

3  
2ej



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Contaduría y Administración

## MODELADO DE REGLAS DEL NEGOCIO EN LA CONSTRUCCION DE SISTEMAS DE INFORMACION

SEMINARIO DE INVESTIGACION INFORMATICA

Que para obtener el título de  
LICENCIADO EN INFORMATICA

p r e s e n t a

RAUL BENITEZ SARABIA



Asesor del Seminario: ING. MIGUEL SANTIAGO SUAREZ CASTARON

México D. F.

TESIS CON  
FALLA DE ORIGEN

275326

1999



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTUDIO DE CASO  
SALUD  
2023

*A mis padres que con su amor, desvelos  
y sacrificios hicieron de mi un hombre de  
bien.*

*A mis hermanos: Omar, Erick y Cristina  
por su cariño y confianza.*

*A Adriana quien siempre estuvo a mi lado  
para brindarme su comprensión y apoyo en  
los momentos en que la cuesta era más difícil.*

*A Bety por su gran amistad y apoyo para hacer  
esto posible*

*A mis grandes amigos por su valiosa  
amistad.*

*Gracias*

*A la Universidad  
Gran recinto de cultura, por haberme dado la oportunidad de formarme en ella.*

*A todos los profesores  
Por haber compartido conmigo sus conocimientos y experiencias.*

*A Santiago Suárez  
Por su apoyo y sus valiosos consejos que hicieron posible la culminación de esta tesis.*

*A DIOS  
Porque a él le debo todo lo que soy.*

*Gracias*

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I. LA ORGANIZACIÓN</b>	
1.1. Introducción .....	6
1.2 Concepto .....	7
1.2.1 Eficiencia y Eficacia .....	8
1.3 Tipos de Organización .....	9
1.4 Propósito y Objetivo de la Organización .....	11
1.4.1 Clasificación de los Objetivos .....	12
1.5 Recursos y Areas funcionales de la Organización .....	15
1.6 Enfoque de Sistemas en la Organización .....	17
1.6.1 Elementos básicos de los sistemas .....	18
1.6.2 Tipos de Sistemas .....	19
1.6.3 La Organización vista como un sistema abierto .....	21
1.6.3.1 Características de los sistemas abiertos .....	21
1.6.4 La organización como un conjunto de subsistemas .....	23
1.6.5 Medio Ambiente Organizacional .....	25
1.6.6 Planeación .....	28
1.6.6.1 Tipos de planes .....	29
<b>CAPITULO II. LAS REGLAS DE NEGOCIO</b>	
2.1 Introducción .....	31
2.2 Concepto .....	33
2.3 Antecedentes de las Reglas del Negocio .....	36
2.4 Características de una Regla del Negocio .....	40
2.5 Objetivo de las Reglas del Negocio .....	42
2.6 La Tecnología de Reglas del Negocio basada en los datos .....	44

---

2.7 El rol del analista .....	45
2.8 Identificando las Reglas del Negocio .....	46
2.8.1 Identificando los Business Rambling .....	46
2.8.2 Identificando los Business Rule Statements .....	48
2.8.3 Identificando las Reglas del Negocio .....	48
<b>CAPITULO III. CLASIFICACION DE LAS REGLAS DEL NEGOCIO</b>	
3.1 Introducción .....	52
3.2 Terms .....	54
3.3 Facts .....	55
3.4 Constraints .....	56
3.4.1 Stimulus/Response Rules .....	59
3.4.2 Operation Constraint Rules .....	60
3.4.3 Structure Constraint Rules .....	61
3.5 Derivation Rules .....	63
3.5.1 Inference Rules .....	64
3.5.2 Computation Rules .....	65
3.6 Reglas del Negocio más complejas .....	66
3.6.1 Asumir Reglas del Negocio implícitas en el Modelo de Datos .....	68
3.6.2 Relaciones Opcionales .....	70
<b>CAPITULO IV. MODELANDO REGLAS DEL NEGOCIO</b>	
4.1 Introducción .....	75
4.2 Lenguaje para modelar reglas del negocio .....	76
4.2.1 Tipos de Reglas del Negocio Atómicas .....	77

---

4.2.2 Esquema sintáctico para modelar una regla del negocio .....	78
4.2.2.1 Restricciones de Integridad y Condiciones .....	79
4.2.2.2 Anclas .....	80
4.2.2.3 Correspondientes .....	82
4.2.2.4 Interpretes .....	84
4.2.2.5 Calificadores .....	86
4.2.2.6 Valor Derivado .....	88
4.2.3 Verificadores de Instancias .....	90
4.2.3.1 Tipo de regla mandatoria .....	91
4.2.3.2 Tipo de regla limitada .....	91
4.2.4 Verificadores de Tipo .....	92
4.2.4.1 Tipo mutuamente .....	93
4.2.4.2 Tipo mutuamente exclusivo .....	93
4.2.4.3 Tipo mutuamente inclusivo .....	94
4.2.5 Verificadores de Posición .....	95
4.2.5.1 Más pequeño, más grande y posición .....	95
4.2.5.2 Tipos: cronológico, más viejo y más reciente .....	97
4.2.6 Verificadores Funcionales .....	99
4.2.6.1 Funcional .....	100
4.2.6.2 Unicas .....	100
4.2.6.3 Fluctuación .....	101
4.2.6.4 Ascendentes .....	101
4.2.6.5 Descendentes .....	102
4.2.6.6 No reutilizables .....	103

---

4.2.7 Evaluadores comparativos .....	104
4.2.7.1 Igual a .....	104
4.2.7.2 No igual a .....	105
4.2.7.3 Mayor que .....	106
4.2.7.4 Mayor que o igual que .....	106
4.2.7.5 Menor que .....	107
4.2.7.6 Menor que o igual que .....	107
4.2.8 Operadores Matemáticos .....	108
4.2.8.1 Reglas Calculadas .....	109
4.2.8.2 Operador Suma .....	109
4.2.8.3 Operador Resta .....	110
4.2.8.4 Operador Multiplicación .....	111
4.2.9 Controladores de proyección .....	112
4.2.9.1 Habilitadores .....	112
4.2.9.2 Copiadores .....	114
4.2.9.3 Ejecutantes .....	115
4.3 Interconectando reglas del negocio .....	116
4.3.1 Reglas Derivadas .....	116
4.3.2 Reglas de inclusión .....	117
<b>CAPITULO V . SISTEMA DE SEGURIDAD</b>	
5.1 Introducción .....	119
5.2 Sistema de Seguridad (SISE) .....	120
5.3 Políticas y Reglas del Negocio del SISE .....	121
5.4 Reglas del Negocio con deficiencias en su representación por el Modelo de Datos .....	126

5.5 Identificando los Business Rambling, Business Rule Statements y Reglas del Negocio .....	131
5.6 Modelando las Reglas del Negocio del SISE .....	137
<b>CONCLUSIONES</b> .....	149
<b>APENDICE A: Códigos de Programas en Lenguaje “C” y Stored Procedures</b> .....	152
<b>BIBLIOGRAFIA</b> .....	170

## INTRODUCCIÓN

Si analizamos la historia del hombre nos daremos cuenta que desde su origen éste siempre ha vivido en grupo ante la necesidad de coordinar esfuerzos para alcanzar un objetivo. Así por ejemplo, en la prehistoria el hombre se organizaba para cazar, recolectar alimentos, construir sus viviendas, etc., pero conforme el hombre evolucionó también su forma de organización se volvió más difícil, dando paso a sistemas de organización más complejos, esto se observa en las majestuosas pirámides del viejo Egipto y del México Prehispánico, así como en los sistemas de gobierno, agricultura, comercio, educación, religión y guerra que las antiguas civilizaciones desarrollaron. Este proceso evolutivo dio paso a las modernas organizaciones que hoy en día conocemos y de las que los individuos forman parte activa.

Las organizaciones se caracterizan por su conducta orientada a un objetivo y a que todas las áreas que las conforman comparten ese objetivo para asegurar el éxito de la empresa. Pero ¿qué es lo que asegura el éxito de una empresa?. El hecho mismo de fijar un objetivo y de organizarse para dirigir las acciones en la consecución del mismo, implica que deban establecerse y seguirse ciertas reglas.

Pongamos como ejemplo el Juego de Fútbol Soccer, sabemos que cada jugador tiene asignada una posición dentro del campo y ciertas funciones. Por ejemplo, el medio campista tiene como objetivo manejar y conducir el balón en el medio campo y de colocar los pases a los delanteros; el defensa se encargará de no dejar pasar a los delanteros del equipo contrario y los delanteros del equipo de anotar goles. Pero lo más importante es que tanto los integrantes del equipo anfitrión como los contrincantes jugarán de acuerdo a ciertas reglas, tales como no tocar el balón con la mano a menos de ser el portero, no cometer faltas dentro del área chica de la portería, etc. Todo este conjunto de reglas asegura la sana convivencia y el logro de un objetivo común: " El Campeonato".

Pero, ¿Ha imaginado qué sería del juego si cada quién siguiera sus propias reglas ?, verdaderamente se convertiría en un campo de guerra y no aseguraría en ningún momento el trabajo en equipo, ni el alcance de las metas. Algo parecido sucede en una empresa cuando cada departamento, área funcional o grupo de trabajo aplica su propio universo de reglas para obtener resultados propios, ocasionando que ésta se fragmente. Es evidente, entonces, que la consecución de los objetivos organizacionales no depende de los resultados de un grupo específico, sino de aquellos obtenidos de manera colectiva.

De esta manera se llega al punto que ocupa a la presente Tesis: "Las Reglas del Negocio". Como ya es conocido las organizaciones y el mundo en general están viviendo la era de la Información, sabemos de la existencia de grandes y complejos sistemas de información en los que las empresas basan su operación diaria. De ahí, la importancia que las organizaciones definan cuál es la misión del negocio, es decir, ¿ Qué mercado explotará?, ¿ Quiénes serán sus clientes ?, ¿ Qué productos y/o servicios brindará?, etc. Una vez definida esta situación, deberá identificar y establecer las reglas, normas y políticas bajo las cuales basará su operación, lo que permitirá al negocio ir en una dirección común.

Anteriormente, los analistas de sistemas describían a la organización en términos de sus estructuras de datos, del flujo de información a través de ésta y de las acciones que se tomaban a lo largo de éste flujo. De alguna manera esto constituía ciertas reglas del negocio, pero no precisamente aquellas que determinaban la manera de operar del mismo. Además, en aquellos tiempos, el desarrollo del software se llevaba a cabo en términos de las herramientas que dominaban el mercado en esos días, actualmente el desarrollo de sistemas debe evolucionar de acuerdo a lo que en la realidad cambia.

La mayoría de las empresas han implementado reglas del negocio a través de sistemas de información, pero el problema es que durante el desarrollo de aplicaciones el responsable de cada área funcional interpreta las reglas del negocio desde su punto de vista, llama a la gente de sistemas, le dice que es lo que quiere y le pide se lleve a cabo el desarrollo de la aplicación. El problema no se detiene ahí, sino que llegado el momento de dar mantenimiento al sistema nadie sabe qué reglas del negocio se implantaron y cuáles no. El problema se vuelve más complejo, porque nadie sabe cómo encontrar esas reglas y modificarlas.

El propósito de la presente tesis es entonces, mostrar que **Las Reglas del Negocio** pueden ser modeladas en la implantación de un sistema de información, es decir, separarse y modelarse durante las etapas de análisis y diseño, permitiendo la generación de sistemas de información de alta calidad y flexibilidad que permitan asegurar el éxito de la organización. Una vez iniciado el desarrollo a partir de la etapa de análisis hasta la de construcción se identificarán y documentarán dichas reglas del negocio con el fin de asegurar la calidad del software.

El hecho de conocer bien las Reglas del Negocio no limita la creatividad de quién las cumple, por el contrario, implica que la organización puede tomar ventaja de eso. Por ejemplo, sabemos que el mantenimiento de sistemas representa un gran problema para las empresas, ya que significa tiempo y dinero. La ventaja de tener claramente definidas e identificadas las reglas del negocio es que los nuevos requerimientos (reglas) se podrían incorporar al sistema con mayor rapidez dando a este la habilidad para asimilar los cambios y por lo tanto, ser capaz de adaptarse. Al final, la empresa podría responder con mayor rapidez a los cambios del mercado y mantenerse en la competencia. El hecho de que la empresa pueda identificarlas rápidamente, medir el impacto del cambio y llevar a cabo el mantenimiento dice mucho de la calidad del producto final.

Con el propósito de mostrar la utilidad que ofrece el enfoque de reglas del negocio en la generación de software de alta calidad, el presente trabajo se ha dividido en 5 capítulos, a través de los cuales se presentan los elementos necesarios para modelar la reglas del negocio.

En el Capítulo I, se define el concepto de organización, los diversos tipos de organismos sociales, así como sus objetivos. Se destaca también la importancia que juegan: el medio ambiente, las políticas, normas y reglas organizacionales, como factores determinantes en su forma de operar para de esta manera conocer y entender los factores propios de la organización que afectan de manera directa el ciclo de vida de un sistema de información. Una vez que éstos conceptos han sido comprendidos es necesario entender el concepto de Reglas del Negocio, que es precisamente el objetivo del Capítulo II, el cual está orientado a entender el concepto de Reglas del Negocio dentro del desarrollo de sistemas de información, apreciar sus características, entender su objetivo y el cambio que proponen en la forma de concebir la construcción de aplicaciones, así como la manera de identificarlas en una sesión de requerimientos, destacar su importancia en la etapa de diseño y mostrar algunas de sus ventajas.

Para dar continuidad al proceso de identificación de Reglas del negocio, es necesario establecer un esquema de clasificación que permita documentarlas y ayude a cotejar que todos los tipos de reglas se analizaron durante el proceso de identificación. Este es precisamente el propósito del Capítulo III, en el que se presenta un esquema de clasificación tomando como base el trabajo de diversos autores, a efecto de contar con un esquema lo más completo posible.

En el Capítulo IV, defino el lenguaje que utilizaré para modelar las reglas del negocio, es decir, la sintaxis básica, la cual incluye: los símbolos utilizados para representar las reglas en el modelo de datos, su semántica, las convenciones para su uso y el esquema básico para modelar una regla, es decir, los elementos necesarios. Una vez establecida la sintaxis para modelar las reglas se explican

los tipos de reglas más comúnmente utilizadas, acompañadas de un pequeño ejemplo en el que se ilustra la forma de modelarlas.

Finalmente con el objeto de aplicar el cambio propuesto por el enfoque de reglas del negocio en la construcción de sistemas de información, el Capítulo V presenta un pequeño caso práctico que ilustra la utilidad de modelar y especificar formalmente las reglas del negocio. El caso presentado en este capítulo no contempla la implantación de las reglas del negocio que se modelan, ya que esto es una actividad meramente mecánica que cualquier equipo de desarrolladores puede realizar una vez que recibe los resultados de las actividades de análisis y diseño; etapas que son determinantes en la calidad del producto final.

De esta manera, espero que la tesis contribuya a elevar la calidad del software y que se refleje en beneficios para las empresas y por lo tanto, para la sociedad.

## LA ORGANIZACION

### 1.1 Introducción

Se sabe que el ser humano es social por naturaleza. Su conducta tendiente a organizarse y cooperar es inherente. La historia de la humanidad así lo demuestra, desde que el hombre apareció sobre la faz de la tierra hasta nuestros días ha formado parte de una organización o grupo.

Desde que nacemos hasta que morimos la mayor parte de nuestro tiempo la pasamos en organizaciones. La familia que nos vio nacer y de la que formamos parte es por sí misma una organización. Las instituciones educativas y de trabajo consumen gran parte de nuestro tiempo, en ellas surgen relaciones informales y formales con otros individuos, mismas que llevamos hasta nuestros hogares y a nuestras actividades recreativas como lo son: el juego de foot ball, las salidas al campo, el coro de la iglesia, etc.

Es evidente que pertenecemos a una gran diversidad de grupos y organizaciones, pero sea cual fuere su tipo, formal o informal todas tienen aspectos en común: el logro de un objetivo, recursos, planes, metas y reglas para la consecución de ese objetivo, así como un medio ambiente con el que interactúan y se desenvuelven.

Las organizaciones no son ajenas al hombre, ellas están íntimamente ligadas a nuestras vidas. Están en todos lugares y a todos nos afectan. Su naturaleza está determinada por el conjunto de valores, gente, recursos, costumbres, etc., que la sociedad (medio ambiente) le impone.

## **1.2 Concepto**

Sabemos que las organizaciones invaden a la sociedad y a nuestra existencia diaria, todos nos desenvolvemos en diferentes organizaciones; como la familia, la escuela, la universidad, la oficina, la iglesia, los clubes deportivos, etc.,

El concepto de organización debe su razón de ser al hecho de que el individuo es incapaz de satisfacer todas sus necesidades y deseos por si mismo, y es sólo a través de la coordinación de esfuerzos que se llega a la conclusión de dichos objetivos y necesidades.

El hecho de nacer en las organizaciones, de desarrollarse y convivir con ellas en nuestra vida cotidiana nos da una idea clara de lo que ellas son, pero para dar una definición más formal de éstas es importante revisar la opinión de algunos autores.

**Max Weber.** "Un sistema de actividad continua encaminada a un propósito de tipo particular"

**Amitai Etzioni.** "Unidades sociales que persiguen fines específicos: su misma razón de ser es el servicios de estos fines."

**Edgar H. Schein.** "La coordinación racional de las actividades de cierto número de personas, que intentan conseguir una finalidad y objetivo común y explícito, mediante la división de las funciones y del trabajo, y a través de una jerarquización de la autoridad y la responsabilidad."

**Bernard Phillips.** " Un grupo constituido deliberadamente para la búsqueda de objetivos específicos."

Como se observa, existen diversas opiniones acerca del concepto de una organización, pero analizando las definiciones anteriores se aprecia que estas tienen elementos en común, tales como: grupo de individuos, objetivos específicos y relaciones entre los integrantes, por lo tanto, concluyo que una organización es un ente o sistema formado por recursos humanos, técnicos y materiales que interactúan coordinadamente con el fin de lograr de manera eficaz y eficiente un objetivo común.

### **1.2.1 Eficiencia y Eficacia**

Es importante establecer la diferencia entre estos conceptos, ya que como hemos visto la coordinación de esfuerzos en las organizaciones está orientada a este propósito.

Cuando hablo de la productividad individual u organizacional me estoy refiriendo al concepto de eficiencia, es decir a la relación existente entre los resultados de una actividad y los medios que se utilizaron para llevarla a cabo. Cuando una organización utiliza la menor cantidad de recursos para generar productos o servicios de alta calidad se dice que es eficiente. Se concluye entonces que la *eficiencia* es el logro de los objetivos satisfactoriamente mediante la utilización de la menor cantidad de recursos.

Por otra parte la organización no sólo debe ser eficiente, sino que también debe cumplir con las metas organizacionales, es decir, ser eficaz. Para entender la diferencia entre eficiencia y eficacia, utilizaré la definición que Peter Drucker, que dice: La eficiencia significa "hacer correctamente las cosas", es decir, utilizar el mínimo de recursos para obtener sus metas, y la eficacia significa "hacer las cosas correctas", dicho en otras palabras, escoger los objetivos apropiados para saber hacia donde dirigir los esfuerzos.

### 1.3 Tipos de Organizaciones

La construcción de tipologías ha surgido con el propósito de realizar un análisis comparativo entre las organizaciones y el medio social, así como la relación que guardan éstas entre sí. En el Cuadro 1 presento diferentes tipología y los autores que las proponen.

Autor	Tipos de organizaciones	Ejemplos
Talcott Parsons	a) De producción b) De metas políticas c) Integrativas d) De mantenimiento de patrones	a) Empresas (bienes y servicios) b) Partidos, sindicatos c) Policía, bomberos d) Educativas
Renate Maynts (con base a sus objetivos)	a) Las que se limitan a la coexistencia de sus miembros. b) Las que actúan de manera determinada sobre las personas admitidas. c) Las que buscan el logro de cierto resultado hacia fuera.	a) Círculos de recreación o esparcimiento. b) Escuelas, Universidades, hospitales, prisiones c) Administración, policía, partidos, asociaciones benéficas
Peter Blau y William Scott	a) De beneficio mutuo. b) Firms comerciales. c) De servicios. d) De bienestar común.	a) Sindicatos, partidos políticos, sectas, clubes. b) Industrias, bancos, almacenes. c) Hospitales, escuelas. d) Oficinas gubernamentales, policía, bomberos, institutos de investigación científica.

Autor	Tipos de organizaciones	Ejemplos
Amitai Etzioni (con base en la función control)	<ul style="list-style-type: none"> <li>a) Coercitivas</li> <li>b) Utilitarias</li> <li>c) Normativas</li> </ul>	<ul style="list-style-type: none"> <li>a) Prisiones</li> <li>b) Empresas (bienes y servicios)</li> <li>c) Iglesias.</li> </ul>
Daniel Katz y Robert Kahn	<ul style="list-style-type: none"> <li>a) Productivas</li> <li>b) De mantenimiento</li> <li>c) De adaptación</li> <li>d) Político - administrativas</li> </ul>	<ul style="list-style-type: none"> <li>a) Empresas (bienes y servicios)</li> <li>b) Escuelas, sectas religiosas</li> <li>c) Universidades, institutos de investigación</li> <li>d) Partidos políticos, sindicatos, organizaciones de profesionales</li> </ul>

Cuadro No. 1

#### 1.4 Propósito y Objetivo de la Organización

Toda organización se construye alrededor de la búsqueda de objetivos específicos. No es posible concebir la idea de formar un grupo sin saber cual es el fin que éste persigue, ya que los esfuerzos concertados serían en vano y posiblemente se convertiría en una anarquía donde cada quién haría valer su autoridad en virtud de conseguir sus propios objetivos. La palabra objetivo comprende misiones, propósitos, metas, fines, cuotas y plazos.

Desde el punto de vista del enfoque de sistemas el objetivo de la organización define la finalidad para la cual fueron ordenados todos los componentes y sus relaciones.

Según Reyes Ponce los fines de la empresa deben analizarse de acuerdo al siguiente Criterio:

- **Fin inmediato**, es decir, la producción de bienes y servicios para un mercado. Se utiliza para legitimar y justificar la función de la organización en la sociedad.
- **Fines mediatos**, que se busca con esa producción de bienes o servicios. Si es un objetivo de tipo económico estaríamos hablando de un Organismo Socio - Económico (empresa privada), o bien, si su objetivo primordial no es el lucro nos referimos a los organismos netamente sociales (Alcohólicos Anónimos o las empresas públicas). Los fines mediatos representan el motivo para la actividad de la organización.

Los objetivos son la razón de ser de la empresa, de ahí la importancia de definir claramente lo que es un objetivo. Para tal efecto nos basaremos en la definición que Reyes Ponce nos da, "La palabra objetivo (de ob - jactum), implica la idea de algo hacia lo cual se lanzan o dirigen nuestras acciones"<sup>1</sup>.

---

<sup>1</sup> Agustín REYES PONCE, Administración de Empresas Teoría y Práctica., (México, Limusa, 1979), p. 108.

Considerando el criterio de Reyes Ponce y agregando un elemento a éste, los objetivos se pueden analizar bajo tres perspectivas: *los objetivos sociales impuestos a la sociedad* (bienes y servicios); *los objetivos de los integrantes de la organización* (remuneración, desarrollo profesional, satisfacción, etc.) y *los objetivos del sistema* (ganancias, labor social).

Los esfuerzos de la empresa por alcanzar sus objetivos afectan la capacidad de ésta para recibir insumos y recursos de la sociedad y justificar así su función con la misma. Influyen en sus relaciones con el medio ambiente, fijan los estándares para evaluar el éxito y ayudan a determinar las tecnologías requeridas. Sientan además las bases para la especialización de las tareas, los esquemas de autoridad, las redes de comunicación, etc.

#### **1.4.1 Clasificación de los Objetivos**

Es importante que al estudiar los objetivos organizacionales se haga una distinción entre los objetivos oficiales y los objetivos operativos, ya que "Los objetivos más importantes para entender el comportamiento organizacional no son los oficiales, sino los que se incluyen en las principales políticas de operación y en las decisiones diarias del personal... Estos objetivos serán conformados por los problemas y actividades particulares que la organización debe destacar, ya que estas tareas determinan las características de aquellos que dominarán la organización."<sup>2</sup>

---

<sup>2</sup> Charles PERROW, *The Analysis of Goals*, (American Review, december 1996) p. 854.

Se entiende por **objetivos oficiales** a aquellos establecidos en términos amplios y ambiguos para justificar las actividades de la organización; los **objetivos operativos** son los que en realidad se persiguen. Esta aseveración está estrechamente relacionada con la cadena de medios para lograr los objetivos. Por ejemplo: los objetivos oficiales establecidos a un nivel más alto en la jerarquía organizacional requieren ciertos medios para su cumplimiento, estos medios se convierten entonces en subobjetivos del siguiente nivel desarrollándose objetivos o metas operativas más específicas. Ejemplo: El objetivo de un departamento de ventas será lograr un alto nivel de ventas, y para conseguirlo, se deberán establecer medidas de control de calidad, disminuir los costos de distribución y ganar mercado a la competencia.

Es importante también que quienes forman parte de una organización o grupo puedan diferenciar entre los objetivos personales dentro de la institución de aquellos propios de esta. En el *Cuadro 2* se muestra una clasificación de objetivos.

Objetivo	Características
A Corto Plazo	Deben cumplirse en un período corto de tiempo, generalmente un año o menos.
A largo plazo	Aquellos que se obtienen más allá de los cinco años
Individuales	Son los objetivos personales de los miembros de una organización, pueden ser económicos, de desarrollo profesional, psicológicos, etc.
Colectivos	Los que son comunes a un grupo de personas.
Generales	Aquel que es más amplio en relación a los que están subordinados a él.
Particulares	Son los medios para la consecución de un objetivo de un nivel jerárquico más elevado.
Colaterales	Aquellos que sin ser buscados directamente como los principales, se obtienen al desarrollar los objetivos particulares.

Cuadro No. 2

Toda organización cuenta con una estructura donde se establecen las líneas de comunicación, autoridad y especialización de las tareas; los objetivos generales a lo largo de toda esta estructura se traducen en metas u objetivos operativos cada vez más específicos, esto es lo que se conoce como jerarquía de objetivos, ya que cada organización cuenta con áreas funcionales específicas como: ventas, finanzas, producción, recursos humanos, etc., cada una de las cuales tiene objetivos específicos relacionados con su función dentro de la empresa.

Considerando la existencia de una jerarquía de objetivos se concluye que un objetivo de nivel general es cualitativo y una meta u objetivo operativo es cuantitativo (mensurable). En cuanto más se asciende en la jerarquía de la empresa los objetivos se vuelven más cualitativos.

### 1.5 Recursos y Areas Funcionales de la organización

Desde el punto de vista de la escuela de sistemas los recursos de la organización se encuentran dentro de ésta; son los medios que utiliza para hacer su trabajo. Son las cosas que la organización puede cambiar y utilizar para su propio provecho. En términos administrativos los recursos básicos de una organización son los siguientes:

- **Recursos Humanos**
- **Recursos Financieros**
- **Recursos Materiales**

La empresa puede decidir como manejar sus recursos para su propio beneficio, es decir "estos determinan la base sobre la cual los actos administrativos pueden ser moldeados."

Una vez que la organización cuenta con los recursos necesarios para emprender su operación, es necesario especificar que tareas deben realizarse, quién las tiene que hacer, cómo deben agruparse, quién reporta a quién, etc., es decir deben establecerse las relaciones entre los componentes o partes de la organización, dicho en otras palabras: la organización formal, compuesta de varios departamentos, puestos administrativos y sus relaciones.

En la medida en que una estructura organizacional se encuentre bien diseñada facilitará el logro de los objetivos. Es importante enfatizar que la estructura interna se ve afectada por el medio ambiente. Las empresas establecen departamentos para manejar sus insumos del exterior y los productos que dirigen a sectores específicos del medio. La tecnología que la organización utilice también impacta en la estructura formal. Por ejemplo el subsistema técnico es el principal determinante en el área de sistemas.

Así mismo el subsistema psicosocial afecta y al mismo tiempo está influido por la estructura. La gente de bajo nivel educativo y poca experiencia laboral responde de manera diferente a los arreglos estructurales.

El patrón básico de muchas estructuras organizacionales gira alrededor de tres actividades fundamentales:

- **Producción**
- **Ventas**
- **Finanzas**

La complejidad en cuanto al número y tipo de departamentos o áreas varía según la complejidad de la organización, pero esencialmente tienen que cumplirse las tres actividades anteriores para que sobreviva la empresa. Estas actividades son importantes, porque la mayoría de las empresas están involucradas en la generación de bienes o servicios, que tienen que ser usados por otros, por lo tanto estos deben ser distribuidos o comercializados; esto es, debe encontrarse alguien que desee el producto o servicio y este dispuesto a aceptarlo. Así mismo para producir y distribuir el producto o servicio se necesita reunir y mantener un capital y para esto es necesario la participación de la gente del área de finanzas.

Dentro del ámbito de las tres áreas fundamentales y la complejidad de la empresa surgen diversas unidades organizacionales, cada una de las cuales tienen objetivos específicos relacionados con su función dentro de la empresa.

## 1.6 Enfoque de Sistemas en la Organización.

La importancia de aplicar el enfoque de sistemas a la organización es que esta puede ser analizada como un sistema social formado por subsistemas, además permite realizar una descripción concreta y objetiva de la misma a fin de describir su comportamiento y explicar su secuencia de acciones. Por lo tanto, es importante definir lo que es un sistema. Existen numerosas definiciones, pero para efectos de la presente tesis utilizaré la siguiente: Un **sistema** es un conjunto de partes que interactúan de manera coordinada para el logro de un objetivo; donde cualquier acción que produzca un cambio en una de sus partes, probablemente producirá un cambio en todos sus otros elementos.

Tomando como base esta definición podemos afirmar que la empresa puede ser entendida como un sistema y cada uno de sus departamentos o áreas funcionales como subsistemas del suprasistema (la empresa), pero extendiendo este análisis podemos pensar que la organización es un subsistema de un sistema mayor: *la sociedad*. Este análisis se puede hacer tan amplio como se desee.

La teoría de sistemas destaca las siguientes características en un sistema:

- **Propósito u objetivo:** Todo sistema debe tener una razón de ser.
- **Globalismo o totalidad:** Cualquier acción que produzca un cambio en alguno de los elementos del sistema invariablemente generará un cambio en todos sus demás elementos.
- **Entropía:** Es la tendencia que tienen los sistemas a dejar de ser útiles. A medida que aumenta la información disminuye la entropía.
- **Homeostacia:** Es el equilibrio dinámico entre las partes integrantes del sistema. Los sistemas tienen una tendencia adaptativa con el fin de alcanzar un equilibrio interno frente a los cambios externos del medio ambiente.

### 1.6.1 Elementos básicos de los sistemas

Un sistema como ya se mencionó es un conjunto de elementos que en forma ordenada y armoniosa interactúan con funciones específicas para el logro de un fin común. Podemos apreciar entonces que las relaciones son los lazos que unen los elementos del sistema. Esas relaciones están dadas por los elementos o parámetros que caracterizan al sistema.

1. **Insumos o entradas:** Son las unidades de influencia o de partida del sistema. Es la energía (suministros) necesaria para la operación.
2. **Proceso:** Es la transformación o mecanismos de conversión de los insumos en salidas o resultados.
3. **Productos o salidas:** Son los resultados de un proceso, los cuales deben ser congruentes con los objetivos del sistema.
4. **Retroalimentación o feedback:** Función del sistema que tiene como objetivo comparar la salida con un criterio o estándar previamente establecido. El fin de la retroalimentación es el control, es decir la medición de los resultados actuales y pasados en relación con lo esperado, ya sea parcial o totalmente.
5. **Medio ambiente:** Es lo que está "fuera" del sistema y que determina la manera de operar del mismo, es una fuente de energía, materiales e información para el sistema. No es posible que los sistemas operen fuera de un medio específico.
6. **Fronteras:** Son las barreras entre el sistema y el ambiente. Definen el campo de acción del sistema y pueden dejar pasar mayor o menor intercambio de información con el ambiente. Esta permeabilidad de las fronteras define el grado de apertura del sistema en relación con el ambiente.

## 1.6.2 Tipos de Sistemas

Criterio	Sistema	Características
Por la determinación de su funcionamiento	Probabilísticos	<ul style="list-style-type: none"> <li>• Existe incertidumbre sobre su futuro; ejemplo: las empresas, ya que el comportamiento humano nunca es totalmente previsible</li> </ul>
	Determinísticos	<ul style="list-style-type: none"> <li>• Aquellos cuyo funcionamiento puede predecirse con certeza, es decir, que un cambio específico en una de sus variables producirá un resultado particular.</li> </ul>
Por sus límites	subsistemas	<ul style="list-style-type: none"> <li>• Las partes de un sistema pueden ser a su vez un sistema por separado</li> </ul>
	Suprasistemas	<ul style="list-style-type: none"> <li>• Todo sistema forma parte de uno mayor que se llama suprasistema</li> </ul>
Por la comunicación	Abiertos	<ul style="list-style-type: none"> <li>• Aquellos que están en constante interacción dual con el ambiente, es decir, que lo influyen y por él son influenciados. Para sobrevivir deben adaptarse a las condiciones del ambiente.</li> </ul>
	Cerrados	<ul style="list-style-type: none"> <li>• Presentan poco intercambio de materia y energía con el medio ambiente</li> </ul>

Criterio	Sistema	Características
Por su constitución	Físicos	<ul style="list-style-type: none"> <li>• Cuando están compuestos por equipos, maquinaria y por objetos y cosas reales.</li> </ul>
	Abstractos	<ul style="list-style-type: none"> <li>• Cuando están compuestos por conceptos, planes, hipótesis e ideas</li> </ul>
Por su dinamismo	Estáticos	<ul style="list-style-type: none"> <li>• Cuando no reacciona ante los influjos del medio ambiente; ejemplo: una silla.</li> </ul>
	Dinámicos	<ul style="list-style-type: none"> <li>• Aquellos que tienen un cambio constante en su estructura y funcionamiento. Ejemplo: la sociedad.</li> </ul>
	Homeostáticos	<ul style="list-style-type: none"> <li>• Aquellos que conservan el dinamismo dentro de ciertos límites y tienen la posibilidad de autoregularse.</li> </ul>
Por su dependencia	Dependientes	<ul style="list-style-type: none"> <li>• Su funcionamiento está dado totalmente en función de otro y su medio ambiente. Su posibilidad de autocontrolarse y autodirigirse es nula y sus metas están determinadas por el exterior</li> </ul>
	Independientes	<ul style="list-style-type: none"> <li>• Cuyo funcionamiento está regido por sí mismo, pueden modificarse por la libertad que tienen para decidir.</li> </ul>
	Interdependientes	<ul style="list-style-type: none"> <li>• Aquellos que dependen uno del otro.</li> </ul>

Cuadro No. 3

### 1.6.3 La organización vista como un sistema abierto

Las organizaciones como se ha visto mantienen una interacción dinámica con su medio ambiente, clientes, proveedores, competidores, instituciones gubernamentales, sindicatos, agrupaciones ecológicas y muchos más, de ahí la importancia de estudiarla como un sistema abierto, ya que una de las principales características de éstos como se verá a continuación es su constante intercambio de información con el medio ambiente.

#### 1.6.3.1 Características de los Sistemas Abiertos

1. Presentan relaciones de intercambio continuo con el ambiente, a través de entradas y salidas. Influencia al ambiente y es por él influenciado.
2. Son eminentemente adaptativos, esto es, para sobrevivir deben reajustarse constantemente a las condiciones del medio.
3. No pueden vivir aislados.
4. Gracias a que pueden adaptarse tienen la capacidad de crecimiento.
5. Pueden competir con otros sistemas.

Las organizaciones poseen todas las características de los sistemas abiertos como se ilustra en la *Figura 1*. Ellas asimilan materiales y energía (entradas) de su medio ambiente, los procesan y generan una salida en productos y servicios; reacciona a los estímulos del ambiente y cambia sus mercados, clientes, proveedores, empleados, estructura, etc., siendo capaz de reproducirse en empresas subsidiarias.

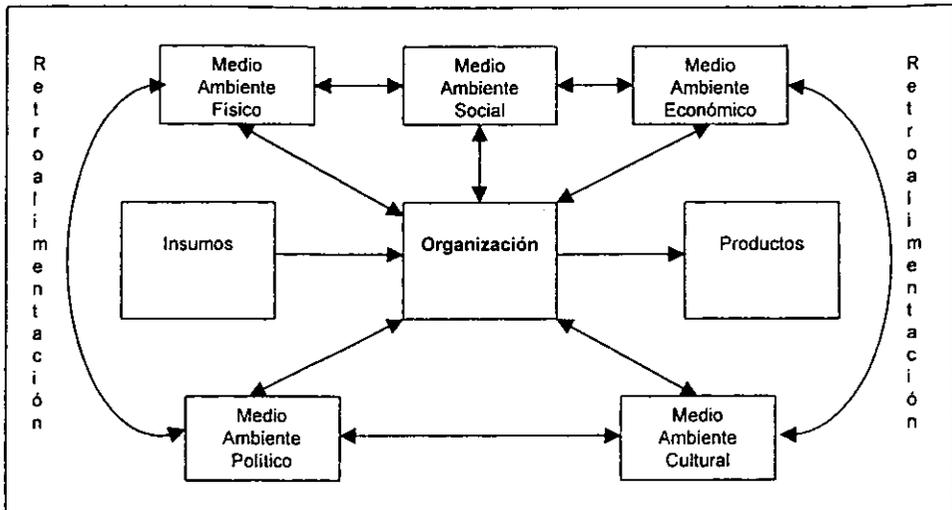


Figura No. 1 La organización vista como un Sistema Abierto

De lo anterior se concluye que una empresa u organización debe ser **adaptativa** a fin de seguir existiendo en su medio, como lo afirma Kleber T. "Un sistema organizacional rígido no podrá sobrevivir en la medida en que no consiga responder eficazmente a los cambios continuos y rápidos del ambiente".

Las características de los sistemas abiertos no sólo pueden ser aplicadas a las organizaciones, sino que pueden extenderse a cualquier tipo de sistemas que no sean cerrados incluyendo a los sistemas de información. Estos últimos reciben datos de entrada (insumos) que procesan para dar una respuesta (salidas) a las peticiones de los usuarios, adicionalmente éstos requieren de constantes cambios que responden a nuevas necesidades de los usuarios, en gran parte determinadas por cambios en el medio ambiente. Estos cambios permiten que el producto (sistema) continúe siendo útil (adaptativo).

#### 1.6.4 La organización como un conjunto de subsistemas.

De acuerdo con la teoría de sistemas se puede afirmar que la organización se compone usualmente de uno o más subsistemas y las capacidades totales de ésta dependen de las capacidades de cada subsistema. Los más importantes son:

**Subsistema de Objetivos:** El objetivo es la meta final hacia la cual la empresa dirige sus esfuerzos. El objetivo al aplicarse se convierte en la razón de existencia de la empresa. El subsistema de objetivos es la etapa de planeación, período en el cual, la organización intenta determinar la forma de coexistir con el ambiente en tanto establece normas de funcionamiento y logra objetivos específicos.

**Subsistema de Valores:** Representan la cultura de los individuos. Es un marco perceptivo relativamente permanente que conforma e influencia la naturaleza general de la conducta, los valores son significativos como determinantes en la toma de decisiones de implantación de objetivos y la estrategia que se tomará para realizarlos. La organización toma muchos de sus valores del amplio medio sociocultural y también ejerce su influencia en los valores de la sociedad.

**Subsistema Técnico:** Se refiere a los conocimientos necesarios para el desarrollo de las tareas. La tecnología organizacional incluye técnicas, equipos, procesos e instalaciones utilizados en la transformación de los insumos en productos. El subsistema técnico está determinado por los propósitos de la organización y varía conforme a los requerimientos de la tarea, con frecuencia la tecnología prescribe el tipo de estructura organizacional y afecta al sistema psicosocial. Ejemplo: al introducir un sistema computacional en la automatización de ciertas tareas de la empresa generalmente los empleados siempre se muestran renuentes y temerosos ante las innovaciones tecnológicas, esto genera un cambio en un su comportamiento y forma de pensar.

**Subsistema Psicosocial:** Lo forman los individuos y grupos de interacción. Consiste en el comportamiento individual y la motivación, relaciones de función y posición, dinámica de grupos y sistemas de influencia. Es afectado por el medio ambiente externo y la tecnología, tareas, valores, actividades y estructura interna de la organización.

**Subsistema Administrativo:** Está relacionado básicamente con la toma de decisiones, fijación de objetivos, diseño de la estructura organizacional y para planear, organizar, dirigir y controlar el efecto de personal en la organización.

### 1.6.5 Medio Ambiente Organizacional

Como se dijo anteriormente, un sistema no puede existir fuera de un medio ambiente específico, por lo tanto, la organización como sistema abierto necesita interactuar con su medio para adquirir insumos, procesarlos y generar productos.

Anteriormente las organizaciones no prestaban mayor atención al medio ambiente social, ya que éste era más estable en comparación con el dinamismo y complejidad del actual. Generalmente sus clientes, competidores y proveedores no representaban números significativos, además las grandes organizaciones que dominaban el mercado encontraban poca, o en la mayoría de los casos una nula competencia. Hoy día si las empresas desean crecer y sobrevivir deben ser capaces de relacionarse con el medio (una organización depende de otras organizaciones para sobrevivir) y de adaptarse a él. Ellas requieren de los recursos y los mercados del medio para continuar en operación. Las condiciones legales, económicas, sociales y políticas generan requerimientos que la empresa debe atender. Se define, entonces, el medio organizacional "como los objetos, personas y demás organizaciones que rodean a un sistema organizacional específico, incluyendo las fuentes de insumo que utiliza la empresa y los consumidores de los bienes producidos".<sup>3</sup>

**Medio Ambiente Económico:** Son aquellos factores económicos o de mercado que influyen sobre las operaciones de las empresas, así por ejemplo, tenemos las preferencias de los clientes respecto a determinadas mercancías y servicios y su capacidad para pagarlos, la demanda de parte de los clientes, los precios más bajos, la disponibilidad y el costo del capital, el nivel de productividad de una sociedad y la magnitud de los mercados:

---

<sup>3</sup> Warren B. BROWN, Teoría de la Organización y la Administración Enfoque Integral, (México, Limusa, 1990), p. 58

Las políticas fiscales y monetarias del gobierno repercuten en la elaboración de productos y la oferta de servicios y por ende en las condiciones del mercado.

**Medio Ambiente Tecnológico:** El nivel de la tecnología tiene un alto grado de determinación sobre que productos y servicios serán ofrecidos, así como, en determinar que equipo se utilizará y cómo se administrarán las operaciones. Los cambios en la tecnología frecuentemente inciden en las acciones de la competencia.

**Medio Ambiente Jurídico:** Tiene que ver con aquellas consideraciones constitucionales, la naturaleza del sistema legal, jurisdicciones de las distintas instituciones gubernamentales, leyes específicas acerca de la formación, tasas impositivas y control de las organizaciones, que afectan al negocio

**Medio Ambiente Político:** Está conformado por el clima político general de la sociedad, el grado del poder político. La naturaleza de la organización política (grado de descentralización), el sistema de partidos políticos. Ejemplo: la práctica de aplicar impuestos sobre las utilidades de las empresas, etc.

**Medio Ambiente Cultural:** Se refiere a los antecedentes históricos, ideológicos, valores y normas de la sociedad; a los puntos de vista sobre las relaciones de autoridad, esquemas de liderazgo, relaciones interpersonales, razonamiento, ciencia y tecnología. El ambiente cultural define la naturaleza de las instituciones sociales.

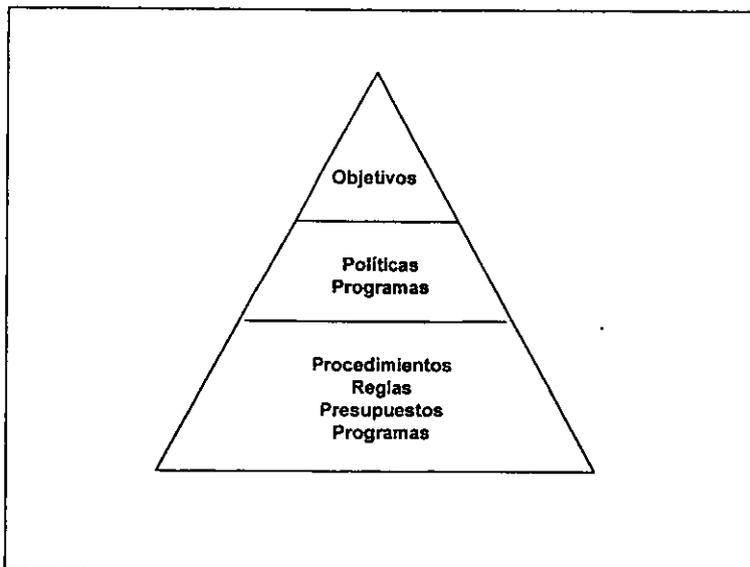
Toda organización cualquiera que fuere su tipo se ve afectada por las condiciones ambientales anteriormente descritas (**Medio Ambiente Relevante**). Pero también existen influencias ambientales específicas que son diferentes para cada organización y se conocen como su **Medio Ambiente Específico**. Algunas de las más importantes son:

1. **Los proveedores:** Considerados una de las grandes dependencias de las organizaciones, ya que el insuficiente abastecimiento afectaría su operación y supervivencia.
2. **Los consumidores o clientes:** Son aquellas personas y grupos que utilizan o consumen los productos o servicios que genera la empresa. Son esenciales para la supervivencia de la organización.
3. **Los competidores:** Toda organización tiene competidores, enfrentan la competencia de otras empresas o firmas que ofrecen productos y servicios similares.

### 1.6.6 Planeación

La organización como cualquier sistema debe definir su fin último, desde el punto de vista del proceso administrativo, la etapa en la cual deben estipularse las misiones globales, definirse las políticas, estrategias, normas, y procedimientos es, la de planeación.

La planeación es la función administrativa básica y sirve para determinar los objetivos y metas organizacionales, así como los medios para alcanzarlos, estos son los planes en acción. En la *Figura 2* se muestran los elementos básicos de la etapa de planeación.



*Figura No. 2 Elementos de la planeación*

Durante el proceso de planeación se debe analizar el medio ambiente a fin de identificar los factores externos (políticos, económicos, sociales, etc.) que pueden incidir en la organización, y poder generar planes flexibles que pueden facilitar la adaptación de la organización.

#### 1.6.6.1 Tipos de Planes

- **Planes Estratégicos:** Abarcan programas y políticas. Los programas especifican ciertos objetivos y las maneras de lograrlos; las políticas, los lineamientos generales bajo los cuales las actividades deben llevarse a cabo. Establecen también los métodos necesarios para asegurarse de que las políticas y programas sean ejecutados.
- **Planes Tácticos:** Incluyen elementos como son procedimientos, reglas, presupuestos y programas.

A continuación se describen algunos de los conceptos involucrados en el proceso de planeación:

1. **Objetivos:** Fines hacia donde se dirigen las actividades de la organización y de los individuos. Representan la razón de existencia de la empresa.
2. **Políticas:** Son criterios generales que tienen por objeto orientar la acción. Son el objetivo en acción. Presentan las siguientes características:
  - Son los medios básicos para delegar autoridad.
  - Establece los límites dentro de los cuales las actividades de la empresa deben desarrollarse.
  - Se caracterizan por tener un margen de libertad, es decir, quien es responsable de hacerlas cumplir cuenta con cierta libertad para decidir.

3. **Programas:** Son los planes en los que además de fijarse los objetivos y la secuencia de operaciones, también se establece el tiempo requerido para realizar cada una de sus partes. Los programas pueden ser generales y particulares, es decir, se pueden referir a toda la organización o a un departamento específico. Existen también los planes a largo y corto plazo, los primeros tienen una duración de hasta un año y los últimos a más de un año.
  
4. **Procedimientos:** Planes que especifican la secuencia cronológica más eficiente para obtener los resultados en cada función de la empresa. Son guías para la acción. Estos existen en cualquier nivel de la empresa, pero más específicamente en el área de operación.
  
5. **Reglas:** Son restricciones o sentencias imperativas de algo que debe realizarse, ya sea de manera específica o genérica. Las reglas genéricas se conocen como políticas. Las reglas específicas no permiten márgenes de libertad en la toma de decisiones, simplemente se analiza si el caso particular se encuentra comprendido dentro de la regla.
  
6. **Presupuestos:** Es un estado de los resultados esperados. Su principal característica es la determinación cuantitativa de los elementos programados. Son utilizados como instrumentos de planeación y control.

## **LAS REGLAS DE NEGOCIO**

### **2.1 Introducción**

Como mencione en la introducción el mundo se encuentra gobernado por reglas de diferente índole. Por ejemplo, tenemos reglas en el trabajo, la escuela, la familia, la guerra, etc. Pero en este trabajo, se estudiarán las reglas que rigen la conducta organizacional desde el punto de vista del desarrollo de sistemas de información; por ser las organizaciones en donde pasamos la mayor parte de nuestro tiempo, con quien interactuamos a diario y lo más importante, el desarrollo que los profesionales de la tecnología de la información tenemos dentro de ellas en la construcción de dichos sistemas.

Así entonces, se tiene que las reglas que gobiernan la conducta de la empresa, que en lo sucesivo llamaré Reglas del Negocio no representan nada nuevo para la organización, ya que son inherentes a ella, son su lenguaje natural de expresión y convivencia con su medio ambiente.

Desde su nacimiento las organizaciones siempre han basado su operación en el cumplimiento de sus Reglas del Negocio, ya sea llevándolas a cabo mediante procesos manuales, administrativos o bien, mediante procesos automatizados como resultado de la introducción de las computadoras en las cuatro últimas décadas, con el objetivo de incrementar el valor del negocio.

La mayoría de estas reglas se encuentran implantadas (automatizadas) en sistemas de información, de hecho para la gente responsable del área de sistemas éstas tampoco representan nada innovador, ya que las han identificado, capturado y automatizado, pero de una manera inconsciente, es decir, no se les ha considerado como un factor que deba reconocerse en una metodología de desarrollo de sistemas, dicho en otras palabras, no se les ha dado un tratamiento más formal, para identificarlas, definir las, documentar las e implantar las.

El propósito de este capítulo está precisamente orientado a entender el concepto de Reglas del Negocio dentro del desarrollo de sistemas de información, apreciar sus características, entender su objetivo y el cambio que proponen en la forma de concebir la construcción de aplicaciones, así como, la manera de identificarlas en una sesión de requerimientos, destacar su importancia en la etapa de diseño y mostrar algunas de sus ventajas.

## **2.2 Concepto**

Para poder establecer el concepto de Regla de Negocio primero será necesario revisar el concepto de Regla desde el punto de vista administrativo, para lo cual, retomaré la definición establecida en el primer capítulo que dice: Las reglas son restricciones o sentencias imperativas de algo que debe realizarse, ya sea de manera específica o genérica. Las reglas genéricas se conocen como políticas, las reglas específicas no permiten márgenes de libertad en la toma de decisiones.

En realidad el concepto de Regla del Negocio no cambia desde el enfoque de sistemas como se puede apreciar en las siguientes definiciones:

- De acuerdo con Ron Ross quien es uno de los más notables estudiosos del tema, sostiene que una Regla del Negocio es "Una restricción o evaluación que se ejecuta con el propósito de mantener la integridad de los datos"<sup>4</sup>
- Según Barbara Von Halle "Una Regla del Negocio es simplemente una sentencia que está orientada a influenciar la conducta del negocio, independientemente si ésta es llevada a cabo por una persona o un sistema"<sup>5</sup>.
- Para Mitchell I. Kramer, "Las Reglas del Negocio son implantaciones programáticas de las políticas y prácticas de un negocio".<sup>6</sup>

---

<sup>4</sup> Ronald G. ROSS, *The Business Rule Book* (Boston, Massachusetts, Data Base Newsletter, 1997), p.11.

<sup>5</sup> Barbara VON HALLE, *The Business Rule Roadmap*, En *Database Programming & Design* (1997), p.1

<sup>6</sup> Mitchell I. KRAMER, *Business Rule Automating, Business Policies and Practices*, En Patricia Seybolds *Distributed Computing Monitor*, p.3.

- Jane Conkey nos dice que "Una Regla del Negocio es una sentencia que define o restringe algún aspecto del negocio".<sup>7</sup>

Antes de continuar con el concepto de Regla del Negocio y de establecer una definición más completa del mismo, quiero aclarar que los términos negocio y empresa serán utilizados a lo largo de éste trabajo de manera indistinta para referirme a la Organización . Lo anterior se aplica de igual forma para los términos Reglas del Negocio y Reglas de Gestión.

Analizando las definiciones anteriores se puede apreciar que ellas coinciden en el uso de términos como políticas, prácticas, sentencias y restricciones orientadas a determinar la manera de operar de la empresa, de lo cual establezco la siguiente definición de Regla del Negocio empleada en lo sucesivo. "Una Regla del Negocio es una sentencia o declaración de una política, práctica o restricción encaminada a regular el comportamiento de la organización".

**De acuerdo con la definición anterior, a continuación se listan algunos ejemplos de Reglas del Negocio:**

1. *Únicamente a aquellos clientes con un saldo promedio mensual de \$2000 en cuenta de cheques, pueden participar en los sorteos semanales de un automóvil.*
2. *Un cliente no puede rentar videos, si este tiene en su poder 5 videos.*
3. *A los consultores de sistemas que participen en proyectos se les pagará una comisión del 1 % del monto total de la factura, pero sólo después de 30 días de cobrada la factura.*
4. *Un cliente del Banco que contrató el servicio de Vía Telefónica no podrá realizar transferencias de fondos entre cuentas de cheques si éste ha excedido el monto máximo transferible por día que es de \$10,000.*

---

<sup>7</sup> Barbara VON HALLE y Jane CONKEY, Explaining the Business Connection, En Database Programming and Design (december 1996), p.i.

5. *A un cliente de banco se le aplicará un cargo de \$10 por uso de cajeros automáticos.*
6. *Únicamente los gerentes de nuestras sucursales podrán autorizar los descuentos sobre ventas.*
7. *Los descuentos sobre ventas se realizan únicamente por montos mayores a \$50,000.*
8. *Ningún cliente podrá retirar dinero de su cuenta de crédito, si esta se encuentra sobregirada.*
9. *Todo cliente del banco debe tener al menos una cuenta de cheques, pero no más de tres.*
10. *Un cliente moroso es aquel que tiene más de tres facturas vencidas.*

Las Reglas del Negocio pueden estar presentes en cualquiera de los siguientes elementos:

- a) **Requerimientos del usuario**
- b) **Prácticas del Negocio**
- c) **Políticas de la empresa**
- d) **Procedimientos**
- e) **En el sistema legal del país (Medio Ambiente Jurídico)**
- f) **En las regulaciones fiscales (Medio Ambiente Económico y Político)**

La importancia de destacar la manera en que las reglas del negocio deben ser manejadas, se debe al siguiente hecho: cuando las reglas se implantan, éstas se traducen en reglas cada vez más atómicas que cuando toman una forma restrictiva tienen como propósito mantener la integridad y acceso a los datos persistentes de la organización. Y es precisamente en una información veraz y oportuna, en la que la organización sustenta la toma de decisiones, además la gran mayoría de los usuarios de las diferentes áreas comparten la información almacenada en sus bases de datos.

Para los analistas y desarrolladores de sistemas estas reglas del negocio también llamadas Reglas de Gestión, no representan nada nuevo, ya que las han estado capturando, registrando e implantando en varios lenguajes de programación, utilizando para ello una gran variedad de métodos, técnicas y herramientas. El problema con el tratamiento que se ha venido dando a las reglas del negocio, es que su identificación, manejo e implantación no se ha hecho de manera formal, lo que ha originado que éstas se oculten o entierren profundamente en código procedural, dando como resultado aplicaciones poco flexibles, incapaces de responder de manera rápida a las necesidades de la empresa producto de los cambios del mercado o de su medio.

### **2.3 Antecedentes de las Reglas del Negocio.**

Por mucho tiempo las Reglas del Negocio han sido embebidas en la lógica de la aplicación, de tal manera, que cuando las políticas o necesidades de la organización cambian resulta difícil reflejar dichos cambios en los sistemas donde se encuentran implantadas dichas reglas.

Cuando las Bases Relacionales surgieron, éstas se convirtieron en la norma para el desarrollo de aplicaciones, las reglas del negocio comenzaron a ser embebidas en Stored Procedures, Triggers y lenguajes host como C y Cobol, por lo tanto, fueron escritas y codificadas en diversas implantaciones de SQL.

Los triggers y stored procedures tienen la ventaja de la modularidad, estos separan las Reglas del Negocio y las reglas técnicas de manipulación de datos de la lógica de la aplicación, pero su principal desventaja es que su desarrollo y mantenimiento resulta difícil, porque éstos se tienen que expresar en un lenguaje SQL propietario, cuya sintáxis y manejo es complejo. Además su especificación es más técnica que orientada al negocio. La mayoría de estos triggers implantan reglas de gestión, pero su objetivo principal es asegurar la integridad y consistencia de los datos.

Cuando se llega el momento de realizar mantenimiento a las aplicaciones, el desarrollador analiza los códigos y encuentra muy difícil el deducir que reglas del negocio están implantadas por los triggers y stored procedures, ya que estos tienden a ser monolíticos y difíciles de entender, generando lo que se conoce como código espagueti.

Con el objeto de ilustrar lo anteriormente dicho en la *Figura 3* muestro un fragmento de código en el que se tienen implantadas las reglas del negocio que deben cumplirse para realizar una Inversión a Plazo en una institución bancaria a través del sistema *El Banco en su Casa* que permite la realización de transacciones bancarias mediante el uso de un teléfono de tonos.

El código se encuentra desarrollado en lenguaje C , con SQL Embebido de Informix y el uso del API de desarrollo del middleware TUXEDO.

Las reglas que se encuentran implantadas son las siguientes:

**ENTRADA:** \*Numero de Cuenta de Cheques  
\*Moneda  
\*Plazo en días (en el rango 0 <= Plazo <= 9999)  
\*Importe en pesos  
\*Tipo de Tasa = (FIJA o VARIABLE)  
\*Modo de Pago de Intereses = (FIN\_DE\_MES o ANIVERSARIO o AL\_VENCIMIENTO)  
\*Manejo de Capital al Vencimiento = (REINV\_MISMO\_PZO o REINV\_UN\_DIA o ABONO\_CTA)  
\*Manejo de Intereses al Vencimiento =(REINV\_MISMO\_PZO o REINV\_UN\_DIA o ABONO\_CTA)

**SALIDA:** \*Tasa de intereses  
\*Folio de Atlantisis

**Notas:**

1. Esta transacción no hace el cargo a la cuenta de cheques en Atlantisis.
2. Si el Plazo es 0 días, solo se utilizan los cuatro primeros argumentos de entrada a la transacción y a los argumentos restantes debe de asignárseles cero.
3. Si 1 <= Plazo <= 30, al argumento Modo de Pago de Intereses debe de asignársele AL\_VENCIMIENTO, al argumento Tipo de Tasa debe de asignársele FIJA.
4. Adicionalmente, cuando Plazo = 1 el Manejo de Capital al Vencimiento y Manejo de Intereses al Vencimiento debe tomar el valor ABONO\_CTA.

5. Cuando Plazo es mayor a 30, son válidas todas las combinaciones de los argumentos Tipo de Tasa, Modo de Pago de Intereses, Manejo de Capital al Vencimiento y Manejo de Intereses al Vencimiento.

**ERRORES:**

- \*Cuenta de cheques inexistente
- \*Código de moneda inválido
- \*La fecha de vencimiento es día no hábil
- \*La fecha de vencimiento sobrepasa límite establecido
- \*Monto fuera de los límites establecidos por el Banco
- \*Servicio fuera de horario
- \*Servicio no disponible

```
#include "vtasimbolic.h"
#include "mixvtasun.h"
#include "apiVTA.h"
#include "sys/times.h"

int VtaMddInvPzo(tTXflags iteTXflags, teVtaMddInvPzo *pteVtaMddInvPzo, tsVtaMddInvPzo *ptsVtaMddInvPzo)
{
    /* tipo de buffer utilizado */
    struct viMddInvPzo *pviVtaMddInvPzo;
    struct voMddInvPzo *pvoVtaMddInvPzo;
    int iRet, iRegreso;
    long IRlen;
    struct tms timebuf;

#ifdef DEBUGTIME
    long ITi, ITf;
    ITi=times(&timebuf);
#endif
    iRegreso = FRACASO;

    /* alojamiento de los buffers de Entrada */
    pviVtaMddInvPzo = (struct viMddInvPzo *)talloc("VIEW", "viMddInvPzo", sizeof(struct viMddInvPzo));
    /* Analisis sobre el resultado del alojamiento */
    if( pviVtaMddInvPzo == NULL ){
        userlog("VtaMddInvPzo: %s", tpstrerror(tperrno));
        return( iRegreso );
    }

    /* alojamiento de los buffers de Salida */
    pvoVtaMddInvPzo = (struct voMddInvPzo *)talloc("VIEW", "voMddInvPzo",
        sizeof(struct voMddInvPzo));
    /* Analisis sobre el resultado del alojamiento */
    if( pvoVtaMddInvPzo == NULL ){
        userlog("VtaMddInvPzo: %s", tpstrerror(tperrno));
        return( iRegreso );
    }
    /* Carga de los datos de entrada al buffer de peticion */

    pviVtaMddInvPzo->iCtoVta = pteVtaMddInvPzo->iCtoVta;
    pviVtaMddInvPzo->shVpsId = pteVtaMddInvPzo->shVpsId;
    strcpy( pviVtaMddInvPzo->sCuenta, pteVtaMddInvPzo->chCuenta.val );
    pviVtaMddInvPzo->shMoneda = pteVtaMddInvPzo->shMoneda;
    pviVtaMddInvPzo->shPlazo = pteVtaMddInvPzo->shPlazo;
    pviVtaMddInvPzo->decImporte = pteVtaMddInvPzo->decImporte;
    pviVtaMddInvPzo->shTipTas = pteVtaMddInvPzo->shTipTas;
    pviVtaMddInvPzo->shModPagInt = pteVtaMddInvPzo->shModPagInt;
    pviVtaMddInvPzo->shManCapVen = pteVtaMddInvPzo->shManCapVen;
    pviVtaMddInvPzo->shManIntVen = pteVtaMddInvPzo->shManIntVen;
```

```

if( pteVtaMddInvPzo->shPlazo == 0){
  if((pteVtaMddInvPzo->shTipTas != 0)||
    (pteVtaMddInvPzo->shModPagInt != 0) ||
    (pteVtaMddInvPzo->shManCapVen != 0) ||
    (pteVtaMddInvPzo->shManIntVen != 0) ){
    iRegreso=ARGS_INVAL;
    return( iRegreso );
  }
}
if( (pteVtaMddInvPzo->shPlazo >=1) && (pteVtaMddInvPzo->shPlazo <=30) ){
  if( (pteVtaMddInvPzo->shModPagInt != MOD_PAG_INT_VEN) ){
    iRegreso= MOD_PAG_INTES_INVAL;
    return(iRegreso);
  }
  if ( pteVtaMddInvPzo->shTipTas != TIP_TASA_FIJA ){
    iRegreso= TIP_TAS_INVAL;
    return( iRegreso );
  }
  if( pteVtaMddInvPzo->shPlazo == 1){
    if( pteVtaMddInvPzo->shManCapVen != MAN_CAP_VEN_ABON_CTAC ){
      iRegreso=MAN_CAP_VEN_INVAL;
      return(iRegreso);
    }
  }
  if( pteVtaMddInvPzo->shManIntVen != MAN_PAG_INT_ABON ){
    iRegreso=MAN_INTES_VEN_INVAL;
    return( iRegreso );
  }
}
}
/* Llamado al Servicio */
iRet = tpcall("VTAMddInvPzo", (char *)pviVtaMddInvPzo, 0,
  (char **)pvoVtaMddInvPzo, (long *)&iRlen, iteTXflags);
if( iRet < 0 ){
  if( tperrno == TPESVCFAIL )
    iRegreso= tpurcode;
  else
    userlog("VtaMddInvPzo: %s", tpsterror(tperrno));
}
else{
  iRegreso = ?purcode;
  /* Carga de los Datos a las Estructuras de Salida */
  ptsVtaMddInvPzo->decTasa = pvoVtaMddInvPzo->decTasa;
  ptsVtaMddInvPzo->iFolio = pvoVtaMddInvPzo->iFolio;
}
GrabaTiempos("VtaMddInvPzo", pviVtaMddInvPzo->iCtoVta, ITf-ITi,0);

/*Liberacion del buffer de peticion, y regreso del codigo de la funcion*/
tpfree((char *)pviVtaMddInvPzo);
tpfree((char *)pvoVtaMddInvPzo);
return( iRegreso );
}

```

Figura No. 3

Como se puede apreciar las reglas de negocio se encuentran enterradas en líneas de código procedural e implantadas en diferentes capas de la arquitectura del sistema "El Banco en su Casa", lo que dificulta su identificación en futuros mantenimientos.

Otro factor que repercute en la flexibilidad del sistema es que sólo el responsable del área de sistemas que desarrollo la aplicación sabe de la existencia de estas reglas, ya que nunca se formalizó su identificación y documentación en un proceso de análisis

Para fundamentar de una manera más sólida lo expuesto anteriormente, en el apéndice A muestro el código de algunos Stored Procedures y programas con SQL embebido en el lenguaje host C que ayudarán a entender el concepto de reglas del negocio y sus antecedentes.

#### **2.4 Características de una regla del negocio**

Una característica es un rasgo que identifica o distingue a un objeto o cosa de los demás. Por ejemplo, los rasgos performance, funcionalidad, robustez y complejidad son distintivos de un sistema de información, como lo son: estado, conducta e identidad de un objeto en el paradigma de Desarrollo de Sistemas Orientado a Objetos.

Para el enfoque de Reglas del negocio las características más sobresalientes de una regla son:

- 1. Están implantadas mediante una actividad combinada que involucra a personas y sistemas.**
- 2. Son explícitas, es decir, que éstas deben ser expresadas en un lenguaje natural (escrito) o gráfico.**
- 3. Siempre se encuentran vivas y listas para ser ejecutadas, ya que estas se traducen en el sistema nervioso electrónico de la organización, reflejando exactamente la política actual del negocio**

4. **Su naturaleza es Declarativa.** Una regla del negocio es declarativa y no procedural, es decir, explica por qué una decisión es tomada o por qué una acción se ejecuta. La regla no describe el cómo. Cuando un proceso del negocio se encuentra operando, muchas reglas del negocio se encuentran trabajando e influenciando la manera de operar del proceso (conducta).

Considerando que los sistemas son un conjunto de funciones, se puede pensar análogamente que una regla del negocio es una función, de tal forma, que una función representa lo que la regla de negocio necesita como entrada, para arrojar una salida. Si sólo es especificada la entrada y la salida, la regla del negocio se expresa declarativamente, pero si se describe el cómo la regla del negocio realiza lo que tiene que hacer, entonces ésta se expresa proceduralmente.

En esta última característica se puede apreciar uno de los cambios que el enfoque de reglas del negocio propone como una de las bases en el desarrollo de sistemas: En lugar de confiar en las restricciones tecnológicas utilizando un enfoque procedural se debe concentrar la atención en las necesidades de la organización. Dicho correctamente es poner las prioridades del negocio por encima de las prioridades tecnológicas. Cada regla del negocio es una política o práctica operacional que se ejecuta en la empresa, pero que no se refiere de ninguna forma a una estrategia de implantación o una tecnología específica.

## **2.5 Objetivo de las Reglas del Negocio**

Los sistemas de información automatizan procesos que se llevan a cabo de manera manual, con el objetivo de incrementar lo que los estudiosos de la información han llamado "Valor del Negocio", es decir, aumentar las ganancias. El término no sólo se refiere a las finanzas, también tiene que ver con la simplificación de tareas, mejorar la calidez y rapidez en el servicio al cliente, mejorar la imagen de la empresa, fomentar su crecimiento, etc.

El enfoque de reglas del negocio sugiere un cambio en el desarrollo de sistemas, orientado precisamente a incrementar el valor del negocio, ya que como afirma Von Halle, cada una de ellas tiene un valor en dinero para el negocio. Este cambio en la manera de concebir la construcción de sistemas pretende dar un tratamiento más formal a las reglas considerándolas como un elemento o componente importante que debe ser tomado en cuenta en cualquier metodología de desarrollo de software, que permita a los desarrolladores concentrarse en las necesidades de la organización durante la etapa de diseño de una aplicación, crear un conjunto de reglas orientadas al negocio que definan los procesos y funciones de éste. Básicamente consiste en aislar y modularizar dichas reglas. Como se muestra en la *Figura 4*.

Organización

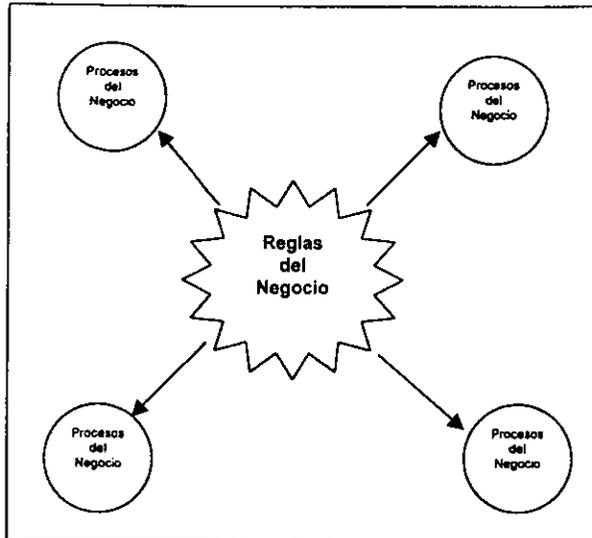


Figura No. 4

Las ventajas que se tienen al separar el modelado, diseño y especificación de las reglas del negocio son:

- **Reusabilidad:** Ya que las políticas y prácticas del negocio se aplican a toda la organización
- **Rápida respuesta a las nuevas necesidades del negocio y un mantenimiento más simple:** Ya no es necesario analizar miles de líneas de código para identificar las reglas, cambiarlas y probarlas. Estas se encuentran aisladas permitiendo que su cambio sea más rápido, como ya no se encuentran embebidas en la lógica de la aplicación, ya no es necesario probar toda la aplicación, basta con probar únicamente las reglas.
- **Disminución de inconsistencias:** que actualmente existen en la interpretación de las reglas del negocio que se encuentran embebidas en varios procesos del negocio o aplicaciones de usuario final.

- **Independencia tecnológica:** Las reglas del negocio no se encuentran implantadas en la aplicación de usuario final (front end), ni en el nivel de datos (back end). Si la organización decide adquirir una nueva herramienta visual para la presentación de información al usuario final o elige una Base de Datos (BD) diferente, este enfoque permite el cambio con un impacto menor al actual.

## **2.6 La Tecnología de Reglas del Negocio Basada en los Datos**

La organización basa la toma de decisiones en una información veraz y oportuna, por lo que es necesario mantener la correctez (integridad) de los datos persistentes de la empresa, esto se logra gracias a que un subconjunto del total de las Reglas del Negocio evalúan los resultados arrojados por los procesos y determinan si éstos se encuentran dentro de la regla, dicho en otras palabras, si éstos están permitidos. El resto de las reglas están orientadas a determinar la manera de operar (nivel operativo) de la empresa, mediante la explotación de los datos persistentes.

Lo anterior sugiere entonces, que las reglas del negocio se encuentren orientadas a los datos, ya que además como destaca Ron Ross, esto permite lograr los siguientes objetivos:

1. *Las reglas dependen de la especificación de tipos de datos fundamentales.*
2. *Para lograr una mayor consistencia y adaptabilidad, las reglas por si mismas deben ser especificadas de una manera discreta y no redundante.*
3. *Los usuarios generalmente no están interesados en cómo las reglas están implantadas. Lo que indica que su especificación debe ser declarativa y no procedural.*

4. *El lenguaje que tiende a ser mas entendible para la gente del negocio al momento de comunicar los resultados del análisis y diseño de sistemas es el gráfico (modelos, dibujos, diagrama, etc.)*

De acuerdo con lo anterior, la conducta de la aplicación se puede expresar como el conjunto de operaciones sobre el modelo de datos, éstas se especifican de una manera declarativa y no son otra cosa que Reglas del Negocio.

## **2.7 El Rol del Analista**

En este nuevo enfoque la principal tarea del analista de sistemas, es reducir la distancia existente entre la gente del negocio y los profesionales de la tecnología de la información, producto del problema generado durante el análisis de requerimientos del usuario en la construcción de sistemas. Además, debe tener la habilidad para identificar y extraer las Reglas del Negocio asociadas a la organización durante una junta de planeación y articulación de reglas, con el propósito de determinar lo que el sistema debe realizar. Para lograrlo, deberá convertirse en un analista de negocios capaz de hablar el mismo lenguaje que la gente del negocio a fin de poder comunicarse con ellos.

"Un enfoque de reglas del negocio en la etapa de análisis, soportado por un enfoque de reglas del negocio en la implantación nos lleva a estar más cerca de una era de continuos requerimientos"<sup>8</sup>

## 2.8 Identificando las Reglas del Negocio

Puesto que los conceptos regla del negocio, política y norma se encuentran definidos, así como, entendido el cambio propuesto por las Reglas del negocio, se tienen los elementos necesarios para poder identificarlas en una sesión de requerimientos.

En párrafos anteriores mencioné que el proceso para identificar las reglas tiene lugar durante las sesiones con la gente del negocio y los analistas de sistemas, lo que indica que éste sea interactivo y heurístico, en el cual las reglas comienzan a aparecer y emerger de manera general, es decir, como sentencias o enunciados que estipulan políticas de la empresa.

### 2.8.1 Identificando los Business Rambling

Cada una de estas sentencias o declaraciones generales expresan una práctica o política que es llevada a cabo por la empresa, sin referirse de ninguna manera a un tipo de implantación especial o tecnología. Ejemplos de políticas son las siguientes:

1. *Es política de la empresa de Consultoría en Sistemas BIT (Business Integration Technologies) incentivar económicamente a sus consultores.*

---

<sup>8</sup> Barbara VON HALLE y David PLOTKING, The Art of letting Go, En Database Programming & Design (1997), p.5

2. *Es política del departamento de Recursos Humanos contratar únicamente personal que permita a la empresa brindar servicios altamente profesionales.*
  
3. *Es política del departamento de Recursos Humanos contratar personal que ayude a la empresa a tener un ambiente de convivencia social e ideológico sano con el resto de los integrantes.*
  
4. *Es política del banco no perder en ningún momento recursos monetarios producto de las transacciones llevadas a cabo (servicios).*
  
5. *Es política del departamento de producción elevar la calidad de los productos.*
  
6. *Es política del banco mantenerse en la preferencia de los clientes otorgando para ello créditos hipotecarios.*

A menudo las políticas resultan ser difusas o "... algunas veces son claras, algunas veces ambiguas y la mayoría de las veces contienen más de una idea"<sup>9</sup>, por lo tanto, no pueden implantarse de manera directa.

---

<sup>9</sup> Defining Business Rules – What are they really?. (Guide International Corporation, 1996).

### **2.8.2 Identificando los Business Rule Statements**

Es aquí donde la labor del analista comienza, ya que, él debe estudiar estas sentencias generales conocidas como Business Rambling, para descomponerlas y poder así extraer un conjunto de reglas más específicas que hacen posible que la política se cumpla, éstas son llamadas Business Rule Statements, algunas de las cuales pueden implantarse directamente. La mayoría sin embargo, requieren de múltiples reglas para poder expresar completamente la sentencia de regla del negocio.

### **2.8.3 Identificando las Reglas del Negocio**

Al igual que los Business Rule Statements, las reglas del negocio son sentencias que definen o restringen algún aspecto del negocio, pero que no pueden ser divididas o descompuestas a un nivel más bajo, ya que se corre el riesgo de perder información importante acerca del negocio. Cada regla del negocio podría basarse en uno o más Business Rule Statements.

Con el objeto de ilustrar lo anteriormente expuesto, tomaré un ejemplo de los Business Ramblings citados y lo descompondré de acuerdo al criterio establecido

**Política**

- La empresa de consultoría en Sistemas incentiva económicamente a sus consultores

**Business Rule Statement**

1. Se pagarán comisiones por proyecto a los consultores participantes
2. Se pagará un Bono de Productividad por proyecto.

**Las Reglas de Negocio**

1. Sólo los consultores que tengan como mínimo un año de antigüedad en la empresa se harán acreedores al pago de comisiones y bono de productividad
2. Las comisiones se otorgarán en base a un porcentaje preestablecido, de acuerdo al nivel del consultor (poner tabla Principal, Senior, Junior )
3. Se tomará como base el 10% del monto de la factura sin incluir el IVA.
4. Sólo se otorgarán comisiones sobre factura cobrada.
5. El pago se realizará después de 30 días de cobrada la factura.
6. El pago de comisiones será quincenal.

**Reglas del Negocio para el Bono de Productividad**

1. El Bono de Productividad será pagado en forma quincenal, siempre y cuando los consultores cumplan con los siguiente:
  - a) Entrega semanal del reporte de actividades firmado por el cliente.
  - b) Conclusión a tiempo de las actividades asignadas al proyecto
  - c) Cumplir con los objetivos de capacitación estipulados.
  
2. El Bono de Productividad no se pagará durante los periodos de no participación en proyectos.

El ejemplo del Business Rambling del Departamento de Recursos Humanos se descompondría de la siguiente manera

**Políticas**

- *El departamento de Recursos Humanos contratará únicamente personal que permita a la empresa brindar servicios altamente profesionales.*
  
- *El departamento de Recursos Humanos contratará personal que ayude a la empresa tener un ambiente de convivencia social e ideológico sano con el resto de los integrantes.*

**Reglas del Negocio**

- a) No se contratará personal que no cuente con Título Profesional que avale sus conocimientos en el área de interés.
  
- b) No se contratará personal que no sea mexicano por nacimiento
  
- c) No se contratará personal que no haya presentado previamente un examen psicológico

En este caso no fue necesario descomponer la política en Business Rule Statement, ya que las Reglas del Negocio expresan claramente la política.

## **CLASIFICACIÓN DE REGLAS DEL NEGOCIO**

### **3.1 Introducción**

Para dar continuidad al proceso de identificación de Reglas del negocio, es necesario establecer un esquema de clasificación que permita documentarlas y ayude a cotejar que todos los tipos de reglas se analizaron durante el proceso de identificación. Desafortunadamente no existe un esquema de clasificación estándar, por tal motivo, presento un esquema tomando como base el trabajo de diversos autores con el propósito de contar con un esquema lo más completo posible.

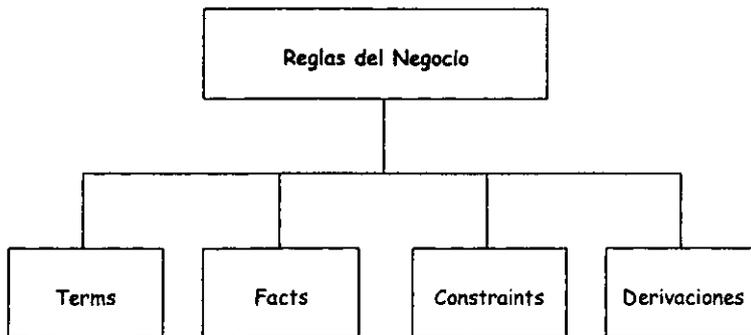
El esquema que se propone se basa en los datos (estructura), ya que la gran mayoría de los sistemas de información están soportados por un modelo aplicativo construido con la técnica entidad-relación, es decir, las bases de datos del negocio fueron diseñadas con esta técnica. Para una mejor claridad ver "Tecnología de Reglas del Negocio basada en los datos" , Capítulo 2.

Los tipos de Reglas del Negocio presentados a continuación se basan en la estructura de datos, por lo que pueden ser expresadas mediante técnicas estándares de modelado de datos, también se conocen como reglas basadas en el negocio (Business Based Rules) y son las que determinan los procesos del negocio y están relacionadas directamente al modelo de datos.

Las reglas basadas en la estructura se encuentran asociadas con objetos de la organización como son: cliente, empleado, almacén, producto, etc., y son las responsables de controlar la forma en que los objetos del negocio desarrollan funciones. Ejemplos de este tipo de reglas son:

1. *Un socio del vídeo no puede rentar más de 5 películas*
2. *Un cliente del banco no puede ser dado de baja si éste tiene cuentas de cheques activas*
3. *Un cliente del banco debe tener al menos una cuenta de cheques, pero no más de tres.*

Pertenecientes a esta clase de reglas se encuentran los siguientes tipos de reglas:

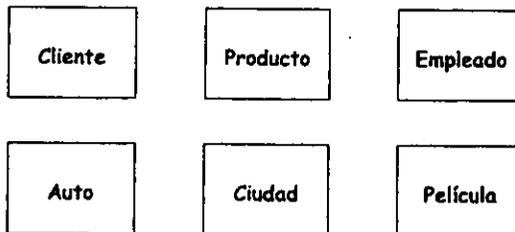


### 3.2 Terms

Son sentencias que representan elementos de importancia para el negocio, pueden existir como un concepto o bien en relación a otros aspectos de interés para la empresa. Un term es un sustantivo o frase que representa o tiene un significado particular para el negocio

Los terms son reglas del negocio que representan objetos o entidades de la organización que son de particular importancia dentro de un contexto específico de la empresa, éste define y delimita el espacio en el que se establecen los objetivos y estrategias del negocio.

"Cada term del negocio debe ser usado en al menos un contexto, y cada contexto debe de ser del uso de uno o más terms del negocio con un significado bien definido"<sup>10</sup>, Algunos ejemplos de terms se muestran a continuación.



La técnica que emplearé en lo sucesivo para representar los tipos de reglas estudiados es Entidad-Relación. Ver apéndice B.

---

<sup>10</sup> Ob. cit.

### **3.3 Facts**

Un fact es "una afirmación de que un term específico tiene una propiedad o juega un rol, o bien, que uno o más objetos participan juntos en una relación"<sup>11</sup>, dicho con otras palabras, un fact establece una asociación entre dos o más terms (sustantivos) y sus atributos.

De acuerdo a lo anterior se tienen los siguientes tipos de facts.

1. *Cada ocurrencia de una entidad debe tener la "propiedad de ...", es decir, que ésta debe ser descrita por un atributo o conjunto de atributos.*
2. *Cada ocurrencia de una entidad debe de ser "un tipo de ...", en otras palabras, un sub-tipo de otra entidad.*
3. *Cada ocurrencia de una entidad debe "jugar el rol de ...", es decir, debe estar relacionada de alguna manera con otros terms.*

Un fact involucra dos o más terms y un term puede participar en uno o más facts. Es necesario identificar el rol o papel semántico que los objetos juegan dentro del fact. Cada term puede tener uno o más roles de objeto, ya que éste convive con otras entidades del negocio, pero debe tener al menos un rol semántico que participe en el fact para poder describir la relación.

---

<sup>11</sup> David C. HAY, Data Modeling by the Rules, en DataBase Newsletter (June 1995).

La siguiente figura muestra la regla del negocio que establece una relación entre los terms cliente y cuenta.



De acuerdo a lo establecido deben existir dos roles semánticos, uno por cada objeto participante en la relación, es decir, en ambas direcciones.

1. *Cada cliente puede contratar varias cuentas bancarias*
2. *Cada cuenta bancaria pertenece sólo a un cliente*

### 3.4 Constraints

Son sentencias que se encuentran relacionadas con aspectos del negocio que constantemente están cambiando y están orientadas a restringir o condicionar los resultados arrojados por un proceso o función. Técnicamente, son las reglas de integridad que protegen los datos, ya que representan lo que debe ser cierto (veraz) para un term, fact o atributo.

De acuerdo con David C. Hay, un constraint es "Una regla que representa una restricción o validación necesaria para que la población de un tipo de relación sea permitida"<sup>12</sup>

<sup>12</sup> *Ob. cit.*

Un constraint sencillamente expresa las reglas para los datos que son permitidos en la organización, es decir, éstas imponen condiciones del negocio del tipo "*debe*" o "*no debe*". Por ejemplo:

1. *Un cliente bancario debe tener un número único que lo identifique.*
2. *Cada factura que se expida debe tener impreso el código y descripción de la moneda.*
3. *La biblioteca debe suspender el servicio de préstamo a domicilio al estudiante que entregue los libros después de la fecha de devolución.*
4. *Cada factura expedida por la venta de productos debe tener la descripción y código del artículo.*
5. *Un cliente bancario no debe ser dado de baja si tiene cuentas activas.*
6. *Un socio del video-club debe tener una de las siguientes categorías: Preferencial o Premier.*
7. *Cada lote de solicitudes de afore enviado por cada sucursal, debe tener el nombre, número de nómina del responsable, número de solicitudes, medio de envío e identificador de la sucursal.*
8. *Un cliente del banco debe tener una cuenta bancaria, pero no más de tres.*

Este tipo de reglas al igual que las anteriores se pueden representar y documentar mediante un modelo de datos utilizando la técnica entidad-relación, aunque existen constraints más complejos que no pueden ser expresados en su totalidad por el modelo de datos, pero más adelante expondré el tratamiento que debe darse a este tipo de constraints.

Cada constraint establece el efecto de uno o más objetos restringidores (constraining objects) sobre otro objeto que será el restringido (constrained object). Esto es aplicable para cualquier tipo de regla del negocio especificada en el modelo de datos, es decir, terms, facts, atributos, constraints, etc.

Para entender mejor el concepto de constraining objects y constrained objects a continuación se analizan los ejemplos mencionados.

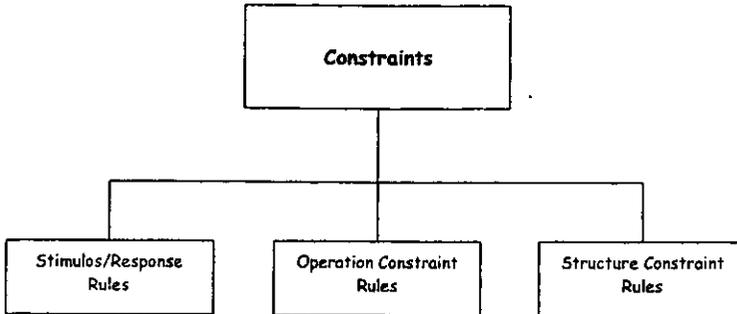
**Constraints**

- *La biblioteca debe suspender el servicio de préstamo a domicilio al estudiante que entregue los libros prestados después de la fecha de devolución.*
- *Un cliente debe tener una cuenta de cheques, pero no más de tres.*

Para el primer constraint, la regla del negocio estudiante que es un term es el constrained object y la segunda regla del negocio (restricción), que exige que el estudiante debe ser sancionado es el constraining object.

En el segundo ejemplo, la regla del negocio cliente es el constrained object y la regla del negocio que expresa que debe tener una cuenta de cheques, pero no más de tres es el constraining object, como se puede apreciar esta última regla es un fact.

Los constraints se pueden clasificar de la siguiente manera:



### 3.4.1 Stimulus/Response Rules

Las reglas que responden a estímulos están orientadas a controlar o restringir la forma de operar de la empresa (conducía) mediante el uso de condiciones del tipo CUANDO y SI, que deben ser verdaderas para que una operación o acción específica se ejecute. Ejemplo:

- CUANDO la factura por servicios profesionales de consultoría se cobre al cliente SI han transcurrido 30 días, ENTONCES pagar comisiones a los consultores participantes en el proyecto.
- CUANDO los 3 intentos permitidos al cliente bancario para digitar su PIN (Personal Id. Number) de acceso a cajeros automáticos ha sido excedido, ENTONCES bloquearle el acceso.
- CUANDO el stock de un producto en el almacén sea menor al límite mínimo, ENTONCES ordenar automáticamente un nuevo pedido vía fax.

Generalmente, este tipo de reglas influyen la manera de operar del negocio mediante la ocurrencia de eventos dentro de un contexto específico.

### 3.4.2 Operation Constraint Rules

Estas reglas especifican las condiciones que deben prevalecer antes y después de la ejecución de una operación para asegurar que ésta se ejecuta correctamente. A diferencia de las reglas que responden a estímulos, éstas son completamente independientes del evento contextual bajo el que la operación se lleva a cabo.

En virtud de que este tipo de reglas expresan cual debe ser la entrada y salida de un proceso específico, entonces toman la forma de pre-condiciones y post-condiciones como lo afirma Bertrand Meyer "Si un proceso es invocado de acuerdo con las entradas establecidas por la precondición, entonces éste garantiza una salida que cumple con lo establecido por la postcondición"<sup>13</sup>. Donde:

- **Precondición:** Es aquello que debe asumirse como verdadero inicialmente, es decir, expresan las reglas bajo las cuales una operación se ejecutará correctamente. Si éstas restricciones no se cubren, entonces la operación no debe ejecutarse.
- **PostCondición:** Especifica lo que debe ser verdadero después de la ejecución de una operación. La precondición garantiza que la postcondición se cumple.

El siguiente ejemplo supone una operación bancaria que cambia un cliente bancario de tipo preferencial a premier.

---

<sup>13</sup> MEYER Bertrand, Object – Oriented Software Construcción (New York, Prentice Hall, 1998).

**Precondición:**

- Ser un cliente preferencial
- Saldo promedio de 100,000 pesos

**Postcondición:**

- Un cliente premier
- Capacidad de crédito por un límite de 30,000 pesos
- Servicio Banco en su casa sin costo adicional

### **3.4.3 Structure Constraint Rules**

Especifican las políticas o condiciones que aplican a todas las reglas del negocio descritas por el modelo de datos, es decir, terms, facts, constraints, atributos, etc. Este tipo de reglas no están relacionadas con las operaciones o funciones, ya que, éstas deben cumplirse bajo cualquier circunstancia operativa.

Son las aseveraciones que siempre deben ser verdaderas, porque prohíben que una operación arroje resultados no veraces. Caso diferente al de una condición que evalúa un valor y dependiendo del resultado una acción u operación se lleva a cabo.

Como su nombre lo dice, estas reglas restringen la estructura, por lo que se tienen diversos tipos de ellas, dependiendo a que tipo de regla del negocio se apliquen. Por ejemplo se tienen:

a) *Reglas que restringen el valor de un atributo:*

- El número de cuenta de un cliente del banco debe ser de 11 dígitos
- El límite de crédito de un cliente premier no debe exceder a \$15,000
- El sueldo de un consultor Jr. no debe exceder al de un consultor Sr,

b) *Reglas que restringen la población de una entidad (term).*

- Un socio del videoclub debe ser mayor de 21 años
- Un cliente del banco debe tener al menos una cuenta de cheques

c) *Reglas que restringen la cardinalidad de una relación (fact)*

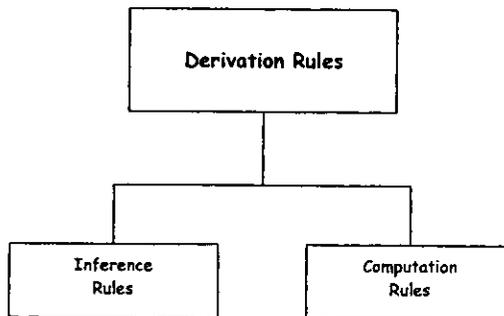
- Un socio del videoclub no puede rentar más de 5 películas

### 3.5 Derivation Rules

Este tipo de reglas se derivan u obtienen a partir de otras reglas y especifican las políticas o condiciones para inferir o computar nuevos facts a partir de otros facts, generando lo que se conoce como facts derivados (derived facts), que son simples aseveraciones que se construyen tomando como base otras afirmaciones.

Un fact derivado se obtiene utilizando una derivación que es un cálculo matemático o inferencia lógica, ésta por si misma, es una regla del negocio y debe estar basada en una o más reglas del negocio, tales como: terms, facts, otras derivaciones e incluso constraints.

Según el concepto de reglas derivadas se tiene la siguiente clasificación de las mismas:



### **3.5.1 Inference Rules**

Especifican que, si ciertos facts son verdaderos una conclusión debe ser inferida. Generalmente son derivaciones de carácter lógico que presentan la forma: SI ... ENTONCES. Por ejemplo:

SI      Un cliente paga su tarjeta de crédito 5 días antes del corte,  
ENTONCES      Se le debe aplicar un 20 % de descuento.

SI      Una persona trabaja en la empresa,  
ENTONCES      Es un empleado

SI      Un lote de solicitudes de afore tiene solicitudes incorrectas,  
ENTONCES      Generar un sublote de solicitudes rechazadas

SI      Un cliente tiene tres facturas vencidas,  
ENTONCES      Es un cliente moroso

Como estas reglas especifican conclusiones lógicas, generalmente generan relaciones entre entidades y entre los valores de los atributos de éstas.

### 3.5.2 Computation Rules

Estas reglas obtienen sus resultados mediante algoritmos y adquieren la forma de una ecuación, ejemplo:

- El precio final de un producto se calculará de la siguiente manera:

$$\text{Precio del Producto} = \text{Precio de Lista} + 15\% \text{ del Precio de Lista}$$

- La Prioridad de atención de una transacción bancaria se obtendrá:

$$\text{Prioridad Total} = \text{Prioridad Tipo Usuario} * \text{Prioridad Cliente} * \text{Prioridad Tipo Transacción}$$

(Afectación Base de Datos o Consulta BD)

- La edad de una persona no es dato requerido y deberá obtenerse de las siguiente manera.

$$\text{Edad Persona} = \text{Fecha Actual} - \text{Fecha Nacimiento}$$

- Las comisiones que se pagan a los consultores se calcularán:

$$\text{Comision Tot.} = 10\% \text{ Monto Total Factura (sin IVA)} * \text{Porcentaje según categoría consultor}$$

### 3.6 Reglas del Negocio más complejas

En el capítulo dos, mencioné que uno de los principales objetivos del enfoque de reglas del negocio es reducir la distancia entre la gente del negocio y los profesionales de la tecnología de la información en el desarrollo de sistemas, ésta brecha en gran parte ha sido generada, porque cada una de las partes tiene su propia perspectiva en la solución de problemas: *negocio vs. tecnología*, en realidad, no es que ninguna de ellas no muestre disposición, simplemente no hablan el mismo lenguaje lo que origina desconfianza y distanciamiento.

A lo anterior también debe agregarse lo siguiente:

- a) La incapacidad de las técnicas estándar de modelado de datos para expresar las reglas más complejas del negocio, ya que la gran mayoría de éstas, generalmente están ocultadas por el mismo modelo.
  
- b) Las reglas son simplemente ignoradas por el analista, quien delega la responsabilidad al desarrollador otorgándole plena libertad en la forma de implantarlas, esto desde luego implica un gran riesgo para la organización, ya que, estas reglas son verdaderas piezas de información que raramente son utilizadas únicamente por un sólo sistema y como tales se aplican a todos los sistemas que hacen uso de los datos de la empresa. Así que, si estas reglas no son capturadas o implantadas de manera consistente en otras bases de datos o sistemas, podrían amenazar la integridad de los datos.
  
- c) El descuido del analista al no cuestionar más profundamente a la gente de la organización (usuario) para obtener información más completa de la forma de operar del negocio.

Con el objeto de ilustrar lo anterior suponga el siguiente caso: En una reunión de requerimientos para construir un sistema de información y generar el modelo de datos de la aplicación, el usuario establece que un empleado debe tener un puesto, así que, de manera inmediata el analista modelaba esto mediante una relación entre las entidades EMPLEADO y PUESTO con una cardinalidad de muchos a uno sobre la entidad PUESTO. Sin embargo, otra persona del negocio asegura que no todos los empleados tienen un puesto solo los empleados activos, es decir, aquellos actualmente productivos para la empresa, por lo que los jubilados o pensionados deben ser excluidos. De esta manera la relación se vuelve opcional, así que el analista, inmediatamente agrega un atributo de status a la entidad EMPLEADO.

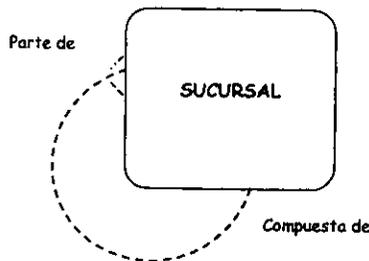
El modelo de datos anterior es correcto, pero éste oculta la regla del negocio, porque el analista no tiene manera de documentarla, así que la ignora esperando que ésta surja más tarde como un nuevo requerimiento. Esto como se aprecia, demanda la importancia de introducir el concepto de reglas del negocio entre la gente de la organización y definir juntos un lenguaje común de comunicación para expresar las reglas de operación del negocio a fin de materializar una liga entre la gente del negocio y la de sistemas.

En virtud de que uno de los factores que han contribuido a materializar la distancia entre el área de sistemas (tecnología) y administrativa de una organización es la ausencia de un lenguaje común para representar y documentar las reglas del negocio, sobre todo aquellas que debido a su complejidad son difíciles de modelar y presentar al usuario, a continuación, presento el análisis y tratamiento de algunas reglas del negocio complejas cuyo tratamiento contempla el uso del lenguaje que será empleado en el capítulo 4 para representar y documentar las reglas del negocio del caso práctico ahí expuesto.

### 3.6.1 Asumir Reglas del Negocio Implícitas en el Modelo de Datos

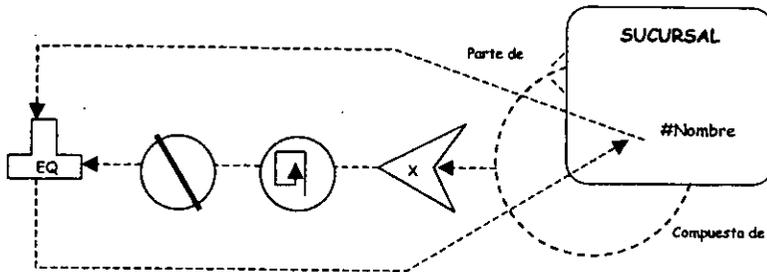
Lo que determina que este tipo de reglas sean consideradas complejas para ser expresadas por el modelo de datos, es que son tan obvias que se asume quedan representadas por éste, sin necesidad de hacerlo de manera explícita. Por ejemplo:

Analicemos el caso actual en México que obliga por disposición oficial a cada trabajador a estar afiliado a una AFORE (Institución responsable del manejo de fondos para el retiro del trabajador), por lo que las instituciones afiliatorias mediante una infraestructura de sucursales de afore desplegada a nivel nacional, recaba las solicitudes de afore de los trabajadores, para lo cual establece la siguiente política para el control de las sucursales: *"Cada sucursal de afore debe pertenecer a una sucursal de afore madre"*. La siguiente política se modela como se expresa en la siguiente figura.



El usuario al ver la figura puede determinar que ésta establece: Que cada sucursal puede estar compuesta de una o más sucursales y que cada sucursal puede ser parte de una y sólo una sucursal, sin embargo, el modelo no indica ninguna restricción para poder afirmar que una sucursal puede ser parte de sí misma o bien, en una relación más compleja puede asumir, que la sucursal del pedregal por ejemplo, puede ser parte de la sucursal Tlalpán, misma que a su vez, es parte de la sucursal del pedregal.

Para poder describir de manera más clara y adecuada la regla del negocio anterior, se propone modificar el modelo de datos mediante una extensión a éste basada en la metodología propuesta por Ronal G. Ross. La regla del negocio expresada completamente quedaría representada como lo muestra la figura 6.



El modelo de datos como se observa sigue expresando la relación recursiva sobre la entidad sucursal según la técnica entidad-relación, sin embargo, éste muestra una extensión representada por símbolos ajenos a ésta técnica. De acuerdo a este lenguaje propuesto, la regla : "Cada sucursal de afore debe pertenecer a una sucursal de afore madre", queda completamente modelada y documentada; expresada sin ambigüedades y expuesta libre de diversas interpretaciones.

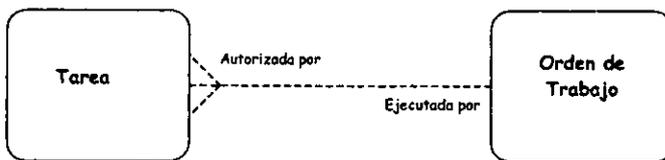
El modelo se lee de la siguiente manera: El símbolo con las letras EQ, es un evaluador comparativo que establece una regla de igualdad, es este caso, estableciendo que los nombres de dos ocurrencias de la entidad sucursal pueden ser iguales, sin embargo, para poder leer la relación en el sentido "una sucursal es parte de", es necesario el constraint mandatorio (flecha con la letra X), que requiere que la condición antes mencionada no evalué a verdadero, esto se consigue poniendo

el círculo con la diagonal que niega la igualdad, no permitiendo que la ocurrencia de la entidad se de en los casos que los nombres de las sucursales sean iguales.

La regla del tipo EQ (Igual) que es un evaluador comparativo, establece que los nombres de dos ocurrencias de la entidad sucursal podrían ser iguales, sin embargo, el constraint mandatorio (flecha con la letra X), requiere que la condición no debe ser verdadera, así que, el círculo con una diagonal niega la regla de EQ asegurando que la condición no sea verdadera. Finalmente, el círculo con la flecha en espiral indica que la regla aplica a las entidades recursivas.

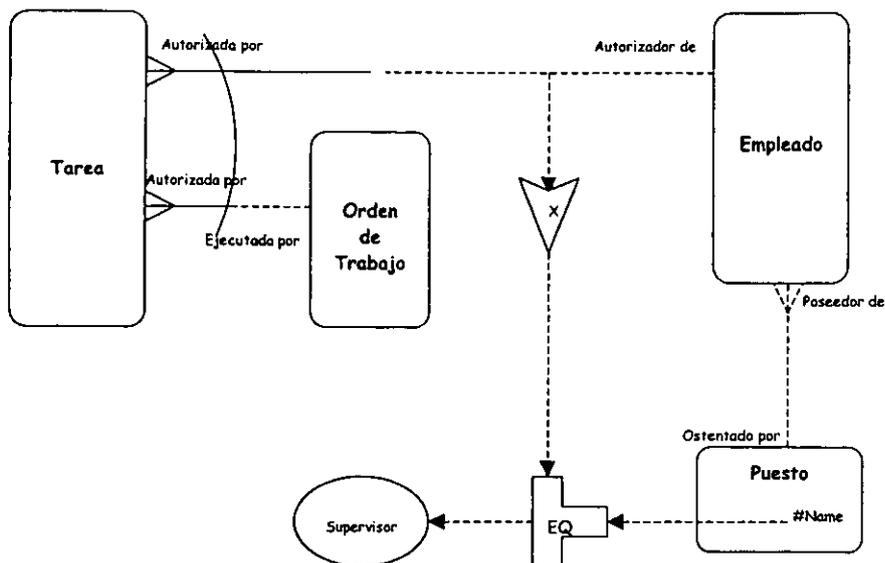
### 3.6.2 Relaciones Opcionales

Las relaciones opcionales indican que una ocurrencia de una entidad puede estar relacionada a una o más ocurrencias de otra entidad. La relación puede representarse en el modelo de datos, pero éste no indica cuando o bajo que circunstancias la relación puede establecerse, lo que la convierte en opcional. En la figura 7 se muestra un ejemplo de este tipo de relaciones



La relación expresada en la figura, establece que una ORDEN DE TRABAJO se cumple mediante una o más tareas, de manera recíproca, una actividad debe ser autorizada por una orden de trabajo, sin embargo, bajo ciertas circunstancias una actividad podría ser ejecutada sin que ésta este autorizada por una orden de trabajo, evidentemente esto no se encuentra representado por el modelo de datos.

Al igual que el problema de las relaciones implícitas, las relaciones opcionales pueden ser modeladas de manera clara y completa bajo el enfoque de reglas del negocio mediante una extensión al modelo que se muestra en la figura 8.



En el modelo anterior, queda estipulado que cualquier actividad puede estar autorizada por una orden de trabajo o por un empleado que necesariamente tenga el puesto de supervisor. La regla del negocio de acuerdo a la extensión hecha al modelo, establece lo siguiente: Si una ACTIVIDAD no es autorizada por una ORDEN DE TRABAJO, ésta debe ser autorizada por un EMPLEADO que debe tener el puesto de SUPERVISOR, ésta regla está dada por la relación (*autorizada por*), entre las entidades EMPLEADO y TAREA; y, para que pueda ser establecida debe pasar la restricción representada por el símbolo con la X en el centro, que requiere que la condición de igualdad indicada por el símbolo con las letras EQ evalúe a verdadero, es decir, que el EMPLEADO sea un SUPERVISOR.

Como se puede ver, uno de los valores agregados del enfoque de reglas del negocio al expresar de una manera clara y completa las reglas de operación de la organización, es su carácter contractual, entre el usuario y el analista, evitando situaciones desgastantes y mal entendidos que surgen más tarde cuando la etapa de desarrollo se encuentra muy avanzada debido a que ciertos requerimientos o reglas del negocio no fueron considerados o simplemente se pasaron por alto durante la etapa de análisis. De esta manera, el modelo que expresa completamente las reglas de gestión del negocio representa el contrato de los acuerdos establecidos entre el usuario (gente del negocio) y el analista (tecnología de la información) al momento de iniciar el desarrollo de sistemas.

Si el modelo expuesto anteriormente para capturar las reglas del negocio resulta muy complejo para la audiencia de negocios o incluso para la gente de sistemas, es importante no descuidar la práctica de documentar las reglas como simple texto, comentarios o, en el mejor de los casos a nivel de miniespecificaciones que toman la forma de pseudocódigo. Por ejemplo:

```
SI fecha_hoy > fecha_entrega ENTONCES
    estatus_estudiante SANCCIONADO
OTRO
    estatus_estudiante asigne NO_SANCCIONADO
```

Es importante además que el analista considere los siguientes elementos durante una junta de análisis de requerimiento con el objeto de obtener de la gente de la organización las reglas del negocio de una manera más efectiva.

- Tratar de ser más perceptivo para poder entender el contexto global en el que la aplicación se desempeña.

- Si encuentra reglas del negocio que no encajan completamente en el modelo de datos, no las ignore, documéntelas como texto, porque indudablemente surgirán más tarde como una petición de mantenimiento o bien durante la etapa de pruebas del usuario.
- No pierda de vista aquellas señales que sugieran reglas del negocio, como lo son los atributos, ya que por lo general, las reglas del negocio se ocultan detrás de éstos.
- Los atributos con un conjunto válido de valores, representan un fuente rica de reglas del negocio. El discutir por qué un atributo cambia de valor, casi siempre nos lleva a descubrir una multitud de reglas.
- Datos Derivados; analice como se obtienen los valores de los atributos de las entidades, porque es realmente sorprendente como los usuarios de los sistemas deben realizar cálculos para obtener el valor de un campo que el sistema requiere, así que, se puede obtener una gran variedad de reglas para datos derivados. Por ejemplo: Un estudiante de la biblioteca cuenta con un campo que indica la fecha de devolución del material en calidad de préstamo y otro que indique la fecha a partir de la cual el estudiante puede volver a tener préstamo a domicilio en los casos que es sancionado, por lo que, el valor del campo fecha sanción se podría automatizar (calcular) a partir del valor del campo fecha de devolución y de la fecha en que el estudiante realmente entrega el material.

- Atributos opcionales analice por qué un atributo es opcional y por qué otro es mandatorio. Los atributos opcionales podrían llegar a ser mandatorios bajo ciertas circunstancias, así que, al discutir dichas circunstancias se puede percatar que ciertas reglas del negocio no estaban siendo cubiertas. Por ejemplo, la fecha de retiro de un empleado normalmente es opcional, pero ésta se convierte en mandatoria cuando el status del empleado cambia a retirado.
- Relaciones opcionales; las relaciones muestran la asociación entre entidades, por lo que es importante entender por qué una relación existe, es decir, cuándo o bajo que circunstancias la relación se establece. Las relaciones opcionales pueden llegar a ser mandatorias bajo ciertas circunstancias.

## **MODELANDO REGLAS DEL NEGOCIO**

### **4.1 Introducción**

Con el objeto de aplicar el cambio propuesto por el enfoque de reglas del negocio en la construcción de sistemas de información es importante presentar un caso práctico que demuestre la utilidad de modelar formalmente las reglas del negocio durante la etapa de análisis, evidentemente antes de exponer dicho caso es necesario definir el lenguaje que utilizaré para modelar las reglas del negocio.

En este capítulo se define la sintaxis básica del lenguaje para modelar reglas del negocio, la cual incluye: los símbolos utilizados para representar las reglas en el modelo de datos, su semántica, las convenciones para su uso y el esquema básico para modelar una regla, es decir, los elementos necesarios. Una vez establecida la sintaxis par modelar las reglas se explican los tipos de reglas más comúnmente utilizados acompañados de un pequeño ejemplo en el que se ilustra la forma de modelarlas.

El lenguaje que se presenta se basa en la metodología propuesta por Ronal G. Ross para modelar y documentar reglas del negocio.

## **4.2 Lenguaje para modelar reglas del negocio**

Antes de presentar el caso práctico en el que expongo la manera de documentar y modelar formalmente las reglas del negocio, primero presentaré el lenguaje que utilizaré para tal propósito. Este lenguaje se basa en la metodología propuesta por Ronal G. Ross quien es uno de los más notables estudiosos del tema.

Como se estableció en el capítulo 2, una regla del negocio es una restricción o evaluación (test) que se lleva a cabo para mantener la integridad de los datos, es decir, controlar la actualización de los datos persistentes del negocio. Este control refleja los patrones que regulan la conducta o forma de operar del negocio, una regla del negocio, por lo tanto, involucra una implantación formal de algún requerimiento del usuario que generalmente se establece en un forma textual que se conoce como sentencia o declaración de regla del negocio. Cada una de estas sentencias expresan una política o práctica operacional que es llevada a cabo por la empresa, sin referirse de ninguna manera a una estrategia particular de implantación o tecnología específica.

Desafortunadamente las sentencias de reglas del negocio a menudo son ambiguas y no pueden ser implantadas directamente, por lo que el analista debe expresar dichas sentencias de reglas del negocio en reglas más precisas (atómicas). De esta manera, debe contar con un esquema de clasificación que le permita agruparlas de acuerdo al tipo de restricción o evaluación que la regla lleva a cabo. Así entonces, se presenta el siguiente esquema de clasificación de reglas del negocio atómicas.

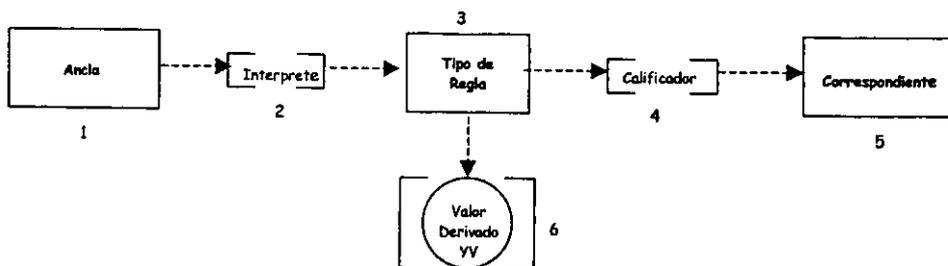
### 4.2.1 Tipos de Reglas del Negocio Atómicas

Los tipos de reglas se han clasificado en familias con el propósito de agruparlas de acuerdo al tipo de evaluación o restricción que éstas llevan a cabo. Cada uno de los tipos de reglas que las familias agrupan son atómicas, es decir, éstas no pueden derivarse u obtenerse a partir de combinaciones de otros tipos de reglas.

Verificadores de Instancias	Verificadores de Tipo	Verificadores de Posición	Verificadores Funcionales	Evaluadores Comparativos	Evaluadores Matemáticas	Controladores de Proyecciones.
X Mandatorio	M Mutuamente	POS Posición	FUNC Funcional	EQ Equal - to	CALC Calculado	EA Habilitado
LIM Limitado	ME Mutuamente Exclusivo	LOW Más Pequeño	UQ Único	NE No igual a	SUM Sumado	COP Copiado
	MI Mutuamente Inclusivo	HIGH Más Grande	FLUC Fluctuación	GT Mayor que	SUB Restado	EX Ejecutante
		CHRO Cronológico	AS Ascendente	GE Mayor que o igual que	MULT Multiplicado	
		OLD Más viejo	DE Descendiente	LT Menor que		
		NEW Más reciente	NRE No Reutilizable	LE Menor que o igual que		

### 4.2.2 Esquema sintáctico para modelar una regla del negocio.

Para poder modelar correctamente una regla del negocio y poder representar los tipos de reglas atómicas presentadas previamente, es preciso contar con un esquema de construcción para expresarlas. El esqueleto general para modelar una regla es el siguiente:



Los términos interprete calificador y valor derivado como se aprecia están contenidos entre corchetes lo que indica que estos son opcionales al momento de modelar la regla. La descripción de cada uno de los elementos involucrados en el diagrama, así como, las convenciones para su uso se explican en los párrafos siguientes.

Como la explicación de cada uno de los elementos requiere de un ejemplo ilustrativo se establece como parte del lenguaje para modelar reglas del negocio, que en lo sucesivo, cada regla deberá ser identificada de manera única mediante un número para poder referirse a ésta de manera clara y rápida.

#### 4.2.2.1 Restricciones de Integridad y Condiciones (Tipos de regla).

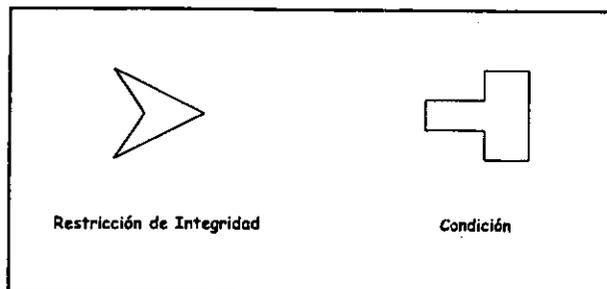
Como mencioné anteriormente, las reglas del negocio son restricciones o evaluaciones que se ejecutan con el objeto de implementar una política o práctica operacional del negocio, por lo que pueden ser vistas como expresiones lógicas que pueden ser verdaderas o falsas. De esta manera se tiene que al nivel más general de clasificación estas pueden ser de dos tipos:

- **Restricciones de Integridad**
- **Condiciones**

Una *restricción de integridad* es una regla que siempre debe cumplirse, es decir, siempre debe ser verdadera, ésta generalmente adquiere la forma de "DEBE".

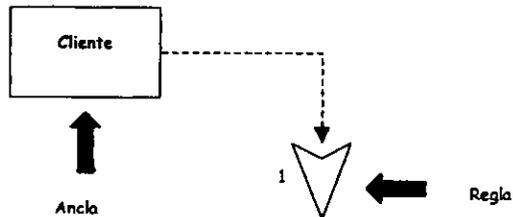
Una *condición* a diferencia de una restricción es una regla que puede ser verdadera o falsa. Su utilidad es brindar un mecanismo de prueba, para la ejecución de otras reglas, ésta generalmente toma la forma de "SI".

Para poder modelar este tipo de reglas es importante poder diferenciar una de la otra dentro del modelo, así que los símbolos que se emplearán para tal propósito son los siguientes:



#### 4.2.2.2 Ancias

Una regla del negocio siempre se refiere a instancias de algún tipo de dato representado por el modelo, este tipo de dato es llamado el ancla de la regla y puede ser un term, fact, atributo o incluso otra regla. El ancla es de primordial importancia, porque brinda un punto de referencia para la interpretación de la regla, como se muestra en la siguiente figura.



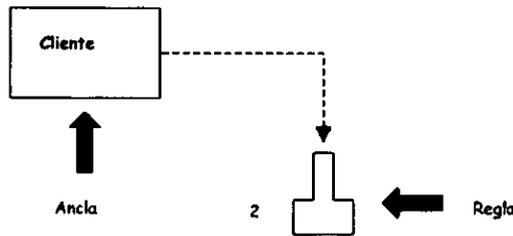
#### **Regla No.1 Un cliente debe tener (un valor en el atributo)**

Antes de interpretar lo que la regla de la figura anterior establece es importante entender la manera en la que la regla se liga con el ancla, ya que de esto depende la correcta interpretación de la regla. Por tal motivo se deben seguir las siguientes convenciones al momento de conectar el ancla con la regla.

- La línea de conexión entre el ancla y la regla siempre será una línea interrumpida, con el objeto de evitar confusiones con la técnica de modelado que se utilice.
- La conexión entre el ancla y la regla (línea interrumpida) puede salir de cualquier parte del perímetro del ancla.
- La conexión entre el ancla y la regla debe ligarse siempre en la parte posterior del símbolo de la restricción de integridad.

La regla No. 1 indica que la entidad cliente es su ancla y que su tipo es restricción de integridad. La interpretación, por lo tanto, establece que cada instancia de la entidad cliente *debe tener ... (valor de atributo)*. Como la especificación de la regla se encuentra parcialmente especificada no es posible concluir la interpretación de la regla. Para poder lograr esto es necesario establecer lo que se conoce como *correspondiente* de la regla, que se explica más adelante.

Las convenciones para conectar anclas y reglas también deben seguirse para modelar los tipos de reglas que son condiciones, con la excepción que la conexión entre el ancla y la regla debe ligarse siempre en la parte más delgada del símbolo de la condición, como se muestra en la siguiente figura:

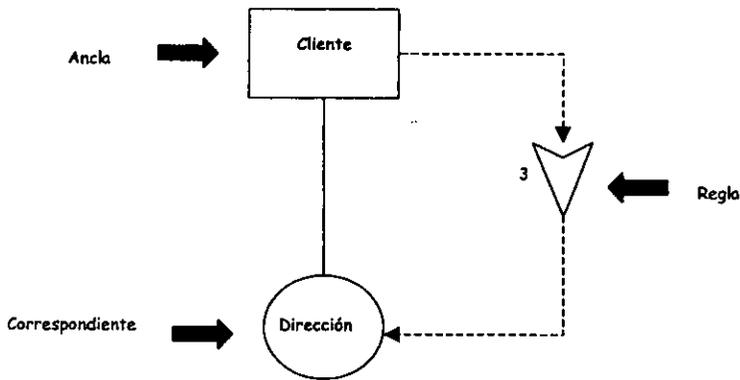


**Regla No. 2 Si un cliente tiene (valor en el atributo) entonces ...**

La evaluación que realiza la regla No. 2 se aplica a cada instancia de la entidad cliente y su interpretación dice que *Si* una instancia de la entidad cliente tiene ... *(valor de atributo)*, posiblemente otra regla sea ejecutada. A diferencia de la regla restricción de integridad, en este tipo de regla para cada instancia de la entidad cliente la regla puede arrojar un resultado verdadero o falso y no necesariamente verdadero como el caso de las reglas de restricción de integridad que siempre deben cumplirse.

#### 4.2.2.3 Correspondientes.

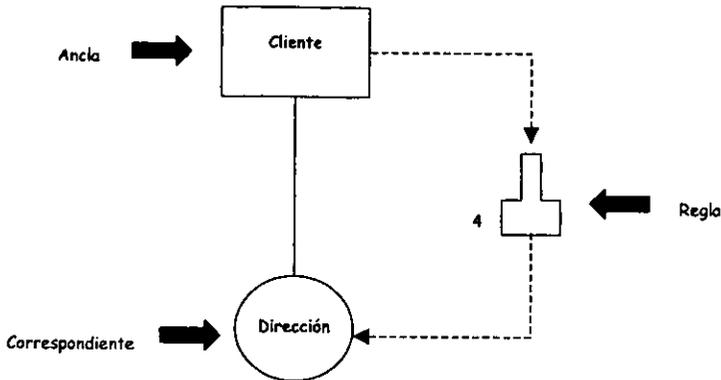
Como una regla hace referencia a las instancias de uno o más tipos de datos representados en el modelo, es necesario definir además del ancla de la regla su *correspondiente*, que no es más que un tipo de dato que al igual que el ancla puede ser una entidad, relación, atributo o incluso otra regla. Cada regla siempre debe indicar uno o más correspondientes. La figura siguiente muestra la especificación de una regla de tipo restricción de integridad involucrando al ancla y al correspondiente.



**Regla No.3 Un cliente debe tener una dirección**

La interpretación final de la regla No. 3 dice que cada instancia de la entidad cliente *debe* tener un *valor en su atributo de dirección*. De esta manera ninguna operación de actualización que intente violar la reglas será permitida.

En el caso de las reglas que son una condición, la especificación completa de la regla incluyendo al correspondiente quedaría como lo ilustra la siguiente figura:



**Regla No. 4** Si un cliente tiene una dirección entonces ...

La interpretación completa de la Regla No. 4 establece que para una instancia de la entidad cliente, la regla evalúa o prueba si ésta tiene un valor en su atributo dirección, de resultar cierto, entonces la regla genera un resultado verdadero o en otro caso falso. Ambos resultados pueden ser utilizados para disparar la ejecución de otra regla.

### Interpretes y Calificadores

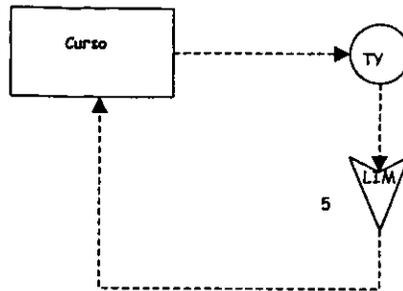
Algunas veces, la especificación de una regla del negocio, requiere de más precisión y el sólo uso de las anclas y los correspondientes no es suficiente para expresar completamente lo que la regla debe ejecutar, por tal motivo, es necesaria la utilización de ciertos modificadores que son llamados interpretes y calificadores de la regla. Estos modificadores según se ilustró en el esquema general de una regla son opcionales y sólo en los casos en que se necesite una mayor claridad en la especificación de la regla deben ser utilizados. Su uso se detalla a continuación:

4.2.2.4 Interpretes

Los interpretes son utilizados en aquellos casos en los que la especificación de la regla con respecto a las instancias del ancla requiere de una mayor claridad o de una interpretación especial. El cuadro siguiente muestra los tipos de interpretes permitidos para referirse a las instancias del ancla de la regla.

Símbolo	Nombre	Significado
	Negación	Niego el resultado de la regla
	Convertidor	Cambiar la interpretación de la regla
  	Seleccionador Aplicado al ancla  Aplicado al correspondiente  Aplicado al ancla y al correspondiente	Default  Se aplica a las instancias del ancla de la regla, es decir, como un criterio de selección.
 	Promotor Inmediato  Intermedio	Evalúa la regla tan pronto como una actualización ocurre. No se espera a que el proceso termine.  Prueba la regla hasta que la aplicación se lo hace saber de manera explícita.
  	Ejecutor Concurrencia  Sugerente  Retener	Permite a una restricción de integridad ser habilitada aun cuando esta ha sido violada, pero no permite los casos siguientes de violación.  No ejecuta la regla, pero indica al usuario que una violación ha ocurrido.  Nunca dispara la regla, pero retiene a esta para una futura referencia.

El siguiente ejemplo muestra cuando un interprete debe ser usado para dar mayor claridad a la especificación de la regla y evitar ambigüedades.



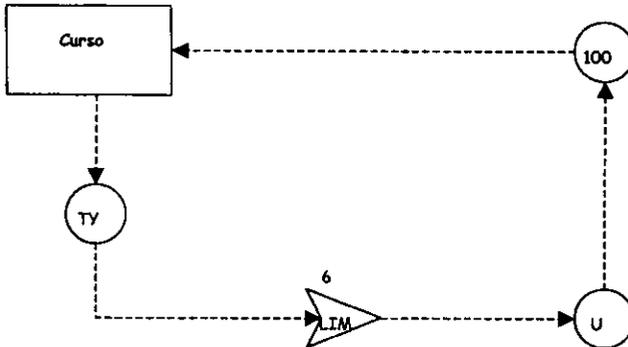
**Regla No. 5 dice que No más de 100 cursos pueden ser ofrecidos en cualquier momento.**

Sabemos que una regla se refiere a instancias de un tipo de dato (ancla, correspondiente o relación), por lo tanto, se asume que la regla se aplica a una sola instancia de la entidad curso, es decir, no es posible dar más de 100 cursos de un mismo tipo, evidentemente la regla que se desea especificar, entonces no tiene ninguna validez, ya que el verdadero sentido de ésta se refiere a que no más de 100 tipos de cursos pueden ser impartidos al mismo tiempo.

Para evitar ambigüedades se agrega un interprete entre el ancla y la regla conocido como convertidor, en el ejemplo anterior, éste se encuentra representado por el círculo con las letras TY. Este calificador indica que todas las instancias del ancla serán del tipo del correspondiente, es decir, una especie de meta-instancia general de un solo tipo que será llamada curso. De esta manera todas las instancias del ancla se agrupan bajo éste nuevo tipo en lugar de referirse a instancias individuales del ancla. El significado del correspondiente no cambia y éste sigue refiriéndose a todas las instancias del ancla, es decir, todas las instancias del nuevo tipo llamado curso. De acuerdo con la regla ahora si se puede establecer que no más de 100 instancias del tipo curso serán permitidas.

#### 4.2.2.5 Calificadores

Los calificadores son empleados cuando es necesaria una especificación más clara respecto a las instancias del correspondiente de la regla. Para ejemplificar lo dicho anteriormente, muestro la siguiente figura que especifica formalmente la regla No. 3 que se utilizó para ilustra el uso de los interpretes.



**Regla No. 6 dice que No más de 100 cursos pueden ser ofrecidos en cualquier momento.**

Como se aprecia en la figura la especificación de la regla No. 6 incluye el uso de interpretes y calificadores. La regla establece que no más de 100 tipos de cursos pueden ser impartidos en cualquier momento, en la especificación anterior de la regla (*Ver. Interpretes*) este límite no se encuentra representado gráficamente, pero ahora con el uso de calificadores el límite permisible se indica mediante un tipo de calificador llamado enumerador que ha sido incluido entre el ancla de la regla y su correspondiente. Este calificador se encuentra representado por el círculo con la letra U mayúscula que significa *más elevado que*, dado que el calificador se refiere a una enumeración, es preciso indicar el límite de ésta, lo cual, se representa por el círculo con el número 100.

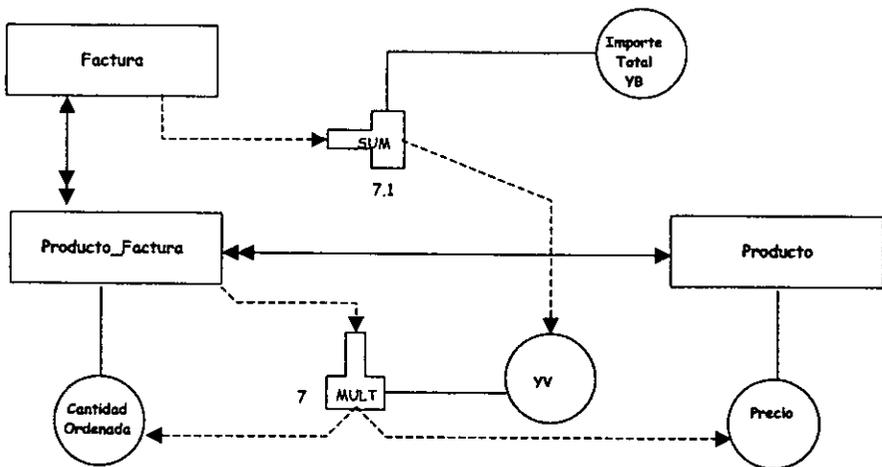
El cuadro siguiente muestra los tipos de calificadores que serán utilizados.

Símbolo	Nombre	Significado
	Cualquier	Rompe la conexión entre el ancla de la regla y el correspondiente.
	Todo	Se refiere a todas las instancias del correspondiente para una sola instancia del ancla
	Recursivo	Se refiere a instancias del correspondiente a través de uno o más niveles de recursividad. Entidades recursivas.
 	Enumerador	Se refiere a un número limitado de instancias del correspondiente donde: m = Tipo de enumeración n = Rango o límite de la enumeración m puede tener cualquiera de los siguientes valores: L : Más bajo que U : Más alto que F : Fijo
	Identificador Lógico	Establece una secuencia, orden o rango dado por el valor de p.

Con este tema se concluye la definición del lenguaje que utilizaré para modelar reglas del negocio, pero antes de continuar, es preciso explicar las características de cada uno de los tipos de reglas atómicas, así como, las convenciones para su uso. En cada uno de los tipos de reglas que se explican a continuación se modela una regla del negocio que utiliza el correspondiente tipo de regla, esto con el propósito de mostrar la aplicación del lenguaje establecido.

#### 4.2.2.6 Valor Derivado

Cada regla realiza una evaluación de acuerdo a su tipo y ésta es aplicada a cada instancia del ancla y del correspondiente. Para cada instancia del ancla la regla siempre genera un valor que se conoce como valor derivado (YV), éste generalmente no se representa gráficamente cuando la regla se modela, porque es usado internamente por ésta para llevar a cabo una evaluación, pero en determinadas ocasiones, las reglas requieren ejecutar una evaluación sobre el valor derivado de otra regla, por lo que es necesario exteriorizarlo, es decir, representarlo gráficamente en la especificación de la regla. Esto se logra dibujando un círculo con la etiqueta YV, como se muestra en el siguiente ejemplo.



**Regla 7** El importe total de una factura debe ser calculado como la suma de cada uno de los totales parciales obtenidos de multiplicar la cantidad ordenada de cada producto por su precio.

La regla 7 es una condición de tipo multiplicación, que multiplica el valor del atributo precio de un producto por su cantidad ordenada en la factura y pone el resultado en el *valor derivado* (YV) de la regla que es el correspondiente de la regla 7.1 (una condición de tipo suma) que se encarga de sumar todos los resultados de la regla 7 (instancias de su valor derivado) para obtener el valor derivado, *importe total* para una instancia de la entidad factura

Es importante considerar las siguientes convenciones en el uso de valores derivados:

- Una regla genera un solo valor derivado.
- El valor derivado de cada regla no puede ser modificado por el usuario
- No es necesario darle un nombre al valor derivado, pero se recomienda su uso con fines de claridad.
- Un valor derivado es visto y tratado como si fuera un atributo del ancla de la regla.

#### 4.2.3 Verificadores de Instancias.

Este tipo de regla se refiere a la posesión de instancias del correspondiente de la regla, es decir, para una instancia del ancla este tipo de regla prueba la posesión de una o más instancias del correspondiente. Este tipo de reglas generalmente indican que una instancia del ancla puede existir únicamente si la prueba de posesión indicada por la regla se cumple. Los tipos de regla de ésta familia son:

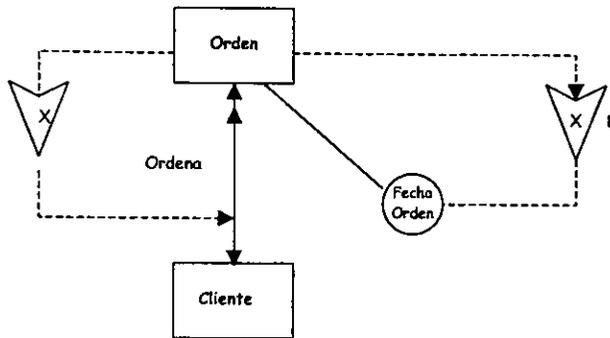
- **Mandatorio:** Realiza una prueba por la posesión de al menos una instancia del correspondiente
- **Limitado:** Prueba por la posesión de un número específico de instancias del correspondiente.

Los verificadores de instancias observan las siguientes convenciones:

- *Este tipo de reglas siempre se refieren a instancias del correspondiente.*
- *Nunca pueden actuar como conectores lógicos AND y OR. Por lo tanto nunca pueden tener más de un correspondiente.*

#### 4.2.3.1. Tipo de regla mandatoria.

Este tipo de regla se representa por la letra X y prueba que dada una instancia del ancla la regla se cumple sólo si una instancia del correspondiente existe. Ejemplo:

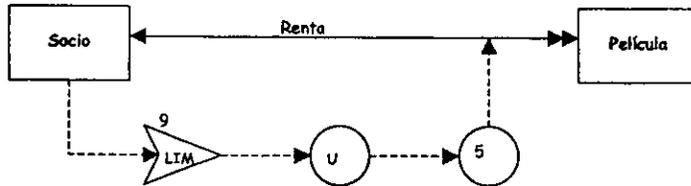


**Regla No.8 Una orden de compra debe indicar la fecha en que se levanta la orden y el cliente que la genera.**

La regla expresa que cada instancia de la entidad orden requiere que ésta posea una instancia de su correspondiente, en este caso, la relación ordena. La regla también requiere que para cada instancia de orden debe existir una fecha de orden, como se estableció anteriormente este tipo de reglas no pueden tener más de un correspondiente, es por eso que se utilizaron dos tipos de reglas mandatorias para expresar la regla completamente.

#### 4.2.3.2. Tipo de regla limitada.

Este tipo de reglas se ejecutan para probar si un número de instancias del correspondiente existen, el número de instancias que la regla prueba se especifica mediante una enumeración que consta de un tipo y un límite. Los tipos de enumeración generalmente son Mayor que, Menor que o Fijo, el límite se especifica mediante un número. Ejemplo:



**Regla No. 9 Un socio del video-centro no puede rentar más de 5 películas**

Para poder dar mayor claridad a la especificación de la regla fue necesario el uso de calificadores entre el correspondiente de la regla y el ancla, esto con el objeto de poder precisar el número de instancias del correspondiente. La regla No. 9 expresa que un socio no puede rentar más de 5 películas.

#### 4.2.4 Verificadores de Tipo

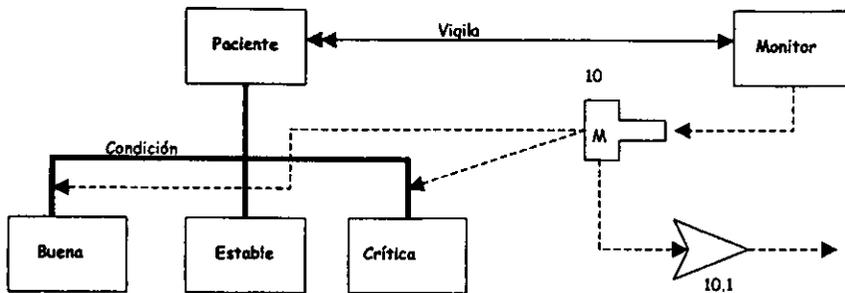
Este tipo de reglas actúan como conectores lógicos AND y OR, por lo que cada regla que se represente mediante este tipo debe contar al menos con dos correspondientes. Es importante mencionar, que cuando un correspondiente es verdadero para una instancia del ancla, se tienen los siguientes casos:

- Si el correspondiente es una entidad, relación u atributo entonces al menos una instancia existe.
- Si el correspondiente es otra regla, entonces significa que al menos una instancia resulta verdadera.

A diferencia de la familia Verificadores de instancias, este tipo de reglas tienen que ver más con el tipo del correspondiente que con el número de instancias de ese tipo de dato.

#### 4.2.4.1 Tipo mutuamente

Este tipo de regla realiza una prueba para determinar si un número dado de correspondientes son verdaderos de manera simultánea. Ejemplo:

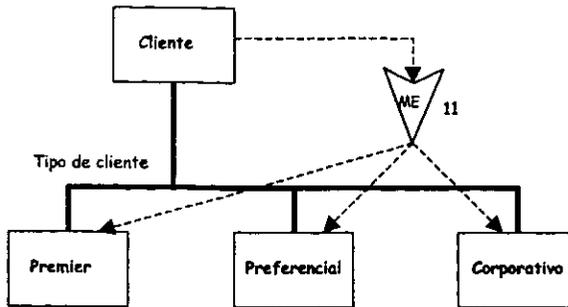


**Regla No. 9** Si un monitor vigila a los pacientes con una situación buena y crítica entonces ...

La regla No. 10 como se observa cuenta con dos correspondientes, así, para una instancia de la entidad monitor, la regla prueba si existen instancias de ambos correspondientes. Si la regla evalúa a verdadero, es decir, que al menos una instancia de cada correspondiente existe, entonces la regla No. 10.1 se dispara.

#### 4.2.4.2 Tipo mutuamente exclusivo

Las reglas de este tipo realizan una prueba para determinar si uno de varios correspondientes es verdadero, es decir, dada una instancia del ancla, la regla prueba si dicha instancia posee a lo más un correspondiente verdadero. La figura siguiente muestra el uso de este tipo de reglas:

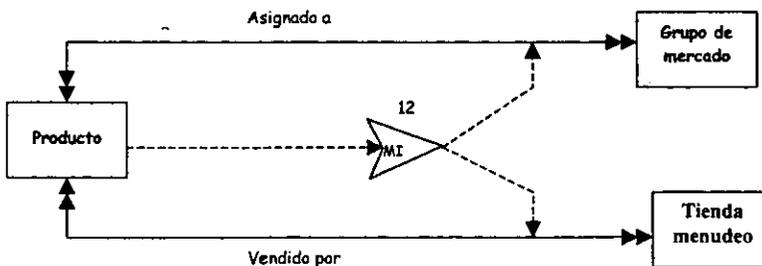


**Regla No. 11** Un cliente del banco debe ser *premier, preferencial o corporativo*, pero nunca más de uno de éstos tipos

La regla 11 requiere que cada instancia de cliente debe estar asociada en cualquier momento con solo uno de los tres tipos de clientes.

#### 4.2.4.3 Tipo mutuamente inclusivo

Este tipo prescribe la realización de una prueba para determinar si uno o más correspondientes de la reglas son verdaderos. Dada una instancia del ancla, la regla evalúa si al menos un correspondiente de la regla es verdadero. El siguiente ejemplo muestra el uso de este tipo de reglas.



**Regla No. 12** Un producto debe ser asignado a un grupo de mercado para su venta, pero también puede ser vendido por una tienda de menudeo, por lo que ambos lo pueden vender.

La regla No. 12 es una restricción de integridad del tipo mutuamente inclusivo que tiene como correspondientes a las relaciones *asignado a* y *vendido por*. Para uno u otro correspondiente, una instancia de la entidad producto existe solo si al menos una instancia de ambas relaciones se da.

#### 4.2.5 Verificadores de Posición

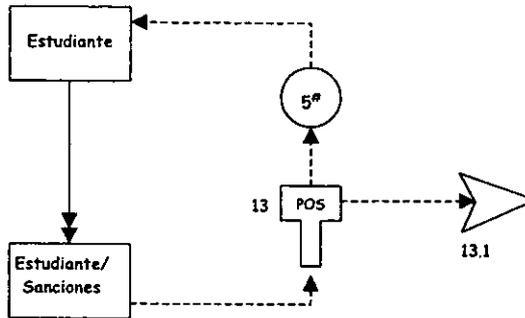
Los tipos de reglas que pertenecen a esta familia se refieren a una secuencia que se encuentra lógicamente implícita en la instancia del ancla o del correspondiente, esta secuencia puede estar basada en la edad de la instancia y es llamada cronológica, si la secuencia se basa en el valor de la instancia entonces recibe el nombre de secuencia de valor.

Las reglas de este tipo no pueden ser utilizadas como conectores lógicos (AND, OR), por lo tanto, nunca pueden tener más de un correspondiente. Esta familia de reglas cuenta con los siguientes tipos de reglas:

##### 4.2.5.1 Más Pequeño, Más grande y Posición

Este tipo de reglas indican una secuencia basada en un valor y son utilizadas para establecer un rango o posición dentro de una secuencia. El rango se fija para las instancias del ancla de la regla y el correspondiente de la regla indica como se forma el valor de la secuencia. Esta categoría de reglas incluye los siguientes tipos:

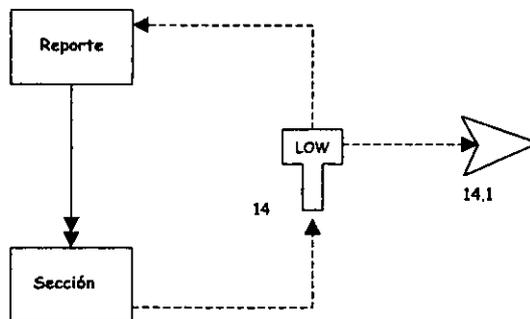
**Posición (POS)** es utilizado para establecer un rango arbitrario. Por ejemplo: el segundo, el cuarto, del tercero al último. La siguiente figura ilustra este caso:



**Regla No. 13** Si esta es la 5ª sanción que comete un estudiante de la biblioteca, entonces ...

La regla 13 hace referencia a una posición específica dentro de una secuencia, pero no hace referencia ni a la más baja ni a la más alta, esta regla sólo disparará la regla 13.1 si el valor derivado de la regla 13 es la quinta posición.

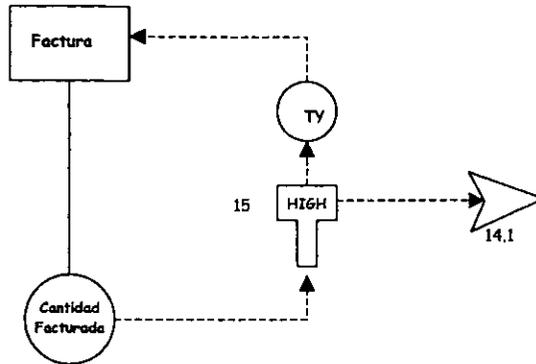
**Más pequeño (LOW):** es usado para seleccionar la posición más baja. Por ejemplo.



**Regla No. 14** Si esta sección es la primera de un reporte, entonces

En este ejemplo la regla 14.1 sólo se ejecutará si la sección de un reporte es la primera.

**Más grande (HIGH):** es empleado para referirse a la posición más grande. Por ejemplo:



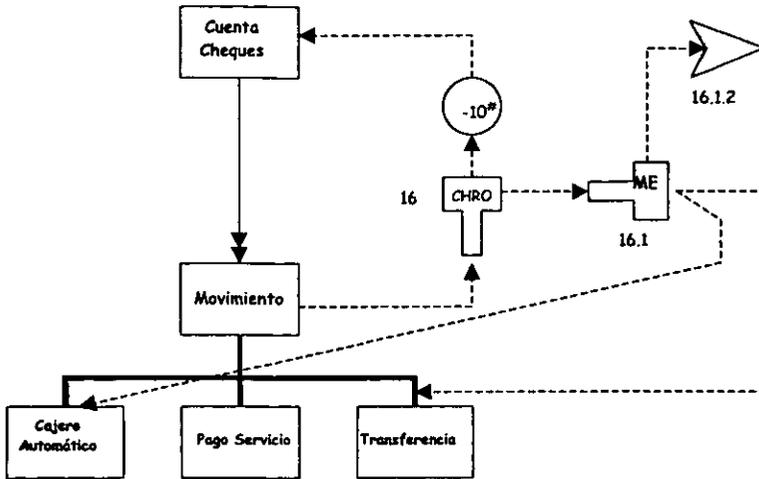
**Regla No. 15** Si esta es la orden con la mayor cantidad facturada entonces ....

En la regla No. 15 para una instancia del correspondiente (cantidad facturada), la regla 15.1 será disparada si el valor de la instancia esta dentro del rango más alto.

#### 4.2.5.2 Tipos: cronológico, más viejo y más reciente.

Este tipo de reglas se refieren a secuencias cronológicas, es decir, basadas en el tiempo. Los tipos básicos son los siguientes:

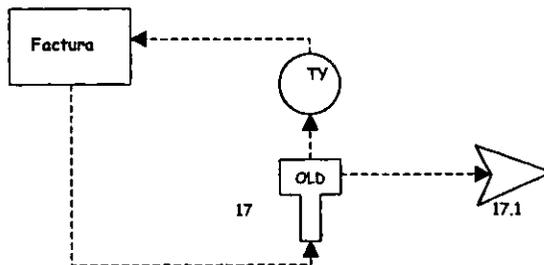
**Cronológico (CHRO) :** Es usado para establecer un rango arbitrario (primero, segundo, etc.). Por ejemplo:



**Regla No. 16** Si estos son los 10 más recientes movimientos de una cuenta de cheques y son transferencias o uso de cajeros automático entonces ...

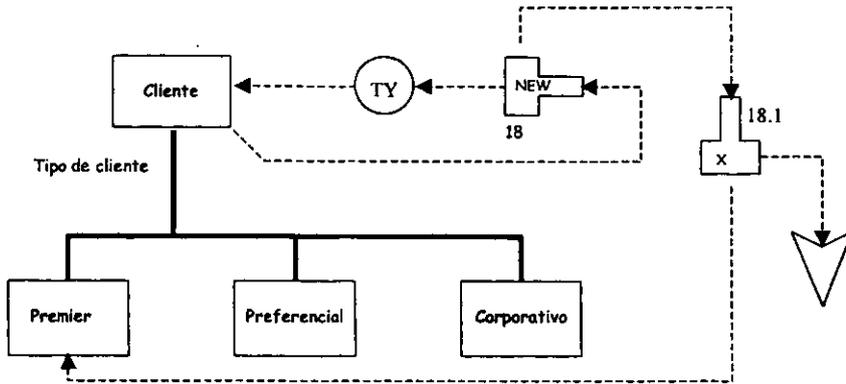
La regla 16 hace referencia a una posición específica dentro de un rango de una secuencia cronológica, pero no hace referencia a las instancias más antiguas o a las más nuevas. La regla por lo tanto, evalúa a verdadero para las 20 más viejas instancias de movimientos de una cuenta de cheques y dispara la regla 16.1 que verifica que éstas sean del tipo cajero automático o transferencia.

**Más Viejo (OLD)** : Es utilizado para hacer referencia al elemento más antiguo. Por ejemplo:



**Regla 17** Si éste es nuestro cliente más antiguo entonces ....

**Más reciente (NEW) :** Este tipo de reglas se emplean para referirse al elemento más nuevo



**Regla No. 18** Si el cliente más reciente del banco esta clasificado como premier entonces ...

La regla18 hace referencia a las instancias más recientes y evalúa a verdadero para el cliente más nuevo del banco disparando así la regla 18.1 que verifica que sea un cliente premier.

#### 4.2.6 Verificadores Funcionales.

Los verificadores funcionales prueban los valores del correspondiente en función de las instancias del ancla, es decir, dependiendo de la posición de la instancia del ancla el tipo de regla indica lo que el valor del correspondiente debe satisfacer. Todos los tipos de regla de esta familia se expresan de la siguiente manera  $y = f(x)$  donde:

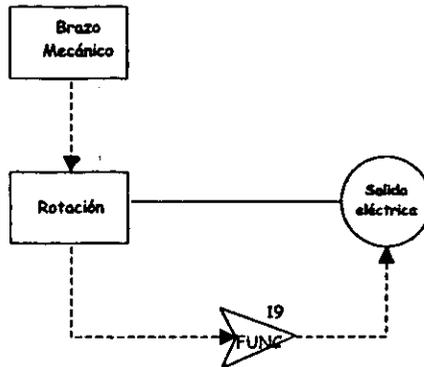
x: Los valores de x están dados por la posición relativa de cada instancia del ancla.

f: Es el tipo de regla y esta indica los patrones que los valores del correspondiente deben satisfacer

y: Los valores de y están dados por los valores del correspondiente de la regla. Estos son los valores que cumplen con lo especificado por el tipo de regla.

#### 4.2.6.1 Funcional (FUNC)

El valor de cualquier instancia del correspondiente se genera en función de la posición relativa de la instancia del ancla. Por ejemplo:

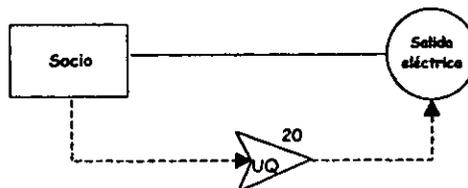


**Regla No. 19** La salida eléctrica de un brazo mecánico debe ser la función seno de su grado de rotación calculada cada 5 grados.

El alcance de la regla se limita a aquellas instancias del ancla de 5 grados, por lo que el valor del correspondiente (salida eléctrica), se debe computar como la función seno de la posición actual (grados) de la instancia del ancla.

#### 4.2.6.2 Único (UQ) :

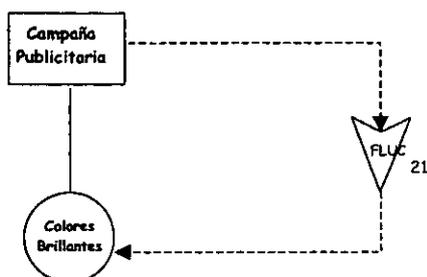
Este tipo de regla prescribe la existencia de unicidad de valores. No permite que dos instancias del ancla posean el mismo valor para una instancia del correspondiente. Por ejemplo:



**Regla No. 20** Los identificadores de socios del video-centro deben ser únicos.

#### 4.2.6.3 Fluctuación (FLUC)

Esta regla señala una variación de valores del correspondiente para aquellas instancias sucesivas del ancla. A diferencia del tipo de regla UQ que requiere que todos los valores del correspondiente sean diferentes, el tipo FLUC exige la unicidad solo para aquellas instancias sucesivas. De esta manera este tipo de reglas permite el uso de un valor anterior. Ejemplo.

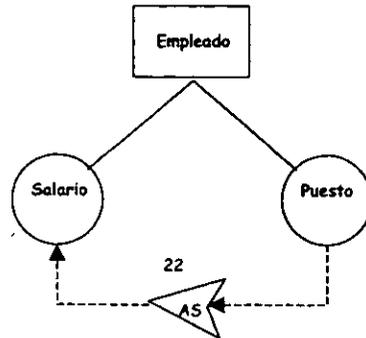


**Regla 21** Los colores brillantes que sean utilizados en campañas publicitarias sucesivas deben ser diferentes.

En la regla 21 a las instancias sucesivas del ancla de la regla, es decir, las campañas publicitarias no se les permite que utilicen los mismos colores brillantes.

#### 4.2.6.4 Ascendentes (AS)

Este tipo de reglas prueba que los valores de las instancias del correspondiente de la regla deben incrementarse para todas las instancias sucesivas del ancla. Estos valores además de seguir estrictamente un orden deben ser únicos. Por ejemplo:

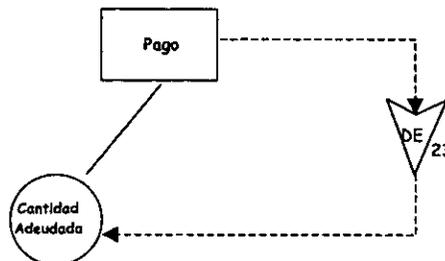


**Regla 22** El salario de un empleado deberá estrictamente incrementarse de acuerdo al puesto que éste tenga en la compañía.

En la regla 22 cada instancia del ancla (puesto) sigue un orden de ascendencia dictado por el tipo de puesto de tal manera que el correspondiente del ancla (salario) debe incrementarse en la misma manera.

#### 4.2.6.5 Descendentes (DE)

Prescriben que los valores de las instancias del correspondiente deben disminuir para cada instancia del ancla. Por ejemplo:

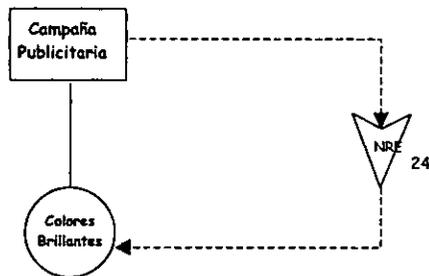


**Regla 23** La cantidad adeudada debe ser menor cada vez que un pago es efectuado.

La regla 23 indica que para todas las instancias del ancia (pago) los valores siguientes del atributo cantidad adeudada siempre deben ser menores al anterior, es decir, no permite que existan dos valores iguales.

#### 4.2.6.6 No reutilizables (NRE)

Establece que los valores del correspondiente sólo pueden repetirse en instancias sucesivas de éste, pero si este valor es interrumpido en las instancias siguientes a su uso, nunca podrá volver a ser utilizado. Por ejemplo:



**Regla 24** Una vez que un color brillante ha sido utilizado por campañas publicitarias contiguas, éste no podrá ser usado otra vez.

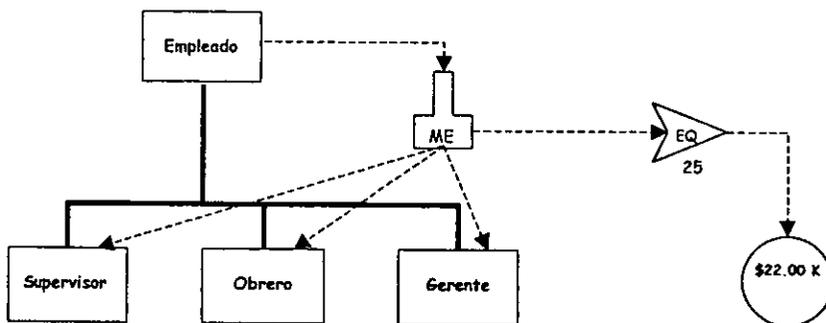
La regla 24 especifica claramente que un color brillante sólo podrá aparecer en una sola sucesión de campañas publicitarias. Una vez que ésta sucesión se interrumpe, es decir, otro color es utilizado, entonces el reutilizar el color anterior no es permitido.

#### 4.2.7 Evaluadores comparativos.

Los tipos de reglas de esta familia establecen una comparación entre dos o más instancias de un tipo de dato , éste generalmente es el valor de un atributo. Su uso se apega a las siguientes convenciones.

- *Nunca comparan valores de instancias sucesivas del ancla.*
- *No pueden actuar como operadores lógicos (AND, OR), por lo tanto, no deben tener más de un correspondiente.*

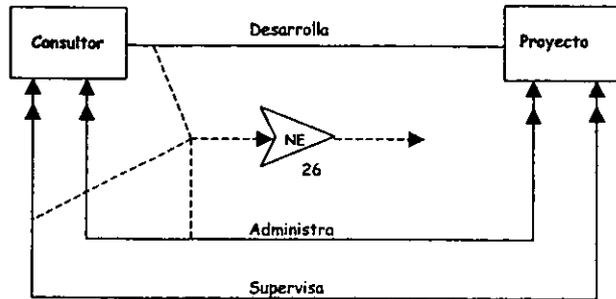
##### 4.2.7.1 Igual a (EQ)



**Regla 25** Si un empleado de la compañía es obrero, su sueldo debe ser igual a \$22.00

En la regla 25 se utiliza una constante como el correspondiente de la regla para expresar el valor del sueldo, esta constante se representa de la misma manera que un atributo de una entidad y puede ser usada como ancla o correspondiente de una regla.

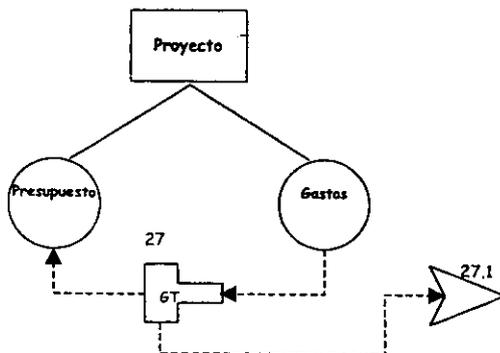
## 4.2.7.2 No igual a (NE)



**Regla No. 26** Un consultor no debe desarrollar código en un proyecto que el administre o supervise; tampoco debe administrar un proyecto en el que el desarrolle o supervise; etc.

La regla 26 es una restricción de integridad de tipo, *no igual a*, ésta tiene tres anclas que son las relaciones: *Administra*, *Supervisa* y *Desarrolla*. La regla 26 realiza una evaluación sobre las instancias de las relaciones anteriores para determinar cual de las tres instancias es la que no se repite, y poder establecer ésta como la relación permitida.

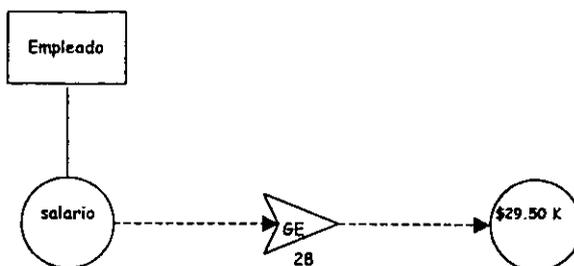
### 4.2.7.3 Mayor que (GT)



**Regla 27** Si los gastos de un proyecto son mayores que la cantidad presupuestada entonces .

La regla 27 es una condición de tipo mayor que, ésta compara el valor de los gastos del proyecto con la cantidad asignada de presupuesto y si arroja un resultado verdadero la regla 27.1 es ejecutada.

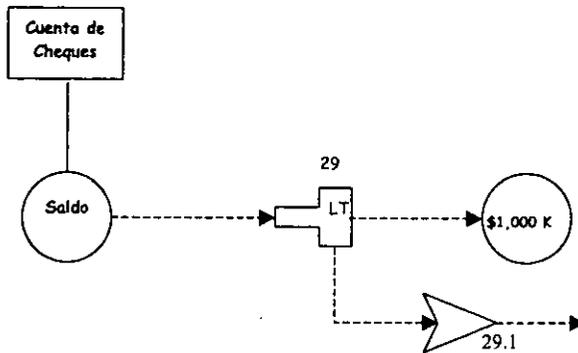
### 4.2.7.4 Mayor que o igual que (GE)



**Regla 28** El salario de un empleado debe ser mayor o igual que \$29.50.

La regla 26 es una restricción de integridad del tipo mayor que o igual que, la cual requiere que el valor del atributo salario para una instancia de la entidad empleado debe ser mayor que o igual que la constante 29.50.

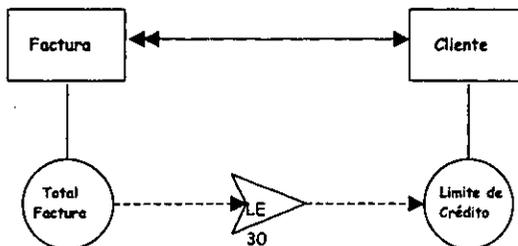
#### 4.2.7.5 Menor que (LT)



**Regla 29** Si el saldo de una cuenta de cheques llega a ser menor que \$ 1000 entonces ...

Esta regla, es una condición del tipo menor que, responsable de comparar el valor del saldo de una cuenta de cheques de una instancia dada contra la constante \$ 1,000 K. Si el resultado es verdadero, entonces la regla 29.1 se dispara.

#### 4.2.7.6 Menor que o igual que (LE)



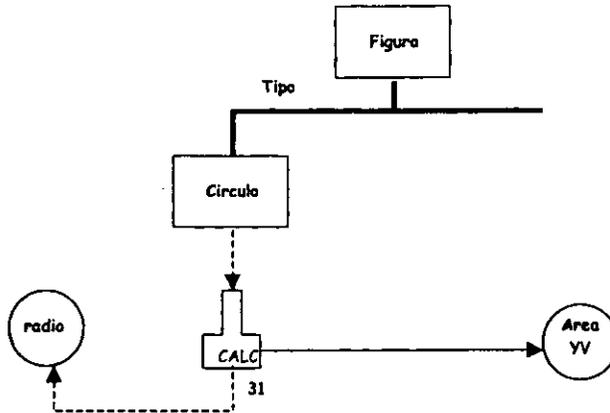
**Regla 30** El total de la factura de compra de un cliente debe ser menor o igual que el limite de crédito del cliente.

#### 4.2.8. Operadores matemáticos

Tienen como tarea evaluar el resultado generado por cálculo matemático que involucra a los valores del correspondiente. Cualquier cálculo matemático puede ser expresado mediante este tipo de reglas, ya que, si dicho cálculo no se encuentra dentro de los tipos predeterminados, se puede utilizar el tipo de regla, calculado (CALC), que es un tipo genérico de esta familia de reglas. Evidentemente el resultado generado se debe obtener mediante la fórmula de cómputo que el analista debe especificar. Las reglas que pertenecen a esta familia deben expresarse de la siguiente manera:

- El ancla de la regla es un objeto de datos, es decir, una entidad.
- El resultado del cálculo debe indicarse utilizando el tipo de regla apropiado.
- El resultado generado por la regla se conoce como **valor derivado de la regla** y debe expresarse gráficamente mediante un círculo con una descripción semántica del resultado.
- Este tipo de reglas pueden ser restricciones de integridad o condiciones.
- El valor derivado de la reglas es el resultado actual de la operación matemática que se efectuó para una instancia determinada del ancla.
- El resultado del cálculo que indica la regla, siempre se basa en el valor(es) del correspondiente(s) de la regla.
- El correspondiente de estas reglas generalmente son valores de atributos.

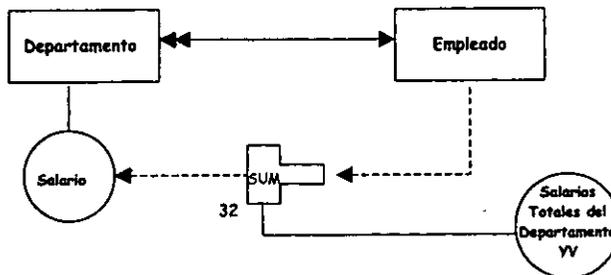
4.2.8.1 Reglas Calculadas.



**Regla 31** El área de un círculo debe ser computada como  $(\pi)r^2$

Este tipo de regla es una forma generalizada de los tipos de reglas de la familia operadores matemáticos, ya que como se puede apreciar la fórmula  $(\pi)r^2$  para calcular el área de un círculo es especificada por el analista. El resultado obtenido se expresa en el valor derivado (YV) de la regla llamado Área.

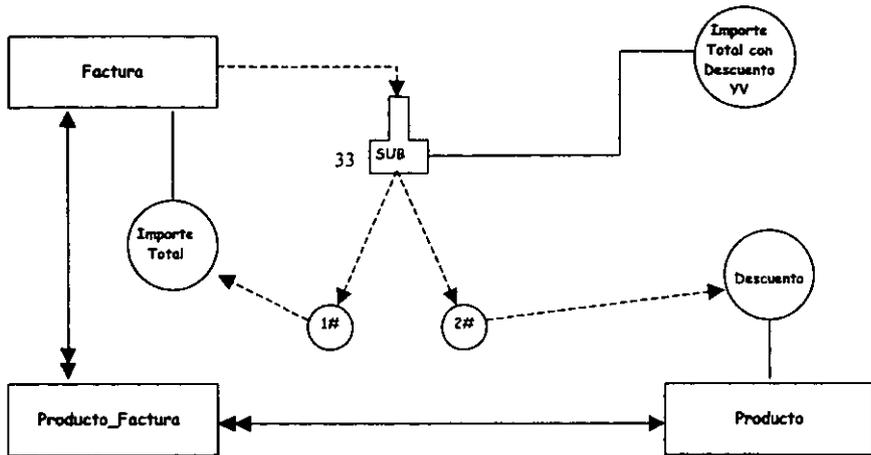
4.2.8.2 Operador Suma



**Regla 32** Los salarios totales del departamento deben ser calculados como la suma de los salarios de todos los empleados que trabajen en el departamento.

Dada una instancia de la entidad departamento la regla 32 automáticamente calcula la suma de los salarios de todos los empleados que trabajan en dicho departamento. El resultado obtenido se expresa en el valor derivado de la regla llamado *Salarios totales del departamento*. Este valor puede ser utilizado por otras reglas, es por eso, que debe representarse en el modelo.

#### 4.2.8.3 Operador Resta

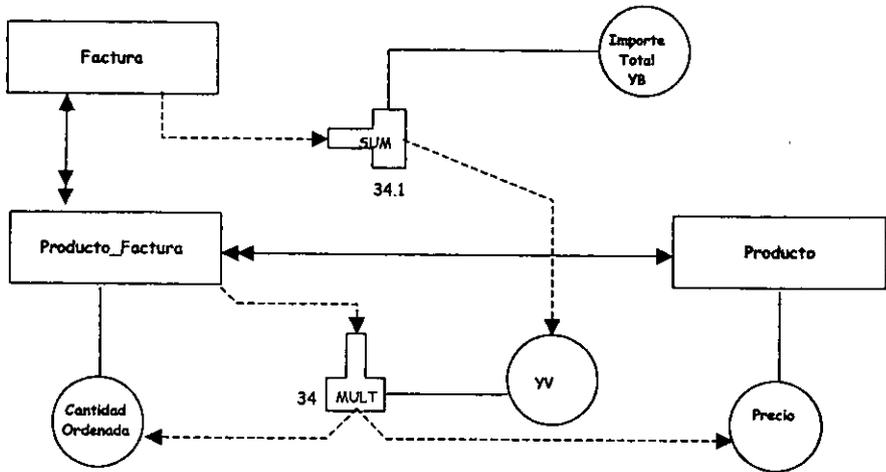


**Regla 33** El importe total con descuento de una factura debe ser calculado de la siguiente manera: *importe total menos el descuento de todos los productos incluidos en la factura.*

Esta regla incluye el uso de identificadores lógicos en el correspondiente de la regla. La utilidad de estos identificadores (círculos pequeños con un número), es indicar el orden de los operandos matemáticos involucrados en el cálculo final. De esta manera, para obtener el importe total con descuento de una factura, se requiere al primer operando que es el valor del Importe total cuya obtención se asume ya se encuentra modelada por otra regla.

El segundo y último operando es el descuento de todos los productos incluidos en la factura que se encuentra identificado lógicamente por el círculo con el número 2. Para una instancia de factura, la regla resta al valor del atributo importe total, el valor del atributo descuento y el resultado lo deja disponible en el valor derivado Importe total con descuento.

#### 4.2.8.4 Operador Multiplicación



**Regla 34** El importe total de una factura debe ser calculado como la suma de cada uno de los totales parciales obtenidos de multiplicar la cantidad ordenada de cada producto por su precio.

La regla 34 multiplica el valor del atributo precio de un producto por su cantidad ordenada en la factura y pone el resultado en el valor derivado de la regla que es el correspondiente de la regla 34.1 (una condición de tipo suma) que se encarga de sumar todos los resultados de la regla 34 (instancias de su valor derivado) para obtener el importe total para una instancia de la entidad factura.

#### **4.2.9 Controladores de proyección**

Cuando una regla es utilizada como una restricción de integridad, su objetivo principal es rechazar cualquier actualización no válida sobre los datos persistentes, por ejemplo, si una actualización esta en proceso, una regla específica se dispara para asegurar la integridad de los datos y si la regla no se cumple, entonces la actualización simplemente no se ejecuta. Sin embargo, existe un conjunto de reglas que se comportan de manera diferente, esto es, cuando la regla en cuestión se dispara el proceso de actualización en curso siempre es permitido por la regla. La regla simplemente proyecta el valor de la instancia del ancla al correspondiente de una manera adecuada según el tipo de regla. A este conjunto de reglas se les llama controladores de proyección y sus tres tipos básicos son los siguientes:

##### **4.2.9.1 Habilitadores (EA)**

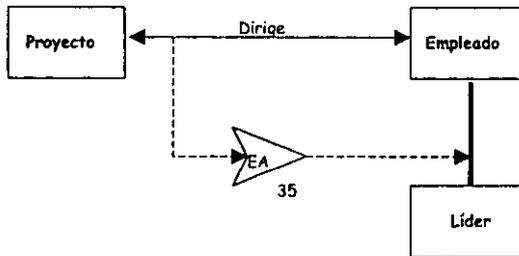
Este tipo de reglas crean (habilitan) instancias del correspondiente cuando una instancia del ancla existe. Durante el proceso de modelado de estas reglas es importante no perder de vista los siguientes puntos:

- Si el correspondiente es una entidad entonces la regla (habilitador) crea una instancia.
- Si el correspondiente es otra regla entonces ésta es habilitada.

En algunos casos, el habilitador puede tener un calificador de tipo negación en su correspondiente, en cuyo caso éste tiene el poder de borrar y no de crear instancias del correspondiente. De esta manera se derivan las siguientes convenciones.

- Si el correspondiente es una entidad, el habilitador borra las instancias correspondientes al ancla.
- Si el correspondiente es otra regla entonces ésta no es habilitada.

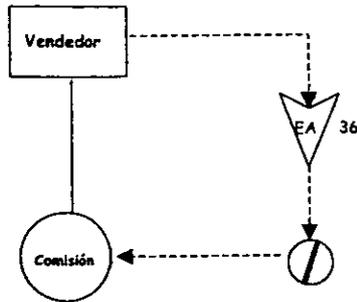
Algunos ejemplos del uso de este tipo de reglas se muestra a continuación:



**Regla 35 un empleado asignado a dirigir un proyecto automáticamente es un líder de proyecto.**

La regla 35 es una restricción de integridad de tipo habilitador. Una vez que la instancia del ancla de la regla 35 (la relación *dirige*), se establece, la regla exige que la instancia asociada de empleado debe ser ahora de tipo líder. La regla automáticamente deberá crear el sub-tipo líder para el empleado correspondiente.

El siguiente ejemplo muestra el uso de un calificador de negación en el correspondiente de la regla para ilustrar los casos en los que el habilitador automáticamente borra las instancias del correspondiente.

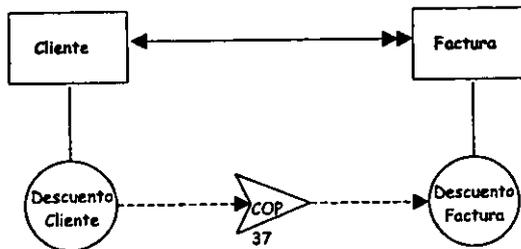


**Regla 36** Un vendedor no debe recibir comisión.

La regla 36 es una restricción de integridad de tipo habilitador que automáticamente borra cualquier posible valor para el atributo comisión para una instancia de la entidad vendedor. El calificador negación (círculo con la diagonal) especifica que las instancias del correspondiente no debe existir.

#### 4.2.9.2 Copiadores (COP)

Establece que el valor de cada instancia del correspondiente debe tener una copia exacta del valor de la instancia del ancla. Este tipo de reglas tiene la forma: "el valor debe verse reflejado". Ejemplo:

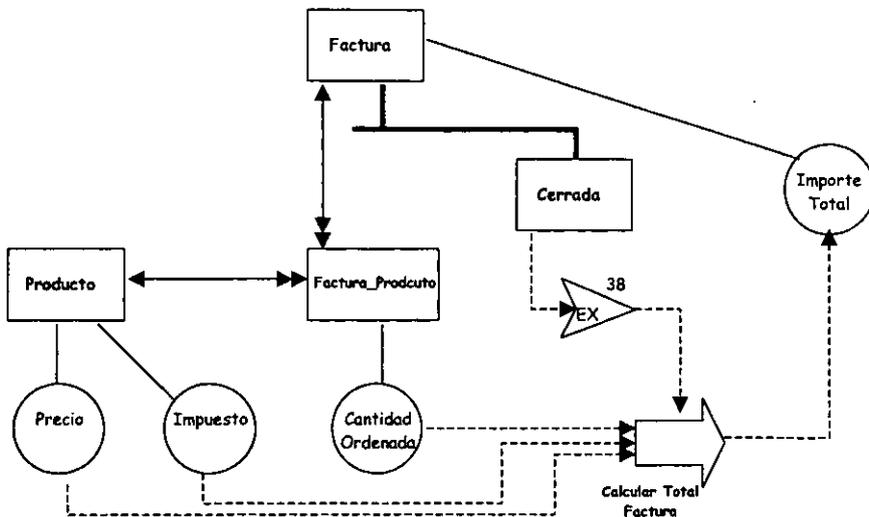


**Regla 37** El descuento que se aplique a una factura debe ser automáticamente el descuento que se le otorgó al cliente que ordena la factura.

La regla 37 es una restricción de integridad de tipo copiador, que establece para cada instancia del ancla de la regla se debe copiar su valor a todas las instancias relacionadas del correspondiente (Descuento de la Factura).

#### 4.2.9.3 Ejecutantes (EX)

Todo sistema de información es un conjunto de partes (módulos) que conviven entre sí para lograr un objetivo común. Muchas de las veces, un sistema debe coexistir con otros sistemas requiriendo de los servicios que brindan los módulos de ese sistema. Estos módulos o componentes incluyen acciones y reglas que se encuentran contruidos en un archivo ejecutable que puede ser invocado en cualquier momento, ya sea directamente por el usuario o bien por otro sistema, el cual, determina la utilización de los servicios del componente basándose en la reglas del negocio. Este es precisamente el propósito del tipo de reglas ejecutorias. Por ejemplo.



**Regla 38** El importe total de una factura debe ser calculado cuando la factura se cierre.

La regla 38 es una restricción de integridad de tipo ejecutoria que establece para una instancia del ancla (factura cerrada) se ejecute el cálculo del importe total de la factura.

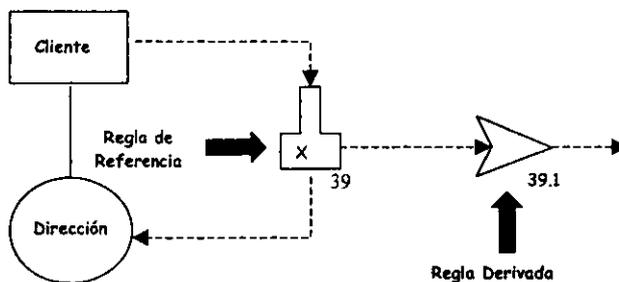
### 4.3 Interconectando reglas del negocio.

Como se estableció en el Capítulo 3, una regla derivada es aquella que se obtiene a partir de otras reglas, lo que sugiera que éstas deben interconectarse creando de esta manera el sistema nervioso bajo el cual opera la organización. Para modelar estas reglas existen dos formas que son las siguientes:

- **Reglas Derivadas.** Cuando una regla asume el papel del ancla de otra regla.
- **Reglas de Inclusión.** Cuando una regla tiene a otra regla como su correspondiente.

#### 4.3.1 Reglas Derivadas

Son aquellas que se obtienen cuando se indica a una regla como el ancla de otra regla. La regla obtenida se llama regla derivada y su ancla es llamada regla de referencia. Por ejemplo:

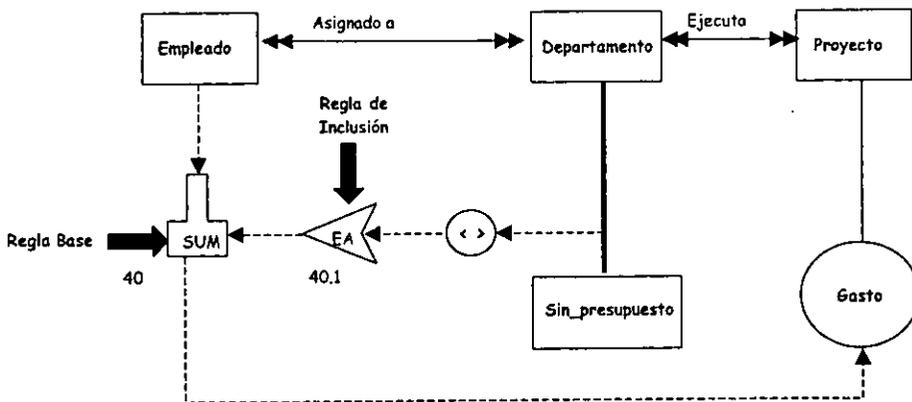


**Regla 39** Si un cliente tiene una dirección, entonces ejecutar la regla 39.1

La regla 39 es una condición de tipo mandatorio que actúa como el ancla de la regla 39.1 que es una regla derivada y por lo tanto, depende del resultado de la regla 39, si el resultado es verdadero, entonces la regla 39.1 es ejecutada, en el caso contrario la regla derivada no se dispara.

#### 4.3.2 Reglas de inclusión.

Se obtienen cuando una regla se convierte en el correspondiente de otra regla. La regla que sirve de correspondiente es llamada Regla Base y, la que se obtiene se conoce como regla de inclusión. Estas reglas de inclusión, frecuentemente afectan la interpretación que debe darse a la regla base, es decir, qué instancias y de qué manera la regla de inferencia se aplica. Un ejemplo de este tipo de reglas es el siguiente.



**Regla No. 40** Para un empleado asignado a un departamento sin presupuesto se debe obtener la suma de los gastos de cada proyecto llevado a cabo por los departamentos a los que el empleado es asignado.

Es importante observar que la conexión de la regla de inclusión con su correspondiente siempre es a un lado del símbolo que representa la regla base.

Si la regla de inclusión 40.1 no hubiese sido definida para la regla 40 ésta habría sido evaluada para las siguientes instancias:

- Cada instancia de la entidad empleado.
- El gasto de cada proyecto al que el empleado se vincula de acuerdo a las relaciones: *asignado a y ejecutado*, es decir, no importa si el departamento tiene presupuesto o no.

Sin embargo, la utilización de una regla de inclusión cambia la interpretación de la regla 40. Esta regla de inclusión tiene el número 40.1 y como se aprecia, utiliza el interprete llamado *seleccionador* que se aplica al ancla y al correspondiente de la regla. El efecto de éste interprete es restringir la evaluación de la regla 40 únicamente a aquellos empleados asignados a un departamento sin presupuesto. De esta manera, la regla 40 permanece inhabilitada para aquellos empleados no asignados a un departamento sin presupuesto. Además, como el interprete se aplica tanto al ancla de la regla como al correspondiente, entonces, el criterio de selección del interprete también es aplicado al correspondiente de la regla 40, por lo tanto, únicamente para aquellos empleados que pertenecen a un departamento sin presupuesto se obtendrá la suma de los gastos de cada proyecto que directamente haya sido llevado a cabo por el departamento sin presupuesto.

## **SISTEMA DE SEGURIDAD SISE**

### **5.1 Introducción**

Con el objeto de aplicar el cambio propuesto por el enfoque de reglas del negocio en la construcción de sistemas de información, a continuación presento un caso práctico que expone la manera de modelar y especificar formalmente las reglas de operación de un negocio. El caso que se presenta no contempla la implantación de las reglas de negocio que se modelen, ya que esto es una actividad meramente mecánica que cualquier equipo de desarrolladores puede realizar una vez que recibe los resultados de las actividades de análisis y diseño; etapas que son cruciales y determinantes en la calidad del producto final.

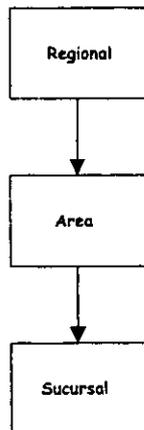
El caso propuesto se apega a la base teórica presentada en capítulos previos y en la utilización de un lenguaje basado en la metodología propuesta por Ronal R. Ross, para documentar y modelar reglas del negocio.

## 5.2 Sistema de Seguridad (SISE)

A fin de mostrar la manera de modelar las reglas de gestión de una organización de acuerdo al enfoque de reglas del negocio, tomaré como base el sistema de seguridad SISE del banco "X" que se encuentra actualmente en operación.

El banco "X" consciente de ser más competitivo y de satisfacer las exigencias de los clientes que cada vez demandan más y mejores servicios, ha emprendido una política de optimización de recursos materiales y financieros. De esta manera, a creado una cultura de ahorro que reflejará el crecimiento institucional, cultural y económico de la organización, ya que ahorrar no significa recortar, sino cuidar los recursos y optimizarlos para acrecentarlos y aumentar los beneficios.

Para lograr lo anterior, el banco desarrolló el Sistema de Control de Energía (SCE) para controlar el consumo de energía eléctrica de todas sus oficinas administrativas, sucursales operativas e instalaciones que se encuentran extendidas en toda la República Mexicana. Estas sucursales están agrupadas de acuerdo a la siguiente estructura jerárquica:



La información que cada sucursal genera por concepto de consumo de energía eléctrica es accesada al sistema SCE por los usuarios autorizados. Como el SCE es un sistema centralizado está diseñado para que cada usuario accese sus servicios de alta de recibos de consumo de energía eléctrica; alta, baja y consultas de programas y presupuestos de ahorro de energía, registro de gastos, etc. de manera remota vía Internet desde su lugar de origen; surgiendo de esta manera, la necesidad de crear un sistema de seguridad (SISE) responsable de autenticar a los usuarios del SCE con el objetivo de proteger la información controlada por éste.

El sistema de seguridad SISE, es responsable de permitir sólo a aquellos usuarios autorizados modificar y consultar la información de consumo de corriente eléctrica de las sucursales, divisiones o regionales que controla el SCE.

### **5.3 Políticas y Reglas del Negocio del SISE**

Es precisamente este sistema de seguridad el que presento, con el fin de mostrar la manera de modelar las reglas del negocio. Actualmente, el sistema tiene implantadas las siguientes políticas y reglas:

- El acceso al sistema SCE deberá ser controlado por un sistema de seguridad responsable de autenticar a los usuarios y asegurarse que éstos tienen los permisos (nivel de autorización) para la ejecución de los servicios ofrecidos por el SCE, así como a la información controlada por éste.

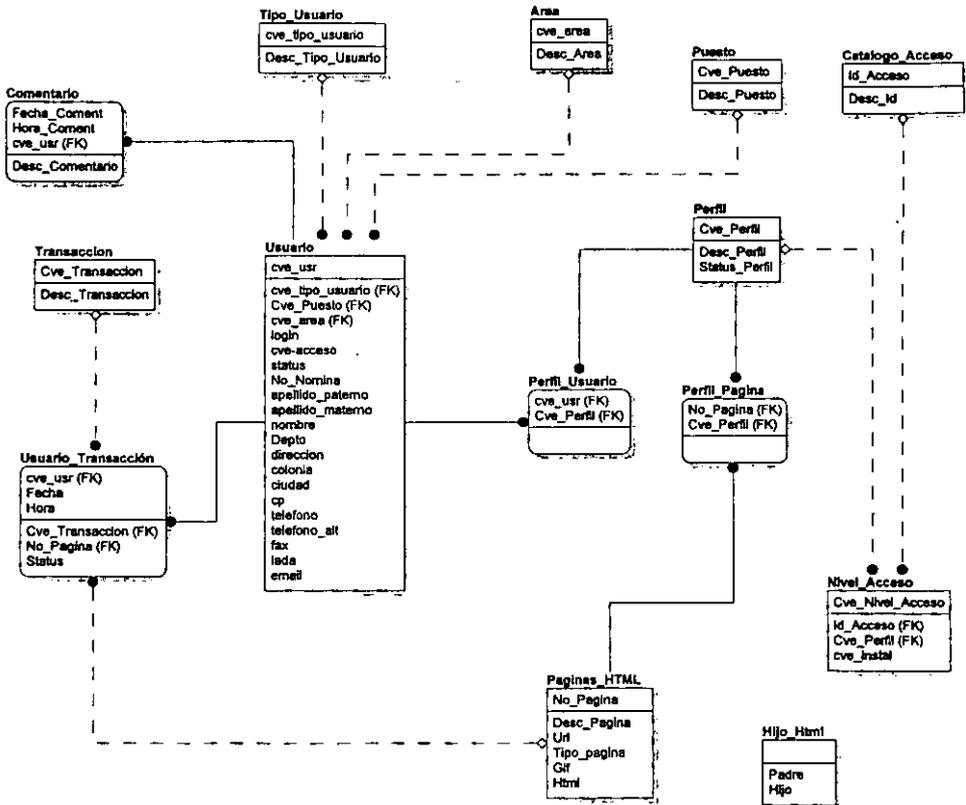
- Para verificar que el usuario que está intentando acceder el sistema SCE, es un usuario autorizado debe tener un nombre de usuario (login) y clave de acceso (password).
- Cada usuario del SCE además de tener un login y password debe contar con un identificador único (clave de usuario).
- El usuario debe contar con un status que determine si éste se encuentra habilitado o suspendido.
- El SISE debe de proporcionar toda la información necesaria para poder deslindar responsabilidades de quién, cuándo y qué se realizó.
- Por cada transacción que el usuario realice, se debe registrar fecha, hora, tipo de transacción, status de ésta, módulo (Página HTML) sobre el que se llevo a cabo la operación y la clave del usuario que la ejecutó.
- Cada transacción ejecutada por un usuario debe contar con un status de EXITO o FRACASO.
- Cada usuario que se de alta en el SISE debe contar con una clave de puesto, área, departamento, tipo de usuario, status, nombre, apellido paterno y No. de nómina.
- Cada usuario que deba de ser autenticado por el SISE debe tener un tipo que puede ser: Capturista, Analista o Administrador.

- El SISE debe restringir el acceso a los usuarios del SCE a través de los siguientes niveles de autorización:
  - *Nivel de acceso*, es decir, la información de qué regionales, divisiones y sucursales puede acceder el usuario.
  - *Acceso a módulos*, es decir, establecer que módulos del SCE (Presupuestos, Proyectos, Gastos, etc.) tiene derecho de acceder, así como las transacciones que puede ejecutar sobre éstos.
- Para implantar el nivel de acceso se deberán crear perfiles de usuario.
- Para cada perfil de usuario se debe establecer su nivel de acceso, es decir, las regionales, divisiones y sucursales que puede acceder.
- Un usuario puede tener varios perfiles dependiendo de su tipo de usuario.
- Si un usuario es del tipo capturista sólo puede tener un perfil de usuario
- Si un usuario es del tipo analista solo puede tener 5 perfiles de usuario.
- El perfil de usuario puede estar activo o inactivo, si este se encuentra activo, quiere decir que el usuario puede realizar todas las transacciones para las que esté facultado, en otro caso no podrá realizar ningún movimiento.

- Cada módulo del SCE, así como las transacciones ofrecidas por cada uno de éstos deben ser páginas HTML.
- Para implantar el nivel de acceso a módulos se debe especificar por cada perfil cuáles son los módulos y transacciones (páginas HTML) del SCE que éste puede acceder.
- Las transacciones (operaciones) que cada uno de los módulos del SCE puede ofrecer son: Actualizaciones, Consultas, Bajas, Altas y Reportes
- Cada página HTML debe pertenecer a una página HTML que será su página madre, pero también una página hija, puede ser la página madre de otra página. De esta manera, una página HTML puede tener muchas páginas hijas y cada página sólo debe pertenecer a una sola página.
- Cada perfil debe tener una página HTML que contenga todos los módulos (menú principal) que éste puede acceder.
- Para identificar que una página HTML es el menú principal de un perfil de usuario, el valor de su atributo `tipo_página` debe ser "Menú"
- Si el valor del atributo `tipo_página` de la entidad `página_HTML` es "Standard", entonces ésta es una página HTML que representa a un módulo del sistema SCE o una transacción ofrecida por un módulo del SCE.

Cada una de las reglas y políticas anteriores se encuentran contenidas en el siguiente modelo de datos:

Modelo de Datos: Sistema de Seguridad  
 Noviembre de 1998



#### 5.4 Reglas del Negocio con deficiencias en su representación por el Modelo de Datos.

A pesar de que el modelo de datos representa un punto de partida para expresar una multitud de reglas del negocio, éste, sin embargo mantiene ocultas muchas de la reglas anteriores y no proporciona una total claridad en la interpretación de ciertas reglas. Por ejemplo: el modelo de datos muestra las siguientes deficiencias en la representación e interpretación de las siguientes reglas:

- ***El usuario debe contar con un status que determine si éste se encuentra habilitado o suspendido.***

El modelo de datos anterior no especifica claramente que una instancia de usuario debe estar asociada con uno de los dos subtipos de usuario: habilitado o suspendido, dicho de otra manera, cuáles son los valores que el atributo status puede tomar en cualquier momento.

- ***Cada transacción ejecutada por un usuario debe contar con un status de EXITO o FRACASO.***

Al igual que la regla anterior, el modelo de datos no representa la regla de manera clara ya que no estipula momento alguno que una instancia de la entidad transacción debe estar asociada con uno de los dos subtipos de transacción: EXITO o FRACASO, dicho de otra manera, cuáles son los valores que el atributo status puede tomar en momento dado.

- **Cada usuario que deba de ser autenticado por el SISE debe tener un tipo que puede ser: Capturista, Analista o Administrador.**

En el modelo de datos no se establece claramente que una instancia de usuario debe estar asociada con uno de los tres subtipos de usuario: Capturista, Analista o Administrador, dicho de otra manera, cuáles son los valores que el atributo `cve_tipo_usuario` puede tomar.

- **El perfil de usuario puede estar activo o inactivo, si éste se encuentra activo, quiere decir que el usuario puede realizar todas las transacciones para las que esté facultado, en otro caso, no podrá realizar ningún movimiento.**

En el modelo de datos no se define que una instancia de la entidad perfil debe estar asociada con uno de los dos subtipos de perfiles: activo o inactivo, dicho de otra manera, cuáles son los valores que el atributo `status` puede tomar en cualquier momento.

- **Si un usuario es del tipo capturista solo puede tener un perfil de usuario**

La técnica de modelado de datos empleada para construir el modelo no es capaz de establecer cuál es el número de instancias de la entidad perfil que puede tener una instancia de la entidad usuario. El modelo de datos sólo expresa que un usuario puede tener asociadas varios perfiles de acceso, sin embargo, éste no especifica que el número de perfiles de acceso por usuario está limitado según su tipo.

- **Si un usuario es del tipo analista solo puede tener 5 perfiles de usuario**

La deficiencia del modelo de datos al especificar esta regla del negocio es la misma que la anterior.

- **Las transacciones (operaciones) que cada uno de los módulos del SCE puede ofrecer son: Actualizaciones, Consultas, Bajas, Altas y Reportes.**

En el modelo de datos no se especifica que una instancia de la entidad perfil debe estar asociada con uno de los dos subtipos de perfiles: activo o inactivo, dicho de otra manera, cuáles son los valores que el atributo status puede tomar en cualquier momento.

- **Cada página HTML debe pertenecer a una página HTML que será su página madre, pero también una página hija, puede ser la página madre de otra página. De esta manera una página HTML puede tener muchas paginas hijas y cada página sólo debe pertenecer a una sola página.**

En el modelo presentado anteriormente en el que esta regla se encuentra contenida, no se especifica en ningún momento que la entidad página\_HTML es una entidad recursiva. De hecho para conocer cuales son las paginas hijas de una página HTML madre se utiliza la entidad hijos que es donde se registran las instancias de las páginas hijas que tienen una página madre.

Pero aunque la definición de esta regla se hubiera hecho estableciendo a la entidad `página_HTML` como una entidad recursiva ésta presentaría las siguientes deficiencias: El modelo no indicaría ninguna restricción para afirmar que una página HTML puede ser parte de sí misma, o bien, en una relación más compleja, se podría asumir que la página No. 1 puede ser parte de la página No. 2, misma que a su vez, es parte de la página No. 1.

- ***Cada perfil debe tener una página HTML que contenga todos los módulos (menú principal) que éste puede acceder.***

El modelo no indica que una vez que se han indicado cuales son las paginas HTML que un perfil de usuario puede acceder (instancias de la entidad `perfil_página`), sólo una de éstas debe ser el menú principal de dicho perfil. Incluso el modelo no restringe que una vez que la relación entre las entidades `perfil` y `página_HTML` se cumple, la primera página que se asocie al perfil debe ser el menú principal.

- ***Si el valor del atributo `tipo_página` de la entidad `pagina_HTML` es "Menú", entonces ésta es el menú principal.***

El modelo de datos no define que una instancia de la entidad `página_HTML` debe estar asociada con uno de los dos subtipos de páginas: `menú` o `standard`, dicho de otra manera, cuáles son los valores que el atributo `tipo_página` puede tomar en cualquier momento.

- ***Si el valor del atributo tipo\_pagina de la entidad página\_HTML es "Standard", entonces ésta es una página HTML que representa a un módulo del sistema SCE o una transacción ofrecida por un módulo del SCE***

La deficiencia presentada por el modelo de datos al representar esta regla es la misma de la regla anterior.

Para corregir las deficiencias anteriores originadas por la especificación de las reglas del negocio en el modelo de datos, adicionar una mayor claridad al modelo de datos, extender la especificación formal de las reglas de gestión, así como expresar las reglas del negocio libres de ambigüedades en su interpretación, a continuación presento la manera de modelar dichas reglas de acuerdo al lenguaje definido en el Capítulo 4 , "Modelando Reglas del negocio".

Todas la reglas del negocio que se identificaron serán modeladas utilizando éste lenguaje, de esta manera aquellas reglas que no presentan deficiencias en su especificación mediante el modelo de datos simplemente se les adicionara una mayor claridad en su definición, sin embargo, aquellas que presentaron deficiencias en su especificación serán corregidas al modelarse bajo el enfoque de reglas del negocio.

### 5.5 Identificando los Business Rambling, Business Rule Statements y Reglas del Negocio

De acuerdo a la metodología de reglas del negocio, primero es necesario identificar las políticas generales de la empresa encaminadas a orientar su manera de operar. Una vez identificadas es necesario estudiarlas y descomponerlas en reglas cada vez más atómicas que puedan ser implantadas directamente.

**Business Rambling  
(Política)**

- *El acceso al sistema SCE deberá ser controlado por un sistema de seguridad responsable de autenticar a los usuarios y asegurarse que éstos tienen los permisos (nivel de autorización) para la ejecución de los servicios ofrecidos por el SCE, así como a la información de las oficinas administrativas que éste controla.*

**Business Rule Statement**

- El sistema SISE debe implantar un nivel de autenticación de usuarios.

**Reglas del Negocio**

- Para verificar que el usuario que está intentando acceder el sistema SCE, es un usuario autorizado debe tener un nombre de usuario (login) y clave de acceso (password).
- Cada usuario del SCE además de tener un login y password debe contar con un identificador único (clave de usuario).
- El usuario debe contar con un status que determine si éste se encuentra habilitado o suspendido.

**Business Rule Statements**

- El SISE debe de proporcionar toda la información necesaria para poder deslindar responsabilidades de quién, cuándo y qué se realizó.

### Reglas del Negocio

- Por cada transacción que el usuario realice, se debe registrar fecha, hora, tipo de transacción, status de ésta, modulo (Página HTML) sobre el que se llevo acabo la operación y la clave del usuario que la ejecutó.
- Cada transacción ejecutada por un usuario debe contar con un status de EXITO o FRACASO.
- Cada usuario que se de alta en el SISE debe contar con una clave de puesto, área, departamento, tipo de usuario, status, nombre, apellido paterno y No. de nómina.
- Cada usuario que deba de ser autenticado por el SISE debe tener un tipo que puede ser: Capturista, Analista o Administrador.

**Business Rule Statements**

- El Sistema de Seguridad debe restringir el acceso a los usuarios del SCE a través de los siguientes niveles de autorización:
  - *Nivel de acceso*, es decir, la información de qué regionales, divisiones y sucursales el usuario puede acceder.
  - *Acceso a módulos*, es decir, establecer que módulos del SCE (Presupuestos, Proyectos, Gastos, etc.) tiene derecho de acceder.
  - *Ejecución de Transacciones*, definir el tipo de operaciones que el usuario puede ejecutar sobre los módulos sobre los que tiene permisos de utilizar.

### Reglas del Negocio

- Para implantar el nivel de acceso se crearán *perfiles de usuario* que deben tener un identificador único (*cve\_perfil*), una descripción y un status.
- El perfil de usuario puede estar activo o inactivo, si éste se encuentra activo, quiere decir que el usuario puede realizar todas las transacciones para las que esté facultado, en otro caso no podrá realizar ningún movimiento.
- Para cada perfil de usuario se debe establecer su nivel de acceso, es decir, las regionales, divisiones y sucursales que puede acceder.
- Un usuario puede tener varios perfiles dependiendo de su tipo de usuario.
  - Si un usuario es del tipo *capturista* solo puede tener un perfil de usuario
  - Si un usuario es del tipo *analista* solo puede tener 5 perfiles de usuario
- Cada módulo del SCE, así como las transacciones ofrecidas por cada uno de éstos deben ser páginas HTML.
  - Para implantar el nivel de acceso a módulos se debe especificar por cada perfil cuáles son los módulos y transacciones (páginas HTML)

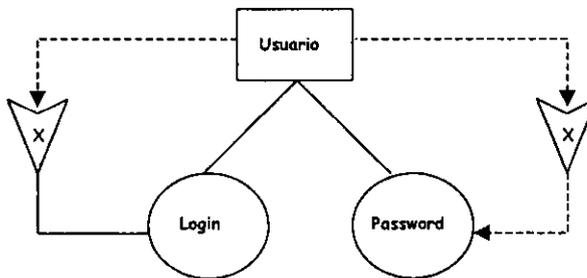
### Reglas del Negocio

- Las transacciones (operaciones) que cada uno de los módulos del SCE puede ofrecer son: Actualizaciones, Consultas, Bajas, Altas y Reportes.
- Cada página HTML debe pertenecer a una página HTML que será su página madre, pero también una página hija, puede ser la página madre de otra página. De esta manera una página HTML puede tener muchas paginas hijas y cada página sólo debe pertenecer a una sola página.
- Cada perfil debe tener una página HTML que contenga todos los módulos (menú principal) que éste puede acceder.
- Si el valor del atributo tipo\_pagina de la entidad página\_HTML es "Menú", entonces ésta es el menú principal.
- Si el valor del atributo tipo\_pagina de la entidad pagina\_HTML es "Standard", entonces ésta es una página HTML que representa a un módulo del sistema SCE o una transacción ofrecida por un módulo del SCE.

## 5.6 Modelando las Reglas del Negocio del SISE

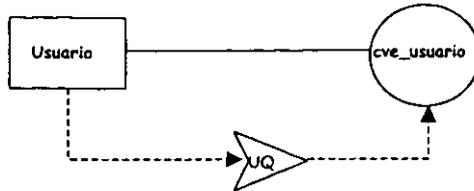
Ya que se han identificado las reglas más atómicas es necesario modelarlas. Esta representación gráfica de las reglas del negocio, obedece en gran parte al hecho que la gran mayoría de los profesionales de la tecnología de la información utilizan técnicas gráficas (Técnicas de Modelado de Datos, Diagrama de Flujo de Datos, Modelos de Objetos, etc.) como una forma de comunicación y establecimiento de acuerdos finales.

**Regla No.1** Para verificar que el usuario que está intentando acceder el sistema SCE, es un usuario autorizado debe tener un nombre de usuario (login) y clave de acceso (password).



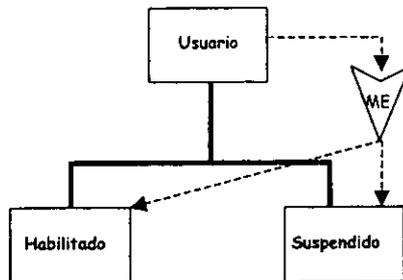
La interpretación final de la regla No. 1 establece que cada instancia de la entidad usuario debe tener un valor en sus atributos login y password. De esta manera, ninguna operación de inserción o actualización que intente violar la regla será permitida.

**Regla No. 2** Cada usuario del SCE además de tener un login y password debe contar con un identificador único (clave de usuario).



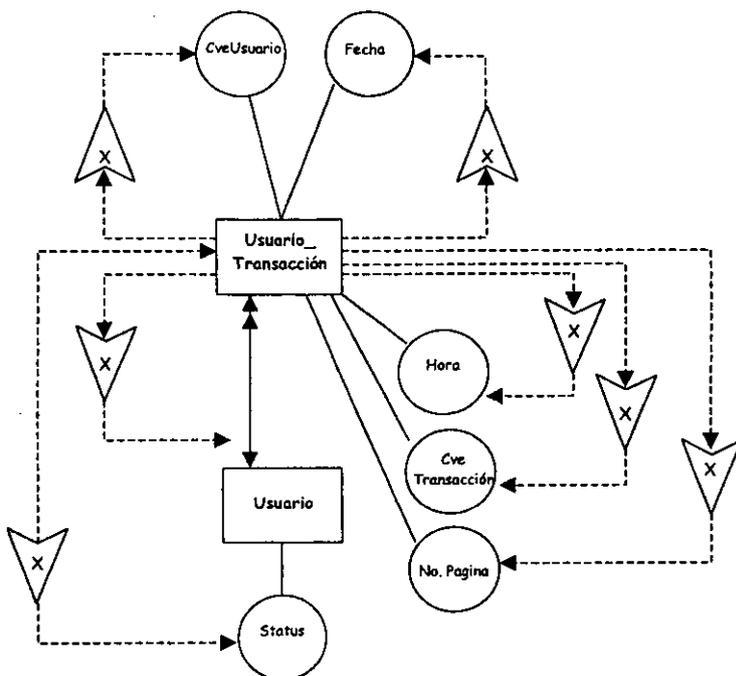
La regla No. 2 no permite que dos o más instancias de la entidad usuario posean el mismo valor del atributo cve\_usuario.

**Regla No. 3** Un usuario debe contar con un status que determine si éste se encuentra habilitado o suspendido.



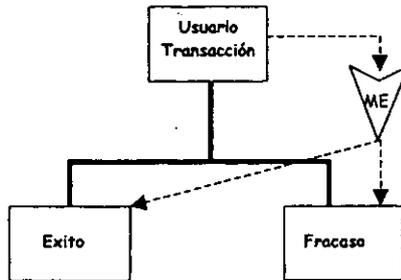
La regla No. 3 especifica que cada instancia de la entidad usuario debe estar asociada con solo uno de los dos subtipos de usuario: Habilitado o Suspendido.

**Regla No. 4** Por cada transacción que el usuario realice, se debe registrar fecha, hora, tipo de transacción, status de ésta, modulo (Página HTML) sobre el que se llevo acabo la operación y la clave del usuario que la ejecutó.



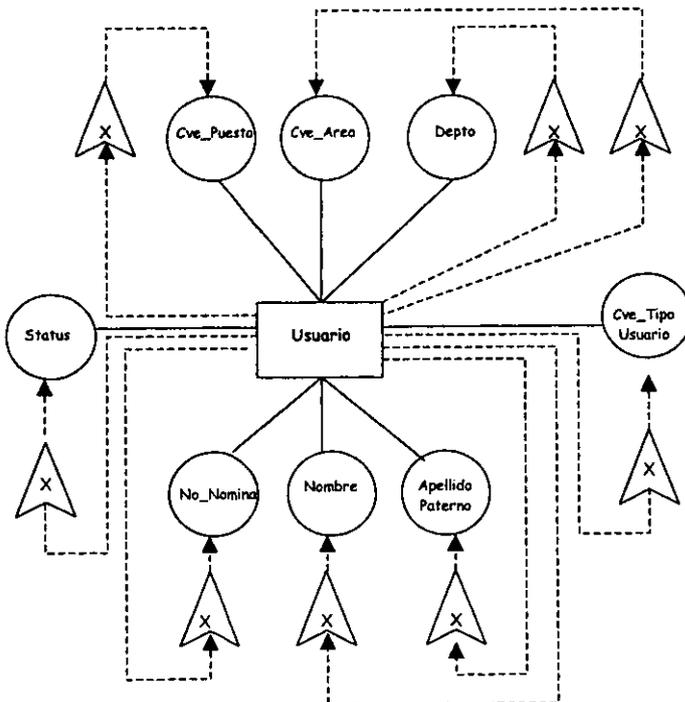
La interpretación de la regla No. 4 dice que cada instancia de la entidad usuario\_transacción debe tener un valor en los siguientes atributos: fecha, hora, tipo de transacción, status de ésta, modulo (Página HTML) sobre el que se llevo acabo la operación y la clave del usuario que la ejecutó. De esta manera cualquier operación de inserción o actualización que intente violar la regla no se permitirá.

Regla No. 5 Cada transacción ejecutada por un usuario debe tener un status de EXITO o FRACASO.



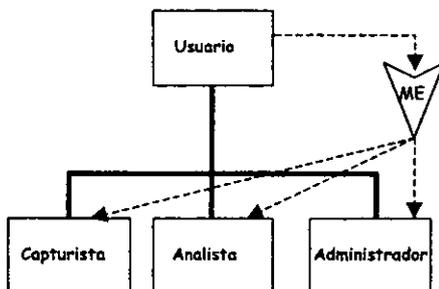
La regla No. 5 especifica que cada instancia de la entidad usuario\_transacción debe estar asociada con solo uno de los dos subtipos de transacción: Exito o Fracaso.

Regla No. 6 Cada usuario que se de alta en el SISE debe tener una clave de puesto, área, departamento, tipo de usuario, status, nombre, apellido paterno y No. de nómina.



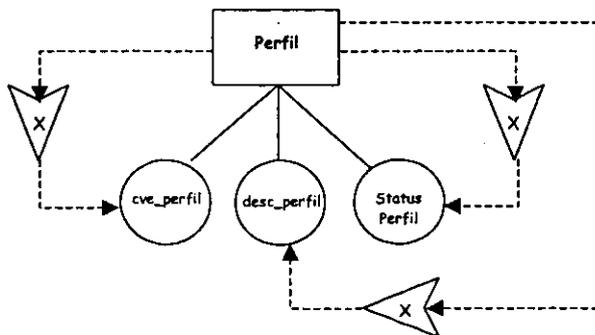
La regla No. 6 establece que cada instancia de la entidad usuario debe tener un valor en los siguientes atributos: clave de puesto, área, departamento, tipo de usuario, status, nombre, apellido paterno y No. de nómina. De esta manera cualquier operación de inserción o actualización que intente violar la regla no se permitirá.

**Regla No. 7** Cada usuario que deba de ser autenticado por el SISE debe tener un tipo que puede ser: **Capturista, Analista o Administrador.**



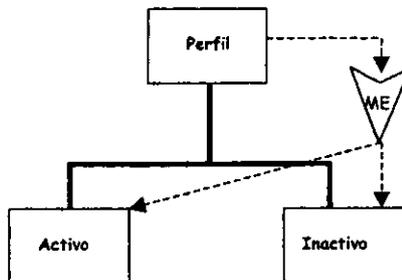
La regla No. 7 requiere que cada instancia de la entidad Usuario debe estar asociada con solo uno de los tres subtipos de usuario: Capturista, Analista o Administrador.

**Regla No. 8** Para implantar el nivel de acceso se crearán *perfiles de usuario* que deben tener un identificador único (cve\_perfil), una descripción y un status.



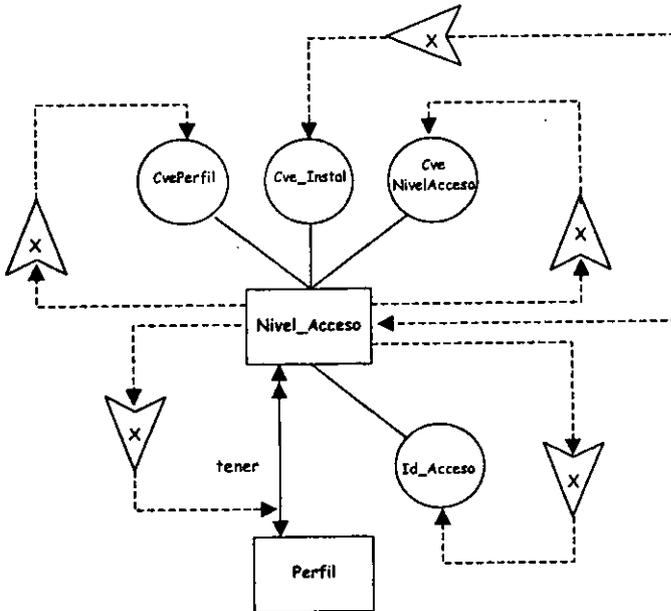
La regla No. 8 establece que cada instancia de la entidad perfil debe tener un identificador único (cve\_perfil), una descripción y un status.. Asegurando de esta manera, que cualquier operación de inserción o actualización que intente violar la regla no se permitirá.

**Regla No. 9** El perfil de usuario puede estar activo o inactivo, si este se encuentra activo, entonces el usuario puede realizar todas las transacciones para las que esté facultado, en otro caso no podrá realizar ningún movimiento.



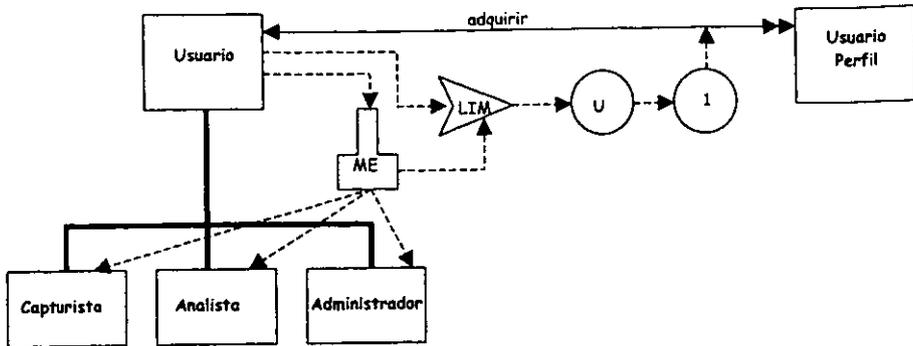
La regla No. 9 requiere que cada instancia de la entidad Perfil debe estar asociada con solo uno de los dos subtipos de perfiles: Activo o Inactivo.

Regla No. 10 Para cada perfil de usuario se debe establecer su nivel de acceso, es decir, si se trata de un acceso a nivel regional, división o sucursales (id\_acceso), las nombres de las respectivas regionales, divisiones y sucursales (cve\_instal), la clave del nivel de acceso y finalmente la clave del perfil.



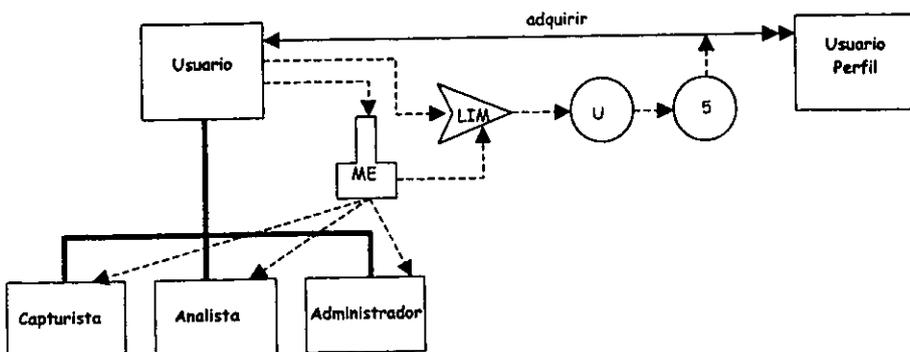
La regla No. 10 establece que cada instancia de la entidad nivel\_acceso requiere que ésta posea una instancia de su correspondiente, la relación tener, es decir, una vez que la relación se establece la regla requiere que para cada instancia de nivel\_acceso debe existir una cve\_perfil, una cve\_instal, cve\_nivel\_acceso y un id\_acceso.

**Regla No. 11** Si un usuario es del tipo capturista entonces solo puede tener un perfil de usuario



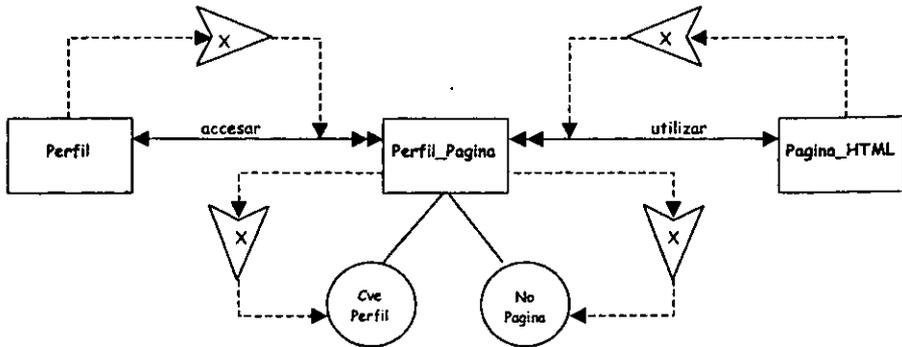
Para poder dar mayor claridad a la especificación de la regla fue necesario el uso de calificadores entre el correspondiente de la regla y el ancla, esto con el objeto de poder precisar el número de instancias del correspondiente. La regla no 11 es disparada por la regla que evalúa cuál es el tipo del usuario, es decir, requiere del resultado de ésta para poder restringir el numero de instancias de la entidad usuario\_perfil una vez que la relación *adquirir* se cumple.

**Regla No. 12** Si un usuario es del tipo analista entonces sólo puede tener 5 perfiles de usuario



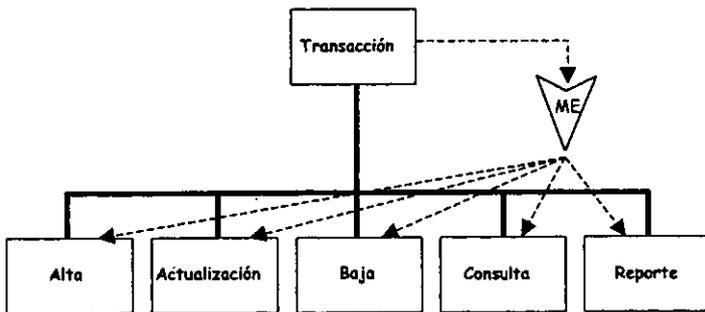
La interpretación de la regla No. 12 es la misma de la regla No. 11

**Regla No. 13** Para implantar el nivel de acceso a módulos se debe especificar por cada perfil cuáles son los módulos y transacciones (páginas HTML) del SCE que éste puede acceder.



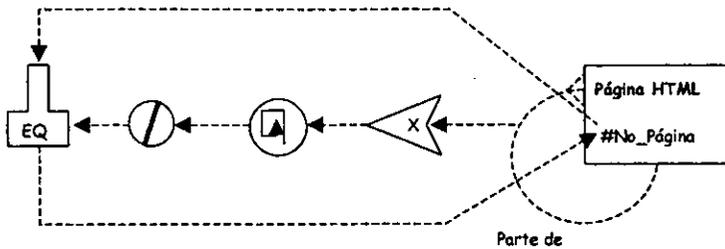
En el momento que las relaciones acceder y utilizar son establecidas la regla No. 13 requiere que para cada instancia de la entidad perfil\_página debe tener un valor los atributos cve\_perfil y No\_página.

**Regla No. 14.** Los tipos de transacciones (operaciones) que cada uno de los módulos del SCE puede ofrecer son: **Altas, Actualizaciones, Bajas, Consultas y Reportes.**



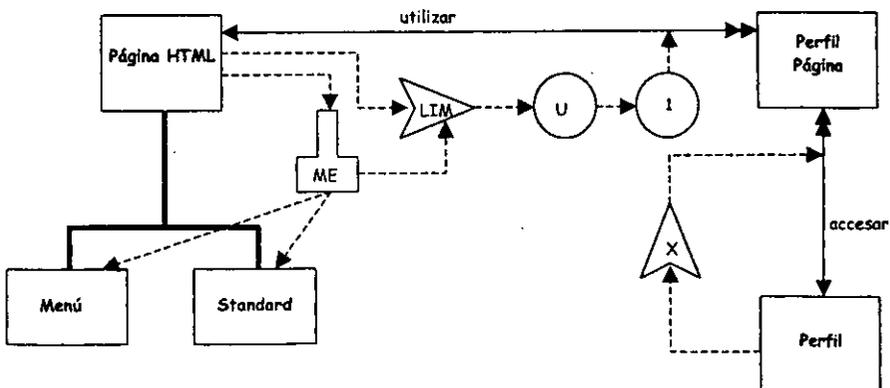
La regla No. 14 especifica que cada instancia de la entidad Transacción debe estar asociada con solo uno de los 5 subtipos de transacción: Alta, Actualización, Baja, Consulta o Reporte.

Regla No. 15 Cada página HTML debe pertenecer a una página HTML que será su página madre, pero también una página hija, puede ser la página madre de otra página. De esta manera, una página HTML puede tener muchas paginas hijas y cada página sólo debe pertenecer a una sola página.



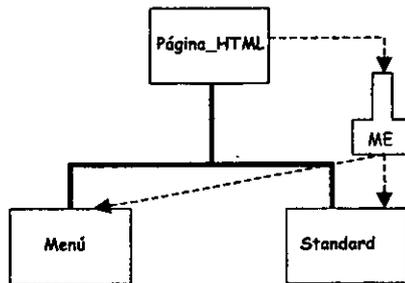
La regla No. 15 se dispara cada vez que la relación *parte de*, se cumple. Así, la regla de tipo EQ (evaluador comparativo), establece que los No\_página de dos ocurrencias de la entidad página\_HTML podrían ser iguales, sin embargo, la regla No. 15 (constraint), requiere que éstos no sean iguales, lo cual se logra mediante el uso del calificador de negación (círculo con la diagonal). De esta manera, la regla No. 15 no permite que ninguna operación de inserción o actualización de una instancia de la entidad página\_HTML que tenga No\_página iguales sea admitida.

Regla No. 16 Cada perfil debe tener una página HTML que contenga todos los módulos (menú principal) que éste puede acceder.

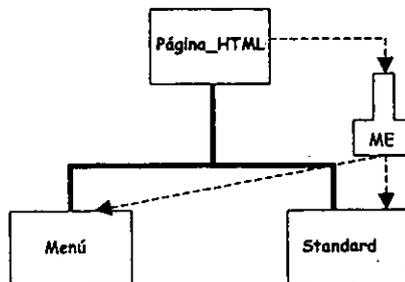


Al igual que la regla No. 11 para poder dar mayor claridad a la especificación de la regla, fue necesario el uso de calificadores entre el correspondiente de la regla y el ancla, esto con el objeto de poder precisar el número de instancias del correspondiente. La regla no 16, es disparada por la regla que evalúa cuál es el tipo de página, es decir, requiere del resultado de ésta para poder restringir el número de instancias de la entidad perfil\_página una vez que las relaciones utilizar y acceder se cumplen.

**Regla No. 17** Si el valor del atributo tipo\_página de la entidad página\_HTML es "Menú", entonces ésta es el menú principal.



**Regla No. 18** Si el valor del atributo tipo\_página de la entidad página\_HTML es "Standard", entonces ésta es una página HTML que representa a un módulo del sistema SCE o una transacción ofrecida por un módulo del SCE.



Las reglas No. 17 y 18 especifican que cada instancia de la entidad página\_HTML debe estar asociada con solo uno de los 2 subtipos de paginas HTML: Menú, o Standard.

Con la representación formal de ésta última regla del negocio del SISE concluyó esta tesis, convencido de la utilidad que el enfoque de reglas del negocio brinda a la comunidad informática en la construcción de sistemas de información de alta calidad. al proponer un cambio de filosofía para la especificación formal de las reglas del negocio en la etapa de análisis y diseño.

## CONCLUSIONES

El desarrollo de esta tesis brinda los elementos necesarios para llegar a la conclusión más importante que es: la gran utilidad que brinda el enfoque de reglas del negocio a la comunidad informática en la construcción de sistemas de información de alta calidad; al identificar, documentar y modelar formalmente las reglas del negocio durante las etapas de análisis y diseño del ciclo de vida de un sistema. Como es conocido éstas etapas son determinantes en la calidad del software.

Otro punto de primordial importancia que se logra al construir sistemas bajo una cultura de reglas del negocio, es la reducción de la distancia existente entre la gente del negocio y los profesionales de la tecnología de la información; que se logra al establecer un lenguaje gráfico para modelar las reglas del negocio, ya que gran parte de la comunicación y establecimiento de acuerdos finales es mediante el uso de este tipo de técnicas, por ser más claras. Pero también la exigencia de esta filosofía sobre la gente de sistemas para tener la habilidad de identificar y extraer las reglas del negocio durante las juntas de planeación, obligándolos a convertirse en analistas de negocios.

Adicionalmente a los puntos mencionados anteriormente, sabemos que el mantenimiento de sistemas representa un gran problema para las empresas, ya que significa tiempo y dinero. La ventaja de tener claramente definidas e identificadas las reglas del negocio, es que los nuevos requerimientos (reglas) se podrían incorporar al sistema con mayor rapidez, dando a éste la habilidad para asimilar los cambios y por lo tanto, ser capaz de adaptarse. Al final la empresa podría responder con mayor rapidez a los cambios del mercado y mantenerse en la competencia, es decir, evolucionar de acuerdo a lo que en la realidad cambia.

El tratamiento formal a las reglas del negocio propuesto en éste trabajo ofrece además, las siguientes ventajas:

- *Reusabilidad*: Ya que las políticas y prácticas del negocio se aplican a toda la organización.
- *Disminución de inconsistencias*: que actualmente existen en la interpretación de las reglas del negocio que se encuentran embebidas en varios procesos del negocio o aplicaciones de usuario final.
- *Independencia tecnológica*: En lugar de confiar en las restricciones tecnológicas utilizando un enfoque procedural, se debe concentrar la atención en las necesidades de la organización. Dicho correctamente, es poner las prioridades del negocio por encima de las prioridades tecnológicas, ya que cada regla del negocio es una política o práctica operacional que se ejecuta en la empresa, pero que no se refiere de ninguna forma a una estrategia de implantación o una tecnología específica.

Es importante resaltar que uno de los valores agregados que se obtienen al aplicar el enfoque de reglas del negocio al expresar de manera clara y completa las reglas de operación de la organización, es su carácter contractual, entre el usuario y el analista, evitando situaciones desgastantes y malos entendidos que surgen más tarde cuando la etapa de desarrollo se encuentra muy avanzada debido a que ciertos requerimientos o reglas del negocio no fueron considerados o simplemente se pasaron por alto durante la etapa de análisis.

De esta manera, el modelo que expresa completamente las reglas de gestión del negocio representa de alguna manera el contrato de los acuerdos establecidos entre el usuario (gente del negocio) y el analista (tecnología de la información) al momento de iniciar el desarrollo de un sistema.

Por último, quiero decir que el enfoque de reglas del negocio sugiere un cambio en el desarrollo de sistemas, orientado precisamente a incrementar el valor del negocio.

## APÉNDICE A

### CÓDIGOS DE PROGRAMAS EN LENGUAJE "C" Y STORED PROCEDURES

A continuación muestro la automatización de reglas del negocio implantadas en diversos lenguajes de programación.

1. El siguiente código, es un servicio que permite realizar transferencias de fondos entre cuentas de cheques de un banco mediante un teléfono de tonos. Este se encuentra desarrollado en lenguaje C , con SQL Embebido de Informix y con el uso del API de desarrollo del middleware TUXEDO. Como puede apreciarse la regla de negocio se encuentra sepultada en líneas de código procedural lo que dificulta su identificación en futuros mantenimientos, otro de los factores que dificultan su mantenimiento es que la regla simplemente se encuentra implantada y sólo el usuario responsable del área de sistemas que la implantó sabe de la existencia de ella, ya que esta nunca fue documentada durante la etapa de análisis.

**La regla del negocio que se tiene implantada es:**

- 1.- *Un cliente del banco con servicio de Vía Telefónica únicamente podrá realizar transferencias de fondos entre cuentas de cheques por un monto no mayor a \$10,000, por día.*

```
/** VTACHqTrfFnds.ec **/  
/.....  
* Objetivo: Servicio que efectua una transferencia de fondos *  
*   entre dos cuentas de cheques de Banco del Atlantico *  
* Funciones que llama: ChqCargo, ChqAbono, ChqReverso, *  
*   insBitaVTA, insAuxTrfVTA, ObtenConRef *  
* Elaboro: Raul Benítez S. USL de Mexico *  
* Fecha de elaboracion: *  
* Modifico: *  
* Fecha ultima Modificacion: 15/07/96 *  
...../  
#include "vtasimbolic.h"  
#include "mixvtahp.h"  
#include "time.h"  
  
EXEC SQL INCLUDE conref.h;  
  
int VTACHqTrfFnds (TPSVCINFO * req)  
{  
    EXEC SQL BEGIN DECLARE SECTION;  
        dec_t   decMontoAcum,decSumaMtos,decMtoMaxDia;  
        long   iCtoVta;  
        tsConRef   itsConRef;  
    EXEC SQL END DECLARE SECTION;  
  
    /** Declaracion de estructuras VIEW ***/  
    struct vChqTrfFnds *pvChqTrfFnds;
```

```

/** Declaracion de estructuras de entrada al API del CORE */
teChqCargo    iteChqCargo;
teChqAbono    iteChqAbono;
teChqRevOp    iteChqRevOp;

/** Declaracion de estructuras de salida del API del CORE */
tsChqCargo    itsChqCargo;
tsChqAbono    itsChqAbono;
tsErmo        itsErmo;

/** Declaracion de estructuras de entrada/salida del API del CORE */
tDummy        itDummy;

/** Declaracion de variables de trabajo */
int    iRetCargo, iRetAbono, iRetReverso, iRet;
long   lTrans;
long   lProtReverso, lProtCargo, lProtAbono;
int    iRegreso, iCompMontos;
long   lRcode, lPasoCode, lTime;
char   chCero[]="0.0";
dec_t  declmporte;
struct tm *itFecha;
time_t itHoy;

$WHENEVER SQLERROR CALL whenexp_chk;
$WHENEVER SQLWARNING CALL whenexp_chk;

#ifdef DEBUGUEANDO
push_rutina("VTACHqTrfFnds");
#endif

pvChqTrfFnds = (struct vChqTrfFnds *) req -> data ;

lProtReverso = 0; /* el reverso fallo y no regreso numero de proteccion */
lProtCargo = 0; /* El cargo no fue exitoso se debe insertar 0 en bitacora */
lProtAbono = 0; /* El abono no fue exitoso se debe insertar 0 en bitacora */
iRegreso = TPFALL;
lRcode = SRV_NO_DISP;

/* Convertir a dec_t cero para pasarlo como importe en falla ChqCargo */
deccvasc(chCero, strlen(chCero), &declmporte);

lCtoVta = pvChqTrfFnds->lCtoVta;

#ifdef DEBUGUEANDO
strcpy(statement, "SELECT ParametrosVTA");
#endif

EXEC SQL SELECT declMaxMontoTrf
INTO :declMtoMaxDia
FROM ParametrosVTA;
if (sqlca.sqlcode == 100) {
userlog("VTACHqTrfFnds:CTO_NO_EXISTE");
}
#ifdef DEBUGUEANDO
pop_rutina();
#endif

#ifdef TUXDEBUG
userlog("<<VTACHqTrfFnds");
#endif
return(iRegreso, lRcode, (char *)pvChqTrfFnds,
sizeof(struct vChqTrfFnds), 0);
}

```

```

#ifdef DEBUGUEANDO
    strcpy(statement,"SELECT CtosVTA");
#endif

    EXEC SQL SELECT declmteAcum
        INTO :decMontoAcum
        FROM CtosVTA
        WHERE seCtoVTA = :iCtoVta;

    if(sqlca.sqlcode == 100){
        IRcode = CTO_NO_EXISTE;
        userlog("VTACHqTrfFnds:CTO_NO_EXISTE");
#ifdef DEBUGUEANDO
        pop_rutina();
#endif
#ifdef TUXDEBUG
        userlog("<<VTACHqTrfFnds");
#endif

        treturn(iRegreso,IRcode,(char *)pvChqTrfFnds,
            sizeof(struct vChqTrfFnds),0);
    }
    else
        if(sqlca.sqlcode == 0){
            iCompMontos = deccmp(&decMontoAcum,&declmteAcum);
            switch (iCompMontos){
                /*decMontoAcum y declmteAcum son iguales primera transfe*/
                case 0:
                    decadd(&decMontoAcum,&pvChqTrfFnds->decMto,&decSumaMtos);
                    if((deccmp(&pvChqTrfFnds->decMto,&decMtoMaxDia))== 1){
                        IRcode = MTO_EXCEDIDO;
                        deccopy(&decMtoMaxDia,&pvChqTrfFnds->decMtoMax);
                        deccopy(&decMontoAcum,&pvChqTrfFnds->decMtoAcumulado);
                        userlog("VTACHqTrfFnds:MTO_EXCEDIDO");
#ifdef DEBUGUEANDO
                            pop_rutina();
                        #endif
#ifdef TUXDEBUG
                            userlog("<<VTACHqTrfFnds");
                        #endif
                            treturn(iRegreso,IRcode,(char *)pvChqTrfFnds,
                                sizeof(struct vChqTrfFnds),0);
                    }
                    break;
                /*decMontoAcum es mayor que cero no es 1a transferencia*/
                case 1:
                    decadd(&decMontoAcum,&pvChqTrfFnds->decMto,&decSumaMtos);
                    if((deccmp(&decSumaMtos,&decMtoMaxDia))== 1){
                        IRcode = MTO_DIA_EXCEDIDO;
                        deccopy(&decMtoMaxDia,&pvChqTrfFnds->decMtoMax);
                        deccopy(&decMontoAcum,&pvChqTrfFnds->decMtoAcumulado);
                        userlog("VTACHqTrfFnds:MTO_DIA_EXCEDIDO");
#ifdef DEBUGUEANDO
                            pop_rutina();
                        #endif
#ifdef TUXDEBUG
                            userlog("<<VTACHqTrfFnds");
                        #endif
                            treturn(iRegreso,IRcode,(char *)pvChqTrfFnds,
                                sizeof(struct vChqTrfFnds),0);
                    }
                    break;
                default:
                    break;
            }
        }
    }
}
/**FIN SWITCH **/

```

```

/* Copia Argumentos a las estructuras de entrada al API del CORE */

/** Copiando datos del VIEW a estructura de entrada al API-CORE ChqCargo*/

iRet = ObtenConRef(T_CHQ_TRF,&itsConRef);
if(iRet == FRACASO){
    userlog("VTACHqTrfFnds:FRACASO:ObtenConRef()");
#ifdef DEBUGUEANDO
    pop_rutina();
#endif

#ifdef TUXDEBUG
    userlog("<<VTACHqTrfFnds");
#endif

    treturn(iRegreso,IRcode,(char *)pvChqTrfFnds,
        sizeof(struct vChqTrfFnds),0);
}
strcpy(iteChqCargo.sCuenta.val,pvChqTrfFnds->sCtaOrigen);
iteChqCargo.shMoneda = pvChqTrfFnds->shMoneda;
iteChqCargo.shConcepto = itsConRef.shConCargo;
strcpy(iteChqCargo.sReferencia1.val,pvChqTrfFnds->sReferencia1);
strcpy(iteChqCargo.sReferencia2.val,itsConRef.chRef2Cargo);
deccopy(&pvChqTrfFnds->decMto, &iteChqCargo.decImporte);

/** Copiando datos del VIEW a estructura de entrada al API-CORE ChqAbono*/

strcpy(iteChqAbono.sCuenta.val,pvChqTrfFnds->sCtaDestino);
iteChqAbono.shMoneda = pvChqTrfFnds->shMoneda;
iteChqAbono.shConcepto = itsConRef.shConAbono;
strcpy(iteChqAbono.sReferencia1.val,pvChqTrfFnds->sReferencia1);
strcpy(iteChqAbono.sReferencia2.val,itsConRef.chRef2Abono);
deccopy(&pvChqTrfFnds->decMto, &iteChqAbono.decImporte);

if( tx_begin() < 0){
#ifdef DEBUGUEANDO
    pop_rutina();
#endif

#ifdef TUXDEBUG
    userlog("<<VTACHqTrfFnds");
#endif
#ifdef TUXDEBUG
    treturn(iRegreso,IRcode,(char *)pvChqTrfFnds,
        sizeof(struct vChqTrfFnds),0);
}
iRetCargo = ICargoChq(req->appkey,TPNOTRAN,&iteChqCargo,&itsChqCargo);
if( iRetCargo==EXITO ){
    /* Cargo exitoso */
    userlog("VTACHqTrfFnds:Cargo Exitoso: %d", iRetCargo);
    if(((iRetAbono= lAbonoChq(req->appkey,TPNOTRAN,&iteChqAbono,
        &itsChqAbono))==EXITO ){
        /* cargo y abono exitosos */
        userlog("VTACHqTrfFnds:Abono Exitoso: %d", iRetAbono);
        IRcode = EXITO;
    }
}
else{
    /* cargo exito y abono fallido */
    switch (iRetAbono){
        case CTA_NO_EXISTE:
            IPasoCode = CTA_D_NO_EXISTE;
            break;
        case CTA_BLOQ_INTER:
            IPasoCode = CTA_D_BLOQ_INTER;
            break;
        case SRV_NO_DISP:

```

```

        IPasoCode = SRV_NO_DISP;
        break;
    case FRACASO:
        IPasoCode = FRACASO;
        break;
    default:
        break;
}
strcpy(iteChqRevOp.sCuenta.val,pvChqTrfFnds->sCtaOrigen);
iteChqRevOp.shMoneda = pvChqTrfFnds->shMoneda;
iteChqRevOp.iFolio = itsChqCargo.iFolio;

#ifdef TUXDEBUG
userlog("VTACHqTrfFnds:Folio Cargo:%d", iteChqRevOp.iFolio);
#endif

deccopy(&pvChqTrfFnds->decMto,&iteChqRevOp.decIimporte);

iHoy = time(NULL);
iFecha = localtime(&iHoy);

iteChqRevOp.sFecha.shYear = iFecha->tm_year;
iteChqRevOp.sFecha.shMes = iFecha->tm_mon + 1;
iteChqRevOp.sFecha.shDia = iFecha->tm_mday;
iRetReverso = ChqRevOp(req->appkey, TPNOTRAN, &iteChqRevOp,
    &iDummy, &itsErmo);
if( iRetReverso != -1){
    /*cargo exito, abono fallido, reverso exito */
    IRcode = (long)IPasoCode;
    IProtReverso = iRetReverso;
    userlog("VTACHqTrfFnds:Reverso Exitoso:%d", iRetReverso);
}
else{
    /*cargo exito, abono fallido, reverso fallido */
    userlog("VTACHqTrfFnds:ChqRevOp:tpurcode:%d",
        itsErmo.tpurcode);
    userlog("VTACHqTrfFnds:ChqRevOp:tperno:%d:%s",
        itsErmo.tpermo, tpsterror(itsErmo.tpermo));
    IRcode = URGE_TRF_REVERSO;
}
}
else {
    /* cargo fallido */
    switch (iRetCargo){
    case CTA_NO_EXISTE:
        CTA_O_NO_EXISTE;
        break;
    case CTA_BLOQ_INTER:
        IRcode = CTA_O_BLOQ_INTER;
        break;
    case INSUFONDOS:
        IRcode = INSUFONDOS;
        break;
    case MTO_EXCEDIDO:
        IRcode = MTO_EXCEDIDO;
        break;
    case SRV_NO_DISP:
        IRcode = SRV_NO_DISP;
        break;
    case FRACASO:
        IRcode = FRACASO;
        break;
    default:
        break;
    }
}
}
}

```

```

if (IRcode == EXITO){
    tx_commit();
    iRegreso = TPSUCCESS;
    IProtCargo = itsChqCargo.IFolio;
    IProtAbono = itsChqAbono.IFolio;
    decocopy(&pvChqTrfFnds->decMto, &declmporte);
    pvChqTrfFnds->IProtCargo = itsChqCargo.IFolio;
    pvChqTrfFnds->IProtAbono = itsChqAbono.IFolio;
    #ifdef DEBUGUEANDO
        strcpy(statement,"UPDATE CtosVTA");
    #endif
    EXEC SQL UPDATE CtosVTA
        SET declmteAcum = :decSumaMtos
        WHERE seCtoVTA = :ICtoVta;
    if((sqlca.sqlcode == 100)
        userlog("VTACHqTrfFnds:Error:UPDATE CtosVTA");
    )
    else
        tx_rollback();
    /* Inserta en la bitacora de Informix */
    EXEC SQL BEGIN WORK;
    ITrans = insBitaVTA(pvChqTrfFnds->shVpsId, (short int) T_CHQ_TRF,
        (short int) IRcode, pvChqTrfFnds->ICtoVta);
    #ifdef TUXDEBUG
    {
        char slmporte[15];
        userlog("VTACHqTrfFnds:ITrans:[%d]",ITrans);
        userlog("VTACHqTrfFnds:sCtaOrigen:[%s]",pvChqTrfFnds->sCtaOrigen);
        dectoasc(&iteChqCargo.declmporte, slmporte, 12, -1);
        userlog("VTACHqTrfFnds:Cargo:declmporte:[%s]",slmporte);
        dectoasc(&iteChqAbono.declmporte, slmporte, 12, -1);
        userlog("VTACHqTrfFnds:Abono:declmporte:[%s]",slmporte);
        userlog("VTACHqTrfFnds:IProtCargo:[%d]",IProtCargo);
        userlog("VTACHqTrfFnds:IProtReverso:[%d]",IProtReverso);
        userlog("VTACHqTrfFnds:sCtaDestino:[%s]",pvChqTrfFnds->sCtaDestino);
        userlog("VTACHqTrfFnds:IProtAbono:[%d]",IProtAbono);
        userlog("VTACHqTrfFnds:shConCargo: [%d]",iteChqCargo.shConcepto);
        userlog("VTACHqTrfFnds:chRef2Cargo: [%s]",iteChqCargo.sReferencia2.val);
        userlog("VTACHqTrfFnds:shConAbono: [%d]",iteChqAbono.shConcepto);
        userlog("VTACHqTrfFnds:chRef2Abono: [%s]",iteChqAbono.sReferencia2.val);
    }
    #endif

    pvChqTrfFnds->IProtTrf = insAuxTrfVTA(ITrans, pvChqTrfFnds->sCtaOrigen,
        IProtCargo,IProtReverso,
        declmporte, pvChqTrfFnds->sCtaDestino,IProtAbono,
        iteChqCargo.sReferencia1);

    EXEC SQL COMMIT WORK;
    /* como excepcion, cambia el código de error que originalmente
    se pone a URGE_TRF_REVERSO para que vaya a la bitacora y ahora
    se cambia para avisarle al usuario del telefono porque fallo su
    abono en la transferencia (y que solo el banco sepa que tiene
    un error que atender urgentemente */
    if(IRcode == URGE_TRF_REVERSO) {
        /* ojo, aqui viene una rutina que avisa al MAC sobre
        la urgencia de regresarle los fondos al cliente */
        IRcode = IPasoCode;
    }
    #ifdef DEBUGUEANDO
        pop_rutina();
    #endif
    #ifdef TUXDEBUG
        userlog("<<VTACHqTrfFnds");
    #endif
    treturn(iRegreso,IRcode,(char *)pvChqTrfFnds,sizeof(struct vChqTrfFnds),0);
}

```

2. El siguiente ejemplo es un conjunto de Stored Procedures desarrollados en PL/SQL de Oracle Ver. 7.3, que tienen implantadas varias reglas del negocio. El propósito del ejemplo es mostrar lo difícil que resulta el mantenimiento de la aplicación al tratar de deducir que reglas están implantadas o que reglas se están ejecutando cuando se altera un campo de una tabla o bien cuando se realiza una inserción, actualización o borrado, ya que el código como se aprecia tiende a ser monolítico, y el lenguaje utilizado, en este caso SQL, es propietario y con una sintaxis y manejo complejo. Se observa además como las reglas se encuentran enterradas en código.

```

-----*/
/* Nombre del Proyecto : Base de datos Comerciales (ARGOS) */
-----*/
/* Nombre del modulo: ACCTRSP2.SQL */
/* Creator : AGO */
/* Role del modulo : Creación del Package de procedimientos */
/* almacenados de emisión del contrato */
/* Descripción de las modificaciones : */
/* Historico del modulo : */
/* <Versión> <Fecha> <Creator> <Descripción> */
/* 1.00 10/01/96 AGO Creación del modulo */
/* 1.01 30/01/97 AGO Eliminación delete de razones de rechazo */
/* Manejo del envío de los mensajes de delete */
/* 1.02 18/02/97 RFSSP_ModificarPapelEntCtr */
/* 1.03 Reemplazo de fecha_baja_serv por fecha_suspension_serv */
/* ya que se almacena en ARGOS y se necesita en SISA */
/* al emitirse el Contrato y sus Servicios */
/* 1.04 12/03/97 CTB Se agregó la fase EMO para poder enviar el */
/* mensaje de modificación del servicio a SISA */
/* 1.05 13/05/97 CTB Se agregó SP_EmitirValorCaractFija para que */
/* se emita la Acometida si está fija */
/* 1.06 20/05/97 CTB Modificación para que acepte servicios en VAL */
/* 1.07 04/06/97 CTB Cambio de la actualización de fecha de emisión */
/* antes de que obtenga los datos de la EAC (ACO) */
/* 1.08 16/06/97 CTB Borrado físico en la emisión de comprobantes */
/* 1.09 01/07/97 CRF Adición de SP_DeleteErrorValidacionCtr, */
/* SP_InsertarErrorValidacionCtr y SP_ActualizarErrorValidaCtr */
/* 1.10 05/08/97 CTB Emitir 7 caracteres de la referencia de comprobante */
/* 1.11 11/08/97 CTB Eliminar 1.10 */
/* 1.12 10/09/97 CRF Eliminar el parámetro vt_FechaValidacionCtr en */
/* SP_ActualizarErrorValidaCtr */
/* 1.13 15/10/97 CTB Agregar SP_CalcTarifaServicio */
/* 1.14 17/10/97 YKS Se añadió la fase y estado de avance RCV */
/* 1.15 04/11/97 CTB Modificar SP_CalcTarifaServicio ya que no tomaba */
/* en cuenta la cantidad para GI */
/* 1.16 21/11/97 YKS Modificar manejo de tarifas: */
/* Adición de SP_CalcTotalGIServ y */
/* SP_CalcTotalMortoComprob y se modificó */
/* SP_CalcTarifaServicio y el paso de parámetros */
/* al alta y modificación de contratos y servicios */
/* 1.17 03/12/97 YKS Modificar SP_CalcTotalGIServ, ahora va a ser */
/* llamado desde Visual Basic */
/* 1.18 09/12/97 CTB SP_CalcTotalGIServ, tomar sólo tipo mov. ALT */
/* 1.19 11/12/97 CTB SP_EmitirServiciosCtr si trae monto especial enviar */

```

```

/* la misma cantidad en la tarifa total de GI y/o rentas */
/* 1.20 13/01/98 CRF Corregir SP_CalcTotalGIServ, por cambio en la tabla */
/* Total_gastos_contrato */
/* 1.21 22/01/98 CRF Adición de la constante global gs_COD_ESTADO_EPP */
/* y modificación de SP_EmitirServiciosCtr, */
/* SP_EmitirValorCaractPuntaCtr, SP_EmitirValorCaractFija, */
/* SP_EmitirValorCaractServCtr y SP_EmitirPuntasCtr */
/* para que tomen en cuenta el estado del servicio EPP */
/* 23/01/98 CRF No borrar en la tabla Total_Gastos_contrato al */
/* momento de emitir un contrato. */
/* Llamar al proc. SP_CalcularGastosContrato */
/* desde SP_CalcTotalGIServ */
/* 1.22 14/03/98 CTB SP_CalcTotalGIServ si tiene monto especial no */
/* importa el que no tenga Tarifa asignada */
-----*/
/* Especificación del package ACCTR2 de manejo de la emisión de ctr */
/* */
/* Contiene las funciones publicas siguientes: */
/* . SP_EmitirContrato : emisión de un contrato */
/* . SP_ModificarPapelEntCtr */
/* . SFi_SelectValidacionCtr */
/* . SP_DeleteErrorValidacionCtr */
/* . SP_InsertarErrorValidacionCtr */
/* . SP_ActualizarErrorValidaCtr */
/* . SP_CalcTarifaServicio */
/* . SP_CalcTotalGIServ */
-----*/

```

CREATE OR REPLACE PACKAGE ACCTR2 AS

```

/* -- Procedimiento de emisión del contrato -- */
PROCEDURE SP_EmitirContrato
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE);

/* -- Función de recup. de los datos de un registro en Validacion_contrato -- */
FUNCTION SFi_SelectValidacionCtr
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE,
vt_Emitido IN Validacion_contrato.Es_emitido_ctr%TYPE,
rx_ValidCtr OUT Validacion_contrato%ROWTYPE
) RETURN INTEGER ;

/* -- Cálculo del total de gastos de instalación de los servicios del contrato en curso -- */
PROCEDURE SP_CalcTotalGIServ
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE
);

END ACCTR2 ; -- Fin de especificación del package
-----*/
/* Cuerpo del package ACCTR2 de emisión de los contratos */
-----*/
CREATE OR REPLACE PACKAGE BODY ACCTR2 AS
-----*/
/* -- CONSTANTES DEL PACKAGE */
-----*/
/* -- Codigos de regreso -- */
gi_ERR_OK CONSTANT INTEGER := 0 ;
gi_ERR_KO CONSTANT INTEGER := -1 ;
gi_ERR_NO_ENCONTRADO CONSTANT INTEGER := 1 ;

/* -- Booleanos -- */
gc_SI CONSTANT VARCHAR2(1) := 'S' ;
gc_NO CONSTANT VARCHAR2(1) := 'N' ;

```

```

/* -- Codigos de fases de servicio -- */
gs_COD_FASE_ALTA CONSTANT Fase.Cod_fase%TYPE := 'ALT';
gs_COD_FASE_RECHAZADO_RCH CONSTANT Fase.Cod_fase%TYPE := 'RCH';
gs_COD_FASE_RECHAZADO_RCV CONSTANT Fase.Cod_fase%TYPE := 'RCV';
gs_COD_FASE_SUSPENSION CONSTANT Fase.Cod_fase%TYPE := 'ESU';
gs_COD_FASE_VALIDACION CONSTANT Fase.Cod_fase%TYPE := 'VAL';
gs_COD_FASE_OPERACION CONSTANT Fase.Cod_fase%TYPE := 'OPE';
gs_COD_FASE_BAJA CONSTANT Fase.Cod_fase%TYPE := 'BAJ';
gs_COD_FASE_MODIFICACION CONSTANT Fase.Cod_fase%TYPE := 'EMO';
/* -- Codigos de estados de avance de servicio -- */
gs_COD_ESTADO_INICIAL CONSTANT Estado_avance_servicio.Cod_estado_avance_serv%TYPE := 'INI';
gs_COD_ESTADO_RECHAZADO_RCH CONSTANT Estado_avance_servicio.Cod_estado_avance_serv%TYPE := 'RCH';
gs_COD_ESTADO_RECHAZADO_RCV CONSTANT Estado_avance_servicio.Cod_estado_avance_serv%TYPE := 'RCV';
gs_COD_ESTADO_ADMIN CONSTANT Estado_avance_servicio.Cod_estado_avance_serv%TYPE := 'ADC';
gs_COD_ESTADO_EPP CONSTANT Estado_avance_servicio.Cod_estado_avance_serv%TYPE := 'EPP';

/* -- Codigos de tipos de movimientos de servicio -- */
gs_COD_TIPO_MOVIMIENTO_ALTA CONSTANT Tipo_movimiento_servicio.Cod_tipo_movimiento_serv%TYPE := 'ALT';
;
/* -- Codigos de papel de interlocutor -- */
gs_COD_PAPEL_INT_FIRMANTE CONSTANT Papel_interlocutor.Cod_papel_int%TYPE := 'RLE';
gs_COD_PAPEL_INT_RESP_INSTAL CONSTANT Papel_interlocutor.Cod_papel_int%TYPE := 'RIN';

/* -- Codigos de papel de actor -- */
gs_COD_PAPEL_ACTOR_FIRMANTE CONSTANT Papel_actor.Cod_papel_actor%TYPE := '2';
gs_COD_PAPEL_ACTOR_EJECUTIVO CONSTANT Papel_actor.Cod_papel_actor%TYPE := 'ECU';
gs_COD_PAPEL_ACTOR_AUTORIZ_CTR CONSTANT Papel_actor.Cod_papel_actor%TYPE := '5';

/* -- Codigos de papel de entidad -- */
gs_COD_PAPEL_ENTIDAD_ADC CONSTANT Papel_entidad.Cod_papel_entidad%TYPE := '2';
/* ----- */
/* -- DECLARACION DE LOS PROCEDIMIENTOS PRIVADOS -- */
/* ----- */

/* -- Emisión de los servicios del contrato -- */
PROCEDURE SP_EmitirServiciosCtr
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE,
vd_FechaUltimaEmision IN DATE,
vd_FechaFirmaCliente IN DATE
);
/* ----- */
/* -- PROCEDIMIENTOS PUBLICOS DEL PACKAG -- */
/* ----- */
/* ----- */
/* Nombre del procedimiento : SP_ModificarPapelEntCtr */
/* ----- */
/* Desc. del procedimiento : actualiza la entidad del contrato segun actor */
/* ----- */
/* Parámetros : */
/* vt_FolioCtr <--- Folio del contrato */
/* vt_FolioEntidad <--- Folio de la entidad */
/* ----- */
PROCEDURE SP_ModificarPapelEntCtr
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE,
vt_FolioEntidad IN Entidad.Folio_entidad%TYPE
) IS
BEGIN

UPDATE PAPEL_ENTIDAD_CONTRATO SET Folio_entidad = vt_FolioEntidad
WHERE Folio_ctr = vt_FolioCtr;

END SP_ModificarPapelEntCtr;

```

```

/* ----- */
/* Nombre del procedimiento : SP_EmitirContrato */
/* ----- */
/* Desc. del procedimiento : Emite un contrato */
/* ----- */
/* Parámetros : */
/* vt_FolioCtr <--- Folio del contrato */
/* ----- */
PROCEDURE SP_EmitirContrato
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE)
IS
/* -- Código de error -- */
li_CodErr INTEGER;
/* -- Variable de recup. de los datos del contrato -- */
lx_Contrato Contrato%ROWTYPE;
/* -- Variable de recup. del firmante cliente del contrato -- */
lt_FolioIntFirmaCtr Interlocutor.Folio_int%TYPE;
/* -- Variable de recup. del ejecutivo del contrato -- */
lt_FolioActorEcuActor.Folio_actor%TYPE;
/* -- Variable de recup. del gerente de venta del contrato -- */
lt_FolioActorGve Actor.Folio_actor%TYPE;
/* -- Variable de recup. de la entidad administradora -- */
lt_FolioEntidadAdc Entidad.Folio_entidad%TYPE;
/* -- Variable de recup. del actor quien autorizo -- */
lt_FolioActorAutorizacion Actor.Folio_actor%TYPE;
/* -- Variable de recup. del total de gi de los serv. del ctr. -- */
lt_GITotal Total_gastos_contrato.Total_gasto_instal_ctr%TYPE;
/* -- Variable de recup. de los datos de valid. de ctr. -- */
lx_ValidCtr Validacion_contrato%ROWTYPE;
/* -- Variable de recup. del monto de comprobantes del ctr. -- */
li_MontoComprob NUMBER;
BEGIN

/* -- Recup. de los datos que emitir -- */
/* ----- */
/* -- Recup. de los datos del contrato -- */
SELECT CTR.*
  INTO lx_Contrato
 FROM Contrato CTR
 WHERE CTR.Folio_ctr=vt_FolioCtr;

/* -- Recup. de la ultima validación de contrato -- */
/* -- emitida -- */
/* ----- */
li_CodErr := SF1_SelectValidacionCtr(vt_FolioCtr, gc_SI, lx_ValidCtr);
IF (li_CodErr = gi_ERR_NO_ENCONTRADO) THEN
  li_CodErr := gi_ERR_OK;
END IF;

/* -- Actualización estatus del Contrato -- */
/* ----- */
IF (li_CodErr = gi_ERR_OK) THEN

  /* -- Asignación de la fecha de emisión -- */
  lx_Contrato.Fecha_emision_ctr := SYSDATE;
  /* -- Contrato en validación; se cambia su fase -- */
  /* -- y se actualizan los indicadores -- */
  IF (lx_Contrato.Cod_fase = gs_COD_FASE_RECHAZADO_RCH OR lx_Contrato.Cod_fase =
gs_COD_FASE_RECHAZADO_RCV) THEN
    lx_Contrato.Cod_fase := gs_COD_FASE_ALTA;
  lx_Contrato.Fecha_fase_ctr := SYSDATE;
  ELSIF (lx_Contrato.Cod_fase = gs_COD_FASE_VALIDACION) THEN
    lx_Contrato.Cod_fase := gs_COD_FASE_OPERACION;
  lx_Contrato.Fecha_fase_ctr := SYSDATE;
  lx_Contrato.Es_instalacion_ok := gc_SI;
  lx_Contrato.Es_facturacion_ok := gc_SI;
  END IF;

```

```

/* -- Se actualiza el registro en la base de datos -- */
/* -- OBS: PONERLO EN UNA FUNCION -- */
UPDATE Contrato SET
Fecha_emision_ctr=lx_Contrato.Fecha_emision_ctr,
Cod_fase=lx_Contrato.Cod_fase,
Fecha_fase_ctr=lx_Contrato.Fecha_fase_ctr,
Es_instalacion_ok=lx_Contrato.Es_instalacion_ok,
Es_facturacion_ok=lx_Contrato.Es_facturacion_ok
WHERE Folio_ctr=vt_FolioCtr ;

/* -- Recup. de los datos ligados al contrato -- */
SELECT PIC_FIR.Folio_int,
PAC_ECU.Folio_actor,
ENT.Folio_actor_responsable,
PEC_ADC.Folio_entidad,
PAC_AUT.Folio_actor,
TGC.Total_gasto_instal_ctr
INTO li_FolioIntFirmaCtr,
li_FolioActorEcu,
li_FolioActorGve,
li_FolioEntidadAdc,
li_FolioActorAutorizacion,
li_GITotal
FROM Contrato CTR, Papel_interlocutor_contrato PIC_FIR,
Papel_actor_contrato PAC_ECU, Actor ACT, Entidad ENT,
Papel_entidad_contrato PEC_ADC, Papel_actor_contrato PAC_AUT,
Total_gastos_contrato TGC
WHERE CTR.Folio_ctr=vt_FolioCtr
AND PIC_FIR.Folio_ctr(+)=CTR.Folio_ctr
AND PIC_FIR.Cod_papel_int(+)=gs_COD_PAPEL_INT_FIRMANTE
AND PAC_ECU.Folio_ctr(+)=CTR.Folio_ctr
AND PAC_ECU.Cod_papel_actor(+)=gs_COD_PAPEL_ACTOR_EJECUTIVO
AND ACT.Folio_actor(+)=PAC_ECU.Folio_actor
AND ENT.Folio_entidad(+)=ACT.Folio_entidad
AND PEC_ADC.Folio_ctr(+)=CTR.Folio_ctr
AND PEC_ADC.Cod_papel_entidad(+)=gs_COD_PAPEL_ENTIDAD_ADC
AND PAC_AUT.Folio_ctr(+)=CTR.Folio_ctr
AND PAC_AUT.Cod_papel_actor(+)=
gs_COD_PAPEL_ACTOR_AUTORIZ_CTR
AND TGC.Folio_ctr = CTR.Folio_ctr;
END IF ;

/* -- Se calcula el monto total de los comprobantes del contrato -- */
/* ----- */
SP_CalcTotalMontoComprob(vt_FolioCtr,li_MontoComprob);

/* -- Se emite el contrato -- */
/* ----- */
IF (li_CodErr = gi_ERR_OK) THEN
IF (lx_ValidCtr.Fecha_validacion_ctr IS NULL) THEN
/* -- Envio msj. de alta de contrato -- */
IDOPECTR.SP_EnviarMsjAltaContrato
(lx_Contrato.Folio_ctr, lx_Contrato.Num_ctr,
lx_Contrato.Desc_ctr, lx_Contrato.Duracion_ctr,
lx_Contrato.Cod_tipo_duracion, lx_Contrato.Fecha_firma_cli_ctr,
lx_Contrato.Fecha_rescision_ctr,
lx_Contrato.Fecha_rev_general_ctr, lx_Contrato.Cod_fase,
lx_Contrato.Fecha_fase_ctr,
li_GITotal, li_MontoComprob,
lx_Contrato.Porcent_descuento_acceso_ctr,
lx_Contrato.Monto_descuento_acceso_ctr,
lx_Contrato.Porcent_descuento_renta_ctr,
lx_Contrato.Monto_descuento_renta_ctr,
lx_Contrato.Folio_ctr_padre, lx_Contrato.Cod_tipo_modif,
lx_Contrato.Folio_ctr_maestro, lx_Contrato.Folio_sitio,
li_FolioIntFirmaCtr, li_FolioActorEcu, li_FolioActorGve,
li_FolioEntidadAdc, lx_Contrato.Cod_etapa_pago_gi,

```

```

ix_Contrato.Ref_autorizacion_ctr, It_FolioActorAutorizacion);
ELSE

/* -- Envío msj. de modif. de contrato -- */
IDOPECTR.SP_EnviarMsjModifContrato
(ix_Contrato.Folio_ctr, ix_Contrato.Num_ctr,
ix_Contrato.Desc_ctr, ix_Contrato.Duracion_ctr,
ix_Contrato.Cod_tipo_duracion, ix_Contrato.Fecha_firma_cli_ctr,
ix_Contrato.Fecha_rescision_ctr,
ix_Contrato.Fecha_rev_general_ctr, ix_Contrato.Cod_fase,
ix_Contrato.Fecha_fase_ctr,
It_GITotal, It_MontoComprob,
ix_Contrato.Porcent_descuento_acceso_ctr,
ix_Contrato.Monto_descuento_acceso_ctr,
ix_Contrato.Porcent_descuento_renta_ctr,
ix_Contrato.Monto_descuento_renta_ctr,
ix_Contrato.Folio_ctr_padre, ix_Contrato.Cod_tipo_modif,
ix_Contrato.Folio_ctr_maestro, ix_Contrato.Folio_sitio,
It_FolioIntFirmaCtr, It_FolioActorEcu, It_FolioActorGve,
It_FolioEntidadAdc, ix_Contrato.Cod_etapa_pago_gi,
ix_Contrato.Ref_autorizacion_ctr, It_FolioActorAutorizacion);
END IF;
END IF ;
/* -- Emisión de los comprobantes del contrato -- */
/* ----- */
SP_EmitirComprobantesCtr(vt_FolioCtr,
ix_ValidCtr.Fecha_validacion_ctr);

/* -- Emisión de los servicios del contrato -- */
/* ----- */
SP_EmitirServiciosCtr(vt_FolioCtr, ix_ValidCtr.Fecha_validacion_ctr,
ix_Contrato.Fecha_firma_cli_ctr);

/* -- Contrato emitido: se actualiza el registro de -- */
/* -- validacion del contrato -- */
/* ----- */
IF (ii_CodErr = gi_ERR_OK) THEN

ii_CodErr := SFi_SelectValidacionCtr(vt_FolioCtr, gc_NO, ix_ValidCtr);
IF (ii_CodErr = gi_ERR_OK) THEN
/* -- Se actualiza la fecha de validacion del contrato -- */
UPDATE Validacion_contrato SET
Fecha_validacion_ctr=SYSDATE,
Es_emitido_ctr=gc_SI
WHERE Folio_validacion_ctr=ix_ValidCtr.Folio_validacion_ctr;
/* -- Se borra el registro de Total_gastos_contrato -- */
/* DELETE Total_gastos_contrato*/
/* WHERE Folio_ctr = vt_FolioCtr;*/
END IF ;

END IF ;

END SP_EmitirContrato ;

```

```

/* ----- */
/* Nombre de la función : SFi_SelectValidacionCtr */
/* ----- */
/* Desc. de la función : Recup. del ultimo registro de
/* Validacion_contrato relacionado a un contrato */
/* ----- */
/* Parámetros :
/* vt_FolioCtr <-- Folio del contrato */
/* vt_Emitido <-- Indica si se quiere el ultimo emitido (gc_SI),
/* no emitido (gc_NO) o no importa (NULL) */
/* rx_ValidCtr --> Datos de Validacion_Contrato regresados */
/* Valor de regreso :
/* . gi_ERR_NO_ENCONTRADO si no se encontró registro */
/* . gi_ERR_OK si se encontró registro */
/* . gi_ERR_KO si no */
/* ----- */

FUNCTION SFi_SelectValidacionCtr
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE,
vt_Emitido IN Validacion_contrato.Es_emitido_ctr%TYPE,
rx_ValidCtr OUT Validacion_contrato%ROWTYPE
) RETURN INTEGER IS
BEGIN
SELECT VCO.* INTO rx_ValidCtr FROM Validacion_contrato VCO
WHERE VCO.Folio_ctr=vt_FolioCtr
AND VCO.Fecha_creacion_validacion_ctr=
(SELECT MAX(VCO2.Fecha_creacion_validacion_ctr)
FROM Validacion_contrato VCO2
WHERE VCO2.Folio_ctr=vt_FolioCtr
AND VCO2.Es_Emitido_ctr=NVL(vt_Emitido,VCO2.Es_Emitido_ctr)
);
/* -- Hemos encontrado datos: regresamos "OK" -- */
RETURN (gi_ERR_OK);
EXCEPTION
WHEN NO_DATA_FOUND THEN
/* -- No registro encontrado -- */
RETURN (gi_ERR_NO_ENCONTRADO);
WHEN OTHERS THEN
/* -- Otro error -- */
RETURN (gi_ERR_KO);
END SFi_SelectValidacionCtr ;

/* ----- */
/* -- PROCEDIMIENTOS PRIVADOS DEL PACKAGE */
/* Nombre del procedimiento : SP_CalcTotalGIServ */
/* ----- */
/* Desc. del procedimiento : Calcula el total de gastos de instalación */
/* del contrato pasado como parámetro */
/* ----- */
/* Parámetros :
/* vi_MontoGITotal <-- Monto por gasto de instalación */
PROCEDURE SP_CalcTotalGIServ
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE
) IS
vi_MontoGI NUMBER:=0;
vi_MontoGITotal NUMBER:=0;
vi_TotalGastoInstalCtr Total_gastos_contrato.Total_gasto_instal_ctr%TYPE;
bandera BOOLEAN;
/* -- Cursor del recup de los datos de servicios -- */
CURSOR lu_Servicio IS
SELECT SER.*, CTR.Fecha_firma_cli_ctr
FROM Servicio SER,
Contrato_servicio CSE,
Contrato CTR
WHERE CSE.Folio_ctr=vt_FolioCtr
AND CSE.Cod_tipo_movimiento_serv=gs_COD_TIPO_MOVIMIENTO_ALTA
AND SER.Folio_serv=CSE.Folio_serv
AND CSE.Folio_ctr = CTR.Folio_ctr
AND SER.Fecha_baja_serv IS NULL;

```

```

/* -- Estructura de recup. de los datos del cursor -- */
lx_Servicio lu_Servicio%ROWTYPE ;
BEGIN
/* -- Se calcula el gasto de instalación de cada uno de los servicios -- */
/* ----- */
vi_MontoGITotal:=0;
FOR lx_Servicio IN lu_Servicio LOOP
vi_MontoGI:=0;
/* -- Si el servicio no tiene tarifa especial para Gastos de Instalación o para renta */
/* -- se calculan y se envían a SISA como tarifa especial */
IF lx_Servicio.Monto_especial_acceso_serv IS NULL THEN
IF lx_Servicio.Folio_Tarifa_Articulo IS NOT NULL THEN
BEGIN
/* -- Búsqueda de los GI del servicio -- */
SELECT Monto_tarif_gj_articulo INTO vi_MontoGI
FROM Tarifa_gasto_instal_articulo TGI
WHERE TGI.Folio_tarifa_articulo = lx_Servicio.Folio_Tarifa_Articulo
AND TGI.Fecha_ini_tarif_gj_articulo = (SELECT MAX(TGI2.Fecha_ini_tarif_gj_articulo)
FROM Tarifa_gasto_instal_articulo TGI2
WHERE TGI2.Fecha_ini_tarif_gj_articulo <= nvl(lx_Servicio.Fecha_firma_cli_ctr,sysdate)
AND TGI2.Folio_tarifa_articulo=TGI.Folio_tarifa_articulo);
vi_MontoGI:= vi_MontoGI * lx_Servicio.Cantidad_serv;
IF lx_Servicio.Porcent_descuento_acceso_serv IS NOT NULL THEN
vi_MontoGI:= vi_MontoGI * (1 - (lx_Servicio.Porcent_descuento_acceso_serv
/ 100));
END IF;
IF lx_Servicio.Monto_descuento_acceso_serv IS NOT NULL THEN
vi_MontoGI:= vi_MontoGI - lx_Servicio.Monto_descuento_acceso_serv;
END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
/* -- No se encontró registro -- */
vi_MontoGI:=0;
END;
END IF;
ELSE
vi_MontoGI:=lx_Servicio.Monto_especial_acceso_serv;
END IF;

vi_MontoGITotal := vi_MontoGITotal + vi_MontoGI;
END LOOP ;

BEGIN
SELECT Total_gasto_instal_ctr
INTO vi_TotalGastoInstalCtr
FROM Total_gastos_contrato
WHERE Folio_ctr = vt_FolioCtr;
EXCEPTION
WHEN NO_DATA_FOUND THEN
bandera := false;
END;

IF (bandera = false) THEN
INSERT INTO Total_gastos_contrato (Folio_ctr, Total_gasto_instal_ctr)
VALUES
(vt_FolioCtr,vi_MontoGITotal);
ELSE
UPDATE Total_gastos_contrato
SET Total_gasto_instal_ctr = vi_MontoGITotal
WHERE Folio_ctr = vt_FolioCtr;
END IF;

ACCTR3.SP_CalcularGastosContrato(vt_FolioCtr);

END SP_CalcTotalGIServ;

```

```

----- */
/* Nombre del procedimiento : SP_EmitirServiciosCtr                */
/*                                                                    */
/* Desc. del procedimiento : Emite los servicios de un contrato    */
/*                                                                    */
/* Parámetros :                                                    */
/* vt_FolioCtr <--- Folio del contrato                             */
/* vt_FechaUltimaEmision <--- Fecha de ultima emision del contrato */
/* vd_FechaFirmaCliente <--- Fecha de firma del cliente           */
----- */
PROCEDURE SP_EmitirServiciosCtr
(vt_FolioCtr IN Contrato.Folio_ctr%TYPE,
vd_FechaUltimaEmision IN DATE,
vd_FechaFirmaCliente IN DATE
) IS
/* -- Cursor del recup de los datos de servicios -- */
CURSOR lu_Servicio IS
SELECT SER.*,
       PIS_RIN.Folio_int AS Folio_int_resp_instal
FROM Servicio SER,
     Contrato_servicio CSE,
     Papel_interlocutor_servicio PIS_RIN
WHERE CSE.Folio_ctr=vt_FolioCtr
AND SER.Folio_serv=CSE.Folio_serv
AND SER.Cod_fase IN (gs_COD_FASE_ALTA, gs_COD_FASE_BAJA,
gs_COD_FASE_VALIDACION, gs_COD_FASE_SUSPENSION,
gs_COD_FASE_MODIFICACION)
AND NVL(SER.Cod_estado_avance_serv, gs_COD_ESTADO_INICIAL) IN
(gs_COD_ESTADO_INICIAL, gs_COD_ESTADO_RECHAZADO_RCH, gs_COD_ESTADO_RECHAZADO_RCV,
gs_COD_ESTADO_EPP)
AND PIS_RIN.Folio_serv(+) = SER.Folio_serv
AND PIS_RIN.Cod_papel_int(+) = gs_COD_PAPEL_INT_RESP_INSTAL;
/* -- Estructura de recup. de los datos del cursor -- */
lx_Servicio lu_Servicio%ROWTYPE;
li_MontoGastoInstal NUMBER:=0;
li_MontoRenta NUMBER:=0;
BEGIN
/* -- Bucle sobre los servicios del contrato -- */
/* ----- */
FOR lx_Servicio IN lu_Servicio LOOP
/* -- Historización del estatus del servicio -- */
/* ----- */
ACSER1.SP_HistorizarEstatusServicio(lx_Servicio.Folio_serv);

/* -- Actualización del estatus del servicio -- */
/* ----- */
/* -- Servicio en validación: pasa en operación -- */
/* ----- */
IF (lx_Servicio.Cod_fase=gs_COD_FASE_VALIDACION) THEN
lx_Servicio.Cod_fase := gs_COD_FASE_OPERACION;
lx_Servicio.Fecha_rev_fase_serv := SYSDATE;
lx_Servicio.Cod_estado_avance_serv := NULL;
lx_Servicio.Cod_tipo_movimiento_serv := NULL;
/* -- Servicio rechazado: cambia el estado a "Inicial" -- */
/* ----- */
ELSIF (lx_Servicio.Cod_estado_avance_serv=gs_COD_ESTADO_RECHAZADO_RCH OR
lx_Servicio.Cod_estado_avance_serv=gs_COD_ESTADO_RECHAZADO_RCV) THEN
lx_Servicio.Cod_estado_avance_serv := gs_COD_ESTADO_INICIAL;
END IF;
lx_Servicio.Fecha_rev_esta_avan_serv := SYSDATE;
/* -- Si el servicio no tiene tarifa especial para Gastos de Instalación o para renta */
/* -- se calculan y se envían a SISA */

```

```

IF (ix_Servicio.Monto_especial_acceso_serv IS NULL) OR
(ix_Servicio.Monto_especial_renta_serv IS NULL) THEN
SP_CalcTarifaServicio(ix_Servicio.Folio_serv,
ix_Servicio.Folio_Tarifa_Articulo,
vd_FechaFirmaCliente,
ix_Servicio.Porcent_descuento_acceso_serv,
ix_Servicio.Monto_descuento_acceso_serv,
ix_Servicio.Cantidad_serv,
ix_Servicio.Monto_especial_acceso_serv,
li_MontoGastoInstal,
ix_Servicio.Porcent_descuento_renta_serv,
ix_Servicio.Monto_descuento_renta_serv,
ix_Servicio.Monto_especial_renta_serv,
li_MontoRenta);
END IF ;
IF (ix_Servicio.Monto_especial_acceso_serv IS NOT NULL) THEN
li_MontoGastoInstal:=ix_Servicio.Monto_especial_acceso_serv;
END IF ;
IF (ix_Servicio.Monto_especial_renta_serv IS NOT NULL) THEN
li_MontoRenta:=ix_Servicio.Monto_especial_renta_serv;
END IF ;
/* -- Emision del servicio -- */
/* -- Se compara la fecha de creación del servicio -- */
/* -- con la fecha de ultima emision del contrato: -- */
/* -- 1. Se dio de alta despues de la primera emision: -- */
/* -- se envia un mensaje de alta de servicio -- */
IF (ix_Servicio.Cod_fase=gs_COD_FASE_OPERACION OR
ix_Servicio.Cod_fase=gs_COD_FASE_ALTA)
AND (vd_FechaUltimaEmision IS NULL OR
ix_Servicio.Fecha_creacion_serv>vd_FechaUltimaEmision) THEN
/* -- Envio de un mensaje de alta -- */
IDOPESER.SP_EnviarMsjAltaServicio
(ix_Servicio.Folio_serv, vt_FolioCtr,
ix_Servicio.Folio_articulo, ix_Servicio.Ref_usr_serv,
ix_Servicio.Desc_serv, ix_Servicio.Coment_serv,
li_MontoGastoInstal,li_MontoRenta,
ix_Servicio.Porcent_descuento_acceso_serv,
ix_Servicio.Monto_descuento_acceso_serv,
ix_Servicio.Monto_especial_acceso_serv,
ix_Servicio.Porcent_descuento_renta_serv,
ix_Servicio.Monto_descuento_renta_serv,
ix_Servicio.Monto_especial_renta_serv,
ix_Servicio.Fecha_creacion_serv,
ix_Servicio.Fecha_ctr_puesta_en_serv,
ix_Servicio.Fecha_rescision_serv, ix_Servicio.Fecha_plan_final_serv,
ix_Servicio.Fecha_rev_final_serv,
ix_Servicio.Fecha_real_puesta_en_serv,
ix_Servicio.Fecha_asign_fact_serv,
ix_Servicio.Fecha_suspension_serv, ix_Servicio.Fecha_reanudacion_serv,
ix_Servicio.Cod_fase, ix_Servicio.Fecha_rev_fase_serv,
ix_Servicio.Cod_estado_avance_serv,
ix_Servicio.Fecha_rev_esta_avan_serv,
ix_Servicio.Cod_tipo_movimiento_serv,
ix_Servicio.Folio_cuenta_fact, ix_Servicio.Num_tel_fact,
ix_Servicio.Folio_int_resp_instal, ix_Servicio.Folio_serv_ini);
/* -- 2. Se dio de alta antes de la primera emision: -- */
/* -- se envia un mensaje de modif. de servicio -- */
ELSE
IF (ix_Servicio.Fecha_baja_serv IS NULL) THEN
/* -- Envio de un mensaje de modif. -- */
IDOPESER.SP_EnviarMsjModifServicio
(ix_Servicio.Folio_serv, vt_FolioCtr,
ix_Servicio.Folio_articulo, ix_Servicio.Ref_usr_serv,
ix_Servicio.Desc_serv, ix_Servicio.Coment_serv,
li_MontoGastoInstal,li_MontoRenta,
ix_Servicio.Porcent_descuento_acceso_serv,
ix_Servicio.Monto_descuento_acceso_serv,

```

```

lx_Servicio.Monto_especial_acceso_serv,
lx_Servicio.Porcent_descuento_renta_serv,
lx_Servicio.Monto_descuento_renta_serv,
lx_Servicio.Monto_especial_renta_serv,
lx_Servicio.Fecha_creacion_serv,
lx_Servicio.Fecha_ctr_puesta_en_serv,
lx_Servicio.Fecha_rescision_serv, lx_Servicio.Fecha_plan_final_serv,
lx_Servicio.Fecha_rev_final_serv,
lx_Servicio.Fecha_real_puesta_en_serv,
lx_Servicio.Fecha_asign_fact_serv,
lx_Servicio.Fecha_suspension_serv, lx_Servicio.Fecha_reanudacion_serv,
lx_Servicio.Cod_fase, lx_Servicio.Fecha_rev_fase_serv,
lx_Servicio.Cod_estado_avance_serv,
lx_Servicio.Fecha_rev_esta_avan_serv,
lx_Servicio.Cod_tipo_movimiento_serv,
lx_Servicio.Folio_cuenta_fact, lx_Servicio.Num_tel_fact,
lx_Servicio.Folio_int_resp_instal, lx_Servicio.Folio_serv_ini);
ELSE
/* -- Envío de un mensaje de baja. -- */
IDOPESER.SP_EnviarMsjBajaServicio(lx_Servicio.Folio_serv);
/* -- Supresión de los servicios dados de baja -- */
DELETE FROM Servicio WHERE Folio_serv=lx_Servicio.Folio_serv;
END IF ;
END IF ;
END LOOP ;
/* ----- */
/* -- OBSERVACION IMPORTANTE: -- */
/* -- Las funciones de emisión de elementos ligados al -- */
/* -- servicio tienen que llamarse antes de actualizar -- */
/* -- los estatus de servicio en la base de datos para -- */
/* -- permitir de saber cuales son las puntas que emitir -- */
/* ----- */
/* -- Emisión de las características de servicios -- */
/* ----- */
SP_EmitirValorCaractServCtr(vt_FolioCtr, vd_FechaUltimaEmision);

/* -- Emisión de las puntas de servicio -- */
/* ----- */
SP_EmitirPuntasCtr(vt_FolioCtr, vd_FechaUltimaEmision);
/* -- Actualización del estatus de los servicios emitidos -- */
/* ----- */
UPDATE Servicio SER SET
Cod_fase=DECODE(Ser.Cod_fase,
gs_COD_FASE_VALIDACION, gs_COD_FASE_OPERACION,
SER.Cod_fase),
Fecha_rev_fase_serv=DECODE(Ser.Cod_fase,
gs_COD_FASE_VALIDACION, SYSDATE,
SER.Fecha_rev_fase_serv),
Cod_tipo_movimiento_serv=DECODE(Ser.Cod_fase,
gs_COD_FASE_VALIDACION, NULL,
SER.Cod_tipo_movimiento_serv),
Cod_estado_avance_serv=DECODE(Ser.Cod_fase,
gs_COD_FASE_VALIDACION, NULL,
gs_COD_ESTADO_ADMIN),
Fecha_rev_esta_avan_serv=SYSDATE
WHERE SER.Folio_serv IN
(SELECT CSE.Folio_serv FROM Contrato_servicio CSE
WHERE CSE.Folio_ctr=vt_FolioCtr)
AND SER.Cod_fase
IN (gs_COD_FASE_ALTA, gs_COD_FASE_BAJA,
gs_COD_FASE_VALIDACION, gs_COD_FASE_SUSPENSION)
AND NVL(SER.Cod_estado_avance_serv, gs_COD_ESTADO_INICIAL) IN
(gs_COD_ESTADO_INICIAL, gs_COD_ESTADO_RECHAZADO_RCH, gs_COD_ESTADO_RECHAZADO_RCV,
gs_COD_ESTADO_EPP);
END SP_EmitirServiciosCtr ;

```

## BIBLIOGRAFÍA

PARDINAS, Felipe, Metodología y Técnicas de Investigación en Ciencias Sociales. 32ª edición, México, Siglo Veintiuno Editores, 1991. 241 pp.

ZUBIZARRETA, F. Armando, La Aventura del Trabajo Intelectual como Estudiar e Investigar. E.U., Addison-Wesley Iberoamericana, 1986. 197 pp.

REYES, Ponce Agustín, Administración de Empresas Teoría y Práctica Primera Parte. México, Limusa, 1979. 189 pp.

CERTO, C. Samuel, Administración Moderna. Traducción de, Jaime Gómez Mont Araiza, México, Mc Graw-Hill, 1993, 628 pp.

STONER, A. F. James, Charles Wankel, Administración, Traducción de, Rosa Ma. Rosas Sánchez. México, Prentice Hall Hispanoamericana, 1989, 826 pp.

KAST, E. Fremont, James E. Rosenzweig, Administración en las Organizaciones. Enfoque de Sistemas y de Contingencias, Traducción de, Marco Antonio Malfaron Martínez. 2ª edición, México, Mc Graw-Hill, 1993, 754 pp.

TERRY, R. George, Stephen G. Franklin, Principios de Administración, Traducción de, Alfonso Vasseur Walls, México, Cía. Editorial Continental, S. A. de C. V., 1993, 747 pp.

BROWN, B. Warren, Dennis J. Moberg, Teoría de la Organización y la Administración. Enfoque Integral, México, Limusa, 1996, 708 pp.

FAIRLEY E. Richard, Ingeniería de Software. Traducción de, Antonio Sanchez Aguilar, México, Mc Graw Hill, 1990. 390 pp.

VILLE Sommer, Ingeniería de Software, Traducción de, Pedro Flores Suárez, México, Addison-Wesley Iberoamericana, 362 pp.

PRESSMAN S. Roger, Software Engineering. A practitioner's Approach. 2ª edición, Singapore, Mc Graw-Hill, 1998, 567 pp.

GHEZZI, Carlo., Mehdi Jazayeri y Dino Mandrioli, Fundamentals of Software Engineering. E.U., Prentice Hall, 1991. 573 pp.

BOOCH, Grady, Object Oriented Analysis and Design with Applications. 2ª. edición, Santa Clara, California, Addison-Wesley, 1994. 589 pp.

MEYER, Bertrand, Object – Oriented Software Construcción. New York, Prentice Hall, 1998.

YOURDON, Edward, Análisis Estructurado Moderno. Traducción de, Alexandra Taylor Armitage, México, Prentice Hall, 1993. 735 pp.

DATE, C. J., Introducción a los Sistemas de Bases de Datos. México, Addison-Wesley Iberoamericana, 1986, 648 pp.

LITTON, M. Gerry, Introduction to Database Management. A Practical Approach, E.U.A., Wm. C. Brown Publishers, 1987, 532 pp.

HANSEN, W. Gary, James V. Hansen, Data Base Management and Design, Englewd Cliffs, New Jersey, E.U.A., Prentice Hall, 1992, 548 pp.

DATE, C. J., The Systems Programing Series. An Introduction to Data Base Systems. 5ª edición. vol. I, E.U.A., Addison Wesley Publishing Company, 1990, 854 pp.

ROSS, G. Ross, The Business Rule Book. 2ª. edición, Boston, Massachusetts, Data Base Newsletter, 1997. 394 pp.

HAY, C. David, "Data Modeling by the Rules", En DataBase Newsletter, Boston, Massachusetts, XXIII (1995). pp. 1-5.

ODELL, J. James, "Specifying Business Rules", En DataBase Newsletter, Boston, Massachusetts, XXI (June 1993). pp. 1-5.

BROWN, G. Robert, "Adding Business Rule to Data Models". En DataBase Newsletter, Boston, Massachusetts, XX (1992). pp. 1-4.

GOTI, J. Carlos, "A Business Classification for Business Rules", En DataBase Newsletter, Boston, Massachusetts, XXII (1994). pp. 1-3.

Defining Business Rules – What are they really?, (Guide International Corporation, 1996).

Business Rules Automation: A proven Approach and Software Technology to Implement Computing Applications. Usoft Company.

VON, Halle Barbara, "The Business Rule Roadmap". En Database Programming & Design (1997). pp. 1-6.

HURWITZ, Judith, "When Rules Meet Development", En Hurwitz Consulting Group (1997). pp. 1-5.

KRAMER, I. Mitchell, "Business Rule Automating, Business Policies and Practices", En Patricia Seybolds Distributed Computing Monitor. pp. 1-35

VON, Halle Barbara y Jane Conkey, "Explaining the Business Connection", En Database Programming and Design (december 1996). pp. 1-6

VON, Halle Barbara y David Plotking, "The Art of letting Go", En Database Programming & Design (1997). pp.1-6.