

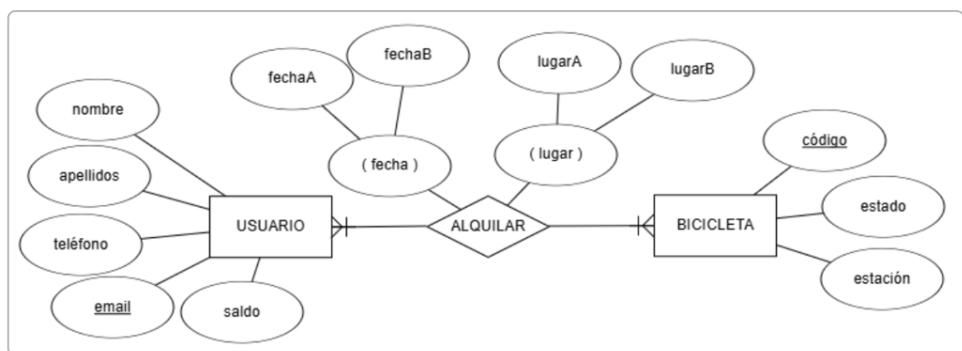
Modelo de datos: Entidad-Relación y Modelo Relacional

Modelo Entidad-Relación (Conceptual)

El **modelo Entidad-Relación (E-R)** es un modelo de datos conceptual de alto nivel que representa el “mundo real” mediante entidades, atributos y relaciones [1](#) [2](#). Permite abstraer información clave: una **entidad** es cualquier objeto, suceso o concepto con existencia independiente (p.ej., *Persona*, *Cliente*, *Alumno*) [3](#), definido por **atributos** (nombre, fecha de nacimiento, etc.) obtenidos mediante un proceso de abstracción [4](#). Cada entidad suele identificarse con uno o más **atributos identificadores** (clave primaria), que deben ser únicos y no nulos. Como indica la regla: “toda entidad debe tener uno o más atributos identificadores” [5](#). Por ejemplo, en una entidad **USUARIO** pueden considerarse identificadores simples como **login** o compuestos como (**tipoVía**, **nombreCalle**) [6](#), eligiéndose uno como clave principal y marcando el resto como claves candidatas.

Las **relaciones** unen dos o más entidades, indicando cómo se asocian entre sí. Cada relación tiene una **cardinalidad** que describe cuántas instancias de una entidad se relacionan con cuántas de otra [7](#). Por ejemplo, una relación “Autor – Libro” de tipo “uno a muchos” significa que un autor puede escribir varios libros, pero cada libro tiene un único autor [7](#). De igual modo existen relaciones uno-a-uno (1:1), uno-a-muchos (1:N) y muchos-a-muchos (N:M) [7](#) [8](#). Las relaciones pueden tener atributos propios (por ejemplo, fecha de matrícula en una relación **Matricular(Alumno,Asignatura)**) [8](#).

Las **entidades débiles** son aquellas que dependen de otra (fuerte) para su existencia. Una entidad se considera débil cuando “no puede existir sin participar en la relación” y no puede identificarse únicamente por sus propios atributos [9](#). En los diagramas E-R, las entidades débiles se representan con un rectángulo de doble borde [10](#). Por ejemplo, si existe una entidad *Garantía* que solo tiene sentido vinculada a un *Producto*, *Garantía* sería débil y heredaría parte de la clave de *Producto* para identificarse.



En el diseño conceptual se indican entidades, atributos y relaciones con sus cardinalidades. En este diagrama de ejemplo, **USUARIO** y **BICICLETA** se relacionan a través de **ALQUILAR** con cardinalidades (0,N)-(0,N), indicando una relación muchos-a-muchos [7](#) [8](#).

Transformación al Modelo Relacional y Normalización

El **modelo relacional** es el modelo lógico que representa la base de datos como un conjunto de tablas (relaciones) con tuplas y dominios de atributos. En el diseño de bases de datos, primero se crea el esquema conceptual E-R, y luego se transforma a un esquema relacional lógico ¹¹. Cada **entidad** del modelo E-R se convierte en una **tabla** en el modelo relacional, donde los atributos de la entidad son columnas de la tabla ¹². A continuación se aplican las siguientes reglas generales de transformación:

- **Entidades → Tablas:** Para cada entidad (fuerte o débil) se crea una tabla, y todos sus atributos se convierten en columnas de esa tabla ¹². La clave primaria de la entidad se marca como columna clave (PK) en la tabla.
- **Relaciones 1:1:** Por lo general **no generan nueva tabla**. Se elige qué lado aporta la clave primaria: la PK de una de las entidades se añade como clave ajena (FK) en la tabla de la otra ¹³. Por ejemplo, si *Usuario* (1,1) y *Perfil* (0,1) están en 1:1, se pone `id_usuario` como FK en *Perfil* ¹³. Solo en el caso de participación (0,1)-(0,1) se puede crear una tabla intermedia como excepción.
- **Relaciones 1:N:** Tampoco generan tabla nueva. La PK de la entidad en el lado "1" se agrega como FK en la tabla de la entidad en el lado "N" ¹⁴. Por ejemplo, en relación *País* (1) – *Presidente* (N), la tabla *PRESIDENTE* incluirá `id_pais` como FK ¹⁴.
- **Relaciones N:M:** Siempre **generan una nueva tabla intermedia**. Esa tabla incluirá como clave primaria compuesta las PKs de ambas entidades (cada una también como FK), e incluirá cualquier atributo de la relación ¹⁵. Por ejemplo, la relación muchos-a-muchos *ALUMNO* – *EXAMEN* produce la tabla *ALUMNO_HACE_EXAMEN* con `id_alumno` y `id_examen` como PKs/FKs ¹⁵.

Además, al diseñar el modelo relacional se asignan **tipos de datos** apropiados a cada columna (numérico, texto, fecha, etc.) y se definen restricciones de integridad. Cada columna debe tener un valor atómico dentro de su dominio ¹⁶ ¹⁷. Se definen las **restricciones de dominio** (p.ej. entero entre 0 y 100), así como las de **clave**: toda tabla debe declarar su **clave primaria** (PK) que garantiza unicidad ¹⁸, y las **claves foráneas** (FK) que referencian PK de otras tablas. También se establecen restricciones de **NO NULO**, **ÚNICO**, **CHECK**, etc., para asegurar la integridad de los datos ¹⁷ ¹⁸.

Normalización y Desnormalización

La **normalización** es el proceso de organizar las tablas del modelo relacional para minimizar redundancias y anomalías. Las primeras tres formas normales (1FN, 2FN, 3FN) son las más usadas:

- **1FN (Primera Forma Normal):** Cada columna debe contener valores atómicos e indivisibles, y no deben existir grupos repetidos de columnas ¹⁶. Es decir, no hay columnas multivaluadas ni estructuras compuestas; cada celda tiene un solo valor.
- **2FN (Segunda Forma Normal):** Además de cumplir 1FN, todos los atributos que no forman parte de la clave primaria deben depender de la clave primaria completa, no de una parte. No puede haber dependencias funcionales parciales respecto a una clave primaria compuesta ¹⁹. Si una tabla en 1FN tiene clave primaria única (un solo atributo), ya está en 2FN por defecto.
- **3FN (Tercera Forma Normal):** Además de 2FN, no debe haber **dependencias transitivas**. Ningún atributo no clave puede depender funcionalmente de otro atributo no clave ²⁰. Esto elimina redundancias adicionales (por ejemplo, separar ciudad y provincia en tablas distintas si ciudad determina provincia).

En resumen, la normalización busca que cada tabla almacene información relacionada de forma mínima y sin repeticiones innecesarias ²⁰. En la práctica, casi todos los diseños se llevan a 3FN, que se considera "mejor" que 2FN y 1FN porque reduce al máximo la redundancia ²¹.

En algunos casos, por motivos de rendimiento (consultas frecuentes), se aplica la **desnormalización**: duplicar intencionalmente algunos datos o combinar tablas para reducir JOINs ²². La desnormalización aumenta la redundancia pero mejora la velocidad de lectura al evitar varias tablas, aunque debe hacerse con cuidado para no comprometer la integridad.

Diccionario de Datos

Un **diccionario de datos** (o de base de datos) es un repositorio centralizado que documenta todos los elementos del modelo relacional ²³. Describe las **tablas**, **columnas**, **tipos de datos** y **relaciones** del esquema de la base de datos ²³, de modo que desarrolladores y administradores comprendan su estructura. Esencialmente, actúa como una "hoja de ruta" de la base de datos.

Por ejemplo, el diccionario incluye: nombres y descripción de cada **tabla**, nombres de **columnas** con su tipo (INT, VARCHAR, DATE, etc.) y descripción, las **relaciones** entre tablas (PK-FK) y las **restricciones** aplicadas (clave primaria, unicidad, validaciones, valores por defecto, etc.) ²⁴. A modo ilustrativo, podría documentarse:

- **Tabla:** CLIENTE – Almacena información de los clientes.
- **Columnas:** id_cliente (INT, PK, no nulo), nombre (VARCHAR), email (VARCHAR, único), teléfono (VARCHAR),
- **Relaciones:** id_cliente es PK, y puede ser FK en tabla PEDIDO.
- **Restricciones:** PK(id_cliente), UNIQUE(email), NOT NULL(nombre).

Contar con este diccionario de datos garantiza una interpretación consistente del esquema ²³ y sirve de referencia (documentación) al diseñar consultas, migraciones o al incorporar nuevo personal.

Fuentes: Definiciones de modelo E-R, relaciones y transformación al modelo relacional ¹ ²⁵; ejemplos de reglas de conversión relacional ¹³ ¹⁴ ¹⁵; normalización de datos en 1FN–3FN ¹⁶ ²⁰; documentación de diccionario de datos ²³ ²⁴. All charts and images are illustrative (commons).

1 2 3 4 5 6 8 11 Diseño Conceptual - Modelo Entidad Relación - Bases de datos

<https://aitor-medrano.github.io/bd/02er.html>

7 ▷ Diagrama entidad-relación (E/R). Teoría y ejemplos

<https://diagramasuml.com/%E2%96%B7-diagrama-entidad-relacion-e-r-teoria-y-ejemplos/>

9 10 dane.gov.co

<https://www.dane.gov.co/files/sen/lineamientos/Recomendaciones-Guiaconrecomendacionesparaelaborarmodelosentidad-relacion.pdf>

12 13 14 15 25 Unidad 03. Del modelo conceptual al modelo relacional

<https://josejuansanchez.org/bd/unidad-03-teoria/index.html>

16 19 20 21 22 Normalización de Bases de Datos

https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/872/mod_resource/content/7/Contenido/index.html

17 18 Tema 3

<https://w3.ual.es/~mtorres/BD/bdt3>

[23](#) [24](#) Diccionario de Base de Datos: Definición e Importancia

<https://www.datasunrise.com/es/centro-de-conocimiento/diccionario-de-base-de-datos/>