

Bases de Datos y Sistemas de Información

Grado en Ingeniería Informática

Unidad Didáctica 4: Diseño de Bases de Datos
Relacionales

Parte 1: Conceptos básicos de diseño
(Doc. UD4.1)

Curso 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Índice

1 Introducción	1
2 Metodología de diseño de bases de datos	1
3 Introducción a los modelos de datos.....	3
4 Diseño de bases de datos	3
5 Ejemplo: diseño de la base de datos Docencia.....	3
5.1 Fase 1: Análisis	4
5.2 Fase 2: Diseño	5
5.2.1 Diseño conceptual	5
5.2.2 Diseño lógico	5
5.2.3 Diseño físico	6
6 Relación con la asignatura de Ingeniería del Software	7
7 Bibliografía.....	8

1 INTRODUCCIÓN

En esta unidad didáctica se presenta una visión general del proceso de diseño de una base de datos relacional. Para ello, resulta interesante introducir el concepto de metodología y explicar para qué sirve; también se recordará qué es un modelo de datos como instrumento necesario para diseñar BD y, por último se presentará la metodología de diseño de bases de datos que se propone en esta asignatura definiendo claramente cuáles son sus fases.

2 METODOLOGÍA DE DISEÑO DE BASES DE DATOS

"Una metodología es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de un producto software"; en el caso que nos ocupa, el producto software es una base de datos. Una metodología nos indica las actividades a seguir en el desarrollo de principio a fin de la base de datos, qué es lo que hay que realizar en cada actividad indicando qué se necesita como entrada, qué se produce como salida e incluso quién está involucrado en la actividad.

Una metodología se apoya en *técnicas* y *modelos* y puede estar soportada por *herramientas CASE*¹.

- Las técnicas representan cómo llevar a cabo cada una de las actividades o pasos de los que consta la metodología; en ocasiones estas técnicas son *procedimentales* (secuencia definida de los pasos a realizar en una tarea como en un algoritmo) y en otros casos son *heurísticas* (reglas, recomendaciones o sugerencias a seguir que en ningún caso establecen el proceso exacto de realización de una tarea y que son utilizadas generalmente en tareas con un alto componente creativo).
- Los modelos son los instrumentos que empleamos para representar una determinada realidad; se utilizan en las técnicas para soportar la actividad que llevan a cabo. Los involucrados en el diseño de una base de datos son *modelos de datos* que se verán en el apartado 3.
- Por último, las herramientas CASE dan soporte automatizado a la aplicación de las técnicas de una metodología, así como a los modelos que incorporan. Los entornos CASE no sólo deben automatizar las técnicas aisladas correspondientes a una metodología, sino que también deben soporte a toda la metodología de desarrollo mediante la incorporación de un conductor metodológico que ayude al analista, diseñador o programador a desarrollar su labor en cada actividad definida en la metodología.

Aunque existen distintas metodologías para el diseño de bases de datos en esta asignatura se seguirá la propuesta ya planteada en [MCC94] que considera tres fases en el desarrollo de un SI, correspondiendo la segunda fase al diseño de la base de datos. En esta metodología se muestra en cada fase qué esquema de la base de datos se está generando de acuerdo con la arquitectura ANSI/SPARC estudiada en la unidad didáctica 3.

- **Fase 1: Análisis**
 - **Investigación:** esta actividad tiene como objetivo "descubrir" el conjunto de requisitos de información, por una parte, y de proceso por otra, que la organización necesita para cumplir sus fines. (Técnicas heurísticas).
- **Fase 2: Diseño**
 - **Diseño conceptual (o modelado conceptual):** esta actividad tiene como objetivo obtener una representación de la realidad que capture las propiedades estáticas y dinámicas de la misma necesarias para satisfacer los requisitos recogidos en la actividad anterior. En este proceso se debe aprehender y conceptualizar el mundo exterior transformándolo en un conjunto de ideas y definiciones que supongan una imagen fiel del comportamiento del mundo real. El resultado de este diseño es por un lado un

¹ Computer Aided Software Engineering

esquema conceptual, que representa las propiedades estáticas, y por otro la *especificación de transacciones* que representa las propiedades dinámicas. Para definir el esquema conceptual existen diferentes propuestas en la literatura. Tradicionalmente, se ha utilizado el diagrama entidad relación [Chen76]. Una descripción detallada del mismo lo podemos encontrar en [EN02]. En esta asignatura se utilizará el diagrama de clases del UML para la modelización conceptual de la parte estática, siguiendo la propuesta de [CB10]. Para la modelización dinámica a nivel conceptual no se presenta ninguna propuesta, sólo se ilustrará mediante un ejemplo.

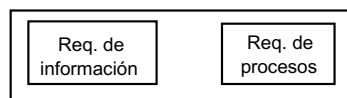
- **Diseño lógico:** esta actividad tiene como objetivo la traducción del esquema conceptual obtenido en la fase anterior en términos del modelo de datos en que se base el sistema de gestión de bases de datos que se vaya a utilizar. Esta actividad de diseño produce un *esquema lógico* y un conjunto de *transacciones* que expresan, a este nivel, la parte estática y dinámica respectivamente. (Técnicas procedimentales).
- **Diseño físico:** teniendo en cuenta detalles de representación física de los datos, y atendiendo a criterios de eficiencia se eligen estructuras de almacenamiento y caminos de acceso específicos para que las aplicaciones que acceden a la información contenida en los ficheros de la base de datos tengan un buen rendimiento. El resultado de esta tarea es el *esquema físico*. (Técnicas heurísticas).

- **Fase 3: Implantación**

Esta fase supone la incorporación de la base de datos y de las aplicaciones desarrolladas a la organización.

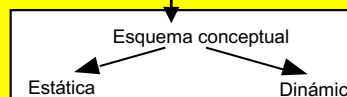
1ª FASE: Análisis

Investigación



2ª FASE: Diseño

Diseño conceptual

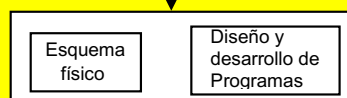


Diseño lógico



SGBD

Diseño físico



3ª FASE: Implantación



En algunos textos se considera que el diseño lógico consta de dos partes, en primer lugar, el *diseño lógico estándar* en el que se utilizan las estructuras del modelo de datos elegido y el *diseño lógico específico* en el que, el resultado del diseño estándar es ajustado al sistema de gestión de bases de datos comercial que se haya elegido. Cuanto más fiel sea este sistema al modelo de datos en que se basa, menor será el trabajo a realizar en el diseño específico.

Tanto en el diseño conceptual como en el diseño lógico es necesario utilizar un modelo de datos para representar la parcela del mundo real (también llamada *Universo del Discurso*) que se quiere modelar. En el apartado siguiente se realiza una introducción a este concepto.

3 INTRODUCCIÓN A LOS MODELOS DE DATOS

Un *modelo de datos* es una herramienta intelectual (conjunto de conceptos y reglas) que permite representar las propiedades estáticas y dinámicas de la parcela del mundo real que es objeto de estudio.

Independientemente de las particularidades de cada modelo de datos, todos ellos deben proporcionar conceptos que permitan representar las siguientes propiedades:

- Propiedades estáticas: objetos de información (clases), propiedades de las clases (atributos), relaciones entre clases (asociaciones) y restricciones sobre las clases o sus asociaciones.
- Propiedades dinámicas: operaciones sobre las clases o sus asociaciones, relaciones entre operaciones (transacciones), y restricciones sobre la evolución de las clases y sus asociaciones.

Los modelos de datos se diferencian entre sí en los conceptos que proporcionan y, en el formalismo utilizado para su definición.

La representación de una parcela del mundo real utilizando un modelo de datos consta de dos partes estrechamente relacionadas: el *esquema conceptual* que es la descripción de las propiedades estáticas y la *especificación de transacciones* que es la descripción de las propiedades dinámicas.

Conectando el concepto de modelo de datos con los conceptos vistos en el apartado anterior, un conjunto de datos estructurados según cierto esquema es una base de datos, y un sistema de gestión de bases de datos es el software que proporciona los lenguajes asociados a un modelo de datos: el lenguaje de definición de esquemas y el lenguaje de manipulación de datos (consulta y actualización).

Los modelos de datos han experimentado una larga evolución, en la cual se pueden distinguir cuatro generaciones:

- Modelos de datos primitivos (basados en ficheros).
- Modelos de datos clásicos. Son los modelos en los que se han basado los sistemas de gestión de bases de datos. Estos sistemas se pueden clasificar en tres grandes familias, correspondientes a los tres modelos de datos clásicos: sistemas jerárquicos (modelo jerárquico), sistemas en red (modelo red), sistemas relacionales (modelo relacional).
- Modelo Orientado a Objetos. Extensión del modelo relacional con características de la orientación a objetos.
- Modelos de datos semánticos. Surgen con la intención de aumentar la capacidad expresiva de los modelos clásicos; para ello, incorporan conceptos y mecanismos de abstracción que permiten modelar la realidad de una forma más natural. Hasta el momento han sido utilizados fundamentalmente como herramientas para el diseño conceptual de la base de datos.

4 DISEÑO DE BASES DE DATOS

Una vez presentada genéricamente en el apartado 2 la metodología de diseño y teniendo presente el concepto de modelo de datos visto en el apartado 3, se puede ya centrar claramente el contenido de la unidad didáctica 4.

- UD4.2: Diseño conceptual con el diagrama de clases de UML.
- UD4.3: Diseño lógico relacional (estándar). Se realizará la transformación del esquema conceptual, obtenido en la fase de diseño conceptual, al modelo relacional de datos, que es el modelo en el que se basan los mejores sistemas de gestión de bases de datos hoy en día.

El diseño físico de una base de datos relacional tiene como objetivo mejorar el rendimiento de una base de datos relacional. Este no es un objetivo de esta asignatura, por lo que solo se darán algunas ideas intuitivas al respecto.

5 EJEMPLO: DISEÑO DE LA BASE DE DATOS DOCENCIA

En la unidad didáctica 1 ya se describió el SI de docencia para el que quería diseñarse una base de

datos. Aquí se trabaja ese mismo ejemplo mostrando los resultados obtenidos en cada una de las fases de diseño.

5.1 Fase 1: Análisis

El Vicerrectorado de Ordenación Académica de una universidad ha decidido crear un sistema de información para la gestión del Plan de Ordenación Docente (POD) del curso académico.

El POD se diseña al principio de cada curso académico, y contiene la planificación de la docencia de todos los estudios que se imparten en la universidad durante el curso.

Algunos datos sobre la estructura de la universidad española que son relevantes para la construcción de este sistema son:

- El contenido de los estudios conducentes a la obtención de un título académico se define en el plan de estudios de la titulación. Los estudios se organizan en asignaturas (materias) valoradas con un número de créditos teóricos (de aula) y prácticos (de laboratorio), lo que determina el número de horas lectivas de la asignatura.
- Las asignaturas de un plan de estudios se organizan en cursos académicos (año académico) divididos en semestres.
- Para el cumplimiento de sus funciones, la universidad se organiza en centros y departamentos.
- Los centros son responsables de organizar la docencia de los estudios de una o varias titulaciones, lo que significa realizar las siguientes tareas: definir los grupos (de teoría y de prácticas) de cada asignatura, organizar los horarios, gestionar la matrícula de los alumnos, distribuir los alumnos por grupos, etc. Por ejemplo, la ETSInf es responsable de organizar la enseñanza de los estudios del Grado en Ingeniería Informática.
- Los departamentos son los responsables directos de impartir la docencia; en ellos se agrupan los profesores pertenecientes a varias materias afines. Algunos departamentos de la universidad son: Matemática Aplicada (DMA), Física Aplicada (DFA), Sistemas Informáticos y Computación (DSIC), etc. Cada departamento tiene asignada la docencia de asignaturas relacionadas con sus áreas de conocimiento; estas asignaturas pueden pertenecer al plan de estudios de cualquier titulación de la universidad. Por ejemplo, el DSIC imparte la docencia de las asignaturas de programación en todos los centros de la universidad.

Antes de esta iniciativa del vicerrectorado, cada centro y cada departamento disponía de su propio sistema de información independiente; por ello, y de acuerdo al carácter integrador de la tecnología de bases de datos, el sistema que se cree, concretamente la base de datos que se diseñe, deberá integrar toda la información sobre la docencia en la universidad, permitiendo al mismo tiempo que los centros y los departamentos sigan teniendo la perspectiva de los datos que les resulte más cómoda para el cumplimiento de sus funciones. A continuación, se presentan las perspectivas de centros y departamentos.

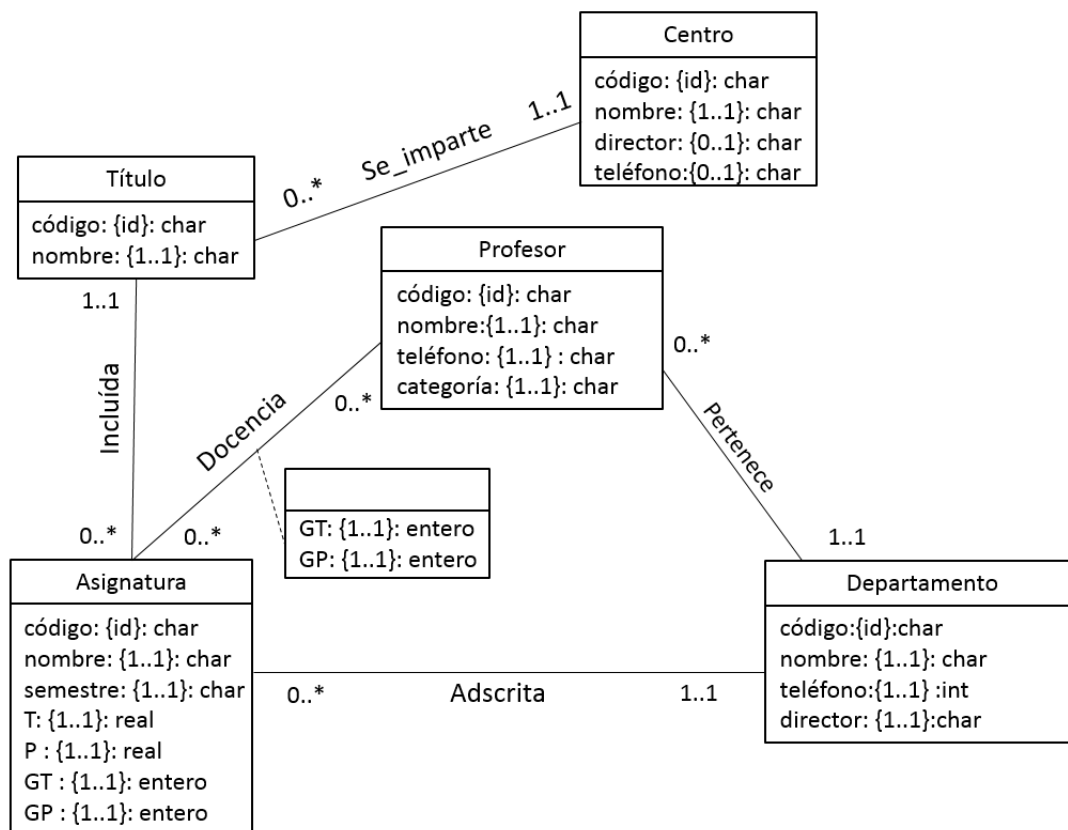
- Perspectiva de un centro: Desde el punto de vista de un centro, la información que necesita para realizar sus funciones es la relacionada con las asignaturas de los planes de estudios de las titulaciones que imparte. De cada una de estas asignaturas necesita conocer: el código, el nombre, el curso y semestre en el que se imparte, el departamento encargado de su docencia, el número de créditos de teoría y de prácticas asignados en el plan de estudios, el número de grupos de teoría y de prácticas creados para el presente curso académico, los profesores encargados de la docencia de la asignatura, y, para cada uno de ellos, el número de grupos de teoría y de prácticas asignados en ella. Para simplificar, se omite la distribución de los profesores de una asignatura en los grupos creados.
- Perspectiva de un departamento: La información que necesita un departamento para realizar sus funciones es la relacionada con las asignaturas de cuya docencia es responsable, así como la relacionada con los profesores que son miembros del departamento. De cada asignatura necesita conocer: el código, el nombre, el curso y semestre, la titulación a la que pertenece, el centro en el que se imparte, el número de créditos de teoría y de prácticas, la cantidad de grupos de teoría y de prácticas creados para el presente curso académico, y los profesores del

departamento asignados a la asignatura, con el número de grupos a su cargo. De cada profesor necesita conocer: el nombre, el teléfono, la categoría y las asignaturas asignadas con el número de grupos que debe impartir.

5.2 Fase 2: Diseño

5.2.1 Diseño conceptual

La parte estática del SI a nivel conceptual se representa gráficamente usando el diagrama de clases del UML:



La parte dinámica a nivel conceptual estaría representada un conjunto de transacciones con operaciones sobre los objetos del diagrama de clases que permitieran representar cómo cambia la base de datos para capturar los cambios que se dan en el mundo real. Algunos ejemplos serían:

- Para añadir un nuevo profesor:
 Transacción Insertar_profesor
 Insertar en Profesor
 Insertar en Pertenece
- Para añadir una nueva asignatura:
 Transacción Insertar_asignatura
 Insertar en Asignatura
 Insertar en Adscrita
 Insertar en Incluida

5.2.2 Diseño lógico

La parte estática, a nivel lógico, se representa mediante un conjunto de esquemas de relación tal como se ha estudiado en la unidad didáctica 1:

```

Centro(código:char(3), nombre:char(50), director:char(50), teléfono:char(8))
  CP: {código}
  VNN: {nombre}
  
```

```
Departamento(código:char(3), nombre:char(50), director:char(50), teléfono:char(8))
  CP:{código}
  VNN:{nombre, director, teléfono}
```

```
Título(código:char(3), nombre:char(50), centro:char(3))
  CP:{código}
  VNN:{nombre}
  CAj:{centro}→Centro(código) Borrado restrictivo
                                Modificación en cascada
```

```
Profesor(código:char(3), nombre:char(50), teléfono:char(8), categoría:char(15),
         dpto:char(3))
  CP:{cod_pro}
  VNN:{nombre, dpto}
  CAj:{dpto}→Departamento(código) Borrado restrictivo
                                    Modificación en cascada
```

```
Asignatura(cod_asg:char(5), nombre:char(50), semestre:char(2), T:real, P:real,
           GT:entero, GP:entero, título:char(3), dpto:char(3))
  CP:{cod_asg}
  VNN:{nombre, semester, T, P, título, dpto}
  CAj:{título}→Título(código) Borrado restrictivo
                                Modificación en cascada
  CAj:{dpto}→Departamento(código) Borrado restrictivo
                                    Modificación en cascada
```

```
Docencia(cod_pro:char(3), cod_asg:char(5), GT:smallint, GP:smallint)
  CP:{cod_pro, cod_asg}
  CAj:{cod_pro}→Profesor(código) Borrado en cascada
                                    Modificación en cascada
  CAj:{cod_asg}→Asignatura(código) Borrado restrictivo
                                    Modificación en cascada
```

La parte dinámica a este nivel sería una traducción de las transacciones diseñadas a nivel conceptual que incluirían ya operaciones de SQL:

- Para añadir un nuevo profesor:
 TRANSACCIÓN Insertar_profesor (códX, nomX, telX, catX, depX)
 INSERT INTO Profesor
 VALUES (códX, nomX, telX, catX, depX)
- Para añadir una nueva asignatura:
 TRANSACCIÓN Inserta_asignatura (codX, nomx, semX, TX, PX, GTX, GPX, titX, depX)
 INSERT INTO Asignatura
 VALUES (codX, nomx, semX, TX, PX, GTX, GPX, titX, depX)

5.2.3 Diseño físico

En el diseño físico, se definiría, para cada una de las relaciones definidas a nivel lógico, el tipo de fichero que se desea utilizar y caminos de acceso alternativos (definición de índices).

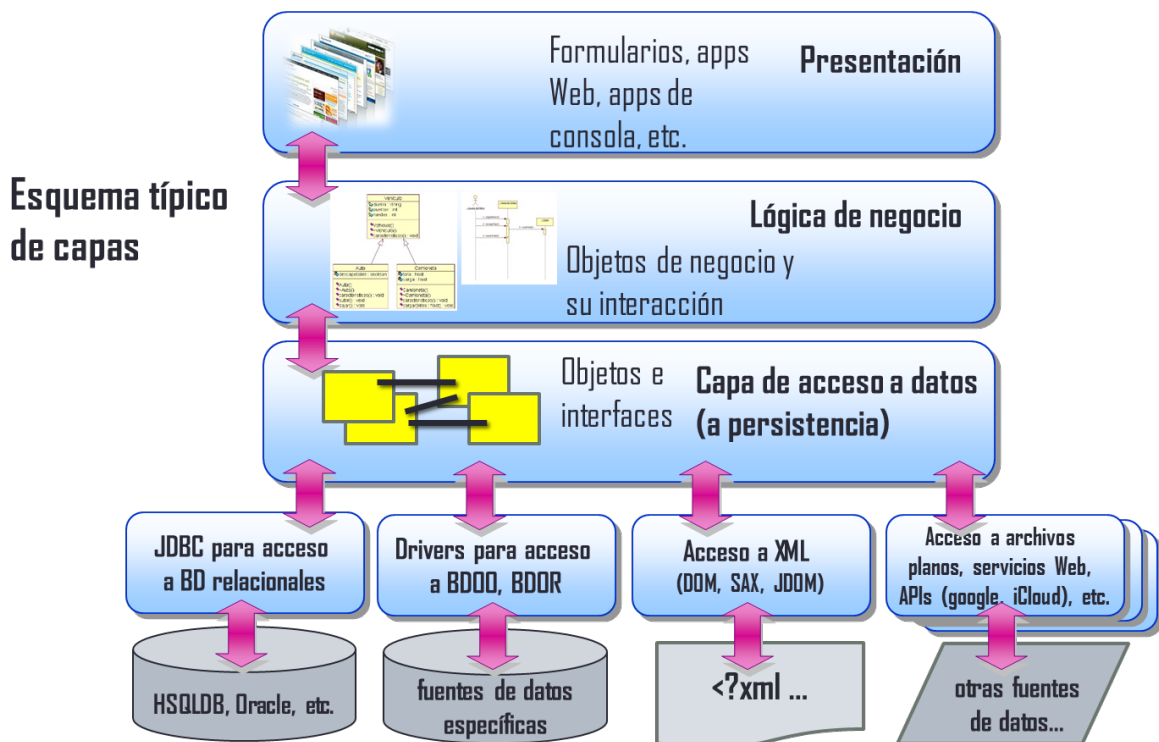
- Centro:
 - Fichero secuencial.
- Título:
 - Fichero indizado por código.
 - Índice sobre *centro*.
- Departamento:
 - Fichero secuencial.

- Índice sobre *nombre*.
- Profesor:
 - Fichero indizado por *código*.
 - Índice sobre *nombre*.
- Asignatura:
 - Fichero indizado por *código*.
 - Índice sobre *nombre*.
- Departamento:
 - Fichero secuencial.
 - Índice sobre *nombre*.
- Docencia:
 - Fichero indizado por *cod_asg*;
 - Índice sobre *cod_pro*.

6 RELACIÓN CON LA ASIGNATURA DE INGENIERÍA DEL SOFTWARE

En la asignatura de ISW se presenta una arquitectura del software en tres capas como se muestra en la figura de más abajo².

Como puede verse en la figura, la metodología de desarrollo de software que se estudia en esa asignatura utiliza el diagrama de clases del UML para definir la lógica de negocio y es independiente de cómo se hagan persistente los datos en memoria secundaria. En el caso de que esa persistencia se realice a través de una base de datos relacional, la definición de la capa de acceso a datos será más sencilla cuando el esquema relacional se haya obtenido a partir de ese diagrama de clases (la transformación de un diagrama de clases en un esquema lógico-relacional se estudia en la UD4.3).



² Esta figura nos la han proporcionado los profesores de ISW.

7 BIBLIOGRAFÍA

- [BRJ06] Booch, G.; Rumbaugh, J.; Jacobson, I.; El lenguaje unificado de modelado (2ª edición). Pearson, Addison Wesley, 2006.
- [CB10] Thomas M. Connolly, Carolyn E. Begg; Database Systems: A Practical Approach to Design, Implementation and Management, 5/E, Addison-Wesley, 2010.
- [CCM03] Celma, M.; Casamayor, J. C.; Mota, L.; Bases de datos relacionales. Pearson, Prentice Hall, 2003.
- [Chen76] Peter Pin-Shan Chen. The Entity-Relation Model- Toward a Unified View of Data. ACM Transactions on Database Systems, Vol. 1, nº 1, March 1976, pages 9-36.
- [EN02] Ramez Elmasri, Shamkant B. Navathe. Fundamentos de sistemas de bases de datos. Addison-Wesley 2002)
- [Lar04] Larman, C.; UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado (2ª edición). Pearson, Prentice Hall, 2004
- [MCC94] Mota, L.; Celma, M.; Casamayor, J. C.; Bases de datos relacionales: teoría y diseño SPUPV 767.94, 1994.
- [Oli01] Olivé, A.; Modelització conceptual de sistemes d'informació. Edicions UPC. 2001.
- [SR07] Stevens, P.; Pooley, R.; Utilización del UML en Ingeniería del Software con objetos y componentes (2ª edición). Pearson, Addison Wesley, 2007.