



# UNIDAD II

Diseño Conceptual y Lógico  
de una Base de Datos:

Modelo Entidad/Relación Extendido  
(Elmasri-Korth)

# Proceso de Construcción de una base de datos





# Diseño Lógico de una Base de Datos

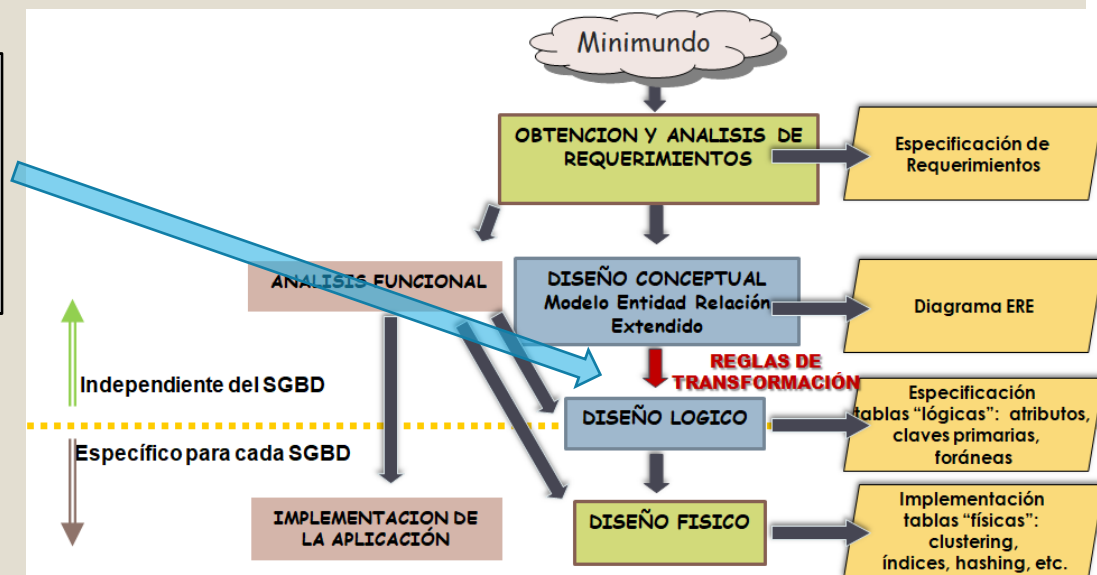
Reglas de Conversión del Modelo Conceptual al  
Modelo Lógico

# Diseño de una Base de Datos

El proceso de construcción de una bd que abarcamos en esta unidad comprende:

1. Elaborar el **Modelo Conceptual** de la base de datos utilizando el enfoque Entidad/Relación Extendido
2. Mapear el Modelo Conceptual obtenido (**Modelo Entidad Relación Extendido**) al **Modelo Lógico** (**Modelo Relacional**)

Reglas de Transformación  
Generales!!!



# Entidades y sus atributos

- Cada **entidad** del esquema conceptual se transforma en una **tabla**, llamadas **tablas base**
- Sus atributos siguen las siguientes reglas:
  - **Simple:** Se convierten en los atributos/columnas de la tabla
  - **Compuestos:** No se implementan como tales, sino que se implementa cada componente, y se convierte en un atributo/columna de la tabla
  - Los **atributos derivados** no se implementan (en principio!!!)
  - Los **atributos multivaluados** generan una tabla adicional (próxima página)
- Clave primaria:
  - La **clave primaria** de la tabla es la identificada para la entidad

# Entidades y sus atributos - Multivaluados

- Los atributos multivaluados **generan una tabla base**
- Descripción de la tabla generada:
  - Columnas/Atributos:
    - El atributo multivaluado
    - La clave primaria de la entidad
  - Clave Primaria:
    - La clave será (en general!!) la unión de ambos atributos

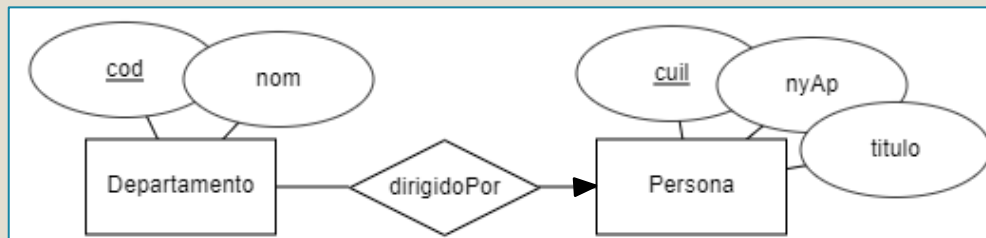
# Relaciones o Vinculaciones (n-1)

## Relaciones binarias n-1 (Idem para 1-n):

- **No generan tabla**
- A la tabla correspondiente a la **entidad del lado n** se le agrega la **clave primaria de la entidad del lado 1** de la relación
- Si la relación tiene atributos, esos atributos se colocan en la tabla a la que se le agregó la clave primaria (entidad del lado n)

### Ejemplo:

- *Un dpto. es dirigido por sólo una persona*
- *Una persona podría dirigir más de un depto.*



- **Departamento** = {cod, nom, cuil}
- **Persona** = {cuil, nyAp, titulo}

\* Si la relación dicta tuviese un atributo, por ejemplo fechaInicio:

- **Departamento** = {cod, nom, cuil, fechaInicio}

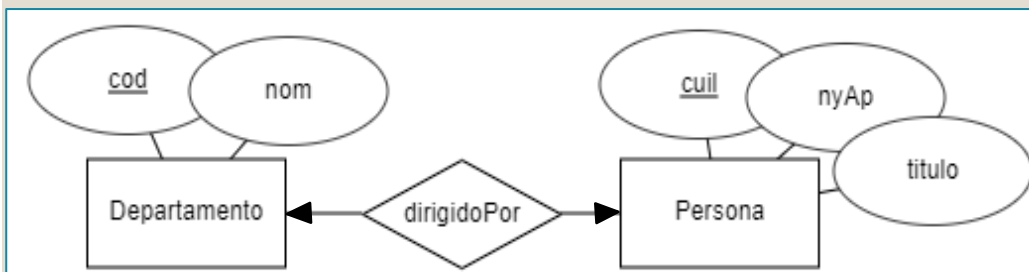
# Relaciones o Vinculaciones (1-1)

## Relaciones binarias 1- 1

- No generan tabla
- Se agrega como atributo, la **clave primaria en alguna de las dos** tablas de las entidades vinculadas
- Si la relación tiene atributos, se coloca en la tabla a la que se le agregó la clave primaria

### Ejemplo:

- *Un dpto. es dirigido por sólo una persona*
- *Una persona puede dirigir sólo un depto.*



Una alternativa:

- **Departamento** = {cod, nom, cuil}
- **Persona** = {cuil, nyAp, titulo}

Otra alternativa:

- **Departamento** = {cod, nom}
- **Persona** = {cuil, nyAp, titulo, cod}

**OBSERVACION:** Las 2 entidades vinculadas podría reunir las en una sola tabla



# Relaciones o Vinculaciones (m-n)

## Relaciones binarias m-n

- **Generan tabla**
- **Atributos:**
  - Las claves de las entidades vinculadas
  - Los atributos propios de la relación, si los tuviera
- **Clave primaria:**
  - **En caso de no tener atributos propios:** La unión de las dos claves de las entidades vinculadas
  - **En caso de tener atributos propios:** Idem anterior, pero además puede necesitar agregar algún/nos de los atributos de la relación

### Ejemplo:

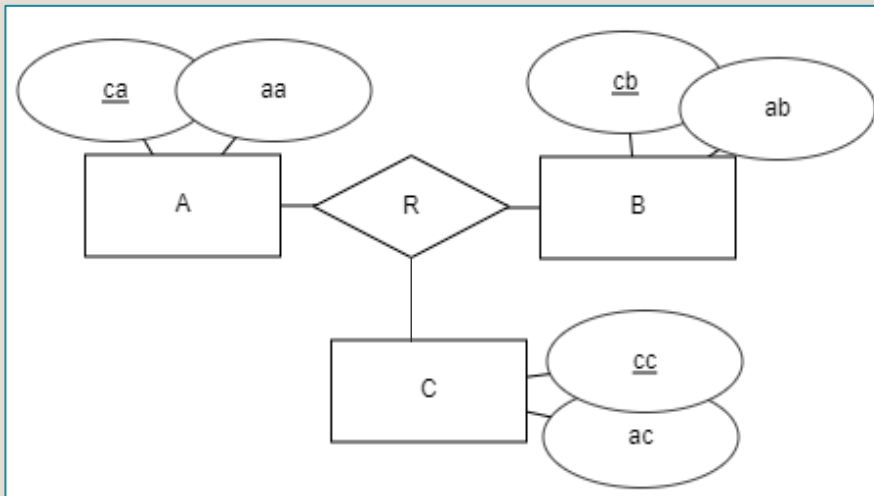
- *Un dpto. puede haber sido dirigido por más de una persona*
- *Una persona puede haber dirigido más de un depto.*



- **Departamento** = {cod, nom}
- **Persona** = {cuil, nyAp, titulo}
- **dirigidoPor** = {cod, cuil, fDesde, fHasta}

# Relaciones Ternarias

- **SIEMPRE generan una tabla**
- **Atributos**: La tabla contendrá las claves de las entidades vinculadas, **más los atributos propios** (si los tuviera) PERO, la **clave primaria de R dependerá de la multiplicidad** que presente la relación

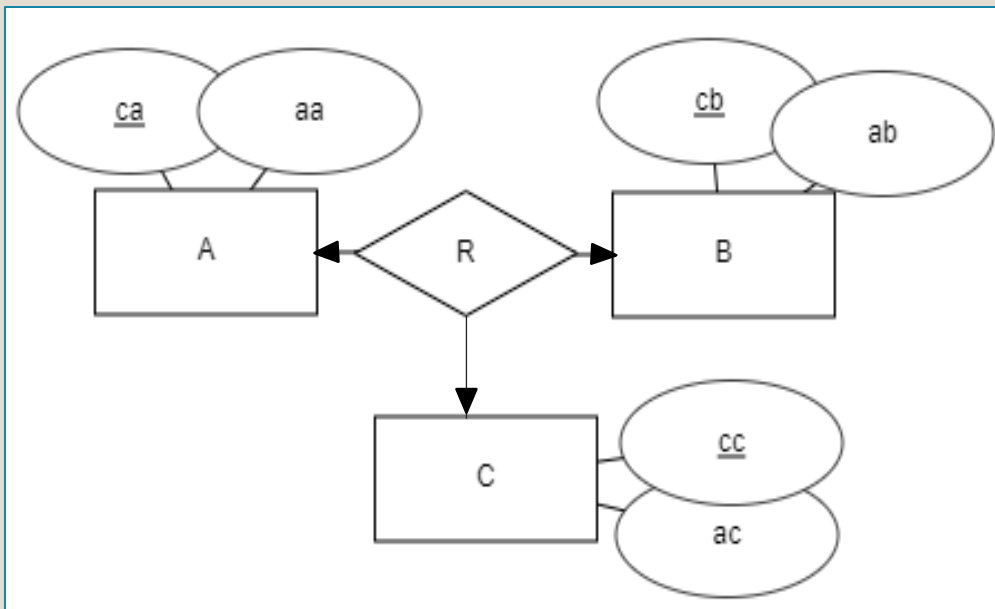


- $A = \{\underline{ca}, aa\}$
- $B = \{\underline{cb}, ab\}$
- $C = \{\underline{cc}, ac\}$
- $R = \{\underline{ca}, \underline{cb}, \underline{cc}\}$

# Relaciones Ternarias

- **Clave primaria:**

- Si la multiplicidad es **1-1-1**, será **ca** o **cb** o **cc**

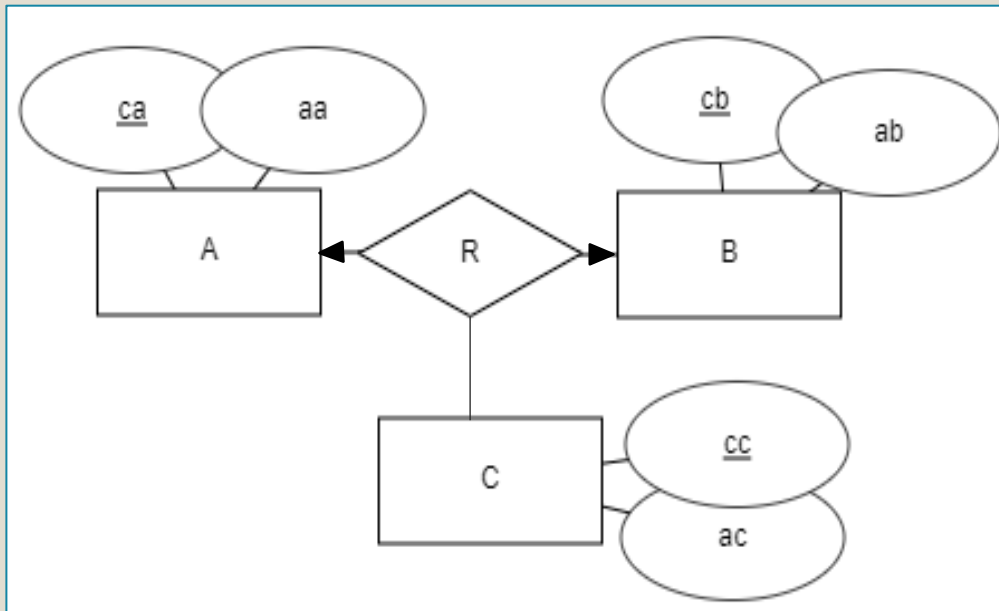


- $A = \{\underline{ca}, aa\}$
- $B = \{\underline{cb}, ab\}$
- $C = \{\underline{cc}, ac\}$
- $R = \{\underline{ca}, \underline{cb}, \underline{cc}\}$

# Relaciones Ternarias

- **Clave primaria:**

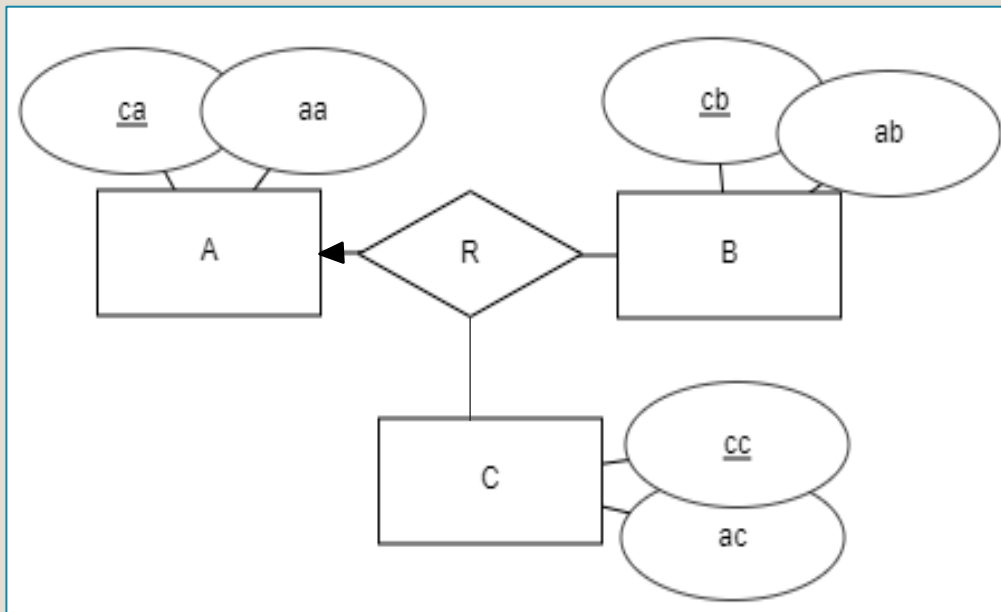
- Si la multiplicidad es **1-1-n**, será **ca+cc** o **cb+cc**



- $A = \{\underline{ca}, aa\}$
- $B = \{\underline{cb}, ab\}$
- $C = \{\underline{cc}, ac\}$
- $R = \{\underline{ca}, \underline{cb}, \underline{cc}\}$

# Relaciones Ternarias

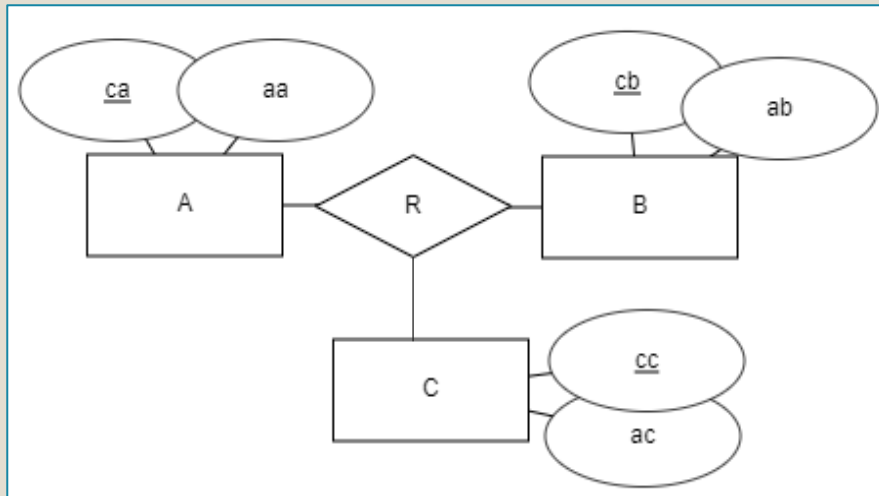
- **Clave primaria:**
  - Si la multiplicidad es **1-n-m**, será **cb+cc**



- $A = \{\underline{ca}, aa\}$
- $B = \{\underline{cb}, ab\}$
- $C = \{\underline{cc}, ac\}$
- $R = \{\underline{ca}, \underline{cb}, \underline{cc}\}$

# Relaciones Ternarias

- **Clave primaria:**
  - Si la multiplicidad es **n-m-p**, será **ca+cb+cc**



- $A = \{\underline{ca}, aa\}$
- $B = \{\underline{cb}, ab\}$
- $C = \{\underline{cc}, ac\}$
- $R = \{\underline{ca}, \underline{ca}, \underline{cc}\}$

# Entidades vinculadas con una Relación Generalización - Especialización

- Se pueden implementar alguna de las siguientes **opciones**, según corresponda:

## 1. Una tabla por cada entidad (superclase y subclases):

- Válida para **cualquier tipo de jerarquía**

## 2. Una tabla por cada subclase:

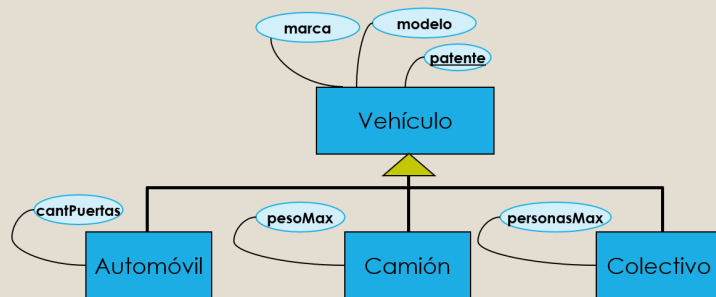
- Válida para **jerarquías totales y exclusivas**

## 3. Integrar todas las entidades en una sola tabla:

- Válida para **cualquier tipo de jerarquía**, pero:
  - Puede genera muchos valores nulos
  - Se debe agregar un atributo que indique el tipo de entidad.

# Entidades vinculadas con una Relación

## Generalización - Especialización



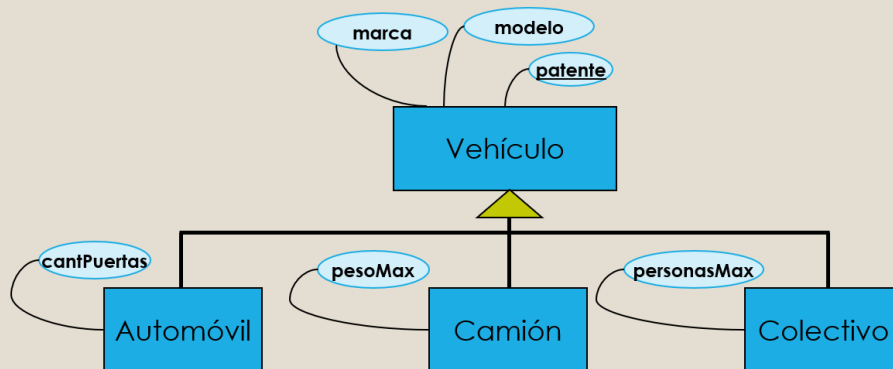
**Opción 1:** Es la mas general.  
**Válida para cualquier tipo de jerarquía**

- **Tabla de la superclase:** La tabla contiene **todas las instancias**.
  - **Atributos:** Todos los atributos de la superclase
  - **Clave primaria:** El atributo clave de la superclase
- **Tablas de cada subclase:** Contienen **sólo las instancias de la subclase correspondiente**.
  - **Atributos:**
    - El atributo clave de la superclase
    - Todos los atributos de la subclase
  - **Clave primaria:** El atributo clave de la superclase



# Entidades vinculadas con una Relación

## Generalización - Especialización

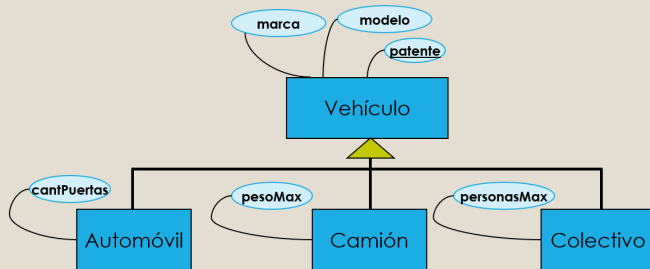


**Opción 2:** Válida para jerarquías totales (completas) y exclusivas.

- **No se genera tabla para la superclase**
- **Se genera una tabla por cada subclase:** Contiene las instancias de la subclase correspondiente
- **Atributos:**
  - Todos los atributos de la superclase
  - Todos los atributos de la subclase
- **Clave primaria:**
  - El atributo clave de la superclase

# Entidades vinculadas con una Relación

## Generalización - Especialización



**Opción 3:** Integrar **todas las entidades en una sola tabla**. **Válida para cualquier tipo de jerarquía** (pero provoca inconvenientes...)

La tabla contendrá las el conjunto de todas las instancias.

▪ **Atributos:**

- Todos los atributos de la superclase
- Todos los atributos de cada subclase
- Un atributo que indique el tipo correspondiente a la subclase a la que pertenece

▪ **Clave primaria:**

- El atributo clave de la superclase

**OBSERVACIONES:**

- Genera Nulos (no es deseable)
- Se debe agregar un atributo que indique el tipo de entidad

# Entidades Débiles

Al igual que una entidad (fuerte) se transforma en **una tabla base**

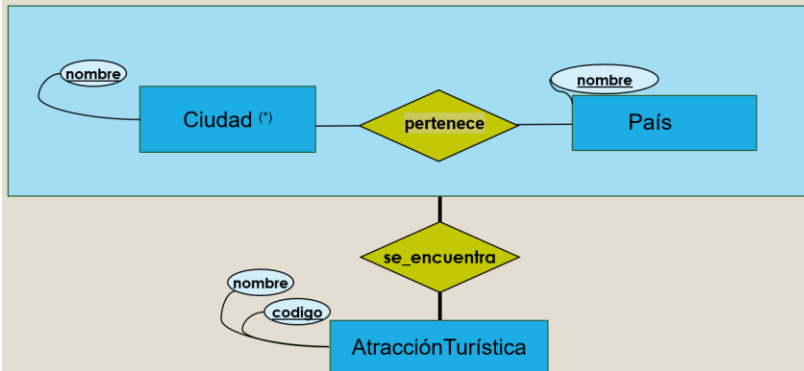
- **Atributos:**

- Los atributos (simples) de la entidad se convierten en los atributos de la tabla.
- Se agrega como atributo, la clave de la entidad fuerte a la que esta vinculada.

- **Clave primaria:**

- La **clave primaria de la entidad fuerte** a la que se vincula
- **+**
- La **clave parcial** de la tabla correspondiente a la **entidad débil** (discriminador)

# Agregación



**Nota:** Estamos considerando que la relación agregada es binaria, y la otra relación, que vincula la agregación con otra entidad, también es binaria

- La relación m-n agregada genera tabla tal como se mencionó en el caso de cualquier relación m-n
- La relación vinculada a la agregación se implementa según su multiplicidad (1-1, 1-n, m-n)
  - Tener cuidado con la clave primaria!!! Justamente en el caso de tener atributos que formen parte de la clave en la agregación

***Y si la relación agregada fuese ternaria, ¿cómo se implementaría?***

# Diseño Lógico: Reglas de Conversión

---

## IMPORTANTE TENER EN CUENTA!

- Estas reglas son generales
- Debieran ser aplicadas con **criterio crítico** a la luz del **uso de los datos (consultas, etc.)** y a la **evolución del minimundo** que se está modelando

# Diseño Físico (totalmente dependiente del SGBD)

- Debe realizarse en base a las **formas de almacenamiento y técnicas de acceso provistas por el SGBD específico** en el que se implemente la base de datos (como muestra el gráfico)
- Por ello, no se aborda en la misma forma que las otras dos etapas de diseño anteriores. Sólo consideraremos generalidades de las estructuras internas (Unidad de Nivel Interno).



# Proceso de Construcción de una base de datos

---

¿Listos para continuar con la práctica?

