

**ENTORNO DE SOPORTE PARA EL PROCESO DE DESARROLLO DE
SOFTWARE A TRAVÉS DE BPM**

Jorge Enrique Otálora Luna

**Trabajo para optar por el título de
Magister en Ingeniería**

**Director:
Raquel Anaya De Páez
Ph.D. Ingeniería de la programación e inteligencia artificial**

**Medellín
Universidad EAFIT
Escuela de Ingenierías
Maestría en Ingeniería
2010**

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Medellín, Julio del 2010

DEDICATORIA

A Dios.

A la memoria de mi papá Jorge Enrique, que siempre me alentó a superarme.

A mi mamá Gloria Inés, ejemplo de tenacidad y de cómo sacar adelante proyectos.

A mis hermanos María Teresa, Rosa Inés y Guillermo Arturo, que siempre me han acompañado en los buenos y malos momentos.

AGRADECIMIENTOS

En mi corazón y en mi mente quedará eternamente el alegre recuerdo de todas las personas que intervinieron de una u otra forma para que ahora pueda alcanzar ésta meta, he podido comprobar que la mano de Dios apareció siempre y aunque en ocasiones no pude entender, se que todos los eventos sucedidos durante éste tiempo tienen un propósito.

Quiero agradecer en primera instancia a **Dios**, en quien me apoyé y que a pesar de estar tan ocupado con el universo siempre tuvo tiempo para mi, también por colocar en mi camino algunos obstáculos que una vez superados se convirtieron en fuertes bases para apoyarme y seguir adelante.

A **Raquel**, mi asesora, quien siempre creyó en mí y pudo entenderme y guiarme magistralmente, ayudándome a alcanzar las metas con sus sabios consejos, con su ejemplo de vida, con su corazón y el de su familia abiertos para que yo pudiera entrar y poder entender que la vida puede ser hermosa, que los buenos valores siempre le guiarán y que el ayudar a los demás siempre le beneficia. Alguna vez me dijo “Si una puerta se cierra, otra se abre”, y he tratado de tener esto presente y he visto como es cierto, pretendo en adelante abrir muchas puertas al éxito de la manera como Ella me enseñó, con perseverancia, con entusiasmo, con disciplina, con sencillez pero sobre todo con **Dios**.

A **mí mamita, a mis hermanas Tere y Rosita y a mi hermano Memo** que siempre me han apoyado, y en quienes he podido encontrar refugio y consejo incondicionales. A **mi papito** que asistirá desde el cielo a mi grado y de seguro no se aguantará las ganas de llorar.

A **Omar y Cristian** de Ingenius-Group que siempre estuvieron dispuestos a prestarme su colaboración, aunque no entendían muy bien que era lo que yo estaba haciendo.

A **Gloría, Paulita, César y Luis Miguel**, mi familia de Medellín por su acogida, por haberme hecho sentir en casa, por aguantarme y dejarme ganar en todos los juegos.

A **Dianita** por sus oraciones, por siempre estar preguntando y regañándome para que terminara rápido y por haber sido firme conmigo a pesar de la incomprensión de muchas personas.

A **mí**, porque quiero aparecer en unos agradecimientos y porque creo fui capaz de superar muchas dificultades.

CONTENIDO

CONTENIDO	5
INTRODUCCION	10
RESUMEN	12
GLOSARIO	12
1. DEFINICION DE LA PROPUESTA DE INVESTIGACION	16
1.1 DESCRIPCION DEL PROBLEMA	16
1.2 JUSTIFICACIÓN.....	17
1.3 OBJETIVOS	22
1.3.1 <i>Objetivo general.</i>	22
1.3.2 <i>Objetivos específicos.</i>	22
1.4 METODOLOGÍA.....	23
1.4.1 <i>Descripción del problema.</i>	23
1.4.2 <i>Recopilación de Información.</i>	23
1.4.3 <i>Desarrollo de Estudio de Herramientas.</i>	23
1.4.4 <i>Propuesta Integrada para el Modelado del Proceso y el Monitoreo de Proyecto Software.</i>	24
2 MARCO TEORICO	26
2.1 PROCESO DE SOFTWARE	26
2.2 MEJORA DE PROCESOS	27
2.2.1 <i>Modelos de procesos de software</i>	28
2.2.2 <i>Modelos de procesos software según los elementos que representan.</i>	29
2.2.3 <i>Modelos de procesos software según su nivel de abstracción.</i>	29
2.3 INGENIERÍA DE PROCESOS.....	30
2.4 METAMODELO SPEM	33
2.5 BUSSINES PROCESS MANAGEMENT (BPM)	33
2.6 BPMN	38
2.7 TECNOLOGIAS IMPLEMENTADAS EN EL DESARROLLO DEL ENTORNO.....	41
2.7.1 <i>Workflow.</i>	41
2.7.2 <i>Monitoreo de procesos.</i>	42
2.7.3 <i>Principios de monitoreo de procesos a través de workflow.</i>	42
2.7.4 <i>Interfaces WfMC.</i>	43
2.7.5 <i>BPEL.</i>	44
2.7.6 <i>XFORMS.</i>	44
2.7.7 <i>WSDL.</i>	45
2.7.8 <i>XML SCHEMA.</i>	46
2.7.9 <i>Gestión y monitoreo de procesos software.</i>	47
2.8 INTEROPERABILIDAD	47
2.8.1 <i>Dimensiones o niveles de interoperabilidad.</i>	48
2.8.2 <i>Interoperabilidad mediante estándares.</i>	50
2.9 ANTECEDENTES DE INGENIERÍA DE PROCESOS EN EMPRESAS DE SOFTWARE	53
3. ESTUDIO COMPARATIVO DE HERRAMIENTAS	55
3.1 ESTUDIO COMPARATIVO HERRAMIENTAS DE GESTIÓN DE PROYECTOS.....	59
3.2 ESTUDIO COMPARATIVO HERRAMIENTAS DE MODELADO BPMN	61
3.3 ESTUDIO COMPARATIVO DE HERRAMIENTAS WORKFLOW	63

4. PROPUESTA INTEGRADA PARA EL MODELADO DEL PROCESO Y EL MONITOREO DE PROYECTO SOFTWARE	66
4.1 ESQUEMA GENERAL DE LA SOLUCION.....	66
4.2 ARQUITECTURA DE LA SOLUCION: VISTA DE DESPLIEGUE.	67
4.3 SOLUCION PROPUESTA EN INTALIO	70
4.4 SOLUCION PROPUESTA EN JBPM.....	71
4.5 ESCENARIO DE USO	74
5 CASO DE ESTUDIO	78
5.1 DESARROLLO DEL ENTORNO EN INTALIO WORKS	78
5.2 DESARROLLO DEL ENTORNO EN JBPM.....	79
5.3 DESARROLLO DE LOS SERVICIOS DE INTEROPERABILIDAD.....	80
5.3.1 <i>Desarrollo de la herramienta de interoperabilidad sobre Intalio Works</i>	81
5.3.2 <i>Obtención de información a través de una API.</i>	81
5.3.3 <i>Obtención de información a través de la base de datos.</i>	81
5.3.4 <i>Obtención de información a través de Apache ODE.</i>	83
5.3.5 <i>Obtención de información a través de modelo del proceso.</i>	84
5.3.6 <i>Obtención de información a través de Intalio BAM.</i>	84
5.3.7 <i>Desarrollo de la herramienta de interoperabilidad sobre JBPM.</i>	86
5.3.8 <i>Obtención de información a través de una API.</i>	86
5.3.9 <i>Desarrollo de la herramienta de interoperabilidad sobre JBPM basada en Aspectos.</i>	88
6 CONCLUSIONES Y TRABAJOS FUTUROS	95
6.1 CONCLUSIONES CON RESPECTO A HERRAMIENTAS.	96
6.1.1 <i>A nivel de herramientas de modelado.</i>	96
6.1.2 <i>A nivel de herramientas workflow.</i>	97
6.1.3 <i>A nivel de herramienta de interoperabilidad.</i>	98
6.1.4 <i>A nivel de herramienta de gestión de proyectos.</i>	99
6.1.5 <i>A nivel del entorno.</i>	99
6.2 TRABAJOS FUTUROS.....	99
6.2.1 <i>Integración de BPM y motores de workflow en una arquitectura MDA.</i>	99
6.2.2 <i>Herramienta de transformación de BPMN a JBPM.</i>	100
6.2.3 <i>Formalizar el modelo de trabajo para diseño y monitoreo del proceso de desarrollo.</i>	100
6.2.4 <i>Configurar el entorno de apoyo bajo la tecnología JBPM en el laboratorio del grupo de ingeniería de software.</i>	100
BIBLIOGRAFIA	102
7. ANEXOS	109
7.1 ANEXO A: ANÁLISIS COMPARATIVO DE HERRAMIENTAS.....	109
7.2 ANEXO B: DESARROLLO DEL CASO DE ESTUDIO SOBRE INTALIO WORKS.	109
7.3 ANEXO C: DESARROLLO DEL CASO DE ESTUDIO SOBRE JBPM.....	109
7.4 ANEXO D: DESARROLLO DE LA HERRAMIENTA DE GENERACIÓN XML PARA HERRAMIENTAS DE GESTIÓN DE PROYECTOS.....	109

LISTA DE FIGURAS

FIGURA 1. MARCO DE TRABAJO PARA LA INGENIERÍA DE PROCESOS SOFTWARE.	31
FIGURA 2 . ESTRUCTURA DE UN PROCESO.	32
FIGURA 3. ESTRUCTURA ESPECÍFICA DEL PROCESO.	32
FIGURA 4. CICLO DE VIDA BPM.	34
FIGURA 5. INTEGRACIÓN DE ESTÁNDARES.	51
FIGURA 6. TIPOS DE INTERCAMBIOS.	52
FIGURA 7. ESQUEMA GENERAL DE LOS ESTUDIOS COMPARATIVOS.	55
FIGURA 8. GRÁFICA DE RESULTADOS CONSOLIDADOS PARA HERRAMIENTAS DE GESTIÓN DE PROYECTOS.	59
FIGURA 9. GRÁFICA DE RESULTADOS HERRAMIENTAS DE GESTIÓN DE PROYECTOS.	60
FIGURA 10. GRÁFICA DE RESULTADOS CONSOLIDADOS.	61
FIGURA 11. GRÁFICA DE RESULTADOS HERRAMIENTAS DE MODELADO BPMN.	63
FIGURA 12. GRÁFICA DE RESULTADOS CONSOLIDADO.	64
FIGURA 13. GRÁFICA DE RESULTADOS DE HERRAMIENTAS WORKFLOW.	65
FIGURA 14. ESQUEMA GENERAL DE LA SOLUCIÓN.	67
FIGURA 15. ARQUITECTURA DE LA SOLUCIÓN.	69
FIGURA 16. VISTA DE DESPLIEGUE SOLUCIÓN EN INTALIO.	71
FIGURA 17. VISTA DE DESPLIEGUE SOLUCIÓN EN JBPM.	72
FIGURA 18. ESCENARIO DE USO ENTORNO DE SOPORTE PARA EL PROCESO DE DESARROLLO SOFTWARE A TRAVÉS DE BPM.	74
FIGURA 19. PRIMERA PARTE FLUJO DEL PROCESO.	75
FIGURA 20. SEGUNDA PARTE FLUJO DEL PROCESO.	76
FIGURA 21. TERCERA PARTE FLUJO DEL PROCESO.	77
FIGURA 22. PROCESO ELABORADO EN INTALIO DESIGNER.	79
FIGURA 23. MODELADO GENERAL DEL PROCESO SOBRE JBPM.	80
FIGURA 24. BASE DE DATOS DERBY DE INTALIO SERVER.	82
FIGURA 25. CONSULTA A LA TABLA APP.TEMPO_TASK DE LA BASE DE DATOS DE INTALIO SERVER.	83
FIGURA 26. INTALIO BAM.	86
FIGURA 27. CLASE GENERADORXMLPROJECT.	90
FIGURA 28. DIAGRAMA DE CLASES PAQUETE COM.JORGE.REVISIONPARES.INTEROPERABILIDAD.	90
FIGURA 29. ASPECTO INFORMACIÓN TAREAS.	91
FIGURA 30. POINTCUT ALFINALIZARTAREAS DEL ASPECTO INFORMACIONTAREAS.	91
FIGURA 31. ASPECTO DESPUÉS DEL PUNTO DE CORTE AL FINALIZARTAREA.	92
FIGURA 32. DIAGRAMA DE CLASES PAQUETE.	92
FIGURA 33. CREAR XML A PARTIR DE LA INFORMACIÓN OBTENIDA EN ASPECTOS EN JBPM.	93
FIGURA 34. ESQUEMA DE ARCHIVO XML.	93
FIGURA 35. CRONOGRAMA GENERADO DESDE JBPM EN OPENPROJ.	94
FIGURA 36. CRONOGRAMA GENERADO DESDE JBPM EN MICROSOFT PROJECT 2003.	94

LISTADO DE TABLAS

TABLA 1. GLOSARIO.....	12
TABLA 2. COMPAÑÍAS COLOMBIANAS QUE HAN LOGRADO ALGÚN NIVEL DENTRO DE LA CLASIFICACIÓN DE CMM.....	21
TABLA 3. PAÍSES QUE IMPLEMENTAN PROYECTOS BPM, FUENTE CLUB DEL BPM.....	35
TABLA 4. OBJETOS DE FLUJO BPMN.	39
TABLA 5. OBJETOS DE CONEXIÓN BPMN.	39
TABLA 6. BPMN SWIMLANES.	40
TABLA 7. ARTEFACTOS NOTACIÓN BPMN.	40
TABLA 8. NIVELES DE INTEROPERABILIDAD SEGÚN AUTOR Y AÑO (MANSO, Y OTROS, 2008).	48
TABLA 9. NIVEL DE IMPORTANCIA DE CARACTERÍSTICAS.	56
TABLA 10. CARACTERÍSTICAS A EVALUAR.	56
TABLA 11. PONDERACIÓN DE CARACTERÍSTICAS.	57
TABLA 12. ESCALA GENERAL DE EVALUACIÓN.	58
TABLA 13. CRITERIOS ESPECÍFICOS DE EVALUACIÓN.....	58
TABLA 14. RESULTADOS CONSOLIDADOS.	58

LISTADO DE ANEXOS

ANEXO A: ANÁLISIS COMPARATIVO DE HERRAMIENTAS.....	109
ANEXO B: DESARROLLO DEL CASO DE ESTUDIO SOBRE INTALIO WORKS.....	109
ANEXO C: DESARROLLO DEL CASO DE ESTUDIO SOBRE JBPM.....	109
ANEXO D: DESARROLLO DE LA HERRAMIENTA DE GENERACIÓN XML PARA HERRAMIENTAS DE GESTIÓN DE PROYECTOS.	109

INTRODUCCION

La realización de una tesis de maestría en el campo de la ingeniería de software es un reto complejo que requiere paciencia y dedicación. La investigación que se presenta en este documento que lleva por nombre “Entorno de soporte para gestión del proceso de desarrollo de software, a través de BPM”, se realizó siguiendo dos aproximaciones:

Desde el punto de vista conceptual, se hizo una revisión de enfoques y tendencias referentes a la mejora de procesos dentro de las organizaciones de software y desde el punto de vista práctico se hizo revisión de soluciones existentes que podrían integrarse para lograr que una organización cuente con recursos que apalanquen los esfuerzos de mejora.

En la actualidad, existen múltiples soluciones informáticas que permiten llevar a cabo tareas de vital importancia para el desarrollo de software tales como la planeación, el diseño y el monitoreo de los procesos; sin embargo, estas herramientas no se encuentran integradas, lo que hace dispendiosa la realización de las actividades propias de esta industria.

Para solventar esta situación, es necesario lograr la interoperabilidad entre las herramientas, la cual se puede conseguir mediante el uso de estándares que permiten crear un mismo lenguaje de comunicación entre programas.

La presente investigación, tiene por objetivo diseñar un entorno que permita automatizar las actividades de planeación, seguimiento y control del proceso de desarrollo de software a través de la integración de herramientas. Con el propósito de que dicha investigación beneficie a empresas de software medianas y pequeñas que no tiene muchos recursos para invertir en infraestructura de soporte, se realizó el análisis de herramientas acotado a soluciones de libre distribución.

El documento está estructurado de la siguiente manera: En el capítulo 1 se hace un resumen del trabajo y una síntesis de los principales términos utilizados a lo largo del trabajo. En el capítulo 2, se podrá encontrar la descripción del tema a tratar durante la investigación, la justificación del proyecto así como los objetivos que se pretenden alcanzar. En el capítulo 3 se presenta la fundamentación teórica que soporta el modelo a realizar. En el capítulo 4 se presenta en análisis comparativo de las herramientas estudiadas, bajo unos criterios establecidos por el autor. El capítulo 5 se presenta la propuesta del entorno integrado para la gestión y monitoreo del proceso software dentro de un proyecto. El capítulo 6 ilustra la manera como se utilizó dicho entorno en un caso de estudio. En el capítulo 7 se presentan los resultados, se realizan las conclusiones respectivas, y

se sugieren algunos trabajos futuros. El capítulo 8 presenta las referencias bibliográficas.

Se incluyen además los siguientes anexos para aquellos lectores que deseen entrar a mayor detalle:

Anexo A: Análisis comparativo de herramientas.

Anexo B: Desarrollo del caso de estudio sobre Intalio|Works.

Anexo C. Desarrollo del caso de estudio sobre JBPM.

Anexo D: Desarrollo de la herramienta de generación xml para herramientas de gestión de proyectos.

RESUMEN

La presente investigación, pretende diseñar un entorno que permita automatizar las actividades de planeación, seguimiento y control del proceso de desarrollo de software a través de la integración de herramientas.

La investigación pretende integrar de manera coherente actividades como el modelado del proceso de desarrollo, la instanciación de dicho proceso en un proyecto específico, así como el monitoreo del proceso a lo largo del desarrollo del proyecto. Dicha integración puede lograrse por medio de la interoperabilidad de una herramienta de modelado BPM, un workflow y una herramienta de gestión de proyectos.

Se realiza un análisis para la selección de herramientas open source de tipo BPM, de workflow y de gestión de proyectos disponibles y se utiliza un caso de estudio que modela en notación BPMN el proceso de revisión de pares, el cual servirá de base para entender las características funcionales de cada una de las herramientas y los mecanismos como estas pueden integrarse.

La interoperabilidad se logra empleando estándares basados en tecnologías XML, y la interfaz cuatro definida por la WfMC, (Workflow Management Coalition) [63] que especifica el formato ideal para intercambiar información entre el motor de workflow y las herramientas deseadas, lo que permite realizar el monitoreo de las tareas que se llevan a cabo [9].

GLOSARIO

Tabla 1. Glosario.

SIGLA	SIGNIFICADO	DEFINICION
BPMN	Business Process Management Notation	Notación gráfica estandarizada, que permite el modelado de procesos de negocio.
BPM	Business Process Management	Disciplina empresarial enfocada a la administración de procesos, por medio de la automatización, modelado, monitorización y optimización de los procesos del negocio.

SIGLA	SIGNIFICADO	DEFINICION
SOA	Service Oriented Architecture	Es una arquitectura que se caracteriza por utilizar servicios para soportar a los requisitos del negocio.
	XSD XML Schemas	Lenguaje utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML.
XFORM	XML Forms	Lenguaje para definir formularios basados en XML.
BPEL	Business Process Execution Language	Lenguaje estandarizado utilizado para la determinación de servicios web, basado en XML y diseñado para el control centralizado de la invocación de diferentes servicios web.
XMI		Estándar para el intercambio de metamodelos usando XML.
BPEL	Business Process Execution Language	Lenguaje estandarizado utilizado para la determinación de servicios web, basado en XML y diseñado para el control centralizado de la invocación de diferentes servicios web.
	WORKFLOW	Software encargado de suministrar las herramientas suficientes para reflejar y automatizar métodos y procesos para organizar un sistema de información, por medio de establecer mecanismos de control.

SIGLA	SIGNIFICADO	DEFINICION
WSDL	WSDL WEB SERVICES DESCRIPTION LANGUAGE	Formato XML que se usa para describir servicios web, describiendo una interfaz pública a los servicios.
BPD	BUSSINES PROCESS DIAGRAM	Interconexión de los artefactos y elementos provistos por la notación BPMN.
UML	UNIFIED MODELING LANGUAGE	Lenguaje de modelado unificado, concebido para modelar sistemas software y uno de los más utilizados en la actualidad.
MOF	META OBJECT FACILITY	Grupo de estándares administrativos de objetos
XML	EXTENSIBLE MARKUP, LANGUAGE	Metalinguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos.
XML	EXTENSIBLE MARKUP, LANGUAGE	Metalinguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos.
DTD	DOCUMENT TYPE DEFINITION	Descripción de una estructura y sintaxis de un documento.
XSL	EXTENSIBLE STYLESHEET LANGUAGE FAMILY	Familia de lenguajes basados en XML que define el formato o las transformaciones necesarias de un documento XML para ser presentadas en un medio.
XSD	XML SCHEMA DEFINITION	Lenguaje empleado para describir la estructura y las restricciones de los contenidos de un documento XML.
W3C	CONSORCIO WORLD WIDE WEB	Consortio internacional encargado de crear estándares web, e

		impulsar las capacidades de las nuevas tecnologías orientadas a la red mundial.
	FRAMEWORK	Entorno de trabajo conformado por un conjunto de estructuras, artefactos o módulos software, con la finalidad de resolver un problema en particular.
XMI		Estándar para el intercambio de metamodelos usando XML.

1. DEFINICION DE LA PROPUESTA DE INVESTIGACION

La investigación se ha denominado “Entorno de soporte para gestión del proceso de desarrollo de software, a través de BPM”. La estructura de esta investigación se define en los siguientes numerales de este capítulo de la siguiente forma:

1.1 DESCRIPCION DEL PROBLEMA

Para las organizaciones del software es de gran importancia el desarrollo de proyectos que cuenten con altos niveles de calidad y que se ajusten a los plazos y costo establecidos y se adapten a los permanentes cambios tecnológicos.

El desarrollo de software es percibido por el gobierno y entidades colombianas como una de las actividades potencialmente promisorias como renglón de exportación. Es por eso que se adelantan iniciativas impulsadas por el gobierno para apoyar a las empresas en programas de mejora de manera que les permita ser más competitivas en el mercado global y lograr la satisfacción de los clientes [59].

Sin embargo, lograr el desarrollo de proyectos de software de calidad es un reto complejo que requiere disciplina, capacidad de gestión y capacidad técnica. Esto implica que las organizaciones deban contar con un conjunto de procesos maduro, que haya sido institucionalizado en la organización, que pueda ser instanciado y configurado en los diferentes proyectos y que pueda ser controlado y monitoreado a lo largo del proyecto de desarrollo [4].

Algunos de los problemas encontrados que dificultan un enfoque integrado de modelado de procesos y monitoreo de los mismos en un proyecto, son los siguientes:

- Las organizaciones emplean diferentes métodos para plasmar los procesos de acuerdo a sus condiciones o preferencias. La forma más generalizada de definir los procesos es por medio de descripciones en prosa consignados en documentos de texto, o plantillas que permitan una mejor estructura del texto. Sin embargo ésta descripción en prosa es poco intuitiva como documento de trabajo por parte de los desarrolladores y poco práctica porque dificulta la evolución del mismo [4].
- A medida que las organizaciones maduran, perciben la necesidad de buscar mecanismos apropiados para presentar sus procesos por medio de diagramas de proceso basadas en UML o algún otro lenguaje gráfico. Si bien estas notaciones gráficas son más amigables de seguir por los

desarrolladores, no se cuenta con mecanismos que permitan controlar el proceso instanciado en un proyecto particular.

- Para definir las actividades básicas que comprenden el proyecto los gerentes de proyecto utilizan un EDT (Estructuras de descomposición de trabajo), y por medio de dicha estructura se define el alcance del proyecto, sirviendo como base para la planificación del proyecto. Generalmente la definición de descomposición del trabajo se hace sin considerar los procesos que se tienen definidos, lo que implica que no hay una manera ágil de verificar la adherencia al proceso que tiene el proyecto. [50].
- De otra parte, el control y seguimiento de las actividades se realiza por cada una de los equipos de trabajo de forma diferente y con múltiples herramientas desde hojas de cálculo hasta herramientas de gestión de proyectos, lo cual dificulta el análisis de resultados y la consolidación de indicadores con miras a mejoras futuras.

Cuando las organizaciones han adquirido cierta madurez, han adoptado diversas herramientas para construir los diagramas de proceso y las herramientas de gestión del proyecto. El principal problema es que dichas herramientas no se encuentran integradas y por lo tanto puede causar situaciones problemáticas [4]:

- No es posible generar el cronograma a partir del proceso.
- Pueden aparecer diferencias entre el proceso diseñado y el proceso que en realidad se lleva a cabo.
- Se hace necesario por parte de los gerentes de proyecto realizar una sincronización manual de las actividades del proyecto.

1.2 JUSTIFICACIÓN

Con el desarrollo de la presente investigación se plantea diseñar un entorno que permita automatizar el proceso de desarrollo de software, y así las empresas puedan contar con una estructura para articular sus procesos con los proyectos que desarrollen, de tal forma, que se integren y puedan ejercer un control y seguimiento automatizado, contribuyendo así en su proceso de mejora continua.

Este tipo de soluciones contribuirá al mejoramiento de la calidad de los procesos a través de la automatización de los mismos, permitiendo a las organizaciones ahorrar tiempo y ser más asertivas a la hora de planear y realizar el seguimiento de las actividades necesarias para la culminación de un proyecto específico.

Esta es una investigación de tipo práctico orientada a solucionar necesidades sentidas en empresas medianas y pequeñas que se dedican al desarrollo de software. El entorno que se propone con esta investigación, se basa en que la organización cuente con procesos definidos para la realización de las actividades propias para el desarrollo de sus productos de software, por esto es fundamental que dichas organizaciones ya cuenten con una política de calidad de mejora de sus procesos.

Afortunadamente, gracias a la preocupación que existe en las organizaciones por ser competitivas, por tener visibilidad internacional, por aumentar la calidad del software que se construye, buena parte de dichas organizaciones ya reconocen la necesidad de estandarizar los procesos de desarrollo y de adoptar buenas prácticas, que son articuladas en modelos de calidad, que organismos internacionales promueven para guiar y asegurar la calidad de los productos o servicios

Con el propósito de contextualizar y justificar este trabajo se presenta a continuación una síntesis de los esfuerzos de mejora de procesos que a nivel mundial o regional se encuentran vigentes:

CMMI: El modelo CMMI (Capability Maturity Model Integration) es uno de los modelos de procesos más influyentes relacionado con la mejora del proceso y se ha posicionado como el principal referente internacional de calidad exigido por las compañías que contratan software a nivel mundial. Este modelo proporciona una base para la evaluación de la madurez de las organizaciones de software y servicios relacionados y ofrece una guía para implementar una estrategia de mejora continua de procesos, que dan como resultado la mejora del producto [67].

La versión 1.2 del Modelo CMMI, actualmente vigente, incluye el concepto de “constelaciones”, las cuales definen un conjunto de componentes diseñados para satisfacer las necesidades de un área de interés específica. Esta versión define tres constelaciones: CMMI for Development (CMMI-DEV), enfocado en la administración, medición y monitoreo de los procesos de desarrollo; CMMI for Services (CMMI-SVC) para la entrega de servicios dentro de la organización y hacia los clientes externos; y una tercera constelación enfocada en la subcontratación, CMMI for Acquisition (CMMI-ACQ).

Para las pequeñas y medianas industrias y sobre todo en el contexto Iberoamericano, se torna difícil cualquier esfuerzo por obtener valoraciones en CMMI, debido a los costos de asesoría requeridos para interpretar e implementar el modelo, a los altos tiempos de dedicación requerido del recurso interno, y a los costos que implica el obtener y mantener este tipo de reconocimiento [50].

Para tratar de solventar este inconveniente aparecen en el contexto Iberoamericano, esfuerzos para conseguir que las empresas desarrolladoras de

software de esta región no queden en desventaja frente a las de países industrializados. A continuación se presenta un compendio de algunos de estos esfuerzos:

- **MoProSoft**¹, es el esfuerzo del gobierno Mexicano para lograr que sus empresas desarrolladoras de software aseguren la calidad en sus productos y puedan ser competitivas en el ámbito internacional, accediendo a un conjunto integrado de las mejores prácticas basadas en los modelos y estándares reconocidos internacionalmente, tales como ISO 9000:2000, CMM-SW, ISO/IEC 15504, PMBOK, SWEBOK entre otros [35][39].
- **El modelo Agile SPI**, es el principal producto del proyecto “Sistema Integral para la Mejora de los Procesos Software en Colombia-SIMEP-SW” financiado por Colciencias y la Universidad del Cauca, Popayán, Colombia. El modelo Agile SPI fue desarrollado por el grupo de investigación y desarrollo en Ingeniería de Software-Grupo IDIS- de la facultad de ingeniería en electrónica y telecomunicaciones de la Universidad del Cauca [12].
- **La metodología española Métrica v3**, desarrollada por el Ministerio de Administraciones Públicas. El objetivo de la interfaz de Aseguramiento de la Calidad de MÉTRICA Versión 3 es proporcionar un marco común de referencia para la definición y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos[20].
- **La metodología Mantema**, desarrollada en el marco de los proyectos de investigación MANTEMA (iniciativa ATYCA), MÁNTICA (CICYT 1FD97-0168) y MANTIS (CICYT 1FD97-1608) por la Escuela Superior de Informática de Ciudad Real en colaboración con varias empresas, entre ellas Atos ODS, S.A.[16].
- **El MPS.BR o Mejora de Procesos de Software Brasileño**, es un movimiento hacia la mejora y un modelo de un proceso orientado a las realidades del mercado de las pequeñas y medianas empresas de desarrollo de software en Brasil. Está basado en CMMI, ISO / IEC 12207 e ISO / IEC 15504 y la realidad del mercado brasileño. En Brasil, una de las principales ventajas del modelo es su bajo costo de la certificación de las normas extranjeras y es ideal para micro, pequeñas y medianas empresas. Uno de los objetivos del proyecto es replicar el modelo en América Latina, incluyendo Chile, Argentina, Costa Rica, Perú y Uruguay.[46].

¹ MoProSoft. Modelo de Procesos para la Industria del Software.

- **El proyecto "Tutelkan"** es una metodología de mejoramiento continuo de la calidad en los procesos de las empresas, que tiene como foco la obtención de Altos Estándares de Calidad en la Industria de Software Nacional de Chile, Utilizando Procesos de Desarrollo de Referencia.
"Tutelkan" es un proyecto de interés público, ya que apoya a empresas Pymes que exporten software, respaldándolas en su certificación de procesos, y en definitiva, mejorando sus posibilidades de expansión.

Su objetivo general es crear un mecanismo sustentable (un proceso público de referencia y una comunidad activa de apoyo) que permita a empresas nacionales definir y documentar sus procesos de desarrollo de software y sistemas con vistas a certificación ISO 9001 y evaluación CMM [53].

- **El proyecto Competisoft** (Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Ibero América) [43], es un esfuerzo Iberoamericano para las PyMEs desarrolladoras de software que busca fomentar los índices de calidad a través de un modelo de procesos alcanzable por estas y que les permita competir a nivel internacional [38]. El modelo de procesos de Competisoft está basado en el definido por MoProSoft. Tiene tres categorías de procesos: Alta Dirección, Gerencia y Operación que reflejan la estructura de una organización.

Experiencia en Colombia.

Colombia es quizás el país que, a nivel latinoamericano, ha impulsado y financiado de manera más decidida, a través del gobierno, programas masivos de mejora de procesos teniendo como referente CMMI. A continuación se presentan dos de las iniciativas recientes en las que la Universidad EAFIT, a través del Grupo de Ingeniería de Software ha tenido un papel activo.

- **El proyecto "Apoyo al Fortalecimiento de la Capacidad Nacional en Calidad de Software"** (2007-2010), es un proyecto colombiano financiado por SENA, COLCIENCIAS y PROEXPORT que logró el acompañamiento masivo a 58 empresas nacionales para que adoptaran CMMI y 18 de estas empresas recibieran la valoración oficial.
- **La conformación de la Red Colombiana de Calidad de Software:** Una de las metas del proyecto anterior es la conformación de un organismo a nivel nacional que apoye y vele por los programas de mejora de procesos en organizaciones de software y servicios relacionados. Esta red está siendo concebida como un espacio modelo para facilitar la construcción colectiva de conocimiento por parte de los diferentes actores involucrados (investigadores, consultores, personal de las empresas), con el propósito inicial de facilitar el

proceso de implementación de referentes mundiales como CMMI, ITIL, MPS.BR, entre otros, adaptado a la cultura y contexto nacional.

A continuación se presenta un cuadro resumen con algunas de las compañías colombianas que ha logrado algún nivel dentro de la clasificación de CMM.

Tabla 2. Compañías colombianas que han logrado algún nivel dentro de la clasificación de CMM.

Empresa	Nivel Actual	Referencia
Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software - CIDLIS	Nivel CMM – 2	http://www.cidlisuis.org/gci-web/entidades/CIDLIS/index.php
Fundación Cardio Vascular de Colombia (División Desarrollo)	Nivel CMM – 2	http://www.fcv.org/Portal/
Servinte - Informática en Salud	Nivel CMM – 2	http://www.servinte.com.co/
Soft - Bolívar	Nivel CMM – 2	http://www.softbolivar.com/index.html
AseSoftware	Nivel CMM – 3	http://www.asesoftware.com/
Avansoft	Nivel CMM – 3	http://www.avansoft.com/
Coomeva (Unidad de Tecnología informática)	Nivel CMM – 3	http://www.coomeva.com.co/
GestionTek	Nivel CMM – 3	http://www.gestiontek.com/
Heinsohn Software House	Nivel CMM – 3	http://web.heinsohn.com.co/www/
Ilimitada S.A. Software	Nivel CMM – 3	http://www.ilightada.info/iWebSite/iWebSite_Home.aspx
InterGrupo	Nivel CMM – 3	http://www.intergrupo.com/
MVM Ingeniería de software	Nivel CMM – 3	http://www.mvm.com.co/
Trebol Software S.A.	Nivel CMM – 3	http://www.trebol.com.co/
PSL	Nivel CMM – 5	http://www.psl.com.co/

Como se puede observar en la tabla anterior, hay un buen número de empresas constructoras de productos de software que ya asumieron el reto de emprender proyectos de mejora con reconocimiento internacional.

Se puede concluir entonces que la presente investigación es totalmente pertinente a la realidad colombiana y representa un aporte tanto para las organizaciones que ya iniciaron proyectos de mejora en modelos como CMMI o para empresas que desean iniciarse en este camino, tomando como referentes otras aproximaciones. Se pretende dar continuidad a este trabajo a través del grupo de investigación de Ingeniería de Software, en el que se enmarca este proyecto.

1.3 OBJETIVOS

Se presentan a continuación las metas que persigue la presente investigación

1.3.1 Objetivo general.

Diseñar un entorno que permita automatizar las actividades de planeación, seguimiento y control del desarrollo de un proyecto de software instanciadas a partir de la definición de un proceso BPM.

1.3.2 Objetivos específicos.

- Realizar una propuesta para integrar herramientas BPM con herramientas de gestión de proyectos, que permita obtener a partir del modelo BPM el correspondiente cronograma, asignando tiempos y actividades según el proceso de desarrollo de software que se esté adelantado.
- Permitir registrar en el cronograma, el avance de las actividades a partir de la ejecución del modelo BPM.
- Comunicar la herramienta de gestión de proyectos con la herramienta BPM para lograr realimentación entre planeación e implementación.

1.4 METODOLOGÍA

La metodología planteada a continuación, define una serie de etapas que se siguieron para dar cumplimiento a los objetivos planteados al comienzo de la investigación. Estas etapas fueron las siguientes:

1.4.1 Descripción del problema.

En esta actividad se define el problema al cual la investigación dará solución por medio del cumplimiento de los objetivos planteados dentro de la misma.

1.4.2 Recopilación de Información.

En esta etapa se realiza la recopilación de información relevante acerca de trabajos anteriores que posean una naturaleza similar a la investigación planteada por el autor. Además de documentar la base teórica de la investigación, se consolida el estado del arte de la misma y se genera un soporte de consulta necesaria para afrontar el desarrollo de la investigación. Los resultados de esta actividad se encuentran agrupados en el capítulo 2.

1.4.3 Desarrollo de Estudio de Herramientas.

Esta actividad define las herramientas que cumplan de mejor forma con las características que demanda la investigación. Para realizar esta tarea se plantean tres estudios tales como la, comparación de las herramientas de gestión de proyectos, la comparación de las herramientas de modelado BPMN y la comparación de las herramientas workflows. A partir de esta actividad se recopiló la información suficiente para elegir la mejor herramienta por cada una de las categorías correspondiente a cada estudio desarrollado. Esta fue una de las fases que implicó más esfuerzo y tiempo de dedicación del proyecto, puesto que la evaluación se hizo de manera práctica. El resultado de esta fase se resume en el capítulo 3 y en el anexo A. Cada uno de los estudios encierra una serie de actividades, definidas de la siguiente forma:

- **Estudio comparativo de herramientas de gestión de proyectos.** Se realizó una investigación de algunas herramientas presentes en el mercado, se definieron las características que poseen, a partir de las cuales se definieron las métricas necesarias para realizar la comparación entre ellas. Una vez obtenidos los resultados de aplicar las métricas a cada herramienta, se concluyó cuál es la adecuada para utilizarla dentro del desarrollo de la investigación.

- **Estudio comparativo de herramientas de modelado BPMN.** Se eligió un conjunto de herramientas de modelado BPMN, que cumplieran con alto grado de acogida dentro del mercado. A partir de esta elección se planteó un conjunto de características en común a todas las herramientas. Definidas las características, se desarrollaron las métricas necesarias para evaluar cada una de las características planteadas anteriormente. Se realizó la evaluación de cada una de las características al aplicar cada métrica y criterio para el desarrollo de la comparación. A partir de los resultados obtenidos dentro de este estudio, se definió la herramienta de modelado que cumple de mejor forma con las particularidades que la investigación demanda.
- **Estudio comparativo de herramientas workflow existentes en el mercado.** En este estudio se tomó un conjunto de herramientas workflow, las más representativas en el mercado. A partir de esta selección, se extrajeron las características comunes entre estas herramientas, se elaboraron las métricas y criterios necesarios para evaluar cada una de las características planteadas. Obtenidos los resultados de la evaluación, se definió la herramienta que proporcione las mejores características para ser utilizada en el desarrollo de la investigación.

1.4.4 Propuesta Integrada para el Modelado del Proceso y el Monitoreo de Proyecto Software.

A partir de las actividades anteriores se seleccionan las herramientas que se usarán en el diseño del entorno objeto de la presente investigación y cuyos resultados se presentan en el capítulo 4 y 5. Las actividades generales que siguió esta fase son las siguientes:

- **Definición de la Arquitectura:** Se seleccionan las estructuras y tecnologías necesarias para llevar a cabo el cumplimiento de los objetivos. Ensamble de la plataforma tecnológica: con base en la arquitectura seleccionada, se procede a realizar la instalación de los artefactos.
- **Definición del caso de estudio:** éste debe permitir evaluar las características deseables que se pretenden abordar a través del entorno.
- **Construcción del caso de estudio:** usando la notación y herramientas seleccionadas diseñar el proceso descrito por el caso de estudio y los formatos o interfaces que sean necesarias.
- **Definición de la interoperabilidad:** a través de los formatos y estándares permitir la comunicación entre el entorno y las herramientas seleccionadas.

- **Pruebas al entorno:** revisión del entorno verificando el cumplimiento de cada uno de los objetivos de la presente investigación.

2 MARCO TEORICO

La investigación se enmarca en el tema de mejora de procesos de una organización de software siguiendo los enfoques de ingeniería de procesos. Para lograr una mejor comprensión de los referentes en los que se sustenta la propuesta, es importante entrar a hacer una síntesis de los principales conceptos involucrados.

2.1 PROCESO DE SOFTWARE

Según Montlva, es el conjunto estructurado y repetitivo de actividades, que se realizan de manera sistemática y cuyo objetivo principal es el desarrollo o evolución de un software [34]. Para Fugetta, es un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software [21][55].

Una de las aproximaciones más tangible a nivel del concepto de proceso de software es la de Somerville, el cual dice que un proceso es un conjunto de pasos ordenados para alcanzar un objetivo [25]. En Ingeniería de software, el objetivo es construir un producto de software nuevo o extender uno existente [47][52].

Las organizaciones definen sus procesos basados en metodologías apropiadas para los proyectos que desean desarrollar. Las actividades que conforman dichos procesos se agrupan por disciplinas y que en general se pueden considerar así [10]:

- **Análisis y diseño:** Se define la finalidad del software o lo que debe hacer y los requisitos necesarios para su desarrollo.
- **Desarrollo:** En esta actividad se realiza la producción del sistema de software.
- **Verificación y Validación:** Comprobar que el software cumple con la finalidad y las especificaciones definidas al inicio del desarrollo.
- **Evolución:** Adecuar y/o cambiar el software según los avances y las necesidades que surjan.

- **Gestión de Proyectos:** Organizar y administrar recursos con el objetivo de culminar con las actividades necesarias para culminar un trabajo.
- **Gestión de Configuraciones:** Conjunto de procesos que aseguran la validez de los productos que se obtienen durante alguna de las etapas de desarrollo de un sistema software.

2.2 MEJORA DE PROCESOS

La mejora de procesos involucra varios enfoques en aras de crear productos de alta calidad, es así que Mouriño considera que la mejora de procesos tiene como fin aumentar la efectividad y eficiencia de una organización a través de la estructuración adecuada de los procesos, implementación y/o mejora de controles y establecimiento de planes de contingencia capaces de responder a las necesidades o problemas que surgen o afectan a la organización [2][36].

El control y mejora de procesos es un método de mejora continua de la calidad, que se basa en la reducción sistemática de la variación de aquellas características que más influyen en la calidad de los productos o servicios. Las herramientas utilizadas para la reducción de la variación son, fundamentalmente, el seguimiento, control y mejora de los procesos, cuyo resultado son estas características [52].

Al crear una estructura sólida de un proceso, se genera un buen producto, por lo tanto es necesario contar con métodos que soporten el desarrollo en mención, reduciendo tiempos muertos y optimizando recursos en lo concerniente a la detección y corrección de errores [57]. De la misma forma, la mejora debe ser progresiva para asegurar la calidad de los sistemas bajo una estructura sólida estandarizada, de ahí la existencia de varios métodos cuyo objetivo gira en torno a la mejora de la calidad de los procesos, tales como el ciclo PHVA e IDEAL.

El ciclo PHVA, involucra una serie de actividades con el objetivo de controlar los procesos. Dichas actividades están enmarcadas por acciones que determinan la intervención de cada una de ellas así:

- **Planear (P):** establece metas, métodos y la forma adecuada de abarcar los procesos.
- **Hacer (H):** hace referencia a todas las actividades involucradas en la ejecución de las tareas tal como fueron planeadas.

- **Verificar (V):** tomando como referencia los datos productos de la ejecución, se compara el resultado obtenido con las metas propuestas en la etapa anterior.
- **Actuar (A):** en ésta etapa se detectan inconsistencias a partir de la etapa de verificación. En éste punto se determinan acciones para que los problemas encontrados no se vuelvan a repetir.

De igual forma, el modelo IDEAL es un ciclo de mejoramiento de los procesos conformado por 5 fases proporcionando un conjunto de actividades para sustentar la adopción de prácticas recomendadas por el SEI como apoyo a proyectos de mejora fundamentados en CMMI. Dichas fases se reflejan a continuación:

- **Iniciar:** Su propósito radica en establecer los fundamentos necesarios para llevar a cabo la iniciativa de mejoramiento de procesos.
- **Diagnosticar:** Evaluación a través de un método formal de las fortalezas y debilidades del proceso en el desarrollo de un proyecto.
- **Establecer:** Realizar la planificación de las mejoras que se desean alcanzar. Es así que se desarrolla un plan detallado que sirve como plan del proyecto.
- **Actuar:** Implementar el plan de mejoramiento definido en la etapa anterior.
- **Difundir:** el propósito de esta etapa es la de aprender del ciclo realizado y aumentar la habilidad de la empresa para mejorar los procesos de forma continua.

2.2.1 Modelos de procesos de software

La mejora de procesos de software ha sido abordada a través del tiempo por diferentes autores, generando modelos a seguir y que según Cabrera, son una representación desde una perspectiva específica de los procesos de software, que se realizan al interior de una organización [9]. Para Ruíz además debe incluir las mejores prácticas a ejecutar [44].

Es así entonces que un modelo del proceso de software es considerado una representación abstracta de un proceso de software. Cada modelo de proceso representa un proceso desde una perspectiva particular, y así proporciona sólo información parcial sobre ese proceso. Estos modelos generales no son descripciones definitivas de los procesos de software, son abstracciones de los

procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. Puede pensarse en ellos como marcos de trabajo del proceso que pueden ser extendidos y adaptados para crear procesos más específicos de ingeniería de software [47].

Existen gran variedad de modelos de procesos de software cada uno con un grado diferente de abstracción, dependiendo de las metas buscada, lo anterior debido a que los modelos son construidos de acuerdo a características específicas, como lenguajes, abstracciones o formalismos. Por tal motivo existen categorizaciones de dichos modelos dependiendo de los elementos que representan, la información que manejan o nivel de abstracción y la metodología que implementan [1]. A continuación se resume algunas de estas concepciones, definidas a partir del análisis realizado por Jesús Galindo en la Universidad de Cataluña [11].

2.2.2 Modelos de procesos software según los elementos que representan.

El proceso de software está enmarcado por los procesos de producción y los procesos de administración. El primero se refiere a la forma como se construye el software, mientras que el segundo está encargado de definir estimados respecto a la construcción, planeación e implementación de controles a los recursos necesarios para la construcción del software. Dentro de éste contexto, existen diferentes elementos que intervienen directamente en la implementación de un modelo para construcción de un producto software. Según Universidad politécnica de Valencia los más importantes se definen así [1][51]:

- **Agentes:** Es definida como aquella entidad que ejecuta un proceso. Puede ser una persona que está involucrada directamente con el proceso software o puede ser una entidad relacionada con los componentes software o hardware que intervienen en el mismo.
- **Rol:** Es un grupo de actores, con habilidades específicas que desempeñan una labor dentro del desarrollo del proceso software.
- **Actividad:** Son las tareas o labores desempeñadas por actores que posean un rol en específico, los cuales están involucrados en una etapa del proceso de desarrollo software.
- **Artefacto:** Son productos obtenidos en alguna etapa del proceso software.

2.2.3 Modelos de procesos software según su nivel de abstracción.

Debido a que existen diferentes tipos de modelos de proceso, pueden ser categorizados dependiendo de su nivel de abstracción, es decir de la forma como delimita la representación de información. De esta forma se plantean cuatro

categorías en la cuales se define la relación funcional que poseen los elementos dentro de un proceso software, de la siguiente forma [10]:

- **Funcional:** Representa que procesos son implementados y que información es pertinente para el desarrollo del proceso.
- **Comportamiento:** Representa cuando, secuencialmente hablando, y bajo qué condiciones los elementos de un proceso son implementados.
- **Organizacional:** Representa cómo y por medio de quien son organizados los elementos de los procesos a implementar.
- **Informativo:** Representa la información manipulada por los procesos, a nivel de su estructura y relaciones que posea.

2.3 INGENIERÍA DE PROCESOS

Para contextualizar la ingeniería de procesos, es necesario tomar una de las definiciones más claras de la misma, donde se considera a la Ingeniería de procesos como el estudio completo del proceso, con los cálculos necesarios para determinar los tamaños de todos los equipos y máquinas que nos permitan alcanzar los volúmenes de producción con la calidad determinada en los estudios previos, incluso tamaño de edificios, sistema de interconexión de equipos e implantación de los mismos [17].

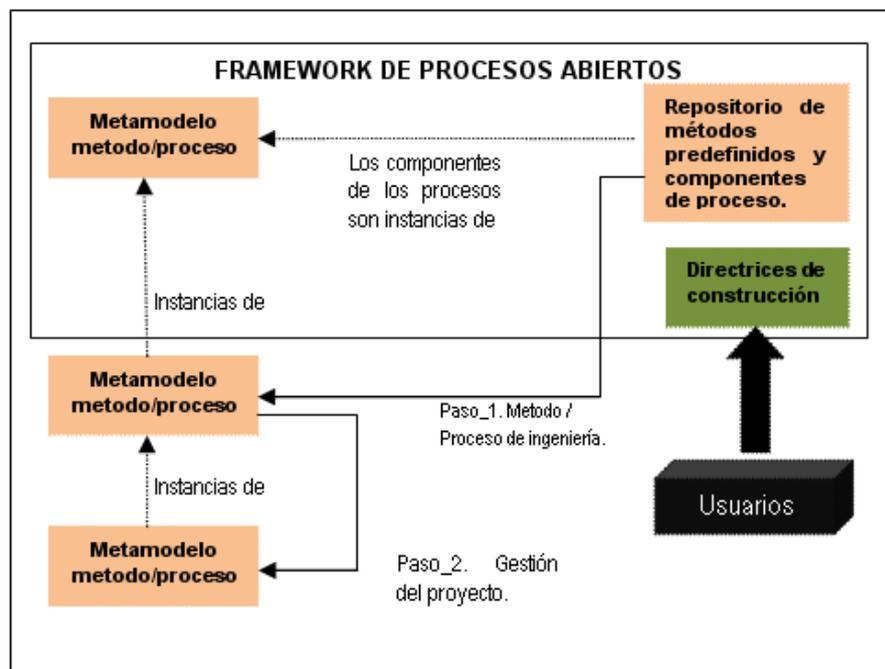
La investigación planteada hace parte de la Ingeniería de Software, por tal motivo se plantea la importancia, el desarrollo y evolución de la misma. La Ingeniería de Software desde su creación, se ha interesado por lograr que los principios de ingeniería sean aplicados al software, a través del desarrollo de diferentes métodos, herramientas y técnicas, tal como lo plantea Paloma Cáceres y otros en [12], empleando como eje central de dicha labor el software como producto. Todo ello a fin de lograr un manejo sistemático y riguroso de los artefactos, que permitirán la obtención del producto o software final [44].

Desde hace algunos años, el enfoque de la Ingeniería del software ha ido cambiando, surgiendo de esta manera una nueva área llamada Ingeniería de procesos de Software, que cumple el mismo propósito de la anterior, pero teniendo como centro principal los procesos. Su objetivo consiste básicamente en “la definición, implementación, medición y mejora de los procesos de Ingeniería del Software” [56].

Sustentada en la premisa de que “el proceso software es también software”, se empezó a utilizar el paradigma dirigido por modelos (MDE, Model Driven

Engineering), para aplicarlo al modelado del proceso software en si. Esto hizo necesario cambiar el tipo de artefacto empleado para obtener un manejo sistemático y riguroso al trabajar con los procesos de software, generando un marco de referencia genérico que se denomina, Modelo de procesos (MP). En la Figura 6 se observa la manera que se empleará ahora para trabajar con los procesos. El modelo de procesos cuenta entonces con una sintaxis y una semántica bien definida para identificar los diferentes elementos involucrados en el proceso software (ver figura 2).

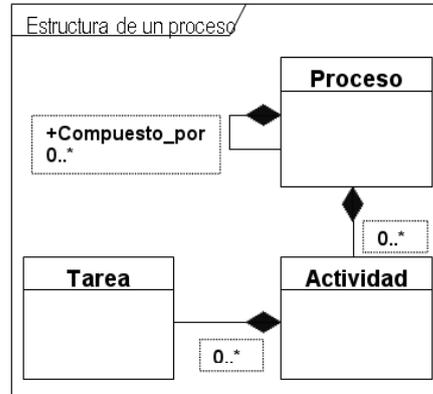
Figura 1. Marco de Trabajo para la Ingeniería de Procesos Software.



. Fuente: RUIZ, Francisco. Software Process Engineering [en línea]. La Mancha 2007 [consultado 03 de Abril de 2009]. Disponible en Internet: <http://alarcos.inf-cr.uclm.es/doc/psgc/doc/lec/parte2b/SPE-IDEA.pdf>

De igual manera como el enfoque MDD enfatiza la importancia de “programar modelos” teniendo claramente definidos los procesos de transformación, haciendo de dichos modelos cumplan un papel de gran importancia en el desarrollo de software, ahora los modelos del proceso software son los motores base del mismo, haciendo que las actuales tecnologías (MDA, MOF/XMI, UML), sean aplicadas en el campo de los procesos de Software.

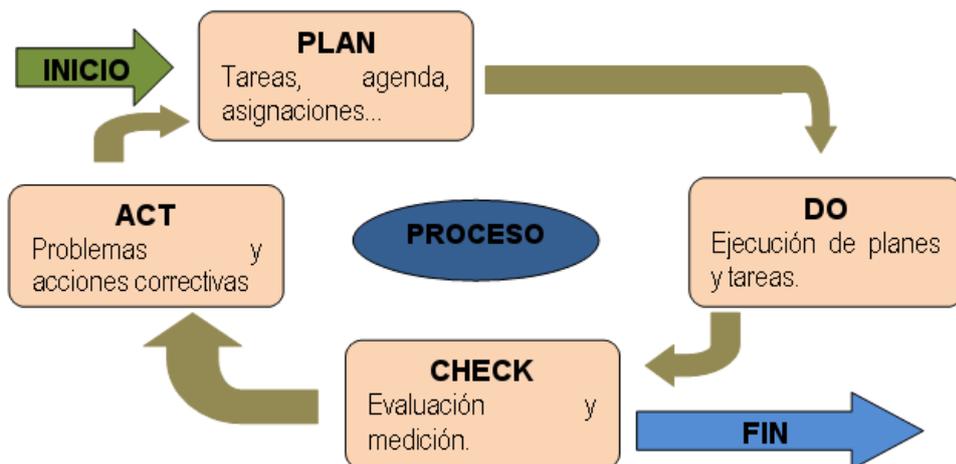
Figura 2 . Estructura de un proceso.



Fuente: BAÑERES PALACIO, Juan. Compendio de Ingeniería del Software II [en línea]. [Consultado 03 de Mayo de 2009]. Disponible en Internet: http://www.navegapolis.net/files/cis/CIS_2_04.pdf.

El ciclo de mejora PDCA "Plan –Do –Chek –Act" (Planificación, ejecución, medición y mejora) se utiliza para lograr la descomposición del proceso en actividades y tareas (ver Figura 3).

Figura 3. Estructura específica del proceso.



Fuente: BAÑERES PALACIO, Juan. Compendio de Ingeniería del Software II [en línea]. [Consultado 03 de Mayo de 2009]. Disponible en Internet: http://www.navegapolis.net/files/cis/CIS_2_04.pdf

Ahora bien, una vez reconocida la importancia de definir un marco genérico para modelar el proceso, es importante entrar a revisar dos importantes vertientes que existen: El metamodelo basado en SPEM y el enfoque basado en BPM.

2.4 METAMODELO SPEM

Es un enfoque que sigue la propuesta de MDA fundamentada en UML. Pavlich y otros autores definieron como metamodelo de procesos, al modelo de un lenguaje de modelado, que describe un conjunto de conceptos genéricos y sus interrelaciones, los cuales sirven de base para la definición de modelos de procesos. (Meta Object Facility) [48]; dicho de otra forma, el metamodelo debe abstraer de los modelos que se pueden instanciar en la realidad tanto los elementos estáticos como los dinámicos, de tal manera que se permita la creación de modelos nuevos que tengan en cuenta las características genéricas y las adecúen al ámbito según los requisitos o necesidades específicas [16].

SPEM, o Software Process Engineering Metamodel, es un metamodelo definido por la OMG² para la Ingeniería del software, considerado un estándar para establecer elementos como métodos, ciclos de vida, técnicas, actividades, procesos, metodologías entre otros. SPEM es usado para describir un proceso de desarrollo de software concreto o una familia de procesos de desarrollo de software relacionados [41].

Aunque la automatización de procesos pueda ser realizada por transformaciones del metamodelo SPEM al metamodelo de un workflow [53], para esta investigación dicha opción no fue tomada en cuenta por considerarse que es un enfoque que se encuentra aún en proceso de maduración [66].

2.5 BUSSINES PROCESS MANAGEMENT (BPM)

BPM, es considerada una disciplina empresarial cuyo objetivo es mejorar la eficiencia de las organizaciones modelando, automatizando, monitorizando y optimizando de forma continua los procesos de negocio a través de la gestión sistemática de ellos, la cual define técnicas y herramientas software para realizar dichas actividades dentro de una organización [26].

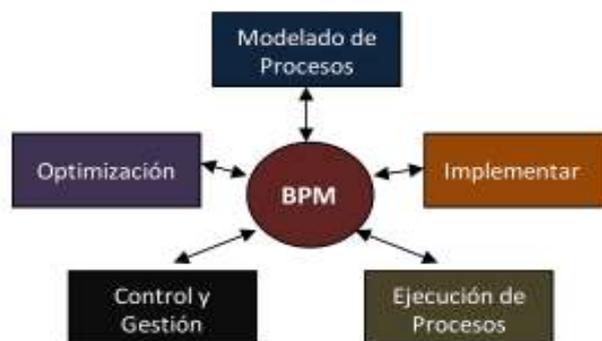
BPM, comprende una serie de actividades las cuales están definidas a través del uso de software para soportar la ejecución de procesos operacionales.

² OMG: Object Mangement Group.

Dentro de las ventajas más importantes provistas por BPM se puede mencionar la de lograr un mejor entendimiento de la lógica del negocio, reduciendo errores en el flujo de la información a través de la automatización, asegurándose que los mismos tengan el comportamiento adecuado, garantizando un control riguroso y proporcionando elementos para visualizar dicho comportamiento [22].

Esta propuesta organizacional alrededor de los procesos plantea el ciclo de vida de procesos que va desde su definición hasta su monitoreo y evolución. Las etapas generales son las siguientes:

Figura 4. Ciclo de vida BPM.



- **Modelado de procesos de negocio:** Esta etapa tiene como producto un modelo de procesos del negocio, en el cual se definen políticas, normas, operaciones, definiciones y restricciones y se definen mejoras, o cambios en los procesos para optimizarlos.
- **Implementar:** En esta etapa se integran todos los componentes necesarios para implementar los procesos diseñados. En esta etapa se define los procesos como soluciones tecnológicas a un determinado número de necesidades dentro de una organización.
- **Ejecución de procesos:** En esta etapa se puede obtener un seguimiento a los procesos implementados, intervienen todos los participantes de un determinado proceso. Como producto de esta etapa, se puede obtener información para control y seguimiento de dichos procesos.
- **Control y gestión:** En esta etapa se realiza un seguimiento más estricto de los procesos, y en donde se realiza un análisis exhaustivo de estos procesos y de su desempeño.
- **Optimización:** En esta etapa se define todas las tareas necesarias para mejorar los procesos implementados, para así tener un veredicto de su

correcto funcionamiento. A partir de esta etapa se implementan proporcionar las soluciones o correctivos para mejorar los procesos en mención.

Los campos de acción de BPM son variados, ya que la estructuración de procesos y las actividades referidas ésta disciplina proporcionan una alto grado de referencia a nivel de soluciones informáticas.

Actualmente las empresas desarrolladoras de software están incursionando en éste campo a través de la implementación de soluciones para diversas organizaciones de bienes y servicios, incluyendo entidades gubernamentales, que están soportadas por herramientas basadas en BPM. A continuación se presenta una muestra de algunas de estas organizaciones.

Tabla 3. Países que implementan proyectos BPM, fuente club del BPM.

SECTOR	PAIS	DEPENDENCIA PROYECTO
Administración Pública.	España.	Ayuntamiento de Barcelona.
	Chile.	Dirección Nacional de Servicios Civil.
	Portugal.	CTT Correos Portugal.
Alimentación.	Italia.	Danone Italia.
Banca/Finanzas.	España	Uniser Corporation.
	Argentina.	Banco de Santa Fé.
	Perú	Banco de Crédito del Perú.
	Uruguay	Banco Central de Uruguay.
Construcción.	España.	Ferrovial.
Farmacéuticas.	México	Laboratorios Kener.
Industria.	España.	Martín Martín.
	Argentina.	Grupo Galileo.
	Colombia.	Promotora de Café de Colombia, Procafecol Holding S.A.
Sanidad.	España	Hospital de Santa Creu i Sant Pau
	España.	Hospital Sant Joan Déu.
	Argentina	Hospital Universitario Austral.
Seguros.	Argentina.	San Cristóbal

		Generales.
	Colombia.	Allianz.
Servicios.	España	Aldeasa.
	España.	Caprabo.
	España.	Eurimadi.
	España.	Fira de Barcelona.
	Argentina	Metrovias S.A.
	Brasil.	Grupò Accor.
	Colombia.	Gas Natural Colombia.
	México.	Administración Portuaria Integral de Veracruz.
Telecomunicaciones.	España.	Motorola.
	España.	R Cable.
Tecnologías de la información.		Apple.

Las ventajas más importantes de la implementación de BPM, de la siguiente forma:

- **Reducción de los plazos de desarrollo de procesos:** Se puede redefinir el desarrollo de procesos, facilitando su elaboración debido a que posibilita desarrollos en paralelo, eliminando tiempos muertos, reduciendo el tiempo global de ejecución de los procesos de negocio, a través de sistemas que sean interoperables[27].
- **Optimización de costos:** Con la implementación de BPM se logra identificar tareas innecesarias y cuantificar en términos de plazos y consumo de recursos, lo anterior por medio del modelado y la definición de métricas, que estructuren elementos imprescindibles para avanzar en un proceso de reducción de costos dentro de la organización [27].
- **Integridad y calidad de procesos:** Por medio de la monitorización se garantiza que los procesos estén desarrollados según lo define los estándares previamente establecidos, asegurando la calidad e integridad de estos [27].
- **Integrar terceras partes en los procesos:** Se hace necesaria la compatibilidad de procesos, adiciona a la accesibilidad a nuevas tecnologías, permitiendo a los clientes, proveedores, y diferentes organismos, participar de forma directa y eficiente, impulsando las

características principales del flujo de la información y al acceso de los procesos [27].

- **Consolidar información:** Por medio de esta información, se logra consolidar un repositorio corporativo de información normalizada, lo que genera una base de una bodega de datos (datawarehouse) para la empresa [27].

BPM, no sólo es presentada como una disciplina abstracta, sino que recurre a una variada gama de tecnologías que permitan desarrollar las actividades propias de ella. Entre las tecnologías más relevantes a la presente investigación se define BPMN como notación de procesos, los workflow, además de funcionalidades adicionales que se explorarán más adelante dentro de la concepción del presente trabajo y que está estructurados bajo estandarizaciones como la proporcionada por la WfMC³. BPEL, XForms, WSDL y XML Schemas como tecnologías basadas en XML y que definen las herramientas dentro del flujo de información de una aplicación basada en BPM y soportada por las características y funcionalidades propias de un workflow.

A lo largo de los desarrollos de nuevas disciplinas y paradigmas, surgen tres principales tecnologías las cuales han ido convergiendo, flujo de trabajo, EAI y la Web, cada uno sin embargo, viniendo desde una perspectiva bastante diferente, y con enfoques propios que determinan su funcionalidad.

Tradicionalmente el flujo de trabajo ha puesto más énfasis en la estructura de la organización y las funciones y responsabilidades conexas. Los procesos de negocio de una perspectiva de organización con vistas a la rendición de cuentas y atributos de la responsabilidad y las funciones y responsabilidades asociadas a actividades de procesamiento de trabajo. Los recursos de trabajo por lo tanto tienden a adoptar tanto humanos y máquinas, en un solo entorno funcional.

El típico enfoque de EAI ha puesto más énfasis en la ingeniería y agentes de automatización de aspectos sofisticados y cualidades transaccionales. Los procesos de negocio suelen comenzar a partir de los flujos de datos de una perspectiva de trabajo o transaccional definiciones enfocadas totalmente a la automatización de las mismas sin intervención humana. Sin embargo, una de las características fundamentales que se requieren en el mundo de BPM, es la capacidad de apoyar la gestión flexible de las empresas. [26].

El entorno planteado en esta investigación utiliza los recursos, fundamentos y actividades planteadas por BPM, debido a que se considera un enfoque más maduro para ser aplicado en organizaciones desarrolladoras de software. De otra parte, las prácticas de mejora dentro de las organizaciones, hacen énfasis en

³ WfMC: Work Flow Management Coalition.

considerar los procesos como activos claves de la organización, que deben ser estandarizados y adecuados a las condiciones del contexto y deben ser articulados alrededor de la cadena de valor; estos aspectos ya se encuentran de manera natural en los enfoques de BPM.

2.6 BPMN

“BPMN (Business Process Modelling Notation) es un estándar de la BPMI (Business Process Management Initiative)[12], organismo que ha sido absorbido recientemente por la OMG, cuyo principal objetivo es según proporcionar una notación fácilmente comprensible por todos los usuarios del negocio, desde los analistas y los desarrolladores técnicos, hasta aquellos que monitorizaran y gestionaran los procesos“ [23].

Otros objetivos importantes que se plantea esta especificación son:

- Crear puentes entre el diseño de los procesos de negocio y la implementación del proceso.
- Que los lenguajes basados en XML para describir procesos (como BPEL4WS)
- tengan una notación gráfica.

Es importante tener en cuenta que BPMN abarca únicamente los procesos de negocio, lo que significa que otro tipo de modelos relacionados (estructura de la organización, recursos, modelos de datos, estrategias, reglas de negocio etc.) quedan fuera de la especificación [62].

Todo este tipo de modelados y sus relaciones con BPMN serán definidos formalmente conforme BPMN y otras especificaciones evolucionen, de hecho, aunque BPMN muestre el flujo de datos (mensajes) y las asociaciones de los artefactos con las actividades, no es un diagrama de flujo de datos.

Al modelar los procesos, es necesario recurrir a una notación estandarizada, ya que la solución planteada en la investigación recurre a las funcionalidades proporcionadas por BPM, es necesario incluir la notación propia de esta, es decir BPMN. En este apartado se incluye la fundamentación necesaria para modelar los procesos mencionados por medio de la notación BPMN.

BPMN es una notación gráfica estandarizada que permite modelar procesos de negocio, por medio de un workflow, proporcionando una notación legible para las partes que interactúan en un proceso de negocio, facilita los comportamientos y las decisiones para que sean expresadas de forma gráfica y simple, facilitando la comunicación entre las dependencias, roles y personas ya que los modelos

gráficos se pueden convertir en estructuras que se integren a las tecnologías de la información que las empresas posean[18].

BPMN basa su funcionalidad en BPD (Business Process Diagram), el cual contiene una serie de elementos gráficos, por medio de los cuales se modela las características de los procesos de negocio.

La propuesta planteada en esta investigación hará uso intensivo de esta notación, así que es importante entrar a analizar su notación. Según Gero Decker la notación está disgregada en cuatro categorías principales [9]:

Flow Objects (Objetos de Flujo): Un diagrama BPD, posee un pequeño grupo de elementos generales denominados objetos de flujo:

Tabla 4. Objetos de flujo BPMN.

NOMBRE	DESCRIPCION	SIMBOLO
EVENT	El evento es una representación de que algo pasa, y se compone de dos partes. La primera hace referencia a un evento que es la causa de alguna acción de impacto, resultado de la acción mencionada.	
ACTIVITY	La actividad es la representación genérica de un trabajo que realiza una organización. Puede ser una actividad atómica, o compuesta.	
GATEWAY	Una compuerta es usada para la divergencia y la convergencia de la secuencia de un flujo.	

Connecting Objects (Objetos de conexión): Para crear la conexión de los objetos de flujo es necesario recurrir a otro tipo de elementos, estos son llamados objetos de conexión, se utilizan tres tipos (Gero Decker):

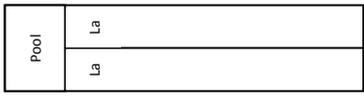
Tabla 5. Objetos de conexión BPMN.

NOMBRE	DESCRIPCION	SIMBOLO
	Es usada para mostrar la secuencia de las actividades	

SEQUENCE FLOW	que se van a desarrollar en un determinado proceso.	
MESSAGE FLOW	Es usada para mostrar el flujo de mensajes entre dos procesos independientes que participan recibiendo o enviando información.	
ASSOCIATION	Asocia datos, textos y otros artefactos con objetos de flujo. Es usada para mostrar las entradas y las salidas de las actividades.	

Swimlanes: Varias metodologías de modelado, utilizan este concepto para organizar las actividades dentro de categorías visuales separadas en orden, para ilustrar las diferentes y funcionalidades. BPMN maneja dos categorías específicas de estos objetos (Gero Decker):

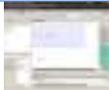
Tabla 6. BPMN swimlanes.

NOMBRE	DESCRIPCION	SIMBOLO
POOL	Representa a un participante en un proceso. También actúa con un contenedor visual de una serie de actividades de otras Pool.	
LANE	Es una sub-partición dentro de una Pool. Son usadas para organizar y categorizar las actividades.	

Artifacts (Artefactos): Los artefactos son herramientas que extienden la funcionalidad de la notación BPMN, ya que proporcionan información adicional que no sería posible a través de los objetos descritos anteriormente. Se plantean tres tipos de artefactos (Gero Decker):

Tabla 7. Artefactos notación BPMN.

NOMBRE	DESCRIPCION	SIMBOLO
DATA OBJECT	Son mecanismos para mostrar como los datos son solicitados o como son producidos por una actividad. Son conectados	

	con las actividades por medio de asociaciones.	
GROUP	Un grupo, es usado para propósitos de documentación o de análisis, pero no afectan el flujo secuencial.	
ANNOTATION	Son mecanismos que proporcionan información adicional en forma de texto, al lector del diagrama, es decir la documentación del modelo.	

2.7 TECNOLOGIAS IMPLEMENTADAS EN EL DESARROLLO DEL ENTORNO

Mientras que BPM representa el enfoque organizacional que enfatiza los procesos como el centro de las operaciones del negocio, desde la perspectiva aplicada que persigue es este trabajo, es necesario entrar a estudiar los componentes que son necesarios para que este enfoque de BPM pueda ser ejecutable y las tecnologías de apoyo que son útiles a la hora de implementar una solución basada en BPM. Estas tecnologías y herramientas proporcionaran los medios suficientes para cumplir con el objetivo principal buscado en la investigación [46].

2.7.1 Workflow.

Desde la perspectiva de este trabajo, el workflow es una tecnología cuyo objetivo primordial es el de automatizar los procesos de negocio proporcionando una estructura basada en tecnologías y herramientas para dar soporte a dichos procesos [37][28]. Consta de una serie de pasos conectados en secuencia lógica de operaciones, definiendo un trabajo para una persona o un grupo de personas, que participan activamente en una serie de mecanismos simples o complejos. Está definido como una abstracción de un trabajo real, con un propósito definido [55][50].

El workflow posee un desarrollo implícito, consiste en automatizar los procesos de negocio, automatizando la secuencia de acciones referidas a un proceso, actividades o tareas necesarias para ejecutar un proceso determinado, incluyendo la gestión involucrada en el desarrollo de los procesos [61].

Es así que dentro del concepto de workflow, se define características tales como personas, tareas, herramientas, datos e información que fluye, definiendo una arquitectura orientada a workflow regida por estamentos en particular, como la

definición de las tres “Rs” que son los roles, las rutas y las reglas, además de las tres “Ps”, procesos, prácticas y políticas[58].

El Workflow define los elementos que componen un flujo de trabajo. Contiene definiciones o declaraciones para manipular actividades. Específicas atributos pueden para la administración de datos relevantes como autor, versión, tiempo de ejecución de datos relevantes, como prioridad.

Un flujo de trabajo puede ser ejecutado como un sub-proceso invocado como una implementación de una actividad de subflujo, el tipo de parámetros en este caso se puede definir como atributos del proceso. Cuando un proceso de definición de flujo de trabajo incluye los parámetros de entrada y es instanciado por otros medios [57].

2.7.2 Monitoreo de procesos.

El entorno producto de la investigación, se basa en actividades específicas parte del proceso de desarrollo software. Como lo es el monitoreo de procesos, y permite tener control sobre las instancias de los procesos modelados a través del caso de estudio. El monitoreo de procesos es una actividad que se realiza con el fin de supervisar el funcionamiento de un proceso específico, verificando si está cumpliendo adecuadamente con la labor para la que ha sido creado [30].

2.7.3 Principios de monitoreo de procesos a través de workflow.

El monitoreo, es una tarea de gran complejidad, que tiene como objetivo realizar un seguimiento estadístico de los procesos de negocio que se estén desarrollando, el cual permitirá identificar de forma oportuna y precisa los diversos inconvenientes de operación que se presenten durante la ejecución de un proceso.

Hay diversos factores que influyen en la realización del monitoreo entre los que se encuentran los recursos, el tamaño, los actores entre otros, razón esta que llevó a que organizaciones como la Workflow Management Coalition (WfMC) y la Object Management Group (A), definieran estándares en este campo. En el caso de la WFMC, estableció dentro del Modelo de Referencia Workflow la Interface 5, que se encarga de especificar claramente, la información que se debe registrar de un proceso workflow que está en ejecución. Dicha información se denomina Common Workflow Audit Data (CWAD) y permite hacer una reconstrucción de la forma como se fueron desarrollando las actividades y eventos dentro del WorkFlow.

De igual manera, en el año 2000 la OMG realizó la publicación del documento Workflow Management Facility el cual define y administra de manera independiente una serie de interfaces para el control, ejecución, monitoreo e interoperabilidad entre procesos WorkFlow. Este documento presenta la definición

del metamodelo, que utiliza diagrama de clases para realizar su representación gráfica y se especifica mediante interfaces IDL [41].

2.7.4 Interfaces WfMC.

En el manejo de la información a través de un workflow, existen diferentes actividades las cuales se deben catalogar dependiendo de su naturaleza, pueden interactuar de forma automática o manual con el usuario. Para esto, la WfMC (Workflow Management Coalition) ha definido las siguientes interfaces que sirven de modelo de referencia para la presente investigación [55].

- **Interface 1:** Propone un cuadro en el cual se puede definir los procesos, soportando el intercambio de datos.
- **Interface 2:** Facilita la integración de aplicaciones externas con el workflow, a través de la implementación de APIs.
- **Interface 3:** Proporciona un marco genérico, para integrar diferentes servicios y aplicaciones. Soporta mecanismo de conexión, utilizando llamadas a APIs.
- **Interface 4:** Facilita la automatización de múltiples ambientes heterogéneos, brindando protocolos y herramientas de interoperabilidad de motores workflow.
- **Interface 5:** Permite monitorear la información relevante del workflow de gran importancia en actividades de auditoría y estadísticas, lo cual proporciona una visión completa del workflow, obteniendo y suministrando información a través de ella.

En el análisis de las interfaces propuestas por la WfCM, se proporciona información suficiente para manipularlas y obtener información a través de ellas, suministrando las herramientas necesarias para desarrollar la interoperabilidad propuesta en esta investigación.

El análisis de cada interfaz se convierte en una tarea tediosa, debido a la complejidad de cada una, por esto, las herramientas workflow que se encuentran en el mercado actualmente proporcionan la capacidad de suplir cada una sin necesidad de analizarlas por separado, facilitando la transferencia de información entre la herramienta de gestión de proyectos y la herramienta de modelado de procesos.

2.7.5 BPEL.

BPEL (Business Process Execution Language), es la estandarización de la ejecución de servicios web, en donde se puede realizar procesos de exportación e importación de información a través de las interfaces de servicios web, exclusivas para estas tareas.

BPEL define una notación propia para especificar el comportamiento de los procesos de negocio basados en servicios web. Se pueden describir procesos de negocio ejecutables tales como la forma en la cual interactúa un participante en un modelo real de negocios. También se pueden describir como protocolo de negocios, donde las descripciones del proceso, especifica el uso e intercambio de mensajes y el comportamiento de cada una de las partes implicadas, sin revelar el comportamiento interno.

A través de BPEL se proporciona funciones para manipular datos y definir de forma sencilla como serán solicitados y procesados dentro de un control de flujo definiendo los procesos de negocio utilizando un lenguaje basado en XML, pero no definen una representación gráfica de los procesos o proporciona una metodología de diseño con lo referente a los procesos.

2.7.6 XFORMS.

XForms es un formato basado en XML diseñado por la W3C con el cual un usuario puede definir interfaces, especialmente si son formularios web. Proporciona grandes prestaciones respecto a su funcionalidad ya que puede ser usado de manera independiente, para describir cualquier tipo de interfaz de usuario, o tareas de manipulación de datos e información.

Uno de los mayores logros encontrados en la implementación de Xforms es poder separar, el propósito para el cual está diseñado un formulario web de la forma en que éste es presentado, con lo cual se obtiene ventajas como:

- Los formularios diseñados pueden ser reutilizados.
- En el momento de desarrollar una interface, ésta posee sólo métodos abstractos, facilita su representación de ésta en cualquier dispositivo que soporte ésta tecnología.
- Separa presentación y contenido, facilitando la mantenibilidad y la ayuda técnica a los usuarios en el momento de navegación.

Xforms, es presentada como la evolución de html, ya que puede realizar las acciones que éste último proporciona, además puede realizar acciones que html no puede realizar, tales como:

- Realizar una validación de los datos de entrada en el momento en el que el usuario esté digitándolos.
- Puede integrar servicios web a través de SOAP o XML RPC.
- Utiliza el resultado de cualquier salida para luego ser enviada como entrada a otro formulario posteriormente.
- Obtener los datos con los cuales se inicializará un formulario, a partir de un archivo externo o un archivo de configuraciones.
- Realizar operaciones sobre los campos, ya sean por ejemplo cálculos matemáticos.
- Mejora la experiencia de navegación de los usuarios.
- Acopla diferentes tecnologías XML.
- Facilita la creación de formularios complejos, minimizando los costos de desarrollo y el tiempo empleado para realizarlo.

Al modelar el caso de estudio planteado por el autor fue necesario recurrir a interfaces de usuario que interactuaran con las personas que manejaran el sistema. Intalio, define dos opciones para realizar dicha tarea. La primera es Ajax Forms y la segunda es XForms. Se escogió XForms debido a todas las características que proporciona, como se planteo anteriormente.

2.7.7 WSDL.

WSDL (Web Service Description Language), es un formato XML usado para ubicar y localizar servicios web en la red, además de documentarlos o describirlos. Su objetivo principal es determinar que puede hacer el servicio web, donde reside y como invocarlo, tareas para las cuales XML no basta, debido a que es necesario contar con patrones para realizar las acciones mencionadas.

WSDL, proporciona una interfaz pública a los servicios web, describiendo la forma en que se van a comunicar, es decir, define los protocolos y formatos en los cuales interactúa la información solicitada, casi siempre lo realiza a partir de la

definición propuesta por SOAP, de la misma forma que es interpretado por Earl Thomas en su libro “**SOA Design Patterns**”.

WSDL, posee un conjunto de elementos por medio de los cuales se puede indagar acerca de que puede hacer un servicio web, donde reside y como invocarlo. Esos elementos son (Garcia Yurena, Mendizábal Hevia)

- **Types:** Contenedor de definiciones del tipo de datos que utiliza algún sistema de tipos (por ejemplo XSD⁴).
- **Message:** Definición abstracta y escrita de los datos que se están comunicando.
- **Operation:** Descripción abstracta de una acción admitida por el servicio.
- **Port Type:** Conjunto abstracto de operaciones admitidas por uno o más puntos finales.
- **Binding:** Especificación del protocolo y del formato de datos para un tipo de puerto determinado.
- **Port:** Punto final único que se define como la combinación de un enlace y una dirección de red.
- **Service:** Colección de puntos finales relacionados.

2.7.8 XML SCHEMA.

XML Schema, es un lenguaje con el cual se describe la estructura, el contenido y la semántica que poseerán los documentos XML, ampliando las posibilidades de control sobre éstos.

Al implementar éstas características se obtienen ventajas tales como:

- Al tener control y restringir el contenido de los documentos XML, se puede compartir información entre varias aplicaciones con seguridad.
- Ya que posee un enfoque modular igual que la programación orientada a objetos, facilita la reutilización de código.
- Permite comprobar la integridad de los datos continuamente.

⁴ XSD: XML Schema Definition.

- Permite especificar los tipos de datos que se van a manejar.
- Permite arquetipos, o tipos definidos por el usuario.
- Es posible agrupar atributos, haciendo más comprensible el uso de un grupo de aspectos de varios elementos distintos.
- Se puede trabajar con namespaces.

Dentro de la investigación se incluyeron los esquemas para el modelado del caso de estudio, para de esta forma definir las reglas para el flujo de información entre los procesos.

2.7.9 Gestión y monitoreo de procesos software.

La gestión y monitoreo de proyectos software es una disciplina que recoge un conjunto de principios, lineamientos y prácticas para gestionar los proyectos relacionados a la construcción y mantenimiento del software [7].

Las herramientas de gestión son consideradas como un conjunto de sistemas, controles, metodologías y demás herramientas que ayudan a las organizaciones en tareas como registro de datos, control y mejora de procesos, consolidación de información y toma de decisiones. Constan de herramientas que proporcionan la capacidad de gestionar proyectos, tanto en recursos humanos, recursos físicos, manejo de tiempos, asignación de roles, tareas, tiempos de ejecución, administración de progresos en el desarrollo de actividades, es decir la calendarización del proyecto.

Estas herramientas son soportadas por arquitecturas como SOA, y se valen de tecnologías como XML.

2.8 INTEROPERABILIDAD

El objetivo de la interoperabilidad dentro de la investigación es garantizar al usuario la transparencia en el uso de herramientas heterogéneas, de forma que pueda acceder a sus funcionalidades sin conocer la forma en que se realiza el intercambio de datos entre estas [24].

Para lograr que se den dichas condiciones, organizaciones como la WC3(WC3, 2009) crean estándares, que definen especificaciones técnicas que pueden ser utilizadas ampliamente y de forma gratuita, como es el caso de HTTP, HTML,

	3														
Tolk, Muguira	2003	x		x				x	x						x
Bermudez	2004			x											
Shekhar	2004		X	x				x							
Schekkerman	2004	x	X	x											
Stroetmann	2005			x				x							
Ding	2005			x				x							
Nowak	2005		X	x				x							
Mohammadi	2006	x			x					x	x	x			
Kalantari	2006	x		x								x			
Assche	2006					x	x	x	x	x					
Turnitsa & Tolk	2006	x		x				x	x					x	x
Dekkers	2007	x		x	x										

Nivel técnico: se centra en permitir la transferencia de datos, la integración de servicios, la seguridad y accesibilidad, mediante el uso de protocolos. [49]. Se considera el nivel más básico de integración [31].

Nivel esquemático/estructural: establece la posibilidad hacer interoperables dos o más sistemas, independientemente del tipo de ambiente en el que se encuentren, ya sea cerrado o distribuido. Este nivel de interoperabilidad se establece mediante protocolos de intercambio y acceso a redes de datos como TCP/IP [43]:

- Nivel semántico: define una interoperabilidad a nivel del significado de la información, para que sea interpretada y entendida de la misma forma por los sistemas implicados.[49][13]. Involucra el uso de estándares o especificaciones como WSDL, SOAP, GML, entre otros [31].
- Nivel organizacional: se establece mediante acuerdos entre las organizaciones, que apunten al cumplimiento de los objetivos propuestos.

Se establecen políticas y procedimientos para la ejecución de las diferentes actividades [31].

- Nivel sintáctico: emplea formatos estandarizados como XML y esquemas como XSD, para establecer la interoperabilidad mediante formatos comunes [31].
- Nivel pragmático: permite la exploración y uso de métodos y procedimientos entre sistemas, empleando las taxonomías de servicios como WFS, WCS, WINS, SOS del OGC, y las interfaces de exploración definidas por los estándares y especificaciones [31].
- Nivel dinámico: se define cuando un sistema tiene la capacidad de corregir su funcionamiento de forma automática, cuando un servicio no es capaz de soportar una transacción, permutándolo por otro que pueda realizar la tarea, aprovechando la información proveniente de los sistemas con los que se establece la interoperabilidad [31].
- Nivel conceptual: el objetivo de este nivel de interoperabilidad es poder entender y explicar el funcionamiento de un sistema a partir de la documentación del mismo, sin por ello depender de los modelos [31].

2.8.2 Interoperabilidad mediante estándares.

Para lograr la interoperabilidad entre diferentes aplicaciones se han definido algunos estándares que permiten llevar a cabo este proceso, algunos de los cuales se describen a continuación [15]:

XML (eXtensible Markup Language): 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. Creado en el año de 1996 por la 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 [54].

XML realiza algunas tareas tediosas especificadas por el lenguaje, como la validación de datos o el recorrido de la estructura, lo que ayudará que al momento que programadores efectúen un trabajo con los datos, enfoquen su atención en las tareas de mayor importancia. De igual manera XML, no tiene etiquetas predefinidas lo que permite al diseñador elegir las etiquetas adecuadas según sus necesidades [39]

Otro punto a destacar de XML es que etiqueta e identifica el contenido, dejando a un lado inicialmente la parte de presentación. De esta manera permitirá tener más

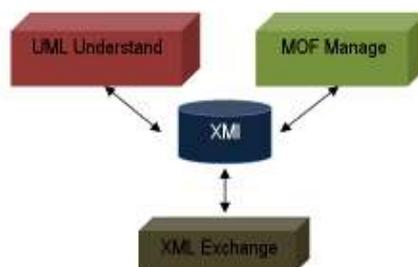
de una vista del contenido, usando para su presentación XSL (Extensible Style Sheet Language) [39]

XML, emplea tecnologías como XSL, lenguaje extensible de hojas de estilo y se encarga de la presentación del contenido, estableciendo la forma en que se debe diseñar, y como se debe paginar en los diferentes medios de presentación; *XPat*, que es el lenguaje de rutas XML y permite el acceso a partes del documento XML; *XLink*, que es el lenguaje de enlace XML y brinda la posibilidad de insertar elementos en documentos XML con el fin de crear diferentes enlaces entre recursos XML; *XPointer* que es el lenguaje de direccionamiento XML y da acceso a los elementos, atributos y contenido de XML, es decir a su estructura interna; y *XQL* que es el lenguaje de consulta XML, y posibilita la ejecución de consultas en los documentos XML, permitiendo a su vez la extracción de datos de documentos XML en la WEB. [54].

XMI (XML Metadata Interchange) es un estándar que permite intercambiar tanto información como metainformación entre diferentes herramientas basadas principalmente en UML, y repositorios de metainformación, basados en MOF, creando un formato de intercambio para entornos de tipo distribuidos. Fue desarrollado por OMG y en sus inicios, se empleaba para la metainformación de modelado y programación, pero ahora se ha implementado en áreas como datawarehouses, componentes entre otros [39].

XMI al incluir tres estándares como XMI, UML y MOF, brinda a los desarrolladores de sistemas distribuidos la posibilidad de compartir diferente información sobre internet. Así es posible usar la web para realizar publicaciones estándar empleando para la parte de diseño de los metamodelos los estándares UML y MOF y para realizar la transferencia de información el estándar XML. En la Figura 5 se puede apreciar la integración de estándares que XMI lleva a cabo.

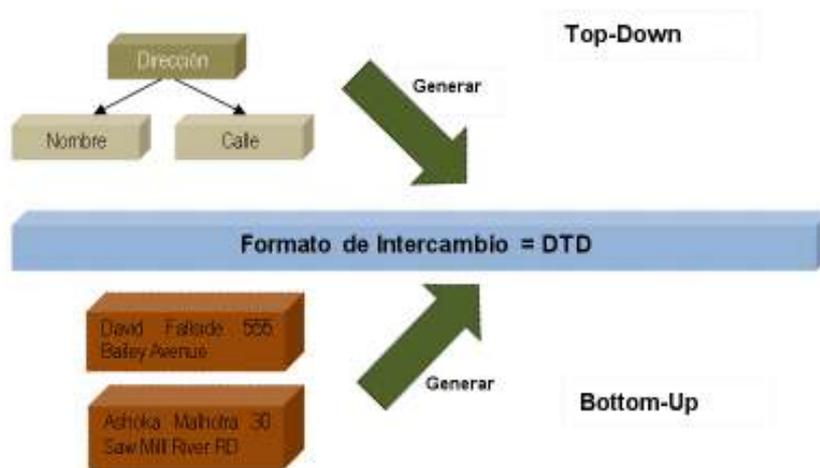
Figura 5. Integración de estándares.



Fuente: Pérez J, García M. XMI: XML Metadata Interchange [en línea]. Valencia España [consultado 21 de Abril de 2009]. Disponible en internet: www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/192000.doc

Se pueden realizar diferentes tipos de intercambios, tanto de arriba hacia abajo como de abajo hacia arriba tal como se aprecia en la Figura 2.

Figura 6. Tipos de intercambios.



Fuente: Pérez J, García M. XMI: XML Metadata Interchange [en línea]. Valencia España [consultado 21 de Abril de 2009]. Disponible en internet: users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/192000.doc

SOA (Arquitectura Orientada a Servicios): es una arquitectura de software que busca lograr la integración de aplicaciones independientes, mediante el uso de diferentes servicios que permitirán acceder a las funcionalidades de los sistemas a través de la red. [29] [33].

Una de las características más importantes de SOA es la interoperabilidad, la cual se logra, a través de la aplicación de un conjunto de estándares abiertos como WSDL, SOAP y UDDI, que buscan crear un enfoque común para definir, publicar y utilizar Web Services. Dichos estándares se basan en XML [15].

La organización que se encarga de promover la interoperabilidad entre los diferentes servicios web es la WS-I (Web Services Interoperability Organization) que fue creada en febrero del año 2002 y complementa otras entidades de estandarización tales como el W3C y el IETF.

WS-I se fundamenta en los estándares basados en XML y tiene como fin la creación de guías y herramientas que faciliten el desarrollo de servicios web interoperables. De igual manera esta organización conforma varios grupos de trabajo (Working Groups, WG), que se encargan de generar documentos más específicos, como perfiles, escenarios de uso y casos de uso, aplicaciones de ejemplo y herramientas de test entre otros, que buscan orientar a los desarrolladores para lograr alcanzar la interoperabilidad [29].

2.9 ANTECEDENTES DE INGENIERÍA DE PROCESOS EN EMPRESAS DE SOFTWARE

Las empresas dedicadas al desarrollo de software, han buscado la manera de automatizar la ejecución de diferentes procesos, con el fin de lograr un aumento en la calidad, eficiencia y rendimiento de los productos que realizan, de igual forma que Somerville lo plantea en [19]. Para ello, se han realizado diversas investigaciones, algunas de las cuales incluyen la utilización de sistemas workflow, para dar soporte al proceso de desarrollo de software, tal es el caso del trabajo elaborado por Daniel K. C. Chan y Karl R. P. H. Leung, quienes en el año de 1997, crearon Valmont, un lenguaje de especificación workflow, fruto de la transformación del "ispw-6software process example". Esta investigación, se centra exclusivamente en la parte de modificaciones a los requisitos [11].

De igual manera, en otras investigaciones realizadas en el año 2000 por Anthony Barnes y Jonathan Gray, buscaron crear una herramienta para la gestión del proceso de desarrollo de software, empleando Workflows. Para ello los autores tomaron como ejemplo la herramienta de gestión de procesos de software Process-centered Software Engineering Environment (PSEE), que trabaja sobre un sistema de gestión de bases de datos relacionales (RDBMS) y crearon el metamodelo PSEE, que es soportado por la Interface I de la Workflow Management Coalition (WfMC)[50].

Para el año 2003, Fang Min, Ying Jin, Wu Minghui, propusieron un marco de trabajo de una plataforma que brinda soporte al proceso de desarrollo de software, permitiendo a la vez controlarlo de manera automática. Para ello realizaron un metamodelo guiados por *Power Systems Analysis Framework* (PSAF), el cual empleaba para su ejecución un motor de workflow [58].

Luego en el año 2006, Debnath, N., Riesco, D., Cota, M.P., García Pérez-Schofield, J.B y Uva, D.R.M, plantearon una correspondencia entre los metamodelos Software Process Engineering Metamodel (SPEM) y el UML Extended Workflow Metamodel (UMLEWM)[14], y en este mismo año Audris Kalnins y Valdis Vitolins publican Use of UML and Model Transformations for Workflow Process Definitions en el que se emplea el lenguaje Model Transformation Language (MOLA) para crear una transformación formal del Lenguaje Unificado de Modelado (UML) a *Business Process Management* (BPM).

Otras investigaciones que vale la pena citar, son las realizadas por Fabio A. Zorzan y Daniel Riesco, quienes en el año 2007 elaboraron una propuesta para lograr la automatización de procesos de desarrollo de software a través de SPEM, empleando motores de workflow [42]. Un estudio similar fue realizado por Martínez Marcelo en el año 2008, en el cual se automatizaba el proceso de desarrollo de

software empleando un motor de workflow que permite administrar los recursos y organizar al equipo durante el desarrollo.

Del mismo modo en el año 2008 F. A. Zorzan y D. Riesco, elaboraron un trabajo que tenía como fin principal, la transformación de los procesos de desarrollo de Software que se han especificado con SPEM a procesos de workflow que se basan en el estándar aceptado por la Object Management Group (OMG), Business Process Modeling Notation (BPMN)[60][59].

También es conveniente mencionar los trabajos realizados por Guipss Jeannira Menéndez Pérez, y Miguel Angel Sic Garcia, donde ellos trabajaron el análisis de workflow y groupware aplicados a los sistemas integrados de gestión pública en Guatemala, donde muestran la utilidad y aplicabilidad de los sistemas integrados a través de la implementación de herramientas workflow o groupware enfocados al sector público en Guatemala. Dicha tarea la realizaron a través de los aspectos teóricos y prácticos de los workflow, su evolución, modelo de implementación, componentes y utilidades [32].

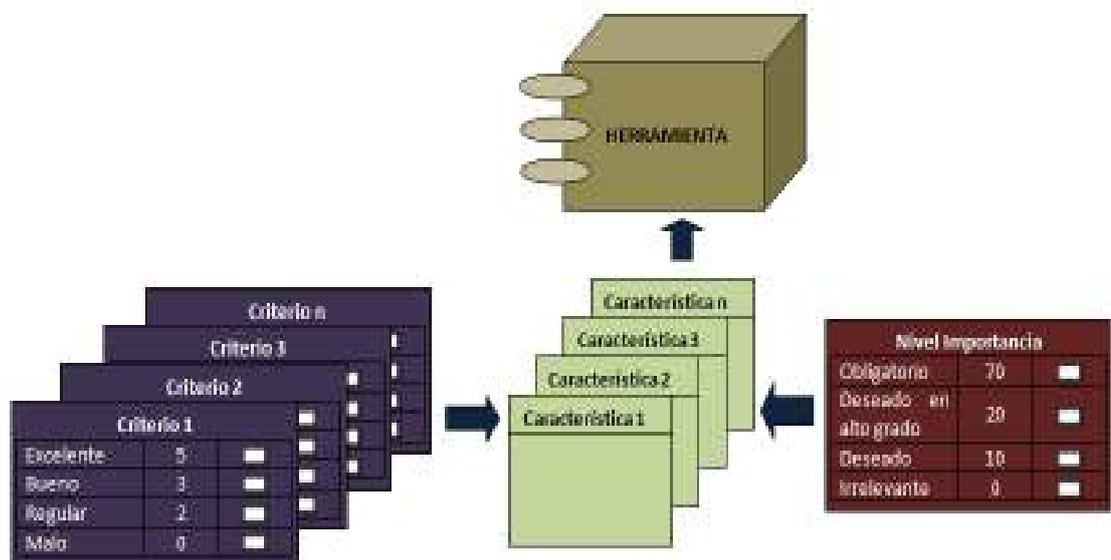
Otro proyecto que entra en el ámbito de esta investigación, la proyecto Batuta, desarrollado en la facultad de Ingeniería de la Universidad de la República, por Pablo Alves, Patricia Foti y Marco Scalone, en el año de 2006, estudio realizado acerca de las herramientas y aplicaciones orquestadoras [3].

El proyecto más cercano a la idea pretendido por parte de la investigación es el planteado en el trabajo "Un workflow que automatice los procesos de negocio del Proceso Unificado Rational", desarrollado por Daniel Riesco y Daniel Romero en el año 2004, este trabajo hace referencia a la automatización de las actividades que se realizan a nivel industrial, para mejorar la producción, la calidad y garantizar el correcto cumplimiento de las reglas del negocio. Así, definieron como objetivo principal la automatización de la producción de software siguiendo las reglas definidas en RUP, e implementando un workflow [41].

3. ESTUDIO COMPARATIVO DE HERRAMIENTAS

Es necesario para el desarrollo del entorno objeto de la presente investigación contar con herramientas de gestión de proyectos, de diseño BPM y motores de workflow. Al ofrecer el mercado una variedad muy alta de este tipo de herramientas se vuelve compleja su selección, en especial cuando se busca características especiales que se adecuen a las necesidades del proyecto, por lo tanto hizo necesario realizar los estudios de las diferentes categorías de herramientas de acuerdo al esquema que observa en la figura 7.,

Figura 7. Esquema general de los estudios comparativos.



El análisis comparativo de herramientas siguió los siguientes pasos generales.

Paso 1. Definición de criterios. Estos criterios fueron definidos a partir del estudio de los referentes. Estos criterios permiten evaluar el nivel de importancia que puede llegar a tener las características de cada herramienta, respecto a las prestaciones que son ideales para el entorno a desarrollar (ver tabla 9). Estos valores fueron asignados buscando que las herramientas que contengan las características necesarias en el desarrollo del proyecto obtengan un puntaje diferencial respecto a las que no, y que aquellas que ofrecen funcionalidad adicional puedan mejorar su puntaje y así contar con la mejor selección posible.

Tabla 9. Nivel de importancia de características.

Nivel de Importancia	Peso
Obligatorio	70
Deseable en alto grado	20
Deseable	10
Irrelevante	0

Paso 2. Definir relevancia de cada criterio. En el siguiente paso se deben relacionar las características de cada tipo de herramienta que serán evaluadas para el desarrollo del estudio comparativo, definiendo su obligatoriedad respecto a las necesidades del entorno a diseñar.

Tabla 10. Características a evaluar.

CARACTERISTICA	OBLIGATORIA
Característica 1	SI
Característica 2	SI
Característica 3	NO
Característica 4	SI
Característica 5	NO

Paso 3. Ponderación de cada criterio. Definidas las características a evaluar, se procede a calificar cada una de acuerdo al nivel de importancia y se calcula su valor de ponderación (ver tabla 11).

Tabla 11. Ponderación de características.

CARACTERISTICA	NIVEL DE IMPORTANCIA	PONDERACION
Característica 1	Valor1	Porcentaje de importancia 1
Característica 2	Valor2	Porcentaje de importancia 2
Característica 3	Valor3	Porcentaje de importancia 3
Característica 4	Valor4	Porcentaje de importancia 4
Característica 5	Valor5	Porcentaje de importancia 5
Totales	Total_Valores	Porcentaje total de importancia = 100%

La columna de *características* se refiere a cada una de las características a evaluar en cada uno de los estudios, la columna nivel de importancia, representa el estimado que posee cada característica, se basa en la tabla Nivel de Importancia de características (ver tabla 9), y de su obligatoriedad. A partir de las anteriores se obtiene los valores de la columna de ponderación, de la siguiente forma:

- Se suma todos los valores correspondientes a la columna *Nivel de Importancia*, en donde se obtiene el valor de la celda *Total_Valores*.

$$Total_Valores = \sum_{1}^{n} Valor(n)$$

- A partir del anterior valor se aplica la siguiente fórmula:

$$Ponderación = Valor * \frac{100}{(Total_Valores)}$$

- Esta fórmula es aplicada a cada una de las características evaluadas.

Para poder calificar las características se definen unos criterios por cada una de ellas, que serán valorados de acuerdo a su desempeño según una escala general de evaluación (ver tabla 12).

Tabla 12. Escala general de evaluación.

Criterio	Peso
Excelente	5
Bueno	3
Regular	2
Malo	0

Para cada característica, se definen los criterios específicos a tener en cuenta para asignación del puntaje (ver tabla 13).

Tabla 13. Criterios específicos de evaluación.

Característica n	
Peso	Criterio
5	Criterio 1.
3	Criterio 2.
2	Criterio 3.
0	Criterio 4.

Como se puede observar, cada criterio definido posee un peso correspondiente a los valores contemplados en la Escala General de Evaluación.

Paso 4. Análisis de resultados. A partir de la declaración planteada en la tabla anterior, se procede a aplicar la evaluación de las características basadas en los criterios específicos de evaluación propios de cada una, se obtiene la tabla de resultados consolidados correspondientes a la tabla 14.

Tabla 14. Resultados consolidados.

PONDERACION	CARACTERISTICA	Herrami enta1	Herrami enta2	Herrami enta3	Herrami enta4
Porcentaje1	Característica1	V1H1	V1H2	V1H3	V1H4
Porcentaje2	Característica2	V2H1	V2H2	V2H3	V1H4
Porcentaje3	Característica3	V3H1	V3H2	V3H3	V3H4
Ponderación total = 100%	RESULTADOS	Resulta do1	Resulta do2	Resulta do3	Resulta do4

V1H1 = Valor número uno de la característica uno de la herramienta uno.

Las celdas concernientes a V1H1, V1H2, V1H3, V1H4, representan los valores correspondientes a la evaluación de la característica 1 presentes en cada una de las herramientas escogidas, basados en los criterios específicos de evaluación de cada característica contemplados en la tabla 12.

Los resultados obtenidos para cada característica surgen de aplicar la siguiente fórmula:

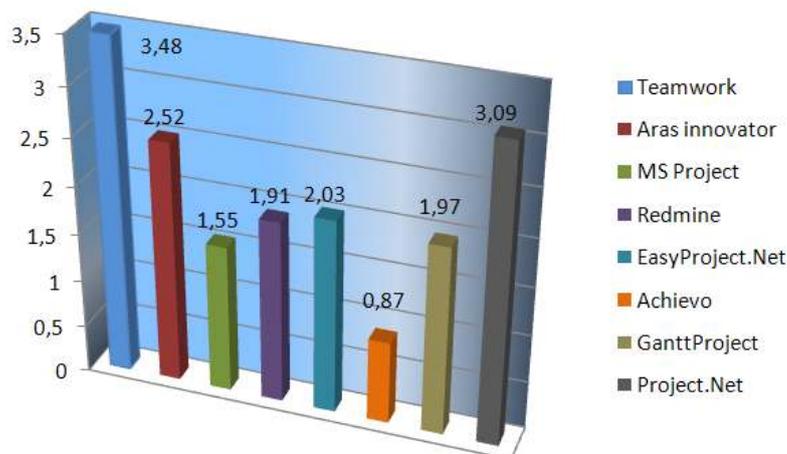
$$Resultado1 = \sum_1^n (VnH1 * Porcentaje(n))/100$$

A partir de dichos resultados se toman las decisiones concernientes para elegir la herramienta que mejor se acomode a las necesidades demandadas por la investigación.

3.1 ESTUDIO COMPARATIVO HERRAMIENTAS DE GESTIÓN DE PROYECTOS

Esta segmento hace referencia al estudio realizado para elegir la herramienta de gestión que mejor se integrara dentro del desarrollo de la investigación, tal como lo muestra el anexo A en el subtítulo “*Estudio comparativo de herramientas de gestión*”. A partir de dicho estudio se obtuvieron los resultados correspondientes a la **tabla 33 del anexo A**, cuyo resumen gráfico puede verse reflejado en la figura 8.

Figura 8. Gráfica de resultados consolidados para herramientas de gestión de proyectos.



La figura 8. Corresponde a los resultados obtenidos por cada una de las herramientas en el estudio. Dichos resultados se basan en una escala de

evaluación de cero a cinco. Y cada fragmento corresponde al valor obtenido por cada herramienta.

Con los resultados arrojados por el estudio, se pueden tener las siguientes consideraciones:

- A partir de las argumentos por medio de los cuales se realizó la evaluación de las herramientas dentro del estudio planteado, Achievo es la peor opción al momento de implementarla en el entorno de soporte para el desarrollo de proceso de software a través de BPM, por tal motivo fue descartada.
- De los resultados obtenidos es posible mencionar que GantProject, MS Project y Redmine, proporcionan aproximadamente la misma capacidad para ser implementada en el desarrollo de la investigación como está reflejado en los resultados obtenidos en la **tabla 33 del anexo A**, pero no son aquellas que poseen los mejores puntajes según el estudio en mención, por consiguiente fueron descartadas.
- Project.Net, Teamwork y Aras innovator, son las herramientas que presentan de mejor forma las características que una herramienta de gestión de proyectos pueda ofrecer para ser parte del desarrollo de la investigación, pero claramente existe una diferencia entre ellas, por tal motivo se plantea la figura 9, correspondientes a el análisis de resultados realizado sobre estas tres herramientas (ver figura 9).

Figura 9. Gráfica de resultados herramientas de gestión de proyectos.



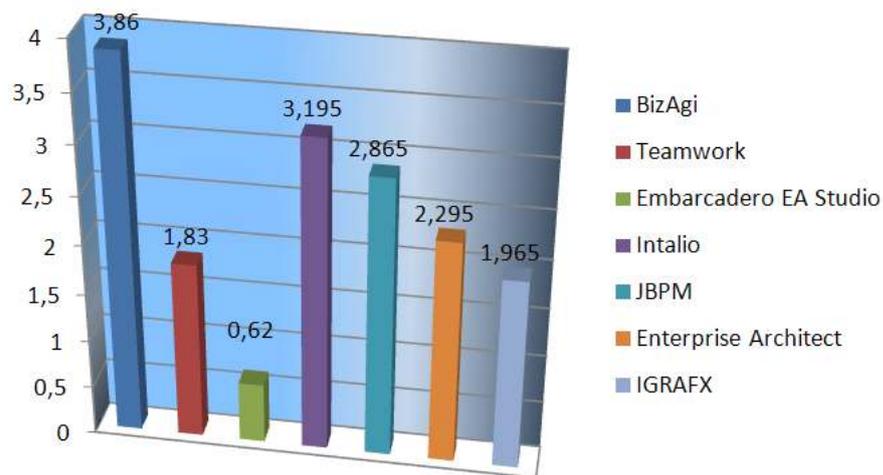
Se puede realizar las siguientes observaciones:

- Project.Net es la herramienta que obtuvo los resultados más altos en comparación con las otras dos herramientas, por tal motivo fue decisión, que fuese la herramienta a implementar en el entorno de soporte para el desarrollo de software a través de BPM.
- De igual forma se presenta que según la evaluación desarrollada es la herramienta que posee una estructura compacta con lo referente a las características demandadas por la investigación.
Aras innovator aunque posee una gran puntaje en las características necesarias para el desarrollo de la investigación presenta una estructura fluctuante con respecto a Project.Net y Teamwork por tal motivo aunque sus resultados sean altos, no sería una buena opción para ser implementada dentro de la investigación.

3.2 ESTUDIO COMPARATIVO HERRAMIENTAS DE MODELADO BPMN

El estudio realizado sobre las herramientas de modelado BPMN [52], para elegir la que mejor cumpla con las características que solicita la investigación, puede ser analizado a partir del subtítulo de “*Estudio comparativo de herramientas de modelado BPMN*” del anexo A. A partir de dicho estudio se puede definir la figura 10.

Figura 10. Gráfica de resultados consolidados.



La figura 10. Corresponde a los resultados obtenidos por cada una de las herramientas en el estudio. Dichos resultados se basan en una escala de

evaluación de cero a cinco. Y cada barra corresponde al valor obtenido por cada herramienta.

La figura 10 y los resultados de este estudio expuestos en la **tabla 55 del anexo A**, proveen de la información suficiente para definir las siguientes consideraciones:

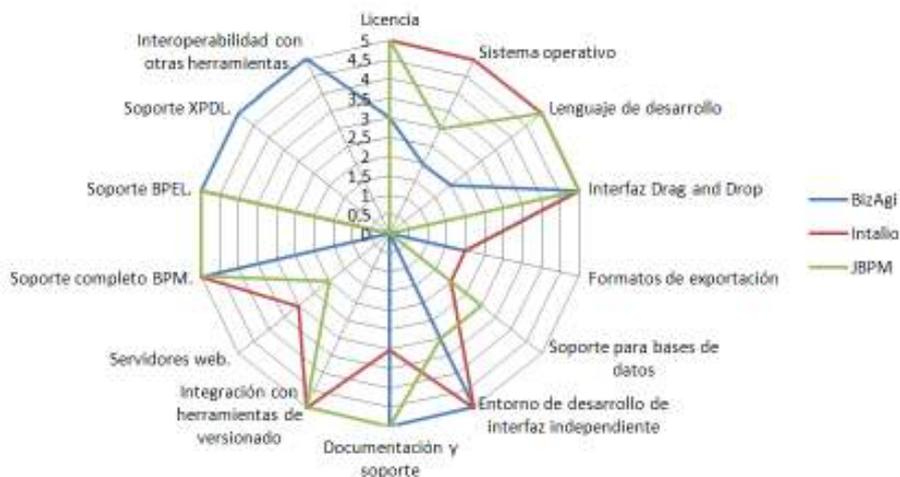
- BizAgi, es la herramienta de modelado BPMN que presenta mejores características necesarias para ser implementada dentro de la investigación para el entorno de soporte para el proceso de desarrollo de software a través de BPM, pero no fue elegida debido al tipo de licencia manejada por esta que aunque sea una característica que no representa mayor significación trae en duda el implantar esta herramienta en el desarrollo de la investigación.
- Intalio designer, resulta ser la mejor herramienta de modelado BPM para ser parte del entorno de soporte planteado en la investigación, por tal motivo se eligió para desarrollar dicha tarea.
- De igual forma JBPM es una muy buena opción para ser integrada dentro de la investigación, aunque la diferencia de resultados entre esta e Intalio no presenta mayor diferencia, pero podría presentar inconvenientes en su implantación dentro de la investigación.

La gráfica anterior puede ser mejor analizada a partir de la figura 11, la cual expone los resultados obtenidos por las herramientas que obtuvieron los mayores resultados según el estudio de herramientas de modelado BPMN.

Como se puede observar en la figura 11, las herramientas que poseen los mayores puntajes según el estudio, arrojan los siguientes argumentos:

- Las herramientas que tuvieron el mejor comportamiento en las características de "*Lenguaje de desarrollo*" y "*Interfaz Drag and Drop*", fueron Intalio y JBP, sino que según la gráfica se muestran sobrepuestas.

Figura 11. Gráfica de resultados herramientas de modelado BPMN.



- Una de las características importantes, buscada para el desarrollo de la investigación por parte de una herramienta de modelado BPMN era que tuviera “*Soporte completo BPM*”, en esta las tres herramientas obtuvieron buen puntaje, pero como existen características de igual o más alta importancia fue necesario analizar a las herramientas en conjunto, al igual del comportamiento de sus características.

3.3 ESTUDIO COMPARATIVO DE HERRAMIENTAS WORKFLOW

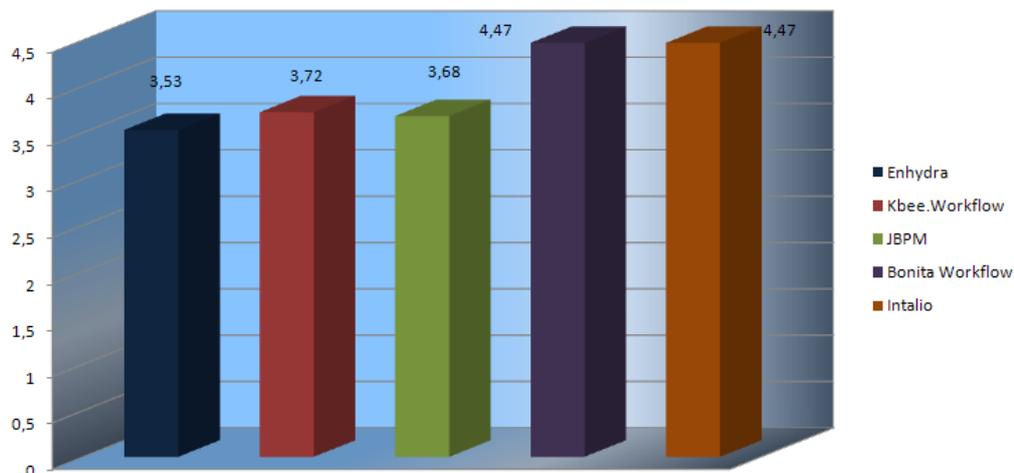
Este apartado se refiere al estudio de herramientas de workflow desarrollado en el anexo A en su subtítulo “Estudio comparativo de herramientas workflow”. Este estudio arrojó los resultados planteados en la figura 11.

La figura 11. Corresponde a los resultados obtenidos por cada una de las herramientas en el estudio. Dichos resultados se basan en una escala de evaluación de cero a cinco. Y cada fragmento corresponde al valor obtenido por cada herramienta.

Es posible plantear las siguientes observaciones a partir de la gráfica anterior y los resultados presentados en la **tabla 55 del anexo A**.

- Intalio y Bonita Workflow fueron las herramientas que obtuvieron el mejor puntaje según las consideraciones planteadas por el estudio en búsqueda de la herramienta workflow que mejor se acomodara a las necesidades del entorno de soporte al proceso de desarrollo software necesite.

Figura 12. Gráfica de resultado consolidado.



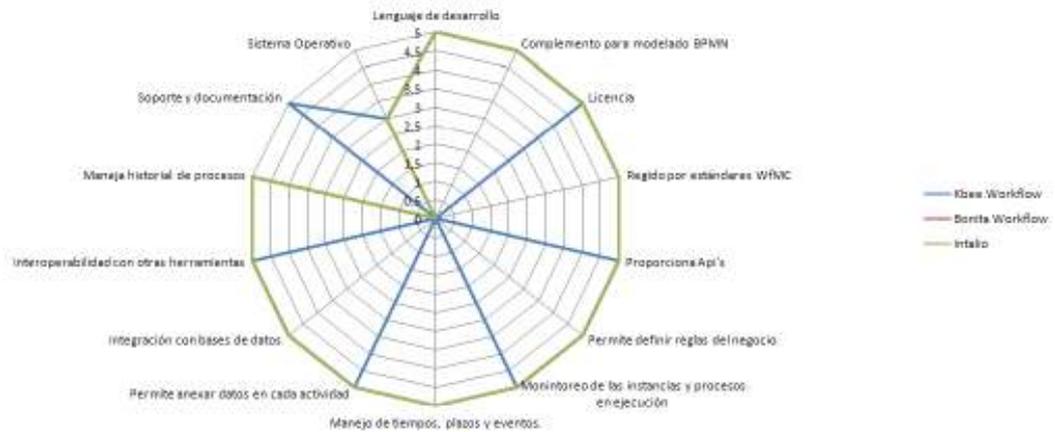
- JBPM, Enhydra y Kbee.Workflow, resultan ser también una buena opción para que hagan parte de la herramienta workflow necesaria para el desarrollo de la investigación, pero no proporcionan de mejor forma las características necesarias.
- Debido a que los valores posibles de los resultados del estudio, se encuentran en un rango de 0 a 5, ninguna de las herramientas logró cumplir con todas las expectativas que solicitaba el desarrollo del entorno con lo referente a las herramientas workflow, por tal motivo no obtuvieron el puntaje máximo posible.

La gráfica anterior no explica muy bien la información obtenida en el estudio, para esto, se plantea la figura 12, en la cual se evidencia de una forma más clara la comparación de las tres herramientas que obtuvieron el mejor puntaje según el estudio.

Esta gráfica proporciona de igual forma información de alto nivel de importancia para las decisiones concernientes a la mejor herramienta workflow que haga parte del entorno de soporte para el desarrollo del proceso software a través de BPM.

- Bonita workflow e Intalio, proporcionan de igual forma las mismas características que la investigación requiere, por tal motivo en la gráfica anterior se ven las características de Bonita Workflow solapadas por las características de Intalio.

Figura 13. Gráfica de resultados de herramientas workflow.



- Kbee.Workflow, posee buenos resultados según los requerimientos solicitados por la investigación pero es claro que no las presenta de forma consistente, sino que en aquellas características que poseen mayor importancia según lo planteado en el estudio, fue donde obtuve los valores más altos.

4. PROPUESTA INTEGRADA PARA EL MODELADO DEL PROCESO Y EL MONITOREO DE PROYECTO SOFTWARE

En este capítulo se presenta la propuesta para la integración de actividades de modelado del proceso software y el monitoreo del proyecto, teniendo como estrategia básica la integración de las herramientas seleccionadas a partir del análisis presentado en el capítulo anterior. Se presenta el esquema general de la solución con los principales roles involucrados, la arquitectura general y su instanciación en dos diferentes tecnologías, además los escenarios de uso del entorno.

4.1 ESQUEMA GENERAL DE LA SOLUCION

El siguiente esquema muestra como se distribuyen los componentes y sus interacciones que hacen parte del entorno de soporte para modelado de procesos y monitoreo de éstos en un proyecto particular, utilizando el enfoque de BPM (ver Figura 14).

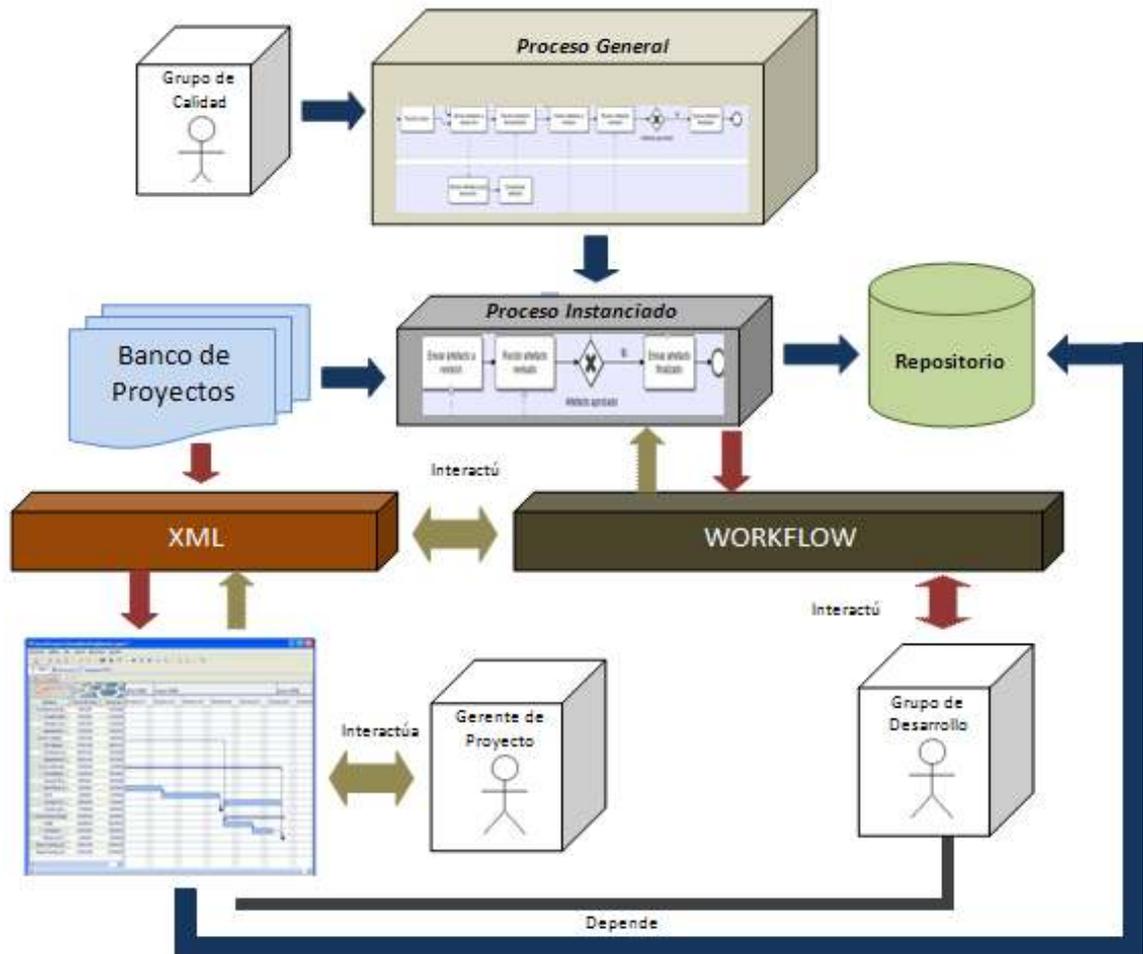
Se muestra en primer lugar como el grupo de calidad es el encargado de mantener y velar los procesos generales de la empresa de acuerdo a las necesidades, políticas y metodologías que usen. Este grupo además debe promover una discusión permanente alrededor de dichos procesos con el fin de optimizarlos, contribuyendo a la mejora continua de la organización.

El Banco de proyectos contiene la descripción de los proyectos que deben ser realizados; el Gerente de Proyecto, de acuerdo al tipo de proyecto, seleccionará y configurará los procesos que sean adecuados.

En el momento en que un proceso se pone en marcha para un proyecto en particular, se dice que el proceso ha sido instanciado, el cual quedará configurado dentro de un motor de workflow., Con ésta instancia deben interactuar los miembros del grupo de desarrollo con el fin de realizar las tareas que les corresponda. El motor de workflow, está en capacidad de alojar las diferentes instancias de los procesos a que haya lugar, éstas son persistidas en un repositorio que también es administrado por el motor. En este punto el entorno permite monitorear las actividades de los diferentes miembros del grupo de desarrollo, y controlar que los procesos se lleven a cabo en forma debida, pero se dificulta para el gerente de proyecto visualizar el progreso de las actividades. Se propone adicionar entonces al entorno una herramienta de gestión de proyectos diseñada para tal fin a través de servicios de interoperabilidad basados en el estándar XML.

Este Esquema general de la solución se complementa con la descripción detallada de los escenarios de uso para cada uno de los actores involucrados, los cuales se presentan en la sección 4.5.

Figura 14. Esquema general de la solución.



4.2 ARQUITECTURA DE LA SOLUCION: Vista de Despliegue.

La arquitectura de la solución refleja la propuesta diseñada para cumplir con el objetivo principal planteado, soportado en una solución tecnológica de libre distribución. La figura xx muestra la vista de despliegue de los diferentes componentes involucrados en la solución.

La solución plantea 5 nodos principales, Servidor WEB, Servidor de Base de Datos, Herramienta Diseño de Proceso, Herramienta Interoperabilidad y Herramienta de Gestión de Proyecto, definidos de la siguiente forma:

- Servidor Web: aloja un paquete de workflow, la definición del proceso y sus posibles instancias.
- Servidor Base de Datos: usado por el workflow para persistir la información correspondiente a procesos e instancias de procesos.
- Herramienta Diseño Proceso: usada en el modelado de procesos bajo notación BPMN.
- Herramienta Interoperabilidad: es la encargada de proporcionar los medios de transformación y comunicación entre el workflow y la Herramienta de Gestión de Proyecto.
- Herramienta de Gestión de Proyecto: es la encargada de administrar los recursos que hacen parte del proyecto en un desarrollo software.

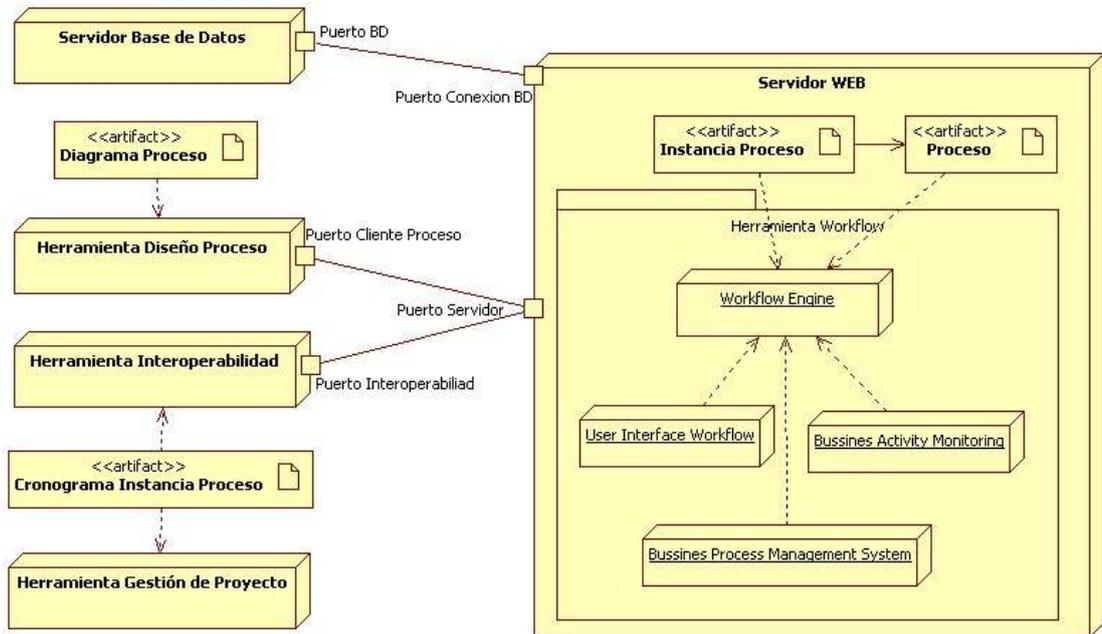
El flujo planteado por la solución se define de la siguiente manera:

Usando la herramienta de modelado BPMN, se diseña un proceso obteniéndose como resultado un artefacto (diagrama de proceso), el cual debe ser publicado dentro del workflow por intermedio de un puerto de conexión o alguna estructura que facilite esa funcionalidad.

Una vez el Workflow Engine aloja el proceso, se encarga de proporcionar las funcionalidades capaces de instanciarlo y manipularlo, a través de componentes como:

- User Interface Workflow: encargada de suministrar los elementos de comunicación entre los procesos y los usuarios validados en el sistema, a través de formularios en dónde el usuario dependiendo de su rol y de la actividad, lo diligencia con el fin de suministrar información que permita la toma de decisiones al motor de workflow y así dar trámite al flujo diseñado en el proceso.

Figura 15. Arquitectura de la solución.



- Bussines Process Management System: permite administrar los procesos y las instancias de estos suministradas por el Workflow Engine.
- Bussines Activity Monitoring: componente que permite monitorear y auditar los procesos y sus instancias.

Para persistir la información relevante de los procesos y sus instancias, como roles, variables, actividades, estados, entre otros, el workflow se comunica con una base de datos diseñada para tal fin, generalmente las herramientas de workflow permite a sus usuarios seleccionar el manejador de base de datos, en otras ocasiones obligan a usar alguna específica o embebida.

La Herramienta de Interoperabilidad por medio de los puertos de acceso correspondientes, consigue la información suficiente para crear el cronograma de la instancia del proceso, el cual será manipulado directamente por la Herramienta de Gestión de Proyecto.

A partir del estudio realizado se puede verificar la existencia de dos corrientes importantes en cuanto al uso de motores de workflow: INTALIO y JBPM. La primera ofrece entornos escalables orientados a empresas o corporaciones, en dónde es posible adicionar funcionalidad al motor a través de aplicativos provistos por la casa matriz creadora del workflow. La segunda es un entorno que ofrece la

posibilidad de acceder directamente al motor a través de APIs, lo que permite su utilización en el desarrollo de sistemas de información.

Tanto el esquema general de solución como la arquitectura de despliegue presentadas en este capítulo pueden ser implementadas utilizando cualquiera de los motores de workflow, tal como puede verse en los dos numerales siguientes.

4.3 SOLUCION PROPUESTA EN INTALIO

A continuación se define como interactúan los recursos provistos por Intalio al ser esquematizados como lo propone la arquitectura general (ver figura 16).

La solución propuesta hace uso de una base de datos MySQL, la cual se comunica por el puerto por defecto 3306, con el servidor que posee el workflow, este servidor utiliza el puerto de comunicación 8080.

La herramienta de modelado BPMN es Intalio Designer, por medio del cual se obtiene el diagrama de proceso en formato BPEL⁵. Este diagrama es consumido por Intalio Workflow a través de Apache ODE⁶, sistema de orquestación de servicios web, encargado de organizar y direccionar los procesos que están escritos en lenguaje BPEL.

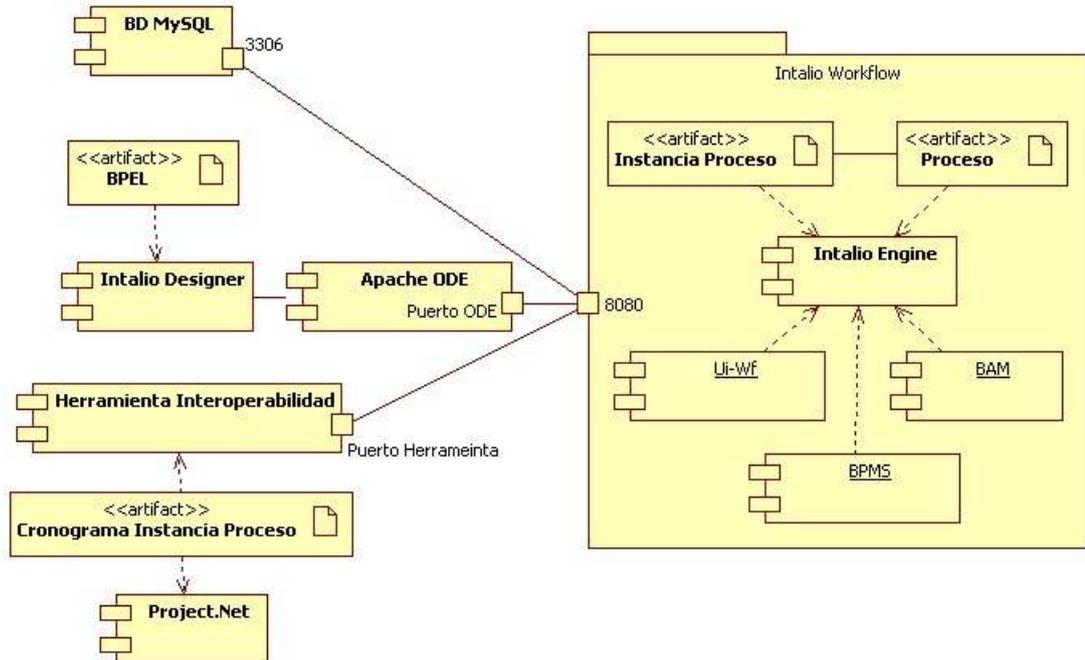
A partir de esto es posible, a través de los componentes del Intalio Engine, realizar diferentes acciones sobre los procesos así:

- Ui-wf, es el encargado de proporcionar los elementos de comunicación entre los procesos y los usuarios que hacen parte del sistema y que tiene permisos suficientes para manipular dichos procesos.
- BPMS-Console, es el encargado de administrar los procesos y las instancias que posean

⁵ BPEL: Bussines Process Ejecution Lenguaje.

⁶ ODE: Orquestation Director Engine

Figura 16. Vista de despliegue solución en Intalio.



- BAM, es el componente delegado para monitorear los procesos y sus instancias.

La Herramienta Interoperabilidad puede establecer comunicación directamente con Intalio Engine, de esta forma puede acceder a la información de los procesos e instancias, definidos dentro del workflow. A partir de dicha información, la Herramienta Interoperabilidad puede generar el Cronograma Instancia Proceso, archivo definido por tecnologías XML que será utilizado por las herramientas de gestión, como Project.Net, Open Project o MS Project, definida como la herramientas de gestión de proyectos que serán utilizadas para la implementación del entorno de soporte para el proceso de desarrollo software a través de BPM.

4.4 SOLUCION PROPUESTA EN JBPM

La arquitectura de la solución implementada sobre JBPM posee una clara diferencia comparada con la propuesta sobre Intalio, ya que JBPM no está definido como un entorno completo, sino que se presenta como un API que proporciona las funcionalidades necesarias para ejecutar el entorno de soporte para el proceso de desarrollo software a través de BPM (ver figura 17).

Como puede observarse, en esta solución no se cuenta con un servidor web para soportar las funcionalidades y componentes del workflow, por eso el flujo de información sobre JBPM se plantea así:

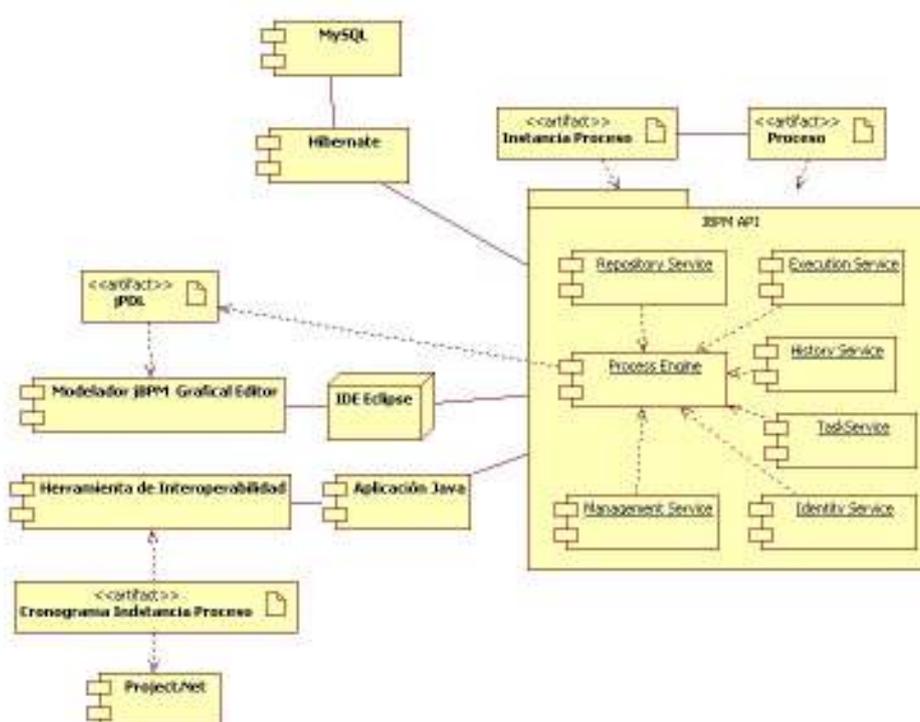
JBPM suministra un framework de persistencia para el manejo de cualquier tipo de base de datos, en este caso MySQL, ya que esta hace parte de los requerimientos técnicos de la investigación. A través del framework de persistencia es posible conectarse con el JBPM API.

Uno de los elementos más representativos de JBPM frente a Intalio, es que oficialmente es soportado principalmente los siguientes componentes:

- jbpm-core, elemento principal del sistema JBPM.
- jbpm-db, paquete de compatibilidad de base de datos basado en hibernate.
- jbpm-bpel, extensión con soporte para BPEL de JBoss JBPM.

Todos estos elementos anteriores se encuentran embebidos dentro de la API de JBPM.

Figura 17. Vista de despliegue solución en JBPM.



Sin embargo, existen otros proyectos de forma anexa, algunos soportados por JBoss, los cuales amplían las posibilidades de uso y mejoran notablemente su usabilidad:

- jbpm-server, un servidor de aplicaciones JBoss pre-configurado. Este componente no es incluido dentro del desarrollo, pues se pretende utilizar JBPM de la forma más natural posible.
- jbpm-designer, plugin para eclipse creado originalmente para JBPM.
- jbpm-gpd o Graphical Process Designer, otro plugin para eclipse perteneciente al paquete JBoss Tools⁷.

La herramienta de modelado utilizada BPMN es jBPM Graphical Editor (jbpm-designer), por medio de esta es posible modelar los procesos de negocio bajo notación BPMN no estándar. Producto de modelar los procesos es posible obtener un proceso escrito sobre lenguaje jPDL⁸, el cual será consumido por el JBPM API, a través del IDE de Eclipse. De esta forma, Eclipse se convierte en la estructura intermediaria entre el JBPM Workflow y JBPM Graphical Editor.

La decisión de usar jbpm-designer sobre jbpm-gpd, se debe a que el primero es el más difundido y utilizado por la comunidad. Además utiliza la versión JBPM 3, sobre el cual se basa este proyecto. La versión 4, posee muchas diferencias frente a la anterior, además de ser relativamente nueva.

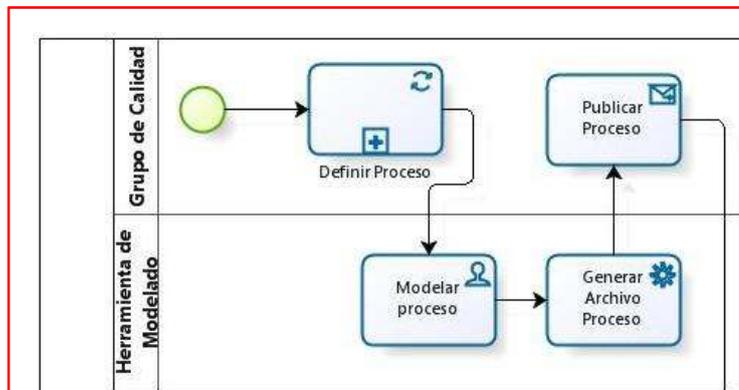
La API de JBPM está conformada por una serie de componentes encargados de manipular los procesos y sus instancias. Estos componentes son las librerías Repository Service, Execution Service, Management Service, History Service, Task Service e Identity Service, a partir de las cuales es posible administrar los procesos, monitorear y definir los elementos de comunicación entre estos y los usuarios que posean los permisos suficientes para manipularlos.

Ya que sobre JBPM no se utiliza un servidor, es posible crear la comunicación directamente entre el API de JBPM y la Herramienta Interoperabilidad construida para obtener la información pertinente de los procesos y sus instancias, con el fin de construir el Cronograma Instancia Proceso, este último será usado por las herramientas de gestión de proyectos, para definir el cronograma de actividades dentro del proceso de desarrollo software.

⁷ JBoss Tools: plugin para eclipse con soporte para JBPM 4.

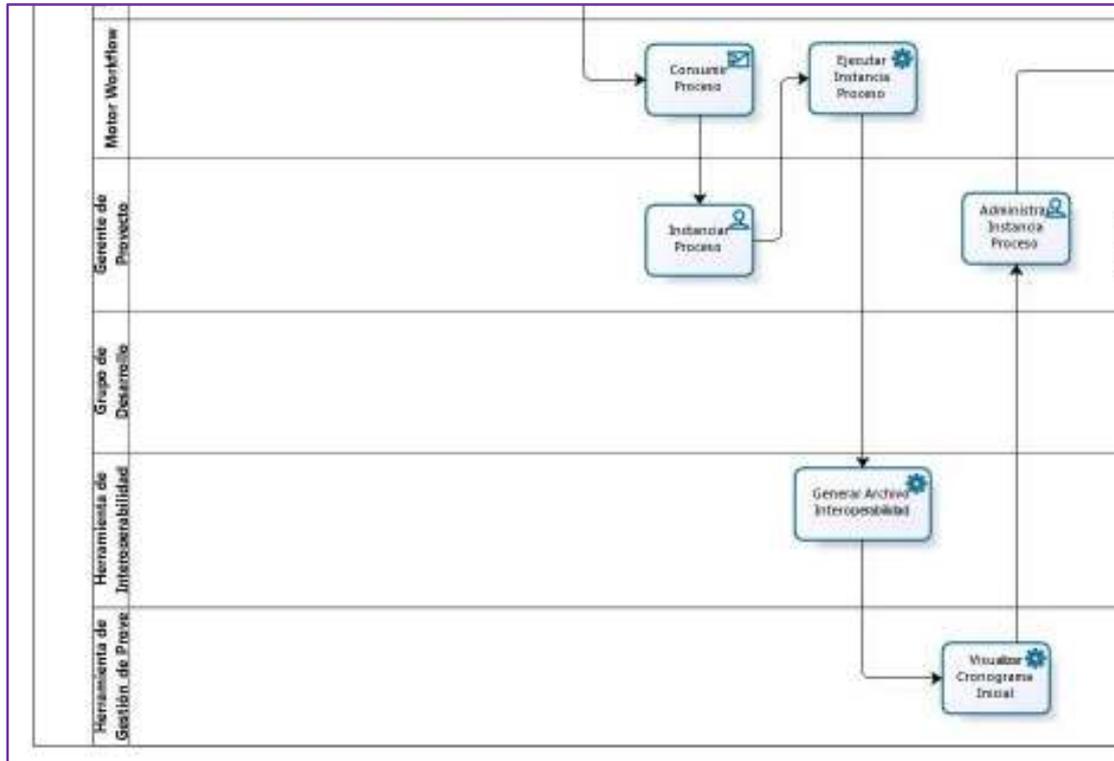
⁸ jPDL: Java Process Diagram Language.

Figura 19. Primera parte flujo del proceso.



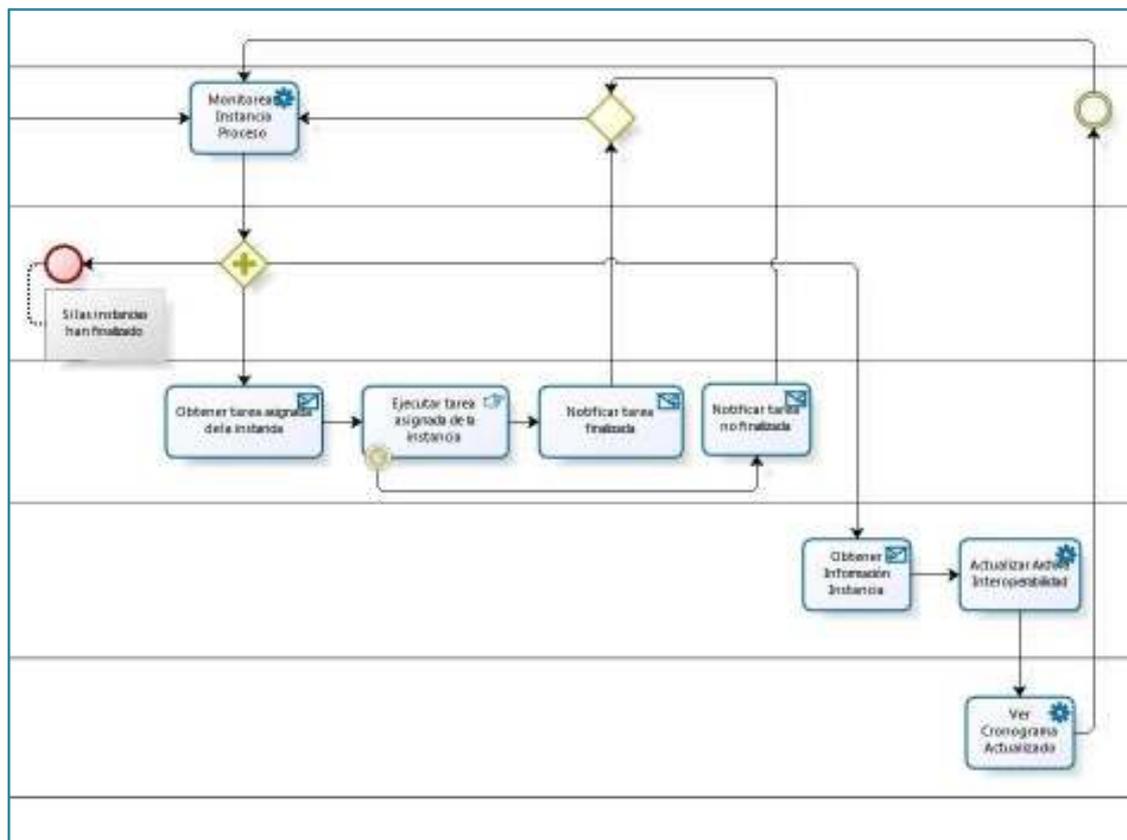
- Una vez, definido todo lo concerniente con el proceso, se procede a modelarlo sobre la herramienta de modelado BPMN, acá es posible obtener un diagrama de procesos. Este diagrama puede tener diferentes tipos de extensiones dependiendo de la herramienta de modelado en la cual se este trabajando.
- El archivo de proceso es posible desplegarlo en el servidor, donde esta alojado el *Motor Workflow*, esta tarea denominada como *Publicar Proceso*, también será realizada por el *Grupo de Calidad* (ver figura 20).
- Una vez el proceso reside en el *Motor Workflow*, se define la tarea *Consumir Proceso*, que permite crear las instancias de este. De esta tarea se encarga el *Gerente de Proyecto*, asignando los responsables de cada una de las actividades que son necesarias para la instancia.
- A partir de ejecutar la instancia del proceso, por medio de la *Herramienta de Interoperabilidad* se genera el *archivo de interoperabilidad*, el que será interpretado por la *Herramienta de Gestión de Proyecto*, para visualizar de este forma el cronograma inicial del proyecto.
- En este momento se inicia la administración de la instancia del proceso por parte del Gerente de proyecto.

Figura 20. Segunda parte flujo del proceso.



- La tarea de *Monitorear Instancias Proceso*, es la encargada de llevar un control de los estados de las instancias y del flujo de información de cada una de estas.
- A partir del monitoreo de las instancias se provee una interfaz con el fin de que los encargados de las actividades reporten el estado de la mismas. Esta tarea posee un tiempo determinado para ser realizada, y posee dos estados: *finalizada* y *no finalizada*. Dicho estado es reportado al proceso de monitoreo de las instancias, a partir del cual se define el fin de las instancias del proceso o un ciclo, terminado sólo hasta cuando dichas tareas sean culminadas en su totalidad.

Figura 21. Tercera parte flujo del proceso.



- Dentro de este flujo, si las instancias no han sido finalizadas, se puede obtener por medio de la *Herramienta de Interoperabilidad* la información de las instancias del proceso, a partir del *Motor Workflow*. Ya que el *Archivo de Interoperabilidad* ha sido creado, solo basta con ser actualizado con la nueva información concerniente al procedimiento realizado sobre las instancias. Una vez definido lo anterior, es permitido visualizar el cronograma creado anteriormente pero actualizado.
- Debido a que el monitoreo de las instancias se realiza de forma cíclica, a partir del cronograma visualizado en la herramienta de gestión de proyectos, es necesario definir una relación directa con la actividad de monitoreo de las instancias. En este punto se garantiza que el monitoreo sea realizado constantemente.

5 CASO DE ESTUDIO

El caso de estudio que se ha trabajado con el propósito de ilustrar la utilización del entorno se ha orientado hacia un proceso de revisión de pares. El objetivo de este proceso es el de someter ciertas productos resultantes de actividades para ser evaluados, criticados y mejorados por un número de actores involucrados con el propósito de garantizar la calidad del producto, independiente de la forma en la cual dicho producto es desarrollado. Para propósitos de este trabajo se definió la revisión de pares para la creación y desarrollo de artefactos en la etapa de análisis.

Una de las características fundamentales de esta práctica es que la evaluación de un determinado artefacto es realizado por un actor que tenga las mismas competencias de la persona que generó el artefacto, es decir si la tarea de definir los requisitos del sistema y es realizada por un Ingeniero de Requisitos, la revisión debe ser realizada por un Ingeniero de Requisitos con lo cual se garantiza una crítica objetiva sobre el artefacto en específico.

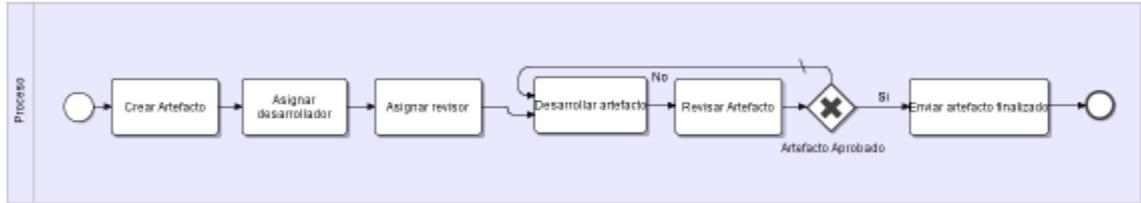
Ahora se procede a desarrollar la solución sobre Intalio|Works y JBPM.

5.1 DESARROLLO DEL ENTORNO EN INTALIO|WORKS

Como parte del entorno de soporte para el proceso de desarrollo software a través de BPM, es necesario realizar el modelado sobre una herramienta de modelado BPMN, en este caso Intalio Designer, e implementar el modelo resultante sobre Intalio Server.

El primer paso se definir el proceso principal en torno al cual gira la funcionalidad principal definida por el caso de estudio. En el proceso principal se definen una serie de tareas, las cuales están relacionadas con un pool denominado *Proceso*, dichas actividades se ven reflejadas en la figura xx.

Figura 22. Proceso elaborado en Intalio Designer.



El proceso anterior muestra el flujo de información general, es decir las actividades libres de roles o tecnologías de entrada y salida. Esto permite realizar el proceso tal como es y no contaminar el proceso con factores externos a este.

Por razones de espacio y teniendo en cuenta los múltiples detalles en la notación que provee la herramienta y que se utilizan en este caso de estudio se decidió describir el detalle de su implementación en el anexo B.

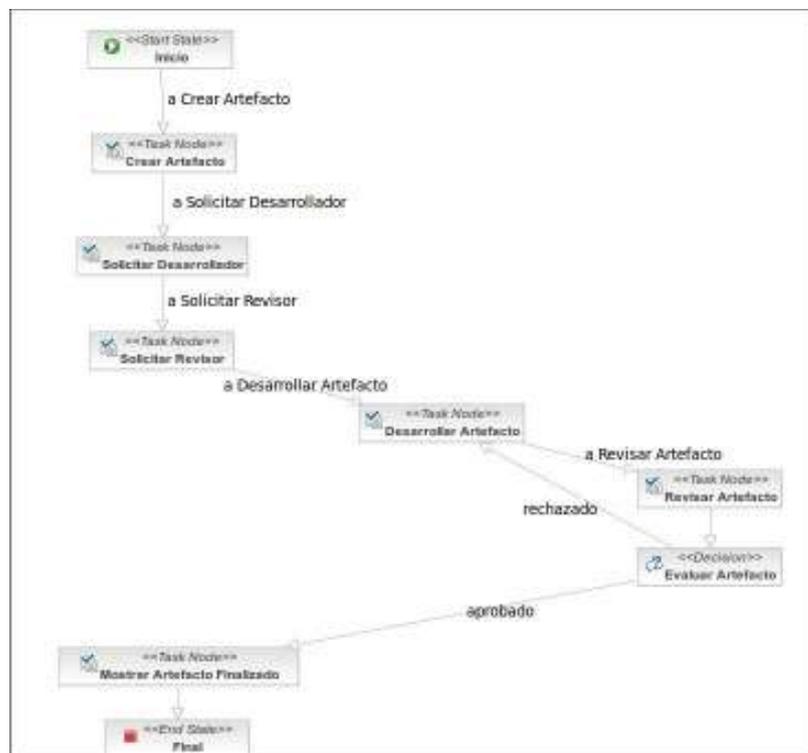
5.2 DESARROLLO DEL ENTORNO EN JBPM

En JBPM a diferencia de cómo lo define el estándar BPMN y de cómo lo hace Intalio Designer, utiliza un elemento denominado transición (*Transition*) capaz de comunicar los modos (como *Start State*, *Task Node*, *Decision* y *End State*) independiente del swimlane⁹ en el que se ejecute. La figura xy muestra el proceso general modelado en JBPM.

De forma similar, por razones de espacio, el detalle de implementación del caso de estudio en JBPM se presenta en al anexo C.

⁹ Swimlane: Son diagramas en los cuales se especifica un flujo de proceso. http://www.conceptdraw.com/en/products/cd5/ap_swim_lane.php

Figura 23. Modelado general del proceso sobre JBPM.



Para ver los detalles del proceso de construcción del modelo y su respectiva implementación, ver anexo C.

5.3 DESARROLLO DE LOS SERVICIOS DE INTEROPERABILIDAD

En el desarrollo del entorno propuesto dentro de la investigación se recurre a una herramienta de interoperabilidad basada en tecnologías Java, por medio de la cual es posible obtener la información pertinente de los procesos y las instancias de los mismos. En este punto se genera un archivo XML que refleje la información obtenida a partir de la herramienta workflow y que sirva para generar el cronograma de las actividades del proceso de desarrollo software dentro de las herramientas de gestión de proyectos.

Dentro de la investigación se entró a analizar, a través de pruebas de concepto, la viabilidad de integración utilizando los dos motores de workflow sobre los cuales se mostró el caso de estudio.

5.3.1 Desarrollo de la herramienta de interoperabilidad sobre Intalio|Works

Una vez se ha construido e implementado el modelo de Revisión de Pares sobre Intalio|Works (Designer y Server), se procede a extraer la información de los procesos y sus respectivas instancias. Para ello son necesarios dos componentes que permitirán obtener el archivo XML a utilizar por las herramientas de gestión:

- Sistema para extraer la información de los procesos y las instancias provenientes de Intalio Server
- Herramienta de generación de XML de las herramientas de gestión. Para ver los detalles, véase anexo D.

Para acceder a dicha información se proponen diferentes opciones que permitirán obtener la información mencionada. Las alternativas diseñadas son:

5.3.2 Obtención de información a través de una API.

Intalio ofrece sus productos Intalio|Works sobre licencias de código abierto, lo que hace pensar que es posible acceder a la información a través de código Java, sin embargo esto no es así. Intalio ofrece sus productos gratis, no libres, lo que hace que el código que como funcionan sea inaccesible.

Sin embargo, no se deseaba acceder al código de cómo está desarrollador los productos de Intalio, sino a una API para el acceso a los procesos e instancias utilizadas. Esto tampoco fue posible. Dentro de la documentación ofrecida por Intalio, se encuentran todo tipo de manuales para utilizar sus herramientas, pero no para llamar procesos ni instancias desde elementos externos a las herramientas ofrecidas por este.

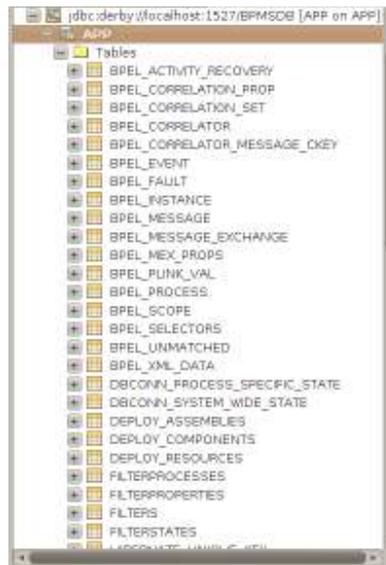
5.3.3 Obtención de información a través de la base de datos.

Intalio Server es la herramienta capaz de instanciar los procesos y hacer uso de los mismos, lo que significa que de alguna forma debe almacenarlos. Gracias a la documentación ofrecida sobre las herramientas, es posible utilizar bases de datos, pero esta se limita a uso dentro de los procesos, mas no de los mismos.

Buscando dentro del código de Intalio Server (debido a que utiliza un servidor basado en Apache Tomcat 5), fue posible encontrar la base datos, la cual es

Apache Derby¹⁰. Gracias al archivo de configuración (*database/Derby/ tomcat-5-resources.properties*) fue posible acceder a la base de datos. Esto permitió obtener la información de los procesos ejecutados en el servidor, tal como se ilustra en la figura 24.

Figura 24. Base de datos Derby de Intalio Server.



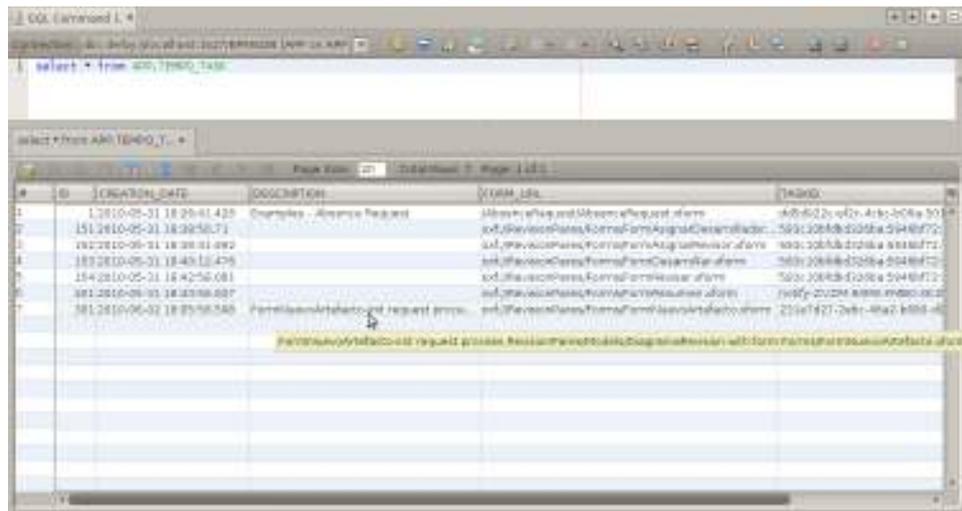
Pese a ser posible el acceso a la información de la base de datos de Intalio Server, su uso se hace dispendioso por varios motivos:

- Dentro de la documentación disponible, no se encuentra el modelo relacional que soporta la base de datos, por lo que para obtener la información sería necesario realizar ingeniería inversa, e intentar adivinar la estructura de la misma para poder obtener la información de forma adecuada.
- Existen muchas tablas de las cuales se desconoce el significado y campos dentro de las mismas que dificulta el proceso de entendimiento de la información contenida, como se observa en la figura 25.
- El hecho de buscar la información en el servidor para acceder a la base de datos no es algo correcto. En entornos reales, las corporaciones no están dispuestas a permitir el acceso a la configuración de sus servidores, ni tampoco a las bases de datos que soportan sus procesos.
- Si se accede directamente a la base de datos, no se estaría interoperando con la herramienta de procesos.

¹⁰ Apache Derby: Motor de base de datos para Java. <http://db.apache.org/derby/>

Por estas razones, la solución a través de la base de datos se hace inviable.

Figura 25. Consulta a la tabla APP.TEMPO_TASK de la base de datos de Intalio Server.



The screenshot shows a SQL query result in a database client. The query is 'select * from APP.TEMPO_TASK'. The result is a table with the following columns: ID, CREATION_DATE, DESCRIPTION, FORM_URL, and TASKID. The table contains several rows of data, including a row with ID 151 and DESCRIPTION 'Completar - Absence Request'.

ID	CREATION_DATE	DESCRIPTION	FORM_URL	TASKID
1	12010-05-31 18:20:41.425			
151	2010-05-31 18:20:50.771	Completar - Absence Request	http://www.intalio.com/Intalio/Forms/RequestForm.aspx	151
152	2010-05-31 18:20:51.082			
153	2010-05-31 18:45:18.425			
154	2010-05-31 18:42:56.081			
155	2010-05-31 18:20:58.087			
156	2010-05-31 18:20:58.540			

5.3.4 Obtención de información a través de Apache ODE.

Apache ODE¹¹ es un sistema para la orquestación de servicios web. Este ofrece la posibilidad de acceder a los procesos a través de servicios web estándar desde cualquier lenguaje que los soporte. Sin embargo esta solución también se hace inviable por:

- Apache ODE no hace parte de Intalio|Works. Intalio Server utiliza esta tecnología, pero no está diseñada para ser utilizada desde afuera a las herramientas provistas por Intalio.
- Apache ODE es una herramienta muy grande y compleja, además de ser nueva. Esto significa que se demoraría el desarrollo del proyecto mas en aprender cómo funciona Apache ODE que en el mismo desarrollo. De hecho, es tan nueva la herramienta que no aparece aun en Wikipedia en español.

¹¹ Apache ODE: Motor de orquestación de servicios web escritos sobre WS-BPEL.

- Al igual que con la solución sobre la base de datos, no se estaría interoperando la herramienta Intalio, sino sería a través de elementos externos a esta.

5.3.5 Obtención de información a través de modelo del proceso.

Esto significa que en el momento de desarrollar el proceso, se crearía un nuevo pool, en el cual se obtendría la información del proceso. Esta solución permitiría obtener de primera mano el estado del proceso, sin embargo acarrearía las siguientes consecuencias:

- Afectaría drásticamente el modelado. Esto debido a que a medida que se diseña el modelo, se deberían agregar tareas que permitan obtener la información de forma externa.
- Los modelos crecerían al doble de su tamaño. Para un modelo como el planteado que cuenta con un número de tareas no superior a diez (10), el número de tareas finales podría superar fácilmente las veinte (20).
- El modelo se contaminaría con elementos externos a él. Todos los modelos planteados sobre BPM tienen un propósito específico, como la revisión de pares, sin embargo sería necesario adicionar elementos diferentes a su propósito original, lo que cambia el rumbo del modelado sobre estas herramientas.
- Es posible olvidar guardar la información. Como para desarrollar esta solución sería necesario agregar tareas que obtengan los datos, implica que por diferentes motivos, no se adicione una tarea para extraer la información, lo que implica que la obtención de la información no sería verídica del todo.

5.3.6 Obtención de información a través de Intalio BAM.

Intalio BAM¹² es una herramienta para realizar monitoreo sobre los procesos e instancias que corren sobre las herramientas soportadas por Intalio. Esto permitiría adquirir la información necesaria para la herramienta de generación XML y así completar la solución planteada.

Pese a ser una solución viable del sistema, en el momento de su utilización se encontraron los siguientes inconvenientes:

¹² Intalio BAM: <http://www.intalioworks.com/products/bpm/enterprise-edition/bam/>

- Intalio BAM no se encuentra dentro de la versión Intalio|Works Community, sino dentro la versión comercial. Esto implica adquirir la licencia del paquete Enterprise que tiene un costo muy elevado, o al menos en un entorno investigativo. A esto se le suma que dejaría de ser una herramienta que integraría el entorno de soporte del proyecto debido a que no es accesible por todas las personas.
- Poca información se conoce sobre la herramienta. Por su carácter privativo, es escasa la información de esta, por lo que no es posible asegurar su uso para los fines buscados.
- Las imágenes ofrecidas por Intalio, al igual que las características, sugieren que es una herramienta para obtener la información de forma muy similar a un reporteador¹³ como se ilustra en la figura 26, es decir, graficas de barras, tablas y similares. Para los fines del proyecto, se requiere que esta información pueda ser extraída e ingresada a otros sistemas, por lo que no cumpliría las metas planteadas.

Una vez se ha finalizado el proceso de exploración sobre Intalio|Works, y se han agotado todas las posibilidades de integración, arrojando una conclusión un tanto inesperada, ya que no hay muchas posibilidades de integrar la vista del gerente del proyecto para monitoreo del proyecto desde una perspectiva del cronograma inicialmente definido.

La figura 26, es un ejemplo de cómo Intalio BAM muestra la información acerca de los procesos y de las instancias de los mismos.

¹³ Reporteador: Sistema generador de reportes.

Figura 26. Intalio BAM.
Fuente: http://2.bp.blogspot.com/_TFEOqgfodxU/SkQTesI9Bel/AAAAAAAAAI8/Zqa96axE29w/s320/bam.intalio.jpg



5.3.7 Desarrollo de la herramienta de interoperabilidad sobre JBPM.

Para construir una solución de interoperabilidad basada en JBPM, se utiliza una metodología similar a la utilizada sobre Intalio|Works, basados en el modelo y su respectiva implementación sobre la herramienta previamente realizados. Para ver detalles de la construcción del modelo sobre JBPM, véase anexo C.

5.3.8 Obtención de información a través de una API.

JBPM es una herramienta de workflow diseñada para ser integrada en otras aplicaciones en entornos de desarrollo. Esto significa que se encuentra ligada con un lenguaje de programación que para el caso es Java. Para que lo anterior ocurra, JBPM ofrece al público una API con posibilidad de acceso al código y a la documentación del mismo para poder desarrollar soluciones basadas en esta tecnología. Es importante aclarar que el hecho que JBPM ofrezca la API para los desarrolladores no significa que sea exclusivamente para estos. De forma similar a Intalio|Works, JBPM ofrece a sus usuarios diseñadores y servidor.

Ahora bien, una vez se ha modelado el caso de estudio Revisión de pares y se ha implementado, es necesario obtener la información del proceso. Esto es relativamente fácil en JBPM, debido a que se tiene acceso al código y se controla totalmente la ejecución de los procesos.

Antes de diseñar el sistema de interoperabilidad, vale la pena recordar los siguientes puntos importantes:

- En caso de estudio solucionado sobre JBPM (ver anexo C), se relaciona cada tarea a realizar con una clase. A esta acción se le denomina *asignar controlador a una tarea*.
- Las clases que se definen como controlador de tarea deben implementar una interfaz llamada *TaskControllerHandler*. Esta interfaz obliga a que la clase tenga como mínimo dos funciones: *initializeTaskVariables* y *submitTaskVariables* que son utilizadas para definir el funcionamiento de la tarea.
- Las tareas ejecutadas cuentan con información sobre el proceso sobre el cual se ejecuta, al usuario o usuarios a los que le pertenece y los tiempos de inicio y fin de cada una.

Con base a la información anterior, se definen unas estrategias para obtener la información requerida:

- La primera se trata de agregar código dentro de cada función ejecutada por una tarea. Esto permitiría obtener de primera mano la información requerida para los fines del proyecto. Para realizarlo sería necesario definir un código estándar que sería adicionado dentro de cada función (generalmente al final), además de diseñar una estructura que permita ir almacenando dicha información.

Sin embargo, la solución presenta una serie de inconvenientes: Para que sea efectiva la captura de información, tal código debe estar dentro de todas y cada una de las clases que sean controladores de una tarea. Además a ello, se estaría modificando el objetivo de la tarea, pues estaría realizando acciones diferentes para las que se ha diseñado.

- Otra forma factible de obtener la información es a través de un objeto tipo *ProcessInstance*¹⁴, el cual es el que ejecuta todo el proceso, por ende tendrá toda la información del mismo. Esta solución (más factible que la primera) permite obtener toda la solución en un mismo punto, sin embargo también sería necesario modificar el código de la aplicación ya realizada. Esto se debe a que el objeto en mención es el que permite enviar y recibir información a las tareas.

¹⁴ *ProcessInstance*: <http://docs.jboss.com/jbpm/v3/javadoc/org/jbpm/graph/exe/ProcessInstance.html>

- Una tercera posibilidad es a través de la base de datos. A diferencia de Intalio Server, JBPM ofrece al público el modelo de la base de datos, el código del mismo y permite su control a través del hibernate¹⁵. Esto permite obtener la información de forma efectiva y verídica, no obstante no es la forma correcta de obtener la información.
- Por último, se pensó en un sistema que pudiese obtener la información de las tareas ejecutadas sin necesidad de modificar código alguno dentro del proyecto ya existente, por lo cual surgió la solución basada en Aspectos. Los aspectos (más conocido con programación orientada a aspectos) permite, entre muchas otras cosas, adicionar código antes y/o después de la ejecución de las funciones, y obtener los parámetros y retornos de las mismas, permitiendo obtener de primera mano la información de la ejecución de los procesos.

Tomando en cuenta las posibilidades anteriores, se define que la posibilidad basada en aspectos es la más adecuada para los propósitos del proyecto.

5.3.9 Desarrollo de la herramienta de interoperabilidad sobre JBPM basada en Aspectos.

La programación orientada a aspectos permite a los desarrolladores escribir, ver, y editar un aspecto diseminado por todo el sistema como una entidad por separado, de una manera inteligente, eficiente e intuitiva. Esto significa que puede definirse una funcionalidad de forma clara, pero que se encuentra repartida por todo el código de un proyecto. Como ejemplo trabajado por Fernando Asteasuain y Bernardo Contreras en su análisis del paradigma orientado a aspectos [5].

Para el caso puntual, la funcionalidad diseminada es la obtención de información de las tareas, de tal forma que se puede crear un aspecto que contendrá todo lo necesario para recopilar dicha información, independientemente que se encuentre repartido por todo el código.

Para realizar esta tarea, a través de un aspecto denominado *InformacionTareas*, se agregará una sección de código que se ejecutara al final de cada función *submitTaskVariables*, la cual es la confirma la finalización de una tarea en una clase controlador de tareas.

Para la construcción de esta solución se utilizo AJDT¹⁶ versión 2.0.2 con librerías AspectJ¹⁷ versión 1.6.7 corriendo sobre Eclipse Galileo RS2¹⁸.

¹⁵ Hibernate: Framework de persistencia para Java soportado por JBoss. <http://www.hibernate.org>

¹⁶ AJDT: AspectJ Development Tools es un plugin para eclipse.

Lo primero que se hace es una clase que generará el archivo XML y que almacenara la información de las tareas. Esta clase se encuentra apoyada en el proyecto de generación de archivos XML para herramientas de gestión de proyectos explicado en el anexo D.

¹⁷ AspectJ: Lenguaje de programación orientado a aspectos construido como extensión a Java.
<http://eclipse.org/aspectj/>

¹⁸ Eclipse Galileo RS2: Entorno de desarrollo integrado.
<http://www.eclipse.org/downloads/packages/release/galileo/sr2>

Figura 27. Clase GeneradorXMLProject.

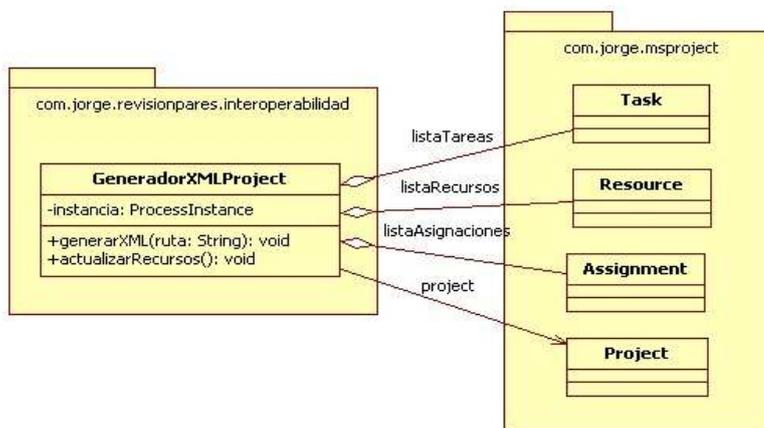
```
GeneradorXMLProject.java 22
package com.jorge.revisionpares.interoperabilidad;
import java.util.ArrayList;

/**
 * Sistema generador de archivos XML Project
 * @author Jorge Enrique Dtalora Luna
 */
public class GeneradorXMLProject {
    private ProcessInstance instancia;
    private List<Task> listaTareas;
    private List<Resource> listaRecursos;
    private List<Assignment> listaAsignaciones;
    private Project project;

    public GeneradorXMLProject() {
        listaTareas = new ArrayList<Task>();
        listaRecursos = new ArrayList<Resource>();
        listaAsignaciones = new ArrayList<Assignment>();
        project = new Project();
    }
}
```

Esta clase (figura 27) tendrá la responsabilidad de generar el archivo XML a partir de la información ingresada dentro de sus atributos listaTareas, listaRecursos y ListaAsignaciones. Los detalles del funcionamiento se detallan en la figura 28.

Figura 28. Diagrama de clases paquete com.jorge.revisionpares.interoperabilidad.



Ahora se crea el aspecto capaz de obtener la información de las tareas. El aspecto se llama Información Tareas (figura 29) y deberá contener la estructura de obtención de información.

Figura 29. Aspecto Información Tareas.

```
InformacionTareas.aj
package com.jorge.revisionpares.interoperabilidad.aspectos;

import org.jbpm.graph.exe.Token;

/**
 * Aspecto para la obtención de información de los procesos ejecutados.<br/>
 * La información recopilada se almacena en un GeneradorXMLProject
 * @author Jorge Enrique Otalora Luna
 * @version 0.1
 */
public aspect InformacionTareas {

    public static GeneradorXMLProject generador = new GeneradorXMLProject();
}
```

Para diseñar tal estructura, lo primero que se hace necesario es crear un pointcut. Un pointcut o punto de corte permite definir en qué lugares se aplicará el aspecto.

A través de este punto de corte (figura 30) se define que se aplicara a cualquier tarea de carácter público, donde el nombre no tiene importancia y el nombre de la función sea *submitTaskVariables*.

Figura 30. Pointcut alFinalizarTareas del aspecto InformacionTareas.

```
/**
 * Punto: En cualquier funcion llamada submitTaskVariables sin tener en cuenta sus parametros
 */
pointcut alFinalizarTareas() :
    execution( public * *.submitTaskVariables(..));
```

Una vez definido el punto de corte, se procede a crear el aspecto que se aplicará después de cada punto de corte alFinalizarTarea (Figura 31).

A través de las cualidades ofrecidas por aspectos, es posible obtener los parámetros que entran a la función, como el TaskInstance¹⁹ el cual es la instancia de la tarea.

¹⁹ TaskInstance: <http://docs.jboss.com/jbpm/v3/javadoc/org/jbpm/taskmgmt/exe/TaskInstance.html>

Figura 31. Aspecto después del punto de corte al FinalizarTarea.

```

/**
 * Después de ejecutarse cualquier punto alFinalizarTareas()
 */
after() : alFinalizarTareas() {
    TaskInstance instancia = (TaskInstance) thisJoinPoint.getArgs()[0];
    //ContextInstance contexto = (ContextInstance) thisJoinPoint.getArgs()[1];
    Token token = (Token) thisJoinPoint.getArgs()[2];
    String nombre = instancia.getName();
    System.out.println( "[ INFORMACION ] Nombre Tarea:" + nombre );
    System.out.println( "[ INFORMACION ] Inicio Tarea:" + token.getStart() );
    System.out.println( "[ INFORMACION ] Final Tarea:" + instancia.getEnd() );

    Task tarea = new Task(generator.getListaTareas().size()+1, nombre, token.getStart(), instancia.getEnd(), 0, 0);
    generator.getListaTareas().add( tarea );

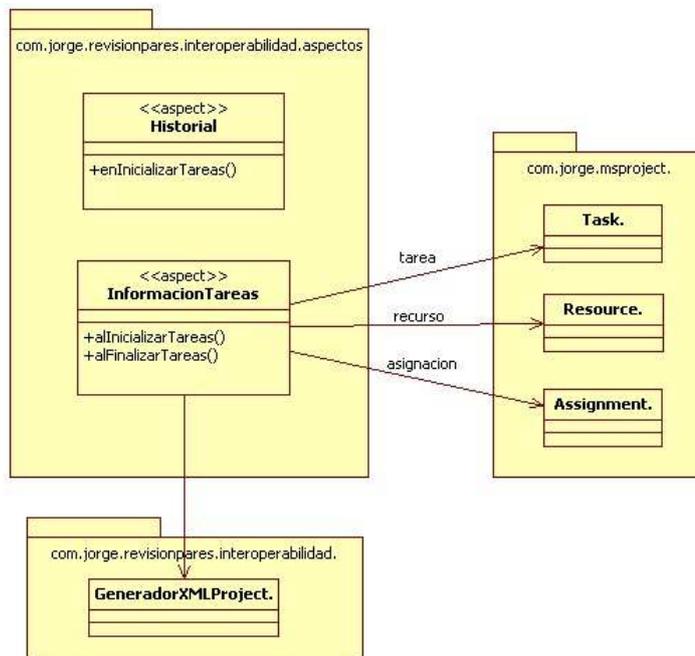
    Resource recurso = new Resource( generator.getListaRecursos().size()+1, instancia.getName(), 1);
    generator.getListaRecursos().add(recurso);

    Assignment asignacion = new Assignment(generator.getListaAsignaciones().size()+1, tarea, recurso);
    generator.getListaAsignaciones().add(asignacion);
}

```

Ahora es posible obtener la información de las tareas ejecutadas en un proceso BPM, independiente del objetivo del mismo, apoyados en la programación orientada a aspectos. Los detalles del funcionamiento de los aspectos para la obtención de información se detallan en la figura 32.

Figura 32. Diagrama de clases paquete.



Ahora solo basta ejecutar el proceso y definir en qué lugar se ha almacenado la información (figura 33).

Figura 33. Crear XML a partir de la información obtenida en aspectos en JBPM.

```
InformacionTareas.generador.setInstancia(instancia);  
InformacionTareas.generador.actualizarRecursos();  
  
InformacionTareas.generador.generarXML( "data/cronograma.xml" );
```

El resultado, un XML que contiene la información de las tareas ejecutadas en formato XML Project. Para obtener más información sobre la generación de XML para herramientas de gestión de proyectos, véase anexo D. Figura 34. XML Generado por la herramienta de interoperabilidad bajo JBPM.

Figura 34. Esquema de archivo XML.

Ahora lo único que falta, es probar su correcto funcionamiento sobre herramientas de gestión de proyectos, como OpenProj (Figura 35), Microsoft Project (Figura 35) y Project.NET (Figura 14).

Figura 35. Cronograma generado desde JBPM en OpenProj.

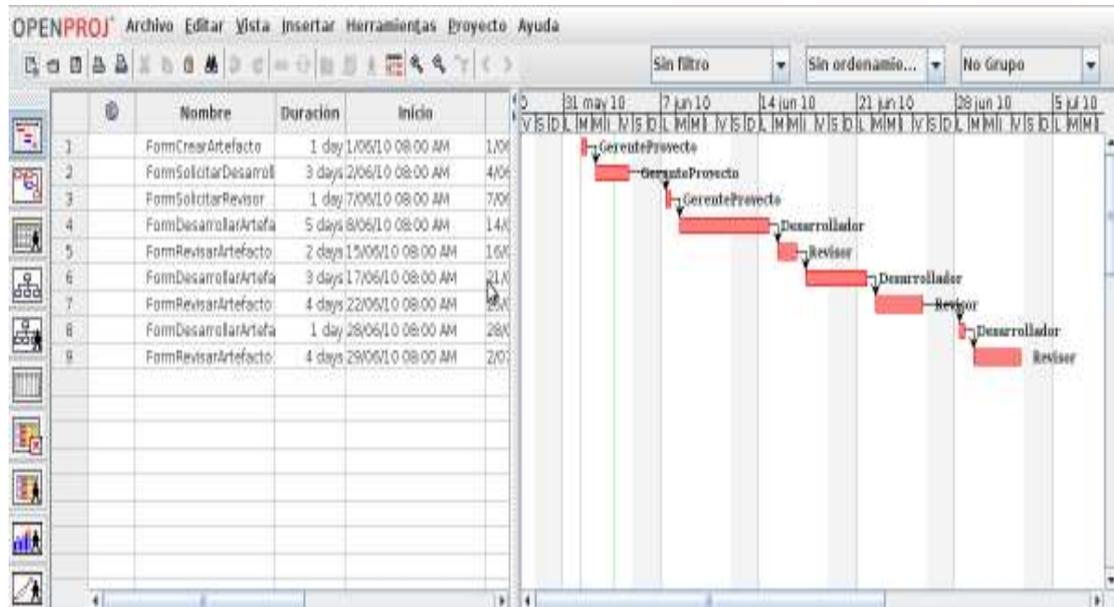
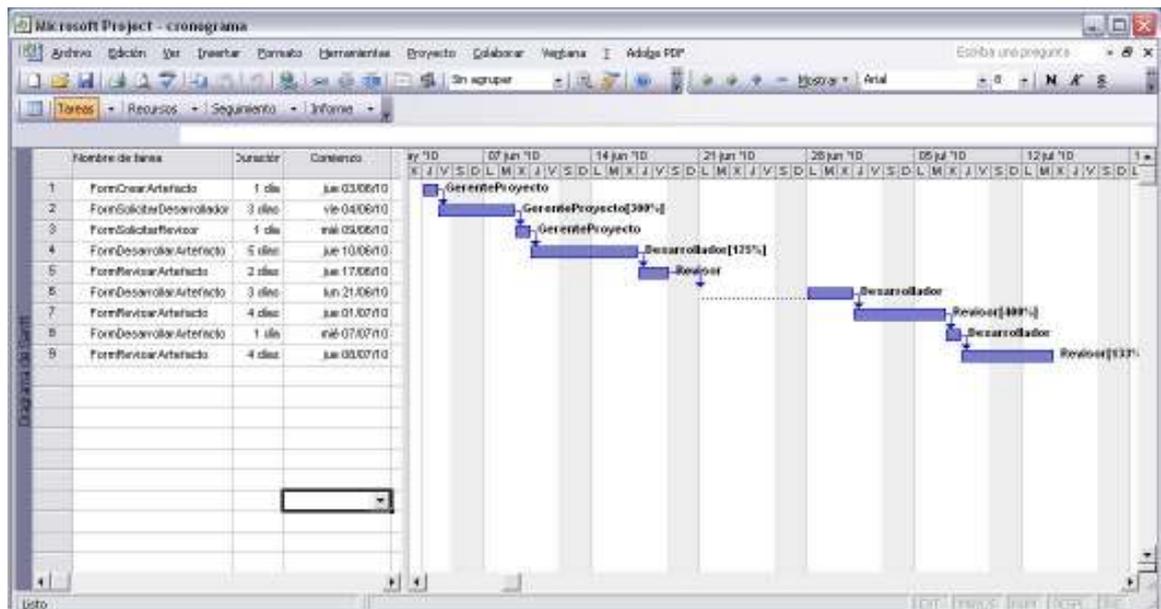


Figura 36. Cronograma generado desde JBPM en Microsoft Project 2003.



6 CONCLUSIONES Y TRABAJOS FUTUROS

El presente trabajo ha propuesto un entorno de apoyo a la ejecución de un proyecto de desarrollo de software utilizando el enfoque BPM sobre tecnologías de dominio público.

Si bien buena parte del esfuerzo de trabajo fue la exploración y pruebas de concepto sobre las tecnologías disponibles, el trabajo quiere dejar claridad que buena parte del éxito de un proyecto de mejora de procesos radica en la cultura y disciplina de la organización para iniciar y mantener una mejora basado en procesos. La adopción de un entorno de soporte integrado implica un nuevo modelo de trabajo para la organización en los que se debe realizar la redefinición de algunas actividades y hacer una definición clara de los roles involucrados.

Previa a la definición del entorno de soporte para el proceso de desarrollo de software, fue vital realizar un estudio de las herramientas que mejor se acoplaran a los requerimientos exigidos por la investigación. Se partió de un estudio de las herramientas de gestión de proyectos que mejor se adaptaran al desarrollo del entorno. A partir de estos estudios se logró obtener una metodología de desarrollo de los estudios comparativos lo suficientemente genérica, para ser implementada en cualquier estudio comparativo de cualquier herramienta, ya que la metodología diseñada define pasos y análisis independientes al tipo y la cantidad de herramientas que se estén comparando.

Este estudio comparativo, presentado en el anexo A, representa un material valioso para aquellas personas u organizaciones que deseen apropiarse una de estas herramientas o un conjunto de ellas. Si bien este estudio se apoyó en los materiales y manuales técnicos divulgados por cada herramienta, se hizo un considerable esfuerzo para realizar las pruebas de concepto que fueron posibles.

El entorno de soporte para el modelado y monitoreo de procesos, propuesto en este trabajo se logró gracias a la unión de las herramientas BPM como JBPM y tecnologías orientadas a aspectos. Las pruebas de concepto realizadas y el trabajo en el caso de estudio demuestran que es factible que una organización de tamaño mediano o pequeño pueda contar con un entorno de apoyo fundamentado en herramientas libres.

Otro factor importante que debe resaltarse, es la importancia de los estándares en el momento de comunicación de herramientas de diferente índole. Pese a que JBPM fue el medio que permitió desarrollar el proyecto, presenta aun muchos problemas por la falta de estandarización de sus productos. Hecho que se está intentando resolver en las próximas versiones pero que hasta su desarrollo, sigue siendo un vacío en el momento de escoger a éste como herramienta de workflow.

Por otra parte, el valor de la estandarización se ve reflejado en las herramientas de gestión, que han tomado como estándar de-facto el esquema XML de Microsoft Project, lo que permite con muy pocas modificaciones exportar cronogramas a diferentes herramientas.

Además de los estándares, las licencias también juegan un papel importante en el proceso de la construcción de entorno. Esto se vio claramente reflejado con lo sucedido con Intalio|Works, debido a que sus productos no son libres, sino gratis, diferencia que a primera vista parece irrelevante; esto genera una serie de inconvenientes en el momento de realizar operaciones que no se encuentran contempladas dentro de la herramienta, como en este caso hacer monitoreo a los procesos de negocio desde código, acción que no se encuentra contemplada en Intalio|Works. Además, el uso de sistemas con código abierto garantiza en cierta medida que la información no se encuentra centralizada por una persona, sino que muchos desarrolladores a través del mundo colaboran y documentan los sistemas desarrollados.

6.1 CONCLUSIONES CON RESPECTO A HERRAMIENTAS.

Debido a que para el desarrollo del entorno se realizó una exploración de tres tipos diferentes de herramientas, cada una arroja un tipo específico de conclusión dependiendo de la experiencia que se tuvo con cada una de ellas, a continuación se definen dichas conclusiones dependiendo de la herramienta trabajada.

6.1.1 A nivel de herramientas de modelado.

Las herramientas de modelado BPMN, pese a existir hace un buen tiempo en el medio, aun no tienen un buen grado de estandarización, debido a que poseen una estructura propia dependiendo de las funcionalidades que ofrece cada una de estas. Esto se evidencio claramente con la unión de Intalio Designer e Intalio Server, las cuales interoperan muy bien pero cierran la puerta a herramientas externas que pudiesen agregar funcionalidades extra. Esto limita en gran medida las funcionalidades y tecnologías que se utilizarían en desarrollos basados sobre estas herramientas.

Por otro lado JBPM, utiliza por defecto Graphical Designer, herramienta de modelado, ya que esta sólo define la forma en la cual va a fluir la información dentro del modelo de procesos de negocio, lo cual ofrece libertad en el momento de crear un desarrollo sobre esta, ya que no liga el modelo de procesos a una sola tecnología. En este caso se implemento sobre las tecnologías ofrecidas por Java.

Sin embargo tiene un gran vacío en cuanto al modelado, y es que no soporta en su totalidad el estándar BPMN, lo que implica que el modelo creado sobre otras herramientas BPMN será diferente en esta.

Dentro de las herramientas BPM o herramientas workflow utilizadas en la implementación del entorno, es notable una gran diferencia respecto a su estructura y las funcionalidades provistas por cada una de ellas. Intalio|Works, es una potente herramienta workflow, en la cual es posible realizara varias de las funcionalidades que requería el entorno, tales como la administración de los procesos del negocio, para definir un flujo continuo de información. Pero en el punto de obtener la información de los procesos y de las instancias de estos posee grandes dificultades, ya que la organización Intalio, ofrece una herramienta diferente encargada de monitorear y auditar la información que hace parte de estos proceso, esta herramienta es denominada Intalio BAM, la cual es propietaria y sale del contexto de software open source que procura el entorno de soporte para el proceso de desarrollo de software a través de BPM.

Lo anterior permite concluir que Intalio|Works es una herramienta orientada a las organizaciones grandes, que pretenden definir sus procesos a través de herramientas ya existentes. Esto debido a que estas entidades no desean invertir en desarrollo de sistemas, sino en definir claramente sus procesos independientes de la plataforma en la que funcionen.

Por otro lado, JBPM resulta ser una gran herramienta workflow que proporciona un API manipulable, completo y administrable, por medio del cual es posible crear una aplicación soportada por las tecnologías ofrecidas por Java para obtener la información de los procesos y sus instancias.

Esto se debe a que JBPM se orienta más hacia ser utilizada en entornos de desarrollo, para definir los flujos de las aplicaciones a desarrollar. Esto claramente se evidencia con la publicación y amplia documentación de las APIs que dan herramientas a los desarrolladores de software para controlar sus flujos a través de esta tecnología.

6.1.2 A nivel de herramientas workflow.

Existe una gran diversidad de herramientas workflow, las cuales de igual forma que las herramientas de modelado BPMN, poseen una estructura propia, la cual especifica funcionalidades propias para herramienta.

Intalio Server, es una gran herramienta workflow que proporciona grandes prestaciones en un desarrollo que este soportado solo por tecnologías propias de la empresa Intalio, además de permitir servicios que otros motores de workflow no ofrecen. Sin embargo estos privilegios solo son para aquellas corporaciones que tengan la capacidad económica de adquirir los servicios. Es una gran herramienta

si se le sabe explotar todas las características que esta ofrece, para lo cual requiere capacitaciones especializadas y adquirir las licencias comerciales de los productos. Sin embargo cabe resaltar que los elementos esenciales, como el diseñador y el motor de workflow se encuentran gratis (aunque no libres) para su uso, es decir se pueden descargar e instalar, sin necesidad de pagar alguna licencia que condicione su uso, pero no es posible cambiar la lógica que las mismas manejan, o no proporcionan las funcionalidades suficientes para cambiar elementos de las herramientas.

La herramienta workflow que mejor se adapta a los requerimientos del entorno es JBPM, ya que esta presenta la mejor forma de obtener la información de los procesos y de sus instancias, punto a partir del cual es posible generar la interoperabilidad entre la herramienta workflow y la herramienta de gestión de proyectos.

6.1.3 A nivel de herramienta de interoperabilidad.

La comunicación entre la herramienta de gestión de proyectos con la herramienta BPM fue realizada a través de las funcionalidades embebidas en la herramienta de interoperabilidad. Esta comunicación se logró gracias a que la herramienta workflow escogida para la implementación del entorno, posee un API público, manipulable y lo suficientemente claro para facilitar el obtener la información referente a los procesos y sus instancias.

Esto demuestra que la comunicación entre este tipo de herramientas es aun una operación compleja, pero no imposible si se eligen las herramientas adecuadas. También dependen del correcto uso de estándares y el acceso adecuado a la documentación.

El esquema planteado en la herramienta de interoperabilidad proporciona un marco eficiente para la obtención de información de los procesos y de sus instancias a partir de la herramienta workflow, con lo cual se logra tener una estructura de interoperabilidad lo suficientemente concreta para desarrollar el entorno propuesto en la investigación.

La elección de tecnologías Java y paradigmas de programación como el orientado a aspectos, suministraron funcionalidades agregadas para utilizar la herramienta de interoperabilidad no solo en el contexto de comunicación entre una herramienta de interoperabilidad y una herramienta de gestión de proyectos, sino que define un entorno adaptable a cualquier tipo de tecnologías y herramientas, definiendo un método por medio del cual se pueda conseguir iguales niveles de integración entre herramientas de otro tipo a las manejadas dentro de la investigación.

La herramienta de interoperabilidad desarrollada en el curso de la investigación ofrece los medios de comunicación suficientes para generar un archivo XML

estandarizado, conteniendo la información más importante de los procesos y sus instancias, dicha información es reflejada de una forma clara sobre el cronograma proporcionado por las herramientas de gestión de proyectos.

6.1.4 A nivel de herramienta de gestión de proyectos.

En el desarrollo del entorno de soporte para el proceso de desarrollo de software, es vital crear un estudio de las herramientas que mejor se acoplaran a los requerimientos exigidos por la investigación. Para esto fue necesario realizar un estudio comparativo de las herramientas de gestión de proyectos que mejor se adaptaran al desarrollo del entorno.

A demás de ello, se logró recoger valiosa información que permite seleccionar la mejor herramienta de gestión de proyectos basados en las necesidades propias.

6.1.5 A nivel del entorno.

Desde la perspectiva de una organización de software se considera importante contar con un entorno que pueda ser aplicado de manera flexible de acuerdo a las condiciones de una empresa y de un proyecto específico. En el actual estudio se consideraron diversidad de escenarios, pero no todas las aproximaciones son adecuadas para todos los escenarios, por lo anterior, para que el entorno pueda ser adoptado con éxito por las organizaciones de software, se deben considerar además de los elementos técnicos, elementos que tienen que ver con el cambio de cultura y con la adopción de mejores prácticas y autodisciplina empresarial a nivel de calidad. El entorno plantea automatización, pero para lograr llevar a cabo este propósito los procesos deben estar maduros, ser eficientes y conocidos por todos los miembros del equipo

6.2 TRABAJOS FUTUROS

A continuación se plantean los trabajos futuros que pueden ser realizados a partir del desarrollo del entorno de soporte para gestión del proceso de desarrollo de software, a través de BPM.

6.2.1 Integración de BPM y motores de workflow en una arquitectura MDA.

Se propone integrar herramientas de diseño de procesos basadas en BPMN y la implementación de una herramienta de homologación con UML, lo que traería entre otras las siguientes ventajas: que pueda mejorar la comunicación entre los expertos del negocio y los arquitectos del sistema, que los arquitectos puedan

realizar sus modelos basados en los procesos diseñados por el grupo de calidad de la organización, que el grupo de calidad pueda realizar cambios al proceso sin que necesariamente se realice un nuevo despliegue de la aplicación. De otro lado al integrar un workflow dentro de la plataforma, se podría conseguir mayor automatización de los flujos de la aplicación, evitando la generación de código redundante en la plataforma de despliegue, minimizando algunos cambios en la aplicación final.

6.2.2 Herramienta de transformación de BPMN a JBPM.

Tomando como base la herramienta desarrollada en el presente proyecto y a partir de los estudios realizados por la OMG, acerca de la homologación de UML y BPMN, se puede conseguir una herramienta de transformación que permita aprovechar los diagramas de tipo UML que ya existen en las empresas. Además se puede aprovechar con el fin de disminuir la curva de aprendizaje de quienes pretenden pasar de UML a BPMN.

6.2.3 Formalizar el modelo de trabajo para diseño y monitoreo del proceso de desarrollo.

De manera que pueda ser divulgado y apropiado por las empresas de software, contribuyendo en sus procesos de mejora continua y aportando a tanto a los grupos de calidad como a los mismos desarrolladores, quienes podrían detectar con mayor facilidad falencias en sus procesos.

6.2.4 Configurar el entorno de apoyo bajo la tecnología JBPM en el laboratorio del grupo de ingeniería de software.

Trabajar un proyecto piloto con una empresa de la ciudad, como un experimento controlado, para mostrar la viabilidad de la solución y de esta manera incentivar el uso de modelo por otras empresas. Además los datos arrojados permitirían una evaluar y detectar fortalezas y debilidades del modelo con el fin de optimizarlo.

En paralelo, trabajar el mismo esquema con un grupo de ingeniería de software de la universidad con el fin de incentivar el uso de estas tecnologías, para que los futuros profesionales, se encarguen de motivar cambios en las organizaciones que lleguen a integrar, desplegando así el concepto de calidad hacia las empresas desde el interior de la universidad.

6.2.5 Modelo de transformación e integración entre SPEM y BPMN.

Siendo SPEM el estándar adecuado para la definición de procesos en el desarrollo de software y BPMN una poderosa notación de los procesos de negocio, es factible pensar en la medios de transformación entre las dos con el fin de

aprovechar las bondades de cada una y permitir mayor fluidez en la comunicación de los miembros del equipo de desarrollo de software con los demás miembros de la organización.

6.2.6 Gestión distribuida de proyectos con apoyo de administración de métricas.

El gerente de proyecto no es el único interesado en conocer el progreso del mismo, otros miembros del equipo desean conocer también su estado aunque posiblemente con perspectivas diferentes, por tal razón el entorno propuesto se puede adaptar para que permita la visualización desde diferentes vistas. De igual forma es posible obtener información sobre métricas deseables para el proceso como son métricas de esfuerzo, de seguimiento, de proximidad entre otras, lo que redundara en la mejora continua de procesos, la detección de puntos críticos, puntos de eficiencia, pasos innecesarios, lo que aumentaría la calidad del proceso como tal.

BIBLIOGRAFIA

1. ACUÑA Silvia. Procedimiento para determinar las necesidades de competencias en organizaciones desarrolladoras de Software. Revista Colombiana de Computación. Vol 2. No 2. P 38-65.
2. ALONSO Laura. PESADO Patricia, BERTONE Rodolfo, RAMÓN Hugo, PASINI Ariel, ESPONDA Silvia, Calidad en el desarrollo de Sistemas de Software. [En línea]. <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/073.pdf>. [Citado 16 de Marzo de 2009].
3. ALVEZ Pablo., FOTI Patricia., SCALONE Marco., Proyecto Batuta - Generador de Aplicaciones Orquestadoras Facultad de Ingeniería - Universidad de la República Junio 2006.
4. Adolfo Guzmán Arenas, Centro de Investigación en Computación (CIC), Las causas más comunes de falla en la implantación de mejoras en software, Instituto Politécnico Nacional (IPN) – México. Enero 2010.
5. ASTEASUAIN Fernando., CONTRERAS Bernardo E., ESTÉVEZ Elsa., FILLOTTRANI Pablo Reprogramación Orientada a Aspectos: Metodología y Evaluación. Departamento de Ciencias e Ingeniería de la Computación Universidad Nacional del Sur. Argentina.2003.
6. ASTEASUAIN Fernando., CONTRERAS Bernardo E., PROGRAMACIÓN ORIENTADA A ASPECTOS, Análisis del paradigma, Tesis de Licenciatura, UNIVERSIDAD NACIONAL DEL SUR, Octubre de 2002.
<http://www.lafhis.dc.uba.ar/~ferto/docs/tesis.pdf>
- 7 Clemens Szyperski. Component Software. Beyond Object-Oriented Programming. ISBN: 0-201-17888-5. Addison-Wesley, 1998.
8. Interoperability for Enterprise Information Systems, Computer Technology Research Corporation; 1st edition, 202 pages. ISBN-10: 1566079756, August 2008.
9. Borges, Alejandro E. 2007; Marzo de 2007. *La interoperabilidad y los estándares abiertos, base del desarrollo de la sociedad de la información*. Colegio Oficial Asociación Española Ingenieros de Telecomunicaciones.. [En línea]. Disponible desde internet en: <http://www.coit.es/publicaciones/bit/bit161/36-39.pdf> [Con acceso el: 14 de Julio de 2009].
- 10 CABRERA Armando., SOLANO Raquel., MONTALVÁN Mayra. Proceso de ingeniería del software. 2007. En Pdf. P. 7-15

11. GALINDO M. Jesús., ALFONSO Àlex., Proceso de ingeniería del software, Universidad Oberta de Catalunya, UOC, 2005, ISBN 8497076680, 9788497076685 146 Pág.
12. CÁCERES Paloma., MARCOS Esperanza., Hacia un proceso Metodológico dirigidos por modelos para el desarrollo ágil de sistemas de información Web. Grupo de Investigación Kybele. Universidad Rey Juan Carlos. Madrid.2002
13. Chan, Daniel K. C.; Leung, Karl R. P. H. Software Development as a Workflow Process. IEEE Computer Society Washington, DC, USA
14. COLCIENCIAS Y SENACONVOCATORIA PARA EL APOYO AL FORTALECIMIENTO DE LA CAPACIDAD NACIONAL EN CALIDAD DE SOFTWARE.
http://afrodita.unicauca.edu.co/~ecaldon/docs/spi/COMPETISOFT_v02_27-11_2315.pdf
15. Consejo Técnico de Interoperabilidad. 2008. 21 de Noviembre de 2008; Documento de Referencia, Marco Genérico de Interoperabilidad (MGI V1.0). [En línea]. http://www.cidge.gob.mx/doc/MG_Interop_Dic08.pdf. [citado 17 de Julio de 2009]
16. Debnath, D. Riesco, G. Montejano, "Supporting the SPEM with a UML Extended Workflow Metamodel", ACS/IEEE International. 2006
17. THOMAS, Erl. SOA Design Patterns, Prentice Hall PTR; 1 edition (January 9, 2009). 800 p. ISBN-13 978-0136135166.
18. Ivar Jacobson , El Proceso Unificado de Desarrollo de Software, ISBN-10: 8478290362. (Noviembre 2000), Addison Wesley Publishing Company . <http://www.sg.com.mx/content/view/487>
19. SOMMERVILLE Ian, Ingeniería del software (Métodos ágiles 361p.) *Pearson educación* 7ed. ISBN 8478290745, 9788478290741. 2005 687 Pág.
20. DOMINGO Eliseo., SENENT Martínez., Pablo ARAGONÉS ., y SANCHEZ Miquel . Ingeniería de proyectos, Caminos de Vera, 2000. 238p. ISBN 84-7721-897-8.
21. F. A. Zorzán y D. Riesco Transformation in QVT of Software Development Process based on SPEM to Workflows, IEEE LATIN AMERICA TRANSACTIONS, VOL. 6, NO. 7, DECEMBER 2008.

http://ewh.ieee.org/reg/9/etrans/ieee/issues/vol6/vol6issue7dec.2008/6tla7_14zorzan.pdf.

22. Fischer Layna, Stephen A. White PhD, Derek Miers, BPMN Modeling and Reference Guide. Future Strategies Inc., Lighthouse Pt, FL. Agosto 28, 2008. ISBN-10: 0977752720.

23. Lederer Antonucci Ph.D, Business Process Management Common Body Of Knowledge, CreateSpace (March 8, 2009), ISBN-10: 1442105666.

24. Fuggetta, A. (2000): Software Process: A Roadmap. International Conference on Software Engineering.[En línea]. <http://portal.acm.org/citation.cfm?id=336521>. [Citado 14 de 05 de 2010.]

25. GARCÍA Francisco., LAGUNA Miquel., MÁRQUES José., MAUDES Jesús., Mecanos: Soporte de Diferentes Niveles de Abstracción en los elementos Software Reutilizables. En Pdf.Octubre 1998.

26. STEPHEN, A. White, DEREK, Miers. BPMN Modeling and Reference Guide. Future Strategies Inc., Lighthouse Pt, FL (August 28, 2008). 226 pages. ISBN-13: 978-0977752720.

27. Gradmann, Stefan. Digital Preservation Europe. *Interoperabilidad. Un concepto clave a las bibliotecas digitales a gran escala persistentes*. [En línea]. http://www.digitalpreservationeurope.eu/publications/briefs/es_interoperabilidad.pdf. [Citado el: 14 de Julio de 2009].

28. HELMUT Sharp. Workflow Modeling: Tools for Process Improvement and Application Development, rtch House Publishers; 2 edition (October 31, 2008), ISBN-10: 1596931922.

29. HOLLINGSWORTH David. ,FUJITSU Services., UNITED Kingdom., TECHNICAL Committee .The Workflow Reference Model 10 Years On David Hollingsworth. En : PC Pdf. Mayo, P. 6-9.

30. Tom Debevoise, The Microguide to Process Modeling in BPMN, BookSurge Publishing (July 11, 2008), English, ISBN 1419693107, 142 pages.

31. TORRE César., GONZÁLEZ Roberto., Arquitectura SOA con tecnología Microsoft (Ventajas 22), Editorial Krasis Press, ISBN 8493548979, 9788493548971

32. KISELEV ivan. Aspect-Oriented Programming with AspectJ. Sams (July 17, 2002). 288 pages. ISBN-13: 978-0672324109.

33. Microsoft. 2006. La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada la mundo real. 12 de 2006.[En línea] www.microsoft.com/soa.
34. Mohammadi M, Binns A, Rajabifard A and Williamson, 2006 Spatial Data Integration, 17th UNRCC-AP Conference and 12th Meeting of the PCGIAP, Bangkok,
<http://www.geom.unimelb.edu.au/research/SDI%research/publicatios/files/Spatial/%20Data%20Integration.doc>
35. Manso, M. A., y otros. 2008. Infraestructura de Datos Espaciales en España. *Modelo de Interoperabilidad Basado en Metadatos (MIBM)*. [En línea] 2008. http://www.ideo.es/resources/presentaciones/JIDEE08/ARTICULOS_JIDEE2008/articulo36.pdf. [Citado 17 de Julio de 2009.]
36. Marsili, Diego. 2007. ¿Qué es SOA, la arquitectura orientada a servicios? [En línea] 16 de 05 de 2007. <http://tecnologia.infobaeprofesional.com/notas/46399-Que-es-SOA-la-arquitectura-orientada-a-servicios.html?cookie>. [Citado el: 25 de 04 de 2009.]
37. MELÉNDEZ Joaquim .,LÓPEZ Beatriz., , WISSEL Heiko., HAASE Henning., LAATZ Kathleen, and Oliver S. Towards Medical Device Maintenance Workflow Monitoring Grosser. World Academy of Science, Engineering and Technology 54 2009. [En línea]. <http://www.waset.ac.nz/journals/waset/v54/v54-19.pdf>. [Citado 14 de Mayo de 2009].
38. Montliva, Jonás A. 2006. Modelado de procesos de Software. [En línea] 24 de 10 de 2006. [En línea]
<http://www.vaneduc.edu.ar/Uai/facultad/sistemas/informes/Conferencia%20Modelado%20de%20Procesos%20SW.pdf>. [Citado 21 de 03 de 2009.]
39. Moprosoft: el nuevo modelo que impondrá una norma mexicana para la calidad en la industria del software. Entrevista con la Dra. Hanna Oktaba, presidenta de la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS). 2003. <http://www.iie.org.mx/boletin032003/ind.pdf>
40. Jean-Marie Chauvet, Services Web avec SOAP, WSDL, UDDI, ebXML, Eyrolles (March 13, 2002), ISBN 2212110472, 544 pages.
41. Mouriño, Fernando Fernández. 2002. Mejora e innovación de procesos. [En línea] 09 de 2002.
<http://www.gestiopolis.com/canales/gerencial/articulos/44/mejinnoproceso.htm>. [Citado el: 05 de 04 de 2009.]

42. NIETO Juan Manuel. Introducción a la Programación Orientada a Aspectos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla, España [En línea]. <http://lsi.ugr.es/~pdo/MaterialTeoriaAlumnos/Tema%206.-%20Dise%F1o%20OO/POA.pdf>. [Citado el: 05 de 04 de 2009.]
43. PIATTINI Velhuis, Mario G. , JADWIGA Oktaba, Hanna ., OROZCO Mendoza., Maria Julia ,ALQUICERA Esquivel . COMPETISOFT, Mejora de Procesos Software para Pequeñas y Medianas Empresas y Proyectos Año de edición: 2008. 250 p. ISBN: 978-84-7897-901-1. Edi Ra-Ma. (Competisoft,).
44. PEREZ J, García M. XMI: XML Metadata Interchange [en línea]. Valencia, España. www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/192000.doc. [consultado 21 de **Abril** de 2009].
45. QUINTERO Antonia M^a Reina .Visión General de la Programación Orientada a Aspectos. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Diciembre, 2000.
46. Riesco, Daniel. 2004. Un Workflow que Automatice los Procesos de Negocios del Proceso Unificado Rational. [En línea]. <http://users.dsic.upv.es/~dromero/papers/wicc2004.pdf> . [Citado 14 de Mayo de 2010]
47. Riesco, Fabio A. Zorzan y Daniel. DESARROLLO DE SOFTWARE DEFINIDOS CON SPEM. [En línea] <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/070.pdf>.2007. [Citado 12 de Octubre de 2009]
48. SEGARAN Toby. Programming the Semantic Web. O'Reilly Media; 1 edition (July 14, 2009). 304 p. ISBN-13: 978-0596153816.
49. RUIZ, J. (2008) "Organizational Analysis of Small Software Organizations: Framework and Case Study" en H. Oktaba y M. Piattini (eds) *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*.
50. Layna Fischer, **2007 BPM & Workflow Handbook**, Future Strategies Inc.,; 1st edition (May 22, 2007), **Language:** English, **ISBN-10:** 0977752712, 320 pages
51. S. Humphrey Watts. Introducción Al Proceso Software Personal . Pearson Publications Company; 1st. edition (October 2001). 328 pages, SBN-10: 8478290524.
52. Schekkerman J 2004 President of the Institute for Enterprise Architecture Development (IFEAD). Roadmap to implementation. WWW document, <http://web-services.gov/region4soa10104.ppt>

53. SHUPPENIES Robert., STEINHAUER Sebastian. Software Process Engineering Metamodel., 2005.
54. Telefónica, España. 2009. Interoperabilidad en la e-administración. [En línea] Mayo de 2009. <http://www.socinfo.info/seminarios/cooperacion4/telefonica.pdf>. [Citado el: 17 de Julio de 2009.]
55. TORAL, Janette., Introduction to Software Process Improvement.2001. [En línea]. <http://www.cmmphilippines.com/intro.html>. [Citado 17 de Julio de 2009.]
56. SOMMERVILLE Ian. Ingeniería del software. 7 ed. Madrid.: Pearson Educación, S.A, 2005. 687p. ISBN10:84-7829-074-5.
57. VILAR José. Cómo mejorar los procesos en su empresa. El control estadístico de procesos (spc).Fundación Confemetal S.A, 1999. 325 p.
58. VILLARROEL Rodolfo H., GÓMEZ Yessica M., GAJARDO Román., RODRÍGUEZ Oscar Implementation of an Improvement Cycle using the Competisoft Methodological Framework and the Tutelkan Platform. CLEI ELECTRONIC JOURNAL, VOLUME 13, NUMBER 1, PAPER 2, APRIL 2010 (Tutelkan, 2009)
59.]WC3. 2009. World Wide Web. [En línea] Julio de 2009. <http://www.w3c.es/>. [Citado el: 14 de Julio de 2009.]
60. WfMC. <http://obe.sourceforge.net/wfmc/wfmc.html>. [En línea]
61. Wil van der Aalst, Kees Max van Hee, Workflow management: models, methods, and systems, Edición ilustrada, MIT Press, 2004.
62. WHITE Stephen ., Introduction to BPMN En : PC Pdf. Marzo,2004, no. 35, P. 1-10.
63. WORKFLOW MANAGEMENT COALITION WORKFLOW STANDARD-PROCESS DEFINITION INTEFACE, 1998. En PC Pdf. Marzo, 2009, p. 21-44
64. WORKFLOW. [Citado el] <http://www.ccee.edu.uy/ensenian/catsistc/docs/Workflow.pdf>
65. ZIMMER, B 1989 Software quality and productivity analysis at Hewlett-Packard, Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International. 1989 articulo de revista

66. Zorzan, F. A.; Riesco, D. Development Process based on SPEM to Workflows.2008

67. Chrissis, Mary Beth & Konrad, Mike & Shrum, Sandy; (2007). CMMI Second Edition Guidelines for Process Integration and Product Improvement. Pearson Education, Inc., Boston, MA, USA, 676 p.

7. ANEXOS

A continuación se definen los anexos que hacen parte del desarrollo del entorno propuesto en la investigación.

7.1 ANEXO A: ANÁLISIS COMPARATIVO DE HERRAMIENTAS.

7.2 ANEXO B: DESARROLLO DEL CASO DE ESTUDIO SOBRE INTALIO|WORKS.

7.3 ANEXO C: DESARROLLO DEL CASO DE ESTUDIO SOBRE JBPM.

7.4 ANEXO D: DESARROLLO DE LA HERRAMIENTA DE GENERACIÓN XML PARA HERRAMIENTAS DE GESTIÓN DE PROYECTOS.