Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

Guía Para El Análisis De Datos Para El Apoyo A La Toma De Decisiones En Sistema De Telefonía Ip De Cisco

Jorge Luis Severino Alvarado

Profesor Guía: **Pamela Hermosilla Monckton**

Profesor Co-referente: **Iván Mercado Bermúdez**

Carrera: **Ingeniería Civil en Informática**

(Diciembre de 2007)

Dedicatoria:

A quienes han creído que se puede, y me lo han hecho saber.
A quienes han creído en mí.
A quienes me han apoyado en las buenas y sobretodo en las malas.
Esto es para ellos, esto es por ellos.

Agradecimientos:
A mis profesores por su paciencia y ánimo en este trabajo.

Resumen

El presente documento realiza un análisis de la arquitectura de voz, video y datos integrados (AVVID) propietaria de Cisco para soluciones de telefonía IP. Luego, se estudian los conceptos asociados con Data Warehousing, las capacidades de análisis de datos de SQL Server 2000 y se utiliza el algoritmo Apriori usando la herramienta libre WEKA.

Se presenta un modelo de Data Mart, como resultado de los estudios previos. Se implementa dicho modelo y se carga con datos de una base de datos real. Se construye un Cubo, y luego se utiliza el algoritmo Apriori, implementado en la herramienta WEKA, para encontrar reglas de asociación. Luego, se presentan los resultados y conclusiones del algoritmo y del proyecto.

Palabras Claves: Data Mart, Data Mining, Telefonía IP, VoIP, Callmanager Cisco, SQL Server 2000, WEKA, AVVID.

Abstract

This document analyzes the architecture of voice, video and integrated data (AVVID) owns Cisco solutions for IP telephony. Then, explores the concepts associated with Data Warehousing, the capabilities of analysis of data from SQL Server 2000 and utilizes the Apriori algorithm using the free tool WEKA.

A model of Data Mart is presented, as a result of previous studies. Is implemented that model and is loaded with data from a real database. It builds a cube, and then uses the Apriori algorithm, implemented with WEKA tool, in order to find associations rules. Finally, the results and conclusions of the algorithm and the project are presented.

Key words: Data Mart, Data Mining, IP Telephony, VoIP, Cisco CallManager, SQL Server 2000, WEKA, AVVID.

Índice

I. Índice de ilustraciones

II. Índice de tablas

III. Glosario

Apriori: Algoritmo que genera reglas de asociación. Propuesto por Rakesh Agrawal y Ramakrishnan Srikant.

Calling Search Space: Corresponde al grupo de destino de llamadas configuradas en el Callmanager, y asociadas a un usuario.

Callmanager: Elemento escencial en el esquema de telefonía IP de Cisco. Consta de Hardware (Servidor) y Software. Permite la gestión de llamadas.

Cluster (de Callmanager): Grupo de servidores Callmanager trabajando en conjunto.

Data Mart: Implementación de un Data Warehouse con un ámbito de datos y funciones de Data Warehouse más pequeño y restringido, que sirve a un departamento único o a una parte de la Organización.

Data Mining: Extracción no trivial de información implícita, desconocida previamente, y potencialmente útil desde los datos mediante técnicas de computación.

Data Warehouse: Es un conjunto de datos integrados orientados a una materia, que varían con el tiempo y que no son transitorios, los cuales soportan el proceso de toma de decisiones de la administración.

Device Pool: Corresponde a un perfil que se crea en Callmanager para agrupar teléfonos con características similares. Por ejemplo, mismo Calling Search Space, Cluster, etc.

Large Itemsets: Son conjuntos de atributos que tienen un soporte por encima de un soporte mínimo dado, generados por el algoritmo Apriori.

Lookups: Recurso de SQL Server que realiza una operación de búsqueda, con un "identificador" de una fila, para traer más información, de la fila específica, desde una tabla. Utilizado en los DTS (Data Transformation Services).

Small Itemsets: Son conjuntos de atributos que tienen un soporte por debajo de un soporte mínimo dado, generados por el algoritmo Apriori.

Soporte: El soporte, para un conjunto de atributos, es el número de transacciones que contienen a este conjunto de atributos

Subscribers: Servidores que proporcionan redundancia y componen un cluster, con la finalidad de mantener el servicio de telefonía operativo en caso de caída de uno de los servidores.

Telefonía IP: Aplicación inmediata de la tecnología VoIP, de forma que permita la realización de llamadas telefónicas ordinarias sobre redes IP u otras redes de paquetes utilizando un PC, gateways, teléfonos estándares, teléfonos IP, y otros dispositivos.

WEKA: Colección de algoritmos de Data Minig. Software para análisis de datos desarrollado por la Universidad de Waikato, Nueva Zelanda, bajo licencia GNU.

IV. Índice de abreviaturas o siglas

ARFF: Attribute-Relation File Format.

AVVID: Architecture for Voice, Video and Integrated Data.

DBMS: DataBase Management System.

DTS: Data Transformation Services.

DW: Data Warehouse.

ETL: Extract, Transform, Load.

TDT: Transform Data Task.

VoIP: Voice over IP.

WEKA: Waikato Environment for Knowledge Analysis (Entorno para Análisis del Conocimiento de la Universidad de Waikato).

Capítulo 1. Introducción

El presente informe corresponde al documento de memoria para la obtención del título de Ingeníero Civil en Informática de la Pontificia Universidad Católica de Valparaíso.

La principal motivación, nace de la necesidad de plasmar los conocimientos adquiridos en los años de estudio en la Universidad, en un problema real, en una empresa real. Para lograr lo anterior, se propone encontrar un mejor uso de los datos generados por los sistemas de telefonía en general, y de telefonía IP en particular, con los que tuvo un acercamiento durante el período de prácticas profesionales.

En el presente informe se comienza revisando el estado en lo referente a las redes y a la telefonía IP. Luego, se estudia la arquitectura en que está inmerso el particular caso de un sistema de telefonía IP de una empresa privada, basado en la solución de Cisco.

Posteriormente, se analizan las tecnologías que se utilizarán en el proyecto, y se describen las herramientas con que se trabajará.

El siguiente paso es el desarrollo del proyecto en sí, que involucra el diseño e implementación de un Data Mart, la generación de un cubo de datos, y la aplicación del algoritmo de clasificación Apriori sobre los datos del Data Mart, usando la herramienta libre WEKA.

Finalemente, se entregan las conclusiones, en las que se evalúa el desarrollo del proyecto.

Capítulo 2. Definición de Objetivos

2.1 Objetivo general

El objetivo general de este proyecto es: "Presentar una guía para el desarrollo de un análisis de los datos históricos que genera un sistema de telefonía IP, usando un Data Mart y un algoritmo de Data Mining. Para así, poder dar un soporte a las decisiones de los encargados de un sistema de telefonía IP".

Muchas veces, se toman decisiones de manera intuitiva, corriendo en riesgo de errar. Tomar decisiones con un mayor grado de conocimiento sobre el tema es de gran interés, tanto para aquellas organizaciones que posean ya un sistema de telefonía sobre IP, como por aquellas que estén buscando cifras y pruebas concretas para justificar su implementación.

Así, mediante este proyecto, aquellas empresas que posean un sistema de telefonía IP basados en Callmanager de Cisco, tendrán una visión completa de su sistema, e información agregada de gran valor.

La guía pretende mostrar los pasos a seguir para poder desarrollar el análisis, habiendo estudiado las propiedades de SQL Server como motor de bases de datos, las características del modelo de bases de datos que utiliza un Cisco Callmanager, y la herramienta open source WEKA. Así, un tomador de decisiones puede saber de grupos, usos y abusos de su sistema de telefonía.

2.2 Objetivos específicos

- Estudiar y comprender el funcionamiento del sistema de telefonía IP de Cisco.
- Obtener los datos que pueden ser más relevantes del Callmanager.
- Estudiar e investigar sobre Data Warehousing, tanto en su desarrollo y modelado, como de Software disponibles.
- Obtener un modelo de Data Mart ad-hoc, que permita responder a las consultas que el proyecto se plantea.
- Utilizar herramientas, y algoritmos de Data Minig sobre el Data Mart.
- Presentar resultados.

Capítulo 3. Estado del arte

Las redes de transmisión de datos actuales son bastante rápidas y eficientes, y se han usado por años en las comunicaciones. Hoy en día, ya es un hecho la transmisión de voz por medio de éste tipo de redes en todo en el mundo, y en Chile ha ido creciendo considerablemente en los últimos años, sobre todo a nivel de empresas, donde la convergencia de las redes de datos y de telefonía han significado una real disminución de los costos de telefonía, justificando la inversión.

Así lo ha pensado Cisco, empresa protagonista del mercado de las comunicaciones a nivel global, y que ha diseñado la llamada: "Arquitectura de Voz, Video y Datos Integrados" o AVVID (del inglés Architecture for Voice, Video and Integrated Data). Con Cisco AVVID, las empresas pueden optimizar la capacidad de la red, utilizar los recursos de forma más eficiente y habilitar nuevas aplicaciones de negocio basadas en Internet y en el uso de tecnologías multimedia para aumentar su ventaja competitiva.

Dentro de la AVVID se encuentra el sistema de comunicaciones de voz por redes de datos, que incluye desde teléfonos IP, sistemas de mensajería (Cisco Unity), sistema gestor de llamadas (Cisco Callmanager), etc.

De estos elementos se considerará, en el presente proyecto, el Callmanager, el cual, consta de una parte de Hardware (Servidor) y Software (SW Callmanager), y esta encargado de registrar los teléfonos IP, usuarios, definir espacios para salidas de llamadas, y todo lo necesario para gestionar llamadas. Por lo mismo, genera gran cantidad de datos que son almacenados en varias bases de datos ubicadas en el mismo servidor (o eventualmente en los servidores Subscribers, si existen más Callmanager conectados en cluster).

Aprovechando esta gran cantidad de datos, el proyecto tiene la intención de obtener información agregada, de alto nivel, implementando un Data Mart y luego aplicando algoritmos de Data Mining sobre él, ya que *"las tecnologías de información están para ayudar al usuario (ejecutivo, administrador, analista) a tomar mejores y más rápidas decisiones"* [Martí, 2005].

No hay mucha información en la Internet sobre análisis de datos del sistema de telefonía IP de Cisco, pero existen trabajos que explotan los datos generados por el Callmanager para hacer sistemas de tarificación de llamadas. De hecho existen empresas Chilenas que se dedican a comercializar éstos productos Software (por ejemplo: el tarificador AQCT de la empresa Ermez), así como también existen empresas Chilenas que prestan servicios de telefonía IP.

El impacto de la telefonía IP no ha llegado masivamente a los hogares Chilenos aún, pero el crecimiento explosivo de los últimos años a nivel de empresas augura la muerte del sistema de telefonía actual, así lo afirma Roberto de la Mora, Gerente Senior de Comunicaciones IP de Cisco, quien dice: "*En América Latina se venden más o menos 1.000.000 de líneas de telefonía empresarial por año. Si seguimos a este ritmo, más del 10% de todas las unidades vendidas van a ser IP y van a ser Cisco. Y este crecimiento es exponencial. Y si sumamos lo que los demás fabricantes venden, significa que el 40 % de la telefonía en América Latina va a ser IP*" [de la Mora, 2006]

Incluso la subsecretaría de telecomunicaciones de Chile (SubTel) reconoce el potencial avance de la voz sobre IP en Chile, debido a los beneficios económicos que ofrece: *"Dentro del sector (telecomunicaciones) se destacan el desarrollo de la telefonía móvil y de los servicios de comunicación prestados principalmente a través de Internet, como el correo electrónico, el chat y la voz sobre Internet. Por otro lado, el gran impulso que ha tenido el desarrollo de nuevas tecnologías ha generado una reducción en el tamaño de la inversión necesaria para satisfacer la demanda. Estas nuevas tecnologías, que son capaces de gestionar tráfico convencional telefónico como tráfico de datos, conlleva a un ahorro para los operadores en términos de inversión debido a que todos los servicios pueden ofrecerse a través de una única red (convergencia), en definitiva los niveles de inversión necesarios para prestar los mismos servicios, son menores que en el pasado. Los usuarios verán en el futuro este cambio tecnológico como un aumento de las capacidades y la calidad de los servicios prestados a través de las mismas redes de acceso telefónicas actuales y a través de nuevas redes"* [Subtel, 2005].

Luego, y considerando la situación actual en nuestro país respecto de la tecnologías de voz sobre IP, resulta interesante el análisis mediante las técnicas informáticas anteriormente mencionadas.

Pero, ¿Por qué un Data Mart?, bueno, porque "*soporta el procesamiento informático al proveer una plataforma sólida, a partir de los datos históricos para hacer el análisis. Facilita la integración de sistemas de aplicación no integrados. Organiza y almacena los datos que se necesitan para el procesamiento analítico, informático sobre una amplia perspectiva de tiempo"* [INEI Perú, 1997], y sobre el Data Mart, se puede obtener información sobre tendencias, predicciones, grupos, etc. Utilizando herramientas de análisis o Data Mining, algunas de libre distribución como WEKA [WEKA, 2008], es posible esto.

Así, el presente proyecto presentará una solución de análisis de datos sobre éste particular sistema de telefonía IP, el planteado por Cisco, y que reside en la AVVID de Cisco.

Capítulo 4. Análisis del entorno

En ese capítulo, se presentan los resultados sobre el estudio del entorno en que se encuentra el problema, contextualizándolo.

AVVID es la arquitectura diseñada por Cisco en la que reside su sistema de telefonía IP, y está compuesta por 3 componentes fundamentales (Ilustración 4.1):

1. Los **clientes** son las estaciones o instrumentos de los usuarios finales, que utilizan para comunicarse con la red u otros usuarios. Por ejemplo: PCs, teléfonos, cámaras de video, etc.
2. Las **aplicaciones** para la AVVID de Cisco están escritas para un ambiente de estándares abiertos y, como tal, será provisto por Cisco y desarrolladores de aplicaciones. Por ejemplo, Call Center, respuesta de voz interactiva (IVR), y mensajería unificada (Cisco Unity).
3. La **infraestructura** es la red en donde los clientes y las aplicaciones residen. La red está basada en IP, usando inteligencia inherente en las plataformas, para proveer de flexibilidad y escalabilidad para soportar la convergencia de distintos medios. Ejemplos de dispositivos de red son los switch Catalyst, los routers Cisco, y los voice gateways.

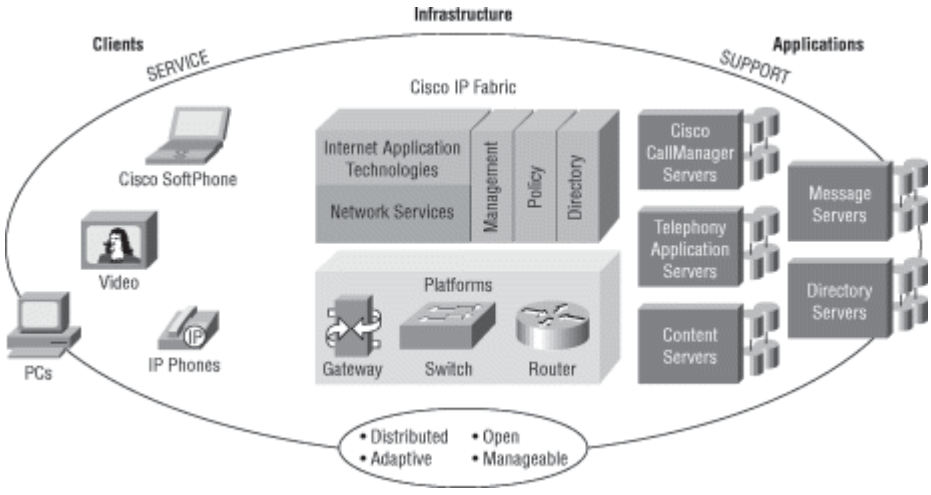


Ilustración 4.1 Cisco AVVID

De este esquema, utilizaremos el Callmanager como fuente de datos. Este es un servidor que se conecta a la red de datos y permite la gestión y gestación de llamadas VoIP (Ilustración 4.2).

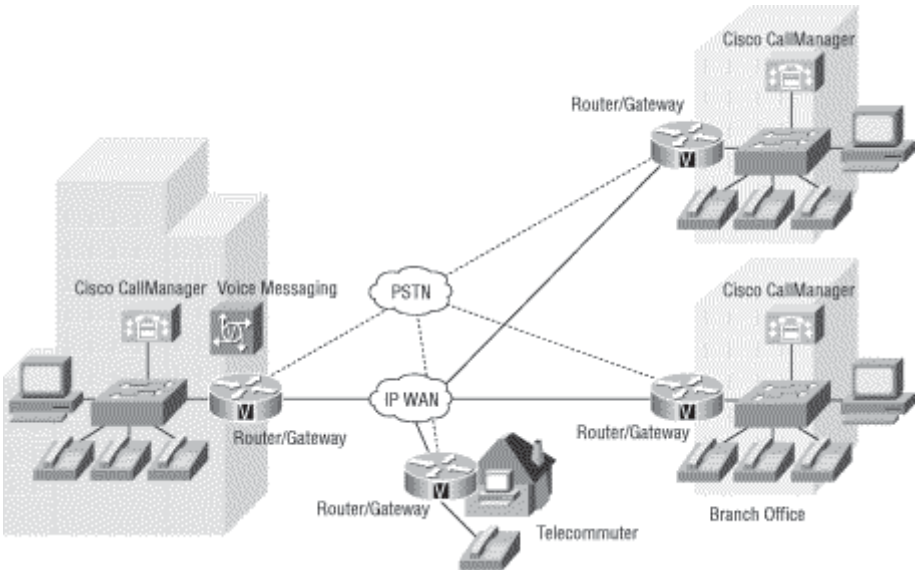


Ilustración 4.2 Esquema de uso de Callmanager

El Callmanager es también un elemento Software, con el cual, se gestionan las cuentas de usuario, números de teléfonos, espacio de llamadas salientes (Calling Search Space), etc.

Luego, se tiene un Callmanager que contiene la información "oficial" del sistema telefónico, llamado **Publisher**, y eventualmente otros Callmanagers que contienen una réplica del contenido del Publisher, y son llamados **Subscribers**. A éste esquema se le llama clustering (Ilustración 4.3).

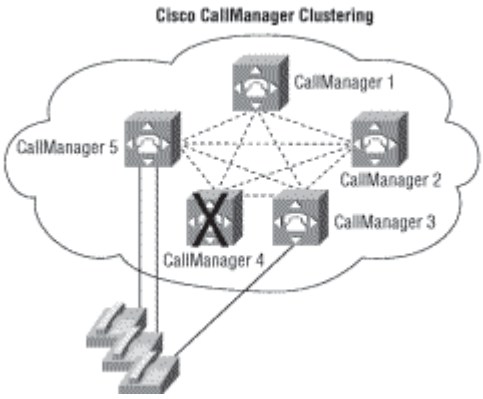


Ilustración 4.3 Callmanagers en Cluster

Así, los datos que son de interés para la generación del Data Mart se encuentra en varios Callmanagers, siendo el Publisher el escogido para la extracción, ya que contiene la información "oficial", mientras que los Subscribers son empleados para mejorar la calidad del servicio telefónico, y/o para la tolerancia ante una caída de un Callmanager, obteniendo excelente disponibilidad.

El Software del Callmanager versión 4.1 (3), se instala sobre Windows 2000, y posee un motor de bases de datos SQL Server 2000. Cuenta con varias bases de datos, siendo las más importantes:

- CCM0300: Contiene los datos de registro de usuarios, teléfonos, Calling Search Space, etc.
- CDR: Contiene el registro de las llamadas, logs, entre otros valores. Usada para la generación de reportes. Tablas como "Calldetailrecord" serán de gran importancia.

Capítulo 5. Análisis de tecnologías de análisis de datos a utilizar

5.1 Conceptos de Data Warehousing

Los sistemas de información operacionales generan gran cantidad de datos, alimentando las bases de datos relacionales con las que operan, pero con el tiempo, estos datos pueden dejar de usarse. Estos datos históricos pueden ser utilizados para obtener información de interés para los tomadores de decisiones en una organización.

Así es como surge la idea del Data Warehousing, el cual, ayuda a minimizar el análisis de grandes volúmenes de datos con más velocidad y precisión, y que tiene como objetivos básicamente [Pérez de Armas, 2003]:

1. Comprender las necesidades de los usuarios por áreas dentro del negocio.
2. Determinar qué decisiones se pueden tomar con la ayuda del DW.
3. Seleccionar un subconjunto del sistema de fuentes de datos que sea el más efectivo y procesable para presentar el DW.
4. Asegurar que los datos sean precisos, correctos y confiables y que mantengan la consistencia.
5. Monitorear continuamente la precisión y exactitud de los datos y el contenido de los reportes generados.
6. Publicar los datos.

"*Generalmente, los sistemas transaccionales o OLTP usan estructuras normalizadas, en las cuales se optimizan las inserciones y actualizaciones de artículos e incluso algunas selecciones, pero es menos probable que el sistema se organice de forma tal que produzca reportes eficientes para datos resumidos con cierta jerarquía. Y es aquí donde debería usarse el DW, que usa los datos relevantes de fuentes existentes y los combina en una estructura que ha sido optimizada para las selecciones.* "[Pérez de Armas, 2003]

5.1.1 *¿Qué es Data Warehousing?*

Data Warehousing es un "*conjunto de herramientas que permite construir un Data Warehouse*" [Martí, 2005], Data Warehousing esta compuesto principalmente de:

- BD de apoyo a la gestión.
- Herramientas de integración de sistemas.
- Herramientas de análisis de datos.
- Hardware ad-hoc.

5.1.2 *¿Qué es un Data Warehouse?*

Existen muchas definiciones para el DW, la más conocida fue propuesta por Bill Inmon (considerado el padre de las Bases de Datos) en 1992: "*Un DW es una colección de datos orientados a temas, integrados, no-volátiles y variante en el tiempo, organizados para soportar necesidades empresariales"* [Martí, 2005]. En 1993, Susan Osterfeldt publica una definición que sin duda acierta en la clave del DW: "*Yo considero al DW como algo que provee dos beneficios empresariales reales: Integración y Acceso de datos. DW elimina una gran cantidad de datos inútiles y no deseados, como también el procesamiento desde el ambiente operacional clásico*" [Gutiérrez, 2006].

En fin, podemos decir que es una técnica para consolidar y administrar datos de variadas fuentes con el propósito de responder preguntas de negocios y tomar decisiones, y se caracteriza por ser [Padrón, 2003]:

- **Temático**: Los datos están almacenados por materias o temas (clientes, campañas, productos). Estos se organizan desde la perspectiva del usuario final, mientras que en las Bases de Datos operacionales se organizan desde la perspectiva de la aplicación, con vistas a lograr una mayor eficiencia en el acceso a los datos

- **Integrado**: Todos los datos almacenados en el *DW* están integrados. Las bases de datos operacionales orientadas hacia las aplicaciones fueron creadas sin pensar en su integración, por lo que un mismo tipo de datos puede ser expresado de diferente forma en dos bases de datos operacionales distintos (Por ejemplo, para representar el sexo: 'Femenino' y 'Masculino' o 'F' y 'M').

- **No volátil**: Únicamente hay dos tipos de operaciones en el *DW*: la carga de los datos procedentes de los entornos operacionales (carga inicial y carga periódica) y la consulta de los mismos. La actualización de datos no forma parte de la operativa normal de un *DW*.

- **Histórico**: El tiempo debe estar presente en todos los registros contenidos en un DW. Las bases de datos operacionales contienen los valores actuales de los datos, mientras que los DW contienen información actual y resúmenes de esta en el tiempo.

5.1.3 *¿Qué es OLAP?*

OLAP (Procesamiento Analítico en Línea) son "*procesos orientados a analizar temas de interés específicos del tomador de decisiones*" [Martí, 2005]. Estrictamente, OLAP define el comportamiento de un sistema de análisis de datos y elaboración de información de:

- Sólo Consulta
- Consultas pesadas y no predecibles
- Gran volumen de información histórica
- Operaciones lentas

5.1.4 *¿Qué es OLTP?*

OLTP (Procesamiento Transaccional en Línea) son "*procesos orientados a las transacciones, situación típica en una base de datos operacional*" [Martí, 2005], en la cual se define el comportamiento habitual de un entorno operacional de gestión de:

- Altas/Bajas/Modificaciones/Consultas
- Consultas rápidas y escuetas
- Poco volumen de información, muchos datos
- Transacciones rápidas
- Gran nivel de concurrencia

5.1.5 *¿Qué es un data mart?*

Es un pequeño Data Warehouse, para un determinado número de usuarios, y para un área funcional especifica de la compañía. También podemos definir que un Data Mart es un subconjunto de un Data Warehouse para un propósito específico.

Su función es apoyar a otros sistemas para la toma de decisiones. En grandes empresas, es muy efectivo, pero la desventaja más notoria es cuando se deben integrar todos los Data Marts para la creación de un Data Warehouse único.

5.1.6 *¿Qué es data mining?*

Data Mining es una tecnología de soporte para usuario final, cuyo objetivo es extraer conocimiento útil y utilizable a partir de la información contenida en las bases de datos, generalmente, de empresas.

Los objetivos de un sistema Data Mining permite analizar factores de influencia en determinados procesos, predecir o estimar variables o comportamientos futuros, sementar o agrupar ítems similares, además de obtener secuencias de eventos que provocan comportamientos específicos.

Los sistemas de Data Mining se desarrollan bajo lenguajes de última generación basados en la inteligencia artificial y utilizando métodos matemáticos y computacionales, tales como:

- Redes neuronales: paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales.
- Árboles de decisión: modelo de predicción, representa y categoriza una serie de condiciones que suceden de forma sucesiva.
- Clustering: procedimiento de agrupamiento de datos según caracteísticas comunes.

La tecnología Data Mining soporta también sofisticadas operaciones de análisis tales como aplicaciones de detección de fraude.

5.2 Arquitectura de un Data Warehouse

Los bloques funcionales que se corresponden con un sistema de información completo que utiliza un DW se muestran gráficamente en la Ilustración 5.1.

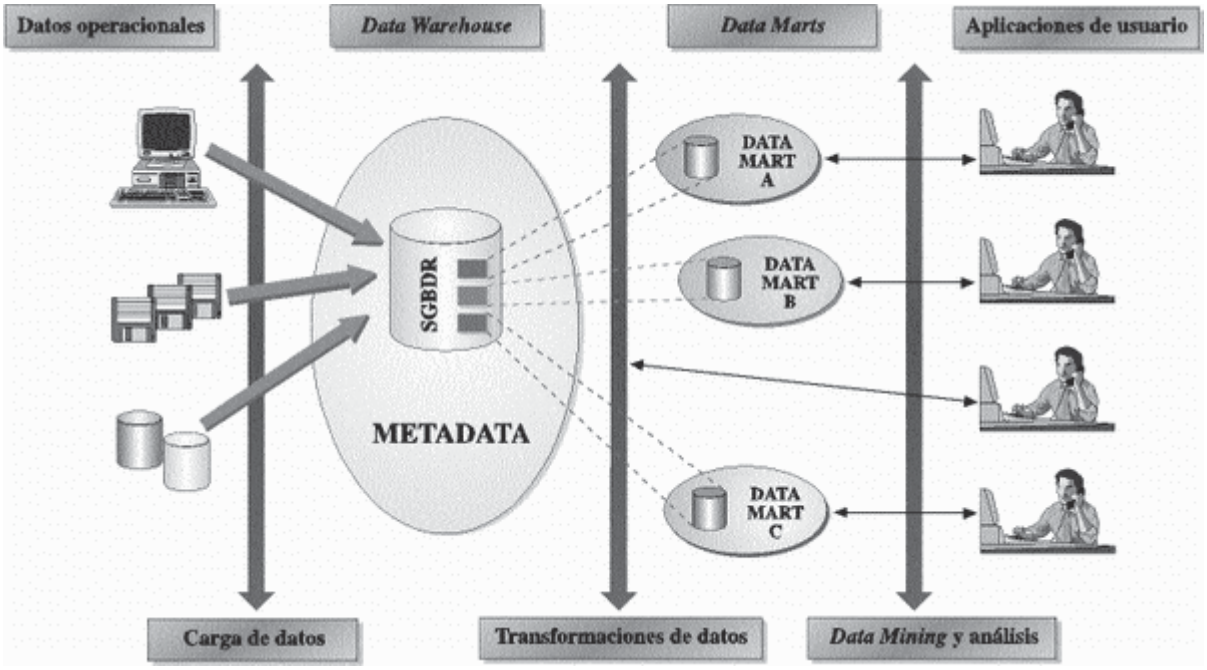


Ilustración 5.1 Arquitectura de un Data Warehouse

- **Nivel operacional:** Contiene datos primitivos (operacionales) que están siendo permanentemente actualizados, usados por los sistemas operacionales tradicionales que realizan operaciones transaccionales.

- **Data Warehouse:** Contiene datos primitivos correspondientes a sucesivas cargas del DW y algunos datos derivados. Los datos derivados son datos generados a partir de los datos primitivos al aplicarles algún tipo de procesamiento (resúmenes).

- **Nivel departamental (Data Mart):** Contiene casi exclusivamente datos derivados. Cada departamento de la empresa determina su nivel departamental con información de interés a dicho nivel. Va a ser el blanco de salida sobre el cual los datos en el Data Warehouse son organizados y almacenados para las consultas directas por los usuarios finales, los desarrolladores de reportes y otras aplicaciones.

- **Nivel individual:** Contiene pocos datos, resultado de aplicar heurísticas, procesos estadísticos, etc., a los datos contenidos en el nivel anterior. El nivel individual es el objetivo final de un Data Warehouse. Desde este nivel accederá el usuario final y se

podrán plantear diferentes hipótesis, así como navegar a través de los datos contenidos en el Data Warehouse.

5.3 Ciclo de vida de un Data Warehouse

El ciclo de vida de un Data Warehouse, puede dividirse en 3 etapas principales (Ilustración 5.2):

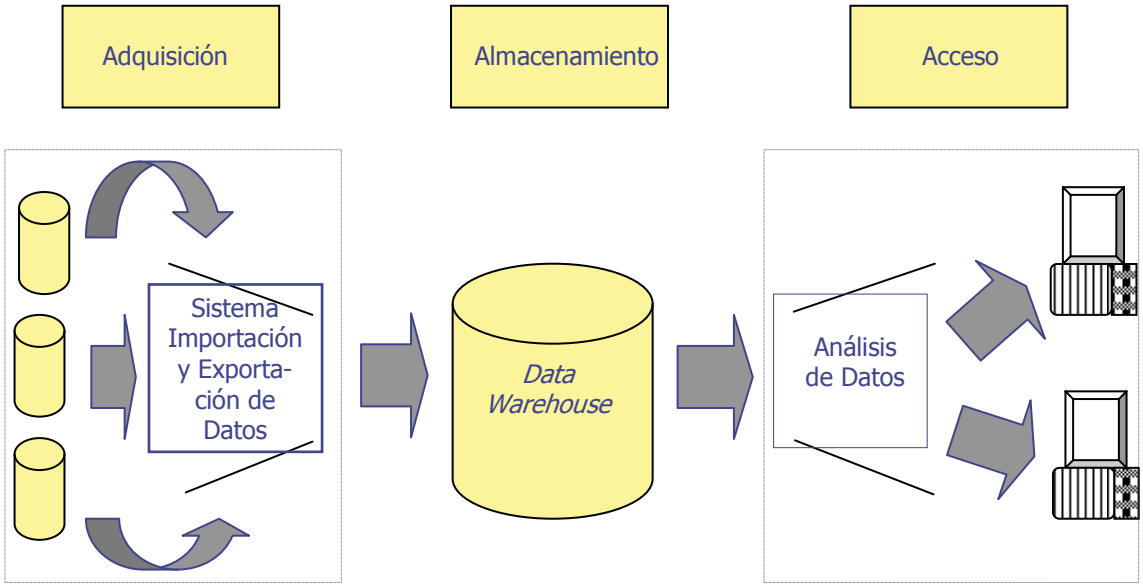


Ilustración 5.2 Ciclo de vida de un Data Warehouse

5.3.1 *Adquisición de datos*

El encargado del mantenimiento del almacén de datos es el sistema ETL (Extracción – Transformación – Carga) (Ilustración 5.3):

- La construcción del sistema ETL es responsabilidad del equipo de desarrollo del DW.
- El sistema ETL es construido específicamente para cada DW, aproximadamente, 50% del esfuerzo.
- En la construcción del ETL se pueden utilizar herramientas del mercado o programas diseñados específicamente.

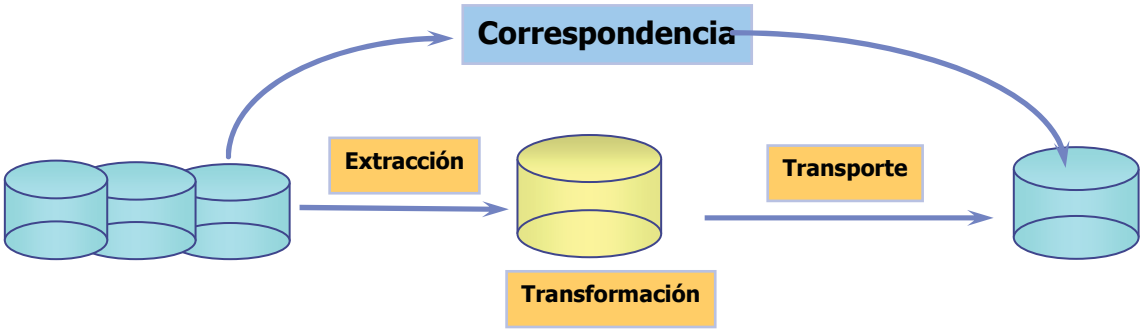


Ilustración 5.3 Proceso ETL

a) **Extracción de Datos:** En esta etapa, se deben tener en cuenta los siguientes aspectos:

- Identificación de los datos que han cambiado
- Extracción (lectura) de datos.
- Obtención de agregaciones
- Mantenimiento de metadatos

Si los datos operacionales están mantenidos en un SABD relacional, la extracción de datos se puede reducir a consultas en SQL o rutinas programadas. Si los datos operacionales están en un sistema propietario o en archivos de texto, hipertexto u hojas de cálculo, la extracción puede ser muy difícil.

Antes de llevar a cabo la extracción, hay que identificar los cambios, determinando los datos operacionales (relevantes) que han sufrido una modificación desde el último refresh.

b) **Transformación de los Datos:** Durante ésta etapa se considera:

- La limpieza y transformación de datos: eliminar datos, corregir y completar datos, eliminar duplicados.
- Integración de datos: datos con formatos distintos, unificados a un formato único.
- Obtención de agregaciones: se generan datos calculados, obtenidos de operaciones que consideran varios datos operacionales.

c) **Carga de Datos:** consiste en mover los datos desde las fuentes operacionales o el almacenamiento intermedio hasta el DW y cargar los datos en las correspondientes estructuras de datos, lo cual, puede consumir mucho tiempo. Se considera una carga inicial, y un *refresh* periódico de su contenido a través del tiempo.

5.3.2 *Almacenamiento*

Al modelar se debe considerar aspectos como:

- Archivos extremadamente grandes.
- Datos con un alto grado de interdependencia con los datos de otras tablas.
- El acceso principal es de tipo lectura.
- Las consultas accedan, en muchos casos, un gran número de archivos.
- Los datos necesitan ser periódicamente actualizados desde distintas fuentes.
- Muchos de sus datos son históricos

5.3.3 *Acceso*

Existen tres tipos de técnicas que permiten al usuario final realizar análisis de los datos, que son:

a) **Consultas y Reportes**: análisis dirigido por el analista.

b) **Análisis Multidimensional (OLAP)**: análisis asistido por el analista. La idea del análisis multidimensional es facilitar la consulta y análisis al usuario al presentar una visión muy sencilla de los datos, muy similar a la forma como él ve la organización. Por ejemplo, un usuario puede solicitar que a información se analice para mostrar una hoja de cálculo que muestre todas las ventas en una ciudad de un producto en particular en un mes en particular, y luego ver la comparación de otros productos en la misma ciudad para el mismo periodo.

c) ***Data Mining****:* análisis dirigido por los datos. Se buscan descubrir patrones, relaciones, reglas, asociaciones o incluso excepciones que sean útiles para la toma de decisiones.

Ya construido el DW, es de interés para la empresa que llegue la información a la mayor cantidad de usuarios pero, por otro lado, se tiene sumo cuidado de protegerla contra posibles 'hackers', 'snoopers' o espías (seguridad).

Además, se deben realizar actividades de backup y restauración de la información, tanto de la almacenada en el DW como de la que circula desde los sistemas fuente al almacén.

5.4 Aspectos de diseño de un Data Warehouse

El diseño de los Data Warehouse (Ilustración 5.4) no es muy diferente al diseño de los sistemas operacionales tradicionales. Se pueden considerar los siguientes puntos:

a) Los usuarios de los DW usualmente no conocen mucho sobre sus requerimientos y necesidades como los usuarios operacionales.

b) El diseño de un DW, con frecuencia involucra lo que se piensa en términos más amplios y con conceptos del negocio más difíciles de definir que en el diseño de un sistema operacional. Al respecto, un DW está bastante cerca a Reingeniería de los Procesos del Negocio (Business Process Reengineering).

"A pesar que el diseño del DW es diferente al usado en los diseños tradicionales, no es menos importante. El hecho que los usuarios finales tengan dificultad en definir lo que ellos necesitan, no lo hace menos necesario. En la práctica, los diseñadores de DW tienen que usar muchos "trucos" para ayudar a sus usuarios a "visualizar" sus requerimientos" [Peralta, 2001].

Como en los sistemas de bases de datos tradicionales, el proceso de diseño del DW puede dividirse en tres etapas secuenciales: diseño conceptual, diseño lógico y diseño físico.

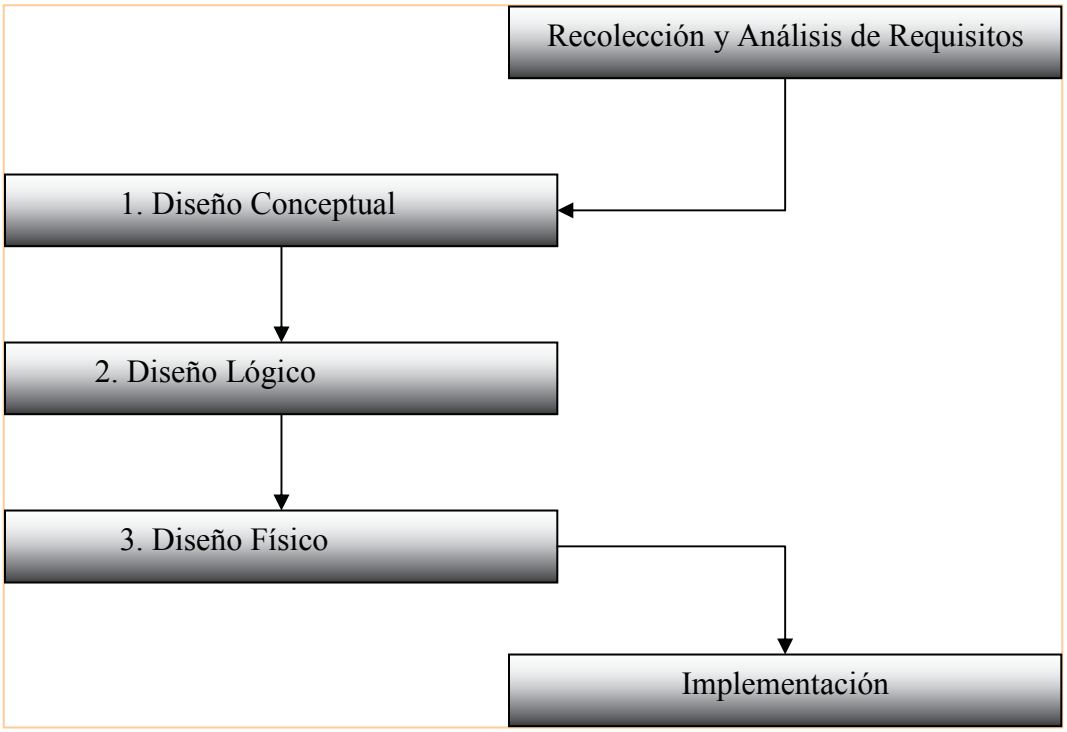


Ilustración 5.4 Esquema representativo del Diseño de un Data Warehouse

En la etapa de diseño conceptual se construye un esquema conceptual de la realidad a partir de los requerimientos y/o bases fuentes. Dicho esquema conceptual es enriquecido con requerimientos de performance y almacenamiento durante la etapa de diseño lógico, y a partir de él se genera un esquema lógico, que es dependiente del tipo de modelo y tecnología de DBMS. *"Hay dos familias de esquemas lógicos: relacionales y multidimensionales, y actualmente se están considerando esquemas híbridos. Por último, en la etapa de diseño físico se implementa el esquema lógico en el manejador de bases de datos elegido, teniendo en cuenta técnicas de optimización física, como son: índices particiones, etc."*. [Peralta, 2001].

5.4.1 *Diseño conceptual*

El diseño conceptual tiene por objetivo la construcción de una descripción abstracta y completa del problema. Comienza con el análisis de requerimientos de los usuarios y de reglas de negocio, y finaliza con la construcción de un esquema conceptual expresado en términos de un modelo conceptual.

En una primera fase se seleccionan los objetos relevantes para la toma de decisiones, y se especifica el propósito de utilizarlos como dimensiones y/o medidas. Para realizar dicha selección hay dos grandes enfoques que se basan respectivamente en un análisis de requerimientos, o en un análisis de las bases de datos fuente. Hay enfoques intermedios.

- En el enfoque basado en requerimientos se analizan los requerimientos de los usuarios, y se identifican en ellos los hechos, dimensiones y medidas relevantes. La realidad se modela como un conjunto de cubos multidimensionales, que se obtienen a partir de los hechos, dimensiones y medidas identificados.

  Los trabajos de diseño conceptual que siguen este enfoque parten de un relevantamiento de requerimientos ya realizado y proponen modelos para formalizarlos.

- En el enfoque basado en las bases fuentes se construyen cubos multidimensionales transformando un esquema conceptual de las bases fuentes. Como modelo conceptual de las fuentes, en general se utiliza el modelo E/R. Las diferentes metodologías comienzan por identificar en el esquema fuente los posibles hechos relevantes para la toma de decisiones. A partir de los hechos identificados navegan por las entidades y relaciones construyendo las jerarquías de las dimensiones.

5.4.2 *Diseño lógico*

La etapa de diseño lógico toma como entrada un esquema conceptual y genera un esquema lógico relacional o multidimensional. La dificultad principal es encontrar un esquema lógico que satisfaga no sólo los requerimientos funcionales de información, sino también requerimientos de performance en la realización de consultas complejas de análisis de datos. Esto tiene particular impacto en el caso de usarse bases relacionales, ya que las consultas de análisis de datos incluyen operaciones muy costosas para DBMS relacionales.

El resultado de esta etapa es la especificación formal de un esquema lógico para el DW. Los modelos propuestos incluyen materialización de vistas, modelos específicos basados en el modelo relacional y optimizados para consultas OLAP, e implementaciones multidimensionales, en general propietarias de los manejadores.

5.4.3 *Diseño físico*

Consiste en plasmar en la práctica, los diseños lógicos de la fase anterior. Incluye la construcción de programas que creen y modifiquen las bases de datos, que extraigan datos de las fuentes, programas para transformación de datos tales como integración, resumen y adición, programas para la actualización de los datos, programas para búsquedas en bases de datos muy grandes.

5.5 Modelo de datos multidimensional

Pertenece a la etapa de diseño lógico. En un esquema multidimensional se representa una actividad que es objeto de análisis (hecho) y las dimensiones que caracterizan la actividad (dimensiones).

La información relevante sobre el hecho (actividad) se representa por un conjunto de indicadores (medidas o atributos de hecho). La información descriptiva de cada dimensión se representa por un conjunto de atributos (atributos de dimensión).

En el enfoque basado en vistas, el DW se ve como un conjunto de vistas materializadas de las bases fuentes. Estos trabajos no se centran en la representación de conceptos multidimensionales, como dimensiones y medidas, sino en materializar algunas vistas para lograr performance en un conjunto dado de consultas.

Otro enfoque consiste en definir estructuras (dentro del modelo relacional) que optimicen las consultas que se realizarán al DW. Dichas consultas contienen gran número de “joins” y sumarizaciones que degradan la performance del sistema.

Ralph Kimball[1] propone el modelo estrella, que consiste de una gran tabla central conteniendo información sobre los hechos, y tablas más pequeñas (relacionadas a la tabla de hechos) con información sobre las dimensiones. Kimball propone también lineamientos prácticos para construir un esquema estrella pero no presenta una metodología general.

[1] Ralph Kimball es Ph. D. en Ingeniería léctrica de la Universidad de Stanford. Es considerado como el "Gurú" del Data Warehousing.

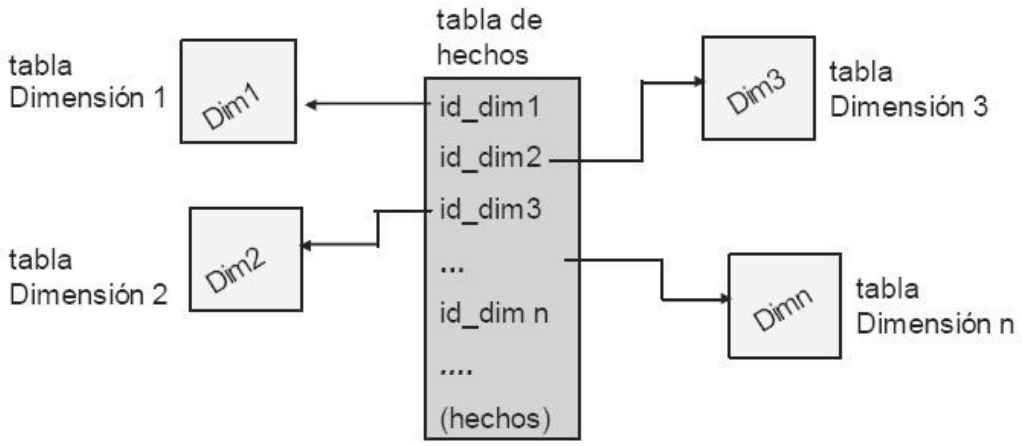


Ilustración 5.5 Modelo estrella multidimensional

También existe el modelo copo de nieve, en que se desglosan las dimensiones. Pero generalmente es suficiente el modelo estrella (Ilustración 5.5), en que se agregan los datos en una tabla de hechos, siendo éstos centralizados, mientras que las dimensiones se ubican a su alrededor.

5.5.1 *Pasos en el diseño de Data Warehouse*

En este punto se presentan 9 pasos a considerar en el diseño de Data Warehouse [Martí, 2005].

a) **Elegir un "proceso" de la organización para modelar**. Entendiendo por proceso a toda actividad de la organización soportada por un OLTP del cual se puede extraer información con el propósito de construir el DW.

   Ejemplos: Pedidos (de clientes), Compras (a suministradores), Facturación, Envíos, Ventas, Inventario, etc.

b) **Decidir el grano (nivel de detalle) de representación**. Es el nivel de detalle al que se desea almacenar información sobre la actividad a modelar. Define el nivel atómico de datos en el almacén de datos. Determina el significado de las tuplas de la tabla de hechos. Determina las dimensiones básicas del esquema: transacción en el OLTP, información diaria, información semanal, información mensual, etc.

c) **Identificar las dimensiones que caracterizan el proceso**. Entendiendo por dimensiones a las dimensiones que caracterizan la actividad al nivel de detalle que se ha elegido.

   - Tiempo: dimensión temporal: ¿cuándo se produce la actividad?

- Producto: dimensión ¿cuál es el objeto de la actividad?
- Almacén: dimensión geográfica: ¿dónde se produce la actividad?
- Cliente: dimensión ¿quién es el destinatario de la actividad?

De cada dimensión se debe decidir los atributos (propiedades) relevantes para el análisis de la actividad. Entre los atributos de una dimensión existen jerarquías naturales que deben ser identificadas (día-mes-año).

d) **Decidir la información a almacenar sobre el proceso**. Se definen los hechos, es decir, la información (sobre la actividad) que se desea almacenar en cada tupla de la tabla de hechos y que será el objeto del análisis. Ejemplos: Precio, Unidades, Importe, etc.

e) **Almacenar precálculos en la tabla de hechos**. Se trata de analizar si ciertas medidas derivadas pueden ser útiles (aunque redundantes).

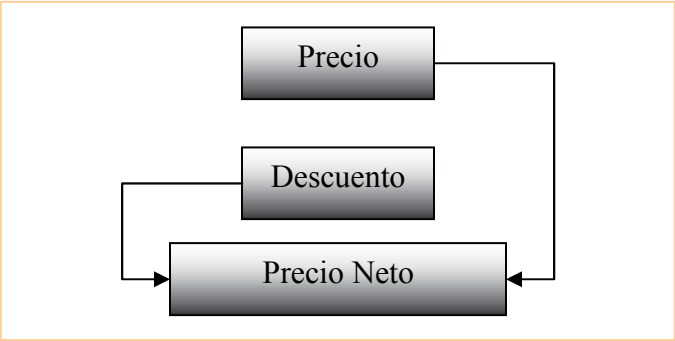


Ilustración 5.6 Ejemplo de medidas derivadas

Por ejemplo, el "Precio neto" puede derivarse de "precio" y "descuento" (Ilustración 5.6), pero puede ser útil tenerlo para evitar malos cálculos por los usuarios y facilitar las agregaciones.

f) **Completar las tablas de dimensiones**. Se considera:

- Repasar y añadir más atributos a las dimensiones que puedan ser útiles.
- Características directas, por ejemplo de Almacén: el código postal, playas de estacionamiento…
- Características derivadas, por ejemplo de Almacén: zona conflictiva, estacionamiento.

g) **Elegir la duración histórica de las bases de datos**. Se debe decir un rango de duración de los datos en las bases de datos, pues es necesario trabajar con datos que representen la realidad actual de la organización.

h) **Rastrear las dimensiones que cambian**. Se considera relevante el caso en que, en el mundo real, para un valor de una dimensión, cambia el valor de un atributo que es significativo para el análisis sin cambiar el valor de su clave.

   Existen tres estrategias para el tratamiento de los cambios en las dimensiones:

   - Tipo 1: realizar la modificación.
   - Tipo 2: crear un nuevo registro.
   - Tipo 3: crear un nuevo atributo.

   Por ejemplo: considerar un DW con una dimensión Cliente. En la tabla correspondiente, un registro representa la información sobre el cliente X, cuyo estado civil cambia el 7 de abril del 2003 de soltero a casado. El estado civil del cliente es usado con frecuencia en el análisis de la información. Una agregación del 2000 al 2004, ¿cuántas veces cuenta el cliente como soltero y cuántas como casado?

i) **Decidir las prioridades de las consultas**. Este paso ya se relaciona con el diseño físico. Una vez claro el diseño lógico, de cara al diseño físico, es necesario analizar ciertas cosas:

   - ¿Qué medidas en la tabla de hechos son más importantes?
   - ¿Qué medidas pueden ser más interesantes de agregar?

5.6 Algoritmo Apriori

Dentro de las técnicas de aprendizaje no supervisado en Data Mining, se encuentran las técnicas descriptivas, donde el conjunto de observaciones no tienen clases asociadas. El objetivo es detectar regularidades en los datos de cualquier tipo: agrupaciones, contornos, asociaciones, valores anómalos. Se incluyen 2 tipos de análisis:

a) **Análisis Exploratorio**: busca correlaciones, asociaciones y dependencias entre los datos.

b) **Segmentación**: persigue la identificación de grupos entre los datos.

Luego, el algoritmo Apriori es considerado de análisis exploratorio, ya que genera reglas de asociación. Propuesto por Rakesh Agrawal y Ramakrishnan Srikant, del Centro de Investigación Almaden de IBM en el artículo "Fast Algorithms for Mining Association Rules" [Agrawal y Srikant, 1994]

5.6.1 *Reglas de asociación*

Las reglas de asociación corresponden a dependencias funcionales, requiere de atributos discretos y corresponde a una asociación unidireccional. Se buscan dependencias de la siguiente forma:

IF *Antecedente* THEN *Consecuente*

Por ejemplo:

IF (X1 = a, X3 = c, X5 = d) THEN (X4 = b, X2 = a)

Sean *n* los casos a evaluar. Se llamará *ra* a los casos en que el antecedente es cierto y *rc* los casos en que se cumple también el consecuente. Con esto, se definen dos parámetros:

a) Confianza (Tc): certeza de la regla. P(*Consecuente*|*Antecedente*)
b) Soporte (Ts): mínimo número de casos, o porcentaje en los que se aplica satisfactoriamente

Donde:

T*c* = *rc/ra*

T*s* = *rc* (*rc/n* en caso de porcentaje)

Para ilustrar estos conceptos, se utilizará el siguiente ejemplo: Se requiere analizar las compras de lápices, cuando se compran cuadernos. Una regla de asociación encontrada por el algoritmo, usando soporte 10% y Confienza 60%, tendría la siguiente forma:

IF Cuaderno THEN Lápiz (Soporte 10%, Confianza 80%)

Un 10% de soporte indica que en un 10% de las compras éstas incluyen un cuaderno. Y una confianza de 80%, indica que en un 80% de las compras que incluyen un cuaderno, se compra además, un lápiz.

Si se aumenta el soporte, por ejemplo a 100%, probablemente no se encontrarán instancias, debido a que es difícil que en todas las transacciones se hayan comprado cuadernos. Por otra parte, si se aumenta la confianza, disminuye la cantidad de reglas encontradas, ya que serían eventualmente menores con casos en que coincida la compra de lápices y cuadernos al la vez.

5.6.2 *Fases del algoritmo*

En su mayoría, los algoritmos de búsqueda de asociaciones y dependencias se basan en descomponer el problema en dos fases:

a) Búsqueda de conjuntos de atributos (itemsets), que tienen un soporte por encima del mínimo soporte dado (T*s*). El soporte para un conjunto de atributos es el número de transacciones que contienen a este conjunto de atributos. Los conjuntos de atributos que tienen un soporte por encima del umbral se denominan *Large Itemsets* (conjuntos de atributos grandes) y los demás *Small Itemsets* (conjuntos de atributos pequeños)

b) Utilizar los *Large Itemsets* para generar las reglas deseadas. Se realizan particiones binarias y disjuntas de los itemsets y se calcula la confianza de cada uno. Se retienen aquellas reglas que tienen confianza mayor o igual a la deseada.

Capítulo 6. Descripción de herramientas a utilizar

La versión de Callmanager a usar será la 4.x, montada por defecto sobre MS SQL Server 2000 en el servidor de Callmanager, y para la creación del Data Mart se utilizará el mismo motor.

Se ha escogido SQL Server porque es el motor de base de datos sobre el cual funciona la versión del Callmanager que se utilizará, así se evitarán los tiempos de eventuales migraciones a otro motor de base de datos, y se obtendrá compatibilidad con lo tipos de datos de la base de datos operativa y el Data Mart. Luego, para un análisis de Data Mining, se utilizará WEKA en su versión 3.4.11, debido a que se trata de una herramienta libre, y que tiene incorporado el algoritmo Apriori que se requiere.

6.1 MS SQL server 2000

MS SQL Server es un administrador de bases de datos relacionales propietario de Microsoft. Permite la gestión de múltiples bases de datos por medio de una interfaz de ventanas, además permite la incorporación de *Triggers y Procedimientos Almacenados* entre otras funcionalidades. Dentro de las características de Bussines Intelligence que posee SQL Server 2000, según Microsoft [Microsoft, 2007], se tiene:

6.1.1 *Analysis Services*

Analysis Services es un componente de los servicios OLAP (procesamiento analítico en línea) que posee SQL Server, ofrece funciones de OLAP y permite diseñar, crear y administrar estructuras multidimensionales que contienen datos agregados desde otros orígenes de datos, como bases de datos relacionales. Posee una interfaz intuitiva (Ilustración 6.1) que permite visualizar los cubos, dimensiones, etc. en una estructura de árbol, mientras que en su ventana de despliegue, se carga rápidamente información asociada a dichos elementos.

La generación de cubos multidimensionales es también bastante intuitiva, gracias a su interfaz de ventanas (Ilustración 6.2) que permite diseñar y procesar el cubo de manera rápida y eficiente. Cuando se procesa un cubo, se ofrece la posibilidad de escoger el modo de almacenamiento, pudiendo ser:

- **MOLAP**: almacena los datos y las agregaciones en una matriz de almacenamiento multidimensional optimizada, más que en una base de datos relacional.

- **ROLAP**: mantiene los datos en sus estructuras relacionales correspondientes, y almacena las tablas de agregaciones en bases de datos relacionales.

- **HOLAP**: mantiene los datos en sus estructuras relacionales correspondientes, y almacena las agregaciones en estructuras multidimensionales.

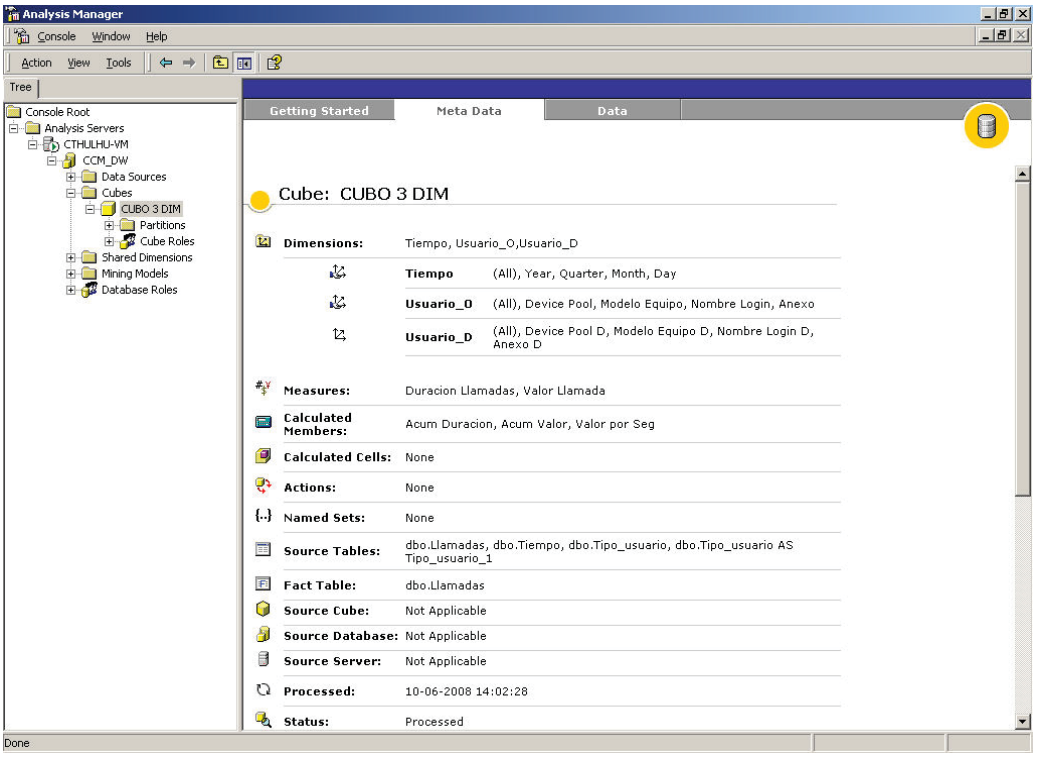


Ilustración 6.1 Interfaz Analysis Services

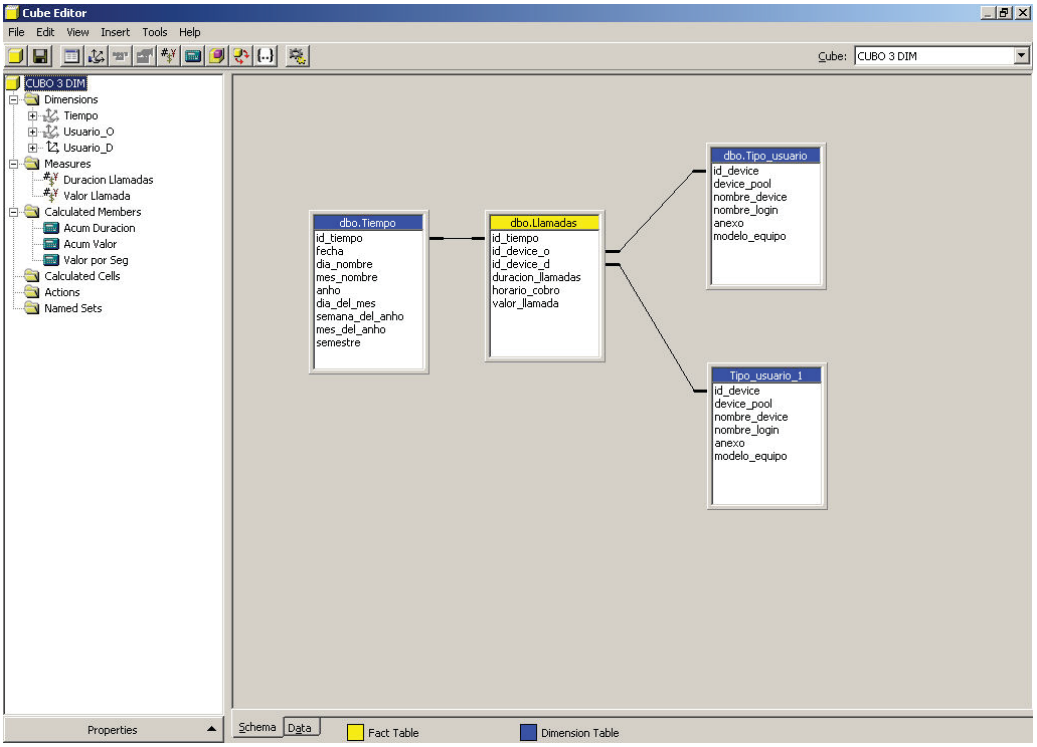


Ilustración 6.2 Generación de Cubos en Analysis Services

6.1.2 *Data transformation services*

Los Data Transformation Services (DTS) facilitan la importación, exportación y transformación de datos heterogéneos con OLE DB, Open Database Connectivity (ODBC), o archivos de texto. DTS también suprime la necesidad de intervención por parte del usuario ya que permite la importación y transformación de datos automáticamente y de forma regular.

6.2 WEKA

Se utilizará una herramienta disponible bajo licencia GNU llamada WEKA, acrónimo de Waikato Environment for Knowledge Analysis. Si bien existen otras herramientas como Knowledge Seeker, Clementine, Enterprise Miner., etc., se escogió WEKA por ser una herramienta bajo licencia GNU, que por ser implementada en Java es multiplataforma, y es definida como: *"una colección de algoritmos de aprendizaje de maquina (machine learning algorithms) para tareas de Data Mining", "contiene herramientas de pre-procesado, clasificación, clustering, regresiones, reglas de asociación, y visualización"* [Witten y Frank, 2005].

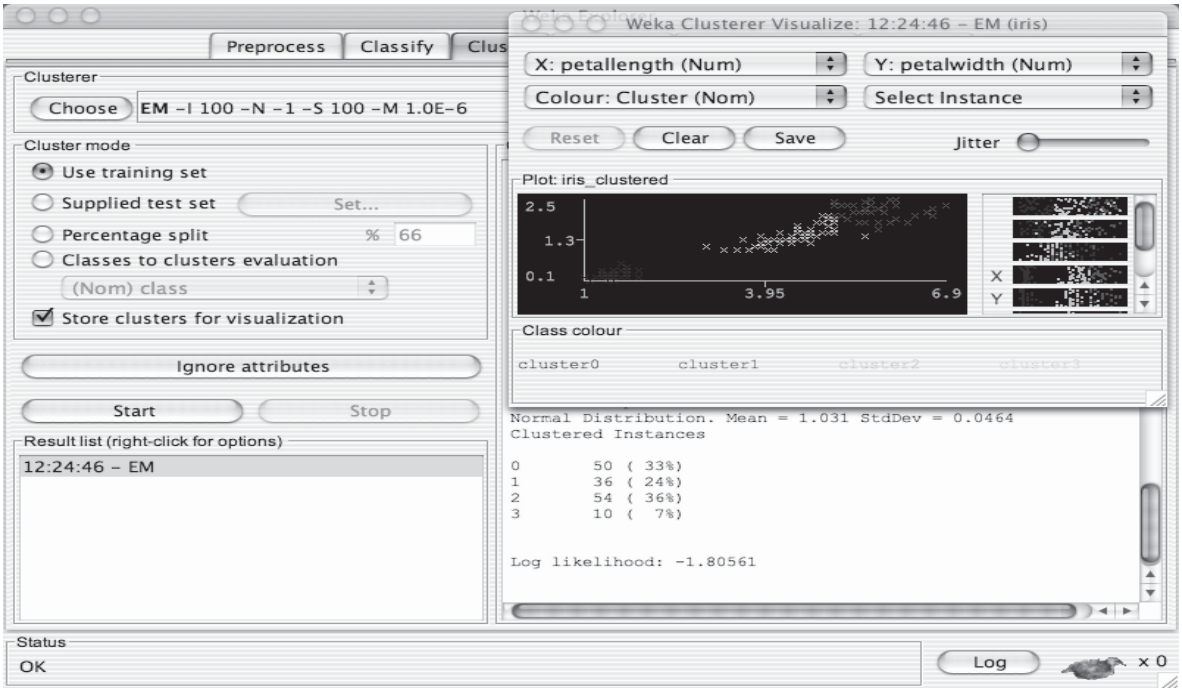


Ilustración 6.3 Interfaz gráfica de WEKA

Además, WEKA tiene una interfaz grafica para el usuario bastante amigable (Ilustración 6.3), y hace relativamente fácil su uso. Eso si, recomiendan conocer bien el negocio, y depurar los datos de manera que no arroje información distorsionada o sesgada. Weka define 4 entornos de trabajo [Hernández y Ferri, 2006]:

6.2.1 *Simple CLI*

Entorno consola para invocar directamente con java a los paquetes de WEKA. Es una abreviación de Simple Client. Esta interfaz proporciona una consola para poder introducir comandos. A pesar de ser en apariencia muy simple es *"extremadamente potente porque permite realizar cualquier operación soportada por WEKA de forma directa"* [García, 2006]; no obstante, es muy complicada de manejar ya que es necesario un conocimiento completo de la aplicación.

6.2.2 *Explorer*

Entorno visual que ofrece una interfaz gráfica para el uso de los paquetes. En Explorer existen 6 sub-entornos de ejecución [Hernández y Ferri, 2006]:

a) **Preprocess**: incluye las herramientas y filtros para cargar y manipular los datos.

b) **Classification**: acceso a las técnicas de clasificación y regresión.

c) **Cluster**: integra varios métodos de agrupamiento.

d) **Associate**: incluye unas pocas técnicas de reglas de asociación.

e) **Select Attributes**: permite aplicar diversas técnicas para la reducción del número de atributos.

f) **Visualize**: permite estudiar el comportamiento de los datos mediante técnicas de visualización.

6.2.3 *Experimenter*

Entorno centrado en la automatización de tareas de manera que se facilite la realización de experimentos a gran escala. *"El modo Experimenter es un muy útil para aplicar uno o varios métodos de clasificación sobre un gran conjunto de datos y, luego poder realizar contrastes estadísticos entre ellos y obtener otros índices estadísticos"* [García, 2006].

6.2.4 *Knowledge flow*

Permite generar proyectos de Data Mining mediante la generación de flujos de información. *"Esta última interface de WEKA es quizá la más cuidada y la que muestra de una forma más explícita el funcionamiento interno del programa. Su funcionamiento es gráfico y se basa en situar en el panel de trabajo, elementos base de manera que creemos un "circuito" que defina nuestro experimento"* [García, 2006]

Además, WEKA tiene incorporados elementos de visualización como árboles de decisión y redes neuronales entre otros (Ilustración 6.4, Ilustración 6.5), potenciando el análisis.

Hay que destacar que el proyecto involucra la creación de un Data Mart, sobre el cual se ejecutará el algoritmo Apriori, por lo que los datos ya estarán filtrados y ordenados.

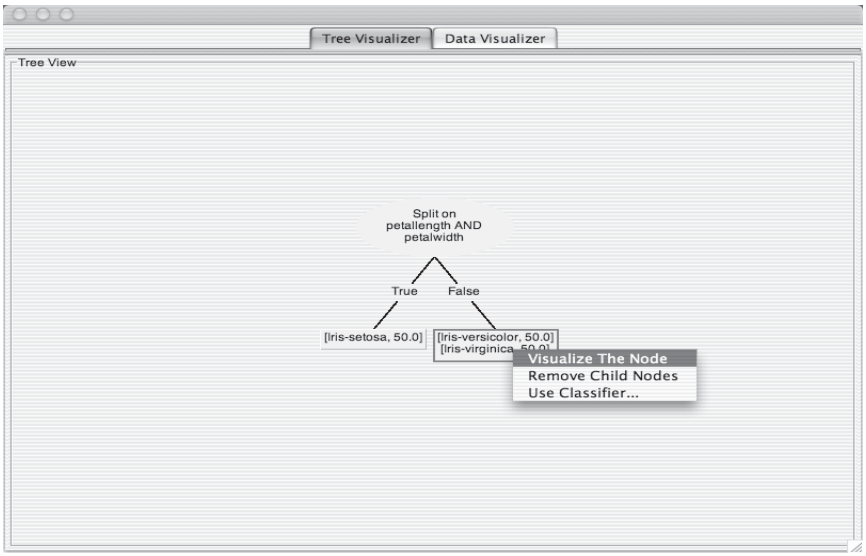


Ilustración 6.4 Ejemplo de árbol de decisión generado por WEKA

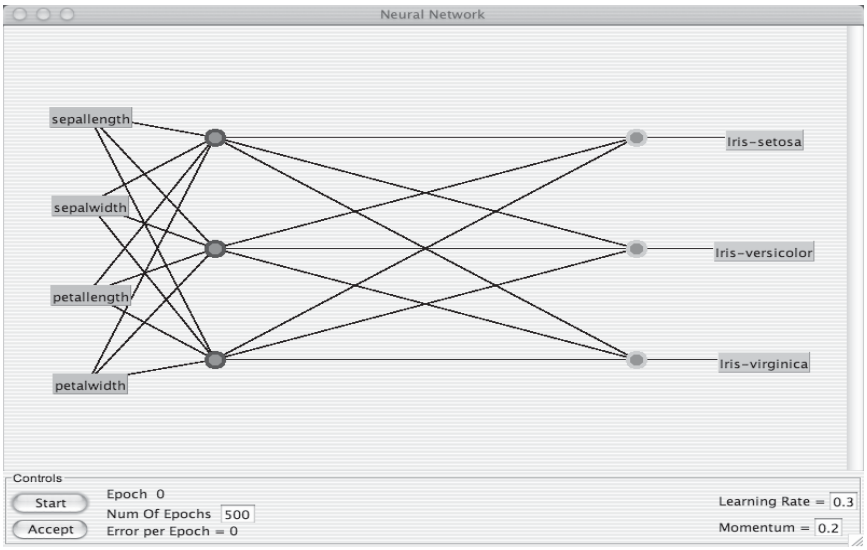
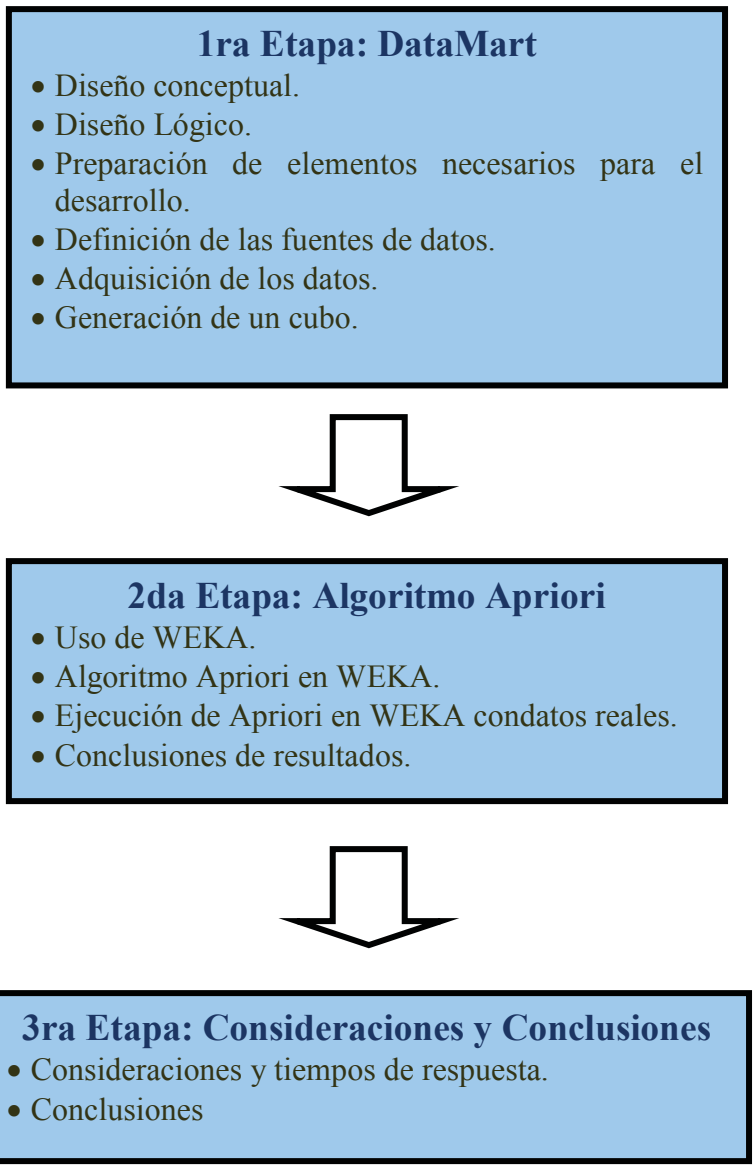


Ilustración 6.5 Ejemplo de red neuronal generado por WEKA

Capítulo 7. Desarrollo del proyecto

Con en análisis anterior respecto al entorno del proyecto, y a los estudios sobre Data Warehousing, Minería de Datos y Software a utilizar, se puede definir una planificación para la resolución del proyecto usando el siguiente esquema:



7.1 Data mart

7.1.1 *Diseño conceptual*

La recolección y análisis de requisitos tiene relación con el discernimiento de las fuentes que serán útiles del Callmanager, y con los requisitos del usuario, que darán el enfoque al análisis. Luego, son de interés para el proyecto aquellos que tienen relación con el tiempo, Calling Search Space, destino de las llamadas, etc.

El modelo debe ser capaz de responder preguntas tales como:

- ¿Cuáles fueron los volúmenes de llamadas (duración de éstas) por cluster, device pool (grupo o área), modelo de teléfono, anexo en el último período (años, mes, trimestre, día)?
- ¿Cuáles fueron los trimestres del año con mayor costo de llamadas, según device pool, respecto de otros trimestres del mismo año, o del mismo trimestre pero en años anteriores?
- Comparar mes actual con mes anterior, o mismo mes del año anterior, respecto de las llamadas salientes a celulares, anexos internos o llamadas nacionales.
- Tendencias de llamadas según departamentos de la organización.
- Cuáles fueron los precios por segundo, por trimestre.
- Llamadas con duración 0, o que no fueron exitosas.

Las dimensiones definidas son:

- Tiempo, con una granularidad de días, semanas, meses, semestre, trimestres y años.
- Tipo usuario, por ejemplo, cluster al que pertenece, device pool, modelo de equipo, etc.
- Tipo de destino, por ejemplo, cluster al que pertenece, device pool, modelo de equipo, etc.

7.1.2 *Diseño lógico*

Utilizando los 9 pasos para el diseño de modelos multidimensionales visto anteriormente se tiene:

a) **Elegir un "proceso" de la organización para modelar**. Para el proyecto, la actividad a modelar serían las llamadas en un sistema de telefonía IP que utiliza un Callmanager de Cisco para gestionar sus llamadas, grupos de usuarios, tarificación, destino de llamadas, etc

b) **Decidir el grano (nivel de detalle) de representación**. En el caso de los usuarios, el detalle incluirá el modelo del teléfono, ya que de éste depende la inclusión de nuevas funcionalidades, además que varían sus precios de adquisición.

   La tarificación se agrupará en periodos de cobro, tales como temporada alta y baja, los que tendrán un valor en pesos por cada mes/año en que se hayan realizado llamadas.

   La información respecto del tiempo, como se vio en la fase de diseño conceptual, contará con una granularidad de días, semanas, meses, semestre, trimestres y años.

   La tabla de hecho constará de elementos que permitan responder las consultas planteadas, lo que incluye datos previamente calculados.

c) **Identificar las dimensiones que caracterizan el proceso**. Ya definidas inicialmente en la etapa de diseño conceptual, las dimensiones son:

   - **Tiempo: dimensión temporal**. (días, semanas, meses, semestre, trimestres y años)
   - **Tipo usuario**: dimensión que identifica el gestor de la actividad, en nuestro caso, una llamada (cluster, device pool, nombre del usuario asociado al teléfono, número telefónico o anexo, modelo de teléfono).
   - **Tipo de destino**: dimensión que utiliza la misma tabla de usuarios.

d) **Decidir la información a almacenar sobre el proceso**. Los hechos de interés son:

   - Duración de cada llamada realizada.
   - Valor en pesos de cada llamada realizada, calculados mediante una tabla de tarificación.

e) **Almacenar precálculos en la tabla de hechos**. Es de interés para el análisis datos calculados como:

   - Segundos Acumulados desde un periodo anterior (trimestre, mes, día).
   - Pesos Acumulados desde un periodo anterior (trimestre, mes, día).

- Valor por segundo, calculado desde los valores totales de llamadas y las duraciones totales de estas.

f) **Completar las tablas de dimensiones**. De ser necesario, se analizarán eventuales cambios en las tablas de dimensiones.

g) **Elegir la duración histórica de las bases de datos**. Lo ideal, es que la base de datos date de los inicios de la implementación del sistema de telefonía IP, para evaluar su evolución.

h) **Rastrear las dimensiones que cambian**. Es necesario mantener observaciones sobre algunas dimensiones, para estar preparado en caso de cambios en ellas. La dimensión que más rápidamente cambia es la de usuario en los primeros tiempos de operación del Callmanager, ya que se agregan teléfonos constantemente, así como usuarios nuevos, pero sus atributos no cambian en el corto plazo.

   También puede cambiar la dimensión de horario de cobro, ya que eventualmente se puede incluir un nuevo intervalo de tiempo para diferenciar tarifas.

i) **Decidir las prioridades de las consultas**. Es probable que los campos relacionados con tarificación tengan mayor prioridad por parte del usuario final de Data Mart, ya que es una forma comprensible por la mayoría de los tomadores de decisiones de una organización de evaluar el sistema de telefonía, además, porque resulta interesante el análisis de los costos que genera el sistema.

Luego, el modelo quedaría como indica la Ilustración 7.1.

Así, con el resultado del diseño lógico, se obtiene el modelo del Data Mart, que permitirá responder las consultas planteadas y otras.

Luego, con lo obtenido, se utilizarán las herramientas de análisis de datos estudiadas anteriormente, éstas son las herramientas de Business Intelligence de SQL Server 2000, además de el algoritmo Apriori usando WEKA.

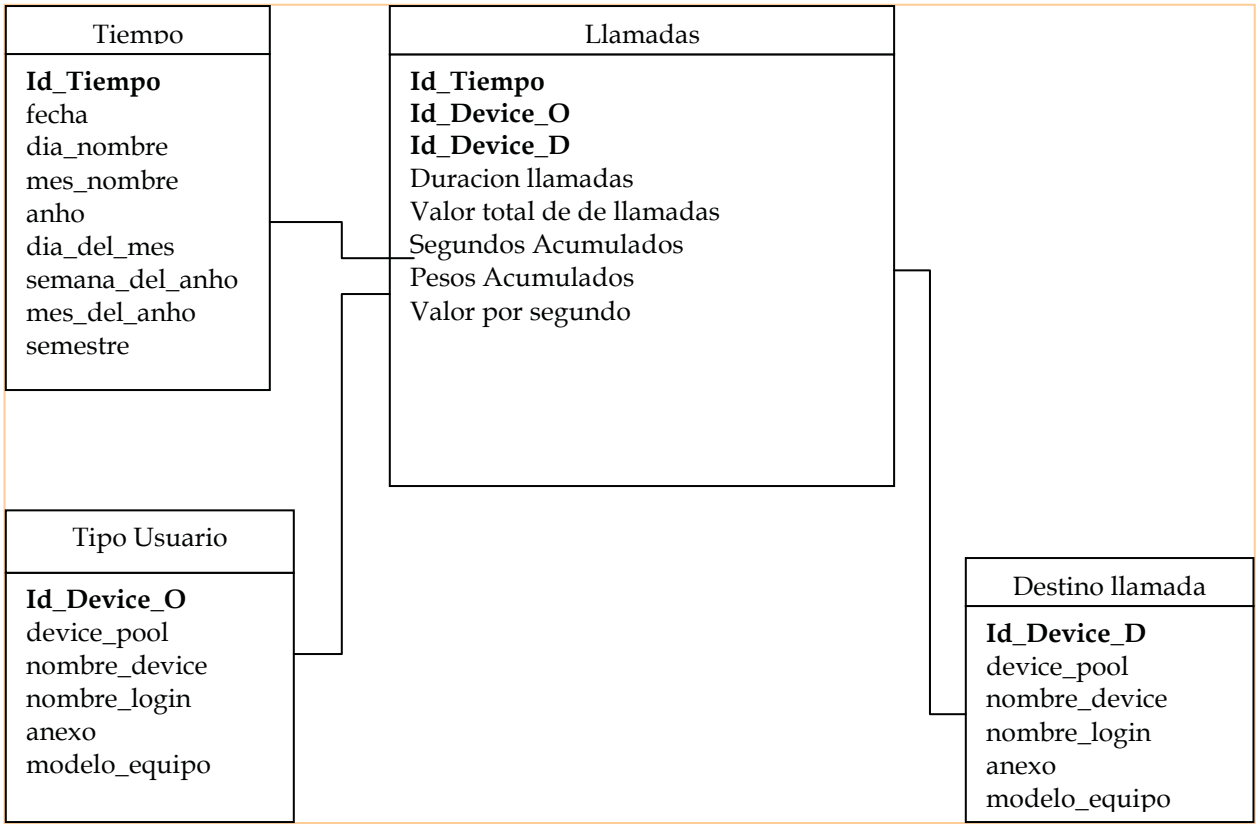


Ilustración 7.1 Modelo estrella del Data Mart

A continuación se especifica los procesos y/o pasos para el desarrollo del proyecto.

7.1.3 *Preparación de elementos necesarios para el desarrollo*

Antes de empezar a trabajar los datos es necesario crear en SQL Server la base de datos que contendrá la información transformada desde las distintas fuentes. Se llamará a esta base de datos CCM_DW, y debe ser consistente con el diseño lógico planteado en la Ilustración 7.1.

7.1.4 *Fuentes de datos*

Las fuentes de datos corresponden a las bases de datos:

a) **CCM0300**. De tipo operacional. Utilizada para obtener información de las dimensiones: "Tipo de usuario" y "Destino de llamadas".

b) **CDR**. Utilizada a modo de "log" por el Callmanager. Y usada en el proyecto para obtener información para la tabla de hecho, así como para la dimensión temporal.

Para probar el modelo, se cargará la base de datos CCM_DW mediante el proceso de adquisición de datos descrito en el punto 7.1.5, utilizando una base de datos de Callmanager real con las características de la Tabla 7.1.

Característica	Descripción
Intervalo de tiempo de registros de CDR	Desde 28 de Septiembre de 2006 a 22 de Octubre de 2007
Cantidad de registros de CDR	1.479.282 registros
Espacio en disco de CCM	25,3 Mb (9 Mb archivo Log)
Espacio en disco CDR	1,37 Gb (739 Mb archivo Log)

Tabla 7.1 Características de datos reales

Validado con el *Diccionario de Datos de Callmanager v4.1(3)*, descargable desde el sitio de Cisco con cuenta gratuita [Cisco, 2006].

7.1.5 *Adquisición de datos*

Para el proceso de extracción, transformación y carga de datos se utilizó la herramienta "Data Transformation Services" incorporada en SQL Server.

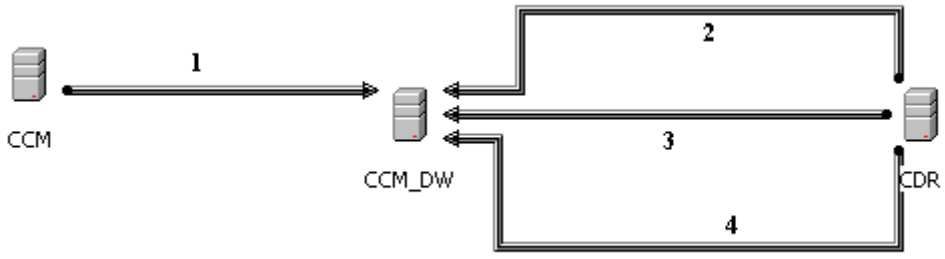


Ilustración 7.2 Data Transformation Services

Como muestra la Ilustración 7.2, tenemos 3 conexiones:

a) **CCM**: base de datos CCM0300
b) **CDR**: base de datos CDR
c) **CCM_DW**: base de datos creada para almacenar las consultas

Y se crearon también 4 TDT o *Transform Data Task* (tarea de transformación de datos). Luego, para obtener la información de los usuarios se utiliza el TDT 1 (Ilustración 7.2), el cual realiza la consulta de la Tabla 7.2 a CCM:

```
SELECT Device.Name,
Device.LoginUserid,
NumPlan.DNOrPattern,
DevicePool.Name AS DevicePool,
TypeModel.Name AS DeviceModel
FROM Device INNER JOIN
        DeviceNumPlanMap ON Device.pkid = DeviceNumPlanMap.fkDevice INNER JOIN
        NumPlan ON NumPlan.pkid = DeviceNumPlanMap.fkNumPlan INNER JOIN
        DevicePool ON Device.fkDevicePool = DevicePool.pkid INNER JOIN
        TypeModel ON Device.tkModel = TypeModel.Enum
```

Tabla 7.2 Consulta para obtener usuarios

La Ilustración 7.3 muestra las relaciones entre las tablas utilizadas en la consulta.

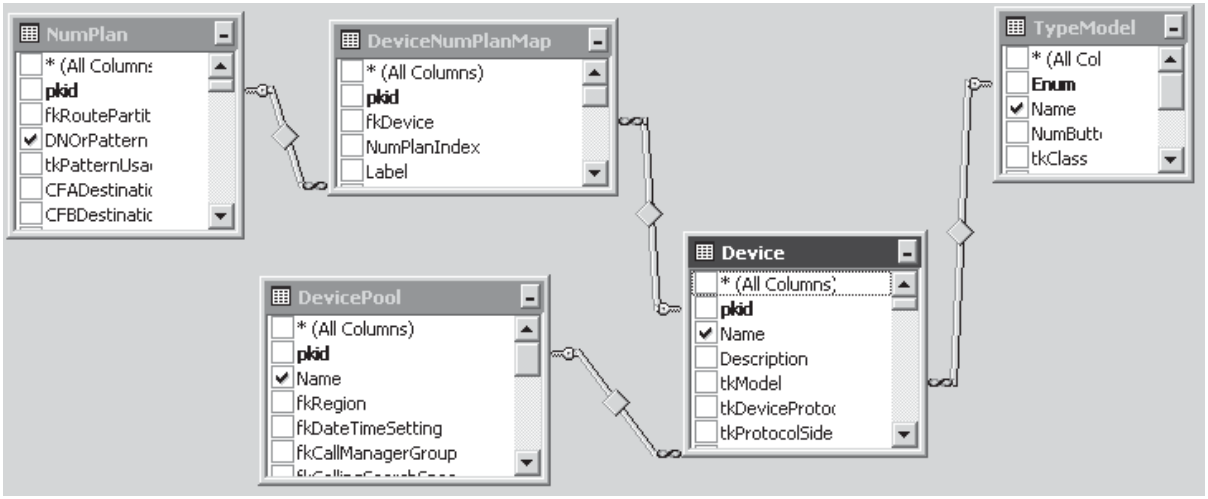


Ilustración 7.3 Relaciones entre las tablas

La información obtenida con TDT 1 se muestra en la figura Ilustración 7.4.

	Column Name	Data Type	Length	Allow Nulls
▶🔑	id_device	uniqueidentifie	16	
	device_pool	varchar	50	✓
	nombre_device	varchar	129	✓
	nombre_login	varchar	250	✓
	anexo	varchar	50	✓
	modelo_equipo	varchar	50	✓

Ilustración 7.4 Tabla Tipo_Usuario

La TDT 2 realiza la consulta de la Tabla 7.3 para obtener los datos de las llamadas.

```
SELECT
CONVERT(varchar, DATEADD(s, dateTimeOrigination, 'Dec 31, 1969 19:00:00'), 101) AS Fecha,
duration AS duracion_llamadas,
origDeviceName,
 destDeviceName,
{ fn HOUR(CONVERT(datetime, DATEADD(s, dateTimeOrigination, 'Dec 31, 1969 19:00:00'), 101))
} AS hora
FROM       CallDetailRecord
```

Tabla 7.3 Consulta para obtener datos de llamadas

Es importante notar que se requiere una conversión de la fecha, esto es porque el Callmanager utiliza el formato Epoch.

La Ilustración 7.5 muestra las transformaciones que son necesarias antes de cargar los datos a la tabla Llamadas en CCM_DW.

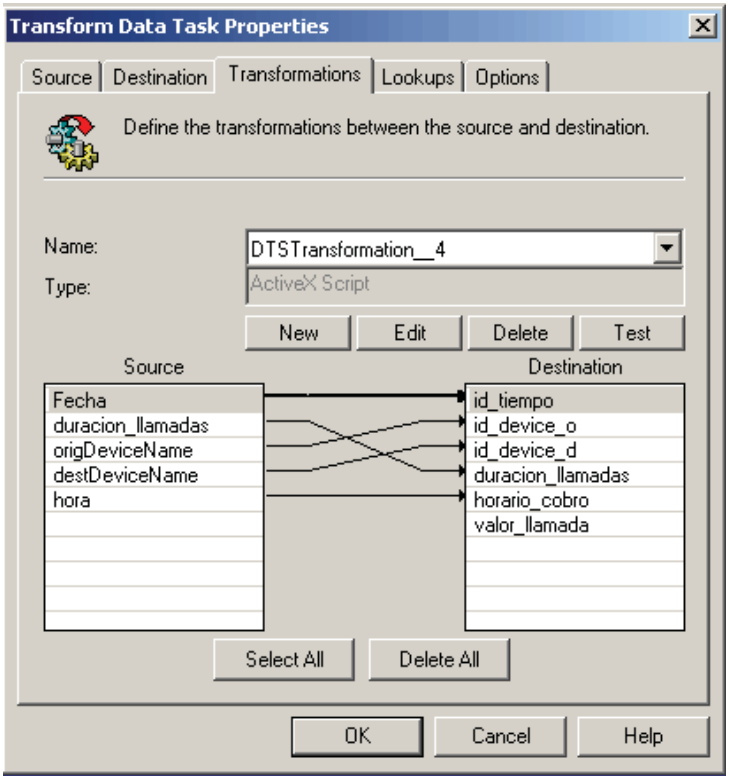


Ilustración 7.5 Transformaciones en tabla Llamadas

La Ilustración 7.5 muestra que el campo valor_llamada no es ingresado, ya que ese valor depende de la tabla de tarificación, la que a su vez, se construye en una transformación que se verá más adelante. Para obtener los datos del usuario (origen o destino) y hacerlos coincidir con la tabla Tipo_Usuario de CCM_DW se debe obtener el ID del usuario que posee en la tabla Tipo_Usuario (que ya ha sido cargada). Para realizar esto, se utilizan los llamados Lookups de Data Transformation Services, que consisten en una

consulta configurable (Tabla 7.4). Esto se realiza análogamente para obtener el ID en la tabla Tiempo.

SELECT id_device FROM Tipo_usuario WHERE (nombre_device = ?)	SELECT id_tiempo FROM Tiempo WHERE (fecha = ?)

Tabla 7.4 Lookup usuario y Lookup tiempo

Luego, éstos *Lookups* son utilizados en los scripts de transformación mediante las funciones descritas en la Tabla 7.5 y en la Tabla 7.6.

```
Function Main()
        DTSDestination("id_tiempo") = DTSLookups("id_tiempo_lookup").Execute(DTSSource("fecha"))
        Main = DTSTransformStat_OK
End Function
```

Tabla 7.5 Script obtiene id Tiempo

```
Function Main()
        DTSDestination("id_device_o")=
DTSLookups("id_device_lookup").Execute(DTSSource("origDeviceName"))
        Main = DTSTransformStat_OK
End Function
```

Tabla 7.6 Script obtiene id Usuario

Para obtener el horario de cobro de una llamada, se debe evaluar la hora en que se realizó, para ello utilizamos el valor *hora* entregado por nuestro query (Tabla 7.3). Luego, mediante el script de transformación que muestra la Tabla 7.7, se asignará *1* a las llamadas realizadas después de las 8pm y antes de las 8 am, de modo de identificarlas posteriormente como aquellas llamadas realizadas en horario alto. Análogamente, se asigna *0* a las llamadas realizadas entre las 8am y las 8pm, siendo estas las efectuadas en un horario bajo.

```
Function Main()
        If DTSSource("hora") > 8 AND DTSSource("hora") < 20 Then
                DTSDestination("horario_cobro") = 0  'Bajo
        Else
                DTSDestination("horario_cobro") = 1  'Alto
        End If
        Main = DTSTransformStat_OK
End Function
```

Tabla 7.7 Script asigna horarios de cobro

Con la TDT 3 se cargará o poblará la tabla de tarificación. Lo que hace es obtener y agrupar todos los *Año/Mes* del CDR mediante la consulta mostrada en la Tabla 7.8, y copiar cada año y su correspondiente mes a la tabla *Horario_de_Cobro* de CCM_DW, para luego, y de forma manual[2] asignar los valores según horario alto/bajo de cobro. Después de

[2] Eventualmente se puede contar con una tabla de tarificación previa, pero para el desarrollo de este proyecto, se usó este método.

asignar éstos valores, la tabla de tarificación *Horario_de_Cobro* queda con el formato que muestra la Ilustración 7.6.

```
SELECT     YEAR(Fecha) AS anho, MONTH(Fecha) AS mes
FROM
     (SELECT     CONVERT(varchar, DATEADD(s, dateTimeOrigination,
'Dec 31, 1969 19:00:00'), 101) AS Fecha


FROM         CallDetailRecord) DERIVEDTBL
GROUP BY YEAR(Fecha),
MONTH(Fecha)
ORDER BY YEAR(Fecha), MONTH(Fecha)
```

Tabla 7.8 Consulta para obtener fechas de cobro

	anho	mes	bajo	alto
▶	2006	9	1	2
	2007	2	1	2
	2007	5	1	2
	2007	6	1	2
	2007	4	1	2
	2006	10	1	2
	2007	9	1	2
	2006	12	1	2
	2007	10	1	2
	2006	11	1	2
	2007	8	1	2
	2007	7	1	2
	2007	1	1	2
	2007	3	1	2
*				

Ilustración 7.6 Tabla Horario_de_Cobro

Donde los valores de alto/bajo corresponden a los factores por los cuales hay que multiplicar la duración de una llamada (en segundos) para obtener el valor de la llamada. Esta tarea se realiza después de la carga de CCM_DW, es decir, después de ejecutarse todos los TDT.

La TDT 4, mediante la consulta de la Tabla 7.9, se obtienen las fechas de las llamadas, se separan en fecha (mm/dd/aaaa), nombre del día, nombre del mes, año, día del mes, semana del año y mes del año. Con esta tarea completa, se poblará la tabla *Tiempo*.

```
SELECT Fecha, DATENAME(dw, Fecha) AS dia_nombre, DATENAME(mm, Fecha) AS mes_nombre,
YEAR(Fecha) AS anho, DATENAME(dd, Fecha) AS dia_del_mes, DATENAME(wk, Fecha) AS
semana_del_anho, MONTH(Fecha) AS mes_del_anho
FROM
  (SELECT CONVERT(varchar, DATEADD(s, dateTimeOrigination, 'Dec 31, 1969 19:00:00'), 101) AS
Fecha FROM CallDetailRecord) DERIVEDTBL
GROUP BY Fecha
```

Tabla 7.9 Consulta para obtener fechas de llamadas

Adicionalmente, se agregan algunas funciones de transformación, como por ejemplo, para la obtención del semestre se utilizará un pequeño script, el cual, calcula el valor del semestre (1 o 2) según el mes actual. El resto de los valores se traspasan tal cual a la tabla Tiempo (Ilustración 7.7).

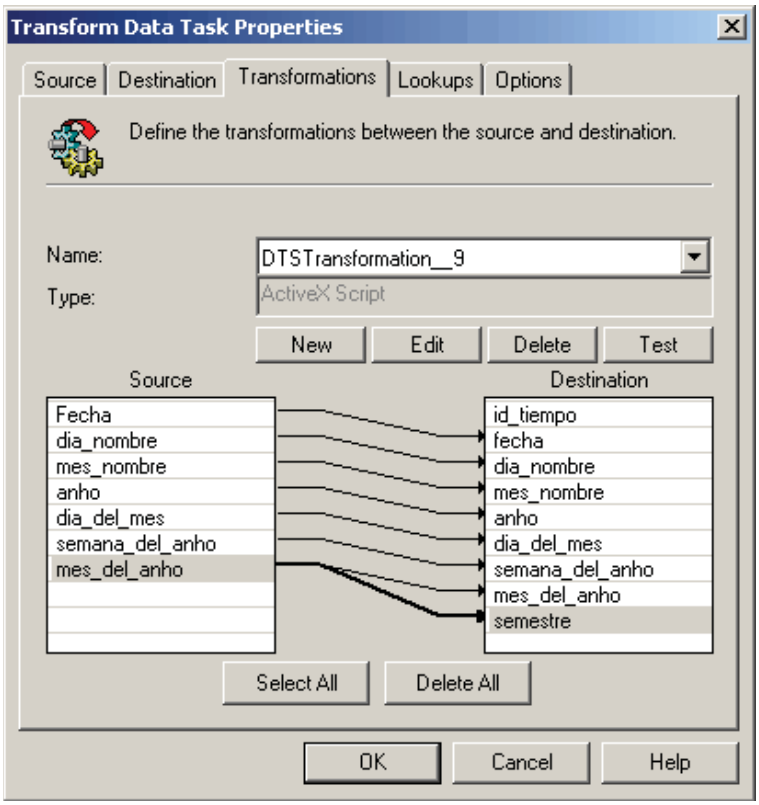


Ilustración 7.7 Transformaciones en tabla Tiempo

Finalmente, y para completar el proceso de extracción, transformación y carga de los datos, se debe asociar a cada fecha de la tabla de tarificación Horario_de_Cobro, un valor que represente el costo de 1 segundo en horario alto/bajo de cada mes, y luego se ejecuta el query de la Tabla 7.10.

Con ello se consigue calcular los valores (en pesos) de cada llamada según la tabla de tarificación.

```
UPDATE LLAMADAS
SET
Llamadas.valor_llamada = CASE
        WHEN AL1.horario_cobro = '1' THEN
                AL1.valor_llamada = AL1.duracion_llamadas*AL3.alto
        ELSE
                AL1.valor_llamada = AL1.duracion_llamadas*AL3.bajo
        END
FROM
   LLAMADAS AS AL1,
   TIEMPO AS AL2,
   HORARIO_DE_COBRO AS AL3
WHERE
   AL1.ID_TIEMPO = AL2.ID_TIEMPO AND
   AL2.ANHO = AL3.ANHO AND
   AL2.MES_DEL_ANHO = AL3.MES;
```

Tabla 7.10 Consulta de cálculo y actualización de valores de llamadas

7.1.6 *Generación de un cubo en Analysis Services*

Utilizando las tablas de CCM_DW se ha creado un cubo que llamado *CUBO 3 DIM*, y hace referencia a las siguientes tablas de CCM_DW:

- Llamadas : como tabla de hecho
- Tiempo: como dimensión temporal
- Tipo_Usuario: en 2 instancias, como dimensión de usuario originador de llamadas y otra como usuario receptor de llamadas.

Se incorporaron las tablas en la herramienta *Analysis Services* de SQL Server, como el mostrado en la Ilustración 7.8.

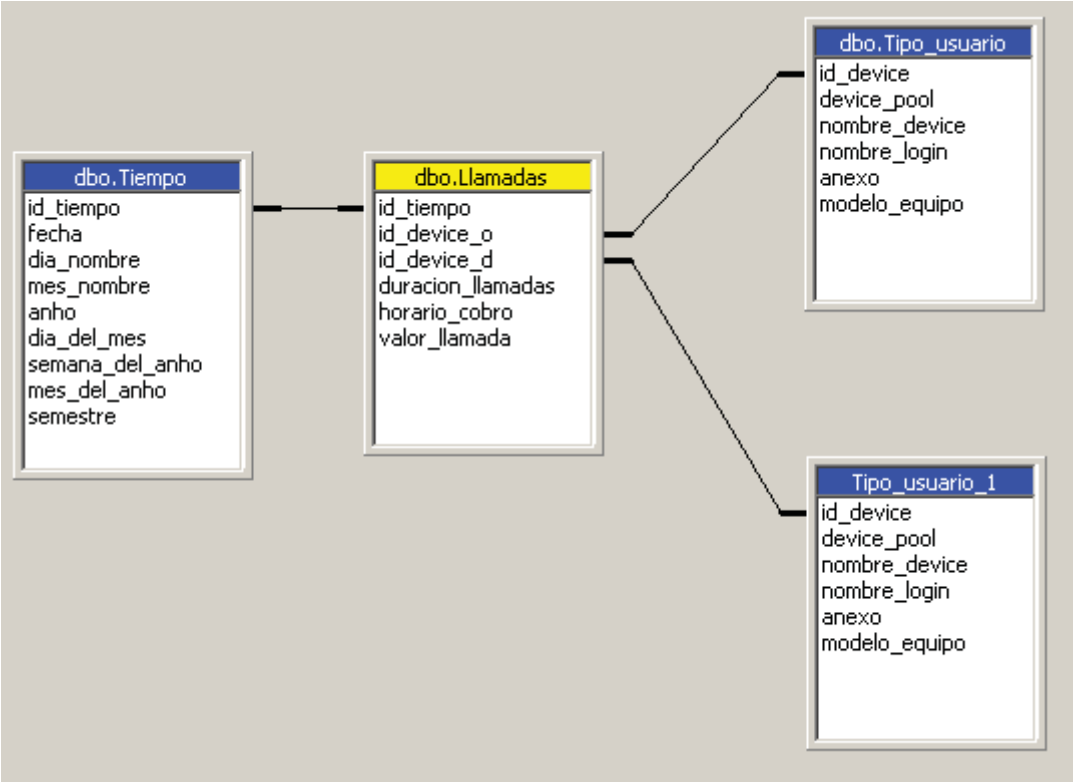


Ilustración 7.8 Tablas de cubo de 3 dimensiones

Las métricas del modelo son:

a) **duracion_ llamadas**: valor en segundos de cada llamada.
b) **valor_llamada**: valor en pesos de cada llamada, calculado previamente con una tabla de tarificación.

Luego, se crean algunos miembros calculados:

a) **Acum Duracion**: va acumulando los segundos partiendo desde *Year*. Utiliza la siguiente fórmula:

SUM(YTD(), [Measures].[Duracion Llamadas])

b) **Acum Valor**, acumula el valor de las llamadas partiendo desde *Year*. Utiliza la siguiente fórmula:

SUM(YTD(), [Measures].[Valor Llamada])

c) **Valor por Seg**, divide el valor de una llamada por la duración de esta. Utiliza la siguiente fórmula:

[Measures].[Valor Llamada]/[Measures].[Duracion Llamadas]

En la barra de la izquierda, el editor de cubos de SQL Analysis Services muestra las dimensiones, métricas, y miembros calculados (Ilustración 7.9).

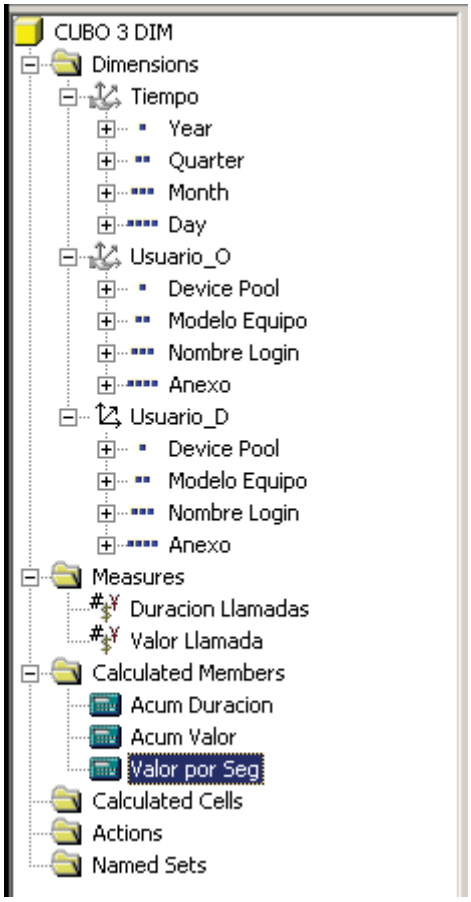


Ilustración 7.9 Árbol de propiedades de Cubo

La Ilustración 7.10 muestra el despliegue de la dimensión temporal, y los valores que toma de acuerdo a las métricas y miembros calculados definidos.

		MeasuresLevel				
- Year	+ Quarter	Duracion Llamadas	Valor Llamada	Acum Valor	Acum Duracion	Valor por Seg
All Tiempo	All Tiempo Total	120.918.146,00	101.429.262,10			0,84
- 2006	2006 Total	28.836.802,00	32.021.794,80	32.021.794,80	28.836.802,00	1,11
	+ Quarter 3	725.897,00	637.119,40	637.119,40	725.897,00	0,88
	+ Quarter 4	28.110.905,00	31.384.675,40	32.021.794,80	28.836.802,00	1,12
- 2007	2007 Total	92.081.344,00	69.407.467,30	69.407.467,30	92.081.344,00	0,75
	+ Quarter 1	26.592.379,00	15.027.845,00	15.027.845,00	26.592.379,00	0,57
	+ Quarter 2	28.611.308,00	19.614.270,60	34.642.115,60	55.203.687,00	0,69
	+ Quarter 3	29.512.907,00	26.548.885,70	61.191.001,30	84.716.594,00	0,90
	+ Quarter 4	7.364.750,00	8.216.466,00	69.407.467,30	92.081.344,00	1,12

Ilustración 7.10 Despliegue de dimensión Tiempo

La Ilustración 7.11 muestra los datos obtenidos cuando se despliega solo la dimensión de Tipo Usuario, la cual, contiene los usuarios que han originado llamados. Nótese que los miembros calculados que requieren de la dimensión temporal no muestran información alguna, ya que no cuentan con la dimensión requieren para hacer los cálculos.

		MeasuresLevel				
- Device Pool	+ Modelo Equipo	Duracion Llamadas	Valor Llamada	Acum Valor	Acum Duracion	Valor por Seg
All Usuario_O	All Usuario_O Total	120.918.146,00	101.429.262,10			0,84
- Kennedy	Kennedy Total	106.417.023,00	89.076.742,10			0,84
	+ Cisco 7912	38.297.997,00	31.775.273,80			0,83
	+ Cisco 7920	338,00	263,50			0,78
	+ Cisco 7940	7.773.649,00	6.525.280,60			0,84
	+ Cisco 7960	2.814.215,00	2.265.079,40			0,80
	+ Cisco ATA	900.463,00	766.493,30			0,85
	+ Cisco IP C	2.692.750,00	2.142.553,70			0,80
	+ CTI Port	1.632,00	1.420,90			0,87
	+ CTI Route					
	+ H.323 Gate	50.480.251,00	42.440.753,70			0,84
	+ Trunk	3.455.728,00	3.159.623,20			0,91
+ MAUSRST	MAUSRST Total	14.498.324,00	12.350.387,70			0,85
+ Unity Port	Unity Port Total	2.799,00	2.132,30			0,76

Ilustración 7.11 Despliegue de dimensión Tipo_Usuario

Muy similar son los datos mostrados en la Ilustración 7.12, ya que esta vez se despliega la información de llamadas de los receptores de llamadas, es decir, la dimensión Destino Llamada.

		MeasuresLevel				
- Device Pool	+ Modelo Equipo	Duracion Llamadas	Valor Llamada	Acum Valor	Acum Duracion	Valor por Seg
All Device Pool	All Device Pool Total	120.918.146,00	101.429.262,10			0,84
- Kennedy	Kennedy Total	108.448.268,00	90.944.811,00			0,84
	+ Cisco 7912	25.273.626,00	20.914.682,10			0,83
	+ Cisco 7920	8,00	7,00			0,88
	+ Cisco 7940	5.964.718,00	5.095.862,50			0,85
	+ Cisco 7960	2.172.038,00	1.756.338,70			0,81
	+ Cisco ATA	271.177,00	222.682,20			0,82
	+ Cisco IP C	936.467,00	745.938,80			0,80
	+ CTI Port	618.120,00	517.253,30			0,84
	+ CTI Route	0,00	0,00			-1,#J
	+ H.323 Gate	72.904.769,00	61.394.887,90			0,84
	+ Trunk	307.345,00	297.158,50			0,97
+ MAUSRST	MAUSRST Total	10.319.208,00	8.658.564,20			0,84
+ Unity Port	Unity Port Total	2.150.670,00	1.825.886,90			0,85

Ilustración 7.12 Despliegue de dimensión Destino_Llamada

Ahora, la Ilustración 7.13 muestra las dimensiones Tiempo y Tipo Usuario, en donde se puede obtener información de mayo nivel, y agregar o desagregar las dimensiones fácilmente mediante Analysis Services.

		MeasuresLevel				
+ Year	+ Device Pool	Duracion Llamadas	Valor Llamada	Acum Valor	Acum Duracion	Valor por Seg
All Tiempo	All Device Pool	120.918.146,00	101.429.262,10			0,84
	+ Kennedy	108.448.268,00	90.944.811,00			0,84
	+ MAUSRST	10.319.208,00	8.658.564,20			0,84
	+ Unity Port	2.150.670,00	1.825.886,90			0,85
+ 2006	All Device Pool	28.836.802,00	32.021.794,80	32.021.794,80	28.836.802,00	1,11
	+ Kennedy	25.909.111,00	28.736.963,70	28.736.963,70	25.909.111,00	1,11
	+ MAUSRST	2.355.989,00	2.643.855,30	2.643.855,30	2.355.989,00	1,12
	+ Unity Port	571.702,00	640.975,80	640.975,80	571.702,00	1,12
+ 2007	All Device Pool	92.081.344,00	69.407.467,30	69.407.467,30	92.081.344,00	0,75
	+ Kennedy	82.539.157,00	62.207.847,30	62.207.847,30	82.539.157,00	0,75
	+ MAUSRST	7.963.219,00	6.014.708,90	6.014.708,90	7.963.219,00	0,76
	+ Unity Port	1.578.968,00	1.184.911,10	1.184.911,10	1.578.968,00	0,75

Ilustración 7.13 Despliegue de dimensión Tiempo y Tipo Usuario

También es posible desplegar la información de la tabla de hechos utilizando las 3 dimensiones definidas, tal y como lo muestra la Ilustración 7.14.

			MeasuresLevel				
+ Year	+ Device Pool D	+ Device Pool	Duracion Llamadas	Valor Llamada	Acum Duracion	Acum Valor	Va
+ 2006	All Device Pool	+ Unity Port	15.209,51	15.209,51	15.209,51	15.209,51	
	+ Kennedy	All Usuario_O	361.225,77	361.225,77	361.225,77	361.225,77	
		+ Kennedy	334.794,61	334.794,61	334.794,61	334.794,61	
		+ MAUSRST	12.334,54	12.334,54	12.334,54	12.334,54	
		+ Unity Port	14.096,62	14.096,62	14.096,62	14.096,62	
	+ MAUSRST	All Usuario_O	13.308,32	13.308,32	13.308,32	13.308,32	
		+ Kennedy	12.334,54	12.334,54	12.334,54	12.334,54	
		+ MAUSRST	454,43	454,43	454,43	454,43	
		+ Unity Port	519,35	519,35	519,35	519,35	
	+ Unity Port	All Usuario_O	15.209,51	15.209,51	15.209,51	15.209,51	
		+ Kennedy	14.096,62	14.096,62	14.096,62	14.096,62	
		+ MAUSRST	519,35	519,35	519,35	519,35	
		+ Unity Port	593,54	593,54	593,54	593,54	
+ 2007	All Device Pool	All Usuario_O	1.210.256,41	1.210.256,41	1.210.256,41	1.210.256,41	
		+ Kennedy	1.121.701,06	1.121.701,06	1.121.701,06	1.121.701,06	
		+ MAUSRST	41.325,83	41.325,83	41.325,83	41.325,83	
		+ Unity Port	47.229,52	47.229,52	47.229,52	47.229,52	
	+ Kennedy	All Usuario_O	1.121.701,06	1.121.701,06	1.121.701,06	1.121.701,06	
		+ Kennedy	1.039.625,38	1.039.625,38	1.039.625,38	1.039.625,38	
		+ MAUSRST	38.301,99	38.301,99	38.301,99	38.301,99	
		+ Unity Port	43.773,70	43.773,70	43.773,70	43.773,70	
	+ MAUSRST	All Usuario_O	41.325,83	41.325,83	41.325,83	41.325,83	
		+ Kennedy	38.301,99	38.301,99	38.301,99	38.301,99	
		+ MAUSRST	1.411,13	1.411,13	1.411,13	1.411,13	
		+ Unity Port	1.612,72	1.612,72	1.612,72	1.612,72	
	+ Unity Port	All Usuario_O	47.229,52	47.229,52	47.229,52	47.229,52	
		+ Kennedy	43.773,70	43.773,70	43.773,70	43.773,70	
		+ MAUSRST	1.612,72	1.612,72	1.612,72	1.612,72	
		+ Unity Port	1.843,10	1.843,10	1.843,10	1.843,10	

Ilustración 7.14 Despliegue de dimensiones Tiempo, Destino_Llamadas y Tipo_Usuario

Con la creación de este cubo se valida el modelo de Data Mart planteado en la etapa de diseño, ya que las dimensiones agregan los datos correspondientes en nuestra tabla de hechos y la tabla de hechos entrega la información sugerida en la etapa de diseño.

7.2 Algoritmo de minería de datos

En esta etapa del proyecto se conecta la herramienta WEKA a la base de datos CCM_DW para utilizar sus algoritmos en busca de un análisis dirigido por los datos.

7.2.1 *Uso de WEKA*

WEKA es una herramienta compleja de configurar, y posee una documentación escasa y poco accesible. Esta parte del proyecto fue guiada en gran parte por el libro *"Data Mining, Practical Machine Learning Tools and Techniques"* [Witten y Frank, 2005]. Además, afortunadamente las últimas versiones de la aplicación poseen varias correcciones, y sobre todo, métodos para solucionar algunos problemas de manera rápida y efectiva. Para el presente proyecto se utilizará la versión 3.4.11 sobre MS-Windows 2000 SP 4. El hardware se describe en la Tabla 7.11.

Procesador	Intel Core2Duo, 3.0 Ghz
Placa Madre	Asus P5K-E
Memoria	Corsair 2 Gb
HHDD	Sata 160 Gb

Tabla 7.11 Hardware para pruebas

Lo primero que se debe realizar después de descargar e instalar la aplicación WEKA, es conectarla a la base de datos. Existen varios métodos, en este proyecto se utilizó ODBC, siguiendo los siguientes pasos:

1. Crear una nueva conexión ODBC a CCM_DW[3].
2. Abrir el archivo *weka.jar*, en la carpeta *weka/experiment* copiar el archivo *DatabaseUtils.props.odbc* dentro de la carpeta de instalación de WEKA.
3. Abrir el archivo DatabaseUtils.props.odbc con un editor de texto (pe. Notepad) y editar la línea: *jdbcURL=jdbc:odbc:<Nombre de conexión ODBC>*
4. Renombrar el archivo *DatabaseUtils.props.odbc* por *DatabaseUtils.props*.

Ahora, hay que considerar que WEKA está escrito en Java y que utiliza JVM, lo cual, puede eventualmente provocar problemas de falta de memoria. Para evitar (o solucionar) este problema, se edita el archivo *RunWeka.ini* modificando la línea *maxheap=128m* (pe.por maxheap=1360m), así se incrementa el valor máximo del *heap size* que utilizará JVM.

Para ilustrar las capacidades de extracción de datos de WEKA, se utilizará la consulta de la Tabla 7.12 en el WEKA-Explorer.

[3] Los pasos para realizar conexiones ODBC en MS-Windows 2000 están fuera del alcance del proyecto, y se consideran como conocimientos básicos del uso del sistema operativo.

```
SELECT
Tipo_usuario.device_pool,
Tipo_usuario.anexo,
Tipo_usuario.modelo_equipo,
Tiempo.mes_nombre,Tiempo.dia_nombre,
Llamadas.duracion_llamadas
FROM Llamadas, Tipo_usuario, Tiempo
WHERE Llamadas.id_device_o = Tipo_usuario.id_device
AND Llamadas.id_tiempo = Tiempo.id_tiempo
AND tiempo.anho='2007'
```

Tabla 7.12 Consulta para obtener 6 atributos para WEKA

Luego de unos minutos, WEKA carga los resultados y despliega información estadística (Ilustración 7.15). Es recomendable que, si los datos cargados son satisfactorios, se guarden como archivo ARFF[4] para que su carga posterior sea mucho más rápida.

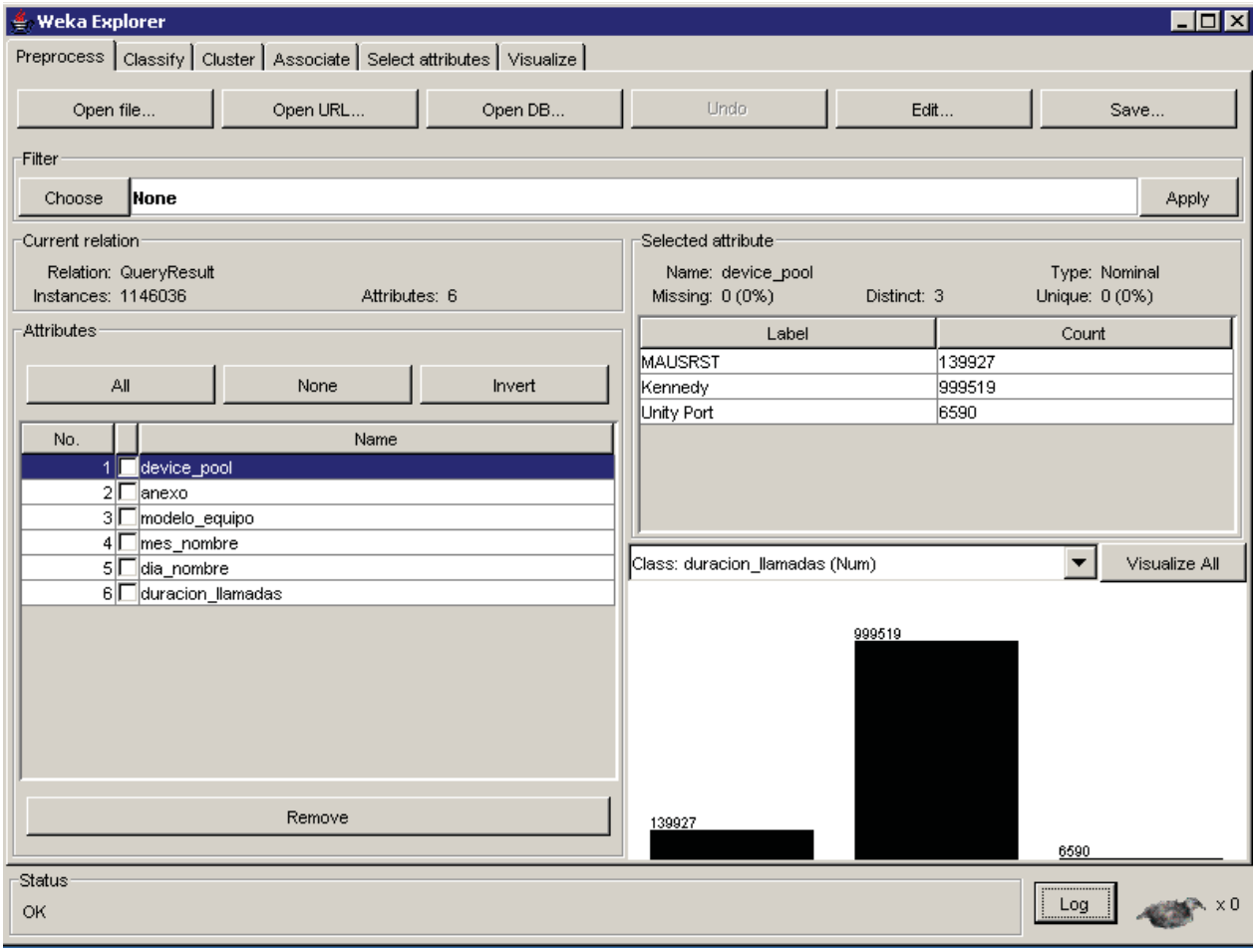


Ilustración 7.15 WEKA Explorer

[4] WEKA utiliza de modo particular el formato de archivo con extensión arff.

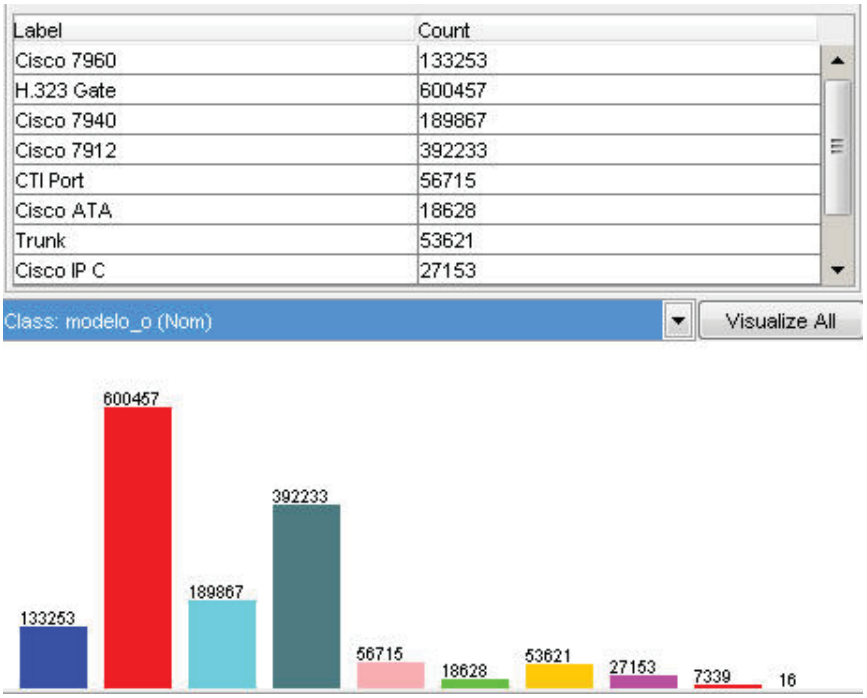


Ilustración 7.16 Instancias de modelos de origen

En la Ilustración 7.16, se ve en detalle el despliegue de información de la opción de pre-procesado de WEKA. En este caso se tiene una estadística del atributo "modelo_o", que representa el modelo del equipo que origina una llamada. El gráfico muestra que el modelo "H.323 Gate" (color rojo) es el de mayor instancias, seguido de "Cisco 7912" (color verde) y "Cisco 7940" (color calipso).

Como muestra la Ilustración 7.15, se cargaron 6 atributos en WEKA-Explorer. Existen 1.146.036 instancias. La Ilustración 7.17, muestra una matriz en que cada atributo se evalúa con los demás.

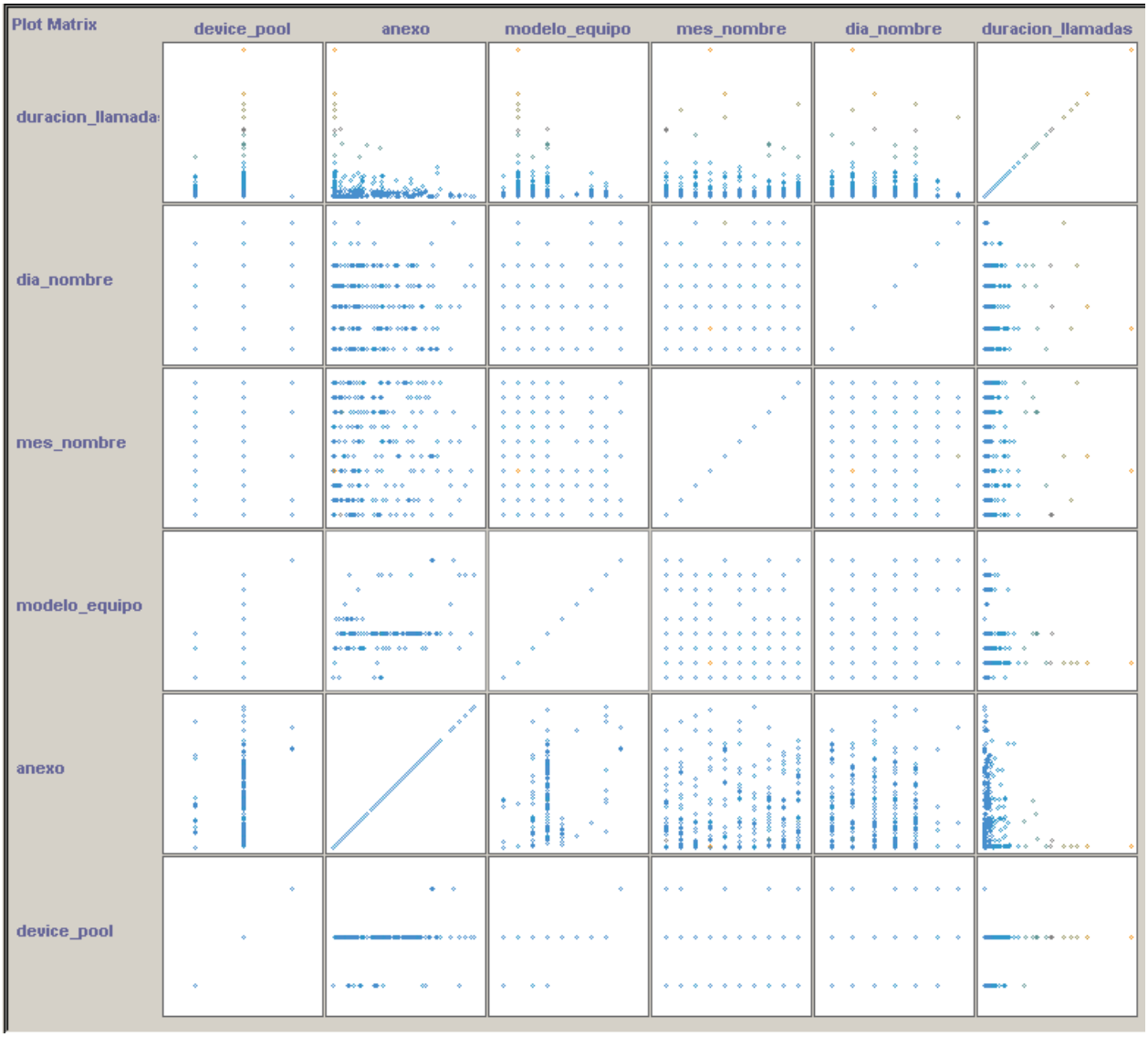


Ilustración 7.17 Matriz de atributos en WEKA

Cada intersección de la matriz es ampliable, y muestra el cruce con mayor resolución. Cada intersección representa el cruce entre 2 atributos. Los que aparecen a la izquierda representan el eje de las Y en un plano cartesiano, los atributos que aparecen arriba, representan el eje X. Y los puntos son las ocurrencias o instancias de in atributo respecto de otro. Por ejemplo, el primer cuado describe la duración de llamadas por cada device pool, y aparecen 3 columnas de puntos, puesto que existen 3 device pool en nuestros datos. La columna central tiene mayor cantidad de puntos porque es la que tiene mayor cantidad de instancias, lo que implica que ese device pool es el que utiliza más la telefonía, respecto a los otros dos.

7.2.2 *Algoritmo Apriori en WEKA*

Para el análisis de minería de datos, se utilizó el algoritmo Apriori por medio de la implementación incluida en la herramienta WEKA, la cual, permite las siguientes configuraciones del algoritmo [Witten y Frank, 2005]:

- **delta**: factor de decremento del soporte en cada iteración. Reduce el soporte hasta que el mínimo requerido es alcanzado o hasta que el número de reglas requeridas han sido generadas.

- **lowerBoundMinSupport**: soporte mínimo para el algoritmo.

- **metricType**: corresponde al tipo de métrica, con el cual, se evalúan las reglas. Pudiendo ser:

    - **Confidence** (Confianza): es la proporción de conjuntos que cumplen el antecedente, que a su vez cumplen con el consecuente.
    - **Lift**: es la confianza dividida por la proporción de todos los conjuntos que cumplen con el consecuente. Mide la independencia de la asociación con el soporte.
    - **Leverage**: es la proporción de conjuntos adicionales que cumplen con el antecedente y el consecuente, que no han sido considerados estos, en caso de que el antecedente fuera independiente del consecuente.
    - **Conviction**: es otra medida basada en la independencia. Está dada por:

$$\text{Conviction} = \frac{P(\text{antecedente})\, P(!\ \text{Consecuencia})}{P(\text{antecedente},!\ \text{Consecuencia})}$$

- **minMetric**: es el mínimo valor que puede tomar la métrica escogida. Se considerarán las reglas que tengan un valor superior a este.

- **numRules**: número de reglas a encontrar

- **outputItemSets**: habilita/deshabilita el despliegue de los itemset encontrados.

- **removeAllMissingCols**: Remueve las columnas con valores no encontrados.

- **significanceLevel**: Nivel de significancia. Test de significancia sólo para la métrica de confianza**.**

- **upperBoundMinSupport**: Límite superior del apoyo mínimo. Este valor va decreciendo en cada iteración hasta llegar al soporte mínimo requerido.

- **verbose**: habilita/deshabilita el despliegue de resultados explicados, no sólo los números.

La configuración base para la aplicación del algoritmo se describe en la Tabla 7.13.

delta	0.05 (valor por defecto)
lowerBoundMinSupport	(variable)
metricType	Confidence (valor por defecto)
minMetric	(variable)
numRules	10 (valor por defecto)
outputItemSets	True
removeAllMissingCols	False (valor por defecto)
significationLevel	-1.0 (valor por defecto)
upperBpundMinSupport	1.0 (valor por defecto)
verbose	True

Tabla 7.13 Configuración base de algoritmo Apriori

7.2.3 *Ejecuciones de Apriori con datos reales*

A continuación, las pruebas realizadas con distintos atributos y configuraciones de Apriori.

Caso a) **Modelo Origen y Modelo Destino**. En este caso, se realizó una consulta SQL para obtener los modelos de teléfonos que originaron llamadas y el modelo de teléfono del receptor.

Se realizaron 4 ejecuciones de Apriori con distintas configuraciones (Tabla 7.14).

Prueba	Soporte (Ts)	Confianza (Tc)
Prueba 1	0.1	0.6
Prueba 2	0.1	0.7
Prueba 3	0.1	0.8
Prueba 4	0.2	0.6

Tabla 7.14 Configuración pruebas caso *a*.

i. Prueba 1: En la primera prueba del caso *a* (Tabla 7.15), el algoritmo realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la confianza planteada para esta prueba (0,6). Se encontraron 7 *Large Itemsets* de 1 atributo y 2 *Large Itemsets* de 2 atributos.

A continuación los resultados:

Instances:	1479282
Attributes:	2
	modelo_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	18
Large itemsets L(1):	7
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	2
	modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293
Best rules found: 1. modelo_o=Cisco 7940 189867 ==> modelo_d=H.323 Gate 148085 conf:(0.78) 2. modelo_o=Cisco 7912 392233 ==> modelo_d=H.323 Gate 261293 conf:(0.67)	

Tabla 7.15 Caso *a*. Prueba 1.

Análisis de los resultados: Apriori encontró 2 reglas de asociación (Tabla 7.15). De ellas se concluye:

1. Quien realiza una llamada desde un teléfono 7940, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7940 está sobre del 10% de las llamadas. Y una confianza del 78%, es decir, el 78% de éstos realiza una llamada a la red pública.
2. Quien realiza una llamada desde un teléfono 7912, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 está sobre del 10% de las llamadas. Y una confianza del 67%, es decir, el 67% de éstos realiza una llamada a la red pública.

ii. Prueba 2: En la segunda prueba del caso *a* (Tabla 7.16), sólo se aumentó el valor de la confianza a 0,7. El algoritmo también realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la nueva confianza planteada para esta prueba (0,7). Se encontraron los mismos *Large Itemsets*, debido a que éstos tienen directa relación con el soporte, el cual, en ésta prueba no varió.

A continuación los resultados:

Instances:	1479282
Attributes:	2
	modelo_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.7
Number of cycles performed:	18
Large itemsets L(1):	7
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	2
	modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293
Best rules found: 1. modelo_o=Cisco 7940 189867 ==> modelo_d=H.323 Gate 148085 conf:(0.78)	

Tabla 7.16 Caso *a*. Prueba 2.

Análisis de los resultados: Apriori encontró 1 regla de asociación (Tabla 7.16), igual a la primera encontrada en la prueba 1, ya que sólo se incrementó la confianza en este caso. De la regla encontrada se concluye que:

1. Quien realiza una llamada desde un teléfono 7940, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7940 está sobre del 10% de las llamadas. Y una confianza del 78%, es decir, el 78% de éstos realiza una llamada a la red pública.

iii. Prueba 3: En la tercera prueba del caso *a* (Tabla 7.17), nuevamente se aumentó el valor de la confianza, ahora a 0,8. El algoritmo también realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la nueva confianza planteada para esta prueba (0,8). Se encontraron los mismos *Large Itemsets*, debido a que éstos tienen directa relación con el soporte, el cual, en ésta prueba no varió.

A continuación los resultados:

Instances:	1479282
Attributes:	2
	modelo_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.8
Number of cycles performed:	18
Large itemsets L(1):	7
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	2
	modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293
Best rules found:	

Tabla 7.17 Caso *a*. Prueba 3.

Análisis de los resultados: Esta ejecución de Apriori no encontró regla de asociación alguna (Tabla 7.17).

iv. En la cuarta y última prueba del caso *a* (Tabla 7.18), nuevamente se aumentó el valor de la confianza, ahora a 0,9. Pero esta vez el algoritmo no realizó ningún ciclo o iteración debido a que no encontró *Large Iterms* ni tampoco reglas de asociación.

Tabla 7.18 Caso *a*. Prueba 4.

Instances:	1479282
Attributes:	2
	modelo_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.9
Number of cycles performed:	0
No large itemsets and rules found!	

Caso b) **Modelo Origen, Device Pool Origen y Modelo Destino**. En este caso, se realizó una consulta SQL para obtener los modelos de teléfonos que originaron llamadas, su device pool y el modelo de teléfono del receptor.

Se realizaron 8 ejecuciones de Apriori con distintas configuraciones (Tabla 7.19).

Prueba	Soporte (Ts)	Confianza (Tc)
Prueba 1	0.1	0.6
Prueba 2	0.1	0.7
Prueba 3	0.1	0.8
Prueba 4	0.1	0.9
Prueba 5	0.2	0.6
Prueba 6	0.3	0.6
Prueba 7	0.4	0.6
Prueba 8	0.5	0.6

Tabla 7.19 Configuración y resultados pruebas caso *b*.

i. Prueba 1: En la primera prueba del caso *b* (Tabla 7.20), el algoritmo realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la confianza planteada para esta prueba (0,6). Se encontraron 9 *Large Itemsets* de 1 atributo, 8 *Large Itemsets* de 2 atributos y 1 *Large Itemset* de 3 atributos.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	18
Large itemsets L(1):	9
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 device_pool_o=MAUSRST 207809 device_pool_o=Kennedy 1264134 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	8
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 device_pool_o=Kennedy 374360 modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 device_pool_o=Kennedy modelo_d=Cisco 7912 209410 device_pool_o=Kennedy modelo_d=Voice Mail 169141 device_pool_o=Kennedy modelo_d=H.323 Gate 565937 device_pool_o=Kennedy modelo_d=Cisco 7940 152793
Large Itemsets L(3):	1
	modelo_o=Cisco 7912 device_pool_o=Kennedy modelo_d=H.323 Gate 245976

Best rules found:

1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1)
2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95)
3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94)
4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94)
5. modelo_d=Cisco 7940 173758 ==> device_pool_o=Kennedy 152793 conf:(0.88)
6. modelo_d=Cisco 7912 246738 ==> device_pool_o=Kennedy 209410 conf:(0.85)
7. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)
8. modelo_o=Cisco 7940 189867 ==> modelo_d=H.323 Gate 148085 conf:(0.78)
9. modelo_o=Cisco 7912 392233 ==> modelo_d=H.323 Gate 261293 conf:(0.67)
10. modelo_o=Cisco 7912 device_pool_o=Kennedy 374360 ==> modelo_d=H.323 Gate 245976 conf:(0.66)

Tabla 7.20 Caso *b*. Prueba 1.

Análisis de los resultados: Apriori encontró 10 reglas de asociación (Tabla 7.20). De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quien realiza una llamada desde la red pública está sobre del 10% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos “Kennedy”.
2. Quien realiza una llamada desde un teléfono 7912, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 está sobre del 10% de las llamadas. Y una confianza del 95%, es decir, el 95% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.
3. Quien realiza una llamada al correo de voz, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quienes realiza una llamada al correo de voz está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.
4. Quien realiza una llamada desde un teléfono 7912 a la red pública, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 para una llamada a la red pública está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstas llamadas se realizan desde grupo de teléfonos “Kennedy”.
5. Quien realiza una llamada a un teléfono 7940, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quienes realizan una llamada a un teléfono 7940 están sobre del 10% de las llamadas. Y una confianza del 88%, es decir, el 88% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.
6. Quien realiza una llamada a un teléfono 7912, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quienes realizan una llamada a un teléfono 7912 están sobre del 10% de las llamadas. Y una confianza del 85%, es decir, el 85% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.

7. Quien realiza una llamada a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realizan llamadas a la red pública están sobre del 10% de las llamadas. Y una confianza del 81%, es decir, el 81% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
8. Quien realiza una llamada desde un teléfono 7940, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7940 está sobre del 10% de las llamadas. Y una confianza del 78%, es decir, el 78% de éstos realiza una llamada a la red pública.
9. Quien realiza una llamada desde un teléfono 7912, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 está sobre del 10% de las llamadas. Y una confianza del 67%, es decir, el 67% de éstos realiza una llamada a la red pública.
10. Quien realiza una llamada desde un teléfono 7912 desde grupo de teléfonos "Kennedy", lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 desde grupo de teléfonos "Kennedy" está sobre del 10% de las llamadas. Y una confianza del 66%, es decir, el 66% de éstos realiza una llamada a la red pública.

ii. Prueba 2: En la segunda prueba del caso *b* (Tabla 7.21), sólo se aumentó el valor de la confianza a 0,7. El algoritmo realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la nueva confianza planteada para esta prueba (0,7). Se encontraron los mismos *Large Itemsets*, debido a que éstos tienen directa relación con el soporte, el cual, en ésta prueba no varió.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.7
Number of cycles performed:	18
Large itemsets L(1):	9
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 device_pool_o=MAUSRST 207809 device_pool_o=Kennedy 1264134 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	8
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 device_pool_o=Kennedy 374360

	modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 device_pool_o=Kennedy modelo_d=Cisco 7912 209410 device_pool_o=Kennedy modelo_d=Voice Mail 169141 device_pool_o=Kennedy modelo_d=H.323 Gate 565937 device_pool_o=Kennedy modelo_d=Cisco 7940 152793
Large Itemsets L(3):	1
	modelo_o=Cisco 7912 device_pool_o=Kennedy modelo_d=H.323 Gate 245976
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94) 5. modelo_d=Cisco 7940 173758 ==> device_pool_o=Kennedy 152793 conf:(0.88) 6. modelo_d=Cisco 7912 246738 ==> device_pool_o=Kennedy 209410 conf:(0.85) 7. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81) 8. modelo_o=Cisco 7940 189867 ==> modelo_d=H.323 Gate 148085 conf:(0.78)	

Tabla 7.21 Caso *b*. Prueba 2.

Análisis de los resultados: Apriori encontró 8 reglas de asociación (Tabla 7.21), iguales a las primeras 8 encontrada en la prueba 1, ya que sólo se incrementó la confianza en este caso. De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quien realiza una llamada desde la red pública está sobre del 10% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos “Kennedy”.
2. Quien realiza una llamada desde un teléfono 7912, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 está sobre del 10% de las llamadas. Y una confianza del 95%, es decir, el 95% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.
3. Quien realiza una llamada al correo de voz, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quienes realiza una llamada al correo de voz está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.
4. Quien realiza una llamada desde un teléfono 7912 a la red pública, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 para una llamada a la red pública está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstas llamadas se realizan desde grupo de teléfonos “Kennedy”.
5. Quien realiza una llamada a un teléfono 7940, lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 10%, es decir, que quienes realizan una llamada a un teléfono 7940 están sobre del 10% de las llamadas. Y una confianza del 88%, es decir, el 88% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.

6. Quien realiza una llamada a un teléfono 7912, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realizan una llamada a un teléfono 7912 están sobre del 10% de las llamadas. Y una confianza del 85%, es decir, el 85% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
7. Quien realiza una llamada a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realizan llamadas a la red pública están sobre del 10% de las llamadas. Y una confianza del 81%, es decir, el 81% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
8. Quien realiza una llamada desde un teléfono 7940, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7940 está sobre del 10% de las llamadas. Y una confianza del 78%, es decir, el 78% de éstos realiza una llamada a la red pública.

iii. Prueba 3: En la tercera prueba del caso *b* (Tabla 7.22), sólo se aumentó el valor de la confianza a 0,8. El algoritmo nuevamente realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la nueva confianza planteada para esta prueba (0,8). Se encontraron los mismos *Large Itemsets*, debido a que éstos tienen directa relación con el soporte, el cual, en ésta prueba no varió.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.8
Number of cycles performed:	18
Large itemsets L(1):	9
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 device_pool_o=MAUSRST 207809 device_pool_o=Kennedy 1264134 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	8
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 device_pool_o=Kennedy 374360 modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 device_pool_o=Kennedy modelo_d=Cisco 7912 209410 device_pool_o=Kennedy modelo_d=Voice Mail 169141 device_pool_o=Kennedy modelo_d=H.323 Gate 565937 device_pool_o=Kennedy modelo_d=Cisco 7940 152793
Large Itemsets L(3):	1

	modelo_o=Cisco 7912 device_pool_o=Kennedy modelo_d=H.323 Gate 245976
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94) 5. modelo_d=Cisco 7940 173758 ==> device_pool_o=Kennedy 152793 conf:(0.88) 6. modelo_d=Cisco 7912 246738 ==> device_pool_o=Kennedy 209410 conf:(0.85) 7. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)	

Tabla 7.22 Caso *b*. Prueba 3.

Análisis de los resultados: Apriori encontró 7 reglas de asociación (Tabla 7.22), iguales a las primeras 7 encontrada en la prueba 1 y la prueba 2, ya que sólo se incrementó la confianza en este caso. De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quien realiza una llamada desde la red pública está sobre del 10% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos "Kennedy".
2. Quien realiza una llamada desde un teléfono 7912, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 está sobre del 10% de las llamadas. Y una confianza del 95%, es decir, el 95% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
3. Quien realiza una llamada al correo de voz, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realiza una llamada al correo de voz está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
4. Quien realiza una llamada desde un teléfono 7912 a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 para una llamada a la red pública está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstas llamadas se realizan desde grupo de teléfonos "Kennedy".
5. Quien realiza una llamada a un teléfono 7940, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realizan una llamada a un teléfono 7940 están sobre del 10% de las llamadas. Y una confianza del 88%, es decir, el 88% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
6. Quien realiza una llamada a un teléfono 7912, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realizan una llamada a un teléfono 7912 están sobre del 10% de las llamadas. Y una confianza del 85%, es decir, el 85% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
7. Quien realiza una llamada a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realizan llamadas a la

red pública están sobre del 10% de las llamadas. Y una confianza del 81%, es decir, el 81% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".

iv. Prueba 4: En la cuarta prueba del caso *b* (Tabla 7.23), sólo se aumentó el valor de la confianza a 0,9. El algoritmo nuevamente realizó 18 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,1), o alcanzando la nueva confianza planteada para esta prueba (0,9). Se encontraron los mismos *Large Itemsets*, debido a que éstos tienen directa relación con el soporte, el cual, en ésta prueba no varió.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.1 (147928 instances)
Minimum metric <confidence>:	0.9
Number of cycles performed:	18
Large itemsets L(1):	9
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7940 189867 modelo_o=Cisco 7912 392233 device_pool_o=MAUSRST 207809 device_pool_o=Kennedy 1264134 modelo_d=Cisco 7912 246738 modelo_d=Voice Mail 179196 modelo_d=H.323 Gate 702426 modelo_d=Cisco 7940 173758
Large Itemsets L(2):	8
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 modelo_o=Cisco 7940 modelo_d=H.323 Gate 148085 modelo_o=Cisco 7912 device_pool_o=Kennedy 374360 modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 device_pool_o=Kennedy modelo_d=Cisco 7912 209410 device_pool_o=Kennedy modelo_d=Voice Mail 169141 device_pool_o=Kennedy modelo_d=H.323 Gate 565937 device_pool_o=Kennedy modelo_d=Cisco 7940 152793
Large Itemsets L(3):	1
	modelo_o=Cisco 7912 device_pool_o=Kennedy modelo_d=H.323 Gate 245976
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94)	

Tabla 7.23 Caso *b*. Prueba 4.

Análisis de los resultados: Apriori encontró 4 reglas de asociación (Tabla 7.23), iguales a las primeras 4 encontrada en la prueba 1, 2 y 3, ya que sólo se incrementó la confianza en este caso. De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quien realiza una llamada desde la red pública está sobre del 10% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos "Kennedy".
2. Quien realiza una llamada desde un teléfono 7912, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 está sobre del 10% de las llamadas. Y una confianza del 95%, es decir, el 95% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
3. Quien realiza una llamada al correo de voz, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quienes realiza una llamada al correo de voz está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
4. Quien realiza una llamada desde un teléfono 7912 a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 10%, es decir, que quien utiliza un teléfono 7912 para una llamada a la red pública está sobre del 10% de las llamadas. Y una confianza del 94%, es decir, el 94% de éstas llamadas se realizan desde grupo de teléfonos "Kennedy".

v. Prueba 5: En la quinta prueba del caso *b* (Tabla 7.24), se utilizó nuevamente el valor de la confianza 0,6, pero ahora incrementamos el soporte a 0,2. El algoritmo esta vez realizó 16 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,2), o alcanzando la confianza planteada para esta prueba (0,6). Se encontraron 4 *Large Itemsets* de 1 atributo, 3 *Large Itemsets* de 2 atributos y ningún *Large Itemset* de 3 atributo.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.2 (295856 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	16
Large itemsets L(1):	4
	modelo_o=H.323 Gate 600457 modelo_o=Cisco 7912 392233 device_pool_o=Kennedy 1264134 modelo_d=H.323 Gate 702426
Large Itemsets L(2):	3
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 modelo_o=Cisco 7912 device_pool_o=Kennedy 374360 device_pool_o=Kennedy modelo_d=H.323 Gate 565937

Large Itemsets L(3):	0
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)	

Tabla 7.24 Caso *b*. Prueba 5.

Análisis de resultados: Apriori encontró 3 reglas de asociación (Tabla 7.24). De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos "Kennedy". Con un soporte del 20%, es decir, que quien realiza una llamada desde la red pública está sobre del 20% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos "Kennedy".
2. Quien realiza una llamada desde un teléfono 7912, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 20%, es decir, que quien utiliza un teléfono 7912 está sobre del 20% de las llamadas. Y una confianza del 95%, es decir, el 95% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".
3. Quien realiza una llamada a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 20%, es decir, que quienes realizan llamadas a la red pública están sobre del 20% de las llamadas. Y una confianza del 81%, es decir, el 81% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".

vi. Prueba 6: En la sexta prueba del caso *b* (Tabla 7.25), se utilizó nuevamente el valor de la confianza 0,6, pero ahora incrementamos el soporte a 0,3. El algoritmo esta vez realizó 14 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,3), o alcanzando la confianza planteada para esta prueba (0,6). Se encontraron 3 *Large Itemsets* de 1 atributo, 2 *Large Itemsets* de 2 atributos y ningún *Large Itemset* de 3 atributo.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.3 (443785 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	14
Large itemsets L(1):	3
	modelo_o=H.323 Gate 600457 device_pool_o=Kennedy 1264134 modelo_d=H.323 Gate 702426
Large Itemsets L(2):	2
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 device_pool_o=Kennedy modelo_d=H.323 Gate 565937

Large Itemsets L(3):	0
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)	

Tabla 7.25 Caso *b*. Prueba 6.

Análisis de los resultados: Apriori encontró 2 reglas de asociación (Tabla 7.25). De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos "Kennedy". Con un soporte del 30%, es decir, que quien realiza una llamada desde la red pública está sobre del 30% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos "Kennedy".
2. Quien realiza una llamada a la red pública, lo hace desde grupo de teléfonos "Kennedy". Con un soporte del 30%, es decir, que quienes realizan llamadas a la red pública están sobre del 30% de las llamadas. Y una confianza del 81%, es decir, el 81% de éstos realiza una llamada desde grupo de teléfonos "Kennedy".

vii. Prueba 7: En la septima prueba de nuestro caso *b* (Tabla 7.26), se utilizó nuevamente el valor de la confianza 0,6, pero ahora incrementamos el soporte a 0,4. El algoritmo esta vez realizó 12 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,4), o alcanzando la confianza planteada para esta prueba (0,6). Se encontraron 3 *Large Itemsets* de 1 atributo, 1 *Large Itemsets* de 2 atributos y ningún *Large Itemset* de 3 atributo.

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.4 (591713 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	12
Large itemsets L(1):	3
	modelo_o=H.323 Gate 600457 device_pool_o=Kennedy 1264134 modelo_d=H.323 Gate 702426
Large Itemsets L(2):	1
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457
Large Itemsets L(3):	0

Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1)

Tabla 7.26 Caso *b*. Prueba 7.

Análisis de los resultados: Apriori encontró 1 regla de asociación (Tabla 7.26). De la que se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos "Kennedy". Con un soporte del 40%, es decir, que quien realiza una llamada desde la red pública está sobre del 40% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos "Kennedy".

viii. Prueba 8: En la octaba y última prueba de nuestro caso *b* (Tabla 7.27), se utilizó nuevamente el valor de la confianza 0,6, e incrementamos el soporte a 0,5. Pero esta vez el algoritmo no realizó ningún ciclo o iteración debido a que no encontró *Large Iterms* ni tampoco reglas de asociación (Tabla 7.27).

A continuación los resultados:

Instances:	1479282
Attributes:	3
	modelo_o device_pool_o modelo_d
Minimum support:	0.5
Minimum metric <confidence>:	0.6
Number of cycles performed:	0
Large itemsets L(1):	0
Large Itemsets L(2):	0
Large Itemsets L(3):	0
No large itemsets and rules found!	

Tabla 7.27 Caso *b*. Prueba 8.

Caso c) **Modelo Origen, Device Pool Origen, Horario de Cobro, Modelo Destino**. En este caso, se realizó una consulta SQL para obtener los modelos de teléfonos que originaron llamadas, su device pool, el horario de cobro en que se realizaron (alto o bajo) y el modelo de teléfono del receptor.

Se realizaron 3 ejecuciones de Apriori con distintas configuraciones (Tabla 7.28).

Prueba	Soporte (Ts)	Confianza (Tc)
Prueba 1	0.3 (0.1)	0.6
Prueba 2	0.4	0.6
Prueba 3	0.5	0.6

Tabla 7.28 Configuración pruebas caso *c*.

i. Prueba 1: En la primera prueba del caso *c* (Tabla 7.29), el algoritmo realizó 14 ciclos o iteraciones, deteniendose antes de llegar al mínimo soporte que se definió (0,1) sino que con un soporte de 0,3, la confianza planteada para esta prueba fue de 0,6. Se encontraron 4 *Large Itemsets* de 1 atributo, 5 *Large Itemsets* de 2 atributos, 2 *Large Itemsets* de 3 atributos y 0 *Large Itemset* de 4 atributos.

A continuación los resultados:

Instances:	1479282
Attributes:	4
	modelo_o device_pool_o horario_cobro modelo_d
Minimum support:	0.3 (443785 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	14
Large itemsets L(1):	4
	modelo_o=H.323 Gate 600457 device_pool_o=Kennedy 1264134 horario_cobro=0 1258841 modelo_d=H.323 Gate 702426
Large Itemsets L(2):	5
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 modelo_o=H.323 Gate horario_cobro=0 519989 device_pool_o=Kennedy horario_cobro=0 1085886 device_pool_o=Kennedy modelo_d=H.323 Gate 565937 horario_cobro=0 modelo_d=H.323 Gate 589644
Large Itemsets L(3):	2
	modelo_o=H.323 Gate device_pool_o=Kennedy horario_cobro=0 519989 device_pool_o=Kennedy horario_cobro=0 modelo_d=H.323 Gate 485093
Large Itemsets L(4):	0
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=H.323 Gate horario_cobro=0 519989 ==> device_pool_o=Kennedy 519989 conf:(1) 3. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy horario_cobro=0 519989 conf:(0.87) 4. modelo_o=H.323 Gate device_pool_o=Kennedy 600457 ==> horario_cobro=0 519989	

conf:(0.87) 5. modelo_o=H.323 Gate 600457 ==> horario_cobro=0 519989 conf:(0.87) 6. horario_cobro=0 1258841 ==> device_pool_o=Kennedy 1085886 conf:(0.86) 7. device_pool_o=Kennedy 1264134 ==> horario_cobro=0 1085886 conf:(0.86) 8. device_pool_o=Kennedy modelo_d=H.323 Gate 565937 ==> horario_cobro=0 485093 conf:(0.86) 9. modelo_d=H.323 Gate 702426 ==> horario_cobro=0 589644 conf:(0.84) 10. horario_cobro=0 modelo_d=H.323 Gate 589644 ==> device_pool_o=Kennedy 485093 conf:(0.82)

Tabla 7.29 Caso *c*. Prueba 1.

Análisis de los resultados: Apriori se encontró 10 reglas de asociación (Tabla 7.29). De ellas se concluye que:

1. Quien realiza una llamada desde la red pública, lo hace al grupo de teléfonos "Kennedy". Con un soporte del 30%, es decir, que quien realiza una llamada desde la red pública está sobre del 30% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada al grupo de teléfonos "Kennedy".
2. Quien realiza una llamada desde la red pública en horario de cobro bajo, lo hace al grupo de teléfonos "Kennedy". Con un soporte del 30%, es decir, que quien realiza una llamada desde la red pública en horario de cobro bajo está sobre del 30% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada al grupo de teléfonos "Kennedy".
3. Quien realiza una llamada desde la red pública, lo hace al grupo de teléfonos "Kennedy" en horario de cobro bajo. Con un soporte del 30%, es decir, que quien realiza una llamada desde la red pública está sobre del 30% de las llamadas. Y una confianza del 87%, es decir, el 87% de éstos realiza una llamada al grupo de teléfonos "Kennedy" en horario de cobro bajo.
4. Quien realiza una llamada desde la red pública al grupo de teléfonos "Kennedy" lo hace en horario de cobro bajo. Con un soporte del 30%, es decir, que quien realiza una llamada desde la red pública al grupo de teléfonos "Kennedy" está sobre del 30% de las llamadas. Y una confianza del 87%, es decir, el 87% de éstos realiza una llamada en horario de cobro bajo.
5. Quien realiza una llamada desde la red pública lo hace en horario de cobro bajo. Con un soporte del 30%, es decir, que quien realiza una llamada desde la red pública está sobre del 30% de las llamadas. Y una confianza del 87%, es decir, el 87% de éstos realiza una llamada en horario de cobro bajo.
6. Quien realiza una llamada en horario de cobro bajo lo hace desde el grupo de teléfonos "Kennedy". Con un soporte del 30%, es decir, que quien realiza una llamada en horario de cobro bajo está sobre del 30% de las llamadas. Y una confianza del 86%, es decir, el 86% de éstos realiza una llamada desde el grupo de teléfonos "Kennedy".
7. Quien realiza una llamada desde el grupo de teléfonos "Kennedy" lo hace en horario de cobro bajo. Con un soporte del 30%, es decir, que quien realiza una llamada desde el grupo de teléfonos "Kennedy" está sobre del 30% de las llamadas. Y una confianza del 86%, es decir, el 86% de éstos realiza una llamada en horario de cobro bajo.

8. Quien realiza una llamada desde el grupo de teléfonos “Kennedy” a la red pública lo hace en horario de cobro bajo. Con un soporte del 30%, es decir, que quien realiza una llamada desde el grupo de teléfonos “Kennedy” a la red pública está sobre del 30% de las llamadas. Y una confianza del 86%, es decir, el 86% de éstos realiza una llamada en horario de cobro bajo.
9. Quien realiza una llamada a la red pública lo hace en horario de cobro bajo. Con un soporte del 30%, es decir, que quien realiza una llamada a la red pública está sobre del 30% de las llamadas. Y una confianza del 84%, es decir, el 84% de éstos realiza una llamada en horario de cobro bajo.
10. Quien realiza una llamada en horario de cobro bajo a la red pública lo hace desde grupo de teléfonos “Kennedy”. Con un soporte del 30%, es decir, que quien realiza una llamada a la red pública en horario de cobro bajo está sobre del 30% de las llamadas. Y una confianza del 82%, es decir, el 82% de éstos realiza una llamada desde grupo de teléfonos “Kennedy”.

ii. Prueba 2: En la segunda prueba del caso *c* (Tabla 7.30), el algoritmo realizó 12 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,4) o al alcanzar la confianza planteada para esta prueba (0,6). Se encontraron 4 *Large Itemsets* de 1 atributo, 2 *Large Itemsets* de 2 atributos, 0 *Large Itemset* de 3 atributos y 0 *Large Itemset* de 4 atributos.

A continuación los resultados:

Instances:	1479282
Attributes:	4
	modelo_o device_pool_o horario_cobro modelo_d
Minimum support:	0.4 (591713 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	12
Large itemsets L(1):	4
	modelo_o=H.323 Gate 600457 device_pool_o=Kennedy 1264134 horario_cobro=0 1258841 modelo_d=H.323 Gate 702426
Large Itemsets L(2):	2
	modelo_o=H.323 Gate device_pool_o=Kennedy 600457 device_pool_o=Kennedy horario_cobro=0 1085886
Large Itemsets L(3):	0
Large Itemsets L(4):	0
Best rules found: 1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. horario_cobro=0 1258841 ==> device_pool_o=Kennedy 1085886 conf:(0.86) 3. device_pool_o=Kennedy 1264134 ==> horario_cobro=0 1085886 conf:(0.86)	

Tabla 7.30 Caso *c*. Prueba 2.

Análisis de los resultados: Apriori se encontró 3 reglas de asociación (Tabla 7.30). De ellas podemos concluir que:

1. Quien realiza una llamada desde la red pública, lo hace al grupo de teléfonos "Kennedy". Con un soporte del 40%, es decir, que quien realiza una llamada desde la red pública está sobre del 40% de las llamadas. Y una confianza del 100%, es decir, el 100% de estos realiza una llamada al grupo de teléfonos "Kennedy".
2. Quien realiza una llamada en horario de cobro bajo lo hace desde el grupo de teléfonos "Kennedy". Con un soporte del 40%, es decir, que quien realiza una llamada en horario de cobro bajo está sobre del 40% de las llamadas. Y una confianza del 86%, es decir, el 86% de estos realiza una llamada desde el grupo de teléfonos "Kennedy".
3. Quien realiza una llamada desde el grupo de teléfonos "Kennedy" lo hace en horario de cobro bajo. Con un soporte del 40%, es decir, que quien realiza una llamada desde el grupo de teléfonos "Kennedy" está sobre del 40% de las llamadas. Y una confianza del 86%, es decir, el 86% de estos realiza una llamada en horario de cobro bajo.

iii. Prueba 3: En la tercera prueba del caso *c* (Tabla 7.31), el algoritmo realizó 10 ciclos o iteraciones, deteniendose al llegar al mínimo soporte que se definió (0,5) o al alcanzar la confianza planteada para esta prueba (0,6). Se encontraron 2 *Large Itemsets* de 1 atributo, 1 *Large Itemsets* de 2 atributos, 0 *Large Itemset* de 3 atributos y 0 *Large Itemset* de 4 atributos.

A continuación los resultados:

Instances:	1479282
Attributes:	4
	modelo_o device_pool_o horario_cobro modelo_d
Minimum support:	0.5 (739641 instances)
Minimum metric <confidence>:	0.6
Number of cycles performed:	10
Large itemsets L(1):	2
	device_pool_o=Kennedy 1264134 horario_cobro=0 1258841
Large Itemsets L(2):	1
	device_pool_o=Kennedy horario_cobro=0 1085886
Large Itemsets L(3):	0
Large Itemsets L(4):	0
Best rules found: 1. horario_cobro=0 1258841 ==> device_pool_o=Kennedy 1085886 conf:(0.86) 2. device_pool_o=Kennedy 1264134 ==> horario_cobro=0 1085886 conf:(0.86)	

Tabla 7.31 Caso *c*. Prueba 3.

Análisis de los resultados: Apriori se encontró 2 reglas de asociación (Tabla 7.31). De ellas se concluye que:

1. Quien realiza una llamada en horario de cobro bajo lo hace desde el grupo de teléfonos "Kennedy". Con un soporte del 40%, es decir, que quien realiza una llamada en horario de cobro bajo está sobre del 40% de las llamadas. Y una confianza del 86%, es decir, el 86% de éstos realiza una llamada desde el grupo de teléfonos "Kennedy".
2. Quien realiza una llamada desde el grupo de teléfonos "Kennedy" lo hace en horario de cobro bajo. Con un soporte del 40%, es decir, que quien realiza una llamada desde el grupo de teléfonos "Kennedy" está sobre del 40% de las llamadas. Y una confianza del 86%, es decir, el 86% de éstos realiza una llamada en horario de cobro bajo.

7.2.4 *Conclusiones de Apriori con datos reales*

Se analizará el algoritmo en su comportamiento con las distintas configuraciones y los resultados.

La Tabla 7.32 muestra los resultados obtenidos del caso de prueba *a*, donde se utilizaron el modelo de equipo de origen de las llamadas y el modelo de equipo de destino. Se aprecia que en los primeros 3 casos, no se modificó el soporte, sino que se incrementó la confianza. Es por ello que los *Large Itemsets* no variaron en estos 3 casos, pero sí las cantidad de reglas encontradas.

Prueba	Soporte (Ts)	Confianza (Tc)	Ciclos	Reglas encontradas	L(1)	L(2)
Prueba 1	0.1	0.6	18	2	7	2
Prueba 2	0.1	0.7	18	1	7	2
Prueba 3	0.1	0.8	18	0	7	2
Prueba 4	0.2	0.6	0	0	0	0

Tabla 7.32 Resultados pruebas caso *a*.

La regla más importante fue la que se repitió en las primeras 2 pruebas (ya que en las otras 2 no se encontraron reglas). De ésta regla se infirió que quien realiza una llamada desde un teléfono 7940, lo hace a la red pública. Con un soporte del 10%, es decir, que quien utiliza un teléfono 7940 está sobre del 10% de las llamadas. Y una confianza del 78%, es decir, el 78% de éstos realiza una llamada a la red pública.

La Tabla 7.33 muestra los resultados obtenidos del caso de prueba *b*, donde se utilizó el modelo de equipo de origen de las llamadas, su device pool y el modelo de equipo de destino. Se ve en los primeros 4 casos que al mantener el valor del soporte e incremetar la confiaza, disminuye la cantidad de reglas encontradas, pero se mantiene la cantidad de *Large Itemsets*. En cambio en los 4 últimos casos, se ve que al mantener la confianza e incrementar el soporte, disminuye drásticamente la cantidad de *Large Ítems* y también las reglas encontradas. Luego, el nivel de soporte determina la cantidad de *Large Ítems* y junto con la confianza, determinan la cantidad de reglas encontradas.

Prueba	Soporte (Ts)	Confianza (Tc)	Ciclos	Reglas encontradas	L(1)	L(2)	L(3)
Prueba 1	0.1	0.6	18	10	9	8	1
Prueba 2	0.1	0.7	18	8	9	8	1
Prueba 3	0.1	0.8	18	7	9	8	1
Prueba 4	0.1	0.9	18	4	9	8	1
Prueba 5	0.2	0.6	16	3	4	3	0
Prueba 6	0.3	0.6	14	2	3	2	0
Prueba 7	0.4	0.6	12	1	3	1	0
Prueba 8	0.5	0.6	0	0	0	0	0

Tabla 7.33 Resultados pruebas caso *b*.

Cuando se incrementa la confianza, con el mismo valor de soporte, se descartan las reglas que están bajo la nueva confianza, siendo las que quedan, las mismas que en la prueba anterior pero sólo las con confianza superior a la nueva. En el caso *b*, esto se aprecia en la Tabla 7.34.

Pruebas	Reglas Ecnontradas
Prueba 1	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94) 5. modelo_d=Cisco 7940 173758 ==> device_pool_o=Kennedy 152793 conf:(0.88) 6. modelo_d=Cisco 7912 246738 ==> device_pool_o=Kennedy 209410 conf:(0.85) 7. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81) 8. modelo_o=Cisco 7940 189867 ==> modelo_d=H.323 Gate 148085 conf:(0.78) 9. modelo_o=Cisco 7912 392233 ==> modelo_d=H.323 Gate 261293 conf:(0.67) 10. modelo_o=Cisco 7912 device_pool_o=Kennedy 374360 ==> modelo_d=H.323 Gate 245976 conf:(0.66)
Prueba 2	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94) 5. modelo_d=Cisco 7940 173758 ==> device_pool_o=Kennedy 152793 conf:(0.88) 6. modelo_d=Cisco 7912 246738 ==> device_pool_o=Kennedy 209410 conf:(0.85) 7. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81) 8. modelo_o=Cisco 7940 189867 ==> modelo_d=H.323 Gate 148085 conf:(0.78)
Prueba 3	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94) 5. modelo_d=Cisco 7940 173758 ==> device_pool_o=Kennedy 152793 conf:(0.88) 6. modelo_d=Cisco 7912 246738 ==> device_pool_o=Kennedy 209410 conf:(0.85) 7. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)
Prueba 4	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=Voice Mail 179196 ==> device_pool_o=Kennedy 169141 conf:(0.94) 4. modelo_o=Cisco 7912 modelo_d=H.323 Gate 261293 ==> device_pool_o=Kennedy 245976 conf:(0.94)

Tabla 7.34 Reglas 4 primeras pruebas caso *b*

En cambio, si se modifica el soporte, se decrementan rápidamente los ciclos y la cantidad de *Large Itemsets* encontrados, como se ve en la Tabla 7.33. Pero en estos casos, las reglas, si bien son las mismas, en una prueba posterior con mayor soporte puede no aparecer una con una confianza mayor que otra. Esto se aprecia en la Tabla 7.35, en la prueba 5 (soporte 0,3) aparece una regla con confianza del 95%, pero en la prueba 6 (soporte 0,4) no pararece esta regla, pero sí una con confianza del 81% que tambien estaba entre los resultados de la prueba 5.

Pruebas	Reglas encontradas
Prueba 5	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_o=Cisco 7912 392233 ==> device_pool_o=Kennedy 374360 conf:(0.95) 3. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)
Prueba 6	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1) 2. modelo_d=H.323 Gate 702426 ==> device_pool_o=Kennedy 565937 conf:(0.81)
Prueba 7	1. modelo_o=H.323 Gate 600457 ==> device_pool_o=Kennedy 600457 conf:(1)

Tabla 7.35 Reglas pruebas 5,6 y 7 caso *b*

Esto se explica porque el valor del soporte tiene efecto en la búsqueda de los *Large Itemsets*, y luego, si estos estan bajo el soporte no se considerarán aunque sirvan en la construcción de reglas de alto grado de confianza.

La regla más importante fue la que se repitío en las primeras 7 pruebas (ya que en la octava prueba no se encontraron reglas). De ésta regla se infirió que quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos “Kennedy”. Con un soporte del 40%, es decir, que quien realiza una llamada desde la red pública está sobre del 40% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo de teléfonos “Kennedy”.

La Tabla 7.36 muestra los resultados obtenidos del caso de prueba *c*, donde se utilizaró el modelo de equipo de origen de las llamadas, su device pool, el horario de cobro (0 bajo, 1 alto) y el modelo de equipo de destino. Se aprecia que en el primer caso el algotirmo acabó con un soporte mayor al mínimo dado, esto debido a que primero se encontraron las 10 reglas requeridas. En este caso no se incrementó la confianza, sino el soporte.

Prueba	Soporte (Ts)	Confianza (Tc)	Ciclos	Reglas encontradas	L(1)	L(2)	L(3)	L(4)
Prueba 1	0.3 (0.1)	0.6	14	10	4	5	2	0
Prueba 2	0.4	0.6	12	3	4	2	0	0
Prueba 3	0.5	0.6	10	2	2	1	0	0

Tabla 7.36 Resultados pruebas caso *c*.

En este caso se encontraron 2 reglas importantes. La primera es una que se repite desde el caso *b*, y es que quien realiza una llamada desde la red pública, lo hace por gateway del grupo de teléfonos “Kennedy”. Con un soporte del 40%, es decir, que quien realiza una llamada desde la red pública está sobre del 40% de las llamadas. Y una confianza del 100%, es decir, el 100% de éstos realiza una llamada por gateway del grupo

de teléfonos “Kennedy”. La segunda regla es aquella que dice que quien realiza una llamada en horario de cobro bajo lo hace desde el grupo de teléfonos “Kennedy”. Con un soporte del 40%, es decir, que quien realiza una llamada en horario de cobro bajo está sobre del 40% de las llamadas. Y una confianza del 86%, es decir, el 86% de éstos realiza una llamada desde el grupo de teléfonos “Kennedy”. Y viceversa. Luego, esta regla es una doble implicancia o un “si solo si”.

Luego de éste análisis, se está en condiciones de recomendar a los encargados del mantenimiento del sistema de telefonía IP, una mayor preocupación por al grupo de teléfonos “Kennedy”, ya que es un actor importante en los análisis y participa en muchas de las reglas generadas por Apriori. Una recomendación podría ser la inclusión de un nuevo Callmanager que trabaje en cluster con el actual, para así mantener la operatividad del sistema.

Tambien es recomendable que se consideren los modelos de teléfonos 7940 como actores importantes en las llamadas telefónicas. Ya que si bien, están en menor cantidad de instancias respecto de los modelos 7912 (Ilustración 7.16), son los terceros en este ranking.

Otro modelo importante, y que aparecío en varias reglas encontradas por Apriori, es el “H.323 Gate” que es el gateway de acceso a la red pública de telefonía. Se puede decir de éste, que es utilizado frecuentemente en horarios de cobro bajo, por lo que no debería presentar mayores costos para la empresa. Además, se entiende por el horario de trabajo, el cual, es de oficina.

En conclusión, es posible afirmar que el grupo de teléfonos “Kennedy” es de gran importancia en éste sistema, que el “H.323 Gate” es ampliamente utilizado en horario de cobro bajo, y que los modelo de teléfono 7940 realiza gran cantidad de llamadas.

7.3 Consideraciones y tiempos de respuesta

Durante el desarrollo del proyecto, se encontró varios detalles que pueden provocar problemas en futuros desarrollos. Estos son:

a) En SQL Server existen algunos problemas de compatibilidad con Analysis Services cuando se utilizan *Unique Identifier* (GUID), como campos de claves primarias que auto-generan un número único. Por lo que a veces es recomendable transformar el tipo de dato del campo identificador a *varchar* (u otro equivalente) cuando la base de datos CCM_DW está cargada.

b) Para evitar o corregir la excepción *OutOfMemory* al usar WEKA, hay que utilizar un PC con mucha RAM, y cambiar el valor *MaxHeap* (por defecto en 128mb) en el archivo *weka.ini*.

A continuación se presenta la Tabla 7.37 con tiempos de respuesta en algunos procesos del desarrollo del proyecto.

Proceso	Tiempo utilizado para realizar la tarea
Ejecución de DTS (4 TDT)	6:14 hrs.
Ejecución consulta de actualización de valores de llamada	1:03 mins
Carga de datos WEKA por ODBC	6:49 mins. (aprox.)
Carga de datos WEKA por archivo arff	9 a 20 segs.
Ejecuciones de Apriori e WEKA	55 a 80 segs.

Tabla 7.37 Tiempos de respuesta de algunos procesos

Cada consulta realizada sobre la base de datos, se guardó en un archivo .arff para optimizar los tiempos de carga en análisis u operaciones de Apriori posteriores.

Capítulo 8. Conclusiones

Del presente informe podemos concluir que se realizaron todas las tareas planteadas para su desarrollo, entre las cuales se incluían:

- Implementación de los diseños lógicos del Data Mart.
- Utilización de herramienta de transformación de datos de SQL Server (DTS).
- Implementación de un cubo con Analysis Services.
- Instalación y conexión de WEKA con el Data Mart.
- Ejecución del algoritmo A priori, usando distintos sets de pruebas, en WEKA.

El resultado ha sido muy satisfactorio. Se ha conseguido trabajar con éxito sobre una base de datos real y se encontraron reglas de asociación interesantes.

La incorporación de este tipo de análisis a una organización proveedora de tecnología agrega valor a sus servicios, y lo diferencia de sus competidores, posicionándola de manera preferencial en el mercado. Como se ha mencionado antes, hay empresas Chilenas que han creado Softwares aplicación para telefonía IP, tarificación de llamadas, etc. pero no hacen un análisis de los datos generados por el Callmanager.

Es posible encontrar información de valor en un conjunto de datos transaccionales de telefonía IP de Cisco, y éste valor se incrementaría si se cruzaran con datos de clientes (por ejemplo sistemas CRM, ERP, etc.) y con tablas o o sistemas de tarificación de llamadas.

Existen herramientas disponibles en el mercado, y herramientas libres que permiten el análisis de grandes volúmenes de datos, como lo son los de telefonía IP, y que pueden ser de gran utilidad en organizaciones que poseen estos sistemas de telefonía, como para aquellas que son proveedoras.

Teniendo claro el funcionamiento del sistema de telefonía IP basados en Callmanager de Cisco, y los Software a utilizar, es posible afirmar que el presente informe cumple con el objetivo del proyecto: “Presentar una guía para el desarrollo de un análisis de los datos históricos que genera un Callmanager de Cisco, utilizado en un sistema de telefonía IP, usando un Data Mart y algoritmos de minería de datos. Para así, poder dar un soporte científico y consistente a las decisiones de los directivos a cargo de un sistema de telefonía IP basados en éstos Callmanager de Cisco”.

Capítulo 9. Referencias

[Agrawal y Srikant, 1994] R. Agrawal y R. Srikant, Fast Algorithms for Mining Association Rules, http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf,
IBM Almaden Research Center, 1994.

[Cisco, 2006] Cisco Systems Inc., Cisco Unified CallManager Database Dictionary, 5.0(4), http://www.cisco.com/univercd/cc/td/doc/product/voice/vpdd/cdd/5_0/datadict/dd504.pdf,
Cisco Systems Inc. 2006.

[De la Mora, 2006] Roberto de la Mora, Telefonía IP, la única opción viable, http://www.gobiernoelectronico.org/?q=node/238,
Gobierno Electronico.org, Enero 2006.

[García, 2006] Diego García Morate, Manual de WEKA,
http://metaemotion.com/diego.garcia.morate,
Diego García Morate, 2006.

[Gutiérrez, 2006] Damián Gutiérrez Echeverría**,** Data Warehouse**,**
http://www.monografias.com/trabajos17/data-warehouse/data-warehouse.shtml,
monografías.com, 2003.

[Hernández y Ferri, 2006] José Hernández Orallo y Cèsar Ferri Ramírez, Curso de Doctorado Extracción Automática de Conocimiento en Bases de Datos e Ingeniería del Software, http://www.dsic.upv.es/~jorallo/docent/doctorat/weka.pdf,
Universitat Politècnica de València, Marzo 2006.

[INEI Perú, 1997] Instituto nacional de estadística e informática del Perú, Manual de construcción de un Data Warehouse,
INEI Perú 1997.

[Martí, 2005] Jose Luis Martí, Apuntes y diapositivas del curso: Tópicos especiales en bases de datos,
PUCV 2005.

[Microsoft, 2007] Microsoft, Información técnica SQL Server,
http://www.microsoft.com/spain/sql/2000/techinfo/default.aspx,
Microsoft 2007.

[Padrón, 2003] Liudmila Padrón Torres, Almacenes de datos: importancia de la estandarización de las direcciones para las empresas de hoy en día.
http://www.monografias.com/trabajos31/almacenes-datos/almacenes-datos.shtml,
monografías.com, Enero 2003.

[Peralta, 2001] Verónika Peralta, Diseño Lógico de Data Warehouses a partir de Esquemas Conceptuales Multidimensionales,
http://www.fing.edu.uy/inco/pedeciba/bibliote/tesis/tesis-vperalta.pdf, Tesis de Maestría, InCo - Pedeciba,
Universidad de la República, Montevideo, Uruguay, Noviembre 2001.

[Pérez de Armas, 2003] Dialys Nerely Pérez de Armas, El Datawarehouse: nueva perspectiva de consulta para las empresas,
http://www.monografias.com/trabajos16/datawarehouse/datawarehouse.shtml,
monografías.com, 2003.

[Subtel, 2005] Subsecretaría de Telecomunicaciones, Informe Estadístico Nº10 (Información a Junio 2005), Estadísticas de desempeño del sector de las telecomunicaciones en Chile. Junio 2004 – Junio 2005,
Subsecretaría de Telecomunicaciones, Diciembre de 2005.

[Witten y Frank, 2005] Ian H. Witten y Eibe Frank, Data Mining: Practical machine learning tools and techniques,
2da Edición, Morgan Kaufmann, San Francisco, 2005.

Anexo 1 : Listado de algoritmos de WEKA

La siguiente lista de algoritmos tiene por finalidad entregar una ayuda a futuras investigaciones usando WEKA, y mostrar sus capacidades en cuanto a la cantidad de algoritmos soportados. No se analizó cada algoritmo, sólo se utilizó Apriori.

- Clasificadores

Grupo	Nombre	Función
Bayesian Classifiers	AODE	AODE (Average done-dependence estimators): es un método Bayesiano que promedia más de un espacio de alternativo de modelos Bayesianos con hipótesis de independencia menor que Naive Bayes. El algoritmo puede ser más preciso en su clasificación que Naive Bayes en bases de datos con atributos no independientes.
	BayesNet	Entrena redes Bayesianas bajo hipótesis sobre atributos nominales (o numéricos discretizados) y sin valores perdidos.
	ComplementNaiveBayes	Construye un clasificador de complementos de Naive Bayes.
	NaiveBayes	Implementa el clasificador probabilístico Naive Bayes. Puede utilizar los estimadores de kernel para la densidad, que mejora el rendimiento si la hipótesis de normalidad es totalmente incorrecta. También puede manejar los atributos numéricos utilizando discretización supervisada.
	NaiveBayesMultinomial	Implementa el clasificador Bayesiano Multinomial.
	NaiveBayesSimple	Utiliza una distribución Normal para modelar atributos numéricos.
	NaiveBayesUpdateable	Es una versión incremental que procesa una instancia a la vez. Puede utilizar un estimador de kernel pero no discretizacion.
Trees	ADTree	Constuye arboles de decisión alternados.
	DecisionStump	Construye árboles de decisión de un nivel.
	Id3	Árbol de decisión de algoritmo dividir y conquistar básico.
	J48	Implemente árbol de decisión C4.5 (revisión 8).
	LMT	Construye árboles de modelado logístico.
	M5P	Learner de árbol de modelado M5'.
	NBTree	Construye un árbol de decisión con clasificadores Naive Bayes en las hojas.
	RandomForest	Construye bosques aleatorios.

	RandomTree	Construye un árbol que considera un número dado de características aleatorias en cada nodo.
	REPTree	Learner de árboles rápido que utiliza un reductor de error pruning
	UserClassifier	Permite al usuario construir su propio árbol de decisión.
Functions	LeastMedSq	Regresión robusta usando mediana en lugar de la media.
	LinearRegression	Regresión lineal estándar.
	Logistic	Construye modelos de regresión lineal logística.
	MultilayerPerceptron	Red neuronal que usa backpropagation.
	PaceRegression	Construye modelos de regresión lineal usando regresiones de Pace.
	RBFNetwork	Implementa una red de funciones básicas radiales.
	SimpleLinearRegresion	Aprende un modelo de regresión lineal basada en un solo atributo.
	SimpleLogistic	Construye modelos de regresiones lineales logisticas con con selección de atributos built-in.
	SMO	Algoritmo de Optimización Secuencial Mínima para soporte de vectores de clasificación.
	SMOreg	Algoritmo de Optimización Secuencial Mínima para soporte de vectores de regresión.
	VotedPerceptron	Algoritmo de perceptrón votado.
	Winnow	Perceptrón impulsado por error con actualizaciones multiplicativas.
Lazy	IB1	Learner basado en instancias de "vecino mas cercano" básico
	IBk	Clasificador de k-"vecinos más cercanos"
	KStar	"Vecino más cercano" con función de distancia generalizada.
	LBR	Clasificador de reglas Bayesianas Lazy
	LWL	Algoritmo general para aprendizaje local ponderado.
Misc.	HyperPipes	Learner rápido, extremadamente simple, basado en hiper-volúmenes en instancia espacial
	VFI	Método de la función de los intervalos de votación, simple y rápido.
Metalearnigs	AdaBoostM1	Boost usando el método de AdaBosstM1.
	AdditiveRegression	Mejora en el rendimiento de un método de regresión por acondicionamiento de residuos iterativo.
	AttributeSelectedClassifier	Reduce la dimensionalidad de la selección de datos por atributos.
	Bagging	Bag a classifier; funciona también para regresiones.
	ClassificationViaRegresion	Realiza una clasificación usando el método de la regresión
	CostSensitiveClassifier	Hace un clasificador sensible al costo
	CVParameterSelection	Realiza una selección de parámetros mediante una validación cruzada.
	Decorate	Construye conjuntos de clasificadores usando ejemplos de entrenamiento artificial especialmente construidos.
	FilteredClassifier	Ejecuta un clasificador en datos filtrados.
	Grading	Metalearners en que sus entradas son predicciones de nivel base, que han sido marcadas como correctas o incorrectas.
	LogitBoost	Realiza regresiones logísticas aditivas.
	MetaCost	Hace sensible al costo a un clasificador.
	MultiBoostAB	Combina boosting y bagging usando el método MultiBoosting.
	MultiClassClassifier	Usa un clasificador de 2 clases para set de datos multi-clases.
	MultiScheme	Usa validación cruzada para seleccionar un clasificador desde varios candidatos.

	OrdinalClassClassifier	Aplica algoritmos de clasificación estándar a problemas con un valor de clase ordinal.
	RacedIncrementalLogitBoost	Aprendizaje basado en lotes incrementales por racing logia-boosted comités.
	RandomCommitte	Construye un conjunto de clasificadores base aleatorios.
	RegressionByDiscretization	Discretiza el atributo de una clase y utiliza un clasificador.
	Stacking	Combina varios clasificadores usando el método de stacking.
	StackingC	Versión más eficiente de stacking.
	ThersholdSelector	Optimiza la medida F para un clasificador probabilístico.
	Vote	Combina clasificadores usando promedios de estimadores de probabilidad o predicciones numéricas.
Rules	ConjuctiveRule	Learner de reglas conjuntivas simple.
	DecisionTable	Construye una tabla de decisión simple con la mayoría de los clasificadores.
	JRip	Algoritmo Ripper para inducción de regla rápida y efectiva.
	M5Rules	Obtiene reglas desde modelos de árboles construidos usando M5'
	NNge	Método del "vecino más cercano" de generación de reglas usando ejemplos generalizados no anidados.
	OneR	Clasificador 1R.
	Part	Obtiene reglas desde árboles de decisión parciales construidos usando J4.8.
	Prism	Algoritmo de cobertura simple para reglas.
	Ridor	Learner de reglas Ripple-down.
	ZeroR	Predice la clase mayor (si es nominal) o el valor promedio (si es numérico).

Tabla Anexo 1.1 Algoritmos de clasificación

- Clusterización

Clusterers	Cobweb	Implementa los algoritmos de Clustering Cobweb y Classit.
	EM	Cluster usando maximización por expectación.
	FarthestFirst	Cluster usando el algoritmo transversal el primero más lejano.
	MakeDensityBasedCluster	Retoma un cluster para hacerlo retornar distribución y densidad.
	SimpleKMeans	Cluster usando el método k-meaans.

Tabla Anexo 1.2 Algoritmos de clusterización

- Asociaciones

Associations	Apriori	Encuentra reglas de asociación usando el algoritmo Apriori.
	PredictiveApriori	Encuentra reglas de asociación ordenados por predicción de ocurrencias.
	Tertius	Descubrimiento de asociaciones o reglas de clasificación guiado por confirmación.

Tabla Anexo 1.3 Algoritmos de asociaciones

- Selección de atributos

Attribute Subset Evaluator	CfsSubsetEval	Considera el valor de predicción de cada atributo individualmente, junto con el grado de redundancia entre ellos.
	ClassifierSubsetEval	Usa un clasificador para evaluar un set de atributos.
	ConsistencySubsetEval	Proyecta un set de entrenamiento en un set de atributos y mide la consistencia en los valores de clases.
	WrapperSubsetEval	Usa un clasificador más validación cruzada.
Single-Attribute evaluator	ChiSquaredAttributeEval	Computa la estadística Chi-Cuadrado de cada atributo respecto de la clase.
	GainRatioAttributeEval	Evalúa atributos basado en índice de ganancia.
	InfoGainAttributeEval	Evalúa atributos basado en ganancia de información.
	OneRAtributeEval	Usa la metodología OneR's para evaluar atributos.
	PrincipalComponents	Realiza un análisis y transformaciones de los principales componentes.
	ReliefFAttibuteEval	Evaluador de atributos basado en instancias.
	SVMAttributeEval	Usa una maquina de vectores de soportes lineales para determinar el valor de los atributos.
	SymmetricalUncertAttributeEval	Evalúa atributo basado en incertidumbre simétrica.

Tabla Anexo 1.4 Algoritmos de selección de atributos