

HEFESTO DATA WAREHOUSING

Guía completa de aplicación teórico-práctica;
metodología Data Warehouse

**Bernabeu R. Dario
García Mattío Mariano**

Introducción

DATA WAREHOUSING: Marco Conceptual

HEFESTO: Metodología Data Warehouse

La elaboración de este texto es el producto de distintos recorridos profesionales, teóricos y técnicos en los que hemos ido construyendo un conocimiento sobre la temática referida al Business Intelligence, y que aquí queremos poner a disposición y compartir. Gran parte de los desarrollos que aquí se escriben han sido elaborados en una primera versión que data del año 2007. Desde este año en adelante hemos ido profundizando, revisando, acumulando un conjunto de saberes que hoy pueden sistematizarse en esta obra.

Los diferentes feedbacks, conversaciones, charlas mantenidas a lo largo de estos años entre los autores y con distintos profesionales en diversos ámbitos profesionales y laborales, han permitido ir puliendo las ideas y argumentos que se expresan en este libro. Ello ha posibilitado una escritura y un abordaje de los conceptos más accesible que sin invalidar la complejidad y solidez conceptual que el tema requiere, posibilita al lector una comprensión y un mayor acercamiento a los contenidos desarrollados.

Distintas instancias y espacios de formación, como el dictado de cursos, clases y talleres que hemos realizado utilizando este material nos ha ayudado mucho a replantear la estructura del documento, y la necesidad de utilizar ejemplos más concretos e imágenes, como recursos necesarios que colaboran en una mayor comprensión del material que se presenta.

A su vez, la aplicación de proyectos Business Intelligence en diversas empresas a lo largo de estos últimos 10 años, ha consolidado todos los conceptos que enunciábamos en nuestros inicios, añadiendo de este modo a la teoría la validación devenida de la práctica. También, una gran cantidad de trabajos finales de grado, tesis y tesinas de las carreras de ingeniería han tomado como fuente principal de consulta e información esa primera versión del 2007, lo cual ha permitido hacer dialogar esos conceptos con los marcos teóricos y técnicos de dichas producciones.

Nuestra satisfacción es grande al saber que nuestro pequeño aporte es bien recibido por la comunidad, y que hemos facilitado la introducción a este complejo mundo del Business Intelligence.

Recurso

Puede consultarse este libro en su versión digital [aquí...](http://troyanx.com/Hefesto) [<http://troyanx.com/Hefesto>]

Y en su versión PDF [aquí...](https://sourceforge.net/projects/bihefesto/files/Hefesto) [<https://sourceforge.net/projects/bihefesto/files/Hefesto>]

Indice

- Capítulo 0: Presentación | página **4**
- Capítulo 1: Business Intelligence | página **15**
- Capítulo 2: Data Warehousing & Data Warehouse | página **24**
- Capítulo 3: Arquitectura Data Warehousing | página **36**
- Capítulo 4: Complementos | página **111**
- Capítulo 5: Metodología HEFESTO | página **122**
- Capítulo 6: Diseño | página **164**

Capítulo 0: Presentación

- Licencia
 - Historial de cambios
 - Contacto
 - Colaboraciones
 - Notación
 - Software
-

Licencia

Este documento está protegido con licencia Creative Commons BY-NC-ND 4.0 International:



En donde:

- BY = Atribución. En caso de utilizar el material se debe dar crédito a sus creador@s e incluir un enlace de la licencia.
- NC = No comercial. NO se puede utilizar el material con fines comerciales.
- ND = Sin obra derivada. NO se puede modificar el material y redistribuirse.

Para ver más información sobre esta licencia [clic aquí...](https://creativecommons.org/licenses/by-nc-nd/4.0/) [https://creativecommons.org/licenses/by-nc-nd/4.0/]

Para ver el código legal de esta licencia [clic aquí...](https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode) [https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode]

Con esta licencia se garantiza la libertad de uso individual del material y se obtiene protección frente a usos fraudulentos.

Historial de cambios

Fecha	Versión	Autor@s	Detalle del cambio
Jueves 28 de Septiembre de 2017	3.0	Ing. Bernabeu R. Dario, Ing. García Mattío Mariano	Reestructuración y actualización.
Lunes 19 de Julio de 2010	2.0	Ing. Bernabeu R. Dario	Actualización.
Lunes 31 de Agosto de 2009	1.2	Ing. Fernández Carlos	Sección: Area de Datos.
Martes 21 de Abril de 2009	1.1	Ing. Bernabeu R. Dario	Actualización.
Sábado 17 de Enero de 2009	1.0	Ing. Bernabeu R. Dario	Actualización.
Miércoles 07 de Noviembre de 2007	0.1	Ing. Bernabeu R. Dario	Versión Inicial.

SourceForge

Este proyecto está alojado en [SourceForge...](http://sourceforge.net/projects/bihefesto/files/Hefesto/) [http://sourceforge.net/projects/bihefesto/files/Hefesto/]

Bernabeu Dario

Soy Bernabeu R. Dario, Ingeniero en Sistemas por el Instituto Universitario Aeronáutico (IUA).



Mi publicación más destacada es precisamente la que tienes en frente:

DATA WAREHOUSING: Marco Conceptual

HEFESTO: Metodología Data Warehouse

La primera versión de esta publicación la confeccioné en el año 2007, y con el paso del tiempo la he ido actualizando para que permanezca vigente, y he tenido colaboraciones concretas, las cuales se detallan más adelante.

Más sobre mi

Me especializo en el desarrollo e implementación de soluciones OSBI (Open Source Business Intelligence), SGBD y tecnologías web.

He publicado además artículos en la [revista Novatica](http://tgx-hefesto.blogspot.com.ar/2011/11/bi-usability-evolucion-y-tendencia.html). [<http://tgx-hefesto.blogspot.com.ar/2011/11/bi-usability-evolucion-y-tendencia.html>]

Soy coescritor de uno de los libros más destacados de Pentaho: [Pentaho 5.0 Reporting](http://www.packtpub.com/pentaho-5-0-reporting-by-example-beginners-guide/book). [<http://www.packtpub.com/pentaho-5-0-reporting-by-example-beginners-guide/book>]

Soy docente, investigador, geek y entusiasta del software libre.

Coordino la red social [Red Open BI](http://www.redopenbi.com/), y realizo numerosos aportes en diferentes foros, wikis, blogs, etc. [<http://www.redopenbi.com/>]

■ Mis canales:



▶ [<https://www.youtube.com/user/dariobernabeu>]

- ▶ [https://twitter.com/bernabeu_dario]
- ▶ [<http://www.linkedin.com/in/bernabeudario>]
- ▶ [<http://tgcx-hefesto.blogspot.com>]
- ▶ [<https://www.facebook.com/troyanx>]

■ Cursos Pentaho:



- ▶ [<http://troyanx.com/pentaho.html>]

■ Mail:

- ▶ darioSistemas@gmail.com
 - ▶ anteponer en el asunto el texto: **[HEFESTO]**
-

Colaboraciones

En esta publicación han colaborado:

- Ing. Fernández Carlos
- Ing. Mattío García Mariano

Fernández Carlos

Ing. Fernández Carlos ha sido uno de los principales promotores de esta publicación.

Le conocí a través de un foro, en donde compartíamos documentación sobre Business Intelligence, y desde entonces hemos realizado diversas colaboraciones.

Carlos fue quien a través de su portal [Dataprix.com](http://www.dataprix.com) puso a disposición el contenido de la publicación en formato html; facilitando de esta manera el acceso a la información mediante motores de búsqueda. [<http://www.dataprix.com/>]

En cuanto al contenido de la publicación Carlos es responsable de escribir la sección Areas de Datos.

Presentación de Carlos

Soy Carlos Fernández, Ingeniero en Informática de Gestión por la Universidad Politécnica de Catalunya (UPC).



A lo largo de los años me he ido especializando en tecnologías de gestión de los datos, especialmente en las áreas de Business Intelligence, Data warehousing, Integración y Arquitectura de datos.

Soy el creador de [Dataprix.com](http://www.dataprix.com), portal sobre tecnologías de la información, en el que he publicado numerosos artículos y compartido en lo posible mis opiniones y conocimientos técnicos con los miembros de la comunidad.

■ Mis canales:



- ▶ [<http://www.dataprix.com/blogs/carlos>]
 - ▶ [<https://www.linkedin.com/in/dataprix/>]
 - ▶ [<https://twitter.com/dataprix>]
-

Mattío García Mariano

Ing. Mattío García Mariano ha sido mi principal mentor a la hora de confeccionar esta publicación.

Mariano fue mi Profesor en la universidad; de ese tipo de profesores que dejan un poco de sí mismos en cada clase y que te inspiran con su ejemplo.

Gran colega, mejor amigo.

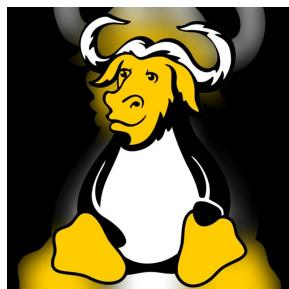
Hemos trabajado desde los inicios de Pentaho, y hemos recorrido muchísimos kilómetros dictando cursos a dúo; con una sinergia increíble.

En cuanto al contenido de la publicación, Mariano siempre ha estado presente para darme una mano, pero en esta ocasión hemos hecho algo que hace tiempo nos debíamos.

Hemos tomado a dúo la publicación y la hemos actualizado completamente. Cada párrafo, cada imagen, cada concepto, cada ejemplo.

Presentación de Mariano

Mi nombre es Mariano, García Mattio, soy Ingeniero en Sistemas del Instituto Universitario Aeronáutico y Especialista en Sistemas y Servicios Distribuidos de FaMAF/UNC.



Soy docente en grado y posgrado en áreas de Sistemas Distribuidos, Bases de Datos, Inteligencia de Negocios y Programación. Soy investigador en áreas de Sistemas Distribuidos y Ciberseguridad. Soy consultor independiente OSBI y desarrollo de sistemas Web.

Soy responsable de la coordinación y operación de las Olimpiadas de Programación/Robocode en el IUA y los cursos de capacitación en programación, bases de datos y sistemas operativos a estudiantes secundarios en las Sierras Chicas (Córdoba).

Soy autor de documentos presentados en diversos congresos de informática y afines.

Soy coescritor de uno de los libros más destacados de Pentaho: [Pentaho 5.0 Reporting By Example](http://www.packtpub.com/pentaho-5-0-reporting-by-example-beginners-guide/book). [<http://www.packtpub.com/pentaho-5-0-reporting-by-example-beginners-guide/book>]

Coordino la red social [Red Open BI](http://www.redopenbi.com/), y realizo aportes en diferentes foros, wikis, blogs, etc. [<http://www.redopenbi.com/>]

- Mis canales:



- ▶ [<https://www.linkedin.com/in/magm3333>]
 - ▶ [<https://twitter.com/magm3333>]
-

Notación

A partir de la versión 2.0 de esta publicación (año 2010), se han dejado de lado todos los términos que tienden a *masculinizar* el lenguaje y en su lugar se ha optado por otra forma de expresión que es inclusiva para todas las personas.

Por ejemplo, en vez de escribir **los usuarios**, se utiliza **l@s usuari@s**.

Software

Software utilizado para la confección del material:

- Suite Pentaho
- eXe Learning

Sistema operativo:

- Linux Mint
- Linux Ubuntu

Software adicional:

- Firefox/Chrome
- GIMP
- Inkscape
- Shutter

Otros recursos:

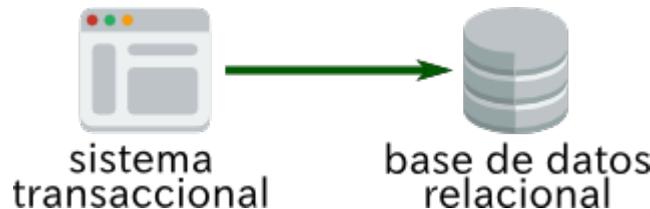
- Foto de portada: Emilce Riot
- Iconos: [ICOFINDER](https://www.iconfinder.com/) [https://www.iconfinder.com/]

Capítulo 1: Business Intelligence

- Datos como producto secundario
 - Definiendo al BI
 - Aplicando BI
 - Ambito de aplicación
 - Orígenes del BI
 - Proceso BI
 - Beneficios
-

Datos como producto secundario

En toda organización se generan datos constantemente, para la ejecución de sus operaciones y transacciones. Es muy común, que estos datos se administren a través de sistemas transaccionales y se almacenen en bases de datos relacionales, aunque esto no es excluyente.



Al pasar de los años, la acumulación de estos datos NO produce utilidad alguna, o lo hace de manera poco relevante a través de esporádicas consultas históricas. HEFESTO tiene como propósito poner en evidencia que esa acumulación deje de ser eso, datos acumulados, para comenzar a jugar un papel mucho más importante, y pueda constituirse en un increíble valor agregado. De allí, nos preguntamos:

- ¿Qué pasaría si de alguna manera procesamos todos estos datos y los utilizamos como fuente de información para la toma de decisiones?
- ¿Qué pasaría si además contamos con herramientas de software especializadas en la presentación de los datos para el estudio analítico?

El Business Intelligence (BI - Inteligencia de Negocios) es quien brindará la solución a nuestros interrogantes, en pos de mejorar el proceso de toma de decisiones.

Definiendo al BI

Se puede describir el Business Intelligence (BI - Inteligencia de Negocios), como la actividad de:

- almacenar y procesar grandes cantidades de datos,
- para que mediante la utilización de herramientas de software especializadas,
- sea sencillo el análisis y exploración de dichos datos,
- con el principal objetivo de obtener conocimiento (knowledge) orientado a tomar decisiones en tiempo real.



Este conocimiento debe ser:

- oportuno,
- relevante,
- útil y
- adaptado al contexto de la organización.

Existe una frase muy popular acerca de BI, que dice:

Inteligencia de Negocios es el proceso de convertir datos en conocimiento; el conocimiento fundamenta decisiones y éstas orienten acciones pertinentes y eficaces según los propósitos asumidos por la organización.

Aplicando BI

Cuando aplicamos BI a una organización, la fuente de datos (data source) principal es la que conforman los datos que la organización ha generado; con esta base analizaremos su comportamiento a lo largo del tiempo, desde diferentes escenarios y puntos de vista.

Una de las premisas del BI es que los objetivos de la organización se traduzcan en Indicadores de Estudio, por ejemplo si se trata de una entidad comercial: cantidad vendida, importe pagado, etc. Estos indicadores serán analizados aplicándoles diversos criterios, por ejemplo: año de venta, proveedor, cliente, rubro, etc; el conjunto de estos criterios conformarán las Perspectivas de Análisis.

Podemos decir entonces, que el BI nos permitirá analizar los Indicadores desde diferentes Perspectivas, y mediante ello responder preguntas sobre:

- lo que está sucediendo en la organización,
- lo que ha sucedido,
- lo que puede llegar a suceder y
- por qué.



Precisamente, la inteligencia de negocios permite que el proceso de toma de decisiones esté fundamentado sobre un amplio conocimiento de los procesos internos de la organización y del entorno, minimizando de esta manera el riesgo y la incertidumbre.

Ambito de aplicación

Las soluciones BI, NO necesariamente son aplicables a grandes y/o medianas empresas, como sí lo fue en sus inicios.

En la actualidad, puede aplicarse BI a cualquier organización, sin importar su tamaño y complejidad. Esto se debe principalmente a dos factores:

- el desarrollo de suites BI con licencia Free Software / Open Source, y
- la proliferación en internet de documentación, comunidades, foros, wikis, tutoriales y cursos.



En sus orígenes, el BI se encontraba orientado y acotado, fundamentalmente, a resolver problemáticas relacionadas al aumento de la rentabilidad, la disminución de costos y la obtención de ventajas competitivas de las organizaciones empresariales. Se trataba de una herramienta aplicada casi con exclusividad al campo económico, y su implementación y despliegue eran muy costosas.

Con el tiempo, sus finalidades y aplicación fueron ampliando y diversificándose, de modo tal que se constituyó en una herramienta para distintos campos sociales, y no circunscrita a lo económico.

Esta evolución permitió que el BI pueda ser aplicado a una biblioteca popular, a un centro vecinal o cualquier institución u organización que necesite tomar decisiones pertinentes en relación a los datos que produce.

Veamos entonces dos ejemplos de aplicación de BI:

1) Empresa de venta de productos: en este caso la aplicación de BI podrá resolver las siguientes preguntas.

- ¿Quiénes son l@s mejores client@s?
- ¿Cómo minimizar costos y maximizar las utilidades?
- ¿Cuál será el pronóstico de ventas del próximo mes?
- ¿Cuáles son los productos más vendidos por estación?
- ¿Han mejorado las ventas respecto del mismo período de un año anterior?

2) Biblioteca vecinal: en este caso la aplicación de BI podrá resolver las siguientes preguntas.

- ¿Cuál es la temática más consultada?
- ¿Qué días hay mayor concurrencia, y por qué?
- ¿Qué libros deben ser adquiridos?
- ¿Cuál es el rango etario que más lee cómics?
- ¿Qué perfil tienen las personas que leen determinada temática?

Orígenes del BI

El Business Intelligence tiene sus raíces en:

- los Executive Information Systems (EIS – Sistemas de Información Ejecutiva) y en
- los Decision Support Systems (DSS – Sistemas para la Toma de Decisiones).

Executive Information Systems

El EIS proporciona medios sencillos para consultar, analizar y acceder a la información del estado del negocio.

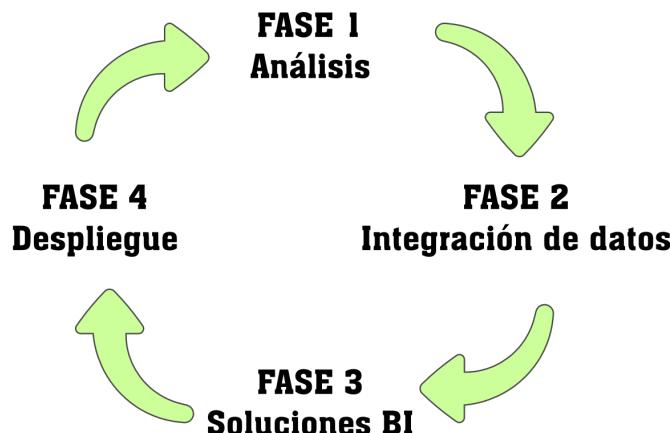
Decision Support Systems

Los DSS son una clase especial de sistemas de información cuyo objetivo es analizar datos de diferentes procedencias y brindar soporte para la toma de decisiones.

Proceso BI

El proceso mediante el cual una organización obtiene rédito de sus datos (los utiliza de forma inteligente) se denomina Proceso BI.

A continuación se enumeran sus fases:



- **FASE 1: Análisis.** Esta es la fase inicial en la que se recolectarán los requerimientos de información de l@s usuari@s. El foco estará puesto en conocer las necesidades de información de la organización, a partir de las cuales se redactarán las preguntas, cuyas respuestas ayudarán a tomar decisiones en pos de alcanzar los objetivos.
- **FASE 2: Integración de datos.** Esta fase tiene como fin extraer e integrar datos de diferentes fuentes (data sources). Las fuentes pueden ser de origen interno y/o externo según la necesidad. Las mismas producirán el flujo de datos que será el fundamento con el cual se encontrarán las respuestas a las preguntas planteadas en el paso anterior. Para la integración de los datos de las diversas fuentes, se les aplicará transformaciones a fin de compatibilizarlos con los requerimientos del análisis, y posteriormente se cargarán en la base de datos final, cuyo propósito es servir a BI.
- **FASE 3: Soluciones BI.** En esta fase se utilizarán herramientas, técnicas y componentes que permitirán la explotación de los datos. El objetivo será producir soluciones BI para que l@s usuari@s obtengan respuestas a las preguntas ya planteadas. Algunos de los componentes más utilizados son: reportes, indicadores, análisis interactivos, dashboards, gráficos estadísticos, etc.
- **FASE 4: Despliegue.** En esta fase se les entregará a l@s usuari@s los componentes BI que le correspondan y se les capacitará en su utilización, como así también en la creación de sus propias soluciones BI. L@s usuari@s obtendrán respuestas a sus preguntas e identificarán nuevas preguntas, cuyas respuestas requerirán que se inicie una nuevo ciclo del Proceso BI.

Beneficios

Entre los beneficios más importantes que BI proporciona a las organizaciones, se destacan los siguientes:

- Reduce el tiempo mínimo que se requiere para recoger todos los datos relevantes de un tema en particular, ya que los mismos se encontrarán integrados en una fuente única de fácil acceso.
- Automatiza la actualización de datos en la fuente integrada, a través de procesos predefinidos.
- Proporciona herramientas de análisis que servirán para establecer comparaciones y explorar los datos, con lo cual se mejorará notablemente la toma de decisiones.
- Completa el círculo que hace pasar de la decisión a la acción.
- Muchos análisis y reportes serán dinámicos y/o definidos por el usuari@ en el momento. Esto independizará a l@s usuari@s de los tradicionales informes pre-programados.
- Permite dar respuesta a preguntas de forma inmediata. También ayuda a la formulación de nuevas preguntas cuyas respuestas son clave para mejorar el desempeño de la organización.
- Permite acceder, analizar y monitorear directamente los Indicadores críticos de la organización.
- Identifica cuáles son los factores que inciden en el buen o mal funcionamiento de la organización.
- Se pueden detectar situaciones fuera de lo normal o potencialmente fuera de curso.
- Permite predecir el comportamiento futuro con un alto porcentaje de certeza, basado en el entendimiento del pasado.
- L@s usuari@s podrán consultar y analizar los datos de manera sencilla e intuitiva.

Capítulo 2: Data Warehousing & Data Warehouse

- Introducción
 - Data Warehousing
 - DWH vs DW
 - Data Warehouse:
 - ▶ Orientada al negocio
 - ▶ Integrada
 - ▶ Variante en el tiempo
 - ▶ No volátil
 - Cualidades del DW
 - Riesgos de aplicación
 - Redundancia
-

Introducción

La aplicación de una solución BI requiere integrar datos internos de la organización con datos externos, procesarlos, y luego almacenarlos en una base de datos que posibilite la accesibilidad y el posterior análisis.

Esto implica la necesidad de contar con un proceso que contemple y defina todas estas tareas. A este proceso se lo denomina Data Warehousing.

Data Warehousing

El Data Warehousing (DWH), es el proceso que reúne y ordena las tareas inherentes a:

- la extracción, transformación, consolidación, integración y
- centralización de los datos internos (datos que una organización genera en su actividad diaria, por ejemplo compras, ventas, producción, etc-) y los datos externos relacionados;
- permitiendo de esta manera el acceso, análisis y exploración,
- con el objetivo de dar soporte al proceso de toma de decisiones estratégico y táctico.

Para almacenar los datos necesarios para el análisis, se empleará un *Data Warehouse (DW)*.

Un Data Warehouse es una base de datos que posee una estructura multidimensional.

DWH vs DW

DWH (Data Warehousing) es un proceso que emplea un DW.

DW (Data Warehouse) es una base de datos con estructura multidimensional.

Pese a que las letras de su abreviatura son muy parecidas, y ello puede llevar a confusiones, es fundamental distinguir lo que es un proceso de lo que es una base de datos.



Data Warehousing (DWH)
es un proceso



Data Warehouse (DW)
es una base de datos

Data Warehouse

Una de las definiciones más famosas sobre DW, es la de William Harvey Inmon, quien es reconocido mundialmente como el padre del DW:

- Un Data Warehouse es una colección de datos orientada al negocio, integrada, variante en el tiempo y no volátil para el soporte del proceso de toma de decisiones de la gerencia.



Nota

Los términos almacén de datos y depósito de datos, son análogos a DW, y se utilizarán de aquí en adelante para referirse al mismo.

Orientada al negocio



Esto significa que:

- al DW solo ingresarán datos relevantes para el análisis y toma de decisiones. Por ejemplo, los datos referidos a los clientes, como su dirección de correo electrónico, fax, teléfono, código de identidad, código postal, etc., que son tan importantes de almacenar en cualquier sistema transaccional, NO son tenidos en cuenta para el DW por carecer de valor analítico, pero sí lo son aquellos que indican el tipo de cliente, su clasificación, ubicación geográfica, edad, etc.
- se manejarán entidades de alto nivel, es decir si una empresa maneja stock, listas de precios, cuentas corrientes, ventas, compras, etc., con lo que trabajaremos en el DW serán con entidades del tipo: clientes, productos, rubros, zonas, etc.
- la estructura será multidimensional, es decir que almacenará sus datos en tablas de Hechos y tablas de Dimensión.

Integrada



La integración implica que todos los datos provenientes de orígenes heterogéneos deben ser analizados a fin de asegurar su calidad y limpieza para luego ser consolidados en el DW. El proceso que permite esta consolidación, se denomina Integración de Datos, y cuenta con diversas técnicas y subprocesos para llevar a cabo sus tareas. Una de estas técnicas es el proceso ETL: Extracción, Transformación y Carga de datos (Extraction, Transformation and Load).

A continuación se describirán los orígenes de datos más comunes:

- Producidas por tipos de usuari@s:
 - ▶ Operacional: produce datos diariamente, en gran cantidad, muchos de los cuales son poco relevantes para el análisis por sí mismos, y la granularidad de estos datos es muy fina. Por ejemplo cuando se registran ventas de productos.
 - ▶ Medio: utiliza los datos operacionales para producir otros datos nuevos que tienen implicancia a corto/medio plazo. Por ejemplo, un control de stock, requerirá una o varias compras a fin de realizar el abastecimiento.
 - ▶ Gerencial: utiliza datos altamente procesados. En general, este perfil de usuari@ será el destinatario del DW, siempre produce una retroalimentación que permite generar nueva información para el análisis.
- Producidas por áreas o departamentos de la organización:
 - ▶ Las organizaciones se subdividen en áreas con responsabilidades bien definidas y diferentes. Cada una de estas áreas produce sus propios datos; éstos serán compartidos con otras áreas.
 - ▶ Las subdivisiones suelen ser geográficas, este tipo de división produce nuevos datos que deben ser incorporados. Por ejemplo: localización, cotizaciones, tipos de cambio, etc.
- Producidas por diferentes data sources:
 - ▶ Fuentes internas: datos que genera la empresa en sus actividades diarias.
 - ▶ Fuentes externas: datos que complementan y suplementan los datos internos. Por ejemplo: datos climáticos, análisis de tendencia, estadísticas, censos, etc.

Variante en el tiempo



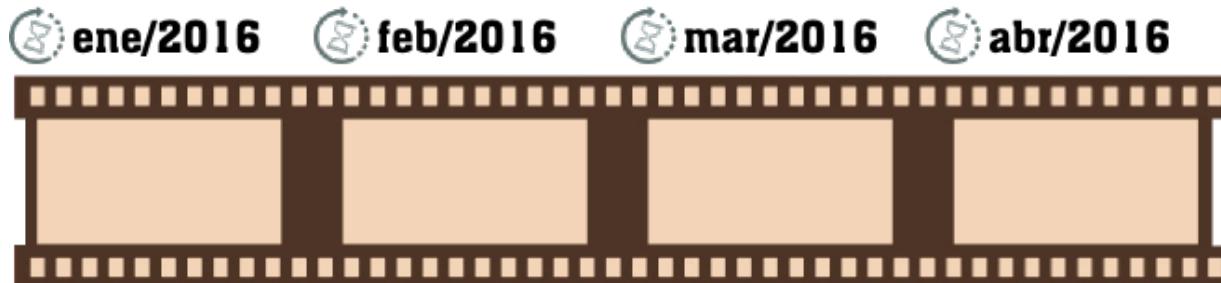
En el DW los datos actuales son almacenados junto a los datos históricos, y cada dato es marcado con su correspondiente sello de tiempo (timestamps).

Mediante este sello de tiempo se podrá tener acceso a diferentes versiones de una misma situación.

Es decir que se podrán observar los datos como si se tratase de una película de rollo, en donde cada fotograma posee:

- una escena en particular (situación de la empresa a analizar) y
- su sello de tiempo correspondiente (fecha/hora en que sucedió).

De esta manera, utilizando el sello de tiempo se podrán avanzar y retroceder los fotogramas del rollo, manteniendo el foco de atención sobre la situación analizada.



Es importante tener en cuenta y planificar la granularidad con que se almacenarán los datos.

No volátil



La información solamente será útil para el análisis y la toma de decisiones siempre y cuando ésta sea estable.

La naturaleza de los sistemas operacionales hacen que los datos que éstos administran varíen de forma permanente. Esto NO ocurre en los DW, ya que una vez que los datos ingresan NO cambian.

En un ambiente operacional son habituales las acciones de actualización (insertar, eliminar y modificar) y consulta; en cambio, en el DW la manipulación de los datos es más simple, solo existen dos tipos de acciones:

- Insertar: esta acción la realizan de forma programada los procesos de Integración de Datos.
- Consultar: esta es la única acción que l@s usuari@s pueden realizar sobre los datos.



DATA SOURCES



Consultar



Insertar



Eliminar



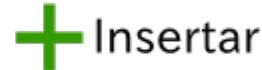
Modificar



DATA WAREHOUSE



Consultar



Insertar

Cualidades del DW

Las siguientes, son las cualidades más significativas del DW:

- Maneja un gran volumen de datos, debido a que integra los datos recolectados durante años, proveniente de diversos orígenes y fuentes, en una sola base de datos centralizada.
- Almacena datos agregados, actuales e históricos.
- Estructura los datos de forma multidimensional.

Riesgos de aplicación

Los riesgos más comunes que se pueden presentar en la implementación de un proceso de Data Warehousing son:

- Requiere una gran inversión, que en muchos casos se subestima. El proceso de DWH implica un amplio abanico de tareas que NO son sencillas de llevar a cabo y que requieren costos específicos (horas de desarrollo, horas de testing, horas de implementación, adquisición de hardware/software, capacitación de l@s usuari@s, etc.).
- Existe resistencia al cambio por parte de l@s usuari@s. La mejor forma de abordar este inconveniente es mediante una correcta capacitación.
- Los beneficios de su implementación NO se apreciarán en el corto plazo, pero sí lo harán en el mediano y largo plazo. Este punto deriva del anterior, y básicamente se refiere a que NO tod@s l@s usuari@s confiarán en el DW en una primera instancia, pero sí lo harán una vez que comprueben su efectividad y ventajas. Además, su correcta utilización surge de la propia experiencia y la capacitación obtenida.
- La manipulación de datos, atentará contra la privacidad de los datos sensibles o confidenciales, por ejemplo: listados de client@s, direcciones, contratos, datos médicos, etc. En este punto es imprescindible definir un esquema de roles que asegure la privacidad de acceso; sin embargo l@s desarrollador@s que se encarguen de los procesos de Integración de Datos, muy probablemente, tendrán acceso a datos sensibles o confidenciales.
- La subvaloración de los recursos necesarios para las soluciones de Integración de datos. Los procesos de integración de datos son sumamente complejos y son en definitiva la base en la que descansan el resto de los procesos.
- La subvaloración del esfuerzo necesario para un diseño correcto y en donde se tengan en cuenta todos los requerimientos del negocio.
- El incremento continuo de los requerimientos de l@s usuari@s. Una vez que el DW comienza a ser utilizado, l@s usuari@s tendrán una nueva forma de ver la información y en consecuencia una explosión de nuevas ideas, que se traducen en requerimientos para cubrir nuevas necesidades.
- La subestimación de las capacidades que puede brindar la correcta utilización del DWH y de las herramientas de BI en general. Una única capacitación NO es suficiente, sobre todo si esta instancia inicial suele ser una capacitación técnica del uso de herramientas. Éstas NO serán utilizadas en todo su potencial si NO se brindan capacitaciones tendientes a que l@s usuari@s conozcan conceptos de BI que devuelven esta potencialidad.

Redundancia

Debido a que el DW recibe información histórica de diferentes orígenes y fuentes, suele deducirse de forma errónea, que existe una repetición masiva de datos entre el ambiente DW y el operacional. Este razonamiento es superficial y por lo tanto erróneo; en realidad existe una mínima redundancia de datos entre ambos ambientes.

Para entender claramente lo antes expuesto, se debe considerar lo siguiente:

- Del ambiente operacional solamente se seleccionan datos relevantes, los cuales se codifican, filtran y transforman antes de ingresar al DW.
- El horizonte de tiempo, nivel de granularidad y agregación, son muy diferentes entre los dos ambientes.
- Algunos estudios indican que la redundancia encontrada al cotejar los datos de ambos ambientes es mínima, y que generalmente resulta en un porcentaje menor del 1%.

Capítulo 3: Arquitectura Data Warehousing

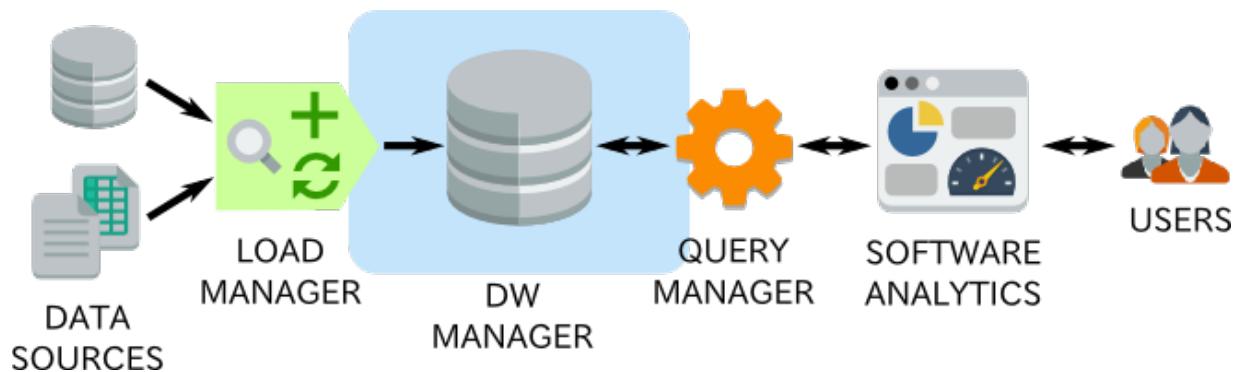
- Introducción
- 1) Data Sources
- 2) Load Manager
 - ▶ Extracción
 - ▶ Transformación
 - ▶ Codificación
 - ▶ Medida de Atributos
 - ▶ Fuentes múltiples
 - ▶ Data Cleaning
 - ▶ Carga
 - ▶ Proceso ETL
- 3) Data Warehouse Manager
 - ▶ Base de datos multidimensional
 - ▶ Tablas de Dimensiones
 - ▶ Tiempo
 - ▶ Tablas de Hechos
 - ▶ Agregadas y Preagregadas
 - ▶ Cubo Multidimensional: introducción
 - ▶ Representación matricial
 - ▶ Modelos del DW
 - ▶ Estrella
 - ▶ Copo de Nieve
 - ▶ Constelación
 - ▶ OLTP vs DW
 - ▶ Implementación
 - ▶ ROLAP
 - ▶ MOLAP
 - ▶ HOLAP
 - ▶ Cubo Multidimensional: profundización
 - ▶ Jerarquías
 - ▶ Metadatos
- 4) Query Manager
 - ▶ Drill-up
 - ▶ Drill-down
 - ▶ Drill-across
 - ▶ Roll-across
 - ▶ Pivot
 - ▶ Page

- ▶ Drill-through
 - 5) Software Analytics
 - ▶ Interacción
 - ▶ Características
 - ▶ Reporting
 - ▶ OLAP
 - ▶ Dashboards
 - ▶ Data Mining
 - ▶ Redes Neuronales
 - ▶ Sistemas Expertos
 - ▶ Programación Genética
 - ▶ Árboles de Decisión
 - ▶ Detección de Desviación
 - ▶ EIS
 - 6) Users
-

Introducción

Teniendo en cuenta las características que definen el Data Warehousing y que se han desarrollado en los apartados anteriores, en este capítulo se describirán los componentes de su arquitectura. .

En el siguiente gráfico se pueden visualizar los diferentes componentes de la estructura del DWH:



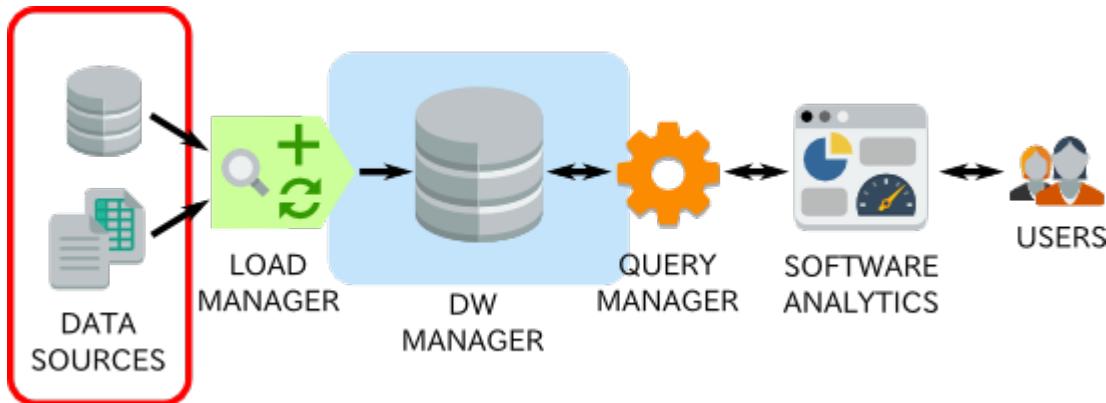
Como se puede apreciar, el ambiente del DWH está formado por diversos componentes que interactúan entre sí y cumplen una función específica dentro del sistema.

Su funcionamiento puede resumirse de la siguiente manera:

- Los datos son extraídos desde distintas fuentes, bases de datos, archivos, servicios web, etc. Estos datos, generalmente, residen en diferentes tipos de sistemas con arquitecturas y formatos variados.
- Los datos son integrados, transformados y limpiados, para luego ser cargados en el DW.
- Con base en el DW, se construirán Cubos Multidimensionales y/o Business Models.
- Los usuarios accederán a los Cubos Multidimensionales, Business Models (u otro tipo de estructura de datos) del DW, utilizando diversas herramientas de consulta, exploración, análisis, reportes, etc.

A continuación se enumerarán y explicarán los componentes de la arquitectura del Data Warehousing, utilizando como guía el gráfico antes expuesto.

1) Data Sources



Los Data Sources (origenes de datos) representan los datos transaccionales que genera la empresa en su accionar diario, junto a otros datos internos y/o externos complementarios.

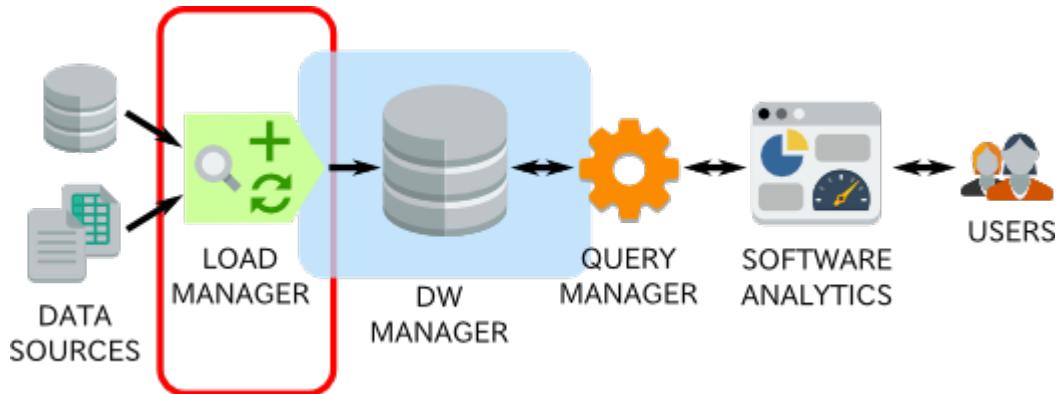
Los Data Sources poseen características muy disímiles entre sí, en formato, procedencia, función, etc. En la actualidad esto se ve potenciado gracias a los web services, redes sociales, y a la utilización cada vez más frecuente de bases de datos NoSQL.



Los Data Sources más habituales para extraer datos relevantes son:

- Archivos de texto
- Hojas de cálculos
- Informes semanales, mensuales, anuales, etc.
- Bases de datos transaccionales, SQL y *NoSQL*
- Información no estructurada (páginas web, mails)
- Redes sociales
- RSS
- Web Services

2) Load Manager



El componente Load Manager es el encargado de la ejecución y calendarización (scheduling) de los diferentes procesos de Integración de Datos a través de los cuales:

- se extraerán los datos desde los Data Sources,
- serán manipulados, integrados y transformados, para luego
- cargar los resultados obtenidos en el DW.

La Integración de Datos es una serie de técnicas y procesos que se encargan de llevar a cabo todas las tareas relacionadas con la extracción, manipulación, control, integración, depuración de datos, carga y actualización del DW, etc. Es decir, todas las tareas que se realizarán desde que se obtienen los datos de los diferentes Data Sources hasta que se cargan en el DW.

Si bien el proceso ETL (Extraction, Transformation, Load) es solo una de las muchas técnicas de la Integración de Datos, es la más importante, incluso en muchos casos constituyen el proceso de integración en si. En este orden, se puede ubicar el resto de las técnicas en las diferentes etapas del ETL:

- el proceso Extracción incluirá técnicas enfocadas por ejemplo a obtener desde diversas fuentes solamente los datos relevantes y mantenerlos en una *Staging Area* (almacenamiento intermedio);
- el proceso Transformación incluirá técnicas encargadas de compatibilizar formatos, filtrar y clasificar datos, relacionar diversas fuentes, etc;
- el proceso Carga incluirá técnicas propias de la carga de datos y actualización del DW.

A continuación, se detallarán cada una de estas etapas, se expondrá cuál es el proceso que llevan a cabo los ETL y se enumerarán cuáles son sus principales tareas.

Extracción

La selección de los Data Sources para proveer todos los datos que sean relevantes, tiene que hacerse teniendo en cuenta las necesidades de l@s usuari@s y requisitos definidos para la solución.

En la mayoría de los casos, los Data Sources a utilizar serán bases de datos relacionales, con lo cual la extracción puede llevarse a cabo mediante consultas SQL o procedimientos almacenados. Pero en el caso de Data Sources NO convencionales o NO estructurados, la obtención será más difícil.

Una vez seleccionados y extraídos, los datos, deben ser persistidos en una base de datos relacional Staging (almacenamiento intermedio), lo cual permitirá:

- Manipular los datos sin interrumpir ni sobrecargar los Data Sources y el DW.
- Crear una capa de abstracción entre la lectura y la carga.
- Almacenar y gestionar los metadatos que se generan en los procesos ETL.
- Facilitar la integración.

El Staging Area, generalmente, se constituye en una o más bases de datos relacionales donde la información puede ser persistida en tablas auxiliares, tablas temporales, etc. Una vez que los datos se encuentren en Staging Area, el proceso puede desconectarse de los Data Sources y continuar con la transformaciones necesarias para poblar el DW.

Transformación

Esta función es la responsable de aplicar todas las acciones necesarias sobre los datos, a fin de hacerlos consistentes, compatibles y congruentes con el DW.

Los casos más comunes en los que se debe realizar integración, son los siguientes:

- Codificación.
- Medida de Atributos.
- Fuentes múltiples.

Además de lo antes mencionado, esta función se encarga de realizar los procesos:

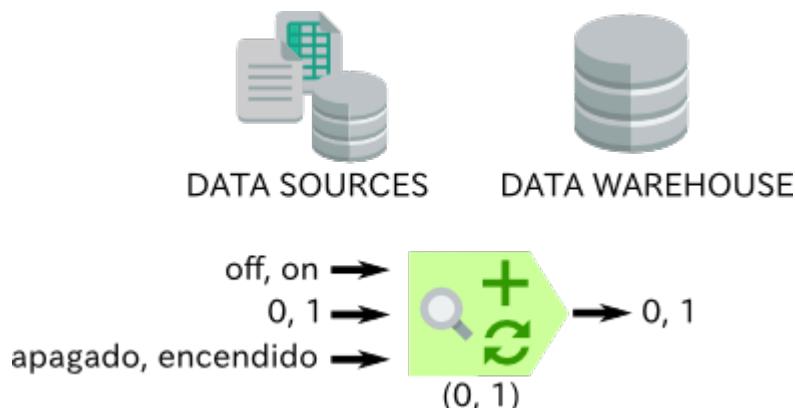
- Data Cleansing (Limpieza de Datos).
- Data Quality (Calidad de Datos).

Codificación

Una inconsistencia típica que se encuentra al integrar varios Data Sources, es la de contar con más de una forma de codificar un Atributo en común. Por ejemplo, al especificar el valor que tendrá el campo **status**, en un Data Source se pueden encontrar valores **0** y **1**, en otros **Apagado** y **Encendido**, en otros **off** y **on**, etc.

Se deberá seleccionar una de las codificaciones existentes o bien aplicar otra, pero todos los datos de ese Atributo deben respetar una única convención de codificación antes de ingresar al DW.

En la siguiente figura, se puede apreciar que de un conjunto de diferentes formas de codificar se escoge solo una. Cuando se obtiene un dato con una codificación que difiere de la seleccionada, es cuando se procede a su transformación.

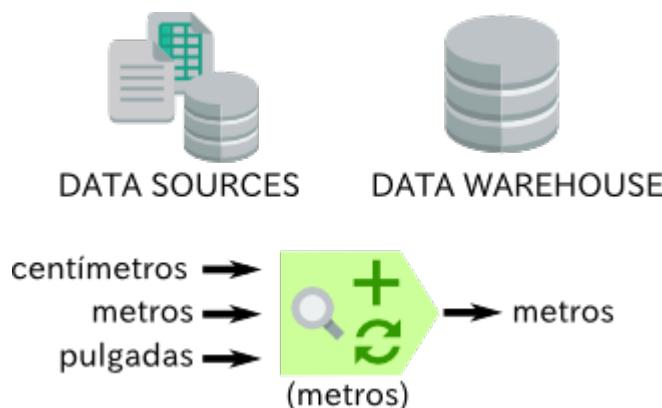


Medida de Atributos

Los tipos de unidades de medidas utilizados para representar los Atributos, pueden variar en cada Data Source. Por ejemplo, al registrar la longitud de un producto determinado, las unidades de medida pueden ser expresadas en centímetros, metros, pulgadas, etc.

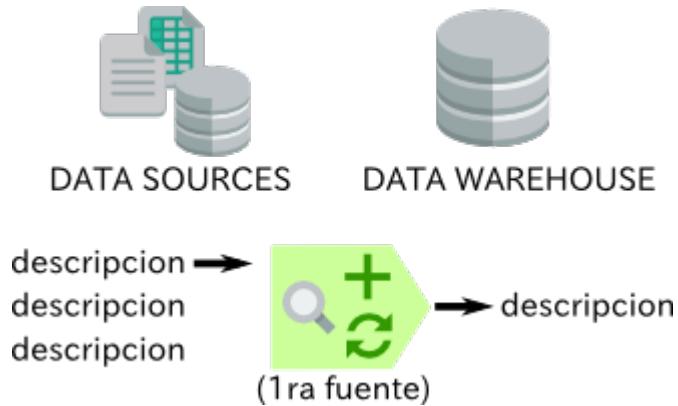
Este proceso implica redimensionar el valor asociado a la unidad de medida. Este redimensionamiento suele ejecutarse multiplicando el valor por un factor de escala. Por ejemplo si la unidad de medida seleccionada es metro y desde un Data Source vienen centímetros, los valores se deben multiplicar por el factor de escala **0,01** a fin de que se mantengan íntegros.

En la siguiente figura, se puede apreciar que de diferentes unidades de medida se escoge solo una. Cuando surge un dato con una unidad de medida que difiere de la seleccionada, es cuando se procede a su transformación.



Fuentes múltiples

Un mismo elemento puede derivarse desde varios Data Sources. En estos casos, se debe elegir aquella fuente que se considere más fiable y apropiada.



Data Cleansing

El objetivo principal del Data Cleansing (Limpieza de Datos) es entregar un flujo de datos lo más sanitizado posible. Por lo tanto podrá llevar a cabo acciones tales como eliminar datos erróneos o irrelevantes y subsanar aquellos que presenten inconsistencias.

Las acciones más típicas que se pueden llevar a cabo al encontrarse con Outliers (Datos Anómalos) son:

- Ignorarlos.
- Eliminar la columna (se eliminan el 100% de los datos).
- Filtrar la columna (se eliminan algunos de los datos).
- Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
- Reemplazar el valor.
- Discretizar los valores de las columnas. Por ejemplo, si un campo numérico presenta un valor de **1** a **2**, utilizamos el texto **Bajo**; de **3** a **7**, **Óptimo**; de **8** a **10**, **Alto**. Entonces, cuando suceda un Outlier se puede reemplazar por **Bajo** o **Alto**.

Las acciones que suelen efectuarse contra Missing Values (Datos Faltantes) son:

- Ignorarlos.
- Eliminar la columna (se eliminan el 100% de los datos).
- Filtrar la columna (se eliminan algunos de los datos).
- Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
- Reemplazar el valor.
- Esperar hasta que los datos faltantes estén disponibles.

Antes de elegir alguna acción, es muy importante que se identifique el por qué de la anomalía, para luego actuar en consecuencia, con el fin de evitar que se repitan, agregando de esta manera más valor a los datos de la organización. Puede suceder que en algunos casos, los valores faltantes sean inexistentes, por ejemplo, cuando l@s nuev@s asociad@s o client@s, no posean consumo medio del último año.

Carga

Esta función es responsable de ejecutar las tareas relacionadas con Carga Inicial (Initial Load) y Actualización periódica (Update).

- La Carga Inicial (Initial Load), se refiere a la primera carga de datos que recibe el DW. Generalmente, esta tarea consume un tiempo considerable, debido a que se insertan gran cantidad de registros pertenecientes a períodos largos de tiempo.
- La Actualización Periódica (Update), se refiere a la inserción de pequeños volúmenes de datos, y su frecuencia está dada en función de la granularidad (cuán resumidos se encuentran los datos) del DW y los requerimientos de l@s usuari@s. El objetivo de esta tarea es añadir al DW solo aquellos datos que se generaron a partir de la última actualización (delta de cambios).

Previo a una actualización, se deben identificar los cambios (delta) en las fuentes originales; esto se realiza, en la mayoría de los casos, mediante la fecha de la última actualización. Para efectuar esta operación, se pueden realizar las siguientes acciones:

- ▶ Cotejar las instancias de los Data Sources involucrados.
- ▶ Utilizar triggers (herramienta de los SGBD que consta de una porción de código que se dispara de forma automática ante un evento) para informar de los cambios sucedidos en los Data Sources.
- ▶ Recurrir a Marcas de Tiempo (Time Stamp), en los registros de los Data Sources.
- ▶ Comparar los datos existentes entre el Data Source y el DW.
- ▶ Hacer uso de técnicas mixtas.

Si este control consume demasiado tiempo y esfuerzo, o simplemente NO puede llevarse a cabo por algún motivo en particular, existe la posibilidad de cargar el DW desde cero: este proceso se denomina Carga Total (Full Load). Esta acción involucra el vaciado previo del DW.

El proceso de Carga también es responsable de mantener la estructura del DW e involucra conceptos como:

- Relaciones muchos a muchos.
- Claves Subrogadas.
- Dimensiones Lentamente Cambiantes.
- Dimensiones Degeneradas.

Proceso ETL

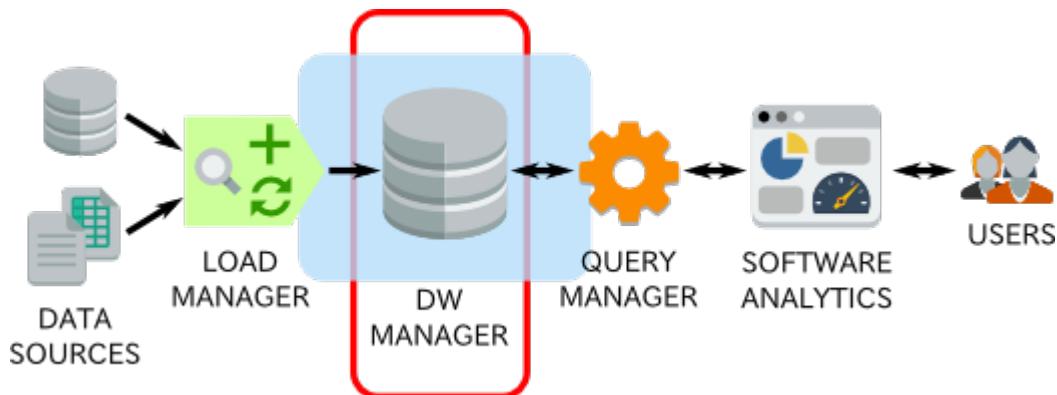
A continuación detallaremos la relación que existe entre los procesos ETL (Extracción, Transformación y Carga), los Data Sources, el Staging Area y el DW:



El proceso ETL trabaja de la siguiente manera:

- Se extraen los datos relevantes desde los Data Sources y se depositan en la Staging Area.
- Se integran y transforman los datos, para evitar inconsistencias.
- Se cargan los datos desde el Staging Area al DW.

3) Data Warehouse Manager



El DW Manager está compuesto por una serie de aplicaciones de software dedicadas a gestionar:

- el DW (SGDB),
- las conexiones a base de datos y otros Data Sources,
- las estructuras de datos (Cubos Multidimensionales, Business Models, etc.),
- información de autenticación y autorización (credenciales de acceso, users, roles, permisos, etc), y
- otros metadatos.

Base de datos multidimensional

Un Data Warehouse es una base de datos con estructura multidimensional, lo cual se traduce en una forma específica de almacenamiento en la cual se definen dos estructuras principales :

- tablas de Hechos y
- tablas de Dimensiones.

La utilización de tablas de Hechos y Dimensiones, facilita la creación de estructuras de datos (Cubos Multidimensionales, Business Models, etc.) y posibilita que las consultas al SGBD sean respondidas con mucha performance.

Tipos de Modelos lógicos

Existen tres formas de modelar las tablas de Hechos y Dimensiones:

-  Esquema en estrella (Star Scheme).
-  Esquema copo de nieve (Snowflake Scheme).
-  Esquema constelación (Starflake Scheme).

Estos modelos están concebidos con el objetivo de facilitar el acceso a consultas complejas y con gran cantidad de agregaciones, es por ello que se encuentran desnormalizadas o semi desnormalizadas, reduciendo de esta manera al mínimo la cantidad de JOINs que deben emplearse para acceder a los datos requeridos.

Tipos de Implementación

Los modelos lógicos pueden ser implementados de diferentes maneras:

-  Relacional – ROLAP.
-  Multidimensional – MOLAP.
-  Híbrido – HOLAP.

Tablas de Dimensiones

Las tablas de Dimensiones proveen el medio para analizar los datos dentro del contexto del negocio. Veamos algunos ejemplos:



Las tablas de Dimensiones:

- contienen datos cualitativos y
- representan los aspectos de interés,
- mediante los cuales l@s usuari@s podrán filtrar y manipular los Hechos almacenados en las tablas de Hechos.

Las tablas de Dimensión contienen los siguientes tipos de campos:

- Clave principal.
- Claves foráneas (solo para esquemas copo de nieve y constelación).
- Datos de referencia *primarios*: datos que identifican la Dimensión. Por ejemplo: nombre del cliente.
- Datos de referencia *secundarios*: datos que complementan la descripción de la Dimensión. Por ejemplo: e-mail del cliente, celular del cliente, etc. Estos datos no son significativos para tomar decisiones, pero son potencialmente valiosos para implementarla.

Usualmente la cantidad de tablas de Dimensiones, aplicadas a un tema de interés en particular, varían entre tres y quince.

Es recomendable que la clave principal de las tablas de Dimensiones sea independiente de las claves de los Data Sources, ya que, por ejemplo, si estos últimos son recodificados, el DW quedaría inconsistente. Estas claves independientes se denominan Claves Subrogadas.

Tiempo

En un DW, la creación y el mantenimiento de una tabla de Dimensión Tiempo es obligatoria, y la definición de granularidad y estructuración de la misma depende de la dinámica del negocio que se esté analizando. Toda la información dentro del DW posee su propio sello de tiempo que determina la ocurrencia de un Hecho específico, representando de esta manera diferentes versiones de una misma situación.

La Dimensión Tiempo NO es sola una secuencia cronológica representada de forma numérica, sino que mantiene niveles jerárquicos especiales que son representativos de las actividades de la organización. Esto se debe a que l@s usuari@s podrán por ejemplo analizar las ventas realizadas teniendo en cuenta el día de la semana en que se produjeron, quincena, mes, trimestre, semestre, año, estación, etc.

La forma de diagramar la Dimensión Tiempo es muy sencilla, se debe tomar el campo que indique la fecha en que sucedieron los Hechos y luego analizar dicha fecha para crear los campos requeridos.

Veamos un ejemplo:



En el ejemplo anterior se ha definido más de una Jerarquía para la Dimensión tiempo. Esto es lógico ya que hay Jerarquías que NO son compatibles, por ejemplo **anio-semestre-bimestre** es una Jerarquía compatible, pero **anio-trimestre-bimestre** NO lo es ya que NO todos los **bimestres** están contenidos en un **trimestre**.

Existen muchas maneras de diseñar esta tabla, por lo cual debemos evaluar la temporalidad de los datos, la forma en que trabaja la organización, los resultados que se esperan obtener del DW relacionados con una unidad de tiempo y la flexibilidad que se desea obtener de dicha tabla.

Fecha & Hora



Si se requiere analizar los datos por fecha (año, mes, día, etc) y por hora (hora, minuto, segundo, etc), lo más recomendable es confeccionar dos tablas de Dimensión Tiempo:

- una conteniendo los datos referidos a la fecha y
- la otra, con los datos referidos a la hora.

Día Juliano

Si bien, el lenguaje SQL ofrece funciones del tipo **DATE**, en la tabla de Dimensión Tiempo, se modelan y presentan datos temporales que, en algunos casos, son complejos de calcular en una consulta.

Es conveniente mantener en la tabla de Dimensión Tiempo un campo que se refiera al día Juliano. El día juliano se representa a través de un número secuencial e identifica únicamente cada día. Mantener este campo permitirá la posibilidad de realizar consultas que involucren condiciones de filtrado de fechas desde-hasta, mayor que, menor que, etc, de manera sencilla e intuitiva.

Por ejemplo, si a partir de determinada fecha se desean analizar los datos de los 81 días siguientes:

- el valor **desde** sería el día Juliano de la fecha en cuestión y
- el valor **hasta** sería igual a **desde** más **81**.

Tablas de Hechos

Las Tablas de Hechos contienen los Hechos que serán utilizados por l@s usuari@s del DW para analizar y responder preguntas de negocio.

Veamos un ejemplo:

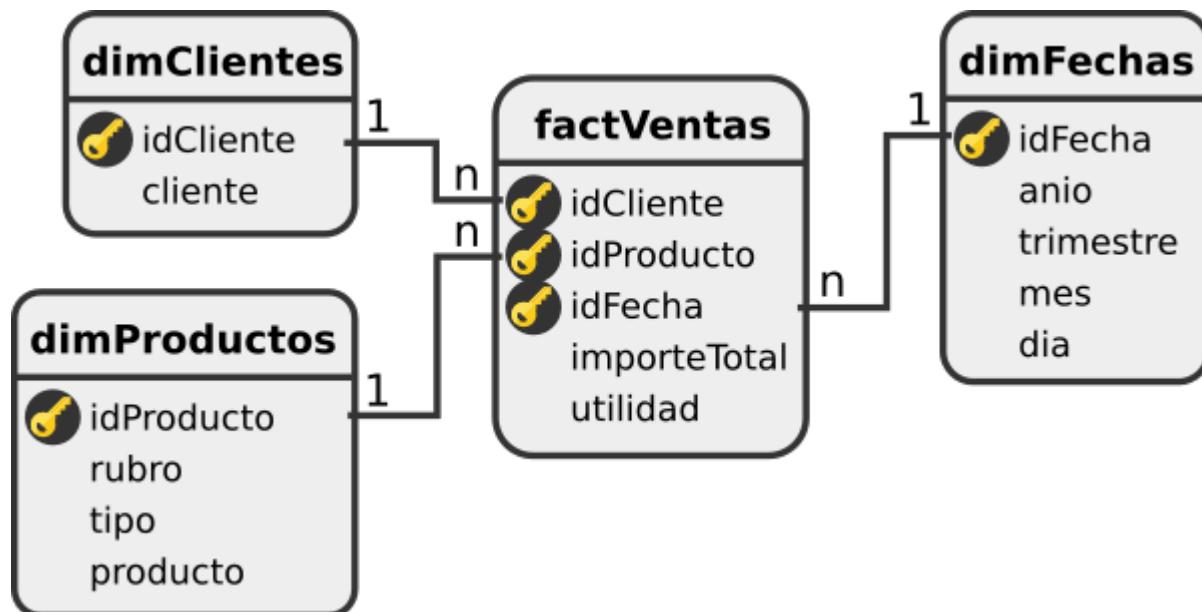


Los Hechos (o Dato agregado):

- son datos cuantitativos,
- que son filtrados, agrupados y explorados a través de condiciones definidas en las tablas de Dimensiones.

Los datos presentes en las tablas de Hechos constituyen el volumen del DW, y pueden estar compuestos por millones de registros dependiendo de su granularidad y la antigüedad de la organización.

El registro del Hecho posee una clave primaria que está compuesta por las claves primarias de las tablas de Dimensiones relacionadas a éste.



En la imagen anterior se puede apreciar un ejemplo de lo antes mencionado. La tabla de Hechos **factVentas** se ubica en el centro, e irradiando de ella se encuentran las tablas de Dimensiones **dimClientes**, **dimProductos** y **dimFechas**, que están conectadas mediante sus claves primarias. Es

por ello que la clave primaria de la tabla de Hechos es la combinación de las claves primarias de sus Dimensiones.

Los Hechos en este caso son `importeTotal` y `utilidad`.

Hechos vs Indicadores

En ocasiones se tiende a confundir los Hechos y los Indicadores. A continuación se expondrán las diferencias entre ellos:

- Los Hechos son aquellos datos que residen en una tabla de Hechos.
- Los Indicadores hacen uso de los Hechos para obtener un valor analizable y se definen mediante una serie de metadatos:
 - ▶ nombre representativo, descripción, I18N, etc;
 - ▶ tipo de agregación al momento de crear una estructura de datos (Cubo Multidimensional, Business Model). Las *agregaciones* más utilizadas son: SUM, MAX, MIN, COUNT, AVG, porcentajes, fórmulas, etc;
 - ▶ agregaciones alternativas;
 - ▶ tipo de datos (siempre numéricos).

Ejemplos de Hechos

Algunos ejemplos de Hechos y su constitución:

- `importeTotal = precioProducto * cantidadVendida`
- `rentabilidad = utilidad / patrimonioNeto`
- `cantidadVentas = cantidad`
- `promedioGeneral = AVG(notasFinales)`

A la izquierda de la igualdad se encuentran los Hechos y a la derecha los campos de los Data Sources. En el último ejemplo se realiza un cálculo de *agregación* para establecer el Hecho.

Hechos Básicos y Derivados

Existen dos tipos de Hechos, los Básicos y los Derivados, a continuación se detallará cada uno de ellos, teniendo en cuenta para su exemplificación la siguiente tabla de Hechos:



-
- Hechos Básicos: son aquellos que se encuentran representados por un campo de una tabla de Hechos. Los campos **precio** y **cantidad** de la tabla anterior son Hechos Básicos.
 - Hechos Derivados: son aquellos que se obtienen a partir de una expresión, en otras palabras, combinando uno o más Hechos con alguna operación matemática/lógica y que también residen en una tabla de Hechos. Tienen la ventaja de almacenarse ya calculados, por lo que serán accedidos a través de consultas SQL sencillas devolviendo resultados rápidamente. Por otro lado, requieren más espacio físico en el DW y más tiempo de proceso en los ETL. El campo **total** de la tabla anterior es un Hecho Derivado, y se conforma de la siguiente manera:
 - ▶ **total = precio * cantidad**

Agregadas y Preagregadas

Las tablas de Hechos Agregadas y Preagregadas se utilizan para almacenar un resumen de los datos, es decir, se guardan los datos en niveles de granularidad superior a los que inicialmente fueron obtenidos y/o gestionados.

Para generar tablas de Hechos Agregadas o Preagregadas, es necesario establecer un criterio con el cual realizar el resumen. Por ejemplo, esto ocurre cuando se desea obtener información de ventas sumarizadas por mes.

Cada vez que se requiere que los datos en una consulta se presenten en un nivel de granularidad superior al que se encuentran almacenados en el DW, se debe llevar a cabo un proceso de agregación.

El objetivo general de las tablas de Hechos Agregadas y Preagregadas es el mismo, pero cada una de ellas tiene una manera de operar diferente:

- Tablas de Hechos Agregadas: se generan luego de que se procesa la consulta correspondiente a la tabla de Hechos que se resumirá. En muchos casos, estos resúmenes son utilizados por las herramientas de software de análisis de forma automática a fin de mejorar la respuesta.
- Tablas de Hechos Preagregadas: se generan antes de que se procese la consulta correspondiente a la tabla de Hechos que se resumirá. De esta manera, la consulta se realiza contra una tabla que ya fue previamente agregada. Posee un nivel de granularidad menor al de la tabla de Hechos. Estos resúmenes deben generarse y almacenarse al momento de poblar/actualizar el DW, utilizando procesos ETL.

Más sobre las tablas de Hechos Preagregadas

Beneficios:

- Reduce la utilización de recursos de hardware en la que se incurre en el momento de calcular las agregaciones.
- Reduce el tiempo utilizado en la generación de consultas por parte de l@s usuari@s.
- Son muy útiles en los siguientes casos generales:
 - Cuando los datos a nivel detalle (menor nivel granular) son innecesarios y/o NO son requeridos.
 - Cuando una consulta a determinado nivel de granularidad es solicitada con mucha frecuencia.
 - Cuando el volumen de datos es muy grande y las consultas demoran en ser procesadas.

Desventajas:

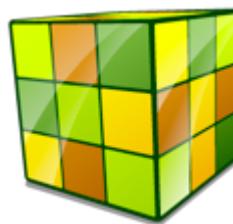
- Requieren de la creación y mantenimiento de nuevos procesos ETL.
- Requieren espacio de almacenamiento adicional en el DW.

- Resulta complejo reconocer qué agregaciones son necesarias mantener en tablas.
-

Cubo Multidimensional: introducción

El Cubo Multidimensional es la estructura de datos más utilizada y requerida, pero también es la más compleja de comprender.

Un Cubo Multidimensional, representa los datos planos (que se encuentran en filas y columnas), en una matriz de N Dimensiones.



Los componentes más importantes que se pueden incluir en un Cubo Multidimensional son:

- Indicadores,
- Atributos y
- Jerarquías.

Indicadores

- Los Indicadores son definiciones a partir de las cuales se obtendrán valores numéricicos mediante los cuales se analizará la situación de la organización.
- Los Indicadores pueden calcularse aplicando funciones de agregación sobre los Hechos o mediante expresiones complejas.
- El valor que tomen los Indicadores, dependerá de los Atributos/Jerarquías que se empleen para su análisis.

Atributos

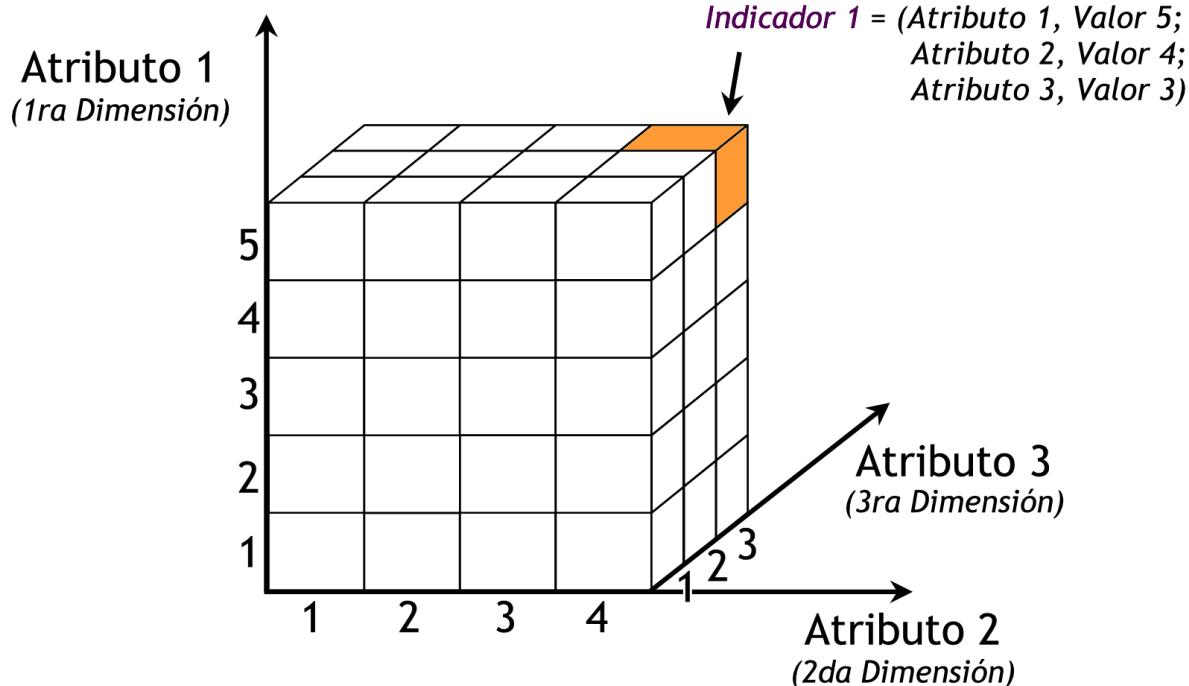
- Los Atributos son los criterios de análisis mediante los cuales analizaremos los Indicadores.
- El valor de los Atributos se obtiene de los campos de las tablas de Dimensión, aunque también pueden calcularse con expresiones complejas.

Jerarquías

- Una Jerarquía (en este contexto) es una relación lógica del tipo padre-hijo entre los Atributos.
- Al emplear Jerarquías podemos analizar los datos desde el nivel más general al más detallado y viceversa.
- Las Jerarquías manejan el nivel de agregación de los Hechos.

Representación matricial

En un Cubo Multidimensional, los Atributos existen a lo largo de varios ejes o Dimensiones, y de su intersección dependerá el valor del Indicador que se está evaluando.



El Cubo de la figura está compuesto de tres Dimensiones y un Indicador (**Indicador 1**). Cada una de estas Dimensiones posee solo un Atributo, es decir:

- 1ra Dimensión posee el **Atributo 1**,
- 2da Dimensión posee el **Atributo 2**, y
- 3ra Dimensión posee el **Atributo 3**.

En la imagen anterior se ha seleccionado una intersección para demostrar la correspondencia con los valores de los Atributos. En este caso, el **Indicador 1**, representa el cruce de:

- El valor **5** de **Atributo 1**,
- el valor **4** de **Atributo 2** y
- el valor **3** de **Atributo 3**.

El resultado del análisis está dado por los cruces matriciales de acuerdo a los valores de las Dimensiones seleccionadas.

Modelos del DW

A continuación se detallarán los tres tipos de estructuras de modelo del DW:

-  Esquema en Estrella (Star Scheme).
-  Esquema Copo de Nieve (Snowflake Scheme).
-  Esquema Constelación (Starflake Scheme).

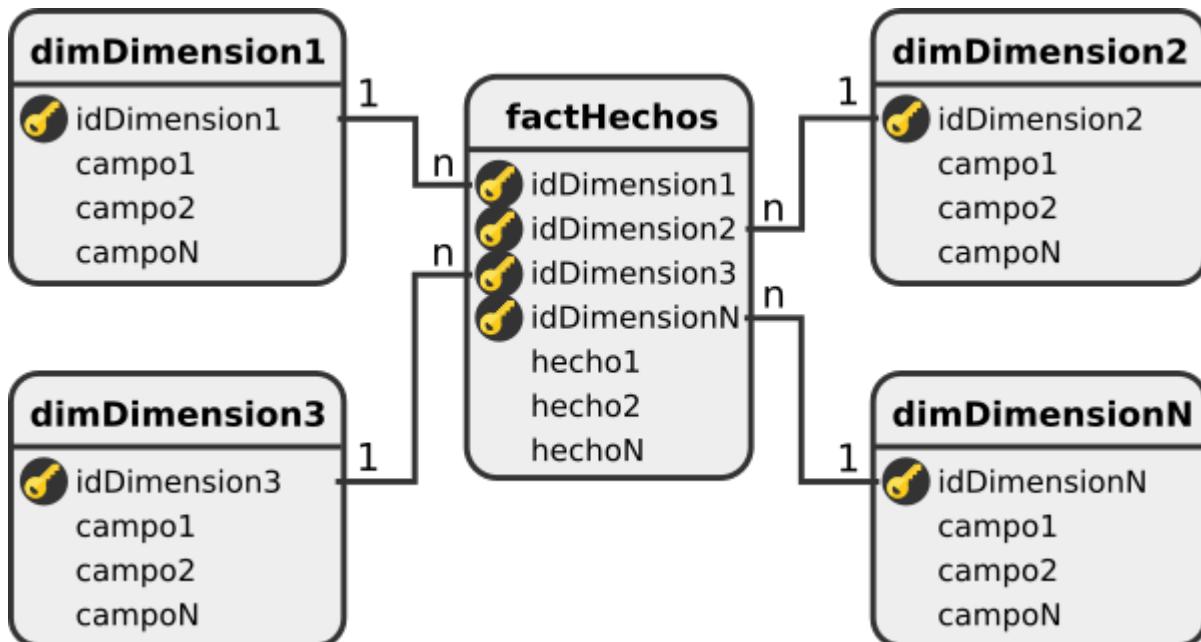
Estrella



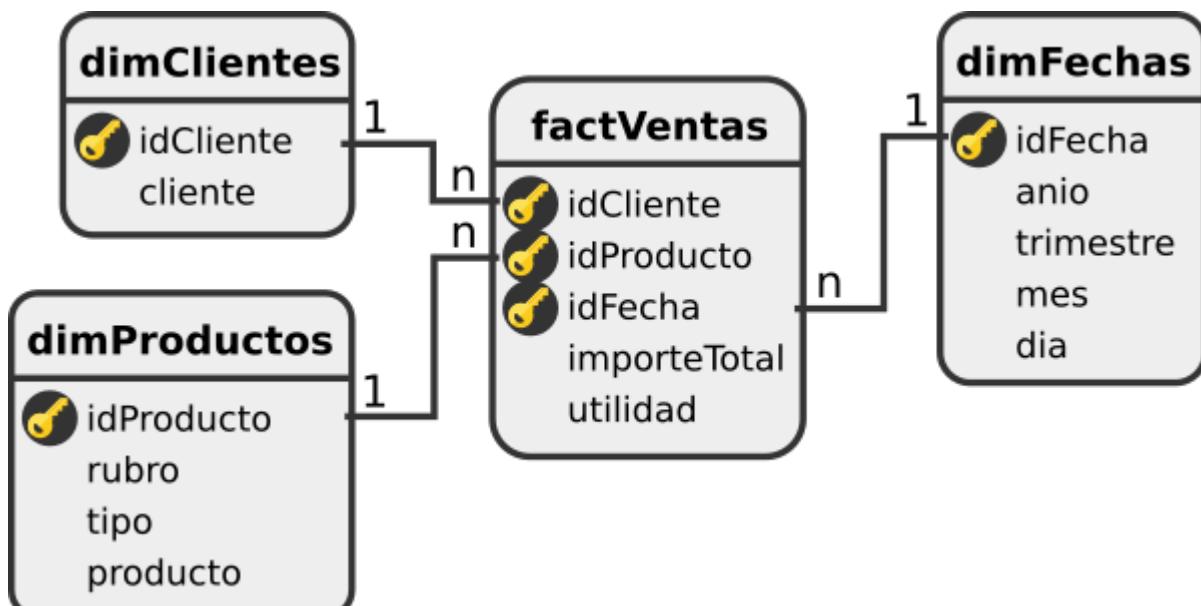
El Esquema en Estrella (Star Scheme) está formado por:

- una tabla de Hechos y
- una o más tablas de Dimensiones relacionadas a través de sus respectivas claves.

La siguiente figura representa un Esquema en Estrella estándar:



El modelo utilizado cuando se abordó el tema de las tablas de Hechos, es un Esquema en Estrella, por lo cual se lo volverá a mencionar para explicar sus cualidades:

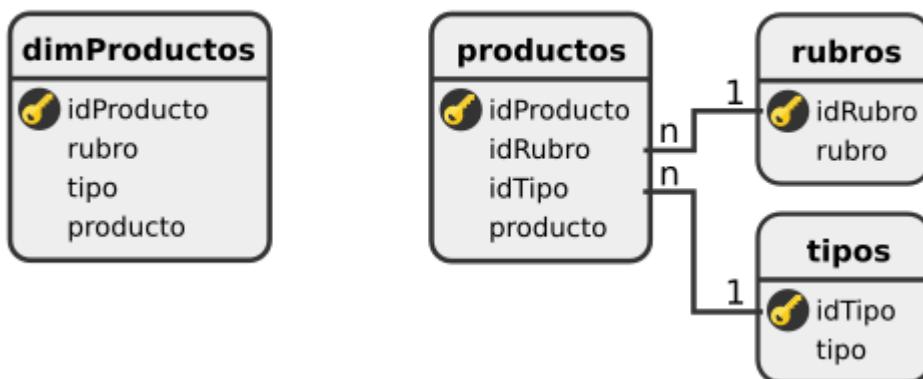


Las tablas de Dimensiones de este modelo, se encuentran desnormalizadas, es decir que no se presentan en tercera forma normal (3ra FN).

En la siguiente imagen se presentan dos modelos:

- **Normalizado:** modelo de la derecha, conformado por tablas relacionales estándar con sus respectivas uniones.
- **Desnormalizado:** modelo de la izquierda, conformado tras desnormalizar el modelo de la derecha.

Desnormalizado Normalizado



Al realizar la desnormalización sobre el modelo de la derecha:

- la tabla de Dimensión **dimProductos** elimina las uniones y las claves y
- mantiene solo los datos descriptivos: **rubro**, **tipo** y **producto**.

Cuando se normaliza se pretende eliminar la redundancia, es decir, la repetición de datos y las dependencias funcionales entre los datos, los modelos multidimensionales requieren precisamente lo contrario.

Entonces, la principal ventaja de la desnormalización es:

- evitar uniones (JOIN) entre las tablas cuando se realizan consultas, procurando así un mejor tiempo de respuesta y una mayor sencillez con respecto a su utilización.

Y la desventaja de la normalización es:

- redundancia y
- consumo adicional de espacio de almacenamiento.

Características del Esquema en Estrella

A continuación se destacarán las principales características del Esquema en Estrella:

- Es el más simple de interpretar .
- Posee los mejores tiempos de respuesta.
- Es soportado por todos los visores OLAP.
- Su diseño es sencillo de mantener y actualizar.

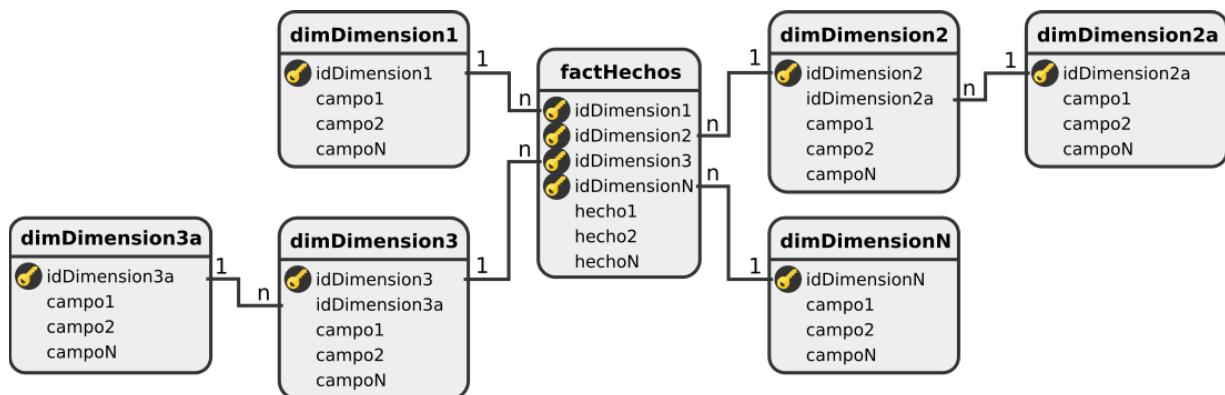
- Existe paralelismo entre su diseño y la forma en que l@s usuari@s visualizan y manipulan los datos.
- Es el modelo elegido para prototipado rápido.

Copo de Nieve



El Esquema Copo de Nieve (Snowflake Scheme) es:

- una extensión del modelo en Estrella, y
- se caracteriza por poseer tablas de Dimensiones organizadas en Jerarquías de Dimensiones.



Como se puede apreciar en la figura anterior:

- existe una tabla de Hechos (**factHechos**) que está relacionada con una o más tablas de Dimensiones, y
- algunas tablas de Dimensiones están relacionadas con otras tablas de Dimensiones.

Este modelo se parece más al modelo transaccional ya que algunas tablas de Dimensiones están normalizadas.

Características del Esquema Copo de Nieve

A continuación se destacarán las principales características del Esquema Copo de Nieve:

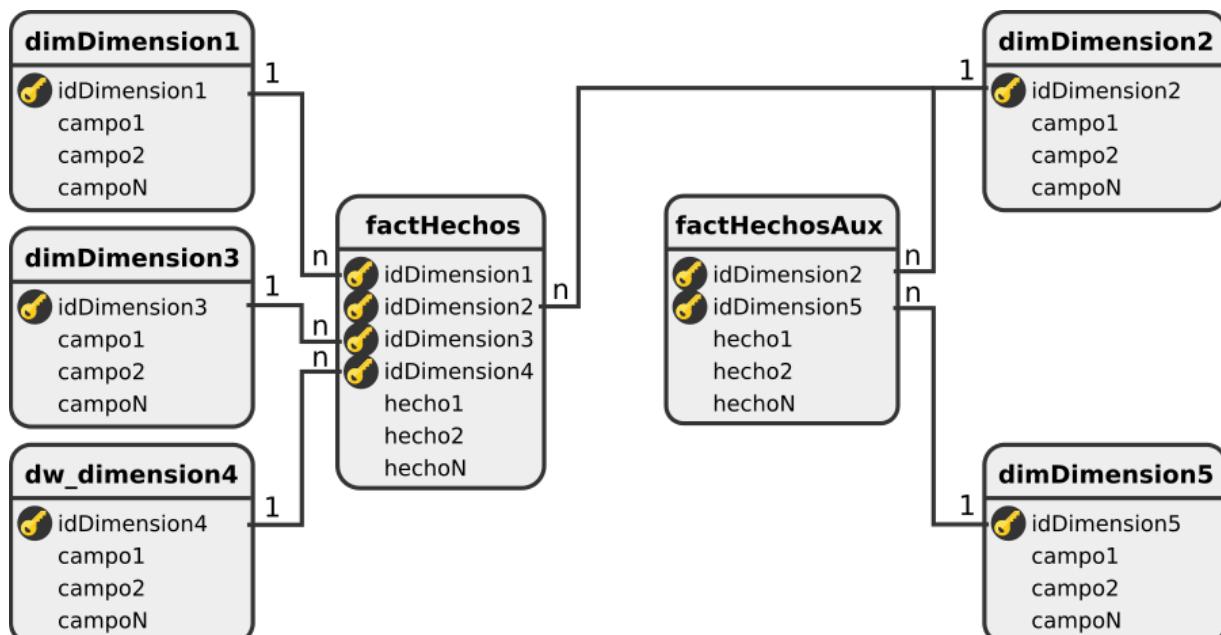
- Posibilita la segregación de los datos de las tablas de Dimensiones.
- Puede implementarse después de que se haya desarrollado un Esquema en Estrella.
- Posee mayor complejidad en su estructura.
- Utiliza menos espacio de almacenamiento.
- Es más eficiente en el caso de tablas de Dimensiones con gran cantidad de registros.
- Su semántica se ajusta a las representaciones de las diferentes Jerarquías de Dimensiones.
- Se deben planificar correctamente las uniones e el indexado, a fin de NO generar sobrecarga en la resolución de consultas.

Constelación



El Esquema en Constelación (Starflake Scheme):

- está compuesto por una serie de Esquemas en Estrella,
- posee una tabla de Hechos Principal (por ejemplo: **factHechos**) y
- una o más tablas de Hechos Auxiliares (por ejemplo: **factHechosAux**), las cuales pueden ser agregaciones de la Principal. Dichas tablas están relacionadas con sus respectivas tablas de Dimensiones.



NO es necesario que las diferentes tablas de Hechos compartan las mismas tablas de Dimensiones. Las tablas de Hechos Auxiliares pueden vincularse con solo algunas de las tablas de Dimensiones asignadas a la tabla de Hechos Principal, y también pueden hacerlo con nuevas tablas de Dimensiones.

Características del Esquema en Constelación

Las características y diseño del Esquema en Constelación son muy similares a las del Esquema en Estrella, con las siguientes diferencias:

- Permite tener más de una tabla de Hechos, por lo cual se tendrá mayor capacidad analítica.
- Contribuye a la reutilización de las tablas de Dimensiones, ya que una misma tabla de Dimensión puede utilizarse para varias tablas de Hechos.

OLTP vs DW

Los OLTP (sistemas transaccionales) se diseñan para dar soporte al procesamiento diario de datos de las organizaciones, es decir, gran cantidad de transacciones en cortos períodos de tiempo. Se enfatiza la maximización de la capacidad transaccional y la concurrencia. Las estructuras de las bases de datos se encuentran altamente normalizadas, de esta forma, se aumenta la eficiencia de las escrituras de datos. En cuanto a las lecturas o consultas, son mínimas y acceden a un pequeño número de registros. Estos sistemas están fuertemente condicionados por los procesos operacionales que deben soportar, administrar datos y muy poca información. A diferencia de los OLTP, los Data Warehouses están diseñados para llevar a cabo procesos de consulta y análisis complejos, que serán la base para la toma de decisiones estratégica y táctica de alto nivel.

A continuación una tabla comparativa entre los dos ambientes, que resume sus principales diferencias:

	OLTP	Data Warehouse
Objetivo	Soportar actividades transaccionales diarias	Consultar y analizar información táctica y estratégica
Tipo de datos	Operacionales	Para la toma de decisiones
Modelo de datos	Normalizado	Desnormalizado
Consulta	SQL	SQL más extensiones
Datos consultados	Actuales	Actuales e históricos
Horizonte de tiempo	60 - 90 días	5 - 10 años
Tipos de consultas	Repetitivas, predefinidas	NO previsibles, dinámicas
Nivel de almacenamiento	Nivel de detalle	Nivel de detalle y diferentes niveles de agregación
Acciones disponibles	Alta, baja, modificación y consulta	Carga y consulta
Número de transacciones	Alto	Medio - Bajo
Tamaño	Pequeño - Mediano	Grande
Tiempo de respuesta	Pequeño (segundos - minutos)	Variable (minutos - horas)
Orientación	Orientado a las aplicaciones	Orientado al negocio
Estructura	Estable	Varía de acuerdo a su propia evolución y utilización

Implementación

Veremos a continuación los tres tipos de implementación del modelo multidimensional:

-  Relacional – ROLAP.
-  Multidimensional – MOLAP.
-  Híbrido – HOLAP.

ROLAP



En los sistemas ROLAP (Relational On Line Analytic Processing), los Cubos Multidimensionales se generan en el momento en que se realizan las consultas.

Este proceso se puede resumir a través de los siguientes pasos:

1. Se describe la metadata del Cubo: Indicadores, Atributos, Jerarquías, etc.
2. Se almacena la metadata.
3. El Motor Multidimensional del visor OLAP que se esté utilizando, carga la metadata con la cual realizará un mapeo entre los datos del DW y los Atributos, Indicadores, etc.
4. Cada vez que se actualiza el DW, se debe borrar la caché del Motor Multidimensional a fin de visualizar los nuevos datos. Esto se debe a que los motores ROLAP hacen uso exhaustivo del caché, lo cual permite que el motor evite consultar dos veces el mismo dato, ya que una vez consultado será almacenado en caché.

MOLAP

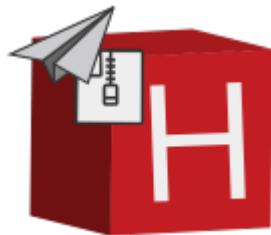


MOLAP (Multidimensional On Line Analytic Processing) precalcula los Cubos Multidimensionales y los almacena físicamente.

Este proceso se puede resumir a través de los siguientes pasos:

1. Se seleccionan los Indicadores, Atributos, Jerarquías, etc., que compondrán el Cubo Multidimensional.
2. Se precisan los datos del Cubo, es decir, todas las combinaciones posibles entre los niveles de las Jerarquías de cada Dimensión.
3. Se ejecutan las consultas sobre los datos precalculados del Cubo.
4. Cada vez que se actualiza el DW, se debe precalcular y guardar el Cubo, para que contenga los nuevos datos.

HOLAP



HOLAP (Hybrid On Line Analytic Processing) constituye un sistema híbrido entre MOLAP y ROLAP, que combina estas dos implementaciones.

Los datos agregados y precalculados se almacenan en estructuras multidimensionales y los de menor nivel de detalle en estructuras relacionalles. Es decir:

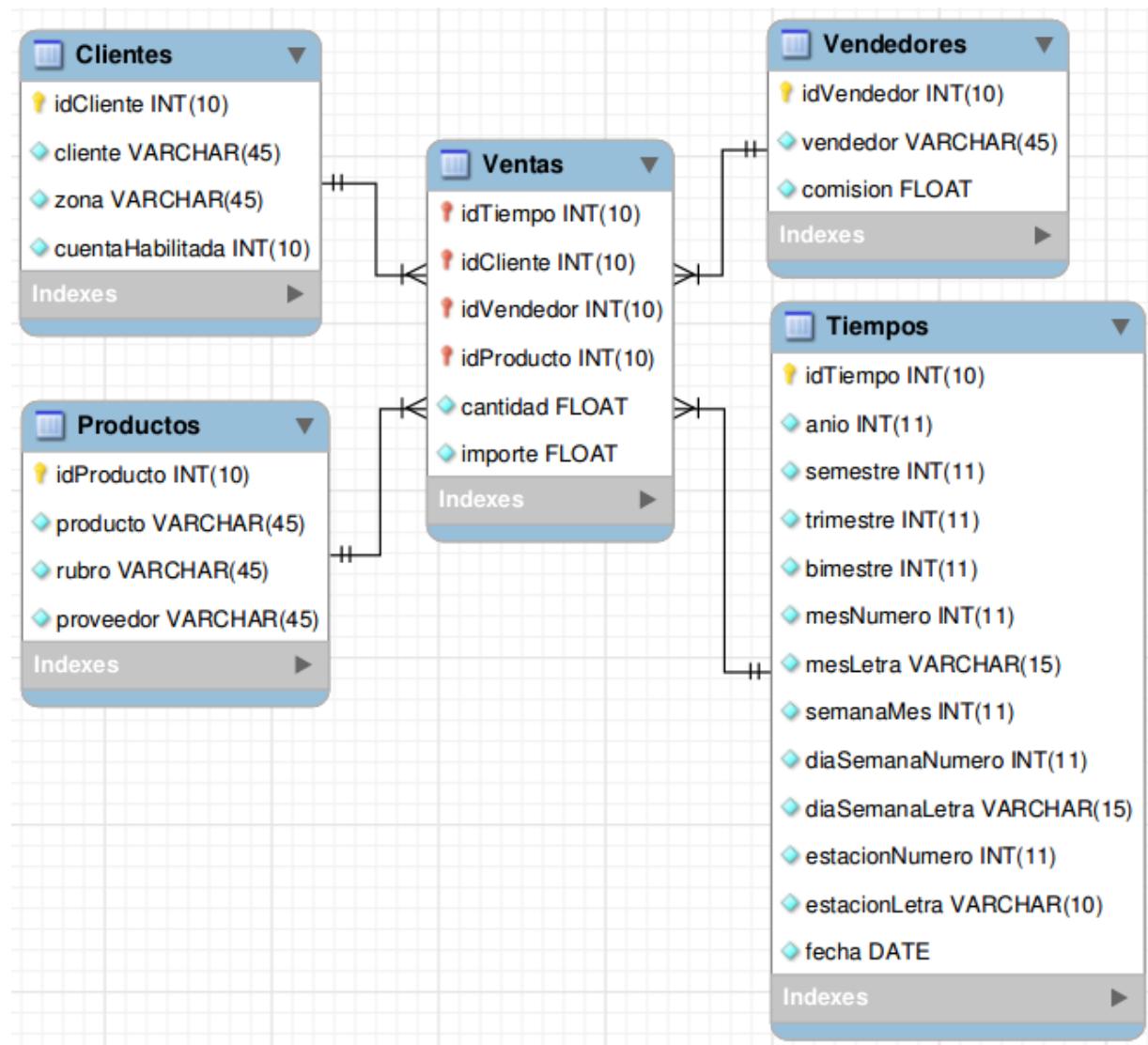
- se utilizará ROLAP para navegar y explorar los datos a bajos niveles de granularidad y
- se utilizará MOLAP para la explotación de datos precalculados, por lo general sumatorias o funciones de alto nivel de agregación, suelen ser los más utilizados en los dashboards.

Cubo Multidimensional: profundización

Con una idea clara acerca de las formas de modelar e implementar un DW, se detallará paso a paso el proceso de construcción de Cubos Multidimensionales.

Se utilizará para el caso una representación genérica, para que luego sea sencillo trasladar los conocimientos aquí adquiridos al momento de trabajar con un software de creación de Cubos Multidimensionales.

El siguiente Esquema en Estrella, constituye la base sobre la cual se exemplificará el desarrollo del Cubo:

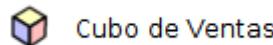


El Esquema en Estrella está compuesto por:

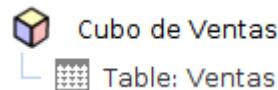
- una tabla de Hechos: **Ventas**.
- cuatro tablas de Dimensiones: **Clientes**, **Productos**, **Vendedores** y **Tiempos**.

Pasos básicos

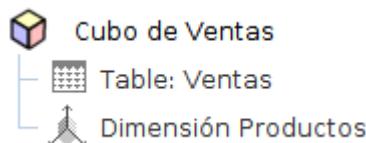
1) Se creará un Cubo Multidimensional llamado **Cubo de Ventas**:



2) Se indicará que la tabla de Hechos a utilizar es **Ventas**:



3) Se creará una Dimensión para analizar los productos. Su nombre será **Dimensión Productos**:



3.1) Se añadirá a **Dimensión Productos** una Jerarquía. Su nombre será **Jerarquía Productos**:



3.2) Se indicará que **Jerarquía Productos** estará basada en los campos de la tabla de **Dimensión Productos**:



3.3) Se añadirá a la **Jerarquía Productos** el Atributo **Producto**, que estará basado en el campo **producto**:



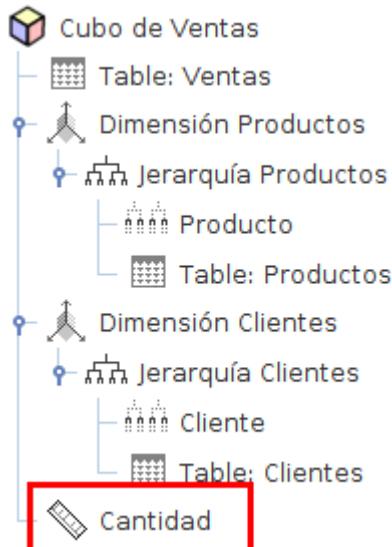
4) De forma similar que se hizo en el punto anterior, se creará la **Dimensión Clientes**:





5) Se creará un Indicador, llamado **Cantidad** que se calculará de la siguiente manera:

- SUM(cantidad)**



En este momento, se ha construido un Cubo Multidimensional de dos Dimensiones y un Indicador.

Si se analiza el Indicador **Cantidad** utilizando los Atributos **Producto** y **Cliente**, la representación matricial será la siguiente:

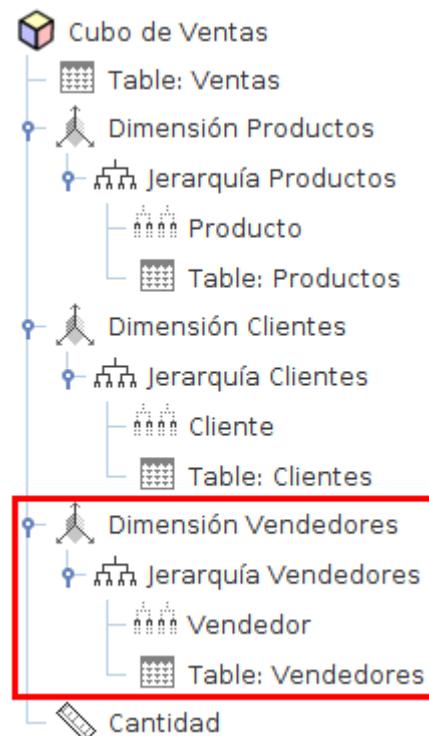
Producto 1	40	25	60
Producto 2	21	55	45
Producto 3	13	32	43
	Cliente 1	Cliente 2	Cliente 3

La intersección de las Dimensiones representa la cantidad vendida de cada producto a cada cliente.

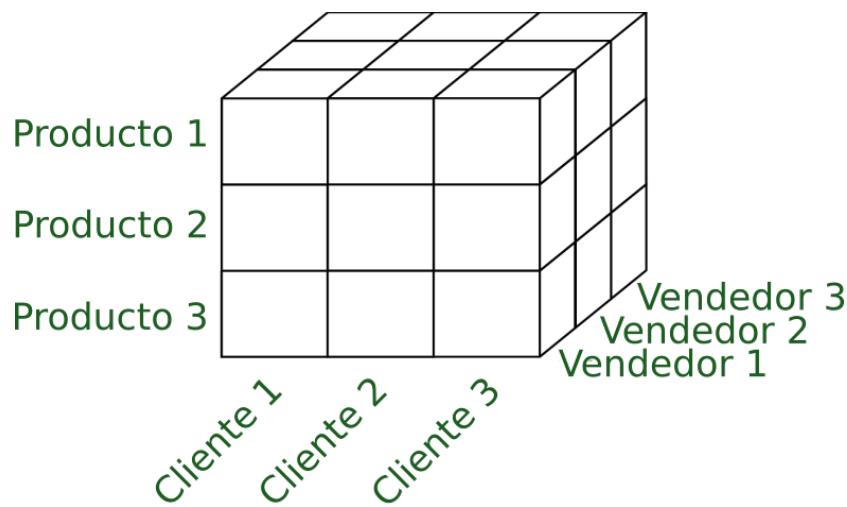
Por ejemplo:

- La cantidad vendida del **Producto 1** al **Cliente 1** es de **40** unidades.
- La cantidad vendida del **Producto 1** al **Cliente 2** es de **25** unidades.
- La cantidad vendida del **Producto 1** al **Cliente 3** es de **60** unidades.

6) De forma similar que se hizo en el punto anterior, se creará la **Dimensión Vendedores**:

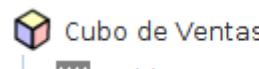


En este momento, se ha construido un Cubo Multidimensional de tres Dimensiones y un Indicador.



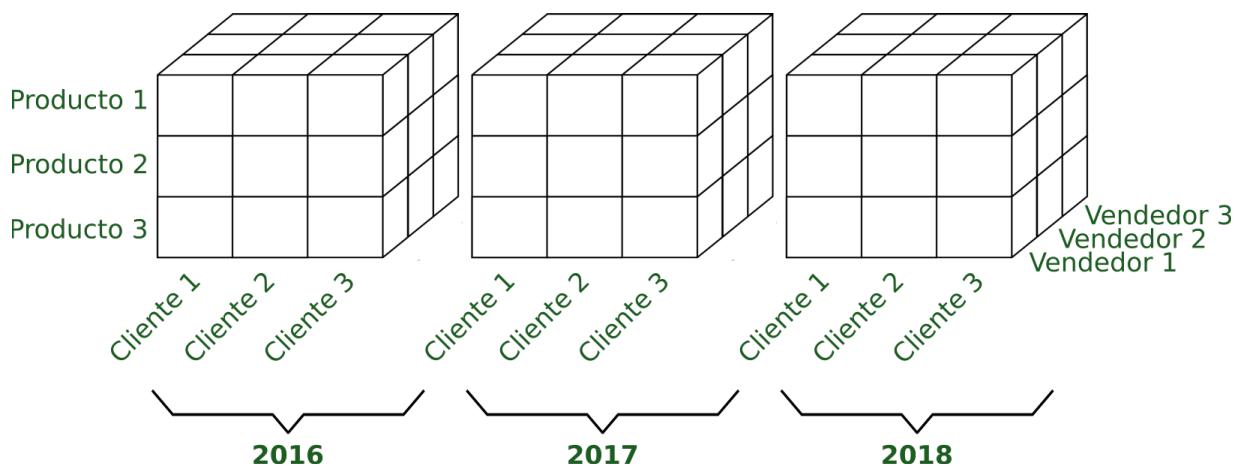
En este caso los valores del Indicador **Cantidad** están definidos por la intersección de tres Dimensiones. Se pueden medir las cantidades vendidas por **Producto**, **Cliente** y **Vendedor**.

7) De forma similar que se hizo en el punto anterior, se creará la **Dimensión Años**:





En este momento, se ha construido un Cubo Multidimensional de cuatro Dimensiones y un Indicador.



En esta versión del Cubo, la lectura del Indicador **Cantidad**, está condicionada por las cantidades vendidas de cada **Producto**, a cada **Cliente**, de cada **Vendedor**, en cada **Año**.

La última imagen expresa de forma muy clara los conceptos expuestos anteriormente sobre la Dimensión Tiempo, en donde se establecía que pueden existir diferentes versiones de la situación del negocio.

Cabe aclarar que pueden crearse tantos Cubos como sean necesarios sin que su coexistencia implique inconveniente alguno.

Jerarquías

En el Cubo Multidimensional que se construyó anteriormente creamos Dimensiones y dentro de ellas añadimos Jerarquías; y en cada Jerarquía definimos un solo Atributo.

Tenemos la posibilidad de añadir más Atributos dentro de las Jerarquías, en donde los Atributos consecutivos posean una relación de padre-hijo (1-n) entre el Atributo superior y el inferior.

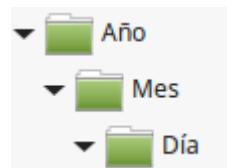
Veamos la siguiente Jerarquía:



Dentro de la **Jerarquía Años** definimos tres Atributos **Año**, **Mes** y **Día**.

El Atributo más general se posiciona en la cabecera y los más particulares hacia abajo:

- Un *año* posee uno o más *meses del año*; y un *mes del año* pertenece solo a un *año*. En otras palabras, el *año 2017* posee muchos meses; y un *mes del 2017* solo pertenece al *año 2017*
- Un *mes del año* posee uno o más *días del año*; y un *día del año* pertenece solo a un *mes de año*. En otras palabras, *Enero de 2017* posee muchos días; y un *día de Enero de 2017* solo pertenece a *Enero de 2017*.



También tenemos la posibilidad de añadir más Jerarquías dentro de una Dimensión. Por ejemplo:



El Atributo Estación presenta las estaciones del año: **Otoño, Invierno, Primavera, Verano**.

Relación jerárquica

Una relación representa la forma en que dos Atributos interactúan dentro de una Jerarquía. Existen básicamente dos tipos de relaciones:

- Explícitas: son las más comunes y se pueden modelar a partir de Atributos directos y están en línea continua de una Jerarquía, por ejemplo, un país posee una o más provincias y una provincia pertenece solo a un país.
- Implícitas: su relación NO es de vista directa, por ejemplo, una provincia tiene uno o más ríos, pero un río pertenece a una o más provincias. Se trata de una *relación muchos a muchos*.

Metadatos



Los metadatos son datos sobre los datos, sirven para describir otros datos, que en este caso, existen en la arquitectura del Data Warehousing.

Brindan principalmente información de localización, estructura y significado de los datos, es decir, mapean a los mismos.

El concepto de metadatos es análogo al uso de índices para localizar objetos en lugar de datos.

Los metadatos NO son exclusivos del DWH, forman parte de las bases de datos transaccionales y de la mayoría de las estructuras de almacenamiento. En el caso del Data Warehousing los metadatos son gestionados y mantenidos por l@s usuari@s, pueden ser modificados, exportados, importados; existe una gran interacción entre usuari@s y metadatos, sea esta interacción manual o automática.

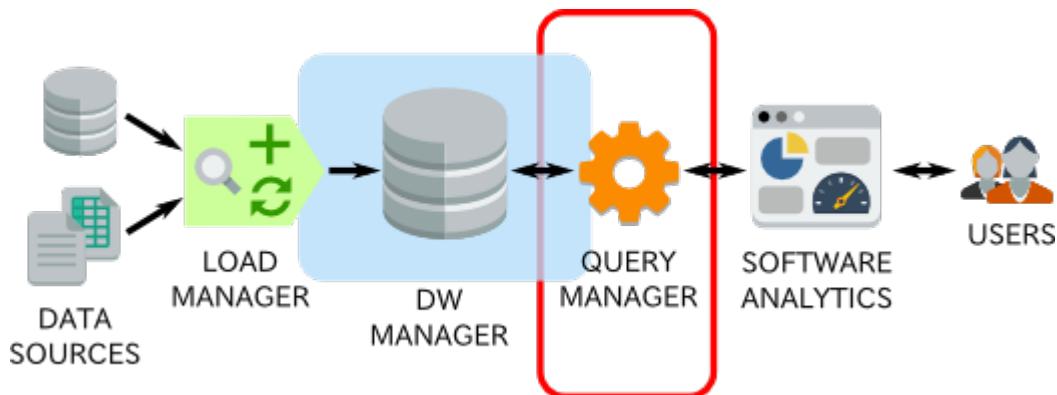
Algunos ejemplos de metadatos en el ambiente DWH son:

- metadatos utilizados por los procesos ETL,
- metadatos de conexión con bases de datos,
- metadatos de estructuras de datos (Cubos Multidimensionales, Business Models, etc.),
- metadatos utilizados por los software analíticos.

Mapping

El término mapping, se refiere a relacionar un conjunto de objetos, tal como actualmente están almacenados en memoria o en disco, con otros objetos. Por ejemplo: una estructura de base de datos lógica, se proyecta sobre la base de datos física. El mapeo se estructura como metadatos.

4) Query Manager



El Query Manager es una pieza fundamental y compleja del proceso de DWH, pues es el encargado de realizar las operaciones necesarias para soportar los procesos de gestión y ejecución de:

- consultas relacionales: como JOIN y agregaciones (SUM, COUNT, AVG, etc), y
- consultas propias del análisis de datos: como DRILL-UP y DRILL-DOWN.

El funcionamiento del Query Manager es el siguiente:

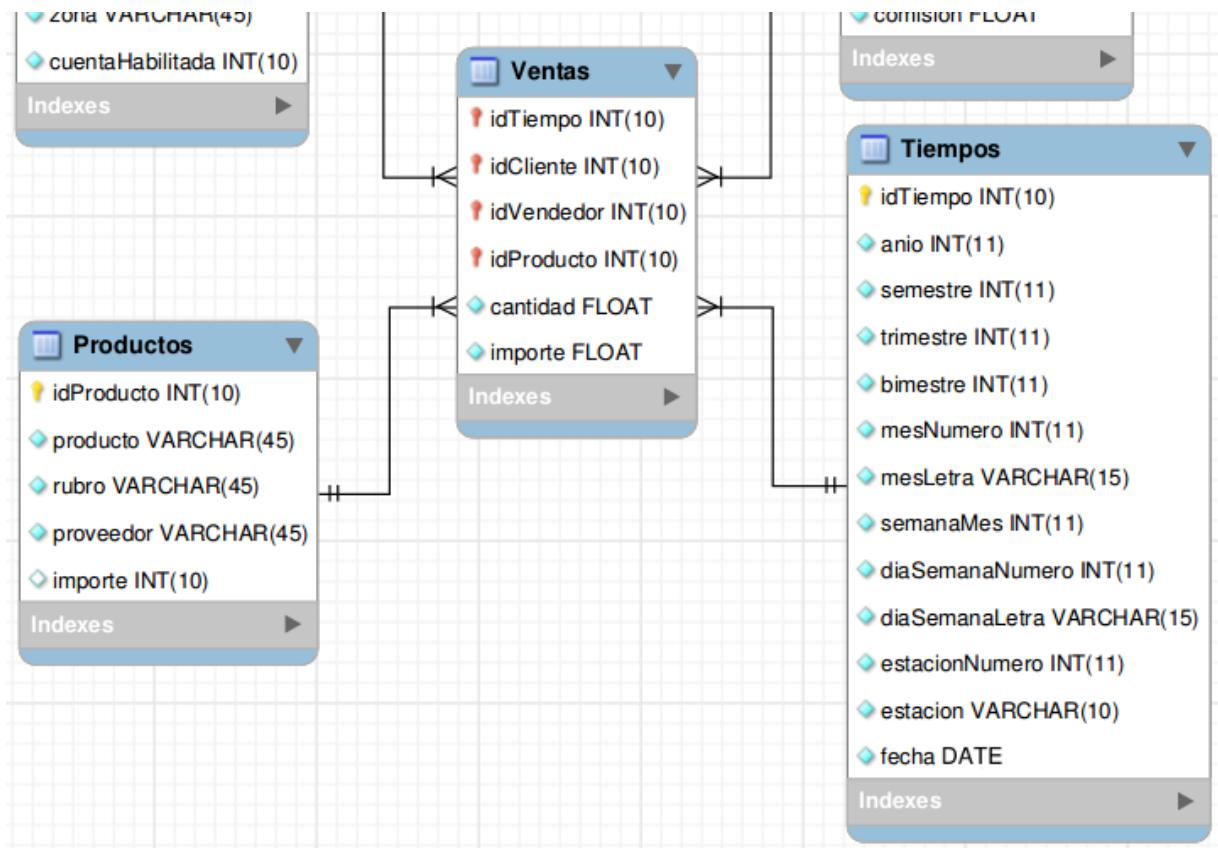
1. recibe consultas de l@s usuari@s, que en general están escritas en un lenguaje de alto nivel (por ejemplo MDX);
2. lee los metadatos que describen los mapeos (Cubo Multidimensional, Business Models, etc.) y reescribe las consultas para que sean ejecutadas en el sistema destino (por lo general SQL);
3. una vez que obtiene los datos y utilizando, nuevamente, las estructuras de metadatos, éstos son transformados a un formato final de alto nivel que será interpretado y renderizado por las herramientas de visualización.

Las principales operaciones que se pueden realizar sobre modelos multidimensionales son:

- Drill-down
- Drill-up
- Drill-across
- Roll-across
- Pivot
- Page
- Drill-through

A continuación, se explicará cada una de ellas y se ejemplificará su utilización, para lo cual se utilizará como guía el siguiente Esquema en Estrella:





El modelo consta de cuatro tablas de Dimensiones y una tabla de Hechos que se encuentra en el centro.

Se volverá a utilizar el Cubo Multidimensional creado anteriormente:



Para simplificar los ejemplos que se presentarán, se utilizará solo una pequeña muestra de datos.

Drill-down

Drill-down es una operación que permite apreciar los datos con un mayor nivel de detalle. Se aplica bajando por los niveles de una Jerarquía definida en un Cubo. Esto brinda la posibilidad de introducir un nuevo nivel o criterio de agregación en el análisis.

Drill-down implica ir de lo general a lo específico.



Para explicar y ejemplificar esta operación se utilizará la siguiente representación tabular:

Dimensión Productos Jerarquía Productos	Dimensión Años Jerarquía Años	Indicador
Producto	Año	Cantidad
Producto 1	2017	40
Producto 2	2017	52

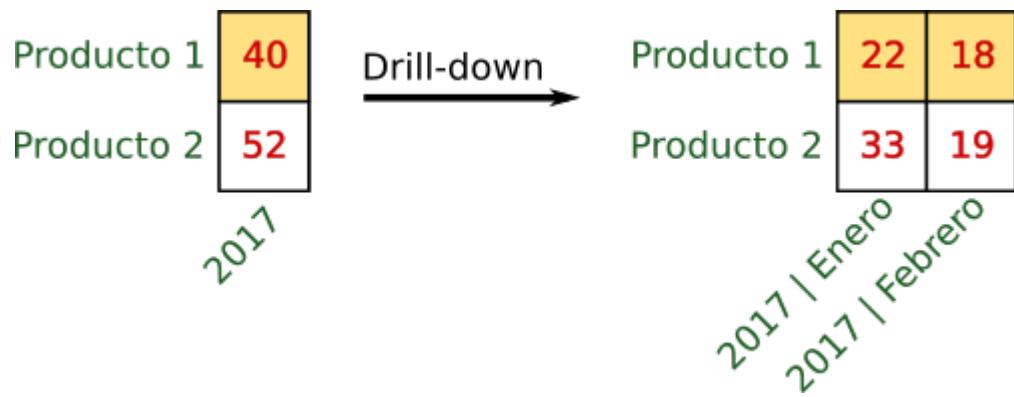
Como puede apreciarse, en la cabecera de la tabla se encuentran los Atributos y el Indicador (destacado con color de fondo diferente) definidos anteriormente en el Cubo Multidimensional; y en el cuerpo se encuentran los valores correspondientes a cada cruce. Se ha resaltado la primera fila, ya que luego se la analizará en detalle.

Se aplicará la operación drill-down sobre **Jerarquía Años**, añadiendo un nivel más de detalle. En este caso, al bajar por la **Jerarquía Años** se añade el Atributo **Mes**.

Dimensión Productos Jerarquía Productos	Dimensión Años Jerarquía Años	Indicador
Producto	Año Mes	Cantidad
Producto 1	2017 Enero	22
Producto 1	2017 Febrero	18
Producto 2	2017 Enero	33
Producto 2	2017 Febrero	19

Como puede apreciarse en los ítems resaltados de la tabla, se agregó un nuevo nivel de detalle (**Año | Mes**) a la lista inicial, y el valor **40** que pertenecía a las ventas del **Producto1**, en el año **2017**, se dividió en dos filas. Esto se debe a que ahora se tendrá en cuenta el Atributo **Mes** para realizar la agregación del Indicador **Cantidad**.

La siguiente imagen muestra este mismo proceso, representado matricialmente:



En adelante se utilizará esta forma para explicar cada operación.

Drill-up

Drill-up es una operación que permite apreciar los datos con un menor nivel de detalle, subiendo por una Jerarquía definida en un Cubo. Esto brinda la posibilidad de quitar un nivel o criterio de agregación en el análisis.

Drill-up implica ir de lo específico a lo general.



Siguiendo con el ejemplo de Drill-down, se tomarán como referencia los resultados anteriores:

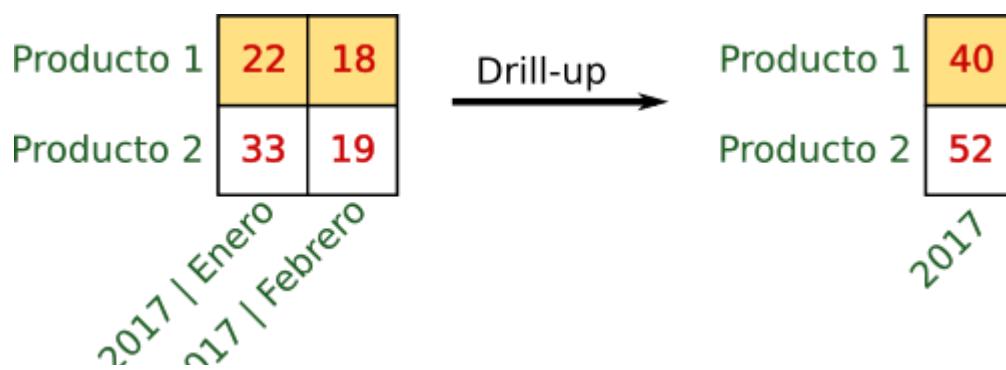
Dimensión Productos Jerarquía Productos	Dimensión Años Jerarquía Años	Indicador
Producto	Año Mes	Cantidad
Producto 1	2017 Enero	22
Producto 1	2017 Febrero	18
Producto 2	2017 Enero	33
Producto 2	2017 Febrero	19

Se aplicará drill-up sobre la **Jerarquía Años**, entonces:

Dimensión Productos Jerarquía Productos	Dimensión Años Jerarquía Años	Indicador
Producto	Año	Cantidad
Producto 1	2017	40
Producto 2	2017	52

Como se puede apreciar, en la fila resaltada se sumarizaron los valores **22** y **18** de la tabla inicial, debido a que al eliminar el Atributo **Mes**, las ventas se agregaron de acuerdo a **Producto** y **Año**.

La siguiente imagen muestra este mismo proceso, representado matricialmente:



v ~

Drill-across

Drill-across es una operación que trabaja de forma similar a drill-down, con la diferencia de que drill-across NO se aplica sobre una Jerarquía, sino que su forma de ir de lo general a lo específico es agregar un Atributo a la consulta como nuevo criterio de análisis.

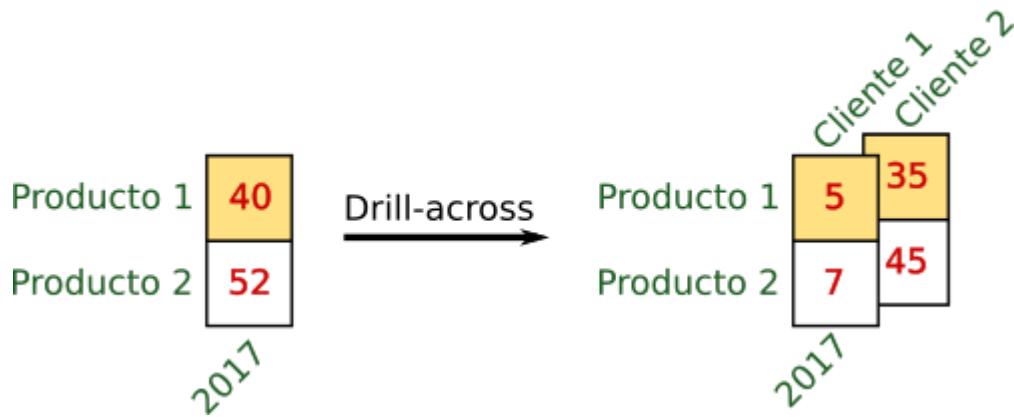
En el siguiente ejemplo, se partirá de los siguientes resultados:

Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año	Indicador Cantidad
Producto 1	2017	40
Producto 2	2017	52

Ahora, se aplicará drill-across, al agregar el Atributo **Cliente**. Entonces:

Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año	Dimensión Clientes Jerarquía Clientes Cliente	Indicador Cantidad
Producto 1	2017	Cliente 1	5
Producto 1	2017	Cliente 2	35
Producto 2	2017	Cliente 1	7
Producto 2	2017	Cliente 2	45

La siguiente imagen muestra este mismo proceso, representado matricialmente:



Roll-across

Roll-across es una operación que trabaja de forma similar a drill-up, con la diferencia de que roll-across NO se hace sobre una Jerarquía, sino que su forma de ir de lo específico a lo general es quitar un Atributo de la consulta, eliminando de esta manera un criterio de análisis.

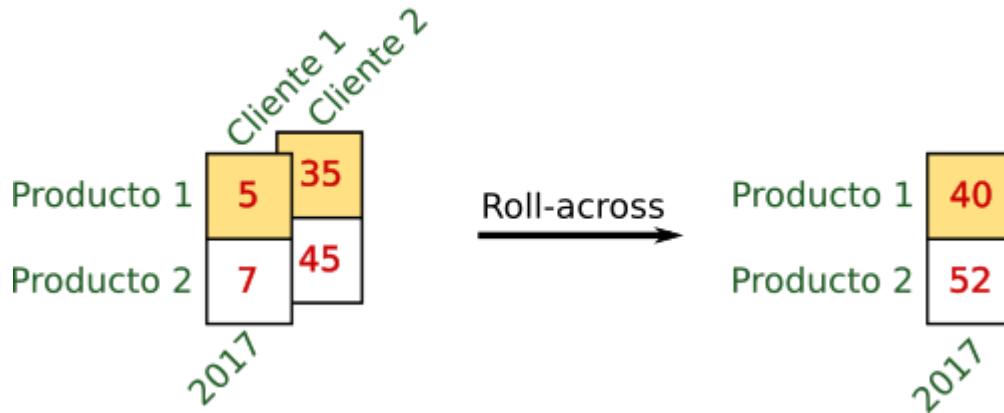
Para mostrar su funcionamiento, se tomará como base la representación tabular anterior:

Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año	Dimensión Clientes Jerarquía Clientes Cliente	Indicador Cantidad
Producto 1	2017	Cliente 1	5
Producto 1	2017	Cliente 2	35
Producto 2	2017	Cliente 1	7
Producto 2	2017	Cliente 2	45

Se aplicará la operación roll-across, quitando de la consulta el Atributo Cliente, entonces:

Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año	Indicador Cantidad
Producto 1	2017	40
Producto 2	2017	52

La siguiente imagen muestra este mismo proceso, representado matricialmente:



Pivot

Pivot es una operación que permite seleccionar el orden de visualización de los Atributos e Indicadores, con el objetivo de analizar la información desde diferentes puntos de vista.

Se tomará como referencia, para explicar esta operación, la siguiente tabla:

Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año	Dimensión Clientes Jerarquía Clientes Cliente	Indicador Cantidad
Producto 1	2017	Cliente 1	5
Producto 1	2017	Cliente 2	35
Producto 2	2017	Cliente 1	7
Producto 2	2017	Cliente 2	45

Como puede apreciarse, el orden de los Atributos es: **Producto, Año y Cliente**.

Ahora, se hará Pivot, reorientando la vista multidimensional:

Dimensión Productos Jerarquía Clientes Cliente	Dimensión Años Jerarquía Años Año	Dimensión Clientes Jerarquía Productos Producto	Indicador Cantidad
Cliente 1	2017	Producto 1	5
Cliente 1	2017	Producto 2	7
Cliente 2	2017	Producto 1	35
Cliente 2	2017	Producto 2	45

El nuevo orden de los Atributos es: **Cliente, Año y Producto**.

Pivot permite realizar las siguientes acciones:

- Mover un Atributo o Indicador desde el encabezado de Fila al encabezado de Columna.
- Mover un Atributo o Indicador desde el encabezado de Columna al encabezado de Fila.
- Cambiar el orden de los Atributos o Indicadores del encabezado de Columna.
- Cambiar el orden de los Atributos o Indicadores del encabezado de Fila.

Page

Page es una operación que presenta el Cubo dividido en secciones, a través de los valores de un Atributo, como si se tratase de páginas de un libro.

Page es muy útil cuando las consultas devuelven muchos registros y es necesario desplazarse por los datos para poder verlos en su totalidad.

Se tomará como referencia, para explicar esta operación, la siguiente tabla:

Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año Mes	Indicador Cantidad
Producto 1	2017 Enero	22
Producto 1	2017 Febrero	18
Producto 2	2017 Enero	33
Producto 2	2017 Febrero	19

Se realizará Page sobre el Atributo **Producto**, entonces se obtendrán las siguientes páginas:

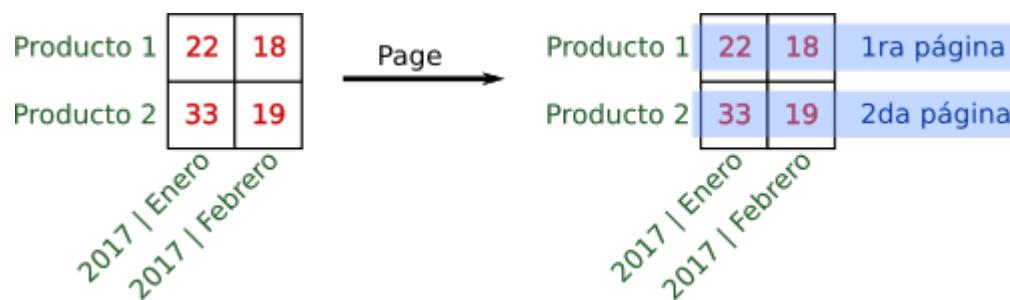
- 1ra página, correspondiente a **Producto 1**:

Dimensión Años Jerarquía Años Año Mes	Indicador Cantidad
2017 Enero	22
2017 Febrero	18

- 2da página, correspondiente a **Producto 2**:

Dimensión Años Jerarquía Años Año Mes	Indicador Cantidad
2017 Enero	33
2017 Febrero	19

Matricialmente se representa de la siguiente manera:



Drill-through

Drill-through es una operación que permite visualizar cuáles son los datos relacionados al valor de un Indicador.

Los datos se mostrarán en su máximo nivel de detalle.

Se tomará como referencia, para explicar esta operación, la siguiente tabla:

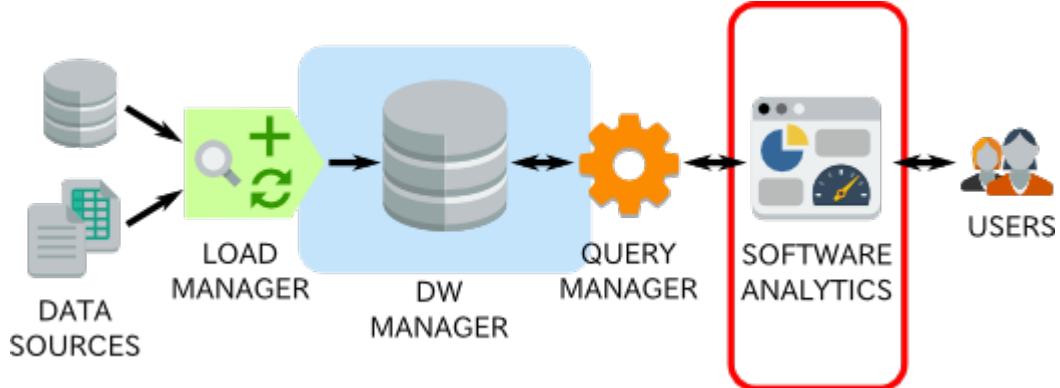
Dimensión Productos Jerarquía Productos Producto	Dimensión Años Jerarquía Años Año	Indicador Cantidad
Producto 1	2017	40
Producto 2	2017	52

Se aplicará la operación drill-through sobre el Indicador de la fila seleccionada, para obtener el detalle de este valor:

Dimensión Años Jerarquía Años Año Mes Día	Dimensión Clientes Jerarquía Clientes Cliente	Dimensión Productos Jerarquía Productos Producto	Dimensión Vendedores Jerarquía Vendedores Vendedor	Indicador Cantidad
2017 Enero 1	Cliente 1	Producto 1	Vendedor 1	3
2017 Enero 5	Cliente 1	Producto 1	Vendedor 1	2
2017 Enero 10	Cliente 2	Producto 1	Vendedor 1	10
2017 Enero 18	Cliente 2	Producto 1	Vendedor 1	6
2017 Enero 21	Cliente 2	Producto 1	Vendedor 1	1
2017 Febrero 4	Cliente 2	Producto 1	Vendedor 1	5
2017 Febrero 12	Cliente 2	Producto 1	Vendedor 1	4
2017 Febrero 16	Cliente 2	Producto 1	Vendedor 1	5
2017 Febrero 23	Cliente 2	Producto 1	Vendedor 1	3
2017 Febrero 24	Cliente 2	Producto 1	Vendedor 1	1

5) Software Analytics

Cuando hablamos de Software Analytics, nos referimos a todas aquellas herramientas de software mediante las cuales podremos explorar y explotar los datos almacenados en el DW.



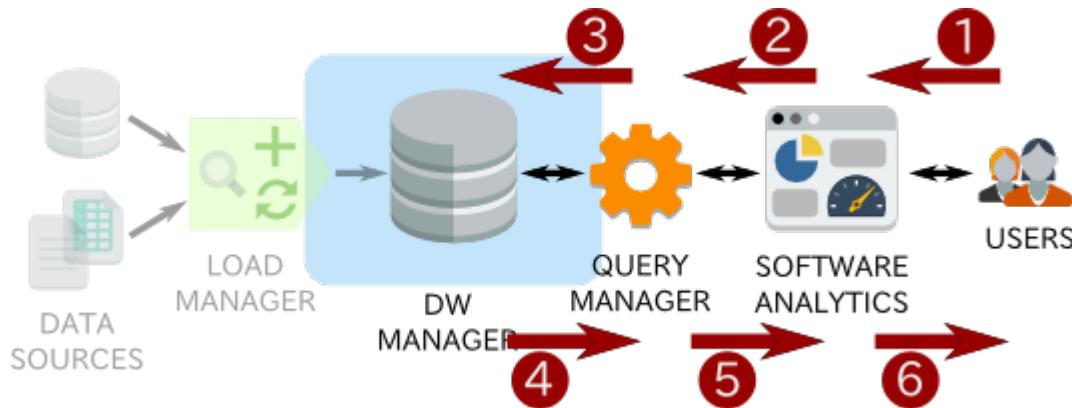
Estas herramientas constituyen el nexo entre el DW y l@s usuari@s, y son la parte visible y/o tangible del proceso de Data Warehousing.

Para la obtención de datos del DW, se utiliza principalmente:

- la metadata de las estructuras de datos que han sido creadas previamente (Cubos Multidimensionales, Business Models, etc.) y
- conexiones a bases de datos (JNDI, JDBC, ODBC)

Interacción

Cada vez que el usuari@ interactúa con el Software Analytic para explorar los datos del DW se llevan a cabo los siguientes pasos generales:



1. L@s Users seleccionan o establecen qué datos desean obtener del DW, mediante la GUI (interfaz gráfica) del Software Analytics.
2. El Software Analytics procesa el pedido de l@s Users, construye las consultas (utilizando la metadata) y las envía al Query Manager.
3. El Query Manager ejecuta las consultas sobre la estructura de datos con la que se esté trabajando (Cubo Multidimensional, Business Model, etc.).
4. El Query Manager obtiene los resultados de las consultas.
5. El Query Manager envía los datos al Software Analytics.
6. El Software Analytics presenta a l@s Users los datos requeridos.

Características

La mayoría de los Software Analytics comparten las siguientes características:

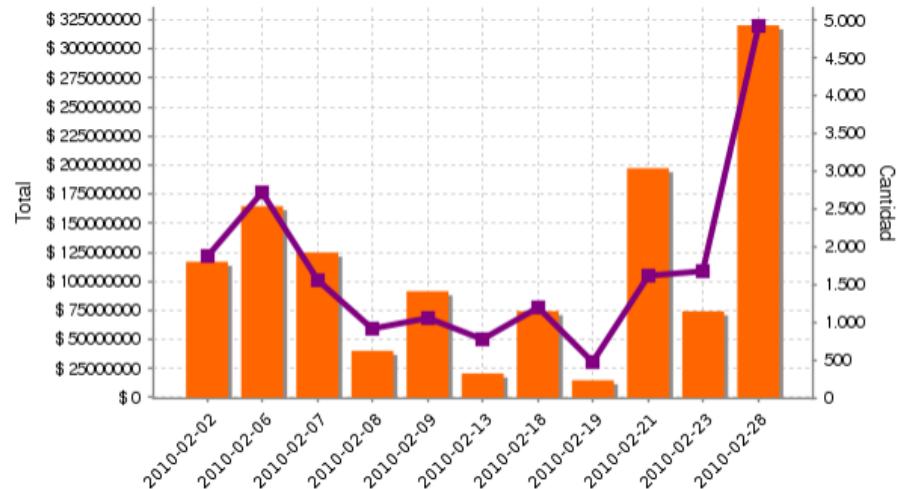
-  Accesibilidad a la información: el acceso a la información es transparente a l@s usuari@s finales. Esto se realiza a través de diferentes estructuras de datos permitiendo al usuari@ enfocarse exclusivamente en el análisis, sin preocuparse del origen y procedencia de los datos.
-  Apoyo en la toma de decisiones: permiten la exploración de los datos, a fin de seleccionar, filtrar y personalizar los mismos, para la obtención de información oportuna, relevante y útil, para apoyar el proceso de toma de decisiones.
-  Orientación a l@s usuari@s finales: disponen de GUIs (interfaz gráfica) amigables e intuitivas, que permiten a l@s usuari@s realizar análisis y consultas complejas, sin poseer conocimientos técnicos. Si bien los activos más valiosos son los datos en si, solo se podrán interpretar y analizar en la medida en que sean correctamente presentados.

Existen diferentes tipos de herramientas de consulta y análisis, y de acuerdo a la necesidad, tipos de usuari@s y requerimientos de información, se deberán seleccionar las más apropiadas al caso. Entre ellas se destacan las siguientes.

- Reporting
- OLAP
- Dashboards
- Data Mining
- EIS

Reporting

Las herramientas de Reporting ofrecen a l@s usuari@s, a través de pantallas gráficas intuitivas, la posibilidad de generar informes avanzados y detallados del tema de interés que se esté analizando. L@s usuari@s sólo deben seguir una serie de simples pasos, como por ejemplo seleccionar opciones de un menú, presionar tal o cual botón para especificar los elementos de datos, sus condiciones, criterios de agrupación y demás características que se consideren significativas. A continuación se presentan ejemplos de reportes:



DIA	CANTIDAD	TOTAL
2 (martes)	1.890	\$ 116.871.833
6 (sabado)	2.724	\$ 164.729.408
7 (domingo)	1.553	\$ 124.876.056
8 (lunes)	914	\$ 40.164.602
9 (martes)	1.058	\$ 91.513.364
13 (sabado)	770	\$ 20.669.667
18 (jueves)	1.200	\$ 74.341.779
19 (viernes)	477	\$ 14.837.290
21 (domingo)	1.623	\$ 197.417.892
23 (martes)	1.686	\$ 74.243.188
28 (domingo)	4.925	\$ 320.472.957

PRODUCTOS	RUBROS	TOTAL	INDICADOR
PRODUCTO 105	Rubro 7	\$ 47.427	
PRODUCTO 107	Rubro 7	\$ 8.805.000	
PRODUCTO 108	Rubro 3	\$ 9.099.984	
PRODUCTO 110	Rubro 4	\$ 261.238	
PRODUCTO 117	Rubro 9	\$ 650.180	
PRODUCTO 118	Rubro 9	\$ 17.822.403	
PRODUCTO 122	Rubro 2	\$ 3.541.584	

PRODUCTO	RUBRO	VALOR	INDICADOR
PRODUCTO 124	Rubro 9	\$ 3.690.168	
PRODUCTO 125	Rubro 1	\$ 22.415.368	
PRODUCTO 127	Rubro 8	\$ 458.834	
PRODUCTO 129	Rubro 4	\$ 23.192.623	
PRODUCTO 13	Rubro 3	\$ 24.745.322	
PRODUCTO 131	Rubro 10	\$ 472.428	

Actualmente las herramientas de Reporting cuentan con muchas prestaciones, las cuales permiten dar variadas formas y formatos a la presentación de la información. Entre las opciones más comunes se encuentran las siguientes:

- Parametrización de los datos devueltos.
- Selección de formatos de salida (planilla de cálculo, HTML, PDF, etc.).
- Inclusión de bar charts, pie charts, sparklines, etc.
- Utilización de plantillas de formatos de fondos.
- Inclusión de imágenes.
- Formatos tipográficos.
- Links a otros reportes.

OLAP

OLAP (On Line Analytic Processing) es el componente más poderoso del Data Warehousing, ya que contiene el motor de consultas multidimensionales especializado del DW.

Su principal objetivo es el de brindar respuestas rápidas a complejas preguntas, para interpretar la situación del negocio y tomar decisiones. Cabe destacar que lo que es realmente interesante en OLAP, NO es la ejecución de simples consultas tradicionales, sino la posibilidad de utilizar operadores tales como Drill-up, Drill-down, etc.

Se presentan aquí ejemplos de consultas OLAP:

Markets			Measures	
(All)	Territory	Country	Sales	Quantity
All Markets	- APAC		1,281,706	12,878
	APAC	+ Australia	630,623	6,246
		+ New Zealand	535,584	5,396
		+ Singapore	115,499	1,236
	+ EMEA		5,008,224	49,578
	- Japan		503,958	4,923
	Japan	+ Hong Kong	48,784	596
		+ Japan	188,168	1,842
		+ Philippines	94,016	961
		+ Singapore	172,990	1,524
	- NA		3,852,061	37,952
	NA	+ Canada	224,079	2,293
		+ USA	3,627,983	35,659
Total			10,645,949	105,331

	2003	2004	2005	
Territory	Sales	Sales	Sales	
APAC	343.082	601.606	337.018	
EMEA	1.681.987	2.396.408	929.829	
Japan	292.558	168.479	42.921	
NA	1.359.757	1.821.247	671.057	

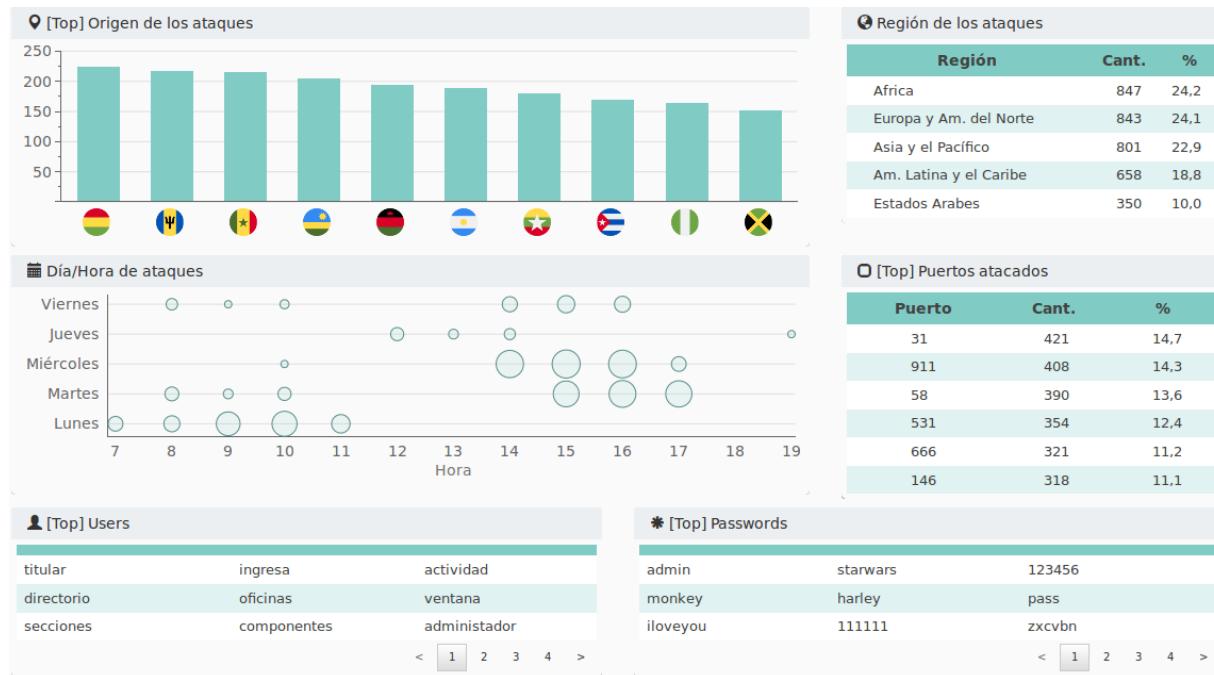
Además, a través de este tipo de herramientas, se puede analizar el negocio desde diferentes escenarios históricos, y proyectar el comportamiento y evolución desde una visión multidimensional, o sea, mediante la combinación de diferentes Perspectivas, temas de interés o Dimensiones. Esto

permite deducir tendencias, por medio del descubrimiento de relaciones entre las Perspectivas, que a simple vista serían difíciles de encontrar.

Las herramientas OLAP requieren que los datos estén organizados dentro del DW en forma multidimensional, por ello se utilizan las estructuras denominadas Cubos Multidimensionales.

Dashboards

Los Dashboards se pueden entender como una colección de componentes gráficos de análisis como: reportes, tablas, gráficos, consultas y análisis interactivos, etc; que hacen referencia a un tema en particular y que están relacionados entre sí. Por ejemplo:



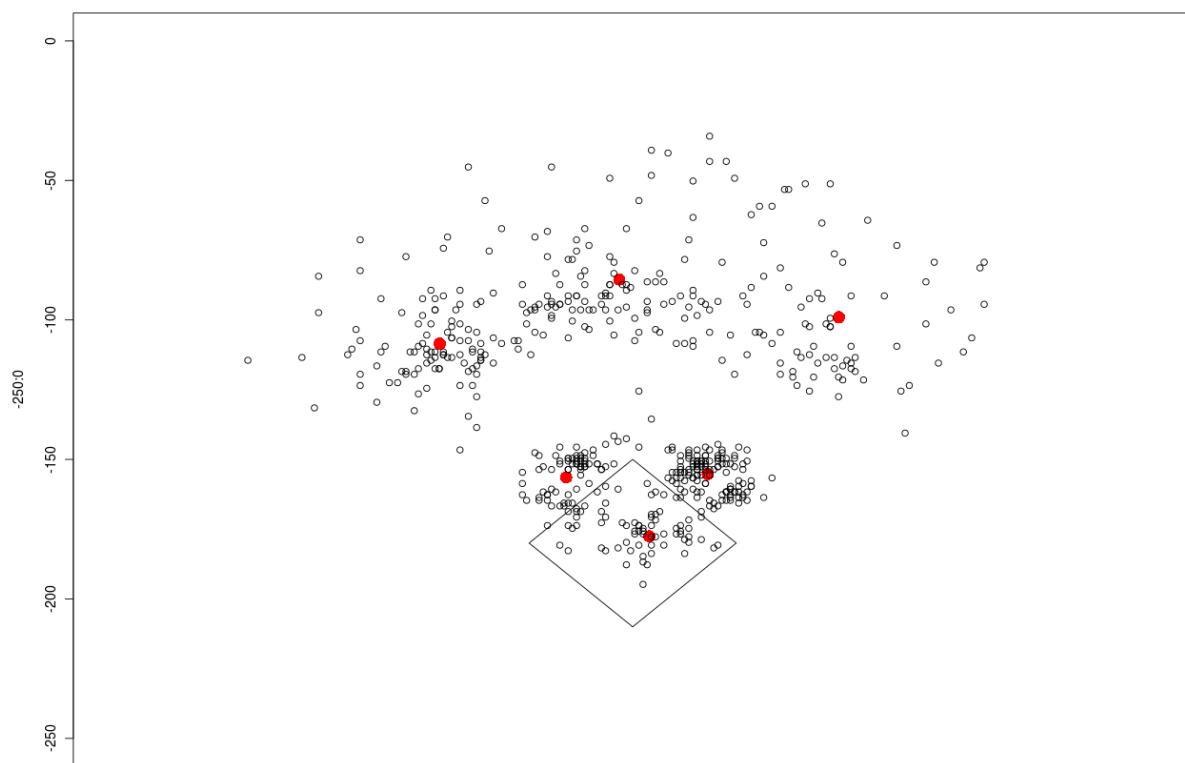
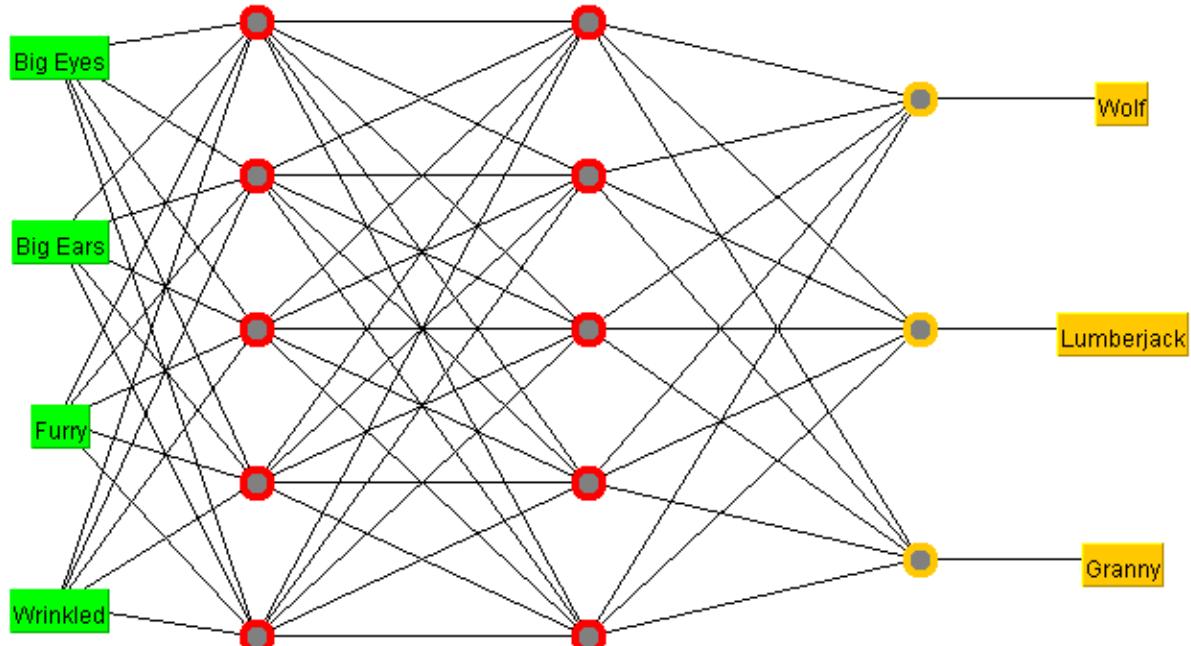
Existen diversas maneras de diseñar un Dashboard, cada una de las cuales tiene sus objetivos particulares, pero a modo de síntesis se expondrán algunas características generales que poseen:

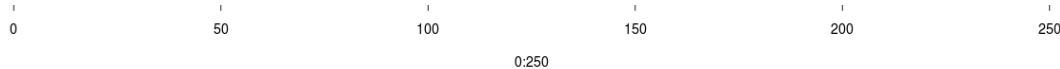
- Presentan la información altamente resumida.
- Se componen de consultas, reportes, análisis interactivos, charts (bar, pie, line, etc), semáforos, Indicadores causa-efecto, etc.
- Permiten evaluar la situación de la empresa con un solo golpe de vista.
- Poseen un formato de diseño visual muy llamativo.

Data Mining

El Data Mining se emplea para analizar factores de influencia en determinados procesos, predecir comportamientos futuros, extraer conocimientos ocultos, agrupar ítems similares, obtener secuencias de eventos que provocan comportamientos específicos.

El Data Mining puede implementarse sobre cualquier conjunto de datos, NO necesariamente sobre un DW.





El Data Mining también permite inferir comportamientos, modelos, relaciones y estimaciones de los datos, para poder desarrollar predicciones, sin la necesidad de contar con patrones o reglas preestablecidas, permitiendo tomar decisiones proactivas y basadas en un conocimiento acabado de la información.

Además brinda la posibilidad de dar respuesta a preguntas complicadas sobre los temas de interés, como por ejemplo:

- ¿Qué está pasando?
- ¿Por qué?
- ¿Qué pasaría si?

Estos cuestionamientos aplicados a una empresa podrían ser:

- ¿Cuál de los productos de tal marca y clase serán más vendidos en la zona norte en el próximo semestre? y ¿por qué?
- ¿Cuáles son los libros que querrá comprar tal cliente en el próximo ingreso?

Además se podrán ver los resultados en forma de reportes tabulares, matriciales, charts, tableros, etc.

Entonces, se puede definir Data Mining como una técnica para descubrir patrones y relaciones entre grandes cantidades de datos, que a simple vista o que mediante otros tipos de análisis NO se pueden deducir, ya que consumiría demasiado tiempo o estaría fuera de las expectativas.

Los sistemas Data Mining se desarrollan bajo lenguajes de última generación basados en Inteligencia Artificial y utilizan métodos matemáticos tales como:

- Redes Neuronales.
- Sistemas Expertos.
- Programación Genética.
- Árboles de Decisión.

Soportan además, sofisticadas operaciones de análisis como los sistemas Scoring, aplicaciones de Detección de Desviación y Detección de Fraude.

Es muy importante tener en cuenta que en las herramientas OLAP y Reporting, el análisis parte de una pregunta o hipótesis generada por l@s usuari@s, en cambio Data Mining permite generar estas hipótesis.

Generalmente las herramientas de Data Mining se integran con plataformas de hardware y software existentes (como DW) para incrementar el valor de los Data Sources existentes y para que puedan ser integradas con nuevos productos y sistemas en línea (como OLAP). Sumado a esto, implementar Data Mining sobre un DW tiene, entre otras ventajas, el soporte y beneficios de los procesos ETL y de las técnicas de limpieza de datos, tan necesarios en este tipo de análisis.

Redes Neuronales

Las Redes Neuronales se utilizan para construir modelos predictivos no lineales que aprenden a través de entrenamiento y que semejan la estructura de una red neuronal biológica.

Una Red Neuronal es un modelo computacional con un conjunto de propiedades específicas, como la habilidad de adaptarse o aprender, generalizar u organizar la información, todo ello basado en un procesamiento eminentemente paralelo.

Las Redes Neuronales pueden emplearse para:

- Resolver problemas en dominios complejos con variables continuas y categóricas.
- Modelar relaciones no lineales.
- Clasificar y predecir resultados.

Sistemas Expertos

Un Sistema Experto, puede definirse como un sistema informático (hardware y software) que simula a l@s expert@s human@s en un área de especialización dada.

La principal ventaja de estos sistemas es que l@s usuari@s con poca experiencia pueden resolver problemas que requieren el conocimiento de una persona experta en el tema.

Los Sistemas Expertos pueden utilizarse para:

- Realizar transacciones bancarias a través de cajeros automáticos.
- Controlar y regular el flujo de tráfico en las calles y en los ferrocarriles, mediante la operación automática de semáforos.
- Resolver complicados problemas de planificación en los cuales intervienen muchas variables.
- Descubrir relaciones entre diversos conjuntos de variables.

Programación Genética

El principal objetivo de la Programación Genética es lograr que las computadoras aprendan a resolver problemas sin ser explícitamente programadas para solucionarlos, generando de esta manera soluciones a partir de la inducción de los programas. El verdadero valor de esta inducción está fundamentado en que todos los problemas se pueden expresar como un programa de computadora.

La Programación Genética se utiliza para:

- Resolver problemas, para los cuales es difícil y NO natural tratar de especificar o restringir con anticipación el tamaño y forma de una solución eventual.
- Analizar sistemas que actúan sobre condiciones inestables en ambientes cambiantes.
- Generar de manera automática programas que solucionen problemas planteados.

Árboles de Decisión

Los Árboles de Decisión son estructuras en forma de árbol que representan conjuntos de decisiones. Estas decisiones generan reglas para la clasificación de un conjunto de datos, las cuales explican el comportamiento de una variable con relación a otras, y pueden traducirse fácilmente en reglas de negocio.

Son utilizados con finalidad predictiva y de clasificación.

Los Árboles de Decisión pueden emplearse para:

- Optimizar respuestas de campañas.
- Identificar clientes potenciales.
- Realizar evaluación de riesgos.

Detección de Desviación

La Detección de Desviación se encarga de analizar una serie de datos similares, y cuando encuentra un elemento que NO coincide con el resto lo considera una desviación.

Usualmente para la Detección de la Desviación en bases de datos grandes, se utiliza la información explícita externa a los datos, así como las limitaciones de integridad o modelos predefinidos. En un método lineal, al contrario, se enfoca el problema desde el interior de los datos, empleando la redundancia implícita de los mismos.

La Detección de Desviación puede utilizarse para:

- Descubrir excepciones a modelos establecidos.
- Delimitar grupos que cumplan con condiciones preestablecidas.

EIS

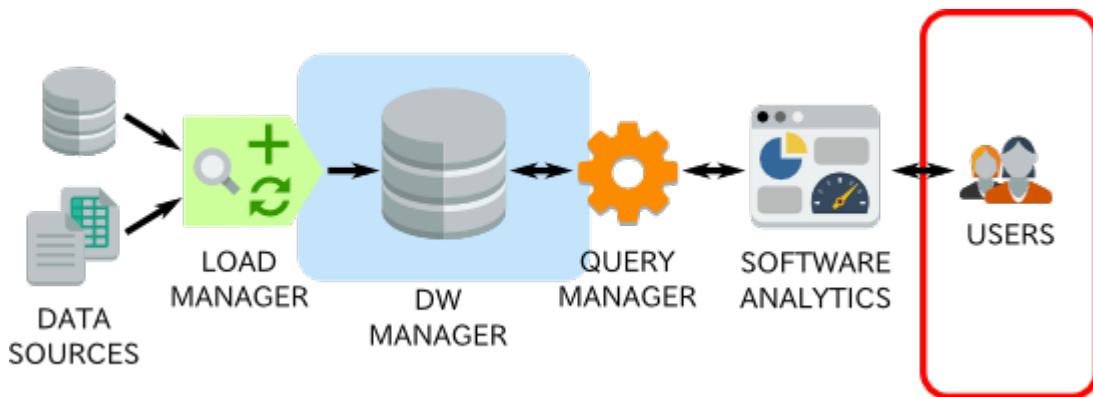
EIS (Executive Information System) proporciona medios sencillos para consultar, analizar y acceder a la información del estado del negocio. Además, pone a disposición facilidades para que l@s usuari@s puedan conseguir los datos buscados rápidamente, empleando el menor tiempo posible para comprender el uso de la herramienta.

Usualmente, EIS se utiliza para analizar los Indicadores de performance y desempeño del negocio o área de interés, a través de la presentación de vistas con datos simplificados, altamente consolidados, mayormente estáticos y preferentemente gráficos.

El concepto principal de esta herramienta, se basa en que quienes ocupan posiciones de responsabilidad empresarial y/o ejecutiva no suelen contar con el tiempo suficiente, ni las habilidades necesarias para analizar grandes cantidades de datos.

Al igual que OLAP y Data Mining, los EIS, se pueden aplicar independientemente del DW. Pero su implementación sobre un DW conlleva todas las ventajas implícitas del mismo.

6) Users



En los inicios del DWH, las soluciones Business Intelligence estaban enfocadas solo a un tipo de usuari@s, a aquell@s que se encargaban de tomar decisiones y de planificar las actividades del negocio.

En la actualidad las soluciones BI abarcan una amplia gama de tipos de usuari@s y se aplican a situaciones muy variadas, NO solo a la toma de decisiones estratégica y táctica.

A continuación veremos las principales diferencias entre l@s usuari@s de DW y l@s usuarios de OLTP (sistemas transaccionales):

- L@s usuari@s que acceden al DW concurrentemente son poc@s, en cambio los que acceden a los OLTP en un tiempo determinado son much@s más, pueden ser cientos o incluso miles. Esto se debe principalmente al tipo de información que contiene cada fuente.
- L@s usuari@s del DW generan por lo general consultas complejas, no predecibles y no anticipadas. Usualmente, cuando se encuentra una respuesta a una consulta se formulan nuevas preguntas más detalladas y así sucesivamente. Es decir, primero se analiza la información actual a nivel de datos para averiguar el qué, luego, para obtener mayor detalle y examinar el cómo, se trabajan con los datos ligeramente resumidos, derivados de la consulta anterior, y desde allí se pueden explorar los datos altamente resumidos. Es necesario tener en cuenta que en el DW es posible utilizar el detalle de datos históricos. Al contrario, l@s usuari@s de los OLTP solo manejan consultas predefinidas.
- L@s usuari@s del DW, generan consultas sobre una gran cantidad de registros, en cambio l@s del OLTP lo hacen sobre un pequeño grupo. Esto se debe a que el DW contiene información histórica e integra varias fuentes de datos.
- Las consultas de l@s usuari@s del DW no tienen tiempos de respuesta críticos, aunque sí se espera que se produzcan en el mismo día en que fueron realizadas. Mientras mayor sea el tamaño del DW y mientras más compleja sea la consulta, mayores serán los tiempos de respuestas. En cambio, las respuestas de las consultas en un OLTP son y deben ser inmediatas.
- En un DW, la única acción que pueden realizar l@s usuari@s es la de consulta. En cambio, en los OLTP, l@s usuari@s típicamente realizan acciones de actualización, tales como agregar, modificar, eliminar y consultar algún registro.

Las diferencias mencionadas entre estos dos tipos de usuari@s se pueden apreciar mejor en la

siguiente tabla comparativa:

	Users OLTP	Users DW
Acceso concurrente	Much@s	Poco@s
Tipo de consultas	Predefinidas	Complejas, NO predecibles y NO anticipadas
Registros consultados	Pocos	Muchos
Tiempo de respuesta	Crítico	NO crítico
Acciones permitidas	Agregar, modificar, eliminar y consultar	Consultar

Capítulo 4: Complementos

- Granularidad
 - Sistema de Misión Crítica
 - Data Mart
 - SGBD
 - Particionamiento
 - Business Models
 - Áreas de Datos
-

Granularidad

La granularidad representa el nivel de detalle con el que se desea almacenar la información sobre el negocio que se esté analizando. Por ejemplo, los datos referentes a ventas o compras realizadas por una empresa, pueden registrarse día a día, en cambio, los datos pertinentes a pagos de sueldos o cuotas de socios, podrán almacenarse a nivel de mes.

Mientras mayor sea el nivel de detalle de los datos, se tendrán mayores posibilidades analíticas, ya que los mismos podrán ser resumidos o agregados. Es decir, los datos que posean granularidad fina (nivel de detalle) podrán ser resumidos hasta obtener una granularidad media o gruesa. NO sucede lo mismo en sentido contrario, ya que por ejemplo, los datos almacenados con granularidad media podrán resumirse, pero NO podrán ser analizados a nivel de detalle. O sea, si la granularidad con que se guardan los registros es a nivel de día, estos datos podrán agregarse por semana, mes, semestre y año, en cambio, si estos registros se almacenan a nivel de mes, podrán agregarse por semestre y año, pero NO lo podrán hacer por día y semana.

La granularidad trabaja conjuntamente con la agregación, que en bases de datos es un cálculo que se realiza a varias filas y produce un único resultado, por ejemplo una sumatoria. Las agregaciones son cálculos NO reversibles, con la sumatoria es fácil de comprender, tenemos una serie de números que sumamos y obtenemos como resultado un nuevo número que representa el total, teniendo el total NO podemos obtener nuevamente los números de origen.

Sistema de Misión Crítica

L@s usuari@s siempre poseen una cierta resistencia al cambio cada vez que se les presenta una nueva herramienta o software. Es por ello que al principio no tod@s confiarán en el DW, y por ende NO lo utilizarán o lo subutilizarán. Esta barrera suele romperse a medida que pasa el tiempo y l@s usuari@s pueden comprobar por sí mism@s su buen funcionamiento, se adaptan, aprenden a usarlo y disuelvan sus dudas e incertidumbres.

Además, a medida que las empresas confian y emplean más el DW, y están más pendientes de la disponibilidad de los datos que él contiene, como así también en su acceso, el DW se torna fundamental para la misión del negocio o área que apoya, convirtiéndose paulatinamente en un Sistema de Misión Crítica. En el proceso de apropiación y uso del DW, es posible que se llegue al punto en que, un error en el mismo puede provocar una falla en las actividades del negocio.

Por este motivo, es de suma importancia que el DW posea una buena performance, seguridad y consistencia, y que todas las aplicaciones o herramientas que lo manipulen estén a disposición en todo momento.

Como la adaptación al DW será gradual, su construcción también debe serlo. Es prácticamente imposible construir un DW perfecto en una primera instancia. Es más, tratar de alcanzar este objetivo terminaría por ralentizar los procesos y devendría un fracaso seguro. De este modo, la maduración del DW se conseguirá paulatinamente con cada nueva iteración.

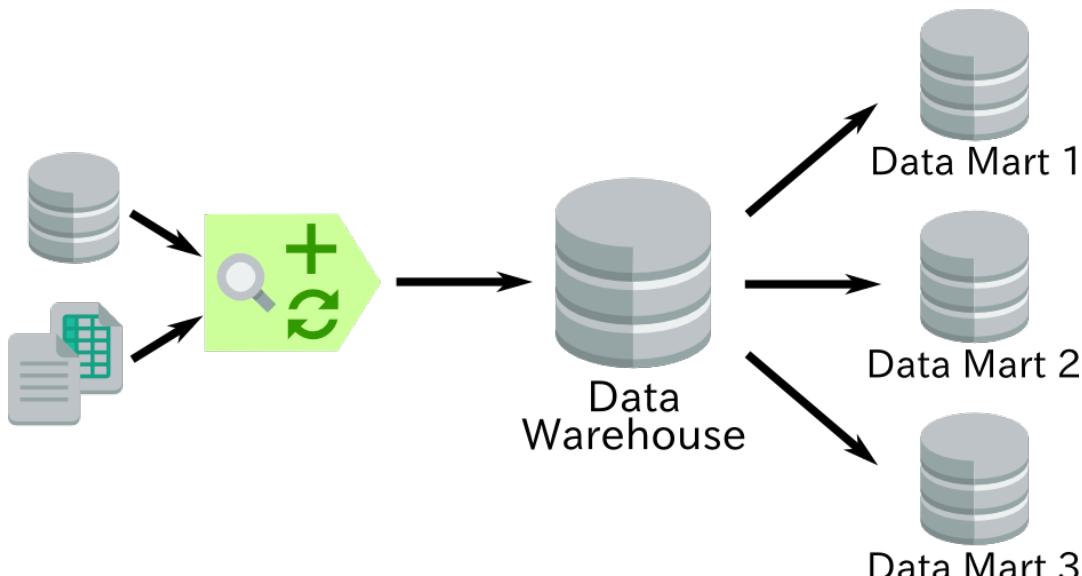
Data Mart

Un Data Mart (DM) es la implementación de un DW con alcance restringido a un área funcional, problema en particular, departamento, tema o grupo de necesidades.

Muchos DW comienzan siendo Data Mart, para, entre otros motivos, minimizar riesgos y producir una primera entrega en tiempos razonables. Una vez que éstos se han implementado exitosamente, su alcance se irá ampliando paulatinamente.

Los Data Marts pueden adoptar las siguientes arquitecturas:

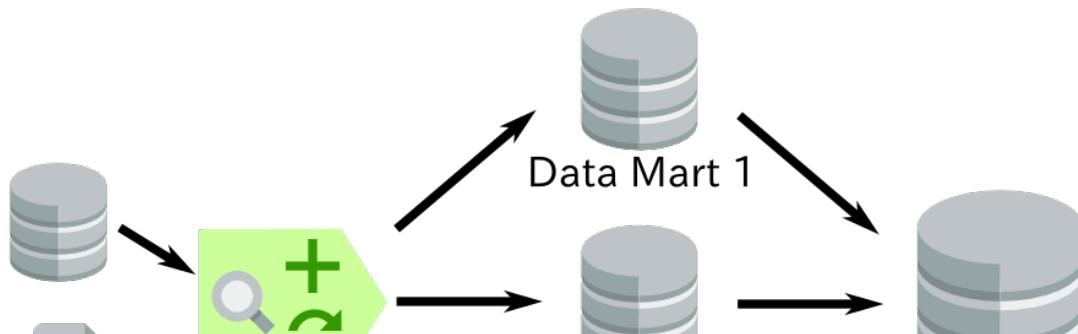
- Top-Down: primero se define el DW y luego se desarrollan, construyen y cargan los Data Marts.

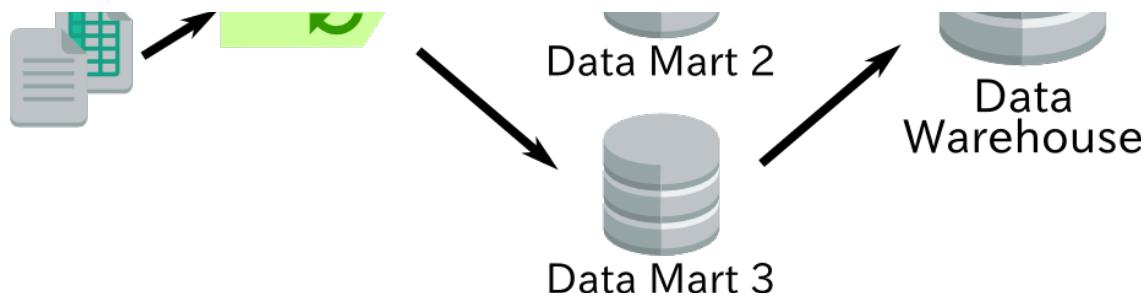


El DW es cargado a través de procesos ETL y luego éste alimenta a los diferentes DM, cada uno de los cuales recibirá los datos que correspondan al tema o departamento que traten.

Esta forma de implementación cuenta con la ventaja de no tener que incurrir en complicadas sincronizaciones de Hechos, pero requiere una gran inversión y una gran cantidad de tiempo de construcción.

- Bottom-Up: se definen previamente los Data Marts y luego se integran en un DW centralizado.





Los Data Marts se cargan a través de procesos de integración de datos (ETL), los cuales suministrarán la información adecuada a cada uno de ellos. En muchas ocasiones, los Data Marts son implementados sin que exista el DW, ya que tienen sus mismas características pero con la particularidad de que están enfocados en un tema específico. Luego de que hayan sido creados y cargados todos los Data Marts, se procederá a su integración con el DW.

- ✓ La ventaja de este modelo es que cada Data Mart se crea y pone en funcionamiento en un corto lapso de tiempo y se puede tener una pequeña solución a un costo NO tan elevado. Luego que todos los Data Marts estén puestos en marcha, se puede decidir si construir el Data Warehouse o NO. El mayor inconveniente está dado en tener que sincronizar los Hechos al momento de la consolidación en el DW.

Beneficios

Los principales beneficios de implementar Data Marts (DM) son:

- Son simples de implementar.
- Conllevan poco tiempo de construcción y puesta en marcha.
- Facilitan la administración de información confidencial.
- Reflejan rápidamente sus beneficios.
- Reducen la demanda del DW.

Data Marts como sub proyectos

Al diseñar e implementar Data Marts como parte de un proyecto DW, se debe tener en cuenta que el análisis que se efectuará, los modelos que intervendrán y el alcance, deben ser globales, con el fin de determinar, por ejemplo, tablas de Dimensiones comunes entre las diferentes áreas de trabajo. Esto evitará que se realicen tareas repetidas y se ahorrará tiempo.

SGBD

Los SGBD (Sistema de Gestión de Base de Datos) son un tipo de software muy específico, dedicados a servir de interfaz entre la base de datos, l@s usuari@s y las aplicaciones que lo utilizan. El SGBD está compuesto de una serie de herramientas que permiten trabajar con el modelo. La principal herramienta es un lenguaje que permite definir estructuras de datos y reglas de integridad, manipular y consultar, como así también definir las autorizaciones.

El propósito general de los SGBD es el de manejar de manera clara, sencilla y ordenada un conjunto de datos.

Existen diferentes objetivos que deben cumplir los SGBD, de los cuales se han enumerado los siguientes:

- Hacer transparente a l@s usuari@s los detalles del almacenamiento físico de los datos, mediante varios niveles de abstracción de la información.
- Permitir la realización de cambios a la estructura de datos, minimizando o anulando la necesidad de modificar las aplicaciones cliente.
- Proveer a l@s usuari@s la seguridad de que sus datos NO podrán ser accedidos, ni manipulados por quien no tenga autorización para hacerlo. Para poder implementarlo, el SGBD administra grupos, roles, usuari@s y permisos que se conjugarán con los distintos componentes que administra.
- Mantener la integridad de los datos mediante la definición y aplicación de reglas sobre las estructuras y los datos.
- Proporcionar una manera eficiente de realizar copias de seguridad de la información almacenada en ellos, y permitir a partir de estas copias restaurar los datos.
- Proporcionar una manera eficiente de realizar copias de seguridad de la información almacenada y permitir la restauración a partir de estas copias. Una copia de seguridad (backup o imagen) permite restaurar la base de datos en un momento dado, NO necesariamente cercano al presente; en consecuencia se pueden perder datos; muchos SGBD permiten, una vez restaurada una copia de seguridad, aplicar los cambios a partir de ella, de esta forma se minimizan las pérdidas de datos. Esto es posible debido a la existencia de bitácoras en las cuales se anotan todos los cambios que suceden en la BD en función del tiempo.
- Controlar el acceso concurrente de l@s usuari@s.
- Facilitar el manejo de grandes volúmenes de información.

Particionamiento

En un DW, el particionamiento se utiliza mayormente para dividir una tabla de Hechos, en varias tablas más pequeñas, a través de un criterio preestablecido. El criterio de particionamiento se denomina también, clave de particionamiento.

Hay dos razones principales para realizar particionamiento en nuestros DW:

Posibilitar un fácil y optimizado mantenimiento del DW y de sus correspondientes ETL.

Aumentar la performance de las consultas.

Las particiones mejoran los resultados de las consultas, ya que reducen al mínimo el número de registros de una tabla que deben leerse para satisfacer las consultas. Mediante la distribución de los datos en varias tablas, las particiones mejoran la velocidad y la eficacia de las consultas al DW. Además, al encontrarse los datos en distintas tablas, estas tablas son factibles de ser distribuidas en distintos SGBD, lo cual permite aprovechar recursos distribuidos mejorando notablemente la performance.

El tiempo es el criterio más comúnmente utilizado para realizar particiones, ya que de esta manera se limita el crecimiento de las tablas y se aumenta la estabilidad.

Las particiones pueden ser lógicas, físicas, horizontales o verticales.

Business Models



Un Business Model es un representación de los datos desde una Perspectiva empresarial, que permite que se pueda visualizar la información del negocio y su respectiva interrelación en forma de entidades de alto nivel.

Se compone de Entidades, Atributos y Relaciones, que están enfocados en dar respuesta a las preguntas de la información que se desea conocer.

El Business Model permite definir el comportamiento que tendrá cada objeto dentro de éste, como por ejemplo indicar cuáles campos serán utilizados para realizar agregaciones, cuál será el criterio empleado a tal fin y cuáles serán los campos que se utilizarán para analizar los datos.

Pero lo más importante de este tipo de estructura de datos, es que el mismo se define a través de reglas de negocio y teniendo en cuenta las áreas temáticas que son de interés en la empresa.

A continuación se listarán algunas de sus características más sobresalientes:

- Es completamente independiente de las estructuras organizacionales.
- Plantea la información de la empresa como si fuesen piezas que encajan entre sí.

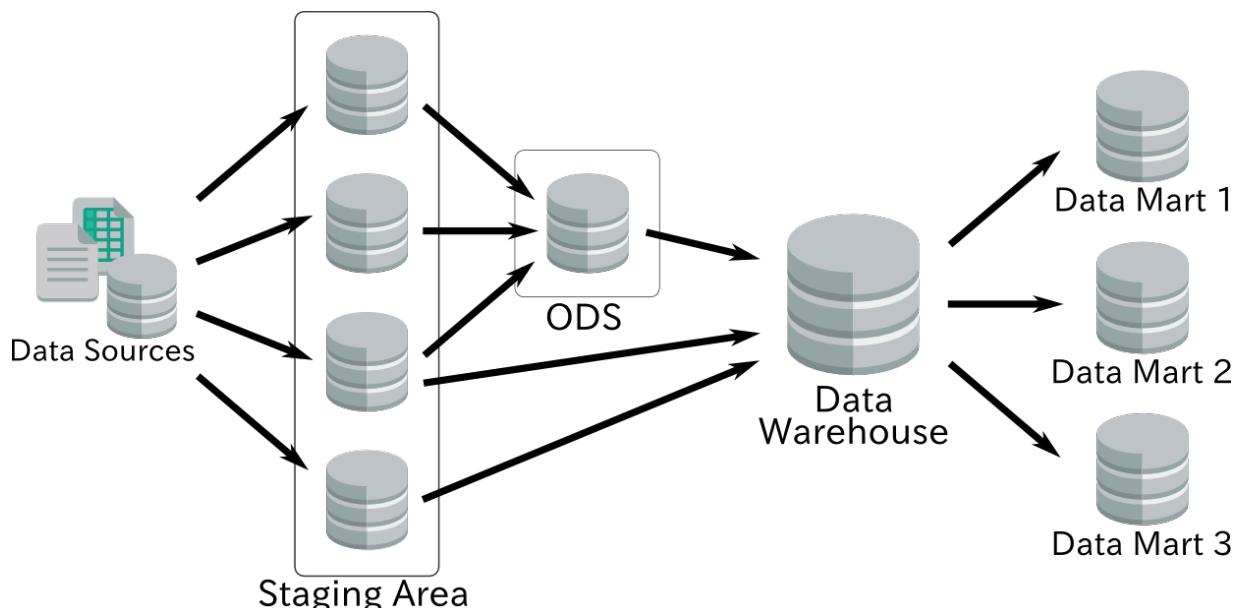
Áreas de Datos

Dentro del diseño de la arquitectura de un sistema de Data Warehouse es conveniente tener en consideración los diferentes entornos por los que han de pasar los datos en su camino hacia el DW o hacia los Data Marts de destino. Dada la cantidad de transformaciones que se han de realizar, y que normalmente el DW, además de cumplir su función de soporte a los requerimientos analíticos, realiza una función de integración de datos que van a conformar el Almacén Corporativo y que van a tener que ser consultados también de la manera tradicional por los sistemas operacionales, es muy recomendable crear diferentes áreas de datos en el camino entre los sistemas origen y las herramientas OLAP.

Cada una de estas Áreas se distingue por las funciones que realiza, de qué manera se organizan los datos en la misma, y a qué tipo de necesidad pueden dar servicio. El Área que se encuentra "al final del camino" es importante, pero no va a ser la única que almacene los datos que van a explotar las herramientas de reporting.

Tampoco hay una convención estándar sobre lo que abarca exactamente cada Área, y la obligatoriedad de utilizar cada una de ellas. Cada proyecto es diferente, e influyen muchos factores como la complejidad, el volumen de información del mismo, si realmente se quiere utilizar el Data Warehouse como almacén corporativo o Sistema Maestro de Datos, o si existen necesidades reales de soporte al reporting operacional.

En los siguientes puntos se explican las áreas de datos que suelen utilizarse, y se perfila una propuesta de arquitectura que hay que adaptar a las necesidades de cada proyecto, teniendo en cuenta que la utilización de cada Área de datos ha de estar justificada. No siempre todas son necesarias.



Staging Area

Es un Área temporal donde se recogen los datos que se necesitan de los sistemas origen.

Se recogen los datos estrictamente necesarios para las cargas, y se aplica el mínimo de

transformaciones a los mismos. No se aplican restricciones de integridad ni se utilizan claves. Los datos se tratan como si las tablas fueran ficheros planos. De esta manera se minimiza la afectación a los sistemas origen, la carga es lo más rápida posible para acotar la ventana horaria necesaria, y se reduce también al mínimo la posibilidad de error. Una vez que los datos han sido traspasados, el DW se independiza de los sistemas origen hasta la siguiente carga. Lo único que se suele añadir es algún campo que almacene la fecha de la carga.

Obviamente estos datos NO van a dar servicio a ninguna aplicación de reporting, son datos temporales que una vez que hayan cumplido su función son eliminados; en el esquema lógico de la arquitectura muchas veces NO aparecen, ya que su función es meramente operativa.

Algun@s autor@s consideran que la Staging Area abarca más de lo comentado, o incluso que engloba todo el entorno donde se realizan los procesos de ETL. En este documento se considera sólo como Área temporal.

Operational Data Store

Como su nombre indica, este Área es la que da soporte a los sistemas operacionales.

El modelo de datos del Almacén de Datos Operacional (ODS) sigue una estructura relacional y normalizada, para que cualquier herramienta de reporting o sistema operacional pueda consultar sus datos. Está dentro del Data Warehouse porque se aprovecha el esfuerzo de integración que supone la creación del Almacén de Datos Corporativo para poder atender también a necesidades operacionales, pero no es obligatorio. Ni siquiera es algo específico del BI; los ODS ya existían antes de que surgieran los conceptos de Data Warehousing y Business Intelligence.

No almacena datos históricos, muestra la imagen del momento actual, aunque eso no significa que no se puedan registrar los cambios.

Los datos del ODS se recogen de la Staging Area, y en este proceso sí se realizan transformaciones, limpieza de datos y controles de integridad referencial para que los datos estén perfectamente integrados en el modelo relacional normalizado.

Se debe tener en cuenta que la actualización de los datos del ODS no es instantánea, los cambios en los datos de los sistemas origen no se ven reflejados hasta que finaliza la carga correspondiente. Es decir, que los datos se refrescan cada cierto tiempo, cosa que hay que explicar a l@s usuari@s finales, porque los informes que se lancen contra el ODS siempre devolverán información a fecha de la última carga.

Por esta razón es recomendable definir una mayor frecuencia de carga para el ODS que para el Almacén Corporativo. Se puede refrescar el ODS cada 15 minutos, y el resto cada día, por ejemplo.

Almacén de Datos Corporativo

El Almacén de Datos Corporativo (DW) sí contiene datos históricos, y está orientado a la explotación analítica de la información que recoge.

Las herramientas DSS o de reporting analítico consultan tanto los Data Marts como el Almacén de Datos Corporativo. El DW puede servir para consultas en las que se precisa mostrar a la vez información que se encuentre en diferentes Data Marts.

En él se almacenan datos que pueden provenir tanto de la Staging Area como del ODS. Si ya se realizan procesos de transformación e integración en el ODS no se repiten para pasar los mismos datos al Almacén Corporativo. Lo que no se pueda recoger desde el ODS, hay que ir a buscarlo a la Staging Area.

El esquema se parece al de un modelo relacional normalizado, pero en él ya se aplican técnicas de desnormalización. No debería contener un número excesivo de tablas ni de relaciones ya que, por ejemplo, muchas relaciones jerárquicas que en un modelo normalizado se implementarían con tablas separadas aquí ya deberían crearse en una misma tabla, que después representará una Dimensión.

Otra particularidad es que la mayoría de las tablas han de incorporar campos de fecha para controlar la fecha de carga, la fecha en que se produce un Hecho, o el periodo de validez del registro.

Si el Data Warehouse no es demasiado grande, o el nivel de exigencia no es muy elevado en cuanto a los requerimientos 'operacionales', para simplificar la estructura se puede optar por prescindir del ODS, y si es necesario adecuar el Almacén de Datos Corporativo para servir tanto al reporting operacional como al analítico. En este caso, el Área resultante sería el DW Corporativo, pero en ocasiones también se denomina como ODS.

Data Mart

Otro Área de datos es el lugar donde se crean los Data Marts.

Éstos acostumbran a obtenerse a partir de la información recopilada en el área del Almacén Corporativo, aunque también puede ser a la inversa. Cada Data Mart es como un subconjunto de este almacén, pero orientado a un tema de análisis, normalmente asociado a un departamento de la empresa.

El Data Mart se diseña con estructura multidimensional, cada objeto de análisis es una tabla de Hechos enlazada con diversas tablas de Dimensiones. Si se diseña siguiendo el Modelo en Estrella habrá prácticamente una tabla para cada Dimensión; ésta es la versión más desnormalizada. Si se sigue un modelo de Copo de Nieve las tablas de Dimensiones estarán menos desnormalizadas y para cada Dimensión se podrán utilizar varias tablas enlazadas jerárquicamente.

Este área puede residir en la misma base de datos que las demás si la herramienta de explotación es de tipo ROLAP, o también puede crearse ya fuera de la BD, en la estructura de datos propia que generan las aplicaciones de tipo MOLAP, más conocida como los Cubos Multidimensionales.

Si se sigue una aproximación Top-down para la creación de los Data Mart, el paso del área de DW a ésta ha de ser bastante simple, lo que proporciona una cierta independencia sobre el software que se utiliza para el reporting analítico. Si por cualquier razón es necesario cambiar la herramienta de OLAP hay que redefinir los metadatos y regenerar los Cubos, y si el cambio es entre dos de tipo ROLAP ni siquiera esto último sería necesario. En cualquier caso, las áreas anteriores no tienen por qué ser modificadas.

Capítulo 5: Metodología HEFESTO

- Resumen
 - Introducción
 - Descripción
 - Características
 - Empresa analizada
 - Paso 1) Análisis de Requerimientos
 - ▶ 1.1) Preguntas de Negocio
 - ▶ 1.2) Indicadores y Perspectivas
 - ▶ 1.3) Modelo Conceptual
 - Paso 2) Análisis de Data Sources
 - ▶ 2.1) Hechos e Indicadores
 - ▶ 2.2) Mapeo
 - ▶ 2.3) Granularidad
 - ▶ 2.4) Modelo Conceptual Ampliado
 - Paso 3) Modelo Lógico del DW
 - ▶ 3.1) Tipología
 - ▶ 3.2) Tablas de Dimensiones
 - ▶ 3.3) Tablas de Hechos
 - ▶ 3.4) Uniones
 - Cubo Multidimensional
-

Resumen

En la segunda parte de esta publicación se presenta la metodología HEFESTO para la construcción de un Data Warehouse.

El principal objetivo de la metodología HEFESTO es facilitar el arduo trabajo que significa construir un Data Warehouse desde cero, aportando información que permitirá mejorar su performance. La metodología está orientada a amortiguar el tedio que provoca seguir pasos sin comprender el por qué de su ejecución.

La metodología tiene como punto de partida la recolección de requerimientos y necesidades de información de l@s usuari@s y concluye con la confección de un esquema lógico y sus respectivos procesos de extracción, transformación y carga de datos. Se exemplificará cada etapa de la metodología a través de su aplicación a una empresa real, que servirá de guía para la visualización de los resultados esperados en cada paso y para clarificar los conceptos enunciados.

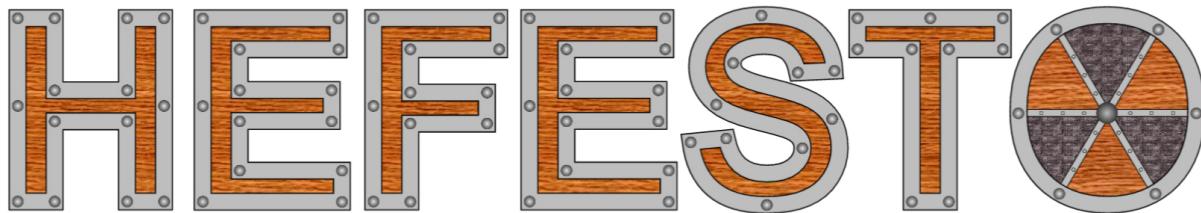
Se describirán los aspectos sobresalientes de la metodología y se detallará cada paso con su respectiva aplicación.

Se expondrán tópicos que deben tenerse en cuenta al momento de construir e implementar un Data Warehouse.

Se sumará a lo anterior, la creación de Cubos Multidimensionales basados en el DW resultante del caso práctico que se desarrolló anteriormente.

Introducción

En esta sección se presentará la metodología HEFESTO, que permitirá la construcción de Data Warehouse de forma sencilla, ordenada e intuitiva. Su nombre fue inspirado en el dios griego de la construcción y el fuego, y su logotipo es el siguiente:



HEFESTO es una metodología, cuya propuesta está fundamentada en una extensa investigación, comparación de metodologías existentes y el aporte de experiencias propias en procesos de diseño e implementación de DW.

Cabe destacar que HEFESTO está en continua evolución, y se han tenido en cuenta, como gran valor agregado, todos los feedbacks que han aportado quienes han utilizado esta metodología en diversos países y con diversos fines.

Es fundamento de esta metodología infundir en l@s lector@s una idea cabal sobre cada paso que se plantea, de tal forma que en tiempo de aplicación se posea argumentos sólidos para defender y sostener la implementación y además, ser capaces de plantear nuevos interrogantes que aporten valor extra al trabajo y a la metodología.

Es deseo de quienes escribimos este libro, que esta metodología siga creciendo y evolucionando con aportes diversos en cuanto a saberes, experiencias y condiciones. La base de dedicar tiempo en escribir un libro y luego publicarlo de forma libre es la compartir conocimiento, sin embargo, es muy importante que este conocimiento NO sea estático, y la forma en que este conocimiento se mantenga en movimiento y constante crecimiento, es sin duda el aporte de la comunidad.

La metodología HEFESTO puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software.

Lo que se busca, es entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del DW y motivar a l@s usuari@s.

Se acompañará cada desarrollo teórico con una implementación basada en una empresa real a fin de mostrar los resultados que se deben obtener y ejemplificar cada concepto de forma concreta y contundente.

Descripción

HEFESTO está compuesto por los siguientes pasos:

- 1) ANÁLISIS DE REQUERIMIENTOS
 - ▶ 1.1) Preguntas del Negocio
 - ▶ 1.2) Indicadores y Perspectivas
 - ▶ 1.3) Modelo Conceptual
- 2) ANÁLISIS DE DATA SOURCES
 - ▶ 2.1) Hechos e Indicadores
 - ▶ 2.2) Mapeo
 - ▶ 2.3) Granularidad
 - ▶ 2.4) Modelo Conceptual Ampliado
- 3) MODELO LÓGICO DEL DW
 - ▶ 3.1) Tipología
 - ▶ 3.2) Tablas de Dimensiones
 - ▶ 3.3) Tablas de Hechos
 - ▶ 3.4) Uniones
- 4) INTEGRACIÓN DE DATOS
 - ▶ 4.1) Carga Inicial
 - ▶ 4.2) Actualización

Como ya se planteó en el apartado anterior, la metodología Hefesto, se inicia con la recolección de las necesidades de información de l@s usuari@s y de esta manera se obtienen las preguntas claves del negocio. Luego, se deben identificar los Indicadores resultantes de los interrogantes realizados y sus respectivas Perspectivas de análisis, a través de las cuales se construirá el modelo conceptual de datos del DW.

Después, se analizarán los Data Sources en pos de determinar cómo se construirán los Indicadores, señalando el mapeo correspondiente y seleccionando los campos de estudio de cada Perspectiva.

Una vez realizado esto, se pasará a la construcción del Modelo Lógico del DW, en donde se definirá cuál será el tipo de esquema que se implementará.

Seguidamente, se confeccionarán las tablas de Dimensiones y las tablas de Hechos, para luego efectuar sus respectivas uniones.

Finalmente, utilizando técnicas de limpieza y calidad de datos, procesos ETL, etc, se definirán políticas y estrategias para la Carga Inicial del DW y su respectiva Actualización.

Características

La metodología HEFESTO posee las siguientes características:

- Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- La piedra fundamental la constituyen los requerimientos de l@s usuari@s, por lo cual, se adapta con facilidad y rapidez a los cambios del negocio.
- Reduce drásticamente la resistencia al cambio, ya que involucra a l@s usuari@s finales en cada etapa para que tomen decisiones respecto al comportamiento y funciones del DW, y además expone resultados inmediatos.
- Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.
- Es independiente del tipo de ciclo de vida que se emplee para contener la metodología.
- Es independiente del software/hardware que se utilicen para su implementación.
- Cuando se culmina con una fase, los resultados obtenidos se constituyen en la entrada de la fase siguiente.
- Se aplica en Data Warehouse y en Data Mart.

Empresa analizada

Antes de comenzar con el primer paso, es menester describir las características principales de la empresa a la cual se le aplicará la metodología HEFESTO, así se podrá tener como base un ámbito predefinido y se comprenderá cada decisión que se tome con respecto al diseño del DW.

Además, este análisis ayudará a conocer el funcionamiento y accionar de la empresa, lo que permitirá examinar e interpretar de forma óptima las necesidades de información de la misma.

Identificación de la empresa

La empresa analizada, desarrolla las actividades comerciales de mayorista y minorista de artículos de limpieza, en un ambiente geográfico de alcance nacional. De acuerdo a su volumen de operaciones, se la puede considerar de tamaño mediano.

Con respecto a su clasificación, es una sociedad de responsabilidad limitada con fines de lucro.

Su estructura está formalizada y posee características de una organización funcional.

Objetivos

Su objetivo principal es el de maximizar sus ganancias. Pero también, se puede adicionar el objetivo de expandirse a un nuevo nivel de mercado, con el fin de conseguir una mayor cantidad de client@s y posicionarse competitivamente por sobre sus rivales.

Otra meta que persigue, pero que aún no está definida como tal, es la de incursionar en otros rubros para lograr diversificarse.

Políticas

La empresa posee pocos grandes clientes con un gran poder adquisitivo, y son precisamente estos, los que adquieren el volumen de los productos que se comercializan. Debido a ello, la política que se utiliza para cubrir los objetivos antes mencionados, es la de satisfacer ampliamente las necesidades de sus client@s, brindándoles confianza y promoviendo un ambiente familiar entre l@s mism@s. Esta acción se realiza con el fin de mantener l@s client@s actuales y para que los nuev@s se interesen en su forma de operar.

Existe otra política que es implícita, por lo cual, no está definida tan estrictamente como la anterior, y es la de mejorar continuamente, con el objetivo de sosegar las exigencias y cambios en el mercado en el que actúa y para conseguir una mejor posición respecto a sus competidor@s.

Estrategias

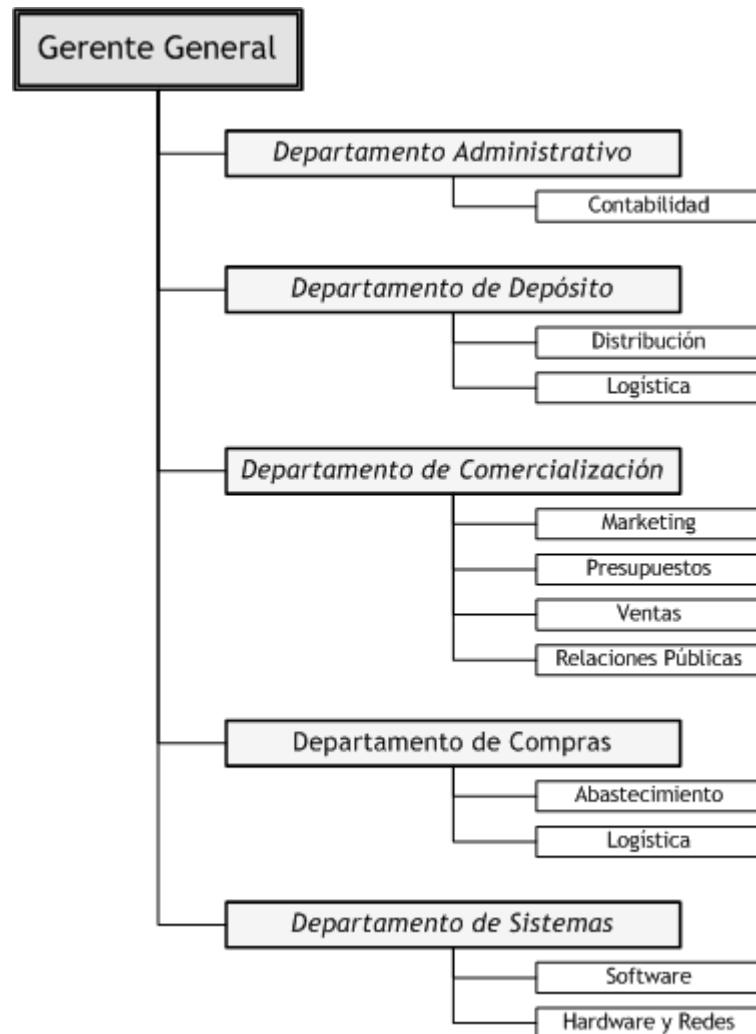
Dentro de las estrategias existentes, se han destacado dos por considerarse más significativas, ellas son:

Expandir el ámbito geográfico, creando varias sucursales en puntos estratégicos del país.

Añadir nuevos rubros a su actividad de comercialización.

Organigrama

A continuación, se expondrá un organigrama que fue confeccionado a partir de los datos suministrados en la empresa, ya que no existía ninguno previamente predefinido.



Datos del entorno específico

L@s client@s con que cuenta son bastantes variad@s y cubren un amplio margen. L@s mism@s son tanto provinciales, como nacionales, con diferentes tipos de poder adquisitivo.

Con respecto a sus proveedor@s, la empresa posee en algunos rubros diversas opciones de las cuales puede elegir y comparar, pero en otros solo cuenta con pocas alternativas.

Además, tiene como rivales a nivel de mayoreo, vari@s competidor@s importantes y ya consolidad@s en el mercado, pero, a nivel minorista aventaja por su tamaño y volumen de actividades a sus principales competidor@s.

Relación de las metas de la organización con las del DWH

El DWH coincide con las metas de la empresa, ya que ésta necesita mejorar su eficiencia en la toma de decisiones y contar con información detallada a tal fin. Esto es vital, ya que es muy importante

para procurar una mayor ventaja competitiva conocer cuáles son los factores que inciden directamente sobre su rentabilidad, como así también, analizar su relación con otros factores y sus respectivos por qué.

El DWH aportará un gran valor a la empresa; entre las principales ventajas e inconvenientes que solucionará se pueden mencionar los siguientes:

- Permitirá a l@s usuari@s tener una visión general del negocio.
- Transformará datos operativos en información analítica, enfocada a la toma de decisiones.
- Se podrán generar reportes dinámicos, ya que actualmente son estáticos y no ofrecen ninguna facilidad de análisis.
- Soportará la estrategia de la empresa.
- Aportará a la mejora continua de la estructura de la empresa.

Procesos

Los principales procesos que se llevan a cabo son los siguientes:

- Venta:
 - ▶ Minorista: es la que se le realiza a l@s client@s particulares que se acercan hasta la empresa para adquirir los productos que requieren.
 - ▶ Mayorista: es la que se le efectúa a l@s grandes client@s, ya sea por medio de comunicaciones telefónicas, o a través de visitas o reuniones.

Al realizarse una venta, el departamento de Depósito se encarga de controlar el stock, realizar encargos de mercadería en caso de no cubrir lo solicitado, armar el pedido y enviarlo por medio de transporte propio o de tercer@s al destino correspondiente.

- Compra:

El departamento de Compras, al recibir del departamento de Depósito las necesidades de mercadería, realiza una comparación de los productos ofrecidos por sus diferentes proveedor@s en cuestión de precio, calidad y confianza. Posteriormente, se efectúa el pedido correspondiente.

Paso 1) Análisis de Requerimientos

Lo primero que se hará será identificar los requerimientos de l@s usuari@s a través de preguntas que expliciten los objetivos de su organización. Luego, se analizarán estas preguntas a fin de identificar cuáles serán los Indicadores y Perspectivas que serán tomadas en cuenta para la construcción del DW. Finalmente se confeccionará un Modelo Conceptual en donde se podrá visualizar el resultado obtenido en este primer paso.

Si, por ejemplo, el requerimiento consta de dos Data Marts, deberá aplicarse la metodología dos veces, una por cada Data Mart. Del mismo modo, si se analizan dos áreas de interés de negocio, como el área de "Ventas" y "Compras", se deberá aplicar la metodología al área de *Ventas* y *Compras* de forma independiente.

1.1) Preguntas de Negocio

El primer paso comienza con el acopio de las necesidades de información, el cual puede llevarse a cabo a través de variadas y diferentes técnicas, cada una de las cuales poseen características inherentes y específicas, como por ejemplo: entrevistas, cuestionarios, observaciones, etc.

El Análisis de los Requerimientos de l@s diferentes usuari@s, es el punto de partida de esta metodología, ya que ell@s son l@s que deben, en cierto modo, guiar la investigación hacia un desarrollo que refleje claramente lo que se espera del DW, en relación a sus funciones y cualidades.

El objetivo principal de esta fase, es la de obtener e identificar las necesidades de información clave de alto nivel, que es esencial para lograr las metas y ejecutar las estrategias de la empresa, además facilitará que la toma de decisiones sea eficaz y eficiente.

Debe tenerse en cuenta que las necesidades de la información que será recolectada, es la que proveerá el soporte para desarrollar los pasos sucesivos, por lo cual, es muy importante que se preste especial atención al relevamiento inicial.

Una forma de asegurarse de que se ha realizado un buen análisis, es corroborar que el resultado del mismo haga explícitos los objetivos estratégicos planteados por la empresa que se está estudiando.

Otra forma de encaminar el relevamiento, es enfocar las necesidades de información en los procesos principales que desarrolle la empresa en cuestión.

La idea central es, que se formulen preguntas complejas sobre el negocio, que incluyan variables de análisis que se consideren relevantes, ya que son estas las que permitirán estudiar la información desde diferentes Perspectivas.

Un punto importante que debe tenerse muy en cuenta, es que la información debe estar soportada de alguna manera por algún Data Source, ya que de otra forma, no se podrá elaborar el DW.

Caso práctico

En las primeras entrevista se indagó a l@s usuari@s en busca de sus necesidades de información, pero las mismas abarcaban casi todas las actividades de la empresa, por lo cual se les pidió que escogieran el proceso que considerasen más importante en las actividades diarias de la misma y que estuviese soportado de alguna manera por algún Data Source.

El proceso elegido fue el de *Ventas*.

Una vez seleccionado el proceso, se comenzó a identificar qué era lo que les interesaba conocer acerca de este proceso y cuáles eran las variables o Perspectivas que deben tenerse en cuenta en los análisis en pos de la toma de decisiones.

Se les preguntó cuáles eran, a su criterio, los Indicadores más representativos del proceso de Ventas y cuál es el análisis que se desea realizar. La respuesta arrojó como resultado que la cantidad de unidades vendidas y el monto total de ventas son los números más relevantes en

este proceso.

Luego se les preguntó cuáles serían las Perspectivas desde las cuales se consultarán dichos Indicadores. Para simplificar esta tarea se les presentó una serie de ejemplos concretos de otros casos similares.

Las Preguntas de Negocio obtenidas fueron las siguientes:

- Se desea conocer cuántas unidades de cada producto fueron vendidas a sus clientes en un periodo determinado. O en otras palabras: **Unidades vendidas de cada producto a cada cliente en un tiempo determinado.**
- Se desea conocer cuál fue el monto total de ventas de productos a cada cliente en un periodo determinado. O en otras palabras: **Monto total de ventas de cada producto a cada cliente en un tiempo determinado.**

Debido a que la Dimensión Tiempo es un elemento fundamental en el DW, se hizo hincapié en él. Además, se puso mucho énfasis en dejar en claro a l@s usuari@s, a través de ejemplos prácticos, que es este componente el que permitirá tener varias versiones de los datos a fin de realizar un correcto análisis posterior.

Como se puede apreciar, las necesidades de información expuestas armonizan con los objetivos y estrategias de la empresa, ya que es precisamente esta información la que proveerá un ámbito para la toma de decisiones, que en este caso permitirá analizar el comportamiento de l@s client@s a quienes que se pretende satisfacer ampliamente, para así lograr obtener una ventaja competitiva y maximizar las ganancias.

1.2) Indicadores y Perspectivas

Una vez que se han establecido las preguntas de negocio, se debe proceder a su descomposición para descubrir los Indicadores que se utilizarán y las Perspectivas de análisis que intervendrán.

Para ello, se debe tener en cuenta que los Indicadores, son valores numéricos y representan lo que se desea analizar concretamente, por ejemplo: saldos, importes, promedios, cantidades, sumatorias, fórmulas, etc.

En cambio, las Perspectivas se refieren a las entidades mediante las cuales se quieren examinar los Indicadores, con el fin de responder a las preguntas planteadas, por ejemplo: clientes, proveedores, sucursales, países, productos, rubros, etc. Cabe destacar, que el Tiempo suele considerarse comúnmente como una Perspectiva.

Caso práctico

A continuación, se analizarán las preguntas obtenidas en el paso anterior y se detallarán cuáles son sus Indicadores y Perspectivas.

"Unidades vendidas de cada producto a cada cliente en un tiempo determinado"



"Monto total de ventas de cada producto a cada cliente en un tiempo determinado".



Los Indicadores son:

- Unidades vendidas
- Monto total de ventas

Y las Perspectivas de Análisis son:

- Clientes
- Productos
- Tiempo

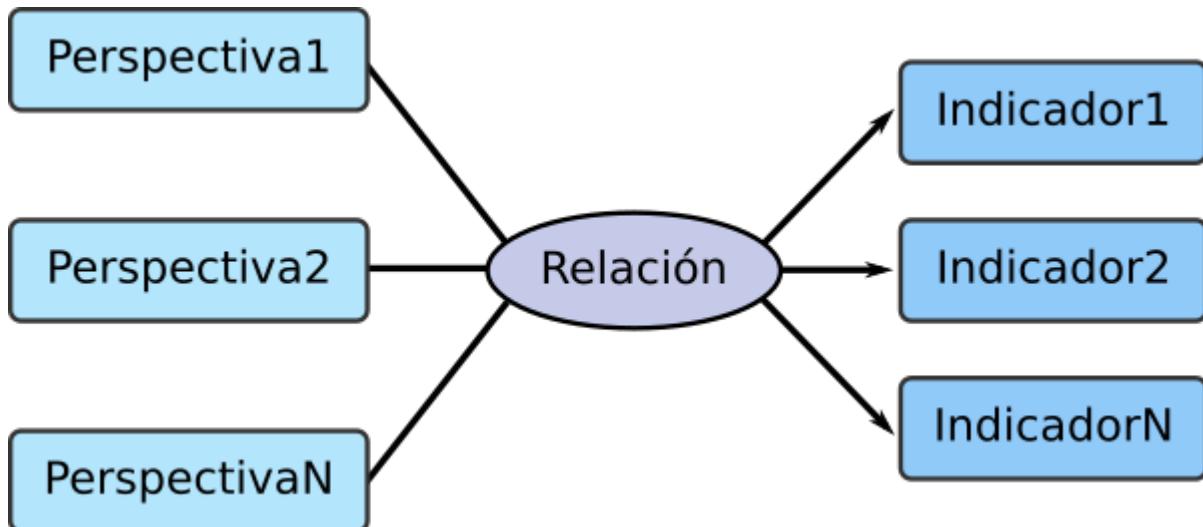
1.3) Modelo Conceptual

En esta etapa, se construirá un Modelo Conceptual a partir de los Indicadores y Perspectivas obtenidas en el paso anterior.

Un Modelo Conceptual es una descripción de alto nivel de la estructura de la base de datos, en la cual la información es representada a través de Objetos, Relaciones y Atributos.

A través de este Modelo, se podrá observar con claridad cuáles son los alcances del proyecto, para luego poder trabajar sobre ellos. Además, al poseer un alto nivel de definición de los datos, permite que pueda ser presentado ante l@s usuari@s y explicado con facilidad.

La representación gráfica del Modelo Conceptual es la siguiente:



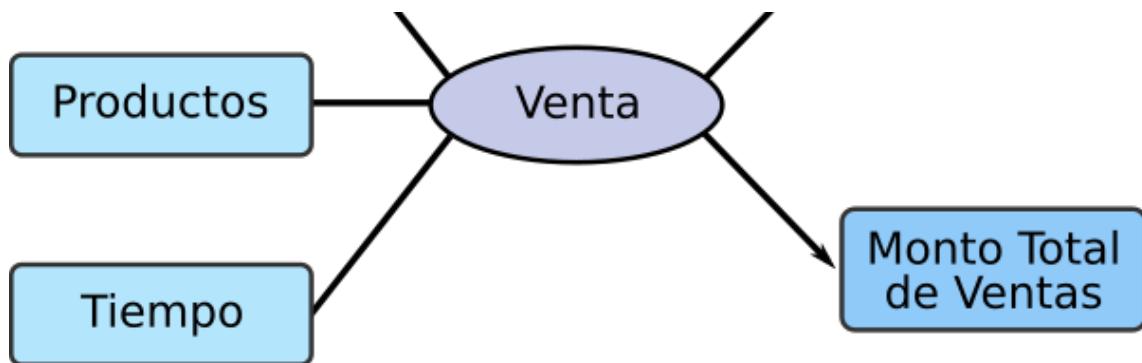
- A la izquierda se colocan las **Perspectivas** seleccionadas, que serán unidas a
- un óvalo central que representa y lleva el nombre de la **Relación** que existe entre ellas;
- la Relación, constituye el proceso o área de estudio elegida;
- de dicha Relación y entrelazadas con flechas, se desprenden hacia la derecha los **Indicadores**.

El Modelo Conceptual permite de un solo vistazo y sin poseer demasiados conocimientos previos, comprender cuáles serán los resultados que se obtendrán, cuáles serán las variables que se utilizarán para analizarlos y cuál es la relación que existe entre ellos.

Caso práctico

El Modelo Conceptual resultante de los datos que se han recolectado, es el siguiente:





Como puede observarse, la Relación mediante la cuál se unen las diferentes Perspectivas, para obtener como resultado los Indicadores requeridos por l@s usuari@s, es precisamente **Venta**.

Paso 2) Análisis de Data Sources

Se analizarán los Data Sources para determinar cómo serán calculados los Indicadores y para establecer el mapeo entre el Modelo Conceptual creado en el paso anterior y los datos de la empresa.

Se definirán qué campos se incluirán en cada Perspectiva.

Y se ampliará el modelo conceptual con la información obtenida en este paso.

2.1) Hechos e Indicadores

En este paso se deberán explicitar cómo se calcularán los Indicadores, definiendo los siguientes conceptos para cada uno de ellos:

- Hecho/s que lo componen, con su respectiva fórmula de cálculo. Por ejemplo: **Hecho1 + Hecho2**
- Función de agregación que se utilizará. Por ejemplo: **SUM, AVG, COUNT**, etc.

Caso práctico

Los Indicadores se calcularán de la siguiente manera:

- Indicador: **Unidades Vendidas**:
 - ▶ Hechos: **Unidades Vendidas**
 - ▶ Función de agregación: **SUM**

Aclaración: el Indicador **Unidades Vendidas** representa la sumatoria de las unidades que se han vendido de un producto en particular.

- Indicador: **Monto Total de Ventas**:
 - ▶ Hechos: **(Unidades Vendidas) * (Precio de Venta)**
 - ▶ Función de agregación: **SUM**

Aclaración: el Indicador **Monto Total de Ventas** representa la sumatoria del monto total que se ha vendido de cada producto, y se obtiene al multiplicar las unidades vendidas, por su respectivo precio.

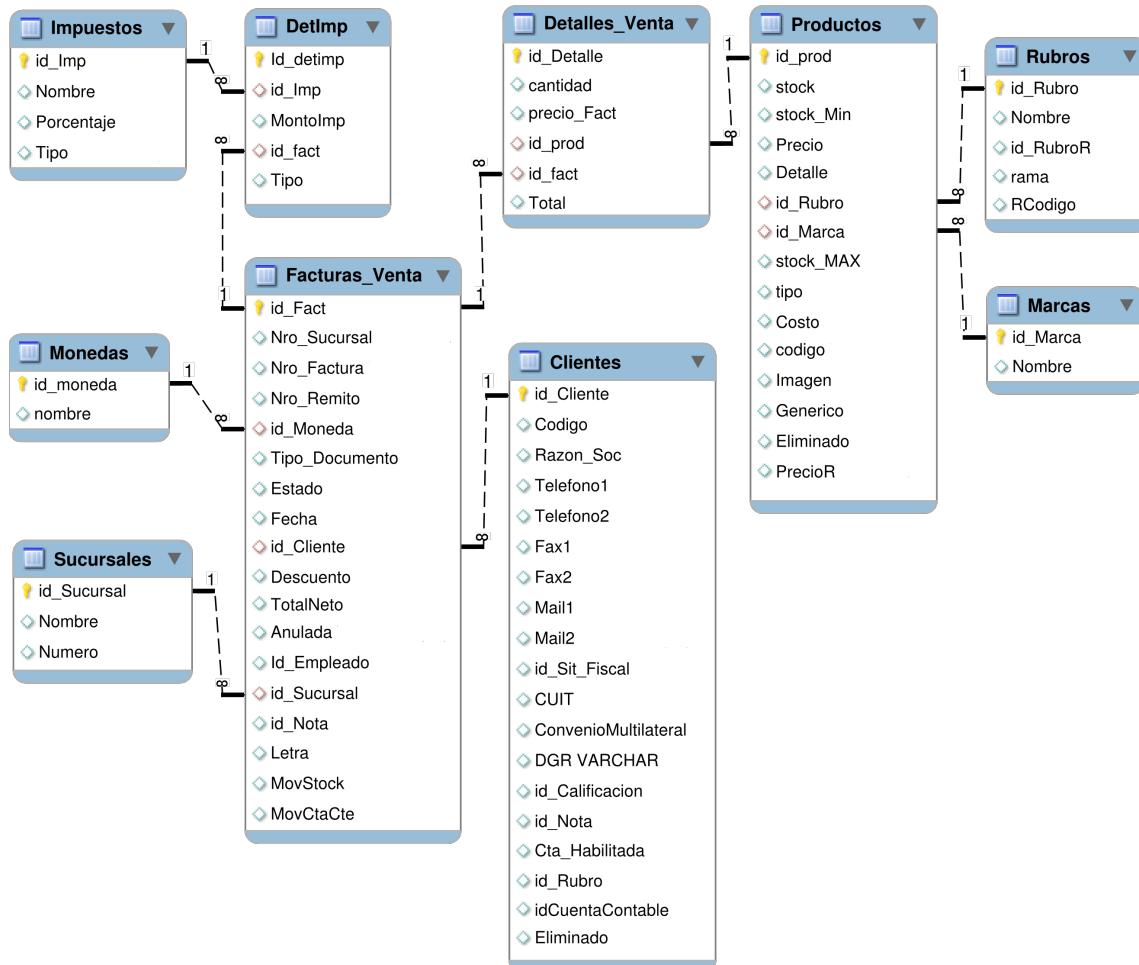
2.2) Mapeo

En este paso debemos examinar los Data Sources e indentificar sus características propias, y asegurarnos que los Data Sources disponibles contengan los datos requeridos.

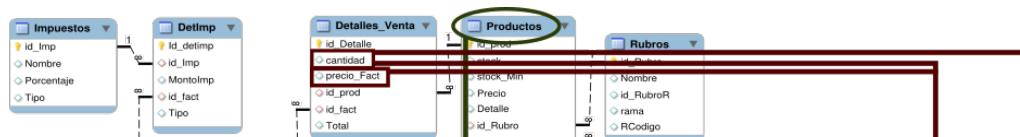
Luego, debemos establecer cómo serán obtenidos los elementos que hemos definido en el Modelo Conceptual, estableciendo de esta manera una correspondencia directa entre elementos del Modelo Conceptual y Data Sources.

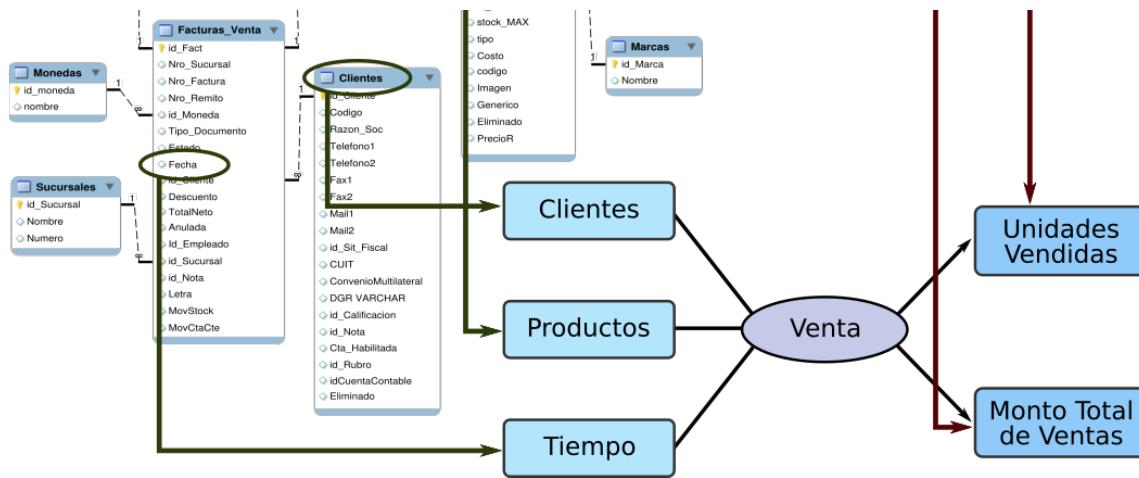
Caso práctico

En el Data Source de la empresa analizada, el proceso de venta está representado por el siguiente Diagrama de Entidad Relación (representa la información a través de Entidades, Relaciones, Cardinalidades, Claves, Atributos y Jerarquías de generalización):



A continuación, se expondrá el Mapeo entre los dos Modelos:





El Mapeo realizado es el siguiente:

- La Perspectiva **Productos** se relaciona con tabla **Productos**.
 - La Perspectiva **Clientes** se relaciona con la tabla **Ciudades**.
 - La Perspectiva **Tiempo** se relaciona con el campo **fecha** de la tabla **Facturas_Venta**, debido a que es la fecha principal en el proceso de venta.
 - El Indicador **Unidades Vendidas** se relaciona con el campo **cantidad** de la tabla **Detalles_Venta**, quedando la fórmula de cálculo como sigue:
 - ▶ **SUM(cantidad)**
 - El Indicador **Monto Total de Ventas** se relaciona con los campos **cantidad** y **precio_Fact** de la tabla **Detalles_Venta**. quedando la fórmula de cálculo como sigue:
 - ▶ **SUM(cantidad * precio_Fact)**

2.3) Granularidad

Una vez que se han establecido el Mapeo con los Data Sources, se deben seleccionar los campos que contendrá cada Perspectiva, ya que a través de estos se analizarán los Indicadores.

Para ello, basándose en el Mapeo establecido en el paso anterior, se debe presentar a l@s usuari@s los datos de análisis disponibles para cada Perspectiva. Es muy importante conocer en detalle qué significa cada campo y/o valor de los datos encontrados en los Data Sources, por lo cual, es conveniente investigar su sentido, ya sea a través de diccionarios de datos, reuniones con l@s encargad@s del sistema, análisis de los datos propiamente dichos, etc.

Luego de exponer frente a l@s usuari@s los datos existentes, explicando su significado, valores posibles y características, est@s deben decidir cuáles son los que consideran relevantes para consultar los Indicadores y cuales NO.

Con respecto a la Perspectiva **Tiempo**, es muy importante definir los períodos mediante los cuales se agregarán los datos. Sus campos posibles pueden ser: día de la semana, quincena, mes, trimestres, semestre, año, etc.

Se debe prestar especial atención al momento de seleccionar los campos que integrarán cada Perspectiva, ya que son estos campos los determinarán la Granularidad de los datos en el DW.

Caso práctico

De acuerdo al Mapeo realizado, se analizaron los campos que constituyen cada tabla a la que se hace referencia a través de dos métodos diferentes. Primero se inspeccionó la base de datos intentando intuir los significados de cada campo, y luego se consultó quién es administrador del sistema para indagar acerca de una serie de aspectos que NO estaban claros.

En este caso, los nombres de los campos eran bastante explícitos, pero aún así fue necesario investigarlos para evitar cualquier tipo de inconvenientes.

Con respecto a la Perspectiva **Clientes**, los datos disponibles son los siguientes:

- **id_Cliente**: es la clave primaria de la tabla **Clientes**, y representa únicamente a un cliente en particular.
- **Codigo**: representa el código del cliente, este campo es calculado de acuerdo a una combinación de las iniciales del nombre del cliente, el grupo al que pertenece y un número incremental.
- **Razon_Soc**: nombre o razón social del cliente.
- **Telefono1**: número de teléfono del cliente.
- **Telefono2**: segundo número telefónico del cliente.
- **Fax1**: número de fax del cliente.
- **Fax2**: segundo número de fax del cliente.
- **Mail1**: dirección de correo electrónico del cliente.

- **Mail2**: segunda dirección de correo del cliente.
- **id_Sit_Fiscal**: representa a través de una clave foránea el tipo de situación fiscal que posee el cliente. Por ejemplo: Consumidor Final, Exento, Responsable No Inscripto, Responsable Inscripto.
- **CUIT**: número de C.U.I.T. (Código Único de Identificación Tributaria) del cliente.
- **ConvenioMultilateral**: indica si el cliente posee o no convenio multilateral.
- DGR: número de D.G.R. (Dirección General de Rentas) del cliente.
- **id_Clasificación**: representa a través de una clave foránea la clasificación del cliente. Por ejemplo: Muy Bueno, Bueno, Regular, Malo, Muy Malo.
- **id_Nota**: representa a través de una clave foránea una observación realizada acerca del cliente.
- **Cta_Habilitada**: indica si el cliente posee su cuenta habilitada.
- **id_Rubro**: representa a través de una clave foránea el grupo al que pertenece el cliente. Por ejemplo: Bancos, Construcción, Educación Privada, Educación Pública, Particulares.
- **idCuentaContable**: representa la cuenta contable asociada al cliente, la cual se utilizará para imputar los movimientos contables que este genere.
- **Eliminado**: indica si el cliente fue eliminado o NO. Si fue eliminado, no figura en las listas de clientes actuales. Es una baja lógica.

En la Perspectiva **Productos**, los datos que se pueden utilizar son los siguientes:

- **id_prod**: es la clave primaria de la tabla "Productos", y representa únicamente a un producto en particular.
- **stock**: stock actual del producto.
- **stock_min**: stock mínimo del producto, se utiliza para emitir un alerta si el stock actual está cerca de este valor o es menor.
- **Precio**: precio de venta del producto.
- **Detalle**: nombre o descripción del producto.
- **id_Rubro**: representa a través de una clave foránea el rubro al que pertenece el producto.
- **id_Marca**: representa a través de una clave foránea la marca a la que pertenece el producto.
- **stock_MAX**: stock máximo del producto. Al igual que "stock_min", se utiliza para dar alertas del nivel de stock actual.
- **tipo**: clasificación del producto. Por ejemplo: Producto, Servicio, Compuesto.
- **Costo**: costo del producto.
- **codigo**: representa el código del producto, este campo es calculado de acuerdo a una combinación de las iniciales del nombre del producto, el rubro al que pertenece y un número incremental.
- **Imagen**: ruta a un archivo de tipo imagen que representa al producto. Este campo NO es utilizado actualmente.
- **Generico**: indica si el producto es genérico o no.
- **Eliminado**: indica si el producto fue eliminado o NO. Si fue eliminado, NO se visualiza en las listas de productos actuales. Es una baja lógica.

- **PrecioR:** precio de lista del producto.

Con respecto a la Perspectiva **Tiempo**, que es la que determinará la granularidad del DW, los datos más típicos que pueden emplearse son los siguientes:

- Año
- Semestre
- Cuatrimestre
- Trimestre
- Número de mes
- Nombre del mes
- Quincena
- Semana
- Número de día
- Nombre del día

Una vez finalizada la recolección de la información pertinente y consultados l@s usuari@s sobre los datos que consideraban de interés para analizar los Indicadores, los resultados obtenidos fueron los siguientes:

Perspectiva **Clientes**:

- **Razon_Soc** de la tabla **Clientes**. Ya que este hace referencia al nombre del cliente.

Perspectiva **Productos**:

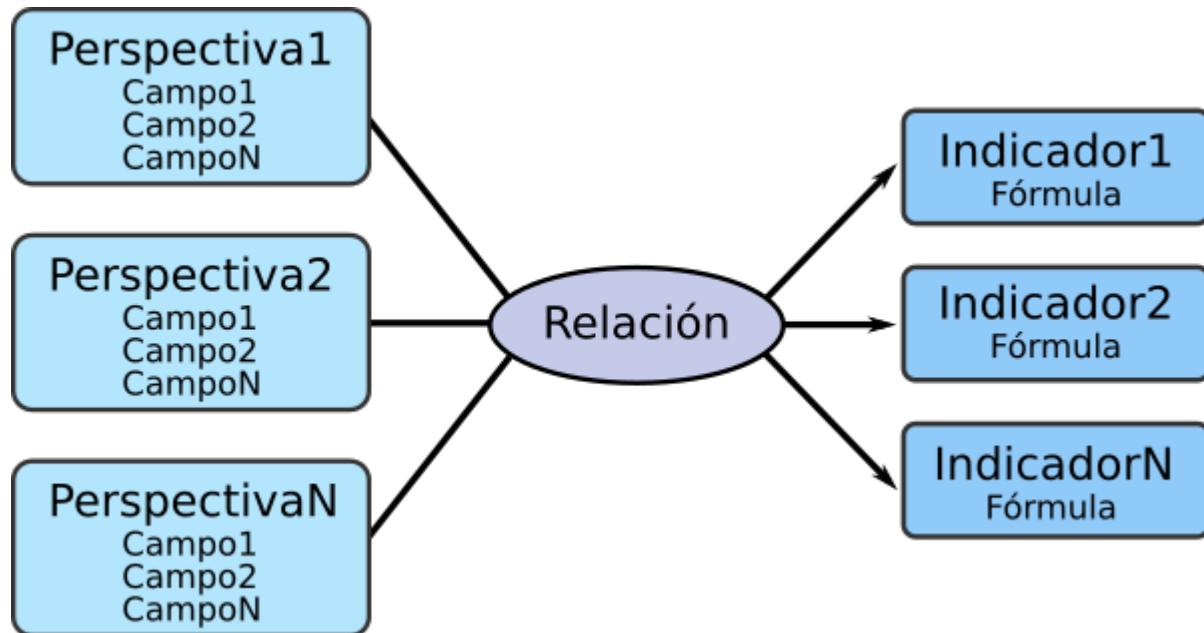
- **detalle** de la tabla **Productos**. Ya que este hace referencia al nombre del producto.
- **Nombre** de la tabla **Marcas**. Ya que esta hace referencia a la marca a la que pertenece el producto. Este campo es obtenido a través de la unión con la tabla **Productos**.

Perspectiva **Tiempo**:

- **Mes**, referido al nombre del mes.
- **Trimestre**.
- **Año**.

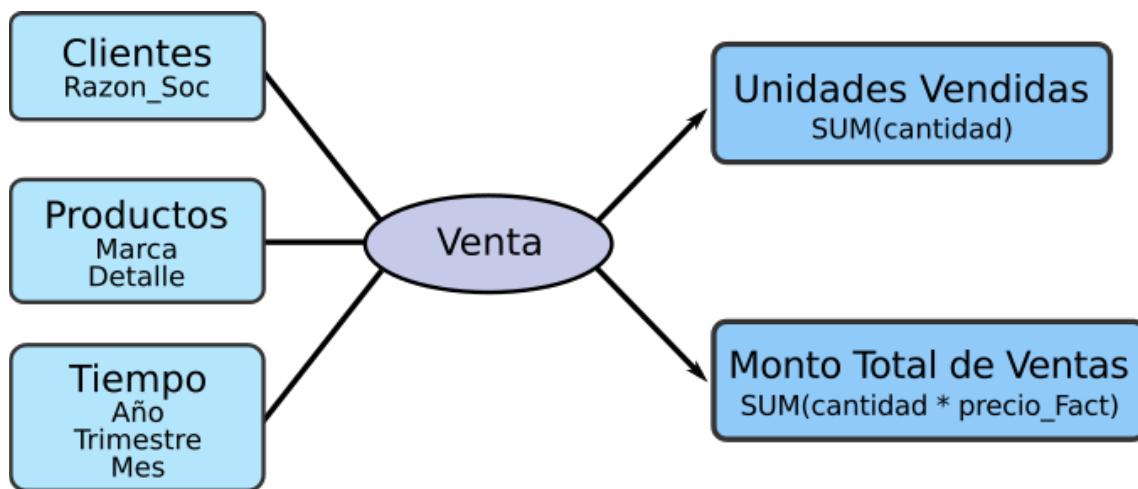
2.4) Modelo Conceptual Ampliado

En este paso, y con el fin de graficar los resultados obtenidos en los pasos anteriores, se ampliará el Modelo Conceptual, colocando debajo de cada Perspectiva los campos seleccionados y debajo de cada Indicador su respectiva fórmula de cálculo.



Caso práctico

Se ampliará el diseño del Diagrama Conceptual:



Paso 3) Modelo Lógico del DW

A continuación, se confeccionará el Modelo Lógico de la estructura del DW, teniendo como base el Modelo Conceptual que ya ha sido creado.

Un Modelo Lógico es la representación de una estructura de datos, que puede procesarse y almacenarse en algún SGBD.

Inicialmente, se definirá el tipo de Modelo Lógico que se utilizará y luego se diseñarán las tablas de Dimensiones y de Hechos con sus respectivas relaciones.

3.1) Tipología

Debemos seleccionar el tipo de Esquema que mejor se adapte a los requerimientos y necesidades de l@s usuari@s. El Modelo Lógico seguirá este esquema.

Es muy importante definir objetivamente si se empleará un Esquema en Estrella, Copo de Nieve o Constelación, ya que esta decisión afectará considerablemente la elaboración del Modelo Lógico.

Caso práctico

Se ha seleccionado el Esquema en Estrella ya que cumple con los requerimientos planteados y es simple de implementar y comprender.

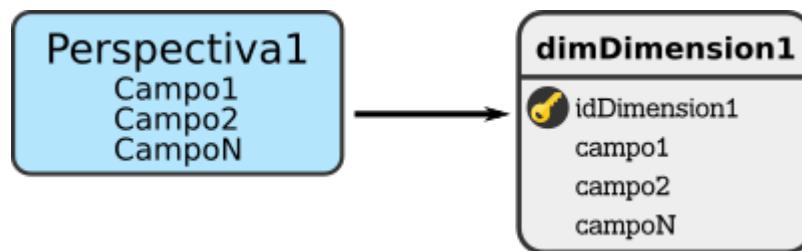
3.2) Tablas de Dimensiones

En este paso diseñaremos las tablas de Dimensiones que formarán parte del DW.

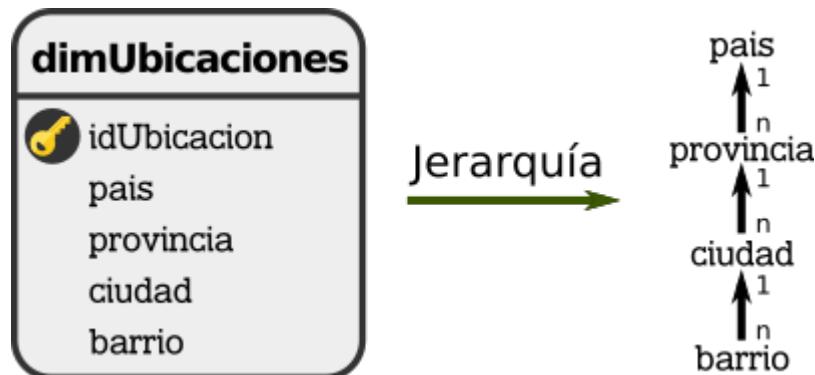
Para los tres tipos de Esquemas, cada Perspectiva definida en el Modelo Conceptual se constituirá en una tabla de Dimensión. Para ello, a partir de cada Perspectiva y sus campos debe realizarse el siguiente proceso:

- Se elegirá un nombre que identifique la tabla de Dimensión.
- Se añadirá un campo que represente su clave principal.
- Se redefinirán los nombres de los campos si es que no son lo suficientemente intuitivos.

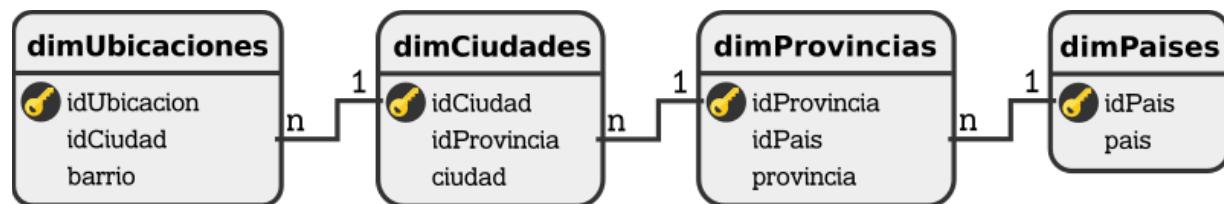
Gráficamente:



Para los Esquemas Copo de nieve, cuando existan Jerarquías dentro de una tabla de Dimensión, esta tabla deberá ser normalizada. Por ejemplo, se tomará como referencia la siguiente tabla de Dimensión y su respectivas relaciones padre-hijo entre sus campos:



Entonces, al normalizar esta tabla se obtendrá:

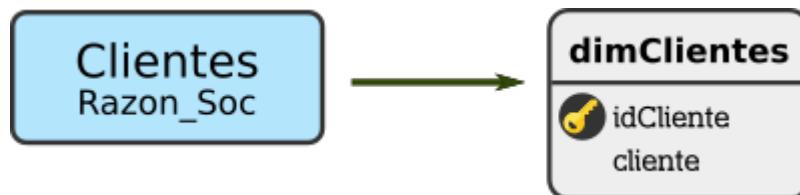


Caso práctico

A continuación, se diseñarán las tablas de Dimensiones.

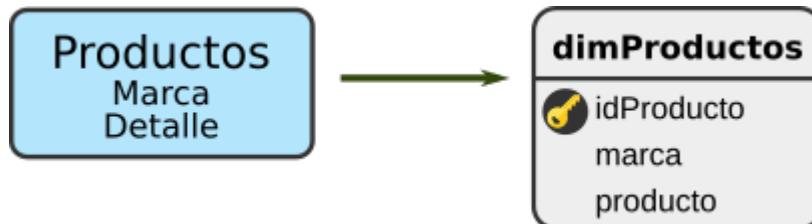
Perspectiva **Ci**entes:

- La nueva tabla de Dimensión tendrá el nombre **dimCi
- Se le agregará una clave principal con el nombre **idCi
- Se modificará el nombre del campo **Razon_Soc** por **cliente**.****



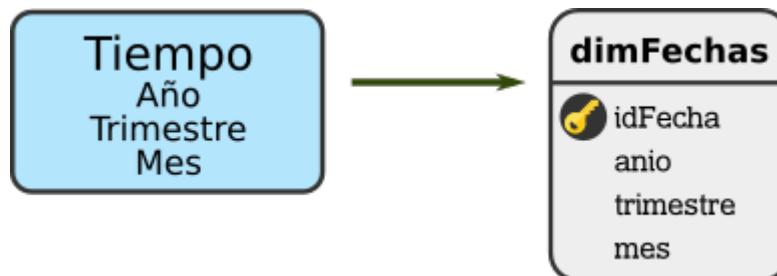
Perspectiva **Pro**ductos:

- La nueva tabla de Dimensión tendrá el nombre **dimPro**ductos.
- Se le agregará una clave principal con el nombre **idPro**ducto.
- Se modificará el nombre del campo **Marca** por **marca**.
- Se modificará el nombre del campo **Detalle** por **producto**.



Perspectiva **Ti**empo:

- La nueva tabla de Dimensión tendrá el nombre **dimFechas**.
- Se le agregará una clave principal con el nombre **idFecha**.
- Se modificará el nombre del campo **Año** por **anio**.
- Se modificará el nombre del campo **Trimestre** por **trimestre**.
- Se modificará el nombre del campo **Mes** por **mes**.



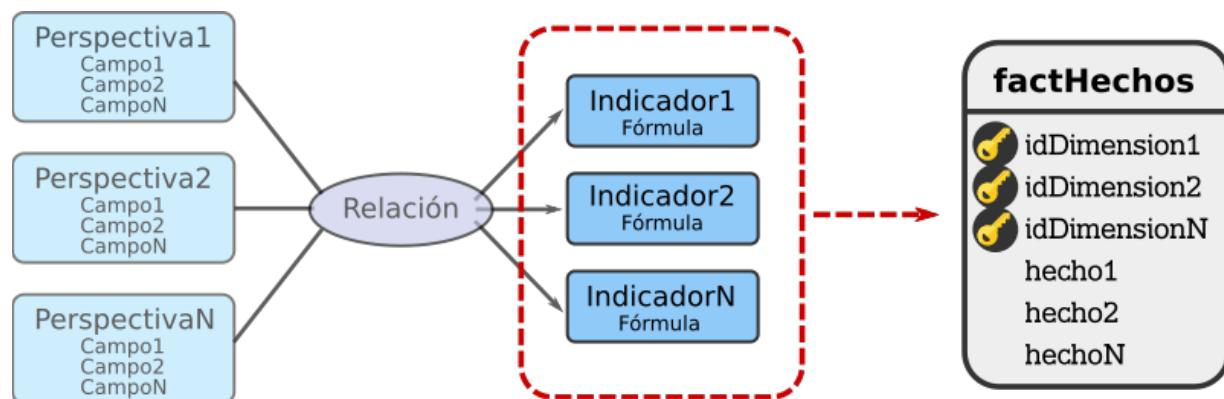
3.3) Tablas de Hechos

En este paso, se definirán las tablas de Hechos.

Esquemas en Estrella y Copo de Nieve

Para los Esquemas en Estrella y Copo de Nieve, se realizará lo siguiente:

- Se le deberá asignar un nombre a la tabla de Hechos que represente la información que contiene, área de investigación, negocio enfocado, etc.
- Se definirá su clave primaria, que se compone de la combinación de las claves primarias de cada tabla de Dimensión relacionada.
- Se crearán tantos campos de Hechos como Indicadores se hayan definido en el modelo conceptual y se les asignará un nombre.



Esquemas Constelación

Para los Esquemas Constelación se realizará lo siguiente:

- Las tablas de Hechos se deben confeccionar teniendo en cuenta el análisis de las preguntas realizadas por l@s usuari@s en pasos anteriores y sus respectivos Indicadores y Perspectivas.
- Cada tabla de Hechos debe poseer un nombre que la identifique y su clave debe estar formada por la combinación de las claves de las tablas de Dimensiones relacionadas.

Al diseñar las tablas de Hechos, se deberá tener en cuenta:

Caso 1:

Si en dos o más preguntas de negocio figuran los mismos Indicadores pero con diferentes Perspectivas de análisis, existirán tantas tablas de Hechos como preguntas cumplan esta condición. Por ejemplo:

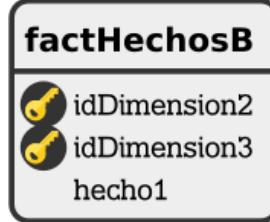
Analizar el Indicador1 por Perspectiva1 y por Perspectiva2
Analizar el Indicador1 por Perspectiva2 y por Perspectiva3

Entonces se obtendrá:

Analizar el **Indicador1** por **Perspectiva1** y por **Perspectiva2**:



Analizar el **Indicador1** por **Perspectiva2** y por **Perspectiva3**:



Caso 2:

Si en dos o más preguntas de negocio figuran diferentes Indicadores con diferentes Perspectivas de análisis, existirán tantas tablas de Hechos como cumplan esta condición. Por ejemplo:

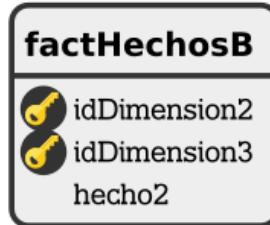
Analizar el **Indicador1** por **Perspectiva1** y por **Perspectiva2**
 Analizar el **Indicador2** por **Perspectiva2** y por **Perspectiva3**

Entonces se obtendrá:

Analizar el **Indicador1** por **Perspectiva1** y por **Perspectiva2**:



Analizar el **Indicador2** por **Perspectiva2** y por **Perspectiva3**:



Caso 3:

Si el conjunto de preguntas de negocio cumplen con las condiciones de los dos puntos anteriores se deberán unificar aquellos interrogantes que posean diferentes Indicadores pero iguales

Perspectivas de análisis, para luego reanudar el estudio de las preguntas. Por ejemplo:

Analizar el Indicador1 por Perspectiva1 y por Perspectiva2

Analizar el Indicador2 por Perspectiva1 y por Perspectiva2

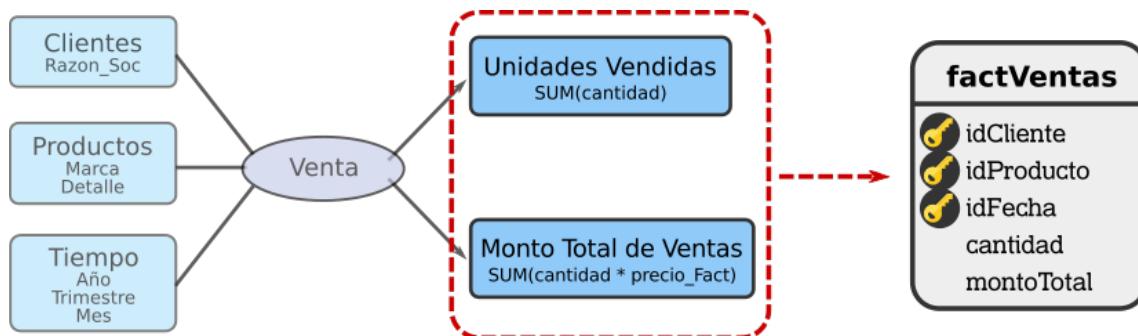
Se unificarán en:

Analizar el Indicador1 y el Indicador2 por Perspectiva1 y por Perspectiva2

Caso práctico

A continuación, se confeccionará la tabla de Hechos:

- La tabla de Hechos tendrá el nombre **factVentas**.
- Su clave principal será la combinación de las claves principales de las tablas de Dimensiones antes definidas: **idCliente**, **idProducto** e **idFecha**.
- Se crearán dos Hechos, que se corresponden con los dos Indicadores:
 - ▶ **Unidades Vendidas** será renombrado como **cantidad** y
 - ▶ **Monto Total de Ventas** será renombrado como **montoTotal**.

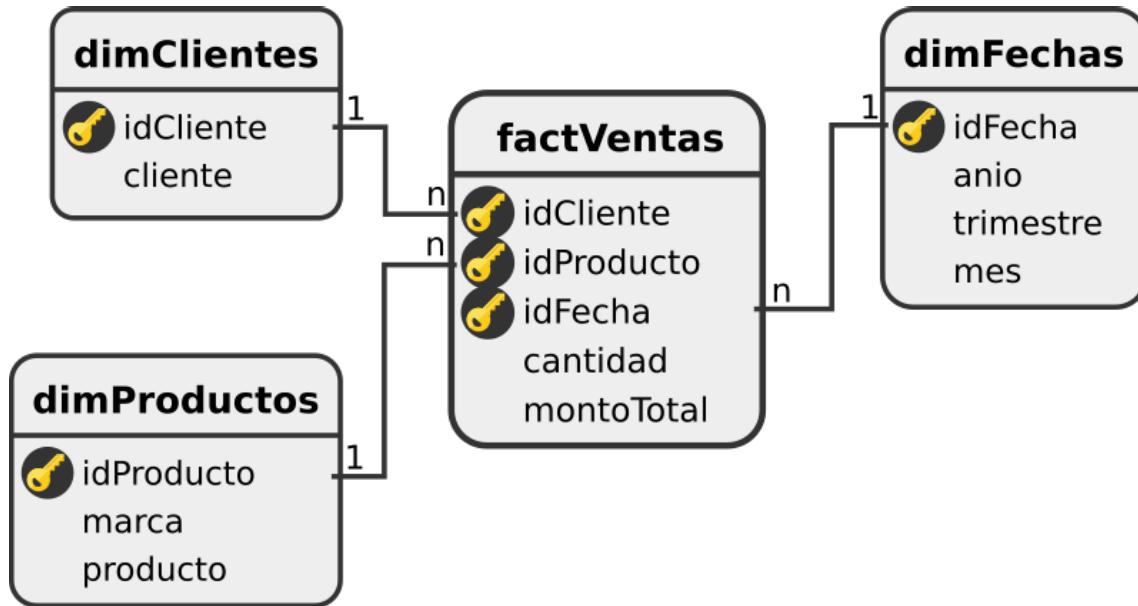


3.4) Uniones

Para los tres tipos de Esquemas, se realizarán las uniones correspondientes entre sus tablas de Dimensiones y sus tablas de Hechos.

Caso práctico

Se realizarán las uniones necesarias según corresponda:



Paso 4) Integración de Datos

Una vez construido el Modelo Lógico, se deberá proceder a poblarlo con datos, utilizando técnicas de limpieza y calidad de datos, procesos ETL, etc.

Luego se definirán las reglas y políticas de actualización, así como también los procesos que la llevarán a cabo.

4.1) Carga Inicial

Debemos en este paso realizar la Carga Inicial del DW, poblando el modelo construido en pasos anteriores. Para lo cual debemos llevar adelante una serie de tareas básicas, tales como asegurar la limpieza y calidad de los datos, procesos ETL, etc.

En muchos casos, las tareas antes mencionadas tienen una lógica compleja. Afortunadamente, en la actualidad existen muchas herramientas de software que se pueden emplear y que nos facilitan en gran parte el trabajo.

Se debe evitar que el DW sea cargado con Missing Values (valores faltantes), Outliers (datos anómalos) o faltos de integridad; se deben establecer condiciones y restricciones para asegurar que solo se utilicen los datos de interés.

Cuando se trabaja con un Esquema Constelación, hay que tener presente que varias tablas de Dimensiones serán compartidas con diferentes tablas de Hechos. Puede darse el caso que algunas restricciones aplicadas sobre una tabla de Dimensión para analizar una tabla de Hechos, se contrapongan con otras restricciones o condiciones de análisis de otras tablas de Hechos.

Primero se cargarán los datos de las Dimensiones y luego los de las tablas de Hechos. En el caso en que se esté utilizando un Esquema Copo de Nieve, cada vez que existan Jerarquías de Dimensiones, se comenzarán cargando las tablas de Dimensiones del nivel más general al más detallado. Esto se debe a la existencia de claves foráneas y se realiza para evitar problemas de rechazo de datos por parte del SGBD.

Concretamente, en este paso se deberán registrar en detalle las acciones llevadas a cabo con los diferentes Software de Integración de datos. Por ejemplo, es común que sistemas ETL trabajen con *Pasos* y *Relaciones*, en donde cada *Paso* realiza una tarea en particular del Proceso ETL y cada *Relación* indica hacia donde debe dirigirse el flujo de datos.

Se debe especificar:

- qué hace el proceso en general y luego
- qué hace cada *Paso* y *Relación*.

Es decir, se partirá de lo más general y se irá a lo más específico, para obtener de esta manera una visión general y detallada de todo el proceso.

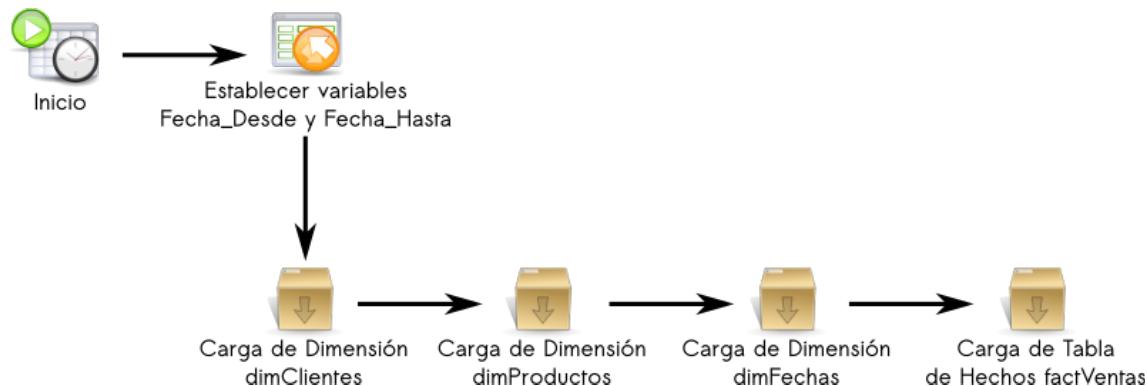
Es importante tener presente, que al cargar los datos en las tablas de Hechos pueden utilizarse preagregaciones con el mismo nivel de granularidad o con niveles menores.

Caso práctico

Para simplificar la aplicación del ejemplo, el caso práctico solo se centrará en los aspectos más importantes del Proceso ETL, obviando entrar en detalle de cómo se realizan algunas funciones y/o pasos.

Proceso ETL Principal

El **Proceso ETL principal** planteado para la Carga Inicial es el siguiente:

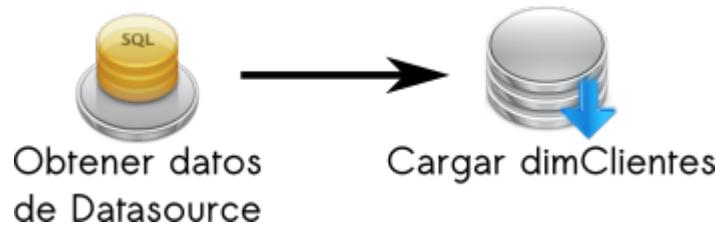


Las tareas que lleva a cabo este proceso son:

- **Inicio**: inicia la ejecución de los pasos en el momento en que se le indique.
- **Establecer variables Fecha_Desde y Fecha_Hasta**: establece dos variables globales que serán utilizadas posteriormente por algunos *Pasos*.
 - ▶ Para la variable **Fecha_Desde** se obtiene el valor de la fecha en que se realizó la primera venta.
 - ▶ Para la variable **Fecha_Hasta** se obtiene el valor de la fecha actual.
- **Carga de Dimensión dimClientes**: ejecuta el contenedor de *Pasos* que cargará la tabla de Dimensión dimClientes (más adelante se detallará).
- **Carga de Dimensión dimProductos**: ejecuta el contenedor de *Pasos* que cargará la tabla de Dimensión dimProductos (más adelante se detallará).
- **Carga de Dimensión dimFechas**: ejecuta el contenedor de *Pasos* que cargará la tabla de Dimensión dimFechas (más adelante se detallará).
- **Carga de Tabla de Hechos factVentas**: ejecuta el contenedor de *Pasos* que cargará la tabla de Hechos factVentas (más adelante se detallará).

Carga de Dimensión dimClientes

A continuación, se especificarán las tareas llevadas a cabo por **Carga de Dimensión dimClientes**. Este *Paso* es un *Contenedor de Pasos*, así que incluye las siguientes tareas:



- **Obtener datos de Datasource**: obtiene a través de una consulta SQL los datos del Datasource necesarios para cargar la tabla de Dimensión dimClientes.

Se tomará como fuente de entrada la tabla **Clients** del Data Source mencionado

anteriormente.

Se consultó con l@s usuari@s y se averiguó que deseaban tener en cuenta solo aquellos clientes que NO estén eliminados y que tengan su cuenta habilitada.

Es importante destacar que aunque existían numerosos movimientos de clientes que en la actualidad NO poseen su cuenta habilitada o que figuran como eliminados, se decidió NO incluirlos debido a que el énfasis está puesto en analizar los datos a través de aquellos clientes que NO se encuentren en estas condiciones.

Los clientes eliminados son referenciados mediante el campo **Eliminado**; el valor **1** indica que éste fue eliminado y el valor **0** que aún permanece vigente. Cuando se examinaron los registros de la tabla, para muchos clientes NO había ningún valor asignado para este campo, lo cual, según comunicó el encargado del sistema, se debía a que este campo se agregó poco después de haberse creado la base de datos inicial, razón por la cual existían Missing Values (valores faltantes). Además, comentó que en el sistema, si un cliente posee en el campo **Eliminado** el valor **0** o un Missing Value (valor faltante), es considerado como vigente.

Con respecto a la cuenta habilitada, el campo del Data Source que le corresponde es **Cta_Habilitada**; el valor **0** indica que su cuenta NO está habilitada y el valor **1** que su cuenta sí está habilitada.

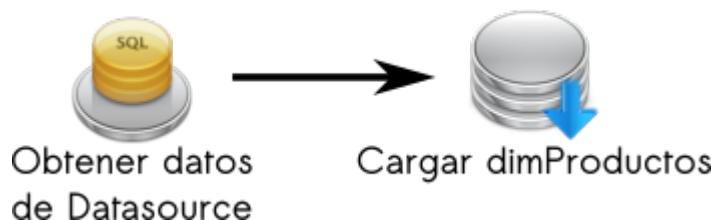
A continuación se expondrá la sentencia SQL configurada en este *Paso*:

```
SELECT
    Clientes.id_Cliente AS idCliente,
    Clientes.Razon_Soc AS cliente
FROM
    Clientes
WHERE
    (Clientes.Eliminado <> 1)
AND (Clientes.Cta_Habilitada <> 0)
ORDER BY
    Clientes.id_Cliente,
    Clientes.Razon_Soc
```

- **Cargar dimClientes:** almacena en la tabla de Dimensión dimClientes los datos obtenidos en el *Paso* anterior.

Carga de Dimensión dimProductos

Se especificarán las tareas llevadas a cabo por **Carga de Dimensión dimProductos**. Este *Paso* es un *Contenedor de Pasos*, así que incluye las siguientes tareas:



- **Obtener datos de Datasource:** obtiene a través de una consulta SQL los datos del Data Source necesarios para cargar la Dimensión dimProductos.

Las fuentes que se utilizarán, son las tablas **Productos** y **Marcas**.

En este caso, aunque existían productos eliminados, los usuarios decidieron que esta condición NO fuese tomada en cuenta, ya que había movimientos que hacían referencia a productos con este estado.

Es necesario realizar una unión entre la tabla **Productos** y **Marcas**, por lo cual se debió asegurar que ningún producto hiciera mención a alguna marca que NO existiese.

La sentencia SQL configurada en este paso es la siguiente:

```
SELECT
    Productos.id_prod AS idProducto,
    Marcas.Nombre AS marca,
    Productos.Detalle AS producto
FROM
    Productos LEFT OUTER JOIN
    Marcas ON Productos.id_Marca = Marcas.id_Marca
ORDER BY
    Marcas.Nombre,
    Productos.Detalle
```

- **Cargar dimProductos:** almacena en la tabla de Dimensión dimProductos los datos obtenidos en el *Paso* anterior.

Carga de Dimensión dimFechas

A continuación, se especificarán las tareas llevadas a cabo por **Carga de Dimensión dimFechas**. Este *Paso* es un *Contenedor de Pasos*, así que incluye las siguientes tareas:



Para generar la tabla de Dimensión dimFechas (la cual debe estar presente en todo DW) existen herramientas y utilidades de Software que proporcionan diversas opciones para su confección.

Lo que se hizo, fue confeccionar un Procedure (procedimiento) que trabaja de la siguiente manera:

- 1) Recibe como parámetros los valores de **Fecha_Desde** y **Fecha_Hasta**.
- 2) Recorre una a una las fechas que se encuentran dentro de este intervalo.
- 3) Analiza cada fecha y realiza una serie de operaciones para crear los valores de los campos de la tabla de la Dimensión dimFechas:

```

"idFecha";"anio";"trimestre";"mes"
20170101; 2017; "1er Trim"; "Enero"
20170102; 2017; "1er Trim"; "Enero"
20170103; 2017; "1er Trim"; "Enero"
20170104; 2017; "1er Trim"; "Enero"
20170105; 2017; "1er Trim"; "Enero"
20170106; 2017; "1er Trim"; "Enero"
20170107; 2017; "1er Trim"; "Enero"
20170108; 2017; "1er Trim"; "Enero"
20170109; 2017; "1er Trim"; "Enero"

```

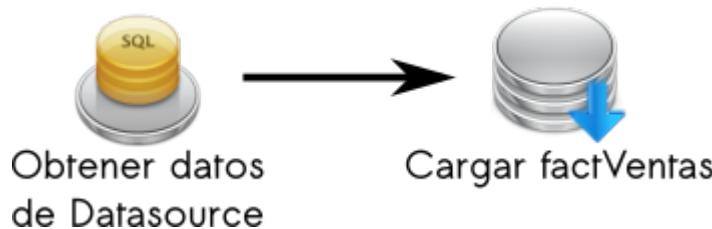
- ▶ idFecha = YEAR(fecha)*10000 + MONTH(fecha)*100 + DAY(fecha)
- ▶ anio = YEAR(fecha)
- ▶ trimestre = CASE WHEN QUARTER(fecha) = 1 then '1er Tri' ... END
- ▶ mes = CASE WHEN MONTH(fecha) = 1 then 'Enero' ... END

- 4) Inserta los valores obtenidos en la tabla de Dimensión dimFechas.

La clave principal idFecha es un campo numérico representado por el formato **yyyymmdd**.

Carga de Tabla de Hechos factVentas

A continuación, se especificarán las tareas llevadas a cabo por **Carga de Tabla de Hechos factVentas**. Este *Paso* es un *Contenedor de Pasos*, así que incluye las siguientes tareas:



- **Obtener datos de Datasource:** obtiene a través de una consulta SQL los datos del Data Source necesarios para cargar la tabla de Hechos factVentas.

Para la confección de la tabla de Hechos, se tomaron como fuente las tablas **Facturas_Ventas** y **Detalles_Venta**. Al igual que en las tablas de Dimensiones, se analizaron las condiciones que deben cumplir los datos para considerarse de interés. En este caso, se trabajará solamente con aquellas facturas que NO hayan sido anuladas.

Se investigó al respecto, y se llegó a la conclusión de que el campo que brinda dicha información es **Anulada** de la tabla **Facturas_Ventas** y si el mismo posee el valor **1** significa que efectivamente fue anulada.

Otro punto a tener en cuenta, es que la fecha se debe convertir al formato numérico **yyyymmdd**.

Se decidió aplicar una preagregación a los Hechos que formarán parte de la tabla de Hechos; es por esta razón que se utilizará la cláusula **GROUP BY** para agrupar

todos los registros a través de las claves primarias de esta tabla.

La sentencia SQL configurada en este *Paso* es la siguiente:

```
SELECT
    Facturas_Venta.idCliente AS idCliente,
    Detalles_Venta.id_pod AS idProducto,
    ((YEAR(Facturas_Venta.Fecha)*10000) +
     (MONTH(Facturas_Venta.Fecha)*100) +
     (DAY(Facturas_Venta.Fecha))) AS idFecha,
    SUM(Detalles_Venta.cantidad) AS cantidad,
    SUM(Detalles_Venta.cantidad * Detalles_Venta.precio_Fact) AS montoTotal
FROM
    Facturas_Venta INNER JOIN
    Detalles_Venta ON Facturas_Venta.id_Fact = Detalles_Venta.id_fact
WHERE
    (Facturas_Venta.Anulada <> 1)
GROUP BY
    Facturas_Venta.id_Cliente,
    Detalles_Venta.id_prod,
    Facturas_Venta.Fecha
ORDER BY
    idFecha,
    idCliente,
    idProducto
```

- **Cargar factVentas:** almacena en la tabla de Hechos factVentas los datos obtenidos en el *Paso* anterior.
-

4.2) Actualización

Cuando se haya ejecutado la carga inicial del DW, se deben establecer las políticas y estrategias de actualización periódica.

Entonces, se deben llevar a cabo las siguientes acciones:

- Determinar el proceso de limpieza de datos y calidad de datos, definir los procesos ETL, etc., que deberán realizarse para actualizar los datos del DW.
- Especificar de forma general y detallada las acciones que deberá realizar cada Software.

Caso práctico

Las políticas de Actualización que se han convenido con l@s usuari@s son las siguientes:

- La información se refrescará: **todos los días a las 00:00hs.**
- Los datos de las tablas de Dimensiones **dimProductos** y **dimClientes** serán cargados siempre en su totalidad.
- Los datos de la tabla de Dimensión **dimFechas** se cargarán de forma incremental teniendo en cuenta la fecha de la última actualización.
- Los datos de la tabla de Hechos **factVentas** que corresponden al último mes (30 días) a partir de la fecha actual, serán reemplazados cada vez.
- Estas acciones se realizarán durante un período de prueba, para analizar cuál es la manera más eficiente de generar las actualizaciones, basadas en el estudio de los cambios que se producen en los Data Sources y que afectan al contenido del DW.

Para evitar que se extienda demasiado la aplicación del ejemplo, el caso práctico solo incluirá lo que debería realizar el proceso ETL para actualizar el DW.

El proceso ETL para la actualización del DW es similar al de **Carga Inicial**, con las siguientes diferencias:

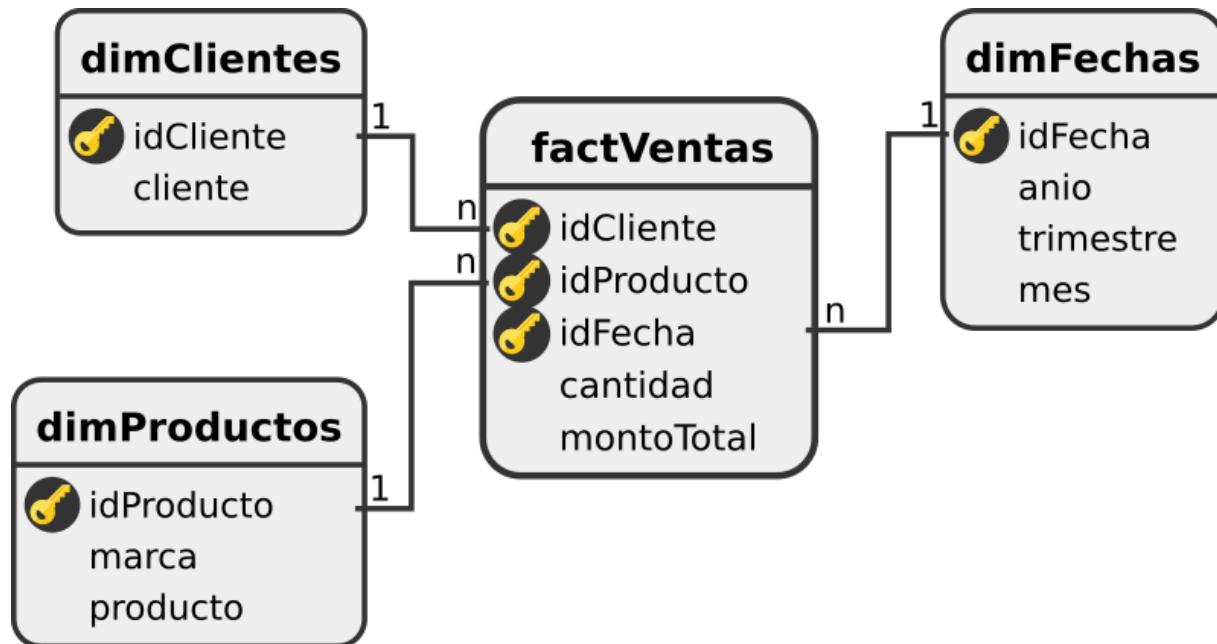
- Inicio: iniciará la ejecución de los pasos todos los días a las doce de la noche.
- Establecer variables Fecha_Desde y Fecha_Hasta:
 - ▶ La variable **Fecha_Desde** obtendrá el valor resultante de restarle a la fecha actual treinta días.
 - ▶ La variable **Fecha_Hasta** obtendrá el valor de la fecha actual.
- Carga de Dimensión dimClientes: a la serie de pasos que realiza esta tarea, se le antepondrá un nuevo paso que borrará los datos que contenga la Dimensión **dimClientes**.
- Carga de Dimensión dimProductos: a la serie de pasos que realiza esta tarea, se le antepondrá un nuevo paso que borrará los datos que contenga la Dimensión **dimProductos**.
- Carga de Dimensión dimFechas: en este paso, se establecerá la variable **Fecha_Desde**, tomando la fecha del último registro cargado en la Dimensión

dimFechas.

- Carga de Tabla de Hechos factVentas:
 - ▶ a la serie de pasos que realiza esta tarea, se le antepondrá un nuevo paso que borrará los datos que contenga la tabla de Hechos **factVentas** en el intervalo entre **Fecha_Desde** y **Fecha_Hasta**.
 - ▶ en el paso Obtener datos de Datasource se modificará la sentencia SQL agregando la siguiente condición:
 - ▶ **WHERE Facturas_Venta.Fecha >= {Fecha_Desde} AND Facturas_Venta.Fecha <= {Fecha_Hasta}**
-

Cubo Multidimensional

Continuando con el ejemplo, se creará un Cubo Multidimensional que estará basado en el modelo lógico diseñado en el caso práctico de la metodología HEFESTO:

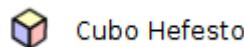


La creación de este Cubo tiene las siguientes finalidades:

- Ejemplificar la creación de Cubos Multidimensionales.
- Propiciar la correcta distinción entre Hechos de una tabla de Hechos e Indicadores de un Cubo.
- Propiciar la correcta distinción entre campos de una tabla de Dimensión y Atributos de un Cubo.

Pasos básicos

1) Se creará un Cubo Multidimensional llamado **Cubo Hefesto**:



2) Se indicará que la tabla de Hechos utilizar es **factVentas**:



3) Se creará un Dimensión para analizar los clientes. Su nombre será **Dimensión Clientes**.

- Se añadirá una Jerarquía. Su nombre será **Jerarquía Clientes**.
- Se indicará que **Jerarquía Clientes** estará basada en los campos de la tabla de Dimensión **dimClientes**.
- Se añadirá a la **Jerarquía Clientes** el Atributo **Cliente**, que estará basado en el campo **cliente**.



4) Se creará un Dimensión para analizar los productos. Su nombre será **Dimensión Productos**.

- Se añadirá una Jerarquía. Su nombre será **Jerarquía Productos**.
- Se indicará que **Jerarquía Productos** estará basada en los campos de la tabla de Dimensión **dimProductos**.
- Se añadirá a la **Jerarquía Productos** el Atributo **Marca**, que estará basado en el campo **marca**.
- Se añadirá a la **Jerarquía Productos** el Atributo **Producto**, que estará basado en el campo **producto**. En donde existe una relación padre-hijo entre **marca** y **producto** respectivamente.



5) Se creará un Dimensión para analizar el tiempo. Su nombre será **Dimensión Fechas**.

- Se añadirá una Jerarquía. Su nombre será **Jerarquía Fechas**.
- Se indicará que **Jerarquía Fechas** estará basada en los campos de la tabla de Dimensión **dimFechas**.
- Se añadirá a la **Jerarquía Fechas** el Atributo **Año**, que estará basado en el campo **anio**.
- Se añadirá a la **Jerarquía Fechas** el Atributo **Trimestre**, que estará basado en el campo **trimestre**. En donde existe una relación padre-hijo entre **anio** y **trimestre** respectivamente.
- Se añadirá a la **Jerarquía Fechas** el Atributo **Mes**, que estará basado en el campo **mes**. En donde existe una relación padre-hijo entre **trimestre** y **mes** respectivamente.





6) Se creará un Indicador, llamado **Unidades Vendidas** que se calculará de la siguiente manera:

- SUM(cantidad)**

7) Se creará un Indicador, llamado **Monto Total de Ventas** que se calculará de la siguiente manera:

- SUM(montoTotal)**



Capítulo 6: Diseño

- Planificación
 - Performance
 - Elección de Columnas
 - Relación Muchos a Muchos
 - Claves Subrogadas
 - Slowly Changing Dimensions
 - ▶ SCD Tipo 1: Sobreescribir
 - ▶ SCD Tipo 2: Añadir fila
 - ▶ SCD Tipo 3: Añadir columna
 - ▶ SCD Tipo 4: Historial separado
 - ▶ SDC Tipo 6: Híbrido
 - Dimensiones Degeneradas
 - Impactos
-

Planificación



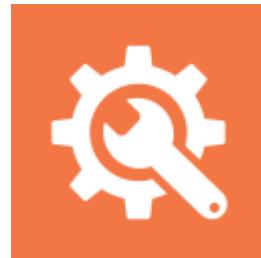
Divers@s autor@s resaltan la importancia del factor tiempo en la construcción de un DW, por lo cual se ha considerado interesante exponer tres frases seleccionadas al respecto:

- “El 70% del tiempo total dedicado al proyecto se insume en definir el problema y en preparar la tabla de datos”.
- “Estime el tiempo necesario, multiplíquelo por dos y agregue una semana de resguardo”.
- “Regla 90 – 90”: el primer 90% de la construcción de un sistema absorbe el 90% del tiempo y esfuerzo asignados; el último 10% se lleva el otro 90% del tiempo y esfuerzo asignado.

Al exponer estas frases, se quiere subrayar que NO debe subestimarse el esfuerzo de la creación de un DW ya que esto puede conducir al fracaso del proyecto.

Performance

En el momento de diseñar los procesos de Integración de Datos, el enfoque NO debe estar centrado en solucionar el problema en el menor tiempo posible.



Los procesos de Integración de Datos insumen gran esfuerzo de construcción ya que deben ser, eficaces y eficientes desde el inicio; esto se debe a que con el transcurrir del tiempo se manejará un gran volumen de datos, ergo, el espacio en disco se tornará un recurso fundamental y los tiempos de procesamiento y acceso a los datos serán esenciales, y más aún si el DW es tomado como un sistema de misión crítica.

Resulta fundamental, entonces, la correcta selección y configuración del SGBD que dará soporte al DW, y la elección de las mejores estrategias para modelar las estructuras de datos.

En cuanto a la configuración del SGBD, los puntos más importantes a tener en cuenta son:

- Configuración y asignación de Buffers Caché
- Indexación
- Algoritmos de acceso
- Particionamiento
- Distribución

Para mejorar la performance del DW, se deben considerar las siguientes acciones:

- Seleccionar cuidadosamente los tipos de datos, por ejemplo, para valores enteros pequeños conviene utilizar `tinyint` o `smallint` en lugar de `int`, con el fin de NO desperdiciar espacio. Esto toma vital importancia cuando se aplica en las claves primarias, debido a que formarán parte de la tabla de Hechos que es la que conforma el volumen del DW, además de que toda clave primaria tiene asociado un índice que la implementa.
- Utilizar Claves Subrogadas.
- Utilizar técnicas de indexación.
- Utilizar técnicas de particionamiento.
- Crear diferentes niveles de agregación.
- Utilizar técnicas de administración de datos en memoria Caché.
- Utilizar distribución de datos.
- Utilizar técnicas de multiprocesamiento distribuido, con el objetivo de agilizar la obtención de resultados, a través de la realización de procesos en forma concurrente.

Elección de Columnas



Cuando se seleccionan los campos que integrarán el DW, se debe tener en cuenta lo siguiente:

- Se deben descartar aquellos campos cuyos valores tengan muy poca variabilidad.
- Se deben descartar los campos que tengan valores diferentes para cada objeto, por ejemplo el número de documento/cédula de identidad, cuando se analizan personas.
- En los casos en que no existan Jerarquías dentro de alguna tabla de Dimensión, en la cual la cantidad de registros que posee la misma son demasiados, es conveniente, conjuntamente con l@s usuari@s, definirlas. Pero, si llegase a suceder que no se encontrase ningún criterio por el cual jerarquizar los campos, es una buena práctica crear Jerarquías propias. El objetivo de llevar a cabo esta acción, es la de poder dividir los registros en grupos, propiciando de esta manera una exploración más amena y controlable. Para exemplificar este punto, se utilizará como referencia la tabla de Dimensión de la siguiente figura. La misma no posee ninguna Jerarquía definida y la cantidad de registros con que cuenta son cientos:



Entonces, lo que se realizará, es crear una nueva Jerarquía a partir de los campos disponibles.

Se añadirá a la tabla el nuevo campo **letra**, el cual representará la primera letra del campo **producto**. Por ejemplo, si el valor de **producto** es **Lapicera**, la **letra** será **L**; si es **Cartuchera** será **C**, etc.

El resultado será el siguiente:



Además, se pueden aplicar algunas de las acciones que se expondrán a continuación sobre los valores de los campos que se incluirán en el DW:

- Factorizar: se utiliza para descomponer un valor en dos o más componentes. Por ejemplo, el campo **codigo** perteneciente a un producto está formado por tres identificadores separados por guiones medios, que representan su rubro, marca y tipo (*idRubro-idMarca-idTipo*), entonces este campo puede factorizarse y separarse en tres valores independientes: **idRubro**, **idMarca** e **idTipo**.
- Estandarizar: se utiliza para ajustar valores a un tipo de formato o norma preestablecida. Por ejemplo, se puede emplear este método cuando se desea que todos los campos del tipo texto sean convertidos a mayúscula.
- Codificar: es utilizado para representar valores a través de las reglas de un código preestablecido. Por ejemplo, en el campo **estado** se pueden codificar sus valores, **0** y **1**, para transformarlos en **Apagado** y **Encendido** respectivamente.
- Discretizar: es empleado para convertir un conjunto continuo de valores en uno discreto. Por ejemplo, en el campo **intensidad** se pueden codificar los valores menores a 100 como **Baja**; los valores mayores a 100 y menores a 500 como **Media**; y los valores mayores a 500 como **Alta**.

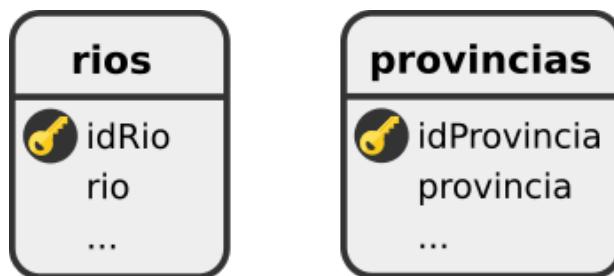
Relación Muchos a Muchos

Se debe evitar mantener en el DW tablas de Dimensiones con relaciones muchos a muchos entre ellas, ya que esta situación puede, entre otros inconvenientes, provocar la pérdida de la capacidad analítica de la información y conducir a una agregación incorrecta de los datos.

Para explicar esta problemática, se tomará como ejemplo la relación existente entre ríos y provincias, es decir:

- Una provincia tiene uno o más ríos, y un río pertenece a una o más provincias.

Se tomará como referencia las siguientes tablas que contienen datos relacionados a ríos y provincias:



Cuando existe este tipo de relación (muchos a muchos) entre dos o más tablas, se pueden realizar diferentes acciones para abordar esta situación. Una posible solución, sería llevar a cabo los siguientes pasos:

- 1) Crear una tabla de Dimensión por cada entidad que pertenece a la relación. Cada una de estas tablas NO debe incluir ninguna correspondencia con las demás. En este caso se crearán dos tablas de Dimensiones, **dimRios** (correspondiente a la entidad **rios**) y **dimProvincias** (correspondiente a la entidad **provincias**).
- 2) Crear otra tabla de Dimensión (en este caso **dimRiosProvincias**) que sea hija de las tablas de Dimensiones recientemente confeccionadas (en este caso **dimRios** y **dimProvincias**), que estará compuesta de los siguientes campos:
 - ▶ Clave principal: dato autoincrementable (en este caso **idRioProvincia**).
 - ▶ Claves foráneas: se deben añadir columnas que refieran la clave principal de las tablas de Dimensiones involucradas. En este caso **idRio** e **idProvincia**.
 - ▶ Otros campos de información adicional.
- 3) Incluir el campo clave principal creado en el paso anterior (**idRioProvincia**) en la tabla de Hechos.

El resultado sería el siguiente:





Otra posible solución sería agregar las dos claves primarias de las tablas de Dimensiones [dimRios](#) y [dimProvincias](#) en la tabla de Hechos.

Existen otras soluciones para solventar esta brecha, pero la primera propuesta posee mucha performance, ya que:

- elimina la relación muchos a muchos,
- solo se necesita un campo clave en la tabla de Hechos, y
- las relaciones entre las tablas resultantes es simple y fácil de visualizar.

La única desventaja es en cuanto a los procesos ETL, ya que se aumenta su complejidad y tiempo de proceso.

Claves Subrogadas



Las claves existentes en los Data Sources se denominan claves naturales; en cambio, las claves subrogadas son aquellas que se definen artificialmente:

- son de tipo numérico secuencial,
- NO tienen relación directa con ningún dato y
- NO poseen ningún significado en especial.

Lo anterior, es solo una de las razones por las cuales utilizar claves subrogadas en el DW, pero se pueden definir una serie de ventajas más:

- Ocupan menos espacio y son más performantes que las tradicionales claves naturales, y más aún si estas últimas son de tipo texto.
- Son de tipo numérico entero (autonumérico o secuencial).
- Permiten que la construcción y mantenimiento de índices sea una tarea sencilla.
- El Data Warehouse NO dependerá de la codificación interna de los Data Sources.
- Si se modifica el valor de una clave en el Data Source, el DW lo tomará como un nuevo elemento, permitiendo de esta manera, almacenar diferentes versiones del mismo dato.
- Permiten la correcta aplicación de técnicas SCD (Dimensiones lentamente cambiantes).

Esta clave subrogada debe ser el único campo que sea clave principal de cada tabla de Dimensión.

Una forma de implementación sería, a través de la utilización de herramientas ETL, mantener una tabla que contenga la clave primaria de la tabla del Data Source y la clave subrogada correspondiente a la Dimensión del DW.

En la tabla de Dimensión Tiempo, es conveniente hacer una excepción y mantener un formato tal como **yyyymmdd**, ya que esto provee dos grandes beneficios:

- Se simplifican los procesos ETL.
- Brinda la posibilidad de realizar particiones de la tabla de Hechos a través de ese campo.

Slowly Changing Dimensions



Las Dimensiones lentamente cambiantes o SCD (Slowly Changing Dimensions) son Dimensiones en las cuales sus datos tienden a modificarse a través del tiempo, ya sea de forma ocasional o constante.

Cuando ocurren estos cambios, se puede optar por seguir una de estas dos opciones:

- Registrar el historial de cambios.
- Reemplazar los valores que sean necesarios.

Inicialmente *Ralph Kimball* planteó tres estrategias a seguir cuando se tratan las SCD: tipo 1, tipo 2 y tipo 3; pero a través de los años la comunidad de personas que se encargaba de modelar bases de datos profundizó las definiciones iniciales e incluyó varios tipos SCD más, por ejemplo: tipo 4 y tipo 6.

A continuación se detallará cada tipo de estrategia SCD:

- SCD Tipo 1: Sobreescribir.
- SCD Tipo 2: Añadir fila.
- SCD Tipo 3: Añadir columna.
- SCD Tipo 4: Historial separado.
- SCD Tipo 6: Híbrido.

Cabe destacar que existe un SCD Tipo 0, que representa el NO tener en cuenta los cambios que pudieran llegar a suceder en los datos de las Dimensiones y por consiguiente NO tomar medidas.

De acuerdo a la naturaleza del cambio se debe seleccionar qué técnica SCD se utilizará; en algunos casos resultará conveniente combinar varias técnicas.

Es importante señalar que si bien hay diferentes maneras de implementar cada técnica, es indispensable contar con claves subrogadas en las tablas de Dimensiones para poder aplicar dichas técnicas.

Al aplicar las diferentes técnicas SCD, en muchos casos se deberá modificar la estructura de la tabla de Dimensión con la que se esté trabajando, por lo cual estas modificaciones son recomendables hacerlas al momento de modelar la tabla; aunque también pueden hacerse una vez que ya se ha modelado y contiene datos; así por ejemplo al añadir una nueva columna se deberán especificar los valores por defecto que adoptarán los registros de la tabla.

Nota:

En todos los ejemplos que se presentarán a continuación se debe tener en cuenta que:

- idProducto es una clave subrogada que es clave principal de la tabla utilizada.
-

SCD Tipo 1: Sobreescribir

Este tipo es el más básico y sencillo de implementar, ya que si bien NO almacena los cambios históricos, tampoco requiere ningún modelado especial y NO necesita que se añadan nuevos registros a la tabla.

En este caso cuando un registro presenta un cambio en alguno de los valores de sus campos, se debe proceder simplemente a actualizar el dato en cuestión, sobreescribiendo el antiguo.

Para exemplificar este caso, se tomará como referencia la siguiente tabla:

idProducto	rubro	tipo	producto
1	Rubro 1	Tipo 1	Producto 1

Ahora, se supondrá que este producto ha cambiado de **rubro**, y ahora ha pasado a ser **Rubro 2**, entonces se obtendrá lo siguiente:

idProducto	rubro	tipo	producto
1	Rubro 2	Tipo 1	Producto 1

Usualmente este tipo es utilizado en casos en donde la información histórica no sea importante de mantener, tal como sucede cuando se debe modificar el valor de un registro porque tiene errores de ortografía.

El ejemplo planteado es solo a fines prácticos, ya que con esta técnica, todos los movimientos realizados de **Producto 1**, que antes pertenecían al **Rubro 1**, ahora pasarán a ser del **Rubro 2**, lo cual creará una gran inconsistencia en el DW.

SCD Tipo 2: Añadir fila

Esta estrategia requiere que se agreguen algunas columnas adicionales a la tabla de Dimensión, para que almacenen el historial de cambios.

Las columnas que suelen agregarse son:

- **fechaInicio**: fecha desde que entró en vigencia el registro actual. Por defecto suele utilizarse una fecha muy antigua, ejemplo: **01/01/1000**.
- **fechaFin**: fecha en la cual el registro actual dejó de estar en vigencia. Por defecto suele utilizarse una fecha muy futurista, ejemplo: **01/01/9999**.
- **version**: número secuencial que se incrementa cada nuevo cambio. Por defecto suele comenzar en **1**.
- **versionActual**: especifica si el campo actual es el vigente. Este valor puede ser en caso de ser verdadero: **true** o **1**; y en caso de ser falso: **false** o **0**.

Entonces, cuando ocurra algún cambio en los valores de los registros, se añadirá una nueva fila y se deberán completar los datos referidos al historial de cambios.

Para ejemplificar este caso, se tomará como referencia la siguiente tabla:

idProducto	rubro	tipo	producto
1	Rubro 1	Tipo 1	Producto 1

A continuación se añadirán las columnas que almacenarán el historial:

idProducto	rubro	tipo	producto	fechaInicio	fechaFin	version	versionActual
1	Rubro 1	Tipo 1	Producto 1	01/01/1000	01/01/9999	1	true

Ahora, se supondrá que este producto ha cambiado de Rubro, y ahora a pasado a ser **Rubro 2**, entonces se obtendrá lo siguiente:

idProducto	rubro	tipo	producto	fechaInicio	fechaFin	version	versionActual
1	Rubro 1	Tipo 1	Producto 1	01/01/1000	06/11/2009	1	false
2	Rubro 2	Tipo 1	Producto 1	07/11/2009	01/01/9999	2	true

Como puede observarse, se lleva a cabo el siguiente proceso:

- Se añade una nueva fila con su correspondiente clave subrogada (**idProducto**).
- Se registra la modificación (**rubro**).
- Se actualizan los valores de **fechaInicio** y **fechaFin**, tanto de la fila nueva, como la antigua (la que presentó el cambio).
- Se incrementa en uno el valor del campo **version** que posee la fila antigua.
- Se actualizan los valores de **versionActual**, tanto de la fila nueva, como la antigua; dejando a la fila nueva como el registro vigente (**true**).

Esta técnica permite guardar ilimitada información de cambios.

SCD Tipo 3: Añadir columna

Esta estrategia requiere que se agregue a la tabla de Dimensión una columna adicional por cada columna cuyos valores se desean mantener en un historial de cambios.

idProducto	rubro	tipo	producto
1	Rubro 1	Tipo 1	Producto 1

Para mantener el histórico de cambios sobre los datos de la columna **rubro** se añadirá la columna **rubroAnterior**:

idProducto	rubro	rubroAnterior	tipo	producto
1	Rubro 1	-	Tipo 1	Producto 1

Ahora, se supondrá que este producto ha cambiado de **rubro**, y ahora ha pasado a ser **Rubro 2**, entonces se obtendrá lo siguiente:

idProducto	rubro	rubroAnterior	tipo	producto
1	Rubro 2	Rubro 1	Tipo 1	Producto 1

Como puede observarse, se lleva a cabo el siguiente proceso:

- En la columna **rubroAnterior** se coloca el valor antiguo.
- En la columna **rubro** se coloca el nuevo valor vigente.

Esta técnica permite guardar una limitada información de cambios.

SCD Tipo 4: Historial separado

Esta técnica se utiliza en combinación con alguna otra y su función básica es almacenar en una tabla adicional los detalles de cambios históricos realizados en una tabla de Dimensión.

Esta tabla histórica indicará por ejemplo qué tipo de operación se ha realizado (**Insert**, **Update**, **Delete**), sobre qué campo y en qué fecha.

El objetivo de mantener esta tabla es el de contar con un detalle de todos los cambios, para luego analizarlos y poder tomar decisiones acerca de cuál técnica SCD podría aplicarse mejor.

Por ejemplo, la siguiente tabla histórica registra los cambios de la tabla de Dimensión **dimProductos**, la cual supondremos emplea el *SCD Tipo 2*:

idProducto rubroCambio tipoCambio productoCambio fecha

1	Insert	-	-	05/06/2000
2	Insert	Insert	-	25/10/2002
3	-	Insert	-	17/01/2005
4	-	-	Insert	18/12/2009

Tomando como ejemplo el primer registro de esta tabla, la información allí guardada indica lo siguiente:

- El día **05/06/2000**, el registro de la tabla de Dimensión **dimProductos** con **idProducto** igual a **1** sufrió un cambio de **rubro**, por lo cual se debió insertar (**Insert**) una nueva fila con los valores vigentes.

SCD Tipo 6: Híbrido

El SCD Tipo 6 se basa en combinar diferentes técnicas SCD, ellas son:

- SCD Tipo 1,
- SCD Tipo 2 y
- SCD Tipo 3.

Y se denomina SCD Tipo 6, simplemente porque:

- $6 = 1 + 2 + 3$.

Dimensiones Degeneradas



El término Dimensión Degenerada, hace referencia a un campo que será utilizado como criterio de análisis y que es almacenado en la tabla de Hechos.

Esto sucede cuando un campo que se utilizará como criterio de análisis posee el mismo nivel de granularidad que los datos de la tabla de Hechos, y que por lo tanto NO se pueden realizar agregaciones a través de este campo. Los números de orden, números de ticket, números de transacción, etc, son algunos ejemplos de Dimensiones degeneradas.

La inclusión de estos campos en las tablas de Hechos, se lleva a cabo para reducir la duplicación y simplificar las consultas.

Se podría plantear la opción de simplemente incluir estos campos en una tabla de Dimensión, pero en este caso estaríamos manteniendo una fila de esta Dimensión por cada fila en la tabla de Hechos, por consiguiente obtendríamos datos duplicados y complejidad adicional.

Impactos



Al implementar un DWH, es fundamental que l@s usuari@s del mismo participen activamente durante todo su desarrollo, debido a que son ell@s l@s que conocen en profundidad su negocio y saben cuáles son los resultados que se desean obtener. Además, es precisamente en base a la utilización que se le de, que el DW madurará y se adaptará a las situaciones cambiantes por las que atraviese la empresa. L@s usuari@s, al trabajar junto a l@s desarrollador@s y analistas podrán comprender más en profundidad sus propios sistemas operacionales, con todo lo que esto implica.

Con la implementación del DW, los procesos de toma de decisiones serán optimizados, al obtener información correcta al instante en que se necesita, evitando pérdidas de tiempo y anomalías en los datos. Al contar con esta información, l@s usuari@s tendrán más confianza en las decisiones que tomarán y por ello, poseerán una base sustentable para justificarlas.

Usualmente, los DW integrarán Data Sources de diversas áreas y sectores de la empresa. Esto tendrá como beneficio contar con una sola fuente de datos centralizada y común para tod@s l@s usuari@s. Esto posibilitará que en las diferentes áreas se compartan los mismos datos, lo cual conducirá a un mayor entendimiento, comunicación, confianza y cooperación entre las mismas.

El DW introducirá nuevos conceptos tecnológicos y de Business Intelligence, lo cual requerirá que se aprendan nuevas técnicas, herramientas, métodos, destrezas, formas de trabajar, etc.

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Sin obra derivada 4.0](#)

Bernabeu R. Dario | García Mattío Mariano