La construcción de la factoría de información corporativa

PID_00270641

Alberto Abelló Gamazo
Josep Curto Díaz
José Samos Jiménez
Juan Vidal Gil
David Díaz Arias

Tiempo mínimo de dedicación recomendado: 7 horas

Universitat Oberta de Catalunya

Alberto Abelló Gamazo

Doctor e ingeniero en Informática por la Universidad Politécnica de Cataluña. Profesor asociado al Departamento de Lenguajes y Sistemas Informáticos de esta universidad. Coordina en la UPC el programa de doctorado Erasmus Mundus IT4BI-DC. Sus intereses de investigación se centran en el área de bases de datos, Business Intelligence, gestión de *Big Data*, flujos de datos y gestión de metadatos.

Josep Curto Díaz

Licenciado en Matemáticas por la Universidad Autónoma de Barcelona, máster en Business Intelligence y Dirección y Gestión de las Tecnologías de la Información por la Universitat Oberta de Catalunya, y MBA por el Instituto de Empresa Business School. Trabaja en los ámbitos de *Business Intelligence*, *Business Analytics* y *Big Data*. Desde 2014 en Delfos Research, empresa de la que es fundador, compagina esta actividad con colaboraciones docentes en IE Business School, UOC, EOI, U-TAD, IEB y Kschool.

José Samos Jiménez

Doctor en Informática por la Universidad Politécnica de Cataluña. Profesor titular del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada, asignado a la Escuela Técnica Superior de Ingeniería Informática.

Juan Vidal Gil

Licenciado en Físicas por la Universidad Complutense de Madrid. Experiencia en soluciones tecnológicas de *Business Intelligence* y *Data Warehouse*, como jefe de proyectos en importantes compañías y como formador especializado en empresas del sector. Profesor colaborador de la UOC.

David Díaz Arias

Ingeniero en Informática por la UOC. Ingeniero Técnico en Informática de Gestión por la UAB. Responsable técnico y analista de datos del área de Business Intelligence en una empresa del ámbito de salud. Profesor Colaborador de la Universitat Oberta de Catalunya.

La revisión de este recurso de aprendizaje UOC ha sido coordinada por la profesora: Àngels Rius Gavidia

Primera edición: febrero 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Índice

Introducción

La factoría de información corporativa (FIC) permite a los analistas disponer de la información que necesitan como apoyo a la toma de decisiones. Aun así, generalmente, la FIC no se puede considerar como un producto o una aplicación empaquetada que se pueda adquirir y, una vez instalada en nuestras organizaciones, empiece a «fabricar» información a partir de los datos de las fuentes de datos operacionales.

Generalmente, la FIC no se puede adquirir, se tiene que construir en las diferentes organizaciones. Podemos plantear la construcción de la FIC según distintos enfoques, y también podemos considerar variantes en la arquitectura de la FIC. En cualquier caso, implementar esta arquitectura o sus variantes no es una operación trivial.

En este módulo estudiaremos diferentes enfoques alternativos para la construcción de la FIC o sus variantes. Revisaremos las implicaciones que tiene empezar la construcción desde los almacenes de datos departamentales y crear posteriormente el almacén de datos corporativo.

En la ejecución de proyectos veremos las dificultades que conlleva crear la FIC en un único proyecto y la alternativa de crear la FIC según un conjunto de proyectos autónomos. Analizaremos el equipo y perfiles necesarios en la construcción de la FIC y la evolución del entorno operacional una vez construida esta.

Posteriormente, se abordará la construcción del componente de integración y transformación, teniendo en cuenta la construcción de los diferentes componentes (transformación, depuración e integración) y haciendo hincapié en cuestiones como la frecuencia y la ventana de actualización. Asimismo, se analizará el papel de las herramientas de apoyo en la construcción de este componente.

Por último, se estudiará en detalle la construcción de los distintos tipos de almacén de datos: departamental, operacional y corporativo. Se revisarán cuestiones tales como el diseño del modelo de datos, la definición de la granularidad, la organización de los datos, su aprovisionamiento y refresco, así como la adicción del elemento temporal que permita la historificación de los datos.

Objetivos

En este módulo se pretende ofrecer una visión global del proceso de construcción de la FIC y de la construcción de sus componentes. Mediante el estudio, se conseguirán los objetivos siguientes:

1. Entender la problemática que supone para las organizaciones que no disponen de FIC como soporte para la toma de decisiones y ver de qué manera se puede transformar un sistema operacional en uno decisional.

2. Conocer los problemas que surgen al tratar de implementar variantes de la FIC en organizaciones. Comprender las ventajas e inconvenientes de los diferentes enfoques.

3. Determinar cómo se puede estructurar el proceso de construcción de la FIC en forma de proyectos. Comprender la dificultad del planteamiento en un único proyecto y saber cómo implementar la FIC en un conjunto de proyectos autónomos.

4. Tener presente la evolución del entorno operacional en el desarrollo de la FIC.

5. Comprender los nuevos roles que aparecen en los equipos de desarrollo que intervienen en la construcción de la FIC.

6. Conocer el proceso de construcción del componente de integración y transformación, desde la construcción de cada componente (transformación, depuración e integración) hasta la planificación de los procesos de actualización de datos.

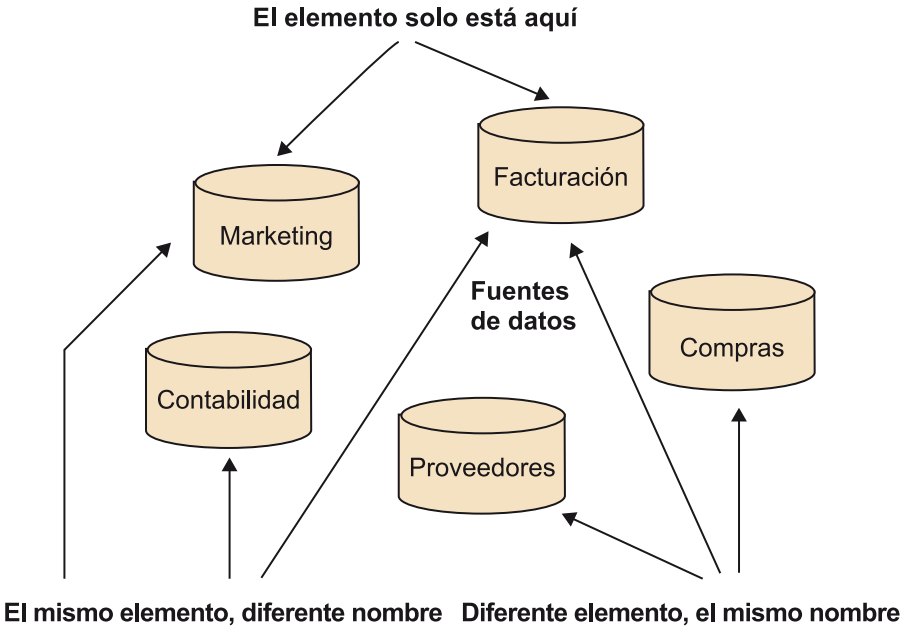
7. Conocer cómo implementar los diferentes tipos de almacenes de datos: departamental, operacional y corporativo. Saber definir el modelo de datos, su organización, el aprovisionamiento de datos y cómo añadir el elemento temporal.

1. Transformación de datos desde el entorno operacional al decisional

1.1. Apoyo a la toma de decisiones desde el entorno operacional

En el módulo anterior, hemos estudiado las características del entorno operacional. Sabemos que los datos en un entorno operacional generalmente están orientados a las aplicaciones o a la funcionalidad y desintegrados, además de ser volátiles y no históricos.

Figura 1. Entorno operacional



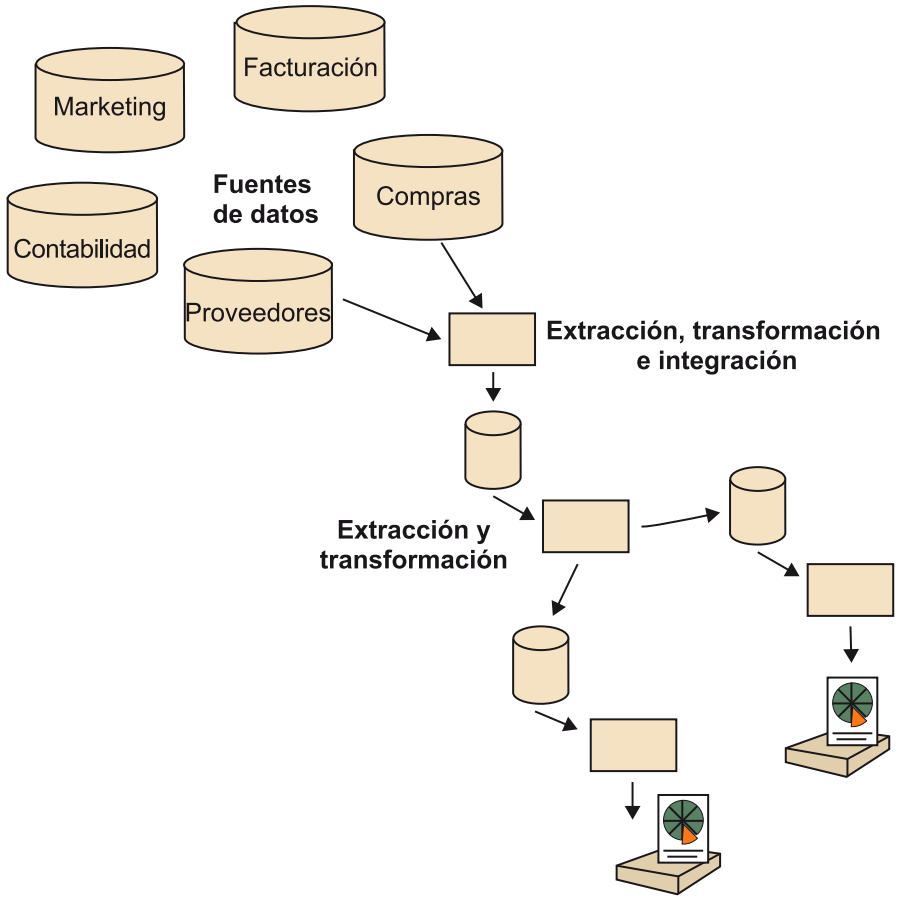
Cada aplicación operacional ofrece las funcionalidades operacionales para las que ha sido diseñada. Además de estas funcionalidades, o como parte de estas, generalmente también permite hacer consultas sobre sus datos y generar informes a partir de estos, habitualmente informes preestablecidos. Estos informes son los que pueden usar los analistas como apoyo en el proceso en que se toman decisiones.

En algunos casos, con la información que se obtiene en los informes preestablecidos no basta y se requiere la generación de nuevos informes, posiblemente con la inclusión de datos de más de una aplicación. Entonces, la solución consiste en desarrollar un conjunto de programas de extracción, transformación e integración de datos cuyo resultado sea el informe deseado. General-

mente, estos programas son desarrollados por el Departamento de Informática de la organización a petición de los analistas que usarán la información obtenida.

Los programas desarrollados transforman e integran los datos obtenidos de las bases de datos de las aplicaciones y generan bases de datos intermedias, o bien exclusivamente transforman los datos de las aplicaciones para adaptarlos a la estructura requerida por los analistas. Para generar un informe, pueden ser necesarios distintos pasos de transformación e integración de datos. En la medida en que se pueda, se trata de reutilizar el trabajo realizado durante el desarrollo de informes previos, por ejemplo, accediendo a alguna de las bases de datos intermedias generadas. De este modo, se ahorra tiempo de desarrollo del informe y de ejecución para obtener los datos requeridos.

Figura 2. Obtención de informes en el entorno operacional



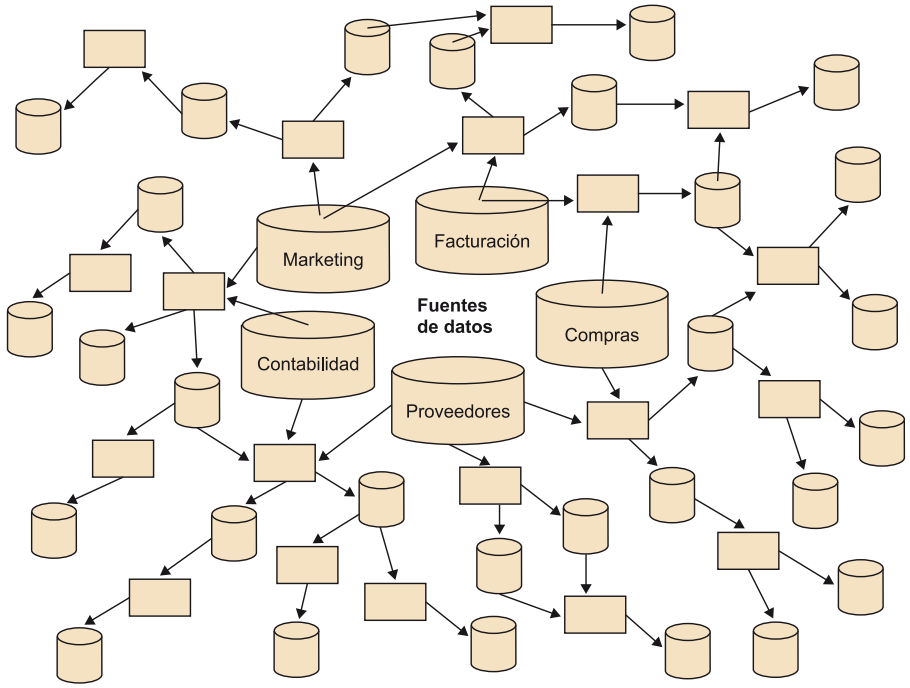
Los informes *ad hoc* que piden los analistas se generan mediante programas desarrollados a medida.

La necesidad de informes a medida por parte de los analistas no es un hecho aislado en las organizaciones, puesto que las posibilidades ofrecidas por los sistemas operacionales en este sentido suelen ser muy limitadas. Esto se produce porque el objetivo principal de estos sistemas es el apoyo a la operativa

de la empresa: sus datos están estructurados y organizados con esta misión. Aun así, los analistas requieren datos estructurados para el apoyo al proceso de toma de decisiones que tienen que llevar a cabo, y estos datos se basan en otros registrados por los sistemas operacionales. Por lo tanto, el desarrollo de programas específicos para la generación de informes a medida es bastante frecuente en las organizaciones.

Con el paso del tiempo, la situación a la que se llega en muchas organizaciones es similar a la mostrada en la figura 3. Es decir, partiendo de las fuentes de datos de los sistemas operacionales se construye una estructura parecida a una telaraña compuesta por programas de extracción, transformación e integración de datos, y también de bases de datos intermedias que son utilizadas por otros programas para producir los informes requeridos. Inmon denomina la estructura resultante como el entorno operacional de telaraña.

Figura 3. Telaraña de un entorno operacional



1.2. Transformación del entorno operacional para satisfacer las necesidades de información

El entorno operacional de telaraña, resultado del desarrollo de informes a medida solicitados por parte de los analistas, presenta graves inconvenientes. El primero de ellos, que se ha intentado reflejar visualmente en la figura 3, es un problema de complejidad: el resultado del desarrollo de informes a petición constituye un entorno complejo, no planificado, ni estructurado.

En el desarrollo de cualquier aplicación informática, cumplir los plazos de entrega suele ser un objetivo muy importante, que a veces prevalece sobre otros, como, por ejemplo, el de generar la documentación necesaria para su mantenimiento posterior. En el caso de las aplicaciones desarrolladas para generar

un informe, el plazo de entrega es especialmente crítico, puesto que ha sido solicitado por un analista que lo necesita como soporte para tomar una decisión, por lo cual es frecuente que no se lleven a cabo las actividades de documentación que serían deseables. El objetivo de cada equipo de desarrollo es generar el informe requerido y para esto desarrolla los programas necesarios, basándose en la medida de lo posible en los programas o bases de datos intermedios previamente desarrollados. Aun así, esto último no siempre es posible porque no hay bastante documentación ni control sobre estos o simplemente porque es más rápido y menos costoso en tiempo de desarrollo hacer algo nuevo que modificar programas existentes y que, por lo general, no están suficientemente documentados.

En relación con esto último, tenemos un problema de falta de productividad: para cada informe solicitado se tienen que localizar los datos necesarios (esta operación no es nada trivial, por la falta de integración de los datos en las fuentes de datos y en las bases de datos intermedias) y desarrollar los programas de extracción, integración y transformación de los mismos. Dado que para hacer estas operaciones es muy difícil reutilizar los esfuerzos dedicados previamente al desarrollo de otros informes, nos encontramos en una situación en la que, para desarrollar cada informe, prácticamente tenemos que partir de cero.

Uno de los problemas más graves de este entorno es el de falta de credibilidad (reflejado en la figura 3): debido a la complejidad del entorno, no es improbable que un mismo informe (por ejemplo, los resultados del departamento en el último mes) se haya obtenido de dos maneras distintas, posiblemente como parte de otro informe que contiene información adicional. Aunque se haya partido de los mismos datos, se pueden haber recorrido caminos diferentes para obtener cada uno de los informes. El problema surge cuando la información presentada en los dos informes no coincide y genera desconcierto en los usuarios, lo que provoca falta de confianza y credibilidad respecto al Departamento de Informática como responsable de los informes generados.

No resulta extraño que se produzca esta situación, puesto que los datos de las fuentes suelen ser no integrados. Basta con que el equipo de desarrollo de uno de los informes interprete algún dato de las fuentes de datos o de las bases de datos intermedios de manera diferente para que se produzcan resultados como los que se han descrito antes. La pérdida de la confianza de los analistas en los datos con los que trabajan y en el departamento que los ha generado es un problema muy grave, dado que es muy difícil de recuperar posteriormente.

El **entorno operacional de telaraña** presenta problemas de complejidad, falta de productividad y falta de credibilidad.

Aparte de los problemas descritos, la pregunta que nos tenemos que hacer es la siguiente: ¿puede satisfacer las necesidades de información de los analistas el entorno operacional de telaraña?

Respecto al tiempo requerido para obtener la información, el plazo que transcurre desde el momento en que el analista hace la petición del informe hasta que lo recibe seguramente supera lo que sería deseable en un entorno tan cambiante como en el que se mueven las organizaciones actualmente.

Por otro lado, en lo que se refiere al contenido de los informes, los analistas suelen requerir la evolución de diferentes datos, y, para ello, precisan realizar un análisis de la información histórica de la organización. Aun así, la información de los sistemas operacionales generalmente es información no histórica, o contiene una historia muy limitada, puesto que se requiere para las operaciones diarias de la organización en las que generalmente se utilizan datos de fechas recientes (no histórica).

El entorno operacional de telaraña no satisface los requerimientos de información de los analistas.

Si el entorno operacional presenta graves problemas y, además, no satisface las necesidades de información por parte de los analistas, ¿lo podemos transformar para corregir estas deficiencias?

Respecto a la historia de la información almacenada en los sistemas operacionales, una solución podría ser almacenar la información histórica requerida por los analistas. Esto representaría un aumento en la complejidad de estos sistemas. Por otro lado, si las aplicaciones disponibles no almacenan la historia de los datos, deberían modificarse para que lo hicieran. Esta operación puede resultar muy costosa, sobre todo en aplicaciones antiguas que hayan sufrido cambios a lo largo de su historia, puesto que, generalmente, los cambios suelen estar poco documentados y las modificaciones requeridas no son sencillas.

Para generar informes fiables más rápidamente y con menos coste, se tendría que reducir la complejidad del entorno operacional de telaraña. Para esto, principalmente necesitaríamos disponer de un entorno que tuviera los datos integrados. El problema es que los datos del entorno operacional, sobre todo si este se ha desarrollado de manera gradual a lo largo de la historia o si se han adquirido aplicaciones empaquetadas, suelen ser no integrados. En este caso, también necesitaríamos modificar todas las aplicaciones para integrar los datos en las mismas.

Por lo tanto, la situación es la siguiente: disponemos de un conjunto de aplicaciones, que forman el entorno operacional y que satisfacen las necesidades operacionales de la organización, pero que no satisfacen las necesidades de in-

formación de los analistas. Para intentar corregir esta situación, necesitaríamos modificar todas las aplicaciones que hay para integrar los datos y almacenar su historia. Esta operación puede resultar inviable por su complejidad y coste.

Por otro lado, si nos fijamos en las necesidades de los diferentes tipos de usuarios, los analistas necesitan hacer consultas muy complejas (por ejemplo, evolución de los resultados del departamento en los últimos años) y obtener los resultados de manera inmediata. Aun así, los usuarios de los sistemas operacionales necesitan conocer de la manera más fiel posible la situación actual del sistema modelado (por ejemplo, situación actual del departamento). Por lo tanto, los sistemas operacionales se tienen que estructurar para reflejar todos los cambios que se produzcan y estos cambios pueden ser muy frecuentes (por ejemplo, interesa que el saldo de una cuenta esté actualizado en el momento en que se produzca cualquier cambio).

Las necesidades de los analistas y de los usuarios de los sistemas operacionales son contrapuestas: difícilmente un sistema se puede diseñar y configurar para ser óptimo tanto en la respuesta a consultas complejas requerida por los analistas como en la ejecución de modificaciones sobre los datos en los que estas se basan.

- Un mismo sistema no puede satisfacer al mismo tiempo las necesidades operacionales de la organización y las de información de los analistas.

- Aunque fuera posible hacerlo, los sistemas operacionales son muy costosos de modificar para integrar todos los datos y almacenar su historia.

- Por lo tanto, los analistas de las organizaciones requieren sistemas específicos como apoyo en el proceso de toma de decisiones.

Los nuevos sistemas requeridos por los analistas necesitan obtener los datos a partir de los sistemas operacionales existentes y evitar modificaciones. La solución consiste en construir sistemas con esta finalidad específica, que estén diseñados para optimizar el tipo de operaciones que necesitan hacer los analistas y que funcionen en plataformas especialmente configuradas para ello.

Por otro lado, sería conveniente que los analistas dispusieran de sistemas que les permitieran obtener directamente la información requerida, en lugar de lograrla mediante peticiones al Departamento de Informática. Esto se puede conseguir diseñando los nuevos sistemas mediante un modelo de datos que esté especialmente orientado a esta finalidad, que es el modelo de datos multidimensional y que les dotará de mayor autonomía.

Por último y a efectos de concurrencia de usuarios puede ser muy diferente el volumen de usuarios y los momentos de acceso en un sistema operacional que en un sistema orientado al trabajo de los analistas.

La solución para ofrecer soporte a las necesidades de información de los analistas consiste en incorporar el concepto de **almacén de datos** a la organización mediante la implementación de la FIC.

1.3. Diferencias entre un entorno operacional y uno informacional

Como se ha mencionado anteriormente existen notables diferencias entre un entorno operacional y un entorno que cubre las necesidades de información de los analistas. Este segundo entorno, que se basa en la FIC, es un entorno informacional. Un resumen de las diferencias es el siguiente:

1) Necesidades de información: los analistas de la FIC se orientan a la consulta y análisis de datos, mientras que los usuarios del entorno operacional trabajan más orientados a la operativa y día a día del negocio.

2) Ubicación de los datos: los entornos operacionales son heterogéneos y su información es de diferente naturaleza y distribuida dentro de la organización, mientras que en la FIC la orientación va dirigida a centralizar información de diferente naturaleza, pero complementaria.

3) Vigencia de los datos: los entornos operacionales guardan una ventana temporal reducida orientada a dar servicio a las consultas y transacciones más recientes en el tiempo, mientras que la FIC deberá asumir una ventana temporal mayor que soporte análisis sobre mayo rango temporal.

4) Tipo de operaciones que hay que realizar: actualización atómica (pocos registros) de tipo transacción en el entorno operacional (ejemplo, alta de un cliente) y consulta y actualizaciones masivas en la FIC (ejemplo, obtener la evolución de la cartera de clientes en los últimos seis meses).

2. Estrategias en la construcción de la FIC

2.1. Diferentes enfoques en la construcción de la FIC

La experiencia de los proyectos *Data Warehouse* demuestra que, aunque el concepto de FIC sea un concepto con una arquitectura bien definida, su implementación ha dado lugar a diferentes enfoques o variantes, a veces por simplificar, a veces por falta de metodología y a veces con el objetivo de conseguir resultados tangibles de manera más rápida.

La complejidad que puede llegar a tener la arquitectura de la FIC según el negocio y el tamaño de una organización, hace conveniente conocer diferentes enfoques de abordaje de este tipo de proyectos. A continuación, vamos a conocer y analizar dichos enfoques en la construcción de la FIC.

2.1.1. Enfoque basado en la construcción de almacenes de datos departamentales

La mayoría de los analistas trabaja directamente sobre los almacenes de datos departamentales. Es decir, para los analistas de información, estos son la «fachada» de la FIC, lo que los usuarios ven. El resto de los componentes de la FIC proporciona los datos a los almacenes de datos departamentales y frecuentemente son invisibles para sus usuarios. Por lo tanto, es habitual que la percepción que los usuarios tienen de la FIC sea, de manera exclusiva, la ofrecida por los almacenes de datos departamentales.

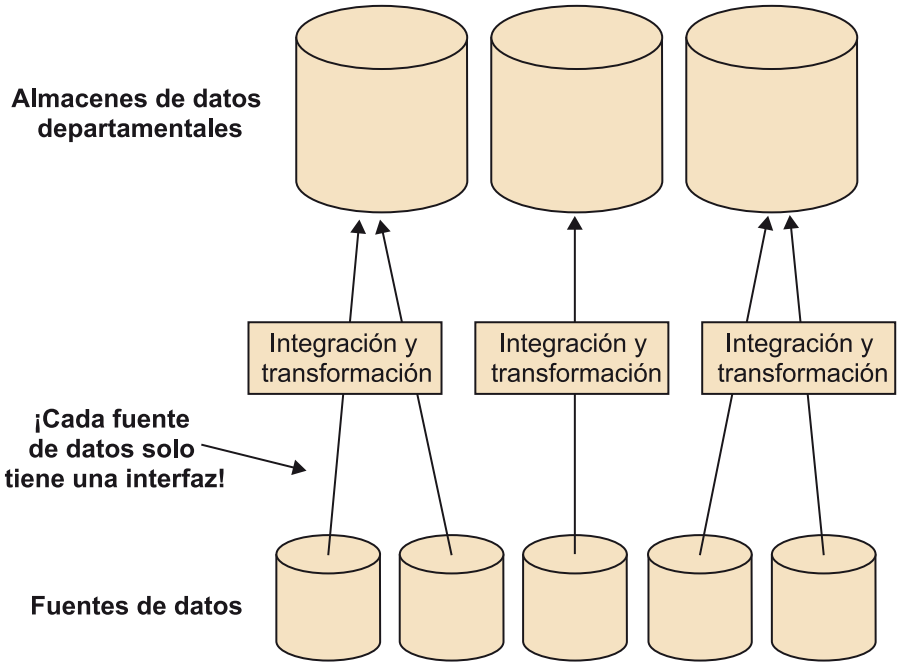
Esta percepción, a veces, es compartida por los desarrolladores, en algunos casos, por desconocimiento. En muchos libros y otros materiales de formación, así como en presentaciones de vendedores de diferentes herramientas, el único componente que se estudia cuando se habla de almacenes de datos es el que corresponde a los departamentales, y no se tiene en cuenta el resto de los componentes de la FIC.

En otros casos, aunque se conozca la arquitectura de la FIC, se ignora de manera consciente para perseguir resultados inmediatos y, aparentemente, construir solo almacenes de datos departamentales resulta más barato, fácil y rápido que construir la FIC.

Esto último es cierto cuando solo necesitamos un almacén de datos departamental o bien un reducido número de estos. Generalmente, esta es la manera de incorporar el concepto de almacén de datos en las organizaciones. Para empezar, se plantea la construcción de un almacén de datos departamental porque así podemos obtener resultados de manera inmediata. Partiendo de los

datos de las fuentes de datos, los transformamos e integramos en un almacén de datos departamental diseñado según algún modelo multidimensional, que se caracteriza por ser un modelo totalmente orientado a la consulta. Es decir, sin duda, la manera más rápida y barata de construir un primer almacén de datos departamental consiste en centrarse de manera específica en su construcción, en lugar de construir previamente la FIC. Además del almacén de datos departamental, también habremos construido un componente de integración y transformación específico para obtener los datos que almacenamos.

Figura 4. Almacenes departamentales con componente de integración y transformación específico



Una vez tenemos un almacén de datos departamental, no es extraño que usuarios de este u otros departamentos requieran la construcción de nuevos almacenes de datos para satisfacer sus necesidades específicas de información. Dado que los diferentes departamentos generalmente trabajan con distintos datos, aunque puedan compartir algunos de ellos, es frecuente que los nuevos proyectos se desarrollen de forma independiente respecto a los anteriores. Así pues, para cada proyecto: se diseña el almacén de datos departamental y se construye un componente de integración y transformación, según los requerimientos del mismo.

La construcción de almacenes de datos de manera independiente es el enfoque «natural» o intuitivo que se lleva a cabo en muchas organizaciones, por desconocimiento de la FIC o, conociéndola, para tratar de ahorrar costes al comienzo. Es un enfoque válido cuando se trata de construir almacenes de datos departamentales totalmente independientes.

El problema surge cuando los almacenes de datos por construir no son totalmente independientes; es decir, cuando hay datos comunes que deben incluirse en diferentes almacenes de datos departamentales. En estas situaciones, nos encontramos lo siguiente:

1) Los diferentes componentes de integración y transformación trabajan sobre las fuentes de datos comunes: generamos múltiples interfaces para los mismos datos. Ello representa un problema de coste en tiempo de desarrollo, mantenimiento y ejecución. Esto último se produce porque se accede diferentes veces a los mismos datos, una vez para cada almacén de datos departamental que los utiliza. Hay fuentes de datos origen compartidas entre almacenes.

2) Cada almacén de datos ha podido hacer su propia interpretación de los mismos datos. Así pues, tenemos una falta de integración de datos comunes en los diferentes almacenes de datos departamentales.

Ejemplo de falta de integración de los datos

Un dato denominado beneficio puede interpretarse de manera distinta y tener significado diferente en cada uno de los almacenes de datos departamentales en los que se defina. En uno se puede tratar del beneficio antes de impuestos y en otro, después de impuestos. Si comparamos el beneficio conseguido en cada uno de los departamentos de los respectivos almacenes de datos, podemos obtener resultados no reales o erróneos.

El problema de la falta de integración se acentúa cuando hay datos replicados entre las fuentes de datos, situación que se produce con frecuencia, y cada almacén de datos departamental hace su integración. En este caso, además de las diferentes posibilidades de interpretación de los datos de las fuentes, se tienen que añadir las diferencias que se puedan producir cuando se integran los datos de manera distinta en cada almacén de datos departamental.

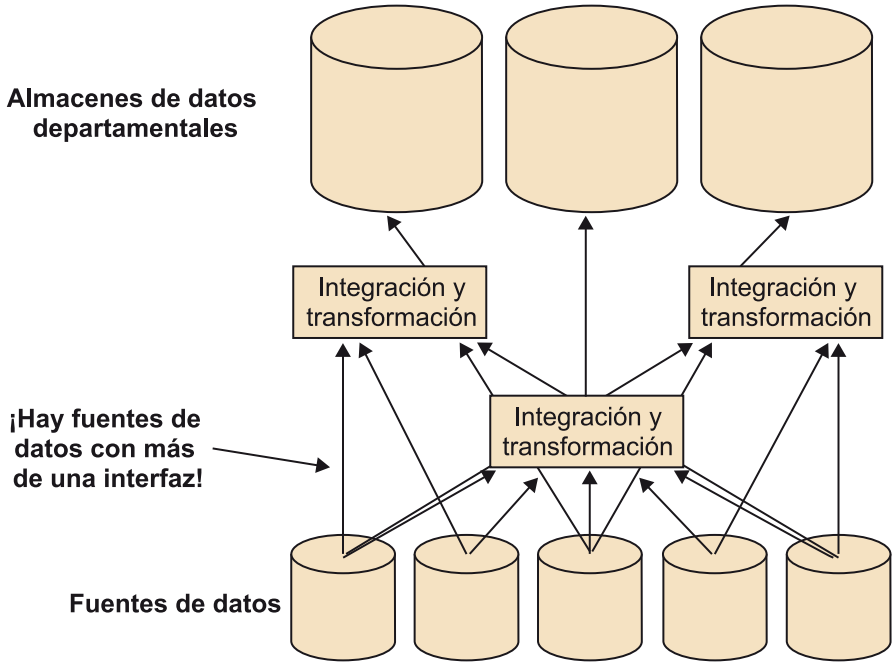
Ejemplo de diferentes interpretaciones en la integración de los datos de las fuentes

En una organización disponemos de una aplicación de marketing con datos de clientes potenciales (clientes del pasado, clientes actuales y posibles clientes) y otra de facturación con los datos de los clientes actuales. En un almacén de datos departamental que necesita trabajar con los datos de los clientes, se ha partido de los datos de clientes en la aplicación de marketing y se han completado con los datos disponibles en la aplicación de facturación. Aun así, en otro almacén de datos departamental se ha procedido de manera inversa: se han tomado como base los datos de los clientes en la aplicación de facturación y se han completado con los datos que hay en la aplicación de marketing. El resultado global en los dos casos no tiene que coincidir; de hecho, es más probable que las bases de datos de clientes resultantes en los dos almacenes de datos departamentales sean muy distintas.

El problema de fuentes de datos comunes entre almacenes departamentales es más habitual de lo que podamos estimar, ya que hay muchas fuentes que serán críticas para analizar nuestro negocio y aparecerán en los distintos almacenes departamentales. Un ejemplo pueden ser los clientes, los productos, los proveedores, las cuentas corrientes, etc.

Por las razones mencionadas anteriormente, en muchas organizaciones el problema de construcción de la FIC se reduce al de construcción de manera independiente de un conjunto de almacenes de datos departamentales, a pesar de que haya dependencias entre estos. En estos casos, la arquitectura que se obtiene como resultado es la que se muestra en la figura 5.

Figura 5. Fuentes de datos comunes entre almacenes departamentales



Esta arquitectura puede resultar válida cuando se empieza a implantar el concepto de almacén de datos en la organización, cuando solo se dispone de unos pocos almacenes de datos departamentales y estos son realmente independientes entre sí. Sin embargo, lo habitual es que los almacenes departamentales tengan dependencias entre sí y que crezca el número de ellos con el tiempo. ¿Qué podemos hacer en esta situación?

Los problemas de integración y de multiplicidad de interfaces sobre las fuentes de datos se solucionan mediante la construcción del almacén de datos corporativo.

2.1.2. Construcción del almacén de datos corporativo *a posteriori*

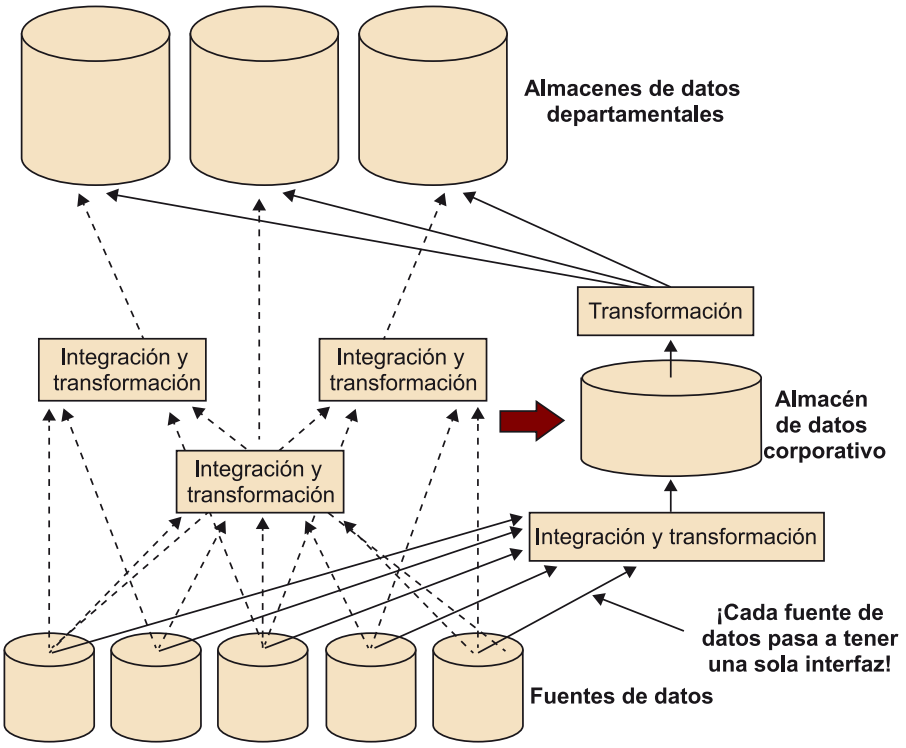
El objetivo al construir un almacén de datos corporativo es tener un repositorio centralizado e integrado de la información de la compañía. En un almacén de datos departamental nos centramos en su consulta, pero en un almacén de datos corporativo es un punto crítico el tener un almacenamiento centralizado e integrado. Para la construcción del almacén de datos corporativo, orientado al almacenamiento de los datos más que a su consulta, se hace lo siguiente:

1) Se reduce el número de interfaces en las fuentes de datos: solamente es necesaria una interfaz para cada fuente para llevar los datos de la fuente de origen al almacén de datos corporativo.

2) Los datos están integrados en el almacén de datos corporativo, lo que evita tener múltiples interpretaciones de los mismos.

Si partimos de cero, la solución consiste en construir el almacén de datos corporativo antes o durante la construcción de los diferentes almacenes de datos departamentales. Aun así, si ya tenemos diferentes almacenes de datos departamentales construidos, tenemos que transformar la arquitectura basada exclusivamente en estos para incluir el almacén de datos corporativo.

Figura 6. Almacén de datos corporativo y almacenes de datos departamentales



El problema para incluir el almacén de datos corporativo cuando ya tenemos construidos diferentes almacenes de datos departamentales radica en que tenemos que realizar un proceso de reingeniería sobre los diferentes componentes de integración y transformación, así como sobre los almacenes de datos departamentales previamente, para adaptarlos a la nueva arquitectura. Esta transformación puede resultar muy compleja y costosa.

Particularmente, en lo que respecta a la transformación de los almacenes de datos departamentales, el principal problema que se presenta es el de la integración de los datos: si se han construido de una manera no integrada, se les tiene que transformar, y, en algunos casos, esta transformación en el ámbito técnico presenta problemas de gestión y no es nada trivial. Además, tenemos

el problema añadido de que los cambios hechos sobre los almacenes de datos departamentales son visibles para los usuarios finales y afectan directamente a su trabajo.

Ejemplo de corrección por falta de integración de datos en los almacenes de datos departamentales

Si el dato «beneficio» del ejemplo utilizado anteriormente en uno o diferentes almacenes de datos departamentales se interpreta como «beneficio antes de impuestos» y la interpretación aceptada en el ámbito corporativo es del «beneficio después de impuestos», la nueva interpretación tiene que incorporarse en los almacenes de datos departamentales que no la considerasen. Por lo tanto, los usuarios deben adaptarse al nuevo significado a la hora de generar los informes que necesiten.

Debido a los problemas que surgen al construir el almacén de datos corporativo después de haber construido los almacenes de datos departamentales, se recomienda construirlo previamente o al mismo tiempo. Es decir, es más adecuado planificar la construcción de la FIC desde un principio.

Un planteamiento intermedio entre la construcción de almacenes de datos departamentales independientes y su construcción de manera conjunta con la FIC es la creación de almacenes departamentales con dimensiones conformadas.

Una **dimensión conformada** es una entidad del modelo de datos compartido por varios almacenes de datos. Por ejemplo, la entidad Cliente o la entidad Producto pueden ser compartidas por el almacén departamental de marketing, el de operaciones y el de finanzas, ya que todos ellos tienen que realizar sus análisis en base a los mismos productos y clientes.

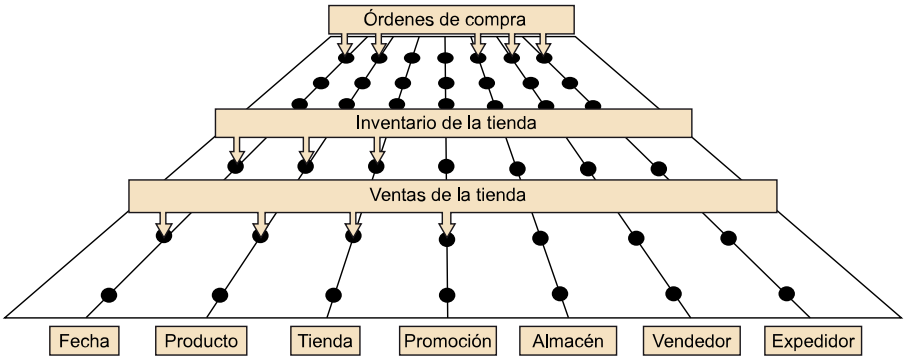
Es importante tener una única entidad para todos los almacenes departamentales que asegure una única versión de la misma y un único proceso de transformación y carga que la actualice. Este planteamiento fue propuesto por Ralph Kimball dentro de la arquitectura conocida como *Enterprise bus matrix* en lugar de basarla en la FIC como plantea Inmon.

La identificación de dimensiones conformadas implica una visión global de la organización, aun en el caso de estar creando un almacén de datos departamental. Es necesario analizar qué procesos de negocio de nuestra organización pueden ser analizados utilizando las entidades que creamos en nuestro almacén departamental. Por ejemplo, en el almacén departamental financiero vamos a utilizar la entidad Producto, pero esta entidad se aplica también al análisis de otros procesos de negocio, como puede ser el proceso de ventas *online* o los procesos de fabricación, propios de otros departamentos. Así mismo,

al crear un nuevo almacén departamental cuando ya existen otros, conviene revisar la existencia de dimensiones conformadas en otros almacenes departamentales, con objeto de reutilizarlas en el nuevo almacén.

En la figura 7 se representa esta arquitectura (*entreprise datawarehouse bus matrix*) propuesta por Kimball, que difiere notablemente de la FIC propuesta por Inmon. La propuesta de Kimball se apoya en la creación de almacenes de datos departamentales que posteriormente quedan conectados por las dimensiones comunes, mientras que la propuesta de Inmon plantea la creación conjunta de los almacenes departamentales con la FIC. En esta asignatura nos vamos a centrar en la construcción basada en la FIC de Inmon, aunque en diferentes momentos se hará referencia al planteamiento de Kimball.

Figura 7. *Enterprise bus matrix*



Fuente: www.kimballgroup.com.

2.1.3. Combinación del almacén de datos operacional y el almacén de datos corporativo

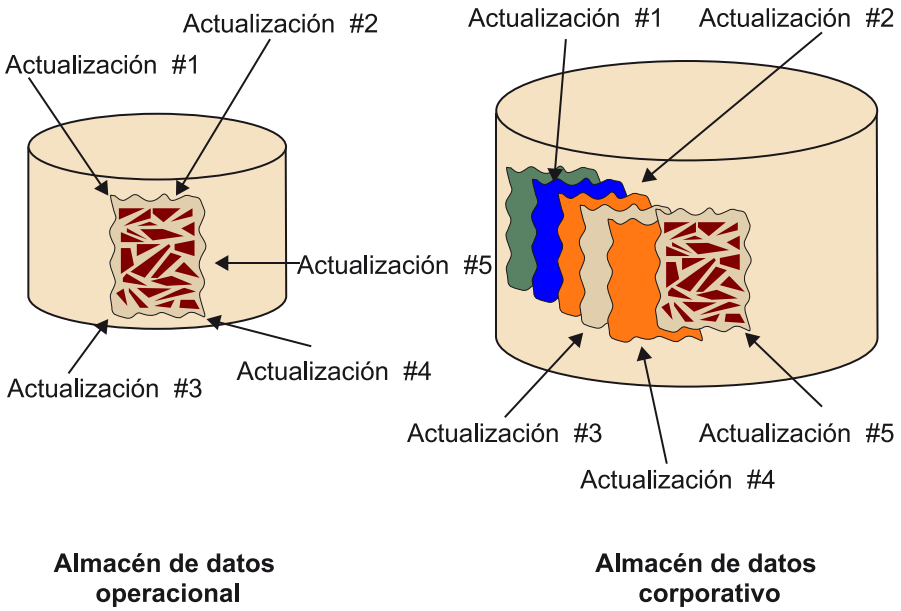
El almacén de datos operacional y el almacén de datos corporativo tienen características similares: los dos están orientados al tema e integrados. Se diferencian en el hecho de que los datos del almacén de datos operacional son volátiles y no históricos, mientras que los datos del almacén de datos corporativo son no volátiles e históricos.

El almacén de datos operacional almacena una imagen actualizada de los datos integrados de la organización. El almacén de datos corporativo almacena una película formada a partir de las diferentes imágenes de los datos de la organización, es decir, la información está historiada.

Ejemplo de almacén de datos operacional y corporativo en un operador de telecomunicaciones

Por ejemplo, en un almacén operacional de un operador de telecomunicaciones interesa saber la tarifa actual de un cliente, pero en un almacén de datos corporativo, interesa saber las distintas tarifas que ha tenido un cliente desde su alta en la compañía y el periodo de tiempo durante el cual ha tenido cada tarifa. En el almacén de datos corporativo se analizará la historia del cliente en la compañía.

Figura 8. Almacén de datos operacional y corporativo



¿Podemos combinar la construcción de los dos almacenes de datos en una única estructura? En teoría sí, pero en la práctica no es conveniente. La razón principal para no combinarlos es que tienen que estar diseñados para soportar diferentes tipos de operaciones, que realizan distintos tipos de usuarios.

El objetivo principal del almacén de datos operacional es mantener una visión integrada y totalmente actualizada de los datos operacionales. Sobre estos, se ejecutarán muchas operaciones de actualización y consultas simples que serán llevadas a cabo principalmente por oficinistas. Su diseño y configuración estarán orientados para realizar este tipo de operaciones de una manera óptima.

Por otro lado, el almacén de datos corporativo no requiere que sus datos estén totalmente actualizados: basta con que estén actualizados según las necesidades de los analistas, que son sus usuarios (en algunos casos, bastará con que se actualicen de manera periódica: semanal o mensualmente). Este almacén está especialmente diseñado para que las actualizaciones se almacenen como imágenes nuevas, de manera que se va a ir formando una película de los datos. Está pues configurado para optimizar las consultas de las imágenes de los datos, no para hacer modificaciones sobre estos.

En ocasiones, nos podemos encontrar que los almacenes de datos operacionales tienen un nivel de granularidad más detallado que el almacén de datos corporativo, ya que este utiliza información agregada.

Si combinamos las dos estructuras, el resultado será una base de datos con muchos registros (tenemos que guardar la historia almacenada en el almacén de datos corporativo), que debe estar configurada para hacer modificaciones sobre los datos (algo requerido por el almacén de datos operacional). De este modo, cualquier transacción será muy costosa, puesto que se pueden combi-

nar consultas complejas con actualizaciones, y el volumen de datos que deben tratar las diferentes transacciones será muy grande porque tendrá que considerar los datos del almacén de datos corporativo.

El almacén de datos operacional y el almacén de datos corporativo tienen objetivos diferentes y están diseñados para conseguirlos de manera óptima. Si los combinamos en una estructura común, se degradará el tiempo de respuesta para las dos funcionalidades.

2.1.4. La FIC sin el almacén de datos operacional

El almacén de datos corporativo contiene todos los datos que necesitan los analistas, que se obtienen directamente a partir de las fuentes de datos operacionales y también a partir del almacén de datos operacional, que a su vez los obtiene de las fuentes de datos operacionales tal como puede verse en la figura 9. El proceso de integración y transformación de los datos de las fuentes operacionales para incluirlos en el almacén de datos corporativo o en el almacén de datos operacional es común a los dos, y por este motivo es suficiente con transformar los datos del almacén de datos operacional para adaptarlos a las estructuras del almacén de datos corporativo.

Figura 9. FIC con almacén de datos operacional

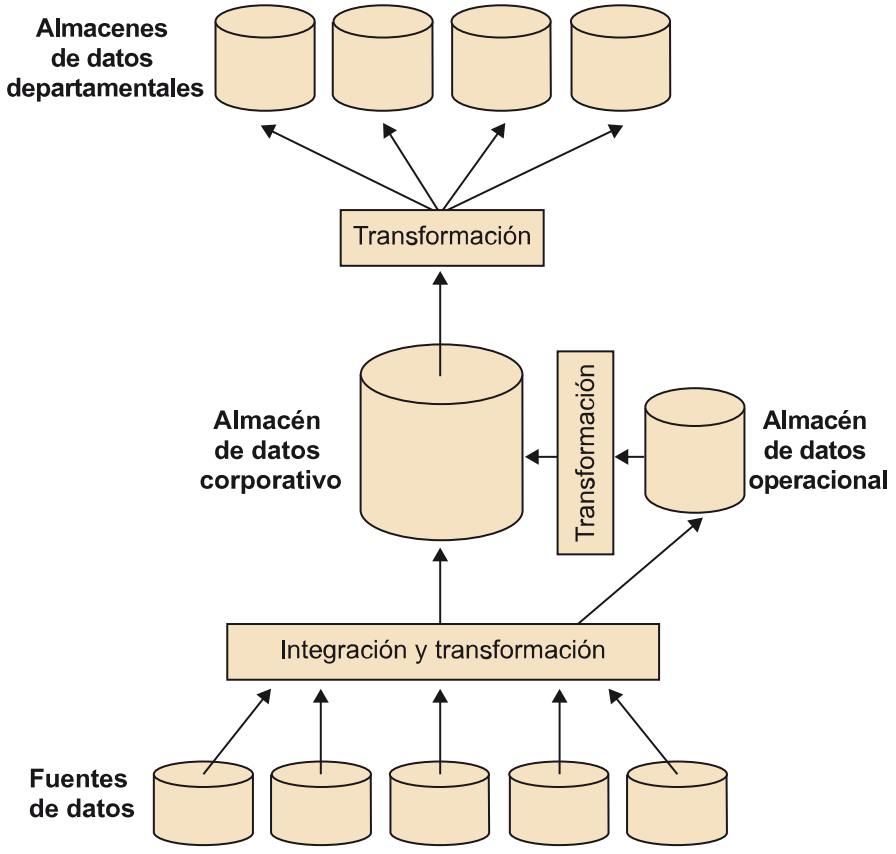
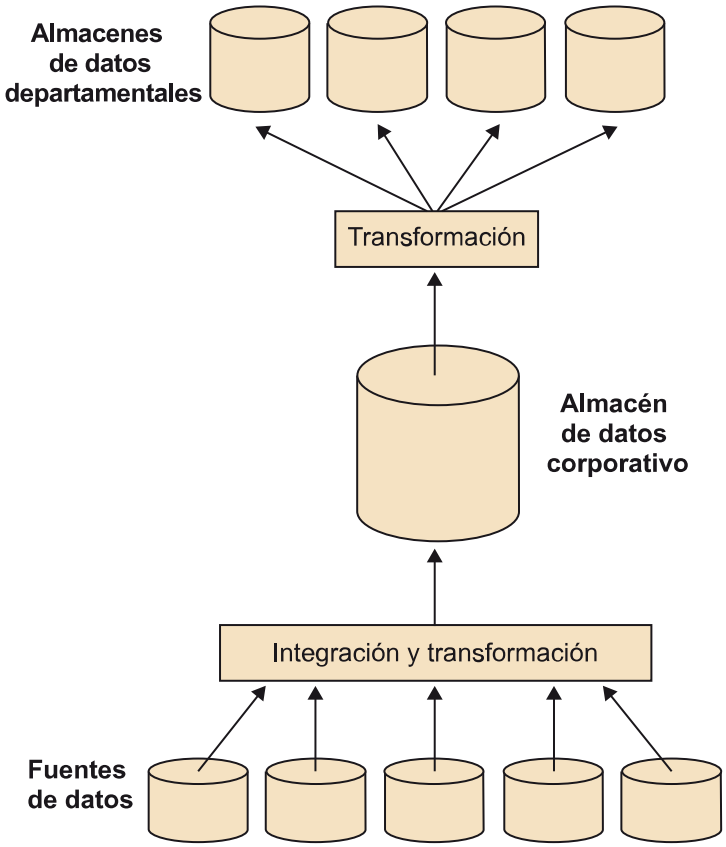


Figura 10. FIC sin almacén de datos operacional



A diferencia del almacén de datos corporativo, el almacén de datos operacional no es estrictamente necesario en la FIC. Es decir, aunque conviene disponer del mismo por las funcionalidades que proporciona a los usuarios, y también para permitir construir más fácilmente el almacén de datos corporativo, no se necesita de manera estricta para construir el almacén de datos corporativo ni tampoco para construir los almacenes de datos departamentales, que son los que proporcionan la funcionalidad principal a los analistas (ver la figura 10).

Las dos arquitecturas representadas en las figuras 9 y 10 son igualmente válidas, aunque la arquitectura de la figura 10 ofrece menos funcionalidades que la de la figura 9.

El **almacén de datos operacional** es una estructura opcional en la FIC, para proporcionar información a los analistas.

A pesar de que el almacén de datos operacional es opcional en la FIC, en algunas organizaciones será más necesario que en otras, dependiendo de distintos factores, entre los cuales destacan los siguientes:

- El tamaño de la organización: cuanto más grande sea la organización, más probable es que necesite el almacén de datos operacional. En las organi-

zaciones pequeñas, los problemas debidos a la falta de integración de los datos operacionales suelen ser menores.

- La naturaleza de los negocios: si la organización necesita acceder de manera inmediata a información integrada y actualizada, por ejemplo, porque interacciona directamente con los clientes o porque la necesita en el proceso de fabricación, es muy probable que necesite el almacén de datos operacional.

- También puede afectar el hecho de que en la compañía tenga una orientación analítica importante y en los distintos niveles de la organización (también el más operativo) se aplique el análisis y se necesite dar soporte a decisiones operativas.

- Finalmente, depende de la cantidad de las aplicaciones operacionales y del grado de integración que hay entre estas. En una organización con un conjunto pequeño de aplicaciones operacionales que están muy integradas, el almacén de datos operacional será menos necesario que en otra con gran cantidad de aplicaciones poco integradas.

2.1.5. La FIC con *Staging Area*

Una *Staging Area* o área de maniobras es una zona de trabajo temporal de la FIC, ubicada entre las fuentes de datos de los sistemas operacionales y el *Data Warehouse*.

A pesar de ser una pieza opcional de la FIC, hay escenarios en los que es realmente útil plantearse el uso de este espacio temporal. Es habitual implementar la *Staging Area* con una base de datos, aunque también podría utilizarse un conjunto de archivos temporales como zona de trabajo.

El uso de una *Staging Area* simplifica el proceso de extracción del componente de integración de la FIC, especialmente cuando es necesario realizar cálculos intermedios para reducir la complejidad del dato, homogenizarlo desde múltiples orígenes heterogéneos o para realizar procesos de limpieza de los datos que mejoren su calidad.

Otra ventaja de su uso es la reducción del impacto sobre los sistemas operacionales que puede provocar, cuando se ejecutan tareas ETL (*Extract Transform Load*) pesadas. Por ejemplo, cuando, desde los sistemas de *Business Intelligence*, nos conectamos a los sistemas operacionales para cargar o actualizar la información del *Data Warehouse*, siendo perceptible incluso por los usuarios finales.

Este problema puede verse agravado cuando no sea posible realizar las cargas en horarios de baja actividad, o cuando, para realizar la carga del *Staging Area*, sea necesario realizar cálculos previos sobre los datos del sistema operacional.

En este escenario, la *Staging Area* nos permite hacer una carga «en bruto» lo más rápida posible de los datos del sistema operacional hacia el área de maniobras. Esto reduce drásticamente el tiempo de conexión al sistema operacional y, por lo tanto, el impacto negativo en su rendimiento. Posteriormente, todos los cálculos necesarios antes de enviar el dato al *Data Warehouse* y todos los procesos de limpieza del dato se realizarán sobre la copia de la *Staging Area*, totalmente desconectada de la fuente de datos original.

Finalmente, otra ventaja importante de la *Staging Area* es la tolerancia a errores que aporta al componente de integración. Si, por algún motivo, se genera un error en los procesos de transformación o carga, podríamos reiniciarlos, sin necesidad de repetir el proceso de extracción y, por lo tanto, sin afectar nuevamente a los sistemas operacionales.

Un aspecto negativo del uso de la *Staging Area* es el incremento del tiempo total de ejecución de los ETL, pues, inevitablemente, estaremos añadiendo más procesos intermedios. No obstante, los beneficios globales que aporta casi siempre compensan este coste en tiempo de ejecución.

2.2. Construcción de la FIC mediante un solo proyecto

Si una organización reconoce la necesidad de la FIC, generalmente considera que necesita todos sus componentes y, además, que los necesita de manera inmediata, la tendencia habitual es plantear la construcción de la FIC mediante un solo proyecto. El problema principal que plantea este proyecto es la complejidad.

Generalmente, en este entorno, los analistas todavía no conocen claramente las funcionalidades que les pueden ofrecer los almacenes de datos departamentales. La idea inicial que tienen sobre sus necesidades de información cambia cuando descubren las posibilidades que les ofrecen los almacenes de datos. En estas condiciones, difícilmente pueden transmitir sus requerimientos a los desarrolladores de la FIC. El alcance funcional de la FIC es complicado de delimitar. Por otro lado, localizar los datos que se necesitan en las aplicaciones operacionales no es una tarea fácil. Si además se pretende desarrollar todo el conjunto de la FIC en un solo proyecto, este será de dimensiones y complejidad demasiado grandes en comparación con los que se suelen desarrollar en las organizaciones.

Por otro lado, precisamente debido a la complejidad del proyecto, se hace muy difícil justificar el coste de desarrollo de la FIC según el beneficio que aporta a la organización, puesto que los costes son muy grandes y el beneficio, aunque está claro, no se ha evaluado lo suficiente.

El riesgo de proyecto en términos de plazos y costes es alto. En un proyecto de tal envergadura, tendremos un alcance de proyecto muy difícil de determinar con exactitud. Por otra parte, no es sencillo gestionar las expectativas y la toma de requisitos de un alto número de analistas de diferentes departamentos involucrados, donde las visiones y expectativas pueden ser muy dispares. Así mismo, existe también el riesgo del cambio de requerimientos con el tiempo, situación habitual en un proyecto de gran alcance al que los cambios en el negocio de la compañía pueden afectar directamente.

El resultado de plantear el desarrollo de la FIC como un solo proyecto es un proyecto muy grande y complejo, cuyos objetivos no están totalmente claros, los requerimientos pueden cambiar, y cuyos costes son difícilmente justificables en lo que respecta a la organización. En esta situación, es más probable que sea un fracaso.

2.3. Construcción de la FIC mediante proyectos autónomos

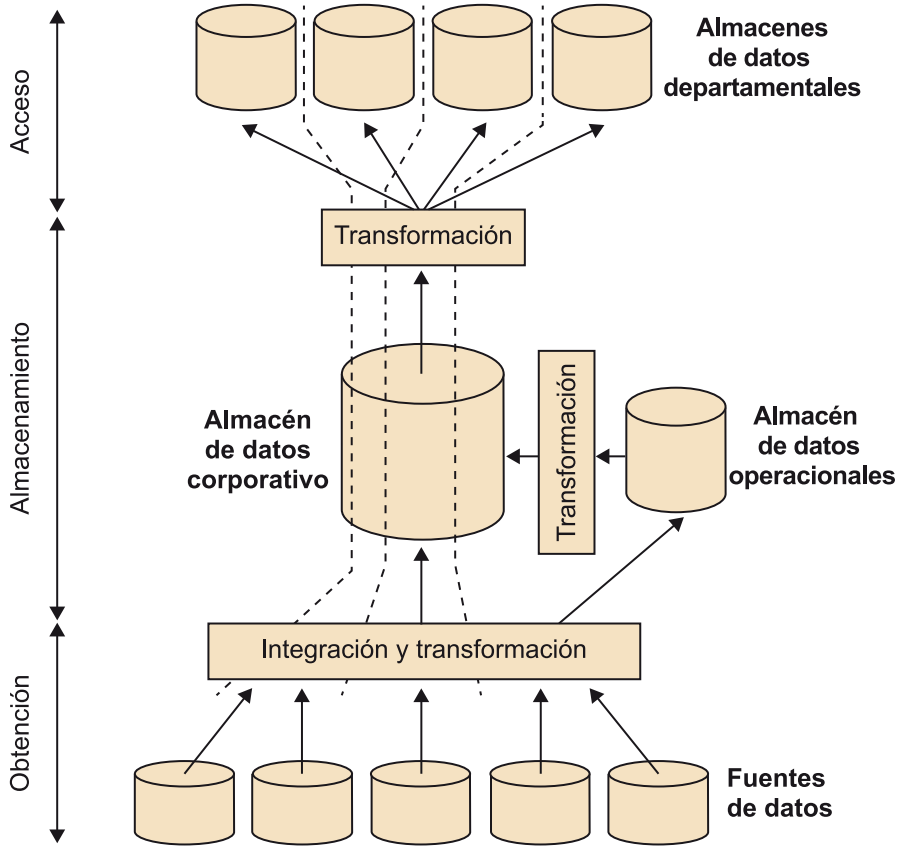
En apartados anteriores, hemos visto que intentar construir la FIC como un solo proyecto es demasiado complejo. Por otro lado, si dividimos la arquitectura de la FIC horizontalmente, según la funcionalidad de los componentes, no se reduce la complejidad tanto como sería deseable y, además, es difícil satisfacer los requerimientos de los analistas de información.

Por lo tanto, otra posibilidad consiste en dividir la arquitectura de la FIC de manera vertical, es decir, dividir la construcción de la FIC en forma de proyectos, de modo que hagan lo siguiente:

- Que proporcionen un valor por sí mismos a la organización para que su coste sea justificable.

- Que sean completos y autónomos en la medida en que se pueda; es decir, que no necesiten otros proyectos para entrar en producción.

Figura 11. División de la FIC en proyectos autónomos



En la figura 11, se muestra una representación de la estructura de los mencionados proyectos. Es decir, en cada proyecto se tienen que construir los componentes siguientes:

- El componente de acceso: el almacén de datos departamental con los datos que necesitan los analistas de este proyecto o bien la herramienta de acceso al almacén de datos corporativo o al almacén de datos operacional.

- El componente de almacenamiento: la parte del almacén de datos corporativo y/o almacén de datos operacional que contiene los datos a los que se necesita acceder, y que todavía no ha sido construida por ningún proyecto anterior.

- El componente de obtención de los datos: el componente de integración y transformación que consigue los datos necesarios para el proyecto, a partir de las fuentes de datos origen. Se considera que uno de los problemas más importantes que hay en la construcción de la FIC es el de la obtención de los datos, es decir, la localización de los datos necesarios y la construcción del componente de integración y transformación. Este es uno de los motivos más frecuentes de fracaso de los proyectos de construcción de la FIC.

Cada proyecto consta de los componentes de acceso, almacenamiento y obtención de los datos provenientes de las fuentes de datos (es una práctica habitual que cada componente sea desarrollado por personas o equipos distintos). Se trata de un proyecto completo que puede entrar en funcionamiento de manera independiente del resto de los proyectos existentes.

De este modo, cada proyecto intenta satisfacer los requerimientos de un grupo de analistas de forma independiente, de modo que sea justificable el coste del proyecto según los beneficios que aporta y se reduzca la complejidad del desarrollo de la FIC.

Bajo este planteamiento de división vertical, vamos a analizar en los siguientes apartados los proyectos en los que se dividiría la construcción de la FIC.

2.3.1. El primer proyecto: proyecto global de desarrollo

En un primer momento, la visión que se tiene de la FIC es dispersa. Conocemos su arquitectura y, además, su aportación como soporte de información para los analistas es positiva para la organización. Aun así, no conocemos todavía con detalle los almacenes de datos departamentales que se necesitan en la organización, ni lo que le aportarán.

El primer objetivo es dividir la construcción de la FIC en proyectos, de modo que cada proyecto:

- Corresponda a un problema concreto.
- Tenga a un responsable en la organización: el analista que utilizará el sistema como resultado del proyecto.
- Ofrezca un beneficio tangible a la organización: podemos conocer cuál será el coste del proyecto y el beneficio que este aportará a la organización.

Generalmente, cada proyecto definido se corresponderá con un almacén de datos departamental que se tiene que desarrollar. Su responsable será el analista (o uno de los analistas) que utilizará el almacén de datos departamental, y el beneficio se calculará según los objetivos que se pretendan conseguir con el almacén de datos desarrollado y su coste estimado de desarrollo.

Para definir los objetivos del proyecto, deberemos mantener reuniones con los analistas de información, que serán sus usuarios y responsables del proyecto. Aun así, no bastará con las reuniones con los usuarios. De manera adicional, para que la definición del proyecto sea realista, nos tendremos que asegurar de que los datos requeridos por los usuarios están en la organización, en las

fuentes de datos operacionales, y que tienen el grado de detalle y la calidad requeridos. Es decir, deberemos hacer una revisión global de todos los componentes de cada proyecto: obtener, almacenar y acceder a los datos requeridos por los usuarios.

Es decir, el objetivo del primer proyecto es transformar la visión dispersa que se tiene de la FIC en una visión concreta en forma de proyectos, de modo que cada uno de ellos satisfaga unos objetivos determinados y aporte un valor concreto a la organización. Esta visión que conforman todos los proyectos de la FIC debe ser lo más global posible, de forma que cada proyecto individual tenga en cuenta su impacto en el resto de los proyectos y viceversa, el impacto del resto de los proyectos en el propio proyecto. En el desarrollo de este tipo de proyectos suele ser útil añadir a la visión departamental la visión de procesos de negocio, dado que un proceso de negocio será transversal a varios departamentos. El conjunto de los almacenes departamentales por desarrollar en la FIC deberá dar una visión integrada y completa de los procesos de negocio.

Ejemplo de proceso de negocio: facturación

La facturación es un proceso de negocio que afecta a varios departamentos como finanzas, control de gestión, ventas. Cada uno de estos departamentos podrá tener su propio almacén de datos departamental de la FIC, pero si nos centramos en la información de facturación, aunque cada almacén ofrezca una perspectiva distinta, la visión global deber dar una información de facturación completa e íntegra.

Este primer proyecto es el más importante en el desarrollo de la FIC, puesto que el resto de los proyectos dependen de él. Aun así, generalmente resulta muy difícil justificar su ejecución en la organización: es preciso plantear un proyecto de estudio que analice toda la organización y cuyo beneficio esté reflejado exclusivamente por el éxito de los futuros proyectos que se desarrollen. Disponer de un patrocinador con suficiente nivel en la organización facilita el desarrollo del primer proyecto y de la FIC.

Los beneficios del proyecto global de desarrollo no son inmediatos y, por este motivo, es más fácil de justificar y llevarlo a cabo, si disponemos de un patrocinador en la organización que ofrezca soporte al desarrollo de la FIC.

2.3.2. Desarrollo de proyectos autónomos

Mediante el desarrollo del primer proyecto, el proyecto global de desarrollo, hemos dividido el desarrollo de la FIC en forma de proyectos.

Habitualmente, son proyectos que se completan en unos seis meses, aunque posteriormente es posible hacer ampliaciones sobre estos.

Aunque no es fácil de conseguir, es muy importante que los proyectos sean autónomos; es decir, que cada proyecto tenga perfectamente delimitados sus objetivos y no dependa del desarrollo de otros proyectos en curso. Esto es así porque se trata de proyectos cuyos requerimientos son cambiantes y los usuarios no tienen totalmente claro qué necesitan ni tampoco qué les puede ofrecer la FIC. Si un proyecto no es autónomo y, por ejemplo, necesita los datos de otro proyecto que está en desarrollo, es muy probable que surjan conflictos entre ellos, puesto que seguramente las necesidades de datos del proyecto no autónomo cambiarán durante el desarrollo, y para el otro proyecto adaptarse a estas necesidades cambiantes será más costoso de lo que se esperaba inicialmente.

Resulta especialmente importante que los proyectos sean autónomos cuando se planifica su desarrollo por parte de diferentes empresas de servicios, sobre todo si estas lo llevan a cabo con un presupuesto cerrado.

Generalmente, muchos proyectos estarán interrelacionados, trabajarán sobre los mismos datos. Por lo tanto, si queremos que dos proyectos que trabajan sobre los mismos datos sean autónomos, deberán desarrollarse de manera secuencial.

Esta última condición frecuentemente no es fácil de imponer, puesto que se necesita disponer de los almacenes de datos departamentales lo antes posible. Aun así, es razonable si se quiere evitar un conflicto entre los proyectos. Si se desarrollan de manera paralela proyectos no autónomos, será crítico que el desarrollo del componente de integración y transformación se haga de manera que no se repitan elementos entre proyectos y tampoco surjan conflictos.

> El conflicto principal entre los proyectos desarrollados de manera paralela aparece al obtener los datos en el componente de integración y transformación.

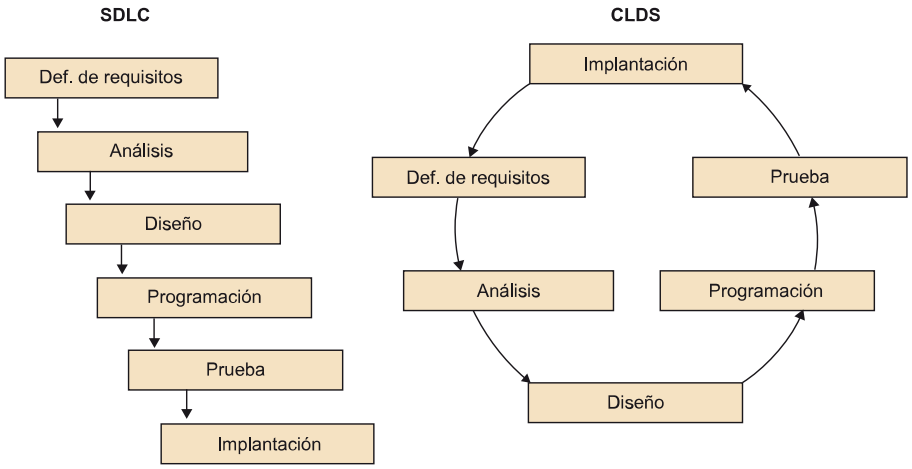
Todos los proyectos tienen que considerar el desarrollo de los componentes de obtención y almacenamiento de los datos y acceso a estos. Aun así, en algunos casos no se tienen que desarrollar todos los componentes porque ya están desarrollados. Es decir, se pueden plantear proyectos que se dediquen a explotar datos que están disponibles en el almacén de datos corporativo (proyectos que solo tienen componente de acceso). También puede haber proyectos cuya misión sea ampliar el conjunto de datos que hay en el almacén de datos corporativo (proyectos que solo tienen componentes de obtención y almacenamiento). En estos casos, la situación es diferente de aquella en la que se llevaba a cabo un desarrollo parcial de la arquitectura de la FIC, puesto que ahora se estudian todos los componentes de cada proyecto, pero algunos de estos ya están desarrollados. De igual modo, en el momento de llevar a cabo algunos proyectos, nos encontraremos tablas de bases de datos que podre-

mos utilizar y que ya están desarrolladas y actualizadas por el componente de transformación y carga de otros proyectos ya implementados. Generalmente nos encontramos tablas compartidas por muchos proyectos; se trata de tablas que serán críticas para la FIC y que acabarán siendo entidades maestras con un tratamiento especial en la FIC.

Para desarrollar cada uno de los proyectos, podemos aplicar la metodología de desarrollo en cascada o SDLC, o la metodología en espiral, también denominada CLDS.

Las fases de la metodología de desarrollo en espiral son las mismas que las de la metodología en cascada. La diferencia entre las dos es el orden de ejecución de sus fases, así como la forma en que se realizan, iterativa en espiral y secuencial en cascada.

Figura 12. Fases de la metodología de desarrollo



Es conveniente aplicar la metodología de desarrollo CLDS cuando no se tienen claros los requerimientos del sistema que hay que desarrollar y estos no se pueden descubrir de manera inmediata por medio de entrevistas convencionales con los usuarios.

Las fases de la metodología de desarrollo CLDS son las siguientes:

a) Se empieza por implantar una primera versión del sistema a los usuarios: esta podría ser un modelo en papel o un prototipo de sistema.

b) Los usuarios prueban esta versión.

c) Se lleva a cabo el desarrollo necesario para obtener, almacenar y analizar los datos de la versión de prueba.

d) Una vez desarrollados los programas necesarios, se realiza un diseño formal del sistema.

e) Se analizan los resultados del diseño, reformular y reprogramar si es necesario.

f) Como último paso, se entienden los requerimientos del sistema.

Estos pasos se repiten hasta tener desarrollado un sistema que cumple las necesidades de los usuarios. Los usuarios van descubriendo sus necesidades mediante el uso de versiones preliminares del sistema que van refinándose sucesivamente, hasta conseguir una versión final.

En el entorno de construcción de un almacén de datos departamental, es frecuente que los analistas no tengan una idea clara de las características del sistema que necesitan hasta que lo ven funcionando. En este caso, la metodología de desarrollo CLDS es más adecuada, puesto que permite hacer refinamientos sucesivos del sistema hasta definir claramente los requerimientos del sistema que se desea.

2.4. Evolución del entorno operacional

En cuanto se construye la FIC, algunos elementos del entorno operacional dejan de ser útiles y se pueden dejar de usar.

2.4.1. Evolución en el entorno operacional de telaraña

El entorno operacional había evolucionado a partir del desarrollo de programas de extracción y almacenamientos temporales de datos en lo que habíamos denominado entorno operacional de telaraña. Cuando se construía cada uno de los proyectos autónomos en los que se ha dividido la construcción de la FIC, parte de los programas de extracción y de los almacenamientos temporales de datos, es decir, parte de la «telaraña» deja de tener utilidad, puesto que su función pasa a ser ejercida por el nuevo almacén de datos departamental y, por lo tanto, se puede proceder a su desmantelamiento.

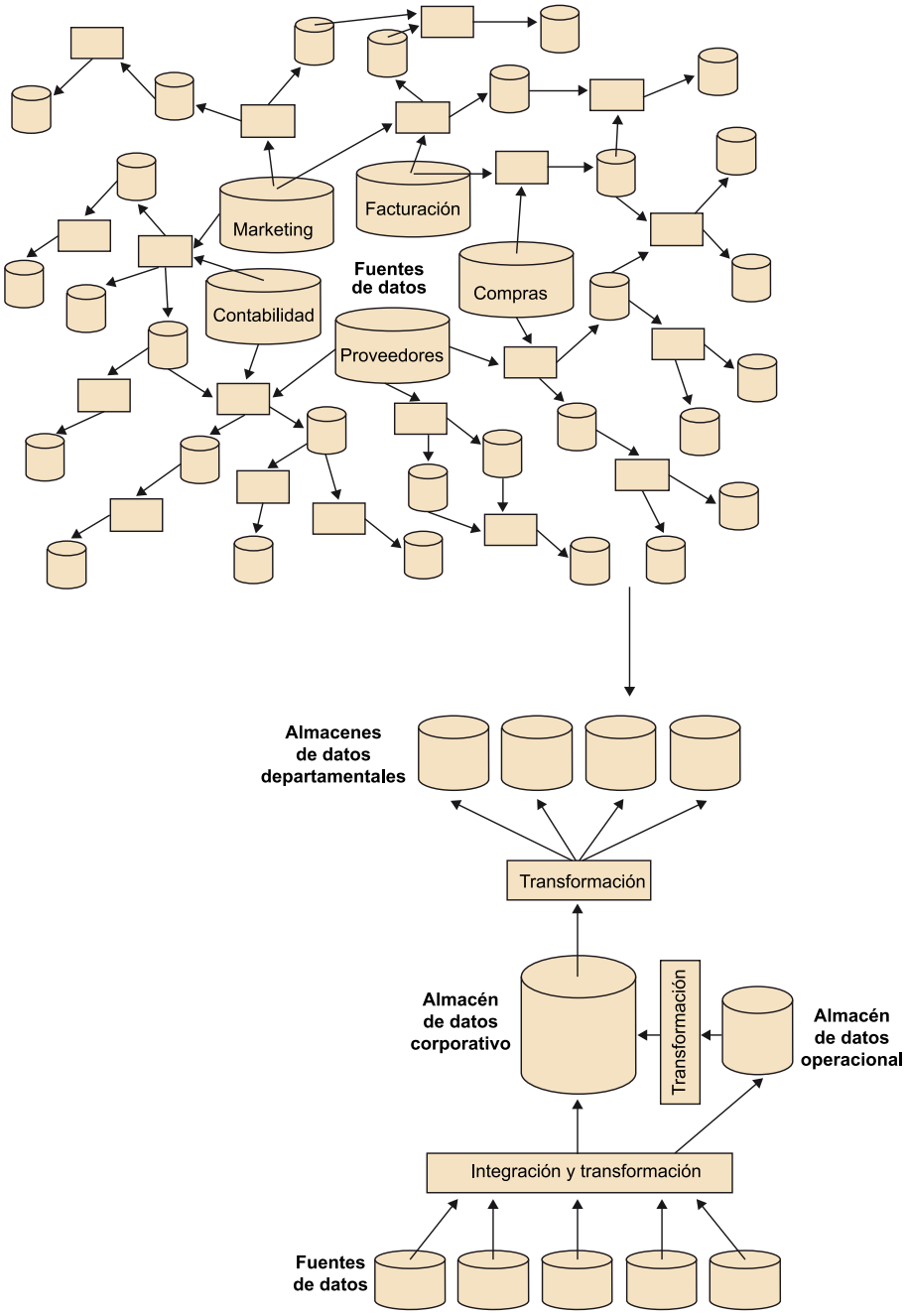
El **desmantelamiento de la telaraña** consiste en localizar los programas de extracción y los almacenamientos temporales de datos que dejan de ser útiles y evitar que se vuelvan a ejecutar o crear.

Es conveniente que las operaciones de desmantelamiento se realicen de manera progresiva dentro de cada proyecto de desarrollo de la FIC. Para desarrollar el componente de integración y transformación, deberían analizarse las diferentes fuentes de datos, lo que implica determinar y documentar la parte de la telaraña que queda cubierta y pasa a ser redundante con el desarrollo del nuevo proyecto.

El desmantelamiento de la telaraña se lleva a cabo de forma progresiva. Al construir un almacén de datos departamental, podemos desmantelar la parte de la telaraña del entorno operacional que ejercía la función del nuevo sistema construido.

El desmantelamiento de la telaraña del entorno operacional debe tenerse en cuenta cuando se planifica la construcción de la FIC. Si no se desmantela, la organización estará malgastando los recursos que consumen los programas de extracción y los almacenamientos temporales de datos.

Figura 13. Desmantelamiento del entorno operacional de telaraña



Podemos ver el resultado de desmantelar el entorno operacional en la figura 13. Pasamos de tener el entorno operacional de telaraña a tener la FIC, en la que las aplicaciones operacionales actúan como fuentes de datos y la extracción de los datos se hace de manera controlada y sistemática mediante el componente de integración y transformación.

Uno de los principales beneficios de la implantación de la FIC es la eliminación de la telaraña del entorno operacional. Su desmantelamiento trae un ahorro de tiempo al eliminar el trabajo de mantenimiento de un entorno de estructura compleja. Un almacén de datos no solo trae beneficios por la mejora en el almacenamiento de los datos y de los procesos de análisis que optimizarán la toma de decisiones y la detección de oportunidades, sino que trae consigo la eliminación de procesos y trabajos recurrentes que aportaban una información que adolecía de calidad y que generalmente era costosa de obtener.

2.4.2. Otros cambios en la organización

Con la FIC se produce una evolución del tipo de informes que utilizan los analistas de la organización. La FIC, principalmente mediante los almacenes de datos departamentales, permite a los analistas generar los informes a medida. Las aplicaciones operacionales y el entorno de telaraña creado a partir de estas permitían a los usuarios obtener informes estructurados de formato fijo.

La FIC permite a los analistas incrementar de manera considerable el uso de informes a medida, como apoyo al proceso de toma de decisiones, en detrimento del número de informes de formato fijo ofrecidos por las aplicaciones operacionales.

Por un lado, los equipos dedicados al desarrollo y mantenimiento de programas para la generación de informes (los programas de extracción que formaban la telaraña del entorno operacional) se tendrán que adaptar a la nueva situación y reubicar en algún otro departamento. Asimismo, en el supuesto de que alguno de los informes se generase de manera manual (situación bastante frecuente en algunas organizaciones) y ahora pasara a generarse dentro de la FIC, las personas encargadas de esta tarea se tendrán que reubicar para llevar a cabo otras tareas en la organización.

La FIC, además de afectar a la manera de trabajar de los analistas, afectará a los equipos de desarrollo que se dedicaban a generar los informes, que ahora pasan a generarse dentro de la FIC.

2.5. Uso del sistema de procesamiento analítico en línea (OLAP) en la FIC

Una vez construidos el *Data Warehouse* o los *Data Marts* (almacenes departamentales), necesitamos herramientas que nos permitan realizar consultas y analizar la información que contienen los almacenes de datos de forma fácil y productiva.

En la actualidad, existen dos tendencias en lo relativo a la explotación de datos. Estos dos modelos pueden ser complementarios y no necesariamente excluyentes, pues, como es habitual, cada uno de ellos aporta ciertas ventajas e inconvenientes al sistema de soporte a la toma decisiones.

2.5.1. Almacenes de datos con sistemas OLAP

Las herramientas OLAP permiten a los analistas realizar consultas complejas sobre grandes volúmenes de datos cargados previamente en los almacenes de datos departamentales, sin necesidad de tener conocimientos técnicos avanzados. Para que el sistema OLAP funcione correctamente, los *Data Marts* deben ser diseñados según el modelo multidimensional, lo que permite crear almacenes de datos multidimensionales, también conocidos como cubos OLAP.

La principal ventaja de este modelo es la robustez, consolidación y validación de los datos que aportan valor y calidad al sistema analítico, tanto al mismo dato como al resultado de las múltiples consultas realizadas. Al tratarse de un modelo preconfigurado según las necesidades y requerimientos de los mismos analistas, se puede garantizar que los cálculos y métricas obtenidas serán válidos para toda la organización y se refuerza la idea del dato único, donde básicamente se pretenden reducir o eliminar ambigüedades y diferentes interpretaciones sobre un mismo dato, lo que daría resultados diferentes para un mismo contexto.

Por contra, el principal inconveniente de los sistemas OLAP y el motivo principal por el que algunas organizaciones se plantean prescindir de estos es el alto coste de desarrollo que implica obtener una solución OLAP robusta. Además del tiempo necesario para su implementación y posterior mantenimiento, es necesario tener en la organización técnicos con conocimientos en estas tecnologías, o bien subcontratar este desarrollo. Para sacar el máximo provecho al modelo OLAP, también será necesario que los analistas conozcan lenguajes de consulta específicos como MDX.

En cualquier caso, si las condiciones de la organización son las adecuadas, disponer de un sistema OLAP de consultas multidimensionales aportará gran valor y conocimiento sobre sus propios datos, facilitando enormemente el trabajo de los analistas y mejorando el proceso de soporte a la toma de decisiones, en los diferentes ámbitos de la empresa.

2.5.2. Almacenes de datos sin OLAP

Con el auge de las tecnologías asociadas al fenómeno del *Big Data*, en los últimos años, han ido apareciendo nuevas herramientas de análisis que, de alguna manera, han ido superando las limitaciones clásicas de los modelos OLAP. Estas herramientas denominadas *self-service BI* están orientadas a los analistas para que puedan acceder directamente a los almacenes de datos y a otros orígenes de datos, sin necesidad de esperar a que los departamentos de TI preconfiguren un entorno OLAP de análisis, que, como hemos visto, puede llegar a ser un proceso bastante costoso.

Ciertamente, estas herramientas de *self-service BI* también requieren cierta formación técnica, pero están diseñadas para que se puedan llegar a utilizar sin necesidad de aprender a programar ni tener conocimientos técnicos avanzados, reduciendo la dependencia de los analistas con los departamentos de TI. Otra característica importante de este tipo de herramientas es que permiten realizar análisis y consultas avanzadas de manera visual, lo que facilita el traspaso del conocimiento hacia el resto de los profesionales de la organización, que no necesariamente tendrán conocimientos técnicos ni analíticos. Algunas herramientas de este tipo son Power BI (Microsoft), Tableau o QlikView, entre muchas otras.

En el ámbito de la ciencia de datos, el uso de herramientas gráficas avanzadas también facilita el descubrimiento de patrones ocultos en los datos, que, de otra manera, resultaría mucho más difícil de encontrar, así como realizar estudios e investigar nuevos escenarios.

Hemos visto que, con el uso de herramientas de *self-service BI*, los analistas tienen más libertad para explotar los datos, pero el resultado de estos análisis es más personal y no necesariamente seguirán los criterios de calidad establecidos por la organización. Puede darse el caso de que dos analistas obtengan resultados diferentes a un mismo problema debido a interpretaciones distintas y, en el peor de los casos, puede llegar a invalidar los dos resultados.

Veamos un ejemplo

Supongamos que un hospital necesita calcular la métrica «Estancia media» de los pacientes ingresados en traumatología en un periodo de tiempo. Este cálculo básico puede ser utilizado con posterioridad para calcular costes, contratar personal, organizar calendarios, provisionar fármacos, etc.

Aparentemente, parece un cálculo simple y podríamos decir que simplemente necesitamos calcular «Cuántos días han estado ingresados cada uno de los pacientes», para posteriormente calcular la media.

Sin embargo, este requerimiento se presta a diferentes interpretaciones de «¿Qué es una estancia hospitalaria?»:

- ¿Son los días que el paciente ha estado ingresado?
- ¿Son las noches que el paciente ha pernoctado en el hospital?
- Si el paciente ingresa un día a las 23:00 h, ¿este primer día cuenta como estancia?

- Si dan el alta hospitalaria al paciente un día a las 08:00 h, ¿este último día cuenta como estancia?

- ¿Cuántas horas debe estar ingresado como mínimo un paciente para que se considere una estancia?

Parece claro que, si la organización no establece un criterio único (típicamente recogido en los sistemas OLAP mediante métricas calculadas) y cada analista hace una interpretación diferente del concepto «Estancia media» en su análisis *self-service BI*, las diferencias en los cálculos derivados pueden ser significativas, llegando incluso a invalidarlos todos. El responsable (usuario final), que debe tomar una decisión basándose en esta información, no tendrá la certeza de disponer de información de calidad ni útil.

2.5.3. Modelos mixtos o complementarios

Muchas organizaciones disponen de sistemas OLAP consolidados, pero no quieren renunciar a los beneficios y agilidad de los sistemas de *self-service BI*. En estos casos, se diseña un modelo mixto, donde los dos modelos se complementan. El sistema OLAP ofrece la robustez y calidad de las explotaciones periódicas típicamente automatizadas, y las herramientas *self-service BI* ofrecen a los analistas de datos y negocio la posibilidad de explotar y explorar los datos con mayor flexibilidad.

En un modelo mixto, los analistas tienen las herramientas para explorar los datos y analizar todas las posibles interpretaciones. Tras un proceso de deliberación y consenso de los resultados, estos se validan por un comité de expertos y finalmente se traspasan al sistema OLAP. De esta manera, se comparte un único criterio en toda la organización y el dato precalculado puede ser utilizado por cualquier profesional de empresa, con garantías de calidad, incluso sin haber participado en el proceso de validación. Disponer de esta información en el sistema OLAP también permite automatizar ciertos procesos que deben ser recalculados periódicamente.

2.6. Perfiles en el equipo de gestión y desarrollo de la FIC

Además de los perfiles habituales en cualquier equipo de desarrollo de proyectos, en el equipo de desarrollo de la FIC aparece un conjunto de perfiles que no son habituales en otros proyectos y que estudiaremos a continuación junto a los perfiles de gestión característicos de la FIC.

2.6.1. El administrador de la FIC

El administrador de la FIC es su responsable ante la organización. Su misión es que la FIC se adapte a las necesidades de la organización. Por este motivo, debe conocer las posibilidades que esta ofrece, así como las necesidades de información de la organización, y controlar que las satisfaga lo suficiente.

Para cada proyecto que se desarrolle, se tiene que asegurar que cumpla los requerimientos de ámbito y funcionalidad, así como los que hacen referencia a coste y tiempo de desarrollo.

El administrador de la FIC es el máximo responsable ante la organización de que la FIC cumpla a lo largo del tiempo sus requerimientos de información.

2.6.2. Los analistas de requerimientos de negocio

El analista de requerimientos de negocio es el responsable del primer proyecto: el proyecto global de desarrollo. Contará con un equipo de analistas para su desarrollo. Su misión es identificar los requerimientos de información por parte de la organización y planificar el desarrollo de la FIC de modo que estos se cumplan.

El administrador de la FIC es el máximo responsable del entorno de la misma, mientras que los analistas de requerimientos de negocio son los interlocutores entre los usuarios de la FIC y el equipo que la tiene que construir y mantener.

Los analistas de requerimientos de negocio recogen las necesidades de información de los usuarios de la FIC y las transmiten al resto del equipo de desarrollo.

Además de ser responsables del proyecto global de desarrollo, también lo son de los diferentes proyectos autónomos en los que se divide la construcción de la FIC, que se desarrollarán posteriormente.

2.6.3. El arquitecto de la FIC

El arquitecto es el responsable del diseño de la FIC, diseña su arquitectura y se responsabiliza de los proyectos de desarrollo de infraestructura.

El **arquitecto de la FIC** analiza las fuentes de datos de la organización y diseña los esquemas de datos y la estructura de la FIC según los requerimientos de información de los usuarios y las posibilidades ofrecidas por las fuentes de datos.

El arquitecto de la FIC trabaja de manera conjunta con el analista de requerimientos de negocio en el proyecto global de desarrollo de la FIC, especialmente para definir los requerimientos de infraestructura de los proyectos.

2.6.4. El patrocinador de la FIC en la organización

El patrocinador de la FIC es una figura política cuyo objetivo es conseguir el apoyo necesario en la organización para desarrollarla salvando los obstáculos internos que surjan.

El ámbito de los proyectos de desarrollo de la FIC abarca toda la organización. Requiere integrar los datos de los diferentes departamentos y, una vez la FIC ya funciona, puede cambiar la manera de trabajar de equipos importantes de personal. Por todo esto, es frecuente que los responsables de algunos departamentos se muestren poco acostumbrados a compartir los datos o, generalmente, que algunos gestores de alto nivel en la organización muestren su oposición a la FIC por los cambios que representa. El patrocinador de la FIC tiene que interaccionar con todos ellos para intentar conseguir su apoyo o, al menos, reducir su oposición.

Por otro lado, según el planteamiento de construcción de la FIC mediante el desarrollo de proyectos autónomos presentado en este módulo, los diferentes proyectos de construcción de almacenes de datos departamentales están justificados, puesto que aportan un beneficio a la organización mayor que el coste que representa su desarrollo. Aun así, otros proyectos muy importantes, como, por ejemplo, el proyecto global de desarrollo y el proyecto de desarrollo de la infraestructura, no tienen una justificación inmediata de su coste. El patrocinador de la FIC en la organización tendrá como misión conseguir los fondos necesarios para el desarrollo de estos proyectos.

Teniendo en cuenta el tipo de personas con las que tiene que interaccionar y la tarea que debe llevar a cabo, interesa que el patrocinador de la FIC en la organización sea un directivo de suficiente nivel para que su tarea resulte más fácil. Además, tiene que estar convencido de las ventajas que la FIC puede proporcionar a la organización.

El **patrocinador de la FIC** en la organización es una figura política y tiene por objetivo conseguir el éxito en el desarrollo de la FIC. Generalmente, se trata de un directivo de alto nivel que conoce las ventajas que la FIC aporta a la organización y que se encarga de obtener los fondos necesarios y de resolver los problemas internos que surjan en la organización.

2.6.5. El gestor de cambios organizacionales

La implantación de cada uno de los proyectos de desarrollo de la FIC representa un cambio en la manera de trabajar en parte de la organización. El gestor de cambios organizacionales tiene como responsabilidad principal gestionar el impacto de la FIC en la organización.

Por un lado, cada proyecto de desarrollo de la FIC representa un cambio en la manera de trabajar de los usuarios del nuevo proyecto. Anteriormente obtenían los informes del Departamento de Informática o mediante personas que se los elaboraban manualmente, y con frecuencia los informes tenían forma de listados. Ahora, con la FIC, pueden generar ellos mismos los informes que necesitan y obtenerlos directamente en tiempo real. Dado que a muchas personas no les gusta ningún tipo de cambio, aunque este sea beneficioso a corto plazo, se tiene que hacer una tarea de marketing para facilitar la aceptación de los cambios. Esta tarea es responsabilidad del gestor de cambios organizacionales.

Por otro lado, el cambio en la manera de generar los informes puede desplazar a distintos trabajadores del Departamento de Informática u otros departamentos, que anteriormente eran los encargados de realizar la tarea que ahora pasa a estar cubierta por la FIC. Estos trabajadores serán muy útiles como miembros del equipo de desarrollo de la FIC o en otras actividades dentro de la organización. El gestor de cambios organizacionales tiene que intentar reducir la incertidumbre de estos trabajadores y colaborar para determinar cuáles serán sus nuevas funciones, así como para minimizar los posibles conflictos que se puedan producir.

El **gestor de cambios organizacionales** tiene como responsabilidad que los miembros de la organización acepten los cambios que implica la FIC.

2.6.6. El gestor de cambios de los metadatos

Los metadatos son un componente muy importante de la FIC, puesto que describen su estructura y contenido y, además, permiten interconectar el resto de los componentes entre sí.

Teniendo en cuenta su importancia, interesa disponer de una persona responsable de los metadatos de la FIC. Su misión es asegurar que estos reflejen la situación actual de la FIC y sean accesibles tanto para los analistas de información como para los equipos de desarrollo que lo requieran. Además, tiene que

asegurar que los metadatos puedan ser entendidos por los diferentes usuarios que acceden a los mismos. Teniendo en cuenta la variedad de herramientas que generan y utilizan metadatos, no resulta una tarea sencilla.

El **gestor de metadatos** es responsable de la situación y del acceso a los metadatos en la FIC.

2.6.7. Los analistas de la calidad del dato

Los analistas de la calidad de los datos tienen como responsabilidad asegurar que aquellos que se han obtenido de las fuentes de datos operacionales satisfagan los requerimientos de información de la organización.

Se encargan de verificar que los datos se han obtenido, transformado y cargado de la manera requerida y el resultado es de la calidad esperada; es decir, que los datos en los distintos almacenes de datos de la FIC son los apropiados y son correctos.

En caso de detectar datos incorrectos, la solución no es corregirlos donde se han detectado, sino encontrar la fuente de esta incorrección y solucionar allí el problema que se haya producido.

Los **analistas de la calidad de los datos** tienen que detectar aquellos que no se adapten al grado de calidad requerido por la FIC. Su misión no consiste en corregir estos datos, sino en identificar los motivos de la baja calidad y recomendar acciones que conduzcan a la solución de este problema.

La cantidad necesaria de analistas de la calidad de los datos dependerá del grado de calidad de los datos operacionales y del grado requerido. Durante las fases de desarrollo de la FIC, seguramente serán necesarios distintos analistas de calidad. Cuando la FIC esté en funcionamiento, debería bastar con una sola persona, incluso a tiempo parcial.

2.6.8. El administrador de bases de datos

En la organización debe haber personal de administración de bases de datos para el entorno operacional, pero también conviene que exista personal específico para el entorno de la FIC.

Aunque el sistema de gestión de bases de datos utilizado en el entorno operacional puede ser el mismo que el que se ha usado en los distintos almacenes de datos, sus características de configuración varían de manera radical entre los dos entornos. Por este motivo, interesa disponer de administradores especia-

lizados en el entorno de los almacenes de datos que sean capaces de obtener el máximo rendimiento de los sistemas y que no tengan que cambiar continuamente la manera de pensar al administrar tanto el entorno operacional como el de los almacenes de datos.

Asimismo, puesto que la FIC obtiene sus datos a partir del entorno operacional, es adecuado que los responsables técnicos de los dos entornos sean personas diferentes, de modo que la integridad de los datos de la FIC no se vea afectada por posibles problemas del entorno operacional. Si el administrador de los dos entornos fuera único, ante cualquier problema se puede producir un conflicto de prioridades; mientras que la máxima prioridad para un administrador de bases de datos propio será la FIC.

El **administrador de bases de datos** tiene que monitorizar de manera continua los procesos de obtención y acceso a datos, y configurar los sistemas para que estos se hagan de manera óptima.

2.6.9. Especialistas en obtener y acceder a los datos

Además de las figuras estudiadas en los apartados anteriores, el equipo de desarrollo de la FIC estará formado por especialistas en las operaciones específicas en las que se descomponen los proyectos de desarrollo de la FIC: obtención y almacenamiento de los datos y acceso a los mismos.

En cuanto a la obtención de los datos, particularmente si esta se hace mediante el uso de alguna herramienta de soporte, el equipo de desarrollo contará con los especialistas desarrolladores del componente de integración y transformación. Estos conocen el contenido y las posibilidades de las fuentes de datos y también los procedimientos y las herramientas para obtener los datos a partir de estos.

En cuanto al componente de acceso, el equipo de desarrollo contará con especialistas en las herramientas de acceso a los datos y metadatos que usarán los diferentes tipos de analistas de información de la organización. Estos especialistas construirán los métodos de acceso necesarios para satisfacer las necesidades de los usuarios en este sentido.

Tal y como ya se ha mencionado, el componente de integración y transformación es un componente clave de la FIC, y la necesidad de recursos en este componente debe quedar bien cubierta.

2.6.10. El Ingeniero de Datos (*Data Engineer*)

En la actualidad, han aparecido nuevos roles profesionales como el del ingeniero de datos. En realidad, son perfiles técnicos ya conocidos pero que, con la revolución del *Big Data*, se han ido especializando en la ingeniería y arquitectura de sistemas de gestión de datos. Estos profesionales se encargan de preparar y procesar la información, garantizando la calidad del dato y estableciendo las relaciones necesarias.

2.7. Usuarios de la FIC

Una vez tenemos construida la FIC en nuestra organización, o como mínimo alguno de los proyectos autónomos identificados, es el momento de dar acceso a todos los usuarios analistas para que realicen sus consultas sobre los diferentes almacenes de datos. Existen varios perfiles de usuarios de una FIC, pero todos ellos son, en mayor o menor grado, usuarios con conocimientos avanzados en análisis, explotación o visualización de datos.

Es necesario diferenciar entre usuario de la FIC y usuario final. Típicamente, el usuario de la FIC es un analista de datos que analiza y prepara la información que será consultada por el usuario final dentro del proceso de soporte a la toma de decisiones. Los usuarios finales pueden ser muy variados, pero es habitual encontrar roles como los de dirección, mandos intermedios, gestores, responsables de área, etc. Es decir, profesionales que necesitan datos actualizados y de calidad para poder tomar las decisiones de gestión, organización, operativas o estratégicas que consideren oportunas, según sus responsabilidades. No obstante, cada vez es más habitual que todo tipo de profesionales, sin un rol de gestión claramente definido, soliciten información a los analistas para mejorar y optimizar sus propios procesos productivos.

En los últimos años, estos perfiles se han ido popularizando y especializando, dando lugar a diferentes tipos de profesiones en el análisis de datos.

2.7.1. El analista de datos (*Data Analyst*)

Es el profesional con capacidad para comprender los datos del usuario final, sintetizarlos, contextualizarlos y obtener información útil para la organización. No dispone de la formación de un *Data Scientist*, pero tiene una visión amplia del negocio, muchas veces como resultado de años de experiencia, que le permite realizar tareas analíticas avanzadas.

2.7.2. El científico de datos (*Data Scientist*)

Es el profesional con conocimientos de negocio y técnicos, capaz de realizar consultas y análisis complejos, cruzando diferentes orígenes de datos. Un científico de datos debe ser capaz de extraer conocimiento de los datos, identificar patrones de comportamiento, implementar algoritmos de procesamiento de

datos y definir modelos predictivos. Para realizar su trabajo, un *Data Scientist* debe tener, entre otros, conocimientos matemáticos, estadísticos, de programación, de bases de datos, de minería de datos, de modelado y de visualización de datos. Otra faceta importante del científico de datos es la capacidad y habilidad de comunicación para poder explicar con claridad y sin ambigüedades el resultado de su trabajo.

2.7.3. El analista de negocio (*Business Analyst*)

Es el profesional experto en analítica de negocio con formación en modelado y visualización de datos mediante herramientas visuales: gráficos, imágenes, infografías... Este tipo de visualizaciones facilitan enormemente el traspaso y asimilación del conocimiento dentro de las organizaciones.

2.7.4. El responsable de datos (*Chief Data Officer*)

Es el líder de la estrategia de gestión y análisis de datos de la organización. Podríamos decir que tiene un perfil similar al del CIO (*Chief Information Officer*), típicamente el director de TI, pero especializado en la gestión del dato.

3. Desarrollo del componente de integración y transformación

En este apartado, estudiaremos las principales soluciones que se suelen aplicar en el desarrollo de algunas de las actividades del componente de integración y transformación. Nos centraremos en la obtención de las actualizaciones de los datos, las transformaciones, la integración y la actualización de los datos del almacén de datos, así como el soporte que ofrecen algunas herramientas del mercado para su implementación y la importancia de los metadatos.

3.1. Construcción de los componentes de extracción y obtención de datos

El proceso para obtener los datos a partir de las fuentes de datos origen y actualizar los datos de los almacenes de datos se lleva a cabo mediante un conjunto de aplicaciones que se ejecutan con esta finalidad. Este proceso se divide en dos fases:

- Obtener la imagen inicial.
- Obtener las actualizaciones.

Este componente es conocido también como ETL siglas del término en inglés *Extract Transform Load*.

3.1.1. Obtener la imagen inicial

La imagen inicial se obtiene con un conjunto de aplicaciones que generalmente se ejecuta una sola vez. El resultado de esta fase es una imagen de la situación actual de los sistemas operacionales, obtenida mediante un vaciado de sus respectivas bases de datos. Normalmente, la imagen inicial se obtiene sin dificultad; aunque, según las características de los sistemas operacionales, esto no siempre es así.

Ejemplo de obtención de la imagen inicial de los datos

En un sistema operacional basado en una base de datos relacional, es posible obtener los datos de manera inmediata, mediante un vaciado con las herramientas que el sistema de gestión de base de datos ofrece para ello. Aun así, en una aplicación empaquetada desarrollada sobre el sistema de ficheros, es posible que solo podamos acceder a los datos mediante las pantallas de la aplicación y se requiera, en este caso, un desarrollo complejo y costoso en el que se tengan que ir capturando los datos presentados en las diferentes pantallas.

Para reflejar en el almacén de datos la evolución que tienen estos, partiendo de su imagen inicial, debemos ir obteniendo las actualizaciones que se van produciendo. Nos encontraremos diferentes tipos de datos según su estructura:

1) Datos estructurados: con estructura de datos conocida, se almacenan principalmente en bases de datos relacionales. La manipulación se hace por medio de gestores de bases de datos, y las consultas, mediante SQL.

2) Datos semiestructurados: encapsulados en ficheros semiestructurados como XML5 o SGML6. En esta situación es posible trabajar con el contexto de negocio, lo que proporciona gran valor a las organizaciones. Actualmente encontramos bases de datos especializadas en XML para manipular este tipo de datos, y también técnicas como *web-mining* (minería de datos aplicada a la web) que permiten recuperar información de páginas web.

3) Datos no estructurados: encapsulados en objetos sin una estructura predefinida (audio, vídeo, PDF o Word) que requiere el uso de técnicas especiales como *text-mining* (minería de datos aplicada a ficheros de texto) o *information retrieval* (técnicas, con frecuencia estadísticas, aplicadas a encontrar información relacionada con un concepto en ficheros).

Actualmente, los datos no estructurados o semiestructurados, es decir, encapsulados en ficheros XML, SGML o incluso en objetos sin una estructura predefinida (PDF o Word), son también fuentes de alto valor potencial para las organizaciones. Permiten desde conocer información de los competidores hasta conocer en profundidad a los clientes. En el primer caso, por ejemplo, la información relativa a los precios del competidor se encuentra en su página web, pero donde realmente está la ventaja competitiva es en la automatización de este proceso.

En esta situación los procedimientos de extracción de información estándar, como ETL, con frecuencia no son suficientes y hay que utilizar técnicas de recuperación de información que utilizan métodos de reconocimiento de patrones, muestreo estadístico, probabilidad, etc. Esto complica la obtención de la imagen inicial de los datos, que no siempre está focalizada en los datos internos de la organización.

Por otra parte, debemos tener en cuenta la serie histórica que hay que obtener. Generalmente, los sistemas operacionales solo guardan una imagen de sus datos o bien una historia reducida de estos.

Si las diferentes imágenes almacenadas en los sistemas operacionales se han ido perdiendo a medida que se han hecho modificaciones, solo podremos disponer de la historia que almacenamos a partir del momento en que se construyan los almacenes de datos.

En algunos casos, por diferentes motivos (por ejemplo, por motivos legales), puede haber una historia más extensa de los datos, a veces fuera de los sistemas operacionales, aunque obtenida a partir de estos. En cada caso, tendremos

que valorar si es útil para los analistas disponer en los almacenes de datos de la historia que había antes; en caso positivo, en lugar de partir de la imagen inicial partiríamos de una película inicial.

Datos históricos en un banco

En un banco, el sistema operacional de gestión de movimientos de las cuentas solo guarda los datos de los últimos doce meses. Los datos de los meses anteriores hasta un total de cinco años se tienen que almacenar por motivos legales. Aun así, estos movimientos históricos no se almacenan en el sistema operacional, sino que mensualmente se extraen de la base de datos del sistema y se almacenan en un medio de almacenamiento más económico. Aunque estos datos permanecen accesibles dentro de la organización, solo se accede a los mismos de manera puntual, por motivos operacionales, no para analizarlos.

Para obtener la imagen inicial, tendremos que desarrollar un conjunto de aplicaciones de obtención de los datos de las fuentes de datos, las cuales generalmente se ejecutarán una sola vez. En algún caso y dado que hay que desarrollar también un proceso para realizar las actualizaciones de datos, podemos aprovechar este proceso para la obtención de la imagen inicial. Por ejemplo, si el proceso de actualización está obteniendo los datos de la fuente origen filtrados por un rango temporal (días, semanas, meses), nos podemos plantear ejecutar este proceso de actualización N veces para todos los periodos que compongan la serie histórica de la imagen inicial.

3.1.2. Métodos para obtener las actualizaciones de los datos

La manera de obtener los datos para las actualizaciones dependerá de los requerimientos de los analistas sobre los almacenes de datos, y también de las posibilidades ofrecidas por las fuentes de datos.

En ocasiones la obtención de actualizaciones consistirá en un simple filtrado sobre la información origen, como puede ser la obtención de datos correspondientes a un rango temporal.

Ejemplo de actualización mediante filtrado

Un operador de telecomunicaciones llamado ATEL tiene un almacén de datos en el que en una de las tablas guarda los datos de las portabilidades de operador realizadas por los clientes en los que el operador ATEL sea origen o destino de esta portabilidad. Los analistas de negocio analizan las portabilidades de cliente, la fecha tope de sus análisis es el día laborable anterior. En este caso la obtención de actualizaciones podría ser un filtrado de datos sobre la fuente origen en el que obtengamos las portabilidades realizadas el día anterior.

En otras ocasiones la obtención de las actualizaciones no se podrá realizar mediante un simple filtrado, debido a que no existe un rango temporal que defina unívocamente los datos que hay que actualizar o porque una frecuencia de actualización alta nos obligue a utilizar métodos de detección de actualizaciones. En la figura 14, se representan los diferentes métodos para obtener las actualizaciones:

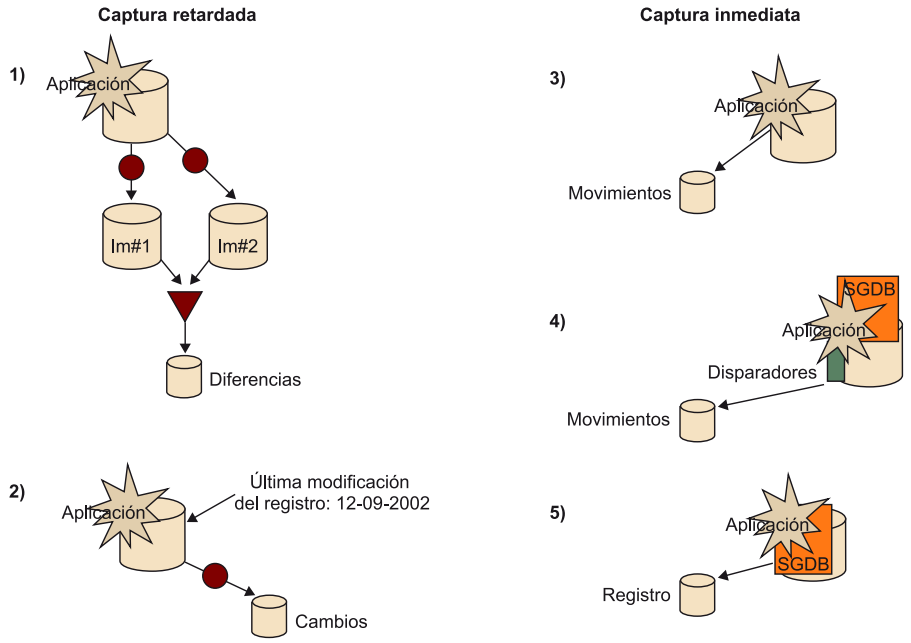
1) Comparación de imágenes (caso 1 de la figura 14): algunas fuentes ofrecen la posibilidad de obtener un vaciado de sus datos de manera masiva. Un ejemplo de este tipo de fuentes son los ficheros ordinarios. Si trabajamos con

bases de datos relacionales, el vaciado se puede obtener mediante consultas que seleccionan los datos que nos interesen. Para este tipo de fuentes de datos, podemos obtener las actualizaciones que se hayan producido comparando imágenes sucesivas que se hayan obtenido.

2) Fuentes con una huella de tiempo (caso 2 de la figura 14): en este caso, las fuentes almacenan para cada registro modificado el momento en que se llevó a cabo la modificación. De este modo, obtener las últimas modificaciones es inmediato: basta con lograr los registros marcados con una huella de tiempo posterior a la del último conjunto de modificaciones obtenido. Si queremos recuperar las operaciones de borrado será necesario realizar un borrado lógico en el que se marque el registro, pero no se elimine físicamente.

Estos dos métodos se denominan de captura retardada, puesto que no se intenta capturar las modificaciones en el momento en que se producen, sino posteriormente. Tienen el inconveniente de que pueden perder movimientos: si se hacen diferentes modificaciones sobre un registro y estas no se obtienen de manera inmediata después de producirse cada una, por medio de los métodos anteriores obtendremos un resumen de todas las operaciones hechas en una única operación de modificación.

Figura 14. Técnicas de obtención de actualizaciones



Ejemplo de pérdida de movimientos de los datos

En la imagen inicial del saldo de una cuenta obtenemos un valor de 1.000, y posteriormente se produce un ingreso de 500 y un reintegro de 700. Si en aquel momento volvemos a tomar la imagen del saldo de la cuenta por cualquiera de los dos métodos anteriores, obtenemos un valor de 800; si comparamos los saldos, obtenemos que se ha producido un reintegro de 200 y hemos perdido el detalle de los movimientos hechos.

3) Análisis del fichero delta de movimientos (caso 3 de la figura 14): algunas aplicaciones almacenan las modificaciones hechas sobre sus datos en un fichero de movimientos denominado fichero delta. Generalmente, se utiliza para auditar las operaciones. Podemos utilizar este fichero para obtener las modificaciones hechas sobre los datos que nos interesen.

4) Detección de movimientos mediante disparadores (caso 4 de la figura 14): si el sistema en el que se basa la fuente de datos ofrece la posibilidad de definir disparadores u otras funcionalidades de bases de datos activas, los podemos usar para obtener las modificaciones sobre los datos que nos interesen, activándolos cuando se produzcan acontecimientos de modificación sobre estos. Estos disparadores pueden comunicar los cambios producidos a las partes interesadas para que actúen de manera adecuada. Una manera habitual de actuar, en este caso, es la creación de un fichero delta de movimientos que se puede analizar posteriormente.

5) Análisis del fichero de registro (caso 5 de la figura 14): algunas fuentes de datos, basadas en SGBD que lo permiten, almacenan las operaciones que se sobre estas en un fichero de registro, generalmente por motivos de recuperación ante errores, o bien para que sea posible deshacer modificaciones. Si analizamos este fichero, se pueden obtener las modificaciones hechas sobre los datos que nos interesen. Esta solución presenta el inconveniente de que, aunque se genere en la fuente de datos, el fichero de registro no es de fácil acceso, puesto que está pensado para ser utilizado por el SGBD y su formato no siempre es de dominio público.

Estos tres últimos métodos de obtención de modificaciones sobre los datos, a diferencia de los dos primeros, tienen como característica principal que no pierden ninguno de los movimientos que se producen. Se les denomina métodos de captura inmediata, puesto que obtienen los movimientos en el preciso instante en el que se producen.

La ventaja que presenta el último caso es que no es invasivo contra la base de datos de origen ya que no accede ni modifica la estructura de las tablas de origen. Cualquiera de los métodos anteriores sí lo es, y puede afectar al rendimiento de la base de datos de origen. La desventaja de este último caso, como se ha dicho, es la dificultad para interpretar el fichero de registro que podrá ser diferente según el SGDB con el que trabajemos. Algunos SGDB disponen de herramientas para interpretar de forma sencilla este fichero, pero suelen ser herramientas propietarias y no disponibles en todos los SGDB.

Otro aspecto que hay que considerar en los diferentes métodos es la independencia respecto al SGBD, tal como se ha comentado anteriormente el caso 5 no es independiente del SGDB, como tampoco lo es la creación de disparadores del caso 4. El resto de casos si son independientes del SGDB.

6) Métodos de recuperación de información semiestructurada: las técnicas de acceso y consulta son las adecuadas para extraer información estructurada, encapsulada en formato XML y SGML. A pesar de que encontramos distintas aproximaciones, no se recomienda el uso de lenguaje de programación ni la creación de rutinas *ad hoc*. Los métodos recomendados son los siguientes.

a) Utilización de herramientas ETL especializadas que incluyen los mecanismos de manipulación de ficheros XML y también la posibilidad de analizar la diferencia entre las versiones de los documentos.

b) Uso de motores especializados de indexación de información semiestructurada como Apache Lucene.

7) Métodos de recuperación de información no estructurada. En el caso de la información no estructurada, también hay diferentes aproximaciones. Los diferentes modelos que encontramos son estos:

a) Booleano: método de recuperación simple fundamentado en la teoría algebraica booleana. Establece una relevancia binaria de modo que un documento es relevante o no lo es. Su simplicidad ha motivado que caiga en desuso.

b) Vectorial: compara la consulta con el texto en un documento y genera un vector. Vectores que apuntan a la misma dirección son similares y el grado de verosimilitud es la función del ángulo entre los vectores. Es uno de los métodos actuales más populares y fiables. En particular, motores Java como Apache Lucene lo utilizan.

c) Modelo LSI (*Latent Semantic Indexing*): el objetivo es calcular la verosimilitud entre la consulta y el documento mediante la similitud entre conceptos, y no entre palabras.

d) Métodos mejorados: los métodos anteriores se han mejorado con el uso de motores semánticos, redes semánticas o el análisis de la regresión.

3.1.3. Criterios de selección del método para obtener las actualizaciones de los datos

¿Qué método utilizaremos de entre los métodos que hemos presentado para obtener modificaciones? La respuesta a esta pregunta dependerá en gran medida de las funcionalidades ofrecidas por la fuente de datos y por los requerimientos de los usuarios de los almacenes de datos. Los criterios siguientes no son estrictos, pero nos pueden servir como orientación:

a) Si la fuente de datos dispone de un fichero delta de movimientos, generalmente esta será la opción que elegiremos, puesto que este fichero suele ser directamente accesible y fácil de analizar.

b) Si no existe fichero delta, se analiza el registro en caso de que lo genere la fuente de datos. En este caso, la accesibilidad es menor y la dificultad para analizarlo, mayor.

c) Si la fuente de datos está basada en una base de datos que permite crear disparadores, podremos utilizar este método. La definición de disparadores hará que las transacciones sean más costosas en tiempos de ejecución; además, el coste de desarrollo de esta solución suele ser mayor que el de las anteriores. Aun así, podemos obtener las modificaciones en el mismo momento en que se han producido, sin necesidad de hacer un análisis posterior.

d) Si la fuente de datos almacena una huella de tiempo para cada modificación hecha, se preferirá esta solución a la de comparación de imágenes, puesto que es más inmediata y el coste para obtener las modificaciones es mucho menor. Aunque el coste también puede ser menor en este caso que en casos anteriores, el hecho de que perdamos movimientos hará que esta solución sea menos atractiva que aquellas que no presentan este inconveniente. Si deseamos recoger las operaciones de borrado deberemos activar un mecanismo en las tablas origen como puede ser el borrado lógico.

e) La opción que se suele elegir como último recurso es la de comparar imágenes. Presenta los inconvenientes de que con esta podemos perder movimientos y además el coste en tiempo de ejecución puede ser bastante más elevado que el de las otras soluciones, dependiendo de la medida de la fuente de datos. Aun así, esta opción es inmediata, no exige ninguna funcionalidad especial a las fuentes de datos (siempre la podemos implementar) y, por lo tanto, frecuentemente es la elegida. Por otro lado, muchas herramientas ETL disponen de componentes para realizar esta comparación.

Ejemplo de problema de obtención de las modificaciones de los datos

Queremos obtener los datos de una fuente que no crea un fichero delta, que está basada en una base de datos a cuyo fichero de registro no se puede acceder y que no permite la definición de disparadores, además de que tampoco almacena una huella de tiempo con las modificaciones. En esta situación, la alternativa más inmediata es la de comparar imágenes. Otra posibilidad sería la de modificar la fuente de datos para que genere un fichero delta de modificaciones; aun así, esto no siempre es factible (por ejemplo, si la fuente de datos es una aplicación empaquetada) o puede resultar demasiado costoso dependiendo del tamaño de la aplicación.

3.2. Construcción de los componentes de transformación, integración y depuración de datos

Cuando ya hemos obtenido los datos de las diferentes fuentes, es necesario preparar la información antes de cargarla en el almacén de datos. Para ello seguiremos los pasos siguientes:

a) Cada conjunto de datos puede tener una estructura distinta dependiendo de la fuente de la que proceda. Los tenemos que transformar para adaptarlos a la estructura del esquema del almacén de datos en el que se almacenarán: operaciones de selección de columnas, filtrado de registros, agregaciones, etc.

b) Tenemos que depurar los errores o conflictos que podamos encontrar dentro de los datos de cada una de las fuentes.

c) Tenemos que integrar los datos depurando errores o conflictos entre datos de fuentes distintas.

d) Tenemos que crear las columnas derivadas de la información de origen y que sean necesarias para el almacén de datos.

Como resultado de este proceso, obtendremos un conjunto de datos directamente utilizable para actualizar el almacén de datos correspondiente.

A continuación, comentaremos algunos detalles de estas operaciones por separado. Esto no significa que se hagan de manera secuencial, sino que se pueden combinar o intercalar algunas de estas según las necesidades

3.2.1. Transformación de los datos

Las transformaciones que hay que hacer sobre los datos para preparar la información, según se ha comentado en el apartado anterior, pueden ser muy variadas. Entre las más frecuentes, encontramos las siguientes.

- Cambiar el formato o el tipo de los datos (por ejemplo, los campos de fecha).
- Cambiar la codificación (por ejemplo, EBCDIC a ASCII).
- Reestructurar los campos (por ejemplo, fusionar o dividir campos o cambiar su orden relativo).
- Cambiar las unidades o los códigos de representación (por ejemplo, cambios de moneda).
- Cambiar el grado de agregación (por ejemplo, calcular las ventas mensuales a partir de las diarias).
- Calcular campos derivados (por ejemplo, calcular la edad a partir de la fecha de nacimiento).

- Generar claves subrogadas: creación de claves subrogadas (claves internas) que nos van a permitir guardar diferentes versiones de una clave de negocio cuyos atributos cambian en el tiempo.

- Discretizar valores: variables continuas que discretizamos clasificándolas según rangos de valores (por ejemplo, en el consumo de clientes, pasamos de un valor continuo a rangos de consumo: 'alto', 'medio', 'bajo').

- Encriptar campos por cuestiones de seguridad.

- Añadir información temporal (por ejemplo, periodo de validez de los datos).

Una de las transformaciones que generalmente siempre debe hacerse es la última mencionada, añadir a los datos información temporal. Se tendrá que añadir la información sobre el periodo de validez de los datos o el momento en el que se haya registrado la modificación (o en que se haya detectado), según sea requerido por el almacén de datos correspondiente. De este modo, secuenciamos las imágenes obtenidas para ir formando la película que almacena el almacén de datos.

Ejemplo de información temporal y fechas de vigencia

En un almacén de datos de una compañía de seguros, que guarde información de siniestros podemos tener la dimensión pólizas, en la cual almacenamos el capital asegurado de la póliza, la provincia del tomador, y otros datos asociados a la póliza que pueden cambiar con el tiempo, y que en cada cambio generarán un nuevo registro (es decir, una nueva versión) en la tabla de pólizas. De este modo, podemos tener por un lado la tabla con los datos de los siniestros de las pólizas que debemos asociar a la tabla con los datos propios de las pólizas. Tener versionados los datos de la tabla de pólizas permitirá asociar a cada siniestro la versión correspondiente de la póliza a la fecha correspondiente (ejemplo, fecha de siniestro).

Las transformaciones aseguran la correcta adecuación de los datos de origen a las tablas del almacén de datos.

3.2.2. Depuración de los datos

El objetivo de depurar los datos obtenidos de las diferentes fuentes es mejorar su calidad. Algunas de las incidencias más comunes que se producen son las siguientes:

- Detectar y corregir valores inconsistentes (por ejemplo, un atributo edad con un valor de trescientos cincuenta).

- Añadir valores por defecto a los campos con valores no definidos. generalmente, se hace de acuerdo con criterios marcados por el almacén de datos al que se destinan estos según la fuente de datos. El valor suministrado

puede ser constante, calculado o, en algunos casos, puede interesar dejarlo sin definir.

- Detectar y corregir información duplicada. A veces es difícil de detectar, puesto que se tienen distintas representaciones del mismo valor (por ejemplo, diferentes maneras de escribir el nombre de una calle en los datos del domicilio). Será más frecuente encontrar información duplicada entre distintas fuentes de datos, pero también la podemos encontrar dentro de una misma fuente.

- Detectar y corregir errores de integridad referencial entre entidades relacionadas (por ejemplo, no puede darse el caso de incorporar al almacén de datos información de ventas de un producto que no existe en la entidad productos de nuestro almacén).

Será necesario definir la acción que hay que realizar en el momento de detectar la incidencia. Existen diferentes opciones:

a) Rechazar información de origen completa (por ejemplo, fichero completo).

b) Rechazar el registro erróneo.

c) Corregir el registro erróneo e insertarlo en al almacén de datos.

Igualmente, será necesario activar el mecanismo de comunicación de la incidencia detectada (alertas, notificación vía e-mail, etc.).

Ejemplo de depuración de datos

Es habitual que en un almacén de datos tengamos falta de sincronización en la actualización de datos que tienen que estar íntegros entre sí, como pueden ser los datos de ventas y los datos de productos. Por ejemplo, podemos tener un código de producto que recibimos en el fichero de ventas y que no existe en la tabla maestra de productos. Una posible solución es integrar el registro con valor «desconocido». En este caso, no rechazamos el registro del fichero de ventas, lo integramos pero con valor «desconocido» o un código que establezcamos para el valor indeterminado de esa dimensión (ejemplo: P9999). Este valor indeterminado, si existe en el maestro de productos, es un registro que sirve para aglutinar todos los productos no clasificados. El registro de ventas erróneo se actualiza en el almacén de datos y se visualiza en los informes, pero clasificado en un valor «desconocido», hasta que el maestro de productos se actualice y el fichero de ventas se reprocese. Esta solución nos permite cargar el registro, aunque no quede bien clasificado en los productos. Esta opción es mejor que rechazar el registro, ya que al rechazar perdemos la información y podríamos tener un porcentaje de registros considerables afectados por esta incidencia.

Así mismo, debemos decidir si las correcciones sobre los datos erróneos deben realizarse en el componente de transformación e integración o en los sistemas de origen. Es preferible, que los errores se solucionen en origen, ya que si los solucionamos en el componente de transformación e integración el error seguirá estando en origen y seguiremos recibiendo información errónea.

El objetivo de la depuración de los datos es garantizar la calidad de los que incorporamos al almacén de datos.

3.2.3. Integración de los datos

Los datos provenientes de diferentes fuentes deben de integrarse entre sí y con los datos del almacén de destino. La forma de conseguirlo puede variar.

El **proceso de integración** será diferente dependiendo de si hacemos la carga inicial del almacén de datos o una actualización de este.

Además del volumen de datos que hay que tratar, la diferencia principal reside en el hecho de que en las actualizaciones, para hacer la integración, podemos usar las correspondencias entre los datos de las fuentes y los del almacén de datos previamente establecidos en la carga inicial o en actualizaciones anteriores. Generalmente, en el proceso de carga inicial se hará una integración de todos los datos previa a la carga en el almacén de datos. Por otro lado, cuando se hace la actualización, es posible que no estén disponibles los datos de todas las fuentes al mismo tiempo e interese integrar los datos de las distintas fuentes por separado en el almacén de datos.

El problema principal con el que nos encontramos consiste en detectar qué datos representan el mismo concepto.

Si las diferentes fuentes de datos utilizan como clave el mismo campo de la entidad (por ejemplo, NIF), se pueden relacionar sin dificultad, excepto por errores en los datos. El problema surge cuando cada fuente de datos emplea su clave (por ejemplo, un código generado) y no hay campos comunes que puedan servir como clave alternativa para establecer relaciones entre sí o, si los hay, sus valores se representan de manera diferente entre las fuentes.

Durante el proceso de integración, se transformarán los datos para homogeneizar su representación y se eliminará la información duplicada.

Se tendrán que establecer los procedimientos adecuados para propagar las correcciones hechas hasta los sistemas operacionales de los que proceden los datos. Estas serán especialmente relevantes después de obtener los datos para la carga inicial del almacén de datos, pero también se deberán tener en cuenta las efectuadas en cada una de las actualizaciones de los datos.

Datos depurados

Si hemos dedicado un esfuerzo considerable para integrar los datos de clientes de distintos sistemas operacionales, depurándolos y eliminando duplicados, lo razonable es uti-

lizar los datos depurados en los sistemas operacionales, en lugar de continuar utilizándolos con errores. Por este motivo, se tendrá que definir un sistema para propagar las correcciones hechas en los datos desde el componente de integración y transformación hasta los sistemas operacionales de los que proceden.

3.3. Construcción del componente de actualización de los datos en los almacenes de datos

Cuando ya hemos obtenido los datos a partir de las fuentes de datos, se transforman, depuran e integran si es necesario y, finalmente, se transportan al almacén de datos para proceder a cargar o actualizar los datos que hay.

3.3.1. Métodos de actualización de los almacenes de datos

La actualización de los datos de los diferentes almacenes de datos se puede llevar a cabo de las siguientes formas:

1) Carga: mediante la operación de carga, el almacén de datos de destino pasa a contener exclusivamente los datos que se indican en esta operación. Si previamente contenía otros datos, estos son reemplazados por los nuevos.

2) Adición: la operación de adición permite añadir los datos indicados en el almacén de datos. Podría darse el caso de que algunos de los datos que se desea añadir estén ya en el almacén de datos. Si se produce esta situación, se puede elegir una de las dos alternativas siguientes:

a) Fusión destructiva: la fusión destructiva permite añadir los datos indicados en la operación a los que ya había previamente de otro modo. Si la clave de un registro de los datos indicados en la operación coincide con la clave de alguno de los datos existentes, el registro previo es reemplazado por el registro nuevo. Los nuevos registros cuya clave no coincide con otros que ya había se añaden directamente al almacén de datos.

b) Fusión constructiva: la diferencia respecto a la destructiva consiste en el hecho de que, cuando coincide la clave de alguno de los registros que se tiene que añadir con la de un registro existente, marca estos registros pero no los reemplaza con los nuevos valores. Se insertan los nuevos registros y los anteriores quedan marcados. De este modo, posteriormente se puede actuar sobre los registros de manera conveniente.

3.3.2. Selección del método de actualización

Para cargar un almacén de datos a partir de los obtenidos de los sistemas operacionales, utilizaremos una combinación de las operaciones que hemos estudiado en el apartado anterior. Usaremos una u otra dependiendo de la situación de los datos que se tienen que cargar y del almacén.

La carga inicial de los datos se hará mediante una operación de carga, puesto que partiremos de un almacén de datos vacío. Esta operación, incluso, nos permitirá crear de manera automática las tablas del almacén de datos si no estaban creadas anteriormente.

La actualización de los datos se hará con una combinación de las otras tres operaciones (adición, fusión destructiva y fusión constructiva). Con frecuencia bastará realizar operaciones de adición, puesto que iremos almacenando nuevas imágenes de los datos. Por lo tanto, no encontraremos datos duplicados.

En algunos casos, interesa actualizar algunos campos de los datos existentes a partir de los nuevos datos obtenidos. Entonces, se utilizará la fusión constructiva.

Ejemplo de fusión constructiva

En el almacén de datos, para cada registro se indica mediante dos campos de tipo fecha (fecha de inicio, fecha final) el periodo de validez del resto de los campos del mismo. Al añadir un registro nuevo, el campo fecha final permanece no definido. Si se produce una modificación sobre cualquier campo del registro en la fuente de datos operacionales, lo que haremos será añadir un nuevo registro en el almacén de datos con los valores de los campos actualizados (el campo fecha final no estará definido). Sin embargo, además, tendremos que actualizar el campo fecha final de la versión previa del registro para indicar que estos valores ya no son válidos. Esta operación de actualización se llevará a cabo mediante la operación de fusión constructiva.

De manera adicional a la operación de carga inicial, se distinguen dos modos de actualización de los distintos tipos de almacenes de datos:

1) Refresco total: todos los datos del almacén de datos se obtienen de nuevo y se cargan mediante la operación de carga. Esta situación se puede dar para cargar los datos de los almacenes de datos departamentales a partir del almacén de datos corporativo.

2) Mantenimiento incremental: se trata de minimizar el tiempo requerido para hacer la actualización, para lo cual se parte de los datos existentes y se buscan procedimientos para actualizarlos según los nuevos datos. Para hacer el mantenimiento incremental, utilizaremos el resto de las operaciones estudiadas. Generalmente, esta será la manera de actualizar los datos en los diferentes almacenes de datos.

Los procesos de actualización en las tablas del almacén de datos pueden ser procesos costosos en caso de trabajar con volúmenes de datos altos. Existen técnicas para optimizar el rendimiento de estos procesos y las herramientas de apoyo suelen tener utilidades para implementarlos. Algunas de estas técnicas se basan en la desactivación de restricciones de las tablas o utilidades de la base de datos con objeto de mejorar los procesos de actualización (desactivación de índices, claves primarias, escritura en log de base de datos) y otras técnicas, en

dividir el conjunto de registros por actualizar en varios subconjuntos y ejecutar en varios hilos el procesos de actualización, poniendo al día cada subconjunto en un hilo independiente.

3.4. Frecuencia y ventana de actualización

3.4.1. Frecuencia de actualización en un almacén de datos

Cada almacén de datos tiene unos requerimientos de actualización propios y, además, estos no necesariamente deben ser homogéneos para todos los datos dentro de un almacén de datos.

Requerimientos de frecuencia de actualización de diferentes campos

En un almacén de datos operacional de un banco, disponemos de las direcciones de los clientes y de los datos del saldo de las cuentas. En el supuesto de que se produzca algún cambio en la dirección de un cliente en un sistema operacional, nos interesa que esta se actualice en el almacén de datos operacional al final del día, puesto que es exclusivamente entonces cuando utilizamos las direcciones para generar las etiquetas de la correspondencia que se envía a los clientes. Aun así, cualquier cambio en el saldo de alguna de sus cuentas registrada en una aplicación operacional interesa que se actualice lo más rápido posible en el almacén de datos operacional, puesto que se utilizará este campo para determinar si se autorizan o no al cliente las operaciones de crédito hechas con sus tarjetas.

Por lo tanto, para cada almacén de datos y cada uno de sus datos se tendrá que definir la frecuencia de actualización. Cuanto más fuertes sean estos requerimientos, el coste de actualización será más elevado. La tendencia en las empresas es a tener un dato cada vez más actualizado, lo que implica una frecuencia de actualización cada vez mayor. Lo que se busca es una actualización próxima al tiempo real. Conseguirlo no es trivial e implica la utilización de técnicas avanzadas tanto en la detección de la actualización como en su propagación al almacén de datos. Por otra parte, la actualización próxima al tiempo real puede generar confusión en el analista de negocio respecto a la actualización de un dato determinado y puede encontrarse resultados distintos al ejecutar un informe dos veces a lo largo del día. En estos casos, lo que se suele hacer es dividir la tabla del almacén en dos: la primera es una tabla que se actualiza en proceso *batch* (partición estática) con una periodicidad definida (por ejemplo diaria) y la segunda tiene una actualización próxima al tiempo real (partición dinámica). Esta segunda tabla contiene los cambios que se han producido desde la última actualización de la partición estática y se actualiza varias veces durante el día. El separar en dos particiones tiene la ventaja de que la actualización de la partición dinámica (varias veces al día) no afecta a las consultas que se realizan sobre la partición estática y que los usuario tienen más claro el nivel de actualización que se van a encontrar cuando acceden a una u otra partición.

Generalmente, el almacén de datos operacional es el que tiene unos requerimientos más estrictos en este aspecto. En muchos casos, lo ideal es que la actualización sea inmediata, pero esto no siempre es factible. En este sentido, depende de las posibilidades que ofrezcan las fuentes de datos.

El almacén de datos corporativo suministra los datos a los almacenes de datos departamentales; por lo tanto, los requerimientos en este sentido de los distintos almacenes departamentales quedarán reflejados en el almacén de datos corporativo.

Para cada dato de los diferentes almacenes de datos se tiene que definir la frecuencia de actualización: cada cuánto tiempo se tiene que actualizar a partir de las fuentes de datos.

3.4.2. Ventana de actualización del almacén de datos

El objetivo principal de los almacenes de datos es apoyar el proceso de toma de decisiones. Por lo tanto, las operaciones que se suelen realizar sobre estos son de consulta. Para optimizar la ejecución de este tipo de operaciones, los SGBD sobre los cuales están implementados los almacenes de datos están configurados de modo que solo permiten hacer operaciones de consulta.

Si se quiere hacer otro tipo de operaciones distintas de las de consulta –por ejemplo, las de actualización del almacén de datos–, hay que parar los sistemas, cambiar la configuración, hacer las operaciones necesarias y restaurar la configuración para que las consultas continúen siendo óptimas. Durante este tiempo, el almacén de datos no está disponible para sus usuarios.

Disponibilidad del almacén de datos corporativo

En el caso extremo de requerir una disponibilidad total (veinticuatro horas al día todos los días del año), una posible solución es tener la base de datos del almacén de datos replicada y actualizar una de las copias mientras se utiliza la otra. Una vez actualizada esta, se puede usar mientras se actualiza la otra.

La **ventana de actualización** de un almacén de datos es el tiempo necesario para hacer las operaciones que lo actualizan. Durante este período de tiempo el almacén de datos no está operativo para realizar consultas.

Relacionado con la ventana de actualización, tenemos el concepto de la ventana de extracción; esta se refiere al tiempo necesario para obtener las modificaciones de los datos a partir de las fuentes de datos.

La **ventana de extracción** es el tiempo necesario para obtener los movimientos a partir de las fuentes de datos.

Según el método de obtención de las modificaciones, la ventana de extracción será más o menos amplia. Generalmente, el método de obtención de modificaciones mediante comparación de imágenes es el que requiere una ventana de extracción más amplia, puesto que las operaciones que tiene que hacer consumen mucho tiempo. Además, junto con el método de obtención de modificaciones basado en fuentes con huella de tiempo, requiere que se haga un proceso en las plataformas de las fuentes de datos. Esto último puede resultar problemático si estas plataformas están sobrecargadas de trabajo.

En el caso del resto de los métodos, es suficiente con obtener el fichero de registro o los respectivos ficheros de movimientos y analizarlos en una plataforma distinta. En el método de comparación de imágenes, la comparación también se puede llevar a cabo en otra plataforma, pero la extracción de cada una de las imágenes (solo se tiene que extraer una cada vez) debe de hacerse en la plataforma de la fuente de datos. Finalmente, si se usa el método basado en la huella de tiempo, generalmente el análisis de los datos se tiene que hacer en la plataforma que los contiene.

El componente de integración y transformación debe tener presentes los requerimientos de disponibilidad de las fuentes de datos, así como los de los almacenes de datos. Por este motivo, resulta muy importante minimizar el tiempo de proceso de los diferentes pasos, como se ha ido señalando cuando se han presentado, de modo que se puedan ejecutar dentro de las ventanas de extracción y actualización disponibles en cada caso.

Ejemplo de ventana de actualización

Es muy habitual la realización de procesos de actualización del almacén de datos en procesos *batch* nocturnos. Por ejemplo, en un banco podemos realizar la actualización de los movimientos de los clientes en un proceso *batch* diario, de forma que los usuarios cada día tienen la «foto» de cierre del día anterior. Esta ventana de actualización no afecta a las consultas que se realizan durante el día y el refresco de los datos se ajusta muy bien al ritmo natural del negocio (movimientos diarios).

3.5. Herramientas de apoyo al desarrollo

Las necesidades de los diferentes usuarios cambian con el tiempo, especialmente las de los analistas; por lo tanto, es habitual que se produzcan cambios tanto en las fuentes de datos como en los almacenes de datos. Por sus características de elemento intermediario entre el resto de los elementos de la FIC, el componente de integración y transformación se verá afectado por los cambios en cualquiera de los otros elementos. Por lo tanto, además de cumplir con las funciones que hemos explicado antes, este componente debe ser lo bastante

flexible como para que se pueda adaptar a los cambios que se produzcan en cualquiera de los componentes con los que interacciona y, además, lo tiene que hacer de manera inmediata.

El componente de integración y transformación está formado principalmente por software. Para desarrollar cualquier componente de software, se presentan dos alternativas:

- Desarrollo manual.
- Desarrollo automático o con apoyo automático.

3.5.1. Funcionamiento de las herramientas

Si lo que necesitamos fundamentalmente en el componente de integración y transformación es reaccionar a los cambios de manera inmediata, la solución más adecuada será llevar a cabo un desarrollo con soporte automático. Puesto que esta necesidad es ampliamente reconocida, en el mercado hay herramientas orientadas de manera especial para el desarrollo de este componente. Estas herramientas ofrecen un conjunto de transformaciones tipo de los datos, así como otras funcionalidades específicas para soportar las operaciones requeridas.

Este tipo de herramientas permiten implementar el componente de transformación e integración utilizando una interfaz gráfica que nos permite definir desde las fuentes origen todo el flujo de proceso hasta la tabla destino del almacén de datos. Todos los mapeos, transformaciones y volcados definidos son guardados en un repositorio de metadatos. Se suele trabajar en dos niveles: de flujo de proceso, entendiéndolo como trabajo, y de transformación, como paso de dicho flujo. Las configuraciones se realizan a estos dos niveles. Casi todas las herramientas guardan cierta similitud en la forma de diseñar los flujos. Existen muchas transformaciones estándares (lectura, filtrado, unión de conjuntos de datos, ordenaciones, agregaciones, actualización de tablas, etc.) habituales en el componente de transformación e integración que vienen predefinidas en las herramientas y lo único necesario es parametrizarlas.

Ejemplo de paso predefinido en herramienta

Es muy habitual crear pasos de lectura en el componente de transformación e integración. Esta lectura puede realizarse desde fichero plano, Excel, XML, base de datos, etc., las herramientas incorporan diferentes conectores para obtener datos de muchas tipologías de fuentes. Por ejemplo, para leer datos de una tabla de un SGBD determinado, en el paso predefinido de lectura de BBDD hay que indicar el SGDB, el servidor donde se aloja la base de datos, el nombre de la base de datos, las credenciales y componer la consulta que va a recuperar los datos.

3.5.2. Ventajas e inconvenientes de las herramientas

Tanto el hecho de usar una herramienta como de llevar a cabo un desarrollo manual presentan una serie de ventajas e inconvenientes. Dada la situación de cada organización, deberá determinarse la solución más conveniente.

Las **ventajas** de hacer el desarrollo manual son las siguientes:

- Podemos empezar el desarrollo inmediatamente.
- Sea cual sea nuestro entorno, los desarrollos manuales se pueden adaptar a cualquier tipo de situación.
- Mayor flexibilidad a la hora de implementar lógica de negocio compleja.
- Mayor facilidad para encontrar perfiles que conozcan el lenguaje que se ha de utilizar.

Y los **inconvenientes** son estos:

- Hay muchos programas que es necesario construir, y todos tienen una estructura similar: lecturas, filtrados, cruces, uniones, etc., se puede plantear la opción de reutilizar código creando funciones, pero estas también hay que desarrollarlas y mantenerlas.
- Los metadatos asociados a los programas deben desarrollarse de manera explícita, como una tarea adicional, por lo que presentan el mismo problema que la mayoría de los desarrollos manuales. Directamente no se desarrollan los metadatos para reducir el coste o el plazo de ejecución o, si se desarrollan, no se actualizan con los cambios que se van produciendo.
- Los programas requieren bastante tiempo para desarrollarse.
- Los requerimientos cambian constantemente y los programas deben adaptarse a los mismos. Por lo tanto, las modificaciones son muy frecuentes y suelen ser costosas. No existen metadatos que ayuden a realizar el análisis de impacto de un cambio.
- Es complicado realizar un linaje de datos para conocer, partiendo de un dato final, todas las transformaciones realizadas. Es necesario ir al código para obtener la trazabilidad del dato.
- Es más difícil marcar unas buenas prácticas y unos estándares al equipo de desarrollo. Es habitual, como sucede en cualquier otra aplicación, encontrarnos con procesos difíciles de mantener.

- El coste total de desarrollo es muy alto.

Las principales **ventajas** del uso de herramientas son las siguientes:

- Los programas de extracción y transformación se pueden construir rápidamente, sobre todo en aquellos pasos comunes muy extendidos.

- Gracias al hecho de que almacenan las definiciones hechas en forma de metadatos, los programas generados se pueden mantener rápida y fácilmente.

- Los metadatos asociados se producen y se mantienen de manera automática.

- Es posible realizar un linaje de datos para conocer, partiendo de un dato final, todas las transformaciones que ha sufrido basándonos en la información guardada en los metadatos.

- Es posible realizar análisis de impacto para conocer las transformaciones afectadas por un cambio en la información origen.

- Los desarrollos tienen más portabilidad. En caso de cambiar de plataforma de fuente o de destino de los datos, los metadatos que definen las correspondencias y transformaciones continúan siendo válidos. Es suficiente con modificar los parámetros de los pasos de transformación teniendo en cuenta los nuevos parámetros de las plataformas de destino.

- Es más sencillo marcar unas buenas prácticas y unos estándares al equipo de desarrollo. Los desarrollos son más estándares y más mantenibles al quedar guardados en flujos de proceso más sencillos de seguir.

- Los costes de desarrollo se reducen de manera significativa.

Por otro lado, las herramientas del mercado presentan distintos **inconvenientes**:

- Están preparadas para ser utilizadas en entornos muy generales; si nuestro entorno tiene características particulares, como es habitual que suceda, es necesario adaptarlas. En caso de soportarla, esta operación puede ser muy compleja y costosa.

- Es preciso dedicar un periodo a la formación de los desarrolladores que las van a utilizar.

- Presentan limitaciones a la hora de implementar lógica de negocio compleja.

- Su precio es muy alto en el caso de herramientas líderes. Existe la opción de emplear herramientas *open source* que cada vez presentan un grado de madurez mayor. Aun así, su coste se compensa con el esfuerzo de desarrollo que ahorran.

Por lo tanto, para empezar a desarrollar un almacén de datos, la solución que suelen adoptar muchas organizaciones es desarrollar el software del componente de integración y transformación de manera manual (o con el apoyo automático de cualquier otro desarrollo). Esta solución puede resultar adecuada al principio, para construir algún almacén de datos departamental. Sin embargo, en ámbitos más amplios o para diferentes almacenes de datos, los costes de desarrollo y las limitaciones en cuanto al tiempo de respuesta ante cambios se disparan. Por este motivo, es recomendable adquirir una herramienta de apoyo desde el principio, y planificar el tiempo necesario para adaptarla a nuestro entorno. El precio de las licencias es un factor que hay que considerar, pero esta inversión puede suponer un ahorro en plazos y siempre podemos evaluar la opción de software libre, ya que las herramientas de apoyo al ETL de este tipo han evolucionado mucho en los últimos años.

Es conveniente utilizar una herramienta como apoyo al desarrollo del componente de integración y transformación, puesto que se requiere que sea muy flexible y se adapte rápidamente a los cambios producidos en el resto de los componentes de la FIC.

3.5.3. Otras herramientas de apoyo

Además de las herramientas anteriores, hay otras herramientas especializadas para ofrecer soporte en las operaciones de depuración e integración de algunos tipos de datos, como, por ejemplo, las que hacen referencia a nombres y domicilios. Estas herramientas están basadas en diccionarios específicos para cada idioma o entorno en el que almacenan nombres de personas, apellidos, nombres de empresas, nombres de calles, etc., así como distintas maneras de representarlos y abreviarlos o maneras erróneas que a veces se utilizan. Generalmente, usan patrones para reconocer los diferentes tipos de ocurrencia que se producen y de este modo identificar los sinónimos.

3.6. Rendimiento del componente de transformación e integración

Un aspecto importante en el desarrollo del componente de transformación e integración es el relativo al rendimiento de los procesos. Generalmente, este componente tratará un volumen de datos alto y las necesidades de actualiza-

ción y disponibilidad de la información son exigentes, de modo que optimizar la ventana de ejecución de este componente será un aspecto que considerar desde el momento de su diseño.

Algunos aspectos relativos a diseño que nos pueden ayudar a mejorar los tiempos de proceso son los siguientes:

- Filtros de registros y columnas: quedarnos en los primeros pasos con los datos estrictamente necesarios, evitando arrastrar al resto de proceso columnas o registros innecesarios.

- Transformaciones en memoria: realizar la mayor parte de las transformaciones en memoria, si es posible. Evitar procesos de lectura y escritura en disco que son costosos en tiempo.

- Transformaciones en base de datos: en caso de disponer de una base de datos con alta capacidad de proceso y arquitectura optimizada para un rendimiento óptimo, nos podemos plantear llevar los datos a base de datos desde la extracción y realizar las transformaciones en base de datos. Este tipo de arquitectura se denomina ELT (siglas del término en inglés *Extract Load Transform*).

- Uso de objetos de base de datos para optimizar los procesos de extracción: definir índices y mantener estadísticas de base de datos actualizadas sobre las tablas origen sobre las que se ejecuten extracciones pesadas.

- Revisión de operaciones pesadas: algunas operaciones como las agregaciones y ordenaciones son costosas en recursos y tiempo. Minimizar su uso a lo estrictamente necesario.

- Paralelización de procesos: en caso de que las operaciones a realizar sean paralelizables y dispongamos de una arquitectura que favorezca el paralelismo.

- Búsqueda del método óptimo de obtención de actualizaciones (apartado 3.1.2).

4. Construcción de almacén de datos: departamental, corporativo y operacional

4.1. Construcción de almacén de datos corporativo

El almacén de datos corporativo ofrece una visión integrada e historiada de los datos de la organización. Su misión es almacenar los datos para suministrarlos a los distintos almacenes de datos departamentales.

En este apartado, estudiaremos diferentes aspectos relacionados con el desarrollo del almacén de datos corporativo. Repasaremos brevemente lo que ya hemos estudiado relacionado con este punto y profundizaremos en un elemento que aún no hemos estudiado: el modelo de datos del almacén de datos corporativo.

4.1.1. Revisión del proceso de desarrollo

Según la estrategia presentada en este módulo, el almacén de datos corporativo se desarrollará de manera gradual. En un primer momento, mediante el proyecto global de desarrollo tendremos una visión general de los datos que contendrá el almacén de datos corporativo. Por lo tanto, diseñaremos su esquema a alto nivel y definiremos las entidades que aparecerán, sin entrar en el detalle de los atributos concretos. Asimismo, podremos hacer una previsión del volumen de datos que llegará a tener y determinar qué máquina y qué SGBD lo podrán soportar.

Mediante el proyecto de desarrollo de infraestructura obtendremos la máquina e instalaremos y configuraremos la SGBD.

El diseño inicial del esquema del almacén de datos corporativo guiará su desarrollo. Lo refinaremos con el desarrollo de los sucesivos proyectos autónomos en que dividimos la construcción de la FIC. El trabajo hecho en estos proyectos estará supervisado por el arquitecto de la FIC, junto con el administrador de bases de datos responsable del entorno. A medida que vamos poniendo en marcha estos proyectos, el almacén de datos corporativo se irá poblando de datos.

Construiremos el almacén de datos corporativo de manera gradual. Partimos de un diseño inicial que refinaremos con los diferentes proyectos de desarrollo de almacenes de datos departamentales que también lo poblarán de datos.

4.1.2. El modelo de datos del almacén de datos corporativo

Los almacenes de datos departamentales están orientados al acceso de sus datos por parte de los analistas, y por este motivo están diseñados según el modelo de datos multidimensional, que ha sido definido para permitir hacer consultas complejas de manera simple. Aun así, la principal misión del almacén de datos corporativo es almacenar y suministrar los datos necesarios a los almacenes de datos departamentales, es decir, está orientado al almacenamiento de los datos. ¿Según qué modelo de datos lo diseñaremos?

Una posibilidad consiste en utilizar el modelo de datos multidimensional del mismo modo que lo hacemos para los almacenes de datos departamentales. Este modelo se ha pensado de manera exclusiva para hacer consultas centradas en los hechos (los hechos son el foco de atención y el modelo permite almacenar datos históricos sobre estos muy fácilmente); aun así, no presenta esta facilidad para almacenar datos históricos sobre las dimensiones que califican los hechos.

Ejemplo de hechos y dimensiones

En un almacén de datos departamental de ventas, los datos relativos a las transacciones de ventas serán los hechos (eventos de negocio) y los datos de productos, clientes o centro de ventas serán dimensiones (ejes para clasificar los hechos).

Aunque podemos hacer consultas sobre el almacén de datos corporativo, este no es su principal objetivo. Tendremos que utilizar un modelo que sea más adecuado para el almacenamiento de datos, particularmente de datos históricos.

Otra posibilidad es utilizar un modelo conceptual de los que usamos para construir sistemas operacionales: entidad/relación, orientado a objetos, etc. Estos modelos nos permitirán definir las entidades que aparecen y las relaciones que hay entre estas. Con estos modelos, llevaremos a cabo diseños similares a los que hacemos para los sistemas operacionales.

La diferencia sustancial aparecerá a la hora de implementar los esquemas diseñados, es decir, al situarnos en el ámbito lógico. Por ejemplo, si empleamos el modelo relacional, para construir un sistema operacional aplicaríamos las reglas de normalización para obtener un esquema normalizado. Uno de los objetivos de normalizar es evitar tener redundancia de datos. De este modo, si tenemos que modificar los datos correspondientes a una entidad, bastará con hacerlo en un solo lugar (puesto que solo están almacenados en una tabla y referenciados por otros). En el almacén de datos corporativo no haremos

modificaciones sobre los datos almacenados: solo almacenaremos imágenes sucesivas de los datos. Por lo tanto, normalizar para evitar problemas con las modificaciones no tiene razón de ser. De este modo, podremos hacer diseños no normalizados, si se adaptan mejor a la estructura de los datos que hay que almacenar.

Para implementar el almacén de datos corporativo podemos utilizar el modelo relacional. Aun así, no será necesario normalizar el diseño del esquema, puesto que no haremos modificaciones sobre los datos almacenados.

También podremos hacer la implementación utilizando el modelo orientado a objetos; en este caso, la implementación estará más cerca del diseño conceptual realizado. Una de las ventajas principales de las bases de datos orientadas a objetos es que permiten asociar directamente los datos con el código que las trata, particularmente para hacer modificaciones sobre estas. Por otro lado, ofrecen más facilidad que las bases de datos relacionales para implementar esquemas muy complejos. En el almacén de datos corporativo, los esquemas diseñados no serán especialmente complejos. Además, no tendremos operaciones de modificación asociadas a los datos. Así pues, parece que al implementar el almacén de datos corporativo en una base de datos orientada a objetos, no se aprovechan las ventajas que ofrece este tipo de base de datos.

No hay nada en contra de implementar el almacén de datos corporativo sobre una base de datos orientada al objeto. A pesar de ello no podemos aprovechar las ventajas principales que ofrece este tipo de base de datos dadas las características del sistema que debe de construirse.

4.1.3. Transformaciones para construir el esquema del almacén de datos corporativo

En este apartado, estudiaremos el conjunto de transformaciones que aplicaremos a los esquemas de las fuentes de datos operacionales para implementar el esquema del almacén de datos corporativo. Construiremos el esquema en cualquiera de los modelos lógicos mencionados en el apartado anterior.

El objetivo de estas transformaciones es doble:

- Por un lado, necesitamos que el esquema diseñado permita reflejar la evolución producida en los datos y sus relaciones, es decir, que almacene la historia de los datos.

- Por otro, puesto que solo haremos consultas sobre los datos, el objetivo es que el diseño esté optimizado para que estas consuman el menor tiempo

posible. Todo esto teniendo presente que el diseño del esquema puede estar desnormalizado.

Modelo del almacén de datos corporativo

En Silverston, Inmon y Graziano (1997), *The Data Model Resource Book*, la primera tarea que se propone para definir el modelo del almacén de datos corporativo a partir de un modelo de datos corporativo operacional es la de eliminar los datos operacionales que se crea que no serán utilizados por los analistas. Es así porque proponen construir el modelo del almacén de datos corporativo en un solo paso, centrándose en el almacenamiento de los datos. Nosotros lo construiremos en diferentes pasos, uno para cada proyecto autónomo que desarrollemos, y en cada paso definiremos en el almacén de datos corporativo de manera exclusiva los datos que necesiten los analistas del almacén de datos departamental asociado al proyecto autónomo. Por lo tanto, no debemos eliminar datos operacionales de manera explícita, puesto que incluiremos exclusivamente los datos que necesitamos para el proyecto.

Adicción de un elemento de tiempo

Siempre tendremos que añadir al menos un elemento de tiempo a cada unidad de datos en el almacén de datos corporativo. El objetivo principal es expresar de alguna manera el periodo de validez de los datos en el entorno operacional, puesto que este es el tiempo que requieren los analistas para estudiar la evolución de los datos.

Además del periodo de validez, podremos disponer de otros datos relacionados con el tiempo; por ejemplo, el momento de la extracción de los datos, de su carga, de la disponibilidad para el analista, etc.

En algunos casos podremos obtener este tiempo del entorno operacional, puesto que habrá quedado registrado el momento en el que se han producido los cambios (es así en todos los métodos de extracción de datos inmediatos; en el de huella de tiempo también, a pesar de que en este caso podemos haber perdido movimientos). En otros, lo tendremos que añadir con el componente de integración y transformación.

Podemos expresar el periodo de validez de los datos de diferentes maneras, entre las cuales las más frecuentes son las siguientes:

- Utilizando dos campos, uno para indicar el momento de inicio y otro para el momento final.

- Mediante un solo campo que indique el momento de inicio. El momento final de validez corresponderá al momento de inicio de otro valor para los mismos datos, en caso de que lo haya.

- Utilizando un solo campo que indique el periodo de validez (por ejemplo, los datos del mes de enero de este año).

Tendremos que añadir elementos de tiempo a diferentes grados de granularidad de los datos si lo requieren.

- Desde el punto de vista del fichero: todos los datos de un fichero corresponden al periodo de validez indicado. Por ejemplo, almacenamos los datos de cada mes en un fichero diferente.

- Desde el punto de vista del registro: para cada registro de un fichero podemos indicar el periodo de validez como parte de la clave del registro.

- Desde el punto de vista del campo: cada campo de un registro tiene asociados los campos necesarios para expresar el periodo de validez.

Asimismo, los campos que añadiremos para indicar el tiempo tendrán granularidades distintas, según se requiera. En algunos casos bastará con indicar el año, y en otros se tendrá que indicar el tiempo en segundos.

Cada dato del almacén de datos corporativo que lo requiera debe tener asociado un periodo de validez. De este modo, los analistas pueden estudiar su evolución.

Organización de datos según su estabilidad

Debemos tener presente que almacenaremos la evolución de todos los datos de la organización en una base de datos, el almacén de datos corporativo. Si no prestamos especial atención al espacio requerido por las soluciones que proponemos, el tamaño total del resultado se puede disparar. Concretamente, en relación con el apartado anterior, tenemos que prestar especial atención a qué grado de granularidad añadimos a un elemento de tiempo para almacenar la historia.

Estabilidad y grado de granularidad de los datos

En un banco, hemos diseñado la entidad cliente en el almacén de datos corporativo mediante una tabla en la que tenemos los datos personales del cliente (dirección, teléfono, etc.) junto a la suma del saldo total de sus cuentas y dos campos de tiempo que indican el periodo de validez de los datos (granularidad de los datos desde el punto de vista del registro). De este modo, cada vez que cambie alguno de los datos del cliente, tenemos que almacenar un nuevo registro con los nuevos datos. Si el saldo cambia de media una vez al día para cada cliente, cada día almacenaremos una nueva versión de cada registro completo, incluidos los datos personales que cambian una vez cada cinco años de media. Es decir, malgastamos mucho espacio, pues hemos almacenado 1.825 veces (365 × 5) la misma dirección.

Este mismo diseño en un sistema operacional puede presentar otros problemas, aunque no de espacio. En este caso, si un atributo de un registro cambia, se pierde el valor anterior y no se almacena una versión completa del registro.

En el ejemplo anterior, los datos del registro de cliente tienen una estabilidad distinta. Aun así, hemos definido la granularidad de almacenamiento de cambios desde el punto de vista del registro, sin tener este hecho en cuenta. Para optimizar la cantidad de datos almacenados, deberíamos haber considerado, por un lado, el campo saldo y, por otro, los datos personales, puesto que tienen estabilidades distintas. Si implementásemos este ejemplo en una base de datos relacional, deberíamos tener en una tabla los datos personales y en otra

el saldo, y definir en los dos casos la granularidad desde el punto de vista del registro, porque no sabemos el número máximo de cambios que puede tener el saldo y, por este motivo, no podríamos tener la granularidad desde el punto de vista del campo.

Para optimizar el espacio ocupado en el almacén de datos corporativo, dentro de cada entidad tenemos que definir la granularidad de los datos a los que añadimos un elemento de tiempo según su grado de estabilidad ante los cambios.

Adicción de datos derivados

Los modelos operacionales generalmente no incluyen datos derivados; es decir, aquellos obtenidos a partir de otros datos almacenados en la base de datos. No tiene mucho sentido almacenar datos derivados en los casos en que los valores base de los cálculos pueden cambiar con frecuencia, por lo que se tendrían que recalcular cada vez que se produjera un cambio.

Las principales **ventajas** de almacenar datos derivados en la base de datos son las siguientes:

- Se tiene más velocidad de acceso a estos datos, puesto que no es preciso calcularlos cada vez que se necesitan.

- Se evitan errores. Si directamente proporcionamos los datos, evitamos que alguien los calcule de manera errónea.

Los **inconvenientes** son los siguientes:

- Si los datos base cambian, deben calcularse los datos derivados a partir de estos.
- Se ocupa más espacio de almacenamiento.

Por definición, los datos almacenados en el almacén de datos corporativo no cambian. Por este motivo, si incluimos datos derivados, salvo situaciones de error, no se tendrán que recalcular y no tendremos este inconveniente. De este modo, el principal inconveniente de incluirlos es el problema del espacio de almacenamiento que requieren.

Las ventajas que aporta incluir datos derivados a los que acceden los analistas son más elevadas que el coste de almacenamiento, especialmente si calcularlos es complejo y costoso.

Por otro lado, también podemos almacenar estos datos derivados en los diferentes almacenes de datos departamentales, lo cual será el medio de acceso para la mayoría de los usuarios de la FIC.

Calcularemos los datos derivados que los analistas necesiten y los almacenaremos en el almacén de datos corporativo, o bien directamente en los almacenes de datos departamentales.

El concepto denominado por Inmon **índice creativo** está asociado a los datos derivados. El planteamiento es el siguiente: dado que para pasar los datos desde las fuentes de datos operacionales al almacén de datos corporativo tenemos que trabajar con los datos al más bajo nivel, con un poco de trabajo adicional podríamos calcular a partir de estos una serie de datos interesantes para los analistas. Es decir, precalculamos requerimientos de los analistas que podemos anticipar y los almacenamos en el almacén de datos corporativo.

A diferencia de los datos derivados comentados anteriormente, los datos calculados para el índice creativo son aquellos que no se consideran en el entorno de las aplicaciones operacionales: los obtenidos de manera exclusiva para los analistas.

Ejemplo de índice creativo

Por ejemplo, en un banco, las cuentas que menos actividad tienen, los ingresos más altos, etc.

Cambio granularidad en los datos

A la hora de diseñar el almacén de datos corporativo, podemos tener tendencia a incluir todos los datos disponibles en el nivel de detalle más bajo que podamos obtener de las fuentes de datos operacionales. De este modo, intentamos prever necesidades futuras que pueden ser muy improbables. El resultado es que el espacio requerido se puede disparar.

El almacén de datos corporativo debe cubrir las necesidades actuales de los usuarios. Antes de añadir cualquier dato que no se necesite, debemos valorar las repercusiones de coste que esto puede tener.

Concretamente, en cuanto a la granularidad, debemos adaptar el nivel de detalle de los datos que obtenemos de las fuentes de datos operacionales al requerido por los usuarios.

Adaptación de la granularidad de los datos de los clientes

En el ejemplo del banco, para cada cliente los analistas requieren estudiar la evolución de su saldo mensual. Particularmente, utilizan para cada mes los valores del saldo máximo, mínimo y medio. A partir del sistema operacional, podemos obtener todos los valores que toma el saldo a lo largo de cada día. Si los analistas conocen la posibilidad de tener los datos con un nivel más bajo, pero consideran que solo necesitan los datos de ámbito mensual para almacenar el saldo de cada cliente, obtendríamos los valores requeridos por

los analistas agregando todos los valores obtenidos para cada mes, y serían exclusivamente estos valores calculados los que almacenaríamos en el almacén de datos corporativo.

En todo momento debemos tener presente que el nivel de granularidad definido va a suponer una restricción en el análisis de datos, en el sentido de que por debajo de este nivel de granularidad no vamos a poder analizar datos y que añadir este nivel más bajo *a posteriori* es muy costoso. Teniendo esto presente, llegaremos a un nivel que satisfaga las necesidades de los analistas, pero que no tiene por qué ser el mismo que el de los sistemas operacionales.

Con el objetivo de minimizar el espacio ocupado, el almacén de datos corporativo tiene que almacenar los datos con la granularidad que necesiten los analistas, no con la granularidad más baja que podemos conseguir de las fuentes de datos operacionales.

Fusión de entidades

Esta fusión se puede hacer por diferentes motivos:

- Las entidades provienen de diferentes sistemas operacionales que se integran en una entidad en el almacén de datos corporativo, puesto que conceptualmente son la misma.

- En los sistemas operacionales, tenemos los datos organizados para que estos soporten de manera óptima las operaciones de modificación, generalmente dividiéndolos en entidades más pequeñas. Al pasar al almacén de datos corporativo podemos agrupar estas entidades, puesto que no tendremos problemas con las modificaciones. Concretamente, dado que no se producen cambios en los datos, sabemos el número de ocurrencias que tenemos y así podemos definir atributos para contener múltiples ocurrencias. De este modo, se facilita el acceso a los datos.

- Puede resultar más fácil representar la evolución de las relaciones entre dos entidades almacenándolas juntas y definir una sola entidad a partir de estas. En este caso, fusionar entidades puede representar replicar datos.

En el almacén de datos corporativo podemos tener los datos desnormalizados y replicados, puesto que no se producirán modificaciones. Mediante la fusión de entidades operacionales se facilita el acceso a los datos, dado que se puede acceder directamente a los mismos en lugar de tener que seguir las relaciones definidas entre los datos para obtenerlos. En caso de replicar datos, tenemos que evaluar el coste que representa en espacio adicional requerido.

4.2. Construcción de almacén de datos departamental

La construcción de los almacenes de datos departamentales difiere notablemente de la construcción de un almacén de datos corporativo, dado que los almacenes departamentales cubren las necesidades de un grupo de analistas, no de toda la organización. Por otra parte los almacenes departamentales no se alimentan directamente de las fuentes de datos, sino del almacén de datos corporativo. Del conjunto de fuentes de datos disponibles solo se utilizarán las que sean de interés para el grupo de analistas del departamento.

Los dos tipos de almacenes gestionan volúmenes de datos de orden de magnitud diferentes. Un almacén de datos departamental contiene un conjunto de datos relacionados con un tema o visión concreta de la organización (departamento). Además, por el tipo de análisis que se realizan, no es habitual tener la necesidad de cargar el máximo nivel de detalle de los datos. Podemos entender que el almacén de datos departamental contiene un subconjunto agregado de los datos disponibles en el almacén de datos corporativos. Por el contrario, el almacén de datos corporativo contiene una versión consolidada de los datos de todos los almacenes de datos departamentales. Muchas veces se desconoce a priori el tipo de análisis que se realizará en el futuro, por lo que es habitual cargar un mayor nivel de detalle, para cubrir futuras necesidades de análisis.

4.2.1. Diseño del modelo y aprovisionamiento de datos

El diseño de modelo de datos sobre el que crearemos el almacén de datos departamental será desnormalizado, ya que su objetivo primordial es la consulta, no la modificación.

Ved también

El tipo de diseño se basará en el modelo dimensional que se verá en detalle en el módulo «Diseño e implementación multidimensional de datos» de esta asignatura.

En lo relativo al aprovisionamiento de datos los procesos que carguen el almacén de datos departamental serán más sencillos que el componente de transformación e integración del almacén de dato corporativo, dado que en este caso la carga de datos se realiza desde el almacén de datos corporativo donde los datos ya están depurados e integrados.

En el proceso de carga desde el almacén corporativo al departamental el tipo de pasos que hay que realizar será del tipo:

- Selección de columnas y registros de interés para el almacén departamental.
- Agregaciones desde los datos del almacén corporativo buscando el nivel de granularidad adecuado; generalmente, el nivel de detalle del almacén corporativo será mayor.
- Transformaciones necesarias para adecuar los datos al esquema multidimensional del almacén departamental.

4.2.2. Enfoque del proyecto

Desde el punto de vista de la ejecución del proyecto, la envergadura del proyecto de creación de un almacén de datos departamental será siempre inferior a un almacén de datos corporativo y su plazo de ejecución, mucho menor.

En caso de plantear la construcción de un almacén de datos departamental sin existir previamente el almacén de datos corporativo, será buena práctica acometerlo como proyecto autónomo dentro de la FIC.

Ved también

El proyecto autónomo dentro de la FIC se ha tratado en el apartado 2.4.3 de este módulo.

Otro posible enfoque en la creación del almacén de datos departamental es la creación de un *virtual data mart*, que consiste en una capa lógica sobre el almacén de datos corporativo que ofrece una visión parcial de los datos necesarios para los usuarios del almacén departamental. Este enfoque evita el movimiento de datos, se realiza en menos tiempo, pero tiene limitaciones cuando tratamos con volúmenes de datos altos, ya que hay muchos datos calculados «al vuelo» y en ocasiones las consultas pueden ser lentas.

4.3. Construcción del almacén de datos operacional

De manera tradicional, el entorno operacional se ha estructurado en forma de aplicaciones independientes. Para adaptarse a los requerimientos del entorno, las organizaciones necesitan tener una visión integrada de sus datos.

Aplicaciones no integradas en un banco

Un banco dispone de aplicaciones independientes para gestionar las tarjetas de crédito, créditos hipotecarios, créditos personales, cuentas de ahorro, etc. Cada aplicación funciona en su plataforma, aunque los trabajadores pueden acceder a la misma desde su terminal. Si un usuario necesita conocer toda la información asociada a un cliente (por ejemplo, el director de una oficina o el empleado de atención telefónica a los clientes) tiene que acceder a cada una de las aplicaciones para hacer la consulta correspondiente. Si las aplicaciones estuvieran integradas, bastaría con una sola consulta para obtener los datos personales y comerciales del cliente.

El almacén de datos operacional ofrece una visión integrada de los datos operacionales de la organización; a diferencia del almacén de datos corporativo, no almacena la historia de los datos, es volátil y está actualizado. Su misión principal consiste en ofrecer apoyo operacional a la organización. Lo utilizan principalmente usuarios operacionales (oficinistas), aunque también lo hacen los analistas.

Ejemplo de aplicación de los almacenes de datos operacional y corporativo

Desde un punto de vista de las ventas de una empresa si un analista realiza un análisis histórico de las ventas por producto accederá al almacén de datos corporativo o departamental, mientras que un usuario más ligado a la operativa diaria que necesita saber el stock de determinado producto en el momento actual obtendrá este dato en el almacén de datos operacional.

En este apartado, estudiaremos diferentes tipos de almacenes de datos operacionales, lo que aporta este concepto a la organización y también alternativas que tiene la organización para disponer de los mismos.

4.3.1. Paquetes de aplicaciones y el almacén de datos operacional

En el mercado, encontramos paquetes de aplicaciones que ofrecen una visión integrada del entorno operacional. Dado que la mayoría de las organizaciones en un sector tienen necesidades similares, la idea general de estos productos es ofrecer un conjunto de aplicaciones integradas que cubran las necesidades generales de una organización prototipo dentro de un determinado sector. En muchos casos, las aplicaciones se pueden adaptar a las características particulares del cliente que las adquiere, generalmente a un coste alto. Con frecuencia, son las organizaciones las que se tienen que adaptar a la manera de trabajar impuesta por las aplicaciones.

Algunas organizaciones intentan «liberarse» de sus antiguas aplicaciones mediante la adquisición de paquetes de aplicaciones integradas. Aun así, es frecuente que organizaciones que adquieren uno de estos paquetes deban conservar parte de sus aplicaciones tradicionales porque las nuevas no cubren totalmente sus necesidades, o bien se ha decidido adquirir solo una parte del paquete de aplicaciones. En estos casos, tienen que convivir los dos entornos y generalmente las aplicaciones del paquete adquirido han de obtener datos de las aplicaciones que había previamente.

¿Qué relación hay entre el almacén de datos operacional y los paquetes de aplicaciones integradas?

Los dos ofrecen una visión integrada de los datos del entorno operacional. Las características de estos también son idénticas para los dos y en los dos casos el objetivo principal es ofrecer apoyo operacional a la organización.

> Los paquetes de aplicaciones integradas cuyo ámbito es toda la organización son implementaciones comerciales del almacén de datos operacional.

Algunos de los paquetes comerciales que hay en el mercado están diseñados teniendo en cuenta la arquitectura de la FIC y ofrecen medios para integrarse en la misma, particularmente para obtener los datos que van a construir el almacén de datos corporativo. En otros casos, van más lejos y ofrecen como ampliación del paquete básico una versión genérica de la FIC para las organizaciones del sector. Aun así, en lo que respecta a la toma de decisiones de los

analistas, ofrecer un apoyo estándar que sea adecuado para todos los analistas es mucho más complejo que lo que se refiere al apoyo al trabajo más operacional de la organización, ya que las necesidades son más diversas.

Si construimos el almacén de datos operacional, una vez diseñada la base de datos que lo soporta y los métodos de obtención y refresco de sus datos a partir de las aplicaciones operacionales, construiremos aplicaciones directamente sobre este. En algunos casos, las nuevas aplicaciones cubrirán necesidades nuevas, y en otros sustituirán aplicaciones antiguas como ocurre con los paquetes de aplicaciones integradas.

El almacén de datos operacional puede ser adquirido en forma de paquete de aplicaciones integradas o construido en la organización.

4.3.2. Velocidad de refresco de los datos

En el caso de que conviva el almacén de datos operacional con las aplicaciones operacionales, caso frecuente en las organizaciones, este tiene que obtener parte de sus datos a partir de los de las aplicaciones. Inmon distingue cuatro clases de almacenes de datos operacionales según la velocidad de refresco de los datos:

- Clase I: se actualiza unos segundos después de que se produzcan las modificaciones en las aplicaciones operacionales.

- Clase II: se actualiza unas horas después de que se produzcan las modificaciones.

- Clase III: el periodo de actualización es superior a las veinticuatro horas.

- Clase IV: se actualiza de manera no planificada.

En los almacenes de datos operacionales de clase I, no se pueden acumular los movimientos obtenidos en las fuentes de datos operacionales para aplicarlos de manera masiva. Se tiene que disponer de un medio de trasladar las modificaciones directamente (mediante disparadores, RPC u otros medios). Generalmente, no se aplican demasiadas transformaciones a los datos en los almacenes de esta clase.

Para actualizar los almacenes de datos operacionales de clase II, III y IV podemos aplicar los métodos de obtención de actualizaciones a partir de las fuentes de datos operacionales estudiadas en este módulo. Para los de clase II, en la mayoría de los casos necesitaremos un método de captura inmediata. Para los de clase III y IV, podremos utilizar cualquiera de los métodos estudiados;

además, puesto que no guardamos la historia de los datos, no hay ningún problema para utilizar un método de captura retardada aunque perdamos detalle de los movimientos.

Cuanto más restrictivas sean las condiciones de actualización, más costará implementarlas. Por este motivo, un almacén de datos de clase I será mucho más caro que uno de clase IV.

4.3.3. Planificación de incorporación del almacén de datos operacional

Plantear la adquisición o la construcción del almacén de datos operacional como un solo proyecto presenta el grave inconveniente de la dificultad de justificar su coste ante la organización.

En situaciones normales, es más adecuado hacer un desarrollo iterativo similar al planteado para desarrollar el resto de la FIC. Frecuentemente, se puede plantear el almacén de datos operacional como una estructura de apoyo para actualizar los datos del almacén de datos corporativo. En estos casos se construirá, como parte de los proyectos atómicos en los que se ha dividido la construcción de la FIC, la parte correspondiente al almacenamiento de los datos, de manera conjunta con el almacén de datos corporativo. Asimismo, dentro de la construcción de la FIC, podemos plantear proyectos de desarrollo o de ampliación del almacén de datos operacional, con las mismas premisas con las que definimos el resto de los proyectos.

Si se planifica su construcción antes de disponer del almacén de datos corporativo, puede servir para construirlo. Aun así, se corre el riesgo de pretender incorporar al almacén de datos operacional funcionalidades propias del almacén de datos corporativo.

Dado que el coste resulta difícilmente justificable, es recomendable adquirir o desarrollar el almacén de datos operacional de manera iterativa, mediante proyectos autónomos.

Resumen

Conocer la arquitectura de la FIC representa un avance para las organizaciones que quieren ofrecer herramientas de apoyo a la toma de decisiones para los analistas. Aunque se pueden considerar arquitecturas alternativas, estas presentan graves deficiencias que las hacen inviables en la mayoría de las situaciones.

La arquitectura de la FIC se tiene que trazar desde un nivel alto, teniendo en cuenta la perspectiva de toda la empresa. Aun así, es un error plantear el desarrollo de la FIC como un solo proyecto: este se tiene que hacer de manera iterativa, por medio de proyectos independientes con objetivos y beneficios claros. Es decir, los sucesivos proyectos de desarrollo de la FIC hacen que su funcionalidad aumente con el tiempo y, con esta, el beneficio que aportan a la organización.

En este módulo, además de estudiar diferentes alternativas en la arquitectura de la FIC, así como distintas maneras de planificar su construcción, hemos estudiado aspectos concretos de la construcción de sus componentes. Particularmente, hemos prestado especial atención al componente de integración y transformación por su dificultad; asimismo, hemos estudiado en detalle las peculiaridades de la construcción del almacén de datos corporativo y del almacén de datos operacional.

Ejercicios de autoevaluación

1. ¿Podemos adquirir la FIC?

2. ¿En qué consiste el entorno operacional de telaraña?

3. ¿Podemos construir el almacén corporativo después de haber construido distintos almacenes de datos departamentales?

4. ¿Es adecuado combinar el almacén de datos operacional y el corporativo en una sola estructura?

5. ¿Tiene que incluir la FIC todos los componentes estudiados en el módulo«"La factoría de información corporativa»?

6. ¿Podemos construir la FIC con un solo proyecto?

7. ¿Qué características deben tener los proyectos de construcción de la FIC?

8. ¿Qué estructura deben tener los proyectos de construcción de la FIC?

9. ¿Según qué metodología desarrollaremos los proyectos de construcción de la FIC?

10. ¿Qué ocurre con el entorno operacional cuando desarrollamos la FIC?

11. ¿Quién es el patrocinador de la FIC?

12. ¿Cuál es el principal inconveniente de los métodos de captura retardada para obtener las modificaciones producidas en los datos de las fuentes de datos operacionales?

13. ¿Qué características tiene el modelo de datos del almacén de datos corporativo?

14. ¿Qué relación hay entre el almacén de datos operacional y los paquetes de aplicaciones integradas?

15. ¿Qué modelo de datos tiene el almacén de datos operacional?

16. Desde el punto de vista de los metadatos del componente de transformación e integración, qué opción es más óptima: ¿desarrollar el componente con código manual o utilizar una herramienta de apoyo?

Solucionario

Ejercicios de autoevaluación

1. Generalmente, no. La FIC se tiene que construir en cada organización. Algunos paquetes de aplicaciones tienen como ámbito toda la organización, incluyen funcionalidades de la FIC y están pensados para cubrir las necesidades de una organización estándar. Si el sistema de información de la organización está implementado mediante un paquete estándar que incluye la FIC, en este caso la FIC habría sido adquirida. Esta es una situación límite; generalmente, la FIC se tiene que construir a medida para las diferentes organizaciones.

2. Es el resultado de la evolución incontrolada del entorno operacional. Mediante la construcción de programas de extracción de información y bases de datos temporales para elaborar informes puntuales, llegamos a un entorno de alta complejidad con una estructura de relaciones entre los diferentes componentes que se asemeja a una telaraña.

3. La construcción de almacenes de datos departamentales debe de estar basada en el almacén de datos corporativo. Si intentamos construir el almacén de datos corporativo a partir de los diferentes almacenes de datos departamentales que había previamente, la integración de los datos de estos será muy compleja y seguramente también requerirá la modificación posterior de los almacenes de datos departamentales.

4. Aunque tienen algunas características similares, sus objetivos son distintos y el tipo de operaciones a las que ofrecen soporte son incompatibles. El almacén de datos operacional tiene que estar configurado para hacer operaciones de modificación. Aun así, el almacén de datos corporativo necesita estar optimizado para hacer de manera exclusiva operaciones de consulta.

5. No necesariamente. El almacén de datos operacional es una estructura opcional, aunque conviene incluirla por la funcionalidad que aporta. El resto de los componentes sí son necesarios.

6. Si planteamos la construcción de la FIC mediante un solo proyecto, este será demasiado complejo y tendremos dificultades para justificar su coste. Por lo tanto, esta estrategia de construcción no será adecuada.

7. El primer proyecto debe tener como objetivo planificar la construcción de la FIC. Posteriormente, tendremos proyectos de desarrollo de infraestructura. Dividiremos la construcción de la FIC en proyectos autónomos que aporten un valor claro a la organización, que tengan un responsable dentro de esta y se desarrollen en un plazo razonable.

8. Los proyectos han de ser completos, es decir, tienen que prever la obtención y el almacenamiento de los datos y el acceso a los mismos.

9. La metodología de desarrollo más adecuada es una metodología iterativa (CLDS), puesto que generalmente no se conocen con todo detalle los requerimientos de los analistas de información y esta metodología nos permite descubrirlos mediante refinamientos sucesivos.

10. Cuando planificamos la construcción de la FIC, también tenemos que planificar el desmantelamiento del entorno operacional en telaraña, de modo que a medida que construimos los proyectos que implementan la FIC también eliminamos los programas de extracción de datos y las bases de datos temporales que se dejan de utilizar.

11. Es una figura política. Suele ser un directivo de alto nivel de la organización que está convencido de los beneficios que puede aportar la FIC y su misión es obtener los recursos para los proyectos cuyo beneficio no es directamente justificable y tratar de solucionar la oposición que pueda surgir en la organización a la construcción de la FIC.

12. El problema principal que presentan estos métodos es que pueden perder el detalle de los movimientos producidos. Mediante estos métodos, obtenemos un resumen de todas las operaciones hechas en una única operación de modificación.

13. Se trata de un modelo de datos orientado a almacenarlos. Los esquemas desde el punto de vista lógico están diseñados de modo que el almacenamiento y las consultas se hagan de manera óptima.

14. Los paquetes de aplicaciones integradas cuyo ámbito es toda la organización son implementaciones comerciales del almacén de datos operacional.

15. El almacén de datos operacional está construido utilizando un modelo de datos como el de las aplicaciones operacionales. Los esquemas desde el punto de vista lógico están diseñados de modo que las operaciones de modificación se hagan de manera óptima.

16. Es más óptimo utilizar una herramienta de apoyo que nos va a generar la metadata de forma automática.

Glosario

ASCII Siglas en inglés de *American Standard Code for Information Interchange* (código estándar estadounidense para el intercambio de información), es un código de caracteres basado en el alfabeto latino.

CASE Siglas en inglés de *Computer aided system engineering*.

CLDS Metodología iterativa de desarrollo de proyectos. Corresponde a las siglas de SDLC puestas a la inversa, como contraposición a los planteamientos de esta metodología.

Data Warehouse Appliances Plataforma de hardware y software orientada a *datawarehousing* y procesos analíticos.

EBCDIC Siglas en inglés de *Extended Binary Coded Decimal Interchange Code*, es un código estándar de 8 bits usado por computadoras mainframe IBM.

Enterprise datawarehouse bus matrix Arquitectura creada por Kimball que propone una construcción basada en almacenes departamentales interconectados

factoría de información corporativa *f* Conjunto de elementos de software y hardware que ayudan a analizar datos para tomar decisiones.

Extract Transform Load Término en inglés para denominar a los procesos de extracción, transformación y carga que alimentan los almacenes de datos.

ETL Véase *Extract Transform Load*.

índice creativo *m* Conjunto de datos interesantes para los analistas calculados en el momento de pasar los datos de las fuentes de datos operacionales hasta el almacén de datos corporativo.

main frame Ordenador central o corporativo.

massively parallel processors Arquitectura de computación en la que distintas plataformas compuestas por procesador y memoria se interconectan mediante una línea de alta velocidad.

MPP Véase *massively parallel processors*.

RPC *Remote Procedure Call*.

SDLC Metodología de desarrollo de proyectos según un ciclo de vida en cascada en *systems development life cycle*.

SGML Siglas de *Standard Generalized Markup Language* (lenguaje de marcado generalizado estándar). Sistema para la organización y etiquetado de documentos.

sistema de gestión de bases de datos *m* Software que gestiona y controla bases de datos. Sus principales funciones son las de facilitar el uso de las bases de datos de manera simultánea a muchos usuarios de tipos distintos, independizar al usuario del mundo físico y mantener la integridad de los datos. Sigla: SGBD.

sistema decisional *m* Aquel que apoya los procesos de toma de decisiones por parte de los analistas en la organización.

sistema informacional *m* Véase sistema decisional.

sistema operacional *m* Sistema que ayuda en las operaciones diarias del negocio de una organización.

symmetric multi processing Arquitectura de computación en la que un conjunto de procesadores comparte memoria común como medio de comunicación entre estos.

SMP Véase *symmetric multi processing*.

XML Siglas en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), es un lenguaje de marcas utilizado para almacenar datos en forma interpretable.

Bibliografía

Devlin, B. (1997). *Data Warehouse from Architecture to Implementation*. Reading, Mass.: Addison Wesley Longman, Inc.

Inmon, W. H. (1996). *Building the Data Warehouse* (2.ª ed.). Nueva York: John Wiley & Sons, Inc.

Inmon, W. H. (1999). *Building the Operational Data Store*. Nueva York: John Wiley & Sons, Inc.

Inmon, W. H. (2005). *Building the Data Warehouse* (4.ª ed.). Nueva York: John Wiley & Sons, Inc.

Inmon, W. H.; Imhoff, C.; Sousa, R. (1998). *Corporate Information Factory*. Nueva York: John Wiley & Sons, Inc.

Inmon, W. H.; Strauss, D.; Neushloss, G. (2010). *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. Burlington, Mass.: Morgan Kaufman Series in Data Management Systems).

Inmon, W. H.; Welch, J. D.; Glassey, K. L. (1997). *Managing the Data Warehouse*. Nueva York: John Wiley & Sons, Inc.

Jarque, M.; Lenzerini, M.; Vassiliou, Y.; Vassiliadis, P. (2000). *Fundamentals of Data Warehouses*. Berlín: Springer Verlag.

Kelly, S. (1997). *Data Warehousing in Action*. Nueva York: John Wiley & Sons, Inc.

Kimball, R. (2002). *The Data warehouse toolkit: the complete guide to dimensional modeling*. Nueva York: John Wiley & Sons, Inc.

Kimball, R. (2009). *Data Warehouse Toolkit Classics: The Data Warehouse Toolkit* (2.ª ed.); *The Data Warehouse Lifecycle Toolkit* (2.ª ed.); *The Data Warehouse ETL Toolkit*. Hoboken: John Wiley & Sons.

Mattison, R. (1996). *Data Warehousing: Strategies, Technologies and Techniques*. Computing McGraw-Hill.

Silverston, L.; Inmon, W. H.; Graziano, K. (1997). *The Data Model Resource Book*. Nueva York: John Wiley & Sons, Inc.