Universidad Central "Marta Abreu" de Las Villas.

Facultad Matemática, Física y Computación

Licenciatura en Ciencia de la Computación

TRABAJO DE DIPLOMA

"Integración de un Sistema de Gestión de Reglas de Negocio al flujo de trabajo de una empresa".

Autora: Sandra Estevez Abrahantes

Tutor: MSc. Yoan Pacheco Cárdenas

Curso Académico: 2012-2013

"Año 55 de la Revolución"

Hago constar que el presente trabajo fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____

Firma del tutor

_____

Firma del jefe del Laboratorio

PENSAMIENTO

El futuro tiene muchos nombres:

Para los débiles, lo inalcanzable.

Para los temerosos, lo desconocido.

Para los valientes es la oportunidad.

Víctor Hugo

DEDICATORIA

A mi mamá por ser mi fuente de inspiración, mi apoyo y mi guía. Por estar en los buenos y malos momentos, por ser madre y padre y fundamentalmente por ser la luz que me ilumina al final del camino.

A mi abuela Panchita por ser mi segunda madre; por sus consejos, ayuda y sobre todo su comprensión.

A mis abuelos Julia y Israel por su amor.

A mi abuelo Güije porque siempre está con nosotros.

A mi papá que a pesar de la distancia siempre me ha brindado su amor infinito y su apoyo incondicional. A mis hermanas Claudia y Yadelsy.

A todos mis tíos y primos.

AGRADECIMIENTOS

A mi mamá por confiar en mí, apoyarme en todo momento y darme las fuerzas necesarias para seguir hacia delante.

A mi abuela Panchita por ser mi paño de lágrimas y de alegrías.

A mi tutor Yoan Pacheco por ayudarme en la realización de esta tesis. Por el apoyo brindado y tener confianza en mí. A Ernesto por su colaboración.

A Elizabeth, Lisbet, Maricel y Duniesky por todos sus consejos y ayuda a lo largo de la carrera.

A Marilín y Eneido por haberme tenido como un miembro más de su familia.

A los Licenciados en Psicologia Danay, Lizmary y Yoan por hacer de mí una persona mejor, por ayudarme a vencer mis miedos y por convertirme en una persona más segura. A ellos mi eterno agradecimiento.

A todos mis compañeros de la carrera.

A toda mi familia por creer en mí y por su ayuda excepcional.

RESUMEN

El enfoque de reglas de negocio propone aislar las mismas en una aplicación especializada de manera que estas puedan ser gestionadas independientemente del sistema que las usa; posibilitando así soluciones adaptables al cambio y más reusables. En el siguiente trabajo se efectúa una caracterización de manera general de los componentes básicos de un Sistema de Gestión de Reglas de Negocio, sus ventajas y desventajas. Investigamos tipos de integración mencionados en la literatura y tecnologías para la Integración de Información Empresarial (EII) y la Integración Aplicaciones Empresariales (EAI). Se abordan aspectos relacionados con el flujo de trabajo de una empresa y la integración de un BRMS a la misma. Analizamos criterios necesarios para la selección de la herramienta a integrar al framework de desarrollo de aplicaciones de una organización, se realiza una valoración de la misma en cuanto a arquitectura, lenguaje de definición para las reglas de negocios, interfaces para interactuar con otras aplicaciones y la construcción de servicios web con la misma. Finalmente es descrito el diseño e implementación de una solución informática que muestra cómo integrar un BRMS al flujo de trabajo "control de historias clínicas para trasplante renal" en el Hospital Universitario "Arnaldo Milián Castro". Se muestra la arquitectura general de un proyecto web que utiliza el motor de reglas Drools y las herramientas que se utilizan para el desarrollo del sistema.

ABSTRACT

The business rules approach attempts to isolate the business rules in a specialized application so that they can be managed independently from the system that uses them. This would allow the finding of solutions that can be adapted to change and used again. In this paper we characterize in a general way the basic components of a business rules management system, including its advantages and limitations. We also present our research on the types of integration mentioned in the materials for the Enterprise Information Integration (EII) and the Enterprise Application Integration (EAI). We refer to aspects related to the work flow of an enterprise and the integration of a BRMS to id. In addition, we analyse the criteria required for choosing the tool that is going to be integrated to the application development framework of an organization. We make an assessment of this tool taking into account its architecture, the language used to define the business rules, the interfaces designed to interact with other applications and building web services with it. We also show the general architecture of a web project that uses the rules engine of the BRMS Drools and the tools used for the development of the system. Finally, it is described the design and implementation of a solution that shows how to integrate a workflow BRMS "control kidney transplant medical records" at the University Hospital "Arnaldo Milián Castro". It shows the general architecture of a web project using Drools rule engine and tools used to develop the system.

ÍNDICE

TABLA DE CONTENIDOS

TABLA DE FIGURAS

TABLAS

INTRODUCCIÓN

INTRODUCCIÓN

En las empresas de estos tiempos es frecuente que cuando sus políticas o reglas de negocio cambian los usuarios de los sistemas informáticos que gestionan los procesos principales descubren que lo que en un momento para ellos fue una herramienta actual, que los ayudaba a desempeñar su trabajo con rapidez y calidad, quedó obsoleto y hoy es un producto que no resuelve las necesidades para las cuales fue concebido inicialmente.

Este problema provoca que se compren o reprogramen constantemente funcionalidades o módulos enteros para responder a las nuevas y siempre cambiantes necesidades empresariales, aumentando los costes de mantenimiento en gran medida.

Los sistemas de gestión flujo de trabajo y de reglas de negocio (del inglés BRMS) son cruciales para resolver dicho problema pues todo proceso de negocio no trivial necesita tomar decisiones complejas. Las herramientas de flujo de trabajo automatizan los procesos de negocio, involucrando tanto actividades manuales como automáticas. Los sistemas de flujo de trabajo son el primer ejemplo de un cambio claro en la orientación de la construcción de sistemas informáticos, pasando de los datos a los procesos provocando la necesidad de rediseñar los procesos de negocio para optimizar el funcionamiento de la organización(Fernández, 2006).

Según (Gualteri, 2011) las plataformas para la gestión de reglas brindan los siguientes beneficios:

- Automatizan las decisiones para la acción rápida de los negocios.
- Permiten la expresión directa de las políticas y decisiones.
- Son la mejor opción cuando la lógica es compleja.

- Habilitan los cambios en el software.
- Potencian a los expertos de negocios para la gestión del cambio.

Un motor de flujo de trabajo es una herramienta capaz de proporcionar un ambiente en línea para inicializar, ejecutar, secuenciar y controlar instancias de una definición de proceso, administrar las tareas que ejecuta cada participante y llamar aplicaciones externas de ser necesario. En principio, tiene las funciones básicas para supervisar la ejecución de cada proceso y permite llevar un control sobre ellos (2006). Programas como los que se presentan a continuación permiten modelar, ejecutar o analizar un flujo de trabajo:

- **ProcessMaker:** es una solución de software de flujos de trabajo, de código abierto conocido como Gestor de procesos empresariales (BPM), ayuda a las organizaciones a diseñar, automatizar e implementar procesos de negocio. La caja de herramientas ProcessMaker permite a los usuarios de negocio crear formas y mapas de flujos de trabajo completamente funcionales. Permite generar, reportar, ejecutar y optimizar flujos de trabajo[1].
- **Microsoft SharePoint Designer 2010:** es un programa de diseño de aplicaciones y páginas web que se usa para diseñar, generar y personalizar sitios web que se ejecutan con Microsoft SharePoint Foundation 2010 y Microsoft SharePoint Server 2010. Con SharePoint Designer 2010, puede crear páginas web de gran cantidad de datos, crear soluciones eficaces habilitadas para flujos de trabajo y diseñar el aspecto de su sitio[2].
- **Microsoft Office Visio Premium 2010:** es la versión más completa de Visio de Microsoft. Destinado a entornos profesionales, Microsoft Office Visio Premium 2010 permite crear sus propios diagramas de flujo de forma visual y con un resultado profesional. Incluye plantillas y herramientas de creación avanzadas para

[1]ProcessMaker Workflow Simplified. http://www.processmaker.com/es
[2] Introducción a SharePoint Designer 2010. http://office.microsoft.com/es-mx/sharepoint-designer-help/introduccion-a-sharepoint-designer-2010-HA010370548.aspx

simplificar el proceso de elaboración de diagramas. Entre ellas, destaca la posibilidad de crear gráficos a partir de tablas en Excel[3].

- **Visual Paradigm:** es un software de modelado UML que nos permite analizar, diseñar, codificar, probar y desplegar. Dibuja todo tipo de diagramas UML, genera código fuente a partir de los diagramas y elaboración de documentos. Visual Paradigm es una herramienta ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que estén interesados en la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos[4].

- **ArchestrA Workflow:** es una aplicación de flujos de trabajo avanzada que permite la automatización de todos los procesos de una empresa sincronizando la actividad humana con los sistemas operativos. Esta sofisticada aplicación de gestión de procesos de negocio (Business Process Management - BPM) permite diseñar, ejecutar, evaluar y mejorar los procesos internos y externos de cualquier organización con el objetivo de conseguir niveles más altos de colaboración, productividad e innovación[5].

- **Soft Expert(SE) Workflow:** provee un editor gráfico de workflows que posibilita el modelado de las instancias de proceso en tiempo de ejecución (*"in-flight" modeling*). SE Workflow también incluye SE BI (Business Intelligence), una herramienta poderosa para la elaboración de análisis por líneas de negocio y detección de ineficiencias de los procesos. SE Workflow minimiza la transferencia de tareas, reduciendo así los tiempos de ciclo. Por medio de SE Workflow, las empresas pueden obtener la capacidad para implementar, automatizar y controlar sus procesos de negocio, garantizando que los procesos sean ejecutados de la forma que deberían[6].

---

[3] Microsoft Office Visio. Diagramas profesionales de la mano de Microsoft. http://microsoft-office-visio-premium.softonic.com/
[4] Visual Paradigm for UML Enterprise Edition. http://es.kioskea.net/download/descargar-28127-visual-paradigm-for-uml-enterprise-edition
[5] ArchestrA Workflow. http://www.wonderware.es/contents/wonderwareArchestrAWorkflow.asp
[6] SE Workflow Automatización de Procesos de Negocio. http://www.softexpert.es/workflow.php

Destacar que gracias al desarrollo de las nuevas tecnologías existen ya BRMS que traen integrados componentes de flujo de trabajo, ejemplo de ello es el BRMS Drools 5.x, el cual presenta cinco componentes de los cuales uno de ellos está relacionado con el flujo de trabajo, este módulo se conoce como Drools Flow permitiendo añadir a las reglas un flujo de ejecución y la integración de tareas manuales, así como el control de los procesos de ejecución.

Existen varias herramientas para la gestión de reglas, propietarias y no propietarias, entre las propietarias se destacan los BRMS IBM ILOG JRules, Fico Blaze Advisor y Pegasystem Pegarules con precios de licencia elevadísimos y código fuente no disponible. Mientras que dentro de los no propietarios encontramos a CLIPS, NxBRE, Jena, OpenRules, Drools, Hammurapi Rules, Coda Server, OpenL Tablets. De las herramientas no propietarias mencionadas anteriormente solamente tres son BRMS, ellas son:

- **Drools (JBoss Drools / JBoss Rules)**: Probablemente el más conocido, y también el más gigantesco de los proyectos. Se puede considerar un BRMS en toda regla ya que tiene herramientas típicas de los BRMS propietarios (repositorio, editor de reglas WUI, etc). Las reglas pueden escribirse en DRL (el típico), Java, Groovy incluso puede extenderse con DSL's vía XML. Uno de los puntos fuertes de Drools es la integración con el resto de servicios de middleware de jBoss, especialmente con jBPM (Fernández, 2009).
- **OpenRules:** La característica más destacada de OpenRules es la versatilidad y adaptabilidad, ya que tiene una arquitectura enormemente flexible y adaptativa a distintas estrategias de despliegue, definición de reglas en ficheros Excel, aunque se pueden definir también en tablas de base de datos, objetos java, xml, permite utilizar múltiples tipos de accesos a repositorios jerarquizados: ficheros, svn, cvs, base de datos, http, ftp, etc (Fernández, 2009).
- **OpenL Tablets (Claus Ibsen, 2011):** Contiene varias herramientas de productividad y aplicaciones específicas de BRMS como son aplicación web para editar reglas llamadas OpenL Webstudio, aplicación web para implementar reglas como servicios web, repositorio de reglas para almacenar y administrar las reglas y plug-in de Eclipse para la edición de reglas.

Drools se destaca como la herramienta con mayores potencialidades pues tiene las siguientes propiedades:

- BRMS completo (motor de reglas, repositorio, creación y gestión de herramientas).
- Código abierto.
- Documentación fácilmente consumible.
- Arquitectura modular.
- Simple y robusto.
- Repositorio basado en estándares.
- Web 2.0 y la interfaz de usuario RIA para el análisis de negocio.
- Posibilita al analista y al programador colaborar mediante el IDE Eclipse, el repositorio de reglas, y el uso de herramientas Web 2.0.
- Brinda facilidades para editar reglas en ambientes de programación y mediante tablas de Microsoft Excel.
- Más de 10 años de experiencia, superior al resto de los BRMS de código abierto.

En nuestro país existe poca experiencia en la integración de herramientas BRMS al flujo de trabajo de una empresa, el estudio del tema ayudaría al desarrollo de sistemas más reusables y adaptables al cambio. En nuestro trabajo nos planteamos el siguiente **problema de investigación**:

¿Cómo integrar al flujo de trabajo de una empresa un Sistema de Gestión de Reglas de Negocio?

Objetivo general de la investigación:

Integrar los componentes de un BRMS al flujo de trabajo de una empresa mediante una arquitectura orientada a servicios para desarrollar aplicaciones empresariales con un enfoque de reglas de negocio.

Para cumplir con el objetivo general se plantean los **s**iguientes **Objetivos específicos**:

1. Identificar los componentes de un BRMS que son necesarios para integrarlo al flujo de trabajo de una empresa.

2. Modelar una arquitectura que muestre como integrar los componentes a una aplicación externa mediante los servicios identificados.
3. Desarrollar un ejemplo que valide como integrar un BRMS al flujo de trabajo de una empresa.

Para la solución del objetivo general y los específicos nos trazamos las siguientes **Preguntas de investigación**:

1. ¿Cómo una aplicación externa, local o remota, puede interactuar con un BRMS?
2. ¿Cómo integrar un BRMS dentro del flujo de trabajo de una empresa?

Como **Justificación** del proyecto tenemos**:**

La integración de los componentes de un BRMS al flujo de trabajo de una empresa permitirá el desarrollo de productos con un enfoque de reglas de negocio que reaccionen con agilidad ante los cambios de mercado, clientes y reglas de negocio.

Para darle solución al problema científico se plantea la siguiente **Hipótesis de investigación**:

La integración de los componentes de un BRMS al flujo de trabajo de una empresa permitirá desarrollar soluciones informáticas que separen las políticas y reglas de negocio claves en un sistema especializado que facilite la gestión de las mismas.

La **estructura de la tesis** es la siguiente:

Capítulo I. Estudio del marco teórico en el área de los BRMS y su integración al flujo de trabajo de una empresa: analiza el estado del arte relacionado con los BRMS, particularmente cuáles son sus componentes básicos, cuándo y quién los utiliza y las principales ventajas y desventajas de estos sistemas. También se mencionan los tipos de integración de aplicaciones que encontramos en la literatura, la definición de flujo de trabajo y los beneficios que se obtienen al integrar un BRMS al mismo.

Capítulo II. Herramientas para la gestión de reglas de negocio: investiga aspectos relacionados de las herramientas para la gestión de reglas, clasificando las mismas en propietarias y no propietarias y cuales son completamente BRMS. Describe las propiedades de la herramienta BRMS Drools haciendo énfasis en su arquitectura, su lenguaje de

definición, las interfaces para interactuar con otras aplicaciones, como construir servicios web y los plug-in web y de eclipse que brinda para la edición de reglas de negocio.

Capítulo III. Integración de un BRMS al flujo de trabajo "control de historias clínicas para trasplante renal": se describe el diseño e implementación de una solución informática que integra un BRMS al flujo de trabajo "control de historias clínicas para trasplante renal" en el Hospital Universitario "Arnaldo Milián Castro". Se muestra la arquitectura general de un proyecto web que utiliza el motor de reglas Drools y las herramientas que se utilizan para el desarrollo del sistema.

Este trabajo de investigación se desarrolla conjuntamente entre el Laboratorio de Bases de Datos de la UCLV y la UCI-FAR, pues ambas instituciones se complementan en los conocimientos e infraestructura necesarios para abordar con éxito la tarea.

ESTUDIO DEL MARCO TEÓRICO EN EL ÁREA DE LOS BRMS Y SU INTEGRACIÓN AL FLUJO DE TRABAJO DE UNA EMPRESA

CAPÍTULO 1. ESTUDIO DEL MARCO TEÓRICO EN EL ÁREA DE LOS BRMS Y SU INTEGRACIÓN AL FLUJO DE TRABAJO DE UNA EMPRESA

En el siguiente capítulo se efectúa una caracterización de manera general de los componentes básicos de un BRMS, las ventajas y desventajas de los mismos así como cuándo y quién utiliza un BRMS. Investigamos tipos de integración mencionados en la literatura como: Replicación de Datos, Extracción, Transformación y Carga de Datos (ETL4), Integración de Información Empresarial (EII) y las tecnologías relacionadas con Integración de Aplicaciones Empresariales (EAI). Finalmente se abordan aspectos relacionados con el flujo de trabajo de una empresa y la integración de un BRMS a la misma.

1.1 Sistema de Gestión de Reglas de Negocio (BRMS)

Un **BRMS o Business Rule Management System** es un sistema de software utilizado para definir, implementar, ejecutar, monitorear y mantener la variedad y complejidad de la lógica de la decisión que se utiliza en los sistemas operativos dentro de una organización o empresa. Esta lógica, también se conoce como *reglas de negocio*, incluye las políticas, requisitos e instrucciones condicionales que se utilizan para determinar las acciones tácticas que tienen lugar en las aplicaciones y sistemas. Al externalizar las reglas de negocio y proporcionar herramientas para su gestión, un BRMS permite a los expertos de negocio definir y mantener las decisiones que orientan el comportamiento de los sistemas, lo que reduce la cantidad de tiempo y esfuerzo necesarios para actualizar los sistemas de producción y el aumento de la capacidad de la organización para responder a los cambios en el entorno empresarial (Graham, 2007).

1.1.1 Componentes Básicos de un BRMS

En la Figura 1.1 se muestra la arquitectura y los principales componentes de un Sistema de Gestión de Reglas de Negocio

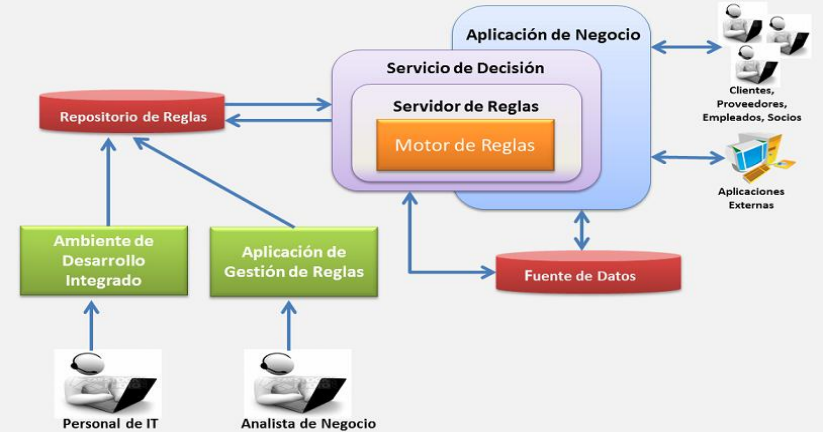


Figura 1. 1Arquitectura de un BRMS

Principales componentes de un BRMS

1. Repositorio de reglas

   Lugar centralizado donde todas las reglas de negocio son guardadas separadas de la lógica de la aplicación, creando una plataforma de fácil migración y transferencia durante los múltiples desarrollos de sistemas. El repositorio de reglas crea la posibilidad de almacenar diferentes versiones de definición de reglas y guardar la historia de los cambios soportando el control de versiones de las reglas.

2. Motor de reglas

   Este componente es capaz de ejecutar y gestionar las reglas de negocio de forma eficiente y adecuada. El motor almacena una serie de hechos que codifican el conocimiento que tiene de un problema. Sobre estos hechos aplica un conjunto de reglas expresadas como sentencias de control condicionales que permiten inferir nuevos hechos. La elección del motor adecuado debe basarse en casos de uso que permitan seleccionar la mejor opción, sin limitarse a una simple comprobación de si el motor cumple o no ciertos requisitos técnicos.

3. Herramientas para la gestión de Reglas

Las herramientas para la gestión de reglas permiten a diseñadores, técnicos y a expertos de negocio definir y manejar la lógica de decisión de la organización. Algunas de estas son: Herramienta de gestión de reglas, editores y plantillas de reglas, componentes para prueba de reglas y plug-in para entornos de desarrollo integrados.

1.1.2 ¿Cuándo se debe utilizar un BRMS?

Un sistema de administración de reglas empresariales es útil en los siguientes escenarios:

- Se necesita administrar un sistema para implementación y modificación de reglas.
- Múltiples usuarios con diferentes niveles de habilidad necesitan acceder y modificar las reglas.
- No existe una infraestructura para administrar las reglas.
- Hay muchas reglas "empresariales" que se deben administrar de manera opuesta a las reglas técnicas que serán parte de una aplicación(Red Hat, 2010).

1.1.3 ¿ Quién utiliza BRMS?

Las siguientes personas con los siguientes cargos en su empresa son principalmente los que utilizarán el sistema de administración de las reglas empresariales:

- Analistas empresariales
- Expertos en reglas
- Desarrolladores
- Administradores de reglas
- JBoss Enterprise BRMS Platform permite asignar diferentes roles a diferentes usuarios para controlar los activos y las funcionalidades que se exponen (Red Hat, 2010).

1.1.4 Ventajas y desventajas

Los BRMS permiten a las aplicaciones basadas en Reglas:

- Capturar definiciones y reglas que son sujeto a cambios frecuentes.

- Implementar esos cambios de manera rápida y fácil dentro de una aplicación.
- Gestionar y escribir reglas en un lenguaje de negocio familiar.
- Los usuarios expertos pueden actualizar la funcionalidad de acuerdo a sus políticas.
- Los cambios requeridos por los usuarios pueden ser implementados sin cambios en el código, aislando el cambio y probando sólo la regla que ha sido modificada.
- El costo de mantenimiento se reduce al o tener que recodificar, reprobar, recompilar y reinstalar el aplicativo por cada cambio menor.

Algunas desventajas de los BRMS:

- Se necesita gran experiencia en la materia requerida para productos de proveedores específicos. Además del análisis orientado a objetos y prácticas de diseño, los desarrolladores técnicos deben saber cómo escribir reglas e integrar el software con los sistemas existentes.
- Ciclo de desarrollo de largo plazo debido a regir la recolección, integración con los sistemas existentes, las restricciones de seguridad, la migración, el seguimiento de regla y edición de éstas.
- La reducción del departamento de TI de confianza nunca puede ser una realidad debido a la continua introducción de nuevas consideraciones de reglas de negocio o perturbaciones del modelo de objetos.
- El acoplamiento de una aplicación de proveedor de BRMS para la aplicación empresarial puede ser demasiado estrecho para sustituirlo por otro proveedor de aplicaciones BRMS. Esto puede llevar a problemas de costo para los beneficios.
- El lenguaje de reglas de Drools no permite definir reglas complejas, solo permite gestionar reglas sencillas.

1.2 Integración de aplicaciones

En este apéndice se realizará una breve caracterización de los cuatro tipos de integración considerados más importantes: Replicación de Datos, Extracción, Transformación y Carga de Datos (ETL4), Integración de Información Empresarial (EII) e Integración de Aplicaciones Empresariales (EAI) dentro de la cual nos referiremos a las dos formas de integración asincrónica: Punto a Punto y Concentrador-clientes.

1.2.1 Replicación de Datos

La Replicación de Bases de Datos es una técnica de integración que se basa en la creación y mantenimiento de múltiples copias de una misma base de datos. En la mayoría de las implementaciones de Replicación, un servidor mantiene la copia primaria de la base de datos y servidores adicionales mantienen las copias esclavas de la misma. La transferencia se realiza de Base de Datos a Base de Datos (este es el estilo más antiguo de integración), lo cual es una desventaja para la resolución del problema a resolver dado la heterogeneidad de las fuentes (Basallo et al., 2010).

1.2.2 Extracción, Trasformación y Carga de Datos (ETL4)

La técnica Extracción, Transformación y Carga de Datos (Extract, Transform and Load), como su nombre lo indica extrae información de un sistema fuente, transforma esos datos para satisfacer los requerimientos del negocio y carga el resultado en el sistema destino. Tanto la fuente como el destino son generalmente Bases de Datos y archivos. La transformación puede implicar la reestructuración y reconciliación del registro de datos, limpieza del contenido de datos (es decir, revisados por si existen discrepancias y eliminación de datos obviamente falsos), y/o agregación del contenido de datos. Esto sucede mediante una serie de procedimientos especiales que permiten obtener un formato unificado común y mejorado. Sólo después de la revisión y unificación de los datos estos son cargados. Esta técnica se encarga de la integración de datos, no de aplicaciones, y obtiene los datos directamente de las Bases de Datos. De acuerdo a las características de esta técnica se concluyó que no es la más adecuada para la solución que se necesita (Basallo et al., 2010).

1.2.3 Integración de Información Empresarial

La Integración de Información Empresarial es un mecanismo de transformación y acceso a datos transparente y optimizado para suministrar una única interfaz a lo largo de los datos de las organizaciones. Dicha interfaz permite acceder a los datos y el resultado de este método es un Sistema de Información Heterogéneo Distribuido, Virtualmente Integrado. Este tipo de solución consiste en crear un intermediario que contenga los directorios de la Base de Datos y que a su vez sirva de canal de consulta y representación de la información recuperada. La información es capturada en tiempo real lo que implica que las fuentes de datos tengan una estructura tecnológica sólida y bien establecida. EII protege a las aplicaciones de la complejidad de recuperar datos de múltiples localizaciones, donde los datos pueden diferir en semántica y formato, y emplear diferentes interfaces de datos. Teniendo en cuenta estos aspectos, para la integración de Datos a Tiempo Real la técnica EII constituye una buena alternativa, sin embargo no es factible para la integración de aplicaciones(Basallo et al., 2010).

1.2.4 Integración de Aplicaciones Empresariales (EAI)

La Integración de Aplicaciones Empresariales es el proceso de integrar múltiples aplicaciones desarrolladas independientemente, que utilizan tecnología incompatible y que son gestionadas de forma independiente, permitiendo que se comuniquen e intercambien transacciones de negocio, mensajes, y datos entre sí. Uno de los principales objetivos de EAI es proporcionar acceso transparente a la amplia gama de aplicaciones que existen en una organización. Las características más importantes de esta tecnología es que se utiliza para la integración de Aplicaciones – a – Aplicaciones y proporciona un enfoque de integración orientado a proceso basado en mensajes XML. Es generalmente utilizada para el procesamiento de transacciones de negocio operacional en tiempo real (Basallo et al., 2010).

La EAI ha demostrado ser un área de aplicación importante en los procesos empresariales, debido a que las empresas se enfrentan al reto de integrar sistemas complejos de software en un entorno de información heterogénea que ha crecido de manera evolutiva durante años. La mayoría de los sistemas de aplicación se han desarrollado independientemente el

uno del otro, y cada aplicación almacena los datos localmente, ya sea en un sistema de base de datos o algún otro almacén de datos (Weske, 2007).

En la Figura 1.2 se muestra una arquitectura que responde a la integración de aplicaciones empresariales.

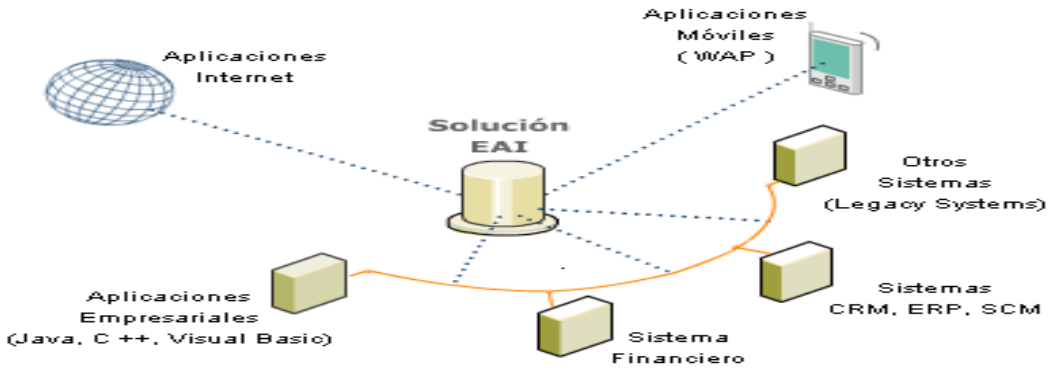


Figura 1. 2 Integración de Aplicaciones Empresariales

El propósito de la EAI (*Enterprise Application Integration*) es lograr la interoperabilidad y organización del flujo de información entre aplicaciones heterogéneas, es decir, asegurar la comunicación entre las distintas aplicaciones y formar el sistema de información de la empresa, incluso de los clientes, socios o proveedores. Por lo tanto, y en primer lugar, un proyecto de EAI implica implementar una arquitectura bajo la cual las distintas aplicaciones se comuniquen entre sí. En consecuencia, esto conlleva el desarrollo de **conectores** (*middleware*) que posibilitan la interfaz de aplicaciones mediante el uso de distintos protocolos de comunicación (por lo general exclusivos). Sin embargo, el proyecto de EAI va más allá de la interoperabilidad de las aplicaciones: ofrece la posibilidad de definir un flujo de trabajo entre las aplicaciones; así, representa una alternativa a la ERP con un enfoque más modular. No obstante, la EAI todavía presenta limitaciones relacionadas con la rigidez de la herencia, porque se debe modificar el middleware cuando hay cambios importantes en las aplicaciones de Servicios Web[7].

[7]2013. Introducción al concepto de EIA. Available: http://es.kioskea.netcontentsentrepriseeai.php3

Existen dos formas de integración asincrónica como son la integración Punto a Punto y la integración Concentrador-clientes. En los siguientes subepígrafes se hará referencia a estas dos formas de integración.

I.2.4 .1 Integración Punto a Punto

Cuando se trata de muy pocas aplicaciones a integrar este método es el más adecuado, deteriorándose al tratar de integrar más sistemas. El número de puntos de integración es el doble del número de sistemas a integrar siendo esto problemático debido al acoplamiento entre los sistemas (Lin, 2005), ejemplo de ello se muestra en la Figura 1.3:

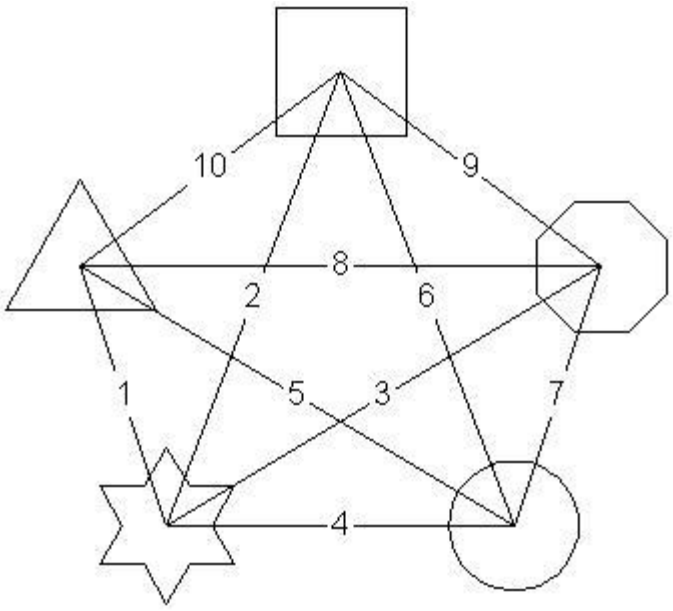


Figura 1. 3Número de conexiones punto a punto (Lin, 2005)

La arquitectura de integración de aplicaciones empresariales resultante de la integración punto a punto no responde bien a los cambios que se realicen en el sistema. La razón se debe al cableado de las interfaces. Cualquier cambio en la arquitectura de la aplicación requiere la adaptación de las interfaces respectivas. Una plataforma middleware orientada a mensajes implementa la integración de aplicaciones empresariales mediante el envío y la recepción de mensajes (Weske, 2007).

I.2.4 .2 Integración Concentrador-clientes

El paradigma de Concentrador-clientes se basa en un nodo centralizado y un número de aplicaciones clientes que están directamente conectadas al concentrador, las aplicaciones

clientes no están conectadas. La aplicación concentradora es un intermediario, que integra las aplicaciones clientes. Una característica importante de las arquitecturas de Concentrador-clientes es que el remitente de un mensaje no necesita codificar el receptor del mensaje. En su lugar, cada mensaje se envía al centro de integración de aplicaciones empresariales, el concentrador, el cual está configurado de tal manera que la estructura del mensaje y el contenido puede ser utilizado para detectar automáticamente el receptor o receptores de un mensaje. La ventaja de estas arquitecturas centralizadas middleware es que el número de conexiones puede ser reducido. Ya no son conexiones en el orden de N x N para conectar N sistemas de aplicación (Weske, 2007). Weske presenta la integración de Concentrador-clientes como se observa en la Figura 1.4:

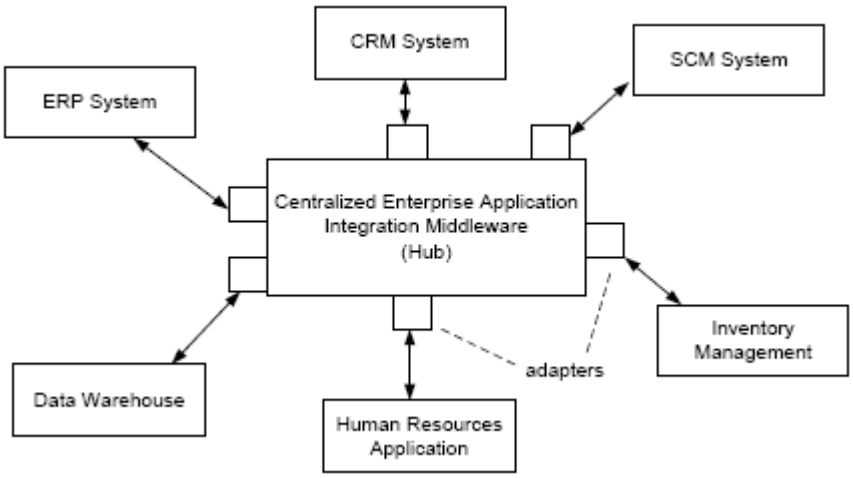


Figura 1. 4 Arquitectura de Integración Concentrador-clientes (Weske, 2007)

1.3 Flujo de Trabajo

La evolución de las arquitecturas de software empresarial y de gestión de la organización a nivel de procesos de negocio llevó a la gestión de flujo de trabajo. El logro importante de la gestión del flujo de trabajo es la representación explícita de las estructuras de procesos en los modelos de procesos y la promulgación controlada de los procesos de negocio de acuerdo con estos modelos. El enfoque basado en modelos facilita un alto grado de flexibilidad, porque los modelos de procesos pueden ser adaptados para cumplir los nuevos requisitos, y los modelos de procesos modificados inmediatamente se pueden utilizar para adoptar los procesos de negocio (Weske, 2007).

Flujo de trabajo –también conocido como BPMS (Business Process Management Systems)- se define como un sistema de secuencia de tareas de un proceso de negocio. Su definición y control puede ser manual, informatizado o mixto. Organiza y controla tareas, recursos y reglas necesarias para completar el proceso de negocio.

El Workflow Management Coalition define los flujos de trabajo y el sistema de gestión de flujo de trabajo de la siguiente manera:

Definición: Un sistema de gestión de flujo de trabajo es un sistema de software que define, crea y gestiona la ejecución de flujos de trabajo mediante el uso de software, que se ejecuta en uno o más motores de flujo de trabajo, que es capaz de interpretar la definición del proceso, interactuar con los participantes del flujo de trabajo, y, en caso necesario, recurrir a la utilización de las herramientas informáticas y aplicaciones (Weske, 2007).

Definición: Flujo de trabajo es la automatización de un proceso de negocio, en su totalidad o en parte, durante el cual los documentos, información o tareas pasan de un participante a otro para la acción, de acuerdo con un conjunto de reglas de procedimiento (Weske, 2007).

Escenarios de integración de aplicaciones empresariales son candidatos típicos para los flujos de trabajo del sistema. El diseño y la ejecución de flujos de trabajo del sistema pueden ser considerados como un tipo de programación de alto nivel, donde la funcionalidad proporcionada por los sistemas de aplicación caracteriza los bloques de construcción que se organizan dentro de un flujo de trabajo del sistema. En las plataformas de integración de aplicaciones empresariales sin un componente de proceso dedicado, la interacción entre los sistemas de aplicación está representada por las reglas, que se utilizan para reenviar los mensajes en función de su tipo o contenido (Weske, 2007).
Las técnicas de modelado de procesos se pueden utilizar para proporcionar una representación explícita de las relaciones entre las aplicaciones de empresa. Los modelos de procesos constituyen la base conceptual para definir cuándo y en qué condiciones las aplicaciones empresariales están realmente invocadas en el contexto de un escenario de integración. Por lo tanto, un componente de proceso dedicado y responsable para el modelado y la promulgación de los procesos empresariales en los escenarios de integración de aplicaciones son adecuados (Weske, 2007).

Un escenario de flujo de trabajo según el autor anteriormente mencionado lo podemos definir como se muestra en la Figura 1.5:

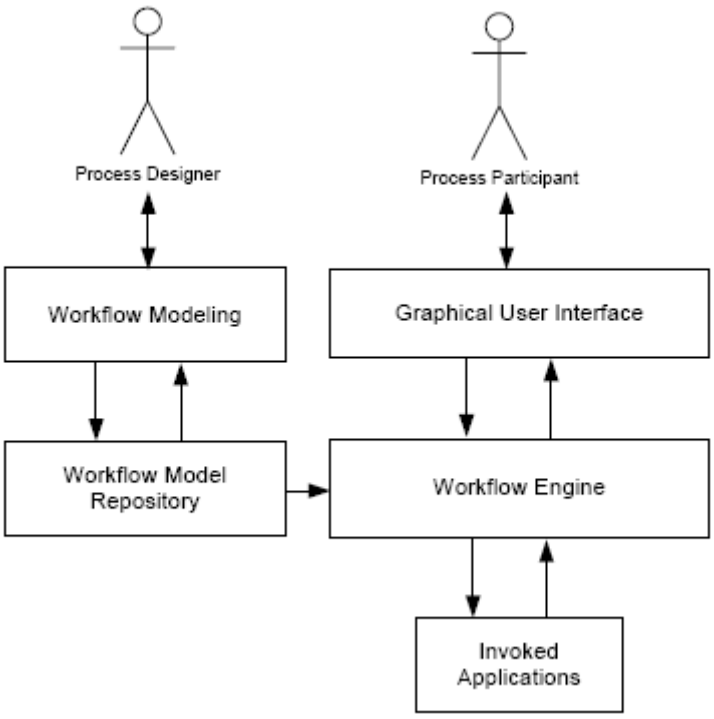


Figura 1. 5 Arquitectura de un sistema de flujo de trabajo (Weske, 2007)

Para que el flujo de trabajo se pueda implementar en una empresa, es necesario que se cumplan condiciones mínimas, relacionadas básicamente con su tamaño (que por su estructura haya necesidad de coordinación) y con la infraestructura computacional necesaria. Esto significa que todos los que participan en el flujo de trabajo, deben contar con un computador conectado a una red, con el fin de ir registrando las actividades que realizan[8].

Las fases a seguir para el flujo de trabajo son fundamentalmente cuatro según el artículo de Cabrera:

1. El cliente realiza la petición o el ejecutor, una oferta.

[8] CABRERA, M. O. 2012. *Workflow en su empresa* [Online]. Laboratorio de Gestión Electrónica Empresarial. Available: http://www.usmp.edu.pe/publicaciones/boletin/fia/info40/workflow.html [Accessed].

2. Se realiza la negociación hasta que se llega a un acuerdo en torno a las condiciones de satisfacción y tiempo de terminación (fin) que han sido presupuestados.
3. El ejecutor realiza su trabajo prometido y deberá reportar el cumplimiento de la acción que le fue solicitada.
4. El cliente deberá en esta fase declarar su satisfacción con el trabajo realizado por el ejecutor o bien manifestar su desacuerdo.

A cada fase le corresponde, por lo tanto, un acto lingüístico donde se dan a conocer cada una de las peticiones y/o respuestas del proceso en cuestión, y dos actores (cliente y ejecutor). Los datos asociados a estos actos lingüísticos o conversaciones deben ser ingresados en formularios diseñados por ó en un flujo de trabajo, los que deberán ser llenados por los usuarios autorizados de acuerdo a las características y peticiones que en ellos figuren(Cabrera, 2012)[3].

1.4 Beneficios que se obtienen al integrar un BRMS a un flujo de trabajo

Las aplicaciones empresariales generalmente están involucradas dentro de procesos de negocio o flujos de trabajo. Integrar un BRMS a flujos de trabajo empresariales, a menudo complejos y automatizados, requiere una nueva aproximación. Externalizar las reglas de negocio hace que los procesos de la empresa funcionen mucho mejor, por tanto integrar un BRMS al flujo de trabajo de la empresa implica la publicación de interfaces que permitan ejecutar reglas por aplicaciones externas y devolver el resultado de las mismas al flujo de trabajo que lo llama.

La integración de un BRMS al flujo de trabajo de una empresa presenta los siguientes beneficios:

- Ahorro de tiempo y mejora de la productividad y eficiencia de la empresa, debido a la automatización de muchos procesos de negocio.
- Mejora del control de procesos a través de la normalización de los métodos de trabajo.
- Mejor atención y servicio al cliente; un incremento en la coherencia de los procesos da lugar a una mayor previsibilidad en los niveles de respuesta a los clientes.

- Mejora en los procesos; mayor flexibilidad de acuerdo con las necesidades empresariales.
- Optimización de la circulación de información interna con clientes y proveedores.
- Integración de procesos empresariales.
- Disminuye el tiempo de desarrollo y mantenimiento de aplicaciones.
- La lógica de negocio es más reusable.
- Consistencia entre las aplicaciones empresariales.
- Mayor alineación y entendimiento entre el negocio y las tecnologías informáticas (TI).

1.5 Conclusiones Parciales

En este capítulo se realizó un estudio del estado del arte de los principales conceptos relacionados con los Sistemas de Gestión de Reglas de Negocio, integración de aplicaciones empresariales y flujo de trabajo. Mencionándose los componentes fundamentales de un BRMS y los principales programas de este tipo, teniendo en cuenta sus ventajas y desventajas. Se abordan conceptos relacionados con la integración de aplicaciones empresariales, sugiriéndose las principales herramientas para la gestión de flujo de trabajo y los beneficios que aporta integrar un BRMS al mismo. En el siguiente capítulo se analizan detalladamente los componentes de los Sistemas para la Gestión de Reglas de Negocio.

HERRAMIENTAS PARA LA GESTIÓN DE REGLAS DE NEGOCIO

CAPÍTULO 2. HERRAMIENTAS PARA LA GESTIÓN DE REGLAS DE NEGOCIO

Este capítulo aborda aspectos generales sobre los criterios tomados para la selección de la herramienta a integrar al framework de desarrollo de aplicaciones de una organización, se realiza una valoración de la misma en cuanto a arquitectura, lenguaje de definición para las reglas de negocios, las interfaces para interactuar con otras aplicaciones y la construcción de servicios web con la misma.

2.1 Preselección de herramientas

Actualmente existen varias herramientas para la gestión de reglas, las mismas se dividen en dos campos las llamadas propietarias y las no propietarias. Dentro de la propietarias se destacan los BRMS IBM ILOG JRules, Fico Blaze Advisor y Pegasystem Pegarules con precios de licencia elevadísimos y código fuerte no disponible por lo que en nuestro análisis descartamos a este tipo de herramientas pues nuestro trabajo va dirigido a organizaciones cuyo principal objetivo es introducir el enfoque de reglas de negocio dentro de su framework de desarrollo y mantener a su vez la soberanía tecnológica por lo cual nos enfocaremos en software Open Source y gratis.

Muchas son las herramientas que existen dentro de la comunidad Open Source; inicialmente se realiza una preselección de aquellas herramientas que cumplan los siguientes criterios:

- Organización o empresa por la que está respaldada.
- Versiones liberadas y actualizadas.
- Componentes de un BRMS que implementan.

Las herramientas preseleccionadas son las siguientes:

- **CLIPS[9]:** (C Language Integrated Production System), es una herramienta de dominio público utilizada para construir sistemas expertos, desarrollada por *Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center*. La última versión liberada es CLIPS_6.30_beta. Contiene herramientas para la gestión de reglas.
- **NxBRE[10]:** Es el primer motor de reglas de código abierto para la plataforma NET, está basado en JxBRE(Sloan Seaman, 2002) y desarrollado por *Trac,Edgewall Software*. Ofrece dos aproximaciones diferentes para utilizarlo: como motor de inferencias o como motor de flujo. Contiene un motor de reglas.
- **Jena[11]:** Es un framework Open Source para la Web Semántica, originario de HP Labs. Si bien incluye un motor de reglas genérico, está enfocado fuertemente dentro del marco de la Web Semántica, brinda un ambiente de desarrollo para interactuar con RDF, RDFS, OWL y SPARQL. Incluye diferentes razonadores de los cuales interesa el de propósito general ya que es un razonador basado en reglas. Soporta razonamiento hacia adelante, atrás y una estrategia híbrida de inferencia. Tiene un motor de reglas de negocio y un repositorio.
- **OpenRules[12]:** Es un BRMS que ofrece la metodología y probadas herramientas de código abierto que permite a los analistas de negocio crear un repositorio de reglas de negocios que se utilizará en toda la empresa como base para las normas basadas

---

[9]Riley, G. (1985). "CLIPS: A Tool for Building Expert Systems." 2012, from http://clipsrules.sourceforge.net/.

[10] (2008, 21 July 2012). "NxBRE open-source Business Rule Engine for .NET plattform." Retrieved 11 November 2012, 2012, from http://sourceforge.net/apps/trac/nxbre/wiki.

[11] The Apache Software Foundation (2010, 24 November 2012). "Apache Jena project." Retrieved 22 November 2012, 2012, from http://jena.apache.org/index.html.

[12] OpenRules Inc (15 October 2012). "OpenRules." Retrieved 22 November 2012, 2012, from http://openrules.com/.

en aplicaciones con negocios complejos, el procesamiento y la lógica de presentación. OpenRules ayuda a los clientes a mantener sus repositorios de reglas utilizando las conocidas interfaces gráficas proporcionadas por MS Excel u OpenOffice. Su última versión es OpenRules 6.2.2 y salió en Octubre 15, 2012. Como componentes tiene reglas de repositorio en Eclipse y motor de reglas.

- **Drools[6]:** Es un BRMS basado en inferencia de encadenamiento hacia adelante (forward chaining), más correctamente conocido como sistema de reglas de producción, usando una implementación avanzada del algoritmo Rete. Incluido dentro de lo que se denomina JEMS (JBoss Enterprise Middleware Systems) y está desarrollado por "JBoss, Red Hat" y la última versión que salió al público hasta el 24 de Octubre de 2012 es Drools 5.5.0.CR1. Tiene motor de reglas, repositorio y herramientas para la gestión de reglas.

- **Hammurapi rules[13]:** Es un motor de evaluación de reglas de tipo forward chaining compatible con el estándar JSR-94. Las reglas se definen directamente utilizando el lenguaje Java. No hay pasos de interpretación, una vez que el conjunto de reglas se instancia a partir de definiciones en XML las mismas se compilan a bytecode Java. Provee un plug-in de Eclipse para editar y modificar reglas. Está desarrollada por *Hammurapi Group* y su última versión es hammurapirules_4.1.1. Actualmente ha sido reemplazado por *Event Bus.*

- **CodaServer[14]:** Es un motor de reglas de negocio; similar a una base de datos, pero especializado en hacer aplicaciones de negocio simples. Cuando sus procesos cambian, su software puede incorporar estos cambios automáticamente sin el recodificar. La última versión disponible es *CodaServer 1.0 Alpha 3.*

---

[13] Hammurapi Group (2010, 12 January 2010). "Hammurapi rules." Retrieved 22 November 2012, from http://www.hammurapi.com/dokuwiki/doku.php/products:hammurapi_rules:start.

[14] RestlessDEV (2008, 15 March 2009). "CodaServer." Retrieved 22 November 2012, from http://codaserver.com/.

- **OpenL Tablets(Claus Ibsen, 2011):** Es un BRMS y un motor de reglas de negocio (BRE) basado en representación de tablas de reglas. El motor implementa un algoritmo optimizado secuencial también incluye los tipos de tablas como tabla de decisión, árboles de decisión y hojas de cálculo. OpenL Tablets contiene varias herramientas de productividad y aplicaciones específicas de BRMS: aplicación web para editar reglas llamadas OpenL Webstudio, aplicación web para implementar reglas como servicios web, repositorio de reglas para almacenar y administrar las reglas y plug-in de Eclipse para la edición de reglas. Es desarrollado por *Exigen Services.* Hasta el primero de noviembre de 2012 su última versión estable era la 5.9.4.

La mayoría de las herramientas preseleccionadas cuentan con motor de reglas y repositorio, pero en general carecen de las facilidades para gestionar las reglas de un BRMS mencionados. Solo Drools, OpenRules y OpenL Tablets son BRMS. La decisión de cuál de estas herramientas seleccionar para integrarla al framework de desarrollo de aplicaciones de una organización la determinó el tipo de licencia, la documentación y el lenguaje de reglas.

1. *La Licencia*

   Las licencias de OpenRules, Drools y OpenL Tablets son open source, sin embargo, la GPL de OpenRules no es conveniente para una empresa pues impone restricciones a la licencia de los productos que utilizan software GPL, los cuales deberán ser open source y gratuitos. Las licencias de Drools y OpenL Tablets no tienen esta limitante. Por tanto descartamos el BRMS OpenRules en nuestro análisis.

2. *La documentación*

   Una de las principales características que influyen en la decisión de qué herramienta es la más indicada para utilizar es cuan de fácil asimilar es la misma dentro de una empresa. Una herramienta bien documentada disminuye la curva de aprendizaje de sus usuarios y facilita reaccionar más rápido ante la competencia. De los BRMS analizados OpenL Tablets es el que tiene la documentación más escasa, lo cual hace tedioso el estudio y utilización de la herramienta. La documentación de OpenL Tablets es adecuada y la de Drools es muy buena.

3. Lenguaje de Reglas

En OpenRules y OpenL Tablets el lenguaje de reglas es XML, el cual es difícil de entender por el ojo humano. Drools además de XML tiene un lenguaje de reglas propio (DRL) y también permite definir lenguajes de dominio específico (DSL), más cercanos al lenguaje natural para ser utilizados por usuarios con menos conocimientos técnicos. Los usuarios más experimentados pueden definir sus reglas en el lenguaje de programación Java.

Los tres BRMS analizados tienen una arquitectura modular y publican sus funcionalidades mediante servicios Web. Esto es muy deseable pues esta tecnología permite construir sistemas inter-empresariales que consumen los servicios del BRMS. Igualmente permiten la edición de reglas utilizando el IDE Eclipse y hojas de cálculo. El uso de hojas de cálculo favorece el uso de la herramienta por usuarios expertos del negocio una vez desplegado el sistema. Los usuarios más cercanos a las TI, como son los desarrolladores de software de la empresa, pueden gestionar reglas del producto que están desarrollando desde Eclipse. En el caso de Drools y OpenL Tablets las reglas también pueden ser editadas a través de una interfaz Web. Las limitantes de la licencia GPL hacen que la introducción de OpenRules no sea factible dentro de una empresa de software pues obliga a que el producto desarrollado sea Open Source y gratis. Drools y OpenL Tablets están bien documentados, aunque Drools tiene una comunidad más activa que genera ayudas y manuales con mayor frecuencia. Drools es el único que posibilita la definición de lenguajes de dominio específico (DSL) lo cual acerca al usuario no especialista en programación o informática a los términos del dominio del problema. Además de los componentes de un BRMS Drools incluye módulos que proporcionan capacidades de flujo de trabajo a su plataforma, procesamiento de eventos y planificación de recursos.

Por lo anteriormente expuesto se recomienda utilizar la herramienta ENTERPRISE BRMS (DROOLS) para integrarla al flujo de trabajo de una empresa. El próximo epígrafe estará dedicado completamente a la herramienta Drools como Sistema de Gestión de Reglas de Negocio.

2.2 Jboss Rules (Drools)

Dentro de los sistemas de gestión de reglas de negocio en Java más populares hoy en día se encuentra **Drools**. **Drools** es un **BRMS** de JBoss con un motor de reglas basado en una adaptación orientada a objetos del algoritmo Rete. Permite expresar de una forma más natural las reglas de negocio interactuando con los objetos de negocio. Provee separación de lógica (reglas) y datos (hechos). Es software libre distribuido según los términos de la licencia Apache. Drools soporta el estándar JSR-94 para su motor de reglas de negocio y framework de empresa para construcción, mantenimiento y refuerzo de políticas de empresa en una organización, aplicación o servicio. Drools usa JCR (JackRabbit) para gestionar el repositorio de reglas, y el estándar JAAS para la autorización y autenticación. También existe un plug-in de Eclipse para facilitar el desarrollo con esta herramienta[15].

2.2.1 Arquitectura

Drools a lo largo de los últimos años se nos ha presentado en varias versiones debido a que se le han ido añadiendo nuevas funcionalidades en cada una de sus versiones. La versión 1.0 empezó con una búsqueda de fuerza bruta lineal, mientras que la versión 2 se basa en el algoritmo Rete el cual mejora el rendimiento de Drools y en un principio las reglas fueron escritas en XML(Bali, 2009). La versión 3 de Drools consistía básicamente en un motor de reglas usando una implementación del algoritmo RETE e introduce el nuevo formato .drl para escribir reglas, la versión 4 comienza la transformación para convertirse en lo que es en la actualidad aunque inicialmente estaba orientado solamente a reglas de negocio.

En la versión 5 actual se convierte en un completo BRMS donde introduce la plataforma de integración de lógica de negocio que unifica e integra las reglas, flujo de trabajo y procesamiento de evento. La cual presenta una arquitectura modular formada por 5 componentes integrados, aunque cada uno puede ser utilizado por sí solo:

---

[15] Mazza, R. G. (mayo 2009). "Primeros Pasos con Drools." 18-12-2012, from http://www.dosideas.com/noticias/java/592-primeros-pasos-con-drools.html.

- *Guvnor:* cuenta con un repositorio centralizado para las Bases de conocimiento, con una web rica basada en GUI, editores y herramientas para ayudar en la administración de números grandes de reglas. El repositorio permite mantener un control de versiones de reglas, modelos, funciones de los procesos que se relacionan con estas bases de conocimiento. El acceso se controla y es posible cerrar con llave y restringir los expertos del dominio evitando el acceso a personal no autorizado.
- *Expert:* es el motor de reglas de Drools, este es un declarativo, basado en reglas y tiene ambiente de codificación. Esto le permite enfocarse en lo que usted "quiere hacer" y no el "cómo hacerlo".
- *Flow (*jBPM 5*):* proporciona las capacidades flujo de trabajo a la plataforma Drools. Un proceso comercial o flujo de trabajo describe el orden en que una serie de pasos necesitan ser ejecutados, usando una gráfica de flujo. Esto hace mucho más fácil el describir una composición compleja de varias tareas. Los procesos son especialmente útiles describiendo los procesos largos. *Drools Flow* le permite a los usuarios finales especificar, ejecutar y supervisar su lógica de negocio.
- *Fusion:* Este módulo le agrega a la plataforma Drools la capacidad de procesamiento complejo de eventos y razonamiento temporal.
- *Planner:* perfecciona el uso de los recursos del negocio. Cada organización enfrenta los problemas de la planificación y este proporciona un juego limitado de recursos y servicios (los empleados, recursos, tiempo y dinero) y ayuda a resolver los problemas de la planificación eficazmente, combinando heurísticas de optimización y metaheurísticas para resolver de forma eficaz cualquier problema NP-completo.

Solo los módulos Expert y Guvrnor pertenecen a la categoría de componentes de un BRMS. El módulo Expert es el motor de reglas de un BRMS, mientras Guvnor es el componente de repositorio de reglas. Los restantes módulos, Flow, Fusion y Planner, hacen de Drools además de un BRMS una plataforma de integración de lógica de negocio.

Drools 5 posee también un módulo de aplicación llamado drools-server.war el cual necesita ser ejecutado por un servidor de aplicaciones. Drools Server utiliza múltiples agentes del conocimiento que permite escalar de manera transparente a las aplicaciones, además de tener varios servidores que pueden albergar diferentes sesiones con diferentes cargas, en

vez de tener una gran cantidad de sesiones dentro de la misma aplicación. Su configuración se proporciona durante el arranque de los archivos de propiedades, tras el arranque el servidor espera peticiones de los clientes. Cuando se recibe una solicitud, se crea la sesión StatelessKnowledgeSession de ejecución de la regla, después de crear esta sesión se establecen las variables globales, se insertan hechos y se ejecuta las reglas. Cuando las reglas terminan los resultados son enviados al cliente. El servidor puede comunicar con el cliente utilizando formatos XML o JSON. Drools Server necesita ser desplegado en un servidor de aplicaciones y se puede llamar mediante REST y SOAP(Bali, 2009) .

Drools 5 ofrece un entorno de desarrollo que permite:

- Definir visualmente las reglas (.drl).
- Crear DSLs (nuevo lenguaje de reglas) enfocado al negocio concreto: soportando esta sintaxis con autocompletado.
- Definición de Flujos de forma visual.
- Depuración de las reglas.
- Testing de Reglas.
- Uso de Tablas de decisión.
- Editor guiado de reglas.

2.2.2 Lenguaje(s) de definición de reglas de negocios

Drools es un sistema de administración de reglas de negocio (BRMS) con un motor de reglas basado en una adaptación orientada a objetos del algoritmo Rete. Permite expresar de una forma más natural las reglas de negocio interactuando con los objetos de negocio, proviniendo separación de lógica (reglas) y datos (hechos). Provee soporte para la programación declarativa, y es lo suficientemente flexible para expresar la semántica del problema con un lenguaje específico de dominio (DSL)[16].

---

[16] Mazza, R. G. (mayo 2009). "Primeros Pasos con Drools." 18-12-2012, from http://www.dosideas.com/noticias/java/592-primeros-pasos-con-drools.html.

Drools cuenta con la implementación completa de la JSR-94 Rule Engine API, existiendo un plug-in de Eclipse para facilitar el desarrollo con esta herramienta.

Para especificar las reglas, Drools utiliza el lenguaje de reglas (DRL) con el objetivo de especificar las condiciones, acciones y funciones de las mismas, las cuales se puede expresar con distintos lenguajes, como Java y MVEL, guardándose las reglas en archivos de texto con la extensión .drl[13].

Con Drools también podemos definir un DSL, que es el lenguaje coloquial del usuario para definir el comportamiento de su sistema de acuerdo al negocio, permitiendo que el usuario pueda modificar su sistema de acuerdo a las necesidades de su negocio, sin demasiada intervención del departamento de desarrollo ni teniendo que lidiar con abstracciones e interfaces[13].

I.2.2 .1 Archivo .dsl

Para crear un lenguaje específico con Drools basta crear un fichero con extensión .dsl donde se definirá las expresiones, que normalmente son expresiones del lenguaje natural y luego se pasa a reemplazar dichas expresiones en el fichero de reglas .drl o .dsrl. Cada fichero .dsl tiene un formato, una definición de variables, utilización de expresiones regulares, funciones y permite añadir condiciones(Browne, 2009).

I.2.2 .2 Archivo .drl

Un archivo DRL (a menudo con la extensión. drl) es el archivo donde Drools almacena sus reglas, casi de la misma manera que un archivo con la extensión .doc es un archivo de Word, y un archivo con la extensión .xls es un archivo de Excel. Un archivo DRL es simplemente un archivo de texto que puede ser abierto en el Bloc de notas, o con uno de los editores de texto, como el IDE de JBoss (Browne, 2009). La estructura de una regla dentro de este archivo se muestra en la Figura 2.1 donde <name> es el nombre de la regla, <attribute> son los posibles atributos que puede tener la misma los cuales pueden ser no-loop, lock-on-active, salience, agenda-group, auto-focus, ruleflow-group, entre otros;

<conditional element> es la sentencia condicional a evaluar por la regla, si se cumple esta sentencia se pasa a realizar una o varias acciones (<conditional element>).

```
1  rule "<name>"
2      <attribute>*
3    when
4      <conditional element>*
5    then
6      <action>*
7  end
```

Figura 2. 1 Sintaxis de una regla en un fichero .drl

2.2.3 Interfaces para interactuar con otras aplicaciones

El Drools Server por defecto viene configurado con dos interfaces: SOAP y RESTful. El REST o RESTful como también se le conoce -según (Asúa, abril 2012)- utiliza casi siempre HTTP como método de comunicación y XML o JSON para intercambiar datos. Cada URL representa un objeto sobre el que se pueden utilizar los métodos POST, GET, PUT y DELETE. Utiliza el idioma de la web.

La tecnología SOAP (Protocolo Simple de Acceso a Objetos) es un protocolo basado en XML que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP, SMTP, etc. SOAP especifica el formato de los mensajes, el cual está compuesto por un envelope (sobre), cuya estructura está formada por los siguientes elementos: header (cabecera) y body (cuerpo), a continuación en la Figura 2.2 se presenta un ejemplo de la estructura de un mensaje SOAP.

El elemento Envelope es el elemento principal del documento por lo que debe estar presente en todos los mensajes, incluye las referencias a los espacios de nombres utilizados por el documento y también puede incluir algunos atributos adicionales. Los espacios de nombres se utilizan para garantizar la unicidad de los elementos y evitar ambigüedades (Moscatelli, 2001).

```
1  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
2    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4
5  </SOAP-ENV:Envelope>
```

Figura 2. 2 Estructura del Envelope en un mensaje SOAP

El Header se trata de un elemento opcional (véase Figura II.4), es un mecanismo genérico para extender las características de los mensajes SOAP de una manera descentralizada y sin un acuerdo previo entre las partes que se comunican. En caso de estar presente debe ser el primer hijo de la construcción ENVELOPE(Moscatelli, 2001).

El Body es el elemento que actúa como contenedor para la información que se envía al receptor del mensaje. Esta construcción debe estar presente siempre en los mensajes SOAP y debe estar a continuación del HEADER, si está presente, o ser el primer hijo de ENVELOPE si el HEADER no está presente. Los usos típicos de esta construcción son proveer un mecanismo simple de intercambiar información con el receptor del mensaje SOAP. En esta parte del mensaje es donde se encuentran las invocaciones RPC o bien el resultado de la invocación(Moscatelli, 2001).

```
5  <SOAP-ENV:Body>
6    <parameters/>
7  </SOAP-ENV:Body>
```

Figura 2. 3 Estructura del Body en un mensaje SOAP

A continuación se muestra el mensaje SOAP completo:

```
1  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="
     http://www.w3.org/2001/XMLSchema-instance"
2    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3
4    <SOAP-ENV:Body>
5      <parameters/>
6    </SOAP-ENV:Body>
7
8  </SOAP-ENV:Envelope>
```

Figura 2. 4 Estructura general de un mensaje SOAP

Un mensaje SOAP es una transmisión de una vía desde un emisor SOAP a un receptor SOAP, y cualquier aplicación puede participar en este intercambio como emisor o receptor. Los mensajes SOAP se pueden combinar para soportar muchos comportamientos de comunicación, incluyendo, solicitud/respuesta, respuesta solicitada, mensajería asíncrona de una vía, o incluso notificación. SOAP es un protocolo de alto nivel que sólo define la estructura del mensaje y unas pocas reglas para su procesamiento.

Según (Asúa, abril 2012) REST es ligero, con poca configuración, se lee fácilmente (son URLs) y no hace falta nada especial para implementarlo mientras que el SOAP, mucho más ambicioso, es fácil de consumir y tiene un tipado fuerte (WSDL).

WSDL (Web Services Description Language) es un documento XML que contiene un conjunto de definiciones que describen un Servicio Web. Proporciona toda la información necesaria para acceder y utilizar un Servicio Web. Un documento WSDL describe qué hace el Servicio Web, cómo se comunica, y dónde reside. El documento WSDL es utilizado en el momento del despliegue para crear las interfaces de servicio.

2.2.4 Construcción de servicios web con Drools

Un Servicio Web es un componente software con las siguientes características:

- Es accesible a través del interface SOAP (Simple Object Access Protocol).
- Su interface se describe en un documento WSDL (Web Service Description Language).

La definición más completa de servicios web es la proporcionada por el consorcio W3C (The Word Wide Web Consortium) que los define como: aplicaciones software identificadas por una URI, cuyos interfaces y enlaces son capaces de ser definidos, descritos y descubiertos como artefactos XML. Soportan directamente interacciones con otros agentes de software usando intercambio de mensajes basados en XML a través de protocolos basados en Internet.

La contribución de los servicios web para resolver las limitaciones del middleware convencional se basa en tres aspectos fundamentales(Fernández and Soto, 2006):

- *Arquitecturas orientadas a servicios:* donde toda la funcionalidad del sistema se expone como un servicio, y los servicios son autónomos e independientes. Esto produce un desacoplamiento de las aplicaciones y hace que sean más modulares.
- *Rediseño de protocolos middleware:* El segundo aspecto importante de los servicios web es el rediseño de los protocolos middleware para trabajar de forma punto a punto entre las compañías, sin intermediarios. Lo que se conseguía hasta ahora con una plataforma centralizada que controlaba todos los procesos tiene que ser rediseñado para conseguirlo de forma descentralizada.
- *Estandarización:* En el problema de la integración de aplicaciones, la estandarización es un punto clave, aunque a veces es muy difícil de conseguir debido a la existencia de sistemas heredados y a que la complejidad y el coste del middleware siguen siendo muy elevados. Esta necesidad de estandarización ha sido reconocida por los principales vendedores de software, por lo que surgen diversos intentos de estandarización por diversas organizaciones y consorcios, como OASIS (Organization for the Advancement of Structured Standards) o W3C (TheWorkflow Management Coalition). Estas organizaciones intentan estandarizar todos los aspectos de la interacción entre aplicaciones, desde la definición de lenguajes hasta el formato de los mensajes y los protocolos de interacción. A veces incluso compiten más de una especificación para cada aspecto de la interacción.

La filosofía a usar para crear servicios web en Drools es a través de los servicios web que nos brinda Java. En la Figura 2.5 se muestra la arquitectura general de un proyecto web utilizando Drools como motor de reglas de negocio.

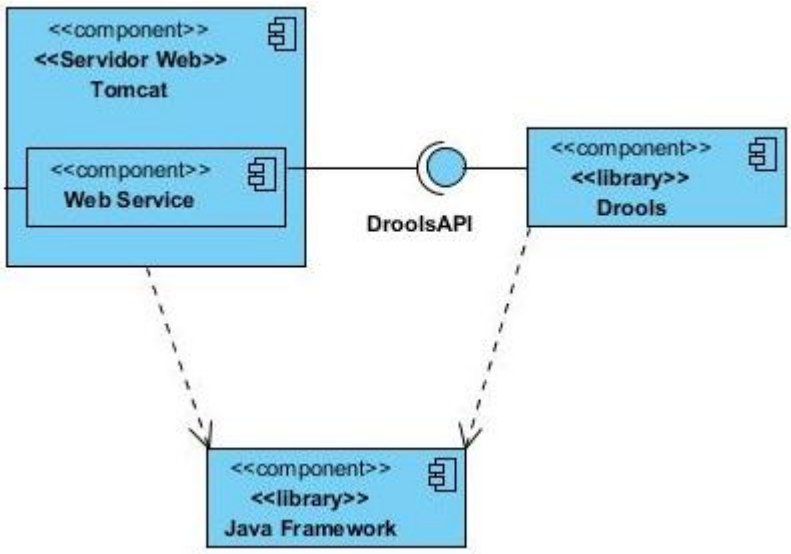


Figura 2. 5 Arquitectura de un proyecto web utilizando Drools

El primer paso a realizar antes de crear el proyecto en NetBeans es configurar nuestro drools-runtime en Eclipse, que no es más que el conjunto de librerías que se necesitan para poder trabajar con Drools, ver anexo 1. Realizado este paso podemos crear nuestro proyecto en NetBeans donde programaremos los servicios webs que llaman a Drools, ver anexo 2.

Se presenta el ejemplo **helloworld** con el objetivo de hacer entender cómo se establece la conexión entre un proyecto web en java y la unión del lenguaje Drools. La estructura del proyecto se muestra en la Figura 2.6.

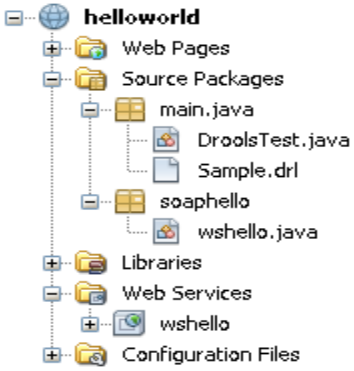


Figura 2. 6 Estructura del proyecto helloworld

En la Tabla 1 se muestra la función de cada archivo del proyecto.

Tabla 1 *Archivos del proyecto helloworld y su respectiva función*

Archivo	Función
DroolsTest.java	Inicializa el motor de reglas con el fichero en el cual se han escrito las reglas del negocio, en este caso Sample.drl. Luego se llama al motor de reglas para que ejecute las mismas.
Sample.drl	Contiene las reglas de negocio.
wshello.java	Contiene el código del servicio web que recibe la llamada del cliente e invoca al BRMS.

Pasos a seguir para codificar el archivo **DroolsTest.java**:

1. Se importan las bibliotecas que se muestran en la Figura 2.7

```
 7  import org.drools.KnowledgeBase;
 8  import org.drools.KnowledgeBaseFactory;
 9  import org.drools.builder.*;
10  import org.drools.io.ResourceFactory;
11  import org.drools.logger.KnowledgeRuntimeLogger;
12  import org.drools.logger.KnowledgeRuntimeLoggerFactory;
13  import org.drools.runtime.StatefulKnowledgeSession;
```

Figura 2. 7 Bibliotecas a importar en el proyecto de NetBeans

2. Se define el método **readKnowledgeBase** para generar memorias de trabajo en base al fichero de reglas, el objeto resultante es un **KnowledgeBase.** En la Figura 2.8 se puede observar un fragmento de este método, mientras que en la Tabla 2 se explica más detallado el método.

```
43  private static KnowledgeBase readKnowledgeBase() throws Exception {
44      KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
45      kbuilder.add(ResourceFactory.newClassPathResource("Sample.drl"),
46              ResourceType.DRL);
47      KnowledgeBuilderErrors errors = kbuilder.getErrors();
48       if (errors.size() > 0) {
49        for (KnowledgeBuilderError error: errors) {
50         System.err.println(error);
51        }
52        throw new IllegalArgumentException("Could not parse knowledge.");
53       }
54        KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
55        kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
56        return kbase;
57  }
```

Figura 2. 7 Inicialización de la memoria de trabajo y asignación del fichero de reglas

Tabla 2 *Descripción del método readKnowledgeBase*

Línea	Descripción
44-45	Se generan memorias de trabajo en base al fichero de reglas Sample.drl
46-52	Se tratan posibles errores.
53-56	Se crea un objeto KnowledgeBase que contiene las reglas definidas en Sample.drl

3. En el mismo archivo DroolsTest definimos la clase Message y sus métodos de acceso los cuales son utilizados tanto en el método mensajehello como en el fichero que contiene las reglas para guardar y acceder al mensaje que se envía. Esta clase se muestra en la Figura 2.9.

```
63    public static class Message {
64      public static final int HELLO = 0;
65      public static final int GOODBYE = 1;
66      private String message;
67      private int status;
68
69      public String getMessage() {
70         return this.message;
71      }
72      public void setMessage(String message) {
73         this.message = message;
74      }
75      public int getStatus() {
76         return this.status;
77      }
78      public void setStatus(int status) {
79         this.status = status;
80      }
81     }
82    }
```

Figura 2. 8 Fragmento de código de la clase Message

4. En la Figura 2.10 se muestra el código del método **mensajehello** donde se inicia la ejecución del motor de reglas de negocio**.** La descripción de este método se puede observar en la Tabla 3.

```
20  public int mensajehello(String msg, int i) {
21    try {
22      // load up the knowledge base
23      KnowledgeBase kbase = readKnowledgeBase();
24      StatefulKnowledgeSession ksession =
25              kbase.newStatefulKnowledgeSession();
26      KnowledgeRuntimeLogger logger =
27             KnowledgeRuntimeLoggerFactory.newFileLogger(ksession, "test");
28      // go !
29      Message mensaje = new Message();
30      mensaje.setMessage(msg);
31      mensaje.setStatus(i);
32      ksession.insert(mensaje);
33      System.out.println("Mensaje Running !!!!!!!");
34      ksession.fireAllRules();
35      logger.close();
36      return i;
37       } catch (Throwable t) {
          t.printStackTrace();
39        return 0;
40    }
```

Figura 2. 9 Inserción de los pedidos en la memoria de trabajo y ejecución de las reglas

Tabla 3 *Descripción del método mensajehello*

Línea	**Descripción**
23	Se realiza una llamada al método readKnowledgeBase declarado anteriormente.
24	Creamos una sesión para crear una memoria de trabajo con estado**.**
27	Se crea un objeto de la clase Message y se le pasan los valores definidos en nuestro método.
30	Se insertan los pedidos en la memoria de trabajo invocando al método insert de StatefulKnowledgeSession.
33	Finalmente se ejecutan las reglas invocando a fireAllRules que actuará sobre los objetos que se insertaron anteriormente.
37-39	Manejo de la excepción.

Las reglas del negocio contenidas en el archivo **Sample.drl** se muestran en la Figura 2.11.

```
 5 rule "Hello World"
 6     when
 7         m : Message( status == Message.HELLO, myMessage : message )
 8     then
 9         System.out.println( myMessage );
10         m.setMessage( "Goodbye cruel world" );
11         m.setStatus( Message.GOODBYE );
12         update( m );
13 end
14
15 rule "GoodBye"
16     when
17         Message( status == Message.GOODBYE, myMessage : message )
18     then
19         System.out.println( myMessage );
20 end
```

Figura 2. 10 Fragmento de código de Sample.drl

Por último explicaremos la codificación del archivo **wshello.java**. En la Figura 2.12 se puede observar el fragmento de código del servicio web y en la Tabla 4 se describe con más detalle este método.

```
22      @WebMethod(operationName = "mensajehello")
23      public int mensajehello(@WebParam(name = "msg") String msg,
24              @WebParam(name = "i") int i) {
25          DroolsTest drools = new DroolsTest();
26          int request = drools.mensajehello(msg, i);
27          return request;
28
29      }
```

Figura 2. 11 Creación del servicio web mensajehello

Tabla 4 Descripción del servicio web mensajehello

Línea	**Descripción**
24	Se crea un objeto de tipo DroolsTest.
25	Adicionar al método mensajehello los parámetros que se definen al inicializar el método.
26	Se devuelve la respuesta del webservice.

Los servicios webs pueden ser probados con herramientas como XMLSpy y SOAPUI. XMLSpy es un avanzado editor XML para modelar, editar, transformar y depurar tecnologías XML relacionadas con los servicios web como WSDL y SOAP. Este editor ofrece la posibilidad de crear las aplicaciones XML y aplicaciones web más avanzadas, permitiendo al mismo tiempo la flexibilidad necesaria para trabajar con cualquier tecnología XML de la forma que mejor se ajuste a la complejidad del documento. El soapUI es una de las herramientas más populares para llamar Web Services por su disponibilidad en múltiples plataformas y su integración con el IDE Eclipse (Amador, January 2012).

Destacar que mediante el servicio web que se mostró en la Figura 2.12 una aplicación cliente puede usar las reglas que están almacenadas en el repositorio del BRMS. El cliente puede ser otro servicio web, una aplicación desktop, una página php u otra. En el próximo capítulo se describe con más énfasis lo planteado anteriormente.

2.2.5 Plug-in web y de Eclipse para la edición de reglas de negocio

Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Ejemplo de ello es el GEF (Graphic Editing Framework - Framework para la edición gráfica) que no es más que un plug-in de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto wysiwyg hasta editores de diagramas UML, interfaces gráficas para el usuario (GUI), etc. Los editores realizados con GEF "viven" dentro de Eclipse, además de poder ser usados conjuntamente con otros plug-in, hacen uso de su interfaz gráfica personalizable y profesional (Amador, January 2012).

Drools presenta un plug-in para Eclipse (**org.drools.updatesite**), con la facilidad de que si se está desarrollando un proyecto en Java bajo el Eclipse sea más fácil la redacción e incorporación de las reglas al proyecto. Con este plug-in podemos ver coloreadas las keywords (lenguaje de DRL) y nos avisa de posibles errores de sintaxis, entre otras cosas. El plug-in proporciona a diseñadores (y los usuarios muy técnicos) un ambiente para revisar y probar las reglas en varios formatos, y lo integra profundamente con sus aplicaciones(Amador, January 2012).

2.3 Conclusiones Parciales

En este capítulo se analizaron las diferentes herramientas (propietarias y no propietarias) que existen en el mercado para la edición de reglas de negocio. Comparándose programas que tienen todos los módulos de un BRMS; Drools fue seleccionado como el software de código abierto con las mejores propiedades. Se describe su arquitectura, lenguaje de definición, las interfaces para interactuar con otras aplicaciones y la definición de servicios web. El siguiente capítulo está dirigido a validar la integración del BRMS Drools al flujo de trabajo del control de historias clínicas para el trasplante renal.

INTEGRACIÓN DE UN BRMS AL FLUJO DE TRABAJO "CONTROL DE HISTORIAS CLÍNICAS PARA TRASPLANTE RENAL"

CAPÍTULO 3. INTEGRACIÓN DE UN BRMS AL FLUJO DE TRABAJO "CONTROL DE HISTORIAS CLÍNICAS PARA TRASPLANTE RENAL"

En este capítulo se describe el diseño e implementación de una solución informática que muestra cómo integrar un BRMS al flujo de trabajo "control de historias clínicas para trasplante renal" en el Hospital Universitario "Arnaldo Milián Castro". Se muestra la arquitectura general de un proyecto web que utiliza el motor de reglas Drools y las herramientas que se utilizan para el desarrollo del sistema.

3.1 Caso de estudio: control de historias clínicas para trasplante renal

El caso de estudio elegido para mostrar la validación de un BRMS al flujo de trabajo de una empresa está relacionado al control de las historias clínicas de pacientes en el área de nefrología, los que son atendidos en las consultas de progresión, hemodiálisis y trasplante renal (Martínez Busto et al., 2009). En este epígrafe se hace una descripción sobre el caso de estudio.

El sistema se inicia cuando un paciente asiste a la consulta de nefrología en su área de atención médica. En esta consulta el paciente es atendido por un especialista, que inserta los datos del paciente en el sistema y verifica su estado de padecimiento. Si el paciente

presenta insuficiencia renal crónica o terminal es remitido a la consulta de progresión, en caso contrario debe ser atendido en la propia consulta de nefrología.

Al asistir un paciente a la consulta de progresión se le solicitan datos generales, tales como: nombre, ci, edad, entre otros. En esta consulta se calcula el estadio en el que se encuentra el paciente dependiendo de resultados de determinados exámenes complementarios. Si el paciente se encuentra en el estadio 1 ó en el estadio 2 es enviado nuevamente a la consulta de su área de salud, si se encuentra en estadio 3 ó 4 es remitido a la misma consulta de progresión para un mejor seguimiento acorde al estado de su salud, si se encuentra en el estadio 5 es remitido a la consulta de hemodiálisis hasta tanto no exista un donante en la base de datos del hospital que satisfaga todas las condiciones necesarias para realizar el trasplante, acorde al protocolo.

En la consulta de Trasplante se realizan varias operaciones, entre ellas se encuentran: insertar un nuevo donante, insertar un nuevo receptor para lo que se considera la restricción que solo puede tener estadio 4 ó 5, entre otras. De los donantes es necesario conocer sus datos generales, tales como: nombre, apellidos, ci y edad. De los receptores es necesario conocer también sus datos generales además de los resultados de diversos exámenes complementarios, entre ellos se encuentran: examen de VIH, examen de hepatitis B y examen de hepatitis C y otras enfermedades que presenta.

Tanto el receptor como el donante potencial deben cumplir requisitos determinados dentro del protocolo de trasplante. Algunas de las reglas de negocio identificadas para el caso de estudio son las siguientes.

Reglas para el receptor:

R1. Un TR debe realizarse cuando la edad del paciente se encuentra entre 15 y 55 años.

R2. Un TR debe realizarse cuando el VIH del paciente fue negativo.

R3. Un TR debe realizarse cuando el Hepatitis B del paciente fue negativo.

R4. Un TR debe realizarse cuando el Hepatitis C del paciente fue negativo.

R5. Un TR debe realizarse cuando el donante no presenta ninguna de las siguientes enfermedades: Corazón, Pulmón, Hígado.

Reglas para el donante:

R1. Un TR debe realizarse cuando la edad del donante se encuentra entre 18 y 55 años.

R2. Un TR debe realizarse cuando el VIH del donante fue negativo.

R3. Un TR debe realizarse cuando el Hepatitis B del donante fue negativo.

R4. Un TR debe realizarse cuando el Hepatitis C del donante fue negativo.

R5. Un TR no debe realizarse cuando el donante padece de diabetes mellitus o hipertensión arterial.

Sobre este modelo se quiere definir un servicio de decisión que permita determinar si un paciente receptor es compatible con un donante para recibir el trasplante renal. Para darle solución al ejemplo planteado anteriormente se utiliza el BRMS Drools como motor de reglas. El BRMS Drools determina qué reglas necesitan ser corridas y en qué orden, además proporcionan medios para dirigirse a la resolución del conflicto. Las reglas son guardadas en la forma humano-leíble en un archivo, para qué puedan cambiarse con editor de texto o editor de la regla. Con la validación de este ejemplo se pretende mostrar cómo es la integración de un BRMS al flujo de trabajo de una empresa.

3.1.1 Diagrama de casos de uso

A continuación se realiza el análisis y diseño del caso de uso para este ejemplo y su respectiva descripción. Se establece dos usuarios: *Paciente y EquipoMédico*. Se identifican cuatro procesos: *Asistir al Paciente en Consulta de Nefrología, Asistir al Paciente en Consulta de Progresión, Analizar Pre-Requisitos para Métodos Sustitutivos* y *Aplicar Métodos Sustitutivos*; cada uno de ellos tiene roles perfectamente definidos (véase la Figura 3.1). La descripción de cada elemento para este caso de uso puede verse en la Tabla 5

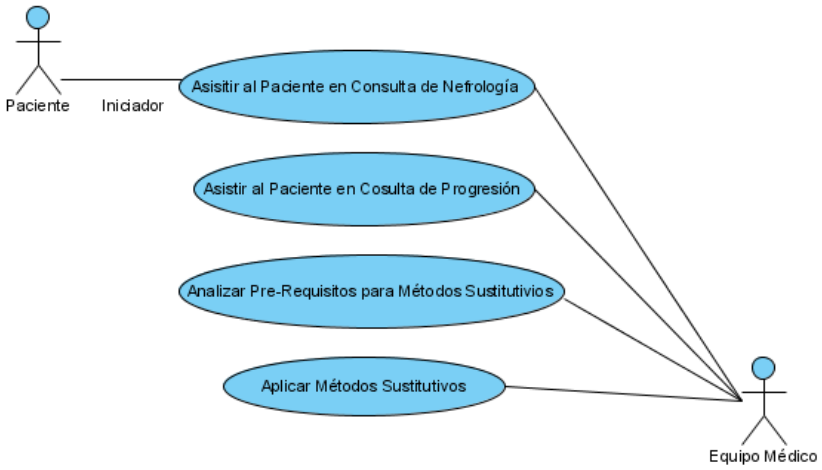


Figura 3. 1 Diagrama de casos de uso *(Martínez Busto et al., 2009)*

Tabla 5 Descripción del diagrama de casos de uso para el ejemplo

Caso de uso	Descripción
Asistir el Paciente en Consulta de Nefrología	Se añade un nuevo paciente a la consulta de Nefrología, si su insuficiencia renal es crónica o terminal el paciente es remitido a la consulta de progresión.
Asistir al paciente en Consulta de Progresión	El paciente llega a la consulta de progresión remitido por la consulta de nefrología. En esta consulta se inserta un nuevo paciente al sistema. Según sea el estadío el paciente es insertado en la consulta de Hemodiálisis o en la consulta de Trasplante.
Analizar Pre-requisitos para Métodos Sustitutivos	Se analizan los requisitos que debe cumplir un paciente para recibir un trasplante renal. Si cumple con las reglas del sistema se inserta un nuevo paciente en la consulta de Trasplante.
Aplicar Métodos Sustitutivos	En caso que el paciente necesite ser tratado con alguno de los métodos sustitutivos (*diálisis*, *hemodiálisis* o *TR*), pasa a ser atendido en las consultas propias del método de sustitución, o sea *hemodiálisis* y *trasplante*.

El sistema implementa funcionalidades básicas de los casos de uso Asistir el Paciente en Consulta de Nefrología, Asistir al paciente en Consulta de Progresión, Analizar Pre-requisitos para Métodos Sustitutivos y Aplicar Métodos Sustitutivos. El caso de uso Analizar Pre-requisitos para Métodos Sustitutivos es implementado haciendo uso del BRMS Drools para validar las reglas que debe cumplir el caso de estudio y es donde nuestro trabajo tiene su mayor contribución.

3.1.2 Diagrama de clases

El diagrama representa un subconjunto de clases del caso de estudio relacionado con el subdominio de trasplante renal. (Ver Figura 3.2).

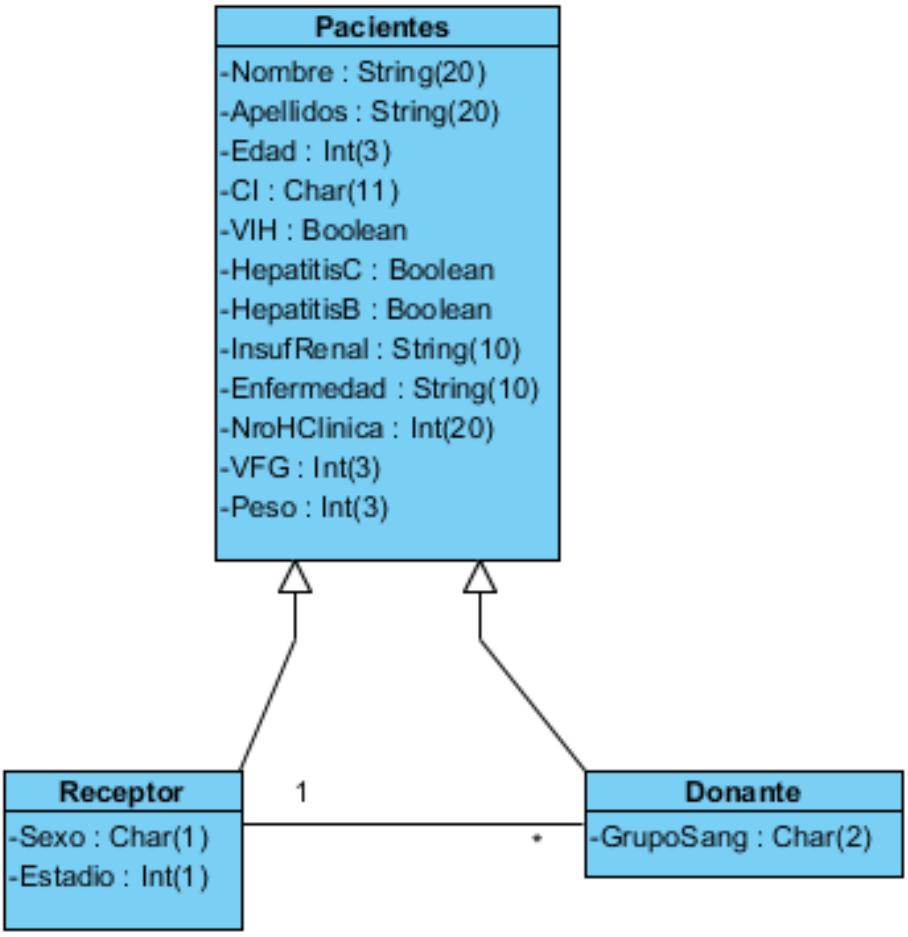


Figura 3. 2 Subconjunto del diagrama de clases para el caso de estudio

3.1.3 Diagrama de flujo de trabajo

Para automatizar la decisión de realizar un trasplante renal o no se diseñó el siguiente flujo de trabajo. Las reglas de negocio son ejecutadas en un BRMS, que valida los datos del posible receptor y donante (véase la Figura 3.3).

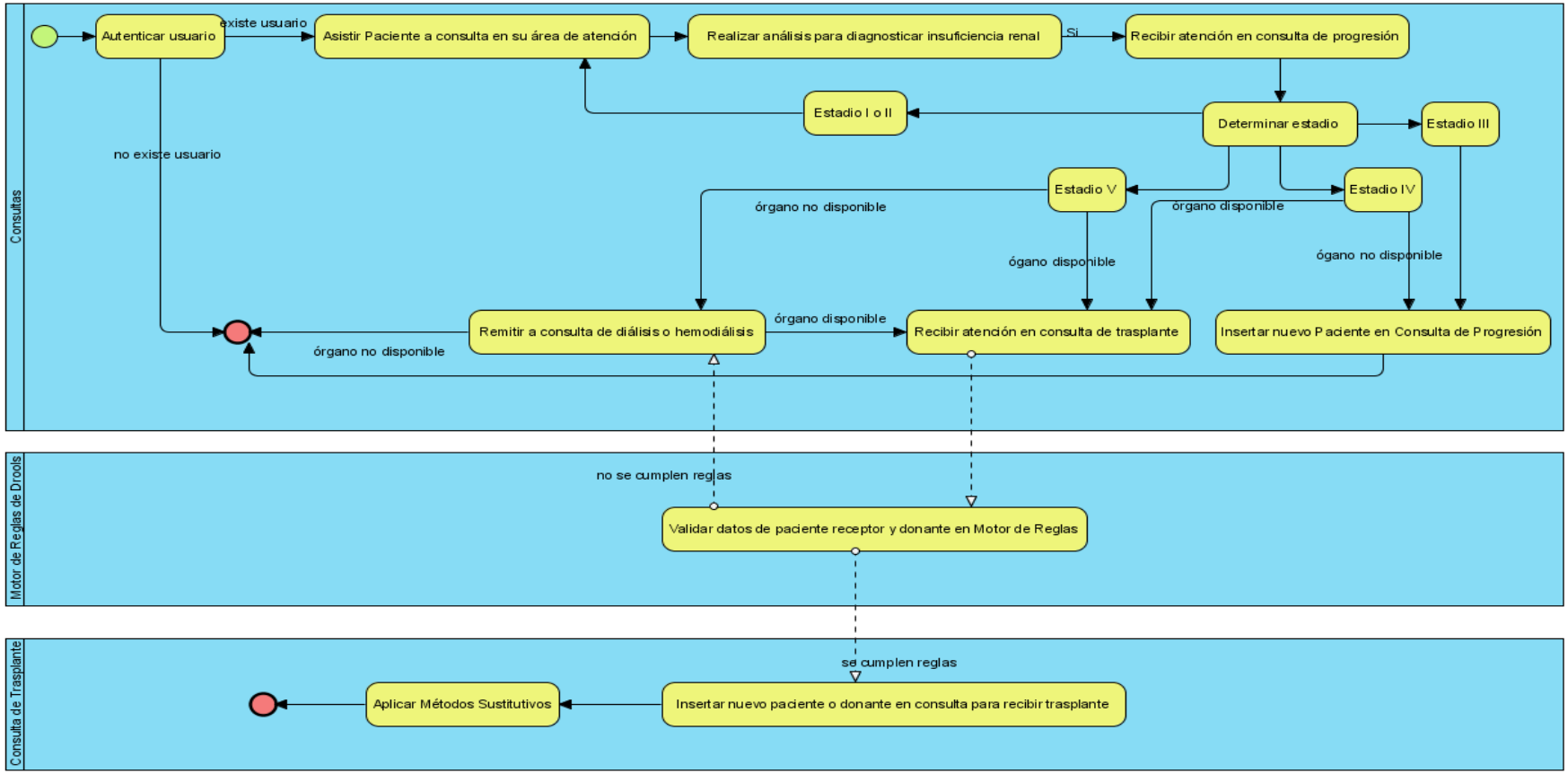


Figura 3.3 Diagrama de flujo de trabajo para el caso de estudio "control de historias clínicas para trasplante renal"

3.1.4 Diagrama de arquitectura del sistema

La arquitectura propuesta integra las siguientes tecnologías:

- Java como lenguaje de programación y para la interfaz del sistema. Es un reconocido lenguaje orientado a objeto y sobre todo un lenguaje muy útil para la programación de aplicaciones Web.
- Drools como el motor de reglas. Permite expresar de forma más natural las reglas del negocio interactuando con los objetos de negocio para ello utiliza el lenguaje de reglas de Drools (DRL) a través del cual especifica las condiciones, acciones y funciones de las mismas, las cuales se pueden expresar en el lenguaje JAVA.
- Apache Tomcat como servidor web con soporte para servlets y JSPs. Tomcat es usado como servidor Web independiente en entornos con alto nivel de tráfico y alta disponibilidad. El hecho de que Tomcat fue escrito en Java, hace posible que funcione en cualquier sistema operativo que disponga de la máquina virtual Java (también se puede utilizar con XAMPP).
- El IDE propuesto para desarrollar el sistema es *NetBean*s, siendo una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas (Sánchez, 2003). El IDE NetBeans soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles), provee soporte para la creación de aplicaciones orientadas a servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL para servicios web, así como permite crear aplicaciones Web con PHP 5. *Eclipse* es el IDE que se propone para editar, crear y depurar las reglas del negocio facilitando el desarrollo con la herramienta Drools, además de ser un entorno de código abierto que admite numerosas extensiones (incluido un módulo para J2EE) y posibilidades.
- XAMPP**:** es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. El servicio que se montó en el servidor web fue el de administrar una base de datos creada sobre phpMyAdmin a través de páginas Web en PHP.

En la Figura 3.4 se muestra la integración de estas tecnologías para dar solución al caso de estudio propuesto.

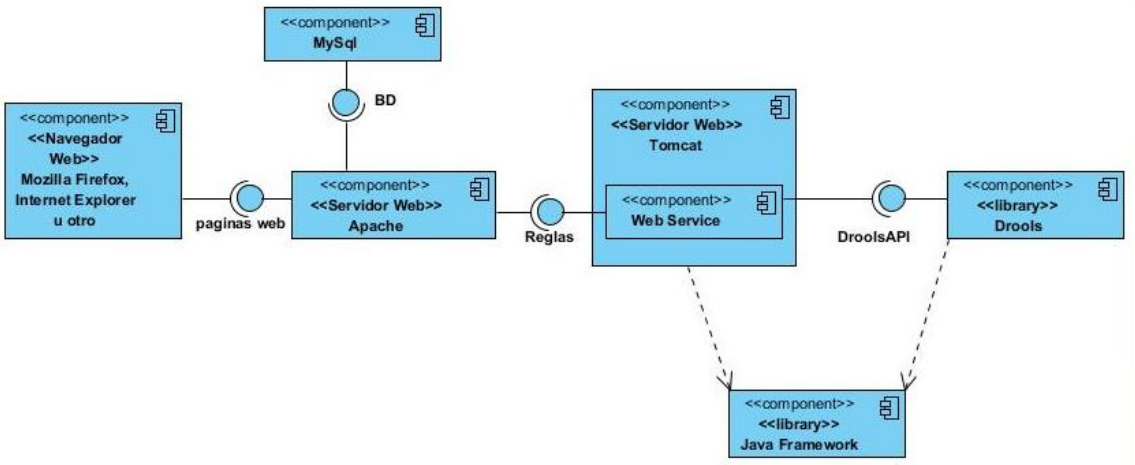


Figura 3. 4 Arquitectura general de un proyecto web con Drools

A continuación describimos la relación entre los distintos componentes de la arquitectura anterior en el caso de uso "Analizar Pre-requisitos para Métodos Sustitutivos" que valida si una persona puede ser un donante o no. Inicialmente un miembro del equipo médico, a través de un navegador web, solicita insertar un nuevo donante en el sistema. Esta solicitud es procesada en el servidor web Apache el cual llama un servicio Web, que corre sobre Tomcat y ejecuta reglas en Drools que validan si la persona puede ser donante o no. En caso afirmativo el donante es insertado en la Base de datos de MySQL.

La aplicación cliente se realizó en la versión 7.1del NetBeans y se utilizó como lenguaje de programación PHP y HTML. El servicio Web se programó en Java. Las reglas de Drools se programaron en DRLy Java.

Destacar que todas estas tecnologías son multiplataforma, es decir el sistema funcionará bajo diferentes sistemas operativos, como Windows o Linux. Otra ventaja es que las herramientas son todas de código abierto y de licencia que no impone restricciones sobre su uso y del software derivado.

3.2 Detalles de la implementación del caso de estudio

En este epígrafe se mostrará la estructura de nuestro servicio web el cual está conformado por los métodos verificardonante y verificareceptor que tienen como función obtener los datos de las personas para verificar si cumplen el conjunto de reglas que el sistema impone para la obtención de un trasplante renal. En el fichero Sample.drl se encuentran todas la reglas que se deben validar en el sistema, este fichero es escrito en lenguaje Drools.

3.2.1 Archivo Sample.drl

El proyecto web *TrasplanteRenal* contiene el fichero Sample.drl, el cual contiene todas las reglas que se desean sean validadas para la decisión de realizar o no un trasplante renal. El fichero .drl contiene diez reglas del negocio de ellas cinco son para los pacientes receptores del órgano renal y las restantes son para los posibles donantes insertados en el sistema.

Para recibir el trasplante renal un paciente debe cumplir reglas relacionadas con su edad, si padece alguna enfermedad, y los resultados del vih, hepatitis b y hepatitis c.

```
17 rule "Enfermedad del receptor "
18   when
19    b: Pacientes( b.getEnfermedad() == "Corazon"
20                || b.getEnfermedad() == "Pulmon"
21                || b.getEnfermedad() == "Higado")
22   then
23    b.setBe_PacReceptor(1);
24    System.out.println("El paciente padece de alguna enfermedad");
25    System.out.println("******************************");
26 end
27
28 rule "VIH del receptor"
29   when
30    b: Pacientes( b.getVIH() == "Postivo")
31   then
32    b.setBe_PacReceptor(1);
33    System.out.println("No debe realizarse trasplante renal. VIH Positivo");
34    System.out.println("******************************");
35 end
```

Figura 3. 5 Subconjunto de las reglas que debe cumplir un paciente para recibir trasplante

Los donantes deben cumplir un conjunto de reglas relacionadas con la edad, el estado de insuficiencia renal, los resultados de hepatitis b, hepatitis c y vih, así como si presenta algún tipo de enfermedad relacionada con diabetes mellitus o hipertensión arterial.

```
55 rule "Verificar edad del donante"
56    when
57     b: Donante(b.getEdad() < 18 || b.getEdad() > 55)
58    then
59     b.setBe_donante(1);
60     System.out.println("La edad del donante debe estar entre 18 y 55 anos");
61     System.out.println("******************************");
62 end
63
64 rule "VIH donante"
65    when
66      b: Donante( b.getVIH() == "Postivo" )
67    then
68      b.setBe_donante(1);
69      System.out.println("No debe realizarse transplante renal. VIH positivo");
70      System.out.println("******************************");
71 end
```

Figura 3. 6 Subconjunto de reglas que debe cumplir un posible donante para realizar el trasplante renal

3.2.2 Servicio web

Los servicios web nos sirven para poder utilizar datos desde otras plataformas, ya que distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Nuestro sistema gestiona posibles trasplantes renales, por lo que programamos un servicio web con el objetivo de poder compartir los datos. El servicio web contiene dos métodos que son los encargados de verificar, llamando al motor de reglas de Drools, si un paciente receptor o un posible donante cumplen con las reglas descritas en el epígrafe anterior.

En la Figura 3.7 se observa el método que validad un paciente receptor, del cual solo nos interesa verificar su edad, si presenta alguna enfermedad, y los resultados de las pruebas de vih, hepatitis b y hepatitis c.

```
49      @WebMethod(operationName = "verificarreceptor")
50      public Integer verificarreceptor(@WebParam(name = "edad") int edad,
51      @WebParam(name = "vih") String vih,
52      @WebParam(name = "hepatitisb") String hepatitisb,
53      @WebParam(name = "hepatitisc") String hepatitisc,
54      @WebParam(name = "enfermedad") String enfermedad) {
55          //TODO write your implementation code here:
56          DroolsTest drools = new DroolsTest();
57          int request = drools.verificarreceptor(edad, vih, hepatitisb,
58                  hepatitisc, enfermedad);
59          return request;
60      }
```

Figura 3. 7 Servicio web para validar un paciente receptor

En la Figura 3.8 se describe el método que valida si una persona puede ser donante. Para lo cual se valida la edad, enfermedades que presenta, estado de insuficiencia renal y los resultados de las pruebas de vih, hepatitis b y hepatitis c.

```
35      @WebMethod(operationName = "verificardonante")
36      public Integer verificardonante(@WebParam(name = "edad") int edad,
37      @WebParam(name = "vih") String vih,
38      @WebParam(name = "hepatitisb") String hepatitisb,
39      @WebParam(name = "hepatitisc") String hepatitisc,
40      @WebParam(name = "enfermedad") String enfermedad,
41      @WebParam(name = "insufrenal") String insufrenal) {
42          //TODO write your implementation code here:
43          DroolsTest drools = new DroolsTest();
44          int request = drools.verificardonante(edad, vih, hepatitisb, hepatitisc,
45                  enfermedad, insufrenal);
46          return request;
47      }
```

Figura 3. 8 Servicio web para validar un posible donante

3.3 Conclusiones Parciales

El capítulo estuvo dirigido a la validación de un ejemplo que demuestra la integración de un BRMS al flujo de trabajo de una empresa. Se realizó la descripción del caso de estudio control de historias clínicas para trasplante renal y modelación del diagrama de casos de uso, de clases, de flujo de trabajo y arquitectura del sistema. Se mostraron detalles de la implementación del mismo como los archivos de reglas y servicios Webs que son llamados por aplicaciones clientes para procesar las reglas en el BRMS.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El desarrollo de sistemas informáticos siguiendo un enfoque de reglas de negocio permite que no se reprogramen nuevamente módulos enteros para cambiar las reglas de negocio, estas pueden ser aisladas del resto de la aplicación y así ser modificadas sin necesidad de recompilar la misma. A manera de conclusión se enuncia que:

- Se identificaron los componentes, repositorio de reglas y motor de reglas, necesarios para lograr la integración del BRMS al flujo de trabajo de una empresa. El BRMS seleccionado fue Drools por ser un motor de reglas con arquitectura modular, que publica sus funcionalidades mediante servicios web e incluye módulos que proporcionan capacidades de flujo de trabajo, procesamiento de eventos y planificación de recursos.
- Se realizó el diseño de la arquitectura de un proyecto web que tiene como componentes los servidores web Apache y Apache Tomcat, MySQL como gestor de Base de Datos, Java como framework de desarrollo y Drools como motor de reglas de negocio, el cual es llamado a través de servicios web, permitiendo su acceso desde otras plataformas.
- Se validó mediante el caso de estudio control de historias clínicas para trasplante renal como integrar un BRMS al flujo de trabajo del hospital Arnaldo Milián Castro. En el BRMS Drools se procesaron las reglas que verifican si una persona puede ser donante y si un paciente puede recibir un trasplante renal.

Recomendaciones

- Profundizar en el estudio de los demás módulos de Drools.
- Analizar la inclusión de una arquitectura SOA específica para la gestión de reglas de negocio.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

2006. Manual de Usuario. Administración del Flujo de Trabajo.

2013. Introducción al concepto de EIA. Available: http://es.kioskea.netcontentsentrepriseeai.php3

AMADOR, L. January 2012. *Drools Developer's Cookbook*, Livery Place 35 Livery Street Birmingham B3 2PB, UK, Packt Publishing.

ASÚA, E. D. abril 2012. APIs... seguimos con el REST vs SOAP. Available from: http://blog.movilforum.com/apis-seguimos-con-el-rest-vs-soap/ 18-12-2012].

BALI, M. 2009. *Drools JBoss Rules 5.0 Developer´s Guide*, 32 Lincoln Road, Olton, Birmingham, Published by Packt Publishing Ltd.

BASALLO, Y. A., ESTRADA, A. D. & GÓMEZ, S. G. 2010. Una experiencia en integración de aplicaciones empresariales.

BROWNE, P. 2009. *JBoss Drools Business Rules*, Packt Publishing.

CABRERA, M. O. 2012. *Workflow en su empresa* [Online]. Laboratorio de Gestión Electrónica Empresarial. Available: http://www.usmp.edu.pe/publicaciones/boletin/fia/info40/workflow.html [Accessed].

CLAUS IBSEN, J. A. 2011. *Camel in Action*, Manning Publications Co.

FERNÁNDEZ, A. R. D. S. E. C. 2006. Nuevas tendencias en Sistemas de Información: Procesos y Servicios.

FERNÁNDEZ, E. C. & SOTO, A. D. 2006. Nuevas tendencias en Sistemas de Información: Procesos y Servicios.

FERNÁNDEZ, S. Z. 2009. Open Source BRE/BRMS JSR-94 compliat.

GRAHAM, I. 2007. *Business Rules and Service Oriented Architecture: A Pattern Language*.

LIN, F. 2005. Enterprise Application Integration (EAI) Techniques.

MARTÍNEZ BUSTO, M. E., MORENO MONTES DE OCA, I., MACHADO PADILLA, C. A. & GONZÁLEZ GONZÁLEZ, L. Year. De los procesos de negocio a la modelación de reglas de negocio para un caso de estudio. *In:* GUADALAJARA, U.

D., ed. Memorias del IV Encuentro de la Red Iberoamericana de evaluación y Decisión Multicriterio, 9-15 de Noviembre 2009 Zapopan, Jalisco, México.

MOSCATELLI, S. 2001. Interconexión de aplicaciones legadas usando el paradigma de mensajes.

RED HAT, I. 2010. *JBoss Enterprise BRMS Platform 5. Manual del usuario de BRMS.* , Red Hat, Inc.

SÁNCHEZ, J. 2003. Java2 incluye Swing, Theards, Programación en red, JDBC y JSP/Servlets

SLOAN SEAMAN. 2002. *JxBRE - Java Business Rules Engine.* [Online]. Available: http://jxbre.sourceforge.net/ [Accessed 22 November 2012 2012].

WESKE, M. 2007. *Business Process Management. Concepts, Languages, Architectures.*, Springer-Verlag Berlin Heidelberg 2007.

ANEXOS

ANEXOS

Anexo I Pasos para registrar las bibliotecas de Drools en Eclipse

En este anexo se realizará una descripción de los pasos a seguir para crear una carpeta con las bibliotecas de Drools en el IDE Eclipse.

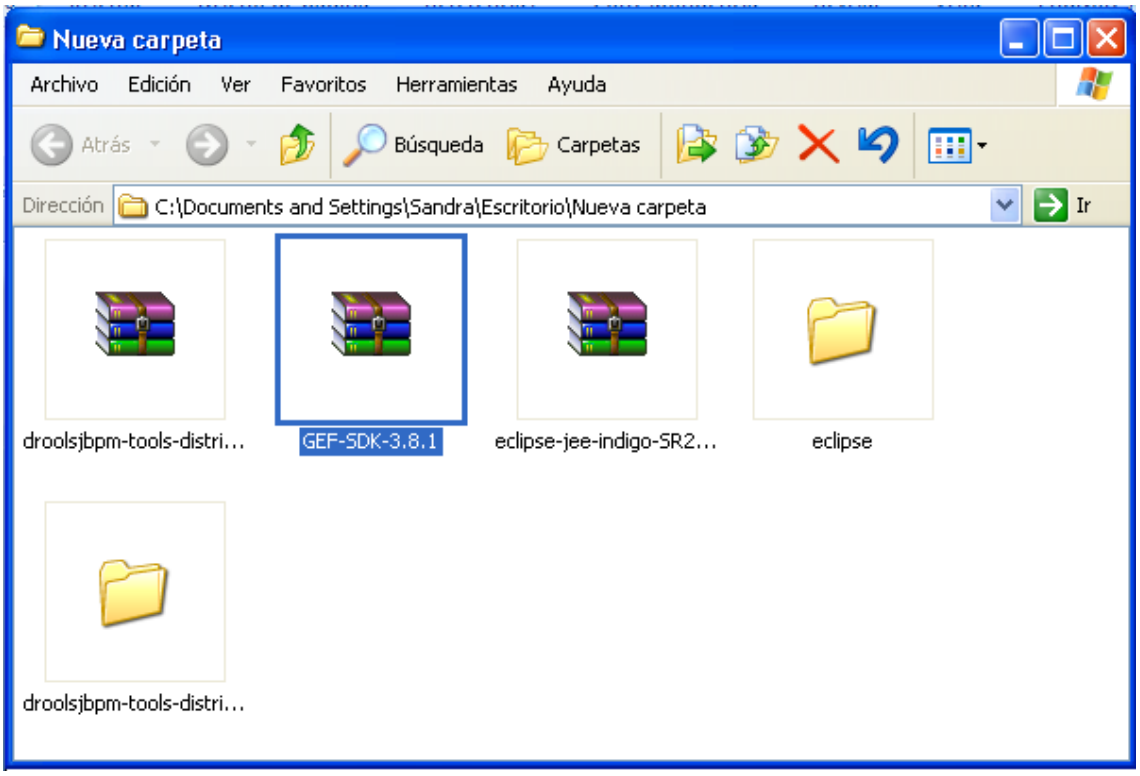
1. Crear una carpeta y copiar para ella los siguientes módulo de Drools:

   droolsjbpm-tools-distribution-5.4.0.Final

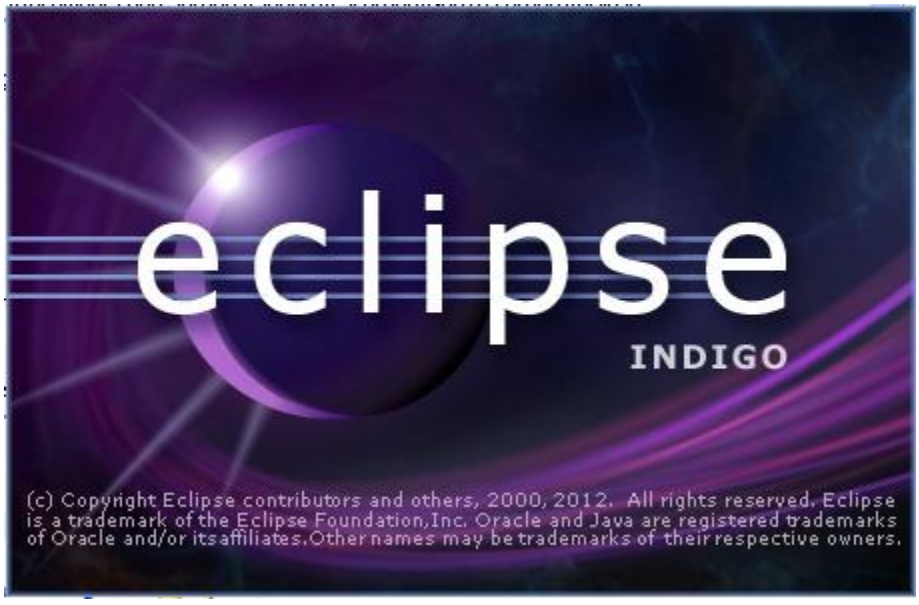
   eclipse-jee-indigo-SR2-win32

   GEF-SDK-3.8.1

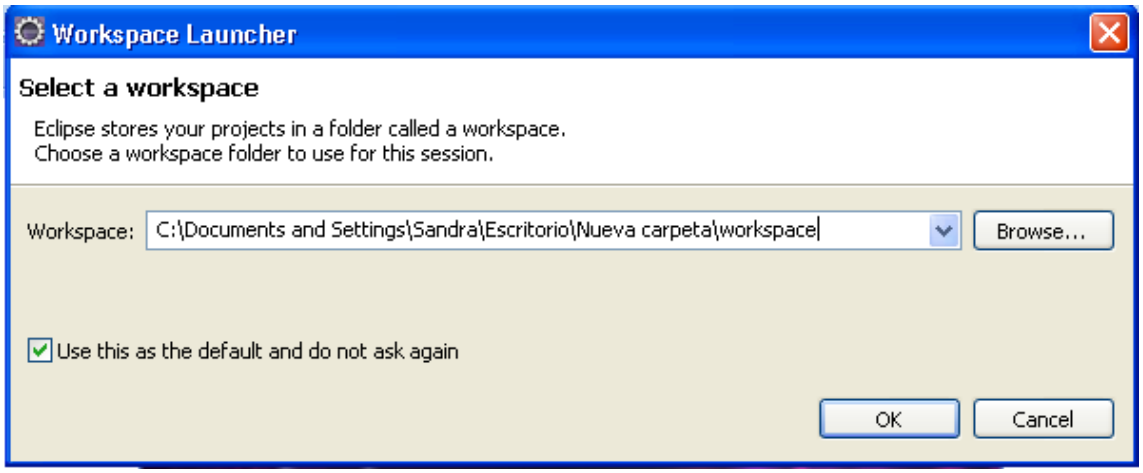
2. Descomprimir los archivos mencionados anteriormente, primero se debe descomprimir eclipse, luego droolsjbpm y por último GEF. Destacar que al descomprimir GEF los archivos que se descomprimen pasan directo a la carpeta de eclipse. Nuestra carpeta debería quedar del siguiente manera:



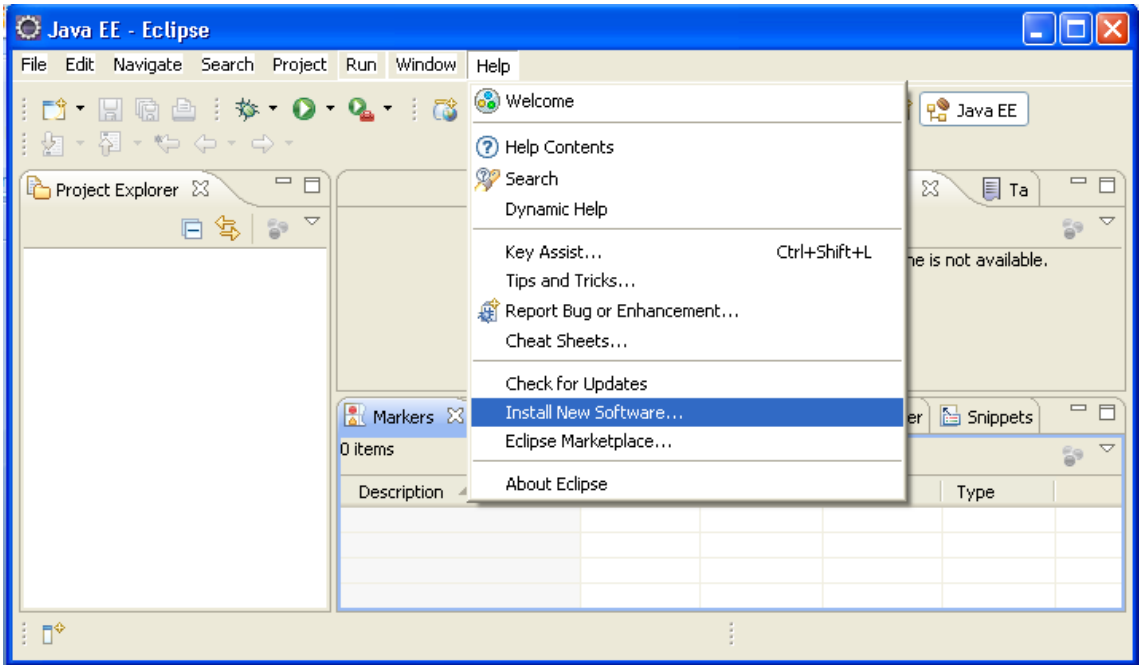
3. Ahora pasamos a ejecutar eclipse.exe, notar que utilizamos la versión INDIGO.



4. Se pasa a dar la dirección de la carpeta donde serán guardados todos nuestros proyectos.



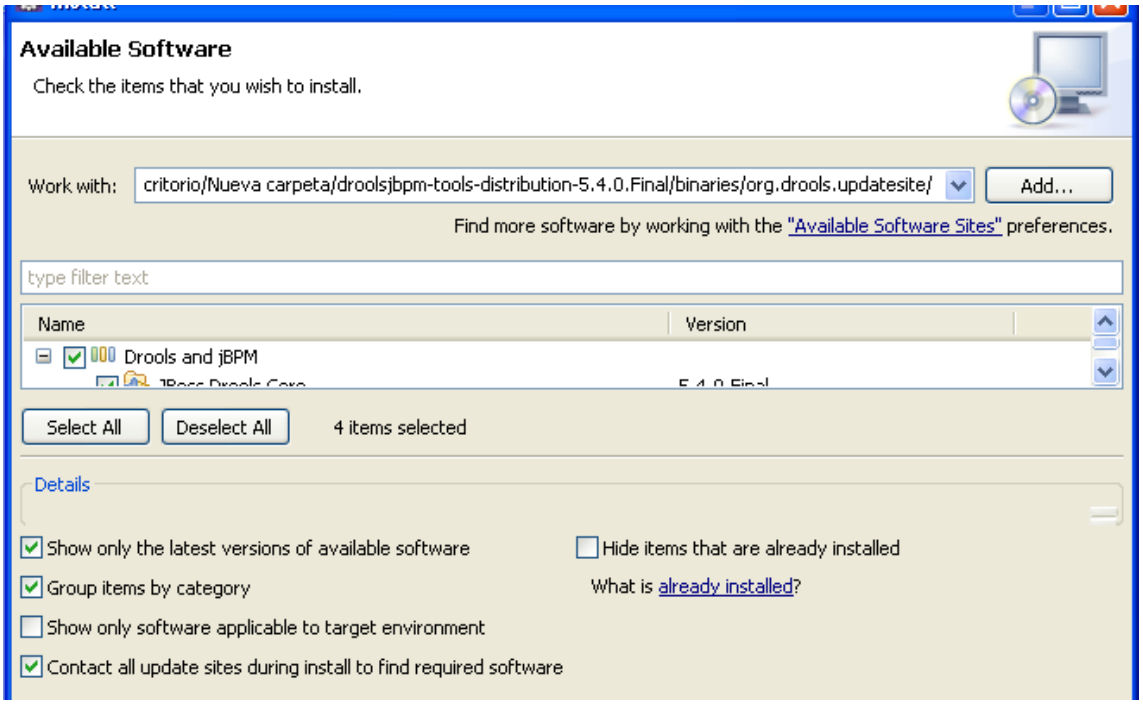
5. Ahora pasamos a cargar las bibliotecas para ello vamos a la pestaña *Help/Install New Software.*



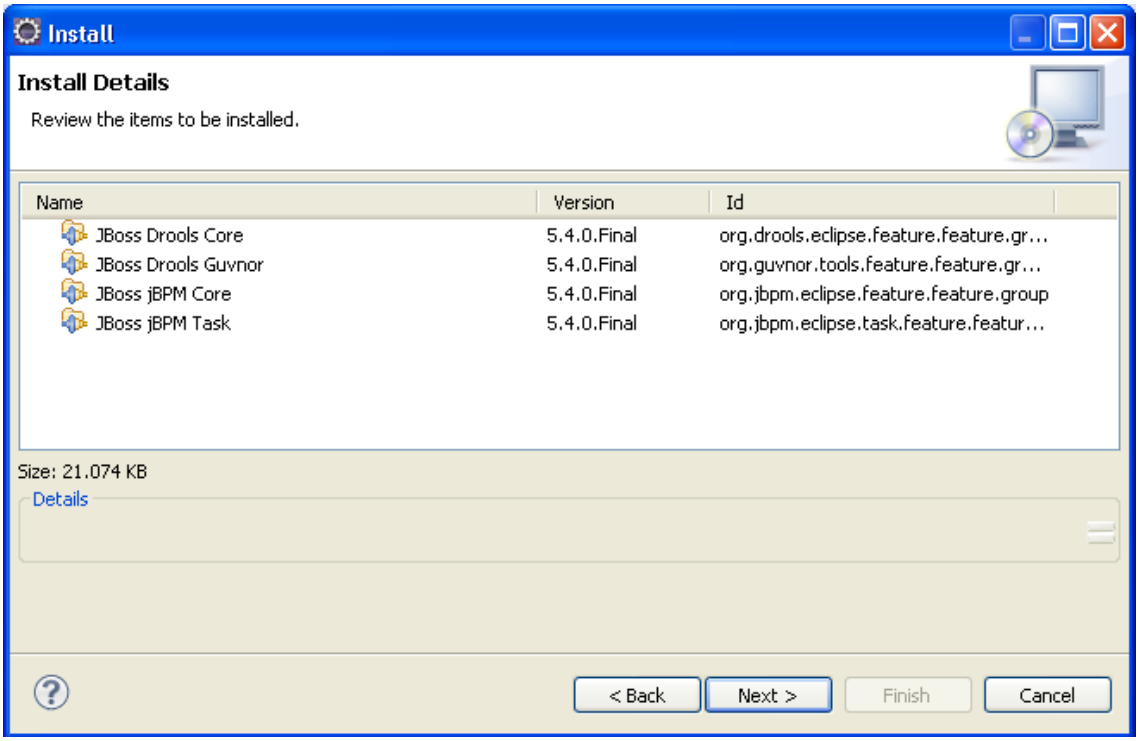
6. Seleccionar Add. En Local seleccionar el camino siguiente:

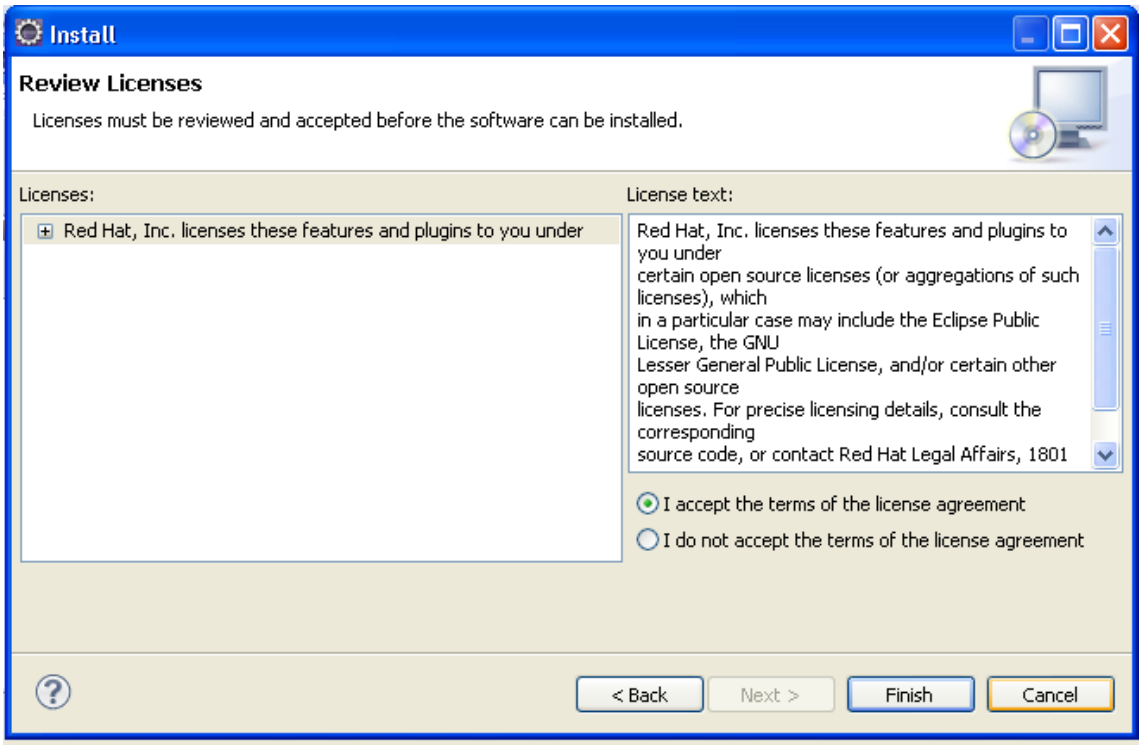
   file:/C:/Documents and Settings/Sandra/Escritorio/Nueva carpeta/droolsjbpm-tools-distribution-5.4.0.Final/binaries/org.drools.updatesite/.

   Tener en cuenta que la dirección del camino es según la que le hemos dado a nuestro proyecto.

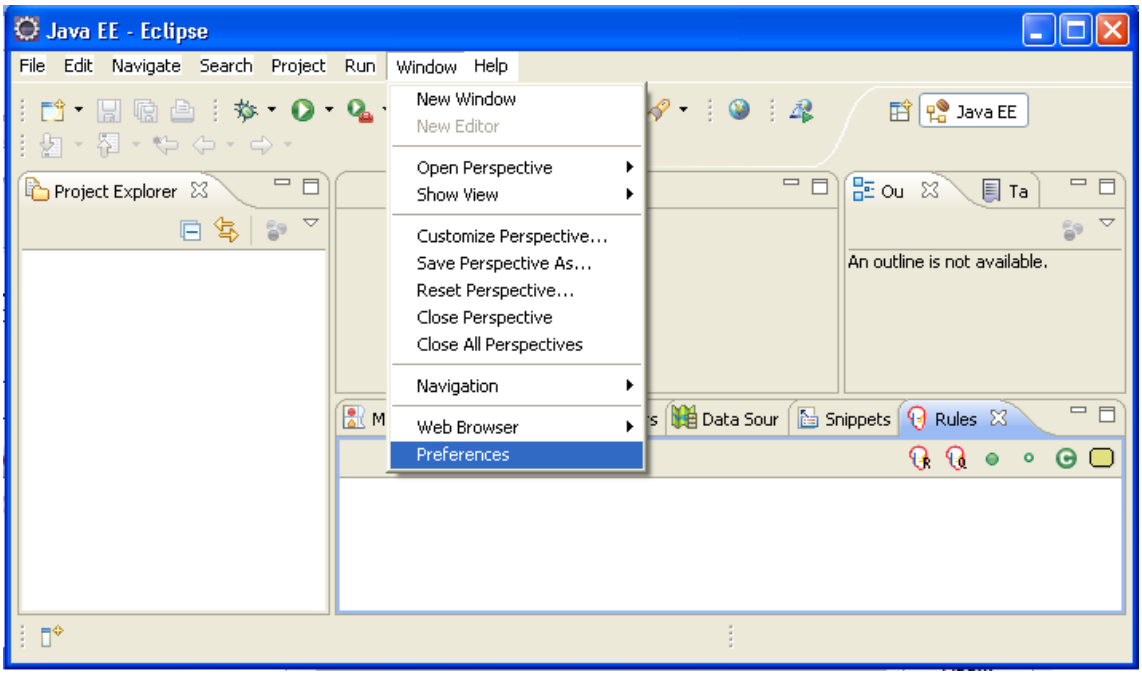


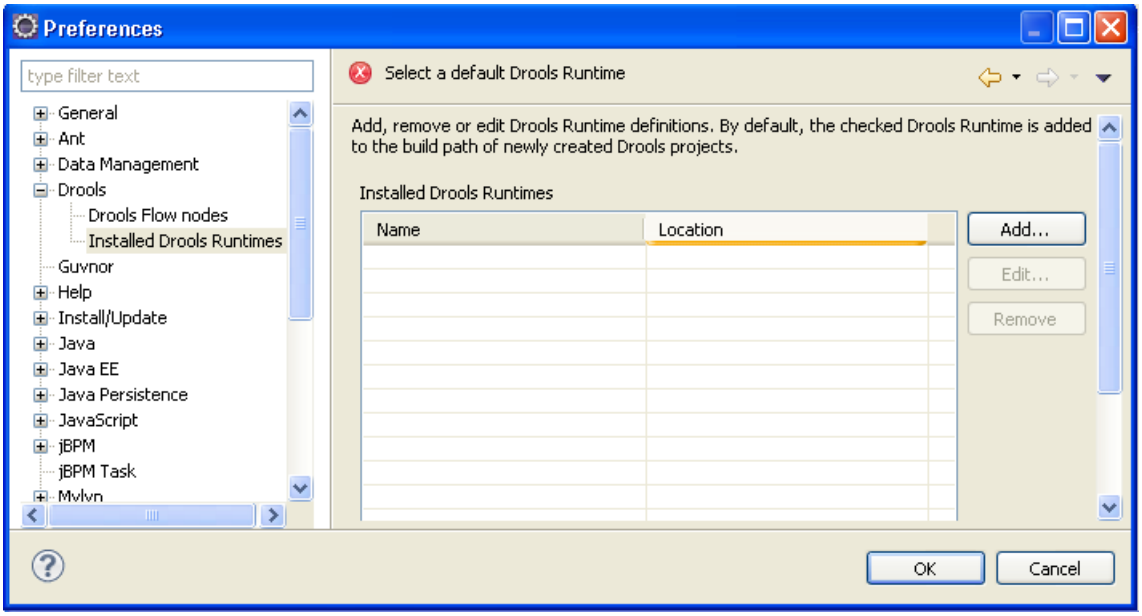
7. Seleccionamos el checkbox Drools and jBPM y damos clic en el botón **Next**.



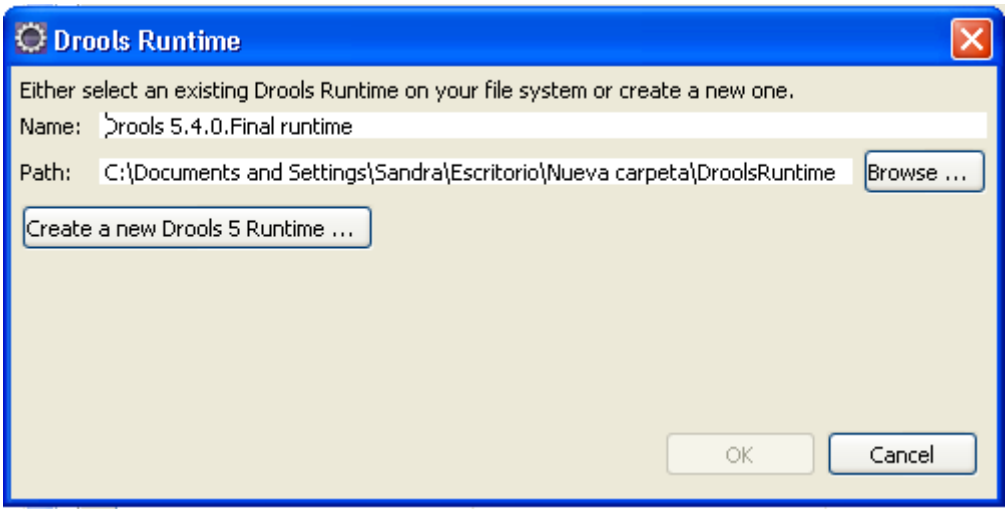


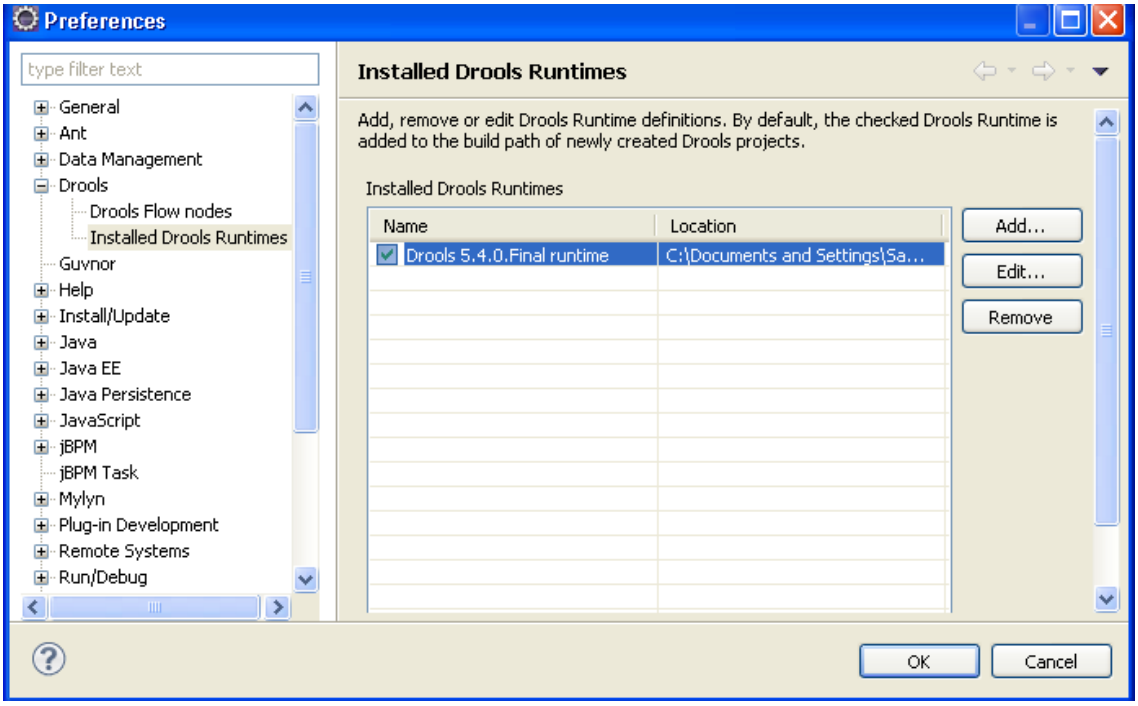
8. Después de dar clic en el botón **Finalizar**. A continuación instalaremos las dependencias necesarias para trabajar con Drools. Para ello pasamos a instalar nuestro drools-runtime que no es más que el conjunto de bibliotecas que necesitará Drools para funcionar. Para ello seleccionamos la siguiente dirección:
   Windows/Preferences/Installed Drools Runtime



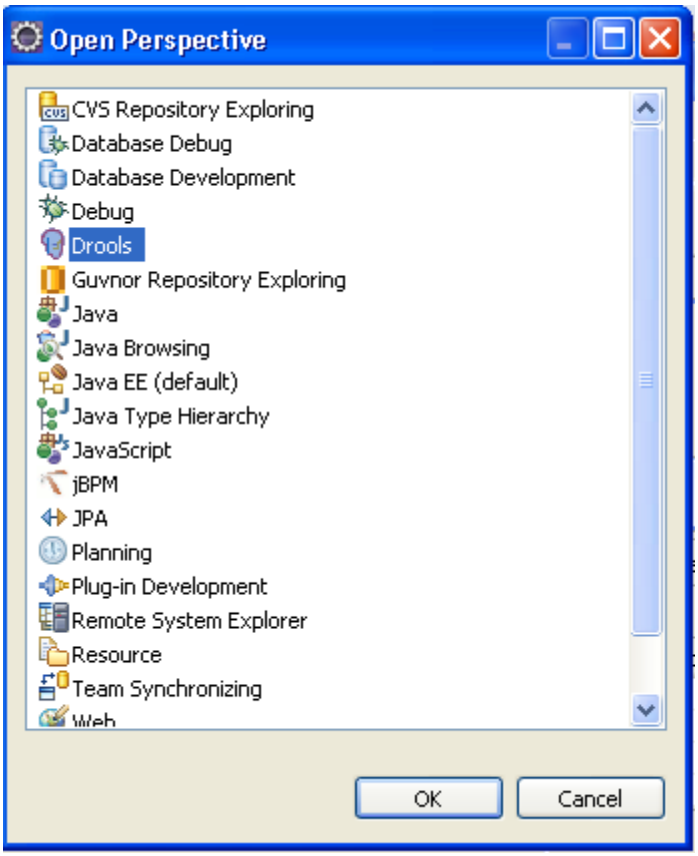


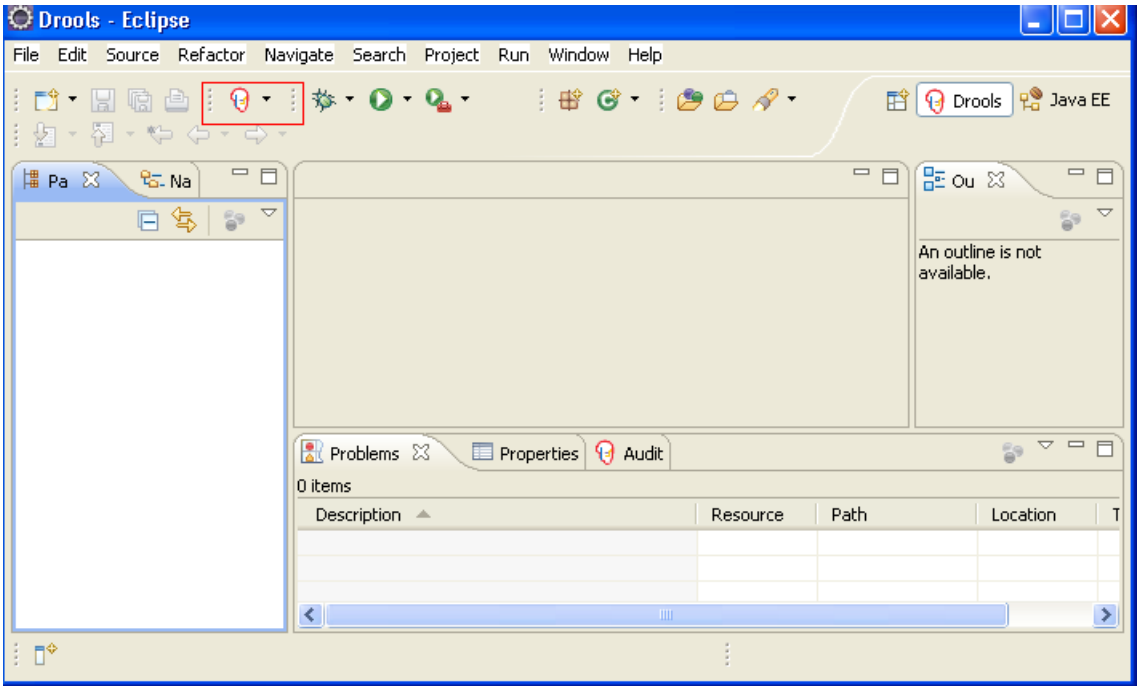
9. Pulsamos en Add y creamos un nuevo Drools Runtime pulsando sobre **Create a new Drools 5 Runtime,** luego seleccionamos el directorio donde guardaremos la carpeta que contendrá las librerías del Drools Runtime.





10. Después de este paso ya tenemos las bibliotecas instaladas en la carpeta DroolsRuntime. Ahora seleccionamos  y escogemos la opción Other y seleccionamos el icono de Drools y hacemos clic en **OK**. En la barra de herramientas aparecerá Drools como nuevo proyecto a definir.

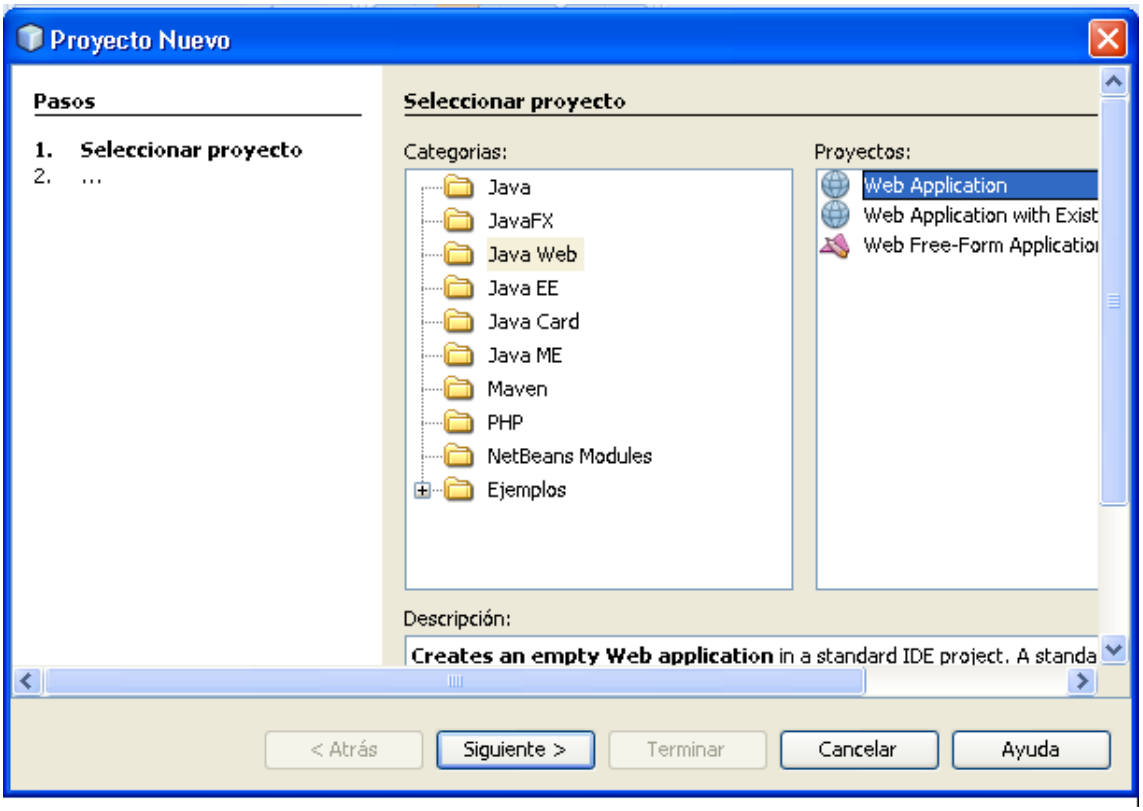

Open Perspective
CVS Repository Exploring
Database Debug
Database Development
Debug
Drools
Guvnor Repository Exploring
Java
Java Browsing
Java EE (default)
Java Type Hierarchy
JavaScript
jBPM
JPA
Planning
Plug-in Development
Remote System Explorer
Resource
Team Synchronizing
Web
OK
Cancel

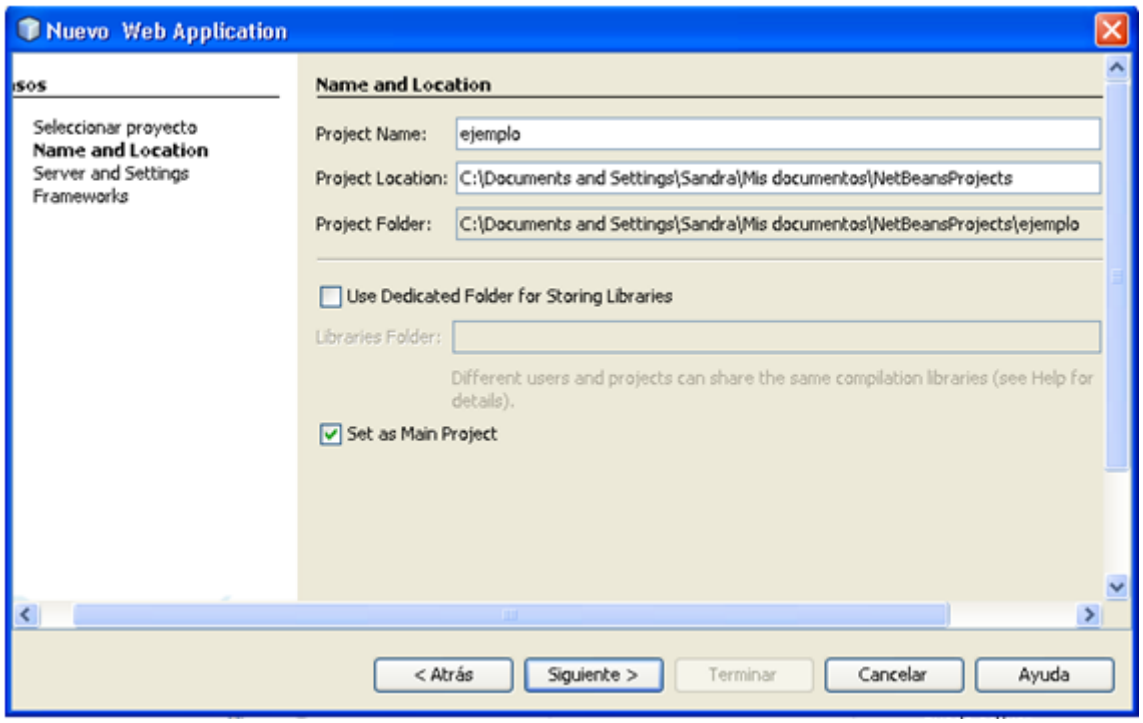

Drools - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Drools
Java EE
An outline is not available.
Problems
Properties
Audit
0 items
Description
Resource
Path
Location

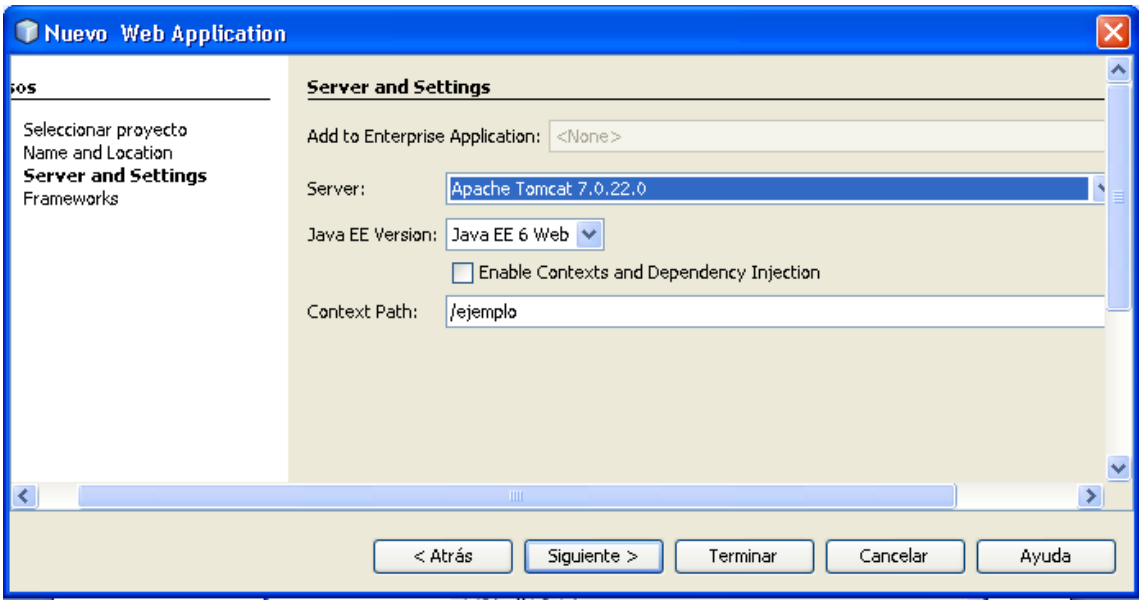
Anexo II Pasos a seguir para crear un proyecto web en NetBeans

Pasos para crear un proyecto web services en NetBeans.

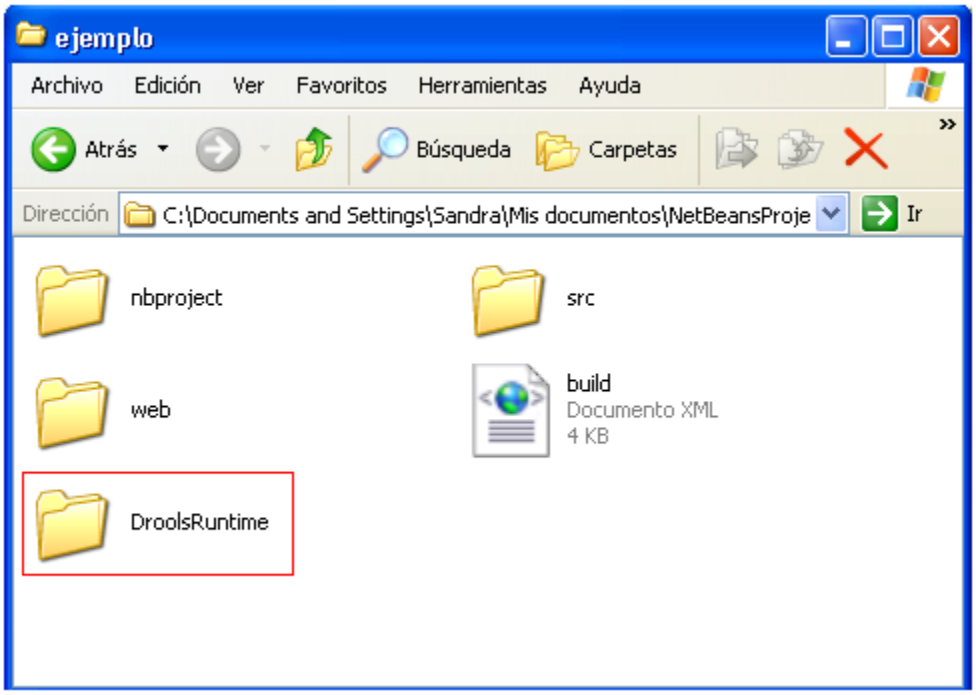
1. Crear un nuevo proyecto.



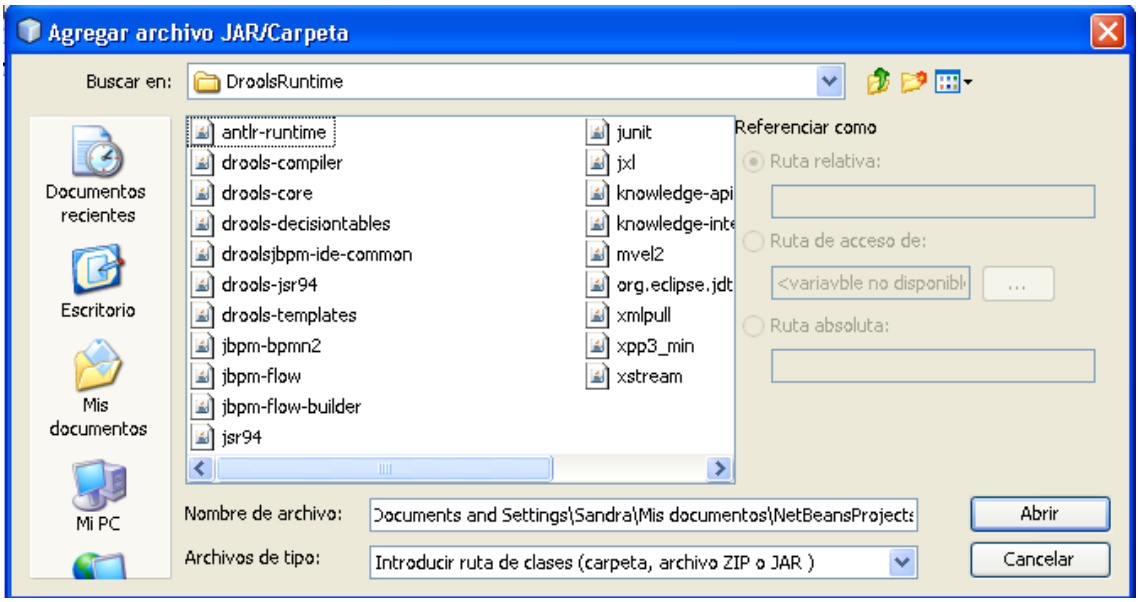




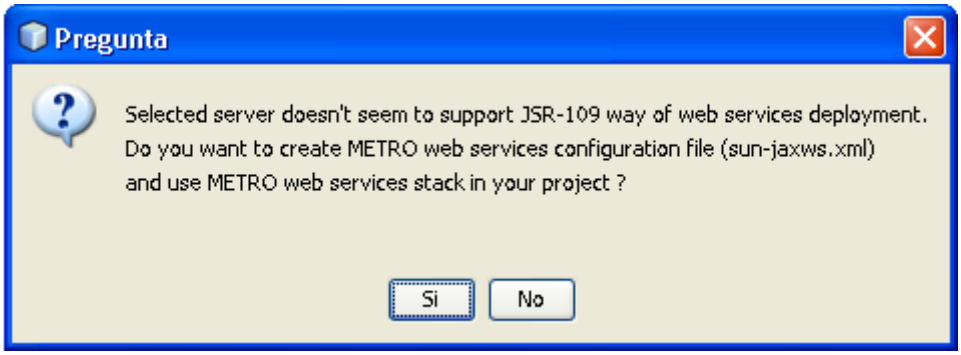
2. Para configurar el NetBeans con Drools se copia la carpeta DroolsRuntime, creada anteriormente con Eclipse, dentro del proyecto como se muestra en la figura.



3. Para añadir las librerías debemos dar clic sobre Libraries>Agregar archivo jar/carpeta, seleccionamos la carpeta copiada anteriormente y cargamos las librerías que se muestran en la figura.



4. Luego se pasa a crear el web services, la ruta a seguir es la siguiente: *Nuevo> Web Service* .Seguido de esto saldrá una ventana como la que se muestra en la siguiente figura, damos clic en el botón **Si** y se cargarán las bibliotecas de Web Service.



Las librerías de Drools y Web Services que deberán aparecer en el proyecto se muestran en la tabla siguiente.

Librerías de Drools	***Librerías de Web Services***
antlr-runtime.jar	METRO 2.0-activation.jar
drools-compiler.jar	METRO 2.0-webservices-api.jar
drools-core.jar	METRO 2.0- webservices-extra-api.jar
drools-decisiontables.jar	METRO 2.0- webservices-extra.jar
drools-jsr94.jar	METRO 2.0- webservices-rt.jar
drools-templates.jar	METRO 2.0- webservices-tools.jar
droolsjbpm-ide-common.jar	
jbpm-bpmn2.jar	
jbpm-flow-builder.jar	
jbpm-flow.jar	
jsr94.jar	
junit.jar	
jxl.jar	
knowledge-api.jar	
knowledge-internal-api.jar	
mvel2.jar	
org.eclipse.jdt.core_3.7.3.v20120119-1537	
xmlpull.jar	
xpp3_min.jar	
xstream.jar	