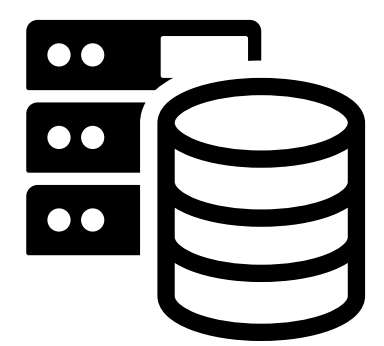




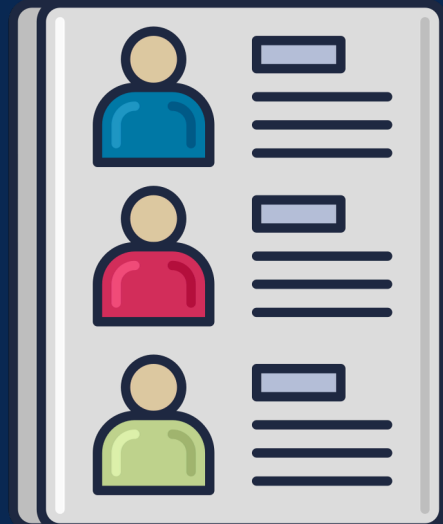
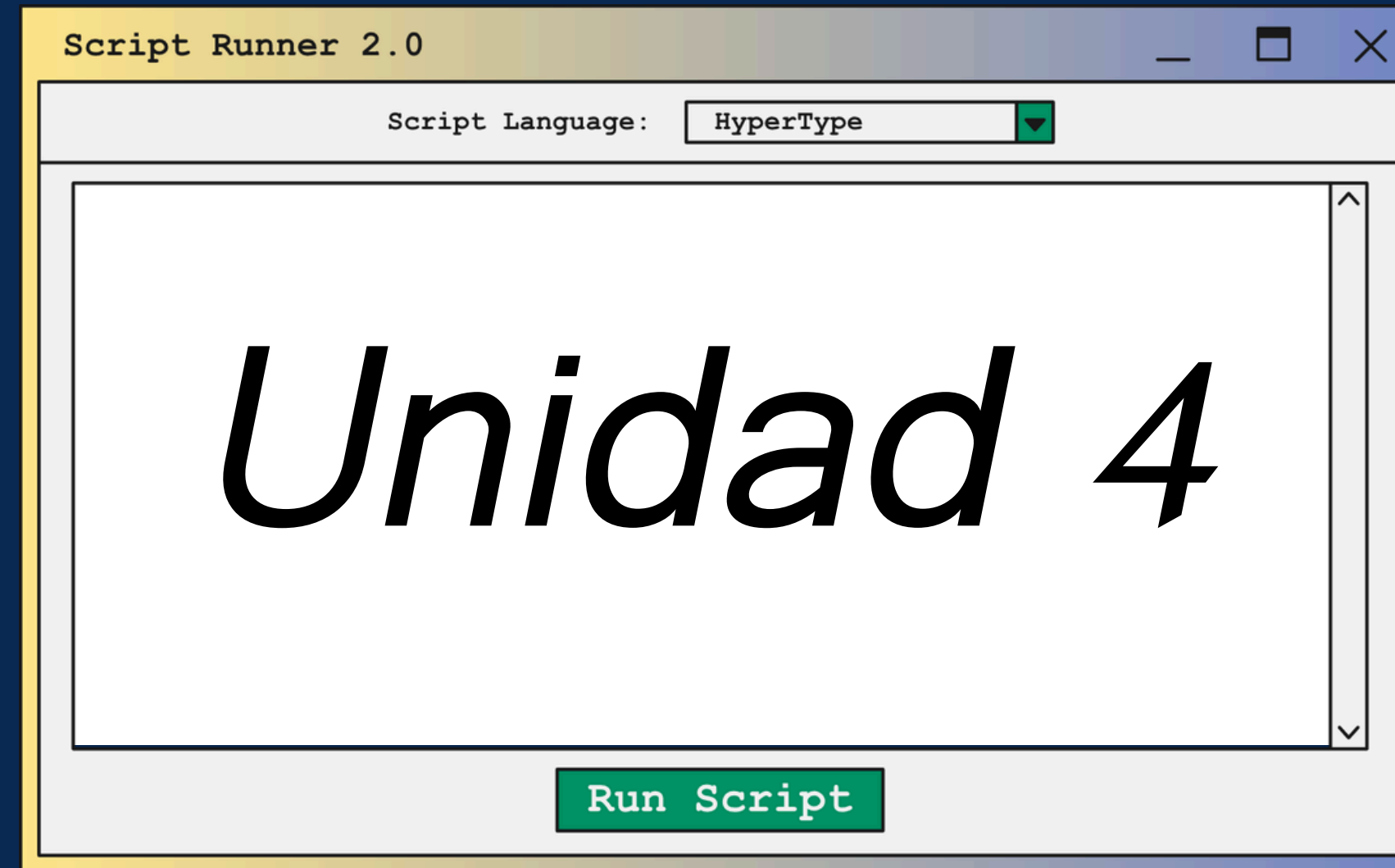
PROGRAMACION AVANZADA

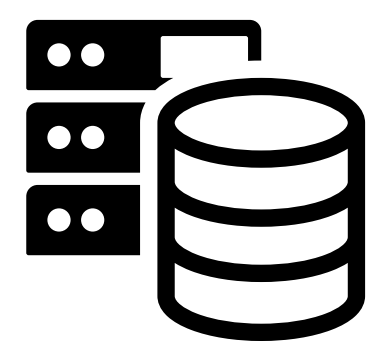
ING. INDUSTRIAL

Esp. Ing.Prof.: Cristina Delia Cruz



BASE DE DATOS

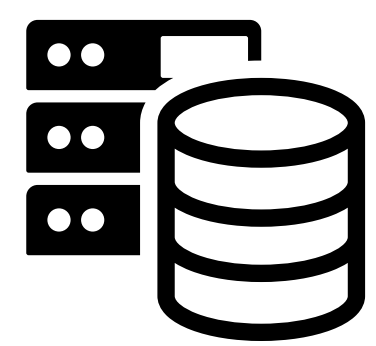




BASE DE DATOS

Conceptos Básicos

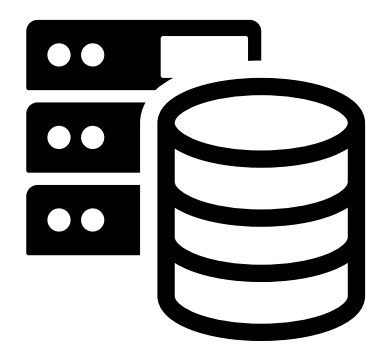
- **Dato:** Es un valor simple y sin procesar, que por sí mismo no tiene significado.
Ejemplo: "25", "Juan".
- **Información:** Es el resultado de procesar y organizar datos de forma que tengan un significado útil. Ejemplo: "Juan tiene 25 años".
- **Campo:** Es la unidad básica de almacenamiento en una base de datos, donde se almacena un dato en particular dentro de un registro. Ejemplo: "Nombre", "Edad" en una tabla de personas.
- **Registro:** Es un conjunto de campos relacionados entre sí que representan una entidad completa. Ejemplo: En una tabla de personas, un registro sería todos los datos de una persona (nombre, edad, dirección).
- **Archivo:** Es una colección de registros almacenados en un medio de almacenamiento. En una base de datos, un archivo puede ser una tabla completa.



CONCEPTO DE BASE DE DATOS

*Una base de datos **es un conjunto organizado de datos que se almacenan y gestionan de manera eficiente para permitir su acceso, manipulación y actualización.** Estos datos se estructuran de acuerdo con un esquema (como tablas, en el caso de bases de datos relacionales) y se pueden consultar para extraer información relevante.*

Las bases de datos permiten almacenar grandes volúmenes de información, garantizando su integridad, seguridad y fácil acceso mediante lenguajes como SQL (Structured Query Language).



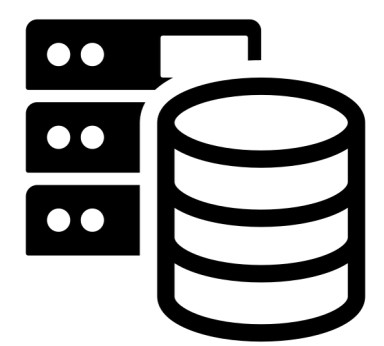
CONCEPTO DE BASE DE DATOS

Base de Datos (BD): Una Base de Datos es un conjunto de datos relacionados entre sí que están organizados y almacenados de manera sistemática para permitir su fácil acceso, manipulación y actualización. Las bases de datos están diseñadas para gestionar grandes cantidades de información y pueden estar formadas por varias tablas, registros, campos y otros elementos que permiten estructurar los datos de manera eficiente.

Esquema de la Base de Datos (Esquema de la BD): El esquema de una base de datos se refiere a la estructura o diseño lógico que define cómo se organiza la base de datos. Es la "plantilla" que describe las tablas, los campos, las relaciones entre las tablas, las restricciones (como las claves primarias y foráneas), y otros elementos como vistas, índices y procedimientos almacenados. El esquema es independiente de los datos almacenados y es esencial para mantener la organización y la integridad de la base de datos.

En otras palabras, el esquema es la descripción de cómo se verá y funcionará la base de datos.

Instancia de la Base de Datos (Instancia de la BD): Una instancia de una base de datos se refiere al conjunto de datos almacenados en la base de datos en un momento específico. Mientras que el esquema es una descripción de cómo deben estar organizados los datos, la instancia es el contenido real de la base de datos en un momento dado.



EN UNA BD INTERVIENEN

DBMS

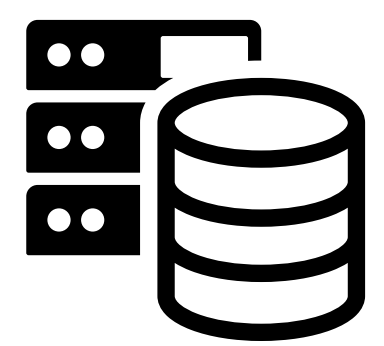


Un DBMS (Database Management System) o Sistema de Gestión de Bases de Datos es un software que permite crear, gestionar, y manipular bases de datos de manera eficiente. Facilita el almacenamiento, consulta, modificación y eliminación de datos, y asegura su integridad y seguridad. Ejemplos de DBMS incluyen MySQL, Oracle, y SQL Server.

DBA



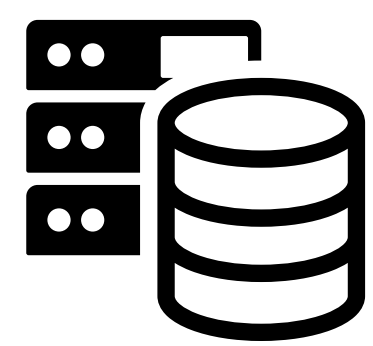
El DBA (Database Administrator) o Administrador de Bases de Datos es el profesional responsable de la gestión, mantenimiento, optimización, y seguridad de una base de datos. Sus tareas incluyen asegurar la disponibilidad, rendimiento, respaldo y recuperación de los datos, así como la implementación de políticas de acceso y seguridad.



DEFINICIONES

Usuario final: Es la persona o grupo de personas que interactúan directamente con la base de datos a través de aplicaciones o interfaces, con el objetivo de consultar, ingresar o modificar datos, sin necesidad de conocer los aspectos técnicos o internos del sistema de gestión de la base de datos.

Usuario especialista: Es un usuario con conocimientos avanzados y técnicos sobre la base de datos, que puede realizar consultas complejas, diseñar bases de datos o realizar análisis detallados. Los usuarios especialistas suelen estar familiarizados con lenguajes de consulta como SQL y tienen un conocimiento profundo de los datos y el esquema.

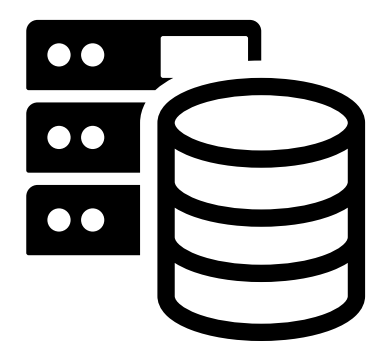


SISTEMA DE GESTIÓN DE BASE DE DATOS

Un Sistema de Gestión de Base de Datos (DBMS, por sus siglas en inglés) permite:

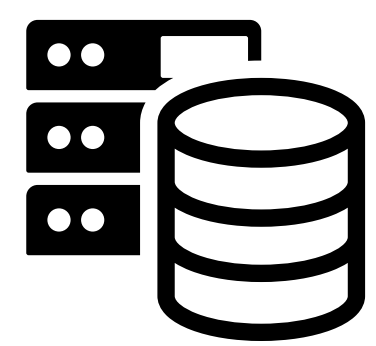
- 1. Crear y gestionar bases de datos:** Facilita la creación, actualización y mantenimiento de bases de datos para almacenar y organizar los datos de forma estructurada.
- 2. Acceso eficiente:** Permite a los usuarios consultar, modificar, insertar y eliminar datos mediante lenguajes como SQL, de manera eficiente y rápida.
- 3. Control de acceso:** Proporciona mecanismos para controlar el acceso a los datos, asegurando que solo usuarios autorizados puedan realizar determinadas operaciones.
- 4. Seguridad de los datos:** Garantiza la protección de los datos contra accesos no autorizados o ciberataques mediante medidas de seguridad como encriptación, autenticación, y auditoría.
- 5. Integridad de los datos:** Se asegura de que los datos almacenados sean precisos y consistentes a través de reglas de integridad (como las claves primarias y foráneas).
- 6. Recuperación ante fallos:** Ofrece mecanismos para recuperar datos en caso de fallos, como copias de seguridad automáticas y sistemas de recuperación ante desastres.
- 7. Gestión de la concurrencia:** Permite que múltiples usuarios accedan y modifiquen la base de datos al mismo tiempo sin generar conflictos entre las transacciones.
- 8. Optimización de consultas:** Mejora el rendimiento de las consultas sobre grandes volúmenes de datos mediante el uso de índices, planes de ejecución y técnicas de optimización de consultas.

En resumen, un DBMS facilita el manejo de grandes volúmenes de datos de manera eficiente, segura y con un control centralizado.



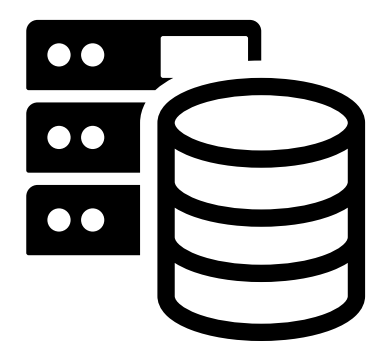
VENTAJAS DE UTILIZAR BASES DE DATOS

- Normalizar los datos de la BD
- Evitar la redundancia de datos
- Evitar la inconsistencia de datos
- Garantizar la integridad de los datos
- Garantizar la seguridad de los datos
- Compartir los datos
- Facilidad de modificar los datos



SGBD

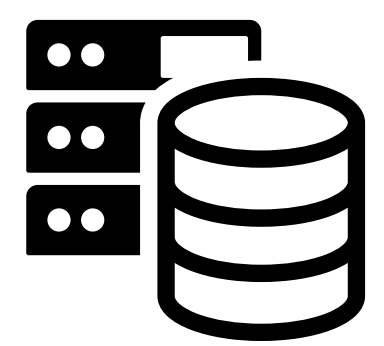
- **Gestión de transacciones:** Es el proceso mediante el cual un sistema de bases de datos asegura que una serie de operaciones (transacción) se realicen completamente o no se realicen en absoluto, garantizando propiedades como la atomicidad, consistencia, aislamiento y durabilidad (ACID).
- **Recuperación ante fallas y rollback de transacciones:** Consiste en restaurar la base de datos a un estado coherente en caso de una falla, y rollback es el mecanismo que deshace una transacción incompleta o con errores para que la base de datos vuelva a su estado anterior a dicha transacción.
- **Manejo de respaldos:** Proceso de crear copias de seguridad periódicas de la base de datos para prevenir la pérdida de datos en caso de fallas. Los respaldos permiten recuperar datos si ocurre un desastre o se corrompe la información.
- **Independencia de los datos:** Capacidad de modificar el esquema de la base de datos en diferentes niveles (físico o lógico) sin afectar los niveles superiores o inferiores, permitiendo que los programas de aplicación sigan funcionando sin cambios significativos.



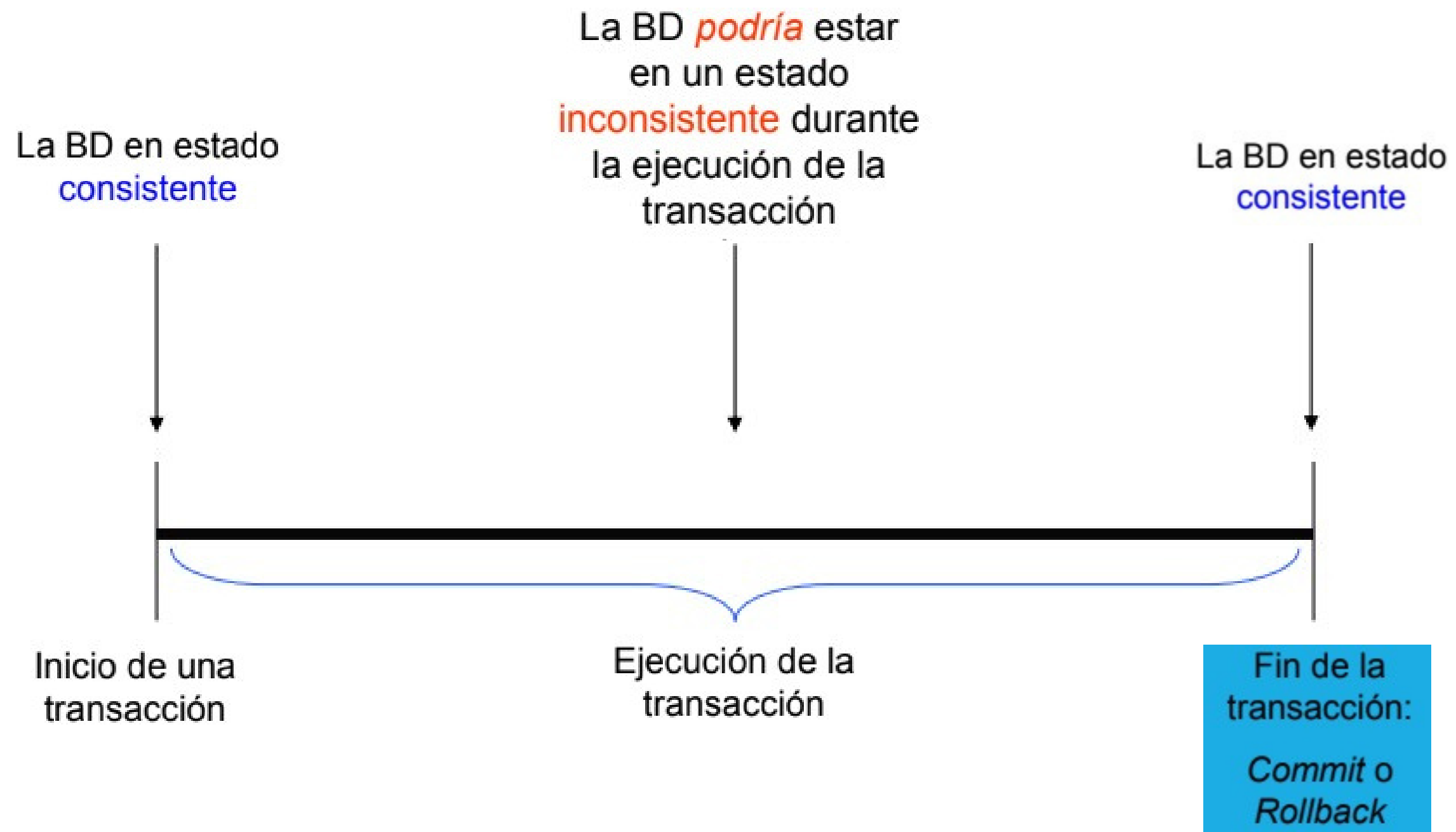
SGBD

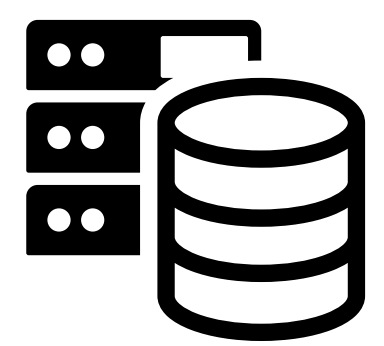
Un SGBD debe garantizar las propiedades ACID para las transacciones:

- Atomicity (Atomicidad)
 - Consistency (Consistencia)
 - Isolation (Aislamiento)
 - Durability (Durabilidad)



SGBD

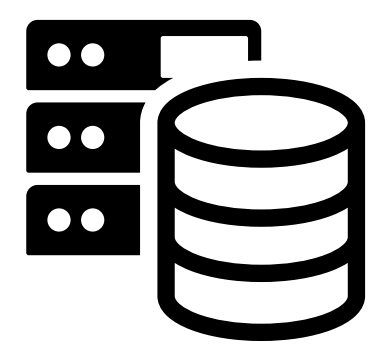




VISIÓN DE LOS DATOS

Objetivo de un SGBD (Sistema de Gestión de Bases de Datos):

El objetivo principal de un SGBD es proporcionar un entorno eficiente, confiable y seguro para el almacenamiento, manipulación y recuperación de datos. Facilita la organización de datos de manera estructurada, permitiendo a los usuarios y aplicaciones realizar operaciones complejas sobre grandes volúmenes de información sin comprometer la integridad ni la seguridad de los datos.



VISIÓN DE LOS DATOS

Niveles de un SGBD:

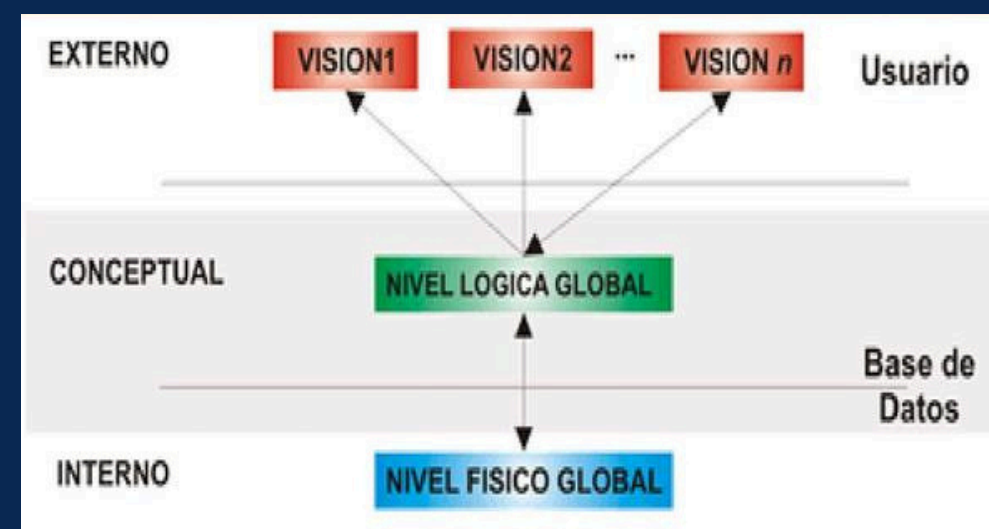
- 1. Nivel Físico:** Se refiere a la forma en que los datos se almacenan físicamente en el hardware del sistema, incluyendo la organización de archivos, registros y estructuras de almacenamiento. Este nivel es responsable de gestionar los aspectos de bajo nivel del almacenamiento de datos, como la asignación de espacio en disco y el manejo de dispositivos de almacenamiento.
- 2. Nivel Lógico:** Este nivel describe la estructura de la base de datos a un nivel más abstracto, especificando qué datos se almacenan y las relaciones entre ellos. En este nivel, se define el esquema de la base de datos, es decir, cómo están organizados los datos en tablas y cómo se relacionan entre sí, pero no se preocupa de cómo se almacenan físicamente.
- 3. Nivel de Vista (o Nivel Externo):** Se refiere a la representación de los datos para los usuarios finales o aplicaciones. En este nivel, se pueden definir diferentes vistas de la base de datos que pueden mostrar solo una parte de los datos o estructurarlos de manera que sean más comprensibles para los usuarios. Esto permite que diferentes usuarios tengan acceso a diferentes representaciones de la misma base de datos, lo que mejora la seguridad y la usabilidad.

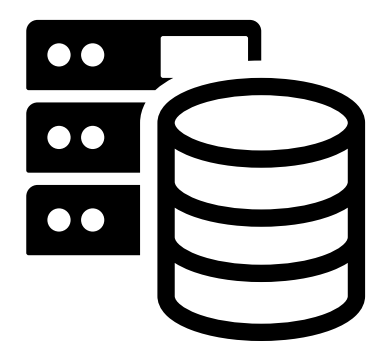


VISIÓN DE LOS DATOS

Niveles de un SGBD:

- 1. Nivel Físico:** Se refiere a la forma en que los datos se almacenan físicamente en el hardware del sistema, incluyendo la organización de archivos, registros y estructuras de almacenamiento. Este nivel es responsable de gestionar los aspectos de bajo nivel del almacenamiento de datos, como la asignación de espacio en disco y el manejo de dispositivos de almacenamiento.
- 2. Nivel Lógico:** Este nivel describe la estructura de la base de datos a un nivel más abstracto, especificando qué datos se almacenan y las relaciones entre ellos. En este nivel, se define el esquema de la base de datos, es decir, cómo están organizados los datos en tablas y cómo se relacionan entre sí, pero no se preocupa de cómo se almacenan físicamente.
- 3. Nivel de Vista (o Nivel Externo):** Se refiere a la representación de los datos para los usuarios finales o aplicaciones. En este nivel, se pueden definir diferentes vistas de la base de datos que pueden mostrar solo una parte de los datos o estructurarlos de manera que sean más comprensibles para los usuarios. Esto permite que diferentes usuarios tengan acceso a diferentes representaciones de la misma base de datos, lo que mejora la seguridad y la usabilidad.





TIPOS DE BASES DE DATOS

Bases de Datos Relacionales (RDBMS

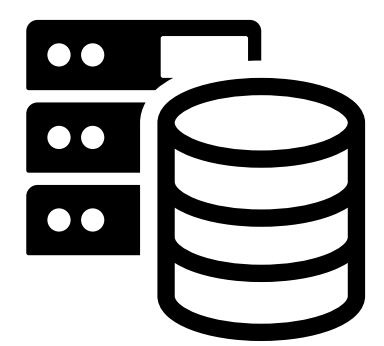
Bases de Datos NoRelacionales(NoSQL)

Bases de Datos en la Nube

Bases de Datos Espaciales

Bases de Datos Temporales

**Bases de Datos en Memoria (In-Memory
Databases)**



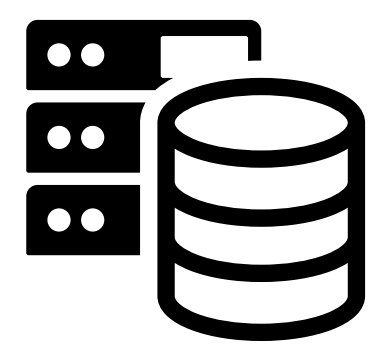
TIPOS DE BASES DE DATOS

Bases de Datos Relacionales (RDBMS)

Concepto: Son bases de datos que almacenan datos en tablas con filas y columnas, donde las relaciones entre los datos se definen a través de claves primarias y foráneas. Utilizan SQL (Structured Query Language) para realizar operaciones sobre los datos.

Características Clave:

- Estructura tabular (filas y columnas).
- Integridad referencial.
- Uso de SQL para consultas y manipulación.
- Soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).



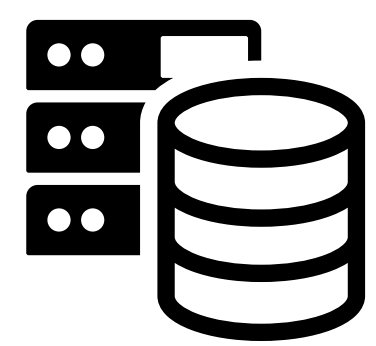
TIPOS DE BASES DE DATOS

Bases de Datos No Relacionales (NoSQL)

Concepto: Son bases de datos que no utilizan el modelo tabular tradicional. Pueden almacenar datos en formatos de documento, clave-valor, columnares o de grafos, y están diseñadas para manejar grandes volúmenes de datos distribuidos.

Características Clave:

- Escalabilidad horizontal.
- Flexibilidad en el esquema de datos.
- Ideal para datos semi-estructurados o no estructurados.
- A menudo se centran en rendimiento y disponibilidad.



TIPOS DE BASES DE DATOS

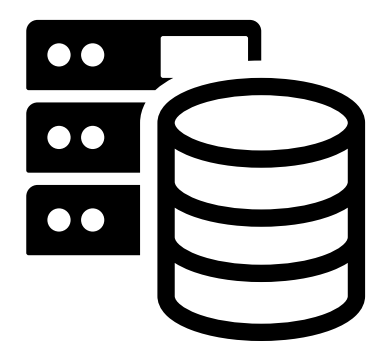
Bases de Datos en la Nube

Concepto:

Son bases de datos que se alojan en servidores en la nube, proporcionando acceso a los datos a través de internet. Los proveedores de servicios en la nube gestionan la infraestructura y el mantenimiento de la base de datos.

Características Clave:

- Acceso remoto y disponibilidad en cualquier lugar.
- Escalabilidad según la demanda.
- Reducción de costos operativos al eliminar la necesidad de hardware propio.
- Seguridad y respaldo gestionados por el proveedor.



TIPOS DE BASES DE DATOS

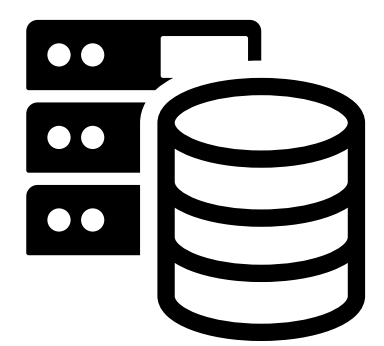
Bases de Datos Espaciales

Concepto:

Son bases de datos diseñadas para almacenar, consultar y analizar datos geoespaciales o de ubicación, permitiendo a los usuarios realizar análisis complejos de datos espaciales.

Características Clave:

- Almacenamiento de datos geográficos (puntos, líneas, polígonos).
- Soporte para consultas espaciales y funciones geométricas.
- Ideal para aplicaciones de sistemas de información geográfica (SIG).
- Análisis de datos de ubicación y mapeo.



TIPOS DE BASES DE DATOS

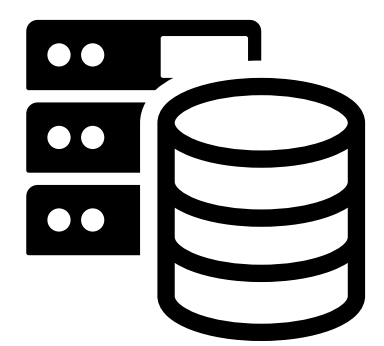
Bases de Datos Temporales

Concepto:

Son bases de datos que gestionan datos que cambian con el tiempo. Permiten almacenar y consultar datos en diferentes puntos temporales, lo que es útil para aplicaciones que requieren un seguimiento histórico.

Características Clave:

- Capacidad para realizar consultas a lo largo del tiempo.
- Almacenamiento de datos en versiones o instantáneas.
- Soporte para operaciones de tiempo y validez temporal.
- Útil para auditoría y seguimiento de cambios.



TIPOS DE BASES DE DATOS

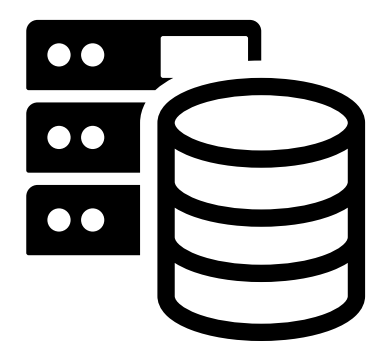
Bases de Datos en Memoria (In-Memory Databases)

Concepto:

Son bases de datos que almacenan datos en la memoria principal (RAM) en lugar de en disco, lo que permite un acceso extremadamente rápido a los datos.

Características Clave:

- Alta velocidad de acceso y procesamiento.
- Ideal para aplicaciones que requieren baja latencia.
- Menor capacidad de almacenamiento en comparación con bases de datos en disco.
- A menudo se utilizan en aplicaciones analíticas y de tiempo real.

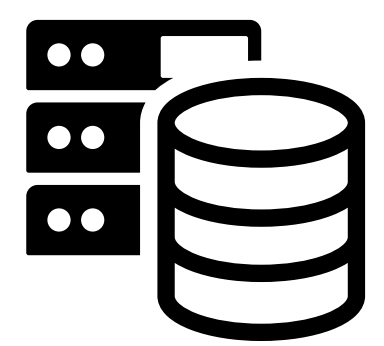


MODELOS DE DATOS

Los modelos de datos son estructuras conceptuales que definen cómo se organizan, almacenan y manipulan los datos en una base de datos. Proporcionan un marco para representar la realidad de manera abstracta, describiendo las entidades, sus atributos y las relaciones entre ellas. Los modelos de datos son fundamentales para el diseño de bases de datos, ya que determinan cómo los datos se relacionan y cómo pueden ser accedidos y manipulados.

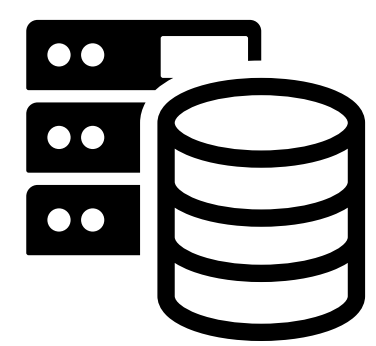
Se describirán dos modelos de datos:

- Modelo entidad-relación
- Modelo relacional



MODELO ENTIDAD-RELACIÓN

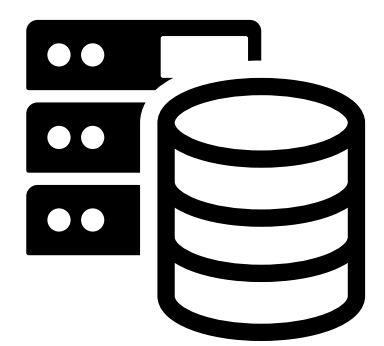
El Modelo Entidad-Relación (ER) es una técnica de modelado de datos que se utiliza para representar la estructura lógica de una base de datos. Fue introducido por Peter Chen en 1976 y se utiliza ampliamente en el diseño de bases de datos relacionales. Este modelo permite a los diseñadores describir las entidades que se almacenarán en la base de datos, sus atributos y las relaciones entre ellas.



MODELO ENTIDAD-RELACIÓN

Componentes Clave del Modelo ER:

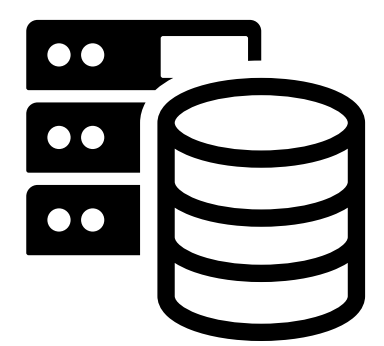
- 1. Entidades:** Representan objetos o conceptos del mundo real que tienen existencia independiente. Por ejemplo, "Cliente", "Producto" o "Orden".
- 2. Atributos:** Son las características o propiedades de las entidades. Por ejemplo, un "Cliente" podría tener atributos como "Nombre", "Dirección" y "Teléfono".
- 3. Relaciones:** Describen cómo las entidades están interconectadas. Por ejemplo, un "Cliente" puede realizar múltiples "Órdenes", estableciendo una relación entre estas dos entidades.
- 4. Cardinalidad:** Indica el número de instancias de una entidad que pueden estar asociadas con instancias de otra entidad (por ejemplo, uno a uno, uno a muchos, muchos a muchos).
- 5. Diagrama ER:** Se utiliza para representar visualmente el modelo ER, mostrando entidades, atributos y relaciones en un formato gráfico.



MODELO ENTIDAD-RELACIÓN

Los atributos en el Modelo Entidad-Relación (ER) se pueden clasificar en varios tipos, cada uno con características específicas. Aquí están los tipos más comunes de atributos:

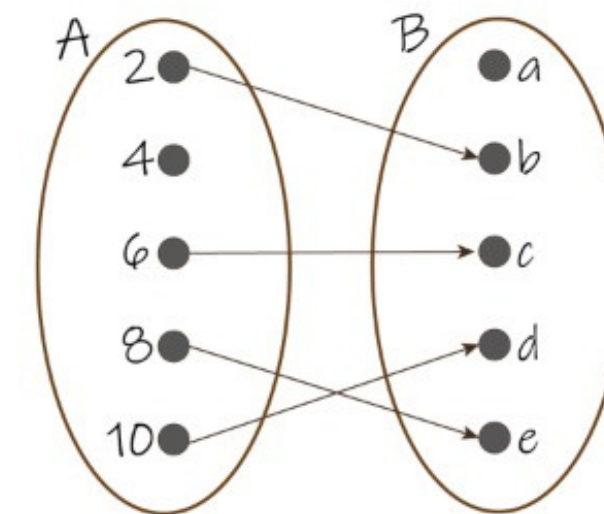
- 1. Atributos Simples:** Son aquellos que no se pueden dividir en subcomponentes. Por ejemplo, el atributo "Nombre" es simple, ya que se refiere a una sola característica.
- 2. Atributos Compuestos:** Son atributos que se pueden descomponer en atributos más simples. Por ejemplo, un atributo "Dirección" podría descomponerse en "Calle", "Ciudad" y "Código Postal".
- 3. Atributos Derivados:** Son aquellos que no se almacenan directamente en la base de datos, sino que se calculan a partir de otros atributos. Por ejemplo, la "Edad" de una persona puede derivarse de su "Fecha de Nacimiento".
- 4. Atributos Clave:** Son atributos que se utilizan para identificar de manera única a una entidad dentro de un conjunto de datos. Por ejemplo, un "ID de Cliente" que es único para cada cliente en una base de datos.
- 5. Atributos Multivaluados:** Son atributos que pueden tener más de un valor para una sola instancia de entidad. Por ejemplo, un atributo "Número de Teléfono" que puede contener múltiples números de contacto para un cliente.
- 6. Atributos de Valor Nulo:** Son atributos que pueden no tener un valor asignado. Esto es útil cuando la información no está disponible o no es aplicable a una entidad en particular.



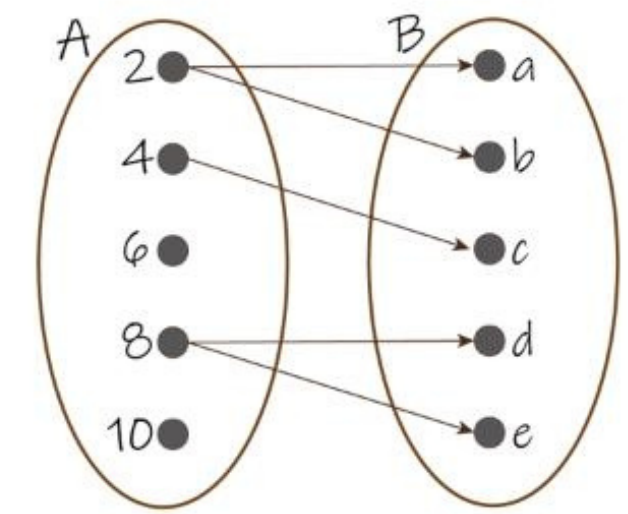
MODELO ENTIDAD-RELACIÓN

La cardinalidad en un modelo entidad-relación (ER) describe la cantidad de instancias de una entidad que pueden estar asociadas con una instancia de otra entidad. Existen varias correspondencias de cardinalidad, que son fundamentales para definir las relaciones entre entidades. A continuación, se describen las principales:

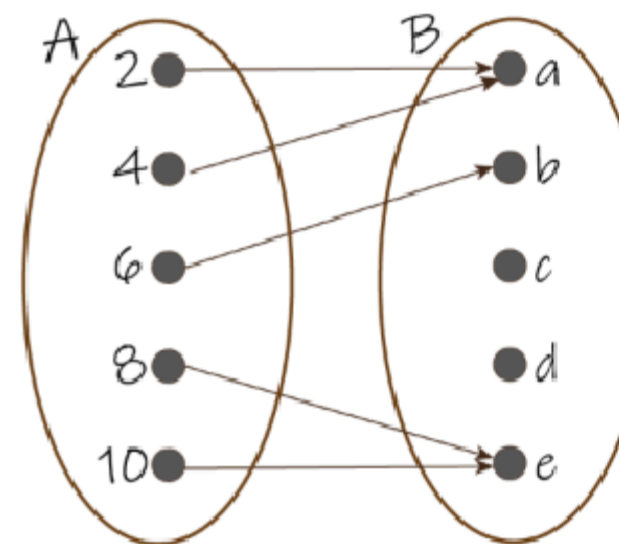
1. **Uno a Uno (1:1):** En una relación uno a uno, una instancia de la entidad A se relaciona con una sola instancia de la entidad B, y viceversa. Por ejemplo, cada persona tiene un único número de identificación.
2. **Uno a Muchos (1:N):** En esta relación, una instancia de la entidad A puede estar relacionada con muchas instancias de la entidad B, pero cada instancia de B solo puede estar relacionada con una instancia de A. Por ejemplo, un profesor puede enseñar a muchos estudiantes, pero cada estudiante tiene un solo profesor.
3. **Muchos a Uno (N:1):** Esta relación es la inversa de la uno a muchos. Muchas instancias de la entidad A pueden estar asociadas a una sola instancia de la entidad B. Por ejemplo, muchos empleados pueden trabajar en una sola empresa.
4. **Muchos a Muchos (M):** En esta relación, muchas instancias de la entidad A pueden estar asociadas con muchas instancias de la entidad B. Por ejemplo, los estudiantes pueden inscribirse en varios cursos, y cada curso puede tener varios estudiantes.



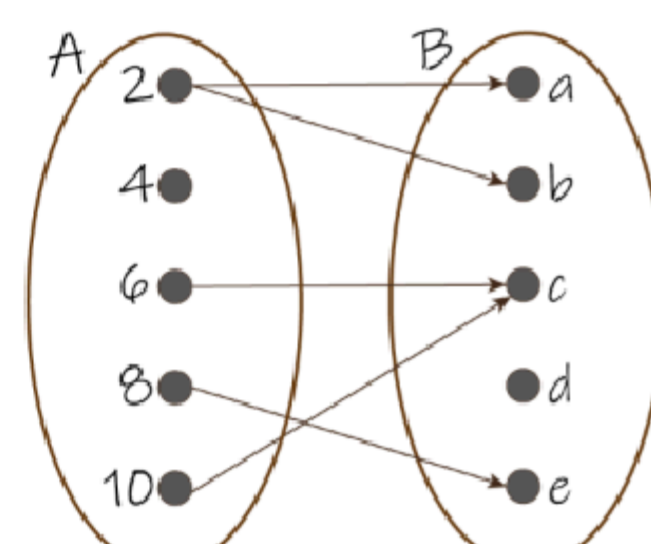
Uno a Uno



Uno a Muchos



Muchos a Uno



Muchos a Muchos

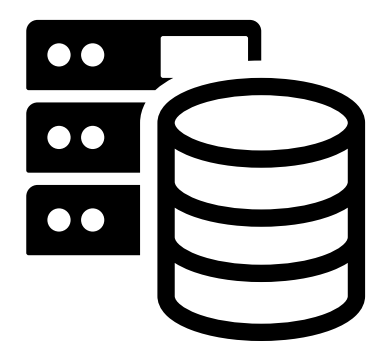


DIAGRAMA ENTIDAD-RELACIÓN

Un Diagrama Entidad-Relación (ER) es una representación gráfica que ilustra las entidades dentro de un sistema y las relaciones entre ellas. Este tipo de diagrama es fundamental en el diseño de bases de datos, ya que permite visualizar cómo interactúan los diferentes componentes de la base de datos.

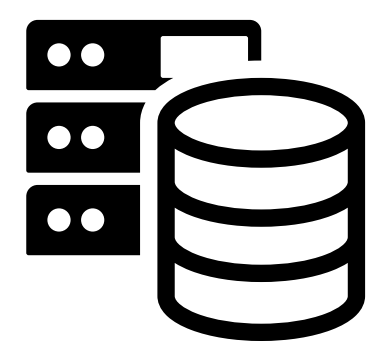


DIAGRAMA ENTIDAD-RELACIÓN

Componentes del Diagrama Entidad-Relación:

1. Entidades: Representadas por rectángulos, las entidades son objetos o conceptos del mundo real que tienen existencia independiente. Por ejemplo, "Cliente", "Producto", "Pedido".
2. Atributos: Representan las características de una entidad y se representan como elipses conectadas a su entidad correspondiente. Por ejemplo, un "Cliente" puede tener atributos como "Nombre", "Dirección" y "Teléfono".
3. Relaciones: Representadas por rombos, las relaciones indican cómo se conectan las entidades. Por ejemplo, una relación entre "Cliente" y "Pedido" podría ser "Realiza", que indica que un cliente puede realizar uno o más pedidos.
4. Cardinalidad: Indica el número de instancias de una entidad que pueden estar asociadas con las instancias de otra entidad. Se muestra mediante notaciones al lado de las relaciones, como 1:1, 1:N, N:1, o M.
5. Llave Primaria: Un atributo (o conjunto de atributos) que identifica de manera única a cada instancia de una entidad. Se suele subrayar en los diagramas.

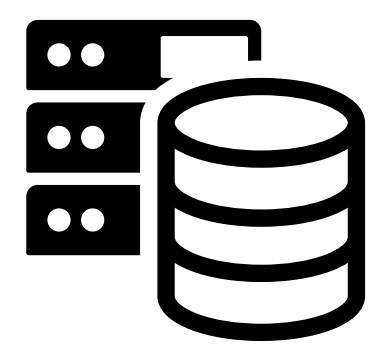


DIAGRAMA ENTIDAD-RELACIÓN

- Rectángulos, para representar las entidades.
- Elipses, para los atributos.
- Diamantes o líneas de conexión para representar las relaciones entre entidades y atributos.
- Elipses dobles, que representan atributos multivalorados.
- Elipses discontinuas, que denotan atributos derivados.
- Líneas dobles, que indican participación total de una entidad en un conjunto de relaciones.
- Rectángulos dobles, que representan conjuntos de entidades débiles.

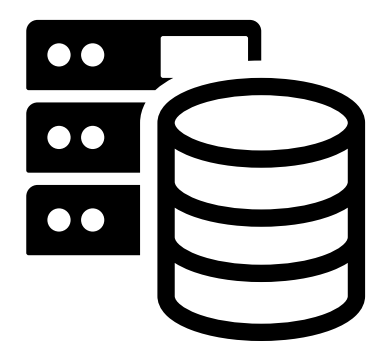


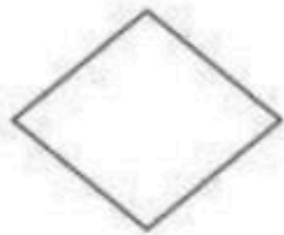
DIAGRAMA ENTIDAD-RELACIÓN



Entidad



Entidad débil



Relación



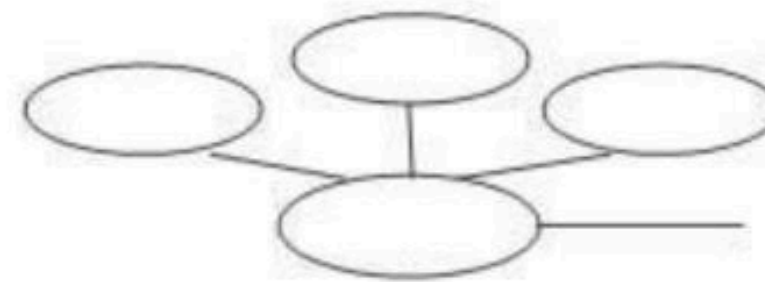
Atributo



Atributo Llave



Atributo Multivaluado



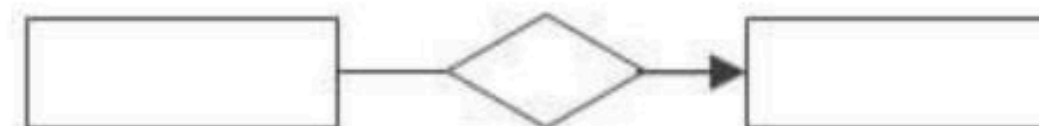
Atributo Compuesto



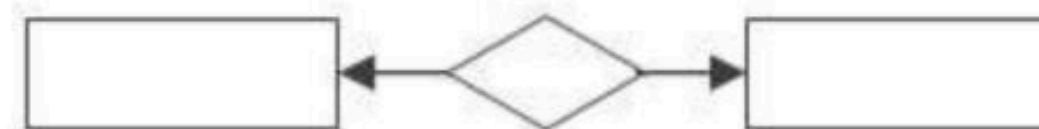
Atributo Calculado



Relación n: m

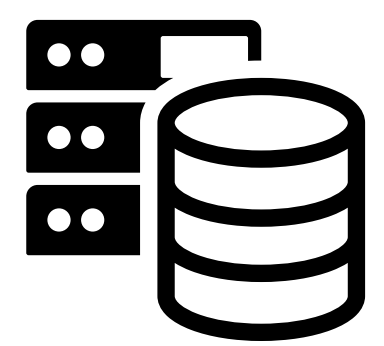


Relación n: 1



Relación 1:1

Platzi



MODELO ENTIDAD-RELACIÓN

El modelo entidad relación pretende ser un reflejo de la estructura gramatical y, por ello, utiliza:

- Sustantivos, comunes o propios, para definir tipos de entidades y entidades.
- Verbos, para definir tipos de relación.
- Adjetivos, como atributos de una entidad.
- Adverbios, como atributos de una relación.

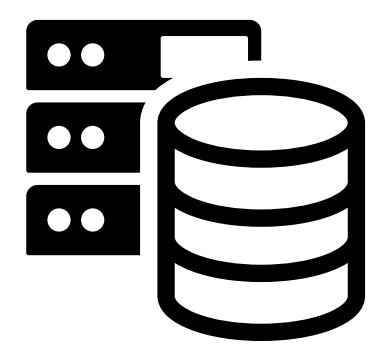
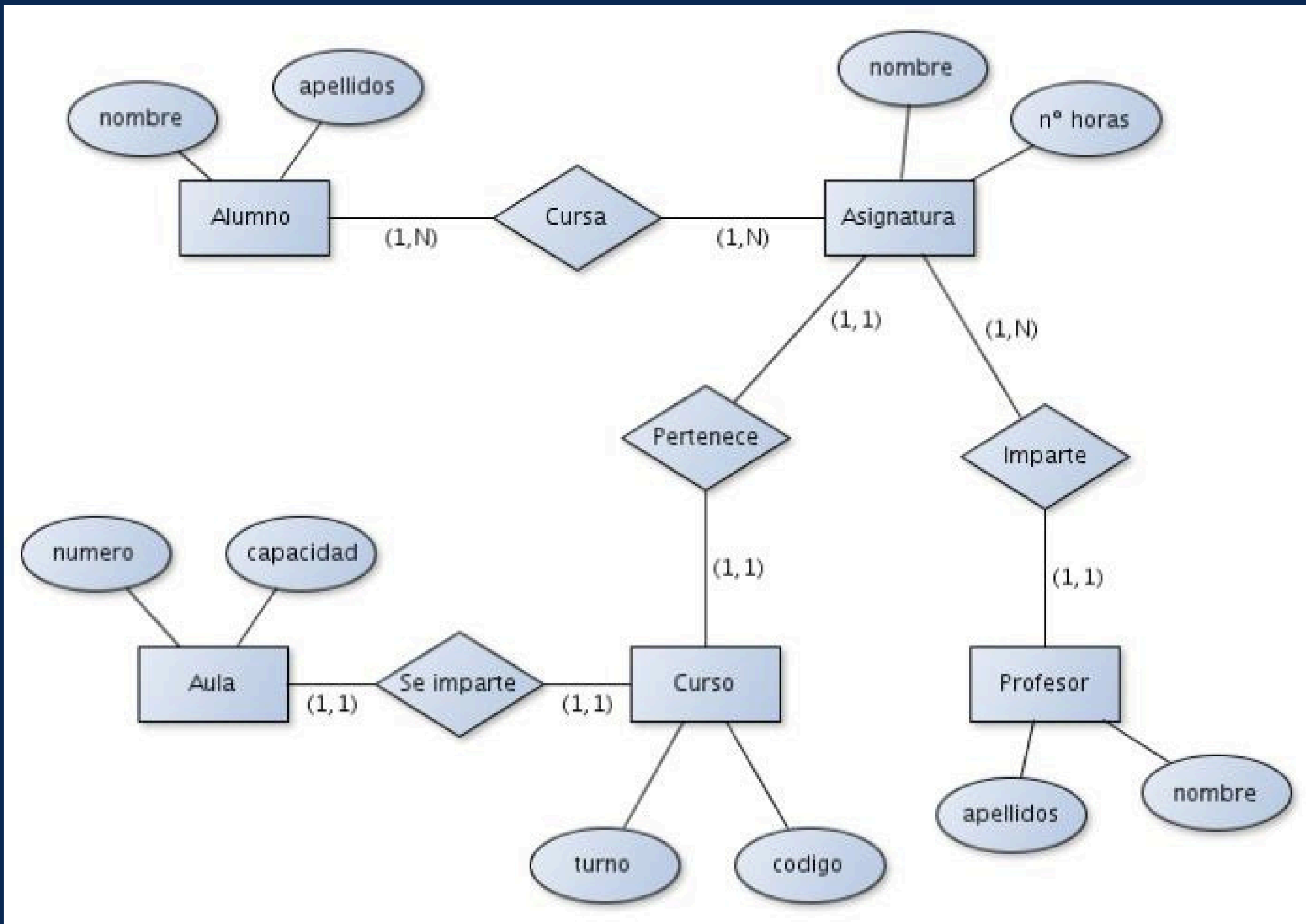
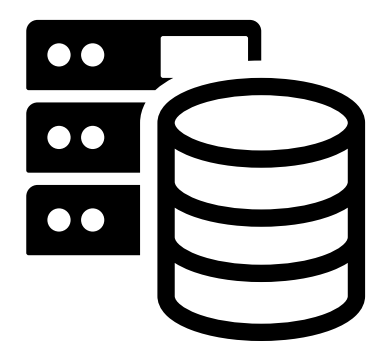


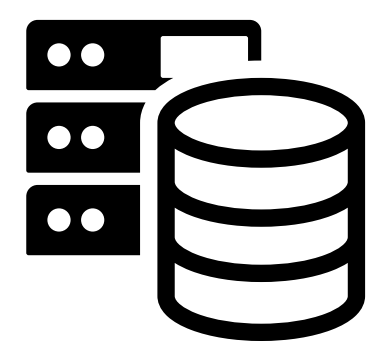
DIAGRAMA ENTIDAD-RELACIÓN



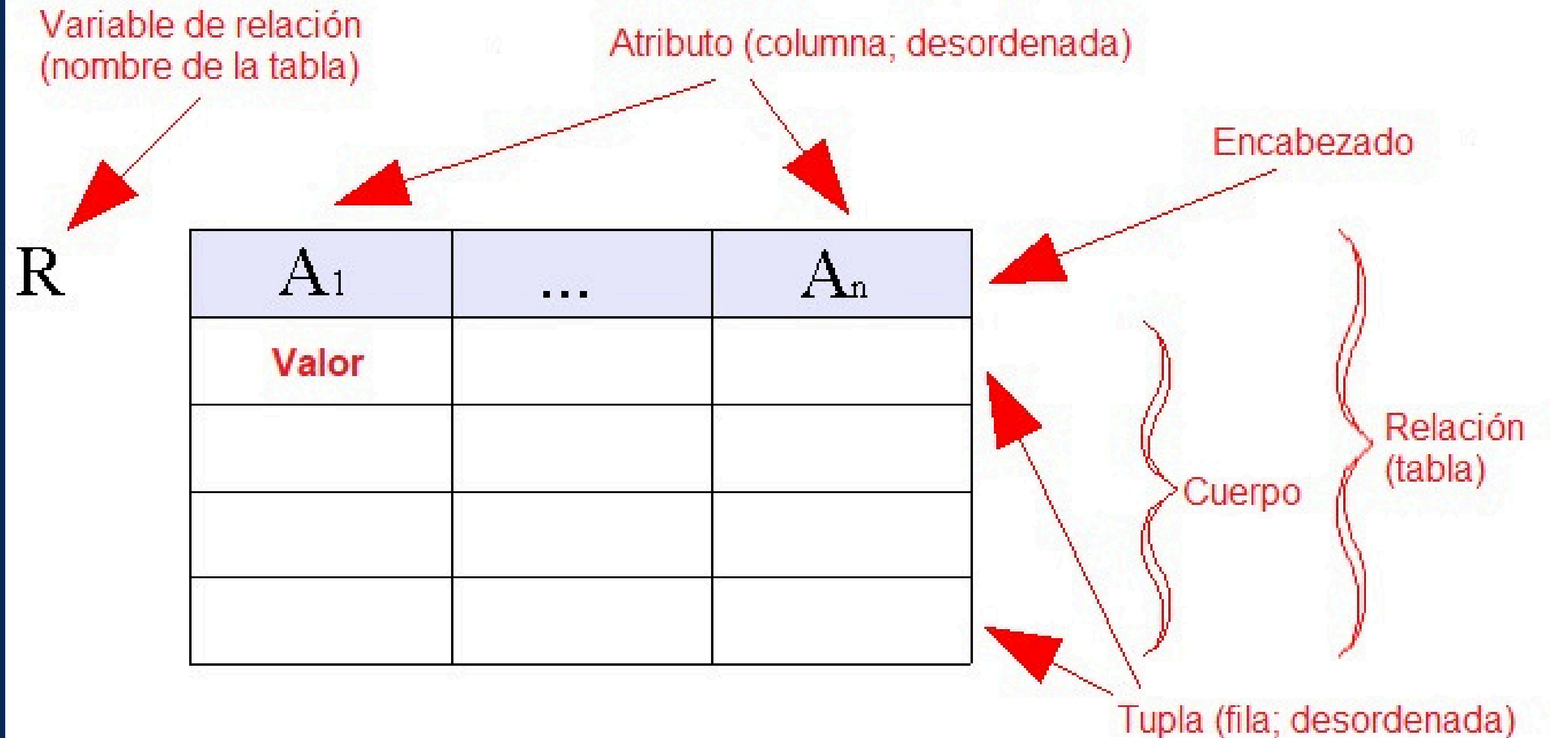


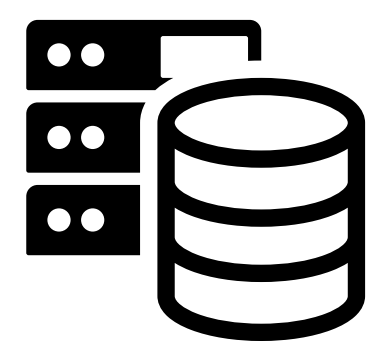
MODELO RELACIONAL

El Modelo Relacional es un enfoque de diseño de bases de datos que organiza los datos en tablas (también llamadas relaciones) que se pueden interrelacionar entre sí. Este modelo fue propuesto por Edgar F. Codd en 1970 y se ha convertido en uno de los modelos más utilizados para la gestión de bases de datos debido a su simplicidad y flexibilidad.



MODELO RELACIONAL

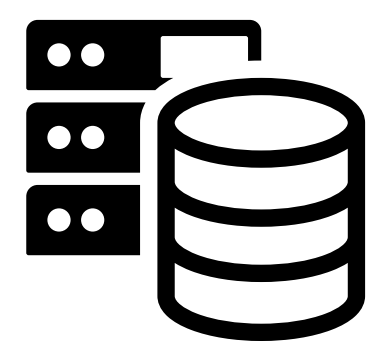




MODELO RELACIONAL

Conceptos Clave del Modelo Relacional:

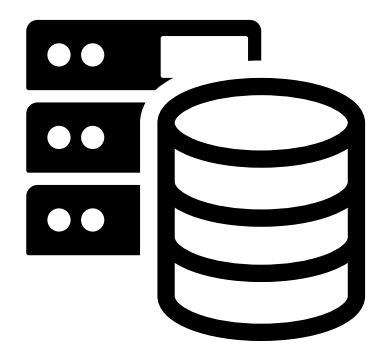
- 1.Tabla (Relación):** La tabla es la estructura principal del modelo relacional, compuesta por filas y columnas. Cada tabla representa una entidad o un conjunto de datos relacionados. Por ejemplo, una tabla "Clientes" podría contener información sobre todos los clientes de una empresa.
- 2.Filas (Tuplas):** Cada fila de una tabla representa un registro único. Por ejemplo, en la tabla "Clientes", cada fila podría contener la información de un cliente específico.
- 3.Columnas (Atributos):** Las columnas de una tabla representan los diferentes atributos o características de la entidad que la tabla describe. Por ejemplo, en la tabla "Clientes", las columnas podrían incluir "ID_Cliente", "Nombre", "Dirección" y "Teléfono".
- 4.Clave Primaria:** Una columna (o conjunto de columnas) que identifica de manera única cada fila en una tabla. Por ejemplo, "ID_Cliente" podría ser la llave primaria en la tabla "Clientes".
- 5.Clave Foránea:** Una columna en una tabla que establece una relación con la llave primaria de otra tabla. Esto permite conectar datos en diferentes tablas. Por ejemplo, en una tabla "Pedidos", podría haber una columna "ID_Cliente" que se relaciona con la llave primaria "ID_Cliente" de la tabla "Clientes".
- 6.Relaciones:** Las tablas pueden estar relacionadas entre sí mediante llaves foráneas, lo que permite realizar consultas complejas que involucren múltiples tablas. Estas relaciones pueden ser de uno a uno, de uno a muchos, o de muchos a muchos.



MODELO RELACIONAL

Ventajas del Modelo Relacional:

- **Simplicidad:** La estructura tabular es intuitiva y fácil de entender.
- **Flexibilidad:** Permite realizar consultas complejas utilizando un lenguaje de consulta estructurado (SQL).
- **Integridad de Datos:** Las restricciones, como las llaves primarias y foráneas, ayudan a mantener la consistencia y la integridad de los datos.
- **Independencia de Datos:** Los cambios en la estructura de la base de datos pueden hacerse sin afectar las aplicaciones que la utilizan.



MODELO RELACIONAL

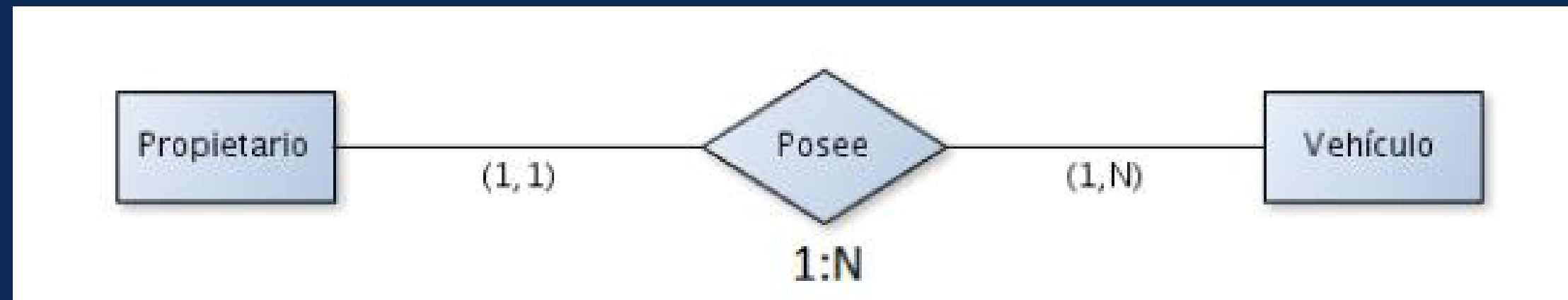
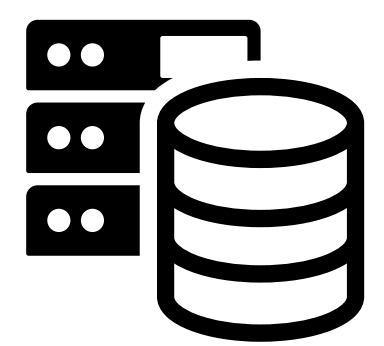


Tabla Propietario

id (PK)	nombre	apellidos	...
13	Fernando	Valdeón	
15	Ana	López	
27	Javier	Alonso	

Tabla Vehículo

id (PK)	matricula	marca	modelo	id_propietario (FK)
2	5422AWB	Seat	León	13
3	7543KGT	Ford	Focus	15
5	4764AAB	Citroen	C3	13
7	7653LTY	Renault	Megane	27



EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. E/R

1. Obtén el diagrama E/R con las tres entidades siguientes:

ALUMNO (Núm_Matrícula, Nombre, FechaNacimiento, Teléfono)

ASIGNATURA (Código_asignatura, Nombre)

PROFESOR (Id_P, NIF_P, Nombre, Especialidad, Teléfono)

Teniendo en cuenta:

Un alumno puede estar matriculado de una o varias asignaturas.

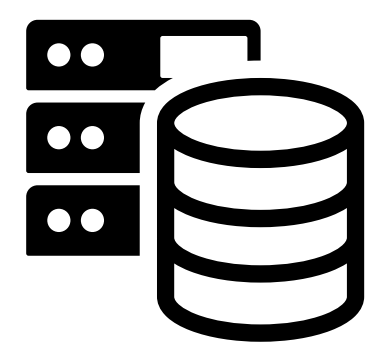
Además puede estar matriculado en la misma asignatura más de un curso escolar (si repite).

Se quiere saber el curso escolar en el que cada alumno está matriculado de cada asignatura.

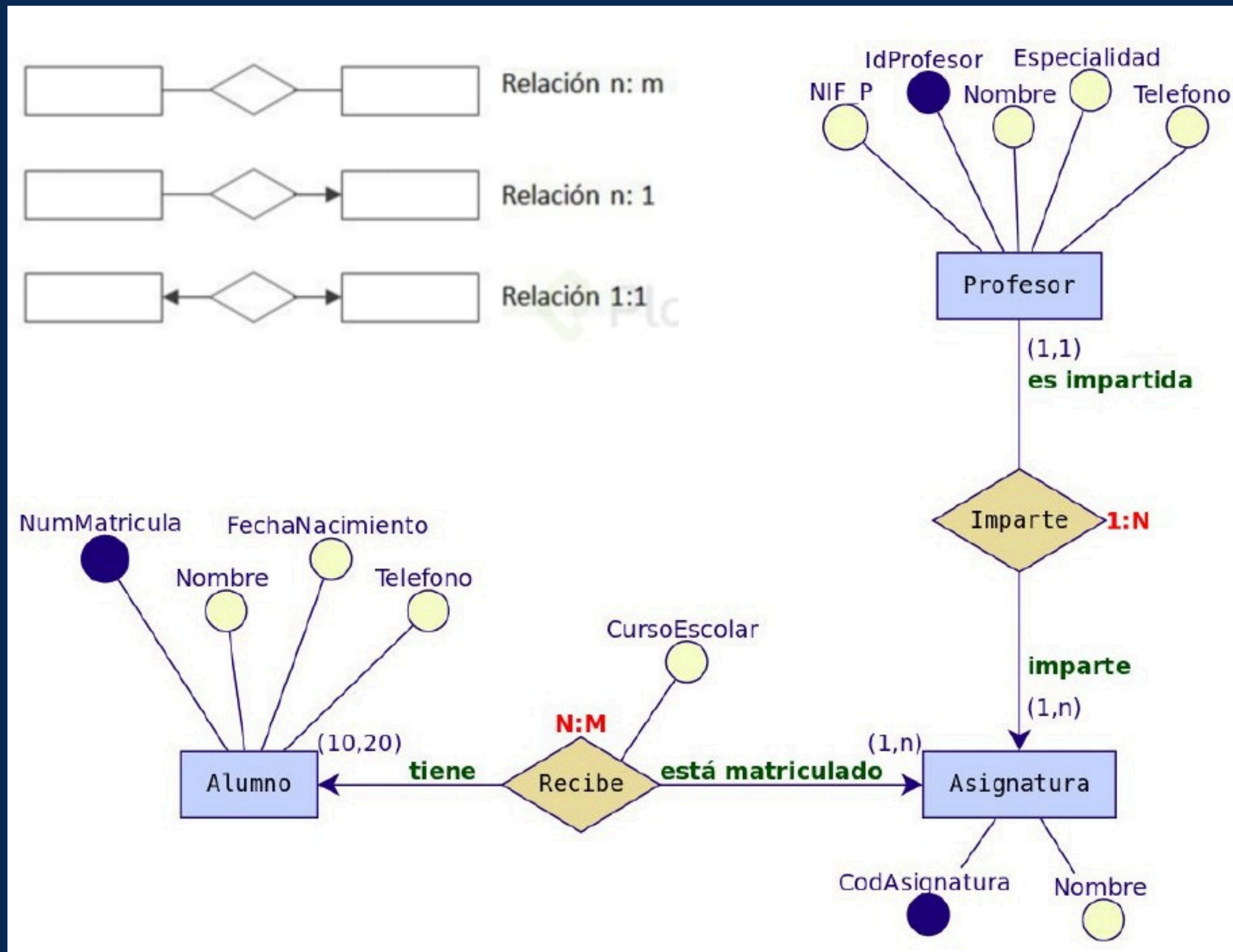
En una asignatura habrá como mínimo 10 y como máximo 25 alumnos.

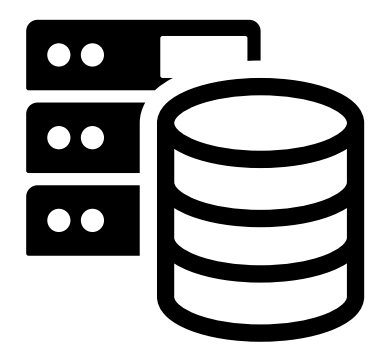
Una asignatura es impartida por un único profesor.

Un profesor podrá impartir varias asignaturas



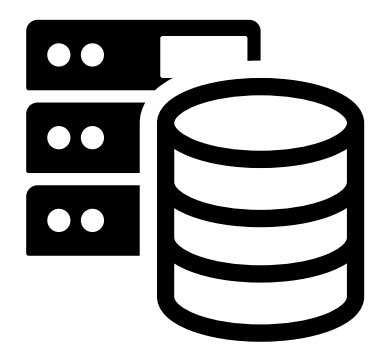
EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. E/R



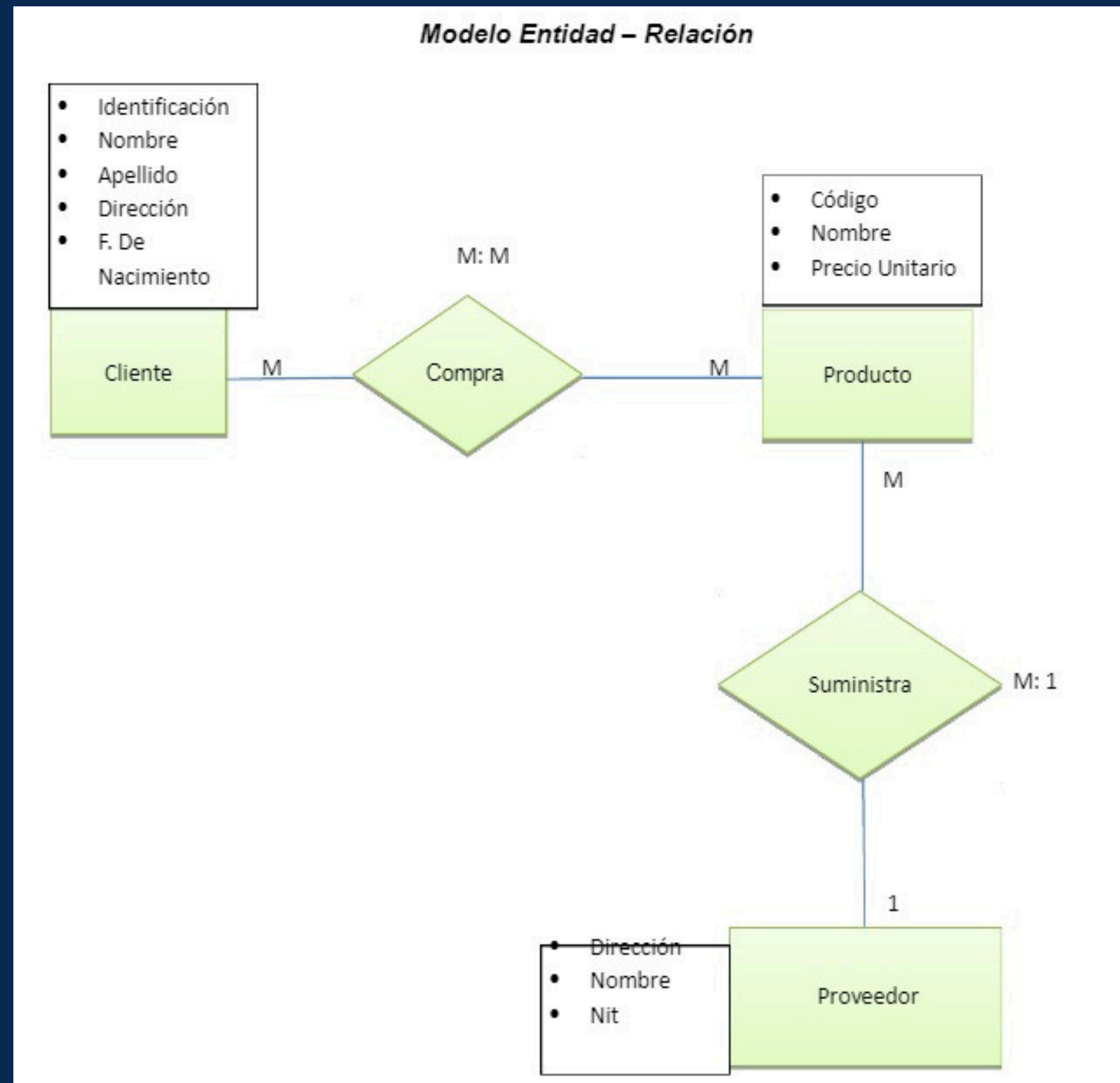


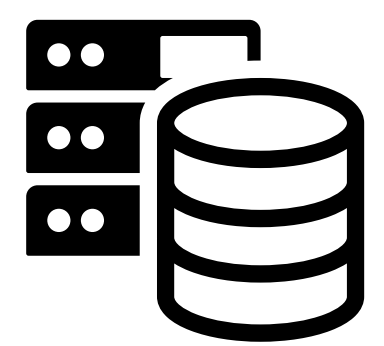
EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R

Una empresa vende productos a sus clientes, se necesita conocer los datos personales de los clientes (Nombre, apellido, identificación, dirección y fecha de nacimiento) cada producto tiene un nombre y un código, así como un precio unitario. Un cliente puede comprar varios productos a la empresa y un mismo producto puede ser comprado por varios clientes, los productos son suministrados por diferentes proveedores. Se debe tener en cuenta que un producto solo puede ser suministrado por un proveedor y que un proveedor puede suministrar diferentes productos, de cada proveedor se desea conocer el NIT, nombre y dirección.



EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R





EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R

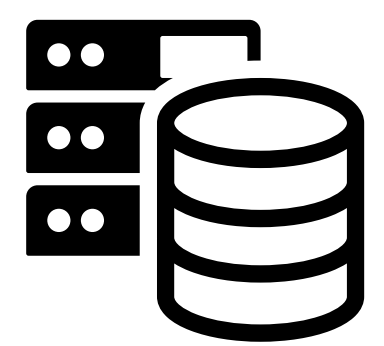
Modelo relacional

Cliente (Id-Cliente, nombre, apellido, definición, fecha de nacimiento).

Producto (código, nombre, precio unitario, Nit).

Proveedor (Nit, nombre, dirección).

Compra (Id-cliente, código).



EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R

Tabla: cliente

Id-cliente	Nombre	Apellido	Dirección	F de nacimiento

Tabla: producto

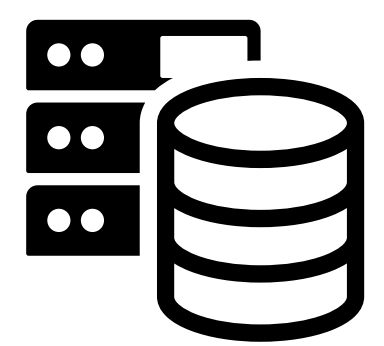
código	Nombre	Precio unitario	Nit

Tabla: proveedor

Nit	Nombre	Dirección

Tabla: compra

Id-cliente	código

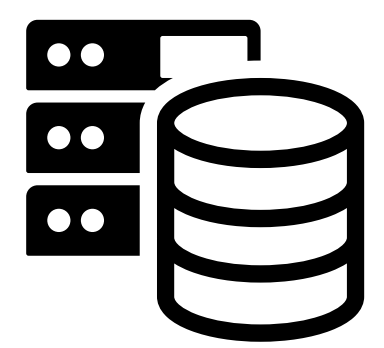


EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R

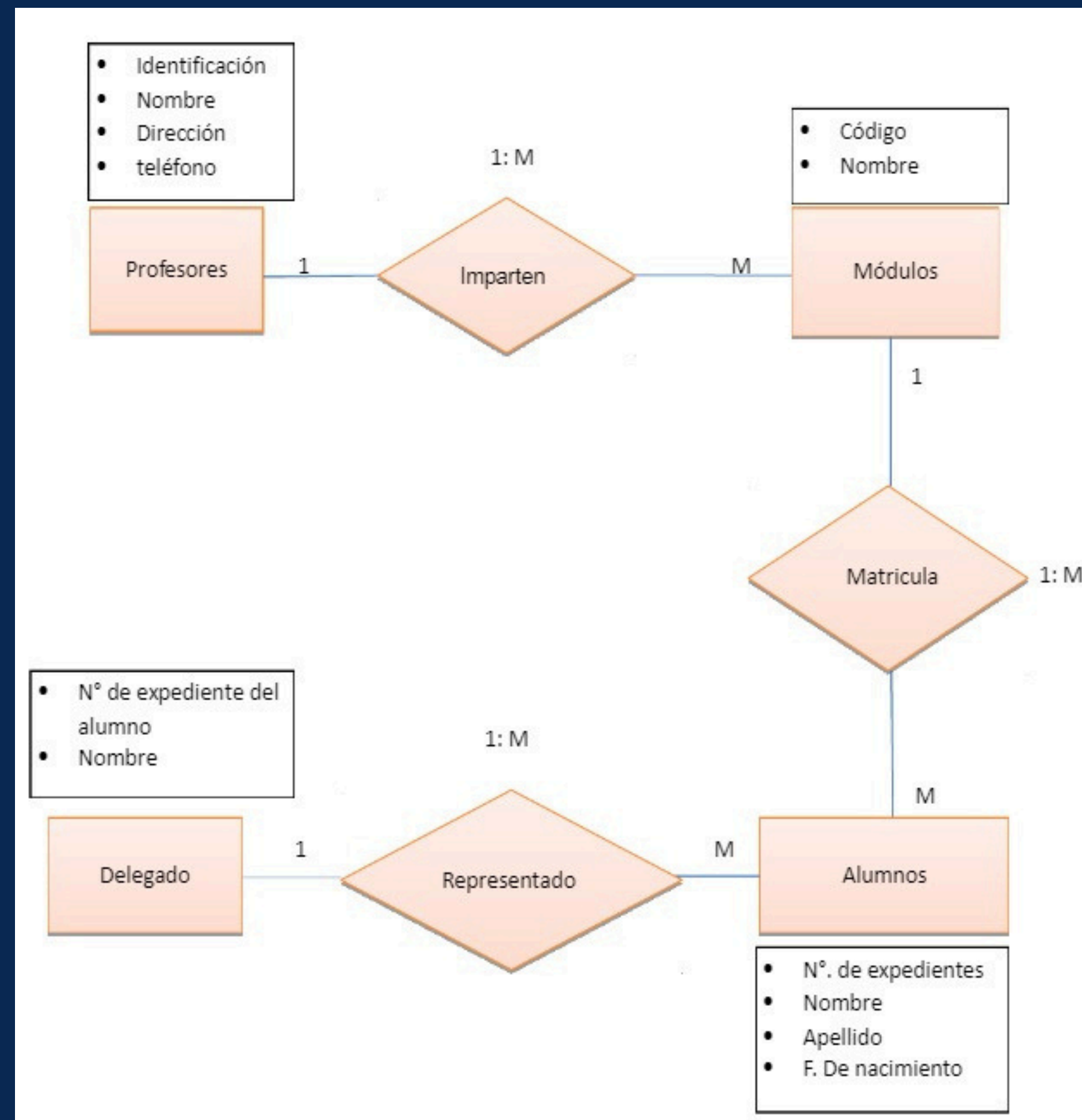
Se desea diseñar la base de datos de un instituto. En la base de datos se desea guardar los datos de los profesores del instituto (DNI, nombre, dirección y teléfono). Los profesores imparten módulos, y cada módulo tiene un código y un nombre.

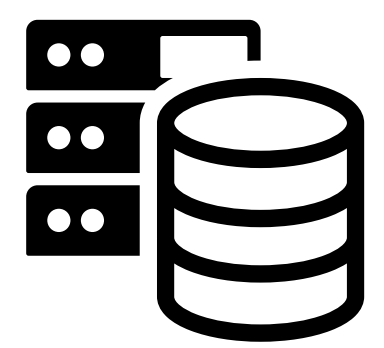
Cada alumno está matriculado en uno o varios módulos. De cada alumno se desea guardar el N° de expedientes, nombre, apellido y fecha de nacimiento. Los profesores pueden impartir varios módulos, pero un módulo solo puede ser impartido por un profesor.

Cada curso tiene un grupo de alumnos, uno de los cuales es el delegado del grupo.



EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R





EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R

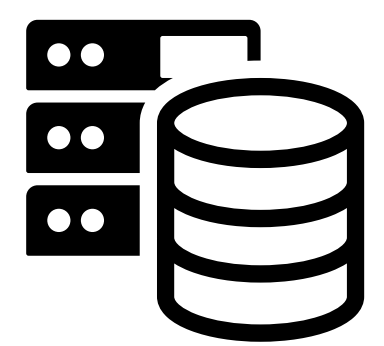
Modelo relacional

Profesor (identificación, nombre, dirección, teléfono)

Modulo (código, nombre, identificación de profesor)

Alumno (N° de expediente, nombre, apellido, fecha de nacimiento, código de modulo)

Delegado (N° de expediente, Nombre)



EJERCICIOS (RESUELTOS) DE BASES DE DATOS - MOD. R

Tabla: profesor

Identificación	Nombre	dirección	Teléfono

Tabla: modulo

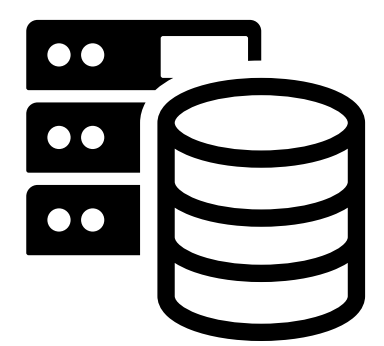
código	nombre	Id-profesor

Tabla: alumno

Nº de expediente	nombre	apellido	Fecha de nacimiento	Código de modulo

Tabla: delegado

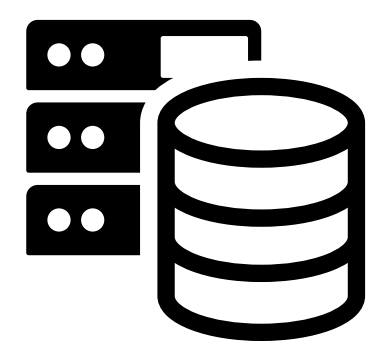
Nº de expediente de alum.	Nombre



BD EN INGENIERÍA INDUSTRIAL

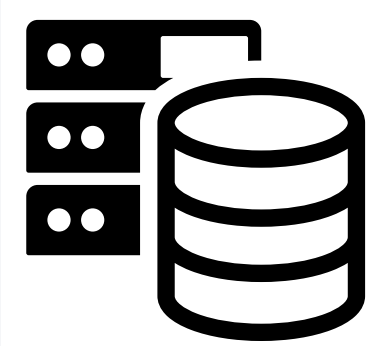
Las bases de datos desempeñan un papel crucial en el campo de la Ingeniería Industrial, ya que proporcionan una infraestructura fundamental para la gestión y el análisis de datos en una variedad de contextos industriales. A continuación, se destacan algunas de las razones clave por las que las bases de datos son esenciales en este campo:

1. Almacenamiento de Información Crítica
2. Gestión de Procesos de Producción
3. Control de Calidad
4. Optimización de Recursos
5. Análisis y Toma de Decisiones
6. Mantenimiento Preventivo
7. Cumplimiento de Normativas



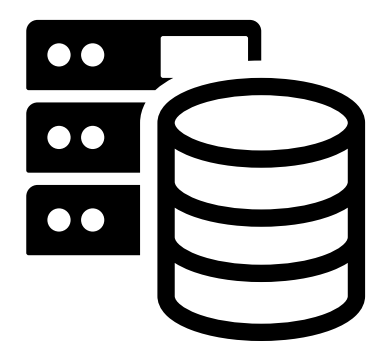
BD EN INGENIERÍA INDUSTRIAL

1. **Almacenamiento de Información Crítica:** La Ingeniería Industrial implica la recopilación de grandes cantidades de datos, desde datos de producción hasta datos de inventario. Las bases de datos permiten almacenar y organizar esta información de manera eficiente, lo que facilita la toma de decisiones basadas en datos precisos.
2. **Gestión de Procesos de Producción:** Las bases de datos son fundamentales para la gestión de procesos de producción, seguimiento de la cadena de suministro y programación de la producción. Ayudan a optimizar la eficiencia de las operaciones y a minimizar costos.
3. **Control de Calidad:** En la Ingeniería Industrial, el control de calidad es esencial. Las bases de datos permiten el seguimiento y análisis de datos relacionados con la calidad de los productos, lo que contribuye a mejorar la calidad y la consistencia de los productos manufacturados.



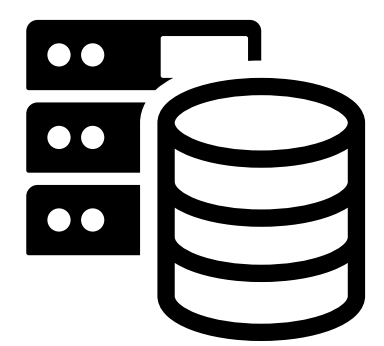
BD EN INGENIERÍA INDUSTRIAL

4. **Optimización de Recursos:** La gestión eficaz de recursos, como mano de obra, maquinaria y materiales, es crucial en la Ingeniería Industrial. Las bases de datos ayudan a programar y asignar recursos de manera óptima, maximizando la productividad.
5. **Análisis y Toma de Decisiones:** Las bases de datos permiten realizar análisis de datos avanzados, como pronósticos de demanda, análisis de tendencias y modelado de escenarios. Estos análisis proporcionan información valiosa para la toma de decisiones estratégicas.
6. **Mantenimiento Preventivo:** La Ingeniería Industrial se beneficia del mantenimiento preventivo de maquinaria y equipos. Las bases de datos ayudan a llevar un registro de las actividades de mantenimiento programado y a prevenir averías costosas.
7. **Cumplimiento de Normativas:** En muchos sectores industriales, existen regulaciones estrictas en cuanto a la recopilación y el almacenamiento de datos. Las bases de datos aseguran el cumplimiento de estas normativas al proporcionar un sistema de gestión de datos seguro y organizado.



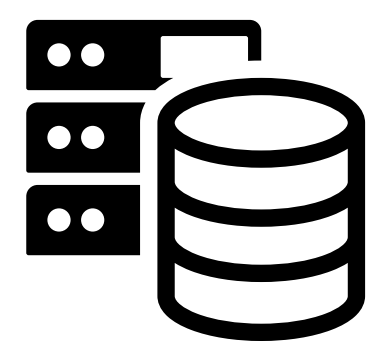
EJEMPLO 1: MANTENIMIENTO DE MAQUINARIA

La planta industrial "MetalInd" necesita modelar el mantenimiento de sus equipos. Cada máquina es intervenida por uno o más técnicos, y cada técnico repara una o más máquinas. De las máquinas se registra `Id_maquina`, `modelo` y `sector`; de los técnicos, `Id_tecnico`, `nombre` y `especialidad`. Además, se registra un historial de intervenciones que almacena la `fecha_intervención` y el `costo_reparación`.



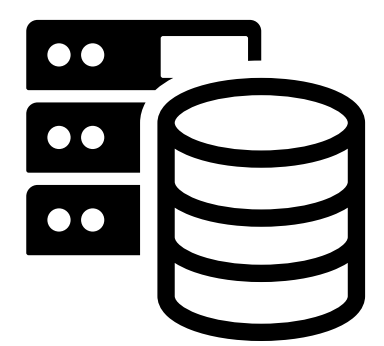
EJEMPLO 2: GESTIÓN DE ALMACÉN E INSUMOS

El área de logística desea modelar el stock de materias primas. Cada proveedor suministra uno o más insumos, y cada insumo es provisto por uno o más proveedores. De los proveedores se almacena `Id_proveedor`, razón social y teléfono; de los insumos, `Id_insumo`, descripción y unidad de medida. Se requiere registrar la cantidad exacta entregada en cada compra mediante la `fecha_entrega`.



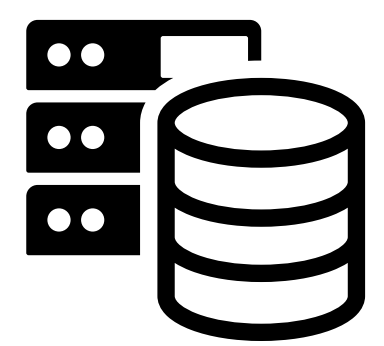
EJEMPLO 3: LÍNEA DE MONTAJE Y CALIDAD

En la línea de ensambaje, cada operario realiza auditorías sobre uno o más lotes de producción, y cada lote es revisado por uno o más operarios. De los operarios se guarda `Id_operario`, nombre y turno; de los lotes, `Id_lote`, tipo de producto y cantidad. Se registra la `fecha_auditoria` y el resultado (aprobado/rechazado) de cada control.



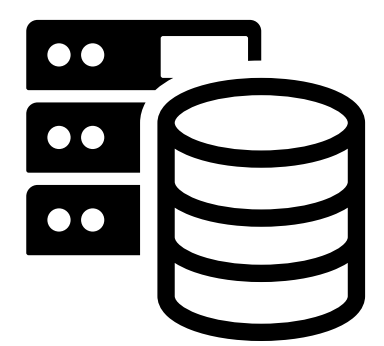
EJEMPLO 4: ÓRDENES DE PRODUCCIÓN

Una fábrica planifica su producción mediante órdenes. Cada orden pertenece a un único producto, pero un producto puede tener múltiples órdenes. De los productos se registra `Id_producto`, nombre y línea de fabricación; de las órdenes, `Id_orden`, fecha_emisión y cantidad planificada. Asimismo, cada orden utiliza uno o más centros de trabajo (máquinas), y cada centro procesa varias órdenes.



EJEMPLO 5 : PAÑOL DE HERRAMIENTAS

El pañol industrial requiere controlar el préstamo de herramientas. Cada operario retira una o más herramientas a lo largo del tiempo, y cada herramienta puede ser utilizada por distintos operarios. De los operarios se almacena `Id_operario` y nombre; de las herramientas, `Id_herramienta`, tipo y estado. Se debe registrar el detalle de cada préstamo guardando la `fecha_retiro`.



EJEMPLO 6: GESTIÓN LOGÍSTICA Y ENVÍOS

Para la distribución, la planta utiliza vehículos de transporte. Cada vehículo realiza uno o más viajes, y cada viaje es asignado a un único vehículo. De los vehículos se registra `Id_vehiculo`, patente y capacidad de carga; de los viajes, `Id_viaje`, destino y `fecha_salida`. Cada viaje es coordinado por un supervisor de logística del cual se conoce `Id_supervisor` y nombre.

¿PREGUNTAS?

```
render() {  
  return (  
    <React.Fragment>  
      <div className="py-5">  
        <div className="container">  
          <Title name="our" title="product">  
            <div className="row">  
              <ProductConsumer>  
                {(value) => {  
                  console.log(value)  
                }}  
            </ProductConsumer>  
          </div>  
        </div>  
      </div>  
    </React.Fragment>  
  )  
}
```



MUCHAS
GRACIAS!!!

Nos Vemos en la Siguiente Clase