2. PROCESO DE CREACIÓN DE BASES DE DATOS

BIBLIOGRAFÍA

[EN97] Elmasri, R.; Navathe, S.B. **Sistemas de bases de datos. Conceptos fundamentales**. Edit. Addison-Wesley Iberoamericana. 2ª Edición, 1997.

[dMP93] De Miguel, A.; Piattini, M. **Concepción y diseño de bases de datos: del Modelo E/R al Modelo Relacional**. Edit. RA-MA, 1993.

2.1 CICLO DE VIDA DE UN S.I. ORIENTADO A BASES DE DATOS.

Introducción

Las bases de datos se han convertido en una parte importante de la gestión de los recursos de información[1] (en general, de los sistemas de información) en muchas organizaciones, 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, el diseño no resulta muy complicado. Pero cuando se diseñan bases de datos medianas o grandes para el sistema de información (SI) de una gran organización, el proceso es complejo pues el SI debe satisfacer las necesidades de muchos usuarios diferentes (de 25 a varios cientos).

Una BD mediana o grande contiene millones de bytes de información y sobre ella se realizan cientos de consultas y programas de aplicación. Ejemplos serían las bases de datos utilizadas en la industria de servicios (reservas de vuelos), las cuales dependen de que sus bases de datos funcionen a la perfección (estén operativas) las 24 horas del día: cientos de transacciones procedentes de terminales locales y remotas acceden a la BD cada minuto.

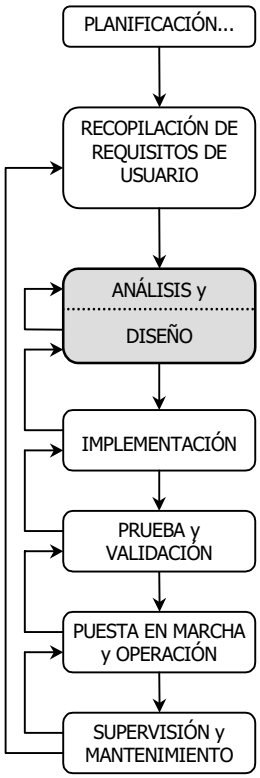
Los sistemas para este tipo de bases de datos son los **sistemas de procesamiento de transacciones**[2]. En estos sistemas es decisivo el promedio de transacciones ejecutadas por minuto, así como el tiempo medio y máximo de respuesta del sistema (a cada transacción). Es imprescindible realizar un adecuado diseño de la base de datos tal que 1) satisfaga las necesidades de procesamiento de la organización (tiempos de respuesta y de procesamiento, espacio de almacenamiento), y 2) permita y facilite la evolución del sistema (como consecuencia de cambios en los requisitos)[3].

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**, es decir: los datos en sí mismos, el SGBD, el hardware y los medios de almacenamiento, el personal que usa los datos (DBA, usuarios finales, etc.), el software de aplicación que accede a los datos y los actualiza, y los programadores que crean estas aplicaciones.

[1] Recursos que participan en la recolección, administración, uso y diseminación de la información.
[2] También sistemas de alto rendimiento.
[3] El coste puede ser muy alto si un sistema grande y complejo es incapaz de evolucionar y se hace necesario sustituirlo por otros productos de SGBD.

Fases del Ciclo de Vida de un Sistema de Información orientado a Bases de Datos[4]



1. *Análisis de factibilidad, definición del sistema y objetivos, plan de trabajo.*
Analizar las posibles áreas de aplicación, estudiar la relación coste / beneficio, establecer prioridades entre las aplicaciones, etc. Por *definición del sistema* se entiende establecer el alcance del sistema de BD, sus usuarios, etc.

2. *Recopilación de requisitos de usuario*
Se recogen los requisitos de los usuarios con el fin de identificar sus problemas y sus necesidades en lo referente a datos, funcionalidades, rendimiento, etc.

3. *Diseño (análisis y diseño)*
Diseño del sistema de 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*
Implementación del sistema de información, carga de datos reales en la base de datos (nuevos u otros ya existentes convertidos al formato adecuado).

5. *Pruebas, validación y ajuste*
Prueba de las aplicaciones (o transacciones, operaciones sobre los datos a través de los programas). Un sistema queda *validado* (es aceptable) cuando satisface los requisitos de los usuarios (especificaciones del comportamiento del sistema y los criterios de rendimiento).

6. *Operación*
Puesta en marcha del sistema. La fase operativa comienza cuando todas las funciones (procesos) del sistema están disponibles y validadas.

7. *Supervisión y mantenimiento*
Durante la fase de operación, el sistema se vigila continuamente. Cuando surgen nuevas necesidades de datos o se requieren nuevas aplicaciones, pasan por todas las fases anteriores hasta, una vez validadas, ser incorporadas al sistema.

El proceso de Diseño de Bases de Datos

[EN97] 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 BD son los siguientes:

a) satisfacer requisitos de *contenido* de información de usuarios y aplicaciones
b) proporcionar una estructuración de los datos natural y fácil de entender
c) apoyar los requisitos de *procesamiento* (objetivos de rendimiento como tiempo de respuesta, tiempo de procesamiento, espacio de almacenamiento...)
d) conseguir un *esquema de BD flexible*, es decir tal que sea posible modificarlo (como consecuencia de cambios en los requisitos del sistema) fácilmente una vez implementada la base de datos.

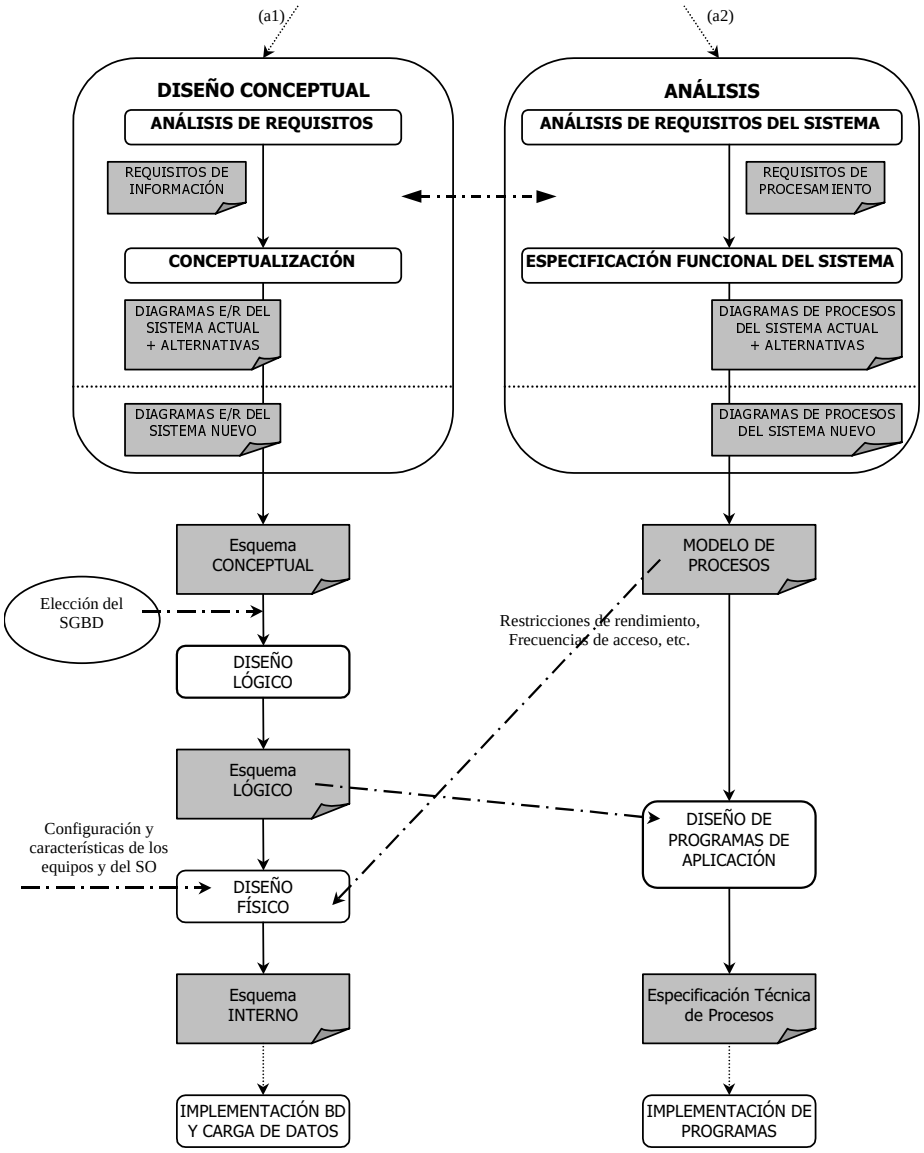
[4] Nota: es necesario remarcar que el Ciclo de Vida que veremos en esta asignatura está enfocado desde la perspectiva del Paradigma de Análisis Estructurado.

Estas son las *fases del diseño de bases de datos*:

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

A veces se considera que las fases *1* y *6* no forman parte del diseño de la BD en sí, sino del ciclo de vida del Sistema de Información más general.

El proceso de diseño consta de **dos** actividades paralelas:
(a1) Diseño del contenido de datos y de la estructura de la base de datos, y
(a2) Diseño del procesamiento de la BD y de las aplicaciones de software[5].



[5] Nota: 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 BD.

(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[6], que es independiente de los usuarios, de aplicaciones, de SGBDs específicos y de equipos informáticos.

Para describir este esquema (el contenido de información de la BD) se utiliza un modelo de datos de alto nivel (ver *tipos de modelos de datos* en el capítulo anterior), como el modelo ER (*Entity/Relationship*, que veremos en el siguiente tema).

Diseño Lógico

En esta fase, el *esquema conceptual*, expresado en el modelo de datos de alto nivel, es transformado en un *esquema lógico*[7] –expresado en …

a) el *modelo de datos del SGBD elegido*, por ejemplo el modelo de datos del SGBD Oracle (diseño lógico dependiente del sistema)
b) el *modelo de datos lógico*, por ejemplo si se decide utilizar algún SGBD relacional, pero no se elige ninguno en particular (diseño lógico independiente del sistema, pero dependiente del modelo de datos)

En términos de la arquitectura del SGBD en tres niveles que ya hemos estudiado, de esta fase se obtiene un *esquema conceptual en el modelo de datos elegido*, además de *esquemas externos* (vistas) para aplicaciones específicas.

Diseño Físico

Su objetivo es conseguir la implementación del esquema lógico más eficiente posible. Esta fase está muy relacionada con la de diseño lógico, pues las decisiones tomadas durante el diseño físico pueden afectar a la estructura del esquema lógico.

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

Una vez completadas estas fases, el esquema lógico y el físico se expresan utilizando el DDL (*data definition language*)[8] del SGBD destino, con lo que ya puede crearse la BD, para posteriormente realizar pruebas, etc.

---

[6] El *Esquema Conceptual* del que hablamos corresponde al *Esquema Lógico de Datos*, según la terminología empleada en METRICA versión 2.1.
[7] El *Esquema Lógico* del que hablamos corresponde al *Esquema Técnico de Datos*, según METRICA versión 2.1.
[8] O, como ya hemos visto, y si el SGBD lo permite, empleará los lenguajes DDL, SDL y VDL.

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

Análisis Funcional del Sistema[9]

Parte de los requisitos de procesamiento y produce un **modelo de procesos** (a veces conocido como *esquema funcional*), que es una descripción de alto nivel de las actividades (operaciones, tareas, funciones...) desarrolladas dentro de la organización y de la información utilizada en cada actividad, intercambiada entre actividades, etc.

En esta fase, 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[10]

Produce descripciones de alto nivel (*especificaciones*) *del comportamiento de las aplicaciones*, es decir, la estructura de los programas, cómo acceden éstos a qué información de la BD, etc.

Implementación

Produce las *especificaciones detalladas* de los programas de aplicación y quizás, el código (expresado en un LP determinado) de los programas.

Tradicionalmente los métodos de diseño de bases de datos se han centrado en una sola de estas actividades (bien *a1*, o *a2*), de forma que podía hablarse de **diseño de bases de datos guiado por los datos** (*data-driven*) o **guiado por los procesos** (*function-driven*).

Actualmente existe el consenso de que debe realizarse **ambas** actividades de forma **coordinada**, de hecho:

1. El modelo de procesos (esquema funcional) y el esquema conceptual deben ser *mutuamente consistentes* (sin conflictos entre ellos) y *completos*, puesto que todos los datos requeridos por las funciones deben estar representados 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 (forma y frecuencia de acceso a los datos, 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, abordaremos el diseño de bases de datos desde un punto de vista **dirigido por los datos**.

[9] Correspondería a las fases ARS y EFS de METRICA v. 2.1
[10] Correspondería a la fase de Diseño Técnico del Sistema de METRICA v. 2.1

2.2 MÉTODOS DE DISEÑO DE BASES DE DATOS

El 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.

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 y mantenimiento, etc.).

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.

A veces, por ejemplo, el diseño de una BD Relacional se ha limitado a aplicar la *teoría de la normalización*[11], cuando debe abarcar muchas otras etapas (como hemos visto: desde su concepción a su implementación).

Concepto y fases de un método de diseño

Concepto de método

Según [dMP93]: "Un método es un conjunto de *modelos de datos*, *lenguajes* y otras *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". Los modelos de datos, los lenguajes de datos y la documentación son herramientas.

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". El esquema obtenido al describir cierto UoD 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 BD.

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 BD–, 2) de adquisición de requisitos, y 3) de diccionario de datos.

[11] La estudiaremos con detalle en un tema posterior de esta asignatura.

No existe un método de diseño de bases de datos “establecido”, debido a la diversidad de opiniones y enfoques existentes dentro de la comunidad de investigadores en el área de las bases de datos.
Aun así, en esta asignatura vamos a estudiar un método de diseño (basado en el propuesto en [dMP93]), que consiste en tres grandes fases.

Fases de un método de diseño de bases de datos

Cada una de las fases del método corresponderá a cada etapa en el proceso de diseño de bases de datos, al que ya nos hemos referido anteriormente:

Diseño Conceptual
Cuyo objetivo es obtener una representación de los requisitos de información de la organización, independiente de usuarios, aplicaciones y de rendimiento o eficiencia del equipo informático, es decir, el *esquema* de datos *conceptual* (EC).

Diseño Lógico
Cuyo objetivo es adaptar (traducir) el EC al modelo de datos de implementación, obteniendo el *esquema lógico.*

Diseño Físico
Cuyo objetivo es conseguir una implementación del esquema lógico lo más eficiente posible.

En posteriores capítulos estudiaremos con detalle cada una de estas fases y las etapas que las constituyen.

Características de un método de diseño

a) **Claridad y comprensibilidad**

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 universo del discurso, 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 la base para una buena documentación del sistema.

c) **Facilitar la portabilidad**

El IEEE considera la *portabilidad* como "*la facilidad con la que un producto de programación puede ser transferido de un sistema informático a otro, o de un entorno 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 diseño *relacional* no habría inconveniente en aplicarlo a otro modelo de datos (como el de red o

CODASYL), pues es posible pasar del esquema conceptual a un esquema en cualquier otro modelo.

2. Una subfase *diseño lógico estándar*, entre el diseño conceptual y el diseño lógico-físico en el SGBD específico que se va a utilizar. Esta subfase 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 6 y 7 de Oracle).

Por otro lado, los propios SGBD comerciales suelen proporcionar sus productos para diversas plataformas diferentes (Oracle, por ejemplo ofrece sus SGBDR para Windows95, Sun SPARC Solaris, Windows NT, 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 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 (como por ejemplo la de integración de vistas), mientras que para diseños menos complejos, se simplificarán algunas de las etapas del método propuesto.

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.

Sin embargo, se procurará en todo momento que el método no resulte 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 de tipo CASE. Estas herramientas deben soportar todas las fases propuestas para el diseño de la BD.

Nosotros utilizaremos modelos, lenguajes y herramientas muy conocidos, como el modelo entidad-interrelación (MER), los diagramas de dependencias funcionales, el lenguaje SQL-92, etc.
Además, el método se puede seguir con facilidad mediante los productos CASE existentes (*System Architect*[12], o *EasyCASE*, por ejemplo).

[12] Herramienta CASE de *Popkin Software & Systems Incorporated*