

# Introducción a las Base de Datos



Profesor:

Eduardo Mónaco



1. Definición de base de datos:
  - Conjunto de datos relacionados y organizados
  - Almacenados sin redundancias perjudiciales
  - Independientes de los programas que los utilizan
2. Componentes de una base de datos:
  - Datos
  - Hardware
  - Software (DBMS)
  - Usuarios
3. Sistema de Gestión de Bases de Datos (DBMS):
  - Conjunto de programas que gestionan y controlan la base de datos
  - Proporciona una interfaz entre los datos, los usuarios y las aplicaciones

# Objetivos de un DBMS

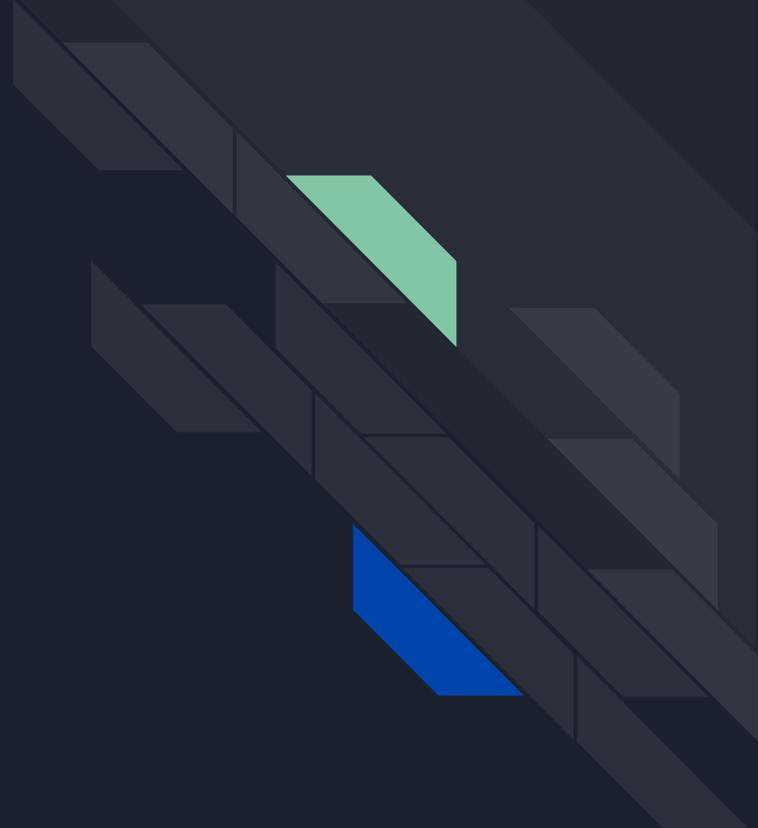
1. **Abstracción de datos:**
  - Niveles de abstracción: físico, conceptual, vistas
2. **Independencia de datos:**
  - Física: cambios en el almacenamiento no afectan a las aplicaciones
  - Lógica: cambios en el esquema conceptual no afectan a las aplicaciones
3. **Consistencia:**
  - Mantener la coherencia de los datos redundantes
4. **Seguridad:**
  - Control de acceso a los datos
  - Ejemplo: definición de permisos para usuarios y roles
5. **Integridad:**
  - Mantener la validez de los datos
  - Ejemplo: restricciones de integridad en el esquema
6. **Respaldo:**
  - Copias de seguridad y recuperación de datos
7. **Control de concurrencia:**
  - Manejo de accesos simultáneos
8. **Manejo de transacciones:**
  - Garantizar la atomicidad, consistencia, aislamiento y durabilidad (ACID)

# Arquitectura de acceso a Bases de Datos

1. Sistema cliente-servidor:
    - Cliente: interfaz de usuario
    - Servidor: gestión de datos
  2. Configuraciones:
    - a. Sala del anfitrión:
      - Cliente y servidor en la misma máquina
    - b. Cliente-Servidor clásico:
      - Clientes acceden a través de red local
    - c. Procesamiento distribuido:
      - Base de datos repartida en varios servidores
    - d. A través de servidores y aplicaciones:
      - Arquitectura de tres niveles: cliente, servidor de aplicaciones, servidor de base de datos
- 
- A decorative graphic on the right side of the slide, consisting of a series of dark grey, 3D-style rectangular blocks arranged in a descending staircase pattern from top-left to bottom-right. One block in the middle is highlighted in a light green color, and another block further down is highlighted in a blue color.

# Modelos de Bases de Datos

1. Modelos conceptuales:
  - Entidad-Relación
  - Orientado a objetos
2. Modelos lógicos basados en registros:
  - Estructura de árbol
  - Ejemplo: sistema de archivos
3. b. De red:
  - Estructura de grafo
  - Ejemplo: red de rutas
4. c. Relacional:
  - Basado en tablas
  - Ejemplo: hojas de cálculo interrelacionadas



## DIAGRAMA ENTIDAD-RELACIÓN

**Entidad:** objeto, real o abstracto, acerca del cual se recoge información de interés para la base de datos. Entidades fuertes: tienen existencia por sí mismas. (Alumnos, Empleados, Dpto.) Entidades débiles: dependen de otra entidad para su existencia (Hijos de Empleados)

**Atributos:** describen las características de una entidad. **Atributo con simple valor:** Cuando un atributo tiene un simple valor para una identidad particular. **Atributo Multivalor:** Cuando un atributo tiene una serie de valores para identificarse.

**Atributos Derivados:** Cuando los valores de un atributo son afines y el valor para este tipo de atributo se puede derivar de los valores de otros atributos. **Atributo Clave:** Una entidad usualmente tiene un atributo sobre el cual los valores son distintos para cada entidad individual, ésta nos permite identificar de forma única a una entidad en un conjunto de entidades.

**Atributos Nulos:** Se usa cuando una entidad no tiene valor para un atributo o que el valor es desconocido.

**Relación:** cualquier asociación R que pueda establecerse entre entidades.

### Gráfico del Diagrama Entidad -Relación

- 1.- Rectángulos, representan conjuntos de entidades.
- 2.- Elipses, representan atributos.
- 3.- Rombos, representan conjuntos de relaciones.

## Cardinalidad

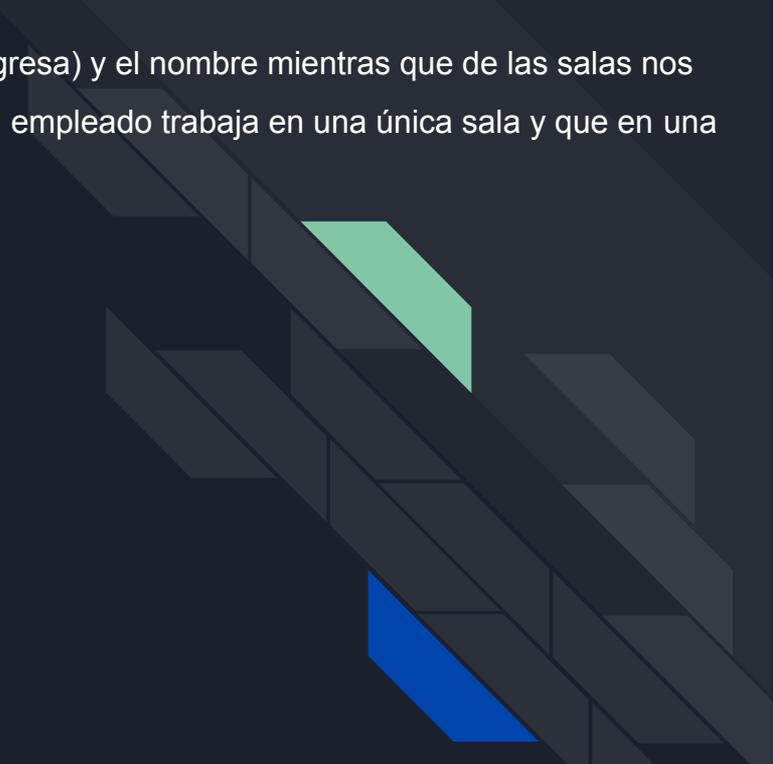
<b>(A) Relación 1:1</b>	<b>(B) Relación 1:N</b>
Cada persona maneja un vehículo, cada vehículo es manejado por una persona	Una persona maneja varios vehículos, un vehículo es manejado por una sola persona.
<b>(C) Relación N:1</b>	<b>(D) Relación N:M</b>
Un vehículo puede ser manejado por algunas personas. Una persona maneja un vehículo.	Una persona puede manejar un número de vehículos, un vehículo puede ser manejado por algunas personas.

## Ejemplo:

En un hospital se tiene un registro de pacientes, un registro de personal y uno de salas con funcionarios que trabajan en esas salas y con pacientes internados en esas salas. Del personal nos interesa el número de empleado, el nombre, la dirección y el teléfono. Sabemos que dos empleados no tienen el mismo número.

De los pacientes nos interesa el número de registro (le es asignado cuando ingresa) y el nombre mientras que de las salas nos interesa el nombre y la cantidad de camas que tiene. También se sabe que un empleado trabaja en una única sala y que en una sala trabajan varios empleados. Lo mismo ocurre con los pacientes.

Resolución: Pasos a seguir

1. identificar las Entidades del problema.
  2. identificar las Relaciones entre las entidades.
  3. representar los Atributos que interesan de las Entidades.
  4. determinar la Cardinalidad que se desea imponer.
- 

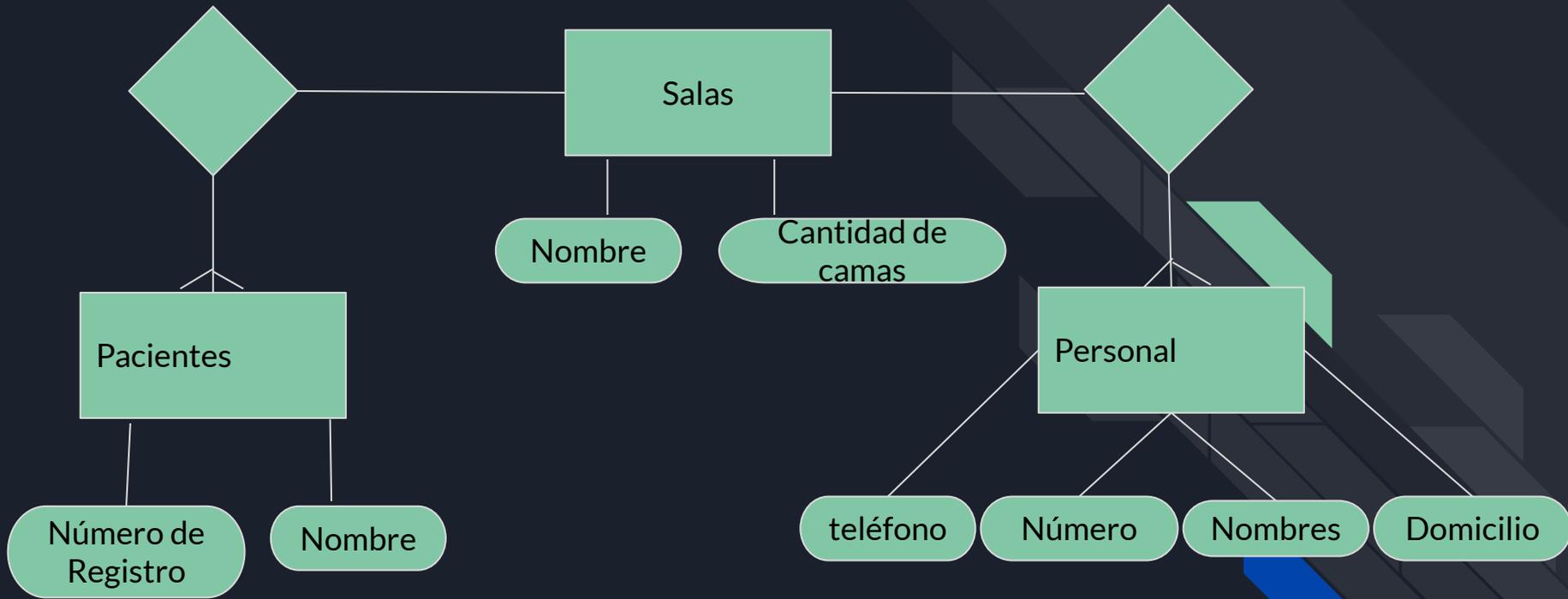
**Entidades:** Pacientes, Sala, Personal.

**Relaciones entre las entidades:** Los pacientes están internados en las salas y el personal trabaja en las salas.

**Atributos de las entidades:**

- Personal: número de funcionario, nombre, dirección y teléfono.
- Pacientes: número de registro, nombre.
- Salas: nombre, cantidad de camas.

**Cardinalidad:** un empleado trabaja en una única sala y en una sala trabajan varios empleados. Un paciente está internado en una sola sala pero en una sala hay varios pacientes.



## Reducción de los DER al Esquema Relacional:

### Principios:

- Todo tipo de entidad se convierte en una tabla.
- Los atributos en columnas de la tabla.
- Todo tipo de relación N:N se transforma en una tabla denominada intermedia.

En este tipo de esquemas cada **fila** de la **tabla** es un **registro único** que posee un **identificador** específico llamado **clave primaria** o **llave primaria** (PK - Primary Key).

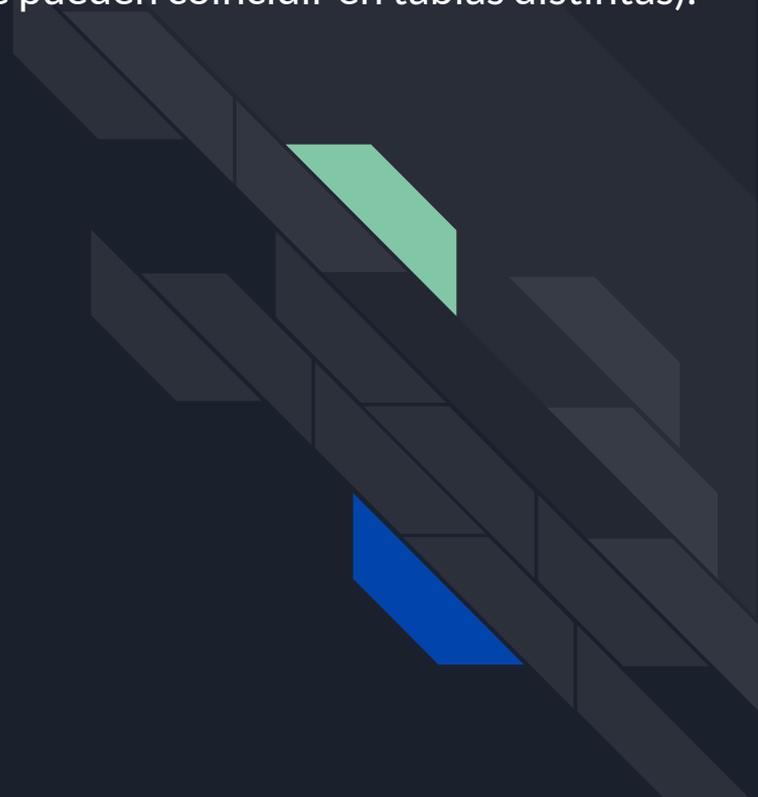
Las **columnas** de una tabla se corresponden con los **atributos** de los datos, y cada fila es un registro que (generalmente) tiene un valor para cada atributo.

atributo: id	atributo: nombre	atributo: apellido
1	Ruben Ramón Sixto	Alegre
2	Roberto	Flores

# Características del modelo relacional

- Cada tabla debe tener un **nombre único**.
  - Los datos de cualquier columna corresponden a un solo **tipo de dato** (**cadena**, **entero**, **doble**).
  - Las columnas de una relación se conocen como **atributos**.
  - El orden de los atributos no afecta: **los atributos no están ordenados**.
- 
- A decorative graphic on the right side of the slide consists of a grid of dark gray rectangular blocks. A green arrow points downwards from the top right towards the center. A blue square is located at the bottom right of the grid.

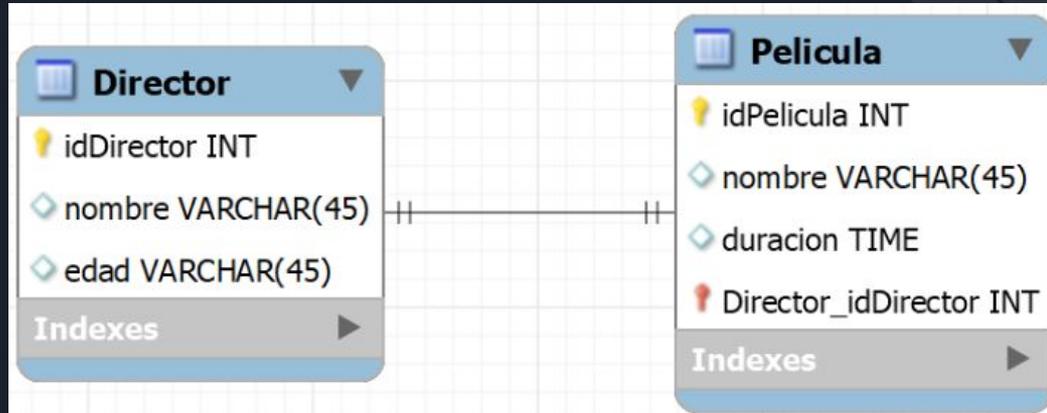
- Cada atributo de la tabla solo puede tener **un valor en cada tupla**.
- Cada **atributo** tiene un **nombre único en cada tabla** (aunque pueden coincidir en tablas distintas).
- El orden de las filas **no importa**.
- No se permiten filas repetidas mediante la **clave primaria**.



# Relaciones entre tablas

Una “relación” es una asociación que se crea entre tablas, con el fin de vincularlas y garantizar la integridad referencial de sus datos.

Para que una relación entre dos tablas exista, la tabla que deseas relacionar debe poseer una **clave primaria (PK)** o identificador único, mientras que la tabla donde estará el lado dependiente de la relación debe poseer una **clave foránea (FK Foreign Key)** o llave foránea de esa clave primaria. **Todas las relaciones tienen sentido bidireccional.**



## Veamos otro ejemplo

Profesor			
id_profesor	Nombre	Apellido	Edad
1	Agustín	Cocco	24

Curso			
id_curso	Nombre	Costo	id_profesor
1	Curso de Programación	500	1



## Relación 1 a 1

Una empresa necesita almacenar datos de contacto de sus empleados En este caso la relación puede establecerse de la siguiente manera:

Un empleado tiene una sola información de contacto.

Y una información de contacto pertenece a un solo empleado.

Dado que la información de contacto es la que depende principalmente del empleado, es en ella donde existirá la clave foránea para representar el vínculo.

## Relación 1 a N

Supongamos que tenemos ciudades en las cuales viven nuestras personas, pero cada persona solo puede pertenecer a una ciudad. Para este caso, pudiésemos leer la relación de esta manera:

Una ciudad tiene **muchas** (o ningún) personas.

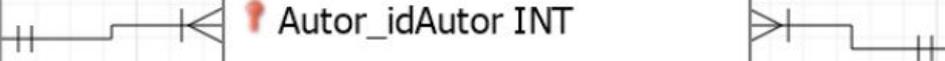
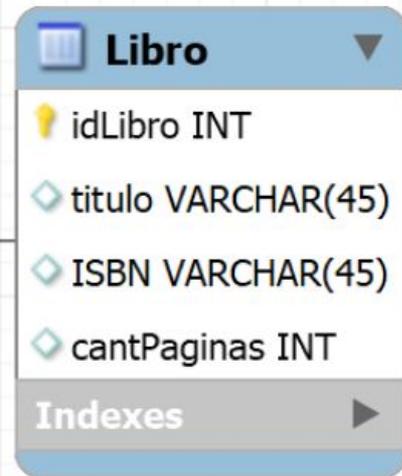
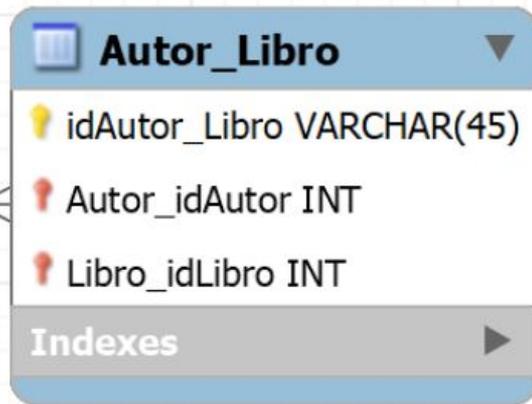
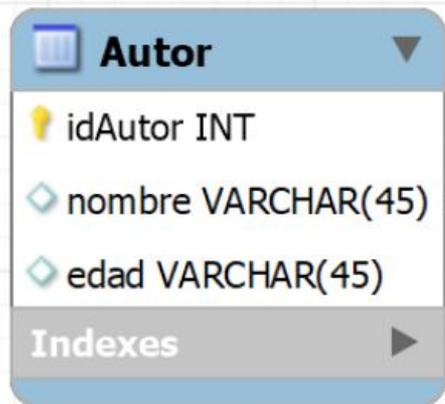
Y una persona vive en una **sola** ciudad.

Dado que la persona es la que necesita de la ciudad, es en ese dato (la persona) en donde existirá la clave foránea para representar el vínculo.

# Relación N a N

Supongamos que estás diseñando una base de datos para un sistema de **gestión de libros y autores**. En este caso, tendrías dos entidades principales: "**Libros**" y "**Autores**". La relación entre ellos es de **muchos a muchos**, ya que **un libro puede tener varios autores** y **un autor puede haber escrito varios libros**.

Para representar esta relación en una base de datos, necesitaremos una **tabla intermedia**, también conocida como **tabla de enlace**. Esta tabla actúa como un **punte** entre las entidades "**Libros**" y "**Autores**" y almacena la información necesaria para establecer la relación entre ellos.



ID_libro	Título	ISBN
1	El Gran Gatsby	9788491052334
2	1984	9780451524935
3	Orgullo y prejuicio	9780141439518
4	Don Quijote de la Mancha	9788423345198

Tabla Libro

ID_autor	Nombre	Apellido
1	F. Scott	Fitzgerald
2	George	Orwell
3	Jane	Austen
4	Miguel de	Cervantes

Tabla Autor

ID_relación	ID_libro	ID_autor
1	1	1
2	1	3
3	2	2
4	3	3
5	4	4

Tabla Autor\_Libro