

**PLATAFORMA PARA ACUERDOS DE NIVEL DE  
SERVICIO BASADA EN MINERÍA DE DATOS Y  
SERVICIOS WEB, PARA LA RED DE DATOS DE LA  
UNIVERSIDAD DEL CAUCA**

**Javier Mesa Durango  
Edwin Ariel Pino Ordoñez**

**Director:  
Ing. Juan Carlos Corrales M.**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICAIONES  
DEPARTAMENTO DE TELEMÁTICA  
Línea de Énfasis en Ingenierías de Sistemas Telemáticos  
Popayán, Febrero de 2005**

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN.....</b>	<b>5</b>
<b>CAPÍTULO 1 VALORACIÓN DE LA RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA ...</b>	<b>8</b>
1.1    INTRODUCCIÓN .....	8
1.2    ÁREA DE SERVIDORES Y SERVICIOS INTERNET .....	9
<b>CAPÍTULO 2 GESTIÓN BASADA EN SERVICIOS WEB .....</b>	<b>12</b>
2.1    SERVICIOS WEB .....	13
2.1.1  ARQUITECTURA .....	15
2.2    WSMF (WEB SERVICES MANAGEMENT FRAMEWORK).....	16
2.2.1  OBJETOS GESTIONADOS .....	18
2.2.2  EVENTOS Y NOTIFICACIONES .....	20
2.2.3  WSMF-FUNDACIÓN.....	21
2.2.4  WSMF-EVENTOS.....	29
2.2.5  WSMF-GESTIÓN DE SERVICIOS WEB. ....	29
2.3    WSDM (WEB SERVICES DISTRIBUTED MANAGEMENT).....	32
2.4    WSMN (WEB SERVICES MANAGEMENT NETWORK) .....	32
2.5    ACUERDOS DE NIVEL DE SERVICIO SLA .....	33
2.5.1  IMPLANTACIÓN DE ACUERDOS DE NIVEL DE SERVICIO CON PROVEEDORES. ....	34
2.5.2  ASP (PROVEEDOR DE SERVICIOS DE APLICACIÓN). ....	35
2.5.3  USUARIO FINAL .....	36
2.5.4  PROVEEDOR DE SERVICIOS DE INTERNET .....	36
2.5.5  IS .....	36
2.5.6  MÉTRICA .....	36
2.5.7  CONCEPTOS ACERCA DE LOS SLA.....	36
<b>CAPÍTULO 3 DESCUBRIMIENTO DEL CONOCIMIENTO EN BASES DE DATOS.....</b>	<b>39</b>
3.1    REPOSITORIO DE DATOS (DATA WAREHOUSE) .....	39
3.1.1  INTRODUCCIÓN .....	39
3.1.2  ARQUITECTURA .....	40
3.1.3  DATA WAREHOUSE COMO SISTEMA DE INFORMACIÓN .....	41
3.1.4  DATOS DE ENTRADA A UN DATA WAREHOUSE .....	41
3.1.5  CARACTERÍSTICAS DE LOS DATA WAREHOUSE.....	41
3.1.6  FASES DEL DATA WAREHOUSE.....	45
3.1.7  USUARIOS DATA WAREHOUSE .....	45
3.2    MINERÍA DE DATOS (DATA MINING) .....	46
3.2.1  TÉCNICAS DE EXTRACCIÓN DE DATOS .....	47
3.2.2  TIPOS DE ANÁLISIS DE DATOS .....	49
3.2.3  DESCUBRIMIENTO DEL CONOCIMIENTO EN BASES DE DATOS .....	50
3.2.4  CATEGORÍAS DE LA MINERÍA DE DATOS .....	52
3.2.5  ALGORITMOS DE MINERÍA DE DATOS .....	53
3.2.6  FASES MINERÍA DE DATOS .....	54
<b>CAPÍTULO 4 WSMINING: PLATAFORMA DE ACUERDOS DE NIVEL DE SERVICIO BASADA EN MINERÍA DE DATOS Y SERVICIOS WEB, PARA LA RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA .....</b>	<b>57</b>
4.1    ARQUITECTURA FUNCIONAL.....	57
4.2    ARQUITECTURA MODULAR .....	58
4.2.1  MÓDULO PRESENTACIÓN. ....	59
4.2.1.1  MODELO .....	59

4.2.1.2	VISTA .....	60
4.2.1.3	CONTROL.....	60
4.2.2	MÓDULO GESTIÓN DE ENTORNO .....	60
4.2.3	MÓDULO RECOLECCIÓN DE DATOS.....	60
4.2.4	MÓDULO ANÁLISIS DE DATOS .....	61
4.2.5	MÓDULO COMPARACIÓN DE RESULTADOS.....	61
4.3	ARQUITECTURA DE VALIDACIÓN .....	62
4.4	DESARROLLO DEL PROTOTIPO.....	64
4.4.1	CASOS DE USO.....	65
4.5	CONSTRUCCIÓN DEL DATA WAREHOUSE .....	75
4.6	BASES DE DATOS FUENTE.....	76
4.6.1	TABLAS .....	76
4.7	PROPUESTA PARA EL PROCESO DE DISEÑO DE UN DATA WAREHOUSE .....	77
4.7.1	FASE 1: ELABORACIÓN DEL MODELO CONCEPTUAL.....	79
4.7.1.1	REQUERIMIENTOS.....	79
4.7.1.2	ESQUEMA CONCEPTUAL.....	79
4.7.1.3	DIMENSIONES.....	80
4.7.1.4	NIVELES DE LAS DIMENSIONES .....	80
4.7.1.5	RELACIONES DIMENSIONALES .....	82
4.7.2	FASE 2: ELABORACIÓN DEL MODELO LÓGICO .....	83
4.7.2.1	LINEAMIENTOS.....	84
4.7.2.2	MAPEOS.....	86
4.7.2.3	APLICACIÓN DEL ALGORITMO .....	90
4.7.3	FASE 3: ELABORACIÓN DE LA ESTRUCTURA FÍSICA DEL DW .....	93
4.8	CONSTRUCCIÓN DE LA MINERÍA DE DATOS .....	96
4.8.1	PATRONES A DESCUBRIR .....	97
4.8.2	TIPO DE CONOCIMIENTO .....	98
4.8.3	INTERPRETACIÓN .....	99
<b>CAPÍTULO 5 CONCLUSIONES .....</b>		<b>101</b>
<b>CAPÍTULO 6 GLOSARIO .....</b>		<b>104</b>
<b>CAPÍTULO 7 REFERENCIAS Y BIBLIOGRAFÍA .....</b>		<b>106</b>

## LISTADO DE FIGURAS

FIGURA 1.1 RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA .....	9
FIGURA 1.2 ARQUITECTURA DE RED PARA EL SERVICIO DE PROXY .....	10
FIGURA 1.3 GRÁFICAS MRTG .....	11
FIGURA 2.1 OPERACIONES Y COMETIDOS DE LOS SERVICIOS WEB.....	15
FIGURA 2.2 PROTOCOLOS DE SERVICIOS WEB.....	16
FIGURA 2.3 ESQUEMA TRADICIONAL DE GESTIÓN DISTRIBUIDA .....	17
FIGURA 2.4 ESQUEMA DE GESTIÓN CON WSMF .....	17
FIGURA 2.5 OBJETOS WSFM-WSM.....	31
FIGURA 3.1 ARQUITECTURA DATA WAREHOUSE .....	40
FIGURA 3.2 DATA WAREHOUSE ORIENTADO AL TEMA.....	42
FIGURA 3.3 DATA WAREHOUSE NO VOLATILE .....	44
FIGURA 3.4 DATA WAREHOUSE INTEGRACIÓN .....	44
FIGURA 3.5 FASES DE CONSTRUCCIÓN DE UN DW .....	45
FIGURA 3.6 APLICACIÓN DE CONSULTA PERSONALIZADA .....	47
FIGURA 3.7 PROCESAMIENTO ANALÍTICO ONLINE .....	48
FIGURA 3.8 TECNOLOGÍAS BUSINESS INTELLIGENCE.....	49
FIGURA 3.9 DATA MINING Y KNOWLEDGE DISCOVERY IN DATABASES.....	50
FIGURA 3.10 PROCESO DE DESCUBRIMIENTO DE CONOCIMIENTO EN BASES DE DATOS KDD .....	52
FIGURA 3.11 FASES DE DATA MINING .....	55
FIGURA 4.1 MÓDULOS FUNCIONALES DE LA APLICACIÓN .....	59
FIGURA 5.1 ARQUITECTURA DE LA APLICACIÓN .....	62
FIGURA 4.2 INTERFAZ ADMINISTRADOR DE LA APLICACIÓN .....	65
FIGURA 4.3 DIAGRAMA DE CASOS DE USO INICIALES – NIVEL 1 .....	66
FIGURA 4.4 MÉTODO PARA EL DISEÑO Y MODELADO DE UN DW .....	78
FIGURA 4.5 DIMENSIÓN DISPOSITIVOS .....	81
FIGURA 4.6 DIMENSIÓN TIEMPO .....	81
FIGURA 4.7 DIMENSIÓN UBICACIÓN.....	81
FIGURA 4.8 DIMENSIÓN TRÁFICO .....	82
FIGURA 4.9 RELACIÓN DIMENSIONAL SLA 1 .....	82
FIGURA 4.10 RELACIÓN DIMENSIONAL SLA 2 .....	83
FIGURA 4.11 CUBOS .....	85
FIGURA 4.12 FRAGMENTACIÓN DEL CUBO TRÁFICOH.....	85
FIGURA 4.13 FRAGMENTACIÓN DEL CUBO TRÁFICO D.....	86
FIGURA 4.14 MAPEO DEL FRAGMENTO ÚNICO DE LA DIMENSIÓN DISPOSITIVOS .....	87
FIGURA 4.15 MAPEO DEL FRAGMENTO ÚNICO DE LA DIMENSIÓN TIEMPO .....	88
FIGURA 4.16 MAPEO DEL FRAGMENTO ÚNICO DE LA DIMENSIÓN UBICACIÓN .....	88
FIGURA 4.17 MAPEO DEL CUBO TRÁFICOH .....	89
FIGURA 4.18 MAPEO DEL CUBO TRÁFICO D .....	89
FIGURA 4.19 MODELO FÍSICO DEL DW .....	93
FIGURA 4.20 ESQUEMA FINAL DEL DATA WAREHOUSE.....	95
FIGURA 4.21 PROCESO DE DESCUBRIR EL CONOCIMIENTO EN BASES DE DATOS KDD.....	96

## LISTADO DE TABLAS

TABLA 3.1 DATA WAREHOUSE VS BASE DE DATOS OPERACIONAL .....	42
TABLA 3.2 USOS EN DATA WAREHOUSE VS USOS DE BASES DE DATOS OPERACIONALES.....	46
TABLA 3.3 CLASIFICACIÓN DE LAS TÉCNICAS DE MINERÍA DE DATOS .....	53
TABLA 4.1 DEFINICIÓN DE LOS ATRIBUTOS DE LA TABLA DE HECHOS SOBRE SLA.....	94
TABLA 4.2 DEFINICIÓN DE LOS ATRIBUTOS DE LA DIMENSIÓN DISPOSITIVOS.....	94
TABLA 4.3 DEFINICIÓN DE LOS ATRIBUTOS DE LA DIMENSIÓN TIEMPO.....	94
TABLA 4.4 DEFINICIÓN DE LOS ATRIBUTOS DE LA DIMENSIÓN UBICACIÓN .....	95

## INTRODUCCIÓN

Los actuales estándares para la gestión SNMP[1] (Simple Network Management Protocol), CIM[2] (Common Information Model), entre otros y los marcos de referencia<sup>1</sup> (frameworks) han evolucionado de una manera similar pero cada uno por su lado, hasta tal punto, que permite tener un entorno administrativo robusto de funcionalidades, pero completamente no inter operable entre las aplicaciones de gestión. Esto representa un gran problema para las necesidades del mundo actual, donde se requiere alta compatibilidad, minimización de “islas” tecnológicas y la tendencia hacia un mundo globalizado.

Otro problema detectado en los actuales estándares de gestión, es que éstos son de un nivel demasiado bajo como para permitirle a las aplicaciones, que gestionan los recursos, una interacción flexible y coordinada. La respuesta a la falta de flexibilidad por parte de SNMP, CIM y otros estándares, radica que su función es la recolección de datos, no el escribir aplicaciones de gestión necesarias para una infraestructura adaptable.

La diversidad de redes, los avances en los protocolos de transporte y la facilidad cada vez mayor en adquirir dispositivos de red, hacen que el volumen de información que se intercambia entre diferentes puntos de una red se incremente en forma exponencial. Esos datos pueden ser manipulados y almacenados de tal forma que sean aprovechados, debido a que proporcionan una valiosa información tanto para el soporte de una futura toma de decisiones sobre el dimensionamiento y comportamiento de una red, como para controlar el desempeño y utilización de los equipos que la conforman.

El análisis de los datos que proporcionan las redes han incluido informes de tráfico (por ejemplo en archivos de registro log), la mayoría de los cuales proporcionan

---

<sup>1</sup> Un marco de referencia o framework es la extensión de un lenguaje mediante una o más jerarquías de clases que implementan una funcionalidad y que (opcionalmente) pueden ser extendidas.

cuentas acumulativas de la actividad del recurso gestionado (equipo de red), pero no representan el rendimiento de los equipos o los perfiles de los usuarios de dichos equipos. Gran parte del software actual desarrollado para el análisis de tráfico, ofrece informes predefinidos acerca de la actividad del recurso, basados en el análisis de los archivos log. Esto limita básicamente el enfoque de estas herramientas a estadísticas acerca de los nombres de dominios, direcciones IP, navegadores, y otros protocolos específicos de actividad máquina a máquina.

Para una gestión más eficiente de redes de datos se requiere de otros mecanismos de análisis, además de los basados en archivos logs, y una buena alternativa son las tecnologías de Data Warehouse y Data Mining. Con la utilización de Data Warehouse (repositorio de datos), que es una tecnología de almacenamiento de datos multidimensional, se cuenta con un repositorio robusto de información relacionada al comportamiento de los equipos de red y al tráfico observado en los mismos. Con Data Mining o Minería de Datos se pueden establecer perfiles sobre los diferentes tipos de usuarios del sistema y lineamientos exactos de comportamiento de los recursos. De esta forma, es posible tomar decisiones acertadas en cuanto a la creación o eliminación de servicios, la optimización de la red y sus diferentes equipos.

En la actualidad, el enfoque de la gestión de red, ha estado dirigido a una gestión por funcionalidad (gestión de fallos, gestión de desempeño, y otros), recurso por recurso (gestión de red, gestión de sistemas, gestión de bases de datos, etc.) o igualmente fabricante por fabricante (donde se tienen una gran variedad de aplicaciones específicas para cada uno de ellos), pero este enfoque lo han dado los estándares de gestión que existen actualmente, como SNMP, CIM, JMX[3] (Java Management Extensions) Y WMI[4] (Windows Management Instrumentation), por nombrar unos pocos. Sin embargo, mientras éstos estándares son eficaces manejando los recursos específicos – tales como equipos de red - de una empresa, no están dirigidos a la gestión de los procesos de negocio siguiendo unos acuerdos de nivel de servicio. Se puede decir entonces, que no hay un estándar disponible que exponga las relaciones que existen entre los recursos, los servicios relacionados y los usuarios de éstos. Para eso se usa WSMF (Web Services Management Framework) que permite ofrecer un valor agregado a la gestión de redes.

En este trabajo de grado se propone construir una *Plataforma de Acuerdos de Nivel de Servicio Basada en Minería de Datos y Servicios Web*, para la Red de Datos de la Universidad del Cauca, el cual tiene el objetivo de Construir una Plataforma de

Acuerdos de Nivel de Servicio que aplique el concepto de Data Warehouse y Data Mining, y esté soportada en una arquitectura lógica para la gestión de recursos utilizando Servicios Web - WSMF

## **Capítulo 1 VALORACIÓN DE LA RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA**

La Universidad del Cauca como generadora y socializadora de tecnología e investigación está en el deber de proveer a su comunidad de los mecanismos necesarios para potenciar su impacto en el medio a través del conocimiento y el acceso a la información. Es así indispensable tener acceso a Internet para convertirla en una fuente importante de información y a la vez en medio de comunicación.

Los servicios de red que surgen en este sentido, como el correo electrónico e Internet constituyen hoy en día el eje fundamental en la comunicación de las organizaciones, convirtiéndose en un componente crítico al que es pertinente invertir esfuerzos en pro de su mantenimiento y mejora.

### **1.1 Introducción**

La Red de Datos de la Universidad del Cauca cuenta actualmente con tres enlaces hacia Internet con diferentes proveedores (Telecom, ETB y Orbitel), un backbone que interconecta los diferentes edificios que conforman la universidad e innumerables conexiones de Intranet, haciéndola una red de datos de grandes proporciones que le presta el soporte a los servicios ofrecidos por la Universidad a sus miles de usuarios. La figura 1.1 detalla la infraestructura actual de la Red de Datos de la Universidad del Cauca, todos los edificios y las conexiones entre ellos.

Dentro del marco de la infraestructura de la red, otro aspecto importante son los servidores Proxy. Los Proxys permiten a los usuarios de la Intranet que no cuentan con direcciones IP reales poder tener acceso a Internet. A través de estos servidores circula la mayor cantidad de tráfico que produce la comunidad universitaria. El servicio de Proxy tiene la característica de ser de alta disponibilidad, ya que si un servidor Proxy falla, otro servidor Proxy toma a los usuarios que están saliendo por él, logrando con ello que las personas puedan navegar en cualquier momento. Un esquema simplificado de la arquitectura de red para este servicio, se presenta en la figura 1.2.

Para el buen funcionamiento de la red los administradores necesitan monitorear, entre otras cosas: la utilización del ancho de banda, el estado de funcionamiento de los enlaces, la detección de cuellos de botella, detectar y solventar problemas con el cableado, administrar la información de enrutamiento entre máquinas, etc. Además el monitoreo de la red es también un buen punto desde el que comenzar el estudio de los problemas de seguridad.

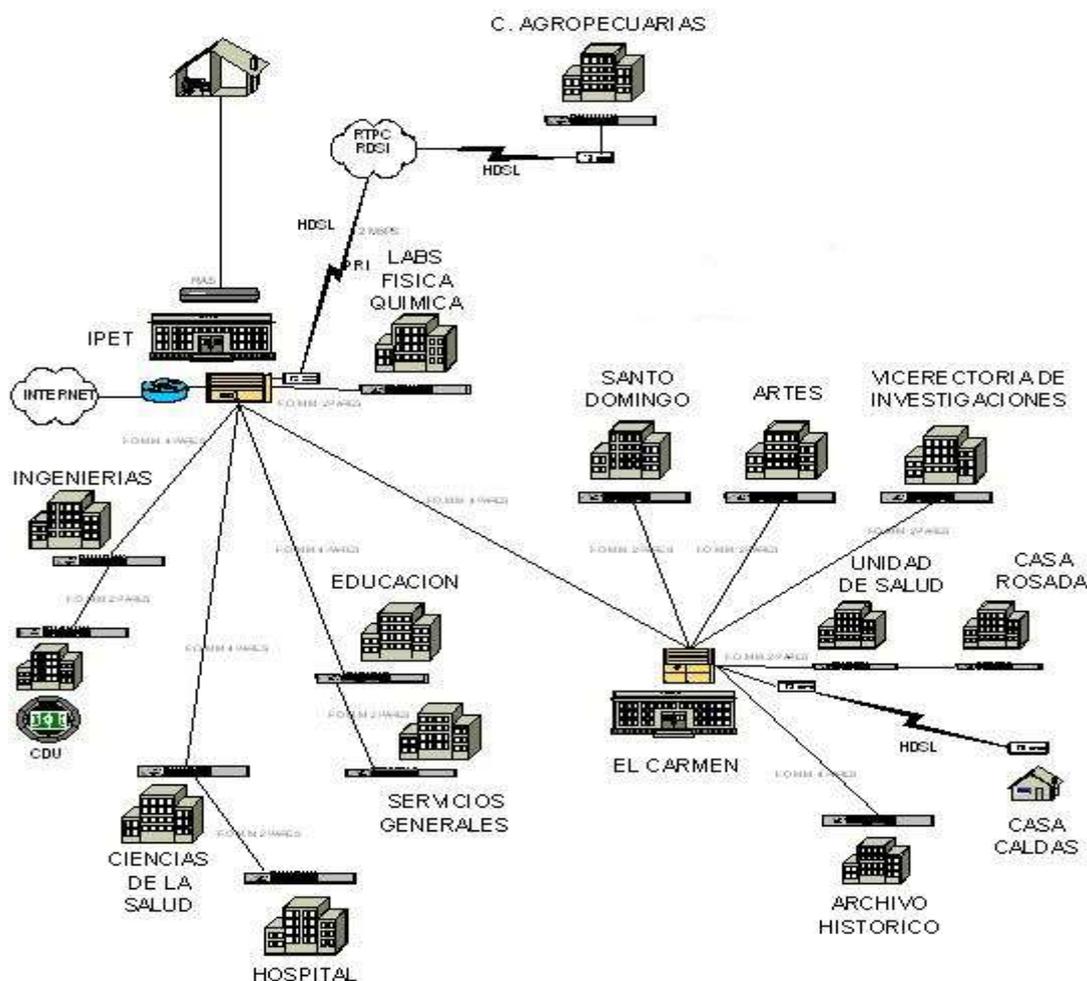


Figura 1.1 Red de Datos de la Universidad del Cauca

## 1.2 Área de Servidores y Servicios Internet

El área de Servidores y Servicios de Internet es la encargada de administrar los recursos de red de la Universidad del Cauca que soportan tanto el acceso a Internet como los servidores de la red interna, con el fin de ofrecer a la comunidad universitaria una serie de servicios dentro de los cuales, los de más relevancia hasta el momento son el de correo electrónico y la Web.

Las funciones del área involucran tres procesos generales, la gestión de servicios, la operación y mantenimiento, y el proceso de investigación en nuevos servicios. Teniendo que cumplir con funciones de operación, mantenimiento y realizando las actividades concernientes a la administración de los servidores y de los equipos de comunicaciones de la red de datos. Las principales tareas de administración son:

- Configuración de los servicios instalados en las máquinas servidoras de acuerdo a los lineamientos establecidos para cada uno, donde una gran parte del esfuerzo está dedicada a la búsqueda del mejoramiento en el rendimiento de los servidores.
- Administración completa del sistema, lo que involucra la realización de cambios sobre la marcha y la solución de problemas que se escapan del alcance de los procesos de gestión.
- Operación cotidiana de los servidores, ofreciendo respuestas a necesidades de los usuarios y de las otras áreas de la Red de Datos.
- Actualización permanente del software servidor para garantizar niveles elevados de confiabilidad en la robustez de los sistemas implementados

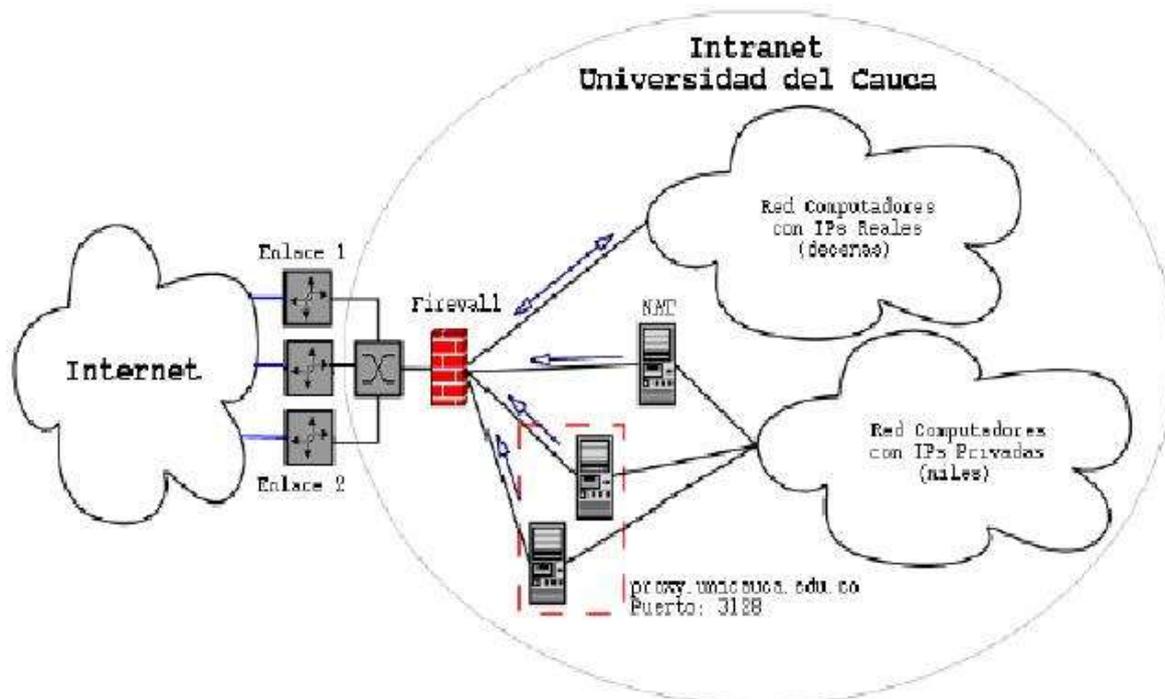


Figura 1.2 Arquitectura de red para el servicio de Proxy

Las consultas SNMP<sup>2</sup> permiten acceder a cualquier dato del equipo gestionado que exista en su árbol MIB (Management Information Base), información jerárquicamente organizada y gestionada por agentes. Para llevar a cabo todo este proceso de consultas la Red de Datos de la Universidad del Cauca optó por utilizar la herramienta de código libre MRTG (Multi Routing Traffic Grapher). Éste programa de libre distribución permite, cubrir las principales áreas funcionales en que se divide la gestión de red, definidas en su día por la ISO (International Organization for Standardization): Gestión de fallos, Gestión de configuración, Gestión de incidencias, Gestión de rendimiento y Gestión de seguridad

MRTG es un sistema desarrollado principalmente en Perl y que se caracteriza por su alto rendimiento y flexibilidad. Presenta gráficas, ver figura 1.3, sobre el tráfico de la red en diferentes periodos de tiempo. Toda esta información es fundamental a la hora de tomar decisiones, y ahí radica su importancia.

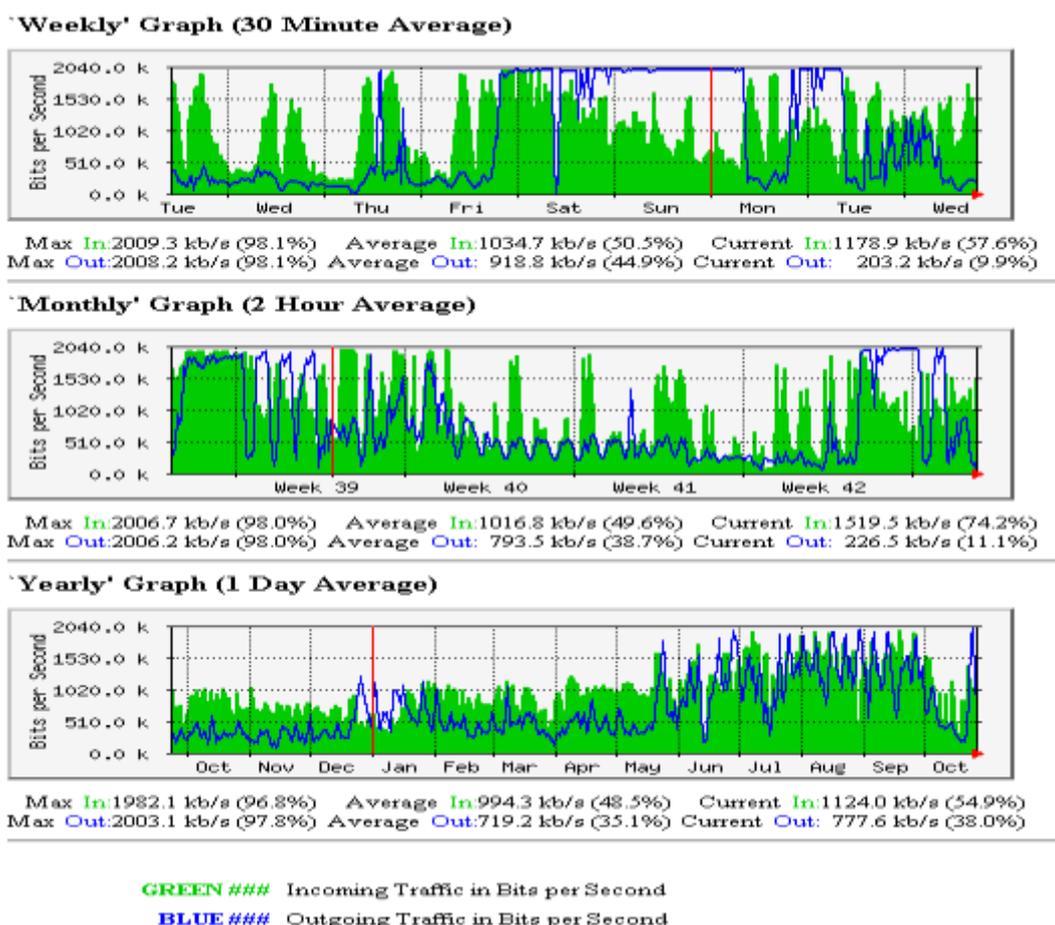


Figura 1.3 Gráficas MRTG

<sup>2</sup> Para información sobre SNMP y MIB ver el Anexo C

## Capítulo 2 GESTIÓN BASADA EN SERVICIOS WEB

La interfaz Web, tras su rápido despliegue en el mundo Internet, se ha revelado como paradigma de interfaz de usuario, gracias a sus características que la hacen ser amigable, intuitiva, independiente de arquitectura y con una curva de aprendizaje rápida. Esta tecnología suele estar articulada en el uso de lenguajes como Java y mecanismos de comunicación distribuida tales como DCOM[5] (Distributed Component Object Model, Modelo de Objetos de Componentes Distribuidos), CORBA [5] (Common Object Request Broker Architecture, Arquitectura Común de Intermediarios de Peticiones de Objetos), RMI [5] (Remote Method Invocation, Invocación de Métodos Remotos) o SOAP[6] (Simple Object Adapter Protocol, Protocolo Simple de Adaptadores de Objetos), que posibilitan que el usuario interactúe, mediante clientes ligeros o páginas generadas dinámicamente, con servidores distribuidos de forma que se aprovechan los recursos eficientemente.

WBM [7] (Web Based Management, Gestión Basada en Web) también trata de aplicar estas ideas, pero a herramientas de gestión de red. Así, arquitecturas tales como JMX, antigua JMAPI (Java Management API, Interfaz de Programación de Aplicaciones de Gestión de Java), definen los componentes que deben poseer un sistema que pretenda utilizar este nuevo paradigma a la gestión.

El DMTF [8] (Distributed Management Task Force, Grupo de Trabajo de la Gestión Distribuida) también apuesta por una gestión basada en Web usando XML y HTTP, pretendiendo la implantación de CIM como modelo de información que unifique los estándares tradicionales en la arquitectura llamada WBEM [7] (Web Based Enterprise Management, Gestión de Empresas Basada en Web). Por otro lado, OMG (Object Management Group, Grupo de Gestión de Objetos), a través del grupo de trabajo de JIDM (Joint Inter-Domain Management, Gestión Inter-Dominios Unificada) ha tratado de definir cómo se debe traducir especificaciones e interacciones CORBA con dominios de gestión tales como CMIP (Common Management Information Protocol, Protocolo Genérico de Información de Gestión) o SNMP, permitiendo la compatibilidad hacia atrás con sistemas existentes. Por otro lado, también se debe modularizar las

aplicaciones de gestión, aprovechando las posibilidades que proporcionan estas nuevas tecnologías: Es posible el desarrollo de gestores que funcionen sobre DPEs (Distributed Processing Environments, Entornos de Procesamiento Distribuido) y a los que se acceda mediante una interfaz basada en Web usando applets encapsulados en páginas HTML, o bien páginas HTML generadas dinámicamente. Los servicios de una plataforma tradicional, tales como el acceso a la pila de protocolos de gestión o un servicio de eventos, podrían ser en este caso servicios estandarizados del DPE, como ocurre con los servicios CORBA.

## **2.1 Servicios Web.**

Los servicios Web son los bloques de construcción básicos en la transición al proceso distribuido en Internet. Los estándares abiertos y el foco en la comunicación y colaboración entre las personas y aplicaciones han creado un entorno donde los Servicios Web se están convirtiendo en la plataforma para la integración de aplicaciones.

Los servicios Web son módulos de software que realizan tareas o conjuntos de tareas discretas, a los que se accede y se llama a través de una red, sobre todo la World Wide Web. Estas tareas pueden consistir en cualquier cosa: Realizar un simple cálculo con unos datos que se le envían como parámetro, acceder a una base de datos para recuperar un conjunto de registros, validar la corrección de una información o contrastarla frente a otros datos, etc.

El Servicio Web podrá ser solicitado desde otro programa informático que se ejecute en un computador conectado a la Web. Junto a la solicitud de la ejecución, se pueden enviar al computador que ofrece el servicio unos parámetros que el Servicio Web remoto tomará como base para el cálculo o la función.

El desarrollador puede crear una aplicación cliente que llama a una serie de servicios Web mediante llamadas a procedimientos remotos (RPC) o un servicio de mensajes en los que se proporciona un fragmento o la mayor parte de la lógica de la aplicación. Los servicios Web publicados se describen de forma que los desarrolladores pueden localizarlos y determinar si se ajustan a sus necesidades.

Por ejemplo, una empresa puede proporcionar a sus clientes un servicio Web por el cual se comprueba el inventario de productos antes de realizar un pedido. Otro ejemplo es el servicio de seguimiento de paquetes de la empresa de mensajería

Federal Express, mediante el cual los clientes pueden examinar el estado de sus envíos.

Los servicios Web utilizan SOAP para la carga XML, así como un sistema de transporte de tipo HTTP para transmitir los mensajes SOAP de un sitio a otro. En realidad, los mensajes SOAP son los documentos XML que se envían entre el servicio Web y la aplicación que efectúa la llamada.

Los servicios Web se pueden escribir en cualquier lenguaje, y se ejecutan en todas las plataformas. Los clientes de servicios Web también se pueden escribir en cualquier lenguaje, y se ejecutan en todas las plataformas. Así, por ejemplo, un cliente escrito en Delphi que se ejecuta en Windows puede llamar a un servicio Web escrito en Java que se ejecuta en Linux.

La aplicación que actúa como cliente debe conocer la URL del servidor remoto que ofrece el servicio, el nombre del servicio que se solicita, y los parámetros que se deben enviar junto con la llamada al servicio; estos datos se enviarán mediante HTTP. El servidor que ofrece el Servicio Web leerá los parámetros que se le han enviado, llamará a un componente o programa encargado de implementar el servicio, y los resultados que se obtengan de su ejecución serán devueltos al cliente que solicitó la ejecución del servicio. La implementación de los Servicios Web es transparente a los ojos del usuario de la aplicación. Dichos servicios son accedidos utilizando XML, generalmente sobre HTTP.

Los Servicios Web proporcionan un modo de describir sus interfaces con suficiente detalle para permitir a un usuario construir una aplicación cliente para “hablar<sup>3</sup>” con ellos. Esta descripción se proporciona generalmente en un documento XML que responde al nombre de documento WSDL (Web Services Description Language) que es un dialecto de XML que contiene información acerca de la interfaz, semántica y administración de una llamada a un servicio Web.

Los Servicios Web se registran de modo que los potenciales usuarios puedan encontrarlos. Esto se realiza mediante UDDI (Universal Discovery Description and Integration) que es un protocolo usado para describir los componentes disponibles de los servicios Web. Este estándar permite a las empresas registrarse en un tipo de

---

<sup>3</sup> Para que la aplicación cliente pueda utilizar los recursos que proporcionan los Servicios Web

directorio sección amarilla de Internet que les ayuda anunciar sus servicios, de tal forma que las compañías se puedan encontrarse unas a otras y realizar transacciones en la Web.

Una de las ventajas principales de la arquitectura de servicios Web es que permite a los programas escritos en diferentes lenguajes sobre diferentes plataformas comunicarse entre sí de un modo basado en estándares.

### 2.1.1 Arquitectura

La arquitectura de los servicios Web permite desarrollar servicios que encapsulan todos los niveles de funcionalidad empresarial. Esta arquitectura también permite la combinación de varios servicios Web con el fin de crear nuevas funciones.

La arquitectura de servicios Web tiene tres cometidos: proporciona datos, los solicita y hace de intermediario. Como proveedor, crea el servicio Web y lo pone a disposición de los clientes que desean utilizarlo. Realizan la solicitud las aplicaciones clientes que consumen el servicio Web. Este servicio solicitado también puede ser un cliente de otros servicios Web. El intermediario, como un registro de servicio, permite la interacción entre la aplicación cliente y el proveedor.

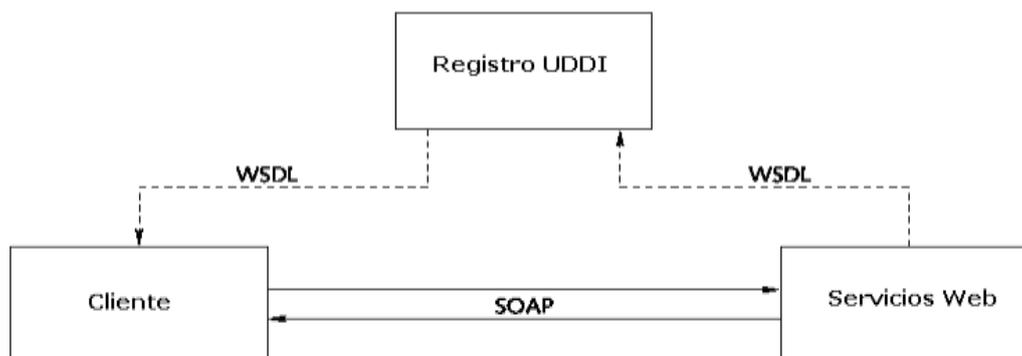
Estos tres cometidos interactúan por medio de las operaciones de publicación, búsqueda y enlace. El proveedor utiliza la interfaz de publicación del intermediario para comunicarle la existencia del servicio Web, con el fin de ponerlo a disposición de los clientes. La información publicada describe el servicio e indica dónde se encuentra. La aplicación que hace la solicitud consulta al intermediario para que busque los servicios Web publicados. Con la información obtenida del intermediario, la aplicación cliente puede enlazarse al servicio Web (llamarlo).

En este diagrama se resume cómo se realiza la interacción entre proveedor, solicitante y gestor.



Figura 2.1 Operaciones y cometidos de los servicios Web

La arquitectura de los Servicios Web se basa en interacciones entre 3 roles: proveedor del servicio, registro del servicio y usuario del servicio. Las interacciones involucran operaciones de publicación, descubrimiento y conexión (enlace). Juntos estos roles y operaciones actúan sobre los objetos de un Servicio Web; el módulo Software y su descripción.



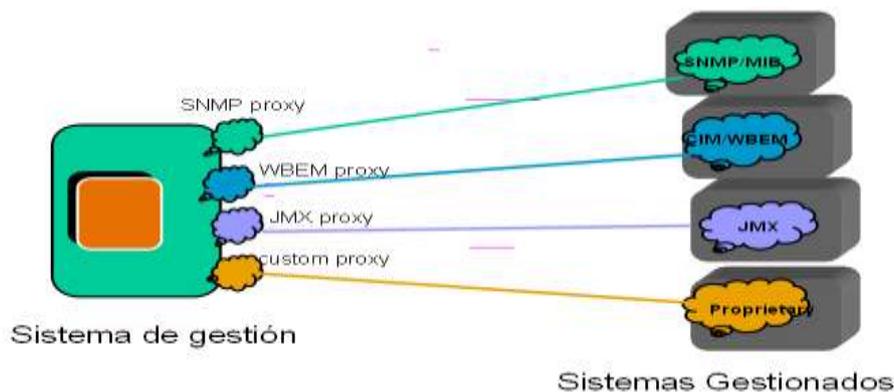
**Figura 2.2 Protocolos de Servicios Web.**

En la figura, se muestra los protocolos básicos para la creación de Servicios Web ellos son: SOAP, WSDL y UDDI. Los cuales se basan en XML.

## **2.2 WSMF (Web Services Management Framework).**

Hasta el momento la gestión ha sido orientada por funcionalidad (gestión de fallas, gestión de desempeño, etc), recurso (gestión de red, gestión de sistemas, etc), o incluso por vendedor. Las infraestructuras de gestión han evolucionado en una forma similar pero fueron desarrolladas para un subconjunto de recursos o funcionalidad de gestión, o estuvieron basadas en tecnologías cuando ellas estaban en sus etapas iniciales. En un principio, dicha aproximación era apropiada pero no permite la gestión de los procesos de negocios y los servicios subyacentes en los que ellos se basan. Además en la actualidad no hay un estándar disponible para expresar las relaciones que existen entre recursos, procesos de negocios y servicios relacionados.

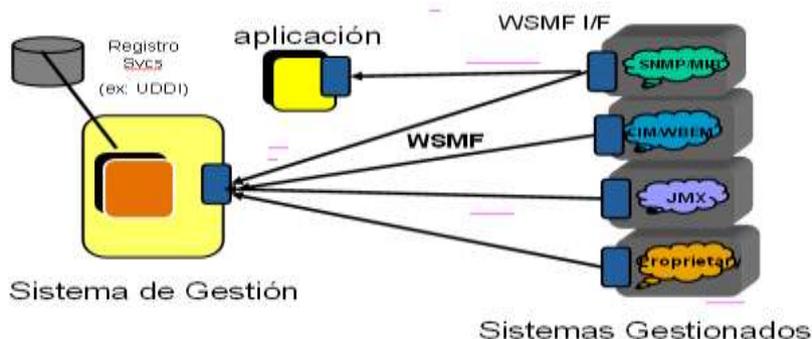
Aunque existe una variedad de plataformas de software distribuido que podría ser útil para implementar aplicaciones de gestión, las tecnologías de Servicios Web forman la plataforma de gestión del futuro, por la misma razón que otras aplicaciones distribuidas usan Servicios Web: independencia del lenguaje y la plataforma, interoperabilidad y la habilidad para exponer interfaces y esconder la implementación.



**Figura 2.3 Esquema tradicional de gestión distribuida**

WSMF [9][10] está basada en la noción de objetos gestionados y sus relaciones. Un objeto gestionado representa un recurso y expone un conjunto de interfaces de gestión a través de las cuales el recurso subyacente se puede gestionar. De la misma forma, las relaciones entre objetos gestionados representan relaciones entre recursos subyacentes.

Para soportar mejor los diferentes dominios que se pueden gestionar a través de Servicios Web, WSMF es un modelo neutral y está diseñado para aplicarse a diferentes dominios con requerimientos de gestión diferentes.



**Figura 2.4 Esquema de gestión con WSMF**

WSMF está especificada en tres diferentes documentos de especificación:

- **WSMF-Fundación:** Define la infraestructura base para gestión usando Servicios Web.
- **WS-Eventos:** Define el mecanismo de notificación de eventos basado en Servicios Web. Este mecanismo lo usa WSMF-Fundación.

- **WSMF-Gestión de Servicios Web:** Define el modelo para gestión de Servicios Web.

Actualmente los mecanismos de WSMF se aplican a la gestión de Servicios Web. En un futuro también serán aplicados a otros dominios de gestión.

WSMF proporciona una infraestructura de gestión que está basada en objetos gestionados, notificaciones de eventos y relaciones.

### 2.2.1 Objetos gestionados

Un objeto gestionado proporciona capacidades de gestión al implementar interfaces de gestión las cuales se describen usando WSDL. Por lo tanto, el objeto gestionado es un Servicio Web. Los recursos que se puedan representar por objetos gestionados con interfaces de gestión WSMF se pueden gestionar a través de WSMF. Un gestor puede usar WSMF para manejar todos los recursos de un dominio de forma uniforme.

Cuando se aplica gestión a Servicios Web, WSMF tiene en cuenta las consideraciones especiales inherentes a los Servicios Web. Por ejemplo, un Servicio Web que necesite ser gestionado a través de WSMF puede implementar directamente interfaces de gestión WSMF y evitar la necesidad de un objeto gestionado (un Servicio Web) separado.

Las capacidades descritas por WSMF incluyen:

- Descubrimiento de las definiciones WSDL de gestión.
- Descubrimiento de las relaciones y notificaciones de eventos de los objetos gestionados.
- Registro y recuperación de notificaciones de eventos.
- Monitoreo, auditoría y control de varios aspectos de los objetos gestionados utilizando las operaciones de gestión soportadas.

Los objetos gestionados se pueden implementar como parte de los recursos gestionados o se pueden proporcionar en una capa separada. La interfaz entre el objeto gestionado y su recurso es parte de la implementación del objeto gestionado y no está definido por WSMF. Sin embargo, hay una interfaz de gestión que debe ser implementada por todos los objetos gestionados, la interfaz `ManagedObjectIdentity`, la cual permite identificar a un objeto gestionado en una forma única. Los objetos

gestionados pueden escoger desde una gran variedad de otras interfaces de gestión para soportar capacidades de gestión apropiadas al recurso subyacente. Se ha definido un mecanismo de herencia de tal forma que un objeto gestionado que implementa una interfaz derivada también debe implementar la interfaz base.

### **2.2.1.1 Interfaces y colecciones de interfaces**

Las interfaces son descripciones de un conjunto de capacidades de gestión relacionadas de los objetos gestionados. Los objetos gestionados implementan interfaces para proporcionar capacidades de gestión a los recursos gestionados que ellos representan. Las interfaces de gestión son agrupadas en colecciones de interfaces para describir un grupo de capacidades de gestión de un tipo específico de objetos gestionados.

Las interfaces se pueden extender para definir nuevas interfaces con más capacidades. WSMF proporciona una colección de interfaces base que describen muchas capacidades de gestión comunes a todos los objetos gestionados. Una colección de interfaces tiene un nombre que es único en su contexto y contiene lo siguiente:

- Atributos: el conjunto de propiedades que representan información de un objeto gestionado.
- Operaciones: el conjunto de funciones que soportan la gestión de un objeto gestionado.
- Tipos de notificación: el conjunto de eventos y cambios de estado que pueden ser reportados por un objeto gestionado.

WSMF ha definido seis categorías basadas en las definidas por OSI, para agrupar atributos y operaciones en conjuntos relacionados y establecer un significado semántico común para el tipo de operaciones. Las categorías son:

- Monitoreo
- Descubrimiento
- Control
- Desempeño
- Configuración
- Seguridad

La colección de interfaces ManagedObject define las interfaces de gestión para estas categorías. Cuando se definen interfaces de gestión para un nuevo tipo de objeto gestionado, las nuevas interfaces deben heredar las interfaces definidas por ManagedObject siempre que sea posible, pero si es necesario, se permite definir nuevas interfaces. Se recomienda que las nuevas operaciones se adicionen las categorías ya definidas.

Separar todas las características de un solo objeto gestionado en varias interfaces de gestión es útil en muchos aspectos:

- Ellas agrupan subconjuntos de características de gestión en paquetes que son comúnmente aceptados en el entorno de gestión y ayudan al ISV y al cliente de gestión a usar un lenguaje común para intercambiar requerimientos y respuestas de gestión.
- La separación de características permite a un objeto gestionado proporcionar capacidades de gestión en una forma gradual.
- Las interfaces de gestión permiten a los propietarios de los recursos decidir qué información de gestión se debería proporcionar a cuál gestor, exponiendo ciertas interfaces.

Las interfaces de gestión son mapeadas a portTypes WSDL, las operaciones y el acceso de los atributos a operaciones WSDL. Por ejemplo, un atributo de solo lectura se traduce a una operación get WSDL, y un atributo de lectura y escritura se traduce en una operación get y set.

#### **2.2.1.2 Asociaciones**

Las asociaciones definen el comportamiento y las dependencias entre objetos gestionados y por lo tanto entre los recursos que ellos representan. Una relación describe el tipo de asociación entre dos objetos gestionados. Cada asociación es una instancia específica de una relación y describe una relación y los dos objetos gestionados. WSMF define algunas relaciones básicas en el namespace WSMF, pero se pueden definir otras relaciones en otros namespaces para describir asociaciones que son importantes en otros dominios de gestión.

#### **2.2.2 Eventos y notificaciones**

Un sistema de eventos es una parte integral de cualquier solución de gestión, sin embargo, actualmente no hay un estándar basado en Servicios Web ampliamente

adoptado para tal subsistema. Después de investigar varias propuestas, WSMF ha definido su propio sistema de eventos basado en Servicios Web para cubrir mejor los requerimientos. La descripción detallada de este subsistema se puede encontrar en el documento de la especificación WS-Eventos.

WS-Eventos define operaciones para suscribirse a notificaciones de eventos y la sintaxis de notificación y reglas de procesamiento para informar a uno o más suscriptores que ha ocurrido un evento. Un evento es un cambio de estado en un recurso gestionado o procesamiento de un requerimiento que puede ser comunicado a través de una notificación. Existen dos formas de obtener estas notificaciones: modo push o modo pull. En el modo push, cuando ocurre un evento, el objeto gestionado envía una notificación a los suscriptores para informarles que hubo un cambio de estado. En el modo pull, el suscriptor hace llamadas al objeto gestionado para requerir todas las notificaciones que se han producido desde la última llamada.

Por razones de eficiencia y escalabilidad, WSMF permite operaciones de suscripción y notificación a gran cantidad (bulk). En el modo push, el gestor puede suscribirse a varios tipos de notificaciones en una sola llamada. En el modo pull, a través de una sola llamada se pueden recuperar notificaciones de varios tipos.

El modo push tiende a consumir más recursos de red, mientras que el modelo pull requiere más recursos de procesamiento y memoria en los objetos gestionados. Además, el modelo pull hace posible que WSMF se pueda aplicar a escenarios donde el objeto gestionado no puede establecer conexión con el suscriptor, por ejemplo cuando está detrás de un firewall.

Las notificaciones pueden ser soportadas por un objeto gestionado, implementando las interfaces descritas en la especificación WS-Eventos, para generar eventos y permitir al gestor suscribirse a eventos, requerir eventos y recibir eventos.

### **2.2.3 WSMF-Fundación**

WSMF-Fundación fue creada para proporcionar un medio estandarizado para describir y exponer información de gestión de un recurso y para que los sistemas de gestión descubran y accedan esa información. Con WSMF, tanto los creadores y los consumidores de información de gestión se benefician de la flexibilidad que proporciona la tecnología de Servicios Web.

Esta especificación describe cómo un recurso gestionable se expone como un Servicio Web, define interfaces de gestión estándar para recursos gestionables como operaciones de Servicios Web, describe cómo se descubre un recurso gestionable, cómo se definen sus capacidades de gestión, cómo está asociado con otros recursos, y cómo extender interfaces para proporcionar capacidades de gestión de recursos en dominios específicos.

WSMF-Fundación no especifica la interacción entre un objeto gestionado y el recurso que representa, sus especificaciones solamente se aplican a la interacción entre el objeto gestionado y el gestor.

Los recursos dentro de un sistema están relacionados de diferentes formas. Estas asociaciones son importantes para proporcionar información de cómo un sistema responde cuando un recurso no funciona bien o no está disponible. WSMF-Fundación representa estas asociaciones en los objetos gestionados, lo que proporciona un método para que otros descubran estas relaciones. Las relaciones también son un medio para descubrir otros objetos y recursos gestionados en el sistema.

WSMF-Fundación define un método para que otros descubran las capacidades de gestión de un objeto gestionado. Estas capacidades son proporcionadas de forma modular para permitir que los objetos gestionados con diferentes tipos de recursos gestionados expongan diferentes interfaces de gestión. Esta modularidad también se puede usar para proporcionar control de acceso a sus capacidades de gestión.

### **2.2.3.1 Infraestructura de gestión**

WSMF-Fundación presenta una infraestructura para manejar recursos a través del uso de Servicios Web. Esto proporciona una metodología independiente de la plataforma para monitorear y controlar recursos distribuidos. La representación de gestión de un recurso es llamada un objeto gestionado, el cual expone una o más interfaces de gestión. Ellas son accedidas a través de mecanismos de Servicios Web y descritas en un documento WSDL. La especificación WSMF-Fundación proporciona definiciones para interfaces de gestión básicas.

Una interfaz de gestión está descrita por un portType. Un objeto gestionado debe exponer una o más interfaces de gestión, por lo tanto debe proporcionar una o más definiciones de portType. Las interfaces de gestión se exponen al gestor definiendo un binding WSDL para un portType de gestión y adicionando un elemento port a un

elemento de servicio en un documento WSDL. Un objeto gestionado debe soportar el binding SOAP sobre HTTP especificado en el documento WSDL asociado con la especificación. Cada interfaz de gestión debe tener su propia interfaz WSDL. Los atributos y operaciones que hacen las capacidades de una interfaz de gestión son mapeados a operaciones WSDL. Si una interfaz de gestión se expone a un gestor, todas las operaciones WSDL en la interfaz WSDL asociada se deben exponer al gestor.

Las interfaces de gestión y por lo tanto las interfaces WSDL son agrupadas en namespaces XML para describir capacidades de gestión para una clase de objeto gestionado. WSMF-Fundación describe una sola clase base de objetos gestionados. Las aplicaciones de WSMF a dominios de gestión específicos definen nuevas clases de objetos gestionados usando namespaces XML específicas a ese dominio. Se recomienda que antes de definir todas las capacidades de gestión, las aplicaciones de la WSMF-Fundación extiendan la colección de interfaces para un objeto gestionado definidas por WSMF-Fundación.

Además de agrupar interfaces WSDL en namespaces XML, a una interfaz de gestión se le asigna una categoría. Una categoría abarca un subconjunto de capacidades de gestión. WSMF-Fundación define seis categorías como se nombró anteriormente. Se recomienda que las interfaces de gestión definidas para nuevas clases de objetos gestionados se les asignen una de las categorías. Esto se puede hacer fácilmente al extender los portTypes base de objetos gestionados. Cuando sea necesario se pueden definir nuevas categorías para capacidades de gestión para una nueva clase de objeto gestionado.

Un gestor puede manejar objetos gestionados que corren sobre diferentes sistemas operativos y que estén implementados sobre diferentes plataformas. Un objeto gestionado debe retornar una respuesta, de éxito o falla, en un tiempo finito para cada requerimiento que recibe del gestor. Usando los protocolos descritos en la WSMF-Fundación, un gestor puede tener acceso, aunque sea limitado, a objetos gestionados que se encuentran detrás de un firewall. Además, un objeto gestionado exponiendo interfaces compatibles con esta especificación es gestionable por gestores implementados sobre diferentes plataformas y corriendo bajo diferentes sistemas operativos, siempre y cuando el gestor también sea compatible con esta especificación.

### Entidad de referencia (EntityReference)

Hay varias operaciones y tipos en las interfaces de gestión, los cuales se refieren a un objeto gestionado. Estas referencias a un objeto gestionado puede indicar o un objeto gestionado real u otro recurso que puede o no puede ser gestionado. Es útil referenciar una entidad no gestionada para ayudar al gestor a descubrir nuevos recursos, identificar dependencias entre recursos y localizar la causa raíz en casos de fallas.

### Descubrimiento

Hay dos aspectos importantes en el descubrimiento: primero, el gestor necesita descubrir los recursos y sus objetos gestionados. Segundo, el gestor necesita descubrir las interfaces de gestión disponibles para cada objeto gestionado.

- Descubrimiento de objetos gestionados. El gestor puede descubrir o ser informado de objetos gestionados en muchas formas, las cuales no están dentro del alcance de la especificación WSMF-Fundación. Algunas de ellas son: argumentos de línea de comando, archivos de configuración y UDDI. El método de descubrimiento de objetos gestionados especificado por WSMF-Fundación usa las relaciones que un objeto gestionado tiene con otro. Se puede obtener una vista de los objetos remotos asociados con un objeto gestionado pidiendo al objeto gestionado su conjunto actual de asociaciones.

Cada asociación identifica un objeto remoto en su campo RelatedObject. Dependiendo de las necesidades del gestor, éste puede usar este nuevo objeto gestionado para descubrir más objetos gestionados usando el mismo mecanismo de solicitar las asociaciones. Algunos objetos gestionados pueden tener una asociación con el objeto gestionado descubierto previamente, por lo tanto se necesita adicionar protección de referencias circulares cuando se siguen las asociaciones. Cuando se usan relaciones para descubrir nuevos objetos, es mejor comenzar desde un objeto gestionado que sea al menos una raíz local para la jerarquía. La razón es que no todos los objetos gestionados tendrán asociaciones recíprocas con los otros. Pedir las asociaciones actuales da una visión de un conjunto de objetos gestionados, pero en muchos casos es importante tener una vista más dinámica de cómo cambian estas asociaciones.

Para lograr esta vista dinámica de las asociaciones de los objetos gestionados, es necesario usar el sistema de eventos expuesto por los objetos gestionados, si ellos

lo soportan. Los tipos de eventos importantes para detectar cambios en las relaciones son `RelationshipAddedNotification` y `RelationshipRemoveNotification`.

- Descubrimiento de interfaces de gestión. Hay dos URLs que son importantes para un objeto gestionado. Un objeto gestionado se identifica con una URL a la WSDL de gestión concreta, la cual describe sus interfaces de gestión. Los mensajes de gestión son enviados a las URLs indicadas en la WSDL de gestión por las interfaces de gestión. Generalmente hay una URL para todas las interfaces de gestión, pero puede haber más de una. Estas URLs son llamadas puntos finales de gestión.

### Asociaciones

En los sistemas de gestión las asociaciones entre objetos gestionados son muy importantes, así como también es importante diferenciar los diferentes tipos de asociaciones, ya que los gestores pueden estar interesados en tipos específicos. Por ejemplo, una asociación de dependencia puede servir para que un gestor determine cuáles recursos se afectan cuando un recurso falla. Mientras que una asociación de duplicación, puede ayudar a determinar cuáles recursos pueden reemplazar a otro. Las asociaciones de un objeto gestionado también ayudan a descubrir nuevos objetos gestionados como se mencionó anteriormente.

Una asociación se define como dos objetos gestionados y la relación entre ellos. Una relación específica solamente el tipo de asociación sin el contexto. Por ejemplo: depende de (`dependsOn`), está duplicado por (`isDuplicatedBy`), etc. Una asociación adiciona el contexto de los participantes en la asociación. WSMF-Fundación define el tipo "Relationship" para representar una asociación. Ya que la asociación es adquirida a partir de un objeto gestionado, él es considerado implícitamente el sujeto de la asociación. Tanto la relación como el objeto de la asociación se incluyen como miembros del tipo "Relationship".

WSMF-Fundación define unos pocos tipos básicos de relaciones. Las aplicaciones de WSMF-Fundación a otros dominios de gestión, deben especificar relaciones apropiadas a ese dominio, al definir la URI para indicar cada relación y describir las semánticas asociadas con esa relación y los objetos gestionados involucrados.

WSMF-Fundación especifica solamente dos relaciones, ya que una relación es específica a los tipos de objeto gestionado y al dominio en el cual ellos existen. Las

relaciones definidas por la WSMF-Fundación son para las asociaciones entre un gestor de colección y un miembro de una colección. Las relaciones son las siguientes:

- ❖ **Manager-of-collection.** Esta relación indica la asociación que un gestor de colección comparte con cada uno de los miembros de su colección. Un gestor de colección debe tener una asociación con esta relación con cada miembro de su colección. El campo RelatedObject de Relationship debe indicar el miembro de la colección.
- ❖ **Member-of-collection.** Esta relación indica que un objeto gestionado es un miembro de una colección. Se recomienda que un objeto gestionado tenga una asociación con el campo HowRelated de Relationship para esta relación para cada colección de la cual es miembro. El campo RelatedObject de Relationship debe indicar el gestor de colección.

### Eventos

Como se nombró anteriormente, la especificación WS-Eventos describe cómo se pueden propagar los eventos en un entorno de Servicios Web usando un mecanismo push o pull. WSMF-Fundación especifica algunos tipos de eventos específicos que soportan gestión básica. Las aplicaciones de WSMF-Fundación a dominios específicos deben definir los tipos de eventos adicionales necesarios para soportar gestión en esos dominios.

WSMF-Fundación no requiere que un gestor o un objeto gestionado soporten un subsistema de eventos. Sin embargo, si alguno de ellos soporta eventos deben hacerlo de acuerdo a la especificación WS-Eventos. Un objeto gestionado puede escoger soportar el modelo de eventos push, el modelo pull o ambos, y puede considerar las interfaces WSDL de WS-Eventos como interfaces de gestión aún si estas interfaces no se marcan con el atributo “baseManagementInterface”. Un gestor debe tomar la determinación si las interfaces de eventos pertenecen a las interfaces de gestión examinando los tipos de eventos soportados por las interfaces.

Si un objeto soporta un modelo de eventos y ocurre una condición bajo la cual se genera un evento, el objeto gestionado debe tomar la acción apropiada relacionada con el evento.

### Estado

El ciclo de vida de un objeto gestionado es importante para el gestor. WSMF-Fundación introduce un atributo llamado "State" para ayudar a un gestor a monitorear un recurso a lo largo de su ciclo de vida. El atributo "State" proporciona una lista de información de estado donde los elementos sucesivos en la lista proporcionan valores más detallados del estado que los elementos previos. Los elementos en la lista se pueden pensar como estado, subestado, sub-sub-estado, etc. Un objeto gestionado debe tener al menos un elemento para representar el estado. Algunos valores para estado y sub-estado son definidos por WSMF-Fundación los cuales son disponibles para que todos los objetos gestionados los usen para describir donde están sus recursos gestionados con respecto a su ciclo de vida. Se pueden definir valores adicionales en las aplicaciones de WSMF-Fundación a otros dominios.

Todos los valores usados para State se deben definir y deben estar completamente descritos o en la especificación WSMF-Fundación o en la aplicación de WSMF-Fundación a un dominio específico. Un objeto no debe usar valores para estado que no hayan sido definidos en la especificación.

Un objeto que detecte un cambio en el estado en el recurso que representa debe reflejar este cambio de estado en su atributo State. Un objeto gestionado que soporta notificaciones debe generar una notificación StateChanged cuando modifica su atributo State.

Los siguientes valores se deben usar para el primer elemento de la lista de estados. Los siguientes elementos de la lista de estados indican sub-estados y sub-sub-estados, y todos juntos describen el estado actual del objeto gestionado.

### **2.2.3.2 Categorías de la interfaz ManagedObject**

A continuación se describen algunas de las interfaces de gestión que proporcionan la base para todos los objetos gestionados. Estas interfaces son útiles para los objetos gestionados en la mayoría de dominios de gestión. Cada interfaz representa una categoría de capacidad de gestión. Esta colección de interfaces es llamado ManagedObject. Un objeto gestionado o implementa todas las operaciones definidas para una interfaz particular o no implementa ninguna. No se requiere implementar todas las categorías de las interfaces de gestión, sin embargo, un objeto gestionado debe implementar la interfaz de gestión ManagedObjectIdentity.

#### Interfaz de gestión ManagedObjectIdentity

Todos los objetos gestionados deben implementar esta interfaz. Ella proporciona un método para encontrar otras interfaces de gestión que son expuestas por los objetos gestionados.

#### Interfaz de gestión ManagedObjectConfiguration

La interfaz de gestión ManagedObjectConfiguration es usada por un gestor para observar información acerca de un objeto gestionado el cual es relativamente estático. A menudo, la información en esta interfaz es configurada en tiempo de despliegue y no cambia durante el tiempo de vida del objeto gestionado. Además, generalmente la información es la misma cada vez que se crea el objeto gestionado y puede ser la misma para cada despliegue del objeto gestionado.

#### Interfaz de gestión ManagedObjectMonitoring

La interfaz de gestión ManagedObjectMonitoring es usada por un gestor para observar información acerca de un objeto gestionado que es propensa a cambiar.

#### Interfaz de gestión ManagedObjectDiscovery

La interfaz de gestión ManagedObjectDiscovery es usada por un gestor para determinar cómo éste objeto gestionado está relacionado con otros objetos. Esto se logra a través de las asociaciones entre objetos gestionados. Esta interfaz es usada por un gestor cuando se inicia y cuando éste determina que el conjunto de asociaciones para el objeto gestionado ha cambiado. Si un objeto gestionado no expone esta interfaz el gestor debe asumir que el objeto no tiene asociaciones con otros objetos.

#### Interfaz de gestión ManagedObjectControl

La interfaz de gestión ManagedObjectControl es usada por un gestor para modificar el estado de un objeto gestionado. La funcionalidad en esta interfaz está relacionada a la proporcionada en las otras interfaces, pero es proporcionada en una interfaz propia para permitir a un objeto gestionado exponer acceso a la información a un conjunto amplio de gestores y exponer modificación de esta información a un conjunto más pequeño de gestores.

#### Interfaz de gestión ManagedObjectCollection

Hay casos dónde un gestor quiere realizar la misma acción sobre varios objetos gestionados. La interfaz ManagedObjectCollection permite a un objeto gestionado actuar como proxy para varios objetos gestionados. Una colección es un conjunto de

objetos gestionados que pueden ser accedidos a través de una interfaz de gestión ManagedObjectCollection implementada por un objeto gestionado. Un gestor de colección es un objeto gestionado que implementa la interfaz ManagedObjectCollection.

#### **2.2.4 WSMF-Eventos.**

La especificación WSMF-Eventos define una sintaxis XML y un conjunto de reglas de procesamiento para publicar, suscribir, producir y consumir eventos de Servicios Web usando modo push o modo pull. En el modo push, el generador de la notificación del evento llama un método del consumidor del evento pasando uno o más notificaciones como parámetros. Por lo tanto el productor envía asincrónicamente notificaciones al consumidor. En el modo pull, el consumidor invoca métodos del generador de eventos para obtener las notificaciones almacenadas. WS-Eventos define lo siguiente:

- Una representación XML extensible de notificaciones de eventos.
- Un protocolo de suscripción simple entre consumidores y generadores de notificaciones.
- Mecanismos para publicar y descubrir eventos.
- Soporte para comunicación asíncrona y síncrona de notificaciones desde el generador al consumidor a través del modo push y pull.
- Una sintaxis XML simple para describir filtros de eventos.

WS-Eventos no define un mecanismo de eventos de propósito general, sino que define un mecanismo adecuado para la infraestructura de Servicios Web.

WS-Eventos se basa en la infraestructura de Servicios Web existente. Usa WSDL y las especificaciones del Esquema XML para describir interfaces de gestión y estructura de mensajes.

#### **2.2.5 WSMF-Gestión de Servicios Web.**

Actualmente no hay una aproximación estándar para gestionar Servicios Web. Lo que los vendedores pueden ofrecer es instrumentación en los puntos finales SOAP e intermediarios (servidores SOAP, servidores UDDI, etc). Sin embargo, esta vista de gestión es incompleta y le falta información del estado del Servicio Web mientras se esta ejecutando y los mensajes están viajando entre varios puntos finales. Para obtener ésta información, los Servicios Web necesitan ser mucho más gestionables.

WSMF-WSM es una aplicación de la especificación WSMF-Fundación para la gestión de Servicios Web. Contiene un modelo definido para este propósito ya que

actualmente no hay un modelo disponible que satisfaga la gestión de Servicios Web. También contiene otras instrucciones obligatorias necesarias para lograr una gestión eficiente de Servicios Web a través del uso de WSMF. Usa los mecanismos descritos en WSMF-Fundación para describir un modelo que se pueda usar para gestionar Servicios Web. Al igual que todos los modelos expuestos por WSMF-Fundación, está compuesto por objetos gestionados, operaciones, atributos, relaciones y notificaciones. Los objetos gestionados definidos en WSMF-WSM son completamente compatibles con la especificación WSMF-Fundación. Los objetos gestionados identificados por WSMF-WSM son Servicio, WSEExecutionEnvironment, Conversation, Registry e Intermediary. La actual versión de WSMF-WSM especifica los tres primeros. Por lo tanto el modelo WSMF para gestión de Servicios Web actualmente soporta las siguientes colecciones de interfaces:

- Servicio: una descripción de las capacidades de gestión de un Servicio Web.
- WSEE (Web Services Execution Environments): una descripción de las capacidades de gestión de un entorno de ejecución de Servicios Web.
- Conversación: una descripción de las capacidades de gestión de un recurso conversación. Una conversación es una vista de servicio del estado asociado con un conjunto de mensajes relacionados.

En la Figura 2.5, se presentan los tres tipos de objetos definidos en WSMF-WSM. En ella se tienen dos entornos de ejecución de Servicios Web. WSEE1 contiene dos Servicios Web, Servicio1 y Servicio2. WSEE2 contiene otros dos Servicios Web, Servicio3 y Servicio4. El Servicio1 depende del Servicio2 para funcionar y el Servicio2 depende del Servicio3. Un gestor puede descubrir las dependencias que afectan al Servicio1 pidiéndole los Servicios Web de los que dependen y recursivamente pedir a estos Servicios Web los servicios de los que ellos dependen.

El Servicio1 contiene dos conversaciones, Conv1 y Conv2, lo que significa que el Servicio1 está actualmente involucrado en dos interacciones con otros Servicios Web. Una de estas interacciones involucra el Servicio3. El objeto gestionado de conversación que representa la vista del Servicio1 de esta interacción es Conv1. El objeto gestionado de conversación que representa la vista del Servicio3 de esta interacción es Conv4. Otra interacción involucra el Servicio3, el Servicio4 y el Servicio5. Las vistas individuales de los Servicios Web de esta interacción están representadas por Conv3 (para el Servicio2), Conv5 (para el Servicio3) y Conv6 (para el Servicio4). Este ejemplo ilustra que un servicio puede tener varias conversaciones

simultáneas (como en el caso del Servicio3) y que los Servicios Web pueden tener conversaciones con los Servicios Web que están o no están contenidos en el mismo entorno de ejecución.

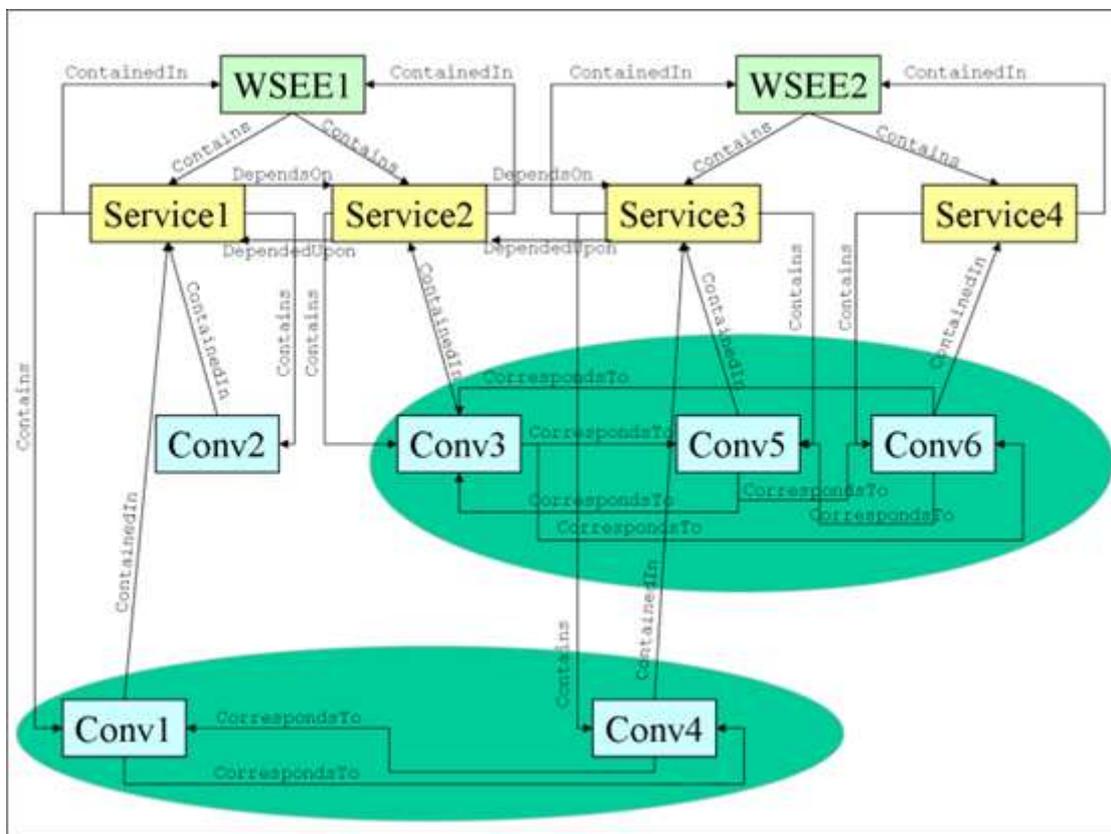


Figura 2.5 Objetos WSFM-WSM

Los objetos gestionados tienen atributos asociados a ellos (como su estado o el número de conversaciones que tiene), operaciones (que permiten monitorearlos y controlarlos) y ellos emiten notificaciones a las que los gestores pueden suscribirse.

### **Objeto gestionado del Servicio Web- Service**

El objeto gestionado Service se usa para gestionar un Servicio Web Simple.

Algunos ejemplos de las capacidades de gestión del objeto gestionado Service permiten al gestor recuperar el contenedor del Servicio Web (si hay alguno definido), las conversaciones en las que este Servicio Web está participando e información de desempeño acerca del Servicio Web

Es frecuente tener un objeto gestionado Service que implemente la interfaz de gestión Collection para permitir gestión centralizada de los objetos gestionados Conversation contenidos en Service, pero esto no es obligatorio.

### **2.3 WSDM (Web Services Distributed Management).**

La creación y adopción de estándares ayudan a mejorar la interoperabilidad. Dentro del área de gestión, la especificación de WSDM [10], actualmente desarrollada por OASIS, es un acercamiento a la obtención de interoperabilidad para la gestión de entornos de computación distribuida. WSDM esencialmente define un protocolo para la interoperabilidad de información de gestión y capacidades utilizando Servicios Web.

En pocas palabras, la especificación de WSDM define a WSDL para exponer la información de gestión y sus capacidades. Define un estándar para la gestión de Servicios Web, así como define un estándar para la gestión utilizando Servicios Web. Considerando el hecho que WSDM utiliza Servicios Web para exponer información de gestión, las aplicaciones con el dominio de gestión ganan todos los beneficios asociados con ese tipo de paradigma distribuido.

La especificación de WSDM presenta interfaces de aplicación a través de un stack técnico completo dentro de la empresa, la cual proporciona una oportunidad única de utilizar la información de gestión de diferentes maneras. No hay duda que dicha información de gestión continuará siendo utilizada concretamente para gestionar niveles de servicio y disponibilidad.

WSDM no sólo define como podemos exponer información de gestión en un modo basado en estándares, sino también como formular las bases para la interoperabilidad de la información de gestión. WSDM puede ser usado para monitorear procesos de negocio dentro de una empresa y relacionarlos entre sí.

### **2.4 WSMN (Web Services Management Network)**

WSMN [10] es una arquitectura de gestión estrechamente relacionada con la gestión de servicios. Los servicios de Internet son implementados utilizando la tecnología de Servicios de Web (SOAP, XML, WSDL) y WSMN está basada en esa tecnología. El concepto crítico de WSMN son los SLA's. Si los SLA's están explícitamente definidos se deben adaptar para poder ser manejables.

WSMN es una red de cubierta lógica para la gestión de SLA's entre servicios, constituidas de comunicaciones intermedias. Esta red esta conformada por unos intermediarios que hacen las funciones de proxy entre el servicio (al cual esta conectado el intermediario) y el mundo exterior. Los aspectos mas críticos e interesantes de WSMN es basar toda la gestión en SLA's, y los protocolos para la colaboración de los intermediarios.

## **2.5 Acuerdos de nivel de servicio SLA**

Un Acuerdo de nivel de servicio SLA, es un contrato entre un Proveedor de servicios de aplicación ASP y un usuario final el cual estipula y confía que el ASP cumpla con un nivel de servicio requerido. Un SLA debe contener un nivel de servicio especificado, opciones de soporte, provisiones de la aplicación o sanciones por los servicios no prestados, un nivel garantizado de desempeño del sistema, un nivel específico de soporte al cliente y que software o hardware debe ser proporcionado y con que fin.

El acuerdo de nivel de servicio es el compromiso de mantener un nivel consistente de transferencia de datos sobre una red. Cada Proveedor de servicios de Internet ISP típicamente tiene un SLA que estipula el compromiso de disponibilidad de datos que la ISP proporcionará a sus clientes. Usualmente los SLA's son solamente dados a clientes del negocio que pagan más por sus conexiones que a los simples usuarios que se conectan desde sus casas. Así las conexiones comerciales son típicamente más confiables, y por lo tanto más costosas. Los SLA's son muy importantes para las compañías que pueden perder millones cuando sus clientes no pueden acceder a sus Servicios Web.

En resumen, un SLA es un contrato entre un proveedor de red y un cliente en el que se especifican, usualmente en términos medibles, que servicios está en capacidad de ofrecer. Muchos proveedores de servicios de Internet (ISP) proporcionan a sus clientes SLA's. Recientemente, los departamentos de IT de la mayoría de las empresas han adoptado la idea de escribir acuerdos de nivel de servicio tales que los servicios que prestan a sus clientes (usuarios en otras dependencias dentro de la empresa) puedan ser medidos, justificados, y en la mayoría de los casos comparados con sus proveedores de servicios externos.

Este modelo no ha de estar relacionado necesariamente con la contratación de servicios a terceras partes, sino que puede implantarse a nivel interno, transformando

una determinada unidad de negocio en centro de servicios que provea a la propia compañía.

Algunas métricas que los SLA's pueden incluir específicamente son:

- Porcentaje de tiempo los servicios pueden estar disponibles.
- El número de usuarios que pueden ser atendidos simultáneamente.
- Patrones específicos de rendimiento con los que los actuales índices de desempeño serán comparados periódicamente.
- Agenda para notificación en avance de los cambios en la red que pueden afectar a los usuarios.
- Ayuda en línea para diferentes tipos de problemas.
- Uso de estadísticas que deberán ser proporcionadas.
- Tipo de servicio.
- Soporte a clientes y asistencia.
- Provisiones para seguridad y datos.
- Garantías del sistema y tiempos de respuesta.
- Disponibilidad del sistema.
- Conectividad.
- Multas por caída del sistema.

Estos puntos son importantísimos a la hora de formalizar de forma contractual una operación.

### **2.5.1 Implantación de acuerdos de nivel de servicio con proveedores.**

Para implantar con éxito un SLA han de tenerse en cuenta un serie de factores clave, de los que va a depender en gran medida la obtención de los resultados deseados:

- **Aspectos críticos.**  
Los aspectos más críticos, son la definición de procedimientos estándares y los mecanismos de evaluación y seguimiento.
- **En la implantación de un SLA se sigue una serie de puntos:**
  1. Definición de Objetivos: mejora de la eficacia, reducción de costes, formalización de la relación.
  2. Identificar expectativas: qué es lo que espera la organización de este acuerdo.

3. Adecuada planificación temporal.
4. Optimización/rediseño de procesos (revisar los procesos si el SLA no asegura ningún cambio o como mínimo formalizarlos).

- **Errores más frecuentes en la implantación.**

- Definir niveles de servicio inalcanzables
- Regulación excesiva.
- Error en la definición de prioridades.
- Complejidad técnica.
- Irrelevancia (si un SLA no tiene ningún efecto sobre el cliente, el objetivo no tiene sentido).

### **2.5.2 ASP (Proveedor de servicios de aplicación).**

Un proveedor de servicios de aplicación es una entidad que gestiona y distribuye servicios basados en software y soluciones a los clientes a través de una red MAN de un centro de datos.

En esencia, los proveedores de servicio de aplicación son el modo en el que las compañías obtienen muchos o casi todos los aspectos de sus necesidades de Tecnologías de Información IT. Los ASP's pueden ser empresas comerciales que abastecen a los clientes proporcionándoles servicios y soporte tecnológico.

Los Proveedores de servicios de aplicación pueden ser clasificados en cinco categorías:

- Enterprise ASPs (ASP's Empresariales) – Ofrecen aplicaciones de negocio high-end.
- Local/Regional ASPs (ASP's regionales o locales) —Provee una amplia variedad de servicios de aplicación para negocios pequeños en áreas locales.
- Specialist ASPs (ASP's especializados) – Proporciona aplicaciones para necesidades específicas, tales como servicios para un sitio Web o recursos humanos.
- Vertical Market ASPs (ASP's de mercado vertical) – Proporciona soporte a una industria del mercado en específico.
- Volume Business ASPs (ASP's de negocio de volumen) – Provee negocios de pequeño y mediano tamaño en general con servicios de aplicación preempaquetados en volúmenes.

### **2.5.3 Usuario Final**

El usuario final es un individuo que usa el producto después de que éste ha sido completamente desarrollado y comercializado. Es un término útil ya que distingue dos clases de usuarios, los que utilizan el producto en modo de prueba aceptando la aparición de inconsistencias y fallos y los que esperan niveles de calidad el mismo de acuerdo al propósito para el que fue desarrollado.

### **2.5.4 Proveedor de servicios de Internet**

Un ISP o Proveedor de servicios de Internet es una compañía que proporciona acceso a Internet a individuos o a empresas y a otros servicios relacionados tales como la construcción de sitios Web y hosting virtual. Un ISP tiene el equipo y el acceso de línea de telecomunicaciones requerido para tener un punto de presencia sobre Internet para el área geográfica atendida. ISP's más grandes tienen sus propias líneas dedicadas de alta velocidad y de este modo son menos dependientes de los proveedores de telecomunicaciones y pueden proporcionar mejor servicio a sus clientes. En ocasiones un ISP es también relacionado con los proveedores de acceso a Internet IAP (Internet Access Provider).

### **2.5.5 IS**

Sistema de información IS (Information System) es la colección de recursos técnicos y humanos que proporciona el almacenamiento, procesamiento, distribución y comunicación para la información requerida por todas y cada una de las partes de una empresa. Una forma especial de IS es el MIS (Management Information System) - Sistema de Información de Gestión, el cual provee la información necesaria para gestionar una empresa.

### **2.5.6 Métrica**

En desarrollo de software, una métrica es la medida de una característica particular de la eficiencia o desempeño de un programa. Similarmente en enrutamiento de redes, una métrica es una medida usada para calcular el siguiente host al cual direccional un paquete.

### **2.5.7 Conceptos acerca de los SLA**

Utilizar servicios ofrecidos por otra empresa puede ser crítico para sobrellevar una dependencia IT (Information Technology) – Tecnología de la Información efectiva. Particularmente en una economía altamente tecnológica. Es por eso que las empresas

que ofrecen la provisión de servicios IT están enfatizando su atención en el desarrollo de SLA's [11].

A continuación mencionaremos diez falsas creencias a tener en cuenta que se tienen acerca de los SLA's:

**1. Los SLA son inútiles.** Un SLA es el contrato que una sociedad de clientes sella con el proveedor de servicios IT. Es el documento que lista las obligaciones del vendedor y señala las sanciones y multas que el proveedor debe afrontar si incumple en la prestación de los servicios.

**2. SLA simplemente se ocupa de los servicios proporcionados.** Aunque los servicios proporcionados son un aspecto muy importante de un SLA, también se tienen en cuenta niveles de rendimiento y desempeño y ramificaciones legales. La información que debe estar contenida dentro de un SLA incluye el objetivo del SLA, la descripción del servicio, la duración del servicio, términos de pago, condiciones de terminación del contrato, y asuntos legales tales como garantías, indemnizaciones, y limitación de la responsabilidad del proveedor de servicios de aplicación ASP.

**3. Los objetivos comerciales no deben ser incluidos dentro de un SLA.** Al escribir los objetivos comerciales y de negocio de un cliente dentro de un SLA se proporciona al vendedor de un alto grado de entendimiento de las prioridades del cliente, las cuales pueden resultar invaluable a la hora de afrontar crisis técnicas.

**4. Los servicios determinan el precio.** El factor más significativo en la determinación del precio, como se especifica en un SLA, es el nivel de desempeño del servicio.

Los clientes pagan a los vendedores de servicio de acuerdo a un criterio predeterminado de rendimiento, disponibilidad y tiempo de respuesta. Un SLA debe incluir también especificaciones que contemplen sanciones financieras, en el caso de que el vendedor no cumpla con los niveles de desempeño indicados en el SLA.

**5. Un SLA estándar del vendedor no puede ser modificado para requisitos particulares.** Muchos vendedores ofrecen SLA's estándares, y en algunos mercados objetivo sus servicios están basados en la fortaleza de su SLA. Pero casi todos los vendedores modifican sus servicios y sus SLA's con el fin de satisfacer los requerimientos de sus clientes.

**6. No existen métricas para el desempeño del servicio.** Dependiendo de la naturaleza de los servicios, los clientes pueden medir el desempeño por parámetros tales como la disponibilidad de la red o la aplicación, tiempo promedio de arranque, latencia, tiempo de respuesta de la ayuda de escritorio e incluso el tiempo durante el cual se mantiene arriba el servicio en un día. El consorcio de la industria ASP (The ASP Industry Consortium) actualmente está desarrollando un conjunto de estándares de métricas mediante las cuales los proveedores de servicio pueden ser medidos.

**7. El vendedor debe monitorear el desempeño.** Los vendedores generalmente no monitorean su propio rendimiento y desempeño, sin embargo un creciente número de vendedores de software están produciendo herramientas que supervisan el desempeño del proveedor del servicio en nombre de sus clientes.

**8. Un SLA solo aplica a lo que el vendedor muestra.** Mientras pareciese que esto es una razón lógica, los proveedores de servicio a menudo participan de una red entera de abastecedores de servicio. Esta red puede incluir funciones IT que pueden ser ofrecidas por socios externos, los cuales, permiten a otras compañías cubrir sus responsabilidades de su servicio. Para minimizar los riesgos de desempeño de una tercera persona, los clientes deben incluir una cláusula estipulando que el vendedor principal debe ser el responsable por cualquier daño ocasionado por socios externos.

**9. Si está en el SLA, está garantizado.** En el calor de la venta, los vendedores en ocasiones prometen servicios que no pueden ofrecer. Por esta razón, los clientes deben ser cuidadosos de exigir demandas en los servicios en los que el vendedor duda al ofrecer. Los clientes deben recordar también que un SLA es sólo un contrato que representa a una parte, y por lo tanto deben supervisar que los vendedores cumplan con sus obligaciones.

**10. Remediar un SLA fallido es imposible.** Muchas firmas consultoras ofrecen servicios de remediación que ayudan a los clientes y proveedores de servicio a renegociar un SLA. De hecho debido a que la prestación de servicios por socios externos se hace cada vez más popular, firmas de consultoría encuentran necesario realizar labores de remediación en los SLA's.

## **Capítulo 3 DESCUBRIMIENTO DEL CONOCIMIENTO EN BASES DE DATOS**

El proceso del descubrimiento del conocimiento en bases de datos, conocido como KDD (Knowledge Discovery in Databases) por las siglas en inglés, se ha definido como la extracción no trivial de información potencialmente útil a partir de un gran volumen de datos en el cual la información está implícita pero sin conocerla previamente. Se trata de interpretar grandes cantidades de datos y encontrar relaciones o patrones siguiendo un proceso que está dividido en varias fases, comenzando con el Data Warehouse seguido del Data Mining y terminando con la interpretación de resultados. Para conseguirlo hace falta técnicas de aprendizaje [Machine Learning], extracción de datos y bases de datos.

### **3.1 Repositorio de Datos (Data Warehouse)**

#### **3.1.1 Introducción**

Un Data Warehouse es una colección de datos en la cual se encuentra integrada la información de una organización y que se usa como soporte para el proceso de toma de decisiones gerenciales. Reunir los elementos de datos apropiados desde diversas fuentes de aplicación en un ambiente integral centralizado, simplifica el problema de acceso a la información y en consecuencia, acelera el proceso de análisis, consultas y el menor tiempo de uso de la información.

Las aplicaciones para soporte de decisiones basadas en un data warehousing, de las que podemos considerar la minería de datos, pueden hacer más práctica y fácil la explotación de datos para una mayor eficacia del negocio, que no se logra cuando se usan sólo los datos que provienen de las aplicaciones operacionales (que ayudan en la operación de la empresa en sus operaciones cotidianas), en los que la información se obtiene realizando procesos independientes y muchas veces complejos. Cuando necesitamos crear un repositorio de datos debemos extraer datos desde una o más bases de datos de aplicaciones operacionales. Los datos extraídos son transformados para eliminar inconsistencias, resumir si es necesario y luego cargarlos en el data warehouse. El proceso de transformar, crear el detalle de tiempo variante, resumir y

combinar los extractos de datos, ayudan a crear el ambiente para el acceso a la información. Este nuevo enfoque ayuda, en todos los niveles de cualquier organización, a efectuar su toma de decisiones con más responsabilidad.

### 3.1.2 Arquitectura

A fin de comprender cómo se relacionan todos los componentes involucrados en una estrategia data warehousing, es esencial tener una Arquitectura Data Warehouse [12]. La Arquitectura de un Data Warehouse viene determinada por su situación central como fuente de información para las herramientas de análisis. Presenta los siguientes componentes, ver la figura 3.1:

- Sistema ETL (*Extraction, Transformation, Load*): realiza las funciones de *extracción* de las fuentes de datos (transaccionales o externas), *transformación* (limpieza, consolidación, y demás) y la *carga* del Data Warehouse, realizando:
  - Extracción de los datos.
  - Filtrado de los datos: limpieza, consolidación, etc.
  - Carga inicial del almacén: ordenación, agregaciones, etc.
  - Refresco del almacén: operación periódica que propaga los cambios de las fuentes externas al almacén de datos
- Repositorio Propio de Datos: información relevante, metadatos.
- Interfaces y Gestores de Consulta: permiten acceder a los datos y sobre ellos se conectan herramientas más sofisticadas (Minería de Datos, OLAP - Online Analytical Processing, EIS - Executive Information Systems).
- Sistemas de Integridad y Seguridad: se encargan de un mantenimiento global, copias de seguridad, etc.

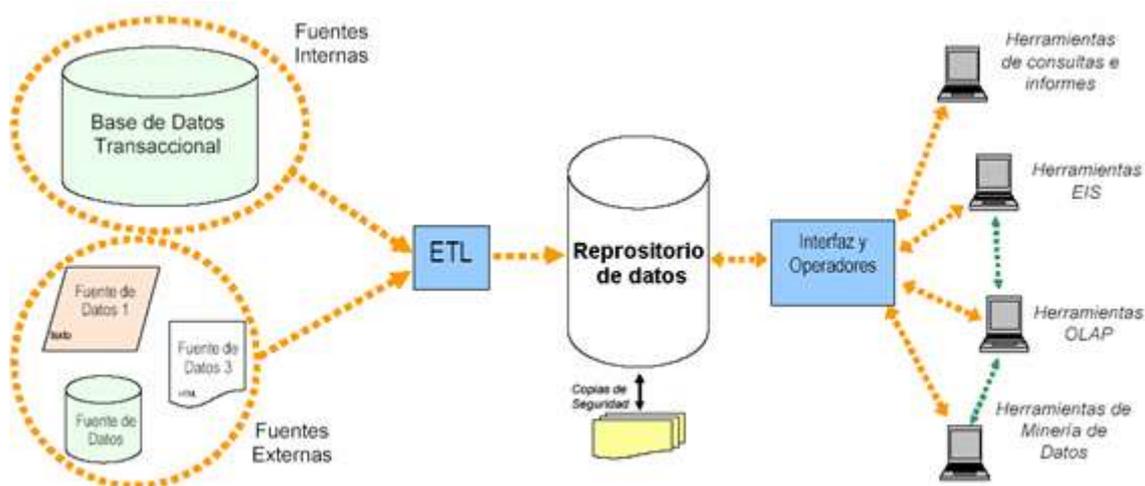


Figura 3.1 Arquitectura Data Warehouse

### 3.1.3 Data Warehouse como sistema de información

La tecnología data warehousing basa sus conceptos y diferencias entre dos tipos fundamentales de sistemas de información en todas las organizaciones: *los sistemas técnico - operacionales* y *los sistemas estratégicos*. Los sistemas técnico – operacional básicamente son los que cubren el núcleo de operaciones tradicionales de captura masiva de datos y servicios básicos de tratamiento de datos con tareas predefinidas, y los sistemas estratégicos son los orientados a soportar la toma de decisiones, facilitan la labor de la dirección, proporcionándole un soporte básico, en forma de mejor información, para la toma de decisiones. Este último es la base de un data warehouse.

### 3.1.4 Datos de entrada a un Data Warehouse

La fuente de casi toda la información del data warehouse es el ambiente operacional. A simple vista, se puede pensar que hay redundancia masiva de datos entre los dos ambientes. Desde luego, la primera impresión es esa pero se debe considerar lo siguiente:

- Los datos se filtran cuando pasan desde el ambiente operacional al de depósito. Existe mucha data que nunca sale del ambiente operacional. Sólo los datos que realmente se necesitan ingresarán al ambiente de data warehouse.
- El horizonte de tiempo de los datos es muy diferente de un ambiente al otro. La información en el ambiente operacional es más reciente con respecto a la del data warehouse. Desde la perspectiva de los horizontes de tiempo únicos, hay poca superposición entre los ambientes operacional y de data warehouse.
- El data warehouse contiene un resumen de la información que no se encuentra en el ambiente operacional.
- Los datos experimentan una transformación fundamental cuando pasa al data warehouse. La mayor parte de los datos se alteran significativamente al ser seleccionados y movidos al data warehouse. Dicho de otra manera, la mayoría de los datos se alteran física y radicalmente cuando se mueven al depósito.

### 3.1.5 Características de los Data Warehouse

Un Data Warehouse o Repositorio de datos es una colección de datos orientado a temas, integrado, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales. Se puede caracterizar un data warehouse haciendo un contraste de cómo los datos almacenados en un data warehouse, difieren de los datos operacionales. Ver la tabla 3.1.

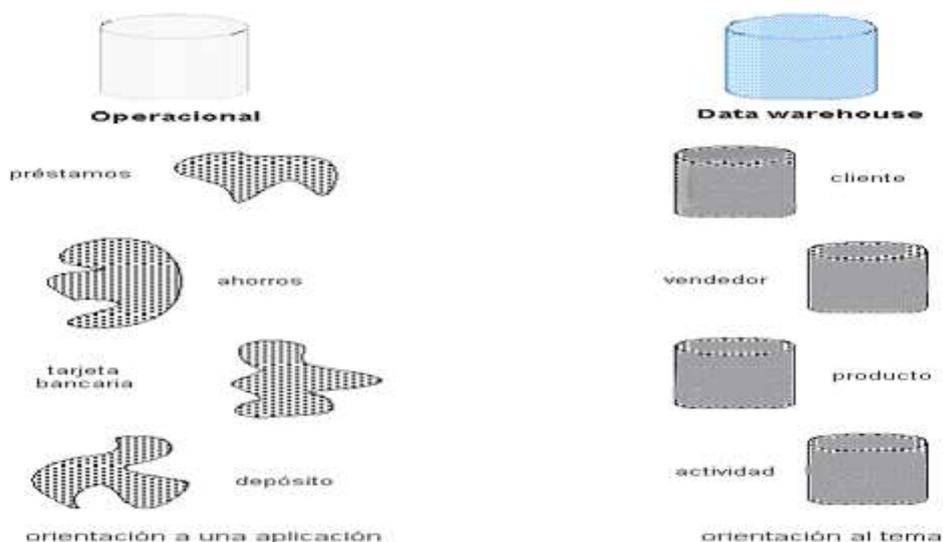
Base da Datos Operacional	Data Warehouse
Datos Operacionales	Datos del negocio para la información
Orientado a la aplicación	Orientado al sujeto
Detallada	Detallada + más resumida
Cambia continuamente	Estable
Actual	Actual + histórico

**Tabla 3.1 Data Warehouse vs Base de Datos Operacional**

### 3.1.5.1 Orientado a temas

Esta primera característica nos indica, que en un data warehouse la información se clasifica con base a los aspectos que son de interés para la compañía. De esta forma, los datos tomados están en contraste con los clásicos procesos orientados a las aplicaciones. En la figura 3.2 se muestra la diferencia entre los dos tipos de orientaciones.

Las diferencias entre la orientación de procesos y funciones de las aplicaciones y la orientación a temas, radican en el contenido de los datos a escala detallada. En él data warehouse se excluye la información que no será usada por el proceso de sistemas de soporte de decisiones, mientras que la información de las orientadas a las aplicaciones, contiene datos para satisfacer de inmediato los requerimientos funcionales y de proceso, que pueden ser usados o no por el analista de soporte de decisiones.



**Figura 3.2 Data Warehouse orientado al tema**

### **3.1.5.2 De tiempo variante**

Toda la información del data warehouse es requerida en algún momento. Esta característica básica de los datos en un repositorio, es muy diferente de la información encontrada en el ambiente operacional. En éstos, la información se requiere al momento de acceder. En otras palabras, en el ambiente operacional, cuando usted accede a una unidad de información, usted espera que los valores requeridos se obtengan a partir del momento de acceso. Como la información en el data warehouse es solicitada en cualquier momento (es decir, no "ahora mismo"), los datos encontrados en el depósito se llaman de "tiempo variante".

Esta característica también la podemos ver desde el tiempo de respuesta que tiene uno y otro ambiente. Las transacciones operacionales necesitan una respuesta inmediata porque un usuario puede estar esperando una respuesta. En el data warehouse, por el contrario, tiene un requerimiento de respuesta no crítico porque el resultado frecuentemente se usa en un proceso de análisis y toma de decisiones. Aunque los tiempos de respuesta no son críticos, los usuarios esperan una respuesta dentro del mismo día en que es hecha la consulta.

### **3.1.5.3 No volátil**

La información es útil sólo cuando es estable. Los datos operacionales cambian sobre una base momento a momento. La perspectiva más grande, esencial para el análisis y la toma de decisiones, requiere una base de datos estable. En la figura 3.3 se muestra que la actualización (insertar, borrar y modificar), se hace regularmente en el ambiente operacional sobre una base de registro por registro. Pero la manipulación básica de los datos que ocurre en el data warehouse es mucho más simple. Hay dos únicos tipos de operaciones: la carga inicial de datos y el acceso a los mismos. No hay actualización de datos (en el sentido general de actualización) en el depósito, como una parte normal de procesamiento.

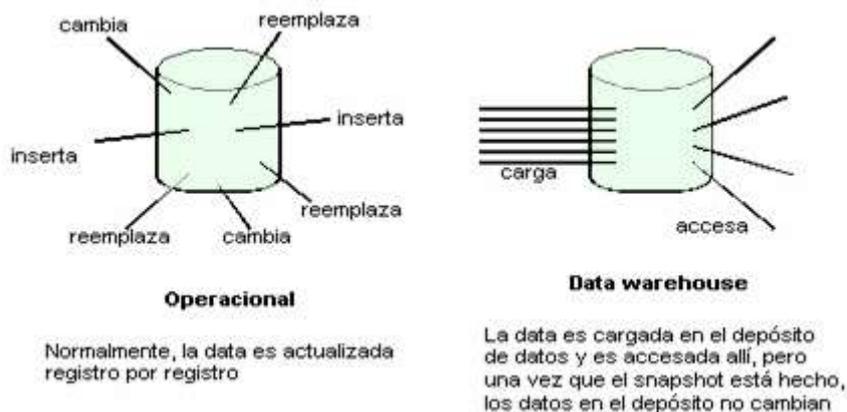


Figura 3.3 Data Warehouse No volatil

### 3.1.5.4 Integración

El aspecto más importante del ambiente data warehousing es que la información encontrada al interior está siempre integrada. El contraste de la integración encontrada en el data warehouse con la carencia de integración del ambiente de aplicaciones, se muestran en la figura 3.4, con diferencias bien marcadas. Se diferencian en la codificación, en las estructuras claves, en sus características físicas, en las convenciones de nombramiento y otros. La capacidad colectiva de muchos de los diseñadores de aplicaciones, para crear aplicaciones inconsistentes, es fabulosa.

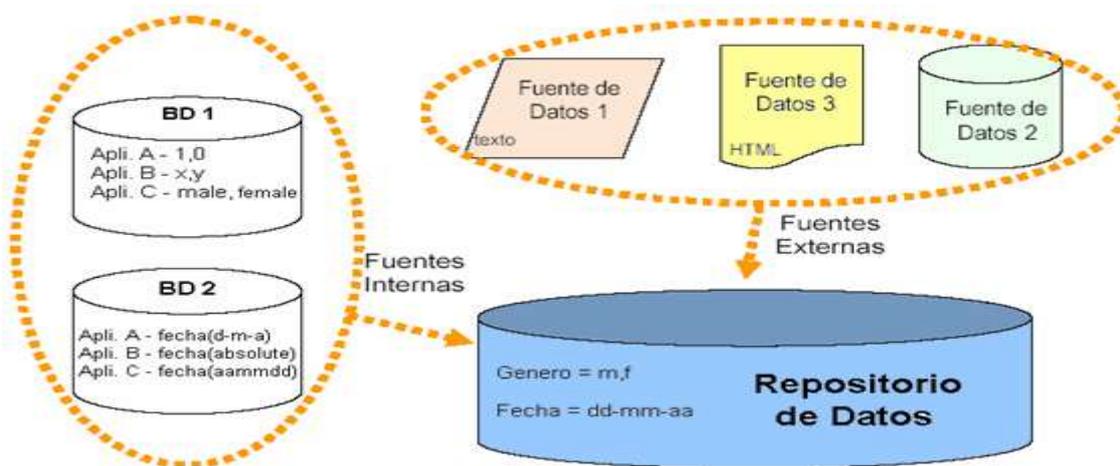
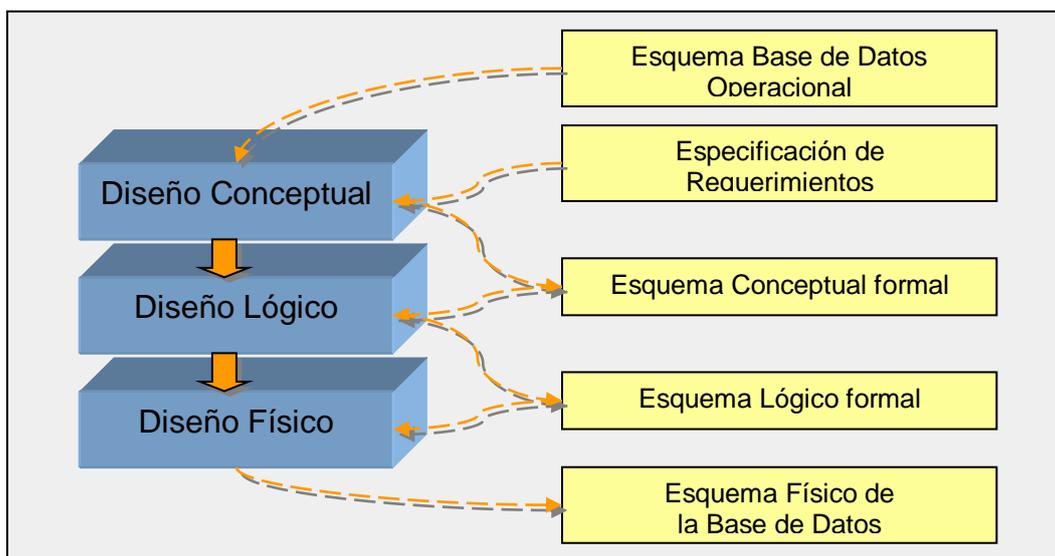


Figura 3.4 Data Warehouse Integración

### 3.1.6 Fases del Data Warehouse

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. En la Figura 3.5 se muestran las etapas con sus respectivas entradas y salidas de información.



**Figura 3.5 Fases de construcción de un DW**

En la etapa de diseño conceptual se construye un esquema de la realidad a partir de los requerimientos y/o bases fuentes. Dicho esquema 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.

### 3.1.7 Usuarios Data Warehouse

Los datos operacionales y los datos del data warehouse son accedidos por usuarios que los usan de maneras diferentes, por ejemplo:

- Los usuarios de un data warehouse necesitan acceder a los datos complejos, frecuentemente desde fuentes múltiples y de formas no predecibles.

- Los usuarios que accedan a los datos operacionales, comúnmente efectúan tareas predefinidas que, generalmente requieren acceso a una sola base de datos de una aplicación. Por el contrario, los usuarios que accedan al data warehouse, efectúan tareas que requieren acceso a un conjunto de datos desde fuentes múltiples y frecuentemente no son predecibles. Lo único que se conoce (si es modelada correctamente) es el conjunto inicial de datos que se han establecido en el depósito.
- Los usuarios del data warehouse generan consultas complejas. A veces la respuesta a una consulta conduce a la formulación de otras preguntas más detalladas, en un proceso llamado *drilling down*<sup>4</sup>. El data warehouse puede incluir niveles de resúmenes múltiples, derivado de un conjunto principal, único, de datos detallados, para soportar este tipo de uso.

Las anteriores características se pueden ver de una forma resumida en la siguiente tabla:

Uso de Base da Datos Operacional	Uso de Data Warehouse
Muchos usuarios concurrentes	Pocos usuarios concurrentes
Consultas predefinidas y actualizables	Consultas complejas, frecuentemente no anticipadas
Cantidades pequeñas de datos detallados	Cantidades grandes de datos detallados
Requerimientos de respuesta inmediata	Requerimientos de respuesta no críticos

**Tabla 3.2 Usos en Data Warehouse vs Usos de Bases de Datos Operacionales**

### 3.2 Minería de Datos (Data Mining)

Los avances tecnológicos actuales posibilitan la rápida obtención de grandes cantidades de datos de fuentes muy diversas, así como el almacenamiento eficiente de los mismos. Dichos datos encierran información muy valiosa que puede tratarse mediante los métodos tradicionales de análisis de datos, sin embargo estos métodos no son capaces de encontrar toda la información útil latente en la gran masa de datos que se maneja. En ese contexto, las técnicas de minería de datos surgen como las mejores herramientas para realizar exploraciones más profundas y extraer información nueva, útil y no trivial que se encuentra oculta en grandes volúmenes de datos.

<sup>4</sup> Drilling Down es el más viejo y venerable tipo de drilling en un data warehouse. Drilling Down significa nada mas que “dame más detalle”

### 3.2.1 Técnicas de extracción de datos

Existen varias técnicas de extracción de datos desarrolladas para identificar tendencias, información y nuevas oportunidades de tratamiento de datos ocultos. Varias de estas técnicas de Minería de Datos se han insertado en aplicaciones de software que procesan complejos algoritmos para obtener un significado coherente de la información.

Las aplicaciones de Minería de Datos no se desarrollan extensamente debido a que habitualmente se desconoce su utilidad y las posibilidades que ofrece. Una forma de descubrir las posibilidades de la extracción de datos es comparándola con otras estrategias BI (Business Intelligence) que son quizá un poco más conocidas, entre las que tenemos: Aplicación de consulta personalizada y OLAP.

- **Aplicación de consulta personalizada**

Mediante la aplicación de consulta personalizada los usuarios tiene la posibilidad de acceder a la información deseada. Por ejemplo, un usuario crea y ejecuta una consulta personalizada que contesta al interrogante: ¿qué cantidad de ingresos ha generado cada cliente durante este año?. Los resultados de esta consulta contendrían los nombres de los clientes y los ingresos para el año seleccionado. La figura 3.6 representa los resultados de la consulta personalizada.



**Figura 3.6 Aplicación de consulta personalizada**

Los ingresos por cliente podrían llevarnos a otra pregunta: ¿qué cantidad de ingresos se generan al año?. Además, también podrían responderse otras preguntas como: ¿qué cliente generó la mayor cantidad de ingresos para la compañía? y ¿qué cliente generó la cantidad más baja?. Mientras que el resultado de la búsqueda sea útil y resuelva varias cuestiones, la tecnología BI no identificará los modelos poco habituales ni revelará las relaciones anormales. Lo que el

usuario solicitó eran los ingresos por cliente para el año actual y ésta es la información que fue proporcionada; ni más ni menos.

- **OLAP**

Las aplicaciones OLAP permiten a los usuarios explorar y analizar manualmente la información resumida y detallada. Por ejemplo, un usuario crea y ejecuta un análisis OLAP que responde a la pregunta: ¿cuáles fueron los ingresos para cada trimestre de este año por región geográfica y cliente?. Los resultados de este análisis contendrían región geográfica, nombre del cliente, ingresos y los trimestres seleccionados. La figura 3.7 representa el resultado del análisis OLAP.



**Figura 3.7 Procesamiento analítico online**

Podrían formularse preguntas adicionales acerca de los datos que subrayasen los modelos de ingresos estacionales por región geográfica. Sin embargo, debe dirigir este proceso un usuario que sepa cómo tratar los datos. El OLAP sólo puede subrayar los modelos de los datos que se solicitaron. Es competencia del usuario identificar las tendencias y modelos subrayados por el análisis OLAP. Esta tecnología BI no identificará relaciones poco usuales ni revelará relaciones ocultas.

Por su parte la Minería de Datos puede decirse que es una tecnología BI que cuenta con diversas técnicas para extraer información útil, oculta y comprensible de un conjunto de datos. La Minería de Datos posibilita el descubrimiento de tendencias y modelos ocultos en extensas cantidades de datos. El resultado de un ejercicio de Minería de Datos puede tomar forma de modelos, tendencias o reglas implícitas a los datos.

Las *aplicaciones de consulta personalizada* dan un valor superficial al existente dentro de la base de datos, mientras que OLAP proporciona usuarios con una gran capacidad de profundización y comprensión. Sin embargo, la Minería de Datos llega más lejos y

proporciona a los usuarios conocimiento a través del descubrimiento de tendencias y relaciones ocultas. La combinación de la Minería de Datos con una consulta personalizada o una aplicación OLAP es extremadamente poderosa y proporciona a los usuarios conocimiento sobre los datos que están siendo analizados y la capacidad de actuar en ellos. La figura 3.8 representa el valor y el propósito de estas tecnologías BI.



Figura 3.8 Tecnologías Business Intelligence

### 3.2.2 Tipos de análisis de datos

La minería de datos ha dado lugar a una paulatina sustitución del análisis de datos *dirigido a la verificación* por un enfoque de análisis de datos *dirigido al descubrimiento del conocimiento*. La principal diferencia entre ambos, se encuentra que el último descubre información sin necesidad de formular previamente una hipótesis. Se dice entonces, que la minería de datos es un proceso que invierte la dinámica del método científico en el siguiente sentido, en el método científico, primero se formula la hipótesis y luego se diseña el experimento para coleccionar los datos que confirmen o refuten la hipótesis. Si esto se hace con la formalidad adecuada (cuidando cuáles son las variables controladas y cuáles experimentales), se obtiene un nuevo conocimiento. En la minería de datos, se coleccionan los datos y esperamos que de ellos emerjan hipótesis. Queremos que los datos nos describan o indiquen por qué son como son.

La más inocente mirada a los datos por un humano, puede inspirarle una hipótesis. Recuérdese que los humanos tenemos grandes poderes de generalización e

identificación de patrones. Luego entonces, validar esa hipótesis inspirada por los datos en los datos mismos, será numéricamente significativa, pero experimentalmente inválida. De ahí que la minería de datos debe presentar un enfoque exploratorio, y no confirmador. Usar la minería de datos para confirmar nuestras hipótesis puede ser peligroso, pues estamos haciendo una inferencia poco válida.

La aplicación automatizada de algoritmos de minería de datos permite detectar fácilmente patrones en los datos, razón por la cual la técnica *dirigida al descubrimiento del conocimiento* es mucho más eficiente que el análisis *dirigido a la verificación* cuando se intenta explorar datos procedentes de repositorios de gran tamaño y complejidad elevada. Dichas técnicas emergentes se encuentran en continua evolución como resultado de la colaboración entre campos de investigación tales como bases de datos, reconocimiento de patrones, inteligencia artificial, sistemas expertos, estadística, visualización, recuperación de información, y computación de altas prestaciones.

### 3.2.3 Descubrimiento del Conocimiento en Bases de Datos

La minería de datos se ubica en el proceso completo de extracción de información conocido como KDD, que se encarga además de preparación de los datos y de la interpretación de los resultados obtenidos. Como nos muestra la figura 3.9 la minería de datos hace parte fundamental dentro del proceso de descubrimiento del conocimiento. No debemos olvidar que de la simple aplicación de técnicas de Minería de Datos sólo se obtienen patrones que no sirven de gran cosa mientras no se les encuentre significado.

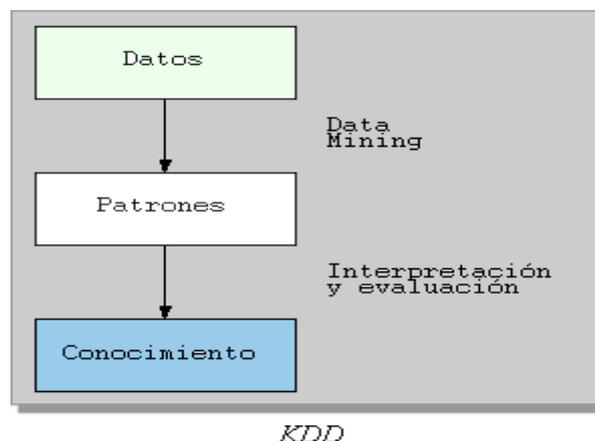


Figura 3.9 Data Mining y Knowledge Discovery in Databases

KDD se ha definido como la extracción no trivial de información potencialmente útil a partir de un gran volumen de datos en el cual la información está implícita (aunque no se conoce previamente). Se trata de interpretar grandes cantidades de datos y encontrar relaciones o patrones. Para conseguirlo harán falta técnicas de aprendizaje [Machine Learning], estadística y bases de datos.

Las investigaciones en estos temas incluyen análisis estadístico de datos, técnicas de representación del conocimiento, razonamiento basado en casos [CBR: Case Based Reasoning], razonamiento aproximado, adquisición de conocimiento, redes neuronales y visualización de datos. Tareas comunes en KDD son la inducción de reglas, los problemas de clasificación y clustering, el reconocimiento de patrones, el modelado predictivo, la detección de dependencias, etc. [13]

Los datos recogen un conjunto de hechos (una base de datos) y los patrones son expresiones que describen un subconjunto de los datos (un modelo aplicable a ese subconjunto). KDD involucra un proceso iterativo e interactivo de búsqueda de modelos, patrones o parámetros. Los patrones descubiertos han de ser válidos, novedosos para el sistema (para el usuario siempre que sea posible) y potencialmente útiles.

Se han de definir medidas cuantitativas para los patrones obtenidos (precisión, utilidad, beneficio obtenido...). Se debe establecer alguna medida de interés que considere la validez, utilidad y simplicidad de los patrones obtenidos mediante alguna de las técnicas de Minería de Datos. El objetivo final de todo esto es incorporar el conocimiento obtenido en algún sistema real, tomar decisiones a partir de los resultados alcanzados o, simplemente, registrar la información conseguida y suministrársela a quien esté interesado.

Las técnicas de Minería de Datos (una etapa dentro del proceso completo de KDD, ver figura 3.10<sup>5</sup>) intentan obtener patrones o modelos a partir de los datos recopilados. Decidir si los modelos obtenidos son útiles o no suele requerir una valoración subjetiva por parte del usuario.

---

<sup>5</sup> La figura 3.10 muestra el proceso guía utilizado para construir la minería de datos y es explicado en detalle en el próximo capítulo.

### 3.2.4 Categorías de la Minería de Datos

Los algoritmos de minería de datos se clasifican en dos grandes categorías: supervisados o predictivos y no supervisados o de descubrimiento del conocimiento [14].

Los algoritmos **supervisados** o *predictivos* predicen el valor de un atributo (etiqueta) de un conjunto de datos, conocidos otros atributos (*atributos descriptivos*). A partir de datos cuya etiqueta se conoce se induce una relación entre dicha etiqueta y otra serie de atributos. Esas relaciones sirven para realizar la predicción en datos cuya etiqueta es desconocida. Esta forma de trabajar se conoce como *aprendizaje supervisado* y se desarrolla en dos fases: Entrenamiento (construcción de un modelo usando un subconjunto de datos con etiqueta conocida) y prueba (prueba del modelo sobre el resto de los datos).

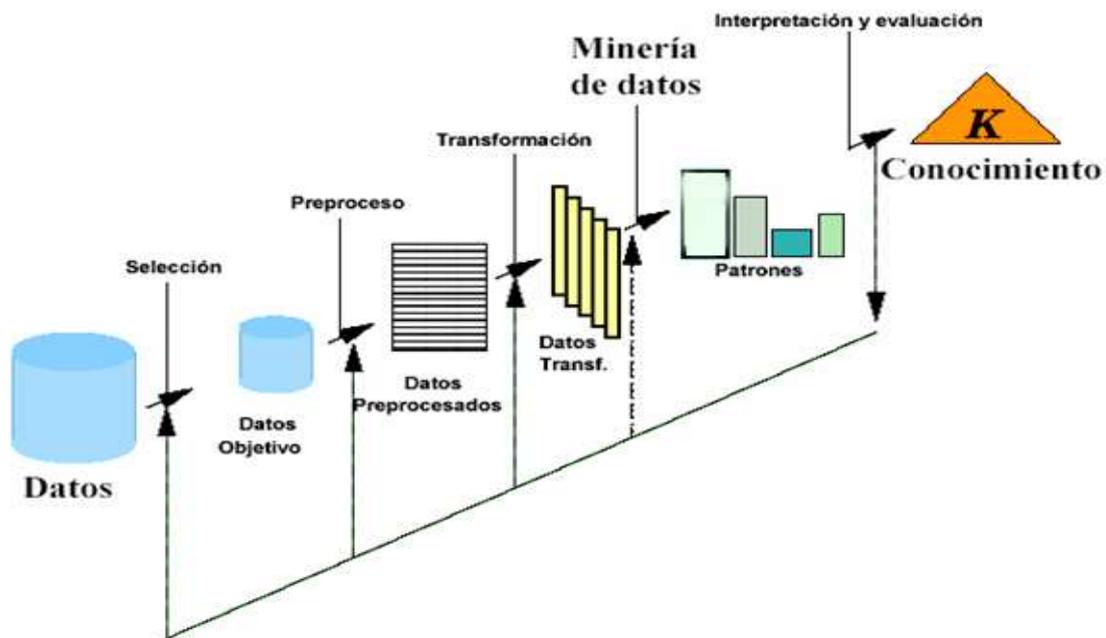


Figura 3.10 Proceso de Descubrimiento de Conocimiento en Bases de Datos KDD

Cuando una aplicación no es lo suficientemente madura no tiene el potencial necesario para una solución predictiva, en ese caso hay que recurrir a los métodos **no supervisados** o *de descubrimiento del conocimiento* que descubren patrones y tendencias en los datos actuales (no utilizan datos históricos). El descubrimiento de esa información sirve para llevar a cabo acciones y obtener un beneficio (científico o de negocio) de ellas. En la tabla siguiente se muestran algunas de las técnicas de minería de ambas categorías.

Supervisados	No Supervisados
Árboles de decisión	Detección de desviaciones
Inducción neuronal	Segmentación
Regresión	Agrupamiento
Series Temporales	Reglas de asociación
Actual	Patrones secuenciales

**Tabla 3.3 Clasificación de las técnicas de minería de datos**

### 3.2.5 Algoritmos de Minería de Datos

Para hacer una selección inteligente de herramientas y tecnologías de Minería de Datos (MD) hay que determinar dónde está la diferencia. Dado que hay demasiadas superposiciones, una de las mejores vías para ver las significativas diferencias entre los algoritmos es ver que es lo que tienen en común. Por ejemplo, cada algoritmo de MD tiene:

- **Estructura de modelo**

La estructura que determina el modelo (un árbol, red neuronal o vecino más cercano) Es un modelo percibido porque la instantaneidad real del modelo puede ser preguntas SQL en caso de árboles de decisiones o sistemas basados en reglas; o ecuaciones matemáticas en caso de regresiones estadísticas.

- **Un plan de investigación**

Como el algoritmo corrige o modifica el modelo todo el tiempo cuando más datos se hacen disponibles. Por ejemplo, redes neuronales investigan a través de espacio de peso de enlaces mediante algoritmos de retro propagación y el algoritmo genético investiga a través de ponderaciones aleatorias y recombinaciones genéticas

- **Un proceso de validación**

¿Cuándo se acaba el algoritmo al haber creado un modelo valido? Por ejemplo, los árboles de decisiones CART utilizan validación cruzada para determinar el nivel óptimo de crecimiento del árbol. Por su parte, las redes neuronales no

tienen una técnica especializada de validación para determinar la terminación, pero la validación cruzada a menudo se usa en el exterior de redes neuronales.

### **3.2.5.1 Árboles de Decisión**

Los árboles de decisión, también denominados árboles de clasificación o de identificación, sirven, como su propio nombre indica, para resolver problemas de clasificación. La construcción de árboles de decisión es el método de aprendizaje inductivo supervisado más utilizado. Como forma de representación del conocimiento, los árboles de decisión destacan por su sencillez. A pesar de que carecen de la expresividad de las redes semánticas o de la lógica de primer orden, su dominio de aplicación no está restringido a un ámbito concreto sino que pueden ser utilizados en diversas áreas (desde aplicaciones de diagnóstico médico hasta juegos como el ajedrez o sistemas de predicción meteorológica) [15].

El conocimiento obtenido en el proceso de aprendizaje se representa mediante un árbol en el cual cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada hoja del árbol se refiere a una decisión (una clasificación). Un árbol de decisión puede usarse para clasificar un caso comenzando desde su raíz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos hasta que encontremos una hoja del árbol.

La construcción de los árboles de decisión se hace recursivamente de forma descendente (se parte de conceptos generales que se van especificando conforme se desciende en el árbol), por lo que se emplea el acrónimo TDIDT [Top-Down Induction on Decision Trees] para referirse a la familia completa de algoritmos de este tipo.

### **3.2.6 Fases Minería de Datos**

La aplicación de los algoritmos de minería de datos requiere la realización de una serie de actividades previas encaminadas a preparar los datos de entrada debido a que, en muchas ocasiones dichos datos proceden de fuentes heterogéneas, no tienen el formato adecuado o contienen ruido. Por otra parte, es necesario interpretar y evaluar los resultados obtenidos [16].



Figura 3.11 Fases de Data Mining

El ciclo de vida de un proyecto data mining consiste en seis fases. La figura 3.11 muestra las fases de un proceso de data mining. La secuencia de las fases no es necesariamente una camisa de fuerza, como nos muestra la figura es un proceso flexible y altamente iterativo, estableciéndose retroalimentación entre los mismos. Además, no todos los pasos requieren el mismo esfuerzo, generalmente la etapa de preprocesamiento es la más costosa ya que representa aproximadamente el 60 % del esfuerzo total, mientras que la etapa de minería sólo representa el 10%. Es casi siempre necesario moverse entre las diferentes fases. Esto depende del resultado de cada fase, que fase o que tarea particular de una fase tenga que desarrollar a continuación. Las flechas indican la más importante y frecuente dependencia entre fases. El círculo exterior en la figura 3.11 simboliza el ciclo natural de data mining.

A continuación se hará una breve descripción de todas las fases [17]:

1. **Determinación de objetivos:** esta fase inicial se enfoca en entender los objetivos del proyecto y requerimientos desde la perspectiva de lo que busca la organización, convirtiendo entonces este conocimiento en la definición de un

problema data mining y en un plan preliminar diseñado para lograr los objetivos.

## 2. Preparación de datos

- a. Selección: Identificación de las fuentes de información externas e internas y selección del subconjunto de datos necesario. Desarrollar actividades para familiarizarse con los datos, identificar problemas con la calidad de los datos, descubrir una primera comprensión de los datos o descubrir subgrupos interesantes desde hipótesis o desde la información oculta.
  - b. Preprocesamiento: Estudio de la calidad de los datos y determinación de las operaciones de minería que se pueden realizar. Se encarga de darle forma a todas las actividades planteadas en la sección anterior para construir el dataset final (datos que se van a introducir en la herramienta de minería) desde los datos iniciales. Las tareas de preparación de datos son probablemente desarrolladas en múltiples tiempos y no en un orden preescrito. Las tareas incluyen tablas, archivos y selección de atributos para una buena transformación y limpieza de datos.
3. **Transformación de datos:** conversión de datos en un modelo analítico. En esta fase, varias técnicas de modelado son seleccionadas y aplicadas a sus parámetros y calibrados a valores óptimos. Típicamente, hay varias técnicas para el mismo tipo de problema de data mining. Algunas técnicas tienen requerimientos específicos en la forma de los datos.
  4. **Minería de datos:** tratamiento automatizado de los datos seleccionados con una combinación apropiada de algoritmos.
  5. **Análisis de resultados:** interpretación de los resultados obtenidos en la etapa anterior, generalmente con la ayuda de una técnica de visualización.
  6. **Asimilación de conocimiento:** aplicación del conocimiento descubierto.

## **Capítulo 4 WSMINING: PLATAFORMA DE ACUERDOS DE NIVEL DE SERVICIO BASADA EN MINERÍA DE DATOS Y SERVICIOS WEB, PARA LA RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA**

### **4.1 Arquitectura funcional**

WSMINING es una plataforma donde cada módulo que la compone fue implementado en el lenguaje de programación java; tiene como objetivo construir una plataforma de acuerdos de nivel de servicio que aplique el concepto de Data Warehouse y Data Mining, y esté soportada en una arquitectura lógica para la gestión de recursos utilizando Servicios Web – WSMF.

La base de la aplicación depende fundamentalmente de los datos recolectados y de la calidad de los mismos. Los datos permiten descubrir algún tipo de conocimiento o información relevante que se encuentre implícita en ellos, de manera que deben ser tratados de la mejor forma para extraerlos de los equipos de red, sin que sufran ninguna alteración. Además hay que tener presente que los equipos a manejar son muy importantes dentro de Intranet y no se les debe aumentar demasiado la carga.

Teniendo en cuenta estas premisas, se proporciona una plataforma que recolecte y trate bien los datos; y que además de eso no perjudique el funcionamiento de la red. La plataforma construida según el framework WSMF, la selección de las tecnologías y herramientas utilizadas permite de una forma estructurada recolectar y manipular los datos.

La utilización del protocolo SNMP es de vital importancia para la extracción de datos de los equipos y es él que permite, gracias a las MIB's instaladas, contar con diversos parámetros. Los parámetros de un Switch son distintos a los del Proxy y deben ser manejados correctamente para la construcción de un SLA; por que de lo contrario, es muy probable que el resultado obtenido al final del proceso no sea coherente.

Una vez se tenga claro los parámetros que se desean manipular de los equipos y queden definidos para un SLA, el proceso de recolección de datos comienza y es

entonces, cuando pasan a ser manipulados por los beans de gestión (MBean) para luego ser almacenados en una base de datos. Esta base de datos la denominamos base de datos fuente, y es el soporte de datos para las fases siguientes.

Cuando se tienen los datos de los dispositivos de red recolectados en la base de datos fuente se empieza con la concepción del Data Warehouse y del Data Mining. El Data Warehouse se encarga de construir la base de datos multidimensional, donde las entradas son los datos recolectados desde los equipos de la Red de Datos estructurados en una base de datos operacional (base de datos fuente) y la salida será otra base de datos pero con un concepto multidimensional, y dejar listos los datos para que pasen hacia la siguiente fase, la minería de datos.

El otro concepto importante dentro del proyecto está relacionado con el Data Mining o Minería de Datos, utilizada como herramienta de predicción según el algoritmo aplicado y una etapa de interpretación del modelo utilizado, que sirve como soporte para la construcción de los Acuerdos de Nivel de Servicio. Los Acuerdos de Nivel de Servicio son el resultado del prototipo, y sirven al administrador de la red, tanto para ver el comportamiento interno que está teniendo la red como para el soporte en la toma de decisiones.

Los dispositivos o elementos de red utilizados, el servidor proxy “hiperion” y el switch principal “Acclar” del backbone, manejan cantidades de datos bastante significativas, pero en formatos y estructuras que no son óptimas, por tal motivo, se debe llevar a cabo la construcción de un Data Warehouse según unas dimensiones acordadas, que estructure los datos de entrada y deje lista una base de datos como entrada para la etapa de Minería de Datos.

Por su parte, la Minería de Datos utiliza los datos organizados y unos algoritmos específicos para obtener resultados siguiendo un proceso de desarrollo. Para la construcción de la aplicación se utiliza un algoritmo de predicción e interfaces gráficas que permitan analizar los resultados obtenidos y que sirvan de soporte para la construcción de unos buenos Acuerdos de Nivel de Servicio.

#### **4.2 Arquitectura Modular**

La arquitectura final de la aplicación está compuesta por cinco módulos funcionales e independientes que interactúan para ofrecer los servicios correspondientes a la plataforma para Acuerdos de Nivel de servicio propuesta.

Dichos módulos funcionales son: Módulo Gestión de Entorno, Módulo Recolección de Datos, Módulo Análisis de Datos, Módulo Comparación de Resultados y Módulo Presentación.

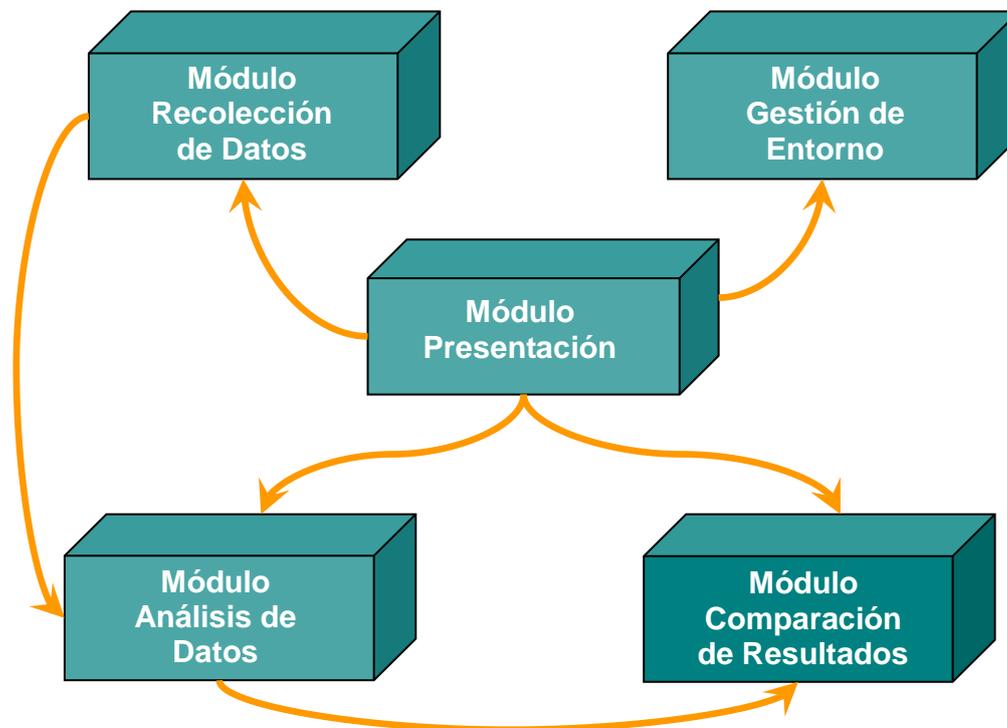


Figura 4.1 Módulos Funcionales de la aplicación

#### 4.2.1 Módulo Presentación.

Aquí encontramos la implementación de un conjunto de funcionalidades encargadas de ofrecer los servicios propios del prototipo de validación de la plataforma de Acuerdos de Nivel de Servicio WSMining por medio de una aplicación Web.

Esta aplicación Web sigue el patrón de diseño **MVC** (Modelo – Vista – Control) bajo la API de Struts en el lenguaje de programación JAVA.

##### 4.2.1.1 Modelo

El modelo de la aplicación Web - implementada para soportar la plataforma de Acuerdos de nivel de Servicio WSMining - corresponde a los Módulos de Configuración de Entorno, de Recolección de Datos, de Análisis de Datos y de Comparación de Resultados. Todos estos módulos, a excepción de Configuración de entorno, están conformados por Componentes de Gestión o MBeans (Management Bean) que son publicados como Servicios Web de acuerdo al a filosofía de WSMF

(Web Services Management Framework). Estos servicios Web son invocados por componentes que residen dentro de la aplicación Web y que dependen estrictamente del control de la misma. El Módulo de Configuración de entorno de la aplicación Web está conformado por simples componentes java que implementan persistencia de datos e información.

#### **4.2.1.2 Vista**

La Vista de la aplicación Web - que se implementó en este módulo de presentación - está conformada por Java Server Pages (JSP) y le permiten al usuario del prototipo de validación tener la posibilidad de utilizar todas las funcionalidades de la plataforma de Acuerdos de nivel de Servicio WSMining desde una conexión a Internet a través de un Browser.

#### **4.2.1.3 Control**

El control de la Aplicación Web – implementada para el prototipo de validación de la plataforma de acuerdos de Nivel de Servicio – está conformado por archivos **.do** que están implementados como Actions en código java y permiten evaluar las peticiones del usuario de la aplicación y redireccionar eficientemente dichas peticiones hacia la salida o respuesta más apropiadas.

### **4.2.2 Módulo Gestión de Entorno**

Este módulo es el encargado de organizar y administrar la información correspondiente a las características de entorno de la plataforma para el desarrollo de acuerdos de nivel de servicio. Tiene como objetivo fundamental estructurar los datos concernientes a los equipos de red y sus parámetros. La información de los usuarios que tienen acceso al prototipo de validación también es administrada por este módulo, permitiendo otorgar niveles de seguridad de los datos, por medio de la verificación y autorización del ingreso al prototipo y la discriminación de usuario.

Este módulo ha sido implementado con Componentes Java (Java Beans) y hacen uso de la persistencia de datos a través de JDBC (Java Database Connectivity) utilizando el motor de Base de datos Firebird Versión 1.5.

### **4.2.3 Módulo Recolección de Datos**

Este módulo tiene como objetivo realizar los procedimientos necesarios para la recolección y almacenamiento de los datos correspondientes a los parámetros de los equipos de red representados por OID's (Object Identifiers).

La recolección está implementada por un Componente de Gestión (MBean) que es publicado como un Servicio Web según los lineamientos estipulados por WSMF. Dicho Componente de Gestión (MBean) está constituido por métodos de obtención de datos SNMP a través de la API JMX (Java Management Extensions).

Una vez los datos son recolectados, el Bean de Gestión (MBean) invoca la interfaz remota de un componente de sesión (Session EJB) el cual tiene la lógica implementada para almacenar los datos recolectados por medio de la conexión con la interfaz local de un Componente de Entidad (Entity EJB CMP 2.0) que es el que accede directamente a la base de datos. Este procedimiento se hace con el fin de minimizar el número de conexiones remotas hacia la base de datos a través de EJB's y es conocido como Patrón de Fachada[18].

#### **4.2.4 Módulo Análisis de Datos**

En este módulo se efectúa el análisis de los diferentes tipos de datos recolectados y almacenados por el Módulo de Recolección de datos. Nuevamente es un Componente de Gestión – publicado como un Servicio Web. El Bean de Gestión hace una transformación de los datos por medio de un proceso de Data Warehouse y llena un repositorio de datos de acuerdo con los perfiles previamente definidos; a continuación el Bean de Gestión aplica procesos de Minería de Datos - al repositorio de datos anteriormente definidos - por medio del uso de la herramienta WEKA.

Una vez que la herramienta ha generado un análisis predictivo de la información, los resultados son devueltos a la aplicación Web a través del Servicio Web invocado.

#### **4.2.5 Módulo Comparación de Resultados**

Finalmente, una vez que los datos recolectados desde los equipos de red han pasado el proceso de análisis predictivo, es posible ahora presentar un reporte fundamentado que permita construir acuerdos de nivel de servicio que reflejen el comportamiento real de los diferentes equipos de red.

Por medio de la invocación de un Componente de Gestión (MBean) – que es publicado como un Servicio Web – se realiza la comparación de resultados según los parámetros, los equipos de red y los datos que se tienen para construir un acuerdo de nivel de Servicio determinado.

Este Componente de Gestión se encarga de presentarle al usuario de la herramienta un estudio detallado de los procesos realizados en el análisis, culminando con un formato sugerido de Acuerdo de Nivel de Servicio que contiene valores reales y posibles de cumplir de acuerdo a las capacidades de los equipos de red.

### 4.3 Arquitectura de validación

La descripción de los módulos da una visión bastante general de como esta conformada la estructura de la plataforma y sirve de referencia para construir la arquitectura de validación. La arquitectura de validación representa exactamente la forma como fue construida la aplicación WSMINING.

A continuación se presenta la arquitectura propuesta para la aplicación con las tecnologías utilizadas<sup>6</sup> para abordar la solución

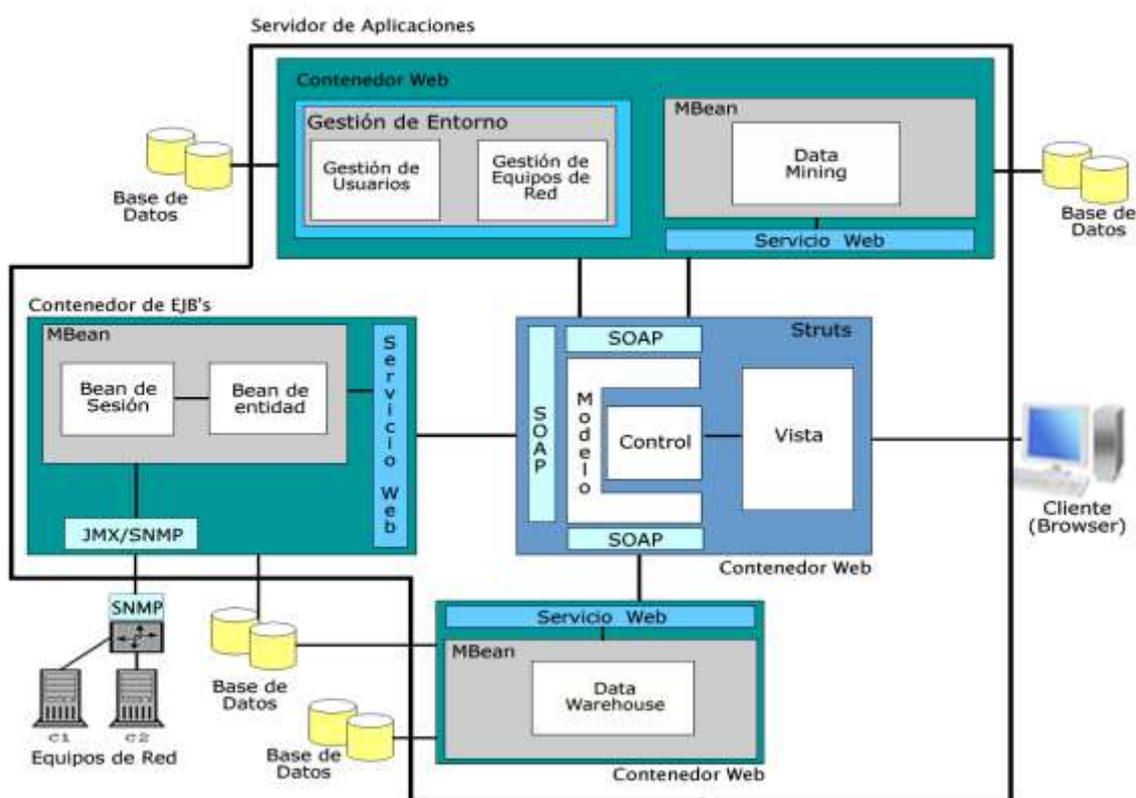


Figura 4.2 Arquitectura de la aplicación

La aplicación Web que soporta la implementación del prototipo está contenida dentro del Servidor Web (Web Container) Apache Tomcat versión 4.1 que viene incluido

<sup>6</sup> La descripción en detalle de todas las tecnologías y herramientas utilizadas para la construcción de la plataforma se explican en el Anexo C

dentro de la distribución del Servidor de Aplicaciones Java JBOSS versión 3.2.5. Los Servicios Web publicados por los diferentes módulos están contenidos dentro del Servidor Web Apache Tomcat Axis versión 1.1 que viene incluido dentro de la distribución del Servidor de Aplicaciones Java JBOSS versión 3.2.5. Finalmente, todos los EJB's (Enterprise Java Beans) y Componentes de Gestión (MBeans) están contenidos dentro del Contenedor de Negocio (Business Container) del Servidor de aplicaciones Java JBOSS versión 3.2.5.

Para la validación de la aplicación la Red de Datos de la Universidad del Cauca asignó dos equipos de red muy importantes que describen perfectamente el comportamiento de la red, como lo son: el servidor Proxy *Hyperion* y el Switch *Accelar*. En la arquitectura estos equipos son los equipos de red C1 y C. Estos equipos manejan grandes volúmenes de datos (alto tráfico) durante las 8 a.m y las 6 p.m, horario durante el cual la red esta siendo utilizada al máximo por sus usuarios. A demás, por la importancia de los equipos dentro de la Intranet, son demasiados sensibles a la carga del procesamiento; razón por la cual, no deben ser sobrecargados (por lo menos en tiempo de mayor uso) con herramientas o aplicaciones de gestión que consuman su procesamiento. La arquitectura presenta a SNMP como el protocolo utilizado para la extracción de la información (parámetros) de los equipos.

Una vez definidos los equipos se deben considerar los parámetros para los cuales el administrador de la red desea tener información de su comportamiento en la red según las necesidades encontradas para definir un SLA y que requiere de un soporte para la creación del mismo.

Los SLA tienen un tiempo de finalización, después de este tiempo la aplicación entrega resultados. Los resultados le permiten al administrador de la red, ver el comportamiento que tiene los equipos y la red de acuerdo a los parámetros escogidos. La aplicación entrega resultados en forma textual (por medio de tablas) y gráfica<sup>7</sup> (en un árbol de decisión).

Los resultados esperados están relacionados con valores máximos, mínimos y promedios de los parámetros tomando como referencia siempre el tiempo de duración del SLA, también de porcentajes del cumplimiento o no de un acuerdo de nivel de servicio. Básicamente el valor promedio de un parámetro se calcula: Por ejemplo,

---

<sup>7</sup> Ver el Anexo B referente al manual de usuario

tomando los datos del parámetro durante todo el día, desde las 8 a.m hasta las 6 p.m; de todos esos valores se discrimina el valor más alto y el más bajo<sup>8</sup>, se calcula un promedio del día y se almacena para luego ser tomado en cuenta en el promedio general. Si el SLA está estipulado para un mes, se toma el promedio de todos los días (por ejemplo los lunes) durante ese mes y se calcula el valor promedio general, este valor promedio es el que se utiliza para hacer la comparación de cumplimiento del SLA (cumple: SI o NO). Por ejemplo si el valor promedio general es X del parámetro Y para los lunes, el valor del parámetro Y de cada hora de los lunes, es comparado con el valor X.

El porcentaje de cumplimiento del SLA le permite al administrador de la red tener una referencia de su comportamiento durante un tiempo estipulado. Y va a indicar que valor debe ser tomado, para un parámetro en particular, que permita construir un SLA.

#### **4.4 Desarrollo del prototipo**

Para la construcción de la plataforma se analizan los requerimientos funcionales que debe tener todo el sistema, además de observar cada componente por separado para descubrir cual es la utilidad y el aporte de cada uno de ellos dentro del sistema.

Como primera medida, en cuanto a los requerimientos, la plataforma WSMINING pretende ser una opción de soporte para la Red de Datos que ayude a construir acuerdos de nivel de servicio relacionados con el comportamiento de la Intranet a nivel de Proxy y Switch. La aplicación debe permitir tener un acceso Web a través de una interfaz amigable y práctica de manejar, de tal forma que facilite la manipulación de usuarios, equipos de red, SLA y resultados.

Los equipos utilizados tienen diferentes parámetros que pueden ser editados y utilizados para extraer información del comportamiento en la red, dependiendo de la MIB que tenga instalada. Los parámetros son agregados a la aplicación según el equipo y las necesidades de un determinado SLA. Cuando los parámetros se agregan pueden ser manipulados y combinados para construir diferentes SLA.

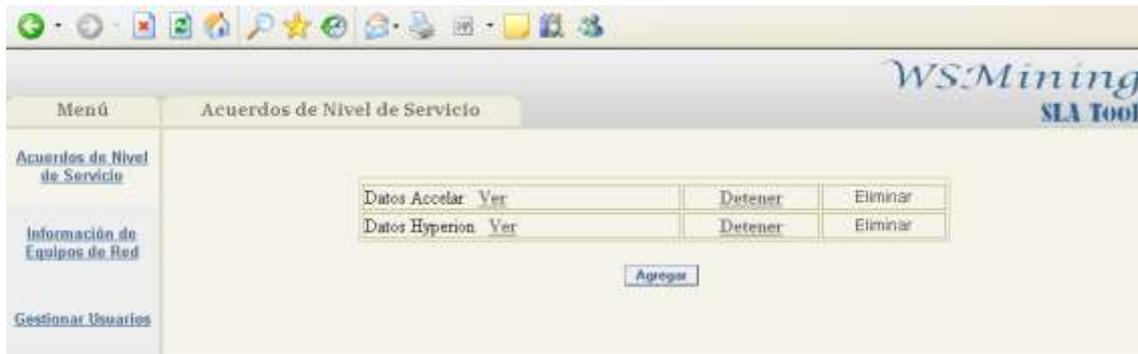
La aplicación cuenta con dos tipos de usuarios (administrador y operador), dependiendo de los permisos con que cuente. Los administradores pueden usar

---

<sup>8</sup> La discriminación se hace por que a veces pueden ser valores que no corresponden al comportamiento normal del parámetro durante el día. Por ejemplo el reinicio del equipo de red de donde esta siendo tomado el dato, casos fortuitos de caída de algún enlace, etc.

libremente toda la aplicación, gestionar usuarios, gestionar equipos de red (Hiperion o Accelar), gestionar los parámetros de un equipo seleccionado, definir y verificar SLA's. Los operadores por su parte, no pueden gestionar ningún parámetro, pero pueden ver los SLA que se encuentran activos, sus respectivos resultados y sugerencias.

La siguiente figura presenta la interfaz de un usuario administrador que muestra dos acuerdos de nivel de servicio activos, y su menú de navegabilidad a la izquierda.



**Figura 4.3 Interfaz administrador de la aplicación**

#### 4.4.1 Casos de Uso

Teniendo en cuenta los requerimientos específicos del sistema, se han definido seis casos de uso principales que engloban su funcionalidad total y son: Validar Acceso, Gestionar Usuarios, Gestionar Información de Equipos de Red, Gestionar Parámetros, Definir un Acuerdo de nivel de Servicio y Verificar el cumplimiento de los valores metas de un Acuerdo de Nivel de Servicio. A continuación se presentan los diagramas de casos de uso de diseño y la descripción de cada caso de uso.

#### 4.4.1.1 Diagrama de casos de uso

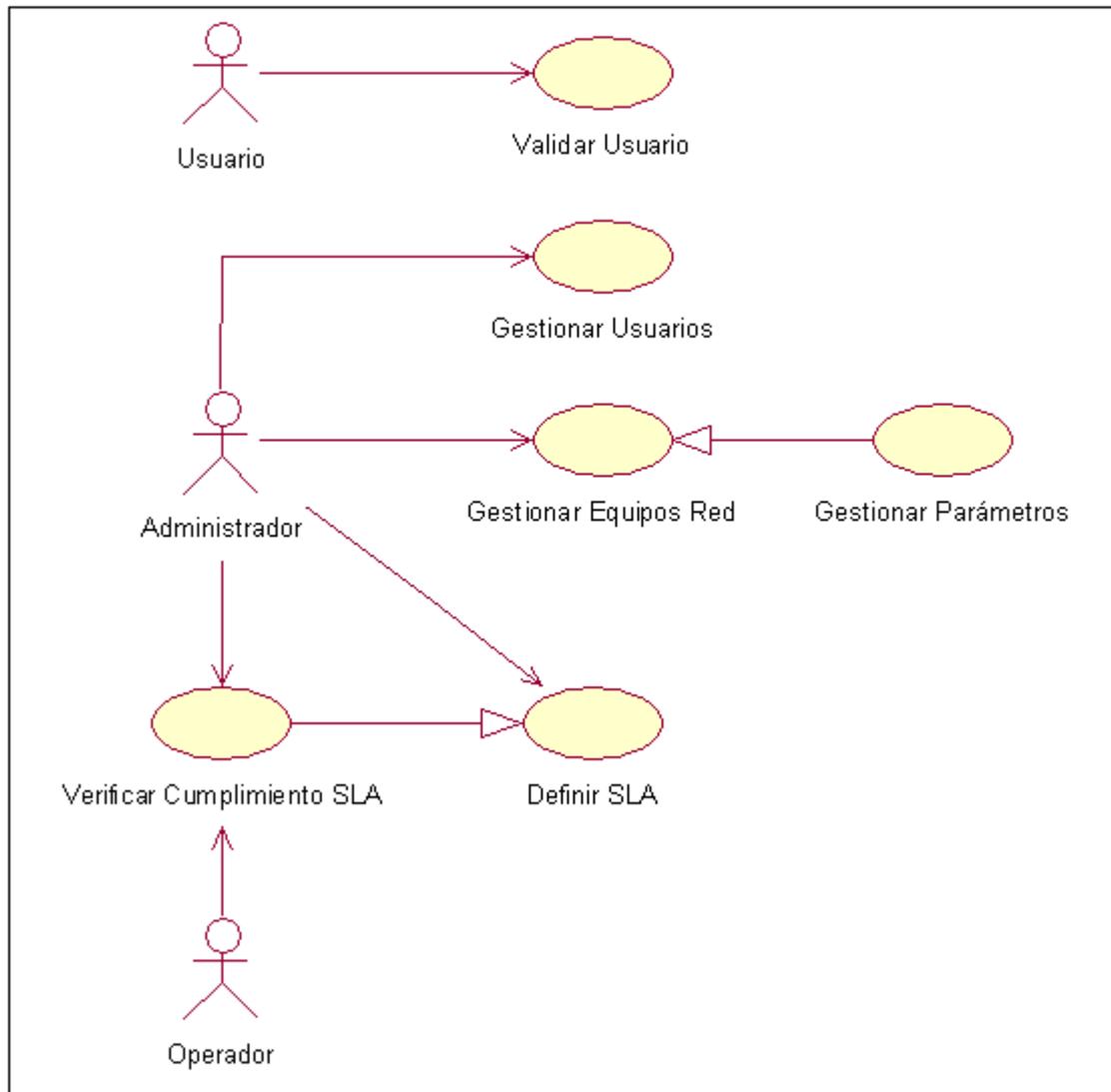


Figura 4.4 Diagrama de Casos de Uso Iniciales – Nivel 1

Los casos de uso anteriormente mencionados están compuestos a su vez por otros casos de uso orientados a una funcionalidad más específica dentro del prototipo de validación.

El caso de uso Gestionar Usuarios puede ser visto como la suma de tres funcionalidades básicas que son: Ingresar Usuario, Modificar Usuario y Eliminar Usuario.

El caso de uso Gestionar Información de Equipos de Red puede ser visto como la suma de tres funcionalidades básicas que son: Ingresar Equipo de Red, Modificar Equipo de Red y Eliminar Equipo de Red.

El caso de uso Gestionar Parámetros puede ser visto como la suma de tres funcionalidades básicas que son: Ingresar Parámetro, Modificar Parámetro y Eliminar Parámetro.

El caso de uso Verificar el Cumplimiento de un Acuerdo de Nivel de Servicio (SLA) está comprendido por tres procesos esenciales que son: Verificar los valores metas de un Acuerdo de Nivel de Servicio (SLA), Análisis de los datos recolectados durante la verificación y Comparación de los resultados obtenidos durante el análisis de los datos.

#### 4.4.1.2 Descripción de los Casos de Uso

Para tener una idea más clara de cómo está compuesto el prototipo de validación de la Plataforma para Acuerdos de Nivel de Servicio Basada en Minería de Datos y Servicios Web, para la Red de Datos de la Universidad del Cauca, se presenta a continuación una descripción detallada de los casos de uso definidos.

<b>CASO DE USO 1:</b>	<b>Validar Usuario</b>	
<b>ACTOR:</b>	Usuario	
<b>PROPÓSITO:</b>	Reconocer el rol de cada usuario cuando ingresa al sistema y de acuerdo a esto valida los niveles de acceso de cada usuario dentro del sistema.	
<b>RESUMEN:</b>	Un Usuario del sistema puede ser un administrador o un Operador. En la validación del acceso, el sistema identifica el rol de cada usuario y según sea, despliega funcionalidades y servicios concernientes al rol.	
<b>PRECONDICIONES:</b>	Estar registrado ante el sistema	
<b>ESCENARIO</b>	<b>USUARIO</b>	<b>SISTEMA</b>
	1. Ingresar dato de identificador y clave.  2. Envía datos.	3. Verifica la correspondencia del identificador y clave del usuario. <b>E1</b>  4. Valida el acceso.  5. Despliega la interfaz principal del usuario dependiendo del rol que este desempeñe. <b>1</b>
<b>POSCONDICIONES:</b>	Interfaz de usuario desplegada	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno.	
<b>NOTAS:</b>	1. Los servicios serán diferentes dependiendo del rol que desempeñe cada usuario así:	

	Si el usuario es un administrador, la interfaz principal de administrador contendrá gestionar Usuarios, y gestionar Acuerdos de Nivel de Servicio y gestionar Equipos de red. Si el usuario es un Operador, los servicios presentados en su interfaz corresponden al análisis de los Acuerdos de nivel de Servicio.
<b>EXCEPCIONES:</b>	<b>E1</b> Si el código de identificación no corresponde al de algún usuario registrado ante el sistema, se despliega un mensaje de error de validación de acceso y retorna al evento 1.

<b>CASO DE USO 2:</b>	<b>Gestionar Usuarios</b>	
<b>ACTOR:</b>	Administrador	
<b>PROPÓSITO:</b>	Desplegar al administrador un menú con las opciones crear, modificar y eliminar los usuarios del sistema.	
<b>RESUMEN:</b>	Este caso de uso permite visualizar al administrador las opciones de gestión de usuarios.	
<b>PRECONDICIONES:</b>	Haber Validado acceso	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	1. Desea gestionar la información de los usuarios del sistema. <b>E1</b>	2. Despliega los servicios correspondientes a la gestión de usuarios. <b>1</b>
<b>POSCONDICIONES:</b>	Los servicios relacionados con la gestión de usuarios son desplegados.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno	
<b>NOTAS:</b>	1 En esta interfaz se presentan los servicios de la gestión de usuarios: Ingresar, modificar y eliminar usuarios.	
<b>EXCEPCIONES:</b>	<b>E1</b> Si el administrador decide cancelar la gestión de un usuario, se devuelve a la interfaz principal del administrador.	

<b>CASO DE USO 3:</b>	<b>Gestionar Información de Equipos de Red</b>	
<b>ACTOR:</b>	Administrador	
<b>PROPÓSITO:</b>	Desplegar al administrador un menú con las opciones crear, modificar y eliminar la información correspondiente a los equipos de red.	
<b>RESUMEN:</b>	Este caso de uso permite visualizar al administrador las opciones para administrar los datos de los equipos de red del sistema.	
<b>PRECONDICIONES:</b>	Haber Validado acceso.	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	1. Desea gestionar la información de los equipos de red. <b>E1</b>	2. Despliega las opciones correspondientes a la gestión de Equipos de red. <b>1</b>

<b>POSCONDICIONES:</b>	Las opciones relacionadas con la gestión de Equipos de red son desplegadas.
<b>FLUJOS ALTERNATIVOS:</b>	Gestionar Parámetros.
<b>NOTAS:</b>	<b>1</b> En esta interfaz se presentan las opciones de la gestión de equipos de red: Ingresar, modificar y eliminar Equipos de red.
<b>EXCEPCIONES:</b>	<b>E1</b> Si el administrador decide cancelar la gestión de la información de un equipo de red, se devuelve a la interfaz principal del administrador.

<b>CASO DE USO 4:</b>	<b>Gestionar Parámetros.</b>	
<b>ACTOR:</b>	Administrador	
<b>PROPÓSITO:</b>	Desplegar al administrador un menú con las opciones propias crear, modificar y eliminar la información correspondiente a los parámetros de equipos de red.	
<b>RESUMEN:</b>	Este caso de uso permite visualizar al administrador las opciones para administrar los datos de los parámetros equipos de red del sistema.	
<b>PRECONDICIONES:</b>	Haber validado acceso. Haber ingresado información de equipos de red.	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	1. Desea gestionar la información de los parámetros de un equipo de red en particular. <b>E1</b>	2. Despliega las opciones correspondientes a la gestión de los parámetros de Equipos de red. <b>1</b>
<b>POSCONDICIONES:</b>	Las opciones relacionadas con la gestión de los parámetros de un Equipo de red son desplegadas.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno	
<b>NOTAS:</b>	En esta interfaz se presentan las opciones de la gestión de los parámetros de un equipo de red: Ingresar, modificar y eliminar parámetro.	
<b>EXCEPCIONES:</b>	<b>E1</b> Si el administrador decide cancelar la gestión de la información de un parámetro de un equipo de red, se devuelve a la interfaz de gestión de equipos de red.	

<b>CASO DE USO 5:</b>	<b>Ingresar Acuerdo de Nivel de Servicio</b>
<b>ACTOR:</b>	Administrador
<b>PROPOSITO:</b>	Permitirle al administrador ingresar un Acuerdo de nivel de Servicio al sistema con su respectiva información.
<b>RESUMEN:</b>	Este caso de uso permite al administrador ingresar nuevos

	Acuerdos de nivel de Servicio al sistema con su respectiva información.	
<b>PRECONDICIONES:</b>	Haber definido junto con el cliente con anterioridad los parámetros de los acuerdos de nivel de servicio. Haber Validado acceso. Estar dentro de la opción de Acuerdos de Nivel de Servicio.	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	1. Desea ingresar un nuevo Acuerdo de nivel de Servicio al sistema. 3. Llena el formato con los datos del Acuerdo de Nivel de Servicio. <b>E1</b> 4. Selecciona enviar los datos. <b>E1</b>	2. Despliega el formato de información y parámetros de los Acuerdos de Nivel de Servicio. <b>1</b> 5. Valida los datos. <b>E2</b> 6. Ingresa el nuevo Acuerdo de Nivel de Servicio al sistema. 7. Actualiza registro de datos.
<b>POSCONDICIONES:</b>	Un nuevo Acuerdo de Nivel de Servicio queda registrado en el sistema.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno.	
<b>NOTAS:</b>	1 El formato que se despliega contiene: Nombre del servicio o parámetro a analizar, Propósito, medio de verificación, tiempo de verificación, valor meta del parámetro o servicio.	
<b>EXCEPCIONES:</b>	<b>E1</b> Si el administrador decide cancelar el ingreso de un Acuerdo de Nivel de Servicio, se devuelve a la interfaz de administrador. <b>E2</b> Si el sistema verifica que el Acuerdo de Nivel de Servicio ya esta registrado o sucede algún error durante el proceso, despliega un mensaje de error y retorna al formato con los campos erróneos en blanco y el resto con la información correcta.	

<b>CASO DE USO 6:</b>	<b>Verificar cumplimiento de los Acuerdos de nivel de Servicio.</b>
<b>ACTOR:</b>	Administrador
<b>PROPOSITO:</b>	Permitirle al administrador u operador verificar el cumplimiento de los valores meta estipulados dentro de un Acuerdo de nivel de Servicio
<b>RESUMEN:</b>	Este caso de uso permite al administrador u operador verificar los valores meta y propósitos señalados dentro de los Acuerdos de nivel de Servicio usando al sistema como medio de análisis para la presentación de resultados.
<b>PRECONDICIONES:</b>	Haber Validado acceso. Haber ingresado con anterioridad un Acuerdo de Nivel de Servicio.

	Estar dentro de la opción de Acuerdos de Nivel de Servicio.	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	<p>1. Desea Verificar un Acuerdo de nivel de Servicio previamente ingresado al sistema.</p> <p>4. Selecciona un Acuerdo de Nivel de Servicio. <b>E1</b></p> <p>5. Selecciona una de las posibles opciones disponibles para la verificación de los Acuerdos de Nivel de Servicio. <b>E1</b></p>	<p>2. Despliega los Acuerdos de Nivel de Servicio previamente ingresados.</p> <p>3. despliega las opciones de verificación correspondientes a los Acuerdos de Nivel de Servicio. <b>1</b></p> <p>6. Despliega la interfaz correspondiente a la acción asociada a la opción elegida.</p>
<b>POSCONDICIONES:</b>	El cumplimiento de los valores metas de los Acuerdos de Nivel de Servicio es verificado.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno	
<b>NOTAS:</b>	1 las opciones de verificación son: Iniciar Verificación, Análisis de Datos recolectados y Comparación de Resultados.	
<b>EXCEPCIONES:</b>	<b>E1</b> Si el administrador decide cancelar la verificación de un Acuerdo de Nivel de Servicio, se devuelve a la interfaz de administrador.	

<b>CASO DE USO 6.1:</b>	<b>Iniciar Verificación del Acuerdo de nivel de Servicio.</b>	
<b>ACTOR:</b>	Administrador	
<b>PROPOSITO:</b>	Permitirle al administrador iniciar la verificación del cumplimiento de los valores meta estipulados dentro de un Acuerdo de nivel de Servicio.	
<b>RESUMEN:</b>	Este caso de uso permite al administrador dar inicio al proceso de verificación de los valores meta y propósitos señalados dentro de los Acuerdos de nivel de Servicio durante el periodo de verificación estipulado.	
<b>PRECONDICIONES:</b>	<p>Haber Validado acceso.</p> <p>Haber ingresado con anterioridad un Acuerdo de Nivel de Servicio.</p> <p>Estar dentro de la opción de Acuerdos de Nivel de Servicio.</p>	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>

	<p>1. Desea iniciar la Verificación de un Acuerdo de nivel de Servicio previamente ingresado al sistema.</p> <p>2. Selecciona un Acuerdo de Nivel de Servicio. <b>E1</b></p> <p>3. Selecciona la opción <b>Iniciar Verificación</b> de las opciones disponibles para la verificación de los acuerdos de Nivel de Servicio. <b>E1</b></p>	<p>4. Captura los parámetros correspondientes al acuerdo de nivel de Servicio seleccionado. <b>1</b></p> <p>5. Invoca el servicio encargado de desarrollar el proceso de captura de datos de los dispositivos de red de acuerdo a los parámetros capturados del acuerdo de Nivel de Servicio seleccionado. <b>E2</b></p> <p>6. El servicio invocado ejecuta el componente encargado de la recolección de datos de los equipos de red. <b>E2</b></p> <p>7. El servicio invocado ejecuta el componente encargado de almacenar los datos recolectados por el componente anterior. <b>E2</b></p>
<b>POSCONDICIONES:</b>	Se ha inicializado la verificación de un Acuerdo de Nivel de Servicio determinado.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno	
<b>NOTAS:</b>	<b>1</b> Los parámetros son: Ubicación del Servicio o dato a analizar, Especificación del dato a analizar.	
<b>EXCEPCIONES:</b>	<p><b>E1</b> Si el administrador decide cancelar la Inicialización de la verificación de un Acuerdo de Nivel de Servicio, se devuelve a la interfaz de administrador.</p> <p><b>E2</b> Si sucede algún error durante el proceso, despliega un mensaje de error y retorna a la interfaz de administrador.</p>	

<b>CASO DE USO 6.2:</b>	<b>Análisis de los datos recolectados.</b>
<b>ACTOR:</b>	Administrador / Operador
<b>PROPOSITO:</b>	Permitirle al administrador u operador hacer un análisis de los datos recolectados en los equipos de red correspondientes a un Acuerdo de Nivel de servicio y presentar resultados acordes a dicho análisis realizado.
<b>RESUMEN:</b>	Este caso de uso permite al administrador u operador dar inicio al proceso de análisis de los datos almacenados correspondientes a los equipos de red que prestan un servicio especificado en el Acuerdo de Nivel de Servicio durante el periodo de verificación estipulado.
<b>PRECONDICIONES:</b>	Haber Validado acceso. Haber ingresado con anterioridad un Acuerdo de Nivel de

	<p>Servicio. Estar dentro de la opción de Acuerdos de Nivel de Servicio. Haber terminado el proceso de verificación de un acuerdo de Nivel de Servicio descrito en el caso de Uso anterior.</p>	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	<p>1. Desea hacer el análisis de los datos almacenados de un equipo de red que presta un servicio señalado en los acuerdos de nivel de servicio.</p> <p>2. Selecciona un Acuerdo de Nivel de Servicio. <b>E1</b></p> <p>3. Selecciona la opción <b>Análisis de Datos Recolectados</b> de las opciones disponibles para la verificación de los acuerdos de Nivel de Servicio. <b>E1</b></p>	<p>4. Captura los parámetros correspondientes al acuerdo de nivel de Servicio seleccionado. <b>1</b></p> <p>5. Invoca el servicio encargado de desarrollar el proceso de análisis de los datos almacenados de acuerdo a los parámetros capturados del acuerdo de Nivel de Servicio seleccionado. <b>E2</b></p> <p>6. El servicio invocado ejecuta el componente encargado del proceso análisis de datos recolectados de los equipos de red. <b>E2</b></p> <p>7. El servicio invocado devuelve la información de análisis organizada en parámetros de análisis. <b>E2</b></p> <p>8. Muestra al usuario los datos, correspondiente al análisis realizado, de forma ordena según parámetros de análisis. <b>E2</b></p>
<b>POSCONDICIONES:</b>	Se presenta al usuario un análisis detallado correspondiente al Acuerdo de Nivel de Servicio seleccionado.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno	
<b>NOTAS:</b>	1 Los parámetros son: Ubicación del Servicio o dato a analizar.	
<b>EXCEPCIONES:</b>	<p><b>E1</b> Si el administrador decide cancelar la Inicialización del análisis de un Acuerdo de Nivel de Servicio, se devuelve a la interfaz de administrador.</p> <p><b>E2</b> Si sucede algún error durante el proceso, despliega un mensaje de error y retorna a la interfaz de administrador.</p>	

<b>CASO DE USO 6.3:</b>	<b>Comparación de Resultados.</b>
<b>ACTOR:</b>	Administrador / Operador
<b>PROPOSITO:</b>	Comparar los datos del análisis realizado a un Acuerdo de Nivel de Servicio con los datos estipulados en la creación de dicho

	Acuerdo de nivel de Servicio.	
<b>RESUMEN:</b>	Este caso de uso permite al administrador dar inicio al proceso de comparación de los valores meta y propósitos señalados dentro de los Acuerdos de nivel de Servicio con los datos obtenidos después del proceso de <b>Análisis de los datos recolectados</b> durante la verificación de un Acuerdo de Nivel de Servicio.	
<b>PRECONDICIONES:</b>	<p>Haber Validado acceso.</p> <p>Haber ingresado con anterioridad un Acuerdo de Nivel de Servicio.</p> <p>Estar dentro de la opción de Acuerdos de Nivel de Servicio.</p> <p>Haber realizado el proceso de <b>Análisis de datos recolectados</b> de Acuerdo de Nivel de Servicio descrito en el caso de Uso anterior.</p>	
<b>ESCENARIO</b>	<b>ADMINISTRADOR</b>	<b>SISTEMA</b>
	<p>1. Desea iniciar la Comparación de los valores meta estipulados en un Acuerdo de nivel de Servicio previamente ingresado al sistema y los obtenidos después del proceso de <b>Análisis de los datos recolectados</b> durante el proceso de verificación.</p> <p>2. Selecciona un Acuerdo de Nivel de Servicio. <b><u>E1</u></b></p> <p>3. Selecciona la opción <b>“Comparación de resultados”</b> de las opciones disponibles para la verificación de los acuerdos de Nivel de Servicio. <b><u>E1</u></b></p>	<p>4. Captura los parámetros correspondientes al acuerdo de nivel de Servicio seleccionado. <b><u>1</u></b></p> <p>5. Invoca el servicio encargado de desarrollar el proceso de comparación de los resultados obtenidos durante el proceso de análisis de los datos recolectados con los valores meta estipulados en la creación del Nivel de Servicio seleccionado. <b><u>E2</u></b></p> <p>6. El servicio invocado ejecuta el componente encargado de la comparación de los resultados obtenidos. <b><u>E2</u></b></p> <p>7. El servicio invocado devuelve un informe detallado de la comparación y las sugerencias pertinentes al resultado de dicha comparación para un Acuerdo de Nivel de Servicio específico. <b><u>E2</u></b></p> <p>8. El informe es presentado de manera ordenada al usuario por medio de una interfaz gráfica. <b><u>E2</u></b></p>
<b>POSCONDICIONES:</b>	Se muestra al usuario un informe detallado correspondiente a la comparación de resultados obtenidos para un Acuerdo de Nivel de Servicio determinado.	
<b>FLUJOS ALTERNATIVOS:</b>	Ninguno	

<b>NOTAS:</b>	<b>1</b> Los parámetros son: Ubicación del Servicio o dato a analizar.
<b>EXCEPCIONES:</b>	<b>E1</b> Si el administrador decide cancelar la Inicialización de la comparación de resultados para un Acuerdo de Nivel de Servicio, se devuelve a la interfaz de administrador. <b>E2</b> Si sucede algún error durante el proceso, despliega un mensaje de error y retorna a la interfaz de administrador.

#### 4.5 Construcción del Data Warehouse

Para comenzar con la construcción del Data Warehouse (DW) se debe tener claro de que se trata, cual es el objetivo, que elementos lo componen, etc. Tener una idea general de todo lo que enmarca el concepto de DW permite estar centrado en los aspectos importantes y relevantes a la hora de diseñarlo. A continuación se quiere retomar algunas de las características más importantes que identifican a un DW y utilizarlas de soporte teórico para la construcción del diseño.

Un DW es una base de datos que almacena información para la toma de decisiones. Dicha información es construida a partir de bases de datos que registran diferentes tipos de operaciones, que en el caso de la plataforma van a ser los datos extraídos de los equipos de red (bases operacionales). El objetivo de los DWs es consolidar información proveniente de diferentes bases operacionales y hacerla disponible para la realización de análisis de datos de tipo gerencial. Particularmente, la construcción del DW sirve como entrada al proceso de Data Mining.

En los Data Warehouse la prioridad es el acceso interactivo e inmediato a información estratégica de una determinada área. Las operaciones principales no son las transacciones, como en las bases de datos operacionales, sino consultas que involucran gran cantidad de datos y agrupaciones de los mismos. Las características de los DWs hacen que las estrategias de diseño para las bases de datos operacionales generalmente no sean aplicables para el diseño de DW ([19], [20]). Los modelos de datos para representar los datos almacenados en el DW también son diferentes.

A nivel conceptual resurgen los modelos multidimensionales [21], que representan la información como matrices multidimensionales o cuadros de múltiples entradas denominados  *cubos* . A los ejes de la matriz se los llama  *dimensiones*  y representan

los criterios de análisis, y a los datos almacenados en la matriz se los llama *medidas* y representan los indicadores o valores a analizar.

A nivel lógico, surgen implementaciones de los cubos tanto para bases de datos relacionales como multidimensionales. Para el caso de bases relacionales surgen nuevas técnicas y estrategias de diseño que apuntan esencialmente a optimizar la performance en las consultas introduciendo redundancia, lo cual eventualmente sacrifica la performance en las actualizaciones. Una de las tareas más importantes en la construcción de un DW es la construcción de su *esquema lógico*. El esquema lógico es una especificación más detallada que el esquema conceptual, donde se incorporan nociones de almacenamiento, performance y estructuración de los datos.

Un componente adicional a tener en cuenta son las bases fuentes. Un DW no es una base de datos para construir desde cero, sino que debe construirse con información extraída de un cierto conjunto de bases fuente. Durante el diseño lógico deben considerarse dichas bases y cómo se corresponden con el esquema conceptual. Por lo tanto es esencial poder relacionar los elementos del esquema conceptual con las tablas y atributos de las bases fuentes.

#### **4.6 Bases de Datos fuente**

Como se menciona anteriormente un Data Warehouse no se construye desde cero, deben existir unas bases fuentes que sirvan como punto de partida. Por lo general la organización interesada en implementar un sistema de DW ya cuentan con una sistema de información soportado en bases de datos operacionales. Particularmente, la aplicación no cuenta con una base de datos de las características necesarias para nuestra aplicación, debemos construirla de acuerdo a las necesidades del proyecto.

Como requisito para la construcción de la base fuente, hay que almacenar información referente al equipo de red que se desea gestionar, por ejemplo, ingresar el tipo y los parámetros de su(s) MIB('s). Esta operación la debe realizar una persona que tenga el suficiente conocimiento sobre el tema, la cual, para la aplicación es conocida como administrador.

##### **4.6.1 Tablas**

Se tiene básicamente las siguientes tablas:

- NetDevice, almacena información referente al dispositivo de red, llámese servidor Proxy o Switch. Esta conformada por los siguientes atributos: ID\_Device, Name, Device\_Type, IPAddress y Switch\_Port.
- OID, contiene información de gestión específica de un dispositivo de red. Existen tipos de MIB's especiales para cada agente que se desea gestionar<sup>9</sup>. Los atributos que conforman esta tabla son: ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port y OID\_Value.

#### 4.7 Propuesta para el proceso de diseño de un Data Warehouse

El desarrollo de los Sistemas de Información Gerencial<sup>10</sup> útiles para la toma de decisiones basados en tecnologías de Data Warehouse, es relativamente reciente y, por lo tanto, no existe una propuesta metodológica universalmente válida y aceptada como tal, por la comunidad académica. Entre las propuestas más conocidas están: 1. Ralph Kimball[19], con un esquema centrado en la identificación de los procesos de la empresa, como elemento clave para la definición de la estructura de variables y dimensiones; 2. W.H. Inmon[20], con un esquema que parte de la construcción del modelo de datos corporativos, elaborado al más alto nivel de abstracción, para luego derivar la estructura del modelo de datos, para el diseño de la bodega; 3. Golfarelli Matteo, Maio Dario, Rizzi Stefano[22], proponen un esquema que parte de los modelos Entidad - Relación descriptivos de los sistemas transaccionales de la organización, para luego derivar el modelo Entidad - Relación de la estructura, para la bodega de datos; y muchos más, otros incluso centran su atención en partes específicas de la construcción del Data Warehouse, por ejemplo, en el diseño conceptual [21], donde se propone la utilización del modelo CMDM para la construcción de un modelo conceptual que especifique bases multidimensionales; o en el diseño lógico [23], basado en esquemas conceptuales multidimensionales y en una base de datos fuente integrada que los utilizan para diseñar el esquema lógico relacional del Data Warehouse aplicando transformaciones basadas en un conjunto de reglas de diseño.

En este trabajo de grado se presenta una propuesta de sistematización del proceso de diseño de un Data Warehouse, que se aparta de los esquemas de diseño referidos, y que utiliza como eje articulador, la identificación de información relevante, para el

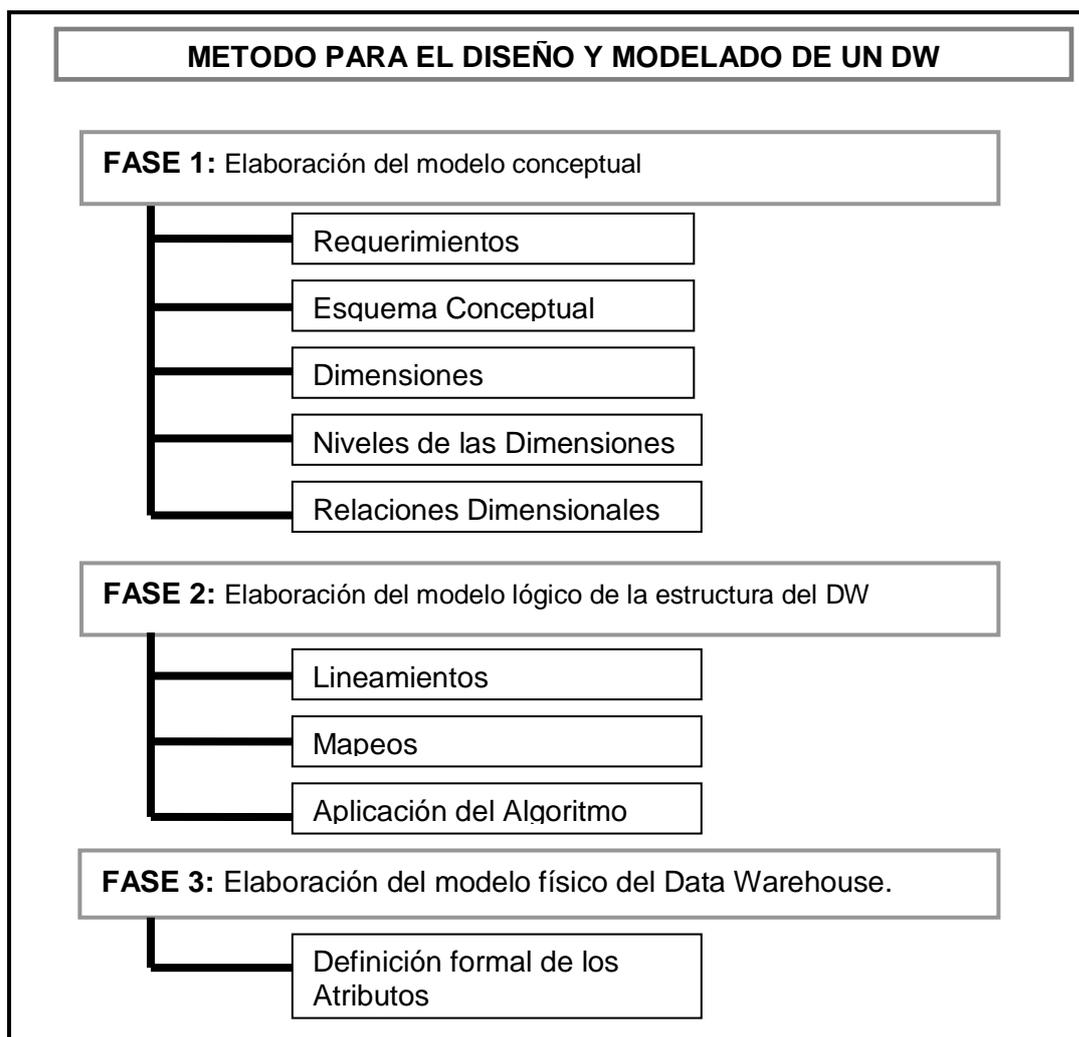
---

<sup>9</sup> Ver información detallada a cerca de las MIB's y OID's en el Anexo C.

<sup>10</sup> Por definición se entiende como Sistema de Información Gerencial al método de poner a disposición de los gerentes la información confiable y oportuna que se necesite para facilitar el proceso de toma de decisiones y permitir que las funciones de planeación, control y operaciones se realicen eficazmente en la organización.

soporte de los procesos de control y de toma de decisiones en los niveles administrativos de la Red de Datos. El método de diseño propuesto está centrado en la identificación de la información clave y relevante para soportar los procesos de dirección y de toma de decisiones en cuanto al comportamiento y estado de la Intranet.

Este método utiliza, como punto de partida, la identificación y el modelado de: qué es lo que se está tratando de alcanzar, para luego elaborar una estructura que apoye el proceso de gestión hacia el logro de las metas definidas.



**Figura 4.5 Método para el diseño y modelado de un DW**

Una vez que la información clave de apoyo a los procesos de gestión y control de la organización ha sido identificada, se inicia la elaboración del modelo conceptual y lógico de la estructura del Data Warehouse, a partir del cual se construirán las bases de datos (proceso de diseño físico) según los indicadores de gestión o parámetros

requeridos por los niveles administrativos de la red. Para darle un orden a este proceso sistémico de diseño, los pasos del método propuesto, tal como se presentan en la Figura 4.5, se han agrupado en las siguientes fases:

- Fase 1: Elaboración del modelo conceptual según las necesidades identificadas, desde la perspectiva de la Red de Datos.
- Fase 2: Elaboración del modelo lógico de la estructura del Data Warehouse.
- Fase 3: Elaboración del modelo físico del Data Warehouse.

#### **4.7.1 Fase 1: Elaboración del modelo conceptual**

La primera fase, a partir de la cual se realiza el proceso de diseño de la estructura para un Data Warehouse, comprende la identificación de las necesidades del administrador de la red para definir unos correctos Acuerdos de Nivel de Servicio con su proveedor. La información necesaria debe quedar estipulada según unos requerimientos administrativos. Es bastante común empezar este proceso de identificación y modelado mediante entrevistas a los administradores, en las cuales la pregunta central es: ¿Cuál es la información que desea obtener del sistema?.

##### **4.7.1.1 Requerimientos**

Se identificaron los siguientes requerimientos, según las necesidades básicas actuales de la Red de Datos de la Universidad del Cauca:

- Realizar un seguimiento más eficiente al Switch principal del backbone de la Red de Datos y a uno de sus servidores Proxy
- Determinar los periodos de tiempo donde la congestión de tráfico que circula por el servidor Proxy HIPERION y el Switch ACCELAR es mayor.
- Identificar, a nivel de intranet, que subred conectada al switch ACCELAR esta produciendo mayor congestión y durante que periodos de tiempo lo están haciendo.

##### **4.7.1.2 Esquema conceptual**

Una vez se tiene claro que tipo de información es la que se desea, es necesario construir el esquema conceptual. Se utilizará el modelo conceptual multidimensional CMDM definido por Carpani en [21]. A continuación se presenta un esquema conceptual diseñado a partir de los requerimientos.

#### 4.7.1.3 Dimensiones

A partir de los requerimientos, hay ciertas dimensiones que parecen claras, aunque podría no estar clara la estructura interna de cada dimensión. En principio, se construye una dimensión para cada objeto que participa en el problema.

De esta forma, las dimensiones que se obtienen inicialmente son:

- Dispositivos de red
- Ubicación
- Tiempo
- SLA's

La dimensión *Dispositivos de Red* surge por la necesidad de poder identificar los diferentes elementos de red que pueden participar en el análisis, servidores o switches, pero puede ser cualquier otro dispositivo. La dimensión *Ubicación* permite saber la ubicación, dentro del campus universitario, de la subred conectada al switch. La dimensión *Tiempo* sirve para representar diferentes periodos de tiempo, lo cual es bastante útil puesto que se puede analizar los dispositivos por hora, jornadas, días, semanas o meses. Por último se encuentra la dimensión de medida *SLA's*, esta dimensión se utiliza como medio de comprobación para los acuerdos de nivel de servicio según los parámetros seleccionados para cada equipo.

#### 4.7.1.4 Niveles de las Dimensiones

Para determinar los niveles iniciales de las dimensiones hay que considerar que cada dimensión debe tener un nivel "mínimo" en donde exista un identificador para cada elemento de la dimensión. Luego, los niveles restantes se construyen de forma que la relación entre un nivel y el superior sea N:1 ya que cada nivel debe agregar varios elementos del nivel inferior. Para determinar en qué nivel se pone un determinado atributo, debe utilizarse la información sobre los requerimientos de información. Según nuestros requerimientos no existen muchos niveles por cada dimensión, en cambio se tienen grandes cantidades de datos por dimensión. Las siguientes gráficas presentan las dimensiones con sus respectivas dimensiones.

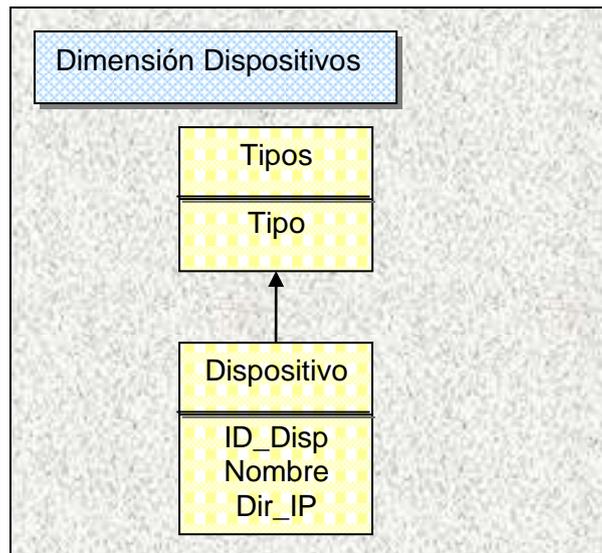


Figura 4.6 Dimensión Dispositivos

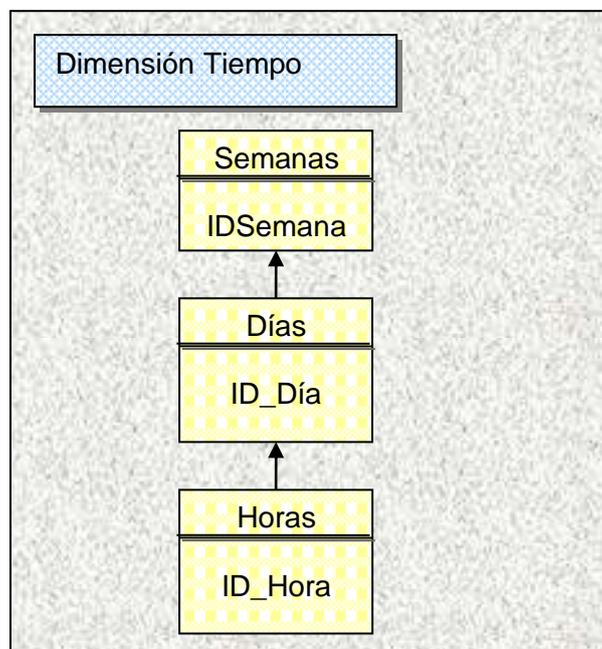


Figura 4.7 Dimensión Tiempo

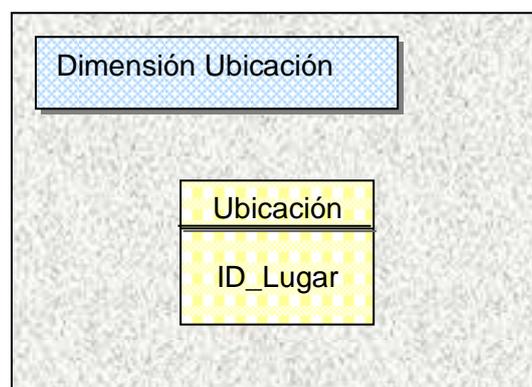


Figura 4.8 Dimensión Ubicación

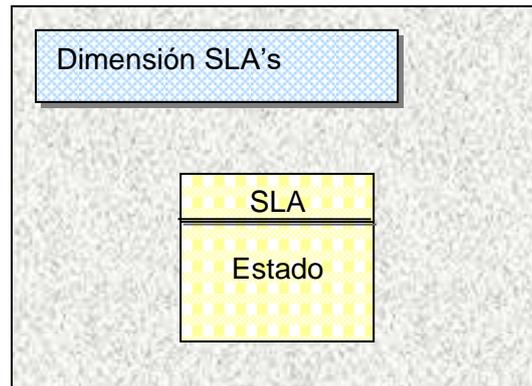


Figura 4.9 Dimensión Tráfico

#### 4.7.1.5 Relaciones Dimensionales

Las relaciones dimensionales representan el conjunto de cubos que se pueden construir tomando al menos un nivel de las dimensiones participantes. De acuerdo con los requerimientos planteados, las relaciones dimensiones son básicamente dos: la primera de ellas se puede ver en la figura 4.10 y esta enfocada hacia los acuerdos de nivel de servicio que están relacionados con las dimensiones, *Dispositivos de Red*, *Tiempo* y *SLA's*, la segunda (ver figura 4.11), por supuesto también tiene que ver con los acuerdos de nivel de servicio, pero esta vez relacionando las cuatro dimensiones *Dispositivos de Red*, *Ubicación*, *Tiempo* y *SLA's*. Los cubos de interés son dos, uno por cada dimensión.

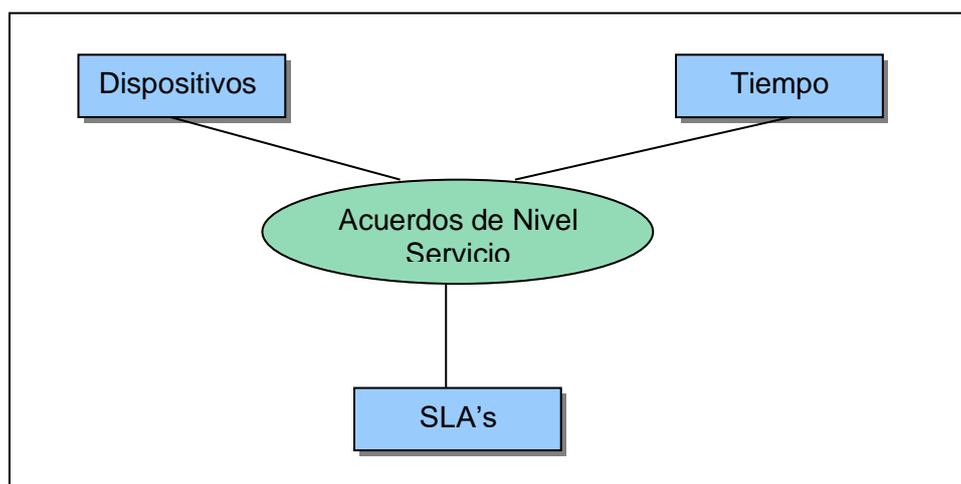


Figura 4.10 Relación Dimensional SLA 1

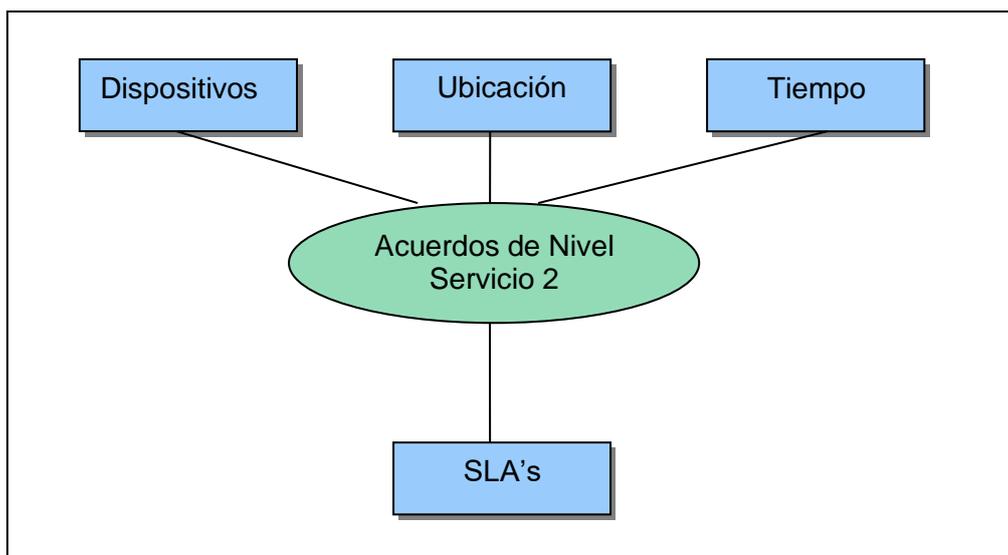


Figura 4.11 Relación Dimensional SLA 2

#### 4.7.2 Fase 2: Elaboración del modelo lógico

Para seguir con el proceso de construcción del DW hay que realizar un modelo lógico a partir del modelo conceptual construido con CMDM en la fase anterior. En esta fase específicamente se desea transformar un esquema conceptual multidimensional a un esquema relacional [23].

El esquema conceptual se complementa con lineamientos de diseño que abstraen estrategias de diseño de DWs y restricciones de performance y almacenamiento para el mismo. Los lineamientos nos permiten elegir la estrategia que mejor se adecua a su problema concreto. Se debe además especificar correspondencias o mapeos entre el esquema conceptual y la fuente, para saber de dónde se obtienen los datos para cada elemento del esquema conceptual.

La metodología propuesta consiste en la aplicación de transformaciones de esquema a las bases fuentes, hasta lograr un esquema lógico que se adecue a los requerimientos expresados a través del esquema conceptual y los lineamientos. Como transformaciones básicas se utiliza una extensión al conjunto de *Transformaciones de Esquemas* propuestas por Marotta en [24]. Dichas transformaciones construyen paso a paso el DW generando una traza de aplicación que permite documentar el proceso y facilita su evolución.

La elección de que transformaciones aplicar puede ser muy amplia, y cada problema puede ser eventualmente resuelto por muchos caminos de aplicación. Algunas de

esos caminos llevan a mejores soluciones que otros, tanto desde el punto de vista de la calidad del esquema resultado, como de la performance en su construcción.

#### 4.7.2.1 Lineamientos

Los lineamientos son información de diseño que complementan al modelo conceptual y a las bases fuentes, y permiten al diseñador dar pautas sobre el esquema deseado para el DW.

Hay 3 tipos de lineamientos: materialización de relaciones, fragmentación de dimensiones y fragmentación de cubos. La materialización de relaciones permite indicar qué cubos se quieren materializar, atendiendo a los requerimientos de performance y almacenamiento. La fragmentación de dimensiones permite elegir el estilo de diseño deseado para el DW, esto incluye obtener un esquema estrella, snowflake, o estrategias intermedias, en este último caso indicando que dimensiones denormalizar, normalizar o fragmentar. La fragmentación de cubos permite almacenar por separado datos históricos, o dividir la instancia de los cubos de acuerdo a criterios del diseñador. A continuación se presentan los lineamientos definidos para el proyecto.

##### 4.7.2.1.a Materialización de relaciones

Debido a la cantidad de parámetros disponibles que se pueden obtener desde las MIB's del agente y a la concepción del modelo lógico para cada uno de ellos que se construye de igual forma, se elige materializar solamente un cubo para cada relación dimensional. A manera de ejemplo, se construye de forma completa los dos diferentes tipos de cubos que se identifican de las relaciones dimensionales, un cubo que relacione las dimensiones *Dispositivos*, *Tiempo* y *SLA's* y otro cubo donde intervienen todas las dimensiones. Con estos dos cubos se tienen los patrones necesarios para construir todos los acuerdos de nivel de servicio, de forma tal, que solamente cambien algunos niveles de detalle (horas, días y semanas) para la dimensión *Tiempo* y el dispositivo al cual se le extrae información (proxy o switch).

Se materializa un cubo para la relación dimensional Acuerdos de Nivel de Servicio 1:

- Con detalle de Tipo, Hora y Tráfico,

Y se materializa un cubo para la relación dimensional Acuerdos de Nivel de Servicio 2:

- Con detalle de Tipo, Lugar, Hora y Tráfico

La Figura 4.12 muestra la representación gráfica de los cubos. El nombre está dentro del cubo, y entre paréntesis el nombre de la relación que materializa. Los rectángulos

amarillos representan los niveles de detalle. Las medidas corresponden al nivel marcado por una flecha.

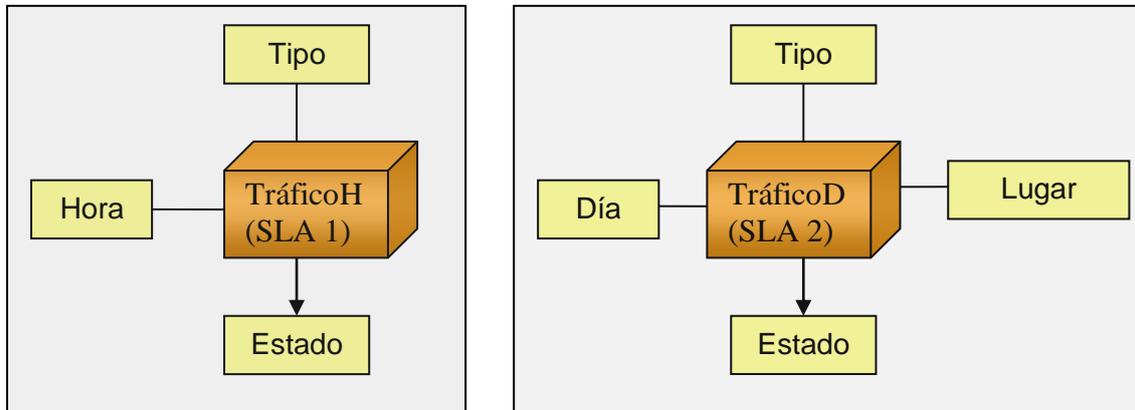


Figura 4.12 Cubos

#### 4.7.2.1.b Fragmentación de cubos

Recordemos que la fragmentación de cubos permite hacer explícito los rangos históricos de los datos que se desea almacenar. Según los cubos anteriores se decide fragmentar los cubos de la siguiente manera:

- Una banda<sup>11</sup> para el tráfico cursado entre las 8 a.m y las 8 p.m. Ver figura 4.13.
- Una banda para el tráfico cursado de lunes a sábado. Ver figura 4.14

Las Figuras 4.13 y 4.14 muestran la representación gráfica de las bandas definidas. Las bandas se indican en la llamada mediante predicados.

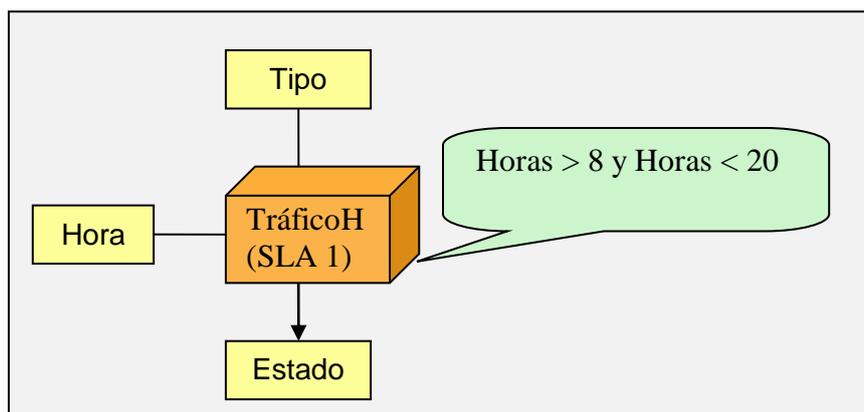
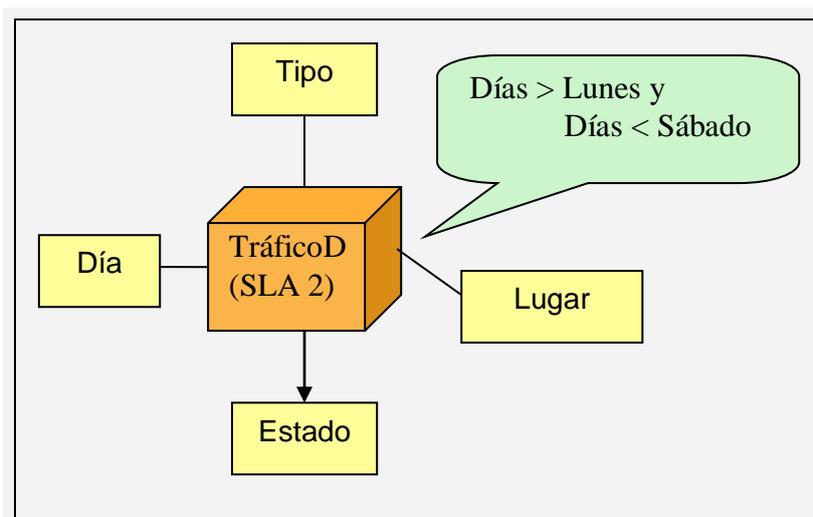


Figura 4.13 Fragmentación del cubo TráficoH

<sup>11</sup> Las bandas o franjas se expresan en términos de los ítems de los niveles del cubo, y se expresarán en términos de atributos de tablas en una etapa posterior.



**Figura 4.14 Fragmentación del cubo TráficoD**

#### 4.7.2.1.c Fragmentación de dimensiones

La forma de fragmentar las dimensiones tiene que ver con el estilo de diseño que se quiere dar al DW. El estilo de diseño escogido es un esquema en estrella definiendo un único fragmento por cada dimensión (con todos los niveles). En [24], Marotta define un conjunto de transformaciones de esquemas que permite llegar a diferentes estilos de diseño, entre ellos el esquema estrella, según los requerimientos de performance y almacenamiento del problema.

#### 4.7.2.2 Mapeos

Un mapeo es una función que muestra como se corresponden los objetos del modelo conceptual con la base fuente.

Los mapeos son funciones que asocian a cada item de un objeto del esquema conceptual una expresión de mapeo, construida en base a las tablas y atributos de la fuente. Estas funciones las representaremos gráficamente como flechas direccionales que van desde los objetos del modelo conceptual hacia las tablas de la base de datos fuente, pero internamente (dentro de la aplicación) las funciones se realizan por medio de líneas de código. Se tiene una función de mapeo para cada fragmento de dimensión, y una función de mapeo para cada cubo.

Una expresión de mapeo puede ser un atributo de una tabla fuente (mapeo directo, gráficamente lo representamos mediante un flecha continua), o un cálculo que involucra por lo menos un atributo de una tupla (cálculo simple, gráficamente se verá

como una flecha discontinua), o una totalización que involucra varios atributos de varias tuplas (cálculo agregado) o algo externo a las fuentes como una constante.

#### 4.7.2.2.a Mapeos de fragmentos

A continuación se presentan los mapeos definidos para los fragmentos de las dimensiones. Las figuras 4.15, 4.16 y 4.17 muestran los mapeos de fragmentación según la fragmentación de dimensiones (fragmentación única para cada dimensión).

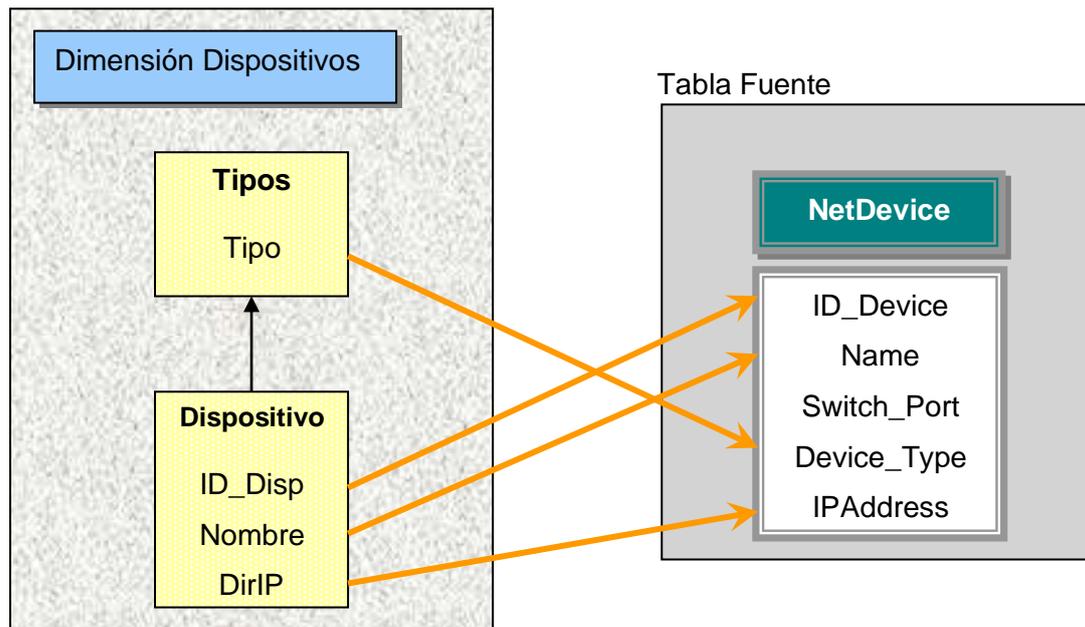


Figura 4.15 Mapeo del fragmento único de la dimensión dispositivos

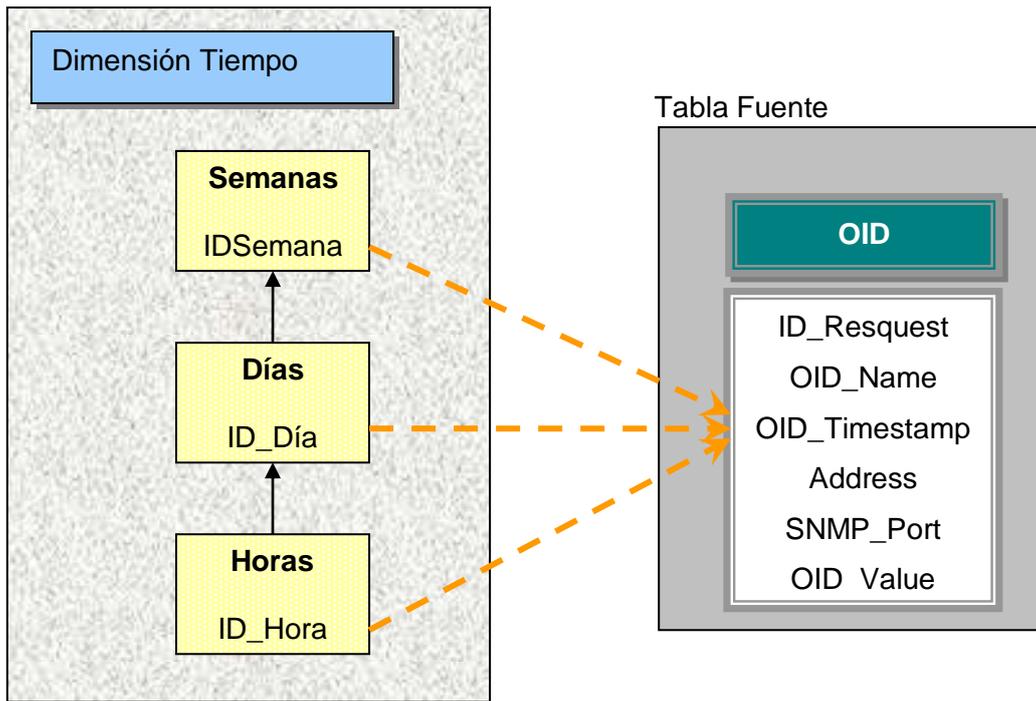


Figura 4.16 Mapeo del fragmento único de la dimensión tiempo

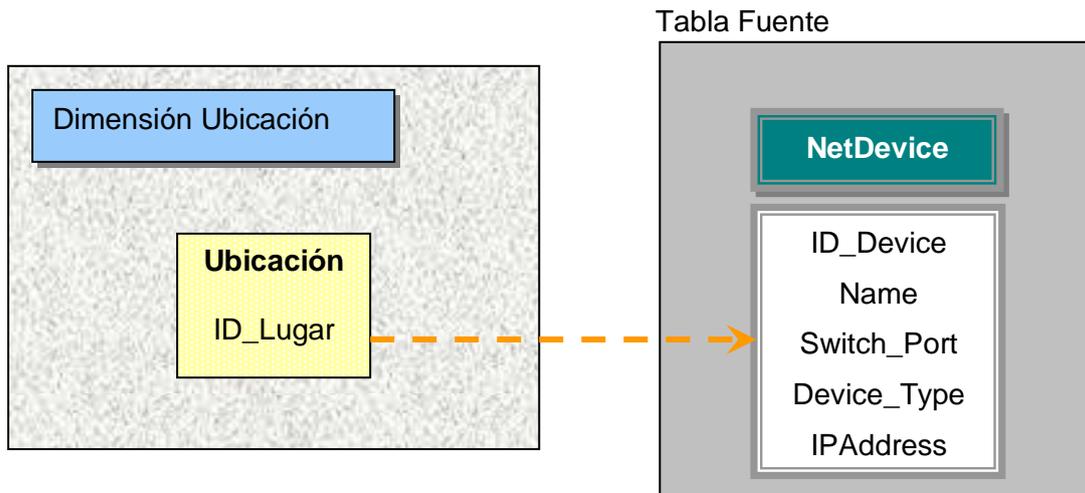


Figura 4.17 Mapeo del fragmento único de la dimensión Ubicación

**4.7.2.2.b Mapeos de cubos**

A continuación se presentan los mapeos definidos para los cubos. La figura 4.18 representa el mapeo del cubo TráficoH, donde el nivel *Estado* de la dimensión *SLA's* esta siendo mapeada hacia la tabla fuente *OID* y más específicamente al atributo *OID\_Value*. La figura 4.19 muestra el mapeo del cubo TráficoD, que es básicamente la misma gráfica anterior puesto que la dimensión de medida *SLA's* es igual en ambos casos. Con el valor de una *OID* se realiza un proceso interno que permite obtener datos, según las dimensiones y los niveles que participan en el acuerdo de nivel de servicio, para determinar cual es estado del mismo.

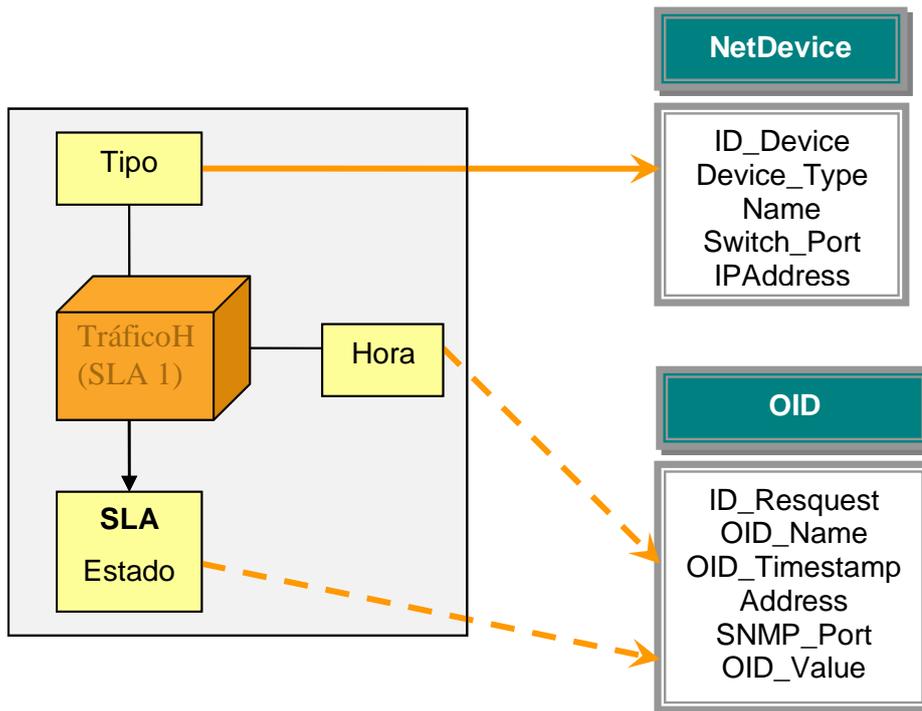


Figura 4.18 Mapeo del cubo TráficoH

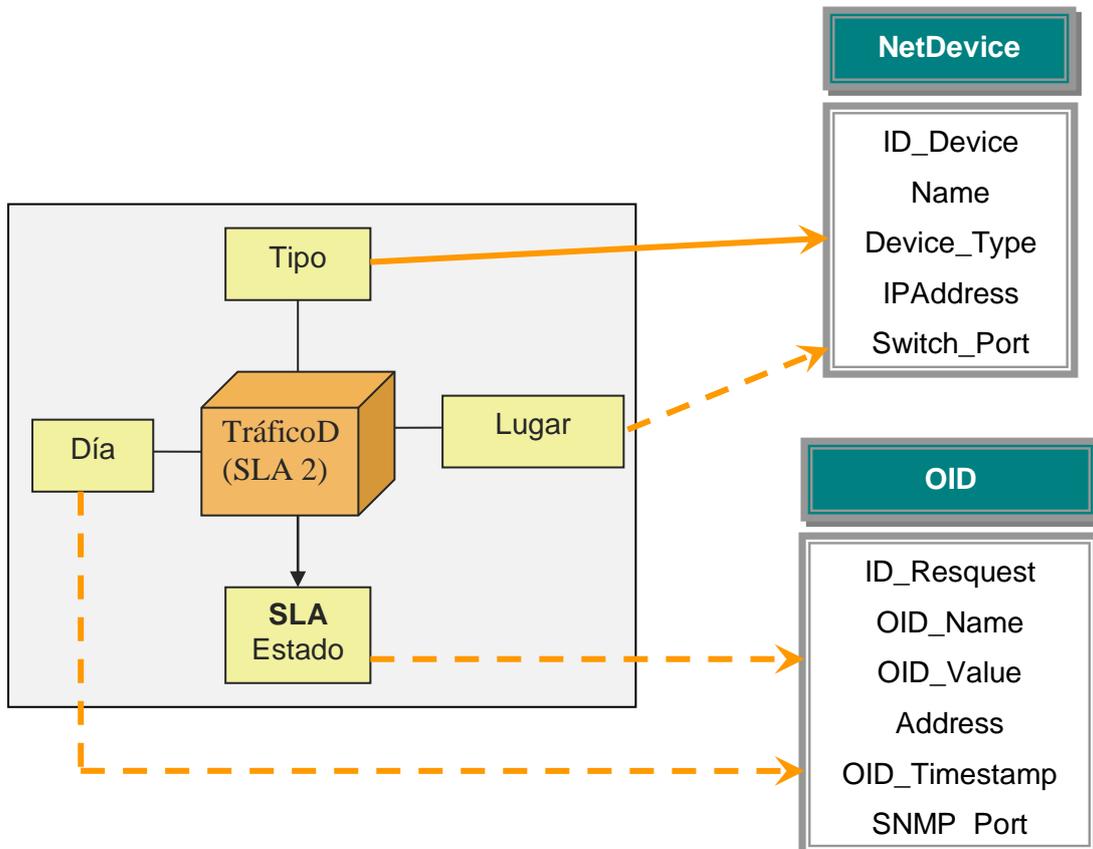


Figura 4.19 Mapeo del cubo TráficoD

### 4.7.2.3 Aplicación del algoritmo

A partir del esquema intermedio y los mapeos se va transformando el esquema fuente aplicando las primitivas de transformación de esquemas definidas por Marotta en [24]. La elección de cuando corresponde aplicar las primitivas y los parámetros adecuados se hace en base a reglas de transformación.

En [23] se presenta un algoritmo que da un orden a las reglas.

A continuación se muestra como será para el proyecto la aplicación de los pasos del algoritmo y las tablas que se generan como resultado. Como el modelo conceptual multidimensional no resulto ser tan complejo, la aplicación del algoritmo es relativamente fácil.

### CONSTRUCCIÓN DE TABLAS DE DIMENSIÓN

Para la construcción de las tablas de dimensión se aplican los pasos 1 a 6 para cada fragmento de dimensión.

#### **Paso 1 – Construir el esqueleto**

Se construyen las tablas de dimensión según todos los argumentos de la(s) tabla(s) que mapee desde la base de datos fuente. Se tienen entonces las siguientes tablas:

- *DwDispositivos* (ID\_Device, Name, Device\_Type, IPAddress, Switch\_Port)
- *DwTiempo* (ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value)
- *DwUbicación* (D\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value)

#### **Paso 2 – Renombrar atributos para ítems con mapeo directo**

La idea es unificar nombres, hacer coincidir los nombres de las bases fuentes con los niveles de las dimensiones. Las tablas *DwTiempo* y *DwUbicación* no sufren ninguna modificación puesto que todos sus mapeos son calculados. La tabla que tiene mapeo directo queda entonces de la siguiente manera:

- *DwDispositivos* (ID\_Dispositivo, Nombre, Tipo, IPAddress, Switch\_Port)

#### **Paso 3 – Generar atributos para ítems con mapeo calculado**

El resultado de un mapeo calculado es un atributo que no pertenece a la base fuente, por tal motivo debe ser agregado a las tablas. Las tablas que no

sufrieron ninguna modificación en el paso anterior, ahora van a cambiar de la siguiente manera:

- *DwTiempo* (ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value, ID\_Hora, ID\_Día, ID\_Semana)
- *DwUbicación* (ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value, ID\_Lugar)

#### **Paso 4 – Aplicar filtros**

La aplicación de filtros tiene como objetivo considerar las condiciones de mapeo que se han aplicado en el momento del diseño a las bases fuentes. Como en ninguna de nuestras dimensiones hemos aplicado condiciones en los mapeos entonces las tablas permanecen iguales en este paso.

#### **Paso 5 – Eliminar atributos sobrantes**

Como su nombre lo indica, en este paso se busca tener las tablas dimensionales únicamente con los atributos que nos interesan. Nuestras tablas quedan de la siguiente forma:

- *DwDispositivos* (ID\_Dispositivo, Nombre, Tipo)
- *DwTiempo* (ID\_Hora, ID\_Día, ID\_Semana)
- *DwUbicación* (ID\_Lugar)

#### **Paso 6 – Ajustar las claves**

Este paso es importante cuando se tiene un esquema de diseño diferente al esquema estrella, puesto que aquí importa identificar bien las claves primarias cuando se tiene diferentes fragmentos dentro de una misma dimensión. Como todas nuestras dimensiones tienen fragmentación única este paso dejamos las tablas de igual forma.

### CONSTRUCCIÓN DE TABLAS DE HECHOS PARA LOS CUBOS

Para la construcción de las tablas de hechos se aplican los pasos 7 a 12 para cada cubo.

#### **Paso 7 – Construir el esqueleto**

Igualmente como en el paso numero 1, se busca construir una tabla donde quede consignado todos los atributos de las bases fuentes que fueron utilizadas para realizar el mapeo. La tabla de hechos para los dos cubos en este primer paso queda de la siguiente manera:

- *DwTraficoH* (ID\_Device, Name, Device\_Type, IPAddress, Switch\_Port, ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value)
- *DwTraficoD* (ID\_Device, Name, Device\_Type, IPAddress, Switch\_Port, ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value)

### **Paso 8 – Renombrar atributos para items con mapeo directo**

El único mapeo directo que hay en los dos cubos es el tipo de dispositivo, de tal forma que solamente cambia ese atributo.

- *DwTraficoH* (ID\_Device, Name, Tipo, IPAddress, Switch\_Port, ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value)
- *DwTraficoD* (ID\_Device, Name, Tipo, IPAddress, Switch\_Port, ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value)

### **Paso 9 – Generar atributos para items con mapeo calculado**

A las tablas de hecho se le agregan los nuevos atributos y toman la siguiente forma:

- *DwTraficoH* (ID\_Device, Name, ID\_Tipo, IPAddress, Switch\_Port, ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value, ID\_Hora, ID\_Estado)
- *DwTraficoD* (ID\_Device, Name, ID\_Tipo, IPAddress, Switch\_Port, ID\_Request, OID\_Name, OID\_Timestamp, Address, SNMP\_Port, OID\_Value, ID\_Días, ID\_Lugar, Estado)

### **Paso 10 – Aplicar filtros**

Los cubos no tienen condiciones en el mapeo por lo que no se aplica la regla de filtros.

### **Paso 11 – Eliminar atributos sobrantes**

Las tablas de hechos quedan de la siguiente forma:

- *DwTraficoH* (ID\_Tipo, ID\_Hora, Estado)
- *DwTraficoD* (ID\_Tipo, ID\_Días, ID\_Lugar, Estado)

### 4.7.3 Fase 3: Elaboración de la estructura física del DW

Durante esta última fase, se realiza la transformación del modelo lógico-conceptual en la estructura física [25], que posteriormente será implementada para ser utilizada por nuestro módulo de minería de datos.

El modelo en estrella que conforma la estructura lógica propuesta para el Data Warehouse, debe ser convertido en una estructura totalmente desnormalizada, tal como se presenta en la Figura 4.20. Este modelo físico está conformado por una tabla de hechos (Nombre\_Cubo), y por las entidades en las cuales se almacenarán los dominios de las dimensiones con sus correspondientes niveles.

Para esta fase vamos a trabajar de una forma general, sin definir específicamente el área del acuerdo de nivel de servicio, ni tampoco el cubo que se desea materializar (Nombre específico del acuerdo de nivel de servicio). Vamos a trabajar con todas las dimensiones pero sabemos de antemano que para algunos SLA la dimensión *Ubicación* no debe ser tomada en cuenta. Esta fase presenta esta facilidad puesto que todos los SLA's tienen básicamente la misma estructura, lo único que se debe hacer es cambiar los nombres de la relación dimensional por el área donde se encuentra ubicada el SLA y colocar el nombre exacto del cubo.

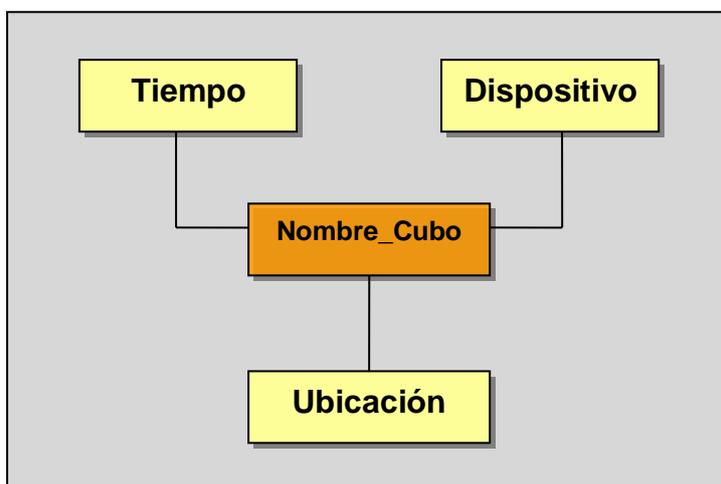


Figura 4.20 Modelo físico del DW

#### 4.7.3.1 Definición formal de los atributos que conforman la tabla de hechos y las dimensiones del modelo.

En este paso final, se identifican para la tabla de hechos y para cada dimensión las características de los atributos que conforman cada estructura.

Nombre de la relación multidimensional	Área del Acuerdo de nivel de servicio	
Tabla de hechos	Nombre_Cubo	
Atributos	Tipo	Descripción
ID_Dispositivo	Varchar	Identificador de la dimensión Dispositivos
ID_Tiempo	Varchar	Identificador de la dimensión Tiempo
ID_Lugar	Varchar	Identificador de la dimensión Ubicación
Estado	Boolean	Valor de la tabla de hechos

**Tabla 4.1 Definición de los atributos de la tabla de hechos sobre SLA**

Nombre de la relación multidimensional	Área del Acuerdo de nivel de servicio	
<b>Dimensión</b>	Dispositivos	
Atributos	Tipo	Fuente
ID_Dispositivo	Numérico	
Nombre	Varchar	
DirIP	Varchar	

**Tabla 4.2 Definición de los atributos de la dimensión Dispositivos**

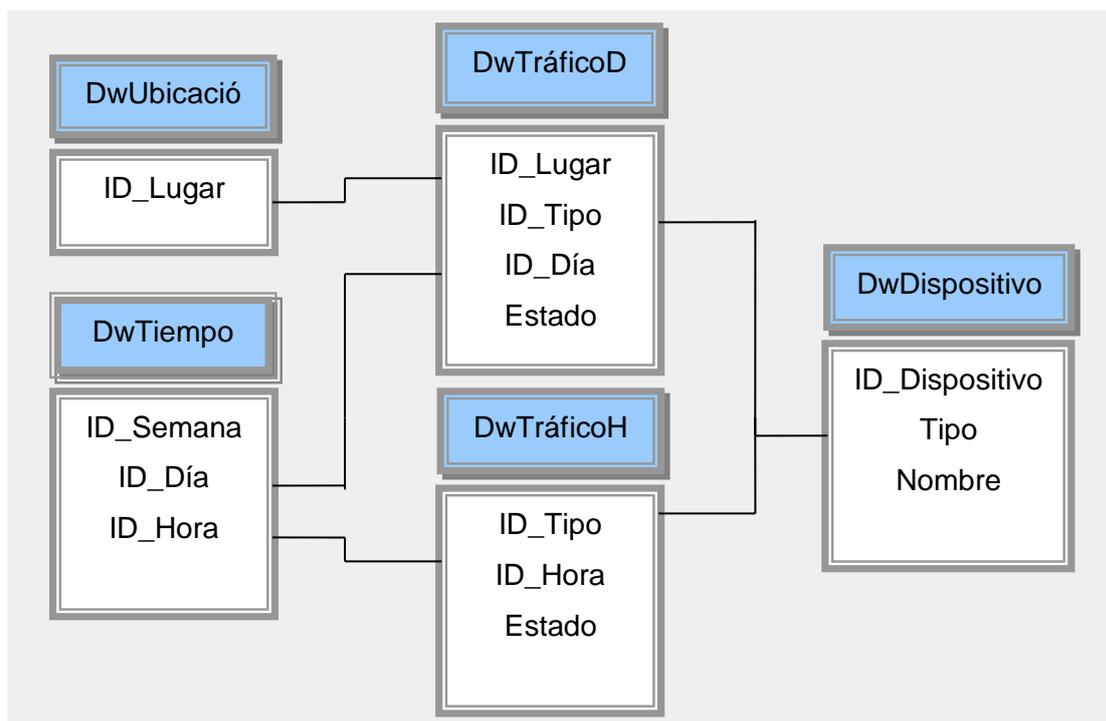
Nombre de la relación multidimensional	Acuerdos de nivel de servicio 1	
<b>Dimensión</b>	Tiempo	
Atributos	Tipo	Fuente
ID_Semana	Numérico	
ID_Día	Varchar	
ID_Hora	Numérico	

**Tabla 4.3 Definición de los atributos de la dimensión Tiempo**

Nombre de la relación multidimensional		Acuerdos de nivel de servicio 1
Dimensión		Ubicación
<b>Atributos</b>	<b>Tipo</b>	<b>Fuente</b>
ID_Lugar	Varchar	

**Tabla 4.4 Definición de los atributos de la dimensión Ubicación**

Como resultado de esta tercera fase tenemos listas las tablas de nuestro DW con sus respectivos atributos. La siguiente figura nos ilustra como se ven las tablas para ser utilizadas por el Data Mining.



**Figura 4.21 Esquema final del Data Warehouse**

#### 4.8 Construcción de la Minería de datos

La minería de datos se enmarcan en el proceso completo de extracción de información conocido como KDD<sup>12</sup>, que se encarga además de preparación de los datos y de la interpretación de los resultados obtenidos. La figura 4.22 muestra como la minería de datos es parte fundamental dentro del proceso de descubrimiento del conocimiento.

Las primeras fases del KDD determinan que las fases sucesivas sean capaces de extraer conocimiento válido y útil a partir de la información original. La primera fase del proceso es la recolección de los datos, seguida de una fase de selección, limpieza y transformación de datos. Estas fases se pueden ver en la figura anterior representadas dentro del segmento azul. El objetivo que se busca con estas primeras etapas es dejar unos datos bien estructurados y organizados, de tal forma que sea una entrada de datos confiable hacia la etapa de minería de datos.

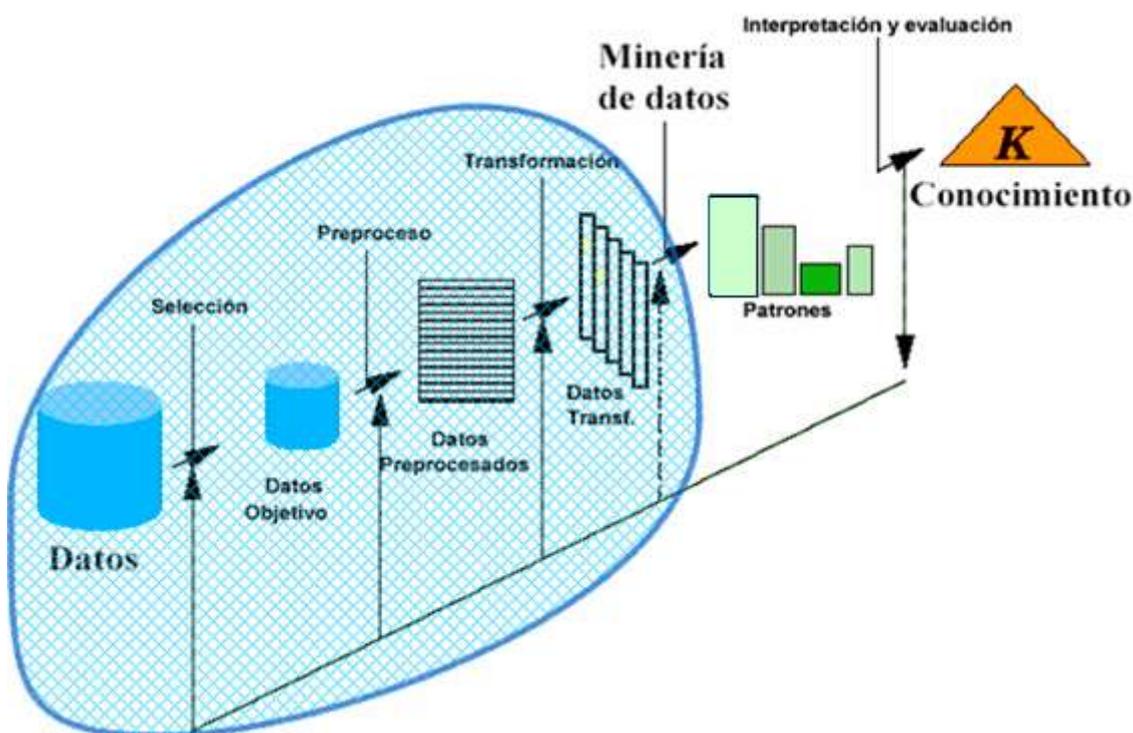


Figura 4.22 Proceso de descubrir el conocimiento en bases de datos KDD

En el capítulo anterior definimos el ciclo de vida para un proyecto data mining que consiste en seis fases: Determinación de objetivos, Preparación de datos,

<sup>12</sup> Proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos.

Transformación de datos, Minería de datos, Análisis de resultados y Asimilación de conocimiento. La secuencia de las fases no es necesariamente una camisa de fuerza, es más bien un proceso flexible y altamente iterativo. Además, no todos los pasos requieren el mismo esfuerzo, generalmente la etapa de preprocesamiento es la más costosa ya que representa aproximadamente el 60 % del esfuerzo total, mientras que la etapa de minería sólo representa el 10%. La etapa de preprocesamiento mencionada es prácticamente el mismo segmento azul que se muestra en la figura 4.22, podemos decir entonces, que indistintamente del método seleccionado para la construcción de un data mining, las etapas previas a la aplicación de un algoritmo de minería de datos buscan el mismo objetivo; “acomodar” los datos fuentes.

En las secciones anteriores en este capítulo se ha: determinado unos objetivos, construido una base de datos fuente, recolectado, preparado y transformado los datos, con el fin de construir un Data Warehouse. Pero todas estas fases cumplen y hacen lo mismo que las etapas de preprocesamiento en un ciclo de vida de un proyecto de minería de datos. Por tal motivo, se puede decir que este preprocesamiento ha sido abarcado por la anterior etapa de Data Warehouse.

Para completar el proceso de minería de datos se debe seguir con los siguientes pasos: Determinar los patrones a descubrir, el tipo de conocimiento, el algoritmo a utilizar e interpretar los resultados obtenidos.

#### **4.8.1 Patrones a descubrir**

Una vez recogidos los datos de interés, se debe decidir qué tipo de patrón quiere descubrir. Para determinar el patrón hay que tener en cuenta básicamente dos formas para buscar el conocimiento: 1. *Data Mining supervisado*, donde se sabe claramente lo que se busca, generalmente predecir unos ciertos datos o clases; y 2. *Data Mining no supervisado*, donde no se sabe lo que se busca, se trabaja con los datos (*¡hasta que confiesen!*).

Según los requerimientos del proyecto es necesario utilizar la minería de datos en modo supervisado, de tal forma que ayude a predecir el estado de un SLA según sus datos históricos. Con la utilización de la minería de datos se desea conocer el cumplimiento o no de unos acuerdos de nivel de servicio que han sido definidos para algún intervalo de tiempo. En el Data Warehouse se tiene como máximo 3 dimensiones, las cuales se utilizan para especificar un SLA según los parámetros definidos para él mismo, y una cuarta dimensión que la utilizamos como nuestra

dimensión de medida para verificar el estado del SLA. El atributo de la dimensión de medida es la variable que la minería de datos debe predecir.

#### **4.8.2 Tipo de conocimiento**

El tipo de conocimiento que se desea extraer va a marcar claramente la *técnica* de minería de datos a utilizar. Existen diferentes tipos de conocimiento: asociaciones, dependencias, clasificación, agrupamiento / segmentación, tendencias / regresión y unas reglas generales.

Para los diferentes tipos de conocimiento hay cuatro categorías principales de algoritmos de Minería de Datos, Estadística, Aprendizaje (Machine Learning), Redes Neuronales e Híbrido, cada una teniendo miembros que son particularmente útiles para funcionar con más exactitud en determinados y característicos conjuntos de datos numéricos o por categorías. Claro esta, que no todos los algoritmos sirven para un tipo de conocimiento, por ejemplo para *clasificación* el algoritmo utilizado es el Machine Learning que es uno de los algoritmos utilizados para Data Mining supervisado; el modo escogido para nuestro proceso de minería de datos.

##### **4.8.2.1 Clasificación**

El tipo de conocimiento a extraer queda dentro de la categoría *clasificación*. Una clasificación se puede ver como el esclarecimiento de una dependencia<sup>13</sup>, en la que el atributo dependiente puede tomar un valor entre varias clases, ya conocidas. Por ejemplo, para un SLA sobre el tráfico cursado por el servidor proxy en un día, nuestra variable dependiente (el atributo de la dimensión de medida) va a tomar un valor de “Yes” o “No” dependiendo si se cumple el SLA para el día escogido.

En la clasificación [26] o el aprendizaje supervisado como también se le conoce, los datos de entrada (denominados conjunto de entrenamiento) son instancias de las clases que se desean modelar e incluyen una serie de atributos o características. El objetivo de la clasificación es obtener una descripción precisa para cada clase utilizando los atributos de los datos de entrada. El modelo así obtenido puede servir para clasificar casos cuyas clases se desconozcan (el estado de los acuerdos de nivel de servicio) o, simplemente, para comprender mejor la información de la que disponemos.

---

<sup>13</sup> Una dependencia funcional (aproximada o absoluta) es un patrón en el que se establece que uno o más atributos determinan el valor de otro.

Las técnicas de clasificación se basan en el descubrimiento de patrones de los datos de entrada, por lo que se dispone de suficientes casos de entrenamiento (muchos casos de entrenamiento que clases diferentes) para obtener un modelo de clasificación fiable. Se necesitan bastantes datos para poder diferenciar patrones válidos de patrones debidos a irregularidades o errores.

#### **4.8.2.2 Modelo de clasificación – Árboles de decisión**

Los algoritmos de Aprendizaje (Machine Learning) usan procedimientos automáticos basados en operaciones lógicas o binarias con el fin de aprender una tarea de una serie de ejemplos. La mayoría de estos algoritmos emplean propuestas de árbol de decisión e inducción de reglas con el fin de generar la clasificación de expresiones lo suficientes simples para que se entiendan. Estos algoritmos se pueden implementar sin el uso de intervención experta, aunque se puede usar conocimiento experto.

Hasta el momento se tiene toda la información necesaria para construir un Data Mining de acuerdo a las necesidades, solamente falta por definir el algoritmo a utilizar. Para determinarlo hay que tener en cuenta las características de los datos, básicamente se debe considerar que serán grandes cantidades de datos recolectados en cortos intervalos de tiempo y que son pocos atributos por cada tabla de la base de datos. El algoritmo escogido es el C4.5 según la recomendación dada por K.E.Born-Thortonm en [27], cuyo trabajo fue determinar cual era la mejor clase de los algoritmos de Minería de Datos a usar con el fin de, rápidamente y con exactitud, estudiar las grandes cantidades de datos de rendimiento que se recogen rutinariamente de una red.

Las investigaciones realizadas por K.E.Born-Thortonm en su trabajo sugieren que el algoritmo C4.5 puede ser el mejor algoritmo para ser usado con el fin de estudiar con exactitud y rapidez los datos de rendimiento multiconjunto, multivariable y con pocas clases.

#### **4.8.3 Interpretación**

El despliegue del modelo a veces es trivial pero otras veces requiere un proceso de implementación o interpretación:

- El modelo puede requerir **implementación** (p.ej. tiempo real detección de tarjetas fraudulentas).

- El modelo es descriptivo y requiere **interpretación** (p.ej. una caracterización de los acuerdos de nivel de servicio según el comportamiento de la Intranet).
- El modelo puede tener muchos usuarios y necesita **difusión**: el modelo puede requerir ser expresado de una manera comprensible para ser distribuido en la organización (p.ej. las cervezas y los productos congelados se compran frecuentemente en conjunto, entonces ponerlos en estantes cercanos).

El modelo planteado cabe dentro de la clasificación de modelo descriptivo que requiere interpretación. Se utiliza tanto Data Warehouse como Data Mining como soporte para definir unos SLA's según las necesidades de la Red de Datos, pero después de aplicar el algoritmo se obtiene una respuesta que se debe interpretar y mostrar; de tal forma que los usuarios de la aplicación utilicen esta respuesta y puedan tomar unas decisiones más acertadas. La interpretación tiene que estar dirigida hacia las dimensiones involucradas y hacia el posible comportamiento (predicción) de la red según los datos históricos.

El modelo permite tener resultados puros, solamente entregados por el algoritmo, pero también se presta para construir un buen análisis de las variables involucradas, todo depende si el pre-procesamiento de los datos es bueno.

## Capítulo 5 CONCLUSIONES

- La plataforma WSMINING ofrece un soporte, basado en datos reales e históricos, a los administradores de la Red de Datos de la Universidad del Cauca para definir acuerdos de nivel de servicio.
- La aplicación desarrollada fue construida y probada en un entorno real; tomando datos de equipos importantes y altamente utilizados en la Red de la Universidad, como lo son: el servidor Proxy Hyperion y el Switch principal de la Intranet Accelar. Que manejan grandes cantidades de tráfico de datos a nivel interno (Switch) y hacia Internet (Proxy). Esto demuestra que la aplicación puede ser utilizada en ambientes reales, donde la estructura de la red de datos sea compleja y que tráfico sea elevado.
- Siguiendo un framework de gestión es posible construir una arquitectura bien estructurada que involucre diversidad de tecnologías y conceptos, pero a su vez permita ser una plataforma altamente inter operable con módulos bien definidos que faciliten la realización de cambios y/o posibles mejoras.
- Con la realización de la plataforma se concluyó que un punto vital para el desarrollo de cualquier proyecto es el conocimiento profundo de las tecnologías que puedan utilizarse y el análisis del entorno donde se desee implementar la aplicación, de tal manera que se optimice el tiempo de ejecución del mismo y se logre dar respuesta a los problemas de manera efectiva.
- La inclusión de nuevos conceptos como Data Warehouse y Data Mining en el trabajo de grado, permite que se realice una real y profunda investigación con el propósito de afianzar los conceptos y buscar interoperabilidad (necesidad altamente importante en el mundo de hoy).
- Las herramientas utilizadas en el desarrollo de la plataforma permite que sea escalable, es decir que el sistema tiene la capacidad de mantener y/o mejorar su

rendimiento conforme aumenta el número de equipos de red, para que de esta manera su crecimiento e integración con soluciones complementarias sea posible.

- La ventaja de usar servicios Web es que aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen. Además los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. Finalmente al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Después de una ardua búsqueda e investigación sobre los proyectos, trabajos y documentos realizados acerca de las temáticas que abarcaban el proyecto, se concluye que a nivel nacional es prácticamente nulo el desarrollo de proyectos que involucren Data Warehouse, Data Mining sobre datos extraídos de una red de datos. Por tal razón con este trabajo de grado se hace un aporte significativo.
- Nuestra plataforma permite concluir que un Data Warehouse puede ser construido según las necesidades específicas de una compañía o empresa. La tecnología no se limita únicamente a las empresas del sector comercial, donde a sido ampliamente difundida (manejo de perfiles de usuarios, comportamiento de algún producto, etc), sino que también puede ser utilizada en telecomunicaciones. WSMINING sirvió como prueba del comportamiento de la base de datos multidimensional en un entorno de mucho tráfico y con tiempos de respuesta rápidos.
- La construcción del Data Warehouse permitió optimizar el manejo de datos en bases de datos operacionales, hacer consultas más efectivas, manejar datos históricos, agregar conceptos multidimensionales. El Data Warehouse hizo que la aplicación tuviera unos datos estructurados y ordenados.
- Los datos históricos presentes en cualquier compañía, empresa o red de datos entregan valiosa información que deja al descubierto el comportamiento de los usuarios, los equipos y la red. Pero muchas veces son simplemente almacenados en bases de datos operacionales y utilizados para hacer consultas simples y/o puntuales. Con la plataforma se puede dar cuenta que la minería de datos, además de poder predecir el comportamiento de los usuarios y productos en

empresas o compañías relacionadas con el sector comercial, es posible predecir el comportamiento de unos elementos de red (Switch y Proxy) que refleja el uso de los usuarios en la red.

## Capítulo 6 GLOSARIO

<b>CIM</b>	El modelo de información común, es un extensible modelo orientado a objetos que contiene información sobre diferentes partes de la empresa.
<b>CORBA</b>	Es la arquitectura común de intermediarios de peticiones de objetos.
<b>DCOM</b>	El Modelo de Objetos de Componentes Distribuidos es uno de los tres más populares paradigmas de objetos distribuidos.
<b>IP</b>	El protocolo IP es el software que implementa el mecanismo de entrega de paquetes sin conexión y no confiable (técnica del mejor esfuerzo).
<b>JMX</b>	La tecnología extensiones de gestión Java provee las herramientas para construir soluciones distribuidas, basadas en Web, modulares y dinámicas para manejar y monitorear dispositivos, aplicaciones y servicios en red.
<b>MIB</b>	La base de la información de gestión contiene un conjunto de todos los objetos (OID) posibles de una red
<b>MRTG</b>	Es un programa de libre distribución que permite, cubrir las principales áreas funcionales en que se divide la gestión de red: Gestión de fallos, Gestión de configuración, Gestión de incidencias, Gestión de rendimiento y Gestión de seguridad
<b>OID</b>	Son los objetos que cada dispositivo mantiene en una o más variables que describen su estado.
<b>SNMP</b>	Protocolo Simple de Gestión de Red, es un protocolo de nivel de aplicación para consulta a los diferentes elementos que forma una red, ( <i>routers, switches, hubs, hosts, modems, impresoras, etc</i> ).

<b>SOAP</b>	Es un protocolo simple basado en XML y que permite a las aplicaciones intercambiar información sobre HTTP.
<b>RMI</b>	La invocación de métodos remotos utilizada por Java se soporta fuertemente en la serialización de objetos java, que permite a los objetos ser transmitidos como una cadena.
<b>TCP/IP</b>	Es un protocolo de interconexión general válido para cualquier plataforma, sistema operativo y tipo de red.
<b>UDDI</b>	Es una norma que trata sobre la descripción, la publicación y la localización de servicios Web.
<b>WMI</b>	La instrumentación de gestión de Windows es un componente del sistema operativo Microsoft Windows y es la implementación Microsoft de WBEM.
<b>WSMF</b>	Esta basada en la noción de objetos gestionados y sus relaciones. Un objeto gestionado representa un recurso y expone un conjunto de interfaces de gestión a través de las cuales el recurso subyacente se puede gestionar
<b>WSMN</b>	Es una red de cubierta lógica para la gestión de SLA's entre servicios, constituidas de comunicaciones intermedias.
<b>WSDM</b>	Define un protocolo para la interoperabilidad de información de gestión y capacidades utilizando Servicios Web
<b>WSDL</b>	Es un dialecto de XML que contiene información acerca de la interfaz, semántica y administración de una llamada a un servicio Web.

## Capítulo 7 REFERENCIAS Y BIBLIOGRAFÍA

- 
- [1] Para mayor información sobre SNMP ver <http://lavisit.upf.edu/tutorial.jsp?content=/tutorial/snmp/index.htm>
- [2] Para mayor información sobre CIM ver [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/common\\_information\\_model.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/common_information_model.asp)
- [3] Para mayor información sobre JMX ver <http://java.sun.com/products/JavaManagement/>
- [4] Para mayor información sobre WMI ver [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi\\_start\\_page.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp)
- [5] Para mayor información sobre DCOM, CORBA y RMI ver <http://my.execpc.com/~gopalan/misc/compare.html>
- [6] Para mayor información sobre el protocolo SOAP ver <http://www.w3schools.com/soap/default.asp>
- [7] Para mayor información sobre WBM y MBEM ver <http://www.dmtf.org/standards/wbem/>
- [8] Para mayor información sobre DTMF ver <http://www.dtmf.org>
- [9] Oscar Mauricio Caicedo, Natalia Carolina Maya. Gestión Basada en Servicios Web, 2004
- [10] Hewlett-Packard Development Company  
<http://devresource.hp.com/drc/specifications/wsmf/index.jsp>
- [11] White Paper Compuware Corporation. Implementing SLA: Tools for Success, 2002
- [12] Claudio Casares, "Tutorial sobre Data Warehouse".  
<http://personal.lobocom.es/claudio/>
- [13] Fayyad, Piatesky-Shapiro & Smyth: "From data mining to knowledge discovery: An overview" Advances in Knowledge Discovery and Data Mining (AAAI / MIT Press, 1996).
- [14] Weiss, S.M. y Indurkha, N. "Predictive Data Mining. A Practical Guide", Morgan Kaufmann Publishers, San Francisco, 1998.
- [15] Berzal Galiano Fernando, "Machine Learning", Capítulo 6 Árboles de decisión.

- [16] Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. Y Zanasi, A. "Discovering Data Mining. From Concept to Implementation", Prentice Hall, 1998.
- [17] The CRISP-DM consortium. "CRISP-DM Step by step data mining guide", August 2000.
- [18] Para mayor información ver <http://www.javahispano.org/articles.article.action?id=49>
- [19] Kimball R. The Data Warehouse Toolkit. John Wiley & Sons, 1996.
- [20] Inmon W.H. Building The Data Warehouse. QED Press / John Wiley & Sons, 1992.
- [21] Fernando Carpani. CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales, 2000.
- [22] Golfarelli M., Maio D., Rizzi S. Conceptual Design of Data Warehouse From E/R Schemes.
- [23] Verónica Peralta. Diseño Lógico de Data Warehouses a partir de Esquemas Conceptuales Multidimensionales, 2001.
- [24] Marotta, A. Peralta, V. "Diseño de Data Warehouses: Un Enfoque Basado en Transformación de Esquemas". Uruguay, 2000.
- [25] José Hernando Bahamón L. Propuesta de un método para el diseño y modelado de una bodega de datos. Sistemas y Telemática de la Universidad del Icesi.
- [26] Fernando Berzal Galiano, Machine Learning, 1999.
- [27] K.E.Born-Thortonm. Gestión de red proactiva usando minería de datos, IEEE 1998.