

## **Integración de un Sistema de Gestión de Reglas de negocio al flujo de trabajo "control de historias clínicas para trasplante renal"**

### **Integration of a Business Rules Management System for controlling medical records of the kidney's transplant workflow**

**MSc. Yoan Pacheco Cárdenas,<sup>I</sup> Sandra Estévez Abrahantes,<sup>II</sup> Dra. María Elena Martínez del Busto<sup>III</sup>**

<sup>I</sup> Master en Ciencias de la Computación. Centro de Estudios de Informática. Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba. E-mail: [yoan@uclv.edu.cu](mailto:yoan@uclv.edu.cu)

<sup>II</sup> Estudiante de Ciencias de la Computación. Facultad de Matemática Física Computación. Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba.

<sup>III</sup> Doctora en Ciencias Técnicas. Centro de Estudios de Informática. Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba.

---

#### **RESUMEN**

Este trabajo selecciona un Sistema de Gestión de Reglas de Negocio y lo integra al flujo de trabajo "control de historias clínicas para trasplante renal". Se evaluaron los programas OpenRules, OpenL Tablets y Drools y se seleccionó Drools como la mejor herramienta. Se muestra el flujo de trabajo automatizado, las herramientas utilizadas y los componentes de la arquitectura propuesta. Es una guía para desarrolladores de sistemas informáticos que automatizan los procesos médicos que incluyan reglas de negocio.

**Palabras clave:** flujo de trabajo, regla, donante, trasplante, riñón.

---

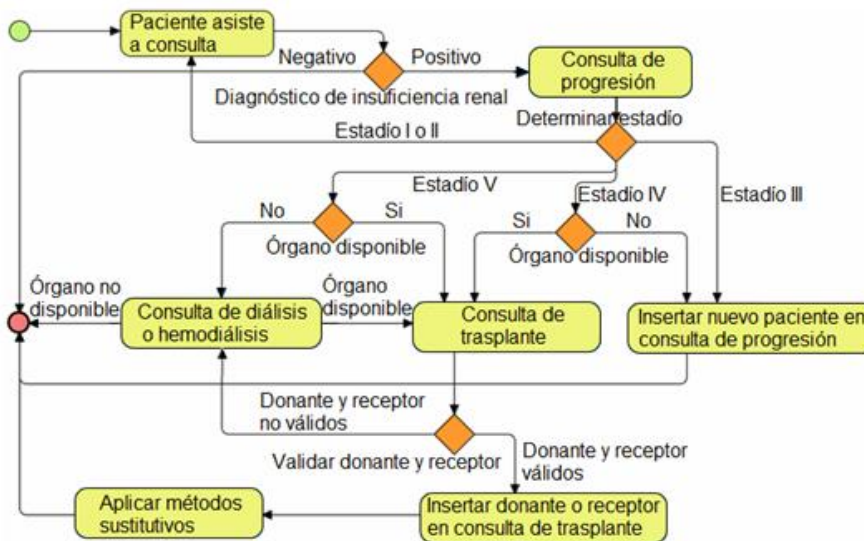
**ABSTRACT**

This paper selects a Business Rules Management System and integrates it to the workflow for "controlling medical records of kidney's transplant". OpenRules, OpenL Tablets and Drools programs are evaluated and Drools is selected as the best tool. The automated workflow is presented, also the tools used and the components of the proposed architecture. It will serve as a guide to developers of computer systems that automate medical processes with business rules.

**Key words:** workflow, rule, donor, transplant, kidney.

**INTRODUCCIÓN**

El control de historias clínicas para trasplante renal (TR) es un flujo de trabajo donde el equipo médico evalúa un grupo de reglas para tomar decisiones relacionadas con el diagnóstico de la insuficiencia renal, la determinación del estadio del paciente, la disponibilidad de órganos para el trasplante y la validación del donante y el receptor. (Fig. 1)



**Fig. 1.** Diagrama de flujo de trabajo para el caso de estudio "control de historias clínicas para trasplante renal"

Algunas de las reglas que se evalúan para validar el donante y el receptor 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.

En la literatura científica relacionada con la informatización del flujo de trabajo de trasplante renal existen investigaciones que proponen un modelo de hechos genérico para modelar las relaciones entre los conceptos de un dominio concreto haciendo uso de ontologías.<sup>1,2</sup> Estas ideas se tuvieron en cuenta en el desarrollo de aplicaciones médicas, para el trasplante renal usando reglas de negocio, en el hospital Arnaldo Millán Castro, de Santa Clara, Cuba.<sup>3</sup> Inicialmente se obtuvo una primera versión del sistema, en la cual se implementó una arquitectura Cliente/Servidor de dos capas. En esta versión inicial se utilizó de forma parcial el enfoque de RN. En una segunda versión se obtuvo una aplicación Web con arquitectura cliente/servidor de tres capas. Para ello se utilizó la plataforma de software libre CakePHP, lo que permitió separar el tratamiento de los datos, la lógica del negocio y su presentación. Sin embargo, aunque esta versión de la aplicación implementa la lógica del negocio en una capa aparte, no brinda facilidades para ejecución de un número elevado de reglas mediante un algoritmo especializado, como lo es Rete. Tampoco controla las versiones de las reglas ni facilita su publicación mediante servicios Web. Solo representa el conocimiento mediante tablas de bases de datos o disparadores. No lo hace en tablas Excel ni mediante lenguajes de dominio específico. La integración de un sistema de gestión de reglas de negocio (del inglés BRMS) al flujo de trabajo "control de historias clínicas para trasplante renal" ayudaría en la solución de estas dificultades pues un BRMS tiene módulos especializados que resuelven los problemas planteados anteriormente y que facilitan la gestión de reglas a los usuarios del negocio, dígase equipo médico, y a los especialistas en tecnologías de la información.

Nuestro trabajo tiene como objetivo responder las siguientes interrogantes:

- ¿Qué BRMS es el adecuado para ser integrado al flujo de trabajo "control de historias clínicas para trasplante renal", teniendo en cuenta criterios como: licencia de código abierto, precio y características tecnológicas?
- ¿Cómo integrar el BRMS seleccionado al flujo de trabajo "control de historias clínicas para trasplante renal"?

## **MATERIALES Y MÉTODOS**

Para seleccionar el BRMS que gestione las reglas del flujo de trabajo "control de historias clínicas para trasplante renal" se listaron las propiedades tecnológicas más importantes de los programas de código abierto y gratis que implementan los componentes de un BRMS. A continuación se evaluaron cuantitativamente sus

---

propiedades tecnológicas para lo cual fue utilizado el método propuesto por Graham.<sup>4</sup> El mismo evalúa 80 propiedades que tributan a categorías generales como las facilidades para captura de conocimiento, calidad de la herramienta de gestión, características del motor de reglas, facilidades multiusuario y otros factores técnicos y culturales. La puntuación total se obtuvo como la suma de los

acumulados individuales de las propiedades multiplicados por sus pesos. Para la informatización del proceso de negocio se tuvieron en cuenta directrices de diseño que garanticen bajo acople entre las capas y alta cohesión entre las actividades interiores, así como fácil interpretación del proceso por el equipo médico.<sup>5</sup>

Para el desarrollo de la solución informática se utilizaron los lenguajes de programación Java y Php. Como servidores Web se seleccionaron Apache para páginas Php y Tomcat para desplegar los servicios Web implementados en Java. El BRMS seleccionado corre en la plataforma Java por lo que también fue desplegado en Tomcat. El sistema gestor de bases de datos utilizado fue MySQL. El paquete XAMP se utilizó para desarrollar el proyecto. XAMP contiene un conjunto de herramientas libres que corren en varias plataformas. Lo conforman el servidor de bases de datos MySQL, el servidor Web Apache e intérpretes para los lenguajes Php y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin.

## RESULTADOS Y DISCUSIÓN

### **Selección del BRMS adecuado para ser integrado al control de historias clínicas para trasplante renal**

Programas de código abierto como CLIPS, NxBRE, Jena, Hammurapi Rules y CodaServer implementan un motor de reglas y proporcionan algunos mecanismos de gestión. Sin embargo, solo Drools, OpenRules y OpenL Tablets tienen los componentes principales de un BRMS, que son: el motor de reglas, el repositorio de reglas y mecanismos variados de gestión. En la tabla 1 se muestra un resumen de las fortalezas tecnológicas de los componentes de Drools, OpenRules y OpenL Tablets, según la bibliografía científica.

El motor de reglas de las tres aplicaciones implementa el algoritmo Rete, aunque en problemas de optimización OpenRules permite optar por un motor específico para ello. En OpenRules y OpenL Tablets el motor solo hace inferencia con encadenamiento hacia adelante, mientras que en Drools realiza inferencia con encadenamiento hacia adelante y hacia atrás, por tanto la ejecución de las reglas no depende del orden en el repositorio. Los tres programas pueden integrarse con otras aplicaciones remotas y ser accedidos a través de internet, pues publican reglas de negocio mediante servicios Web. La creación y ejecución de casos de prueba permite a los tres productos validar la ejecución de reglas antes de ser desplegados en un escenario empresarial. Drools representa las reglas en un mayor número de formatos, lo que posibilita su integración con otras herramientas y su uso por personal de informática y expertos del negocio. Drools es un programa más robusto y completo, pues puede gestionar un mayor número de reglas complejas, transacciones y usuarios; y además de los componentes de un BRMS, tiene módulos para gestionar flujo de trabajo, procesamiento de eventos y planificación de tareas, a diferencia de OpenRules y OpenL Tablets, que solo poseen los módulos de un BRMS. La licencia de Drools es ASL2 y la de OpenL Tablets es LGPL, ambas son menos restrictivas que la GPL de OpenRules que obliga al software derivado a usar GPL también.

**Tabla 1.** Fortalezas tecnológicas de los componentes de Drools, OpenRules y OpenL Tablets

BRMS	Motor de reglas	Repositorio de reglas	Mecanismos de gestión	Fuente
Drools	Ejecuta inferencia con encadenamiento hacia adelante y hacia atrás. Implementa algoritmo Rete y el estándar JSR-94. Manipula elevado número de reglas, pedidos/transacciones y usuarios.	Usa JCR para gestionar el repositorio de reglas. Tiene control de versiones. Usa el estándar JAAS para la autorización y autenticación.	Permite editar reglas en Java, XML, en el lenguaje nativo de Drools (drl), en lenguajes de dominio específico y en Excel. Tiene complemento para el programa Eclipse y editor Web. Publica reglas como servicios Web. Genera reportes. Permite crear y ejecutar casos de prueba. Facilita la depuración de reglas. Permite crear vocabulario organizacional.	6
OpenRules	Realiza inferencia con encadenamiento hacia adelante. Implementa el algoritmo Rete y un algoritmo específico para problemas de optimización. Implementa el estándar JSR-33	Almacena y administra reglas. Tiene control de versiones.	Posibilita representar reglas en lenguaje XML y tablas de decisión en Excel, OpenOffice o GoogleDocs. Tiene complemento para el programa Eclipse y editor Web. Publica reglas como servicios Web. Tiene aplicación Web para editar reglas. Genera reportes. Permite crear y ejecutar casos de prueba. Facilita	7
			la creación de un vocabulario organizacional.	
OpenL Tablets	Ejecuta inferencia con encadenamiento hacia adelante. Implementa el algoritmo Rete.	Almacena y administra reglas. Tiene control de versiones.	Tiene editor Web, publica reglas como servicios Web. Hace posible la edición de reglas en lenguaje Java simplificado y en tablas de Excel. Tiene complemento para el programa Eclipse. Permite crear y ejecutar casos de prueba.	8

Teniendo en cuenta que Drools tiene licencia de código abierto y que no establece restricciones sobre el software derivado, que además es gratis y que según los resultados de la aplicación del método de Graham es tecnológicamente superior a los otros dos softwares analizados anteriormente, se decidió seleccionar este BRMS para integrarlo al flujo de trabajo "control de historias clínicas para el trasplante renal".

### Integración de Drools al flujo de trabajo "control de historias clínicas para el trasplante renal"

En el flujo de trabajo de la solución propuesta las reglas de negocio, que validan las características del posible receptor y del posible donante son ejecutadas en un módulo separado del resto de la aplicación, por el motor del sistema de gestión de reglas de negocio Drools. El resto de las actividades son ejecutadas por el equipo médico o un responsable de controlar las historias clínicas de los pacientes. En esta nueva versión de la aplicación las reglas son guardadas en un repositorio que brinda interfaces para editar las mismas así como otras facilidades para su gestión. Separar las reglas del código que controla el acceso a bases de datos, la interfaz de usuario y las decisiones del negocio favorece la encapsulación lo cual hace al sistema más flexible. Un cambio en el interior de uno de los módulos tiene menos probabilidades de afectar al resto de la aplicación. (Fig. 2)

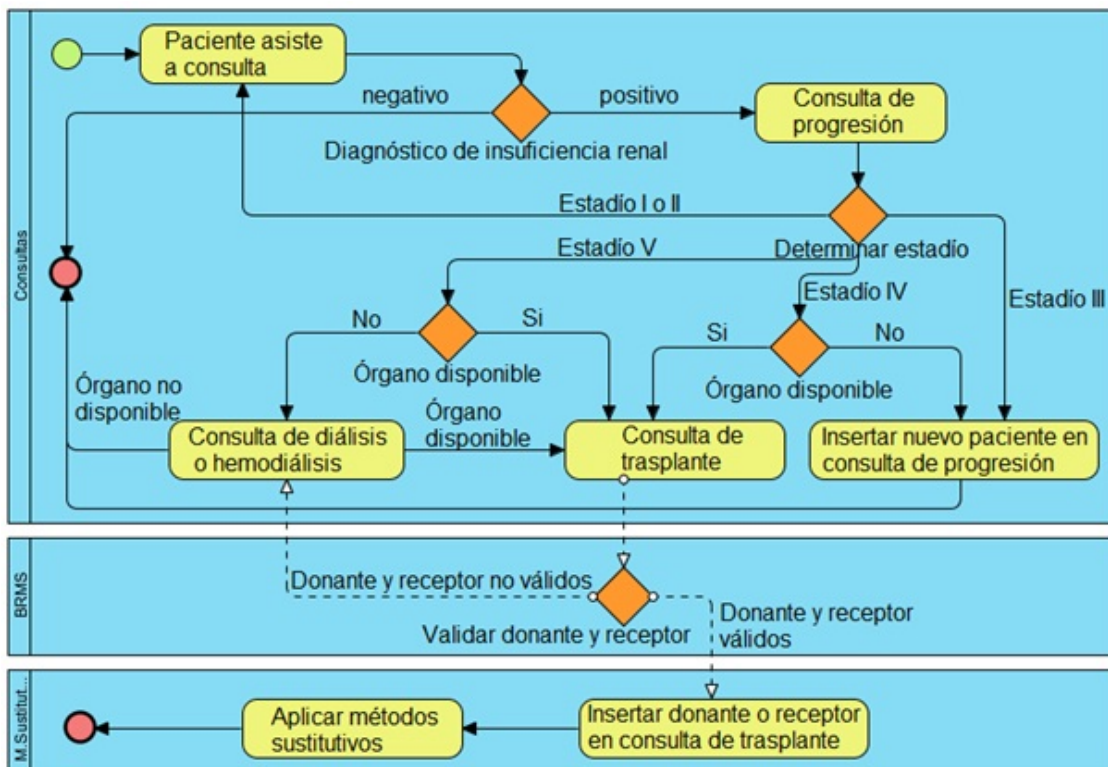
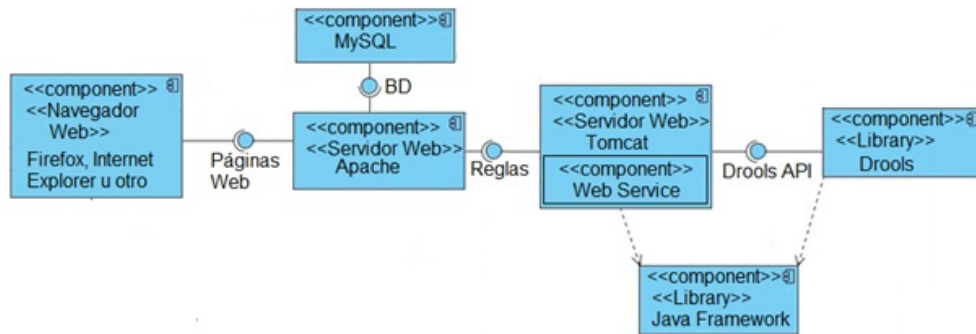


Fig. 2. Diagrama informatizado de flujo de trabajo para el caso de estudio "control de historias clínicas para trasplante renal"

El flujo de trabajo se descompone en tres etapas, en la primera se realizan las actividades relacionadas con las consultas. En la segunda las acciones relacionadas con la validación del donante y el receptor. En la tercera etapa se ejecutan las actividades concernientes con los métodos sustitutivos. A su vez, las tres etapas y sus actividades están distribuidas en las cinco capas de la arquitectura del sistema. (Fig. 3)



**Fig. 3.** Arquitectura del sistema que integra el BRMS Drools al flujo de trabajo "control de historias clínicas para trasplante renal"

La arquitectura del sistema está dividida en las siguientes capas:

1. Interfaz de usuario: posibilita al equipo médico interactuar con el sistema y ejecutar las actividades informatizadas del flujo de trabajo. Para ello invoca operaciones de las interfaces publicadas por la capa de negocio. Es un conjunto de páginas HTML y Php alojadas en un servidor Web Apache que son mostradas en un navegador Web.
2. Capa de negocio: es la encargada de realizar cálculos, consultas e invocar la ejecución de reglas de negocio. Para ello accede a la capa datos y a la capa de servicios de reglas. Es un conjunto de páginas Php alojadas en un servidor Web Apache.
3. Capa de datos: es la encargada de gestionar la base de datos del sistema, la cual contiene información sobre todos los actores del proceso y las etapas del mismo para cada paciente. Brinda interfaces para que la capa de negocio pueda accederla. Para implementar esta capa se utilizó el sistema gestor de bases de datos MySQL.
4. Capa de servicios de reglas de negocio: expone mediante servicios Web interfaces a reglas de negocio. Es invocada desde la capa de negocios y ejecuta las reglas que contiene la capa de reglas. La componen servicios Web, programados en Java, que corren en un servidor Web Tomcat.
5. Capa de reglas: posibilita gestionar las reglas de negocio. Para lo cual brinda interfaces a la capa de servicios de reglas. Para implementar esta capa se utilizó un BRMS desarrollado en java, gratis y de código abierto. El BRMS se despliega en un servidor Web Tomcat.

En la solución informática propuesta, para validar si una persona puede ser donante o no, un gestor del sistema solicita insertar un nuevo donante. Esta solicitud es procesada en el servidor Web Apache, el cual invoca un servicio Web, que corre sobre Tomcat y ejecuta reglas en Drools para validar si la persona cumple con todas las características que debe tener un donante. En caso afirmativo, la persona es insertada en la lista de donantes que contiene la base de datos y el gestor del sistema es informado mediante la interfaz de usuario.

## CONCLUSIONES

Se realizó un estudio de los BRMS existentes en el mercado y se seleccionó Drools como el programa adecuado para ser integrado al flujo de trabajo "control de historias clínicas para trasplante renal". Para ello se tuvieron en cuenta criterios como: licencia, precio, rendimiento de la herramienta y facilidades para el equipo médico y para los usuarios especialistas de informática. Se diseñó una arquitectura

de cinco capas que integra el BRMS Drools al flujo de trabajo analizado. Se utilizaron componentes gratis y de código abierto en la implementación del sistema. Esta investigación puede servir de guía a desarrolladores y arquitectos de software que necesiten informatizar procesos médicos que tengan un número elevado de reglas de negocio.

## REFERENCIAS BIBLIOGRÁFICAS

1. Martínez del Busto ME, Moreno I, Hernández P. Modelo de hechos genérico para procesar reglas de negocio mediante para trasplante renal. Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology 2009, San Cristobal, Venezuela [citado 2014 Nov 20]. Disponible en: <http://laccei.org/LACCEI2009-Venezuela/tableOfContent.html>
2. Martínez del Busto ME, Moreno I, Rodríguez A, Castro M, González L. Vocabulario de negocio para trasplante renal con enfoque ontológico para un modelo de hechos genérico. Rev Facultad de Ingeniería. 2010 [citado 2014 Nov 20]; (53). Disponible en: <http://www.redalyc.org/articulo.oa?id=43019325014>
3. Martínez del Busto ME, et al. Aplicación médica para trasplante renal usando reglas de negocio. Rev haban cienc méd. 2012 [citado 2014 Nov 20]; 11(1):176-84. Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_pdf&pid=S1729-519X2012000100021&lng=es&nrm=iso&tIng=es](http://scielo.sld.cu/scielo.php?script=sci_pdf&pid=S1729-519X2012000100021&lng=es&nrm=iso&tIng=es)
4. Graham I. Business Rules Management and Service Oriented Architecture. Chichester: John Wiley & Sons, Inc., 2007.
5. Moreno I, Rodríguez A, Moreno R, Casas G, González L. Directrices prácticas y métricas de calidad en la modelación de procesos de negocio: un caso de estudio. Revista Cubana de Ciencias Informáticas. 2014 [citado 2014 Nov 20]; 8(2):1-18. Disponible en: <http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path%5B%5D=709>
6. Bali M. Drools JBoss Rules 5.X Developer's Guide. Birmingham: Packt Publishing. 2013.
7. OpenRules, Inc. OpenRules User Manual. 2014 May [citado 2014 Nov 20]. Disponible en: <http://openrules.com/pdf/OpenRulesUserManual.pdf>
8. OpenL Tablets. OpenL Tablets Documentation. [citado 2014 Nov 20]. Disponible en: <http://openl-tablets.sourceforge.net/documentation>

Recibido: 15 de octubre de 2014.

Aprobado: 7 de abril de 2015.