UNIVERSIDAD DE OVIEDO

ASTURIAS
CAMPUS DE EXCELENCIA
INTERNACIONAL
| AD FUTURUM |

MÁSTER UNIVERSITARIO EN INGENIERÍA WEB

TRABAJO FIN DE MÁSTER

"IMPLANTACIÓN DE UN WORKFLOW EN LA UNIVERSIDAD DE OVIEDO"

NATALIA GARCIA MENENDEZ

El tutor autoriza la defensa del Trabajo Fin de Máster

Fdo. D. Secundino José González Pérez .....

Oviedo, Junio de 2014

Agradecimientos

Aprovecho esta oportunidad para agradecer todo el apoyo recibido a diversas personas que me han brindado su apoyo durante el transcurso de todo el máster y el PFM.

Entre estas personas cabe destacar a Daniel el cual ha sido mi apoyo en todo momento.

Por otra parte me gustaría agradecer el apoyo recibido a Cundi, mi director de proyecto por haberme brindado la oportunidad de realizar este proyecto, a la vez que poder realizarlo en un entorno laboral real.

También me gustaría agradecer todo el apoyo y ayuda recibido por mis compañeros de trabajo, los cuales me facilitaron mucho la comprensión y trabajo con la plataforma eGob!

Por último me gustaría agradecer el apoyo recibido por mi familia y amigos, sin los cuales no habría llegado hasta aquí, ya que aunque en muchas no he estado debido a todo el trabajo realizado a lo largo del máster ellos siempre han estado ahí para mí cuando los he necesitado.

A todos ellos MUCHISÍMAS GRACIAS.

Resumen

El desarrollo de la Administración Electrónica por la Universidad de Oviedo debe tener como primer objetivo hacer efectivo el **derecho de la Comunidad Universitaria a relacionarse electrónicamente** con la Universidad. La entrada en vigor de la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos, LAECSP, ha focalizado la atención en los derechos de los ciudadanos al acceso electrónico a los servicios públicos, al obligar a las Administraciones a disponer de la totalidad de sus servicios accesibles por medios electrónicos antes de finales de 2009.

La LAECSP además de incidir en el papel de las tecnologías como facilitadores de la relación entre ciudadanos y administraciones, complementa las actuaciones que en materia de administración electrónica vienen desarrollando los distintos organismos públicos para mejorar su funcionamiento interno y el servicio que prestan a los ciudadanos por medios telemáticos.

El impacto principal para la Universidad será la puesta a disposición de todos sus servicios públicos y trámites administrativos de forma electrónica, con los requisitos que establece la Ley: **información**, **descarga** y **envío de formularios** cumplimentados, **pago telemático, seguimiento del trámite** por sus sucesivas fases y **finalización** del procedimiento sin requerir la personalización del ciudadano en las dependencias universitarias, salvo circunstancias muy determinadas. Además, los sistemas de información de la Universidad deberían incluir facilidades proactivas, tales como aviso de necesidad de iniciar el procedimiento o cumplimentación automática de los datos en poder de la Universidad. En el diseño de estos servicios se llevará a cabo la **simplificación de los trámites administrativos** con el fin de reducir su carga administrativa.

Como complemento de las actuaciones anteriores, y en línea con la LAECSP, la Universidad debería incorporar un sistema de **tramitación electrónica** que permita ofrecer soluciones para la automatización de procedimientos administrativos, usados en su relación con el estudiante, PDI, PAS y otros agentes externos. Su ámbito de actuación sería el complemento de los sistemas de gestión académica, incorporando capacidades específicamente administrativas. Para ello, es necesario que ofrezca las capacidades necesarias para que el procedimiento administrativo pueda ser realizado íntegramente por medios electrónicos y facilidades para la integración con los sistemas actuales de gestión y sistemas externos, como medios de pago o firma electrónica.

Debido a todas estas necesidades, y en vista de las exigencias de la ley LAECSP se vio la necesidad de integrar un sistema que ayudara y simplificada la tramitación electrónica, puesto que el tramitador actual de la Universidad está basado en módulos y construir un procedimiento de ejecución de distintos tipo de tareas era una tarea muy complicada y laboriosa debido a que tenía que ser totalmente manual.

Por esos motivos se decidió realizar este proyecto que tiene como objetivo principal es implantar un BPM en la que se integre con el gestor de trámites de la Universidad de Oviedo.

Además, pretende mejorar y evolucionar ciertos aspectos del rendimiento del negocio, muchas veces, consistirá en la simplificación de los procesos actuales.

Por tanto, con este proyecto se busca identificar, diseñar, ejecutar, documentar, monitorear, controlar y medir los procesos de negocios que una organización implementa. Dicho enfoque debe contemplar tanto procesos manuales como automatizados y no orientarse meramente a la implementación de software.

La tecnología que se ha decido implementar es Activiti. Gracias a esta tecnología será mucho más fácil realizar procedimientos, tener un control de los trámites iniciados y ya no será necesario el control manual de cada una de las tareas, ya que esto se realizará se forma automática con las herramientas de Activiti.

El lenguaje de programación utilizado ha sido Java, pues que el gestor de trámites de la Universidad estaba desarrollado en éste lenguaje y Activiti también está desarrollado en Java. Las ventajas que ha proporcionado el uso de este lenguaje son debidas a que yo ya tenía una amplia experiencia de trabajo con dicho lenguaje.

Palabras Clave

Activiti, BPM, workflow, procedimiento, gestión de trámites, J2EE, monitorización.

Abstract

The development of electronic government from the University of Oviedo should have as its primary objective to **implement the right of the university community to interact electronically with the University**. The coming into force of Law 11/ 2007 of 22 June, on electronic access of citizens to Public Services, LAECSP, has focused attention on the rights of citizens to electronic access to public services by forcing Administrations to have all their services accessible by electronic means before the end of 2009.

The LAECSP besides to influence the role of technologies as facilitators of the relationship between citizens and administrations, complemented the actions in eGovernment that have been developing various public institutions to improve their internal operations and their service to citizens telematic means.

The main impact for the University will be the availability of all public services and administrative procedures in electronic form, with the requirements of the Act: **information handling and shipping of completed forms, online payment, tracking processed by successive stages and completion of the procedure without requiring customization of the citizen in university departments, except in very limited circumstances**. In addition, information systems at the University should include proactive, such as notice of the need to initiate the procedure or automatically fill the data held by the University facilities. During the design of these services, would have taken out **the simplification of administrative procedures** in order to reduce their administrative burden.

In addition to the above actions, and in line with the LAECSP, the University should incorporate an electronic processing system that allows to offer solutions for automation of administrative procedures used in its relationship with the student, PDI, PAS and other external agents. Its scope would be the complement of academic management system, specifically incorporating administrative capacities. For this it is necessary to provide the necessary administrative procedure can be performed entirely by electronic means and facilities for integration with existing management systems and external systems as means of payment or electronic signature capabilities.

Due to all these needs, and in view of the requirements of the law LAECSP, a need to integrate a system to help and simplify electronic processing was identified, since the current procedures manager University is based on modules and build a method of execution different types of tasks was a very complicated and laborious task because had to be completely manual.

For these reasons it was decided to undertake this project whose main objective is to implement a BPM in that integrates with the procedures manager of the University of Oviedo. Also aims to improve and evolve certain aspects of business performance, many times, will consist on the simplification of the current processes.

Therefore, this project seeks to identify, design, implement, document, monitor, control and measure business processes that an organization implements. This approach should include both automated and manual processes, and to not point merely to the software implementation.

The technology has been decided to implement is Activiti. Thanks to this technology will be much easier to perform procedures, keep track of the paperwork started and having a control on manual tasks will no longer be necessary, as this is done automatically by Activiti tools.

The programming language used is Java, as the procedures manager of the University was developed by using this language and Activiti was developed in Java, too. The benefits provided by the use of this language are due to I already had extensive experience working with it.

Keywords

Activiti, BPM, workflow, procedures, management procedures, J2EE, monitoring.

Índice General

Índice de Figuras

Capítulo 1. Memoria del Proyecto

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

El proyecto está basado en la implantación de una serie de workflows que se integran con la plataforma eGob!, un gestor de trámites de la Universidad de Oviedo.

La plataforma eGob! es un conjunto de componente, que permiten, agilizar la implantación de la administración electrónica en la misma.

La arquitectura se basa en los siguientes principios:

- **Modular:** de forma que se trata de una serie de componentes que ofrecen funcionalidades requeridas para el desarrollo de la eAdministración.
- **Reutilizable**: de forma que estos componentes se implementan una vez y se reutilizan múltiples veces.
- **Flexible**: al permitirse incorporar los elementos específicos a cada proyecto en concreto.
- **Basada en estándares**: por ejemplo con servicios web.
- **Multi-canal**: de forma que los servicios pueden iniciarse a través de Internet, continuarse de forma presencial y seguirse a través de cualquiera de los dos canales.
- **Potente**: pues da cobertura a un importantísimo número de escenarios de eAdministracion.

Es importante resaltar que, si bien todos estos componentes están perfectamente integrados, es posible que alguna aplicación solamente se integre con los módulos que requiera (por ejemplo la firma digital).

Dentro de esta plataforma, el módulo de tramitación da cobertura a las necesidades que plantean la tramitación administrativa y la legislación asociada. El objetivo que aquí nos interesa es desde los puntos de vista tecnológico y funcional.

Es posible utilizar este módulo para la construcción de sistemas de tramitación a medida, de forma que se incorporan determinados automatismos (por ejemplo, integraciones con otros sistemas de información) a fin de dar una solución funcional completa en un ámbito concreto. En este régimen de uso, el sistema ofrece un conjunto de APIs que ofrecen operaciones básicas de tramitación administrativa (creación de expedientes, gestión de personas, etc.).

La plataforma de tramitación plantea el soporte a la tramitación administrativa a través de la parametrización de los procesos administrativos.

Este proyecto además, pretende mejorar y evolucionar ciertos aspectos del rendimiento del negocio, muchas veces, consistirá en la simplificación de los procesos actuales.

Por tanto, con este proyecto se busca identificar, diseñar, ejecutar, documentar, monitorear, controlar y medir los procesos de negocios que una organización implementa. Dicho enfoque debe contemplar tanto procesos manuales como automatizados y no orientarse meramente a la implementación de software.

1.2 Resumen de Todos los Aspectos

1.2.1 Introducción

En esta primera parte de la documentación se pretende que la persona que lo lea pueda ir comprendiendo las ideas fundamentales del proyecto, he ir introduciéndose en todos sus detalles poco a poco.

1.2.2 Aspectos teóricos

En este apartado se pretende explicar aquellos conceptos teóricos los cuales el usuario no tiene por qué conocer, por tanto se lleva a cabo una descripción que aunque sea breve intenta explicar los conceptos más importantes.

1.2.3 Planificación del proyecto

En este apartado se describe la planificación temporal del proyecto mediante un diagrama Gantt.

1.2.4 Análisis

El objetivo de este apartado es realizar una especificación detalla de todo el sistema.

1.2.5 Diseño del sistema

En este apartado se lleva a cabo una definición de la arquitectura del sistema y componentes de los cuáles está formado.

1.2.6 Implementación del sistema

Este apartado se describen todas las herramientas utilizadas para llevar a cabo el desarrollo del proyecto, junto con una descripción de las clases realizadas.

1.2.7 Desarrollo de las pruebas

En este apartado se indican todos los resultados obtenidos en las pruebas realizadas.

1.2.8 Manuales del sistema

En este apartado se pueden encontrar descripciones detalladas de todos los manuales realizados para los diversos perfiles de usuarios.

1.2.9 Conclusiones y ampliaciones

El objetivo de este apartado es reflejar mi opinión una vez he finalizado el proyecto. También las posibles ampliaciones que puede tener el proyecto.

1.2.10 Presupuesto

En este apartado se puede ver un presupuesto detallado de costes del proyecto.

1.2.11 Referencias Bibliográficas

En este apartado se reúnen todas las fuentes de información consultadas durante el transcurso del proyecto.

1.2.12 Apéndices

Este último apartado contiene el código fuente de la aplicación y las indicaciones del contenido del CD.

Capítulo 2. Introducción

2.1 Justificación del Proyecto

Hoy en día en la Universidad de Oviedo existen numerosos procesos que todavía son gestionados de forma manual, lo cual en una institución del tamaño de ésta provoca y tiene algunos inconvenientes.

Debido al gran número de procesos y personas con diferentes roles por los que pasa cada uno de ellos es muy fácil que se pierda algún documento, o que no se pueda conocer en qué situación se encuentra cada trámite.

Para solucionar éste problema la Universidad dispone de un gestor de trámites, el cual hasta el momento era un desarrollo propio y "manual", por tanto los trámites se definían una sola vez y no se adaptaban a posibles mejoras o simplificaciones, ya que cualquier mínimo cambio en un trámite lleva consigo muchas modificaciones.

Gracias a la implantación de Activiti todos estos problemas se reducen eficazmente, ya que ahora la creación de un nuevo proceso, o la modificación de uno ya existente se puede realizar de una manera muchos más simple y ágil.

Además, Activiti también permite una eficaz monitorización de cada proceso, y por tanto ya se puede conocer en qué estado se encuentra o a la espera de quién está y porque no se puede continuar con su tramitación.

2.2 Objetivos del Proyecto

El objetivo principal del proyecto es la implantación de un sistema de control y creación de workflows que permita la simplificación, mejora y automatización de una serie de procesos administrativos de la Universidad de Oviedo.

Pretende mejorar y evolucionar ciertos aspectos del rendimiento del negocio, muchas veces, consistirá en la simplificación de los procesos actuales.

Con este proyecto se busca:

- Identificar
- Diseñar
- Ejecutar
- Documentar
- Monitorizar
- Controlar
- Y medir los procesos de negocios que una organización implementa.

2.3 Estudio de la Situación Actual

En la actualidad existe diversas suite BPM, algunas de las más importantes que han sido evaluadas antes de escoger Activiti se puede ver en detalle a continuación. El principal motivo por el que se ha escogido Activiti ha sido porque es de las pocas que es completamente gratuita, y dentro de las gratuitas la cual en el momento de inicio del proyecto era más estable, puesto que jBPM en aquel momento estaba en la versión 5.4 y actualmente ya está en la 6, por lo tanto estaba en transición.

2.3.1 Evaluación de Alternativas

2.3.1.1 Bonita

2.3.1.1.1 Descripción

Bonita Open Solution, es una suite ofimática para la Gestión de procesos de negocio (BPM) y realización de Workflows, creada en 2001 y de código abierto bajo GPL v2 para Bonita Studio y LGPL para el motor de ejecución y las aplicaciones web.

La última versión disponible es la 6.0.4 de Octubre de 2013.

Bonita está compuesta de 3 partes principales:

Bonita Studio: Aplicación de escritorio que permite al usuario definir y modificar gráficamente los procesos de negocio siguiendo el estándar BPMN. Además permite conectar dichos procesos con otros sistemas o piezas de información como mensajería, ERP, ECM, bases de datos, etc. También permite la definición de formularios, generar aplicaciones web autónomas. Para el desarrollo de conectores se basa en Eclipse.

Bonita BPM Engine (motor): Bonita puede integrarse con otras aplicaciones así, Bonita execution engine, proporciona una API en Java, que permite al usuario interactuar programáticamente con el proceso o los procesos. Está disponible bajo licencia LGPL y basada en Hibernate.

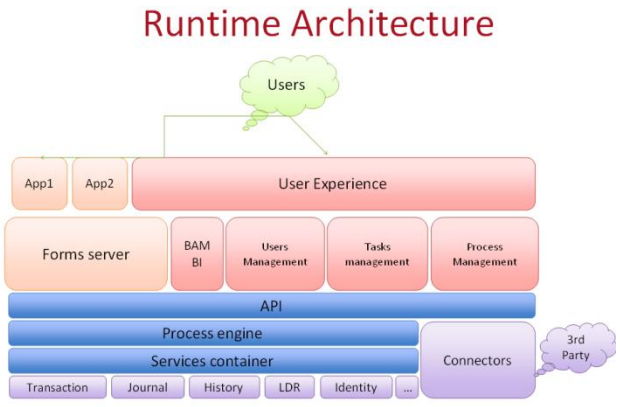


Figura 2.1 Arquitectura Bonita

API:

- ManagementAPI: Operaciones relacionadas con la instalación y eliminación de procesos, gestión de recursos.
- QueryDefinitionAPI: Operaciones de consulta relacionadas con la definición del modelo de objetos (Definition).
- RuntimeAPI: Modificación de operaciones relacionadas con el modelo de objetos de ejecución (Runtime).
- QueryRuntimeAPI: Operaciones de consulta relacionadas con el objetos de modelo de ejecución (Runtime).
- RepairAPI: Operaciones de adminsitración avanzadas
- CommandAPI: Operaciones para ejecutar ordenes disponibles en un proceso.
- IdentityAPI: Operaciones relacionadas con el módulo de usuarios
- BAMAPI: operaciones de consulta para obtener estadísticas

Bonita User Experience (experiencia de usuario): es un portal web que permite a cada usuario final gestionar en una interfaz similar a la del correo web (webmail-like) todas las tareas y procesos en las cuales él está involucrado. El portal también permite al propietario de un proceso administrarlo y obtener informes sobre procesos.

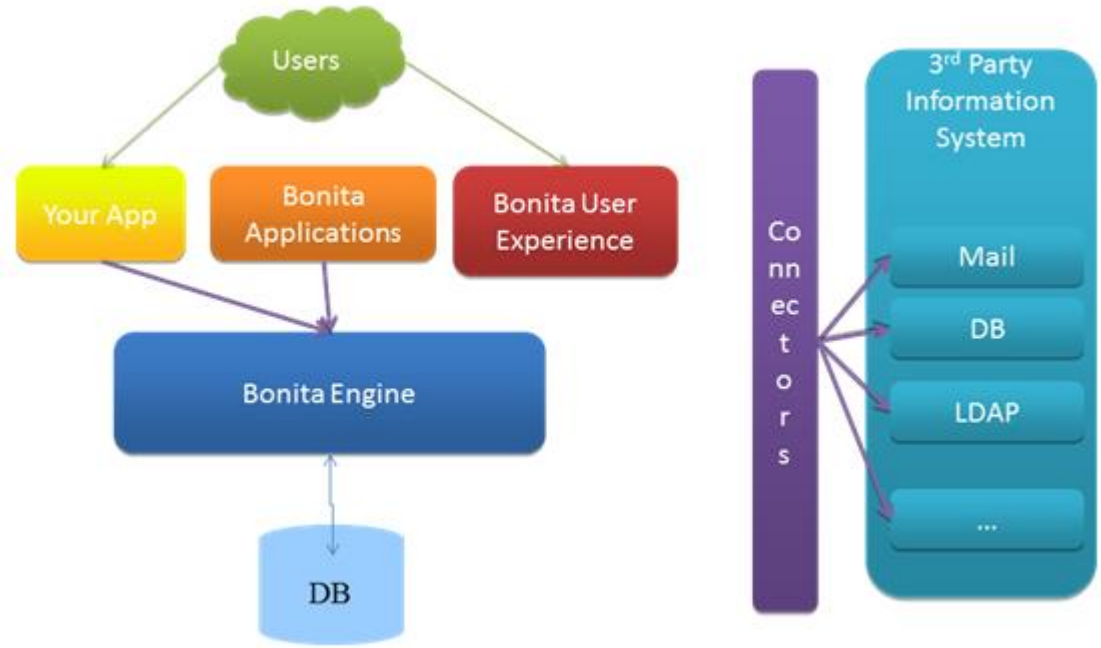


Figura 2.2 Estructura Bonita

Algunas aplicaciones BPM desarrolladas con Bonita son:

- Gobierno de las Islas Canarias (BPM en aplicaciones electrónicas del gobierno)
- Estado del Cantón de Ginebra (Desarrollo de aplicaciones electrónicas gubernamentales)
- Universidad Complutense de Madrid (BPM en la educación)
- CSI Piemonte (Plataforma de gestión de documentos para la e-administración)
- Andago, OpenCities (plataforma de e-administración)

2.3.1.1.2 Ventajas

- Software muy completo, que permite una alta parametrización y adaptabilidad a las necesidades del usuario.
- Facilidad para el diseño de procedimientos, con el interfaz gráfico de Bonita Studio.
- Accesibilidad al código y amplia documentación, ejemplos y librerías ya desarrolladas sobre todo para versiones 5.3, no tanto para la actual 5.4.
- Integración con múltiples bases de datos y servidores como Tomcat 6.
- Dispone de múltiples conectores con otros sistemas Alfresco, Liferay, etc.

2.3.1.1.3 Problemas e Inconvenientes

- Lentitud y algunos fallos en la aplicación de desarrollo Bonita Studio.
- Entre los principales problemas encontrados para la realización de estas pruebas, la dificultad para conocer cuál es la correcta instalación o modo de trabajo del sistema (como librería, como servicio web Rest etc.) y la mejor forma de instalar dicho software en entornos de explotación.
- Gestión de usuarios externos: autenticación de los usuarios y conexión con LDAP.

2.3.1.2 Intalio

2.3.1.2.1 Descripción

Sistema de gestión de procesos Open Source, basado en Java. Está construido en torno a las normas basadas en BPML (Business Process Modeling Language) y BPEL (Business Process Execution Language). Proporciona todos los componentes necesarios para el diseño, despliegue y gestión en cualquier proceso de negocio.

Sus principales componentes:

- Una herramienta para el diseño de los procesos de negocio, basada en Eclipse
- Un motor que ejecuta los artefactos de software generados por el diseñador de procesos.
- Un servidor de Aplicaciones donde residirán lo servicios de procesos de negocio que despleguemos.

Versiones:

Works Community: De utilización totalmente gratuita, contiene los módulos básicos de una solución BPM y no tiene ningún tipo de soporte

Works Enterprise Edition*:* Incluye módulos empresariales que le dan un valor añadido al producto.

	Community Edition	Enterprise Edition
Coste	Gratis	Desde 9,500 USD/EUR al año
Código Disponible	80% Código Abierto	100% Código Abierto
Distribución	Binaria	Binaria + Código Abierto
Soporte	Comunidad Online	Acuerdo de Nivel de Servicio dela Empresa
Mantenimiento	Actualización manual	Actualización automática
Indemnización	Ninguna	Hasta $1,000,000 (Opcional)

Figura 2.3 Versiones Intalo

Intalio|BPMS Designer: Es una herramienta para el diseño de procesos utilizando la notación BPMN. Esta soportado por el proyecto Eclipse Europa, y puede ser instalado en ambientes windows, linux y Mac OS X.

Formado por:

- ***Diseñador de Formularios***: Proporciona un diseñador de formularios para su integración dentro de los procesos de las tareas de usuario. Está basado en le edición Open Source de TIBCO General Interface y permite la realización de formularios AJAX. Dichos formularios de guardan en Xforms.
- ***Data Mapper****:* Permite la asignación y transformación de datos entre los sistemas participantes del proceso de una manera visual. Genera código XPathó XSLT a partir de los diagramas gráficos. Soporta transformaciones XSLT externas

Intalio|BPMS Server: Es el motor de procesos de Intalio. Ejecuta los procesos creados con Intalio Designer.

Está basado en J2EE y una arquitectura SOA (Axis 2). Integra dos componentes Open Source interconectados: Apache ODE (Motor de BPEL 2.0) e IntalioTempo (Motor de Workflow con soporte tareas humanas).

Ofrece dos interfaces visuales: una de Administración y otra de Usuarios finales de los procesos. Además de interfaces de interconexión con aplicaciones externas: Publicación de los procesos de negocio como servicios web y llamadas a servicios web externos.

Intalio|BPMS Workflow Es la implementación de Workflow basada en el proyecto Apache Tempo, el cual integra formas de trabajo mediante controles Xform, soportando el intercambio de mensajes con los procesos, la definición de usuarios, roles, y diversos patrones de Workflow.

IntalioBRE (Business Rule Engine-Motor de reglas): Permite la creación de reglas de negocio complejas, invocadas desde cualquier punto del proceso de negocio.

IntalioBAM (Business ActivityMonitoring–Monitor procesos): Cuadros de mando, editor de métricas y diseñador de informes.

Intalio|ECM (módulo adicional de pago): Gestor documental Alfresco integrado.

Intalio|Portal (módulo adicional de pago): Gestor de contenidos y portales Liferay. Incluye un portlets específico para el acceso al área de usuario de procesos desde Liferay

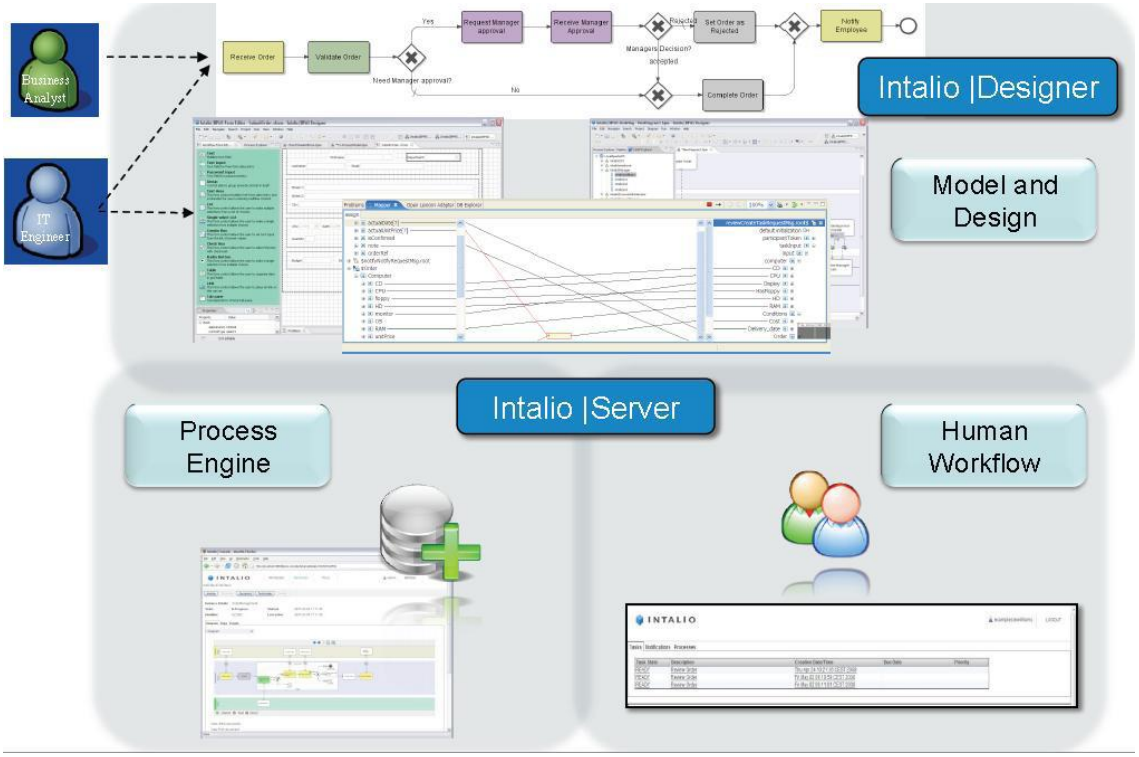


Figura 2.4 Arquitectura Intalio

2.3.1.2.2 Ventajas

- Fácil de instalar y múltiples ejemplos
- Proporciona un entorno de desarrollo basado en eclipse bastante completo y cómodo
- Completo editor de formularios, permite la opción de desarrollo con Ajax
- Permite lo conexión con servicios web (Soap y Rest) de manera muy simple.
- Proporciona portal para la gestión de procesos tanto por parte del administrador como de los usuarios.

2.3.1.2.3 Inconvenientes

- No soporta conectores java, es decir no es posible la conexión directa con una librería sino que deberá ser vía servicio web.
- El mapeo de datos es complejo y poco intuitivo.
- El diseño de formularios no es tan automático como en otras herramientas.
- Proporciona la posibilidad de conectar con diversas bases de datos, Api java y otros conectores como Alfresco pero solo la versión empresarial.

2.3.1.3 jBPM

Es un gestor de procesos de negocio flexibles, Business Process Management (BPM). Es el nexo de unión entre los analistas del negocio, los desarrolladores y los usuarios finales.

Actualmente se encuentra en la versión 6 la cual da soporte completo a la versión de BPMN 2.0.

Está compuesto por un núcleo ligero que incorpora un motor de flujo de trabajo extensible escrito en Java, éste permite ejecutar procesos de negocio usando la última especificación BPMN 2.0. Puede funcionar en cualquier entorno Java, incrustado en su aplicación o como un servicio. También ofrece una gran cantidad de funciones y herramientas para apoyar los procesos de negocio a lo largo de todo su ciclo de vida:

- Pluging para Eclipse basado en la web para apoyar la creación gráfica de sus procesos de negocio.
- Persistencia basada en JPA.
- Dispone de tareas manuales para la inclusión de tareas que deben ser realizadas por actores humanos.
- Gestión de instancia de proceso de soporte de la consola de gestión, listas de tareas y formulario de tareas de gestión y presentación de informes
- Registro de un historial de los procesos (para realizar consultas / monitoreo / análisis)
- Integración con Seam, Spring, OSGi, etc

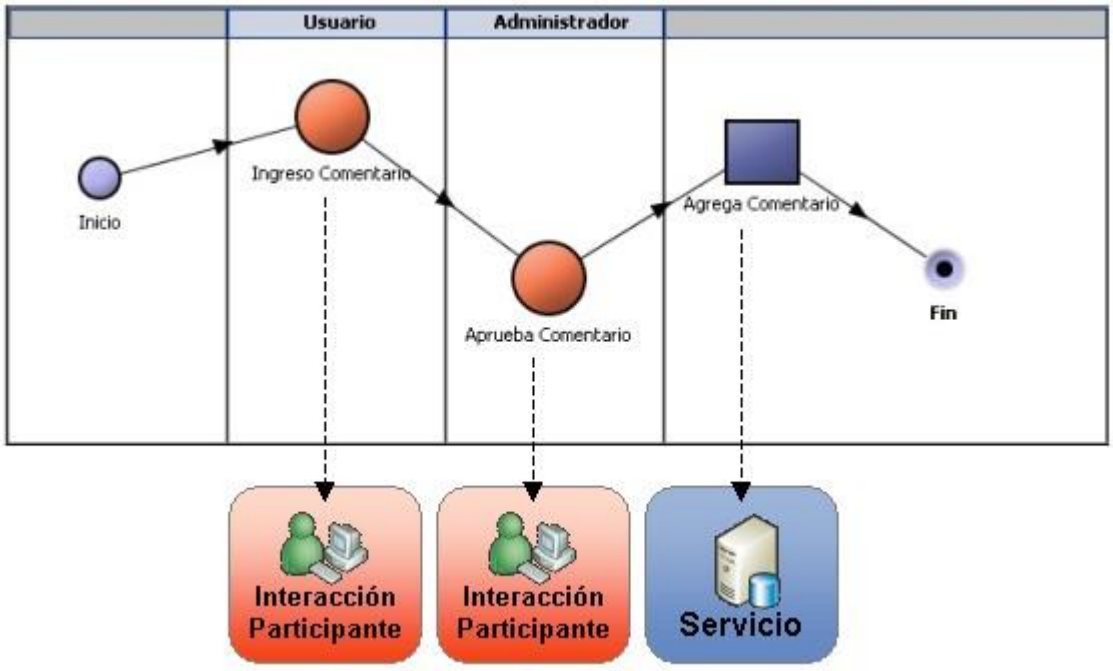


Figura 2.5 Integración de jBPM

jBPM6 es la última versión de la comunidad del proyecto jBPM. Se basa en **BPMN 2.0** y soporta todo el ciclo de vida del proceso de negocio, desde la creación a la supervisión y de gestión.

Dispone de código abierto de ejecución y gestión de procesos de negocio, incluyendo:

- Un motor interno ligero en Java y la ejecución nativa de BPMN 2.0.
- Interacción humana mediante un servicio de tarea WS-HT independiente.
- Modelado de procesos BPMN modelado de procesos a través de Eclipse para los desarrolladores y a través de la web para los usuarios empresariales.
- Herramientas web para modelar, implementar, ejecutar y supervisar sus procesos, incluyendo, por ejemplo, un modelador de datos y la forma, de simulación, de despliegue, listas de tareas, etc.
- La gestión y la implementación de sus procesos utilizando tecnologías de bajo como Git y Maven
- Un servidor de ejecución que se puede conectar a distancia con (REST, JMS) y se puede implementar en un entorno de clúster de equilibrio de carga y alta disponibilidad.

2.3.1.3.1 Ventajas

- Fácil de instalar y múltiples ejemplos.
- Proporciona un entorno de desarrollo basado en eclipse bastante completo y cómodo.
- Completo editor de formularios.
- Proporciona portal para la gestión de procesos tanto por parte del administrador como de los usuarios.

2.3.1.3.2 Inconvenientes

- Gestión de usuarios externos: autenticación de los usuarios y conexión con LDAP.
- El mapeo de datos es complejo y poco intuitivo.
- El diseño de formularios no es tan automático como en otras herramientas.

2.3.1.4 Activiti

2.3.1.4.1 Descripción

Es un software ligero para la gestión de procesos orientado a gente de negocios, desarrolladores y administradores de sistema. Su núcleo desarrollado en java y basado en BPMN2. Es software abierto bajo licencia Apache.

Activiti puede integrarse con cualquier aplicación java, puede desplegarse en cualquier servidor y se integra con Spring.

Uno de los principales objetivos de Acitivi es proporcionar un entorno ligero basado en java y que fácil de utilizar por los desarrolladores.

Activiti está formada por diferentes componentes que combinados dan lugar a una solución integral de BPM.

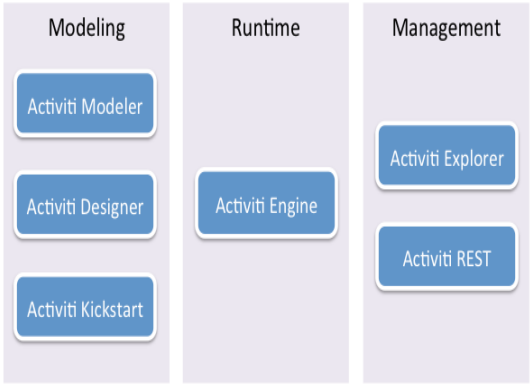


Figura 2.6 Componentes Activiti

- **Activiti Designer:** plugin de Eclipse, permite al desarrollador trabajar tanto con el proceso en BPML como en el esquema gráfico. Facilita el acceso entre las diferentes piezas relacionadas con la lógica.
- **Activiti Modeler, Signavio** (activiti-modeler): Versión adaptada del editor web de procesos Signavio, de código abierto. Esta herramienta web permite el diseño de procesos en BPMN2.0 Los procesos son almacenador por el servidor en un sistema central, el sistema de ficheros actúa como un repositorio.

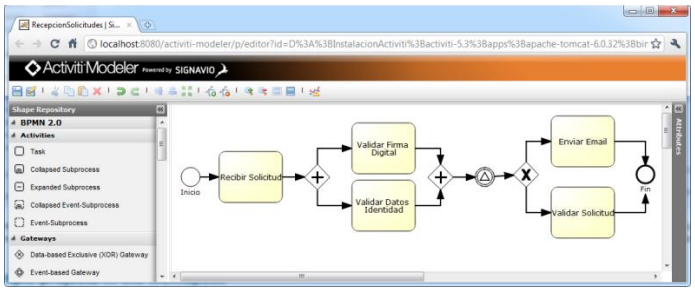


Figura 2.7 Activiti Modeler de Signavio

- **Activiti Administrator (activiti-administrator):** Aplicación web para la gestión de usuarios y grupos.

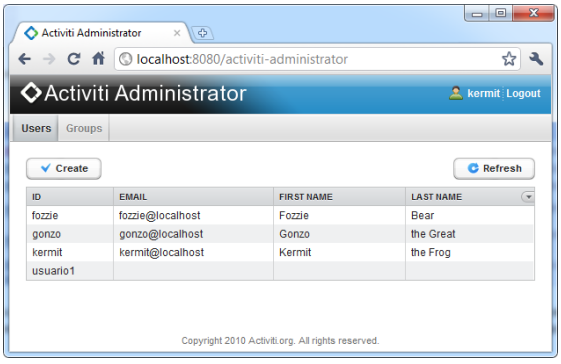


Figura 2.8 Activiti Administrator

- **Activiti Probe (activiti-probe):** Web de administración, permite la gestión de los procesos (desplegar, eliminar, consultar su estado, etc.), y la visualización de las tablas de la base de datos.

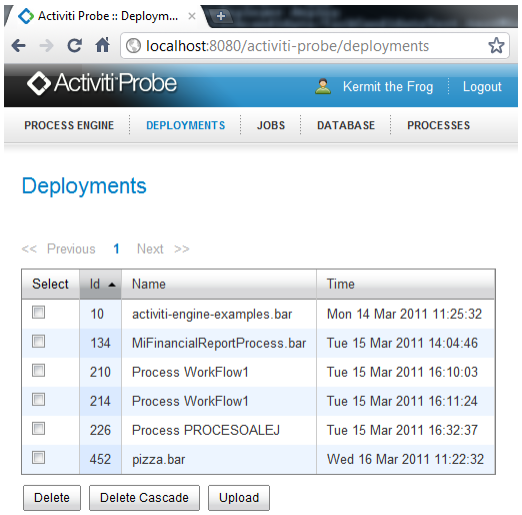


Figura 2.9 Activiti Probe

- **Activiti Explorer (activiti-explorer):** Interfaz es la interfaz de usuario para los participantes en los procesos, permite iniciar nuevas tareas y consultar y completar las existentes.

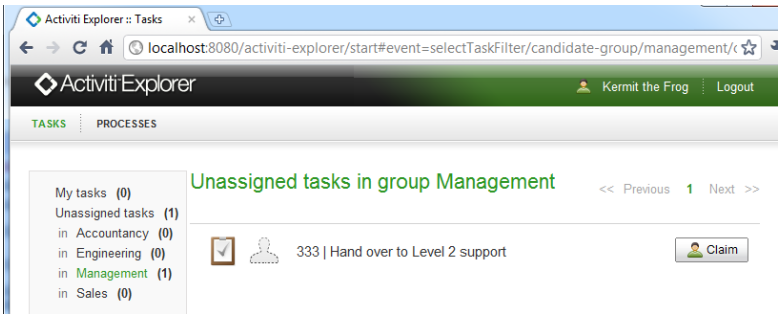


Figura 2.10 Activiti Explorer

- **Activiti Cycle (activiti-cycle):** módulo con el que se dota de colaboración social a la aplicación. Utilizada para navegar entre los repositorios con modelos de proceso y permite la transformación entre diversos formatos. Es una aplicación web para facilitar la colaboración entre personas de negocio desarrolladores.

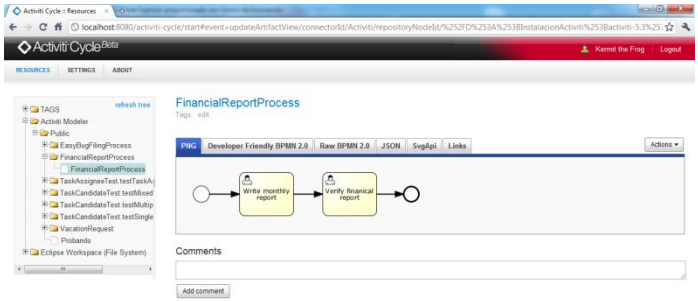


Figura 2.11 Activiti Cycle

- **Activiti Engine**: motor de procesos Java que ejecuta los procesos BPMN2, es el núcleo de la aplicación. Proporciona una API java y permite a aportar del modelo de procesos, ejecutar código Java y Scripts. También desarrollar nuevos componentes o elementos adaptados a las necesidades de los desarrolladores.
  Incorpora una sencilla API de Java que permite a las organizaciones definir sus propias tareas. Si se quieren implementar un conjunto básico de funcionalidades, se pueden modelar gráficamente y ejecutarlas mediante el motor. De esta forma, la actividad personalizada puede ser reutilizado y reconfigurada por un analista mediante la interfaz gráfica de usuario.

El Activiti Engine es totalmente abierto. Además de acceder a través de la API de Java, se puede acceder vía REST como interfaz de consultas y control de la aplicación. Capacidad de automatizar los procesos de prueba. Incorpora la posibilidad de hacer pruebas con JUnit y testear los procesos

2.3.1.4.2 Ventajas

- Documentación clara y acceso al código
- API sencilla para la gestión de procesos, tareas, envió de señales etc.
- Proporciona diversas aplicaciones web. Además de la interfaz de usuario y administración muy completa (permite por ejemplo ver las tablas de la base de datos), también proporciona un navegador para la visualización de los procedimientos y su conversión si el proceso es sencillo a diversos formatos, etc.
- Dispone de tareas predefinidas que facilitan el diseño del procedimiento, email, servicios web, espera de señales, conectores Java etc. Todas ellas fáciles de configurar.
- Soporta BPMN 2.

2.3.1.4.3 Problemas e Inconvenientes

- La instalación se realiza mediante un Script y no es tan sencilla como otras aplicaciones.
- A pesar de soportar BPMN2 y permite la importación de procesos no funciona correctamente
- No tiene editor gráfico de formularios se deben realizar con HTML, aunque son sencillos de diseñar
- Cuando se ejecuta muchas veces en algún explorador como Firefox la aplicación web de usuario o administrador pueden surgir errores de autenticación inesperados.
- No dispone de conectores con Liferay Alfresco etc.

Capítulo 3. Aspectos Teóricos

En este apartado se va a tratar de explicar algunos de los conceptos básicos para poder entender bien las partes más importantes del proyecto así como poder comprender la finalidad y función del mismo.

3.1 Proceso de negocio y workflows

La lógica de negocio se puede definir en forma de una serie de tareas relacionadas que se deben ejecutar si se cumplen unas determinadas condiciones, de forma secuencial o paralela, o mantenerse en espera de algún evento que reanude su ejecución.

Al conjunto de estas tareas se le conoce como proceso de negocio. La gestión de procesos de negocio (BPM–Business Process Management) se encarga del modelado de los procesos de negocio, y define el flujo de trabajo (workflow) que rige su funcionamiento.

Un workflow modela de forma gráfica las relaciones entre las distintas tareas. Un sistema de gestión de procesos de negocio es una aplicación que gestiona estos flujos de trabajo, y se encarga de automatizar la secuencia de acciones, actividades o tareas que componen el proceso y se encarga de la persistencia de su estado.

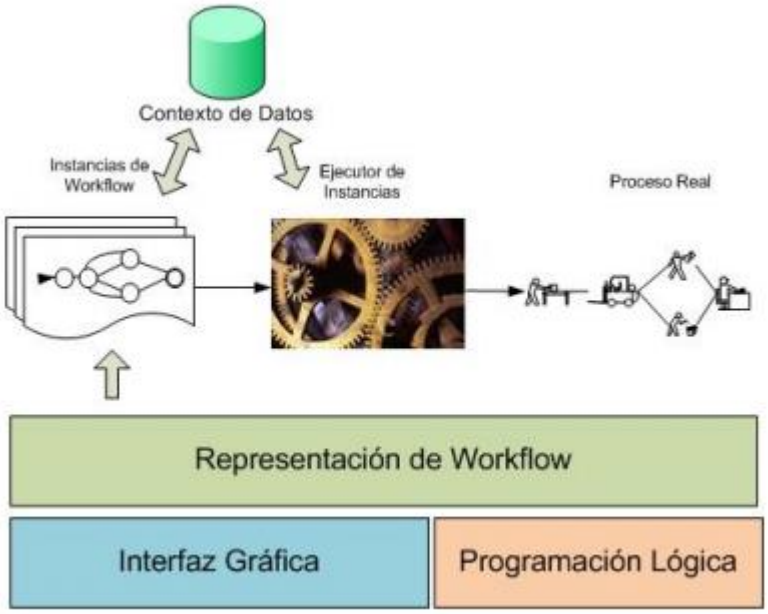


Figura 3.1 Arquitectura general de interpretación de Workflows

Esta automatización permite integrar los procesos de la empresa y rediseñarlos de acuerdo a nuevas estrategias. Existen varios sistemas de gestión de procesos de negocio, y uno de ellos es Activiti, se trata de una tecnología Open Source.

3.2 BPMN

Business Process Modeling and Notation (BPMN) es uno de éstos estándares de modelado de procesos de negocio, proporciona una notación específica que está basada en diagramas de flujo y es similar a los diagramas de actividad de UML.

Actualmente, existen muchos estándares que compiten para ser el mejor lenguaje de modelado de procesos de negocio para su utilización en las herramientas de modelado de procesos.

Cabe destacar que uno de los propósitos para el desarrollo de BPMN es crear un mecanismo sencillo y comprensible para la creación del *BPM*, mientras que al mismo tiempo ser capaz de manejar la complejidad inherente a los procesos de negocios.

El enfoque adoptado para manejar estos dos requerimientos contradictorios fue organizar la notación gráfica en categorías específicas. Esto proporciona un pequeño conjunto de categorías de notación para que el lector de un diagrama BPMN pueda reconocer fácilmente los tipos básicos de elementos para entender el diagrama.

Dentro de las categorías base de elementos, la variación y la información adicional puede ser añadida para soportar los requisitos de complejidad sin cambiar drásticamente el aspecto base del diagrama.

BPMN Versión 1.x: Nace como evolución de BPML (Business Process Modeling Language), desarrollado por el BPMI.org en 2001. Es un lenguaje basado en XML para la ejecución de procesos y no tiene representación gráfica. Intenta "normalizar" la gran variedad de notaciones existentes para la representación de procesos. Las versiones 1.x ofrecen una notación gráfica estandarizada para la representación de los procesos de negocio, aunque no tiene un modelo que diga como serializar dicho modelo. (Ver anexo BMPN)

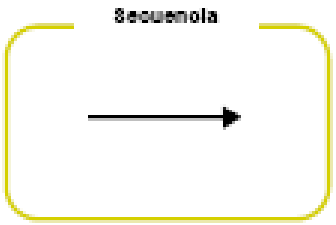
BPMN Versión 2.0: En 2009 salió al mercado la nueva versión, que cambia el nombre a "Business Process Modeland Notation". Incluye una forma estándar basada en XML para la serialización o almacenamiento de los modelos y por tanto, facilita la portabilidad de los modelos entre herramientas. Además, añade soporte para nuevos tipos de diagramas y mejor soporte para las tareas de usuario. En esta nueva versión del lenguaje de BPMN se amplía la sintaxis original de modelado de procesos de negocio para poder captar más detalles y describir una mayor variedad de procesos de negocio. (Ver anexo BPMN2.0)

Los principales Elementos de BPMN son:

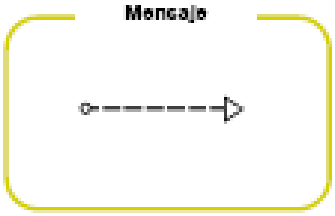
- Un **pool** representa un participante en un proceso. Este actúa como un contenedor gráfico de las actividades que pueden ser ejecutadas por el participante.
- Un **Lane** representa una partición lógica del pool, verticalmente u horizontalmente. Los lanes son usados para organizar y categorizar las actividades.
- Una **actividad** se representa con un rectángulo redondeado y es un término genérico para el trabajo que un participante ejecuta dentro de un proceso de negocio. Una actividad puede ser atómica o compuesta. Los tipos que hay son: Tarea y Sub-Proceso.

El Sub-Proceso se distingue por una pequeña marca de suma en la parte central inferior de la figura.

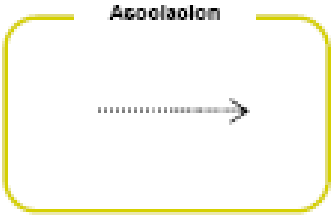
- **Flujo de Secuencia**: Se usa para mostrar el orden o secuencia de las actividades que son ejecutadas en el proceso.



- **Flujo de Mensaje:** Se usa para mostrar flujo de mensajes entre dos participantes del proceso (dos pools separados en el diagrama representan dos participantes).



- **Flujo de Asociación**: Se usa para asociar datos, texto, y otros artefactos con los objetos de flujo. Las asociaciones se usan para mostrar entradas y salidas de actividades.

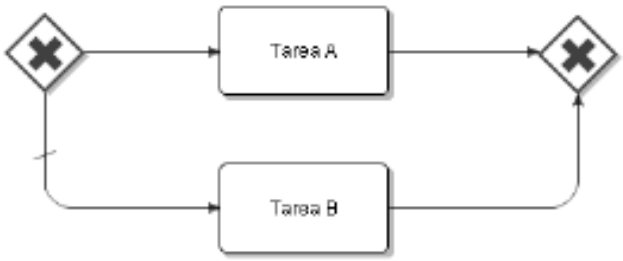


- **Evento**: Es algo que pasa durante el curso del proceso de negocio. Estos eventos afectan al flujo del proceso y suelen tener una causa o resultado. Hay tres tipos de eventos, basados en cuando afectan al flujo: Inicio, Intermedio, y de término.
- **Evento de Inicio:** Establece donde un proceso inicia su ejecución.

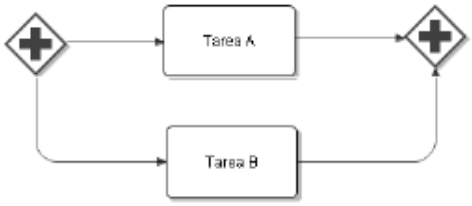
- **Evento Intermedio:** Establece puntos de ejecución intermedios en un flujo de proceso.

- **Evento de Fin:** Finaliza el flujo de un proceso bajo ciertas condiciones: envió de mensajes, gestión de excepciones, compensación, entre otros.

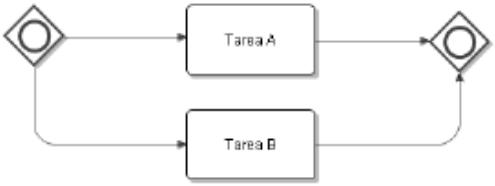
- Una **gateway** (pasarela, bifurcación o compuerta), representa puntos de decisión en el proceso, para que el flujo sea condicionado y permita su direccionamiento sobre diversas rutas de ejecución. Se representa por la típica figura de diamante y se usa para controlar la divergencia o convergencia de la secuencia de flujo. Así, esto determina las decisiones, así como la creación de nuevos caminos, la fusión de estos o la unión. Los marcadores internos indicarán el tipo de control de comportamiento.
- Gateway **Exclusivo**: Solo una rama puede ser ejecutada.



- Gateway **Paralelo**: Todas las ramas puede ser ejecutada.



- Gateway **Inclusivo Eventos**: Una o más ramas pueden ser ejecutada.



3.3 Suite BPM

Una Suite BPM es una herramienta que cuenta con todo un conjunto de funciones para llevar a cabo todo el trabajo con los BPMN, cuenta con un proceso de modelado de procesos BPMN, una herramienta de simulación de procesos, motores de reglas de negocio, sistemas de monitorización de procesos, etc.

Figura 3.2 BPMS: Business Process Management Suites

jBPM, en sus orígenes, era un motor de workflow, pero hoy en día se distribuye como una Suite BPM, ya que integra motores de reglas, visores de instancias de procesos, repositorios, etc.

Intalio y Bonita están dentro de la categoría de Suites BPM por el contrario Signavio se distribuye entre otras herramienta como modelador web de procesos en JBPM. Estas tres herramientas tienen una pequeña parte inicial que es gratuita, pero posteriormente para poder realizar un sistema completo con BPM para un gran número de usuarios y con toda la funcionalidad disponible habría que escoger opciones de pago.

Activiti es una clara alternativa a jBPM ya que ha sido fundada por una parte del grupo de creadores de jBPM. Activiti está licenciado bajo la licencia Apache 2.0 para fomentar el uso generalizado y la adopción del motor de Activiti para BPMN 2.0.

3.4 Otros conceptos

3.4.1 BPM

Business Process Management o Gestor de procesos de negocio. Es el conjunto de herramientas y metodologías para modelar, automatizar, monitorizar y optimizar los procesos de una organización. Un BPM permite definir y gestionar que pasa, desde el principio al final del proceso.

3.4.2 BPMS

Business Process Management System o sistema de gestión de procesos de negocio. Es el producto que comprende todas aquellas herramientas y elementos software que permiten desde el modelado hasta la ejecución y monitorización de los procesos software.

3.4.3 XPDL

XML Process Definition Language o Lenguaje de definición de procesos XML. Es la representación basada en XML de un proceso. Proporciona una notación estándar para la representación de procesos, permitiendo que puedan ser importados/exportados por cualquier editor que lo soporte.

Surge en 2001 y fue el primer intento de definición de un estándar para la definición y ejecución de procesos de negocio. La WfMC publicó la especificación XPDL 2.0 en octubre de 2005, modificada posteriormente en 2008 dando lugar a XPLD v2.1.

3.4.4 JPDL

Java Process Definition Language o Lenguaje de definición de procesos Java. Es el lenguaje específico definidos por Jboss para los procesos definidos en Java, el cual permite la descripción de procesos de negocio mediante la definición de tareas y a través de un lenguaje orientado a grafos. Lenguaje maduro y estable, utilizado por el motor jBPM (aunque a partir del 2007 también incorpora una extensión de BPEL).

3.4.5 BPML´s

Business Process Modeling Languages o Lenguajes para el modelado de procesos de negocio. Son metalenguajes que permiten llevar a cabo el modelado de procesos basándose en XML y realizar la integración con sistemas de información y de gestión utilizando servicios web (Generalmente a través de WSDL). Dentro de los BPML destacan el BPML desarrollado por la BPMI y el WS BPEL (antes de su estandarización BPEL4WS)

3.4.6 BPML

Business Process Modeling Language. Es el metalenguaje para modelar los procesos desarrollado por BPMI. Lenguaje en el que se modelan los procesos de negocio, se permite la traducción de un formato gráfico a un formato entendible por las máquinas.

3.4.7 BPEL o WS BPEL

Web Services Business Process Execution Language o Lenguaje de Ejecución de Procesos de Negocio con Servicios Web. Es el estándar de OASIS para ejecutar procesos de negocio, utilizado para la orquestación de procesos y servicios.

WS BPEL es un lenguaje XML de ejecución de procesos de negocio mediante servicios Web, permite Orquestar la comunicación entre diferentes servicios Web. Nace como combinación de WSFL (Language de IBM, orientado a grafos) y XLANG (Lenguaje de Microsoft, basado en un control de flujos con secuencias, condiciones, etc.).

No existe una especificación gráfica estándar para las composiciones WS-BPEL, pero BPMN, sí que es una notación estándar para procesos de negocio que incluye, traducción automática a código WS-BPEL ejecutable.

3.4.8 BPEL4People

Es el estándar para el flujo de trabajo entre los participantes de un proceso de negocio. Es una extensión de BPEL, que soporta roles de usuario mediante la introducción de una nueva actividad de usuario o persona.

3.4.9 PVM

Process Virtual Machine o Máquina virtual de procesos. Es la máquina virtual de procesos que permite ejecutar múltiples lenguajes de ejecución. Iniciativa formada por diversos fabricantes Open Source (Jboss, Bonita).

3.4.10 SOA

Service Oriented Architecture o Arquitectura Orientada a Servicios. Es una arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Según el W3C es un "Conjunto de componentes que pueden ser invocados, cuyas descripciones de interfaces se pueden publicar y descubrir"

3.4.11 ESB

Es una plataforma de servicios, que permite la disponibilidad de funcionalidades existentes en diversos sistemas heterogéneos, realizando tareas de conexión, adaptación, transporte, transformación, integración, etc, mediante servicios.

Es un combinado de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos.

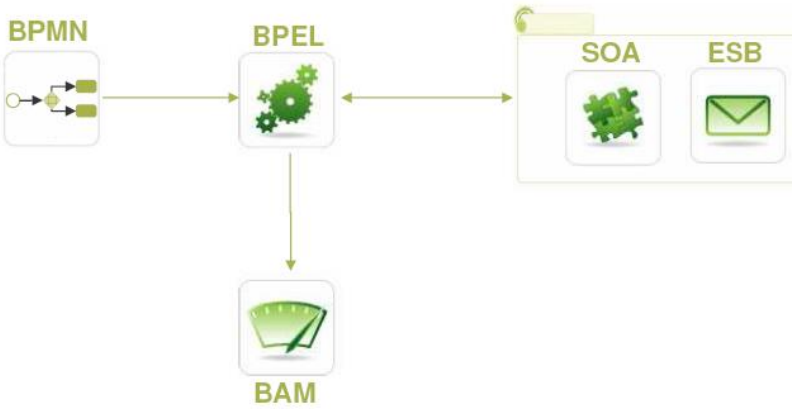


Figura 3.3 Disciplinas de BPM

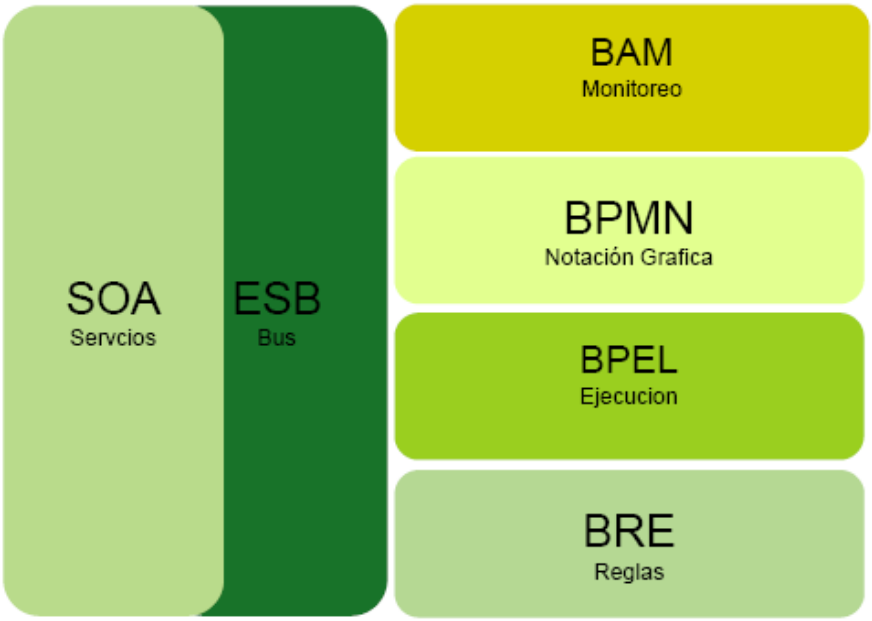


Figura 3.4 Arquitectura ESB + BPM

Capítulo 4. Planificación del Proyecto y Resumen de Presupuestos

4.1 Contexto del proyecto

Aunque se trata de un proyecto fin de máster y que realmente sólo ha sido realizado por una persona, por mí, en la asignatura de Dirección y Gestión de Proyectos Web se nos indicó que para realizar una planificación y un presupuesto más reales debíamos encuadrar el proyecto dentro del contexto de una posible empresa real. Por tanto a continuación explico brevemente el contexto de la empresa.

El proyecto será realizado por una pequeña empresa que cuenta con unos 50 trabajadores. El equipo está formado por dos directores los cuales son también propietarios de la empresa. Uno de ellos es Ingeniero Informática y cuenta con una experiencia de unos 10 años en la dirección de proyectos de informática, el segundo es Economista y también tiene en torno a 10 años de experiencia en la gestión de empresas.

El equipo de trabajo de la empresa está conformado de un grupo de diseño con tres personas, una de ellas diseñadora gráfica y las otras dos diseñadoras web. Un grupo de sistemas conformado por dos personas, y un grupo de desarrollo, el cual está compuesto por 2 jefes de proyecto, 3 analistas, 10 consultores senior, 25 consultores junior y 5 especialistas técnicos.

La empresa se encuentra situada en un edificio de oficinas en un polígono industrial a las afueras de la ciudad. Por la oficina se está pagando un alquiler. Los trabajadores de la empresa trabajan unas 40 horas semanales. Distribuidas de 9:00 a 18:00 de lunes a viernes.

La empresa está compuesta por tres departamentos, los cuales se concuerdan con los grupos de trabajo indicados anteriormente.

En el grupo de desarrollo se encuentran personas con diferentes perfiles técnicos, todos los desarrolladores tienen una base de Java ya que la mayoría de los desarrollos están relacionados con esta tecnología. También se cuenta con personas especialistas en diferentes Frameworks de Java como son MVC, JSF, Struts2, Spring, JEEE, otras personas son especialistas en Liferay. Hay un grupo de personas que también tienen conocimientos de PHP. Y todas las personas también tienen unos conocimientos al menos básicos en Maven ya que los proyectos son siempre desarrollados con esta tecnología, por lo que la empresa cuenta con su propio Repositorio de proyectos para Maven. Todo el desarrollo se gestiona y coordina entre los equipos con un gestor de versiones SVN.

La empresa cuenta con una impresora láser a color especialmente utilizada por el equipo de diseño, y dos impresoras láser en blanco y negro para el resto de trabajadores. Cada trabajador tiene un ordenador de sobremesa, disponiendo la empresa también de unos 3

portátiles que los trabajadores suelen utilizar cuando es necesario salir de la oficina para ir a ver a un cliente. En la empresa además del desarrollo de aplicaciones web, también se realiza todo el soporte de puesta y mantenimiento en la red de las aplicaciones, por ello se cuenta con un clúster de servidores los cuales dan servicio a todos las aplicaciones. También se ofrece un servicio de correo electrónico, y almacenamiento de la información en un clúster de Bases de Datos.

Los principales gastos de la empresa son los salarios de los empleados, los de la oficina (el alquiler, el agua, la luz…), el mantenimiento de los equipos de los empleados, los clúster de servidores web y de bases de datos, el material de oficina (papel, tóner para las impresoras, bolígrafos…), los desplazamientos que se han de realizar para las reuniones con los clientes y los teléfonos que hay en la oficina.

Este proyecto será realizado por un jefe de proyecto, un analista y 4 desarrolladores especialistas en Java. Además de la intervención de una persona de sistemas.

4.2 Recursos humanos

El equipo de desarrollo del proyecto está compuesto por los siguientes recursos humanos:

Nombre del recurso	Tipo	Iniciales	Grupo	Capacidad máxima	Tasa estándar
Jefe de Proyectos	Trabajo	JP		100%	70,00 €/hora
Programador 1	Trabajo	P1	Desarrolladores	100%	30,00 €/hora
Programador 2	Trabajo	P2	Desarrolladores	100%	30,00 €/hora
Programador 3	Trabajo	P3	Desarrolladores	100%	30,00 €/hora
Programador 4	Trabajo	P4	Desarrolladores	100%	30,00 €/hora
Analista	Trabajo	A		100%	50,00 €/hora
Personal de Sistemas	Trabajo	S		100%	40,00 €/hora

Tabla 4-1 Recursos humanos del proyecto

4.3 Planificación

En este apartado se va a detallar mediante un diagrama de Gantt y el listado de las tareas la planificación del proyecto, correspondiente a las principales tareas. Para la realización del mismo he utilizado el programa Microsoft Office Project 2013.

4.3.1 Diagrama de Gantt donde se pueden ver las principales tareas.

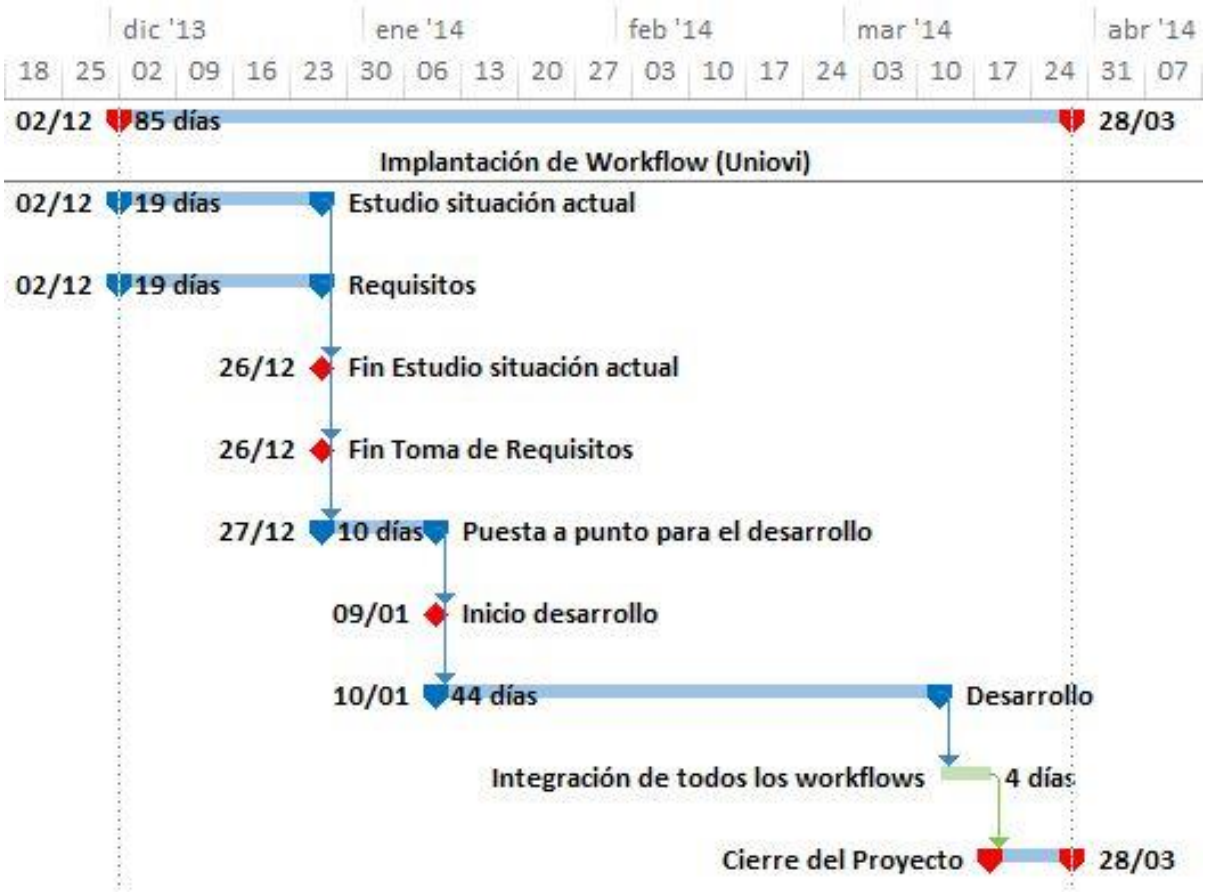


Figura 4.1 Planificación tareas principales

4.3.2 Diagrama de Gantt completo

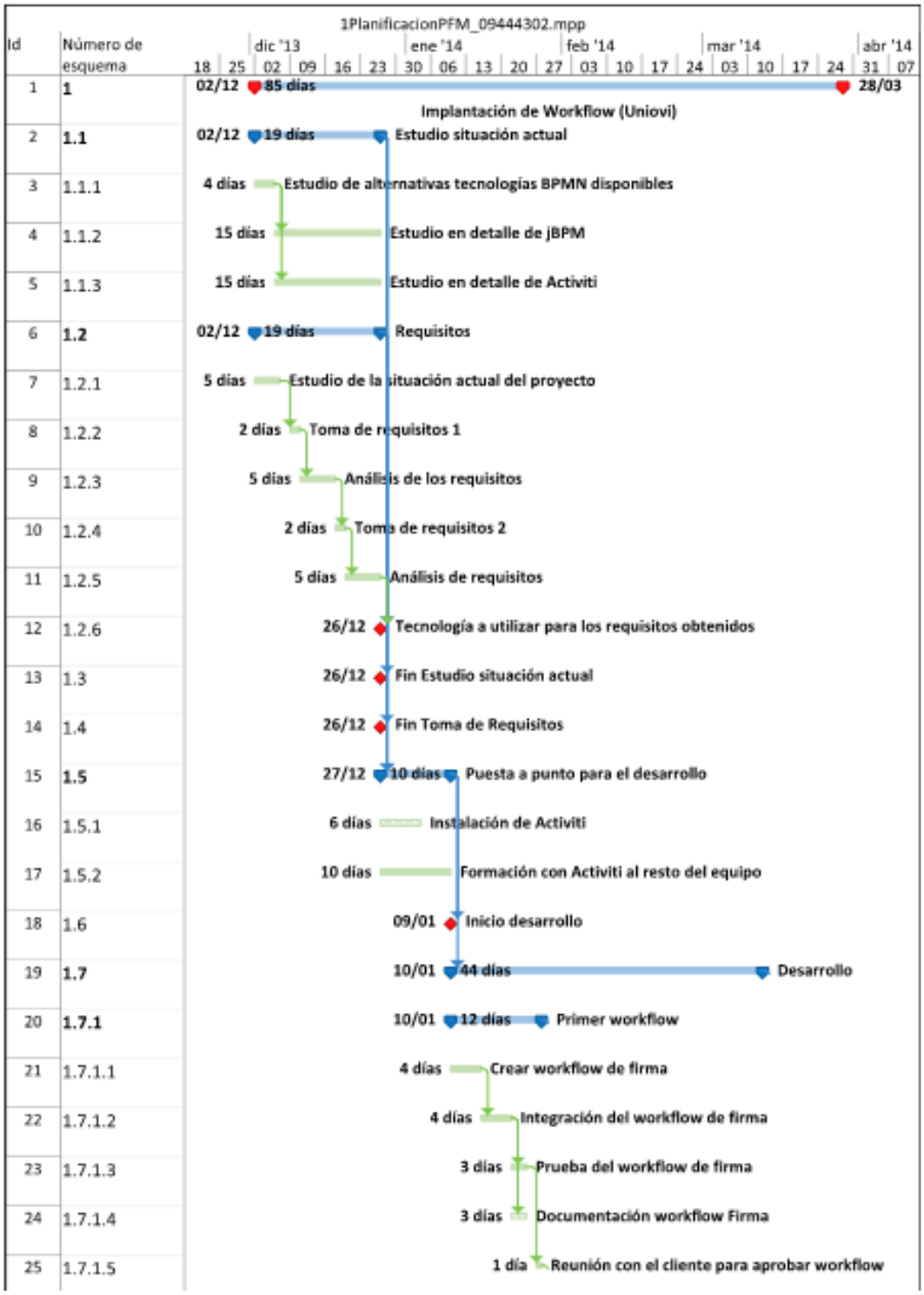


Figura 4.2 Planificación en detalle I

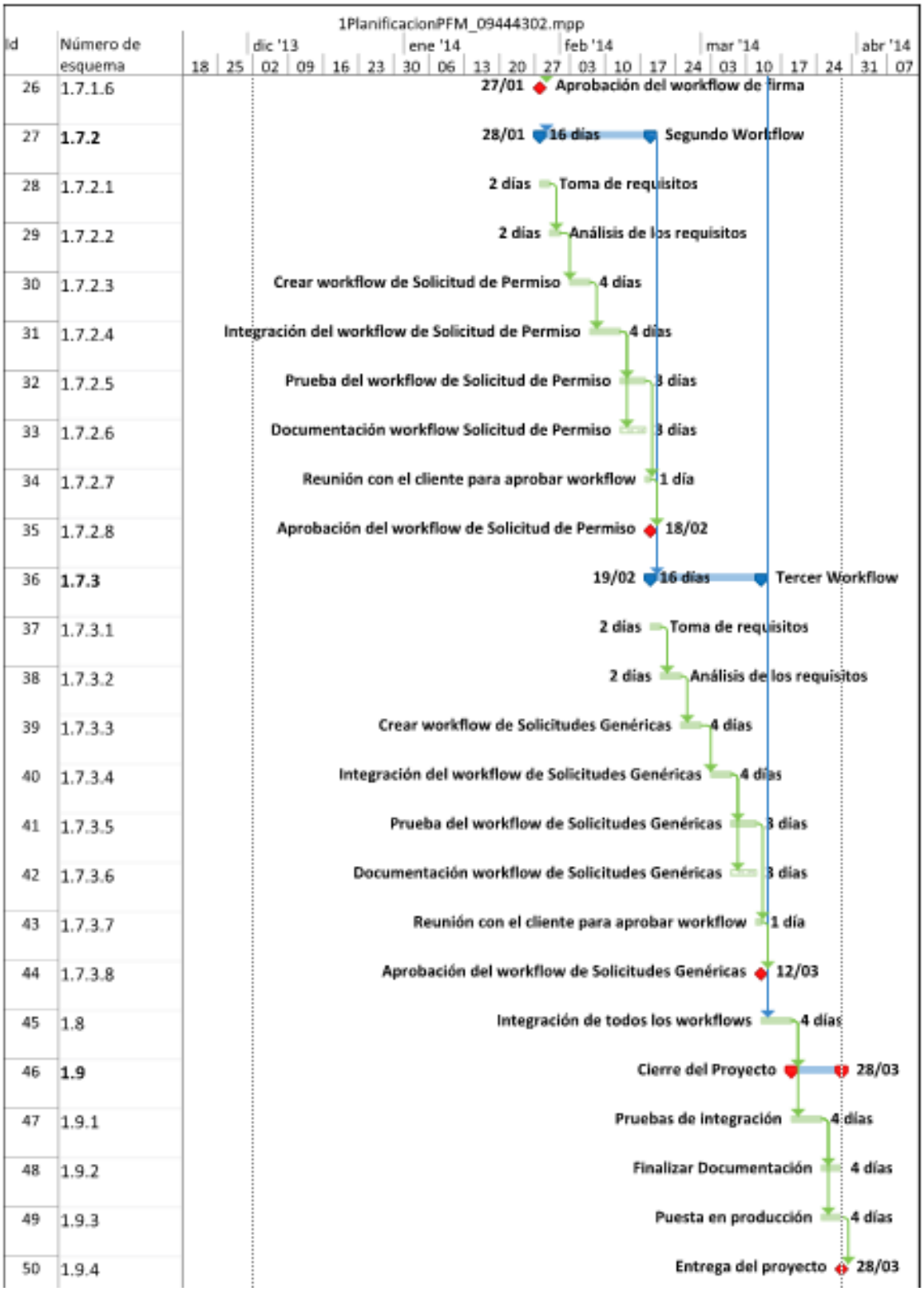


Figura 4.3 Planificación en detalle II

4.3.3 Detalle de las principales tareas

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	**Implantación de Workflow (Uniovi)**	**85 días**	**lun 02/12/13**	**vie 28/03/14**	
1.1	**Estudio situación actual**	**19 días**	**lun 02/12/13**	**jue 26/12/13**	
1.2	**Requisitos**	**19 días**	**lun 02/12/13**	**jue 26/12/13**	
1.3	Fin Estudio situación actual	0 días	jue 26/12/13	jue 26/12/13	2
1.4	Fin Toma de Requisitos	0 días	jue 26/12/13	jue 26/12/13	6
1.5	**Puesta a punto para el desarrollo**	**10 días**	**vie 27/12/13**	**jue 09/01/14**	**6;2**
1.6	Inicio desarrollo	0 días	jue 09/01/14	jue 09/01/14	15
1.7	**Desarrollo**	**44 días**	**vie 10/01/14**	**mié 12/03/14**	**15**
1.8	Integración de todos los workflows	4 días	jue 13/03/14	mar 18/03/14	19
1.9	**Cierre del Proyecto**	**8 días**	**mié 19/03/14**	**vie 28/03/14**	**45**

Tabla 4-2 Principales tareas

4.3.4 Listado de todas las tareas

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	**Implantación de Workflow (Uniovi)**	**85 días**	**lun 02/12/13**	**vie 28/03/14**		
1.1	**Estudio situación actual**	**19 días**	**lun 02/12/13**	**jue 26/12/13**		
1.1.1	Estudio de alternativas tecnologías BPMN disponibles	4 días	lun 02/12/13	jue 05/12/13		Programador 1; Programador 2; Programador 3; Programador 4
1.1.2	Estudio en detalle de jBPM	15 días	vie 06/12/13	jue 26/12/13	3	Programador 1; Programador 2
1.1.3	Estudio en detalle de Activiti	15 días	vie 06/12/13	jue 26/12/13	3	Programador 3; Programador 4
1.2	**Requisitos**	**19 días**	**lun 02/12/13**	**jue 26/12/13**		

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1.2.1	Estudio de la situación actual del proyecto	5 días	lun 02/12/13	vie 06/12/13		Analista
1.2.2	Toma de requisitos 1	2 días	lun 09/12/13	mar 10/12/13	7	Jefe de Proyectos; Analista
1.2.3	Análisis de los requisitos	5 días	mié 11/12/13	mar 17/12/13	8	Analista
1.2.4	Toma de requisitos 2	2 días	mié 18/12/13	jue 19/12/13	9	Analista; Jefe de Proyectos
1.2.5	Análisis de requisitos	5 días	vie 20/12/13	jue 26/12/13	10	Analista
1.2.6	Tecnología a utilizar para los requisitos obtenidos	0 días	jue 26/12/13	jue 26/12/13	2;11	Analista; Jefe de Proyectos; Programador 1; Programador 2; Programador 3; Programador 4
1.3	Fin Estudio situación actual	0 días	jue 26/12/13	jue 26/12/13	2	
1.4	Fin Toma de Requisitos	0 días	jue 26/12/13	jue 26/12/13	6	
1.5	**Puesta a punto para el desarrollo**	**10 días**	**vie 27/12/13**	**jue 09/01/14**	**6;2**	
1.5.1	Instalación de Activiti	6 días	vie 27/12/13	vie 03/01/14		Programador 1[50%]; Programador 2[50%]; Programador 3; Programador 4
1.5.2	Formación con Activiti al resto del equipo	10 días	vie 27/12/13	jue 09/01/14		Programador 1[50%]; Programador 2[50%]
1.6	Inicio desarrollo	0 días	jue 09/01/14	jue 09/01/14	15	
1.7	**Desarrollo**	**44 días**	**vie 10/01/14**	**mié 12/03/14**	**15**	
1.7.1	**Primer workflow**	**12 días**	**vie 10/01/14**	**lun 27/01/14**		
1.7.1.1	Crear workflow de firma	4 días	vie 10/01/14	mié 15/01/14		Programador 2; Programador 3; Programador 4
1.7.1.2	Integración del workflow de firma	4 días	jue 16/01/14	mar 21/01/14	21	Programador 1; Programador 3; Programador 4

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1.7.1.3	Prueba del workflow de firma	3 días	mié 22/01/14	vie 24/01/14	22	Programador 1; Programador 2
1.7.1.4	Documentación workflow Firma	3 días	mié 22/01/14	vie 24/01/14	22	Programador 3; Programador 4
1.7.1.5	Reunión con el cliente para aprobar workflow	1 día	lun 27/01/14	lun 27/01/14	23	Analista; Jefe de Proyectos
1.7.1.6	Aprobación del workflow de firma	0 días	lun 27/01/14	lun 27/01/14	25	
1.7.2	**Segundo Workflow**	**16 días**	**mar 28/01/14**	**mar 18/02/14**	**20**	
1.7.2.1	Toma de requisitos	2 días	mar 28/01/14	mié 29/01/14		Analista; Jefe de Proyectos
1.7.2.2	Análisis de los requisitos	2 días	jue 30/01/14	vie 31/01/14	28	Analista
1.7.2.3	Crear workflow de Solicitud de Permiso	4 días	lun 03/02/14	jue 06/02/14	29	Programador 2; Programador 3; Programador 4
1.7.2.4	Integración del workflow de Solicitud de Permiso	4 días	vie 07/02/14	mié 12/02/14	30	Programador 1; Programador 2; Programador 3; Programador 4
1.7.2.5	Prueba del workflow de Solicitud de Permiso	3 días	jue 13/02/14	lun 17/02/14	31	Programador 1; Programador 2
1.7.2.6	Documentación workflow de Solicitud de Permiso	3 días	jue 13/02/14	lun 17/02/14	31	Programador 3; Programador 4
1.7.2.7	Reunión con el cliente para aprobar workflow	1 día	mar 18/02/14	mar 18/02/14	32	Analista; Jefe de Proyectos
1.7.2.8	Aprobación del workflow de Solicitud de Permiso	0 días	mar 18/02/14	mar 18/02/14	34	
1.7.3	**Tercer Workflow**	**16 días**	**mié 19/02/14**	**mié 12/03/14**	**27**	
1.7.3.1	Toma de requisitos	2 días	mié 19/02/14	jue 20/02/14		Analista; Jefe de Proyectos
1.7.3.2	Análisis de los requisitos	2 días	vie 21/02/14	lun 24/02/14	37	Analista
1.7.3.3	Crear workflow de Solicitudes Genéricas	4 días	mar 25/02/14	vie 28/02/14	38	Programador 1;Programador 2;Programador 3;Programador 4

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1.7.3.4	Integración del workflow de Solicitudes Genéricas	4 días	lun 03/03/14	jue 06/03/14	39	Programador 1;Programador 2;Programador 3;Programador 4
1.7.3.5	Prueba del workflow de Solicitudes Genéricas	3 días	vie 07/03/14	mar 11/03/14	40	Programador 1;Programador 2
1.7.3.6	Documentación workflow de Solicitudes Genéricas	3 días	vie 07/03/14	mar 11/03/14	40	Programador 3;Programador 4
1.7.3.7	Reunión con el cliente para aprobar workflow	1 día	mié 12/03/14	mié 12/03/14	41	Analista; Jefe de Proyectos
1.7.3.8	Aprobación del workflow de Solicitudes Genéricas	0 días	mié 12/03/14	mié 12/03/14	43	
1.8	Integración de todos los workflows	4 días	jue 13/03/14	mar 18/03/14	19	Analista; Personal de Sistemas; Programador 1; Programador 2
1.9	**Cierre del Proyecto**	**8 días**	**mié 19/03/14**	**vie 28/03/14**	**45**	
1.9.1	Pruebas de integración	4 días	mié 19/03/14	lun 24/03/14	45	Programador 1; Programador 3; Programador 4
1.9.2	Finalizar Documentación	4 días	mar 25/03/14	vie 28/03/14	47	Programador 3; Programador 4
1.9.3	Puesta en producción	4 días	mar 25/03/14	vie 28/03/14	47	Jefe de Proyectos; Personal de Sistemas; Analista; Programador 1; Programador 2
1.9.4	Entrega del proyecto	0 días	vie 28/03/14	vie 28/03/14	49	

Tabla 4-3 Listado completo de todas las tareas

4.3.5 Descripción planificación

En los diagramas y listados de tareas del punto anterior, se puede observar que la duración del proyecto ha sido de 85 días. Divididos en un total de 50 tareas.

Las principales tareas son 8, las cuales engloban y distribuyen las principales cargas de trabajo. Para empezar hay una tarea de Requisitos y otra de Estudio situación actual previas en las cuales se ha estudiado el mercado y comprobado las opciones que este ofrecía, una parte del trabajo se puede ver en el apartado de "Estudio de la Situación Actual", también se realizó la toma de y análisis de requisitos.

Una vez conocida la situación actual y adquirida una cierta formación sobre las tecnologías a utilizar comencé a trabajar para poner a punto el desarrollo propio del proyecto. El desarrollo del proyecto está básicamente dividido en tres partes, una por cada workflow que se ha diseñado e implantado.

Cada una de estas tres partes está compuesta por varias subtareas de creación, integración, documentación y aprobación del workflow.

Una vez completados los tres workflows desarrollados se realizó una tarea de integración de todos los workflows.

Para finalizar el proyecto se realizaron pruebas de integración, se finalizó la documentación y se puso a funcionar todo el proyecto.

4.4 Resumen del Presupuesto

En este apartado se puede ver el presupuesto resumido, este sería el presupuesto que le entregaría al cliente, por lo que las partidas se encuentran agrupadas en función de lo que el cliente percibe.

Para ver un detalle más amplio del presupuesto se puede recurrir al Capítulo 11.

Item	Concepto	Precio unitario	Cantidad	Total
1	Desarrollo Implantación Workflows	92.434,00 €	1	92.434,00 €100,00 €
2	Implantación	830,00 €	1	830,00 €
3	Formación	6.000,00 €	1	6.000,00 €
4	Desplazamientos	20,00 €	5	100,00 €
	IVA 21%	99.364,00 €	21%	20.866,44 €
	Total sin IVA			99.364,00 €
	TOTAL			**120.230,44 €**

Tabla 4-4 Presupuesto del Cliente

Capítulo 5. Análisis

Este apartado contendrá toda la especificación de requisitos y toda la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño.

5.1 Definición del Sistema

5.1.1 Determinación del Alcance del Sistema

Este proyecto se basa en el estudio de los trámites que se realizan en la Universidad de Oviedo, para posteriormente introducir un sistema de gestión de workflows con el que poder hacer mejoras en el rendimiento del negocio y simplificación de los procesos.

El BPM que se ha decidido implantar después de realizar el Estudio de la Situación Actual ha sido Activiti, éste ha de integrarse dentro del gestor de trámites que la universidad ya posee.

Con la herramienta Activiti se podrán definir nuevos procesos o modificarlos de una forma mucho más sencilla y sin tener que realizar nuevas versiones del gestor de trámites. Con lo que ello implica no tener que parar esta aplicación y por tanto el servicio que se les ofrece a los clientes.

Activiti ha sido integrado de forma que los procesos se pueden monitorizar desde el propio gestor de trámites de manera que se puede ver los procesos en curso y los procesos terminados.

Algunas de las tareas realizadas anteriormente de forma manual como la asignación de tareas, o envío de email que se tenía que realizar de forma manual, o a través de código propio ahora se puede realizar a través de Activiti de una forma mucho más automática y configurable.

5.2 Requisitos del Sistema

5.2.1 Obtención de los Requisitos del Sistema

El objetivo principal del sistema como ya se ha comentado anteriormente es la integración de Activiti con el gestor de trámites de la Universidad de Oviedo. Esto conlleva un estudio de los trámites que actualmente se gestionan y cuales sería posible gestionar en un futuro.

En el tramitador existente en la Universidad de Oviedo ya existe una capa de abstracción de los procesos de tramitación y BPM, ya que anteriormente se utilizaba JBPM para realizar un proceso, por ello lo primero que hay que realizar es la capa correspondiente a Activiti. Una vez realizada esta parte, se estudió, mejoró y cambió el proceso existente y que funcionaba con JBPM para que funcionara con Activiti. Posteriormente, se hizo un estudio de las necesidades del sistema y procedieron a realizar dos nuevos procesos con Activiti he integrarlos en todo el funcionamiento actual de tramitador de solicitudes de la Universidad.

5.2.1.1 Requisitos funcionales

Código	Nombre Requisito	Descripción del Requisito
RF1.1	Capa de abstracción de Activiti	Se ha de realizar una capa que permita la abstracción del tramitador de Activiti
RF1.2	Incluir la capa de abstracción de Activiti en el tramitador	Se ha incluir la capa de abstracción en el tramitador para poder trabajar con Activiti
RF1.3	Workflow para el proceso de firma	Modificación del proceso de firma para utilizar Activiti
RF1.4	Estudio de la situación actual	Se estudian los posibles trámites a realizar utilizando Activiti
RF1.5	Creación del proceso Solicitud de Permiso	Creación del workflow con Activiti
RF1.6	Inclusión del proceso de Solicitud de Permiso	Inclusión en el tramitador del nuevo workflow
RF1.7	Creación convocatoria para Solicitud de Permiso	Crear la convocatoria con la configuración del nuevo workflow
RF1.8	Creación del proceso de Solicitudes Genéricas	Creación del workflow con Activiti
RF1.9	Inclusión del proceso de Solicitudes Genéricas	Inclusión en el tramitador del nuevo workflow
RF1.10	Creación convocatoria para Solicitud de Permiso	Crear la convocatoria con la configuración del nuevo workflow
RF.1.11	Diseñar workflow	Se puede diseñar los workflows y desplegarlos
RF1.12	Asignar los workflows a las convocatorias	Cuando se crea una convocatoria se le asigna el workflow correspondiente

Tabla 5-1 Requisitos funcionales

5.2.1.2 Requisitos no funcionales

Código	Nombre Requisito	Descripción del Requisito
RNF1.1	Lenguaje de desarrollo	El lenguajes utilizado es Java
RNF1.2	Sistema gestos de base de datos	El producto utilizado será PostgreSQL
RNF1.3	BPMN utilizado	El BPMN que se va a integrar es Activiti

Tabla 5-2 Requisitos no funcionales

5.2.2 Identificación de Actores del Sistema

5.2.2.1 Administrador

Un usuario administrador es el encargado de crear las convocatorias para los procesos, y de monitorizar y tramitar los procesos.

Sus funciones son:

- Crear las convocatorias desde la herramienta de Solicitudes Genéricas
- Monitorizar las solicitudes desde el Tramitador
- Resolver las solicitudes:
  - Aceptar las solicitud
  - Solicitar una subsanación al usuario
  - Rechazar la solicitud

5.2.2.2 Usuario

Los usuarios que no son administradores pueden realizar solicitudes de las convocatorias disponibles.

Sus funciones son:

- Crear una nueva solicitud de una convocatoria disponible
- Consultar el estado de una solicitud realizada a través de “Cómo va lo mío”, en la Intranet de Uniovi
- Resolver una subsanación que ha sido pedida por parte del administrador

5.2.2.3 Administrador del Sistema

Los usuarios administradores del sistema, que no son los administradores de las administraciones públicas, pueden realizar las siguientes operaciones:

- Diseñar workflows para los procedimientos deseados
- Asignar workflows a las convocatorias

5.2.3 Especificación de Casos de Uso

5.2.3.1 Administrador

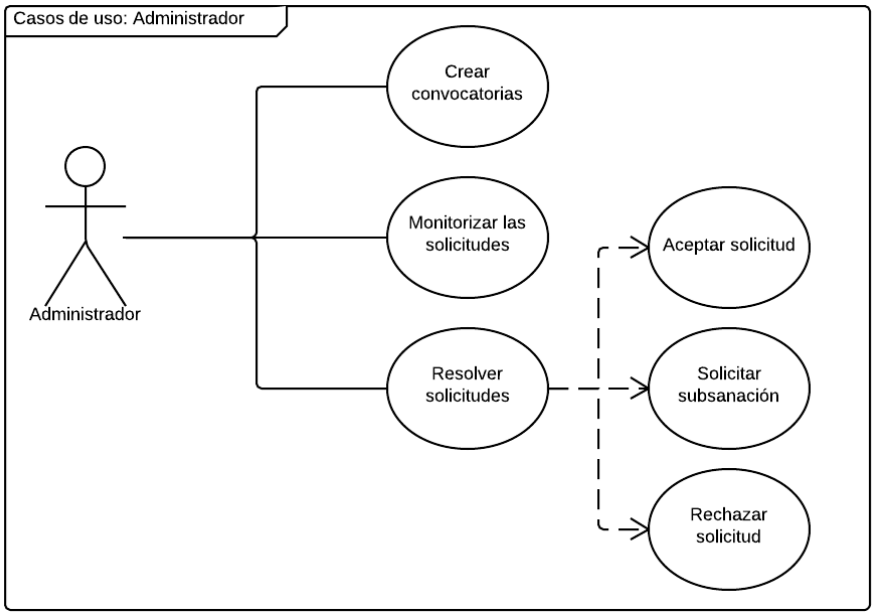


Figura 5.1 Casos de uso: Administrador

Nombre del Caso de Uso
Monitorizar las solicitudes
Descripción
Los administradores desde el tramitador pueden consultar en qué estado se encuentra cada solicitud, en qué estado del workflow.

Nombre del Caso de Uso
Aceptar Solicitud
Descripción
Los administradores pueden consultar las solicitudes y si consideran que está todo correcto pueden aprobarlas, haciendo así que finalice el workflow de esa solicitud.

Nombre del Caso de Uso
Rechazar Solicitud
Descripción
Los administradores pueden consultar las solicitudes y si consideran que no está todo correcto pueden rechazarlas, haciendo así que finalice el workflow de esa solicitud.

Nombre del Caso de Uso
Solicitar Subsanación
Descripción
Los administradores en caso de detectar algún problema en una solicitud pueden solicitar al usuario una subsanación, de esta forma hacen avanzar el workflow pero no que este finalice.

Nombre del Caso de Uso
Crear convocatorias
Descripción
Cuando los administradores crean una convocatoria para realizar cualquier tipo de gestión, se almacena en base de datos el workflows que se va a utilizar para todas las solicitudes de esa convocatoria.

5.2.3.2 Usuario

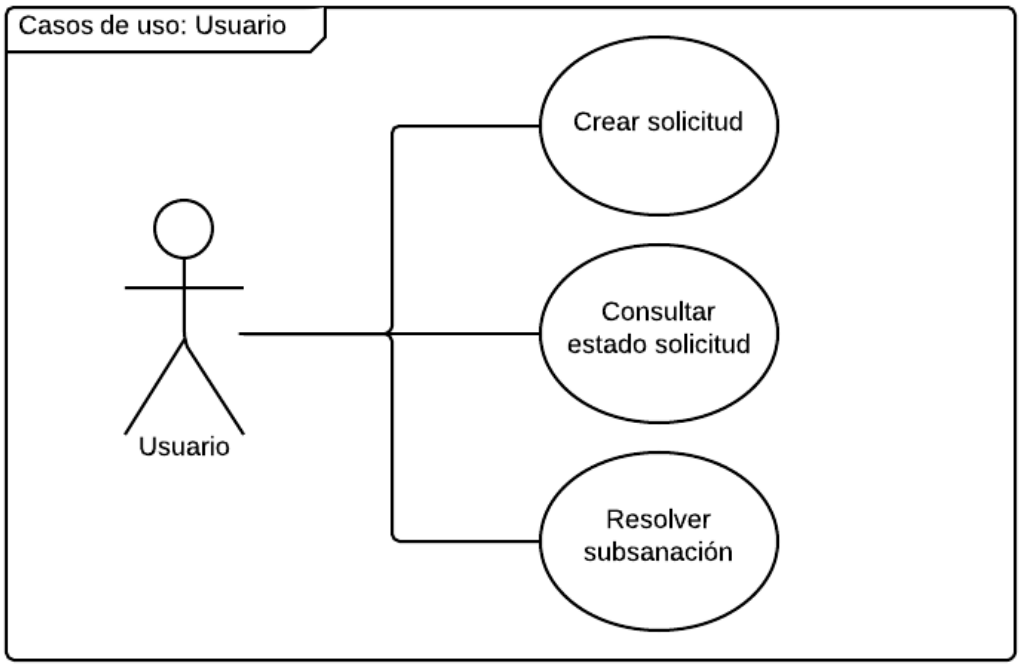


Figura 5.2 Casos de uso: Usuario

Nombre del Caso de Uso
Crear solicitud
Descripción
Los usuarios podrán crear una solicitud de una convocatoria existente, iniciando de esta manera una nueva instancia del workflow asociado a esa convocatoria

Nombre del Caso de Uso
Consultar estado Solicitud
Descripción
Los usuario podrán consultar el estado en el que se encuentra su solicitud, el estado del workflow en el que se encuentra

Nombre del Caso de Uso
Resolver subsanación
Descripción
Un usuario si se le requiere una subsanación en el la solicitud, puede realizar y de esta forma hacer avanzar el workflow.

5.2.3.3 Administrador Del Sistema

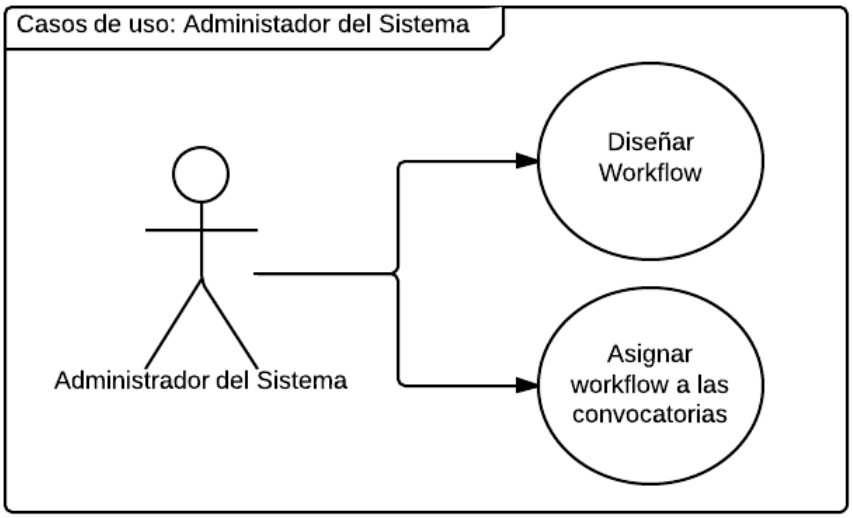


Figura 5.3 Casos de uso: Administrador del Sistema

Nombre del Caso de Uso
Diseñar workflow
Descripción
Los administradores del sistema pueden diseñar y desplegar nuevos workflow para los procedimientos

Nombre del Caso de Uso
Asignar workflow a las convocatorias
Descripción
Los administradores del sistema cuando crean una convocatoria se le asigna un workflow

5.3 Identificación de los Subsistemas en la Fase de Análisis

5.3.1 Descripción de los Subsistemas

El sistema se puede dividir en tres partes principales:

- Gestor de trámites de la Universidad de Oviedo
- Subsistema para la definición de workflows.
- Subsistema para la integración de Activiti y con el gestor de trámites.

5.3.1.1 Gestor de trámites de la Universidad, eGob!

Funcionalmente la plataforma eGob! posee los siguientes elementos:

- Catálogo de Procedimientos. En esta base de datos se definen y parametrizan los aspectos relevantes de los procedimientos administrativos (por ejemplo, los actos administrativos que componen el procedimiento).
- Base de datos de Expedientes. En esta base de datos se almacenan los datos relativos a los expedientes que se están ejecutando.
- Gestor Documental. En el gestor documental se va incorporando la documentación administrativa que se genera durante la tramitación de los expedientes.
- Sistema gestor de expedientes. Se trata de la aplicación de usuario final a través de la cual pueden realizarse operaciones sobre los expedientes.
- Integración con el resto de componentes de la plataforma. Por ejemplo, la plataforma incorpora un módulo de publicación de expedientes, que permite la consulta de los expedientes desde internet.

5.3.1.2 Subsistema para la definición de workflows

Este subsistema es donde se crean los diagramas de los workflows y se despliegan en base de datos para posteriormente ser utilizados por las convocatorias y solicitudes.

5.3.1.3 Subsistemas para la integración de Activiti con el Gestor de Trámites

Subsistema compuesto a su vez por dos partes, una de ellas donde se definen las clases que sirven de modelo para pasar de Activiti al gestor de trámites, y una segunda parte donde se implementan las operaciones de la fachada de servicio de operaciones que posee el gestor de trámites para trabajar con los workflows.

5.3.2 Descripción de los Interfaces entre Subsistemas

La comunicación entre los diferentes subsistemas es diferente en función de cada uno de ellos.

- El subsistema de definición de workflows almacena en una tabla en base de datos, los datos relativos al workflow creado.
- El subsistema de integración de Activiti con eGob! Se comunica con el primero a través de los datos contenidos en base de datos, y con Activiti a través de la API que esta proporciona.
- La comunicación entre el subsistema de integración y el gestor de trámites se realiza a través de la fachada que implementa el subsistema de integración.

5.4 Diagrama de Clases Preliminar del Análisis

5.4.1 Diagrama de Clases

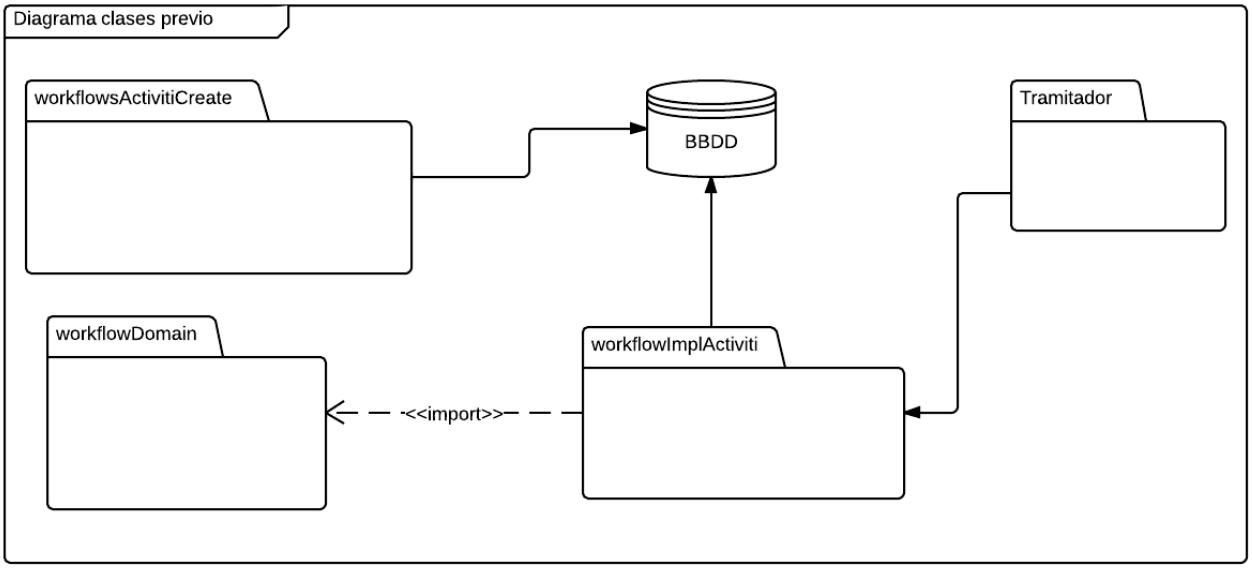


Figura 5.3. Diagrama de Clases Previo

5.4.2 Descripción de las Clases

5.4.2.1 Gestor de trámites de la Universidad, eGob!

Nombre de la Clase
Tramitador
Descripción
Se trama del gestor de trámites que tiene la Universidad de Oviedo
Responsabilidades
Interacción con los usuarios, para iniciar los workflows o hacer que estos avancen cuando dependen de tareas manuales.

5.4.2.2 Subsistema para la definición de workflows

Nombre de la Clase
workflowsActivitiCreate
Descripción
Se crean los diagramas de los workflows y se despliegan en base de datos
Responsabilidades
Es el encargado de crear en base de datos los workflows para poder ser asignados a convocatorias y posteriormente se inicien instancias de ellos con las solicitudes

5.4.2.3 Subsistemas para la integración de Activiti con el Gestor de Trámites

Nombre de la Clase
workflowDomain
Descripción
Define las clase que se utilizan para trabajar con tareas y workflows desde el gestor de trámites
Interfaces Propuestos
WorkflowService: define las operaciones que se pueden realizar desde el gestor de trámites con Activiti

Nombre de la Clase
workflowImplActiviti
Descripción
Implementa el Interfaz de workflowDomain
Responsabilidades
Realiza todas las operaciones necesarias con los workflows
Métodos Propuestos
cancelWorkflow: finaliza la instancia del workflow que se le indica **createTask**: crea una tarea de Activiti a partir de la tarea del modelo recibida **delegateTask**: asigna la tarea recibida al usuario indicado **gestTasks**: devuelve las tareas del workflow y del creador que se indican en el criterio de búsqueda **getVariable**: devuelve la variable del workflow que se solicita **getWorkflow**: devuelve la instancia del workflow al que pertenece el identificador **getWorkflowRunTask**: devuelve la tarea que se está ejecutando en ese momento en la instancia del workflow que se indica **closeTask**: finaliza la tarea que se indica, añadiendo los parámetros que recibe **getWorkflowTasks**: devuelve una lista con todas las tareas del workflow que se le indica **setVariable**: añade la variable que se indica en la instancia del workflow indicado **startWorkflow**: crea una nueva instancia del workflow que se indica y añade los argumentos recibidos **updateTask**: modifica la tarea de Activiti con los datos recibidos en la tarea del modelo

5.5 Análisis de Casos de Uso y Escenarios

5.5.1 Casos de Uso: Administrador

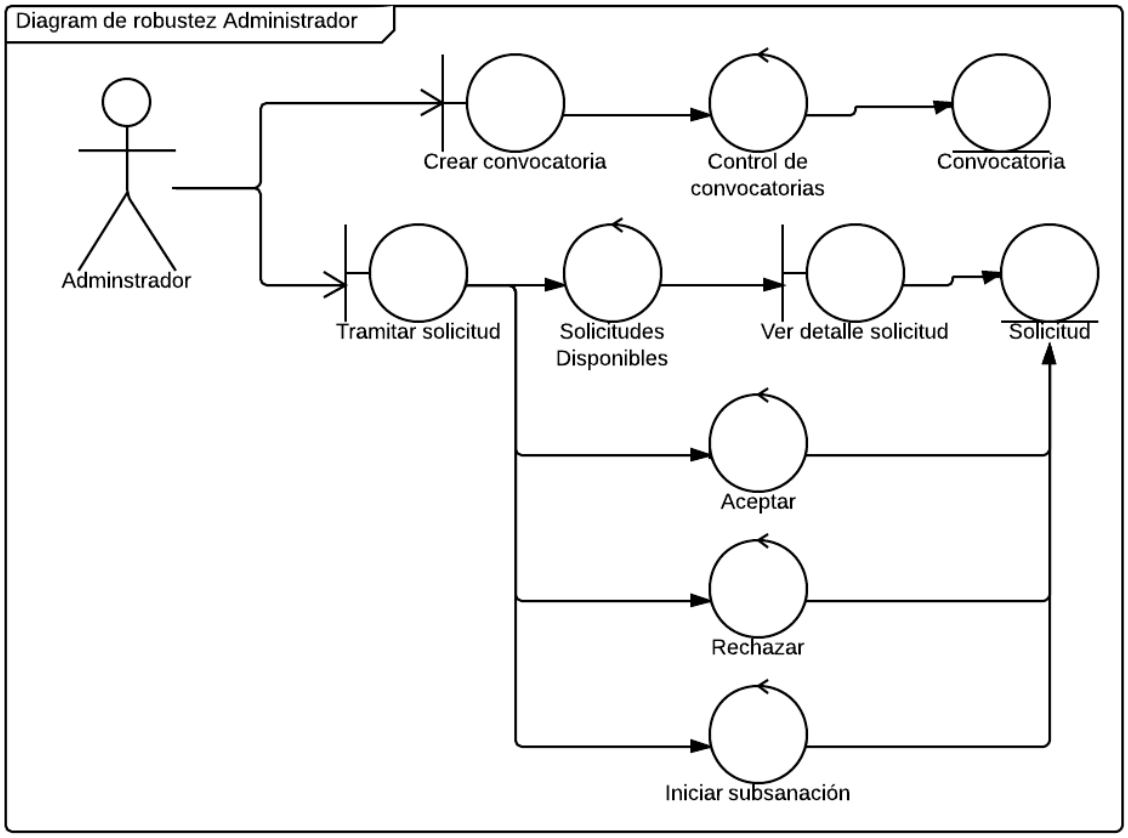


Figura 5.4 Diarama de robustez: Administrador

5.5.1.1 Crear convocatorias

Crear convocatorias	
Precondiciones	Debe de existir un workflow creado y desplegado para asignar por defecto a la convocatoria
Poscondiciones	Una convocatoria de la que es posible crear solicitudes por los usuarios que se ha indicado
Actores	Administrador
Descripción	El administrador accede a la creación de solicitudes genéricas y crea una nueva convocatoria con el workflow prestablecido para ello
Excepciones	Si no está definido el workflow por defecto, las solicitudes de esta convocatoria no tendrían asociado un procedimiento

5.5.1.2 Monitorizar las solicitudes

Monitorizar las solicitudes	
Precondiciones	Deben existir solicitudes realizadas
Poscondiciones	Listado de todas las solicitudes realizadas, por una parte las no finalizadas y por otra todas las realizadas
Actores	Administrador
Descripción	El administrador accede al tramitador de solicitudes y allí puede verlas todas, de las cuáles es el encargado de supervisar, agrupadas por la convocatoria a la que pertenecen
Excepciones	Si no existen solicitudes este listado será vacío

5.5.1.3 Aceptar solicitud

Aceptar solicitud	
Precondiciones	Debe existir una solicitud presentada y no finalizada
Poscondiciones	La solicitud se da por aceptada y finalizada
Actores	Administrador
Descripción	El administrador accede al tramitador y tras revisar que toda la solicitud esté correcta decide aceptarla.

5.5.1.4 Rechazar solicitud

Rechazar solicitud	
Precondiciones	Debe existir una solicitud presentada y no finalizada
Poscondiciones	La solicitud se da por rechazada y finalizada
Actores	Administrador
Descripción	El administrador accede al tramitador y tras revisar toda la solicitud decide que está no cumple con los requisitos y la rechaza

5.5.1.5 Iniciar subsanación

Iniciar subsanación	
Precondiciones	Debe existir una solicitud presentada y no finalizada
Poscondiciones	La solicitud se queda pendiente de subsanación por parte del usuario y no finalizada
Actores	Administrador
Descripción	El administrador accede al tramitador y tras revisar toda la solicitud ve que a ésta le falta algún dato y solicita al usuario que subsane el error correspondiente

5.5.2 Casos de Uso: Usuario

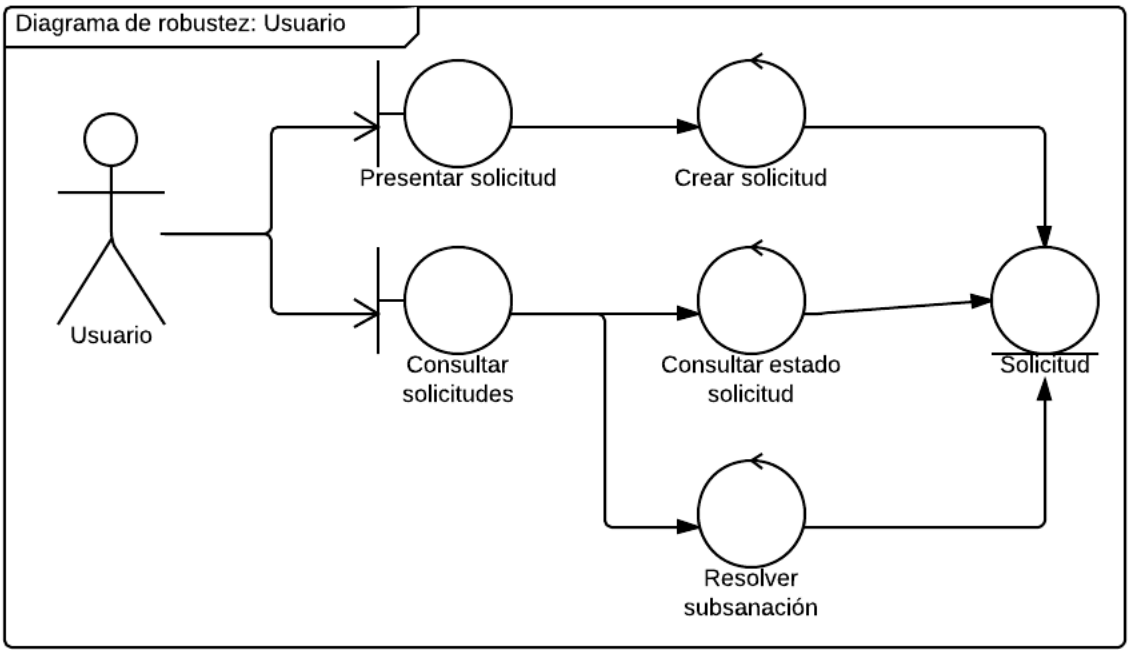


Figura 5.5 Diagrama de robustez: Usuario

5.5.2.1 Crear solicitud

Crear solicitud	
Precondiciones	El usuario debe de tener disponible una convocatoria para su perfil
Poscondiciones	Debe existir una nueva solicitud de ese usuario en la convocatoria seleccionada
Actores	Usuario
Descripción	El usuario accede a Presentar Solicitudes y selecciona la convocatoria deseada. Realiza todos los pasos de la creación de solicitud aportando la documentación requerida.
Excepciones	La convocatoria no tiene asociado el workflow correspondiente y por tanto no se inicia la solicitud

5.5.2.2 Consultar estado solicitud

Consultar estado solicitud	
Precondiciones	El usuario debe de haber realizado alguna solicitud para poder ver su estado
Poscondiciones	Listado de todos los pasos por los que ha pasado la solicitud y le afectan al usuario
Actores	Usuario
Descripción	El usuario accede a Consultar solicitudes (Cómo va lo mío), y ve por qué pasos del workflow, que le afecten al estado, ha pasado su solicitud
Excepciones	No se ha iniciado de forma correcta el workflow y no se puede ver en qué estado se encuentra la solicitud

5.5.2.3 Resolver subsanación

Resolver subsanación	
Precondiciones	El usuario debe de tener una solicitud realizada y con estado subsanación iniciada
Poscondiciones	El estado de la solicitud para a estar de nuevo presentada
Actores	Usuario
Descripción	El usuario recibe un email o accede a Consultar solicitudes (Cómo va lo mío), y ve una solicitud en estado de iniciar subsanación, entonces realiza la subsanación indicando los datos que le han sido requeridos

5.5.3 Casos de uso: Administrador del Sistema

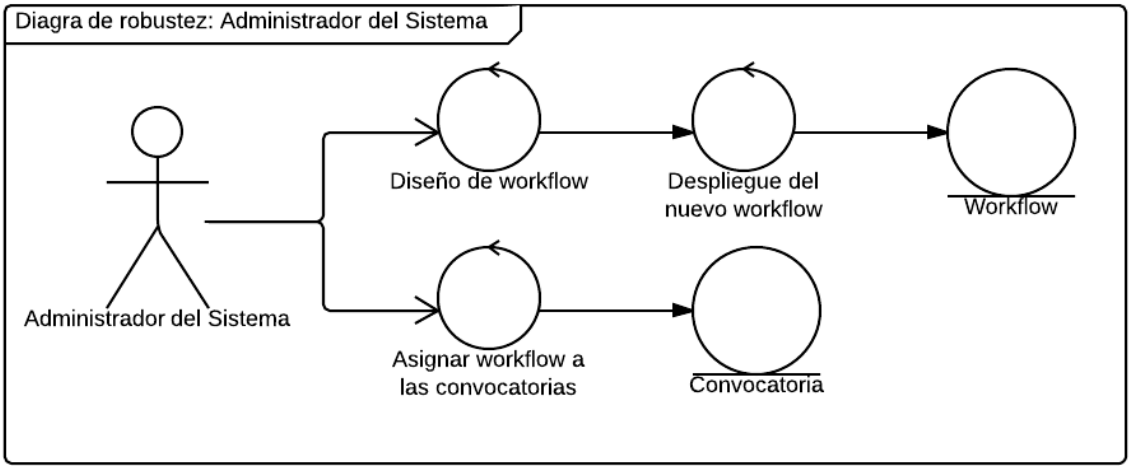


Figura 5.6 Diagrama de robustez: Adminstrador del Sistema

5.5.3.1 Diseñar y asignar workflow a las convocatorias

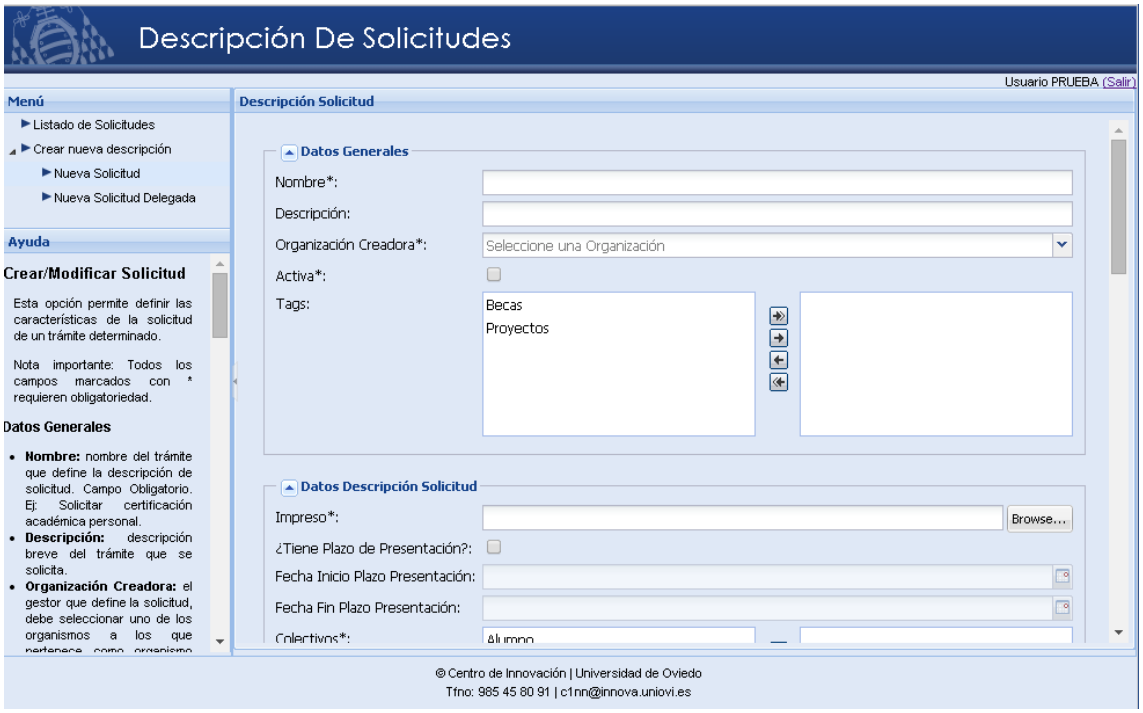
Diseñar workflow. Asignar workflow a las convocatorias	
Poscondiciones	Se ha creado y desplegado un nuevo workflow disponible para ser utilizado
Actores	Administrador del Sistema
Descripción	El administrador desde Eclipse diseña un nuevo workflow y posteriormente lo despliega. Una vez creado se lo asigna a la convocatoria deseada
Variaciones (escenarios secundarios)	Si todavía no existe la convocatoria se queda desplegado pero ninguna solicitud va a generar instancias de éste workflow. También se puede asignar a las nuevas convocatorias que se generen
Excepciones	Si se despliega un workflow con el mismo identificador que otro ya existente, automáticamente Activiti cuando se solicite una nueva instancia de éste workflow se proporciona el último desplegado

5.6 Análisis de Interfaces de Usuario

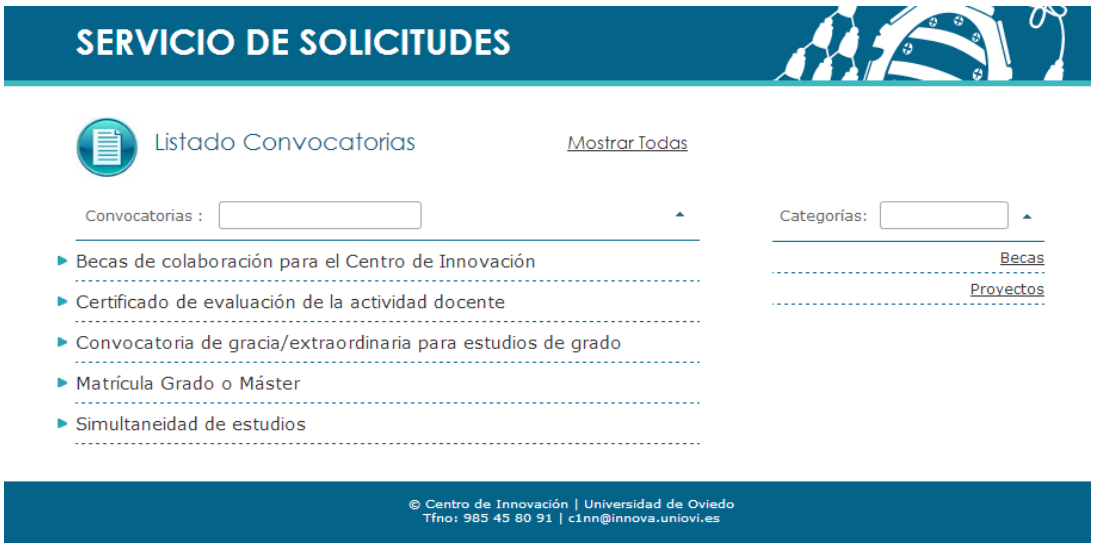
Debido a la naturaleza del proyecto, puesto que se trata un desarrollo para mejorar y ampliar la funcionalidad de la plataforma eGob!, no se definen interfaces con el usuario.

Estas interfaces ya se encuentran en dicha plataforma. Ya se proporciona una plataforma para:

- Crear las convocatorias "Solicitudes Genéricas"



- Realizar solicitudes "Presentar Solicitud"



- Consultar el estado de las solicitudes del usuario “Cómo va lo mío”

- Gestión de las solicitudes “Tramitador”

5.7 Especificación del Plan de Pruebas

5.7.1 Pruebas Unitarias

Para el desarrollo de las pruebas unitarias he utilizado JUnit lo cual me permitió automatizar las pruebas.

5.7.2 Pruebas de Integración

Para realizar las pruebas de integración ya que mi proyecto está integrado en otro sistema he creado alguna de las clases las cuales tenía que ejecutar el workflow y las ejecuté desde el propio subsistema de creación de workflows para probar que las llamadas y ejecución eran correctas.

5.7.3 Pruebas del Sistema

Para realizar las pruebas del sistema es necesario tener disponible la plataforma eGob! y realizar las operaciones desde este interfaz comprobando que mis subsistemas realizan correctamente las operaciones y se llevan a cabo los procesos completos.

5.7.4 Pruebas de Usabilidad

Las pruebas de usabilidad no tienen relación con este proyecto, ya que la interacción del usuario es a través de los interfaces disponibles en la plataforma eGob!

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

6.1.1 Diagramas de Paquetes

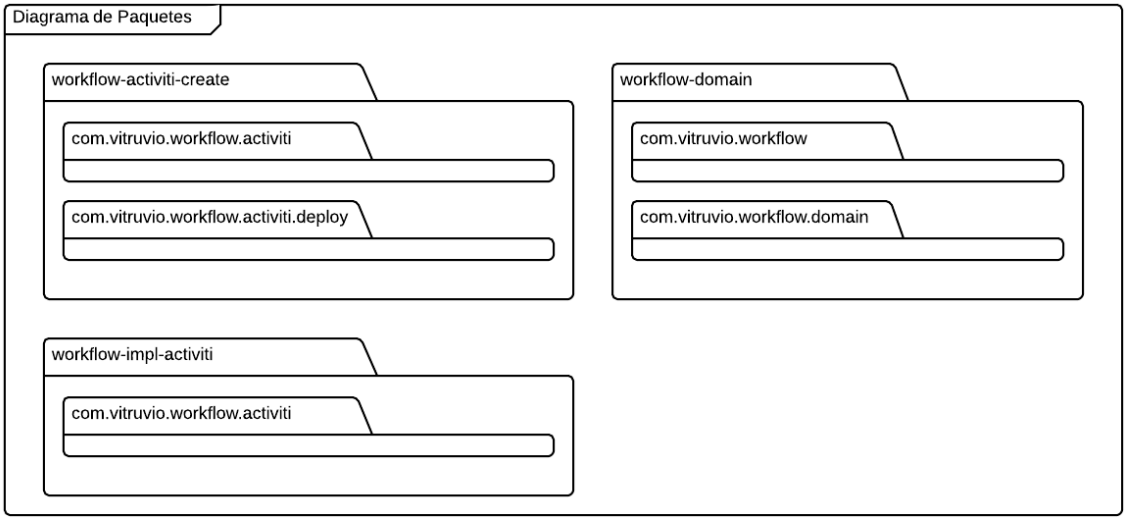


Figura 6.1 Diagrama de Paquetes

6.1.1.1 Paquete workflow-activiti-create

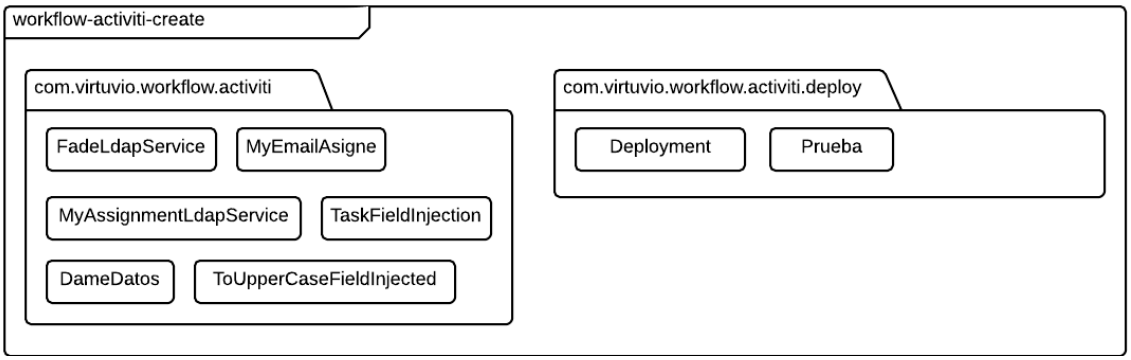


Figura 6.2 Diagrama de paquetes: activiti-create

El paquete **workflow-activiti-create** está compuesto por el paquete ***deploy*** que es el paquete en con el cuál se despliegan los workflows diseñados, también está compuesto por el paquete ***activiti*** el cual contiene las clases se utilizaron de prueba para poder realizar pruebas de integración.

6.1.1.2 Paquete workflow-domain

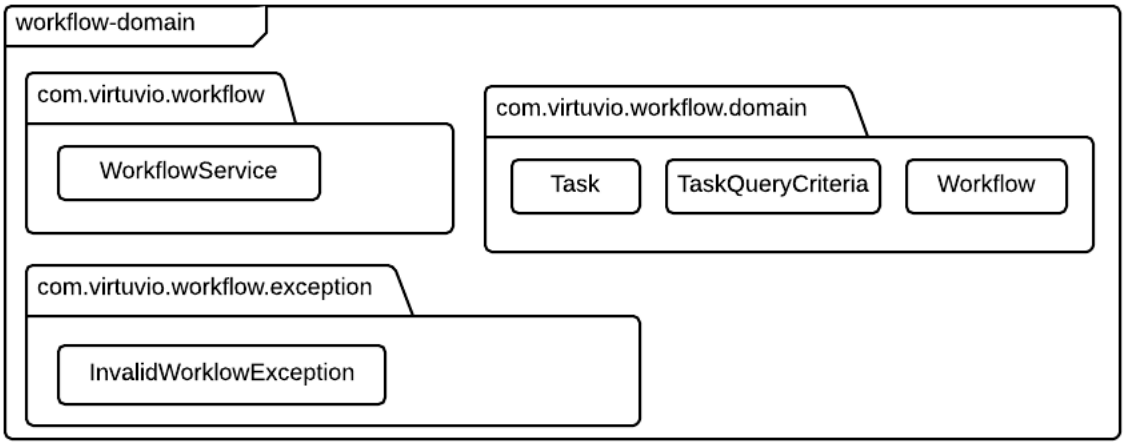


Figura 6.3 Diagrama de paquetes: domain

El paquete **workflow-domain** está compuesto por varios subpaquetes donde se encuentran las clases del modelo que utiliza la plataforma eGob! para trabajar con los workflows, el interfaz con las operaciones que puede realizar con los workflows y una excepción que se a creado para tratar los posibles errores con la interacción con los workflows.

6.1.1.3 Paquete workflow-impl-activiti

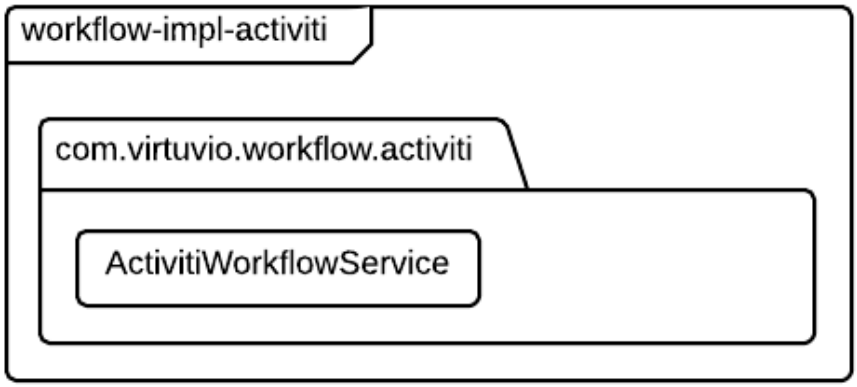


Figura 6.4 Diagama de paquetes: impl-activiti

El paquete **workflow-impl-activiti** está compuesto por una clase que implementa el interfaz del paquete **workflow-domain**. Se ha decidido realizar esta separación para para así en cualquier momento cambiar Activiti por otra tecnología, y que tan sólo haya que crear un nuevo módulo con la implementación del interfaz del **workflow-domain**.

6.1.2 Diagramas de Despliegue

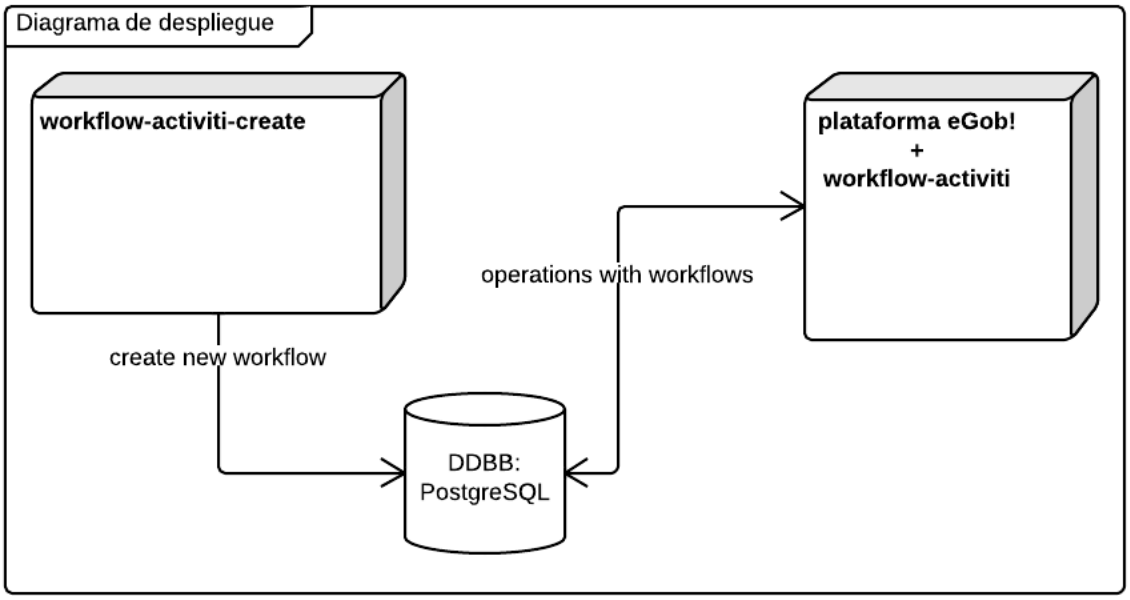


Figura 6.5 Diagrama de Despliegue

6.1.2.1 Elemento workflow-activiti-create

Aplicación Java la cual permite diseñar los workflows y después desplegarlos en base de datos.

6.1.2.2 Elemento BBDD: PostgreSQL

Sistema gestor de bases de datos que utilizan los otros dos componentes para realizar las operaciones con los workflows.

6.1.2.3 Elementos Plataforma eGob! + workflow-activiti

La plataforma eGob! está conformada por un conjunto de aplicaciones web que permiten realizar las diferentes interacciones con los usuarios y sus operaciones se encuentra desplegada en el servidor Tomcat junto con los módulos creados en este proyecto para el trabajo con los workflows diseñados con Activiti.

6.2 Diseño de Clases

6.2.1 Diagrama de Clases

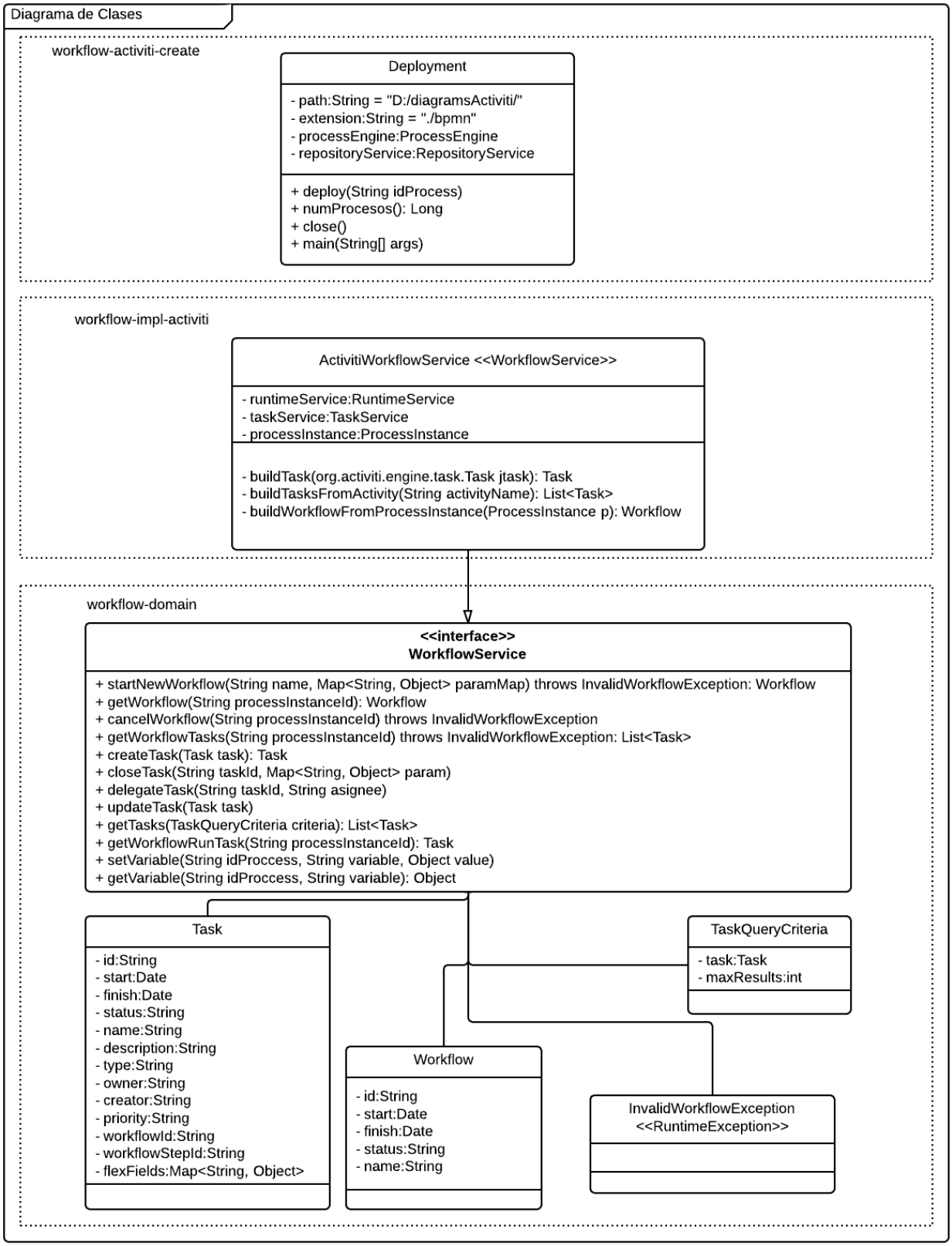


Figura 6.6 Diagrama de Calses

6.3 Diagramas de Interacción y Estados

6.3.1 Funcionamiento workflows

6.3.1.1 Diagrama de Secuencia

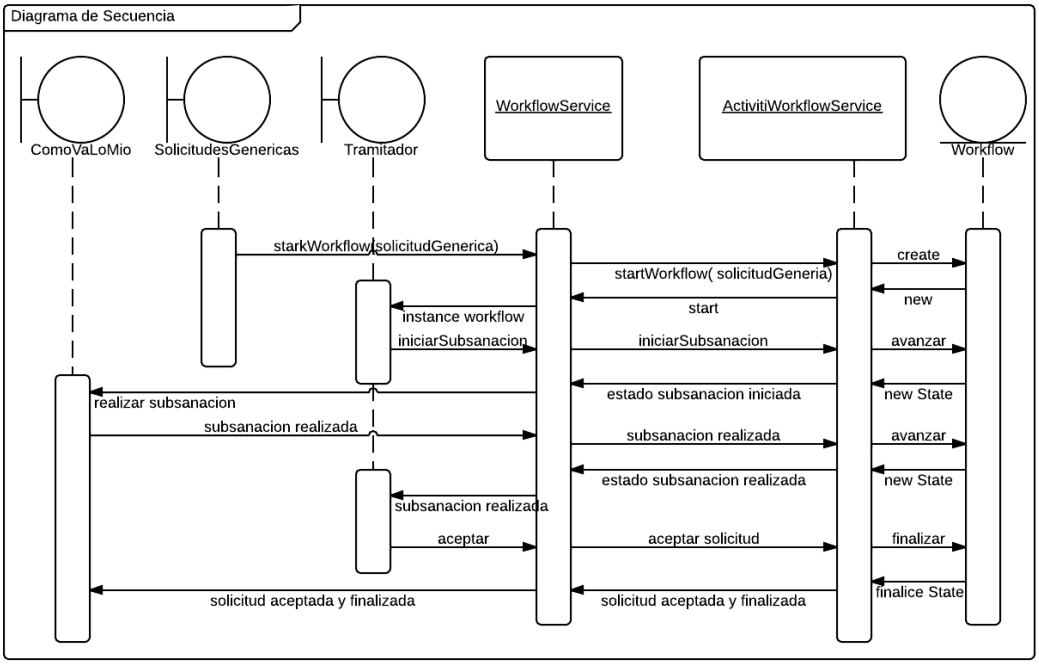


Figura 6.7 Diagrama de Secuencia

6.3.1.2 Diagrama de Estados

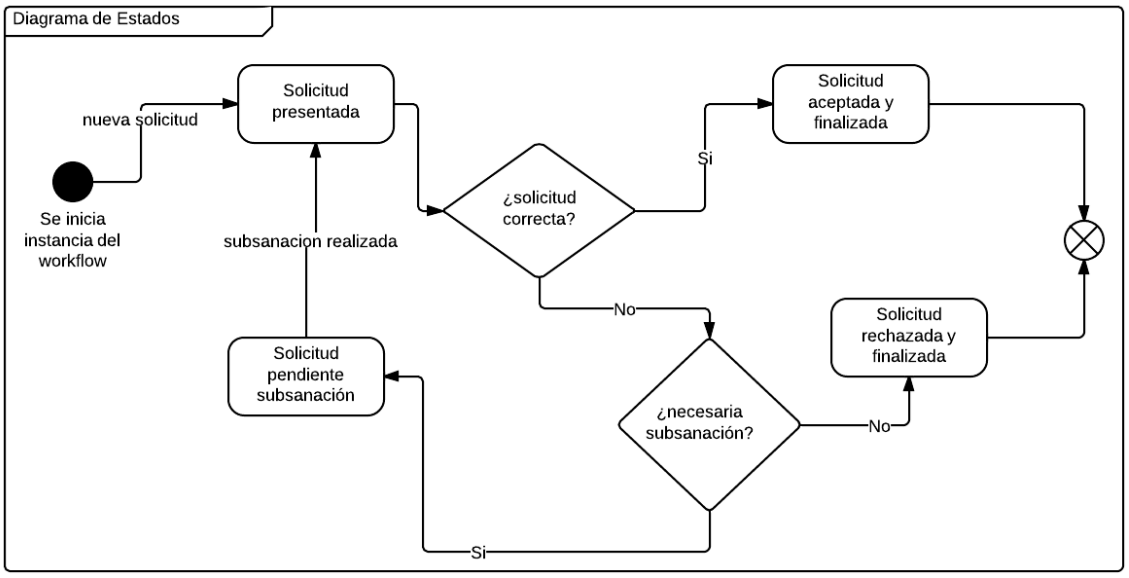


Figura 6.8 Diagram de Estados

6.4 Diseño de la Base de Datos

6.4.1 Descripción del SGBD Usado

6.4.1.1 PostgreSQL

El sistema gestor de bases de datos que se ha utilizado para el desarrollo del proyecto ha sido PostgreSQL versión 9.1.5, se ha decido utilizar esta porque se requería un base de datos SQL relacional, y dentro de las posibles opciones que teníamos como Oracle, MySQL y PostgreSQL entre otras, se optó primero por una "gratuita" entre las que nos quedamos con MySQL y PostgreSQL, posteriormente entre ellas, la decisión de cuál escoger se debió a que el gestor de trámites ya trabajaba con esta base de datos y por tanto la integración iba a ser mucho más fácil que si se decidía utilizar otra para el despliegue de los workflows.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y de código abierto más potente del mercado.

El proyecto PostgreSQL tal y como lo conocemos hoy en día comenzó en 1996, aunque las bases y el trabajo en la que se asienta tienen sus comienzos en la década de los 70.

Su funcionamiento se basa en el modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Una breve descripción sobre sus principales componentes:

- **Aplicación cliente**: es la aplicación que se utiliza como administrador de las bases de datos. Se puede conectar mediante una conexión TCP/IP ó mediante sockets.

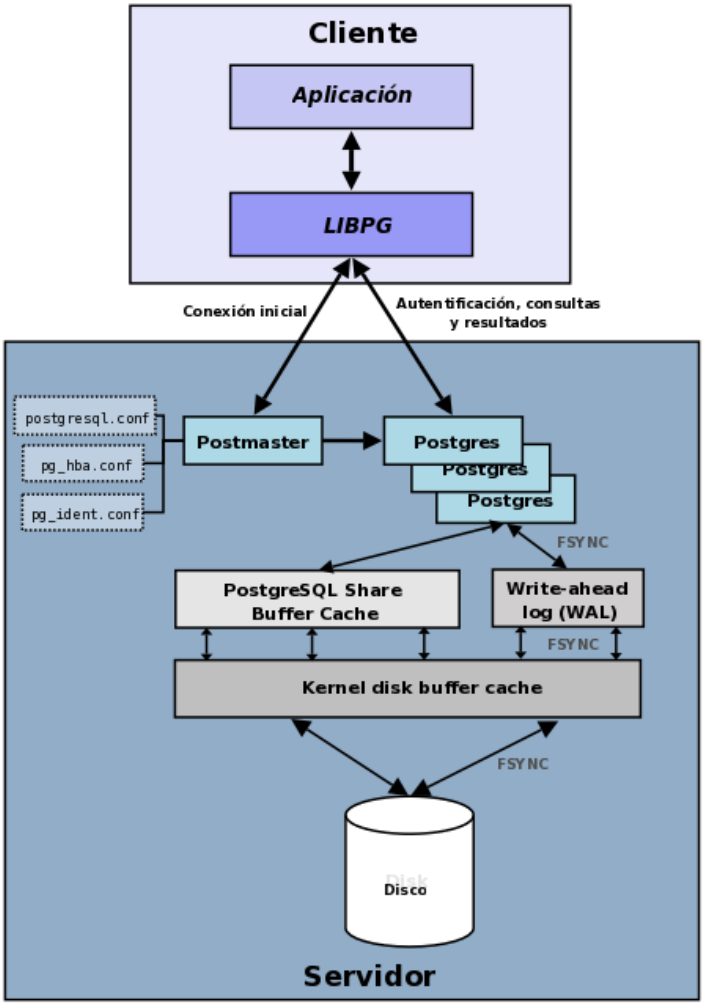


Figura 6.9 Estructura PosgreSQL

- **Demonio postmaster**: es el proceso principal y el encargado de escuchar en un puerto o socket por conexiones entrantes, también es el encargado de crear los procesos hijos.
- **Ficheros de configuración**: los 3 ficheros principales de configuración son postgresql.conf, pg_hba.conf y pg_ident.conf
- **Procesos hijos**: son los que se encargan de autentificar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **PostgreSQL share buffer cache**: es la memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** es el componente encargado de asegurar la integridad de los datos.
- **Kernel disk buffer cache**: es la caché de disco del sistema operativo
- **Disco**: es el disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

6.4.2 Integración del SGBD en el proyecto

A continuación se puede ver el diagrama entidad-relación, que se genera cuando se despliega un workflow y posteriormente cuando se van creando instancias de dicho workflow. Este diagrama lo genera Activiti, y se mantiene a través de los dos subsistemas que he creado.

Por otro lado está el diagrama Entidad-Relación de la plataforma eGob! el cual almacena el workflow que se asigna a cada convocatoria, y los identifacadores de las instancias de workflow de cada solicitud. No se adjunta diagrama de ésta parte porque éste ya existía y es mantenido por eGob!

6.4.3 Diagrama E-R

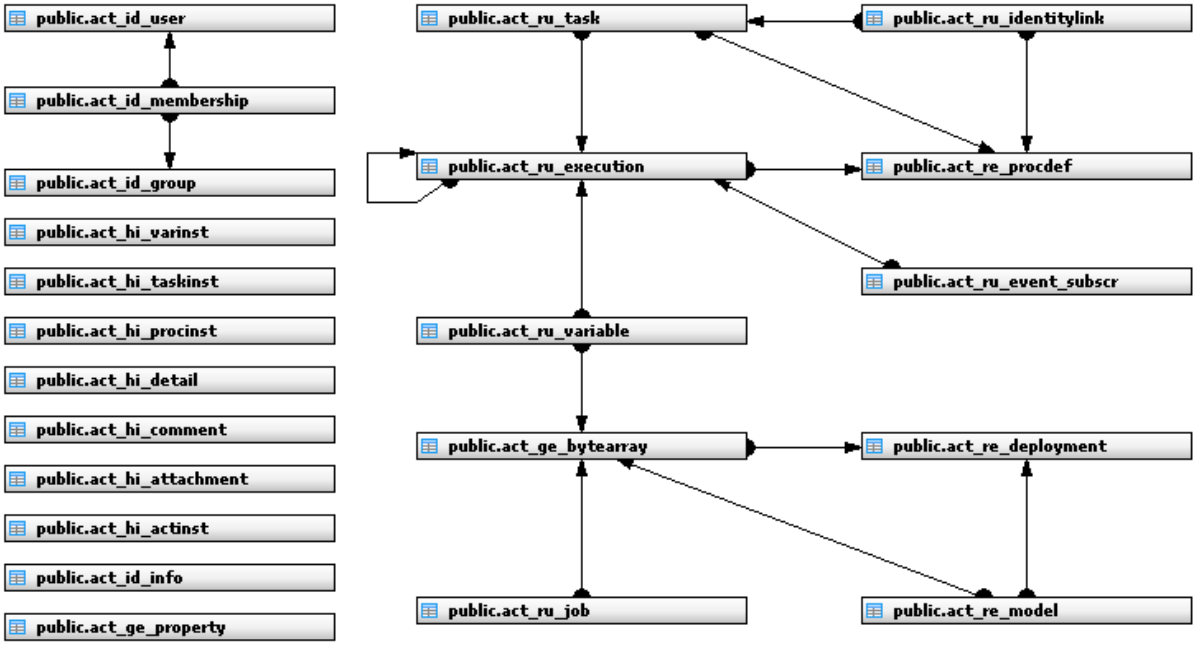


Figura 6.10 Diagram Entidad-Relación (sólo nombre tablas)

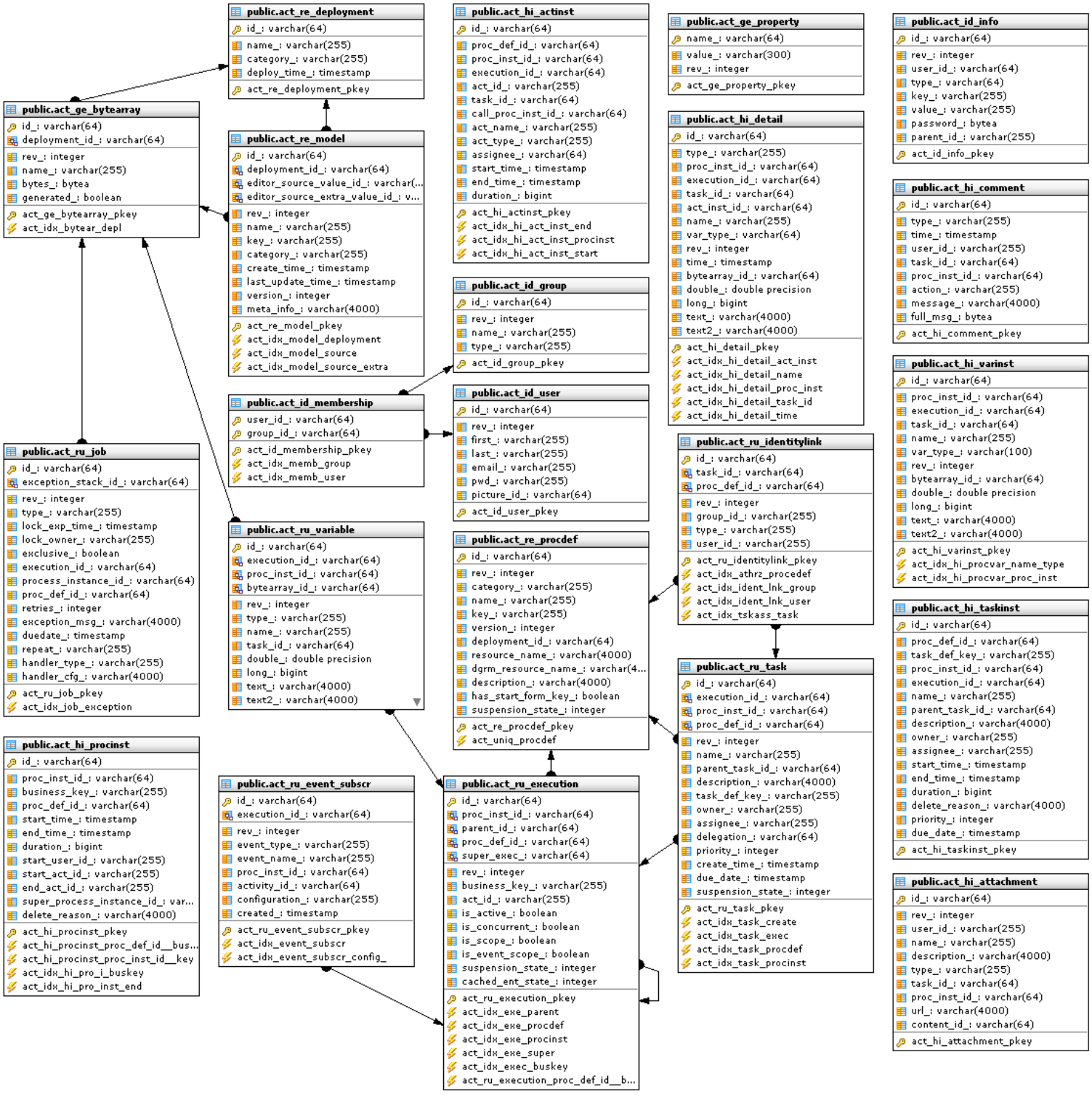


Figura 6.11 Diagram Entidad-Relación (Completo)

6.5 Especificación Técnica del Plan de Pruebas

6.5.1 Pruebas Unitarias

Para la realización de las pruebas unitarias como ya se ha comentado anteriormente se ha utilizado JUnit. En estas pruebas se ha probado la correcta implementación del Interfaz WorkflowService con la clase ActivitiWorkflowService. Además de una pequeña prueba para ver que la configuración con base de datos y Activiti es correcta desplegando un workflow de prueba.

6.5.1.1 Pruebas ActivitiWorkflowService

```
package com.vitruvio.workflow.activiti;

import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.annotation.Resource;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.vitruvio.workflow.WorkflowService;
import com.vitruvio.workflow.domain.Task;
import com.vitruvio.workflow.domain.TaskQueryCriteria;
import com.vitruvio.workflow.domain.Workflow;

/**
 * @author nataliagm
 */
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = { "classpath*:activiti-context-test.xml" })
public class TestActivitiWorkflowService {

	@Resource(name = "activitiWorkflowService")
	WorkflowService activitiWorkflowService;

	@Test
	public void testInit() {
		System.out.println("Init");
	}

	@Test
	public void testStartNewWorkflow() {

		Map<String, Object> m = new HashMap<String, Object>();
		m.put("documentid", 102026);
		activitiWorkflowService.startNewWorkflow("activiti-firmaunusuario", m);
		System.out.println("StartNewWorkflowCorrecto");
	}
```

```
	@Test
	public void testCancelWorkflow() {

		Map<String, Object> m = new HashMap<String, Object>();
		m.put("documentid", 102026);
		Workflow p = activitiWorkflowService.startNewWorkflow(
				"activi-firmaunusuario", m);
		activitiWorkflowService.cancelWorkflow(p.getId());
		activitiWorkflowService.getWorkflow(p.getId());
		System.out.println("testCancelWorkflow");
	}

	@Test
	public void testDelegateTask() {
		activitiWorkflowService.delegateTask("3305", "UO213296");
	}

	@Test
	public void testCloseTask() {
		Map<String, Object> m = new HashMap<String, Object>();
		m.put("documentid", 102026);
		activitiWorkflowService.closeTask("3205", m);
		System.out.println("testCloseTask");
	}

	@Test
	public void testGetTasks() {
		TaskQueryCriteria tqc = new TaskQueryCriteria();
		List<Task> task = activitiWorkflowService.getTasks(tqc);
		System.out.println("Numero tareas: " + task.size());
	}

	@Test
	public void testCreateTask() {

		Task t = new Task();
		t.setCreator("UO213296");
		t.setDescription("Tarea prueba TEST");
		t.setName("TEST");
		t.setOwner("prueba_peri");
		t.setWorkflowId("activi-firmaunusuario");
		t.setStart(new Date());
		activitiWorkflowService.createTask(t);

		System.out.println("testCreateTask");

	}

	@Test
	public void testUpdateTask() {
		Task t = new Task();
		t.setCreator("UO213296");
		t.setDescription("Tarea MODIFICADA TEST");
		t.setName("TEST");
		t.setOwner("prueba_peri");
		t.setWorkflowId("activi-firmaunusuario");
		t.setStart(new Date());
		t.setId("4301");

		activitiWorkflowService.updateTask(t);
		System.out.println("testUpdateTask");
	}

	@Test
	public void testGetWorkflowTasks() {
		List<Task> tasks = activitiWorkflowService.getWorkflowTasks("7401");
		System.out.println("Numero tareas 7401: " + tasks.size());
	}

	@Test
	public void testGetWorkflow() {
		Workflow w = activitiWorkflowService.getWorkflow("7401");
		System.out.println(w.getId() + " / " + w.getName() + w.getStatus());
	}

}
```

6.5.1.2 Pruebas Desplegar workflow de pruebas

```
package es.uniovi.workflow.activiti.prueba;

import java.util.HashMap;

import org.activiti.engine.ProcessEngine;
import org.activiti.engine.ProcessEngines;
import org.activiti.engine.RuntimeService;
import org.activiti.engine.runtime.ProcessInstance;
import org.springframework.stereotype.Component;

/**
 *
 * @author nataliagm
 *
 */
@Component
public class Prueba {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		ProcessEngine processEngine = ProcessEngines.getDefaultProcessEngine();

		HashMap<String, Object> map = new HashMap<String,Object>();
		Object t = new String();
		t = "Prueba";
		map.put("tarea", t);

		ProcessInstance p =
processEngine.getRuntimeService().startProcessInstanceByKey("activiti-
solicitudgenerica",map);
		RuntimeService r = processEngine.getRuntimeService();

		System.out.println(p.getProcessInstanceId());
		r.suspendProcessInstanceById(p.getProcessInstanceId());
		r.deleteProcessInstance(p.getId(), null);


	}
}
```

6.5.2 Pruebas de Integración y del Sistema

Para realizar las pruebas de integración como ya se ha comentado he realizado algunas de las clases con las cuales tenía que ejecutar el workflow y las ejecuté desde el propio subsistema de creación de workflows para probar que las llamadas y ejecución eran correctas

Las pruebas realizadas han sido las siguientes, con las clases que he creado:

- Solicitar datos para introducirlos en una tarea.
- Solicitar datos para el envío de emails desde Activiti.
- Solicitar datos para delegar una tarea.
- Obtener datos de los parámetros de un workflow.
- Insertar datos en los parámetros de un workflow.

6.5.3 Pruebas de Usabilidad y Accesibilidad

6.5.3.1 Actividades de las Pruebas de Usabilidad

Como ya se ha comentado anteriormente no se aplican pruebas usabilidad, debido a que la interacción con el Usuario no se realiza en el desarrollo de éste proyecto sino en la plataforma eGob!

6.5.3.2 Pruebas de Accesibilidad

Al igual que las pruebas de usabilidad, las pruebas de accesibilidad tampoco se aplican en la creación de éste proyecto.

6.5.4 Pruebas de Rendimiento

Debido a que éste proyecto se trata de un subsistema en el back-end de la plataforma ya existente, no dispone de ningún cuello de botella y no tiene que realizar procesos lentos, no he realizado pruebas de rendimiento.

Las pruebas de rendimiento se han realizado a la plataforma eGob! la cual si tiene que soportar múltiples peticiones simultáneas y puede llegar a tener problemas de rendimiento.

Capítulo 7. Implementación del Sistema

7.1 Estándares y Normas Seguidos

Para el desarrollo de este proyecto he seguido las buenas prácticas de programación Java y también los consejos de BPMN para la creación de los workflows.

7.2 Lenguajes de Programación

7.2.1 Java

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems.

El equipo original que desarrolló Java lo hizo por necesidad, al no cumplir los lenguajes existentes con todo lo que necesitaban para el proyecto en marcha. Pero, como buenos diseñadores, no inventaron todo de nuevo, sino que se basaron en lo ya hecho y probado. Es por esa razón que el código Java se expresa en archivos

de texto comunes, y tiene una apariencia muy familiar para los programadores de C/C++ y para los programadores en general.

Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

El proceso de compilación no produce código para un procesador en particular, como en los compiladores C/C++, sino que genera código para un procesador ideal, denominado máquina virtual Java (Java Virtual Machine, o Java VM). La razón de esta conducta es simple: la necesidad de poder ejecutarse en cualquier plataforma, sin necesidad de cambiar el código fuente, ni aun de recompilar.

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

Veremos que aun los tipos primitivos de datos quedan definidos de un solo golpe, para todas las plataformas. Nos evitamos así las pesadillas de portabilidad, conocidas por los programadores de C/C++, al cambiar, por ejemplo, de ambiente a ambiente, el tamaño de los enteros, o el conjunto de caracteres soportados. En Java, esos problemas no existen: sólo existe una máquina virtual.

En Java hay un manejo automático de la memoria, los objetos pueden y deben crearse, y tienen una vida que dura hasta su destrucción. Mientras que la creación de los objetos se deja

bajo la voluntad del programador, la destrucción definitiva de un objeto ocurre cuando no es más referenciado por otros objetos del programa. De esta forma, se elimina una de las causas más comunes de error en otros lenguajes, como la destrucción por el programador de objetos aun en uso en el programa, o la falta de destrucción de objetos que ya son inútiles, pues no se usan en el resto de la ejecución, pero que molestan con empleo de recursos. Esta técnica de manejo automático de memoria ocupada por los objetos se denomina garbage collection, recolección de basura. En una aplicación Java hay siempre un proceso, ejecutado como un hilo de ejecución separado, que se ocupa de recorrer la memoria donde se encuentran los objetos, y determinan cuáles son pueden liberarse y destruirse.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

Actualmente Java se encuentra la versión JAVA SE 8, aunque para el proyecto ya que se debía de integran con el gestor de Trámites de la Universidad se ha utilizado la versión JAVA SE 6.

7.3 Herramientas y Programas Usados para el Desarrollo

7.3.1 Eclipse Juno

El programa utilizado para el desarrollo del código ha sido Eclipse Juno, es una plataforma que se utiliza para desarrollar entornos de desarrollo integrados (IDE), se ha utilizado el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse.

7.3.2 Tomcat 6

Servidor de aplicaciones web donde está desplegada la plataforma eGob! y donde se despliegan también mis subsistemas.

7.3.3 Microsoft Office Word 2010

Para todo el desarrollo de la documentación he utilizado Microsoft Office Word 2010.

7.3.4 Microsoft Office Excel 2010

Programa utilizado para el desarrollo de los presupuestos.

7.3.5 Lucidchart

He utilizado el programa Lucidchart para realizar todos los diagramas necesarios para el análisis y diseño del proyecto.

7.3.6 Microsoft Office Project 2013

Para el desarrollo de la planificación del proyecto he utilizado Microsoft Office Project.

7.3.7 SVN

Para almacenar el código y mantener un sistema de control de versiones he utilizado un SVN, disponible en el entorno de trabajo donde se encontraba el gestor de trámites de la Universidad.

7.3.8 pgAdmin III

He utilizado pgAdmin III para gestionar la base de datos de PostgreSQL, ya que se trata de un interfaz gráfico muy completo.

7.4 Creación del Sistema

Para el desarrollo del proyecto, puesto que se trataba de algo nuevo para mí, tuve que comenzar estudiando el contexto del proyecto, conociendo el tema de los procesos administrativos que se desarrollaban en la Universidad, que se estaba realizando actualmente en el gestor de trámites y que herramientas habían en el mercado y pudieran ayudar en dicho proceso.

7.4.1 Problemas Encontrados

Los problemas encontrados se han debido principalmente a la complejidad del gestor de trámites de la Universidad y a la dificultad de interactuar con él, para trabajar con los workflows. Otro de los problemas encontrados ha sido la falta de una definición oficial de los procedimientos, por tanto a veces fue complicado definir qué proceso implementar, y luego hubo que retocarlo por cambios en las necesidades del procedimiento.

7.4.2 Descripción Detallada de las Clases

A continuación se pueden ver las clases más importantes de mis subsistemas.

7.4.3 Subsistema workflow-domain

7.4.3.1 Interface WorkflowService

`public interface` **`WorkflowService`**
Interfaz que define las operaciones que el gestor de becas puede realizar con los workflows de Activiti

Author: nataliagm

7.4.3.1.1 Method Summary	
Void	**`cancelWorkflow`**`(java.lang.String processInstanceId)` Finaliza la instancia del workflow que se le indica
Void	**`closeTask`**`(java.lang.String taskId, java.util.Map<java.lang.String,java.lang.Object> param)` Finaliza la tarea que se indica, añdiendo los parámetros que recibe
Task	**`createTask`**`(Task task)` Crea una tarea de Activiti a partir de la tarea del modelo recibida
Void	**`delegateTask`**`(java.lang.String taskId, java.lang.String asignee)` Asigna la tarea recibida al usuario indicado

7.4.3.1.1 Method Summary	
java.util.List <Task>	**getTasks**(TaskQueryCriteria criteria) Devuelve las tareas del workflow y del creador que se indican en el criterio de búsqueda
java.lang.Object	**getVariable**(java.lang.String idProccess, java.lang.String variable) Devuelve la variable del workflow que se solicita
Workflow	**getWorkflow**(java.lang.String processInstanceId) Devuelve la instancia del workflow al que pertenece el identificador
Task	**getWorkflowRunTask**(java.lang.String processInstanceId) Devuelve la tarea que se está ejecutando en ese momento en la instancia del workflow que se indica
java.util.List <Task>	**getWorkflowTasks**(java.lang.String processInstanceId) Devuelve una lista con todas las tareas del workflow que se le indica
Void	**setVariable**(java.lang.String idProccess, java.lang.String variable, java.lang.Object value) Añade la variable que se indica en la instancia del workflow indicado
Workflow	**startNewWorkflow**(java.lang.String name, java.util.Map<java.lang.String,java.lang.Object> paramMap) Crea una nueva instancia del workflow que se indica y añade los argumentos recibidos
void	**updateTask**(Task task) Modifica la tarea de Activiti con los datos recibidos en la tarea del modelo

7.4.3.1.2 Method Detail

startNewWorkflow

Workflow **startNewWorkflow**(java.lang.String name,

java.util.Map<java.lang.String,java.lang.Object> paramMap)

throws InvalidWorkflowException

Crea una nueva instancia del workflow que se indica y añade los argumentos recibidos

Parameters:

- `name` - nombre del workflow que se quiere instanciar
- `paramMap` - parámetros que se van a incluir en la creación de la instancia del workflow

Returns:

- Workflow creado

Throws:

- `InvalidWorkflowException`

getWorkflow

Workflow **getWorkflow**(java.lang.String processInstanceId)

Devuelve la instancia del workflow al que pertenece el identificador

Parameters:

- `processInstanceId` - identificador del workflow que se solicita

Returns:

- Workflow solicitado

cancelWorkflow

void **cancelWorkflow**(java.lang.String processInstanceId)

throws InvalidWorkflowException

Finaliza la instancia del workflow que se le indica

Parameters:

- processInstanceId - identificador de la instacia de workflow que se quiere cancelar

Throws:

- `InvalidWorkflowException`

getWorkflowTasks

java.util.List<Task> **getWorkflowTasks**(java.lang.String processInstanceId)

throws InvalidWorkflowException

Devuelve una lista con todas las tareas del workflow que se le indica

Parameters:

- `processInstanceId` - identificador de la instancia de workflow

Returns:

- lista de Tareas

Throws:

- `InvalidWorkflowException`

createTask

Task **createTask**(Task task)

Crea una tarea de Activiti a partir de la tarea del modelo recibida

Parameters:

- `task` - Tarea que se desea crear

Returns:

- Tarea creada

closeTask

void **closeTask**(java.lang.String taskId,

java.util.Map<java.lang.String,java.lang.Object> param)

Finaliza la tarea que se indica, añadiendo los parámetros que recibe

Parameters:

- `taskId` - identificador de la tarea
- `param` - Datos que se desean añadir a la tarea

delegateTask

void **delegateTask**(java.lang.String taskId,

java.lang.String asignee)

Asigna la tarea recibida al usuario indicado

Parameters:

- `taskId` - identificador de la tarea
- `asignee` - usuario al que se va a asignar la tarea

updateTask

void **updateTask**(Task task)

Modifica la tarea de Activiti con los datos recibidos en la tarea del modelo

Parameters:

- `task` - Tarea para modificar

getTasks

java.util.List<Task> **getTasks**(TaskQueryCriteria criteria)

Devuelve las tareas del workflow y del creador que se indican en el criterio de búsqueda

Parameters:

- `criteria` - Criterio de búsqueda de la tareas

Returns:

- lista con las tareas que cumplen con el criterio de búsqueda

getWorkflowRunTask

Task **getWorkflowRunTask**(java.lang.String processInstanceId)

Devuelve la tarea que se está ejecutando en ese momento en la instancia del workflow que se indica

Parameters:

- `processInstanceId` - identificador de la instancia del workflow

Returns:

- Tarea que se encuentra en ejecución

setVariable

void **setVariable**(java.lang.String idProccess,

java.lang.String variable,

java.lang.Object value)

Añade la variable que se indica en la instancia del workflow indicado

Parameters:

- `idProccess` - identificador de la instancia del workflow
- `variable` - Nombre de la variable que se desea añadir
- `value` - Valor de la variable que se desea añadir

getVariable

java.lang.Object **getVariable**(java.lang.String idProccess,

java.lang.String variable)

Devuelve la variable del workflow que se solicita

Parameters:

- `idProccess` - identificador de la instancia del workflow
- `variable` - Nombre de la variable que se quiere obtener

Returns:

- Variable que se está solicitando

7.4.3.2 Class Task

```
public class Task
extends java.lang.Object
```

Modelo de Tarea que se utiliza desde el gestor de trámites

Author: nataliagm

7.4.3.2.1 Field Summary

private java.lang.String	creator
private java.lang.String	description
private java.util.Date	finish
private java.util.Map<java.lang.String, java.lang.Object>	flexFields
private java.lang.String	id
private java.lang.String	name
private java.lang.String	owner
private java.lang.String	priority
private java.util.Date	start
private java.lang.String	status
private java.lang.String	type
private java.lang.String	workflowId
private java.lang.String	workflowStepId

7.4.3.2.2 Constructor Summary

Task()	

7.4.3.2.3 Method Summary

java.lang.String	getCreator()
java.lang.String	getDescription()
java.util.Date	getFinish()
java.util.Map<java.lang.String, java.lang.Object>	getFlexFields()
java.lang.String	getId()
java.lang.String	getName()
java.lang.String	getOwner()
java.lang.String	getPriority()
java.util.Date	getStart()
java.lang.String	getStatus()
java.lang.String	getType()
java.lang.String	getWorkflowId()
java.lang.String	getWorkflowStepId()
void	setCreator(java.lang.String creator
void	setDescription(java.lang.String description)

7.4.3.2.3 Method Summary

void	**setFinish**(java.util.Date finish)
void	**setFlexFields**(java.util.Map<java.lang.String,java.lang.Object> flexFields)
void	**setId**(java.lang.String id)
void	**setName**(java.lang.String name)
void	**setOwner**(java.lang.String owner)
void	**setPriority**(java.lang.String priority)
void	**setStart**(java.util.Date start)
void	**setStatus**(java.lang.String status)
void	**setType**(java.lang.String type)
void	**setWorkflowId**(java.lang.String workflowId)
void	**setWorkflowStepId**(java.lang.String workflowStepId)

7.4.3.2.4 Method Detail

Getters and Setter de todos los atributos de la clase.

7.4.3.3 Class Workflow

```
public class Workflow
extends java.lang.Object
```

Modelo Workflow con el que trabaja el gestora de Trámites

Author: nataliagm

7.4.3.3.1 Field Summary	
private java.util.Date	finish
private java.lang.String	id
private java.lang.String	name
private java.util.Date	start
private java.lang.String	status

7.4.3.3.2 Constructor Summary	
Workflow()	

7.4.3.3.3 Method Summary	
java.util.Date	getFinish()
java.lang.String	getId()
java.lang.String	getName()
java.util.Date	getStart()
java.lang.String	getStatus()
void	setFinish(java.util.Date finish)
void	setId(java.lang.String id)
void	setName(java.lang.String name)
void	setStart(java.util.Date start)
void	setStatus(java.lang.String status)

7.4.3.4 Class TaskQueryCriteria

```
public class TaskQueryCriteria
extends java.lang.Object
```

Modelo TaskQueryCriteria que se utiliza desde el gestor de trámites, para crear consultar en las Tareas

Author: nataliagm

7.4.3.4.1 Field Summary

private int	**maxResults**
private Task	**task**

7.4.3.4.2 Constructor Summary

TaskQueryCriteria()	

7.4.3.4.3 Method Summary

int	**getMaxResults**()
Task	**getTask**()
void	**setMaxResults**(int maxResults)
void	**setTask**(Task task)

7.4.4 Subsistema workflow-impl-activiti

7.4.4.1 Class ActivitiWorkflowService

All Implemented Interfaces: com.vitruvio.workflow.WorkflowService

```
@Service(value="activitiWorkflowService")
public class ActivitiWorkflowService
extends java.lang.Object
implements com.vitruvio.workflow.WorkflowService
```

Implementaciónn del interfez WorkflowService, donde se realiza la implementación con Activiti

Author: nataliagm

7.4.4.1.1 Field Summary	
private org.activiti.engine.runtime.ProcessInstance	process
private org.activiti.engine.RuntimeService	runtimeService
private org.activiti.engine.TaskService	taskService

7.4.4.1.2 Constructor Summary	
ActivitiWorkflowService()	

7.4.4.1.3 Method Summary	
Private com.vitruvio.workflow.domain.Task	buildTask(org.activiti.engine.task.Task jtask)
private java.util.List<com.vitruvio.workflow.domain.Task>	buildTasksFromActivity(java.lang.String activityName)
private com.vitruvio.workflow.domain.Workflow	buildWorkflowFromProcessInstance(org.activiti.engine.runtime.ProcessInstance p)
void	cancelWorkflow(java.lang.String processInstanceId)
void	closeTask(java.lang.String taskId, java.util.Map<java.lang.String,java.lang.Object> param)
com.vitruvio.workflow.domain.Task	createTask(com.vitruvio.workflow.domain.Task task)
void	delegateTask(java.lang.String taskId, java.lang.String asignee)
java.util.List<com.vitruvio.workflow.domain.Task>	getTasks(com.vitruvio.workflow.domain.TaskQueryCriteria criteria)
java.lang.Object	getVariable(java.lang.String idProccess, java.lang.String variable)
com.vitruvio.workflow.domain.Workflow	getWorkflow(java.lang.String processInstanceId)

7.4.4.1.3 Method Summary	
com.vitruvio.workflow.domain.Task	**getWorkflowRunTask**(java.lang.String processInstanceId)
java.util.List<com.vitruvio.workflow.domain.Task>	**getWorkflowTasks**(java.lang.String processInstanceId)
void	**setVariable**(java.lang.String idProccess, java.lang.String variable, java.lang.Object value)
com.vitruvio.workflow.domain.Workflow	**startNewWorkflow**(java.lang.String name, java.util.Map<java.lang.String,java.lang.Object> paramMap)
void	**updateTask**(com.vitruvio.workflow.domain.Task task)

7.4.4.1.4 Field Detail

runtimeService

```
@Autowired
private org.activiti.engine.RuntimeService runtimeService
```

taskService

```
@Autowired
private org.activiti.engine.TaskService taskService
```

process

```
private org.activiti.engine.runtime.ProcessInstance process
```

7.4.5 Subsistema workflow-activiti-create

7.4.5.1 Class Deployment

```
public class Deployment
extends java.lang.Object
```

Clase para desplegar en base de datos los workflow creados

Author: nataliagm

7.4.5.1.1 Field Summary	
private java.lang.String	extension
private java.lang.String	path
private org.activiti.engine.ProcessEngine	processEngine
private org.activiti.engine.RepositoryService	repositoryService

7.4.5.1.2 Constructor Summary	
Deployment()	

7.4.5.1.3 Method Summary	
void	close()
void	deploy(java.lang.String idProcess)
static void	main(java.lang.String[] args)
long	numProcesos()

Capítulo 8. Desarrollo de las Pruebas

8.1 Pruebas Unitarias

Las pruebas realizadas como ya se ha comentado han sido principalmente para ver que la implementación del Interfaz WorkflowService, con la clase ActivitiWorkflowService era correcta.

Se han probado todos los métodos que tiene la clase, y finalmente el resultado ha sido correcto en todas las pruebas.

Se ha probado la creación y cancelación de workflows. La creación, finalización, actualización y obtención de tareas de un workflow.

Se ha probado a delegar las tareas a otros usuarios y a obtener la tarea que se encontraba en ejecución es ese momento.

8.2 Pruebas de Integración y del Sistema

Las pruebas de integración se han realizado primero con las clases auxiliares que yo he creado para simular los procesos y correcto funcionamiento de diferentes operaciones a realizar por las tareas de los workflows, y posteriormente ya se pasó a ejecutar los workflows utilizando las clases existentes en la plataforma eGob!

El resultado de las pruebas siempre y cuando la plataforma eGob! se encontrada correctamente en funcionamiento ha sido siempre correcto.

Las pruebas realizadas han sido las siguientes:

- Solicitar datos para introducirlos en una tarea
    - Se obtienen los datos y estos eran introducidos correctamente en la tarea, pudiendo ser consultados y utilizados en los siguientes pasos del workflow.
- Solicitar datos para el envío de emails desde Activiti:
    - El email enviado desde Activiti se realiza correctamente a la dirección solicitada
- Solicitar datos para delegar una tarea
    - La tarea se delega correctamente al usuario indicado, siendo éste el que deba realizarla
- Obtener datos de los parámetros de un workflow
    - Se obtienen correctamente los datos de la solicitud almacenados en parámetros del workflow
- Insertar datos en los parámetros de un workflow
    - Se insertan correctamente los datos y estos están disponibles en los siguientes pasos del workflow

Capítulo 9. Manuales del Sistema

9.1 Manual de Instalación

Para llevar a cabo la instalación y correcto funcionamiento de éste proyecto es necesario tener Java 6 instalado en el equipo, ya que el proyecto ha sido desarrollado con éste lenguaje como ya se ha comentado. Java 6 se puede obtener de la página oficial de Oracle, se recomienda descargar una jdk.

Se debe de disponer de un servidor de aplicaciones, concretamente el utilizado para el desarrollo ha sido Tomcat 6 como ya se ha comentado. El servidor de aplicaciones se puede obtener de la página oficial del producto Apache Tomcat, tan sólo es necesario descargarlo e iniciarlo, para posteriormente añadir la aplicación.

También es necesario disponer de un servidor de Bases de Datos, el utilizado para este proyecto ha sido PostgreSQL, éste producto también se puede obtener de la página oficial de Postgre. Para que la aplicación funcione correctamente es necesario crear una base de datos con el nombre "***vworkflowActiviti***". El resto de bases de datos que se utilizan depende de la plataforma eGob!

El proyecto ha sido desarrollado utilizando Eclipse Juno el cual se puede obtener de la página oficial de Eclipse. Para poder ver los diagramas de los workflows o modificar o desarrollar alguno nuevo es necesario instalar el plugin de Activiti en el Eclipse, el plugin utilizado es **Activiti Eclipse BPMN 2.0 Designer.**

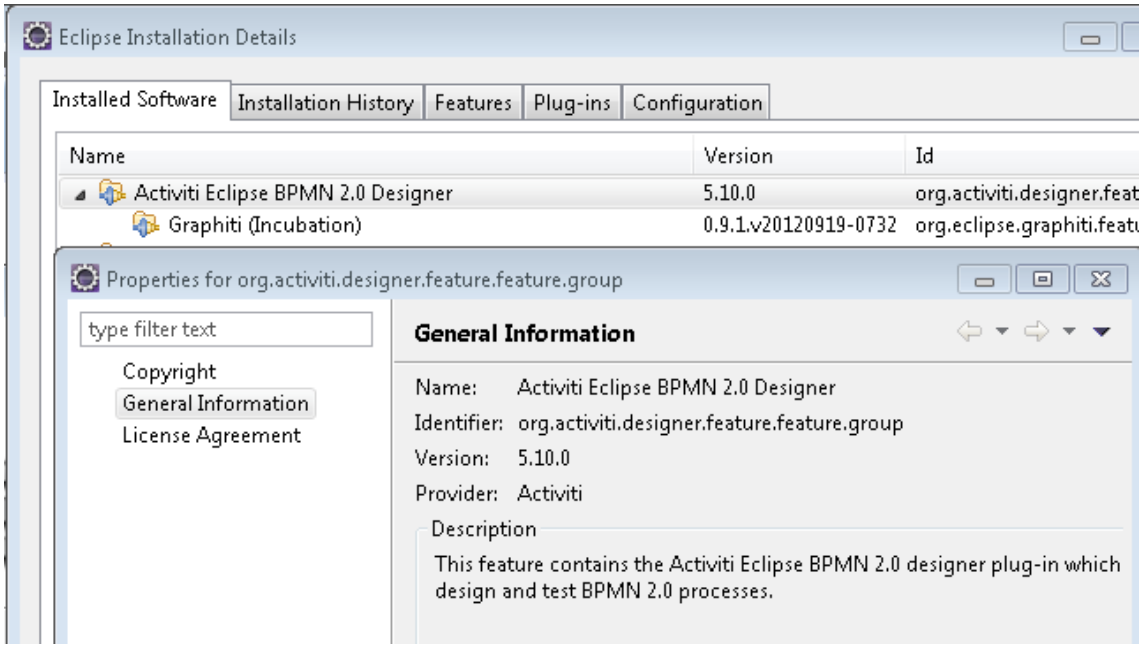


Figura 9.1 Plugin Eclipse Activiti BPMN 2.0 Designer

El proyecto ha sido desarrollado con Maven para la gestión de dependencia y los subproyectos, ya que eGob! ya estaba desarrollada con Maven, por tanto es necesario tenerlo en nuestro ordenador, para ello tan sólo hay que acceder a la página oficial de Maven descargarlo, descomprimir el fichero y configurar en Eclipse y las variables de entorno para poder ser utilizado desde allí. En las variables de entorno hay meterlo en la variable path indicando donde se encuentra la carpeta "***apache-maven-XX\bin***".

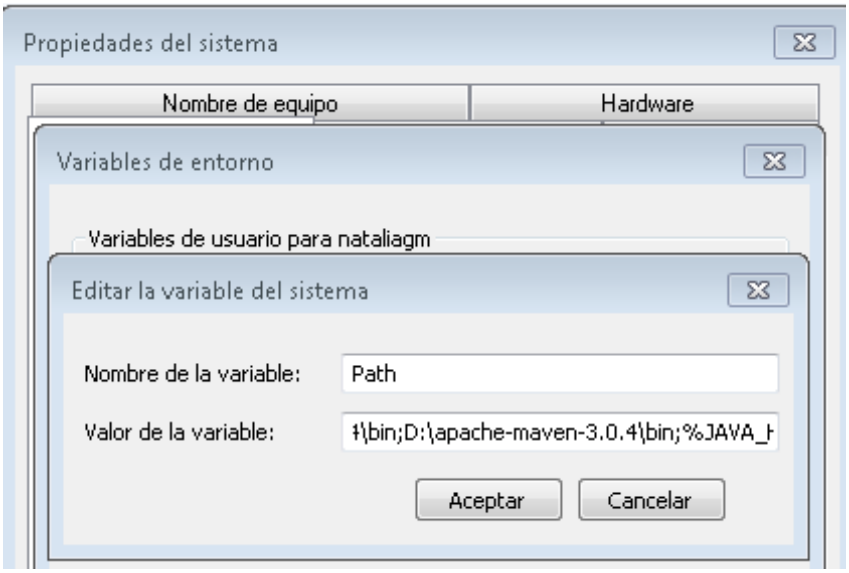


Figura 9.2 Configuara Path con apache-maven

En Eclipse se debe configurar en el menú Window->Preferences

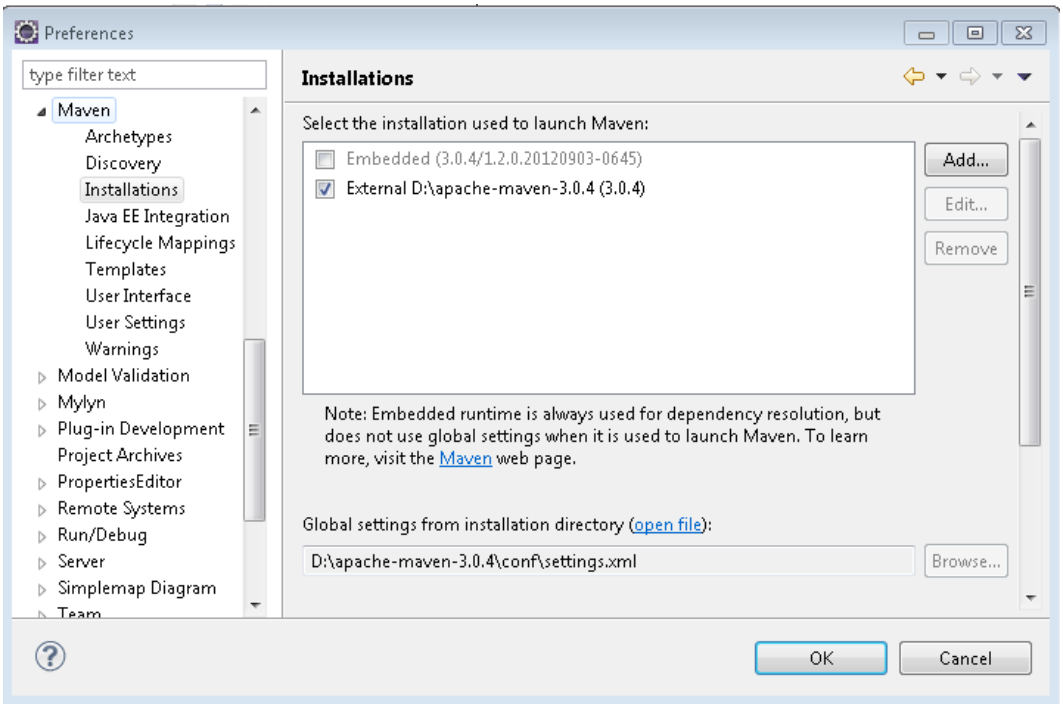


Figura 9.3 Configurar apache-maven en Eclipse

Estos serían todos los componentes que es necesario tener instalador para poder ejecutar el proyecto.

9.2 Manual de Ejecución

Para ejecutar el primer subsistema de éste proyecto una vez se dispone de toda la instalación completa tan sólo es necesario abrir el proyecto workflow-activiti-create, acceder al paquete diagrams (src/main/resorces/diagrams), donde se pueden ver los diagramas que se han ido creando.

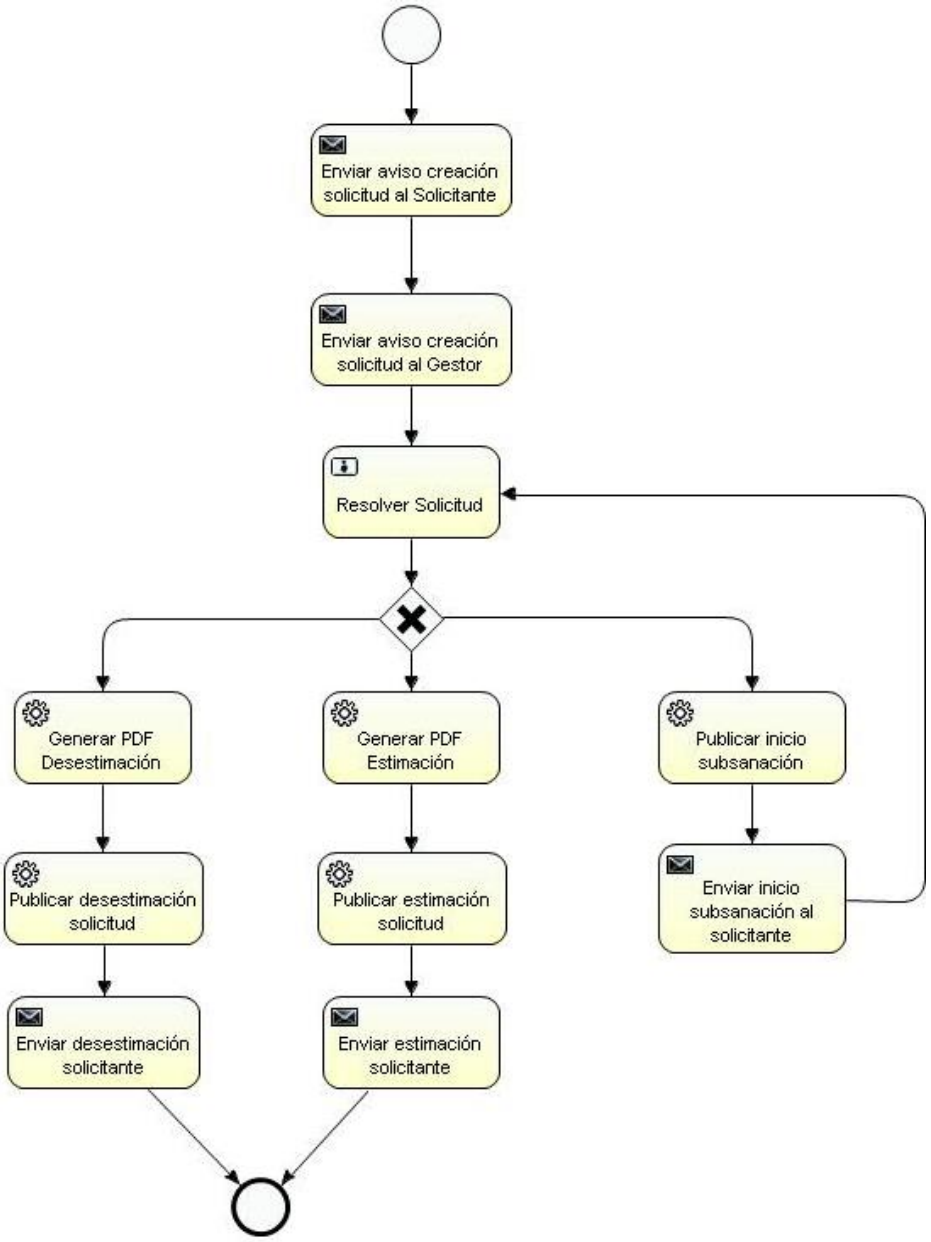


Figura 9.4 Diagrama para realizar una solicud genérica con subsanación

Una vez que hemos decidido que diagrama queremos desplegar tan sólo hay que ir a la clase Deployment (src/main/java/es.uniovi/workflow/Activiti/prueba) indicar en el método main el identificador del workflow que queremos desplegar y ejecutar la clase, automáticamente éste se creará en la base de datos vworkflowActiviti y podremos usar el identificador de ese workflow para asignarlo a las convocatorias.

Para probar el otro subsistema desarrollado en éste proyecto es necesario disponer del proyecto del gestor de trámites, ya que éste subsistema está compuesto como ya se ha comentado por dos proyectos que formar parte del tramitador.

Si se dispone del tramitador completo tan sólo sería necesario generar mediante Maven el proyecto y posteriormente desplegar la aplicación en el servidor Tomcat instalado.

9.3 Manual de Usuario

Para utilizar este proyecto una vez que se encuentra en ejecución, es decir ya tenemos desplegados los workflows que deseamos utilizar y también está en funcionamiento el gestor de trámites se deben seguir los siguientes pasos:

- Se crea una nueva convocatoria la cual tiene por defecto asociado el workflow de solicitud genérica, si se desea cambiar eso se puede cambiar en base de datos donde se acaba de puede asignar otro tipo de procedimiento, esto puede hacerlo tan sólo el Administrador del Sistema.
- Una vez que se encuentra la convocatoria creada, los usuarios pueden acceder a "Presentar Solicitudes" y ver las convocatorias disponibles, si hay alguna para su perfil, podrían crear una solicitud. Cuando se inicia la solicitud se crea una nueva instancia del workflow asignado.
- El Administrador puede acceder al "Tramitador" y ver las solicitudes que tienen para la Organización que está asociado su perfil. Desde aquí puede revisar toda la documentación, y posteriormente aceptar, rechazar o solicitar una subsanación al usuario.
- El usuario en cualquier momento puede acceder a "Cómo va lo mío" y consultar el estado de su solicitud.

9.4 Manual del Programador

Como es posible que un futuro, y puede que no muy lejano, se requiera añadir funcionalidad a este proyecto o modificar alguna de la ya existente, voy a comentar algunos de los aspectos más importantes de este desarrollo, para que sea más fácil la creación o modificación de los workflows con el gestor de trámites.

El proyecto sigue la estructura básica de un proyecto creado con Java creado Maven, por lo que para entender la organización del mismo tan sólo hace falta tener conocimientos de dicho lenguaje.

Por otra parte he intentado seguir todos los convenios de nomenclatura de dicho lenguaje a la vez de intentar que todos los nombres de métodos, variables, clases, ficheros, paquetes etc, sean lo más explicativos posibles, es decir que al menos con tan sólo el nombre ya tener una idea de que puede tratar o hacer dicho elemento.

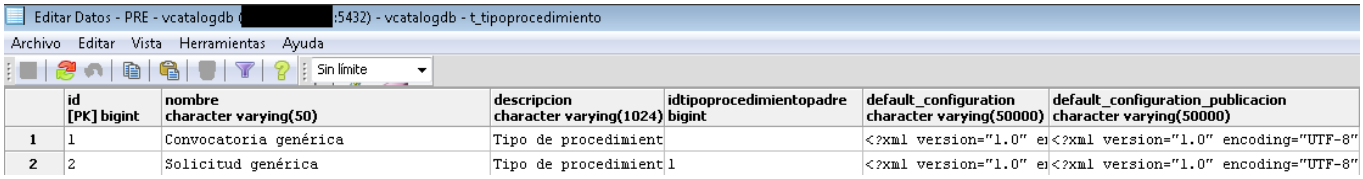
Para facilitar la comprensión del código se ha comentado los métodos más importantes. Se ha intentado que los métodos fueran lo más reducidos posible.

En cuanto al primer subsistema para hacerlo funcionar y entenderlo tan sólo es necesario tener conocimientos de Activiti, y como funciona cada tipo de actividad.

Para utilizar el segundo subsistema hay que comprender que realiza cada método del interfaz de WorflowService, para ello esto se encuentra comentado y explicado en el Javadoc del proyecto.

En cuando a la asignación de workflows a las convocatorias es necesario entender algunas tablas de la plataforma eGob!

En la base de datos vcatalog en la tabla t_tipoprocedimiento, es donde se definen los tipos de convocatorias y sus solicitudes.



	id [PK] bigint	nombre character varying(50)	descripcion character varying(1024)	idtipoprocedimientopadre bigint	default_configuration character varying(50000)	default_configuration_publicacion character varying(50000)
1	1	Convocatoria genérica	Tipo de procedimient		<?xml version="1.0" e	<?xml version="1.0" encoding="UTF-8"
2	2	Solicitud genérica	Tipo de procedimient	1	<?xml version="1.0" e	<?xml version="1.0" encoding="UTF-8"

Figura 9.5 Configuración Convocatorias y Solicitudes

Es aquí cuando se define la configuración de la solicitud dónde se indica el workflow que se debe de instanciar.

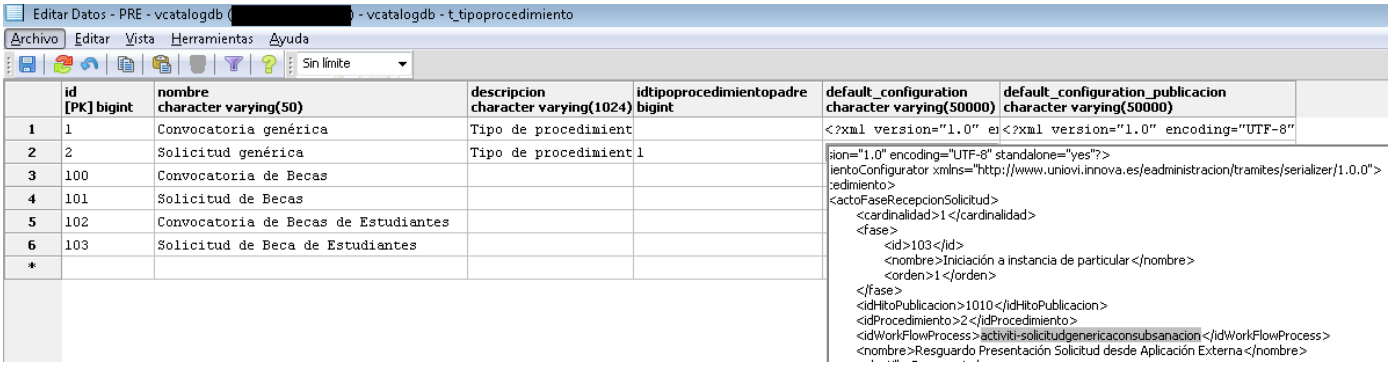


Figura 9.6 Configuración solicitud con workflow

Esto sería todo lo necesario para entender el funcionamiento y poder seguir trabajando con este proyecto, el resto de conocimiento necesario ya depende de la operación que se quiera realizar con eGob!, los conocimientos ya depende de ésta plataforma y no de los módulos realizados en éste proyecto.

Capítulo 10. Conclusiones y Ampliaciones

10.1 Conclusiones

Una vez finalizado el proyecto tengo una sensación muy buena sobre él y sobre cómo se ha finalizado, ya que uno de los workflows ya se encuentra en producción. Aunque también es cierto que los inicios y el final del proyecto han sido buenos, ha habido partes intermedias en las que las cosas no salían en los tiempos que deberían salir y por tanto me he llegado a agobiar en algunos momentos.

El inicio del proyecto lo cogí con muchas ganas ya que me pareció un tema muy interesante, y además de algo real, una integración con un proyecto real, él cuál me permitió integrarme ya en un entorno de trabajo, y desarrollar el proyecto en dicho entorno.

Después vinieron algunos de los problemas, ya que la primera idea de que el proyecto se integrara con uno ya existente, pues tiene también sus inconvenientes ya que antes de ponerme a desarrollar el proyecto propiamente dicho tienes que entender cómo funciona lo ya existente y como se pueden acoplar las dos partes. Esto supuso retraso en los tiempos, ya que me costó un poco entender cómo funcionaba y conseguir hacerlo acoplar todo junto.

Posteriormente vino la fase de estudio de la situación actual para ver que tecnología era la mejor para integrar, en éste momento surgieron bastantes dudas ya de primeras se pensaba más en JBPM pero aunque actualmente podría ser una buena opción, cuando se comenzó en desarrollo de éste proyecto no era así, ya que se encontraba en una fase de transición, ya que parte de los desarrolladores se habían ido y juntado para formar Activiti. Por este motivo me decanté más por Activiti ya que jbpm en ese momento era un poco inestable.

La fase de desarrollo con algunos de los problemas típicos de encontrarte con algo que no conoces hasta el momento, pero por lo demás avanzó sin mayores inconvenientes.

Por tanto ahora una vez finalizado el proyecto me encuentro satisfecha con el trabajo realizado.

10.2 Ampliaciones

Algunas de las posibles ampliaciones de éste proyecto sería la del desarrollo de nuevos workflows, ya que así se podrían realizar de forma más sencilla muchos más procesos administrativos de la Universidad de Oviedo.

Otra posible ampliación de éste proyecto sería integrar el funcionamiento con workflow en otras partes de la Universidad y no tan sólo en el gestor de trámites genéricos que existe actualmente.

Otra posible ampliación sería integran el monitor de procesos de workflow de Activiti en el gestor de trámites, ya que así la información que se proporcionaría en cuando a todos los estados por los que pasa la tarea sería mucho más completa y la experiencia con el usuario sería mucho mayor, aunque esto también supone que habría que formar a las personas que trabajan con el gestor de trámites para entender y saber manejar esa información.

Capítulo 11. Presupuesto

A continuación se puede ver un presupuesto detallo de los costes que conlleva el desarrollo del proyecto.

Item	Subitem	Concepto	Descripción	Precio	Unidades	Total
01		**Desplazamientos**	**Desplazamientos**	**20,00 €**	**5**	**100,00 €**
02		**Amortización de material utilizado**				**520,00 €**
	001		PC portátil de desarrollo	50,00 €	8	400,00 €
	002		Pantalla auxiliar	15,00 €	8	120,00 €
03		**Licencias de herramientas utilizadas**				**699,60 €**
	001		Microsoft Office Professional	699,60 €	1	699,60 €
04		**Formación**				**5.300,00 €**
	001		BPMN	20,00 €	90	1.800,00 €
	002		Estudio de alternativas	10,00 €	150	1.500,00 €
	003		Activiti	20,00 €	100	2.000,00 €
05		**Desarrollo**				**87.120,00 €**
	001		Jefe proyectos	70,00 €	120	8.400,00 €
	002		Equipo de Desarrollo	30,00 €	2032	60.960,00 €
	003		Analista	50,00 €	304	15.200,00 €
	004		Equipo de Sistemas	40,00 €	64	2.560,00 €
06		**Beneficio 6%**		**93.739,60 €**	**6%**	**5.624,38 €**
		IVA 21%		**99.363,98 €**	**21%**	**20.866,43 €**
		Total sin IVA				**99.363,98 €**
		TOTAL				**120.230,41 €**

Tabla 11-1 Presupuesto detallado

Capítulo 12. Referencias Bibliográficas

12.1 Referencias en Internet

[BPMN] "Business Process Model and Notation" http://www.bpmn.org/. 2013

[BPMN-Diagram] "Business Process Model Notation, Diagram" http://bpt.hpi.uni-potsdam.de/pub/Public/BPMNCorner/BPMN1_1_Poster_EN.pdf. 2013

[BPMN2.0-Diagram] "Business Process Model Notation 2.0, Diagram" http://bpmb.de/images/BPMN2_0_Poster_EN.pdf. 2013

[Zapata] "BPMN 2.0" http://www.scribd.com/doc/72853677/BPMN-2-0. 2013

[Activiti] "Activiti BPM Platform" http://www.activiti.org/. 2013

[BonitaSoft] "BonitaSoft Community" http://www.bonitasoft.org/. 2013

[BonitaSoft] "BonitaSoft Open Source" http://www.bonitasoft.com/. 2013

[Pupier] "Time to build and test results 3x faster - how we did it" http://www.slideshare.net/BonitaSoft. 2013

[Bertrand] "Del modelado a la automatización y monitorización de procesos de negocio" (Figura 2) http://www.di.uniovi.es/~cueva/conferencias/EBertrand.pdf. 2013

[Intalio] "Intalio" http://www.intalio.com/. 2013

[Intalio] "Intalio Community" http://community.intalio.com/. 2013

[jBPM] "jBPM Documentation" http://www.jboss.org/jbpm/documentation. 2013

[JDPL] "jBPM Process Definition Language" http://docs.jboss.org/jbpm/v3/userguide/jpdl.html. 2013

[Roldán] "Principales estándares BPM y Suite Open Source" http://www.novayre.es/articulos/bpm-opensource.html. 2013

[jotadeveloper] "Análisis de algunos BPM Open Source para la Empresa" http://blog.jotadeveloper.com/2008/08/19/analisis-de-algunos-bpm-opensource-para-la-empresa/. 2013

[suite101] "Best bpm software download and comparison" http://www.suite101.com/content/best-bpm-software-download-and-comparison-2011-reviews-a340615. 2013

[Nie] "Open Source Power on BPM - A Comparison of JBoss jBPM and Intalio BPMS" http://jannekorhonen.fi/project_report_final_BPMS.pdf. 2013

[SoftwareForEnterprise] "List of top open source BPM / workflow solution" http://www.softwareforenterprise.us/2009/03/13/list-of-top-open-source-bpm-workflow-solution/. 2013

[God] "JBPM - Jboss Business Proccess Manager" http://www.fugu.ec/productos-y-servicios/consultoria/127-jbpm-jboss-business-proccess-manager-bpm-ecuador.html. 2013

[Garcia] "jBPM (Modelado de procesos de negocio en entornos informáticos)" http://es.scribd.com/doc/50262685/jBPM-Modelado-de-procesos-de-negocio-en-entornos-informaticos. 2013

[Sánchez] "jBPM Form Builder: generación de formularios para jBPM5 y su integración en Guvnor" http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jbpmFormBuilderGuvnorIntegration. 2013

[Canales] "Revisión de jBPMN5" http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jbpm. 2013

[Wikipedia] "Java (Lenguaje de programación)" http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n). 2014

[Apache] "Apache Tomcat 6" http://tomcat.apache.org/download-60.cgi. 2014

[Fernández] "Representación, Interpretación y Aprendizaje de Flujos de Trabajo basado en Actividades para la estandarización de Vías Clínicas" (Figura 1) http://riunet.upv.es/bitstream/handle/10251/4562/tesisUPV3021.pdf. 2013

[Signavio] "Signavio" http://www.signavio.com/es/. 2013

[PostgreSQL] "PostgreSQL" http://www.postgresql.org.es/. 2014

[Apache] "Apache Maven" http://maven.apache.org/. 2014

Capítulo 13. Apéndices

13.1 Glosario y Diccionario de Datos

I. **Activiti**: Es una suite BPM que está respaldada por Alfresco y tiene integración directa con dicha plataforma, se trata también de una herramienta gratuita

II. **BPM**: Business Process Management, es la gestión de procesos de negocio.

III. **BPMN**: Business Process Modeling and Notation, es un lenguaje estándar de modelado de procesos de negocio.

IV. **BPMN 2.0**: Business Process Modeling and Notation, es la segunda versión que ha salido en 2009, la cual mejora la estandarización de los componentes y añade algunos nuevos.

V. **BPMS**: Business Process Management Suite. Son las herramientas que trabajan con BPM y añade un conjunto más de herramientas para añadir más funcionalidad.

VI. **jBPM**: jBoss Business Process Management. Es la suite de trabajo que proporciona jBoss para trabajar con BPM, es una herramienta open source.

VII. **Workflow**: flujo de trabajo. Es el diagrama que representa el proceso de ejecución de un procediemiento.

13.2 Contenido Entregado

13.2.1 Contenidos

13.2.1.1.1 Introducción

Debido al nuevo tipo de entrega que se ha de realizar se ha modificado la estructura general que se recomienda.

Se realiza la entrega del documento principal (éste propio documento), y por otra parte se realiza la entrega de los anexos.

En los anexos se adjunta el resto de información del proyecto, siguiendo la siguiente estructura.

13.2.1.1.2 Estructura general del Fichero Anexos

Directorio	Contenido
./ Directorio raíz	Contiene un fichero leeme.txt explicando toda esta estructura.
./workflow-activiti	Contiene toda la estructura de directorios del proyecto para desarrollo.
./documentacion	Contiene toda la documentación asociada al proyecto en formato *.docx* ya que el fichero en formato .pdf ya se ha entregado en el documento principal
./documentacion/img	Directorio que contiene las imágenes utilizadas en la documentación.
./documentacion/diagramas	Todos los diagramas que se pueden encontrar en la documentación
./ documentacion/auxiliar	Planificación, presupuesto y ficheros anexos
./desarrollo	Código fuente de los proyectos

13.2.1.1.3 Estructura del Directorio de "desarrollo"

Directorio	Contenido
./workflow-activiti-create	Contiene el proyecto del primer subsistema creado. Ficheros que configuración del proyecto (.classpath, .project, pom.xml)
./workflow-activiti-create /doc	Contiene toda la documentación relativa al proyecto, incluyendo los ficheros generados por *Javadoc*
./workflow-activiti-create /lib	Se encuentra vacío ya que los jar necesarios se gestionan mediante Maven.
./workflow-activiti-create /src	Ficheros fuente
./workflow-activiti-create /src/main/java	Todos los ficheros *Java*, lógicamente agrupados en los paquetes correspondientes.

Directorio	Contenido
./workflow-activiti-create /src/main/resources	Este directorio contiene los ficheros de configuración además de los diagramas bmpn diseñados.
./target/classes	Directorio donde se guardan los ficheros compilados
./workflow-activiti-create /src/test/java	Contiene todas las pruebas unitarias utilizadas en el proceso de prueba automatizado.
./workflow-activiti-create /src/test/resources	Directorio base para todos los ficheros de configuración utilizados para las pruebas.
./modulosGestorTramites/workflow-impl-activiti	Contiene el proyecto del segundo subsistema creado para realizar la implementación de Activiti. Ficheros que configuración del proyecto (.classpath, .project, pom.xml)
./modulosGestorTramites/workflow-impl-activiti /doc	Contiene toda la documentación relativa al proyecto, incluyendo los ficheros generados por *Javadoc*
./modulosGestorTramites/workflow-impl-activiti /src	Ficheros fuente
./modulosGestorTramites/workflow-impl-activiti /src/main/java	Todos los ficheros *Java*, lógicamente agrupados en los paquetes correspondientes.
./modulosGestorTramites/workflow-impl-activiti /src/main/resources	Este directorio contiene los ficheros de configuración además de los diagramas bmpn diseñados.
*./modulosGestorTramites/workflow-impl-activiti /*target/classes	Directorio donde se guardan los ficheros compilados
./modulosGestorTramites/workflow-impl-activiti /src/test/java	Contiene todas las pruebas unitarias utilizadas en el proceso de prueba automatizado.
./modulosGestorTramites/workflow-impl-activiti /src/test/resources	Directorio base para todos los ficheros de configuración utilizados para las pruebas.
./modulosGestorTramites/workflow-domain	Contiene el proyecto del segundo subsistema creado para realizar una separación entre la implementación de Activiti y el gestor de trámites. Ficheros que configuración del proyecto (.classpath, .project, pom.xml)
./modulosGestorTramites/workflow-domain/doc	Contiene toda la documentación relativa al proyecto, incluyendo los ficheros generados por *Javadoc*
./modulosGestorTramites/workflow-domain/src	Ficheros fuente
./modulosGestorTramites/workflow-domain/src/main/java	Todos los ficheros *Java*, lógicamente agrupados en los paquetes correspondientes.
./modulosGestorTramites/workflow-domain/src/main/resources	Este directorio contiene los ficheros de configuración además de los diagramas bmpn diseñados.
*./modulosGestorTramites/workflow-domain/*target/classes	Directorio donde se guardan los ficheros compilados

Directorio	Contenido
./modulosGestorTramites/workflow-domain /src/test/java	Contiene todas las pruebas unitarias utilizadas en el proceso de prueba automatizado.
./modulosGestorTramites/workflow-domain/src/test/resources	Directorio base para todos los ficheros de configuración utilizados para las pruebas.

13.2.2 Ficheros de Configuración

Descripción los principales ficheros necesarios para poder hacer funcionar la aplicación.

13.2.2.1 activiti.properties

```
vworkflowActiviti.jdbc.driverClassName=org.postgresql.Driver
vworkflowActiviti.jdbc.url=jdbc:postgresql://127.0.0.1:5432/vworkflowActiviti
vworkflowActiviti.jdbc.username=vworkflowdb
vworkflowActiviti.jdbc.password=passworkd


vworkflowActiviti.mailServerHost=relay.uniovi.es
vworkflowActiviti.mailServerPort=25
vworkflowActiviti.mailServerDefaultFrom=c1nn@innova.uniovi.es
```

13.2.2.2 activiti.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">



       <bean id="dataSource"
class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
           <property name="driverClass" value="org.postgresql.Driver" />
           <property name="url"
value="jdbc:postgresql://localhost:5432/vworkflowActiviti" />
           <property name="username" value="vworkflowdb" />
           <property name="password" value="password" />
       </bean>

       <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
           <property name="dataSource" ref="dataSource" />
       </bean>


  <bean id="processEngineConfiguration"
class="org.activiti.spring.SpringProcessEngineConfiguration">
       <property name="dataSource" ref="dataSource" />
    <property name="transactionManager" ref="transactionManager" />
    <property name="databaseSchemaUpdate" value="true" />
    <property name="jobExecutorActivate" value="false" />

        <!-- SendMail configurations -->
    <property name="mailServerHost" value="relay.uniovi.es" />
    <property name="mailServerPort" value="25" />
    <property name="mailServerDefaultFrom" value="c1nn@innova.uniovi.es" />

  </bean>

  <bean id="processEngine" class="org.activiti.spring.ProcessEngineFactoryBean">
```

```
    <property name="processEngineConfiguration" ref="processEngineConfiguration" />
  </bean>

  <bean id="delegateEnviar"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.EnviarNotificacion"
/>

</beans>
```

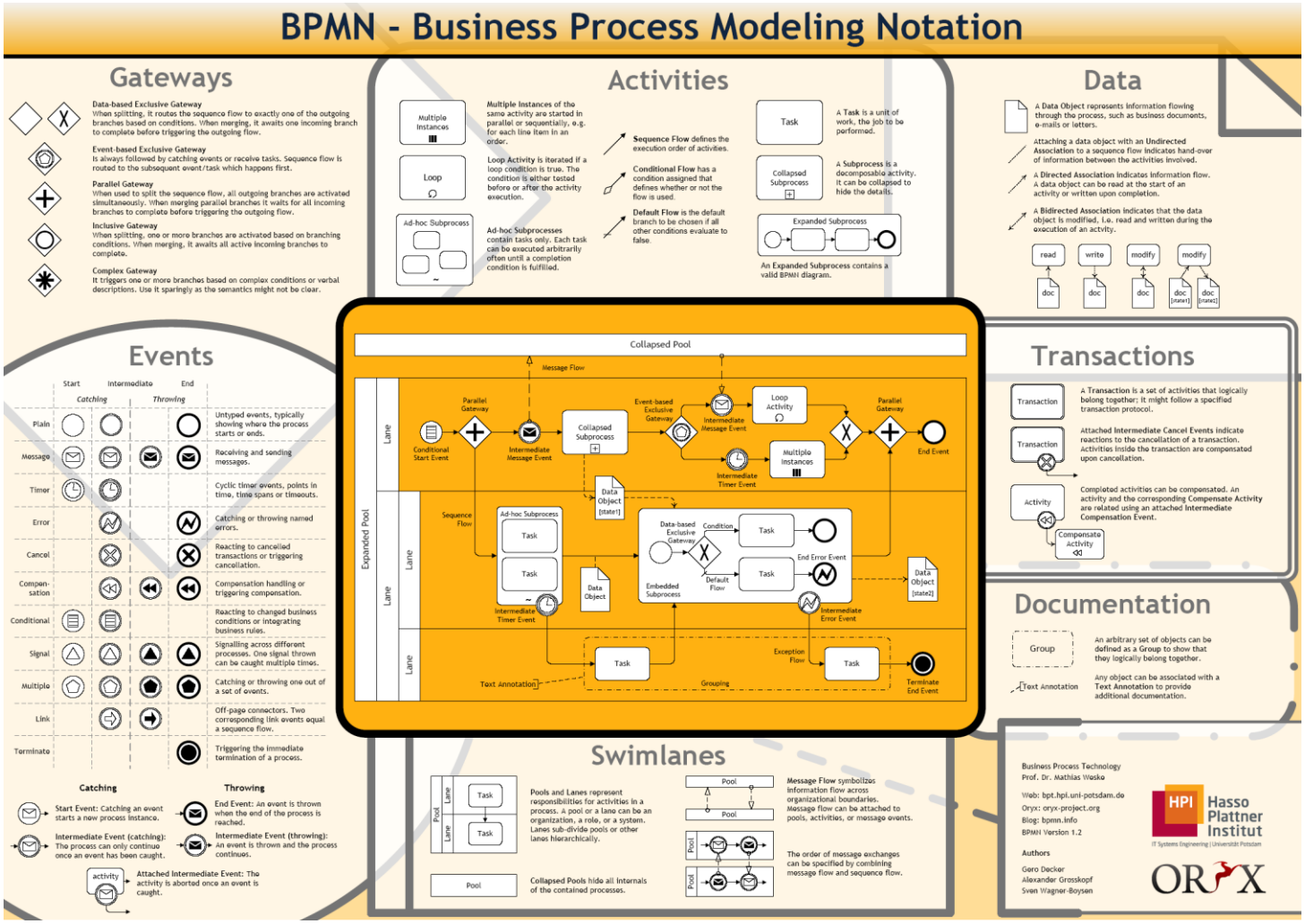
13.2.2.3 Fichero activiti-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:jee="http://www.springframework.org/schema/jee"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd
                           http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
                           http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee.xsd">

        <context:annotation-config />

        <context:component-scan base-package="com.vitruvio.workflow" />

        <bean id="vworkflowActiviti-propertyConfigurer"

        class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
                <property name="location">
                        <value>classpath:/activiti.properties</value>
                </property>
                <property name="placeholderPrefix">
                        <value>$vworkflowActiviti{</value>
                </property>
        </bean>

        <bean id="vworkflowActiviti-dataSource"
class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
                <property name="driverClass"
value="$vworkflowActiviti{vworkflowActiviti.jdbc.driverClassName}" />
                <property name="url"
value="$vworkflowActiviti{vworkflowActiviti.jdbc.url}" />
                <property name="username"
value="$vworkflowActiviti{vworkflowActiviti.jdbc.username}" />
                <property name="password"
value="$vworkflowActiviti{vworkflowActiviti.jdbc.password}" />
        </bean>

        <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
                <property name="dataSource" ref="vworkflowActiviti-dataSource" />
        </bean>

        <bean id="processEngineConfiguration"
class="org.activiti.spring.SpringProcessEngineConfiguration">
                <property name="dataSource" ref="vworkflowActiviti-dataSource" />
                <property name="transactionManager" ref="transactionManager" />
                <property name="databaseSchemaUpdate" value="false" />
                <property name="mailServerHost"
value="$vworkflowActiviti{vworkflowActiviti.mailServerHost}" />
                <property name="mailServerPort"
value="$vworkflowActiviti{vworkflowActiviti.mailServerPort}" />
                <property name="mailServerDefaultFrom"
value="$vworkflowActiviti{vworkflowActiviti.mailServerDefaultFrom}" />
        </bean>

        <bean id="processEngine" class="org.activiti.spring.ProcessEngineFactoryBean">
```

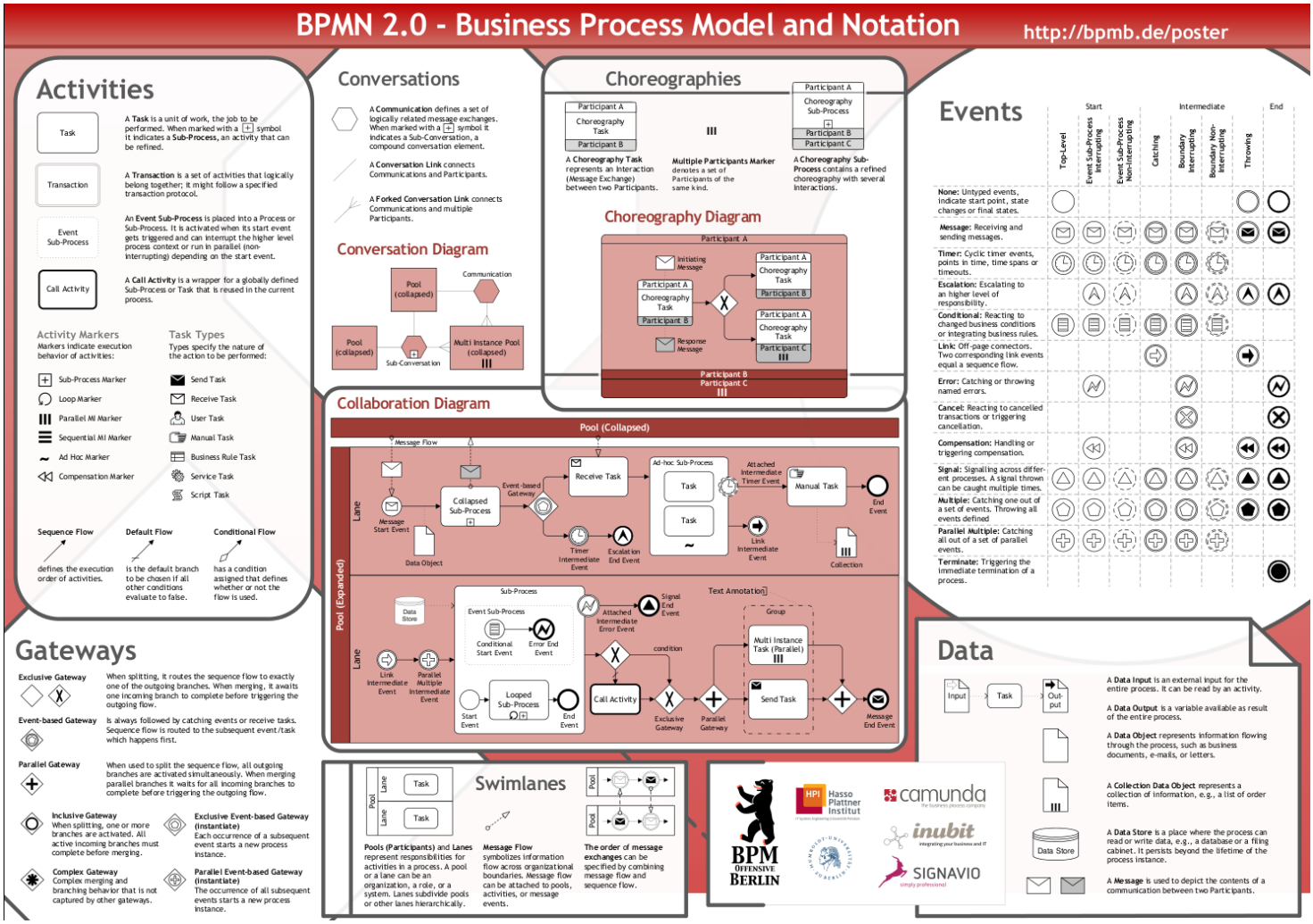
```
            <property name="processEngineConfiguration"
ref="processEngineConfiguration" />
        </bean>

        <bean id="runtimeService" factory-bean="processEngine" factory-
method="getRuntimeService" />
        <bean id="taskService" factory-bean="processEngine" factory-
method="getTaskService" />

</beans>
```

13.3 Anexos BPMN

13.3.1 BPMN



13.3.2 BPMN2.0


BPMN 2.0 - Business Process Model and Notation
http://bpmb.de/poster
Activities
Task
A Task is a unit of work, the job to be performed. When marked with a [+] symbol it indicates a Sub-Process, an activity that can be refined.
Transaction
A Transaction is a set of activities that logically belong together; it might follow a specified transaction protocol.
Event Sub-Process
An Event Sub-Process is placed into a Process or Sub-Process. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.
Call Activity
A Call Activity is a wrapper for a globally defined Sub-Process or Task that is reused in the current process.
Activity Markers
Markers indicate execution behavior of activities:
Sub-Process Marker
Loop Marker
Parallel MI Marker
Sequential MI Marker
Ad Hoc Marker
Compensation Marker
Task Types
Types specify the nature of the action to be performed:
Send Task
Receive Task
User Task
Manual Task
Business Rule Task
Service Task
Script Task
Sequence Flow
defines the execution order of activities.
Default Flow
is the default branch to be chosen if all other conditions evaluate to false.
Conditional Flow
has a condition assigned that defines whether or not the flow is used.
Gateways
Exclusive Gateway
When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.
Event-based Gateway
is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.
Parallel Gateway
When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.
Inclusive Gateway
When splitting, one or more branches are activated. All active incoming branches must complete before merging.
Exclusive Event-based Gateway (instantiate)
Each occurrence of a subsequent event starts a new process instance.
Complex Gateway
Complex merging and branching behavior that is not captured by other gateways.
Parallel Event-based Gateway (instantiate)
The occurrence of all subsequent events starts a new process instance.
Conversations
A Communication defines a set of logically related message exchanges. When marked with a [+] symbol it indicates a Sub-Conversation, a compound conversation element.
A Conversation Link connects Communications and Participants.
A Forked Conversation Link connects Communications and multiple Participants.
Conversation Diagram
Pool (collapsed)
Communication
Pool (collapsed)
Sub-Conversation
Multi Instance Pool (collapsed)
Choreographies
Participant A
Choreography Task
Participant B
A Choreography Task represents an Interaction (Message Exchange) between two Participants.
Multiple Participants Marker denotes a set of Participants of the same kind.
Participant A
Choreography Sub-Process
Participant B
Participant C
A Choreography Sub-Process contains a refined choreography with several interactions.
Choreography Diagram
Participant A
Initiating Message
Participant A
Choreography Task
Participant B
Participant A
Choreography Task
Participant B
Response Message
Participant A
Choreography Task
Participant C
Participant B
Participant C
Collaboration Diagram
Pool (Collapsed)
Message Flow
Lane
Message Start Event
Collapsed Sub-Process
Event-based Gateway
Receive Task
Ad-hoc Sub-Process
Task
Task
Attached Intermediate Timer Event
Manual Task
End Event
Data Object
Timer Intermediate Event
Escalation End Event
Link Intermediate Event
Collection
Pool (Expanded)
Lane
Data Store
Sub-Process
Event Sub-Process
Conditional Start Event
Error End Event
Attached Intermediate Error Event
Signal End Event
Text Annotation
Group
Link Intermediate Event
Parallel Multiple Intermediate Event
Looped Sub-Process
Start Event
End Event
Call Activity
condition
Exclusive Gateway
Parallel Gateway
Multi Instance Task (Parallel)
Send Task
Message End Event
Swimlanes
Pool
Lane
Task
Task
Pools (Participants) and Lanes represent responsibilities for activities in a process. A pool or a lane can be an organization, a role, or a system. Lanes subdivide pools or other lanes hierarchically.
Message Flow symbolizes information flow across organizational boundaries. Message flow can be attached to pools, activities, or message events.
The order of message exchanges can be specified by combining message flow and sequence flow.
Events
Start
Intermediate
End
Top-Level
Event Sub-Process Interrupting
Event Sub-Process Non-Interrupting
Catching
Boundary Interrupting
Boundary Non-Interrupting
Throwing
None: Untyped events, indicate start point, state changes or final states.
Message: Receiving and sending messages.
Timer: Cyclic timer events, points in time, time spans or timeouts.
Escalation: Escalating to an higher level of responsibility.
Conditional: Reacting to changed business conditions or integrating business rules.
Link: Off-page connectors. Two corresponding link events equal a sequence flow.
Error: Catching or throwing named errors.
Cancel: Reacting to cancelled transactions or triggering cancellation.
Compensation: Handling or triggering compensation.
Signal: Signalling across different processes. A signal thrown can be caught multiple times.
Multiple: Catching one out of a set of events. Throwing all events defined
Parallel Multiple: Catching all out of a set of parallel events.
Terminate: Triggering the immediate termination of a process.
Data
Input
Task
Output
A Data Input is an external input for the entire process. It can be read by an activity.
A Data Output is a variable available as result of the entire process.
A Data Object represents information flowing through the process, such as business documents, e-mails, or letters.
A Collection Data Object represents a collection of information, e.g., a list of order items.
Data Store
A Data Store is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.
A Message is used to depict the contents of a communication between two Participants.
BPM Offensive Berlin
Hasso Plattner Institut
camunda
inubit
SIGNAVIO

13.4 Código Fuente

A continuación se proporciona el código fuente de las clases más importante del proyecto, el código fuente en su totalidad se puede consultar en el proyecto.

13.4.1 Subsistema workfow-activiti-create:

13.4.1.1 Fichero "Deployment.java":

```
package es.uniovi.workflow.activiti.prueba;

import org.activiti.engine.ProcessEngine;
import org.activiti.engine.ProcessEngines;
import org.activiti.engine.RepositoryService;

/**
 * Clase para desplegar en base de datos los workflow creados
 *
 * @author nataliagm
 *
 */
public class Deployment {

	private String path = "D:/diagramsActiviti/";
	private String extension = ".bpmn";

	private ProcessEngine processEngine;
	private RepositoryService repositoryService;

	public Deployment() {
		processEngine = ProcessEngines.getDefaultProcessEngine();
		repositoryService = processEngine.getRepositoryService();
	}

	public void deploy(String idProcess) {
		repositoryService.createDeployment()
				.addClasspathResource(path + idProcess + extension)
				.name(idProcess).deploy();
	}

	public long numProcesos() {
		return repositoryService.createProcessDefinitionQuery().count();
	}

	public void close() {
		processEngine.close();
	}

	public static void main(String[] args) {
		Deployment deployService = new Deployment();

		deployService.deploy("activiti-firmaunusuario");
		deployService.deploy("activiti-firmadosusuarios");
		deployService.deploy("activiti-solicitudgenerica");
		deployService.deploy("activiti-solicitarpermiso");
		deployService.deploy("activiti-solicitudgenericaconsubsanacion");
		deployService.deploy("activiti-solicitudgenericaconsubsanacionAsyn");

		System.out.println("Número de procesos totales definidos: "
				+ deployService.numProcesos());

		deployService.close();

	}
}
```

13.4.1.2 Fichero activiti-firmadosusuarios.bpmn

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:activiti="http://activiti.org/bpmn"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
typeLanguage="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.w3.org/1999/XPath"
targetNamespace="http://www.activiti.org/test">
  <process id="activiti-firmadosusuarios" name="Process Firma dos usuarios"
isExecutable="true">
    <startEvent id="startFirma" name="Start"></startEvent>
    <userTask id="FIRMA_1" name="Firma prueba_peri">
      <extensionElements>
        <activiti:taskListener event="create"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.AsignarAPersonal"></
activiti:taskListener>
      </extensionElements>
    </userTask>
    <serviceTask id="MAIL_prueba_peri" name="Mail Task Prueba Peri"
activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:string>nataliagm@innova.uniovi.es</activiti:string>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:string>Nueva tarea pendiente</activiti:string>
        </activiti:field>
        <activiti:field name="html">
          <activiti:string>&lt;html&gt;
                                                &lt;body&gt;
                                                        &lt;p&gt;Buenos días,
&lt;br/&gt;&lt;br/&gt;
                                                                Tiene usted una nueva
tarea pendiente de firmar.&lt;br/&gt;
                                                                Acuda a la bandeja de
entrada del &lt;a href="http://localhost:8080/becasuniovi-war-1.2.8-SNAPSHOT/"&gt;Gestor
de becas&lt;/a&gt;.

        &lt;br/&gt;&lt;br/&gt;Un saludo.
                                                        &lt;/p&gt;
                                                        &lt;p&gt;Universidad de
Oviedo&lt;br/&gt;
                                                                Gestor de becas
                                                                &lt;img
src="http://localhost:8080/becasuniovi-war-1.2.8-SNAPSHOT/login/images/escudoUniovi.gif"
alt="logo UNIOVI" /&gt;
                                                        &lt;/p&gt;
                                                &lt;/body&gt;
                                        &lt;/html&gt;</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <serviceTask id="MAIL_UO213296" name="Mail Task UO213296" activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:string>UO213296@uniovi.es</activiti:string>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:string>Nueva tarea pendiente</activiti:string>
        </activiti:field>
        <activiti:field name="html">
          <activiti:string>&lt;html&gt;
                                        &lt;body&gt;

                                                &lt;p&gt;Buenos
días,&lt;br/&gt;&lt;br/&gt;

                                                Tiene usted una nueva tarea
pendiente de firmar.&lt;br/&gt;
                                                Acuda a la bandeja de entrada del
&lt;a href="http://localhost:8080/becasuniovi-war-1.2.8-SNAPSHOT/"&gt;Gestor de
becas&lt;/a&gt;.
```

```
                                                    &lt;br/&gt;&lt;br/&gt;Un
saludo.&lt;/p&gt;
                                                    &lt;p&gt;Universidad de
Oviedo&lt;br/&gt;
                                                    Gestor de becas
                                                    &lt;img
src="http://localhost:8080/becasuniovi-war-1.2.8-SNAPSHOT/login/images/escudoUniovi.gif"
alt="logo UNIOVI" /&gt;
                                                    &lt;/p&gt;
                                                &lt;/body&gt;
                                            &lt;/html&gt;</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <userTask id="FIRMA_2" name="Firma UO213296"
activiti:assignee="UO213296"></userTask>
    <endEvent id="finFirma" name="finFirma"></endEvent>
    <sequenceFlow id="flow1" sourceRef="startFirma"
targetRef="MAIL_prueba_peri"></sequenceFlow>
    <sequenceFlow id="flow2" sourceRef="MAIL_prueba_peri"
targetRef="FIRMA_1"></sequenceFlow>
    <sequenceFlow id="flow3" sourceRef="FIRMA_1"
targetRef="MAIL_UO213296"></sequenceFlow>
    <sequenceFlow id="flow4" sourceRef="MAIL_UO213296"
targetRef="FIRMA_2"></sequenceFlow>
    <sequenceFlow id="flow5" sourceRef="FIRMA_2" targetRef="finFirma"></sequenceFlow>
  </process>
  <bpmndi:BPMNDiagram id="BPMNDiagram_activiti-firmadosusuarios">
    <bpmndi:BPMNPlane bpmnElement="activiti-firmadosusuarios" id="BPMNPlane_activiti-
firmadosusuarios">
      <bpmndi:BPMNShape bpmnElement="startFirma" id="BPMNShape_startFirma">
        <omgdc:Bounds height="35.0" width="35.0" x="50.0" y="180.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="FIRMA_1" id="BPMNShape_FIRMA_1">
        <omgdc:Bounds height="55.0" width="105.0" x="330.0" y="170.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="MAIL_prueba_peri" id="BPMNShape_MAIL_prueba_peri">
        <omgdc:Bounds height="55.0" width="105.0" x="180.0" y="170.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="MAIL_UO213296" id="BPMNShape_MAIL_UO213296">
        <omgdc:Bounds height="55.0" width="105.0" x="490.0" y="170.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="FIRMA_2" id="BPMNShape_FIRMA_2">
        <omgdc:Bounds height="55.0" width="105.0" x="660.0" y="170.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="finFirma" id="BPMNShape_finFirma">
        <omgdc:Bounds height="35.0" width="35.0" x="820.0" y="180.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNEdge bpmnElement="flow1" id="BPMNEdge_flow1">
        <omgdi:waypoint x="85.0" y="197.0"></omgdi:waypoint>
        <omgdi:waypoint x="180.0" y="197.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow2" id="BPMNEdge_flow2">
        <omgdi:waypoint x="285.0" y="197.0"></omgdi:waypoint>
        <omgdi:waypoint x="330.0" y="197.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow3" id="BPMNEdge_flow3">
        <omgdi:waypoint x="435.0" y="197.0"></omgdi:waypoint>
        <omgdi:waypoint x="490.0" y="197.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow4" id="BPMNEdge_flow4">
        <omgdi:waypoint x="595.0" y="197.0"></omgdi:waypoint>
        <omgdi:waypoint x="660.0" y="197.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow5" id="BPMNEdge_flow5">
        <omgdi:waypoint x="765.0" y="197.0"></omgdi:waypoint>
        <omgdi:waypoint x="820.0" y="197.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
    </bpmndi:BPMNPlane>
  </bpmndi:BPMNDiagram>
</definitions>
```

13.4.1.3 Fichero activiti-solicitapermiso.bpmn

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:activiti="http://activiti.org/bpmn"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
typeLanguage="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.w3.org/1999/XPath"
targetNamespace="http://www.activiti.org/test">
  <process id="activiti-solicitarpermiso" name="Proceso para la solicitud de un Permiso"
isExecutable="true">
    <startEvent id="startevent1" name="Start"></startEvent>
    <serviceTask id="enviarNotificacion" name="Enviar notificación publicador"
activiti:delegateExpression="${enviarNotificacionActivitiTask}"></serviceTask>
    <userTask id="confirmaDepartamento" name="Confirmar por el departamento">
      <extensionElements>
        <activiti:taskListener event="create"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.AsignarDepartamento"
></activiti:taskListener>
      </extensionElements>
    </userTask>
    <serviceTask id="servicetask1" name="Publicar confirmación pepartamento"
activiti:delegateExpression="${publicarHitoActivitiTask}">
      <extensionElements>
        <activiti:field name="codigoHito">
          <activiti:string>CONFIRMACION_DEPARTAMENTO</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <userTask id="confirmaPersonal" name="Confirmar por personal">
      <extensionElements>
        <activiti:taskListener event="create"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.AsignarAPersonal"></
activiti:taskListener>
      </extensionElements>
    </userTask>
    <userTask id="confirmaGerencia" name="Confirmar por gerencia">
      <extensionElements>
        <activiti:executionListener event="start"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.FinalizarWorkflowAct
ivitiTask"></activiti:executionListener>
        <activiti:taskListener event="create"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.AsignarAGerencia"></
activiti:taskListener>
      </extensionElements>
    </userTask>
    <serviceTask id="publicaAceptacionPermiso" name="Publicar aceptacion permiso"
activiti:delegateExpression="${publicarHitoActivitiTask}">
      <extensionElements>
        <activiti:field name="codigoHito">
          <activiti:string>ACEPTACION_SOLICITUD</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <serviceTask id="enviarConfirmacionSolicitante" name="Enviar confirmación
solicitante" activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:string>Nueva tarea pendiente</activiti:string>
        </activiti:field>
        <activiti:field name="html">
          <activiti:string>&lt;html&gt;
                                                &lt;body&gt;
                                                        &lt;p&gt;Buenos días,
&lt;br/&gt;&lt;br/&gt;
                                                                Tiene usted una nueva
tarea pendiente de firmar.&lt;br/&gt;
                                                                Acuda a la bandeja de
entrada del &lt;a href="http://localhost:8080/becasuniovi-war-1.2.8-SNAPSHOT/"&gt;Gestor
de becas&lt;/a&gt;.
```

```
        &lt;br/&gt;&lt;br/&gt;Un saludo.
                                                                &lt;/p&gt;
                                                                &lt;p&gt;Universidad de
Oviedo&lt;br/&gt;
                                                                        Gestor de becas
                                                                        &lt;img
src="http://localhost:8080/becasuniovi-war-1.2.8-SNAPSHOT/login/images/escudoUniovi.gif"
alt="logo UNIOVI" /&gt;
                                                                &lt;/p&gt;
                                                        &lt;/body&gt;
                                                &lt;/html&gt;</activiti:string>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailSolicitanteWorkflowActivitiTask}"></activiti:executionL
istener>
      </extensionElements>
    </serviceTask>
    <endEvent id="endevent1" name="End"></endEvent>
    <sequenceFlow id="flow1" sourceRef="startevent1"
targetRef="enviarNotificacion"></sequenceFlow>
    <sequenceFlow id="flow2" sourceRef="enviarNotificacion"
targetRef="confirmaDepartamento"></sequenceFlow>
    <sequenceFlow id="flow3" sourceRef="confirmaDepartamento"
targetRef="exclusivegateway1"></sequenceFlow>
    <sequenceFlow id="flow4" sourceRef="servicetask1"
targetRef="confirmaPersonal"></sequenceFlow>
    <sequenceFlow id="flow5" sourceRef="confirmaPersonal"
targetRef="exclusivegateway2"></sequenceFlow>
    <sequenceFlow id="flow6" sourceRef="confirmaGerencia"
targetRef="exclusivegateway3"></sequenceFlow>
    <sequenceFlow id="flow7" sourceRef="publicaAceptacionPermiso"
targetRef="enviarConfirmacionSolicitante"></sequenceFlow>
    <sequenceFlow id="flow8" sourceRef="enviarConfirmacionSolicitante"
targetRef="endevent1"></sequenceFlow>
    <exclusiveGateway id="exclusivegateway1" name="Exclusive Gateway"
default="flow9"></exclusiveGateway>
    <serviceTask id="publicarRechazoPermiso" name="Publicar rechazo permiso"
activiti:exclusive="false" activiti:delegateExpression="${publicarHitoActivitiTask}">
      <extensionElements>
        <activiti:field name="codigoHito">
          <activiti:string>CANCELACION_SOLICITUD</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <sequenceFlow id="flow9" sourceRef="exclusivegateway1"
targetRef="servicetask1"></sequenceFlow>
    <sequenceFlow id="flow10" sourceRef="exclusivegateway1"
targetRef="publicarRechazoPermiso">
      <conditionExpression
xsi:type="tFormalExpression"><![CDATA[${cancelar}]]></conditionExpression>
    </sequenceFlow>
    <exclusiveGateway id="exclusivegateway2" name="Exclusive Gateway"
default="flow11"></exclusiveGateway>
    <sequenceFlow id="flow11" sourceRef="exclusivegateway2"
targetRef="confirmaGerencia"></sequenceFlow>
    <sequenceFlow id="flow12" sourceRef="exclusivegateway2"
targetRef="publicarRechazoPermiso">
      <conditionExpression
xsi:type="tFormalExpression"><![CDATA[${cancelar}]]></conditionExpression>
    </sequenceFlow>
    <exclusiveGateway id="exclusivegateway3" name="Exclusive Gateway"
default="flow13"></exclusiveGateway>
    <sequenceFlow id="flow13" sourceRef="exclusivegateway3"
targetRef="publicaAceptacionPermiso"></sequenceFlow>
    <sequenceFlow id="flow14" sourceRef="exclusivegateway3"
targetRef="publicarRechazoPermiso">
      <conditionExpression
xsi:type="tFormalExpression"><![CDATA[${cancelar}]]></conditionExpression>
    </sequenceFlow>
    <serviceTask id="enviarCancelacionSolicitante" name="Enviar cancelación solicitante"
activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
```

```
          <activiti:string>Denegado permiso vacaciones</activiti:string>
        </activiti:field>
        <activiti:field name="html">
          <activiti:string>prueba</activiti:string>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailSolicitanteWorkflowActivitiTask}"></activiti:executionL
istener>
      </extensionElements>
    </serviceTask>
    <sequenceFlow id="flow15" sourceRef="publicarRechazoPermiso"
targetRef="enviarCancelacionSolicitante"></sequenceFlow>
    <sequenceFlow id="flow16" sourceRef="enviarCancelacionSolicitante"
targetRef="endevent1"></sequenceFlow>
  </process>
  <bpmndi:BPMNDiagram id="BPMNDiagram_activiti-solicitarpermiso">
    <bpmndi:BPMNPlane bpmnElement="activiti-solicitarpermiso" id="BPMNPlane_activiti-
solicitarpermiso">
      <bpmndi:BPMNShape bpmnElement="startevent1" id="BPMNShape_startevent1">
        <omgdc:Bounds height="35.0" width="35.0" x="323.0" y="20.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarNotificacion"
id="BPMNShape_enviarNotificacion">
        <omgdc:Bounds height="55.0" width="105.0" x="288.0" y="80.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="confirmaDepartamento"
id="BPMNShape_confirmaDepartamento">
        <omgdc:Bounds height="55.0" width="105.0" x="288.0" y="160.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="servicetask1" id="BPMNShape_servicetask1">
        <omgdc:Bounds height="55.0" width="141.0" x="270.0" y="311.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="confirmaPersonal" id="BPMNShape_confirmaPersonal">
        <omgdc:Bounds height="55.0" width="105.0" x="288.0" y="384.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="confirmaGerencia" id="BPMNShape_confirmaGerencia">
        <omgdc:Bounds height="55.0" width="105.0" x="288.0" y="530.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="publicaAceptacionPermiso"
id="BPMNShape_publicaAceptacionPermiso">
        <omgdc:Bounds height="55.0" width="105.0" x="288.0" y="668.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarConfirmacionSolicitante"
id="BPMNShape_enviarConfirmacionSolicitante">
        <omgdc:Bounds height="55.0" width="105.0" x="288.0" y="754.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="endevent1" id="BPMNShape_endevent1">
        <omgdc:Bounds height="35.0" width="35.0" x="323.0" y="840.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="exclusivegateway1"
id="BPMNShape_exclusivegateway1">
        <omgdc:Bounds height="40.0" width="40.0" x="320.0" y="247.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="publicarRechazoPermiso"
id="BPMNShape_publicarRechazoPermiso">
        <omgdc:Bounds height="55.0" width="105.0" x="500.0" y="668.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="exclusivegateway2"
id="BPMNShape_exclusivegateway2">
        <omgdc:Bounds height="40.0" width="40.0" x="320.0" y="459.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="exclusivegateway3"
id="BPMNShape_exclusivegateway3">
        <omgdc:Bounds height="40.0" width="40.0" x="320.0" y="610.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarCancelacionSolicitante"
id="BPMNShape_enviarCancelacionSolicitante">
        <omgdc:Bounds height="55.0" width="105.0" x="500.0" y="754.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNEdge bpmnElement="flow1" id="BPMNEdge_flow1">
        <omgdi:waypoint x="340.0" y="55.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="80.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow2" id="BPMNEdge_flow2">
        <omgdi:waypoint x="340.0" y="135.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="160.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
```

```
      <bpmndi:BPMNEdge bpmnElement="flow3" id="BPMNEdge_flow3">
        <omgdi:waypoint x="340.0" y="215.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="247.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow4" id="BPMNEdge_flow4">
        <omgdi:waypoint x="340.0" y="366.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="384.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow5" id="BPMNEdge_flow5">
        <omgdi:waypoint x="340.0" y="439.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="459.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow6" id="BPMNEdge_flow6">
        <omgdi:waypoint x="340.0" y="585.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="610.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow7" id="BPMNEdge_flow7">
        <omgdi:waypoint x="340.0" y="723.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="754.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow8" id="BPMNEdge_flow8">
        <omgdi:waypoint x="393.0" y="781.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="807.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="840.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow9" id="BPMNEdge_flow9">
        <omgdi:waypoint x="340.0" y="287.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="311.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow10" id="BPMNEdge_flow10">
        <omgdi:waypoint x="360.0" y="267.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="267.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="668.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow11" id="BPMNEdge_flow11">
        <omgdi:waypoint x="340.0" y="499.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="530.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow12" id="BPMNEdge_flow12">
        <omgdi:waypoint x="360.0" y="479.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="479.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="668.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow13" id="BPMNEdge_flow13">
        <omgdi:waypoint x="340.0" y="650.0"></omgdi:waypoint>
        <omgdi:waypoint x="340.0" y="668.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow14" id="BPMNEdge_flow14">
        <omgdi:waypoint x="360.0" y="630.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="629.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="668.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow15" id="BPMNEdge_flow15">
        <omgdi:waypoint x="552.0" y="723.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="754.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow16" id="BPMNEdge_flow16">
        <omgdi:waypoint x="552.0" y="809.0"></omgdi:waypoint>
        <omgdi:waypoint x="552.0" y="857.0"></omgdi:waypoint>
        <omgdi:waypoint x="358.0" y="857.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
    </bpmndi:BPMNPlane>
  </bpmndi:BPMNDiagram>
</definitions>
```

13.4.1.4 Fichero Activiti-solicitudgenericaconsubsanacion.bpmn

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:activiti="http://activiti.org/bpmn"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
typeLanguage="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.w3.org/1999/XPath"
targetNamespace="http://www.activiti.org/test">
  <process id="activiti-solicitudgenericaconsubsanacion" name="Proceso de Solicitud
General con Subsanación" isExecutable="true">
    <startEvent id="startevent1" name="Start"></startEvent>
    <userTask id="resolverSolicitud" name="Resolver Solicitud">
      <extensionElements>
        <activiti:taskListener event="create"
class="com.vitruvio.ego.vcasemgr.web.service.impl.workflow_activiti.AsignarAGrupoRespons
able"></activiti:taskListener>
      </extensionElements>
    </userTask>
    <serviceTask id="publicarConfirmacionSolicitud" name="Publicar estimación solicitud"
activiti:delegateExpression="${publicarHitoActivitiTask}">
      <extensionElements>
        <activiti:field name="codigoHito">
          <activiti:string>RESGUARDOSOLICITUD_TRAMITADO</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <endEvent id="endevent1" name="End"></endEvent>
    <exclusiveGateway id="exclusivegateway1" name="Exclusive Gateway"
default="flow15"></exclusiveGateway>
    <serviceTask id="enviarCancelacionSolicitante" name="Enviar desestimación
solicitante" activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:expression>Resolución solicitud Universidad de
Oviedo</activiti:expression>
        </activiti:field>
        <activiti:field name="html">
          <activiti:expression>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
        &lt;head&gt;
                &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt;
                &lt;title&gt;Resoluci&amp;oacute;n solicitud Universidad de
Oviedo&lt;/title&gt;
        &lt;/head&gt;
        &lt;body&gt;
                &lt;p&gt;Estimada/o usuaria/o,&lt;/p&gt;
                &lt;p&gt;Su solicitud ha sido desestimada en el Registro
Telem&amp;aacute;tico de la Universidad de Oviedo.&lt;/p&gt;
                &lt;ul&gt;
                        &lt;li&gt;N&amp;uacute;mero Expediente:${numExp}&lt;/li&gt;
                        &lt;li&gt;Nombre:${nombreSolicitud}&lt;/li&gt;
                        &lt;li&gt;Fecha:${fecha}&lt;/li&gt;
                &lt;/ul&gt;

                &lt;p&gt;El organismo encargado de su gesti&amp;oacute;n ha
sido:&lt;/p&gt;${org}

                &lt;p&gt;Puede consultar su tramitaci&amp;oacute;n en la web de la &lt;a
href="https://intranet.uniovi.es/group/intranetuniovi/servicios/comovalomio"&gt;Intranet
de la Universidad de Oviedo&lt;/a&gt; &lt;/p&gt;

                &lt;p&gt;Atentamente.&lt;/p&gt;

                &lt;p style="font-size:x-small; text-align:right"&gt;
                        &amp;copy; Centro de Innovaci&amp;oacute;n | Universidad de
Oviedo&lt;br/&gt;
                        Tfno: 985 45 80 91 | Fax: 985 45 80 81 |
c1nn@innova.uniovi.es&lt;br/&gt;
```

```
                        Campus de Barredo (3&amp;ordm; planta) | C/ Gonzalo
Guti&amp;eacute;rrez Quir&amp;oacute;s s/n 33600 &lt;br/&gt;
                        Mieres - Asturias&lt;br/&gt;
                &lt;/p&gt;
        &lt;/body&gt;
&lt;/html&gt;</activiti:expression>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailSolicitanteWorkflowActivitiTask}"></activiti:executionL
istener>
      </extensionElements>
    </serviceTask>
    <serviceTask id="mailcreacionSolicitante" name="Enviar aviso creación solicitud al
Solicitante" activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:expression>Presentación solicitud Universidad de
Oviedo</activiti:expression>
        </activiti:field>
        <activiti:field name="html">
          <activiti:expression>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
        &lt;head&gt;
                &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt;
                &lt;title&gt;Presentaci&amp;oacute;n solicitud Universidad de
Oviedo&lt;/title&gt;
        &lt;/head&gt;
        &lt;body&gt;
                &lt;p&gt;${debug}&lt;/p&gt;
                &lt;p&gt;Estimada/o usuaria/o,&lt;/p&gt;
                &lt;p&gt;Su solicitud ha sido recibida en el Registro
Telem&amp;aacute;tico de la Universidad de Oviedo.&lt;/p&gt;
                &lt;ul&gt;
                        &lt;li&gt;N&amp;uacute;mero Expediente:${numExp}&lt;/li&gt;
                        &lt;li&gt;Nombre:${nombreSolicitud}&lt;/li&gt;
                        &lt;li&gt;Fecha:${fecha}&lt;/li&gt;
                &lt;/ul&gt;

                &lt;p&gt;El organismo encargado de su gesti&amp;oacute;n
es:&lt;/p&gt;${org}

                &lt;p&gt;Puede consultar su tramitaci&amp;oacute;n en la web de la &lt;a
href="https://intranet.uniovi.es/group/intranetuniovi/servicios/comovalomio"&gt;Intranet
de la Universidad de Oviedo&lt;/a&gt; &lt;/p&gt;

                &lt;p&gt;Atentamente.&lt;/p&gt;

                &lt;p style="font-size:x-small; text-align:right"&gt;
                        &amp;copy; Centro de Innovaci&amp;oacute;n | Universidad de
Oviedo&lt;br/&gt;
                        Tfno: 985 45 80 91 | Fax: 985 45 80 81 |
c1nn@innova.uniovi.es&lt;br/&gt;
                        Campus de Barredo (3&amp;ordm; planta) | C/ Gonzalo
Guti&amp;eacute;rrez Quir&amp;oacute;s s/n 33600 &lt;br/&gt;
                        Mieres - Asturias&lt;br/&gt;
                &lt;/p&gt;
        &lt;/body&gt;
&lt;/html&gt;</activiti:expression>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailSolicitanteWorkflowActivitiTask}"></activiti:executionL
istener>
      </extensionElements>
    </serviceTask>
    <serviceTask id="enviarAvisoCreacionSolicitante" name="Enviar aviso creación
solicitud al Gestor" activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:expression>Presentación solicitud Universidad de
Oviedo</activiti:expression>
```

```
        </activiti:field>
        <activiti:field name="html">
          <activiti:expression>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
        &lt;head&gt;
                &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt;
                &lt;title&gt;Presentaci&amp;oacute;n solicitud Universidad de
Oviedo&lt;/title&gt;
        &lt;/head&gt;
        &lt;body&gt;
                &lt;p&gt;${debug}&lt;/p&gt;
                &lt;p&gt;Estimada/o usuaria/o,&lt;/p&gt;
                &lt;p&gt;Se ha recibido una nueva solicitud en el Registro
Telem&amp;aacute;tico de la Universidad de Oviedo.&lt;/p&gt;
                &lt;ul&gt;
                        &lt;li&gt;N&amp;uacute;mero Expediente:${numExp}&lt;/li&gt;
                        &lt;li&gt;Nombre${nombreSolicitud}&lt;/li&gt;
                        &lt;li&gt;Fecha:${fecha}&lt;/li&gt;
                        &lt;li&gt;Solicitante:${usuario}&lt;/li&gt;
                &lt;/ul&gt;
                &lt;p&gt;Puede consultarla en la web del &lt;a
href="https://euniovi.uniovi.es/Tramitador/"&gt;Tramitador&lt;/a&gt;&lt;/p&gt;

                &lt;p&gt;Atentamente.&lt;/p&gt;

                &lt;p style="font-size:x-small; text-align:right"&gt;
                        &amp;copy; Centro de Innovaci&amp;oacute;n | Universidad de
Oviedo&lt;br/&gt;
                        Tfno: 985 45 80 91 | Fax: 985 45 80 81 |
c1nn@innova.uniovi.es&lt;br/&gt;
                        Campus de Barredo (3&amp;ordm; planta) | C/ Gonzalo
Guti&amp;eacute;rrez Quir&amp;oacute;s s/n 33600 &lt;br/&gt;
                        Mieres - Asturias&lt;br/&gt;
                &lt;/p&gt;
        &lt;/body&gt;
&lt;/html&gt;</activiti:expression>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailGestorWorkflowActivitiTask}"></activiti:executionListen
er>
      </extensionElements>
    </serviceTask>
    <serviceTask id="generarPdfResolucionDesestimacion" name="Generar PDF Desestimación"
activiti:delegateExpression="${generarPdfResolucionActivitiTask}">
      <extensionElements>
        <activiti:field name="resolucion">
          <activiti:string>DESESTIMADA</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <serviceTask id="generarPdfResolucionEstimacion" name="Generar PDF Estimación"
activiti:delegateExpression="${generarPdfResolucionActivitiTask}">
      <extensionElements>
        <activiti:field name="resolucion">
          <activiti:string>ESTIMADA</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <serviceTask id="enviarAceptacionSolicitante" name="Enviar estimación solicitante"
activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:expression>Resolución solicitud Universidad de
Oviedo</activiti:expression>
        </activiti:field>
        <activiti:field name="html">
          <activiti:expression>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
        &lt;head&gt;
                &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt;
                &lt;title&gt;Resoluci&amp;oacute;n solicitud Universidad de
Oviedo&lt;/title&gt;
```

```
        &lt;/head&gt;
        &lt;body&gt;
                &lt;p&gt;Estimada/o usuaria/o,&lt;/p&gt;
                &lt;p&gt;Su solicitud ha sido estimada en el Registro
Telem&amp;aacute;tico de la Universidad de Oviedo.&lt;/p&gt;
                &lt;ul&gt;
                        &lt;li&gt;N&amp;uacute;mero Expediente:${numExp}&lt;/li&gt;
                        &lt;li&gt;Nombre:${nombreSolicitud}&lt;/li&gt;
                        &lt;li&gt;Fecha:${fecha}&lt;/li&gt;
                &lt;/ul&gt;

                &lt;p&gt;El organismo encargado de su gesti&amp;oacute;n ha
sido:&lt;/p&gt;${org}

                &lt;p&gt;Puede consultar su tramitaci&amp;oacute;n en la web de la &lt;a
href="https://intranet.uniovi.es/group/intranetuniovi/servicios/comovalomio"&gt;Intranet
de la Universidad de Oviedo&lt;/a&gt; &lt;/p&gt;

                &lt;p&gt;Atentamente.&lt;/p&gt;

                &lt;p style="font-size:x-small; text-align:right"&gt;
                        &amp;copy; Centro de Innovaci&amp;oacute;n | Universidad de
Oviedo&lt;br/&gt;
                        Tfno: 985 45 80 91 | Fax: 985 45 80 81 |
c1nn@innova.uniovi.es&lt;br/&gt;
                        Campus de Barredo (3&amp;ordm; planta) | C/ Gonzalo
Guti&amp;eacute;rrez Quir&amp;oacute;s s/n 33600 &lt;br/&gt;
                        Mieres - Asturias&lt;br/&gt;
                &lt;/p&gt;
        &lt;/body&gt;
&lt;/html&gt;</activiti:expression>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailSolicitanteWorkflowActivitiTask}"></activiti:executionL
istener>
      </extensionElements>
    </serviceTask>
    <serviceTask id="publicarInicioSubsanacion" name="Publicar inicio subsanación"
activiti:delegateExpression="${publicarHitoActivitiTask}">
      <extensionElements>
        <activiti:field name="codigoHito">
          <activiti:string>SUBSANACION_AVISO</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <serviceTask id="enviarInicioSubsanacionSolicitante" name="Enviar inicio subsanación
al solicitante" activiti:type="mail">
      <extensionElements>
        <activiti:field name="to">
          <activiti:expression>${to}</activiti:expression>
        </activiti:field>
        <activiti:field name="subject">
          <activiti:expression>Inicio Subsanación solicitud Universidad de
Oviedo</activiti:expression>
        </activiti:field>
        <activiti:field name="html">
          <activiti:expression>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
        &lt;head&gt;
                &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt;
                &lt;title&gt;Inicio Subsanaci&amp;oacute;n solicitud Universidad de
Oviedo&lt;/title&gt;
        &lt;/head&gt;
        &lt;body&gt;
                &lt;p&gt;Estimada/o usuaria/o,&lt;/p&gt;
                &lt;p&gt;Debe realizar la siguiente subsanaci&amp;oacute;n en relación
con la siguiente solicitud:&lt;/p&gt;
                &lt;ul&gt;
                        &lt;li&gt;N&amp;uacute;mero Expediente:${numExp}&lt;/li&gt;
                        &lt;li&gt;Nombre:${nombreSolicitud}&lt;/li&gt;
                        &lt;li&gt;Fecha:${fecha}&lt;/li&gt;
                &lt;/ul&gt;

                ${contenidoemail}
```

```
                &lt;p&gt;El organismo encargado de su gesti&amp;oacute;n
es:&lt;/p&gt;${org}

                &lt;p&gt;Puede consultar su tramitaci&amp;oacute;n en la web de la &lt;a
href="https://intranet.uniovi.es/group/intranetuniovi/servicios/comovalomio"&gt;Intranet
de la Universidad de Oviedo&lt;/a&gt; &lt;/p&gt;

                &lt;p&gt;Atentamente.&lt;/p&gt;

                &lt;p style="font-size:x-small; text-align:right"&gt;
                        &amp;copy; Centro de Innovaci&amp;oacute;n | Universidad de
Oviedo&lt;br/&gt;
                        Tfno: 985 45 80 91 | Fax: 985 45 80 81 |
clnn@innova.uniovi.es&lt;br/&gt;
                        Campus de Barredo (3&amp;ordm; planta) | C/ Gonzalo
Guti&amp;eacute;rrez Quir&amp;oacute;s s/n 33600 &lt;br/&gt;
                        Mieres - Asturias&lt;br/&gt;
                &lt;/p&gt;
        &lt;/body&gt;
&lt;/html&gt;</activiti:expression>
        </activiti:field>
        <activiti:executionListener event="start"
delegateExpression="${enviarEmailSolicitanteWorkflowActivitiTask}"></activiti:executionL
istener>
      </extensionElements>
    </serviceTask>
    <serviceTask id="publicarRechazoSolicitud" name="Publicar desestimación solicitud"
activiti:delegateExpression="${publicarHitoActivitiTask}">
      <extensionElements>
        <activiti:field name="codigoHito">
          <activiti:string>CANCELACION_SOLICITUD</activiti:string>
        </activiti:field>
      </extensionElements>
    </serviceTask>
    <sequenceFlow id="flow3" sourceRef="resolverSolicitud"
targetRef="exclusivegateway1"></sequenceFlow>
    <sequenceFlow id="flow6" sourceRef="exclusivegateway1"
targetRef="generarPdfResolucionDesestimacion">
      <conditionExpression
xsi:type="tFormalExpression"><![CDATA[${cancelar}]]></conditionExpression>
    </sequenceFlow>
    <sequenceFlow id="flow8" sourceRef="enviarCancelacionSolicitante"
targetRef="endevent1"></sequenceFlow>
    <sequenceFlow id="flow10" sourceRef="startevent1"
targetRef="mailcreacionSolicitante"></sequenceFlow>
    <sequenceFlow id="flow11" sourceRef="mailcreacionSolicitante"
targetRef="enviarAvisoCreacionSolicitante"></sequenceFlow>
    <sequenceFlow id="flow12" sourceRef="enviarAvisoCreacionSolicitante"
targetRef="resolverSolicitud"></sequenceFlow>
    <sequenceFlow id="flow14" sourceRef="generarPdfResolucionEstimacion"
targetRef="publicarConfirmacionSolicitud"></sequenceFlow>
    <sequenceFlow id="flow15" sourceRef="exclusivegateway1"
targetRef="generarPdfResolucionEstimacion"></sequenceFlow>
    <sequenceFlow id="flow16" sourceRef="publicarConfirmacionSolicitud"
targetRef="enviarAceptacionSolicitante"></sequenceFlow>
    <sequenceFlow id="flow17" sourceRef="enviarAceptacionSolicitante"
targetRef="endevent1"></sequenceFlow>
    <sequenceFlow id="flow18" sourceRef="publicarInicioSubsanacion"
targetRef="enviarInicioSubsanacionSolicitante"></sequenceFlow>
    <sequenceFlow id="flow19" sourceRef="exclusivegateway1"
targetRef="publicarInicioSubsanacion">
      <conditionExpression
xsi:type="tFormalExpression"><![CDATA[${subsanar}]]></conditionExpression>
    </sequenceFlow>
    <sequenceFlow id="flow23" sourceRef="enviarInicioSubsanacionSolicitante"
targetRef="resolverSolicitud"></sequenceFlow>
    <sequenceFlow id="flow24" sourceRef="generarPdfResolucionDesestimacion"
targetRef="publicarRechazoSolicitud"></sequenceFlow>
    <sequenceFlow id="flow25" sourceRef="publicarRechazoSolicitud"
targetRef="enviarCancelacionSolicitante"></sequenceFlow>
  </process>
  <bpmndi:BPMNDiagram id="BPMNDiagram_activiti-solicitudgenericaconsubsanacion">
    <bpmndi:BPMNPlane bpmnElement="activiti-solicitudgenericaconsubsanacion"
id="BPMNPlane_activiti-solicitudgenericaconsubsanacion">
      <bpmndi:BPMNShape bpmnElement="startevent1" id="BPMNShape_startevent1">
        <omgdc:Bounds height="35.0" width="35.0" x="277.0" y="20.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
```

```
      <bpmndi:BPMNShape bpmnElement="resolverSolicitud"
id="BPMNShape_resolverSolicitud">
        <omgdc:Bounds height="55.0" width="105.0" x="242.0" y="280.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="publicarConfirmacionSolicitud"
id="BPMNShape_publicarConfirmacionSolicitud">
        <omgdc:Bounds height="55.0" width="112.0" x="239.0" y="520.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="endevent1" id="BPMNShape_endevent1">
        <omgdc:Bounds height="35.0" width="35.0" x="188.0" y="712.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="exclusivegateway1"
id="BPMNShape_exclusivegateway1">
        <omgdc:Bounds height="40.0" width="40.0" x="274.0" y="362.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarCancelacionSolicitante"
id="BPMNShape_enviarCancelacionSolicitante">
        <omgdc:Bounds height="55.0" width="114.0" x="56.0" y="605.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="mailcreacionSolicitante"
id="BPMNShape_mailcreacionSolicitante">
        <omgdc:Bounds height="55.0" width="119.0" x="235.0" y="90.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarAvisoCreacionSolicitante"
id="BPMNShape_enviarAvisoCreacionSolicitante">
        <omgdc:Bounds height="55.0" width="119.0" x="235.0" y="190.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="generarPdfResolucionDesestimacion"
id="BPMNShape_generarPdfResolucionDesestimacion">
        <omgdc:Bounds height="55.0" width="105.0" x="60.0" y="425.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="generarPdfResolucionEstimacion"
id="BPMNShape_generarPdfResolucionEstimacion">
        <omgdc:Bounds height="55.0" width="105.0" x="242.0" y="425.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarAceptacionSolicitante"
id="BPMNShape_enviarAceptacionSolicitante">
        <omgdc:Bounds height="55.0" width="105.0" x="242.0" y="605.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="publicarInicioSubsanacion"
id="BPMNShape_publicarInicioSubsanacion">
        <omgdc:Bounds height="55.0" width="111.0" x="440.0" y="425.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="enviarInicioSubsanacionSolicitante"
id="BPMNShape_enviarInicioSubsanacionSolicitante">
        <omgdc:Bounds height="65.0" width="111.0" x="440.0" y="515.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="publicarRechazoSolicitud"
id="BPMNShape_publicarRechazoSolicitud">
        <omgdc:Bounds height="55.0" width="118.0" x="54.0" y="520.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNEdge bpmnElement="flow3" id="BPMNEdge_flow3">
        <omgdi:waypoint x="294.0" y="335.0"></omgdi:waypoint>
        <omgdi:waypoint x="294.0" y="362.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow6" id="BPMNEdge_flow6">
        <omgdi:waypoint x="274.0" y="382.0"></omgdi:waypoint>
        <omgdi:waypoint x="112.0" y="382.0"></omgdi:waypoint>
        <omgdi:waypoint x="112.0" y="425.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow8" id="BPMNEdge_flow8">
        <omgdi:waypoint x="113.0" y="660.0"></omgdi:waypoint>
        <omgdi:waypoint x="205.0" y="712.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow10" id="BPMNEdge_flow10">
        <omgdi:waypoint x="294.0" y="55.0"></omgdi:waypoint>
        <omgdi:waypoint x="294.0" y="90.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow11" id="BPMNEdge_flow11">
        <omgdi:waypoint x="294.0" y="145.0"></omgdi:waypoint>
        <omgdi:waypoint x="294.0" y="190.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow12" id="BPMNEdge_flow12">
        <omgdi:waypoint x="294.0" y="245.0"></omgdi:waypoint>
        <omgdi:waypoint x="294.0" y="280.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow14" id="BPMNEdge_flow14">
```

```
        <omgdi:waypoint x="294.0" y="480.0"></omgdi:waypoint>
        <omgdi:waypoint x="295.0" y="520.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow15" id="BPMNEdge_flow15">
        <omgdi:waypoint x="294.0" y="402.0"></omgdi:waypoint>
        <omgdi:waypoint x="294.0" y="425.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow16" id="BPMNEdge_flow16">
        <omgdi:waypoint x="295.0" y="575.0"></omgdi:waypoint>
        <omgdi:waypoint x="294.0" y="605.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow17" id="BPMNEdge_flow17">
        <omgdi:waypoint x="294.0" y="660.0"></omgdi:waypoint>
        <omgdi:waypoint x="205.0" y="712.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow18" id="BPMNEdge_flow18">
        <omgdi:waypoint x="495.0" y="480.0"></omgdi:waypoint>
        <omgdi:waypoint x="495.0" y="515.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow19" id="BPMNEdge_flow19">
        <omgdi:waypoint x="314.0" y="382.0"></omgdi:waypoint>
        <omgdi:waypoint x="403.0" y="381.0"></omgdi:waypoint>
        <omgdi:waypoint x="495.0" y="381.0"></omgdi:waypoint>
        <omgdi:waypoint x="495.0" y="425.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow23" id="BPMNEdge_flow23">
        <omgdi:waypoint x="551.0" y="547.0"></omgdi:waypoint>
        <omgdi:waypoint x="597.0" y="549.0"></omgdi:waypoint>
        <omgdi:waypoint x="597.0" y="414.0"></omgdi:waypoint>
        <omgdi:waypoint x="597.0" y="309.0"></omgdi:waypoint>
        <omgdi:waypoint x="372.0" y="309.0"></omgdi:waypoint>
        <omgdi:waypoint x="347.0" y="307.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow24" id="BPMNEdge_flow24">
        <omgdi:waypoint x="112.0" y="480.0"></omgdi:waypoint>
        <omgdi:waypoint x="113.0" y="520.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
      <bpmndi:BPMNEdge bpmnElement="flow25" id="BPMNEdge_flow25">
        <omgdi:waypoint x="113.0" y="575.0"></omgdi:waypoint>
        <omgdi:waypoint x="113.0" y="605.0"></omgdi:waypoint>
      </bpmndi:BPMNEdge>
    </bpmndi:BPMNPlane>
  </bpmndi:BPMNDiagram>
</definitions>
```

13.4.2 Subsistemas para la integración de Activiti con el Gestor de Trámites

13.4.2.1 Fichero Task.java

```
package com.vitruvio.workflow.domain;

import java.util.Date;
import java.util.Map;

/**
 * Modelo de Tarea que se utiliza desde el gestor de trámites
 * @author nataliagm
 *
 */
public class Task {

	private String id;
	private Date start;
	private Date finish;
	private String status;
	private String name;
	private String description;
	private String type;
	private String owner;
	private String creator;
	private String priority;
	private String workflowId;
	private String workflowStepId;
	private Map<String, Object> flexFields;

	public Task(){
		super();
	}

	public String getId() {
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public Date getStart() {
		return start;
	}

	public void setStart(Date start) {
		this.start = start;
	}

	public Date getFinish() {
		return finish;
	}

	public void setFinish(Date finish) {
		this.finish = finish;
	}

	public String getStatus() {
		return status;
	}

	public void setStatus(String status) {
		this.status = status;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
```

```
                this.name = name;
        }

        public String getDescription() {
                return description;
        }

        public void setDescription(String description) {
                this.description = description;
        }

        public String getType() {
                return type;
        }

        public void setType(String type) {
                this.type = type;
        }

        public String getOwner() {
                return owner;
        }

        public void setOwner(String owner) {
                this.owner = owner;
        }

        public String getCreator() {
                return creator;
        }

        public void setCreator(String creator) {
                this.creator = creator;
        }

        public String getPriority() {
                return priority;
        }

        public void setPriority(String priority) {
                this.priority = priority;
        }

        public String getWorkflowId() {
                return workflowId;
        }

        public void setWorkflowId(String workflowId) {
                this.workflowId = workflowId;
        }

        public String getWorkflowStepId() {
                return workflowStepId;
        }

        public void setWorkflowStepId(String workflowStepId) {
                this.workflowStepId = workflowStepId;
        }

        public Map<String, Object> getFlexFields() {
                return flexFields;
        }

        public void setFlexFields(Map<String, Object> flexFields) {
                this.flexFields = flexFields;
        }

}
```

13.4.2.2 Fichero TaskQueryCriteria.java

```
package com.vitruvio.workflow.domain;

/**
 * Modelo TaskQueryCriteria que se utiliza desde el gestor de trámites, para
 * crear consultar en las Tareas
 *
 * @author nataliagm
 *
 */
public class TaskQueryCriteria {

	private Task task;
	private int maxResults;

	public TaskQueryCriteria() {
		this.task = new Task();
	}

	public Task getTask() {
		return this.task;
	}

	public void setTask(Task task) {
		this.task = task;
	}

	public int getMaxResults() {
		return this.maxResults;
	}

	public void setMaxResults(int maxResults) {
		this.maxResults = maxResults;
	}

}
```

13.4.2.3 Fichero Workflow.java

```
package com.vitruvio.workflow.domain;

import java.util.Date;

/**
 * Modelo Workflow con el que trabaja el gestora de Trámites
 *
 * @author nataliagm
 *
 */
public class Workflow {

	private String id;
	private String name;
	private Date start;
	private Date finish;
	private String status;

	public String getId() {
		return this.id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getName() {
		return this.name;
	}

	public void setName(String name) {
		this.name = name;
	}
```

```
	public Date getStart() {
		return this.start;
	}

	public void setStart(Date start) {
		this.start = start;
	}

	public Date getFinish() {
		return this.finish;
	}

	public void setFinish(Date finish) {
		this.finish = finish;
	}

	public String getStatus() {
		return this.status;
	}

	public void setStatus(String status) {
		this.status = status;
	}

}
```

13.4.2.4 Fichero WorkflowService.java

```
package com.vitruvio.workflow;

import java.util.List;
import java.util.Map;

import com.vitruvio.workflow.domain.Task;
import com.vitruvio.workflow.domain.TaskQueryCriteria;
import com.vitruvio.workflow.domain.Workflow;
import com.vitruvio.workflow.exception.InvalidWorkflowException;

/**
 * Interfaz que define las operaciones que el gestor de becas puede realizar con
 * los workflows de Activiti
 *
 * @author nataliagm
 *
 */
public interface WorkflowService {

	/**
	 * Crea una nueva instancia del workflow que se indica y añade los
	 * argumentos recibidos
	 *
	 * @param name
	 *            nombre del workflow que se quiere instanciar
	 * @param paramMap
	 *            parámetros que se van a incluir en la creación de la instancia
	 *            del workflow
	 * @return Workflow creado
	 * @throws InvalidWorkflowException
	 */
	public abstract Workflow startNewWorkflow(String name,
			Map<String, Object> paramMap) throws InvalidWorkflowException;

	/**
	 * Devuelve la instancia del workflow al que pertenece el identificador
	 *
	 * @param processInstanceId
	 *            identificador del workflow que se solicita
	 * @return Workflow solicitado
	 */
	public abstract Workflow getWorkflow(String processInstanceId);

	/**
	 * Finaliza la instancia del workflow que se le indica
```

```
     *
     * @param processInstanceId
     *            identificador de la instacia de workflow que se quiere
     *            cancelar
     * @throws InvalidWorkflowException
     */
    public abstract void cancelWorkflow(String processInstanceId)
                    throws InvalidWorkflowException;

    /**
     * Devuelve una lista con todas las tareas del workflow que se le indica
     *
     * @param processInstanceId
     *            identificador de la instacia de workflo
     * @return lista de Tareas
     * @throws InvalidWorkflowException
     */
    public abstract List<Task> getWorkflowTasks(String processInstanceId)
                    throws InvalidWorkflowException;

    /**
     * Crea una tarea de Activiti a partir de la tarea del modelo recibida
     *
     * @param task
     *            Tarea que se desea crear
     * @return Tarea creada
     */
    public abstract Task createTask(Task task);

    /**
     * Finaliza la tarea que se indica, añadiendo los parámetros que recibe
     *
     * @param taskId
     *            identificador de la tarea
     * @param param
     *            Datos que se desean añadir a la tarea
     */
    public abstract void closeTask(String taskId, Map<String, Object> param);

    /**
     * Asigna la tarea recibida al usuario indicado
     *
     * @param taskId
     *            identificador de la tarea
     * @param asignee
     *            usuario al que se va a asignar la tarea
     */
    public abstract void delegateTask(String taskId, String asignee);

    /**
     * Modifica la tarea de Activiti con los datos recibidos en la tarea del
     * modelo
     *
     * @param task
     *            Tarea para modificar
     */
    public abstract void updateTask(Task task);

    /**
     * Devuelve las tareas del workflow y del creador que se indican en el
     * criterio de búsqueda
     *
     * @param criteria
     *            Criterio de búsqueda de la tareas
     * @return lista con las tareas que cumplen con el criterio de búsqueda
     */
    public abstract List<Task> getTasks(TaskQueryCriteria criteria);

    /**
     * Devuelve la tarea que se está ejecutando en ese momento en la instancia
     * del workflow que se indica
     *
     * @param processInstanceId
     *            identificador de la instancia del workflow
     * @return Tarea que se encuentra en ejecución
     */
    public abstract Task getWorkflowRunTask(String processInstanceId);
```

```
        /**
         * Añade la variable que se indica en la instancia del workflow indicado
         *
         * @param idProccess
         *            identificador de la instacia del workflow
         * @param variable
         *            Nombre de la variable que se desea añadir
         * @param value
         *            Valor de la variable que se desea añadir
         */
        public abstract void setVariable(String idProccess, String variable,
                        Object value);

        /**
         * Devuelve la variable del workflow que se solicita
         *
         * @param idProccess
         *            identificador de la instancia del workflow
         * @param variable
         *            Nombre de la variable que se quiere obtener
         * @return Variable que se está solicitando
         */
        public abstract Object getVariable(String idProccess, String variable);

}
```

13.4.2.5 Fichero InvalidWorkflowException.java

```
package com.vitruvio.workflow.exception;

/**
 * Excepción propia que se genera cuando se produce algún error en la
 * interacción con los workflows
 *
 * @author nataliagm
 *
 */
public class InvalidWorkflowException extends RuntimeException {

        private static final long serialVersionUID = 1L;

}
```

13.4.2.6 Fichero ActivitiWorkflowService.java

```
package com.vitruvio.workflow.activiti;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import org.activiti.engine.RuntimeService;
import org.activiti.engine.TaskService;
import org.activiti.engine.runtime.ProcessInstance;
import org.activiti.engine.runtime.ProcessInstanceQuery;
import org.activiti.engine.task.TaskQuery;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.vitruvio.workflow.WorkflowService;
import com.vitruvio.workflow.domain.Task;
import com.vitruvio.workflow.domain.TaskQueryCriteria;
import com.vitruvio.workflow.domain.Workflow;

/**
 * Implementación del interfez WorkflowService, donde se realiza la
 * implementación con Activiti
 *
 * @author nataliagm
 *
 */
```

```
@Service(value = "activitiWorkflowService")
public class ActivitiWorkflowService implements WorkflowService {

	@Autowired
	private RuntimeService runtimeService;

	@Autowired
	private TaskService taskService;

	private ProcessInstance process;

	@Override
	public void cancelWorkflow(String processInstanceId) {
		runtimeService.deleteProcessInstance(processInstanceId, null);
	}

	@Override
	public Workflow startNewWorkflow(String name, Map<String, Object> paramMap) {
		process = runtimeService.startProcessInstanceByKey(name, paramMap);
		return buildWorkflowFromProcessInstance(process);
	}

	@Override
	public void closeTask(String taskId, Map<String, Object> param) {
		taskService.complete(taskId, param);
	}

	@Override
	public void delegateTask(String taskId, String asignee) {
		taskService.delegateTask(taskId, asignee);
	}

	@Override
	public Task createTask(Task task) {
		org.activiti.engine.task.Task jtask = taskService.newTask();

		jtask.setAssignee(task.getCreator());
		jtask.setName(task.getName());
		jtask.setDescription(task.getDescription());
		jtask.setDueDate(task.getFinish());
		jtask.setOwner(task.getOwner());
		taskService.saveTask(jtask);
		task.setId(jtask.getId());

		return task;
	}

	@Override
	public void updateTask(Task task) {

		org.activiti.engine.task.Task jtask = taskService.createTaskQuery()
				.taskId(task.getId()).singleResult();
		jtask.setAssignee(task.getOwner());
		jtask.setDescription(task.getDescription());
		jtask.setDueDate(task.getFinish());
		jtask.setName(task.getName());

		taskService.saveTask(jtask);
	}

	@Override
	public List<Task> getTasks(TaskQueryCriteria criteria) {
		String processId = criteria.getTask().getWorkflowId();

		List<Task> result = new ArrayList<Task>();

		TaskQuery query = taskService.createTaskQuery();
		query.processInstanceId(processId);

		if (criteria.getTask().getOwner() != null) {
			query.taskAssignee(criteria.getTask().getOwner());
		}

		List<org.activiti.engine.task.Task> jtasks = query.list();

		for (org.activiti.engine.task.Task jtask : jtasks) {
			result.add(buildTask(jtask));
```

```
		}

		return result;
	}

	@Override
	public List<Task> getWorkflowTasks(String processInstanceId) {
		TaskQuery tq = taskService.createTaskQuery().processInstanceId(
				processInstanceId);
		List<Task> tasks = new ArrayList<Task>();
		for (org.activiti.engine.task.Task activity : tq.list()) {
			tasks.addAll(buildTasksFromActivity(activity.getName()));
		}
		return tasks;
	}

	@Override
	public Workflow getWorkflow(String processInstanceId) {
		ProcessInstanceQuery a = runtimeService.createProcessInstanceQuery()
				.processInstanceId(processInstanceId);
		ProcessInstance pp = a.singleResult();
		if (pp != null) {
			return buildWorkflowFromProcessInstance(pp);
		} else {
			return null;
		}
	}

	@Override
	public Task getWorkflowRunTask(String processInstanceId) {
		org.activiti.engine.task.Task t = taskService.createTaskQuery()
				.executionId(processInstanceId).singleResult();
		return buildTask(t);
	}

	@Override
	public void setVariable(String idProccess, String variable, Object value) {
		runtimeService.setVariable(idProccess, variable, value);
	}

	@Override
	public Object getVariable(String idProccess, String variable) {
		return runtimeService.getVariable(idProccess, variable);
	}

	private Workflow buildWorkflowFromProcessInstance(ProcessInstance p) {
		Workflow w = new Workflow();
		w.setId(p.getId());
		w.setName(p.getProcessDefinitionId());
		if (p.isEnded()) {
			w.setStatus("CLOSED");
		} else if (p.isSuspended()) {
			w.setStatus("SUSPENDED");
		} else {
			w.setStatus("OPEN");
		}
		return w;
	}

	private List<Task> buildTasksFromActivity(String activityName) {
		List<Task> tasks = new ArrayList<Task>();
		TaskQuery query = taskService.createTaskQuery();
		query.taskName(activityName);
		List<org.activiti.engine.task.Task> jtasks = query.list();

		for (org.activiti.engine.task.Task jtask : jtasks) {
			Task t = buildTask(jtask);
			tasks.add(t);
		}
		return tasks;
	}

	private Task buildTask(org.activiti.engine.task.Task jtask) {
		Task t = new Task();
		t.setDescription(jtask.getDescription());
		t.setFinish(jtask.getDueDate());
		t.setId(jtask.getId());
```

```
                t.setName(jtask.getName());
                t.setOwner(jtask.getAssignee());
                t.setPriority("" + jtask.getPriority());
                t.setCreator(jtask.getAssignee());
                t.setStart(jtask.getCreateTime());
                t.setStatus("RUNNING");
                t.setWorkflowId(jtask.getExecutionId());
                t.setType(jtask.getTaskDefinitionKey());

                return t;
        }
}
```