METODOLOGÍA DE DISEÑO DE CUBOS OLAP PARA INTELIGENCIA DE NEGOCIOS USANDO MONDRIAN Y JPIVOT A PARTIR DE UNA BASE DE DATOS TRANSACCIONAL

JOHN HERMAN MANTILLA HERNANDEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO – MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2011

METODOLOGÍA DE DISEÑO DE CUBOS OLAP PARA INTELIGENCIA DE NEGOCIOS USANDO MONDRIAN Y JPIVOT A PARTIR DE UNA BASE DE DATOS TRANSACCIONAL

JOHN HERMAN MANTILLA HERNANDEZ

Trabajo de Investigación presentado como requisito parcial para optar al título de Ingeniero de Sistemas

Director
SERGIO HENRY RICO RANGEL
Ingeniero de Sistemas
Magister (c) Ingeniería de Sistemas e Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO – MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2011

Dedicatorias a Dios todo Poderoso por darme la salud e inteligencia necesarias.
A mis Padres, Fanny y Herman, por su amor y su apoyo, constante e incondicional.
A mi futuro hijo quién me impulsó a culminar con mi carrera, y en especial a la madre de ese hijo que estamos esperando, Andrea.

CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

RESUMEN

TITULO: METODOLOGÍA DE DISEÑO DE CUBOS OLAP PARA INTELIGENCIA DE NEGOCIOS USANDO MONDRIAN Y JPIVOT A PARTIR DE UNA BASE DE DATOS TRANSACCIONAL *

AUTOR: MANTILLA HERNANDEZ, John Herman**

PALABRAS CLAVE: OLAP, Reporte Multidimensional, ETL, Inteligencia de Negocios, MDX, Mondrian

DESCRIPCIÓN: El presente trabajo de investigación se enfoca en elaborar una metodología que apoye en diseño y visualización de cubos OLAP haciendo uso de herramientas de software libre, para ello se realiza una introducción de los diferentes conceptos que engloba la inteligencia de negocio en cuánto a cubos multidimensionales, bodegas de datos, técnicas de ETL (Extracción, transformación y carga de datos), cubos OLAP, reportes JPivot y se describe la metodología apoyándose de un caso práctico que se irá desarrollando a lo largo del documento para facilitar la comprensión de los conceptos presentados.

Para desarrollar el caso práctico principalmente se han utilizado las herramientas de software libre Mondrian y JPivot, junto a elementos de la suite Pentaho y herramientas complementarias tales como Apache Tomcat y la versión libre de Oracle Express Edition; conforme a los conceptos de la comunidad y el rendimiento frente a herramientas similares se considera la suite más completa y madura del mercado de software libre en el momento de la redacción del documento.

Como resultado del presente documento, pretende crear expectativas en el uso de tecnologías que sirvan de apoyo para la toma de decisiones para generar valor en las organizaciones; de lo que se espera la estructuración de proyectos futuros relacionados con la temática establecida y de esta manera se pueda aumentar la competitividad de las organizaciones mediante el análisis de su propia información y generación de conocimiento. Llevando a un nivel más alto en el uso de las tecnologías en la empresas de la región.

* Trabajo de Grado, Trabajo de Investigación

** Facultad de Ingeniería Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Ing. Sergio Henry Rico Rangel

ABSTRACT

TITLE: DESIGN METHODOLOGY OF OLAP CUBE FOR BUSINESS INTELLIGENCE USING MONDRIAN AND JPIVOT FROM ATRANSACTIONAL DATABASE *

AUTOR: MANTILLA HERNANDEZ, John Herman**

KEYWORDS: OLAP, Multidimensional Reporting, ETL, Business Intelligence, MDX, Mondrian

DESCRIPTION: This research focuses on developing a methodology to support design and display OLAP cubes using free software tools, for it is an introduction of different concepts encompassing business intelligence in how multidimensional cubes, datawarehousing, ETL (Extraction, transformation and Loading) OLAP cubes, JPivot reports and describes the methodology supports a case study that will be developed throughout the document to facilitate understanding of the concepts presented.

To develop the case study have been mainly used free software tool JPivot and Mondrian, with elements of the Pentaho suite and complementary tools such as Apache Tomcat and the free version of Oracle Express Edition; according to the concepts of community and performance against similar tools suite is considered the most complete and mature free software market at the time of writing the document.

As a result of this document is intended to create expectations in the use of technology to support decision-making to create value in organizations; of what is expected structuring future projects related to the topic established and in this way can increase the competitiveness of organizations through the analysis of their own information and knowledge generation. Leading to a higher level in the use of technologies in enterprises in the region.

* Project Grade, Research Project

** Faculty of Engineering Physics and Mechanics. College of Engineering System and Informatics. Director Eng. Sergio Henry Rico Rangel

INTRODUCCION

El presente trabajo de investigación se enfoca en elaborar una metodología que apoye en el diseño y visualización de cubos OLAP haciendo uso de herramientas de software libre para medianas y grandes empresas, para ello se realiza una introducción de los diferentes conceptos que engloba la inteligencia de negocio y se describe la metodología apoyándose de un caso práctico que se irá desarrollando a lo largo del documento para facilitar la comprensión de los conceptos presentados.

Para desarrollar el caso práctico principalmente se han utilizado las herramientas de software libre Mondrian y JPivot, de la suite Pentaho; conforme a los conceptos de la comunidad y el rendimiento frente a herramientas similares se considera, ésta es la suite más completa y madura del mercado en el momento de la redacción del documento.

En el primer capítulo del documento, se establecen los aspectos que motivaron la realización del proyecto, describiendo el problema que se pretende abarcar y analizando la viabilidad del desarrollo del proyecto, teniendo en cuenta el estado actual de la implementación de este tipo de tecnologías en las organizaciones colombianas.

El capítulo 2, presenta a manera de guía un flujo tabular de los pasos de la metodología propuesta en el libro. En esta tabla se detalla la descripción, el rol responsable, los recursos de entrada - salida y la página correspondiente de cada uno de los pasos de la metodología. En éste mismo capítulo se encuentran los iconos diseñados para facilitar la identificación de la metodología y el caso práctico.

En el capítulo 3, se hace una introducción de los conceptos básicos de inteligencia de negocios, identificando las ventajas de la implementación de este tipo de tecnologías para la toma de decisiones en una organización, se presenta el modelo de madurez del estado de la inteligencia de negocio en las organizaciones.

En los capítulos 4 y 5, se definen los conceptos y tecnologías necesarias para la implementación de una bodega de datos a partir de sistemas de información o bases de datos transaccionales ya implementadas en la organización. En estos capítulos se explican los diseños de estas bodegas de datos para su implementación en cubos OLAP.

Posteriormente en los capítulos 6 y 7, se define la estructuración de los cubos OLAP y la visualización de sus tablas dinámicas mediante reportes basados en JPivot. En el capítulo sexto se describen los tipos y reglas principales para el diseño de un cubo OLAP; y en el último capítulo se identifica los distintos tipos de reportes que ayudan a presentar la información para una mayor comprensión en su lectura y análisis.

1. PRESENTACION

1.1 Formulación del problema

Las tecnologías de motores de bases de datos permiten almacenar gran cantidad de datos obtenidos de transacciones de los procesos de las organizaciones, pero pocas empresas usan las tecnologías y metodologías disponibles para realizar extracción de información de dichos datos para ayudar a proveer conocimientos y toma de decisiones.

Existe la opción de analizar la información a partir de reportes multidimensionales, el carácter multidimensional hace referencia a la posibilidad de visualizar los datos desde distintas perspectivas o dimensiones establecidas por quién visualiza el reporte. La presentación de reportes multidimensionales más usada son las tablas dinámicas de las hojas de cálculo como Microsoft Excel, herramienta que obliga a obtener la información directamente de tablas del mismo programa; en la actualidad las empresas tienden a almacenar la información en repositorios más robustos y confiables como lo son las bases de datos transaccionales, pero no se realiza análisis de la información generada en el almacenamiento de estos datos ni basándose en este tipo de reportes multidimensionales.

Surge entonces la siguiente pregunta de estudio:

¿Cómo favorecer el análisis en las organizaciones con reportes multidimensionales tomados a partir de bases de datos transaccionales para proveer conocimientos de la información que ya poseen?

La propuesta planteada en este trabajo es mediante la construcción de cubos y reportes OLAP, y la contribución que se presenta es la documentación de una metodología que guie dicha construcción.

1.2 Objetivos

1.2.1 Objetivo General

Proponer una metodología para el diseño de Cubos OLAP a partir de bases de datos Transaccionales usando las tecnologías Mondriam y JPivot, que sirva de guía para la implementación de Cubos OLAP como opción de inteligencia de negocios en el sector empresarial

1.2.2 Objetivos Específicos

Hacer una revisión del estado del arte referente a las tecnologías de Inteligencia de Negocios con el propósito de identificar las tecnologías líderes y las aplicaciones empresariales.

Analizar las herramientas de software libre de mayor relevancia para la construcción de cubos OLAP.

Documentar la metodología para el diseño de cubos OLAP que contenga:

- El procedimiento para el diseño de DataMarts
- La estructura de Dimensiones y Jerarquias
- La estructura de tablas de Hecho
- La estructura del esquema Mondrian
- Estructura de las Consultas MDX[1] para la visualización de los Reportes OLAP
- Implementación de JPivot[2] para la visualización de los Reportes de Cubos OLAP

Aplicar la metodología propuesta mediante la implementación de un demo de Cubo OLAP que sea alimentado por una base de datos transaccional.

1.3 Impacto y Viabilidad

1.3.1 Impacto

- La comunidad empresarial podrá contar con una herramienta guía y de ejemplo para diseñar sus propias bases de datos OLAP usando tecnologías de Software libre. Con el fin de apoyar a la toma de decisiones y análisis de información.
- La comunidad estudiantil tendrá una guía para el uso de una metodología de gran impacto para el sector empresarial del país, con base a esta investigación, despertaría el interés en el manejo y administración de Datos.
- La investigación y documentación de este tema está acorde a las necesidades del sector empresarial, estando a fin con la actualidad tecnológica.
- Al no existir un manual de guía en español completo desde el diseño de una base de datos OLAP hasta la construcción de los Reportes Multidimensionales, será una herramienta de innovación para la comunidad estudiantil, universitaria y empresarial.

1.3.2 Viabilidad

- La Universidad Industrial de Santander y en especial la Escuela de ingeniería de Sistemas apoyan la investigación de las tecnologías actuales y que son de gran demanda en el sector empresarial.
- La investigación se centra en el uso de herramientas de código abierto, por lo tanto no es necesario la adquisición de licencias y puede ser adoptado por empresas de distintos niveles económicos.
- El recurso humano para la investigación tiene experiencias y conocimientos necesarios para inicializar el proyecto propuesto.
- El tiempo requerido para este proyecto no abarca grandes espacios, puesto que la dedicación y puesta en práctica de las metodologías/ejemplos que se realizarían en el proyecto son de tiempo completo.

[1] **MDX** es el acrónimo en inglés de Expresiones Multidimensionales (*Multidimensional Expressions*)

[2] Librería de Componentes JSP para construir Tablas OLAP

1.4 Justificación

El uso de información consolidada en las organizaciones es de vital importancia, ya que es la principal fuente para la toma de decisiones, planeación del futuro y la asignación de recursos de manera eficiente. Normalmente, dicha información ha sido consultada mediante reportes en línea o impresos, presentación de diapositivas de datos obtenidos en consultas determinadas, las cuáles eran apoyadas por el departamento de tecnología o el departamento financiero. Aún cuando estos tipos de reportes no pierden vigencia, cada vez es más necesario la necesidad de reportes en otros tipos de formatos, más fáciles de usar, que tengan una consulta centralizada, puedan ofrecer un mayor nivel de detalle y flexibilidad en su consulta. Los analistas de información (knowledge workers) están buscando hacer un análisis más sofisticado y necesitan tener a su alcance la información de manera más rápida. Así mismo dicha información debe estar respaldada por los resultados obtenidos en los distintos departamentos de la organización. La tecnología OLAP, suple la necesidad de tener este tipo de reportes dinámicos ad-hoc[3] para facilitar la toma de decisiones de manera más rápida, inteligente, a nivel horizontal y vertical de la organización.

Los reportes OLAP constituyen entonces una buena alternativa para favorecer la creación de nuevos reportes y recolectar información de valor para las organizaciones, no obstante la creación de dichos reportes a partir de bases de datos transaccionales podría parecer confuso y complicado tanto para académicos como para las organizaciones. La intención con el presente trabajo es elaborar una guía metodológica que ayude en el diseño y elaboración de cubos y reportes OLAP utilizando herramientas de software libre.

[3] Personalizable en línea

2. FLUJO TABULAR DE LA METODOLOGÍA

Iter	DESCRIPCIÓN	RESPONSABLE (ROL)	RECURSOS		Pág.
			ENTRADAS	SALIDAS	
1.	Inteligencia de Negocios (Análisis y Requerimientos)				21
1.1	**Describir la información que se requiere ver en reportes OLAP** En este paso se debe identificar cuál es la información que se necesita para la toma de decisiones en la organización. Teniendo en cuenta que dicha información debe encontrarse en algún medio de almacenamiento digital (Bases de datos, hojas de cálculo ó textos planos)	Analistas de Información	Guía de los sistemas de Información de la organización	Documento de especificación de la información que se desea consultar para la toma de decisiones	26
1.2	**Estrategia de Inteligencia de Negocios** En este paso se debe identificar los puntos clave para los analistas, usuarios finales, profesionales de TI, Tomadores de decisiones y su participación en el proyecto de Inteligencia de Negocios. Se debe tener claro el estado actual de la información de donde proviene los datos y su disponibilidad para el desarrollo del proyecto.	Administrador TI, Analistas de Información	Documento de especificación de la información que se desea consultar para la toma de decisiones	Especificación de Sistemas de Información y Participación de los implicados	27
2.	Data Warehouse (Modelización)				30
2.1	**Modelo Conceptual de Datos** Basado con el Documento del paso 1.1, se debe identificar cuáles son los procesos y vistas de negocios que responden a las preguntas de los Usuarios Finales y Analistas de Información. Se deben identificar tablas de Hecho y Dimensiones	Analista OLAP	Documento de especificación de la información que se desea consultar para la toma de decisiones Especificación de Sistemas de Información y Participación de los implicados	Diagrama de especificación de Tablas de Hecho y Dimensiones (Modelo Conceptual de Datos)	38
2.2	**Modelo Lógico de Datos** Basado con el Documento del paso 2.1, se debe identificar cuáles son las métricas de las tablas de hecho y los atributos de las dimensiones.	Analista OLAP	Diagrama de especificación de Tablas de Hecho y Dimensiones	Diagrama de especificación de Métricas y Atributos (Modelo Lógico	40

	Se deben identificar las llaves principales de las dimensiones y su parametrización como llave foránea en las tablas de hecho.		(Modelo Conceptual de Datos)	de Datos)	
2.3	**Modelo Físico de Datos** Basado en los Documentos de los pasos 2.1 y 2.2, se debe diseñar el modelo Físico de los datos dependiendo del motor de base de datos elegido. Éste modelo es basado en los diagramas entidad relación, respetando los esquemas para el diseño de un Data Mart o Data Warehouse (Diseño en estrella ó Diseño en copo de nieve).	Administrador TI	Diagrama de especificación de Tablas de Hecho y Dimensiones (Modelo Conceptual de Datos) Diagrama de especificación de Métricas y Atributos (Modelo Lógico de Datos)	Documento de especificación del Módelo Físico de datos (Tablas y atributos) y Diagrama E/R del Modelo Físico de datos.	41
3.	Extracción, Transformación y Carga de Datos (Integración)				43
3.1	**Identificación y construcción de las consultas** En este paso se debe crear y/o diseñar las estrategias de obtención de datos desde las distintas fuentes de información (Probablemente proviene de diferentes sistemas de información). Este paso es conocido como la Extracción de Información. En la metodología de este proyecto se hace uso de la herramienta Kettle de Pentaho Data Integration.	Administrador TI	Documento de especificación del Módelo Físico de datos (Tablas y atributos) y Diagrama E/R del Modelo Físico de datos.	Documento de especificación de Consultas SQL y métodos de extracción de datos	52
3.2	**Diseño de Transformaciones** En este paso se deben crear las distintas transformaciones necesarias para la alimentación de las tablas del diagrama E/R del Data Mart o Data Warehouse diseñado. Éstas transformaciones serán diseñadas con base a la información del paso anterior. En estas transformaciones se debe hacer el cargue a las tablas del esquema diseñado. Para este paso, en la metodología se usan las transformaciones del menú de Kettle.	Administrador TI	Documento de especificación de Consultas SQL y métodos de extracción de datos. Documento de especificación del Módelo Físico de datos (Tablas y atributos) y Diagrama E/R	Archivos de Transformaciones	57

			del Modelo Físico de datos.		
4.	Esquematización (OLAP en el contexto Mondrian)				61
4.1	**Definición del Esquema Mondrian** En este paso se debe realizar el respectivo esquema mondrian en el archivo .xml, en el cual se debe especificar las Dimensiones, jerarquías, medidas y propiedades de las mismas (Basado en los documentos del paso 2.) El esquema debe a su vez contener las respectivas conexiones al DW o DataMart creado en el paso 3. Para este paso se puede usar la herramienta Schema-Workbench para construir el archivo XML	Analista OLAP	Documento de especificación del Módelo Físico de datos (Tablas y atributos) y Diagrama E/R del Modelo Físico de datos. Documento de especificación de la información que se desea consultar para toma de decisiones	Especificación de Esquema mondrian	69
4.2	**Definición de Consulta MDX Base** En este paso se debe definir una Consulta MDX que enlace dimensiones del esquema definido en el paso 4.1	Analista OLAP	Especificación de Esquema mondrian	Especificación de Consulta MDX	70
5.	Presentación de la Información en Reportes (OLAP en el contexto Jpivot)				78
5.1	**Generación de Reporte OLAP** En este paso, con base al esquema creado en el paso 4.1 y la consulta del paso 4.2, se debe configurar para la generación del Reporte web usando las clases de Jpivot. En este paso se debe validar la funcionalidad de las distintas dimensiones y medidas configuradas. Para este paso se debe usar una	Analista OLAP	Especificación de Esquema mondrian Especificación de Consulta MDX	Reporte Olap generado en la web	82

	aplicación web con la clase de Java JPivot, por ejemplo la Suite Visión Empresarial				

Tabla 1. Flujo tabular de la Metodología [Fuente: Diseño propio de la Metodología]

2.1 Iconos de Guía

En cada uno de los siguientes capítulos encontrará estos dos iconos que le permite identificar sí se está refiriendo a la metodología o al caso práctico que ilustra el uso de la metodología:

 Referencia a la metodología propuesta en el libro.

La metodología es una guía propuesta para estudiantes y profesionales de inteligencia de negocio sobre como diseñar y elaborar reportes y cubos OLAP a partir de bases de datos transaccionales.

 Referencia al caso práctico que se desarrolla en el libro.

El caso práctico basado en una organización comercial es un ejemplo de la utilización de la metodología descrita que busca facilitar la comprensión de los conceptos presentados.

3. INTELIGENCIA DE NEGOCIOS

El desarrollo de las empresas en el contexto del análisis de la información ha propiciado la necesidad de tener mejores, más rápidos y más eficientes métodos para extraer y transformar los datos de una organización en información y distribuirla a lo largo de la cadena de valor.

La inteligencia de negocio[4] responde a esta necesidad, y se puede entender, en una primera aproximación, que es una evolución de los sistemas de soporte a las decisiones (DSS, Decissions Suport Systems). El concepto de Inteligencia de negocios, a pesar de ser un tema crítico e importante en las empresas, no es un tema nuevo. En octubre de 1958 Hans Peter Luhn, investigador de IBM, acuñó el término en el artículo "A Business Intelligence System":

La habilidad de aprender las relaciones de hechos presentados de la forma que guíen las acciones hacia una meta deseada.

Se puede precisar hoy como:

La adquisición y utilización de conocimiento basado en hechos para mejorar la estrategia del negocio y las ventajas tácticas del mercado. [Fuente: **(Oramas, 2009, pág. 21)**]

Desde entonces, el concepto ha evolucionado apilando diferentes tecnologías, metodologías y términos bajo éste concepto. Es necesario, por lo tanto, establecer una definición formal:

Se entiende por **Business Intelligence** al conjunto de metodologías, prácticas, aplicaciones y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización. [Fuente: **(Díaz, Introducción al Bussines Intelligence, 2010)**]

Entre las tecnologías que forman parte de la Inteligencia de Negocios se encuentran:

- OLAP
- Data Warehouse
- Cuadro de mando
- Cuadro de mando integral
- Minería de datos
- Dashboards
- Integración de datos
- Previsiones

[4] O Bussiness Intelligence (BI)

- Reporteadores

3.1 Ventajas de un sistema de Inteligencia de Negocios

La implantación de estos sistemas de información proporciona diversas ventajas entre las que se puede destacar:

- Crear un círculo virtuoso de la información (los datos se transforman en información que genera un conocimiento que permite tomar mejores decisiones que se traducen en mejores resultados y que generan nuevos datos). Como se ilustra en la siguiente figura:

Figura 1. ***Círculo virtuoso de la información*** *[Fuente:* **(Urquizu, 2010)***]*

- Permitir una visión única, conformada, histórica, persistente y de calidad de toda la información.
- Crear, manejar y mantener métricas, indicadores claves de rendimiento (KPI, Key Performance Indicator) e indicadores claves de metas fundamentales para la empresa.
- Aportar información actualizada tanto a nivel agregado como en detalle.
- Reducir el diferencial de orientación de negocio entre el departamento de Tecnologías de la Información y la organización.

- Mejorar la comprensión y documentación de los sistemas de información en el contexto de una organización.

3.2 Modelo de Madurez para Inteligencia de Negocios

El modelo de madurez permite clasificar la organización desde el punto de vista del grado de madurez de implantación de sistemas de Inteligencia de Negocios en la misma.

Las características que presentaría cada etapa del modelo de madurez, serían las siguientes: [Fuente: **(Ordoñez, 2011)**]

- **Fase 1: No existe BI.** Los datos se hallan en los sistemas de procesamiento de transacciones en línea (OLTP, On-Line Transaction Processing), dispersos en otros soportes o incluso sólo contenidos en el know-how de la organización. Las decisiones se basan en la intuición, en la experiencia, pero no en datos consistentes. El uso de datos corporativos en la toma de decisiones no ha sido detectado y tampoco el uso de una herramienta adecuada al hecho.
  En general, el valor de la información en la toma de decisiones no es suficientemente apreciado y promovido en la organización; aparecen varias versiones de "la verdad", dependiendo de cómo cada ejecutivo define conceptos como "Utilidad", "Ingresos", "Facturación" o cualquier otro que requiere para su análisis y por supuesto, de qué fuente los toma; cada funcionario maneja un argot propio y no existe un acuerdo corporativo sobre los términos del negocio.

- **Fase 2: No existe BI, pero los datos son accesibles.** No existe un procesado normal de los datos para la toma de decisiones, aunque algunos usuarios tienen acceso a información de calidad y son capaces de justificar decisiones con dicha información. Frecuentemente este proceso se realiza mediante Excel o algún tipo de reporting. Se intuyen que deben existir soluciones para mejorar este proceso pero se desconoce la existencia de la Inteligencia de Negocios.

- **Fase 3: Aparición de procesos formales de toma de decisiones basada en datos.** Se establece un equipo que controla los datos y que permite hacer informes contra los mismos que permiten tomar decisiones fundamentales. Los datos son extraídos directamente de los sistemas transaccionales sin data cleansing[5] ni modelización, ni existe un data warehouse.
  En un escenario menos desalentador (aunque todavía lejano de la solución óptima), o bien, una vez capitalizada la mala experiencia anterior, se decide disponer de una infraestructura tecnológica separada de los sistemas OLTP; pero como no hay recursos para una adecuada planeación y la solución se necesita "ya", se implementa un primer datamart para resolver el problema crítico: típicamente la información de ventas. Si bien se lleva a cabo un proceso deETLC en alguna extensión y eventualmente se combinan las fuentes necesarias, normalmente no se realiza una identificación adecuada de requerimientos, y la información incorporada

[5] Data Cleasing consiste en el proceso de detectar y mejorar (o borrar) registros incorrectos e incompletos con el objetivo de conseguir datos coherentes y consistentes.

en eldatamart no es completa, no obedece a criterios ni necesidades corporativas sino departamentales, y típicamente se agrega, limitando así su potencial para el análisis.

- **Fase 4: Data Warehouse.** El impacto negativo contra los sistemas OLTP lleva a la conclusión de que un repositorio de datos es necesario para la organización. Se percibe la bodega de datos como una solución deseada. El reporting sigue siendo personal.
  En vista de la experiencia adquirida, es claro que la unión de los datamarts en una bodega de datos corporativa, no es posible a menos que se lleve a cabo un proceso de planeación y análisis de requerimientos con ese alcance, es decir, con alcance corporativo. Un proceso que lleve a la identificación juiciosa de dimensiones y medidas fundamentales del negocio, identificando y seleccionando sus fuentes potenciales e incluyendo toda la información inherente a cada componente. Un proceso que logre capturar en toda su extensión a la esencia de la información del negocio, y que vaya de la mano con la creación de un glosario de términos y conceptos que idealmente guíe la consulta y explotación de la información del negocio.

- **Fase 5: Data warehouse crece y el reporting se formaliza.** La bodega de datos funciona y se desea que todos se beneficien del mismo, de forma que el reporting corporativo se formaliza. Se habla de OLAP, pero sólo algunos identifican realmente sus beneficios.
  Es hacer las cosas bien: Planear global y construir local, ahora sí, como debió hacerse desde el principio; el concepto de bodega de datos corporativa toma fuerza, se entiende la utilidad y complejidad del proceso de ETL[6]. Los problemas de calidad de los sistemas OLTP empiezan a solucionarse de raíz. La planeación estratégica de la organización se mide con base en indicadores y éstos ya pueden calcularse de una manera más automática y segura con base en la bodega de datos. Además las herramientas para construcción de tableros de control demuestran su efectividad, ahora que además no debe llevarse a cabo un proceso manual tortuoso para alimentarlas.

- **Fase 6: Despliegue de OLAP.** Después de cierto tiempo, ni el reporting ni la forma de acceso al data warehouse es satisfactoria para responder a preguntas sofisticadas. OLAP se despliega para dichos perfiles. Las decisiones empiezan a impactar de forma significativa en los procesos de negocios de toda la organización.
  Se tiene una verdadera integración entre la gestión estratégica y la gestión operacional. La información es el activo principal de la organización y se cumplen las metas: La información es “accesible” (entendible, navegable y con alto desempeño), la información es consistente, adaptable y soporta los cambios en el negocio, hay control de acceso y visibilidad sobre el uso de la información, y UNA y solo una verdad y fuente única de información.

[6] ETL (Extract, Transform and Load) Extraer, transformar y cargar a base de datos, data marts o data warehouse.

- **Fase 7: Business Intelligence se formaliza.** Aparece la necesidad de implantar otros procesos de inteligencia de Negocio como Data Mining, Balanced Scorecard, entre otros; y procesos de calidad de datos impactan en procesos como Customer Relationship Management (CRM), Supply Chan Management (SCM)… Se ha establecido una cultura corporativa que entiende claramente la diferencia entre sistemas OLTP y DSS[7].
  El sistema de Inteligencia de Negocio desborda los límites corporativos y se extiende a clientes, proveedores, socios de negocio y en general los "stakeholders", término en inglés que agrupa a todos los terceros de interés relacionados con la empresa. BI es un recurso corporativo estratégico que definitivamente orienta el negocio.

3.3 ¿Cuándo es necesario la Inteligencia de Negocios?

Existen situaciones en las que la implantación de un sistema de Inteligencia de Negocios resulta adecuada. Se destaca, entre todas las que contienen:

- La toma de decisiones se realiza de forma intuitiva en la organización
- Identificación de problemas de calidad de información
- Uso de Excel como repositorios de información corporativos o de usuario. Lo que se conoce como Excel caos[8].
- Necesidad de cruzar información de forma ágil entre departamentos.
- Las campañas de marketing no son efectivas por la información usada.
- Existe demasiada información en la organización para ser analizada de la forma habitual. Se ha alcanzado la masa crítica de datos.
- Es necesario automatizar los procesos de extracción y distribución de información.

3.4 Metodología para aplicar Inteligencia de Negocios en una organización: ANÁLISIS Y REQUERIMIENTOS

"No hay ningún viento favorable para el que no sabe a qué puerto se dirige"
Arthur Schopenhauer (1788-1860) Filósofo alemán.

En primer lugar se debe identificar qué se quiere obtener a partir de la Inteligencia de Negocios. Ya conocidas las ventajas y los casos en los que se recomienda implementar un proyecto de inteligencia de negocios (el cuál no es un proyecto a corto plazo), se debe prestar gran atención al objetivo que se quiere lograr con esta implementación.
La frase que inicia este apartado hace alusión a lo que se desea obtener en este primer paso de la metodología: *Saber a dónde se debe dirigir*.

[7] DSS (Decision Support System) se conoce como un Sistema de Apoyo a las Decisiones.

[8] Se entiende como Excel caos el problema resultante del uso intensivo de Excel como herramienta de análisis. Cada usuario trabaja con un archivo personalizado. Como resultado, la información no cuadra entre departamentos y el coste de sincronización es sumamente elevado.

En definitiva, los sistemas de Inteligencia de Negocios buscan responder a las siguientes preguntas:

- ¿Qué sucedió?
- ¿Qué sucede ahora?
- ¿Por qué sucedió?
- ¿Qué sucederá?

3.4.1 Estrategia de Inteligencia de Negocios

Implementar un proyecto de inteligencia de negocios en una organización es una tarea de gran dedicación. Las buenas prácticas enseñan que, para llegar a un buen fin, es importante tener una estrategia de inteligencia de negocio que coordine de forma eficaz las tecnologías, el uso, y los niveles en el proceso de madurez.

Pero, ***¿cómo se puede detectar que no existe una estrategia?***
Es posible detectar que no existe una estrategia definida a través de los siguientes ítems y percepciones en el interior de la organización:

- Los usuarios de la organización indican que el área de informática o tecnologías de información, son el origen de sus problemas de inteligencia de negocio.
- Las directivas consideran que implementar un proyecto de inteligencia de negocio es otro costo más.
- El area de Informática continúa preguntando a los usuarios finales sobre las necesidades de los informes
- El sistema de Inteligencia de Negocio está soportado por el soporte del área de tecnología.
- No hay diferencia entre BI y gestión del rendimiento
- No es posible medir el uso del sistema de inteligencia de negocio
- No es posible medir el retorno de inversión (ROI, Return On Invest) del proyecto de Inteligencia de Negocios
- Se considera que la estrategia para la bodega de datos es la misma que para el sistema de inteligencia de negocio
- No hay un plan para desarrollar, contratar, retener y aumentar el equipo de Inteligencia de Negocio
- No se ha socializado que la empresa tiene una estrategia para la Inteligencia de Negocio
- No existe un responsable funcional (o bien, el asignado no es el adecuado).
- No existe un centro integrado de gestión de competitividad
- Existen múltiples soluciones en la organización distribuidas en diferentes departamentos que repiten funcionalidad
- No hay un plan de formación real y consistente de uso de las herramientas

- Alguien cree que es un éxito que la información consolidada esté a disposición de los usuarios finales al cabo de dos semanas
- Los usuarios creen que la información de la bodega de datos no es correcta.

El desarrollo de una estrategia de negocio es un proceso a largo plazo que incluye múltiples actividades, entre las que es conveniente destacar:

- Crear un equipo de gestión de Inteligencia de Negocio. Tiene el objetivo de generar conocimiento en tecnologías, metodologías, estrategia, con la presencia de un sponsor a nivel ejecutivo y con analistas de negocio implicados y que tengan responsabilidad compartida en éxitos y fracasos.
- Establecer los estándares de Inteligencia de Negocio en la organización para planear tanto las tecnologías existentes como las futuras adquisiciones.
- Identificar qué procesos de negocio necesitan diferentes aplicaciones analíticas que trabajen de forma continúa para asegurar que no existen silos de funcionalidad.
- Desarrollar un framework de métricas a nivel empresarial como el pilar de una gestión del rendimiento a nivel corporativo.
- Incluir los resultados de metodologías analíticas (minería de datos u otras) en los procesos de negocio con el objetivo de añadir valor a todo tipo de decisiones.
- Revisar y evaluar el portafolio actual de soluciones en un contexto de riesgo/recompensas.
- Considerar inversiones inteligentes cuyo retorno de inversión estén dentro de un periodo de tiempo de un año. Además, tener en cuenta los diferentes análisis de mercado, de soluciones e incluso el *hype cycle*[9] de Gartner para conocer el estado del arte.
- Aprender de los éxitos y fracasos de otras empresas revisando casos de estudio y consultando a las empresas del sector para determinas qué ha funcionado y qué no.
- Culturizar la organización con la orientación de inteligencia de negocio.
- Alinear los departamentos, en especial de Tecnología y la estrategia de la organización en caso de no poder organizar un Centro de Competencia de Inteligencia de Negocio, fundamental para trabajar como equipo integrado. El departamento de tecnología debe entender las necesidades y entregar la mejor solución ajustada a la necesidad particular y escalable a otras futuras.
- Poner atención en las necesidades que requieren inteligencia de negocio en la organización porque se acostumbra a satisfacer a los usuarios o departamentos que gritan más fuerte, y esto no significa que den mayor valor a la organización. Por ejemplo, los departamentos de finanzas son un caso típico de baja atención en este tipo de soluciones.

[9] El *hype cycle* de Gartner es una representación gráfica de la madurez, adopción y aplicación de negocio de una o varias tecnologías específicas. Es decir, muestra el ciclo de vida de dichas tecnologías.

3.4.2 **Caso Práctico**

Para aplicar la metodología en el desarrollo de este libro se aplicará inteligencia de negocios y todo el desarrollo de la metodología posterior a un caso de ejemplo:

TEXTICOL

Una organización comercial que se denominará **Texticol**, el negocio de esta organización es la fabricación, distribución y venta de productos de vestir. Tiene puntos de ventas ubicados en distintas ciudades de Colombia y usa el sistema de distribuidores para llegar a otros mercados tantos nacionales como internacionales (Venta de ropa en almacenes de grandes superficies).

Los productos que comercializa esta organización son productos para niños, hombres y mujeres. Abarcando líneas de ropa sport, casual y formal.

Los analistas de mercadeo de la organización buscan obtener la mayor información proveniente de las ventas que se realizan, teniendo en cuenta que actualmente la organización usa un software para las ventas que se realizan desde los puntos de venta. Un software que administra las ventas y transacciones de los distribuidores. Además otras aplicaciones software que gestionan otra información que complementa la empresa (Contabilidad, recursos humanos, control de viáticos, sistema de gestión de calidad, producción, etc).

Al conocer las ventajas que puede brindarle la implementación de inteligencia de Negocios para extraer la información de las distintas aplicaciones transaccionales que mantienen la operación de la organización. Analizan y buscarán el siguiente objetivo en la implementación de este proyecto:
“Conocer las ventas operacionales que se realizan en toda la organización (puntos de venta, distribuidores) identificando por las distintas variables que se entregan los productos (líneas de ropa, vendedores)”

Necesidades de Negocio:

- Lugares con mejores ventas
- Fechas de las mejores ventas
- Líneas de ropa mejor vendidas
- Ropa por genero de sexo mejor vendida
- Canal de negocio que más ventas genera

4. DATA WAREHOUSE

La inteligencia de Negocios y su enfoque ha tenido una interesante dinámica tanto académica como industrial en los últimos años. Uno de los conceptos claves que más ha tomado fuerza es el repositorio de datos, también conocido como data warehouse.

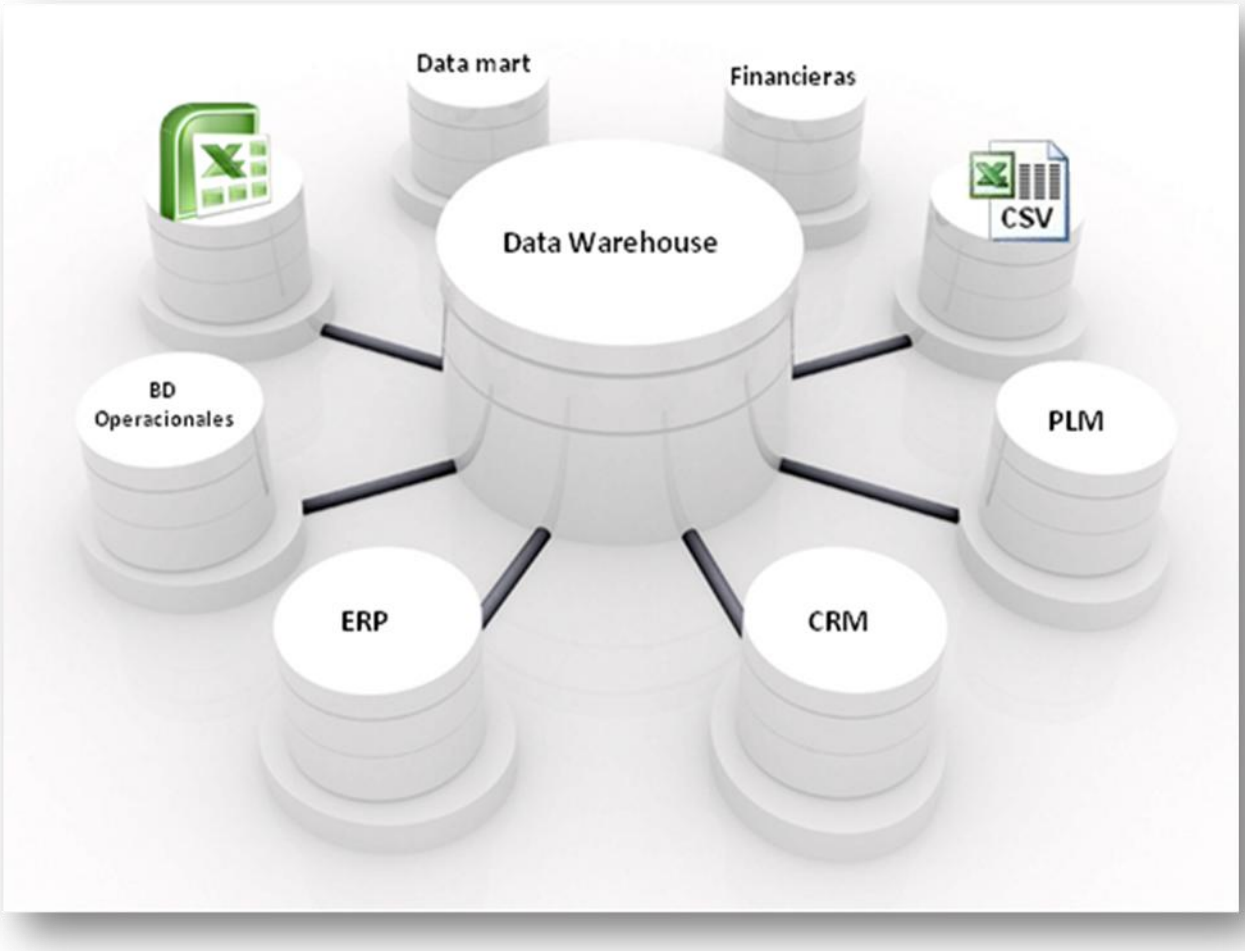


Figura 2. Data Warehouse [Fuente: Diseño propio a partir de imágenes en internet]

Este capítulo se enfoca en la introducción de la metodología ya consolidada en múltiples proyectos y las cuales han dado la evolución actual de este concepto.

El objetivo de este capítulo es la introducción del concepto de data warehouse y la aplicación de la metodología usando un ejemplo con una solución open source, en este caso Pentaho Data Integration y cómo repositorio de datos el motor de base de datos Oracle Express edition.

4.1 La base de un sistema de Inteligencia de Negocios: La Bodega de datos

Un data warehouse[10] es un repositorio de datos que proporciona una visión global, común e integrada de los datos de la organización, con las siguientes propiedades:

- Estable
- Coherente
- Fiable
- Con información histórica

En distintos artículos de internet identifican como funcionalidades de un sistema de datawarehuse tales como: [Fuente: (Velasco, 2004)]

1. Integración de bases de datos heterogéneas (relacionales, documentales, geográficas, archivos, etc.).

2. Ejecución de consultas complejas no predefinidas visualizando el resultado en forma de gráfica y en diferentes niveles de agrupamiento y totalización de datos.

3. Agrupamiento y desagrupamiento de datos en forma interactiva.

4. Análisis de problema en términos de dimensiones. Por ejemplo, permite analizar datos históricos a través de una dimensión tiempo.

5. Control de calidad de datos para asegurar, no solo la consistencia de la base, sino también la relevancia de los datos en base a los cuales se toman las decisiones.

Al abarcar el ámbito global de la organización y con un amplio alcance histórico, el volumen de datos puede ser demasiado grande (centenas de terabytes). Las bases de datos relacionales son el soporte técnico más comúnmente usado para almacenar las estructuras de estos datos y sus grandes volúmenes. [Fuente: **(Curto, INFORMATION MANAGEMENT, 2007)**] Presenta las siguientes características:

- *Orientado a un tema:* organiza una colección de información alrededor de un tema central. Las transacciones operacionales están diseñadas alrededor de aplicaciones y funciones, como por ejemplo pagos, ventas, entregas de mercadería, para una institución comercial. Un Data Warehouse está organizado alrededor de los temas más globales, como cliente, vendedor, producto y actividades.
- *Integrado:* incluye datos de múltiples orígenes y presenta consistencia de datos. Cuando los datos son copiados del ambiente operacional, son integrados antes de entrar en el date warehouse. Por ejemplo, un diseñador puede representar el sexo como "M" y "F", otro puede representarlo como "0" y "1", o "x" e "y", y otro usar las

[10] Según W. H. Inmon (considerado por muchos el padre del concepto), un data warehouse es un conjunto de datos orientados por temas, integrados, variantes en el tiempo y no volátiles, que tienen por objetivo dar soporte a la toma de decisiones.

palabras completas "masculino" y "femenino". No importa la fuente de la cual el sexo llegue al data warehouse, debe ser guardado en forma consistente.

- *Variable en el tiempo:* se realizan fotos de los datos basadas en fechas o hechos. Los datos en la bodega de datos son precisos para un cierto momento, no necesariamente ahora; por eso se dice que los data warehouse son variantes en el tiempo. La varianza en el tiempo de los datos de un warehouse se manifiestan de muchas maneras. La bodega de datos contiene datos de un largo horizonte de tiempo. Las aplicaciones operacionales, sin embargo, contienen datos de intervalos de tiempo pequeños, por cuestiones de performance (tamaño de las tablas). Toda estructura clave en un data warehouse contiene implícita o explícitamente un elemento del tiempo. Esto no necesariamente pasa en el ambiente operacional.
- *No volátil:* sólo de lectura para los usuarios finales. Updates, inserts y deletes son efectuados regularmente, en una base de datos transaccional. La manipulación de datos en un data warehouse, es mucho más sencilla. Solo ocurren dos operaciones, la carga inicial, y el acceso a los datos. No hay necesidad de updates (en su sentido general).

Frecuentemente la bodega de datos está constituido por una base de datos relacional, pero no es la única opción factible, también es posible considerar las bases de datos orientadas a columnas o incluso basadas en lógica asociativa. [Fuente: **(Díaz, Introducción al Bussines Intelligence, 2010)**]

Se debe tener en cuenta que existen otros elementos en el contexto de una bode de datos:

- Data Warehousing: es el proceso de extraer y filtrar datos de las operaciones comunes de la organización, procedentes de los distintos sistemas de información operacionales y/o sistemas externos, para transformarlos, integrarlos y almacenarlos en un almacén de datos con el fin de acceder a ellos para dar soporte en el proceso de toma de decisiones de una organización.
- Data Mart: es un subconjunto de los datos de la bodega de datos cuyo objetivo es responder a un determinado análisis, función o necesidad, con una población de usuarios específica. Al igual que en un data warehouse, los datos están estructurados en modelos de estrella o copo de nieve, y en un data mart puede ser dependiente o independiente de un data warehouse. Por ejemplo, un posible uso sería para la minería de datos o para la información de marketing. El data mart está pensado para cubrir las necesidades de un grupo de trabajo o de un determinado departamento dentro de la organización.
- Operational Data Store: es un tipo de almacén de datos que proporciona sólo los últimos valores de los datos y no su historial; además, generalmente admite un pequeño desfase o retraso sobre los datos operacionales.
- Staging Area: es el sistema que permanece entre las fuentes de datos y la bodega de datos con el objetivo de:

    - Facilitar la extracción de datos desde fuentes de origen con una heterogeneidad y complejidad grande.
    - Mejorar la calidad de datos
    - Ser usado como caché de datos operacionales con el que posteriormente se realiza el proceso de data warehousing
    - Uso de la misma para acceder en detalle a información no contenida en la bodega de datos
- Procesos ETL: tecnología de integración de datos basada en la consolidación de datos que se usa tradicionalmente para alimentar data warehouse, data mart, staging area y ODS. Usualmente se combina con otras técnicas de consolidación de datos.
- Metadatos: datos estructurados y codificados que escriben características de instancias; aportan informaciones para ayudar a identificar, descubrir, valorar y administrar las instancias descritas.

4.2 Elementos de un Data warehouse

La estructura relacional de una base de datos operacional sigue las formas normales en su diseño. Un data warehouse no debe seguir ese patrón de diseño. La idea principal es que la información sea presentada desnormalizada para optimizar las consultas. Para ello se debe identificar, en el seno de la organización, los procesos de negocio, las vistas para el proceso de negocio y las medidas cuantificables asociadas a los mismos. De esta manera se habla de:

- *Tabla de Hecho:* es la representación en la bodega de datos de los procesos de negocio de la organización. Por ejemplo, una venta puede identificarse como un proceso de negocio de manera que es factible, si corresponde en la organización, considerar la tabla de hecho de ventas.

- *Dimensión*: es la representación en la bodega de datos de una vista para un cierto proceso de negocio. Retomando el ejemplo de una venta, para la misma se tiene el cliente que ha comprado, la fecha en que se ha realizado. Estos conceptos pueden ser considerados como vistas para este proceso de negocio. Puede ser interesante recuperar todas las compras realizadas por un cliente. Ello hace entender por qué se identifica como una dimensión.

- *Métrica:* son los indicadores de negocio de un proceso de negocio. Aquellos conceptos cuantificables que permiten medir este proceso de negocio. Por ejemplo, en una venta se tiene el importe de la misma.

4.4.1 Tipos de Tablas de Hecho

A nivel de diseño una tabla de hecho es aquella que permite guardar dos tipos de atributo diferenciados:

- Medidas del proceso/actividad/flujo de trabajo/evento que se pretende modelizar.
- Claves foráneas hacia registros en una tabla de dimensión (en otras palabras, hacia una vista de negocio)

Existen diferentes tipos de tablas de hecho:

- *Transaction Fact Table:* representan eventos que suceden en un determinado espacio-tiempo. Se caracterizan por permitir analizar los datos con el máximo detalle. Por ejemplo, se puede pensar en una venta que tiene como resultado métricas como el importe de la misma.
- *Factless Fact Tables/Coverage Table:* son tablas que no tienen medidas, y tiene sentido dado que representan el hecho de que el evento suceda. Frecuentemente se añaden contadores a dichas tablas para facilitar las consultas SQL. Por ejemplo, se puede pensar en la asistencia en un acto benéfico en el que por cada persona que asiste se tiene un registro pero se podría no tener ninguna métrica asociada más.
- *Periodic Snapshot Fact Table:* Son tablas de hecho usadas para recoger información de forma periódica a intervalo de tiempo regulares. Dependiendo de la situación medida o la necesidad de negocio, este tipo de tablas de hecho son una agregación de las anteriores o están diseñadas específicamente. Por ejemplo, se puede pensar en el balance mensual. Los datos se recogen acumulados de forma mensual.
- *Accumulating Snapshot Fact Table:* representan el ciclo d evida completo –con un principio y un final- de una actividad o un proceso. Se caracterizan por tener múltiples dimensiones relacionadas con los eventos presentes en un proceso. Por ejemplo, se puede pensar en un proceso de matrícula de un estudiante: recopila datos durante su periodo de vida que suelen sustituir los anteriores (superación y recopilación de asignaturas, por ejemplo).

[Fuente: **(Curto, INFORMATION MANAGEMENT, 2008)**]

4.4.2 Tablas de Dimensiones

Las dimensiones recogen los puntos de análisis de un hecho. Por ejemplo, una venta se puede analizar en función del día de venta, producto, cliente, vendedor o canal de venta, entre otros. Respecto al punto de vista de la gestión histórica de los datos, éstos se pueden clasificar como:

- *SCD [11]Tipo 0:* no se tiene en cuenta la gestión de los cambios históricos y no se realiza esfuerzo alguno. Nunca se cambia la información, ni se reescribe.

[11] SCD son las siglas de Slowly Changing Dimensión, y se refiere a la política de actualización de datos en una dimensión.

- *SCD Tipo 1:* No se guardan datos históricos. La nueva información sobrescribe la antigua siempre. La sobrescritura se realiza, principalmente, por errores de calidad de datos. Este tipo de dimensiones son fáciles de mantener, y se usan cuando la información histórica no es importante.
- *SCD Tipo 2:* Toda la información histórica se guarda en la bodega de datos. Cuando hay un cambio se crea una nueva entrada con fecha y surrogate key apropiadas. A partir de ese momento será el valor usado para futuras entradas. Las antiguas usarán el valor anterior.
- *SCD Tipo 3:* Toda la información histórica se guarda en la bodega de datos. En este caso se crean nuevas columnas con los valores antiguos y los actuales son remplazados con los nuevos.
- *SCD Tipo 4:* Es lo que se conoce habitualmente como tablas históricas. Existe una tabla con los datos actuales y otra con los antiguos o los cambios.
- *SCD Tipo 6/Híbrida:* Combina las aproximaciones de los tipos 1, 2 y 3 (y, claro, entonces 1+2+3=6). Consiste en considerar una dimensión de tipo 1 y añade un par de columnas adicionales que indican el rango temporal de validez de una de las columnas de la tabla. Si bien su diseño es complejo, entre sus beneficios se puede destacar que reduce el tamaño de las consultas temporales. Existe otra variante para este tipo de dimensión que consiste en tener versiones del registro de la dimensión (numerados de 0 a n+1, donde 0 siempre es la versión actual).

[Fuente: **(Curto, INFORMATION MANEGEMENT, 2008)**]

Existen otros tipos de dimensiones cuya clasificación es funcional:

- **Degenaradas:** se encuentran como atributos en la tabla de hecho, si bien tiene el significado de un punto de vista de análisis. Contiene información de baja cardinalidad formada por relaciones dicotómicas. Frecuentemente contienen sólo un atributo y, por ello, no se crea una tabla aparte. Por ejemplo, el sexo de un paciente.
- **Monster:** es conveniente comentar que algunas dimensiones pueden crecer desmesuradamente. Una buena práctica es romper la dimensión en dos tablas: una que contenga los valores estáticos y otra que contenga los valores volátiles. Un ejemplo claro puede ser la información de cliente. Se debe ser conscientes de cuál es la información primordial del mismo y cuál la que sólo se usa puntualmente en los informes u otros análisis.
- **Junk:** que contiene información volátil que se usa puntualmente y que no se guarda de forma permanente en la bodega de datos.
- **Conformadas:** que permite compartir información entre dimensiones. Consiste en dimensiones definidas correctamente para que sean usadas por dos tablas y poder así realizar consultas comunes. El ejemplo más fácil es la dimensión temporal.
- **Bridge:** que permiten definir relaciones n a m entre tablas de hecho. Necesarias para definir por la relación entre un piloto y sus múltiples patrocinadores.
- **Role-playing:** que tienen asignado un significado. Por ejemplo, se puede tener la dimensión fecha, pero también fecha de entrega.

- **Alta cardinalidad:** que contienen una gran cantidad de datos difícilmente consultables en su totalidad. Por ejemplo, cada uno de los habitantes de un país.

4.4.3 Tipos de Métricas

Se puede distinguir diferentes tipos de medidas, basadas en el tipo de información que recopilan así como su funcionalidad asociada:

- *Métricas:* valores que recogen el proceso de una actividad o los resultados de la misma. Estas medidas proceden del resultado de la actividad de negocio.
    - *Métricas de realización de actividad (leading):* miden la realización de una actividad. Por ejemplo, la participación de una persona en un evento.
    - *Métricas de resultado de una actividad (lagging):* recogen los resultados de una actividad. Por ejemplo, la cantidad de puntos de un jugador en un partido.
- *Indicadores clave***:** valores correspondientes que hay que alcanzar y que suponen el grado de asunción de los objetivos. Estas medidas proporcionan información sobre el rendimiento de una actividad o sobre la consecución de una meta.
    - *Key Performance Indicator (KPI):* indicadores clave de rendimiento. Más allá de la eficacia, se definen unos valores que explican en qué rango óptimo de rendimiento se debería situar al alcanzar los objetivos. Son métricas del proceso.
    - *Key Goal Indicator (KGI):* indicadores de metas. Definen mediciones para informar a la dirección general si un proceso TIC ha alcanzado su requisito de negocio, y se expresan por lo general en términos de criterios de información.

[Fuente: **(Curto, INFORMATION MANAGEMENT, 2008)**]

Se debe añadir que existen también indicadores de desempeño. Los indicadores clave de desempeño (en definitiva, son KPI) definen mediciones que determinan cómo se está desempeñando el proceso de TI para alcanzar la meta. Son los indicadores principales que señalan si será factible lograr una meta o no, y son buenos indicadores de las capacidades, prácticas y habilidades. Los indicadores de metas de bajo nivel se convierten en indicadores de desempeño para los niveles altos.

4.3 **Esquemas para un Data warehouse**

Existen principalmente dos tipos de esquemas para estructurar los datos en un almacén de datos:

- *Esquema en estrella:* A nivel de diseño, consiste en una tabla de hechos (o lo que en los libros se encontrará como fact table) en el centro para el hecho objeto de análisis y una o varias tablas de dimensión por cada punto de vista da análisis que participa de la descripción de ese hecho. Consiste en estructurar la información en procesos, vistas y métricas recordando a una estrella (por ello el nombre). En la tabla de hecho

se encuentran los atributos destinados a medir (cuantificar): sus métricas. La tabla de hechos solo presenta uniones con dimensiones. (Curto, INFORMATION MANAGEMENT, 2007)

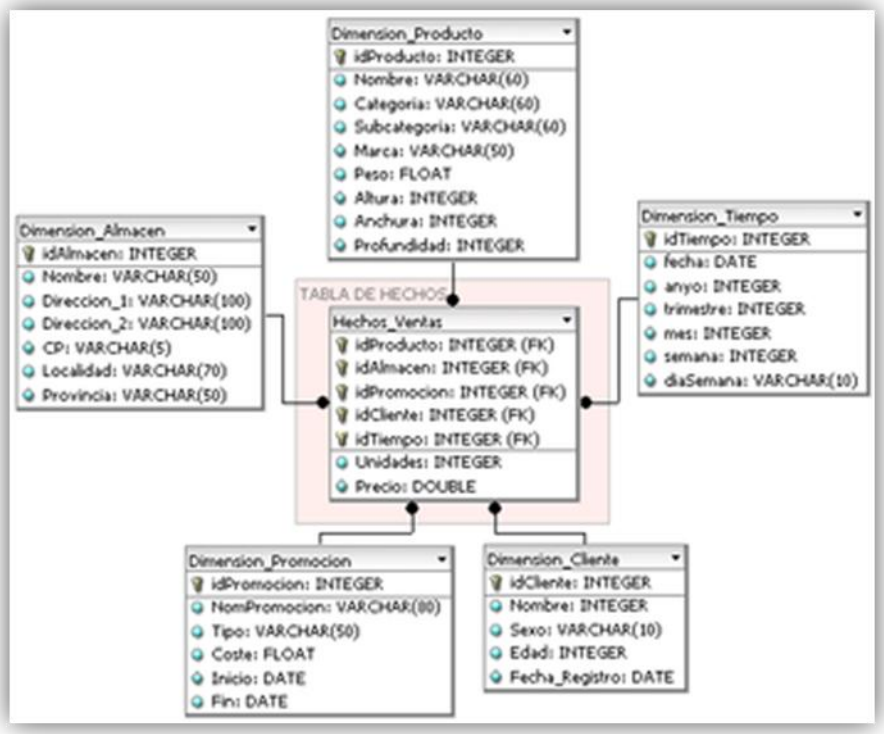


Figura 3. Esquema en estrella [Fuente: **(Wikipedia, 2008)**]

- *Esquema en copo de nieve:* es un esquema de representación derivado del esquema de estrella, en el que las tablas de dimensión se normalizan en múltiples tablas. Por esta razón, la tabla de hechos deja de ser la única tabla del esquema que se relaciona con otras tablas y aparecen nuevas uniones. Es posible identificar dos tipos de esquemas en copo de nieve: [Fuente: (Curto, INFORMATION MANAGEMENT, 2007)]
  - *Completo:* en el que todas las tablas de dimensión en el esquema de estrella aparecen ahora normalizadas.
  - *Parcial:* sólo se lleva a cabo la normalización de algunas de ellas.

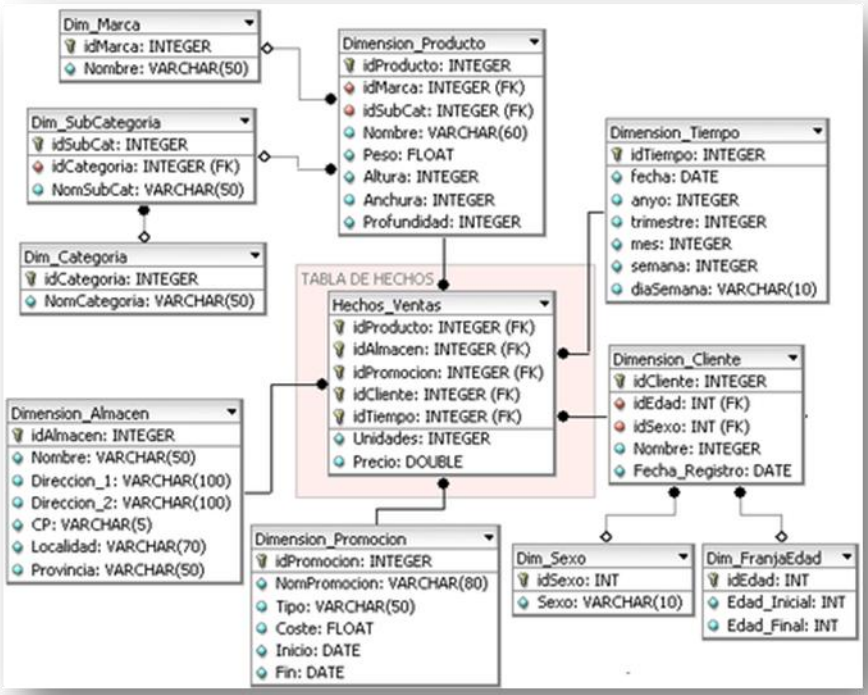


Figura 4. Esquema en copo de nieve [Fuente: **(Wikipedia, 2008)**]

4.4 Metodología para diseñar un Data Warehouse: MODELIZACIÓN

"Quién adelante no mira, atrás se queda"
Refrán anónimo

Para efectos prácticos se diseñará a partir de data marts, con el objetivo de ir cubriendo las necesidades por departamento de la organización. La consolidación de data marts de los distintos departamentos dará origen al data warehouse de la organización.
La metodología para diseñar propuesta se basa en la identificación de modelos de diseños conceptuales, lógicos y físicos de los datos:

4.4.1 Modelo Conceptual de datos

El modelo conceptual se basa en identificar qué tipo de procesos y vistas de negocio proporciona las respuestas a las preguntas que tienen los usuarios finales. Normalmente en esta fase, se debe ser previsor y pensar más allá de las necesidades actuales y de poder cubrir las futuras.
Dado que la información transaccional puede que se encuentre en diferentes plataformas, no todas las bases de datos están unificadas. Ello dificulta la consolidación de información. Cuando esto sucede, lo habitual es crear una staging area. Para mayor acerca de stanging area consultar la siguiente fuente bibliográfica: **(Curto,**

INFORMATION MANAGEMENT, 2010). El objetivo de crear la staging area es facilitar el proceso de transformación de datos. Los beneficios de ésta son:

- Realizar otro tipo de análisis a posteriori distinguiendo los datos mediante ciertos criterios.
- Mejorar el rendimiento de la carga de datos al realizar la carga previa a la staging area y, a posteriori, realizar las transformaciones de los datos entre bases de datos.

En el caso de crear una staging area, su modelo sería:
En el momento de hacer el diseño conceptual es necesario identificar las tablas de hecho y las dimensiones que se pueden deducir.

Para el caso práctico, teniendo en cuenta la información que se extrae se identifica para el ejemplo una tabla de hecho o proceso del negocio: **la venta**. Cada transacción se traduce en una venta.
Cada venta puede analizarse desde diferentes puntos de vista (lo que proporciona las dimensiones del proceso de negocio):

- Dimensión Localización: El lugar físico donde se encuentra el medio de venta
- Dimensión Canal: El tipo de medio de venta (Distribuidores o ventas directas)
- Dimensión Genero: Hace referencia al sexo para el cual está diseñado el producto vendido.
- Dimensión Línea de venta: La línea de ropa vendida (Casual o Formal)
- Dimensión Tiempo: la fecha en que se realiza la venta.

De modo que se obtiene el siguiente diseño Conceptual:

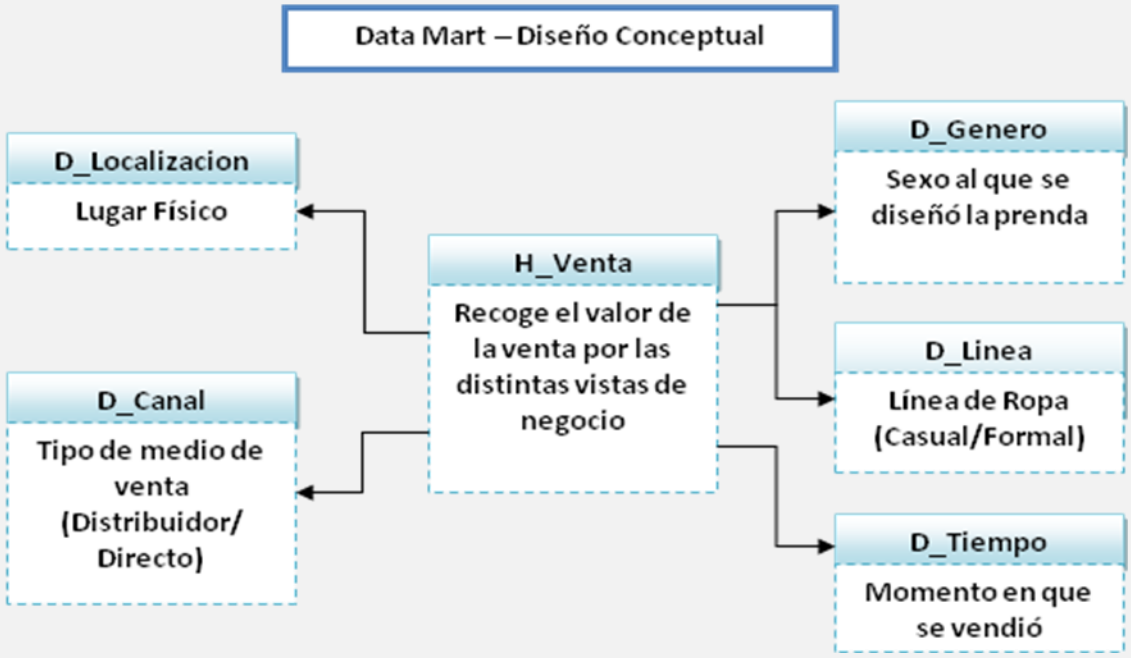


Figura 5. Diseño conceptual Data mart [Fuente: Diseño propio]

4.4.2 Modelo Lógico de Datos

Después del modelo conceptual, a través del cual se ha identificado las tablas de hecho y las dimensiones, es necesario realizar el diseño lógico con el que se identifican las métricas de las tablas de hecho y los atributos de las dimensiones.
La tabla de hecho contiene la clave subrogada que identifica de manera única cada registro, las claves foráneas a las dimensiones relacionadas con la tabla de hecho y las métricas.
En la siguiente fuente bibliográfica se puede ampliar mayor información relacionada con la descripción de un modelo lógico de datos: **(Franco, 2010)**

Se considera la medida más natural en este caso práctico para la métrica: el valor de las ventas y la cantidad vendida.

Así, de esta manera, para la tabla de hecho h_venta se tiene:

Tabla de Hecho	Claves foráneas	Métricas
H_Venta	Id_localizacion, id_canal, id_genero, id_linea, id_tiempo	Valor de ventas, Cantidad vendida

Tabla 2. Tabla de Hecho Ventas [Fuente: Diseño Propio]

Y los atributos de cada una de las dimensiones:

Dimensión	Clave Primaria	Atributos
D_Localizacion	Id_localizacion	Id_pais, nomb_pais, id_ciudad, nomb_ciudad
D_Canal	Id_Canal	Nomb_canal, id_ptoventa, nomb_ptoventa
D_Genero	Id_genero	Nomb_genero
D_Linea	Id_linea	Nomb_linea
D_Tiempo	Id_tiempo	Num_anno, num_mes, nomb_mes

Tabla 3. Atributos de Dimensiones [Fuente: Diseño propio]

Por lo que el diseño lógico resultante es el siguiente esquema en estrella:

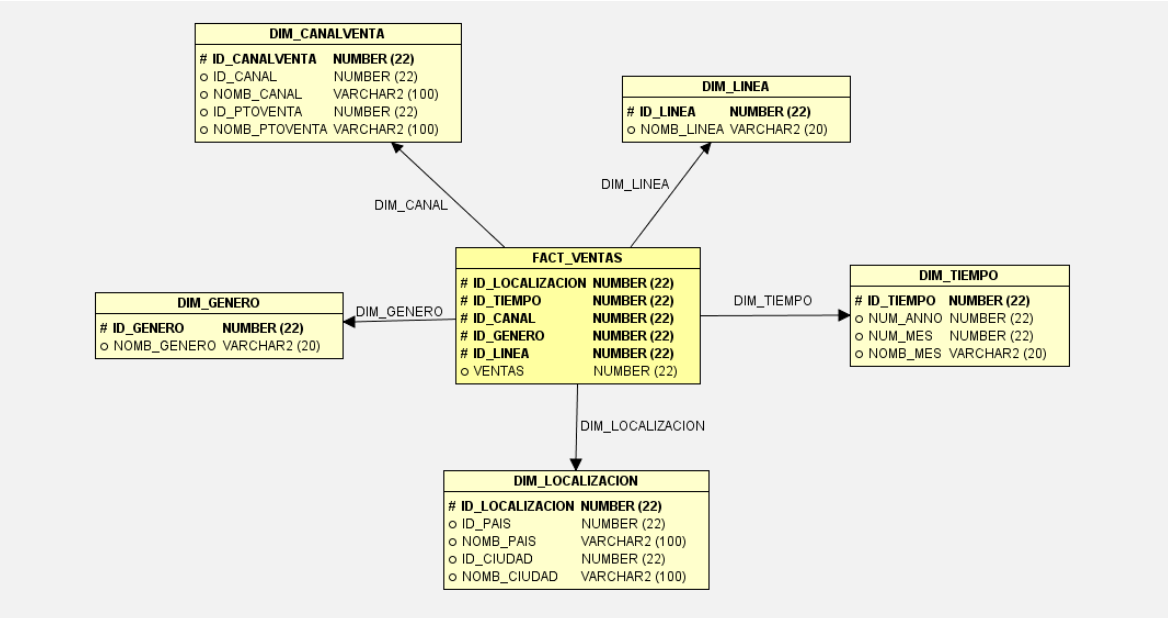


Figura 6. Diseño Lógico Data mart [Fuente: Wikipedia 2008]

4.4.3 Modelo Físico de Datos

El siguiente paso es el diseño físico. Donde se definirá que motor de base de datos es la que creará el Data mart o Data Warehouse.
Un Data mart o Data Warehouse está conformado por una colección de tablas. El objetivo en este paso es definir, para cada tabla, el formato de cada clave y atributo.
Se debe recordar los siguientes criterios:

- Se recomienda que las claves sean enteros y que sean independientes de las fuentes de origen.
- Las métricas pueden ser aditivas (números), semiaditivas (con particularidades en el momento de agrupar las cantidades) o no aditivas (que entonces se estaría hablando de atributos cualitativos). En las tablas de hecho se debería decantar porque todas las métricas fueran aditivas.
- En un caso real, sería necesario incluir campos que permitieran la trazabilidad del dato, por ejemplo fecha de carga, fecha de modificación, autor, fuente de origen. Para simplificar el modelo, no se incluyen.

Para ampliar la conceptualización del modelo físico de datos, está disponible en varias fuentes, la basada para éste documento es: **(Chavez, 2005)**

En este caso, se trabajará con Oracle XE como la base de datos será usada como data mart y con una herramienta de modelización de base de datos. En este caso particular se va a usar Oracle Designer.

Si se consideran cada una de las tablas, entonces el diseño físico resultante es:

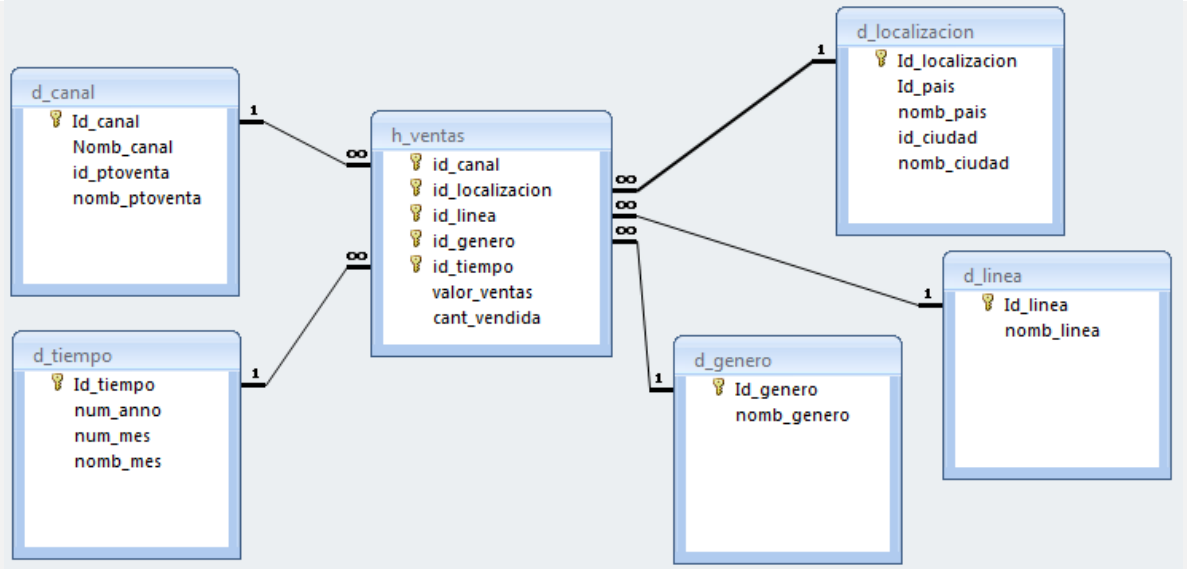


Figura 7. Diseño Físico Data mart [Fuente: Wikipedia 2008]

Este diseño es un ejemplo que puede extenderse para responder muchas más preguntas, y por lo tanto, incluir mucha más información consolidada.

5. INTEGRACIÓN DE DATOS: ETL

Se consideran las siguientes áreas, cuando se refiere a integración en un contexto empresarial:

- ***Integración de datos:*** proporciona una visión única de todos los datos de negocio, sin importar su ubicación. Este es el ámbito de la inteligencia de negocio.
- ***Integración de aplicaciones:*** proporciona una visión unificada de todas las aplicaciones tanto internas como externas a la empresa. Esta integración se consigue mediante la coordinación de los flujos de eventos (transacciones, mensaje o datos) entre aplicaciones.
- ***Integración de procesos de negocio:*** proporciona una visión unificada de todos los procesos de negocio. Su principal ventaja es que las consideraciones de diseño del análisis e implementación de los procesos de negocio son aislados del desarrollo de las aplicaciones.
- ***Integración de la interacción de los usuarios:*** proporciona una interfaz segura y personalizada al usuario del negocio (datos, aplicaciones y procesos de negocio).

Figura 8. Integración de datos: ETL [Fuente: http://es.123rf.com]

Este capítulo se centrará en la integración de datos en general y en los procesos ETL[12] (Extracción, Trasformación y Carga) en particular, que una de las tecnologías de integración de datos que se usa en los proyectos de implantación de Bussiness Intelligence.

[12] **Extract, Transform and Load** (Extraer, transformar y cargar en inglés, frecuentemente abreviado a **ETL**) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

El objetivo de este capítulo es conocer las diferentes opciones de integración de datos en el ámbito de la inteligencia de negocio y, en particular, conocer el diseño de procesos ETL.

5.1 Integración de datos

Dentro del contexto de la inteligencia de negocios, las herramientas ETL han sido la opción usual para alimentar la bodega de datos. La funcionalidad básica de estas herramientas está compuesta por:

- Gestión y administración de servicios
- Extracción de datos
- Transformación de datos
- Carga de datos
- Gestión de datos

En la siguiente gráfica se ilustran los distintos componentes del proceso de ETL para un bodega de datos o data warehouse:

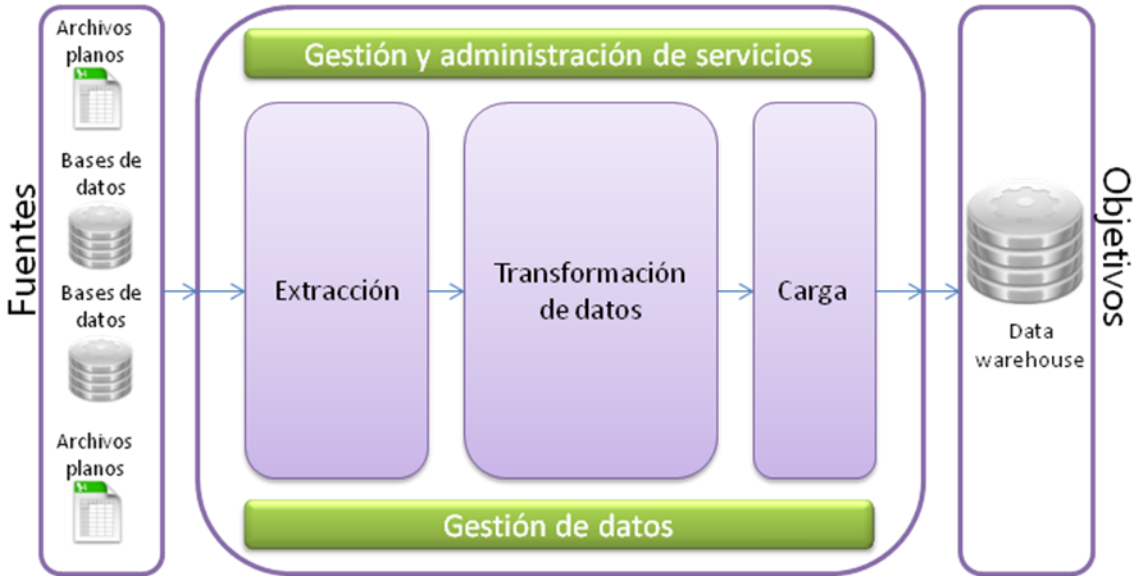


Figura 9. Proceso Integración de datos [Diseño propio a partir de Imágenes en Internet]

En los últimos años, estas herramientas han evolucionado incluyendo más funcionalidades propias de una herramienta de integración de datos. Se puede destacar:

- Servicios de acceso/entrega de datos (vía adaptadores/conectores)
- Gestión de servicios
- Perfiles de datos (Data profiling)
- Calidad de datos (Data quality)
- Procesos operacionales

- Servicios de transformación: CDC (Captura de datos modificados), SCD (Dimensiones de variación lenta), validación, agregación.
- Servicios de acceso a tiempo real
- Extract, Transform and Load (ETL)
- Integración de información empresarial (EII de sus siglas en inglés Enterprise Information Integration)
- Integración de aplicaciones empresariales (EAI de sus siglas en inglés Enterprise Applications Integration)
- Capa de transporte de datos
- Gestión de datos

En la siguiente gráfica se detalla la integración de las funcionalidades descritas en la evolución de los procesos ETL:

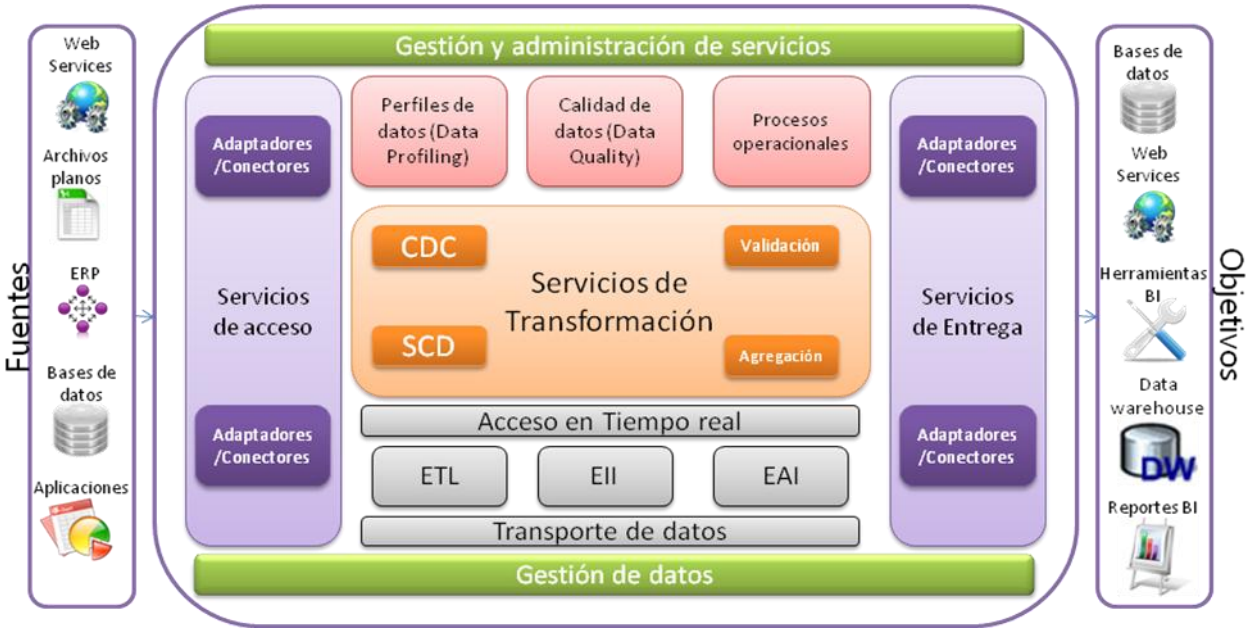


Figura 10. Suite Integración de datos [Fuente: Diseño propio basado en imagen de **(Díaz, Introducción al Bussines Intelligence, 2010)**]

Esta evolución es consecuencia de diversos motivos, entre los que se puede destacar los diferentes tipos de datos que existen:

- *Estructurados:* contenidas en bases de datos
- *Semiestructurados:* en formatos legibles para máquinas, si bien no están completamente estructurados: HTML tabulado, Excel, CSV…, que pueden obtenerse mediante técnicas estándar de extracción de datos.
- *No estructurados:* en formatos legibles para humanos, pero no para máquinas: Word, HTML no tabulado, PDF…, que pueden obtenerse mediante técnicas avanzadas como text mining u otras.

[Fuente: **(Díaz, Introducción al Bussines Intelligence, 2010)**]
Así como la evolución de las necesidades de negocio.
Por ello el punto de partida adecuado es definir formalmente el concepto de integración de datos:

Se entiende por **integración de datos** al conjunto de aplicaciones, productos, técnicas y tecnologías que permiten una visión única consistente de los datos de negocio. [Fuente: **(Inmon, 2005)**]

Respecto a la definición:

- Las aplicaciones son soluciones a medida que permiten la integración de datos en base al uso de productos de integración.
- Los productos comerciales desarrollados por terceros capacitan la integración mediante el uso de tecnologías de integración.
- Las tecnologías de integración son soluciones para realizar la integración de datos.

5.1.1 Técnicas de integración de datos

Existen diferentes técnicas de integración de datos: [Fuente: **(McBurney, 2008)**]

- **Propagación de datos:** Consiste en copiar datos de un lugar de origen a un entorno destino local o remoto. Los datos pueden extraerse del origen mediante programas que generen un archivo que debe ser transportado al destino, donde su utilizará como archivo de entrada para cargar en la base de datos de destino. Una aproximación más eficiente es descargar sólo los datos que han cambiado en origen respecto a la última propagación realizada, generando un archivo de carga incremental que también será transportado al destino. Este tipo de procesos son habitualmente de tipo en línea y trabajan con una arquitectura push[13]. Puede realizarse como:
  - Distribución
  - Intercambio bidireccional. Puede ser master-slave o peer-to-peer.

- **Consolidación de datos:** Consiste en capturar los cambios realizados en múltiples entornos de origen y propagarlos a un único entorno destino, donde se almacena una copia de todos estos datos. Ejemplos son un data warehouse o un

[13] La técnica push consiste en la actualización continua en línea del entorno destino mediante aplicaciones de integración de datos que capturan los cambio en origen y los trasmiten a destino, donde son almacenados, en la que los datos son automáticamente enviados al entorno remoto.

ODS, alimentado por varios entornos de producción. Con esta técnica es difícil trabajar con tiempos de latencia[14] bajos:

- Cuando no se requiere latencia baja, se suele proveer los datos mediante procesos batch en intervalos prefijados (superior a varias horas). Se usan consultas SQL para conseguir los datos (lo que se denomina técnica pull).
- Cuando se requiere latencia baja, se utiliza la técnica push. En este caso, la aplicación de integración de datos debe identificar los cambios producidos en origen para transmitir sólo esos cambios, y no todo el conjunto de datos del origen. Para ello, se suele emplear algún tipo de técnica de tipo CDC (change data capture).

- **Federación de datos:** Proporciona a las aplicaciones una visión lógica virtual común de una o más bases de datos. Esta técnica permite acceder a diferentes entornos origen de datos, que pueden estar en los mismos o en diferentes gestores de datos y máquinas, y crear una visión de este conjunto de bases de datos como si fuese en la práctica una base de datos única e integrada. Cuando una aplicación de negocio lanza una consulta SQL contra esta vista virtual, el motor de federación de datos descompone la consulta en consultas individuales para cada uno de los orígenes de datos físicos involucrados y la lanza contra cada uno de ellos. Cuando ha recibido todos los datos respuesta a las consultas, integra los resultados parciales en un resultado único, realizando las sumatorias, agregaciones y/o ordenaciones necesarias para resolver la consulta original, y devuelve los datos a la aplicación que lanzó la petición original. Uno de los elementos claves del motor de federación es el catálogo de datos común. Este catálogo contiene información sobre los datos: su estructura, su localización y, en ocasiones, su demografía (volumen de datos, cardinalidad de las claves, claves de clustering, etc). Ello permite que se pueda optimizar la división de la consulta original al enviarla a los gestores de bases de datos, y que se elija el camino más eficiente de acceso global a los datos.

- **CDC (Change Data Capture):** Se utiliza para capturar los cambios producidos por las aplicaciones operacinales en las bases de datos de origen, de tal manera que pueden ser almacenados y/o propagados a entornos destino para que éstos mantengan la consistencia con los entornos origen. A continuación se listan las cuatro principales técnicas del CDC:
  - *CDC por aplicación:* consiste en que la propia aplicación es la que genera la actualización de datos en origen, y se encarga de actualizar directamente los entornos destino, o almacenar localmente los cambios en

[14] En redes informáticas de datos, se denomina latencia la suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red.

una tabla de paso (staging) mediante una operación de INSERT dentro de la misma unidad lógica de trabajo.

- *CDC por timestamp:* se puede emplear cuando los datos de origen incorporan un timestamp (por ejemplo a nivel de fila si el origen es una tabla relacional) de la última actualización de ésta. El CDC se limitará a escanear los datos de origen para extraer los datos que posean un timestamp posterior al de la última vez que se ejecutó el proceso de CDC: estos datos son los que han cambiado desde la última captura de datos y, por tanto, son los que deben actualizarse en los entornos destino.
- *CDC por triggers:* los triggers o disparadores son acciones que se ejecutan cuando se actualizan (por UPDATE, DELETE o INSERT) los datos de una determinada tabla sobre la que están definidos. Esos triggers pueden utilizar estos datos de la actualización en sentencias SQL para generar cambios SQL en otras tablas locales o remotas. Por lo tanto, una forma de capturar cambios es crear tiggers sobres las tablas de origen, cuyas acciones modifiquen los datos de las tablas destino.
- *CDC por captura de log:* consiste en examinar constantemente el archivo de log de la base de datos de origen en busca de cambios en las tablas que se deben monitorizar. Estos programas basan su eficiencia en la lectura de buffers de memoria de escritura en el log, por lo que la captura de la información no afecta el rendimiento del gestor relacional al no requerir acceso al disco que contiene el archivo de log.

- **Técnicas hibridas:** la técnica elegida en la práctica para la integración de datos dependerá de los requisitos de negocio para la integración, pero también en gran medida de los requisitos tecnológicos y de las probables restricciones presupuestales. A la práctica se suelen emplear técnicas de integración constituyendo lo que se denomina una técnica hibrida.

5.1.2 Tecnologías de integración de datos

Existen diferentes tecnologías de integración de datos basadas en las técnicas presentadas:

- **ETL*:*** permite extraer datos del entorno origen, transformarlos según nuestras necesidades de negocio para integración de datos y cargar estos datos en los entornos destino. Los entornos origen y destino son usualmente bases de datos o archivos, pero en ocasiones también pueden ser colas de mensajes de un determinado middleware, así como archivos u otras fuentes estructuradas, semi-estructuradas o no estructuradas. Está basada en técnicas de consolidación. Las herramientas de ETL en la práctica mueven o transportan datos entre entornos origen y destino, pero también documentan como estos datos son transformados

(si lo son) entre el origen y el destino almacenando esta información en un catalogo propio de metadatos; intercambian estos metadatos con otras aplicaciones que puedan requerirlos y administran todas las ejecuciones y procesos de la ETL: planificación del transporte de datos, log de errores, los de cambios y estadísticas asociadas a los procesos de movimiento de datos. Este tipo de herramientas suelen tener un interfaz de usuario de tipo GUI y permiten diseñar, administrar y controlar cada uno de los procesos del entorno ETL.

- *ETL de generación de código:* Consta de un entorno grafico donde se diseñan y especifican los datos de origen, sus transformación y los entornos destino. El resultado generado es un programa de tercera generación (típicamente COBOL) que permite realizar las transformaciones de datos. Aunque estor programas simplifican el proceso ETL, incorporan pocas mejoras en cuanto al establecimiento y automatización de todos los flujos de procesos necesarios para realizar la ETL. Usualmente los administradores de datos los encargados de distribuir y administrar el código complicado, planificar y ejecutar los procesos en lotes, y realizar el transporte de los datos.
- *ETL basados en motor:* Permite crear flujos de trabajo en tiempo de ejecución definidos mediante herramientas graficas. El entorno grafico permite hacer un mapping de los entornos de datos de origen y destino, las transformaciones de datos necesarios, el flujo de procesos y los procesos por lotes necesarios. Toda esta información referente a diseño y procesos del ETL es almacenada en el repositorio del catalogo de metadatos. Se compone por diversos motores:

  a). Motor de extracción: utiliza adaptores como ODBC, JDBC, JNDI, SQL nativo, adaptadores de archivos planos u otros. Los datos pueden ser extraídos en modo pull planificado, típicamente soportando técnicas de consolidación en proceso por lotes, o mediante modo push, típicamente utilizando técnicas de propagación en procesos de tipo en línea. En ambos casos se pueden utilizar técnicas de changed data capture (CDC) ya vistas.

  b). Motor de transformación: Proporciona una librería de objetos que permite a los desarrolladores transformas los datos de origen para adaptarse a las estructuras de datos de destino, permitiendo, por ejemplo, la sumarizacion de los datos en destino en tablas resumen.
  c). *Motor de carga:* Utiliza adaptadores a los datos de destino, como el SQL nativo, o cargadores masivos de datos o archivos de destino.

  d). *Servicios de administración y operación:* Permiten la planificación, ejecución y monitorización de los procesos ETL, así como la visualización de eventos y la recepción y resolución de errores en los procesos.

- *ETL integrado en la base de datos:* algunos fabricantes incluyen capacidades ETL dentro del motor de la base de datos (al igual que lo hacen con otro tipo de características, como soporte OLAP y minería de datos). En general, presentan menos funcionalidades y complejidad, y son una solución menos completa que los ETL comerciales basados en motor o de generación de código. Por ello, a los ETL integrados en base de datos se les clasifica en tres clases en relación con los ETL comerciales (basados en motor o de generación de código):

  e). *ETL cooperativos:* con ellos, los productos comerciales pueden usar funciones avanzadas del gestor base de datos para mejorar los procesos de ETL. Ejemplos ETL cooperativos son aquellos que pueden utilizar procedimientos almacenados y SQL complejo para realizar las transformaciones de los datos en origen de una forma más eficiente, o utilizar paralelismo de CPU en consultas para minimizar el tiempo de los procesos ETL.

  f). *ETL complementarios:* Cuando los ETL de bases de datos ofrecen funcionalidades complementarias a los ETL comerciales. Por ejemplo, hay gestores de bases de datos que ofrecen soporte a MQT (Materializad Query Tables) o vistas de sumarizacion precalculadas, mantenidas y almacenadas por el gestor que puede usarse para evitar transformaciones de datos realizadas por ETL comercial. Además, otros gestores permiten la interacción directa mediante SQL con middleware de gestión de mensajes (por ejemplo, leyendo una cola de mensajes mediante un UDF o permitiendo la inserción de nuevos mensajes en colas mediante SQL) o con aplicaciones que se comunican mediante web services.

  g). *ETL competitivos:* algunos gestores ofrecen herramientas graficas integradas que explotan sus capacidades ETL en lo que claramente es competencia con los ETL comerciales.

- **EII:** el objetivo de la tecnología EII es permitir a las aplicaciones el acceso a datos dispersos (desde una data mart hasta archivo de texto o incluso web services) como si estuviesen todos residiendo en una base de datos común. Por lo tanto se basa en la federación. El acceso a datos dispersos implica la descomposición de la consulta inicial (habitualmente en SQL) direccionada contra la vista virtual federada en subcomponentes, que serán procesados en cada uno de los entornos donde residen los datos. Se recogen los resultados individuales de cada uno de los subcomponentes de la consulta, se combinan adecuadamente y se devuelve el resultado a la aplicación que lanzo la consulta. Los productos de EII han evolucionado desde dos entornos origen diferenciados: las bases de datos

relacionales y las bases de datos XML. Actualmente, la tendencia en productos EII es que soporten ambas interfaces a datos, SQL (ODBC y JDBC) y XML (XQuery y XPath). Los productos comerciales que implementan EII varían considerablemente en las funcionalidades que aportan; el área más diferenciadora es la optimización de las consultas distribuidas. Las características básicas de los productos que implementan soluciones de integración de datos EII son:

- *Transparencia*: los datos parecen estar en un origen único.
- *Heterogeneidad*: integración de datos de diferentes fuentes (relacionales, XML, jerárquicos) y también no estructurados.
- *Extensibilidad*: posibilidad de federar cualquier frente de datos.
- *Alta funcionalidad:* acceso con lectura y escritura a cualquier fuente soportada.
- *Autonomía*: acceso no disruptivo para los datos o las aplicaciones.
- *Rendimiento*: posibilidad de optimizar las consultas dependiendo del tipo y fuente de datos.

- **EDR**: Tiene el objetivo de detectar los cambios que suceden en las fuentes de origen esta soportada por las técnicas de integración de datos de CDC (chance data capture) y por la técnica de propagación de datos. Consta básicamente de los siguientes elementos:
  - Programa de captura: se encarga de recuperar los cambios producidos en la base de datos de origen. Esta captura puede ser realizada a través de una salida que lea constantemente el log de recuperación de la base de datos, a través de triggers o mediante una aplicación externa de usuario. El programa de captura se apoya en una serie de tablas donde se almacena información de control del proceso captura, como por ejemplo las tablas que son orígenes de replicación.

  - Sistema de transporte: Los sistemas de transportes más comunes son a través de tablas de paso (staging), que dan lugar a la denominada replicación de tipo SQL, o a través de un middleware de gestión de colas, la denominada queue-replication o Q-replication.

  - Programa de aplicación de cambios: es la pieza que, o bien leen mediante SQL de las tablas de staging los cambios de colas en la Q-replication, y mediante la información de control almacenada en tablas realiza el mapeo entre datos de origen y de destino, realiza las transformaciones necesarias a los datos y actualiza los datos de destino mediante SQL si se trata de destinos relacionales, o publica un registro XML para que pueda ser tratado por aplicaciones de propósito general.

- Programa de administración: permite las definiciones necesarias de origen de datos y destinos, mapeos, transformaciones y establecer los intervalos de aplicación de cambios. Usualmente es una herramienta de tipo grafico.

- Utilidades: programas de utilidad que sirven para, por ejemplo, planificar una carga de datos inicial del destino a partir de los datos de origen.

[Fuente: (Díaz, Introducción al Bussines Intelligence, 2010)]

5.1.3 Uso de la integración de datos

Los procesos de integración de datos se usan en múltiples tipologías de proyectos. Se puede destacar los siguientes:

- Migración de datos.
- Procesos de calidad de datos.
- Corporate Performance Management (CPM).
- Master Data Management (MDM).
- Customer Data Integration (CDI).
- Product Information Management (PIM).
- Enterprise Information Management (EIM).
- Data Warehousing.
- Business Intelligence (BI).

5.2 Metodología para ETL usando Pentaho: INTEGRACIÓN

"Cualquier poder, si no se basa en la unión, es débil."
Jean de la Fontaine (1621-1695) Escritor y poeta francés

PDI es una solución de integración de datos programada en java orientada completamente al usuario y basada en un enfoque de metadatos. Los procesos ETL se encapsulan en metadatos que se ejecutan a través del motor ETL.

Pentaho Data Integration (PDI), anteriormente llamado Kettle[15], fue iniciado en 2001 por Matt Casters. En el año 2006, Pentaho adquirió Kettle y lo renombró después de que éste pasara a ser open source. De esta forma continuaba con la política de crear una suite completa de inteligencia de negocio open source. Matt Casters pasó a formar parte del equipo de Pentaho.

[Fuente: **(Díaz, Introducción al Bussines Intelligence, 2010)**]

[15] Por sus siglas en ingles "K Extraction Transformation Transportation Load E"

Figura 11. Logo Kettle Pentaho data integration [Fuente: **(Pentaho Open Source BI, 2011)**]

Esta herramienta permite cargar datos de múltiples fuentes de origen, cargar dichos datos en un data warehouse para que posteriormente la información consolidada sea de utilidad a nivel operativo, táctico y estratégico.

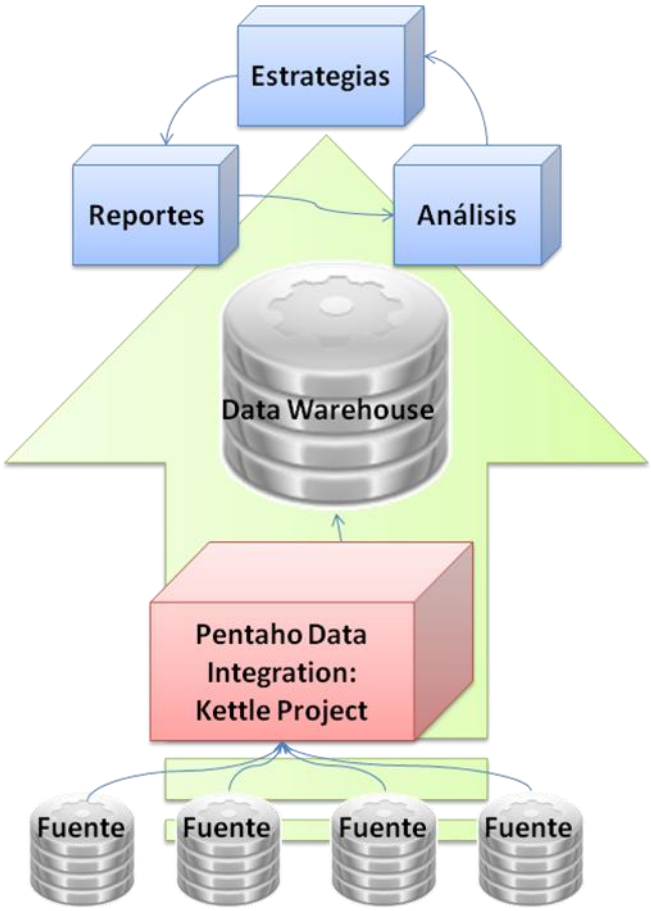


Figura 12. Orientación de la ETL de Kettle [Fuente: Diseño propio basado en imagen de **(Cutro, 2009)**]

Las principales características de PDI son:

- Entorno gráfico orientado al desarrollo rápido y ágil basado en dos áreas: la de trabajo y la de diseño/vista.
- Multiplataforma
- Incluye múltiples conectores a bases de datos, tanto propietarias como comerciales. Así como conectores a archivos planos, Excel, XML u otros.
- Arquitectura extensible mediante plugins

- Soporta uso de cluster, procesos ETL en paralelo y arquitecturas servido Maestro/Esclavo
- Completamente integrado con la suite de Pentaho
- Basado en el desarrollo de dos tipos de objetos:
  - Transformaciones: permiten definir las operaciones de transformación de datos.
  - Trabajos: permiten gestionar y administrar procesos ETL a alto nivel

En la siguiente imagen se presenta el entorno gráfico de ésta herramienta de kettle denominada Spoon, en la que se construyen los trabajos y transformaciones. En la gráfica se aprecia la división del área de trabajo y área de vista/diseño:

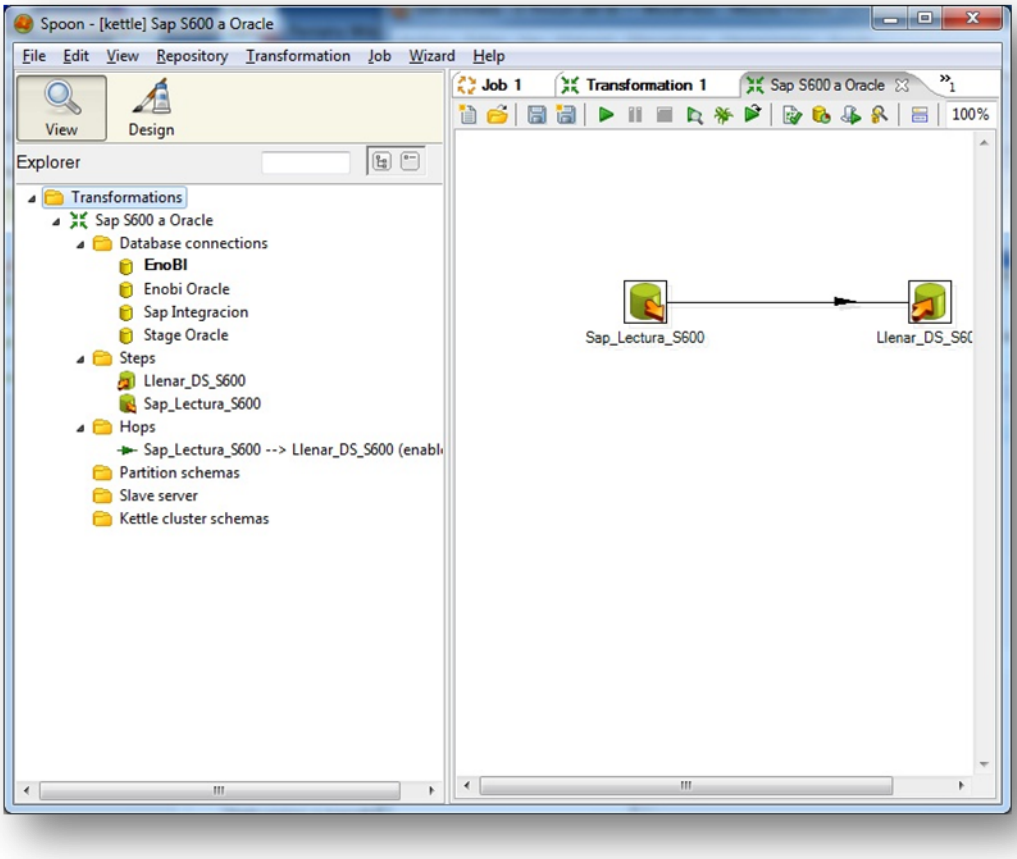


Figura 13. Entorno gráfico Kettle [Fuente: Tomado de Software Kettle]

El siguiente diagrama proporciona una idea de cómo se relacionan trabajos y transformaciones.

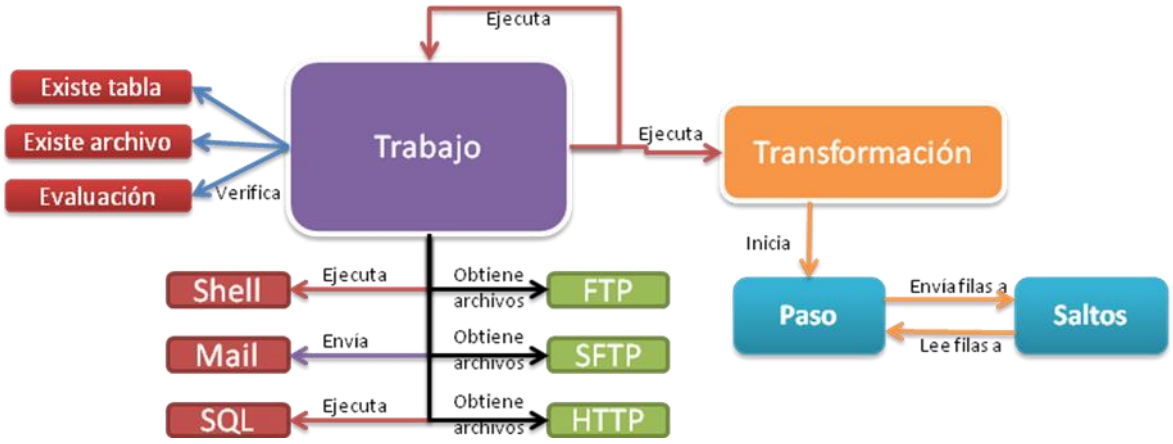


Figura 14. Relación trabajos y transformaciones Kettle PDI [Fuente: Diseño propio basado en información de **(Pentaho Open Source BI, 2011)**]

- Está formado por cuatro componentes:
  - Spoon: entorno gráfico para el desarrollo de transformaciones y trabajos.
  - Pan: permite ejecutar transformaciones
  - Kitchen: permite ejecutar trabajos
  - Carte: es un servidor remoto que permite la ejecución de transformaciones y trabajos.
- Pasos disponibles para trabajos:
  - Generales: permite iniciar un trabajo, ejecutar transformaciones o trabajos entre otras operaciones.
  - Correo: permite enviar correos, recuperar cuentas o validarlas.
  - Gestión de archivos: permite realizar operaciones con archivos como crear, borrar, comparar o comprimir.
  - Condiciones: permite realizar comprobaciones necesarias para procesos ETL como la existencia de un archivo, una carpeta o una tabla.
  - Scripting: permite crear scripts de Java Script, SQL y Shell.
  - Carga bulk: permite realizar cargas bulk a MySQL, MSSQL, Acces y archivos.
  - XML: permite validar XML y XSD.
  - Envió de archivos: permite enviar o coger archivos desde FTP y SFTP.
  - Repositorio: permite realizar operaciones con el repositorio de transformaciones y trabajos.
- Pasos disponibles para transformaciones
  - -Entrada: permite recuperar datos desde base de datos (JDBC), Acces, CSV, Excel archivos, LDAP, Mondrian, RSS u otros.
  - -Salida: permite cargar datos en bases de datos u otros formatos de salida.
  - -Transformar: permite realizar operaciones con datos como filtrar, ordenar, partir añadir nuevos campos, mapear…
  - -Utilidades: permite operar con filas o columnas y otras operaciones como enviar un email, escribir a un log.

-Flujo: permite realizar operaciones con el flujo de datos como fusionar, detectar flujos vacios, realizar operaciones en función de una condición…
-Scripting: permiten crear scripts de JavaScript, SQL, expresiones regulares, formulas y expresiones java.
-Búsqueda de datos: permite añadir información al flujo de datos mediante la búsqueda en base de datos y otras fuentes.
-Uniones: permite unir filas en función de diferentes criterios.
-Almacén de datos: permite trabajar con dimensiones SCD.
-Validación: permite validar tarjetas de crédito, datos, direcciones de correo o XSD.
-Estadística: permite realizar operaciones estadísticas sobre un flujo de datos.
-Trabajos: permite realizar operaciones propias de trabajo.
-Mapeado: permite realizar el mapeo entre campos de entrada y salida.
-Embebido: permite realizar operaciones con sockets.
-Experimental: incluye los pasos en fase de validación.
-Obsoleto: incluye los pasos que desaparecerán en la siguiente versión del producto.
-Carga bulk: permite realizar cargas bulk a Infobright, LucidDB, MonetDB y Oracle.
-Historial: recopila los pasos frecuentemente usados por el desarrollador.

Diferencias con la versión Enterprise:

- Inclusión de una cónsula web para la administración y monitorización de procesos ETL.
- Soporte profesional

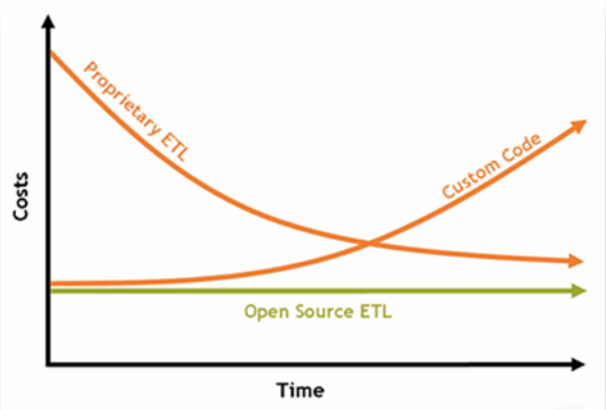


Figura 15. Costo/tiempo ETL propietario VS Open source [Fuente: Wikipedia 2009]

5.2.1 Caso Práctico

Con el objetivo de entender cómo se diseñan los procesos ETL, en el caso práctico se partirá de una situación real simplificada.

Sí se considera que se administra una única aplicación transaccional que está alojada en un motor de base de datos Microsoft Access. Esta aplicación contabiliza las ventas realizadas en la oficina de despachos principal a todos los clientes.
Las tablas de la base de datos suministra la siguiente información:

- Ciudad y País del Cliente que realizó la compra
- Tipo de negocio del Cliente (distribuidor o directo)
- ID del Cliente
- Genero del producto comprado por el cliente (Hombre/Mujer)
- Línea de Producto comprado por el cliente (Casual/Formal)
- Mes y Año en que se realizó la venta
- Valor de la venta realizada

También se considerará que, conociendo las necesidades informacionales, se ha preparado un archivo en texto plano que contiene el nombre de los clientes, exportado desde otro sistema de información; el cuál se tiene para facilitar la carga de algunas de las dimensiones de la bodega de datos.

Resumiendo, la situación de partida está formada por un archivo de texto plano (Clientes.csv) y una base de datos de Microsoft Access, con los que se procederá a una carga inicial de la bodega de datos.

La estrategia que se seguirá en el proceso ETL será:

1. Cargar las dimensiones localización, tiempo, línea, canal, género y la tabla de hecho ventas a partir de la base de datos de MS Access.
2. Complementar la tabla de hecho ventas a partir el archivo CantidadVentas.csv y la base de datos MS Access.
3. Crear un trabajo para lanzar todas las transformaciones de una manera única.

Se usara la siguiente notación:

- Para las transformaciones: TRA_ETL_INI_(nombre de la dimensión o tabla de hecho a cargar)
- Para los trabajos: JOB_CARGA_INI_(nombre de la dimensión o tabla de hecho a cargar).

Para el almacenamiento se puede trabajar por archivos individuales o almacenarlos en una carpeta del proyecto que se está realizando. Éstos archivos hacen referencia a los distintos trabajos o transformaciones generadas en el proceso de ETL. Se recomienda asignar un nombre descriptivo para todo este tipo de archivos.

En el caso de Pentaho, se puede trabajar de dos maneras: mediante el repositorio de datos (que permite guardar las transformaciones en base de datos) o mediante archivos. Se trabajará de esta segunda manera para compartir de manera más fácil los cambios.

En primer lugar, para todas las transformaciones y trabajos que se diseñen en PDI se definirá la conexión a la base de datos. En este caso práctico son dos conexiones existentes: Microsoft Access y Oracle.

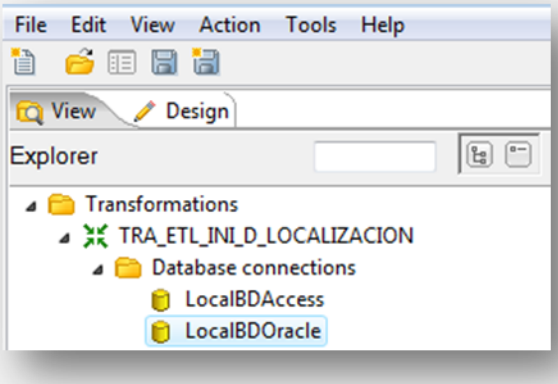


Figura 16. Conexiones a bases de datos [Fuente: Elaboración propia]

En la bodega de datos debe existir la estructura de tablas diseñada en el capítulo 3.

Siguiendo la estrategia que se ha definido anteriormente, el primer paso es la carga de las tablas de dimensiones y tablas de hechos. A continuación se explica la transformación de una de las dimensiones, las siguientes son la misma estructura.

La transformación ETL llamada TRA_ETL_INI_D_LOCALIZACION tiene dos pasos:

Figura 17. PDI transformación dimensión localización [Fuente: Elaboración propia]

1. Lectura Tablas Localizacion, que contiene la sentencia SQL que obtiene las distintas ciudades y países de la base de datos de Access.

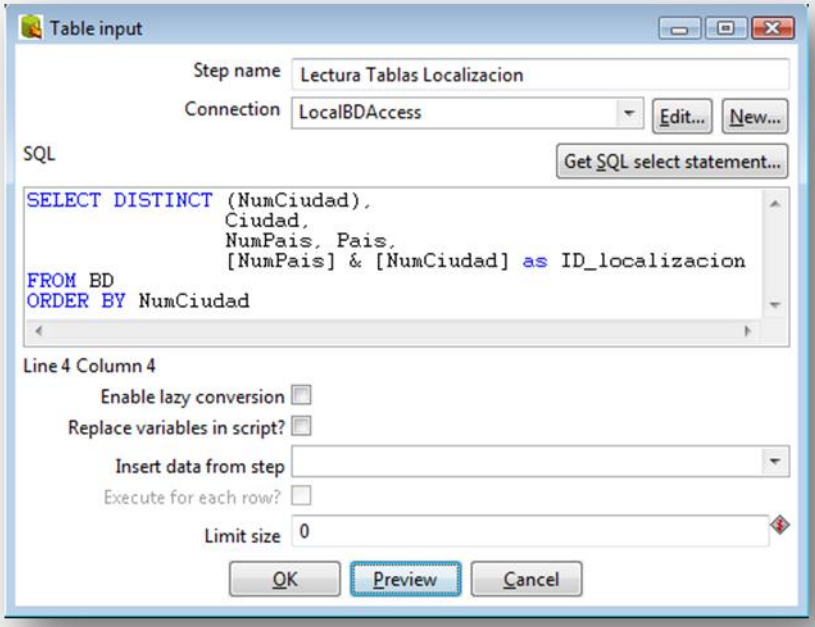


Figura 18. PDI Paso Lectura Tablas localización [Fuente: Elaboración propia]

2. Insertar/Actualizar D_Localizacion, la cual alimenta la bodega de datos.

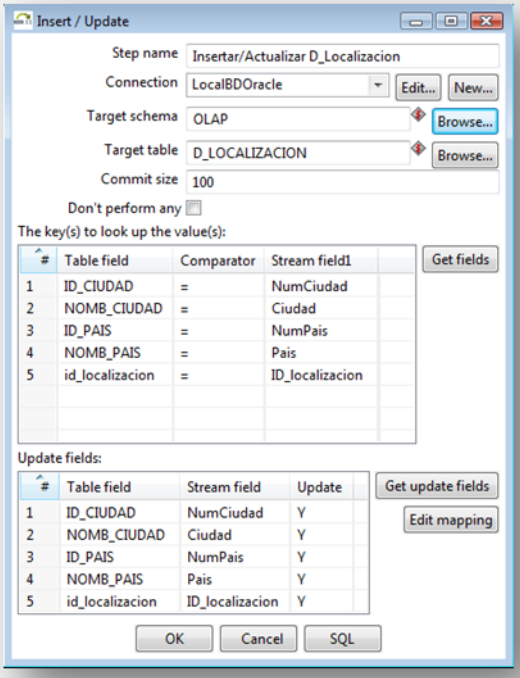


Figura 19. PDI paso Insertar/Actualizar dimensión localización [Fuente: Elaboración propia]

Se crean las transformaciones para cada una de las dimensiones diseñadas: tiempo, línea, canal, género y la tabla de hecho ventas:
- TRA_ETL_INI_D_CANAL
- TRA_ETL_INI_D_GENERO
- TRA_ETL_INI_D_LINEA
- TRA_ETL_INI_D_TIEMPO
- TRA_ETL_INI_H_VENTAS

El siguiente paso de la estrategia es alimentar la Tabla de hecho ventas, con la cantidad vendida que se tiene almacenada en un archivo de texto CantidadVentas.csv (exportado desde otro sistema). Para este caso se crean los siguientes pasos a la Transformación de la tabla de hecho ventas:

1. Se crea la transformación encargada de leer la información del archivo CSV, identificando las columnas del mismo:

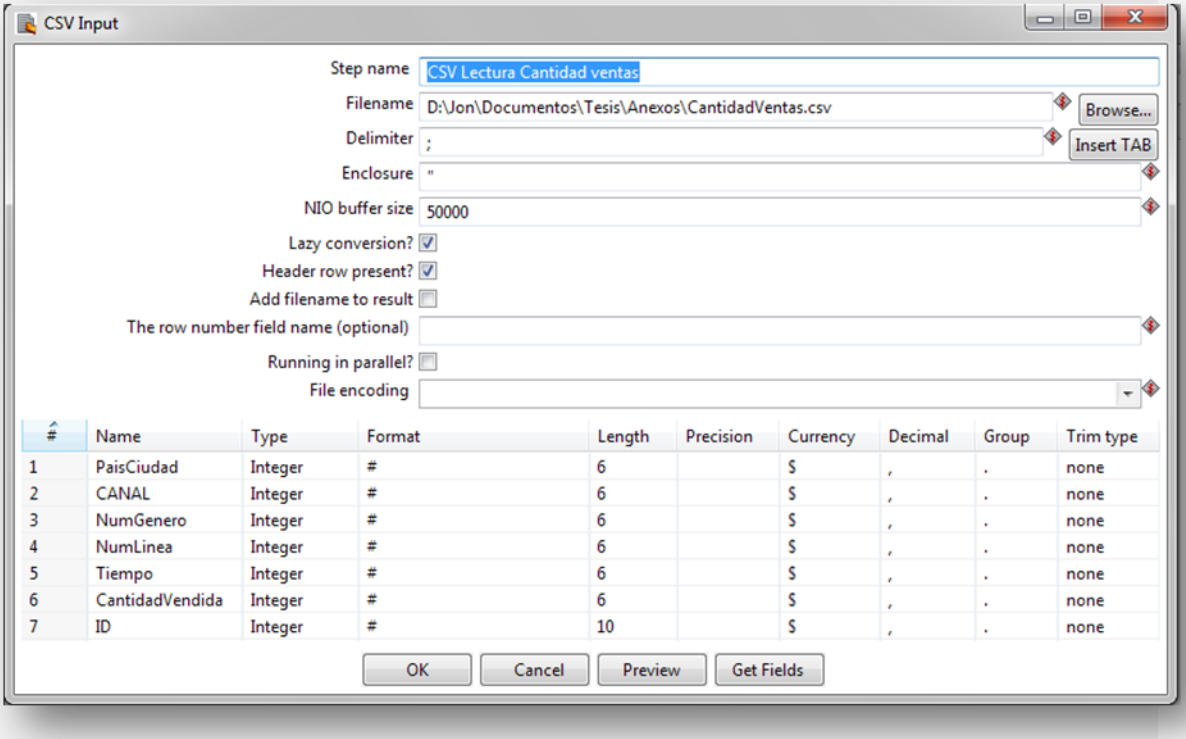


Figura 20. PDI paso lectura cantidad ventas [Fuente: Elaboración propia]

2. Se actualizan los registros de la tabla H_Ventas basando en la búsqueda de las llaves fóraneas (Las claves de las dimensiones) y se agrega el valor de la cantidad de

ventas realizadas, como se ve en la siguiente transformación *Insertar/Actualizar H_Ventas 2*

En ésta transformación se usa la propiedad para buscar y comparar las distintas columnas obtenidas del archivo CSV y las columnas ya ingresadas por la transformación anterior. Esto con el objetivo de no obtener registros repetidos.

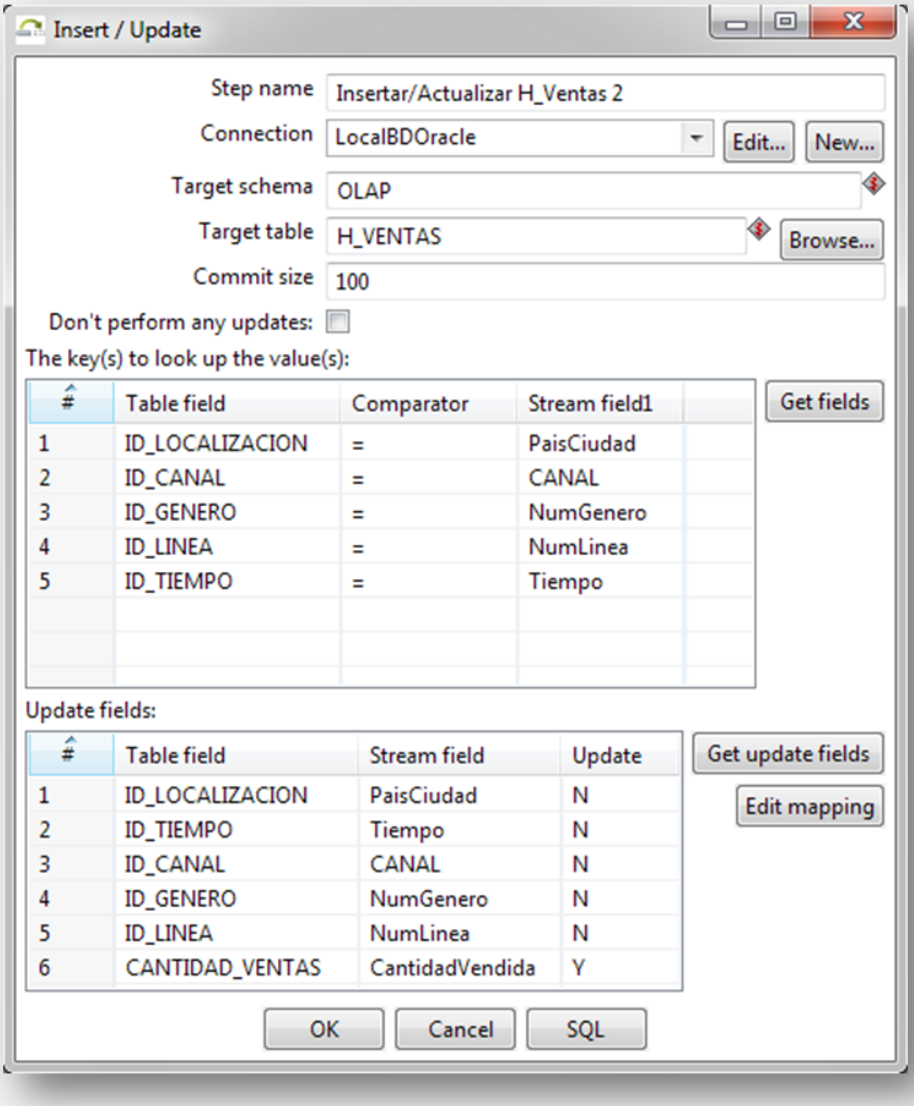


Figura 21. PDI paso actualziación cantidad de ventas en Tabla de hecho [Fuente: Elaboración propia]

Como resultado de ésta fase de la metodología, se debe asegurar el cargue y calidad de los datos. Se concluye este capítulo con la bodega de datos respectivamente poblada con los datos de las distintas fuentes de información.

6. DISEÑO DE CUBOS OLAP

OLAP es el acrónimo en inglés de procesamiento analítico en línea (*On-Line Analytical Processing*), Es una solución utilizada en el campo de la llamada Inteligencia de Negocios[16] cuyo objetivo es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP)[17].

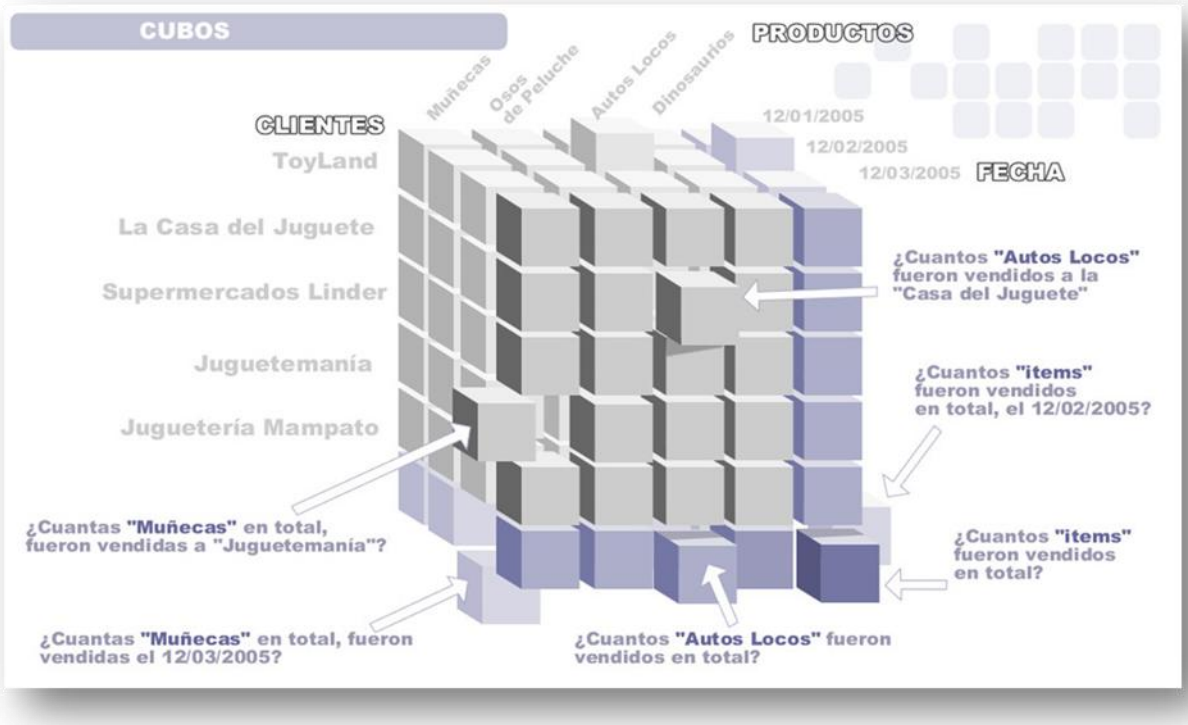


Figura 22. Cubo OLAP de tres dimensiones [Fuente: **(Yahazee, 2009)**]

OLAP es uno de los conceptos más importantes de inteligencia de negocio. Si bien el término OLAP se introduce por primera vez en 1993, los conceptos base del mismo, como por ejemplo el análisis multidimensional[18], son mucho más antiguos.

Es necesario, antes de continuar, introducir una definición formal de OLAP:

---

[16] Ó Business Intelligence (BI)

[17] **O**n**L**ine **T**ransaction **P**rocessing (Procesamiento de Transacciones En Línea)

[18] En 1962, se introduce el análisis multidimensional en el libro de Ken Iverson A Programming Language

Se entiende por OLAP, o proceso analítico en línea, al método ágil y flexible para organizar datos, especialmente metadatos, sobre un objeto o jerarquía de objetos como en un sistema u organización multidimensional, y cuyo objetivo es recuperar y manipular datos y combinaciones de los mismos a través de consultas o incluso informes. [Fuente: **(Wrembel, 2006)**]

A pesar de ser una tecnología que ya tiene más de cuatro décadas, sus características y su evolución han provocado que la gran mayoría de soluciones de soluciones del mercado incluya un motor OLAP.

Es necesario precisar:

- Las herramientas OLAP de los diferentes fabricantes, si bien son similares, no son completamente iguales dado que presentan diferentes especificaciones del modelo teórico.
- Las soluciones open source OLAP han sido las últimas a añadirse a la lista y, por ahora, no tienen tanta variedad como su equivalente propietaria.
- En el mercado Open Source OLAP sólo hay dos soluciones actualmente: el motor ROLAP Mondrian y el motor MOLAP PALO.

A diferencia del conocido OLTP, OLAP describe una clase de tecnologías diseñadas para mantener específicamente el análisis y acceso a datos. Mientras el procesamiento transaccional generalmente confía solamente en las bases de datos relacionales, OLAP viene a ser un sinónimo con vistas multidimensionales de los datos del negocio. Estas vistas multidimensionales se apoyan en la tecnología de bases de datos multidimensionales.
OLAP se está convirtiendo rápidamente en la base fundamental para Soluciones Inteligentes incluyendo Business Performance Management, Planificación, presupuestos, previsiones, informes financieros, análisis, modelos de simulación, Descubrimiento de Conocimiento, e informes de Bodegas de datos.

¿Por qué OLAP?
Para comprender las ventajas de la tecnología OLAP es necesario, primero, hacer una comparación con el procesamiento transaccional en línea (OLTP), de tal forma que se pueda valorar el alcance de esta tecnología de información.

OLTP (Relacional)	OLAP(Multidimensional)
Automatizado	Resumido
Presente	Historico
Un registro por tiempo	Muchos registros al tiempo
Orientado a Proceso	Orientado al tema

Tabla 4. Comparación OLTP vs OLAP [Fuente: **(Universidad Peruana de Ciencias Aplicadas, 2008)**

La razón de usar OLAP para las consultas es la velocidad de respuesta. Una base de datos relacional almacena entidades en tablas discretas si han sido normalizadas. Esta estructura es buena en un sistema OLTP pero para las complejas consultas multitabla es relativamente lenta. Un modelo mejor para búsquedas (aunque peor desde el punto de vista operativo) es una base de datos multidimensional.

Las aplicaciones OLTP se caracterizan por la creación de muchos usuarios, actualizaciones o recuperación de registros individuales. Por consiguiente, las bases de datos OLTP se perfeccionan para actualización de transacciones. Las aplicaciones OLAP son usadas por analistas y gerentes que frecuentemente quieren una vista de datos de nivel superior, como las ventas totales por línea de producto, por región, etc. Las bases de datos OLAP normalmente se actualizan en lote, a menudo de múltiples fuentes, y proporcionan un back-end[19] analítico poderoso a las aplicaciones de múltiples usuarios. Por tanto, las bases de datos OLAP se perfeccionan para el análisis.

Mientras las bases de datos relacionales son buenas al recuperar un número pequeño de archivos rápidamente, ellas no son buenas al recuperar un número grande de archivos y resumirlos sobre la marcha. Un tiempo de respuesta lento y el uso excesivo de recursos del sistema son las características comunes de las aplicaciones de soporte de decisión construidas exclusivamente sobre la tecnología de bases de datos relacionales. Debido a la facilidad con la cual se puede emitir un "ejecutar una consulta SQL externa", muchos distribuidores de Sistemas de Información no brindan acceso directo a los usuarios a sus bases de datos relacionales.

Muchos de los problemas que las personas intentan resolver con la tecnología relacional son realmente multidimensionales en naturaleza. Por ejemplo, una consulta SQL para crear resúmenes de ventas del producto por la región, las ventas de la región por producto, y así sucesivamente, podrían involucrar la revisión de la mayoría, si no todos, los registros en una base de datos de mercadeo y podría tomar horas de proceso. Un servidor OLAP podría ocuparse de estas preguntas en unos segundos. [Fuente: **(Tindys, 2010)**]

Las aplicaciones OLTP tienden a tratar con datos atomizados "registro a un tiempo", considerando que las aplicaciones de OLAP normalmente se tratan de los datos resumidos. Mientras las aplicaciones OLTP generalmente no requieren de datos históricos, casi cada aplicación de OLAP se preocupa por ver las tendencias y por consiguiente requiere de datos históricos. Como consecuencia, las bases de datos OLAP necesitan la capacidad de ocuparse de datos series de. Mientras las aplicaciones OLTP y bases de datos tienden a ser organizados alrededor de procesos específicos (como ordenes de entrada), las aplicaciones OLAP tienden a ser "orientadas al tema",

[19] Front-end y back-end son términos que se relacionan con el principio y el final de un proceso

respondiendo a preguntas como "¿Qué productos están vendiendo bien?" o "¿Dónde están mis oficinas de ventas más débiles?". [Fuente: **(Jaac316, 2010)**]

6.1 Tipos de OLAP

Existen diferentes tipos de OLAP, que principalmente difieren en cómo se guardan los datos: [Fuente: **(Díaz, Tecnología al instante, 2010)**]

- **MOLAP (Multidimensional OLAP):** es la forma clásica de OLAP y frecuentemente es referida con dicho acrónimo. MOLAP utiliza estructuras de bases de datos generalmente optimizadas para recuperación de los mismos. Es lo que se conoce como bases de datos multidimensionales (o, más coloquialmente, cubos). En definitiva, se crea un archivo que contiene todas las posibles consultas precalculadas. A diferencia de las bases de datos relacionales, estas formas de almacenaje están optimizadas para la velocidad de cálculo. También se optimizan a menudo para la recuperación a lo largo de patrones jerárquicos de acceso. Las dimensiones de cada cubo son típicamente atributos tales como periodo, localización, producto o código de cuenta. La forma en la que cada dimensión será agregada se define por adelantado.

- **ROLAP (Relacional OLAP):** trabaja directamente con las bases de datos relacionales, que almacenan los datos base y las tablas dimensionales como tablas relacionales mientras se crean nuevas tablas para guardar la información agregada.

- **HOLAP (Hibrido OLAP):** no hay acuerdo claro en la industria en cuanto a qué constituye el OLAP híbrido, exceptuando el hecho de que es una base de datos en la que los datos se dividen en almacenaje relacional y multidimensional. Por ejemplo, para algunos vendedores, HOLAP consiste en utilizar las tablas relacionales para guardar cantidades más grandes de datos detallados, y utiliza el almacenaje multidimensional para algunos aspectos de cantidades más pequeñas de datos menos detallados o agregados.

- **DOLAP (Desktop OLAP):** es un caso particular de OLAP ya que está orientado a equipos de escritorio. Consiste en obtener la información necesaria desde la base de datos relacional y guardarla en el escritorio. Las consultas y los análisis son realizados contra los datos guardados en el escritorio.

- **In-memory OLAP:** es un enfoque por el que muchos nuevos fabricantes están optando. Consiste en que la estructura dimensional se genera sólo a nivel de memoria y se guarda el dato original en algún formato que potencia su despliegue de esta forma (por ejemplo, comprimido o mediante una base de datos lógica asociativa). En este último punto es donde cada fabricante pone su énfasis.

Cada tipo tiene ciertas ventajas, aunque hay desacuerdo sobres las ventajas específicas de los diferentes proveedores.

- MOLAP es mejor en sistemas más pequeños de datos, es más rápido para calcular agregaciones y retornar respuestas y necesita menos espacio de almacenaje. Últimamente, in-memory OLAP está apuntándose como una opción válida al MOLAP.

- ROLAP se considera más escalable. Sin embargo, el preproceso de grandes volúmenes es difícil de implementar eficientemente, así que se desecha con frecuencia. De otro modo, el funcionamiento de consultas pude ser no óptimo.

- HOLAP está entre los dos en todas las áreas, pero puede preprocesar rápidamente y escalar bien.

Todos los tipos son, sin embargo, propensos a la explosión de la base de datos. Éste es un fenómeno que causa la cantidad extensa de espacio de almacenaje que es utilizado por las bases de datos OLAP cuando se resuelven ciertas, pero frecuentes, condiciones: alto número de dimensiones, de resultados calculados de antemano y de datos multidimensionales escasos.

La dificultad en la implementación OLAP deviene en la formación de las consultas, elegir los datos base y desarrollar el esquema. Como resultado, la mayoría de los productos modernos vienen con bibliotecas enormes de consultas preconfiguradas. Otro problema está en la baja calidad de los datos, que deben ser completos y constantes.

6.2 Elementos OLAP

OLAP permite el análisis multidimensional. Ello significa que la información está estructurada en ejes (puntos de vista de análisis) y celdas (valores que se están analizando).
En el contexto OLAP existen diferentes elementos comunes a las diferentes topologías OLAP (que en definitiva se diferencian a nivel práctico en que en MOLAP se precalculan los datos, en ROLAP no, y en in-memory se generan al iniciar el sistema):

- *Esquema:* un esquema es una colección de cubos, dimensiones, tablas de hecho y roles.
- *Cubo:* es una colección de dimensiones asociadas a una tabla de hecho. Un cubo virtual permite cruzar la información entre tablas de hecho a partir de sus dimensiones comunes.
- *Tabla de hecho, dimensión y métrica*
- *Jerarquía:* es un conjunto de miembros organizados en niveles. En cuanto a bases de datos, se puede entender como una ordenación de los atributos a una dimensión.
- *Nivel:* es un grupo de miembros en una jerarquía que tienen los mismos atributos y nivel de profundidad en una jerarquía.
- *Miembro:* es un punto de la dimensión de un cubo que pertenece a un determinado nivel de una jerarquía. Las métricas (medidas) en OLAP se consideran un tipo especial de miembro que pertenece a su propio tipo de dimensión. Un miembro puede tener propiedades asociadas.
- *Roles:* permisos asociados a un grupo de usuarios.
- *MDX:* es un acrónimo de Multidimensional eXpressions (aunque también es como Multidimensional Query eXpression). Es el lenguaje de consulta de estructuras OLAP, fue creado en 1997 por Microsoft y, sí bien no es un lenguaje estándar, la gran mayoría de fabricantes de herramientas OLAP lo han adoptado como estándar de hecho.

6.3 Las 12 reglas OLAP de E. F. Codd

La definición de OLAP presentada anteriormente se basa en las 12 leyes que acuñó Edgar F. Codd en 1993. Estas reglas son las que, en mayor o menor medida, intentan cumplir los fabricantes de software: [Tomado de: **(Wikipedia, 2011)**]

1. Vista conceptual multidimensional: se trabaja a partir de métricas de negocio y sus dimensiones.
2. Transparencia: el sistema OLAP debe formar parte de un sistema abierto que soporta fuentes de datos heterogéneas (lo que se llama actualmente arquitectura orientada a servicios).
3. Accesibilidad: se debe presentar el servicio OLAP al usuario con un único esquema lógico de datos (lo que, en definitiva, no sindica que debe presentarse respecto una capa de abstracción directa con el modelo de negocio).
4. Rendimiento de informes consistente: el rendimiento de los informes no debería degradarse cuando el número de dimensiones del modelos se incrementa.
5. Arquitectura cliente/servidor: basado en sistemas modulares y abiertos que permitan la interacción y la colbaración.
6. Dimensionalidad genérica: capacidad de crear todo tipo de dimensiones y con funcionalidades aplicables de una dimensión a otra.
7. Manejo dinámico de matriz de baja densidad: la manipulación de datos en los sistemas OLAP debe poder diferenciar valores vacíos de valores nulos y además poder ignorar las celdas sin datos.
8. Soporte para múltiples usuarios: Acceso simultáneo, seguridad e integridad para múltiples usuarios.
9. Operaciones cruzadas entre dimensiones sin restricciones: todas las dimensiones son creadas igual y las operaciones entre dimensiones no debe restringir las relaciones entre celdas.
10. Manipulación de datos intuitiva: dado que los usuarios a los que destinan este tipo de sistema son frecuentemente analistas y altos ejecutivos, la interacción debe considerarse desde el prisma de la máxima usabilidad de los usuarios.
11. Generación de informes flexible: los usuarios deben ser capaces de manipular los resultados que se ajusten a sus necesidades conformando informes. Además, los cambios en el modelo de datos deben reflejarse automáticamente en esos informes.
12. Niveles de dimensiones y de agregación ilimitados: no deben existir restricciones para construir cubos OLAP con dimensiones y con niveles de agregación ilimitados.

6.4 Metodología para el diseño de Cubos: OLAP en el contexto Mondrian

Para la aplicación de la metodología expuesta en este documento, se usará la tecnología Mondrian; Mondrian es el motor OLAP integrado en Pentaho, se combina con un visor OLAP llamado JPivot y como herramienta de desarrollo se usa el componente de Pentaho Schema Workbench.

En el anexo se encontrará Pentaho se detalla mayor información de la herramienta Workbench. Y en el anexo de otras tecnologías de inteligencia de negocio se puede encontrar qué esta herramienta de Jpivot y Mondrian son las más usadas en open source.

Resumiendo:

- **Motor OLAP:** Mondrian
- **Visor OLAP:** JPivot
- **Herramienta de desarrollo:** Pentaho Schema Workbench

6.4.1 Mondrian

[Fuente: **(Díaz, Introducción al Bussines Intelligence, 2010)**]

Mondrian es un servidor/motor OLAP escrito en Java que está licenciado bajo la Eclipse Public License (EPL). Existe como proyecto desde 2003 y fue adquirido por Pentaho en 2005.

Mondrian se caracteriza por ser un motor ROLAP con caché, lo cual lo sitúa cerca del concepto HOLAP. ROLAP significa que en Mondrian no residen datos (salvo en la caché) sino que éstos están en una base de datos en la que existen las tablas que conforman la información multidimensional con la que el motor trabaja. El lenguaje de consulta es MDX.

Mondrian se encarga de recibir consultas dimensionales a un cubo mediante MDX y de devolver los datos. El cubo es, en este caso, un conjunto de metadatos que definen cómo se ha de mapear la consulta por sentencias SQL al repositorio que contiene realmente los datos.

Esta forma de trabajar tiene ciertas ventajas:

- No se generan cubos/estructuras OLAP estáticas y por lo tanto se ahorra en tiempo de generación y en espacio.
- Se trabaja con datos actualizados siempre al utilizar la información residente en la base de datos.
- Mediante el uso del caché y de tablas agregadas, se pretende simular el mejor rendimiento de los sistemas MOLAP.

Mondrian funciona sobre las bases de datos estándar del mercado: Oracle, DB2, SQL Server, MySQL, PostgreSQL, LuciDB, Teradata y otras; lo que habilita y facilita el desarrollo de negocio.

Los últimos desarrollos de Mondrian se caracterizan por incluir olap4j[20]. Es una iniciativa del mismo desarrollador de Mondrian: Julian Hyde.

En la siguiente imagen se ilustra el funcionamiento del Motor Mondrian:

[20] Olap4j es una API java cuyo objetivo es permitir la creación de aplicaciones OLAP intercambiables entre los diferente motores OLAP del mercado.

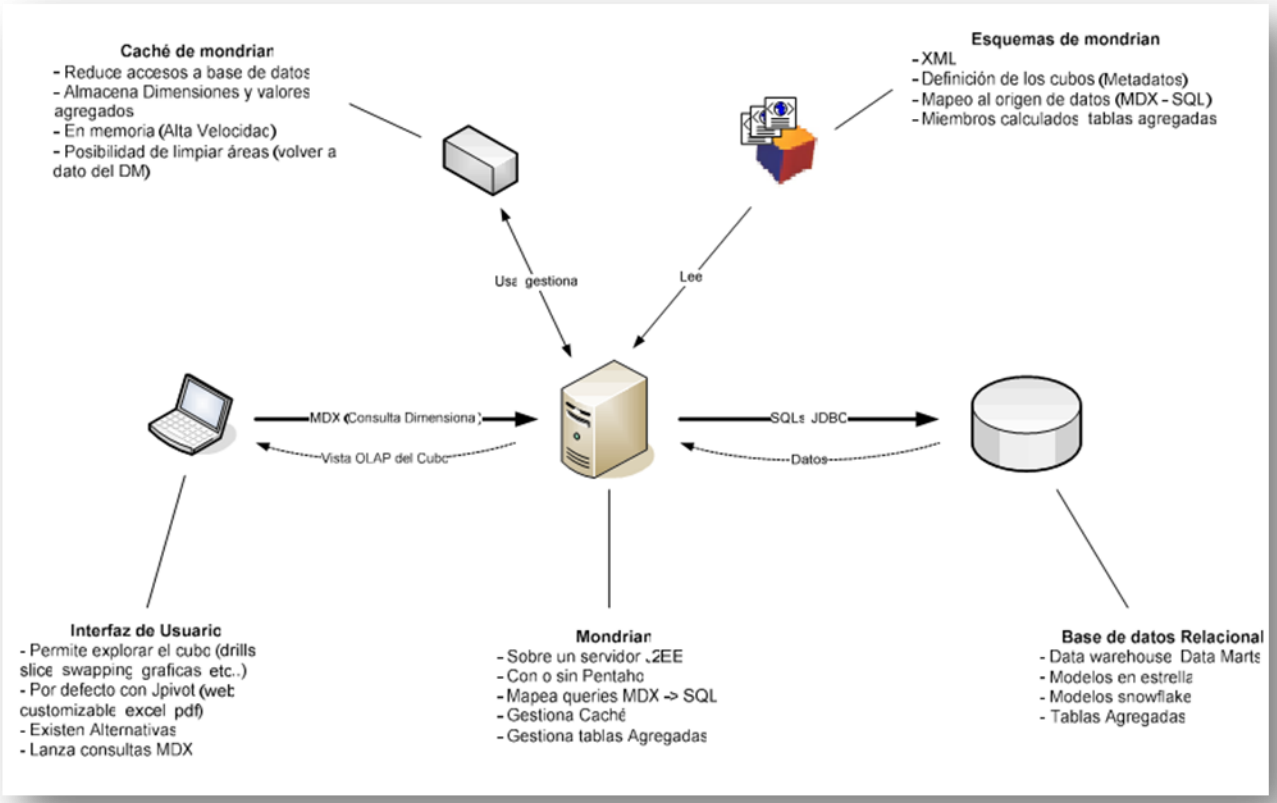


Figura 23. Funcionamiento de mondrian [Fuente: **(Giménez, 2007)**]

Mondrian es un caso atípico en el contexto OSBI. Para la gran mayoría de herramientas de inteligencia de negocio existen una o varias opciones. En el caso de soluciones ROLAP, Mondrian es el único producto.

Varios fabricantes han incluido Mondrian en sus soluciones (JasperSoft, OpenReports, SpagoBI, SQLPower, Suite Visión Empresarial). El hecho de existir una única solución y de existir toda una comunidad de fabricantes y usuarios a su alrededor, hace que el equipo de desarrollo de Mondrian (dirigido por Julian Hyde) tenga ciclos de desarrollo mucho menores que otras soluciones.

6.4.2 Herramienta de Desarrollo Pentaho Schema Workbench

Pentaho Schema Workbench (PSW) es una herramienta de desarrollo que permite crear, modificar y publicar un esquema de Mondrian. Es un programa java multiplataforma.
[Fuente: **(Plauchu, 2011)**]
Es una herramienta orientada al desarrollador conocedor de la estructura de un esquema Mondrian. Permite crear todos los objetos que soporta Mondrian: esquema, cubo, dimensiones, métricas.

Tiene dos aéreas: la zona en la que se muestra la estructura jerárquica del esquema OLAP y la zona de edición de las propiedades de cada elemento.
Como se observa en la figura 24, Workbench presenta un menú superior para crear cubos, dimensiones, dimensiones conformadas, métricas, miembros calculados, subconjuntos (named set) y roles, así como operaciones estándar como cortar, copiar y pegar.

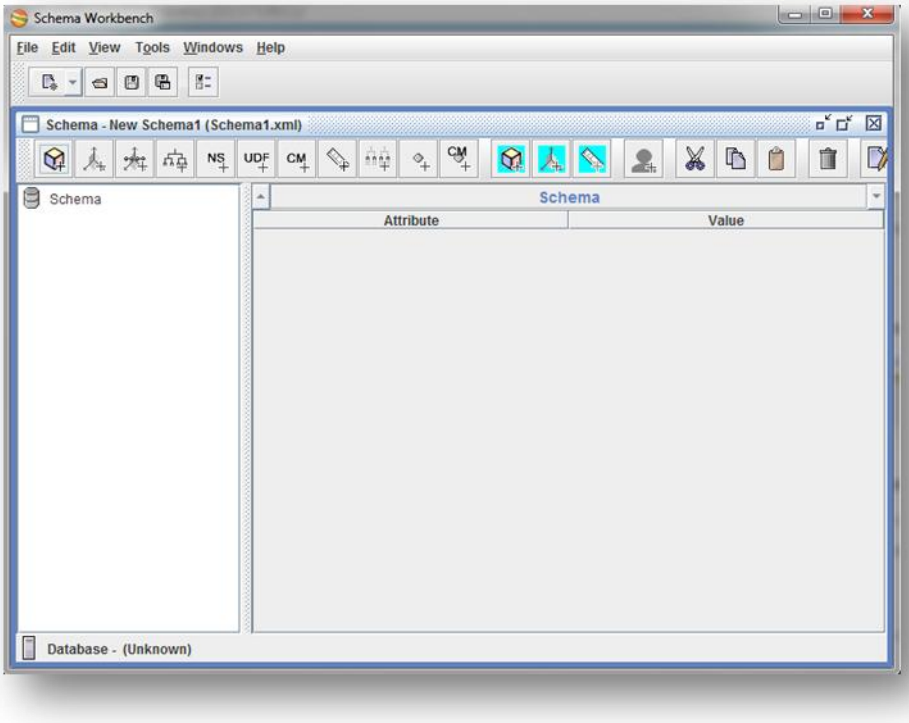


Figura 24. Pentaho Scheme workbench [Fuente: Elaboración propia]

Además, entre sus características incluye:

- Realizar consultas MDX contra el esquema creado (requiere conocer la sintáxis del lenguaje).
- Consultar la base de datos que sirven de origen para el esquema de Mondrian.
- Publicar directamente el esquema en el servidor Pentaho.

6.4.3 Caso Práctico

Con el objetivo de entender y aplicar las tecnologías descritas en la metodología propuesta, y dando continuidad con el caso práctico que se ha venido desarrollando en el presente libro, a continuación se detalla el diseño de un cubo OLAP.

6.4.3.1 Diseño de OLAP con Schema Workbench

El diseño de estructuras OLAP es común en Pentaho y otras soluciones open source del mercado dado que las herramientas que proporciona sólo difieren en pequeños puntos de

rediseño de la interfaz GUI. El punto realmente diferente es cómo se publican en una u otra plataforma.

En el proceso de creación de una estructura OLAP se debe tener presente que lo que se hace es mapear el diseño de la base de datos (tablas de hecho y dimensiones) con el diseño, de forma que:

- Es posible crear un esquema con menos elementos que los existentes en la base de datos (no interesa contemplar todos los puntos de vista de análisis, por ejemplo).
- Es posible crear un esquema con la misma cantidad de elementos. Se consideran todas las tablas de hecho y las dimensiones.
- Es posible crear un esquema con más elementos que los existentes en la base de datos. Por ejemplo, es posible crear dimensiones u otros objetos que sólo existen en el esquema OLAP y que se generan en memoria.

Para este primer ejemplo, se va a considerar un mapeo uno a uno (todos los elementos de la bodega de datos tendrán su correspondencia en la estructura OLAP).
Esta herramienta permite crear elementos o bien a través del despliegue de los elementos disponibles en cada elemento de la arquitectura o bien a través del menú superior que incluye la creación de cubos, dimensiones, jerarquías, niveles, medidas, medidas calculadas, elementos virtuales (cubos dimensiones y métricas), roles y operaciones estándar como copiar, cortar, pegar; e incluso la edición del XML de forma directa.

Figura 25. Menú Scheme workbench [Fuente: Elaboración propia]

Por otra parte, esta herramienta incluye un explorador de la base de datos que, una vez creada la conexión a la base de datos, permite explorar la estructura de las tablas para recordar cuál es el nombre de los campos y atributos a usar.

Los pasos en el proceso de creación son los siguientes:

- Creación de una conexión al data warehouse. La herramienta de diseño necesita conocer cuál es la fuente de origen de tablas y datos. Por ellos, antes de empezar cualquier diseño es necesario dicha conexión. Una vez creada se guarda en memoria y queda grabada para futuras sesiones. En este caso particular los parámetros de conexión son:
  - *Connection Type:* Oracle
  - *Access:* Native JDBC
  - *Host Name:* localhost
  - *Database name:* XE

- *Tablespace for Data:* users
- *Port Number:* 1521
- *User name:* DemoOlap
- *Password:* DemoOlap

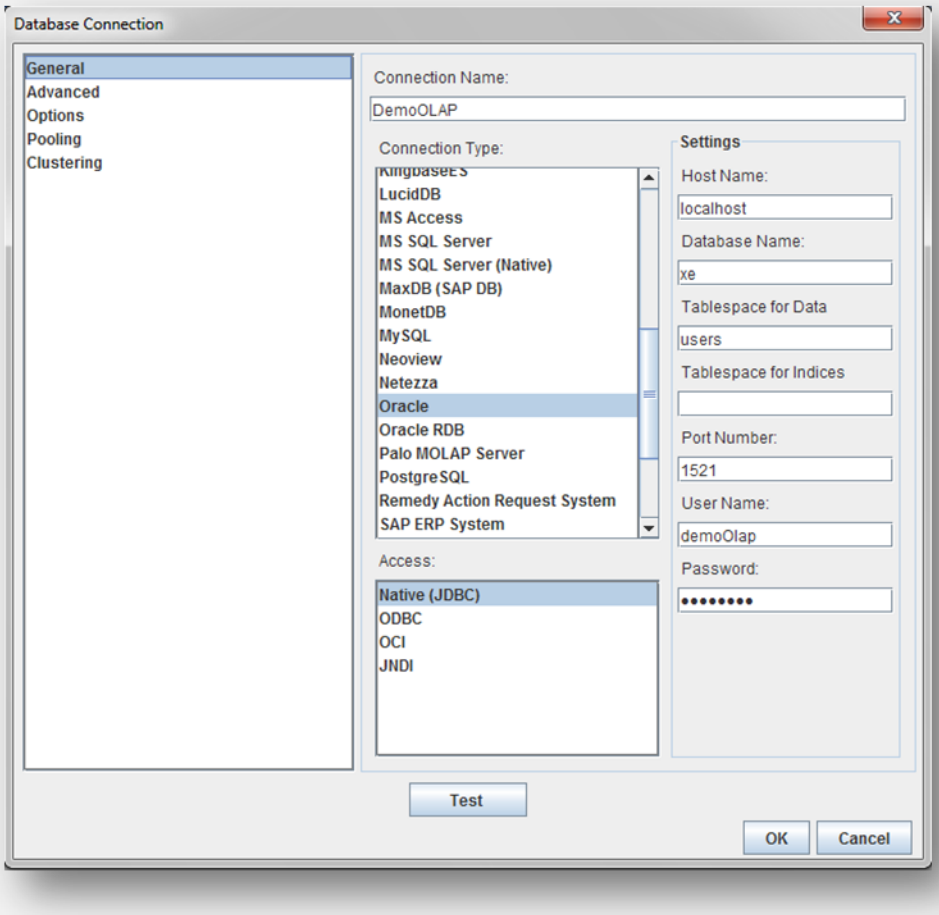


Figura 26. Conexión data warehouse [Fuente: Elaboración propia]

Es necesario recordar que en caso de que la base de datos fuera diferente, estos parámetros serían diferentes. En tal caso también sería necesario comprobar la inclusión del plugin jdbc en la herramienta. (En este caso se está usando el plugin de Oracle *ojdbc14for10.jar*)

- Una vez creada la conexión se puede crear este primer esquema. Los pasos son: crear un esquema, uno o varios cubos, una o varias tablas de hecho, una o varias dimensiones y una o varias métricas.

Para entender el proceso se analiza un esquema creado completamente.

- Primero se completa el esquema introduciendo el nombre del esquema. En este caso, por ejemplo ventas. En caso de que se haya creado un esquema que

contiene diversos cubos, la recomendación sería nombrarlo con el nombre del proyecto.

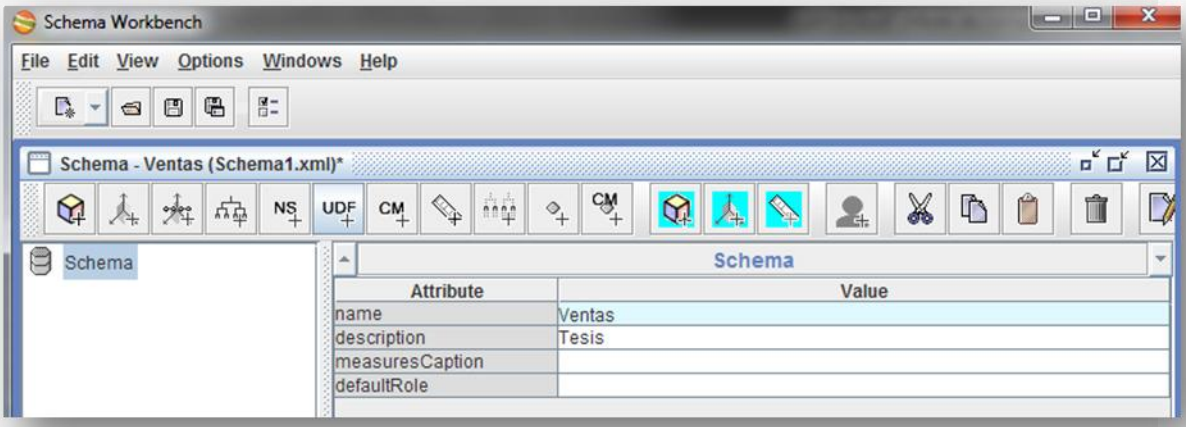


Figura 27. Creación de esquema en Workbench [Fuente: Elaboración propia]

- Se crea un cubo. Se debe definir el nombre y activar las opciones enabled y caché. Esta última opción es importante dado que indica al motor Mondrian que las consultas que se hagan deben guardarse en la caché.

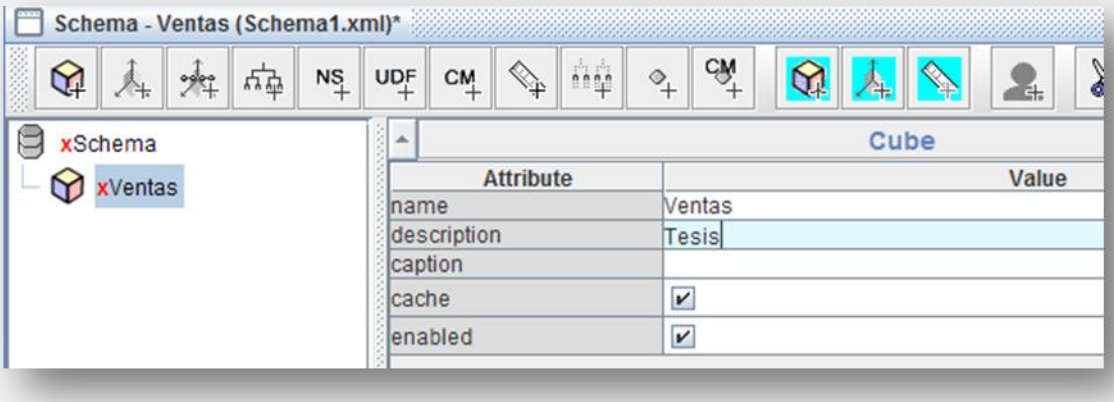


Figura 28. Creación del cubo en Workbench [Fuente: Elaboración propia]

- Todo cubo necesita de una tabla de hecho. En este caso, la de ventas. En el campo *Add Table*, se agrega la tabla que tiene el rol de Tabla de hecho. En este caso *H_Ventas*

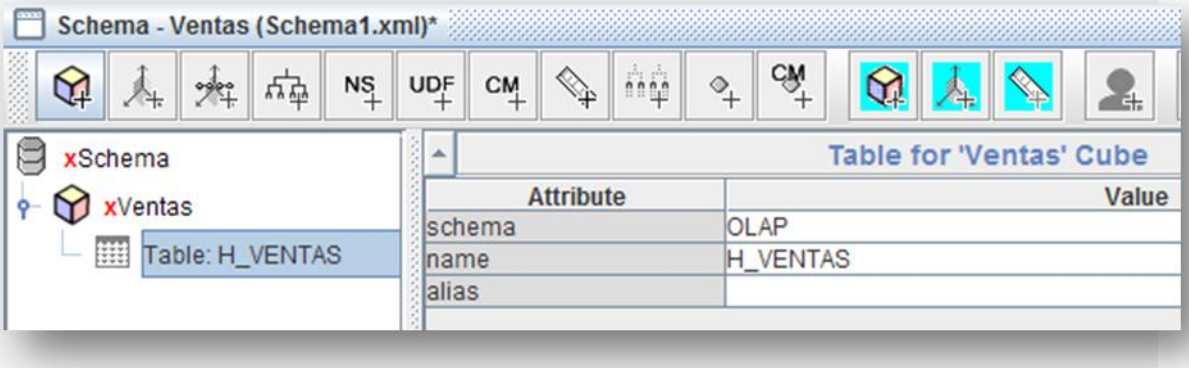


Figura 29. Asociación de tabla h_ventas al Cubo ventas [Fuente: Elaboración propia]

- Un cubo necesita al menos una dimensión. Se analiza el caso de creación de la dimensión Localización. Definimos su nombre y la correspondiente clave foránea.

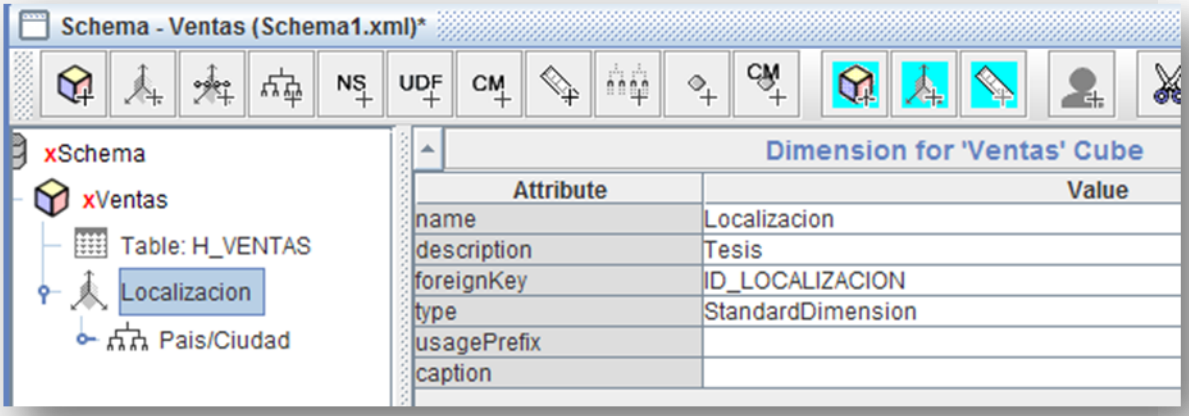


Figura 30. Dimensión localización en workbench [Fuente: Elaboración propia]

- Toda dimensión tiene una o varias jerarquías. Para definir cada jerarquía es necesario definir su nombre, indicar si acumula valores (hasAll), el nombre de dicha acumulación y, finalmente, su clave primaria.

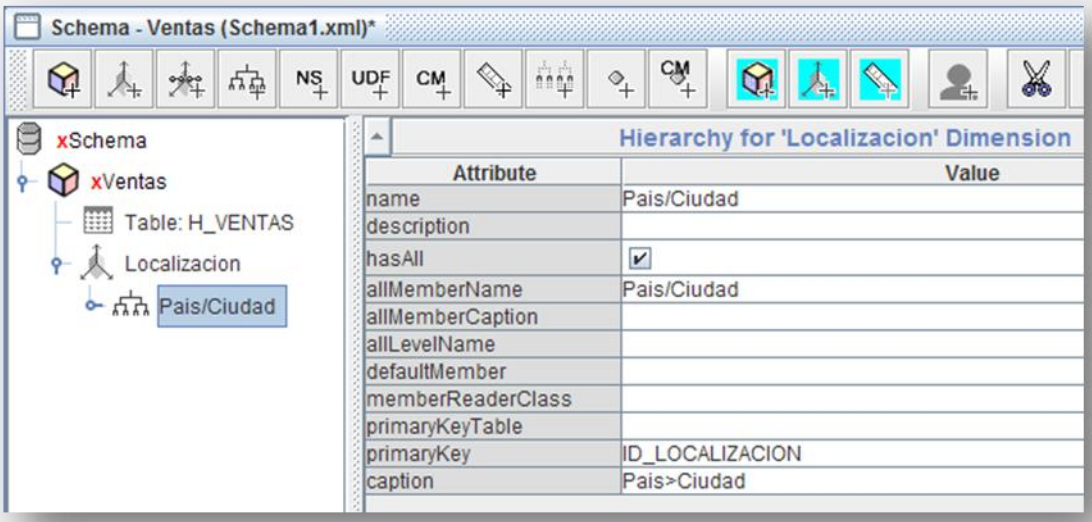


Figura 31. Jerarquía País/Ciudad en workbench [Fuente: Elaboración propia]

- Toda jerarquía necesita como mínimo un nivel que lo componga. Para definir correctamente el nivel se debe especificar el nombre del nivel, la tabla donde está la información, la columna que contiene dicha información, cómo se ordenan los resultados, la tipología del valor, la tipología de nivel, si los valores son únicos y si es necesario ocultar el nivel en algún caso.

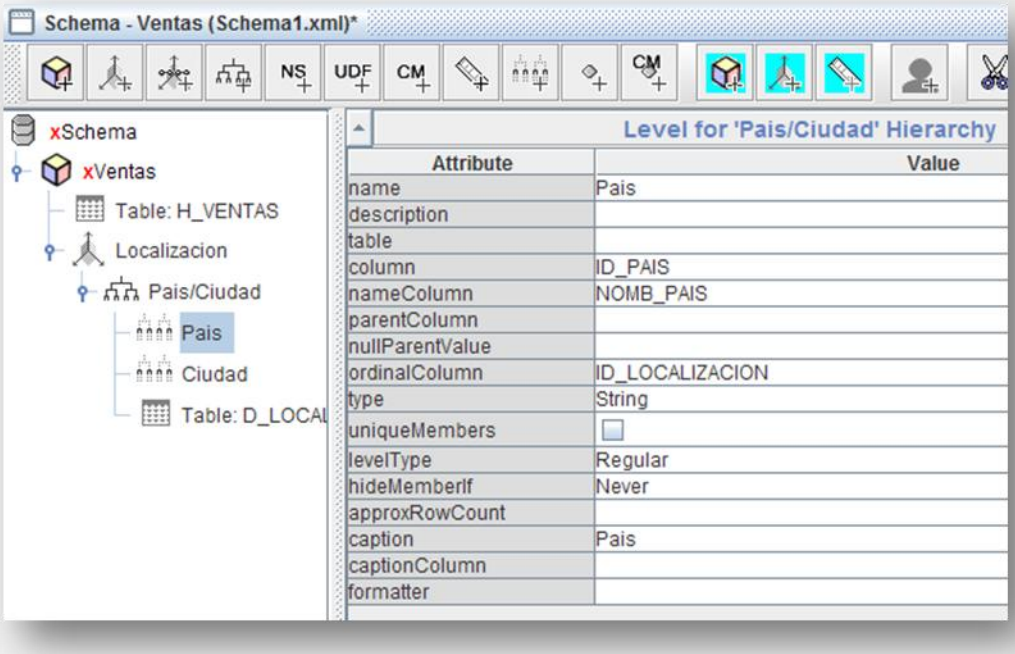


Figura 32. Nivel país en workbench [Fuente: Elaboración propia]

- Se realiza una configuración similar para las otras dimensiones:

-Canal
-Línea
-Género
-Tiempo

- Una vez definidas las dimensiones, es necesario definir las métricas. En la siguiente imagen se muestra la definición del Valor de Ventas:

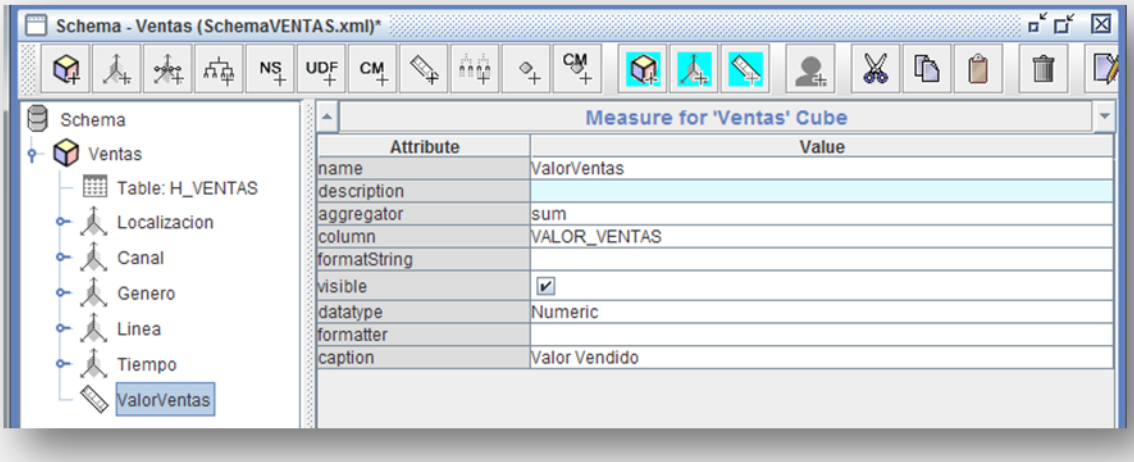


Figura 33. Métrica del valor ventas en Workbench [Fuente: Elaboración propia]

A medida que se van creando el esquema, el sistema WorkBench comprueba que esté bien definido, en caso contrario muestra un mensaje en rojo en la parte inferior.

Se puede comprobar que el diseño que se ha realizado funciona correctamente. Para ello se puede usar el MDX Query. Sí la conexión a la base de datos se ha definido correctamente y el esquema OLAP está bien definido, se conectará con éxito. Para poder comprobar el funcionamiento es necesario escribir una consulta MDX. El lenguaje MDX es similar a SQL pero mucho más complejo. A continuación se usará una consulta sencilla en la que se pone la medida del Valor de Ventas en columnas y la dimensión Género en Filas:

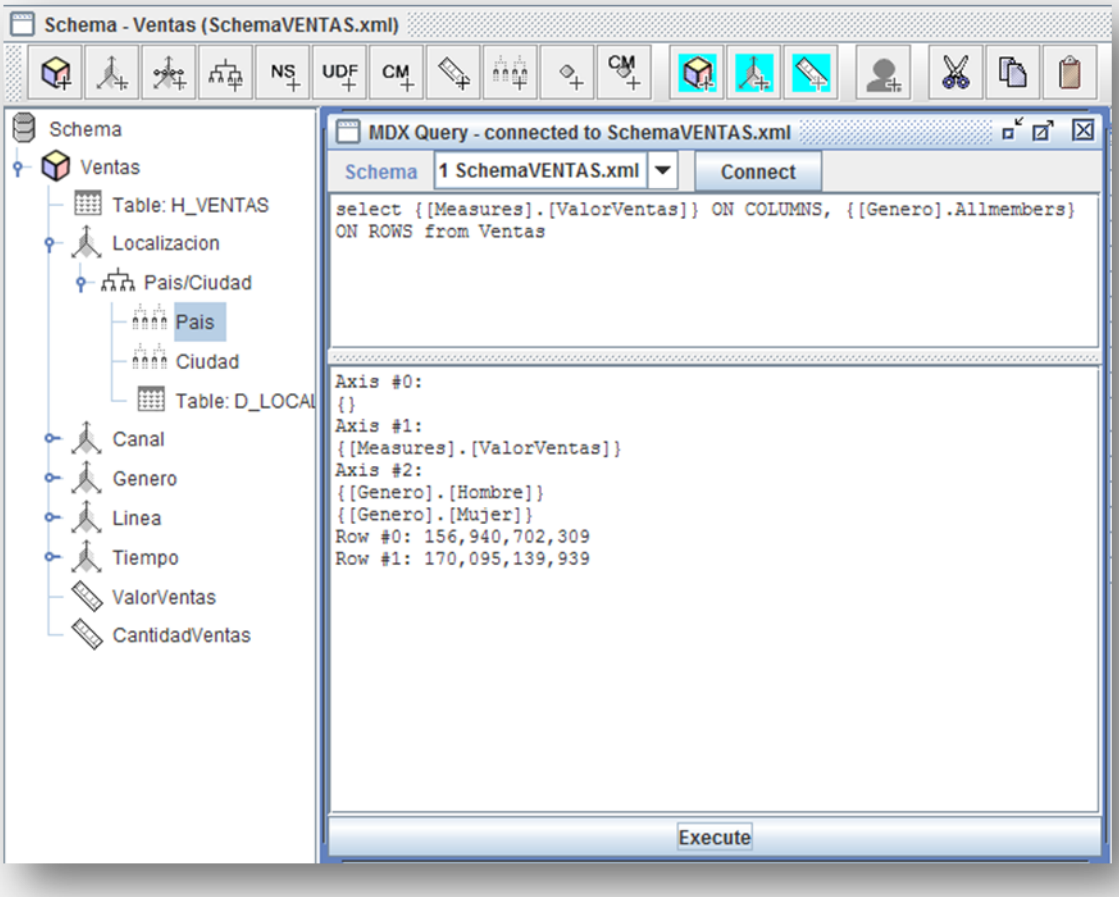


Figura 34. MDX Query en workbench [Fuente: Elaboración propia]

El resultado obtenido es Para las ventas de Ropa de Hombre: $156,940,702,309 y para la ropa de Mujer: $170,095,139,939.

Concluyendo éste capítulo se debe obtener el esquema en un archivo XML, el cual describe la organización de los objetos de la bodega de datos para ser leídos en un cubo OLAP.

7. REPORTES DE CUBOS OLAP

Una característica clave para una herramienta de inteligencia de negocio en el contexto de una organización es la necesidad de reportes operacionales.
A lo largo de la vida de una empresa, la cantidad de datos que se generan por su actividad de negocio crece de forma exponencial. Y esa información se guarda tanto en las bases de datos de las aplicaciones de negocio como en archivos de múltiples formatos.

Es necesario generar y distribuir reportes para conocer el estado del negocio y poder tomar decisiones a todos los niveles: operativo, táctico, y estratégico.

El primer enfoque es modificar las aplicaciones de negocio para que las mismas puedan generar los reportes. Frecuentemente el impacto en las aplicaciones es considerable, y afecta tanto el rendimiento de los reportes como de las operaciones que soporta la aplicación.

Es en ese momento cuando se busca una solución que permita generar reportes sin impactar en el rendimiento de las *aplicaciones de negocio.*

El objetivo de éste capítulo es presentar los elementos de un reporte, criterios de realización y un ejemplo a través de la implementación de un visor de Reportes OLAP usando el software libre JPivot y Mondrian en un contenedor de aplicaciones JSP; para éste caso se usará Apache Tomcat.

7.1 Reportes e Inteligencia de Negocio

Las herramientas de reportes (o también llamadas de reporting) permiten responder principalmente a la pregunta de ¿Qué paso? Dado que esa es la primera pregunta que se formulan los usuarios de negocio, todas las soluciones de Business Intelligence del mercado incluyen un motor de reporting.

Para tener claro, esta es una definición global de un reporte o informe:

> Es un documento a través del cual se presentan los resultados de uno o varios procesos de negocio. Suele contener texto acompañado de elementos como tablas o gráficos para agilizar la comprensión de la información presentada. [Fuente: (Díaz, Introducción al Bussines Intelligence, 2010)]

Los reportes están destinados a usuarios de negocio que tienen la necesidad de conocer la información consolidada y agregada para la toma de decisiones. Ahora se puede definir formalmente las herramientas de reporting:

Se entiende por plataforma de reporting aquellas soluciones que permiten diseñar y gestionar (distribuir, planificar y administrar) reportes en el contexto de una organización o en una de sus áreas. . [Fuente: (Díaz, Introducción al Bussines Intelligence, 2010)]

7.2 Tipos de Reportes

Existen diferentes tipos de reportes en función de la interacción ofrecida al usuario final y la independencia respecto al departamento TI:

- Estáticos: tienen un formato preestablecido inamovible
- Parámetrizados: presentan parámetros de entrada y permiten múltiples consultas.
- Ad-hoc: son creados por el usuario final a partir de la capa de metadatos que permite usar el lenguaje de negocio propio.

7.3 Elementos de un Reporte

Principalmente un reporte puede estar formado por:

- Texto: que describe el estado del proceso de negocio o proporciona las descripciones necesarias para entender el resto de elementos del reporte.
- Tablas: este elemento tiene forma de matriz y permite presentar una gran cantidad de información.
- Gráficos: este elemento persigue el objetivo de mostrar información con un alto impacto visual que sirva para obtener información agregada o sumarizada con mucha más rapidez a través de tablas.
- Mapas: este elemento permite mostrar la información geolocalizada.
- Métricas: que permiten conocer cuantitativamente el estado de un proceso de negocio.
- Alertas visuales y automáticas: consiste en avisos del cambio de estado de información que puede estar formadas por elementos gráficos como fechas o colores resultados y que deben estar automatizadas en función de reglas de negocio encapsuladas en el cuadro de mando. [Fuente: (Díaz, Introducción al Bussines Intelligence, 2010)]

7.4 Tipos de métricas

Los reportes incluyen métricas de negocio. Es por ello necesario definir los diferentes tipos de medidas existentes basadas en el tipo de información que recopilan así como la funcionalidad asociada:

- **Métricas**: valores que recogen el proceso de una actividad o los resultados de la misma. Estas Estas medidas proceden del resultado de la actividad de negocio.
    - *Métricas de realización de actividad (leading)*: miden la realización de una actividad. Por ejemplo, la participación de una persona en un evento.

- *Métricas de resultado de una actividad (lagging)*: recogen los resultados de una actividad. Por ejemplo, la cantidad de puntos de un jugador en un partido.

- **Indicadores claves**: se entiende por este concepto, valores correspondientes que hay que alcanzar, y que suponen el grado de asunción de los objetivos. Estas medidas proporcionan información sobre el rendimiento de una actividad o sobre la consecución de una meta.
  - *Key performance Indicator (KPI)*: indicadores clave de rendimiento. Más allá de la eficacia, se definen unos valores que van a explicar en que rango óptimo de rendimiento se debería situar al alcanzar los objetivos. Son métricas del proceso. Por ejemplo, la radio de crecimiento de altas en un servicio.
  - *Key Goal Indicator (KGI)*: indicadores de metas. Definen mediciones para informar a la dirección general si un proceso TIC ha alcanzado sus requisitos de negocio, y se expresan por lo general en términos de criterios de información. Si se considera el KPI anterior, seria marcar un valor objetivo de crecimiento del servicio que se pretende alcanzar, por ejemplo, un 2%.

Existen también indicadores de desempeño. Los indicadores clave de desempeño (son, en definitiva, KPI) definen mediciones que determinan como se está desempeñando el proceso de TI para alcanzar la meta. Son los indicadores principales que indican si será factible lograr una meta o no, y son buenos indicadores de las capacidades, prácticas y habilidades.
Los indicadores de metas de bajo nivel se convierten en indicadores de desempeño para los niveles altos. [Fuente: (Curto, INFORMATION MANAGEMENT, 2008)]

7.5 Metodología para la presentación de Reportes OLAP: El Visor OLAP JPivot

En el desarrollo de la presente metodología, se hace uso de la tecnología que se menciona en el capitulo anterior JPivot, que combinada con el motor Mondrian y el contenedor de aplicaciones Apache Tomcat se puede hacer uso de la información de la bodega de datos y presentarla en reportes tipo ad-hoc.

7.5.1 JPIVOT

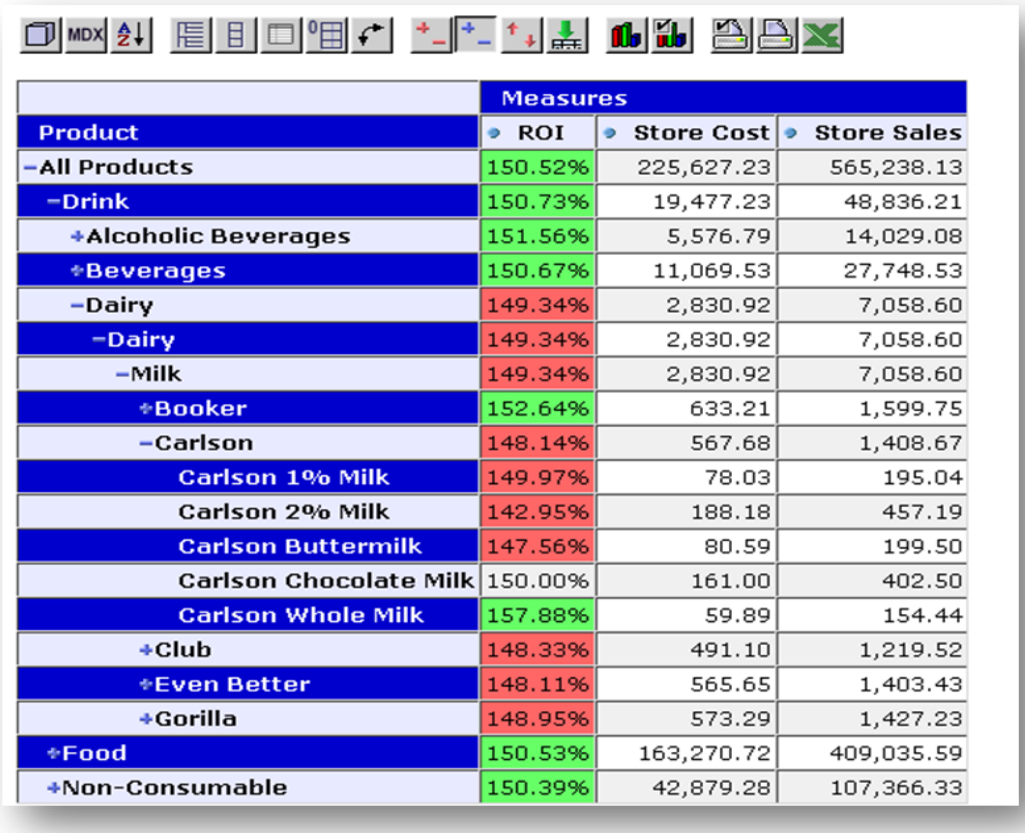
JPivot es un cliente OLAP basado en JSP que empezó en 2003. Puede considerarse un proyecto hermano de Mondrian dado que combinado con él permite realizar consultas tanto MDX como a partir de elementos gráficos que se renderizan en un navegador web. Durante largo tiempo ha sido el único visor existente para Mondrian. Pentaho ha adaptado su estilo para diferenciarlo de su interfaz original. [Tomado de: **(Díaz, Introducción al Bussines Intelligence, 2010)**]

Los reportes en JPivot permiten mostrar tablas y gráficos dinámicos, para mostrar la navegación típica de los entornos OLAP: drill-down, rotar ejes, dril-through, etc. Utiliza Mondrian como servidor OLAP preferente, pero también podría acceder a los cubos OLAP de Microsoft Analysis Services. La conexión con las Bases de Datos se realiza vía JDBC y realiza los cálculos en memoria, sin generar nuevos archivos y bases de datos que mantener y almacenar. La principal diferencia de JPivot respecto a otras bibliotecas en Javascript es que JPivot únicamente realiza la consulta de los datos necesarios, es decir los que se muestran en ese momento a diferencia de las bibliotecas de Javascript que contienen todos los datos de la consulta y los muestran de diferente manera en función de los filtros aplicados. [Fuente: (Díaz, Introducción al Bussines Intelligence, 2010)]

Las características principales de este visor analítico son:

- Capacidades de análisis interactivo a través de un acceso web basado en Excel, lo que proporciona una alta funcionalidad.
- Está basado en estándares de la industria (JDBC, JNDI, SQL, XML/A y MDX).
- Posibilidad de extenderse mediante desarrollo.

El menú de JPivot ofrece diversas opciones al usuario final:



	Measures		
Product	ROI	Store Cost	Store Sales
−All Products	150.52%	225,627.23	565,238.13
−Drink	150.73%	19,477.23	48,836.21
+Alcoholic Beverages	151.56%	5,576.79	14,029.08
+Beverages	150.67%	11,069.53	27,748.53
−Dairy	149.34%	2,830.92	7,058.60
−Dairy	149.34%	2,830.92	7,058.60
−Milk	149.34%	2,830.92	7,058.60
+Booker	152.64%	633.21	1,599.75
−Carlson	148.14%	567.68	1,408.67
Carlson 1% Milk	149.97%	78.03	195.04
Carlson 2% Milk	142.95%	188.18	457.19
Carlson Buttermilk	147.56%	80.59	199.50
Carlson Chocolate Milk	150.00%	161.00	402.50
Carlson Whole Milk	157.88%	59.89	154.44
+Club	148.33%	491.10	1,219.52
+Even Better	148.11%	565.65	1,403.43
+Gorilla	148.95%	573.29	1,427.23
+Food	150.53%	163,270.72	409,035.59
+Non-Consumable	150.39%	42,879.28	107,366.33

Figura 35. Reporte JPivot [Fuente: **(PHI Integration)**]

- Navegador OLAP: permite determinar las dimensiones que aparecen en las filas y/o columnas así como los filtros aplicados y las métricas por aparecer.
- Consulta MDX: permite visualizar y editar la consulta MDX que genera el informe OLAP.
- Configurar tabla OLAP: permite configurar aspectos por defecto de la tabla OLAP como el orden ascendente o descendente de los elementos.
- Mostrar miembros padre: permite mostrar u ocultar el padre del miembro de una jerarquía.
- Ocultar cabeceras: muestra u oculta las cabeceras repetidas para facilitar la comprensión del contenido.
- Mostrar propiedades: muestra u oculta las propiedades de los miembros de una jerarquía.
- Borrar filas o columnas vacías: muestra u oculta los valores sin contenido.
- Intercambiar ejes: intercambia filas por columnas.
- Botones de Drill: Member, Position y replace
    - Member: permite expandir todas las instancias de un miembro
    - Position: permite expandir la instancia seleccionada de un miembro
    - Replace: permite sustituir un miembro por sus hijos
- Drill through: permite profundizar en el detalle de información a partir de un nivel de información agregado superior.
- Mostrar gráfico: muestra un gráfico asociado a los datos. No todos los datos son susceptibles de generar gráficos consistentes.
- Configuración del gráfico: permite configurar las propiedades del gráfico. Desde el tipo del mismo hasta propiedades de estilo como tipo de letra, tamaño o color.
- Configuración de las propiedades de impresión/exportación: permite configurar las propiedades de la impresión como el título, disposición del papel, tamaño.
- Exportar a PDF/Excel: permite generar un PDF/Excel con el contenido del informe.

Cabe comentar que JPivot no es una herramienta orientada al usuario. No dispone de funcionalidades drag & drop ni tampoco otras funcionalidades como la creación de nuevas métricas o jerarquías.

5.2.2 Caso Práctico: Generación de reportes dinámicos usando JPivot

Para crear páginas con tablas y gráficos dinámicos utilizando JPivot y Mondrian es necesario el empleo de la tecnología JSP, las cuales contendrán las etiquetas propuestas por las librerías JPivot y WCF para mostrar los elementos deseados.

La tecnología que interpretará en este caso las páginas JSP es el Apache Tomcat, servidor de aplicaciones donde se hace el despliegue del motor **mondrian.war** el cuál contiene las librerías de JPivot

Figura 36. Logo Apache Tomcat *[Fuente:* **(The Apache Software Foundation, 2011)***]*

En la aplicación desplegada *mondrian.war* están contenidos los siguientes archivos los cuáles deben ser modificados con la información de la conexión JDBC a la base de datos donde se encuentra la Bodega de datos:

-Ruta_Tomcat/webapps/mondrian/WEB-INF/web.xml

```
Agreement, available at the following URL:
http://www.eclipse.org/legal/epl-v10.html.
(C) Copyright 2003-2009 Julian Hyde and others
All Rights Reserved.
You must accept the terms of that agreement to use this software.
-->

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

  <display-name>Mondrian</display-name>
  <description/>

  <!-- optional? now in JPivot by default -->
  <context-param>
    <param-name>contextFactory</param-name>
    <param-value>com.tonbeller.wcf.controller.RequestContextFactoryImpl</param-value>
  </context-param>

  <context-param>
    <param-name>connectString</param-name>
    <param-value>Provider=mondrian;Jdbc=jdbc:oracle:thin:@localhost:1521:xe;JdbcUser=olap;JdbcPassword=olap;
    JdbcDrivers=oracle.jdbc.driver.OracleDriver;Catalog=/WEB-INF/queries/SchemaVENTAS.xml</param-value>
  </context-param>

  <!-- optional
 <context-param>
  <param-name>chartServlet</param-name>
```

Figura 37. Configuracion web.xml **[Fuente: Elaboración propia]**

-Ruta_Tomcat/webapps/mondrian/WEB-INF/dataresources.xml

```
<?xml version="1.0"?>
<!--
<DataSources>
    <!--
    <DataSource>
        <!--
        <DataSourceName>Provider=Mondrian;DataSource=SchemaVENTAS;</DataSourceName>
        <!--
        <DataSourceDescription>Mondrian Ventas Data Warehouse</DataSourceDescription>
        <!--
        <URL>http://localhost/mondrian/xmla</URL>
        <!--
        <DataSourceInfo>Provider=mondrian;Jdbc=jdbc:oracle:thin:@localhost:1521:xe;JdbcUser=olap;JdbcPassword=olap;
        JdbcDrivers=oracle.jdbc.OracleDriver;Catalog=/WEB-INF/queries/SchemaVENTAS.xml</DataSourceInfo>
        <!--
        <ProviderName>Mondrian</ProviderName>
        <!--
        <ProviderType>MDP</ProviderType>
        <!--
        <AuthenticationMode>Unauthenticated</AuthenticationMode>
        <!--
        <Catalogs>
            <!--
                <Definition>/WEB-INF/queries/SchemaVENTAS.xml</Definition>
            </Catalog>
        </Catalogs>
    </DataSource>
</DataSources>
```

Figura 38. Configuración dataresources.xml **[Fuente: Elaboración propia]**

Usando las librerías y páginas JSP del Mondrian se puede editar y modificar los archivos contenidos en la carpeta *Ruta_Tomcat/webapps/mondrian/WEB-INF/queries/*

A continuación se muestra un ejemplo de archivo JSP, que recupera cuatro dimensiones del hecho Ventas, nótese los parámetros de la conexión ORACLE y la referencia al esquema SchemaVENTAS.xml del capítulo anterior y la consulta MDX para recuperar las dimensiones y los hechos del esquema:

Archivo ubicado en Ruta_*Tomcat/webapps/mondrian/WEB-INF/queries/fourhier.jsp*

```
<%@ page session="true" contentType="text/html; charset=ISO-8859-1" %>
<%@ taglib uri="http://www.tonbeller.com/jpivot" prefix="jp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>


<jp:mondrianQuery id="query01" jdbcDriver="oracle.jdbc.OracleDriver" jdbcUrl="jdbc:oracle:thin:@localhost:1521:xe"
    catalogUri="/WEB-INF/queries/SchemaVENTAS.xml" jdbcUser="olap" jdbcPassword="olap" connectionPooling="false">
select {[Measures].[CantidadVentas]} ON COLUMNS,
  {([Tiempo.Anno/Mes].[Anno/Mes], [Linea].[Linea], [Localizacion.Pais/Ciudad].[Pais/Ciudad], [Canal.Canal/PtoVenta].[Canal/PtoVenta])} ON ROWS
from [Ventas]

</jp:mondrianQuery>

<c:set var="title01" scope="session">4 Jerarquias en un eje</c:set>
```

Figura 39. Pagina JSP con query al esquema **[Fuente: Elaboración propia]**

Continuando con el ejemplo de este caso práctico, al acceder desde una navegador web a la página JSP con el código de la imagen anterior, se puede visualizar el resultado de la consulta MDX representado en una tabla dinámica:

				Medidas
Año>Mes	Canal > Punto de Venta	Linea	Pais>Ciudad	Valor Vendido
+2007	+Canal/PtoVenta	+Linea	+Pais/Ciudad	45.205.117.634
+2008	+Canal/PtoVenta	+Linea	+Pais/Ciudad	50.272.929.670
+2009	+Canal/PtoVenta	+Linea	+Pais/Ciudad	43.133.050.576
+2010	+Canal/PtoVenta	+Linea	+Pais/Ciudad	18.329.604.429

Figura 40. Reporte 4 Dimensiones **[Fuente: Elaboración propia]**

Y otros ejemplos de reportes usando este mismo cubo OLAP:

		Medidas		
Año>Mes	Pais>Ciudad	PU	Valor Vendido	Cantidad Vendida
+2007	+Pais/Ciudad	422.110,85	45.205.117.634	107.093
+2008	+Pais/Ciudad	413.561,34	50.272.929.670	121.561
+2009	-Pais/Ciudad	409.251,39	43.133.050.576	105.395
	-Colombia	490.375,89	32.497.700.392	66.271
	Bogotá	675.001,88	12.586.759.979	18.647
	Bucaramanga	570.543,84	7.012.554.347	12.291
	Medellín	397.972,86	7.237.534.424	18.186
	Cali	330.136,56	5.660.851.642	17.147
	+USA	458.545,62	4.003.103.292	8.730
	+Pánama	276.967,66	1.247.462.346	4.504
	+Venezuela	176.150,50	986.971.258	5.603
	+Chile	157.661,28	947.228.990	6.008
	+Perú	144.556,88	844.645.857	5.843
+2010	+Pais/Ciudad	420.230,28	18.329.604.429	43.618

Figura 41. Reporte JPivot Ventas **[Fuente: Elaboración propia]**

CONCLUSIONES

- El análisis de estado de las tecnologías utilizadas y disponibles para el desarrollo de un proyecto de inteligencia de negocios, presentado en el anexo 3; permitió profundizar en las distintas herramientas de software libre y propietario que están a la vanguardia de éste campo.

- Con el estudio de las herramientas de inteligencia de negocios se logró confirmar que las aplicaciones usadas en el desarrollo del presente documento corresponden a las idóneas para la implementación de este tipo de proyectos por ser las de mayor grado de completitud y ser también tecnologías maduras.

- La utilización de las herramientas Mondrian, JPivot y Pentaho permitió plantear una metodología que oriente en la construcción de cubos OLAP partiendo de bases de datos transaccionales. En cada uno de los pasos de la metodología se pudo observar cómo estas herramientas, siguiendo un orden lógico permiten diseñar este tipo de cubos.

- La documentación de conceptos de inteligencia de negocio, permitió apropiarse de terminología y definiciones necesarias para la implementación de éste tipo de proyectos.

- Las tecnologías usadas en el documento se han ido profundizando a medida que se abarcaban los temas que hacían uso de las mismas. Esta profundización permitió relacionar los conceptos de inteligencia de negocios a la metodología propuesta en el documento.

- El desarrollo de un caso práctico ligado a la metodología permitirá una mejor lectura de estudiantes o profesionales interesados en el tema.

- El desarrollo del presente trabajo de grado creó expectativas en quienes pudieron conocer acerca de los estudios y trabajos realizados, de lo que se espera la estructuración de proyectos futuros relacionados con la temática establecida y de esta manera se pueda aumentar la competitividad de las organizaciones mediante el análisis de su propia información.

BIBLIOGRAFIA

Díaz, J. C. (2010). *Introducción al Bussines Intelligence.* Barcelona: Editorial UOC.

Biosca, E. (19 de Marzo de 2009). *Open Source – Business Intelligence*. Recuperado el 20 de Junio de 2011, de Tutorial Introducción al MDX: http://www.enricbiosca.es/2009/05/tutorial-introduccion-al-mdx-capitulo-1.html

Chavez, C. A. (13 de Abril de 2005). *mailxmail.com*. Recuperado el 29 de Junio de 2011, de Diseño de bases de datos relacionales: http://www.mailxmail.com/curso-diseno-base-datos-relacionales/diseno-fisico-bases-datos

Curto, J. (Octubre de 2007). *INFORMATION MANAGEMENT*. Recuperado el 12 de Mayo de 2011, de Data Warehousing, Data Warehouse y Datamart: http://informationmanagement.wordpress.com/category/data-mart/

Curto, J. (19 de Noviembre de 2007). *INFORMATION MANAGEMENT*. Recuperado el 1 de Junio de 2011, de Diseño de un data warehouse: estrella y copo de nieve: http://informationmanagement.wordpress.com/2007/11/19/diseno-de-un-data-warehouse-estrella-y-copo-de-nieve/

Curto, J. (12 de Julio de 2008). *INFORMATION MANAGEMENT*. Recuperado el 30 de Mayo de 2011, de Diseño de un data warehouse: tabla de hecho: http://informationmanagement.wordpress.com/tag/factless-fact-tablescoverage-tables/

Curto, J. (15 de Octubre de 2010). *INFORMATION MANAGEMENT.* Recuperado el 2 de Julio de 2011, de ¿Qué es una Staging Area?: http://informationmanagement.wordpress.com/2007/10/15/%C2%BFque-es-una-staging-area/

Curto, J. (26 de Diciembre de 2008). *INFORMATION MANEGEMENT*. Recuperado el 30 de Mayo de 2011, de Diseño de un data warehouse: Slowly changing dimensions: http://informationmanagement.wordpress.com/2007/12/26/diseno-de-un-data-warehouse-slowly-changing-dimensions/

Cutro, A. (24 de Julio de 2009). *DataPrix*. Recuperado el 15 de Mayo de 2011, de Características de Pentaho: http://www.dataprix.com/723-caracter-sticas-pentaho

Díaz, J. C. (6 de Diciembre de 2010). *Tecnología al instante*. Recuperado el 20 de 6 de 2011, de OLAP: http://www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=5249

Franco, R. (2 de Agosto de 2010). *Blog Universidad Distrital*. Recuperado el 29 de Junio de 2011, de SIG. Resumen Unidad III. cap.3b: http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/m_logico.htm

Giménez, J. (5 de Junio de 2007). *TodoBI*. Recuperado el 20 de Febrero de 2011, de Como Funciona Mondrian OLAP: http://todobi.blogspot.com/2007/06/como-funciona-mondrian-olap.html

Inmon, W. H. (2005). *Building the Data Warehouse.* Octubre: Wiley.

Jaac316. (3 de Mayo de 2010). *Wikispaces*. Recuperado el 15 de Mayo de 2011, de Diferencias entre OLTP y OLAP: http://oltp.wikispaces.com/Diferencia+entre+OLTP+y+OLAP

McBurney, V. (28 de Marzo de 2008). *Knowledge sharing communities*. Recuperado el Junio de 2011, de Data Integration: http://it.toolbox.com/wiki/index.php/Data_integration_techniques

Oramas, J. (Agosto de 2009). *Asociación Colombiana de Ingenieros de Sistemas.* Recuperado el 12 de Junio de 2011, de ACIS: http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXIX_Salon_de_Informatica/2-JoaquinOramas-Arquitecturas_Empresariales.pdf

Ordoñez, M. E. (Mayo de 2011). *Un modelo de madurez de BI - Parte II.* Recuperado el 26 de Junio de 2011, de ACIS: http://www.acis.org.co/intelinfo/wp-content/uploads/2011/05/Un-Modelo-de-Madurez-de-BI-Parte-II.pdf

Pentaho Open Source BI. (2011). *Pentaho*. Recuperado el Junio de 2011, de Pentaho data Integration: http://www.pentaho.com/products/data_integration/

PHI Integration. (s.f.). *Pentaho PHI Integrations*. Recuperado el 10 de Junio de 2011, de JPivot with colors: http://pentaho-en.phi-integration.com/mondrian/configuring-mondrian-sample/jpivot-with-colors

Plauchu, E. (21 de Abril de 2011). *Gulsin 2.Org*. Recuperado el 1 de Junio de 2011, de Creando Mondrian Schemas con PSW: http://gulsin.org/2011/04/29/creando-mondrian-schemas/

The Apache Software Foundation. (2011). *Apache Tomcat*. Recuperado el 2011, de http://tomcat.apache.org/

Tindys. (28 de Febrero de 2010). *Buenas tareas*. Recuperado el 14 de Junio de 2011, de BD Dimensionales: http://www.buenastareas.com/ensayos/Bd-Dimensionales/142094.html

Torres, L. V. (Noviembre de 2008). *Google Docs.* Recuperado el 12 de Marzo de 2011, de La Inteligencia de Negocio. Su implementacion mediante la Plataforma Pentaho:

https://docs.google.com/viewer?url=http://www.redciencia.info.ve/memorias/ProyProsp/trabajos/l3.doc&pli=1

Universidad Peruana de Ciencias Aplicadas. (2008). *El Rincón del Vago.* Recuperado el 1 de Julio de 2011, de OLAP: http://html.rincondelvago.com/olap.html

Urquizu, P. (5 de Enero de 2010). *Business Intelligence fácil*. Recuperado el 23 de Junio de 2011, de http://www.businessintelligence.info/dss/datos-informacion-conocimiento.html

Velasco, R. H. (24 de Abril de 2004). *RHernando.* Recuperado el 29 de Marzo de 2011, de Tutorial de Data warehousing: http://www.rhernando.net/modules/tutorials/doc/bd/dw.pdf

Wikipedia. (5 de Mayo de 2011). Recuperado el 30 de Junio de 2011, de 12 reglas de Codd: http://es.wikipedia.org/wiki/12_reglas_de_Codd

Wikipedia. (15 de Abril de 2008). *Wikipedia*. Recuperado el 15 de Febrero de 2011, de Tabla de hechos: http://es.wikipedia.org/wiki/Archivo:Esquema_en_estrella.png

Wrembel, R. (2006). *Data Warehouses and OLAP Concepts, Architectures and Solutions.* IGI Global.

Yahazee. (13 de Diciembre de 2009). *Assembla*. Recuperado el 23 de Abril de 2011, de http://www.assembla.com/spaces/tabd_olap/wiki/Trabajo_Te%C3%B3rico/print

ANEXOS
Anexo 1: MDX

MDX es un lenguaje de consultas OLAP creado en 1997 por Microsoft. No es un estándar pero diversos fabricantes lo han adoptado como el estándar de hecho.
Tiene similitudes con el lenguaje SQL, si bien incluye funciones y fórmulas especiales orientadas al análisis de estructuras jerarquizadas que presentan relaciones entre los diferentes miembros de las dimensiones.

Sintaxis de MDX

La sintaxis de MDX es compleja; la mejor manera de ejemplificarla es a través de un caso determinado. Un ejemplo de un cubo de ventas con las siguientes dimensiones:

- Temporal de ventas con niveles de año y mes.
- Productos vendidos con niveles de familia de productos y productos.
- Medidas: importe de las ventas y unidades vendidas.

Para obtener, por ejemplo, el importe de las ventas para el año 2008 para la familia de productos lácteos, la consulta sería:

```
SELECT
    {[medidas].[importe ventas]}
    ON COLUMNS,
    {[tiempo].[2008]}
    ON ROWS FROM [cubo ventas] WHERE ([Familia].[lacteos])
```

Es posible observar que la estructura general de la consulta es de la forma SELECT…FROM…WHERE…:

- En el que SELECT se específica el conjunto de elementos que se quiere visualizar y debe especificarse si se devuelve en columnas o filas.
- En el FROM, el cubo donde se extrae la información.
- En el WHERE, las condiciones de filtrado.
- { } permite crear listas de elementos en las selecciones
- [ ] encapsulan elementos de las dimensiones y niveles.

Funciones de MDX

MDX incluye múltiples funciones para realizar consultas a través de la jerarquía existente en el esquema OLAP. Se puede destacar entre ellas:

- *CurrentMember:* permite acceder al miembro actual.
- *Children:* permite acceder a todos los hijos de una jerarquía.
- *prevMember:* permite acceder al miembro interior de la dimensión.
- *CrossJoin(conjunto_a,conjunto_b):* obtiene el producto cartesiano de dos conjuntos de datos.

- *BottomCount(conjunto_datos,N):* obtiene un número determinado de un conjunto, empezando por abajo, opcionalmente ordenado.
- *BottomSum(conjunto_datos,N,S):* obtiene de un conjunto ordenado los N elementos cuyo totoal es como mínimo el especificado (S).
- *Except(conjunto_a,conjunto_b):* obtiene la diferencia entre dos conjuntos.
- *AVG COUNT VARIANCE* y todas las funciones trigonométricas (seno, coseno, tangente, etc).
- *PeriodsToDate:* devuelve un conjunto de miembros del mismo nivel que un miembro determinado, empezando por el primer miembro del mismo nivel y acabando con el miembro en cuestión, de acuerdo con la restricción del nivel especificado en la dimensión de tiempo.
- *WTD(<miembro>):* devuelve los miembros de la misma semana del miembro especificado.
- *MTD(<miembro>):* devuelve los miembros del mismo mes del miembro especificado.
- *QTY(<miembro>):* devuelve los miembros del mismo trimestre del miembro especificado.
- *YTD(<miembro>):* devuelve los miembros del mismo año del miembro especificado.
- *ParallelPeriodo:* devuelve el miembro de un periodo anterior en la misma posición relativa que el miembro especificado.

Miembros calculados en MDX

Una de las funcionalidades más potentes que ofrece el lenguaje MDX es la posibilidad de realizar cálculos tanto dinámicos (en función de los datos que se están analizando en ese momento) como estáticos. Los cubos multidimensionales trabajan con medidas (del inglés *measures*) y con miembros calculados (del inglés *calculated members*).

Las medidas son las métricas de la tabla de hechos a las que se aplica una función de agregación (count, distinct count, sum, max, avg, etc.).
Un miembro calculado es una métrica que tiene como valor el resultado de la aplicación de una fórmula que puede utilizar todos los elementos disponibles de un cubo, así como otras funciones de MDX disponibles. Estas fórmulas admiten desde operaciones matemáticas hasta condiciones semafóricas pasando por operadores de condiciones.

[Fuente: (Biosca, 2009)]

Anexo 2: PENTAHO

PENTAHO es un proyecto iniciado por una comunidad OpenSource[21], provee una alternativa de soluciones de Inteligencia de Negocio en distintas áreas como en la Arquitectura, Soporte, Funcionalidad e Implantación. Estas soluciones al igual que su ambiente de implantación están basados en JAVA, haciéndolo flexible en cubrir amplias necesidades empresariales. A través de la integración funcional de diversos proyectos de OpenSource permite ofrecer soluciones en áreas como: Análisis de información, Reportes, Tableros de mando conocido como "DashBoards", Flujos de Trabajo y Minería de Datos.

Los módulos de la plataforma Pentaho BI son:

Pentaho Reporting: es una solución basada en el proyecto JFreeReport y permite generar informes ágiles y de gran capacidad. Pentaho Reporting permite la distribución de los resultados del análisis en múltiples formatos - todos los informes incluyen la opción de imprimir o exportar a formato PDF, XLS, HTML y texto. Los reportes Pentaho permiten también programación de tareas y ejecución automática de informes con una determinada periodicidad.

Pentaho Analysis: suministra a los usuarios un sistema avanzado de análisis de información. Con uso de las tablas dinámicas, generadas por Mondrian y JPivot, el usuario puede navegar por los datos, ajustando la visión de los datos, los filtros de visualización, añadiendo o quitando los campos de agregación. Los datos pueden ser representados en una forma de SVG o Flash, los cuadros de mando, o también integrados con los sistemas de minería de datos y los portales web. Además, con el Microsoft Excel Analysis Services, se puede analizar los datos dinámicos en Microsoft Excel (usando la conexión a OLAP server Mondrian).

Pentaho Dashboards: todos los componentes del modulo Pentaho Reporting y Pentaho Análisis pueden formar parte de un Dashboard[22]. En Pentaho Dashboards es fácil incorporar una gran variedad en tipos de gráficos, tablas y velocímetros e integrarlos con los Portales JSP, en donde podrá visualizar informes, gráficos y análisis OLAP.

Pentaho Data Mining: permite minería de datos a través de sofisticados algoritmos para descubrir patrones y correlaciones significativas que de otra manera se puede ocultar con el análisis y los reportes estándar. Éstos se pueden utilizar para ayudar a entender mejor la empresa y también para mejorar el rendimiento futuro a través de análisis predictivo.

[21] Código Abierto
[22] Tablero de Mando

Pentaho Date Integration: permite implementar los procesos ETTL[23]. Ofrece integración de datos de gran alcance mediante un enfoque innovador de metadatos.
[Fuente: **(Torres, 2008)**]

[23] En ingles *Extract, Transform, Transport and Load*. Extraer, transformar, transportar y cargar

Anexo 3: OTROS RECURSOS BI

Adicional a las herramientas usadas en éste libro, existe una gran variedad de herramientas tanto de software libre y propietario que apoyan en la implementación de proyectos de inteligencia de negocios. A continuación se listarán las más conocidas.

1. Soluciones Sofware libre:

- Pentaho: Una de las soluciones más completas líderes del mercado de software libre que integra ETL, reporting, OLAP, data mining y dashboards. En éste documento encontrará un anexo con más detalles de esta solución. URL: http://www.pentaho.com
- JasperSoft: Una de las soluciones completas líderes en el mercado de software libre que integra ETL, reporting, OLAP, data mining y dahsboards. Comparte el motor OLAP con Pentaho y su herramienta de ETL es la de Talend. URL: http://www.jaspersoft.com
- LucidDB: Base de datos en columnas de código abierto, optimizadas para análisis OLAP. URL: http://www.luciddb.org
- SpagoBI: Una de las soluciones completa líderes del mercado de software libre que integra ETL, reporting, OLAP, data mining y dashboards. Se diferencia del resto en que sólo existe una versión Community y que es completamente modular. URL: http://www.spagoworld.org/ecm/faces/public/guest/home/solutions/spagobi
- OpenReports: Solución que se basa en la integración de los tres motores de reporting de software libre existente y el motor OLAP Mondiran. URL: http://oreports.com
- BeeProject: Una de las primeras soluciones completas que actualmente es un proyecto abandonado. URL: http://sourceforge.net/projects/bee
- OpenI: Solución Business Intelligence basada en Mondrian. URL: http://www.openi.org
- MonetDB: Base de datos en columnas de código abierto, óptima para análisis OLAP. URL: http://monetdb.cwi.nl
- Ingres: Base de datos relacional de gran escalabilidad y rendimiento. Ofrece ampliaciones con JasperSoft, SpagoBI y Alfresco. URL: http://www.ingres.com
- Infobright: Motor analítico de gran rendimiento para procesos de Data Warehousing. Integrada con MySQL. URL: http://www.infobright.com
- Rapid Miner: Solución de minería de datos madura. Ofrece versión comercial y comunitaria. URL: http://rapid-i.com
- PMML (Predictive Model Markup Languaje): Es una markup language para el diseño de procesos estadísticos y de minería de datos. Usado por la gran mayoría de soluciones del mercado. URL: http://sourceforge.net/projects/pmml
- Vainilla/BPM-Conseil: Suite Business Intelligence de origen francés que nace con el objetivo de suplir las carencias de Pentaho y que cubre las principales necesidades de un proyecto de inteligencia de negocio. URL: http://www.bpm-conseil.com
- DataCleaner: Solución de software libre para la calidad de datos. URL: http://eobjects.org

- Palo BI Suite: Solución para la gestión de Spreadmarts, planificación y presupuestación basada en un motor MOLAP. URL: http://www.jedox.com
- Octopus: Solución de software libre para el desarrollo de procesos ETL. URL: http://octopues.objectweb.org
- Xineo: Solución de software libre para el desarrollo de procesos ETL. URL: http://sourceforge.net/projects/cb2xml
- CloverETL: Solución de software libre para el desarrollo de procesos ETL. URL: http://www.cloveretl.com
- Joost: Solución de software libre para el desarrollo de procesos ETL sobre archivos XML. URL: http://joost.sourceforge.net
- jRubik: Cliente OLAP para Mondrian. URL: http://rubik.sourceforge.net
- Apatar: Solución de software libre para el desarrollo de procesos ETL. URL: http://atapar.com
- Weka: Solución complete de minería de datos basada en algoritmos de aprendizaje automático procedente del contexto universitario. Ha sido comprado por Pentaho. URL: http://www.cs.waikato.ac.nz/ml/weka
- KETL: Solución de software libre para el desarrollo de procesos ETL. URL: http://sourceforge.net/projects/ketl

2. Soluciones Propietarias:

- Information Builders: Plataforma de desarrollo de aplicaciones BI. También tienen una solución de integración sumamente potente con más de 300 conectores. URL: http://www.informationbuilders.com
- IBM Cognos: IBM ha comprado Cognos y SPSS para incluir en su portafolio de productos una potente solución de inteligencia de negocio. URL: http://www-01.ibm.com/software/data/cognos
- SAP: Ofrece dos soluciones de BI: Business Object y Netweaver, que se hallan en un proceso de integración de roadmap. URL: http://www.sap.com
- Microstrategy: Una de las pocas soluciones de BI que no han sido compradas. Destaca por su potente capa de elementos de análisis. No incluye una herramienta de ETL. Actualmente existe una versión gratuita de funcionalidades reducidas. URL: http://www.microstrategy.com
- ORACLE: Oracle ha comprado Hyperion y otras soluciones para tener una suite de productos BI versátil y completa. URL: http://www.oracle.com
- Panorama: Una de las empresas tradicionales del sector que frecuentemente hace productos innovadores que son comprados por otras empresas. URL: http://www.panorama.com
- Apesoft: Empresa española que ofrece una suite flexible con un enfoque basado en Excel siguientdo un enfoque pragmático. URL: http://www.apesoft.com
- Actuate: Solución completa de Business Intelligence propietaria qu ofrece su motor de reporting BIRT en versión de software libre. URL: http://www.actuate.com

- Data Miner: Solución que incluye algoritmos de minería de datos en cuadros de mando e informes. URL: http://www.bissantz.com
- SAS: Solución de Minería de datos que incluye otros módulos que la convierte en una suite completa. URL: http://www.sas.com
- SVE: Empresa Santandereana con una solución de Balanced Scorecard que incluye otros módulos de SGC, Integridad Operativa y OLAP basada en Mondrian, que la convierten en una suite muy completa. URL: http://www.pensemos.com
- PushBI: Empresa orientada a soluciones de movilidad en el entorno de la inteligencia de negocio. URL: http://www.pushbi.com
- Informatica: Empresa con potentes soluciones de integración de datos así como de maestre data management. URL: http://www.informatica.com
- Expressor: Solución para integración de datos mediante el uso de capa de metadatos. URL: http://www.expressor-software.com
- I-lluminate: Empresa que ofrece una solución de data warehouse basada en la correlación de datos. URL: http://www.i-lluminate.com

[Fuente: (Díaz, Introducción al Bussines Intelligence, 2010)]