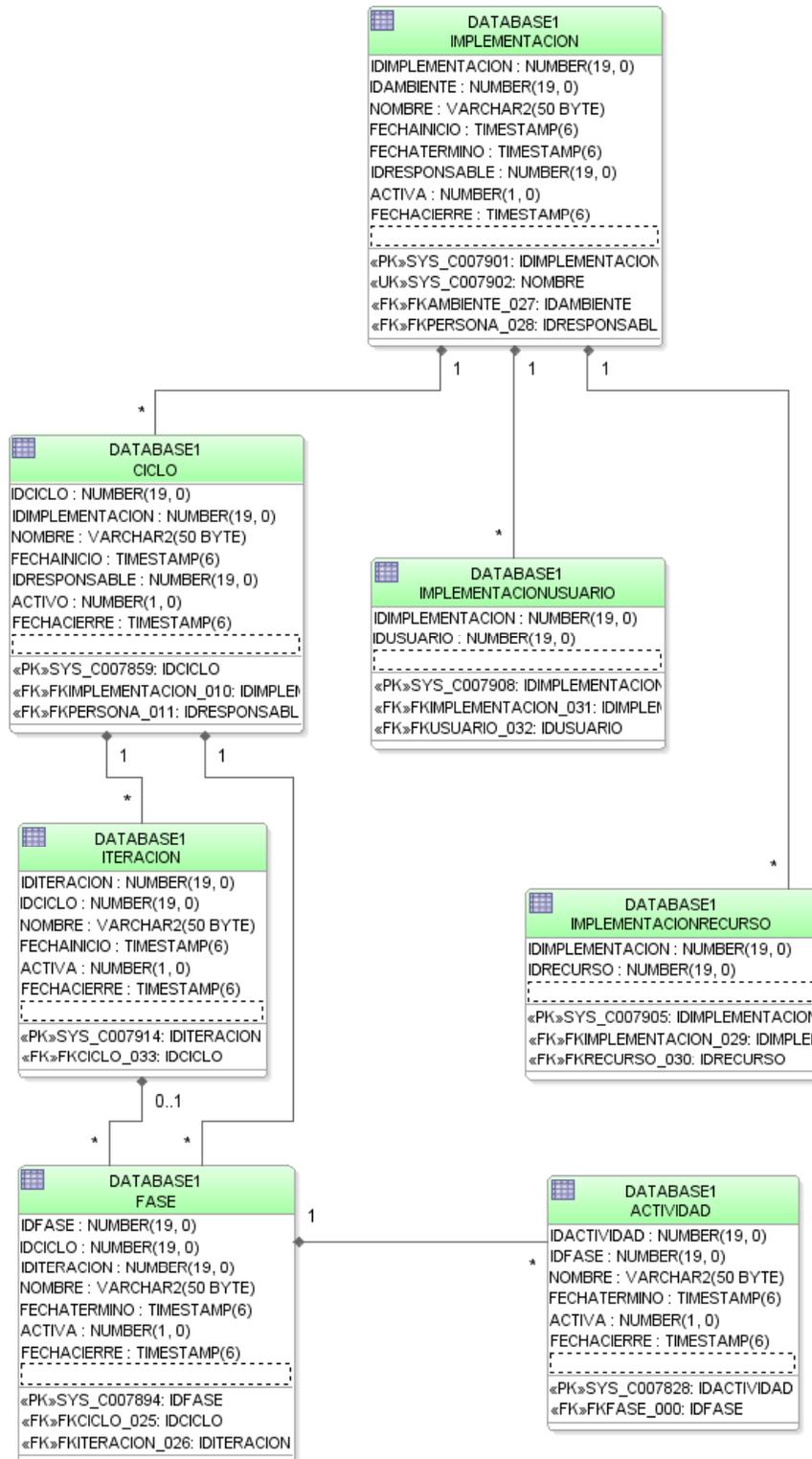


Figura 23: Tablas relacionadas con la implementación de mejora.

4.4 CONSTRUCCIÓN Y PRUEBAS

En las primeras etapas de la construcción del sistema se realizó una prueba de la arquitectura, dicha prueba sirvió para confirmar el método e interfaces necesarias para realizar la conexión entre las diferentes tecnologías utilizadas (Adobe® Flex®, Java, Oracle®), esto permitió implementar cada caso de uso de manera individual haciendo independientes entre sí los casos de uso.

Para cada caso de uso se validó su comportamiento realizando el plan de pruebas definido en su detalle de caso de uso. Estas pruebas de manera unitaria fueron utilizadas solo para corroborar la construcción y comportamiento correctos de las clases, tablas y vistas asociadas a cada caso de uso, como ejemplo se muestra el plan de pruebas ejecutado para el caso de uso 01.1.

Plan de pruebas del CU-01.1

| Entrada | Esperado | Obtenido |
|--|---|----------|
| Seleccionar la opción <i>Guardar</i> . | Cierre de la pantalla de Datos de Usuario, el usuario nuevo queda registrado en el <i>sistema</i> . | Correcto |
| Seleccionar la opción <i>Cerrar</i> . | Cierre de la pantalla de Datos de Usuario. | Correcto |
| De existir una excepción. | El <i>sistema</i> despliega información de la excepción, el usuario no se guarda hasta corregir la excepción. | Correcto |

4.5 EPFCOMPOSER, OBTENCIÓN DE DATOS DEL MODELO DE REFERENCIA DE PROCESOS

4.5.1 MODELADO DE PROCESOS

Los Modelos de Referencia de Procesos de la industria de software poseen entre ellos muchas similitudes en cuanto a los conceptos abstractos que los definen, (roles, tareas, productos, etc.) para definir un proceso y que sea fácilmente entendible se utiliza un lenguaje unificado que permite modelarlo y así permite definir, gestionar y reutilizar aspectos comunes de métodos y procesos.

Uno de estos lenguajes de modelado de procesos es SPEM “Software & Systems Process Engineering Metamodel (7)” este lenguaje contiene elementos que permiten definir participantes, productos de entrada y salida, actividades y tareas de un proceso.

4.5.2 EPF COMPOSER

EPFComposer (46) es una herramienta de código abierto diseñado para que los ingenieros de proceso y los directores de proyecto puedan adaptar y publicar los métodos y procesos para sus organizaciones.

Soporta estándares como XML Meta Interchange (XMI), Unified Modeling Language (UML), Extensible Stylesheet Language Transformation (XSLT) y Dynamic HTML (DHTML), utiliza una biblioteca de clases llamada Unified Method Architecture (UMA), esta biblioteca define el meta-modelo para el contenido y la forma de los procesos. Ha sido desarrollado como un método de unificación de los diferentes lenguajes de la ingeniería de procesos como SPEM, el utilizado para IBM Rational RUP v2003, Unified Process, IBM Global Services Method, así como IBM Rational Summit Ascendant.

Como tal, ofrece conceptos y capacidades de todas las fuentes de estos modelos en la unificación de una manera coherente, pero que permite expresar cada uno de estos métodos fuente con sus características específicas. Una vez modelados permite exportar el modelo en un archivo XML que contiene la estructura UMA asociada al modelo.

4.5.3 CARGA DE MODELOS DE REFERENCIA DE PROCESOS

Para permitir que GENESIS utilice el modelo de un archivo generado en el EPFComposer se creó una biblioteca que permite recuperar un archivo de modelo UMA y reconstruirlo como objetos dentro de la máquina virtual, estos objetos simulan la estructura definida en la biblioteca UMA que la creó. Utilizando los esquemas definidos para el EPF como validadores de los XML. Por lo que a la fecha GENESIS puede traducir XML realizados con el EPFComposer 1.5.0

Capítulo 5

UN PASEO POR GENESIS



5.1 REGISTRO DE RECURSOS Y USUARIOS

Dentro de las necesidades de administración comunes durante la ejecución de un proceso de mejora que el RMP debe realizar, se encuentra el control de los recursos que se asignan a los proyectos de mejora, así como los costos que son generados por dichos recursos y que están específicamente relacionados al esfuerzo de mejora.

Considerando esta necesidad, el sistema GENESIS permite la administración de 3 tipos de recursos:

- **Recurso humano:** representa directamente a las personas involucradas con la mejora. Permite almacenar información de contacto (correo electrónico, teléfonos), información jerárquica (líder inmediato o superior) así como información del costo económico por unidad de tiempo (salario), este es el costo que representa para la mejora. (Ver Figura 25: Pantalla para captura de Datos de Recurso Humano).

The screenshot shows a web form titled "Datos de Recurso Humano". The form contains the following fields and controls:

- Nombre ***: Text input field.
- Apellidos ***: Text input field.
- Superior**: Dropdown menu with "Ninguno" selected.
- Información Económica**: Checked checkbox.
- Contrato**: Dropdown menu with "Mes" selected.
- Salario**: Text input field with "0.00" entered.
- Jornada**: Two spinners for "Horas" (set to 8) and "Dias" (set to 5).
- Inicio**: Date picker showing "05/01/2009".
- Teléfonos**: Text input field with a "+" icon for adding more.
- Correo**: Text input field with a "+" icon for adding more.

Figura 25: Pantalla para captura de Datos de Recurso Humano.

- **Equipo de Trabajo:** permiten representar las herramientas físicas ó electrónicas que se utilizan para llevar a cabo el esfuerzo de mejora (Computadoras de escritorio o portátiles, impresoras, software especializado, etc.). Permitted almacenar información de identificación única (Numero de Serie), información responsiva

única (Numero de Serie), información responsiva (persona responsable o encargada) y por la naturaleza de estos recursos, se puede registrar el costo económico por unidad de tiempo ó en una sola exhibición. (Ver Figura 26: Pantalla para la captura de Datos de Recurso Equipo.)

The screenshot shows a web form titled "Datos de Recurso Equipo". The form contains the following elements:

- Nombre ***: A text input field.
- #Serie**: A text input field.
- Responsable**: A dropdown menu with "Ninguno" selected.
- Información Económica**: A checked checkbox.
- Frecuencia**: A dropdown menu with "Mes" selected.
- Costo**: A text input field containing "0.00".
- Inicio**: A date picker showing "05/01/2009".

Figura 26: Pantalla para la captura de Datos de Recurso Equipo.

- **Espacio Colaborativo**: permiten representar, lugares que se utilizan para llevar a cabo el esfuerzo de mejora (Salas de cómputo, Salones de conferencias, aulas de aprendizaje, etc.). GENESIS permite almacenar información de la cantidad de personas que alberga dicho lugar (Cupo), información responsiva (persona responsable, encargada o dueño) así como información del costo económico por unidad de tiempo (renta ó gastos generados por el lugar). (Ver Figura 27: Pantalla para la captura de Datos de recurso espacio colaborativo.)

Datos de Recurso Espacio

Nombre *

Ubicacion

Cupo

Responsable

Información Económica

Frecuencia

Costo

Inicio

Figura 27: Pantalla para la captura de Datos de recurso espacio colaborativo.

Existen 2 formas de registrar un recurso dentro de GENESIS, la primera es registrarlo en el ambiente. Con esto, el recurso estará disponible para asignarlo a una implementación de mejora (Configuración > Configuración de Recursos). (Ver Figura 28: Pantalla de Recursos del Sistema).

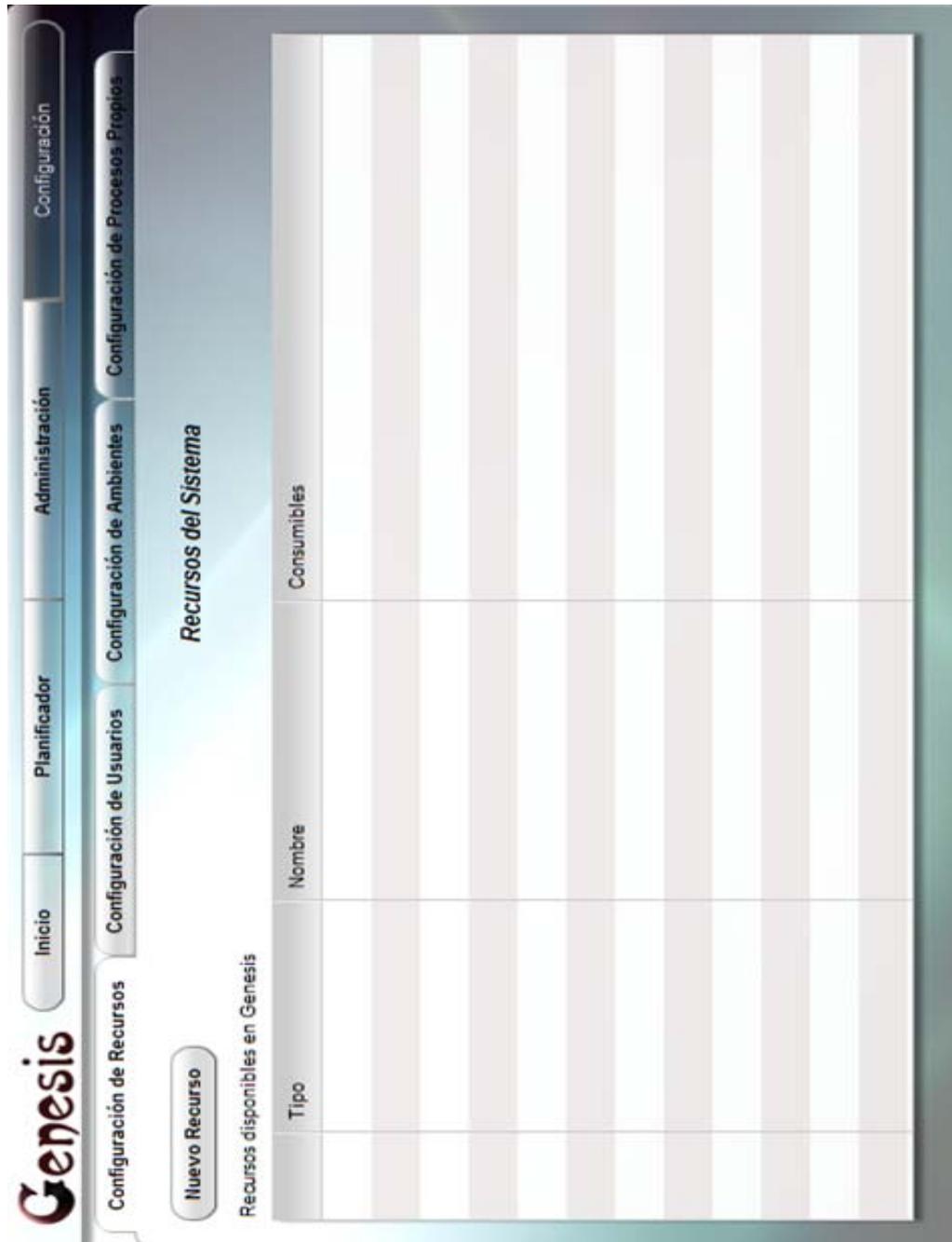


Figura 28: Pantalla de Recursos del Sistema.

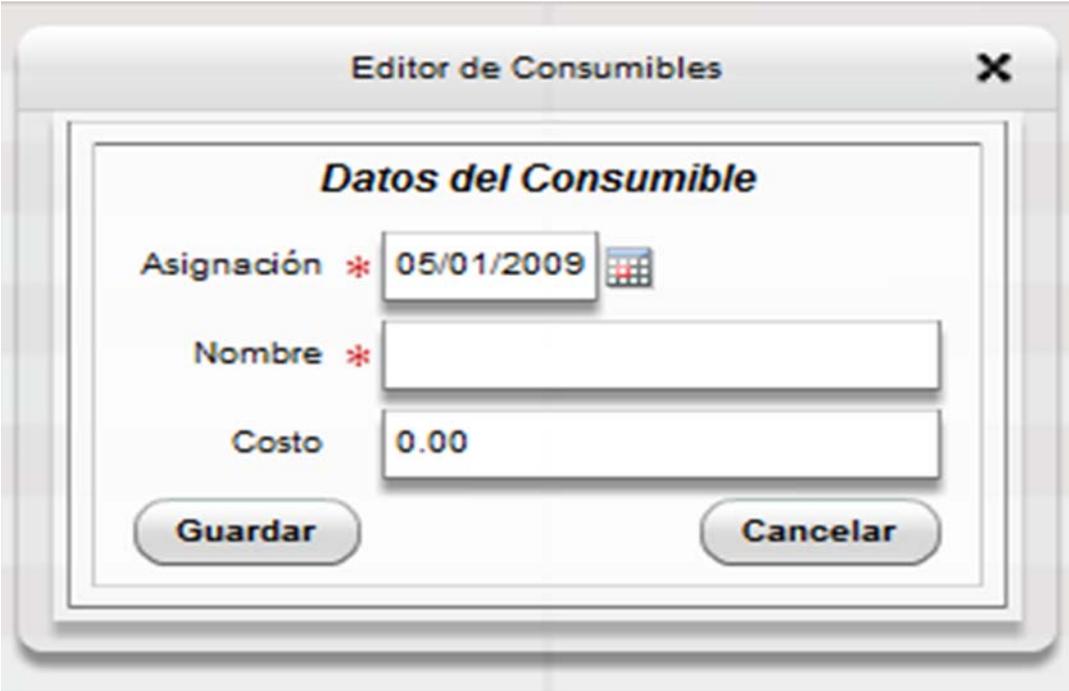
La segunda forma de registrar un recurso, es asignarlo a la implementación de mejora, con esto, el recurso se queda registrado en el ambiente directamente y se asigna a la implementación seleccionada en la pantalla inicial (Administración > Recursos de implementación). (Ver Figura 29 Pantalla de administración de recursos (Implementación)).

que estén asociados a un recurso y que no requieren darse de alta como recursos en sí.

Ejemplos de estos son, tóner para una impresora, la reparación ó expansión de un equipo de computo, bebidas y comidas para algún espacio colaborativo, viáticos ó capacitación para una persona, etc.

GENESIS permite registrar este tipo de gastos a través del concepto de **consumible**. Los consumibles están asociados a los recursos permitiendo almacenar información temporal (fecha de asignación), información descriptiva (Nombre que represente al consumible) así como información del costo económico.

Solo se pueden agregar consumibles a los recursos que estén asociados a una implementación, considerando que los recursos pueden estar asociados a diferentes implementaciones, se realiza una separación de los consumibles por la implementación a las que fueron asignados. (Ver Figura 30: Pantalla Editor de Consumibles.



The image shows a software window titled "Editor de Consumibles" with a close button (X) in the top right corner. The window contains a form titled "Datos del Consumible" with the following fields:

- Asignación ***: A date field containing "05/01/2009" and a calendar icon.
- Nombre ***: An empty text input field.
- Costo**: A text input field containing "0.00".

At the bottom of the form, there are two buttons: "Guardar" (Save) and "Cancelar" (Cancel).

Figura 30: Pantalla Editor de Consumibles.

Existe un tipo de recurso que solo puede ser creado desde la *Pantalla de Administración de Recursos (Implementación)*.

Figura 29.

Este es el recurso económico, que a diferencia de los demás recursos, representa dinero asignado a la implementación de mejora. Permitiendo almacenar información de la cantidad de dinero asignada, información de la fecha en la que fue asignado a la implementación de mejora, así como la información de la procedencia (partida, presupuesto, etc.). (Ver *Figura 31: Pantalla para los Datos de Recurso Dinero*).

Figura 31: Pantalla para los Datos de Recurso Dinero.

El sistema ahora contiene información sobre los recursos que se usarán en el sistema, y es muy posible que algunas de las personas registradas en el sistema tengan que hacer uso del mismo. Es por esto que GENESIS permite la creación de usuarios del tipo RMP o del tipo RCM

Un usuario de GENESIS es un recurso humano, por lo tanto hay 2 formas de crear un usuario, la primera forma de registrarlo es asociando un usuario a un recurso humano creado previamente. La segunda forma es crear un usuario nuevo, de ser así, el sistema creará automáticamente un recurso humano y le asociará los datos del usuario. Ambas formas están en la misma pantalla. (Ver

Figura 32: Pantalla para captura de Datos de Usuario.

Datos de Usuario

Recurso **Nuevo** ▼

Tipo **Responsable de Mejora de Procesos** ▼

Nombre *

Apellido *

Superior **Ninguno** ▼

Información Económica

Contrato **Mes** ▼

Salario

Jornada Horas Dias

Inicio

Implementaciones disponibles

Implementación

Acceso a **Implementación**

Usuario *

Contraseña *

Teléfonos

Correos

Figura 32: Pantalla para captura de Datos de Usuario.

5.2 PLANIFICACIÓN DE MEJORAS

Recordemos que GENESIS es un sistema para mejora de procesos basado en PmCompetiSoft, es por esto que la estructura en la que están separadas las actividades corresponden a las definidas en dicho proceso de mejora.

En la primera etapa de una mejora, se requiere hacer una planificación de la implementación de mejora y las fases que contendrán, también se requiere identificar que procesos se desean mejorar. Para estas necesidades GENESIS permite definir un ambiente, que básicamente es un entorno fijo donde se pueden definir una o varias implementaciones de mejora.

A dicho ambiente se puede asociar el nombre de la organización, el modelo de referencia a usar, el nivel de capacidades y los procesos que se desean mejorar. (Ver Figura 33: Pantalla para la captura de los Datos de Ambiente.

Datos de Ambiente

Activo

Nombre *

Organización *

Modelo **Competi Soft v0.6** ▼

Nivel **Nivel 1 Realizado** ▼

Procesos disponibles para Nivel 1 Realizado

| Nombre | Modelo |
|--|------------------|
| AP. Administración del Proyecto | CompetiSoft v0.6 |
| DS. Desarrollo de Software | CompetiSoft v0.6 |
| GBSI. Gestión de Bienes, Servicios e Infraestructura | CompetiSoft v0.6 |
| GC. Gestión de Conocimiento | CompetiSoft v0.6 |
| GCP. Gestión de Cartera de Proyectos | CompetiSoft v0.6 |
| GN. Gestión de Negocio | CompetiSoft v0.6 |

Procesos a mejorar

| Nombre |
|--------|
| |
| |
| |
| |
| |
| |

Figura 33: Pantalla para la captura de los Datos de Ambiente.

Los procesos que GENESIS puede gestionar son los modelados en la herramienta EPFComposer.

GENESIS permite gestionar al mismo tiempo los procesos del modelo de referencia de procesos y los procesos propios de la organización. Estos procesos son cargados por GENESIS a la biblioteca de procesos propios (Ver Figura 34: Pantalla de la Biblioteca de Procesos Propios).



Figura 34: Pantalla de la Biblioteca de Procesos Propios.

Una vez creado el ambiente, se puede proceder a realizar la planificación de la implementación de mejora, usando la estructura del proceso PmCompetiSoft, que permite poner fechas y distribuir en el tiempo una implementación. GENESIS permite crear una implementación la cual se asocia

a un ambiente, se asigna fecha de inicio y de término y opcionalmente un responsable. Una vez creada, la implementación aparece un icono que la representa y que permite seleccionarla (clic) o ver su contenido (doble clic). (Ver Figura 35: Pantalla Datos de Implementación y Navegador de Implementaciones).

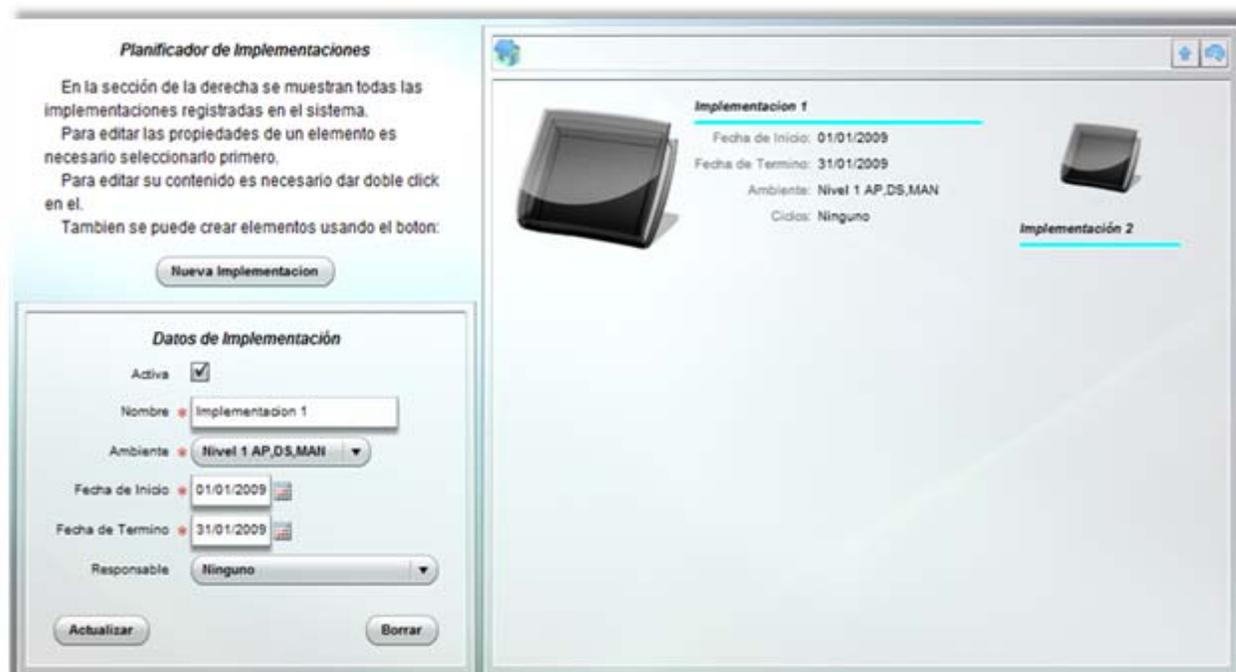


Figura 35: Pantalla Datos de Implementación y Navegador de Implementaciones.

Dentro de una implementación de mejora, el proceso de mejora PmCompetiSoft define el concepto de ciclo que es un conjunto de 5 fases de mejora (Instalación, Diagnóstico, Formulación, Ejecución, Revisión), GENESIS permite crear ciclos usando una plantilla de PmCompetiSoft. La creación de los ciclos se realiza a través del editor de ciclos en 7 pasos.

Paso 1: Se debe asignar un nombre al ciclo de mejora, así como una fecha de inicio y opcionalmente asignar un responsable. (Ver Figura 36: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 1)).



Figura 36: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 1).

Paso 2: Se debe asignar una fecha de término a la fase de Instalación de ciclo, ya que su fecha de inicio es igual a la fecha de inicio del ciclo. (Ver Figura 37: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 2)).



Figura 37: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 2).

Paso 3: Se debe asignar una fecha de término de la fase de Diagnóstico de Proceso ya que su fecha de inicio es igual a la fecha de término de la fase de Instalación de ciclo. (Ver Figura 38: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 3)).

Figura 38: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 3).

Paso 4: Se debe asignar una fecha de inicio a la Iteración de mejora. De acuerdo al proceso PmCompetiSoft una iteración está formada por 5 actividades agrupadas en 2 fases (Formulación de mejoras y Ejecución de mejoras), un ciclo de mejora debe contener al menos una iteración de mejora.

En caso de existir más de una iteración de mejora dentro de un ciclo de mejora, se pueden ejecutar de forma secuencial o de forma paralela (Ver Figura 39: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 4)).



Figura 39: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 4).

Paso 5: Se debe asignar una fecha de término a cada una de las 5 actividades que conforman una iteración de mejora. (Ver Figura 40: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 5)).

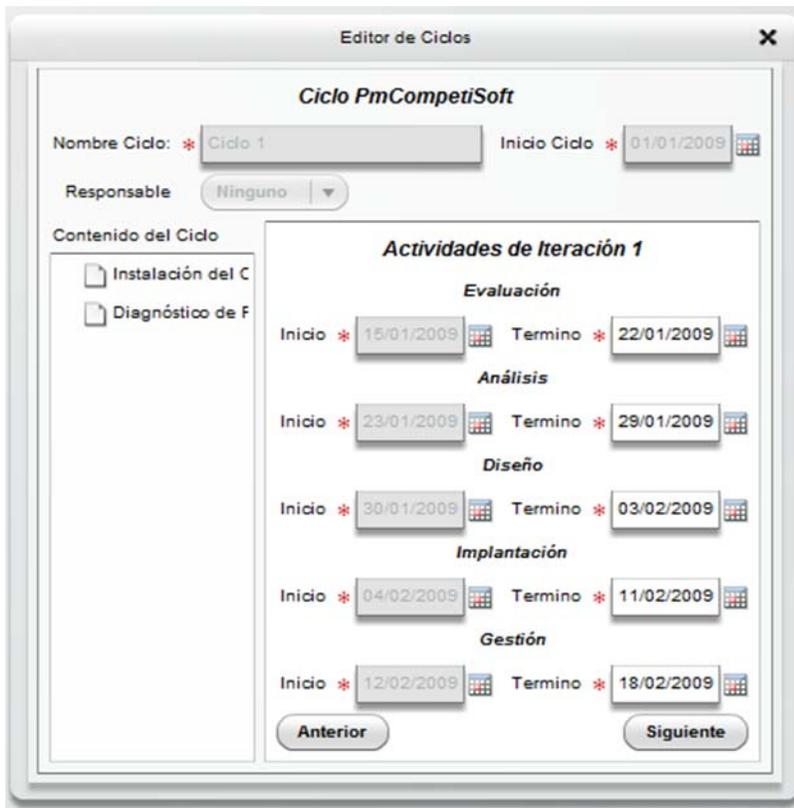


Figura 40: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 5).

Paso 6: En este paso se pueden crear nuevas iteraciones de mejora, para lo cual es necesario seleccionar la fecha de inicio de la nueva iteración de mejora, posteriormente se selecciona "nueva iteración", esto nos regresa al paso 5, si no se desean mas iteraciones de mejora se continua en el siguiente paso. (Ver Figura 41: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 6)).

Figura 41: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 6).

Paso 7: Este es el último paso, aquí se asigna la fecha de término para la fase Revisión del Ciclo y se crea el ciclo de mejora. (Ver Figura 42: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 7)).



Figura 42: Plantilla PmCompetiSoft Editor de Ciclo de mejora (Paso 7).

GENESIS utiliza esta plantilla para crear los ciclos de mejora según lo definido en el proceso PmCompetiSoft, sin embargo está diseñado para personalizar esta estructura permitiendo retirar y agregar los elementos que la conforman (fases), esto en caso de ser necesario en implementaciones de mejora más experimentadas.

GENESIS representa los ciclos contenidos en una implementación como carpetas. (Ver Figura 43: Contenido de la Implementación). Dentro de un ciclo creado usando la plantilla podemos encontrar las representaciones de las fases e iteraciones en forma de elementos puntuales y carpetas respectivamente. (Ver Figura 44: Contenido del Ciclo).

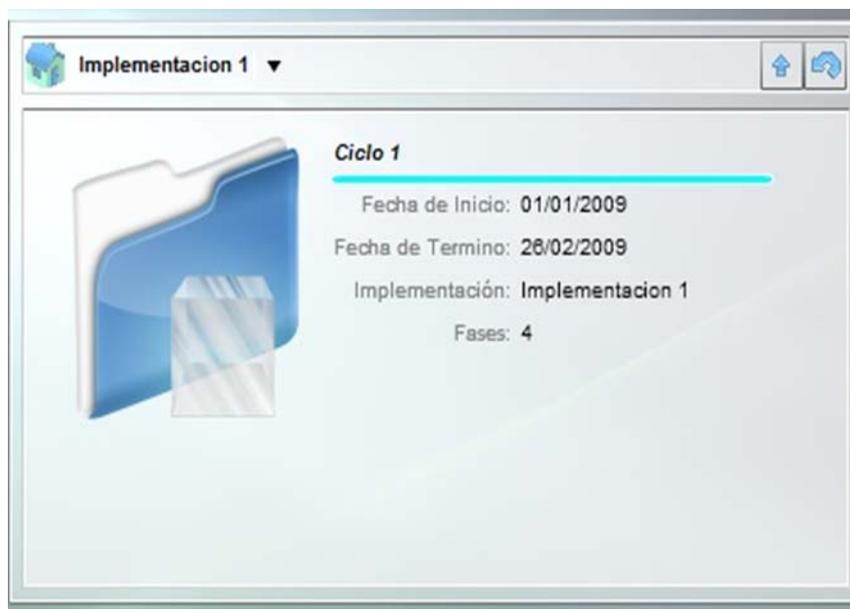


Figura 43: Contenido de la Implementación.

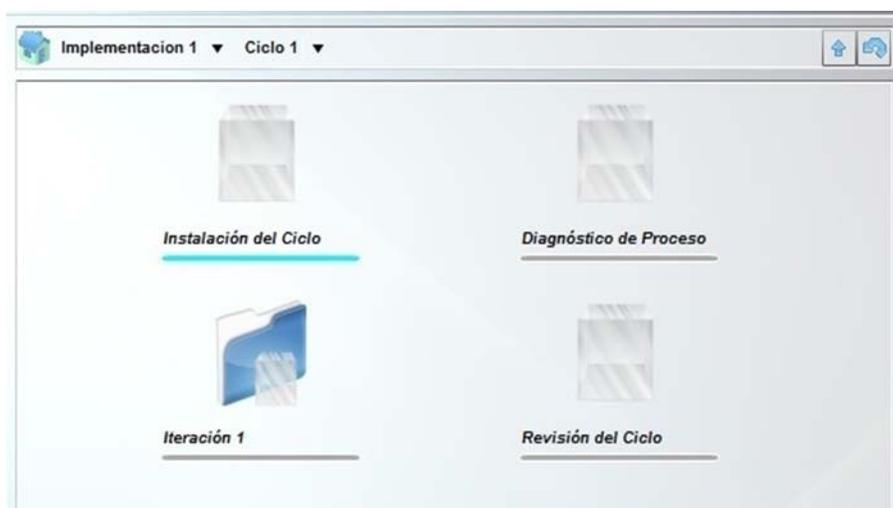


Figura 44: Contenido del Ciclo.

Dentro de una iteración de mejora creada con la plantilla encontramos 2 fases, en esta ocasión las fases no son representadas como elementos puntuales. (Ver Figura 45: Contenido de la Iteración. Esto es porque una iteración de mejora está integrada por las fases: Formulación de mejora y Ejecución de mejoras.



Figura 45: Contenido de la Iteración.

En la *Figura 46: Contenido de la fase Formulación de Mejora*, se muestran las actividades que contiene la fase Formulación de mejoras.



Figura 46: Contenido de la fase Formulación de Mejora.

Y en la *Figura 47: Contenido de la fase Ejecución de Mejoras*, se muestran las actividades correspondientes a la fase Ejecución de mejoras.



Figura 47: Contenido de la fase Ejecución de Mejoras.

De esta forma se genera una planificación y distribución de tiempos para una implementación de mejora en GENESIS.

5.3 APLICACIÓN DE UNA VALORACIÓN RÁPIDA

Durante la ejecución de mejoras es necesario evaluar de manera continua el estado de las oportunidades de mejora y en cada revisión del ciclo, es importante hacer una comparación del estado anterior y posterior de los procesos. Desafortunadamente una evaluación formal debe ser hecha por un revisor externo, requiere mucho tiempo y tiene costos elevados. Es por esto que PmCompetiSoft y muchos procesos ágiles, sugieren la aplicación de valoraciones rápidas, que son valoraciones realizadas por personal de la propia organización para diagnosticar de manera temprana los cambios en las oportunidades de mejora y tomar acciones correctivas lo antes posible.

Con la finalidad de no despreciar las evaluaciones robustas, pero presentar agilidad en su uso, GENESIS en su módulo de evaluación, permitirá en versiones futuras la importación de evaluaciones generadas por herramientas especializadas en evaluación de procesos. Por el momento se incluye una utilería para realizar valoraciones rápidas a través de la aplicación de un cuestionario (Ver Figura 48: Pantalla del Cuestionario de Valoración. Este cuestionario permite registrar para cada ciclo de mejora valoraciones de los procesos a mejorar en la implementación de mejora.

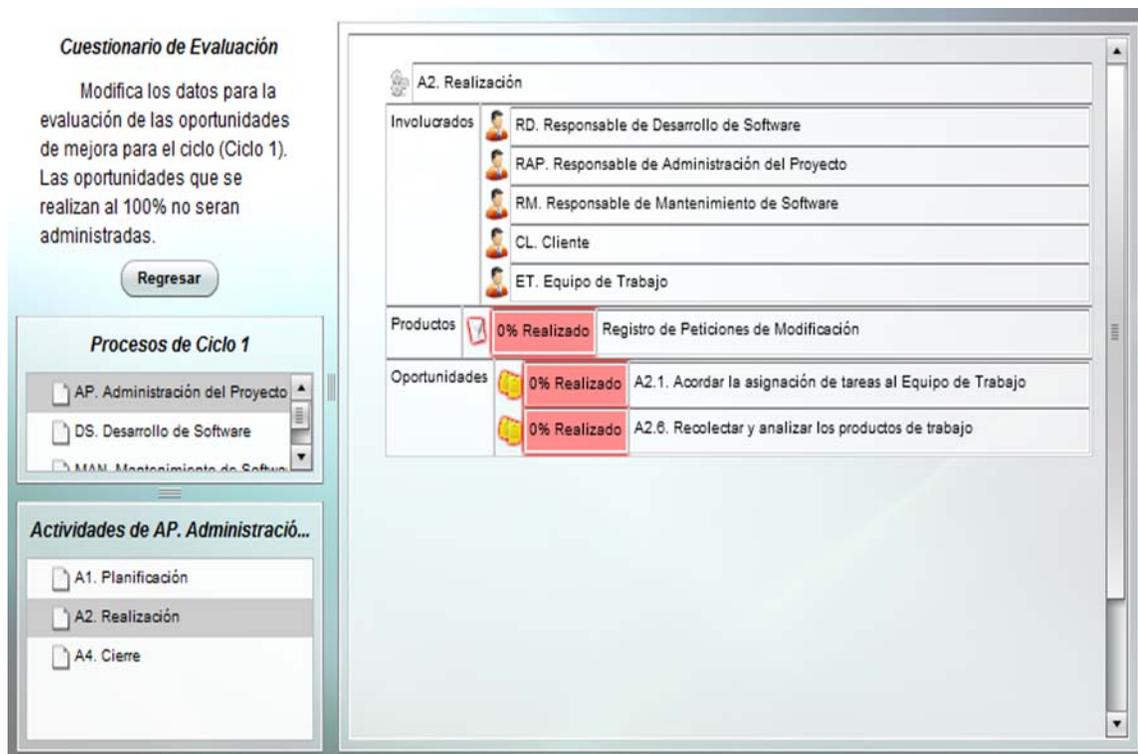


Figura 48: Pantalla del Cuestionario de Valoración.

Durante la aplicación del cuestionario, se puede registrar para cada oportunidad de mejora el porcentaje realizado, así la evidencia física que respalda la valoración y comentarios al respecto de la oportunidad. (Ver Figura 49: Pantalla para captura de Datos de Valoración).

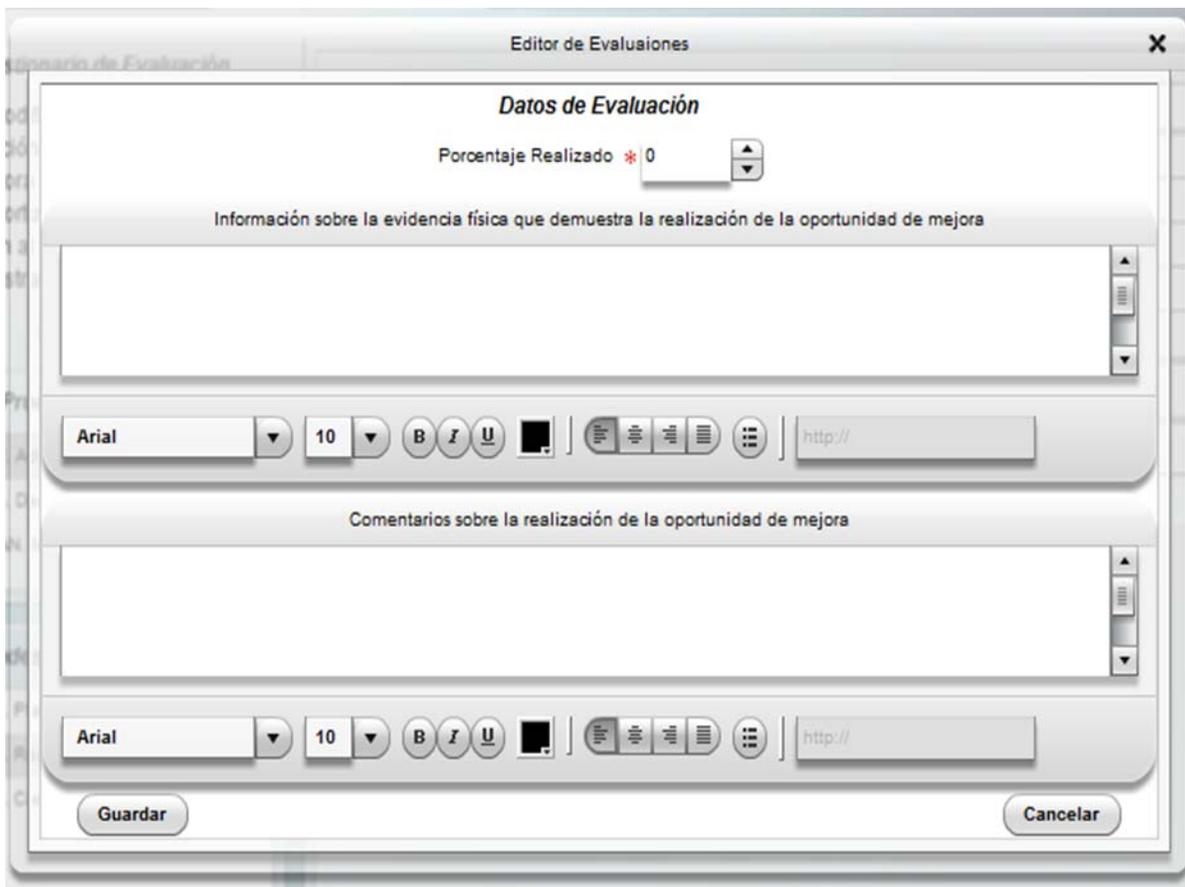


Figura 49: Pantalla para captura de Datos de Valoración.

5.4 CREACIÓN DE CASOS DE MEJORA

A partir de la aplicación de una valoración rápida, se pueden identificar las oportunidades de mejora, según lo sugerido por el PmCompetiSoft, las oportunidades de mejora se ejecutarán durante las actividades de la iteración de mejora. Estas oportunidades de mejora podrán ser agrupadas en casos de mejora.

Durante la creación de casos de mejora, se puede: registrar una prioridad de ejecución para el caso de mejora, elegir las oportunidades que se

incluirán en el caso de mejora, así como la prioridad individual de cada oportunidad de mejora relativa al caso. (Ver Figura 50: Pantalla Editor de Casos de mejora). En los casos de mejora solo pueden agregarse oportunidades de mejora que tengan menos del 100% de realización. Las oportunidades de mejora que estén calificadas como realizadas al 100% no aparecerán en la lista de oportunidades que se pueden agregar a un caso de mejora.

Estos casos de mejora serán asociados a una iteración de mejora, de esta manera, la ejecución de las oportunidades de mejora contenidas en el caso de mejora, tendrán ahora un tiempo en el cual realizarse, este tiempo es el tiempo planificado para la iteración y cada una de sus actividades.

The screenshot shows a window titled "Editor de Casos" with a close button (X) in the top right corner. The main content area is titled "Datos de Caso de Mejora" and contains the following elements:

- Nombre ***: A text input field.
- Prioridad ***: A dropdown menu with "Media" selected.
- Responsable**: A dropdown menu with "Ninguno" selected.
- Recurso**: A dropdown menu with a plus sign (+) to its right.
- Recursos**: A table with two columns: "Nombre" and an empty cell. The table is currently empty.
- Proceso**: A dropdown menu with "AP. Administración del Proyecto" selected.
- Actividad**: A dropdown menu with "Todas las actividades" selected.
- Oportunidad**: A dropdown menu with "Todas las oportunidades" selected.
- Prioridad ***: A dropdown menu with "Media" selected and a plus sign (+) to its right.
- Oportunidades**: A table with three columns: "Prioridad", "Nombre", and an empty cell. The table is currently empty.

At the bottom of the window, there are two buttons: "Guardar" on the left and "Cancelar" on the right.

Figura 50: Pantalla Editor de Casos de mejora (creación del caso).

Una vez creado el caso de mejora es deber de los RCM, reportar el avance de las oportunidades de mejora, Al final de la iteración de mejora, el caso de mejora se cierra con la actualización de los datos de las oportunidades de mejora. (Ver Figura 51: Pantalla Editor de Casos de mejora (cierre del caso).

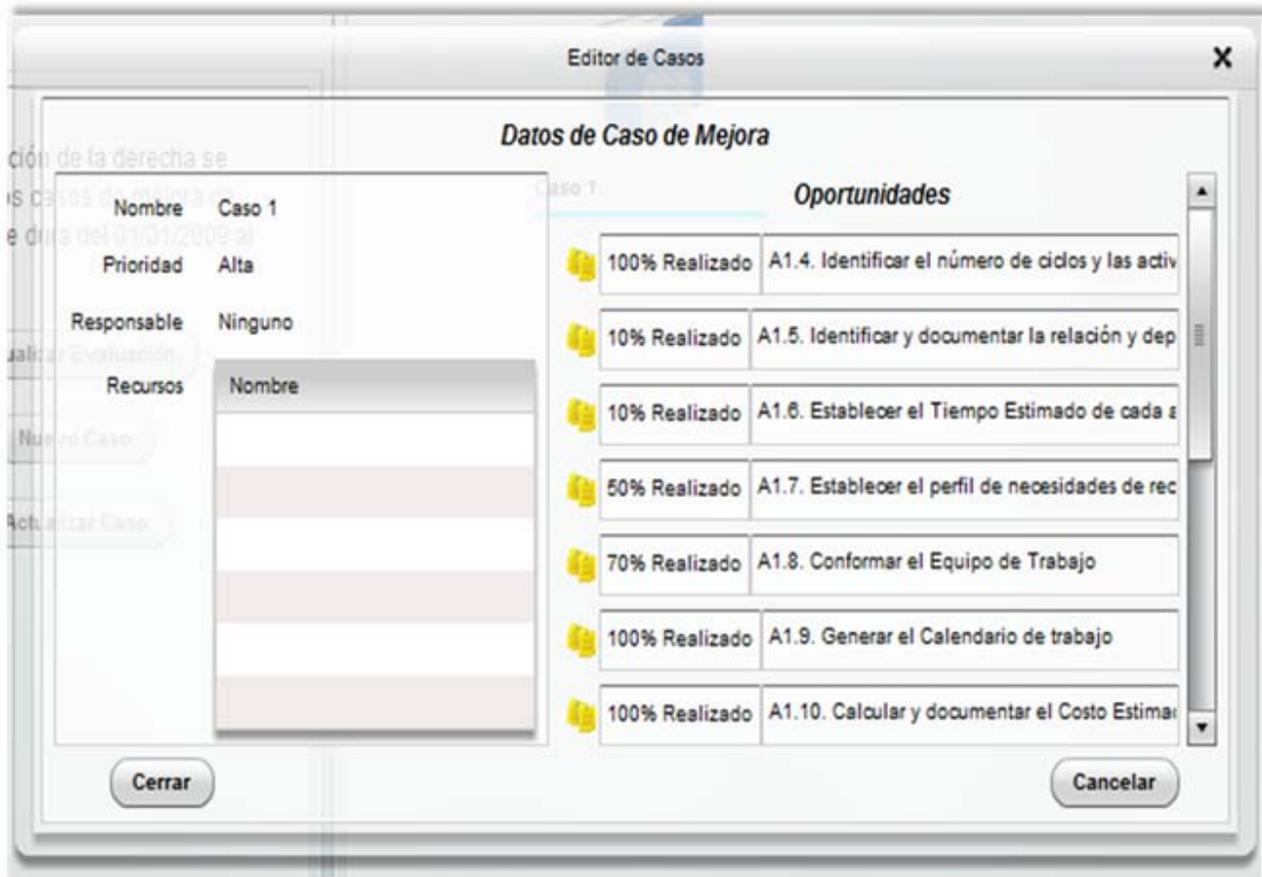
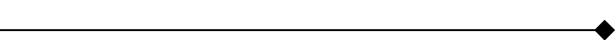


Figura 51: Pantalla Editor de Casos de mejora (cierre del caso).

GENESIS permite que el registro de esta información se pueda hacer remotamente y que la información se centralice para que los RCM y los RMP encargados de la gestión de las mejoras puedan tomar decisiones lo antes posible, Además permite tener como referentes a implementaciones anteriores que convierten a esta información en parte importante del conocimiento de la organización que lo utilice.

Conclusiones



RESULTADOS Y CONCLUSIONES

Se identificó la necesidad de contar con una metodología para la implantación de un modelo de referencia de procesos, por lo tanto se decidió utilizar el proceso PmCompetiSoft como guía para la implementación de mejora continua de procesos.

Se analizó la estructura del proceso PmCompetiSoft para obtener y definir un conjunto de funcionalidades en los 6 módulos, 3 de los cuales fueron desarrollados en este proyecto de tesis.

Como resultado del desarrollo de este proyecto de tesis, GENESIS tiene las siguientes funciones para cada módulo:

- **Inicialización**
 - ✓ Permite crear ambientes donde se puede seleccionar el modelo de referencia de procesos a utilizar.
 - ✓ Permite seleccionar el conjunto de procesos a mejorar, ya sean procesos pertenecientes a un modelo de referencia o procesos propios de la organización.
 - ✓ Permite importar una biblioteca con procesos propios modelados en SPEM mediante el uso de EPFComposer.
 - ✓ Permite administrar recursos humanos, recursos materiales y recursos económicos.
 - ✓ Permite gestionar los elementos definidos en el proceso PmCOMPE-TISOFT (implementaciones de mejora, ciclos de mejora, fases de mejora, actividades de mejora, iteraciones de mejora, casos de mejora, oportunidades de mejora).
 - ✓ Permite priorizar casos de mejora y oportunidades de mejora asociadas a los casos de mejora.

- **Diagnóstico**
 - ✓ Permite identificar las oportunidades de mejora a través de la aplicación de valoraciones rápidas.
 - ✓ Permite realizar valoraciones rápidas por cada ciclo de mejora.

- **Correspondencia**
 - ✓ Permite registrar información sobre la evidencia física que avala las actividades de los procesos realizadas.

- ✓ Permite realizar cambios a la información de correspondencia durante la ejecución de las mejoras.

Cabe destacar que para permitir la compatibilidad con EPFComposer, fue necesario desarrollar una biblioteca que permite la traducción de los archivos XML generados por la herramienta EPFComposer, en un modelo de clases, usando como patrón el modelo UMA definido en la API del EPFComposer, por lo que no se utilizan herramientas extras para la carga de modelos de referencia de procesos.

Es por lo anterior que se alcanzaron las metas propuestas en un principio para esta tesis y para el sistema desarrollado en ella, no solo se completaron con las funcionalidades planeadas, sino que también se revisaron las funcionalidades propuestas con expertos en mejora de procesos de software, lo cual ofrece más veracidad a los requerimientos y permite que la herramienta cuente con mejores fundamentos.

CONTRIBUCIONES

Este sistema fue diseñado originalmente con el objetivo de beneficiar a la industria de software mexicana, buscando permitir a las organizaciones el control de sus procesos con el modelo de referencia de procesos MoProSoft. Sin embargo, al tomar como referencia el modelo de mejora PmCompetiSoft para la implementación de las mejoras, y al trabajar con especialistas en procesos pertenecientes al proyecto CompetiSoft, se pudo incluir en la biblioteca de modelos de referencia a CompetiSoft, por lo que permite aumentar la proyección del beneficio hacia la industria iberoamericana.

GENESIS es una herramienta que busca fomentar el uso de buenas prácticas de Ingeniería de software. Es un aplicación RIA no solo para mejorar la experiencia del usuario, sino también para fomentar el trabajo cooperativo durante la implementación de mejoras, por lo que al ser una herramienta web permite ser utilizada desde distintos lugares al mismo tiempo.

LIMITACIONES

El uso de tecnologías RIA presenta un gran reto, aunque la curva de aprendizaje fue alta, se requirió de un poco más de tiempo para el aprendizaje tanto del lenguaje como de la arquitectura que utilizan este tipo de

herramientas (por ejemplo el modelo vista-controlador). Debido a la falta de tiempo, la herramienta no pudo ser probada en condiciones reales de uso.

Así mismo, la creación de una biblioteca con funciones destinadas a la importación de procesos modelados con EPFComposer requirió más tiempo de lo calculado, al final las funcionalidades del sistema se completaron, pero tomaron más tiempo de lo previsto al inicio del desarrollo del proyecto.

TRABAJO FUTURO

Como trabajo futuro se tiene la implementación de un módulo que permita la carga de evaluaciones realizadas con alguna o algunas herramientas especializadas en evaluaciones de procesos, con esto se podría ofrecer una evaluación de procesos más robusta con resultados más concretos.

También sería útil la implementación de un módulo que permita a GENESIS ser una herramienta multilenguaje, para poder ofrecer su uso a organizaciones de otros países que estén interesados en la mejora de procesos de software de PyME's y/o MyPE's (ó como son nombradas en ingles: Very Small Enterprises (VSE's)).

Bibliografía

1. Batista, J. and A. Figueiredo, *SPI in a very small team: a case with CMM. Software Process: Improvement*. Vol. 5, 2000.
2. Miluk, G., *Results of a field study of CMMI for small settings using rapid applied ethnography*. Proceedings of the first International Research Workshop for Process Improvement in Small Settings 2005, 2006: p. 45-56.
3. Jhonson, D.L. and J.G. Brodman, *Tailoring the CMM for Small Businesses, Small Organizations, and small projects*. Elements of software process assessment and improvement, 1999.
4. Paul, M.C., *Using the software CMM in small organizations*. Proc. joint 16th Pacific Northwest software quality conf and 8th int'l conf. software quality, 1998: p. 350-360.
5. Kelly, G., *Barriers to adoption of the CMMI process model in small settings*. Proceedings of the first International Research Workshop for Process Improvement in Small Settings 2005, 2006: p. 36-40.
6. Mondragón, O., *Addressing infrastructure issues in very small settings*. Proceedings of the first International Research Workshop for Process Improvement in Small Settings 2005, 2006: p. 23-29.
7. *Software Process Engineering MetaModel*. OMG:[Available from: <http://www.omg.org/technology/documents/formal/spem.htm>]
8. Sommerville, I., *Ingeniería del Software*. 2005, Madrid: Pearson Education, S.A.
9. Paulk, M.C., *Capability Maturity Model for Software, Version 1.1 (Technical Report CMU/SEI-93-TR-024)*. Software Engineering Institute, 1993.
10. *CMMI for Systems Engineering/Software Engineering, Version 1.1*. 2002, Software Engineering Institute: Pittsburgh.
11. *ISO/IEC 12207:2002/FDAM 2. Information technology - Software life cycle process*. 2004, International Organization for standardization: Geneva.
12. *ISO/IEC 15504-5:2006(E). Information technology - Process assessment - Part 5: An exemplar Process Assessment Model*. 2006, International Organization for Standardization: Geneva.
13. *ISO 9001:2000. Quality management systems - Requirements*. 2000, International Organization for Standardization: Geneva.

14. Hareton, L. and Y. Terence, *A process framework for small projects*. Software process: improvement and practice, 2001. **6**(2): p. 67-83.
15. Saiedian, H. and N. Carr, *Characterizing a software process maturity model for small organizations*. ACM SIGICE Bulletin, 1997. **23**(1): p. 2-11.
16. Oktaba, H., *Modelo de Procesos para la Industria de Software - MoProSoft - Versión 1.3, Agosto 2005. NMX-059/01*. Organismo nacional de normalización y evaluación de la conformidad - NYCE, 2005.
17. McFeeley, R., *IDEAL: A users Guide for Software Process Improvement, Handbook CMU/SEI-96-HB-001*. Software Engineering Institute, Carnegie Mellon University, 1996.
18. Weber, K., et al., *Brazilian Software Process Reference Model and Assessment Method*. Computer and Information Sciences, 2005: p. 402-411.
19. Amescua, A., et al., *Ramala: A SPI Service Provider for SMEs*. Proceedings of the First International Research Workshop for Process Improvement in Small Settings, 2005: p. 193-201.
20. Calvo-Manzano, J.A., *Experiences in the Application of software process Improvement in SMES*. Software quality Journal, 2002. **10**(3): p. 261-273.
21. Hurtado, J., et al., *Agile SPI: Software Process Agile Improvement, A Colombia Approach to Software Process Improvement in Small Software Organizations, in Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*. Idea Group Inc.: USA. p. In press, 2007.
22. Scott, L., *Practical Software Process Improvement - The Impact Project*. Proceedings of the Australian software Engineering Conference, 2001: p. 182-189.
23. Oktaba, H., et al., *Software Process Improvement: The COMPETISOFT Project*. IEEE Computer, 2007. **10**(40): p. 21-28.
24. ProSoft. Secretaría de Economía: [Available from: <http://www.economia.gob.mx/?P=1128>]
25. *Diario Oficial de la Federación*. Secretaría de Gobernación: [Available from: <http://www.dof.gob.mx/index.php?year=2005&month=08&day=15>]
26. Pino, F., et al., *Modelo para la Implementación de mejoras de procesos en pequeñas organizaciones software*. XII jornadas de ingeniería de software y bases de datos JISBD'2007, 2007.
27. Pino Correa, F.J., J.C. Vidal Rojas, and J.A. Hurtado Alegría, *Guía del consultor para la mejora de procesos software Versión: 0.7*. 2007, COMPETISOFT.
28. Pino, F.J., et al., *Proceso de Mejora PmCOMPETISOFT Ver. 0.6*. 2007, COMPETISOFT.

-
29. Allaire, J., *Macromedia Flash MX—A next-generation rich client*. Macromedia Whitepaper, 2002.
 30. *Macromedia® Flex™: La solución de niveles de presentación para entregar aplicaciones dinámicas de Internet empresariales*. Macromedia Whitepaper, 2004.
 31. *OpenLaszlo, the premier platform for rich internet applications*. Laszlo Systems, Inc.: [Available from: <http://www.openlaszlo.org/>].
 32. *JavaFX Preview SDK Release Notes*. Sun Microsystems: [Available from: <http://java.sun.com/javafx/reference/releasenotes/javafx-sdk-release-notes.html>]
 33. *The Official Microsoft Silverlight Site*. Microsoft: [Available from: <http://silverlight.net/>]
 34. *Adobe: Flex 3*. Adobe Systems Incorporated: [Available from: <http://www.adobe.com/es/products/flex/>]
 35. *Adobe - Adobe AIR*. Adobe Systems Incorporated: [Available from: <http://www.adobe.com/es/products/air/>]
 36. *ISO/IEC 15504-2:2003/Cor.1:2004(E) Information technology - Process assessment - Part 2: Performing an assessment*. 2004, International Organization for standardization: Geneva.
 37. *I.T.Mark*. Europe Software Institute: [Available from: www.esi.es/en/main/iitmark.html].
 38. Fayad, M.E., M. Laitinen, and R.P. Ward, *Software Engineering in the Small*. Communications of the ACM, 2000. **43**(3): p. 115-118.
 39. *Panorama de la industria del software en Latinoamérica*. 2004; Mayer&Bunge: [Available from: www.mbi.com.br/200409_panorama_industria_software_america_latina.pdf]
 40. Pino, F., F. García, and M. Piattini, *Revisión Sistemática de mejora de procesos software en micro, pequeñas y medianas empresas*. Revista Española de innovación, calidad e ingeniería del software (REICIS), 2006. **1**(2): p. 6-23.
 41. *Software process improvement and control*. MKS: [Available from: http://www.mks.com/process_improvement].
 42. *TargetProcess. Agile Project Management Software*. TargetProcess: [Available from: [ww.targetprocess.com](http://www.targetprocess.com)]
 43. Beck, K., *Extreme Programming Explained: Embrace Change*. 2000: Addison-Wesley.
 44. *SIMPLe*. AlturaSoluciones: [Available from: <http://alturasoluciones.com>]

45. Horvat, R.V., I. Rozman, and J. Györkös, *Managing the complexity of SPI in small companies*. Software Process: Improvement and Practices, 2000. 5(1): p. 45-54.
46. *Eclipse Process Framework Project (EPF)*. The Eclipse Foundation: [Available from: <http://www.eclipse.org/epf/>]
47. Hernández Durán, M., *GENESIS, sistema de apoyo a un proceso de mejora: gestión y revisión*. 2009: México, Distrito Federal.
48. Jacobson, I., G. Booch, and J. Rumbaugh, *El proceso unificado de desarrollo de software*, España: Prentice Hall.