

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**HERRAMIENTA PARA GESTION DE PROYECTOS BASADA EN
XPDL PARA EL PROYECTO COMPETISOFT**
Construcción, Pruebas e Integración

Tesis para optar por el Título de Ingeniero Informático, que presenta los bachilleres:

Evelyn Lindsay Ocampo Moreno
Carlos Gonzáles Cajahuanca

ASESOR: Abraham Eliseo Dávila Ramón

2011

RESUMEN

En el ambiente de negocios de hoy, más que nunca las organizaciones dependen del buen resultado de sus proyectos para estar en condiciones de alcanzar una multitud de objetivos; desde objetivos estratégicos hasta las mejoras operacionales diarias.

El mundo en la actualidad está cambiando a velocidades inusitadas y las organizaciones deben reaccionar rápidamente abordando proyectos que las ayuden a alcanzar nuevos objetivos. La gestión de proyectos basada en una metodología ordenada, sistemática y rigurosa facilita la administración de los proyectos que enfrentan cada día las empresas y sus administradores. El adecuado conocimiento y aplicación de estas metodologías para gestión de proyectos permiten crear un ambiente de trabajo propicio y con menor variabilidad para obtener resultados efectivos.

XPDL (XML Process Definition Language) es un lenguaje para la definición de un flujo de trabajo propuesto por la WfMC (Workflow Management Coalition). El objetivo de este lenguaje es proporcionar una manera estándar que permita la importación y exportación de las definiciones de procesos.

El presente trabajo de tesis presenta el desarrollo de una herramienta software basada en el lenguaje XPDL, la cual fue concebida con el propósito de realizar el seguimiento y control de cualquier tipo de proyecto de software, gestionando su avance, plazos, esfuerzos, recursos y ofreciendo la información necesaria sobre cada elemento para su administración oportuna, permite crear la instancia de una metodología a través de una interfaz gráfica, así como apoyar con el manejo de otros elementos críticos en los proyectos informáticos como es la gestión de la configuración.

Cabe resaltar que el presente proyecto es parte del componente de desarrollo de herramientas que viene realizando el Grupo de Investigación y Desarrollo en Ingeniería de Software y Sistemas de Información de la PUCP como parte del Proyecto COMPETISOFT (Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria de Software de Ibero América).

FACULTAD
 DE INGENIERÍA
 Especialidad de Ingeniería Informática

 90
 AÑOS

 PONTIFICIA
 UNIVERSIDAD
 CATÓLICA
 DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

TÍTULO: Herramienta para Gestión de Proyectos basada en XPD L para el proyecto COMPETISOFT – Construcción, Pruebas e Integración

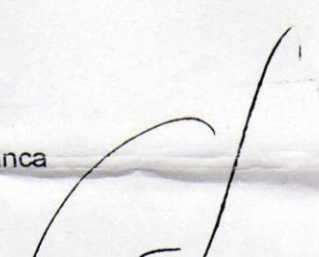
ÁREA: Ingeniería de Software

ASESOR: Ing. Abraham Dávila

ESTUDIANTES	CODIGO	NOMBRES
	1998.7191.3.12	Evelyn Ocampo Moreno
	1999.2077.N.12	Carlos Gonzales Cajahuanca

TEMA N°: 258

FECHA: Lima, 06 de julio de 2007



DESCRIPCIÓN

Las organizaciones realizan sus actividades a través de un conjunto de procesos de diversa naturaleza como procesos administrativos, procesos de fabricación y procesos para la ejecución proyectos, entre otros. Lamentablemente las empresas medianas y pequeñas no suelen ser conscientes de la necesidad de tener claramente identificados sus procesos; en particular las empresas desarrolladoras de software.

De otro lado, la presencia de modelos de calidad de procesos, la reorientación de la norma ISO 9000 hacia procesos, la presión del mercado para que las empresas sean más formales en sus actividades (ejecución de procesos) ha provocado la aparición de alternativas para modelado de procesos adecuados para las organizaciones, como es el caso de XPD L (de XML Process Definition Language) y BPMN (Business Process Modeling Notation). En particular, las empresas que desarrollan software están presionadas por modelos de Madurez y Capacidad como CMMI, MoProSoft y MPS; lo que las lleva a la necesidad de contar con herramientas que les permita modelar sus procesos, pero que además les permita compararse con respecto a ciertos modelos de referencia.

El presente proyecto de tesis busca implementar una solución que permita la gestión de proyectos de software, crear la instancia de una metodología y administración de un proyecto específico. El modelo se basará en la definición de procesos usando XPD L internamente y BPMN como notación gráfica.

OBJETIVO

Elaborar una herramienta que permita administrar proyectos de software y crear la instancia de una metodología a través de una interfaz grafica y basada en el lenguaje XPD L.




FACULTAD
 DE INGENIERÍA
 Especialidad de Ingeniería Informática



90
 AÑOS

PONTIFICIA
 UNIVERSIDAD
 CATÓLICA
 DEL PERÚ

ALCANCES

La herramienta permitirá la gestión de proyectos de software, crear la instancia de una metodología y administración de un proyecto específico usando XML como repositorio para de los datos de los proyectos. Las entidades que pueden ser asociadas a una metodología son: los proyectos, las actividades y los artefactos. Se podrá administrar gráficamente las actividades y artefactos usando un esquema sencillo.

La herramienta proveerá un mecanismo de gestión del repositorio de artefactos (gestión de configuración) para cada proyecto, esta permitirá a los usuarios almacenar y modificar con seguridad los artefactos que se definan por cada actividad. La información ayudará a los administradores a controlar el avance de los proyectos siguiendo la metodología elegida.

En este documento se presenta el trabajo de Evelyn Ocampo y Carlos Gonzales en la parte de construcción, pruebas e integración que se complementa al trabajo de Sara Villegas y Anita Silva.

COMPONENTES	RESPONSABLES
Creación de instancia de la metodología	Carlos Gonzales
Administración de proyectos	Evelyn Ocampo
Administración de proyecto específico	Evelyn Ocampo
Administración de la configuración	Carlos Gonzales

Este proyecto se integra a otros proyectos que se vienen desarrollando como parte del Grupo de Investigación y Desarrollo en Ingeniería de Software y del proyecto COMPETISOFT.

Máximo: 100 páginas





DEDICATORIA

*A Dios, que me dio la fuerza y salud para todo lo emprendido.
A mis padres, Ana y José quienes me enseñaron desde siempre a luchar para alcanzar mis metas, por el apoyo incondicional y los sabios consejos, ¡los amo!
A mis hermanos que me acompañaron en los momentos difíciles, los quiero mucho.
A mis amigos, quienes se convirtieron en parte de mi familia.
A Sara, Anita y Carlos por ayudarme en cada momento y hacerme sentir bien cuando lo necesité.*

Evelyn Ocampo Moreno

*A Dios, que me dio la fuerza y salud para todo lo emprendido.
A mi madre Doris quien me enseñó desde siempre a luchar para alcanzar mis metas, por su apoyo incondicional y sus sabios consejos.
A Evelyn, Anita y Sara por ayudarme en cada momento.*

Carlos Gonzáles Cajahuanca

AGRADECIMIENTOS

A la Pontificia Universidad Católica del Perú y en especial a la Facultad de Ciencias e Ingeniería que nos dieron la oportunidad de formar parte de ellas.

Un agradecimiento muy especial al Ing. Abraham Dávila, nuestro asesor, por su amistad, apoyo, dedicación y orientación para el desarrollo del presente trabajo de tesis y llegar a la culminación del mismo.

A toda las personas que de una forma o de otra nos brindaron su apoyo para la realización del presente trabajo.

RECONOCIMIENTOS

El presente trabajo está enmarcado dentro del proyecto 506AC0287 COMPETISOFT (Mejora de Procesos Para Fomentar la Competitividad de la Pequeña y Mediana Industria de Software de Ibero América) del programa CYTED (Ciencia y Tecnología para el Desarrollo), parcialmente financiado por PUCP- VRI - 2009-0008 PYMESOFT: Incremento de la productividad de pymes desarrolladoras de software: determinación de patrones de problemas y soluciones en un contexto de mejora de procesos (COMPETISOFT Perú 2da Fase) y con el apoyo del Departamento de Ingeniería de la Pontificia Universidad Católica del Perú.

Índice General

Introducción	1
Capítulo 1: Generalidades	3
1.1. Gestión de Proyectos.	3
1.1.1. Definición de proyectos	3
1.1.2. Definición de Gestión de proyectos.	5
1.1.3. Problemática de los proyectos.	6
1.1.4. Procesos según PMI.	9
1.2. Proyectos Informáticos	11
1.2.1. Procesos Principales.	11
1.2.2. Procesos de Apoyo.	13
1.2.3. Procesos Organizativos.	13
1.3. Gestión de la Configuración del Software (GCS)	14
1.3.1. Definición de la Gestión de la Configuración del Software.	14
1.3.2. Conceptos en Gestión de la Configuración del Software.	17
1.4. RUP	18
1.5. Lenguaje XPDL.	21
1.6. Proyecto COMPETISOFT	23
1.6.1. COMPETISOFT- CYTED.	24
1.6.1.1. Modelo de procesos (MoProSoft)	24
1.6.2. COMPETISOFT- PUCP.	28
1.6.3. COMPETISOFT- PUCP Tools.	29
1.7. Revisión de Productos.	34
1.7.1. Tecnológico	34
1.7.2. Descripción y Sustentación de la Solución	40
1.7.3. Características principales de la herramienta propuesta	41
Capítulo 2: Construcción	44
2.1. Consideración de Implementación.	44
2.1.1. Lenguajes de programación	45
2.1.2. Tecnologías usadas en la herramienta	46
2.1.2.1. Java 2 Software Development Kit (J2SDK)	46
2.1.2.2. JSP (Java Servlet Pages)	47
2.1.2.3. Spring	48
2.1.2.4. Librería JavaScript	51
2.1.2.5. Librería para diagrama de Gantt	51
2.1.3. Marco de trabajo de Presentación.	54
2.1.4. Marco de trabajo de Lógica del negocio.	57
2.1.5. Entorno de desarrollo integrado (IDE)	58
2.1.6. Procesamiento de datos XML	60
2.1.7. Servidor de Aplicación y Web	67
2.2. Mapeo de datos XML a clases.	69
2.3. Implementación de Componentes	76
2.3.1. Estructuración Interna del Proyecto Web	76
2.3.2. Configuración de Spring	84
2.3.3. Configuración de Java Server Faces	86
2.4. Administración de datos.	89
2.4.1 ServiceLocatorBean	89
2.4.2 Implementación de Servicios	90
2.4.3 MotorXML	90
2.5. Creación de Instancia de Metodología.	91
2.5.1 Artefactos de la Metodología	91
2.5.2 Actividades de la Metodología	92
2.5.3 Gestor de Actividades de la Metodología	92
2.5.4 Gestor de Artefactos de la Metodología	93
2.5.5 Metodología	93
2.5.6 Paquete de Metodología	94
2.5.7 Configuración de la Metodología	95

2.5.8	Gestor de la Configuración de la Metodología	95
2.6	Administración de Proyectos.	96
2.6.1	Proyecto	96
2.6.2	Gestor de Proyectos	98
2.7	Administración de Proyecto Específico	98
2.7.1	Actividades del Proyecto	98
2.7.2	Gestor de Actividades	100
2.7.3	Horas trabajadas de la Actividad	100
2.7.4	Usuarios de la Actividad del Proyecto	101
2.8	Gestión de la Configuración	102
2.8.1	Versión de Artefactos	102
2.8.2	Artefactos	103
2.8.3	Gestor de Artefactos	105
2.9	Diagrama de Gantt	106
2.9.1	Configuración de scripts	106
2.9.2	Generación de datos	106
2.9.3	Implementación del diagrama	108
Capítulo 3: Pruebas		110
3.1.	Prueba caso de uso: Mantener Proyecto.	110
3.2.	Prueba caso de uso: Mantener Actividades.	111
3.3.	Prueba caso de uso: Mantener Artefactos.	112
3.4.	Prueba caso de uso: Reservar Artefactos.	112
3.5.	Prueba caso de uso: Crear Versión Artefacto.	113
3.6.	Prueba caso de uso: Mantener Usuario.	113
3.7.	Prueba caso de uso: Mantener Empresas.	114
3.8.	Prueba caso de uso: Ingresar Horas Trabajadas en las Actividades.	114
3.9.	Pruebas integrales del sistema	115
3.10.1.	Registrar proyecto completo	115
3.10.2.	Reservar Artefactos, cambiar y subir versión	122
3.10.	Flujo de Eventos	127
3.11.1.	Proceso de Administración de proyectos.	127
3.11.2.	Proceso de Administración de proyectos específicos.	141
3.11.3.	Gestión de la Configuración	148
Capítulo 4: Observaciones, conclusiones y recomendaciones		152
4.1.	Observaciones	152
3.2.	Conclusiones	153
3.3.	Recomendaciones	154
Bibliografía		155

ANEXOS

- A. Diagrama de Casos de Uso.
- B. Diagrama de Clases Análisis.
- C. Diccionario de Datos Clases Análisis.
- D. Diagrama de Clases Diseño.
- E: Diagrama de Secuencia.
- F. Plan de Pruebas del Sistema EACS Project Manager.
- G. Pruebas integrales del Sistema EACS Project Manager.

Índice de Figuras

Figura 1.1. Estadísticas de proyectos de software [Standish Group, 2009].	8
Figura 1.2. Grupos de Procesos en la Gestión de Proyectos según PMI [Adaptado por EACS].	11
Figura 1.3. Estructura de la norma técnica peruana [Francisco Ruiz, 2007].	12
Figura 1.4. Grafo de evolución de versiones [SWEBOK, 2004].	18
Figura 1.5. Los flujos de trabajo en cada una de las fases e iteraciones del RUP [JACOBSON, 2000].	20
Figura 1.6. Modelo de Referencia de Procesos [MOPROSOFT].	27
Figura 1.7. Relación entre las 3 herramientas del proyecto COMPETISOFT PUCP	31
Figura 2.1. Módulos de Spring Framework [SPRING]	49
Figura 2.2. Funcionamiento de SAX [SAX]	61
Figura 2.3. Funcionamiento de DOM [DOM]	62
Figura 2.4. Generación de clases mediante data binding [DBINDING].	64
Figura 2.5. Estructura de archivos del proyecto	77
Figura 2.6. Carpeta Páginas Web.	78
Figura 2.7. Estructura de paquetes.	81
Figura 2.8. Librerías utilizadas en el proyecto.	83
Figura 2.9. Diagrama de Gantt – Formato en pantalla	109
Figura 3.1. Pantalla Ingresar al Sistema.	128
Figura 3.2. Pantalla Inicial.	128
Figura 3.3. Pantalla Mantener Proyectos.	129
Figura 3.4. Pantalla Nuevo Proyecto.	129
Figura 3.5. Pantalla grabar Nuevo Proyecto.	130
Figura 3.6. Pantalla Buscar Proyectos.	130
Figura 3.7. Pantalla Editar Proyecto.	131
Figura 3.8. Pantalla Mis Actividades.	131
Figura 3.9. Pantalla Buscar Mis Actividades.	132
Figura 3.10. Pantalla Ingreso de Horas trabajadas.	132
Figura 3.11. Pantalla Mantener Usuarios.	133
Figura 3.12. Pantalla Nuevo Usuario.	133
Figura 3.13. Pantalla grabar Nuevo Usuario.	134
Figura 3.14. Pantalla Mantenimiento Usuarios.	134
Figura 3.15. Pantalla Editar Usuario.	135
Figura 3.16. Pantalla Mantener Empresas.	135
Figura 3.17. Pantalla Nueva Empresa.	136
Figura 3.18. Pantalla grabar Nueva Empresa.	136
Figura 3.19. Pantalla Buscar Empresas.	137
Figura 3.20. Pantalla Edición Empresas.	137
Figura 3.21. Pantalla Listado Reportes.	138
Figura 3.22. Pantalla elegir Reporte Detallado.	138
Figura 3.23. Pantalla Reporte Detallado del Proyecto.	139
Figura 3.24. Pantalla elegir Reporte de Esfuerzo.	140
Figura 3.25. Pantalla Reporte de esfuerzo del Proyecto.	140
Figura 3.26. Pantalla Mantener Personal del Proyecto.	141
Figura 3.27. Pantalla Agregar Personal al Proyecto.	142
Figura 3.28. Pantalla Editar Personal del Proyecto.	142
Figura 3.29. Pantalla Mantener Actividades del Proyecto.	143
Figura 3.30. Pantalla Nueva Actividad del Proyecto.	143
Figura 3.31. Pantalla Editar Actividad del Proyecto.	144
Figura 3.32. Pantalla Mantener Artefactos del Proyecto.	144
Figura 3.33. Pantalla Nuevo Artefacto del Proyecto.	145
Figura 3.34. Pantalla elegir ver Versiones del Artefacto del Proyecto.	145
Figura 3.35. Pantalla Listado de Versiones del Artefacto del Proyecto.	146
Figura 3.36. Pantalla Descarga de Versiones Artefacto del Proyecto.	146
Figura 3.37. Pantalla seleccionar ver Diagrama de Gantt del Proyecto.	147
Figura 3.38. Pantalla Diagrama de Gantt del Proyecto.	147

Figura 3.39. Pantalla Listado Artefactos del Proyecto. 148

Figura 3.40. Pantalla Reserva Artefactos del Proyecto. 149

Figura 3.41. Pantalla Artefacto reservado del Proyecto. 149

Figura 3.42. Pantalla elegir Subir Artefacto del Proyecto. 150

Figura 3.43. Pantalla Subir Artefacto del Proyecto. 150

Figura 3.44. Pantalla Artefacto liberado del Proyecto. 151



Índice de Código

Código 1.1. Estructura definición de procesos [MJS].	32
Código 1.2. Estructura de las actividades definidas en una metodología [MJS].	33
Código 2.1. Archivo XML de la herramienta EACS.	71
Código 2.2. Elemento raíz del documento.	71
Código 2.3. Elementos que contiene la raíz.	72
Código 2.4. Final del elemento raíz.	72
Código 2.5. Clase BGenTipos.	74
Código 2.6. Archivo de Mapeo para la clase BGenTipos.	75
Código 2.7. Archivo XML obtenido del mapeo de la clase BGenTipos.	75
Código 2.8. Código de una página Java Server Faces.	79
Código 2.9. Código de una clase controlador Java Server Faces.	82
Código 2.10. Configuración SPRING en web.xml.	84
Código 2.11. Configuración SPRING en faces-config.xml.	85
Código 2.12. Configuración SPRING en applicationContext.xml.	86
Código 2.13. Configuración JSF en web.xml.	87
Código 2.14. Configuración controladores JSF en faces-config.xml.	88
Código 2.15. Configuración navegación JSF en faces-config.xml.	89
Código 2.16. ServiceLocatorBean constructor.	89
Código 2.17. Implementación de servicios constructor.	90
Código 2.18. MotorXML método escribir.	91
Código 2.19. MotorXML método leer.	91
Código 2.20. Esquema de Datos – Artefactos de la Metodología.	92
Código 2.21. Esquema de Datos – Actividades de la Metodología.	92
Código 2.22. Esquema de Datos – Gestor de Actividades de la Metodología.	93
Código 2.23. Esquema de Datos – Gestor de Artefactos de la Metodología.	93
Código 2.24. Esquema de Datos – Metodología.	94
Código 2.25. Esquema de Datos – Paquete de Metodología.	94
Código 2.26. Esquema de Datos – Paquete de Metodología.	95
Código 2.27. Esquema de Datos – Gestor de la Configuración de la Metodología.	95
Código 2.28. Esquema de Datos – Proyecto.	98
Código 2.29. Esquema de Datos – Gestor de Proyectos.	98
Código 2.30. Esquema de Datos – Actividades del Proyecto.	100
Código 2.31. Esquema de Datos – Gestor Actividades.	100
Código 2.32. Esquema de Datos – Horas trabajadas de la Actividad.	101
Código 2.33. Esquema de Datos – Usuarios de la Actividad del Proyecto.	102
Código 2.34. Esquema de Datos – Versión de Artefactos.	103
Código 2.35. Esquema de Datos – Artefacto.	105
Código 2.36. Esquema de Datos – Gestor de Artefactos.	105
Código 2.37. Diagrama de Gantt – Configuración de scripts.	106
Código 2.38. Diagrama de Gantt – Generación de datos.	107
Código 2.39. Diagrama de Gantt – Componente DIV.	108
Código 2.40. Diagrama de Gantt – Código javascript para mostrar el diagrama.	108

Índice de Tablas

Tabla 1.1. Comparación de MoProSoft con otros modelos [MOPROSOFT, 2003].....	24
Tabla 1.2. Características de las herramientas existentes en el mercado.	35
Tabla 2.1. Comparación de enfoques de procesamiento de datos XML en aplicaciones Java	63



Introducción

Un proyecto se define según el Instituto de Manejo de Proyectos (PMI de sus siglas en inglés de Project Management Institute), como un esfuerzo temporal y único que se realiza para lograr un objetivo, este objetivo puede ser un producto, servicio o resultado. En general se puede decir que la realización de uno o más proyectos puede cambiar el status-quo de una organización y que un producto puede ser el esfuerzo de una sola persona o bien de varios miles de individuos y durar unos pocos días o tomar varios años; los proyectos pueden involucrar a una sola unidad de una organización o bien pueden traspasar las fronteras organizacionales. Por tanto, en proyectos no triviales una mala planificación o ejecución puede ocasionar grandes pérdidas para las organizaciones involucradas.

Las empresas relacionadas a las tecnologías de información (TI) realizan principalmente sus actividades basadas en proyectos informáticos de diversos tipos, que van desde proyectos de construcción de software a proyectos de implantación de grandes productos comerciales; pasando por una gama variada de posibilidades. Las empresas de TI, ejecutan los proyectos sea para clientes internos en su propia organización o clientes externos.

La realidad del sector de TI, en cuanto a la gestión de proyectos, es deficiente pero puede ser mejorado con la adopción de buenas prácticas y el soporte de herramientas adecuadas.

La realidad del sector de TI también se puede mejorar con la adopción de modelos de procesos que han sido previamente probados y que ayude a la realización de la mejora continua. Este último hecho es una tendencia hoy en el mundo informático con modelos como CMMI, ITIL, MoProSoft, ISO9000, etc.

En el mercado existen diversas soluciones para el manejo de proyectos, sin embargo la mayoría están focalizadas en estos aspectos específicos de la gestión y no integradas a otros productos para el análisis y mejora de los procesos involucrados en los proyectos informáticos.

El proyecto EACS Project Manager es una iniciativa para la construcción de una herramienta que soporte en principio las tareas usuales de la gestión de un proyecto informático e integre la gestión de la Configuración para facilitar el trabajo de los equipos de desarrollo.

El presente trabajo está enmarcado dentro del proyecto COMPETISOFT (mejora de procesos para fomentar la competitividad de la pequeña y mediana industria de software de Iberoamérica) del programa CYTED (Ciencia y Tecnología para el Desarrollo) y apoyado parcialmente por la Dirección de Gestión de Investigación (2009-0008 PYMESOFT) y el Departamento de Ingeniería de la Pontificia Universidad Católica del Perú.

Este documento presenta el análisis y diseño de la herramienta EACS Project Manager, para Gestión de Proyectos basado en XPDL, dentro de la plataforma PS-PUCP y que se complementa con la tesis de Sara Villegas Ortega y Anita Silva Lazo que presentan el análisis y diseño.

Capítulo 1: Generalidades

En este capítulo se presenta los aspectos básicos necesarios para entender el desarrollo del proyecto EACS Project Manager. Se cubren los temas de Gestión de Proyectos, Proyectos Informáticos, el Lenguaje XPDL, el proyecto COMPETISOFT-PUCP y la revisión de proyectos existentes.

Este capítulo se ha desarrollado de manera conjunta con Anita Silva Lazo y Sara Villegas Ortega, quienes realizaron la tesis complementaria, el cual se presenta en ambos documentos para comprender el contexto en el que se desarrolla el Proyecto.

1.1. Gestión de Proyectos.

En esta sección se presentan aspectos relacionados a la gestión de proyectos, definición de proyectos, problemática de los proyectos y procesos según el PMI.

1.1.1. Definición de proyectos

Existen diversas definiciones de proyectos, a continuación citaremos algunas de ellas:

- Proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio resultado único. Completar con éxito un proyecto significa cumplir con los objetivos dentro de las especificaciones técnicas, los costos y los plazos establecidos. Un proyecto tiene un propósito único, algo que no ha sido realizado de la misma manera anteriormente y es, por lo tanto, único y distinto [PMBOK, 2008].
- Proyecto es un trabajo no repetitivo, que ha de planificarse y realizarse según unas especificaciones técnicas determinadas, y con objetivos de costos, inversiones y plazos prefijados. También se define un proyecto como un trabajo de volumen y complejidad considerables, que ha de realizarse con la participación de varios departamentos de la empresa y tal vez también con la colaboración de terceros [BOVERI, 1990].
- Proyecto es una operación de envergadura y complejidad notables, de carácter no repetitivo, que se acomete para realizar una obra de importancia [PEREÑA, 1996].

Los proyectos tienen las siguientes características:

- Son temporales, definen un comienzo y un final, es decir una duración finita. Temporal no necesariamente significa de corta duración; muchos proyectos duran varios años. En cada caso, sin embargo, la duración de un proyecto es limitada. Los proyectos no son esfuerzos continuos. Además lo temporal no es aplicable generalmente al producto, servicio o resultado creado por el proyecto [PMBOK, 2008].
- Crean productos entregables únicos. Productos entregables son productos, servicios o resultados [PMBOK, 2008].
- Se elaboran gradualmente, ésta es una característica de los proyectos que acompaña a los conceptos de temporal y único. La elaboración gradual significa desarrollar en pasos e ir aumentando mediante incrementos [PMBOK, 2008].
- Tienen como finalidad alcanzar su objetivo y luego concluirlo. El proyecto concluye cuando se alcanzan sus objetivos específicos o éste se cancela [PMBOK, 2008].
- Son importantes y que suponen un esfuerzo notable para la entidad que lo acomete porque requiere inversiones cuantiosas [PEREÑA, 1996].
- Pueden involucrar una sola persona o varios miles.
- Pueden involucrar a distintas áreas de una organización.

- Se planifican, ejecutan y controlan.
- Cuentan con un patrocinador, quien es la persona o grupo que proporciona los recursos financieros, en efectivo o en especie, para el proyecto [PMBOK, 2008].
- Todo proyecto está destinado a finalizar en un plazo predeterminado, consistiendo dicha finalización en la entrega de un producto [PEREÑA, 1996].
- El proyecto está en continua evolución y se caracteriza por un notable dinamismo derivado de su carácter de operación inusual tendente a crear algo nuevo [PEREÑA, 1996].
- Algunos proyectos suponen un fuerte riesgo, económico o de otra naturaleza, estando sometidos a contingencias difícilmente dominables e incluso azarosas [PEREÑA, 1996].

1.1.2. Definición de Gestión de proyectos.

Existen diversas definiciones de gestión de proyectos, a continuación citaremos algunas de ellas:

- La Gestión de Proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuadas de los 42 procesos de la dirección de proyectos, agrupados lógicamente, que conforman los 5 grupos de procesos.

Los grupos de procesos de la gestión de proyectos son: inicio, planificación, ejecución, seguimiento y control, y cierre. Dirigir un proyecto por lo general implica:

- Identificar requisitos.
- Abordar las diversas necesidades, inquietudes y expectativas de los interesados según se planifica y efectúa el proyecto.
- Equilibrar las restricciones contrapuestas del proyecto que se relacionan, entre otros aspectos, con: el alcance, la calidad, el cronograma, el presupuesto, los recursos y el riesgo.

El proyecto específico influirá sobre las restricciones en las que el director del proyecto necesita concentrarse.

La relación entre estos factores es tal que si alguno de ellos cambia, es probable que al menos otro se vea afectado. Por ejemplo, un adelanto en el

cronograma a menudo implica aumentar el presupuesto, a fin de añadir recursos adicionales para completar la misma cantidad de trabajo en menos tiempo. Si no es posible aumentar el presupuesto, se puede reducir el alcance o la calidad, para entregar un producto en menos tiempo por el mismo presupuesto. Los interesados en el proyecto pueden tener opiniones diferentes sobre cuáles son los factores más importantes, lo que crea un desafío aún mayor. Cambiar los requisitos del proyecto puede generar riesgos adicionales. El equipo del proyecto debe ser capaz de evaluar la situación y equilibrar las demandas a fin de entregar un proyecto exitoso [PMBOK, 2008].

- La Gestión de Proyectos es un conjunto de métodos y técnicas de gestión que, inspirados por el sentido común y el rigor profesional, están encaminadas a mejorar, definir, planificar, impulsar y controlar las operaciones. La gestión de proyectos representa un todo completo y coherente. No se trata de tomar una terminología vistosa ni de aplicar esporádicamente técnicas puntuales o instrumentos parciales. Si se quiere obtener un efecto perceptible y duradero en la calidad de la gestión, hay que adoptar una metodología en su conjunto, prestando especial atención a los aspectos culturales y de fondo pero sin descuidar los de naturaleza operativa e instrumental.

Un pilar de la gestión profesional del proyecto es el empleo de técnicas de gestión conocidas a un terreno “sui generis”, pero variando sensiblemente la forma de aplicar dichas técnicas y poniendo el énfasis en ciertos puntos que son especialmente sensibles en las operaciones discontinuas.

La correcta Gestión de los Proyectos será una inversión de máxima rentabilidad al evitar caer en todo ese conjunto de causas de fracaso o ineficacia. Como es lógico, no se puede considerar la metodología de gestión como un fin en sí misma, sino como una ayuda encaminada a facilitar la consecución de los resultados [PEREÑA, 1996].

1.1.3. Problemática de los proyectos.

En la actualidad un alto porcentaje de proyectos en tecnologías de información y comunicación tienen fracasos financieros, debido a la mala dirección de los mismos o a que los actores perdieron el horizonte al momento de poner en práctica sus planes de acción.

Información reciente sobre la gerencia de proyectos de TI muestra que [STANDISH GROUP, 2009]:

- El mundo gasta casi \$10 trillones en proyectos de toda clase.
- Más de 16 millones de personas están involucradas en proyectos en el mundo.
- Tom Peters considerado por muchos el padre de la administración moderna y de la innovación aplicada en su obra 50 claves para la dirección de proyectos dice: “Para ganar hoy debemos dominar el arte del proyecto”.

Un estudio realizado por Standish Group analizó el desarrollo de 8000 proyectos de software, realizados por 350 empresas diferentes y concluyó que solamente el 32% de los proyectos fueron exitosos y el 24% fueron cancelados antes de su terminación, costando billones de dólares.

El estudio identificó como principales causas de los problemas:

- Especificaciones y requerimientos cambiantes o incompletos.
- Deficiencias en la aplicación de procesos y desconocimiento del ciclo de vida del proyecto.
- Falta de involucramiento de usuarios.
- Falta de recursos.
- Expectativas no realistas.
- Falta de soporte ejecutivo.
- Falta de planificación.
- Falta de gestión de las tecnologías de la información.
- Desconocimiento tecnológico.

Los criterios para determinar el éxito de un proyecto son:

- Sin desviaciones en las fechas previstas.
- Sin desviaciones en los costos estimados.
- Que el producto final cubra las expectativas y necesidades del cliente.
- Que funcione correctamente.

Los factores que afectan el éxito de los proyectos según Baker, Murphy y Fisher, citados por McManus [MCMANUS, 2003], quienes estudiaron 650 proyectos en los Estados Unidos son los siguientes:

- Compromiso con el proyecto en el establecimiento de cronogramas, presupuestos y objetivo de desempeño técnicos.

- Frecuente retroalimentación de la organización patrocinadora.
- Compromiso del cliente y del patrocinador, comprometido en el establecimiento de cronogramas, presupuestos y objetivos de desempeño técnicos.
- Estructura de la organización adecuada al equipo del proyecto.
- Participación del equipo del proyecto en la determinación del cronograma y los presupuestos.
- Entusiasmo del patrocinador.
- Deseo del patrocinador de crear las capacidades internas.
- Cambios de personal durante el proyecto.
- Falta de una adecuada identificación de riesgos.
- Falta de seguimiento periódico del proyecto, control de la planificación y revisiones para corregir las desviaciones.

La Figura 1.1 muestra las estadísticas de proyectos de software según estudios realizados por Standish Group.

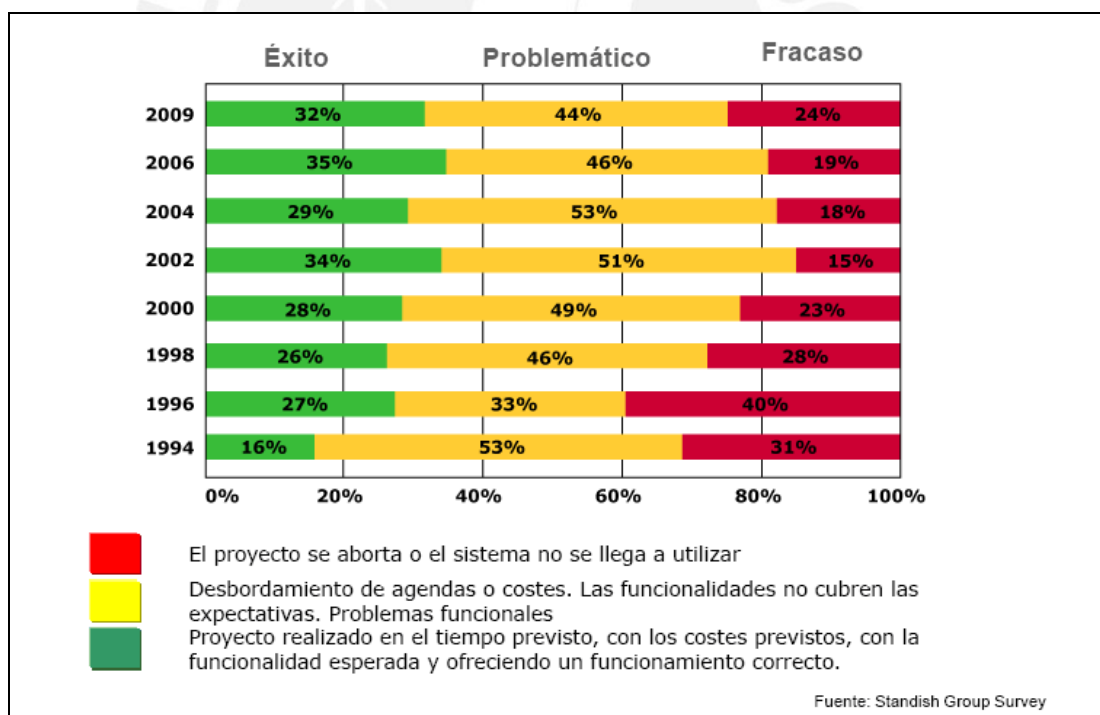


Figura 1.1. Estadísticas de proyectos de software [Standish Group, 2009].

Algunas de las deficiencias frecuentes en gestión de proyectos son: [PEREÑA, 1996]

- Ausencia total de planificación, lo que hace que las diversas tareas se vayan acometiendo desordenadamente y a medida que se presentan dificultades.

- Pese a que cada responsable actúa con celeridad cuando se le encarga algo, el proyecto acumula retrasos por falta de planificación y por la dificultad existente de tomar decisiones.
- Las decisiones se toman en órganos colectivos, faltando una cabeza que de unidad e impulse el desarrollo del proyecto.
- Los plazos son enormemente dilatados.
- Las deficiencias de gestión no solo desembocan en grandes problemas de plazo sino en defectos de calidad que obligan a cancelar el proyecto.

1.1.4. Procesos según PMI.

El Project Management Institute (PMI) es una asociación sin fines de lucro, líder en la Industria de la Gerencia de Proyectos, dedicada al progreso y fomento de su aplicación efectiva a través de la práctica. Fundada en 1969 en Pensilvania, Estados Unidos de Norteamérica, actualmente está presente en 172 países, con más de 420,000 miembros y profesionales certificados, organizados en 250 Capítulos [PMI].

Entre sus principales objetivos se encuentran formular estándares profesionales, generar conocimiento a través de la investigación, y promover la Gestión de Proyectos como profesión a través de sus programas de certificación.

La Guía del PMBOK es un estándar en la gestión de proyectos desarrollado por el Project Management Institute (PMI). PMI considera la norma como una referencia fundamental en el ámbito de la dirección de proyectos para sus certificaciones y programas de desarrollo profesional.

El PMBOK es una colección de procesos y áreas de conocimiento generalmente aceptadas como las mejores prácticas dentro de la gestión de proyectos.

El PMBOK es un estándar reconocido internacionalmente que provee los fundamentos de la gestión de proyectos que son aplicables a un amplio rango de proyectos, incluyendo construcción, software, ingeniería, etc.

El PMBOK reconoce 5 grupos de procesos básicos y 9 áreas de conocimiento comunes a casi todos los proyectos. Los conceptos básicos son aplicables a proyectos, programas y operaciones.

Los cinco grupos de procesos básicos son: [PMBOK, 2008]

- **Iniciación:** aquellos procesos realizados para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la obtención de la autorización para comenzar dicho proyecto o fase.
- **Planificación:** aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto.
- **Ejecución:** aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones del mismo.
- **Seguimiento y Control:** aquellos procesos requeridos para dar seguimiento, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.
- **Cierre:** aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, a fin de cerrar formalmente el proyecto o una fase del mismo.

En la Figura 1.2 se muestran los cinco grupos de procesos en la gestión de proyectos según PMI.

Los procesos se traslapan e interactúan a través de un proyecto o fase. Los procesos son descritos en términos de: Entradas (documentos, planes, diseños, etc.), Herramientas y Técnicas (mecanismos aplicados a las entradas) y Salidas (documentos, productos, etc.).

Las nueve áreas del conocimiento mencionadas en el PMBOK son:

- Gestión de la Integración del Proyecto.
- Gestión del Alcance del Proyecto.
- Gestión del Tiempo del Proyecto.
- Gestión de los Costos del Proyecto.
- Gestión de la Calidad del Proyecto.
- Gestión de los Recursos Humanos del Proyecto.
- Gestión de las Comunicaciones del Proyectos.
- Gestión de los Riesgos del Proyecto.
- Gestión de Adquisiciones del Proyecto.

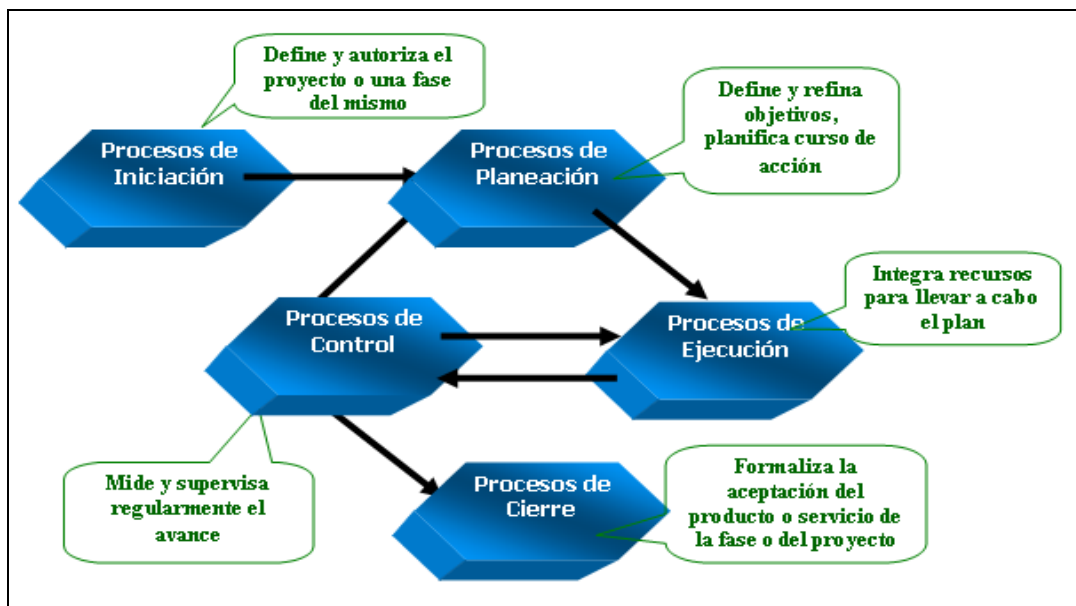


Figura 1.2. Grupos de Procesos en la Gestión de Proyectos según PMI [Adaptado por EACS].

1.2. Proyectos Informáticos

Los proyectos informáticos de acuerdo al objetivo que se sigue se pueden clasificar principalmente en proyectos de desarrollo o proyectos de implantación. En particular en los proyectos de desarrollo de software se ejecutan diversos procesos, la Norma Técnica Peruana 12207 agrupa las actividades en procesos principales, procesos de apoyo y procesos organizativos [NTP-ISO/IEC 12207:2006] y los agrupa como se muestra en la Figura 1.3.

1.2.1. Procesos Principales.

Son los procesos para iniciar o llevar a cabo el desarrollo, operación o mantenimiento del software.

Los procesos principales son: [NTP-ISO/IEC 12207:2006]

- Proceso de adquisición.
- Proceso de suministro.
- Proceso de desarrollo.
- Proceso de operación.
- Proceso de mantenimiento.

Considerando estos grupos, se desarrolla el proceso de desarrollo, por ser de mayor interés para ésta tesis.

El proceso de desarrollo contiene las actividades para el análisis de requisitos, diseño, codificación, integración, pruebas, e instalación y aceptación relativas al software. El desarrollador selecciona y realiza, o presta apoyo, a las siguientes actividades de acuerdo con un contrato.



Figura 1.3. Estructura de la norma técnica peruana [Francisco Ruiz, 2007].

Este proceso consta de las siguientes actividades:

- Implementación del proceso.
- Análisis de los requerimientos del sistema.
- Diseño de la arquitectura del sistema.
- Análisis de los requerimientos software.
- Diseño de la arquitectura del software.
- Diseño detallado del software.

- Codificación y pruebas del software.
- Integración del software.
- Pruebas de calificación del software.
- Instalación del sistema.
- Pruebas de calificación del sistema.
- Instalación del software.
- Soporte a la aceptación del software.

1.2.2. Procesos de Apoyo.

Un proceso de apoyo es el que apoya a otro proceso como parte esencial del mismo, con un propósito bien definido y contribuye al éxito y calidad del proyecto de software. Un proceso de apoyo se emplea y ejecuta por otro proceso, según sus necesidades [NTP-ISO/IEC 12207:2006].

Los procesos de apoyo son: [NTP-ISO/IEC 12207:2006]

- Proceso de documentación.
- Proceso de gestión de la configuración.
- Proceso de aseguramiento de la calidad.
- Proceso de verificación.
- Proceso de validación.
- Proceso de revisión conjunta.
- Proceso de auditoría.
- Proceso de solución de problemas.

1.2.3. Procesos Organizativos.

Se emplean por una organización para establecer e implementar una infraestructura constituida por procesos y personal asociado al ciclo de vida y para mejorar continuamente esta infraestructura. Se usan habitualmente fuera del ámbito de proyectos y contratos, contribuye a la mejora de la organización [NTP-ISO/IEC 12207:2006].

Los procesos organizativos son: [NTP-ISO/IEC 12207:2006]

- Proceso de gestión.
- Proceso de infraestructura.
- Proceso de mejora de proceso.
- Proceso de recursos humanos.

- Proceso de gestión de activos.
- Proceso de gestión del programa de reutilización.
- Proceso de ingeniería del dominio.

Considerando estos grupos, se desarrolla el proceso de gestión, por ser de mayor interés para ésta tesis.

El proceso de gestión contiene las actividades genéricas y tareas que pueden ser empleadas por cualquier parte que tenga que gestionar sus respectivos procesos. El gerente es responsable de la gestión del producto, gestión del proyecto y gestión de las tareas de los procesos aplicables, tales como el de adquisición, suministro, desarrollo, operación, mantenimiento o soporte.

Este proceso consta de las siguientes actividades:

- Inicio y definición del alcance.
- Planificación
- Ejecución y evaluación.
- Finalización.

1.3. Gestión de la Configuración del Software (GCS)

A continuación se revisan diversas definiciones de GCS, sus aspectos funcionales, los beneficios de su aplicación y se definen los conceptos básicos de esta disciplina.

1.3.1. Definición de la Gestión de la Configuración del Software.

Existen diversas definiciones de Gestión de Configuración del Software, citaremos algunas de las más importantes:

- El proceso de Gestión de Configuración es aplicar procedimientos técnicos y administrativos a lo largo del ciclo de vida del software para: identificar, definir y establecer la línea base de los elementos software en un sistema: controlar modificaciones y emisiones de los elementos: registrar e informar del estado de los elementos y peticiones de modificación: asegurar la completitud,

consistencia y corrección de los elementos: y controlar el almacenamiento, manipulación y entrega de los elementos [NTP-ISO/IEC 12207:2006].

Este proceso consta de las siguientes actividades:

- Implementación del proceso.
 - Identificación de la configuración.
 - Control de la configuración.
 - Determinación del estado de la configuración.
 - Evaluación de la configuración.
 - Gestión de releases y entrega.
-
- La Gestión de la Configuración es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. La meta es maximizar la productividad minimizando los errores [BAB, 1986].
 - La Gestión de la Configuración es una disciplina de gestión que permite controlar formalmente la evolución del software, garantizando la viabilidad en el desarrollo y en el producto y la trazabilidad en el producto [BRYAN-SIEGEL, 1984].
 - La Gestión de Configuración del Software es una actividad de autoprotección que se aplica a lo largo del proceso de ingeniería de software. Como el cambio se puede producir en cualquier momento, las actividades de GCS sirven para (1) identificar el cambio, (2) controlar el cambio, (3) garantizar que el cambio se implementa adecuadamente y (4) informar del cambio a todos aquellos a los que le interese [PRESSMAN, 1998].
 - La Gestión de la Configuración, es la disciplina que permite identificar la configuración del sistema en diferentes puntos del tiempo. La implementación de esta disciplina parte de unos lineamientos administrativos para identificar los artefactos (ítems) que van a entrar al control de configuración y técnicos para seleccionar las herramientas de apoyo al control de versiones [SWEBOOK, 2004].

- Gestión de la Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las soluciones de cambio, y verificando que los elementos sean correctos y completos [IEEE Std. 729-1983].
- Gestión de la Configuración es una disciplina que aplica conceptos técnicos y administrativos para identificar y documentar características funcionales y físicas de un ítem de configuración, controlar cambios en características, registrar y reportar procesamiento de cambios y estado de la implementación, verificar adecuación con requerimientos especificados [IEEE Std. 828-1998].

El Instituto de Gestión de la Configuración (Institute of Configuration Management) [CMII] proporciona la siguiente definición: Gestión de la Configuración es el proceso de administrar los elementos físicos y los procesos gestionando la información sobre ellos, incluyendo cambios, y asegurando la conformidad en cada caso. El ámbito de la Gestión de Configuración incluye toda la información que puede impactar en la seguridad, la calidad, la planificación, el costo, el beneficio o el entorno.

El énfasis de la Gestión de configuración esta en:

- (1) acomodar el cambio,
- (2) optimizar la reutilización de los estándares y mejores prácticas,
- (3) asegurar que todos los requerimientos se mantengan claros, concisos y válidos,
- (4) comunicar (1), (2) y (3) a cada usuario de manera exacta y precisamente y
- (5) verificar que los resultados sean conformes en cada caso.

El Modelo de Capacidad y Madurez (CMMI, del inglés Capability Maturity Model Integration), promueve la mejora continua desde (1) hasta (5).

El Cuerpo de Conocimiento de la Ingeniería de Software [SWEBOK 2004] define las siguientes actividades de la gestión de configuración:

- Administración del proceso de gestión de configuración de software
 - Contexto y restricciones
 - Planificación y Seguimiento
- Identificación de la configuración del software
 - Ítems a ser controlados

- Versiones del software
- Línea base del software
- Control de la configuración
 - Proceso para cambio
- Contabilidad del estado de la configuración
 - Información y reportes
- Auditorias de la configuración
 - Control líneas base
- Gestión de liberación y entrega del software
 - Definición y pautas

1.3.2. Conceptos en Gestión de la Configuración del Software.

A continuación definimos algunos de los conceptos más importantes en GCS.

- Elementos de Configuración de Software o Artefacto: Son todos los productos utilizados en el proyecto, documentos, código fuente, imágenes, manuales, etc, cuya evolución interna se desea controlar.
- Checkin: Acción acontecida cuando una copia local modificada de un artefacto es integrada al repositorio, esta actualización del artefacto produce generalmente una nueva versión del mismo.
- Checkout: Acción acontecida cuando se crea una copia local de un artefacto desde el repositorio, se puede utilizar una versión específica o por defecto la versión vigente del artefacto, esta copia local se realiza generalmente para realizar cambios o modificaciones sobre la copia.
- Bloqueo: La posibilidad de que varios usuarios necesiten realizar modificaciones sobre un mismo artefacto, produciría conflictos de versiones. El bloqueo sobre un artefacto se realiza para evitar estos conflictos, de esta forma es posible que un solo usuario realice modificaciones sobre un artefacto a la vez.
- Trazabilidad: La posibilidad de crear versiones de un mismo artefacto, hace necesario la posibilidad de seguir la traza en el histórico de las versiones para conocer la dependencia entre los artefactos.
- Soporte a equipos: Posibilidad de que el equipo de trabajo tenga acceso al repositorio y facilite la integración de los artefactos y sus cambios.
- Versión: es una instancia de un elemento de configuración. La creación y sucesivas modificaciones de los elementos de configuración pueden ser visualizadas en un grafo de evolución, como el que se muestra en la Figura 1.4.

Los elementos de software evolucionan conforme un proyecto de software progresa. Una versión es un elemento particular identificado y especificado. Puede ser tomado como un estado de la evolución del elemento [SWEBOK, 2004].

- Revisión: es una nueva versión de un elemento que es propuesta para reemplazar a la versión anterior del elemento [SWEBOK, 2004].
- Variante: es una nueva versión de un elemento que será añadida a la configuración sin reemplazar a la configuración anterior [SWEBOK, 2004].

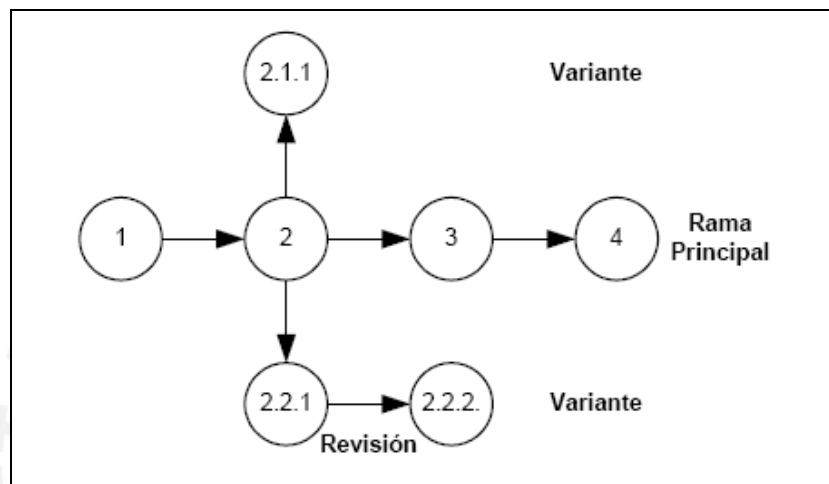


Figura 1.4. Grafo de evolución de versiones [SWEBOK, 2004].

1.4. RUP

El Proceso Unificado de Rational (RUP, del inglés Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye una de las herramientas más utilizadas para el análisis, implementación y documentación de sistemas orientados a objetos [JACOBSON, 2000].

RUP es un proceso de desarrollo de software, es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (define quién está haciendo qué, cuándo lo hace y cómo alcanzar cierto objetivo, en este caso el desarrollo de software [KRUCHTEN]), cuyo objetivo es asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones).

RUP tiene 3 importantes características [MELOCHE, 2002]:

- Manejo de Casos de Uso: el proceso emplea casos de uso para manejar el proceso de desarrollo desde el inicio hasta el despliegue.
- Centrado en la arquitectura: el proceso busca entender el más significativo aspecto estático y dinámico en términos de arquitectura de software. La arquitectura se encuentra en función de las necesidades del usuario y es capturada en el núcleo de los casos de uso.
- Iterativo e incremental: el proceso reconoce que es práctico dividir largos proyectos en medianos y pequeños proyectos. Cada pequeño proyecto comprende una iteración que resulta en un incremento. Una iteración podría abarcar todos los flujos en el proceso. Las iteraciones son planeadas usando los casos de uso.

Fases de RUP

El ciclo de vida de RUP, como se conoce al trazado de las actividades de desarrollo en el tiempo, está dividido en cuatro fases: inicial, elaboración, construcción y transición.

A continuación se describen los objetivos de cada una de las fases [LARMAN, 2003].

- Inicio: alcanzar un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto, generando el ámbito del proyecto, el caso de negocio, síntesis de arquitectura posible y el alcance del proyecto.
- Elaboración: establecimiento de la línea base para la arquitectura del sistema y proporcionar una base estable para el diseño y el esfuerzo de implementación de la siguiente fase, mitigando la mayoría de los riesgos tecnológicos.
- Construcción: completar el desarrollo del sistema basado en la línea base de la arquitectura.
- Transición: garantizar que el software está listo para entregarlo a los usuarios.

Flujos de Trabajo

En RUP se definen nueve flujos de trabajo distintos, separados en dos grupos.

Flujo de Trabajo del Proceso, las etapas de esta sección son:

- Modelado de negocio.
- Requisitos.
- Análisis y Diseño.

- Implementación.
- Pruebas.
- Despliegue.

Flujo de Trabajo de Soporte: En esta parte nos encontramos con las siguientes etapas:

- Gestión del cambio y configuraciones.
- Gestión del proyecto.
- Entorno.

Beneficios fuentes [LARMAN, 2003]

- Lograr gobernabilidad en tecnologías de información, mediante control y monitoreo en el ciclo de vida del desarrollo de software.
- Reducir la redundancia e incrementar la productividad.
- Promover el uso y re uso de activos en la organización.
- Mitigar riesgos en proyectos estratégicos de la organización.
- Unir al equipo de trabajo, simplificando su operación y monitoreo.
- Eliminar ambigüedades en la comunicación del equipo de trabajo.
- Generar calidad en los productos de trabajo.
- Automatizar de manera efectiva sus áreas de desarrollo.

En la Figura 1.5 se muestra los flujos de trabajo y las fases de RUP.

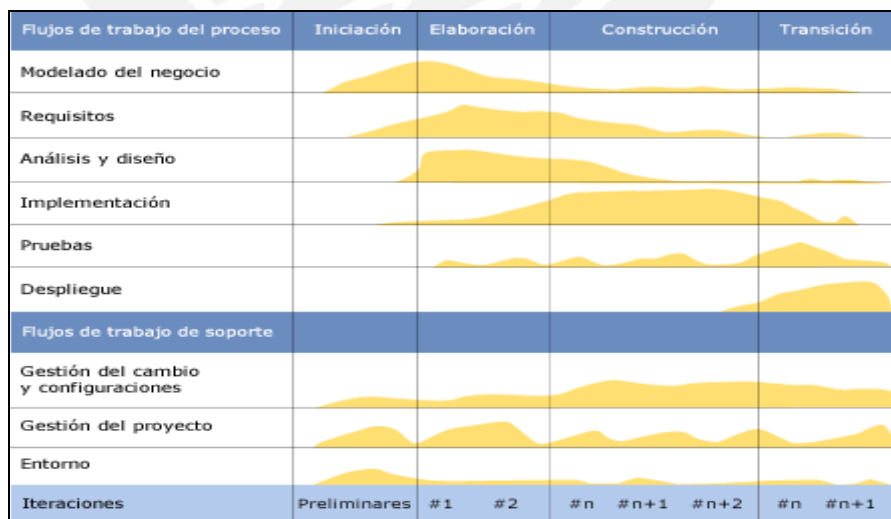


Figura 1.5. Los flujos de trabajo en cada una de las fases e iteraciones del RUP [JACOBSON, 2000]

En resumen, la meta de RUP es: [KRUCHTEN, 1999]

- Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- Proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.
- Reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

1.5. Lenguaje XPDL.

XPDL es un lenguaje para la definición de un flujo de trabajo, posee un formato de archivo basado en XML que puede ser usado para intercambiar modelos de procesos de negocio entre distintas herramientas y que utiliza la notación gráfica BPMN para la definición de un flujo de trabajo. Dicho lenguaje puede ser usado para almacenar o intercambiar modelos de procesos entre distintas herramientas [XPDL].

Los conceptos básicos que son la base de XPDL versión 1.0 fueron formulados por la WfMC (Workflow Management Coalition) y las compañías que desarrollaban herramientas de administración de procesos de negocio y del flujo de trabajo. Estos conceptos fueron incorporados en un meta-modelo y detallados en un glosario; los cuales fueron utilizados en la especificación de las diversas interfaces para los conceptos que formaban parte de un sistema del flujo de trabajo [BPMI].

La primera versión del lenguaje estándar de intercambio denominado WPDL (Workflow Process Definition Language) fue publicada por la WfMC en 1994 [BPMI], este lenguaje define la pieza esencial de la administración de procesos, el intercambio de las definiciones de procesos entre diversas herramientas y también distintos vendedores.

En Octubre del 2002 fue lanzado oficialmente el XPDL 1.0, este se basó en la combinación del WPDL en el flujo de trabajo y herramientas de BPMN. XPDL

mantuvo la semántica utilizada en WPDL pero definió una nueva sintaxis usando el esquema XML. Sin embargo, ni WPDL ni XPDL 1.0 propusieron una representación gráfica específica para el modelado de procesos a pesar que el metamodelo subyacente para un proceso estaba conformado por actividades (nodos) y caminos conectores entre ellos (transacciones).

XPDL, estándar que tiene por objetivo el archivo de los diagramas de procesos y el intercambio o portabilidad de estos entre distintas herramientas. Es un formato de archivo XML que representa el “dibujo” de la definición del proceso. Contiene el tamaño y las coordenadas X e Y del nodo. Tiene un concepto de líneas que señalan el camino a seguir. Los nodos y las líneas tienen atributos que pueden especificar información ejecutable tales como roles, descripción de actividades, temporizadores, llamadas a web services, etc. XPDL 2.0 contiene extensiones para ser capaz de representar todos los aspectos del BPMN (BP Modeling Notation) [XPDL_DOCS].

El objetivo de XPDL es proporcionar un lenguaje formal que permita la importación y exportación de las definiciones de procesos tanto a nivel gráfico y semántico entre una gran variedad de herramientas que hagan uso del mismo lenguaje. Esto ofrece una manera estándar para representar procesos de tal manera que puedan ser importados/exportados por cualquier aplicación que implemente el estándar [XPDL].

XPDL, especifica un formato de diseño de los procesos, permite una representación gráfica de los procesos incluyendo coordenadas X e Y para cada nodo implementado. Además, los nodos pueden especificar atributos tales como roles, descripción de actividades, temporizador, llamadas a servicios Web, etc. Suele ser preferido cuando se trata de implementar procesos o workflows con interacciones humanas [XPDL].

El 3 de octubre del 2005 fue lanzado XPDL versión 2.0, esta versión contempla los conceptos de diagramado propuestos por BPMN (Business Process Management Notation) y ofrece un meta-modelo extendido que unifica XPDL Y BPMN.

La Object Management Group, Inc. (OMG) junto con la Business Process Modeling Initiative (BPMI), han desarrollado una notación, denominada BPMN [BPMN_DOCS], para el modelado de procesos de negocio, BPMN define una notación para la definición de procesos de negocio, lo que es una plataforma

independiente con respecto a definiciones específicas (por ejemplo XPD) de procesos de negocio. Esta notación define una representación abstracta para la especificación de procesos de negocio que se ejecutan dentro de una empresa. Partiendo de un modelo BPMN se puede obtener, mediante la transformación, la definición de un proceso de negocio en un lenguaje específico.

BPMN fue desarrollado por BPMI (Business Process Management Initiative) con la finalidad de adoptar las técnicas empleadas en las herramientas de esquematización, así como unificar y extender los gráficos utilizados en ellas para expresar la semántica de los procesos. BPMN 1.0 fue lanzado en mayo del 2004, BPMN modela tanto la secuencia de actividades como los datos o mensajes intercambiados entre los distintos participantes [BPMN].

BPMN es un estándar de notación de proceso, es decir, define la forma gráfica de diseñar, modelar y construir un proceso, así como los diferentes objetos que se pueden utilizar para tal efecto. La característica principal y más destacable del estándar BPMN es que es un tipo de notación común entre las personas de negocio y los técnicos, construyendo un lenguaje común para intentar unir estos dos mundos [BPMN].

1.6. Proyecto COMPETISOFT

Es un esfuerzo iberoamericano que busca la competitividad de la industria desarrolladora de software a través del establecimiento de modelos de calidad en el proceso software.

COMPETISOFT tiene como objetivo desarrollar un marco metodológico común ajustado a la realidad socio-económica de las organizaciones desarrolladoras de software iberoamericanas, orientado a la mejora continua de sus procesos.

Este proyecto utiliza un modelo de procesos, un modelo de evaluación y un modelo de mejora siendo los modelos MoProSoft, EvalProsoft y Agile SPI los que han sido tomados como base respectivamente.

En esta sección se presentará COMPETISOFT-CYTED el proyecto en la PUCP (COMPETISOFT-PUCP) y en particular COMPETISOFT-PUCP tools.

1.6.1. COMPETISOFT- CYTED.

El Proyecto Competisoft se ha desarrollado sobre la base de los siguientes modelos: MoProSoft, EvalProsoft.y Agile SPI.

1.6.1.1. Modelo de procesos (MoProSoft)

Es un modelo que fomenta la estandarización de su operación a través de la incorporación de buenas prácticas basadas en los modelos y estándares reconocidos internacionalmente, tales como ISO 9001:2000, CMM-SW, ISO/IEC 15504, PMBOK entre otros, la comparación con estos modelos y estándares se muestra en la Tabla 1.1

Características	Modelos			
	ISO 9000:2000	CMMI-SW	ISO 15540	MoProSoft
1. Para SW.	NO	SI	SI	SI
2. Comprensible.	NO	NO	SI	SI
3. Procesos.	NO	SI	SI	SI
4. Práctico.	NO	NO	NO	SI
5. Mejora de procesos orientada al objetivo de negocio.	NO	NO	SI	SI
6. Evaluación con vigencia.	SI	NO	NO	NO
7. Aplicable como norma.	SI	NO	NO	NO

Tabla 1.1. Comparación de MoProSoft con otros modelos [MOPROSOFT, 2003].

La adopción del modelo permite elevar la capacidad de las organizaciones que desarrollan o mantienen software para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad. MoProSoft es también aplicable en áreas internas de desarrollo de software de las empresas de diversos giros. Actualmente MoProSoft es una norma mexicana [MOPROSOFT, 2003] y norma técnica peruana.

Las principales características de MoProSoft son: [MOPROSOFT, 2003]

- Pocos procesos que abarcan todos los niveles de una organización: directivo, gerencial y operativo.
- Procesos integrados como una red de comunicación.
- Definición explícita de roles responsables por las actividades de cada proceso y la capacitación requerida.
- Definición explícita del propósito, objetivos específicos, indicadores, metas cuantitativas y mediciones para cada proceso.
- Definición explícita de productos de entrada, salida e internos de cada proceso y sus características mínimas.

- Definición de flujos de trabajo con las actividades, tareas, roles involucrados y productos generados.
- Existencia de una Base de Conocimiento de la organización en la cual se resguardan todos los productos generados, se administran y se consultan de acuerdo con los mecanismos definidos.
- Definición de las actividades para recaudar lecciones aprendidas y usarlas en proyectos futuros.
- Definición de un mecanismo específico para la reacción a las situaciones excepcionales durante el desarrollo de las actividades.
- Definición explícita de las actividades de verificación, validación y pruebas para fomentar la calidad de los productos.
- Definición explícita de guías de ajuste que sugieren la adaptación de los procesos a las necesidades de las organizaciones, sin perder de vista el cumplimiento de los objetivos de los procesos.
- Los objetivos y metas cuantitativas son las que guían a los demás procesos y proyectos y son los que se evalúan para conocer cuantitativamente la efectividad de los procesos de la organización.
- Las sugerencias de mejora a los procesos se identifican y se reportan a los responsables de gestión de procesos.
- Los procesos del modelo pueden ser ajustados con base al contexto de la organización.

El propósito de contar con un modelo de estas características es apoyar a la industria de software a incrementar la productividad a un nivel donde la calidad de los productos de software será la consecuencia natural de la madurez en los procesos de las organizaciones.

MoProSoft proporciona a las pequeñas y medianas empresas desarrolladoras de software, un modelo basado en las mejores prácticas internacionales con los siguientes beneficios:

- Mejora la calidad del software producido por la empresa que adopta el modelo.
- Eleva la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.
- Integra todos los procesos de la organización y mantiene la alineación con los objetivos estratégicos.
- Inicia el camino a la adopción de los modelos ISO 9001 o CMMI.

- Sirve para implantar un programa de mejora continua.
- Facilita la selección de proveedores.
- Permite obtener acceso a las prácticas de ingeniería de software de clase mundial.

Estructura de MoProSoft

El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. Sintetiza las mejores prácticas en un conjunto pequeño de procesos que abarcan las responsabilidades asociadas a la estructura de una organización que son: la Alta Dirección, Gestión y Operación [MOPROSOFT].

MoProSoft es un modelo integrado donde las salidas de un proceso están claramente dirigidas como entradas a otros; las prácticas de planeación, seguimiento y evaluación se incluyeron en todos los procesos de gestión y administración; por su parte los objetivos, los indicadores, las mediciones y las metas cuantitativas fueron incorporados de manera congruente y práctica en todos los procesos; las verificaciones, validaciones y pruebas están incluidas de manera explícita dentro de las actividades de los procesos; y existe una base de conocimientos que resguarda todos los documentos y productos generados.

MoProSoft tiene tres categorías de procesos: Alta Dirección, Gestión y Operación que corresponden a la estructura organizacional de las empresas de software, como se muestra en la Figura 1.6

A continuación se describen las categorías de procesos:

1. Alta Dirección (**DIR**): Categoría de procesos que aborda las prácticas de Alta Dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos [MOPROSOFT].

Los elementos de entrada y salida de esta categoría son:

- Misión, Visión y Valores.
- Objetivos y la forma de medirlos, estrategias.
- Procesos con sus indicadores y metas.
- Cartera de proyectos.

- Estructura organizacional y estrategia de recursos.
- Presupuesto.
- Plan de comunicación con el cliente.
- Proceso de mejora continua.

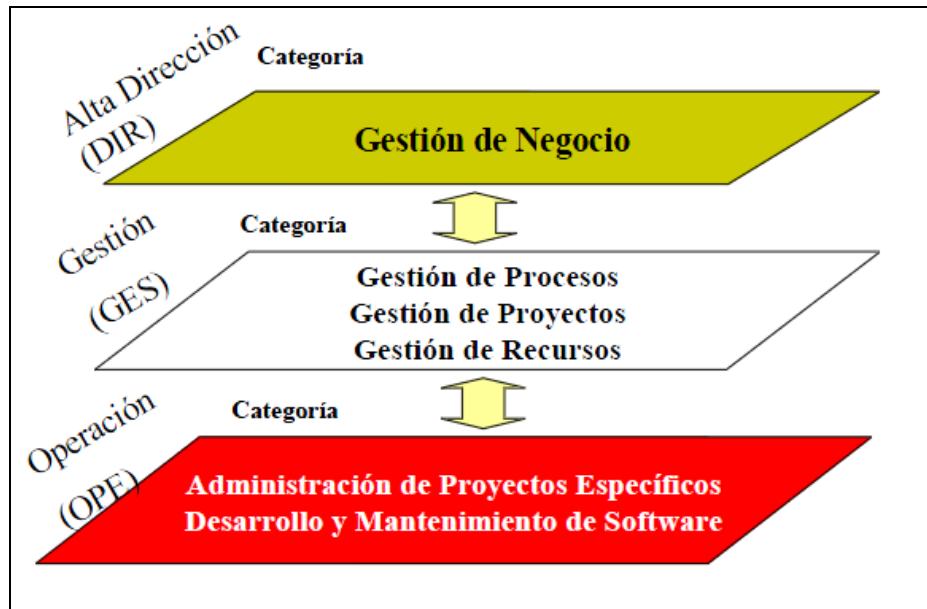


Figura 1.6. Modelo de Referencia de Procesos [MOPROSOFT].

2. Gestión (**GES**): Categoría que aborda la práctica de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos por la alta dirección. Proporciona los elementos para el funcionamiento de los procesos de la categoría de operación. Recibe y evalúa la información generada por estos y la comunica a la alta dirección [MOPROSOFT].

Esta categoría es compuesta por los siguientes elementos:

- Definir los elementos de los procesos, calendario y mediciones de procesos.
- Plan operativo de bienes, servicios e infraestructura, capacitación de recursos humanos y ambiente de trabajo.
- Plan de manejo de riesgos.
- Asignar responsables a proyectos.
- Implementación de procesos.
- Reporte de mediciones y sugerencias de mejoras.
- Plan de evaluación.

3. Operación (**OPE**): Categoría de procesos que aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados [MOPROSOFT].

Esta categoría es compuesta por los siguientes elementos:

- Definir el protocolo de entrega y tiempo estimado para cada actividad.
- Calcular costo estimado del proyecto.
- Documentar y realizar las actividades del plan de proyecto, el plan de desarrollo y evaluar su cumplimiento.
- Cierre del contrato.
- Generar reporte de mediciones sugerencias de mejora.

1.6.2. COMPETISOFT- PUCP.

Actualmente, el proyecto COMPETISOFT (Proyecto CYTED # 3789) involucra a 13 países (entre los cuales se encuentra el Perú) y 100 investigadores.

En el Perú, las empresas vienen introduciendo distintos marcos de referencia para la mejora de sus procesos, sin embargo estos no calzan con las distintas realidades que se presentan en las empresas peruanas, por lo que el Perú se convierte en un escenario propicio para implantar un marco metodológico orientado a las pequeñas y medianas empresas como el que propone COMPETISOFT.

COMPETISOFT- PUCP es un proyecto desarrollado por el Grupo de Investigación y Desarrollo en Ingeniería de Software de la Pontificia Universidad Católica del Perú (GIDIS-PUCP), el cual tiene como objetivo principal implantar un modelo de mejora de procesos de software en empresas dedicadas a este rubro en el Perú, planteándose las siguientes metas:

- Desarrollar herramientas de apoyo para la implantación del modelo de mejora.
- Desarrollar instrumentos de evaluación diagnóstica para COMPETISOFT.
- Evaluar el nivel de empresas peruanas según este modelo.
- Desarrollar mapeo de procesos entre modelos para apoyar a las empresas a elegir sus respectivos caminos de mejora.

El conjunto de proyectos que forman parte de COMPETISOFT-PUCP se encuentran categorizados de la siguiente forma:

- Plataforma para gestión de procesos: mejorar la gestión de procesos, metodologías y sus evoluciones en las empresas, utilizando un lenguaje de definición de procesos (XPDL y BPMN) a través de una plataforma en Internet que soporte la definición, administración, evolución, evaluación y auditoría de procesos.
- Procesos de mejora en empresas: mejorar la productividad de empresas desarrolladoras de software bajo el enfoque de adhesión a un modelo de referencia de procesos (MoProSoft). Este componente del proyecto se realiza a través de Action-Research, entre empresas y el equipo COMPETISOFT-PUCP. El esquema de trabajo implica que un estudiante entrenado en el modelo participe en una empresa como un practicante.
- Mapeo de modelos de procesos: determinar la correspondencia entre modelos de procesos referenciales más importantes respecto del nuevo modelo; para determinar el grado en que este nuevo modelo se alinea a los modelos existentes. El trabajo se realizará a través de un estudio comparado entre distintos modelos de referencia y evaluación a procesos.

Dentro de la primera categoría se encuentra el proyecto PS-PUCP, proyecto que congrega un conjunto de herramientas que permiten reflejar y manejar los procesos de una organización, evaluar dichos procesos en base a marcos de referencia y proveer una auditoría a través del monitoreo y mediciones en todos los pasos de un proceso.

1.6.3. COMPETISOFT- PUCP Tools.

El presente proyecto forma parte de una serie de herramientas desarrolladas por el Grupo de Investigación y Desarrollo en Ingeniería de Software (GIDIS) en el proyecto COMPETISOFT- PUCP Tools.

Los proyectos que se ejecutan son:

- Proyecto para la definición de procesos (MJS Process Designer): Permite definir procesos de una organización para posteriormente construir metodologías en base a los procesos definidos, se ha tenido que realizar extensiones al lenguaje XPDL para realizar la definición de las metodologías. Esta herramienta será desarrollada por el grupo COMPETISOFT – JAMESA.

- Proyecto para la gestión de proyectos (EACS Project Manager): Permite administrar proyectos y gestionar la configuración, los proyectos se basan en la metodología definida en XPDL por el Proyecto para la Definición de Procesos. Las actividades y artefactos que se generen en el proyecto también hacen referencia a las actividades y artefactos definidos en la metodología. Esta herramienta será desarrollada por el grupo COMPETISOFT – EVANCASA.
- Proyecto para la evaluación de procesos (LMB Process Audit)): Permite realizar una evaluación de los procesos definidos a través de los proyectos realizados y artefactos almacenados de dicho proyecto y usando la metodología definida por el tipo de proyecto. Es una herramienta basada en el lenguaje XPDL y será desarrollada por el grupo COMPETISOFT – LIMEBO.

En la Figura 1.7 se muestra la relación entre las 3 herramientas del proyecto COMPETISOFT-PUCP Tools. La presente herramienta se integrará a las herramientas: MJS Process Designer y herramienta LMB Process Audit.

La herramienta MJS Process Designer se encargará de generar una metodología en base a los procesos definidos en la organización evaluada en lenguaje XPDL.

La herramienta EACS Project Manager utilizará las metodologías generadas para implementar los proyectos, además las actividades y artefactos del proyecto podrán estar relacionados con las actividades y artefactos provenientes de la metodología.

La herramienta LMB Process Audit utilizará las metodologías y proyectos para realizar la evaluación y auditoría de las actividades y artefactos, de esta forma se realizará una comparación entre las actividades y artefactos de la metodología con los implementados en los proyectos.

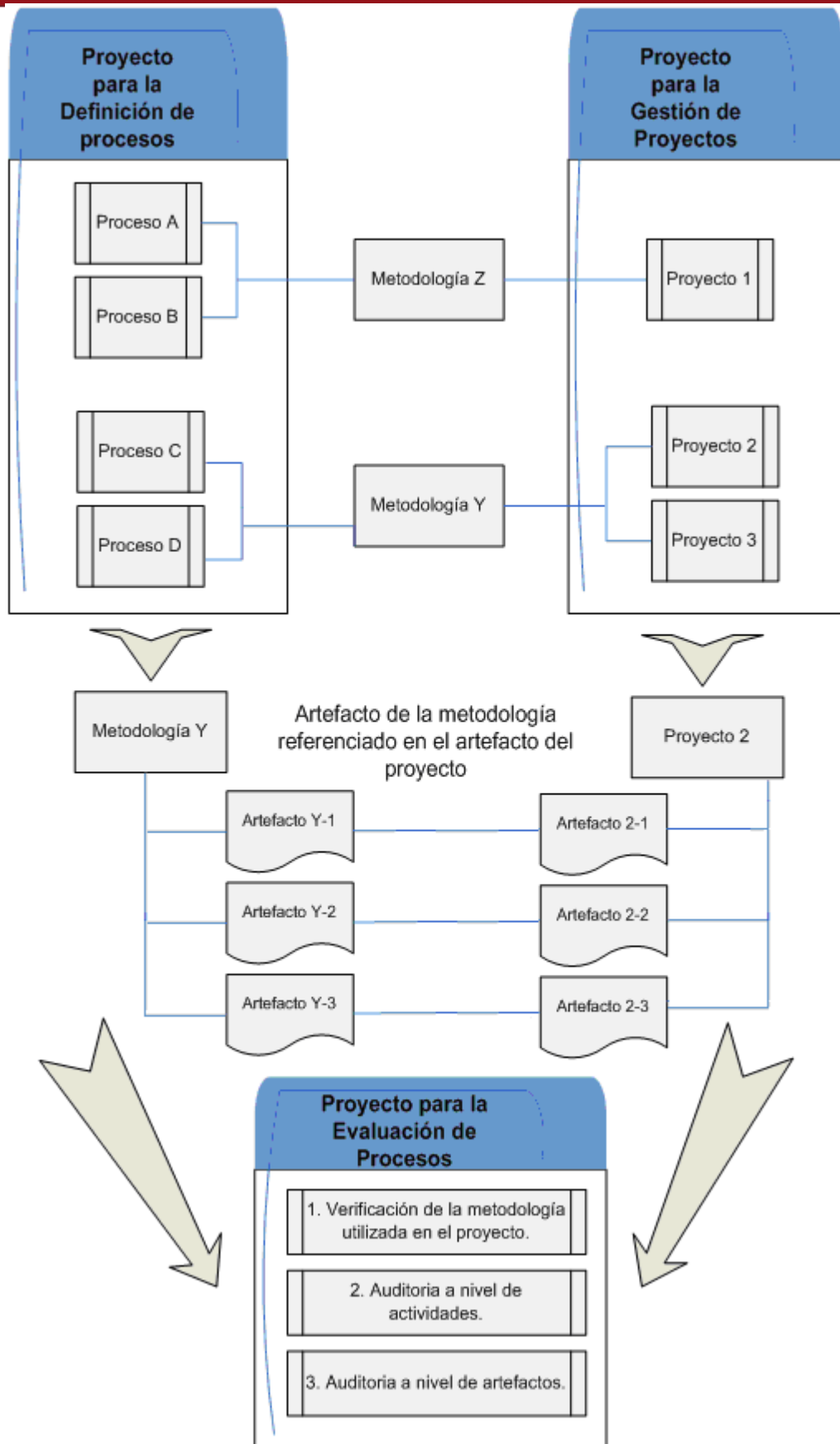


Figura 1.7. Relación entre las 3 herramientas del proyecto COMPETISOFT PUCP

Los siguientes códigos corresponden a la tesis desarrollada por el grupo COMPETISOFT – JAMESA y la explicación de la estructura y los “tags” se encuentran en detalle en dicha la tesis [MJS].

En el Código 1.1 se muestra la estructura de definición de procesos en la cual se define la metodología a seguir en el desarrollo del proyecto. Una metodología se compone de un conjunto de actividades, estas al relacionarse de forma adecuada se transforman en un flujo de procesos.

```

<xsd:complexType name="ProcessCategory">
  <xsd:choice>
    <xsd:element name="Normal">
      <xsd:element ref="mjsprocess:VersionInformation"
        minOccurs="0"/>
    </xsd:element>
    <xsd:element name="Metodologia">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="mjsprocess:VersionInformation"
            minOccurs="1" />
          <xsd:element ref="mjsprocess:ProcessesReference"
            minOccurs="1"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Explotacion">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="mjsprocess:ProcessParent" minOccurs="1" />
          <xsd:element ref="mjsprocess:ActivityParent" minOccurs="1" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ActivitySourceType">
  <xsd:choice>
    <xsd:element name="Normal"/>
    <xsd:element name="Metodologia">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="mjsprocess:ActivityReference" minOccurs="1" />
          <xsd:element ref="mjsprocess:ProcessReference" minOccurs="1" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Explotacion">
      <xsd:simpleType>
        <xsd:attribute name="LevelNumber" use="required" />
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
  
```

Código 1.1. Estructura definición de procesos [MJS].

En el Código 1.2 se muestra el conjunto de actividades definidas en la metodología asignada al proyecto, sirven para determinar el flujo del proceso de desarrollo del proyecto, cada actividad define una serie de artefactos a utilizar.

```

<xsd:element name="Activity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Description" minOccurs="0"/>
      <xsd:element ref="xpdl:Limit" minOccurs="0"/>
      <xsd:choice minOccurs="0">
        <xsd:element ref="xpdl:Route"/>
        <xsd:element ref="xpdl:Implementation"/>
        <xsd:choice minOccurs="0">
          <xsd:element ref="deprecated:BlockActivity"/>
          <xsd:element ref="xpdl:BlockActivity"/>
        </xsd:choice>
        <xsd:element ref="xpdl:Event"/>
      </xsd:choice>
      <xsd:element ref="xpdl:Transaction" minOccurs="0"/>
      <xsd:element ref="xpdl:Performer" minOccurs="0"/>
      <xsd:element ref="xpdl:Performers" minOccurs="0"/>
      <xsd:element ref="deprecated:StartMode" minOccurs="0"/>
      <xsd:element ref="deprecated:FinishMode" minOccurs="0"/>
      <xsd:element ref="xpdl:Priority" minOccurs="0"/>
      <xsd:choice minOccurs="0">
        <xsd:element ref="deprecated:Deadline" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element ref="xpdl:Deadline" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:choice>
      <xsd:element ref="xpdl:Documentation" minOccurs="0"/>
      <xsd:element ref="xpdl:TransitionRestrictions" minOccurs="0"/>
      <xsd:element ref="xpdl:InputSets" minOccurs="0"/>
      <xsd:element ref="xpdl:OutputSets" minOccurs="0"/>
      <xsd:element ref="xpdl:Object" minOccurs="0"/>
      <xsd:element ref="xpdl:NodeGraphicsInfos" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:choice minOccurs="0">
        <xsd:choice>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string" use="optional"/>
    <xsd:attribute name="StartActivity" type="xsd:boolean" use="optional"/>
    <xsd:attribute name="Status" use="optional" default="None">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="None"/>
        <xsd:enumeration value="Ready"/>
        <xsd:enumeration value="Active"/>
        <xsd:enumeration value="Cancelled"/>
        <xsd:enumeration value="Aborting"/>
        <xsd:enumeration value="Aborted"/>
        <xsd:enumeration value="Completing"/>
        <xsd:enumeration value="Completed"/>
      </xsd:restriction>
    </xsd:simpleType>
    </xsd:attribute>
    </xsd:attribute>
    <xsd:attribute name="StartQuantity" type="xsd:integer" use="optional" default="1"/>
    <xsd:attribute name="IsATransaction" type="xsd:boolean" use="optional" default="false"/>
  </xsd:complexType>
</xsd:element>

```

Código 1.2. Estructura de las actividades definidas en una metodología [MJS].

En el Código 1.3 se muestra los artefactos definidos en la metodología asignada al proyecto, sirven para definir los artefactos reales dentro del proyecto, los cuales serán utilizados para la administración del repositorio.

```

<xsd:element name="Input">
<xsd:complexType>
  <xsd:sequence>
    <xsd:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="ArtifactId" type="xsd:NMTOKEN" use="required"/>
  <xsd:element ref="mjsprocess:InputIdentifier" minOccurs="0" />
  <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
</xsd:element>

```

Código 1.3. Estructura de los artefactos definidos en una metodología [MJS].

1.7. Revisión de Productos.

En esta sección se presenta un breve resumen con las características más importantes de algunas de las herramientas de Gestión de proyectos que existen en el mercado así como las características más importantes de la herramienta propuesta.

1.7.1. Tecnológico

En la Tabla 1.2 se muestra las características más importantes de algunas herramientas de gestión de proyectos existentes en el mercado.

Características de las herramientas que existen en el mercado:

Microsoft Project (o MSP)

Es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo [MSPROJECT].

Características:

- Almacena la información del proyecto incluyendo fechas, recursos, asignaciones, duraciones, relaciones entre tareas, secuencias de tareas, calendarios y más.

Características	MS Project	B-KIN PROJECT MONITOR	BUGTRACK	KPLATO	dotProject	PRIMAVERA
Administración de Tareas	✓	✓	✓	✓	✓	✓
Retroalimentación de Tareas	x	✓	✓	✓	✓	✓
Calendarios	✓	✓	✓	✓	✓	✓
Reportes Financieros	✓	x	✓	✓	✓	✓
Plantillas de Proyectos	✓	x	✓	x	✓	✓
Administración de Recursos	✓	✓	✓	✓	✓	✓
Habilidades de los Recursos	✓	✓	✓	x	x	✓
Impresión de Reportes y Documentos	✓	✓	✓	✓	✓	✓
Administración de Artefactos	x	x	x	✓	✓	✓
Notificaciones vía correo electrónico	✓*	✓	✓	✓	✓	✓
Sistema en Línea	✓*	✓	✓	✓	✓	✓

Tabla 1.2. Características de las herramientas existentes en el mercado.

✓*: característica disponible en MS Project Server.

- Calcula la información incluyendo fechas, programaciones, costos, duraciones, rutas críticas, valor ganado, variaciones y más.
- Presenta las vistas de la información que se está recuperando. Se pueden especificar las vistas, las tablas, los filtros, los grupos, los campos o informes, dependiendo del aspecto que el modelo del proyecto necesite mostrar.
- Diagramas de Gantt e informes.
- Configuración de los recursos y asignaciones de los mismos a las tareas.
- Comparación de las variaciones entre la información de la tarea planeada y real.

B-KIN PROJECT MONITOR

Es una herramienta Web comercial que permite monitorizar proyectos, tareas, programas, personas, en general todos los aspectos y componentes de un proyecto

en forma distribuida. Permite también utilizar la herramienta de prueba gratuita por 30 días previa inscripción [B-KIN].

Características:

- En línea, permite la administración de proyectos bajo un entorno distribuido, la herramienta se encuentra alojada en el servidor de B-kin. La herramienta no se vende, simplemente se cobra una mensualidad por derecho de uso.
- Entorno web: aplicación 100% programada para ser utilizada vía Internet. Se trabaja siempre mediante un navegador (Internet Explorer o Firefox) desde cualquier lugar con acceso a Internet.
- Multiproyecto, permite realizar varios proyectos a la vez y formar grupos.
- Integración e-mail: Notificaciones a terceros vía mail de acciones y tareas. Recepción automática de mails
- Colaborativo, permite el flujo de información mediante foros y documentos compartidos dentro de cada proyecto.
- Permite generar una amplia gama de reportes para la revisión permanente y actualizada de los proyectos y tareas. Así mismo los reportes permiten ser exportados a formatos PDF, HTML, Microsoft Project y Microsoft Excel.

Módulos complementarios:

- **Mi BPM:** Permite a todos los integrantes del equipo informar sobre la dedicación de horas de trabajo en los distintos proyectos que se encuentren asignados.
- **B-kin Work Report:** Permite asignar tareas desde el exterior de la organización (por ejemplo: tareas de soporte técnico por terceros).

Aplicaciones complementarias:

- **B-kin Content Manager:** Permite administrar los documentos asociados al proyecto, noticias, foros y crear el portal Web del proyecto.
- **B-kin Web Galleries:** Permite alojar un espacio Web para compartir noticias, foros y documentación con clientes o personal externo a la organización.

BUGTRACK

Es una herramienta Web comercial que permite administrar proyectos bajo un entorno distribuido, la herramienta es alquilada mensualmente para su uso y existen

dos ediciones: Estándar y Profesional (permite realizar reportes y administrar repositorios de documentos).

La herramienta permite el seguimiento de un proyecto de software y los posibles errores de código [BUGTRACK].

Características:

- En línea, permite el acceso Web a la herramienta las 24 horas y los 365 días del año, solo se necesita conexión a Internet.
- Permite reportar nuevos errores del proyecto, para la posterior revisión de estos.
- Notificaciones vía correo electrónico cuando se realicen cambios sobre el proyecto o nuevas notificaciones de errores.
- Permite el versionamiento del proyecto.
- Administración de roles y acceso por cada proyecto.
- Seguridad, todos los enlaces están encriptados bajo seguridad SSL (Secure Socket Layer).
- Exportación de registros para una revisión fuera de línea.
- Web-query, permite la ejecución de consultas y la exportación de datos hacia otras herramientas.
- Integración con CVS, Subversión y Microsoft Visual Source Safe. Permitiendo de esta forma que los miembros del proyecto tengan siempre la versión correcta del proyecto.
- Proyectos compartidos, permite compartir el proyecto con otras compañías que tengan suscripción con BUGtrack.

KPLATO

Es una herramienta gratuita de escritorio de gestión de proyectos perteneciente a KOffice (office suite de KDE) [KPLATO].

Características:

- Permite mostrar el costo planificado del proyecto a una fecha determinada.
- Permite la organización de las tareas mediante en una estructura de división del trabajo (WBS).
- Los recursos son organizados por medio en una estructura de desglose de recursos (RBS).

- Los costos son organizados en una estructura de desglose de costos (CBS).
- Generación de Diagramas GANTT: tareas dependientes, nombre de la tarea, recursos asignados, tareas críticas.
- Cuadros de dialogo para crear y corregir el proyecto, diversos tipos de tareas, los calendarios, los recursos, los costos y el progreso del proyecto.

Características no incluidas

- No permite restringir las tareas, si ya se había ingresado una tarea no puede validarse si esta existía.
- Revisión en línea.

DotPROJECT

Es una herramienta Web de código abierto para la administración de proyectos. No existe una empresa encargada de la herramienta debido a que es mantenido por un grupo de voluntarios y usuarios.

El software es libre de descargarse desde la página Web de la herramienta, la empresa que desee utilizar la herramienta debe encargarse de hospedarlo en un servidor de aplicaciones [DOTPROJECT].

Características:

- Administración de usuarios, clientes y empresas.
- Jerarquía de lista de tareas
- Administración de repositorio de archivos.
- Permite generar un Calendario en base a los tiempos registrados en las tareas.
- Permite asignar recursos no humanos (oficinas, equipamiento, entre otros) a un proyecto.
- Permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto del foro.
- Permite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos.

Primavera Project Planner

Es el producto más importante de Primavera System Inc, empresa líder de software de programación de proyectos desde 1982 y, los proyectos de mayor complejidad a

nivel mundial se realizan con este software. En octubre del 2008 Oracle adquirió Primavera System [PRIMAVERA].

Características:

- Incorpora todos los elementos necesarios para garantizar un adecuado control y seguimiento de los proyectos.
- Permite la gestión de múltiples proyectos de gran tamaño.
- Permite crear grupos de proyectos, dispone de herramientas para realizar planificaciones y de nivelaciones avanzadas que pueden realizarse de forma manual o automática. Todo ello dentro de un entorno multiusuario, donde cada participante puede tener acceso a todo el proyecto o sólo a las partes deseadas mediante las capas.
- La comunicación de los planes entre los diferentes usuarios se realiza mediante correo electrónico o a través de páginas Web. Esta herramienta dispone de todas las características de gestión de proyectos que se puedan necesitar, por complicado que sea el proyecto.
- Integra un módulo para gestión de contratos de proyectos, Expedition, y una herramienta de seguimiento y control de proyectos, Sure Track Project Manager. Además, esta suite permite la integración con los sistemas ERP más utilizados en las empresas, como SAP, Oracle, Baan y People Soft.
- La aplicación dispone de una opción en la que permite seleccionar el idioma en el que se desea trabajar, entre otros el castellano. Los informes, calendarios, etc. aparecerán en el idioma especificado. Sin embargo no puede decirse que exista realmente una versión en castellano de la aplicación, pues las barras de menús y los campos seguirán apareciendo en inglés.
- Tiene un poderoso acceso Web por todos los usuarios y a todos los proyectos simultáneamente.
- Manejo fácil de Gantt (herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado)

Los principales módulos funcionales de este producto, en constante desarrollo, son:

- Cliente Windows orientado a los que ejecutan funciones de programación y/o control, elaboran cronogramas etc.

- Cliente Web o My Primavera orientado a los administradores de proyectos o Project Managers, directivos, ingenieros y técnicos a pie de obra así como a la realización de análisis del portafolio.
- Módulo Contract Manager para controlar los aspectos administrativos y contractuales de los proyectos.

1.7.2. Descripción y Sustentación de la Solución

En esta sección se presenta la definición del producto en base a las necesidades detectadas en el mercado y definidos en el proyecto COMPETISOFT-PUCP.

Como resultado del análisis de la situación actual, se diseñó la herramienta EACS Project Manager en busca de satisfacer las necesidades señaladas en capítulos anteriores. A pesar de existir una gran variedad de herramientas en el mercado, no se encontró ninguna que abarque el proceso de instanciar una metodología ya definida por algún proceso, manejo de sus actividades, artefactos de estas metodologías y gestión de la configuración haciendo uso de un lenguaje formal.

La presente herramienta forma parte del proyecto PS-PUCP la cual se integrará a las herramientas: MJS Process Designer la que se encargará de generar una metodología en base a los procesos definidos en la organización evaluada, y herramienta LMB Process Audit que utilizará las metodologías definidas y los proyectos registrados para realizar la evaluación, de esta forma se realizará una comparación entre las actividades y artefactos de la metodología con los implementados en los proyectos.

Otra de las bondades que la herramienta brinda al usuario es la posibilidad de crear un proyecto en base a una metodología ya definida en algún proceso. A este proceso se le denomina crear la instancia de una metodología que consiste en extraer de la metodología sus actividades y artefactos pertenecientes a un proceso o marco de referencia con la finalidad de asociarlas a una o más actividades o artefactos del proyecto creado.

La gestión de la configuración de los artefactos creados en el proyecto tiene un papel fundamental en la correcta gestión y administración de los proyectos dentro de una organización. En el módulo de Gestión de la Configuración, el usuario podrá crear y administrar versiones de los artefactos definidos en el sistema a lo largo del tiempo.

La herramienta EACS Project Manager se desarrollará en plataforma Web para facilitar la distribución de la aplicación y el acceso de los usuarios desde cualquier ubicación. Para ello, se utilizará un repositorio de datos común, en el cual se registrarán versiones de los artefactos manejados por las empresas que tengan acceso a la aplicación.

EACS Project Manager será desarrollada de tal forma que se pueda integrar a un conjunto de herramientas de administración de procesos que abarque a todos los miembros del proyecto PS-PUCP.

La herramienta será desarrollada usando el lenguaje XPDL para el manejo de los procesos. Con ello, se logra contar con una herramienta que además de satisfacer las necesidades del usuario, también asegure su interoperabilidad con otras herramientas que manejen el mismo lenguaje.

1.7.3. Características principales de la herramienta propuesta

Cuando se tiene un proyecto, es necesario tener un control de los recursos, tanto materiales, humanos y del tiempo. Siempre la necesidad de conocer el avance y estado de los proyectos que se desarrollan es importante, de otra forma se propician retrasos, desorganización y por ende pérdidas económicas en el proyecto.

La herramienta propuesta ofrecerá el seguimiento y control de cualquier tipo de proyecto de software, gestionando su avance, plazos, esfuerzos, recursos y ofreciendo la información necesaria sobre cada elemento.

Las principales características que la herramienta propuesta contiene, se mencionan a continuación:

- Definir y actualizar un proyecto, actividades, tiempos y recursos.
- Permite gestionar múltiples proyectos.
- Crear la instancia de una metodología, asociando un proyecto a una metodología ya definida por algún proceso.
- Clasificar los proyectos según una categorización definida por el usuario para agruparlos o sólo identificarlos.
- Permite registrar recursos que participan en determinado proyecto.

Las actividades son las tareas que realizan las personas involucradas en cada proyecto y éstas tienen las siguientes características:

- Permite registrar actividades y sus dependencias.
- Permite asociar una o más actividades de una metodología ya seleccionada con una actividad creada en el proyecto.
- Registrar el avance de las actividades conforme se desarrollan.
- Permite visualizar las actividades agrupadas por las actividades padre a la que fueron relacionadas.
- Mostrar el listado de las actividades del usuario, las cuales pueden ser filtradas por diferentes criterios de búsqueda.
- Permite registrar el esfuerzo realizado por cada usuario para las actividades seleccionadas.
- Permite visualizar las actividades asignadas a un miembro del equipo, mostrando que actividades son futuras, las que deberían haber empezado y actividades que les faltan para acabar en diferentes colores, con lo que permitirá al usuario detectarlas a tiempo. El color rojo indica las actividades cuyo tiempo transcurrido es mayor o igual al 80% del tiempo programado; si el tiempo transcurrido ha superado el tiempo programado, esta actividad se encuentra retrasada. El color anaranjado indica las actividades cuyo tiempo transcurrido es mayor o igual al 40% y menor o igual al 80% del tiempo programado. El color azul indica las actividades cuyo tiempo transcurrido es menor al 40% del tiempo programado.

Los artefactos son todos los elementos producidos en un proyecto y tienen las siguientes características:

- Permite por cada proyecto crear un repositorio para gestionar los artefactos que se utilicen en el proyecto.
- Permite asociar uno o más artefactos de una metodología ya seleccionada con un artefacto creado en el proyecto.
- Permite asociar una o más actividades del proyecto con un artefacto creado en el proyecto.
- Permite gestionar las versiones de los archivos, las cuales son almacenadas en el repositorio, que controla la creación de nuevas versiones.
- El usuario podrá subir y descargar los archivos de acuerdo al acceso registrado manteniendo la integridad de los elementos.
- El usuario podrá reservar y liberar los artefactos de acuerdo al acceso registrado manteniendo la integridad de los elementos.
- Permite evaluar y ejecutar los cambios a estos elementos de forma controlada.

Entre otras características encontramos:

- Permite registrar empresas.
- Permite registrar usuarios al sistema.
- Permite validar accesos a los diferentes proyectos.
- Permite generar diferentes tipos de reportes.
- Permite generar un diagrama de Gantt, en el cual se podrá visualizar el tiempo programado, las fechas de iniciación y terminación para las diferentes actividades a lo largo de un tiempo total determinado.



Capítulo 2: Construcción

El presente capítulo se dedica a presentar los aspectos más relevantes acerca de la construcción, las secciones desarrolladas, tecnología y consideraciones en esta etapa.

2.1. Consideración de Implementación.

En esta sección se presenta el lenguaje, las herramientas, entornos de desarrollo y tecnología considerada para la implementación de la herramienta.

El presente trabajo tiene como propósito académico mostrar que es posible crear una herramienta que pueda integrar el uso de archivos XPDL para la definición de procesos que se instancia a través de proyectos y para la gestión de la configuración.

La seguridad ha sido implementada solo a nivel de ingreso del usuario al sistema mediante un usuario y contraseña, además de la sección de gestión de versiones donde se verifica el usuario que reserva un artefacto. No se ha considerado el uso de archivos de bitácora (logs) debido a que la herramienta no ha sido desarrollada para fines comerciales, por lo que no es necesario almacenar los eventos,

excepciones y mensajes en archivos de bitácoras para posteriores evaluaciones o revisiones.

Para la integración de los componentes se ha aprovechado la modularidad de los marcos de trabajo utilizados y únicamente ha sido necesario copiar los paquetes y páginas, luego se han modificado los archivos de configuración que se describirán en el presente capítulo. Debido a esta modularidad y el desarrollo ordenado de los módulos, primero se crearon los módulos independientes y luego los más complejos o que tenían dependencia de los primeros, no se ha requerido utilizar un gestor de la configuración o de versiones.

2.1.1. Lenguajes de programación

Para este proyecto definimos como lenguaje de programación Java, la definición correspondió a los coordinadores del proyecto, nuestro equipo se encargó de definir los elementos a usar en coherencia con los demás componentes de la herramienta.

Es un lenguaje de programación orientado a objetos y que fue desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de la sintaxis de C y C++, pero tiene un modelo de objetos más simple, además elimina herramientas de bajo nivel que suelen inducir a muchos errores como la manipulación directa de punteros o memoria. [JAVA]

Java es de uso libre, y cuenta con una gran cantidad de herramientas, componentes y librerías desarrolladas por diferentes organizaciones que facilitan la programación. Además es independiente de la plataforma y permite distribuir aplicaciones que puedan luego ejecutarse tanto en entornos de uso libre como comerciales.

Entre sus principales características encontramos:

- Las aplicaciones Java están típicamente compiladas en un bytecode. En tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución.
- Utilización de un recolector automático de basura.
- Brinda la posibilidad de desarrollar applets que luego pueden ser incrustados en páginas HTML.
- Permite el manejo de Servlets y JSP en el lado del servidor, para el desarrollo de aplicaciones Web dinámicas.
- Posibilita el uso de extensiones como JDBC y JavaMail.

- Permite el manejo de marcos de trabajo como Struts, Webwork y Spring.
- Es multiplataforma.

2.1.2. Tecnologías usadas en la herramienta

Dentro de la arquitectura de la herramienta EACS Project Manager se han utilizado algunas de estas tecnologías debido a las diferentes ventajas que ofrecen para el cumplimiento de las características funcionales de la misma.

A continuación se listan cada una de las tecnologías usadas:

2.1.2.1. Java 2 Software Development Kit (J2SDK)

La base para poder operar cualquier producto que utiliza Java es el JDK de la plataforma correspondiente ("Write Once, Run Everywhere"), por lo cual antes de instalar el servidor de jakarta-tomcat, se debe instalar el JDK.

El paquete J2SDK contiene el entorno de desarrollo de Java de Sun. Sirve para desarrollar programas Java y proporciona el entorno de ejecución necesario para ejecutar dichos programas. También incluye un módulo (plugin) para que los navegadores puedan ejecutar programas Java [J2SDK].

El paquete J2SDK contiene appletviewer, extcheck, idlj, jar, jarsigner, java, javac, javadoc, javah, javap, jdb, keytool, native2ascii, orbd, policytool, rmic, rmid, rmiregistry, rmiregistry, serialver, servertool y tnameserv. Appletviewer ejecuta subprogramas (applets) Java fuera del contexto de un navegador.

Principales descripciones utilizadas en el proyecto [J2SDK]:

- Extcheck: comprueba en un fichero JAR los conflictos en el título y la versión con cualquier extensión instalada en el software JDK.
- Idlj: genera vínculos Java a partir de un fichero IDL.
- Jar: combina múltiples ficheros en un archivo JAR simple.
- Jarsigner: firma ficheros JAR (Java ARchive) y verifica las firmas e integridad de un JAR firmado.
- Java: lanza una aplicación Java iniciando el entorno de ejecución Java, cargando la clase especificada e invocando el método principal de esa clase.
- Javac: lee definiciones de clases e interfaces, escritas en el lenguaje de programación Java, y las compila en un fichero de clases de código de bytes.

- Javadoc: procesa las declaraciones y comentarios de documentación en un conjunto de ficheros fuente Java y genera un conjunto de páginas HTML con la descripción de clases, interfaces, constructores, métodos y campos.
- Keytool: es una utilidad para administrar llaves y certificados.
- Native2ascii: convierte ficheros que contienen codificaciones de caracteres no soportados a ficheros con codificaciones Latin-1 o Unicode.
- Rmiregistry: crea e inicia un registro de objeto remoto en el puerto indicado de la máquina actual.
- Servertool: proporciona una interfaz fácil de usar para programadores que permite registrar, desregistrar, iniciar y parar un servidor.
- Tnameserv: inicia el servidor de nombres Java IDL.

Otras descripciones [J2SDK]:

- Javah: genera las cabeceras C y los ficheros fuente necesarios para implementar métodos nativos.
- Javap: desensambla un fichero de clases Java.
- Jdb: es un depurador simple en línea de comandos para clases Java.
- Orbd: se usa para permitir a los clientes localizar e invocar transparentemente objetos persistentes en servidores en el entorno CORBA.
- Policytool: crea y maneja gráficamente ficheros acerca de las políticas de seguridad Java de la instalación.
- Rmic: genera cabos y ficheros de esqueleto de clases para objetos remotos a partir de los nombres de clases Java compiladas que contienen implementaciones de objetos remotos.
- Rmid: inicia el demonio de activación del sistema.
- Serialver: devuelve el serialVersionUID para una o más clases en un formato adecuado para copiarlo en una clase envolvente.

2.1.2.2. JSP (Java Servlet Pages)

Considerando el uso del lenguaje Java, se ha optado por utilizar la tecnología JSP para la creación y manejo de las páginas Web. A continuación se ofrece una breve descripción.

JSP es un acrónimo de Java Server Pages, que en castellano se puede traducir como “Páginas de Servidor Java”. Es una tecnología orientada a crear páginas Web

con programación en Java y que permite generar contenido dinámico en forma de documentos HTML, XML o de otro tipo [JSP].

La tecnología de JSPs es una extensión de la tecnología de Java Servlets. La principal ventaja de los JSP frente a otros lenguajes es que permite integrarse con clases java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento HTML en el archivo jsp.

La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información de manera que los diseñadores puedan diseñar y actualizar páginas sin aprender Java y los programadores Java pueden escribir código sin ocuparse de los aspectos de diseño de las páginas.

Las características ofrecidas por JSP como alternativa a la generación de contenido dinámico para la Web se pueden resumir en:

- Mejoras en el rendimiento.
- El contenedor servlet puede ser ejecutado como parte del servidor Web.
- Los java beans utilizados en páginas .jsp pueden ser utilizados en servlets, applets o aplicaciones java.
- Cambios realizados en el código HTML relativos a cómo son mostrados los datos, no interfieren en la lógica de programación y viceversa.
- Los diseñadores de páginas pueden centrarse en el código HTML y los programadores en la lógica del programa.
- Los desarrollos pueden hacerse independientemente.
- Las frecuentes modificaciones de una página se realizan más eficientemente.

2.1.2.3. Spring

Spring Framework o simplemente Spring, es un marco de trabajo de código abierto de desarrollo de aplicaciones para la plataforma Java. También hay una versión para la plataforma .NET, Spring .NET.

Spring Framework no obliga a usar un modelo de programación en particular, a pesar de ello, se ha popularizado en la comunidad de programadores en Java al

considerársele una alternativa y sustituto del modelo de Enterprise JavaBean. Por su diseño el marco de trabajo ofrece mucha libertad a los desarrolladores en java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria. [SPRING]

Uno de los principales objetivos de Spring es no ser intrusivo. Esto quiere decir, que aquellas aplicaciones configuradas para utilizar “Beans” mediante Spring no necesitan depender de interfaces o clases de Spring, obteniendo su configuración a través de las propiedades de sus "Beans". Este concepto puede ser aplicado a cualquier entorno, desde una aplicación J2EE hasta un applet.

Spring está compuesto por un conjunto de características, las cuales están agrupadas en seis módulos principales, como se muestra en la Figura 2.1.

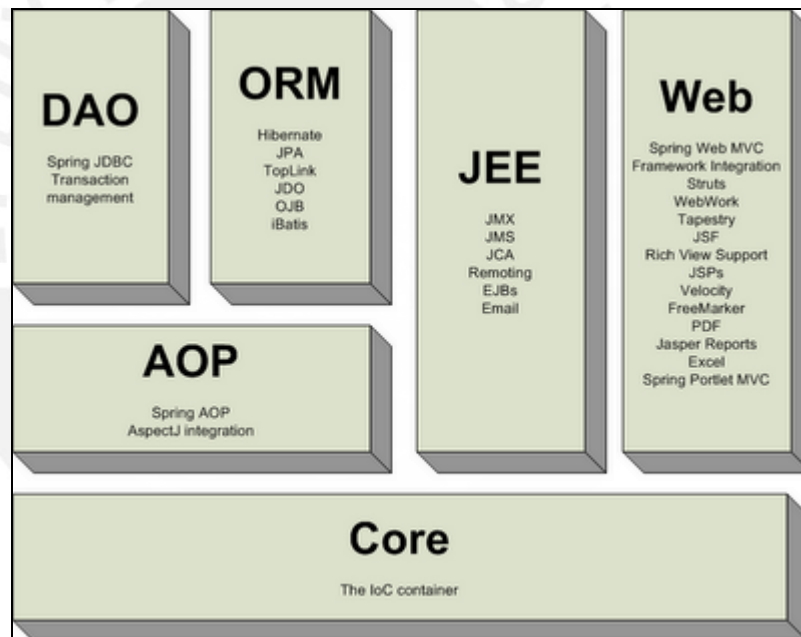


Figura 2.1. Módulos de Spring Framework [SPRING]

El paquete “Core” es la parte más fundamental del marco de trabajo ya que provee a este, las características de Inversión de Control (IoC) e Inyección de dependencias (ID). El concepto principal dentro del “Core” es el “BeanFactory”, el cual provee una sofisticada implementación del “Patrón de diseño Factory”, que remueve la necesidad generalizada de “Singletons” y nos permite desacoplar la configuración y especificación de las dependencias de nuestra lógica de programación.

El paquete “DAO” provee una capa de abstracción a JDBC, eliminando la tediosa codificación propia de JDBC y el parseo de los códigos de errores específicos de cada proveedor de base de datos.

El paquete “ORM” provee una capa de integración con las APIs más populares de “Mapeo Objeto-Relacional”, tales como JPA, JDO, Hibernate e iBatis. Debido a ello, se pueden utilizar estos ORM en combinación con todas las otras características que Spring ofrece.

El paquete “AOP” de Spring, provee una implementación para la programación “Orientada a Aspectos” compatible con el “AOP Alliance”.

El paquete “Web” de Spring provee características de integración orientadas a la Web, tales como funcionalidades para la carga de archivos, la inicialización del contenedor IoC usando “Servlet Listeners” y un contexto de aplicación orientado a la Web.

Mientras que las características fundamentales de este marco de trabajo pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en Web por encima de la plataforma empresarial de Java (Java Enterprise Platform).

Permite configurar las clases en un archivo XML y definir en él las dependencias. De esta forma la aplicación se vuelve muy modular y a la vez no adquiere dependencias con Spring, la introducción de aspectos, plantillas de utilidades para Hibernate, iBatis y JDBC así como la integración con JSF.

A nivel de soporte de la comunidad, Spring es uno de los proyectos con más actividad, con desarrollos dentro y fuera del propio marco de trabajo. Actualmente dispone de soporte comercial a través de Interface21, la empresa creadora, así como otros fabricantes que dan soporte en su área.

Entre las principales características de Spring utilizadas en la herramienta EACS Project Manager se incluyen [SPRING]:

- Inyección de Dependencia: radica en resolver las dependencias de cada clase generando los objetos cuando se arranca la aplicación y luego inyectarlos en los

demás objetos que los necesiten a través de métodos set o bien a través del constructor, pero estos objetos se instancian una vez, se guardan en una factoría y se comparten por todos los usuarios.

- Manejo de beans con contexto de aplicación: Spring puede organizar de forma efectiva nuestros objetos de la capa central. Además permite utilizar buenas prácticas de programación orientada a objetos, ya que utiliza interfaces.

2.1.2.4. Librería JavaScript

jQuery es una librería rápida y concisa de JavaScript que simplifica la manera de interactuar con los documentos HTML, el manejo de eventos, las animaciones y las interacciones con Ajax para el desarrollo Web [JQUERY].

jQuery está diseñado para cambiar la forma en que se usa JavaScript y se ha elegido para el presente trabajo por las ventajas que ofrece, entre las que figuran:

- Nos ahorra muchas líneas de código.
- Nos hace transparente el soporte de nuestra aplicación para los navegadores principales.
- Permite realizar una variedad de validaciones de una manera muy sencilla.
- Soporta los siguientes navegadores: Firefox, Internet Explorer, Safari, Opera y Chrome.
- Nos provee de un mecanismo para la captura de eventos.
- Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
- Integra funcionalidades para trabajar con AJAX y admite un gran número de parámetros.
- Existe una extensa comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, efectos diversos, etc.

2.1.2.5. Librería para diagrama de Gantt

Para mostrar el reporte con el diagrama de Gantt de cada proyecto se evaluaron varias librerías libres, la más conocida es JfreeChart sin embargo la que finalmente se utilizó para realizar los reportes es jsGantt. A continuación se presenta la comparación de ambas librerías.

JFreeChart

JFreeChart es una biblioteca java 100% libre que facilita a los desarrolladores mostrar gráficos de calidad profesional en sus aplicaciones de manera sencilla.

Es importante tener en cuenta la dependencia que la librería en si misma tiene de otro proyecto de código abierto, el proyecto JCommon.

Jcommon, contiene una colección de clases utilitarias usada por JFreeChart, Pentaho Reporting y puede ser utilizado en cualquier tipo de aplicación Java que necesite manejar fechas, serialización, etc [JFREECHART].

Entre las características principales de esta biblioteca tenemos:

- Un API consistente y bien documentado con soporte para un amplio rango de tipos de gráficos.
- Un diseño flexible fácilmente extendible, y la posibilidad de ser usado tanto en tecnologías de servidor (aplicaciones Web) y de cliente (Swing, por ejemplo).
- Soporte para varios tipos de salida, incluyendo componentes Swing, archivos de imagen como PNG y JPEG, y formatos gráficos de vectores (incluyendo PDF, EPS y SVG).

JFreeChart es de fuente abierta o más específicamente software libre, éste está distribuido bajo los términos de GNU Lesser General Public Licence (LGPL), que permite el uso en aplicaciones propietarias.

Entre los principales tipos de gráficos encontramos:

- Los gráficos XY (línea, estrías y dispersión).
- Los gráficos circulares.
- Diagramas de Gantt.
- El cuadro de barras (horizontales y verticales, apilados e independientes).
- Gráficas de intervalos de tiempo.

Si bien JFreeChart es una librería bastante difundida y documentada, esta no lograba integrarse adecuadamente al proyecto web, debido a que este proyecto contiene una serie de librerías y Servlets propios de JSF y SPRING. El formato XML que posee JSF para mostrar las páginas JSP dificultaba la inserción del código necesario para mostrar el diagrama Gantt del JFreeChart, ya que esta librería al ser codificada en JAVA necesitaba de un Servlet especial para esta tarea.

jsGantt

jsGantt es una librería gratuita desarrollada por Shlomy Gantz / Bluebrick inc y Brian Twidt. La librería es distribuida bajo licencia BSD, la cual nos permite redistribuirla y modificarla siempre y cuando se indique el nombre de los creadores de la librería.

jsGantt se encuentra codificado íntegramente en Javascript, CSS y con soporte AJAX, a diferencia de la librería anterior, ésta únicamente permite realizar diagramas de Gantt [JSGANTT].

Las características principales de la librería son:

- Permite mostrar las actividades y sus grupos en cascada, según la dependencia entre estas.
- Diferenciación entre actividades por colores, para el caso de este proyecto se ha utilizado una diferenciación de colores según la fecha final de cada actividad, de esta manera, el color rojo indican las actividades que les quedan menos del 20% del tiempo para que finalice, el color anaranjado indican las actividades que falta más del 20% pero menos de 60% del tiempo para terminar y las de color azul indican las actividades que faltan más del 60% del tiempo para terminar.
- También permite mostrar gráficamente el porcentaje de avance por cada actividad.
- La generación del diagrama de Gantt es realizado vía Javascript; los datos del diagrama serán obtenidos directamente llamando a las funciones javascript con sus parámetros correspondientes o también puede leerse desde un archivo XML que contenga los datos en un formato establecido por jsGantt.

Para superar los problemas ocurridos con la librería anterior se tenía que utilizar una librería que se pueda comunicar adecuadamente con los Servlets de JSF o que en todo caso no dependa del código JAVA configurado para el proyecto.

jsGantt al ser una librería codificada completamente en javascript no depende del código JAVA, motivo por el cual pudo integrarse con las páginas JSF, simplemente colocando los scripts necesarios.

La utilización del diagrama de Gantt permitió mostrar las actividades según su distribución de acuerdo al calendario, de manera tal que se pueda visualizar el período de duración de cada actividad, sus fechas de iniciación y terminación e

igualmente el tiempo total requerido para la ejecución de una actividad determinada.

2.1.3. Marco de trabajo de Presentación.

A continuación, analizaremos varias tecnologías y soluciones existentes que aportan soluciones para la capa de presentación entre los que se puede elegir: Java Server Faces, Struts, Spring MVC y Tapestry.

Java Server Faces

Java Server Faces (JSF) es un marco de trabajo que se basa en el modelo MVC (Modelo Vista Controlador) y en la interacción de componentes. Al igual que Struts, JSF pretende normalizar y estandarizar el desarrollo de aplicaciones Web, fue realizada por la empresa Sun y está basada en tecnología Java [JSF].

JSF tiene un alto nivel de abstracción que permite una mayor separación entre el diseño y el código permitiendo dividir realmente el desarrollo de una aplicación en diseño y desarrollo, donde los diseñadores aunque todavía necesitan tener unos conceptos básicos de la tecnología no tienen que preocuparse por la lógica de la aplicación.

JSF es un entorno de desarrollo de interfaces de usuario para aplicaciones Web creadas en Java. Está diseñado para facilitar el desarrollo y mantenimiento de las aplicaciones que se ejecutan en un servidor y devuelven el resultado en forma de interfaz al cliente [JSR 127].

Entre los principales beneficios de JSF se encuentran:

- Una clara separación entre vista y modelo.
- Desarrollo basado en componentes, no en peticiones.
- Las acciones del usuario se vinculan muy fácilmente al código en el servidor.
- Creación de familias de componentes visuales para acelerar el desarrollo.
- Ofrece múltiples posibilidades de elección entre distintas implementaciones.
- El código JSF con el que se crean las vistas (etiquetas jsp) es muy parecido al HTML estándar. Lo pueden utilizar fácilmente desarrolladores y diseñadores Web.
- Resuelve validaciones, conversiones, mensajes de error e internacionalización.

- Es independiente del tipo de aplicación a desarrollar, existiendo implementaciones tanto para aplicaciones Web como para aplicaciones clientes o incluso para aplicaciones en dispositivos móviles.
- Los componentes JSF están basados en tecnologías estándar que se pueden visualizar en cualquier navegador moderno.

La misión de JSF es aislar al programador tanto del tipo de cliente que hace la petición como del protocolo usado para tal menester, permitiendo la creación de las vistas de una manera abstracta, usando un modelo de componentes estable, estándar y extensible. De esta manera al aislarse del entorno y sus particularidades, los desarrollos pueden ser más productivos, rápidos y eficaces, al usar componentes ya probados.

La tecnología está diseñada para ser flexible, por lo que establece los conceptos sin limitar a los desarrolladores a utilizar un lenguaje concreto. Los componentes de interfaz de usuario incluidos con la tecnología JSF encapsulan la funcionalidad del componente, no la presentación, permitiendo a éstos ser presentados en distintos dispositivos y permitiendo al programador a hacer uso de tags específicos de acuerdo al dispositivo esperado. Un cliente típico son las páginas HTML, por lo que se provee de una librería de etiquetas para su uso.

La tecnología incide en la importancia de la separación de los datos de su presentación y en la división de los roles de desarrollo.

Struts

Struts es la implementación de código abierto más popular del patrón MVC para Java, mantenida por la Fundación Apache. Aunque es el marco de trabajo que ofreció un camino hacia el MVC en Java para desarrollos Web, es también uno de los que más ha acusado el paso de los años en tiempo de desarrollo y flexibilidad [STRUTS].

Las facilidades de desarrollo que ofrece son:

- Lógica de navegación entre páginas.
- Binding(mapeo) entre java y el HTML.
- Validación de entradas.
- Internacionalización.
- Independencia del motor de visualización.
- Maquetación.

Hoy existen marcos de trabajo que comparten los mismos principios pero con más potencia, elegancia y flexibilidad.

Spring MVC

Es uno de los módulos del marco de trabajo de Spring y como su propio nombre nos indica implementa una arquitectura MVC. Una de las alternativas a Struts que ha incorporado una lógica de diseño más sencilla y que cuenta con todo el abanico de librerías de Spring. No obstante sigue también la misma filosofía y no ofrece nuevas mejoras que sean comparables a JSF [SPRING MVC].

Tapestry

Es un marco de trabajo de código abierto para la creación de aplicaciones Web de forma dinámica, robusta y altamente escalable en Java. Tapestry divide una aplicación Web en un conjunto de páginas, cada una compuesta de componentes. Reconocido como uno de los marcos de trabajo más potentes para la Web, es también uno de los más complejos y de los que tienen una curva de aprendizaje más pronunciada [TAPESTRY].

Ofrece un sistema de componentes al igual que JSF pero no dispone de la comunidad, la documentación, el soporte y por supuesto la estandarización de JSF.

Considerando las definiciones anteriores y las características que ofrece cada marco de trabajo, se ha elegido JSF para implementar la capa de presentación de la herramienta EACS Project Manager, ya que esta capa recoge la entrada del usuario, presenta los datos, controla la navegación por las página y delega la entrada del usuario a la capa de la lógica del negocio.

Entre las características de JSF utilizadas en la herramienta EACS Project Manager se incluyen:

- **Control de Beans-de-Respaldo:** los beans de respaldo son componentes JavaBeans asociados con componentes UI utilizados en la página. El control de beans-de-respaldo separa la definición de los objetos componentes del UI de los objetos que realizan el procesamiento específico de la aplicación y que además contienen los datos. La implementación de JSF almacena y maneja estos ejemplares de beans de respaldo en el ámbito apropiado.

- Modelo de componentes UI extensible: los componentes UI de JSF son elementos configurables, reutilizables que componen las interfaces de usuario de aplicaciones JSF. Se puede extender los componentes UI estándar y desarrollar componentes más complejos, como barras de menú y árboles.
- Definir la navegación de las páginas: el manejador de navegación proporciona una simple y poderosa forma de controlar la navegación del usuario a través de la aplicación. La especificación JSF permite definir reglas de navegación que permite enlazar las vistas que verá el usuario en función a los resultados de las operaciones que realiza. Se define la navegación de la aplicación mediante el fichero de configuración.
- Modelo de renderizado flexible: un renderizador separa la vista y la funcionalidad de los componentes UI. Se pueden crear y utilizar varios renderizadores para definir diferentes apariencias del mismo componente para el mismo o diferentes clientes.
- Modelo de conversión y validación extensible: basados en los convertidores y validadores estándar, se pueden desarrollar convertidores y validadores personalizados, que proporcionan un mejor modelo de protección.
- Personalización de los mensajes de error: en JSF, se pueden configurar paquetes de recursos y personalizar los mensajes de error para convertidores y validadores.

2.1.4. Marco de trabajo de Lógica del negocio.

Para la construcción de la capa de lógica de negocio se puede utilizar EJB (Enterprise Java Beans) o POJO (Plain Old Java Objects). Dichos conceptos se muestran a continuación:

POJO (Plain Old Java Objects)

Es un acrónimo de Plain Old Java Object, sigla creada por Martin Fowler, Rebecca Parsons y Josh MacKenzie en septiembre del año 2000 y utilizada por programadores en el lenguaje de programación Java para enfatizar el uso de clases simples y que no dependen de un marco de trabajo en especial.

Surgió como una reacción en el mundo Java a los marcos de trabajo cada vez más complejos, y que requieren un complicado armazón que esconde el problema que realmente se está modelando. En particular surgió en oposición al modelo planteado por los estándares EJB anteriores a la versión 3.0, con el objetivo de

lograr una revalorización de la programación simplemente orientada a objetos. [POJO].

EJB (Enterprise Java Beans)

Es un acrónimo de Enterprise Java Beans. Es una interfaz que forma parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, además tienen las siguientes características:

- Comunicación remota utilizando CORBA.
- Transacciones.
- Control de la concurrencia.
- Eventos utilizando JMS (Java messaging service).
- Servicios de nombres y de directorio.
- Seguridad.
- Ubicación de componentes en un servidor de aplicaciones.

Los EJBs proporcionan un modelo de componentes distribuido estándar del lado del servidor, cuyo objetivo es permitir al programador abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad, entre otros) para centrarse en el desarrollo de la lógica de negocio. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. [EJB].

Considerando las definiciones anteriores y las características que ofrece cada marco de trabajo, bastaría utilizar POJO para implementar la capa de lógica del negocio de la herramienta EACS Project Manager, puesto que no se requiere que sea un sistema distribuido.

2.1.5. Entorno de desarrollo integrado (IDE)

Considerando el lenguaje de programación definido (Java), se planteó la posibilidad del uso de dos entornos de desarrollo de software libre bastante reconocidos: NetBeans y Eclipse, cuyas descripciones se muestran a continuación.

NetBeans IDE

Entorno de desarrollo de uso libre desarrollado originalmente por Sun Microsystems. Se apoya en el J2SDK. Netbeans es un entorno integrado de desarrollo escrito en java que agrupa un conjunto de utilidades para facilitar la edición, compilación, depuración, análisis y ejecución de cualquier programa java [NETBEANS].

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios y una comunidad en constante crecimiento.

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE.

NetBeans IDE es un producto libre y gratuito sin restricciones de uso. También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio.

Las principales características de NetBeans son:

- Mejoras en el editor de código.
- Soporte para Ruby, JRuby, y Ruby on Rails.
- Instalación y actualización más simple.
- Enlazar datos con el Swing GUI.
- Profiling integrado, profiling “points”.
- Características visuales para el desarrollo Web.
- Creador gráfico de juegos para celulares.
- Mejoras para SOA y UML.
- Soporte para PHP.
- Desarrolla aplicaciones colaborativamente.
- Mensajería instantánea.
- Comparte el entorno de desarrollo completo.
- Edición de un archivo por varios.
- Revisión de código.
- Debugging colaborativo.

Eclipse IDE

Eclipse IDE es un entorno de desarrollo integrado y software libre (bajo la licencia GNU GPL), que permite el desarrollo de aplicaciones en casi cualquier lenguaje de programación, incluyendo Java, C/C++ y Python.

La idea de su construcción es que sea una especie de herramienta universal de código abierto y que sirva para todo y para nada en particular [OTI].

Este entorno emplea módulos (llamados plug-in) para proporcionar toda su funcionalidad según el requerimiento de uso del usuario. Esto lo diferencia de otros entornos de desarrollo, puesto que dichos entornos son monolíticos ya que incluyen todas las funcionalidades, así las necesite el usuario o no. Por esta característica, Eclipse IDE es bastante utilizado ya que no consume muchos recursos de CPU [ECLIPSE]. Sin embargo, la instalación de este entorno es un tanto complicada puesto que el usuario debe conocer previamente cuáles son las funcionalidades que necesita según los módulos existentes e instalar cada uno por separado.

Considerando las definiciones anteriores, se ha optado por utilizar el IDE de NetBeans ya que ofrece buena estabilidad, usabilidad y primordialmente porque incorpora la instalación de Tomcat como servidor Web y Glassfish como servidor de aplicaciones, lo cual ofrece una ventaja en cuanto a menor costo y facilidad de instalación de los componentes requeridos para la arquitectura definida.

2.1.6. Procesamiento de datos XML

Actualmente hay tres métodos principales para procesar datos XML en aplicaciones Java: SAX, DOM y data binding. Los dos primeros están centrados en los documentos, es decir, manejan los datos en representación XML y trabajan dentro de la estructura del documento XML. La técnica de data binding, en cambio, sigue un enfoque completamente diferente y busca mapear los componentes XML a clases Java. Seguidamente se describen someramente estos tres métodos:

SAX

SAX (Simple API for XML) [SAX] es una API de bajo nivel basada en un mecanismo de eventos para parsear documentos XML. SAX reporta eventos de parseado tales como el comienzo y el fin de documento, el comienzo y el fin de elementos, y datos de tipo carácter, que hacen saltar métodos callback en una aplicación que

implementa un manejador de eventos. En la Figura 2.2. se muestra el funcionamiento de SAX.

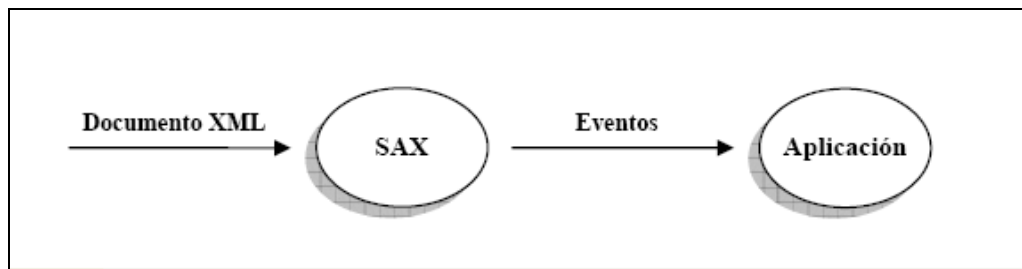


Figura 2.2. Funcionamiento de SAX [SAX]

SAX realiza un acceso secuencial al documento sin crear un modelo de objetos del mismo en memoria, lo cual convierte a esta API en un método de procesamiento de documentos XML con bajo coste computacional. Sin embargo, la falta de una estructura de datos residente en memoria es un importante inconveniente en el caso de aplicaciones que necesitan mantener información de estado, por ejemplo aplicaciones que necesitan conocer qué elementos han sido anteriormente parseados. Además, la programación de un manejador de eventos SAX puede ser una tarea tediosa y propensa a errores.

DOM

DOM (Document Object Model) es una API que se posiciona en un nivel ligeramente superior y que permite a programas y scripts el acceso dinámico y la actualización de documentos XML.

DOM crea en la memoria una estructura en árbol del documento XML que representa objetos con función e identidad propios [DOM]. Una vez construido el DOM las aplicaciones pueden procesarlo fácilmente: añadir y eliminar elementos, reposicionarlos, cambiar su contenido y escribir la estructura de datos resultante de nuevo en el documento XML son algunas de las posibilidades. En la Figura 2.3. se muestra el funcionamiento de DOM.

Aunque la API de DOM es mucho más fácil de usar y proporciona más flexibilidad que SAX, como por ejemplo en la manipulación de documentos mediante acceso aleatorio a sus contenidos, DOM no siempre resulta apropiado. El consumo de memoria y la velocidad de procesamiento, debido a la creación del modelo del documento, dependen del tamaño del documento y en el caso de documentos

grandes requiere una gran cantidad de memoria e influye negativamente en la velocidad de la aplicación.

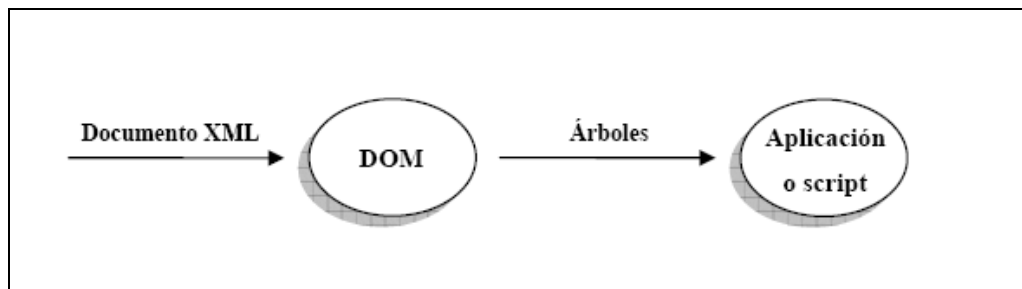


Figura 2.3. Funcionamiento de DOM [DOM]

Data binding

Data binding es el proceso de mapeo de los componentes de un determinado formato de datos a su representación específica para un lenguaje de programación concreto que se ajusta al significado propio de los datos. Esto permite modelar la estructura lógica de los datos sin seguir forzosamente la estructura impuesta por el formato en el que se habían guardado [BINDING_XML]. Más específicamente, el data binding entre XML y Java se refiere al mapeo de los componentes estructurales de XML, como elementos y atributos, al modelo de objetos de Java. De esta manera se preserva la jerarquía lógica de los componentes y se expone su significado real representándolos en el formato nativo del lenguaje de programación Java. En la práctica esto conduce a interactuar con objetos, tipos primitivos y estructuras nativas de Java en lugar de trabajar con modelos de eventos o árboles de parseado.

En comparación con SAX y DOM, el data binding entre XML y Java presenta una serie de ventajas [DBINDING]:

- Provisión de un mecanismo eficiente de validación de datos XML.
- Facilidad para crear un documento XML a partir de objetos Java y viceversa.
- Legibilidad del código generado, lo cual facilita el mantenimiento de las aplicaciones.
- Reducción del esfuerzo dedicado al desarrollo.

Tras haber presentado los enfoques de procesamiento de datos XML correspondientes a SAX, DOM y data binding, podemos ahora compararlos utilizando los siguientes criterios:

- Validación de los datos: se refiere a la provisión de un mecanismo que valide los datos contenidos en un documento XML.
- Creación de un modelo de objetos: denota la generación de una estructura de objetos en memoria para representar el documento XML.
- Utilización de memoria: hace mención a la cantidad de memoria necesaria para la representación del documento XML.
- Esfuerzo de desarrollo: denota la cantidad de programación necesaria por parte del desarrollador de aplicaciones para crear y mantener una aplicación de procesamiento XML.

La tabla 2.1. resume de manera compacta los resultados de la comparación en base a los criterios expuestos:

	SAX	DOM	Data Binding
Validación de datos	NO	NO	SI
Creación de un modelo de objetos	NO	SI	SI
Utilización de memoria	BAJA	ALTA	VARIABLE
Esfuerzo de desarrollo	ALTA	MEDIO	BAJO

Tabla 2.1. Comparación de enfoques de procesamiento de datos XML en aplicaciones Java

Una vez identificados los beneficios de la elección del data binding entre XML y Java sobre SAX y DOM, que usaremos en la herramienta EACS Project Manager para procesar documentos XML, explicaremos más detalladamente qué es y cómo se realiza el data binding entre XML y Java.

Por medio de un compilador de esquemas se generan directamente clases Java específicas para un esquema XML origen, ver Figura 2.4. Dichas clases incluyen código necesario para la validación y comprobación de errores, junto con un completo conjunto de métodos de acceso (métodos set y get) para asegurar la consistencia de los documentos XML con el esquema XML origen [DBINDING]. Partiendo de estas clases instanciaremos objetos Java que realizarán el marshalling y el unmarshalling de documentos XML. Los términos marshal y unmarshal se refieren a la acción de convertir un objeto a un flujo de datos y viceversa.

Estos objetos Java parsearán los documentos XML para construir un modelo de objetos interno y validarán el contenido de los datos. El resultado es un

funcionamiento muy eficiente de las aplicaciones que trabajan utilizando data binding.

El binding real entre una instancia XML y su modelo de objetos Java puede automatizarse haciendo uso de alguna de las herramientas de data binding disponibles, las cuales incluyen un conjunto de aplicaciones que permiten un fácil desarrollo de soluciones basadas en data binding.

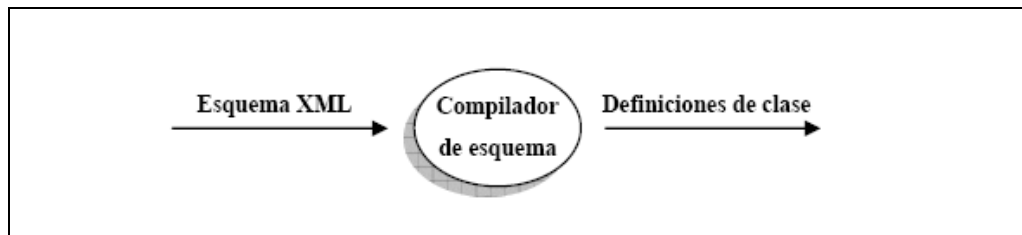


Figura 2.4. Generación de clases mediante data binding [DBINDING]

Herramientas de Data binding

Actualmente existe una gran variedad de herramientas de data binding, como Castor, JAXB, JBind, Quick o Zeus, entre las cuales elegir a la hora de realizar data binding entre XML y Java. Debido a que en cada una se han seguido diferentes enfoques de diseño, presentan capacidades de binding dispares.

Aún más, el soporte que ofrecen para la personalización del data binding varía significativamente de unas a otras.

Para la elección de la herramienta de data binding que se utilizará durante el desarrollo de la herramienta EACS Project Manager se optó por comparar dos tipos de herramientas: Castor y JAXB, dos herramientas desarrolladas por Exolab y Sun, respectivamente.

Castor

Castor de ExoLab es una herramienta de data binding de código abierto para Java que soporta binding con XML, bases de datos relacionales y LDAP. A nosotros sólo nos interesarán sus capacidades de data binding con XML, que se agrupan en la herramienta denominada Castor XML. Castor XML trabaja con los datos definidos en documentos XML a través de un modelo de objetos propietario (Castor Schema Object Model) que los representa. Está diseñado para trabajar con modelos de objetos Java ya existentes y para generar nuevos modelos basados en un esquema

XML origen, proporcionando un binding entre componentes de XML Schema y construcciones propias del lenguaje Java [CASTOR].

Castor mapea la mayor parte de los tipos primitivos de XML Schema en tipos primitivos y/o nativos de Java. Los tipos simples que son restricciones de otros tipos simples son mapeados al tipo Java más cercano y se genera código adicional de validación que será llamado durante las operaciones de marshalling y unmarshalling.

En cuanto a tipos complejos y elementos, Castor ofrece dos posibilidades para mapearlos a clases Java: el método “element” y el método “type”. El método “element” es el utilizado en Castor por defecto, y mapea todas las definiciones de nivel superior de tipos complejos así como cualquier elemento de tipo complejo de nivel superior a clases Java. El método “type”, por el contrario, no genera clases Java para declaraciones de elementos cuyo tipo es un tipo complejo de nivel superior.

Como ya se ha mencionado, el Source Code Generator crea el código Java que representa el modelo de objetos de un esquema XML. Más específicamente, es llamado por línea de comandos recibiendo un esquema XML como entrada y puede ser invocado utilizando un pequeño conjunto de opciones pasadas como parámetros. A partir de un esquema XML origen el Source Code Generator de Castor genera directamente dos tipos de clases: clases de implementación y sus descriptores de clase asociados. Las clases de implementación son JavaBeans [JAVABEANS] que incluyen también métodos marshal y unmarshal. Los descriptores de clase son clases que guardan la información de binding y de validación de su clase asociada y que se utilizan en las operaciones de marshalling y unmarshalling [BINDING_XML].

JAXB

JAXB (Java Architecture for XML Binding) de Sun [JAXB] es una herramienta de data binding que proporciona una API y un variado grupo de herramientas para automatizar el mapeo entre documentos XML y objetos Java, facilitando la tarea de los desarrolladores de incorporar datos XML y funciones para su procesamiento en aplicaciones Java. Partiendo de esquemas XML, JAXB genera clases Java que incorporan métodos para realizar el unmarshalling de documentos XML

convirtiéndolos en un árbol de objetos Java, así como para realizar el marshalling de dicho tipo de árboles de vuelta a un documento XML.

Los componentes principales de JAXB son:

- Un compilador que realiza el binding entre un esquema XML origen y un conjunto de clases Java.
- Una herramienta de ejecución de binding que proporciona las interfaces para acceder a las operaciones de marshalling, unmarshalling y validación que permiten manipular documentos XML y objetos Java.
- Un lenguaje basado en XML que describe el binding de un esquema XML a una representación en Java y que sirve para personalizar el proceso de data binding.

JAXB mapea la mayor parte de los tipos primitivos de XML Schema a tipos primitivos y nativos de Java de manera análoga a Castor. Los tipos simples derivados por restricción también son mapeados de modo similar a como sucede en Castor.

Donde JAXB se diferencia de Castor es en la manera en que se mapean tipos complejos y elementos, pues los transforma en interfaces Java.

La generación de código Java a partir de un esquema XML se lleva a cabo en JAXB por medio del compilador en línea de comandos que recibe un esquema XML y produce ficheros Java en un paquete previamente especificado. Los componentes definidos en un esquema XML se convierten, mediante el compilador, en interfaces de contenido, interfaces de elemento y clases de implementación para ambos conjuntos de interfaces. De todos éstos las interfaces de contenido son las más interesantes, pues definen métodos de acceso get y set para los datos contenidos en definiciones no anónimas de tipos complejos. Además de las interfaces y sus correspondientes clases de implementación, el compilador de JAXB genera una clase más, la clase factory de elementos, que contiene métodos para construir nuevas instancias de cada interfaz de contenido Java y de cada interfaz de elemento Java.

Debido a que cada uno de ellos presenta ventajas y desventajas en su uso, se optó por escoger aquella herramienta que se ajustara más a las necesidades de la aplicación.

Entre las características a tomar en cuenta para la elección de la herramienta de encuentran:

- Simplicidad en la construcción y modificación de estructuras de datos: la utilización de interfaces en JAXB, que por una parte hace que el código generado sea muy fácil de utilizar con documentos XML existentes, resulta en cambio algo incómoda en cuanto respecta a la construcción y modificación de estructuras de datos. Castor, al no emplear interfaces, resulta algo más sencillo de usar para crear o modificar estructuras de datos de documento: se evita la necesidad de pasar por una clase factory haciendo posible directamente el uso de un constructor para la clase apropiada.
- Poder expresivo del lenguaje de personalización: tanto Castor como JAXB siguen enfoques similares para personalizar el binding entre un esquema XML y un conjunto de clases Java. El lenguaje de personalización de Castor resulta intuitivo y fácil de aprender y el lenguaje de personalización de JAXB se demuestra superior, soportando herencia y redefinición de los valores de personalización. Sin embargo su potente expresividad tiene como contrapartida una intrincada definición de alcances para los valores de personalización definidos dentro de declaraciones de binding.

Por todo ello, la mejor opción es utilizar Castor como herramienta de mapeo ya que, como se mencionó anteriormente, ofrece sencillez, seguridad de la información y permite mapear la información sin inconvenientes y dado que para la implementación de la herramienta EACS Project Manager no se ha considerado el uso de herencia es conveniente el uso de este.

2.1.7. Servidor de Aplicación y Web

En informática se denomina servidor de aplicaciones a un equipo que forma parte de una red de computadores, en el cual se ejecuta una o más aplicaciones para ser utilizadas por los clientes definidos dentro de la red. Como consecuencia del éxito del lenguaje de programación Java, el término servidor de aplicaciones usualmente hace referencia a un servidor de aplicaciones J2EE [SERV_APLIC].

El servidor de aplicaciones ejecuta los programas de negocio en lugar del cliente (un navegador de Internet), del servidor Web o sistemas finales. Físicamente separa la lógica del negocio del cliente y los datos dentro de una arquitectura multicapa. Por ello, permite desarrollar y desplegar aplicaciones rápida y fácilmente

e incrementar la cantidad de usuarios sin perder la calidad de servicio y sin hacer cambios en la programación.

Un servidor Web es el responsable de aceptar las peticiones de tipo HTTP (diseñado para transferir hipertextos, páginas Web o páginas HTML). Dicho programa se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que responde a estas peticiones adecuadamente mediante una página Web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error [SERV_WEB].

Instalar un servidor Web propio permitiría, entre otras cosas, poder montar la propia página Web sin necesidad de contratar hosting (servicio de alojamiento Web). Entre los servidores de aplicaciones y servidores Web de código libre más conocidos se encuentran:

JBoss

Es un servidor de aplicaciones J2EE de código abierto certificado para J2EE 1.4 disponible en el mercado. Al estar basado en lenguaje Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte.

Entre sus características se destaca que tiene una flexibilidad consistente, ofrece servicios de middleware para cualquier objeto de Java y está orientado a una arquitectura de servicios y con una licencia de código abierto JBoss puede ser descargado, utilizado, incrustado y distribuido sin restricción alguna [JBoss].

Apache Tomcat

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation.

Tomcat es la implementación de referencia para las Java Server Pages (JSP) y las especificaciones Java Servlet. Esto significa que es el servidor Java disponible que más se ajusta a los estándares establecidos para la distribución de aplicaciones basadas en JSP [TOMCAT].

Existe una diferencia entre estos servidores, mientras que JBoss es un servidor de aplicaciones que puede manejar EJB, Tomcat es un servidor Web que sirve a su vez como contenedor de servlets y JSP.

Considerando el entorno de desarrollo elegido (NetBeans), los conceptos mostrados anteriormente y dado que para la implementación de la herramienta no se ha considerado el uso de EJB, sino solamente el uso de JSP y servlets, se optó por usar Tomcat como servidor Web y contenedor de estos dos últimos elementos.

Otro motivo por el cual se decidió usar Tomcat en lugar de JBoss es la diferencia de tiempos de inicialización requeridos por estos servidores al realizar algún cambio durante la etapa de desarrollo.

2.2. Mapeo de datos XML a clases.

Para poder convertir una clase o un conjunto de clases relacionadas debemos mapearlas dentro de un archivo XML que luego deberá ser cargado en el contexto de ejecución de Castor, dado que se modelaran los datos en documentos XML, se presenta a continuación su descripción.

XML

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. XML es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones [XML].

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Entre las principales ventajas encontramos:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles.
- Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.
- XML simplifica los cambios de plataforma, la actualización a los nuevos sistemas (software o hardware), siempre consume mucho tiempo. Grandes cantidades de datos se deben convertir y se pierden muchas veces los datos incompatibles.
- XML se almacena en formato de texto. Esto hace que sea más fácil de ampliar o actualizar a los nuevos sistemas operativos, nuevas aplicaciones o nuevos navegadores, sin perder datos.

Estructura de un documento XML

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información. Estas partes se llaman elementos, y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma <nombre>, donde nombre es el nombre del elemento que se está señalando.

A continuación en el Código 2.1. se muestra un documento XML de la herramienta EACS Project Manager para entender su estructura.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <GestorTipos>
  - <GenTipos IdGrupo="1">
    <Descripcion>ESTADOS DEL PROYECTO</Descripcion>
    <Orden>0</Orden>
  </GenTipos>
  - <GenTipos IdGrupo="1">
    <Descripcion>Pendiente</Descripcion>
    <Orden>1</Orden>
  </GenTipos>
  - <GenTipos IdGrupo="1">
    <Descripcion>en Progreso</Descripcion>
    <Orden>2</Orden>
  </GenTipos>
  - <GenTipos IdGrupo="2">
    <Descripcion>TIPOS DE ACTIVIDADES</Descripcion>
    <Orden>0</Orden>
  </GenTipos>
  - <GenTipos IdGrupo="2">
    <Descripcion>Documentacion</Descripcion>
    <Orden>1</Orden>
  </GenTipos>
</GestorTipos>
  
```

Código 2.1. Archivo XML de la herramienta EACS.

La primera línea es la declaración XML. Se define la versión de XML (1,0) y la codificación utilizada (UTF-8). La siguiente línea describe el elemento raíz del documento, como se muestra en el Código 2.2.

```
<GestorTipos>
```

Código 2.2. Elemento raíz del documento.

Las próximas líneas describen los elementos que contiene la raíz, todos los elementos <GenTipos> en el documento están contenidos dentro de <GestorTipos>. El elemento <GenTipos> tiene tres atributos: <IdGrupo>, <Descripcion>, <Orden>, como se muestra en el Código 2.3.

```
<GenTipos IdGrupo="1">
  <Descripcion>ESTADOS DEL PROYECTO</Descripcion>
  <Orden>0</Orden>
</GenTipos>
```

Código 2.3. Elementos que contiene la raíz.

Y finalmente la última línea define el final del elemento raíz:

```
</GestorTipos>
```

Código 2.4. Final del elemento raíz.

La diferencia entre XML y HTML

XML no es un sustituto de HTML. XML y HTML fueron diseñados con objetivos diferentes:

- XML fue diseñado para transportar y almacenar datos, con especial atención en los datos.
- HTML fue diseñado para mostrar los datos, con especial atención a cómo los datos se ven.

HTML es acerca de cómo mostrar la información, mientras que XML es para llevar consigo la información.

- XML no es un reemplazo para el HTML.
- XML es un complemento de HTML.

En la mayoría de aplicaciones Web, XML se utiliza para el transporte de datos, mientras que HTML se utiliza para formatear y mostrar los datos.

XML se utiliza para crear nuevos lenguajes de Internet

Una gran cantidad de lenguajes de Internet son creados con XML. He aquí algunos ejemplos:

- XHTML
- WSDL para describir los servicios Web disponibles.

- WAP y WML como lenguajes de marcado para dispositivos de mano.
- RDF y OWL para la descripción de los recursos.
- SMIL para describir multimedia para la Web.

Bases de Datos XML

El uso principal de XML es estructurar datos, recibirlos y/o enviarlos, pero también podemos guardar datos en nuestros documentos para que sean tratados luego por cualquier lenguaje.

En general, una Base de Datos XML es una Base de Datos almacenada o gestionada en forma de documentos XML. En una Base de Datos XML es posible caracterizar tres tipos de archivos XML:

- Centrados en Datos.
- Centrados en Documentos.
- Híbridos (mezcla partes de los dos tipos anteriores, BD XML nativa).

Entre las principales características del tipo centrados en datos encontramos:

- Muchos elementos de datos.
- Con estructura regular y bien definida.
- Datos muy estructurados o semi-estructurados.
- Usados como mecanismo de intercambio o para mostrar datos en la Web.
- Dirigidos a utilización automática (por máquinas).

Entre las principales características del tipo centrados en documentos encontramos:

- Pocos elementos.
- Con grandes cantidades de texto.
- Con estructuras impredecibles en tamaño y contenido.
- Datos poco estructurados.
- Orientados a ser interpretadas por humanos.
- Enfocados a sistemas documentales y de gestión de contenidos.

De modo que una base de datos XML es aquella que define un modelo lógico de un documento XML además almacena y recupera documentos de acuerdo a ese modelo. En el presente trabajo de tesis hacemos uso de archivos XML centrados en

los datos ya que tenemos una estructura bien definida y los datos actualizables son usados en maneras diversas en la herramienta.

Proceso del mapeo de datos.

Para realizar el mapeo de datos debemos tener en cuenta lo siguiente:

- Cada tipo de elemento XML de más alto nivel se modela como una clase de objetos.
- Los atributos XML se modelan como atributos de las clases.

A continuación en el Código 2.5. se muestra un ejemplo de una de las clases de la herramienta EACS Project Manager que utilizaremos para convertirlas a XML.

El documento XML y la clase java BGenTipos pueden ser conectados usando el archivo de mapeo que se muestra en el Código 2.6.

Dado que tenemos clases gestoras que contienen relaciones de clases, tenemos que adicionalmente realizar el mapeo de colecciones de objetos para lo cual agregamos un tag <collection> que contiene el tipo de colección que manipularemos.

```

public class BGenTipos implements Serializable {

    private String strIdGrupo= "";
    private String strDescripcion= "";
    private String strOrden= "";

    public BGenTipos() {
    }

    public String getStrIdGrupo() {
        return strIdGrupo;
    }

    public void setStrIdGrupo(String IdGrupo) {
        this.strIdGrupo = IdGrupo;
    }

    public String getStrDescripcion() {
        return strDescripcion;
    }

    public void setStrDescripcion(String Descripcion) {
        this.strDescripcion = Descripcion;
    }

    public String getStrOrden() {
        return strOrden;
    }

    public void setStrOrden(String Orden) {
        this.strOrden = Orden;
    }

}

```

Código 2.5. Clase BGenTipos.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <mapping>
- <class name="beans.BGenTipos">
  <map-to xml="GenTipos" />
  - <field name="strIdGrupo" type="java.lang.String" get-method="getStrIdGrupo"
    set-method="setStrIdGrupo">
    <bind-xml name="IdGrupo" node="attribute" />
  </field>
  - <field name="strDescripcion" type="java.lang.String" get-
    method="getStrDescripcion" set-method="setStrDescripcion">
    <bind-xml name="Descripcion" node="element" />
  </field>
  - <field name="strOrden" type="java.lang.String" get-method="getStrOrden"
    set-method="setStrOrden">
    <bind-xml name="Orden" node="element" />
  </field>
</class>
</mapping>

```

Código 2.6. Archivo de Mapeo para la clase BGenTipos.

El archivo xml que obtenemos como resultado del mapeo es como el que se muestra en el Código 2.7.

En el Código 2.6 tenemos el tag <mapping>, es el elemento raíz del archivo a mapear, la clase principal es beans.BGenTipos y esta clase en el mapeo posee un tag <map-to>, este tag determina el nombre del tag raíz del xml resultante, también se debe notar que los atributos type de cada tag <field> determinan el tipo de objeto que será creado (que puede ser un tipo primitivo de java o uno propio) y el atributo name establece el método accesor que será buscado en la clase (no el nombre del atributo), el tag <bind-xml> (hijo de field) determina el nombre que en el xml poseerá el atributo correspondiente a la clase.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <GestorTipos>
- <GenTipos>
  <IdGrupo />
  <Descripcion />
  <Orden />
</GenTipos>
</GestorTipos>

```

Código 2.7. Archivo XML obtenido del mapeo de la clase BGenTipos.

El orden de los tags en el xml obtenido corresponde al orden establecido en el mapeo, es decir, si necesitamos un orden diferente de los tags en el xml no tenemos que modificar nada más que el mapeo de nuestras clases al igual que si queremos que cambiar el nombre a algún tag, solo hay que modificar el mapping.

2.3. Implementación de Componentes

A continuación se presentan algunas de las principales consideraciones que se tomaron en cuenta durante la etapa de construcción de la herramienta EACS Project Manager.

Para la construcción, la herramienta ha sido dividida por componentes, de forma que no ha sido necesaria una mayor gestión en la configuración del código fuente, el equipo de trabajo ha realizado la unificación de los módulos y componentes en conjunto.

Además debido a la estructura de la herramienta, la integración ha sido realizado de forma muy sencilla debido a que cada módulo funciona casi de forma independiente y sólo es necesario integrarse por medio del archivo de configuración faces-config.xml (colocar los beans para el manejo de las páginas y navegabilidad entre ellas) y colocar los paquetes con los beans, archivos de controlador y las páginas en sí de los módulos a integrar.

2.3.1. Estructuración Interna del Proyecto Web

En la presente sección se revisará la estructura de archivos del proyecto, las carpetas y el contenido de cada una de ellas.

Dado que la herramienta EACS Project Manager fue elaborada dentro del entorno de desarrollo NetBeans, el tipo de proyecto que se utilizó fue el denominado Web Application con soporte para Visual Java Server Faces.

En la Figura 2.5. se muestra la estructura interna del proyecto, de las cuales nos ocuparemos de las carpetas: páginas Web, archivos de configuración, paquetes fuente y librerías. Las demás carpetas no han sido utilizadas en el presente trabajo pero permanecen ya que son un estándar para la herramienta con la que se ha desarrollado el sistema.

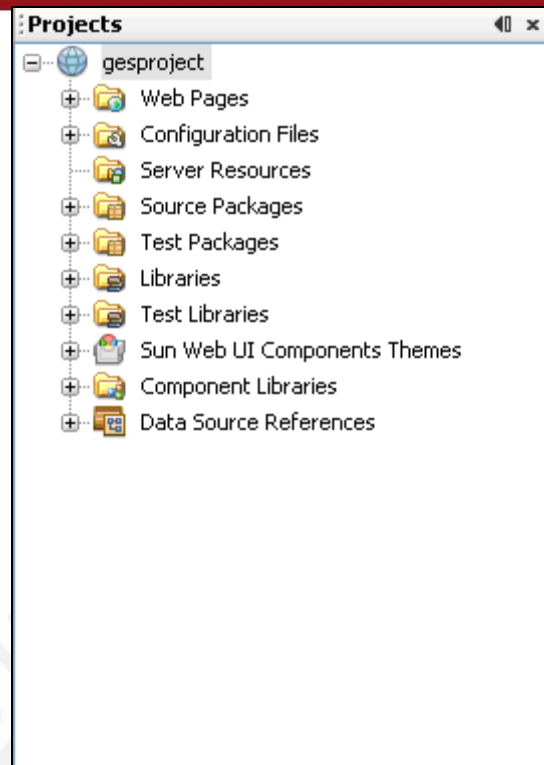


Figura 2.5. Estructura de archivos del proyecto

Páginas Web

En la Figura 2.6. se muestra la estructura interna de la carpeta páginas Web, la cual contiene todos los archivos correspondientes a la capa de interacción con el usuario, estas comprenden páginas JSP, JSPF (fragmentos de páginas Web), archivos de configuración para el proyecto Web (generalmente archivos XML), archivos CSS (hojas de estilo), archivos JS (javascript) e imágenes JPG, PNG o GIF.

Las primeras carpetas: META-INF y WEB-INF contienen archivos de configuración para el proyecto Web estos son generalmente archivos XML que contienen la configuración de los beans, páginas y las librerías utilizadas en el proyecto.

Las carpetas: actividades, actividadesxusuarios, actividadesxhoras, artefactos, empresas, proyectos, proyectoxusuarios, reportes, seguridad y usuarios contienen las páginas Web Java Server Faces agrupadas en carpetas según su funcionalidad.

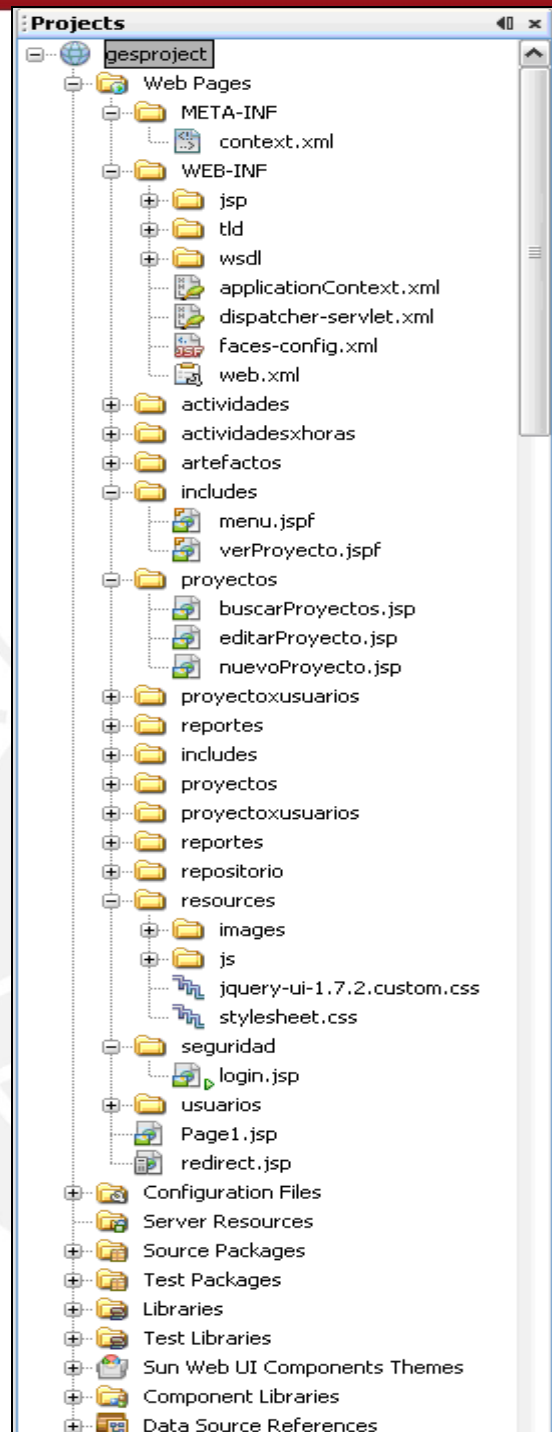


Figura 2.6. Carpeta Páginas Web

Estas tienen extensión .jsp como cualquier página jsp, sin embargo como puede apreciarse en el Código 2.8 internamente contienen un formato similar a un XML utilizando las etiquetas que proporciona Java Server Faces, esto permite agregar controles enriquecidos con propiedades personalizadas como grillas, botones, etc y no simples controles de formularios HTML.

```

<jsp:root version="1.2" xmlns:f="http://java.sun.com/jspf/core" xmlns:h="http://java.sun.
<jsp:directive.page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"/>
<f:view>
  <ui:page id="page1">
    <ui:html id="html1">
      <ui:head id="head1" title="Ingreso al Sistema Gesproject">
        <ui:link id="link1" url="/resources/stylesheet.css"/>
      </ui:head>
      <ui:body id="Login" style="-rave-layout: grid">
        <ui:form id="formLogin">
          <h:panelGrid id="gridPanelLogin">
            <h:panelGrid columns="2" id="gridPanelLogin">
              <ui:image id="imageLogin"/>
              <h:panelGrid id="gridPanelLogin">
                <ui:staticText id="staticText1" text="Ingreso al Sistema"/>
                <h:panelGrid columns="2" id="gridPanelLogin">
                  <ui:staticText id="lblUsuario" text="Usuario"/>
                  <ui:textField id="txtUsuario"
                    text="#{seguridad$login.strUsuario}" />
                  <ui:staticText id="lblPassword" text="Password"/>
                  <ui:passwordField id="txtPassword"
                    text="#{seguridad$login.strUsuario}" />
                  <ui:staticText id="lblPassword" text="Password"/>
                  <ui:passwordField id="txtPassword"
                    password="#{seguridad$login.strPassword}" />
                </h:panelGrid>
              </h:panelGrid>
              <h:panelGrid id="gridPanelLogin" >
                <ui:button action="#{seguridad$login.btnIngresar_action}"
                  id="btnIngresar" text="Ingresar"/>
                <ui:staticText id="lblMensaje"
                  text="#{seguridad$login.strMensaje}" />
              </h:panelGrid>
            </h:panelGrid>
          </h:panelGrid>
        </ui:form>
      </ui:body>
    </ui:html>
  </ui:page>
</f:view>
</jsp:root>

```

Código 2.8. Código de una página Java Server Faces.

La carpeta includes contiene archivos con extensión .jspf (Java Server Pages Fragment) estos son fragmentos de páginas, generalmente para las zonas comunes como menús, cabeceras y pies de pagina, las cuales deben ser incluidas en todas las demás páginas; de esta forma ahorramos código y facilita el mantenimiento de las páginas, ya que las modificaciones en estas secciones se realizarán en un solo lugar.

La carpeta resources contiene las hojas de estilo (archivos CSS), javascripts (archivos JS) e imágenes utilizados en el sistema. Cabe resaltar entre estos archivos aquellos relacionados a la librería javascript JQuery:

- jquery-1.3.2.min.js: esta librería proporciona una serie de funciones que ayudan a escribir código javascript reducido.

- `jquery-ui-1.7.2.custom.min.js`: esta librería extiende la anterior y proporciona además funciones javascript para implementar controles tales como ventanas emergentes (popups), pestañas, calendarios, acordeones y diversos efectos.
- `validaciones.js`: este archivo propio contiene funciones javascript diversas que permiten realizar validaciones de los formularios.

Archivos de configuración

La carpeta correspondiente a los archivos de configuración es en realidad una carpeta virtual que contiene los mismos archivos vistos en las carpetas WEB-INF y META-INF de la sección anterior de páginas Web.

Esta contiene los siguientes archivos:

- **Context.xml**: el nombre del contexto del proyecto y configuración de las conexiones a las bases de datos.
- **Web.xml**: configuración de los JARs (librerías utilizadas) y TLDs (librería de etiquetas para JSF, Spring, etc).
- **Faces-config.xml**: configuración de los beans para las páginas Java Server Faces y la navegación entre ellas.

Paquetes fuente

En la Figura 2.7. se muestra la estructura interna de los paquetes utilizados en el sistema, la cual contiene todos los archivos fuentes correspondientes a la capa de entidades, negocio (beans y clases java), datos (archivos XML) y controlador (clases de acción java).

A continuación se explica en detalle la función y el contenido de cada paquete:

- **beans**: este contiene todas las clases java de las entidades del sistema así como sus correspondientes gestores que los administran.
- **gesproject**: este paquete es el creado por defecto por Netbeans con el nombre con que se creó inicialmente el proyecto. En la raíz de este paquete se encontrarán los beans de request, sesión y aplicación, estos servirán como su nombre lo indica para almacenar objetos en los tres ámbitos.
- **Configuración de rutas de archivos XML y XPD**: En el paquete `gesproject` también se almacena un archivo de configuración con extensión `.properties`. Este archivo es bastante útil para almacenar datos constantes esto sirve para configurar la instalación de la aplicación en cualquier servidor, de esta forma se

puede configurar las rutas donde se encuentren los archivos XML de mapping y los XML y XPDL con la data de las metodologías y de los proyectos, también se ha guardado la ruta donde se encuentra la raíz del repositorio.

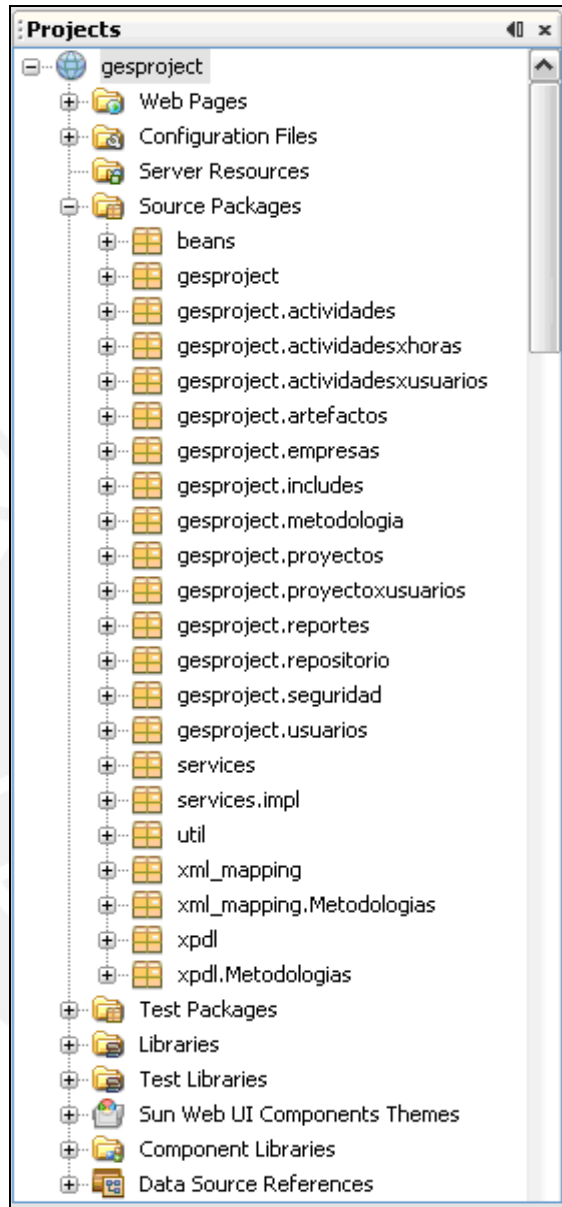


Figura 2.7. Estructura de paquetes

- **gesproject.paquete:** dentro del paquete gesproject también se encuentran una serie de sub-paquetes que contienen a las clases de la capa controlador del sistema. Esta estructura de carpetas de los sub-paquetes corresponde idénticamente a la estructura de carpetas de las páginas Web, estrictamente a las carpetas que contienen archivos JSP, esto debido a que el marco JSF

trabaja de esta forma para relacionar los JSP con sus correspondientes controladores.

De esta forma por cada página jsp también se crea un paquete con el nombre de la carpeta del jsp y dentro una clase con el mismo nombre de la página. Esta clase java corresponde con el controlador de la página y como vemos en el Código 2.9. esta extiende la clase `AbstractPageBean` la cual le proporciona métodos básicos para inicializar las variables o controles utilizados en la página.

```

package gesproject.seguridad;

import com.sun.rave.web.ui.appbase.AbstractPageBean;
import gesproject.RequestBean1;
import gesproject.SessionBean1;
import gesproject.ApplicationBean1;
import beans.BUusuario;

/**
 * <p>Modulo Seguridad</p>
 **/

public class login extends AbstractPageBean {

}

```

Código 2.9. Código de una clase controlador Java Server Faces.

- **services:** este paquete corresponde a la capa de servicios y contiene todos los servicios y sus correspondientes métodos para trabajar con las entidades. Esta capa contiene interfaces, por lo que sólo se encuentran las definiciones de los métodos a utilizar.
- **services.impl:** en el sub-paquete impl se encuentran todas las clases que implementan los servicios del paquete padre services.
- **util:** este paquete contiene clases de apoyo con diversas funciones que se utilizan en todo el proyecto pero que no pertenecen a una entidad en específico. Una clase importante en este paquete es `MotorXML`, esta clase contiene los métodos leer y escribir que permite al sistema obtener y registrar la data en los archivos xpdL por medio de un mapeo y asociación con las entidades.
- **xml_mapping:** en este paquete se encuentran almacenados los archivos xml que sirven para mapear los xpdL con sus correspondientes entidades. Los archivos XML para el mapeo de datos puede también configurarse en otra ruta,

por motivos de seguridad o respaldo, esta nueva ruta debe colocarse en el archivo .properties del paquete gesproject.

- **xpdl:** este paquete contiene los archivos xpdl los cuales almacenan los datos de las entidades. Los archivos XPD y XML con los datos de las metodologías y los proyectos puede también configurarse en otra ruta, por motivos de seguridad o respaldo, esta nueva ruta debe colocarse en el archivo .properties del paquete gesproject.

Librerías

En la Figura 2.8. se muestra el listado de librerías utilizadas en el sistema, estas corresponden a las relacionadas a Java Server Faces, tanto como controlador y para las etiquetas para la capa de vista. También se han agregado las librerías correspondientes al marco de trabajo Spring.

Adicionalmente se ha agregado las librerías para manejar la capa de datos, para esto se ha utilizado castor, lo cual nos permite realizar el mapeo entre la data en los xpdl y las entidades.

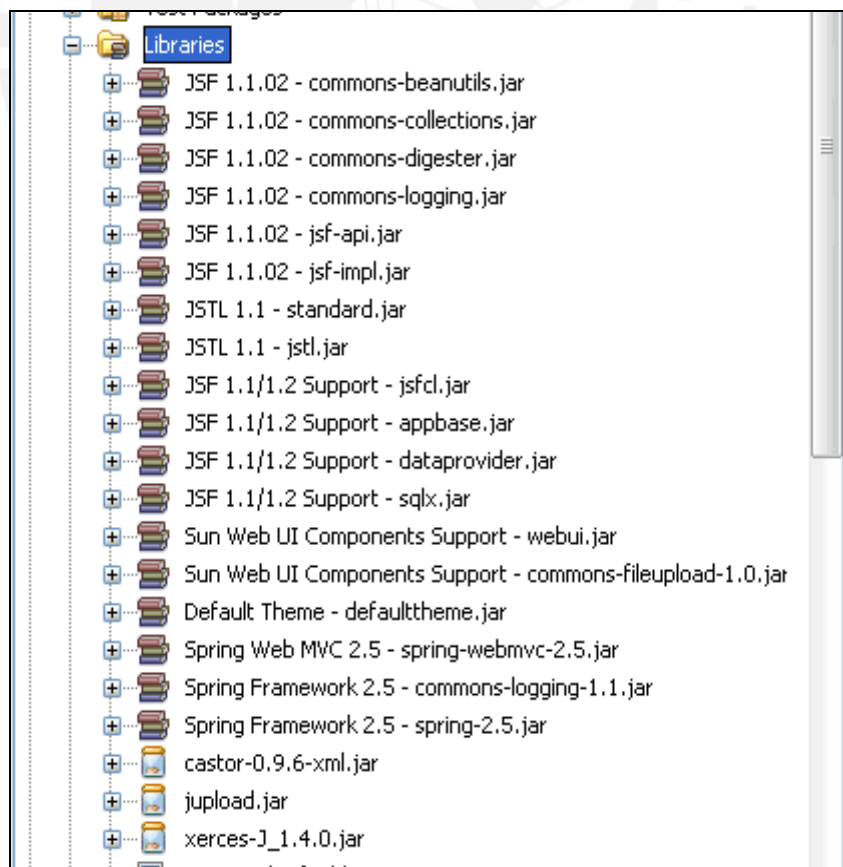


Figura 2.8. Librerías utilizadas en el proyecto.

2.3.2. Configuración de Spring

Una de las características principales de Spring es que permite configurar y crear aplicaciones complejas a partir de un conjunto de componentes sencillos que son declarados en archivos xml.

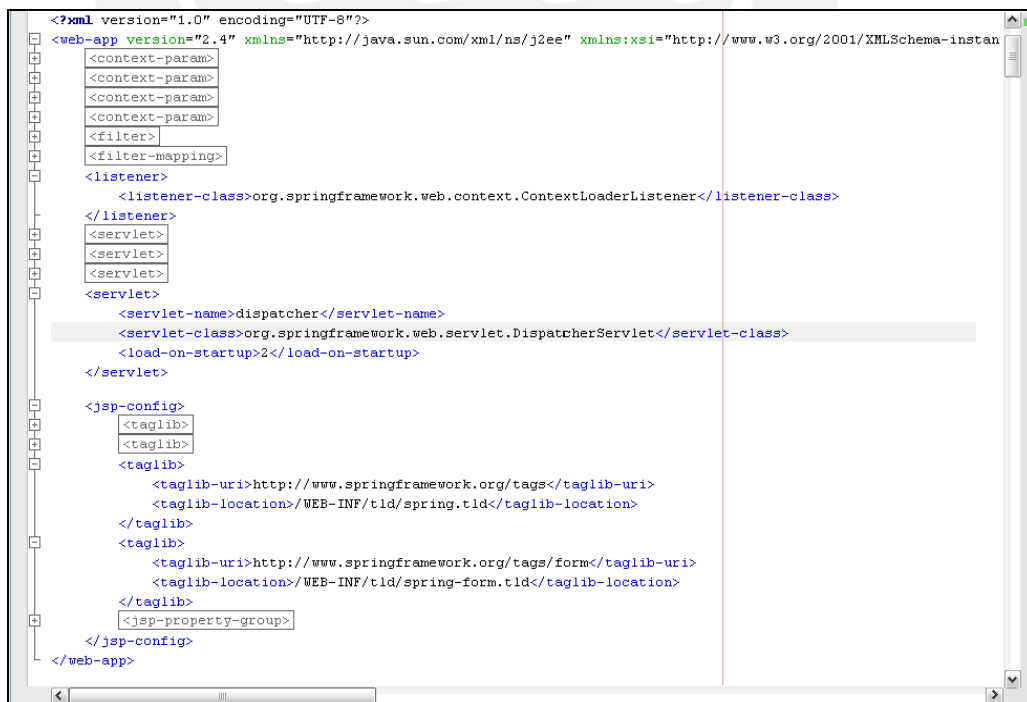
A continuación se procederá a explicar cuáles fueron las configuraciones realizadas en la herramienta EACS Project Manager para el manejo de las clases JavaBean, Service y Controller a través del marco de trabajo Spring.

Instalación de Spring

Para empezar a utilizar Spring es necesario agregar en el proyecto Web su respectivo JAR, el cual puede ser descargado de forma gratuita a través de la página Web de dicho marco de trabajo. Este JAR debe ser agregado en la carpeta librerías pertenecientes al proyecto creado en el IDE de NetBeans.

Configuración del archivo web.xml

En el Código 2.10. se muestra las líneas que se han agregado a este archivo de configuración para trabajar con Spring, básicamente se ha agregado el listener y el servlet que serán los encargados de recibir y procesar los llamados a Spring.



```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <context-param>
  <context-param>
  <context-param>
  <context-param>
  <filter>
  <filter-mapping>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <servlet>
  <servlet>
  <servlet>
  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>2</load-on-startup>
  </servlet>
  <jsp-config>
    <taglib>
    <taglib>
    <taglib>
      <taglib-uri>http://www.springframework.org/tags</taglib-uri>
      <taglib-location>/WEB-INF/tld/spring.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>http://www.springframework.org/tags/form</taglib-uri>
      <taglib-location>/WEB-INF/tld/spring-form.tld</taglib-location>
    </taglib>
    <jsp-property-group>
  </jsp-config>
</web-app>

```

Código 2.10. Configuración SPRING en web.xml.

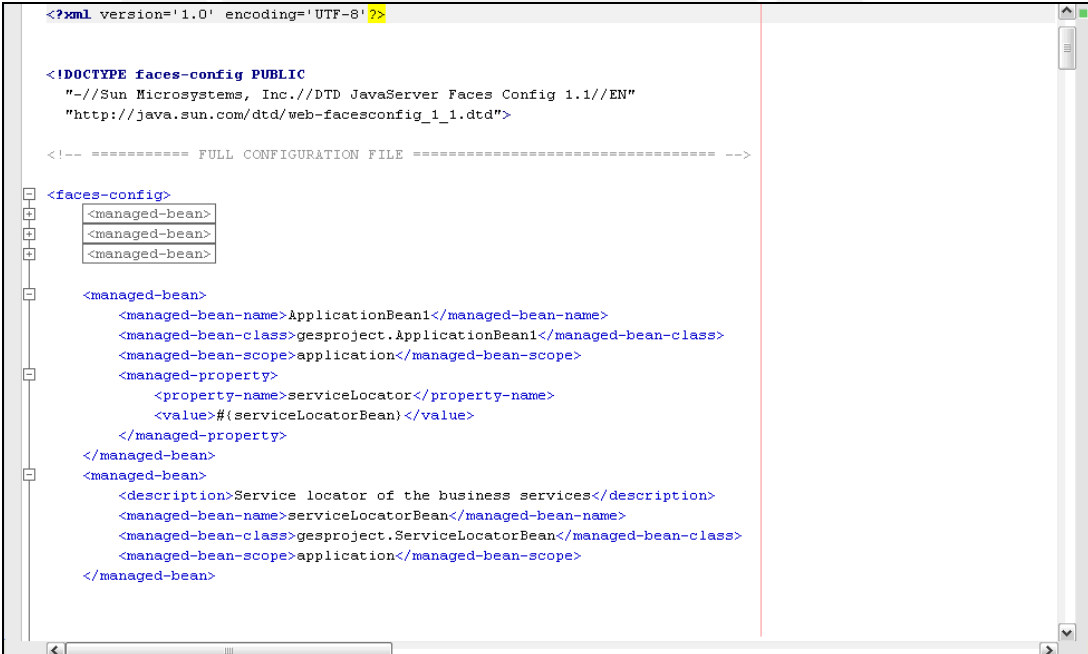
Configuración del archivo faces-config.xml

Debido a que tenemos un proyecto que utiliza dos marcos de trabajo Spring y JSF, debemos manipular el archivo de configuración propio de JSF para que pueda utilizar la persistencia de objetos que nos brinda Spring.

En el Código 2.11. se muestra las líneas que hemos modificado en el archivo de configuración faces-config.xml, para lograr este objetivo se ha agregado una nueva propiedad para el bean de Aplicación llamada serviceLocator.

Así mismo se ha agregado líneas abajo la referencia de este nuevo bean y la ubicación de su clase en el proyecto y el ámbito en donde debe trabajar, en este caso a nivel de aplicación.

Esta configuración nos permite utilizar este nuevo bean a nivel de toda la aplicación de forma que permanecerá disponible por los servicios que lo requieren desde cualquier parte del sistema. A partir de este único bean podremos incluir atributos que instancien a las clases que nos servirán para manejar los proyectos, actividades y artefactos.



```

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE faces-config PUBLIC
"-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.1//EN"
"http://java.sun.com/dtd/web-facesconfig_1_1.dtd">

<!-- ===== FULL CONFIGURATION FILE ===== -->

<faces-config>
  <managed-bean>
  <managed-bean>
  <managed-bean>

  <managed-bean>
    <managed-bean-name>ApplicationBean1</managed-bean-name>
    <managed-bean-class>gesproject.ApplicationBean1</managed-bean-class>
    <managed-bean-scope>application</managed-bean-scope>
    <managed-property>
      <property-name>serviceLocator</property-name>
      <value>#{serviceLocatorBean}</value>
    </managed-property>
  </managed-bean>
  <managed-bean>
    <description>Service locator of the business services</description>
    <managed-bean-name>serviceLocatorBean</managed-bean-name>
    <managed-bean-class>gesproject.ServiceLocatorBean</managed-bean-class>
    <managed-bean-scope>application</managed-bean-scope>
  </managed-bean>

```

Código 2.11. Configuración SPRING en faces-config.xml.

Configuración del archivo applicationContext.xml

Este archivo nos ayuda a definir los objetos que serán utilizados a nivel del bean de aplicación. En el Código 2.12. se visualiza los beans de los servicios que serán llamados desde el bean serviceLocator definido anteriormente.

Por lo tanto los servicios serán instanciados al iniciar la aplicación, de esta forma en sus correspondientes constructores obtendremos la data de los xpdl y estos quedarán almacenados a nivel de la aplicación en la estructura de objetos correspondientes a sus entidades. De esta forma cuando se necesite un objeto en un formulario ya no será necesario obtenerlo nuevamente de los archivos xpdl sino simplemente se pedirá del bean de aplicación serviceLocator.



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/sp
Comment
<beans>
  <bean id="SProyecto" class="services.impl.SProyectoImpl"></bean>
  <bean id="SProyectoUsuarios" class="services.impl.SProyectoUsuariosImpl"></bean>
  <bean id="SActividad" class="services.impl.SActividadImpl"></bean>
  <bean id="SActividadUsuarios" class="services.impl.SActividadUsuariosImpl"></bean>
  <bean id="SGenTipos" class="services.impl.SGenTiposImpl"></bean>
  <bean id="SEmpresas" class="services.impl.SEmpresasImpl"></bean>
  <bean id="SMetodologia" class="services.impl.SMetodologiaImpl"></bean>
  <bean id="SUsuario" class="services.impl.SUsuarioImpl"></bean>
  <bean id="SArtefacto" class="services.impl.SArtefactoImpl"> </bean>
  <bean id="SCarpeta" class="services.impl.SCarpetaImpl"></bean>
</beans>
  
```

Código 2.12. Configuración SPRING en applicationContext.xml.

2.3.3. Configuración de Java Server Faces

En la presente sección se explicará la forma como se ha realizado la configuración del proyecto para que pueda utilizar el marco de trabajo Java Server Faces, sus archivos de configuración y librerías incluidas.

Instalación de Java Server Faces

Para empezar a utilizar JSF es necesario agregar en el proyecto Web su respectivo JAR. Este JAR debe ser agregado en la carpeta librerías pertenecientes al proyecto creado en el IDE de NetBeans.

Configuración del archivo web.xml

En el Código 2.13. se muestra las líneas que se han agregado a este archivo de configuración para trabajar con JSF, básicamente se ha agregado el servlet que será encargado de recibir y procesar los llamados a JSF, también se ha configurado el mapeo para que toda página llamada bajo el nombre “faces” se envíe al servlet controlador de JSF.

Líneas abajo también se ha agregado la librería para utilizar las etiquetas de JSF en las páginas JSP y sus formularios.



```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <context-param>
  </context-param>
  <context-param>
  </context-param>
  <context-param>
  </context-param>
  <context-param>
  </context-param>
  <filter>
  </filter>
  <filter-mapping>
  </filter-mapping>
  <listener>
  </listener>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet>
  </servlet>
  <servlet>
  </servlet>
  <servlet>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>faces/seguridad/login.jsp</welcome-file>
  </welcome-file-list>
  <error-page>
  </error-page>
  <error-page>
  </error-page>
  <error-page>
  </error-page>
  <error-page>
  </error-page>
  <jsp-config>
    <taglib>
      <taglib-uri>http://java.sun.com/jsp/jstl/core</taglib-uri>
      <taglib-location>/WEB-INF/tld/c.tld</taglib-location>
    </taglib>
    <taglib>
    </taglib>
    <taglib>
    </taglib>
    <taglib>
    </taglib>
    <jsp-property-group>
      <url-pattern>*.jspx</url-pattern>
      <is-xml>true</is-xml>
    </jsp-property-group>
  </jsp-config>
</web-app>

```

Código 2.13. Configuración JSF en web.xml.

Configuración del archivo faces-config.xml

Este es el archivo que contiene casi toda la configuración del marco de trabajo JSF, en el Código 2.14. se muestra las líneas donde se ha definido los beans de request, sesión y aplicación, estos como su nombre lo indica nos servirán para obtener objetos que necesitemos obtener o utilizar en cualquiera de los tres ámbitos.

- **request:** es el más conocido y utilizado por defecto por casi todos los marcos de trabajo, los datos permanecerán sólo en un envío.

- **sesion:** en este ámbito los datos permanecerán en la sesión del usuario que se encuentra utilizando la aplicación.
- **aplicación:** este ámbito es el que se ha utilizado para la persistencia de datos con Spring ya que los datos permanecen para toda la aplicación y por lo tanto disponible para todos los usuarios.

```

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE faces-config PUBLIC
    "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.1//EN"
    "http://java.sun.com/dtd/web-facesconfig_1_1.dtd">

<!-- ***** FULL CONFIGURATION FILE ***** -->

<faces-config>
  <managed-bean>
    <managed-bean-name>SessionBean1</managed-bean-name>
    <managed-bean-class>gesproject.SessionBean1</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>
  <managed-bean>
  </managed-bean>
  <managed-bean>
    <managed-bean-name>RequestBean1</managed-bean-name>
    <managed-bean-class>gesproject.RequestBean1</managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
  </managed-bean>

  <managed-bean>
    <managed-bean-name>ApplicationBean1</managed-bean-name>
    <managed-bean-class>gesproject.ApplicationBean1</managed-bean-class>
    <managed-bean-scope>application</managed-bean-scope>
    <managed-property>
      <property-name>serviceLocator</property-name>
      <value>#{serviceLocatorBean}</value>
    </managed-property>
  </managed-bean>
</faces-config>

```

Código 2.14. Configuración controladores JSF en faces-config.xml.

En el Código 2.15. se muestra la forma como se define las reglas de navegación por cada página JSP creada. Este archivo de configuración se utiliza para definir la navegación entre las páginas.

- **from-view-id:** corresponde al nombre de la página origen desde donde se quiere definir sus reglas de navegación.
- **navigation-case:** se crearán tantos tags de este tipo como reglas de navegación hacia otras páginas se desee desde la página origen.
- **from-outcome:** este campo es muy importante ya que define el nombre con el que se representará la regla. Cuando en el controlador de la página en un determinado evento se retorne este nombre en formato de cadena se estará llamando a esta regla de navegación en específico.
- **to-view-id:** este campo indicará la página destino hacia donde debe navegarse para la regla de navegación definida.

```

<navigation-rule>
<managed-bean>
  <managed-bean-name>proyectos$nuevoProyecto</managed-bean-name>
  <managed-bean-class>gesproject.proyectos.nuevoProyecto</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>proyectos$buscarProyectos</managed-bean-name>
  <managed-bean-class>gesproject.proyectos.buscarProyectos</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>proyectos$editarProyecto</managed-bean-name>
  <managed-bean-class>gesproject.proyectos.editarProyecto</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>includes$verProyecto</managed-bean-name>
  <managed-bean-class>gesproject.includes.verProyecto</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>actividades$buscarActividades</managed-bean-name>
  <managed-bean-class>gesproject.actividades.buscarActividades</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean>

```

Código 2.15. Configuración navegación JSF en faces-config.xml.

2.4. Administración de datos.

En esta sección se muestra la estructura de trabajo para la administración de datos que es utilizada para todas las estructuras y entidades del sistema.

2.4.1 ServiceLocatorBean

Esta clase inicia los servicios de todas las entidades con la que se trabaja en el sistema: usuarios, proyectos, artefactos, clientes. Este código se ejecutará cada vez que la aplicación sea iniciada en el servidor de aplicaciones.

En el Código 2.16. se muestra parte del código fuente del constructor de la clase donde se inicia cada servicio.

```

public ServiceLocatorBean() {
    ServletContext context = FacesUtils.getServletContext();
    this.appContext = WebApplicationContextUtils.getRequiredWebApplicationContext(context);
    this.ServiceAlumno=
(SAlumno)this.lookupService(Constants.ALUMNO_SERVICE_BEAN_NAME);
    this.ServiceProyecto=
(SProyecto)this.lookupService(Constants.PROYECTO_SERVICE_BEAN_NAME);
    this.ServiceProyectoUsuarios= (SProyectoUsuarios)this.lookupService
(Constants.PROYECTOUSUARIOS_SERVICE_BEAN_NAME);

```

Código 2.16. ServiceLocatorBean constructor.

2.4.2 Implementación de Servicios

En esta sección se muestra como se han implementado los servicios de las entidades. Como estos son llamados cuando el servidor inicia la aplicación, entonces en esta sección se obtienen los datos que se cargan en las estructuras de las entidades correspondientes.

En el Código 2.17. se muestra parte del código fuente del constructor de la clase, primero se obtienen las rutas donde se encuentran los archivos XML de configuración para el mapeo de los datos y el XPDL que contiene los datos almacenados previamente. Luego con las rutas se realiza el pedido a la clase MotorXML para que nos entregue la estructura de las entidades cargada con los datos del XPDL.

```
Public SUsuarioImpl() {
    gestorUsuario= new BGestorUsuario();
    this.strRutaMapping=
java.util.ResourceBundle.getBundle("gesproject.Bundle").getString("ruta.mapping");
    this.strRutaXpdl=
java.util.ResourceBundle.getBundle("gesproject.Bundle").getString("ruta.xpdl");

    try{
        gestorUsuario=
(BGestorUsuario)MotorXML.leer(this.strRutaMapping+"mappingGestorUsuario.xml",
this.strRutaXpdl+"XmlListaUsuarios.xml");
    }

    catch(Exception ex){
        System.out.println("*** excepcion obteniendo BGestorUsuario del XML ***");
        ex.printStackTrace();
    }
}
```

Código 2.17. Implementación de servicios constructor.

2.4.3 MotorXML

Esta clase se utiliza para la obtención de los datos de los archivos XPDL por medio de la utilización de los archivos de configuración del mapeo de datos en XML.

En el Código 2.18. se muestra el código fuente del método escribir, este se utiliza para escribir en los archivos XPDL los datos contenidos en el tercer parámetro enviado, por tanto se enviará los gestores de los objetos usuarios, proyectos, actividades, artefactos y empresas.

```

public static void escribir(String strArchivoMapping, String strArchivoXML, Object objeto) throws
Exception {
    FileWriter writer = new FileWriter(strArchivoXML);
    Mapping mapping = new Mapping();
    mapping.loadMapping(strArchivoMapping);
    Marshaller marshaller = new Marshaller(writer);
    marshaller.setMarshalAsDocument(true);
    marshaller.setMapping(mapping);
    marshaller.marshal(objeto);
    writer.close();
}

```

Código 2.18. MotorXML método escribir.

En el Código 2.19. se muestra el código fuente del método leer, este se utiliza para obtener los datos desde los archivos XPDL, estos se mapean con los archivos mapping y finalmente devuelven un objeto que es el gestor de las estructuras de objetos de los usuarios, proyectos, actividades, artefactos y clientes.

```

public static Object leer(String strArchivoMapping, String strArchivoXML) throws Exception {
    FileReader reader = new FileReader(strArchivoXML);
    Mapping mapping = new Mapping();
    mapping.loadMapping(strArchivoMapping);
    Unmarshaller unmarshaller = new Unmarshaller(mapping);
    Object objeto = unmarshaller.unmarshal(reader);
    reader.close();
    return objeto;
}

```

Código 2.19. MotorXML método leer.

En las siguientes secciones se presentarán las estructuras de los archivos XML que se utilizan para el mapeo entre los datos contenidos en los XPDL y las estructuras de las entidades. El mapeo se utiliza para relacionar las entidades y sus atributos con los archivos XPDL, de esta forma la librería castor se encarga de generar automáticamente las estructuras cargadas con sus correspondientes datos.

2.5 Creación de Instancia de Metodología.

En la presente sección se revisará las estructuras utilizadas para almacenar la información de: metodología, gestor de metodologías, configuración de metodologías, artefactos y actividades de la metodología.

2.5.1 Artefactos de la Metodología

En el Código 2.20. se muestra la estructura que define un artefacto de la metodología. La estructura será utilizada para mapear los datos que se encuentran en el archivo XPDL con la estructura de la entidad.

```

<mapping>
  <class name="beans.BMetodologiaArtefacto">
    <map-to xml="Artifact"/>
    <field name="strId" type="java.lang.String" get-method="getStrId"
      set-method="setStrId">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strName" type="java.lang.String" get-method="getStrName"
      set-method="setStrName">
      <bind-xml name="Name" node="attribute"/>
    </field>
    <field name="strArtefactType" type="java.lang.String" get-method="getStrArtefactType"
      set-method="setStrArtefactType">
      <bind-xml name="ArtifactType" node="attribute"/>
    </field>
  </class>
</mapping>

```

Código 2.20. Esquema de Datos – Artefactos de la Metodología.

2.5.2 Actividades de la Metodología

En el Código 2.21. se muestra la estructura de datos para manejar las actividades de una metodología definida en un proceso. La estructura será utilizada para mapear los datos que se encuentran en el archivo XPDL con la estructura de la entidad.

```

<mapping>
  <class name="beans.BMetodologiaActividad">
    <map-to xml="Activity"/>
    <field name="strId" type="java.lang.String" get-method="getStrId"
      set-method="setStrId">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strNombre" type="java.lang.String" get-method="getStrNombre"
      set-method="setStrNombre">
      <bind-xml name="Name" node="attribute"/>
    </field>
  </class>
</mapping>

```

Código 2.21. Esquema de Datos – Actividades de la Metodología.

2.5.3 Gestor de Actividades de la Metodología

En el Código 2.22. se muestra la estructura de datos que permite manejar todas las actividades de las metodologías definidas en un proceso. En el campo "listMetodologiaActividades" se guarda ésta información.


```

<mapping>
  <include href="mappingMetodologiaActividad.xml"/>
  <class name="beans.BGestorMetodologiaActividad">
    <map-to xml="Activities"/>
    <field name="listMetodologiaActividades" type="beans.BMetodologiaActividad"
      get-method="getListMetodologiaActividades"
      set-method="setListMetodologiaActividades" collection="vector">
      <bind-xml name="Activity" node="element"/>
    </field>
  </class>
</mapping>

```

Código 2.22. Esquema de Datos – Gestor de Actividades de la Metodología.

2.5.4 Gestor de Artefactos de la Metodología

En el Código 2.23. se muestra la estructura de datos que permite manejar todos los artefactos de las metodologías definidas en un proceso. En el campo “listMetodologiaArtefactos” se guarda ésta información.

```

<mapping>
  <include href="mappingMetodologiaArtefacto.xml"/>
  <class name="beans.BGestorMetodologiaArtefacto">
    <map-to xml="Artifacts"/>
    <field name="listMetodologiaArtefactos" type="beans.BMetodologiaArtefacto"
      get-method="getListMetodologiaArtefactos"
      set-method="setListMetodologiaArtefactos" collection="vector">
      <bind-xml name="Artifact" node="element"/>
    </field>
  </class>
</mapping>

```

Código 2.23. Esquema de Datos – Gestor de Artefactos de la Metodología.

2.5.5 Metodología

En el Código 2.24. se muestra la estructura de datos para manejar las metodologías definidas en un proceso. La estructura será utilizada para mapear los datos que se encuentran en el archivo XPDL con la estructura de la entidad.

```

<mapping>
  <include href="mappingGestorMetodologiaActividad.xml"/>

  <class name="beans.BMetodologia">
    <map-to xml="WorkflowProcess"/>
    <field name="strId" type="java.lang.String" get-method="getStrId"
      set-method="setStrId">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strNombre" type="java.lang.String" get-method="getStrNombre"

```

```

    set-method="setStrNombre">
    <bind-xml name="Name" node="attribute"/>
  </field>
  <field name="strEstatus" type="java.lang.String" get-method="getStrEstatus"
    set-method="setStrEstatus">
    <bind-xml name="status" node="attribute"/>
  </field>
  <field name="objGestorMetodologiaActividad" type="beans.BGestorMetodologiaActividad"
    get-method="getObjGestorMetodologiaActividad"
    set-method="setObjGestorMetodologiaActividad">
    <bind-xml name="Activities" node="element"/>
  </field>
</class>
</mapping>

```

Código 2.24. Esquema de Datos – Metodología.

2.5.6 Paquete de Metodología

En el Código 2.25. se muestra la estructura de datos para manejar los paquetes de la metodología definida en un proceso. La estructura será utilizada para mapear los datos que se encuentran en el archivo XPDL con la estructura de la entidad.

```

<mapping>
  <include href="mappingGestorMetodologiaArtefacto.xml"/>
  <include href="mappingMetodologia.xml"/>

  <class name="beans.BMetodologiaPaquete">
    <map-to xml="Package"/>
    <field name="strId" type="java.lang.String" get-method="getStrId"
      set-method="setStrId">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strNombre" type="java.lang.String" get-method="getStrNombre"
      set-method="setStrNombre">
      <bind-xml name="Name" node="attribute"/>
    </field>
    <field name="objGestorMetodologiaArtefacto" type="beans.BGestorMetodologiaArtefacto"
      get-method="getObjGestorMetodologiaArtefacto"
      set-method="setObjGestorMetodologiaArtefacto">
      <bind-xml name="Artifacts" node="element"/>
    </field>
    <field name="objMetodologia" type="beans.BMetodologia"
      get-method="getObjMetodologia"
      set-method="setObjMetodologia">
      <bind-xml name="WorkflowProcess" node="element"/>
    </field>
  </class>
</mapping>

```

Código 2.25. Esquema de Datos – Paquete de Metodología.

2.5.7 Configuración de la Metodología

En el Código 2.26. se muestra la estructura de datos para manejar la configuración de la metodología. La estructura será utilizada para mapear los datos que se encuentran en el archivo XPDL con la estructura de la entidad.

```
<mapping>
  <class name="beans.BConfMetodologia">
    <map-to xml="Metodologia"/>
    <field name="strId" type="java.lang.String" get-method="getStrId"
      set-method="setStrId">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strArchivo" type="java.lang.String" get-method="getStrArchivo"
      set-method="setStrArchivo">
      <bind-xml name="Archivo" node="element"/>
    </field>
  </class>
</mapping>
```

Código 2.26. Esquema de Datos – Paquete de Metodología.

2.5.8 Gestor de la Configuración de la Metodología

En el Código 2.27. se muestra la estructura de datos que permite manejar todos los artefactos de las metodologías definidas en un proceso. En el campo “listMetodologiaArtefactos” se guarda esta información.

```
<mapping>
  <include href="mappingConfMetodologia.xml"/>
  <class name="beans.BGestorConfMetodologia">
    <map-to xml="GestorMetodologias"/>
    <field name="listMetodologias" type="beans.BConfMetodologia" get-
method="getListMetodologias"
      set-method="setListMetodologias" collection="vector">
      <bind-xml name="Metodologia" node="element"/>
    </field>
  </class>
</mapping>
```

Código 2.27. Esquema de Datos – Gestor de la Configuración de la Metodología.

2.6 Administración de Proyectos.

En la presente sección se revisará las estructuras utilizadas para almacenar la información de los proyectos.

2.6.1 Proyecto

En el Código 2.28. se muestra la estructura de datos para manejar los proyectos de una metodología definida en un proceso, cuando se realiza la creación de un proyecto se crea la estructura de archivos para el repositorio del proyecto. La estructura será utilizada para mapear los datos que se encuentran en el archivo XPDL con la estructura de la entidad.

El campo “strAvanceProy” almacena el porcentaje de avance del trabajo realizado del proyecto completo, este campo puede ser ingresado manualmente cuando se crea el proyecto, pero luego es recalculado automáticamente en función al promedio ponderado del progreso de sus actividades, cada vez que estas son actualizadas.

De forma similar los campos “datFechaIniProy” y “datFechaFinProy” pueden ser ingresados manualmente al crearse el proyecto pero son recalculados automáticamente obteniéndose de la menor fecha de inicio y de la mayor fecha de fin de las actividades del proyecto.

```

<mapping>
  <class name="beans.BProyecto">
    <map-to xml="Proyecto"/>
    <field name="strIdProyecto" type="java.lang.String" get-method="getStrIdProyecto"
      set-method="setStrIdProyecto">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strCodProyecto" type="java.lang.String" get-method="getStrCodProyecto"
      set-method="setStrCodProyecto">
      <bind-xml name="Codigo" node="element"/>
    </field>
    <field name="strNombre" type="java.lang.String" get-method="getStrNombre"
      set-method="setStrNombre">
      <bind-xml name="Nombre" node="element"/>
    </field>
    <field name="strDescProy" type="java.lang.String" get-method="getStrDescProy"
      set-method="setStrDescProy">
      <bind-xml name="DescripcionProy" node="element"/>
    </field>
    <field name="strIdPackage" type="java.lang.String" get-method="getStrIdPackage"
  
```

```

    set-method="setStrIdPackage">
    <bind-xml name="IdPackage" node="element"/>
  </field>
  <field name="strIdMetodologia" type="java.lang.String"
    get-method="getStrIdMetodologia"
    set-method="setStrIdMetodologia">
    <bind-xml name="IdMetodologia" node="element"/>
  </field>
  <field name="strIdResponsableProy" type="java.lang.String"
    get-method="getStrIdResponsableProy"
    set-method="setStrIdResponsableProy">
    <bind-xml name="IdResponsableProy" node="element"/>
  </field>
  <field name="strIdEmpresa" type="java.lang.String" get-method="getStrIdEmpresa"
    set-method="setStrIdEmpresa">
    <bind-xml name="IdEmpresa" node="element"/>
  </field>
  <field name="strAvanceProy" type="java.lang.String" get-method="getStrAvanceProy"
    set-method="setStrAvanceProy">
    <bind-xml name="AvanceProy" node="element"/>
  </field>
  <field name="strIdMoneda" type="java.lang.String"
    get-method="getStrIdMoneda"
    set-method="setStrIdMoneda">
    <bind-xml name="IdMoneda" node="element"/>
  </field>
  <field name="strIdEstadoRepositorio" type="java.lang.String"
    get-method="getStrIdEstadoRepositorio"
    set-method="setStrIdEstadoRepositorio">
    <bind-xml name="IdEstadoRepositorio" node="element"/>
  </field>
  <field name="strIdEstadoProy" type="java.lang.String" get-method="getStrIdEstadoProy"
    set-method="setStrIdEstadoProy">
    <bind-xml name="IdEstadoProy" node="element"/>
  </field>
  <field name="strIdPrioridadProy" type="java.lang.String"
    get-method="getStrIdPrioridadProy"
    set-method="setStrIdPrioridadProy">
    <bind-xml name="IdPrioridadProy" node="element"/>
  </field>
  <field name="strIdTipoProy" type="java.lang.String" get-method="getStrIdTipoProy"
    set-method="setStrIdTipoProy">
    <bind-xml name="IdTipoProy" node="element"/>
  </field>
  <field name="datFechaIniProy" type="java.util.Date" get-method="getDatFechaIniProy"
    set-method="setDatFechaIniProy">
    <bind-xml name="FechaIniProy1" node="element"/>
  </field>
  <field name="datFechaFinProy" type="java.util.Date" get-method="getDatFechaFinProy"
    set-method="setDatFechaFinProy">
    <bind-xml name="FechaFinProy1" node="element"/>
  </field>
  <field name="strIdEtapaProy" type="java.lang.String" get-method="getStrIdEtapaProy"
    set-method="setStrIdEtapaProy">
    <bind-xml name="IdEtapaProy" node="element"/>
  </field>
  <field name="strPresupuestoProy" type="java.lang.String"
    get-method="getStrPresupuestoProy"

```

```

    set-method="setStrPresupuestoProy">
    <bind-xml name="Presupuesto" node="element"/>
  </field>
</class>
</mapping>

```

Código 2.28. Esquema de Datos – Proyecto.

2.6.2 Gestor de Proyectos

En el Código 2.29. se muestra la estructura de datos que permite manejar todos los proyectos de las metodologías definidas en un proceso. En el campo “listProyectos” se guarda ésta información.

```

<mapping>
  <include href="mappingProyecto.xml"/>
  <class name="beans.BGestorProyecto">
    <map-to xml="GestorProyecto"/>
    <field name="listProyectos" type="beans.BProyecto" get-method="getListProyectos"
      set-method="setListProyectos" collection="vector">
      <bind-xml name="Proyecto" node="element"/>
    </field>
  </class>
</mapping>

```

Código 2.29. Esquema de Datos – Gestor de Proyectos.

2.7 Administración de Proyecto Específico

En la presente sección se revisará las estructuras utilizadas para almacenar la información de los proyectos específicos.

2.7.1 Actividades del Proyecto

En el Código 2.30. se muestra la estructura de datos utilizada para la creación de una actividad asociada a un proyecto determinado, a la actividad se le puede relacionar una o más actividades de una metodología empleada en el proyecto que se ha creado, este dato se guarda en el campo “strIdActividadMetodologia”.

Las actividades creadas pueden ser hijos de una actividad principal o padre, este dato se guarda en el campo “strIdActividadPadre”. Las horas que esperamos tome la actividad se almacena en el campo “strHorasTrabajo”, cabe resaltar que las horas reales de la actividad se calcularán sumando las horas de trabajo de la estructura de las Horas de la Actividad que puede revisarse en la sección 2.7.3.

El campo "strProgreso" guarda el porcentaje de avance del trabajo realizado en la actividad, este campo es calculado para las actividades principales en función al promedio ponderado del progreso de sus actividades hijas.

```

<mapping>
  <class name="beans.BActividad">
    <map-to xml="Actividad"/>
    <field name="strIdActividad" type="java.lang.String"
      get-method="getStrIdActividad"
      set-method="setStrIdActividad">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strIdActividadMetodologia" type="java.lang.String"
      get-method="getStrIdActividadMetodologia"
      set-method="setStrIdActividadMetodologia">
      <bind-xml name="IdActividadMetodologia" node="element"/>
    </field>
    <field name="strIdProyecto" type="java.lang.String" get-method="getStrIdProyecto"
      set-method="setStrIdProyecto">
      <bind-xml name="IdProyecto" node="element"/>
    </field>
    <field name="strNombre" type="java.lang.String" get-method="getStrNombre"
      set-method="setStrNombre">
      <bind-xml name="Nombre" node="element"/>
    </field>
    <field name="strIdActividadPadre" type="java.lang.String"
      get-method="getStrIdActividadPadre"
      set-method="setStrIdActividadPadre">
      <bind-xml name="ActividadPadre" node="element"/>
    </field>
    <field name="strIdResponsable" type="java.lang.String"
      get-method="getStrIdResponsable"
      set-method="setStrIdResponsable">
      <bind-xml name="UsuarioPropietario" node="element"/>
    </field>
    <field name="strIdTipo" type="java.lang.String"
      get-method="getStrIdTipo"
      set-method="setStrIdTipo">
      <bind-xml name="Tipo" node="element"/>
    </field>
    <field name="strIdEstado" type="java.lang.String"
      get-method="getStrIdEstado" set-method="setStrIdEstado">
      <bind-xml name="Estado" node="element"/>
    </field>
    <field name="strIdPrioridad" type="java.lang.String"
      get-method="getStrIdPrioridad"
      set-method="setStrIdPrioridad">
      <bind-xml name="Prioridad" node="element"/>
    </field>
    <field name="strProgreso" type="java.lang.String" get-method="getStrProgreso"
      set-method="setStrProgreso">
      <bind-xml name="Progreso" node="element"/>
    </field>
    <field name="strHito" type="java.lang.String" get-method="getStrHito"
  
```

```

    set-method="setStrHito">
    <bind-xml name="Hito" node="element"/>
  </field>
  <field name="strDescripcion" type="java.lang.String" get-method="getStrDescripcion"
    set-method="setStrDescripcion">
    <bind-xml name="Descripcion" node="element"/>
  </field>
  <field name="datFechalnicio" type="java.util.Date"
    get-method="getDatFechalnicio"
    set-method="setDatFechalnicio">
    <bind-xml name="Fechalnicio" node="element"/>
  </field>
  <field name="datFechaFin" type="java.util.Date"
    get-method="getDatFechaFin"
    set-method="setDatFechaFin">
    <bind-xml name="FechaFin" node="element"/>
  </field>
  <field name="strHorasTrabajo" type="java.lang.String"
    get-method="getStrHorasTrabajo"
    set-method="setStrHorasTrabajo">
    <bind-xml name="HorasTrabajo" node="element"/>
  </field>
</class>
</mapping>

```

Código 2.30. Esquema de Datos – Actividades del Proyecto.

2.7.2 Gestor de Actividades

En el Código 2.31. se muestra la estructura de datos que permite manejar todas las actividades registradas en el sistema. En el campo “listActividades” se guarda esta información.

```

<mapping>
  <include href="mappingActividades.xml"/>
  <class name="beans.BGestorActividades">
    <map-to xml="GestorActividad"/>
    <field name="listActividades" type="beans.BActividad"
      get-method="getListActividades"
      set-method="setListActividades" collection="vector">
      <bind-xml name="Actividad" node="element"/>
    </field>
  </class>
</mapping>

```

Código 2.31. Esquema de Datos – Gestor Actividades.

2.7.3 Horas trabajadas de la Actividad

En el Código 2.32. se muestra la estructura de datos utilizada para almacenar los trabajos realizados para cumplir las actividades, estas brindan las horas reales trabajadas de las actividades. En los campos “datFecha” y “strHoras” se guardan las fechas y horas en que se realizó el trabajo, además en el campo “strIdUsuario”

se almacena el id del usuario que trabajo la actividad, sólo los usuarios responsables de la actividad tienen acceso a ingresar las horas trabajadas de dicha actividad.

```

<mapping>
  <class name="beans.BActividadHoras">
    <map-to xml="ActividadHoras"/>
    <field name="strId" type="java.lang.String" get-method="getStrId"
      set-method="setStrId">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strIdActividad" type="java.lang.String" get-method="getStrIdActividad"
      set-method="setStrIdActividad">
      <bind-xml name="IdActividad" node="element"/>
    </field>
    <field name="datFecha" type="java.util.Date" get-method="getDatFecha"
      set-method="setDatFecha">
      <bind-xml name="Fecha" node="element"/>
    </field>
    <field name="strHoras" type="java.lang.String" get-method="getStrHoras" set-
      method="setStrHoras">
      <bind-xml name="Horas" node="element"/>
    </field>
    <field name="strIdUsuario" type="java.lang.String" get-method="getStrIdUsuario" set-
      method="setStrIdUsuario">
      <bind-xml name="IdUsuario" node="element"/>
    </field>
    <field name="strIdProyecto" type="java.lang.String" get-method="getStrIdProyecto" set-
      method="setStrIdProyecto">
      <bind-xml name="IdProyecto" node="element"/>
    </field>
  </class>
</mapping>

```

Código 2.32. Esquema de Datos – Horas trabajadas de la Actividad.

2.7.4 Usuarios de la Actividad del Proyecto

En el Código 2.33. se muestra la estructura de datos utilizada para asociar usuarios a una actividad de un proyecto determinado. Se guarda la relación entre el proyecto y la actividad en los campos “strIdProyecto” y “strIdActividad” respectivamente. En el campo “listActividadRecursos” se guarda la relación de usuarios que van a estar relacionados a la actividad.

```

<mapping>
  <include href="mappingActividadRecursos.xml"/>
  <class name="beans.BGestorActividadRecursos">
    <map-to xml="GestorActividadRecursos"/>
    <field name="strIdProyecto" type="java.lang.String"
      get-method="getStrIdProyecto"

```

```

    set-method="setStrIdProyecto">
    <bind-xml name="IdProyecto" node="attribute"/>
  </field>
  <field name="strIdActividad" type="java.lang.String"
    get-method="getStrIdActividad"
    set-method="setStrIdActividad">
    <bind-xml name="IdActividad" node="attribute"/>
  </field>
  <field name="listActividadRecursos" type="beans.BActividadRecursos"
    get-method="getListActividadRecursos"
    set-method="setListActividadRecursos" collection="vector">
    <bind-xml name="ActividadRecursos" node="element"/>
  </field>
</class>
</mapping>

```

Código 2.33. Esquema de Datos – Usuarios de la Actividad del Proyecto.

2.8 Gestión de la Configuración

En la presente sección se revisará las estructuras utilizadas para almacenar los datos de la gestión de configuración, esta incluye el gestor de los artefactos del repositorio, los artefactos en sí son los entes que están compuestos por los documentos y código fuente generado en las diferentes actividades además de las versiones de cada uno de estos.

2.8.1 Versión de Artefactos

En el Código 2.34. se muestra la estructura de datos para manejar las versiones de un determinado artefacto, en esta debe almacenarse la extensión del archivo, el ID del usuario y la fecha en que fue cargada la versión.

```

<mapping>
  <class name="beans.BArtefactoVersion">
    <map-to xml="ArtefactoVersion"/>
    <field name="strIdArtefactoVersion" type="java.lang.String"
      get-method="getStrIdArtefactoVersion"
      set-method="setStrIdArtefactoVersion">
      <bind-xml name="IdArtefactoVersion" node="attribute"/>
    </field>
    <field name="strNroVersion" type="java.lang.String" get-method="getStrNroVersion"
      set-method="setStrNroVersion">
      <bind-xml name="NroVersion" node="attribute"/>
    </field>
    <field name="strRutaArtefacto" type="java.lang.String" get-method="getStrRutaArtefacto"
      set-method="setStrRutaArtefacto">
      <bind-xml name="RutaArtefacto" node="attribute"/>
    </field>
    <field name="strIdCarpeta" type="java.lang.String" get-method="getStrIdCarpeta"

```

```

    set-method="setStrIdCarpeta">
    <bind-xml name="IdCarpeta" node="attribute"/>
  </field>
  <field name="strDescripcion" type="java.lang.String" get-method="getStrDescripcion"
    set-method="setStrDescripcion">
    <bind-xml name="Descripcion" node="attribute"/>
  </field>
  <field name="strIdUsuario" type="java.lang.String" get-method="getStrIdUsuario"
    set-method="setStrIdUsuario">
    <bind-xml name="IdUsuario" node="attribute"/>
  </field>
  <field name="strNomUsuario" type="java.lang.String"
    get-method="getStrNomUsuario"
    set-method="setStrNomUsuario">
    <bind-xml name="NomUsuario" node="attribute"/>
  </field>
  <field name="strExtension" type="java.lang.String" get-method="getStrExtension"
    set-method="setStrExtension">
    <bind-xml name="Extension" node="attribute"/>
  </field>
  <field name="strFecha" type="java.lang.String" get-method="getStrFecha"
    set-method="setStrFecha">
    <bind-xml name="Fecha" node="attribute"/>
  </field>
  <field name="datFecha" type="java.util.Date" get-method="getDatFecha"
    set-method="setDatFecha">
    <bind-xml name="Fecha1" node="attribute"/>
  </field>
</class>
</mapping>

```

Código 2.34. Esquema de Datos – Versión de Artefactos.

2.8.2 Artefactos

En el Código 2.35 se muestra la estructura de datos que permite manejar los datos de un artefacto: “strIdArtefacto” se refiere al identificador del artefacto, el número y el identificador de la versión vigente se encuentran en los campos “strNroVersion” y “strIdArtefactoVersionVigente” respectivamente.

En el campo “strIdActividades” se almacena los identificadores de las actividades a las que se relaciona el artefacto separados por comas, de igual forma en el campo “strIdArtefactoMetodologia” se encuentra la relación con los artefactos correspondiente a la metodología.

El campo “strIdEstado” indica el estado en que puede estar el artefacto (reservado para un usuario en específico o libre), además en caso que un usuario reserve el artefacto debe almacenarse el identificador de usuario y la fecha en que se realizó la reserva en los campos “strIdUsuarioReservado” y “datFechaReservado”.

También se almacena el historial de versiones que contiene el artefacto, en el campo “listArtefactoVersiones”.

```

<mapping>
  <include href="mappingArtefactoVersion.xml"/>
  <class name="beans.BArtefacto">
    <map-to xml="Artefacto"/>
    <field name="strIdArtefacto" type="java.lang.String"
      get-method="getStrIdArtefacto"
      set-method="setStrIdArtefacto">
      <bind-xml name="Id" node="attribute"/>
    </field>
    <field name="strNomArtefacto" type="java.lang.String"
      get-method="getStrNomArtefacto"
      set-method="setStrNomArtefacto">
      <bind-xml name="Name" node="attribute"/>
    </field>
    <field name="strIdArtefactoVersionVigente" type="java.lang.String"
      get-method="getStrIdArtefactoVersionVigente"
      set-method="setStrIdArtefactoVersionVigente">
      <bind-xml name="IdArtefactoVersionVigente" node="attribute"/>
    </field>
    <field name="strIdActividades" type="java.lang.String" get-method="getStrIdActividades" set-
method="setStrIdActividades">
      <bind-xml name="IdActividades" node="attribute"/>
    </field>
    <field name="strIdArtefactoMetodologia" type="java.lang.String" get-
method="getStrIdArtefactoMetodologia" set-method="setStrIdArtefactoMetodologia">
      <bind-xml name="IdArtefactoMetodologia" node="attribute"/>
    </field>
    <field name="strDescripcion" type="java.lang.String"
      get-method="getStrDescripcion"
      set-method="setStrDescripcion">
      <bind-xml name="Descripcion" node="attribute"/>
    </field>
    <field name="strIdUsuarioReservado" type="java.lang.String"
      get-method="getStrIdUsuarioReservado"
      set-method="setStrIdUsuarioReservado">
      <bind-xml name="IdUsuarioReservado" node="attribute"/>
    </field>
    <field name="strNomUsuarioReservado" type="java.lang.String"
      get-method="getStrNomUsuarioReservado"
      set-method="setStrNomUsuarioReservado">
      <bind-xml name="NomUsuarioReservado" node="attribute"/>
    </field>
    <field name="strIdEstado" type="java.lang.String"
      get-method="getStrIdEstado"
      set-method="setStrIdEstado">
      <bind-xml name="IdEstado" node="attribute"/>
    </field>
    <field name="strNomEstado" type="java.lang.String"
      get-method="getStrNomEstado"
      set-method="setStrNomEstado">
      <bind-xml name="NomEstado" node="attribute"/>
    </field>
  </class>

```

```

<field name="datFechaReservado" type="java.util.Date"
  get-method="getDatFechaReservado"
  set-method="setDatFechaReservado">
  <bind-xml name="FechaReservado1" node="attribute"/>
</field>
<field name="strNroVersion" type="java.lang.String"
  get-method="getStrNroVersion"
  set-method="setStrNroVersion">
  <bind-xml name="NroVersion" node="attribute"/>
</field>
<field name="listArtefactoVersiones" type="beans.BArtefactoVersion"
  get-method="getListArtefactoVersiones"
  set-method="setListArtefactoVersiones" collection="vector">
  <bind-xml name="ArtefactoVersion" node="element"/>
</field>
</class>
</mapping>

```

Código 2.35. Esquema de Datos – Artefacto

2.8.3 Gestor de Artefactos

En el Código 2.36. se muestra la estructura de datos que permite manejar los artefactos que están contenidos en el repositorio.

Además los artefactos se encuentran físicamente almacenados siguiendo una jerarquía de carpetas de tal forma que primero se encuentran los identificadores de los proyectos, dentro las carpetas con los identificadores de los artefactos y finalmente los archivos físicos con los identificadores de la versión a la que corresponde.

```

<mapping>
<include href="mappingArtefacto.xml"/>
<class name="beans.BGestorArtefactos">
  <map-to xml="GestorProyectoArtefactos"/>
  <field name="strIdProyecto" type="java.lang.String" get-method="getStrIdProyecto"
    set-method="setStrIdProyecto">
    <bind-xml name="IdProyecto" node="attribute"/>
  </field>
  <field name="strRutaLocal" type="java.lang.String" get-method="getStrRutaLocal"
    set-method="setStrRutaLocal">
    <bind-xml name="RutaLocal" node="attribute"/>
  </field>
  <field name="listArtefactos" type="beans.BArtefacto" get-method="getListArtefactos"
    set-method="setListArtefactos" collection="vector">
    <bind-xml name="Artefacto" node="element"/>
  </field>
</class>
</mapping>

```

Código 2.36. Esquema de Datos – Gestor de Artefactos.

2.9 Diagrama de Gantt

Los diagramas de Gantt de la herramienta fueron generados gracias a la utilización de la librería jsGantt [JSGANTT], esta se encuentra codificada en Javascript y por lo tanto permite integrarse fácilmente a cualquier tecnología o lenguaje de programación. En esta sección se explicará cómo se realizó la configuración para elaborar los diagramas Gantt de los proyectos, así como la forma en que fueron implementados.

2.9.1 Configuración de scripts

Debido a que la librería jsGantt se encuentra desarrollada en Javascript, sólo es necesario incluir la ruta donde se encuentra la librería y su correspondiente hoja de estilos en el JSP donde se desee mostrar el diagrama de Gantt, tal como se muestra en el código 2.37.

```
<ui:head id="head1">
  <ui:link id="link2" url="/resources/jsgantt.css"/>
  <ui:script id="script1" url="/resources/js/jsgantt.js"/>
```

Código 2.37. Diagrama de Gantt – Configuración de scripts.

2.9.2 Generación de datos

La librería utilizada nos proporciona dos formas de poder ingresar los datos que se mostrarán en el diagrama de Gantt:

- La primera por medio de la cual se escribe código javascript y se va invocando las funciones de la librería pasando por parámetros los datos de las actividades.
- La segunda nos permite crear un archivo XML que contendrá los datos de las actividades, el archivo XML debe tener un formato establecido en la documentación de la librería.

Debido a que la segunda opción nos permite utilizar un archivo XML y estos archivos son un estándar para cualquier tecnología, se ha optado por esta opción. En el Código 2.38. se muestra el formato que debe tener el archivo XML con los datos de las actividades.

```

<project>
<task>
  <pID>1</pID>
  <pName>activ 1</pName>
  <pStart>28/10/2010</pStart>
  <pEnd>06/11/2010</pEnd>
  <pColor>CC0000</pColor>
  <pLink></pLink>
  <pMile>0</pMile>
  <pRes>Silva Lazo, Anita Yesenia</pRes>
  <pComp>70</pComp>
  <pGroup>1</pGroup>
  <pParent>0</pParent>
  <pOpen>1</pOpen>
  <pDepend></pDepend>
  <pCaption></pCaption>
</task>
</project>

```

Código 2.38. Diagrama de Gantt – Generación de datos.

A continuación se especifica el significado de cada tag:

- **pID:** Se refiere al indicador único de cada actividad, el cual se podrá utilizar en el tag “pParent” en el caso que posea actividades hijas.
- **pName:** Debe ingresarse el nombre de la actividad a mostrarse.
- **pStart:** La fecha de inicio de la actividad.
- **pEnd:** La fecha de fin de la actividad
- **pColor:** Se utiliza para especificar un color a la actividad, para el software se han utilizado tres colores: rojo (actividades que han transcurrido el 80% o más del tiempo estimado), anaranjado (para las actividades que han transcurrido más del 40% pero menos del 80% del tiempo estimado) y azul (para las actividades que han transcurrido menos 40% del estimado).
- **pRes:** Debe ingresarse el nombre del responsable de la actividad que va a mostrarse.
- **pComp:** Indica el porcentaje de avance de la actividad, este número además de mostrarse también permitirá generar una barra interna que muestre el avance de forma gráfica.
- **pGroup:** Para el caso que se trate de una actividad que contenga actividades hijas, este valor debe estar en 1.
- **pParent:** Como ya se indicó este valor se refiere al indicador “pID” de la actividad padre en caso se trate de una actividad hija.
- **pOpen:** Para el caso de las actividades que tengan actividades hijas esta indica si se debe mostrar las actividades hijas desplegadas o cerrada.

Los demás tags no se han utilizado para el presente proyecto.

2.9.3 Implementación del diagrama

Finalmente para mostrar el diagrama de Gantt en un JSP solo debemos agregar un DIV con la configuración mostrada en el Código 2.39. gráficamente en este tag se mostrará el diagrama de Gantt.

```
<div class="gantt" id="GanttChartDIV" style="left: 10%; top: 312px; position: absolute; width: 100%"></div>
```

Código 2.39. Diagrama de Gantt – Componente DIV.

Además del DIV ingresado debe escribirse algunas líneas de código javascript, como puede verse en el Código 2.40. se debe indicar:

- El Id del DIV creado donde se mostrará el diagrama.
- Además puede configurarse, para esto debe indicarse por parámetros (1: si se quiere mostrar o 0: si no se quiere mostrar) a los métodos: setShowRes (se refiere al responsable de la actividad), setShowDur (para mostrar los días de duración), setShowComp (para mostrar el porcentaje de avance), setDateInputFormat (formato de fechas que debe tener las actividades) y setDateDisplayFormat (formato de fechas en que se debe mostrar las actividades).
- Debido a que hemos utilizado un archivo XML con los datos de las actividades, debe indicarse la ruta donde se encuentra este archivo en la instrucción: JSgant.parseXML("ruta del xml").
- Finalmente las instrucciones g.Draw y g.DrawDependencies utilizarán el archivo XML con los datos y la configuración establecida para mostrar el diagrama de Gantt en el DIV especificado.

```
<script>
  var g = new JSgant.GanttChart('g',document.getElementById('GanttChartDIV'), 'day');
  g.setShowRes(1);
  g.setShowDur(0);
  g.setShowComp(1);
  g.setCaptionType('Complete');
  g.setDateInputFormat('dd/mm/yyyy')
  g.setDateDisplayFormat('dd/mm/yyyy')
  if ( g ) {
    var archivo= document.getElementById("form1:hiddenField1").value;
    JSgant.parseXML("../gantt/"+archivo,g)
    g.Draw();
    g.DrawDependencies();
  }else{
    alert("not defined");
  }
</script>
```

Código 2.40. Diagrama de Gantt – Código javascript para mostrar el diagrama.

Al final se muestra la Figura 2.9 con el diagrama de Gantt generado y presentado en la pantalla JSP, esta muestra la siguiente información:

- Todas las actividades ingresadas para el proyecto, tal como se ha configurado estas pueden desplegar actividades hijas.
- Usuario responsable de cada actividad.

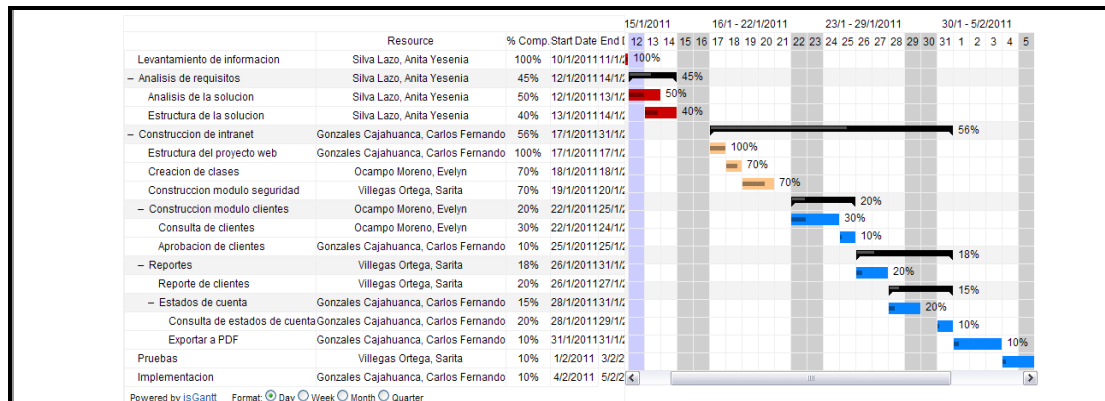


Figura 2.9. Diagrama de Gantt – Formato en pantalla

- Porcentaje de avance de la actividad, esta es actualizada por el usuario, por lo que representa el avance real de la actividad.
- Fechas de inicio y fin de la actividad.
- Diagrama de Gantt en el calendario con las barritas de actividades según los colores para avisar sobre el fin de cada una de ellas, además se indica el porcentaje de avance.

Capítulo 3: Pruebas

El presente capítulo presenta las pruebas unitarias y prueba del sistema realizadas a la herramienta EACS Project Manager, para verificar su correcto funcionamiento. Se consideró realizar pruebas basadas en los casos de uso, que verifiquen la correcta implementación de los flujos básicos y alternativos presentados en la especificación de casos de uso, con el fin de detectar los defectos y poder corregirlos, los casos negativos no se realizaron de manera formal. No se realizó otros tipos de pruebas de manera formal, ni desarrolló test drivers o stub de pruebas para todos los casos, debido a que la aplicación es sencilla en la parte de interacción con el usuario y la mayor complejidad está en el manejo de XML y manipulación de los archivos a la gestión de configuración.

A continuación se detallan las pruebas positivas:

3.1. Prueba caso de uso: Mantener Proyecto.

Caso de Prueba	Registrar Proyecto
Descripción Breve	Probar el flujo principal del caso de uso "Mantener Proyecto".
Precondición(es)	El usuario del sistema debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción "Proyectos". 2. El sistema muestra la relación de proyectos registrados en el sistema. 3. En el sistema seleccionar la opción "Nuevo Proyecto". 4. El sistema muestra un formulario donde se deberá ingresar los siguientes datos:

	<ul style="list-style-type: none"> • Código: PBR • Nombre: Proyectos Bienes Raíces • Empresa: SistemasABC • Estado: Activo • Fecha Inicio: 15/01/2010 • Fecha Fin: 29/07/2010 • Jefe de Proyecto: Juan Pérez • Metodología: Metodología Nueva • Tipo: Inversión de Capital • Prioridad: Alta • Avance: 25% • Presupuesto: 2000 • Moneda: Dólares • Descripción: Proyecto para la inversión de capital de la empresa SistemasABC <p>5. En el sistema presionar el botón “Grabar”.</p>
Resultado(s) esperado(s)	Se muestra el nuevo proyecto creado con sus principales datos dentro de la relación de proyectos creados en el sistema.

3.2. Prueba caso de uso: Mantener Actividades.

Caso de Prueba	Registrar Actividades.
Descripción Breve	Probar el flujo principal del caso de uso "Mantener Actividades".
Precondición(es)	El usuario debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción “Proyectos”. 2. El sistema muestra la relación de proyectos registrados en el sistema. 3. En el sistema seleccionar el proyecto al que se le asociará una actividad con la opción “Ver”. 4. El sistema muestra los principales datos del proyecto seleccionado. 5. En el sistema seleccionar la opción “Actividades”. 6. El sistema muestra la relación de actividades registradas. 7. En el sistema seleccionar el botón “Nueva Actividad” para registrar una nueva actividad. 8. El sistema muestra un formulario donde se deberá ingresar los siguientes datos: <ul style="list-style-type: none"> • Nombre actividad: Crear EAR Deploy • Estado: Activo. • Responsable: Carlos Torres. • Avance: 25% • Fecha inicio: 16/01/2010 • Fecha final: 20/01/2010 • Tipo: Implementación. • Prioridad: Normal. • Actividad Padre: ninguna. • Horas Esperadas: 4 horas • Descripción: Se realizara la creación del EAR para el proyecto. • Actividades Metodología: Implementación del proyecto. 9. En el sistema presionar el botón “Grabar”.
Resultado(s)	Se muestra la actividad creada asociada al proyecto con sus

esperado(s)	principales datos dentro de la relación de actividades creadas en el proyecto seleccionado.
--------------------	---

3.3. Prueba caso de uso: Mantener Artefactos.

Caso de Prueba	Registrar Artefactos.
Descripción Breve	Probar el flujo principal del caso de uso "Mantener de Artefactos".
Precondición(es)	El usuario debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción "Proyectos". 2. El sistema muestra la relación de proyectos registrados en el sistema. 3. En el sistema seleccionar el proyecto al que se le asociará un artefacto con la opción "Ver". 4. El sistema muestra los principales datos del proyecto seleccionado. 5. En el sistema seleccionar la opción "Artefactos". 6. El sistema muestra la relación de artefactos registrados. 7. En el sistema seleccionar la opción "Nuevo Artefacto" para registrar un nuevo artefacto. 8. El sistema muestra un formulario donde se deberá ingresar los siguientes datos: <ul style="list-style-type: none"> • Nombre del artefacto: Plan de pruebas. • Archivo: Plan Pruebas.doc. • Descripción: Documento para realizar el plan de pruebas para el pase a producción. • Actividades: ninguna • Artefactos Metodología: Plan de Pruebas 9. En el sistema presionar el botón "Grabar".
Resultado(s) esperado(s)	Se muestra el artefacto creado asociado al proyecto con sus principales datos dentro de la relación de artefactos creados en el proyecto seleccionado.

3.4. Prueba caso de uso: Reservar Artefactos.

Caso de Prueba	Reservar Artefactos.
Descripción Breve	Probar el flujo principal del caso de uso "Reservar Artefactos".
Precondición(es)	El usuario del sistema debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción "Proyectos". 2. El sistema muestra la relación de proyectos registrados en el sistema. 3. En el sistema seleccionar el proyecto deseado con la opción "Ver". 4. El sistema muestra los principales datos del proyecto seleccionado. 5. En el sistema seleccionar la opción "Artefactos". 6. El sistema muestra la relación de artefactos encontrados según el proyecto seleccionado. 7. En el sistema seleccionar el artefacto a reservar con la opción "Reservar". 8. El sistema protege el artefacto y cambia el estado del artefacto a "bloqueado", así ningún otro usuario podrá modificarlo solo consultarlo, registrando la fecha de

	reserva y el usuario que realizo dicha acción.
Resultado(s) esperado(s)	Se muestra el artefacto en estado bloqueado.

3.5. Prueba caso de uso: Crear Versión Artefacto.

Caso de Prueba	Crear Versión Artefacto.
Descripción Breve	Probar el flujo principal del caso de uso "Crear Versión Artefacto".
Precondición(es)	El usuario del sistema debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción "Proyectos". 2. El sistema muestra la relación de proyectos registrados en el sistema. 3. En el sistema seleccionar el proyecto deseado con la opción "Ver". 4. El sistema muestra los principales datos del proyecto seleccionado. 5. En el sistema seleccionar la opción "Artefactos". 6. El sistema muestra la relación de artefactos encontrados según el proyecto seleccionado. 7. En el sistema seleccionar el artefacto a crear una nueva versión con la opción "Subir", el artefacto ha debido ser reservado previamente para que se habilite esta opción. 8. El sistema muestra un formulario donde se deberá ingresar los siguientes datos: <ul style="list-style-type: none"> • Archivo: C:\Repositorio\Plan Pruebas.doc • Observación: Se actualizo el plan de pruebas según las modificaciones realizadas por el área de calidad. 9. En el sistema presionar el botón "Grabar". 10. El sistema registra la nueva versión del artefacto y cambia el estado del artefacto a "desbloqueado", así los demás usuarios podrán reservarlo, registrando la fecha de generación de la nueva versión y el usuario que realizó dicha acción.
Resultado(s) esperado(s)	Se muestra el artefacto en estado desbloqueado.

3.6. Prueba caso de uso: Mantener Usuario.

Caso de Prueba	Registrar Usuario.
Descripción Breve	Probar el flujo principal del caso de uso "Mantener Usuario".
Precondición(es)	El administrador debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción "Usuarios". 2. El sistema muestra la relación de usuarios registrados en el sistema. 3. En el sistema seleccionar la opción "Nuevo Usuario". 4. El sistema muestra un formulario donde se deberá ingresar los siguientes datos: <ul style="list-style-type: none"> • Apellido Paterno: Pérez • Apellido Materno: Cáceres. • Nombres: Juan • Perfil: Administrador • Cargo: Jefe de Proyecto

	<ul style="list-style-type: none"> • Estado: Activo. • Usuario: jperez • Contraseña: jperez <p>5. En el sistema presionar el botón “Grabar”.</p>
Resultado(s) esperado(s)	Se muestra el usuario creado con sus principales datos dentro de la relación de usuarios registrados en el sistema.

3.7. Prueba caso de uso: Mantener Empresas.

Caso de Prueba	Registrar Empresa
Descripción Breve	Probar el flujo principal del caso de uso "Mantener Empresas".
Precondición(es)	El administrador debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción “Empresas”. 2. El sistema muestra la relación de empresas registradas en el sistema. 3. En el sistema seleccionar la opción “Nueva Empresa”. 4. El sistema muestra un formulario donde se deberá ingresar los siguientes datos: <ul style="list-style-type: none"> • Nom.Empresa: ABC International • Tipo: Cliente • Correo electrónico: sistemas@abc.com • Url : http://www.abc.com.pe • Teléfono1: 4474111 • Teléfono2: 4645879 • Fax: 4645879 • Dirección: Av. Begonias 405 • Cod.Postal: 27 • País: Perú. • Descripción: Tienda por departamentos. 5. En el sistema presionar el botón “Grabar”. 6. Esperar mensaje de éxito y presionar “Aceptar”
Resultado(s) esperado(s)	Se muestra la empresa creada con sus principales datos dentro de la relación de empresas del sistema.

3.8. Prueba caso de uso: Ingresar Horas Trabajadas en las Actividades.

Caso de Prueba	Registrar Horas Trabajadas
Descripción Breve	Probar el flujo principal del caso de uso "Ingresar Horas Trabajadas en las Actividades".
Precondición(es)	El usuario debe haber iniciado una sesión en el sistema con sus respectivos nombres de usuario y contraseña.
Acciones	<ol style="list-style-type: none"> 1. En el sistema seleccionar la opción “Mis Actividades”. 2. El sistema muestra la relación de actividades registradas del usuario que ha ingresado al sistema. 3. En el sistema seleccionar la actividad a la que se le registrará el número de horas trabajadas con la opción “Ver Horas”. 4. El sistema muestra un formulario donde se deberá ingresar los siguientes datos: <ul style="list-style-type: none"> • Fecha: 16/01/2010

	<ul style="list-style-type: none"> • Horas: 6 <ol style="list-style-type: none"> 5. En el sistema presionar el botón “Grabar”. 6. Esperar mensaje de éxito y presionar “Aceptar”
Resultado(s) esperado(s)	Se muestra las horas registradas para la actividad seleccionada.

3.9. Pruebas integrales del sistema

Con la finalidad de probar el correcto funcionamiento de la herramienta EACS Project Manager se realizaron las pruebas integrales del sistema.

3.9.1. Registrar proyecto completo

Nombre Paso	Descripción	Resultados Esperados
Flujo A: Registrar Usuarios		
Paso1	<ol style="list-style-type: none"> 1. Ingrese al aplicativo EACS Project 2. Ingresar con un usuario en el campo “Usuario”: 3. Ingresar el valor de la contraseña en el campo “Contraseña”. 4. Seleccionar la opción “Ingresar”. 	<p>El Sistema inicia sesión y se muestra la pantalla del listado de proyectos, con las siguientes secciones:</p> <p>Sección “Búsqueda de Proyectos” con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado <p>Sección “Lista de Proyectos”:</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado
Paso 2	Seleccione la opción “USUARIOS” del menú principal.	<p>El sistema muestra la pantalla del listado de usuarios del sistema, con las siguientes secciones:</p> <p>Sección “Búsqueda de Usuarios” con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Apellidos y Nombres - Estado - Perfil - Cargo <p>Sección “Lista de Usuarios”:</p> <ul style="list-style-type: none"> - Apellidos y nombres - Estado - Perfil - Cargo
Paso 3	Seleccione la opción “Nuevo Usuario”.	El sistema muestra la pantalla de ingreso del nuevo usuario con los siguientes

		<p>campos editables:</p> <ul style="list-style-type: none"> - Ap. Paterno - Ap. Materno - Nombres - Usuario - Contraseña <p>Y los siguientes campos seleccionables:</p> <ul style="list-style-type: none"> - Cargo - Perfil - Estado
Paso 4	<ol style="list-style-type: none"> 1. Ingrese un apellido paterno en el campo "Ap. Paterno". 2. Ingrese un apellido materno en el campo "Ap. Materno". 3. Ingrese nombres en el campo "Nombres". 4. Seleccione la opción "Analista" del campo "Cargo". 5. Seleccione la opción "Equipo" del campo "Perfil". 6. Seleccione la opción "Activo" del campo "Estado". 7. Ingrese el usuario en el campo "Usuario". 8. Ingrese la contraseña en el campo "Contraseña". 9. Seleccione la opción "Grabar". 	<p>El sistema guarda los datos ingresados por el usuario correctamente.</p> <p>El Sistema genera automáticamente un identificador que será único para este usuario.</p>
Flujo B: Registrar Empresas		
Paso 1	<p>Seleccione la opción "EMPRESAS" del menú principal.</p>	<p>El sistema muestra la pantalla del listado de empresas del sistema, con las siguientes secciones:</p> <p>Sección "Búsqueda de Empresas" con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Nombre - Tipo <p>Sección "Lista de Empresas":</p> <ul style="list-style-type: none"> - Empresa - Tipo - Correo - Teléfono
Paso 2	<p>Seleccione la opción "Nueva Empresa".</p>	<p>El sistema muestra la pantalla de ingreso de la nueva empresa con los siguientes campos editables:</p> <ul style="list-style-type: none"> - Nom. Empresa - Correo - Teléfono 1 - Teléfono 2 - Fax - Dirección - País - Cod. Postal - URL - Descripción <p>Y los siguientes campos seleccionables:</p> <ul style="list-style-type: none"> - Tipo
Paso 3	<ol style="list-style-type: none"> 1. Ingrese el nombre de la 	<p>El sistema guarda los datos ingresados para</p>

	<p>empresa en el campo "Nom. Empresa".</p> <p>2. Ingrese el correo de la empresa en el campo "Correo".</p> <p>3. Seleccione la opción "Consultora" en el campo "Tipo".</p> <p>4. Ingrese el primer teléfono en el campo "Teléfono 1".</p> <p>5. Ingrese el segundo teléfono en el campo "Teléfono 2".</p> <p>6. Ingrese la dirección de la empresa en el campo "Dirección".</p> <p>7. Ingrese el fax de la empresa en el campo "Fax".</p> <p>8. Ingrese el país de procedencia de la empresa en el campo "País".</p> <p>9. Ingrese el código postal de la empresa en el campo "Cod. Postal".</p> <p>10. Ingrese la url de la empresa en el campo "URL".</p> <p>11. Ingrese una descripción en el campo "Descripción".</p> <p>12. Seleccione la opción "Grabar".</p>	<p>la empresa correctamente.</p> <p>El Sistema genera automáticamente un identificador que será único para esta empresa.</p>
Flujo C: Registrar un proyecto		
Paso 1	<p>Seleccione la opción "PROYECTOS" del menú principal.</p>	<p>El Sistema muestra la pantalla del listado de proyectos, con las siguientes secciones:</p> <p>Sección "Búsqueda de Proyectos" con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado <p>Sección "Lista de Proyectos":</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado
Paso 2	<p>Seleccione la opción "Nuevo Proyecto"</p>	<p>El sistema muestra la pantalla de ingreso del nuevo proyecto con los siguientes campos editables:</p> <ul style="list-style-type: none"> - Código - Nombre - Descripción - Presupuesto <p>Y los siguientes campos seleccionables:</p> <ul style="list-style-type: none"> - Empresa - Estado - Fecha Inicio - Fecha Fin - Moneda

		<ul style="list-style-type: none"> - Prioridad - Tipo - Jefe de Proyecto - Porcentaje de Avance - Metodología
Paso 3	<ol style="list-style-type: none"> 1. Ingrese un código en el campo "Código". 2. Ingrese un nombre en el campo "Nombre". 3. Seleccione una opción del campo Empresa. 4. Seleccione la opción "en Progreso" del campo "Estado". 5. Ingrese la Fecha Inicio y Fecha Fin del proyecto. 6. Seleccione la opción "Dólares Americanos" del campo "Moneda". 7. Ingrese una descripción del proyecto. 8. Seleccione la opción "Alta" del campo "Prioridad". 9. Seleccione la opción "Desarrollo de Software" del campo "Tipo". 10. Seleccione una opción del campo "Jefe de Proyecto". 11. Seleccione la opción "20%" del campo "% avance". 12. Ingrese el presupuesto de "1000", en el campo "Presupuesto". 13. Seleccione una opción del campo "Metodología". 14. Seleccione la opción "Grabar". 	<p>El sistema guarda los datos ingresados por el usuario correctamente.</p> <p>El Sistema genera automáticamente un identificador que será único para este proyecto.</p>
Flujo D: Asignar Personal al Proyecto registrado		
Paso 1	<p>Seleccione la opción "PROYECTOS" del menú principal.</p>	<p>El Sistema muestra la pantalla del listado de proyectos, con las siguientes secciones:</p> <p>Sección "Búsqueda de Proyectos" con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado <p>Sección "Lista de Proyectos":</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado.
Paso 2	<p>Seleccione la opción "Ver".</p>	<p>El sistema muestra los siguientes campos:</p> <p>Sección: "Datos del Proyecto"</p> <ul style="list-style-type: none"> - Código

		<ul style="list-style-type: none"> - Nombre - Fecha Inicio - Fecha Fin - Responsable - Empresa - Estado - Tipo - Avance - Descripción <p>Sección Opciones:</p> <ul style="list-style-type: none"> - Personal - Actividades - Artefactos - Gantt <p>Sección "Personal del Proyecto", se muestra los campos seleccionables:</p> <ul style="list-style-type: none"> - Persona - Rol <p>Sección "Lista del Personal del Proyecto", donde se muestra el listado del personal del proyecto en los campos:</p> <ul style="list-style-type: none"> - Nombre - Rol
Paso 3	<ol style="list-style-type: none"> 1. Seleccione alguna persona del campo "Persona" 2. Seleccione el rol que desempeñará la persona seleccionada del campo "Rol". 3. Seleccione la opción "Agregar". 	El sistema muestra en la sección "Lista del Personal del Proyecto" el registro del personal ingresado.
Flujo E: Agregar Actividades al Proyecto		
Paso 1	Seleccione la opción "Actividades"	El Sistema muestra la pantalla del listado de actividades asignadas al proyecto, con los siguientes datos: <ul style="list-style-type: none"> - Nombre Actividad - Personal -%Avance - Fecha Inicio - Fecha Fin
Paso 2	Seleccione la opción "Nueva Actividad".	El sistema muestra la pantalla para el registro de la nueva actividad con los siguientes campos editables: <ul style="list-style-type: none"> - Nombre Actividad - Descripción - Horas Esperadas <p>Y los siguientes campos seleccionables:</p> <ul style="list-style-type: none"> - Estado - Responsable - % Avance - Fecha Inicio - Tipo - Prioridad - Actividad Padre

		<ul style="list-style-type: none"> - Fecha Fin - Actividades Metodología
Paso 3	<ol style="list-style-type: none"> 1. Ingrese el nombre actividad en el campo "Nombre Actividad". 2. Seleccione la opción "en Progreso" del campo "Estado". 3. Seleccione alguna opción del campo "Responsable". 4. Ingrese el porcentaje de avance en el campo "% Avance". 5. Seleccione la fecha de inicio del campo "Fecha Inicio". 6. Ingrese la descripción de la actividad en el campo "Descripción". 7. Seleccione la opción "Documentación" del campo "Tipo". 8. Seleccione la opción "Media" del campo "Prioridad". 9. Seleccione una opción del campo "Actividad Padre". 10. Ingrese el número de horas en el campo "Horas Esperadas". 11. Ingrese la fecha de fin del campo "Fecha Fin". 12. Selecciona una o varias actividades de la metodología del campo "Actividades Metodología". 13. Seleccione la opción "Grabar". 	<p>El sistema muestra la pantalla del listado de las actividades del proyecto, donde se encuentra la actividad registrada.</p>
Flujo F: Agregar Artefactos al Proyecto		
Paso 1	<p>Seleccione la opción "Artefactos"</p>	<p>El Sistema muestra la pantalla del listado de artefactos asignadas al proyecto:</p> <p>Sección "Datos del Proyecto":</p> <ul style="list-style-type: none"> - Código - Fecha Inicio - Responsable - Estado - Avance - Nombre - Fecha Fin - Empresa - Tipo - Descripción. <p>Sección "Opciones del Proyecto":</p> <ul style="list-style-type: none"> - Personal - Actividades - Artefactos - Gantt <p>Sección "Listado Artefactos":</p> <ul style="list-style-type: none"> - Nombre - Versión - Actividades - Estado - UsuarioReserva - Fec. Reserva

		<ul style="list-style-type: none"> - Versiones - Descargar - Reservar - Liberar - Subir
Paso 2	<p>Seleccione la opción “Nuevo Artefacto”.</p>	<p>El sistema muestra la pantalla para el registro del nuevo artefacto para el proyecto registrado con los siguientes campos editables:</p> <ul style="list-style-type: none"> - Nombre Artefacto - Descripción <p>Y los siguientes campos seleccionables:</p> <ul style="list-style-type: none"> - Archivo - Actividades - Artefactos Metodología
Paso 3	<ol style="list-style-type: none"> 1. Ingrese el nombre artefacto en el campo “Nombre Artefacto”. 2. Seleccione la opción “Examinar” del campo “Archivo” y busque el archivo a cargar. 3. Ingrese una descripción en el campo “Descripción”. 4. Seleccione un (as) actividades listadas del campo “Actividades”. 5. Seleccione un (os) artefactos del campo “Artefactos Metodología”. 6. Seleccione la opción “Grabar”. 	<p>El sistema muestra la pantalla del listado de los artefactos del proyecto, donde se encuentra el artefacto registrado.</p>
<p>Flujo G: Ingresar Horas Trabajadas a las actividades</p>		
Paso 1	<p>Seleccione la opción “MIS ACTIVIDADES” del menú principal.</p>	<p>El Sistema muestra la pantalla de Actividades x Hora, con las siguientes secciones:</p> <p>Sección “Búsqueda de Actividades” con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Proyecto - Estado - Semáforo <p>Sección “Lista de Actividades”</p> <ul style="list-style-type: none"> - Proyecto - Actividad - Estado - Inicio - Fin - Horas
Paso 2	<p>Seleccione la opción “Ver horas” de la actividad ingresada.</p>	<p>El Sistema muestra la pantalla de Tiempos a Ingresar, con las siguientes secciones:</p> <p>Sección Tiempos a Ingresar</p> <ul style="list-style-type: none"> - Actividad - Fecha - Horas <p>Sección “Horas Ingresadas”</p> <ul style="list-style-type: none"> - Fecha - Horas
Paso 3	<ol style="list-style-type: none"> 1. Ingrese la fecha en que se realiza la actividad. 	<p>El Sistema muestra en la pantalla de Tiempos a Ingresar la fecha y horas</p>

	2. Ingresa el total de horas generadas. 3. Seleccione la opción "Grabar".	registradas para esta actividad.
Flujo H: Mostrar Diagrama de Gantt		
Paso 1	Seleccione la opción "Gantt"	El Sistema muestra el Diagrama de Gantt donde se muestran las actividades según su distribución de acuerdo al calendario, de manera tal que se pueda visualizar: <ul style="list-style-type: none"> - Período de duración de cada actividad - Fecha de inicio y término. - Tiempo total requerido para la ejecución de una actividad determinada.

3.9.2. Reservar Artefactos, cambiar y subir versión

Nombre Paso	Descripción	Resultados Esperados
Flujo A: Reservar artefacto por un determinado usuario		
Paso1	1. Ingrese al aplicativo EACS Project 2. Ingresar el valor de "cgonzales" en el campo "Usuario". 3. Ingresar el valor de la contraseña en el campo "Contraseña". 4. Seleccionar la opción "Ingresar".	El Sistema inicia sesión y se muestra la pantalla del listado de proyectos, con las siguientes secciones: Sección "Búsqueda de Proyectos" con los criterios de búsqueda: <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado Sección "Lista de Proyectos": <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado
Paso 2	Seleccione del proyecto la opción "Ver".	El sistema muestra la pantalla de proyectos los siguientes campos: Sección: "Datos del Proyecto" <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Fecha Fin - Responsable - Empresa - Estado - Tipo - Avance - Descripción Sección Opciones: <ul style="list-style-type: none"> - Personal - Actividades - Artefactos - Gantt Sección "Personal del Proyecto", se muestra los campos seleccionables:

		<ul style="list-style-type: none"> - Persona - Rol <p>Sección “Lista del Personal del Proyecto”, donde se muestra el listado del personal del proyecto en los campos:</p> <ul style="list-style-type: none"> - Nombre - Rol
Paso 3	<p>Seleccione la opción “Artefactos” de la sección “Opciones”</p>	<p>El sistema muestra la pantalla del listado de los artefactos del proyecto, con las siguientes secciones:</p> <p>Sección: “Datos del Proyecto”</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Fecha Fin - Responsable - Empresa - Estado - Tipo - Avance - Descripción <p>Sección Opciones:</p> <ul style="list-style-type: none"> - Personal - Actividades - Artefactos - Gantt <p>Sección “Lista de Artefactos”</p> <ul style="list-style-type: none"> - Nombre - Versión - Actividades - Estado - Usuario Reserva - Fec. Reserva - Versiones - Descargar - Reservar - Liberar - Subir
Paso 4	<p>Seleccione la opción “reservar”.</p>	<p>El Sistema bloquea el artefacto reservado y se habilita las opciones:</p> <ul style="list-style-type: none"> - Liberar - Subir <p>Deshabilita la opción:</p> <ul style="list-style-type: none"> - Reservar <p>El sistema actualiza los siguientes campos:</p> <ul style="list-style-type: none"> - Estado del artefacto a reservado. - UsuarioReserva. - Fec. Reserva
<p>Flujo B: Ingreso al sistema con usuario diferente al que reservó el artefacto</p>		
Paso1	<ol style="list-style-type: none"> 1. Ingrese al aplicativo EACS Project 2. Ingresar el valor “eocampo” en el campo “Usuario”: 3. Ingresar el valor de la 	<p>El Sistema inicia sesión y se muestra la pantalla del listado de proyectos, con las siguientes secciones:</p> <p>Sección “Búsqueda de Proyectos” con los</p>

	<p>contraseña en el campo "Contraseña".</p> <p>4. Seleccionar la opción "Ingresar".</p>	<p>critérios de búsqueda:</p> <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado <p>Sección "Lista de Proyectos":</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado
Paso 2	<p>Seleccione del proyecto la opción "ver".</p>	<p>El sistema muestra la pantalla de proyectos los siguientes campos:</p> <p>Sección: "Datos del Proyecto"</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Fecha Fin - Responsable - Empresa - Estado - Tipo - Avance - Descripción <p>Sección Opciones:</p> <ul style="list-style-type: none"> - Personal - Actividades - Artefactos - Gantt <p>Sección "Personal del Proyecto", se muestra los campos seleccionables:</p> <ul style="list-style-type: none"> - Persona - Rol <p>Sección "Lista del Personal del Proyecto", donde se muestra el listado del personal del proyecto en los campos:</p> <ul style="list-style-type: none"> - Nombre - Rol
Paso 3	<p>Seleccione la opción "Artefactos" de la sección "Opciones".</p>	<p>El sistema muestra la pantalla del listado de los artefactos del proyecto.</p> <p>El Sistema muestra el artefacto con los campos:</p> <ul style="list-style-type: none"> - Estado = Reservado - UsuarioReserva = Nombre del usuario que lo reservó. - Fec. Reserva = Fecha y hora de reserva <p>Las opciones habilitadas:</p> <ul style="list-style-type: none"> - Versiones - Descargar <p>Las opciones deshabilitadas:</p> <ul style="list-style-type: none"> - Reservar

		<ul style="list-style-type: none"> - Liberar - Subir
Flujo C: Subir versión del artefacto		
Paso1	<ol style="list-style-type: none"> 1. Ingrese al aplicativo EACS Project 2. Ingresar el valor "cgonzales" en el campo "Usuario": 3. Ingresar el valor de la contraseña en el campo "Contraseña". 4. Seleccionar la opción "Ingresar". 	<p>El Sistema inicia sesión y se muestra la pantalla del listado de proyectos, con las siguientes secciones:</p> <p>Sección "Búsqueda de Proyectos" con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado <p>Sección "Lista de Proyectos":</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado
Paso 2	<p>Seleccione del proyecto la opción "ver".</p>	<p>El sistema muestra la pantalla de proyectos los siguientes campos:</p> <p>Sección: "Datos del Proyecto"</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Fecha Fin - Responsable - Empresa - Estado - Tipo - Avance - Descripción <p>Sección Opciones:</p> <ul style="list-style-type: none"> - Personal - Actividades - Artefactos - Gantt <p>Sección "Personal del Proyecto", se muestra los campos seleccionables:</p> <ul style="list-style-type: none"> - Persona - Rol <p>Sección "Lista del Personal del Proyecto", donde se muestra el listado del personal del proyecto en los campos:</p> <ul style="list-style-type: none"> - Nombre - Rol
Paso 3	<p>Seleccione la opción "Artefactos" de la sección "Opciones".</p>	<p>El sistema muestra la pantalla del listado de los artefactos del proyecto.</p> <p>El Sistema muestra el artefacto con los campos:</p> <ul style="list-style-type: none"> - Estado = Reservado - UsuarioReserva = Nombre del usuario que lo reservó.

		<p>- Fec. Reserva = Fecha y hora de reserva</p> <p>Las opciones habilitadas:</p> <ul style="list-style-type: none"> - Versiones - Descargar - Liberar - Subir
Paso 4	1. Seleccione la opción "Subir".	<p>El Sistema muestra la pantalla de Subir Versión con los siguientes campos:</p> <ul style="list-style-type: none"> - Archivo - Observaciones
Paso 5	<p>1. Ingrese la ruta donde se ubica el artefacto en el campo "Archivo".</p> <p>2. Ingrese las observaciones en el campo "Observaciones".</p> <p>3. Seleccione la opción "Grabar".</p>	<p>El Sistema muestra la pantalla del listado de artefactos donde se observa que se genera una nueva versión del artefacto seleccionado.</p> <p>El Sistema cambia de estado el artefacto así puede ser modificado por otros usuarios y se registran los datos del usuario que generó la versión y fecha.</p>
Flujo D: Cambiar versión del artefacto		
Paso1	<p>1. Ingrese al aplicativo EACS Project</p> <p>2. Ingresar el valor "cgonzales" en el campo "Usuario":</p> <p>3. Ingresar el valor de la contraseña en el campo "Contraseña".</p> <p>4. Seleccionar la opción "Ingresar".</p>	<p>El Sistema inicia sesión y se muestra la pantalla del listado de proyectos, con las siguientes secciones:</p> <p>Sección "Búsqueda de Proyectos" con los criterios de búsqueda:</p> <ul style="list-style-type: none"> - Responsable - Empresa - TipoProyecto - Estado <p>Sección "Lista de Proyectos":</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Responsable - Empresa - Tipo - Estado.
Paso 2	Seleccione del proyecto la opción "ver".	<p>El sistema muestra la pantalla de proyectos los siguientes campos:</p> <p>Sección: "Datos del Proyecto"</p> <ul style="list-style-type: none"> - Código - Nombre - Fecha Inicio - Fecha Fin - Responsable - Empresa - Estado - Tipo - Avance - Descripción <p>Sección Opciones:</p> <ul style="list-style-type: none"> - Personal - Actividades - Artefactos

		<ul style="list-style-type: none"> - Gantt <p>Sección “Personal del Proyecto”, se muestra los campos seleccionables:</p> <ul style="list-style-type: none"> - Persona - Rol <p>Sección “Lista del Personal del Proyecto”, donde se muestra el listado del personal del proyecto en los campos:</p> <ul style="list-style-type: none"> - Nombre - Rol
Paso 3	<p>Seleccione la opción “Artefactos” de la sección “Opciones”.</p>	<p>El sistema muestra la pantalla del listado de los artefactos del proyecto.</p> <p>El Sistema muestra el artefacto con los campos:</p> <ul style="list-style-type: none"> - Estado = Reservado - UsuarioReserva = Nombre del usuario que lo reservó. - Fec. Reserva = Fecha y hora de reserva <p>Las opciones habilitadas:</p> <ul style="list-style-type: none"> - Versiones - Descargar - Liberar - Subir
Paso 4	<p>Seleccione la opción “Versiones”</p>	<p>El Sistema muestra la pantalla del listado de versiones, con el campo seleccionable Versión Vigente.</p>
Paso 5	<ol style="list-style-type: none"> 1. Seleccione una versión del listado del campo “Versión Vigente” 2. Seleccione la opción “Cambiar”. 	<p>El Sistema muestra la pantalla del listado de artefactos con el artefacto seleccionado y la versión vigente ingresada.</p>

3.10. Flujo de Eventos

Con la finalidad de probar el cumplimiento de los requerimientos establecidos para la herramienta EACS Project Manager se eligió el modelado de los procesos principales.

3.10.1. Proceso de Administración de proyectos.

La herramienta EACS Project Manager consta de un módulo de administración de proyectos, el cual tiene como propósito la planificación, realización, evaluación y control de todos los proyectos. Se ocupa de los proyectos internos y externos de la organización.

A continuación se describe el flujo de eventos respectivo para la opción “Proyectos”. Se describe tanto el ingreso al sistema, en el que está incluida la validación al ingreso, así como la creación y modificación de un determinado proyecto creado en el sistema.

1. El usuario ingresa su usuario y contraseña y elige la opción “Ingresar”, como se muestra en la Figura 3.1, el sistema realiza la verificación de los datos de acceso para confirmar que el usuario tiene los permisos para el ingreso al sistema, caso contrario se muestra un mensaje de error,



Figura 3.1. Pantalla Ingresar al Sistema.

2. Por defecto el sistema muestra la pantalla con el listado de proyectos, en el menú superior se encuentran las opciones para ver los proyectos, mis actividades, usuarios, empresas y reportes. Para ingresar a dar mantenimiento a los proyectos se debe seleccionar la opción de “Proyectos”, como se muestra en la Figura 3.2.

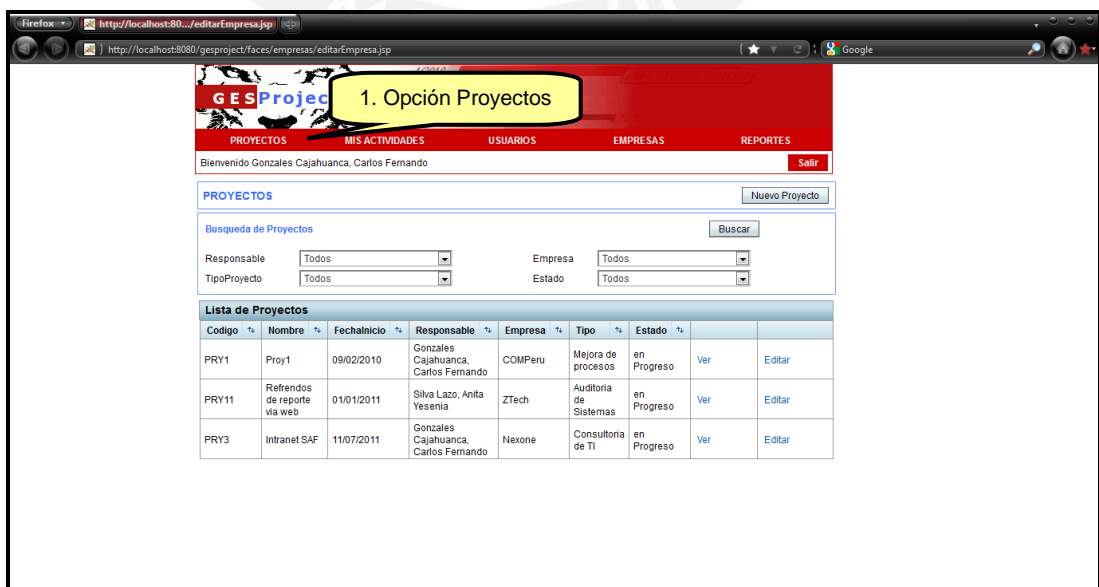


Figura 3.2. Pantalla Inicial.

3. La Figura 3.3 muestra la pantalla “Mantener Proyectos” con el listado de los proyectos y las opciones para “Buscar”, “Nuevo Proyecto”, “Ver” y “Editar” proyectos.

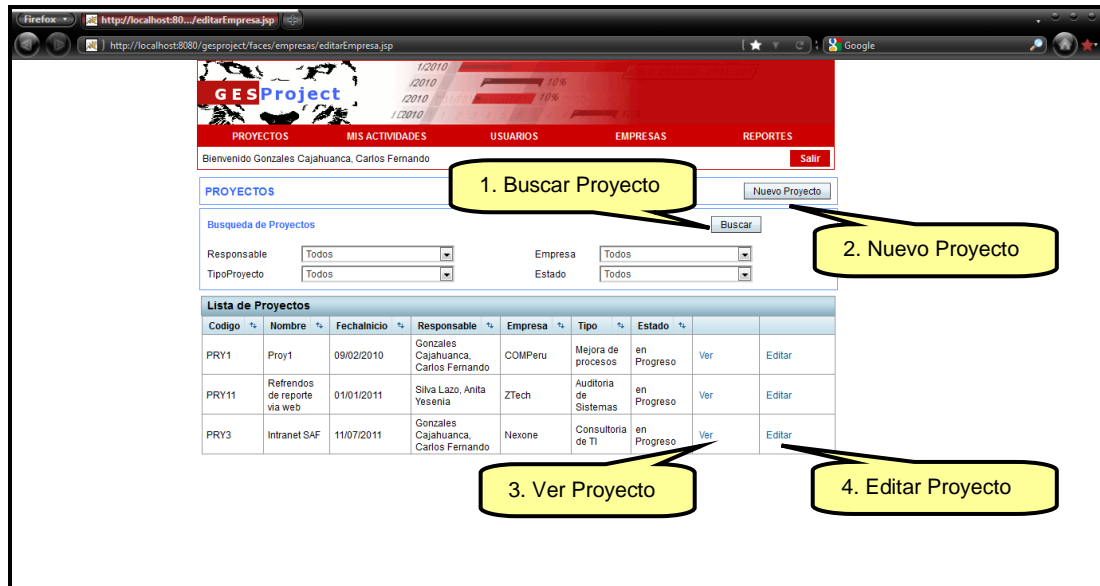


Figura 3.3. Pantalla Mantener Proyectos.

4. Si se desea ingresar un nuevo proyecto dentro del sistema, se presiona el botón “Nuevo Proyecto” y el sistema muestra la pantalla de la Figura 3.4.

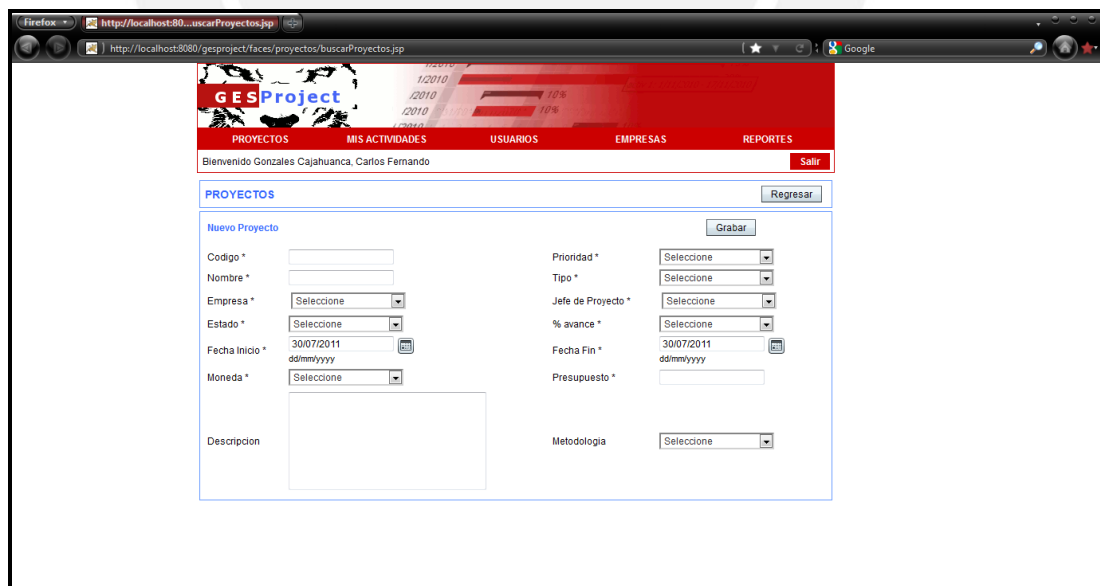


Figura 3.4. Pantalla Nuevo Proyecto.

5. Se ingresa los datos solicitados por el sistema en la pantalla, además se puede relacionar el proyecto creado con una metodología definida en un proceso, como se muestra en la Figura 3.5. Para grabar la información ingresada se debe elegir la opción “Grabar” o para cancelar el registro del proyecto se elige “Regresar” el cual lleva a la pantalla “Mantener Proyectos”.

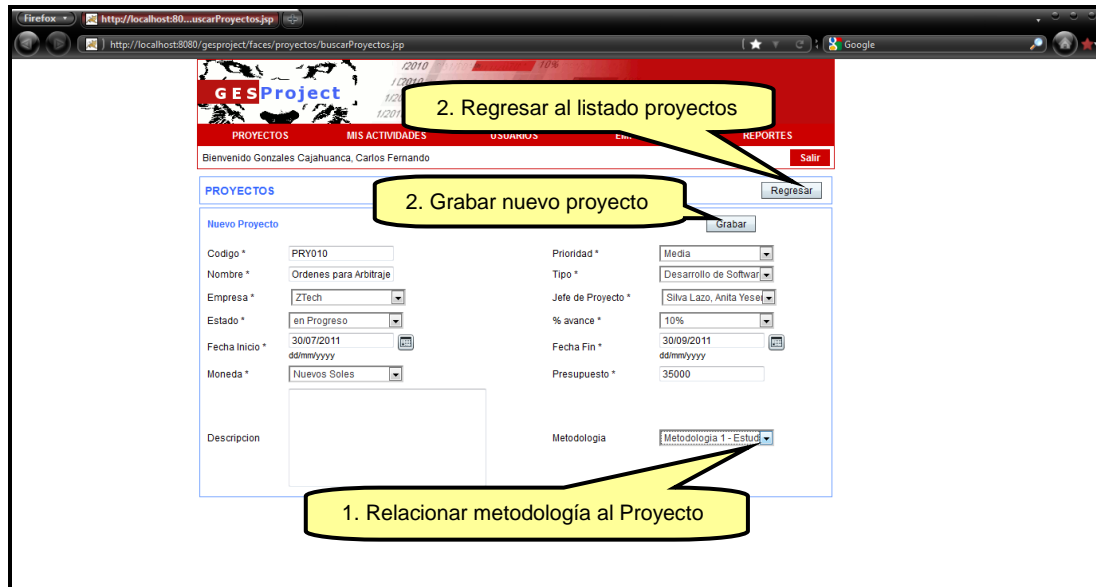


Figura 3.5. Pantalla grabar Nuevo Proyecto.

6. Para realizar la búsqueda de los proyectos se debe elegir los parámetros de búsqueda y luego elegir la opción “Buscar”, tal como se muestra en la Figura 3.6. Además por cada proyecto se puede utilizar las opciones “Editar” y “Ver” que permiten editar los datos registrados del proyecto así como ver el proyecto completo, sus usuarios, actividades y artefactos.

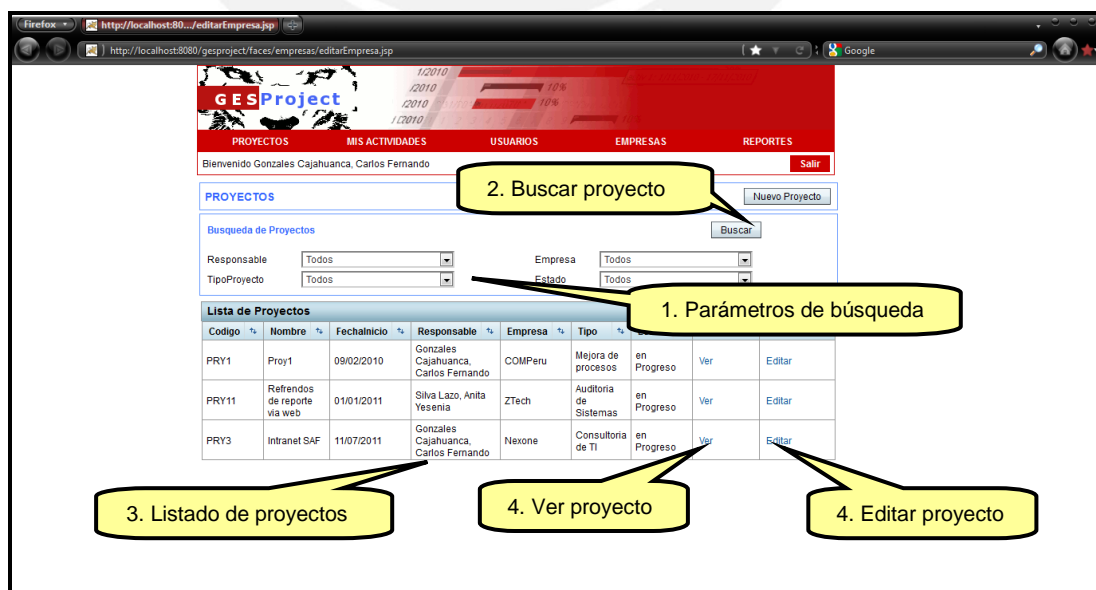


Figura 3.6. Pantalla Buscar Proyectos.

- Si se ha seleccionado el botón “Editar”, se mostrará el formulario del proyecto con los datos registrados previamente, como se muestra en la Figura 3.7.
Se actualiza los datos que se desea modificar y se presiona el botón “Grabar” o se presiona el botón “Regresar” si se desea mantener los datos actuales.

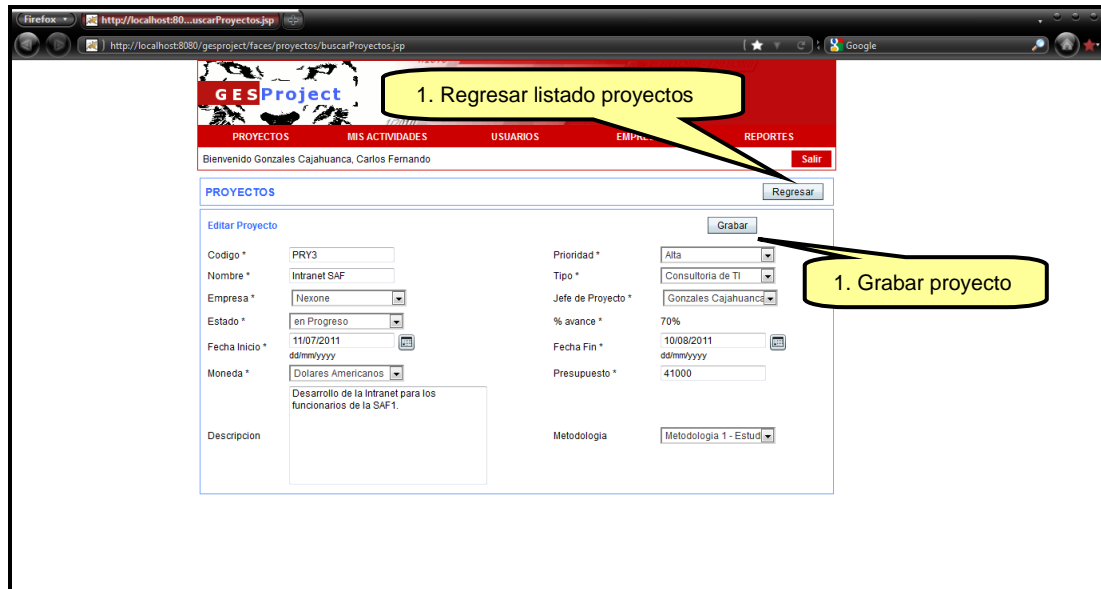


Figura 3.7. Pantalla Editar Proyecto.

Flujo de eventos respectivo para la opción “Mis Actividades”

- Se ingresa al sistema con un usuario válido una vez allí, se debe ingresar a la opción del menú “Mis Actividades”, como se muestra en la Figura 3.8.



Figura 3.8. Pantalla Mis Actividades.

- La pantalla “Buscar Mis Actividades” tiene la opción “Buscar” para realizar la búsqueda de actividades tal como se muestra en la Figura 3.9, además por cada actividad se puede visualizar e ingresar las horas trabajadas en la opción “Ver horas”.



Figura 3.9. Pantalla Buscar Mis Actividades.

- Si se desea ingresar las horas de esfuerzo de la actividad seleccionada en la pantalla de “Mis actividades”, se presiona el botón “Ver horas”, y se ingresan los datos como se muestra en la Figura 3.10.

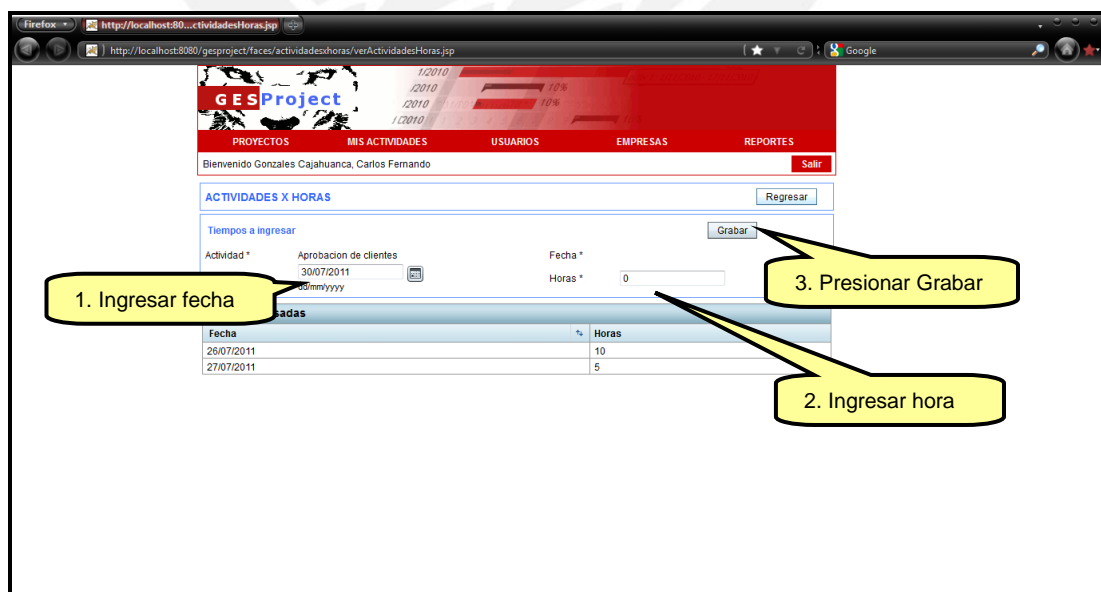


Figura 3.10. Pantalla Ingreso de Horas trabajadas.

Flujo de eventos respectivo para la opción “Usuarios”.

1. Se ingresa al sistema con un usuario válido una vez allí, se debe ingresar a la opción de “Usuarios”, como se muestra en la Figura 3.11 también se muestran las opciones para crear “Nuevo usuario”, “Buscar” y “Editar” usuarios.

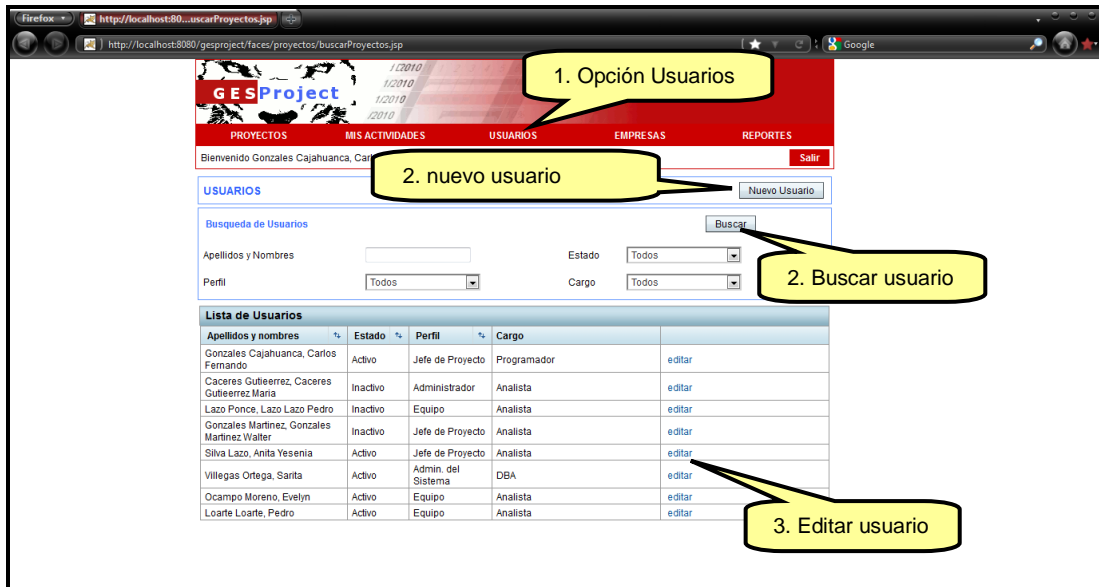


Figura 3.11. Pantalla Mantener Usuarios.

2. Si se desea ingresar un nuevo usuario dentro del sistema, se presiona el botón “Nuevo Usuario” y el sistema muestra la pantalla como la Figura 3.12.

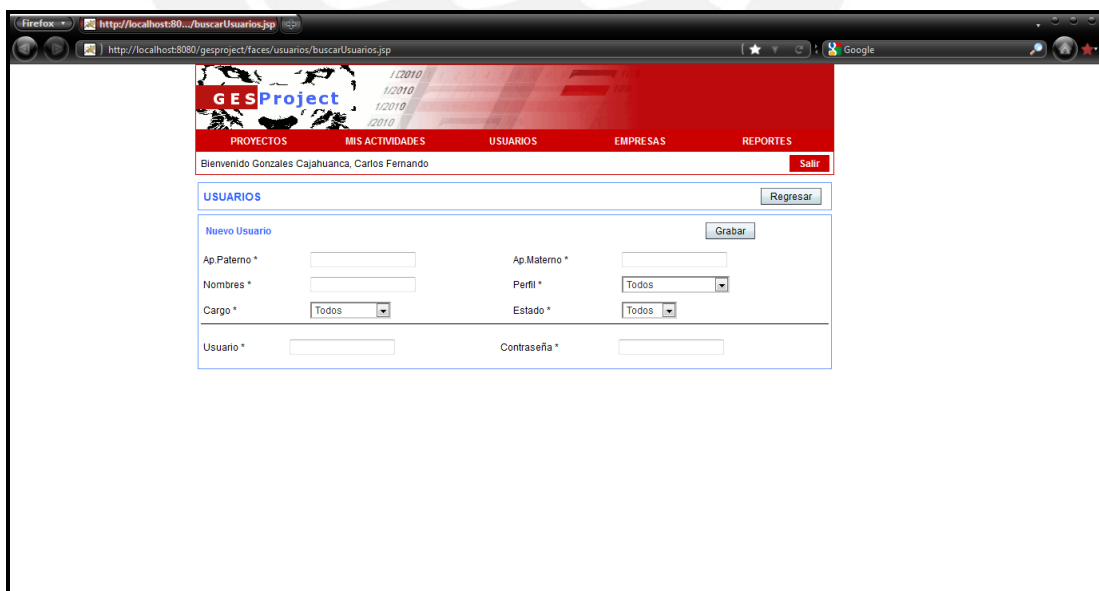


Figura 3.12. Pantalla Nuevo Usuario.

- Luego de ingresar los datos solicitados por el sistema en la pantalla se presiona el botón “Grabar” o para cancelar la acción de registro del usuario se presiona el botón “Regresar”, como se muestra en la Figura 3.13.



Figura 3.13. Pantalla grabar Nuevo Usuario.

- En la pantalla “Buscar Usuarios”, donde se puede realizar la búsqueda de los usuarios al presionar el botón “Buscar”, como se muestra en la Figura 3.14.

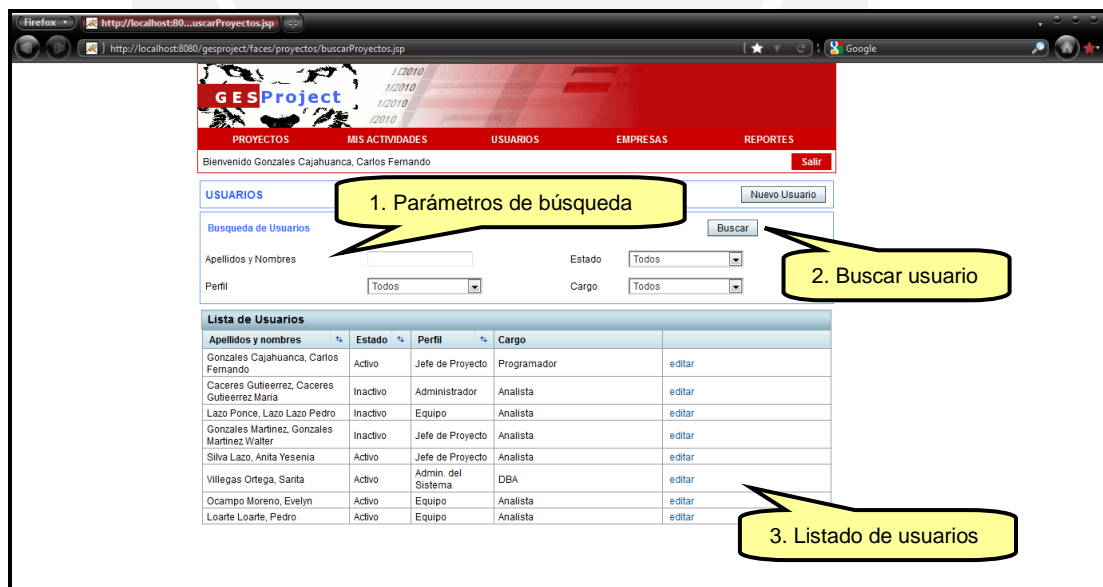


Figura 3.14. Pantalla Mantenimiento Usuarios.

5. La Figura 3.15. muestra el formulario del Usuario con los datos registrados en el sistema. Se actualiza los datos que se desea modificar para registrar los cambios se debe elegir la opción “Grabar” o “Regresar” si se desea mantener los datos actuales.

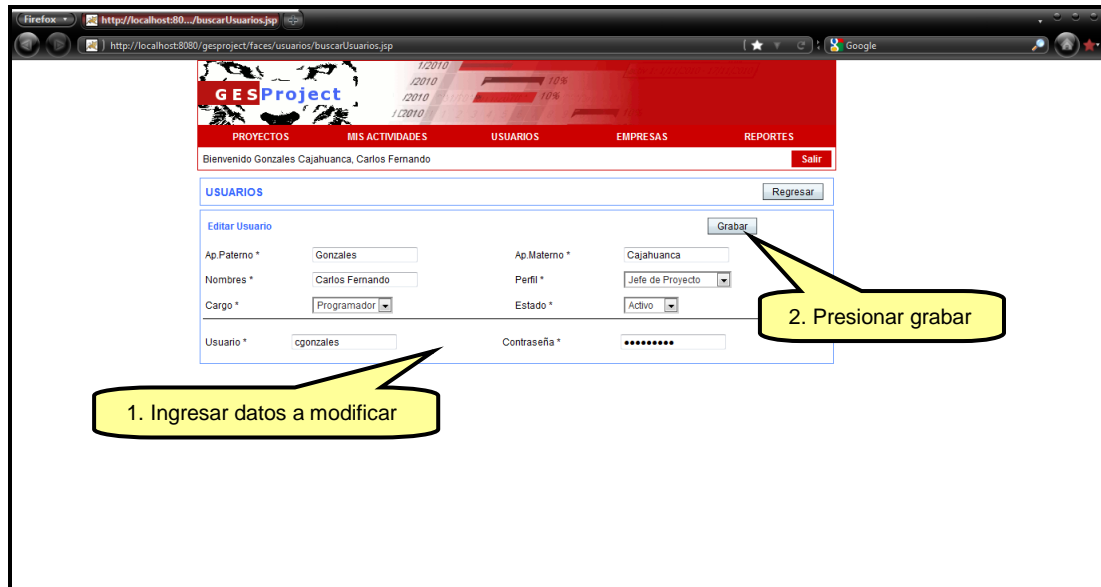


Figura 3.15. Pantalla Editar Usuario.

Flujo de eventos respectivo para la opción “Empresas”.

1. Se ingresa al menú del sistema con un usuario válido una vez allí, ingrese a la opción de “Empresas”, como se muestra en la Figura 3.16 también se cuenta con las opciones “Nueva empresa”, “Buscar” y “Editar” empresas.



Figura 3.16. Pantalla Mantener Empresas.

- Si se desea ingresar una nueva empresa dentro del sistema, se elige la opción “Nueva Empresa” tal como se muestra en la Figura 3.17.

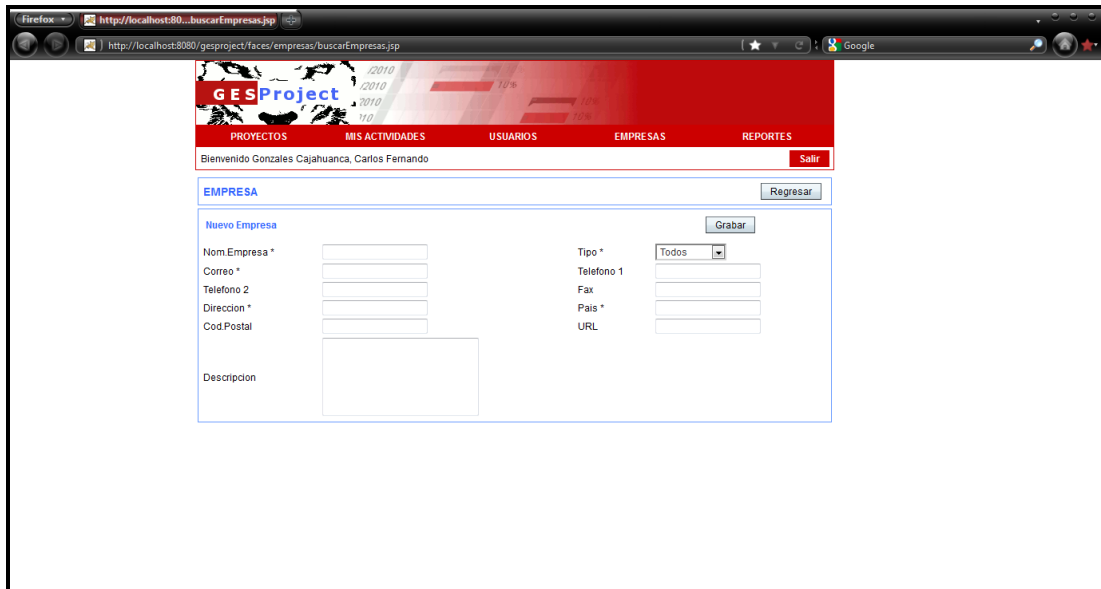


Figura 3.17. Pantalla Nueva Empresa.

- Se ingresa los datos solicitados por el sistema en el formulario. Para grabar la información ingresada se elige la opción “Grabar” o “Regresar” para cancelar la acción de registro de la empresa, como se muestra en la Figura 3.18.



Figura 3.18. Pantalla grabar Nueva Empresa.

4. Puede realizarse búsquedas de empresas utilizando los filtros de búsqueda y luego eligiendo la opción “Buscar”, como se muestra en la Figura 3.19. Además por cada empresa registrada se puede utilizar la opción “Editar” para editar los datos de la empresa.



Figura 3.19. Pantalla Buscar Empresas.

5. La Figura 3.20 muestra el formulario de la empresa pero con los datos registrados previamente. Se actualiza los datos que se desea modificar y se elige la opción “Grabar” o “Regresar” si se desea mantener los datos actuales.



Figura 3.20. Pantalla Edición Empresas.

Flujo de eventos respectivo para la opción “Reportes”.

1. Se ingresa al sistema con un usuario válido una vez allí, ingrese a la opción “Reportes”, como se muestra en la Figura 3.21. El sistema muestra la pantalla de “Listado Reportes”, los tipos de reportes que se pueden visualizar son: Reporte Detallado del Proyecto y Reportes de Esfuerzo.

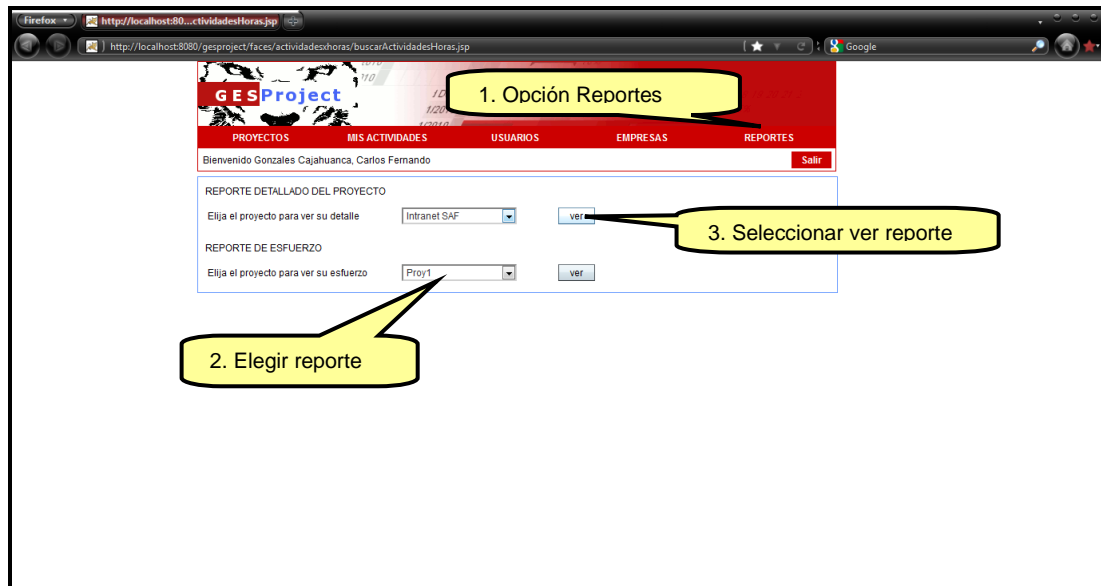


Figura 3.21. Pantalla Listado Reportes.

2. Para visualizar el reporte “Reporte Detallado del Proyecto”, se debe elegir el reporte y luego elegir la opción “Ver”, como se muestra en la Figura 3.22. y el sistema muestra el reporte en formato HTML.

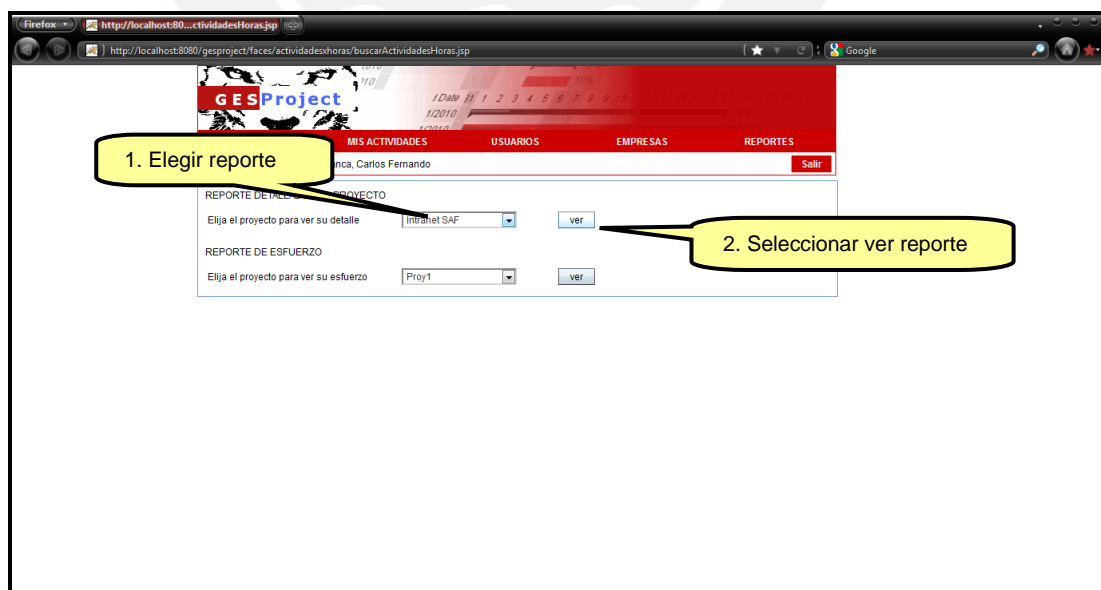
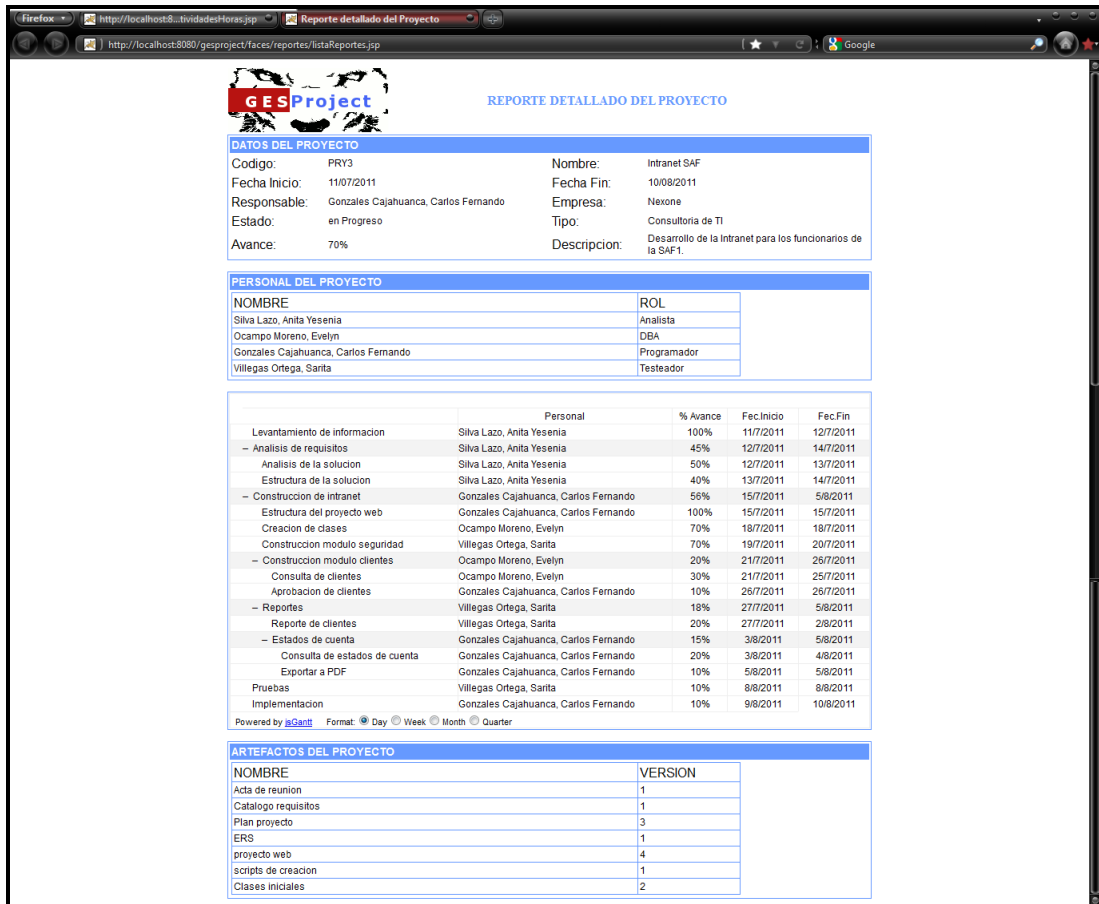


Figura 3.22. Pantalla elegir Reporte Detallado.

El reporte tiene tres secciones, como se muestra en la Figura 3.23.



REPORTE DETALLADO DEL PROYECTO

DATOS DEL PROYECTO

Codigo:	PRV3	Nombre:	Intranet SAF
Fecha Inicio:	11/07/2011	Fecha Fin:	10/08/2011
Responsable:	Gonzales Cajahuana, Carlos Fernando	Empresa:	Nexone
Estado:	en Progreso	Tipo:	Consultoria de TI
Avance:	70%	Descripcion:	Desarrollo de la Intranet para los funcionarios de la SAF1.

PERSONAL DEL PROYECTO

NOMBRE	ROL
Silva Lazo, Anita Yesenia	Analista
Ocampo Moreno, Evelyn	DBA
Gonzales Cajahuana, Carlos Fernando	Programador
Villegas Ortega, Sarita	Testeador

	Personal	% Avance	Fec.Inicio	Fec.Fin
Levantamiento de informacion	Silva Lazo, Anita Yesenia	100%	11/7/2011	12/7/2011
- Analisis de requisitos	Silva Lazo, Anita Yesenia	45%	12/7/2011	14/7/2011
- Analisis de la solucion	Silva Lazo, Anita Yesenia	50%	12/7/2011	13/7/2011
Estructura de la solucion	Silva Lazo, Anita Yesenia	40%	13/7/2011	14/7/2011
- Construccion de intranet	Gonzales Cajahuana, Carlos Fernando	56%	15/7/2011	5/8/2011
Estructura del proyecto web	Gonzales Cajahuana, Carlos Fernando	100%	15/7/2011	15/7/2011
Creacion de clases	Ocampo Moreno, Evelyn	70%	18/7/2011	18/7/2011
Construccion modulo seguridad	Villegas Ortega, Sarita	70%	19/7/2011	20/7/2011
- Construccion modulo clientes	Ocampo Moreno, Evelyn	20%	21/7/2011	26/7/2011
Consulta de clientes	Ocampo Moreno, Evelyn	30%	21/7/2011	25/7/2011
Aprobacion de clientes	Gonzales Cajahuana, Carlos Fernando	10%	26/7/2011	26/7/2011
- Reportes	Villegas Ortega, Sarita	18%	27/7/2011	5/8/2011
Reporte de clientes	Villegas Ortega, Sarita	20%	27/7/2011	2/8/2011
- Estados de cuenta	Gonzales Cajahuana, Carlos Fernando	15%	3/8/2011	5/8/2011
Consulta de estados de cuenta	Gonzales Cajahuana, Carlos Fernando	20%	3/8/2011	4/8/2011
Exportar a PDF	Gonzales Cajahuana, Carlos Fernando	10%	5/8/2011	5/8/2011
Pruebas	Villegas Ortega, Sarita	10%	8/8/2011	8/8/2011
Implementacion	Gonzales Cajahuana, Carlos Fernando	10%	9/8/2011	10/8/2011

Powered by [ASanti](#) Format: Day Week Month Quarter

ARTEFACTOS DEL PROYECTO

NOMBRE	VERSION
Acta de reunion	1
Catalogo requisitos	1
Plan proyecto	3
ERS	1
proyecto web	4
scripts de creacion	1
Clases iniciales	2

Figura 3.23. Pantalla Reporte Detallado del Proyecto.

Datos de Proyecto: sección del reporte “Reporte Detallado del Proyecto”, donde se muestra el código proyecto, nombre proyecto, fecha inicio, fecha fin, responsable del proyecto, empresa, estado proyecto, tipo proyecto, porcentaje de avance y descripción.

Personal del proyecto: sección del reporte “Reporte Detallado del Proyecto”, donde se muestra el nombre y rol del personal que forma parte del proyecto.

También se visualiza las actividades y sub actividades que tiene el proyecto, así como el personal a cargo de cada una de las actividades, el porcentaje de avance, la fecha inicio y fecha fin de la actividad.

Artefactos del Proyecto: sección del reporte “Reporte Detallado del Proyecto”, donde se muestra todos los artefactos que tiene el proyecto, así como la versión vigente del artefacto.

3. Para visualizar el reporte “Reporte de Esfuerzo” se debe elegir el reporte y la opción “Ver”, tal como se muestra en la Figura 3.24.



Figura 3.24. Pantalla elegir Reporte de Esfuerzo.

Este reporte mostrará el esfuerzo realizado por los usuarios para cumplir con sus actividades designadas, tal como se muestra en la Figura 3.25 este reporte presenta las siguientes secciones.

Datos del proyecto: Muestra los datos básicos del proyecto seleccionado para el “Reporte de Esfuerzo del Proyecto”: código, nombre, fecha inicio, fecha fin, empresa, estado y porcentaje de avance.



Figura 3.25. Pantalla Reporte de esfuerzo del Proyecto.

Listado de actividades del proyecto: sección del reporte “Reporte de Esfuerzo del Proyecto” que muestra el listado de actividades que tiene el proyecto elegido.

Para cada actividad se muestran los datos principales de ésta como: fecha inicio, fecha fin, porcentaje de avance, responsable de la actividad, horas estimadas y horas reales, estas se calculan en base a las horas reales ingresadas por los mismos responsables del proyecto.

También se listan a nivel de detalle los usuarios que participaron a lo largo de la vida de la actividad así como las horas que trabajaron en ésta, estas son en realidad un resumen y muestran el total de horas trabajadas por usuario, ya que en realidad los usuarios ingresan las horas trabajadas día a día.

3.10.2. Proceso de Administración de proyectos específicos.

La herramienta EACS Project Manager consta de un módulo de administración de proyectos específicos, el cual tiene como propósito configurar y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costos esperados mediante la coordinación y el manejo de los recursos del mismo, además de llevar el control de la configuración del proyecto y sus artefactos. A continuación se describe el flujo de eventos respectivo para la opción “Personal” que se encuentra involucrado en el proyecto y que posteriormente podrá ser utilizado para asignarlos a las actividades creadas.

1. Para visualizar el personal asignado al proyecto, el usuario debe seleccionar la opción “Personal”, como se muestra en la Figura 3.26. También se puede agregar nuevo personal y editarlo con las opciones “Agregar” y “Editar”.



Figura 3.26. Pantalla Mantener Personal del Proyecto.

- Si se desea ingresar personal al proyecto debe seleccionar el personal y rol que desempeñará la persona seleccionada para el proyecto elegido y seleccionar la opción “Agregar”, como se muestra en la Figura 3.27.

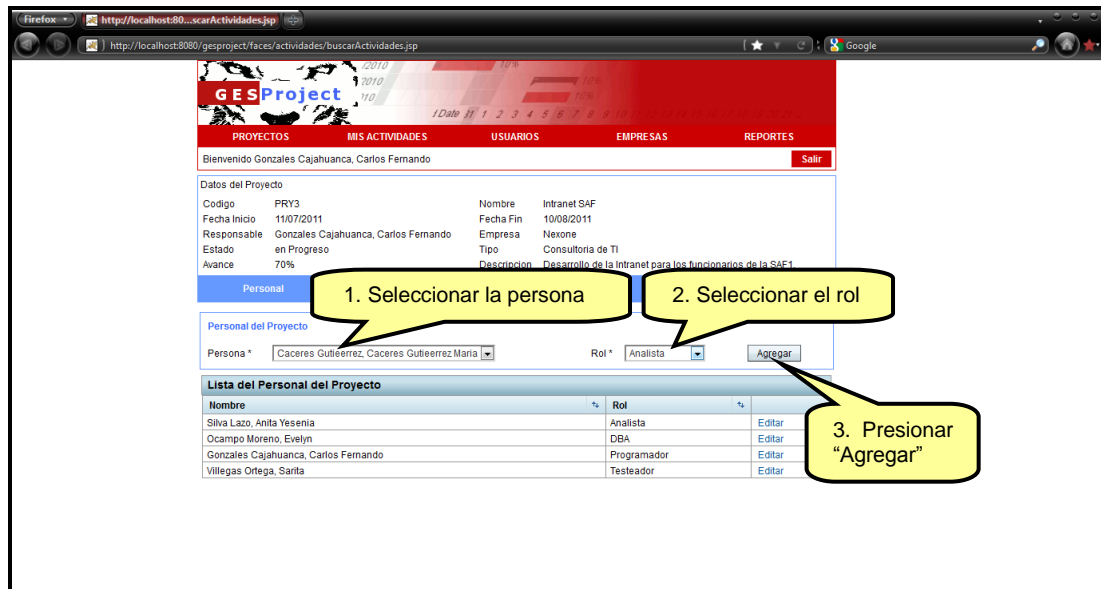


Figura 3.27. Pantalla Agregar Personal al Proyecto.

- Si se desea modificar los datos de algún registro del listado mostrado, se elige la opción “Editar”, como se muestra en la Figura 3.28.

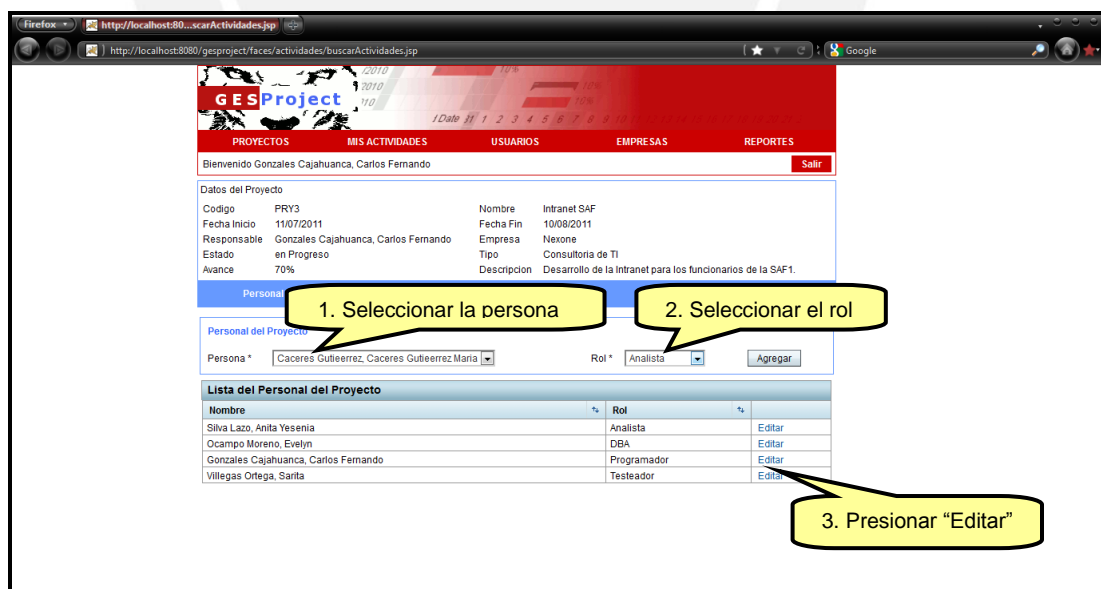


Figura 3.28. Pantalla Editar Personal del Proyecto.

Flujo de eventos respectivo para la opción “Actividades”.

1. Para realizar el mantenimiento de una actividad perteneciente al proyecto registrado, el usuario debe seleccionar la opción “Actividades”, donde se listan los registros existentes en el sistema, como se muestra en la Figura 3.29. También se muestran las opciones “Nueva actividad”, “Editar” y “Ver Artefactos” que permiten crear nuevas actividades, editar y ver los artefactos de la actividad respectivamente.

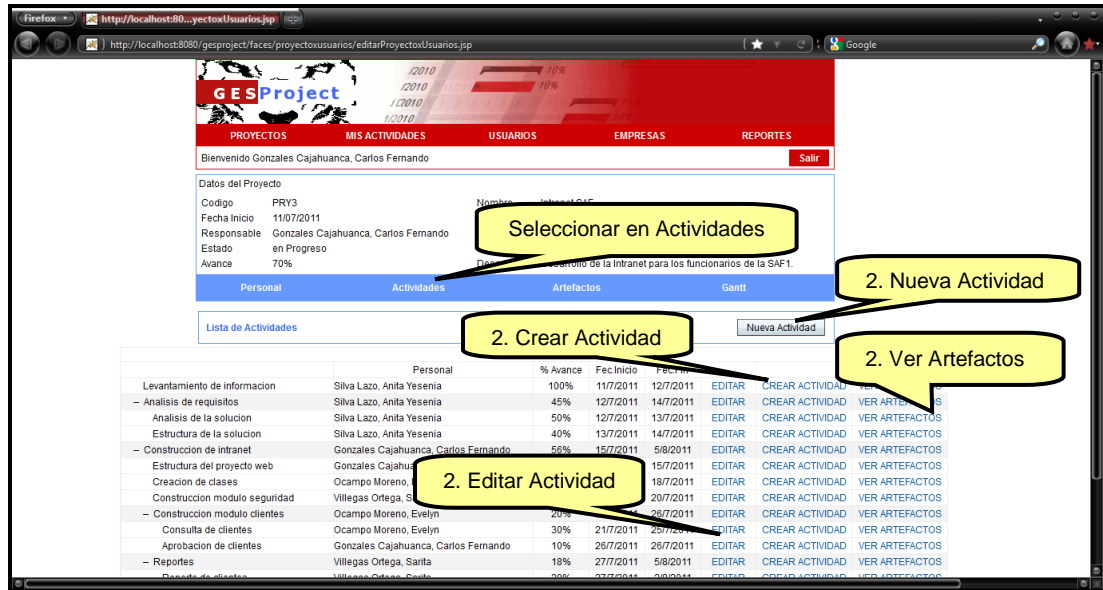


Figura 3.29. Pantalla Mantener Actividades del Proyecto.

2. Si se desea ingresar los datos de una actividad, debe seleccionar el botón “Nueva Actividad” e ingresar los datos solicitados en la pantalla, como se muestra en la Figura 3.30.

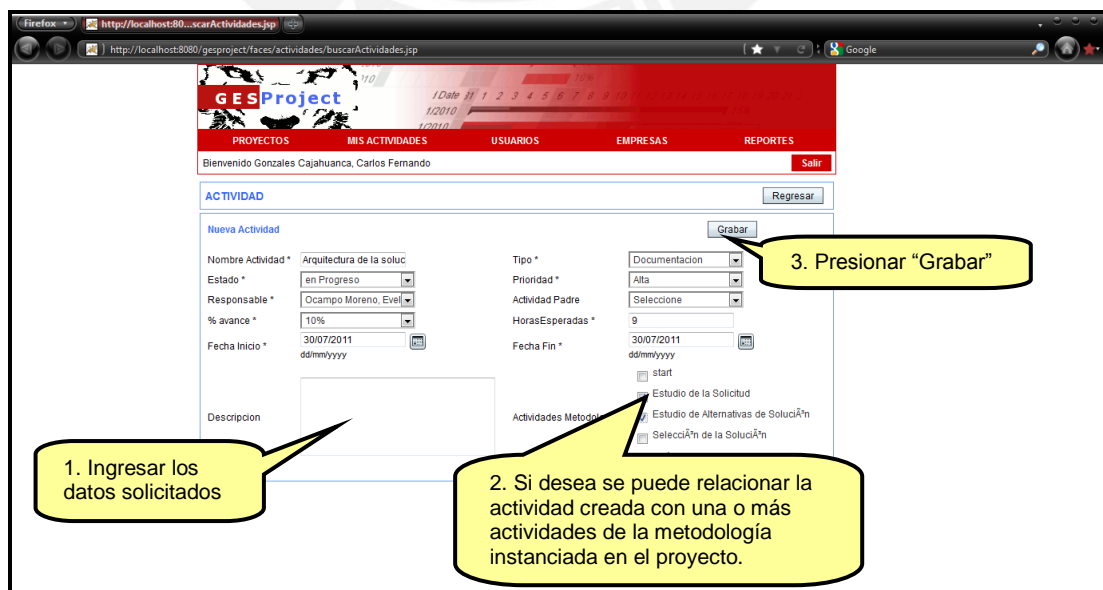


Figura 3.30. Pantalla Nueva Actividad del Proyecto.

- Si se desea modificar los datos de algún registro del listado, se elige la opción “Editar”, se modifican los campos necesarios y luego presionar el botón “Grabar”, como se muestra en la figura 3.31.

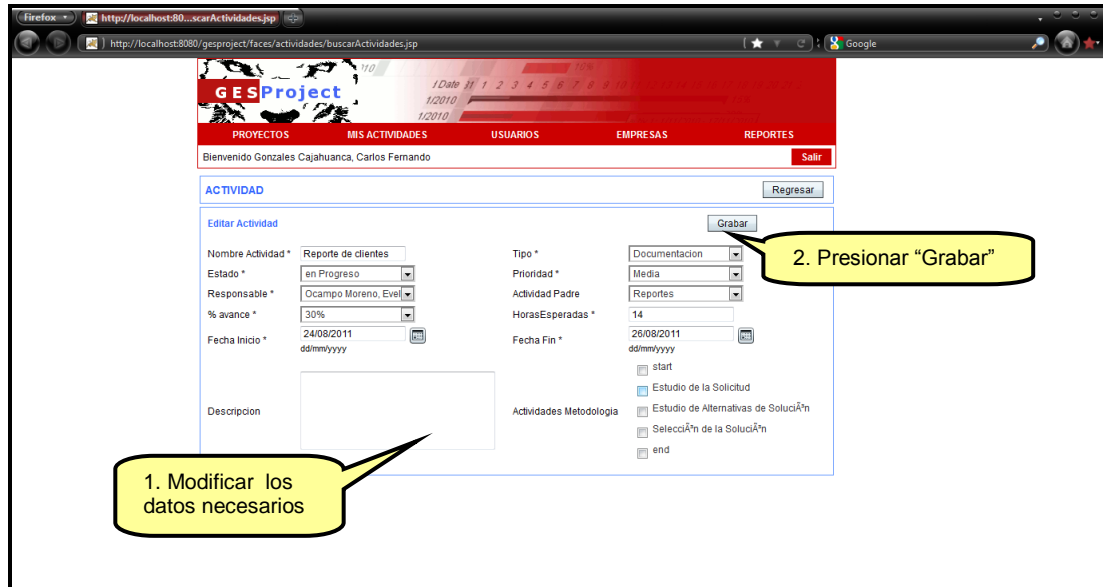


Figura 3.31. Pantalla Editar Actividad del Proyecto.

Flujo de eventos respectivo para la opción “Artefactos”.

- Para realizar el mantenimiento de un artefacto perteneciente al proyecto registrado, el usuario debe seleccionar la opción “Artefactos”, donde se listan los registros existentes en el sistema, como se muestra en la Figura 3.32. además se muestran las opciones “Nuevo Artefacto”, “Versiones”, “Descargar”, “Reservar”, “Liberar” y “Subir”.



Figura 3.32. Pantalla Mantener Artefactos del Proyecto.

- Si se desea ingresar los datos de un artefacto, debe seleccionar el botón “Nuevo Artefacto” e ingresar los datos solicitados en la pantalla, como se muestra en la Figura 3.33.

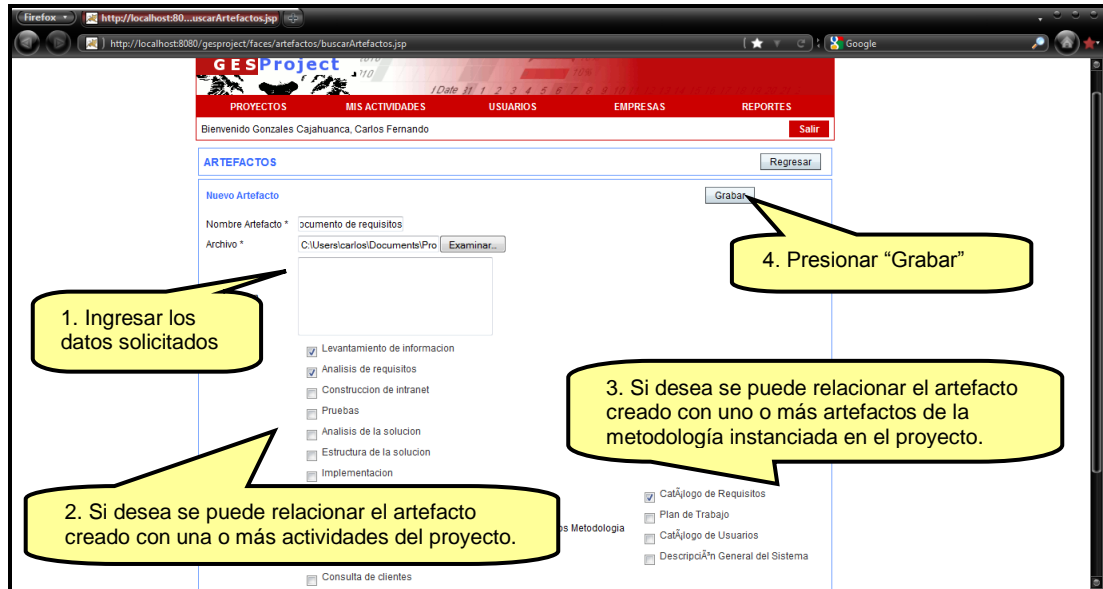


Figura 3.33. Pantalla Nuevo Artefacto del Proyecto.

- Si se desea visualizar las versiones de un artefacto determinado, debe seleccionar en “Versiones”, como se muestra en la figura 3.34.

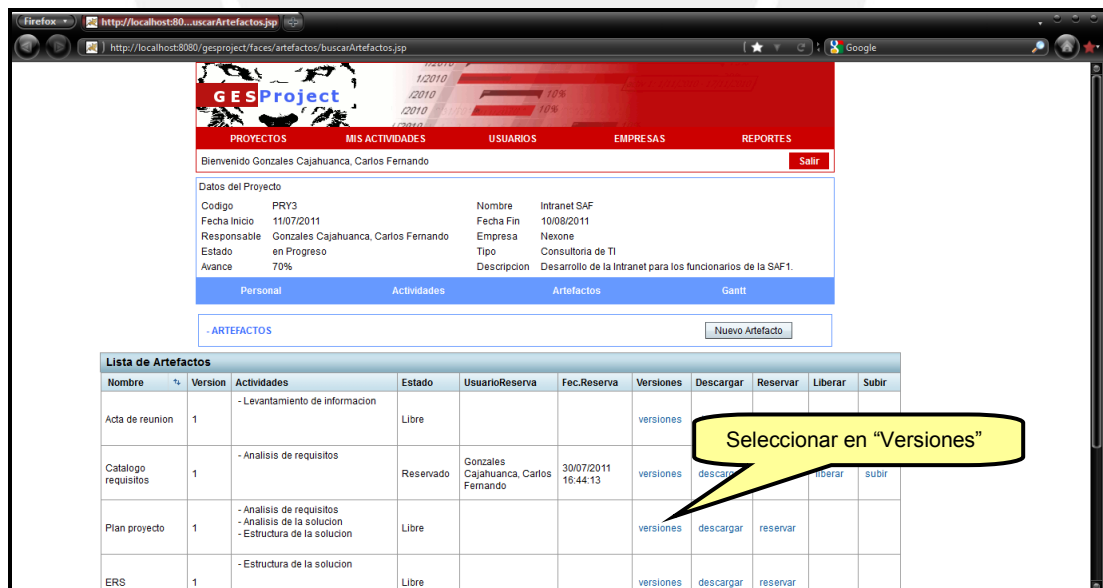


Figura 3.34. Pantalla elegir ver Versiones del Artefacto del Proyecto.

- Se muestra el listado de versiones registradas y si se desea cambiar la versión vigente del artefacto seleccionado, se selecciona el número de versión y luego presionar el botón “Cambiar”, como se muestra en la Figura 3.35.

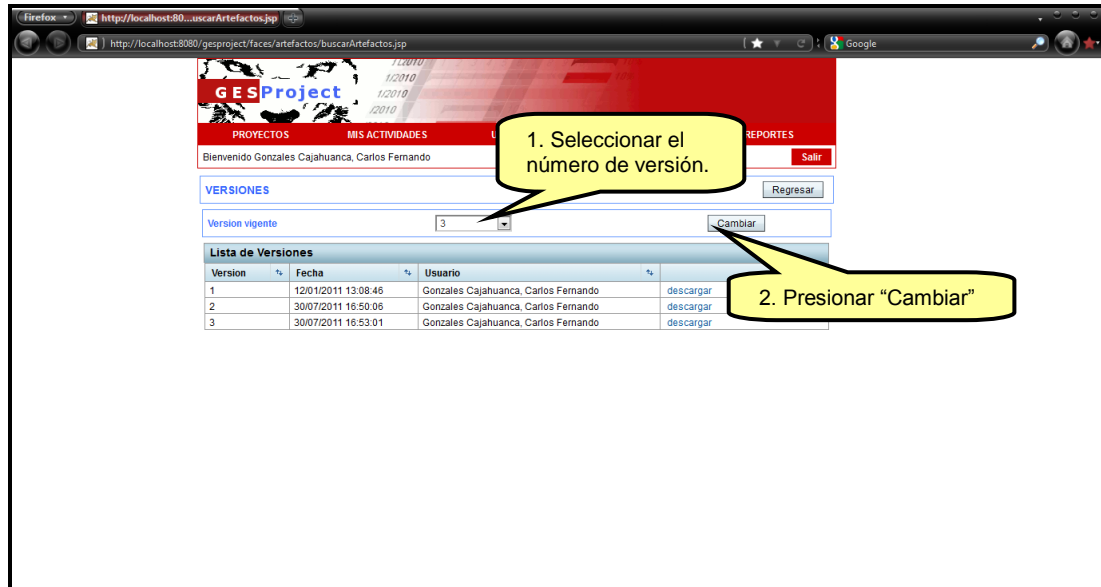


Figura 3.35. Pantalla Listado de Versiones del Artefacto del Proyecto.

- Si se desea descargar algún registro del listado mostrado, se selecciona el registro y luego hace clic al botón “Descargar”, como se muestra en la Figura 3.36.

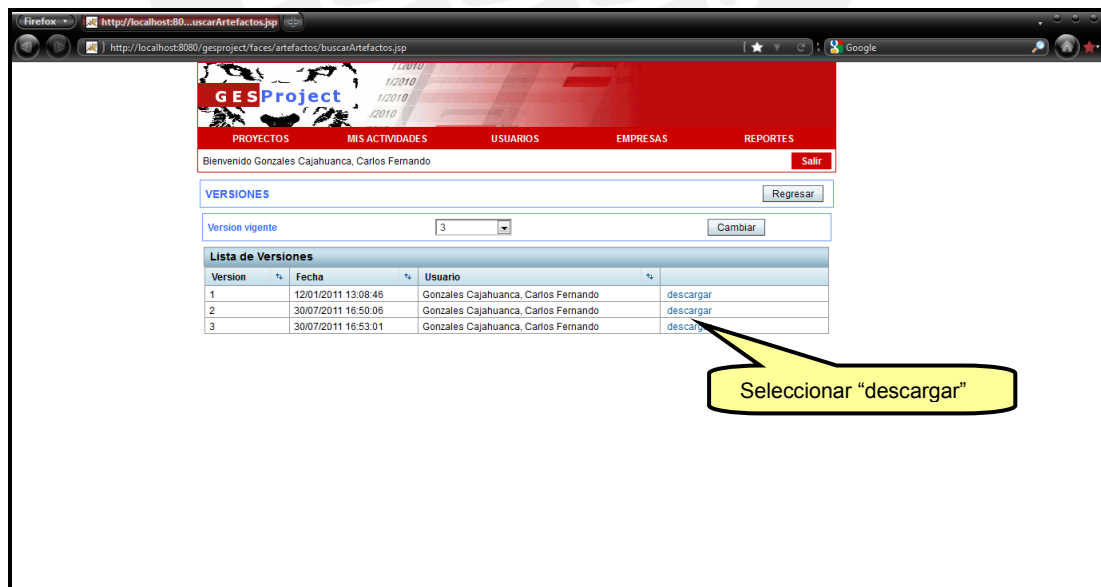


Figura 3.36. Pantalla Descarga de Versiones Artefacto del Proyecto.

Flujo de eventos respectivo para la opción “Gantt”.

1. Para visualizar el diagrama de Gantt del proyecto seleccionado, el usuario debe seleccionar la opción “Gantt”, como se muestra en la Figura 3.37.

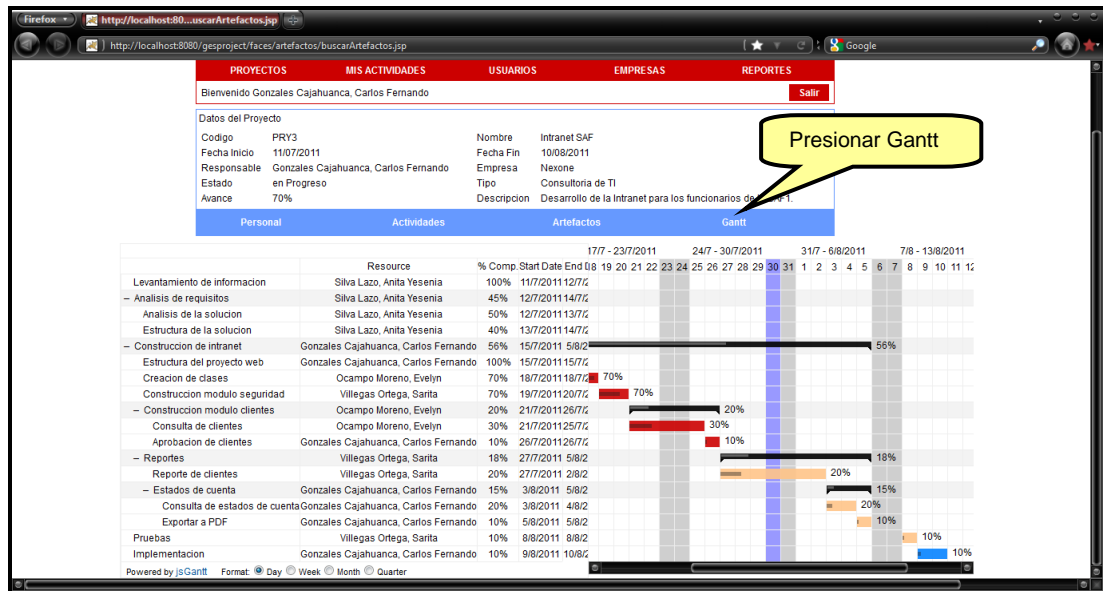


Figura 3.37. Pantalla seleccionar ver Diagrama de Gantt del Proyecto.

2. Al seleccionar la opción se muestra el “Diagrama de Gantt” se muestran las actividades según su distribución de acuerdo al calendario, además se muestra la jerarquía entre actividades contenedoras y sus actividades internas esto se muestra en forma de árbol, como se muestra en la Figura 3.38.

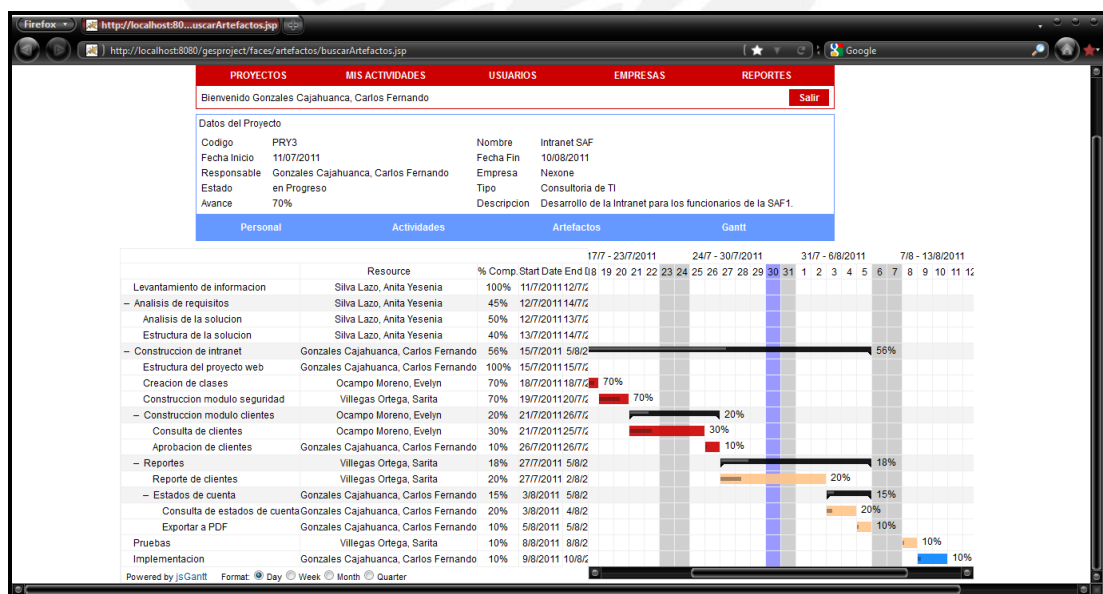


Figura 3.38. Pantalla Diagrama de Gantt del Proyecto.

También se puede visualizar el período de duración de cada actividad, las fechas de inicio y fin, el tiempo total requerido para la ejecución de una actividad determinada y el personal asignado como responsable de la actividad. En la sección derecha se muestra el gráfico del diagrama de Gantt relacionado a las actividades del proyecto, cada actividad se muestra como una barra horizontal con el porcentaje de avance del trabajo a realizar en la actividad, además de mostrarse con un determinado color indicando el retraso de dicho trabajo.

3.10.3. Gestión de la Configuración

La herramienta EACS Project Manager consta de un módulo de gestión de la configuración, el cual tiene como propósito asegurar la validez y disponibilidad de las versiones de los artefactos en todas las etapas de vida del software, manteniendo la integridad de los elementos de trabajo identificando, controlando y auditando dichos elementos.

A continuación se describe el flujo de eventos respectivo para la gestión de la configuración.

1. Al seleccionar la opción “Artefactos” se muestra el listado de los artefactos que tiene un determinado proyecto dentro del sistema, como se muestra en la Figura 3.39.

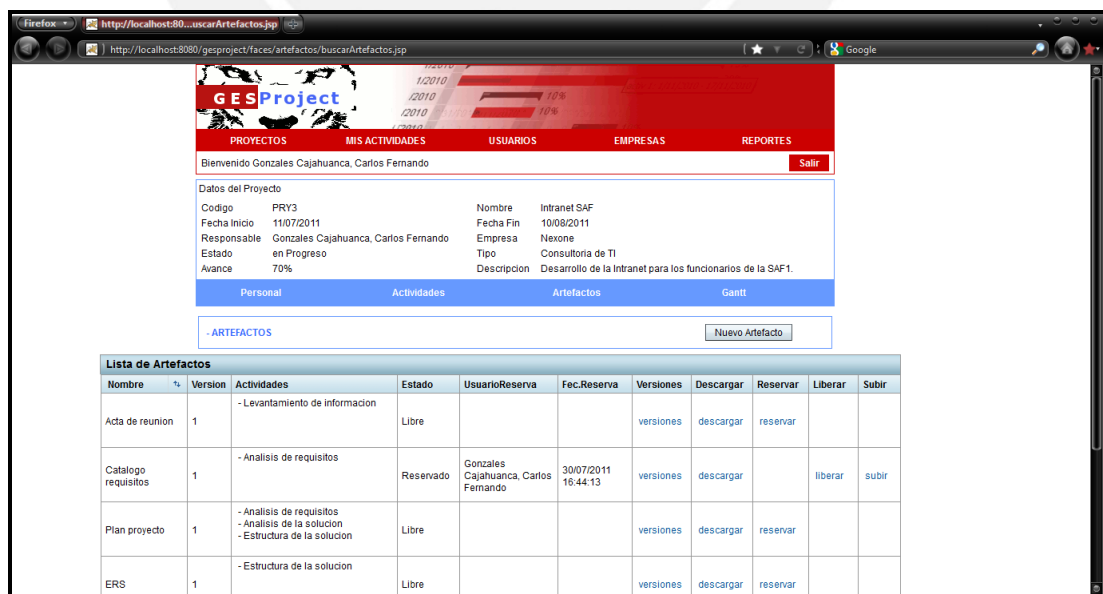


Figura 3.39. Pantalla Listado Artefactos del Proyecto.

- Para reservar el artefacto, se selecciona el registro del listado mostrado y luego se selecciona “Reservar”, como se muestra en la Figura 3.40.

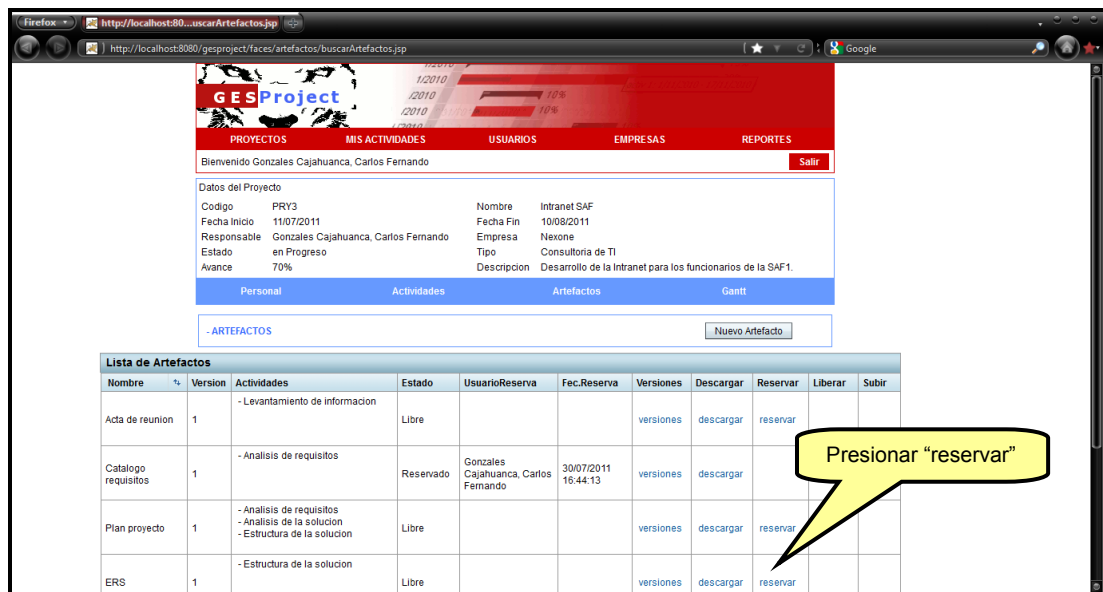


Figura 3.40. Pantalla Reserva Artefactos del Proyecto.

- Se regresa a la pantalla “Mantener Artefactos del Proyecto”, donde se listan los registros existentes en el sistema y el registro anteriormente reservado es “bloqueado”, así ningún otro usuario podrá modificarlo solo consultarlo, habilitándose la opción de “Liberar” y “Subir”, como se muestra en la Figura 3.41.

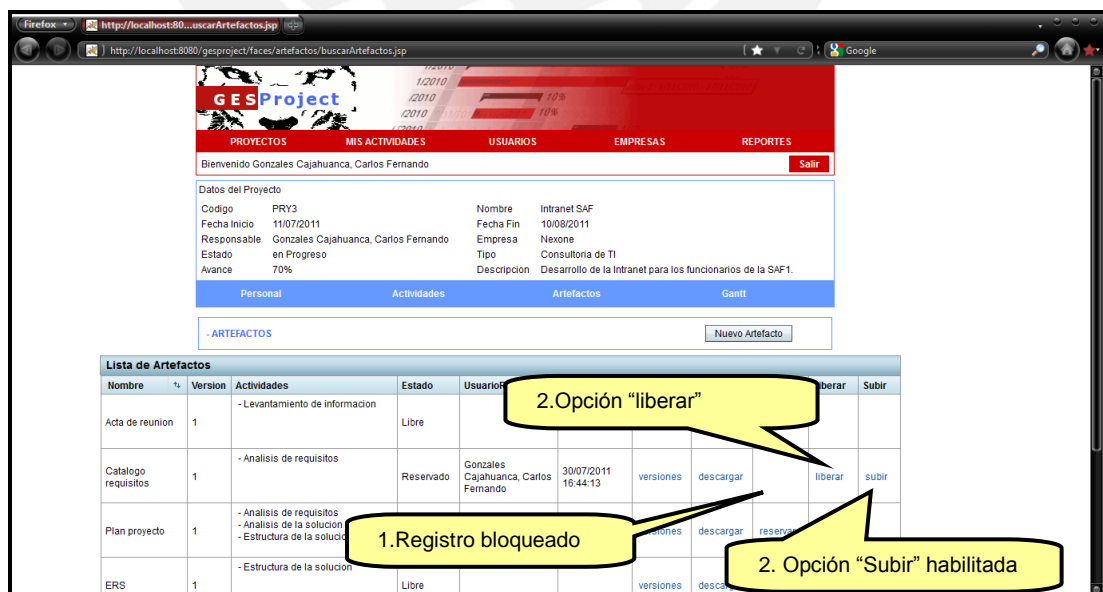


Figura 3.41. Pantalla Artefacto reservado del Proyecto.

- Para agregar artefactos y sus nuevas versiones se selecciona algún registro del listado mostrado que tenga habilitado la opción de “Subir”, como se muestra en la Figura 3.42.



Figura 3.42. Pantalla elegir Subir Artefacto del Proyecto.

- Luego de seleccionar la opción “Subir”, se ingresa los datos solicitados en la pantalla, en la cual se genera una nueva versión del artefacto seleccionado. Se debe seleccionar el artefacto a subir de una ruta determinada, así como ingresar alguna observación. Al presionar el botón “Grabar” el sistema cambia de estado el artefacto así puede ser modificado por otros usuarios y se registran los datos del usuario que generó la versión y fecha, como se muestra en la figura 3.43.

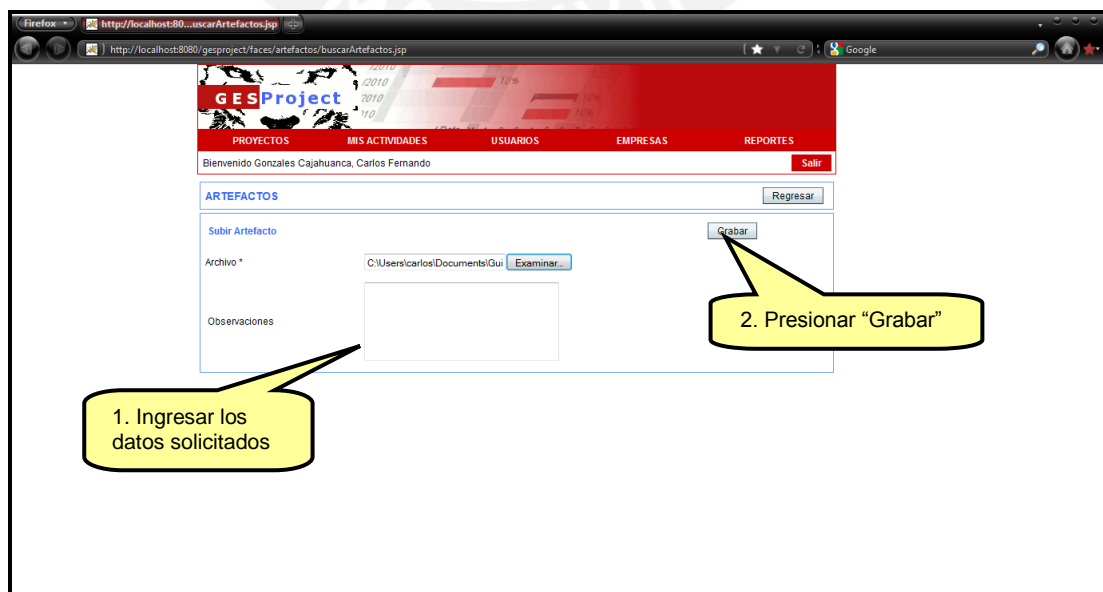
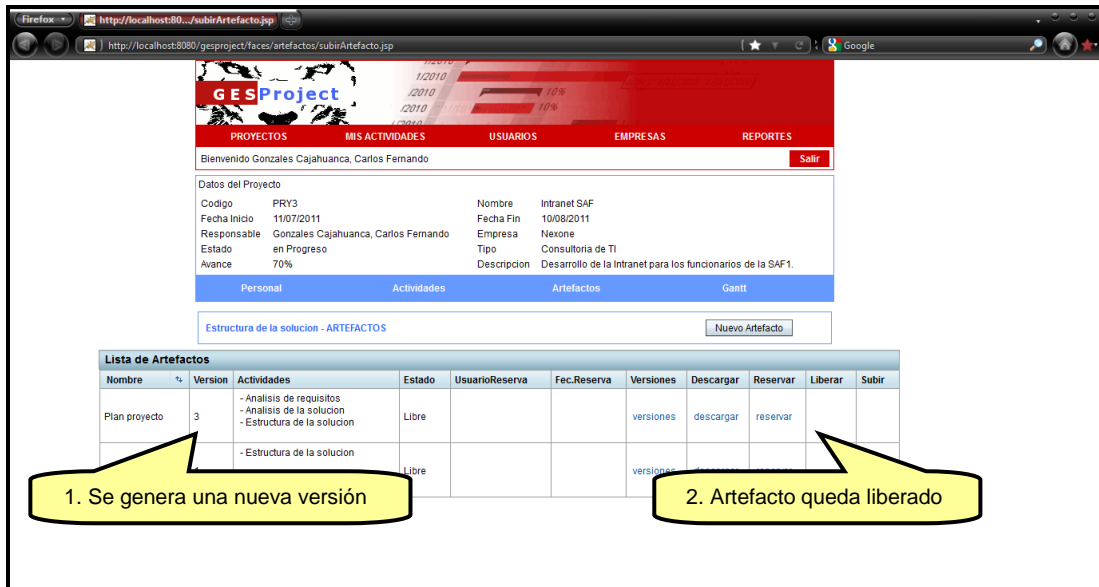


Figura 3.43. Pantalla Subir Artefacto del Proyecto.

6. Se muestra la pantalla del listado de artefactos donde se observa que se genera una nueva versión del artefacto seleccionado y queda liberado para que otro usuario pueda hacer uso de éste, como se muestra en la figura 3.44.



GESProject

PROYECTOS MIS ACTIVIDADES USUARIOS EMPRESAS REPORTE

Bienvenido Gonzales Cajahuanca, Carlos Fernando [Salir](#)

Datos del Proyecto

Codigo	PRY3	Nombre	Intranet SAF
Fecha Inicio	11/07/2011	Fecha Fin	10/08/2011
Responsable	Gonzales Cajahuanca, Carlos Fernando	Empresa	Nexone
Estado	en Progreso	Tipo	Consultoria de TI
Avance	70%	Descripcion	Desarrollo de la Intranet para los funcionarios de la SAF1.

Personal Actividades Artefactos Gantt

Estructura de la solucion - ARTEFACTOS [Nuevo Artefacto](#)

Nombre	%	Version	Actividades	Estado	UsuarioReserva	Fec.Reserva	Versiones	Descargar	Reservar	Liberar	Subir
Plan proyecto		3	- Analisis de requisitos - Analisis de la solucion - Estructura de la solucion	Libre			versiones	descargar	reservar		
			- Estructura de la solucion	Libre			versiones				

1. Se genera una nueva versión

2. Artefacto queda liberado

Figura 3.44. Pantalla Artefacto liberado del Proyecto.

Capítulo 4: Observaciones, conclusiones y recomendaciones

En este capítulo se resumirá las observaciones y conclusiones que se han podido obtener durante la ejecución del presente trabajo. También se brindará algunas recomendaciones para trabajos relacionados al mismo tema.

4.1. Observaciones

- La utilización de archivos XPDL para la definición de metodologías permitió simplificar la integración entre las herramientas del Grupo de Investigación y Desarrollo en Ingeniería de Software y del proyecto COMPETISOFT.
- La utilización del marco de trabajo Spring ha ayudado a la administración de los objetos, facilitando la concurrencia de los datos de los objetos para toda la aplicación.
- Mediante la utilización de metodologías previamente definidas, aseguramos que el proyecto se ejecute de una manera estructurada y ordenada.
- El uso de Software Libre redujo significativamente los costos de desarrollo.
- El sistema tiene una interfaz de usuario amigable que permite a los equipos de proyecto concentrarse en la ejecución del proyecto y no en cuestiones técnicas que tienen que ver con el uso del sistema.

- La utilización del marco de trabajo JSF (Java Server Faces) permite enlazar los controles de un formulario web con los objetos creados, lo que permitió mostrar la información rápidamente y escribiendo menos código en las páginas.
- Para el diagrama de Gantt, la utilización de una librería independiente de la tecnología y lenguaje de programación, facilitó su integración.
- La utilización de un archivo XML para la carga de datos del diagrama de Gantt permite su fácil integración con cualquier tecnología web utilizada en el desarrollo.
- La cantidad de código escrito en Javascript para realizar las tareas del proyecto es dramáticamente reducida gracias a la inclusión de librerías como JQuery, además de ofrecer la estandarización y correcta presentación en diferentes navegadores Web.
- La herramienta permitirá un mecanismo de gestión del repositorio de los artefactos (gestión de la configuración) para cada proyecto, permitiendo a los usuarios administrar y modificar con seguridad los artefactos que se definirán por cada actividad, esta información ayudará a los administradores a controlar el avance de los proyectos siguiendo la metodología elegida.

3.2. Conclusiones

Como resultado del trabajo desarrollado a lo largo de todos los capítulos que componen esta tesis, se detallan a continuación las conclusiones obtenidas según el alcance establecido al inicio de este documento y cuyos objetivos fueron cumplidos:

- Se construyó una herramienta bajo un entorno web, que permite la administración de proyectos de software.
- La herramienta desarrollada permite crear la instancia de una metodología y asociarla a un proyecto específico.
- Mediante la herramienta se administran proyectos específicos, además se utilizaron archivos XML como repositorio de datos.

3.3. Recomendaciones

Como resultado del trabajo realizado, se ha podido identificar algunas recomendaciones que contribuirán a la mejora del proceso para el desarrollo de la plataforma:

- El sistema se encuentra basado en XPDL y XML como archivos de datos, es entera responsabilidad del administrador de cuidar los permisos sobre los archivos para evitar daños y pérdida de datos.
- El hardware del servidor web debe ser el adecuado para la ejecución de aplicaciones Java web, esto para brindar tiempos de respuesta mínimos al usuario.
- Para el presente trabajo se utilizó el contenedor web Tomcat ya que ofrece el rendimiento mínimo para el desarrollo de la herramienta, sin embargo si se desea utilizar en un ambiente de producción sugiere utilizar contenedores Web de mejores prestaciones que Tomcat tales como Glassfish o JBoss que también es son contenedores gratuitos.
- Previamente a la ejecución de la aplicación debe realizarse la configuración de rutas desde donde la aplicación buscará los archivos xml, xpdl y el repositorio. Para esta tarea se debe modificar el archivo de configuración bundle.properties ubicado en el directorio raíz de la aplicación /WEB-INF/classes/gesproject/Bundle.properties.
- Se recomienda desarrollar módulos de costos, esfuerzo, ruta crítica, manejo del valor devengado y manejo de desviaciones.

Bibliografía

- [BAB, 1986] W. Babich, Software ConFIGuration Management, Addison-Wesley, 1986.
- [BRYAN-SIEGEL, 1984] W. L. Bryan y S. G. Siegel, Software product assurance. Techniques for reducing software risks, Elsevier, 1984.
- [BOVERI, 1990] Boveri Brown . "Manual de Gestión de Proyectos";
- [PEREÑA, 1996] Pereña Brand, Jaime. "Dirección y Gestión de Proyectos".
- [PRESSMAN, 1998] Pressman, Roger S. "Ingeniería del software : un enfoque práctico".
- [CMII] Model for Configuration Management. Institute of Configuration Management.
- [SWEBOK, 2004] Guide to the Software Engineering Body of Knowledge.
- [JACOBSON, 2000] Ivar Jacobson, Grady Booch, James Rumbaugh. "El proceso unificado de desarrollo de software".
- [KRUCHTEN] Philippe Kruchten. "The rational unified process made easy: a practitioner's guide to the RUP".
- [MELOCHE, 2002] Thomas Meloche. "The Rational Unified Process (RUP)"
<http://www.menloinnovations.com/freestuff/whitepapers/Rational%20Unified%20Process.pdf>
- [NTP12207, 2007] NTP-ISO/IEC 12207 Norma Técnica Peruana NTP-ISO/IEC 12207.
- [IEEE Std.729-1983] IEEE Standard Glossary of software Engineering Terminology. The Institute of Electrical and Electronics Engineers, Inc. New York, EEUU.
- [IEEE Std.828-1998] IEEE Standard for Software Configuration Management Plans. The Institute of Electrical and Electronics Engineers, Inc. New York, EEUU.
- [PMBOK, 2008] Guía de los Fundamentos para la Gestión de Proyectos (Guía del PMBOK) Cuarta edición.
- PMI] Project Management Institute (PMI) www.pmi.org
- [MCMANUS, 2003] McManus, John y Wood-Harper, Trevor (2003) "Information Systems project management: The price of failure", Management Services; May; 47, 5; ABI/INFORM Global, pp. 16-19
- [KRUCHTEN, 1999] P.B. Kruchten: The Rational Unified Process (An Introduction).
- [NTP-ISO/IEC 12207:2006] Norma Técnica Peruana NTP-ISO/IEC 12207.
- [COMPETISOFT, 2006] COMPETISOFT: Proyecto de mejora de procesos para fomentar la competitividad de la pequeña y mediana industria del software de Iberoamérica, Versión 0.2. Diciembre 2006.
- [BPMI] Business Process Management Initiative. Business Process Modeling Language. BPMI.org. <http://www.bpmi.org/bpmn-spec.htm>
- [MOPROSOFT, 2003] Modelo de Procesos para la Industria de Software, MoProSoft V 1.1, mayo 2003, pp 121
- [MOPROSOFT] Oktaba, H., Alquicira, C., Su, A., Martinez, A., Pérez, C., López, F.: Modelo de procesos para la industria del software.

<http://www.e-computacion.net/MoProSoft%20HannaOktaba.ppt#259,2,Contenido> [visitado en Marzo del 2008]

[B-KIN] En <http://www.b-kin.com>, [visitado en Marzo del 2008]

[BUGTRACK] En <http://www.bug-track.com>, [visitado en Marzo del 2008]

[KPLATO] En <http://www.koffice.org/kplato>, [visitado en Marzo del 2008]

[DOTPROJECT] En <http://www.dotproject.net>, [visitado en Marzo del 2008]

[MSPROJECT] <http://www.microsoft.com/project>, [visitado en Noviembre del 2010]

[PRIMAVERA] http://www.oracle.com/us/corporate/Acquisitions/Primavera_Systems [visitado en Marzo del 2011]

[XPDL] En <http://www.xpdl.org>, [visitado en Julio del 2008]

[STANDISH GROUP, 2009]. En www.standishgroup.com, [visitado en Agosto del 2009]

[MVC] Modelo, Vista y Controlador
En http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, [visitado en Marzo del 2009]

[MJS] Herramienta para Modelado de Procesos: MJS Process Designer. Tesis para optar el Título de Ingeniero Informático presentada por: Camarena M., Pedreschi J. y Rondón S.

[JSR 127] Original Java Specification Request. En <http://www.jcp.org/en/jsr>

[JSF] En www.java.sun.com/javaee/javaserverfaces, [visitado en Agosto del 2008]

[SPRING MVC] En www.spring.org [visitado en Agosto del 2008]

[STRUTS] En www.struts.apache.org [visitado en Agosto del 2008]

[TAPESTRY] En www.tapestry.apache.org [visitado en Agosto del 2008]

[NETBEANS] En www.netbeans.org [visitado en Agosto del 2008]

[WEBSERVICES] En www.w3.org/2002/ws [visitado en Agosto del 2008]

[APPLETS] En www.java.sun.com/applets [visitado en Agosto del 2008]

[J2SDK] En www.java.sun.com/j2se [visitado en Agosto del 2008]

[JQUERY] En www.jquery.com [visitado en Septiembre del 2009]

[JFREECHART] En www.jfreechart.com [visitado en Julio del 2010]

[JSGANTT] Shlomy Gantz and Brian Twidt En code.google.com/p/jsgantt/ [visitado en Septiembre del 2010]

[JBOSSE] JBoss Application Server. En <http://labs.jboss.com> [visitado en Agosto del 2008]

[TOMCAT] The Apache Software Foundation. En <http://www.apache.org> [visitado en Agosto del 2008]

[SERV_APLIC] Servidor de Aplicaciones.
En http://es.wikipedia.org/wiki/Servidor_de_aplicaciones [visitado en Agosto del 2008]

[SERV_WEB] Web Server. En http://en.wikipedia.org/wiki/Web_server [visitado en Agosto del 2008]

[XML] En <http://www.w3.org/standards/xml> [visitado en Julio del 2008]

[JAVA] James Gosling, Henry McGilton. White Paper: The Java Language Environment. 1996. Web. En: <http://java.sun.com/docs/white> [visitado en Julio del 2008]

[SPRING] En http://es.wikipedia.org/wiki/Spring_Framework [visitado en Agosto del 2008]

[SAX] SAX - The Simple API for XML. En <http://www.saxproject.org> [visitado en Agosto del 2008]

[DOM] A. Le Hors, P. Le Hégarret, L. Wood, et al. Document Object Model (DOM) Level 2 Core Specification, Version 1.0, W3C Recommendation. En <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113> [visitado en Agosto del 2008]

[BINDING_XML] J. Pinnock, S. Mohr, J. Duckett, et al. Professional XML 2nd Edition, Wrox Press, May 2001.

[DBINDING] E. Armstrong. Code fast, run fast with XML data binding. En <http://java.sun.com/xml/jaxp/dist/1.0.1/docs/binding/DataBinding.html> [visitado en Septiembre del 2008]

[CASTOR] En <http://www.castor.org> [visitado en Marzo del 2008]

[JAVABEANS] En <http://java.sun.com/products/javabeans> [visitado en Marzo del 2008]

[JAXB] En <http://java.sun.com/xml/jaxb> [visitado en Agosto del 2008]

[POJO] Richardson, Chris. "POJOs in Action", 1ra edición. Manning Publications. 2006.

[EJB] Monson-Haefel, Richard; Burke, Bill; Labourey, Sacha. "Enterprise JavaBeans", 4ta edición. O'Reilly Media, Inc. 2004.

[OTI] Object Technology International. Eclipse Platform Technical Overview. 2003. En: <http://www.eclipse.org/whitepapers/eclipse-overview.pdf> [visitado en Agosto del 2008]

[ECLIPSE] The Eclipse Foundation. Eclipse IDE. 2008. Web. En: <http://www.eclipse.org> [visitado en Agosto del 2008]

[JSP] Sun Developer Network. JavaServer Pages Overview. 2008. Web. En: <http://java.sun.com/products/jsp/overview.html> [visitado en Agosto del 2008]

[EACS] Herramienta para Gestión de Proyectos basada en XPDL para el proyecto COMPETISOFT – Construcción, Pruebas e Integración.