

**Universidad de las Ciencias Informáticas**

**Facultad 15**



**Título:** Motor de Reglas de Negocio para la arquitectura de dominio específico del polo de Sistemas Tributarios.

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas**

**Autor:** Ricardo Pablo Avila Alfaro

**Tutor:** Msc. Julio Cesar Diaz Vera

**Ciudad de La Habana,  
“Año del 50 Aniversario del Triunfo de la Revolución”**

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autor de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Ricardo Pablo Avila Alfaro

Julio Cesar Díaz Vera

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor



*"... las cualidades morales suben de precio cuando están  
realzadas por las cualidades inteligentes..."*

*José Martí*

Deseo agradecer en primer lugar a toda mi familia que sin su guía no hubiera podido llegar hasta aquí, en especial a mi mamá, mi papá y mi hermana. Tampoco hubiera llegado hasta aquí sin la buena intervención de los doctores Víctor Dubois y Jorge Luis Rojas. Además quisiera agradecer a Gretter por apoyarme en todo momento. A Julio por confiar en mí para desarrollar este trabajo y a Lázaro por ayudarme a hacerlo realidad, y claro está a todos mis hermanos de la UCI, y sin olvidar a la misma universidad por darme la oportunidad de formarme en ella. En fin le agradezco a todas la personas que contribuyeron a la realización de este trabajo.

A mi familia, que sin su guía no hubiera podido llegar hasta aquí.

La gestión de cualquier empresa es dirigida por innumerables reglas que determinan la manera en que esta debe operar. Tradicionalmente los sistemas de información que automatizan la gestión de las empresas colocaban estas reglas directamente dentro del código de la aplicación y si bien esto era posible en los entornos más o menos rígidos de hace unas décadas es imposible con el dinamismo de la empresa actual debido a los costes que implicaría mantener actualizadas las aplicaciones a medida que cambian las reglas.

En el presente trabajo se desarrolla un motor de reglas de negocio que facilita la gestión de las reglas en el dominio aduanal, se sigue un enfoque de desarrollo centrado en los datos y se desarrollan rutinas capaces de generar automáticamente el código necesario para ejecutar las reglas de forma tal que al ser modificadas las mismas se actualiza automáticamente el componente necesario para ejecutarlas ya sea a nivel de interfaz o dentro de la lógica de negocio o el modelo persistente de las aplicaciones.

---

<b>1. CAPITULO 1. INTRODUCCION.....</b>	<b>1</b>
1.1. Situación y propósito. ....	2
<b>2. CAPITULO 2. MARCO CONCEPTUAL .....</b>	<b>6</b>
2.1. Introducción.....	6
2.2. Reglas de Negocio .....	7
2.3. Componentes de los Sistemas de Reglas de Negocios .....	10
2.3.1. Estructura.....	11
2.3.1.1. Términos.....	12
2.3.1.2. Hechos.....	13
2.3.1.3. El modelo de hechos y el modelo de datos .....	14
2.3.2. Poder.....	16
2.3.3. Control.....	20
2.4. Motor de Regla de Negocio .....	27
2.4.1. Historia .....	27
2.4.2. Cuando usar un motor de reglas de negocio .....	29
2.4.3. ¿Por qué construir un motor de reglas de negocio? .....	34
<b>CAPITULO 3. DISEÑO DEL SISTEMA.....</b>	<b>36</b>
3.1. Requerimientos no funcionales.....	36
3.1.1. Restricciones en el diseño y la implementación .....	36
3.1.2. Software.....	36
3.1.3. Hardware.....	36
3.2. Descripción de la solución .....	37
3.2.1. Repositorio de Reglas de Negocio .....	38
3.2.1.1. Metadatos del motor de reglas .....	38
3.2.1.2. Definición de las reglas .....	50
3.2.1.3. Procesos de negocio.....	52
3.2.2. Interfaz de definición de las reglas .....	57

3.2.3. Interfaz de invocación de las reglas .....	58
3.2.4. Rutinas de generación de código .....	59
<b>CONCLUSIONES.....</b>	<b>61</b>
<b>RECOMENDACIONES.....</b>	<b>62</b>
<b>BIBLIOGRAFIA.....</b>	<b>63</b>



Tabla 1. Ejemplo de Tipos de Control .....	21
Tabla 2: Clasificación de los tipos de tablas .....	41
Tabla 3: Descripción de la tabla "m_tc_tabla" .....	43
Tabla 4: Clasificación de los tipos de columna .....	44
Tabla 5: Descripción de la tabla "m_tc_campo".....	46
Tabla 6: Descripción de la tabla "r_campo_tipo_dato" .....	49
Tabla 7: Descripción de la tabla "s_bre_regla" .....	52
Tabla 8: Clasificación de los tipos de errores .....	52
Tabla 9: Descripción de la tabla "s_bre_proceso" .....	55
Tabla 10: Descripción de la tabla "s_bre_proceso_elemental" .....	55
Tabla 11: Descripción de la tabla "s_bre_proceso_relacion" .....	56
Tabla 12: Descripción de la tabla "s_bre_parametro".....	56
Tabla 13: Clasificación de los tipos de pasos de un proceso de negocio .....	56
Tabla 14: Clasificación de los tipos de reglas .....	57

Figura 1: Componentes del Motor de Reglas de Negocio .....	37
Figura 2: Modelo lógico de la entidad "m_tc_tabla" .....	42
Figura 3: Modelo lógico de la entidad "m_tc_campo" .....	46
Figura 4: Modelo lógico de los tipos de datos .....	48
Figura 5: Modelo lógico del componente de los metadatos .....	49
Figura 6: Modelo lógico de la definición de las reglas. ....	51
Figura 7: Modelo lógico para definir los proceso empresariales .....	53
Figura 8: Interfaz de la definición de las reglas de negocio.....	58
Figura 9: Modelo lógico del componente de invocación de las reglas de negocio .....	59

### 1. CAPITULO 1. INTRODUCCION

Hablar de reglas es inevitable cuando se habla de sistemas de información, todo negocio requiere de cientos de reglas para controlar la manera en que se desarrollan sus procesos fundamentales, ellas forman parte del núcleo básico de la organización. A tal punto es así que para algunos autores la capacidad de gestionar las reglas de negocio constituye un elemento de supervivencia para la empresa (Graham, 2006).

A la par que las empresas han ido desarrollando sistemas de información para gestionar las operaciones que llevan a cabo, han tenido que incorporar mecanismos para adicionar a estos sistemas las reglas de negocio que rigen el comportamiento de cada uno de los procesos que desarrollan. La mayoría de las aproximaciones iniciales atacaban esta situación desde enfoques de “caja negra” que ocultaban a los especialistas del negocio el modo en que las reglas serían manejadas y responsabilizaba a los desarrolladores con esta tarea.

Este enfoque terminó atando las reglas a los componentes de software en los que se implementaban y dificultando sobre manera la gestión de las mismas. Inicialmente hubo éxito en estos desarrollos gracias al carácter estático de los negocios en décadas pasadas. Sin embargo esta situación no es sostenible en medio de la dinámica actual del mundo empresarial que obliga a las empresas a cambiar sus formas de hacer para sobrevivir.

Una buena parte de lo que se denomina formas de hacer de una empresa descansa sobre las reglas de negocio que esta implementa y por ello se ha hecho necesario desplazar la gestión de reglas de negocio desde los enfoques de caja negra hasta enfoques de caja blanca o caja de cristal (Chisholm, 2007). De manera tal que los especialistas del negocio tengan control sobre las mismas y aumenten el dinamismo en su gestión.

La conjunción entre el enfoque de reglas de negocio y la utilización del mismo dentro de los sistemas de información radica justamente en que la lógica que se necesita para manejar las operaciones diarias si son definidas correctamente, es posible implementarla en aplicaciones de software. Con esto no se hace referencia al hecho trivial de que toda aplicación contiene lógica que controla la manera en que se ejecutan los procesos sino más bien al hecho de que es posible identificar estas reglas y separarlas del

código de aplicación en que se ejecutan, de esta forma sería posible gestionar las reglas por parte de los especialistas de negocio y al mismo tiempo si la definición es suficientemente consecuente permitiría a los desarrolladores definir rutinas para ejecutar las reglas de manera automática. Esta aproximación es conocida como motores de reglas de negocio lo cual constituye en última instancia el interés primario para este trabajo.

### 1.1. Situación y propósito.

La situación de dinamismo en el actuar de las organizaciones es especialmente significativa en un dominio específico como el de las instituciones aduanales debido a la doble misión de control del comercio y facilitación del mismo que les obliga a cambiar las reglas que se aplican a los controles al comercio para no permitir que estos sean modelados y por lo tanto violados por comportamientos delictivos, al mismo tiempo necesitan capacidad para re-configurar las facilidades que se dan al comercio para que estas no sean utilizadas para violar las leyes establecidas en el comercio internacional.

La aduana cubana dentro del sistema informático SUA<sup>1</sup>, el cual utiliza para la gestión de los procesos de la organización, sigue un enfoque de gestión de las reglas de negocio de caja negra en el que los especialistas de negocio no pueden controlar de manera directa las reglas propiamente dichas. Si bien es necesario acotar que la implementación del sistema intenta flexibilizar, en la medida de lo posible para este tipo de esquema, el código de la aplicación de forma tal que todos los elementos que se consideran pueden variar, producto de cambios en el negocio, han sido parametrizados y se toman de la base de datos en lugar de estar directamente cableados en la aplicación.

De cualquier forma un grupo de dificultades han sido detectadas en su accionar diario tanto desde el punto de vista de los sistemas de información como puramente desde el punto de vista del negocio, y de manera particular son de interés los siguientes.

- Algunos sistemas no se utilizan debido a que cambios en las reglas del negocio les han hecho perder utilidad. La mayor incidencia en este sentido radica en los sistemas de enfrentamientos.

---

<sup>1</sup> Sistema Único de Aduanas.

- Los cambios en las reglas del negocio requieren un arduo trabajo de mantenimiento que provoca jornadas maratónicas por parte del departamento de tecnologías de la información.
- Se atrasan los desarrollos de nuevos componentes que aumentan el valor añadido del sistema.

Es por ello que se define como **problema** a tratar en este trabajo *¿Cómo disminuir los tiempos necesarios para mantener y desarrollar aplicaciones en el dominio aduanal siguiendo un enfoque de reglas de negocio?*

El **objeto de estudio** estará centrado en los sistemas de gestión de reglas de negocio y el **campo de acción** será el proceso de definición de un mecanismo de gestión de reglas de negocio para la arquitectura de dominio específica definida en (Cobo Rodríguez, 2008) para las aplicaciones de la Aduana General de la República.

Este trabajo persigue como **objetivo general** el desarrollo de un motor de reglas de negocio para la arquitectura de dominio específico del sistema SUA. Para cumplimentar este objetivo son definidos los siguientes objetivos específicos.

- Determinar la tecnología a utilizar (Revisión Bibliográfica).

La idea central es obtener un listado de las tecnologías utilizadas en el desarrollo de algún tipo de componente de reglas de negocio y decidir si es posible utilizar alguna implementación particular o será necesario desarrollar una propia, ya sea por la adaptación de uno de los existentes o desde cero. En este sentido se definen un grupo de tareas que son listadas a continuación.

1. *Identificación de las fuentes primarias de información en reglas de negocio.*
  2. *Recopilar la información disponible.*
  3. *Determinar las fuentes relevantes.*
  4. *Definir los criterios de comparación.*
  5. *Realizar las comparaciones.*
  6. *Documentar los resultados.*
- Desarrollar un intérprete XPD (Implementación).
  - Desarrollar un visualizador de procesos (Implementación).

- Desarrollar un motor de reglas de negocio (Implementación).

La implementación de los tres componentes previamente expuestos posibilitará contar con el motor de reglas de negocio que constituye el objetivo primario de la investigación. Para ello deben ser ejecutadas las siguientes tareas:

1. Determinación de la arquitectura de desarrollo a seguir. En este punto debe ser documentada la arquitectura de forma tal que pueda validarse la implementación realizada por otros desarrolladores que pudiesen estar interesados en estos temas y serian capaces de constatar la efectividad del código realizado y la validez de la solución propuesta.
  2. Implementación propia de los componentes siguiendo la arquitectura referenciada, documentación de los mismos con el objetivo de que se puedan comprobarse por otro investigador la validez de los resultados.
- Demostrar la factibilidad de la solución (Caso de Estudio).

Como elemento adicional es necesario demostrar la factibilidad de la utilización de un motor de reglas de negocios con vista a disminuir los tiempos de desarrollo de las aplicaciones dentro del dominio aduanal. Para ello se deben desarrollar las siguientes tareas:

1. *Escoger los casos.* Para este punto deben ser considerados los casos que hayan sido implementados antes y de los que se disponga los que tengan suficiente información asociados a sus tiempos de implementación.
2. *Implementar los casos.* Utilizando el motor de reglas de negocio.
3. *Medir los tiempos de desarrollo.*
4. *Comparar con los resultados anteriores.*

El resto del informe de tesis está estructurado como se describe a continuación. En el segundo capítulo se desarrolla el marco conceptual de la investigación, se presentan y analizan las metodologías y las tecnologías relevantes para el problema de esta investigación y se asume una posición acerca de las características de éxito para un motor de reglas de negocio.

El capítulo tres, describe la solución propuesta; son identificados los requerimientos que debe garantizar el sistema, se presentan y describen los componentes que conforman la solución, como por ejemplo la estructura para almacenar los metadatos de la aplicación, el mecanismo de definición de las reglas de negocio y las rutinas para la generación de la lógica de las reglas de negocio.

## 2. CAPITULO 2. MARCO CONCEPTUAL

### 2.1. Introducción

El término reglas de negocio cada vez es más presente en el espectro de las tecnologías de la información (en lo adelante TI) y en la industria del software. Sin embargo, la idea de las reglas en los sistemas de negocio es, irónicamente, ajena a la mayoría profesionales de TI (Chisholm, 2007). Dígase reglas de negocio y muchos profesionales de TI piensan vagamente en sistemas expertos o sistemas con enfoque de inteligencia artificial, los cuales se consideran convenientes para problemas muy especializados y/o muy avanzado. Existe poco reconocimiento de cómo las reglas de negocio son en realidad la base de las operaciones del día a día de la empresa. No es coincidencia que muchos trabajadores del negocio están tan adoctrinados en las vistas procedimentales para la elaboración de requisitos que pensar en términos de reglas de negocio, les resulta, inicialmente, extraño y/o abstracto a la vez.

Muchas de las cuestiones a las que los profesionales de las TI se enfrentan implican interacción con una variedad importante de campos especializados, todos ellos controlados por lógica específica que es necesario gestionar y es justamente a esto lo que los especialistas en sistemas de información denominan reglas de negocio.

Probablemente esta aproximación es la principal responsable de la aceptación a largo plazo que han ganado las reglas de negocio y las tecnologías asociadas a ella. Sin embargo, es muy difícil encontrar una definición detallada de las reglas de negocio. Diferentes personas tienden a ver las reglas de negocio a través de sus propios “prismas” en particular. Según lo planteado en (Chisholm, 2007) algunos hacen hincapié en los aspectos analíticos de las reglas de negocio, mientras que otros piensan en las “tuercas y los tornillos” necesarios para que se puedan implementar en las aplicaciones, incluso los usuarios de negocio con responsabilidades fuera de las TI se están convirtiendo en conocedores de las reglas de negocio y están agregando nuevos puntos de vista al debate (Morgan, 2002), exigiendo que sus puntos de vista sean referencias importantes en la implementación de los sistemas.



En cierto modo, todo el mundo intuye lo que son las reglas de negocio; ya que guía su negocio en la gestión de las operaciones de su día a día. Sin reglas de negocio, se tienen que tomar decisiones sobre la marcha, además de evaluar y elegir caso por caso las alternativas. Hacer las cosas de esa manera sería muy lento. Es probable que se produzcan resultados tremendamente incoherentes y resultaría bastante difícil ganarse la confianza de gran parte de los clientes. Con un enfoque de reglas de negocio, es posible saber en qué solicitud de cualquier pieza de la lógica operacional de la empresa es ejecutada una regla de negocio; dónde se encuentra dentro de estas aplicaciones; cómo se implementa en cada solicitud; quién tiene la responsabilidad de la ejecución, y así sucesivamente.

### 2.2. Reglas de Negocio

El enfoque de reglas de negocio es un mundo real, cuya fuerza motriz es el éxito del negocio, no la tecnología. Surgió de la visión de los profesionales con muchos años de experiencia en las pruebas y los retos de sistemas empresariales. Su objetivo: ofrecer a las empresas la mejor solución posible para el desarrollo de soluciones empresariales que involucren a los sistemas automatizados.

Como se señala en el epígrafe anterior, el término de las reglas de negocios tiene un diverso significado para muchas personas, que puede ser utilizado ampliamente. Sin embargo, Ronald G. Ross en (Ross, 2003) aclara, antes de tomar cualquier conclusión formal del concepto de reglas de negocio, lo que no son las reglas de negocio.

Antes que todo es válido aclarar que las reglas de negocio no es un sistema informático. En muchas ocasiones solo están implementadas dentro de él. El enfoque de implementación tiene dos alternativas:

- *Presente dentro de los procedimientos manuales.* Se refiere al hecho de que las reglas de negocio se encuentran codificadas dentro de los requerimientos que fueron identificados en las aplicaciones informáticas, lo que convierte a esta alternativa, en muchas ocasiones necesarias, pero no muy eficiente a la hora de brindar mantenimiento a la lógica operacional.
- *Utilización de las tecnologías de la información.* Con la utilización de las TI, tales como plataformas para la toma de decisión y sistemas de lógica de negocio, se brindan facilidades a la hora de administrar y mantener la lógica operacional, lo que convierte esto una alternativa más factible.

La idea central, de todo lo expuesto anteriormente, es que las reglas de negocio surgen como un elemento de la empresa y no de ningún hardware especial o plataforma de software.

Las reglas de negocio tampoco son procesos. De manera coloquial en la comunidad de reglas de negocio se está planteando la necesidad de separar el “saber de la corriente” con lo cual se hace referencia al hecho de que el “saber” (entendimiento del negocio que permite definir en cada instante de tiempo y dada una configuración de entrada determinada, cual debe ser la acción que el negocio debe ejecutar como respuesta) es diferente de la “corriente” (elementos de flujo que describen cual es la secuenciación de las actividades que un negocio lleva a cabo para ejecutar sus procesos). Las reglas de negocio representan el “saber” y la “corriente” es la guía de las tareas por separado.

Como ha sido expresado anteriormente existen diversas definiciones acerca de las reglas de negocio, incluso existe un manifiesto publicado en más de 10 idiomas estableciendo las condiciones que estas deben cumplir (Trilles, 2008). Desde mediados de la década del ‘80 del siglo pasado se manejan definiciones formales acerca de lo que representan las reglas de negocios. Daniel Appleton en (Appleton, 1984), introduce académicamente el término por primera vez; definiendo las reglas de negocio como: una restricción explícita al comportamiento empresarial. En 1987 de manera más general aparece otro concepto, publicado por Ronald G. Ross en (Ross, 1987), donde se define una regla de negocio como una política específica para la empresa que restringe o gobierna el comportamiento empresarial y la diferencia del resto. No fue hasta (Ross, 1994) que aparece otro aporte al concepto, agregándole a todo lo anterior que una regla de negocio “puede ser considerado como un requisito del usuario que se expresa en forma no procedimental y no técnica, es decir son expresadas en declaraciones textuales”.

Durante el periodo, 1995-1998, aparecieron otros elementos que enriquecieron y brindaron otros enfoques a las reglas de negocio, tales como:

- “Una declaración o sentencia que define o restringe algún aspecto del negocio. Que pretende hacer valer, influir y/o controlar el comportamiento de la empresa. Una regla de negocio no se puede dividir o descomponer aún en más detalles debido a que se perdería información importante sobre el negocio” (GUIDE Project, 1995).
- En (Ross, 1997), el autor le agrega al concepto descrito en la primera edición del libro que “las reglas de negocio son un término; un hecho que representa un predicado”.

- “Lineamiento empresarial que guía a la empresa en el desarrollo de la política empresarial y es creado en respuesta a los riesgos, amenazas u oportunidades” (GUIDE Project, 1998).

En la presente década, hasta el 2008, se han agregado otros enfoques acerca del mundo de las reglas de negocio, entre las que podemos enunciar:

- “Una pieza atómica de lógica empresarial reusable, especificada declarativamente” (Ross, y otros, 2000).
- “Una sentencia que contiene información de los procesos organizacionales y genera otra información a partir de la misma” (Chisholm, 2001).
- “Condiciones que rigen un evento empresarial que se produce de tal manera que sea aceptable para el negocio” (Von Halle, 2002).
- “Básicamente, una regla de negocio es una formulación sintética sobre un aspecto del negocio. Es una limitación, en el sentido de que se especifica lo que debe o no debe ser para el caso” (Morgan, 2002).
- “Son construidas directamente sobre los términos y hechos. De hecho, una regla, simplemente debe añadir el sentido de permitir o denegar a los términos y hechos que ya han sido definidas en el modelo de hechos y el catálogo de conceptos. En los problemas empresariales que involucren a cientos o miles de reglas - no del todo infrecuente - no hay manera de aportar coherencia entre un número tan grande de las normas sin una base común de términos y hechos” (Ross, 2003).
- “Representan la lógica de las decisiones empresariales” (Taylor, y otros, 2007).
- “Una regla es un ente que está bajo la jurisdicción de los negocios” (OMG, 2008).

El punto de vista acerca de las reglas de negocio que interesa, en este trabajo, es su aplicación práctica en un sistema de gestión empresarial por procesos (BPM<sup>2</sup>, por sus siglas en inglés) y de manera más concreta en su utilización para desarrollar sistemas de información que contribuyan a la gestión de empresas que sigan el enfoque de procesos. Por lo que se puede acercar su definición y alcance a los siguientes enunciados:

---

<sup>2</sup> BPM: Business Process Management

- Se define como regla de negocio toda pieza atómica reusable de la lógica empresarial, que se expresa de manera declarativa, que define o restringe cualquier elemento del negocio y tiene como objetivo hacer valer, influir y/o controlar el comportamiento empresarial.
- Se trata de entes pasivos que contienen información con carácter meramente declarativo y que por tanto no ejecutan nada sino que están a disposición de los motores de reglas de negocio para que éstos, siguiendo lo prescrito en las reglas, procedan a la ejecución de lo previsto en los modelos (Trilles, 2008).
- Deben definirse y mantenerse de manera **independiente** de los modelos y de los procesos con los que la empresa funciona, por lo que definimos un repositorio de reglas con tal propósito. Esto no significa que no tengan conexión con los procesos, si no que mantienen una relación íntima y constante. Sin embargo, su existencia y personalidad se derivan de la propia concepción de la empresa y su misión es definir de manera granular sus políticas y modos operativos (Trilles, 2008).

### 2.3. Componentes de los Sistemas de Reglas de Negocios

En la actualidad todo sistema que se desee construir bajo el enfoque de reglas de negocios, debe tener dentro de su estructura tres componentes, desde el punto de vista de los procesos de negocios. Estos componentes son denominados acorde a (Ross, 2003) en:

1. **Estructura.** Es proporcionada por conocimientos organizados acerca de las cosas más básicas que podemos saber sobre el negocio. Se compone de conceptos básicos de la empresa y las conexiones de base lógica que hacemos entre ellos.
2. **Poder.** Este componente es proporcionado por procesos, que operan en los términos y hechos. Mientras que el modelo de datos proporciona la estructura, y los procesos permiten la actividad.
3. **Control.** El control es proporcionado por las reglas, que limitan los procesos para actuar sólo en la forma que se considera mejor para el negocio.

Luego de la descripción de cada uno de los componentes, brindada anteriormente, se puede concluir que estos componentes son esenciales y sin ellos un sistema que pretenda seguir el enfoque de reglas de negocio literalmente se desintegra.

Por ejemplo, los procesos actúan para guardar información acerca de los elementos representados en el modelo de hechos. Estas acciones están sujetas a reglas de negocio y el éxito del comportamiento empresarial depende de una integración efectiva, es obvio que hay que tener en cuenta estas interrelaciones fundamentales. Cada uno de los tres componentes está especializado en un rol o responsabilidad y es capaz de optimizar su tarea específica. La mezcla o combinación de ellas sería una solución menos eficaz por lo tanto se reconoce la importancia de independizar las reglas de negocio, para lograr, como un beneficio adicional, una gran simplificación en los procesos.

### 2.3.1. Estructura

Los sistemas de negocios, como todo sistema, deben tener una estructura correspondiente. En el enfoque de reglas de negocio, esta estructura se visualiza por medio de un modelo hecho. Para muchos profesionales de las TI, un modelo de hechos se asemeja a un modelo de datos de alto nivel o a un diagrama de clase. Existen elementos de comparación, que son validos resaltar, que diferencian estos modelos, los cuales podemos observar a continuación:

- *Modelo de Datos:* Los utilizan como presentación o base para el diseño de una base de datos o de integración de datos, es decir, como el anteproyecto para la construcción de un almacén de datos. No importa cómo es desarrollado, sino que termina siendo principalmente como una herramienta de desarrollo del sistema. Una vez creada, no es probable que veamos los trabajadores del lado de la empresa haciendo uso continuo del mismo a partir de entonces. Además, aunque muchos sí reconocen la importancia práctica en la creación de un modelo de datos de buena calidad, los modelos de datos en general, no tienden a ser sobre el vocabulario general de negocios. Por lo general, esta práctica generalmente no causa problemas a los modelos dado que no se usa normalmente para apoyar las especificaciones declarativas de las reglas de negocios (Ross, 2003).
- *Modelo de Hechos:* En contraste con los modelos de datos, para llegar al éxito, un modelo de hechos necesita llegar a ser el aparato central de la operativa del negocio. La razón es simple: el modelo de hechos representa el vocabulario básico para la expresión de las reglas de negocio. Aunque un modelo de hechos debe servir como un marco inicial para el diseño del modelo de datos, no implica que lo usemos con tal propósito, si no que el fin de un modelo de hechos es,

literalmente, la estructura de los conocimientos básicos de la empresa. La confección exitosa de un modelo de hecho significa capturar todo el conocimiento del negocio que poseen los trabajadores del negocio, por lo que podemos decir que esta capacidad es fundamental para el planteamiento de las reglas de negocio (Ross, 2003).

Un modelo de hechos representa la base de la estructura para el conocimiento de un sistema de negocio, el cual se compone por dos componentes básicos: los términos y los hechos. Por lo que se puede concluir que estos componentes son las estructuras básicas del conocimiento empresarial.

### 2.3.1.1. Términos

Un *término* es una palabra o frase básica, que los trabajadores del negocio reconocen y comparten entre sí. Un *término* tiene un significado especial para el negocio; debe disponer de un contexto particular inequívoco para su uso y siempre son nombres o sustantivos calificados (Ross, 2003). Como por ejemplo: cliente, factura, envío, fecha de entrega, etc.

Es importante señalar que la palabra especial o frase seleccionada como un término, no constituye sino la punta del iceberg en lo que respecta a su significado. En el enfoque de reglas de negocio, una definición precisa para cada término, debe ser expresada orientada al negocio. Todos los hechos y reglas que referencian al término dependerán de su significado. Para incluir un término dentro de un modelo de hechos, el mismo debe satisfacer todos los siguientes puntos:

- *Básico*: el término dentro del modelo de hecho representa lo más básico definido por el negocio, es decir, no se puede derivar de otros términos. Cualquier término que se derive de otro, debería ser expresado como una regla de negocio (Ross, 2003).
- *Atómico*: los términos dentro del modelo de hecho no son divisibles, es decir no son compuestos, al menos es como se define desde el punto de vista del negocio. Los términos que tienen un sentido colectivo, por ejemplo la mercancía, el inventario etcétera, deben desglosarse en sus componentes atómicos para su representación en el modelo de hechos (Ross, 2003).
- *Cognoscible*: los términos siempre deben representar elementos de los que se conoce en lugar de términos a los que le sucede algo. En otras palabras, un modelo de hechos es acerca del

conocimiento, no de la acción, ni procesos o procedimientos que no aportan conocimiento (Ross, 2003).

En el enfoque de reglas de negocio, la colección de todos los términos y definiciones se llama Catálogo de Conceptos. En cierto sentido, esto no es más que un nombre disfrazado de glosario. Dado que las definiciones son tan cruciales para organizar grandes conjuntos de reglas, este glosario debe ser automatizado, para dar la posibilidad de modificar los términos que aparecen directamente en todas las reglas. Utilizando un catálogo de conceptos podemos garantizar el éxito de la aplicación de un principio fundamental en el enfoque de reglas de negocio: sería más eficaz, si inevitablemente, todos los trabajadores del negocio hablaran el mismo idioma. Se puede concluir, al respecto, que cada término requiere de una definición y cada definición pertenece a un catálogo de conceptos, logrando que un modelo de hechos establezca un vocabulario común empresarial.

### 2.3.1.2. Hechos

Los hechos son sentencias simples de manera declarativa que se relacionan con un determinado término. Muchas observaciones se han hecho respecto al tema, como por ejemplo:

1. Los hechos representan verbos comunes o frases verbales dentro del dominio empresarial. En otras palabras, un modelo de hechos extiende el vocabulario común empresarial de diversas maneras importantes.
2. Los hechos estructuran las conexiones lógicas que se hacen entre los conceptos.
3. Un modelo de hechos establece las bases del conocimiento empresarial que debe ser compartido.
4. Los hechos reconocen lo que es posible "conocer" sin especificar otras restricciones
5. El modelo de hechos organiza el conocimiento básico acerca de los resultados de las operaciones, no acerca de cómo realmente estas actividades son ejecutadas.

Si en un momento se le pregunta a los trabajadores del negocio sobre que entienden acerca de requerimientos para el desarrollo de sistemas de negocios, por lo general se obtienen respuestas centradas en las funciones que se ejecutan o en el aspecto de cómo el sistema se comporta a través de las interfaces visuales, sin llegar a la respuesta de "términos y hechos", que de hecho, son una especie de requerimientos. No son el único tipo de requerimientos dentro del enfoque de reglas de negocios, pero sin

embargo sin ellos, no se puede proporcionar un significado real o coherente a todos los demás, especialmente a las reglas. Por esa razón, los términos y los hechos, representan el tipo de requerimiento más fundamental. Vamos agregar otro punto aquí, un modelo de hechos puede y debe proporcionar un primer corte de cómo los datos serán organizados dentro de la base de datos.

### 2.3.1.3. *El modelo de hechos y el modelo de datos*

Como se dijo anteriormente el modelo de hechos se asemeja a un modelo de datos de alto nivel, pero existen diferencias entre los mismos, las cuales se listan a continuación:

- *Es categoría de.* Los hechos del tipo “es categoría de”, se representan a menudo de manera implícita en un modelo de hechos. El enfoque de modelos de datos varía en su tratamiento con respecto al modelo de hechos, del que se originó. En algunos casos se representan como subtipos de alguna entidad de negocio u objeto de datos y en otros, los subtipos no se reconocen en absoluto. En cualquiera de los casos existen dos distinciones entre ambos modelos que deben ser considerados (Ross, 2003).
  1. Debido al enfoque de amplio vocabulario que aparece en los modelos de hecho, los hechos “es categoría de” por lo general se extienden mucho más que la tipificación que se aplica en los modelos (Ross, 2003).
  2. En el modelado de hechos, no existe ninguna hipótesis acerca de cómo representan físicamente estos tipos de hechos. En los modelos de datos, en cambio, el uso de subtipos es a menudo moldeado por la suposición acerca de la representación física de los hechos “es categoría de” (Ross, 2003).
- *Tiene un.* Los hechos de este tipo también son representados de manera implícita en los modelos de hechos. Para evitar confusión, estos tipos de hechos en ocasiones son omitidos de la gráfica del modelo de hechos, a menos que estén implicados dentro del núcleo de las reglas de negocio o en las operaciones computacionales de los sistemas de negocio. En cierto sentido, los hechos “tiene un” representan detalles que pueden ser mejor manejados en otro lugar, por ejemplo en un repositorio de reglas de negocio, incluso él nunca puede expresarse en forma de sentencias. Sin embargo, es importante reconocerlas como otro tipo de hecho bajo el enfoque de reglas de negocio. Los modelos de datos tratan a los “tiene un” como una representación de atributos de



alguna entidad de negocio o objeto de datos. Dependiendo del enfoque, estos atributos pueden o no ser mostrados explícitamente en el modelo de datos (Ross, 2003).

- *Cardinalidad y opcionalidad.* Los modelos de hechos no corresponden a ciertas clases de especificaciones que pueda tener acostumbrados a los modeladores. Por ejemplo ni la cardinalidad ni la opcionalidad son representados de manera explícita. Estos hechos explícitos corresponden bruscamente a las relaciones en los modelos de datos. Algunas técnicas del modelado de datos, incluyendo por lo general algunas indicaciones las cardinalidad de estas relaciones. La cardinalidad es importante en el diseño de base de datos, sin embargo, cualquier tipo de cardinalidad para muchas personas es una forma de reglas de negocio. Por ejemplo, considere la declaración de un pedido realizado por más de un cliente. En realidad, esta declaración representa una regla expresada para el hecho de que un pedido es solicitado por un cliente. Lo mismo se aplica para la opcionalidad, es decir, indicar que una relación debe ser mandataria u obligatoria estamos en presencia de una regla de negocio. El enfoque de reglas de negocio destaca que la elaboración de términos y hechos es sumergirse profundamente en la expresión de cualquier forma de reglas, incluyendo las relativas a los rectores de cardinalidad y opcionalidad. Más encima, todas las reglas de negocio deben constar de forma explícita para que sus motivaciones empresariales puedan ser revisadas aparte del sistema o los diseños de los datos (Ross, 2003).

Un buen modelado tiene la intención de garantizar que cada término o hecho sea representado en el modelo de hechos una sola vez, y que no se superponga a ningún otro término o hecho; en otras palabras el modelo de hechos se cerciora de que cada término o hecho sea unificado y único, luego esto proporcionaría un camino para garantizar que todas las reglas de negocio sean definidas consistentemente y que las diferentes acciones sean operadas consistentemente. Los términos y hechos de un modelo de hechos deben representar un conjunto necesario para proporcionar un marco adecuado para otros componentes, algunos términos y hechos extras no ayudan al buen desempeño del sistema empresarial, en otras palabras, un modelo de hechos ayuda de que no existan términos ni hechos adicionales dentro del conocimiento básico empresarial.

### 2.3.2. Poder

En el enfoque de reglas de negocio debe existir un componente encargado del poder para los sistemas de negocio, este poder que produce el movimiento del sistema es proporcionado por los procesos de negocio. Por lo general un proceso de negocio toma como entrada lo que es proporcionado por el usuario final y lo transforma en una salida deseada, funcionando estos acorde a un algoritmo construido por los diseñadores o programadores de los sistemas de negocio. El único problema a todo esto es que con los enfoques tradicionales los procesos de negocio no son tan simples, se tornan bastante complejos y difíciles de cambiar.

El enfoque de reglas de negocio ofrece una poderosa ayuda para los procesos, en esencia toma las reglas separadas de los procesos de negocio, lo que lo hace más simple en otras palabras se puede decir que el enfoque de reglas de negocio genera procesos que son más simples de gestionar. Esta simplificación tiene repercusiones mucho más allá del alcance del negocio, de hecho esto puede cambiar toda la naturaleza del trabajo empresarial, por lo que se puede decir que las reglas de negocio revolucionan el trabajo en la empresa como por ejemplo en el siguiente caso. Los modelos de desarrollo basados en reglas de negocio siguen un enfoque de

- *Procedimientos basados en términos y hechos*, con el objetivo de proporcionar a los desarrolladores un modelo de hechos y un catálogo de conceptos. Estos procedimientos deben cumplir con los términos y hechos predefinidos con anterioridad, esto significa que los procedimientos siempre deben reutilizar estos términos y hechos en vez de desarrollar sus propias versiones. Así que el primer y más básico aspecto de cómo organizar el trabajo dentro del enfoque de reglas de negocio es el principio de que un procedimiento nunca define datos para sí mismo sino que intenta compartirlos.
- *Procedimientos basados en la independencia de las reglas*. El enfoque de reglas de negocio cambia toda la manera de pensar, al reconocer que las reglas de negocio son parte del “saber”, y que como toda parte del “saber” no deben estar incluidas en el “hacer”, este hecho se conoce como independencia de las reglas de negocio. Solo una parte relativamente pequeña de los sistemas tradicionales, apoyan literalmente los pasos actuales de los procedimientos, es decir el “hacer”, desde una perspectiva empresarial. La mayoría del código está dedicado a la edición,

validación y cálculos para apoyar las reglas de negocio. En otras palabras, cuando se toman las reglas fuera de la lógica tradicional se obtiene como resultado un *proceso simple*. Este proceso simple no es más que un procedimiento que describe el conjunto de pasos necesarios para lograr un resultado deseado y excluye todas las reglas y manipulaciones de errores cuando una regla es violada (Ross, 2003).

En el enfoque de reglas de negocio, a la secuencia de pasos para lograr el resultado deseado, se le denomina *guión*, que en ciertas ocasiones se les llama simplemente “script” o “una serie prescrita de peticiones”, las cuales no incluyen las reglas de negocio, ni la manipulación de las violaciones de estas reglas, en esencia, un guion consiste en una serie de peticiones sin reglas embebidas. Cuando se habla de que es “una serie prescrita de peticiones”, se entiende por “*prescrita*” para que la serie de pasos se puedan seguir para lograr los resultados deseados, pero no de carácter obligatorio. Diciendo que es de carácter obligatorio, se estaría en presencia de una regla de negocio sobre la secuencia, la cual no debería ser empotrada en el proceso o procedimiento. Por otro lado, las “peticiones” se refieren a las peticiones de acción, es decir, cuando se realiza la petición a algún componente de aplicación, esta acción puede ser tomada de forma que se va a ejecutar algún programa. Estos componentes se podrían incluir dentro de cualquiera de estas categorías:

- Sistemas de gestión de base de datos.
- Componentes de interfaz de usuarios.
- Suministradores de servicios, tales como las rutinas de impresión.
- Interfaces con sistemas heredados.
- Tecnologías para reglas de propósito específico

No se puede decir que las secuencias de pasos o guión solamente responden a solicitudes entre los componentes, en muchos aspectos la fuente más importante o destinatarios de las solicitudes son las personas reales. Estas personas pueden ser o bien dentro de la empresa, es decir, los trabajadores o fuera de la empresa, por ejemplo, los clientes. Aunque todas estas personas podrían ser “usuarios” del sistema de negocio, para las secuencias de pasos se toma la expresión de “agente”. El usuario representa un punto de vista fuera de los beneficiarios de los servicios del sistema, debido a que su obra e interacciones están fuera del alcance del sistema. El actor, por el contrario, sugiere a alguien cuya propia

actividad o función es esencial para entender y hacer el trabajo, además es alguien cuyo trabajo se encuentra, sin duda, dentro del ámbito del sistema.

Claro que los actores para realizar su trabajo, pueden ejecutar acciones, realizar peticiones para la acción, peticiones a los componentes de la aplicación y de manera indirecta o directa a otros actores. Esto comienza aclarar más la nueva imagen del trabajo en el enfoque de reglas de negocio, puesto que el trabajo se realiza como actores humanos y/o agentes del sistema que interactúan entre sí, siguiendo las secuencias de comandos. En el proceso simple, se hace hincapié en la colaboración entre los actores, tanto humanos como máquina, en fin, las reglas de negocio adaptables permiten involucrar a las personas y a las máquinas.

No todo el trabajo debe ser escrito con antelación; de hecho, en un entorno empresarial dinámico, no todos los eventos y circunstancias se pueden predecir, y mucho menos su indicación. Este hecho tiene varias implicaciones importantes.

- El acceso a la base de datos siempre será un factor significativo para los sistemas empresariales. Esta es una razón más del por qué es tan importante definir términos, hechos, y las reglas de negocio de forma independiente a los procesos y procedimientos (Ross, 2003).
- Los criterios deben ser identificados para determinar cuándo puede que no sea rentable la redacción de guiones de antemano, incluso para cuando algún evento se puede predecir. Por ejemplo, las secuencias de comandos de trabajo para eventos de baja frecuencia realizada por sólo unos pocos actores, a menudo no son rentables (Ross, 2003).

La construcción de secuencias de comandos para el trabajo predecible y repetitivo, es bastante útil, las colaboraciones que prescriben no tienen por qué ser estáticas, sino, más bien, pueden ser dinámicas y en constante evolución. Esto es exactamente lo que las empresas del siglo 21 necesitan para afrontar el reto del cambio rápido. Este enfoque es coherente con el pensamiento actual acerca de cómo debe ser utilizado para transformar las estructuras organizativas, en la actualidad la idea de profundas jerarquías, con muchas capas intermedias de gerencia, están fuera del pensamiento, sino que se sigue la idea de jerarquías aplanadas, con los usuarios finales y los patrones flexibles de colaboración.

Otros centros de negocios actuales tienen la tendencia de la automatización de la cadena de valor extendida, cruzando fronteras organizacionales entre proveedores, productores y clientes. El objetivo real es la compactación de la cadena de valor, lo que permite una interacción directa y dinámica entre los actores. Es interesante señalar que las tendencias mencionadas implican la eliminación de ciertos tipos de roles organizativos. En el caso de la reingeniería, el objetivo ha sido a menudo los gerentes de nivel medio. En el caso de las cadenas de valor compactado, el objetivo son los intermediarios en general, y con ese enfoque, el objetivo es la desintermediación. Todos estos roles en tiempos pasados sirvieron no sólo para filtrar la información, sino también para conocer y hacer cumplir las reglas de negocio. Tener un libro de reglas de negocio en la empresa, es muy ventajoso porque de lo contrario, ¿cómo puede la empresa comunicar su normativa y velar que el comportamiento de todos los actores sigue siendo coherente con los objetivos del negocio? La importancia de este principio es objeto de mayor atención.

La reutilización de las secuencias de comando, de modo normal, implica la reutilización de muchos componentes de la aplicación como un hecho, sin embargo, el objetivo es la reutilización a nivel de negocio. Centrarse exclusivamente en la reutilización a nivel de aplicación pierde esta amplia perspectiva. Por lo que podemos decir que el enfoque de reglas de negocio se centra en la reutilización a nivel de negocio, no sólo en la reutilización a nivel de aplicación. La reutilización de secuencias de comando es muy utilizada en la construcción de sistemas de negocio, la manera más obvia de la reutilización que se conoce es el hecho simplemente de invocar un procedimiento o secuencia de comandos dentro de otra. Esta manera de reutilización de procedimientos es muy importante debido a:

- La aplicación de apoyo no necesita ser reescrita, pero puede ser reutilizada tal y como es, posibilitando el ahorro de tiempo en el desarrollo y mantenimiento (Ross, 2003).
- La reutilización de cualquier tipo de procedimiento produce la coherencia, y a su vez aumenta la productividad en general (Ross, 2003).
- Un trabajador que ya conoce la secuencia de comandos invocada no es necesario que aprenda algo nuevo para hacer esa parte de la obra (Ross, 2003).

La reutilización anormal implica que una secuencia de comandos que se utiliza en circunstancias normales también se use en circunstancias anormales. La reutilización anormal es la pieza final y crucial en el replanteamiento del trabajo para las empresas del siglo 21. Para entender la reutilización anormal,

es importante entender exactamente cuando ocurren las circunstancias anormales. En muchos enfoques, definir dónde y cuándo se producen las circunstancias anormales es un misterio. El enfoque de reglas de negocio, en cambio, es muy preciso en este aspecto: *las circunstancias anormales de negocios se producen cuando y sólo cuando un trabajador presenta una solicitud que resulta de una violación de una regla*. Por lógica, si no se infringe ninguna norma, las circunstancias no son anormales. Las reglas de negocio siempre definen el umbral que separa las circunstancias comerciales normales y anormales.

Aquí existe un principio general para el diseño de reutilización anormal de las secuencias de comandos y es el hecho de que cualquier secuencia de comandos invocada por una regla de negocio en circunstancias anormales debe ser una secuencia de comandos utilizada para el mismo tipo de trabajo en circunstancias normales. Se debe agregar que la secuencia de comandos seleccionada para las circunstancias anormales se pueden modificar independientemente de la secuencia de comandos en condiciones normales, y todos estos cambios pueden ocurrir independientemente de la regla de negocio, es decir, que se puede lograr sin ningún impacto en la propia regla. De esta forma, la reutilización anormal alcanza todo el potencial de las reglas de negocio para la organización del trabajo en las empresas del siglo 21. Nada de esta capacidad de adaptación habría sido posible, por supuesto, si la regla de negocio se hubiera incrustado en la propia secuencia de comandos. Aquí, la independencia de la regla aporta otro beneficio importante: *un nuevo tipo de seguridad para limitar el impacto de los cambios al procedimiento*.

### 2.3.3. Control

Todo sistema de reglas de negocio debe tener un componente encargado del control del comportamiento empresarial, que bajo este enfoque, el mismo es proporcionado por las reglas de negocio. El enfoque de reglas de negocio no sólo depende de las buenas reglas, sino también ofrece nuevas perspectivas sobre lo que son.

El primer paso para comprender el papel central del enfoque de reglas de negocio consiste simplemente en que ellas se refieren a la cuestión del control. La tabla que se muestra a continuación, presenta una muestra de las reglas de negocio típicas, categorizadas según el tipo de control que aborda, que en realidad muestra el amplio rango de estas categorías de control. Todos los aspectos de control en las operaciones empresariales pueden ser reguladas por las reglas de negocio (Ross, 2003).

Tipo de Control	Regla de Negocio
<i>Restricción</i>	Un cliente no debe hacer más de tres pedidos urgentes con cargo a su cuenta de crédito.
<i>Heurística</i>	Un cliente con el estado de preferencia se le debe cubrir inmediatamente las órdenes.
<i>Computacional</i>	El volumen de pedidos anuales de un cliente debe ser calculado como total de ventas cerradas durante el año fiscal de la empresa.
<i>Inferencia</i>	Un cliente debe considerarse preferido, si el mismo hace más de cinco pedidos de más de \$1.000.
<i>Temporizador</i>	Un cliente debe ser eliminado si realiza ningún pedido por 36 meses consecutivos.
<i>Disparador</i>	"Enviar aviso previo" debe ser ejecutado por una orden cuando se envíe.

**Tabla 1. Ejemplo de Tipos de Control**

El segundo paso, crucial además, en la comprensión de las reglas de negocio es entender cómo se relacionan con términos y hechos. En el enfoque de reglas de negocio, las reglas se construyen directamente sobre los términos y hechos. De hecho, una regla, simplemente debe añadir el sentido de permitir o denegar a los términos y hechos que ya han sido definidos en el modelo de hechos y el catálogo de conceptos. En los problemas empresariales que involucren a cientos o miles de reglas, no del todo infrecuente, no hay manera de aportar coherencia entre un número tan grande de las normas sin una base común de términos y hechos. El tercer paso, también crucial, es comprender cómo las reglas de negocio se refieren a los acontecimientos o hechos (Ross, 2003).

Apreciando la importancia del principio de que las reglas de negocio deben ser percibidas y expresadas de forma declarativa, independiente de los procesos y procedimientos, se pueden desarrollar sistemas exitosos sobre este enfoque. Inevitablemente este principio, nos aleja del punto de vista de los requisitos para los sistemas de negocios y nos acerca hacia ver como las reglas de negocio son un problema real

para las empresas. La comprensión de este principio fundamental del enfoque de reglas de negocio exige un estudio detenido de la relación entre las normas y los eventos. Intuitivamente, sabemos que las reglas de negocio deben aplicarse cuando se producen determinados eventos, de lo contrario no sucede nada, las reglas no pueden ser violadas, y por lo tanto pueden permanecer en estado latente.

En primer lugar, es importante reconocer que las reglas de negocio y los eventos no significan lo mismo. Esto puede parecer obvio, pero no deja de ser una fuente común de confusión. Para entender esto, debemos identificar que es un evento. Hay por lo menos dos maneras de entender su significado, ambos correctos.

- *Desde el punto de vista empresarial:* Para el negocio, un evento es cuando sucede algo que requiere que la empresa responda, aunque sea de una manera trivial. (Por lo general, la respuesta no es trivial.) Por ejemplo, un cliente puede realizar un pedido. Este es un evento que requiere una respuesta bien organizada. A menudo tratamos de organizar nuestra respuesta a los eventos empresariales, de antemano, por ejemplo, con modelos de flujo de trabajo, procedimientos, scripts, etc. (Ross, 2003).
- *Desde el punto de vista de las TI:* Para los sistemas de negocios, un evento es cuando sucede algo que necesita ser observado o registrado porque el saber sobre el evento es potencialmente importante para otras actividades, ya sea los que se producen durante el mismo período de tiempo o los que podrían ocurrir más adelante. En el enfoque de reglas de negocio, por supuesto, las condiciones se basan siempre en términos y hechos predefinidos, es decir, principalmente sobre la base del modelo de hechos. Un sistema de negocio puede apoyar el modelo de hechos de varias maneras, por ejemplo, con un diseño de base de datos, un diagrama de clase, y así sucesivamente. Para simplificar, digamos que existen algunos datos en alguna parte del sistema de negocio que deben ser actualizados para registrar el evento. De lo contrario, el negocio no puede saber sobre el evento (Ross, 2003).

¿Cómo conectar las reglas de negocios a los eventos? Podemos realizar la conexión entre estos dos elementos si la regla de negocio por si sola se ha expresado de manera declarativa, entiéndase con esto que en ella no se indica ningún procedimiento o proceso en particular para hacerla cumplir, es simplemente una regla y nada más que eso. Declarativa también significa que la regla de negocio no hace referencia a cualquier evento de negocio o evento de actualización donde pueda ser violado y por lo tanto



habrá que examinarla. En general, todas las reglas en forma declarativa producen dos o más tipos de eventos de actualización, en caso de ser violadas.

En el caso de violar alguna regla de negocio podrían pasar al menos dos cosas como se muestra a continuación:

- No importa de qué evento se trate, la regla debe ser disparada de manera tal que la restricción prevista sea aplicada (Ross, 2003).
- Suponiendo que la regla de negocio tiene por finalidad evitar errores, debería mostrarse un mensaje al usuario final, quien no es más que el trabajador del negocio, para explicar por qué ocurrió la violación (Ross, 2003).

El mensaje que se debería mostrar al usuario final, no es más que la misma sentencia con que fue construida la regla de negocio. Para poner esto en mayor medida, en el enfoque de reglas de negocio la declaración de la regla es el mensaje de error. La arquitectura de los sistemas de negocio debe considerarse ante todo como un problema de negocio, no como un problema técnico. Este tratamiento de mensajes de error en el enfoque de reglas de negocio es compatible con ese objetivo en una manera fundamental y además tiene importantes implicaciones para el proceso de presentación (front-end) de la recopilación de requisitos, como se comenta en los elementos que aparecen a continuación:

- En el enfoque de reglas de negocio, los principales mensajes de error que los usuarios finales verán una vez que el sistema está en funcionamiento serán las mismas definiciones de reglas de negocio que algunos trabajadores bien capacitados del lado comercial dieron como requisitos en el diseño anterior del sistema empresarial. Los mensajes de error y los requisitos son literalmente la misma cosa. Si durante el proceso de captura de requisitos se expresan en buen sentido las reglas de negocio presentes en el negocio se obtendrán mensajes de error bien expresados; de lo contrario se expresarán mal los mensajes de error (Ross, 2003).

Vale la pena hacer varias observaciones acerca de este principio. En primer lugar, la asistencia directa en la expresión de las normas será extremadamente valiosa para los trabajadores afectados. Lo cual es una habilidad importante para el analista de negocios.

- En segundo lugar está la posibilidad de cerrar la brecha entre los requisitos de la parte del negocio y la parte que sigue afectando a tantas empresas. En los enfoques tradicionales se pierde mucha expresividad en la transformación de los requisitos iniciales para el sistema de funcionamiento real. En el enfoque de reglas de negocio, los especialistas del negocio pueden gestionar cualquier cosa que hayan definido, como debería ser en sistemas dirigidos por el negocio (Ross, 2003). En esencia el enfoque de reglas de negocio ayuda a eliminar la brecha de los requerimientos.

En el aspecto técnico, descubrir y apoyar los eventos de actualización se convierte en una preocupación fundamental de la aplicación. Afortunadamente, la tecnología de la lógica de negocios a menudo puede hacer esto automáticamente, y brindar un gran impulso a la productividad en la construcción de software de negocios confiable. En fin, el apoyo automático para la identificación y el apoyo a los eventos de actualización de las reglas de negocio es una característica importante para la tecnología de lógica de negocios.

El énfasis en las reglas y su separación de eventos abre muchas nuevas puertas de oportunidades para las TI y el negocio. Entre estas posibilidades son las siguientes:

- *Simple coherencia*: cuando una regla de negocio es violada por dos o más eventos de actualización que probablemente estén incorporados en al menos dos procesos o procedimientos diferentes. Sin embargo, para todos ellos, sólo hay una sola regla. Esa misma regla, en caso de ser violada, debe disparar los eventos de actualización en cualquiera de los procesos o procedimientos en los que está presente. De este modo, el enfoque de reglas de negocio garantiza la total coherencia en los criterios de edición, aplicada a través de todos estos procedimientos (Ross, 2003).
- *Adaptabilidad*: consiste en la separación de la regla de negocio a partir de los procesos y procedimientos en los que se puede violar la propia regla permitiendo la especificación en un solo lugar de la misma. Este tipo de especificación tiene la ventaja de encontrar la regla de una manera más fácil y cambiarla rápidamente, una vez que el sistema de negocios se aplica (Ross, 2003).
- *Reingeniería*: los procesos de negocio y los procedimientos son organizados generalmente como respuesta a eventos de negocios. Las reglas declarativas, sin embargo, se especifican con

formalismos carentes de eventos. Esta definición abre totalmente nuevos horizontes para los procesos de reingeniería de negocios (Ross, 2003).

Las reglas de negocio se caracterizan en tres variedades básicas con respecto al tipo de control que ejercen: las que rechazan, las productoras, y las proyectoras. Todas las reglas se dividen en una de estas tres categorías funcionales, que se discuten brevemente a continuación.

- *Rechazadoras*: Muchas reglas de negocio son de este tipo por naturaleza, es decir, simplemente rechazan cualquier evento originado por la violación. La secuencia específica de la actividad se producirá más o menos como sigue (Ross, 2003):
  - Un usuario final inicia un procedimiento para hacer un trabajo, por ejemplo, tomar una solicitud de un cliente.
  - La actividad del usuario final produce un evento de actualización para registrar los resultados del trabajo, por ejemplo, crear una instancia de solicitud.
  - Un evento de actualización provoca que se dispare una regla, por ejemplo, una orden debe tener una dirección de envío.
  - La regla comprueba si el usuario final ha dado efectivamente una dirección de envío del pedido. Supongamos que el usuario final no lo ha hecho.
  - La regla de ser rechazada hace que el evento de actualización también se rechace, es decir, si la acción fracasa, la orden o solicitud del cliente no se crea.
- *Productoras*: Este tipo de regla no rechaza los eventos de actualización. Por el contrario, siempre los acepta, se calcula de forma automática o se deriva algo para el usuario final. En el ejercicio de este cálculo el productor utiliza todos los datos pertinentes, ya sean antiguas o nuevas. Por ejemplo, en la adopción de una orden, una regla de tipo productoras podría volver a calcular el total del pedido actual cada vez que se agrega una nueva línea para la prescripción. Las productoras son en realidad funciones definidas de manera declarativa y parecen mucho más amigable que las que rechazan porque no inhiben los eventos de actualización, sino más bien proporcionan un kilometraje adicional. Su objetivo general es impulsar el usuario final y la productividad de programación (Ross, 2003).
- *Proyectoras*: Una regla de negocio perteneciente a este tipo es exactamente lo contrario de un reflector en una manera fundamental. En concreto, una regla de tipo proyector no rechaza los

eventos de actualización. En cambio, como las productoras, siempre los acepta y toma automáticamente algunas otras acciones como resultado. En general, una regla proyectora se puede expresar como “si esto, entonces eso también”. En otras palabras, una proyectora, literalmente proyecta de "esto" a "eso". En realidad hay muchos tipos de eventos automáticos que los proyectores pueden causar, pero entre las más importantes son los siguientes (Ross, 2003):

- *Inferencia*: Inferir hechos nuevos, es decir, el conocimiento de los hechos existentes, de asistencia automatizada en la toma de decisiones.
- *Disparadores*: Ejecutar los procesos y procedimientos de forma automática.

Al igual que las productoras, las proyectoras parecen mucho más amigables que las que rechazan porque no inhiben los eventos de actualización, sino más bien brindan un kilometraje adicional. Al igual que las productoras, las proyectoras proporcionan un comportamiento automático de valor agregado, es decir, eliminan algún comportamiento brindado como una responsabilidad del usuario final, además que brinda asistencia automatizada para realizar trabajos operativos.

Cuando se presenta el enfoque de reglas de negocio, la primera reacción de algunas personas es que su negocio tiene muchas más excepciones que las reglas en sí. Se preguntan cómo todas estas excepciones se pueden controlar de forma organizada. Esta es una preocupación válida. El enfoque de reglas de negocio no ofrece ninguna solución mágica al problema de negocio de tener demasiadas excepciones a las reglas. Lo que hace, sin embargo, es ofrecer una respuesta técnica muy simple. Esta respuesta técnica, a su vez, tiene implicaciones para el proceso de reingeniería de negocios y para optimizar las operaciones de negocios, en fin, la conclusión es que la excepción a la regla no es más que otra regla.

El hecho de reconocer que las excepciones a las reglas de negocio sólo representan más reglas, implica una razón esencial para una reingeniería de negocios, que afecta a las necesidades de desarrollo, el proceso de negocio, y, de hecho, en el propio negocio. Simplemente la cuestión es que todas las reglas representan un costo. El costo de las reglas de negocio no es simplemente el costo directo de su aplicación y mantenimiento de sistemas de negocio. El coste real a menudo se esconde en la documentación asociada, formación, administración, y el tiempo que se necesita para comunicar las reglas y el tiempo que se tarda en cambiar las reglas. En el negocio del siglo 21, por supuesto, el tiempo es uno de los aspectos más preciados de los productos básicos.

Para concluir este debate, vamos a revisar la cuestión de cómo las reglas proporcionan un control en el enfoque de reglas de negocio. En un sistema de reglas de negocio, las reglas se conectan a los procesos, que no son más que la fuente de poder para el comportamiento. Las reglas controlan a los procesos y, por este medio, al comportamiento resultante. Sin embargo, como hemos visto, las reglas de negocio en realidad no deben ser empotradas dentro de los procesos.

Los procesos o procedimientos se conectan a las reglas de negocio a través de eventos de actualización. Cuando un proceso o procedimiento intenta satisfacer algún evento de negocio, el evento de actualización correspondiente se producirá. Esto puede disparar una o más reglas, que determinará si el evento se lleva a cabo correctamente o se produce algún que otro resultado. Dependiendo del resultado, la acción apropiada será tomada.

El control se exterioriza de los procesos o procedimientos y se establece en una capa separada de las reglas o en un componente. Esto permite la gestión directa de las reglas, que a su vez permite mucha más vinculación a la parte comercial. Esta idea de la independencia de las reglas de negocio es una pieza central del enfoque de reglas de negocio.

### 2.4. Motor de Regla de Negocio

#### 2.4.1. Historia

Según (Vera, 2010), en el año 1975 aparece el primer sistema experto, programa que mediante una serie de reglas preestablecidas, comparación de resultados y aplicación de nuevas reglas de acuerdo a dicho resultado, facilita la solución a problemas que requieren un gran conocimiento sobre un tema específico. Para el año de 1984, acorde a (Vera, 2010), el departamento de investigación en Inteligencia Artificial de la agencia espacial norteamericana (NASA, por sus siglas en inglés) había implementado varios prototipos de sistemas expertos, sin embargo, casi nadie los utilizaba porque habían sido desarrollados en LISP<sup>3</sup>; que en aquél entonces este lenguaje incluía un alto costo de implementación, dificultad para integrarlo con otras aplicaciones y poco hardware con la capacidad para ejecutarlo. Entonces, en dicho departamento,

---

<sup>3</sup> Lenguaje de programación funcional que es muy útil en el campo de la Inteligencia Artificial. Su nombre es derivado del LISP Processing (Proceso de listas).

se decidieron por crear un lenguaje propio para sistemas expertos que denominaron CLIPS<sup>4</sup>. Luego, según el autor de (Vera, 2010), a mediados de 1995, el Dr. Ernest Friedman-Hill, investigador del Laboratorio Nacional de Sandia, desarrolló la herramienta JESS<sup>5</sup>, un motor de inferencia basado en CLIPS y el primero programado en Java.

Con herramientas disponibles públicamente para construir y ejecutar sistemas expertos, muchos departamentos de TI y unidades de negocio reconocieron la posibilidad de crear aplicaciones basadas en conocimiento sin la necesidad de adquirir software comercial. Igualmente, por las fuerzas del mercado, muchas empresas surgieron con el fin de crear sus propios motores para implementar soluciones a la medida.

Con el pasar del tiempo, la competencia entre diferentes proveedores provocó que muchos de éstos incluyeran características adicionales: manejo de bases de datos para almacenar las reglas, editores gráficos para representarlas, flujos de trabajo para publicarlas, etcétera. Poco a poco fueron evolucionando de simples herramientas de parseo y ejecución de reglas a complejos sistemas para administrar su ciclo de vida, incluyendo monitoreo y detección de errores. Así surgieron los ahora denominados Sistemas de Gestión de Reglas de Negocio (Business Rules Management Systems, BRMS por sus siglas en inglés) (Vera, 2010).

Durante los últimos años se ha encontrado un uso muy particular a los BRMS, ahora también denominados motores de reglas de negocio los cuales permiten encapsular la lógica de negocio o base de conocimientos y separarla del código aplicativo. Elemento que es común a la mayoría de las definiciones que se manejan de este elemento tanto a nivel académico como industrial:

- Una aplicación que contiene la definición de las reglas de negocio (Ross, 2003).
- En su forma más simple, un motor de reglas interpreta y ejecuta una serie de enunciados, es decir las reglas de negocio, y de acuerdo a las diferentes validaciones genera un resultado (Vera, 2010).
- Es un componente que, a partir de una información inicial y un conjunto de reglas, detecta qué reglas deben aplicarse en un instante determinado y cuáles son los resultados de esas reglas (Planeta Código, 2005).

---

<sup>4</sup> Sistemas de Producción Integrado en Lenguaje C. Por sus siglas en inglés “C Language Integrated Production System”.

<sup>5</sup> Interfaz de Sistemas Expertos en Java. Por sus siglas en inglés “Java Expert System Shell”.

### 2.4.2. Cuando usar un motor de reglas de negocio

Los motores de regla de negocio se han utilizado tradicionalmente en aplicaciones financieras para satisfacer requisitos complejos asociados a la puntuación de crédito y suscripción, debido a que estas aplicaciones utilizan un gran número de complejas reglas de negocio. La alternativa consiste en desarrollar esas reglas en complejas capas de implementación plagadas de sentencias condicionales y tablas de consulta sobre la que descansa toda la aplicación (Moran, 2004). Una alternativa viable hace unos años, cuando los desarrolladores permanecían inmovibles en sus puestos de trabajo prácticamente de por vida y podían transmitir estos detalles a las nuevas generaciones, pero algo imposible en el dinámico y convulso entorno de este siglo donde nadie se planifica a cinco años en el futuro. Los motores de reglas de negocio ofrecen un marco para el aislamiento de la lógica empresarial en sus aplicaciones; este marco es más simple y más flexible, pero lo más importante, es alcanzar el difícil objetivo de reutilizar el código y mantener seguras las capacidades de la empresa de gestionar la lógica del negocio sin necesidad de triplicar los gastos para el mantenimiento de las aplicaciones.

La diferencia ocurre cuando el usuario que ayudó a especificar un sistema, quiere cambiar las reglas del negocio, que normalmente están como parte del código. El proceso de realizar el cambio, tiene que pasar desde la fase de compilación, empaquetamiento y liberación a producción, resultando en que cualquier cambio a las reglas, por mínimo que sea, no es cuestión de minutos, sino de horas o días inclusive.

El objetivo del motor de reglas de negocio, es proporcionar una infraestructura desacoplada del código fuente, de tal manera, que el sistema pida servicios al motor, para evaluar una regla. En un repositorio residen las reglas que se detectan para el sistema, y que son invocadas por medio de un protocolo de aplicaciones. Además, se puede construir un sistema para que el usuario pueda definir sus reglas y cambiarlas. Esto permite que los desarrolladores no sean los principales responsables en cambiar las reglas ya hechas y no tienen la "responsabilidad" de que por ellos no se libera el sistema.

El motor de reglas de negocio debe contener las reglas encontradas del sistema y debe ser diseñado para soportar el cambio. Y para ser incorporado en los sistemas, se debe buscar que se pueda exponer como

servicios o invocarlo por medio de un API<sup>6</sup>. Normalmente, estos productos ofrecen sistemas para construir la aplicación de mantenimiento de las reglas.

Existen diversas herramientas para la gestión de reglas del negocio como:

- onRules (Delta-R)
- ILOG JRules.
- RuleBurst.
- Oracle Rules Engine.
- Microsoft Business Rules Engine.
- JBoss Rules

### **onRules.**

Permite gestionar su sistema de reglas de negocio de forma ágil y dinámica. La aplicación proporciona herramientas totalmente gráficas e intuitivas para facilitar el acceso de los usuarios, con independencia de su nivel de conocimientos técnicos, a todas las fases de la implantación de las reglas (definición, pruebas, despliegue, ejecución y seguimiento). Los usuarios técnicos y de negocio podrán definir reglas con distinto grado de complejidad, crear gráficamente interrelaciones entre ellas (rule flow) y desplegar reglas de negocio de forma autónoma pero integrada con el resto de las aplicaciones y sistemas de la organización.

Las diferentes características técnicas de onRules se condensan en tres pilares en los que se basa la potencia tecnológica de la solución.

1. *J2EE*. El sistema onRules se ha construido sobre la plataforma empresarial de Java (J2EE<sup>7</sup>) que está diseñada para desarrollar aplicaciones potentes y escalables. Además de conseguir un

---

<sup>6</sup> Interfaz de programación de aplicaciones. (Application Programming Interface).

<sup>7</sup> Java 2 Enterprise Edition.



software multiplataforma, la utilización de frameworks estándares como Spring<sup>8</sup>, Hibernate<sup>9</sup> o JSF<sup>10</sup> garantiza la calidad tecnológica de onRules.

2. *SOA*. Este sistema se ha desarrollado sobre una arquitectura 100% orientada a servicios (SOA<sup>11</sup>). Esto permite una sencilla integración con el resto de aplicaciones de la compañía ya que los servicios de onRules cumplen con el estándar SOAP<sup>12</sup> y se describen mediante WSDL<sup>13</sup>.
3. *Web Rich Client*. El entorno de usuario de onRules es un ambiente enriquecido desarrollado con tecnología AJAX<sup>14</sup> y JSF. Esto permite unir la accesibilidad de una plataforma web con la usabilidad del software de escritorio usando técnicas de Drag&Drop y selección dinámica de elementos.

### WebSphere ILOG JRules.

Es un sistema que provee:

- Potente funcionalidad, que incluye: autocorrección en la edición de reglas, detección de redundancias y conflictos, asistentes y herramientas de configuración para la creación de infraestructuras de realización de pruebas e integración de control del código fuente; la capacidad de ampliar fácilmente los proyectos de reglas para la gestión continua a los usuarios empresariales mediante IBM WebSphere ILOG Rule Team Server.

---

<sup>8</sup> Es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java.

<sup>9</sup> Es una herramienta de mapeo objeto-relacional para la plataforma Java.

<sup>10</sup> JavaServer Faces. Es un framework para aplicaciones Java basadas en web.

<sup>11</sup> Service-Oriented Architecture.

<sup>12</sup> Es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. De las siglas de Simple Object Access Protocol.

<sup>13</sup> De las siglas de Web Services Description Language. Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.

<sup>14</sup> Acrónimo de Asynchronous JavaScript And XML. Es una técnica de desarrollo web para crear aplicaciones interactivas y/o enriquecidas.

- Un motor de ejecución robusto, escalable y seguro para aplicaciones basadas en reglas, con un entorno de supervisión y despliegue gestionado (Rule Execution Server).
- Un completo conjunto de funciones de ejecución, que incluyen: un motor de reglas de alto rendimiento y escalabilidad, que proporciona una ejecución de reglas de inferencia o basadas en secuencias, despliegue dinámico de cambios de producción sin tener que reiniciar el servidor, despliegue con un sólo clic de conjuntos de reglas como servicios web para la integración basada en SOA, supervisión mediante la consola web de administración o mediante la conectividad basada en JMX con herramientas de gestión de sistemas empresariales, como IBM Tivoli.
- Sistemas operativos admitidos: AIX, HP Unix, Linux, Sun Solaris, Windows, z/OS.

Sus últimas versiones permiten a los desarrolladores desplegar reglas de negocio almacenadas en el repositorio de reglas JRules, tanto para la plataforma Java, utilizando el motor de reglas nativo en Java, como para la plataforma Microsoft y .NET, por medio del motor de reglas .NET. Los usuarios de JRules ahora también serán capaces de desplegar reglas idénticas para su ejecución desde aplicaciones Java y .NET, por ejemplo, desplegando las reglas en un servidor de aplicaciones J2EE y con un cliente .NET en Windows, lo que facilita el uso compartido de reglas en ambas plataformas.

### **Oracle Business Rules**

Oracle Business Rules permite a los analistas de negocio definir, actualizar y administrar fácilmente las decisiones y políticas clave que rigen las aplicaciones y los procesos de negocio. Por ejemplo, las políticas de negocio dentro de los procesos propensos al cambio pueden ser capturadas utilizando Oracle Business Rules.

Oracle Business Rules consta de una herramienta de autoría de reglas, un motor de reglas y un conjunto de desarrollo de software (SDK). La herramienta de autoría presenta un paradigma propio del inglés para declarar reglas que pueden ser utilizadas tanto por programadores como por analistas de negocio. El

motor de reglas es un motor rápido y eficiente basado en RETE<sup>15</sup> y escrito en Java. Además permite la generación de reglas a través de aplicaciones personalizadas para la edición de reglas.

### **JBoss Drools BRMS**

Este es un motor de reglas de negocios de código abierto basado en estándares para fácil acceso a la política comercial, el cambio y la administración. JBoss Rules es un motor de reglas rápido y altamente eficiente que facilita a un analista comercial o auditor ver las reglas de negocios, a medida que son codificadas en la infraestructura de su aplicación, para verificar que las reglas codificadas implementan en efecto las políticas comerciales documentadas. JBoss Rules también soporta una variedad de ingresos de tablas de decisión y lenguaje, facilitando la rápida modificación de las políticas comerciales para responder a las oportunidades y amenazas competitivas.

Presenta las siguientes características:

- Agilidad y simplificación de la aplicación: simplifica las aplicaciones separando la política o regla comercial de los procesos, infraestructura y lógica de presentación.
- Eficiente y escalable: es una gran forma de recolectar lógica compleja de toma de decisiones y trabajar con conjuntos de datos demasiado grandes para que las personas usen con eficacia.
- Transferencia de la base de conocimientos: facilita la transferencia de conocimientos a repositorios centralizados y ayuda a combatir los problemas debido a la pérdida de tomadores de decisiones claves, administradores, ejecutivos, especialistas y empleados con alto nivel de creatividad.
- Reutilización de reglas en las aplicaciones y Arquitecturas Orientadas a Servicios: Una vez que sus reglas comerciales están separadas de otras lógicas, pueden reutilizarse con más facilidad en muchas aplicaciones y en entornos de arquitectura orientada a servicios.

---

<sup>15</sup> Es un algoritmo de reconocimiento de patrones eficiente para implementar un sistema de producción de reglas

### 2.4.3. ¿Por qué construir un motor de reglas de negocio?

A pesar de que existen motores de reglas de negocio genéricos en el mercado, o aplicaciones específicas que son de hecho, motores de reglas de negocio, hay una serie de razones por las cuales es deseable construir un motor de reglas de negocio en lugar de comprarlo. Una razón importante para la construcción de un motor de reglas de negocio es el costo de un motor de reglas de negocio comercial, el cual puede costar cientos de miles de dólares por la compra o licencia, incluso después de eso, deben ser integradas en el entorno de la empresa. En particular, el motor de regla tiene que ser configurado para obtener los datos que necesita para procesar, lo cual puede aumentar considerablemente el costo total.

Otra razón para no comprar un motor de reglas de negocio es que si se utiliza para aumentar un paquete que se compra, el cliente ahora tendría que tratar con el proveedor del nuevo sistema, y también con el proveedor del motor de reglas de negocio. El cliente preferiría tener un solo proveedor a tratar, antes que dos proveedores diferentes de software para sus necesidades; esto a menudo es el doble de las gestiones necesarias en comparación a tratar con una sola entidad. Además los clientes también son conscientes de que es probable que encuentren problemas de integración si el motor de reglas y el paquete de software nunca antes habían trabajado juntos. Este es un riesgo que la mayoría de los clientes prefieren evitar.

En caso de que un problema dentro de un dominio de negocio requiere una aplicación de software específico, más algún grado de las funcionalidades de un motor de reglas de negocio, tiene importantes ventajas que estos elementos puedan venir de la construcción de un motor de reglas en el marco de la aplicación de software. La primera ventaja es que el aspecto de la funcionalidad del motor de reglas puede ser similar a la de la solicitud. Esto aumenta la aceptación del usuario con respecto al motor de reglas y hace que sea más fácil para los usuarios trabajar con la funcionalidad del motor. La otra ventaja surge de la necesidad de un motor de reglas de negocio para obtener información acerca de la base de datos de la aplicación empresarial y, normalmente, tendrá que escribir los datos de nuevo en esa base de datos. El motor de reglas de negocio debe ser capaz de acceder a la base de datos de la aplicación, leer de ella, y muy a menudo escribir nuevamente en la misma base de datos.

Si un motor de reglas de negocio se construye como parte de la aplicación desde el principio, el motor tiene mucho más probabilidades de mantenerse al día con la información del modelo de datos que requiere, lo cual no puede lograrse con una arquitectura orientada a servicios. Además, con ello se logra la integración de los datos necesarios para que el motor pueda leer y escribir en la base de datos de la

aplicación, y también elimina la necesidad de manejar dos bases de datos separadas. Aunque existen motores de reglas orientados a servicios, por lo que se lleva a cabo por separado, y entrenados para entender el modelo de la aplicación de datos, puede ser necesario ampliar el modelo de datos, como para aplicar reglas de negocio que calculen los valores que deben ser almacenados en una nueva columna, por lo que el motor de reglas de negocio tendrá que reflejar estos cambios, lo cual sería mucho más fácil si el motor de reglas es parte de la solicitud.

A los elementos anteriores se suma con gran fuerza el hecho de que al desarrollar motores de reglas de negocio para aplicaciones específicas, o para dominios de aplicaciones específicas, como se pretende realizar en este trabajo es posible manejar e integrar tecnologías asociadas a lenguajes de dominio específico, la integración de ontologías y la descripción semántica de las reglas y los procesos en los que estarán inmersos.

### CAPITULO 3. DISEÑO DEL SISTEMA

El diseño del sistema de gestión de reglas de negocio debe manejar un conjunto de componentes que actuando de manera coordinada garantizan el funcionamiento de toda la aplicación. En este capítulo se presentan los elementos que fueron considerados para su diseño y las técnicas utilizadas en su implementación.

#### 3.1. Requerimientos no funcionales

##### 3.1.1. Restricciones en el diseño y la implementación

Dentro de su implementación tiene como limitación que la misma deberá ser de código abierto y codificada en un lenguaje multiplataforma, tal es el caso, que para la misma se seleccionó PHP 5.2.x. Por otro lado con respecto al diseño se tiene que el sistema se encuentra atado al motor de flujos de trabajo del framework ezComponents. Además de estar construido para la arquitectura de dominio específico y utilizar una versión de la rama 1.2 del marco de trabajo definido en (Cobo Rodríguez, 2008).

##### 3.1.2. Software

El sistema operativo donde se instalaría la aplicación deberá estar soportado sobre las plataformas \*nix; por lo que se recomienda la utilización de las distribuciones de Debian, Ubuntu y OpenSuse, y como gestor de base de datos se puede utilizar Oracle 11g o superior o PostgreSQL 8.x.

##### 3.1.3. Hardware

Este motor de reglas de negocio, como otra aplicación del SUA que es, cumple con todas los requerimientos de este tipo definidos en (Cobo Rodríguez, 2008). Vale destacar que este sistema es solo para instalar en el lado de la aplicación o el servidor, por lo que los requerimientos del lado del cliente no tienen ningún sentido para este sistema.

## 3.2. Descripción de la solución

Este motor de reglas de negocio está construido sobre el enfoque de las reglas de negocios orientadas a los datos de la aplicación, lo que convierte a su arquitectura a un modo muy general y muchos de sus componentes pueden ser utilizados en otro tipo de motor de reglas de negocio, pero no en todos, por ejemplo los motores de inferencia no necesariamente necesitan actualizar sus operaciones en la base de datos.

Dentro del marco de trabajo de este motor de reglas de negocio, existen ciertos componentes como se muestra en la Figura 1, que forman parte del núcleo del sistema.

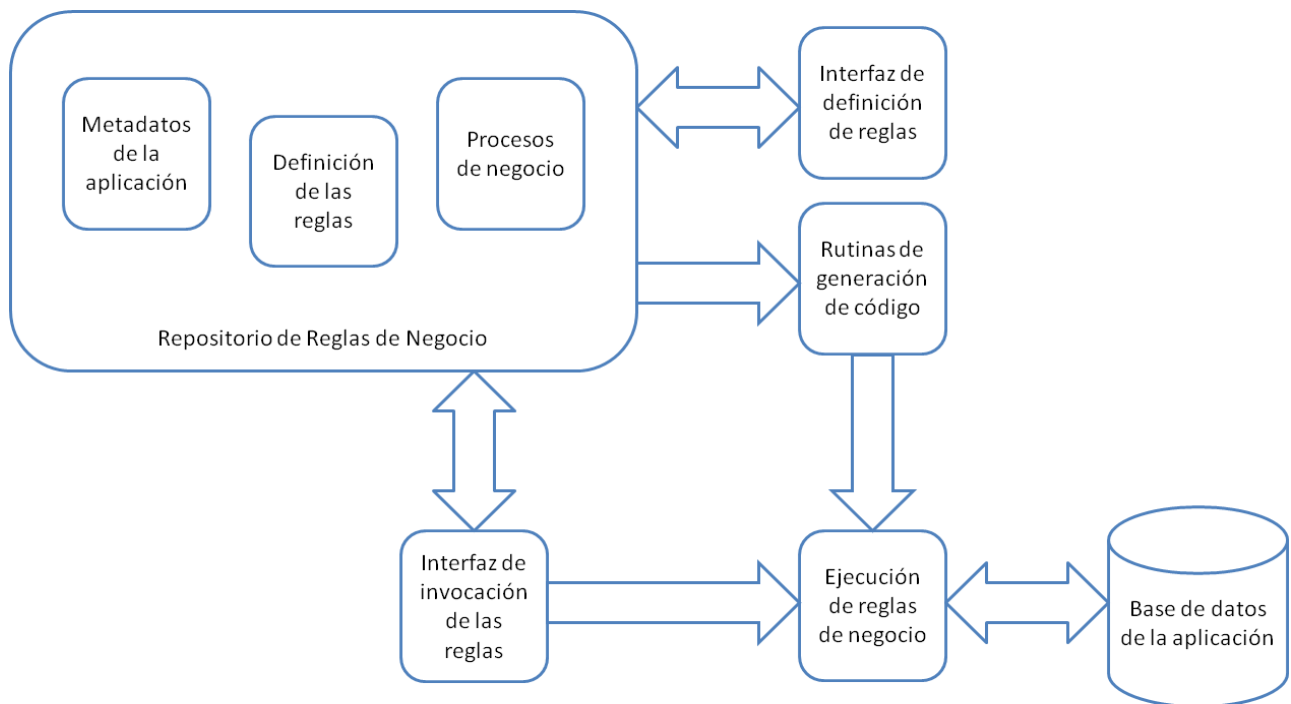


Figura 1 : Componentes del Motor de Reglas de Negocio

### 3.2.1. Repositorio de Reglas de Negocio

Se dice que este componente es el corazón de todo motor de reglas de negocio. Dentro de su estructura contiene un conjunto de tablas que almacenan información acerca de la base de datos de la aplicación de negocio, muy conocidos como los metadatos; las definiciones de las reglas de negocio y las definiciones de los procesos de negocio que son ejecutados por la aplicación. Aunque es muy apropiado en otras circunstancias separar el Repositorio en una base de datos distinta, por ejemplo en caso de que exista un requerimiento que exija que este componente sea propietario y lleve un nivel de seguridad para evitar accesos no autorizados, o por decisión propia del diseño del motor.

#### 3.2.1.1. Metadatos del motor de reglas

El Repositorio de Reglas de Negocio es una base de datos, y como tal debe diseñarse. Requiere la construcción del modelo de datos, y luego la transformación de dicho modelo al modelo lógico y a una base de datos física, secuencialmente. Este subcomponente es el más importante dentro del Repositorio, debido a que describe la estructura de la información de las entidades y las tablas. El motor de reglas de negocio debe entender esta estructura para que así pueda acceder a los datos y actualizar todos los datos relevantes que necesite. Además de las tablas y columnas de la aplicación de negocio y del motor de reglas, los metadatos, recopilan las relaciones entre las tablas, lo cual es un aspecto extremadamente importante para los motores de reglas de negocio porque muestra cómo debe ser navegada la base de datos del negocio.

#### *Información de las entidades*

La unidad básica dentro del Repositorio es la estructura que guarda información acerca de las entidades y tablas de todo el sistema. Con tal propósito se creó una tabla llamada “*sis\_tc\_tabla*”, la cual contiene información acerca de las entidades lógicas y/o tablas físicas de la aplicación empresarial subyacente, con la cual el motor de reglas de negocio operará. Las piezas de la información que necesita esta entidad se describen a continuación.



### *Nombre físico*

Este atributo no es más que el nombre físico de la tabla en la base de datos implementada, hay que tener en cuenta que no es el nombre lógico o de negocio de la tabla. En algunas implementaciones de motores de reglas de negocio el nombre físico de la tabla también es utilizado como la clave principal de la entidad, pero en esta solución se adoptó por crear un identificador de manera autoincrementado y agregar una restricción de unicidad a la propiedad del nombre físico.

Aunque no es el objetivo del trabajo realizar un análisis acerca de las convenciones de nomenclatura para las tablas y otros componentes de base de datos, es importante establecer una pauta elemental para que la ingeniería de metadatos desempeñe un papel importante y no afecte el diseño del motor de reglas de negocio. Es importante que el valor del nombre físico de la tabla tenga un límite en su tamaño, debido a que los nombres largos son difíciles de manejar, y los hace difícil para discutir los aspectos técnicos entre el personal de un proyecto. Existen algunos gestores relacionales que tienen limitaciones en el número de caracteres que se le puede asignar a un nombre de tabla. Para lograr que el diseño del motor de reglas de negocio sea genérico, independiente de la plataforma de base de datos, dicho motor debe poder ajustarse dentro de los límites establecidos para la plataforma en que será implementado. Para este motor se decidió que 30 caracteres como límite es una buena elección debido a que permanece dentro de los límites de nombrado de los principales gestores de base de datos que dominan el mercado.

Los nombres de tablas se utilizarán una y otra vez en las tablas que guardan información sobre las reglas de negocio. Sus nombres deben ser útiles, no deben exigir decodificación difícil para saber lo que son. El nombre físico de la tabla debe ser comprensible para el personal que construye el motor de reglas y otros que tienen algún papel en la administración de bases de datos. Los nombres dados tienen que ser capaces de hacer referencia a tablas de la base de datos de una manera que no sólo identifica a cada tabla, sino que también describe lo que cada tabla contiene.

Muchas empresas ya han creado convenciones de nomenclatura para las tablas de la base de datos. Durante un proyecto de reglas de negocio, se debe considerar en cuanto a si estos convenios deben ser seguidos. Si el proyecto de reglas de negocio solo va a utilizarse dentro de la empresa, entonces hay una buena razón para seguir estas normas. Incluso si las normas son pobres, la aplicación de reglas de negocio será coherente. Por otra parte, si el paquete que se está desarrollando será integrado a clientes

distintos, no hay justificación para seguir un conjunto establecido de convenciones de nombres específicos de un cliente. Las convenciones de nomenclatura utilizadas para este motor son las siguientes:

- El nombre de la tabla no puede tener más de 30 caracteres de longitud.
- Solo las letras en mayúsculas, dígitos y guiones bajos pueden componer el nombre de la tabla.
- Si el nombre esta dividido por más de una palabra, estas deben ser unidas por guiones bajos.
  - El primer nodo no puede ser más de tres caracteres y define la clasificación a la cual pertenece la tabla, acorde a lo establecido en la Tabla 2.
  - El segundo nodo es una palabra que describa lo que la tabla contiene.

### *Nombre lógico*

Este es el nombre por el que se conoce a la entidad en la empresa. Se trata de una información recogida por los analistas o los modeladores de datos y no por los programadores o administradores de bases de datos, es muy importante tener el nombre correcto y completo. En las situaciones en que estos nombres son largos, se puede almacenar un nombre "oficial largo", y uno no oficial "corto". Para lograr esto, un atributo independiente llamado "nombre corto" se podría agregar a la entidad de la tabla. La palabra corta es particularmente significativa en este contexto, ya que si la longitud se mantiene dentro de un tamaño máximo estándar, entonces será posible usar este atributo en las pantallas e informes, donde el espacio es siempre un problema. Aunque esto no es lo que se diseñó para este motor.

### *Descripción*

Esta es la definición del negocio de la entidad. A menudo se descuida esta propiedad de los diseños de base de datos, probablemente por la razón de que se considera solo en la documentación o que no tiene relación evidente con la lógica del programa o con las reglas de negocio. La realidad es que las definiciones de algunas entidades tienen una conexión muy directa con las reglas de negocio y debe ser aclarado por completo. Este atributo es vital, dentro del diseño de la entidad, para la comprensión de los

analistas, programadores, y también usuarios del negocio. Debe estar construido con cuidado y debe ser lo más claro posible.

Teniendo la descripción de la entidad dentro de la aplicación aumenta la utilidad de este atributo en otros aspectos. Por ejemplo con frecuencia, las definiciones de las entidades se colocan en la documentación, lo que constituye la funcionalidad base de la ayuda de una aplicación. Por lo tanto, las definiciones a menudo se almacenan y aparecen en varios lugares en una aplicación, como resultado de las diversas instancias de las definiciones; además tienden a ser diferentes con el tiempo, causando confusión. Brindando la posibilidad de tener la descripción de la entidad en un solo lugar, en este caso el Repositorio, donde se puede acceder fácilmente, se puede prevenir toda esta confusión.

### *Tipos de entidades*

El esquema de clasificación para las entidades en la base de datos se presenta en la Tabla 2. Estas convenciones de nombres son para facilitar el uso de las entidades por los seres humanos, lo que se recomienda que no deban ser utilizados en el procesamiento de la lógica. Es muy importante poder distinguir a cada tipo de tabla que necesita ser procesada por el motor de reglas y además diferenciar las tablas de datos de referencia de otros tipos de tablas en la base de datos. Esto se debe a que las tablas de datos de referencia se utilizan ampliamente en las funcionalidades del motor de reglas.

Nombre	Código	Descripción
<i>Metadatos</i>	M	Tabla que contiene metadatos.
<i>Referenciales</i>	R	Tabla que contiene datos de referencia.
<i>Entidades</i>	E	Tabla que contiene lógica de negocio.
<i>Motor de Regla</i>	S	Tabla que contiene datos del motor de reglas de negocio.

**Tabla 2: Clasificación de los tipos de tablas**

El diseño básico de la entidad “*m\_tc\_tabla*” en el Repositorio se muestra en el modelo lógico de la Figura 2. Es preciso señalar que el diseño presentado en Tabla 3 es un punto de partida, no un diseño completo, que apoyará todas las implementaciones de un motor de reglas de negocio. Para otras implementaciones podría representar un mínimo de lo que se necesita, por lo que en caso de no ajustarse esta solución a la

implementación del motor de reglas deseado no existirá más remedio que ampliar las propiedades de la entidad para satisfacer las necesidades de las reglas de negocio.

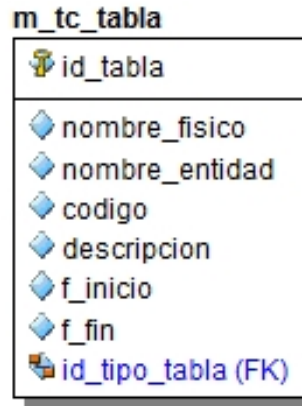


Figura 2: Modelo lógico de la entidad "m\_tc\_tabla"

Nombre	<i>m_tc_tabla</i>	
Descripción	Tabla que cumple el estándar de nomenclador y su misión es controlar todas las tablas tanto del motor de reglas de negocio como de la aplicación empresarial.	
Atributo	Tipo	Descripción
<i>Id_tabla</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Nombre_fisico</i>	Varchar	Contiene la designación física de la tabla en la base de datos
<i>Nombre_entidad</i>	Varchar	En este campo se almacena el nombre lógico-empresarial para la tabla física.
<i>Código</i>	Varchar	Es un valor, que cumple una restricción de unicidad, con el cual se identifica la tabla.
<i>Descripción</i>	Varchar	Es un texto complementario vital para el entendimiento de los analistas, programadores y el usuario final.
<i>F_inicio</i>	Date	Atributo que indica el inicio de uso y puesta en marcha de la tabla en cuestión.
<i>F_fin</i>	Date	Atributo que indica el fin de uso y puesta en marcha de la tabla en

		cuestión.
<i>Id_tipo_tabla</i>	Integer	Propiedad donde se almacena una referencia al tipo de tabla.

Tabla 3: Descripción de la tabla "m\_tc\_tabla"

Como se puede notar en la estructura de esta tabla, se muestran dos propiedades que siguen el estándar de todos los nomencladores definidos para las aplicaciones del SUA, los cuales dentro de este dominio se le denominan "tablas de control". Muchos de los datos en la AGR tienen carácter temporal y la mayoría de los gestores relacionales no tienen soporte directo a estos tipos de datos y por esto se añade en el diseño, a todas las tablas que tengan la misión de un nomenclador, los atributos "*f\_inicio*" y "*f\_fin*", para poder llevar el histórico o temporalidad de sus artículos. Por lo que de ahora en adelante omitiremos la descripción de estos atributos en las tablas que cumplan con esta estructura.

### *Información de las columnas*

El siguiente elemento en los metadatos que hay que tratar, es la información acerca de las columnas que se incluyen en todas las tablas de la base de datos. El Repositorio de Reglas de Negocio debe ser capaz de mantener los metadatos de las columnas que se utilizan para contener datos de negocio relevantes. Si se examina cualquier modelo de datos a nivel lógico, se puede ver que cada entidad tiene atributos. Cuando se construye una base de datos físicamente, las entidades están representadas por las tablas y los atributos por columnas. Ahora se pasará a analizar la estructura de las entidades necesarias para la captura de información de la columna, tanto a nivel lógico, como los nombres y descripciones, y en el nivel físico, tales como tipos de datos, que tienen que ser almacenados y puestos a disposición del motor de reglas de negocio.

### *Nombre físico*

Desde la perspectiva de las reglas de negocio, este atributo es el más importante en todo el Repositorio y su misión es contener el nombre físico de la columna que describe. Lo que conduce inmediatamente a la cuestión de las convenciones de nomenclatura.

Así como tenía que existir un límite de tamaño para el nombre físico de la tabla, se necesita uno para el nombre físico de la columna. Debe estar construido de manera que no exceda los límites de tamaño de los sistemas de gestión de base de datos de uso común. Los nombres largos son difíciles de manejar en cualquier caso, sobre todo para los programadores y otras personas que realmente tienen que utilizar estos nombres en las discusiones. Un valor para este atributo debe ayudar a las personas que lo utilizan tanto como sea posible, sin causar problemas para su uso dentro de una aplicación informática. Existen, al igual que las nomenclaturas para las tablas, convenciones de nomenclatura para los nombres físicos de las columnas y para muchas empresas pueden ser un requisito para usarlas. Para este sistema se adoptaron las siguientes nomenclaturas.

- El nombre de la columna no puede tener más de 30 caracteres de longitud.
- Solo las letras en mayúsculas, dígitos y guiones bajos pueden componer el nombre de la columna.
- Si el nombre está dividido por más de una palabra, estas deben ser unidas por guiones bajos.
  - El primer nodo no puede ser más de tres caracteres y define la clasificación a la cual pertenece la columna, la cual se muestra en la Tabla 4.
  - El segundo nodo es una palabra que describa el objetivo de lo que almacena la columna.

Nombre	Código	Descripción
<i>Llave Primaria</i>	C	Columna que forma parte de la llave primaria de una tabla
<i>Contador</i>	N	Columna que almacena un contador de cualquier elemento.
<i>Fechas</i>	D	Columna que almacena tipo de datos de fechas.
<i>Monto Final</i>	A	Columna que guarda un valor de una cuenta financiera.
<i>Años</i>	Y	Columna que almacena valores de años.
<i>Porcentaje</i>	P	Columna que almacena un valor de porcentaje.
<i>Cantidad</i>	Q	Columna que almacena un cantidad o monto, pero no es financiera.
<i>Meses</i>	M	Columna que almacena valores de meses.
<i>Indicador o bandera</i>	I	Columna que almacena una bandera o un valor booleano.
<i>Radio</i>	R	Columna que almacena un valor de radio.
<i>Texto</i>	T	Columna que almacena una cadena de caracteres.

Tabla 4: Clasificación de los tipos de columna

### *Relación referencial con la entidad tabla*

La entidad de las columnas debe poseer la información básica acerca de las columnas que contienen datos relevantes para la empresa. Una pregunta inmediata para la entidad es si la columna debe ser una entidad dependiente de la entidad tabla o independiente de la misma. En este diseño se sigue la concepción de poder crear los atributos de las columnas por sí mismos y tan solo clasificarlos dentro de las entidades. Aunque, en otras implementaciones o diseño, también se puede seguir la idea de que un atributo depende tanto de su existencia y la identificación de una entidad que no puede existir sin la misma. Por lo tanto lo que hace la relación de no identificación de la tabla a la columna significa que la clave principal de la tabla va a terminar en la zona de los atributos que no son clave principal de la columna.

Lo que es innegable es que es mejor si existe una sola clave principal aplicada a la definición física de la entidad de las columnas. Esto hace que sea mucho más fácil construir un motor de reglas de negocio, ya que gran parte de las funcionalidades requieren la manipulación de los metadatos de las columnas sin necesidad de consultar los metadatos de la tabla asociada. Por lo que en el diseño de esta implementación la entidad contiene una clave o llave principal que tan solo es una secuencia autoincrementada para la identificación de los atributos de la columna.

Se diseñó una entidad principal para mantener la información de la columna en el Repositorio de Reglas de Negocio, la cual se denomina “*m\_tc\_campo*”, y su diseño, a nivel lógico, se muestra en la Figura 3 y su descripción física se puede observar en la Tabla 5. Debe quedar claro que todo el diseño del modelo de datos presentado en este trabajo responde a la implementación realizada para este motor de reglas de negocio. En caso de realizar otra implementación habría que analizar si este diseño es básico o no para el alcance de la empresa o del motor de reglas de negocio.



Figura 3: Modelo lógico de la entidad "m\_tc\_campo"

<b>Nombre</b>	<i>m_tc_campo</i>	
<b>Descripción</b>	Tabla que cumple el estándar de nomenclador y su misión es controlar todas las columnas de las tablas tanto del motor de reglas de negocio como de la aplicación empresarial	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>Id_campo</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Nombre</i>	Varchar	Contiene la designación física de la columna de una tabla en la base de datos. Además que cumple las restricciones de la <b>¡Error! No se encuentra el origen de la referencia..</b>
<i>Código</i>	Varchar	Es un valor, que cumple una restricción de unicidad, con el cual se identifica la columna.
<i>Id_tabla</i>	Integer	Propiedad donde se almacena una referencia a la tabla a la cual pertenece la columna.
<i>Id_tipo_campo</i>	Integer	Atributo que sostiene la categoría de la columna acorde con la Tabla 4.
<i>Id_columna_fk</i>	Integer	Aquí se almacena una referencia a la columna donde este registro se comporta como llave foránea.



Tabla 5: Descripción de la tabla "m\_tc\_campo"

### *Tipos de datos*

El tipo de datos de la columna es algo que necesita ser conocido en muchas circunstancias para un motor de reglas de negocio. Por ejemplo, a menudo desempeña un papel en los algoritmos de generación de código. Por desgracia, los tipos de datos son un tanto específicos de cada sistema de gestión de base de datos. Si el motor de reglas está diseñado para trabajar con un producto de base de datos, no hay mucho problema, sin embargo, si el motor de reglas tiene que trabajar con más de un gestor de base de datos, las cosas pueden ponerse difíciles.

Existen otras cuestiones con respecto a los tipos de datos. Algunos pertenecen a categorías básicas, como los numéricos. Otros son subtipos con sus propias características especiales, por ejemplo, tipos de datos que tienen una longitud, precisión, escala e incluso una máscara para su validación. El problema es que los diferentes servidores de bases de datos utilizan el mismo valor de nombre a tipos de datos para referirse a cosas diferentes. Por ejemplo, el tipo de dato "texto" en Microsoft Access es un tipo de datos de caracteres que pueden ser de hasta 250 caracteres de longitud, mientras que en Microsoft SQL Server 2000, representa un tipo de datos que puede almacenar bloques de tipo memo, los cuales son de mayor información textual. Otra razón para tomar este enfoque de diseño es que también soporta los tipos de datos definidos por el usuario, algo muy útil para los motores de reglas de negocio.

En la Figura 4 se muestra el modelo lógico para los tipos de datos soportados. En el diseño se contempla los dos enfoques de tipos de datos, el primero abarca los tipos de datos base, es decir, la definición de los tipos de datos soportados por los gestores de base de datos, lo cual es una responsabilidad de la entidad "m\_tc\_tipo\_dato". El segundo enfoque es acerca de los subtipos definidos para el motor, mencionado anteriormente, los mismos se registran en la entidad "r\_campo\_tipo\_dato".

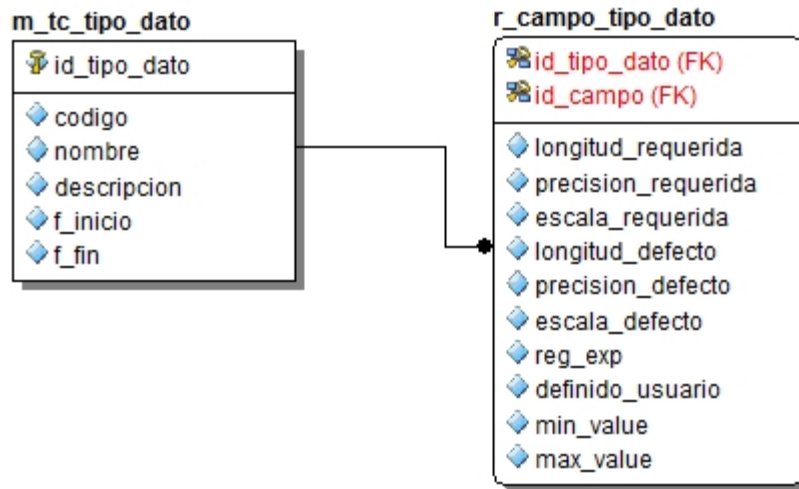


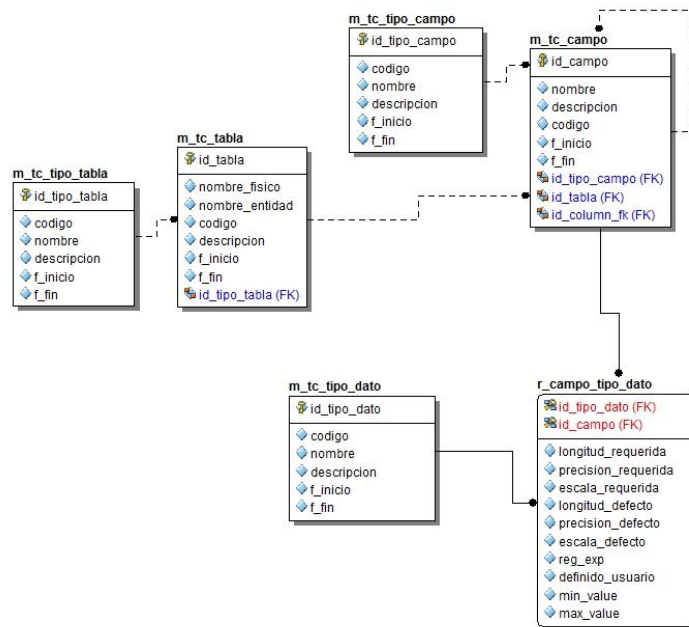
Figura 4: Modelo lógico de los tipos de datos

Nombre	<i>r_campo_tipo_dato</i>	
Descripción	Su misión es controlar todos los subtipos de datos existentes para el gestor de base de datos y los definidos por el usuario, asociados a las columnas de las tablas.	
Atributo	Tipo	Descripción
<i>Id_tipo_dato</i>	Integer	Es una referencia al tipo de base soportado por el gestor de base de datos.
<i>Id_campo</i>	Integer	Es una referencia a la columna a la que pertenece el tipo de dato definido.
<i>Longitud_requerida</i>	Integer	Se almacena la longitud que se define por el usuario para la columna.
<i>Precision_requerida</i>	Integer	Se almacena la precisión, para los tipos de datos numéricos, y es definido por el usuario.
<i>Escala_requerida</i>	Integer	Se almacena la escala, para los tipos de datos numéricos, y es definido por el usuario.
<i>Longitud_defecto</i>	Integer	Se almacena la longitud que soporta el gestor de base de datos para el tipo de dato de la columna.

<i>Precision_defecto</i>	Integer	Se almacena la precisión, para los tipos de datos numéricos, que soporta el gestor de base de datos.
<i>Escala_defecto</i>	Integer	Se almacena la escala, para los tipos de datos numéricos, que soporta el gestor de base de datos.
<i>Reg_exp</i>	Varchar	Almacena una máscara para la validación de la columna.
<i>Definido_usuario</i>	Booleano	Indica si el tipo de dato para la columna es definido o no por el usuario.
<i>Min_value</i>	Integer	Guarda el valor mínimo soportado por la columna.
<i>Max_value</i>	Integer	Guarda el valor máximo soportado por la columna.

**Tabla 6: Descripción de la tabla "r\_campo\_tipo\_dato"**

Hasta este punto, se tiene el diseño lógico completo de los metadatos del Repositorio para la implementación de este motor de reglas, que se muestra en la Figura 5.



**Figura 5: Modelo lógico del componente de los metadatos**

### 3.2.1.2. Definición de las reglas

Este componente es para el cual el motor de reglas está diseñado con el objetivo de administrar y ejecutar. La definición de la regla es realmente la definición de una forma ejecutable de la regla en alguna forma de lenguaje de programación, que puede ser un idioma comercial, o un "lenguaje de scripting" desarrollado para el motor de reglas en cuestión.

La definición de una regla de negocio consiste en un grupo de metadatos estructurados que describe la regla. Más que esto, si la regla de negocio se refiere a otros objetos en la aplicación informática, los metadatos sobre estos objetos también deben ser capturados en la definición de la regla. Si se hace esto, las reglas pueden estar relacionadas con otro tipo de información que se lleva a cabo sobre estos objetos en otros lugares en el Repositorio de Reglas de Negocio. Los objetos principales que se refieren a las reglas de negocio son las tablas y columnas, ya que estos contienen los datos de negocio en el que las reglas funcionan. Por lo tanto, como mínimo, las definiciones de las reglas de negocio deben ser almacenados en las tablas del Repositorio que están relacionados con las tablas de la base de datos de los metadatos de la aplicación. Esto significa que existen dos enfoques necesarios para la gestión de reglas de negocio en el motor de reglas. El primer enfoque es capturar la definición de las reglas en forma de metadatos en las tablas del Repositorio. Esto abre las reglas de negocio a toda una serie de técnicas de gestión de gran valor. El segundo enfoque es utilizar la definición de reglas para crear una forma ejecutable de cada regla de negocio.

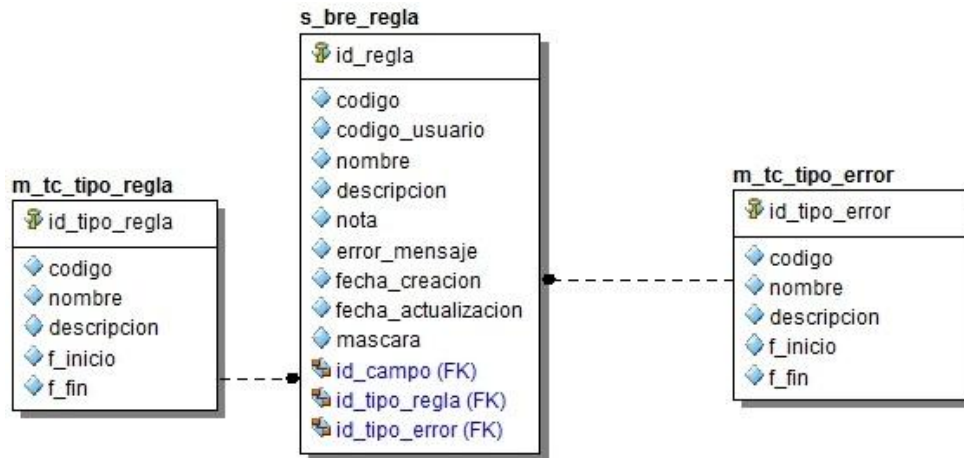


Figura 6: Modelo lógico de la definición de las reglas.

<b>Nombre:</b>	<i>s_bre_regla</i>	
<b>Descripción:</b>	Su objetivo es registrar todas las reglas del negocio disponibles para el motor.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>Id_regla</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Código</i>	Varchar	Es un valor, que cumple una restricción de unicidad, con el cual se identifica la regla.
<i>Nombre</i>	Varchar	Nombre lógico que define a la regla.
<i>Descripción</i>	Varchar	Definición empresarial de la regla de negocio.
<i>Nota</i>	Varchar	Texto aclaratorio para entender aun más la definición de la regla.
<i>Error_mensaje</i>	Varchar	Texto que se muestra al usuario en caso de violar la regla.
<i>Fecha_creacion</i>	Date	Fecha en la que se crea la regla.
<i>Fecha_actualizacion</i>	Date	Fecha de la última modificación de la regla.
<i>Mascara</i>	Varchar	Solo para el caso de las reglas de negocio de cálculo se guarda la notación polaca de la ecuación.
<i>Id_campo</i>	Integer	Es una referencia a la columna a la cual se le aplica la regla.
<i>Id_tipo_regla</i>	Integer	Es una referencia a la categoría a la que pertenece la regla, estas categorías se muestran en la Tabla 14.

<i>Id_tipo_error</i>	Integer	Es una referencia al tipo de error a mostrar en el caso de violar la regla. Los tipos de errores se listan en la Tabla 8.
----------------------	---------	--

Tabla 7: Descripción de la tabla "s\_bre\_regla"

Nombre	Código	Descripción
<i>Información</i>	04	Para el caso en que solo se desee mostrar una ventana de información.
<i>Advertencia</i>	08	Para el caso en que solo se desee mostrar una ventana de advertencia.
<i>Error</i>	12	Para el caso en que solo se desee mostrar una ventana de error.
<i>Error Crítico</i>	16	Para el caso en que solo se desee mostrar una ventana de error crítico.

Tabla 8: Clasificación de los tipos de errores

### 3.2.1.3. Procesos de negocio

El hecho de definir un conjunto de reglas de negocio no es suficiente para construir un motor de reglas de negocio. Aun cuando el motor de reglas de negocio puede determinar la secuencia de ejecución entre las reglas de negocio es necesario que este reconozca cuando han sido combinadas diferentes reglas y si estas fueron ejecutadas en el ámbito específico de una combinación concreta. Expresado de otra manera sería aseverar que el motor de reglas de negocio precisa de manejar los procesos de negocio que constituyen el contexto de ejecución de cada regla particular.

Según la definición que hace (Chisholm, 2007) los procesos de negocio, son definidos como:

*“Un proceso de negocio es una unidad lógica independiente de trabajo. De hecho, debería consistir en de procesamiento de extremo a extremo que comienza con un evento de algún tipo y termina con la salida que responde a ese evento”.*

Sin embargo cada uno de estos procesos de negocio puede contener un número de pasos, que cada cual implica un conjunto diferente de acciones coherentes que por lo general implican a un conjunto diferente de actores, este término está dado a los participantes en una fase del proceso, ya sean humanos, de organización, o sistema. Estos pasos del proceso de negocio son importantes para un motor de reglas de negocio porque están más cerca a las unidades lógicas de trabajo que utilizan un conjunto bien definido de reglas de negocio.

Los pasos del proceso de negocio son un elemento importante dentro del enfoque de motor de reglas de negocio. Los cuales se utilizan para definir conjuntos de reglas que se ejecutan conjuntamente, por lo tanto necesitan ser capturados en el Repositorio de Reglas de Negocio y definir, inevitablemente, la secuencia de estos pasos, para que el motor de reglas de negocio sea capaz de determinar el orden en que las reglas de negocios son ejecutadas en cada paso por lo que esta secuencia debe ser definida por los usuarios.

La Figura 7 muestra las adaptaciones a los metadatos necesarios para definir los procesos empresariales.

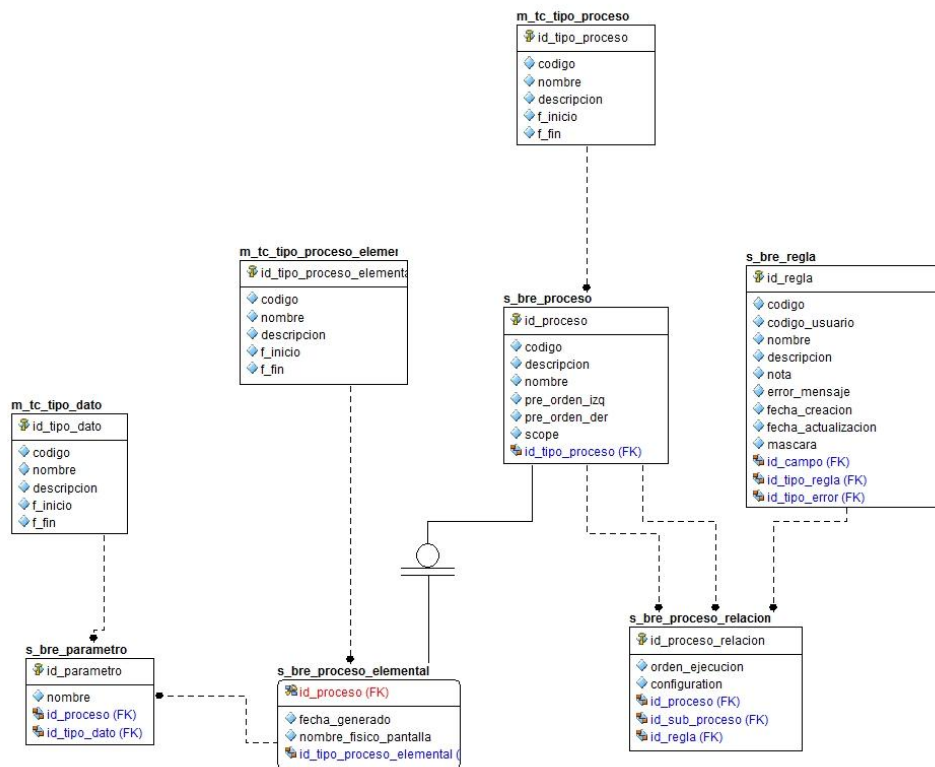


Figura 7 : Modelo lógico para definir los proceso empresariales

Las nuevas entidades agregadas al Repositorio son las siguientes:

- *Procesos de negocio*: la responsabilidad de esta entidad recae en la entidad “*s\_bre\_proceso*”, y su misión es definir los procesos generales que realiza la empresa.
- *Pasos del proceso de negocio*: es un paso concreto dentro de un proceso de negocio. Un punto importante a señalar aquí es que un paso del proceso de negocio se basa en una tabla base, llamada tabla base del paso del proceso de negocio. Sólo las columnas de esta tabla están disponibles para la definición de reglas para un determinado paso del proceso de negocio. Esto sirve para asegurar que las reglas asociadas a un paso son un verdadero conjunto coherente de reglas. Un paso de un proceso de negocio puede estar relacionado con una pantalla, pero no siempre. Todo esto se garantiza dentro del diseño de la implementación a través de la especificación de “*s\_bre\_proceso\_elemental*”, y de la validez de la asociación del conjunto coherente de reglas de negocio se encarga la entidad “*s\_bre\_proceso\_relacion*”. La descripción del modelo físico de ambas entidades se observa en la Tabla 10 y Tabla 11, respectivamente.
- *Parámetros de los pasos del proceso de negocio*: en ocasiones los pasos de un proceso de negocio se encuentran relacionados con una pantalla o interfaz gráfica de usuario (GUI, por sus siglas en inglés). Esto significa que el paso del proceso de negocio requiere de una interacción con el usuario final, por lo que el paso del proceso lleva asociado ciertos parámetros de entradas para continuar con el flujo de trabajo del proceso general. Con tal objetivo se diseñó la entidad “*s\_bre\_parametro*”, la cual guarda todos los parámetros necesarios para un paso del proceso de negocio. La descripción de esta entidad se muestra en la Tabla 12.

<b>Nombre:</b>	<i>s_bre_proceso</i>	
<b>Descripción:</b>	Su misión es definir los procesos generales que realiza la empresa.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>Id_proceso</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Código</i>	Varchar	Es un valor, que cumple una restricción de unicidad, con el cual se identifica el proceso de negocio.
<i>Descripción</i>	Varchar	Definición empresarial del proceso de negocio.
<i>Nombre</i>	Varchar	Nombre lógico que define al proceso de negocio.
<i>Pre_orden_izq</i>	Integer	Almacena el elemento izquierdo del pre orden en una jerarquía.



<i>Pre_orden_der</i>	Integer	Almacena el elemento derecho del pre orden en una jerarquía.
<i>Scope</i>	Integer	Identificador único para la jerarquía.
<i>Id_tipo_proceso</i>	Integer	Referencia para diferenciar entre un proceso general y un paso del proceso.

Tabla 9: Descripción de la tabla "s\_bre\_proceso"

<b>Nombre:</b>	<i>s_bre_proceso_elemental</i>	
<b>Descripción:</b>	Su objetivo es controlar los pasos de un proceso de negocio empresarial.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>Id_proceso</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Fecha_generado</i>	Date	Almacena la fecha en la que se genera la lógica del paso.
<i>Nombre_fisico_pantalla</i>	Varchar	Nombre de la lógica del programa asociado al paso.
<i>Id_tipo_proceso_elemental</i>	Integer	Es una referencia a los tipos de pasos de procesos de negocio los cuales se muestran en la Tabla 13.

Tabla 10: Descripción de la tabla "s\_bre\_proceso\_elemental"

<b>Nombre:</b>	<i>s_bre_proceso_relacion</i>	
<b>Descripción:</b>	Su objetivo es la asignación de los pasos al proceso de negocio.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>Id_proceso_relacion</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Orden_ejecucion</i>	Integer	Solo para los pasos del proceso y almacena el orden de ejecución de los mismos.
<i>Configuracion</i>	Varchar	Guarda la configuración de los valores de los parámetros asignados a los pasos del proceso de negocio.
<i>Id_proceso</i>	Integer	Es una referencia al proceso de negocio al que pertenece el paso.

<i>Id_sub_proceso</i>	Integer	Es una referencia al paso del proceso de negocio.
<i>Id_regla</i>	Integer	Es una referencia al conjunto de reglas asociados al paso del proceso de negocio.

Tabla 11: Descripción de la tabla "s\_bre\_proceso\_relacion"

<b>Nombre:</b>	<i>s_bre_parametro</i>	
<b>Descripción:</b>	Su misión es almacenar una posible configuración de los parámetros de entradas al paso del proceso de negocio.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>Id_parametro</i>	Integer	No es más que una secuencia de forma autoincrementada y constituye la llave primaria de la entidad.
<i>Nombre</i>	Varchar	Nombre lógico del parámetro.
<i>Id_proceso</i>	Integer	Referencia al paso del proceso de negocio al que pertenece dicho parámetro.
<i>Id_tipo_dato</i>	Integer	Referencia al tipo de dato que soporta el parámetro.

Tabla 12: Descripción de la tabla "s\_bre\_parametro"

Nombre	Código	Descripción
<i>SCREEN</i>	SCREEN	Para los pasos que requieran de una interacción con el usuario.
<i>BATCH</i>	BATCH	Para los pasos que solo requieren de la ejecución de la lógica del programa.
<i>PARALLEL</i>	PARALLEL	Para los pasos que contienen otros pasos y estos se ejecutan de manera paralela.
<i>EXCLUSIVE CHOICE</i>	EXCLUSIVE	Para los pasos que contienen otros pasos y estos se ejecutan con una condición de exclusividad.
<i>MULTI-CHOICE</i>	MULTI-CHOICE	Para los pasos que contienen otros pasos y estos se ejecutan con múltiples condiciones.

Tabla 13: Clasificación de los tipos de pasos de un proceso de negocio

### 3.2.2. Interfaz de definición de las reglas

El Repositorio de Reglas de Negocio contiene los metadatos que definen las reglas de negocio, pero para estas se debe crear una interfaz que permita que las reglas se definan. Esta interfaz se compone de un conjunto de pantallas que permite al usuario definir reglas, que para esta implementación se muestra en la Figura 8. En este punto es necesario introducir el concepto de tipos de reglas de negocio. Hay muchas similitudes entre las reglas de negocio, pero aún así es posible agruparlas en categorías. Estas categorías se denominan tipos de reglas de negocio y se listan en la tabla a continuación y el diseño del modelo lógico fue visto en la Figura 6.

Nombre	Código	Descripción
Campos de formulario requeridos	SCRREQ	Chequear todos los campos en el formulario que deben tener valores.
Validar tipos de datos en los formularios	SCRVDTYPE	Comprobar que todos los campos en el formulario tengan un valor esperado.
Validar rangos de fechas en los formularios	SCRVRANGE	Comprobar que los campos de fecha estén entre los rangos especificados.
Validar registros duplicados	REFINTDUP	Comprobar que no existan registros duplicados.
Integridad Referencial en los registros	REFINTDEL	Comprobar que exista integridad referencial en los registros.
Validar opciones seleccionadas	SCRXCHK	Comprobar las opciones múltiples en los formularios.
Campos de formulario requeridos	SCRREQ	Chequear todos los campos en el formulario que deben tener valores.
Validar tipos de datos en los formularios	SCRVDTYPE	Comprobar que todos los campos en el formulario tengan un valor esperado.
Validar rangos de fechas en los formularios	SCRVRANGE	Comprobar que los campos de fecha estén entre los rangos especificados.
Validar registros duplicados	REFINTDUP	Comprobar que no existan registros duplicados.
Integridad Referencial en los registros	REFINTDEL	Comprobar que exista integridad referencial en los registros.

Tabla 14: Clasificación de los tipos de reglas



Figura 8: Interfaz de la definición de las reglas de negocio.

### 3.2.3. Interfaz de invocación de las reglas

En la sección anterior se explicó la manera en que las reglas de negocio se crean y empaquetan como rutinas que representan un conjunto de reglas. Cuando se ejecuta una aplicación, estas rutinas de alguna manera se deben invocar. Esto se realiza frecuentemente por un usuario, cuando selecciona una opción en un menú o al hacer clic en un botón en una pantalla. Estos componentes o rutinas de una aplicación se construyen a mano por los programadores, lo que significa que los programadores tienen que saber cómo llamar a las rutinas que contienen los conjuntos de reglas.

Con el fin de implementar un mecanismo de invocación para las reglas de negocio, el primer requisito es tener un modelo claro del proceso del sistema. Este debe contener una lista de todos los procesos de negocio dentro del sistema y los pasos de los procesos de negocio de forma secuencial que se ejecutan dentro de cada proceso de negocio. Este modelo de proceso debe identificar claramente los nombres de las rutinas generadas que corresponden a las etapas del proceso de negocio. Además, los parámetros que se deben pasar por un programador a las rutinas, los cuales deben estar claramente identificados. También los programadores deben tener presente que las rutinas que ejecutan las reglas de negocio en ocasiones devuelven alguna que otra información, por lo que el código del programa creado por el mismo debe ser capaz de hacer frente a esta posibilidad.

Por supuesto, es más coherente con el planteamiento de reglas de negocio si los parámetros y valores devueltos pueden ser tratados por el código generado en lugar de cualquier lógica creada por los programadores, pero esto no siempre es posible. Sin embargo, es bastante factible para reducir lo que los programadores tienen que aplicar para llamar a las reglas de negocios ejecutables a unas simples declaraciones. Con tal propósito se creó en esta implementación el modelo conceptual que aparece en la Figura 9, para gestionar las rutinas de invocación de las reglas de negocio.

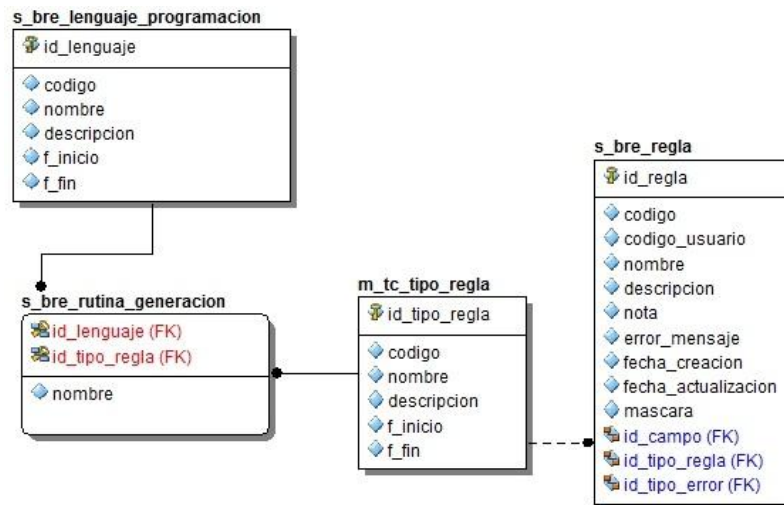


Figura 9: Modelo lógico del componente de invocación de las reglas de negocio

### 3.2.4. Rutinas de generación de código

La implementación dada para este componente, es la construcción de una rutina de generación de código para cada tipo de regla, y para cada entorno de lenguaje de programación en que las reglas de negocio tienen que ser aplicadas. Cuando los metadatos de la regla se crean, o cuando cambian, la rutina de generación de código se llama y crea el código fuente para el entorno de programación determinado. Sin embargo, en lugar de piezas individuales de código de programa para cada regla de negocio, el código fuente se agrupa en conjuntos de reglas. Cada paso del proceso de negocio es considerado como un conjunto de reglas y se implementa como una rutina distinta. Por lo tanto, aunque sólo sea una modificación en alguna regla, el motor de reglas regenera el código del programa para todos los pasos del proceso de negocio en el que participa el cambio de la regla. Una razón importante de esto es que si la

definición de una regla se cambia, puede afectar a la secuencia en que las normas se ejecutan dentro de una fase del proceso de negocio. Por lo tanto, es necesario re-secuenciar todas las reglas de los pasos del proceso de negocios en el que participa el cambio de la regla. Esto realmente no se puede evitar, y por lo tanto la regeneración de sólo el código de programa de la regla modificada, no es una opción.

Hay algunas grandes ventajas de la generación de código de manera completa para los pasos del proceso de negocio en vez de la generación para reglas individuales.

1. Es muy fácil de ejecutar un paso del proceso de negocio. Es simplemente una cuestión de la invocación de la rutina asociada.
2. Es posible que un programador pueda encontrar la rutina de una fase del proceso de negocio y entender con confianza lo que ocurre cuando el paso del proceso de negocio se ejecuta. Esto es simplemente cuestión de mirar el código que se genera para la rutina. Lo cual no sería lo mismo si el motor de reglas invocara, en tiempo de ejecución, las rutinas individuales para la regla de negocio que pertenece a una etapa del proceso de negocio.
3. Es posible generar comentarios de programa, así como en la lógica del programa. Muchos programadores no incluyen comentarios en el código de programa que escriben, lo que lo hace en muchas ocasiones muy difícil de entender. Sin embargo, el Repositorio contiene una gran cantidad de metadatos que se puede utilizar para crear las observaciones pertinentes cuando el código se genera.

Además de crear una rutina de generación de código para cada tipo de regla, en el diseño del motor de reglas se debe crear una rutina de generación de código para cada paso del proceso de negocio. Esta rutina es la necesaria para la lógica del programa del paso del proceso de negocio.

### CONCLUSIONES

Como resultado de esta investigación se diseñó un sistema que permite gestionar reglas de negocio como elementos distintivos de la organización. La aplicación desarrollada tiene el poder expresivo necesario para que las reglas puedan ser expresadas con una versión acotada del lenguaje natural muy cercana a la lógica de predicados de primer orden. De manera adicional fueron construidas las estructuras de metadatos necesarias para transformar las reglas de negocio desde su forma declarativa hacia componentes de software que las ejecuten. Fueron creados los mecanismos computacionales necesarios para inyectar los componentes asociados a las reglas dentro del código perteneciente a las aplicaciones del dominio específico de la Aduana, explotando las particularidades de la arquitectura para que las modificaciones a las reglas puedan elaborarse en tiempo real y “en caliente”. Se desarrolló un prototipo funcional para un visualizador de procesos de negocio que permite insertar las reglas de manera gráfica y se desarrolló un analizador sintáctico de XPDL a ezComponents workflow XML.

### RECOMENDACIONES

A lo largo de esta investigación fueron identificados un grupo de elementos que no fueron abordados al no ser parte del alcance inicial pero se considera que son elementos que incrementarían la utilidad de este trabajo:

- Extender la estructura interna del motor para que pueda utilizar stack base query language.
- Extender los tipos de reglas que pueda interpretar el motor.
- Definir una ontología para la representación semántica de las reglas de forma tal que estas puedan ser interpretadas de manera automática.



## BIBLIOGRAFIA

**Appleton, Daniel S. 1984.** *Business Rules: The Missing Link*. 1984. págs. 145–150.

**Burlton, Roger T. 2001.** *Business Process Management: Profiting from Success*. Indianapolis : Sams Publishing, 2001.

**Chisholm, Malcolm. 2007.** *How to Build a Business Rules Engine*. San Francisco : Morgan Kaufmann Publishers, 2007.

—. **2001.** *Managing Reference Data in Enterprise Databases*. 2001.

**CMS-Spain. 2009.** La nueva versión 5.1 del motor de reglas de negocio ILOG JRules funcionará tanto en Java como .NET. *sitio Web de CMS-Spain*. [En línea] 04 de Junio de 2009. <http://www.ecm-spain.com/noticia.asp?IdItem=5930>.

**Cobo Rodríguez, Jose Antonio. 2008.** *Línea Base Arquitectónica para el Polo Sistemas Tributarios y de Aduanas*. Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2008.

**Code Java. 2010.** WebSphere ILOG JRules. *sitio Web de Code Java*. [En línea] 4 de Enero de 2010. [Citado el: 20 de Abril de 2010.] [http://www.codejava.org/detalle\\_notas.htm?idnota=97806&destacada=1](http://www.codejava.org/detalle_notas.htm?idnota=97806&destacada=1).

**Delta-[R]. 2008.** onRules, la nueva generación de BRMS (Business Rules Management System). *sitio Web de Delta-[R]*. [En línea] 2008. [Citado el: 23 de Abril de 2010.] <http://www.delta-r.com/es/BRMS-onrules.html>.

- Graham, Ian. 2006.** *Business Rules Management and Service Oriented Architecture*. s.l. : John Wiley & Sons, Ltd, 2006.
- GUIDE Project. 1998.** *Business Rules Group*. s.l. : GUIDE Project, 1998.
- . **1995.** *Defining Business Rules ~ What Are They Really?* s.l. : GUIDE Project, 1995.
- Medina, Carlos. 2007.** Reglas de Negocio en Biz Talk Server, Parte I. *sitio Web de Microsoft*. [En línea] 27 de Mayo de 2007. [Citado el: 5 de Abril de 2010.] <http://www.microsoft.com/latam/technet/articulos/tn/2007/may-15.msp>.
- Moran, Chris. 2004.** Java Developer's Journal. *Java Developer's Journal*. [En línea] 3 de Junio de 2004. [Citado el: 23 de Abril de 2010.] <http://java.sys-con.com/node/45082>.
- Morgan, Tony. 2002.** *Business Rules and Information Systems*. s.l. : Addison-Wesley, 2002.
- OMG. 2008.** *Semantics of Business Vocabulary and Business Rules* . s.l. : OMG, 2008.
- Oracle. 2008.** *Oracle Data Sheet: Oracle SOA Suite*. 2008.
- Planeta Código. 2005.** Drools I. Introducción a los motores de reglas de negocios. *sitio Web de Planeta Código*. [En línea] 30 de Noviembre de 2005. [Citado el: 15 de Abril de 2010.] <http://www.planetacodigo.com/planeta/1308/drools-i-introduccion-a-los-motores-de-reglas-de-negocios/>.

**Ross, Ronald G. 1987.** *Entity Modeling: Techniques and Application*. s.l. : Database Research Group, Inc., 1987.

—. **2003.** *Principles of the Business Rule Approach*. s.l. : Addison Wesley, 2003.

—. **1994.** *The Business Rule Book*. Primera edición. s.l. : Business Rule Solutions, LLC., 1994.

—. **1997.** *The Business Rule Book*. Segunda edición. s.l. : Business Rule Solutions, LLC., 1997.

**Ross, Ronald G y Lam, Gladys S. W. 2000.** *Capturing Business Rules*. 2000.

**Taylor, James y Raden, Neil. 2007.** *Smart (Enough) Systems*. s.l. : Prentice Hall, 2007.

**Trilles, Pablo. 2008.** *LA SINERGIA DE LOS BPMR (MOTORES DE REGLAS DE NEGOCIO) CON BPM Y SOA*. 2008.

**Vera, Eduardo. 2010.** Motores de reglas de negocio: pasado presente y futuro. *sitio Web de Window Live*. [En línea] 5 de Febrero de 2010. [Citado el: 15 de Abril de 2010.]

<http://everac99.spaces.live.com/blog/cns!B2296C467C188917!1688.entry>.

**Von Halle, Barbara. 2002.** *Business Rules Applied: Building Better Systems Using the Business Rule Approach*. s.l. : Wiley Computer Publishing, 2002.