

# Diseño Conceptual de Bases de Datos

guía de aprendizaje

**Autor:** Jorge Sánchez ([www.jorgesanchez.net](http://www.jorgesanchez.net)) año 2004  
e-mail: <mailto:info@jorgesanchez.net>



Este trabajo está protegido bajo una licencia de **Creative Commons** del tipo **Attribution-NonCommercial-ShareAlike**.

Para ver una copia de esta licencia visite:

<http://creativecommons.org/licenses/by-nc-sa/2.0/>

o envíe una carta a:

**Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.**





C O M M O N S D E E D

Los contenidos de este documento están protegidos bajo una licencia de **Creative Commons** del tipo **Attribution-Noncomercial-Share Alike**. Con esta licencia:

**Eres libre de:**

- Copiar, distribuir y mostrar este trabajo
- Realizar modificaciones de este trabajo

**Bajo las siguientes condiciones:**



**Attribution (Reconocimiento)**. Debe figurar siempre el autor original de este trabajo



**Noncommercial (No comercial)**. No puedes utilizar este trabajo con propósitos comerciales.



**Share Alike (Compartir igual)**. Si modificas, alteras o construyes nuevos trabajos a partir de este, debes distribuir tu trabajo con una licencia idéntica a ésta

- Si estas limitaciones son incompatible con tu objetivo, puedes contactar con el autor para solicitar el permiso correspondiente
- No obstante tu derecho a un uso justo y legítimo de la obra, así como derechos no se ven de manera alguna afectados por lo anteriormente expuesto.

Esta nota no es la licencia completa de la obra, sino una traducción del resumen en formato comprensible del texto legal. La licencia original completa (jurídicamente válida y pendiente de su traducción oficial al español) está disponible en

<http://creativecommons.org/licenses/by-nc-sa/2.0/legalcode>



# índice

<b>índice</b> .....	<b>5</b>
<b>bases de datos</b> .....	<b>7</b>
qué es una base de datos .....	7
ventajas de las bases de datos .....	8
desventajas .....	9
sistema gestor de bases de datos.....	9
niveles ANSI/SPARC .....	12
<b>modelado de datos</b> .....	<b>15</b>
modelos de datos .....	15
clasificación de los modelos de datos .....	15
<b>modelo entidad relación</b> .....	<b>17</b>
introducción .....	17
entidades .....	17
relaciones .....	18
atributos.....	21
entidades <i>is a</i> .....	22
pasos para el diseño .....	23
<b>apéndice: términos técnicos</b> .....	<b>25</b>



# bases de datos

## qué es una base de datos

---

### el problema de los datos

---

Todas las empresas requieren almacenar información. Desde siempre lo han hecho. La información puede ser de todo tipo. Cada elemento informativo (nombre, dirección, sueldo, etc.) es lo que se conoce como **dato** (en inglés **data**).

Las soluciones utilizadas por las empresas para almacenar los datos son diversas. Antes de la aparición de la informática se almacenaban en ficheros con cajones y carpetas y fichas. Tras la aparición de la informática estos datos se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (a veces en archivos binarios, o en hojas de cálculo, ...).

Además las empresas requieren utilizar aplicaciones informáticas para realizar tareas propias de la empresa a fin de mecanizar a las mismas. Estas aplicaciones requieren manejar los datos de la empresa.

En los inicios de la era informática, cada programa almacenaba y utilizaba sus propios datos de forma un tanto caótica. La ventaja de este sistema (la única ventaja), es que los procesos eran independientes por lo que la modificación de uno no afectaba al resto. Pero tiene grandes inconvenientes:

- ⊙ Coste de almacenamiento elevado
- ⊙ Datos redundantes (se repiten continuamente)
- ⊙ Probabilidad alta de inconsistencia en los datos
- ⊙ Dificil modificación en los datos y facilidad de problemas de inconsistencia al realizar esas modificaciones (ya que es difícil que esa modificación afecte a todos los datos)

Lógicamente la solución a este problema es hacer que todas las aplicaciones utilicen los mismos datos. Esto provoca que los datos deban estar mucho más protegidos y controlados. Además los datos forman una estructura física y funcional que es lo que se conoce como **base de datos**.

De esta forma una base de datos es una serie de datos relacionados que forman una **estructura lógica**, es decir una estructura reconocible desde un programa informático. Esa estructura no sólo contiene los datos en sí, sino la forma en la que se relacionan.

Las bases de datos empiezan a aparecer en los años 60 y triunfan en los años setenta y ochenta.

### sistema de bases de datos

---

Un sistema de bases de datos sirve para integrar los datos. Lo componen los siguientes elementos:

- ⊙ **Hardware**. Máquinas en las que se almacenan las bases de datos. Incorporan unidades de almacenamiento masivo para este fin.

## Diseño conceptual de bases de datos

### bases de datos

---

- ⦿ **Software.** Es el sistema gestor de bases de datos. El encargado de administrar las bases de datos.
- ⦿ **Datos.** Incluyen los datos que se necesitan almacenar y los **metadatos** que son datos que sirven para describir lo que se almacena en la base de datos.
- ⦿ **Usuarios.** Personas que manipulan los datos del sistema. Hay tres categorías:
  - ⋄ **Usuarios finales.** Aquellos que utilizan datos de la base de datos para su trabajo cotidiano que no tiene por qué tener que ver con la informática. Normalmente no utilizan la base de datos directamente, si no que utilizan aplicaciones creadas para ellos a fin de facilitar la manipulación de los datos. Estos usuarios sólo acceden a ciertos datos.
  - ⋄ **Desarrolladores.** Analistas y programadores encargados de generar aplicaciones para los usuarios finales.
  - ⋄ **Administradores.** También llamados DBA (*Data Base Administrator*), se encargan de gestionar las bases de datos.

Hay que tener en cuenta que las necesidades de los usuarios son muy diferentes en función del tipo de usuario que sean: a los finales les interesa la facilidad de uso, a los desarrolladores la potencia y flexibilidad de los lenguajes incorporados del sistema de bases de datos, a los administradores herramientas de gestión avanzada para la base de datos.

### estructura de una base de datos

---

Las bases de datos están compuestas (como ya se han comentado), de **datos** y de **metadatos**. Los metadatos son datos (valga la redundancia) que sirven para especificar la estructura de la base de datos; por ejemplo qué tipo de datos se almacenan (si son texto o números o fechas ...), qué nombre se le da a cada dato (nombre, apellidos,...), cómo están agrupados, cómo se relacionan,....

De este modo se producen dos visiones de la base de datos:

- ⦿ **Estructura lógica.** Indica la composición y distribución teórica de la base de datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando.

Es una estructura que permite idealizar a la base de datos. Sus elementos son objetos, entidades, nodos, relaciones, enlaces,... que realmente no tienen presencia real en la física del sistema. Por ello para acceder a los datos tiene que haber una posibilidad de traducir la estructura lógica en la estructura física.
- ⦿ **Estructura física.** Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos (en los metadatos).

### ventajas de las bases de datos

---

- ⦿ **Independencia de los datos y los programas y procesos.** Esto permite modificar los datos sin modificar el código de las aplicaciones.



- ⊙ **Menor redundancia.** No hace falta tanta repetición de datos. Aunque, sólo los buenos diseños de datos tienen poca redundancia.
- ⊙ **Integridad de los datos.** Mayor dificultad de perder los datos o de realizar incoherencias con ellos.
- ⊙ **Mayor seguridad en los datos.** Al limitar el acceso a ciertos usuarios.
- ⊙ **Datos más documentados.** Gracias a los metadatos que permiten describir la información de la base de datos.
- ⊙ **Acceso a los datos más eficiente.** La organización de los datos produce un resultado más óptimo en rendimiento.
- ⊙ **Menor espacio de almacenamiento.** Gracias a una mejor estructuración de los datos.

## desventajas

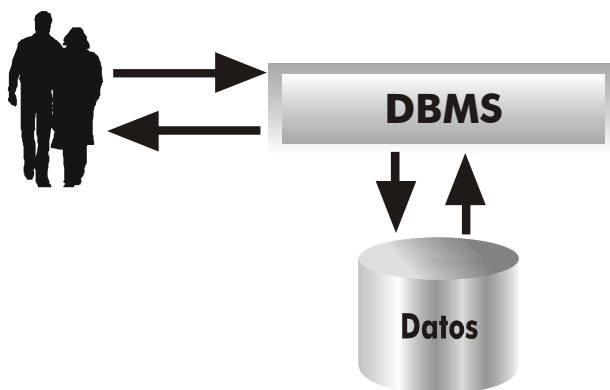
---

- ⊙ **Instalación costosa.** El control y administración de bases de datos requiere de un software y hardware poderoso.
- ⊙ **Requiere personal cualificado.** Debido a la dificultad de manejo de este tipo de sistemas.
- ⊙ **Implantación larga y difícil.** Debido a los puntos anteriores. La adaptación del personal es mucho más complicada y lleva bastante tiempo.
- ⊙ **Ausencia de estándares reales.** Lo cual significa una excesiva dependencia hacia los sistemas comerciales del mercado. Aunque hay una buena parte de esta tecnología aceptada como estándar de hecho.

## sistema gestor de bases de datos

---

Un sistema gestor de bases de datos o **SGBD** (aunque se suele utilizar más a menudo las siglas **DBMS** procedentes del inglés, *Data Base Management System*) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.



**Ilustración 1, Esquema del funcionamiento y utilidad de un sistema gestor de bases de datos**

## Diseño conceptual de bases de datos

### bases de datos

---

El éxito del DBMS reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios. Entre las herramientas que proporciona están:

- ⊙ Herramientas para la creación y especificación de los datos. Así como la estructura de la base de datos.
- ⊙ Herramientas para administrar y crear la estructura física requerida en las unidades de almacenamiento.
- ⊙ Herramientas para la manipulación de los datos de las bases de datos, para añadir, modificar, suprimir o consultar datos.
- ⊙ Herramientas de recuperación en caso de desastre
- ⊙ Herramientas para la creación de copias de seguridad
- ⊙ Herramientas para la gestión de la comunicación de la base de datos

### funciones de un DBMS

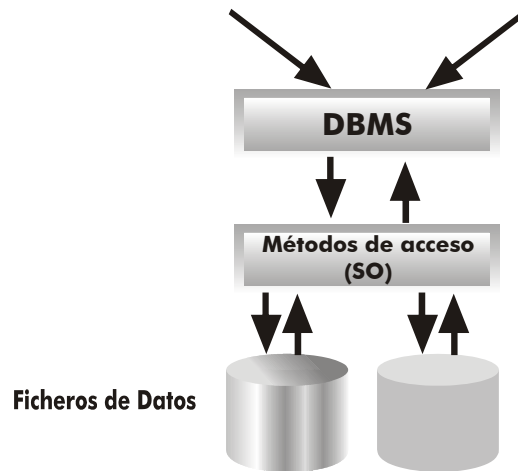
---

- 1 > **Función de descripción.** Sirve para describir los datos, sus relaciones y sus condiciones de acceso e integridad. Además del control de vistas de usuarios y de la especificación de las características físicas de la base de datos. Para poder realizar todas estas operaciones se utiliza un **lenguaje de definición de datos** o **DDL**.
- 2 > **Función de manipulación.** Permite buscar, añadir, suprimir y modificar datos de la base de datos. El DBMS proporciona un **lenguaje de manipulación de datos (DML)** para realizar esta función.
- 3 > **Función de control.** Incorpora las funciones que permiten una buena comunicación con la base de datos. Además proporciona al DBA los procedimientos necesarios para realizar su labor.

## funcionamiento de los DBMS

Los datos son responsabilidad del DBMS, por lo que cualquier acceso debe ser realizado por éste. Lógicamente el DBMS va a acabar comunicándose con el Sistema Operativo ya que el acceso a los ficheros de datos implica utilizar funciones del sistema operativo.

En la página siguiente se observa cómo se produce la interacción completa entre un proceso de usuario y un sistema gestor de bases de datos. Los pasos explicados del esquema son:



**Ilustración 2, Esquema del acceso a los datos de un sistema gestor de base de datos**

- 1> El proceso lanzado por el usuario llama al DBMS indicando la porción de la base de datos que se desea tratar
- 2> El DBMS traduce la llamada a términos del esquema lógico de la base de datos. Accede al esquema lógico comprobando derechos de acceso y la traducción física
- 3> El DBMS obtiene el esquema físico
- 4> El DBMS traduce la llamada a los métodos de acceso del Sistema Operativo que permiten acceder a los datos requeridos
- 5> El Sistema Operativo accede a los datos tras traducir las órdenes dadas por el DBMS
- 6> Los datos pasan del disco a una memoria intermedia o buffer. En ese buffer se almacenarán los datos según se vayan recibiendo
- 7> Los datos pasan del buffer al área de trabajo del usuario (ATU) del proceso del usuario.
- 8> El DBMS devuelve indicadores en los que manifiesta si ha habido errores o advertencias a tener en cuenta. Esto se indica al área de comunicaciones del proceso de usuario. Si las indicaciones son satisfactorias, los datos de la ATU serán utilizables por el proceso de usuario.

## Diseño conceptual de bases de datos

bases de datos

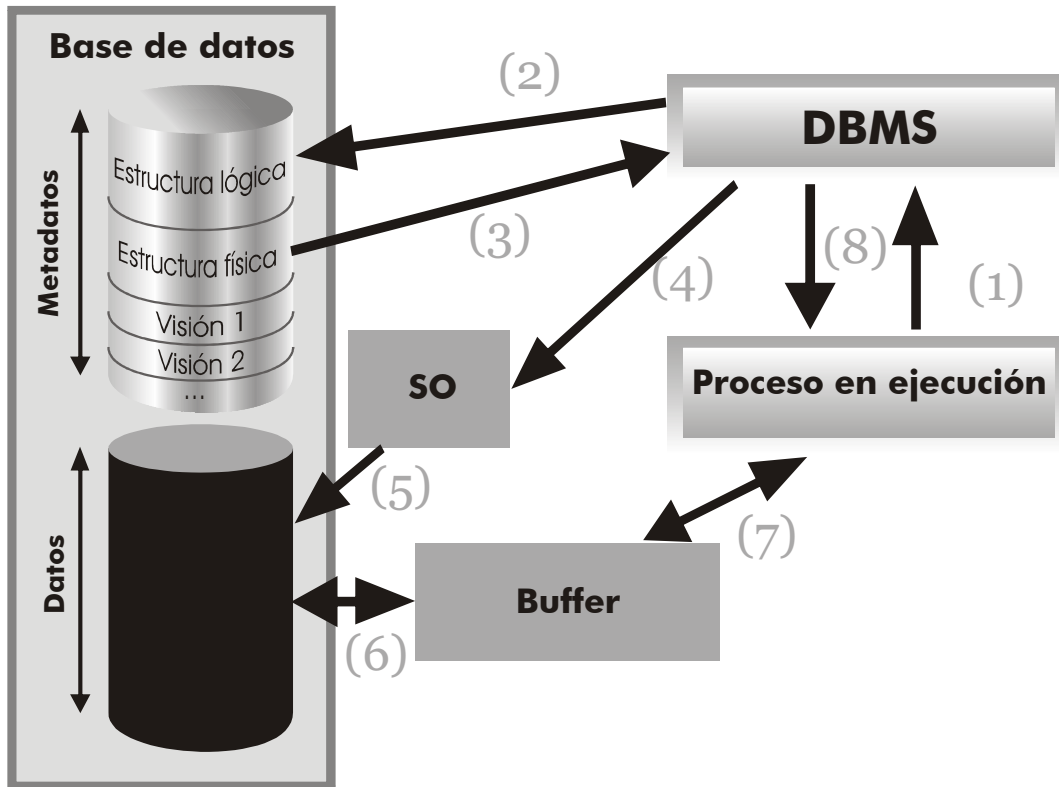


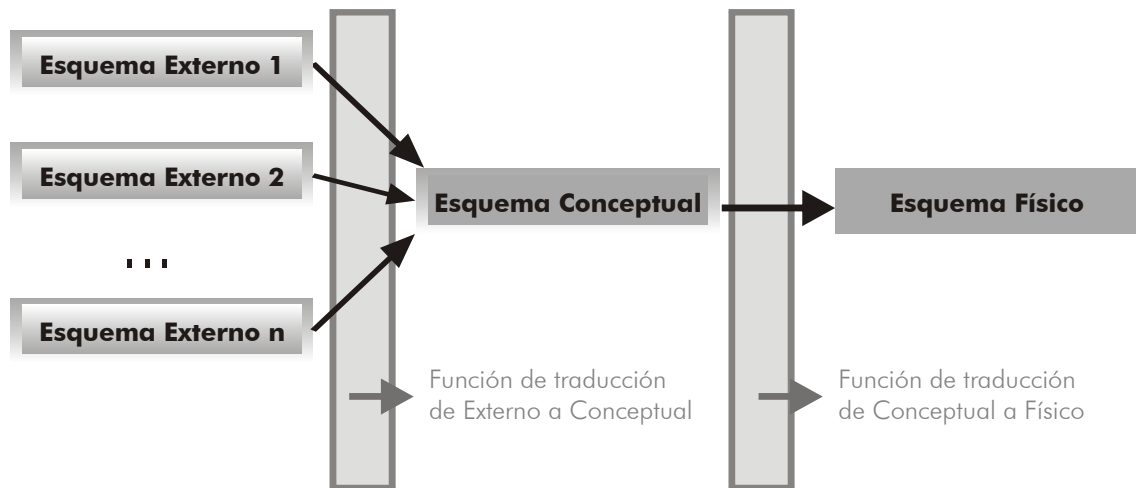
Ilustración 3, Esquema completo de la comunicación entre procesos de usuario, DBMS y Sistema Operativo

## niveles ANSI/SPARC

Una base de datos se puede ver de diferentes formas. Cada programa que accede a la base de datos manipula sólo ciertos datos y estructuras. Así cada programa posee una visión de la base de datos. La unión de todos los datos y sus relaciones forman el llamado **esquema conceptual**. Mientras que el **esquema físico** representa el almacenamiento de los datos y sus formas de acceso.

El DBMS es el encargado de realizar las traducciones para pasar del esquema conceptual al físico.

Desde la ANSI (instituto de estándares americano) se creó una sección llamada SPARC dedicada a estándares de sistemas de información. Propusieron tres niveles de abstracción en las bases de datos, de acuerdo con el siguiente esquema:



**Ilustración 4, Niveles ANSI/SPARC**

- ⊙ **Esquema externo.** Visión de la base de datos que ofrece cada aplicación. Lógicamente es distinta en cada aplicación. Representan vistas concretas de la base de datos.
- ⊙ **Esquema conceptual.** Representación teórica de los datos y de sus relaciones. Representa la lógica de la base de datos.
- ⊙ **Esquema físico.** Representa los datos según son almacenados en el medio físico (en los discos).

### independencia lógico/física

---

El esquema conceptual debe ser absolutamente independiente del físico. Esto significa:

- ⊙ **Independencia física de los datos.** Aunque el esquema físico cambie, el esquema conceptual no debe verse afectado. En la práctica esto significa que aunque se añadan o cambien discos u otro hardware, o se modifique el sistema operativo u otros cambios relacionados con la física de la base de datos, el esquema conceptual permanece invariable.
- ⊙ **Independencia lógica de los datos.** Significa que aunque se modifique el esquema conceptual, la vista que poseen las aplicaciones (los esquemas externos) no serán afectados.



# modelado de datos

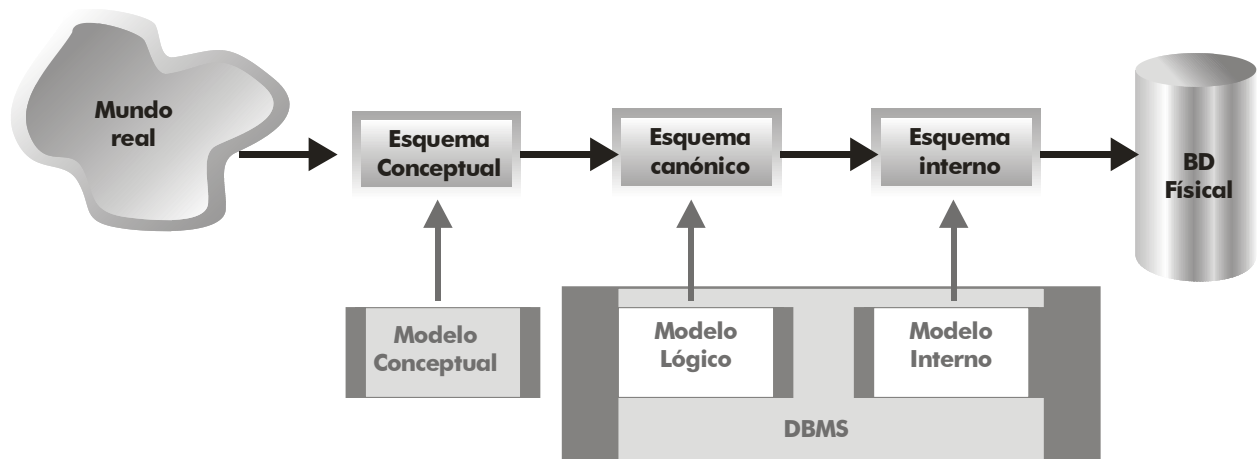
## modelos de datos

Los **modelos** se utilizan en todo tipo de ciencias. Su finalidad es la de simbolizar una parte del mundo real de forma que sea más fácilmente manipulable. En definitiva es un esquema mental (conceptual) en el que se intentan reproducir las características de una realidad específica.

En el caso de los **modelos de datos**, lo que intentan reproducir es una información real que deseamos almacenar en un sistema informático.

Se denomina **esquema** a una descripción específica en términos de un modelo de datos. El conjunto de datos representados por el esquema forma la base de datos.

## clasificación de los modelos de datos



**Ilustración 5, Clasificación de los modelos de datos**

En la ilustración anterior aparecen los distintos esquemas que llevan desde el mundo real a la base de datos física. Como se ve aparecen varios esquemas intermedios. Los que están más a la izquierda se alejan más de las características físicas. Los elementos de ese esquema son:

- ⊙ **Mundo real.** Contiene la información tal cual la percibimos como seres humanos. Es el punto de partida
- ⊙ **Esquema conceptual.** Representa el modelo de datos de forma independiente del DBMS que se utilizará.
- ⊙ **Esquema canónico (o de base de datos).** Representa los datos en un formato más cercano al del ordenador
- ⊙ **Esquema interno.** Representa los datos según el modelo concreto de un sistema gestor de bases de datos (por ejemplo Oracle)
- ⊙ **Base de datos física.** Los datos tal cual son almacenados en disco.

## **Diseño conceptual de bases de datos**

modelado de datos

---

Para conseguir estos esquemas se utilizan modelos de datos. El paso entre cada esquema se sigue con unas directrices concretas. Estas directrices permiten adaptar un esquema hacia otro.

Los dos modelos fundamentales de datos son el **conceptual** y el **lógico**. Ambos son conceptuales en el sentido de que convierten parámetros del mundo real en abstracciones que permiten entender los datos sin tener en cuenta la física de los mismos.

### diferencias entre el modelo lógico y el conceptual

---

- ⊙ El modelo conceptual es independiente del DBMS que se vaya a utilizar. El lógico depende de un tipo de SGBD en particular
- ⊙ El modelo lógico es más cercano al ordenador
- ⊙ Es más cercano al usuario el modelo conceptual, el lógico forma el paso entre el informático y el sistema.

Algunos ejemplos de modelos conceptuales son:

- ⊙ **Modelo E/R**
- ⊙ **Modelo RM/T**
- ⊙ **Modelos semántico**

Ejemplos de modelos lógicos son:

- ⊙ **Modelo relacional**
- ⊙ **Codasyl**
- ⊙ **Jerárquico**



# modelo entidad relación

## introducción

Fue ideado por Peter Chen en los años 1976 y 1977 a través de dos artículos. Se trata de un modelo que sirve para crear esquemas conceptuales de bases de datos. De hecho es prácticamente un estándar para crear esta tarea.

Se le llama modelo E/R e incluso EI (Entidad / Interrelación). Sus siglas más populares son las E/R por que sirven para el inglés y el español.

Inicialmente (en la propuesta de Chen) sólo se incluían los conceptos de entidad, relación y atributos. Después se añadieron otras propuestas (atributos compuestos, generalizaciones,...) que forman el llamado **modelo entidad relación extendido** (se conoce con las siglas **ERE**)

## entidades

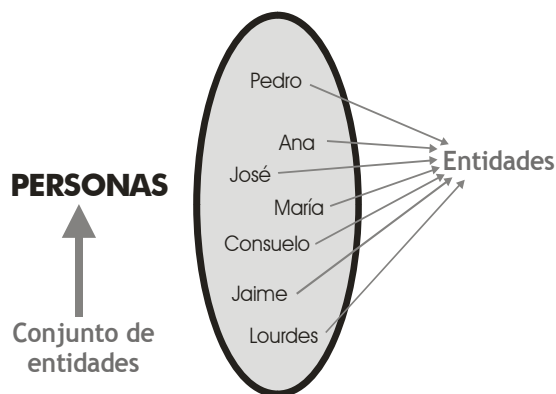
### entidad

Se trata de cualquier objeto u elemento (real o abstracto) acerca del cual se pueda almacenar información en la base de datos. Ejemplos de entidades son Pedro, la factura número 32456, el coche matrícula 3452BCW.

Una entidad no es un propiedad concreta sino un objeto que puede poseer múltiples propiedades (atributos).

### conjunto de entidades

Las entidades que poseen las mismas propiedades forman conjuntos de entidades. Ejemplos de conjuntos de entidades son los conjuntos: personas, facturas, coches,...



**Ilustración 6, Ejemplos de entidad y conjunto de entidad**

En la actualidad se suele llamar **entidad** a lo que anteriormente se ha definido como conjunto de entidades. De este modo hablaríamos de la entidad PERSONAS. Mientras que cada persona en concreto sería una **ocurrencia** o un **ejemplar** de la entidad **persona**.

## Diseño conceptual de bases de datos

modelo entidad relación

---

### representación gráfica de las entidades

---

En el modelo entidad relación los conjuntos de entidades se representan con un rectángulo dentro del cual se escribe el nombre de la entidad:



**Ilustración 7, Representación de la entidad persona**

### tipos de entidades

---

- ⦿ **Regulares.** Son las entidades normales que tienen existencia por sí mismas sin depender de otras. Su representación gráfica es la indicada arriba
- ⦿ **Débiles.** Su existencia depende de otras. Por ejemplo la entidad **tarea laboral** sólo podrá tener existencia si existe la entidad **trabajo**. Las entidades débiles se presentan de esta forma:



**Ilustración 8, Entidad débil**

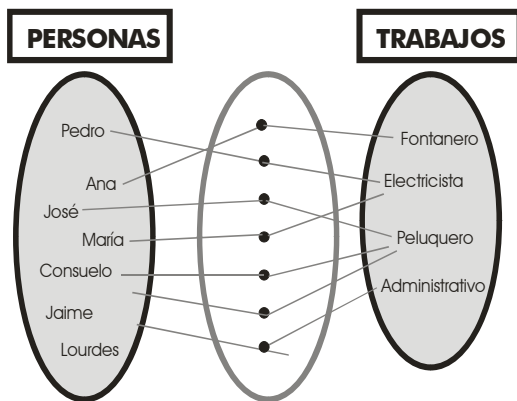
## relaciones

---

### qué es una relación

---

Representan asociaciones entre entidades. Es el elemento del modelo que permite relacionar en sí los datos del modelo. Por ejemplo, en el caso de que tengamos una entidad personas y otra entidad trabajos. Ambas se realizan ya que las personas trabajan y los trabajos son realizados por personas:



**Ilustración 9, ejemplo de relación**

representación gráfica

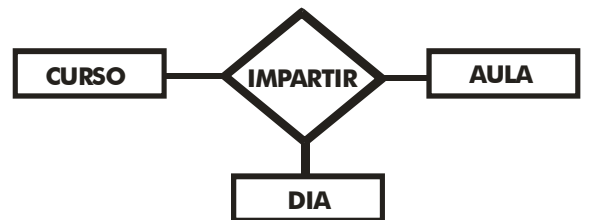
La representación gráfica de las entidades se realiza con un rombo al que se le unen líneas que se dirigen a las entidades, las relaciones tienen nombre (se suele usar un verbo). En el ejemplo anterior podría usarse como nombre de relación, trabajar:



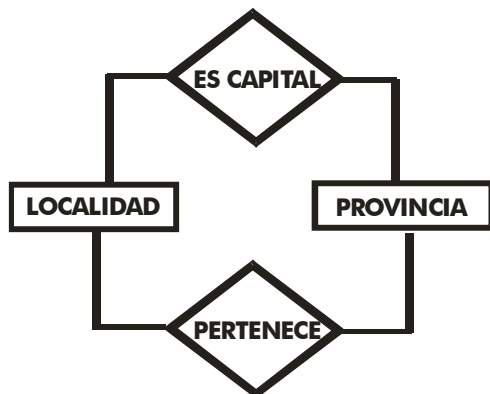
ejemplos de relaciones



Relación binaria



Relación ternaria



Relación doble



Relación reflexiva

## Diseño conceptual de bases de datos

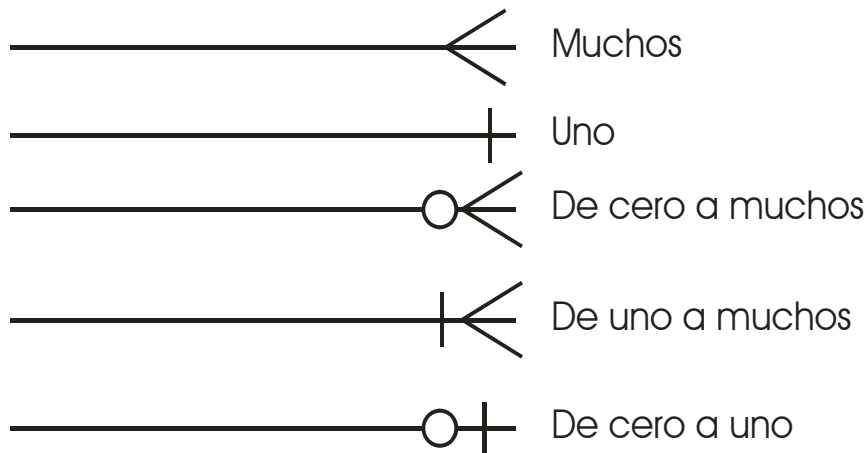
modelo entidad relación

### cardinalidad

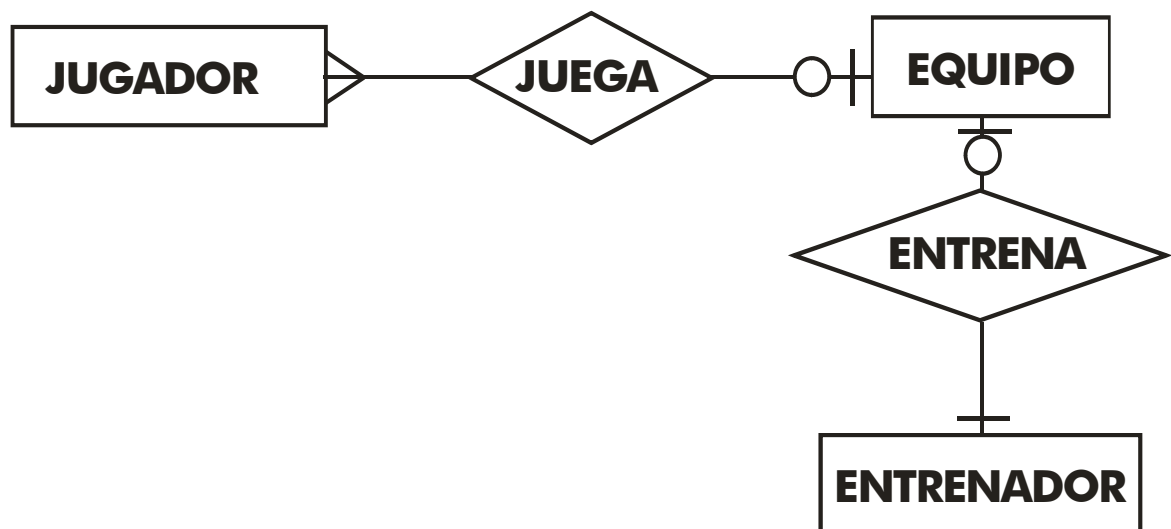
Indica el número de relaciones en las que una entidad puede aparecer. Se anota en términos de:

- **cardinalidad mínima.** Indica el número mínimo de asociaciones en las que aparecerá cada ejemplar de la entidad (el valor que se anota es de cero o uno)
- **cardinalidad máxima.** Indica el número máximo de relaciones en las que puede aparecer cada ejemplar de la entidad (puede ser uno o muchos)

En los esquemas entidad / relación la cardinalidad se puede indicar de muchas formas. Actualmente una de las más populares es esta:



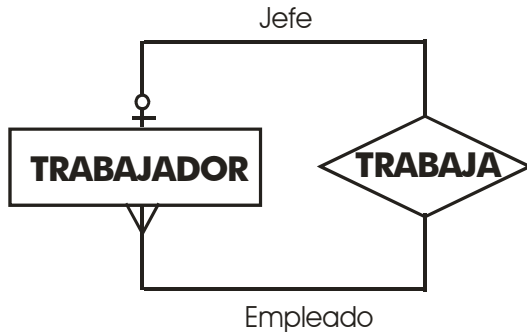
Ejemplo:



En el ejemplo, cada equipo cuenta con varios jugadores. un jugador juega como mucho en un equipo y podría no jugar en ninguno. Cada entrenador entrena a un equipo (podría no entrenar a ninguno), el cual tiene un solo entrenador

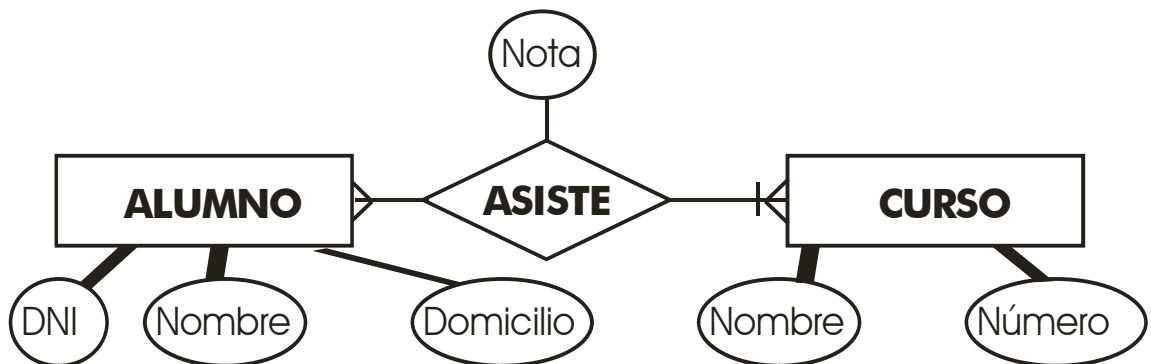
## roles

A veces en las líneas de la relación se indican **roles**. Los roles representan el papel que juega una entidad en una determinada relación. Ejemplo:



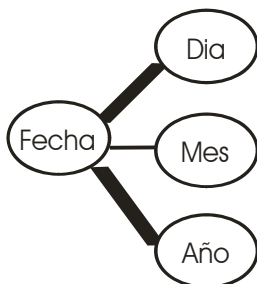
## atributos

Describen propiedades de las entidades y las relaciones. En este modelo se representan con un círculo, dentro del cual se coloca el nombre del atributo. Ejemplo:



## tipos de atributos

### compuesto



### múltiples

Pueden tomar varios valores:



## Diseño conceptual de bases de datos

modelo entidad relación

---

opcionales

Lo son si pueden tener valor nulo:



identificador

Se trata de uno o más campos cuyos valores son únicos en cada ejemplar de una entidad. Se indican subrayando el nombre del identificador.

Para que un atributo sea considerado un buen identificador tiene que cumplir:

- 1 > Deben distinguir a cada ejemplar teniendo en cuenta las entidades que utiliza el modelo. No tiene que ser un identificador absoluto.
- 2 > Todos los ejemplares de una entidad deben tener el mismo identificador.
- 3 > Cuando un atributo es importante aun cuando no tenga una entidad concreta asociada, entonces se trata de una entidad y no de un atributo

## entidades *is a*

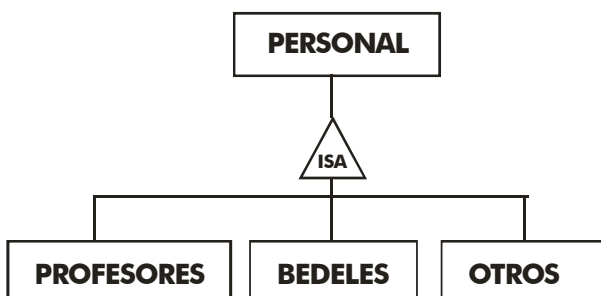
---

Son relaciones de tipo **is a** (es un) aquellas en las que una entidad se descompone en entidades especializadas. Hay dos tipos de entidades *is a*: **especializaciones** y **generalizaciones**.

Las especializaciones consisten en que una entidad se divide en entidades más concretas. La entidad general comparte con las especializadas sus atributos. Se observa una especialización cuando hay ejemplares para los que no tienen sentido algunos de los atributos, mientras que para otros sí.

Se denomina generalización si se agrupan varias entidades en una o más entidades generales. Se observa una generalización si en varias entidades se observan atributos iguales, lo que significa que hay una entidad superior que posee esos atributos.

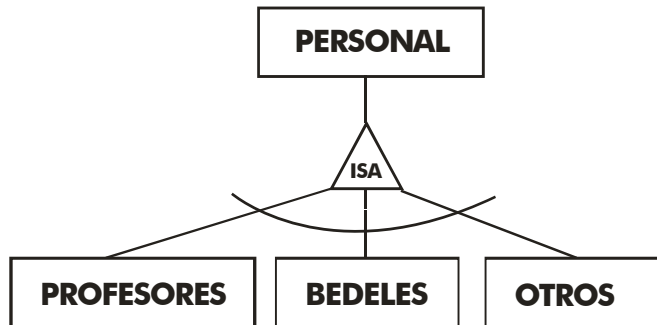
En cualquier caso la representación en el modelo es la misma, se representan con un triángulo que tiene el texto **ISA**. Ejemplo:



En estas relaciones se habla también de herencia, ya que tanto los profesores como los bedeles como los otros, heredan atributos de la entidad personal (se habla de la superentidad personal y de la subentidad profesores)

Se puede colocar un círculo (como el del número cero) en lado de la superentidad para indicar que es opcional la especialización, de otro modo se tomará como obligatoria (el personal tiene que ser alguna de esas tres cosas)

Se puede indicar también exclusividad. Esto ocurre cuando entre varias líneas hacia una relación, las entidades sólo pueden tomar una. Se representa con un ángulo en el diagrama:



En el diagrama el ángulo indica que el personal sólo puede ser o profesor o bedel u otros. No puede ser dos cosas a la vez

## **pasos para el diseño**

---

- 1 > Encontrar entidades (conjuntos de entidades)
- 2 > Identificar atributos de las entidades
- 3 > Buscar identificadores
- 4 > Especificar las relaciones y cardinalidades
- 5 > Identificar entidades débiles
- 6 > Especializar y generalizar entidades donde sea posible





## apéndice: términos técnicos

<b>ANSI</b>	<i>American National Standards Institute</i> , Instituto de estándares de Estados Unidos. Uno de los organismos de estandarización más importantes.
<b>ATU</b>	Área de trabajo de usuario. Parte de la memoria que utilizan los procesos de usuario para almacenar los datos recibidos de una base de datos.
<b>BD</b>	Abreviatura de <i>Base de Datos</i> .
<b>Buffer</b>	Zona de la memoria que se utiliza para almacenar temporalmente algunos datos.
<b>DB</b>	Abreviatura de <i>Data Base</i> , base de datos
<b>DBA</b>	<i>Data Base Administrator</i> , nombre que recibe el administrador de la base de datos
<b>DBMS</b>	<i>Data Base Management System</i> , Sistema gestor de bases de datos. El software encargado de administrar y producir bases de datos.
<b>DCL</b>	<i>Data Control Language</i> , lenguaje de control de datos. Lenguaje que proporcionan las DBMS para controlar los usuarios de la base de datos.
<b>DDL</b>	<i>Data Definition Language</i> , lenguaje de definición de datos. Lenguaje que proporcionan las DBMS para definir la base de datos.
<b>DML</b>	<i>Data Modification Language</i> , lenguaje de modificación de datos. Lenguaje que proporcionan las DBMS para realizar operaciones de búsqueda y modificación de datos.
<b>ERE</b>	Modelo entidad relación extendido
<b>OS</b>	<i>Véase SO</i>
<b>SGBD</b>	<i>Véase DBMS</i>
<b>SO</b>	Sistema operativo
<b>SPARC</b>	<i>System Planing and Repairments Comitte</i> , comité de planificación de sistemas y reparaciones, subsección de ANSI.
<b>X3</b>	Sección de ANSI encargada de los estándares de ordenadores y m