

Índice general

. Agradecimientos	3
. Resumen	5
. Abstract	7
1. Introducción	9
1.1. Motivación	10
1.2. Planteamiento del problema	13
1.3. Objetivos	16
1.4. Descripción del trabajo	17
1.4.1. Contribuciones	17
1.4.2. Organización de la tesis	17
2. Antecedentes	19
2.1. Data Warehouse	19
2.1.1. Arquitectura	22
2.1.2. Componentes básicos	26
2.1.3. Mantenimiento(ETL)	30
2.2. Reglas de Análisis y PL/pgSQL	32
2.2.1. PL/pgSQL	33
2.2.2. Reglas de Análisis	37
2.2.3. Modelo de conocimiento	38

2.2.4. Modelo de ejecución	42
2.3. Sistema de Soporte a la Toma de Decisión	44
2.3.1. Definición	46
2.3.2. Clasificación	46
3. Sistemas de Apoyo a la Toma de Decisiones con Lógica Fuzzy y DWH Activas.	51
3.1. DSS	51
3.2. Lógica <i>fuzzy</i>	52
3.3. DWH	60
4. Implementación del DWH Activo.	63
4.1. Introducción.	63
4.2. Concepto	64
4.3. Diseño	65
4.4. Arquitectura	66
4.5. Ambiente de desarrollo	67
4.6. Diagrama de clases	70
4.7. Diseño de la Interfaz Gráfica.	71
5. Reglas ECA Fuzzy para modelar la Toma de Decisiones.	75
5.1. Funciones para modelar recursión ECA.	75
5.2. Reglas ECA Fuzzy	80
5.3. Modelado para la toma de decisiones	92
5.4. Comentarios finales	97
6. Caso de Estudio.	99
6.1. Tablas y Relaciones	99
6.2. Tablas de Hechos	100
6.3. Tablas Resumen y Conjuntos Fuzzy	101
6.4. Reglas ECA Fuzzy	103

ÍNDICE GENERAL	III
7. Aplicabilidad del Modelo	119
7.1. Entorno de simulación	119
7.2. Análisis de viabilidad	124
7.3. Análisis de resultados	125
8. Conclusiones	131
8.1. Resultados obtenidos	131
8.2. Trabajo futuro	132
. Bibliografía	133

“La dicha suprema de la vida es la convicción de que somos amados, amados por nosotros mismos; mejor dicho, amados a pesar de nosotros”.

Victor Hugo. Los miserables

Agradecimientos

Agradezco al Dr. Matías Alvarado por su valioso apoyo en la creación de este proyecto de tesis, así como el transmitir sus conocimientos y experiencias, que me fueron aportados tanto en lo profesional como en lo personal, muchas gracias, fue un placer caminar en este trayecto a su lado.

Resumen

En esta tesis se incide en el proceso de transformación de los datos operacionales, en información y conocimiento, así como en su automatización dentro de un sistema de soporte a la toma de decisiones (DSS por sus siglas en inglés). Se utilizan reglas evento – condición – acción (ECA) *fuzzy*: eventos generadores, condiciones a seguir y acciones a realizar; la extensión de las reglas ECA a un ámbito de lógica *fuzzy* es un aporte de la tesis. Partiendo de los datos en una Data Warehouse (DWH), las reglas ECA *fuzzy* conforman un módulo de apoyo para el análisis de circunstancias y la posterior toma de decisiones. Esta combinación de la DWH y el módulo de reglas para la toma de decisiones conforma un DWH Activo.

En el caso práctico que se presenta en la tesis, las reglas ECA *fuzzy* apoyan la toma de decisiones en el reabasto emergente de artículos en una tienda comercial. Una disrupción o ruptura de lo planeado en la existencia de artículos en una tienda, es un evento imprevisto que afecta el consumo normal del artículo, y plantea su reabastecimiento inmediato para no interrumpir el servicio al cliente. En la tesis se presenta un DSS asociado al control de inventario de una tienda, tal que de manera automática indica realizar la petición de artículos cuyo reabasto está fuera de lo habitual debido a una disrupción. Más aún, el sistema de soporte a la toma de decisiones sugiere los posibles proveedores para realizar el reabasto del artículo considerando las condiciones más adecuadas en términos de costo y beneficio para la tienda.

Abstract

This thesis takes on the process of transform the operational data, into information and knowledge. Alongside, it concerns on the automation of knowledge management fashioned on a decision support system (DSS). There are used ECA (Event-Condition-Action) Fuzzy rules: generated events, conditions to follow and actions to carry out; the extension of ECA rules to a fuzzy logic is a thesis' contribution. Starting off from the data in the Data Warehouse (DWH), the ECA Fuzzy rules conform a support module to analyze the circumstances and the later decision making. Combining a DWH with the module of decision making rules, sets an Active DWH.

In the practical case that appears in the thesis, the rules ECA Fuzzy support the decision making in the supply chain emergent of articles in a commercial store. A disruption or rupture of the plan in the article stock in a store, is an unexpected event affecting the normal consumption of article, and minds its immediate supply for not to interrupt the service to the client.

There is a DSS presented in the thesis associated to the inventory control of the store, so it automatically stock replenishment requisition of articles of which replenishment outside the habitual due to a disruption.

Also, the support system of to the decision making suggests the possible suppliers to make stock replenishment of the article considering the best cost and benefit condition for the store.

Capítulo 1

Introducción

Desde su aparición, a mediados de los años 70's, las BD's (y la teoría sobre BD) no se han detenido generando versiones de estas. Las primeras versiones de las BD's se centraron alrededor de un único repositorio que sirva a todos los propósitos orientados al procesamiento de la información (desde el transaccional, pasando por el procesamiento batch, hasta lo analítico). En la mayoría de los casos, el principal foco de las primeras BD's fueron los sistemas operacionales o transaccionales. En las últimas décadas, ha surgido una noción más sofisticada de las BD's. Por un lado, el objetivo de servir a las necesidades operacionales y por otro, cubrir las necesidades analíticas de la información, para así surgir a finales de los 80's el concepto de DWH para dar soporte al proceso de toma de decisiones gerenciales [6].

El modelo relacional tiene entre sus objetivos guardar la integridad de los datos obtenidos en los procesos transaccionales en línea (On-Line Transactional Processing, OLTP). Sin embargo, este modelo no corresponde con la forma en la que el usuario percibe el conocimiento de un negocio, en general y del conocimiento comercial, en particular.

Aunque los sistemas de gestión de BD relacionales, han sido muy beneficiosos para los usuarios, nunca han sido diseñados para proporcionar funciones potentes de síntesis, análisis y consolidación de los datos. Por ello, muchas empresas deciden introducir o desarrollar soluciones para la gestión relacionada con el cliente (Customer Relationship Management, CRM) [16] que les permiten automatizar la actividad comercial, dar soporte a campañas orientadas a segmentos de clientes y una serie de más funcionalidades que les brindan la información necesaria para construir indicadores

sobre la actividad comercial y características de la relación del cliente con la corporación, además de poder detectar necesidades de sus clientes.

La lógica *fuzzy* puede usarse para describir el mundo en el que vivimos, permite dar seguimiento al comportamiento humano de razonar, sacando conclusiones a partir de hechos observados.

La simulación del comportamiento humano es una de las principales investigaciones en la Inteligencia Artificial y por lo tanto sus técnicas pueden ser empleadas para simular el comportamiento social humano en particular en la iteración social y su comportamiento en un equipo de trabajo [23].

Este esfuerzo puede constituir un primer paso para comenzar a obtener información o mejorar la calidad de ésta y dar seguimiento a una parte importante de la operatividad de la empresa, que es la gestión comercial. Un segundo paso, consiste en la creación de un repositorio histórico de información cuyas unidades principales son el cliente, el producto y la organización.

El objetivo de conformar este repositorio es el de tener acceso a una visión histórica y sobre distintos aspectos de nuestros datos con el objetivo de crear indicadores de gestión y suministrar información para mejorar los procesos de marketing, de rentabilidad y control de riesgos, debido a una correcta toma de decisión.

Así como la lógica clásica integra la base de los sistemas expertos convencionales, la lógica *fuzzy* da forma a las bases de los sistemas expertos *fuzzy*. Además de tratar con la incertidumbre, los sistemas expertos pueden modelar razonamiento de sentido común, que es muy difícil para los sistemas convencionales [26].

Una regla ECA esta relacionada sobre el principio básico que cuando un cierto evento ocurre y una condición es especificada, entonces una acción puede ser ejecutada. En conclusión, se ilustra cómo el DWH complementado con reglas ECA *fuzzy* puede dar respuesta al reto actual de las corporaciones de obtener “actividad estratégica automática” dentro de su DWH para disrupciones dentro de esta, sin la iteración constante de un analista.

1.1. Motivación

Desde que se inició la era de la computación, las organizaciones han usado los datos desde sus sistemas operacionales para atender sus necesidades de información. Algunas proporcionan acceso directo a la información, contenida dentro de las aplicaciones operacionales. Otras, han extraído

los datos desde sus BD's operacionales para combinarlos de varias formas no estructuradas, en su intento por atender a los usuarios en sus necesidades de información.

Ambos métodos han evolucionado a través del tiempo y ahora las organizaciones manejan sus datos no limpios e inconsistentes, sobre las cuales, en la mayoría de las veces, se toman decisiones importantes.

La gestión administrativa reconoce que una manera de elevar su eficiencia está en hacer el mejor uso de los recursos de información, que ya existen dentro de la organización. Una forma es almacenando y tratando información imprecisa o incierta en donde la lógica *fuzzy* puede ayudar a deducir hechos importantes mediante la implementación de reglas ECA *fuzzy* dentro de un DWH.

Pero bien se sabe que un DWH no solo es una BD como tal, sino una metodología y tecnología, que valiéndose de éstas, sirven de soporte para la toma de decisiones, pero siempre tiene el soporte de una o varias BD's y por tál, es necesario valerse de todo método que mejore el uso de las BD's, para el óptimo manejo y análisis de los datos para la toma de decisiones estratégicas. En la tabla 1.1 se muestran las diferencias más importantes de usar una BD operacional y un DWH; en donde un DWH el tema de análisis esta enfocado a las necesidades del negocio en contraparte con una BD operacional que su enfoque es muy general y sus tablas son actuales a solo unos cuantos meses, en cambio un DWH incluye informacion detallada más resumida e histórica de hasta 10 años, ya que su finalidad es para dar soporte al análisis.

Tabla 1.1 Principales diferencias entre un BD.

Base de Datos Operacionales	Data Warehouse
Datos Operacionales	Datos del negocio para información
Orientado a la aplicación	Orientado al sujeto
Actual	Actual + histórico
Detallada	Detallada + resumida
Cambia continuamente	Estable

Un DWH tiene la capacidad de extraer información adicional a partir de los datos contenidos en las bases de datos y así poder definir reglas dentro de ellas. Puesto que gran parte de los fundamentos teóricos de estos sistemas es la lógica matemática, a menudo se las denomina Bases de Datos Lógicas.

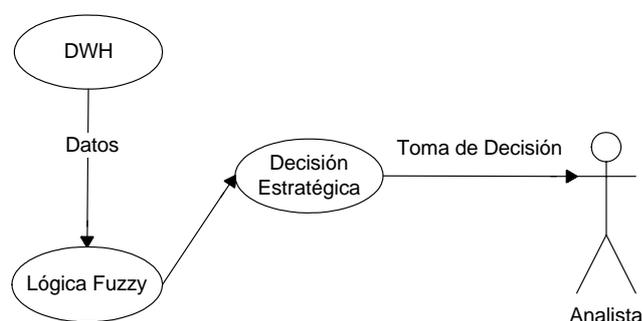


Figura 1.1: Contexto general del sistema para toma de decisiones estratégicas.

En donde las reglas son definidas en un lenguaje declarativo, permitiéndonos definir lo que queremos que el programa realice, apoyándonos del concepto de la lógica fuzzy y así deducir nuevos hechos en el DWH.

La estructura de estas bases de datos es muy similar a las de las bases de datos relacionales, pero adicionalmente se le añaden restricciones lógicas y reglas de composición con el fin de deducir información que originalmente no está contenida en la base de datos.

Reunir los elementos de datos apropiados, desde diversas fuentes de aplicación, en un ambiente integral centralizado, simplifica el problema de acceso a la información y en consecuencia, acelera el proceso de análisis y consultas brindando un menor tiempo de uso de la información para soluciones estratégicas.

Las aplicaciones para el soporte de decisiones basadas en lógica *fuzzy*, pueden hacer más factible la toma de decisión, agrupando datos para la asignación de valores lingüísticos, a veces difíciles de valorar pero que proporcionan aciertos válidos definidos a las necesidades y características de nuestras decisiones (Fig. 1.1). En donde podemos observar que un DWH en conjunto con la Lógica fuzzy puede dar soporte a la Toma de Decisiones y así aminorar el trabajo del analista.

Pero en la actualidad se ha de señalar que también se puede contar con dichas tecnologías para dar apoyo en aquellas actividades donde la aplicación del conocimiento, la experiencia y la propia coordinación juegan un papel fundamental en la eficiencia y productividad. Especialmente en los procesos para mejorar la calidad en la atención y satisfacción de los clientes, hecho que sin duda redundará en la eficacia comercial de las empresas.

Un ejemplo claro es la tienda Wal-Mart, tiene implementado un sistema para llevar un mejor control de sus artículos. Este sistema automatiza existencias reales de los artículos, pedidos, ventas por día, ventas por semanas, entre otras operaciones necesarias para la optimización de éstos y mejorar su servicio al cliente.

En la actualidad la mayor problemática que presenta es el “desabasto de artículos”, el cual aun no se ha automatizado.

1.2. Planteamiento del problema

En la actualidad las personas tienen más opciones para satisfacer sus necesidades diarias, en donde los artículos perecederos y no perecederos son de interés y el ¿donde? adquirirlos es una clave esencial para cubrir estas necesidades. Por ello mucha gente trata siempre de buscar establecimientos comerciales, que ofrezcan abastecer sus requerimientos sin afectar tanto a su economía. Es por eso que una tienda comercial siempre debe enfocarse a ofrecer mejor servicio y calidad al cliente, que puede consistir en tener siempre artículos en exhibición, ofrecer los mejores precios, tener las mejores ofertas, calidad en los artículos entre otras características necesarias para convencer a un cliente que esa tienda comercial es su mejor opción.

Muchas tiendas de autoservicio esporádicamente y en la actualidad presentan una problemática de prioridad, que es el “desabastecimiento de artículos”, ya que no han podido implementar este proceso en el sistema, debido a que en ocasiones presentan poca o nula exhibición de cierta mercancía puesta al público, ocasionada por clientes mayoristas que en períodos, sean o no de promoción, abarcan casi o totalmente de estos cuando tienen poco tiempo de resurtido (Fig. 1.2).

Cada artículo tiene programada fecha de pedido y no puede alterarse, porque se encuentra automatizado por el sistema, por lo tanto, tienen que esperar hasta la fecha ya establecida para volver a surtir, aquel producto o productos agotados.

Los proveedores entran en la problemática de abastecer a la tienda, porque también tienen programadas fechas para surtir a todos sus clientes.

En general, la cadena de supermercado tiene un sistema eficaz para ciertos procesos, pero poco eficiente para el caso del “desabastecimiento de artículos”, fuera del período de resurtido.

Se propone diseñar e implementar reglas ECA fuzzy para la automatización de la toma de decisiones estratégicas y así minimizar el trabajo de los analistas, creando con esto un DWH Activo

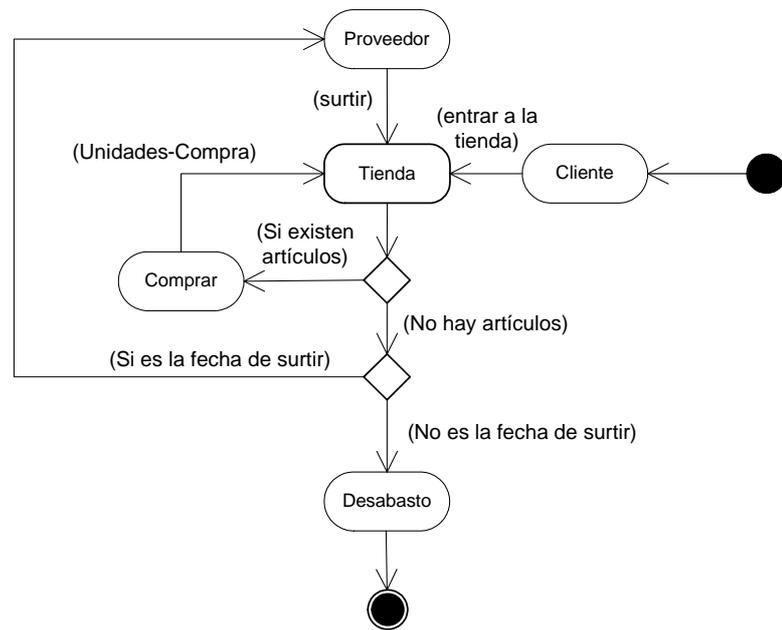


Figura 1.2: Diagrama de "Desabasto de artículos" en una tienda de autoservicio.

para el reabasto de artículos en tiempos optimos para tiendas de autoservicio.

Para crear la automatización nos apoyamos del principio de las reglas ECA ya que nos proporcionará las capacidades de actividad en el DWH, por su estructura evento, condición y acción, en donde al hacer uso de cursores para el desarrollo de funciones en nuestro warehouse, en un lenguaje procedimental (PL/pgsql) nos dará un soporte más robusto para realizar funciones y cálculos complejos dentro de nuestro DWH.

En la figura 1.3 se presenta el esquema del DWH Activo, compuesto por varios módulos:

El *contenedor de reglas ECA Fuzzy* el cual se encuentran contenidos dentro del administrador de PostgreSQL.

El *DWH pasivo* compuesto por los metadatos y la estructura del modelo multidimensional.

El *proceso ETL* (Extraction - Transform - Load, Extracción-Transformación-Carga) interfaz en el cual los datos son introducidos al DWH.

El *modulo de Acciones* que es un metadatos en el cual las decisiones son realizadas en ella, por la condición del DWH, siendo solo de lectura, y por tal en esta tabla se introducen los resultados.

En la figura 1.3 el analista, es el que declara las reglas de análisis (reglas ECA) en el lenguaje procedural (PL), ya que PL hace la conexión con los metadatos, los cuales se encuentran dentro del DWH y así detectar los eventos necesarios para realizar acciones fuera del DWH (OLTP) automatizando las tareas de rutinas de decisión operacionales e implementando reglas ECA *fuzzy* para la toma la decisión estratégica y abarcando la problemática de el “Desabasto de artículos” en una tienda de autoservicio, declaradas en un módulo dentro del DBMS PostgreSQL.

El simulador tiene la tarea de, generar datos de forma aleatoria, en donde finge el comportamiento de una tienda de autoservicio, aparentando compras en ciertos departamentos de la tienda, simulando dicha acción; los datos generados por este proceso son almacenados dentro del DWH para que valide los parámetros de las reglas ECA *Fuzzy* con los generados por el simulador obteniendo un comparativo de comportamientos cuantitativos(simulador) con los declarados en forma empírica(reglas ECAF).

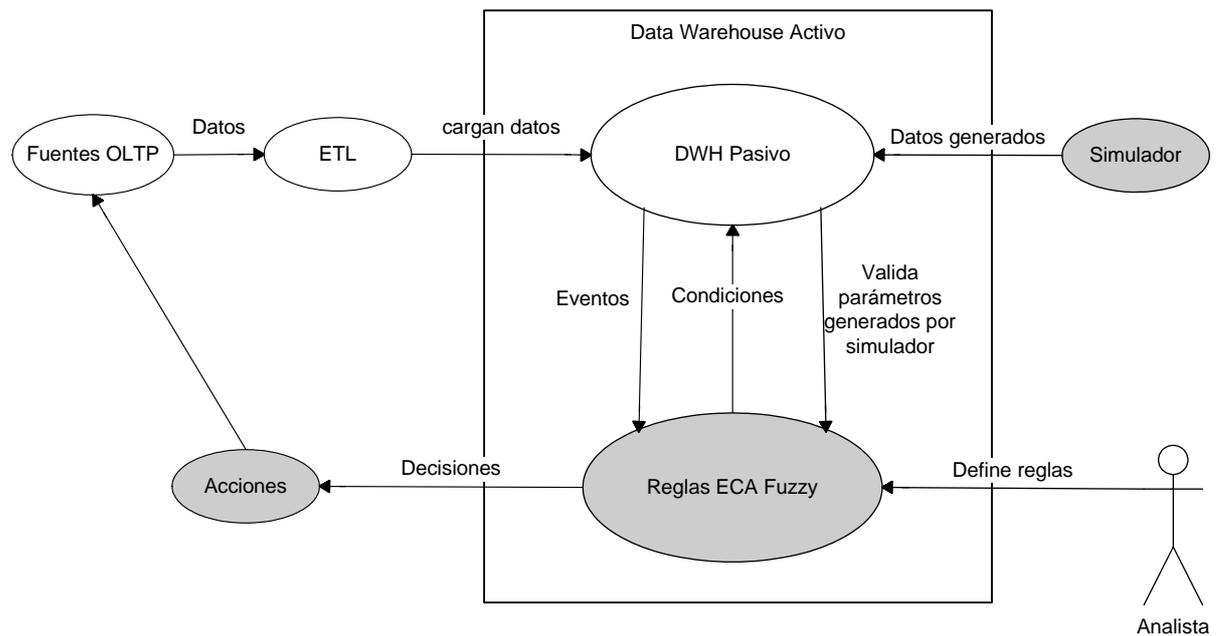


Figura 1.3: Esquema del sistema de apoyo para la toma de decisión empleando reglas ECAF.

1.3. Objetivos

Objetivo General

- Dar soporte a la toma de decisión para administrar la disrupción de “desabastecimiento de artículos” concerniente en determinar al proveedor que mejor se adecúe, para el reabasto de artículos, considerando costos de abastecimiento, ubicación de proveedores, tiempo de entrega, entre otras características, necesarias para la asignación estratégica .

Objetivos Particulares

- Crear un Sistema para la Toma de Decisiones Estratégicas (EDSS) basado en reglas ECA *fuzzy*.

- Proponer solución al problema de “desabasto de artículos” para el sistema de control de inventario en tiendas de autoservicio. En donde, suele presentarse consumo masivo de ciertos artículos que fueron semi o completamente agotados de exhibición, previo al período programado de abastecimiento.
- Desarrollar reglas ECA *fuzzy* para dar soporte a la toma de decisión (Base de Conocimiento).
- Implementar un generador de eventos tipo operacional que simule un desabastecimiento.
- Realizar pruebas de simulación.
- Análisis de resultado.

1.4. Descripción del trabajo

1.4.1. Contribuciones

Las contribuciones más importantes que ofrece la presente tesis de maestría son las siguientes:

- Automatizar la toma de decisiones estratégicas, para replanificar el sistema de abasto en una tienda de autoservicios haciendo uso de reglas ECA *fuzzy*, proponiendo una solución al problema de “Desabasto de Artículos”
- Apoyará los analistas, al obtener un DWH Activa para toma de decisiones estratégicas.
- La implementación de reglas *fuzzy* para el EDSS.
- El uso de cursores dentro del lenguaje PL/pgSQL para realizar funciones más complejas y usar menos memoria en cada ejecución.

1.4.2. Organización de la tesis

La escritura de la tesis se desarrollo de la siguiente manera:

En el capítulo 2 se presentan los fundamentos de los DWH, donde hablamos de las características principales, su arquitectura y sus propiedades. Así como tambien se mencionan los fundamentos

teóricos de la lógica *fuzzy* y de los conjuntos *fuzzy*. Se trata el tema de los Sistemas de Soporte a la Toma de Decisión y su clasificación.

En el capítulo 3 se mencionan los trabajos relacionados con DWH e incluso los DWH Activas comerciales y no comerciales que en la actualidad se encuentran en el mercado, también se mencionan trabajos relacionados con la lógica *fuzzy* para dar soporte a la toma de decisión.

En el capítulo 4 se presenta los componentes básicos para la construcción del DWH Activo, desde el diseño y la arquitectura. Se habla también del ambiente de desarrollo, así como de los diagramas de clases que se usaron para la creación del ambiente gráfico.

En el capítulo 5 mencionamos como se desarrollaron las reglas ECA *Fuzzy* para implementarlas en el DWH para dar soporte a este en las tomas de decisión y crea la automatización del sistema. Se detallan la creación de los conjuntos *fuzzy* como parte para la creación del modelo de conocimiento para la lógica *fuzzy* y así obtener reglas ECA *Fuzzy*.

En el capítulo 6 se explica el caso de estudio en el que la tesis fue desarrollada, detallando cada tabla contenida en el DWH y de los datos que fueron extraídas de estas, para crear los conjuntos *fuzzy* e implementarlas en las reglas de análisis.

La creación y objetivo de la implementación de un simulador se detalla en el capítulo 7, así como el comportamiento de este en las reglas ECA *Fuzzy*; se discute la viabilidad de nuestros datos como parámetros usados para la definición de nuestras reglas de análisis.

Por último en el capítulo 8 se explican los resultados obtenidos de este trabajo de investigación así como las limitaciones que presenta. Se enlistan los posibles trabajos futuros.

Capítulo 2

Antecedentes

2.1. Data Warehouse

El principal foco de las BD's fueron los sistemas operacionales o transaccionales. Pero hoy en día las empresas necesitan operar con muchos procesos OLTP para obtener la situación actual de su empresa. Para operar eficientemente con esos datos, se almacenaron en un sistema separado y específico, con el objetivo de servir a las necesidades operacionales y, por otro cubrir las necesidades analíticas de la información, creándose en 1992 y 1993 el DWH donde dos de los pioneros en este campo son Bill Inmon y Ralph Kimball. Inmon es universalmente conocido como el “padre del data warehouse”.

El entorno empresarial, le ha dado importancia al concepto DWH, implicándolo como la plataforma que concentra toda la información de interés para la organización, sus fuentes de información son tanto las BD's corporativas, como otras fuentes externas, por ejemplo, actualmente Internet se ha convertido en la fuente más importante de suministro de datos.

Los DWH se utilizan exclusivamente para hacer consultas en los datos históricos, de forma especial para que lleven a cabo estudios los analistas financieros, los analistas de mercado, etc. Actualmente, muchos DBMS se adaptan a este tipo de aplicación, incorporando herramientas como las siguientes:

- La creación y el mantenimiento de réplicas, con una cierta elaboración de los datos,
- La consolidación de datos de orígenes diferentes,

- La creación de estructuras físicas que soporten eficientemente el análisis multidimensional.

Definición

Existen numerosas definiciones de Data Warehouse, si bien la más conocida fue propuesta por Inmon [6], “Un DWH es una colección de datos orientados a temas, integrados, no-volátiles y variante en el tiempo, organizados para soportar necesidades empresariales”.

Richard Kimball define un Data Warehouse “Sistema que extrae, limpia, conforma y distribuye datos en un almacén de datos (en el original, data store) dimensional, y entonces soporta e implementa consultas y análisis con el propósito de toma de decisiones” [1].

Objetivos

Los datos del DWH, se obtienen por una replicación más o menos elaborada de la información que hay en las BD's, utilizadas en el trabajo cotidiano de la empresa, esta integración tiene varios objetivos:

1. Facilitar el acceso a la información de la compañía.
2. Aumentar la consistencia de la información.
3. Ser una fuente de información adaptativa y resistente.
4. Proteger la inversión en datos.
5. Ser la base para la toma de decisiones en la empresa.

Características

En general, las BD's que conforman el componente de almacenamiento físico del DWH se caracterizan por los siguientes aspectos:

- **Integradas:** deben constituir un conjunto de datos y metadatos perfectamente integrados con respecto al nombre de las variables, formatos de los distintos campos, medida de los atributos, codificación, etc. Para poder comprender mejor el proceso de integración se ha ejemplificado este con el formato para definir género en las personas; algunas fuentes OLTP pueden definirlo

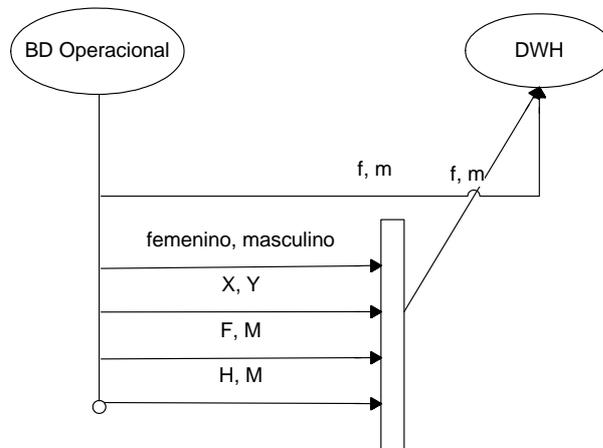


Figura 2.1: Integración de los datos desde fuentes OLTP hacia un DWH.

como m para especificar al género *masculino* y f para la *mujer*. Otros pueden referirse a este con un 0 para el género *masculino* y 1 para asignarlo a la *mujer*; de igual manera otras fuentes de datos suelen usar “ x ” *masculino*, “ y ” *femenino* respectivamente para los géneros mencionados anteriormente. Todo esto tiene que ser generalizado en un solo formato, para poder ser introducido al DWH. A todo esto se le conoce como *Integración* (Fig. 2.1).

- **Temáticas:** las BD's deben conformarse en base a temas específicos tales, como clientes, productos, campañas, etc., a diferencia de las BD's de los sistemas operacionales, más orientadas a procesos administrativos.
- **Históricas:** éste es un factor clave en la toma de decisiones, contar con información histórica para comparar datos en distintos períodos e identificar tendencias. El tiempo debe estar en todos y cada uno de los registros del DWH, de manera que, cuando un dato entra en el DWH se sepa en qué momento tenía ese valor.
- **No Volátiles:** la información una vez incorporada al DWH debe mantenerse, en general, invariable, cargándose una vez cada cierto tiempo y no permitiendo actualizaciones de los datos (Tabla 2.1).

Tabla 2.1 Comparación BD operacional y DWH, aspecto volatilidad.

BD Operacional	Data Warehouse
Horizonte de tiempo: 60-90 días	Horizonte de tiempo: 3-10 años
Los datos pueden ser actualizados	Datos solo para consulta

En la comparación anterior (tabla 2.1) entre una BD operacional y un DWH, se observa que una BD tiende a mantener en ella datos de entre 60 y 90 días mayormente, en contraparte con un DWH que su horizonte de almacenamiento de estos es de 5 a 10 años para consultas y referencias, ya que también incluye datos actuales y estos los llamados históricos (5 a 10 años).

Otra diferencia importante es que los datos no pueden ser modificados ya que son, solo para consultas en un DWH, en cambio en una BD estos pueden sufrir cualquier tipo de agregada, e incluso ser borrados.

2.1.1. Arquitectura

Una Arquitectura de un DWH es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, que existe para los usuarios finales que disponen de una computadora dentro de la empresa (Fig. ??).

La arquitectura se constituye de un número de partes interconectadas:

- Base de datos operacional / Nivel de base de datos externo
- Nivel de acceso a la información
- Nivel de acceso a los datos
- Nivel de directorio de datos (Metadata)
- Nivel de gestión de proceso
- Nivel de mensaje de la aplicación
- Nivel de data warehouse
- Nivel de organización de datos.

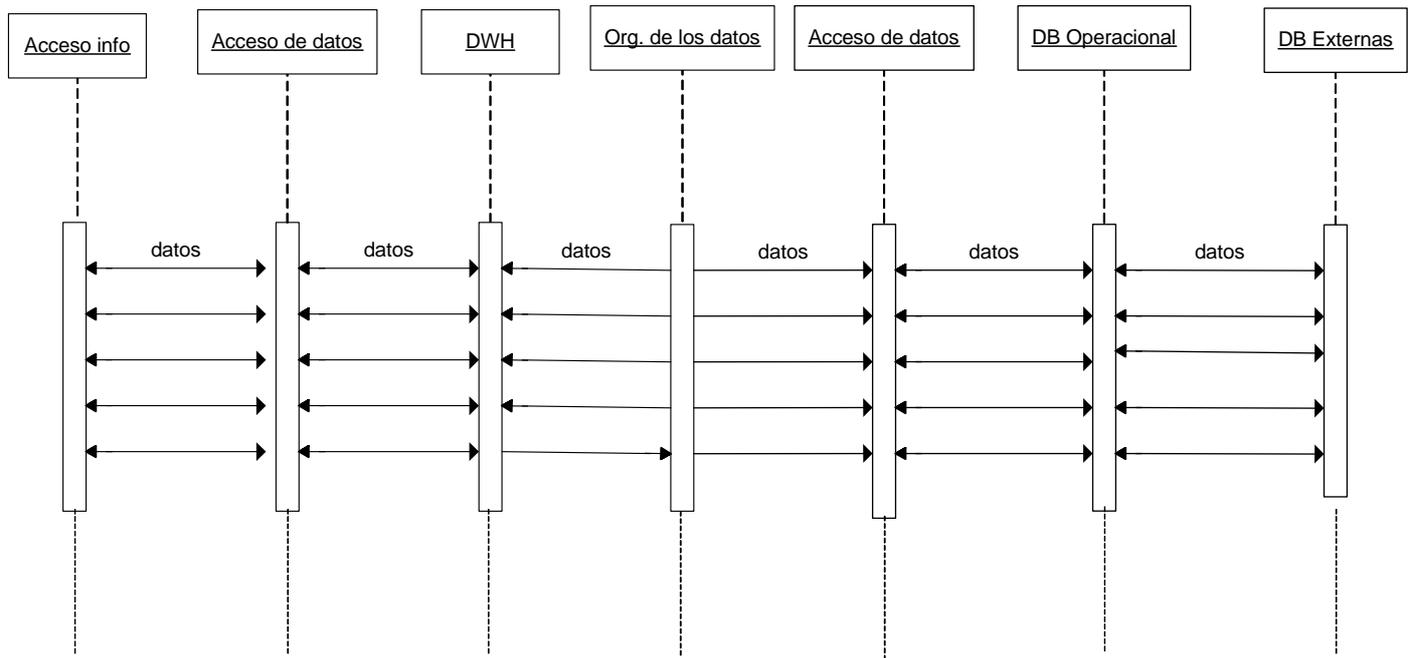


Figura 2.2: Secuencia de datos en un DWH.

En la figura 2.2 se muestra los niveles de datos en un DWH y como estos interactúan con los diferentes niveles según su acceso por estos.

A continuación se detalla cada nivel según su acceso de datos:

Base de datos operacional / Nivel de base de datos externo Los sistemas operacionales procesan datos para apoyar las necesidades operacionales críticas. Para hacer eso, se han creado las BD's operacionales históricas que proveen una estructura de procesamiento eficiente, para un número relativamente pequeño de transacciones comerciales bien definidas.

Sin embargo, a causa del enfoque limitado de los sistemas operacionales, las BD's diseñadas para soportar estos sistemas, tienen dificultad al acceder a los datos para otra gestión o propósitos informáticos [10].

Ciertamente, la meta del data warehousing es liberar la información que es almacenada en BD's operacionales y combinarla con la información desde otra fuente de datos, generalmente externa.

Cada vez más, las organizaciones grandes adquieren datos adicionales desde BD's externas como internet que también provee el acceso a más recursos de datos todos los días.

Nivel de acceso a la información Es el nivel del que el usuario final se encarga directamente. En particular, representa las herramientas que el usuario final normalmente usa día a día. Por ejemplo: Excel, Lotus 1-2-3, Access, etc.

Este nivel también incluye el hardware y software involucrados en mostrar información en pantalla y emitir reportes de impresión, hojas de cálculo, gráficos y diagramas para el análisis y presentación.

Actualmente, existen herramientas más sofisticadas para manipular, analizar y presentar los datos, sin embargo, hay problemas significativos al tratar de convertir los datos tal como han sido recolectados y que se encuentran contenidos en los sistemas operacionales en información fácil y transparente para las herramientas de los usuarios finales(ETL). Una de las claves para esto es encontrar un lenguaje de datos común que puede usarse a través de toda la empresa (integración) [10].

Nivel de acceso a los datos Está involucrado con el nivel de acceso a la información para interactuar en el nivel operacional. No solamente conecta DBMS diferentes y sistemas de archivos

sobre el mismo hardware, sino también protocolos de red. Una de las claves de una estrategia data warehousing es proveer a los usuarios finales con “acceso a datos universales”.

El acceso a los datos universales significa que, teóricamente por lo menos, los usuarios finales sin tener en cuenta la herramienta de acceso a la información o ubicación, deberían ser capaces de acceder a cualquier o a todos los datos en la empresa que es necesaria para ellos, para hacer su trabajo. El nivel de acceso a los datos es responsable de la interfaz entre las herramientas de acceso a la información y las BD's operacionales. En algunos casos, esto es todo lo que un usuario final necesita.

Nivel de Directorio de Datos (Metadata) A fin de proveer el acceso a los datos universales, es absolutamente necesario mantener alguna forma de directorio de datos o repositorio de la información metadata. La metadata es la información alrededor de los datos dentro de la empresa.

A fin de tener un depósito totalmente funcional, es necesario tener una variedad de metadatos disponibles, información sobre las vistas de datos de los usuarios finales e información sobre las BD's operacionales. Idealmente, los usuarios finales deberían de acceder a los datos desde el data warehouse (o desde las BD's operacionales), sin tener que conocer dónde residen los datos o la forma en que se han almacenado [10].

Nivel de Gestión de Procesos Éste tiene que ver con la programación de diversas tareas que deben realizarse para construir y mantener el DWH y la información del directorio de datos. Este nivel puede depender del alto nivel de control de trabajo para muchos procesos (procedimientos) que deben ocurrir para mantener el DWH actualizado.

Nivel de Mensaje de la Aplicación Tiene que ver con el transporte de información alrededor de la red de la empresa. El mensaje de aplicación se refiere también como “subproducto”, pero puede involucrar sólo protocolos de red. Puede usarse por ejemplo, para aislar aplicaciones operacionales o estratégicas a partir del formato de datos exacto, recolectar transacciones o los mensajes y entregarlos a una ubicación segura en un tiempo seguro.

Nivel Data Warehouse (Físico) En el DWH (núcleo) es en donde se obtienen los datos usados principalmente para usos estratégicos. En algunos casos, uno puede pensar del data ware-

house simplemente como una vista lógica o virtual de datos. En muchos ejemplos, el DWH puede no involucrar almacenamiento de datos.

En un DWH físico, se almacenan copias de los datos operacionales y/o externos, que son almacenados realmente en una forma que es fácil de acceder y es altamente flexible. Cada vez más, los data warehouses son almacenados sobre plataformas cliente/servidor, pero por lo general se almacenan sobre mainframes.

Nivel de Organización de Datos El componente final de la arquitectura DWH es la organización de los datos. Se llama también gestión de copia o réplica, pero de hecho, incluye todos los procesos necesarios como seleccionar, editar, resumir, combinar y cargar datos en el depósito y acceder a la información desde BD's operacionales y/o externas.

La organización de datos involucra con frecuencia una programación compleja, pero cada vez más, están creándose las herramientas data warehousing para ayudar en este proceso. Involucra también programas de análisis de calidad de datos y filtros que identifican modelos y estructura de datos dentro de la data operacional existente [10].

2.1.2. Componentes básicos

A la hora de realizar un modelo dimensional del negocio se hace necesaria la presentación de los siguientes componentes:

- Dimensiones
- Atributos
- Elementos
- Relaciones
- Jerarquías
- Tablas de Hechos o Indicadores.

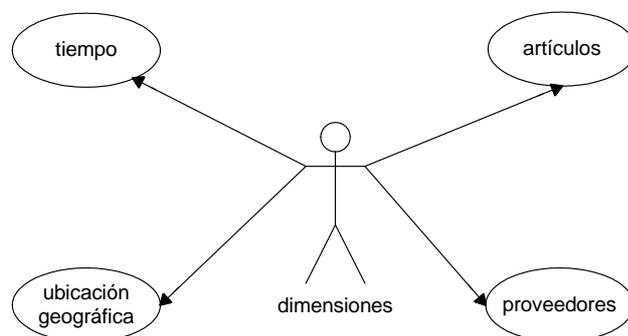


Figura 2.3: Ejemplo de dimensiones.

Dimensiones Son las áreas temáticas, líneas del negocio o sujetos del negocio. Las mismas proveen un método general para organizar la información corporativa. Definidas como un grupo de uno o más atributos, separados y distintos uno de otros (es decir, que no se comparten atributos), según el esquema de implementación elegido, puede ser que no se encuentren explícitamente en el DWH, sino que se presenten sólo como un recurso conceptual o lógico que ayuda a mostrar múltiples perspectivas de los datos permitiendo realizar análisis por diferentes dimensiones o incluso cruzando información entre distintas dimensiones.

En el ejemplo de dimensiones (Fig. 2.3) podemos ver que el tiempo, los artículos en la tienda, los proveedores y la ubicación geográfica de las tiendas, dan la dimensión básica para obtener una tabla de hechos.

Atributos Los atributos son una agrupación de elementos o ítems dentro de una dimensión. Representan categorías o clases de elementos que tienen el mismo nivel lógico dentro de una dimensión donde todos los elementos de un atributo se relacionan con otros atributos de la dimensión de la misma forma. La finalidad de los atributos es ver la información de cada dimensión a diferentes niveles de detalle y agrupar los datos para ser analizados.

Por ejemplo, ítem de un producto; los ítems pueden tener algunos atributos característicos como ser el color y el tamaño. En un modelo dimensional, ítem, color y tamaño son todos tratados como atributos de la dimensión producto.

En el ejemplo (Fig. 2.4) se puede ver como la dimensión tiempo puede desglosarse en varios

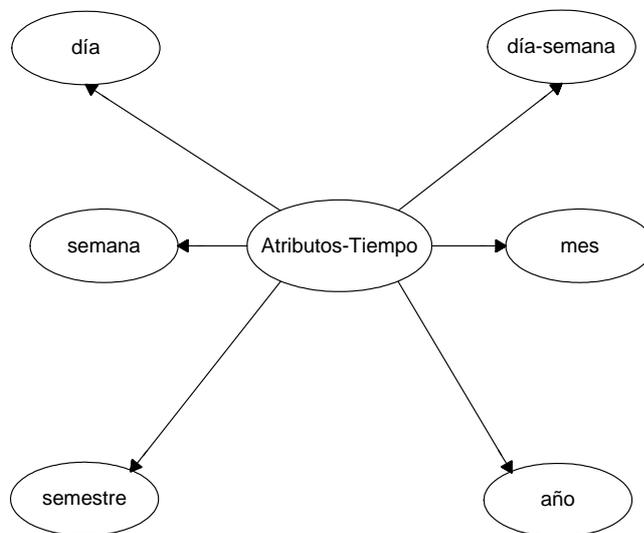


Figura 2.4: Ejemplo de atributos tiempo.

atributos (día, semana, día de la semana, mes, año, etc.) que en conjunto conforman los calificadores de una dimensión.

Elementos Son las instancias o valores de los atributos que, como componentes atómicos del modelo, permiten clasificar el rendimiento del negocio. Es importante aclarar que si bien no forman parte indispensable del modelado multidimensional es aconsejable su incorporación para un mayor entendimiento del modelo en etapas tempranas del ciclo de vida.

Relaciones Son asociaciones lógicas de atributos dentro de una jerarquía definida por las instancias de los atributos y transitivas dentro de una jerarquía.

Es importante aclarar que los atributos dentro de una dimensión están directamente relacionados uno con otros a través de los diferentes tipos de relaciones. En cambio, los atributos en diferentes dimensiones están relacionados uno con otros a través de los indicadores o variables del negocio definidas como intersección de las dimensiones (Fig. 2.5).

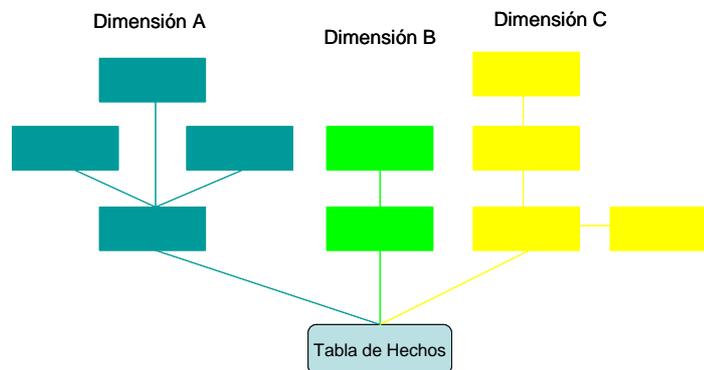


Figura 2.5: Esquema de un modelo dimensional.

Jerarquías Representadas por un ordenamiento lógico dentro de la dimensión, se encuentran formadas por los diferentes tipos de relaciones entre los atributos de una misma dimensión. Pueden existir múltiples jerarquías dentro de una dimensión pero siempre es posible identificar una jerarquía principal o columna vertebral de la dimensión y jerarquías secundarias o descriptivas compuestas por atributos característicos definidos desde la jerarquía principal.

Indicadores o Tabla de hechos Son las variables o métricas que ayudarán a medir la representación del negocio. Existen dos tipos de indicadores: básicos y derivados.

Los indicadores básicos, primitivos o crudos existen físicamente en el warehouse junto a los atributos que los caracterizan, pueden venir de diferentes sistemas fuentes y tener distintos niveles de granularidad o agregación. Por ejemplo, la variable **Venta(\$)** es llevada diariamente mientras que el indicador **Unidades en Existencias o Inventario** es seguido semanalmente. Por otro lado, los indicadores derivados o métricas calculadas se construyen a partir de los indicadores base y pueden o no estar almacenados físicamente en el DWH. Un ejemplo clásico de métrica derivada es Margen de Ganancia, la cual se define como la resta entre Precio y Costo (ambos indicadores básicos).

En la figura 2.6 se puede ver que los datos derivados de las dimensiones dan origen a una tabla general que concentra los datos necesarios para poder obtener un análisis específico según los requerimientos, por ejemplo en una refaccionaria se quiere obtener el estado actual de las ventas del

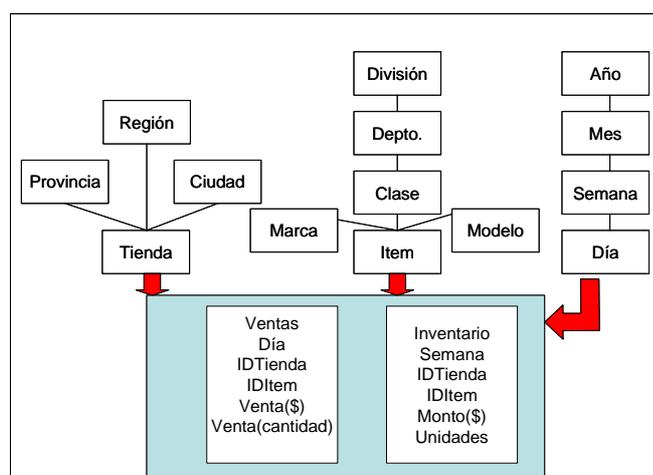


Figura 2.6: Diagrama de composición de la tabla de hechos o indicadores.

negocio, obteniendo los datos necesarios de las dimensiones *Ubicación*, *Tiempo* y *Artículos* hacia una tabla de hechos que es la que monitorea el comportamiento de las ventas.

2.1.3. Mantenimiento(ETL)

Los datos recopilados en las diferentes fuentes de almacenamiento son considerados datos “sucios” por contener muchas de las veces información innecesaria para el análisis dentro del DWH y por tener desigualdad de criterio para describir cierto dato.

La limpieza de los datos “sucios” es un proceso multifacético y complejo que consiste en analizar los datos corporativos para descubrir inexactitudes, anomalías y transformar los datos para asegurar que sean precisos y coherentes. Asegurar la integridad referencial, le proporciona al DWH la capacidad de identificar correctamente al instante, cada objeto del negocio, tales como un producto, un cliente o un empleado.

Los Sistemas ETL realizan las funciones de *extracción de las fuentes de datos* (transaccionales o externas), *transformación* (limpieza, consolidación,...etc.) y la *carga de los datos* una vez que ya hayan pasado por los dos anteriores pasos, para así almacenarse en el DWH.

En la figura 2.7 se muestra la secuencia que el proceso ETL ejecuta para realizar el mantenimiento del DWH, en el cual el primer proceso es la extracción de los datos necesarios para el

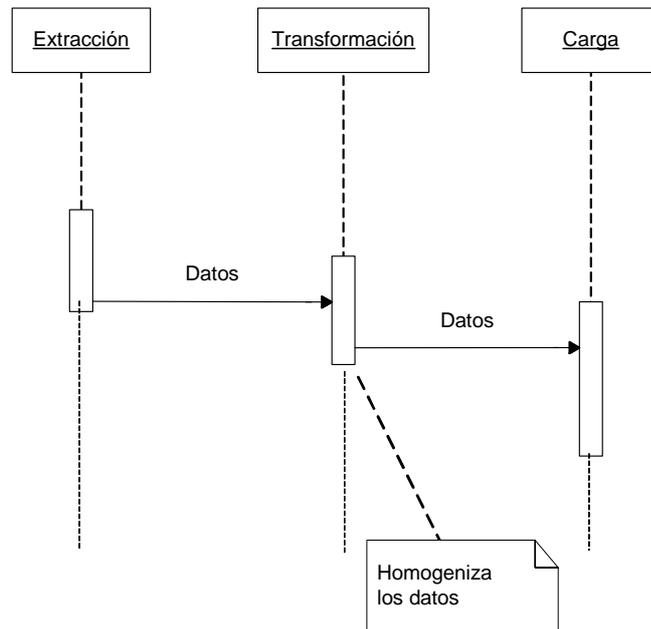


Figura 2.7: Secuencia del proceso ETL.

análisis de la información, una vez hecho la extracción el proceso siguiente es la transformación de los datos en datos basados según el formato estandarizado dentro del DWH y por último se realiza la carga de los datos en el DWH.

Los pasos para el proceso ETL consiste en:

- Validar los datos que usa la aplicación del DWH para realizar las consultas de prueba.
- Producir la metadata, una descripción del tipo de datos, formato y el significado relacionado al negocio de cada campo.
- La documentación del proceso completo, para que se pueda ampliar, modificar y arreglar los datos en el futuro con más facilidad.
- La carga de los datos en el warehouse

Existen varias razones que justifican la creación del DWH para obtener la información necesaria

en los procesos de gestión comercial, en lugar de obtener esa información directamente de las BD's de las aplicaciones operacionales:

Rendimiento: se tarda mucho menos en acceder a los datos del repositorio del DWH que en hacer una consulta a varias BD's distintas. Además hacer consultas complicadas a las BD's de los sistemas operacionales puede empeorar el tiempo de respuesta de estos sistemas para otros usuarios.

Múltiples orígenes de datos: combinar los datos de distintas fuentes suele ser una tarea bastante complicada para las personas encargadas de tomar decisiones con esa información. Normalmente hay que homogenizar los datos de una forma u otra. Por ejemplo, es probable que no se utilicen los mismos criterios de almacenamiento (nombres de las entidades, atributos considerados, tipos, etc.) en las BD's de distintos departamentos. Sin embargo, en el DWH los datos se homogenizan durante el proceso de carga (ETL) todo esto para obtener datos específicos y poder realizar un adecuado análisis para la toma de decisión.

En general el DWH es una herramienta útil para el análisis específico de datos y anudado a reglas de análisis pueden ofrecer apoyo a la toma de decisiones al implementarle dichas reglas.

2.2. Reglas de Análisis y PL/pgSQL

PostgreSQL y PL/pgSQL PL/pgSQL es un lenguaje de bloques estructurado, imperativo (todas las variables tienen que ser declaradas). PostgreSQL añade dentro de sus características el uso de lenguajes procedimentales (PL/pgSQL) para consultas más complejas. PL/SQL de Oracle tiene muchas más características que su homónimo de PostgreSQL, pero PL/pgSQL permite una gran funcionalidad y es mejorado constantemente.

Además por ser software libre es portable ofreciendo la migración de la BD. El proceso de migración consiste básicamente en el cambio de plataforma de los servicios, aplicaciones y desarrollos, de una empresa, institución o uso personal.

En lugar de ejecutar una consulta completa de inmediato, es posible definir un cursor que encapsule la consulta, para después leer los resultados de la consulta recuperando solo algunos renglones a la vez. Una razón para proceder de esta manera es evitar el consumo de la memoria

cuando el resultado contiene una gran cantidad de renglones. Un uso más interesante es regresar una referencia a un cursor que ha creado una función. Esto proporciona una manera eficiente de regresar, desde las funciones, conjuntos grandes de renglones.

Ventajas de usar PL/pgSQL

- Mayor rendimiento.

PL/pgSQL puede agrupar una serie de consultas dentro del servidor de BD, teniendo así la potencia de un lenguaje procedural y la sencillez del Lenguaje de consulta estructurado (Structured Query Language, SQL), pero ahorrando una gran cantidad de tiempo porque no tiene la sobrecarga de una comunicación cliente/servidor. Esto puede redundar en un considerable aumento del rendimiento.

- Soporta sentencias SQL.

PL/pgSQL añade a la potencia de un lenguaje procedural la sencillez de SQL. Con PL/pgSQL se puede usar todos los tipos de datos, columnas, operadores y funciones de SQL.

- Portabilidad.

Debido a que las funciones PL/pgSQL se ejecutan dentro de PostgreSQL, estas pueden ser usadas en cualquier plataforma donde PostgreSQL pueda ser ejecutado. Así podrá reusar el código y reducir costos de desarrollo. Por ejemplo en UNIX, MAC'S, Windows.

2.2.1. PL/pgSQL

PostgreSQL tiene algunas interfaces cliente como Perl, Tcl, Python y C, así como Lenguajes Procedurales (PL).

PostgreSQL proporciona tres tipos de funciones:

- funciones de lenguaje de consultas (funciones escritas en SQL)
- funciones de lenguaje procedural (funciones escritas en, por ejemplo, PLTCL o PLSQL)

- funciones de lenguaje de programación (funciones escritas en un lenguaje de programación compilado tales como C) también es posible llamar a funciones C como acciones trigger.

Habitualmente los SGBD se dividen en varias partes:

1. un *servidor*, que da acceso al espacio estructurado como una BD usando una variedad de interfaces diferentes, aparte de otra serie de servicios, como autenticación, autorización, etc.,
2. un *cliente*, que permite al usuario o a los programas acceder a esos datos.

Postgres utiliza el modelo cliente/servidor tal como se muestra en la figura 2.8. El servidor es el que tiene la administración de los requerimientos del cliente, donde el cliente hace las peticiones al servidor y espera respuesta de este con dicha petición, por ejemplo puede el cliente hacer la petición de una consulta para obtener los datos de alguna tabla de la BD, en donde se hace uso de un SELECT, inmediatamente es enviado al servidor que será el encargado de proporcionarle la respuesta a dicha consulta.

Estructura de PL/pgSQL

PL/pgSQL es un lenguaje estructurado de bloques. El texto completo de una definición de función debe ser un bloque escrito de la siguiente manera:

```
[ <<etiqueta>> ]
[ DECLARE
  declaraciones ]
BEGIN
  sentencias,
END;
```

Triggers

Los triggers (ejecución automática de funciones) son eventos disparadores que notifican a la BD si algún evento está ocurriendo. Los triggers son fragmento de código, que le dice a PostgreSQL que se ejecute antes de, o después de que una tabla sea modificada. Un trigger tiene el poder de:

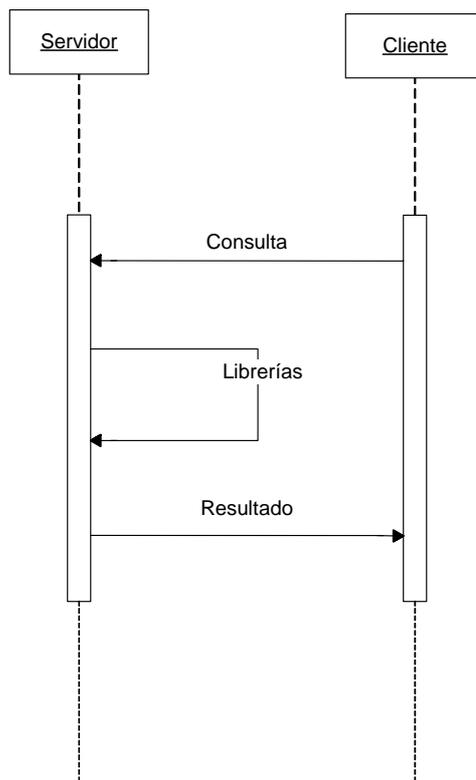


Figura 2.8: Esquema Cliente-Servidor.

- Asegurar que una columna contiene valores predefinidos.
- Asegurar que una fila de auditoría sea insertada en otra tabla.
- Luego de encontrar que la nueva información es inconsistente con otras cosas en la base de datos, levanta un error que causará que toda la transacción se aborte.

La sintaxis para la creación de triggers es la siguiente:

```
CREATE TRIGGER <nombre del trigger>  
<BEFORE |AFTER> <INSERT |DELETE|UPDATE>  
ON <nombre tabla> FOR EACH <ROW | STATEMENT>  
EXECUTE PROCEDURE <nombre del procedimiento> (<function args>).
```

Los triggers lanzados ANTES (BEFORE) pueden regresar un nulo para notificar al administrador de triggers que debe saltarse el resto de la operación (es decir, los triggers subsecuentes no deben dispararse, y el INSERT/ UPDATE/DELETE no ocurre para este renglón). Si se regresa un valor no nulo entonces la operación procede con ese valor de renglón. Regresar un valor de renglón diferente del valor original de NEW altera al renglón que será insertado o actualizado (pero no tiene un efecto directo en el caso DELETE). Para alterar el renglón que se va a almacenar, es posible reemplazar valores particulares directamente en NEW y regresar el NEW modificado o construir completamente el nuevo registro que se va a regresar.

El valor de retorno de un trigger de nivel sentencia BEFORE o AFTER o un trigger de nivel renglón siempre se ignora; también puede ser nulo. Sin embargo, cualquiera de estos tipos de triggers pueden abortar la operación entera al detectarse un error.

Un uso de los triggers es mantener una tabla como resumen de otra tabla. El resumen resultante puede usarse en lugar de la tabla original para ciertas consultas rápidas y obtener un gran ahorro en el tiempo de ejecución. Esta técnica suele usarse en data warehousing, en donde las tablas de datos medidos u observados (llamadas tablas factuales) pueden ser extremadamente grandes.

Funciones

PL/pgSQL puede usarse para definir procedimientos disparadores (triggers).

Un procedimiento trigger se crea con el comando CREATE FUNCTION, declarándola como una función sin argumentos y un tipo de retorno trigger.

Una función trigger debe regresar ya sea un NULL o un valor registro/renglón que tenga exactamente la misma estructura de la tabla para la cual se inició el trigger.

Cursores

Un cursor es un fragmento de memoria que se encarga de almacenar algún tipo de consulta o procedimiento, para su futura manipulación.

Además de ejecutar una consulta completa al mismo tiempo, es posible configurar un cursor para que encapsule la consulta y luego leer el resultado de la consulta en unas cuantas filas al mismo tiempo. Una razón de hacer esto es para evitar sobrecargas de memoria cuando el resultado contiene un gran número de filas. Un uso más interesante es retornar una referencia a un cursor que él ha creado, permitiendo al solicitante leer las filas. Esto proporciona una forma más eficiente de retornar largas filas de consultas desde funciones.

Todo el acceso a cursores en PL/pgSQL se realiza a través de variables cursor, las cuales son siempre del tipo de datos especial *refcursor* [13]. Una forma de crear una variable tipo cursor es declararla como de tipo *refcursor*.

```
nombre CURSOR [ ( argumentos ) ] FOR select_query;
```

Una vez que un cursor ha sido abierto, éste puede ser manipulado. Estas manipulaciones no necesitan ocurrir en la misma función que abrió el cursor. Puede retornar un valor *refcursor* de una función y permitir al solicitante operar con el cursor (internamente, un valor *refcursor* es simplemente la cadena nombre de un espacio de memoria conteniendo la consulta activa para el cursor). Este nombre puede ser pasado, asignado a otras variables *refcursor*, sin afectar al procedimiento.

Todos los cursores deben ser implícitamente cerrados al final de la transacción. Por tanto un valor *refcursor* es útil para referenciar un cursor abierto sólo hasta el final de la transacción.

2.2.2. Reglas de Análisis

Las reglas ECA que imitan el trabajo de una analista son llamadas *reglas de análisis*. Estas son modeladas desde la perspectiva de un analista quién especifica el tiempo exacto en el cual las reglas de análisis van a ser disparadas.

Una regla de análisis, es un proceso que surge como consecuencia de las tendencias de los sistemas de información, obtenidos según el giro del warehouse.

Una regla de análisis está formada por tres componentes: el *evento*, la *condición* y la *acción* [4]. La forma general para representar a una regla ECA es la siguiente:

on *evento*
if *condición*
then *acción*

Por lo tanto la automatización de un DWH no es un proceso trivial, se debe tener mucho cuidado con la definición de las reglas ya que el uso incorrecto de estos puede crear resultados no eficientes.

2.2.3. Modelo de conocimiento

El modelo de conocimiento del SBDA (Sistema de Base de Datos Activas)[3] indica al sistema la manera en que éste debe de comportarse ante la presencia de ciertos eventos relevantes y el estado que guarde la BD en el momento en que ocurran estos eventos.

El enfoque utilizado para el modelo de conocimiento del DWHA es la definición de reglas de análisis [3].

Elementos

Evento *Evento* es algo que ocurre en un punto del tiempo. Por lo que la especificación de un evento involucra la descripción de un suceso que va a ser monitoreado. La naturaleza de la descripción y la manera en que el evento puede detectarse, depende en gran medida de la fuente o generador de eventos. Los generadores de eventos pueden ser: *operaciones de estructura, de transacción, definidas por el usuario, de reloj o externos* [4].

Se citan a continuación en que consisten los generadores y se dan ejemplos basados en un supermercado para enfatizar éstos:

- **Operaciones de estructura:** el evento es generado a partir de una operación sobre algún elemento de la estructura de la BD.

Ejemplo 2.1 *al efectuar una venta agregar el producto a la tabla de vendidos, borrar el artículo de la tabla de existentes, etc.*

- **Transacción:** el evento es generado a partir de un comando de transacción.

Ejemplo 2.2 *al generar la transacción de compra-venta con un proveedor.*

- **Definidas por el usuario:** el evento es generado a partir de un mecanismo de programación para que explícitamente la aplicación detecte el evento.

Ejemplo 2.3 *cuando a la llegada de un trailer con uno o varios pedidos, se hace uso de una aplicación para introducir los artículos a la BD por parte de un usuario.*

- **Reloj:** se genera el evento en un punto determinado del tiempo.

Ejemplo 2.4 *el 20 de noviembre agregar artículos navideños.*

- **Externos:** un suceso externo a la BD genera el evento.

Ejemplo 2.5 *la venta de más artículos debido a una promoción del proveedor.*

Los eventos pueden ser *primitivos* o *compuestos*. El primero de éstos es alcanzado cuando se da una sola ocurrencia que pertenece a una de las posibilidades de los generadores de eventos mencionados anteriormente. Por otro lado, el *compuesto* se da a partir de alguna combinación entre primitivos y/o compuestos, utilizando una serie de constructores de eventos llamada *álgebra de eventos*.

- Eventos Primitivos

Eventos temporales absolutos: el evento es generado en el momento definido explícitamente.

Eventos temporales periodicos: son generados por períodos de tiempo.

Eventos temporales relativos: el evento se genera después de un suceso definido anteriormente y concluído.

- Eventos Compuestos

Los constructores u operadores del álgebra de eventos varían de acuerdo a los diferentes SBDA, pero debido al enfoque de la toma de decisiones operacionales para un DWH se dan ejemplos aplicables a este tipo de eventos compuestos. Entre los más importantes se encuentran:

Disyunción, $e_1 \vee e_2 \vee \dots \vee e_n$, en donde el evento compuesto de la disyunción ocurre cuando se ha suscitado uno de los eventos $e \in \{e_1, e_2, \dots, e_n\}$.

Ejemplo 2.6 *En el departamento de verdulería se desea quitar de exhibición todo aquel artículo, cuya fecha de caducidad o temporada haya expirado o terminado.*

Conjunción, $e_1 \wedge e_2 \wedge \dots \wedge e_n$, en donde el evento compuesto de la conjunción ocurre cuando se han suscitado todos los eventos $e_i \in \{e_1, e_2, \dots, e_n\}$.

Ejemplo 2.7 *Se desea hacer pedidos para surtir almacén, si el artículo x se vendió más del umbral esperado y, si en bodega, la existencias de éstos, sea menor del 30 % de las existencias.*

Secuencia, $\text{seq}(e_1, e_2, \dots, e_n)$, en donde el evento compuesto *secuencia* es alcanzado cuando ocurre en primera instancia el evento e_1 , posteriormente el evento e_2 y así sucesivamente hasta e_n .

Ejemplo 2.8 *Si se desea hacer algún convenio con un banco, se necesita ofrecer temporalmente una promoción (6 meses sin intereses pagando con tarjeta) y luego ver si su impacto financiero rebasó el umbral esperado, para poder acordar definitivamente con esta instancia.*

Simultáneo, $\text{sim}(e_1, e_2, \dots, e_n)$, este evento compuesto es alcanzado cuando ocurren los eventos e_1, e_2, \dots, e_n en el mismo instante de tiempo.

Ejemplo 2.9 *Se desea hacer un descuento adicional del 5 % en el cobro por venta especial, se requiere que la compra se efectúe después de las 10 pm y antes de las 12 am y que exista una venta especial Nocturna.*

Negación, $\sim e_1$ en *intervalo*, es la no ocurrencia del evento e_1 en el *intervalo* de tiempo especificado. Se dice que el evento ha tomado lugar al término del intervalo si e_1 no ocurre.

Ejemplo 2.10 *Si algún cliente quiere realizar su pago con tarjeta, se necesita la autorización del banco, pero si en el transcurso de 1 minuto no hay respuesta de transacción, se rechaza el pago de ésta.*

Primero, $first(e_1)$ en *intervalo*, este evento compuesto ocurre cuando en determinado *intervalo* se presenta en al menos una ocasión el evento e_1 , tomando para efecto de evaluación de la condición de la regla a la primera ocurrencia de e_1 dentro del *intervalo*.

Ejemplo 2.11 *Es muy común que en farmacia no se encuentre un cierto medicamento. Por lo tanto, se hacen pedidos para surtirse al día siguiente, entonces para el primer artículo no existente dentro de un tiempo de 9 am a 1pm se crea una lista de pedidos dentro de la tabla correspondiente.*

Ultimo, $last(e_1)$ en *intervalo*, a semejanza del evento anterior, este evento compuesto ocurre cuando ha tomado lugar al menos una vez el evento e_1 dentro del *intervalo*. Sin embargo, en este caso se considera a la última ocurrencia de e_1 dentro del *intervalo* especificado.

Ejemplo 2.12 *Al ejecutarse el último evento comprendido dentro de un tiempo de 9 am a 1pm, se manda una lista de pedidos al proveedor.*

Historia, $times(n, e_1)$ en *intervalo*, ocurre cuando el evento e_1 se ha suscitado n veces dentro del *intervalo* especificado.

Ejemplo 2.13 *Si algún empleado tiene más de 20 quejas reportadas externamente en 1 mes, se aplicará un correctivo.*

Alguno, $any(m, e_1, e_2, \dots, e_n)$, $m < n$, este evento compuesto ocurre cuando se han suscitado m eventos e_i distintos que pertenecen al conjunto $\{e_1, e_2, \dots, e_n\}$.

Ejemplo 2.14 *Si algún empleado tiene menos quejas reportadas, mayores ventas registradas y puntualidad, se le aplicará un bono extra.*

Condición La parte condicional de una regla es una expresión lógica la cual determina si la acción de la regla será ejecutada. La condición se evalúa contra el estado que guarde el estado de la BD en el momento en que el evento de la regla a ocurrido, si la evaluación de dicha expresión resulta verdadera entonces la acción ó conjunto de acciones especificadas en la acción de la regla serán ejecutadas.

En algunos sistemas, la especificación de la condición es opcional, formando de esta manera reglas de tipo Evento-Acción, donde a partir de la ocurrencia del evento se dispara la regla y la acción es ejecutada [3].

Acción La acción de una regla de análisis describe el conjunto de tareas que serán realizadas si la evaluación de la parte condicional de la regla resulta verdadera.

La acción de la regla puede ser la modificación de la estructura en la BD (operaciones de modificación, inserción ó eliminación de datos), el envío de una señal a un dispositivo externo a la BD, informar a un usuario o al administrador del sistema sobre alguna situación que esté ocurriendo en el estado de la BD, abortar una transacción o bien, tomar alguna acción alternativa utilizando el comando do-instead. [15]

La acción de la regla puede a su vez ser el evento que active alguna regla de análisis o formar parte de un evento compuesto que active a una regla de análisis.

Además de especificar en un DWHA el modelo de conocimiento correspondiente a los elementos necesarios para especificar correctamente a cada uno de ellos, es de vital importancia saber la manera en que serán manejados en tiempo de ejecución, por lo cual se hace necesaria la definición de un modelo de ejecución.

2.2.4. Modelo de ejecución

Además de un modelo para la definición de cada una de los atributos y características con que deben contar los elementos de la regla de análisis en el modelo de conocimiento, también se cuenta con un modelo de ejecución para el DWHA.

El modelo de ejecución es la manera en que la base de reglas de análisis serán manejadas en tiempo de ejecución, es decir, como se detectarán los eventos para activar reglas, realizar la verificación de la parte condicional de las reglas activadas, así como la ejecución de las acciones dentro o fuera de la BD.

El modelo de ejecución está muy relacionado con cada SBDA (Sistema de Base de Datos Activa), pero de manera general en todos los casos siguen las mismas fases en la ejecución de las reglas de análisis [3].

La primera fase se refiere a la aparición de los eventos que dispararán a las reglas (signaling). En la segunda fase se toman los eventos encontrados en la fase anterior y dispara a aquellas reglas cuyo evento de la regla hayan ocurrido (triggering). La tercera fase realiza la evaluación de la parte condicional de la regla (evaluation), formando un conjunto con aquellas reglas cuya evaluación de la condición haya resultado verdadera. En la cuarta fase (scheduling) se toma el conjunto de reglas



Figura 2.9: Principales pasos que toman lugar durante la ejecución de reglas.

formado en la tercera fase y se decide el orden en que serán procesadas. Finalmente, en la última fase (execution) se llevan a cabo las acciones correspondientes a las reglas disparadas (Fig. 2.9).

Las acciones ejecutadas en la última fase, pueden a su vez generar eventos a ser considerados dentro de la primera etapa, produciendo un disparo de reglas en cadena.

El modelo de ejecución determina el momento en que una regla de análisis, cuyo evento activador ha ocurrido, será disparada; además del momento en que su parte condicional será evaluada; así como el tiempo en que la acción será ejecutada. Existen tres tipos de modos de acoplamiento utilizados más frecuentemente entre los elementos de la regla de análisis evento-condición y condición-acción[3]:

- Modo inmediato.- En esta modalidad, la segunda parte de la pareja de elementos de regla ECA se lleva a cabo inmediatamente después de que la primera es terminada. Es decir, en evento-condición, la condición es evaluada inmediatamente después de que el evento es detectado; y en condición-acción, la acción es ejecutada inmediatamente después de que la condición es evaluada.
- Modo pospuesto.- Esta modalidad es utilizada cuando las operaciones del SBD se hacen mediante transacciones. En esta modalidad, la segunda parte de la pareja de elementos de regla de análisis se lleva a cabo hasta que la transacción donde se ocurrió la primera ha terminado. Es decir, en evento-condición, la condición es evaluada después de que termina la transacción donde el evento es detectado; y en condición-acción, la acción es ejecutada después de que termina la transacción donde la condición es evaluada.
- Modo separado.- Al igual que en la modalidad anterior, ésta también es utilizada cuando las operaciones del SBD se realiza mediante transacciones. En el modo separado, la segunda parte

de la pareja de elementos de análisis se lleva a cabo inmediatamente después que la primera es terminada; sin embargo, se realiza en una transacción distinta. Es decir, en evento-condición, la condición es evaluada inmediatamente después de que el evento es detectado, pero en una transacción distinta a aquella donde el evento es detectado; y en condición-acción, la acción es ejecutada inmediatamente después de que la condición es evaluada, pero en una transacción distinta a la de la evaluación de la condición.

Ya teniendo una definición de los modelos de conocimiento y de ejecución que un SBDA debe ofrecer, es necesario observar algunas de las vicisitudes que se presentan durante el desarrollo de base de reglas de análisis ya que darán apoyo a la toma de decisión.

2.3. Sistema de Soporte a la Toma de Decisión

El término de DSS surgió en 1971 con los profesores G. Anthony Gorry y Michael S. Scott Morton[9]. Ellos proponían apoyar a la toma de decisión mediante la elaboración de aplicaciones computacionales, que les permitieran agrupar a los problemas y niveles administrativos de forma estructurada, así como implementar reglas o procesos específicos.

Después es definido más ampliamente en [19], donde se dice que es un “sistema de información basado en computadora que combina modelos y datos en una tentativa para resolver problemas semiestructurados con un involucramiento pleno del usuario”.

Los DSS se presentan como la opción viable para poder dar soporte y consistencia a la información que se diversificó enormemente. Con una metodología más estricta y más herramientas a su alcance, los DSS comenzaron a ser la contraparte de los sistemas OLTP, entendiéndose a partir de sus características de funcionalidad en una empresa como apoyo a la toma de decisión en ésta.

El carácter genérico del término DSS ha dado origen a sistemas específicos enfocados a tipos concretos de problemas, como pueden ser:

- Sistemas de Información Gerencial (Management Information Systems, MIS), sistema integrado usuario-máquina para proveer información que apoye las operaciones, la administración y las funciones de toma de decisiones en una empresa. Utiliza equipo de cómputo (hardware) y programas (software), procedimientos manuales, modelos para el análisis, la planeación, el control y la toma de decisiones, además de una BD.

- **Sistemas Expertos (Expert Systems, ES)**, un sistema experto es una rama de la Inteligencia Artificial y es aquel que imita las actividades de un humano para resolver problemas de distinta índole (no necesariamente tiene que ser de Inteligencia Artificial). También se dice que un ES se basa en el conocimiento declarativo (hechos sobre objetos, situaciones) y el conocimiento de control (información sobre el seguimiento de una acción).
- **Sistemas de Información para Ejecutivos (Executive Information Systems, EIS)**, están dirigidos a apoyar el proceso de toma de decisiones de los altos ejecutivos de una organización, presentando información relevante y usando recursos visuales y de fácil interpretación, con el objetivo de mantenerlos informados. Están diseñados para cubrir las necesidades específicas y particulares de la alta administración de la empresa.
- **Ayuda a la Decisión en Grupos (Group Decision Support Systems, GDSS)**, se entiende la toma de decisión en grupo como un proceso en el cual un grupo de personas están interesadas en solucionar un problema común. En la mayoría de las organizaciones las decisiones se toman de manera colectiva, independientemente de si se trata del sector público o privado, nacional o internacional. Suele ser difícil conseguir el consenso en el seno del grupo. Es por ello que los sistemas de ayuda a la decisión en grupo (GDSS) se desarrollan para apoyar estos procesos de decisión.
- **Ayuda a la administración (Management Support Systems, MSS)**, la función del MSS es facilitar una amplia gama de servicios centralizados de apoyo administrativo, asesoramiento e información sobre gestión a los directores de las direcciones y a los administradores de los departamentos en los sectores de finanzas, presupuesto, personal y viajes, de conformidad con el reglamento y estatuto establecidos de la Organización.
- **Ayuda a los Ejecutivos (Executive Support Systems, ESS)**, ayudan a los gerentes con problemas no estructurados y se enfocan en las necesidades de información de la alta dirección, combinando los datos de fuentes internas y externas, crean un entorno generalizado de computación y comunicaciones que se pueden enfocar y aplicar a una serie cambiante de problemas, ayudan a los directores a monitorear el desempeño organizacional, dar seguimiento a las actividades de los competidores, señalar problemas, identificar oportunidades y a pronosticar tendencias.

Las aplicaciones de estas herramientas, las más complejas, se dan a partir del análisis multidimensional de los datos corporativos, las cuales, proporcionan la habilidad de manipular y explorar los datos de la empresa desde cualquier ángulo concebible, con lo cual pueden obtener una visión verdaderamente multidimensional de la empresa.

2.3.1. Definición

Un DSS es un modelo basado en un conjunto de procedimientos para procesar datos y juicios que ayudan a los administradores a tomar decisiones.

Por otra parte [15] define un DSS como un sistema basado en computadoras con tres componentes claves:

- un sistema de lenguaje (un mecanismo para proveer comunicación entre el usuario y otros componentes del DSS),
- un sistema de conocimiento (repositorio de conocimiento sobre problemas, datos o procesos) y
- el sistema procesador de problemas (la unión entre los otros dos componentes).

Otra definición la considera como sistemas basados en sistemas de cómputo que ayudan a quien toma decisiones enfocados a problemas mal estructurados a través de una directa interacción con datos y modelos de análisis [12].

Existen otras definiciones de DSS [11], sin embargo, en un sentido amplio, se define a este sistema, como un conjunto de programas y herramientas que permiten obtener de manera oportuna la información que se requiere mediante el proceso de la toma de decisiones que se desarrolla en un ambiente de incertidumbre. Ayudan a la toma de decisiones de los administradores al combinar datos, modelos analíticos sofisticados y software amigable en un solo sistema poderoso que puede dar soporte a la toma de decisiones semiestructuradas o no estructuradas.

2.3.2. Clasificación

Los DSS se pueden clasificar dependiendo el nivel de decisión según [12] (ver figura 2.10):

- objetivos,

- horizontes,
- tipos de datos,

Como se ha indicado, una clasificación de los modelos podría ser en base a sus objetivos: búsqueda de soluciones que maximicen una determinada función (de beneficios o cualquier otro tipo), o bien modelos puramente descriptivos, que se limitan a describir el comportamiento del sistema analizado, sin intentar optimizar ninguna función objetivo [18].

Otra clasificación de interés desde el punto de vista empresarial, es la que identifica cuatro tipos de modelos, jerarquizables según el horizonte temporal, el nivel de agregación de los datos, y las características del usuario, en donde, en este último, los sistemas de información (SI) están categorizados como:

- Sistemas Estratégicos
- Sistemas Tácticos
- Sistemas Técnico-Operativos u operacionales
- Sistemas Interinstitucionales

Sistemas Estratégicos, orientados a soportar la toma de decisiones, facilitan la labor de la dirección, proporcionándole un soporte básico, en forma de mejor información, para la toma de decisiones. Se caracterizan porque son sistemas sin carga periódica de trabajo, es decir, su utilización no es predecible, al contrario de los casos anteriores, cuya utilización es periódica. Trabajan con horizontes a largo plazo, con datos agregados (ventas totales anuales, plantilla global de la compañía, etc.).

Destacan entre estos sistemas: los Sistemas de Información Gerencial (MIS), Sistemas de Información Ejecutivos (EIS), Sistemas de Información Georeferencial (GIS), Sistemas de Simulación de Negocios (BIS y que en la práctica son sistemas expertos o de Inteligencia Artificial-AI).

Sistemas Tácticos, diseñados para soportar las actividades de coordinación de actividades y manejo de documentación, definidos para facilitar consultas sobre información almacenada en el sistema, proporcionar informes y, en resumen, facilitar la gestión independiente de la información por parte de los niveles intermedios de la organización. Tiene horizontes temporales menores (varios

meses), los datos son de mayor detalle, destacan entre ellos: los Sistemas Ofimáticos (OA), Sistemas de Transmisión de Mensajería (E-mail y Fax Server), coordinación y control de tareas (Work Flow) y tratamiento de documentos (Imagen, Trámite y BD's Documentarios).

Sistemas Técnico-Operativos, que cubren el núcleo de operaciones tradicionales de captura masiva de datos (Data Entry) y servicios básicos de tratamiento de datos, con tareas predefinidas (contabilidad, facturación, almacén, presupuesto, personal y otros sistemas administrativos). Estos sistemas están evolucionando con la irrupción de sensores, autómatas, sistemas multimedia, BD relacionales más avanzadas y data warehousing. Son de máximo detalle, con decisiones a muy corto plazo (días) y manejadas por personal de planta (secuenciación de trabajos, gestión de inventarios, etc).

Sistemas Interinstitucionales, este último nivel de sistemas de información recién está surgiendo, es consecuencia del desarrollo organizacional orientado a un mercado de carácter global, el cual obliga a pensar e implementar estructuras de comunicación más estrechas entre la organización y el mercado (Empresa Extendida, Organización Inteligente e Integración Organizacional), todo esto a partir de la generalización de las redes informáticas de alcance nacional y global (internet), que se convierten en vehículo de comunicación entre la organización y el mercado, no importa dónde esté la organización (intranet), el mercado de la institución (extranet) y el mercado (Red Global).

En la figura 2.10 puede observarse la clasificación de Watson que solo propone 3 decisiones de las 4 obtenidas de otros analistas, ya que Watson supone estos conceptos como el origen de la toma de decisión según el horizonte de los datos hacia la decisión.

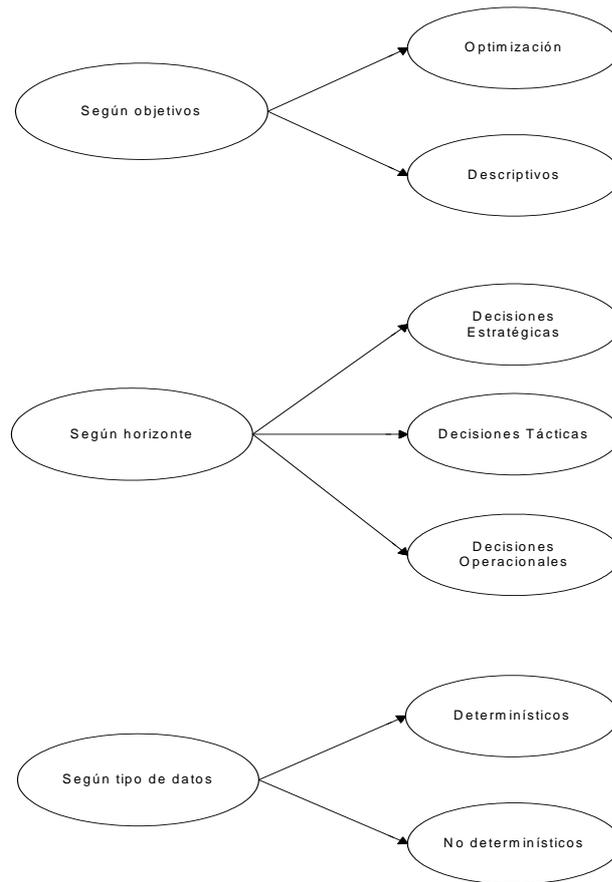


Figura 2.10: Clasificación de los modelos en los DSS según Sprague, Watson [12].

Capítulo 3

Sistemas de Apoyo a la Toma de Decisiones con Lógica Fuzzy y DWH Activas.

En este capítulo se mencionan algunos softwares existentes para la toma de decisión con lógica fuzzy y de la creación de DWH's para apoyo a la toma de decisión así como algunas propuestas para obtener DWH activas.

3.1. DSS

En [9] proponían apoyar a la toma de decisión mediante la elaboración de aplicaciones de computo, que les permitieran agrupar los problemas y niveles administrativos de forma estructurada, así como implementar reglas o procesos específicos.

El trabajo es ampliado en [19], los autores dicen que es un “sistema de información basado en computadora que combina modelos y datos, en una tentativa para resolver problemas semiestructurados con un involucramiento pleno del usuario”.

Los DSS se presentan como la opción viable para poder dar soporte y consistencia a la información que se diversificó enormemente. Con una metodología más estricta y más herramientas a su alcance, los DSS comenzaron a ser la contraparte de los sistemas OLTP, entendiéndose a partir

de sus características de funcionalidad en una empresa, sirviendo de apoyo a la toma de decisión en ésta.

En [23] se presenta una herramienta basado en agentes *fuzzy* llamado TEAKS (TEAm Knowledge-based Structuring), con el objetivo de ser usado para simular el desarrollo del trabajo en equipo. Cada agente TEAKS simula a una persona, mediante un conjunto de parámetros(características humanas), dando como respuesta comportamiento humano a través del agente, siendo consecuencia de estos parámetros, para así poder asignarle una tarea específica a cada agente.

Al definir los agentes y los parámetros para las tareas requeridas, se simuló el comportamiento del agente, mediante la declaración de conjuntos fuzzy, obteniendo un buen resultado, al evaluar sus respuestas para el trabajo en equipo, siguiendo las características de cada individuo. Esta interfaz sirvió para la correcta asignación de actividades a cada tipo de agente, según su perfil, haciendo que la toma de decisión fuera específica para cada candidato.

El concepto de lógica *fuzzy* apareció en 1965, en la Universidad de California en Berkeley, introducido por Lotfi A. Zadeh [25]. La lógica *fuzzy* procura crear aproximaciones matemáticas en la resolución de ciertos tipos de problemas. Pretende producir resultados exactos a partir de datos imprecisos, por lo cual son particularmente útiles en aplicaciones electrónicas o computacionales. En aspectos computacionales puede ser usado para definir los DSS.

3.2. Lógica *fuzzy*

Es un tipo de lógica que reconoce más que simples valores verdaderos y falsos. Con lógica *fuzzy*, las proposiciones pueden ser representadas con grados de veracidad o falsedad. Por ejemplo, la sentencia “hoy es un día soleado”, puede ser 100 % verdad si no hay nubes, 80 % verdad si hay pocas nubes, 50 % verdad si existe neblina y 0 % si llueve todo el día.

La lógica *fuzzy* ha probado ser particularmente útil en sistemas expertos y otras aplicaciones de inteligencia artificial. Es también utilizada en algunos correctores de voz para sugerir una lista de probables palabras a reemplazar en una frase mal dicha. La lógica *fuzzy*, que hoy en día se encuentra en constante evolución, nació en los años 60 como la lógica del razonamiento aproximado, y en ese sentido podría considerarse una extensión de la Lógica Multivaluada. La lógica multivaluada incluye sistemas lógicos que admiten varios valores de verdad posibles.

La lógica *fuzzy* (llamada también lógica difusa por otros autores) o Fuzzy Logic es básicamente

una lógica con múltiples valores, que permite definir valores en las áreas oscuras entre las evaluaciones convencionales de la lógica precisa: Si / No, Cierto / Falso, Blanco / Negro, etc. Con la lógica *fuzzy*, las proposiciones pueden ser representadas con grados de certeza o falsedad.

Por medio de la lógica *fuzzy* pueden formularse matemáticamente nociones, para que sean procesadas por computadoras y cuantificar expresiones humanas vagas, tales como “Muy alto” o “poca agua”. Permite también cuantificar aquellas descripciones imprecisas que se usan en el lenguaje y las transiciones graduales en electrodomésticos, como pasar de agua sucia a agua limpia en una lavadora, lo que permite ajustar los ciclos de lavado a través de sensores. La habilidad de la lógica *fuzzy* para procesar valores parciales de verdad ha sido de gran ayuda para la ingeniería. En general, se ha aplicado a:

- Ingeniería:

- Sistemas expertos,

- Verificadores de ortografía, los cuales sugieren una lista de palabras probables para reemplazar una palabra mal escrita,

- Control de sistemas de trenes subterráneos,

- Administración Social:

- Configuración de equipos de trabajo,

- Administración de cadenas de suministro, entre otros.

Los esquemas de razonamiento utilizados son “esquemas de razonamiento aproximado”, que intentan reproducir los esquemas mentales del cerebro humano en el proceso de razonamiento. Estos esquemas consistirán en una generalización de los esquemas básicos de inferencia en Lógica Binaria (silogismo clásico)[29].

Tan importante será la selección de un esquema de razonamiento como su representación material, ya que el objetivo final es poder desarrollar un procedimiento analítico concreto para el diseño de controladores *fuzzy* y la toma de decisiones en general. Una vez que dispongamos de

representaciones analíticas de cada uno de los elementos lógicos, estaremos en disposición de desarrollar formalmente un controlador “heurístico” que nos permita inferir el control adecuado de un determinado proceso en función de un conjunto de reglas “lingüísticas”, definidas de antemano tras la observación de la salida y normas de funcionamiento de éste.

Enfoque lingüístico fuzzy

En esta sección, presentamos algunos conceptos básicos sobre el enfoque lingüístico *fuzzy* utilizado para representar la información lingüística en los problemas de toma de decisión.

Cuando trabajamos con conocimiento impreciso, no podemos estimar de forma precisa un valor numérico. Entonces, un enfoque más realista es el uso de etiquetas lingüísticas en lugar de valores numéricos, es decir, asumimos que las variables que participan en el problema son valoradas mediante términos lingüísticos [26].

Este enfoque es apropiado para gran cantidad de problemas, ya que permiten una representación de la información de una forma más directa y adecuada en caso de ser incapaces de expresarla de forma precisa.

Normalmente, dependiendo del dominio del problema, se elige un conjunto de términos lingüísticos adecuado que se utiliza para describir el conocimiento vago o impreciso. El número de elementos en el conjunto de etiquetas determina lo que se denomina la granularidad, es decir, el grado de distinción entre diferentes grados de incertidumbre. En [26] se describe el uso de conjuntos de etiquetas lingüísticas con un número impar de etiquetas, donde el término medio representa “aproximadamente 0.5”, con el resto de etiquetas distribuidos simétricamente a su alrededor. Ejemplo de un conjunto de etiquetas lingüísticas:

$$\{P, MA, A, M, B, MB, N\}$$

La semántica de las etiquetas lingüísticas viene dada por números difusos definidos en el intervalo $[0,1]$, los cuáles son descritos mediante funciones de pertenencia.

Debido a que los términos lingüísticos no son más que aproximaciones dadas por individuos, consideraremos que el uso de funciones de pertenencia trapezoidales son lo suficientemente buenas como para capturar la vaguedad de las valoraciones lingüísticas, ya que obtener valores más precisos es imposible o innecesario. Esta representación se obtiene mediante una 4-tupla (a,b,d,c) , con b y d indicando el intervalo donde la función de pertenencia es 1 y con a y c siendo los límites izquierdo

y derecho de la función de pertenencia. Un caso particular de este tipo de funciones de pertenencia son las triangulares que son aquellas en que, $b=d$, por lo que se representan como (a,b,c) . Por ejemplo, podríamos asignar la siguiente semántica al anterior conjunto de 7 etiquetas:

1. N=nada, $(0, 0, 0.17)$,
2. MB=Muy_Bajo $(0, 0.17, 0.33)$,
3. B=bajo, $(0.17, 0.33, 0.5)$
4. M=medio $(0.33, 0.5, 0.67)$,
5. A=alto $(0.5, 0.67, 0.83)$,
6. MA= Muy_Alto $(0.67, 0.83, 1)$
7. P=Perfecto $(0.83, 1, 1)$,

Etiquetas lingüísticas y operadores.

Desde su raíz, una variable lingüística es el nombre de un conjunto *fuzzy*. Si tenemos un conjunto *fuzzy* llamado “largo” éste es una simple variable lingüística y puede ser empleada como una regla-base en un sistema basado en la longitud de un proyecto en particular.

En el campo de la semántica *fuzzy* cuantitativa el significado de un término “x” se le representa como un conjunto *fuzzy* $\mu(x)$ del universo de discusión. Desde este punto de vista, uno de los problemas básicos en semántica es que se desea calcular el significado de un término compuesto.

La idea básica sugerida por Zadeh [28] es que una etiqueta lingüística tal como “muy”, “más o menos”, “ligeramente”, etc... puede considerarse como un operador que actúa sobre un conjunto *fuzzy* asociado al significado de su operando. Por ejemplo, en el caso de un término compuesto “muy alto”, el operador “muy” actúa en el conjunto *fuzzy* asociado al significado del operando “alto”. Es importante aclarar que se hará mayor énfasis en que estas representaciones se proponen principalmente para ilustrar el enfoque, más que para proporcionar una definición exacta de las etiquetas lingüísticas. Zadeh también considera que las etiquetas lingüísticas pueden clasificarse en dos categorías que informalmente se definen como sigue:

- Tipo I: las que pueden representarse como operadores que actúan en un conjunto *fuzzy*: “muy”, “más o menos”, “mucho”, “ligeramente”, “altamente”, “bastante”, etc. y,
- Tipo II: las que requieren una descripción de cómo actúan en los componentes del conjunto *fuzzy* (operando): “esencialmente”, “técnicamente”, “estrictamente”, “prácticamente”, “virtualmente”, etc...

En otras palabras, las etiquetas lingüísticas pueden ser caracterizadas como operadores más que operaciones primitivas de conjuntos *fuzzy*.

Conjuntos *fuzzy*

Los conjuntos *fuzzy* fueron introducidos por primera vez en 1965; la creciente disciplina de la lógica *fuzzy* provee por sí misma un medio para acoplar estas tareas [25]. En cierto nivel, la lógica *fuzzy* puede ser vista como un lenguaje que permite trasladar sentencias sofisticadas en lenguaje natural a un lenguaje matemático formal. Mientras la motivación original fue ayudar a manejar aspectos imprecisos del mundo real, la práctica temprana de la lógica *fuzzy* permitió el desarrollo de aplicaciones prácticas.

En 1994, la teoría de la lógica *fuzzy* se encontraba en la cumbre, pero esta idea no es nueva, para muchos, estuvo bajo el nombre de lógica *fuzzy* durante 25 años, pero sus orígenes se remontan hasta 2,500 años. Aún Aristóteles consideraba que existían ciertos grados de veracidad y falsedad. Platón había considerado ya grados de pertenencia.

Los conjuntos clásicos se definen mediante un predicado que da lugar a una clara división del Universo de Discurso X en los valores “Verdadero” y “Falso”. Sin embargo, el razonamiento humano utiliza frecuentemente predicados que no se pueden reducir a este tipo de división: son los denominados predicados vagos.

Por ejemplo, tomando el Universo de Discurso formado por todas las posibles temperaturas ambientales en una ciudad, se puede definir en dicho universo el conjunto A como aquél formado por las temperaturas “cálidas”. Por supuesto, es imposible dar a A una definición clásica, ya que su correspondiente predicado no divide el universo X en dos partes claramente diferenciadas. No podemos afirmar con rotundidad que una temperatura es “cálida” o no lo es. El problema podría resolverse en parte considerando que una temperatura es “cálida” cuando su valor supera cierto

umbral fijado de antemano. Se dice que el problema tan sólo se resuelve en parte, y de manera no muy convincente, por dos motivos: de una parte el umbral mencionado se establece de una manera arbitraria, y por otro lado podría darse el caso de que dos temperaturas con valores muy diferentes fuesen consideradas ambas como “cálidas”. Evidentemente, el concepto “calor” así definido nos daría una información muy pobre sobre la temperatura ambiental.

La manera más apropiada de dar solución a este problema es considerar que la pertenencia o no pertenencia de un elemento x al conjunto A no es absoluta sino gradual. En definitiva, definiremos A como un conjunto *fuzzy*. Su función de pertenencia ya no adoptará valores en el conjunto discreto $\{0,1\}$ (lógica booleana), sino en el intervalo cerrado $[0,1]$. En conclusión podemos observar que los conjuntos *fuzzy* son una generalización de los conjuntos clásicos.

Mediante notación matemática se define un Conjunto *fuzzy* A como:

$$\mu_A(x) = \{\mu_1/x_1, \mu_2/x_2, \dots, \mu_n/x_n\},$$

La función de pertenencia se establece de una manera arbitraria, lo cual es uno de los aspectos más flexibles de los conjuntos *fuzzy*. Por ejemplo, se puede convenir que el grado de pertenencia de una temperatura de “45°C” al conjunto A es 1, el de “25°C” es 0.4 , el de “6°C” es 0, etc.; cuanto mayor es el valor de una temperatura, mayor es su grado de pertenencia al conjunto A

$$\mu_A(x) = f(x).$$

Tipos de funciones de pertenencia.

Existen diferentes tipos de funciones de pertenencia de todos los posibles:

- el tipo triangular, que puede ser un caso concreto del trapezoidal en el que los dos valores centrales son iguales,
- el de forma de campana gaussiana,
- entre otros.

Tómese ahora el Universo de Discurso de la edad. El Conjunto *fuzzy* “Joven” representa el grado de pertenencia respecto al parámetro juventud que tendrían los individuos de cada edad. Es decir, el conjunto expresa la posibilidad de que un individuo sea considerado joven. Un Conjunto *fuzzy* podría ser considerado como una distribución de posibilidad, que es diferente a una distribución de probabilidad.

El grado de pertenencia de x en A , como ya se ha señalado anteriormente, se representa por $\mu_A(x)$. El Conjunto *fuzzy* A es la unión de los grados de pertenencia para todos los puntos en el Universo de Discurso X , que también puede expresarse como:

Bajo la notación de los Conjuntos *fuzzy*, $\mu_A(x)/x$ es un elemento del conjunto A .

Operaciones entre Conjuntos *fuzzy*:

Los Conjuntos *fuzzy* se pueden operar entre sí del mismo modo que los conjuntos clásicos. Puesto que los primeros son una generalización de los segundos, es posible definir las operaciones de intersección, unión y complemento haciendo uso de las mismas funciones de pertenencia:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

En realidad, estas expresiones son bastante arbitrarias y podrían haberse definido de muchas otras maneras. Esto obliga a considerar otras definiciones más generales para las operaciones entre los Conjuntos *fuzzy*.

Variables Lingüísticas

La Teoría de Conjuntos *fuzzy* puede utilizarse para representar expresiones lingüísticas que se utilizan para describir conjuntos o algoritmos. Los Conjuntos *fuzzy* son capaces de captar por sí mismos la vaguedad lingüística de palabras y frases comúnmente aceptadas, como “gato pardo” o “ligero cambio”. La habilidad humana de comunicarse mediante definiciones vagas o inciertas es un atributo importante de la inteligencia.

Una Variable Lingüística es aquella variable cuyos valores son palabras o sentencias que van a enmarcarse en un lenguaje predeterminado. Para estas variables lingüísticas se utilizará un nombre y un valor lingüístico sobre un Universo de Discurso. Además, podrán dar lugar a sentencias generadas por reglas sintácticas, a las que se les podrá dar un significado mediante distintas reglas semánticas.

Los Conjuntos *fuzzy* pueden utilizarse para representar expresiones tales como:

Ejemplo 3.1 X es PEQUEÑO.

Ejemplo 3.2 La velocidad es RÁPIDA.

Ejemplo 3.3 *El ganso es CLARO.*

Las expresiones anteriores pueden dar lugar a expresiones lingüísticas más complejas como:

Ejemplo 3.4 *X no es PEQUEÑO.*

Ejemplo 3.5 *La velocidad es RÁPIDA pero no muy RÁPIDA.*

Ejemplo 3.6 *El ganso es CLARO y muy ALEGRE.*

Así, se pueden ir complicando las expresiones. Por ejemplo, la expresión “x no es PEQUEÑO” puede calcularse a partir de la original calculando el complemento de la siguiente forma:

$$\mu_{\text{no_PEQUEÑO}}(x) = 1 - \mu_{\text{PEQUEÑO}}(x)$$

Tratando de esta forma los distintos modificadores lingüísticos (muy, poco, rápido, lento...) pueden ir calculándose todas las expresiones anteriores.

La simulación del comportamiento humano es una de las principales investigaciones en la Inteligencia Artificial y por lo tanto sus técnicas pueden ser empleadas para simular el comportamiento social humano en particular en la iteración social y su comportamiento en un equipo de trabajo [23].

Otra situación que pone en claro el uso de la lógica *fuzzy* fue el replanteamiento del sistema de la tienda 7-eleven en donde proponen con conjuntos *fuzzy* y reglas, la clasificación y la exposición de sus artículos en anaqueles en tiempos claves para la tienda, en donde analizan con tiempos, el consumo de bebidas y así saber cada cuando deben reabastecer la exposición de estos para el consumidor [24].

Debido a la complejidad del gravamen de transportar artículos (agilidad), en [31] desarrollaron un índice *fuzzy* de la agilidad (*fuzzy agility index*, FAI) basado en abastecedores para mejorar la agilidad en el transporte de la mercancía usando lógica *fuzzy*. El FAI abarca grados de los atributos de los artículos y los pesos correspondientes, y es agregado a los conjuntos *fuzzy*. Para ilustrar la eficacia del método, este estudio también evalúa la agilidad de transporte de mercancía para la cadena de abastecimiento de una compañía Taiwanesa. Esta evaluación demuestra que el método puede proveer a analistas una información más específica y más confiable para la toma de decisión.

3.3. DWH

Los sistemas OLAP son categorizadas como la tecnología de software que habilita a analistas, directores y ejecutivos para obtener conocimiento de los datos, valiéndose de la gran variedad de vistas con la información que ha sido transformada para reflejar la real dimensionalidad de la empresa. Donde el DWH, apoyada por ésta, es considerada como una BD para el soporte de toma de decisión.

Muchas empresas se apoyan de la tecnología DWH para mejorar su potencial, un ejemplo es el caso de la organización Soriana que desarrolló un sistema basado en tecnología de atención al cliente. Así, mediante desarrollos internos y una aplicación CRM, conoce la frecuencia de compra por hora y día de la semana y evalúa los programas de lealtad y explota esa información gracias al DWH de MicroStrategy.

- MicroStrategy [10].

Es un motor de análisis de información, que rescata de todas las áreas departamentales de la organización y accede directamente a la misma a partir de un solo DWH corporativo, presentado en un solo reporte.

Con una arquitectura unificada y un solo metadato los usuarios pueden moverse desde los tableros de control hasta los reportes y análisis, sin cambiarse de herramientas dentro de su interfaz.

Se puede obtener varias vistas directamente desde los sistemas operacionales e información en estructuras multidimensionales tales como SAP Business Warehouse (SAP BW). Los usuarios pueden obtener este análisis junto con otra información empresarial dentro de un solo reporte o tablero de control altamente formateado [10].

- INFORMIX-MetaCube Aggregator [8].

Con esta herramienta se puede fácilmente administrar y afinar el DWH según las necesidades de la empresa, creando y manteniendo datos pre-calculados y resumidos, conocidos como datos agregados.

- TERADATA [8]

Es una compañía que ofrece software a la medida, complementándolos con programas comerciales y donde maneja su propia DBMS Teradata para dar análisis OLAP y soporte a los DWH.

- Warehouse Manager de Prism Solutions[11]

Es un servidor para DWH creado por William H. Inmon es el primer software que se implementó para Warehouse. Está basado en modelado multidimensional y ofrece reportes OLAP para el análisis de sus datos.

- OLAP Server [10].

Es un servidor para consultas OLAP, en donde los datos usando el modelo objeto-relacional, multidimensional y relacional, optimiza las consultas de sus datos en forma gráfica y de reportes; todo en un ambiente dinámico y gráfico.

DWH Activas no comerciales

Wal-Mart de México es una de las cadenas comerciales más importantes de México, actualmente tiene implementado un sistema para llevar un mejor control de sus artículos. Este sistema automatiza existencias reales de los artículos, pedidos, ventas por día, ventas por semanas, entre otras operaciones necesarias para la optimización de éstos y mejorar su servicio al cliente.

En [2] proponen una arquitectura de DWH para automatizar la toma de decisión empleando la idea de evento, condición y acción desde los sistemas de base de datos activas complementando las herramientas OLAP con reglas de análisis para obtener un Data Warehouse Activo(DWHA), automatizan las tareas de decisión en un DWHA requiriendo solamente de un simple modelo de evento, el cuál consiste de eventos temporales (reloj) compuestos por tres características, eventos temporales absolutos(ejemplo, 15 de julio, 2005), eventos temporales periódicos (ejemplo, al final de cada semestre) y eventos temporales relativos (tres semanas después de que un cierto artículo haya sido lanzado al mercado), excluyendo los demás generadores de eventos ya que aseguran que un DWH sólo depende absolutamente del *tiempo* como característica básica de estos.

En [22] define y estructura una regla de negocio como el principal componente de los sistemas de análisis y presenta un meta-modelo para estas reglas de negocio, el cual consiste en concentrar en un almacén todas las especificaciones para llevar a cabo un análisis basado en el concepto de

las reglas ECA extendiendo con su modelo al concepto ECAA (Evento-Condición-Acción-Activa) donde propone una última acción en caso de no llevarse a cabo la primera. Para así obtener una BD Activa.

DWH Activas Comerciales.

Por otro lado la compañía Teradata creó un DWH Activa que conjuntamente con minería de datos y CRM, busca entender y anticipar las necesidades de los clientes existentes y también de los potenciales, apoyándose en soluciones tecnológicas que facilitan su aplicación, desarrollo y aprovechamiento, en donde lleva relación de sus eventos en un cierto tiempo para poder modelar su comportamiento y poder notificar este, mediante reportes y triggers al analista y poder tomar decisiones estratégicas y tácticas [8].

Capítulo 4

Implementación del DWH Activo.

4.1. Introducción.

El primer objetivo de un DWH es obtener los datos necesarios para la manipulación y análisis de los datos, recopilados de los diferentes sistemas OLTP. Una vez extraídos estos, se puede proceder a realizar el diseño del warehouse.

En este capítulo se describen los pasos necesarios para llegar a desarrollar nuestro STDE_ECAF (Sistema para la Toma de Decisiones Estratégicas con reglas ECA Fuzzy), en donde se especifica el diseño que se aplicó para el DWH, las reglas de análisis que fueron modeladas haciendo uso de cursores, los diagramas de clases que se implementaron en nuestra interfaz gráfica como adición para implementar reglas de análisis directamente a nuestro DWH y así poder crear automatización en las operaciones cotidianas en una tienda comercial.

Este proceso es previo al desarrollo e implementación del Sistema para la Toma de Decisiones Estratégicas (capítulo siguiente), ya que el correcto diseño y construcción del DWH, nos dará la pauta para desarrollar e incorporar las reglas ECA *Fuzzy* dentro del warehouse, y así hacer activo la Toma de Decisiones Estratégicas.

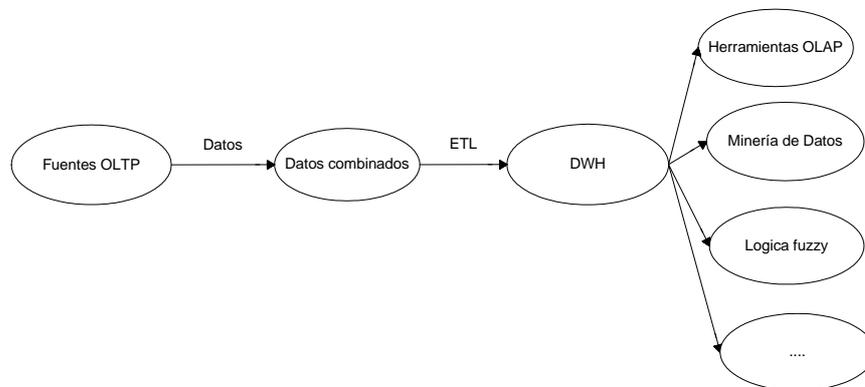


Figura 4.1: Estructura de un Data Warehouse.

4.2. Concepto

DWH

En la figura 4.1 se muestran dos procesos importantes para llegar a obtener un DWH, siendo los sistemas OLTP los cuales son la fuente de datos de diferentes DBMS, mostrando el estado actual de la empresa, en el cual los datos son seleccionados y almacenados temporalmente hasta por tres meses en un repositorio(datos combinados), para así pasar al proceso de limpieza de los datos “sucios”, ya que son datos no estandarizados y no definidos, en donde el proceso ETL se encarga de extraerlos de esta fuente(datos combinados), transformarlos en un formato general(integración) y después cargarlos al DWH. Una vez que los datos han sido cargados en el DWH están listos para su manipulación por diferentes interfaces, que van desde herramientas OLAP para el análisis de estos hasta minería de datos y así poder tomar decisiones según el giro del DWH.

DWHA

Un DWH Activa (DWHA) es un warehouse que mediante reglas de análisis y triggers puede detectar eventos necesarios para automatizar la toma de decisiones, que debieron ser anticipadas por los usuarios conocedores del comportamiento de los datos.

Algunas de las aplicaciones del DWHA son:

- monitoreo de transacciones financieras,
- identificación de actividades inusuales en el sistema,
- cumplimiento de restricciones de integridad,
- mantenimiento de datos derivados, generación oportuna de reportes,
- realización de procesos periódicos, entre otras.

4.3. Diseño

Un DWH organiza sus datos usando el modelo multidimensional, el cual clasifica estos dentro de *medidas* y *dimensiones*. Las medidas son la unidad básica de interés para el análisis, tal como figuran las ventas o costos. Las dimensiones son las diferentes perspectivas para ver las medidas, como por ejemplo: producto, ubicación y tiempo.

La definición de los requerimientos del negocio determina los datos necesarios para cumplir los requerimientos analíticos de los usuarios. Diseñar los modelos de datos para soportar estos análisis requieren un enfoque diferente al usado en los sistemas operacionales. Básicamente se comienza con una matriz donde se determina la dimensionalidad de cada indicador y luego se especifican los diferentes grados de detalle (*atributos*) dentro de cada concepto del negocio (*dimensión*), así también la granularidad de cada indicador (*variable o métrica*) y las diferentes jerarquías que dan forma al modelo dimensional del negocio o mapa dimensional.

Supongamos que una cadena de tiendas de refacciones quiere crear un DWH para obtener un mejor análisis de sus datos y mejorar su condición en el mercado. Es necesario hacer un análisis muy detallado del concepto de la tienda, el primer paso para esto es crear una matriz en donde capturará y determinará los puntos más importantes para determinar la dimensión de sus datos, los atributos necesarios y las relaciones entre ellas.

La figura 4.2 muestra una matriz tridimensional, la cual determina la dimensionalidad de cada indicador y el modelo dimensional del negocio ilustrando las diferentes jerarquías dentro de cada dimensión en una tienda de refacciones.

Nuestro diseño del warehouse está basado en el diseño conceptual Entidad-Relación(E-R), diseño lógico en la teoría del modelado multidimensional.

	Tiempos	Productos	Ubicación
1. Matriz			
2. Concepto del negocio (dimensión)			
3. Atributos (grados de detalle)			
4. Jerarquías (relaciones)			
5. Granularidad de cada indicador (variable o hechos)			
Monto de venta	✓	✓	
Cantidad de venta	✓	✓	✓
Presupuesto Cantidad	✓		✓
Descuento Monto	✓	✓	
Ganancia	✓	✓	
Monto	✓	✓	

Figura 4.2: Matriz para determinar los requerimientos

A continuación se muestra un esquema de una tabla de hechos y sus dimensiones (Fig. 4.3). Podemos observar como el diseño lógico se torna a 3 dimensiones ejemplificando la multidimensionalidad.

El ejemplo (Fig. 4.3) hace referencia a como la tabla de hechos (ventas) está compuesta básicamente de la *dimensión tiempo*, *dimensión producto* y *dimensión ubicación*. Estas dimensiones a su vez están relacionadas por diversas tablas, por ejemplo la dimensión producto tiene la tabla artículo y donde el ID_artículo hace relación con la tabla categoría. La dimensión ubicación hace referencia a la tabla tienda y el id_tienda se relaciona con la tabla ciudad donde también esta hace relación con la tabla región. Como se puede ver conservan parte del modelo E-R al llevar relación las tablas unas con otras, pero conceptualizando el modelo multidimensional. Así se obtiene una tridimensionalidad en la tabla de hechos ventas. Este diseño fue modelado en el ambiente gráfico de PostgreSQL para Windows XP.

En la fig. 4.4 se muestran las tablas creadas para nuestro DWH, éstas fueron desarrolladas en el ambiente gráfico que proporciona postgresQL para Windows. Esta aplicación ayuda en forma visual para la creación de tablas, funciones, triggers y BD's.

4.4. Arquitectura

DWHA esta basado en el diseño lógico del modelo multidimensional en donde se tienen desarrolladas 4 tablas de hechos en donde cada una tiene diferentes dimensiones según los requerimientos

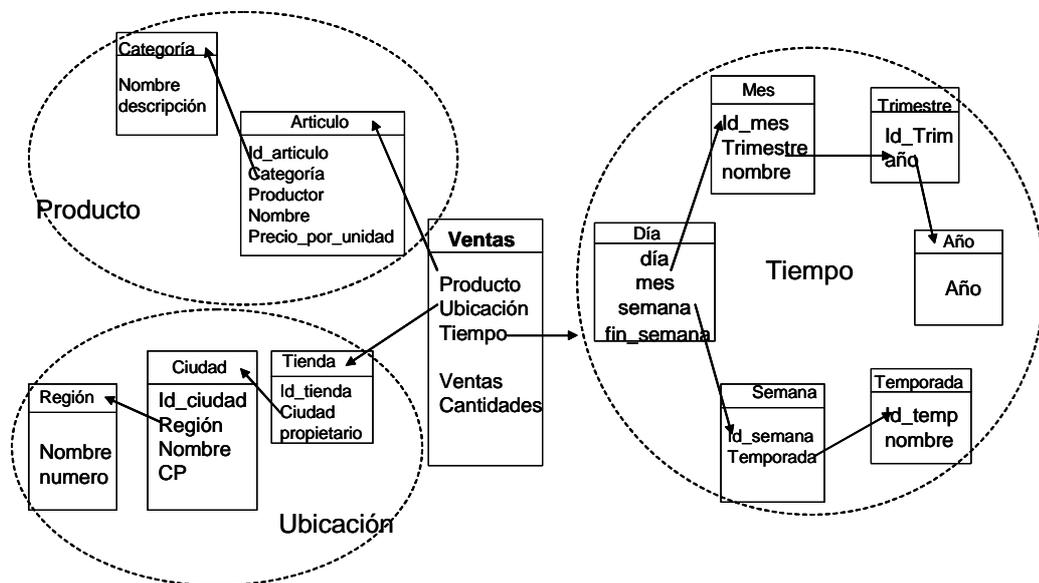


Figura 4.3: Dimensión de tabla de hechos ventas

de éstas.

Como dato, se muestra las dimensiones de la tabla de hechos **Ventas** para el DWHA (Fig. 4.5).

Dentro de este modelo se tienen conceptualizadas las reglas de análisis. Un punto muy importante es el uso de más datos genéricos en el modelo multidimensional desarrollado, ya que optimizará las consultas y agilizará los resultados de una forma mucho más eficiente y rápida, gracias al poco uso de memoria para cada dato. Para esto se deben crear catálogos que hagan referencia a cada registro, en donde el uso de este tipo de dato pueda proporcionar un esquema más detallado de nuestros datos (catálogos), además las operaciones serán más concisas y más eficientes.

4.5. Ambiente de desarrollo

La interfaz gráfica del DWHA se desarrolló con el lenguaje de programación orientado a objetos (POO) Java. Se decidió usar este lenguaje debido a la amplia variedad de librerías para gráficos y la funcionalidad que da a cada elemento como objeto.

Otra particularidad es la portabilidad que ofrece para ejecutar los programas en diferentes

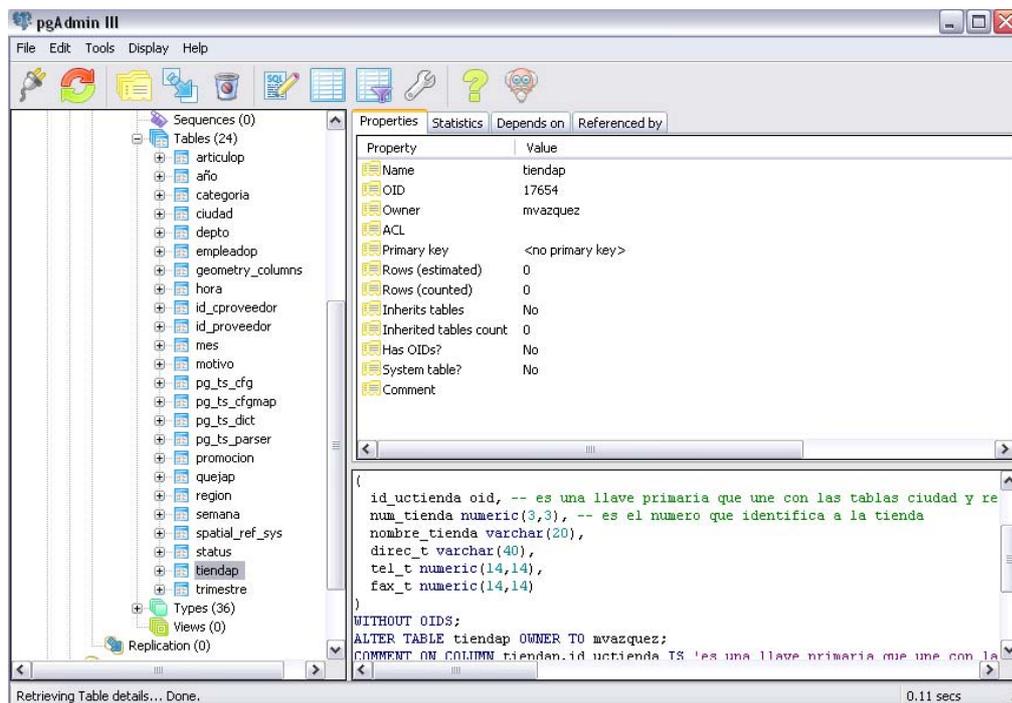


Figura 4.4: Administrador de postgres con las referencias de las tablas multidimensionales.

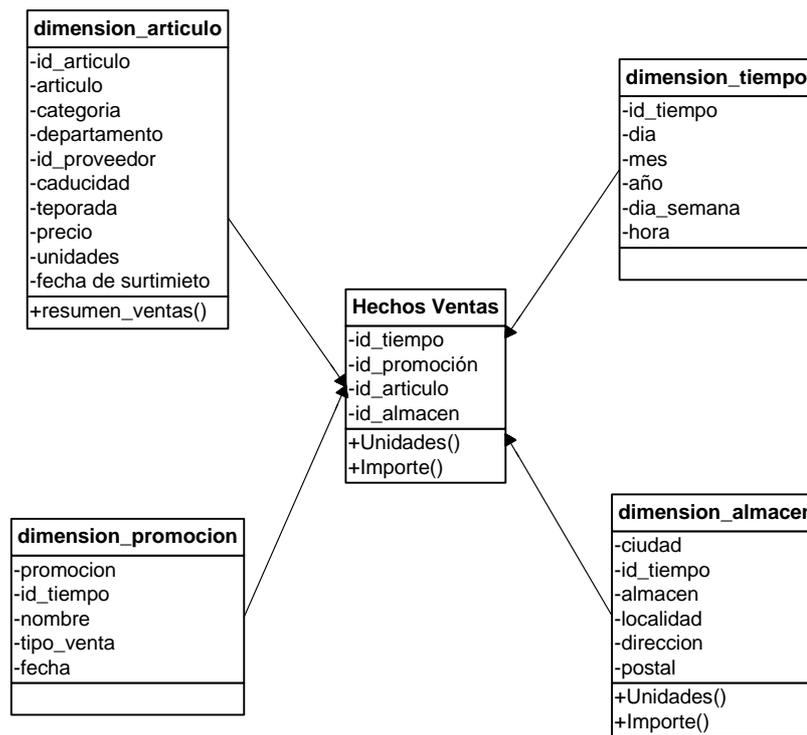


Figura 4.5: Gráfica de esquema de tabla de Hechos Ventas.

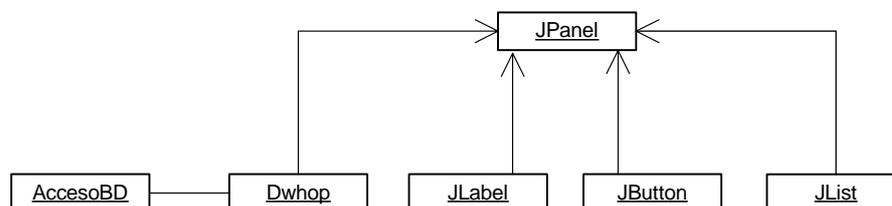


Figura 4.6: Diagrama de clases del DWHA.

plataformas; además postgres ofrece una librería que puede ser aplicada en Java llamada PL/Java el cual te permite explotar los recursos de java para obtener mejor resultados en la conexión java-postgresql.

Cabe mencionar que el DWHA puede migrar a otras BD's que implementen el uso de triggers como por ejemplo Oracle o Firebird ya que también ofrecen el Lenguaje procedimental e implementan el uso de triggers, solo hay que recordar que las sintaxis de estas varían de una a otra e incluso el DWHA puede ser implementada en otras plataformas de Sistemas Operativos como Unix y Mac.

4.6. Diagrama de clases

En POO se utilizan los conceptos de herencia y composición. En el diseño del diagrama de clases, la herencia se representa con una flecha que conecta a la clase hija con la superclase, la clase hija hereda atributos y métodos de la superclase. Gráficamente la clase hija se coloca por debajo de la superclase y la flecha llega a la parte inferior de la superclase. Una clase es una composición de otra cuando forma parte de sus atributos. La composición se representa con una flecha que parte de la clase atributo hacia el lado izquierdo de la clase que la contiene. El diagrama de las clases que se utilizaron en la implementación de DWHA se representan en la figura 4.6.

La clase principal es *Dwhop* donde esta es una extensión de *JPanel* y es una composición de la clase *Acceso BD* que es la encargada de generar la conexión a la BD la cual contiene tres métodos, *creaFuction*, *ejecutaSQL* y *columnName*.

El método *creaFuction* es el encargado de crear una función en la base de datos generando la regla de análisis.

El método *ejecutaSQL* se utiliza para realizar una consulta y retorna una lista que se utiliza

para desplegar las tablas de la BD.

El método *columnName* retorna como resultado el nombre de los campos de una tabla específica.

La interfaz gráfica está compuesta por etiquetas, botones y listas. La clase *JLabel* implementa todas las características inherentes a los componentes Swing, como pueden ser los aceleradores de teclado, bordes, y etiquetas. *JButton* descende de *AbstractButton* define un pequeño API para activar los botones.

En las listas se almacenaron las tablas del sistema de BD a los nombres de los campos de estas utilizando el método *ejecutaSQL* y *columnName* según sea el caso.

JPanel es el panel principal que sirve como contenedor de las clases antes mencionadas.

Por último el método *creaFuction* recibe como parámetros los valores de los objetos gráficos (los nombres de las tablas, columnas, condiciones, etc.) para crear una regla de análisis.

4.7. Diseño de la Interfaz Gráfica.

DWHA ofrece una interfaz gráfica en donde se despliegan automáticamente las tablas generadas en el DWH y en otra lista se proporciona los campos de dicha tabla y así poder escoger el o los campos requeridos para la creación de las reglas de análisis, una vez seleccionadas las tablas y campos requeridos se da la opción de escoger que tipo de operación relacional se necesita y por último se activa un campo para proporcionar la condición que deberá cumplir para generar un evento (Fig. 4.7).

En la figura 4.7 se muestra la interfaz DWHA creado para dar soporte al desarrollo de las reglas de análisis. La aplicación está compuesta de 6 listas y de un campo para introducir valores:

1. *lista de tablas A*, es la que despliega las tablas de la BD desarrolladas en postgresQL,
2. *lista de campos A*, despliega los campos de la tabla seleccionada en **1**.
3. *lista de operadores del álgebra de eventos*, tiene como operadores *And*, *Or*, *Simultaneo*, *Secuencia*, *Negación*, *Primero*, *Último* y *Alguno* descritos en el capítulo 3 de esta tesis.
4. *lista de tablas B*, enlista también las tablas de la BD desarrolladas en el DWH, pero éstas se activan, si se hace uso de algún operador del álgebra de eventos es decir si se hace selección de algún operador de la lista **3**.

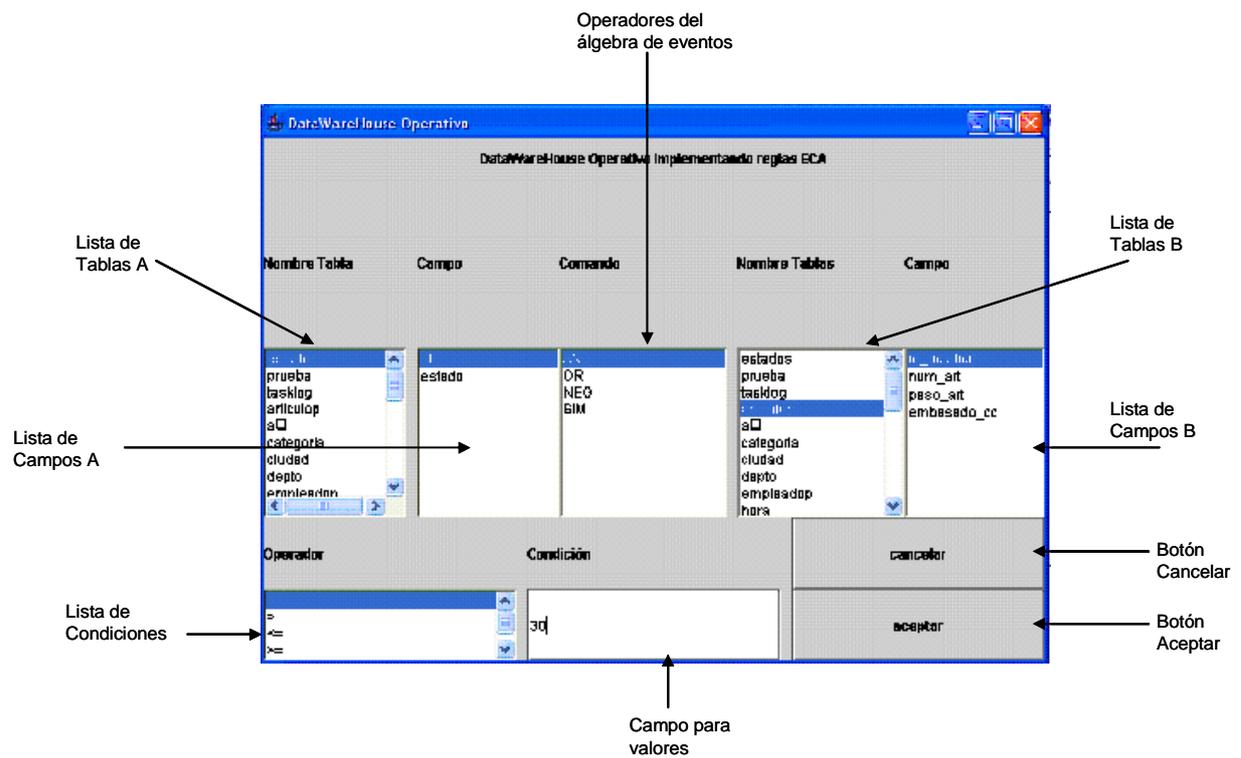


Figura 4.7: Interfaz DWHA para desarrollo de reglas de análisis.

5. *lista de campos B*, esta se activa en función de la lista de tablas B arrojando los campos de dicha tabla seleccionada.
6. *lista de condiciones*, muestra las condiciones de =, <=, =>, <>, <, > para crear condiciones en base algún valor a introducir en *campo para valores*.
7. *campo para valores*, es para introducir algún valor ya sea numérico o texto, es aquí donde se especifica los umbrales.

Capítulo 5

Reglas ECA Fuzzy para modelar la Toma de Decisiones.

La Toma de Decisiones es un área que está relacionada con gran cantidad de tareas y procesos que las personas realizan de forma habitual en sus tareas cotidianas. De esta gran variedad de campos nos centraremos en el contexto de las interrupciones de abastecimiento de una tienda comercial. Frente a esta problemática, nos encontramos con una gran cantidad de información dada por un gran número de fuentes de abastecimiento y en la mayoría de los casos valorando aspectos cualitativos donde la información referida a los mismos presenta un alto grado de incertidumbre. Debido a este grado de incertidumbre en la información el modelo de representación lingüístico anudado con el concepto de reglas ECA, puede verse como una solución adecuada para representar la información a problemas del contexto del sector comercial que mejoren este tipo de información vaga e imprecisa.

5.1. Funciones para modelar recursión ECA.

Las BD's se usan a través de los llamados sistemas de gestión de bases de datos (SGBD), en el cual haciendo uso de PostgreSQL, se pueden desarrollar funciones en varios lenguajes. En el lenguaje PL/pgSQL se desarrollaron nuestras funciones debido al soporte que brinda para desarrollar funciones complejas.

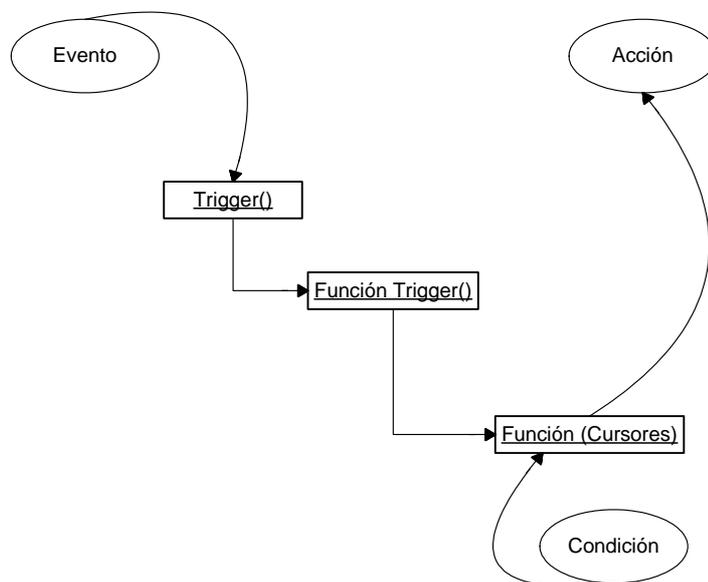


Figura 5.1: Diagrama de comportamiento de reglas de análisis dentro de la interface.

Se desarrollaron tres funciones básicas para crear la automatización de las reglas, de las cuales estas se encuentran anidadas, generando recursividad en dichas funciones.

Cada función desempeña un papel primordial en el modelamiento de las reglas ECA.

El algoritmo propuesto lo podemos ejemplificar en la figura 5.1 donde se muestra la secuencia de nuestras funciones tornándose en el principio de una regla ECA.

La figura 5.1 podemos ver que se tiene los tres componentes básicos de una regla ECA en donde nuestro evento es modelado y especificado en un trigger, que a la vez hace referencia a una funcion_trigger, en donde esta es la que encapsula y hace llamado a otra función que contiene los cursores encargados de almacenar los resultados de las condiciones para así dar resultado a dicha petición(Acción).

Se desarrollaron funciones usando cursores, para poder hacer uso de varias tablas a la vez e implementar con ellas operaciones complejas que las involucre, por lo tanto se propuso un algoritmo genérico para cada tabla de hechos.

En la figura 5.2 se describe el diagrama de estado de una función cursor como una función toma

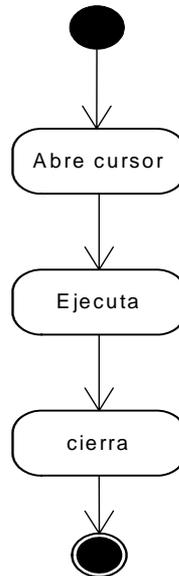


Figura 5.2: Diagrama de Estado de una función cursor.

como argumento un cursor para realizar alguna sentencia. Una vez que el cursor sea tomado este ser abierto para ejecutar alguna consulta que cumpla alguna condición, y ya abierta puede ser usada y manipulada según los requerimientos de la sentencia; ya concluido esto, debe cerrarse todos los cursores para así liberar la memoria que consume cada cursor usada para tal propósito.

Algoritmo para la modelación de las funciones recursivas. El siguiente algoritmo tiene el objetivo de consultar las condiciones y acciones a realizar por cada evento generado a partir del trigger propuesto.

Algoritmo 5.1 *CREATE OR REPLACE FUNCTION disyuncion(refcursor, refcursor,..., argumentos...)*

```

  RETURNS SETOF refcursor AS
  declare
  argumento;
  BEGIN

```

```

OPEN cursor FOR EXECUTE 'query' OR 'query' OR 'n....' ;
FETCH cursor INTO argumento;
inicio ciclo si true
insert into tabla_resumen values (argumentos);
FETCH cursor INTO argumentos;
END ciclo;
CLOSE cursor;
query;
RETURN NEXT cursor;
OPEN cursor2 .....;
RETURN NEXT cursor2;
.....
END;
LANGUAGE 'plpgsql'

```

Una vez creado el anterior algoritmo se debe hacer llamar en otra función para poder desde allí introducir los argumentos necesarios, para la creación de las reglas de análisis.

Algoritmo 5.2 *funcion ()*

```

Open cursor FOR EXECUTE ' SELECT * FROM disyuncion("argumentos", "argumentos", "...", "....")';
fetch cursor in qry;
RETURN cursor;
close cursor;
RETURN;
END;

```

Para poder ejecutar estas funciones automáticas se debe crear una función trigger que será la encargada de automatizar nuestro DWH.

Algoritmo 5.3 *CREATE OR REPLACE FUNCTION funcion_trigger()*

```

RETURNS trigger AS
declare

```

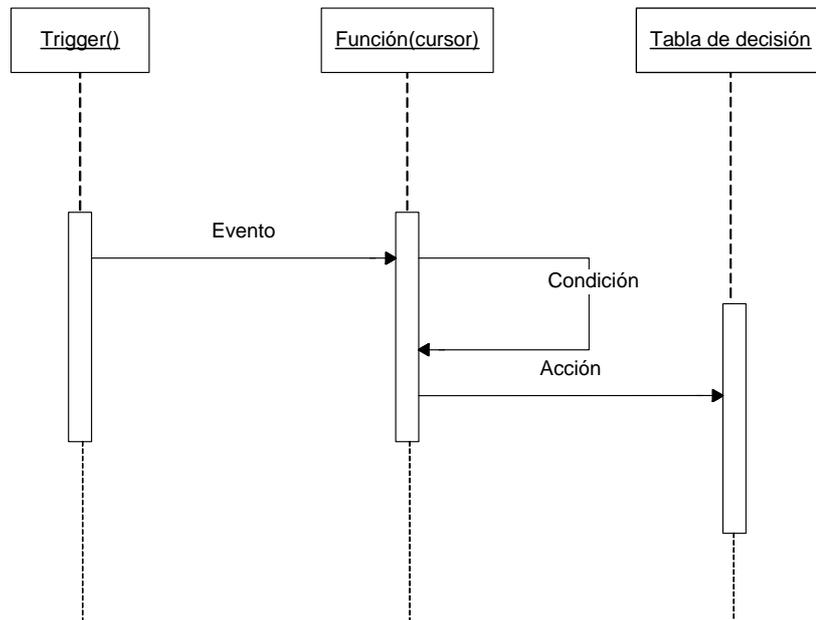


Figura 5.3: Secuencia ECA de funciones triggers para crear actividad en un DWH.

variables ;

INICIO

Llamada a la función();

execute qry;

return;

FIN;

Como último paso al crear un trigger hay que hacer un llamado a una función_trigger para que esta sea ejecutada cada vez que detecte un evento capaz de generar alguna acción, ya que un trigger no es capaz de recibir como argumentos a los cursores, se tiene que encapsular en otra función (Fig. 5.3). En consecuencia se puede ver como un trigger hace llamado a una función trigger encapsulando las condiciones y las acciones a realizar según sea definido.

Una vez modelado las funciones triggers, la implementación de las reglas de análisis es el siguiente paso para la automatización de la toma de decisión con reglas ECA Fuzzy.

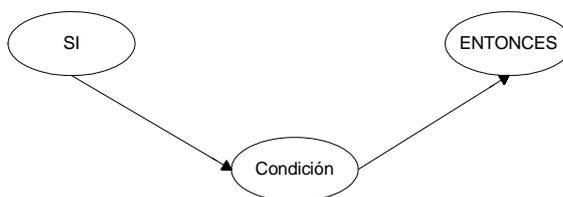


Figura 5.4: Esquema general de una regla fuzzy.

5.2. Reglas ECA Fuzzy

La lógica *fuzzy* es entonces definida como un sistema matemático que modela funciones no lineales, que convierte unas entradas en salidas acordes con los planteamientos lógicos que usan el razonamiento aproximado.

Se fundamenta en los denominados conjuntos *fuzzy* y un sistema de inferencia *fuzzy* basado en reglas de la forma “SI..... ENTONCES..... ” (Fig. 5.4), donde los valores lingüísticos de la premisa y el consecuente están definidos por conjuntos *fuzzy*, es así como las reglas siempre convierten un conjunto *fuzzy* en otro.

Clasificación de los Conjuntos Fuzzy

Para atacar el problema del “desabastecimiento de artículos”, fuera del período de resurtido, los conjuntos fuzzy nos ayudaran a construir mejores modelos de la realidad.

Un primer paso es definir el objetivo del sistema, el cual es determinar y apoyar la toma de decisión, al planteamiento del problema de “desabasto”.

Hay que determinar a que proveedor acudir en ciertos casos, a un proveedor mayorista o proveedor minorista, los proveedores mayoristas tienen ya programadas sus fechas de entrega de pedidos a sus clientes y muchas de las veces sus entregas son en vehiculos voluminosos como trailers y dependen de estos para su entrega. En situaciones de emergencia no se disponen de estos o no tienen fechas disponibles para su entrega. En cambio los proveedores minoristas por su perfil de entrega muchas de las veces son más flexibles en estas situaciones e incluso en su ubicación territorial ya que tienden a ubicarse en localidades estratégicas.

Ya definido el objetivo a modelar, hay que identificar las variables de entrada y salida, que son

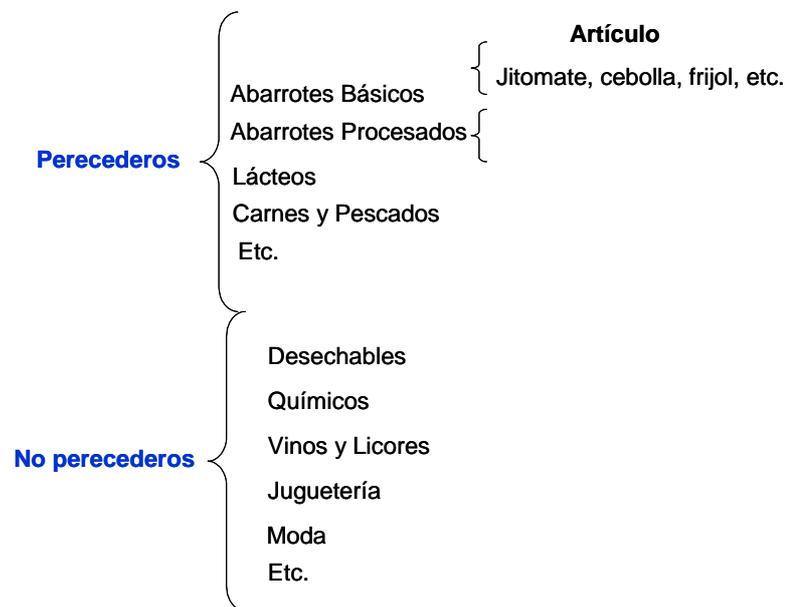


Figura 5.5: Variables lingüísticas

las variables y las etiquetas lingüísticas. Estas variables surgen de la clasificación adecuada de los artículos en conjuntos fuzzy. En la figura 5.5 se hace una clasificación de los artículos, según el nivel de necesidad en *perecederos* o *no perecederos*, donde éstas darán lugar a etiquetas lingüísticas.

En la parte del diseño se definen los conjuntos *fuzzy* de cada etiqueta lingüística es aquí donde se expresa el grado de pertenencia al conjunto que tiene cada uno de los elementos. Por ejemplo definimos los conjuntos:

$$(A) \text{Perecederos} = \{\text{abarrotes_básicos}(A_1), \text{abarrotes_procesados}(A_2), \text{lácteos}(A_3), \text{carnes y pescados}(A_4)\}$$

$$(B) \text{No perecederos} = \{\text{desechables}(B_1), \text{químicos}(B_2), \text{vinos_licores}(B_3), \text{juguetería}(B_4), \text{moda}(B_5), \text{plásticos}(B_6), \text{vidrio}(B_7)\}$$

$$A_1 = \{\text{jitomate}, \text{cebolla}, \text{papa}, \text{tomate}, \text{etc}\}$$

$$A_2 = \{\text{atùn}, \text{chiles_vinagre}, \text{mayonesa}, \text{sardina}, \text{etc}\}$$

$$A_3 = \{\text{leche}, \text{queso_manchego}, \text{queso_fresco}, \text{mantequilla}, \text{etc}\}$$

$$A_4 = \{\text{pollo_entero}, \text{pechuga_pollo}, \text{bistec}, \text{carne_molida}, \text{etc}\}$$

$$B_1 = \{\text{papel_higienico}, \text{servilletas}, \text{platos}, \text{vasos}, \text{tenedor}, \text{cuchara}, \text{etc}\}$$

$B_2 = \{cloro, jabón_trastes, jabón_corporal, desodorantes, etc.\}$
etc.

Cada conjunto clasifica los artículos en relaciones comunes entre ellos, para así poder definirle etiquetas lingüísticas, según el grado de membresía al que representa sus valores numéricos. Existen diferentes parámetros a modelar en la definición de etiquetas, una de ellas es la frecuencia de venta de cada artículo adquirido por los clientes.

Hay que obtener un promedio de ventas diarias de cada artículo, en donde asumimos que el artículo jitomate, tiene una frecuencia de venta alta ya que se encuentra dentro del conjunto abarrotes básicos y es un artículo de mucha afluencia por los consumidores.

Para esto definimos el grado de membresía $\mu_a(x)$ en tres rangos y en consecuencia se le asignan tres etiquetas lingüísticas al parámetro “Frecuencia de venta” como lo muestra la tabla 5.1.

Etiqueta Lingüística	Grado de membresía
Alta	1
Media	0.5
Baja	0

Tabla 5.1. Grado de membresía parámetro “Frecuencia de venta”.

La frecuencia surge de hacer un comparativo de ventas diarias en relación con su propio conjunto *fuzzy* y así tener un estimado de las ventas.

Asumimos que todos los abarrotes básicos que pertenecen al conjunto de perecederos tienen mayor frecuencia de venta, pero en la realidad no todos los artículos básicos tienen la misma frecuencia de venta.

En la figura 5.6 se ilustra la jerarquía de la variable lingüística “frecuencia de venta”. Las etiquetas lingüísticas ‘alta’, ‘media’, ‘baja’, se tornan como una función triangular en donde hay superposición entre ellos, el cuál parametriza a los artículos según la cantidad de ventas que este tenga en la tienda en un cierto periodo. Los grados de membresía se limitan entre 0 y 1, en donde el mínimo valor de venta se encuentra entre 0.5 y 1 como rangos límites y del 0 a 0.5 como límites de mayor venta.

Otro factor importante a definir es el grado de membresía parámetro “porcentaje de existencia” de artículos en la tienda, con esto se le asigna una calificación o valor lingüístico, necesario para

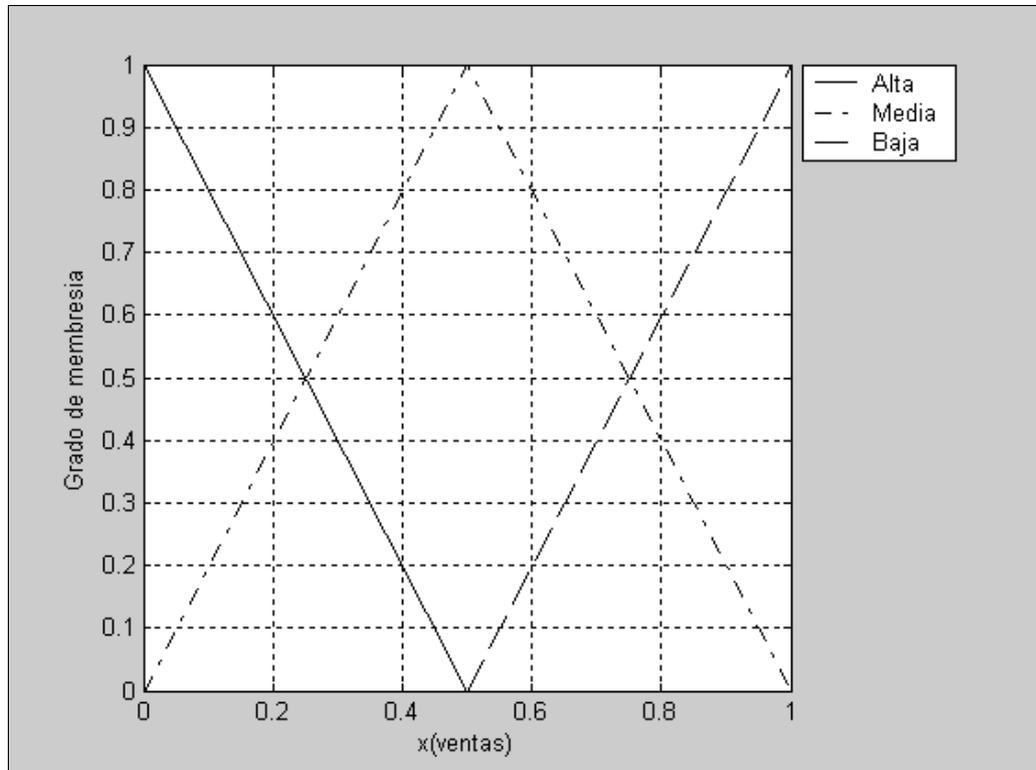


Figura 5.6: Grado de membresía parámetro "ventas".

saber la urgencia de abastecer el almacén para su venta pronta y dar un mejor servicio al cliente y saber con quien acudir, a un proveedor mayorista o minorista.

El umbral es un requisito indispensable para poder asignarle un grado de pertenencia (ver tabla 5.2), ejemplo si las existencias están en un factor de 0.2 el valor lingüístico indica que el resultado es muy urgente.

Existencias	Grado de membresía	Etiqueta lingüística
100 %	1	totalmente completo
80 %	0.8	semi completo
60 %	0.6	moderado
40 %	0.4	levemente agotado
20 %	0.2	semi agotado
0 %	0.01	agotado

Tabla 5.2. Grado de membresía parámetro “porcentaje de existencias”.

En la figura 5.7 se muestra como se limitan los grados de membresía parámetro ‘existencias’, en el cual calificamos las existencias de los articulos dentro de la tienda en etiquetas que detallan la cantidad de la mercancia de dicho articulo, por ejemplo la etiqueta ‘Totalmente completo’ abarca los grados de membresía x (existencias) entre el valor 0.6 y 1 en forma degradante(limites) donde el valor 0.8 corresponde a un grado de referencia.

Otro factor importante es el *tiempo* y se debe asignar también un grado de pertenencia (ver tabla 5.3), ejemplo si los atunes presentan un valor lingüístico de “muy urgente” parámetro existencias, pero presenta “nulo” su valor lingüístico en días faltantes a resurtir, entonces indica que no es necesario un resurtimiento con un proveedor minorista, sino con el proveedor mayorista, ya que indica que fue una venta normal.

Días faltantes para resurtir	Grado de membresía	Etiqueta lingüístico
100 %	1	muy urgente
80 %	0.8	urgente
60 %	0.6	moderado
40 %	0.4	leve
20 %	0.2	nulo

Tabla 5.3. Grado de membresía parámetro “días faltantes” para resurtir anaquel.

Se ilustra en la figura 5.8 las variables lingüística “tiempo faltante”. En donde en las etiquetas lingüísticas “muy urgente”, “urgente”, “moderado”, “leve”, “nulo”, hay superposición y toman la

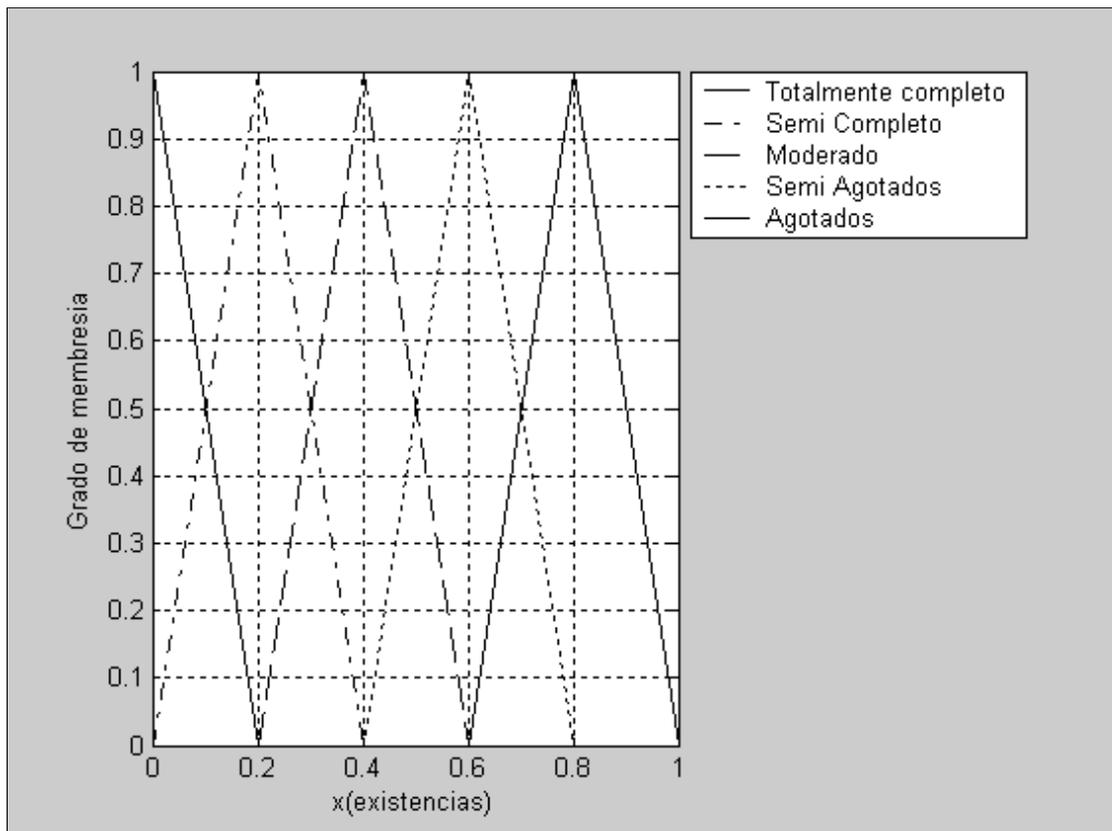


Figura 5.7: Grados de membresía parámetro “existencias”.

función triangular; siendo el rango de 0 a 0.2 el que le corresponda a la etiqueta ‘Muy urgente’ indicandonos que el artículo ya debe ser surtido, puesto que el tiempo programado para abastecer almacén ya se agotó; y el rango que nos indica lo contrario se encuentra entre el valor 0.6 y 1 como límites de la etiqueta ‘Nulo’.

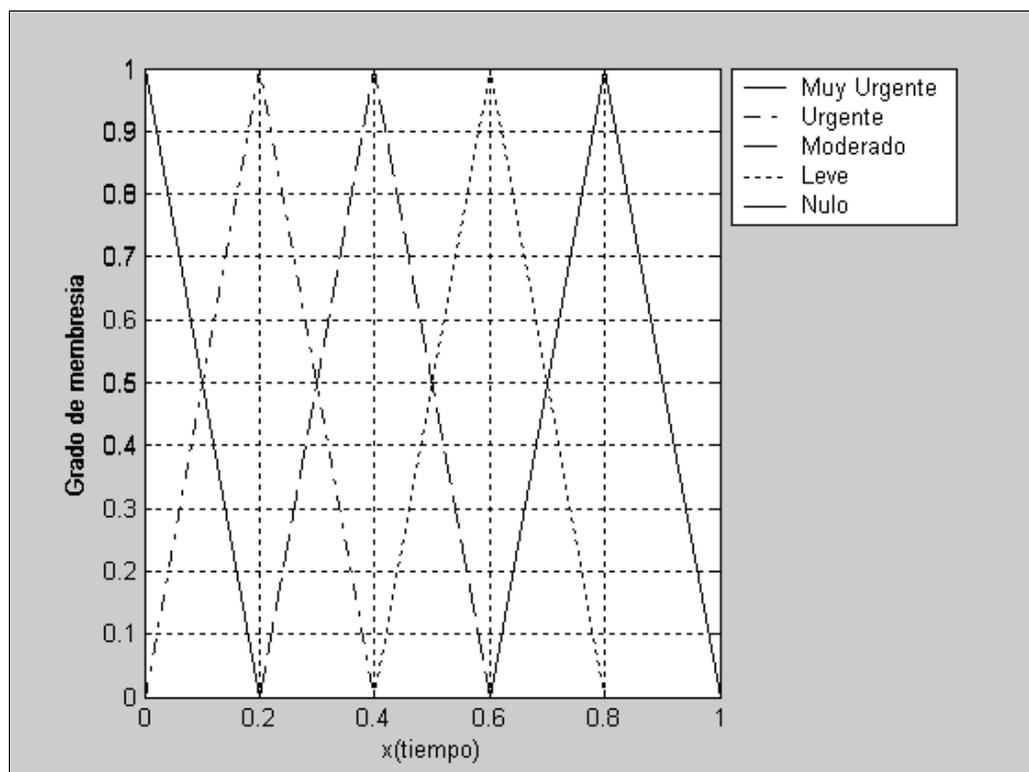


Figura 5.8: Grados de membresías parámetro “tiempo faltante para resurtir”.

La ubicación geográfica es considerado también otro factor importante, ya que ¿el donde? se encuentre el proveedor, dará un parámetro de costos por cobrar, debido a los gastos que implican el traslado de la mercancía a proveer. En la tabla 5.4 se detallan los parámetros que le corresponden a cada etiqueta lingüística. Para poder parametrizar este factor distancia, fue necesario crear límites y kilometrajes, en donde el uso del código postal nos dió estas características, ya que cada poblado, ciudad o municipio le corresponde un código postal diferente y el primer dígito de los 5

que conforman el código postal nos dá un estimado de la distancia, que hay entre la ubicación de la tienda con el proveedor si este se encuentra en otro estado y el segundo digito nos dá este mismo estimado pero si la ubicación fuera en el mismo estado.

Para asignarle un grado de membresía a este factor, fue necesario delimitar codigos postales cerca de la ubicación de la tienda a esta circunferencia se le asignó la etiqueta “Muy cerca” (0.2) y conforme se alejaba de la tienda le corresponde el siguiente grado y así sucesivamente.

Ubicación del proveedor	Grado de membresía	Etiqueta lingüística
100 %	1	Muy Lejos
80 %	0.8	Lejos
60 %	0.6	Media
40 %	0.4	Cerca
20 %	0.2	Muy Cerca

Tabla 5.4. Grado de membresía parámetro “ubicación geográfica” del proveedor.

La figura 5.9 nos muestra las variables lingüísticas del parámetro “distancia”. En donde podemos visualizar que los rangos de la etiqueta “Muy cerca” le corresponde los valores de 0 y 0.2, el cual nos asume que los codigos postales que tomen alguno de esos valores se encuentran relativamente muy cerca de la tienda a surtir. En lo contrario, los códigos postales que se encuentren fuera del diametro cercano a la tienda, se les signa valores correspondientes a lo relativo de la lejanía, siendo el valor más lejano el que se encuentra en el rango de 0.6 a 1.

Así como la ubicación geográfica es considerado importante, también otro factor importante es lo complejo del traslado de la mercancía a surtir, ya que no es lo mismo transportar abarrotes básicos que vidrio, o alimentos refrigerados, entre varios ejemplos que podemos citar, pero en definitiva el trasladar un pedido de un lugar a otro, es necesario de un medio de transporte, en donde influye la distancia entre la tienda y el proveedor, así como que tipo de artículo se va a transportar. Para esto fue necesario obtener varios factores a parametrizar, que va desde que tipo de transporte se usará y el tonelaje de la mercancía y su volumen.

Estos parámetros nos dió el estimado de nuestras etiquetas lingüísticas, para el cual se consideró “Muy complejo” el traslado de aquellos artículos que tuvieran un volumen grande o tonelaje

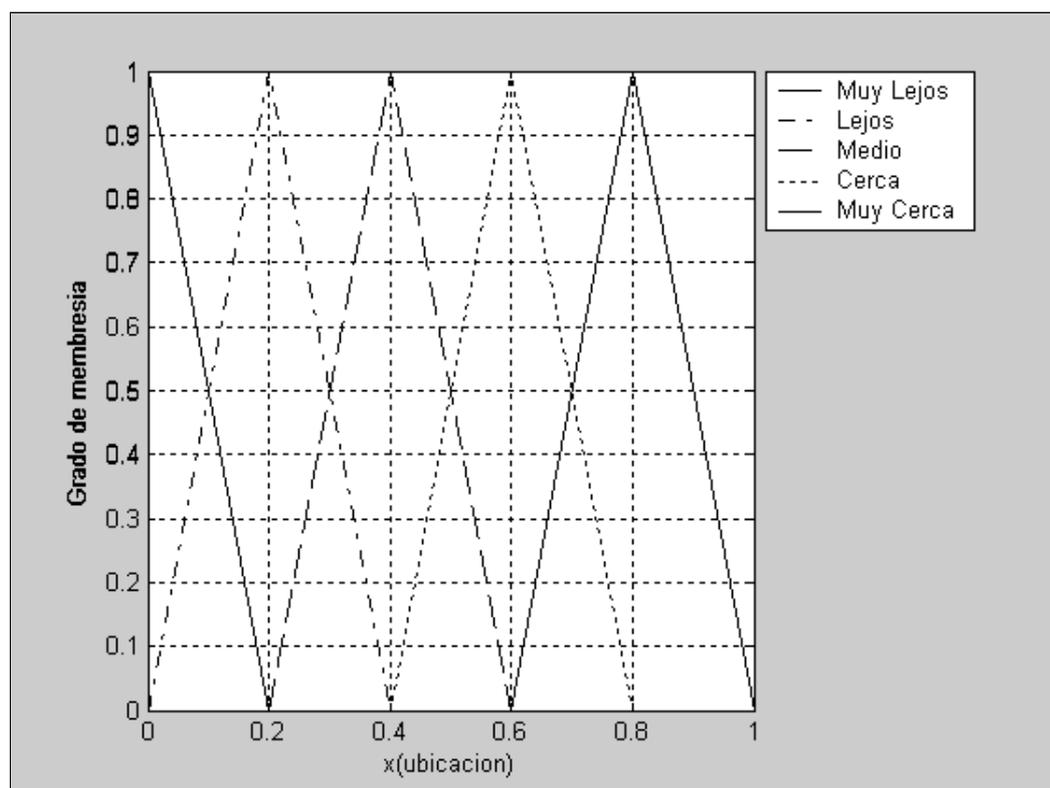


Figura 5.9: Grados de membresía parámetro "ubicación geográfica" de los proveedores.

excesivo y que necesitaran de un transporte muy pesado(trairlers) o con equipamiento especial como refrigeración.

Ubicación del proveedor	Grado de membresía	Etiqueta lingüística
100 %	1	Muy Complejo
80 %	0.8	Complejo
60 %	0.6	Moderado
40 %	0.4	No Muy Complejo
20 %	0.2	No complejo

Tabla 5.5. Grado de pertenencia parámetro “dificultad de traslado” de la mercancía.

En la figura 5.10 se gráfica los parámetros y limites entre cada grado de membresía definida por nuestras etiquetas lingüísticas. Siendo los valores encontrados entre el rango de 0 y 0.2 considerados como muy compleja su dificultad de traslado y considerado lo contrario, aquellos valores que se encuentren entre el rango de 0.6 y 1.

De los dos factores anteriores (ubicación geográfica y dificultad de traslado) surge este último parámetro llamado “costo”, y es una consecuencia de la distancia que hay entre el proveedor y la tienda así como también de lo complejo de la mercancía a trasladar, ya que entre más lejos se encuentre el proveedor de la tienda a surtir, más combustible se usará para transportar la mercancía, entre otros factores a considerar como inclemencias, casetas si fuese necesario y honorarios al conductor del transporte entre otros posibles factores. Si la mercancía a transportar es considerado complejo el traslado eso involucre posiblemente un vehículo de un costo más excesivo que si fuese una camioneta por ejemplo, entonces ambos factores crearan que los costos se incrementen o en lo contrario que lo minimice. En la tabla 5.6 se detallan los grados correspondientes a cada etiqueta lingüística.

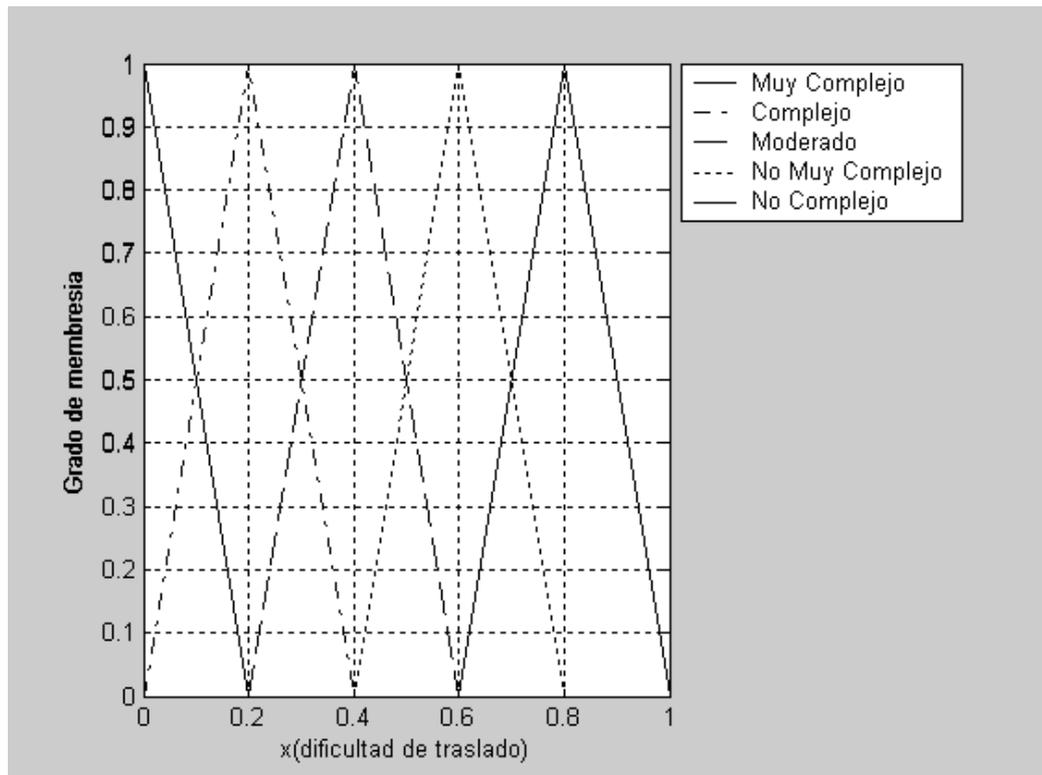


Figura 5.10: Grados de membresía parámetro "dificultad de traslado" de la mercancía.

Costos del proveedor	Grado de membresía	Etiqueta lingüística
100 %	1	Muy Excesivo
80 %	0.8	Excesivo
60 %	0.6	Considerable
40 %	0.4	Barato
20 %	0.2	Muy Barato

Tabla 5.6. Grado de pertenencia parámetro “costo” de la mercancía.

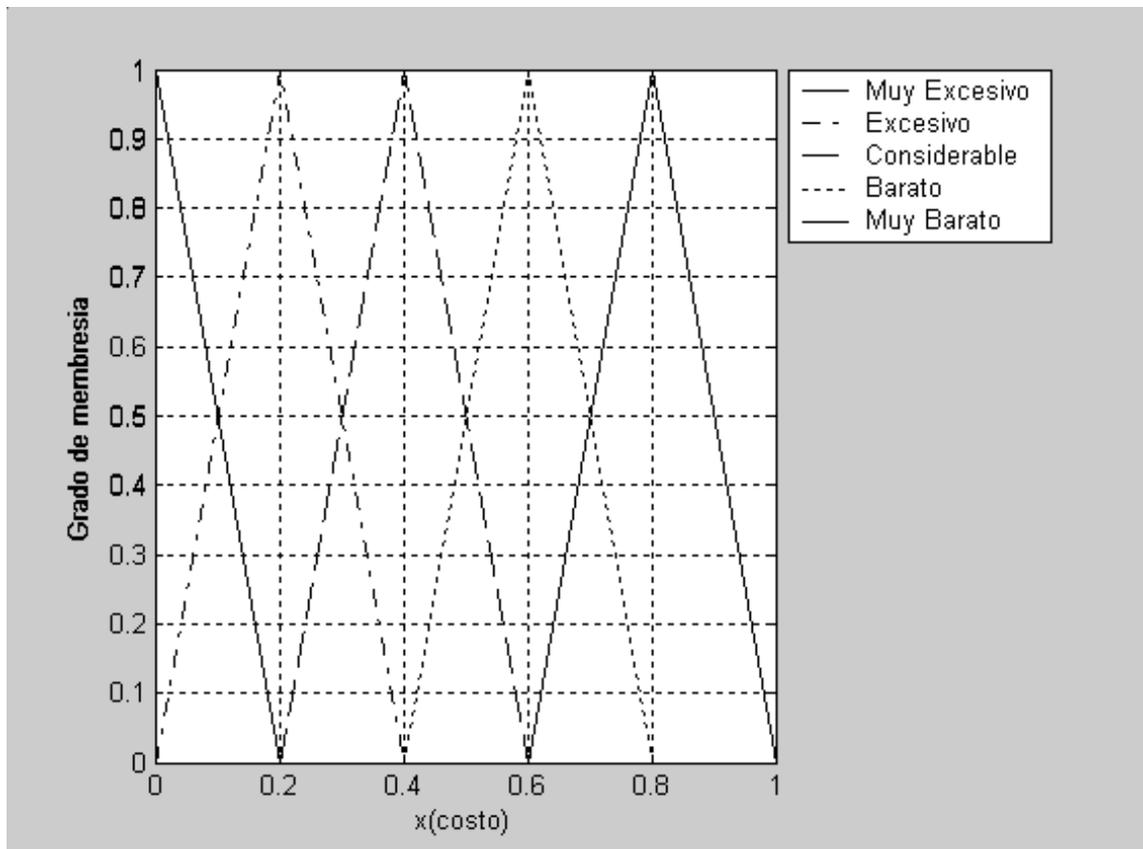


Figura 5.11: Grados de membresía parámetro “ubicación de la mercancía” y “dificultad de traslado”.

Podemos visualizar que en la figura 5.11 se gráfica los parámetros y límites existentes entre cada grado de membresía definida por nuestras etiquetas lingüísticas. Como ya se mencionó en el párrafo anterior, esta gráfica surge de una combinación de los factores ubicación geográfica y dificultad de traslado, siendo los más altos rango considerados los que le dan la etiqueta de un costo muy excesivo por surtir cierta mercancía.

Los factores de pertenencia sirven para definir el marco de representación de la solución y así dar una base de conocimiento mediante reglas *fuzzy*.

Teniendo bien especificado el objetivo a tratar, los conjunto *fuzzy*, sus grados de pertenencia y los valores lingüísticos, la definición de las reglas requiere de un antecedente o lado izquierdo *SI* y un consecuente o lado derecho *ENTONCES*, ejemplo:

SI jitomate es perecedero, ENTONCES frecuencia es ALTA

SI existencias es agotado Y tiempo muy urgente ENTONCES proveedor_minorista.

SI jitomate es perecedero, ENTONCES frecuencia es ALTA

SI existencias es semi completo Y tiempo leve ENTONCES proveedor_mayorista.

La definición de las reglas, son la base de conocimiento del sistema para la toma de decisión, encargada de establecer los criterios para la asignación de si recurrir a proveedores minoristas o mayoristas.

5.3. Modelado para la toma de decisiones

Parte fundamental en la especificación del comportamiento reactivo es la definición de una base de reglas ECA Fuzzy, que proporcionen los mecanismos de reacción al DSS. En la figura 5.12 se presenta un modelo conceptual del STDE_ECAF. La *Base de conocimiento* contiene el conocimiento asociado al dominio de la aplicación y los objetivos de control, está formada por una base de datos y un conjunto de reglas ECA fuzzy de la forma:

SI (X_1 es A_1) y (X_2 es A_2) *ENTONCES* (Y es B)

donde

X_i son las variables de estado del sistema a controlar.

Y es una variable de control del sistema a controlar.

Los A_i y B son etiquetas lingüísticas con una función de pertenencia asociada que dan valor a sus variables.

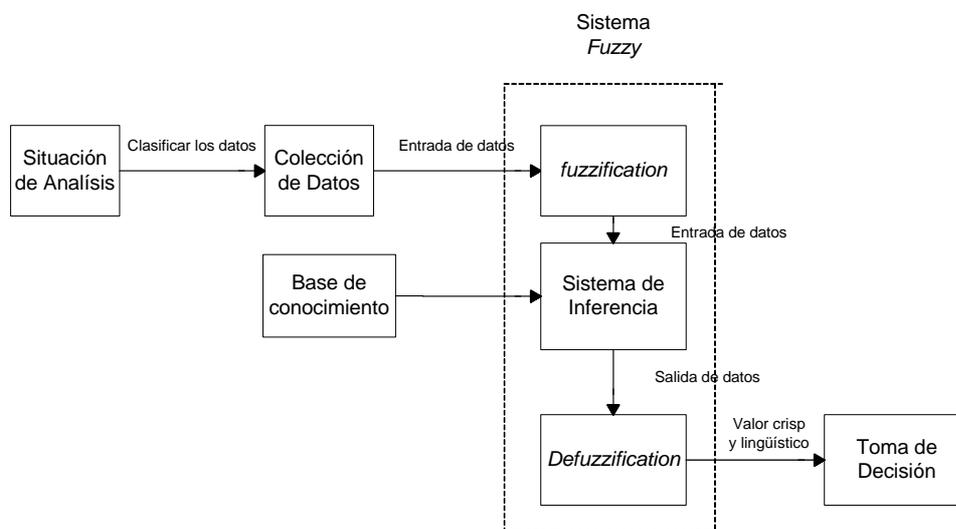


Figura 5.12: Modelo conceptual del STDE_ECAF.

La *fuzzification* obtiene los valores de las variables de entrada al controlador fuzzy; convierte los datos reales (crisp) de entrada en conjuntos fuzzy, lo cual permite ser tratados como tales.

El *Sistema de inferencia* es el núcleo del controlador fuzzy: infiere las acciones de control simulando el proceso de decisión humano usando una implicación fuzzy y las reglas de inferencia de la lógica fuzzy. Utiliza las técnicas de los Sistemas Basados en reglas para la inferencia de los resultados.

La *defuzzification* convierte los valores fuzzy de las variables de salida en valores concretos dentro del universo de discurso correspondiente.

En vista de lo anterior, un DWHA debe ofrecer un modelo de conocimiento y un modelo de ejecución en el manejo de las reglas activas.

Reglas *Fuzzy*

(A) Percepciones = {abarrotes_básicos(A_1), abarrotes_procesados(A_2), lácteos(A_3), carnes y pescados(A_4)}

(B) No percepciones = {desechables(B_1), químicos(B_2), vinos_licores(B_3), juguetería(B_4), moda(B_5), plásticos(B_6), vidrio(B_7)}

$A_1 = \{jitomate, cebolla, papa, tomate, etc\}$

$A_2 = \{atùn, chiles_vinagre, mayonesa, sardina, etc.\}$

$A_3 = \{leche, queso_manchego, queso_fresco, mantequilla, etc.\}$

$A_4 = \{pollo_entero, pechuga_pollo, bistec, carne_molida, etc.\}$

$B_1 = \{papel_higienico, servilletas, platos, vasos, tenedor, cuchara, etc.\}$

$B_2 = \{cloro, jabón_trastes, jabón_corporal, desodorantes, etc.\}$

$B_3 = \{tequilas, whisky, coñage, vino_tinto, vino_blanco, etc.\}$

$B_4 = \{muñecas, carros, peluches, pelotas, bicicletas, juegos_mesa, etc.\}$

$B_5 = \{ropa_dama, ropa_caballero, niños, niñas, bebes, calzado, etc.\}$

$B_6 = \{vasos, botes_basura, botes_agua, platos, cucharas, accesorios_cocina, etc.\}$

$B_7 = \{platos, vasos, moldes, jarras, floreros, etc.\}$

SI A_1 es perecedero, ENTONCES frecuencia es ALTA

SI A_2 es perecedero, ENTONCES frecuencia es ALTA

SI A_3 es perecedero, ENTONCES frecuencia es MEDIA

SI A_4 es perecedero, ENTONCES frecuencia es ALTA

SI existencias es muy urgente Y tiempo muy urgente ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo muy urgente ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo urgente ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo urgente ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo moderado ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy leve ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo nulo ENTONCES proveedor_mayorista.

SI existencias es urgente Y tiempo muy nulo ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo nulo ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy nulo ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo nulo ENTONCES proveedor_mayorista.

SI B₁ es no_perecedero, ENTONCES frecuencia es MEDIA

SI B₂ es no_perecedero, ENTONCES frecuencia es MEDIA

SI existencias es muy urgente Y tiempo muy urgente ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo muy urgente ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo urgente ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo urgente ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo moderado ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy leve ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo nulo ENTONCES proveedor_mayorista.

SI existencias es urgente Y tiempo muy nulo ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo nulo ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy nulo ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo nulo ENTONCES proveedor_mayorista.

SI B₃ es no_perecedero, ENTONCES frecuencia es BAJA

SI B₄ es no_perecedero, ENTONCES frecuencia es BAJA

SI B₅ es no_perecedero, ENTONCES frecuencia es BAJA

SI B₆ es no_perecedero, ENTONCES frecuencia es BAJA

SI B₆ es no_perecedero, ENTONCES frecuencia es BAJA

SI existencias es muy urgente Y tiempo muy urgente ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo muy urgente ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo urgente ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo urgente ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo moderado ENTONCES proveedor_minorista.

SI existencias es urgente Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo moderado ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy leve ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo leve ENTONCES proveedor_mayorista.

SI existencias es muy urgente Y tiempo nulo ENTONCES proveedor_mayorista.

SI existencias es urgente Y tiempo muy nulo ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo nulo ENTONCES proveedor_mayorista.

SI existencias es leve Y tiempo muy nulo ENTONCES proveedor_mayorista.

SI existencias es menos urgente Y tiempo nulo ENTONCES proveedor_mayorista.

5.4. Comentarios finales

Las reglas de análisis por su contexto son las que dan el comportamiento activo en una BD y por consiguiente también apoyan a un DWH en su reactividad para poder definir reglas de análisis en conjunto con la lógica fuzzy y así apoyar a la Toma de Decisiones, tanto operacionales como estratégicas. Los cursores del lenguaje PL/pgSQL son una buena opción para el encapsulamiento de varias consultas en donde sea necesario hacer referencia a diferentes tablas dentro de una misma BD, en una forma en la que éstas puedan ser asignados en un espacio de memoria y así no sobrecargarla. Muchas corporaciones que tienen implementadas un warehouse hacen uso de los lenguajes procedurales, debido a la programación estructural y a las funciones con sentencias SQL.

Capítulo 6

Caso de Estudio.

Se desarrolló un caso de estudio para definir las reglas en una tienda de autoservicio en, donde por la gran cantidad de información de sus artículos que ofrecen a sus clientes, es posible implementar en un DWH dichas reglas. Todo esto de acuerdo a los procesos, restricciones, movimientos y decisiones que automáticamente pueden ser ejecutados, de acuerdo a ciertos parámetros requeridos para la toma de decisión estratégica que existan en nuestro DWH.

Se utilizó el modelo conceptual Entidad-Relación(E-R), en donde el diseño lógico esta basado en la teoría del modelado multidimensional para este caso de estudio.

6.1. Tablas y Relaciones

- `dimension_almacen` (tienda, localidad, ciudad, región, dirección). En esta tabla se obtienen los datos necesarios para ubicar a cada tienda de autoservicio, según su ubicación geográfica.
- `dimension_articulo` (artículo, categoría, departamento, proveedor, caducidad del artículo, temporada del artículo, precio, unidades en almacén, fecha de surtimiento). Para describir a cada artículo de la tienda, en esta tabla se introduce los datos necesarios que lo analizan desde su proveedor hasta su fecha de caducidad.
- `dimension_empleado` (empleado, nombre, apellido, edad y dirección). Tabla que provee los datos de cada empleado que figura en la tienda de autoservicio.

- `dimension_promoción` (promoción, nombre, tipo de venta, tipo de pago, fecha de inicio de la promoción, fecha fin de la promoción, banco, convenio, ¿tiene promoción?, descuento, hora de inicio promoción, hora fin de la promoción). Aquí se introducen los datos necesarios para figurar una promoción dentro de la tienda, desde si algún banco esta incluido hasta cuanta duración tendra la promoción.
- `dimension_proveedor` (proveedor, nombre, del proveedor, giro del proveedor, región a la que pertenece, tipo de proveedor, dirección, fechas de entrega, costos de entrega, tipo de transporte, volumen de transporte). Esta tabla es de las más importante, ya que nos proporciona datos de interés acerca del proveedor, como por ejemplo su ubicación geográfica, así sabremos que tan cerca o lejos se encuentra dicho proveedor de la tienda comercial, también que tipo de proveedor si es mayorista o minorita, sus rangos de costo por entrega, el tiempo que tarda cada entrega, el tipo de vehiculo con el o con los que cuenta para sus entregas, entre otros datos importantes para definir los *valores lingüísticos*.
- `dimension_queja` (queja, fecha de la queja, motivo de la queja, estatus de la queja). Aquí se almacenan las quejas realizadas en la tienda comercial.
- `dimension_tiempo` (dia, mes, año, temporada, dia de la semana). Esta tabla también es muy importante ya que el tiempo se encuentra relacionada con cada tabla dentro del DWH.

6.2. Tablas de Hechos

- `hechos_pedidos` (producto, proveedor, tienda, fecha de pedido, unidades vendidas, costo del artículo, monto de la venta, estatus del pedido, hora en que se hizo el pedido). Concentra en una tabla todos los pedidos que fueron realizados y que fueron almacenados en sus respectivas tablas, proporcionado concretamente solo información relacionada con los pedidos realizados.
- `hechos_quejas` (número de queja, empleado, producto, proveedor, fecha de la queja, motivo de la queja, estatus de la queja). Reune los datos necesarios para crear y conformar en una sola tabla los datos necesarios para llevar un mejor análisis de las quejas realizadas.
- `hechos_ventas` (fecha de las ventas, período de la venta, artículo, tienda, promoción, unidades vendidas, monto vendido de la ventas, monto vendido por artículo). En esta tabla se concentra

las ventas realizadas en la tienda de autoservicio. Reune los elementos necesarios de cada tabla respectivamente, para poder llevar la automatización del inventario. En esta tabla de hechos se obtienen los datos necesarios para hacer un buen análisis de los datos *ventas*, apartir de aqui se crean tablas resumen en el cual se realizan operaciones y cálculos para poder crear la automatización operacional dentro del DWH.

6.3. Tablas Resumen y Conjuntos Fuzzy

Las siguientes tablas surgen a raíz de las tablas de hechos, para poder realizar la definición de las reglas ECA Fuzzy y así obtener automatización en nuestro DWH.

- *resumen_ventas* (temporada, unidades vendidas, monto total vendido, monto vendido por artículo). Surge de las relaciones entre varias tablas para actualizar lo artículos de el inventario de la tienda. De aquí se pueden hacer las estadísticas de ventas de los artículos para obtener sus valores de pertenencia según la frecuencia de venta que presenta cada artículo.
- *resumen_articulo* (artículo, categoría a la que pertenece, departamento, proveedor, frecuencia de venta, fecha de resurtimiento). Del *resumen_ventas* se pueden asignar los valores de pertenencia del artículo y los conjuntos fuzzy según la categoría de los artículos.
- *resumen_proveedor* (proveedor, frecuencia de entrega, ubicación, costo). El resumen es resultado de obtener los datos necesarios de la tabla *dimension_proveedor* de aquí surgen los valores de pertenencia para poder asignar los valores lingüísticos.
- *resumen_existencias* (artículo, existencias, surtir, precio del artículo). Aquí se calculan y se actualiza del inventario las existencias reales de los artículos, se dan los valores de pertencia de las existencias para asignarle su valor lingüístico.

De la tabla *dimension_artículo* se obtiene los conjuntos fuzzy para asignarle sus valores de pertenencia y sus valores lingüísticos, y así poder definir las reglas ECA Fuzzy de la siguiente manera:

Perecederos = {abarrotes_básicos, abarrotes_procesados, lácteos, carnes y pescados} (Fig. 6.1)

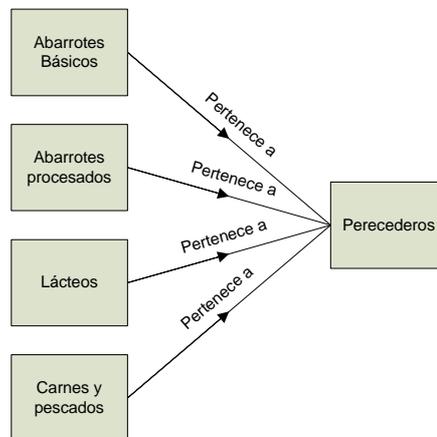


Figura 6.1: Clasificación de los artículos en conjuntos perecederos.

No perecederos = {desechables, químicos, vinos_licores, juguetería, moda, plásticos, vidrio }
 (Fig. 6.2)

(A) Perecederos.

(A₁) abarrotes_básicos,

(A₂) abarrotes_procesados,

(A₃) lácteos,

(A₄) carnes y pescados.

(B) No perecederos.

(B₁) desechables,

(B₂) químicos,

(B₃) vinos_licores,

(B₄) juguetería,

(B₅) moda,

(B₆) plásticos,

(B₇) vidrio.

$A_1 = \{jitomate, cebolla, papa, tomate, etc\}$

$A_2 = \{atùn, chiles_vinagre, mayonesa, sardina, etc.\}$

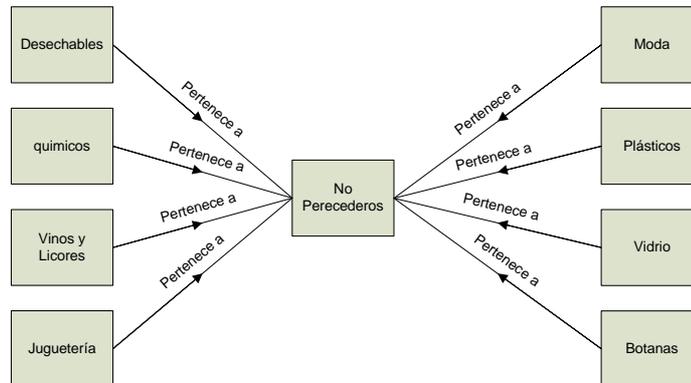


Figura 6.2: Clasificación de los artículos en conjuntos No perecederos.

$A_3 = \{leche, queso_manchego, queso_fresco, mantequilla, etc.\}$

$A_4 = \{pollo_entero, pechuga_pollo, bistec, carne_molid, etc.\}$

$B_1 = \{papel_higienico, servilletas, platos, vasos, tenedor, cuchara, etc.\}$

$B_2 = \{cloro, jabón_trastes, jabón_corporal, desodorantes, etc.\}$

$B_3 = \{tequilas, whisky, coñage, vino_tinto, .vino_blanco, etc.\}$

$B_4 = \{muñecas, carros, peluches, pelotas, bicicletas, juegos_mesa, etc.\}$

$B_5 = \{ropa_dama, ropa_caballero, niños, niñas, bebes, calzado, etc.\}$

$B_6 = \{vasos, botes_basura, botes_agua, platos, cucharas, accesorios_cocina, etc.\}$

$B_7 = \{platos, vasos, moldes, jarras, floreros, etc.\}$

6.4. Reglas ECA Fuzzy

Cuando los artículos fueron clasificados dentro de sus respectivos conjuntos y se le asignaron valores de pertenencia, se definieron las reglas ECA Fuzzy. Estas reglas son ejecutadas automáticamente cada vez que un evento es disparado dentro del DWH y así activa a esta. Las reglas que fueron definidas dentro del DWH son las siguientes:

Reglas que corresponden a la asignación de la frecuencia de venta de los artículos por día (unidades vendidas por día), según al conjunto que pertenecen, asignándole su correspondiente valor lingüístico.

SI A₁ es perecedero, ENTONCES frecuencia es ALTA
 SI A₂ es perecedero, ENTONCES frecuencia es ALTA
 SI A₃ es perecedero, ENTONCES frecuencia es MEDIA
 SI A₄ es perecedero, ENTONCES frecuencia es ALTA.

Valores lingüísticos determinados, según el consumo de los clientes por día, donde podemos ver que los perecederos tienen más consumo por día que los No perecederos.

SI B₁ es no_perecedero, ENTONCES frecuencia es MEDIA
 SI B₂ es no_perecedero, ENTONCES frecuencia es MEDIA
 SI B₃ es no_perecedero, ENTONCES frecuencia es MEDIA
 SI B₄ es no_perecedero, ENTONCES frecuencia es BAJA
 SI B₅ es no_perecedero, ENTONCES frecuencia es BAJA
 SI B₆ es no_perecedero, ENTONCES frecuencia es BAJA
 SI B₇ es no_perecedero, ENTONCES frecuencia es BAJA

En la figura 6.3 se dá gráficamente los rangos de cada grado de membresía, en donde parametrizamos que las ventas altas, estan dadas si cumplen entre un rango de 150 a 200 (correspondientes a un promedio de ventas semanales) por arriba del grado de membresía de 0.5, ya que hay por debajo del 0.5 puede considerarse una frecuencia de venta media; puede visualizarse superposición entre valores, definiendo limites en forma triangular.

Reglas que corresponden a la asignación del valor lingüístico de los costos de los proveedores tomando en cuenta varios factores indispensables para clasificar los costos por surtido. En donde la dificultad de traslado es importante como factor determinante para el cobro de los servicios, así como la ubicación del proveedor para valorar la distancia en kilometros e incluirlo en el factor de costos.

SI dif_traslado es Muy Complejo Y ubicacion_proveedor ES Muy Lejos
 ENTONCES costos_proveedor ES Muy Excesivo
 SI dif_traslado es Muy Complejo Y ubicacion_proveedor ES Lejos
 ENTONCES costos_proveedor ES Muy Excesivo

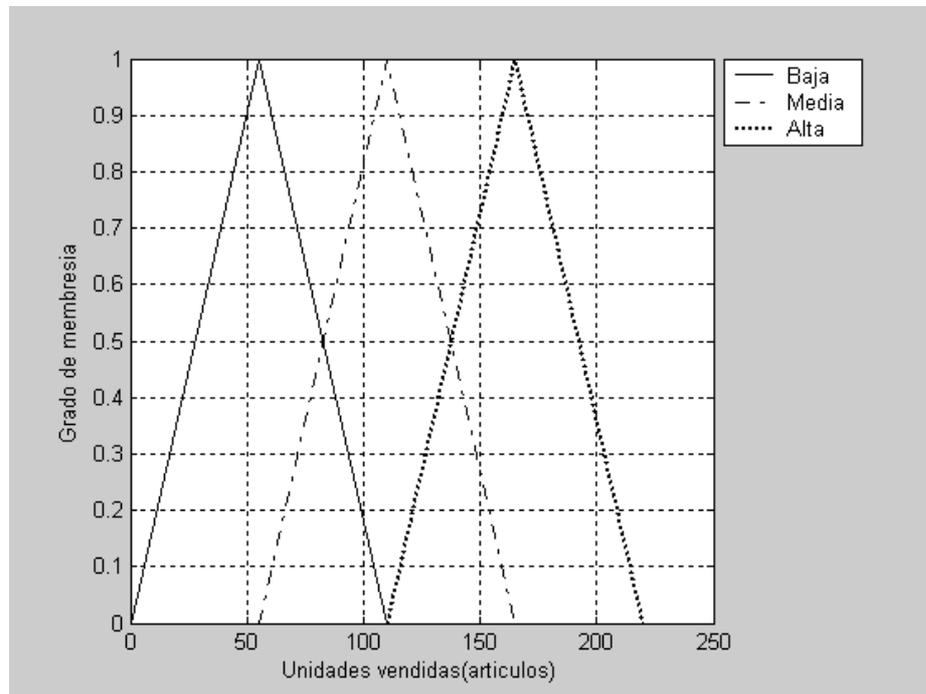


Figura 6.3: Grados de membresía parámetro “ventas por día”.

SI dif_traslado es Muy Complejo Y ubicacion_proveedor ES Medio

ENTONCES costos_proveedor ES Excesivo

SI dif_traslado es Muy Complejo Y ubicacion_proveedor ES Cerca

ENTONCES costos_proveedor ES Excesivo

SI dif_traslado es Muy Complejo Y ubicacion_proveedor ES Muy Cerca

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es Complejo Y ubicacion_proveedor ES Muy Lejos

ENTONCES costos_proveedor ES Muy Excesivo

SI dif_traslado es Complejo Y ubicacion_proveedor ES Lejos

ENTONCES costos_proveedor ES Muy Excesivo

SI dif_traslado es Complejo Y ubicacion_proveedor ES Medio

ENTONCES costos_proveedor ES Excesivo

SI dif_traslado es Complejo Y ubicacion_proveedor ES Cerca

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es Complejo Y ubicacion_proveedor ES Muy Cerca

ENTONCES costos_proveedor ES Barato

SI dif_traslado es Moderado Y ubicacion_proveedor ES Muy Lejos

ENTONCES costos_proveedor ES Excesivo

SI dif_traslado es Moderado Y ubicacion_proveedor ES Lejos

ENTONCES costos_proveedor ES Excesivo

SI dif_traslado es Moderado Y ubicacion_proveedor ES Medio

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es Moderado Y ubicacion_proveedor ES Cerca

ENTONCES costos_proveedor ES Barato

SI dif_traslado es Moderado Y ubicacion_proveedor ES Muy Cerca

ENTONCES costos_proveedor ES Muy Barato

SI dif_traslado es No Muy Complejo Y ubicacion_proveedor ES Muy Lejos

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es No Muy Complejo Y ubicacion_proveedor ES Lejos

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es No Muy Complejo Y ubicacion_proveedor ES Medio

ENTONCES costos_proveedor ES Barato

SI dif_traslado es No Muy Complejo Y ubicacion_proveedor ES Cerca

ENTONCES costos_proveedor ES Barato

SI dif_traslado es No Muy Complejo Y ubicacion_proveedor ES Muy Cerca ENTONCES costos_proveedor ES Muy

Barato

SI dif_traslado es NoComplejo Y ubicacion_proveedor ES Muy Lejos

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es No Complejo Y ubicacion_proveedor ES Lejos

ENTONCES costos_proveedor ES Considerable

SI dif_traslado es No Complejo Y ubicacion_proveedor ES Medio

```

ENTONCES costos_proveedor ES Barato
SI dif_traslado es No Complejo Y ubicacion_proveedor ES Cerca
ENTONCES costos_proveedor ES Barato
SI dif_traslado es No Complejo Y ubicacion_proveedor ES Muy Cerca
ENTONCES costos_proveedor ES Muy Barato

```

La figura 6.4 nos grafica los rangos que comprende a cada etiqueta lingüística, basada en el peso de la mercancía a transportar en toneladas, ya que el peso de la mercancía esta ligada con el vehiculo que se requiere para este fin, ademas del volumen de la mercancía. Se define en la grafica los valores de 0 a 10 correspondiente a las toneladas de la mercancía en donde si el peso de la mercancía a transporta esta entre los valores de 8 a 10 por arriba del grado de membresía de 0.5 su etiqueta lingüística es “muy complejo”.

Para definir el parámetro distancia, se hace uso de las etiquetas lingüísticas “Muy lejos” para definir, sí la distancia entre el proveedor y la tienda a surtir se encuentra en un diametro de distancia de 1500 y 2000 kilometros en la figura 6.5 se detallan todas las etiquetas lingüísticas para cada distancia en Kilometros.

La gráfica 6.6 detallan los grados de membresía de los incrementos del costo del artículo, los cuales son en porcentaje(%), siendo resultado de los parámetros “dificultad de traslado” y “ubicacion geográfica del proveedor”. El minimo incremento va desde 0% correspondiente si no hay ningun incremento en el costo del articulo, y un 100% de incremento si los parámetros fuesen los limites mayores.

Se dice que un almacen esta completo cuando las existencias de los articulos estan al 100% de su capacidad, en la implementación de las reglas ECA Fuzzy se le asignó rangos de membresía entre 80 y 100 para definir que las existencias de artículos en almacen esta Totalmente completo como lo muestra la figura 6.7.

Cuando la fecha de resurtido esta próxima, nos da un estimado del tiempo que falta para surtir almacen por el proveedor. En la figura 6.8 se gráfica las etiquetas lingüísticas correspondiente a él porcentaje del tiempo (días) para resurtimiento en almacen dentro de la tienda; en donde el valor 100 corresponde cuando apenas se acaba de surtir el articulo en la tienda, por lo tanto el tiempo

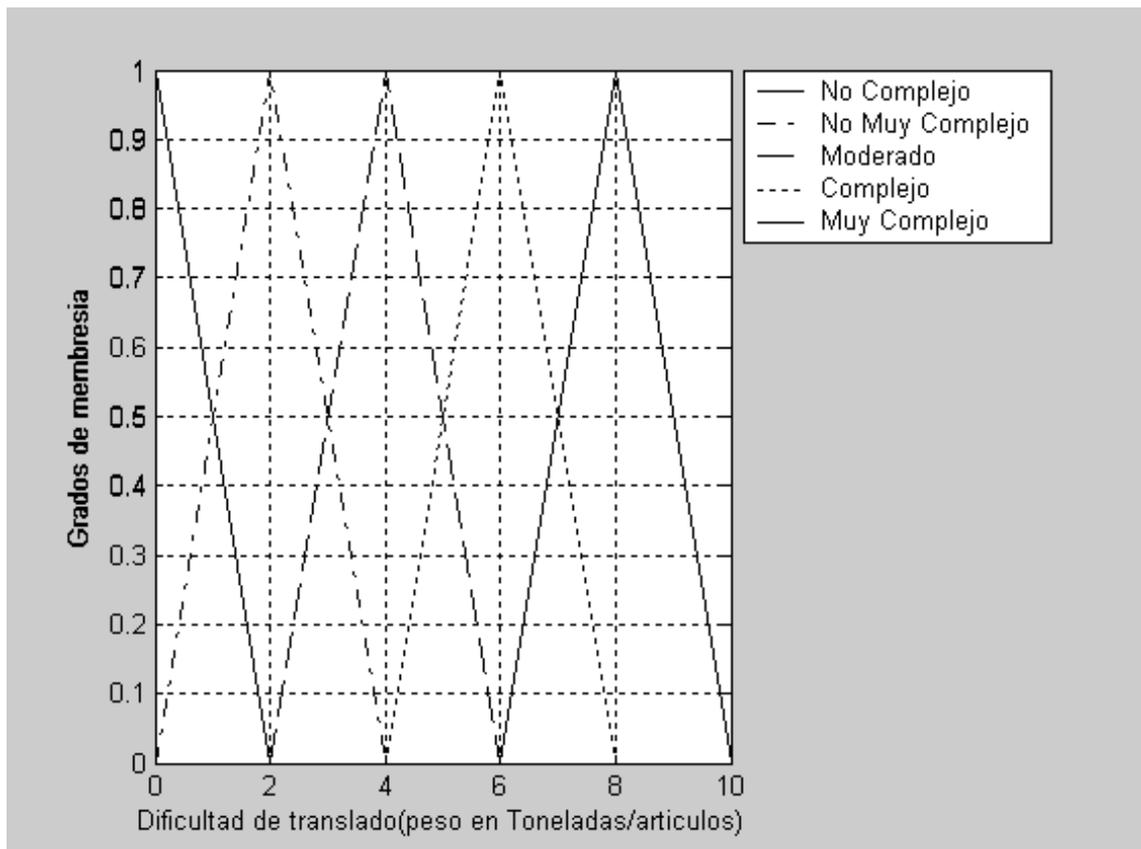


Figura 6.4: Graficación del grado de membresía parámetro “peso de los artículos” a transportar.

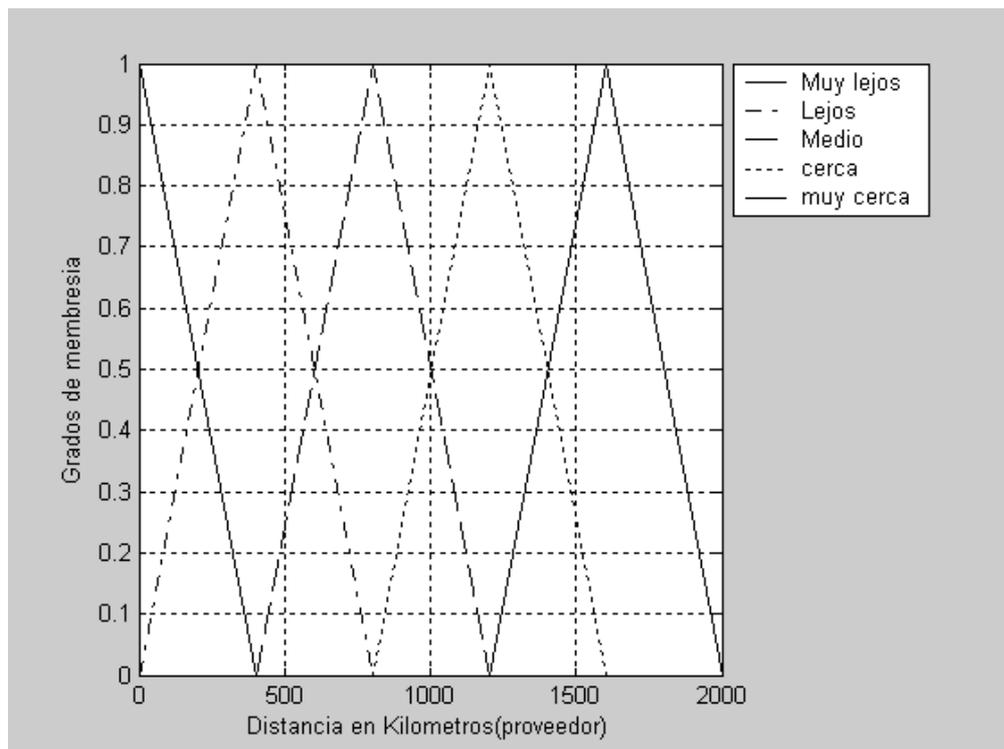


Figura 6.5: Grados de membresía parámetro “distancia del proveedor”.

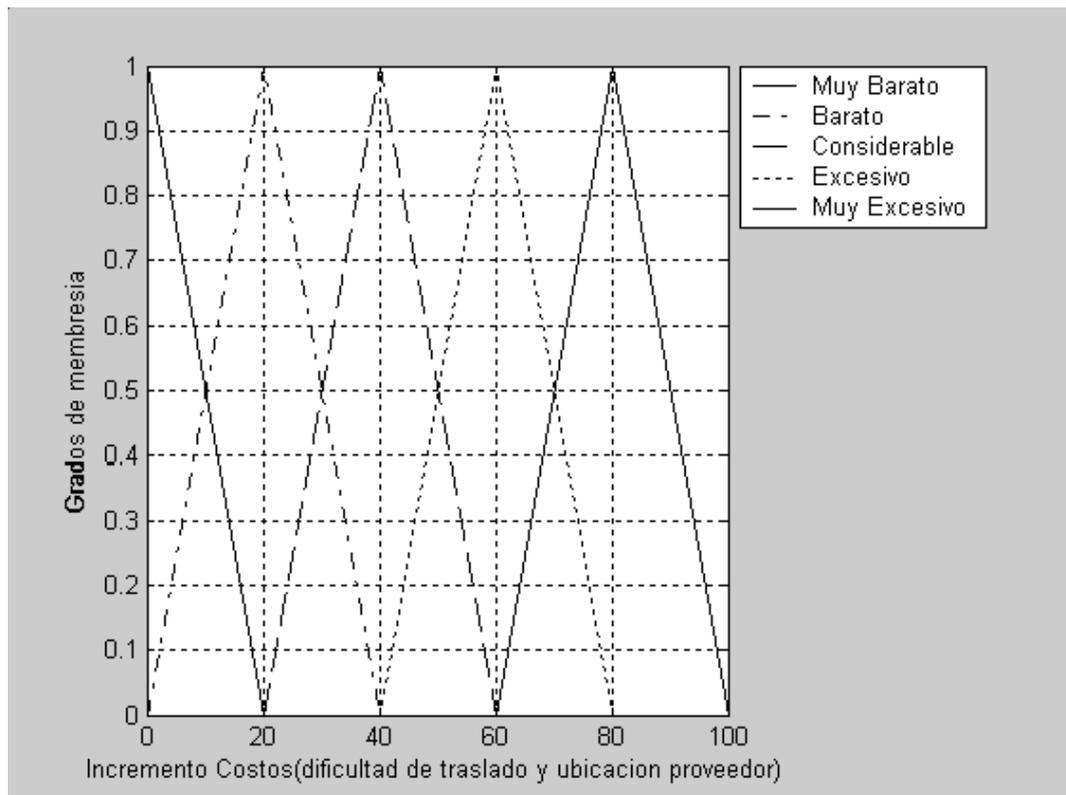


Figura 6.6: Grados de membresía parámetros “costo”.

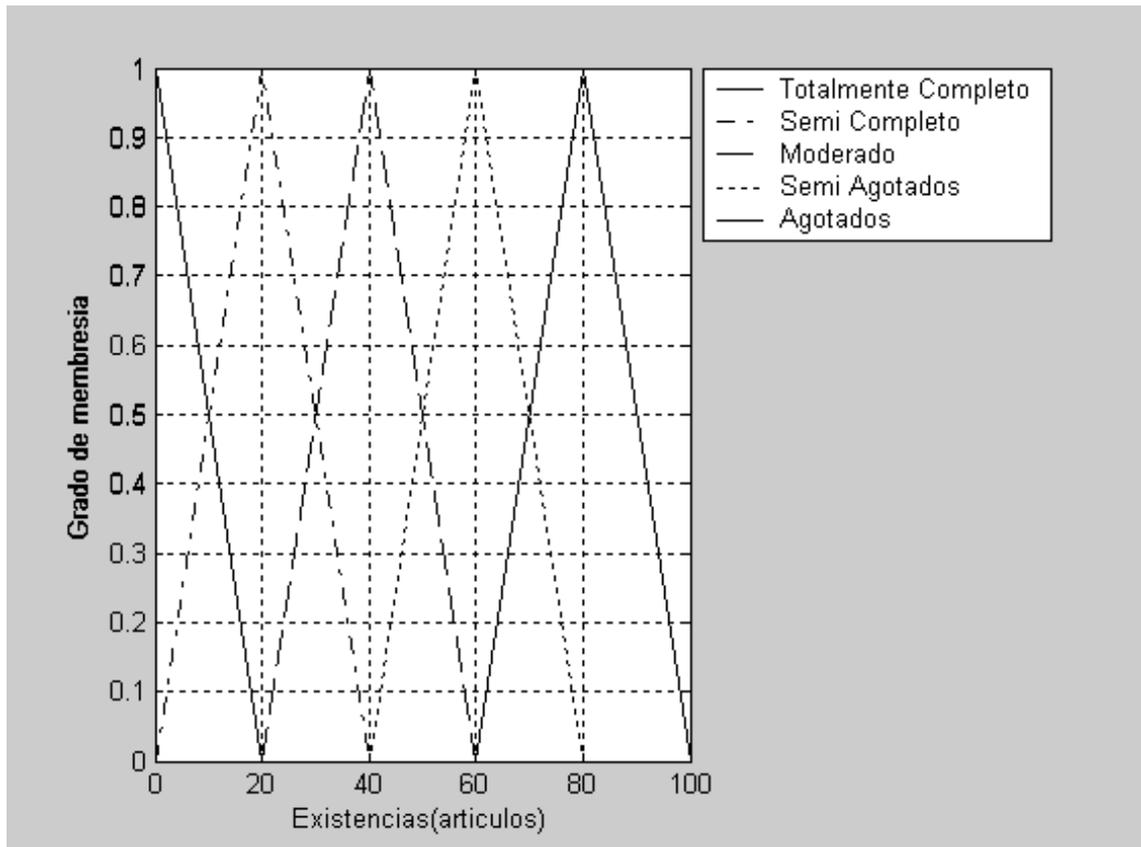


Figura 6.7: Grados de membresía de los artículos parámetro “porcentaje de existencias en almacén”.

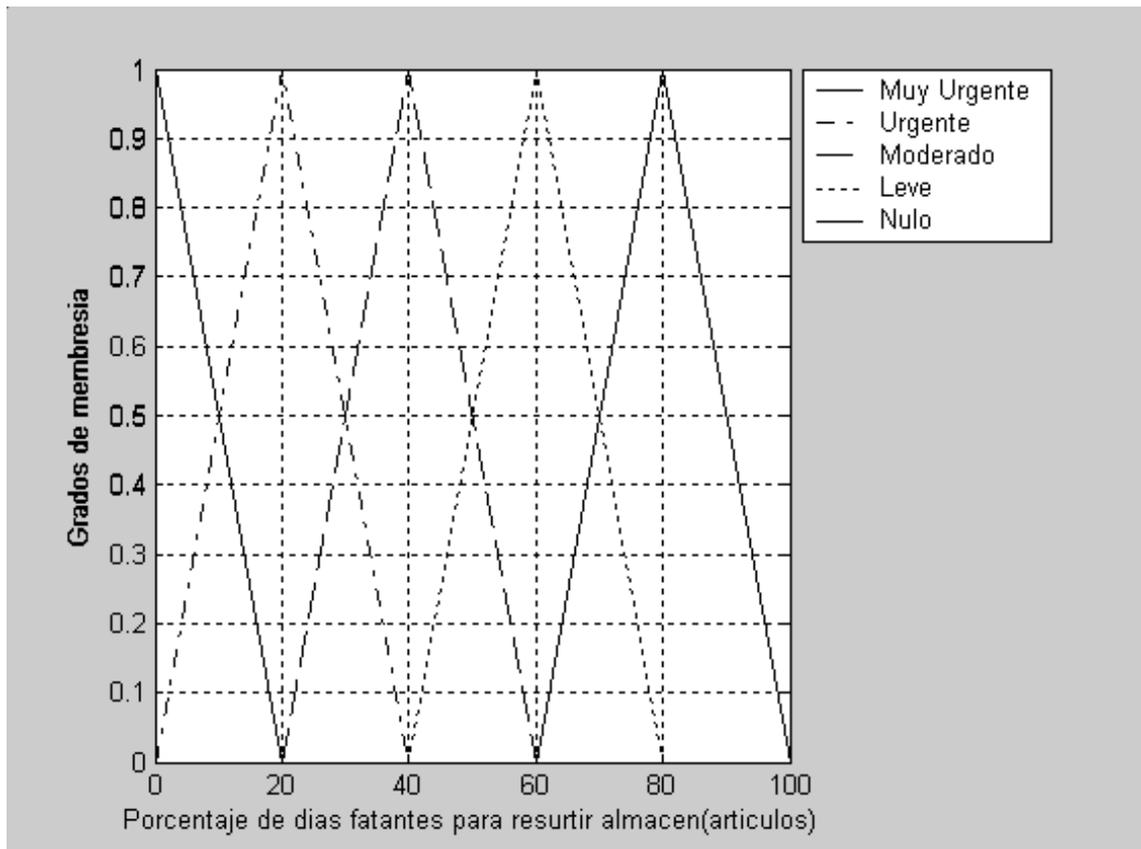


Figura 6.8: Clasificación de los artículos parámetro “porcentaje de días faltantes” para resurtir en la tienda .

que falta por resurtir es de un 100 % y en caso de que haya una disrupción en la tienda la etiqueta que le corresponde es de “Muy urgente”.

SI existencias es Agotado Y tiempo muy urgente Y costos Muy Barato
ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo muy urgente Y costos Muy Barato
ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo muy urgente Y costos Muy Barato
ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo muy urgente Y costos Muy Barato
ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo muy urgente Y costos Muy Barato
ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo urgente
ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo urgente
ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo urgente
ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo urgente
ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo urgente
ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo moderado
ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo moderado
ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo moderado
ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo moderado
ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo leve

ENTONCES proveedor_mayorista.

SI existencias es Semi Agotados Y tiempo leve

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo leve

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo muy leve

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo leve

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo nulo

ENTONCES proveedor_mayorista.

SI existencias es Semi Agotados Y tiempo muy nulo

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo nulo

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo muy nulo

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo nulo

ENTONCES proveedor_mayorista.

- Estas reglas corresponden cuando ya no hay artículos en exhibición o están por agotarse y aun faltan días para abastecer anaquel.

SI existencias es Agotado Y tiempo muy urgente

ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo muy urgente

ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo urgente

ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo urgente

ENTONCES proveedor_minorista.

SI existencias es moderado Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo moderado

ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo muy urgente

ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo muy urgente

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo urgente

ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo urgente

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo moderado

ENTONCES proveedor_minorista.

SI existencias es Semi Agotados Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo moderado

ENTONCES proveedor_mayorista.

SI existencias es Agotado Y tiempo leve

ENTONCES proveedor_mayorista.

SI existencias es Semi Agotados Y tiempo leve

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo leve

ENTONCES proveedor_mayorista.
SI existencias es Semi Completo Y tiempo muy leve
ENTONCES proveedor_mayorista.
SI existencias es Totalmente Completo Y tiempo leve
ENTONCES proveedor_mayorista.
SI existencias es Agotado Y tiempo nulo
ENTONCES proveedor_mayorista.
SI existencias es Semi Agotados Y tiempo muy nulo
ENTONCES proveedor_mayorista.
SI existencias es moderado Y tiempo nulo
ENTONCES proveedor_mayorista.
SI existencias es Semi Completo Y tiempo muy nulo
ENTONCES proveedor_mayorista.
SI existencias es Totalmente Completo Y tiempo nulo
ENTONCES proveedor_mayorista.

Estas reglas corresponden cuando ya no hay artículos en exhibición o están por agotarse y ya están por abastecer anaquel por los proveedores establecidos de la tienda.

SI existencias es Agotado Y tiempo leve
ENTONCES proveedor_mayorista.
SI existencias es Semi Agotados Y tiempo leve
ENTONCES proveedor_mayorista.
SI existencias es moderado Y tiempo leve
ENTONCES proveedor_mayorista.
SI existencias es Semi Completo Y tiempo muy leve
ENTONCES proveedor_mayorista.
SI existencias es Totalmente Completo Y tiempo leve
ENTONCES proveedor_mayorista.
SI existencias es Agotado Y tiempo nulo
ENTONCES proveedor_mayorista.

SI existencias es Semi Agotados Y tiempo muy nulo

ENTONCES proveedor_mayorista.

SI existencias es moderado Y tiempo nulo

ENTONCES proveedor_mayorista.

SI existencias es Semi Completo Y tiempo muy nulo

ENTONCES proveedor_mayorista.

SI existencias es Totalmente Completo Y tiempo nulo

ENTONCES proveedor_mayorista.

Capítulo 7

Aplicabilidad del Modelo

En este capítulo se justifican los datos usados para nuestro modelo de datos, que fueron usados dentro de nuestro DWH como parámetros para generar nuestras reglas ECA Fuzzy.

7.1. Entorno de simulación

Un simulador se crea para representar el funcionamiento de un sistema o de un fenómeno antes de su construcción o utilización. En nuestro caso se implementó para representar el comportamiento de una tienda de autoservicio, fingiendo compras dentro de cada departamento de la tienda.

Objetivo

El objetivo principal del simulador es el del validar parámetros entre nuestras reglas ECA Fuzzy declaradas para la automatización de la toma de decisión y los parámetros estadísticos generados en forma aleatoria.

Esta validación nos dará un estimado certero o no certero de los datos, que se implementaron dentro de los parámetros, para definir las etiquetas lingüísticas que conforman las reglas modeladas ECA Fuzzy.

La figura 7.1 se detalla el comportamiento del simulador y como se obtiene los datos para acatar el objetivo de este. En donde se crean threads con diferentes personalidades de clientes, los cuales pueden ser “ama de casa”, “profesionista”, “padre de familia”, “estudiante”, “obrero”;

estos son generados de forma aleatoria en cada threads, una vez que son generados cada perfil del cliente, estos entran a la tienda, en donde pasan por otro generador aleatorio para determinar a que departamentos tendrá que visitar y comprar de forma aleatoria artículos del depto., una vez que han sido adquiridos los articulos por los clientes en tiempos aleatorios, se hace conexión con una tabla que nuestro DWH, para después validar estos datos con los definidos en las reglas ECAF.

Diseño

En primera instancia se crearon threads (hilos) para cada tipo de cliente, ya que cada perfil tiende a frecuentar más ciertos departamentos de la tienda.

Los departamentos que se justifican en el modelo en una tienda son solamente:

- Abarrotes básicos,
- Abarrotes procesados,
- lácteos,
- carnes y pescados,
- vinos y licores,
- desechables,
- químicos,
- juguetería,
- moda,
- plasticos,
- vidrio.

Una vez que los clientes son generados, la segunda instancia es generar aleatoriamente la frecuencia de compra para cada departamento que otro generador de forma aleatoria les asignará.

La tercera instancia es obtener los datos de la frecuencia generada por cada cliente en cada departamento y depositados en alguna tabla del DWH para su manipuleo.

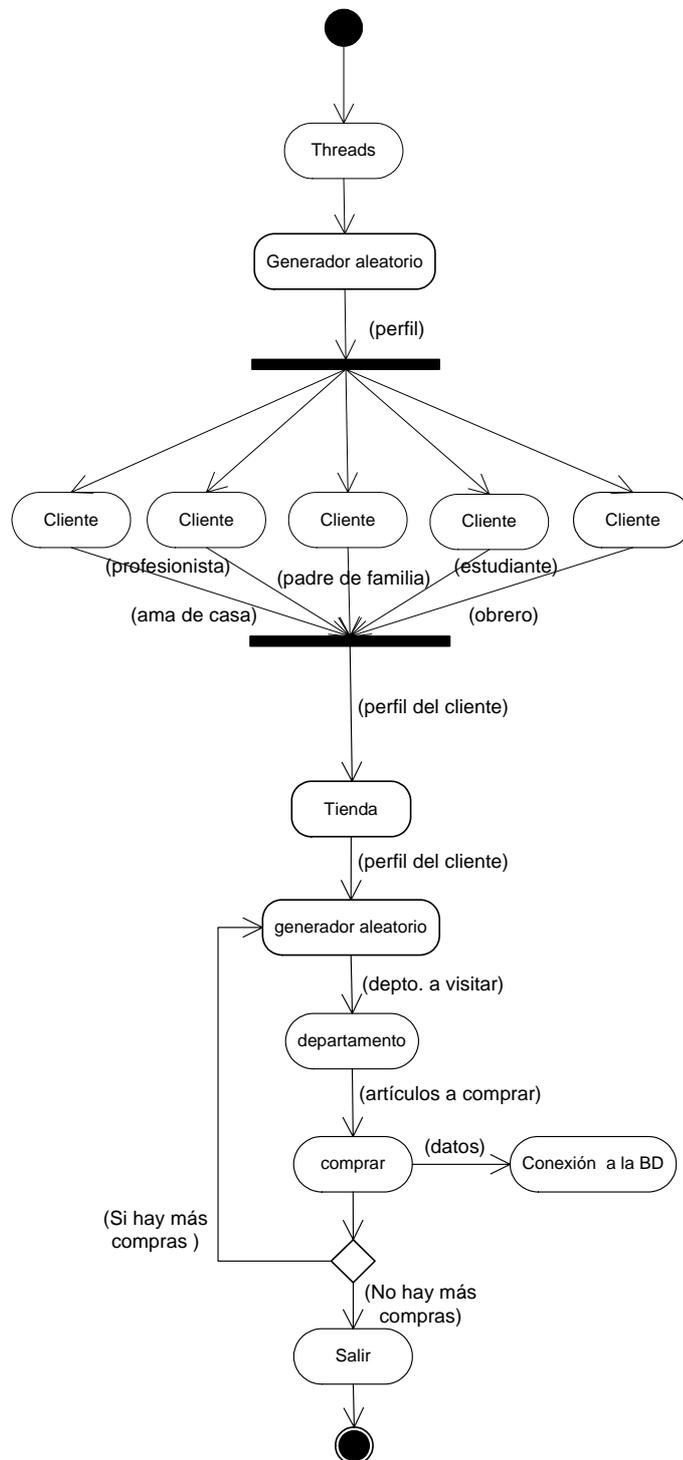


Figura 7.1: Diagrama de actividad del Simulador.

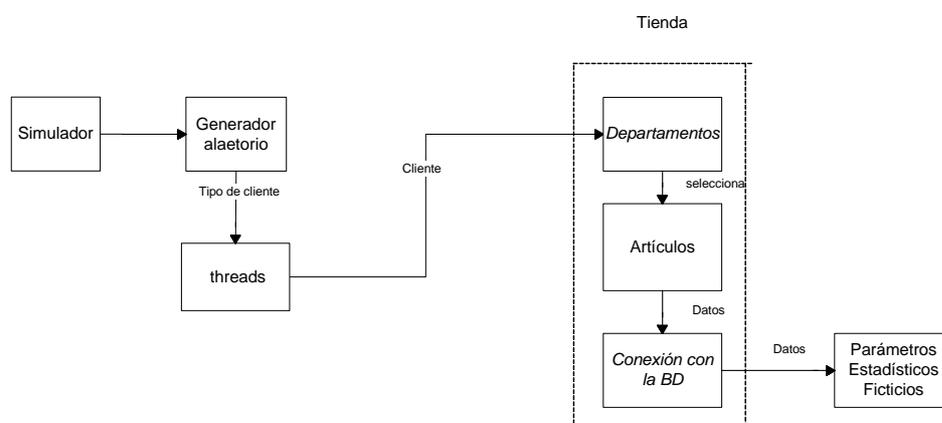


Figura 7.2: Modelo conceptual del simulador.

Modelo conceptual

En la figura 7.2 se muestra el modelo que sigue el simulador, pasando por cada una de las instancias antes mencionadas, hasta llegar a la conexión con la BD y así poder definir mediante estadísticas los parámetros que serán como modelo en nuestras reglas ECA como validador.

Ambiente de desarrollo

Se desarrolló una aplicación en la plataforma Java, para generar aleatoriamente nuestras entradas a los departamentos dentro de la tienda de autoservicio.

En la figura 7.3 se muestra como está implementado el simulador en ambiente gráfico, ilustrando cada departamento de la tienda. Se incluye un botón llamado simulación que es el encargado de ejecutar la aplicación y los comportamientos de cada perfil de los clientes.

El botón cancelar sale de la aplicación simulación y termina todo proceso activo.

La figura 7.4 muestra como es accedido cada departamento por los clientes, pintado de color azul el departamento visitado. Al salir del departamento se regresa a su color original el botón.

A la par con el Simulador anterior se ejecutan y despliegan en modo DOS (Fig. 7.5) los procesos que se lanzaron el cual incluye detalles de los artículos que son comprados por cada perfil del cliente,



Figura 7.3: Simulador generador de frecuencias de ventas

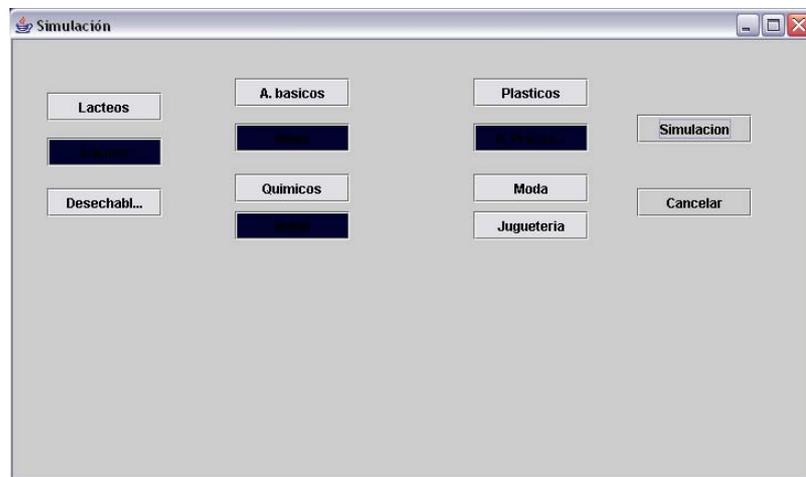
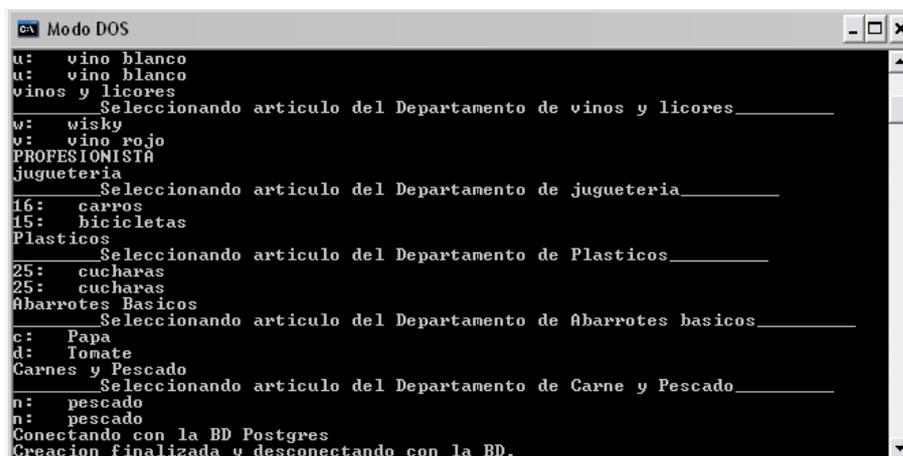


Figura 7.4: Simulador mostrando las compras por departamento.



```

Modo DOS
u: vino blanco
u: vino blanco
vinos y licores
  Seleccionando articulo del Departamento de vinos y licores_____
w: wisky
v: vino rojo
PROFESIONISTA
jugueteria
  Seleccionando articulo del Departamento de jugueteria_____
16: carros
15: bicicletas
Plasticos
  Seleccionando articulo del Departamento de Plasticos_____
25: cucharas
25: cucharas
Abarrotes Basicos
  Seleccionando articulo del Departamento de Abarrotes basicos_____
c: Papa
d: Tomate
Carnes y Pescado
  Seleccionando articulo del Departamento de Carne y Pescado_____
n: pescado
n: pescado
Conectando con la BD Postgres
Creacion finalizada y desconectando con la BD.

```

Figura 7.5: Pantalla mostrando los resultados que son generados por el simulador

al finalizar la salida de cada cliente se hace conexión con la BD para ingresar los resultados obtenidos por la compra, una vez hecho la transferencia de datos la BD es cerrada con éxito.

Se lanzaron varias ejecuciones del simulador para obtener los parámetros y validarlos con los parámetros dados a nuestras reglas ECA Fuzzy. Un día en la tienda de autoservicio fue modelada con 450 transacciones en nuestro simulador, representado a 450 clientes.

Para verificar que siguieran el mismo comportamiento se representó una semana de ventas en la tienda, el cuál ocasionó 3150 registros en nuestra tabla de articulos dentro de nuestro DWH.

7.2. Análisis de viabilidad

La viabilidad es una condición que hace posible el funcionamiento del sistema, proyecto o idea al que califica, atendiendo a sus características tecnológicas y a las leyes de la naturaleza involucradas [30].

La viabilidad técnica se evalúa ante un determinado requerimiento o idea para determinar si es posible llevarlo a cabo satisfactoriamente y en condiciones de seguridad con la tecnología disponible, verificando factores diversos como durabilidad, operatividad, implicaciones energéticas, mecanismos de control, según el campo del que se trate.

En el área de comercio, como es nuestro caso, donde los datos de las ventas para una tienda de autoservicio son necesario para la toma de decisión, la viabilidad de los datos debe venir de datos reales y estadísticos de ventas ya realizadas.

Para nuestro caso los datos no pudieron ser tomados de datos reales debido a las políticas de las tiendas de autoservicio de no violación de sustracción de datos. Por tal motivo se opto por generar nuestro datos mediante un simulador, el cual es el encargado de generar ventas de forma aleatoria.

Estos datos nos dan un estimado de ventas diarias, y por lo tanto los datos generados son enviados a una tabla del DWH para su manipuleo.

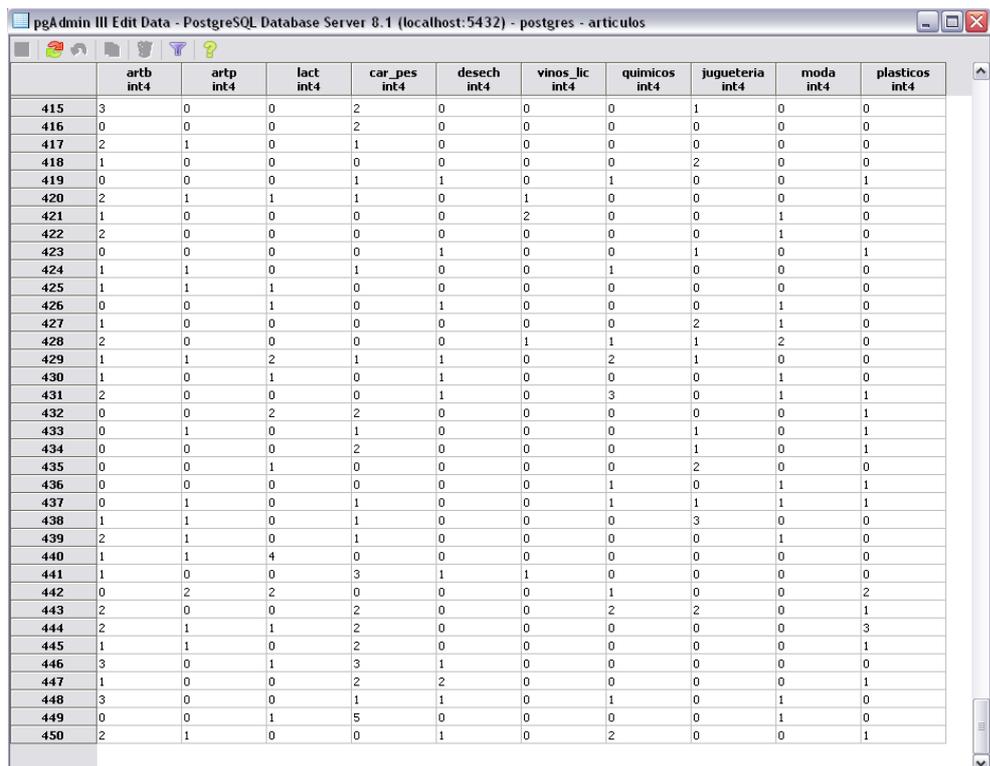
7.3. Análisis de resultados

Los registros que fueron resultado de simular el comportamiento de una tienda de autoservicio en 7 días, fueron anexadas en una tabla llamada articulos dentro de nuestro DWH. La siguiente figura 7.6 despliega los valores obtenidos de un día, son 450 registros y da detalladamente la afluencia que presenta cada departamento por los clientes en la tienda.

Esta tabla contiene los campos que epresentan a cada sección de la tienda.

Los resultados que se obtuvieron que hacer la simulación de 7 días nos muestra que el departamento de abarrotes básicos tiene más preferencia por los clientes, seguido por el depto. de alimentos procesados. El tercer lugar lo ocupa el área de Carnes y pescados. Como podemos ver vinos y licores, juguetería y lácteos aparecen como menos visitados. Los demás departamentos tiene frecuencia de venta media.

Tabla 7.1 Resultado obtenido de un muestreo de 7 días, generados por el simulador.



The image shows a screenshot of the pgAdmin III 'Edit Data' window for a PostgreSQL database. The window title is 'pgAdmin III Edit Data - PostgreSQL Database Server 8.1 (localhost:5432) - postgres - articulos'. The table contains 36 rows of data with 11 columns. The columns are: 'id' (implied), 'artb int4', 'artp int4', 'lact int4', 'car_pes int4', 'desech int4', 'vinos_lic int4', 'quimicos int4', 'jugueteria int4', 'moda int4', and 'plasticos int4'. The data represents sales values for various categories across different items.

	artb int4	artp int4	lact int4	car_pes int4	desech int4	vinos_lic int4	quimicos int4	jugueteria int4	moda int4	plasticos int4
415	3	0	0	2	0	0	0	1	0	0
416	0	0	0	2	0	0	0	0	0	0
417	2	1	0	1	0	0	0	0	0	0
418	1	0	0	0	0	0	0	2	0	0
419	0	0	0	1	1	0	1	0	0	1
420	2	1	1	1	0	1	0	0	0	0
421	1	0	0	0	0	2	0	0	1	0
422	2	0	0	0	0	0	0	0	1	0
423	0	0	0	0	1	0	0	1	0	1
424	1	1	0	1	0	0	1	0	0	0
425	1	1	1	0	0	0	0	0	0	0
426	0	0	1	0	1	0	0	0	1	0
427	1	0	0	0	0	0	0	2	1	0
428	2	0	0	0	0	1	1	1	2	0
429	1	1	2	1	1	0	2	1	0	0
430	1	0	1	0	1	0	0	0	1	0
431	2	0	0	0	1	0	3	0	1	1
432	0	0	2	2	0	0	0	0	0	1
433	0	1	0	1	0	0	0	1	0	1
434	0	0	0	2	0	0	0	1	0	1
435	0	0	1	0	0	0	0	2	0	0
436	0	0	0	0	0	0	1	0	1	1
437	0	1	0	1	0	0	1	1	1	1
438	1	1	0	1	0	0	0	3	0	0
439	2	1	0	1	0	0	0	0	1	0
440	1	1	4	0	0	0	0	0	0	0
441	1	0	0	3	1	1	0	0	0	0
442	0	2	2	0	0	0	1	0	0	2
443	2	0	0	2	0	0	2	2	0	1
444	2	1	1	2	0	0	0	0	0	3
445	1	1	0	2	0	0	0	0	0	1
446	3	0	1	3	1	0	0	0	0	0
447	1	0	0	2	2	0	0	0	0	1
448	3	0	0	1	1	0	1	0	1	0
449	0	0	1	5	0	0	0	0	1	0
450	2	1	0	0	1	0	2	0	0	1

Figura 7.6: Tabla con los valores generados por el simulador en un día de ventas.

Departamento	1	2	3	4	5	6	7	Promedio
A. Básicos	208	199	181	227	232	211	217	210.7
A. Procesados	136	150	149	139	177	150	166	152.4
Lácteos	61	76	75	66	75	70	69	70.2
Carnes y Pescados	161	150	145	149	142	155	152	150.5
Desechables	62	88	85	71	71	62	60	71.2
Vinos y Licores	79	69	75	69	72	55	52	67.2
Químicos	79	89	76	78	66	62	70	74.2
Juguetería	85	53	62	67	72	61	60	65.7
Moda	87	78	79	72	75	73	65	75.5
Plásticos	68	77	78	83	66	62	70	72

Si hacemos un comparativo entre los resultados de la tabla 7.1 y los grados de membresía asignados a nuestras reglas, el resultado muestra una similitud entre nuestras etiquetas lingüísticas altas.(ver tabla 7.2).

Conjunto Fuzzy	Depto.	Promedio- ventas (reglas ECAF)	Etiqueta L.	Promedio-ventas (Simulador)	Etiqueta L.
A ₁	A. Básicos	148-220	Alta	210.7	Alta
A ₂	A. Procesados	148-220	Alta	152.4	Alta
A ₃	Lácteos	74-147	Media	70.2	Baja
A ₄	Carnes y Pesc.	148-220	Alta	150.5	Alta
B ₁	Desechables	74-147	Media	71.2	Baja
B ₂	Químicos	74-147	Media	74.2	Media
B ₃	Vinos y Licores	74-147	Media	67.2	Baja
B ₄	Juguetería	0-73	Baja	65.7	Baja
B ₅	Moda	0-73	Baja	75.5	Media
B ₆	Plásticos	0-73	Baja	72	Baja

La figura 7.7 muestra de forma gráfica los límites de cada conjuntos fuzzy y su correspondiente etiqueta lingüística; en donde podemos observar que son definidos dentro del rango de etiqueta “Alta” solo los conjuntos A₁,A₂ y A₄. Los demas conjntos estan dentro de las otras dos etiquetas restantes.

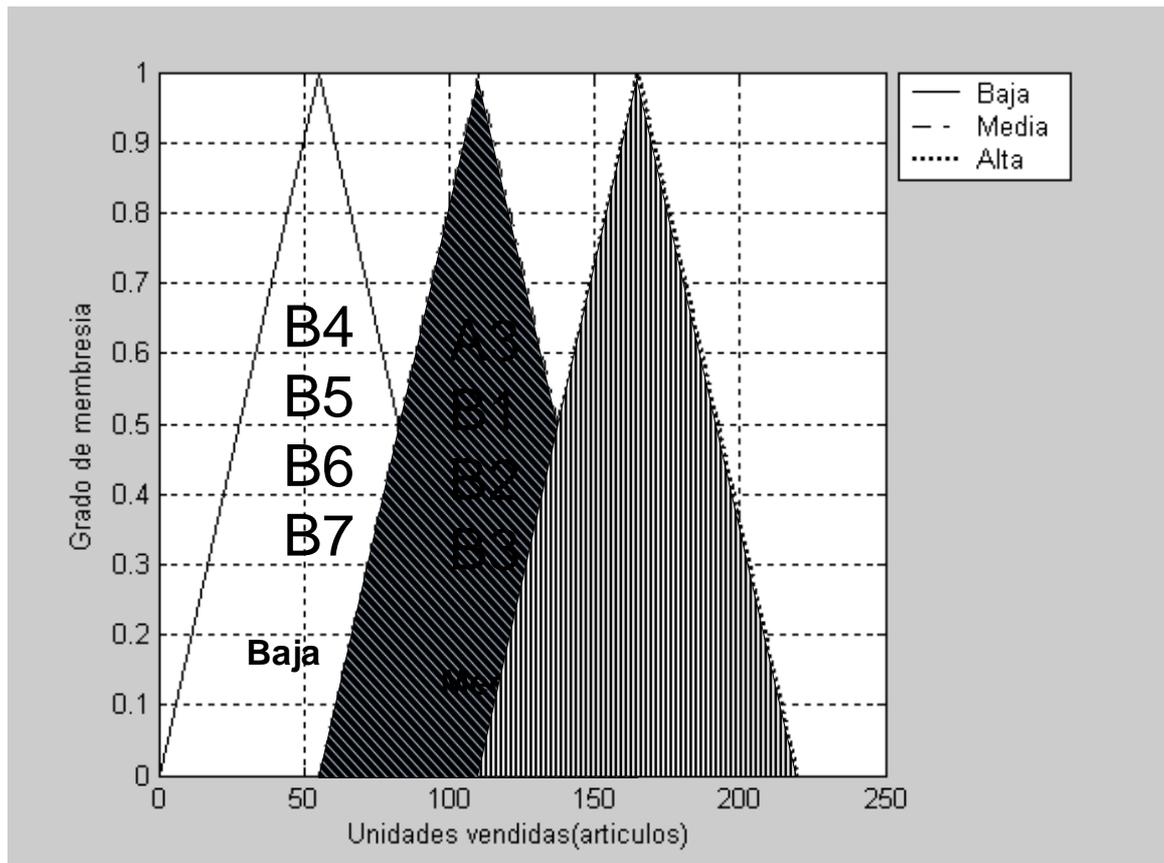


Figura 7.7: Conjuntos *fuzzy* según su frecuencia de venta (reglas ECA *Fuzzy*).

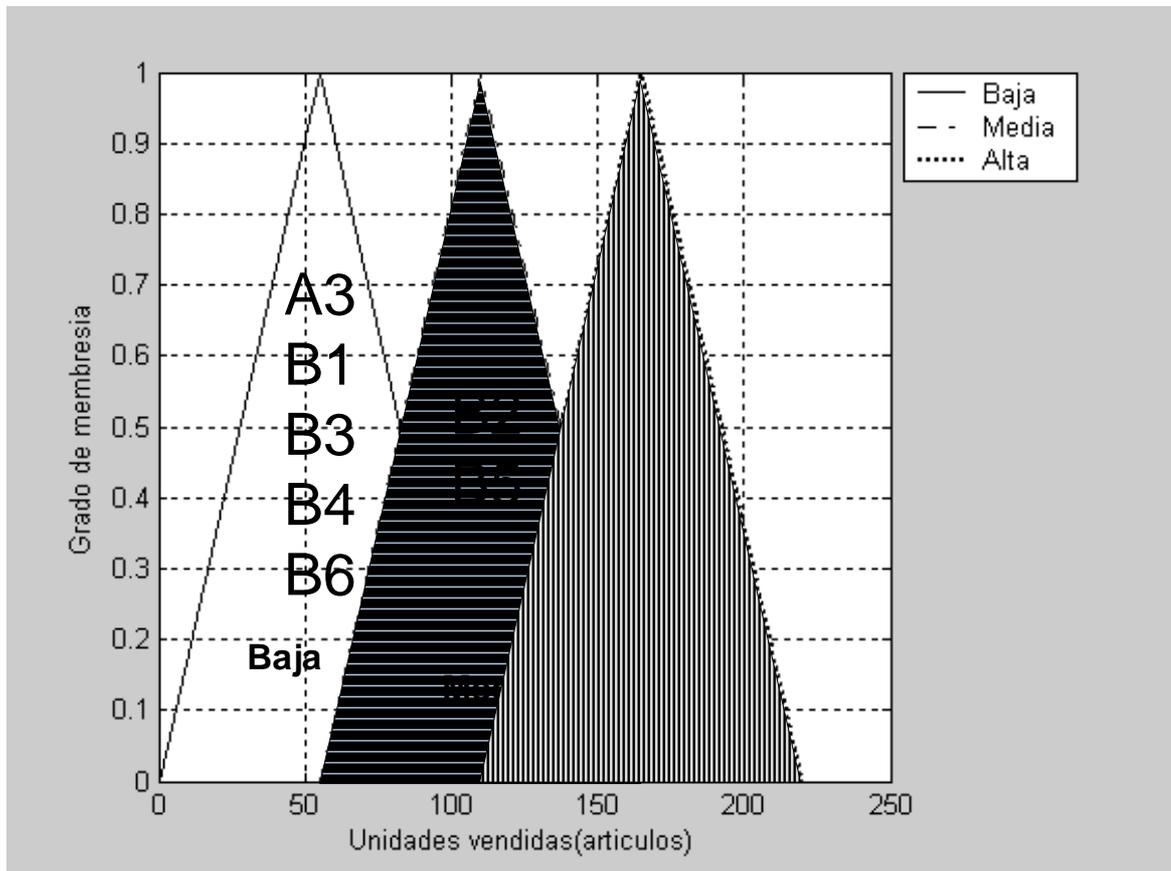


Figura 7.8: Conjuntos *fuzzy* según su frecuencia de venta(Simulador).

En contraparte, el simulador generó de la siguiente forma los siguientes resultados (Fig. 7.8), podemos observar que solo hace uso de las 3 etiquetas lingüística “Alta”, “Media” y “Baja”; por lo tanto todavía haciendo uso del simulador, las reglas para la frecuencia de venta pueden crear resultados similares.

Este simulador es solo una forma de validar nuestro datos al compararlos con los valores que empíricamente definimos para el desarrollo de nuestras reglas ECA Fuzzy. No asumimos estos valores como definitivos para validación, seguimos nuestro sentido común y empírico para asignar

frecuencias de ventas de nuestros artículos en nuestro DWH.

Capítulo 8

Conclusiones

8.1. Resultados obtenidos

Se planteó el uso de DWH_OP para la creación de reglas de análisis o también llamadas reglas de negocio en un DWH y así obtener un DWH Activo que minimice el trabajo operacional de los analistas y sirva de soporte a la toma de decisiones.

Esta interfaz genera de forma visual la implementación de reglas de análisis para toma de decisiones operacionales, siendo la primera etapa para la automatización de nuestro DWH.

Una vez que las actividades operacionales fueron automatizadas, se procedió a implementar con reglas ECA Fuzzy soluciones a la problemática de disrupción en una tienda de autoservicio, donde el principio de la lógica fuzzy fue la base para dichas reglas. Estas reglas fueron modeladas para el apoyo de la toma de decisiones, ofreciendo posibles soluciones a problemas planteadas en situaciones de escasez de artículos cuando estos aun no están en lista para reabasto en dicha tienda.

Para modelar reglas de análisis con fuzzy se establecieron parámetros a las variables lingüísticas, en donde estas fueron implementadas para ciertos comportamientos necesarios para desarrollar políticas en las reglas. Estos parámetros fueron comparados con datos ficticios generados por un simulador que se desarrolló para tal fin.

El implemento de reglas generó satisfactoriamente acciones para dar solución a ciertos eventos disparados, cumpliendo con las condiciones anticipadas en nuestras reglas, obteniendo así el comportamiento ECA que combinado con la lógica fuzzy dio lugar a reglas ECA Fuzzy generándose así

al STDE _ECAF.

8.2. Trabajo futuro

Como trabajo futuro a esta implementación y tema podría ser:

- Extender el sistema a otra toma de decisiones estratégicas,
- Agragarle al simulador otros parámetros para diferentes periodos de compras en el año
- y un detector de reglas de conflictos en la implementación del generador de reglas de análisis.

Bibliografía

- [1] Ralph Kimball, “The Data Warehouse Toolkit series”. John Wiley & Sons 1996-2004.
- [2] Thomas Thalhammer, Michael Schrefl, “Active data warehouses”, IEEE Transactions on Data & Knowledge Engineering, vol. 39, 2001, pages 241-269.
- [3] N.W. Patón, O. Diaz , “Active Database Systems”, ACM Computing Surveys, Vol. 31, No.1,1999, pp 64-103.
- [4] Joselito Medina, “Desarrollo de reglas ECA en BDA: un enfoque de Red de Petri”, Tesis de Doctorado, 2005.
- [5] Frank S. C. Tseng, “Design of a multi-dimensional query expression for document warehouses”, Information Sciences and International Journal, vol. 174, 2005, pages 55-79.
- [6] W. H. Inmon, “Building the Data Warehouse”, Edit. Wiley, Third Edition, 2002.
- [7] Hugh J. Watson, Dale L. Goodhue, “The benefits of data warehousing: why some organizations realize exceptional payoffs”, Information & Management, vol. 39, 2002, pages 491-502.
- [8] Stephen Brobst, Carrie Ballinger, “Active data warehousing”, Teradata and NCR Corporation, October, 2003, pages 1-12.
- [9] Salvatore T. March, Alan R. Hevner, “Integrated Decision Support: A Data Warehousing Perspective”, Information Systems, vol. 28, July 2003, pages 369-392.
- [10] Jeff Bedell, “Data Warehousing, Data Modeling and Design”, DSSTraining, MicroStrategy, Inc., 1997.

- [11] <http://microstrategy.com.ar/>
- [12] Sprague, R.H., y H. J. Watson, "Decision Support for Management", Edit. Prentice Hall, 1996
- [13] The PostgreSQL Global Development Group, 1996-2002.
- [14] Stonebraker M., Rowe L. A., y Hirohama M., "The Implementation of Postgres ", Transactions on Knowledge and Data Engineering 2(1), March 1990, IEEE.
- [15] Stonebraker M., "On Rules, Procedures, Caching and Views in Database Systems", June 1990, Conference on Management of Data, ACM-SIGMOD.
- [16] <http://www.sinergiacorp.com.mx>.
- [17] Poe, V. , "Building a data warehouse for decision support", Edit. Prentice Hall, 1996.
- [18] Sprague, R. H., "A Frame for the development of Decision Support Systems", Management Information Systems Quarterly, No. 4 pages 1-26, 1980.
- [19] Turban E., McLean E., Wetherbe J. (2002). Information Technology for Management: Transforming Business in de Digital Economy. Ed. Wiley.
- [20] Minuto Mauricio, y Vaisman Alejandro, "Revising aggregation hierarchies in OLAP: a rule-based approach", Data & Knowledge Engineering, vol. 25, 2003, pages 225-256.
- [21] <http://www.bitam.com/>
- [22] Herbst, H., "Bussiness Rules in systems analysis: A meta-model and repository system", Information Systems, vol. 21, No. 2, 1996, pages 147-166.
- [23] Martinez J., Aldea A., Bañares R., y Alvarado M., "TEAKS: Simulation of human performance at work to support team configuration", AAMAS Conference, 2006.
- [24] Leung R.W.K., Lau H.C.W. and C.K. Kwong, "On a responsive replenishment system: a fuzzy logic approach", Expert Systems, vol. 20, No. 1, Febrero 2003.
- [25] <http://delta.cs.cinvestav.mx/~gmorales/ldiffl/node1.html>, "Introducción a la Lógica difusa".

- [26] Guiarratano, J-Riley, G., “Sistemas Expertos, principios y programación”, Edit. Thomson Editores, Third. Edition. 2003.
- [27] Carreño Liliana, Cardona Dario, “Sistema Experto para la toma de decisiones de habitabilidad y reparabilidad en edificios después de un sismo”, Asociación Colombiana de Ingeniería Sísmica, junio de 2003.
- [28] <http://www.scielo.org.ve/scielo.php?pid=s0378-18442005000500>, “Uso de la teoría de lógica difusa en la clasificación de imágenes satelitales con coberturas mixtas: el caso urbano de Mérida”.
- [29] <http://prof.usb.ve/montbrun/ps2320/fuzzy/fuzzy.html>.
- [30] <http://es.wikipedia.org/wiki/>
- [31] Petrovic D. and Duenas Alejandra, “A fuzzy logic based production scheduling/rescheduling in the presence of uncertain disruptions”, Fuzzy sets and Systems, vol. 157, 2006, pages 2273-2285.
- [32] Serrano Victor, Alvarado Matias and Coello Carlos, “Optimization to manage supply chain disruptions using the NSGA-II”, Fuzzy sets and Systems, vol. 157, 2006, pages 2273-2285.