

## Tema 2. Proceso de Creación de Bases de Datos

### Objetivos:

- Identificar y conocer las fases del ciclo de vida de un sistema de aplicación de bases de datos.
- Identificar las etapas de un método de diseño de bases de datos.

### Esquema de Contenidos:

#### 2.1 CICLO DE VIDA DE UN SISTEMA DE APLICACIÓN DE BASES DE DATOS.

##### 2.1.1 Introducción

##### 2.1.2 Fases del ciclo de vida de un sistema de información

##### 2.1.3 Fases del ciclo de vida de un sistema de aplicación de base de datos

#### 2.2 MÉTODOS DE DISEÑO DE BASES DE DATOS

##### 2.2.1 El proceso de diseño de bases de datos

##### 2.2.2 Conceptos y etapas de un método para el desarrollo de bases de datos

##### 2.2.3 Características deseables de un método de diseño

#### BIBLIOGRAFÍA

#### REFERENCIAS

## 2.1 CICLO DE VIDA DE UN SISTEMA DE APLICACIÓN DE BASES DE DATOS.

### 2.1.1 Introducción

Las bases de datos se han convertido en una parte importante de la gestión de los recursos de información<sup>1</sup> (en general, de los sistemas de información) en muchas organizaciones (gobierno, industrias, comercio, etc.) como consecuencia de las ventajas que conlleva su uso, tal y como vimos en el tema anterior.

Cuando se crea una base de datos pequeña, utilizada por pocos usuarios (quizá hasta 20), el diseño no resulta muy complicado. Pero cuando se trata de bases de datos medianas o grandes para el sistema de información de una gran organización, el proceso de diseño es complejo, pues dicho sistema de información debe satisfacer las necesidades de muchos usuarios diferentes (de varias decenas a cientos).

La complejidad del diseño de una base de datos no está determinada por el tamaño total de la información que contiene, sino por su esquema. Si una base de datos tiene un esquema complejo, con más de 30 o 40 elementos y un número similar de relaciones entre ellos, entonces su diseño debe realizarse de forma cuidadosa.

Con el término *base de datos grande*, nos referimos a una base de datos...

- con varias decenas de gigabytes (Gb) de información,
- un esquema con más de 30 o 40 elementos distintos,
- sobre la que se realizan cientos de consultas, y
- que da servicio a varios grupos de aplicaciones, cada una con decenas o cientos de usuarios.

Ejemplos de bases de datos grandes serían las utilizadas comúnmente en la industria de servicios (esto es: bancos, reservas de vuelos o de hoteles, energía eléctrica, agua, seguros, etc.).

Los sistemas para este tipo de bases de datos son los **sistemas de procesamiento de transacciones de alto rendimiento**, (conocidos como OLTP, *online transaction processing*). Estos sistemas dependen de que sus bases de datos funcionen a la perfección (estén operativas) las 24 horas del día, los 7 días de la semana: cientos de transacciones procedentes de terminales o equipos locales y remo-

<sup>1</sup> Recursos que participan en la recopilación, administración, uso y difusión de la información.

tos acceden a la base de datos cada minuto. Como su propio nombre indica, en estos sistemas es decisivo el rendimiento de las transacciones, en términos del *promedio de transacciones por minuto* (productividad de transacciones), así como del *tiempo medio y máximo de respuesta* del sistema (a cada transacción).

Es imprescindible, por tanto, realizar un adecuado diseño de la base de datos tal que...

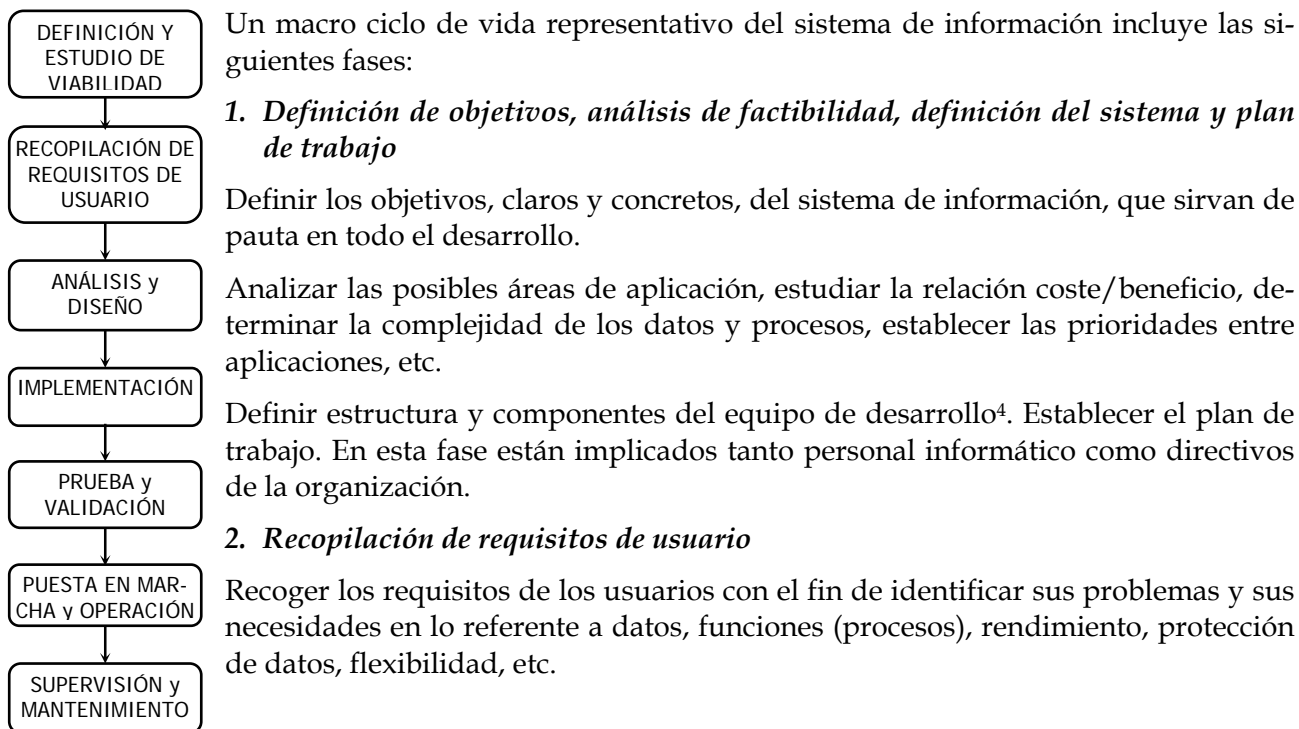
- proporcione una estructuración de la información natural y fácil de entender,
- satisfaga los requisitos tanto de *contenido* de información como de *procesamiento* (productividad, tiempos de respuesta, y espacio de almacenamiento), y
- permita y facilite la evolución del sistema (necesaria si surgen cambios en los requisitos)<sup>2</sup>.

Con el término **diseño de bases de datos** nos referimos al proceso de diseño en un entorno de este tipo, en una organización grande.

El sistema de base de datos suele ser parte de un sistema de información mucho mayor, con el que la organización controla sus **recursos de información**, que son: los datos en sí mismos, el SGBD, el hardware y los medios de almacenamiento, el personal que usa los datos (ABD, usuarios finales, etc.), el software de aplicación que accede y actualiza los datos, y los programadores que crean estas aplicaciones.

A continuación veremos un *ciclo de vida representativo* de un sistema de información y el lugar que ocupa el sistema de bases de datos dentro de dicho ciclo de vida. En aquellos sistemas de información en los que las bases de datos son el componente fundamental, es difícil establecer la separación entre el ciclo de vida del sistema de información (*macro ciclo de vida*) y el del sistema de base de datos (*micro ciclo de vida*).

### 2.1.2 Fases del ciclo de vida de un sistema de información<sup>3</sup>



<sup>2</sup> El coste puede ser muy alto si un sistema grande y complejo es incapaz de evolucionar y hay que sustituirlo por otros productos de SGBD.

<sup>3</sup> Es necesario señalar que el ciclo de vida que vemos en esta asignatura está enfocado desde la perspectiva del paradigma de **Análisis Estructurado**. Además, no se ha considerado la "Creación de Prototipos".

<sup>4</sup> Incluyendo la composición y responsabilidades del equipo a cargo de la creación, administración y uso de la BD.

### 3. Diseño (análisis y diseño)

Diseñar la estructura del sistema de información, esto es: de la base de datos y de los sistemas de aplicación (programas) que usan los datos y los procesan.

### 4. Implementación, carga o conversión de datos y aplicaciones

Implementar el sistema de información, cargar datos reales en la base de datos (nuevos u otros ya existentes convertidos al formato adecuado) e implementar y probar las transacciones de la base de datos (operaciones sobre los datos a través de los programas).

### 5. Pruebas de aceptación, validación y ajuste

Probar el sistema con base en los criterios de rendimiento y las especificaciones de comportamiento. Un sistema queda *validado* (es aceptable) cuando satisface los requisitos de los usuarios y criterios de rendimiento.

### 6. Operación

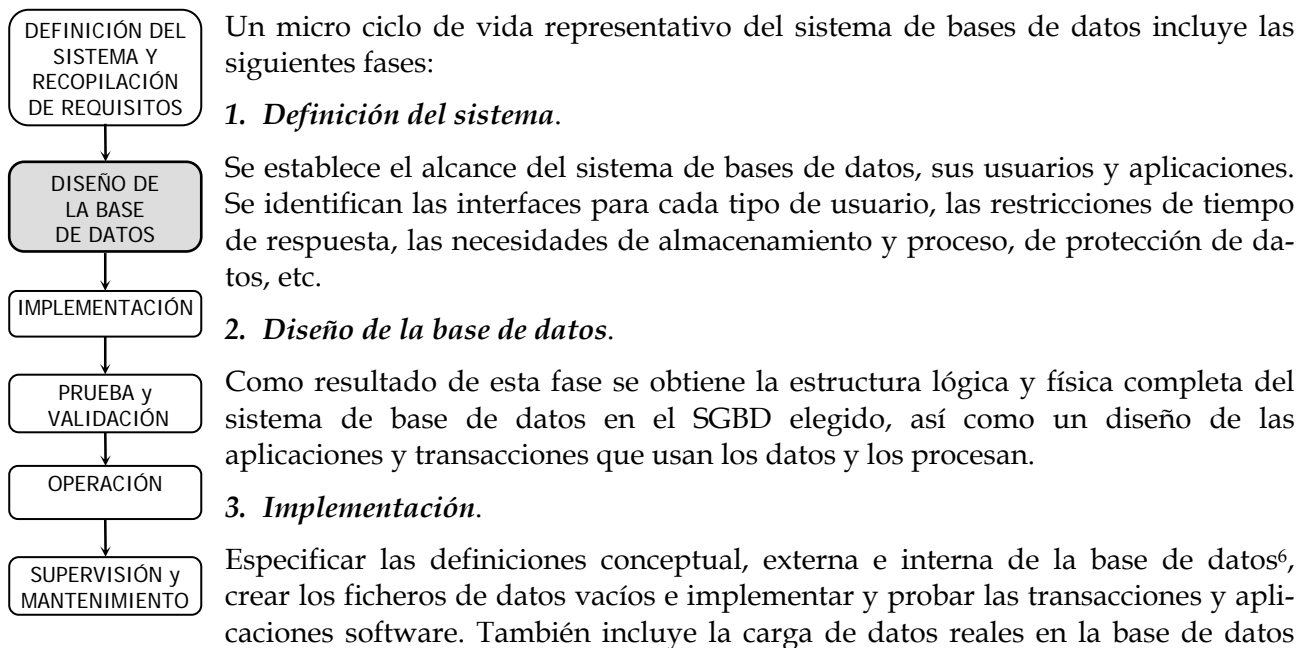
Puesta en marcha del sistema, quizá tras proporcionar a los usuarios la formación adecuada. La fase operativa comienza cuando todas las funciones (procesos) del sistema están disponibles y han sido validadas.

### 7. Supervisión y mantenimiento

Una vez que el sistema está operativo (en fase de operación), se vigila continuamente. Es posible que surjan nuevas necesidades de datos o se requieran nuevas aplicaciones o se deban corregir errores no detectados en la validación. Si esto ocurre, estos nuevos requisitos pasan por todas las fases anteriores hasta, una vez validados, ser incorporados al sistema.

Cada una de estas fases es, en sí, un proceso **iterativo**. Además, en general suele existir **retroalimentación** entre las diversas fases del ciclo de vida de un sistema de información: durante la realización de una de las fases podemos vernos obligados a volver a una fase anterior<sup>5</sup>.

#### 2.1.3 Fases del ciclo de vida de un sistema de aplicación de base de datos



<sup>5</sup> No veremos con detalle estos aspectos, pues son estudiados en la asignatura “Fundamentos de Ingeniería del software”, de 3º de Ingeniero en Informática.

<sup>6</sup> En términos de la arquitectura de tres esquemas, ANSI/X3/SPARC, de los sistemas de gestión de bases de datos (véase Tema 1)

(nuevos u otros ya existentes convertidos al formato adecuado) y la conversión de aplicaciones software de sistemas anteriores al nuevo sistema.

#### 4. *Prueba y validación.*

Probar y validar el nuevo sistema. Si es necesario por cuestiones de optimización del rendimiento, modificar (ajustar) la estructura física de la base de datos (e incluso la lógica). Si la base de datos ha sido bien diseñada, estas modificaciones no afectarán a los programas de aplicación.

#### 5. *Operación.*

Puesta en marcha del sistema de bases de datos y sus aplicaciones.

#### 6. *Supervisión y mantenimiento.*

Una vez que el sistema de bases de datos está operativo, se vigila y mantiene constantemente. Es posible que de vez en cuando se necesiten modificaciones y reorganizaciones importantes, como respuesta a cambios en los requisitos de contenido de datos o de aplicaciones de software.

## 2.2 MÉTODOS DE DISEÑO DE BASES DE DATOS

Nos centraremos ahora en la segunda de las fases del micro ciclo de vida, es decir, en explicar en qué consiste el proceso de diseño de bases de datos, la dificultad que entraña su realización y la necesidad de llevarlo a cabo siguiendo procedimientos sistemáticos y ordenados, es decir, aplicando un método.

### 2.2.1 El proceso de diseño de bases de datos<sup>7</sup>

Según se indica en [EN 2002], es el proceso de *diseñar la estructura lógica y física de una o más bases de datos para satisfacer las necesidades de información de los usuarios en una organización, para un conjunto definido de aplicaciones.*

Los **objetivos del diseño de bases de datos** son los siguientes:

- Satisfacer requisitos de *contenido* de información de usuarios y aplicaciones
- Proporcionar una estructuración de los datos natural y fácil de entender
- Soportar los requisitos de *procesamiento* y de rendimiento (tiempo de respuesta, tiempo de procesamiento, espacio de almacenamiento)
- Conseguir un *esquema de base de datos flexible*, es decir, que sea posible modificarlo (como consecuencia de cambios en los requisitos del sistema) una vez implementada la base de datos.

El proceso de diseño de bases de datos es una tarea larga y costosa; durante mucho tiempo fue considerada una tarea de expertos (*más un arte que una ciencia*), sin embargo, en la actualidad numerosos autores reconocen que el diseño ha de llevarse a cabo siguiendo procedimientos ordenados y metódicos, es decir, aplicando *métodos*<sup>8</sup>.

En distintas áreas de la Ingeniería del Software se ha realizado grandes esfuerzos para encontrar los métodos de diseño más adecuados. Y es que el uso de un método tiene un importante impacto en el desarrollo de un producto software (costes y plazos de entrega, calidad, mantenimiento, etc.).

Según Sommerville [S 1988], *“un buen diseño es la clave de una eficiente ingeniería del software. Un sistema software bien diseñado es comprensible y fiable, fácil de aplicar y de mantener. Un sistema software mal diseñado puede funcionar, pero será costoso de mantener, difícil de probar y poco fiable”*.

---

<sup>7</sup> También “Proceso de Desarrollo de Bases de Datos”

<sup>8</sup> Aunque también suele emplearse el término “metodología”, actualmente se prefiere el uso de “método”.

## 2.2.2 Conceptos y etapas de un método para el desarrollo de bases de datos

### *Concepto de método*

Según [MPM 1999]: “Un método de diseño de bases de datos es un conjunto de *herramientas* que facilitan la representación de los datos en cada fase del proceso de diseño de una base de datos, junto con las *reglas* que permiten el paso de una fase a la siguiente”.

Una *herramienta* es “cualquier recurso particular a disposición del método, para realizar las operaciones que en él se prevén”. Son herramientas los *modelos de datos*, los *lenguajes de datos*, la *documentación*, y *otras herramientas*, que debido a su especial interés son considerados de forma separada:

Un *modelo de datos* es “un conjunto de conceptos, reglas y convenciones que permiten describir y manipular los datos de la parte del mundo real que constituye el universo de discurso” (minimundo). El esquema obtenido al describir cierto minimundo mediante las construcciones de un modelo de datos, será la visión del mundo real que tiene el diseñador, que lo ve en el contexto del sistema de información que está creando.

Un *lenguaje de datos* es el resultado de definir una sintaxis sobre un modelo de datos; permite expresar un esquema (por ejemplo basado en el modelo relacional) en una sintaxis concreta (por ejemplo la de SQL).

La *documentación* permite describir los resultados de cada etapa, facilita el trabajo de los integrantes del equipo de diseño, la comunicación entre ellos, y la revisión y mantenimiento de la base de datos.

Las *reglas* actúan sobre los elementos de entrada de cada fase del diseño, para conseguir las salidas; permiten el paso de una fase a la siguiente.

Cuando se habla de *otras herramientas*, se suele hacer referencia a herramientas software 1) de tipo CASE -*computer aided software engineering*: ingeniería del software asistida por ordenador, que permiten el diseño conjunto de esquemas y aplicaciones de base de datos-, 2) de adquisición de requisitos, y 3) de diccionario de datos.

No existe un método de desarrollo de bases de datos que sea considerado como un “estándar”, debido a la diversidad de opiniones y enfoques que existen dentro de la comunidad de investigadores en el área de las bases de datos.

En esta asignatura estudiamos un método de diseño de bases de datos basado en los métodos propuestos en los libros [EN 2002] y [MPM 1999], y está especialmente orientado al desarrollo en el modelo de datos relacional.

### *Etapas de un método de diseño de bases de datos relacionales*

1. Recopilación de requisitos de usuario
- 2. Diseño conceptual**
3. Elección del SGBD
- 4. Diseño lógico**
- 5. Diseño físico**
6. Implementación

Consideraremos que las etapas 1. *Recopilación de requisitos de usuario* y 6. *Implementación* no forman parte del diseño de la base de datos en sí, sino del desarrollo del sistema de información más general.

La etapa 3. *Elección del SGBD* consiste en la evaluación y selección (y adquisición, si es el caso) de equipo físico y lógico, con base en las necesidades (SGBD, memoria principal y secundaria, capacidad de proceso, etc.) que tenga el sistema de información que se está desarrollando.

A lo largo de esta asignatura, nos centraremos en el estudio de las tres etapas que constituyen el núcleo del proceso de diseño de bases de datos, esto es: 2. *Diseño Conceptual*, 4. *Diseño Lógico* y 5. *Diseño Físico*.

Por otro lado, el proceso de diseño de bases de datos no sólo aborda los aspectos de diseño relacionados con la información, sino que también presta atención al diseño de las operaciones que manipularán los datos. De hecho, el proceso de diseño de bases de datos consta de dos grandes actividades, que es necesario realizar en paralelo:

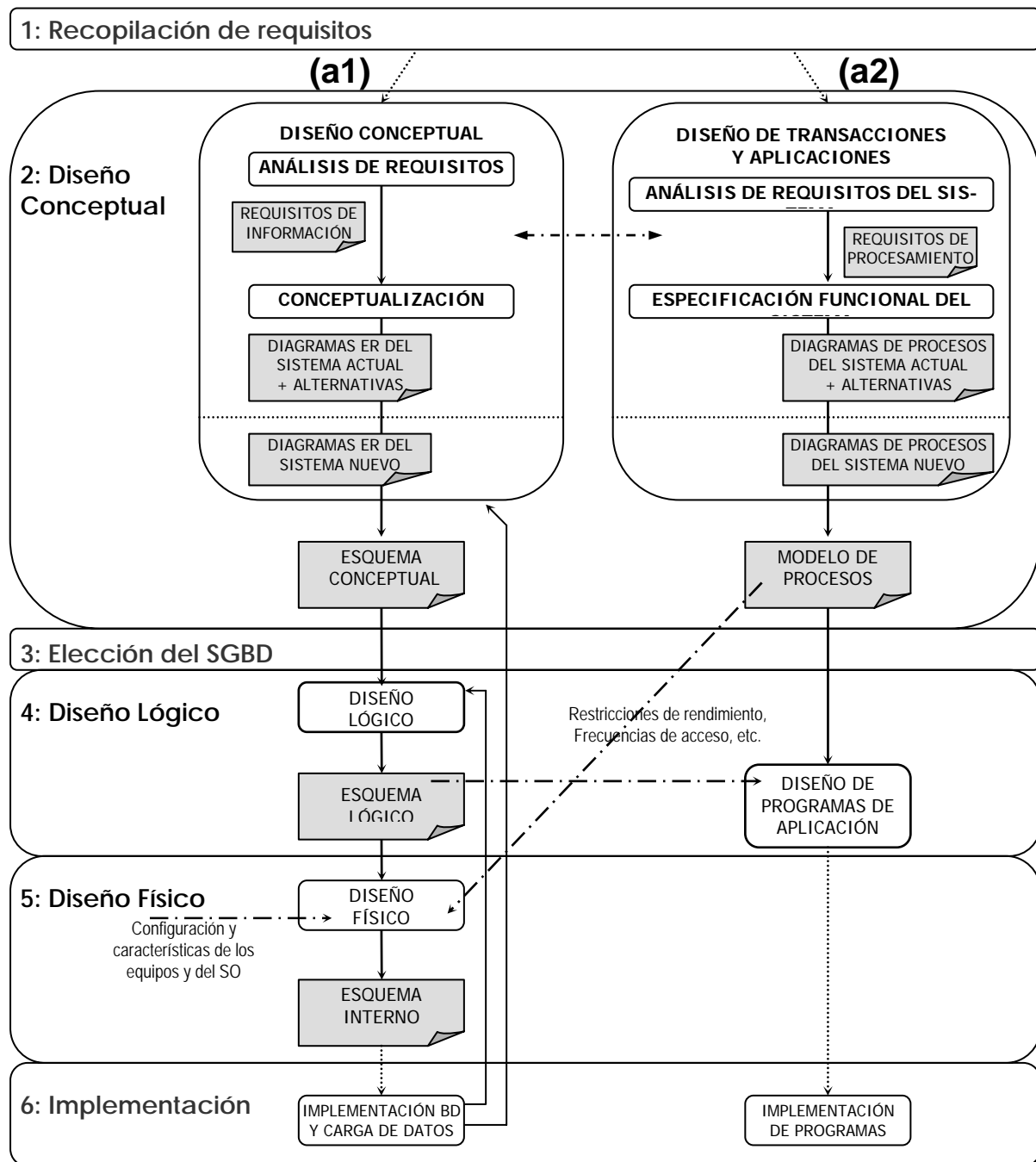
- a1) Diseño del *contenido de datos* y de la *estructura de la base de datos*, y
- a2) Diseño del *procesamiento de la base de datos* y de las *aplicaciones de software*.

En el diagrama que se muestra más adelante, en la figura 1, se puede observar las diferentes etapas del método de diseño, y las fases dentro de cada etapa, que permiten llevar a cabo cada una de estas dos actividades (a1 y a2).

Cada una de las etapas de diseño es en sí un proceso iterativo y como tal, va produciendo refinamientos sucesivos antes de pasar a la siguiente (retroalimentación dentro de cada etapa).

Por otro lado, las diferentes etapas no tienen que realizarse estrictamente de forma secuencial; de hecho, es probable que durante una determinada etapa sea necesario modificar los resultados obtenidos en una etapa anterior (retroalimentación entre etapas).

Con el fin de no complicar en exceso el diagrama mostrado en la figura 1, sólo aparecen reflejados un par de ciclos de retroalimentación, en particular los que afectan a las etapas de Diseño Conceptual y Diseño Lógico como resultado de la implementación y ajuste del sistema. También aparecen algunas de las interacciones entre ambas actividades (a1 y a2).



**Figura 1 Fases de un método de diseño de bases de datos relacionales**

Veamos ahora, de forma resumida, en qué consiste cada una de las etapas que configuran las dos actividades del proceso de diseño de bases de datos.

### (a1) Diseño del contenido de datos y de la estructura de la base de datos

#### **Diseño Conceptual**

Su objetivo es obtener una buena representación (descripción) de los requisitos de información del sistema, es decir, el *esquema conceptual* de la base de datos<sup>9</sup>, que es independiente de los usuarios, de aplicaciones, de SGBDs específicos y de equipos informáticos.

Para describir este esquema se utiliza un modelo de datos de alto nivel (véase *tipos de modelos de datos* en el Tema 1), como el modelo ER (Entidad-Relación, que veremos en el siguiente tema).

#### **Diseño Lógico**

En esta etapa, el *esquema conceptual*, expresado en el modelo de datos de alto nivel, es transformado en un *esquema lógico*<sup>10</sup> expresado en...

- a) el *modelo de datos lógico*, por ejemplo si se decide utilizar el modelo de datos relacional, pero no se elige ningún SGBD en particular. En este caso, se realiza un diseño lógico independiente del sistema, pero dependiente del modelo de datos.
- b) el *modelo de datos del SGBD elegido*, por ejemplo el modelo de datos del SGBD Oracle. En este caso, se realiza un diseño lógico dependiente del sistema.

En términos de la **arquitectura de tres niveles** del SGBD que ya hemos estudiado en el tema anterior, como resultado de esta etapa se obtiene un *esquema conceptual* en el modelo de datos elegido. Además, es habitual que durante esta etapa se realice el diseño de los *esquemas externos* (vistas) para aplicaciones específicas (o grupos de usuarios concretos).

#### **Diseño Físico**

Su objetivo es conseguir la implementación más eficiente posible del esquema lógico.

El resultado de esta etapa es el *esquema interno* (en términos de la arquitectura del SGBD de tres niveles), que depende del SGBD, de la configuración y características de los equipos y del sistema operativo elegidos, y que describe la implementación de la base de datos en el almacenamiento secundario (es decir, las estructuras de almacenamiento físicas, orden de colocación de registros, y caminos de acceso usados para acceder eficientemente a la información almacenada).

Esta etapa está muy relacionada con la de diseño lógico: las decisiones tomadas durante el diseño físico pueden afectar a la estructura del esquema lógico (si es necesario *adaptarlo* para conseguir mayor eficiencia del sistema). Esto no es conveniente, pues el esquema lógico ha de representar los recursos de información de la organización con independencia de aspectos técnicos o de implementación.

Una vez completadas estas etapas, los esquemas lógico y físico (y externos) de la base de datos se expresan (o especifican) utilizando el LDD (lenguaje de definición de datos)<sup>11</sup> del SGBD elegido, con lo que ya puede crearse la base de datos, para posteriormente cargar los datos, realizar pruebas, etc.

---

<sup>9</sup> El *Esquema Conceptual* del que hablamos corresponde al *Esquema Lógico de Datos*, según la terminología de METRICA ver. 2.1.

<sup>10</sup> El *Esquema Lógico* del que hablamos corresponde al *Esquema Técnico de Datos*, según METRICA versión 2.1.

<sup>11</sup> O, como ya hemos visto en el Tema 1, y si el SGBD lo permite, empleará los lenguajes LDD, LDA y LDV.



## (a2) Diseño del procesamiento de la base de datos y de las aplicaciones de software

### Diseño de transacciones y aplicaciones<sup>12</sup>

Toma como punto de partida los requisitos de procesamiento<sup>13</sup>, y produce un **modelo de procesos** (a veces conocido como *esquema funcional*), que es una descripción de alto nivel de las aplicaciones y transacciones desarrolladas dentro de la organización (operaciones, tareas, funciones, procesos...), así como de la información utilizada en cada una de ellas o intercambiada entre varias, etc.

En esta etapa, la base de datos suele ser considerada como un *conjunto de simples depósitos de información*, perdiéndose la visión de los datos como *recurso global* de la organización (visión que sí proporciona el esquema conceptual).

### Diseño de programas de aplicación<sup>14</sup>

En esta etapa, a partir del modelo de procesos, se produce un conjunto de descripciones de alto nivel (*especificaciones*) *del comportamiento de las aplicaciones*, es decir, la estructura de los programas, con qué información (de la base de datos) trabaja cada programa, cómo accede cada programa a los datos, etc.

### Implementación de transacciones y aplicaciones

Esta etapa produce las *especificaciones detalladas* de los programas de aplicación y quizás, el código de los programas, expresado en un lenguaje de programación concreto.

----

Tradicionalmente los métodos de *diseño de bases de datos* se han centrado sobre todo en la primera de estas actividades (*a1*), mientras que el *diseño de software* se ha centrado en la segunda (*a2*), de forma que podía hablarse de diseño o desarrollo **guiado por los datos** (*data-driven development*) o **guiado por los procesos** (*function-driven development*).

Actualmente, en la comunidad de diseñadores de bases de datos e ingenieros de software, existe el consenso de que **ambas** actividades deben realizarse de forma **coordinada**<sup>15</sup>:

1. El modelo de procesos (esquema funcional) y el esquema conceptual deben ser *mutuamente consistentes* (sin conflictos entre ellos) y *completos*, puesto que toda la información requerida por las funciones debe estar representada en el esquema conceptual, y todas las operaciones requeridas por la base de datos deben estar reflejadas en las funciones del sistema.
2. El diseño físico de la base de datos depende de las aplicaciones que van a utilizar los ficheros de la base de datos (su prioridad, su forma y frecuencia de acceso a los datos, sus restricciones de rendimiento, etc.).
3. En el diseño de los programas de aplicación se hace referencia a los elementos que tiene el esquema lógico de la base de datos.

En esta asignatura enfocamos el proceso de diseño desde un punto de vista **guiado por los datos**, puesto que prestaremos especial atención al estudio y aplicación de las etapas de la actividad *a1*. *Diseño del contenido de datos y de la estructura de la base de datos*.

---

<sup>12</sup> Estaría incluida en los módulos *ARS (análisis de requisitos del sistema)* y *EFS (especificación funcional del sistema)* de METRICA v. 2.1.

<sup>13</sup> Los requisitos de procesamiento pueden clasificarse en *funcionales* y *no funcionales*. Los requisitos no funcionales (de rendimiento, frecuencias de acceso, etc.) son los que se utilizan en la fase de *Diseño Físico* de la base de datos.

<sup>14</sup> Estaría incluida en el módulo *DTS (diseño técnico del sistema)* de METRICA v. 2.1

<sup>15</sup> Además, las herramientas de diseño las están combinando cada vez más.

### 2.2.3 Características deseables de un método de diseño

En este apartado comentaremos las propiedades que debería reunir un método de desarrollo para ser considerado un *buen método*, siguiendo el libro [MPM 1999] (capítulo 8).

#### a) Claridad y facilidad de comprensión

Es imprescindible que distintos tipos de personas (usuarios, técnicos de sistemas, analistas, etc.) participen en el proceso de diseño; por tanto, el método debe ser lo suficientemente sencillo para ser entendido por diferentes tipos de usuarios.

#### b) Capacidad de soportar la evolución de los sistemas

*“Diseñar y programar para el cambio”*. Un buen método de diseño soportará la evolución del sistema de información sin traumas, produciendo en sus distintas etapas *esquemas evolutivos*, de forma que cuando cambie el minimundo sea posible modificar los esquemas para que recojan dichos cambios sin tener que realizar de nuevo el diseño completo de la BD. Para conseguir este objetivo es fundamental que el método proporcione bases para una buena documentación del sistema.

#### c) Facilitar la portabilidad

El estándar IEEE-610<sup>16</sup> considera la *portabilidad* como *“la facilidad con la que un sistema o componente puede ser transferido (traslado) de un entorno hardware o software a otro”*; es esencial para conseguir sistemas abiertos.

Un método que pretenda conseguir esquemas “portables”, utilizará los siguientes recursos:

1. Unas *etapas de diseño independientes* que permitan desviarse, en determinados momentos, hacia otro tipo de sistemas. Así, aunque el método estudiado en esta asignatura está orientado al *modelo relacional*, podría aplicarse a otro modelo de datos, pues es posible pasar del esquema conceptual a un esquema en cualquier otro modelo.
2. Una subetapa *diseño lógico estándar*, entre el diseño conceptual y el diseño lógico en el SGBD específico que se va a utilizar. Esta subetapa permitirá disponer de un *esquema relacional general*, que podría traducirse posteriormente al modelo relacional específico (como el de Oracle, DB2, Informix, Ingres, etc.). Así se facilita la migración entre diferentes SGBDR, o entre versiones distintas del mismo producto (como las versiones 7 y 9i de Oracle).

Por otro lado, los propios SGBD comerciales suelen proporcionar productos para diversas plataformas (por ejemplo, Oracle ofrece sus SGBDR para Windows 95/98/NT, Sun SPARC Solaris, etc.), lo cual hace que sea posible portar o transferir las bases de datos de unos entornos a otros.

#### d) Versatilidad respecto a tipos de aplicaciones

Un método no debe estar orientado a un tipo de aplicaciones concreto, sino que debe poder utilizarse en aplicaciones diversas, como la gestión de una biblioteca, de un hospital, de una universidad, etc., o para el diseño de bases de datos estadísticas, científicas o de cualquier otro tipo, aunque, en determinados casos, hubiera que hacer las oportunas adaptaciones.

#### e) Flexibilidad (Independencia de la dimensión (o tamaño) de los proyectos)

Un método debe poder utilizarse tanto en proyectos grandes como pequeños. Para abordar ambos tipos de proyectos se utilizan modelos, herramientas y lenguajes análogos, aunque para proyectos grandes se completarán con otras técnicas (por ejemplo la de *integración de vistas*), mientras que para diseños menos complejos se simplificarán algunas de las etapas del método propuesto.

---

<sup>16</sup> IEEE Std. 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Computer Society, 1999. (Este estándar es la revisión y renombramiento del IEEE-Std-729-1983).

#### f) Rigurosidad

Es conveniente dar un carácter riguroso al método de diseño, apoyándose, siempre que sea posible (como en el caso de la *normalización*) en fundamentos teóricos formales, y dando la máxima rigurosidad a las descripciones (diagramas, lenguajes) utilizadas en el proceso de diseño.

Sin embargo, el método no debe resultar demasiado formal, ya que un excesivo formalismo suele ser rechazado por cierto tipo de usuarios. La experiencia demuestra que es posible no ser demasiado formalista y seguir siendo riguroso.

#### g) Adopción de estándares

Aunque no existen demasiados estándares en el área de las bases de datos, un método debería aplicar todos aquellos que –para la Ingeniería del Software en general y para las bases de datos en particular– recomiendan distintas organizaciones internacionales (como ISO, ACM, OSF, etc.). Por ejemplo, para describir el esquema lógico general relacional, nos basaremos en el estándar SQL de ISO.

#### h) Automatización

Para que un método resulte útil debe poderse automatizar, aplicando herramientas CASE (*Computer Aided Software Engineering*) que soporten todas las fases propuestas para el diseño de la BD.

Nosotros utilizaremos modelos, lenguajes y herramientas muy conocidos, como el modelo Entidad-Relación (MER), el lenguaje SQL-92, los diagramas de dependencias funcionales, etc.

Además, el método se puede seguir con facilidad mediante productos CASE existentes (*System Architect*, por ejemplo).

----

Para terminar este tema, indicar que existen otros enfoques de diseño de bases de datos relacionales, que permiten llegar directamente al esquema relacional partiendo de una *relación universal* que contiene los atributos considerados aisladamente y las restricciones semánticas (dependencias funcionales, sobre todo), es decir se limitan a aplicar la *teoría de la normalización* (que estudiaremos con detalle en un tema posterior).

En cambio, y como veremos a lo largo de esta asignatura, nuestro enfoque consiste en la aplicación de un método de diseño, que permite obtener un esquema de la bases de datos consistente en un conjunto de relaciones, las cuales pueden ser entonces sometidas al proceso de normalización.

---

## BIBLIOGRAFÍA

- [EN 2002] Elmasri, R.; Navathe, S.B.: **Fundamentos de Sistemas de Bases de Datos. 3ª Edición**. Madrid [etc.]: Addison-Wesley, Pearson Educación, 2002. (Capítulo 16)
- [EN 1997] Elmasri, R.; Navathe, S.B.: **Sistemas de bases de datos. Conceptos fundamentales. 2ª Edición**. Wilmington, Delaware, USA: Addison-Wesley Iberoamericana, 1997. (Capítulo 14)
- [MPM 1999] De Miguel, A.; Piattini, M.; Marcos, E.: **Diseño de bases de datos relacionales**. Madrid: Ra-Ma, 1999. (Capítulo 8)
- [MP 1993] De Miguel, A.; Piattini, M.: **Concepción y diseño de bases de datos: del Modelo E/R al Modelo Relacional**. Madrid: Ra-Ma, 1993.
- [CBS 1998] Connolly, T.; Begg C.; Strachan, A.: **Database Systems: A Practical Approach to Design, Implementation and Management. 2ª edición**. Harlow, England: Addison-Wesley, 1998. (Capítulo 4)
- [BCN 1994] Batini, C.; Ceri, S.; Navathe, S.B.: **Diseño conceptual de bases de datos: un enfoque de entidades-interrelaciones**. Wilmington, Delaware, USA: Addison-Wesley Iberoamericana, 1994. (Capítulo 1)

---

## REFERENCIAS

- [S 1988] Sommerville, I. **Ingeniería del Software**. México: Addison-Wesley Iberoamericana, 1988.