

# Administración de los sistemas de *data warehouse*

Carles Llorach Rius

PID\_00236087



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

<b>Introducción.....</b>	<b>5</b>
<b>Objetivos.....</b>	<b>6</b>
<b>1. Desarrollo del sistema de <i>data warehouse</i>.....</b>	<b>7</b>
1.1. Diseño del sistema de <i>data warehouse</i> .....	7
1.1.1. Infraestructura .....	8
1.1.2. Plan de implantación .....	14
1.2. Puesta en marcha .....	15
<b>2. Administración del sistema.....</b>	<b>17</b>
2.1. Monitorización del funcionamiento del sistema .....	20
2.2. Mantenimiento del sistema .....	21
2.3. Optimización del rendimiento y del sistema .....	24
2.4. Crecimiento .....	26
<b>3. Roles en la administración del sistema.....</b>	<b>28</b>
<b>Glosario.....</b>	<b>33</b>
<b>Bibliografía.....</b>	<b>35</b>



## Introducción

De la misma forma que el almacén de datos es un factor crítico para el éxito de un proyecto de *business intelligence*, también lo son su creación y posterior administración.

Actualmente, las organizaciones toman conciencia de que implementar un almacén de datos requiere un considerable equipo de desarrolladores, hardware, software, tiempo y dinero.

Las necesidades de diferentes áreas organizativas, a veces contrapuestas, pueden ser sobrellevadas en su conjunto, haciendo compatibles los cambios externos, los mantenimientos y las demandas de las distintas áreas.

Será necesario gestionar el ciclo de vida del *data warehouse*, desde las primeras etapas de planificación y diseño; pasando por su puesta en marcha y crecimiento; llegando hasta las fases de monitorización, mantenimiento y optimización.

En todas ellas, se deberán contemplar siempre dos perspectivas: por una parte, aquello que hacemos y, por otra, los elementos sobre los que actuamos. Surgen así figuras concretas encargadas de llevar a cabo conjuntos de tareas especializadas, que las organizaciones deben formalizar para asegurar su ejecución y, a la vez, el buen desempeño del almacén de datos.

## Objetivos

Los contenidos incluidos en este módulo se orientan a conseguir que el estudiante alcance los objetivos siguientes:

1. Comprender la importancia de una adecuada planificación y diseño de un proyecto de almacén de datos, antes de su desarrollo y puesta en funcionamiento.
2. Tomar conciencia de las tareas que acompañan el normal funcionamiento del almacén de datos y que hacen necesaria la existencia de personas dedicadas a su administración.
3. Establecer un plan ordenado de expansión/crecimiento de nuestro *data warehouse* para poder conseguir su máxima adopción en la organización, asegurando que siga plenamente operativo, aunque se vaya incrementado su nivel de sofisticación.
4. Identificar los principales perfiles profesionales que aparecen vinculados al desarrollo y la administración de los sistemas BI, así como sus principales responsabilidades.

## 1. Desarrollo del sistema de *data warehouse*

Como todo proyecto o sistema informático, nace de una necesidad de la organización y deberá pasar por diferentes fases consecutivas: planificación; desarrollo y despliegue; y finalmente, su posterior evolución y administración.

En el módulo «Evaluación de sistemas de *data warehouse*» de esta asignatura, encontramos las principales actividades que debemos llevar a cabo previamente en la fase de desarrollo del proyecto: planificación, conceptualización y evaluación. Podríamos decir que el almacén de datos empieza a existir a partir del momento en que el proyecto supera estas primeras etapas y es aprobado por la dirección.

En la figura 1, podemos observar el esquema-resumen que presenta Ralph Kimball como propuesta de las etapas del ciclo de vida de proyectos de *data warehouse*.

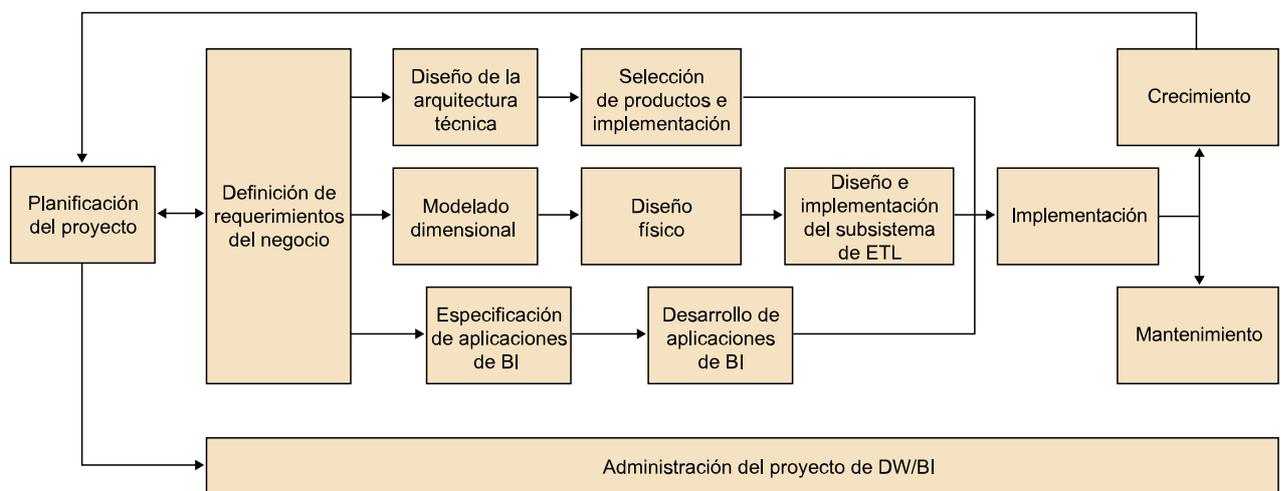


Figura 1. Ciclo de vida Kimball, para *data warehouse*  
Fuente: Kimball Group. *The Data Warehouse Lifecycle Toolkit*.

A lo largo de este módulo, iremos detallando el contenido y alcance de las etapas propuestas, haciendo hincapié en las actividades concretas que se espera que se lleven a cabo en ellas.

### 1.1. Diseño del sistema de *data warehouse*

Cuando hablamos de diseño de un almacén de datos, normalmente estamos pensando en lo relativo a los **datos** que vamos a gestionar: modelo dimensional, modelo físico, ETL, etc. Este enfoque se aborda ampliamente en la asignatura: diseño y construcción del almacén de datos.

En esta asignatura, y concretamente en este módulo, queremos complementar la fase de diseño desde una perspectiva más amplia, que además de los datos, incluye la **tecnología** y las **aplicaciones** que los van a soportar. Nos estamos refiriendo a temas como la arquitectura técnica, selección de productos, instalación de aplicaciones, etc.

En la figura 2, se observa cómo las tareas propias de la fase de diseño (ubicadas en la parte central) se agrupan en tres bloques (cada uno de un color) que se corresponden con los tres ejes de la perspectiva que hemos descrito anteriormente. Estas tres líneas de actuación se deben gestionar de manera específica, sin cometer el error de intentar cubrir todo el alcance de esta fase desarrollando una sola de ellas.

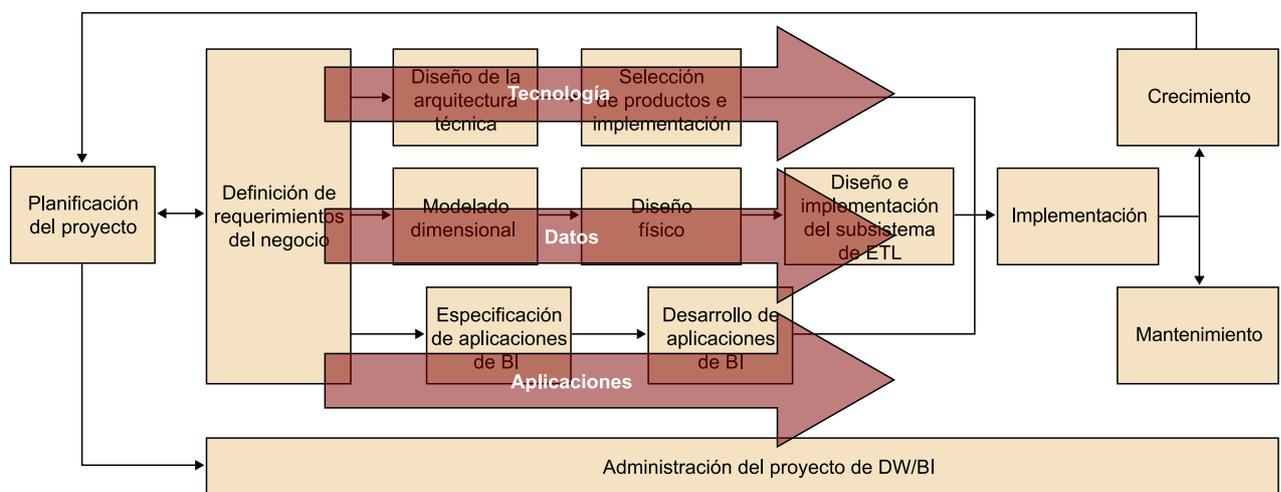


Figura 2. Líneas de actuación de la fase central del ciclo de vida para *data warehouse*

Se presentan a continuación los principales elementos sobre los que vamos a actuar (infraestructura), así como aquellas acciones que se van a llevar a cabo durante las fases de diseño e implementación (plan de acción).

### 1.1.1. Infraestructura

Para cubrir de manera adecuada la parte de **tecnología** del proyecto, se deberá establecer la infraestructura tecnológica que vamos a adoptar. Los elementos mínimos que debemos contemplar son:

1) **Arquitectura:** se adoptará uno de los tres principales enfoques existentes para la arquitectura corporativa de un *data warehouse*:

- *Enterprise bus architecture* (o *data warehouse* virtual), consistente en una arquitectura basada en un conjunto de *data mart* independientes y federados.

- *Corporate information factory* (o FIC), que consiste en una arquitectura en la que existe un *data warehouse* corporativo y un conjunto de *data mart* dependientes de él.
- *Enterprise data warehouse 2.0* (de Bill Inmon), que separa la información (tanto estructurada como no estructurada) por la edad de la misma y la clasifica por su uso.

También debemos tener en consideración una propuesta de arquitectura *data lake* y su vinculación con el almacén de datos, actuando este último como una fuente de datos más.

2) **Plataforma hardware:** a la hora de evaluar un hardware u otro, se tendrán en consideración dos características principales:

- Por un lado, que a este tipo de sistemas suelen acceder pocos usuarios con unas necesidades muy grandes de información, a diferencia de los sistemas operacionales, con muchos usuarios y necesidades puntuales de información. Debido a la flexibilidad requerida a la hora de hacer consultas complejas e imprevistas, y al gran tamaño de información manejada, son necesarias unas altas prestaciones de la máquina.
- Por otro lado, debido a que estos sistemas suelen comenzar con una funcionalidad limitada, que se va expandiendo con el tiempo, es necesario que los sistemas sean escalables para dar apoyo a las necesidades crecientes de equipamiento.

En el mercado, se han desarrollado tecnologías basadas en el procesamiento paralelo que dan el apoyo preciso a las necesidades de altas prestaciones y escalabilidad de los *data warehouse*. Estas tecnologías son de dos tipos: SMP (*symmetric multiprocessing*) y MPP (*massively parallel processing*).

En este apartado no podemos descartar como opción un *data warehouse appliance* (DWA), que incluye un conjunto integrado de servidores, almacenamiento, sistema operativo y SGBD.

#### ***Data warehouse appliance***

Término acuñado por Foster Hinshaw para definir los dispositivos de almacenamiento de datos con una nueva arquitectura orientada específicamente al análisis de grandes volúmenes de datos y al descubrimiento de patrones ocultos.

3) **Sistema gestor de base de datos (SGBD):** de manera independiente de que la información almacenada en el *data warehouse* se pueda analizar mediante visualización multidimensional, el SGBD puede hacerse utilizando tecnología de bases de datos relacionales o multidimensionales.

Las bases de datos relacionales se han popularizado en los sistemas operacionales, pero no siempre cubren las necesidades de información de los entornos *data warehouse*. Aparecen así las bases de datos postrelacionales, que parten de una tecnología consolidada, aunque están implantadas en una arquitectura diseñada de manera más eficiente. Sin olvidar las emergentes bases de datos NoSQL.

Para la correcta elección de una BD NoSQL, tomaremos en consideración básicamente:

- ¿Cómo es el esquema de la BD? Por ejemplo, el esquema de los datos es variable e incluso inexistente. En estos casos, nos podemos decantar por BD basadas en modelo de agregación (clave-valor, documentos, orientada a columnas, dependiendo del caso). En el caso de que sobre nuestro esquema de datos y lo que predomine sean las relaciones entre los conceptos que deseamos representar, quizá la mejor opción sea una BD basada en un modelo en grafo.

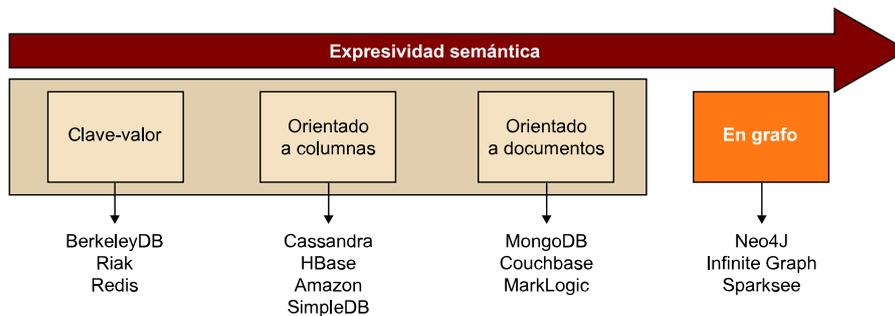
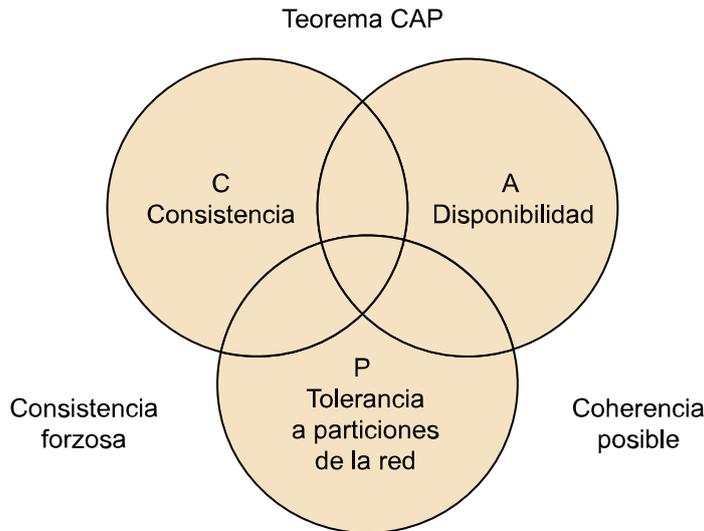


Figura 3. Modelos de datos en NoSQL  
Fuente: UOC.

- ¿Necesitan estar siempre en línea? Trabajan en general en entornos altamente distribuidos que gestionan una gran cantidad de datos, y suelen crecer a medida de las necesidades (escalabilidad horizontal). En este sentido, hay que tener en cuenta los requisitos de consistencia que requiere el entorno de aplicación y el de disponibilidad. Podemos usar el teorema CAP, que se basa en los tres principios que observamos en la siguiente figura:



- Desarrollado en 2000 por Eric Brewer
- Confirmado formalmente en 2002 por Seth Gilbert y Nancy Lynch

Figura 4. Teorema CAP

La restricción añadida para el uso de este teorema consiste en que solo se pueden cumplir de manera simultánea dos de los tres criterios siguientes:

- Consistencia: todos los nodos ven los mismos datos al mismo tiempo.
- Disponibilidad (*availability*): garantiza que cada solicitud recibe una respuesta, ya sea exitosa o fallida.
- Tolerancia a las particiones: el sistema sigue funcionando a pesar de la pérdida arbitraria de mensajes o la caída de parte del sistema.

Podemos ver en el siguiente gráfico algunas de las herramientas disponibles, según los tipos de datos que vamos a gestionar y los principios que exigimos a nuestro SGBD:

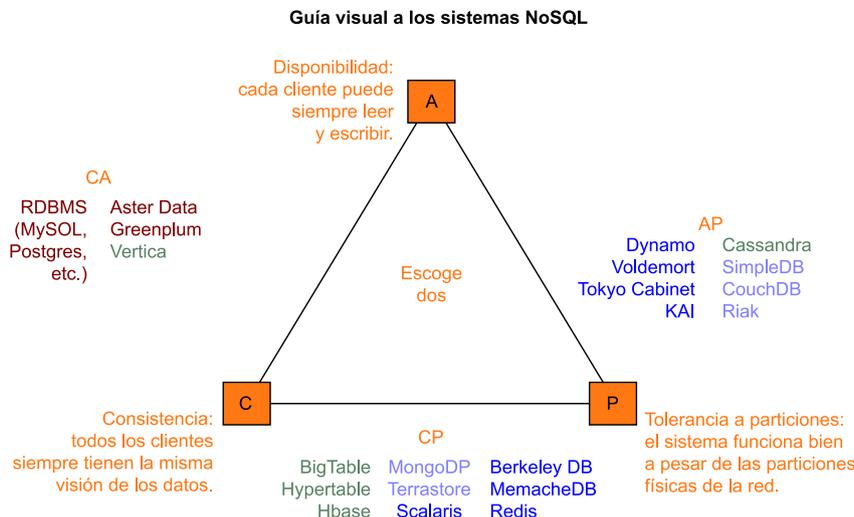


Figura 5. Guía para la selección de gestores de bases de datos NoSQL

Fuente: Matthew Mombrea. *Essential reading for choosing a NoSQL database*.

**4) Sistemas de autenticación, autorización y acceso:** es obvio que nuestro sistema deberá gestionar los dos procesos distintos que intervienen cuando se trata de permitir a los usuarios acceder a los datos de nuestro almacén:

- La autenticación es el proceso de identificación de un individuo sobre la base de sus credenciales (normalmente nombre de usuario y contraseña). El objetivo de la autenticación es decidir si «alguien es quien dice ser».
- Y la autorización, que se refiere a la gestión del acceso a los recursos protegidos y al proceso de determinar si un usuario está autorizado a acceder a un recurso particular.

En muchas ocasiones, para implementarlo será necesario que el sistema se integre con las herramientas de gestión de usuarios y seguridad ya existentes en la organización: LDAP, *active directory*, *single sign on*, CAS, etc.

Será imprescindible documentar los procedimientos de acceso y registrar los accesos efectuados, estableciendo controles de seguridad internos en las diferentes aplicaciones que tratan datos.

El siguiente aspecto de la infraestructura que debemos abordar es la elección de las **aplicaciones** que formarán parte de nuestro almacén de datos. Es una difícil tarea, que se verá condicionada por múltiples factores, a menudo difíciles de compaginar: el ecosistema de aplicaciones existentes (muchos paquetes ofimáticos comerciales ofrecen soluciones BI); la experiencia previa y conocimientos del equipo de TI sobre unas aplicaciones u otras; el presupuesto disponible para *software*, etc.

Sea cual sea la selección de aplicaciones que hagamos, se deberán cubrir suficientemente las siguientes necesidades:

**5) Herramienta de ETL:** el software de extracción y manipulación de datos entra dentro del ámbito de los profesionales de tecnologías de la información y deberá proporcionar, como mínimo, las siguientes funcionalidades:

- Control de la extracción de los datos y su automatización.
- Acceso a diferentes tecnologías.
- Uso de la arquitectura de metadatos.
- Acceso a una gran variedad de fuentes de datos distintas.
- Manejo de excepciones.

- Planificación, *logs*, *interfaces a schedulers* de terceros.
- Interfaz independiente del hardware.

Hoy por hoy, nos encontramos no solo con las ETL como una opción, sino que en general se habla de *data integration* (integración de datos), lo que proporciona mucha más información y enfoques más amplios que responden a nuevos escenarios.

**6) Aplicaciones BI:** en el módulo «Explotación de datos», se identifican las principales herramientas disponibles en el ámbito de BI, ya sean de código abierto o propietarias, que cubren los requerimientos de los usuarios. Destacamos:

- Herramientas de consulta y explotación de datos.
- Portal o interfaz de navegación.
- Gestión de los metadatos de negocio.
- Herramientas analíticas.
- Diseño y desarrollo de aplicaciones específicas.

**7) Herramientas operacionales:** no podemos olvidar el conjunto de herramientas auxiliares para los sistemas que acabamos de presentar:

- Monitorización y registro de eventos: son sistemas capaces de obtener información del entorno donde se despliegan para su posterior análisis.
- Optimizadores de consultas: son el componente del sistema de gestión de base de datos que intenta determinar la forma más eficiente de ejecutar una consulta.
- Limpieza de datos (*data cleansing*) es el acto de descubrimiento, corrección o eliminación de datos erróneos de una base de datos.
- Editores de cubos OLAP: herramienta gráfica que permite la construcción de los esquemas multidimensionales.

Al igual que en otros ámbitos profesionales, también aparecen empresas especializadas que ayudan a mantener la coherencia de nuestro sistema, detectando las carencias, las nuevas oportunidades y la duplicación de recursos (informes, consultas OLAP, etc.)

Sin duda, hoy por hoy se debe apostar por nuevas fórmulas de almacén de datos que permitan la flexibilidad, una rápida incorporación y aprovechamiento de las nuevas tecnologías, etc., y todo esto bajo el concepto *disruptive BI*.

### ***Disruptive BI***

Las principales características son:

Término acuñado por Teradata Corporation, en su publicación *The rise of the disruptive data warehouse*, para presentar los nuevos retos a los que se enfrentan los almacenes de datos para satisfacer a los usuarios del siglo XXI.

Las principales características son:

- **Inteligente:** permite optimizar el tratamiento y la utilización de los recursos y la gestión dinámica de los datos (clasificándolos de calientes a fríos, según su prioridad)
- **Inclusivo:** de muchos tipos de usuarios, muchos tipos de análisis, muchos tipos de datos, etc.
- **Estratégico:** respuestas rápidas, el uso inteligente de los recursos, etc.
- **Flexible:** todos los modelos de implementación, ya sea en las propias instalaciones, hospedado en la nube o híbrido.

### **1.1.2. Plan de implantación**

Más allá de las infraestructuras (herramientas y aplicaciones) identificadas en el apartado anterior, se deben desarrollar una serie de procedimientos que garanticen la correcta y ordenada ejecución de las tareas, y que nos permitirán que los datos que vamos a cargar en el almacén estén disponibles para su normal uso en toda la organización.

El **plan de implantación** es el documento que recoge todas las actividades de índole técnica, y también aquellas actividades de comunicación y formación que acompañan las anteriores para asegurar el éxito del proyecto.

El plan debe ser consensuado por el equipo de proyecto y comunicado al resto de la organización, especialmente a las áreas y personas directamente implicadas como usuarios del nuevo sistema.

Un ejemplo de las actividades que encontraríamos recogidas en un plan sería:

#### 1) Diseño

- Definir las autorizaciones y accesos

- Configurar controles de consistencia de metadatos y datos
- 2) Implantación y despliegue
- Instalación de todos los componentes de la solución
  - Configuración e integración con el resto de las aplicaciones/servicios del ecosistema (configuración de bases de datos, integración con Active Directory, etc.).
  - Diseño e implantación de una arquitectura ajustada a las necesidades.
- 3) Primer desarrollo
- Procesos de carga de datos (ETL).
  - Modelo de datos de ejemplo.
  - Cuadro de mandos de ejemplo.
  - Informe de ejemplo.
- 4) Formación en las tecnologías
- Formación en las herramientas.
  - Ejemplos según los desarrollos efectuados.

Una de las claves del éxito en la construcción de un *data warehouse* es el desarrollo de manera gradual, seleccionando un departamento usuario como piloto y expandiendo progresivamente el almacén de datos a los demás usuarios/departamentos. Por ello, es importante elegir a este usuario inicial o piloto, así como que sea un departamento con pocos usuarios, en el que la necesidad de este tipo de sistemas sea muy alta y se puedan obtener y medir resultados a corto plazo.

## 1.2. Puesta en marcha

Para el despliegue exitoso del proyecto de almacén de datos, el equipo de proyecto debe llevar a cabo su actividad de manera sincronizada, según lo establecido en el **plan de implantación**.

Será necesario que todas las «piezas» que resultan de las fases anteriores (formación, documentación, validación de datos, etc.) estén a punto o se deberá aplazar la puesta en funcionamiento del sistema.

Por ejemplo:

Asegurarse de que ha sido suministrado un diccionario de datos antes de empezar con las etapas fuertes de desarrollo e implementación.

Es común que algunas tareas de desarrollo comiencen mientras otras tareas de análisis están haciéndose en paralelo. En este tipo de escenario, los desarrolladores juegan a los «detectives» por tiempos indefinidos, mientras llevan a cabo la codificación o el mapeo de datos, generando restricciones de tiempo. Por eso, un diccionario de datos debe estar disponible antes de que se inicie cualquier proceso crítico. Que al menos contenga reglas, validaciones y descripciones. Este diccionario debe estar almacenado en la intranet de la empresa, y disponible tanto para usuarios finales como para desarrolladores.

Los *data warehouse* que son construidos sin un diccionario de datos resultarán en duplicidad de datos y procedimientos, así como en confusiones y problemas con los resultados finales.

## 2. Administración del sistema

Ya sabemos que, una vez culminada la implantación de un almacén de datos en una organización, se requiere un conjunto de prácticas que aseguren su buen funcionamiento a lo largo del tiempo

En la figura 6, observamos la existencia del bloque transversal administración del proyecto DW/BI, que recoge las tareas propias de la gestión del proyecto. En este apartado de administración, no nos referimos a él sino a los bloques que aparecen más a la derecha: crecimiento y mantenimiento.

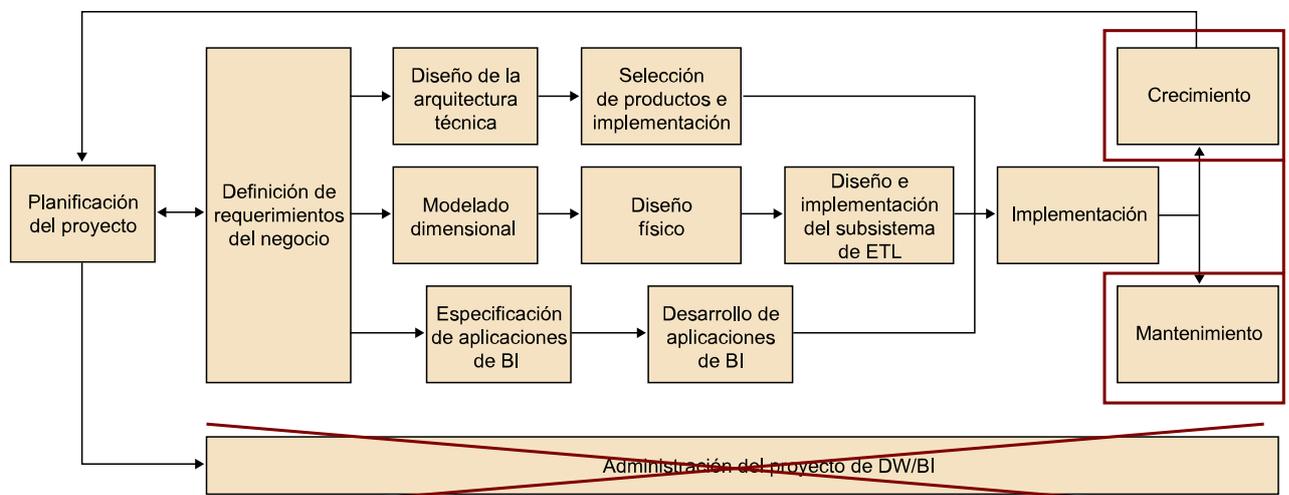


Figura 6. Ciclo de vida Kimball, para *data warehouse*  
Fuente: Kimball Group. *The Data Warehouse Lifecycle Toolkit*.

En el bloque de mantenimiento, se recogen las tareas operacionales de nuestro almacén. Normalmente, se agrupan en tres grandes bloques: monitorización, mantenimiento (propriadamente dicho) y optimización.

En este material vamos a desarrollar de manera separada, junto con el bloque de crecimiento, los tres bloques anteriores, reforzando así la importancia de la monitorización y optimización de nuestro sistema.

Para llevar a cabo las tareas de estos bloques, los responsables del almacén de datos disponen de múltiples herramientas que les serán de gran utilidad, y que aportarán flexibilidad y robustez, de manera independiente de la plataforma donde reside y del volumen de datos que gestionamos.

Para contextualizar las actividades propias de la administración de sistemas de inteligencia de negocio, nos será de gran utilidad identificar el grado de madurez de nuestra organización en el ámbito BI. A medida que nuestra organización es más madura, incorpora nuevas tecnologías y requiere acciones más complejas.

Por ejemplo, en el 2004, P. Eckerson describe el ciclo de madurez del DW, con las siguientes etapas: prenatal, infantil, niñez, adolescencia, adultez y madurez.

- En la prenatal, la empresa trabaja con informes generados por la computadora central y son codificados manualmente.
- En la etapa infantil, la empresa trabaja con hojas de cálculo y/o con un *data mart*.
- En la etapa de la niñez, la empresa trabaja con varios *data mart*, y luego en la adolescencia trabaja con un *data warehouse* que agrupa varios *data mart*.
- Ya en la etapa de adultez, trabaja con un *enterprise data warehouse*, que sí agrupa toda la información de la empresa en un solo punto, y utiliza *scorecards* para hacer un seguimiento del mismo.
- En la última etapa, la de madurez, con los servicios de BI, la empresa se conecta con clientes y proveedores y comparte los datos. En esta etapa, también se utilizan los motores de decisiones que automatizan muchas funciones.

Se constata claramente que los requerimientos asociados a cada una de estas etapas suponen una administración más compleja a medida que evoluciona una organización hacia etapas más maduras.

Si entendemos el sistema *data warehouse* como un servicio TI que prestamos a las distintas áreas de la organización, podemos organizar su administración usando guías y normas existentes que recopilan **buenas prácticas** en el ámbito de la gestión de servicios y proyectos TI. Los referenciales más extendidos y usados son ITIL, ISO, COBIT, PMBOK, etc.

Debemos tener en mente que en ellos se contemplan metodologías de desarrollo de software (o cualquier tipo de soluciones de IT) y/o métodos de control y mejoras para los servicios/productos que se encuentran en la etapa productiva.

Es muy habitual que las organizaciones adopten uno o más de estos referenciales para establecer sus procesos y procedimientos de gestión TI, constituyendo lo que se conoce como gobierno TI.

En la figura 7, se muestra una comparativa/resumen de los aspectos que contempla cada una de ellos.

#### La biblioteca de infraestructura de TI (ITIL)

ITIL es un marco de gestión de servicios de TI (ITSM) que se alinea con las necesidades de la organización.

#### Objetivos de control para la información y tecnologías relacionadas (COBIT)

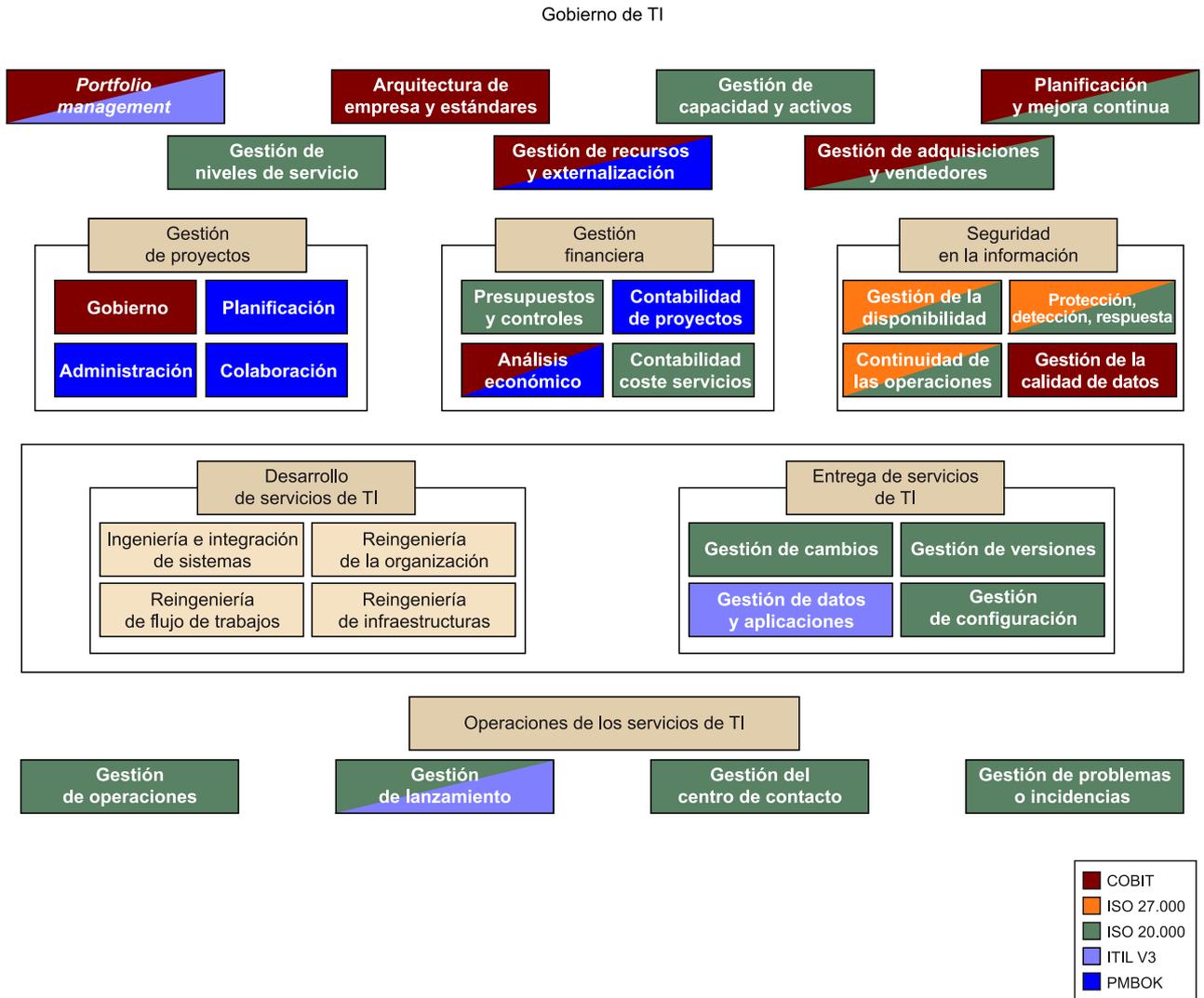
COBIT es reconocido internacionalmente como «la» solución para el Gobierno de TI, con aspectos adicionales de seguridad, calidad y auditoría.

#### Project Management Body of Knowledge (PMBOK)

PMBOK es la guía de los fundamentos para la dirección de proyectos, escrita por PMI.

#### International Standard Organization (ISO)

Conjunto de normas o estándares que también están relacionados con el negocio de TI (ISO 27000, ISO 31500, ISO 20000, etc.).



**Figura 7.** Comparativa referenciales gobierno TI  
Fuente: *Guía de Gestión: operación del servicio basada en ITILv3.* Van Haren Publishing.

Por otra parte, nos encontramos con iniciativas y experiencias que extienden el alcance de las metodologías ágiles (por ejemplo, SCRUM y Kanban) y comúnmente aplicadas al desarrollo de software, para que puedan ser aplicadas también a la administración de los sistemas de almacenes de datos.

Los principios sobre los que se fundamenta esta filosofía son:

- Los individuos y sus interacciones están por encima de los procesos y las herramientas.
- El software que funciona está por encima de la documentación exhaustiva.
- La colaboración con el cliente pasa por encima de la negociación y el cumplimiento de contratos.
- La respuesta ante el cambio pasa por delante de seguir un plan cerrado.

Las principales ventajas que aporta la filosofía *agile* son la reducción del tiempo de entrega de la solución al usuario (*time to client*) y el incremento de valor para el negocio.

En la figura 8, se muestra esquemáticamente su funcionamiento:

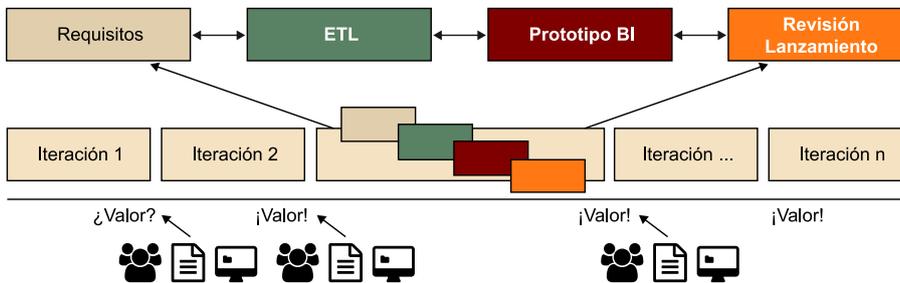


Figura 8. KMS Technology. *Agile BI /DW Life Cycle*.

## 2.1. Monitorización del funcionamiento del sistema

Como no puede ser de otra manera, la supervisión de los sistemas es una de las actividades de alto valor que debemos indiscutiblemente incorporar a nuestra operativa, sobre todo teniendo en consideración el elevado grado de automatización que se puede alcanzar en este tipo de mecanismos.

Enumeramos a continuación algunas de las actividades más comunes:

- 1) Seguimiento del crecimiento del volumen de datos y del espacio de **almacenamiento** requerido.
- 2) Medición del **tiempo de respuesta** de las consultas más frecuentes, para detectar la posible degradación del sistema y corregirla.

Por ejemplo:

- Probar, ejecutar y administrar consultas y vistas de consulta.
- Supervisar el rendimiento de los procesos y aplicaciones.

- 3) **Control de errores** o incidencias que puedan surgir y que deberán ser atendidas lo antes posible para no perder valor.

Por ejemplo:

- Hacer comprobaciones de coherencia de datos y metadatos en el sistema.
- Supervisar la carga de base de datos.

- 4) Será de gran importancia en este apartado proveer de un sistema donde mostrar las **estadísticas de uso**.

- 5) Se deben supervisar las **cadena de ejecución** de procesos periódicos.
- 6) Establecer mecanismos de auditoría que permitan detectar sucesos, a través de la inspección de los archivos de registro (en inglés, *logs*).

Por ejemplo:

- Guardar planes de consultas, tiempos de ejecución y referencias de rendimiento en la base de datos se puede hacer fácilmente si guardamos un tiempo de inicio y un fin de tiempo para cada proceso crítico. Puede ser implementado a través de *stored procedures*, *scripts* o ETL. *queries*  
Guardar estos puntos de referencia ayuda a señalar los puntos problemáticos en el rendimiento, y por tanto a que el equipo de desarrollo se enfoque en la resolución de estos. Estos puntos de referencia forman parte de los metadatos, y son una extensión lógica para fortalecerlos. Se trata de una forma de mejorar mediante el rastreo de la actividad del usuario, identificación de cuellos de botella, más usadas, etc.
- Guardar ETL, validaciones y errores en tablas compartidas de la base de datos.  
La práctica de «atrapar» todos los errores de proceso es similar al punto anterior. Estos errores deben ser detectados, consolidados y enviados a una tabla. Es importante establecer límites para los errores y qué acciones deben ser tomadas cuando estos límites son sobrepasados. Una meta importante en este punto es automatizar el envío de un correo electrónico cuando se sobrepasen los límites en la ocurrencia de un error.

El aumento de potencia de los servidores y el crecimiento del volumen de datos que se generan en las organizaciones exige el uso de soluciones en **tiempo real**, pensadas inicialmente para *big data*.

Por ejemplo:

Splunk es una herramienta que permite buscar, monitorizar y analizar datos de aplicaciones, sistemas e infraestructura IT, generados automáticamente.

En la asignatura de *big data*, se verán detalladamente las herramientas que nos van a permitir la recogida, explotación y análisis de datos masivos, así como los criterios que deberemos tomar a la hora de decidir cuáles implantamos en nuestro almacén.

## 2.2. Mantenimiento del sistema

Bajo este concepto, se engloban tradicionalmente aquellas tareas propias del «día a día» del sistema. Cada vez se amplía más el alcance de esta fase del proyecto, debido a que los sistemas de inteligencia de negocio evolucionan y crecen en complejidad y requerimientos.

Las principales tareas serán: detectar, definir e implementar acciones preventivas para asegurar la sostenibilidad del sistema y la disponibilidad de las plataformas informacionales y de comunicación asociadas.

Al igual que muchos otros sistemas TI, los sistemas de *data warehouse* están fuertemente influenciados por el entorno en el que se encuentran operativos.

En el sentido clásico, los principales elementos que deberemos gestionar adecuadamente son:

1) **Copias de seguridad:** resaltamos la importancia de hacer copias de seguridad de los datos en las empresas, porque el mayor activo de las mismas es su información. Aun así, contemplamos a diario cómo muchas de ellas descuidan este proceso convirtiéndolo en algo accesorio, y no se le da toda la importancia que se merece.

La verdadera utilidad de hacer copias de seguridad de los datos consiste en permitir su posterior restauración (en caso de pérdida accidental, fallo de los dispositivos de almacenamiento, catástrofes, etc.). Si no se comprueba periódicamente el proceso, es imposible garantizar que se está cumpliendo el objetivo de proteger los datos.

Las copias de seguridad pueden ser de tres tipos:

- **Completas o totales:** como su propio nombre indica, este tipo de *backup* copia la totalidad de los datos en otro soporte (cintas, discos duros externos, DVD, etc.).
- **Diferenciales:** solo se hace copia de los archivos que han cambiado desde la última copia de seguridad completa.
- **Incrementales:** en ellas, solo se hacen copias de los datos que han cambiado desde la última copia de seguridad (ya sea completa o incremental).

2) **Estrategia de cargas totales/incrementales:** es muy importante determinar de qué manera se va a cargar la información en las tablas del DW. Una mala decisión en esta etapa puede comprometer la exactitud de los datos, e incluso su validez.

En general, la carga de dimensiones y tablas de hechos tiene dos tipos de abordaje:

- **Totales:** como su nombre indica, es la carga completa de datos cada vez que hay la implementación de un nuevo proceso de ETL. En este tipo de carga, todos los datos fuente se extraen y se procesan, y se vuelven a cargar los datos antiguos, incrementados con los nuevos.
- **Incrementales:** solo se tienen en cuenta los nuevos registros de sistemas operativos en el ETL, y se insertan en el repositorio de DW. Por lo general, las tablas de pequeño tamaño tienen carga completa.

Sin embargo, a diferencia de lo que ocurre con los hechos, por lo general los datos de las dimensiones no se eliminan antes de cada proceso de carga. Además de ser referenciados por tablas, estos datos son «versionados» para reflejar la historia de la información (fenómeno conocido como *slowly changing dimension*).

**3) Riesgos inherentes de todo proyecto tecnológico:** se identifica como riesgo el grado de exposición a resultados inciertos, no cumplimiento de plazos, desviaciones de presupuestos o insuficiencia de resultados alcanzados frente a los planificados. Entre los factores que en este caso concreto han moderado el riesgo del proyecto, destaca el hecho de que se ha elegido como primera área de implantación y utilización una con un alto grado de conocimiento, implicación y uso de tecnologías de información y comunicación en esta empresa en concreto: el área técnico-actuarial. La elección no se hizo en función del área más crítica, sino más bien por el perfil de usuario del área. Los usuarios de esta área normalmente ya utilizaban herramientas de tecnología de información para llevar a cabo tareas típicas en los procesos que intervienen: previsiones o simulaciones con grandes volúmenes de datos.

Por ejemplo:

Destacamos los elementos que han influido en el agravamiento del nivel de riesgo en una organización. Concretamente:

- 1) Existieron dificultades específicas para seleccionar una buena herramienta. Porque no se tenía ninguna experiencia en el uso de las mismas y se desconocían puntos de referencia en otras compañías.
- 2) Por el propio tamaño del proyecto en costes y tiempo necesario para su elaboración, así como la exigencia de dedicación en exclusiva de una parte importante de los implicados, tanto por parte de la organización como por parte de la empresa consultora que ha participado.
- 3) Por la complejidad en el diseño de la base de datos: se parte de modelos ya efectuados, que tienen inconsistencias e incoherencias en la información que albergan. Se ha hecho un esfuerzo especial para tratar de homogeneizar esta información inconsistente.

Conocer el riesgo de un proyecto de implantación de *data warehouse* de manera explícita es importante, puesto que ayuda a evaluar de manera periódica los factores del riesgo del proyecto en cada una de las fases y se puede actuar sobre ellos.

**4) Gestión de cambios** que permiten adaptarse a las nuevas circunstancias y requerimientos que van surgiendo. Principalmente:

- Aplicar **actualizaciones** en la tecnología.
- **Modificar** a demanda los tratamientos de datos.
- Gestionar los **cambios externos** en las fuentes de datos, marcos legislativos, etc.
- Gestión de los **permisos** y autorizaciones.

En este apartado, también se deben gestionar las **peticiones de usuarios**. Cada organización establece un mecanismo para su aprobación por parte de un usuario de negocio autorizado; luego, su evaluación en un ámbito técnico y de recursos requeridos por el Departamento de TI; y finalmente su implementación, si procede. Si al evaluar la petición se determina que excede de un número determinado de horas, suele considerarse fuera del **mantenimiento evolutivo** de la herramienta. En este caso, se deriva la petición para tratarla como una nueva funcionalidad o ampliación del sistema. (Podéis ver el apartado 2.4, «Crecimiento».)

También será muy importante prestar **apoyo** a los usuarios, de tal modo que las dudas, incidencias y consultas que se generen sean atendidas de manera eficiente. Si los usuarios perciben que la herramienta no satisface sus necesidades de información o aparecen dudas sobre los datos mostrados que no resuelven satisfactoriamente, el proyecto se acercará de manera peligrosa, y casi irremediable, al fracaso.

Finalmente, hay que destacar que una de las tareas de administración del *data warehouse* de mayor valor para la organización es la extracción de la información del almacén de datos para la **realimentación** hacia los sistemas operacionales de los resultados obtenidos, enriqueciendo y/o corrigiendo los datos existentes.

### 2.3. Optimización del rendimiento y del sistema

Como todos los sistemas, siempre será posible mejorarlo y optimizarlo de tal modo que la usabilidad y agilidad que percibe el usuario final sean notablemente mejores. Este proceso de refinamiento o ajuste del sistema suele conocerse en el argot como *tuning*.

No obstante, muchas veces un «normal rendimiento» ya es suficiente y aceptable. Hay que evitar la urgencia para llevar a cabo mejoras en un ciclo sin fin, con el objetivo de optimizar el código de las herramientas, la base de datos, etc.

Las herramientas clásicas usadas para este cometido nos permiten establecer una **configuración muy afinada de los parámetros** del sistema:

- Cambiar la asignación de recursos de máquina del sistema.
- Activar agregaciones para mostrar los datos en las jerarquías y atributos de los cubos OLAP.
- Ajustar parámetros relacionados con el rendimiento.
- Reconfigurar las opciones de caché del servidor.

- Activar un sistema de balanceo de carga.
- Definir índices en un ámbito de SGBD para las consultas más costosas.

Por ejemplo:

- Evitar transacciones con tiempos de ejecución largos. *stored procedures rollbacks*  
Puede darse el caso de operaciones sobre millones de registros que llenen completamente la bitácora de transacciones de la base de datos, haciendo que el procesamiento prácticamente se detenga. Para evitar esto, es necesario programar modularmente los y hacer que las transacciones tengan operaciones pequeñas. Además, esto permite una recuperación más rápida y un menor número de cambios () cuando un error suceda.
- Usar la integridad referencial cuidadosamente.  
Es recomendable no abusar de la integridad referencial. Mientras que las restricciones en claves foráneas ayudan a la integridad de los datos, tienen asociado un costo en cada inserción, actualización o eliminación. Se puede considerar la ventaja de implementar ciertas validaciones en los ETL.

Por otra parte, los **analizadores y aceleradores de consultas** nos permiten optimizar tiempos de respuestas en las necesidades analíticas, o de carga de los diferentes datos desde los sistemas operacionales hasta el *data warehouse*.

Trabajan volcando sobre un fichero de log las consultas ejecutadas y datos asociados a las mismas (tiempo de respuesta, tablas accedidas, método de acceso, etc.). Este log se analiza, bien automáticamente o mediante la supervisión del administrador de datos, para mejorar los tiempos de accesos.

El hecho de implantar un sistema analizador de consultas en el entorno real tiene una serie de ventajas, tales como:

- Se pueden implantar mecanismos de optimización de las consultas, reduciendo la carga del sistema.
- Se pueden implantar mecanismos de bloqueo para las consultas que vayan a implicar un tiempo de respuesta excesivo.

Por ejemplo:

Entender siempre la optimización de la base de datos y los planes de consultas (*query plans*).

Algunos consejos en este apartado incluyen comprender la naturaleza de los datos y su relación con las demás entidades. Antes de ejecutar una consulta, se puede compilar y ejecutar en modo *non-exec* y verificando el *query plan*. Hay que limitar los resultados de las consultas cuando se esté en fase de prueba.

Más allá de la optimización del rendimiento del sistema, no debemos olvidar la necesidad de optimizar mediante la revisión del propio sistema: reescribiendo los procesos ETL, revisando el modelo físico, rediseñando los informes y cuadros de mando, reduciendo las cargas de datos en la medida de lo posible, etc.

Para llevar a cabo estas tareas, existen distintas estrategias: *pair programming*, *test driven development*, *refactoring* de código, etc.

### ***Pair programming***

En la programación en parejas: se recomienda que las tareas de desarrollo sean llevadas a cabo por dos personas en un mismo puesto. La mayor calidad del código escrito de esta manera –el código es revisado y discutido mientras se escribe– es más importante que la posible pérdida de productividad inmediata.

Finalmente, no podemos dejar de lado la tarea de **vigilancia tecnológica**, que nos permitirá estar al corriente de las nuevas herramientas y técnicas que van apareciendo y que contribuyen sin duda a un mejor desempeño de los sistemas.

Se deberá evaluar cuáles son de aplicación a nuestro sistema, para poderlas aprovechar al máximo, mejorando la usabilidad y tiempo de respuesta del sistema.

## **2.4. Crecimiento**

Los almacenes de datos tienden, por naturaleza, a expandirse. Cuando esto sucede, es considerado un signo de éxito, puesto que denota un alto índice de adopción del nuevo sistema a lo largo de la organización.

Normalmente, encontramos dos formas de crecimiento:

1) Por un lado, se deberán atender la **demand**a de peticiones y cambios que nos hagan los usuarios. La apropiada gestión de los nuevos requerimientos que aparecen será una de las prioridades que hay que gestionar. Una buena opción es empezar el ciclo de nuevo, construyendo sobre las bases ya establecidas y sin perder el enfoque en los nuevos requerimientos.

A menudo, se requieren nuevas funcionalidades que por su complejidad, tiempo de desarrollo o simplemente porque las tecnologías del sistema no lo soportan, no se van a poder resolver desde el mantenimiento evolutivo. Será necesario llevar a cabo pequeños desarrollos o pequeños proyectos que van ampliando las posibilidades que ofrece el sistema

2) Por otra parte, se deberá continuar con la estrategia de **ampliación** del *data warehouse*, incorporando nuevos *data mart* para hacerlo extensivo al resto de las áreas de nuestra organización.

Finalmente, las dos circunstancias nos obligarán a llevar a cabo una adecuada **planificación de datos**, que nos permita que el sistema siga funcionando con normalidad.

### **Refactoring**

La refactorización (del inglés *refactoring*) es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

Por ejemplo, en algunas organizaciones recogen la planificación en el «plan de expansión».

Dado que el almacén evoluciona y los datos que contiene llegan a ser más accesibles, los empleados externos podrían descubrir también el valor de sus datos. Al enlazar su *data warehouse* con otros sistemas (tanto internos como externos a la organización), se puede compartir información con otras entidades comerciales con poco o ningún desarrollo. Los mensajes de correo electrónico, servidores WEB y conexiones intranet/internet pueden entregar listas por niveles a sus proveedores o según su condición, a sus socios de negocio.

Puesto que los *data warehouse* continúan creciendo en sofisticación y uso, los datos acumulados dentro de una empresa llegarán a ser más organizados, más interconectados, más accesibles y, en general, más disponibles a más empleados.

El resultado será la obtención de mejores decisiones en el negocio, más oportunidades y más claridad de trabajo.

### 3. Roles en la administración del sistema

A lo largo de este material, hemos ido observando aquellas tareas que se deben hacer para una apropiada gestión de un sistema de *data warehouse*, y también aquellos elementos sobre los que llevamos a cabo estas actuaciones.

Ahora llega el momento de determinar quién debería hacer estas acciones, atendiendo a su perfil, formación, responsabilidad, experiencia, etc.

Relacionamos a continuación los principales **perfiles profesionales** involucrados en la administración de un almacén de datos:

#### a) Director BI (o gestor de sistemas de información).

- Le corresponde la coordinación, organización y planificación de los diferentes proyectos de *business intelligence* destinados a cubrir las necesidades de información dentro de la organización.
- Un requisito importante será tener una buena comprensión del modelo de negocio de las áreas cliente y detectar las oportunidades que puede ofrecerles el BI, así como coordinar, organizar y planificar la estrategia general de *business intelligence*.
- Gestión del área de *business intelligence* dentro del Departamento de TI, con el fin de dar servicio a las diferentes áreas: *data marts*, herramientas de análisis, de información y de apoyo a los sistemas de análisis de negocios.
- Será el enlace entre las diferentes áreas de negocio, en particular las ventas, control financiero y el Departamento de Gestión de Clientes.

#### b) Analista de sistema / modelador de datos

Los analistas de sistemas del área informática de la compañía han sido los encargados fundamentales de subsanar el problema. Son personas capaces de comprender y comunicarse con los usuarios para establecer los requerimientos. El analista tendrá también conocimientos técnicos de informática. Transmite los requerimientos en los términos que los programadores puedan comprender. Es importante que el analista sea un buen comunicador y pueda pensar desde el punto de vista del usuario, así como del programador.

La tarea del analista no ha quedado únicamente restringida a suministrar especificaciones a los programadores. El analista también ha tenido que cumplir con un conjunto de responsabilidades:

- Investigar y analizar el sistema que existe y recabar los nuevos requerimientos.
- Juzgar la viabilidad del desarrollo de un sistema de información de este tipo para las áreas concretas.
- Diseñar el sistema nuevo, especificando programas, hardware, datos, estructuras de control y otros procedimientos.
- Probar y revisar la instalación del sistema nuevo, generando documentación que facilita sus funciones y evaluando el desarrollo.

### c) Administrador de base de datos (DBA)

Las responsabilidades incluyen el diseño, implementación y mantenimiento del sistema de base de datos; el establecimiento de políticas y procedimientos relativos a la gestión, la seguridad, el mantenimiento y el uso del sistema de gestión de base de datos. Se espera que un DBA se mantenga al tanto de las nuevas tecnologías y los nuevos enfoques de diseño.

Se encarga de la creación y mantenimiento del modelo de datos en el sistema gestor de base de datos (SGBD). Entre sus competencias, también se encuentran la implementación del modelo, el desarrollo de volumen y estimaciones de capacidad, optimización del rendimiento y administración general del DBMS.

### d) Desarrollador de integración

Es el responsable del diseño, desarrollo y despliegue de los procesos ETL. Sabemos que los procesos de extracción, transformación y carga (ETL) son importantes, ya que se trata de la forma en que los datos se guardan en un almacén de datos (o en cualquier base de datos). Será el encargado del modelado y diseño de la arquitectura de los datos de las ETL: *operational data store, staging area, data warehouse*, etc.

### e) Desarrollador de aplicaciones

Los programadores han sido los responsables de convertir los requerimientos en programas. El programador es un técnico en informática que tiene que ser capaz de entender las necesidades de los usuarios. En muchos casos, el lenguaje que utilizan usuarios y programadores es tan diferente que aparecen malentendidos en la comunicación.

Algunas de las funciones que va a llevar cabo son:

- Automatización y cálculo de indicadores y cuadros de mando.

- Desarrollo de soluciones BI.
- Integración de nuevas plataformas y mantenimiento de correctivos / evolutivos.
- Consolidación y aplicación de mejoras tecnológicas en el *data warehouse* corporativo.
- Automatización y documentación de los procesos definidos.
- Generación de manuales de usuario.

#### f) Servicios de terceros (consultores)

Un consultor de BI necesita aportar un valor diferencial que haga avanzar el proyecto en el que participa, no solo desde el punto de vista tecnológico, sino desde el punto de vista de análisis de negocio. Debe saber de tecnología y de negocio, con lo que recuerda un poco a la figura del CIO.

#### g) Otros

A veces, se incorporan al equipo otras personas:

- **Especialista de aplicación:** se encarga de proporcionar información detallada sobre los modelos de datos, metadatos, los controles de auditoría y de proceso para los analistas funcionales, arquitectos técnicos y otros participantes, en relación con el sistema de origen o destino.
- **Desarrollador de *data quality*:** es responsable de diseño, desarrollo y documentación de los procesos de limpieza, estandarización y enriquecimiento de datos.
- **Analista *tester* con orientación técnica y/o funcional:** sus principales responsabilidades son la ejecución de tareas del plan de *testing*, generación de lotes de prueba, documentación de apoyo de las pruebas, etc.

Según el tamaño de la organización y su Departamento de TI, no encontraremos tantas personas como roles. Con frecuencia, una persona del equipo asume uno o varios roles. Es importante establecer quién hace qué, para que las responsabilidades queden claramente definidas, evitando malentendidos innecesarios.

Evidentemente, se trabaja en equipo y las diferentes personas deben «alinearse» sus decisiones desde una perspectiva de conjunto.

También podemos encontrarnos con **equipos multidisciplinares** que se forman de manera específica para un proyecto. En este caso, la coordinación entre ellos y la sintonización de sus distintas formas de actuar será vital para el buen funcionamiento del proyecto.

El caso es distinto cuando el equipo es estable y experimentado y va desarrollando proyectos de esta índole mientras crece y se extiende el almacén de datos.

No podemos cerrar este módulo sin mencionar al **usuario final** que, sin ser del Departamento de TI, va a ser el que rentabilice el almacén de datos. En este sentido, podemos decir que el usuario es la pieza clave sobre la que debe girar la estrategia de implantación. Será quien haga de prescriptor de esta herramienta.



## Glosario

**almacén de datos** *m* Bases de datos orientadas a áreas de interés de la empresa que integran datos de distintas fuentes con información histórica y no volátil, y que tienen como objetivo principal dar apoyo en la toma de decisiones. Puede ser corporativo o departamental.  
*en* data warehouse

**base de datos NewSQL** *f* Clase de sistemas modernos de gestión de bases de datos relacionales que tratan de conseguir el mismo rendimiento escalable de las bases de datos NoSQL para el procesamiento de transacciones en línea (lectura-escritura), manteniendo durante las cargas de trabajo las garantías ACID de un sistema de base de datos tradicional.

**base de datos NoSQL** *f* Por lo general, los investigadores académicos se refieren a este tipo de bases de datos como almacenamiento estructurado, término que abarca también las bases de datos relacionales clásicas. A menudo, las bases de datos NoSQL se clasifican según su forma de almacenar los datos, y comprenden categorías como clave-valor, las implementaciones de *big table*, bases de datos documentales y bases de datos orientadas a grafos.

**base de datos posrelacionales** *f* Base de datos cuyo modelo de datos incorpora relaciones pero no limitadas por las restricciones del principio de información de E. F. Codd, que requiere que toda información en la base de datos debe ser modelada en términos de valores en relaciones nada más.

**big data** *m* Conjunto de estrategias, tecnologías y sistemas para el almacenamiento, procesamiento, análisis y visualización de conjuntos de datos complejos.

**corporate information factory** *f* Véase **factoría de información corporativa**.

**data governance** *m* Véase **gobierno del dato**.

**data mart** *m* Subconjunto de los datos del *data warehouse*, con el objetivo de responder a un determinado análisis, función o necesidad y con una población de usuarios específica.

**data warehouse** *m* Véase **almacén de datos**.

**disruptive data warehouse** *m* Término que identifica un almacén de datos evolucionado, que integra las mejores tecnologías existentes y hace que sean fáciles de usar. Se reduce la barrera de entrada, poniendo la innovación al alcance de cualquier organización.

**ETL** *m* Véase **extract transform load**.

**extract transform load** *f* Término en inglés para denominar los procesos de extracción, transformación y carga que alimentan los almacenes de datos.  
sigla **ETL**

**factoría de información corporativa** *f* Conjunto de elementos de software y hardware que ayudan al análisis de datos para tomar decisiones.  
sigla **FIC**  
*en* corporate information factory

**FIC** *m* Véase factoría de información corporativa.

**gestión de datos maestros** *f* Metodología que identifica la información más crítica de una organización y crea una sola fuente fiable.

**gobierno del dato** *m* Metodología que tiene por objeto asegurarse de que los datos son siempre fiables y válidos en cada contexto empresarial, que su calidad se mantiene a lo largo del tiempo y que existen mecanismos de control sobre quién puede hacer qué con los datos en cada momento.  
*en* data governance

**information technology infrastructure library** *m* Conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general. ITIL da descripciones detalladas de un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI. Estos procedimientos son independientes del proveedor, y han sido desarrollados para servir como guía que abarque toda infraestructura, desarrollo y operaciones de TI.  
sigla **ITIL**

**ITIL** *m* Véase **information technology infrastructure library**.

**massively parallel processing (o MPP)** *m* Tecnología que cubre las necesidades de altas prestaciones y escalabilidad de los Data Warehouse, en la que los sistemas suelen ser casi independientes, comunicados por intercambiadores de alta velocidad que permiten gestionarlos como un único sistema. Se conocen por ello como arquitecturas de “nada compartido”. Su escalabilidad es mayor que la de los SMP.

**master data management** *m* Véase gestión de datos maestros.  
sigla **MPP**

**MPP** *m* Véase **massively parallel processing**.

**metadato** *m* Dato sobre datos que representa características de otros datos que facilitan su administración y uso.

**minería de datos** *f* Extracción de información de alto valor añadido (patrones ocultos, tendencias y correlaciones) a partir de datos en bruto.

**OLAP** *m* Siglas que hacen referencia a las herramientas de análisis, normalmente multidimensional en *on-line analytical processing*.

**OLTP** *m* *On-line transactional processing*.

**Operational Data Store** *m* Almacén de datos operacional.

**SCD** *f* Véase **slowly changing dimensión**.

**SGBD** *m* Véase **sistema de gestión de bases de datos**.

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

**slowly changing dimension** *f* Estrategia de tratamiento de los datos de las dimensiones de análisis de un modelo multidimensional que cambia con el tiempo, es decir, una dimensión que contiene datos que cambian lentamente.  
sigla **SCD**

**symmetric multiprocessing** *m* Tecnología que cubre las necesidades de altas prestaciones y escalabilidad de los *data warehouse*, en la que los sistemas tienen múltiples procesadores que comparten un único bus y una gran memoria, repartiéndose los procesos que genera el sistema, y el sistema operativo es el que gestiona esta distribución de tareas. Estos sistemas se conocen como arquitecturas de «casi todo compartido». El aspecto más crítico de este tipo de sistemas es el grado de rendimiento relativo respecto al número de procesadores presentes, debido a su crecimiento no lineal.  
sigla **SMP**

## Bibliografía

**Hughes, R.** (2013). *Agile Data Warehousing Project Management: Business Intelligence Systems Using Scrum*. Ed. Morgan Kaufman.

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

**Kimball Group** (2008). *The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems*. Ed. Wiley.

**Phillips, J.** (2011). *IT Project Management: On Track from Start to Finish*. Ed. McGraw Hill.

