

A thick dark green vertical bar runs down the left side of the page. A green arrow-shaped banner points to the right from this bar, containing the text 'Base De Datos 1ºDAW/DAM'. In the bottom left corner, there are several thin, curved, light grey lines that sweep upwards and to the right.

Base De Datos 1ºDAW/DAM

# Resumen tema 2

Diseño De Bases De Datos

ANDREU, ORENGA RAMON

INDICE .....	0
1.DISEÑO DE UNA BASES DE DATOS .....	2
1    INTRODUCCION .....	2
2    ARQUITECTURA DE LAS BASES DE DATOS .....	2
2.1    Arquitectura De Tres Niveles.....	2
3    MODELO DE DATOS .....	3
3.1    Modelo conceptual .....	3
4    FASES DEL DISEÑO DE UNA BASE DE DATOS .....	3
4.1    FASE 1: Recogida y análisis de requisitos .....	3
4.2    FASE 2: Diseño conceptual .....	4
4.3    FASE 3: Diseño lógico.....	4
4.4    FASE 4: Diseño físico.....	4
4.5    FASE 5: Implementación y optimización .....	4
2.BASES DE DATOS RELACIONALES .....	4
1    ELEMENTOS DEL MODELO ENTIDAD/RELACION .....	4
1.1    Entidades .....	4
1.2    Atributos.....	5
1.3    Cardinalidad .....	5
1.4    Dominios.....	6
1.5    Relaciones.....	6
1.6    Claves.....	7
1.7    Valor nulo .....	7
2    REGLAS DE INTEGRIDAD .....	8
2.1    Reglas De Integridad De Entidades .....	8
2.2    Reglas De Integridad Referencial.....	8
3    EJEMPLO MODELO E/R .....	8

# 1.DISEÑO DE UNA BASES DE DATOS

## 1 INTRODUCCION

- Los usuarios no tienen por qué saber cómo se organizan y almacenan los datos.
- Una BD debe presentar los datos de forma que el usuario pueda interpretarlos y modificarlos.
- El diseño de la BD es el nivel más cercano al usuario.

## 2 ARQUITECTURA DE LAS BASES DE DATOS

Hay tres características que siempre siguen los sistemas de bases de datos:

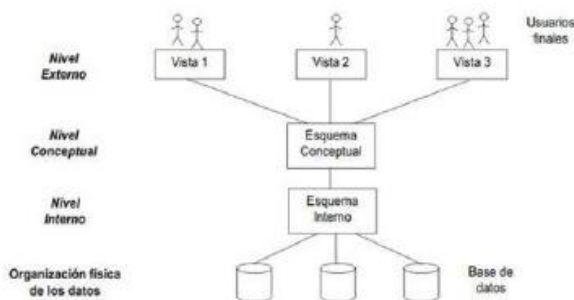
- La separación entre los programas de aplicación y los datos.
- El manejo en múltiples vistas.
- Se utiliza un catálogo para almacenar el esquema de la BD.

### 2.1 ARQUITECTURA DE TRES NIVELES

- El objetivo de la arquitectura de tres niveles es separar los programas de la aplicación de la base de datos física.
- Proporciona independencia lógica (se pueden realizar cambios en el nivel conceptual), y física (es posible modificar la ubicación de los ficheros).

Se define en tres niveles:

- Nivel externo
  - Es el nivel más cercano al usuario, con los datos que más le interesan.
  - Es lo que ven los usuarios finales.
  - Se compone de diferentes aplicaciones basadas en vistas de la BD.
  - Cada vista describe al usuario la parte que le interesa y oculta el resto.
- Nivel conceptual
  - Se representan los datos que se van a utilizar, sin tener en cuenta aspectos que se representan en el nivel interno.
  - Se compone de tablas y atributos.
  - Se utiliza para describir la estructura de la BD, se centra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones.
  - Es el nivel donde se manejan los programadores.
- Nivel interno
  - Es el nivel más cercano al almacenamiento físico de los datos.
  - Permite escribir los datos tal y como se almacenan.
  - Se crean los archivos de configuración.
  - Define que discos y archivos forman la BD.
  - Solo acceden a este nivel los administradores.



### 3 MODELO DE DATOS

- Conjunto de conceptos, reglas y convenciones que sirven para describir la estructura de una BD.
- Contiene también un conjunto de operaciones básicas para realizar consultas y actualizaciones de datos.
- Algunos permiten especificar un conjunto de operaciones definidas por el usuario.

Los modelos de datos se clasifican en tres tipos

- Modelos conceptuales: Entidades, atributos y relaciones.

- Modelos lógicos: Registros y objetos.

- Modelos físicos: Formato de los registros, estructura de los ficheros y métodos de acceso.

Modelo conceptual	Modelo Lógico	Modelo físico
Comunicación directa con el usuario. Se utiliza el modelo entidad-relación.	Es más técnico. Es más difícil de entender para el usuario y tiene traducción directa con el nivel físico. Dependerá de la implementación de la BD. Se utiliza el modelo relacional.	Resultado de aplicar el modelo lógico a un SGBD concreto. Se expresa con lenguaje SQL.

#### 3.1 MODELO CONCEPTUAL

Se conoce como esquema conceptual y consta de entidades atributos y relaciones, en esta etapa de la creación de la base de datos se deben entender los siguientes puntos:

- La perspectiva de los datos que tiene cada usuario.
- La naturaleza de los datos independientemente de su representación física.
- El uso funcional de los datos.

El diseño conceptual es completamente independiente de los aspectos de implementación, se prueba y se valida durante todo el proceso con los requisitos de los usuarios y se trata de una fuente fundamental de información para el diseño lógico de la base de datos.

### 4 FASES DEL DISEÑO DE UNA BASE DE DATOS

El esquema conceptual debe representar todos los aspectos relevantes del mundo real que va a interpretar.

Para un diseño de calidad, primero se deben negociar con el usuario/cliente el modelo conceptual.

Después, se transforma el modelo conceptual en modelo lógico.

Finalmente, se transforma el modelo lógico en físico, obteniendo la BD final.

#### 4.1 FASE 1: Recogida y análisis de requisitos

Consiste en conocer y analizar las expectativas, las necesidades y los objetivos de los futuros usuarios.

- Recogida de requisitos.
- Estructuración y refinamiento de los requisitos.
- Formalización de los requisitos.

## 4.2 FASE 2: Diseño conceptual

Tiene como objetivo crear un esquema conceptual de alto nivel e independiente de la tecnología a partir de los requisitos recogidos en la fase anterior.

- El modelo ER.
- El lenguaje unificado de modelización.

## 4.3 FASE 3: Diseño lógico

Se transforma un modelo conceptual en un modelo lógico dependiente del tipo de SGBD en el que se quiere implementar.

- Reconsideraciones del modelo conceptual.
- Transformación del modelo conceptual en modelo lógico.
- Normalización.

## 4.4 FASE 4: Diseño físico

Adapta el esquema lógico obtenido en la fase anterior al SGBD concreto que se vaya a utilizar.

- El nivel físico y el nivel virtual.
- Transforma el modelo lógico en el modelo físico.

## 4.5 FASE 5: Implementación y optimización

Consiste en hacer la carga de los datos y posteriormente ajustar algunos parámetros para optimizar el rendimiento.

- Procesamiento y optimización de consultas.
- Procesamiento de vistas.
- Administración de seguridad.

# 2.BASES DE DATOS RELACIONALES

## 1 ELEMENTOS DEL MODELO ENTIDAD/RELACION

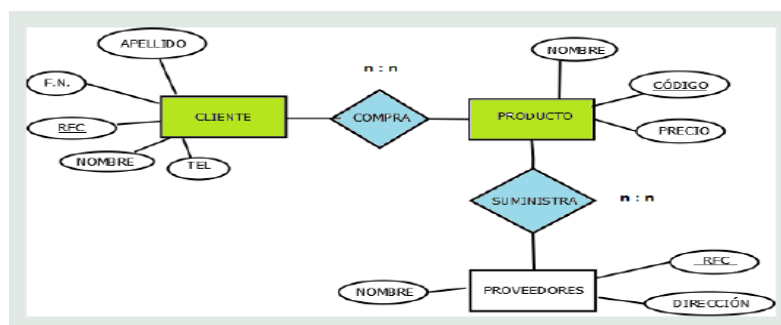
En un modelo de datos se representa la realidad a través de entidades, que son objetos que se distinguen de otros por sus características (atributos).

Se utiliza el modelo E/R en la etapa de análisis y diseño de una base de datos.

### 1.1 ENTIDADES

La entidad es un objeto mediante el cual se recoge información de interés para la base de datos y se representan gráficamente mediante rectángulos.

- Entidades fuertes: Son aquellas que no dependen de otra entidad.
- Entidades débiles: Son las que dependen de otra entidad para su existencia.

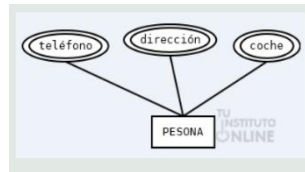
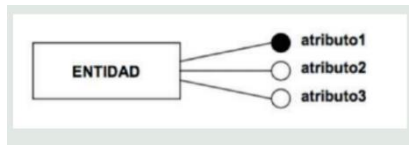


## 1.2 ATRIBUTOS

Los atributos son cada una de las características de una entidad o una relación.

Se representa gráficamente con un círculo unido por una línea a la entidad que le corresponda o dentro de un ovalo.

Es obligatorio que haya un atributo identificador que identifica de forma única la entidad en la que se encuentra.



### 1.2.1 TIPOS DE ATRIBUTOS.

Según su composición:

- Simples: Solo tiene un solo componente y no se pueden dividir en partes más pequeñas.
- Compuestos: Están formado por varios componentes con significado afín, están unidos a cada atributo del que se componen.

Según el valor:

- Monovaluados: Tienen un solo valor por cada ocurrencia
- Multivaluados: Puede tener varios valores por cada ocurrencia de la entidad.

Según su origen:

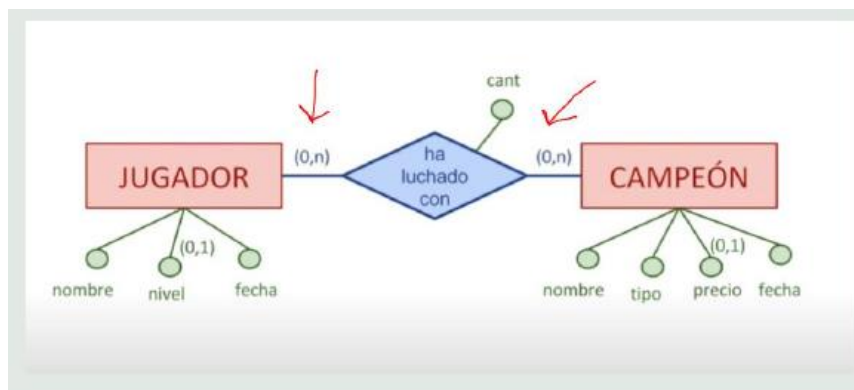
- Almacenados: Los datos de estos atributos están almacenados directamente en la base de datos sin tener ningún trámite intermedio.
- Derivados: Son aquellos que se obtiene el valor de uno o varios atributos que provienen de la misma o otra entidad.

## 1.3 CARDINALIDAD

La cardinalidad es el numero de tuplas que contiene una entidad o una relación.

Expresamos la cardinalidad definiendo el número máximo y mínimo de ocurrencias de cada tipo de entidad.

Tiene los siguientes valores: (0,1), (1,1), (0,N), (1,N) y (N,M).



## 1.4 DOMINIOS

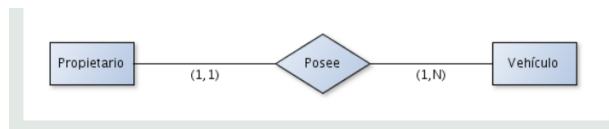
Los dominios definen la colección de valores de donde uno o mas atributos obtienen sus valores reales.

Permite que el usuario defina el significado y la fuente de los valores que los atributos deben tomar, esto hace que hay más información para el sistema a la hora de ejecutar una operación relacional y evitar operaciones semánticamente incorrectas.

Los SGBD relacionales no ofrecen soporte para los dominios ya que su implementación es extremadamente compleja.

## 1.5 RELACIONES

La relación es una asociación entre una o más entidades, tiene un nombre que la identifica del resto de las relaciones ya que cada una tiene un significado específico. Se representan gráficamente como rombos y es donde se incorpora la cardinalidad.



### 1.5.1 TIPOS DE RELACIONES

- **Uno a Uno (1:1):** A cada ocurrencia de A le corresponde como máximo una ocurrencia de B.
- **Uno a Muchos (1: N):** A cada ocurrencia de A le pueden corresponder varias de B, y a cada una de B solo le corresponde una de A, si la relación es al revés sería (N:1).
- **Muchos a Muchos (N:M):** A cada ocurrencia de A le pueden corresponder varias de B y viceversa.
- **Relaciones reflexivas:** La entidad ocupa los dos lados de la relación, la cardinalidad nos indicara si todos los elementos están relacionados o solo algunos (Ej. empleado – supervisa – empleado).

### 1.5.2 GRADOS DE RELACIONES

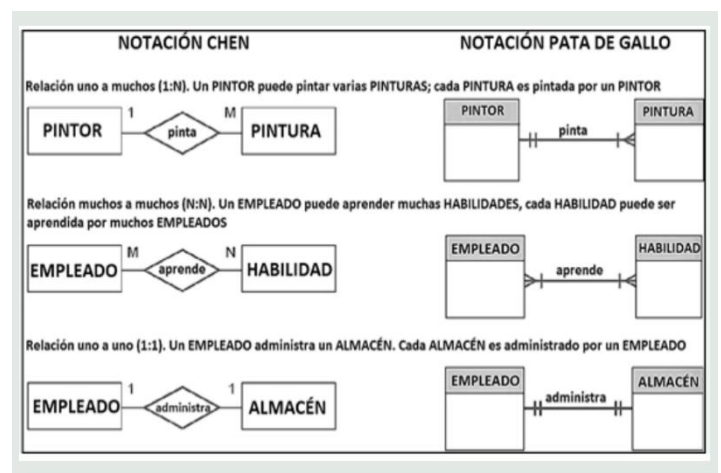
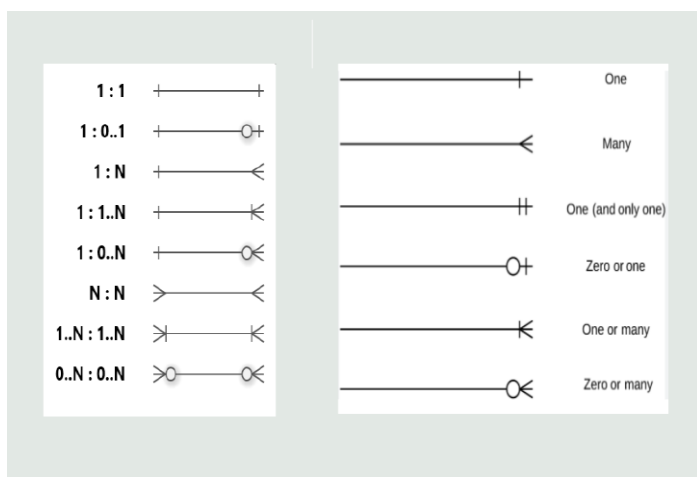
**Grado 1:** Sería la relación reflexiva, se relaciona una entidad consigo misma.

**Grado 2:** Relaciones binarias, la relación se compone de dos entidades.

**Grado 3:** Relaciones ternarias, la relación contiene 3 entidades.

**Grado n:** n entidades.

### 1.5.3 REPRESENTACION DE LAS RELACIONES



---

## 1.6 CLAVES

---

### 1.6.1 CLAVE CANDIDATA

Son columnas cuyo valor no se repite en ninguna de otra fila de esta tabla, atributos que identifican cada tupla de la relación, todas las tablas deben tener al menos una clave candidata.

---

### 1.6.2 CLAVE PRIMARIA

Se reconoce como el identificador de la tabla y que identifican cada fila de esta. Son los únicos datos en el modelo relacional que provocan redundancia ya que se duplican en las claves secundarias para establecer las relaciones, hay que intentar que contengan datos pequeños (nombres, títulos, etc.).

---

### 1.6.3 CLAVE ALTERNATIVA

Es cualquier clave candidata que no ha sido elegida primaria.

---

### 1.6.4 CLAVE AJENA EXTENSA O SECUNDARIA

Son los datos de atributos de otra tabla cuyos valores están relacionados con atributos de otra tabla.



---

## 1.7 VALOR NULO

Otorga un significado al atributo como si tuviera un valor numérico o textual (un valor nulo en una clave ajena indica que la fila donde se ha puesto ese valor no tiene ninguna clave principal asociada, en un atributo teléfono indicara que esa persona no tiene teléfono, etc.). El texto vacío o el 0 no significan valor nulo.

Así se utiliza el valor nulo en las operaciones lógicas:

- **verdadero Y (AND) nulo** da como resultado, nulo. Siguiendo la aritmética planteada antes:  $1 \cdot 0,5 = 0,5$
- **falso Y (AND) nulo** da como resultado, falso ( $0 \cdot 0,5 = 0$ )
- **verdadero O (OR) nulo** da como resultado, verdadero ( $1 + 0,5 > 1$ )
- **falso O nulo** da como resultado nulo ( $0 + 0,5 = 0,5$ )
- **la negación de nulo** da como resultado nulo



## 2 REGLAS DE INTEGRIDAD

Son restricciones que se deben cumplir en todas las bases de datos relacionales, estas reglas son:

- Reglas de integridad de entidades.
- Reglas de integridad referencial.

### 2.1 REDGALS DE INTEGRIDAD DE ENTIDADES

Es la primera regla de integridad y se aplica a las claves primarias de las relaciones: *ninguno de los atributos de la clave primaria puede ser nulo*.

Si se permitiera que fuera nula se estaría diciendo que no todos sus atributos son necesarios, con lo que se estaría contradiciendo la irreductibilidad.

### 2.2 REGLAS DE INTEGRIDAD REFERENCIAL

Esta regla se aplica a las claves ajenas: deben coincidir sus valores con valores de la clave primaria a la que hacen referencia o bien ser nulos.

Cada clave ajena debe contestar las siguientes tres preguntas:

- ¿Tiene sentido que acepte nulos? (regla de los nulos).
- ¿Qué ocurre si se intenta borrar la tupla a la que hace referencia la clave ajena? (regla de borrado).
- ¿Qué ocurre si se intenta modificar la tupla a la que hace referencia la clave ajena? (regla de modificación).

Ante el borrado y la modificación se pueden ver los siguientes comportamientos:

- Restringir: No se permite borrar/modificar la tupla referenciada.
- Propagar: Se propaga el cambio o el borrado a las tuplas que hace referencia la clave ajena.
- Anular: Se borra/modifica la tupla y las tuplas a las que hace referencia ponen en nulo la clave ajena (si acepta nulos).
- Valor por defecto: Se borra/modifica la tupla y las tuplas a las que hace referencia ponen un valor establecido por defecto en la clave ajena.

## 3 EJEMPLO MODELO E/R

### PROBANDO . . .

Representa un diagrama E/R que contenga:

Una entidad EMPLEADO formada por los atributos: n° empleado, Apellido, Salario y Comisión.

Una entidad DEPARTAMENTO con los atributos: n° departamento, nombre y localidad.

Estas dos entidades se relacionan de forma que a un departamento le pertenecen cero o más empleados y un empleado pertenece a un departamento y sólo a uno.

### EJEMPLO DEPARTAMENTO - EMPLEADOS

La relación "PERTENECE" entre las entidades EMPLEADOS y DEPARTAMENTO tiene la correspondencia 1:N.

Es decir, a un departamento le pertenecen cero o más empleados (0,N) y un empleado pertenece a un departamento y sólo a uno (1:1).

