



UNIVERSIDAD OBERTA DE CATALUNYA

*Herramienta Visual para Diseñar
formularios Web “WformDesigner”*

Especialidad:

**Administración Web y comercio electrónico en
entornos de software
libre**

Autor:

Wilman Chamba Zaragocín

Consultor:

Manel Zaera Idiarte

18 de Enero del 2011

Loja – Ecuador
2011

Licencia

Copyright (c) 2011 Wilman Patricio Chamba Zaragocín. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada GNU Free Documentation License.

Resumen

El auge de las aplicaciones bajo entorno web, hoy en día hace que dichas aplicaciones posean arquitecturas bien definidas tales como multi capas (n-tiers), servicios, componentes, etc., originando que sean escalables, seguras y amigables.

En cuanto a la amigabilidad de las aplicaciones web, se centran en sus vistas que sean más atractivas y fácil de usar para un usuario común, llegando con esto obtener Rich Clients (clientes con interfaz rica) para web. Sin embargo a un desarrollador de software para entornos web, le toma mucho tiempo desarrollar formularios o vistas con los componentes estándares del HTML, peor aún si se quiere agregar a las vistas ciertos comportamientos adicionales en la apariencia necesitando el uso de la tecnología AJAX para las diferentes plataformas de programación como Java, PHP, Python, Microsoft, entre otras.

El uso en gran medida de las aplicaciones Web, ha originado que los enfoques de desarrollo evolucionen y se valgan en gran medida de las arquitecturas web, como es el caso de BPM (Business Process Management), el cual su corazón es el manejo de los flujos de trabajo en los diferentes procesos de negocio de una empresa, se lo haga todo a través de la red necesitando de las aplicaciones web. Una suite BPM necesita la máquina de ejecución de procesos, el proceso y los formularios de entrada para la alimentación de los procesos. Una de estas suites es jBPM (Jboss BPM) el cual entre sus principales ventajas son: su máquina de ejecución de procesos es abierta y libre pudiendo hacer cualquier tipo de modificación y adaptación, es 100% Java y utiliza tecnología para entornos empresariales (JEE), pero carece de un diseñador de formularios propio de la herramienta, sino que deje la libertad de hacerlo con cualquier IDE o subir el código necesario de dicho formulario para cada uno de los procesos.

El presente proyecto tiene como finalidad facilitar al desarrollador y usuario en general el diseño de formularios en forma visual, haciendo uso de los componentes frecuentemente utilizados como cajas de texto, etiquetas, áreas de texto, calendarios, casillas de selección, listas desplegables, entre otros para generar código XHTML con JavaServerFaces.

Para este fin me he basado en el diseñador de formularios de Joget Workflow, el cual posee una licencia GPLv2, permitiendo modificar su código, esta desarrollado bajo Java, dicho diseñador de formulario tiene un entorno web, los cuales son utilizados he interpretados únicamente por su suite, utiliza en la vista páginas JSP y dentro de ellas JSON (JavaScript Object Notation), para el negocio utiliza el framework Spring y para la persistencia Hibernate. Joget persiste en la Base de Datos MySQL y se sugiere alojarse en un servidor Apache Tomcat, pero podría alojarse en cualquiera servidor de aplicaciones web para Java como Jboss Server Application, Glassfish, entre otros.

El diseñador de formularios permite la gestión de los formularios como crear, editar y/o eliminar formularios, y generar archivos código XHTML para JSF (Java Server Faces) ayudándolo en gran medida a un desarrollador especialmente de JAVA a visualizar como quedará su formulario, menor tiempo en la construcción y personalizar su formulario a través de la modificación del código generado.

Índice de contenido

Licencia.....	2
Resumen.....	3
Introducción.....	6
Objetivos del Proyecto.....	7
Objetivo General.....	7
Objetivos Específicos.....	7
Estudio de viabilidad.....	8
Situación Actual.....	8
Alternativas de Solución.....	9
Joget workflow.....	10
Descripción.....	10
Arquitectura.....	10
Puntos a favor.....	11
Debilidades o comportamientos a mejorar.....	11
Bonita Open Solution v5.3.....	12
Descripción.....	12
Arquitectura.....	12
Puntos a favor.....	13
Debilidades o comportamientos a mejorar.....	13
jBPM - JBoss Business Process Management.....	13
Descripción.....	13
Arquitectura.....	14
Puntos a favor.....	15
Debilidades o comportamientos a mejorar.....	16
Solución Seleccionada.....	16
Diagrama de Módulos y roles que intervienen en el proyecto.....	17
Análisis del Sistema.....	19
Determinación de Requerimientos.....	19
Requerimientos Funcionales.....	19
Requerimientos del Sistema.....	19
Perfiles de usuario.....	19
Identificación de Casos de Uso.....	20
Diagrama de Casos de Uso.....	20
Interfaz de usuario.....	20
Principios Generales de la Interfaz de Usuario.....	20
Esquema de Interfaz de Usuario.....	20
Software para el Desarrollo.....	23
Plan de Pruebas.....	23
Pruebas de Integración.....	23
Pruebas de Implantación.....	24
Pruebas de Aceptación.....	24
Diseño.....	25
Arquitectura.....	25
Diagrama de Componentes del sistema.....	25
Descripción del Componente parser2XHTML.....	25

Diagrama de Despliegue del Sistema.....	26
Modelo de Clases Final.....	26
Diagrama de Paquetes	26
Diagrama de Clases por paquete.....	27
Diagrama de Clases por Casos de Uso y/o Componentes.....	28
Especificaciones de Desarrollo.....	28
GNU/Linux.....	28
Servidor de Aplicaciones Web.....	28
MySQL.....	28
Java Developer Kit 1.60.22.....	29
Eclipse Helios.....	29
Apache Maven.....	29
Implementación y Pruebas.....	30
Planificación de actividades para el desarrollo.....	30
Presupuesto.....	31
Desarrollo de Actividades de Implementación.....	32
Pruebas.....	33
Pruebas de Funcionamiento.....	33
Pruebas de Implantación.....	33
Pruebas de Aceptación.....	34
Formación de usuarios.....	34
Experiencias y Resultados.....	35
Conclusiones.....	35
Recomendaciones.....	36
Ampliaciones.....	37
Anexos.....	38
Anexo A: Licencia de Documentación Libre de GNU.....	38
Anexo B: Manual de Usuario.....	44
Ingreso a la aplicación.....	44
Gestión de Formularios.....	46
Diseñando Formularios y generando código xhtml.....	48
Previsualizando Formularios.....	50
Fuentes Bibliográficas.....	53

Introducción

El auge de las aplicaciones bajo entorno web, hoy en día hace que dichas aplicaciones posean arquitecturas bien definidas tales como multi capas (n-tiers), servicios, componentes, etc., originando que sean escalables, seguras y amigables.

En cuanto a la amigabilidad de las aplicaciones web, se centran en sus vistas que sean más atractivas y fácil de usar para un usuario común, llegando con esto obtener Rich Clients (clientes con interfaz rica) para web, ocasionando para el desarrollador de software bajo entornos web, le toma mucho tiempo desarrollar formularios o vistas con los componentes estándares del HTML, peor aún si se quiere agregar a las vistas ciertos comportamientos adicionales en la apariencia necesitando el uso de la tecnología AJAX.

El uso en gran medida de las aplicaciones Web, ha originado que los enfoques de desarrollo evolucionen y se valgan en gran medida de las arquitecturas web, como es el caso de BPM (Business Process Management), el cual su corazón es el manejo de los flujos de trabajo en los diferentes procesos de negocio de una empresa, se lo haga todo a través de la red necesitando de las aplicaciones web. Una suite BPM necesita la máquina de ejecución de procesos, el proceso y los formularios de entrada para la alimentación de los procesos.

La finalidad del proyecto es facilitar al desarrollador y usuario en general el diseño de formularios en forma visual, haciendo uso de los componentes frecuentemente utilizados como cajas de texto, etiquetas, áreas de texto, calendarios, casillas de selección, listas desplegadas, entre otros para generar código XHTML con JavaServerFaces.

WformDesigner se basa en el diseñador de formularios de Joget Workflow, el cual posee una licencia GPLv2, permitiendo modificar su código, esta desarrollado bajo Java, dicho diseñador de formulario tiene un entorno web, el cual permite la gestión de los formularios como crear, editar y/o eliminar formularios, y generar archivos código XHTML para JSF (Java Server Faces) ayudándolo en gran medida a un desarrollador especialmente de JAVA a visualizar como quedará su formulario, menos tiempo en la construcción y personalizar su formulario a través de la modificación del código generado.

Objetivos del Proyecto

Objetivo General

Desarrollar y/o personalizar una herramienta visual para el diseño de formularios web, para ser llamadas desde aplicaciones Web, en este caso desde la suite BPM de JEE “Oficina sin papeles”.

Objetivos Específicos

- Determinar los requerimientos de la herramienta visual para el diseño de formularios web.
- Determinar el(los) framework(s) más acordes para la construcción del software y/o personalización del mismo.
- Establecer el diseño y la arquitectura de la aplicación.
- Realizar el llamado de ejecución del diseñador de formularios desde otras aplicaciones Web y/o de la aplicación BPM.
- Contribuir con la comunidad de software libre y open source con una herramienta de este tipo.

Estudio de viabilidad

Situación Actual

Loxageek es una empresa dedicada a la Ingeniería de Sistemas bajo la plataforma JAVA en Entornos Empresariales, radicada en la ciudad de Quito, Ecuador; pese a ser una empresa novel en el mercado Ecuatoriano se ha ganado el prestigio y reconocimiento en entidades gubernamentales y privadas, debido al manejo y uso en sus desarrollos de tecnologías open source de última generación para la construcción de software, logrando con ello un amplio portafolio de empresas satisfechas.

A mediados del mes de Octubre de este año, se ha establecido con una entidad gubernamental llamada "Gobierno Provincial de Loja" la iniciación del proyecto denominado "Oficina sin papeles" que es un software para la gestión documental de los trámites en los diferentes procesos que se ejecutan de la institución, tomando como plan piloto el departamento Financiero en la cual se acumulan los trámites y se centralizan los procesos más críticos.

Para el proyecto se ha considerado el enfoque BPM (Business Process Management), el cual trata de un enfoque de gestión integral que promueve la eficacia empresarial y la eficiencia mientras se esfuerza por la innovación, la flexibilidad y la integración con la tecnología, por lo tanto, podría ser descrito como un "proceso de optimización de procesos", permitiendo a las organizaciones a ser más eficientes, eficaces y capaces de cambiar su funcionalidad de acuerdo a las mejoras de sus procesos.¹

Entonces considerando el enfoque BPM, se necesita una suite en entorno web con interfaz *Rich - Client* para los técnicos del departamento de Tecnología con herramientas que permitan:

- Modelar gráficamente los flujos de los procesos que se dan en el manejo de los trámites documentales dentro de la institución ya sea para la definición de los nuevos procesos o modificación de alguno de ellos que este implementado. Para conseguir la comunicación correcta entre los técnicos y los encargados de cada departamento dentro de la institución en el modelo de los procesos la notación de BPM 2.0 (BPMN2.0) que consisten en un lenguaje estandarizado para el modelado de procesos con artefactos fáciles de entender para usuarios de sistemas y personas que manejan bien los procesos dentro de la institución.
- Diseñar gráficamente los formularios o pantallas para la gestión documental en base a las necesidades de las actividades y/o tareas desempeñadas en cada uno de los procesos modelados previamente.
- Gestionar el manejo de usuarios y roles dentro la institución.
- Motor de Ejecución del BPM que ejecute las diferentes instancias de los procesos modelados conjuntamente con las pantallas generadas en el diseñador gráfico.
- Consola de monitorización del estado del trámite dentro del flujo de los procesos.

¹ **Wikipedia**, BPM - Business Process Management, http://en.wikipedia.org/wiki/Business_process_management, [2010-10-30]

En base a los requerimientos de la suite todos los módulos o herramientas para su funcionalidad tienen un alto grado de importancia, pero se ha encontrado un problema para el diseño de los formularios, debido a que a la mayoría de suites BPM open source existentes en el mercado no tienen este diseñador, se lo tiene que configurar e integrar con Entorno de desarrollo (IDE), necesitando para ello de desarrolladores con altos niveles de conocimiento en tal IDE y suite BPM; y, por otra parte existen otras pocas suites BPM permiten personalizar los formularios generados pero no tienen componentes propios para al gestión documental.

Como la empresa trabaja con tecnología JAVA para Entornos Empresariales (JEE), debido a que brinda escalabilidad, desempeño, portabilidad y seguridad, se ha elegido la implementación open source para BPM denominada Jboss BPM, que es una suite de la comunidad Jboss (de la empresa Red Hat) que utiliza componentes JEE y se ejecuta bajo esta plataforma. Pese a ser una suite de altas prestaciones y que contiene todos los módulos y herramientas mencionadas anteriormente tiene una debilidad que no existe en su suite un generador de formularios, para ello se lo debe realizar a parte integrándolo con el IDE de Eclipse.

Debido a la naturaleza del proyecto y a las características de la tecnología utilizada que no tiene una herramienta liviana integrada de este tipo, urge la necesidad de integrar una aplicación que facilite el diseño visual de los formularios para la aplicación ofertada, dicho módulo debe empaquetarse en un archivo con extensión war o jar, permitiendo llamarlo desde una aplicación web ya sea a través de un link o enlace, applet o vía jnpl, para que posteriormente se genere el código necesario de los formularios web e integrarlo dentro de la suite del BPM. Este diseñador, debe facilitar el diseño visual de los formularios, con componentes comunes utilizados en los formularios web y además permitir componentes adicionales para la gestión documental, debe permitir guardar los nuevos diseños, como además recuperar los previamente guardados.

Alternativas de Solución

Las suites BPM (Business Process Management) de mayor utilización y prestigio disponibles en el mercado poseen un generador de formularios web, el cual puede venir: integrado pero con componentes simples y básicos como caja de textos, labels, check button, botones entre otros, mientras que otras suites ofrecen su diseñador con configuraciones engorrosas para Entorno de desarrollo (IDE), necesitando para ello de desarrolladores con altos niveles de conocimiento en tal IDE y suite BPM; y, por otra parte existen otras pocas suites BPM permiten personalizar los formularios generados pero no tienen componentes propios para al gestión documental, además la mayoría trabajan con un lenguaje de procesos extensible por ellos mismos y no con la notación estándar de BPM llamado BPMN2.0

La empresa *Loxageek* trabaja con tecnología JAVA para Entornos Empresariales (JEE), debido a que brinda escalabilidad, desempeño, portabilidad y seguridad, se ha elegido la implementación open source para BPM denominada Jboss BPM, que es una suite de la comunidad Jboss (de la empresa Red Hat) que utiliza componentes JEE y se ejecuta bajo esta plataforma. Pese a ser una suite de altas prestaciones y que contiene todos los módulos y herramientas mencionadas anteriormente tiene una debilidad que no existe en su

suite un generador de formularios, para ello se lo debe realizar a parte integrándolo con el IDE de Eclipse o a través de FreeMaker, que no es más un lenguaje que puede ser interpretado por JBPM.

Debido a la naturaleza del proyecto “oficina sin papeles” y a las características de la tecnología utilizada que no tiene una herramienta liviana integrada de este tipo, urge la necesidad de integrar una aplicación que facilite el diseño visual de los formularios para la aplicación ofertada, dicho módulo debe empaquetarse en un archivo con extensión jar y/o war, permitiendo llamarlo desde una aplicación web, para que posteriormente se genere el código necesario de los formularios web e integrarlo dentro de la suite del BPM. Este diseñador, debe facilitar el diseño visual de los formularios, con componentes comunes utilizados en los formularios web y además permitir componentes adicionales para la gestión documental, guardando los nuevos diseños, como además recuperar los previamente guardados.

Entonces, el proyecto de fin de maestría denominado “WformDesigner” esta encaminado a cumplir esta necesidad primordial de la suite BPMN del proyecto “Oficina sin papeles” del Gobierno Provincial de Loja, Ecuador, para lo cual se ha estudiado las siguientes alternativas de solución, todas ellas son open source y escritas en java.

Joget workflow

Descripción

Joget es un sistema manejador de flujos de trabajo basado en formularios orientado para ser conducido y/o impulsado por personas. Esta aplicación stand-alone puede gestionar y automatizar los procesos que son únicos para cada organización, creando un valor añadido.

El sistema sirve como una plataforma a los usuarios para diseñar, implementar y ejecutar los diferentes tipos de flujos de trabajo para los diferentes tipos de procesos organizativos. Con Joget los diseñadores pueden crear flujos de trabajo dinámicos que se pueden actualizar en cualquier momento. Cada actividad se puede asignar a un formulario que captura la entrada del usuario, mientras que actúa como un plug-in para la integración del sistema.²

Arquitectura

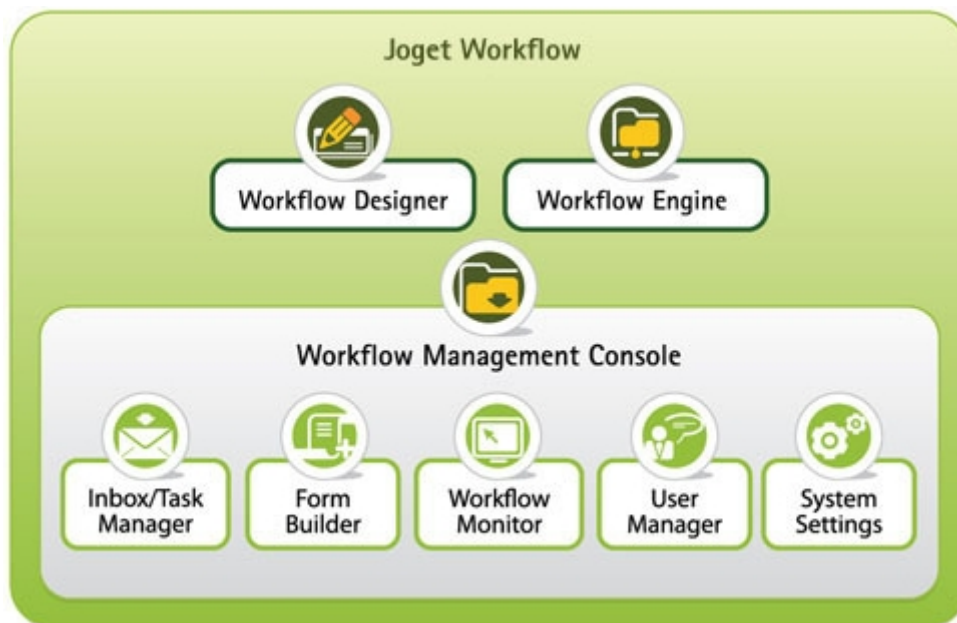
Joget Workflow ha sido desarrollado bajo la plataforma Java, por lo que en su desarrollo utiliza: Beans de Java, JSP, JavaScript, el framework de persistencia Hibernate, Spring framework, Jped, JQuery, JSON, maven, entre otros, y para su ejecución necesita de la configuración de una Base de Datos en MySQL y un servidor de aplicaciones web para java como Apache-Tomcat, GlassFish etc. Por defecto la suite viene embebido con el servidor Apache-Tomcat v6.0.18.

Joget Workflow es compatible con el lenguaje de procesos XPD L y tiene una arquitectura plug-in para ampliar su utilidad. El sistema puede ser utilizado por sí mismo para manejar el flujo de procesos y datos capturados de los formularios, o se pueden integrar perfectamente en diversas formas a los sistemas existentes para crear aplicaciones completas a medida de

2 Joget Workflow, Vista General del Producto, www.joget.org, [2010-11-10]

ajuste a sus necesidades únicas. Capacidad de integración sincrónica y asincrónica. Joget workflow proporciona la capacidad de automatizar, gestionar y mejorar continuamente los procesos de negocio, al tiempo que reduce la carga de trabajo en papel. Se puede integrar con otros portales o aplicaciones del sistema utilizando la API AJAX o JSON.

A continuación se puede observar el esquema de arquitectura y módulos que utiliza Joget Workflow:



1. Ilustración: Arquitectura Joget Workflow

Puntos a favor

- Eliminación de pasos innecesarios debido a la automatización de procesos de negocio.
- Completación de procesos manuales más rápido, coherentes y con errores mínimos.
- Estandarización de procesos y disponibilidad para el seguimiento de la auditoría.
- Pasos predefinidos y documentados.
- Visualmente configurable las reglas y acciones, permitiendo a los procesos de negocio ser automatizados sin la personalización de programadores.
- Intuitiva Interfaz de usuario para facilidad de diseño de procesos.
- Diseñador de formulario integrado con la aplicación web y fácil de agregar componentes preestablecidos y es muy intuitiva.
- La configuración de apuntar-y-click, activa el desarrollo y despliegue rápido de las aplicaciones.

Debilidades o comportamientos a mejorar

- Tiene un motor de BPM propio el cual solo puede ejecutar los procesos modelados por su propia herramienta de diseño.

- Diseñador de formulario necesita tener amplios conocimiento de sus componentes.
- Trabaja con lenguaje de definición de procesos XPD, que es un lenguaje estandarizado poco utilizado, necesitando que trabaje con la notación de BPMN o JPD.

Bonita Open Solution v5.3

Descripción

Bonita Open Solution es una suite BPM que combina tres soluciones en una, la primera es un innovador estudio de modelado de procesos, al segunda un potente motor de BPM y de flujo de trabajo, y la tercera una interfaz de usuario de configuración y ejecución.

Su lema principal es: “Crear aplicaciones basadas en procesos en un solo día con Bonita Open Solution”³, a través de realizar estas tres tareas sencillas como:

Dibujar los procesos como en una pizarra, tratando de ahorrar tiempo y esfuerzo en el diseño de los procesos. Bonita Studio cumple con los estándar de Business Process Management Notation (BPMN) reto que lo ha realizado con una solución sencilla, intuitiva y gráficamente.

Conectar a sistemas o construir aplicaciones personalizables, pues Bonita Studio contiene conectores a más de 100 aplicaciones open source y comerciales externas como mensajería, ERP, CRM, BD y muchos más. Además para personalizar las aplicaciones desarrolladas en Bonita Studio contiene Widgets de arrastrar y soltar, entradas de texto, listas desplegables, botones de radio, etc., para crear formularios de usuario correspondientes a los pasos humanos de su proceso. Trata de que se sienta la optimización de la presentación de los formularios en unos pocos clics, y la importación de plantillas para adaptar su aplicación a su imagen corporativa.

Ejecutar la aplicación con un click, es decir, cuando se han modelado los procesos, agregó los conectores a los sistemas existentes, y se personaliza los formularios de usuario y generar la aplicación de BPM, todo esto en un solo clic.

Arquitectura

Bonita Open Solution (Bonito Studio) esta basado en la plataforma de Desarrollo de Eclipse EMF por lo que su core esta desarrollado en la plataforma JAVA, y trabaja con el estándar de la notación de BPM denominada BPMN2.0.

Bonita Open Solution está compuesto por los siguientes componentes:

- Bonita Studio, una interfaz gráfica totalmente nueva que soporta BPMN (Business Process Management Notation) estándar. Cuenta con una amplia gama de conectores, listo para ser vinculados a las especialidades más comunes soportados son las soluciones (IBM, Oracle, Microsoft, Google), así como la fuente abierta es ampliamente utilizado soluciones (Talend, eXo Plataforma, Alfresco, Jaspersoft, SugarCRM, SpringSource, Xwiki).
- Bonita experiencia del usuario (Bonita XP), un portal de BPM concebido para ser lo

3 **Bonita Open Solution**, Vista General del Producto, www.bonitasoft.org, [2010-11-12]

más fácil de usar como un buzón de correo electrónico, que hace que la ejecución de un proceso intuitivo, y ofrece una visión general de la marcha los pasos para la gestión de procesos mejor.

- BPM generación de aplicaciones, generación automática de aplicaciones de BPM completamente independientes que se pueden exportar e instalarse con unos pocos clics en el entorno de producción del usuario.⁴

Puntos a favor

- Eliminación de pasos innecesarios debido a la automatización de procesos de negocio.
- Aplicabilidad del estándar BPMN2.0, permitiendo abrir diagramas almacenados con esta extensión
- API Rica, para la personalización de las aplicaciones.
- Personalización completa de las interfaces Web a través de paneles de propiedades.
- Visualmente configurable las reglas y acciones, permitiendo a los procesos de negocio ser automatizados sin la personalización de programadores.
- Intuitiva Interfaz de usuario para facilidad de diseño de procesos.
- Comunicación amigable con clases de Java.

Debilidades o comportamientos a mejorar

- Tiene un motor de BPM propio dividido en edición comunitario ya profesional el cual solo puede ejecutar los procesos modelados por su propia herramienta de diseño.
- El motor BPM de la edición comunitaria es lento.
- El monitoreo de procesos contiene demasiado información, que muchas de las veces un usuario normal no la entendería.
- No se puede extender la personalización de los formularios, además es poco intuitiva.

jBPM - JBoss Business Process Management

Descripción

jBPM es una flexible suite para el Manejo de Procesos de Negocio (BPM). Establece el puente entre los analistas de negocio y los desarrolladores. Los motores BPM tradicionales tienen un enfoque que se limitan solamente a personas no técnicas, mientras jBPM le da un doble enfoque, ofreciendo características de gestión de procesos en una forma que, tanto los usuarios del negocio y como a los desarrolladores les agrada.

JBPM toma las descripciones de los procesos como entradas, cada proceso representa un flujo de ejecución, esto se lo hace a través de un diagrama gráfico de procesos que es usado como la más básica comunicación entre usuarios no técnicos y desarrolladores.

4 **Bonita Open Solution**, eclipse marketplace, <http://marketplace.eclipse.org/content/bonita-open-solution>
[2010-11-12]

Cada ejecución de una definición de proceso se conoce como una instancia de proceso. jBPM administra las instancias de proceso. Algunas actividades, como el envío de un correo electrónico o ejecutar un script, son automáticos. Otras actividades implican la espera de un suceso externo, como una persona que completa una tarea o una aplicación que llama de vuelta con los resultados de una solicitud. jBPM realiza un seguimiento del estado de la ejecución del proceso durante los períodos de espera.

jBPM se basa en un motor de proceso genérico, que es la base para soportar múltiples idiomas proceso de forma nativa. jBPM5 se centra en BPMN 2.0 como el lenguaje para expresar los procesos de negocio. BPMN 2.0 es una especificación estándar que define una serialización XML y visualización de procesos de negocio, y se puede ampliar (si es necesario) para incluir características más avanzadas. Mientras que jBPM4.4 soporta lenguajes JPDL y BPMN1.2.⁵

Arquitectura

jBPM es un motor de flujo de trabajo escrito en Java y compuesto por: API de Java, herramientas y una definición de lenguaje open source que puede operar como una aplicación web o una aplicación independiente de Java, puede ejecutar procesos descritos en BPEL, BPMN2.0 (en las versiones jBPM5) o en su propio lenguaje de definición de proceso JBoss jBPM que es JPDL. Es liberado bajo la licencia LGPL por la Comunidad de Jboss.

jBPM se basa en el proceso de la máquina virtual (PVM), que es la fundación de la Comunidad de JBoss para soportar múltiples idiomas proceso de forma nativa. La Comunidad de JBoss en la actualidad se centra en la expansión de las lenguas proceso BPMN 2.0 y JBoss jBPM. La fundación PVM hace la prueba pra el futuro jBPM. A medida que cambia el alcance de BPM, jBPM puede agregar nuevos idiomas proceso fácilmente.⁶

JBoss jBPM define en los archivos de definiciones de proceso escritos usando el lenguaje de definición de proceso de Jboss. JPDL es un lenguaje de programación orientada a gráficos (GOP) basado en un modelo de nodos, transiciones y acciones. En este modelo, los nodos son los comandos ejecutados, ya que se encuentran en el flujo de un proceso de definición. Transiciones dirigir el flujo de ejecución de una definición de proceso, y realizar acciones específicas como la lógica de un nodo o en transición evento.

En jBPM, definiciones de proceso se empaquetan como archivos de proceso. Un archivo de proceso se pasa al motor de proceso de JBoss jBPM para la ejecución. El motor de proceso de JBoss jBPM atraviesa un gráfico de proceso, ejecuta las acciones definidas, mantiene el estado del proceso, y registra todos los eventos del proceso.

JBoss jBPM es encapsulado dentro de los siguientes componentes⁷:

Process engine (motor de proceso): Este componente ejecuta las acciones definidas del proceso, mantiene el estado del proceso, y registra todos los eventos del proceso a través

5 **jBPM**, Makes your work flow , <http://www.jboss.org/jbpm/>, [2010-11-14]

6 **jBPM**, wikipedia, <http://en.wikipedia.org/wiki/JBPM>, [2010-11-14]

7 **Jeff Hanson**, Manage your business processes with JBoss jBPM, JavaWorld.com, 05/22/06, <http://www.javaworld.com/javaworld/jw-05-2006/jw-0522-jbpm.html>, [2010-11-15]

de los siguientes componentes:

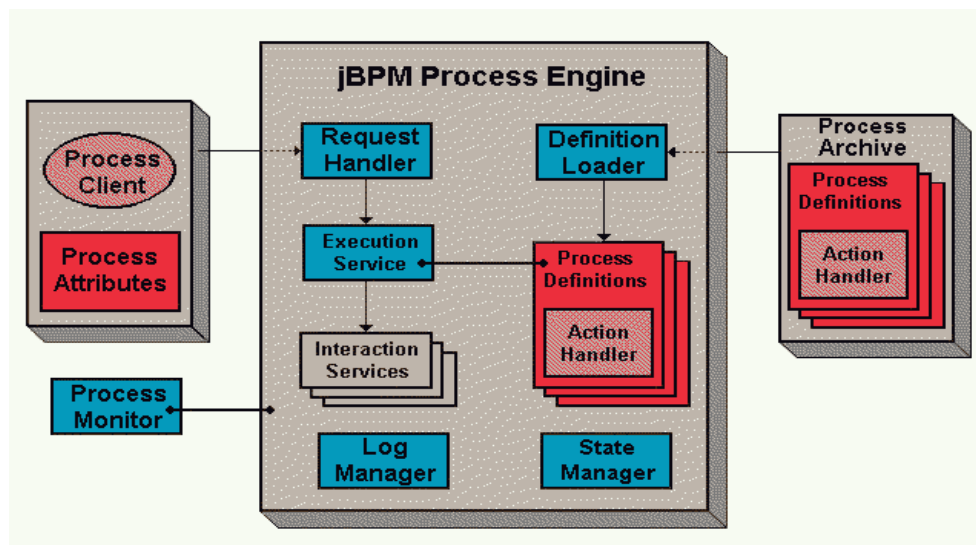
- Un controlador de solicitudes.
- Un administrador de estado.
- Un administrador de registro.
- Un cargador de definición.
- Un servicio de la ejecución.

Process monitor (seguimiento del proceso): Este módulo rastrea, audita y reporta el estado de los procesos que se ejecutan.

Process language (lenguaje del proceso): El lenguaje de definición de proceso jPDL y BPMN que actualmente se esta implementando se basa en la programación orientada a gráficos GOP.

Interaction services (servicios de Interacción): Estos servicios exponen la legalidad de las aplicaciones como las funciones o los datos que se utilizará en ejecuciones de procesos.

A continuación se muestra el esquema de arquitectura de jBPM:



2. Ilustración: Esquema de Arquitectura jBPM

Puntos a favor

- Aplicabilidad del estándar abierto BPMN con las versiones 1.2 y 2.0, en el programación orientada a gráficos GOP, dentro de su lenguaje de definición de procesos JPDL.
- API Rica, para la personalización de las aplicaciones.
- Escalabilidad y personalización de aplicaciones para entornos empresariales.
- Comunicación amigable con clases de Java, 100%.
- Implementación y agregación sencilla de servicios web, Frameworks como: Hibernate, EJB, RichFaces, Seam.
- Capa de persistencia independiente pudiendo se implementada bajo MySQL, Postgres, Oracle.

- Comunidad de usuarios robusta y con gran experiencia.
- El Motor de BPM es 100 % open source, siendo configurable, adaptable, personalizable, y extendible de acuerdo a las necesidades de la empresa.
- Es una suite completa que aplica la mayoría de los estándares abiertos de BPM para el desarrollo de aplicaciones con este enfoque.
- Posee herramienta para diagramación de procesos, monitorización y ejecución con entorno web.

Debilidades o comportamientos a mejorar

- Proporciona una sofisticada plataforma para diseñar y desarrollar flujos de trabajo y sistemas de gestión de procesos empresariales, haciendo necesario de usuarios del proceso y de desarrolladores experimentados.
- No posee una herramienta de diseño y personalización de formularios web para cada tarea y/o actividad de los procesos.
- Integración con entorno de desarrollo de Eclipse para su personalización.
- Esta orientada al usuario de los procesos para el modelado de los procesos, dejando la implementación para el desarrollador.

Solución Seleccionada

Entre las alternativas anteriormente mencionadas me basaré en el Diseñador de formularios Web de la suite Joget WorkFlow debido a que su constructor de Formularios permite:

La creación y gestión de los formularios a través de un entorno visual para la construcción de formularios de múltiples páginas en estilos y tamaños. Dichos formularios se pueden diseñar y editar con una amplia gama de elementos de forma dinámica disponibles en el constructor de Formularios. Alternativamente dispone de una construcción personalizable de formulario que pueden ser fácilmente incorporados dentro del constructor de formularios. También se puede incluir subformularios en un formulario otorgando más detalles e información requerida. Una vez completado de construir un formulario, estos pueden ser mapeados a actividades y/o tareas definidos dentro del proceso de trabajo o workflow.

Tomando en consideración, entonces se estudiará el funcionamiento de este Form Builder con los componentes que utiliza y en base a su estructura de almacenamiento y organización generar archivos fuentes primero en formato xhtml con código para HTML y posteriormente xhtml para código compatible con la especificación de Java Server Faces JSF con la implementación de Jboss llamada RichFaces, lo cual es el valor agregado que se le puede dar a la suite de joget y a la comunidad en general.

El principal riesgo que se puede tener es, que este módulo no permita personalizarlo de acuerdo a las necesidades e integrarse dentro del JBPM con formularios de seam, para lo cual nos quedaría como plan de contingencia generar el código que sea posible de los componentes visuales y/o diseñar y construir nuestro propio diseñador de formulario desde cero pero basándose en la ideal esencial de este.

Cualquiera de estas soluciones son económicamente factibles ya que se economizaría en el uso de herramientas, la cuales deben ser licencias de tipo libre y abiertas, ya que tanto en la suite que tomaremos como base su diseñador y con la que pretendemos integrar son de este tipo por lo que pasaría a ser una prioridad y una exigencia de ellas, evitando el uso de software privativo que ocasionaría pagar por su uso y más aún por el número de usuarios que la utilizarían originando un costo excesivo solo en este rubro para el proyecto.

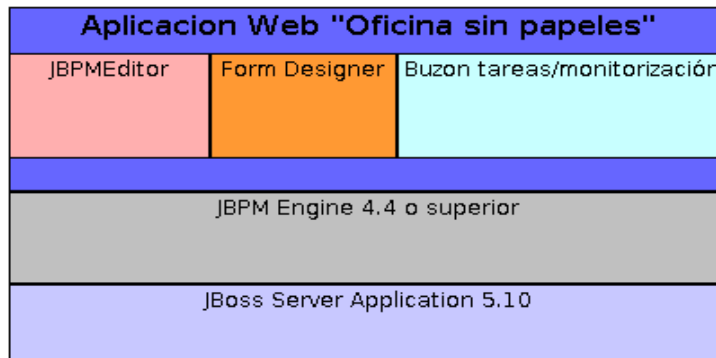
Los principales problema que he encontré fueron:

El módulo FormBuilder de joget no lo pude separar debido a que no es independiente y que los módulos de la suite joget todos tienen una estrecha relación, son complementarios y dependientes entre sí, entonces para solucionar ello, tuve que ir desactivando algunas vistas que no necesitó ya que en su corazón de la aplicación y sus clases todas tienen estrecha cohesión y dependencia como lo mencione anteriormente para no afectar el funcionamiento.

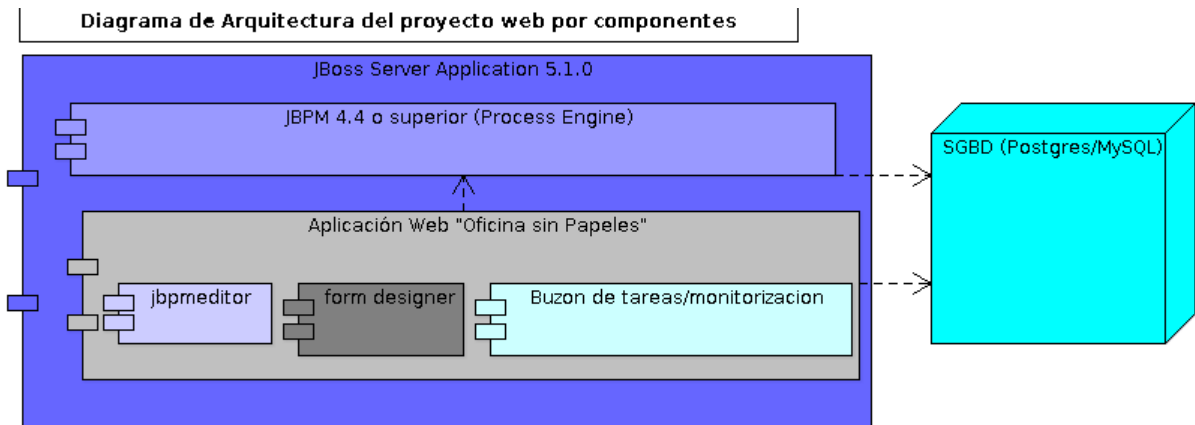
Joget es una caja negra para la ejecución de formularios con componentes no comunes al HTML, tales como editores de HTML y subidores de archivos, ya que en su esquema de formularios sólo indica una marca para que internamente cuando lo encuentre ejecute esa porción mediante código Script, ocasionando que no se pueda generar desde el formulario propiamente dicho estos componentes para la aplicación actual ya que oculta prácticamente toda esa estructura para realizar el parsing o traducción respectiva.

Diagrama de Módulos y roles que intervienen en el proyecto

Diagrama de Arquitectura del Proyecto Web por Capas



3. Ilustración: Diagrama de Arquitectura por capas "Oficina Sin papeles"



4. Ilustración: Diagrama de Arquitectura por componentes "Oficina Sin papeles"

Análisis del Sistema

Determinación de Requerimientos

Requerimientos Funcionales

El diseñador formularios web debe permitir:

- Contener los componentes más frecuentes de un formulario web, como son: cajas de texto, áreas de texto, combobox, labels, botones, mensajes; como además un componente para la subida de archivos que servirá para la gestión documental.
- Visualizar la distribución de los componentes en el diseño de formularios.
- Personalizar las propiedades de los componentes en el diseñador de formularios.
- Guardar los formularios diseñados, generando archivos con extensión xhtml con soporte para las plantillas de facelets y estilos que genera Jboss Seam, tomando en cuenta el código de los componentes de JSF y Jboss Richfaces conjuntamente con lenguaje de expresiones (EL).
- Permitir recuperar y/o editar formularios previamente creados por la herramienta.
- Enlazarse o llamarse fácilmente con aplicaciones web independientes y/o con la suite BPM del proyecto "Oficina sin papeles".

Requerimientos del Sistema

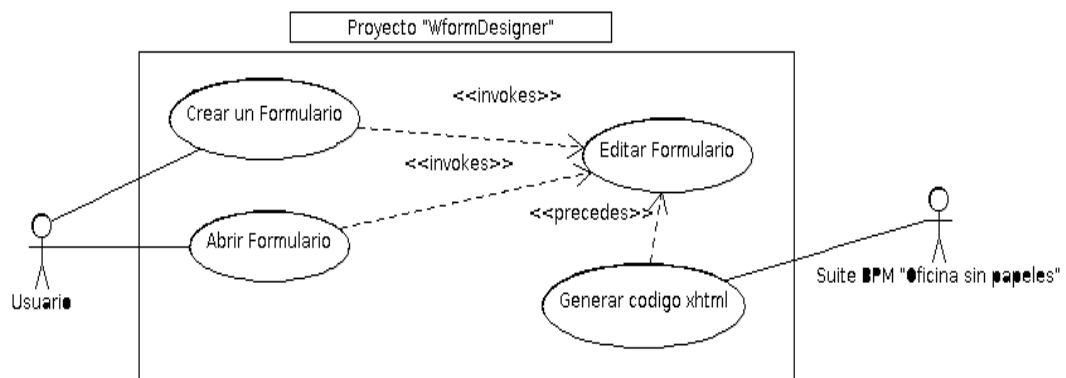
- La plataforma de programación debe ser Java.
- La plataforma operativa debe ser independiente (debe funcionar en cualquier sistema operativo).
- La interfaz gráfica de usuario debe ser sencilla, agradable y fácil de usar.

Perfiles de usuario

Para nuestra herramienta WformDesigner se necesita usuarios con un perfil de conocimientos en procesos o de desarrollo de software que sepan que información debe estar presente en cada formulario para lo cual se puede establecer permisos a dichos usuarios a través de un rol, obviamente dichos roles establecen los permisos necesarios para cada usuario de la aplicación y dependen de la aplicación web o suite BPM que pueden hacer ejecutar cada rol de usuario; con esta explicación que dependemos de la aplicación externa lo llamaremos como usuario.

Identificación de Casos de Uso

Diagrama de Casos de Uso



5. Ilustración: Diagrama de Casos de Uso, proyecto WformDesigner

Interfaz de usuario

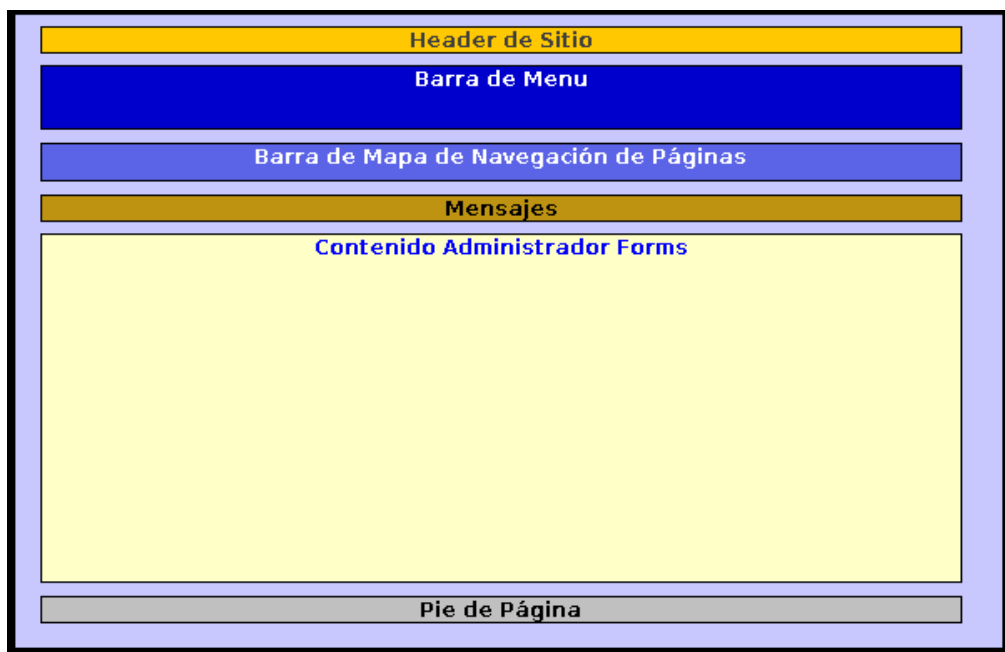
Principios Generales de la Interfaz de Usuario

La aplicación "WformDesigner" poseerá las siguientes características:

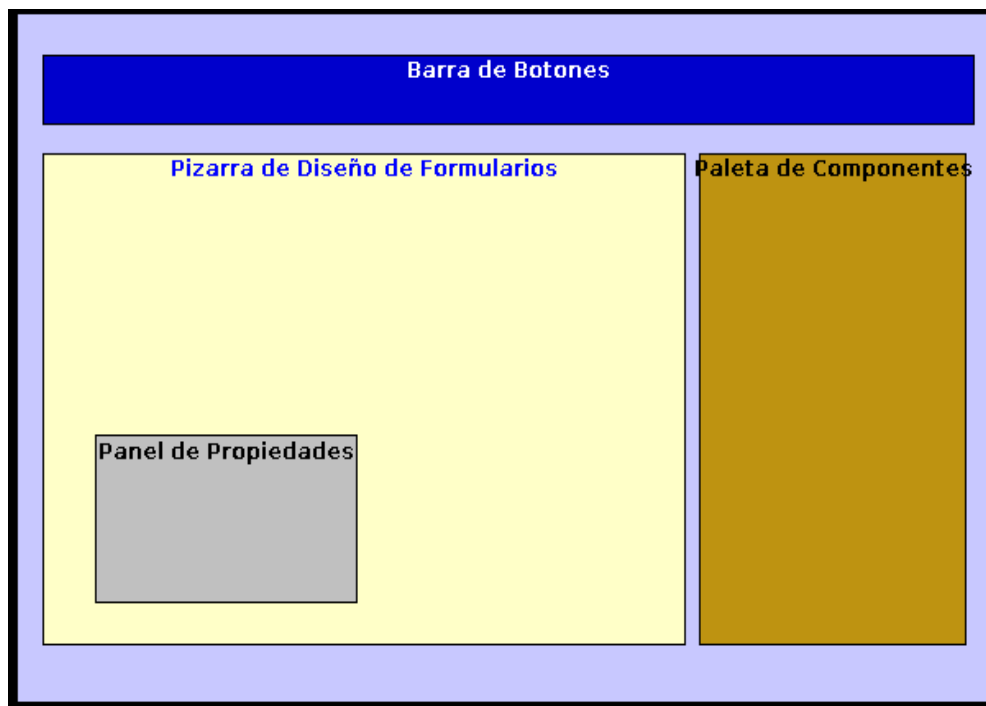
- El acceso a la aplicación y su uso se lo realizará a través de un navegador web.
- La edición de los formularios se lo realizará a través de un panel al estilo pizarra para visualizar como va quedando el formulario.
- Los mensajes de error se mostrarán por pantalla indicando el tipo de error.
- Existirán paneles para: paleta de componentes disponibles para el diseño, edición de propiedades de cada componente, mensajes de error.
- Existirá una barra de menú y herramientas con las opciones disponibles para el diseñador.

Esquema de Interfaz de Usuario

Los siguientes diagramas recogen la interfaz de usuario de la aplicación de WformDesigner tanto para su gestión y diseño visual de formularios basándose en la suite joget workflow.



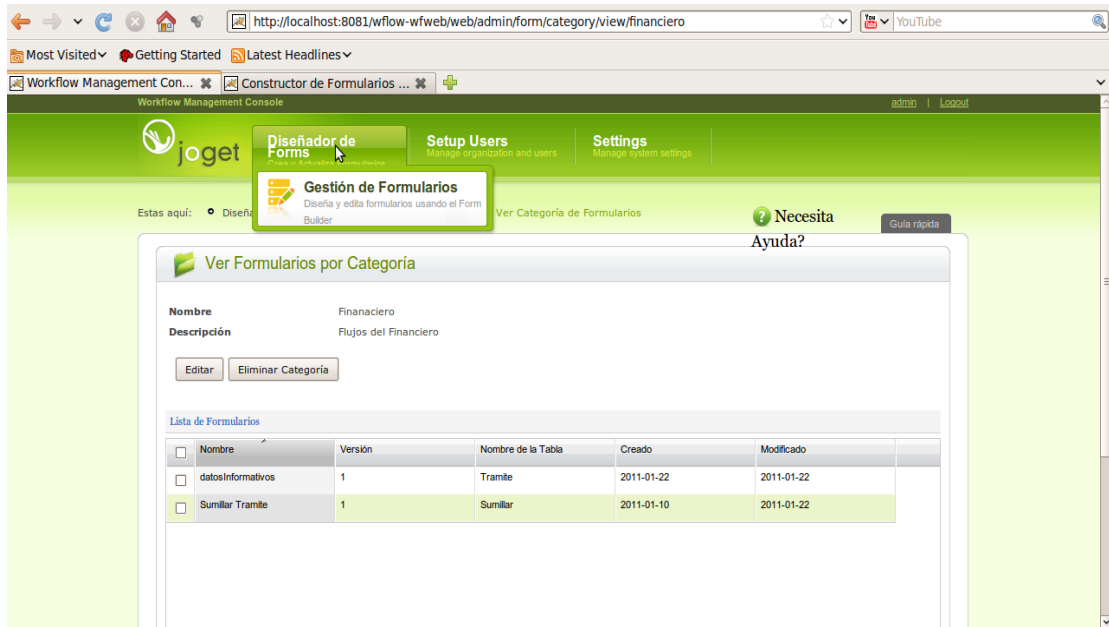
6. Ilustración: Interfaz Gráfica de Usuario para la gestión de Formularios



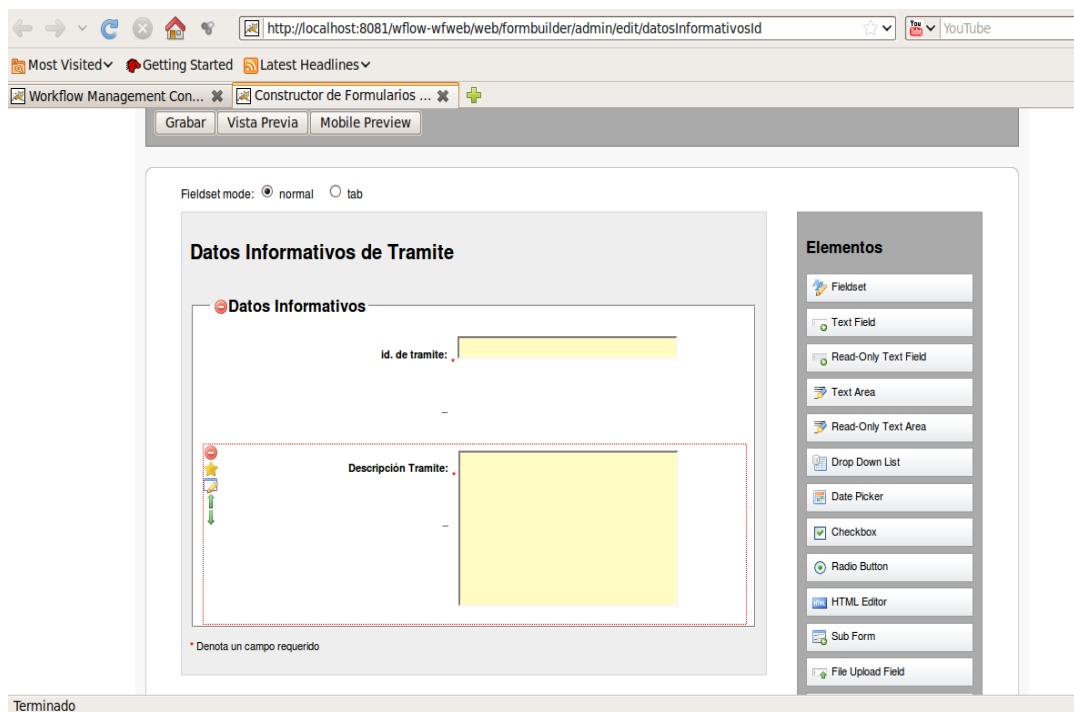
7. Ilustración: Interfaz Gráfica de Usuario para el diseño visual de Formularios

El acceso a la aplicación se realizará a través de un navegador web, donde el usuario que desee acceder a la aplicación, deberá iniciar sesión.

A continuación se indican unas pantallas del sistema en funcionamiento con los módulos activados para el proyecto:



8. Ilustración: Pantalla en Funcionamiento de la Gestión de Formularios



9. Ilustración: Pantalla en Funcionamiento del diseñador de formularios

Software para el Desarrollo

- Eclipse Helios que es Entorno para desarrollo de aplicaciones con los plugins para maven, subversión, Jboss Tools, JEE.
- Herramientas de modelado y diagramación como: ArgoUML y/o Dia,
- Frameworks para el desarrollo, los cuales son clases empaquetadas o librerías que ayudan a la programación aplicando patrones de diseño y desarrollo, estos fueron específicamente:
 - Jtidy utilizado para transformar la estructura de la porción de código generada en html a código completo en xhtml,
 - JDOM utilizado como base para realizar el paser del código de los componentes de Joget a código de los componentes de JSF y la implementación de Richfaces
- software base del constructor de FormBuilder (joget workFlow).

Todo el software a utilizar es OpenSource, la suite Joget Workflow, las herramientas para el modelado y los frameworks a utilizar de Java contienen licencia libres y opensource entre las cuales la principal licencia que ellos utilizan es la GPL (General Public Licence) versión 2 y 3, la cual, esta licencia nos permite mantener el copyright sobre nuestro desarrollo, y es coherente con las licencias del resto de componentes. Nos obliga a distribuir el código fuente resultante e impide su futura comercialización bajo una licencia propietaria.

El IDE Eclipse utiliza la licencia EPLv1.0 (Eclipse Public License)⁸, la cual es una licencia basada en la licencia GPL adaptándola a la organización de Eclipse nombrándola de esta manera, pero que no pierde las libertades de la comunidad libre y OpenSource.

Como es un software comunitario y para la comunidad se adoptará la licencia GPL para nuestra herramienta “WformDesigner”, debido a que coincide de acuerdo a nuestra filosofía.

Plan de Pruebas

Pruebas de Integración

A continuación, se enumera las principales pruebas unitarias y de funcionamiento, extraídas de las funcionalidades e interfaces del sistema :

- Prueba de Recuperación de formularios previamente almacenados.
- Prueba de Generación de código fuente de los formularios al momento de guardar.
- Prueba de visualización de la información y componentes.
- Prueba de Obtención de cada uno de la información que se deberán mostrar al usuario.

⁸ Eclipse Public License v1.0, Eclipse, <http://www.eclipse.org/legal/epl-v10.html> [2010-12-17]

Pruebas de Implantación

Para las pruebas de implantación se contará con un servidor Apache Tomcat, completamente configurado el cual nos servirá para la correcta realización de las pruebas, aquí se testeará la velocidad de recuperación y registro de datos así como el correcto funcionamiento en todos los navegadores

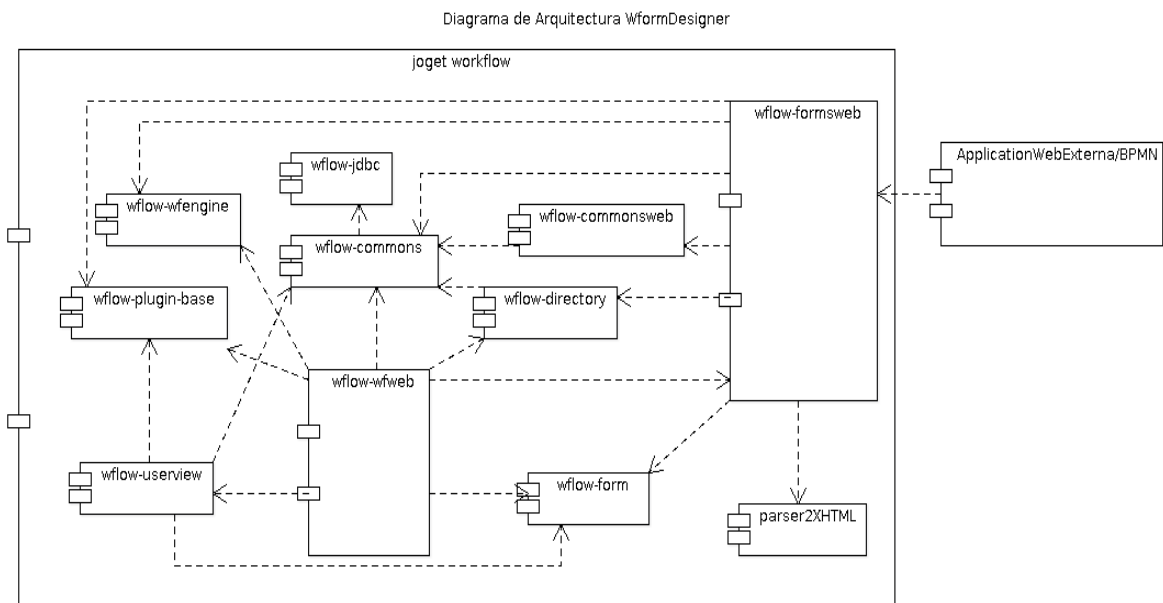
Pruebas de Aceptación

Para las pruebas de aceptación se llevarán los archivos .xhtml generados por la herramienta a una aplicación Web desarrollada en Jboss Seam para ver su presentación.

Diseño

Arquitectura

Diagrama de Componentes del sistema



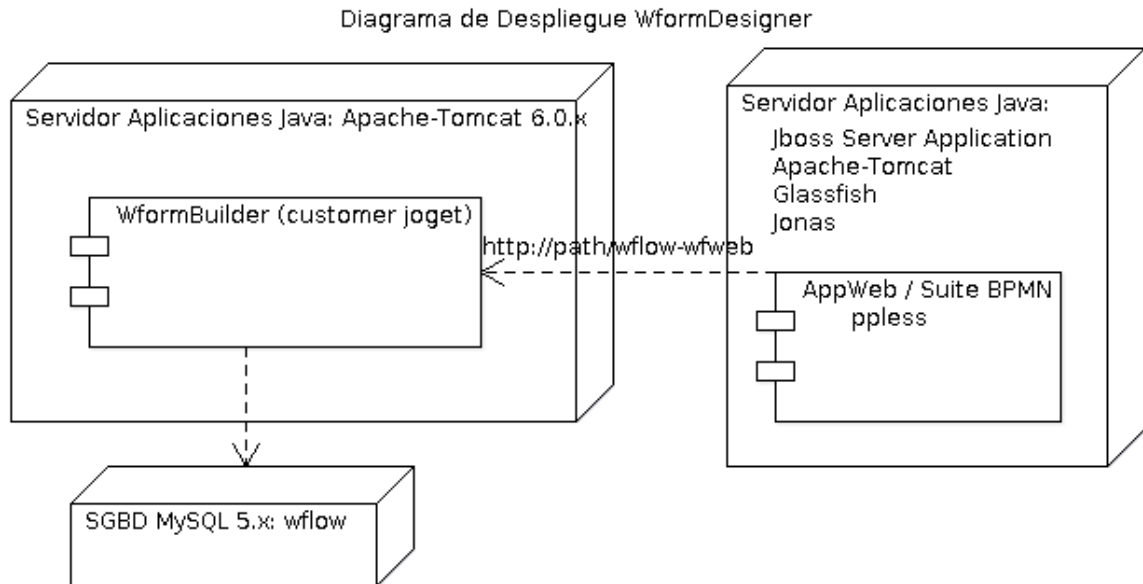
10. Ilustración: Diagrama de Componentes del Sistema

Descripción del Componente parser2XHTML

El componente adicional que permite realizar la transformación de los formularios de joget a archivos XHTML para la especificación de código JSF (Java Server Faces) con la implementación de RichFaces para aplicaciones de Jboss Seam es el **parser2XHTML**, el cual en sí está diseñado para realizar el siguiente trabajo:

A partir del formulario actual que se está editando al momento de guardar en joget, parser2xhtml primero toma el archivo fuente generado por la suite, seguidamente lo transforma a un archivo xhtml bien formado y estructurado, posteriormente realiza el parser a partir de esta archivo para hacer el mapeo respectivo a los componentes JSF y de Richfaces para vistas dentro de aplicaciones Seam, guardando el archivo con extensión .xhtml en la ruta predeterminada por la configuración de la aplicación.

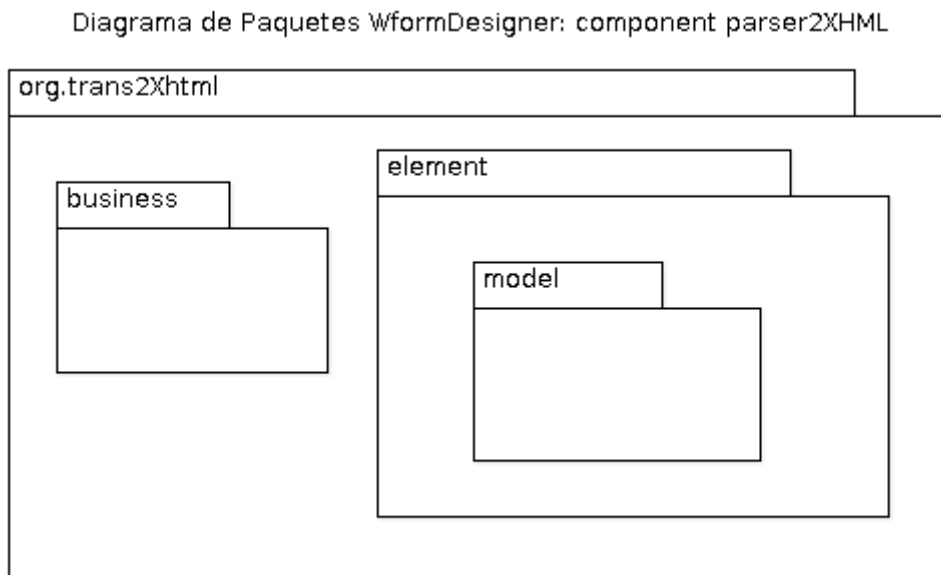
Diagrama de Despliegue del Sistema



11. Ilustración: Diagrama de Despliegue

Modelo de Clases Final

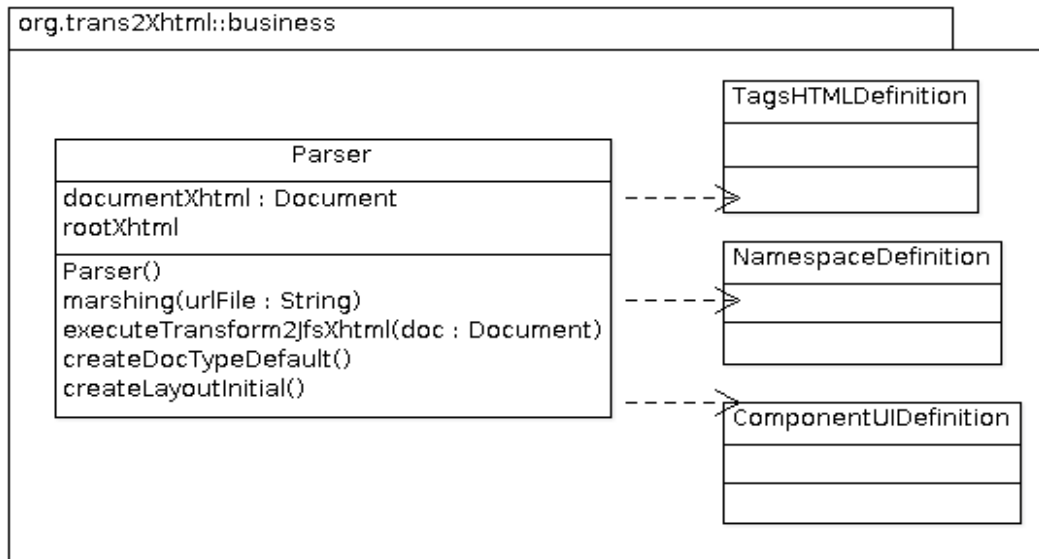
Diagrama de Paquetes



12. Ilustración: Diagrama de paquetes WformDesigner: componente parser2XHTML

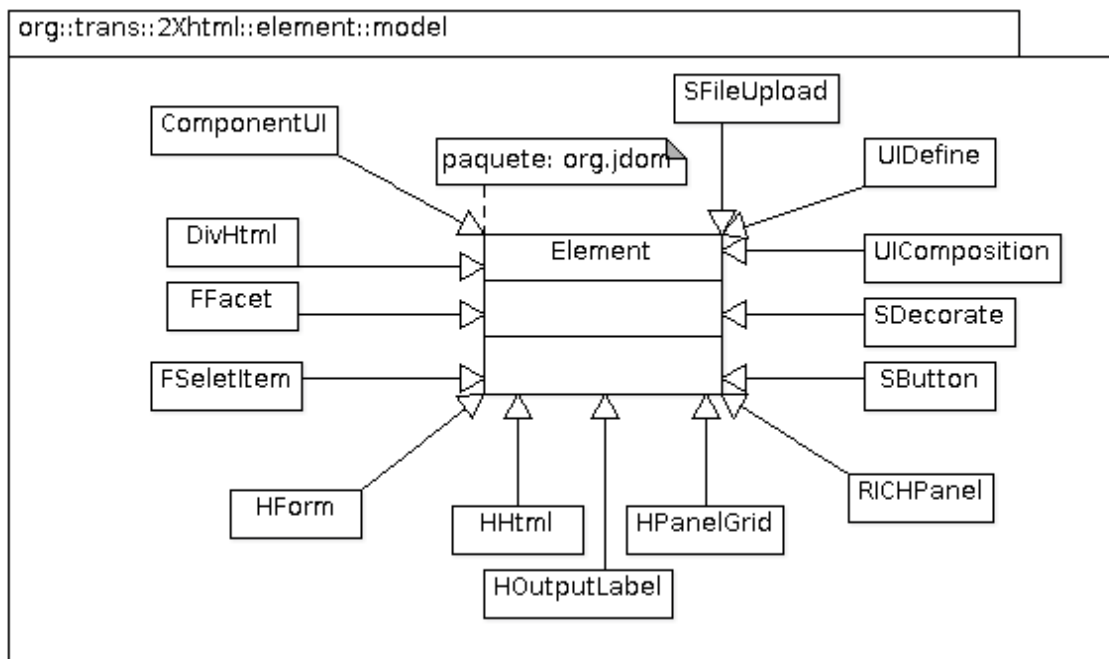
Diagrama de Clases por paquete

Diagrama de Clases para paquete: org.trans2Xhtml.business



13. Ilustración: Diagrama de clases: paquete org.trans2Xhtml.business

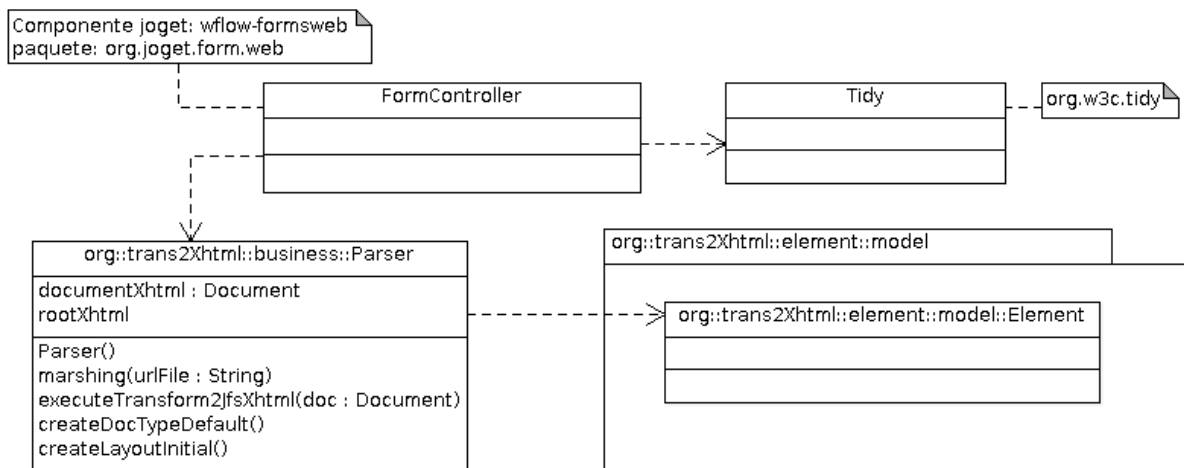
Diagrama de Clases: Paquete: org.trans.2Xhtml.element.model



14. Ilustración: Diagrama de Clases: Paquete org.trans2Xhtml.element.model

Diagrama de Clases por Casos de Uso y/o Componentes

Diagrama de Clases por componente parser2XHTML y Caso de Uso "generar código xhtml"



15. Ilustración: Diagrama de Clases por CU "generar código xhtml" - componente parse2XHTML

Especificaciones de Desarrollo

GNU/Linux

Sistema operativo libre con gran estabilidad, con acceso al código fuente, con independencia del proveedor, rápido al momento de incorporar los nuevos adelantos tecnológicos, con una comunidad activa de desarrollo que tienen a su alrededor, con abundancia de documentación relativa a los procedimientos .

Servidor de Aplicaciones Web

Es un software que permite alojar aplicaciones web bajo la plataforma Java, dichos servidores que se utilizarán deben ser multi plataforma, que soporte su personalización, tecnología para Entornos Empresariales (EE) y populares para las aplicaciones java, entre ellos podemos contar con: Apache-Tomcat 6.0.18+ para alojar WformDesigner (personalización de Joget) debido a que es un requerimiento de Joget, y puede alojar otras aplicaciones web, también se puede utilizar Jboss Server Application 5.1.0 GA para alojar las otras aplicaciones web e inclusive WformDesigner. En conclusión WformDesigner se utilizará con Apache-Tomcat.

MySQL

Sistema de gestión de base de datos relacional, multihilo y multiusuario, la que se ofrece

bajo la licencia GNU GPL, el cual funciona sobre multiplataformas

Java Developer Kit 1.60.22

Lenguaje de programación orientado a objetos creado por Sun Microsystems Inc, independiente de la plataforma, potente, seguro para el trabajo en red y su facilidad de uso.

Eclipse Helios

Es un Entorno de Desarrollo Integrado (IDE), la cual es una herramienta pensada para escribir, compilar, depurar y ejecutar programas, Está escrito en java pero sirve para otros lenguajes de programación. Las ventajas importantes es que existen diferentes plugins que le agregan gran funcionalidad a este IDE.

Apache Maven

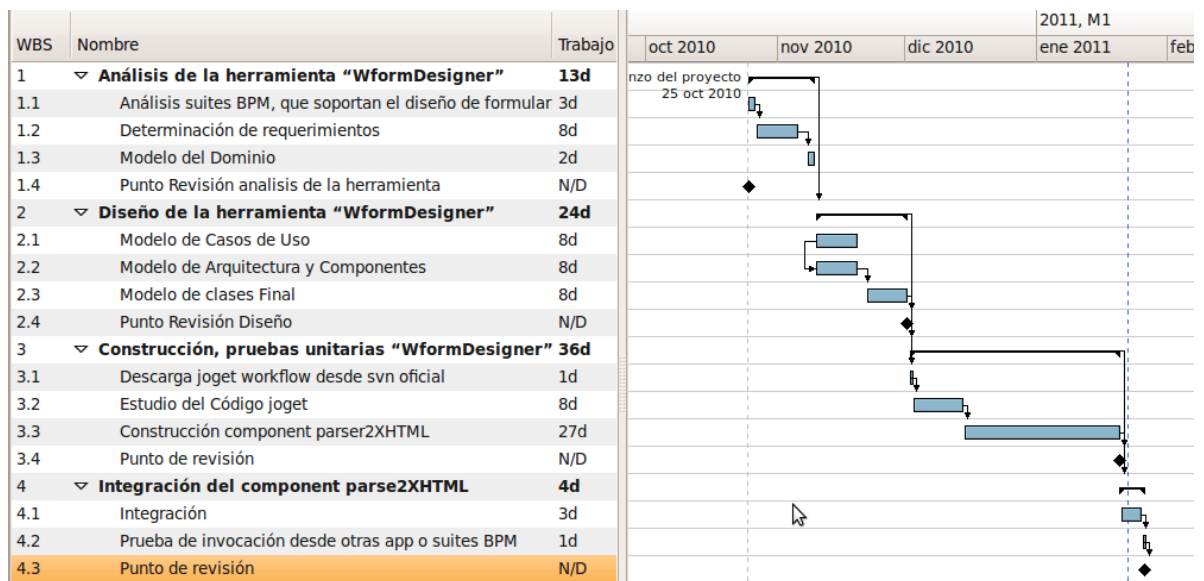
Apache Maven es una herramienta para la gestión de proyectos de software y a su compresión, basado en el concepto de objetos de proyecto (POM), con maven puede administrar un proyecto de construcción, información y documentación en una pieza centralizada.⁹

⁹ Welcome Apache Maven, Apache Maven Project, <http://maven.apache.org/>, [2011-01-10]

Implementación y Pruebas

Planificación de actividades para el desarrollo

A continuación se indica la planificación de las etapas con sus respectivas tareas, visualizado tanto en barras como en fechas tentativas.



16. Ilustración: Diagrama de Gantt de la Planificación

WBS	Nombre	Inicio	Fin	Trabajo	Duración
1	▼ Análisis de la herramienta "WformDesigner"	oct 25	nov 9	13d	12d
1.1	Análisis suites BPM, que soportan el diseño de formularios	oct 25	oct 26	3d	2d
1.2	Determinación de requerimientos	oct 27	nov 5	8d	8d
1.3	Modelo del Dominio	nov 8	nov 9	2d	2d
1.4	Punto Revisión análisis de la herramienta	oct 25	oct 25	N/D	N/D
2	▼ Diseño de la herramienta "WformDesigner"	nov 10	dic 1	24d	16d
2.1	Modelo de Casos de Uso	nov 10	nov 19	8d	8d
2.2	Modelo de Arquitectura y Componentes	nov 10	nov 19	8d	8d
2.3	Modelo de clases Final	nov 22	dic 1	8d	8d
2.4	Punto Revisión Diseño	dic 1	dic 1	N/D	N/D
3	▼ Construcción, pruebas unitarias "WformDesigner"	dic 2	ene 20	36d	36d
3.1	Descarga joget workflow desde svn oficial	dic 2	dic 2	1d	1d
3.2	Estudio del Código joget	dic 3	dic 14	8d	8d
3.3	Construcción component parser2XHTML	dic 15	ene 20	27d	27d
3.4	Punto de revisión	ene 20	ene 20	N/D	N/D
4	▼ Integración del component parse2XHTML	ene 21	ene 26	4d	4d
4.1	Integración	ene 21	ene 25	3d	3d
4.2	Prueba de invocación desde otras app o suites BPM	ene 26	ene 26	1d	1d
4.3	Punto de revisión	ene 26	ene 26	N/D	N/D

17. Ilustración: Distribución de la Planificación en tiempos

Cabe destacar que la planificación que se visualiza es en días, pero se tomará en cuenta que para cada día se trabajará 2 a 3 horas aproximadamente para un desarrollador, para lo cual tenemos el siguiente resumen en horas-hombre por etapas:

Etapa	Días	Horas-Hombre / día	Total Horas-hombre
Análisis de la Herramienta	13	2	26
Diseño de la Herramienta	24	2	48
Construcción, pruebas unitarias	36	2.5	90
Integración	4	3	12
Total	77		176

1. Tabla: Resumen de Etapas con tiempo en horas-hombre por día

Presupuesto

El presupuesto se lo puede obtener en base a la planificación de acuerdo al número de horas-hombre de trabajo y al material necesario para desarrollar el proyecto.

TALENTO HUMANO				
Etapa	Horas-hombre	Talento Humano	Costo/hora (USD)	Precio Total Costo/Hora (USD)
Análisis de la Herramienta	26	1	20,00	520,00
Diseño de la Herramienta	48	1	20,00	960,00
Construcción, pruebas unitarias	90	1	25,00	2.250,00
Integración	12	1	25,00	300,00
Total Talento Humano	180			4.030,00
OTROS COSTOS				
Descripción				Precio Total (USD)
Servicio Internet para Investigación				105, 00
Energía Eléctrica				100,00
Materiales de Oficina				100,00
Imprevistos				400,00
Total Otros Costos				705,00
COSTO TOTAL DEL PROYECTO				4.735,00

2. Tabla: Presupuesto del proyecto

Para el desarrollo de la herramienta “WformDesigner” se necesita de un presupuesto aproximado de 4.735,00 USD.

Desarrollo de Actividades de Implementación

Para la codificación de la personalización de joget workflow y obtener nuestra herramienta WformDesigner, se debe realizar las siguientes tareas:

1. Instalar JDK 1.6.0.22 y configurar las variables de entorno como: JAVA_HOME, PATH.
2. Instalar el servidor de aplicaciones Apache-Tomcat 6.0.26
3. Instalar el servidor de SGBD MySQL 5.0 o superior.
4. Instalar Eclipse Helios
5. Instalar los plugins de Maven (m2eclipse) y subversive (SVN) para eclipse Helios desde el sitio eclipse marketplace
6. Descargar los source de joget desde el repositorio SVN oficial (<https://github.com/jogetworkflow/jogetworkflow>)
7. Descomprimir todos los componentes de joget y ubicarlos en un workspace para eclipse.
8. Importar todos los proyectos para eclipse desde el workspace de trabajo, automáticamente serán reconocidos como proyectos de eclipse y maven.
9. Utilizar la herramienta de maven clean y maven install de eclipse para comenzar a

solucionar los problemas de dependencia.

10. Crear una base de datos llamada **wflowdb** en MySQL.
11. Una vez solucionados todos estos problemas copiar el archivos wflow-wfweb.war al servidor Apache-tomcat para probar su funcionamiento.
12. Las tablas de la Base de Datos se crean automáticamente al momento de iniciar por primera vez la aplicación.
13. Insertar un usuario administrador en la tabla **dir_user** de la Base de Datos **wflowdb**.
14. Construir el parser2xhtml respectivo.
15. Realizar la codificación y personalización respectiva en el componente de joget llamado **wflow-formsweb** para el llamado respectivo del parser.
16. Realizar las pruebas necesarias de generación de código xhtml para componentes de JSF y RichFaces en aplicaciones Seam.

Pruebas

Tomando en cuenta el plan de pruebas del diseño y algunas adicionales, tenemos las siguientes:

Pruebas de Funcionamiento

PRUEBA	Nro. test	Error
Recuperación de formularios diseñados previamente almacenados	20	0%
Generación de código fuente de los formularios al momento de guardar	15	5.00%
Visualización de la información y componentes	15	7.00%
Obtención de cada uno de la información que se deberán mostrar al usuario,	15	7.00%
Parser de todos los componentes del formulario diseñado a JSF y/o RichFaces	15	5.00%
Llamada a la aplicación WformDesigner desde otras aplicaciones web y/o herramienta de BPM en java	10	0.00%
Internacionalización del lenguaje de la Aplicación	10	15.00%

Pruebas de Implantación

PRUEBA	Nro. test	Error
Deploy de la aplicación en el servidor Apache-Tomcat	20	0%
Conexión a la Base de Datos desde la aplicación	20	0.00%
Almacenamiento físico en disco de los archivos generados en formato xhtml	15	3.00%

Pruebas de Aceptación

PRUEBA	Nro. test	Error
Archivos generados en .xhtml se acoplan a las plantillas de diseño por aplicaciones de java generadas por su wizard	20	3.00%
Archivos generados en .xhtml se visualizan en un proyecto de seam a través de un IDE como eclipse	20	3.00%
Archivos generados en .xhtml se visualizan en un proyecto de seam a través de un browser de internet	20	3.00%

Revisando las pruebas ejecutadas, se puede decir que la herramienta WformDesigner tiene un 90 – 95 % de funcionalidad y aceptación

Formación de usuarios

Como la aplicación esta dirigida principalmente para desarrolladores de software en aplicaciones web para java, no se necesitaría de una formación, basta con los conocimientos que ellos tenga sobre la tecnología Java con JSF, RichFaces, Seam y JBPM, además cuenta con una interfaz sumamente amigable y obvia para realizar las cosas.

Experiencias y Resultados

Conclusiones

Con el presente proyecto se llegó a obtener las siguientes conclusiones:

- Existen alternativas de software que se pueden tomar como base e idea para el desarrollo de la herramienta “WformDesigner” como proyecto, tal es el caso de joget workflow.
- La idea principal de la herramienta WformDesigner consiste en ser un componente adicional e independiente que facilite el diseño visual de formularios web para IGU bajo la especificación de Java Server Faces (JSF) con la implementación de Jboss RichFaces para aplicaciones web de Seam, o para suites BPM de JBPM.
- WformDesigner se pudo enlazar con la suite BPM de “Oficina sin Papeles” basada en jBPM para el Gobierno Provincial de Loja, para que los usuarios con rol de desarrollador generen sus vistas.
- El presente proyecto consiste en la primera versión de este tipo de herramientas en la generación de código XHTML en JSF para suites basadas en Jboss Business Process Management (jBPM) con Jboss Seam framework.
- Establecer una correcta metodología, planificación y presupuesto otorga una idea clara hacia donde debe enfocarse la herramienta, tratando de asegurar el éxito del proyecto y software de calidad, indicando que pasos deben cumplirse en sus tiempos y el costo de cada uno de ellos.
- Las diferentes comunidades de Software Libre y OpenSource siempre están a la vanguardia de la tecnología de software ya sea, innovando, mejorando, colaborando con las diferentes soluciones, con la finalidad de generar conocimiento y dando independencia en decidir cual es su mejor opción para su trabajo y no atarse a un solo tipo de software.
- El Análisis entre las diferentes herramientas que existen en el mercado, estableciendo pros y contras entre su tipo de tecnología que utilizan, arquitectura, frameworks y grado de conocimiento ayudan a tener claro lo que se pueden hacer o no con cada una de ellas para su personalización y/o mejoramiento.
- El uso de Software libre ayudan en gran medida al desarrollador de software en sus aplicaciones, ya que no necesariamente tiene que comenzar desde cero, sino puede reutilizar código funcional mantenido por comunidades para su aplicación y adaptarlo de acuerdo a sus necesidades.
- La integración de diferentes aplicaciones con diferentes tecnologías de software es

complicada, ya que se debe realizar un estudio de cada una de ellas, además se debe tener un alto grado de conocimientos en arquitectura de software para obtener una verdadera sinergia de las aplicaciones involucradas.

- Joget WorkFlow resultó complicado al momento de generar los formularios debido a que los componentes complejos como editores WYSIWYG y subidores de archivos no genera código en sus fuentes, sino que asigna una marca reconocidos solo por la aplicación que internamente al momento de ejecutar una vista del formulario en su suite asigna el código necesario para que lo interprete el browser.
- Joget workFlow es un proyecto totalmente integrado y dependiente entre cada uno de sus componentes por lo que en primera instancia y por falta de tiempo se dificulto en separar sólo el componente de construcción de formularios, lo que se optó por desactivar en la vista los componentes adicionales que no son necesarios para la herramienta WformDesigner.
- Para realizar un parser que permita transformar componentes de un formato a otro se debe tener en cuenta el conocimiento completo cual es su inicial estructura y que reglas deben cumplir en la estructura de salida.

Recomendaciones

- Tener siempre presente en cuenta, los requerimientos del sistema, para evitar confusiones en lo que se pretende desarrollar, pérdidas de tiempo en cosas intrascendentes y distracciones innecesarias por cosas irrelevante para el proyecto.
- Tener presente la configuración de la herramienta para poder obtener la ruta de almacenamiento de los archivos generados.
- Tener un IDE (Eclipse) con los plugins necesarios para cada proyecto, ya que en gran medida aceleran el desarrollo de aplicaciones.
- Utilizar la herramienta WformDesigner para generar formularios para aplicaciones web que implemente Jboss Seam Framework.
- El uso de componentes software libre ya que brindan una gran ayuda para la programación, debido a que, por lo general existen cosas ya hechas para nuestros propósitos.
- Planificar, estimar y gestionar correctamente los proyectos de software para llevar una velocidad de desarrollo constante, para cumplir a cabalidad los requerimientos iniciales.
- Tener presente la tecnología y arquitectura de los componentes de software libre a utilizar, para sacarle el máximo provecho al mismo, y perder lo menos tiempo posible en su funcionamiento.

Ampliaciones

Entre las principales ampliaciones que he observado luego de realizar el proyecto son las siguientes:

- Obtener solamente el módulo de diseño de formularios desde joget para que la aplicación sea mas liviana y pueda acoplarse como un componente adicional a cualquier aplicación web.
- Realizar un componente que permita obtener las variables directamente desde los procesos definidos en suites BPM's y/o otras aplicación web JEE.
- Implementar un componente para descarga de los archivos generados en xhtml y permitan ser guardados en cualquier parte de una aplicación.
- Internacionalización de Lenguaje completa para la aplicación.
- Implantación de gestor de temas visuales y un wizard para una fácil configuración.

Anexos

Anexo A: Licencia de Documentación Libre de GNU

Versión 1.2, Noviembre 2002

Ésta es una traducción no oficial de la GNU Free Document License a Español (Castellano). No ha sido publicada por la Free Software Foundation y no establece legalmente los términos de distribución para trabajos que usen la GFDL (sólo el texto de la versión original en Inglés de la GFDL lo hace). Sin embargo, esperamos que esta traducción ayude los hispanohablantes a entender mejor la GFDL. La versión original de la GFDL esta disponible en la Free Software Foundation.

Esta traducción está basada en una de la versión 1.1 de Igor Támara y Pablo Reyes. Sin embargo la responsabilidad de su interpretación es de Joaquín Seoane.

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. Se permite la copia y distribución de copias literales de este documento de licencia, pero no se permiten cambios[1].

A.1. PREÁMBULO

El propósito de esta Licencia es permitir que un manual, libro de texto, u otro documento escrito sea libre en el sentido de libertad: asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, de manera comercial o no. En segundo término, esta Licencia proporciona al autor y al editor[2] una manera de obtener reconocimiento por su trabajo, sin que se le considere responsable de las modificaciones realizadas por otros.

Esta Licencia es de tipo copyleft, lo que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Complementa la Licencia Pública General de GNU, que es una licencia tipo copyleft diseñada para el software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: un programa libre debe venir con manuales que ofrezcan la mismas libertades que el software. Pero esta licencia no se limita a manuales de software; puede usarse para cualquier texto, sin tener en cuenta su temática o si se publica como libro impreso o no.

Recomendamos esta licencia principalmente para trabajos cuyo fin sea instructivo o de referencia.

A.2. APLICABILIDAD Y DEFINICIONES

Esta Licencia se aplica a cualquier manual u otro trabajo, en cualquier soporte, que contenga una nota del propietario de los derechos de autor que indique que puede ser distribuido bajo los términos de esta Licencia. Tal nota garantiza en cualquier lugar del mundo, sin pago de derechos y sin límite de tiempo, el uso de dicho trabajo según las condiciones aquí estipuladas. En adelante la palabra Documento se referirá a cualquiera de dichos manuales o trabajos. Cualquier persona es un licenciataria y será referido como Usted. Usted acepta la licencia si copia, modifica o distribuye el trabajo de cualquier modo que requiera permiso según la ley de propiedad intelectual.

Una Versión Modificada del Documento significa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma.

Una Sección Secundaria es un apéndice con título o una sección preliminar del Documento que trata exclusivamente de la relación entre los autores o editores y el tema general del Documento (o temas

relacionados) pero que no contiene nada que entre directamente en dicho tema general (por ejemplo, si el Documento es en parte un texto de matemáticas, una Sección Secundaria puede no explicar nada de matemáticas). La relación puede ser una conexión histórica con el tema o temas relacionados, o una opinión legal, comercial, filosófica, ética o política acerca de ellos.

Las Secciones Invariantes son ciertas Secciones Secundarias cuyos títulos son designados como Secciones Invariantes en la nota que indica que el documento es liberado bajo esta Licencia. Si una sección no entra en la definición de Secundaria, no puede designarse como Invariante. El documento puede no tener Secciones Invariantes. Si el Documento no identifica las Secciones Invariantes, es que no las tiene.

Los Textos de Cubierta son ciertos pasajes cortos de texto que se listan como Textos de Cubierta Delantera o Textos de Cubierta Trasera en la nota que indica que el documento es liberado bajo esta Licencia. Un Texto de Cubierta Delantera puede tener como mucho 5 palabras, y uno de Cubierta Trasera puede tener hasta 25 palabras.

Una copia Transparente del Documento, significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público en general, apto para que los contenidos puedan ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por puntos) con programas genéricos de manipulación de imágenes o (para dibujos) con algún editor de dibujos ampliamente disponible, y que sea adecuado como entrada para formateadores de texto o para su traducción automática a formatos adecuados para formateadores de texto. Una copia hecha en un formato definido como Transparente, pero cuyo marcaje o ausencia de él haya sido diseñado para impedir o dificultar modificaciones posteriores por parte de los lectores no es Transparente. Un formato de imagen no es Transparente si se usa para una cantidad de texto sustancial. Una copia que no es Transparente se denomina Opaca.

Como ejemplos de formatos adecuados para copias Transparentes están ASCII puro sin marcaje, formato de entrada de Texinfo, formato de entrada de LaTeX, SGML o XML usando una DTD disponible públicamente, y HTML, PostScript o PDF simples, que sigan los estándares y diseñados para que los modifiquen personas. Ejemplos de formatos de imagen transparentes son PNG, XCF y JPG. Los formatos Opacos incluyen formatos propietarios que pueden ser leídos y editados únicamente en procesadores de palabras propietarios, SGML o XML para los cuáles las DTD y/o herramientas de procesamiento no estén ampliamente disponibles, y HTML, PostScript o PDF generados por algunos procesadores de palabras sólo como salida.

La Portada significa, en un libro impreso, la página de título, más las páginas siguientes que sean necesarias para mantener legiblemente el material que esta Licencia requiere en la portada. Para trabajos en formatos que no tienen página de portada como tal, Portada significa el texto cercano a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del texto. Una sección Titulada XYZ significa una parte del Documento cuyo título es precisamente XYZ o contiene XYZ entre paréntesis, a continuación de texto que traduce XYZ a otro idioma (aquí XYZ se refiere a nombres de sección específicos mencionados más abajo, como Agradecimientos, Dedicatorias, Aprobaciones o Historia. Conservar el Título de tal sección cuando se modifica el Documento significa que permanece una sección Titulada XYZ según esta definición[3].

El Documento puede incluir Limitaciones de Garantía cercanas a la nota donde se declara que al Documento se le aplica esta Licencia. Se considera que estas Limitaciones de Garantía están incluidas, por referencia, en la Licencia, pero sólo en cuanto a limitaciones de garantía: cualquier otra implicación que estas Limitaciones de Garantía puedan tener es nula y no tiene efecto en el significado de esta Licencia.

A.3. COPIA LITERAL

Usted puede copiar y distribuir el Documento en cualquier soporte, sea en forma comercial o no,

siempre y cuando esta Licencia, las notas de copyright y la nota que indica que esta Licencia se aplica al Documento se reproduzcan en todas las copias y que usted no añada ninguna otra condición a las expuestas en esta Licencia. Usted no puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3.

Usted también puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

A.4. COPIADO EN CANTIDAD

Si publica copias impresas del Documento (o copias en soportes que tengan normalmente cubiertas impresas) que sobrepasen las 100, y la nota de licencia del Documento exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible todos esos Textos de Cubierta: Textos de Cubierta Delantera en la cubierta delantera y Textos de Cubierta Trasera en la cubierta trasera. Ambas cubiertas deben identificarlo a Usted clara y legiblemente como editor de tales copias. La cubierta debe mostrar el título completo con todas las palabras igualmente prominentes y visibles. Además puede añadir otro material en las cubiertas. Las copias con cambios limitados a las cubiertas, siempre que conserven el título del Documento y satisfagan estas condiciones, pueden considerarse como copias literales.

Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros (tantos como sea razonable colocar) en la verdadera cubierta y situar el resto en páginas adyacentes.

Si Usted publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente, que pueda ser leída por una máquina, con cada copia Opaca, o bien mostrar, en cada copia Opaca, una dirección de red donde cualquier usuario de la misma tenga acceso por medio de protocolos públicos y estandarizados a una copia Transparente del Documento completa, sin material adicional. Si usted hace uso de la última opción, deberá tomar las medidas necesarias, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio establecido por lo menos un año después de la última vez que distribuya una copia Opaca de esa edición al público (directamente o a través de sus agentes o distribuidores).

Se solicita, aunque no es requisito, que se ponga en contacto con los autores del Documento antes de redistribuir gran número de copias, para darles la oportunidad de que le proporcionen una versión actualizada del Documento.

A.5. MODIFICACIONES

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada haciendo el rol del Documento, por lo tanto dando licencia de distribución y modificación de la Versión Modificada a quienquiera posea una copia de la misma.

Además, debe hacer lo siguiente en la Versión Modificada:

- A. Usar en la Portada (y en las cubiertas, si hay alguna) un título distinto al del Documento de sus versiones anteriores (que deberían, si hay alguna, estar listadas en la sección de Historia del Documento). Puede usar el mismo título de versiones anteriores al original siempre y cuando quien las publicó originalmente otorgue permiso.
- B. Listar en la Portada, como autores, una o más personas o entidades responsables de la autoría de las modificaciones de la Versión Modificada, junto con por lo menos cinco de los autores

principales del Documento (todos sus autores principales, si hay menos de cinco), a menos que le eximan de tal requisito.

- C. Mostrar en la Portada como editor el nombre del editor de la Versión Modificada.
- D. Conservar todas las notas de copyright del Documento.
- E. Añadir una nota de copyright apropiada a sus modificaciones, adyacente a las otras notas de copyright.
- F. Incluir, inmediatamente después de las notas de copyright, una nota de licencia dando el permiso para usar la Versión Modificada bajo los términos de esta Licencia, como se muestra en la Adenda al final de este documento.
- G. Conservar en esa nota de licencia el listado completo de las Secciones Invariantes y de los Textos de Cubierta que sean requeridos en la nota de Licencia del Documento original.
- H. Incluir una copia sin modificación de esta Licencia.
- I. Conservar la sección Titulada Historia, conservar su Título y añadirle un elemento que declare al menos el título, el año, los nuevos autores y el editor de la Versión Modificada, tal como figuran en la Portada. Si no hay una sección Titulada Historia en el Documento, crear una estableciendo el título, el año, los autores y el editor del Documento, tal como figuran en su Portada, añadiendo además un elemento describiendo la Versión Modificada, como se estableció en la oración anterior.
- J. Conservar la dirección en red, si la hay, dada en el Documento para el acceso público a una copia Transparente del mismo, así como las otras direcciones de red dadas en el Documento para versiones anteriores en las que estuviese basado. Pueden ubicarse en la sección Historia. Se puede omitir la ubicación en red de un trabajo que haya sido publicado por lo menos cuatro años antes que el Documento mismo, o si el editor original de dicha versión da permiso.
- K. En cualquier sección Titulada Agradecimientos o Dedicatorias, Conservar el Título de la sección y conservar en ella toda la sustancia y el tono de los agradecimientos y/o dedicatorias incluidas por cada contribuyente.
- L. Conservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección.
- M. Borrar cualquier sección titulada Aprobaciones. Tales secciones no pueden estar incluidas en las Versiones Modificadas.
- N. No cambiar el título de ninguna sección existente a Aprobaciones ni a uno que entre en conflicto con el de alguna Sección Invariante.
- O. Conservar todas las Limitaciones de Garantía.

Si la Versión Modificada incluye secciones o apéndices nuevos que califiquen como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, añada sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección.

Puede añadir una sección titulada Aprobaciones, siempre que contenga únicamente aprobaciones de su Versión Modificada por otras fuentes --por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como la definición oficial de un estándar.

Puede añadir un pasaje de hasta cinco palabras como Texto de Cubierta Delantera y un pasaje de hasta 25 palabras como Texto de Cubierta Trasera en la Versión Modificada. Una entidad solo puede añadir (o hacer que se añada) un pasaje al Texto de Cubierta Delantera y uno al de Cubierta Trasera.

Si el Documento ya incluye un textos de cubiertas añadidos previamente por usted o por la misma entidad que usted representa, usted no puede añadir otro; pero puede reemplazar el anterior, con permiso explícito del editor que agregó el texto anterior.

Con esta Licencia ni los autores ni los editores del Documento dan permiso para usar sus nombres para publicidad ni para asegurar o implicar aprobación de cualquier Versión Modificada.

A.6. COMBINACIÓN DE DOCUMENTOS

Usted puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, listadas todas como Secciones Invariantes del trabajo combinado en su nota de licencia. Así mismo debe incluir la Limitación de Garantía.

El trabajo combinado necesita contener solamente una copia de esta Licencia, y puede reemplazar varias Secciones Invariantes idénticas por una sola copia. Si hay varias Secciones Invariantes con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único añadiéndole al final del mismo, entre paréntesis, el nombre del autor o editor original de esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones Invariantes de la nota de licencia del trabajo combinado.

En la combinación, debe combinar cualquier sección Titulada Historia de los documentos originales, formando una sección Titulada Historia; de la misma forma combine cualquier sección Titulada Agradecimientos, y cualquier sección Titulada Dedicatorias. Debe borrar todas las secciones tituladas Aprobaciones.

A.7. COLECCIONES DE DOCUMENTOS

Puede hacer una colección que conste del Documento y de otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en todos los documentos por una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para cada copia literal de cada uno de los documentos en cualquiera de los demás aspectos.

Puede extraer un solo documento de una de tales colecciones y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los demás aspectos relativos a la copia literal de dicho documento.

A.8. AGREGACIÓN CON TRABAJOS INDEPENDIENTES

Una recopilación que conste del Documento o sus derivados y de otros documentos o trabajos separados e independientes, en cualquier soporte de almacenamiento o distribución, se denomina un agregado si el copyright resultante de la compilación no se usa para limitar los derechos de los usuarios de la misma más allá de lo que los de los trabajos individuales permiten. Cuando el Documento se incluye en un agregado, esta Licencia no se aplica a otros trabajos del agregado que no sean en sí mismos derivados del Documento.

Si el requisito de la sección 3 sobre el Texto de Cubierta es aplicable a estas copias del Documento y el Documento es menor que la mitad del agregado entero, los Textos de Cubierta del Documento pueden colocarse en cubiertas que enmarquen solamente el Documento dentro del agregado, o el equivalente electrónico de las cubiertas si el documento está en forma electrónica. En caso contrario deben aparecer en cubiertas impresas enmarcando todo el agregado.

A.9. TRADUCCIÓN

La Traducción es considerada como un tipo de modificación, por lo que usted puede distribuir traducciones del Documento bajo los términos de la sección 4. El reemplazo las Secciones Invariantes con traducciones requiere permiso especial de los dueños de derecho de autor, pero usted puede añadir traducciones de algunas o todas las Secciones Invariantes a las versiones originales de las mismas. Puede incluir una traducción de esta Licencia, de todas las notas de licencia del documento, así como de las Limitaciones de Garantía, siempre que incluya también la versión en Inglés de esta Licencia y las versiones originales de las notas de licencia y Limitaciones de Garantía.

En caso de desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la nota de licencia o la limitación de garantía, la versión original en Inglés prevalecerá.

Si una sección del Documento está Titulada Agradecimientos, Dedicatorias o Historia el requisito (sección 4) de Conservar su Título (Sección 1) requerirá, típicamente, cambiar su título.

A.10. TERMINACIÓN

Usted no puede copiar, modificar, sublicenciar o distribuir el Documento salvo por lo permitido expresamente por esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y dará por terminados automáticamente sus derechos bajo esa Licencia. Sin embargo, los terceros que hayan recibido copias, o derechos, de usted bajo esta Licencia no verán terminadas sus licencias, siempre que permanezcan en total conformidad con ella.

A.11. REVISIONES FUTURAS DE ESTA LICENCIA

De vez en cuando la Free Software Foundation puede publicar versiones nuevas y revisadas de la Licencia de Documentación Libre GNU. Tales versiones nuevas serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar nuevos problemas o intereses. Vea <http://www.gnu.org/copyleft/>.

Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que se aplica una versión numerada en particular de esta licencia o cualquier versión posterior, usted tiene la opción de seguir los términos y condiciones de la versión especificada o cualquiera posterior que haya sido publicada (no como borrador) por la Free Software Foundation.

Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada (no como borrador) por la Free Software Foundation.

A.12. ADENDA: Cómo usar esta Licencia en sus documentos

Para usar esta licencia en un documento que usted haya escrito, incluya una copia de la Licencia en el documento y ponga el siguiente copyright y nota de licencia justo después de la página de título:

Copyright (c) AÑO SU NOMBRE. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada GNU Free Documentation License.

Si tiene Secciones Invariantes, Textos de Cubierta Delantera y Textos de Cubierta Trasera, reemplace la frase sin ... Trasera por esto: siendo las Secciones Invariantes LISTE SUS TÍTULOS, siendo los Textos de Cubierta Delantera LISTAR, y siendo sus Textos de Cubierta Trasera LISTAR.

Si tiene Secciones Invariantes sin Textos de Cubierta o cualquier otra combinación de los tres, mezcle ambas alternativas para adaptarse a la situación.

Si su documento contiene ejemplos de código de programa no triviales, recomendamos liberar estos ejemplos en paralelo bajo la licencia de software libre que usted elija, como la Licencia Pública General de GNU (GNU General Public License), para permitir su uso en software libre.

Notes

[1] Ésta es la traducción del Copyright de la Licencia, no es el Copyright de esta traducción no autorizada.

[2] La licencia original dice publisher, que es, estrictamente, quien publica, diferente de editor, que es más bien quien prepara un texto para publicar. En castellano editor se usa para ambas cosas.

[3] En sentido estricto esta licencia parece exigir que los títulos sean exactamente Acknowledgements, Dedications, Endorsements e History, en inglés.

Anexo B: Manual de Usuario

Ingreso a la aplicación

Para ingresar a la aplicación se debe indicar en el browser la ruta donde se encuentra el servidor, seguidamente con el nombre de la aplicación **/wflow-wfweb**. Así:

http://{NOMBRE_SERVIDOR_O_DIRECCION_IP:PUERTO}/wflow-wfweb

Para nuestro caso sería:

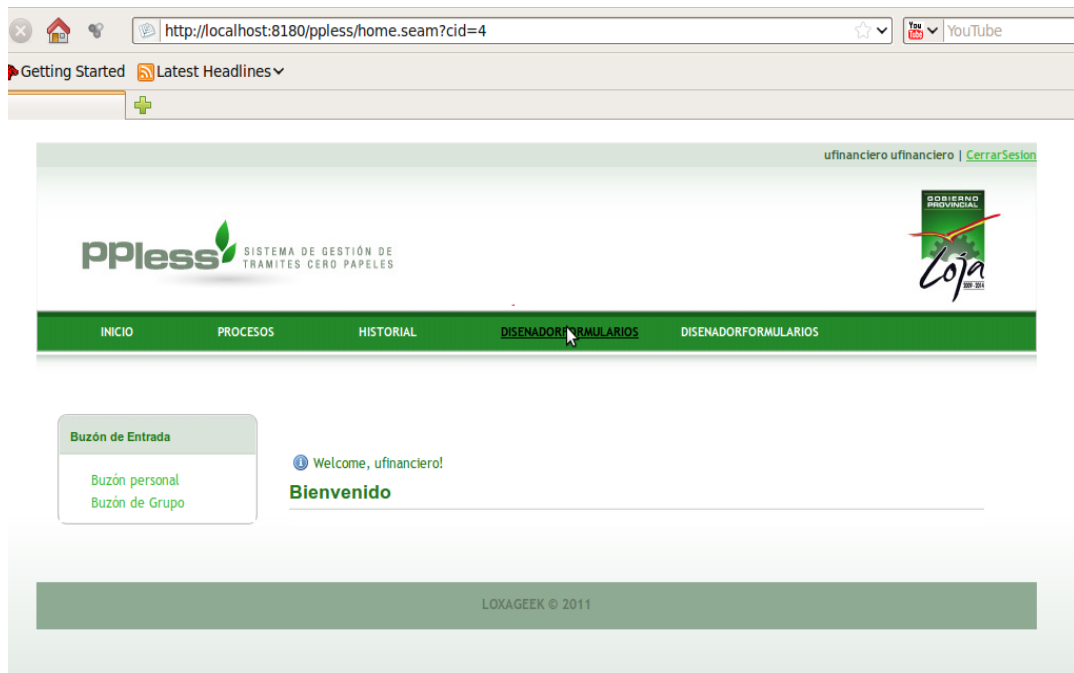
<http://xpsdell:8081/wflow-wfweb/>

Para enlazar desde una aplicación independiente a nuestra aplicación, específicamente desde una aplicación web con seam se debe crear un componente `<s:link>` o `<h:outputLink>` e indicarle la ruta específica. De esta manera:

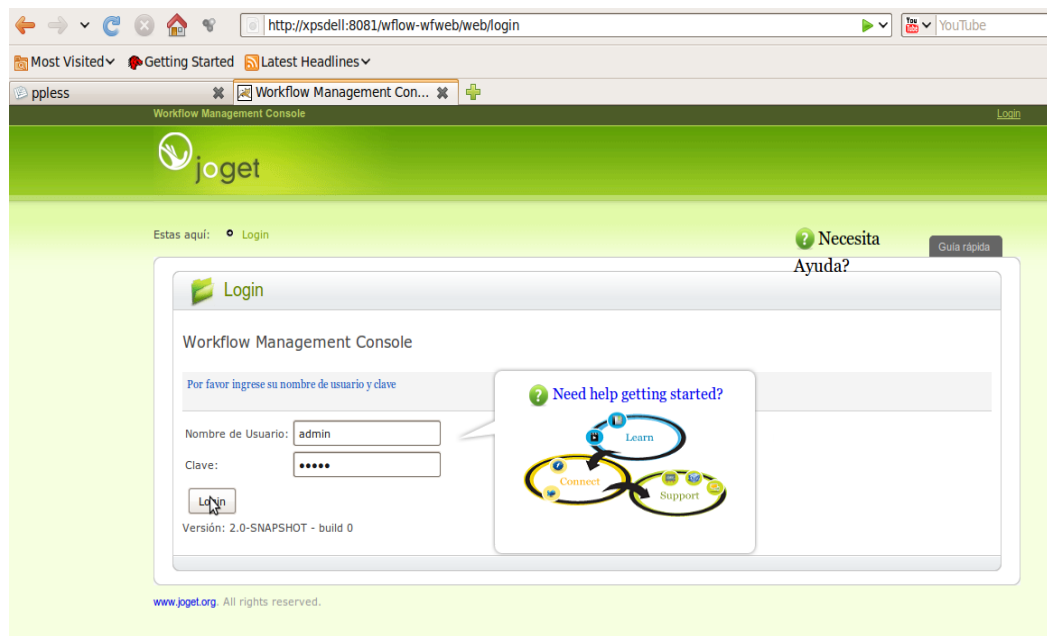
```
<s:link id="menuFormDesignerJogetId"
  onclick="javascript:window.open('http://xpsdell:8081/wflow-
wfweb/web/admin/form/category/list');return true;"
  propagation="none">
  <!-- f:param name="mode" value="history"/ -->
  <h:outputText value="#{messages['DisenadorFormularios']}" />
</s:link>
```

```
0
<h:outputLink value="http://xpsdell:8081/wflow-
wfweb/web/admin/form/category/list" target="_blank">
  <h:outputText value="#{messages['DisenadorFormularios']}" />
</h:outputLink>
```

Como ejemplo indicaremos la pantalla que hace el llamado desde la suite BPM de oficina sin papeles a la aplicación.



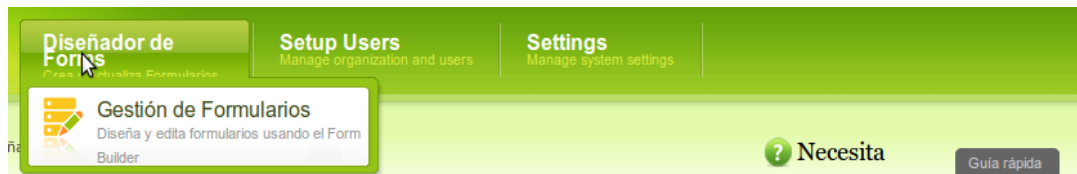
Luego de llamar a la aplicación lo primero que se nos solicita es el ingreso a través de un usuario autenticado.



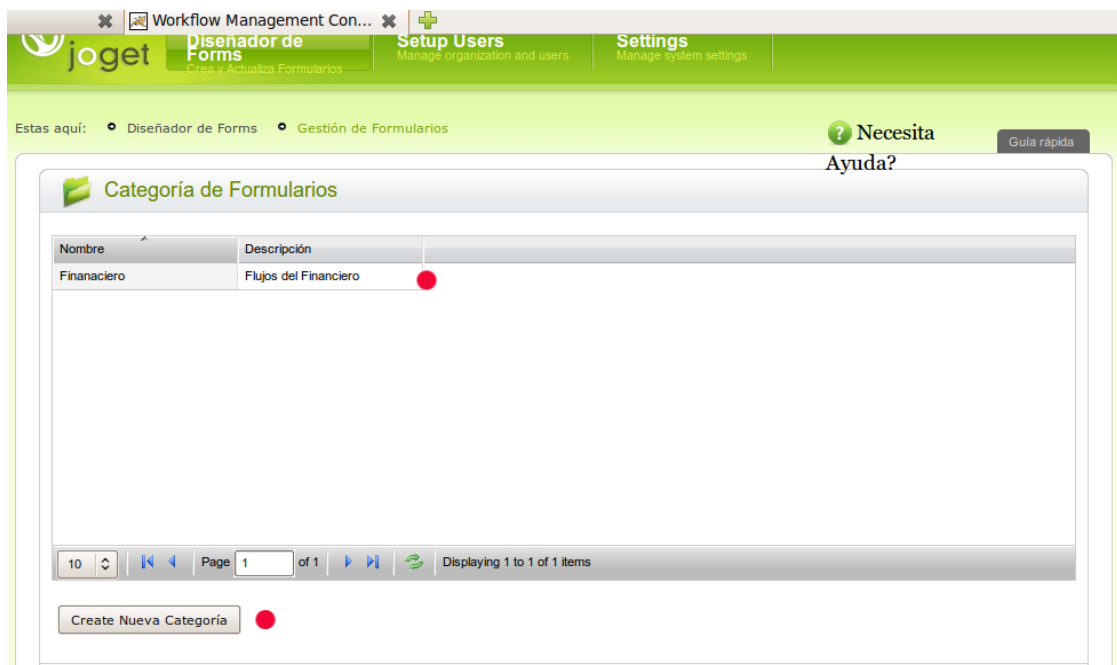
Una vez ingresado como un usuario autenticado con un rol de administrador, se podrá gestionar los formularios y/o usuarios.

Gestión de Formularios

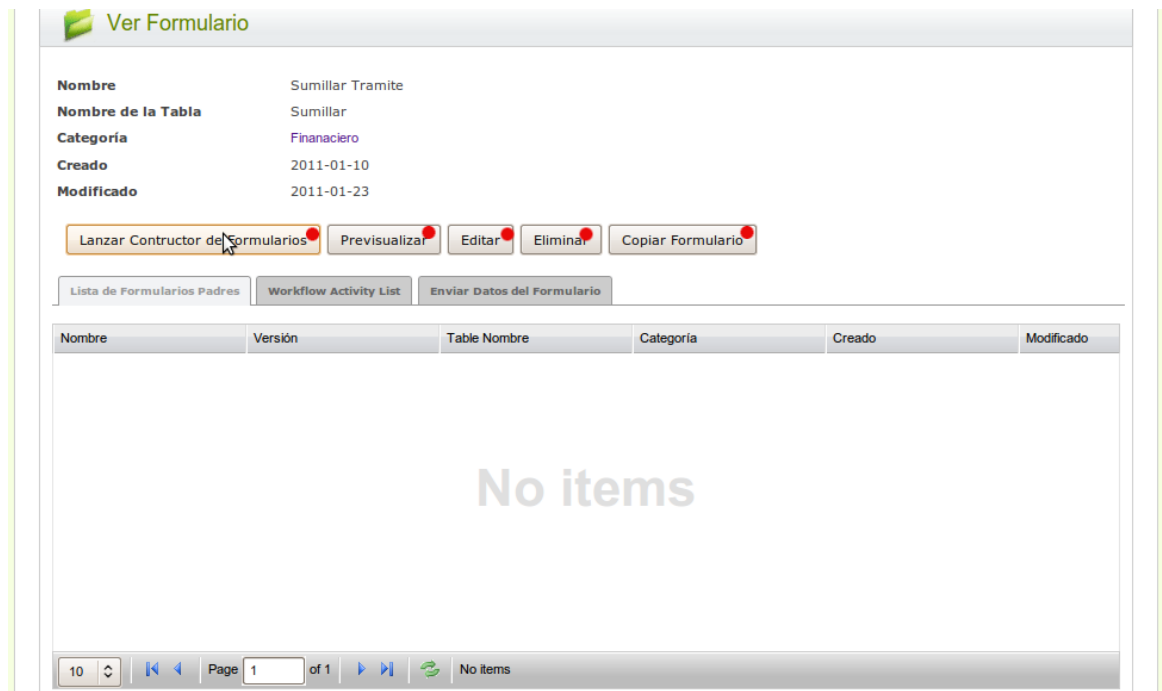
1. Elegir el menú Diseñador de Formularios / Gestión de Formularios.



2. Se mostrará un listado de categorías de formularios, donde podemos crear una nueva categorías de formulario eligiendo el botón **Crear Nueva Categoría** o a su vez elegir una categoría existente para ver sus formularios contenidos en ella.



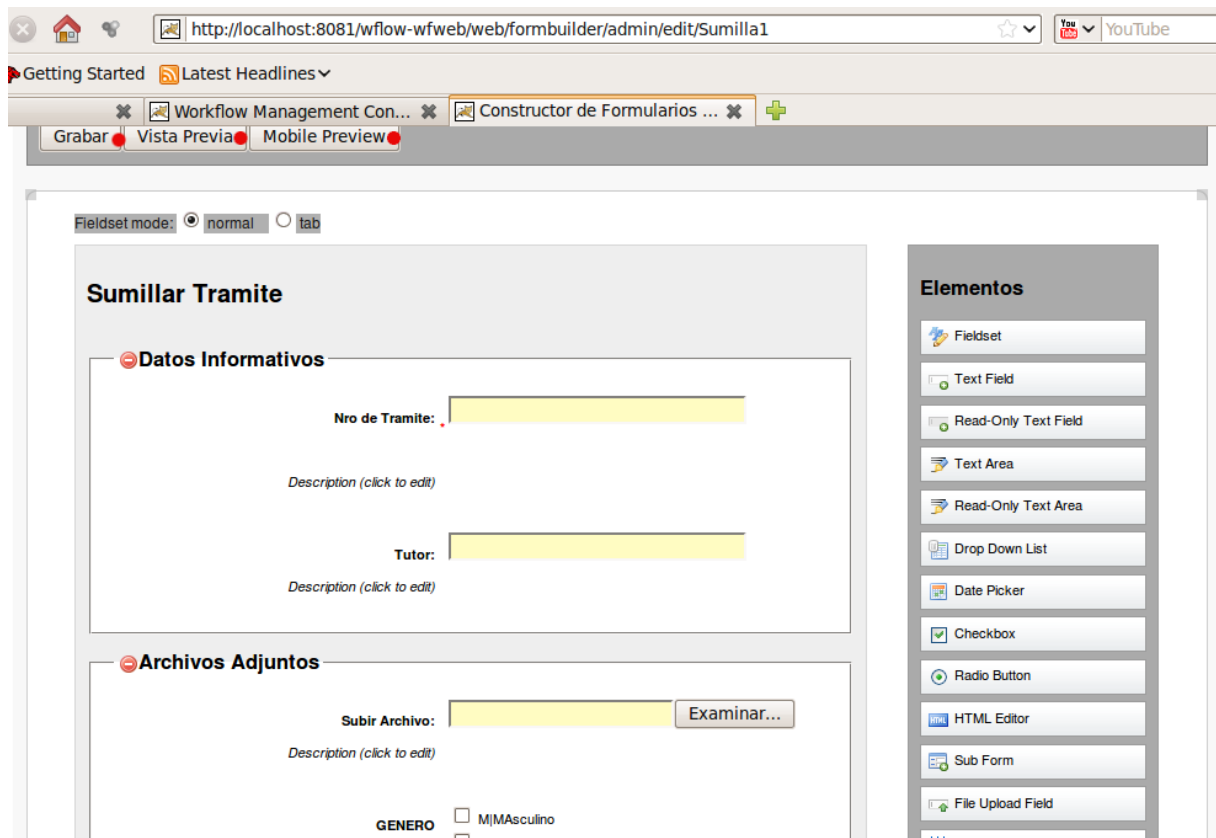
3. Eligiendo una categoría se la puede gestionar a ella: editándola para realizar las modificaciones eligiendo el botón Editar; Eliminar, el cual eliminará la categoría con todos los formularios en su interior con el botón Eliminar; agregar formularios a ala categoría con el botón **Crear nuevo Formulario**, eliminar un formulario seleccionado de la lista de formularios de la categoría a través del botón **Eliminar**; y editar un formulario eligiéndolo de la lista de formularios



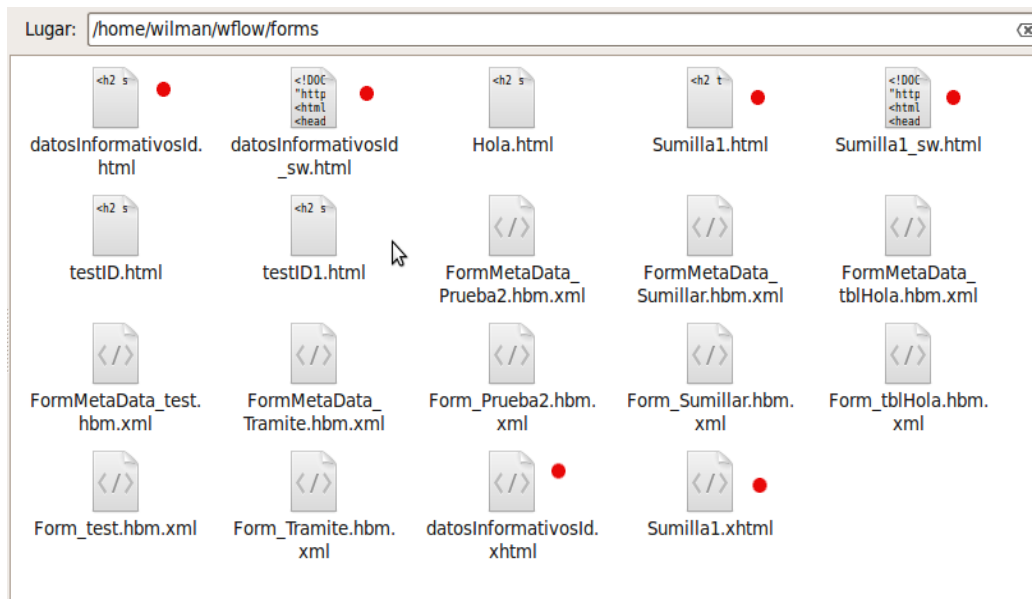
Diseñando Formularios y generando código xhtml

El Diseñador de Formularios nos permite agregar los diferentes componentes visuales a nuestros formularios, como también realizar la generación de código xhtml y JSF

1. Para lanzar el diseñador de formulario primeramente debimos elegir o crear el formulario, escoger el botón **Lanzar Constructor de Formulario** desde la vista **Ver Formulario**. Ahí se puede agregar, eliminar o modificar los componentes del formulario de acuerdo a nuestra conveniencia, fijar opciones requeridas o hacer referencia a variables desde nuestras aplicaciones ingresándolas como cadenas en las respectivas propiedades de los componentes.



2. Una vez diseñado nuestro formulario se lo procede a guardarlo, eligiendo el **botón Grabar** de la parte superior, con ello se guardará la información del formulario con sus componentes relacionados, como además se generará 2 archivos: uno con extensión xhtml para JSF de seam con el mismo identificador del formulario y el otro archivo con extensión .html con el identificador del formulario seguido de **sw** que contiene etiquetas html puras en el directorio de configuración de la aplicación en: **wflow/forms** (por lo general ubicado dentro del directorio de trabajo del usuario). A continuación se muestran los archivos generados en dicho directorio



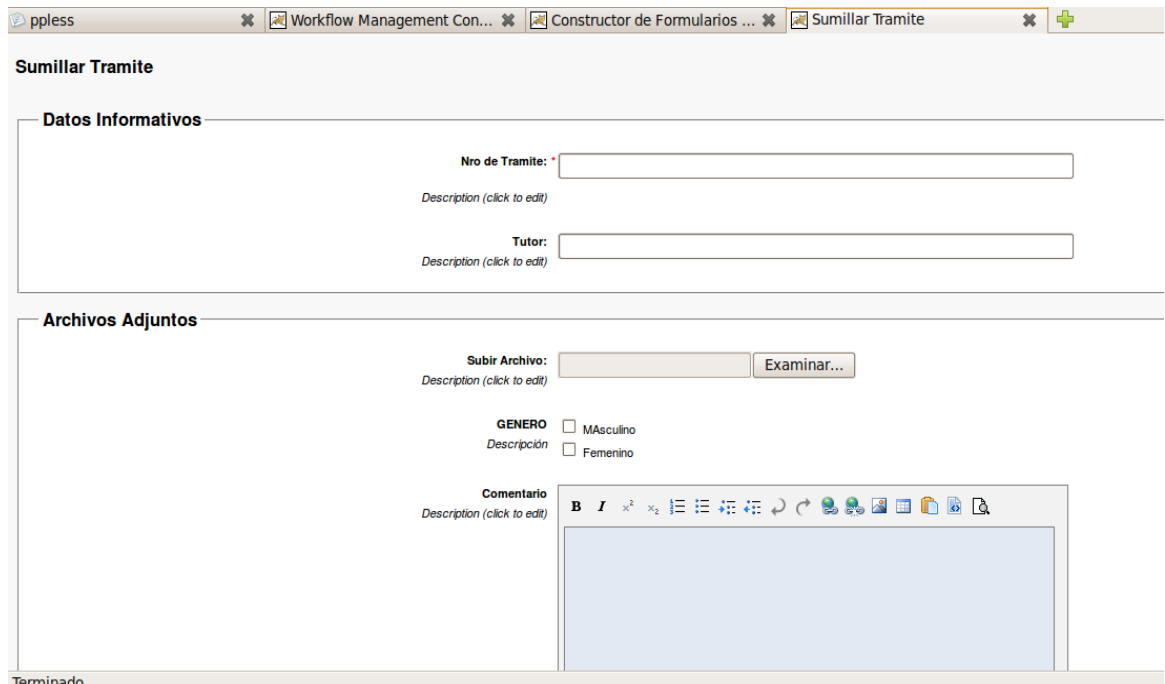
Desde un editor de texto para ver su generación

```
datosInformativosId.xhtml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

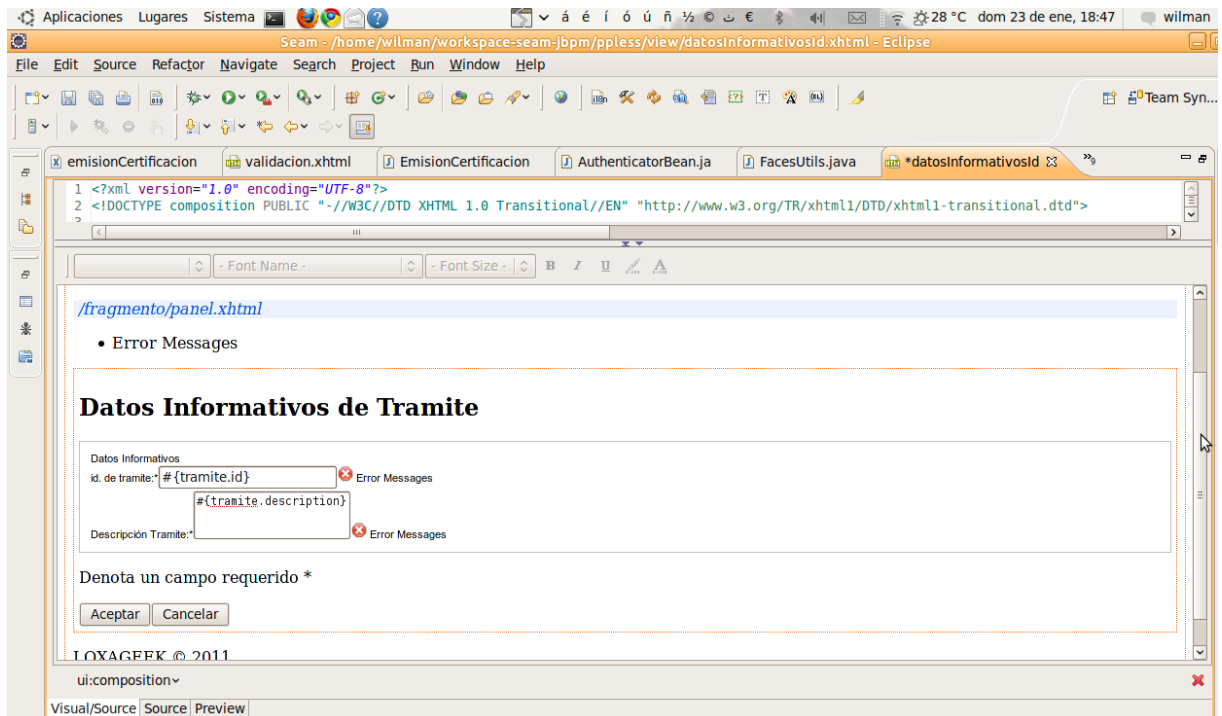
<ui:composition xmlns:ui="http://java.sun.com/jsf/facelets" xmlns:s="http://jboss.com/products/seam/taglib" xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html" xmlns:rich="http://richfaces.org/rich" xmlns:a="http://richfaces.org/a4j" template="/layout/template">
  <ui:define name="body">
    <h:form>
      <h2 xmlns="http://www.w3.org/1999/xhtml">Datos Informativos
de Tramite</h2>
      <div xmlns="http://www.w3.org/1999/xhtml" id="_ex_formbuilder_rows">
        <rich:panel>
          <f:facet name="header">Datos
Informativos</f:facet>
          <s:decorate template="/layout/edit.xhtml" id="ex_row_0">
            <ui:define name="label">id. de tramite:</ui:define>
            <h:inputText id="ex_text_0" value="#{tramite.id}" required="true" />
          </s:decorate>
          <s:decorate template="/layout/edit.xhtml" id="ex_row_1">
            <ui:define name="label">Descripción Tramite:</ui:define>
            <h:inputTextarea id="_ex_textarea_1" value="#{tramite.description}" required="true" />
          </s:decorate>
        </rich:panel>
      </div>
      <p xmlns="http://www.w3.org/1999/xhtml" class="formbuilder_required">
Denota un campo
requerido
      <span class="formbuilder_required_marker">*</span>
      <h:inputHidden id="ex_counter_rows" value="" />
      <h:inputHidden id="ex_counter_fieldset" value="" />
    </p>
      <div xmlns="http://www.w3.org/1999/xhtml" class="actionButtons">
        <h:commandButton action="" value="Aceptar" />
        <s:button view="" value="Cancelar" />
      </div>
    </h:form>
  </ui:define>
```

Previsualizando Formularios

A continuación se muestra la pantalla de vista previa de un formulario luego de haber elegido el botón **Previsualizar** ya sea desde la vista **Ver Formulario** o **Vista Previa** de la pantalla de diseñador de formularios



Una vez generados se pueden copiar esos archivos y llevar al IDE Eclipse que con su plugin de Jboos Tools permite tener una vista previa del formulario que se lo ha generado.



Ahora se puede visualizar el formulario generado desde una aplicación web, en este caso desde la suite BPM "Oficina sin papeles", adaptándose en su totalidad a los estilos y distribuciones establecidas por los facelets de Seam

http://localhost:8180/ppless/datosInformativosId.seam

Inicio de sesión

PPlless SISTEMA DE GESTIÓN DE TRAMITES CERO PAPELES

GOBIERNO PROVINCIAL Loja

INICIO PROCESOS HISTORIAL DISENADORFORMULARIOS DISENADORFORMULARIOS

Buzón de Entrada

Buzón personal
Buzón de Grupo

Datos Informativos de Tramite

Datos Informativos

id. de tramite: * Denota un campo requerido *

Descripción Tramite: *

Aceptar Cancelar

LOXAGEEK © 2011

Fuentes Bibliográficas

- **Wikipedia**, BPM - Business Process Management, http://en.wikipedia.org/wiki/Business_process_management, [2010-10-30].
- **Netbeans org**, Bienvenido a NetBeans y netbeans.org, www.netbeans.org, [2010-11-03]
- **Wikipedia**, Eclipse software, [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)), [2010-11-04]
- **Joget Workflow**, Vista General del Producto, www.joget.org, [2010-11-10]
- **Bonita Open Solution**, Vista General del Producto, www.bonitasoft.org, [2010-11-12]
- **Bonita Open Solution**, eclipse marketplace, <http://marketplace.eclipse.org/content/bonita-open-solution> [2010-11-12]
- **jBPM**, Makes your work flow , <http://www.jboss.org/jbpm/>, [2010-11-14]
- **jBPM**, wikipedia, <http://en.wikipedia.org/wiki/JBPM>, [2010-11-14]
- **Jeff Hanson**, Manage your business processes with JBoss jBPM, JavaWorld.com, 05/22/06, <http://www.javaworld.com/javaworld/jw-05-2006/jw-0522-jbpm.html>, [2010-11-15]
- **Eclipse Public License v1.0**, Eclipse, <http://www.eclipse.org/legal/epl-v10.html> [2010-12-17]
- **Welcome Apache Maven**, Apache Maven Project, <http://maven.apache.org/>, [2011-01-10]