

# Modelamiento de Datos

## Modelo Entidad-Relación

El Modelo Entidad-Relación (MER) es una herramienta fundamental en el diseño conceptual de bases de datos. Proporciona una representación clara y estructurada de cómo se organizan los datos en un sistema, permitiendo a los diseñadores abstraerse de detalles técnicos para enfocarse en la lógica del negocio.

En esta clase, profundizaremos en los conceptos clave del MER, sus componentes principales (entidades, atributos y relaciones), y cómo utilizar Diagramas Entidad-Relación (DER) para modelar sistemas complejos. Además, resolveremos ejemplos prácticos y propondremos ejercicios para afianzar los conocimientos adquiridos.

## Objetivos

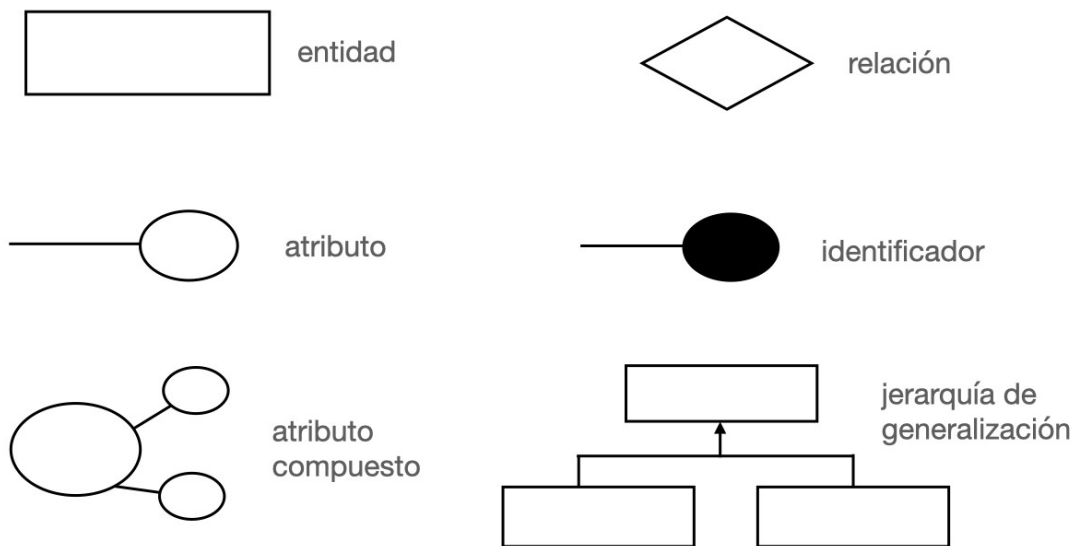
1. Comprender los fundamentos del Modelo Entidad-Relación.
2. Identificar y aplicar correctamente los elementos del MER: entidades, atributos, relaciones y cardinalidades.
3. Diseñar DER utilizando casos prácticos.
4. Resolver problemas mediante ejercicios de diseño conceptual.

# Fundamentos del Modelo Entidad-Relación

## ¿Qué es el Modelo Entidad-Relación?

El MER es un modelo conceptual que describe las entidades (objetos o cosas sobre las que se desea almacenar información), sus propiedades (atributos) y las conexiones entre ellas (relaciones). Este modelo se utiliza como base para el diseño lógico de una base de datos antes de implementarla en un Sistema de Gestión de Bases de Datos (SGBD).

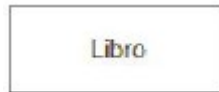
## Componentes del MER



**Entidad:** Representa un conjunto de objetos o cosas que comparten características comunes. Por ejemplo:

- "Cliente"
- "Producto"
- "Empleado"

- **Entidades regulares:** son entidades por las cuales las ocurrencias tienen existencia propia. Gráficamente, una entidad se representa como un rectángulo.



- **Entidades débiles:** son entidades que dependen de la existencia de una concurrencia de la entidad regular, de forma que si desaparece una concurrencia de la entidad regular, desaparecerán automáticamente todas las ocurrencias de la entidad débil dependientes. Una entidad débil se representa con un rectángulo concéntricos con el nombre de la entidad en el interior.



2. **Atributo:** Es una propiedad o característica de una entidad. Cada entidad tiene uno o más atributos que describen su estado. Por ejemplo:

- Para la entidad "Cliente": `ID\_Cliente`, `Nombre`, `Dirección`, `Teléfono`.
- Para la entidad "Producto": `ID\_Producto`, `Nombre`, `Precio`.

Los atributos pueden ser:

- **Simple:** No se descomponen en subpartes (ejemplo: `Nombre`).
- **Compuesto:** Se dividen en subpartes (ejemplo: `Dirección` → `Calle`, `Ciudad`, `País`).
- **Multivaluado:** Pueden tener múltiples valores (ejemplo: `Teléfonos` de un cliente).

Gráficamente, los atributos se representan como óvalos conectados a la entidad.

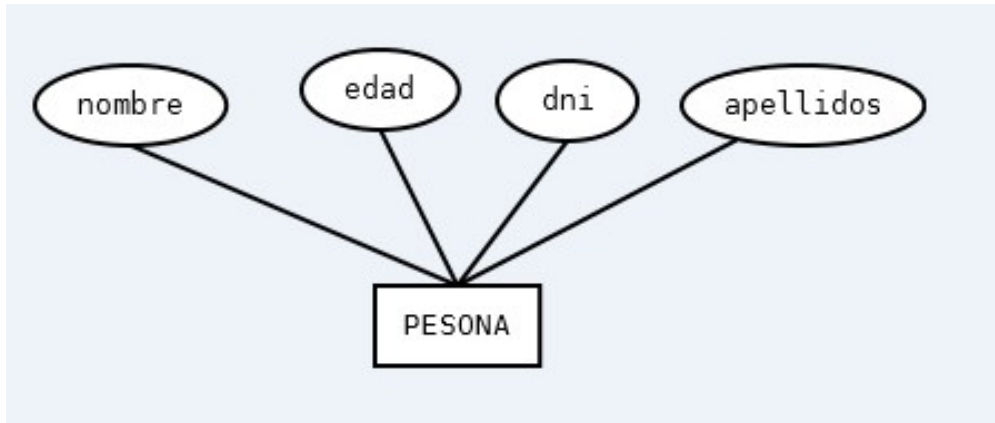
## Atendiendo a los valores

Otra forma de clasificarlos es atendiendo al número de valores que puede tener cada atributo:

- **Monovaluados**

Un atributo monovaluado es aquel que tiene un solo valor por cada ocurrencia de la entidad a la que pertenece. Se representan mediante un círculo.

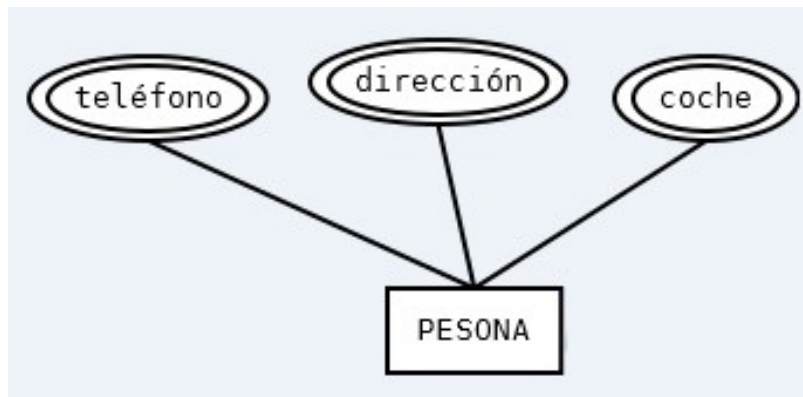
- Ejemplos: nombre, edad, dni, apellidos



### Multivaluados

Por otra parte, un atributo multivaluado puede tener varios valores por cada ocurrencia de la entidad. Se representan de manera similar, pero en lugar de un círculo son dos, uno dentro de otro.

- Ejemplos: teléfono, dirección, coche



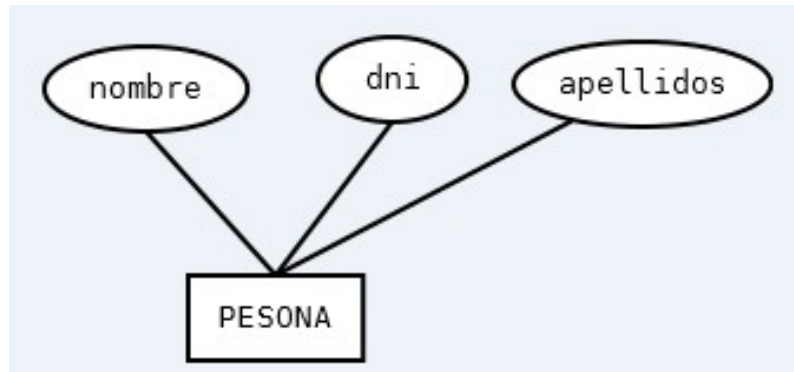
## Atendiendo al origen

Se basa en el origen de los datos.

### Almacenados

Son aquellos cuyos datos se almacenan directamente en la base de datos sin necesidad de realizar ningún trámite intermedio. Se representan mediante círculos.

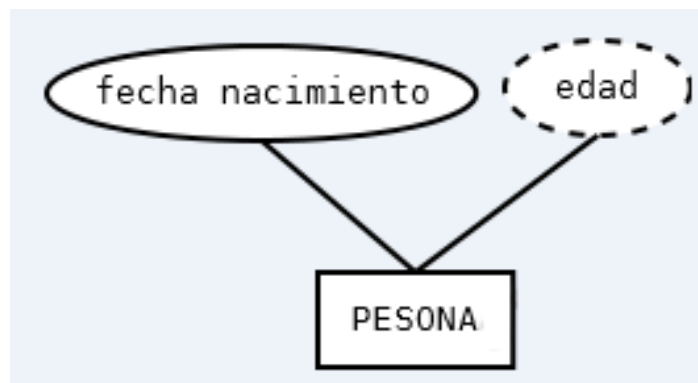
- Ejemplos: nombre, dni, apellidos



### Derivados

Por contra, los atributos derivados son aquellos que son obtenidos a partir del valor de uno o varios atributos existentes en la misma o en otras entidades. Se representan mediante círculos discontinuos.

- Ejemplos: edad (a partir de la fecha de nacimiento)



3. **Relación:** Define cómo interactúan dos o más entidades entre sí. Por ejemplo:

- Un "Cliente" realiza una "Compra".
- Un "Profesor" imparte un "Curso".

Gráficamente, una relación se representa como un rombo conectado a las entidades involucradas.

**Características de una relación:**

- Nombre: Toda relación debe tener un nombre único en el esquema.
- Grado: Numero de entidades que participan en una relación. Se clasifican en:

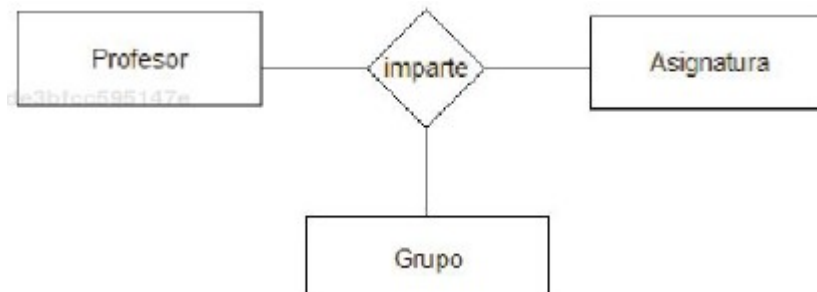
a) Relaciones binarias o de grado 2: Relacionan a dos entidades.



b) Relaciones reflexivas o de grado 1: Relacionan una entidad consigo misma.



c) Relaciones ternarias, cuaternarias,... o de grado 3,4,...: Relaciones tres, cuatro,... entidades respectivamente.



Las relaciones tienen **cardinalidades**, que indican cuántas instancias de una entidad están relacionadas con otra. Las cardinalidades más comunes son:

- Uno a uno (1:1): Una instancia de una entidad se relaciona con una sola instancia de otra entidad. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso).

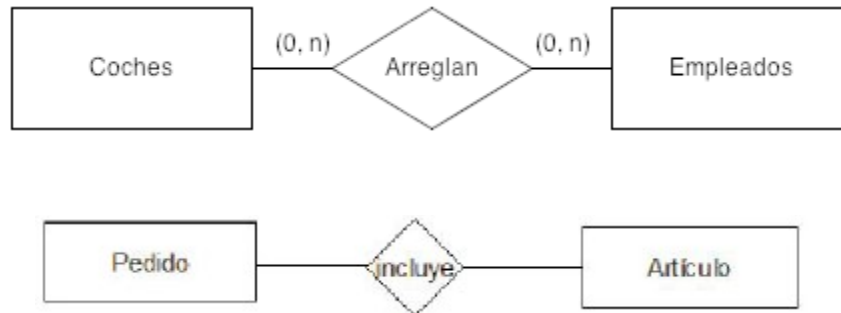


*Cada Persona (entidad) está casado (relación) con una única Persona (entidad) y viceversa. Relación 1:1.*

- Uno a muchos (1:N): Una instancia de una entidad se relaciona con varias instancias de otra entidad.



- Muchos a muchos (M:N): Varias instancias de una entidad se relacionan con varias instancias de otra entidad.



## Dominios

Cada atributo puede tener un conjunto de valores posibles. La descripción de los posibles valores de un atributo es lo que denominamos dominio

### Ejemplos

El atributo `NombredelDepartamento` puede definirse como "el conjunto de cadenas con más de siete caracteres que representan los departamentos de Universidad".

Otros ejemplos de dominios son:

- Nombre: cadena de 10 caracteres
- Edad: número
- Fecha: date
- Peso: número con dos decimales
- Ciudad: cadena de 20 caracteres

### Ejemplo Práctico: Sistema de Gestión de Biblioteca

#### Descripción del Problema

Diseñar un MER para gestionar una biblioteca donde se almacene información sobre libros, usuarios y préstamos.

#### Entidades y Atributos

##### 1. Libro:

- Atributos: ``ID_Libro`` (clave primaria), ``Título``, ``Autor``, ``Género``, ``Año_Publicación``.

##### 2. Usuario:

- Atributos: ``ID_Usuario`` (clave primaria), ``Nombre``, ``Dirección``, ``Teléfono``.



### 3. Préstamo:

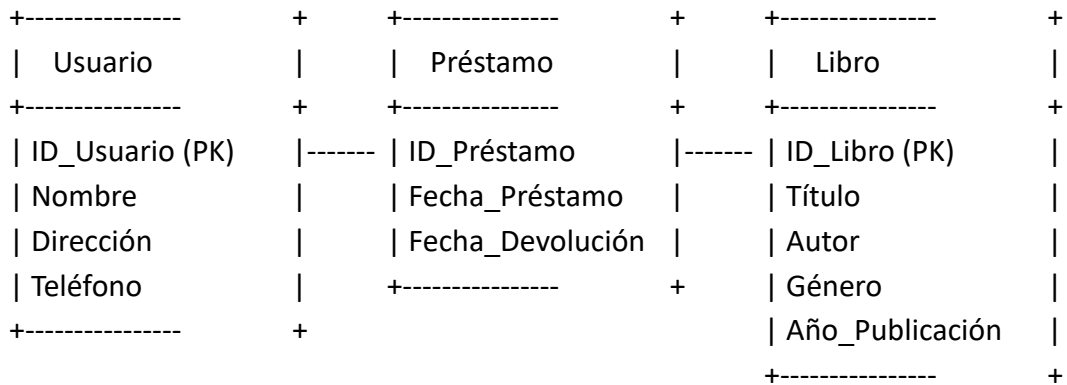
- Atributos: `Fecha\_Préstamo`, `Fecha\_Devolución`.

#### Relaciones

- Un *Usuario* puede tomar prestados varios *Libros* (1:N).

- Un *Libro* puede ser prestado a varios *Usuarios* en diferentes momentos (M:N).

#### Diagrama Final - Clases



#### Explicación:

La relación M:N entre "Usuario" y "Libro" se resuelve creando una nueva entidad llamada "Préstamo", que actúa como una tabla intermedia.

## Modelamiento Lógico

Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos.

### Características

- Incluye todas las entidades y relaciones entre ellos.
- Todos los atributos para cada entidad están especificados.
- La clave principal para cada entidad está especificada.
- Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).
- La normalización ocurre en este nivel.

### Etapas

- Especifique claves primarias para todas las entidades.
- Encuentra las relaciones entre diferentes entidades.
- Encuentra todos los atributos para cada entidad.
- Resuelva las relaciones de muchos a muchos.
- Normalización.

## Cardinalidades y Restricciones

Las cardinalidades definen las reglas de participación de las entidades en una relación. Algunas restricciones importantes incluyen:

1. Participación Total: Todas las instancias de una entidad deben participar en la relación.  
- Ejemplo: Todo "Empleado" debe estar asignado a un "Departamento".
2. Participación Parcial: Solo algunas instancias de una entidad participan en la relación.  
- Ejemplo: No todos los "Clientes" han realizado una "Compra".

Gráficamente, la participación total se representa con una línea doble, mientras que la participación parcial se representa con una línea simple.

### Ejercicio Práctico

Problema 1: Sistema de Gestión de Tiendas

Diseñe un MER para un sistema de gestión de tiendas que cumpla con los siguientes requisitos:

1. La tienda gestiona *Productos* y *Clientes*.

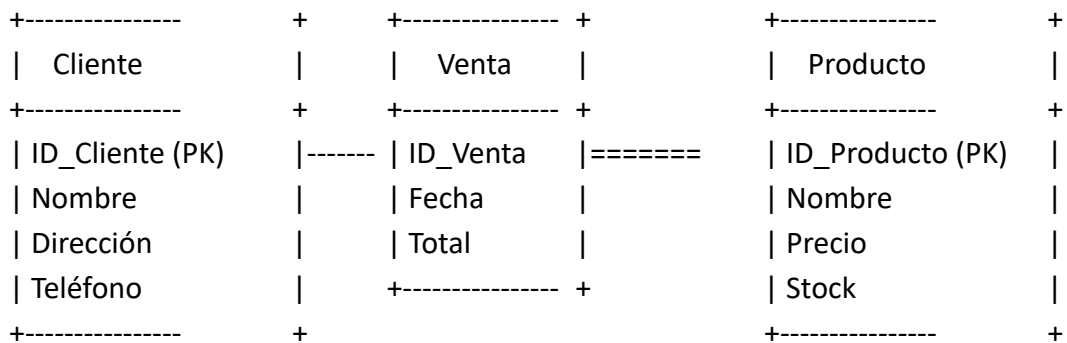
- Un *Producto* tiene un `ID\_Producto`, `Nombre`, `Precio` y `Stock`.
- Un *Cliente* tiene un `ID\_Cliente`, `Nombre`, `Dirección` y `Teléfono`.

2. Los *Cientes* generan *Ventas*.

- Una venta puede incluir varios productos.
- Un producto puede estar en varias ventas.

### Solución

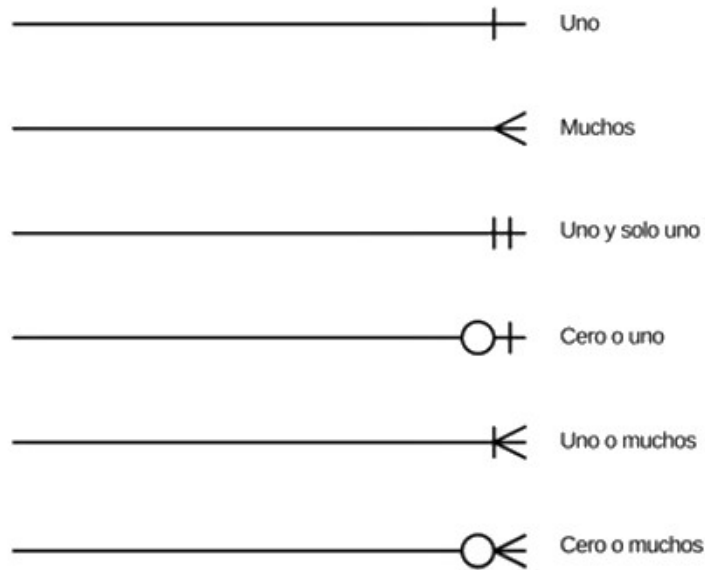
#### Diagrama Final - Clases



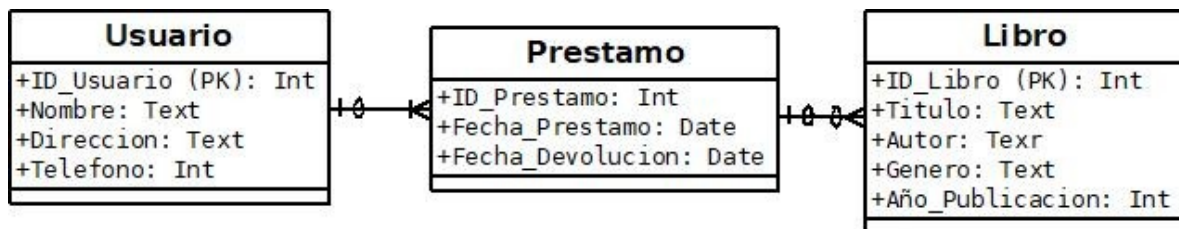
#### Explicación:

- La relación M:N entre "Venta" y "Producto" se resuelve con una tabla intermedia llamada "Detalle\_Venta", que podría incluir atributos como `Cantidad` y `Subtotal`.

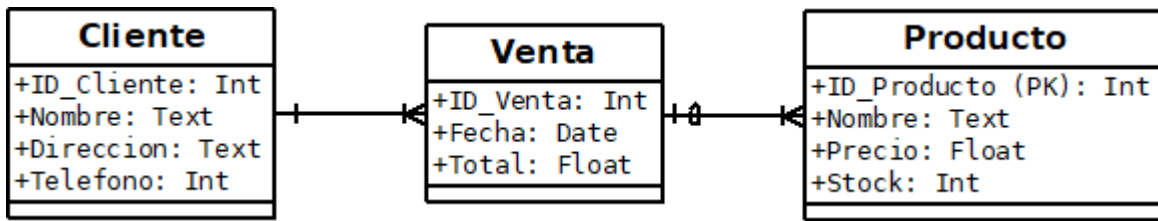
# Tipos Cardinalidad



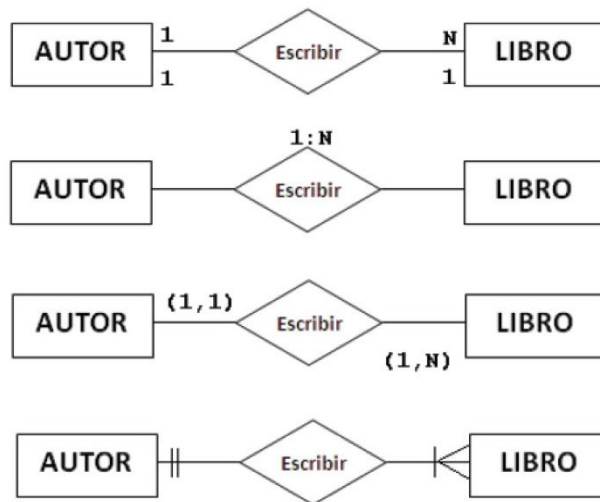
## Ejemplo 1



## Ejemplo 2

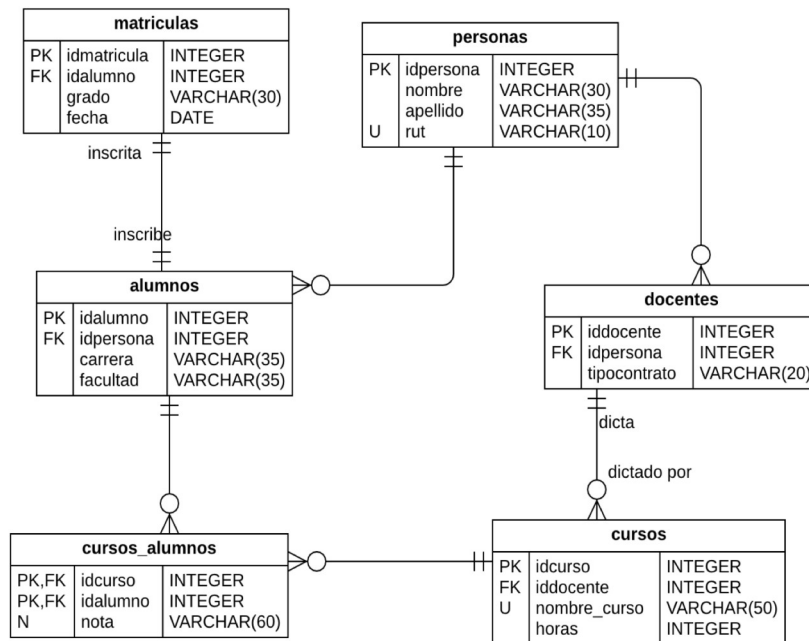


## Formas de representar Cardinalidades



*El Modelo Entidad-Relación es una herramienta esencial para el diseño conceptual de bases de datos. Permite representar de manera clara y estructurada cómo se organizan y relacionan los datos en un sistema, facilitando la comunicación entre diseñadores, desarrolladores y usuarios finales.*

# Modelamiento Físico



El modelo de datos físicos representa cómo se construirá el modelo en la base de datos. Un modelo de base de datos física muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.

## Características

- Especificación de todas las tablas y columnas.
- Las claves externas se usan para identificar relaciones entre tablas.
- La desnormalización puede ocurrir según los requisitos del usuario.

## Etapas

- Convertir entidades en tablas.
- Convertir relaciones en claves externas.
- Convertir atributos en columnas.
- Modificar el modelo de datos físicos en función de las restricciones / requisitos físicos.

Leer la siguiente documentación:

<https://aws.amazon.com/es/compare/the-difference-between-logical-and-physical-data-model/>

<https://huridocs.org/community-resources/disenando-tu-modelo-de-datos-conceptual/>

Cuadro comparativo detallado de los modelos de modelamiento de datos mas comunes

| Aspecto               | Modelo Conceptual  | Modelo Entidad-Relación (ER)   | Modelo Lógico  | Modelo Físico   |
|-----------------------|--|--|--|---|
| Objetivo              | Capturar requisitos empresariales y conceptos clave.   | Representar entidades, atributos y relaciones de forma visual.   | Definir la estructura detallada sin dependencia técnica.   | Implementar la base de datos en un sistema específico (DBMS).   |
| Elementos principales | <ul style="list-style-type: none"> <li>- Entidades</li> <li>- Relaciones (sin detalles técnicos)</li> <li>- Conceptos clave</li> </ul> | <ul style="list-style-type: none"> <li>- Entidades</li> <li>- Atributos</li> <li>- Relaciones</li> <li>- Cardinalidades</li> </ul> | <ul style="list-style-type: none"> <li>- Entidades detalladas</li> <li>- Atributos con tipos de datos</li> <li>- Claves primarias y foráneas</li> <li>- Normalización</li> </ul> | <ul style="list-style-type: none"> <li>- Tablas</li> <li>- Columnas con tipos específicos</li> <li>- Índices</li> <li>- Restricciones (PK, FK)</li> <li>- Optimizaciones (particiones)</li> </ul> |
| Nivel de abstracción  | Muy alto (orientado a negocio)   | Alto (enfocado en estructura visual)   | Medio (detalles técnicos sin implementación)   | Bajo (implementación concreta)  |
| Público objetivo      | Stakeholders y equipos no técnicos   | Analistas de datos, diseñadores y stakeholders   | Arquitectos de datos y analistas técnicos  | DBAs y desarrolladores  |
| Detalle               | Sin atributos ni   | Incluye  | Tipos de datos   | Tipos de datos  |

|   |  |  |  |   |
|---|--|--|--|---|
| <b>técnico</b>                                    | detalles técnicos                                    | cardinalidades y atributos básicos                             | genéricos, normalización                                 | específicos del DBMS, índices, SQL                                      |
| Tipos de datos específicos del DBMS, índices, SQL | No depende   | No depende   | No depende   | Totalmente dependiente  |
| <b>Ejemplo</b>                                    | Mapa conceptual con "Ventas" vinculado a "Clientes". | Diagrama ER con entidades "Cliente" y "Pedido", y su relación. | Tabla "Clientes" con campos: ID (INT), Nombre (VARCHAR). | Script SQL:<br>CREATE TABLE Clientes (ID INT PRIMARY KEY, ...);         |
| <b>Fase de desarrollo</b>                         | Fase inicial de requisitos                           | Fase inicial de diseño conceptual                              | Fase intermedia de diseño                                | Fase final de implementación  |
| <b>Herramientas comunes</b>                       | Diagramas de flujo, UML (simplificado)               | Lucidchart, Draw.io, ER/Studio                                 | ERwin, PowerDesigner                                     | SQL Server Management Studio, phpMyAdmin, pgAdmin, Oracle SQL Developer |

Este cuadro refleja la progresión desde lo abstracto (negocio) hasta lo concreto (técnico), adaptándose a las necesidades de cada etapa del desarrollo.

## ¿Qué es el proceso de modelado de datos?

El proceso de modelado de datos sigue una secuencia de pasos que tiene que hacer repetidamente hasta crear un modelo de datos completo. Dentro de cualquier organización, varias partes interesadas se reúnen para crear una visión completa de los



datos. Aunque los pasos varían según el tipo de modelado de datos, lo que sigue es una visión general.

### **Paso 1: identificar las entidades y sus propiedades**

Identifique todas las entidades de su modelo de datos. Cada entidad tiene que ser lógicamente distinta de todas las demás y puede representar personas, lugares, cosas, conceptos o eventos. Cada entidad es distinta ya que tiene una o más propiedades únicas. Puede pensar en las entidades como sustantivos y en los atributos como adjetivos en su modelo de datos.

### **Paso 2: identificar las relaciones entre entidades**

Las relaciones entre las distintas entidades son el núcleo del modelado de datos. Las reglas empresariales definen inicialmente estas relaciones en un nivel conceptual. Puede pensar en las relaciones como los verbos de su modelo de datos. Por ejemplo, el vendedor vende muchos autos, o la sala de exhibiciones emplea a muchos vendedores.

### **Paso 3: identificar la técnica de modelado de datos**

Después de entender conceptualmente sus entidades y sus relaciones, puede determinar la técnica de modelado de datos que mejor se adapte a su caso de uso. Por ejemplo, puede usar el modelado de datos relacional para los datos estructurados, pero el modelado de datos dimensional para los datos no estructurados.

### **Paso 4: optimizar y repetir**

Puede optimizar aún más su modelo de datos para adaptarlo a sus necesidades tecnológicas y de rendimiento. Primero hay que determinar cómo se va a acceder a los datos y, a continuación, modelar los datos en la forma en que se va a acceder a ellos.

Por lo general, estos pasos se repiten a medida que la tecnología y los requisitos cambian con el tiempo.

# Actividades Propuestas

## **Ejercicio 1: Sistema de Gestión de Clínicas**

Diseñe un MER para una clínica que gestione pacientes, médicos y citas. Considere lo siguiente:

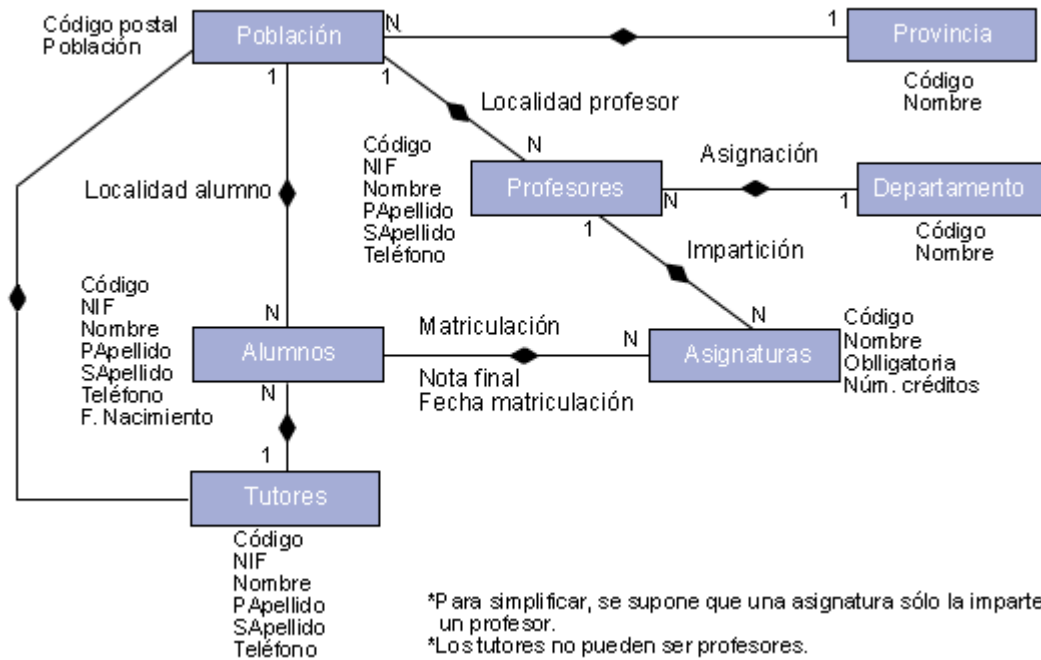
1. Un *Paciente* puede tener varias *Citas*.
2. Un *Médico* puede atender varias *Citas*.
3. Cada *Cita* tiene una fecha y hora específica.

## **Ejercicio 2: Sistema de Gestión de Universidades**

Diseñe un MER para una universidad que gestione departamentos, profesores y cursos. Considere lo siguiente:

1. Un *Departamento* puede tener varios *Profesores*.
2. Un *Profesor* puede impartir varios *Cursos*.
3. Un *Curso* puede ser impartido por varios *Profesores* en diferentes semestres.

### Ejercicio 3: Construir las entidades y relaciones de acuerdo al siguiente diagrama



1. Crear las entidades (tablas) según el esquema, en una hoja de calculo preferiblemente.
2. Definir las llaves primarias
3. Definir las llaves foraneas
4. Están todas las llaves foraneas creadas?
5. Completar las entidades, agregando las llaves foraneas
6. Crear diagrama de clases, con las tablas completas.

## Referencias

- Elmasri, R., & Navathe, S. B. (2016). \*Fundamentals of Database Systems\*. Pearson.
- Date, C. J. (2019). \*Database Design and Relational Theory\*. Apress.
- UML.org. (2023). \*Unified Modeling Language Resources\*.
- [TuInstitutoOnline.com](https://www.tuinstitutoonline.com) M.Donosó, G.García, P.Gargallo, A.Martínez. v. 2.0.2.1.0
- <https://bookdown.org/paranedagarcia/database/modelamiento-de-datos.html>

## Videos

<https://www.youtube.com/watch?v=Z0yLerU0g-Q>

<https://www.youtube.com/watch?v=TKuxYHb-Hvc>

<https://www.youtube.com/watch?v=jshi9VCTm7g>