Logotipo

El contenido generado por IA puede ser incorrecto.

Santiago Yosa González

**Ingeniero Sistemas de Información**



Logotipo

El contenido generado por IA puede ser incorrecto.

**Conceptos Generales Bases de Datos.**

* Conceptos generales bases de datos.
* Diagramación de bases de datos
* Modelos relacionales.
* Identidad-relación

**Unidad I**

# Situación:

Imaginen que son dueños de una tienda de barrio.

* **Escenario:** Llega un proveedor y les deja 50 productos. Llega un cliente y pide fiado. Llega otro y paga.

¿Cómo anotas esto en un cuaderno para que no se pierda nada?

* **Demo:** Escribamos en una hoja (Excel/Cuaderno), ¿Cómo llevaríamos el listado y seguimiento de las compras y de los productos que llegan? Anotemos un listado de pedidos…

**Análisis Crítico:**

¿Si quiero saber cuánta leche vendí en total el último mes, cuánto tiempo tardas en darme el dato con tu cuaderno?

Tener "Datos anotados" no es lo mismo que tener "Información gestionable".

**Solución:** Necesitamos romper esa hoja de cálculo en pedazos lógicos que se relacionen entre sí. Necesitamos una **Base de Datos Relacional**.

# Conceptos Generales: Dato vs. Información vs. Base de Datos

**Dato:** Es la unidad mínima sin contexto. (Ej: "Azul", "1.80", "Santiago"). Por sí solo es ruido.

**Información:** Es el dato procesado y con contexto que permite tomar decisiones. (Ej: "Santiago mide 1.80m").

**Base de Datos (DB):** No es el software. Es el *conjunto organizado* de datos pertinentes a un contexto. Una guía telefónica es una base de datos (análoga).

**SGBD (Sistema de Gestión de Bases de Datos):** Es la herramienta (software) que usaremos después para administrar esa base de datos. Hoy somos nosotros el SGBD manual.

# ¿Qué es una Base de Datos?

Una base de datos (BD) es un conjunto organizado de información que se almacena de manera estructurada para facilitar su acceso, administración y actualización. Se utilizan en diversas aplicaciones, desde sistemas bancarios hasta redes sociales y tiendas en línea.

**Propósito y aplicaciones de una Base de Datos**

El propósito principal de una base de datos es almacenar, organizar y gestionar datos de forma eficiente. Algunos ejemplos de aplicaciones son:

**Empresas**: Gestión de clientes, ventas, empleados.

**Hospitales**: Historial clínico de pacientes.

**Universidades**: Información de estudiantes, cursos, calificaciones.

Pero aquí nos preguntaríamos:

¿Qué base de datos usar?

¿Cómo escoger?

¿Qué las diferencia?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Característica | NoSQL | SQL Server | Excel | Power BI |
| Uso principal | Gestión de datos no relacionales, escalabilidad horizontal | Gestión de bases de datos relacionales | Hoja de cálculo, análisis de datos a pequeña escala | Visualización y análisis de datos, inteligencia empresarial |
| Escala de datos | Escalabilidad horizontal masiva | Grandes volúmenes de datos | Limitado (aprox. 1 millón de filas) | Grandes volúmenes de datos (conectado a fuentes externas) |
| Rendimiento | Alto rendimiento para operaciones de lectura/escritura, especialmente con datos no estructurados | Alto rendimiento, optimizado para operaciones complejas | Rendimiento decrece con grandes conjuntos de datos | Depende de la fuente de datos, optimizado para visualización |
| Acceso concurrente | Excelente soporte para alta concurrencia | Soporte robusto para múltiples usuarios | Limitado, problemas de integridad con múltiples usuarios | Soporte para múltiples usuarios (depende de la fuente de datos) |
| Integridad de datos | Consistencia eventual, CAP theorem (variedad de modelos) | Restricciones, transacciones, control de concurrencia (ACID) | Propenso a errores humanos | Depende de la fuente de datos, seguridad a nivel de datos y usuarios |
| Seguridad | Depende del sistema NoSQL específico | Seguridad granular a nivel de usuario y objeto | Protección con contraseña limitada | Seguridad a nivel de datos y usuarios |
| Lenguaje de consulta | Varía (JSON, APIs, etc.) | SQL | Funciones de hoja de cálculo | DAX |
| Manipulación de datos | Modelos de datos flexibles (documentos, clave-valor, grafos, etc.) | Operaciones complejas, uniones, agregaciones | Funciones de hoja de cálculo limitadas | Transformación de datos, modelado |
| Automatización | APIs, scripts, herramientas de administración | Procedimientos almacenados, desencadenadores, trabajos | Macros de VBA limitadas | Actualización automatizada de datos |
| Programación | APIs, SDKs para varios lenguajes | Integración con C#, Python, etc. | Macros de VBA | Limitado, enfoque en visualización |
| Visualización | Depende de herramientas externas o APIs | Limitada | Gráficos básicos | Visualizaciones interactivas y avanzadas |
| Idoneidad | Aplicaciones web, móviles, big data, tiempo real, datos no estructurados | Entornos empresariales con grandes volúmenes de datos estructurados | Análisis de datos personales o a pequeña escala | Informes y paneles de control, análisis de tendencias |

# Tipos de Bases de Datos

Existen distintos tipos de bases de datos, pero las más comunes son:

**Relacionales:** Organizan datos en tablas con relaciones entre ellas (Ejemplo: MySQL, PostgreSQL, SQL Server).

**No Relacionales (NoSQL):** Almacenan datos en estructuras más flexibles como documentos, claves-valor o grafos (Ejemplo: MongoDB, Firebase).

**Jerárquicas:** Organizan la información en una estructura de árbol (Ejemplo: bases de datos en mainframes).

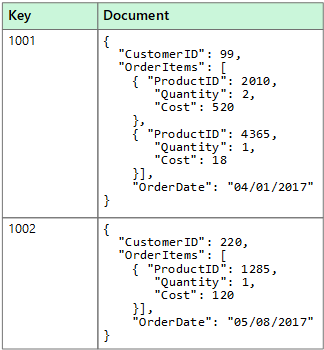
**En Red:** Similar a la jerárquica, pero con relaciones más complejas entre registros.

Fuentes:

[Diferencias entre relacionales y no relacionales](https://aws.amazon.com/es/compare/the-difference-between-relational-and-non-relational-databases/)

[Tipos de bases de datos](https://www.datacamp.com/es/blog/types-of-databases-overview)

[Sql vs nosql databases](https://www.datacamp.com/es/blog/sql-vs-nosql-databases)



**No Relacionales (NoSQL):** MongoDB, Apache Cassandra, CouchDB, Redis, Neo4j, BigTable, HBase, Elasticsearch.

**Fuente:**

[non-relational-data](https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data)

Diagrama

El contenido generado por IA puede ser incorrecto.

# Componentes Clave de una Base de Datos Relacional

**Tablas**: Son la estructura principal donde se almacenan los datos.

**Registros**: Cada fila dentro de una tabla representa un conjunto de datos relacionados.

**Campos**: Cada columna dentro de una tabla representa un atributo o propiedad de los datos.

**Llave Primaria**: Un identificador único para cada registro en una tabla.

**Llave Foránea**: Un campo que establece una relación con otra tabla.

# El Modelo Entidad-Relación (MER)

El MER es un lenguaje gráfico universal. Es el "plano del arquitecto".

**La Entidad (El Sustantivo)**

Es cualquier objeto, persona, lugar, concepto o evento del mundo real sobre el cual *necesitamos* guardar datos.

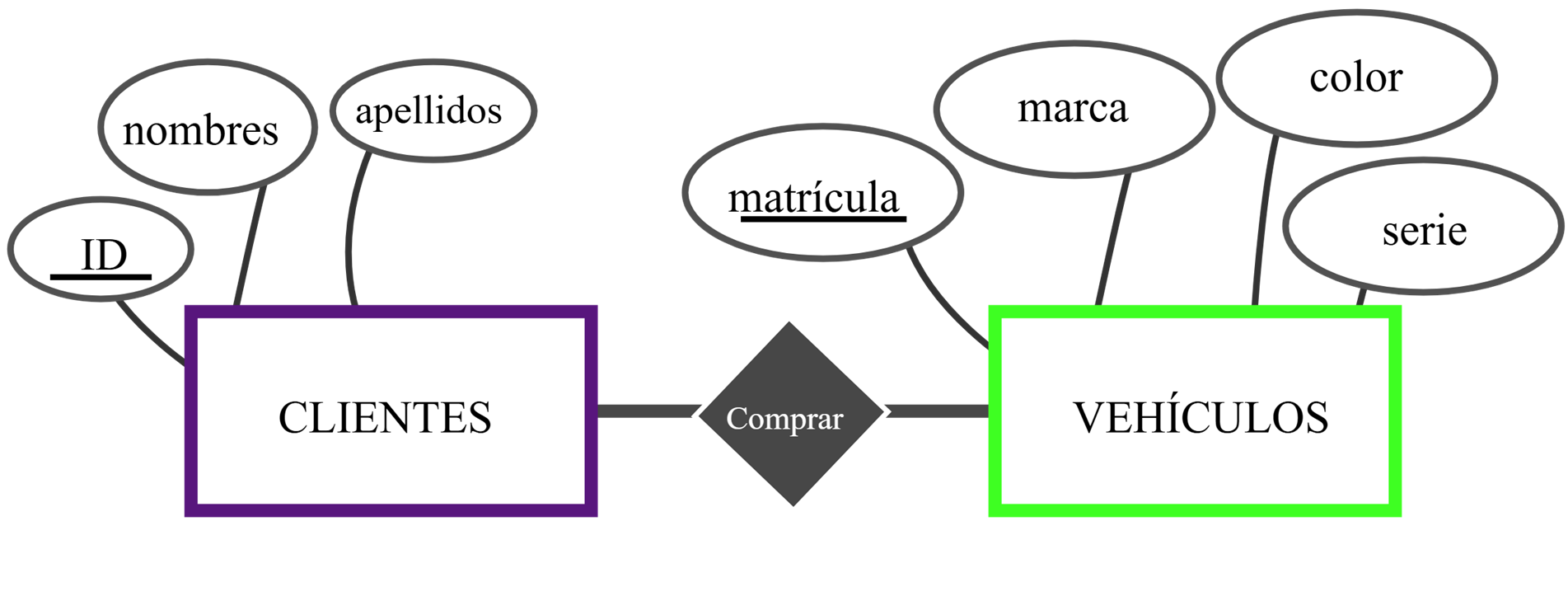
**La prueba de fuego:** ¿Tiene características propias? Si la respuesta es sí, es una Entidad.

* ***Ejemplos:*** Un Coche (físico), un Vuelo (evento), una Materia (concepto).
* ***Representación Gráfica:*** Un **Rectángulo**.

**El Atributo (El Adjetivo)**

Son las características o propiedades que describen a una Entidad. Son los datos atómicos.

* ***Ejemplos:*** Para la entidad Coche: *Color, Marca, Modelo, Placa*.
* ***Representación Gráfica:*** Un **Óvalo** o círculo conectado a la entidad.



**La Relación (El Verbo)**

Es la asociación natural que existe entre dos entidades. Describe cómo interactúan.

* ***Ejemplo:*** Un Estudiante *cursa* una Materia. El verbo "cursar" es la relación.
* ***Representación Gráfica:*** Un **Rombo** que une dos rectángulos.

**Del Diagrama al Modelo Relacional (Nivel Lógico)**

Aquí explicamos cómo esos dibujos se convierten en "Tablas" mentalmente.

* **Entidad** = Se convierte en **Tabla**.
* **Atributo** = Se convierte en **Columna** (Campo).
* **Instancia** = Cada ocurrencia real (fila) se llama **Registro** o **Tupla**.

# Tipos de Relaciones en un Modelo Relacional

**1 a 1 (1:1):**

Un registro en una tabla se relaciona con un solo registro en otra tabla.

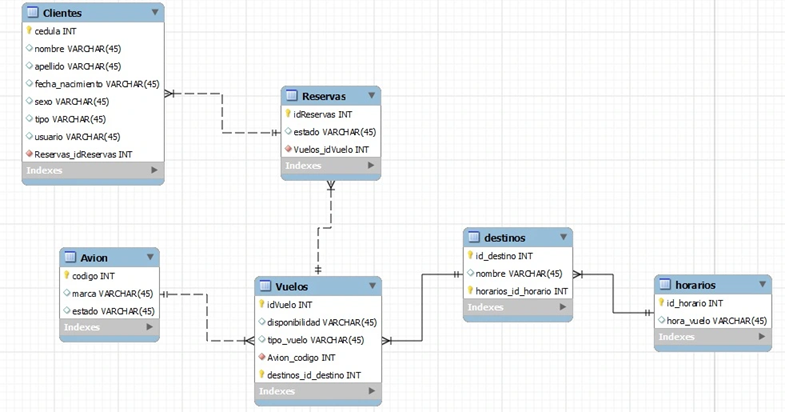
**1 a muchos (1:M):**

Un registro en una tabla se relaciona con varios registros en otra (Ejemplo: un cliente puede hacer varios pedidos).

**Muchos a muchos (M:M):**

Varios registros de una tabla se relacionan con varios registros en otra (Ejemplo: estudiantes y materias, ya que un estudiante puede inscribirse en varias materias y una materia puede tener varios estudiantes).

[Cardinalidad](https://informaticosinlimites.com/base-de-datos/cardinalidad/)



# Representación Gráfica de un DER

Las entidades se representan con rectángulos.

Las relaciones se representan con rombos y conectan entidades.

Los atributos se colocan en óvalos alrededor de las entidades.

Las llaves primarias se subrayan para indicar que identifican de manera única a la entidad.

¿Dónde realizar?

<https://app.diagrams.net/>

<https://www.lucidchart.com/pages/es>

<https://miro.com/es/diagrama/>

# Preguntas Frecuentes

**¿Cómo sé si algo es una Entidad o un Atributo?**

**Pregúntale al dato:** "¿Tengo información ADICIONAL sobre ti?".

**Ejemplo:** "Color". ¿Necesito guardar información sobre el "Rojo"? No. -> Es un Atributo.

Ejemplo: "Fabricante". ¿Necesito guardar información sobre "Toyota" (dirección, teléfono, país)? Sí. -> Entonces "Fabricante" es una Entidad, no un simple texto.

**¿Puedo poner dos atributos iguales en una entidad? (Ej: Teléfono1, Teléfono2)**

Conceptualmente es una mala práctica llamada "grupos repetitivos". Si un cliente tiene 10 teléfonos, ¿crearás 10 columnas? No. Eso nos indica que "Teléfono" quizás debería ser tratado de forma especial (pero por hoy, lo dejaremos como atributo simple, advirtiendo que esto evolucionará).

# Demo La Biblioteca de Barrio

**Paso 1: Narrativa (“El Requerimiento”)**: Necesitamos organizar los libros. De cada libro nos interesa el título y el número de páginas. También tenemos usuarios inscritos con nombre y dirección. Los usuarios sacan libros prestados.

**Paso 2: Identificación**

* Sustantivos importantes: Libros, Usuarios. -> **Entidades**.
* Adjetivos/Características: Título, Páginas, Nombre, Dirección. -> **Atributos**.
* Verbos: Sacan (Prestar). -> **Relación**.

**Paso 3: Diagramación**

1. Representa a **LIBRO**.
   * Saca líneas con óvalos: Título, Páginas.
2. Representa a **USUARIO**.
   * Saca líneas con óvalos: Nombre, Dirección.
3. Representa a **PRESTAR**.
   * Une LIBRO con PRESTAR.
   * Une USUARIO con PRESTAR.

**Paso 4: Traducción a Tablas (Lógico)**

* Tabla: **LIBROS**. Columnas: Título, Páginas.
* Tabla 2: **USUARIOS**. Columnas: Nombre, Dirección.

# Práctica Acompañada - La Liga de Fútbol

Vamos a modelar una liga de fútbol o voleibol pequeña.

1. Tenemos **Equipos**. De cada equipo queremos saber su Nombre y su Año de Fundación.
2. Tenemos **Jugadores**. Queremos saber su Nombre, Dorsal (número de camiseta) y Posición.
3. Los Jugadores *juegan* en los Equipos.

**Solución paso a paso:**

1. **Pregunta:** ¿Cuáles son los rectángulos?
   * EQUIPO y JUGADOR.
2. **Pregunta:** ¿Qué va dentro de los óvalos de EQUIPO?
   * Nombre, AñoFundacion.
3. **Pregunta:** ¿Cuál es el rombo?
   * JUGAR (o PERTENECER).
4. **Reto conceptual:** "¿Dónde anoto que Santiago juega en el Real Madrid?".

En el diagrama conceptual solo decimos que EXISTE una relación. Aún no definimos CÓMO se anota técnicamente (eso serán las llaves foráneas en el futuro.

# Refuerzo Independiente

**Enunciado:** Modela una **Academia de Música**.

* Identifica al menos 2 Entidades principales.
* Asigna al menos 2 atributos lógicos a cada una.
* Establece la relación entre ellas.
* *Pista:* Piensa en quién enseña y qué se enseña.

**Pasos mentales a realizar:**

1. Detectar actores: Profesores? Instrumentos? Clases? Alumnos?
2. Seleccionar los más obvios.
3. Dibujar.

# ¿Qué son las bases de datos NoSQL?

Las bases de datos NoSQL, también conocidas como bases de datos personalizadas, están diseñadas para modelos de datos específicos y almacenan los datos en esquemas flexibles que se escalan con facilidad para aplicaciones modernas. Muchas cargas de trabajo de bases de datos pueden beneficiarse de la rentabilidad y el rendimiento de las bases de datos NoSQL. Por ejemplo, Amazon DynamoDB no tiene servidor, por lo que la utilización de los recursos se optimiza automáticamente y nunca tendrá que pagar por exceso de aprovisionamiento. Además, las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar.

**Fuente:**

<https://aws.amazon.com/es/nosql/>

<https://www.ibm.com/es-es/think/topics/nosql-databases>

# Terminología de SQL en comparación con NoSQL

|  |  |
| --- | --- |
| SQL | MongoDB |
| Tabla | Conjunto |
| Fila | Documento |
| Columna | Campo |
| Clave principal | ObjectId |
| Índice | Índice |
| Ver | Ver |
| Tabla u objeto anidado | Documento incrustado |
| Matriz | Matriz |

# ¿Debería saber algo más?

Hoy hemos visto un mundo ideal. En las próximas sesiones veremos que:

1. A veces un atributo debe ser único (como el serial del computador) -> Introducción futura a **Llaves Primarias**.
2. Eventualmente tendremos que decirle al computador que "Serial" es un número y "Marca" es texto -> Introducción futura a **Tipos de Dato**.

Logotipo

El contenido generado por IA puede ser incorrecto.