

## 1.1 Introducción.



### ¿Qué es?

Colección organizada de datos que se almacena y gestiona electrónicamente

### ¿Qué permiten?

La *inserción, consulta, actualización y eliminación* de **datos** de manera eficiente

### ¿Cómo se administran?

**Sistemas de Gestión de Bases de Datos (SGBD)**

### ¿Qué ventajas ofrecen?

Interactuar con la información **sin necesidad de conocer los detalles técnicos**

### ¿Qué tipos existen? ¡¡Muchos tipos!! cómo relacionales, NoSQL, jerárquicas,



# Historia de las BBDD



Los orígenes de las BBDD se remontan a la antigüedad, con las bibliotecas

Existían sistemas de catalogación de obras y relacionadas con otros ámbitos prácticos como la recogida y archivo de información sobre cosechas o censos.

¿Qué problemas tenían?

- Búsquedas eran lentas
- Procesos poco eficaces (búsqueda, consulta, actualización, etc.)
- Todo era manual → podía existir fallos

FICHA DE PRÉSTAMO DE LIBROS			
Lector/a:			
TÍTULO	FECHA DE SALIDA	FECHA DE ENTRADA	OBSERVACIONES

FICHA DE LECTURA LIBROS (BIBLIOTECA DE CLASE)	
Nombre	Fecha
Curso	
Título	
Autor	
Editorial	

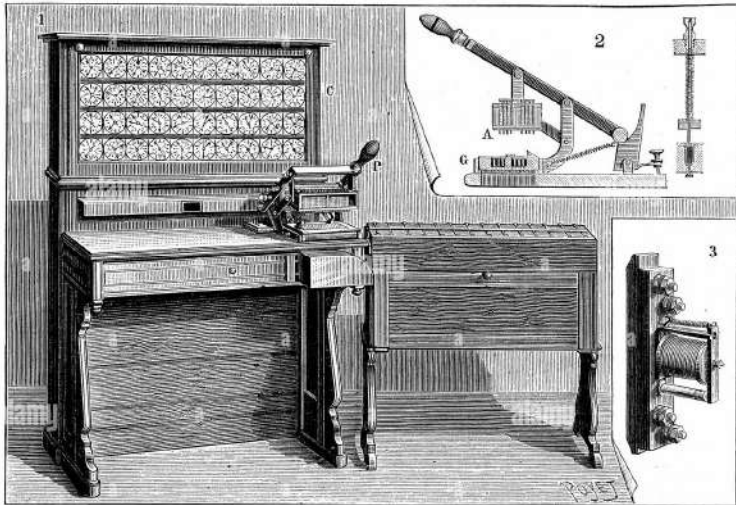
## 1.1 Introducción

Prof. Román-Herrera, J.C.



Realiza por parejas una búsqueda en la www con el objetivo de:

- i. Buscar quién fue la primera persona en la historia moderna (siglos XIX – XX) en realizar el tratamiento automático de la información.
- ii. ¿Cómo lo hizo?



**Herman Hollerith** (Buffalo, 29/02/1860 - 17/11/1929) está considerado como el primer ingeniero estadístico de la Historia, es decir, el primero que logró el tratamiento automático de información.

Hollerith observó que las preguntas contenidas en los censos se podían contestar con un "sí" o un "no". Entonces **ideó una tarjeta perforada**, que consistía en una cartulina en la que, según estuviera perforada o no en determinadas posiciones, se contestaban este tipo de preguntas.

### Década de 1950



Se inventan las cintas magnéticas con la finalidad de **suplir nuevas necesidades** ... *¿por ejemplo?*

**Automatizar información** de las nóminas, como por ejemplo el aumento de salario en un ascenso ... *¿Cómo lo hacían?*

Leían la información de una cinta (o más de una) y transferían la información a otra cinta ... *¿Qué suponía esto?*

¡¡ Podíamos tener un **BackUp**!!

*¿Qué problema existía con esta tecnología?*

Solo podíamos **leer la información de manera secuencial y ordenada**



### Década de 1960



#### ¿Qué relevancia tuvo en especial los HD?

- Agilización de las consultas en tiempo  
(no había que conocer la secuencia de las cintas)

#### ¿Qué eventos supusieron un avance para las BBDD?

- Adquisición masiva de ordenadores por bajada de precios
- Aparición de los primeros discos
- Aparición de los primeras BBDD en red (jerárquicas)



Mártil, I., & Mártil, I. (2020, 10 mayo). Los orígenes del almacenamiento de la información (V): Ilega el disco duro magnético. Un Poco de Ciencia, Por Favor. [URL](#)

Investigar la importancia histórica de la organización *Data Base Task Group* (**DBTG**) y su rol en el desarrollo de *Conference on Data Languages* (**CODASYL**) y el lenguaje de programación COBOL, con un enfoque en cómo esta iniciativa condujo al establecimiento de estándares de bases de datos y programación.

### 1. Orígenes de CODASYL:

- Investiga el contexto histórico de la creación de CODASYL ¿En qué año sucedió?.
- ¿Qué motivó la creación de esta organización y cuál era su propósito principal?

### 2. El rol del Data Base Task Group (DBTG):

- ¿Qué papel desempeñó el DBTG en CODASYL?
- ¿Cuáles fueron los objetivos específicos del DBTG en términos de bases de datos?

### 3. Desarrollo del lenguaje COBOL:

- Explica cómo CODASYL impulsó la creación de COBOL.
- ¿Qué características hacían a COBOL innovador en ese momento?
- ¿Por qué fue relevante para las empresas de la época?

### 4. La falta de un estándar inicial:

- CODASYL no llegó a establecer un estándar para bases de datos. ¿Por qué crees que esto sucedió?
- ¿Qué papel jugó posteriormente la ANSI en la creación de estándares?

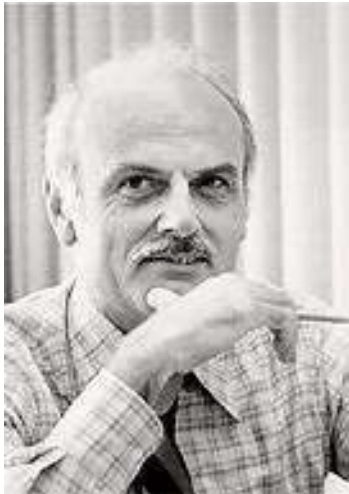
### 5. Influencia de CODASYL y DBTG en la informática moderna:

- ¿Cómo crees que el trabajo de CODASYL y DBTG ha influido en el desarrollo de tecnologías de bases de datos y lenguajes de programación actuales?



Investigación

### Década de 1970



Edgar Frank Codd  
1923 - 2003

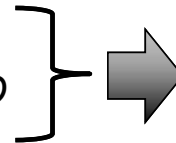
¿Quién provocó una gran revolución en el campo de las BBDD?

Ingeniero de la empresa IBM publica...

*“Un modelo relacional de datos para grandes bancos de datos compartidos”*

¿De qué habla este artículo?

- Del modelo relacional
- De las 12 reglas de CODD



Aparecen los primeros **Sistemas de Gestión de Bases de Datos Relacionales (DBMS)**

¿Qué proyecto se desarrolló a posteriori del artículo?

- System R (IBM)



**Mejora SUSTANCIAL** del **MODELO RELACIONAL** de las BBDD vs Modelos Jerárquicos y de red

¿Qué DBMS surge?

- ORACLE

El título original es "A Relational Model of Data for Large Shared Data Banks" y fue publicado por la revista Communications of the ACM (<http://cacm.acm.org>)

### Década de 1980 - 1990



Structure Query Language

Se desarrolla el lenguaje llamado SEQUEL que más tarde paso a llamarse SQL **¿Qué significa su acrónimo?**

En 1986 ANSI adopta SQL como estándar para lenguajes relacionales

En 1987 se transformó en estándar ISO

En 1990 las BBDD evolucionaron a las BBDD orientadas a objetos, surgiendo programas como...





Comprender las 12 reglas de Codd, que definen los principios fundamentales de los sistemas de bases de datos relacionales, e investigar cómo estas reglas se aplican en las bases de datos modernas.

### 1. Contexto histórico de las 12 reglas:

- ¿Quién fue Edgar F. Codd y qué importancia tuvo en el desarrollo de las bases de datos relacionales?.
- ¿Por qué Codd formuló estas 12 reglas? ¿Cuál era su objetivo con ellas?



### 2. Las 12 reglas de Codd:

- Explica cada una de las 12 reglas de Codd.
- ¿Por qué cada una de estas reglas es importante para un sistema de bases de datos relacional?

Investigación

### 3. Impacto de las reglas en la evolución de los sistemas de bases de datos:

- ¿Cómo crees que las 12 reglas de Codd influyeron en el desarrollo de bases de datos relacionales modernas?.
- ¿Qué limitaciones ves en las reglas de Codd cuando se comparan con los sistemas de bases de datos no relacionales, como NoSQL?



VS



Por ejemplo, MySQL y PostgreSQL aplican las reglas de Acceso Garantizado (Regla 2) y Tratamiento Sistemático de Nulos (Regla 3), asegurando que los datos se accedan fácilmente y los valores nulos se manejen adecuadamente.



NoSQL

Los sistemas NoSQL no siguen las 12 reglas, ya que están diseñados para otros tipos de arquitecturas y cargas de trabajo, por lo que no se enfocan en el modelo relacional.

**BBDD** → serie de datos organizados y relacionados entre sí, los cuales son almacenados y explotados por los sistemas de información de una empresa (California, 1963)

### ¿Qué funciones aportan?

1. Almacenamiento
2. Organización
3. Acceso Seguridad
4. Manipulación
5. Salvaguarda



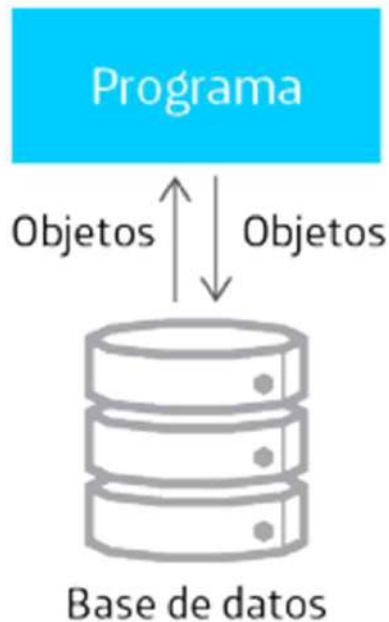
¿Qué son las BBDD relacionales? → Aquella que se basa en establecer **interconexiones** (*relaciones*) entre los **datos guardados en las tablas** para obtener información de las entidades

# CARACTERÍSTICAS DE LAS BBDD RELACIONALES



1. Están compuestas de tablas y/o relaciones
2. No pueden existir dos tablas con el mismo nombre
3. Cada tabla tiene su conjunto de filas y registros
4. La relación entre una tabla principal y una secundaria se lleva a cabo por medio de las claves primarias y foráneas
5. La clave primaria es el identificador principal de un registro dentro de una tabla y debe cumplir con la integridad de los datos.
6. Las claves externas se colocan en la tabla secundaria, contienen el mismo valor que la clave primaria del registro principal; por medio de estas claves se establecen las relaciones.

# BASE DE DATOS ORIENTADAS A OBJETOS (BBDD O-O)



Sistema de gestión de bases de datos que almacena información en forma de objetos, tal como ocurre en la programación orientada a objetos.

### ¿Cómo organizan los datos?

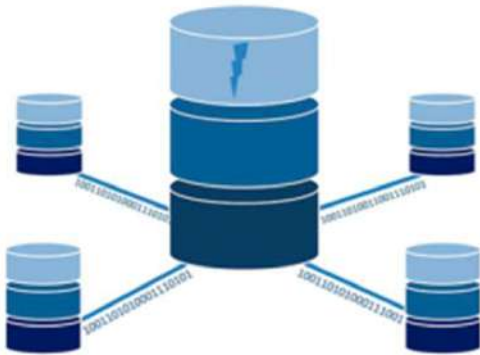
En lugar de usar tablas (como en las bases de datos relacionales), los datos se organizan en objetos que tienen atributos (datos) y métodos (funciones) asociados.

### ¿Esto qué permite?

Esto permite que los datos y la lógica que los manipula estén integrados, facilitando el manejo de estructuras complejas y relaciones jerárquicas



# CARACTERÍSTICAS DE LAS BBDD O-O



## Encapsulación

Propiedad que permite ocultar la información (atributos y métodos) al resto de los objetos, impidiendo así accesos incorrectos o conflictivos

## Herencia

Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases

## Polimorfismo

Capacidad de que los métodos de un objeto puedan comportarse de manera diferente según el tipo de objeto que los invoque, aunque compartan el mismo nombre. *¿Ejemplo?*

**Clase "Animal"** con un método llamado **hacerSonido()**, diferentes **subclases** como **"Perro"** y **"Gato"** pueden implementar ese método de forma distinta: el *perro puede devolver "ladrar"* y el *gato "maullar"*. Al invocar **hacerSonido()** en un objeto, **el sistema sabrá cuál es la implementación correcta según el tipo del objeto** (Perro o Gato).

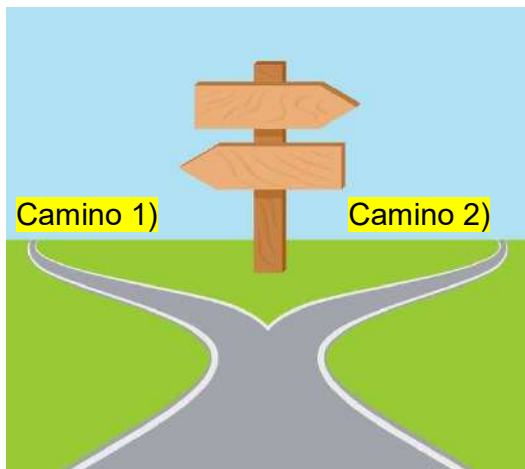


Realiza por parejas una búsqueda en la www con el objetivo de:

- i. ¿Qué es un Sistema de Gestión de BBDD?.
- ii. ¿Qué funciones aporta?
- iii. ¿Qué es un Sistema de Gestión de Bases de Datos Distribuidas (SGBDD)?
- iv. ¿Cuáles son las ventajas de utilizar un SGBDD?
- v. ¿Qué diferencia a un SGBDD de un SGBD centralizado?
- vi. ¿Qué son la fragmentación, replicación y asignación en un SGBDD?
- vii. ¿Cuáles son los principales retos al implementar un SGBDD?
- viii. ¿Qué tipos de arquitecturas existen para un SGBDD?
- ix. ¿Cómo se asegura la consistencia de los datos en un SGBDD?



# ¿QUÉ CAMINOS QUE PUEDE SEGUIR EL DISEÑO DE UNA BBDD?



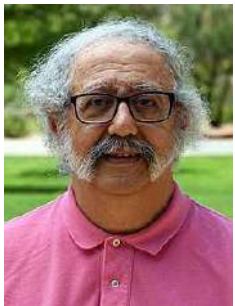
### Camino 1

Partiendo de la observación del universo de estudio con el fin de obtener un conjunto de esquemas relacionales, que contengan atributos y restricciones

### Camino 2

Dividir el diseño en 2 fases:

1. Definir el modelo conceptual y su esquema
2. Transformar el esquema conceptual a esquema relacional dado una serie de reglas



Ramez A. Elmasri



Shamkant Navathe

**Elmasri** y **Navathe** explican que el diseño de los sistemas de información y de las **bases de datos** sigue un proceso con varias etapas el cual **consta de seis fases importantes**. Estas fases *representan los pasos que se deben seguir para crear una base de datos que funcione bien y cumpla con los requisitos del sistema.*

# FASES DE DISEÑO DE ELMASARI Y NAVATHE

## Fase 1: Recopilación y análisis de requerimientos

Se quiere conocer las expectativas del usuario sobre la base de datos

Se identifican:

- los grupos de usuarios reales y posibles
- las áreas de aplicación

Se revisa la documentación existente

Se analiza el entorno operativo y los requerimientos de procesado

Se realizan entrevistas y cuestionarios con los usuarios.



# FASES DE DISEÑO DE ELMASARI Y NAVATHE

**Fase 2: Diseño conceptual de la base de datos** → Se subdivide en 2:

**Fase 2.a** : Diseño del esquema conceptual

Se crea un modelo de datos a alto nivel que define cómo se organizarán los datos, pero sin preocuparse todavía por cómo se implementará en un sistema de gestión de bases de datos (SGBD) específico.

Este diseño es abstracto y no está vinculado a ningún software o tecnología específica de bases de datos.

Es una especie de mapa que describe la estructura de la base de datos sin enfocarse en los detalles técnicos.

Se recopilan los requerimientos de los usuarios antes de crear el esquema, y se integran todos en una única vista global.

Se desarrollan varios esquemas individuales, cada uno enfocado en un requerimiento específico, y luego se combinan para formar un esquema global después de crear las vistas parciales.



# FASES DE DISEÑO DE ELMASARI Y NAVATHE

**Fase 2:** Diseño conceptual de la base de datos → Se subdivide en 2:

**Fase 2.b** : Diseño de transacciones

Se enfoca en identificar y diseñar las transacciones o procesos que manejarán los datos en la base de datos.

Las transacciones son las aplicaciones que permiten a los usuarios interactuar con los datos, como leer, modificar o eliminar información.

Para descubrir estas transacciones, se estudian las **entradas y salidas de datos** y cómo se comportan en el sistema. A partir de esto, se clasifican las transacciones en:

- Transacciones de recuperación: Solo leen datos (consultas).
- Transacciones de actualización: Modifican los datos (inserciones, eliminaciones, actualizaciones).
- Transacciones mixtas: Hacen ambas cosas, leer y modificar datos.

# FASES DE DISEÑO DE ELMASARI Y NAVATHE

## Fase 3: Elección de un SGBD

Se consideran diferentes factores:

- **Factores técnicos**  
Se refiere a las características del sistema o del software, como la capacidad de almacenamiento, velocidad de procesamiento, compatibilidad con otras tecnologías, etc.
- **Factores económicos**  
Se analizan los costos asociados, como la compra de licencias de software, hardware, mantenimiento, y si la inversión ofrece un buen retorno en términos de beneficios.
- **Factores de servicio técnico y formación**  
Se considera el soporte técnico necesario para operar el sistema y el entrenamiento que se debe proporcionar a los usuarios para que puedan utilizarlo eficazmente.
- **Factores organizativos**  
Estos están relacionados con cómo el sistema afectará la estructura y funcionamiento de la organización, como la coordinación entre departamentos o el impacto en la toma de decisiones.
- **Factores de rendimiento**  
Se evalúa el rendimiento del sistema en términos de velocidad, capacidad de respuesta y eficiencia en el manejo de grandes volúmenes de datos.



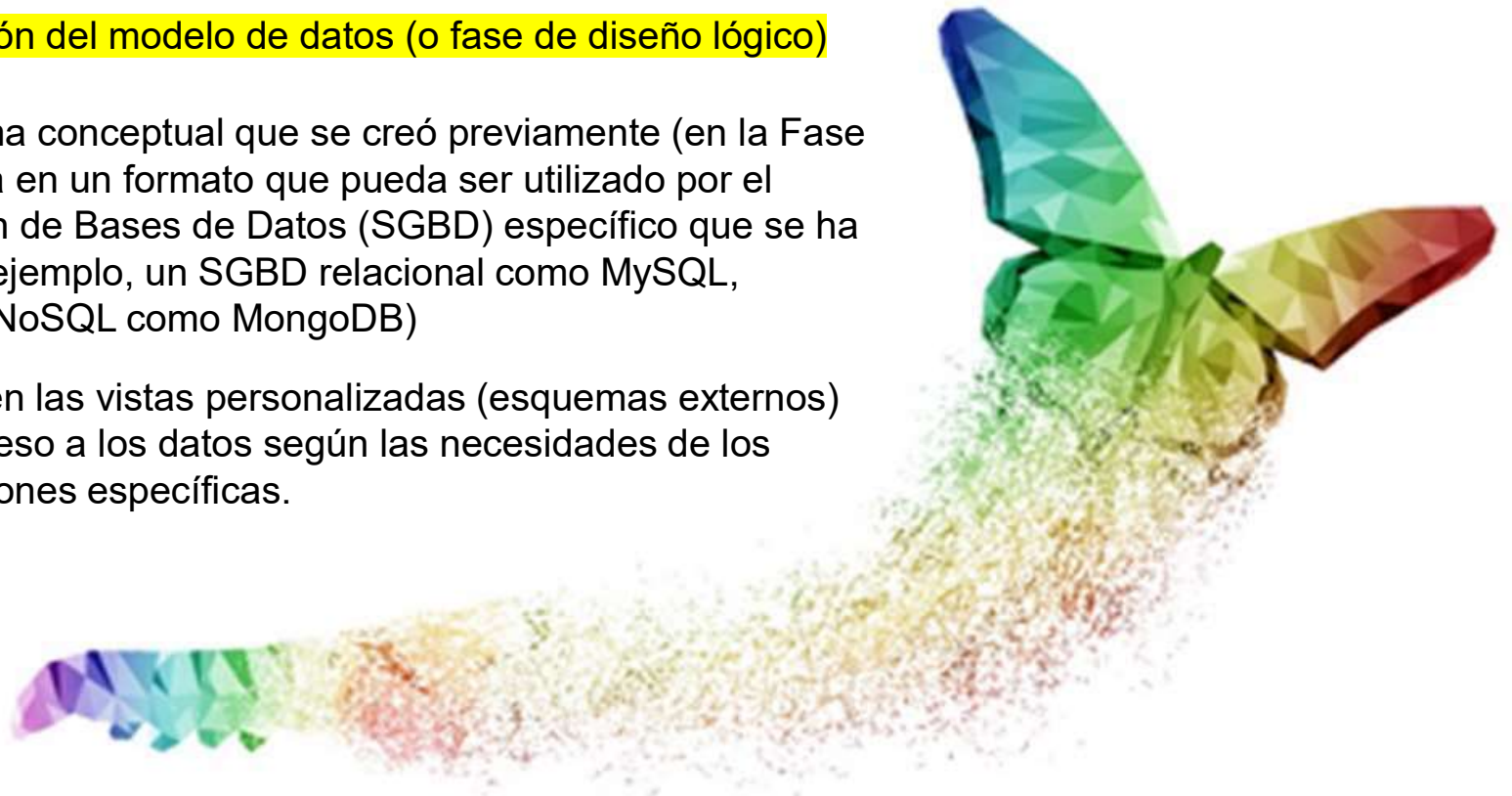
Resulta **difícil la medida y cuantificación** ponderada de los diferentes factores, con el fin de escoger el que mejor se adapte

# FASES DE DISEÑO DE ELMASARI Y NAVATHE

## Fase 4: Transformación del modelo de datos (o fase de diseño lógico)

Se toma el esquema conceptual que se creó previamente (en la Fase 2a) y se transforma en un formato que pueda ser utilizado por el Sistema de Gestión de Bases de Datos (SGBD) específico que se ha seleccionado (por ejemplo, un SGBD relacional como MySQL, PostgreSQL, o un NoSQL como MongoDB)

Además, se definen las vistas personalizadas (esquemas externos) que facilitan el acceso a los datos según las necesidades de los usuarios o aplicaciones específicas.



# FASES DE DISEÑO DE ELMASARI Y NAVATHE

## Fase 5: Diseño de la base de datos física

Se decide cómo se almacenarán y accederá a los datos para que las aplicaciones que usen la base de datos funcionen de manera eficiente.

Elementos clave:

### 1. Estructuras de almacenamiento:

Se refiere a cómo y dónde se guardarán los datos físicamente en el sistema (por ejemplo, en discos duros o en memoria).

### 2. Estructuras de acceso:

Hace referencia a los métodos o técnicas que se usarán para recuperar los datos almacenados. Esto puede incluir la creación de índices, la definición de claves primarias y foráneas, y la optimización de las consultas para asegurar que los datos se puedan buscar y recuperar de manera rápida y eficiente.

# FASES DE DISEÑO DE ELMASARI Y NAVATHE

## Fase 6: Implementación del sistema de base de datos

Se lleva a cabo la implementación real del sistema. Es decir, todo lo que se ha diseñado previamente se transforma en un sistema funcional.

¿Qué es lo que sucede en esta fase?

### 1. Creación del esquema de la base de datos:

Se construyen las tablas, índices y relaciones definidas durante el diseño conceptual y físico. Estas estructuras organizan los datos y permiten que el sistema funcione según lo planificado.

Se "compila" o se carga este esquema en el **Sistema de Gestión de Bases de Datos (SGBD)** elegido, haciendo que el diseño teórico se convierta en una base de datos operativa.

### 2. Creación de los archivos de bases de datos:

Se generan los archivos físicos donde se almacenarán los datos. Estos pueden ser archivos en disco que el SGBD utilizará para guardar y gestionar la información que se va insertando en la base de datos

### 3. Implementación de las transacciones:

Se crean las transacciones que interactuarán con la base de datos, es decir, los programas o aplicaciones que permitirán a los usuarios consultar, modificar, insertar o eliminar datos.

Estas transacciones son el resultado del diseño hecho previamente y suelen implementarse mediante aplicaciones que interactúan con la base de datos, ejecutando las operaciones necesarias.



# TIPOS DE USUARIOS



### USUARIOS FINALES

- Uso limitado de las capacidades del sistema
- Realizan operaciones de introducción manipulación y consulta de datos

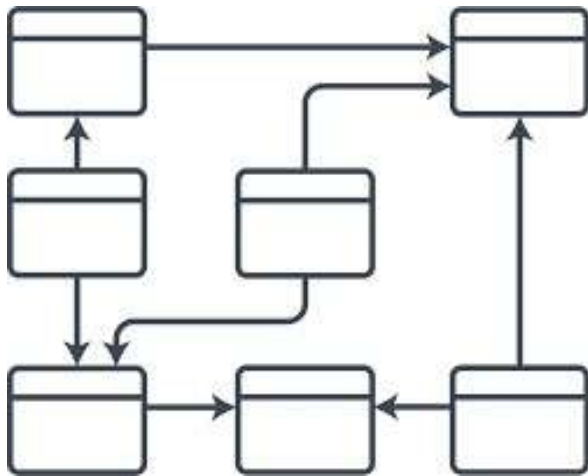
### PROGRAMADORES DE BBDD

- Escriben aplicaciones para ejecución de tareas de los usuarios finales

### ADMINISTRADOR DE LA BBDD (DBA)

- Realizan **estrategias de respaldo** y **recuperación** de la bbdd
- Definen controles de autorización y procedimientos de validación
- *Crean* y *almacenan* **estructuras** en la bbdd

### MODELOS DE DATOS



→ Forma en que representamos y estructuramos la **información** en una base de datos para poder organizar y gestionar los datos correctamente.

Es una **abstracción del "universo de discurso"**, lo que significa que es una representación simplificada de la realidad o del contexto del negocio, para poder trabajar con esa información dentro del sistema.

¿Qué tipos hay? → Hay 3 tipos:

- a) Modelos lógicos basados en objetos
- b) Modelos lógicos basados en registros
- c) Modelos físicos de datos

¿Cuál es la definición oficial de modelo?

"...al instrumento que se aplica a una parcela del mundo real (universo del discurso) para obtener una estructura de datos a la que denominamos esquema. Esta distinción entre el modelo (instrumento) y el esquema (resultado de aplicar el instrumento) es importante... Es importante también distinguir entre mundo real y universo del discurso, ya que este último es la visión que del mundo real tiene el diseñador... podemos definir un modelo de datos como un conjunto de conceptos, reglas y convenciones que nos permiten describir los datos del universo del discurso." - (MIGUEL y PIATTINI, 1993)

### MODELOS FÍSICOS BASADOS EN DATOS

Este tipo de modelo describe cómo los datos están realmente almacenados en el hardware, es decir, cómo los datos son organizados físicamente en disco, memoria u otros dispositivos de almacenamiento.

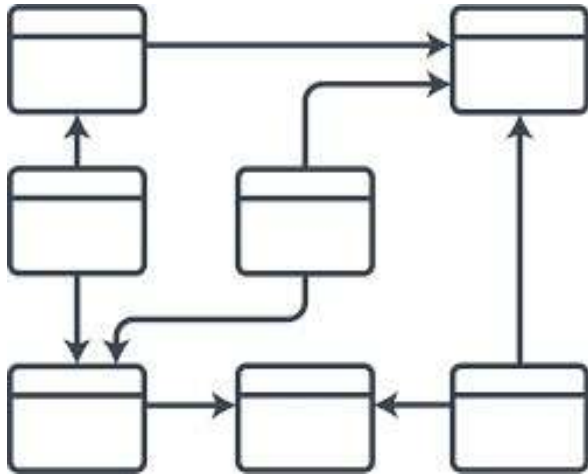
Son muy poco usados, y algunos autores lo definen como “*modelos de datos primitivos*”...

*¿Qué modelos hay?*

- a) Modelo unificador
- b) Modelo de elementos



### MODELOS DE DATOS



→ Forma en que representamos y estructuramos la **información** en una base de datos para poder organizar y gestionar los datos correctamente.

Es una **abstracción del "universo de discurso"**, lo que significa que es una representación simplificada de la realidad o del contexto del negocio, para poder trabajar con esa información dentro del sistema.

¿Qué tipos hay? → Hay 3 tipos:

a) Modelos lógicos basados en objetos

b) Modelos lógicos basados en registros

c) Modelos físicos de datos

# MODELOS LÓGICOS BASADOS EN OBJETOS

Estos modelos representan los datos en términos de objetos del mundo real y sus relaciones.

*¿Qué es un objeto?*

- Un objeto podría ser cualquier cosa del universo real que tenga atributos, como una persona, un producto, o una orden.

*¿Qué modelos hay?*

### a) Entidad Relación (E-R)

- Se basa en una percepción del mundo compuesta por **elementos**, llamados **entidades**, y **relaciones** entre ellos.
- Las *entidades* se diferencian unas de otras **a través** de *atributos*.

### b) Orientado a objetos

- Extiende el concepto del modelo E-R y se basa en la **programación orientada a objetos**.
- Contienen valores y métodos, entendidos como órdenes que actúan sobre los valores, en niveles de anidamiento. Los **objetos** se agrupan en clases.



# MODELOS LÓGICOS BASADOS EN OBJETOS

*¿Cuándo usamos estos modelos?*

- Ideal cuando trabajas con información compleja que tiene muchas relaciones y características, y cuando el software está diseñado de manera orientada a objetos.

*¿Podrías poner un ejemplo?*

- Un modelo de datos orientado a objetos en el que cada "persona" tiene atributos como nombre, dirección, y teléfono.
- Este enfoque es común en bases de datos orientadas a objetos, que tratan a los datos como objetos con propiedades y métodos, como en la programación orientada a objetos

# MODELOS LÓGICOS BASADOS EN OBJETOS

*¿Cuáles son las características clave de los modelos basados en objetos?*

### 1. Flexibilidad y complejidad

- Pueden modelar estructuras complejas y relaciones detalladas.

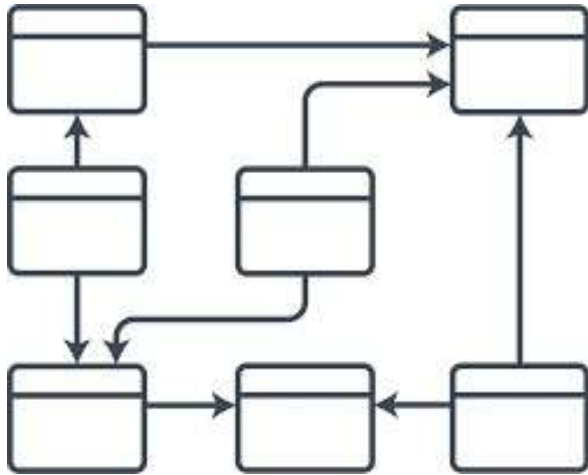
### 2. Alineación con la programación

- el modelo orientado a objetos es coherente con la programación orientada a objetos, lo que facilita la integración con sistemas que usen este paradigma.

### 3. Relaciones más explícitas

- El modelo E-R pone un fuerte énfasis en las relaciones entre entidades..

### MODELOS DE DATOS



→ Forma en que representamos y estructuramos la **información** en una base de datos para poder organizar y gestionar los datos correctamente.

Es una **abstracción del "universo de discurso"**, lo que significa que es una representación simplificada de la realidad o del contexto del negocio, para poder trabajar con esa información dentro del sistema.

¿Qué tipos hay? → Hay 3 tipos:

a) Modelos lógicos basados en objetos

b) Modelos lógicos basados en registros

c) Modelos físicos de datos

# MODELOS LÓGICOS BASADOS EN REGISTROS

Estos modelos representan los datos utilizando **registros fijos** (también llamados "**filas**" o "**tuplas**").

Son muy **comunes en bases de datos relacionales**. En este tipo de modelo, la información se organiza en tablas (entidades) y campos (columnas).

*¿Qué modelos hay?*

- a) **Modelo relacional** → Los datos se organizan en tablas que contienen filas (registros) y columnas (atributos). Modelo usado en MySQL, PostgreSQL o SQL Server
- b) **Modelo de red** → Los datos se organizan en una estructura de árbol en la que cada registro tiene un padre y puede tener varios hijos. Solo existe una relación de uno a muchos.
- c) **Modelo jerárquico** → Es una extensión del modelo jerárquico, pero permite relaciones muchos a muchos, lo que significa que un registro puede tener varios padres y varios hijos.

# MODELOS LÓGICOS BASADOS EN REGISTROS

*¿Cuándo usamos estos modelos?*

- Muy útil para manejar datos estructurados que pueden organizarse en filas y columnas.
- Este es el enfoque más común en bases de datos empresariales y es utilizado por sistemas como SQL.

*¿Podrías poner un ejemplo?*

- En una base de datos relacional como MySQL o SQL Server, una tabla de "Clientes" puede tener columnas como "Nombre", "Correo Electrónico", "Dirección". Cada registro (o fila) de la tabla sería un cliente con esos datos específicos

# MODELOS LÓGICOS BASADOS EN REGISTROS

*¿Cuáles son las características clave de los modelos basados en registros?*

### 1. Estructura fija

- Los modelos basados en registros tienen una estructura muy rígida (tablas, jerarquías o redes).

### 2. Optimización

- Suelen ser más fáciles de optimizar y consultar para grandes volúmenes de datos, especialmente el modelo relacional..

### 3. Acceso mediante claves

- Las relaciones entre tablas o registros se manejan a través de claves (primarias y foráneas), lo que facilita las consultas.

# ¿CUÁLES SON LOS OBJETIVOS DE LOS MODELOS DE DATOS?



### 1. LA FORMALIZACIÓN

*Definir formalmente las estructuras permitidas y las restricciones*

*Facilitar la elección del lenguaje de datos.*

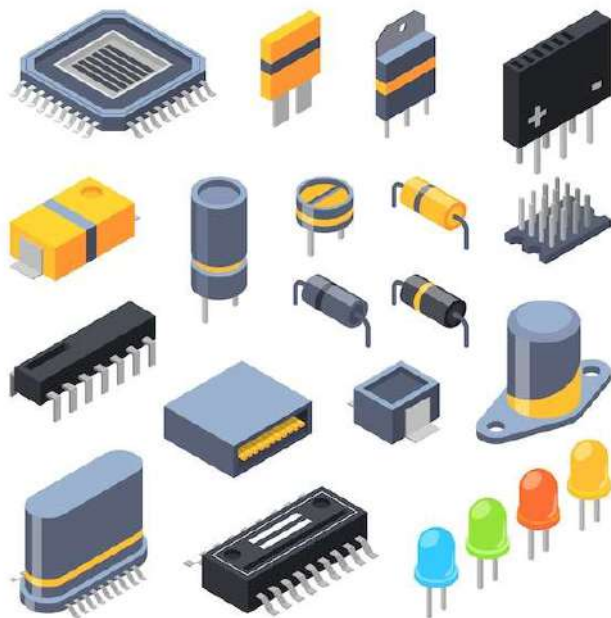
### 2. EL DISEÑO

*Establecen las bases, junto con los lenguajes, documentación y otras herramientas, para el desarrollo de metodologías de diseño de Bases de Datos.*

*Pronostican posibles impactos en las modificaciones que sufran los S.I*



# ¿CUÁLES SON LOS COMPONENTES DE LOS MODELOS DE DATOS?



### 1. COMPONENTES ESTÁTICOS

- Aquellos que describen la estructura de los datos en un momento dado, es decir, cómo se organizan las entidades y las relaciones entre ellas.
- Estos elementos son más "permanentes" o fijos, porque no cambian con cada operación en la base de datos.
- Constituidos por:
  - ✓ Entidad
  - ✓ Relación
  - ✓ Relación estática
  - ✓ Objeto compuesto
  - ✓ Generalización

### 2. COMPONENTES DINÁMICOS

- Describen el comportamiento o la evolución de los datos a través de acciones, operaciones o eventos que pueden modificar la base de datos.
- Los componentes dinámicos se centran en cómo los datos cambian con el tiempo.
- Constituidos por:
  - ✓ Operación
  - ✓ Transacción
  - ✓ Restricción dinámica

# COMPONENTES ESTÁTICOS DE LOS MODELOS DE DATOS

1. **ENTIDAD** → Una identidad es cualquier cosa u objeto que es distinguible del resto, y se encuentra presente en el mundo real. ¿Qué tipos puede ser? **Concreta** VS **Abstracta**
2. **RELACIÓN** → Se denomina “relación” a la forma en la que se vinculan diferentes identidades. ¿Ejemplo?  
la relación entre un cliente y un pedido (un cliente puede **hacer varios pedidos**).
3. **RESTRICCIÓN ESTÁTICA** → Son relaciones entre entidades que no cambian frecuentemente. ¿Ejemplo?  
la relación de un cliente con su dirección (que probablemente cambie poco).
4. **OBJETO COMPUESTO** → Son entidades que se forman a partir de otras entidades. ¿Ejemplo?  
un “Pedido”, que se compone de varios “Productos”.
5. **GENERALIZACIÓN** → Proceso de agrupar entidades en clases más amplias. ¿Ejemplo?  
se puede generalizar las entidades “Cliente Corporativo” y “Cliente Individual” bajo la entidad más general “Cliente”.

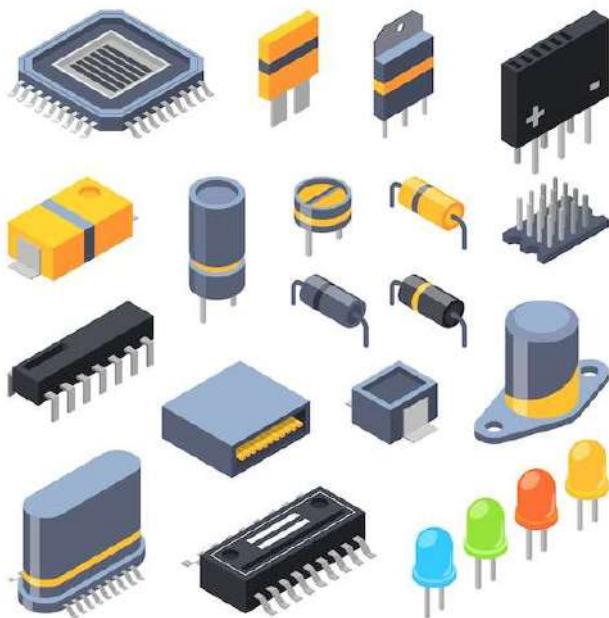
# ¿CUÁLES SON LOS COMPONENTES DE LOS MODELOS DE DATOS?

## 1. COMPONENTES ESTÁTICOS

- Aquellos que describen la estructura de los datos en un momento dado, es decir, cómo se organizan las entidades y las relaciones entre ellas.
- Estos elementos son más "permanentes" o fijos, porque no cambian con cada operación en la base de datos.
- Constituidos por:
  - ✓ Entidad
  - ✓ Relación
  - ✓ Relación estática
  - ✓ Objeto compuesto
  - ✓ Generalización

## 2. COMPONENTES DINÁMICOS

- Describen el comportamiento o la evolución de los datos a través de acciones, operaciones o eventos que pueden modificar la base de datos.
- Los componentes dinámicos se centran en cómo los datos cambian con el tiempo.
- Constituidos por:
  - ✓ Operación
  - ✓ Transacción
  - ✓ Restricción dinámica



# COMPONENTES DINÁMICOS DE LOS MODELOS DE DATOS

1. **OPERACIÓN** → Son las acciones realizadas sobre los datos **¿Cuáles son?**  
*Inserción , Actualización , Borrado y Lectura*
2. **TRANSACCIÓN** → Son un conjunto de operaciones que se ejecutan de forma completa o no se ejecutan en absoluto **¿Qué garantizan?** la *integridad* y *consistencia* de los datos
  - *integridad*: mantener la exactitud y corrección de los datos en todo momento.
  - *consistencia*: los datos deben cumplir con todas las reglas y restricciones establecidas en la base de datos, tanto antes como después de la transacción.
3. **RESTRICCIÓN DINÁMICA** → Son un conjunto reglas o limitaciones que deben cumplirse mientras se realizan operaciones o transacciones **¿Ejemplo?**  
*una restricción dinámica podría ser que no se puede realizar un pedido si el cliente no tiene fondos suficientes en su cuenta.*

**¿Qué es el Modelo E-R?** → representación gráfica y conceptual utilizada en el diseño de BBDD propuesta por **Peter Chen** en 1976

**¿En qué consiste?** → Modelar en base al concepto de:

### 1. Entidad

- Persona, objeto (real o abstracto) o animal sobre el cual queremos tener información y que tiene existencia por sí mismo en el mundo real.

### 2. Interrelación

- asociación o correspondencia entre entidades.

**¿Qué objetivo persigue el *modelado de datos*?**

1. Obtener toda la información de interés para el sistema
2. Representar la información
3. Independizar los datos de los recursos físicos y requerimientos del usuario
4. Proporcionar a los usuarios un sistema de información tal y como lo necesitan.



*Peter Chen*