Sistema DataWarehousing

Carga y control de Calidad

INFORME PRINCIPAL

Raquel Abella, Lucía Cóppola, Diego Olave

Proyecto de Grado de la Carrera Ingeniería en Computación

Facultad de Ingeniería
Universidad de la República,1999

Raquel Abella
Lucía Cóppola
Diego Olave

Tutor: Raúl Ruggia

RESUMEN DE CONTENIDO

INTRODUCCION

Contexto
Objetivos del Proyecto
Experiencia Anterior
Pricipales Aportes
Visión Global del Sistema
Organización del Informe
Cómo leer el Informe

INTRODUCCION

CONOCIMIENTO EXISTENTE

Introducción al Datawarehousing
Calidad de Datos
Arquitectura de DW
Diseño de DW

CONOCIMIENTO EXISTENTE

SISTEMA DESARROLLADO

Planteo del Sistema
Estrategia de Desarrollo
Descripción Técnica del Desarrollo

SISTEMA DESARROLLADO

CONCLUSIONES

Conclusiones Finales
Agradecimientos
Bibliografia

CONCLUSIONES

DESARROLLO SISTEMA DW

DataMart: Presupuesto
Requerimientos **Análisis**
Diseño **Implementación**
DataMart: Bedelía
Requerimientos **Análisis**
Diseño **Implementación**
Apéndice: Técnica para Identificación de datos Requeridos

APARTADO 1

METADATA DEL SISTEMA DW

Analisis General
Mapeos Modelos Conceptuales - Modelos Físicos
Análisis Detallado
Mapeos Modelos de Datos Seleccionados - Modelos Físicos
Base de Datos del DW
Pseudocódigo Procesos de Carga

APARTADO 2

MANUAL DE USUARIO

Manual de usuario del sistema desarrollado

APARTADO 3

CONTENIDO

1 INTRODUCCIÓN 6

1.1 CONTEXTO 7
1.2 EXPERIENCIA ANTERIOR 8
1.3 OBJETIVOS DEL PROYECTO 8
1.4 PRINCIPALES APORTES 9
1.5 VISIÓN GLOBAL DEL SISTEMA 10
1.6 ORGANIZACIÓN DEL INFORME 12
1.7 CÓMO LEER EL INFORME 14

2 CONOCIMIENTO EXISTENTE 16

2.1 INTRODUCCIÓN AL DATA WAREHOUSING 17
2.1.1 Data warehouse 17
2.1.2 DATA WAREHOUSING 19
2.2 CALIDAD DE DATOS 23
2.2.1 Un enfoque práctico 24
2.2.2 Evolución 25
2.2.3 Soluciones 26
2.2.4 Herramientas 28
2.2.5 Calidad en un DATA WAREHOUSE 31
2.3 ARQUITECTURA DEL DATA WAREHOUSE 41
2.3.1 Data warehouse Central 41
2.3.2 Estrategias Data Marts 42
2.4 DISEÑO DE UN DATA WAREHOUSE 44
2.4.1 Introducción 44
2.4.2 'A transformations based approach for designing the Data Warehouse' 44
2.5 REFERENCIAS 48

3 SISTEMA DESARROLLADO 49

3.1 PLANTEO DEL SISTEMA 50
3.2 ESTRATEGIA DE DESARROLLO 51
3.2.1 Desarrollo de un Data Mart 55
3.3 DESCRIPCION TECNICA 62

4 CONCLUSIONES 71

4.1 CONCLUSIONES FINALES 72
4.2 AGRADECIMIENTOS 78
4.3 BIBLIOGRAFIA 79

INTRODUCCIÓN

1

1.1 CONTEXTO

Este proyecto surge como propuesta del área Base de Datos del Instituto de Computación (I.N.CO) de la Facultad de Ingeniería de la Universidad de la República (F.I.). El mismo se enmarca dentro de un conjunto de trabajos presentados a los estudiantes como opciones al proyecto de grado de la carrera Ingeniería en Computación.

La idea es continuar el desarrollo de un Sistema de Data Warehousing (DW) para el I.N.CO. Dividir el proyecto global en dos grupos de desarrollo y así presentar dos proyectos diferentes que apunten al mismo objetivo.

Se presentan entonces, los proyectos 'Sistema de DW: OLAP (OnLine Analytical Process)' y 'Sistema de DW: Carga y Control de Calidad'. Ambos tienen un fin común, continuar el desarrollo del Sistema de DW para el I.N.CO realizado en el año 1997 y seguir investigando en el tema data warehousing.

Este proyecto 'Sistema de DW: carga y control de calidad', apunta a construir el data warehouse relacional, con el cual el proyecto 'Sistema de DW: OLAP' trabajará, para mostrar la información requerida por el usuario.

Las característica principal del sistema a desarrollar, es su naturaleza gerencial. El mismo, facilitará el proceso de toma de decisiones de dos Comisiones dentro de la F.I., la Comisión de Seguimiento de la Carrera y la Comisión de Administración del Presupuesto. Para esto, el sistema permitirá el seguimiento de los estudiantes en sus carreras y el análisis en el tiempo del manejo del presupuesto de la facultad. Esto permitirá detectar tendencias dentro de cada una de éstas áreas, información valiosa al momento de tomar decisiones importantes.

La duración del proyecto es de 9 meses a partir de junio de 1998, con opción a prórroga de 3 meses y es monitoreado por el profesor Ing. Raul Ruggia.

1.2 EXPERIENCIA ANTERIOR

El proyecto continua el camino comenzado en el año 1997 por el proyecto 'Estudio de Técnicas y Software para la construcción de sistemas de DW' [P21]. El mismo, fue la primer experiencia en el desarrollo de un sistema de DW dentro del Area Base de Datos del I.N.CO.

Los puntos a destacar en dicho proyecto son, el desarrollo de un Sistema de DW para el I.N.CO y la metodología utilizada. Las sugerencias presentadas en dicho proyecto, promovieron nuevos estudios e investigaciones. Resaltamos entre ellas, la necesidad de profundizar en la formalización y documentación del desarrollo de un Sistema de DW y la importancia de enriquecer la metodología propuesta con conclusiones extraídas de desarrollos teóricos y otras experiencias prácticas.

Se cita éste trabajo por considerarlo la base a partir de la cual nace la propuesta de ésta proyecto. El cual, mucho tiene que ver con la experiencia realizada, las concluciones obtenidas y las sugerencias propuestas en el proyecto anterior.

1.3 OBJETIVOS DEL PROYECTO

- Realizar el diseño, carga, control de calidad e implementación de un data warehouse relacional.
- Investigar el tema Calidad de Datos en un sistema DW.
- Continuar la experiencia realizada el año 1997 por el proyecto 'Estudio de Técnicas y Software para la construcción de sistemas de DW'.
- Experimentar una nueva forma de desarrollar un Sistema de DW, dividiéndolo de forma horizontal en dos módulos:
  A. Diseño, carga, control de calidad e implementación de un data warehouse relacional
  B. Construcción del Modelo Multidimensional, diseño e implementación de consultas de usuario final.
- Aplicar en el diseño del data warehouse, la teoría desarrollada en el trabajo 'A transformations based aproach for designing the Data Warehouse' presentado como tesis de maestría por la profesora Ing. Adriana Marotta.

1.4 PRINCIPALES APORTES

El principal aporte de éste proyecto es el Sistema de DW construido, la experiencia realizada a lo largo de su desarrollo y su documentación. La estrategia de desarrollo creada, los problemas que se presentaron, cómo se resolvieron y las conclusiones que se obtuvieron.

El **Sistema de DW** construido permite al usuario el análisis de la evolución universitaria de los estudiantes y del presupuesto de la Facultad de Ingeniería. Mediante la formulación de consultas a partir de la información presentada, el usuario podrá visualizar datos valiosos para su área de negocio en particular.

La **Estrategia de Desarrollo** surge a partir del estudio sobre el conocimiento existente en la materia, de la inventiva del grupo y de la experiencia obtenida durante el transcurso del taller.

Dentro del conocimiento existente en la materia se puso énfasis en los temas *Calidad de Datos* y *Diseño de un Data Warehouse Relacional*.
Del estudio sobre *Calidad de Datos*, se destaca la importancia del mismo en un sistema de DW, su evolución y las soluciones existentes al momento.
El estudio sobre el tema *Diseño de un Data Warehouse Relacional*, se basó en el trabajo 'A transformations based aproach for designing the Data Warehouse' presentado por la prof. Ing. Adriana Marotta en su tesis de maestría. La experiencia fue enriquecedora en ambos sentidos ya que éste proyecto aplicó por primera vez el conjunto de primitivas presentadas en el trabajo mencionado. La experiencia realizada actuó de feedback para el desarrollo de dicho trabajo el cual, se desarrolló en paralelo al proyecto.

Se incluyó ambos temas dentro de la estrategia de desarrollo presentada. Respecto del tema *Calidad de Datos*, se logró que la estrategia de desarrollo asegure de alguna manera la calidad en los datos del data warehouse relacional. En cuanto al *tema Diseño de un Data Warehouse Relacional* se aplica el conjunto de primitivas presentadas en el trabajo mencionado al realizar el diseño del data warehouse.

Consideramos interesante, la experiencia realizada en cuanto a la división horizontal del trabajo. La forma en que interactuamos con los estudiantes del otro proyecto, como resultó, qué dificultades se presentaron y como se solucionaron.

1.5 VISIÓN GLOBAL DEL SISTEMA

El sistema de DW construido, constituye uno de los resultados principales de este proyecto. El sistema apoya el proceso de toma de decisones de dos comisiones de la F.I., la Comisión de seguimiento de los estudiantes en la carrera y la Comisión de administración del presupuesto. Se construyen dos data marts identificados como Presupuesto y Bedelía. Cada data mart mantiene información relevante a su área de negocio. El data mart de Presupuesto almacena información sobre el presupuesto de la facultad y el de Bedelía sobre el comportamiento de los estudiantes en sus carreras.

Los datamarts están constituídos por tablas relacionales que forman parte del datawarehouse y cubos multidimensionales. En este proyecto, el termino data mart se refiere al conjunto de tablas relacionales. En estos terminos, el data warehouse puede verse como la integración de ambos data marts, la cual en éste caso es trivial por no existir información compartida.

La figura 1 detalla los componentes del sistema DW construído.

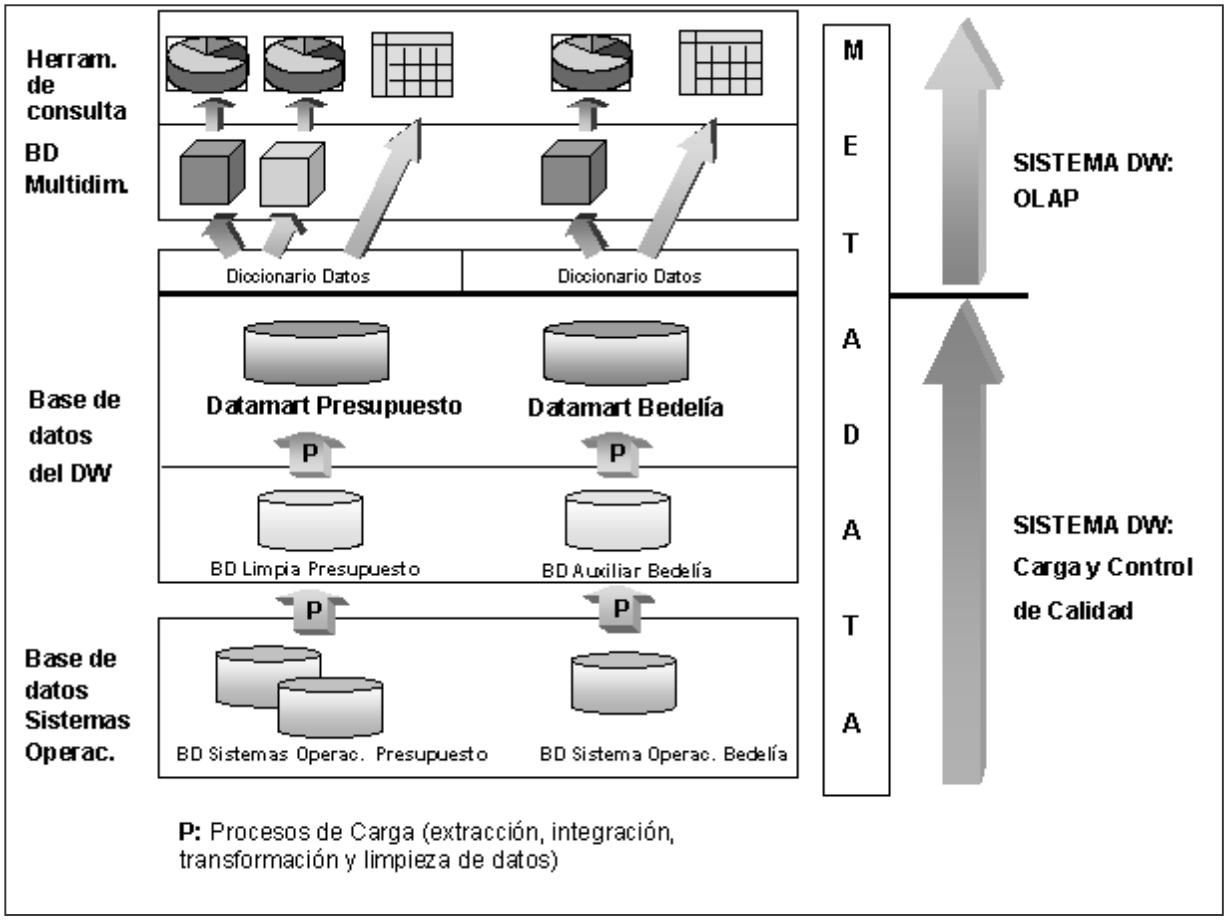


Figura 1: Visión global del Sistema

Base de datos Sistemas Operacionales: Conjunto de bases de datos de los sistemas operacionales que son fuente de datos del data warehouse. Distintos procesos de carga se encargan de extraer los datos de dichas bases y de ingresarlos al data warehouse.

Base de datos del Data Warehouse: Base de datos donde se encuentran los datos de los data marts de Presupuesto y Bedelía. En el primer subnivel de la base de datos del data warehouse están las bases de datos auxiliares. En el caso de presupuesto son bases limpias porque contienen los datos extraídos de las fuentes luego de ser transformados, limpiados e integrados. En el caso de bedelía, se encuentran los datos extraídos de la base de datos fuente de Bedelía. Estos son ingresados a la bd. del datawarehouse mediante el comando LOAD de Oracle, pero no se realiza ningún tipo de transformación o limpieza en éste proceso. Cabe resaltar que el proceso de carga de los datos fuente a la bd. auxiliar de Bedelía no forma parte de éste proyecto.
Distintos procesos de carga son los encargados de llevar los datos desde las bases de datos auxiliares, transformarlos adecuadamente e ingresarlos a su respectivo data mart.
Las bases de datos de los data marts de Presupuesto y Bedelía constituyen los data marts relacionales del sistema, se encuentran en una base Oracle. Los distintos procesos de carga constituyen la carga del sistema, son procesos PLSQL almacenados en el data warehouse. Son llamados desde una aplicación Visual Basic que oficia de interfaz gráfica de integración de procesos de carga.

Dentro del data warehouse se encuentran también los diccionarios de datos ambos data marts. Cada diccionario está constituído por el catálogo que provee la herramienta Impromptu de Cognos.

Bases de datos multidimensionales: Constituídas por diferentes cubos multidimensionales. Los cubos extraen los datos de los data marts que se encuentran en el data warehouse. Toman los datos según lo definido en el diccionario de datos correspondiente a cada data mart.
Los cubos están implementados con la herramienta PowerPlay de Cognos[*www.Cognos.com*].

Herramientas de consulta: Distintas herramientas de usuario final que permiten consultar los cubos multidimensionales o directamente el data warehouse. Las herramientas de consulta utilizadas son las brindadas por el paquete Cognos.

Metadata del sistema: Abarca todos los niveles, documenta la información sobre los datos manejados en todo el sistema.

La división horizontal del trabajo de construír el sistema completo puede verse con facilidad en la figura. Las flechas a la derecha indican el alcance de la tesis 'Sistema DW: Carga y control de calidad' y de la tesis 'Sistema DW: OLAP' [P24]. La primera se ocupa de construir el data warehouse y los procesos de carga. La segunda se ocupa de construir las bases de datos multidimensionales y configurar las herramientas de usuario final.

1.6 ORGANIZACIÓN DEL INFORME

Capítulo 1: Introducción

Introducción a la tesis 'Sistema DW: Carga y Control de calidad'.

Contexto. Descripción del contexto en el que se desarrolla el proyecto.

Experiencia anterior. Introducción general a la tesis 'Estudio de Técnicas y Software para la construcción de sistemas de Data Warehousing' .

Principales aportes. Puntos importantes a destacar de ésta tesis.

Visión global del sistema. Descripción global del sistema construído.

Organización del informe..

Cómo leer el informe. Breve guía de cómo leer el informe según los intereses del lector.

Capítulo 2: Conocimiento Existente

Brinda una visión general de los temas relevantes al proyecto.

Introducción. Define los conceptos principales en data warehousing.

Calidad de Datos. Estudio del tema calidad de datos en un sistema de DW. Resume la importancia del tema, su evolución, soluciones existentes en el mercado y propone una metodología de trabajo para asegurar la calidad de los datos en un data warehouse.

Arquitectura del Data Warehouse. Describe las posibles arquitecturas de un data warehouse.

Diseño del Data warehouse. Introduce la técnica utilizada para diseñar el data warehouse relacional. Presenta el trabajo de maestría 'A transformations based aproach for designing the Data Warehouse' de la Ing. Adriana Marotta.

Capítulo 3: Sistema Desarrollado

Visión global del desarrollo del proyecto.

Planteo del Sistema. Describe las carácterísticas principales del sistema DW.

Estrategia de Desarrollo. Propone Estrategia de Desarrollo en base al planteo anterior.

Overview del Desarrollo. Diagramas de las etapas principales de la estrategia Propuesta y como éstas se aplicaron en el desarrollo realizado.

Capítulo 4: Conclusiones

Conclusiones Finales. Conclusiones principales del proyecto.

Agradecimientos.

Bibliografía.

Apartado 1: Desarrollo del Sistema de DW

Documentación del desarrollo . Aplicación de estrategia propuesta.

Data Mart: Presupuesto. Documenta el desarrollo del Data Mart de Presupuesto.

Data Mart: Bedelía. Documenta el desarrollo del Data Mart de Bedelía.

Apéndice: Técnica para Identificar Datos Requeridos. Describe la técnica utilizada para la identificación de datos requeridos en el data warehouse.

Apartado 2: Metadata del Sistema de DW

Análisis General. Análisis General de Bases de Datos Fuente

Mapeo: M.Conceptuales-Modelos Físicos. Mapea Modelo Conceptuales de Bases Fuente con Modelo Físico

Análisis Detallado. Análisis detallado de Bases de Datos Fuente

Mapeo: M.Datos Seleccionados. Mapea Modelo de Datos Seleccionados con Modelo Físico

Base de Datos del Data Warehouse. Documenta Base de Datos del Data warehouse

Pseudocódigo Procesos de Carga. Documenta procesos de Carga

Apartado 3: Manual de Usuario

Manual de Usuario del sistema desarrollado. Especifica como utilizar el sistema desarrollado.

1.7 CÓMO LEER EL INFORME

Se divide el informe en cuatro libros para facilitar su lectura:

1. **Informe Principal.**
Informe central del proyecto.
Su lectura brinda una visión general del proyecto pero no documenta al detalle el desarrollo realizado.

2. **Apartado 1. Desarrollo del sistema**
Documenta al detalle el desarrollo realizado. La documentación del desarrollo sigue el orden propuesto en la estrategia de desarrollo planteada en el Informe Principal.

3. **Apartado 2. Metadata del sistema**
Contiene Metadata del sistema desarrollado. Toda información del desarrollo realizado, relevante para el mantenimiento y administración del mismo.

4. **Apartado 3. Manual de usuario del sistema**
Explica cómo utilizar el sistema construído.

Para obtener una visión general del sistema construído referirse al punto 1.5 del capítulo del Informe Principal.

Si no se conoce del tema DW, es fundamental la lectura del capítulo 2 del **Informe Principal**. De todos modos, algunos puntos del capitulo 2 son importantes para comprender el desarrollo realizado. En particular, dentro de la sección 2.2, el punto 'Calidad en un Data Warehouse', fundamenta la postura de la estrategia de desarrollo respecto al tema calidad. Dentro de la sección 2.4, el punto 'Diseño de un Data Warehouse', documenta la técnica utilizada al diseñar cada data mart.

Si se está interesado en conocer la forma de trabajo utilizada, puede referirse a la sección 3.2 'Estrategia de desarrollo'. En la misma se documenta una estrategia para construir un data warehouse, procesos de carga, asegurar la calidad de los datos dadas las condiciones descriptas en la sección 3.1 'Planteo del proyecto'.

Las experiencias de desarrollo de los data marts de Presupuesto y Bedelía están documentadas en el **Desarrollo del Sistema**. Para una buena comprensión de las mismas se recomienda leer las secciones 3.1 y 3.2 del **Informe Principal**.

La metadata del sistema se encuentra en **Metadata del Sistema**. En el **Desarrollo de Sistema** también se encuentra información del tipo Metadata pero relacionada al desarrollo.

El manual de usuario del sistema desarrollado se encuentra en **Manual de Usuario.**

A lo largo del informe se utilizaron los siguientes tips:

1.

Resalta las características prinicipales de la sección.

2.

Comentarios de nuestra experiencia en particular.

ESTADO DEL ARTE

2

2.1 INTRODUCCIÓN AL DATA WAREHOUSING

El siguiente capítulo, delinea las ideas principales que hay detrás del concepto *data warehousing*. Las mismas, fueron recogidas de una serie de libros y artículos que documentamos al final del informe.

2.1.1 DATA WAREHOUSE

'A data warehouse is a copy of transaction data specifically structured for query and analysis' [Ralph Kimball]

La necesidad de obtener información de las bases de datos operacionales para apoyar el soporte de toma de decisiones, nace junto con el desarrollo de las primeras aplicaciones pero su importancia es reconocida mucho tiempo después. El primer intento por satisfacer ésta necesidad fue la construcción de reportes, éstos listaban la información existente en las bases de datos. Luego se utilizaron los extractos, recabando información de más de una base de datos, no pudiéndose resolver los problemas de consistencia de datos e integración que se presentaban. Con el advenimiento del PC, los usuarios creyeron que se independizaban de los desarrolladores pudiendo acceder ellos mismos a las bases de datos para obtener la información de su interés. Pero, con todo el poder que brinda el PC, hubo frustración al descubrir que los datos que llegaban no eran mejores que aquellos con los que se disponía antes.

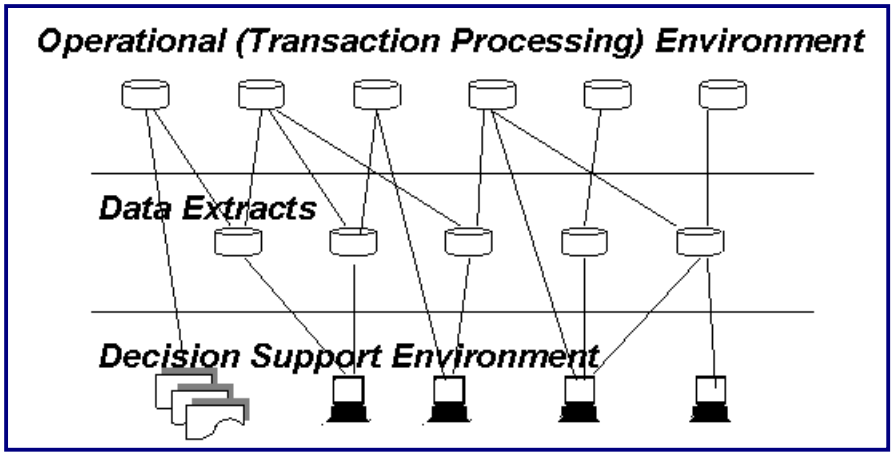


Figura 2: Arquitectura decisional antes del data warehouse

Se acepta finalmente, la incapacidad de las antiguas aplicaciones de satisfacer la necesidad de información que requieren las organizaciones para ser más competitivas y eficientes. Los

sistemas operacionales, satisfacen los requerimientos operativos de la organización pero no están diseñadas para apoyar al proceso de toma de decisiones.
Es aquí cuando se decide separar el procesamiento de datos en dos grandes categorías: *Operacional* y *Decisional*. Aparece entonces una nueva arquitectura, la cual tiene como principal componente, el ***Data Warehouse***.

Las siguientes figuras ilustran las soluciones para la obtención de información decisional, antes y después de la arquitectura data warhouse.

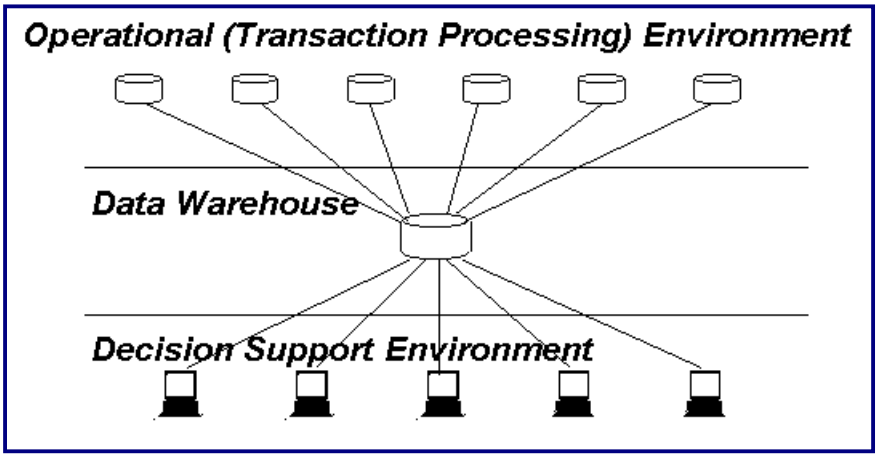


Figura 3: Nueva arquitectura para manejo de información

Un data warhouse es una base de datos diferente a la de los sistemas operacionales en cuanto a que:

- Es orientado a sujetos
- Maneja grandes cantidades de información
- Comprende múltiples versiones de un esquema de bases de datos
- Condensa y agrega información
- Integra y asocia información de muchas fuentes de datos

Un data warehouse organiza y orienta los datos desde la perspectiva del usuario final. Muchos sistemas operativos organizan sus datos desde la perspectiva del negocio, para mejorar la rapidez de acceso y actualización de los datos. Este tipo de organización, no es la más indicada para operaciónes de consulta, típicas en el ambiente decisional.

La mayoría de los data warehouses contienen información histórica que se retira con frecuencia de los sistemas operativos porque ya no es necesaria para las aplicaciones operacionales y de producción. Por la necesidad de administrar tanto la información histórica como la actuales, un data warehouse es mayor que las bases de datos operacionales.

Un data warehouse guarda información histórica y la administra. Como la información histórica muchas veces es manejada por diferentes versiones de esquemas de bases de datos, el data warehouse debe controlar la información originada por bases de datos diferentes.

Un data warehouse condensa y agrega la información para presentarla en forma comprensible a las personas. La condensación y adición es esencial para retroceder y entender la imagen global.

Debido a que las organizaciones han administrado históricamente sus operaciones utilizando numerosas aplicaciones de software y múltiples bases de datos, se requiere de un data warehouse para recopilar y organizar en un solo lugar la información que éstas aplicaciones han acumulado al paso de los años.

2.1.2 DATA WAREHOUSING

' A warehouse is a place, warehousing is a process' [R. Hackathorn]

Un data warehouse es una base de datos, pero por sí sola no significa nada, hay una gran cantidad de procesos detrás de una arquitectura de data warehouse de suma importancia para el mismo. Estos comprenden desde procesos de extracción que estudian y seleccionan los datos fuente adecuados para el data warehouse hasta procesos de consulta y análisis de datos que despliegan la información de una forma fácil de interpretar y analizar.

Procesos básicos de un data warehouse [L2]:

Extracción:
El proceso de extracción consiste en estudiar y entender los datos fuente, tomando aquellos que son de utilidad para el data warehouse.

Transformación:
Una vez que los datos son extraídos, éstos se transforman. Este proceso incluye corrección de errores, resolución de problemas de dominio, borrado de campos que no son de interés, generación de claves, agregación de información, etc.

Carga e Indices:
Al terminar el proceso de transformación, se cargan los datos en el data warehouse.

Chequeo de Calidad:
Una vez ingresada la información al data warehouse, se realizan controles de calidad para asegurar que la misma sea correcta.

Liberación/Publicación:
Cuando la información se encuentra disponible, se le informa al usuarios. Es importante publicar todo cambios que se hallan realizado.

Consulta:
El usuario final debe disponer de herramientas de consulta y procesamiento de datos. Este proceso incluye consultas ad hoc, reportes, aplicaciones DSS, data mining, etc.

Feedback:

Aveces es aconsejable seguir el camino inverso de carga. Por ejemplo, puede alimentarse los sistemas legales con información depurada del data warehouse o almacenar en el mimso alguna consulta generada por el usuario que sea de interés.

Auditoría:

Los procesos de auditoría permiten conocer de donde proviene la información así como también qué cálculos la generaron.

Seguridad:

Una vez construído el data warehouse, es de interés para la organización que la información llegue a la mayor cantidad de usuarios pero, por otro lado, se tiene sumo cuidado de protegerlo contra posibles 'hackers', 'snoopers' o espías. El desarrollo de Internet a incrementado éste dilema.

Respaldo y Recuperación:

Se deben realizar actividades de backup y restore de la información, tanto la almacenada en el data warehouse como la que circula desde los sistemas fuente al data warehouse.

➢ ELEMENTOS BÁSICOS EN UN SISTEMA DW

Un data warehouse es una base de datos, data warhousing es un proceso. Un sistema de DW está compuesto por todas esas cosas y algunas más. A continuación detallamos los componentes fundamentales de un sistema de DW [L2].

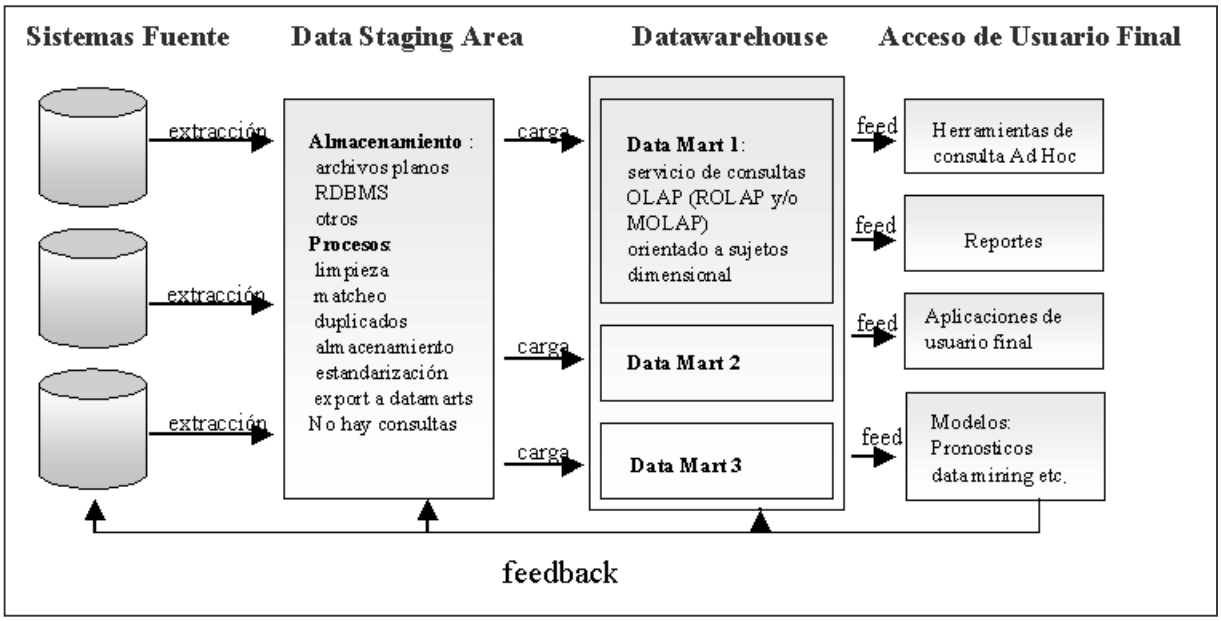


Figura 4: Elementos de un data warehouse

Sistemas Fuente :

Es un sistema operacional, cuya función es capturar las transacciones típicas del negocio. La prioridad en este tipo de sistema la tienen las operaciones de actualización. Las consultas sobre estos sistemas son simples y su demanda está restringida. Mantienen pocos datos históricos.

Data Staging Area :

Area de almacenamiento y un conjunto de procesos que limpian, transforman, combinan, quitan duplicados, archivan y preparan los datos fuente para ser utilizados en el data warehouse.

Servidor de Presentación :

Máquina física en donde se almacenan y organizan los datos del data warehouse para luego ser consultados por reportes, consultas de usuario y otras aplicaciones.

Modelo Dimensional :

Disciplina específica para modelar datos que es una alternativa al modelo entidad-relación.

Data Mart :

Es un conjunto lógico del data warehouse. Un data warehouse se construye como la integración de varios data marts.

Data Warehouse :

Fuente de datos apta para consultas en una organización.

Operational Data Store :

Lugar en donde se integran los sistemas operacionales antes de ser cargados al data warehouse.

OLAP (On-Line Analytic Processing) :

La actividad de consultar y presentar de una forma específica los datos del data warehouse.

ROLAP (Relational OLAP) :

Conjunto de interfaces de usuario y aplicaciones que dan a las bases relacionales un sabor multidimensional.

MOLAP (Multidimensional OLAP) :

Conjunto de interfases, aplicaciones y bases de datos propietarias que tienen un fuerte sabor multidimensional.

Aplicación Usuario Final :

Colección de herramientas que consultan, analizan y presentan la información necesaria para apoyar las necesidades del negocio.

Herramientas de Acceso de Usuario Final :

Son clientes del data warehouse, mantienen una sesión con el servidor de presentación mandándole sentencias SQL. La información obtenida es presentada de forma de pantalla, reporte, gráfico o alguna otra forma de análisis del usuario .

Herramientas de Consultas Ad Hoc :

Es un tipo específico de herramienta de acceso a los datos que invita al usuario a formularse sus propias consultas manipulando directamente las tablas relacionales.

Aplicaciones de Modelado :

Cliente sofisticado del data warehouse, con capacidad analítica. Entre ellas se encuentran las herramientas de data mining.

Metadata :

Toda información en el data warehouse que no sean los datos almacenados.

2.2 CALIDAD DE DATOS

En este capítulo desarrollaremos el tema de calidad de datos, donde se muestra la evolución del tema, los problemas frecuentes en los datos, las características de las soluciones planteadas hasta el momento y las herramientas que han surgido en el mercado. Por último se plantean un proceso de limpieza de datos a realizarse durante la carga de un data warehouse y una guía para mantener la calidad en los datos.

Desde un principio las organizaciones empresariales se han visto atraídas por la evolución tecnológica. En el mundo de hoy es difícil encontrar empresas que no utilicen la tecnología informática como herramienta para manejar su negocio o parte de él, y más difícil aún es encontrar empresas que dejando de lado la tecnología, logren dominar su mercado.

En el comienzo, el uso mayoritario que se les daba a las computadoras, era el de procesamiento de transacciones, de modo de disminuir el tiempo de los procesos internos del negocio. Esta focalización en la automatización de los procesos, sin embargo, ignoraba el valor que tienen los datos dentro de los sistemas, y desaprovechaba la ganancia que brinda su análisis. El rol de la información en crear ventajas competitivas para empresas de negocios y otras compañías es ahora bien conocido, y casi se ha convertido en un axioma dentro del mundo de los negocios.

Sin embargo darle un uso productivo a la información es un objetivo que sólo es alcanzado cuando se logran identificar tres puntos fundamentales: la utilidad que se le quiere dar a la información, la fuente de donde extraerla y la ubicación mas apropiada que tendrá la misma dentro de la infraestructura de la organización, de modo de sacarle su mayor beneficio.

Surge entonces el concepto de *arquitectura de información* que se desarrolla en proyectos como Operational Data Store (ODS), DW y Data Marts, como soluciones a las dificultades de convertir montañas de datos, producidos por sistemas operacionales, en datos productivos para el negocio. A su vez, aparecen tecnologías complementarias que ayudan en los procesos de recolección, manejo, procesamiento y entrega de información útil a partir de la masa inicial de datos “crudos” y “desconectados” generados en los sistemas operacionales.

Desde el momento en que las compañías comienzan a usar DW para compartir datos a través de aplicaciones y unidades de negocio, aparece el concepto que alcanza todos los puntos anteriores, el de **Calidad de Datos**.

La frase siguiente resume la importancia del tema calidad y justifica el estudio del mismo.

Si vamos a confiar en la información de nuestra organización para tomar decisiones de negocio debemos estar seguros que los datos sobre los cuales estamos tomando estas decisiones críticas, son: ***exactos, completos y relevantes****.*

El área de procesamiento de datos es común enfrentarse a los siguientes planteos:

- Acceso a los datos ("Tenemos los datos pero no podemos acceder a ellos").
- Herramientas de consulta ("Quiero un sistema que me muestre qué es importante y por que").
- Calidad de datos ("Sabemos que alguno de nuestros datos no son buenos, por ejemplo, no tenemos una lista centralizada de clientes").

El mercado de base de datos ha respondido a la necesidad de acceso a los datos con la arquitectura Cliente-Servidor, software y hardware especial para data warehouses y familias enteras de esquemas de comunicación para conectar usuarios con sus datos. Por otro lado, el mercado de las herramientas de consultas brinda una gran variedad de productos que van desde herramientas *ad-hoc*, generadores de reportes, ambientes de desarrollo de aplicaciones, hasta herramientas OLAP/ROLAP y datamining. En cambio, el tema calidad de datos, a pesar de su conocida importancia, ha sido dejado de lado en la mayoría de los casos.

*Ni la tecnología de base de datos más sofisticada, ni la interfaz gráfica más seductora puede asegurar el éxito de un data warehouse si los datos son **incorrectos**.*

2.2.1 UN ENFOQUE PRÁCTICO

En el mercado competitivo de hoy en día una compañía exitosa es aquella que construye los vínculos más fuertes con sus clientes, distribuidores, proveedores y cualquier otra entidad clave en su negocio, y conoce todas las relaciones entre estas entidades. Es aquella que tiene la habilidad de extraer información significativa a partir de millones de registros de datos. Es aquella que sabe que la exactitud de la información es un componente crítico para su éxito.

Con información de calidad una compañía puede entender completamente su base de clientes y sus elecciones de compra, el completo espectro de "holdings" de clientes y la historia de costos de su proveedor. Puede realmente focalizarse en el cliente. Con información de calidad una compañía tiene lo que necesita para ser competitiva.

¿ Pero cómo puede un negocio reajustar décadas de viejos datos para obtener la información exacta y temporal necesaria para el complejo ambiente empresarial de nuestros días ?

¿ Cómo se puede obtener información valiosa y consolidada sobre las relaciones entre clientes de legiones de registros, bases de datos, elementos de datos escondidos dentro de campos de texto sin formato ?

¿ Cómo se puede reajustar los antiguos sistemas centralizados en las cuentas a uno centralizado en el cliente ?

La calidad de los datos es el desafío más grande que enfrentan aquellas organizaciones que contemplan el cambio hacia un amplio sistema empresarial o Data Warehouse.

Las compañías están de acuerdo en la importancia del Data Warehouse para el crecimiento de sus negocios. Sin embargo, no es tan claro el camino a tomar para alcanzar el éxito dentro de todas las tendencias de desarrollo.

2.2.2 EVOLUCIÓN

Inicialmente, la idea fue construir un repositorio de datos único, centralizado y de alcance empresarial, el cual combinase todos los datos de todos los sistemas y teóricamente brindase a todos los usuarios acceso a la información apropiada. Aunque algunos proyectos tuvieron éxito, la mayoría fueron extremadamente costosos en dinero, trabajo y tiempo. Uno de los principales temas que surgió, es la inmensa cantidad de datos y metadatos que llegan al warehouse desde distintos sistemas, todos ellos con valores en diferentes formatos, cumpliendo diferentes estándares y teniendo diferentes significados dependiendo del contexto en el cual se originaron.

Respondiendo a estos problemas, la industria decide cambiar la arquitectura del Data Warehousing y desarrolla el concepto de Data Mart. Con esto, múltiples data marts se distribuyen dentro de la organización y son usados por los individuos dentro de cada departamento para analizar su parte del negocio. Este método ayudó a hacer la tecnología warehouse mas accesible consumiendo menos tiempo. Pero poco aporta esta arquitectura al tema de la calidad de los datos. Ahora distintos departamentos acceden a los mismos datos y tratan de interpretarlos simultáneamente desde diferentes contextos. Además el problema que puede ocasionar un error en un dato, ahora impacta aplicaciones a nivel empresarial.

La siguiente lista de problemas en los datos nos ayudará a comprender más la importancia de la calidad de datos y cómo los datos corruptos pueden impactar en las organizaciones [P6][P19][P2].

PROBLEMA	EJEMPLO
Falta de estándares	Los datos se representan en múltiples formatos. Por ejemplo: diferentes formas de escribir un nombre, diferentes formas de escribir fechas.
Información perdida en campos de texto	Información decisiva (identificación de entidades de negocio) es ingresada en campos de texto. Por ejemplo: se tiene un campo descripción y ahí se ingresa la cédula del cliente.
Información no consolidada	Múltiples identificadores de una misma entidad. Por ejemplo: un mismo cliente con distintos códigos.
Comparación compleja y consolidación	Entidades de negocio representadas en una amplia variedad de formas dificulta relacionar todas las instancias o condensarlas en una representación única.

Sorpresas de datos dentro de campos individuales	Valores en los datos que escapan de las descripciones de los campos y de las reglas de negocio. Por ejemplo: nombres comerciales mezclados con nombres personales, indicaciones de ubicación en campos de direcciones, uso inconsistente del espacio en blanco, caracteres especiales y límites de campos (cortar una palabra en un campo y continuar en el siguiente)
Errores	Dentro de las cuales se encuentran errores tipográficos, faltas de ortografías, valores fuera de rango y tipos de datos incorrectos.
Homónimos	Palabras que se escriben igual y tienen diferente significados, a veces hasta sin relación o conflictivos, y su significado correcto depende del contexto.
Datos que faltan o datos invisibles	Datos con estructura y valor apropiados pueden omitir información inadvertidamente. Por ejemplo: la dirección de una persona “J.M. Perez 324” puede ser valida pero si es un edificio se estaría omitiendo el numero de apto.
Datos Fantasma	En ocasiones se ingresan valores especiales indicando que el campo tiene valor desconocido o que no se utiliza más. Por ejemplo: en un campo de fecha se puede encontrar el valor 99/99/9999

2.2.3 SOLUCIONES

Hasta el momento las compañías lo que han hecho para combatir los problemas vistos anteriormente, fue “limpiar” múltiples veces los mismos datos para las diferentes aplicaciones data marts. Por supuesto esto se traduce en esfuerzos duplicados y altos costos y muestra la necesidad de llevar la solución del problema al nivel de la funcionalidad del sistema. Esto es, realizando la limpieza de datos a nivel de los sistemas funcionales, como una utilidad básica de la empresa, y obtener así una eficiencia significativa.

Los proyectos de calidad de datos entonces comienzan a crearse, pero vale la pena analizar en que momento y como. En la mayoría de los casos, la calidad de los datos surge cuando ocurre una crisis y algún proyecto clave corre peligro. Desafortunadamente entonces se desarrollan proyectos especiales que no son permanentes ni consistentes en toda la organización, y por lo general resultan en meses de esfuerzo y miles de dólares en soluciones que no son por naturaleza permanentes.

Tradicionalmente los proyectos de reingeniería de datos carecían del factor principal para su éxito en la compañía: programas para el manejo de la calidad en los datos, un conjunto de procesos tecnológicos consistentes que institucionalizaran la calidad de los datos como un bien estratégico, y procesos para explotar su ventaja competitiva. Los requerimientos de reingeniería son de alcance global por naturaleza. A la vez que las compañías quieren crecer en el mercado global, los procesos deben poder soportar una variedad de datos y valores internacionales. Mientras que existe una necesidad de establecer estándares de información e

identificación de clientes a nivel mundial, existen pocas referencias disponibles para validar elementos de datos internacionales como nombres de corporaciones, títulos, productos y servicios. Una razón más para insistir en un estándar a nivel empresarial para la calidad de datos.

Se llega a la conclusión que el costo asociado a la limpieza de datos no agrega valor, simplemente lleva los datos a un estado de utilidad y relativa confianza. Es necesario mejorar la calidad de los datos, buscar eliminar los costos, problemas y oportunidades perdidas causados por datos de baja calidad.

➢ EDQM: ENTERPRISE DATA QUALITY MANAGEMENT.

*Para que un data warehouse sea existoso dos procesos paralelos e igualmente importantes deben ponerse en práctica: la **limpieza** de los datos y la mejora en la **calidad** de los mismos.*

Para mejorar la calidad de datos se debe prevenir que datos sin calidad entren a la base de datos. Una consecuencia positiva de eliminar la entrada de datos incorrectos es la reducción de los considerables costos que trae aparejado el arreglo de los problemas causados por datos sin calidad. Para establecer procesos de limpieza de datos y mejoramiento de calidad efectivos se debe desarrollar un plan bien definido que incluya un modelo de data warehouse focalizado en la empresa, identifique fuentes de datos significativas y establezca el orden en que se propagarán y transformarán los datos [P2].

La calidad no aparece, requiere planificación. Los datos deben ser nombrados y definidos de manera consistente a lo largo de las áreas de negocio para soportar procesos estratégicos y análisis cruzados de las funcionalidades del negocio. Cuando se consoliden clientes, productos y órdenes de múltiples fuentes de datos se debe desarrollar una arquitectura común con definiciones de datos consistentes, formatos y dominios de valores de datos.

Luego de algunos años de intentar manejar el tema de la calidad de los datos, una nueva disciplina ha emergido dentro del desarrollo de arquitecturas de información para dirigir la necesidad de un correcto manejo de la calidad de los datos. Esta disciplina conocida como Enterprise Data Quality Managment (EDQM) tiene como cometido asegurar la exactitud, temporalidad, relevancia y consistencia de los datos en una organización o en múltiples unidades de negocio dentro una organización, y de esta forma asegurar que las decisiones se basen en información consistente y exacta.

El enfoque efectivo de EDQM puede reducir significativamente los costos de la limpieza de datos. Si los datos estuvieran correctos en las fuentes y en un formato definido a nivel empresarial, no se necesitaría gastar tanto en la limpieza de los datos.

Esta disciplina pretende redefinir los procesos de negocio y crear estándares a nivel de la organización, lo que puede ser extremadamente difícil y costoso si se quiere aplicar en conjunto a toda la organización (big band). Sin embargo, si se comienza de a una unidad de negocio, utilizando un conjunto de estándares consistentes con toda la organización, hasta

abarcar toda la empresa (step by step), se pueden alcanzar los objetivos con menos recursos y aprovechando la experiencia de las unidades que ya lo realizaron.

Desarrollar programas para convertir datos de un formato a otro no es difícil. Diseñar procesos para limpiar y estandarizar los datos en una escala empresarial, incluyendo valores de datos que pueden no ser obvios, es un gran desafío. Afortunadamente la nueva generación de soluciones para el manejo de datos provee herramientas de reingeniería de datos y procesos, junto con programas de conversión para asistir a las compañías en la implementación de programas EDQM.

2.2.4 HERRAMIENTAS

Una vez que los expertos en data warehouses y practicantes descubrieron la necesidad de calidad en los datos la pregunta fue: ¿ cómo alcanzarla ?. Inicialmente la reingeniería de datos consistía en código escrito manualmente e interpuesto entre las fases de extracción de datos y la de carga al data warehouse. Cada proyecto tenía necesidades específicas asociadas con estructuras de datos y contexto específico y es por esto que cada proyecto requería lógicas adaptadas a las necesidades de cada cliente para poder alcanzar la calidad de datos requerida para el data warehouse. La limpieza de datos ha crecido desde este proceso inicial de edición hasta una serie de herramientas de primera y segunda generación que permiten manejar la calidad de los datos.

Iniciativas proactivas para el manejo de la calidad de los datos comienzan en la fase de la entrada de datos. Las validaciones en la entrada de datos son la primer línea de defensa contra datos erróneos, chequean rangos de datos y aseguran que todos los campos requeridos sean llenados durante el proceso de entrada de datos. La ventaja de chequear la calidad de los datos al momento de la entrada es bastante obvia: los errores son "arrancados de cuajo" cuando la información es fresca y de esta forma evitar la necesidad de trabajar doblemente en el futuro. Las validaciones en la entrada de datos no son de todas formas a prueba de tontos, así como un corrector ortográfico no detecta errores gramaticales, el personal de entrada de datos puede ingresar códigos incorrectos en los campos adecuados en el correcto formato y rango y el error no será detectado. Este es el motivo por el cual las herramientas deben ser usadas en conjunto con estándares empresariales. Estos deben ser implementados a un nivel empresarial para asegurar que todos los departamentos involucrados en la entrada de datos usen las convenciones consistentemente.

➢ HERRAMIENTAS DE PRIMERA GENERACIÓN

Las herramientas de primera generación consisten en procesos batch que analizan los datos en busca de defectos y construyen rutinas para reparar los datos al momento que son pasados de la fuente al sistema destino. Desafortunadamente estos procesos basados en el paradigma de crear capas de lógica programática, producen aplicaciones complejas, difíciles de manejar.

Este tipo de producto de primera generación que requiere un enorme compromiso de tiempo y recursos es difícil de mantener, mover o reutilizar; tornándose poco útil a nivel empresarial. Además estas herramientas tienden a ser retrospectivas, detectando y corrigiendo errores en vez de proactivas, previniendo errores y corrigiendo los datos a traves de procesos en la fuente.

Algunas de estas herramientas son [P2]:

- *Herramientas para el formateo de correspondencia*: identifican direcciones erróneas comparándolas con listas de datos válidos y formatean la información de forma que cumplan con normas de distintas organizaciones de servicio postal.
  Ventajas: requieren poco esfuerzo en la preparación de los datos, hacen casi todo el trabajo con poca interacción con el usuario.
  Desventajas: sirven para un único propósito y realizan una función mecánica, y no sacan ningún beneficio de los datos limpios.

- *Herramientas verticales para industrias:* soluciones específicas para una industria (bancos, aseguradoras, etc.), que surgen a partir de exigencias regulatorias. Formatean registros usando terminología estándar de la industria y diseño de archivos. Son efectivas identificando relaciones familiares entre los registros.
  Ventajas: Se ajustan a los requerimientos del negocio del usuario final.
  Desventajas: Son caras debido a que tienen un mercado limitado; la lógica es en general propietaria y difícil de modificar, y requiere programación para adaptarla al cliente; frecuentemente requieren de ayuda de consultores y profesionales expertos en la lógica del negocio para poder ponerlas en marcha, esto implica que la organización debe confiar sus procesos de negocios a expertos externos que una vez terminado el proyecto se llevan el conocimiento consigo.

- *Herramientas de programación*: consisten en algoritmos propietarios que se aplican a casi todas las industrias.
  Desventajas: son herramientas para codificar, no soluciones. Cada problema nuevo requiere generación de código adicional, lo cual agrega capa sobre capa de complejidad y necesita un gran mantenimiento para asegurar el éxito. Estas herramientas no tienen inteligencia propia, y deben ser continuamente mantenidas. En muchos casos se limitan a bases de datos y plataformas específicas. Los usuarios invierten esfuerzo y tiempo considerable en aprender el lenguaje de programación de la herramienta y su metodología, y su uso efectivo requiere de soporte consultor caro.

Muchas compañías han encontrado que el uso de herramientas de primera generación ha agregado complejidad a sus sistemas. Es común encontrar en grandes compañías una gran variedad de soluciones para la limpieza de datos, cada una con sus herramientas y técnicas, las cuales requieren entrenamiento y soporte. Esto genera problemas de mantenimiento y entrenamiento para múltiples herramientas además del problema potencial de resultados diferentes sobre los mismos datos debido a las diferencias en el software.

➢ HERRAMIENTAS DE NUEVA GENERACIÓN

En muchas maneras EDQM es similar a TQM. Esta última promueve la importancia en la prevención de faltas, mientras que EDQM entiende la naturaleza de la inconsistencia de los datos y provee de reingeniería en la fuente y dentro de la infraestructura organizacional. Ambas apuntan a la empresa en su totalidad y buscan tecnología que permita la implementación exitosa de programas.

Ha surgido un conjunto de herramientas de nueva generación para la reingeniería de datos que soporta el EDQM. Difieren significativamente de las herramientas de primera generación en que son [P2]:

- Versátiles y poderosas: soluciones óptimas proveen procesamiento sensible al contexto para grandes volúmenes de datos al momento de la entrada de datos y en modo batch dentro de los sistemas.

- Portable e independiente de la plataforma.

- Basados en estándares: facilitan el mantenimiento y entrenamiento, así como ayudan a codificar y estandarizar toda la compañía alrededor de la calidad de datos.

- Funcionalidad global: como todas las tecnologías, la limpieza de datos y prácticas de reingeniería deben focalizarse en el mercado global. Los requerimientos para procesar clientes internacionales presentan grandes desafíos, debido a que hay una gran variación en los estándares de datos de los diferentes países y en el nivel de completitud de la información capturada.

- Extensible y adaptable: son los suficientemente flexibles para adaptarse a los distintos ambientes de tipos de datos, y pueden manejar una gran variedad de conjuntos de reglas.

- Fáciles de usar: enfocan la limpieza de datos en forma intuitiva. Utilizan archivos de texto en vez de generar código.

Por todo esto las herramientas de segunda generación son mejores para un enfoque empresarial que puede acarrear la implementación de múltiples data marts, así como ajustarse a los sistemas OLTP para prevenir problemas de datos en vez de arreglarlos posteriormente. También puede ser reconfigurado más fácilmente para responder a las condiciones cambiantes del negocio.

2.2.5 CALIDAD EN UN DATA WAREHOUSE

Lo fundamental es tomar un enfoque balanceado que incluya: una buena definición de los datos, la limpieza de los mismos tanto en su origen como en el data warehouse y la mejora de la calidad de los datos atacando la raíz de las causas de los problemas de calidad.

Ubicándonos en el marco de referencia de un proyecto de data warehousing y basándonos en la información recabada en el transcurso del taller acerca del tema calidad de datos, presentamos una solución para construir un data warehouse de calidad. Es una propuesta a nivel general, donde se presentan pautas a seguir para lograr dicho objetivo.

El objetivo es lograr construir un data warehouse que refleje una imagen válida y consistente del negocio para el cual se está implementando. La credibilidad en los datos reportados es fundamental en el éxito del proyecto, ya que en ellos se basará el usuario final para tomar sus decisiones de negocio.

Consideramos que para alcanzar dicho objetivo se deben poner en práctica los siguientes procesos: **limpieza de datos** y **mejoramiento de la calidad** de los datos.

La limpieza de datos es el proceso de llevar a un estado de calidad los datos que serán ingresados en el warehouse. Este paso es necesario para evitar la entrada al data warehouse de datos en mal estado generados durante años por procesos erróneos de los sistemas operacionales.

El mejoramiento de la calidad de los datos es un proceso que va mas allá de la construcción del data warehouse en sí. A diferencia de la limpieza de datos que apunta a corregir errores, el proceso de mejoramiento de la calidad busca prevenirlos atacando los problemas de raíz, en la fuente de datos.

Para realizar la propuesta nos basamos en distintos artículos y white papers que hablan del tema. Tomamos distintas ideas de cada uno, las cuales adaptamos, ordenamos y agrupamos para asi convertirlas en una solución general y completa. Dentro de las publicaciones leídas no encontramos mucha información concreta de cómo construir un data warehouse de calidad. Salvo propuestas de algunas herramientas, tanto artículos como libros dan ideas generales de tareas puntuales que se deben realizar. Esta fue nuestra motivación principal para definir esta propuesta, donde se intenta abarcar todos los puntos importantes a considerar. Aunque solamente se dan pautas, la propuesta servirá como punto de partida para el desarrollo de futuras soluciones más sofisticadas [A1][A15][P6].

➢ PROCESO DE LIMPIEZA DE DATOS

El proceso de limpieza de los datos puede verse como una actividad dentro del proceso global de carga de datos al data warehouse. En general, dentro de los procesos de carga estándar se pueden diferenciar dos etapas: el *In-flow* y el *Up-flow* (figura 5). El In-flow describe el flujo de los datos desde su creación o captura hasta su ingreso al warehouse. El Up-flow abarca las etapas donde los datos se resumen a formas relevantes a los usuarios. A través del uso de proyecciones, funciones de agregación y agrupamientos, los datos son empaquetados dentro de vistas que focalizan asuntos de negocios específicos.

*Se presenta el proceso, como un **conjunto de rutinas** distribuídas tanto en el in-flow como en el up-flow. Dichas rutinas reportan errores, estandarizan, transforman e integran los datos, resultando en un data warehouse consistente que refleja correctamente el negocio que documenta.*

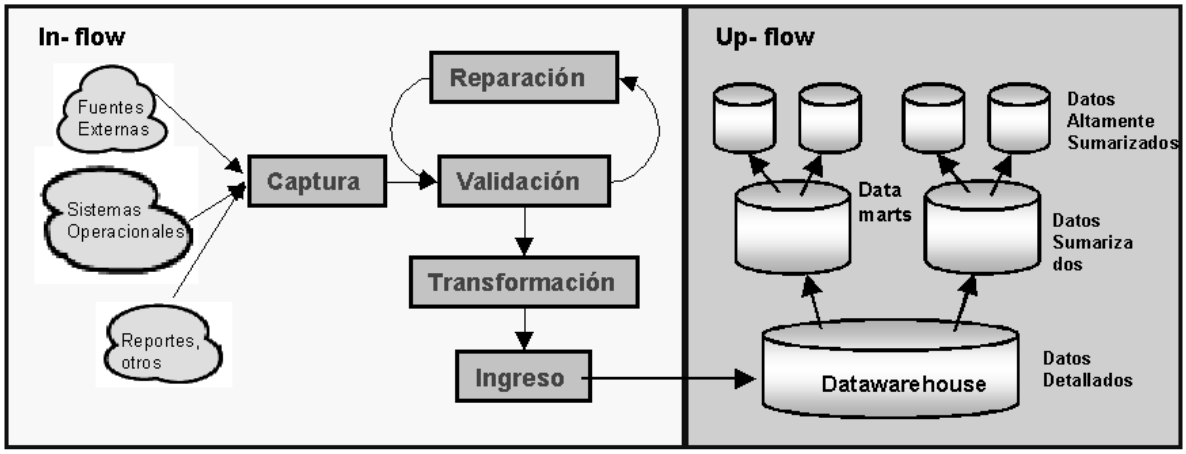


Figura 5: Proceso global de carga al data warehouse

Las rutinas de limpieza se convierten en otro de los productos a construir dentro del desarrollo del data warehouse, a menos que se cuente con herramientas que faciliten la tarea.

Veremos cuáles son estas rutinas de limpieza de datos y una metodología para asegurar la limpieza de datos en el proceso de desarrollo de un data warehouse.

A. RUTINAS PARA LA LIMPIEZA DE LOS DATOS

Presentamos una lista de rutinas que deben ser tenidas en cuenta dentro del proceso de carga de datos al data warehouse. Si bien, las necesidades específicas de limpieza pueden variar de un proyecto a otro, las rutinas son aplicables a la mayoría.

1. Chequeo de versión

El chequeo de versión se realiza para detectar cambios en la especificación del meta-data. Los cambios en la codificación de datos fuente, pueden ser capturados comparando la especificación del metadata.
Un caso de utilidad es cuando se tiene un campo booleano codificado con valores 1 o 0 y se modifica la codificación a TRUE o FALSE.
El chequeo se realiza apenas se extraen los datos de sus fuentes. La detección a tiempo de estos casos previene la entrada de datos que pueden ser mal interpretados posteriormente en el data warehouse.

2. Chequeo de uniformidad

Asegura que los valores de los datos que estan siendo cargados, se encuentren dentro de limites preestablecidos. Las reglas de negocio determinan qué valores son factibles en cada campo de datos, este chequeo verifica que los datos que se ingresen al data warehouse cumplan dichas reglas.
El chequeo se realiza durante el in-flow e intenta evitar que datos inválidos pasen al data warehouse.

3. Transformación de datos

Las rutinas de transformación de datos, convierten los datos a una forma adecuada para el data warehouse. Si bien los datos fuente pueden ser correctos para los sistemas operacionales, por su forma, no siempre son de utilidad al data warehouse. Dado que dichos datos contienen la información necesaria, se utilizan rutinas para que los transforme apropiadamente.

Podemos distinguir las transformaciones a dos niveles diferentes:

- Acondicionamiento y estandarización de datos → se modifican los datos cuando tienen error o no tiene valores (campos vacíos).
- Utilizar reglas de negocio → aplicar reglas características del negocio para transformar los datos adecuadamente.

Las transformaciones también ocurren durante el in-flow.

En general en toda construcción de un data warehouse se necesitan realizar transformaciones de los datos. Los datos operacionales no se adaptan directamente a los requerimientos de un data warehouse, la misma información se necesita ver desde otra óptica.

4. Integración de datos

Las rutinas de integración de datos tienen como objetivo identificar y consolidar registros. Implica la tarea de identificar entidades de datos para que no vuelvan a ser cargadas equivocadamente como entidades nuevas al data warehouse, y de resolver conflictos entre datos provenientes de diferentes fuentes.

La integración se debe realizar durante el in-flow.

En los proyectos donde se extraen datos de diferentes fuentes, los conflictos de integración se dan a menudo. Incluso pueden ocurrir estos mismos conflictos dentro del mismo sistema, donde la información representa una misma entidad pero se encuentra guardada de distinta manera. Determinar automáticamente cuando dos entidades en realidad son la misma no siempre es una tarea simple.

5. Supervivencia y formateo de datos

Ocurre en la última etapa del in-flow, donde se termina de preparar la información, para asegurar que los datos mas relevantes sean tenidos en cuenta y se encuentren en la forma adecuada. La rutina se encarga de varias tareas:

- Data filling → ingresar valores faltantes en registros remplazándolos con valores de registros relacionados, correspondientes a la misma entidad.

- Resolución de conflictos de datos → resolver problemas de registros múltiples que se refieren a la misma entidad con atributos conflictivos.
  Un ejemplo a este caso son registros de clientes que coinciden en el nombre y dirección pero tienen diferente número de RUC.

- Enriquecimiento de datos → agregar datos de fuentes externas que provean información suplemental.

- Supervivencia de datos → determinar los datos apropiados a cargar en caso de múltiples posibilidades. La información se puede encontrar en más de un lugar y debe determinarse cuál seleccionar para el data warehouse.

- Salida de datos → adaptar la salida a los requerimientos técnicos y de negocio. El data warehouse es consultado utilizando distintas herramientas de consulta y análisis, por lo que sus datos deben poder ser accesibles a las mismas.

6. Chequeo de completitud

Determina la completitud y correctitud de las agregaciones de datos. Las agregaciones son útiles pero pueden ocultar datos importantes.
Un ejemplo es sacar un promedio de ventas a partir de campos con valores nulos, el promedio puede estar bien calculado pero no refleja la realidad correctamente.
Ocurre durante el up-flow cuando se realizan las agregaciones de datos.

7. Chequeo de conformidad

Correlaciona los datos con fuentes de datos estándares. Valida si los datos se adecuan a otras fuentes de datos y reportes. Este chequeo permite descubrir casos excepcionales, donde se debe investigar si la causa de los mismos proviene de datos erróneos o que los resultados reflejan un cambio en la realidad.
Por ejemplo, si el promedio de ventas en un departamento del pais se mantiene constante durante todos los meses del año y en determinado mes aumenta al doble, los resultados pueden deberse a datos ingresados en forma errónea por los operadores de los sistemas operacionales o que las ventas en realidad subieron en dicho mes.

Ocurre en el up-flow cuando los datos se encuentran en la forma adecuada para el data warehouse.

8. Genealogico

Provee un auditoria completa de la fuente de datos. Ocurre durante el reporte de datos cuando el consumidor de la información cuestiona la validez de los datos. Con las herramientas OLAP, el chequeo genealógico se refiere a realizar un drill-down sobre una agregación y llegando a un nivel de granularidad mas fino en los datos.

B. METODOLOGÍA PARA ASEGURAR LA LIMPIEZA DE DATOS

Se plantea una metodología de trabajo para construir un sistema de DW de buena calidad, donde el proceso de carga se encarge de ejecutar las rutinas vistas en el punto anterior. Se define un conjunto de tareas que deben ser realizadas durante las distintas etapas de desarrollo del data warehouse.

La metodología aplica un enfoque bottom-up, comenzando al nivel de los valores de los datos. Mediante la investigación de cada valor en los campos, descubriendo información dentro de campos sin formato, resolviendo problemas de estandarización y reconociendo información perdida, se puede alcanzar el conocimiento necesario acerca del nivel de calidad en que se encuentran los datos fuente. Dado el nivel de calidad de las fuentes podemos determinar entonces si los datos son lo suficientemente confiables para implementar un data warehouse en base a los mismos. Ubicado el conjunto de problemas en los datos fuente,

podemos definir las rutinas de limpieza necesarias para transformar dichos datos en información correcta, completa y consistente para el data warehouse.

La metodología se aplica al proceso de construcción de un data warehouse, pero más precisamente, al proceso de desarrollo de la carga y construcción del data warehouse relacional, que compone el proceso global.

Pensamos que la metodología es útil para proyectos de tamaños chicos y medianos y se ajusta a las necesidades actuales de nuestro medio.

En la figura de abajo podemos observar como participa la metodología en el proceso de desarrollo de un data warehouse. Las llamadas contienen en su interior el nombre de las tareas que deben realizarse durante las etapas de análisis, diseño e implementación.

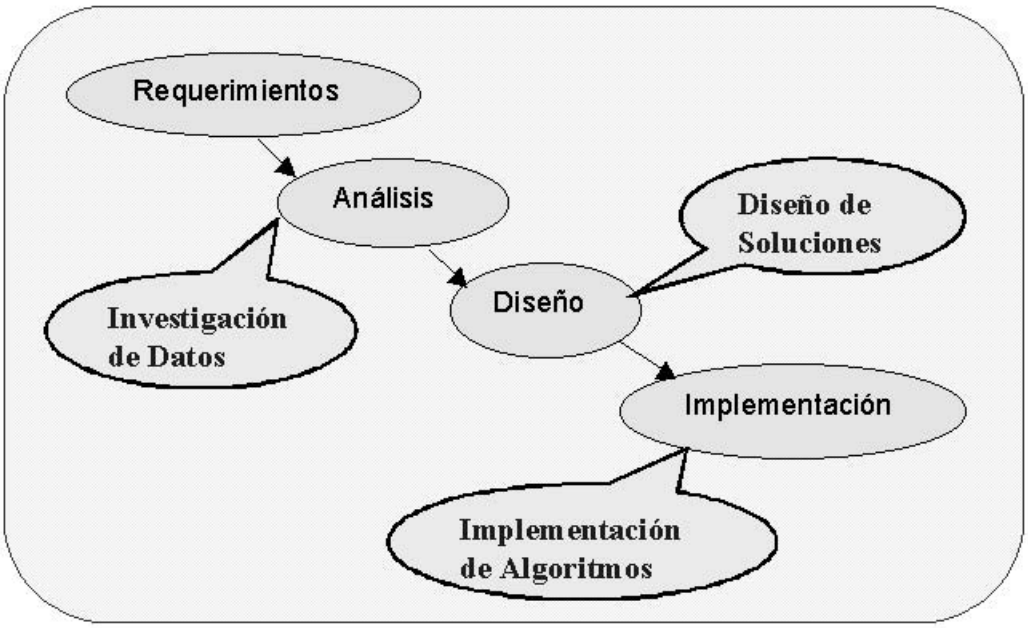


Figura 6: Metodología de aseguramiento de la limpieza en los datos

Veamos en detalle que implica cada tarea.

1. Investigación de datos

La investigación de los datos brinda una visión sobre la condición real de los mismos, ayudando a conocer y comprender mejor la información que se encuentra en las fuentes de datos y su nivel de calidad.

La documentación de los sistemas operacionales puede establecer qué información manejan dichos sistemas, pero esto no nos asegura que lo documentado es realmente lo almacendo. El sistema fuente puede estar diseñado para trabajar con cierta información pero por muchos

motivos (ingreso de información errónea, procesos con bugs, manejo inadecuado del sistema, etc.) la información que realmente se encuentre almacenada puede ser muy distinta o contener muchos errores. La investigación detallada de los datos se convierte en la base de conocimiento más importante sobre los datos fuente, sobre la cual se determina si el estado de calidad de la fuente es adecuado para poblar el data warehouse.

El proceso de investigación se realiza en la etapa de análisis del proceso de desarrollo del data warehouse. El análisis de los datos, en conjunto con la documentación del sistema fuente y el conocimiento de las reglas de negocio, permiten determinar los datos válidos para el data warehouse, datos que se necesiten transformar o limpiar, datos ocultos de utilidad y datos que se tienen que descartar, tanto porque no tienen información necesaria o porque la cantidad de errores en los mismos es irremediable.

Se realiza un análisis detallado de los datos fuente que se compone de las siguientes actividades principales:

- *Data typing*
  Asignarle un significado relativo al negocio, a cada valor encontrado en los campos de datos. Se estudia que el mismo tenga sentido dentro del negocio.

- *Metadata mining*
  Descubrir metadata oculta y prácticas de negocio no documentadas.

- *Parsing de datos y extraccion de entidades y atributos*
  Extraer datos especificos que se encuentren mezclados en campos de diferente dominio.

En el mercado han surgido algunas herramientas que facilitan este tipo de investigación de los datos. En particular INTEGRITY de Vality Technology define y utiliza en su metodología las tres actividades mencionadas, lo hace en forma automática aplicando análisis lexicográfico, procesamiento de formatos (pattern processing) y matcheo estadístico. También otra herramienta, WizRule utiliza algoritmos matemáticos para revelar las reglas que gobiernan los datos. La característica más sobresaliente en estas herramientas es su elevado precio, que obligan en muchos casos a descartar su uso. Sin embargo aún es posible realizar la investigación de los datos. Si estamos analizando una fuente con poca cantidad de datos, la labor puede realizarse usando herramientas ad-hoc para ejecutar consultas significativas. En caso de tener que analizarse una fuente de gran cantidad de datos, esta tarea puede llevar mucho tiempo por lo que sería conveniente desarrollar aplicaciones propias que apliquen algoritmos, al estilo de las herramientas mensionadas, para revelar la información que ocultan las bases fuentes de datos.

2. Diseño de soluciones.

Dentro del proceso de desarrollo del data warehouse, después de finalizada la etapa de análisis, se está en condiciones de definir las rutinas de limpieza que se necesitan incluir en la carga. Se conoce el conjunto de datos fuente seleccionados para cargar, se conoce el nivel de calidad en que se encuentran los datos y está ya analizado todo lo requerido por el data warehouse.

En la etapa de diseño entonces, se define el conjunto de rutinas de limpieza que se incluirán para solucionar los problemas de calidad encontrados durante el análisis. Se pueden encontrar herramientas que realicen las rutinas, herramientas que en general que se encargan de realizar todo el proceso de carga. Pero también las rutinas pueden generarse dentro del proceso de desarrollo del data warehouse, en cuyo caso en esta etapa se realizaría el diseño de las mismas.

Las rutinas pueden ser diseñadas como soluciones automáticas, semi-automáticas o directamente manuales, donde debe participar el usuario encargado de la carga. El diseño además de influir en el proceso de carga, también puede influir en el diseño de las estructuras de almacenamiento del data warehouse. Por lo que el diseño de estas soluciones debe realizarse en conjunto con el diseño de las estructuras de almacenamiento y del proceso de carga.

Por otro lado, si bien las rutinas pueden diseñarse como soluciones a problemas particulares a un sistema, sería interesante que se diseñaran en forma que se adapten a distintos casos. De esta manera en cada proyecto solamente se tendrían que 'customizar' las rutinas de limpieza. Por ejemplo, definir la rutina de chequeo de versión en forma genérica, que acceda a la especificación del metadada y verifique si los datos fuente cumplen con la especificación. Podría también definirse los distintos chequeos para que permitan 'customizar' las reglas que deben cumplirse en cada ocasión y entonces los chequeos utlizarían estas reglas para validar los datos.

3. Implementación de algoritmos*.*

En caso de realizar las propias rutinas de limpieza de datos, en la etapa de implementación del data warehouse, se desarrollarían las mismas. Si utilizamos soluciones externas debemos incluirlas al proceso de carga a menos que el mismo ya las contenga.

Estas tres etapas constituirían la metodología de aseguramiento de limpieza de datos. Metodología que debiera aplicarse al desarrollo de cada data warehouse o data mart que implementemos.

➢ CALIDAD EN LOS DATOS

Sin duda alguna, para asegurar la calidad en los datos del data warehouse se deben mejorar los procesos de negocio que producen los datos y concientizar la organización de la importancia que tiene la calidad de los datos para lograr beneficios en el negocio.

Alcanzar la calidad significa colmar las expectativas del cliente y en un data warehouse el cliente es el trabajador experto, el usuario que se basa en los datos para tomar las decisiones en su negocio.

¿ Quién provee los datos al data warehouse ? ¿ Los sistemas de información ? No !. Los proveedores son los productores de datos que crean y actualizan los datos en la medida que hacen su trabajo. Los productores de datos deben ingresar los datos de una manera que satisfaga no sólo los requerimientos de calidad de su departamento sino que también satisfaga a todos los usuarios del data warehouse, quienes no están relacionados directamente con ellos. Hacer entender esto a los usuarios de los sistemas operacionales no siempre es una tarea fácil. Tampoco lo es, plantearle a la gerencia, que los sistemas operacionales no tienen la calidad suficiente para utilizar sus datos en el data warehouse, siendo nosotros mismos quienes desarrollamos esos sistemas.

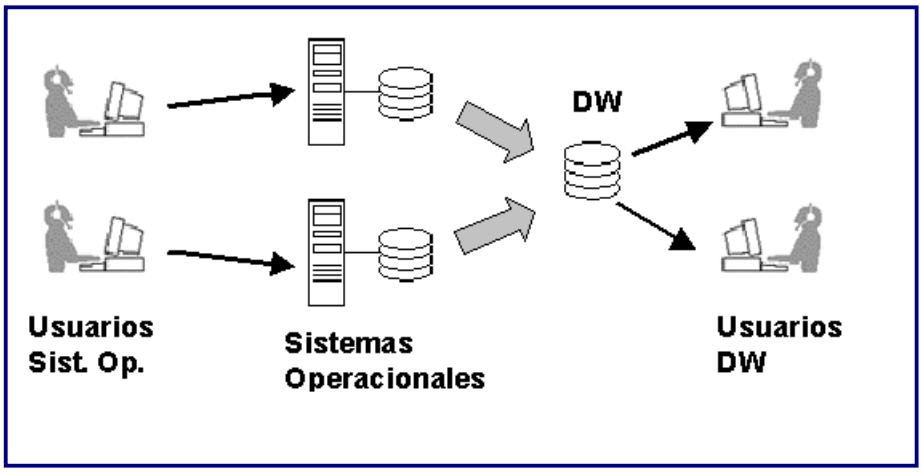


Figura 7: Influencia de usuarios Sist. Operac. sobre los usuarios del data warehouse

A. GUÍA PARA ASEGURAR LA CALIDAD DE DATOS

La guía que presentamos resume los puntos más importantes a considerar dentro de una organización para asegurar la calidad de datos.

- Desarrollar una filosofía de calidad en la producción de los datos. Ingeniar los procesos de negocio y aplicaciones tales que los datos creados colmen las expectativas de los usuarios del data warehouse.
- Definir los datos consistentemente entre todos los futuros usuarios del data warehouse.
- Reestructurar los procesos de negocio antes de automatizarlos para eliminar pasos intermedios que pueden agregar costos e introducir errores sin agregar valor.
- Ubicar los programas de captura de datos lo más cerca posible del evento de negocio que origina esos datos.
- Ingresar reglas de validación automática que se disparen al momento que se ingresan los datos y validen si los mismos son correctos.
- Capturar todos los datos del negocio, requeridos por usuarios del data warehouse o procesos subsiguientes.
- Permitir actualizar los datos siempre.
- Automatizar captura de datos para evitar errores intermedios.
- Permitir cargar el valor "desconocido" en cada uno de los campos para cuando el productor de los datos no conoce el valor real.
- Mantener comunicación frecuente y feedback entre usuarios del data warehouse y productores de datos para asegurar la vigencia de los datos volátiles.
- Estimular a la gente de negocios a tener los datos lo más actualizados posible.
- Identificar y designar registro origen, registro de referencia y base de datos replicadas.
- Distribuir los datos mediante replicación y controlar fuertemente cualquier transformación de los mismos.
- Entrenar a los productores de datos y proveerles acceso a la definición de los datos.
- Hacer que tanto los encargados de ingresar los datos como los encargados de los procesos de negocios se sientan responsables de la calidad de los datos.

2.3 ARQUITECTURA DEL DATA WAREHOUSE

El éxito de la tecnología de DW ha sido muy grande en los últimos años y ha permitido evolucionar mucho la materia. Las distintas experiencias demostraron, ideas que en un principio parecían correctas, no lo eran tanto. En este capítulo nos centraremos en los cambios que ha sufrido el DW con respecto a la arquitectura del sistema.

2.3.1 DATA WAREHOUSE CENTRAL

El concepto inicial detrás del DW era el de crear un repositorio de alcance empresarial que homogeneizara y uniera todos los datos de la organización en una única estructura, desde donde todos los departamentos pudieran obtener una visión coherente de la organización. Este concepto implicaba que todos los sistemas de producción provieran de información al warehouse y que todas las extracciones y transformaciónes dentro de toda la organización estuvieran bajo el control de un único proceso.

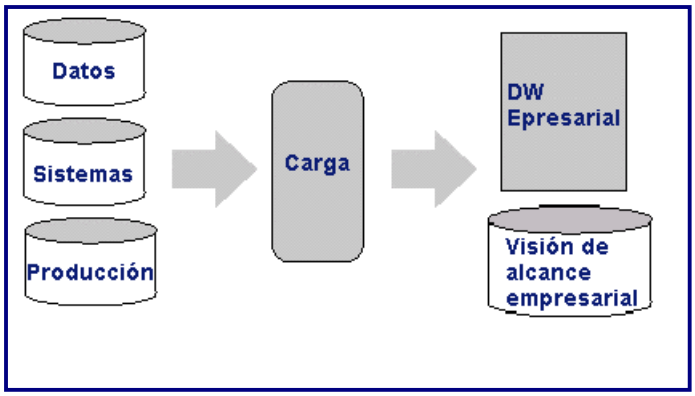


Figura 8: Arquitecutra data warehouse global centralizado

Los problemas en proyectos que aplicaron este concepto estilo 'big bang' fueron apareciendo. Construir un data warehouse global centralizado es complicado por varias razones que incluyen temas políticos empresariales, temas financieros, temas de performance y temas de estrategia. Los temas políticos principalmente se relacionan con pertenencia de datos y provincialismo, algunos departamentos no están de acuerdo con compartir sus datos. Financiar estos proyectos implica invertir mucho dinero y esperar varios meses para obtener algo de utilidad. Surgen problemas también de performance al momento que usuarios de distintos departamentos quieren consultar su información al mismo tiempo dentro del warhouse. Por último, dado el tiempo que se necesita para desarrollar un sistema de este tamaño desalienta a los usuarios dado que al momento que se complete los requerimientos ya habrán cambiado.

2.3.2 ESTRATEGIAS DATA MARTS

En contrapartida a la estrategia anterior los Data Mart se basan en la teoría 'Divide and Conquer', donde se construyen almacenes de información especifica que apuntan a una área del negocio en particular. El concepto en este caso deriva de la certeza que cualquier usuario tiene necesidades de información limitada, y aunque tipicamente existen requerimientos para analisis funcionales cruzados, el tamaño de los requerimientos es reducido materialmente si limitamos el tamaño del warehouse en sí mismo.

Dos estrategias distintas se desarrollan a partir del concepto de data marts, la de data marts dependientes y la de data marts independientes.

➢ DATAMART DEPENDIENTES

En esta arquitectura los datos son cargados desde los sistemas de producción hacia el data warehouse empresarial y entonces subdivididos en data marts. Se llaman data marts dependientes porque utilizan los datos y metadatos del data warehouse en lugar de obtenerlos de los sistemas de producción.

Esta solución resuelve los problemas de performance, estrategia, finanzas e incluso algunos de los problemas políticos. Aunque tiene esos puntos a favor, sigue teniendose que construir el data warehouse global antes que los data marts sean implementados.

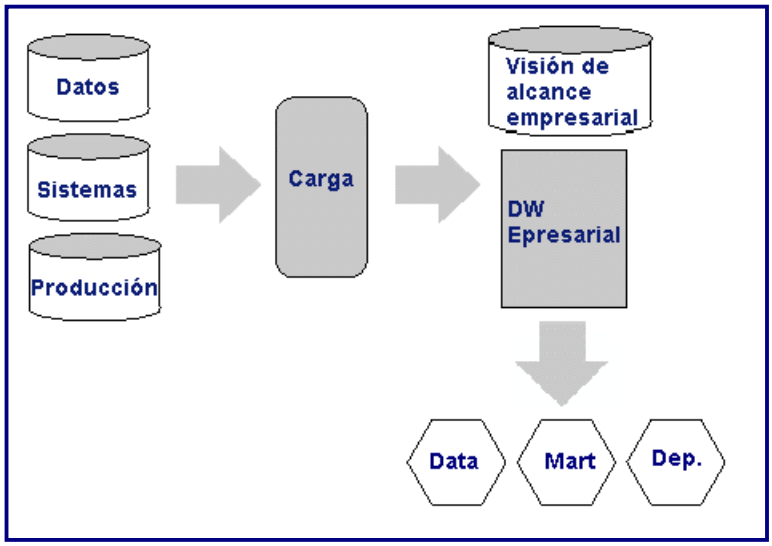


Figura 9: Arquitectura de Datamart dependientes

➢ DATA MART INDEPENDIENTES

Esta óptica es considerada por muchos como una alternativa del data warehouse central. Con ella es posible comenzar con un sistema pequeño, invirtiendo menos dinero y obteniendo resultados limitados entre tres a seis meses. Los que proponen esta arquitectura, argumentan que luego de comenzar con un data mart pequeño, otros marts pueden proliferar en otras lineas de negocio o departamentos que tengan necesidades, y que satisfaciendo las distintas necesidades divisionales, una organización puede construir su camino para el data warehouse completo, de una manera bottom up.

En este caso de data marts múltiples también se tienen procesos de carga múltiples donde los datos son extraidos desde sistemas de producción quizás en forma redundante.

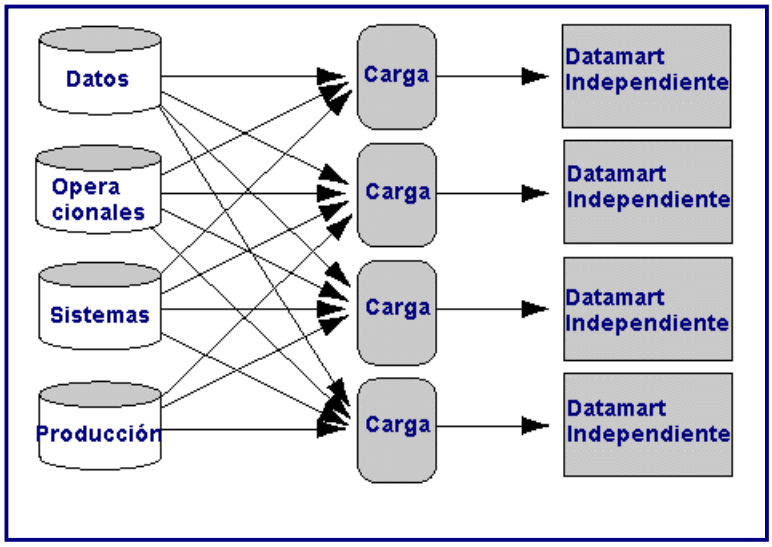


Figura 10: Arquitectura de Datamarts independientes

2.4 DISEÑO DE UN DATA WAREHOUSE

2.4.1 INTRODUCCIÓN

El estudio realizado sobre el tema diseño de un data warehouse, se centra en el trabajo 'A transformations based approach for designing the Data Warehouse' de la Ing. Adriana Marotta, presentado como tesis de su maestría al departamento de Computación de la Facultad de Ingeniería de la Universidad de la República del Uruguay. Presentamos aquí una breve descripción del trabajo.

2.4.2 'A TRANSFORMATIONS BASED APPROACH FOR DESIGNING THE DATA WAREHOUSE'

Las características particulares de un data warehouse hacen que el proceso de diseño sea diferente del diseño tradicionalmente utilizado para bases de datos relacionales. Se debe tener en cuenta por ejemplo, que la existencia de redundancia es necesaria a la hora de diseñar un data warehouse ya que mejora la performance de las consultas. Además, la redundancia no implica anomalías al actualizar los datos ya que por lo general la actualización de un data warehouse se realiza mediante procesos de carga batch.

El trabajo 'A transformations based approach for designing the Data Warehouse', propone un conjunto de primitivas para apoyar el diseño de un data warehouse por considerar que 1) las primitivas materializan el conocimiento existente sobre diseño de data warehouses, 2) permiten el seguimiento del diseño una vez realizado. Además, aumentan la productividad del diseñador, ya que el mismo dispone de piezas de diseño predefinidas, solo le resta seleccionar aquellas que considere de su utilidad para el caso que enfrente y combinarlas de forma de obtener el diseño final del data warehouse.

La mayor contribución de éste trabajo es la propuesta de un conjunto de primitivas para el diseño de un data warehouse. Estas primitivas deben ser aplicadas a la integración de los sistemas fuente para luego obtener el data warehouse final. Junto con cada primitiva, se provee la especificación de las transformaciones que deben aplicarse, a instancias del esquema fuente, para realizar la carga al data warehouse.

➢ PRIMITIVAS PARA EL DISEÑO DE UN DATA WAREHOUSE

El siguiente es un resumen de las primitivas definidas, donde se presenta la descripción de cada una de ellas.

1	Data Filter	Dada una tabla de medidas, filtrar los atributos que son de interés. Utilidad: Eliminar los datos puramente operacionales.
2	Temporalization	Dada una tabla de medidas o de dimensión, agregar un elemento de tiempo al conjunto de atributos de la tabla.
3	Key Generalization	Dada una relación de dimensión, se generaliza la clave, agregando 2 dígitos de versión a cada valor de este atributo.
3.1	Version digitis	Dada una relación de dimensión, se generaliza la clave agregando dos dígitos de versión a cada valor de este atributo.
3.1	Key Extension	Dada una relación de dimensión, se extiende la clave, haciendo que este compuesta por más atributos.
4	DD-Adding	Estas primitivas tienen como objetivo general agregar a una relación un atributo que es derivado a partir de otros.
4.1	DD-Adding 1-1	Dada una relación, agregar un atributo que es derivado (calculado) a partir de otros de la misma. Este atributo no debe cambiar la granularidad de la tabla.
4.2	DD-Adding N-1	Dadas dos relaciones, agregar en una de ellas un atributo que se deriva de la otra. En este caso la función de cálculo trabaja sobre solo una tupla de la otra relación. Esta tupla debe poderse obtener en forma única mediante un join. El atributo derivado puede definirse como clave externa con respecto a otra relación.
4.3	DD-Adding N-N	Dadas dos relaciones, agregar en una de ellas un atributo que se deriva de la otra. En este caso la función de cálculo trabaja sobre un conjunto de tuplas de la otra relación. Este se obtiene mediante un join de las dos relaciones.
5	Attribute Adding	Dada una relación de dimensión, agrega uno o más atributos a ésta. Utilidad: Mantener en una misma tupla más de una versión de un mismo atributo.
6	N:1 Dim Relationships	Primitivas para transformar un esquema compuesto por: una tabla de movimientos R1 y dos tablas de dimensiones R2 y R3, en donde existe una relacion N:1 entre R2 y R3, la cual cambia lentamente.
6.1	N:1 Dim Relationships	Dado un esquema compuesto por: una tabla de movimientos R1 y dos tablas de dimensiones R2 y R3, en donde existe una relacion N:1 entre R1 y R2 y una relación N:1 entre R2 y R3, generaliza la clave de la tabla R2, y hace que la tabla de movimientos la referencie.
6.2	N:1 Dim Relationships	Dado un esquema compuesto por: una tabla de movimientos R1 y dos tablas de dimensiones R2 y R3, en donde existe una relacion N:1 entre R1 y R2 y una relación N:1 entre R2 y R3, hace que la tabla de movimientos, además de referenciar a la tabla de dimensión R2 también referencia a R3.

6.3	N:1 Dim Relationships	Dado un esquema compuesto por: una tabla de movimientos R1 y dos tablas de dimensiones R2 y R3, en donde existe una relacion N:1 entre R1 y R2 y una relación N:1 entre R2 y R3, generaliza la clave de la tabla R2, y hace que la tabla de movimientos la referencie. Incluye los datos de la tabla R3 en R2 (desnormaliza).
6.4	N:1Dim Relationships	Dado un esquema compuesto por: una tabla de movimientos R1 y dos tablas de dimensiones R2 y R3, en donde existe una relacion N:1 entre R1 y R2 y una relación N:1 entre R2 y R3, incluye los datos de la tabla R3 en la tabla de movimientos R1 y en la tabla R2 (desnormaliza).
7	Hierarchy Roll Up	Dada una tabla de medidas R1 y una tabla de dimensión R2 que contiene una jerarquía, obtiene una tabla de medidas R'1 en donde se aumentó el nivel en la jerarquía (se disminuyó la granularidad) del atributo correspondiente (que es clave externa) a la dimensión R2. Además puede generar una tabla R'2 derivada de R2, con la granularidad tomada anteriormente.
8	Aggregate Generation	Dada una tabla de medidas R, obtiene una tabla de medidas R'1 en donde los datos están resumidos (o agrupados) según un conjunto de atributos dado.
9	Table Merge	Dadas dos relaciones R1 y R2, que representan información de tipo cabezal-renglones, genera una única relación R' cuyos atributos son la unión de los de R1 y R2.
10	Data Array Creation	Dada una relación de medidas, con un atributo de medida A, un atributo B que representa un conjunto de valores predeterminados (por ej., mes), y un atributo de tiempo C, genera una relación en donde en vez de los atributos B y A, aparece un conjunto de atributos que representan las medidas correspondientes a cada valor posible del atributo B.
11	Partition by Stability	Estas primitivas particionan una relación, de forma de facilitar el almacenamiento de la historia de los datos de ésta. Según el criterio que se desee aplicar, se podráutilizar la primera (Vertical Partition) o la segunda primitiva (Horizontal Partition) del grupo.
11.1	Vertical Partition	Dada una relación de dimensión, ésta se particiona en varias relaciones distrubuyendo sus atributos de forma que queden agrupados según su propensión a los cambios.
11.2	Horizontal Partition	A partir de una relación R se generan dos relaciones Ract y Rhist, que contienen los mismos atributos que R, y una restricción de integridad para cada uno que especifica que los datos deben ser actuales o históricos, según el caso.
12	Hierarchy Generation	Estas primitivas generan relaciones de dimensión correspondientes a una jerarquía, a partir de relaciones que incluyen a la jerarquía o a parte de ésta. Además, hace que estas relaciones referncien a la jerarquía con una clave externa.
12.1	De-Normalized	Esta primitiva genera una relación de dimensión que es una jerarquía, a partir de relaciones que incluyen a la jerarquía o a parte de esta. Además, hace que las relaciones referencien a la jerarquía con una clave externa.

12.2	Snowflake	Esta primitiva genera varias relaciones de dimensión que corresponden a una jerarquía, representada en forma normalizada, a partir de relaciones que incluyen a la jerarquía o a parte de ésta. Además, hace que estas relaciones referencien a la relación correspondiente al nivel más bajo de la jerarquía con una clave externa.
12.3	Free Decomposition	Esta primitiva genera varias relaciones de dimensión que corresponden a una jerarquía, a partir de relaciones que incluyen a la jerarquía o a parte de esta. Además, hace que estas relaciones referencien a la relación correspondiente al nivel más bajo de la jerarquía, con una clave externa. Las relaciones generadas para la jerarquía tendrán la forma (distribución de atributos) que el diseñador decida.
13	Minidimension Break off	Dada una relación de dimensión, se separa de ésta un conjunto de atributos, formando con ellos una nueva dimensión.
14	New Dimension Crossing	Dadas varias relaciones de dimensión o de cruzamiento, donde cada una tiene un atributo en común con alguna de loas otras, genera una única relación de cruzamiento cuyos atributos son la unión de algunos atributos de las anteriores.

2.5 REFERENCIAS

Introducción al Datawarehousing. [L1],[L2],[L3],[L7]

Calidad de Datos. [A1],[A4],[A6],[A15]
[P1],[P2],[P6],[P7],[P10],[P11],[P14],[P15],[P17],[P18],[P19]

Arquitectura de DW. [P1]

Diseño de DW. [P22]

SISTEMA DESARROLLADO

3

3.1 PLANTEO DEL SISTEMA

Se plantea construír un sistema DW que incluya tres data marts:

- Presupuesto, debe contener información de utilidad para la comisión encargada de administrar el Presupuesto de la F.I.
- Bedelía, debe contener información de utilidad para la comisión del I.N.CO. encargada del seguimiento de los estudiantes
- Asignaciones, debe contener información acerca de los docentes del I.N.CO y sus asignaciones a cargos en dicho instituto.

El sistema de DW debe construírse aplicando una estrategia Bottom Up, construyendo los distintos data mart que lo componen. Cada data mart debe ser desarrollado aplicando una estrategia Top Down, comenzando con el relevamiento de los requerimientos de usuario, para luego realizar las etapas de análisis, diseño e implementación.

Los data mart de Presupuesto y Bedelía deben comenzar su desarrollo en paralelo, hasta llegar al diseño, en ese momento comienza el desarrollo del data mart de Asignaciones, cuyo diseño debe integrarse al de los data marts anteriores. Luego de finalizados los diseños se realizan las implementaciones paralelamente (*).

Los data marts deben construírse dividiendo el desarrollo horizontalmente entre dos grupos de desarrollo. Un grupo es el encargado de relevar los requerimientos de usuario, definir el modelo multidimensional, y generar las aplicaciones de usuario final. El otro grupo, en particular nuestro grupo, es el encargado de construir el data mart relacional y su carga. Se sabe que ambos equipos deben interactuar durante el desarrollo, no están completamente definidas estas interacciones. En un principio este equipo de desarrollo recibe del otro equipo el modelo mutlidimensional de los datamarts, construye cada data mart relacional y su carga, el otro equipo extrae entonces los datos almacenados para cargar los cubos.

La tarea del primer grupo forma parte del proyecto: ‘Sistema de data warehousing: OLAP’. La tarea del segundo grupo, forma parte del proyecto que estamos documentando en este informe.

(*) La idea planteada incialmente de desarrollar en paralelo los data marts de Presupuesto y Bedelía no pudo llevarse a cabo. Como no se contaba con información del sistema operacional de Bedelía, y la información recaudada por el taller anterior no contemplaba las modificaciones a causa del cambio de plan, se comenzó unicamente con el data mart de Presupuesto, defasándose asi los desarrollos de ambos data marts.
El data mart de Asignaciones no fue construído debido a que se pensaban extraer los datos de un sistema que finalmente no se realizó.

3.2 ESTRATEGIA DE DESARROLLO

El sistema data warehousing que debe construírse es planteado bajo ciertas condiciones, especificadas en la sección anterior, donde se indican algunas características del sistema y puntualizaciones de cómo organizar el trabajo. Se define entonces una estrategia para desarrollar un sistema data warehousing, de modo que tanto el sistema como el desarrollo cumplan las condiciones planteadas. La estrategia se centra en el desarrollo del data mart relacional y su carga *(Figura 11)* y tiene como objetivo final la construcción del data warehouse.

Se consideran principalmente los siguientes puntos para la construcción de la estrategia:

A. ¿ Cómo desarrollar el data warehouse dada la estrategia Bottom Up ?
B. ¿ Cómo asegurar la limpieza de los datos del data warehouse ?
C. ¿ Cómo interactuar con el desarrollo OLAP ?

En respuesta al punto A) se decide focalizar la estrategia al desarrollo de un data mart. De esta manera el desarrollo de un sistema DW, se define como varios desarrollos más pequeños, uno por cada data mart, los cuales son construídos aplicando la estrategia.

La respuesta al punto B) integra la '*Metodología para asegurar la limpieza de un data warehouse'* a la estrategia de desarrollo de cada data mart. Dicha metodología se especifica en detalle en la sección 'Calidad de un data warehouse' del capítulo anterior.

Finalmente se estudia cómo coordinar este trabajo, la construcción del data warehouse relacional y su carga, con el desarrollo de la parte OLAP del sistema, resolviendo de ésta forma el punto C). Dado que la estrategia se centra en el desarrollo de un data mart, se define la interacción durante el desarrollo del mismo.

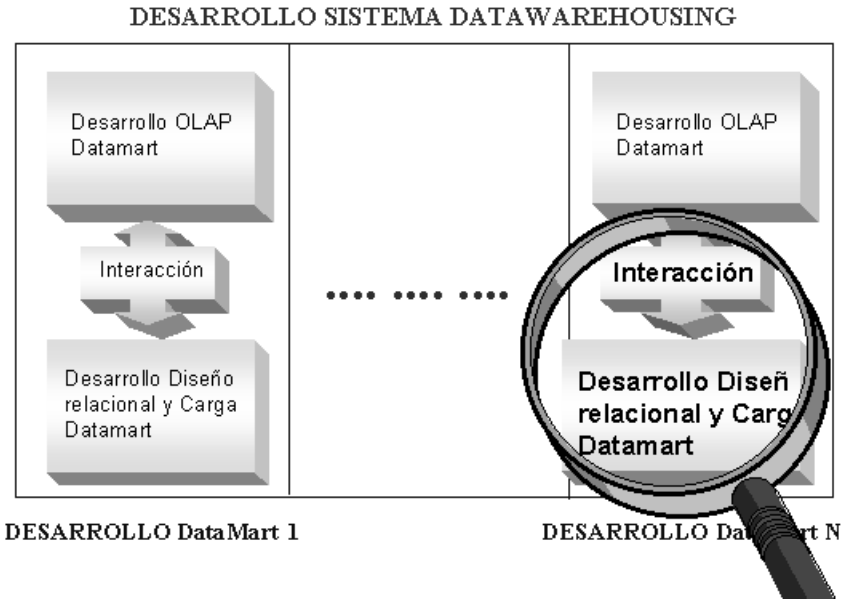


Figura 11: Alcance de la estrategia

A. DESARROLLO DEL SISTEMA DW

Se divide el desarrollo del Sistema DW en pequeños desarrollos, uno por cada data mart que lo componga.

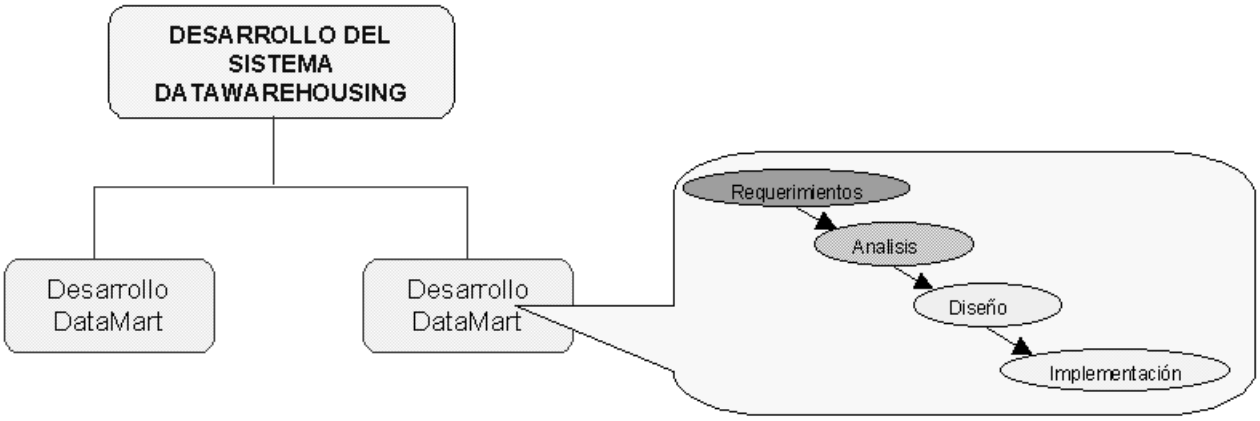


Figura 12: División del desarrollo en data marts

El sistema DW se va constituyendo mediante la construcción de nuevos data marts. Los data marts se crean a medida que surjan requerimientos específicos a un área del negocio. Por lo que pueden darse, tanto situaciones en que se construyan data marts secuencialmente, como situaciones en que surjan data marts en forma conjunta. La estrategia se adapta a ambas situaciones. En particular, en el segundo caso puede aplicarse en forma paralela, resaltándose la importancia de la comunicación que debe existir entre los desarrollos de cada data mart a modo de evitar conflictos de diseño y trabajo duplicado *(Figura 12)*.

En este tipo de estrategia Bottom Up, hay dos puntos importantes a considerar, la integración con otros data marts y la reutilización de componentes. Al desarrollar data marts en forma independiente, quizás alejados en el tiempo, estos temas pueden llegar a ser olvidados y por consiguiente aumentar inecesariamente la complejidad del desarrollo.

En cuanto a integración con otros data marts, nos referimos a que tanto los ya construídos como otros en vías de construcción, pueden manejar la misma información. Debe analizarse si la información cumple con los requerimientos actuales, si es conveniente realizar una integración o no, etc. La estrategia presentada tiene en cuenta este punto en la etapa de análisis.
La reutilización de componentes es también interesante, evita realizar trabajos en forma duplicada. Puede ser de mucha utilidad principalmente en el área de procesos de carga.

Se destaca la importancia de la metadata del sistema. Toda documentación sobre sistemas operacionales, datos fuente, modelos lógicos, modelos multidimensionales, datos del data mart, procesos de carga, configuración de herramientas, etc. sirve y facilita la resolución de los puntos antes discutidos. La estrategia define en cada etapa la documentación que debe generarse.

B. ASEGURAMIENTO DE CALIDAD DE DATOS

Se enriquece el desarrollo de cada data mart con la 'Metodología para asegurar la limpieza de datos'.

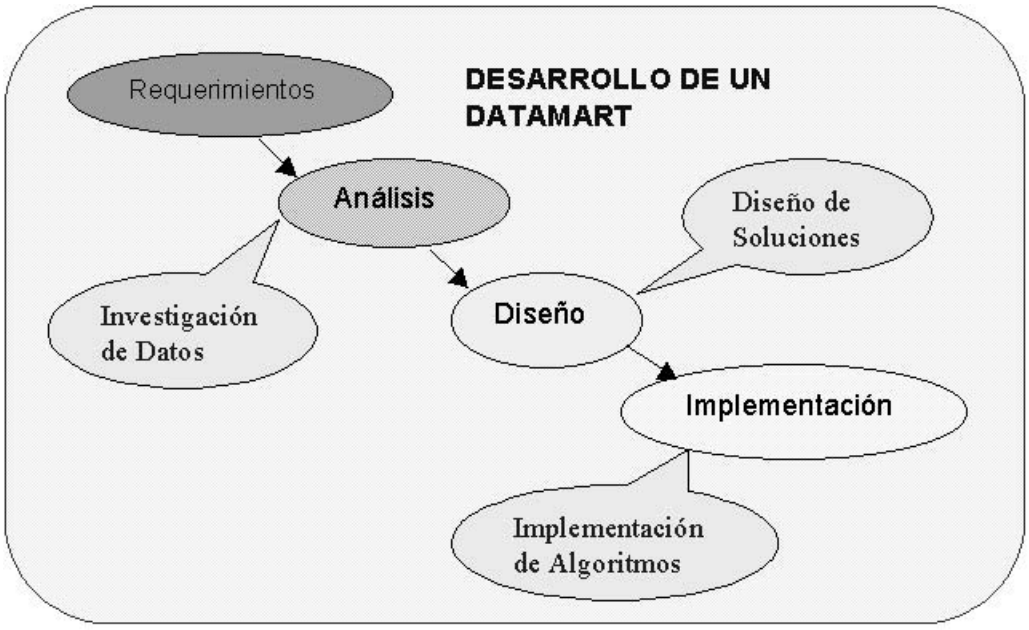


Figura 13: Desarrollo teniendo en cuenta Metodologia para asegurar la limpieza de datos

La importancia que tiene el tema calidad en un sistema DW no es discutible, sin embargo, la forma de considerarlo sí lo es. En este caso se optó por adaptar la estrategia de desarrollo de cada data mart, para que incluyera la 'Metodología para asegurar la limpieza de datos' *(Figura 13)*. Es una solución que permite construir data marts confiables.

En la sección Calidad de un data warehouse del capítulo anterior, se habla de la necesidad de realizar dos procesos en paralelo, limpieza y mejoramiento de la calidad de los datos. Esta estrategia considera principalmente la limpieza pero no deja de lado lo segundo ya que la guía de calidad presentada puede aplicarse en las distintas etapas de desarrollo. Si a nivel de la organización empresarial se define un plan de mejoramiento de calidad, la estrategia puede adaptarse a dicho plan.

C. INTERACCIÓN CON DESARROLLO OLAP

Dado que la estrategia se centra en el desarrollo de cada data mart, la interfase con el desarrollo OLAP se define dentro de la construcción de cada data mart.

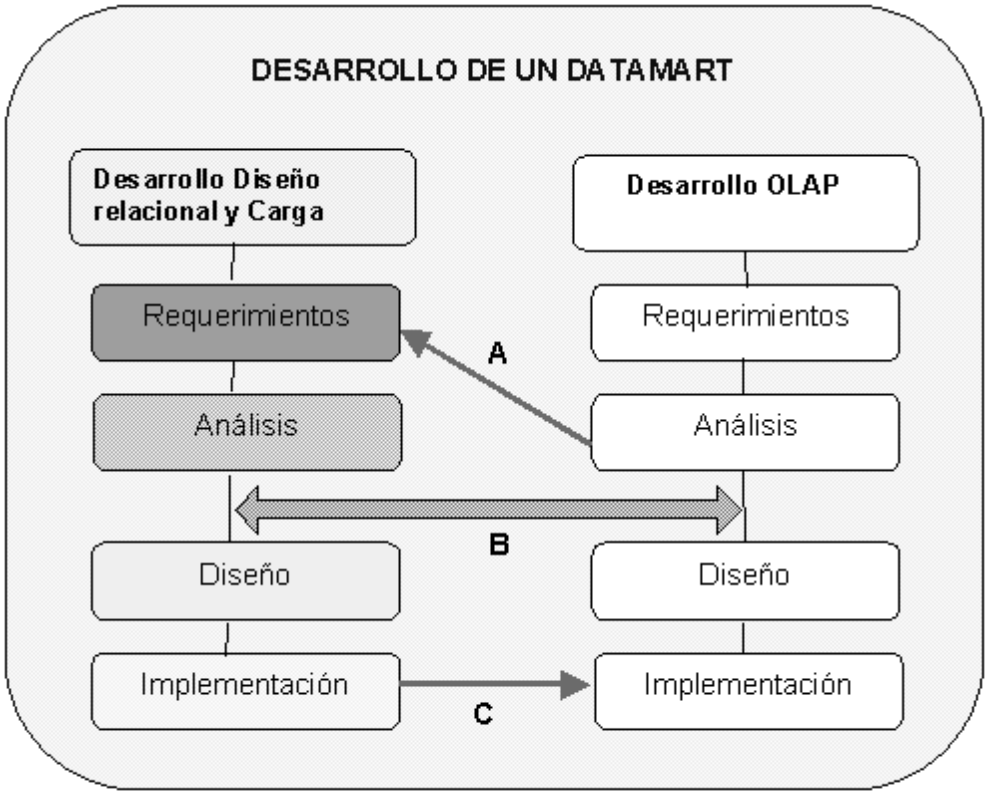


Fig 14: Interacción del desarrollo Carga y Control de calidad con el desarrollo OLAP

En el diagrama *(figura 14)* pueden observarse tres puntos de interacción, los cuales especificamos a continuación.

La interacción A, es una entrada de información para el desarrollo del data mart relacional y su carga, en donde se recibe el modelo multidimensional del data mart. El mismo es construído en el desarrollo OLAP, luego de haber relevado los requerimientos de usuarios finales del sistema y de haber modelado el negocio. No especificamos las tareas que componen las etapas de requerimientos y análisis del desarrollo OLAP pero sí, lo que se considera importante recibir como entrada en esta etapa.

El modelo multidimensional debe representar toda la información que el data mart necesite manejar. Documentar el significado de todo lo requerido, especificar las dimensiones, los cruzamientos de interes, las medidas y el nivel de granularidad en los datos. Debe especificar la evolución en el tiempo de cada medida, si es diaria, mensual, semestral, anual, etc.

La interacción B tiene la particularidad de ser de entrada y de salida. El grupo responsable de construir el data mart relacional y su carga informa al grupo OLAP, los datos con los que se cuenta para poblar el data mart. Lo que puede llevar a que el grupo OLAP realice modificaciones al modelo multidimensional para que éste se adapte a la realidad y/o que el grupo de diseño busque la forma de obtener la información faltante. En la interacción deben aclararse inquietudes, como por ejemplo: ¿qué formatos especiales que deben tener los datos almacenados?, ¿quién se encarga de calcular ciertos datos, la herramienta de usuario final o se guardan calculados?, ¿cómo se cargarán los cubos, en forma incremental o total? (no todas las herramientas soportan carga incremental), etc. Las respuestas a las anteriores inquietudes influyen sistemáticamente en el diseño de las tablas del datamart y sus procesos de carga.

Por último, en la interacción C, creadas las tablas del data mart se informa de su estructura y contenido al grupo OLAP para que éste cree el diccionario de datos.

3.2.1 DESARROLLO DE UN DATA MART

Se detallan los pasos que componen el desarrollo de un data mart, aplicando la metodología para asegurar la limpieza en los datos e interactuando con el desarrollo OLAP. Se especifican las distintas etapas dentro del desarrollo encargado de construir el data mart relacional y su carga.

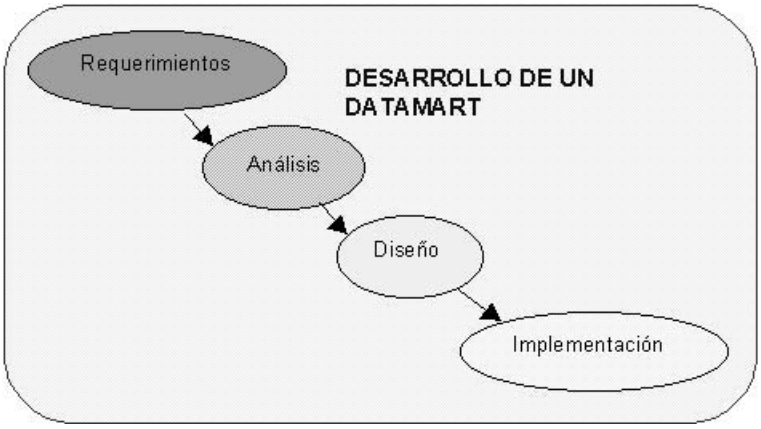
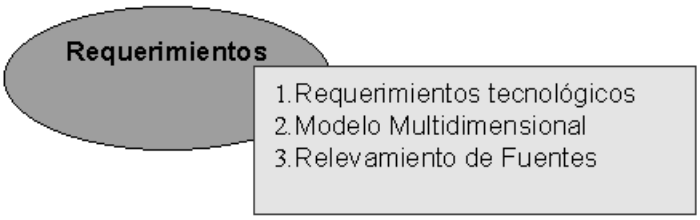


Figura 15: Etapas desarrollo de un data mart

Las etapas en el desarrollo de cada data mart son: requerimientos, análisis, diseño e implementación. En ese orden, a continuación se explican las mismas. Se define el objetivo de cada etapa, las fases que las componen, las tareas a realizar en ellas y la metadata que se debe generar.

Requerimientos



La etapa de requerimientos tiene como objetivos, especificar las características y funcionalidades del data mart y describir el ambiente operativo en que se entregará el mismo.

La lista de requerimientos es grande, pero se resumen en tres puntos fundamentales para construir el data mart realcional y su carga: Requerimientos tecnológicos, Modelo Multidimensional y Relevamiento de Fuentes.

La recopilación de los requerimientos de usuario es realizada por el equipo de desarrollo OLAP y es una de las tareas más importantes en la construcción de un data mart. Dicho equipo a partir de esos requerimientos elabora en la etapa de análisis, un modelo multidimensional que es recibido como un requerimiento más por el equipo de desarrollo encargado del diseño relacional y la carga del data mart. En el diagrama 'Interacción con desarrollo OLAP' (*figura 14*) se aprecia esta interacción.

1. Requerimientos tecnológicos

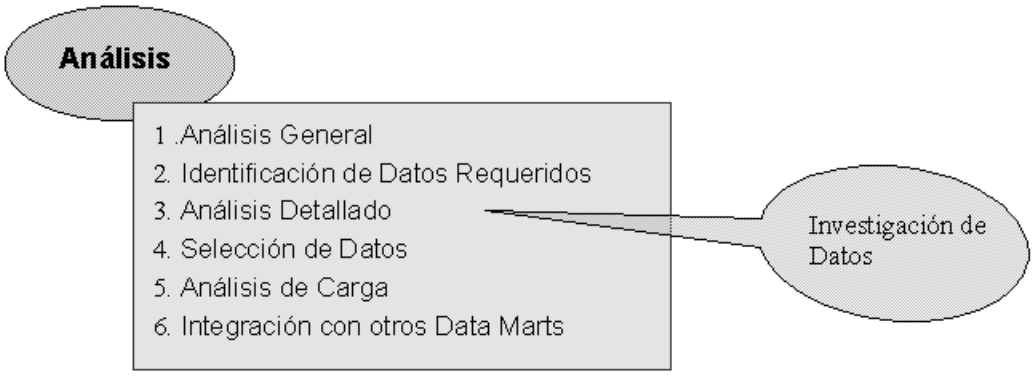
Especifican el ambiente de producción del data mart: procesadores, ambiente operativo, almacenamiento de datos, metadatos, redes, comunicaciones, herramientas disponibles para la extracción, transformación y limpieza de datos, etc.
Metadata a generar: Documentación de requerimientos.

2. Modelo multidimensional

Es construído por el equipo de desarrollo OLAP como resultado de su etapa de análisis y es el pilar del desarrollo del data mart. Del análisis del modelo surje la información que se necesita almacenar en el data mart. Es utilizado para diseñar el data mart relacional y los procesos de carga.
El modelo define las áreas temas de interés, la granularidad de los datos (nivel de detalle de los mismos), las dimensiones (constituídas por los conjuntos de elementos existentes en la realidad) y las mediciones que se quieren realizar sobre los movimientos que sufren las relaciones entre las dimensiones en el tiempo.
Para mas información sobre las características del modelo ver el informe correspondiente a la tesis: 'Sistema DW: OLAP'.
Metadata a generar: Documentación del modelo multidimensional recibido.

3. **Relevamiento de fuentes**
   Especifica los sistemas operacionales disponibles para ser fuente de datos al data mart. Determina la funcionalidad de los sistemas, las características tecnológicas, la información que manejan , su volatilidad y el uso que se les da a los mismos.
   La información se releva mediante el estudio de la documentación de los sistemas operacionales y entrevistas a personas conocedoras de los mismos (administradores, desarrolladores, usuarios).
   Metadata a generar: Documentación de sistemas operacionales.
   Documentación de información recabada.

Análisis



El objetivo de la etapa de análisis es identificar los datos que se necesitan almacenar en el data mart, investigar con qué datos se dispone y en qué estado de calidad se encuentran, para luego definir qué datos se van a extraer de los sistemas operacionales. En caso de no disponer de todos los datos requeridos, se define qué hacer al respecto. También se analizan todos los problemas a enfrentar en el proceso de carga del data mart.

1. **Análisis General**
   Se estudia la información que manejan los sistemas operacionales osea, lo almacenado en sus bases de datos. Se estudian las estructuras físicas de las tablas, los datos que manejan y como éstos se relacionan. Se analizan las reglas de negocio que gobiernan los datos. Esta fase brinda una visión global de la información que mantienen los sistemas operacionales.
   Es importante contar con modelos conceptuales ya que permiten visualizar las relaciones existentes entre los datos. Si no existe documentación de los mismos, se deben construir.
   Metadata a generar: Modelos físicos de las bases de datos fuente.
   Modelos conceptuales de las bases de datos fuente.
   Mapeo entre modelos conceptuales y modelos físicos de las bases de datos (bd) fuente.
   Documentación sobre las reglas de negocio.

2. Identificación de datos requeridos

Se identifican los objetos de los modelos conceptuales de las bd fuente que manejan la información requerida por el modelo multidimensional. Se realiza un mapeo entre los elementos del modelo multidimensional y los objetos de los modelos conceptuales.

Este tipo de análisis permite identificar, *Datos requeridos*: información requerida mantenida en un sistema fuente, *Datos que Faltan:* información requerida no mantenida en ningun sistema y *Datos a Integrar:* información requerida mantenida por más de un sistema. También se pueden identificar datos que si bien no son requeridos por el modelo multidimensional, pueden ser de utilidad para futuros requerimientos. Ver apéndice por información acerca de la técnica utilizada para realizar el mapeo.

Metadata a generar: Mapeo entre elementos del modelo multidimensional y objetos de los modelos conceptuales.

3. Análisis detallado (Investigación de datos)

Siguiendo la medodología para la limpieza de datos, se realiza un análisis minucioso de los datos en cada tabla, estudiando todos los valores posibles en cada campo. La especificación de las actividades que se deben realizar se encuentra en la sección 2.3 Calidad de datos. Las tablas a analizar son aquellas que se correspondan con los objetos identificados como de interés en la fase de identificación de datos requeridos.

El análisis detallado da una visión sobre el estado de calidad en que se encuentran los datos fuente. Permite descubrir si la información que dice manejar el sistema, realmente es la que está almacenada, permite encontrar problemas en los datos y permite conocer la forma real de los mismos. Además se pueden detectar manejos no documentados de los sistemas operacionales.

Metadata a generar: Documentación de resultados de las actividades.

4. Selección de datos

Se seleccionan los datos a extraer de las bases fuente. Se construye un modelo conceptual que permite visualizar en forma integrada el conjunto de datos de los sistemas fuente con los que se cuenta para cargar el futuro data mart.

La selección se realiza en base al estado de calidad de los datos, características tecnológicas de los sistemas fuente, confiabilidad de los datos, volatilidad de la información, accesibilidad de la información y cualquier aspecto considerado de importancia a la hora de optar entre un sistema u otro cuando ambos mantienen la información de forma duplicada e incluso para no optar por ninguno cuando éstos no mantienen la información de forma correcta.

Metadata a generar: Modelo conceptual de integración de datos seleccionados.
Especificación de integración (en caso trabajar con más de un sistema fuente).
Mapeo entre modelo conceptual integrado y modelos físicos de las bd fuente.

5. Análisis de carga

Se analizan los puntos principales que deben considerarse para realizar la carga de datos al data mart. El diseño del proceso de carga debe contemplar todos los puntos analizados.

- *Datos que Faltan*: datos identificados como requeridos pero que no se encuentran en los sistemas fuente o que sí se encuentran pero se descartan por su bajo nivel de calidad.
- *Datos a Integrar*: distintos datos fuente que se corresponden a un mismo objeto del modelo conceptual de datos seleccionados.
- *Datos a Calcular*: elementos del modelo multidimensional (en general medidas) que se calculan a partir de objetos del modelo conceptual de datos seleccionados.
- *Datos a Transformar*: campos de objetos seleccionados cuyos valores deben transformarse a la forma especificada en el modelo multidimensional.
- *Datos a Limpiar*: datos fuente que requieren limpiarse para ingresar al data mart.
- *Granularidad de Datos*: análisis del nivel de granularidad en el que se almacenarán los datos del data mart. El nivel esta dado en un principio por el modelo multidimensional pero puede ser de utilidad almacenar los datos a uno o varios niveles más de granularidad, como también puede no contarse con los datos al nivel requerido.
- *Volatilidad de la Información*: la volatilidad de los datos fuente influye en la periodicidad con que se deben cargar dichos datos al data mart. Se debe analizar la volatilidad de los datos correspondientes a objetos del modelo conceptual de datos seleccionados.
- *Evolución en el Tiempo*: debe analizarse si se puede obtener la evolución en el tiempo de todos los datos requeridos según el modelo multidimensional.
- *Tecnología Disponible*: todos los problemas relacionados a la tecnología de los sistemas operacionales, el data warehouse, vías de comunicación, etc, influyen en diseño del proceso de carga y por lo tanto deben analizarse.
- *Volumen de datos*: el volumen de datos que manejará periodicamente el proceso de carga y que irá haciendo incrementar el volumen de datos del data mart, influye tanto en el diseño del data mart relacional como del proceso de carga.
- *Accesibilidad a los Datos Fuente*: debe analizarse si todos los datos seleccionados de los sistemas operacionales podrán ser extraídos periódicamente. Por problemas organizacionales, políticos, de confiabilidad, etc. no siempre puede contarse con toda la información requerida.

Metadata a generar: Documentación de los puntos mencionados.

6. Integración con otros Data Marts

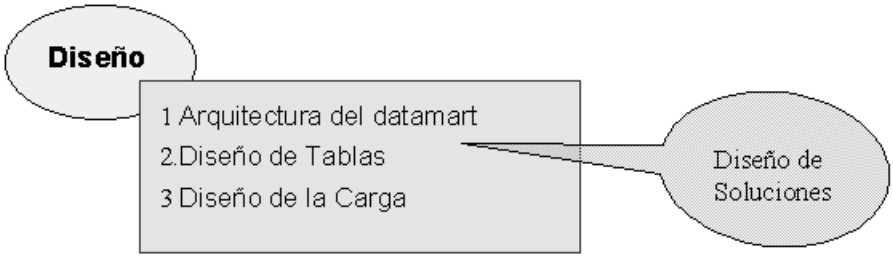
Se analizan posibles problemas de integración con la información almacenada en el data warehouse relacional. Se debe analizar si dentro de los data mart ya construídos se encuentra información que es requerida por el data mart en construcción y/o que pueda entrar en conflicto con el mismo. Se deben analizar posibles soluciones de integración.

Es importante tener una buena documentación de los data marts que constituyen el data warehouse para asi poder acceder a la misma a la hora de realizar este análisis.

En caso de desarrollar varios data marts en paralelo, se deben estudiar los análisis de dichos data marts para detectar puntos de contacto que puedan presentarse y resolver los conflictos.

Metadata a generar: Documentación de conflictos y posibles soluciones.

Diseño



El objetivo de esta etapa es, diseñar el proceso global de carga y diseñar el data mart relacional. Por lo primero se entiende definir el conjunto de procesos que cargan los datos al data mart. Lo segundo consiste en definir las tablas que componen el data mart relacional y cualquier otra estructura auxiliar necesaria.

1. Arquitectura del data mart

Se definen en forma general los componentes del data mart, tanto procesos como estrucuturas de almacenamiento.
La construcción de la arquitectura se ve influenciada por las siguientes variantes: ubicación de los datos fuente, manejador de bd (RDBMS) del data warehouse, herramientas disponibles para desarrollar el proceso de carga, espacio de almacenamiento disponible, complejidad del proceso de carga, etc.
Se toman decisiones acerca de herramientas a utilizar, lenguajes de programación, protocolos de comunicación, manejadores de base de datos, etc.
Metadata a generar: Documentación de componentes del data mart.

2. Diseño de tablas

Se realiza el diseño de las tablas relacionales del Data Mart y de cualquier otra estructura auxiliar necesaria tanto para facilitar la carga como para la limpieza y tansformación de datos.

Para realizar el diseño de las tablas del Datamart, se utiliza el conjunto de primitivas presentado en la tesis 'A transformations based approach for designing the Data Warehouse'. Las primitivas reciben como entrada, esquemas de tablas relacionales y devuelven esquemas de tablas diseñadas adecuadamente para un data warehouse. Además documentan la forma de transformar los datos para pasarlos de la tabla original a la tabla destino del data warehouse.

Es a partir de la total comprensión del modelo multidimensional que se puede seleccionar qué primitiva aplicar a la hora de diseñar el data warehouse. La selección de las primitivas se ve influenciada por:

- nivel de granularidad requerido
- performance de las consultas al data mart
- espacio de almacenamiento disponible
- complejidad y performance del proceso de carga

El proceso de diseño parte de un esquema de tablas relacionales, diseñadas para mantener datos operacionales, a las cuales se les aplican repetidamente las primitivas de transformación, hasta obtener un diseño adecuado para el modelo multidimensional del data mart.

Metadata a generar: Documentación de primitivas aplicadas y proceso de aplicación.
Modelo físico de tablas del data mart.
Documentación del contenido del data mart relacional.

3. Diseño de la carga

Se realiza el diseño de las distintas rutinas de extracción y carga de datos al data mart. Se diseñan rutinas de limpieza de datos que resuelvan los problemas de calidad detectados en la etapa de análisis.

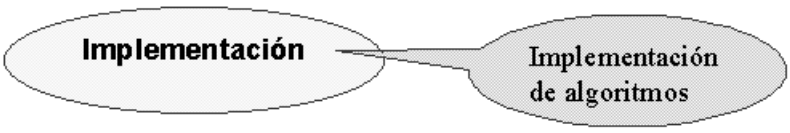
La complejidad del proceso de carga está dada por la complejidad del negocio y del modelo multidimensional junto a la calidad en que se encuentran los datos.

Se define la periodicidad de la carga teniendo en cuenta la volatilidad de los datos en los sistemas fuente y la realidad del negocio.

En caso de utilizar una herramienta que se encarge de realizar toda o parte del proceso de carga, en esta etapa se define la parametrización necesaria en lugar de diseñar las rutinas de carga.

Metadata a generar: Documentación del proceso de carga/Parametrización.

Implementación



En esta etapa se implementan los diseños desarrollados en la etapa anterior. En caso de disponer de herramientas que faciliten el proceso de carga, se parametrizan las mismas.

Metadata a generar: Ubicación y descripción de: servidores, bases de datos, tablas y programas de carga.
Información de herramientas, lenguajes de programación utilizados y estándares aplicados.

3.3 DESCRIPCION TECNICA

El sistema DW es introducido en el capítulo anterior, donde se presentan sus componentes principales. En las secciones 3.1 y 3.2, se plantea el sistema a construír en este proyecto y la estrategia de desarrollo utilizada para realizarlo. Para completar lo que sería una visión global del proyecto, se presenta ahora un resumen de las etapas en la construcción de los datamarts de Presupuesto y Bedelía, haciendo incapié en los resultados obtenidos en cada una de ellas. La descripción detallada se encuentra en el *Apartado 1, Desarrollo del sistema DW*.

Dentro del sistema construído se encuentra el **data warehouse,** diseñado para apoyar la toma de decisiones de las comisiones encargadas del presupuesto de la facultad y del seguimiento de los estudiantes. La información relativa a cada área de negocios se almacena en los distintos **data marts** del data warehouse (Presupuesto y Bedelía). Las **bases de datos operacionales** participan del sistema brindando la información requerida. Diferentes **procesos de carga** transladan los datos de los sistemas operacionales hacia el data warehouse. Los datos sufren una serie de modificaciones antes de ingresar al data warehouse, que incluyen chequeos de calidad, integraciones, estandarizaciones, formateos y transformaciones. La **bd limpia** de Presupuesto apoya al proceso de carga, almacenando los datos fuente luego de haber sido chequeados, integrados y estandarizados. La **bd fuente auxiliar** de Bedelía contiene una copia de algunos datos de la bd del sistema operacional de Bedelía. La **metadata** del sistema documenta la información relativa a la estructura y contenido de las bases de datos involucradas (bd. sistemas operacionales, bd. auxiliares y data warehouse) y los procesos de carga desarrollados. En la *figura 16* se visualiza el sistema construído.

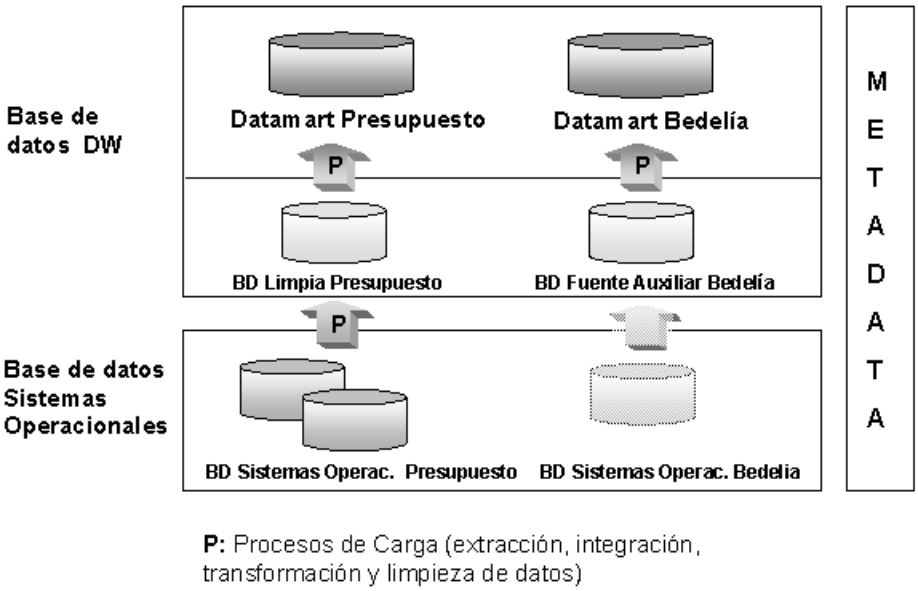


Figura 16: Arquitectura del sistema construído

En un principio se construye el data mart de Presupuesto, generándose la primer versión del data warehouse. Luego se desarrolla el data mart de Bedelía y se lo integra al data warehouse. La integración es trivial porque los datamarts no tienen datos en común. Ambos data mart se desarrollan aplicando la *Estrategia de Desarrollo* propuesta en la sección 3.2.

➢ REQUERIMIENTOS

Los datamarts de Presupuesto y Bedelía deben construirse sobre una base de datos Oracle que se encuentra en un servidor Unix dentro del INCO. El servidor está conectado a una intranet por la cual también se podrá acceder al sistema.

Presupuesto

El modelo conceptual multidimensional de Presupuesto es entregado por el grupo 'SISTEMA DW: OLAP' y tomado como requerimiento en el desarrollo del data mart de Presupuesto. Documenta la información de relevancia para el data mart y de su análisis se deriva la información que deberá almacenar el data warehouse.

El Reelevamiento de Fuentes de Presupuesto, se realiza en base a entrevistas con los funcionarios de la Sección de Personal de la F.I. La misma utiliza dos sistemas operacionales independientes para llevar el presupuesto de la facultad, los llamamos ‘Sistema Original’ y ‘Sistema Nuevo’. Manejan información relativa a los cargos ejecutados (un funcionario ocupa el cargo y recibe un sueldo por ello), cargos vacantes y comprometidos, movimientos de los cargos, funcionarios, institutos de la F.I., fuentes de financiación y relación entre los sueldos de los funcionarios con su grado y cantidad de horas trabajadas. Algunos datos son manejados unicamente por un sistema y otros datos son manejados por ambos sistemas

Bedelía

El modelo multidimensional de Bedelía es entregado por el grupo 'SISTEMA DW: OLAP' y tomado como requerimiento en el desarrollo del data mart de Bedelía.

El Reelevamiento de Fuentes de Bedelía es complejo porque no se tiene acceso a la bd del sistema operacional de Bedelía de la F.I. Se utiliza como fuente de datos para el data mart una base de datos fuente auxiliar donde periódicamente se copia información proveniente del sistema operacional de Bedelía. Dicha bd maneja información de los estudiantes, carreras, materias, inscripciones de estudiantes en carreras y resultado de los estudiantes en las actividades de las materias. El relevamiento se realiza analizando la documentación de la bd. fuente auxiliar.

Por una descripción más detallada de la etapa de requerimientos y los resultados obtenidos ver la sección 1.1 y 2.1 del Apartado 1.

➢ ANALISIS

El análisis es una etapa crucial dentro del desarrollo, todo lo analizado influye en el diseño del data warehouse y su carga, todo lo pasado por alto repercute la feliz culminación del sistema DW.

La figura 17 presenta un esquema general de la etapa de análisis aplicada en el desarrollo del data mart de Presupuesto. Se presenta la misma para el caso de este data mart porque tiene la particularidad de partir de dos sistemas operacionales independientes que manejan información de interés. Se muestran los resultados obtenidos y las flechas que representan la relación entre los mismos.

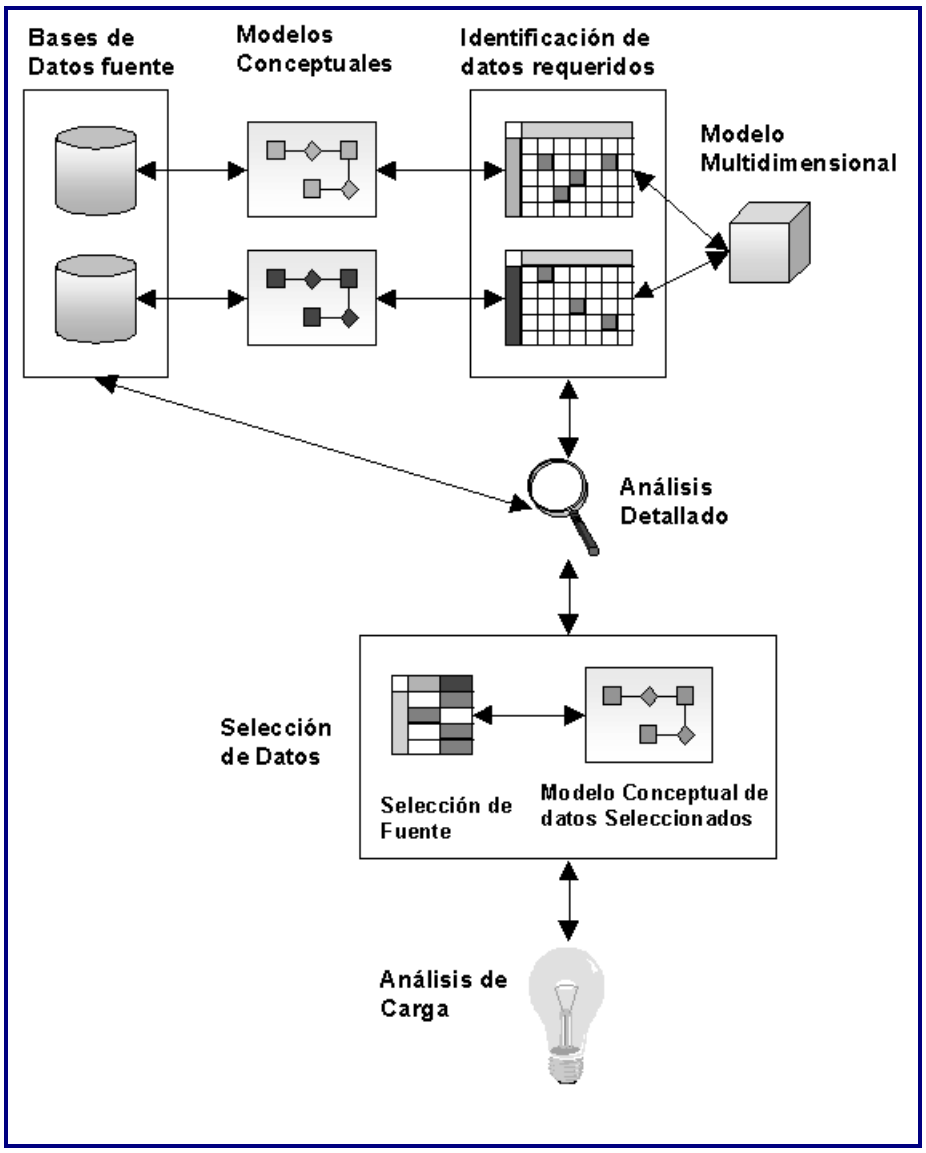


Figura 17: Overview de la etapa de análisis del data mart Presupuesto

Presupuesto

El análisis de presupuesto comienza estudiando las bases de datos fuente, en particular la bd del 'Sistema Original' y del 'Sistema Nuevo'. Se analiza su estructura y contenido. La bd del Sistema Original es completamente desnormalizada. La bd del Sistema Nuevo tampoco es una base normalizada y algunas tablas tienen la misma estructura solo difieren en su contenido. Como no existen modelos conceptuales de las mismas, se construyen aplicando técnicas de reingeniería de datos. Al no poseer documentación de los sistemas, la construcción de los modelos conceptuales se vuelve por momentos compleja.

Una vez obtenidos los modelos conceptuales, se realiza la identificación de datos requeridos, mapeando las entidades del modelo multidimensional (dimensiones y medidas) con las entidades de los modelos conceptualues. En la figura 17 puede observarse que el mapeo se realiza para cada sistema fuente. Con ello se analiza si la información requerida en el modelo mutlidimensional existe en las bases de datos fuente y cuáles son las entidades conceptuales que la manejan. Algunas entidades requeridas no se encuentran en las bd fuente y algunas otras se encuentran en ambas.

El análisis detallado permite visualizar el estado de calidad de los datos fuente y la 'forma' real de los mismos. El Sistema Nuevo supera en calidad al Sistema Original, lo que constituye un resultado decisivo para realizar la selección de datos. El análisis detallado revela que los sistemas operacionales son manejados en forma especial por los usuarios de la Sección de Personal para que cumplan con sus necesidades. Los datos del Sistema Original no cumplen estándares, se encuentra mucha información importante dentro de campos de texto sin formato alguno.

Se realiza la selección de datos fuente teniendo en cuenta la identificación de datos requeridos y el análisis detallado. En la figura puede verse un cuadro que representa la selección de sistema fuente cuando los datos están en más de uno. Se construye como resultado, el modelo conceptual integrado de los datos fuente seleccionados. Este modelo brinda una visión completa de los datos disponibles para cargar el data mart de Presupuesto. Para un mejor entendimiento de la fuente física específica de los datos seleccionados, se documenta el mapeo entre el modelo conceptual y las estructuras físicas correspondiente (de las bd fuente).

Habiendo finalizado la selección de datos, se analizan todos los asuntos relacionados a la carga de los datos fuente al data warehouse. Se utiliza como guía los puntos de análisis de carga presentados en la estrategia de desarrollo. Todo lo analizado influye luego en el diseño del data mart relacional de Presupuesto y su carga.

Al finalizar el análisis se interactúa con el grupo 'SISTEMA DW: OLAP'. Se les informa de los datos requeridos que están disponibles y de los no disponibles. Se discuten asuntos particulares de la carga, algunos cálculos se desiden materializar en el data warehouse para facilitar la implementación multidimensional. Se determinan formatos especiales de los datos, tipos de carga de los cubos(incremental o total), etc.

Bedelía

Se comienza analizando la estructura y contenido de la bd fuente auxiliar de Bedelía en base a documentación de la misma y se construye un modelo conceptual de la misma. La bd tiene una estructura en general normalizada. Algunas tablas tienen la misma estructura y difieren en su contenido según los datos correspondan a estudiantes del plan nuevo, de planes viejos o adaptados al plan nuevo.

A partir del modelo multidimensional y el modelo conceptual se realiza la identificación de datos requeridos. La mayoría de la información requerida se encuentra en la bd fuente.

El análisis detallado brinda una visión del nivel de calidad que tienen los datos fuente. Revela que los datos son de buena calidad, están estandarizados y no presentan grandes problemas.
El análisis de Bedelía fue más sencillo por solo involucrar una base fuente. En éste caso, se Identificaron los datos requeridos, pero la selección de fuentes fue trivial. También resultó simple la construcción del modelo conceptual de datos seleccionados, por no existir más de una fuente.

La selección de datos es relativamente sencilla porque no se presentan problemas de calidad en los datos y solamente se cuenta con un sistema fuente de datos. Se construye un modelo conceptual de los datos seleccionados.

En el análisis de carga se estudian los distintos puntos a considerar en la carga del datamart para que los datos fuente cumplan con la forma requerida por el modelo multidimensional.

Al ser Bedelía el segundo data mart en construírse, se realiza la etapa de integración con otros data marts. En este caso es trivial porque no hay datos en común con el data mart de Presupuesto.

Por una descripción más detallada de la etapa de análisis y los resultados ver la sección 1.2 y 2.2 del Apartado 1.

DISEÑO

En la etapa de diseño se define la arquitectura general del data mart, se diseñan las tablas relacionales y el procesos de carga. Las tablas se diseñan utilizando las primitivas vistas en la sección 2.4.

En la figura 18 se grafica la etapa de diseño de un datamart. Se muestra el caso en que se crea una bd limpia para facilitar el proceso de limpieza si éste es muy complejo (no siempre es necesario). El data mart de Presupuesto es un ejemplo en el que se diseña una bd limpia.

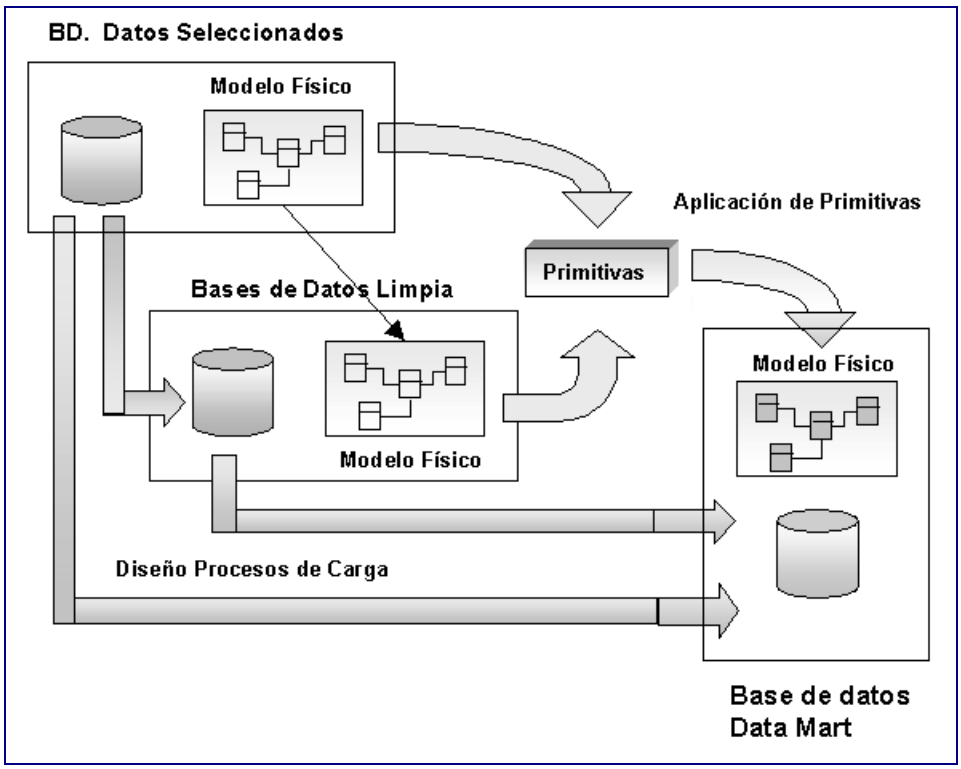


Figura 18: Overview de la etapa de diseño

Presupuesto

Como primer instancia se define la arquitectura del data mart de Presupuesto. Dados los problemas de calidad encontrados en los datos fuente se decide crear una bd limpia (tablas limpias) donde ingresar temporalmente dichos datos luego de haber sido chequeados, integrados, limpiados y verificados que cumplan con las restricciones de integridad. En esa base se ingresan unicamente datos validos.

En el análisis se encontró que algunos datos requeridos no se hallaban en las bd fuente. Se decide almacenar esos datos en el data warehouse (tablas base) y crear procesos para que se ingresen los mismos.
Se diseñan las tablas limpias para almacenar temporalmente los datos fuente luego de ser validados. El diseño de estas tablas se realiza en base a la estructura de las bd fuentes para no complicar el proceso de carga. En la figura 18 la flecha entre el modelo físico fuente y el modelo físico de la bd limpia representa el diseño realizado. El modelo físico fuente se obtiene del mapeo entre el modelo conceptual de los datos seleccionados y las bd fuente de Presupuesto (se realiza en el análisis), representa las estructuras físicas que almacenan los datos seleccionados.

Se diseñan las tablas base de modo que permitan almacenar la información que indica el modelo multidimensional de Presupuesto y que no se encuentra en las bd fuente. Estas tablas forman parte del data mart de Presupuesto, su diseño no se muestra explícitamente en la figura 18.

El modelo físico del data mart de Presupuesto se diseña aplicando el conjunto de primitivas a las estructuras del modelo físico de la bd limpia. En la figura 18 se representa la realización del diseño mediante las flechas anchas que pasan por las primitivas hasta llegar al modelo físico del data mart. Las tablas se diseñan de forma de optimizar en lo posible la consulta a las mismas teniendo en cuenta de no complicar el ingreso de datos que debe realizar periodicamente el proceso de carga. Los nuevos datos se ingresan mediante inserts unicamente.

Se diseñan varios procesos de carga representados en la figura 18 por las flechas celestes rectas. Uno de los procesos lleva ciertos datos seleccionados directamente al data mart, transformándolos y chequéandolos antes de ser ingresados. Otro en cambio extrae el resto de los datos seleccionados, los chequea, integra, limpia e ingresa en la bd. limpia temporal. Por último el siguiente proceso tomará los datos limpios y los transformará para ingresarlos finalmente al data mart. Este último se basa en las sentencias definidas en las primitivas utilizadas para realizar el diseño del data mart.

Los tres procesos de carga se unen en un gran proceso que se debe ejecutar mensualmente.

Bedelía

La arquitectura que se define para el data mart de Bedelía es más simple que la definida para el de Presupuesto. En este caso se decide que los datos fuente se ingresen al data mart directamente, no hay bd limpia intermedia. Sí se definen vistas para unificar las tablas que tienen igual estructura.

El modelo físico del data mart se diseña utilizando las primitivas. Se parte del modelo físico de los datos seleccionados y se aplican las primitivas reiteradamente hasta alcanzar un diseño adecuado al modelo multidimensional de Bedelía (de modo de optimizar las consultas).

El proceso de carga toma los datos seleccionados, los chequea, limpia y transforma para ingresarlos luego al data mart. El proceso se divide en tres subprocesos que se deberán

ejecutar periódicamente en distintos períodos del año. Las inscripciones a carreras se cargan anualmente luego de finalizado el período de inscripciones. Los resultados de las actividades se cargan luego de finalizados los períodos de feb-marzo y julio-agosto. El desempeño de los estudiantes se carga anualmente luego de finalizado cada año escolar (finaliza en marzo).

Todos los procesos de carga (de ambos data marts) ingresan en un log de errores los problemas encontrados en los datos fuente. Se ingresa el tipo de error y el registro de datos fuente. El tipo de error indica si los datos fueron ingresados al data mart a pesar del problema o directamente descartados por ello.

Por una descripción detallada de la etapa de diseño de ambos data marts ver las secciones 1.3 y 2.3 del Apartado 1.

IMPLEMENTACION

El data warehouse se implementa en una base de datos Oracle, que se encuentra en el servidor Unix ('Felipe') del INCO.
Los procesos de carga de ambos data marts se implementan en PLSQL y se almacenan en la misma bd del data warehouse.
Se genera una aplicación, desarrollada en Visual Basic, como interfase para ejecutar los procesos de carga. La aplicación unifica todos los procesos de carga y brinda abm´s para ingresar ciertos datos de presupuesto que no se encuentran en las bd fuente. Permite visualizar la lista de errores encontrados durante los procesos de carga en los datos fuente.

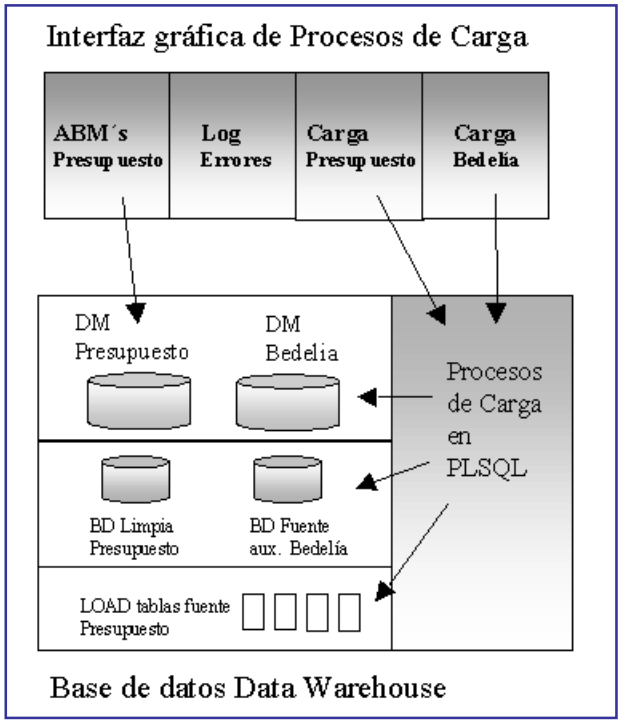


Figura 19: Implementación

En la figura 19 se distinguen las diferentes funcionalidades que brinda la aplicación gráfica de los Procesos de Carga. Los distintos ABM´s trabajan directamente con las tablas correspondientes en el data mart de Presupuesto. La interfaz gráfica de las cargas de Presupuesto y Bedelía llaman a procedimientos PLSQL almacenados en la bd del data warehouse. Los procedimientos PLSQL se encargan de realizar todos los procesos de carga. Las flechas salientes señalan los niveles de componentes del data warehouse con los que trabajan los procesos.

En la secciones 1.4 y 2.4 del Apartado 1 se encuentra una descripción detallada de lo implementado. En la secciones 5 y 6 del Apartado 2 se encuentran la definición de la bd del data warehouse y los pseudocódigos de los procesos de carga implementados en PLSQL respectivamente. El apartado 3 describe cómo usar la aplicación de interfaz gráfica de los procesos de carga.

CONCLUSIONES

4

4.1 CONCLUSIONES FINALES

Al llegar al final del proyecto, podemos decir que el desarrollo de un Sistema de DW es complejo. Abarca gran cantidad de personas, usuarios finales del DW, usuarios de los sistemas fuente, desarrolladores, etc. También comprende diferentes componentes, sistemas fuente, herramientas de extracción, herramientas de consulta, redes, aplicaciones de usuario final, bases de datos, etc. Un Sistema de DW debe combinar todos éstos elementos y brindar un producto final consistente, amigable y confiable que a su vez esté preparado para enfrentar los continuos cambios que surjan debido a su tan variada estructura. Un trabajo no trivial.

El desarrollo de un sistema de DW es diferente al desarrollo de un sistema operacional. A grandes rasgos, uno apoya al negocio transaccional y el otro al decisional. El área de DW es un terreno nuevo en el cual queda mucho por experimentar. No existe aún, una metodología universalmente reconocida para construir un sistema de esta índole. Es por esto que construímos una estrategia de trabajo que sirva de guía durante el desarrollo. La misma, fue elaborada por nosotros a lo largo del proyecto y actualizada a medida que la ibamos experimentando. Consideramos la estrategia un aporte interesante al proyecto que puede ser de utilidad a futuros desarrollos.

No disponer de herramientas de apoyo específicas para la construcción de un sistema de DW como ser, herramientas de extracción transformación e integración de datos, detectores de reglas en los datos, reconocedores de patterns, etc., agregó complejidad al desarrollo, afectó los tiempos del mismo y privó de experimentar nuevas tecnologías existentes en la actualidad. De todas formas se desarrolló un sistema lo más completo y confiable posible.

MODELO MULTIDIMENSIONAL

El modelo multidimensional resultó de suma importancia a la hora de construir los data marts. Especifica los puntos fundamentales para realizar el diseño del data mart relacional y su carga.

La dimensión tiempo, siempre presente, influye cualitativamente en el diseño del data mart relacional. El volumen de datos a almacenar en el data mart depende directamente de esta dimensión. No siempre es posible cumplir con el nivel de granularidad requerida (día, semana, quincena, etc.) en el modelo multidimensional, para ésta dimensión. Dependerá de la capacidad de almacenamiento, la volatilidad de los datos fuente y la complejidad de los cálculos.

En el comienzo del proyecto, contábamos unicamente con los requerimientos de usuario. No estaba definida la interacción con el desarrollo OLAP y entonces se decidió comenzar con el análisis a partir de los requerimientos de usuario, mientras se esperaba el modelo multidimensional. Con esa información, el diseño del data mart era complejo de realizar.

Se pensó, hacer un modelo conceptual relacional del data mart para comenzar con el diseño. Pudimos experimentar que dicho modelo conceptual no tiene la capacidad para modelar los datos variantes en el tiempo y tampoco es eficaz para modelar los cruzamientos entre todas las dimensiones.

En este proyecto utilizamos finalmente el modelo conceptual definido en el trabajo 'Modelo Conceptual Multidimensional' presentado por el profesor Ing. Fernando Carpani en su tesis de maestría. Independientemente del modelo específico a utilizar, se encontraron algunas características mínimas necesarias que debe tener dicho modelo para ser de utilidad. Debe:

- definir las dimensiones (o perspectivas de análisis), especificando su significado y la información que manejan
- definir las medidas (o indicadores), especificando su significado
- definir los cruzamientos, indicando las medidas interesantes a obtener en cada uno de ellos (no siempre interesan todas las medidas para todos los cruzamientos)
- especificar el mayor nivel de granularidad por el que se quieren ver las medidas

Actualmente en los desarrollos de sistemas de DW se manejan diferentes modelos, no existe uno universalmente aceptado. La estrategia de desarrollo planteada en esta tesis es independiente del modelo, pero si en un futuro surje modelo radicalmente diferente podría ser necesario modificarla.

DIVISION HORIZONTAL DEL TRABAJO

En este proyecto se experimentó una forma de trabajo en particular, donde el sistema de DW se construye como dos desarrollos paralelos que interactúan en su camino. Es una forma interesante de trabajar, por lo menos en lo que respecta al desarrollo del data mart relacional y su carga. Las tareas que nos competen, pueden ser complejas y necesitar de varios meses para su elaboración, lo mismo ocurre con respecto al desarrollo OLAP. Trabajando en paralelo, se reduce el tiempo total de construcción del sistema y es posible centrar la atención en los puntos importantes dentro de cada desarrollo.

Fue un primer acercamiento a esta forma de trabajo y por lo tanto quedan elementos por definir. La interacción entre ambos equipos debe discutirse más en detalle, evitando trabajos duplicados. La pieza fundamental en la interacción es el modelo multidimensional. Debe profundizarse en temas como, el momento en que el grupo de desarrollo OLAP entrega el modelo, si el modelo debe ser armado considerando los requerimientos de usuario y conociendo los datos que manejan los sistemas operacionales, o solamente a partir de la realidad del negocio. Estos puntos influyen en el proceso de desarrollo del sistema y su duración.

Es interesante destacar, que la segunda interacción, donde se comunica al desarrollo OLAP si se cuenta realmente con todos los datos requeridos, puede tener un impacto importante en el diseño de los cubos multidimensionales. O sea, es probable que los cubos a diseñar no cumplan directamente con lo analizado en el modelo multidimensional porque la información no existe. En cambio, dicha segunda interacción no influye de manera tan importante a este

grupo de desarrollo. El diseño se realiza principalmente en base a los resultados de la etapa de análisis, teniendo en cuenta las consideraciones que surjen de la interacción.

ESTRATEGIA DE DESARROLLO

Tener una estrategia de desarrollo es fundamental para construir un sistema de DW. Con una estrategia definida, es posible planear la totalidad del desarrollo, estimar plazos, organizar las tareas. Sirve como guía, marcando los objetivos en cada etapa, definiendo las tareas a realizar y la documentación a elaborar. Un sistema de DW significa mucho trabajo de análisis, por cierto complejo en muchos casos. La estrategia pone énfasis en esta etapa, permitiendo realizar un análisis en forma completa y detallada. También destaca la importancia de la calidad de los datos en el data warehouse, definiendo tareas específicas para alcanzar este fin.

La estrategia planteada es un primer acercamiento a la definición de un proceso de desarrollo de un sistema de DW. Como se explicó en su presentación, se centra en la construcción del data mart relacional y su carga. Consideramos importante continuar la formalización de la estrategia para poder alcanzar un proceso de desarrollo más automatizado. La formalización en la especificación de la Metadata a generar en el desarrollo es otro punto interesante de estudio. La interacción con el grupo OLAP es otro aspecto a mejorar el cual se irá perfeccionando con la puesta en práctica de la estrategia.

DISEÑO DEL DATA WAREHOUSE RELACIONAL

Si desarrollamos un sistema operacional, sabemos con certeza diseñar su base de datos relacional. En ese caso, el diseño debe permitir almacenar información sin redundancia innecesaria y a la vez permitir recuperar la información fácilmente. Son universalmente conocidas las técnicas para diseñar bases de datos normalizadas, por ejemplo utilizando dependencias funcionales.

Sin embargo para los sistemas de DW las técnicas de diseño están aún en investigación y experimentación. Una base de datos normalizada no es una cualidad en este caso, sino todo lo contrario, el objetivo principal pasa a ser el acceso rápido a los datos. Podría pensarse entonces que un diseño a imagen y semejanza del modelo multidimensional sería lo mejor, nos referimos a tener una tabla por cada dimensión y una gran fact table con todos los cruzamientos y medidas. La carga de los cubos multidimensionales sería óptima, pero qué hay de la carga de los datos fuente a dichas tablas? No siempre se cuenta con todos los datos al mismo tiempo, algunos datos se calculan a partir de otros, no todas las medidas tienen sentido para todos los cruzamientos de dimensiones posibles, etc. Por todo esto, la carga del data warehouse se volvería compleja, de baja performance y difícil de mantener, teniendo que actualizar datos, realizar cálculos complejos, ingresar valores dummys, etc.

El punto justo está entonces en diseñar el data warehouse de forma de optimizar las consultas sin complicar su carga y no olvidar que el data warehouse debe almacenar todo lo especificado en el modelo multidimensional. Es aquí donde juegan un papel importante las primitivas documentadas en el trabajo 'A transformations based approach for designing the Data Warehouse'. Las mismas resumen las técnicas de diseño existentes y están especificadas de forma clara por lo que son fáciles de utilizar. Las primitivas brindan un conjunto de soluciones de diseño, el diseñador solamente debe encontrar la que sea más apropiada para el caso en particular y aplicarla. Se parte de la estructura física de las tablas fuente y se aplican las primitivas reiteradamente de forma de transformar el diseño de las tablas fuente a tablas con un diseño adecuado para un data warehouse.

¿ Qué se necesita para poder utilizar las primitivas ? Conocer las estructuras de tablas donde se encuentran almacenados los datos fuente seleccionados. Entender el contenido de dichas tablas fuente y comprender perfectamente el modelo multidimensional.

En la experiencia obtenida durante este proyecto observamos que si bien las primitivas pudieron ser aplicadas en la mayoría de los casos, existían situaciones que las mismas no consideraban. Resumiendo, están pensadas para ser aplicadas a diseños normalizados de bases de datos y no siempre los sistemas operacionales cuentan con dicha cualidad. A su vez, especifican las transformaciones que deben realizarse a los datos pero suponen que los datos fuente son ‘limpios’ y no necesitan ser validados. Entonces, cuando se está frente a una fuente de datos de bajo nivel de calidad, para poder utilizar las primitivas, primero se debe crear una ‘data staging area’ (o base de datos limpia) donde ingresar los datos limpios y aplicar las primitivas a esas tablas, fue el caso del data mart de Presupuesto. Para el caso de Bedelía, los problemas de calidad no eran graves y no fue necesario crear tablas limpias, sin embargo las transformaciones especificadas en las primitivas no eran suficientes para diseñar el proceso de carga ya que igualmente se necesitaban realizar validaciones, filtros y transformaciones a los datos fuente.

Quedan asuntos interesantes por discutir, como ser qué almacenar en el data warehouse ¿datos primitivos o datos calculados?, ¿datos al mayor nivel de granularidad especificado en el modelo multidimensional o a otro nivel aun mayor (si es posible)? Son preguntas que surgieron en este proyecto y para las que se tomaron soluciones adecuadas a cada caso. Pero son también preguntas que pueden aparecer en cualquier otro caso y sobre las que se encuentran respuestas dispares.

CALIDAD

Durante la investigación realizada sobre el tema calidad, descubrimos la importancia que ésta tiene en los sistemas de DW. Datos que pueden ser aceptables para un sistema operacional pueden ser inutilizables para un sistema DW. Es un tema que no se tuvo en cuenta en el inicio de la tecnología DW y que surgió a partir de muchos fracasos millonarios de proyectos por esta causa. En un futuro los sistemas operacionales podrán ser creados para que sus datos sean fácilmente utilizados por un sistema DW pero en la actualidad el reto está en tratar de aprovechar los billones de datos que ya hay almacenados.

En este proyecto se dan pautas de cómo construir un data warehouse de calidad y cómo lograr la calidad a nivel empresarial. El primer punto fue considerado durante la construcción de ambos data marts (presupuesto y bedelía), e incluso en la estrategia de desarrollo planteada.

El segundo punto, fue tenido en cuenta en la medida de lo posible, pero en general las soluciones planteadas estaban fuera del alcance de este proyecto, no teniendo la autoridad necesaria sobre los sistemas operacionales para realizarlas. Cabe resaltar que igualmente, todo lo documentado sobre dichos sistemas es un importante aporte para la reestructuración y mejora de los mismos. Además, se informó a los encargados de los sistemas de la sección de Personal la importancia de ingresar datos correctos y en fecha y se dieron sugerencias para mejoramiento de calidad a largo plazo (modificar el sistema) y a corto plazo (cambios en ciertas operativas del mismo). Por último, como feedback entre el usuario final del data warehouse y los usuarios de los sistemas operacionales se implementó un log de errores adecuado para poder identificar claramente problemas en los datos fuente.

Dada la importancia de la calidad y el auge de los sistemas de DW, podrían crearse talleres para profundizar el tema. Actualmente están surgiendo en el mercado herramientas que brindan soluciones para limpiar datos, transformarlos, descubrir reglas en los datos, etc. Sería interesante investigar dichas herramientas y experimentar de construir un data warehouse utilizándolas. Por otro lado, sería también interesante desarrollar soluciones propias, automatizando algunos puntos de la metodología de limpieza planteada. Por ejemplo, que se almacenen en la bd las reglas de negocio y que el proceso de carga verifique que los datos cumplan esas reglas. Sería interesante estudiar la potencialidad que tiene el metadata del sistema sobre la limpieza de datos. Almacenando cierta metadata especifíca, en una forma adecuada permitiría realizar varios chequeos en forma automática.

El análisis detallado de los datos es una tarea muy importante dentro del desarrollo que no debe dejarse de lado pero también es una tarea engorrosa de realizar si no se cuenta con herramientas que faciliten su realización. Están surgiendo varios productos en el mercado que intentan realizar esta labor o parte de ella [P6, P9, P11]. Consideramos importante evaluar estos nuevos productos y su utilización en proyectos DW en nuestro medio.

Definir métricas para medir la calidad de los datos fuente sería un complemento útil. La selección (y por lo tanto descarte) de datos se basa en parte en los resultados obtenidos en el análisis detallado. Una métrica del estado de calidad de los datos brindaría argumentos contundentes sin tener que ‘bajar’ a los problemas particulares en los datos. Es posible que se dé el caso de descartar una base de datos fuente entera por problemas de calidad y los fundamentos de tal decisión deben poder ser comprendidos por personas tanto de perfiles técnicos como gerenciales. El argumento no puede ser ‘la calidad es mala’ y tampoco un reporte con el análisis detallado. Las métricas además dan información de gran utilidad para estimar la complejidad y tiempo de desarrollo de los procesos de carga. Para definir la métrica se deben agrupar los problemas de calidad en categorías y definir las medidas interesantes sobre cada una de ellas. A su vez deben definirse los niveles mínimos o máximos (depende de que se esté midiendo) de aceptación en cada una. Por ej. si calculadas las medidas la mayoría tienen niveles aceptables la base de datos se toma como fuente al DW y en caso contrario no.

TRABAJO FUTURO

- Investigación del tema Metadata, cómo aprovechar al máximo su información.
- Administración de un sistema DW. ¿Cómo enfrentar los cambios del negocio y de los sistemas operacionales?
- Investigación y desarrollo de técnicas de diseño del data warehouse que se adapten a los cambios.
- Reutilización en sistemas DW y la relación con la metadata.
- Tunning del data warehouse.
- Optimización de procesos de carga. Que conviene más: utilizacion de data staging relacional o procesamiento secuencial.
- Planeamiento del desarrollo, estimación de tiempos, recursos y costos.
- Experimentación de nuevas tecnologías en el área.
- Profundizar en la técnica de identificación de datos requeridos.

4.2 AGRADECIMIENTOS

Queremos agradecer de forma especial la colaboración brindada en el transcurso del proyecto por los profesores Raul Ruggia, Adriana Marotta y Joaquin Goyoaga. Sus aportes e inquietudes fueron de gran utilidad para nosotros. También agradecer el enorme apoyo recibido por parte de la sección de Personal de la Facultad de Ingeniería. Gracias por recibirnos y facilitarnos toda la información a su alcance para poder llevar a cabo nuestra tarea. Debemos agradecer también a nuestros lugares de trabajo por la flexibilidad e interés demostrados. Por último, queremos agradecer a nuestros familiares y amigos, su apoyo incondicional y su fuerza nos ayudaron en los momentos más díficiles. Muchas Gracias.

4.3 BIBLIOGRAFIA

➢ LIBROS

[L1] Ralph Kimball. 'The data warehouse Toolkit'. 1996

[L2] Ralph Kimball. 'The data warehouse Lifecycle Toolkit'. 1998

[L3] Hardjinder , Gil, Rao. 'Data warehousing: La integración de información para la mejor toma de decisiones'. 1996

[L4] Len Silverston, W.H. Inmon, Kent Graziano. 'The datamodel resource book'.

[L5] W.H. Inmon. 'Building the data warehouse'.

[L6] Batini, Ceri, Navathe. 'Diseño conceptual de bases de datos', Un enfoque de entidades-interrelacionales. 1994

[L7] Robert Orfali, Dan Harkey, Jeri Edwards. 'The Essential Client/Server Survival Guide'. 1996

➢ ARTICULOS

[A1] Richard Hackathorn. 'Data Warehousing´s Credibility Crisis'. [*BYTE Magazine, Agosto 1997*]

[A2] Ralph Kimball. 'Mastering Data Extraction'. [*DBMS, Junio 1996*]

[A3] Ralph Kimball. 'Automating Data Extraction'. [*DBMS, Julio 1996*]

[A4] Ralph Kimball. 'Dealing with Dirty Data'. [*DBMS, Setiembre 1996*]

[A5] Ralph Kimball. 'It´s time for Data Compression'. [*DBMS, Octubre 1996*]

[A6] Joseph Williams. 'Tools for Traveling Data'. [*DBMS, Junio 1997*]

[A7] Kathy Bohn. 'Converting Data for Warehouses'. [*DBMS, Junio 1997*]

[A8] Ralph Kimball. 'It´s Time for Time'. [*DBMS, Julio 1997*]

[A9] Ralph Kimball. 'A Dimensional Modeling Manifesto – Drawing the Line Between Dimensional Modeling and ER Modeling Techniques'. [*DBMS, Agosto 1997*]

[A10] Ralph Kimball. 'Is Data Staging Relational?'. [*DBMS, Abril 1998*]

[A11] Jill Dyche. 'Scoping Your Datamart Implementation'. [*DBMS, Agosto 1998*]

[A12] Michele Bokun, Carmen Taglienti. 'Incremental Data Warehuose Updates: Approaches and Strategies for Capturing Changed Data'. [*DM Review, Mayo 1998*]

[A13] Neil Raden. 'Data, Data Everywhere'. [*Information week, Octubre 1995*]

[A14] Neil Raden. 'Modeling The Data Warehouse'. [*Information week, 1996*]

[A15] Larry P. English. 'Help For Data Quality'. [*Information week, Octubre 1996*]

[A16] Andy Feibus. 'An Easy-to-Use Tool For Data Transformation'. [*Information week, 1999*]

[A17] Wayne W. Eckerson. 'Building the Legacy Systems of Tomorrow. Recipes of Prevention and Integration'. [*Open Information Systems, Noviembre 1995 – Diciembre 1995*]

[A18] Rosemary Cafasso. 'Quality Control'. [*PC Week, Junio 1996*]

➢ PUBLICACIONES

[P1] The applied Technologies Group. 'Meeting the Data Integration Challenge'.[1997] [*www.techguide.com*]

[P2] The applied Technologies Group. 'A practical Guide to Achieving Enterprise Data Quality'. [1998][*www.techguide.com*]

[P3] The applied Technologies Group. 'Right Sizing Your Data Warehouse'. [1998] [www.techguide.com]

[P4] The applied Technologies Group. 'Managing the Warehouse'. [1998] [*techguide.com*]

[P5] The applied Technologies Group. 'Putting Metadata to Work in the Warehouse'. [1998] [*www.techguide.com*]

[P6] Vality Technology Inc. 'The Five Legacy Data Contaminants That You Will Encounter In Your Data Migrations'. [1998] [www.vality.com]

[P7] Duane Hufford. 'Quality and the Data Warehouse'. [1998] [www.data warehouse.com]

[P8] Knosys Inc. 'Transforming Data into Understanding'. [1998] [*www.data warehouse.com*]

[P9] Kamran Parsaye, Mark Chignell. 'Quality Unbound'. [1998]

[*www.data warehouse.com*]

[P10] George Burch. 'Five Common Excuses For Not Reengineering Legacy Data'. [*www.datawarehouse.com*]

[P11] Abraham Meidan. 'Wiz Rule: A New Approach to Data Cleansing'. [1998] [*www.data warehouse.com*]

[P12] Louis Rolleigh and Joe Thomas. 'Data Integration: The Warehouse Foundation'. [1998] [*www.data warehouse.com*]

[P13] Bob Lambert. 'Data Warehousing Fundamentals: What You Need to Know to Succed' [1998] [*www.data warehouse.com*]

[P14] Charles B. Darling. 'How To Integrate Your Data warehouse'. [1996] [*www.datamation.com*]

[P15] Anne Knowles. 'Dump your dirty data for added profits'. [1997] [*www.datamation.com*]

[P16] Esprit Project 22469 – DWQ. 'Foundations of Data Warehouse Quality' [1998] [*www.cordis.lu/esprit/home.html*]

[P17] Ron Dvir, Dr. Stephen Evans. 'A TQM Approach to the Improvement of Information Quality'.

[P18] Sophie Allen. 'Name and Address Data Quality'. [*www.mastersoft.com*]

[P19] K.I. Gordon. 'The why of data standards - Do you really know your data?'. [*www.island.net/~gordon*]

[P20] IBM Corporation. 'The Road to Business Intelligence'. [*www.ibm.com*]

[P21] Veronika Peralta, Alvaro Illarze. 'Estudio de Técnicas y Software para la Construcción de Sistemas de Data Warehousing'. [1998] [*Facultad de Ingeniería, Universidad de la Republica*]

[P22] Adriana Marotta. 'A transformations based approach for designing the data warehouse'. [1999] [*Facultad de Ingeniería, Universidad de la Republica*]

[P23] Fernando Carpani. 'Modelo Conceptual Multidimensional'. [1999] [*Facultad de Ingeniería, Universidad de la Republica*]

[P24] Osvaldo Varallo, Andrea Pereira. 'Sistema Data Warehousing: OLAP'. [1999] [*Facultad de Ingeniería, Universidad de la Republica*]