

INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

PROGRAMACIÓN: DESARROLLO BACKEND

SQL

Bases de Datos y Backend

# Contenido

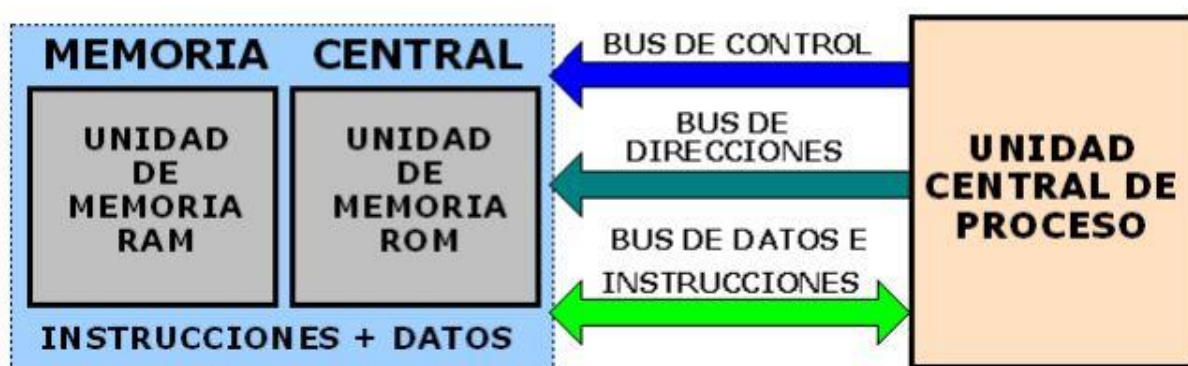
Introducción a las Bases de Datos .....	2
Tipos de Bases de Datos.....	2
Representación de las Bases de Datos: Nomenclatura de Chen .....	3
Ejemplo de Base de Datos: Blog Post.....	8
Referencias.....	11



# Introducción a las Bases de Datos

Las bases de datos ayudan a complementar la **arquitectura de Von Neumann**, que es la arquitectura utilizada en ordenadores, la cual **a diferencia de la arquitectura Harvard utilizada en microcontroladores, utiliza una memoria centralizada para realizar sus funciones**. La necesidad de extender la capacidad de la memoria central es la de conservar los datos más allá de la memoria RAM o ROM, ya que en la arquitectura Von Neumann si se contempla el procesamiento de datos, pero no el almacenamiento de datos persistentes, por eso es que es de suma importancia la utilización de las bases de datos.

## ARQUITECTURA VON NEUMANN



Para resolver esta situación, donde se busca que de una forma fácil se pudiera guardar y extraer la información, se concluyeron dos soluciones:

- **Bases de datos basadas en archivos:** Este método de almacenamiento de datos persistentes consiste en guardar datos en un archivo de texto plano, hojas de cálculo, etc. usualmente separados por comas o de alguna otra forma ordenada.
- **Bases de datos basadas en documentos:** En este tipo de base de datos, la unidad básica de almacenamiento es el documento, que puede contener datos en forma de texto, números, listas, objetos JSON (JavaScript Object Notation) y a veces incluso otros documentos anidados.

## Tipos de Bases de Datos

Los diferentes tipos de bases de datos existentes son los siguientes:

- **Relacionales o RDB:** Son bases de datos basadas en documentos que se rigen por las 12 reglas de Edgar Codd, que dan como resultado el álgebra relacional, a través de las cuales se indican las reglas con las que los datos de las RDB se pueden mezclar o relacionar entre sí.
  - **Privadas:** Microsoft SQL Server, Oracle, etc.
  - **Open Source:** PostgreSQL, MySQL, MariaDB, etc.

## Ejemplos de bases de datos relacionales



- No relacionales:
  - Memcached, Cassandra (Facebook), DynamoDB, ElasticSearch, BigQuery, Neo4j (GraphQL), MongoDB, Firestore (Firebase).

## Bases de datos no relacionales



- Auto Administrados: En este tipo de bases de datos se instala, actualiza y mantiene el software en un ordenador de forma local y la consistencia de datos se realiza de forma manual.
- Administrados: Este tipo de base de datos se ofrece por las nubes modernas como las proporcionadas por Amazon, Google, Azure (Microsoft), para ello la instalación no se realiza de forma local y por lo tanto, no se mantiene la consistencia de datos de forma manual, sino que se realiza de forma automática por el servicio de la nube.

## Representación de las Bases de Datos: Nomenclatura de Chen

- **Entidad:** Una entidad es algo muy similar a un objeto, el cual se puede asociar con ciertos **atributos (características)** y **métodos (funciones)**, de la misma forma como se maneja en POO.
  - **Atributo:** Se representa por medio de un **óvalo simple** cuando la entidad solo posee **uno solo de ese atributo**, si cuenta con más de uno, esto se indica con **dos óvalos anidados** que rodeen el nombre del atributo, a esto se le llama **atributo multivalor**.
    - **Ejemplo 1:** Cualquier automóvil posee un solo volante, pero varias llantas, por lo cual el atributo “volante” será rodeado por un óvalo simple y el atributo “llantas” se rodeará de un óvalo doble.

- **Ejemplo 2:** Ahora se representará a través de un diagrama de Chen las entidades (objetos) laptops, donde cabe mencionar que **los atributos donde se subraye su nombre** se llaman **atributos clave** y diferencian cada laptop individualmente (instancia), además los atributos que tengan un **óvalo con línea punteada** representan los **atributos derivados**, que corresponden a datos que se pueden obtener a través de otros y además que los atributos pueden tener otros atributos relacionados.

## Entidades



## Atributos



## Atributos

no de serie	color	año	pantalla
LKJ789JKAS	gris	2017	AX4829i
KCO3100KJH	negro	2019	AX4930i
NSDJOIH128	negro	2018	AX4930i
09KSIHBD71	gris	2017	AX4829i

- **Tipos de atributos clave:** Los atributos clave pueden ser naturales, esto significa que son pertenecientes al objeto y no se pueden remover y los atributos clave artificiales, que se asignan de manera arbitraria.
- **Entidades fuertes:** No dependen de otra entidad para existir, estas se rodean de un cuadrado simple.
  - **Entidades débiles:** Sí dependen de otra entidad para existir, estas se rodean de un cuadrado doble, así como los atributos multivalor. Además, existen dos tipos de debilidad:
    - Debilidad por identidad: Que para que se puedan diferenciar, deben tomar el atributo clave de la entidad de la que dependen.
    - Debilidad por existencia: Que pueden tener un identificador propio, pero aun así dependen de otra identidad para existir.

## Entidades débiles



## Entidades débiles: identidad

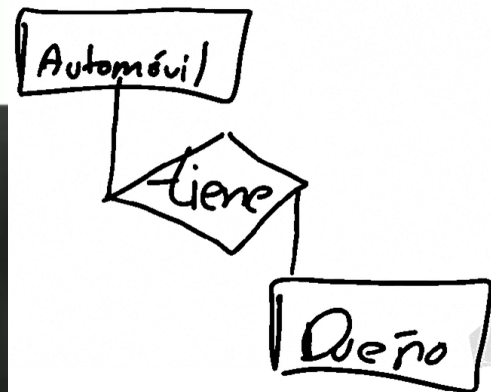
Libros			Ejemplares		
id	título	...	libro_id	localización	edición
LKJ789JKAS	Viaje al cent...	...	LKJ789JKAS	pasillo 1	1
KCO3100KJH	El señor de ...	...	KCO3100KJH	pasillo 1	1
NSDJOIH128	De la tierra...	...	NSDJOIH128	pasillo 1	3
09KSIHBD71	Amor en tie...	...	09KSIHBD71	pasillo 1	1

## Entidades débiles: existencia

Libros			Ejemplares		
id	título	...	id	localización	edición
LKJ789JKAS	Viaje al cent...	...	JKE7823CLK	pasillo 1	1
KCO3100KJH	El señor de ...	...	JKFE1093JD	pasillo 1	1
NSDJOIH128	De la tierra...	...	82938ISHDIK	pasillo 1	3
09KSIHBD71	Amor en tie...	...	838439JHDUI	pasillo 1	1

- **Relación:** Es la conexión con las que se ligan las diferentes entidades entre sí, para ello dentro de las relaciones se utilizan verbos que conecten una relación con la otra.

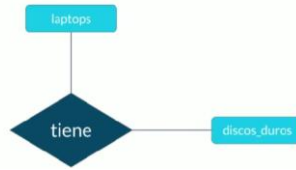
## Relaciones



- **Cardinalidad:** Una peculiaridad de las relaciones es que a través de ellas se deben separar los atributos multivaluados, ya que cada uno puede tener características específicas y se relacionan con el concepto de cardinalidad porque este se relaciona con el número de veces que se repite un atributo en una entidad.
  - Cuando se utiliza la cardinalidad se utiliza todo el concepto del verbo de conexión que utiliza la relación para indicar cual es el número de entidades

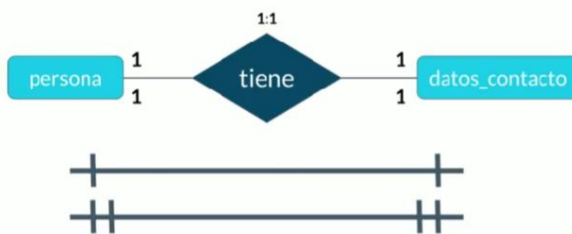
con las que cuenta una entidad, de esta manera es como se puede separar los atributos multivaluados.

#### Relaciones

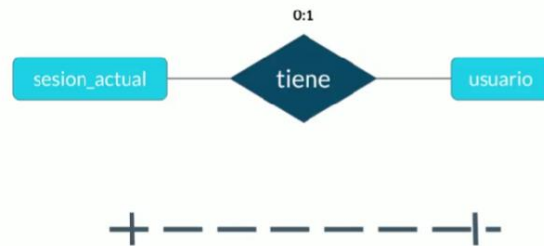


- De igual manera se obtiene la cardinalidad de ambos lados, ya que esto puede variar cuando se ve desde perspectivas distintas. Además, se maneja cierta nomenclatura en el diagrama para denotarlo.

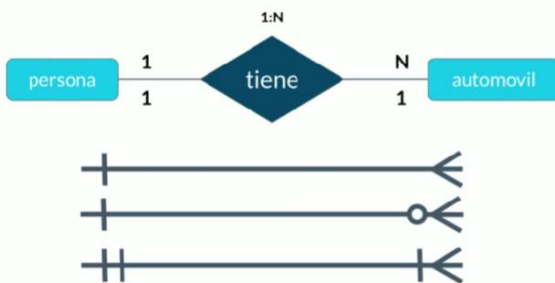
#### Cardinalidad: 1 a 1



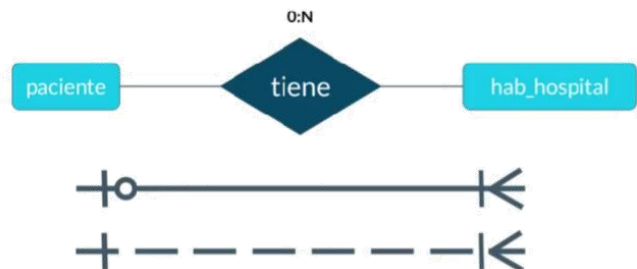
#### Cardinalidad: 0 a 1



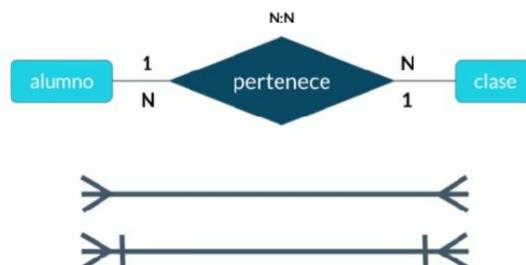
#### Cardinalidad: 1 a N



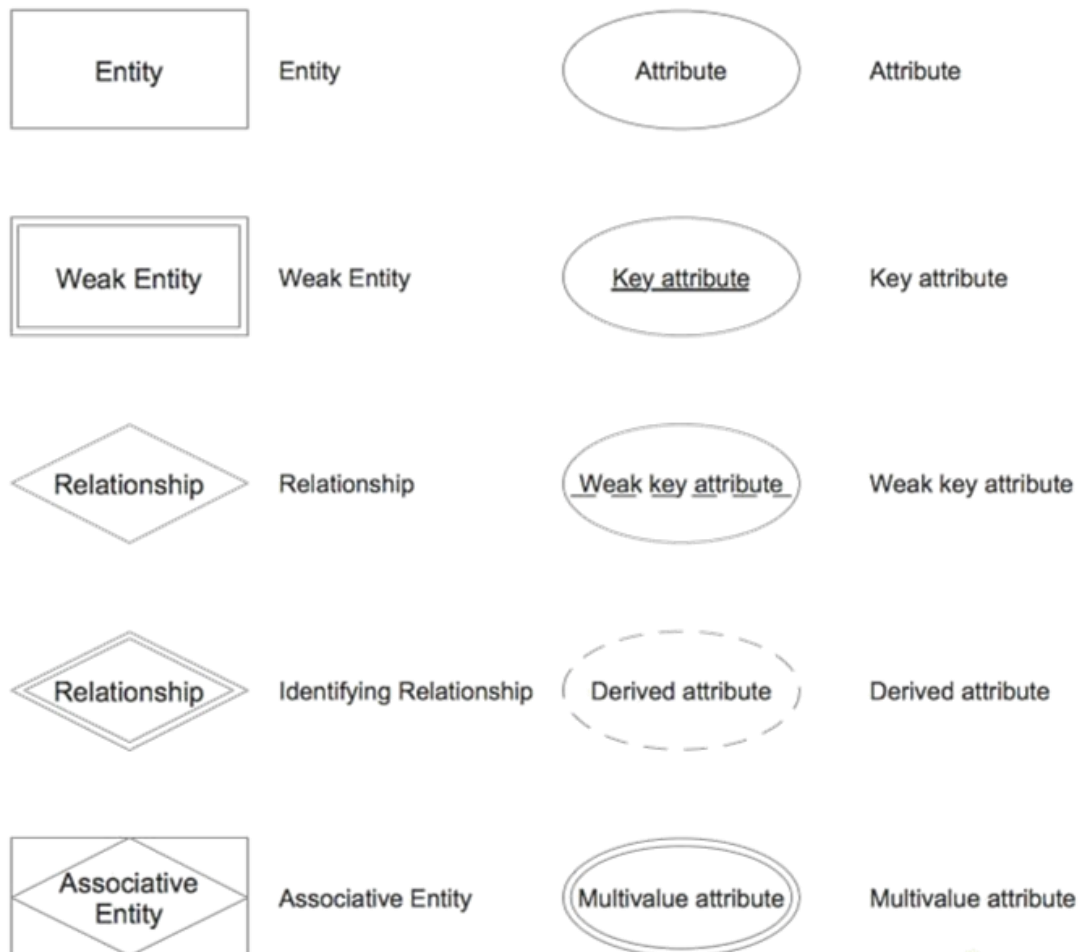
#### Cardinalidad: 0 a N



#### Cardinalidad: N a N



Todos los conceptos explicados previamente que describen los datos almacenados en una base de datos relacional se deben plasmar en un diagrama ER (entidad-relación), para ello se utiliza la siguiente simbología:



Además del diagrama ER (entidad-relación) existe otro llamado diagrama físico, que se deriva del ER, el cual es más específico ya que menciona los tipos de datos, que pueden ser los siguientes:

- Texto:
  - Char(n): Minimiza el espacio de memoria a solo los caracteres que ocupa el texto.
  - VarChar(n): Utiliza el espacio de memoria de forma dinámica, reservando como mínimo un espacio de memoria y extendiéndolo si es necesario hasta 255 caracteres.
  - Text: Reserva el espacio de memoria para cadenas de caracteres (palabras u oraciones) muy grandes.
- Números:
  -
- Fecha/Hora:
  -
- Lógicos:
  -



## Tipos de dato

Texto	Números	Fecha/hora	Lógicos
CHAR(n)	INTEGER	DATE	BOOLEAN
VARCHAR(n)	BIGINT	TIME	
TEXT	SMALLINT	DATETIME	
	DECIMAL(n, s)	TIMESTAMP	
	NUMERIC (n, s)		

## Ejemplo de Base de Datos: Blog Post

Para crear una base de datos, primero que nada, debemos pensar en las **entidades** que se utilizaran en ella y posteriormente deberemos pensar en los **atributos** que le pertenecen, **todo esto se coloca en un diagrama ER (entidad-relación) para modelar cómo los datos se relacionan entre sí**. En el caso del blog post son los siguientes:

- **Entidades:**
  - Posts (Publicaciones).
  - Usuarios.
  - Comentarios.
  - Categorías.
  - Etiquetas.

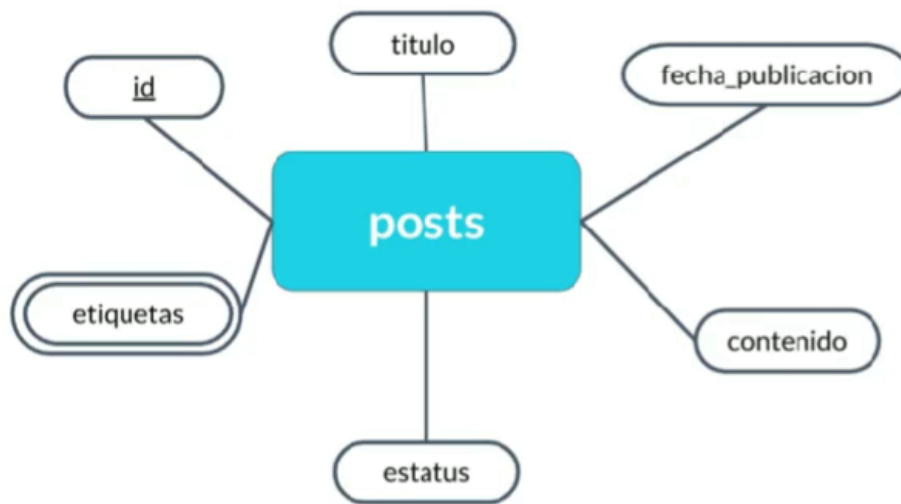
## Diagrama ER: Platziblog



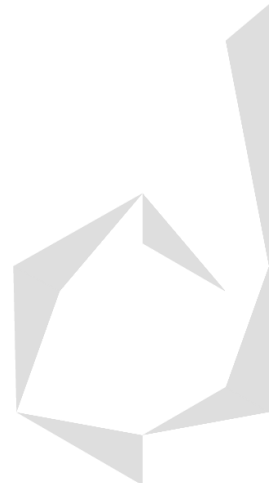
- **Atributos de las entidades:**
  - **Atributos Entidad Posts:**
    - Título.
    - Fecha\_publicacion.
    - Contenido.
    - Estatus (Activo, Inactivo, etc.).
    - Etiquetas (Categoría interna).
    - Id (Clave única).

---

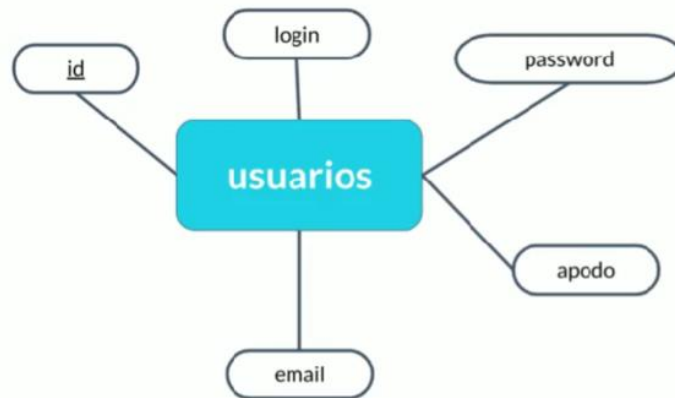
## Entidades Platziblog



- **Atributos Entidad Usuarios:**
  - Login (Nombre de Usuario).
  - Password.
  - Nickname.
  - Email.
  - Id.

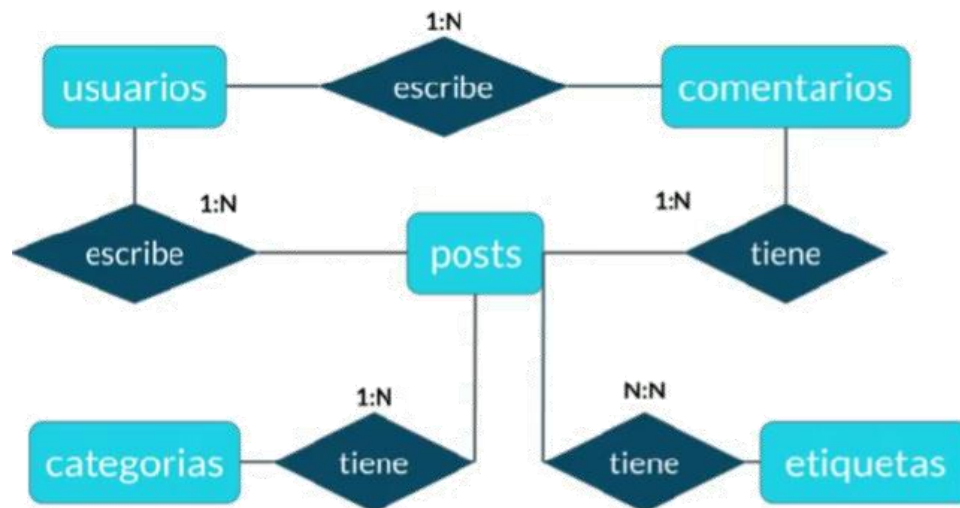


## Entidades Platziblog



- **Diagrama ER (Entidad-Relación):**
  - **Relaciones y cardinalidad:**
    - Un usuario tiene (puede escribir) varios posts.
    - Un usuario tiene (puede escribir) varios comentarios.
    - Un post tiene varios comentarios.
    - Una categoría tiene (engloba) varios posts.
    - Un post tiene varias etiquetas y una etiqueta tiene (engloba) varios posts.

## Diagrama ER: Platziblog



○

# Referencias

Platzi, Israel Vázquez, “Curso de Fundamentos de Bases de Datos”, 2018 [Online], Available: [https://platzi.com/new-home/clases/1566-bd/19781-bienvenida-conceptos-basicos-y-contexto-historico-/](https://platzi.com/new-home/clases/1566-bd/19781-bienvenida-conceptos-basicos-y-contexto-historico/)

