



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

**DEPARTAMENTO DE INFORMÁTICA ADMINISTRATIVA**

**EVALUACIÓN DE SISTEMAS - IA220**

**CATEDRÁTICA: KARLA MELISA GARCÍA**

Base de Datos 1  
Introduccion

# TABLA DE CONTENIDO

1. Aplicaciones de sistemas de Bases de datos
2. Sistema de Base de datos versus Sistema de Archivos
3. Principales proveedores de bases de datos
4. [Concepto de Base de datos](#)
5. [Sistema Gestor de Base de datos](#)
6. Estructura de un SGBD
7. Personas que interactúan con la Base de datos
  1. Desarrolladores
  2. DBA
  3. Usuarios Finales
8. Modelado de Datos
9. Modelo de datos
10. Tipos de modelos

9. Modelo E-R
  1. Entidades
  2. Atributos
    1. Tipos de Atributos
  3. Relaciones
    1. Participación
    2. Papel
    3. Ejemplar
  4. Restricciones
    1. Correspondencia de cardinalidades
    2. Restricciones de participación
    3. Tipos de cardinalidades
  5. Llaves o claves
    1. Super-clave
    2. Claves candidatas
    3. Claves primarias
10. Modelo Relacional
11. Diagrama Entidad-Relación crow's foot
  1. Entidades
  2. Atributos
  3. Relaciones
  4. Cardinalidades
12. Ejemplo de Diagrama Entidad-Relación UML.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Introducción

- En esencia, un sistema de base de datos no es más que un sistema para archivar en un computador.
- La base de datos en sí puede considerarse como una especie de archivero electrónico; dicho de otra manera es un lugar donde se almacenan un conjunto de archivos de datos computarizados.
- Al usuario del sistema se le brindarán recursos para realizar diversas operaciones sobre estos archivos, incluidas entre otras las siguientes:
  - Agregar archivos nuevos (vacíos) a la base de datos.
  - Insertar datos nuevos en archivos ya existentes.
  - Obtener datos de archivos ya existentes.
  - Actualizar datos en archivos ya existentes.
  - Borrar datos en archivos ya existentes
  - Eliminar archivos ya existentes (vacíos o no) de la base de datos.

# SGBD FRENTE A SISTEMAS DE ARCHIVOS

- Redundancia e Inconsistencia de Datos: Debido a que los archivos y programas de aplicación son creados por diferentes programadores en un largo período de tiempo, los diversos archivos tienen probablemente diferentes formatos y los programas pueden estar escritos en diferentes lenguajes. Más aún la información puede estar duplicada en diferentes lugares (archivos).
- Dificultad en el acceso a los datos: La cuestión aquí es que el entorno de procesamiento de archivos convencional no permite que los datos necesarios sean obtenidos de una forma práctica y eficiente.
- Aislamiento de Datos: Debido a que los datos están dispersos en varios archivos, y los archivos pueden estar en diferentes formatos, es difícil escribir nuevos programas de aplicación para recuperar los datos apropiados.
- Problemas de Integridad: Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de **restricciones de consistencia**.
- Problemas de Atomicidad: Un computador, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallo. En muchas aplicaciones es crucial asegurar que, una vez que un fallo ha ocurrido y se ha detectado, los datos se restauran al estado de consistencia que existía antes del fallo.
- Anomalías en el acceso concurrente: Conforme se ha ido mejorando el conjunto de ejecución de los sistemas y ha sido posible una respuesta en tiempo más rápida, muchos sistemas han ido permitiendo a múltiples usuarios actualizar los datos simultáneamente. En tales sistemas un entorno de interacción de actualizaciones concurrentes puede dar lugar a datos inconsistentes.
- Problemas de Seguridad: No todos los usuarios de un sistema de bases de datos deberían poder acceder a todos los datos.



# Historia

- **Antes de 1960**

- 1945 Invención de la cinta magnética (primer medio que permite búsquedas)
- 1957 Instalación del primer computador comercial este almacena datos electrónicamente.

- **Los años 60**

- IBM desarrolla su Sistema de gestión de información (Information Management System, IMS),
- IBM y American Airlines crean SABRE.

- **Los años 80**

- 1985 se publica la norma preliminar de SQL.

- **Los años 90**

- Aparición de SGBD comerciales orientados a objetos.



# Concepto de Base de Datos

Una Base de Datos es un conjunto de datos relacionados entre sí.

Una base de datos es un conjunto estructurado de datos coherentes, donde no hay pérdida de información.

Este es un concepto muy general, el término Base de Datos es más restringido y cumple las siguientes propiedades implícitas:

Una base de datos representa algún aspecto del mundo real, llamado mini mundo.

Una base de datos es un conjunto de datos lógicamente coherente. Una colección aleatoria de datos no sería una base de datos.

Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

**(Silberschatz-Korth-Sudarshan)**



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Estructuras

- Una base de datos, es una colección de tablas en las que se almacena un conjunto específico de datos estructurado.
- Una tabla contiene una colección de *filas*, también denominada tuplas o *registros*.
- *Columnas*, también llamadas atributos. Cada columna de la tabla se ha diseñado para almacenar un determinado tipo de información, ej: fecha, nombres, importes en moneda, o números

Una base de datos puede ser de cualquier tamaño y complejidad.





# BD no estructurados

No estructurado significa simplemente que se trata de conjuntos de datos (colecciones grandes típicas de archivos) que no se almacenan en un formato de base de datos estructurado. Los datos no estructurados tienen estructura interna, pero no están predefinidos por modelos de datos.

¿Qué son los datos no estructurados ejemplos?

Estos son algunos ejemplos de datos no estructurados: datos de redes sociales, datos de vigilancia, datos geospaciales, audios, datos meteorológicos... Pero también informes, facturas, registros, correos electrónicos, aplicaciones de productividad, etc

¿Qué base de datos usa Big Data?

- En el Big Data, las bases de datos más utilizadas son las NoSQL por diversas razones: Elevado número de fuentes de datos: internet, IoT, estudios, etc. Distintos tipos de datos: estructurados (tablas), no estructurados (documentos, vídeos...), semiestructurados, etc.





**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proveedores

¿Qué es un proveedor de base de datos?

- Los proveedores de datos representan diversos orígenes de datos como bases de datos SQL, archivos secuenciales indizados, hojas de cálculo, almacenes de documentos y archivos de correo. Los proveedores exponen datos uniformemente mediante una abstracción común que se denomina conjunto de filas.

¿Cuáles son las principales bases de datos o mas bien gestores de base de datos?

- Oracle, DB2 o SQL Server son algunas de las bases de datos más utilizadas hoy en día. Los tipos de bases de datos relacionales como Oracle, DB2 de IBM y SQL Server e incluso Microsoft Access, forman la columna vertebral para el almacenamiento de datos y la gestión en la mayoría de organizaciones de hoy en día.

# Sistema Gestor de Base de Datos

*SGBD, DBMS (Database Management System)*

EL **sistema gestor de bases de datos** (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos.

El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto *práctica* como *eficiente*.

## Principales Funciones

- Gestionar grandes cantidades de información : Implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información.
- Deben proporcionar la fiabilidad de la información almacenada.
- Debe evitar posibles resultados anómalos.

***Silberschatz-Korth-Sudarshan***

# Sistema Gestor de Base de Datos

El DBMS es un *sistema de software de propósito general* que facilita los procesos de ***definición, construcción, manipulación y compartición*** de bases de datos entre varios usuarios y aplicaciones.

- **Definir** una base de datos implica especificar los tipos de datos, estructuras y restricciones de los datos que se almacenarán en la base de datos. La definición o información descriptiva de una base de datos también se almacena en esta última en forma de catálogo o diccionario de la base de datos; es lo que se conoce como metadatos.
- **La construcción** de la base de datos es el proceso consistente en almacenar los datos en algún medio de almacenamiento controlado por el DBMS.
- **La manipulación** de una base de datos incluye funciones como la consulta de la base de datos para recuperar datos específicos, actualizar la base de datos para reflejar los cambios introducidos en el minimundo y generar informes a partir de los datos.
- **Compartir** una base de datos permite que varios usuarios y programas accedan a la base de datos de forma simultánea.

# Estructura Gestor de Base de Datos

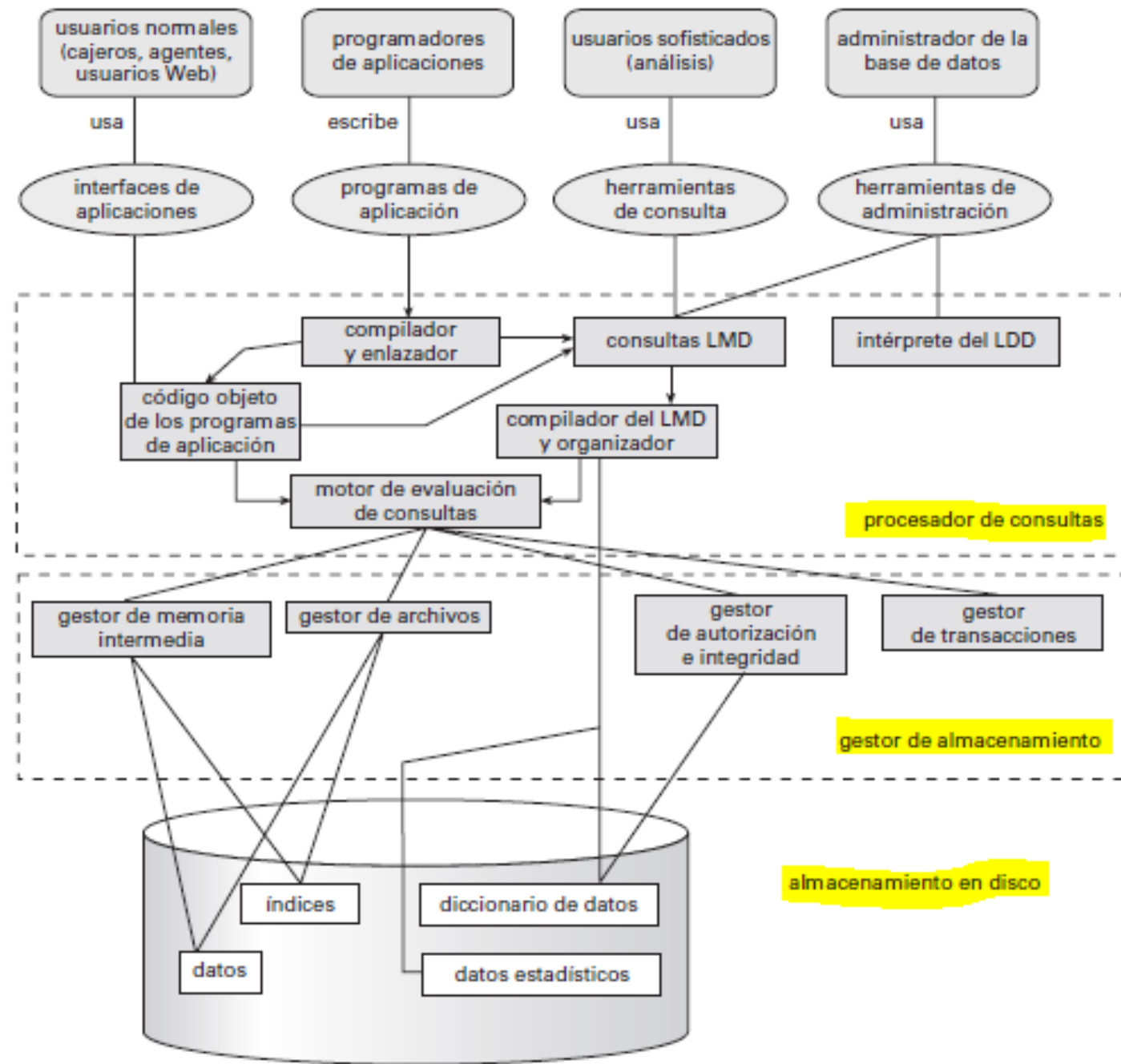
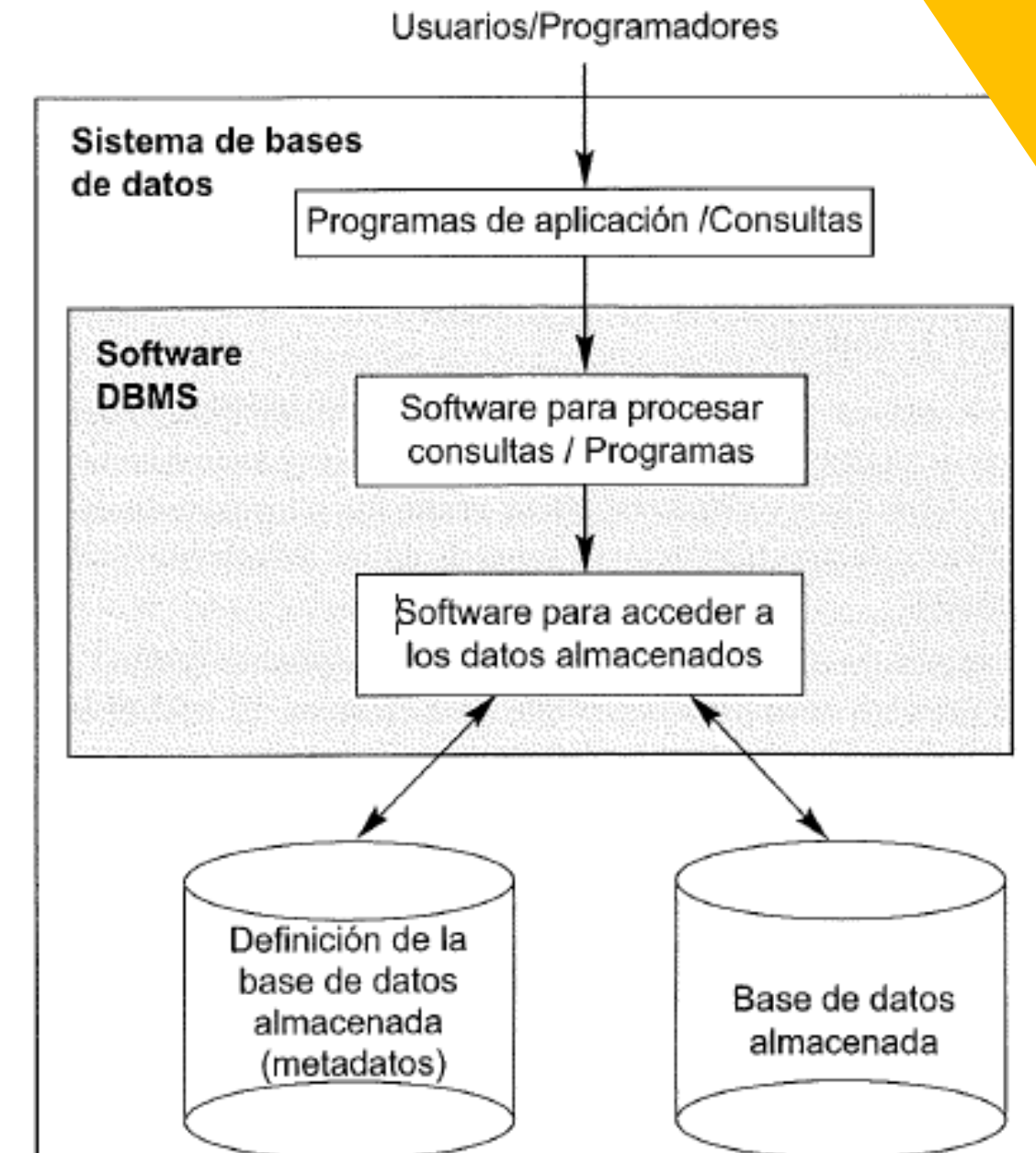


FIGURA 1.4. Estructura del sistema.

Figura 1.1. Entorno de un sistema de bases de datos simplificado.

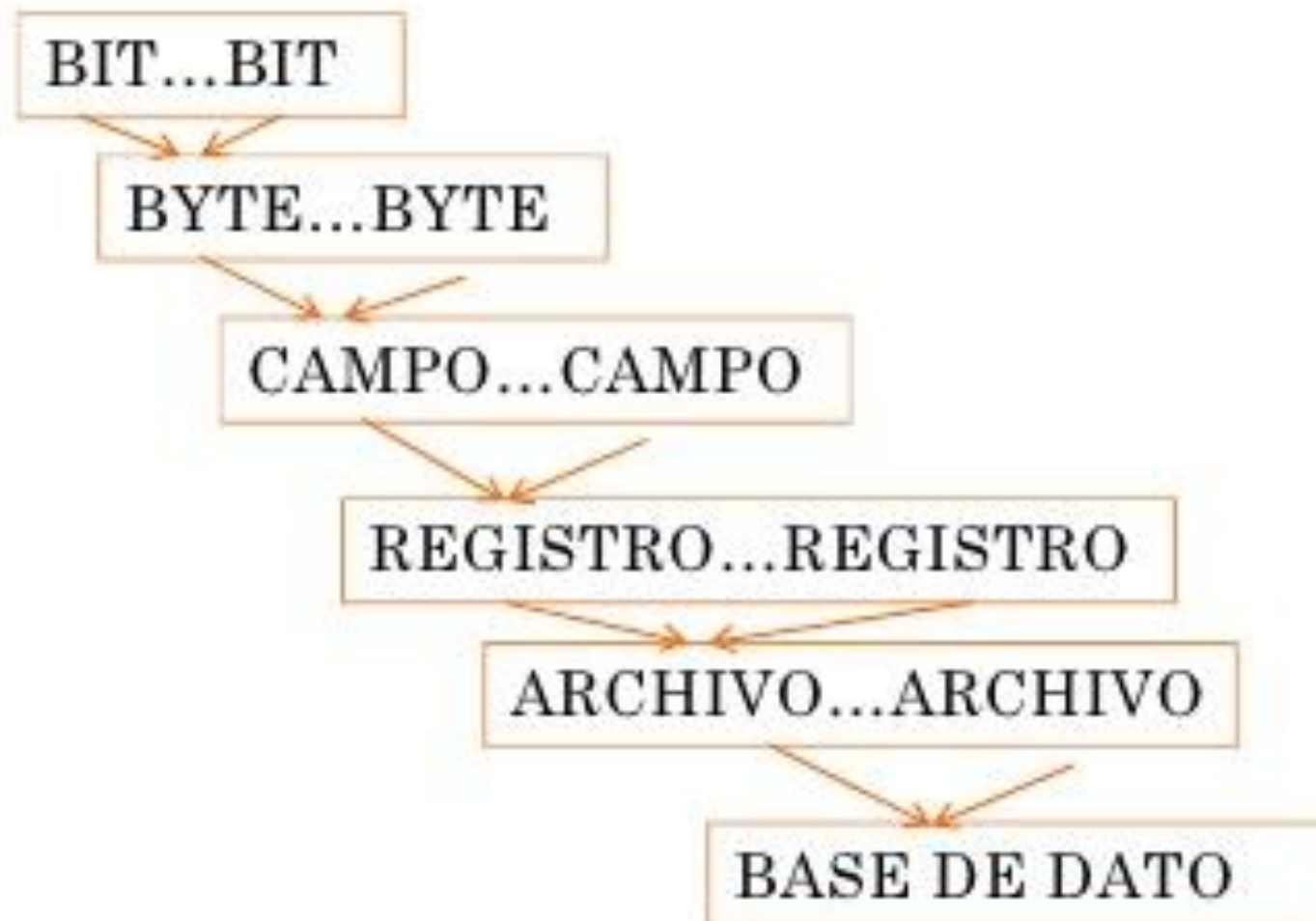






# CONFORMACION DE UNA BASE DE DATOS:

- Una base de dato esta compuesta por estructuras lógicas de almacenamiento, denominadas TABLAS.
- Una tabla esta compuesta por columnas o campos y filas o registros.
- Un registro es la ocurrencia de conjunto de campos de una tabla.



# Modelo ACID

## ACID en bases de datos

Las cuatro propiedades pertenecientes al modelo ACID en bases de datos pueden detallarse de la siguiente forma:

### *Atomicity / Atomicidad*

El modelo ACID permite que se lleven a cabo las operaciones atómicas. También es **conocida como el «todo o nada» de la transacción**, debido a que, en el caso de que se completen todos los pasos de la transacción, se obtendrán las modificaciones requeridas en la base de datos. Si una parte de la transacción falla, el sistema debe ser capaz de hacer que el resto de operaciones fallen, por lo que la base de datos no sufrirá ningún cambio indeseado.

### *Consistency / Consistencia*

Hace referencia a la **capacidad que tiene un sistema para iniciar solo las operaciones que puede concluir**. Solo se pueden ejecutar pasos de la transacción que no incumplan con las reglas de integridad definidas, incluyendo los *triggers*, *cascades* y *constraints*, así como sus combinaciones.



# Modelo ACID

## *Isolation / Aislamiento*

En bases de datos se refiere a la manera y el momento en el que los cambios resultantes de una operación se harán visible para las demás operaciones concurrentes. Es decir, la realización de una operación no debería afectar a las otras, debido a que **cada una de las transacciones debe ser ejecutada en aislamiento total**, sin importar si se llevan a cabo de manera simultánea.

## *Durability / Durabilidad*

La durabilidad de **ACID** hace referencia a la propiedad que garantiza que, una vez se haya llevado a cabo una determinada operación, estas tengan la **capacidad de persistir y no puedan ser deshechas incluso si el sistema falla**. Implica que los datos y cambios en una transacción que ya se ha realizado deben ser permanentes y no puede ocurrir una pérdida de los mismos en el sistema.

# base de datos en la organizaciones

Un sistema de base de datos en una organización grande está formado por:

- El hardware
- El software
- Los datos
- Las personas

# El hardware comprende

- uno o más computadores, servidores, terminales
- unidades de disco, memoria, unidades de cinta magnética,
- terminales, conexiones de red y otros dispositivos físicos.

## El software incluye

- un sistema de gestión de bases de datos (SGBD)
- Los programas de aplicación utilizan el SGBD para tener acceso y manipular la base de datos

## Los datos

- Representan los hechos importantes para la organización, radican físicamente en el disco, pero se estructuran lógicamente de forma que se logre un acceso fácil y eficiente.

# Las personas

Que hacen uso del sistema, es decir los usuarios.

**Usuarios:** Se toman en cuenta tres clases de usuarios:

- *Programador de aplicaciones, BI, Científico..etc:* quien se encarga de escribir programas de aplicación que utilizan la base de datos.
- *Usuario final:* quien interactúa con el sistema desde una terminal en línea. Un usuario final puede tener acceso a la base de datos utilizando una interfaz incluida como parte integral de los programas del sistema de la base de datos, estas interfaces también trabajan mediante aplicaciones en línea.
- *Administrador de base de datos ó DBA (database administrator):* su función es decidir en primer término cuales datos deben almacenarse en la base de datos y establecer políticas para mantener y manejar los datos una vez almacenados. El DBA es un gerente, no un técnico, el DBA garantiza el funcionamiento adecuado del sistema

# Cargos vinculados a un SGBD

- ✓ **Administrador de bases de datos**
- ✓ **Diseñador / Arquitectos de bases de datos**
- ✓ **Usuarios finales**
- ✓ **Analistas de sistemas y programadores**
- ✓ **Diseñadores e implementadores del SGBD**
- ✓ **Creadores de herramientas**
- ✓ **Operadores y personal de mantenimiento**
- ✓ **BI**
- ✓ **Científicos de Datos O analistas de Datos**

# Cargos y Funciones vinculados a un SGBB

- ✓ **Administrador de bases de datos** : Se encarga de autorizar el acceso a la base de datos, de coordinar y vigilar su empleo, y de adquirir los recursos necesarios de software y hardware.
- ✓ **Diseñador de bases de datos** : Antes de implementar la base de datos identifica los datos que se almacenarán y elige las estructuras apropiadas para representar y almacenar dichos datos.
- ✓ **Usuarios finales** : Son los principales destinatarios de la base de datos.  
  
Son quienes tiene acceso a la base de datos para consultarla, actualizarla y generar informes. Este es el rol que más frecuentemente desempeña el **Técnico en Administración** con relación de las Bases de Datos.
- ✓ **Operadores y personal de mantenimiento**: Tienen a su cargo el funcionamiento y mantenimiento reales del entorno de hardware y software del sistema de base de datos.

# Cargos y funciones vinculados a un SGBB

## Analistas de sistemas y programadores, BI, Científicos

Los analistas y programadores para implementar las especificaciones de un sistema deben conocer a la perfección todas las capacidades del SGBD. El **Técnico en Administración** que integra un equipo de desarrollo interactúa con los analistas y programadores.

## ✓ Diseñadores e implementadores del SGBD

Se encargan de diseñar e implementar los módulos e interfaces del SGBD en forma de paquetes de software.

## ✓ Creadores de herramientas


Desarrollan e implementar paquetes para diseñar bases de datos, vigilar el rendimiento, proporcionar interfaces de lenguaje natural o de gráficos, elaborar prototipos, realizar simulaciones y generar datos de prueba.




# Justificación del uso del SGBD


- **Potencial para imponer normas:** esto facilita la comunicación y cooperación entre diversos departamentos, proyectos y usuarios de una organización.
- **Menor tiempo de creación de aplicaciones:** una vez que esté lista la base de datos, se requerirá mucho menos tiempo para crear nuevas aplicaciones con los recursos del Sistema de Gestión de Base de Datos.
- **Flexibilidad:** en ocasiones es necesario modificar la estructura de una base de datos, ; algunos Sistemas de Gestión de Base de Datos permiten efectuar estas modificaciones sin afectar los datos almacenados y los programas de aplicación existentes.
- **Disponibilidad de información actualizada:** los Sistemas de Gestión de Base de Datos ponen la base de datos a disposición de todos los usuarios y cuando uno de ellos la actualiza el resto puede ver de inmediato esta actualización.
- **Economías de escala:** el Sistema de Gestión de Base de Datos permite consolidar datos y las aplicaciones reduciéndose así el desperdicio por duplicación de datos.
- **Respaldo y recuperación**

# APLICACIONES DE LOS SGBD


 **BANCA:** Para información de los clientes, cuentas y préstamos, y transacciones.


 **LÍNEAS AÉREAS:** Para reservas e información de planificación.


 **UNIVERSIDADES:** Para información de los estudiantes, matriculas de las asignaturas y cursos.


 **TRANSACCIONES DE TARJETAS DE CRÉDITO:** Para compras con TC y generación mensual de extractos.

 **RECURSOS HUMANOS:** para información sobre los empleados, salarios, impuestos y beneficios, y para la generación de nominas.

 **TELECOMUNICACIONES:** Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas prepago y para almacenar información sobre redes de comunicaciones.

 **FINANZAS:** Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.

 **VENTAS:** Para información de clientes productos y compras.

 **PRODUCCIÓN:** Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.

## DESVENTAJAS

- Coste
- Hardware requerido
- Complejidad

# Niveles de Observacion de una base de datos

- En cualquier sistema de información se considera que se pueden observar los datos desde dos puntos de vista:
- **Nivel externo.** Esta es la visión de los datos que poseen los usuarios del Sistema de Información.
- **Nivel físico.** Esta es la forma en la que realmente están almacenados los datos.
- En el caso de los Sistemas de Base de datos, se añade un tercer nivel, un tercer punto de vista, es el **nivel conceptual**. Ese nivel se sitúa entre el físico y el externo.
- En cada nivel se manejan esquemas de la base de datos.

# Esquemas de la base de datos.

- **Esquema físico** : Representa la forma en la que están almacenados los datos. Esta visión sólo la requiere el **administrador/a**. **El administrador la necesita para poder gestionar más eficientemente la base de datos.**
- En este esquema se habla de archivos, directorios o carpetas, unidades de disco, servidores,...
- **Esquema conceptual** : Se trata de un esquema teórico de los datos en el que figuran organizados en estructuras reconocibles del mundo real y en el que también aparece la forma de relacionarse los datos. Este esquema es el paso que permite modelar un problema real a su forma correspondiente en el ordenador.
- Este esquema es la base de datos de todos los demás. Como se verá más adelante, es el primer paso a realizar al crear una base de datos. En definitiva es el plano o modelo general de la base de datos.
- El esquema conceptual lo realiza **diseñadores/as o analistas**.

- **Esquema externo** : En realidad son varios. Se trata de la visión de los datos que poseen los usuarios y usuarias finales. Esa visión es la que obtienen a través de las aplicaciones. Las aplicaciones creadas por los desarrolladores abstraen la realidad conceptual de modo que el usuario no conoce las relaciones entre los datos, como tampoco conoce dónde realmente se están almacenando los datos.
- Los esquemas externos los realizan las **programadoras/es según las indicaciones formales de los y las analistas**.
- Realmente cada aplicación produce un esquema externo diferente (aunque algunos pueden coincidir) o **vista de usuario**. El conjunto de todas las vistas de usuario es lo que se denomina **esquema externo global**.

# Funciones de los lenguajes en los SGBD

- Los SGBD tienen que realizar tres tipos de funciones para ser considerados válidos.
  - Función de Descripción o Definición :
  - Función de Manipulación
  - Función de Control



## Función de Descripción o Definición :

- Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos.
- Se dice que esta función es la que permite definir las tres estructuras de la base de datos (relacionadas con los tres niveles de abstracción).
  - Estructura interna
  - Estructura conceptual
  - Estructura externa
- Realmente esta función trabaja con metadatos.

Los metadatos es la información de la base de datos que realmente sirve para describir a los datos.

*Es decir, **Sánchez Rodríguez y Crespo son datos; pero Primer Apellido es un metadato.***

***También son datos decir que la base de datos contiene Alumnos o que el id lo forman 9 caracteres de los cuales los 8 primeros son números y el noveno un carácter en mayúsculas.***

- La función de definición sirve pues para **crear, eliminar o modificar metadatos. Para ello permite usar un lenguaje de descripción de datos o DDL. Mediante ese lenguaje:**
  - Se definen las estructuras de datos
  - Se definen las relaciones entre los datos
  - Se definen las reglas que han de cumplir **los datos**

Función de Manipulación: Permite modificar y utilizar los **datos de la base de datos**. Se realiza mediante un lenguaje de modificación de datos o DML. Mediante ese lenguaje se puede:

- Añadir datos
  - Eliminar datos
  - Modificar datos
  - Buscar datos
- 
- Actualmente se suele distinguir aparte la función de buscar datos en la base de datos (**función de consulta**). Para lo cual se proporciona un lenguaje de **consulta de datos o DQL**.

## Función de control

- Mediante esta función los administradores poseen mecanismos para proteger los datos; de modo que se permite a cada usuario ver ciertos datos y otros no; o bien usar ciertos recursos concretos de la base de datos y prohibir otros.
- Es decir simplemente permite controlar la seguridad de la base de datos. El lenguaje que implementa esta función es el **lenguaje de control de datos o DCL**.



# Modelo de Datos

Un **modelo de datos** es una colección de herramientas **conceptuales** para la descripción de datos, relaciones de datos, semántica de los datos y restricciones de

- Modelo entidad-relación
- Modelo relacional

*Silberschatz-Korth-Sudarshan*

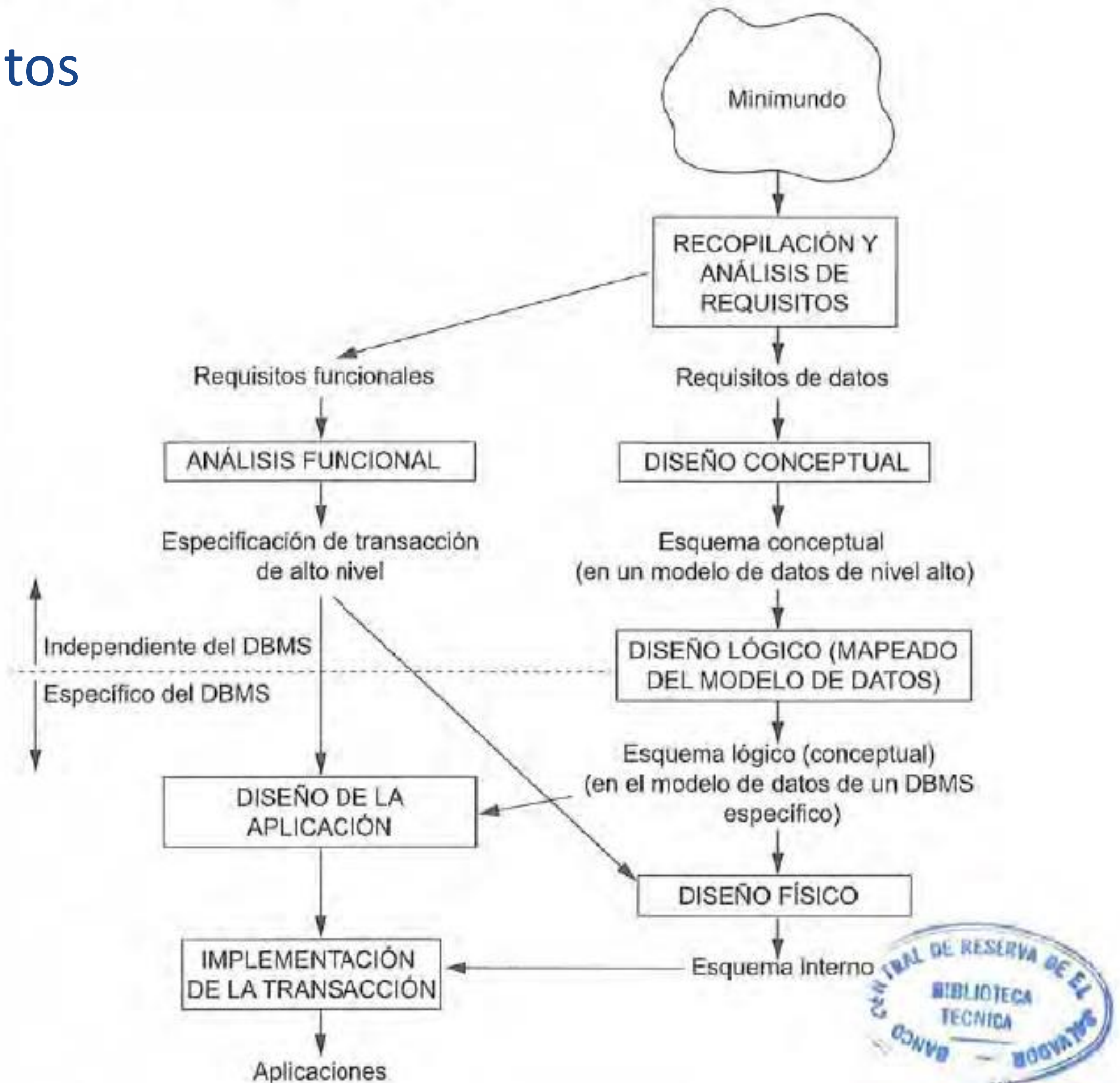
*Diferencias:*

<https://www.youtube.com/watch?v=ePa3f0kc>

*Ejemplos*

<https://www.calameo.com/read/0009814479>

Digamos que el modelo ER se hace previamente a realizar un 'boceto' de nuestra base de datos. Posteriormente, con ese diseño lo implementamos en un modelo Relacional (es decir, basado en SQL). Evidentemente, como uno es consecuencia del otro, tienen sutiles diferencias que hay que entenderlos correctamente.

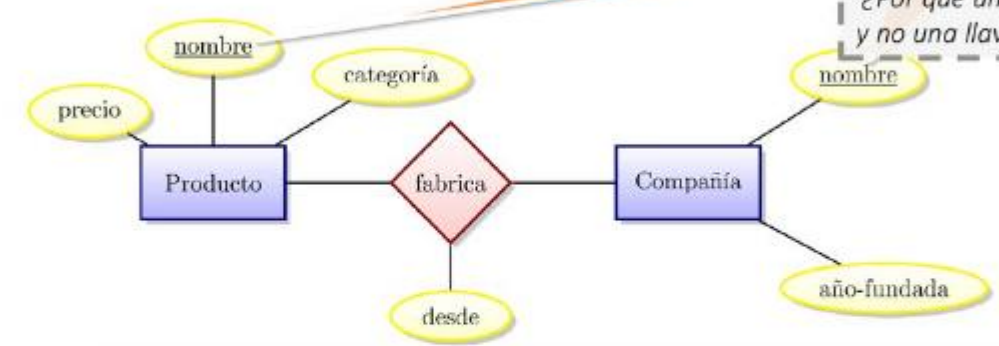




|             | MODELO ER  | MODELO RELACIONAL   |
|-------------|--|---|
| Labor       | Representa la colección de objetos llamada entidades y la relación entre esas entidades.                           | Representa la colección de tablas y la relación entre esas tablas.                          |
| Describir   | El Modelo de relación de entidad describe los datos como conjunto de entidades, conjunto de relaciones y atributo. | El Modelo Relacional describe los datos en una tabla como Dominio, Atributos, Tuplas.       |
| Relación    | ER Model es más fácil de entender la relación entre las entidades.   | Comparativamente, es menos fácil derivar una relación entre tablas en el Modelo Relacional. |
| Cartografía | El modelo de ER describe la asignación de cardinalidades.  | El Modelo Relacional no describe las cardinalidades de mapeo.                               |



Modelo E-R: Relación (con atributos)  
 → Modelo Relacional: Tabla



Las llaves de las entidades *juntas* forman una súper llave para la relación  
 ¿Por qué una súper llave y no una llave candidata?

**Producto**(nombre:string,precio:int,categoría:string)  
**Compañía**(nombre:string,año-fundada:int)  
**Fabrica**(p-nombre:string,c-nombre:string,desde:date)

| Producto                |          |           |
|-------------------------|----------|-----------|
| <u>nombre</u>           | cantidad | categoría |
| Tarapacá Carménère 2014 | 4000     | Vino      |
| Austral Calafate 330ml  | 2000     | Cerveza   |
| Austral Yagar 330ml     | 2200     | Cerveza   |
| Pall Mall Rojo 20       | 2500     | Tabaco    |

| Fabrica                |                          |       |
|------------------------|--------------------------|-------|
| <u>p-nombre</u>        | <u>c-nombre</u>          | desde |
| Austral Calafate 300ml | Cervecería Austral       | 1983  |
| Austral Yagar 300ml    | Cervecería Austral       | 2006  |
| Pall Mall Rojo 20      | British American Tobacco | 1907  |

| Compañía                 |             |
|--------------------------|-------------|
| <u>nombre</u>            | año-fundada |
| British American Tobacco | 1902        |
| Viña Tarapacá            | 1874        |
| Cervecería Austral       | 1896        |

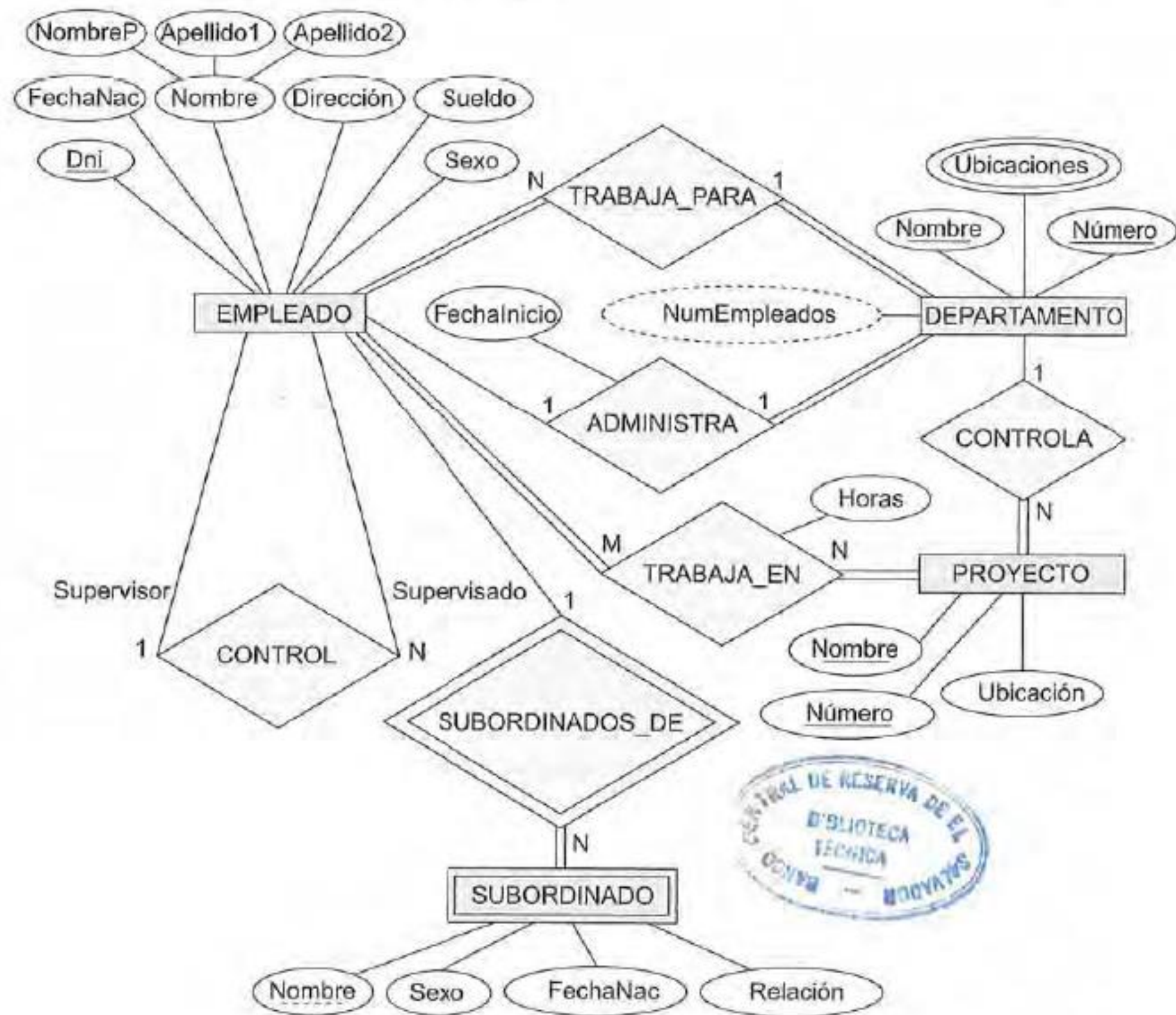


# Modelo Entidad-Relacion

EL modelo de datos **entidad-relación (E-R)** está basado en una percepción del mundo real consistente en objetos básicos llamados *entidades* y de *relaciones* entre estos objetos.

El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual.

**Silberschatz-Korth-Sudarshan**





# Modelo Entidad-Relacion

## Entidad

Hay tres nociones básicas que emplea el modelo de datos E-R: conjuntos de entidades, relaciones y atributos.

Una **entidad** es una «cosa» u «objeto» en el mundo real que es distinguible de todos los demás objetos.

**Entidad:** El objeto básico representado por el modelo ER es una entidad, que es una *cosa* del mundo real con una existencia independiente.

Una entidad puede ser un objeto con una existencia física (por ejemplo, una persona en particular, un coche, una casa o un empleado) o puede ser un objeto con una existencia conceptual (por ejemplo, una empresa, un trabajo o un curso universitario).

**Las entidades es TODO aquello donde podemos guardar información.**

# Modelo Entidad-Relacion

## TIPOS DE ENTIDAD BASE DE DATOS

Según el tipo de información obtenida:

**Entidad tangible:** Son todas las existencia físicas del mundo real.

**Ejemplos de entidades tangibles**

- Vehículos
- Personas
- Productos

**Entidades intangibles:**

Son las existencia que no tienen presencia física en el mundo real, solo lógica.

**Ejemplos de entidades intangibles**

- Una cuenta bancaria
- Sistema de materias y notas de una institución
- Cuentas en redes sociales

# Modelo Entidad-Relacion

## TIPOS DE ENTIDAD BASE DE DATOS

Según la conformación en el modelo entidad relación: (Estas son las importantes)

### Entidades fuertes:

Para comprender cuales son las entidades fuertes, **primero tienes que entender que es una clave primaria...**

Antes te dije que **cada entidad tiene características propias que llamaremos atributos**, Uno de esos atributos se llama clave primaria, La clave primaria es **una columna cuyo valor es único e irrepetible**

**Las entidades fuertes son las que tienen una clave primaria que identifica el valor de cada tupla (cada fila que contiene varias columnas de información es llamada tupla. ) de manera única.**

Por consiguiente **no necesitan de ninguna entidad para su existencia.**

### Ejemplo entidad fuerte

- Empleados (***id***, nombres, apellidos)
- Productos (***serial***, modelos, pesos)
- Materias (***cod***, nombres, valor curricular)

*Las cursivas + **negritas** son las claves primarias*

Fuerte

# Modelo Entidad-Relacion

## TIPOS DE ENTIDAD BASE DE DATOS

### Entidades debiles:

Las entidades débiles **no dependen de una clave primaria propia sino de una entidad fuerte**, y no pueden existir sin una entidad que la represente.

No se puede identificar completamente por sus propios atributos y **toma la clave foránea de la entidad fuerte para poder existir.**

### Notación o símbolos de una entidad débil

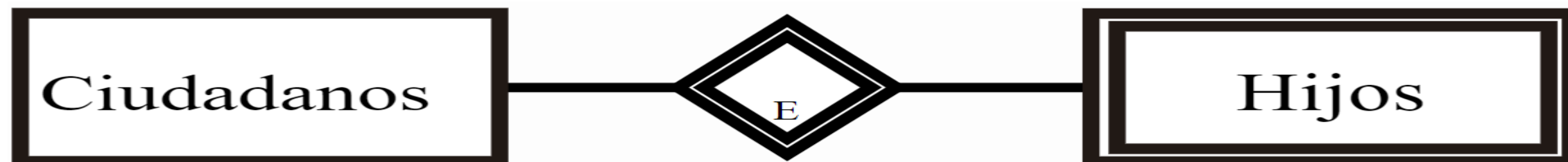
#### Ejemplo entidad débil

- Cuenta (num, saldo) -> Transacción (num. transacción, fecha, cantidad.)
- Facturas (num, fecha) -> Detalle\_facturas (num\_det, articulos, precio)



Las entidades débiles tienen dos tipos de dependencias:

**Dependencia por existencia:** Es cuando la entidad débil **necesita de la entidad fuerte para existir.** (*No son comunes*): En un censo para analizar los ciudadanos que tienen hijos, **la entidad hijo no podría existir sin la participación de la entidad ciudadanos.** Por lo cual si un ciudadano no tiene hijos, es imposible que exista un hijo.



# Modelo Entidad-Relacion

## TIPOS DE ENTIDAD BASE DE DATOS

### Dependencia por identidad

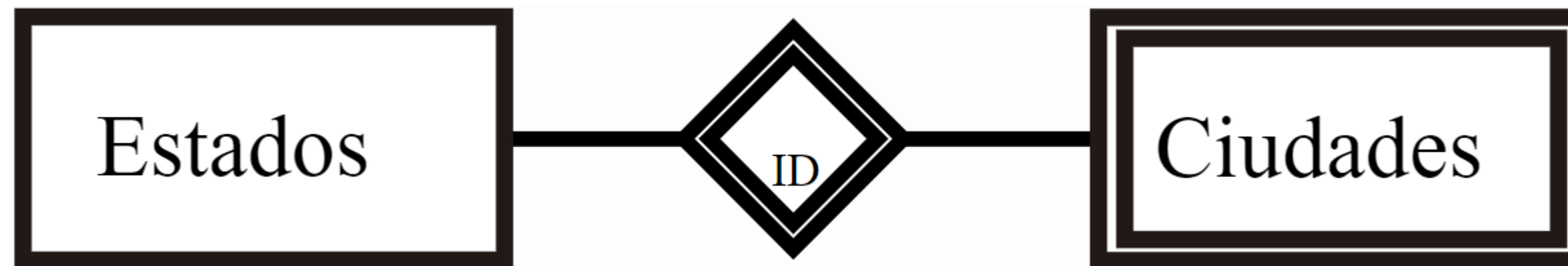
Ocurre cuando **necesitamos la entidad fuerte para poder identificar la entidad débil.**

Así se representan

**Ciudades es una entidad débil, con una dependencia por identificación.**

Porque **pueden existir dos ciudades con el mismo nombre**, como la ciudad de Salem. Una esta en el estado de Massachusetts – USA y la otra se encuentra en el estado de Baden-Wurtemberg en Alemania.

Entonces **necesitamos el estado para poder identificar a cual de las dos pertenece** y a diferencia de la anterior. Si la entidad estado cambia, las ciudades pueden seguir existiendo mientras tengan la dependencia.



## Ejemplos Entidad Débil

*A- Orden -> Productos Orden.*

*B- Estudiantes -> Casilleros*

Ambas opciones (A y B) son entidades débiles.

**La opción A es una entidad débil**, que tiene dependencia por existencia.

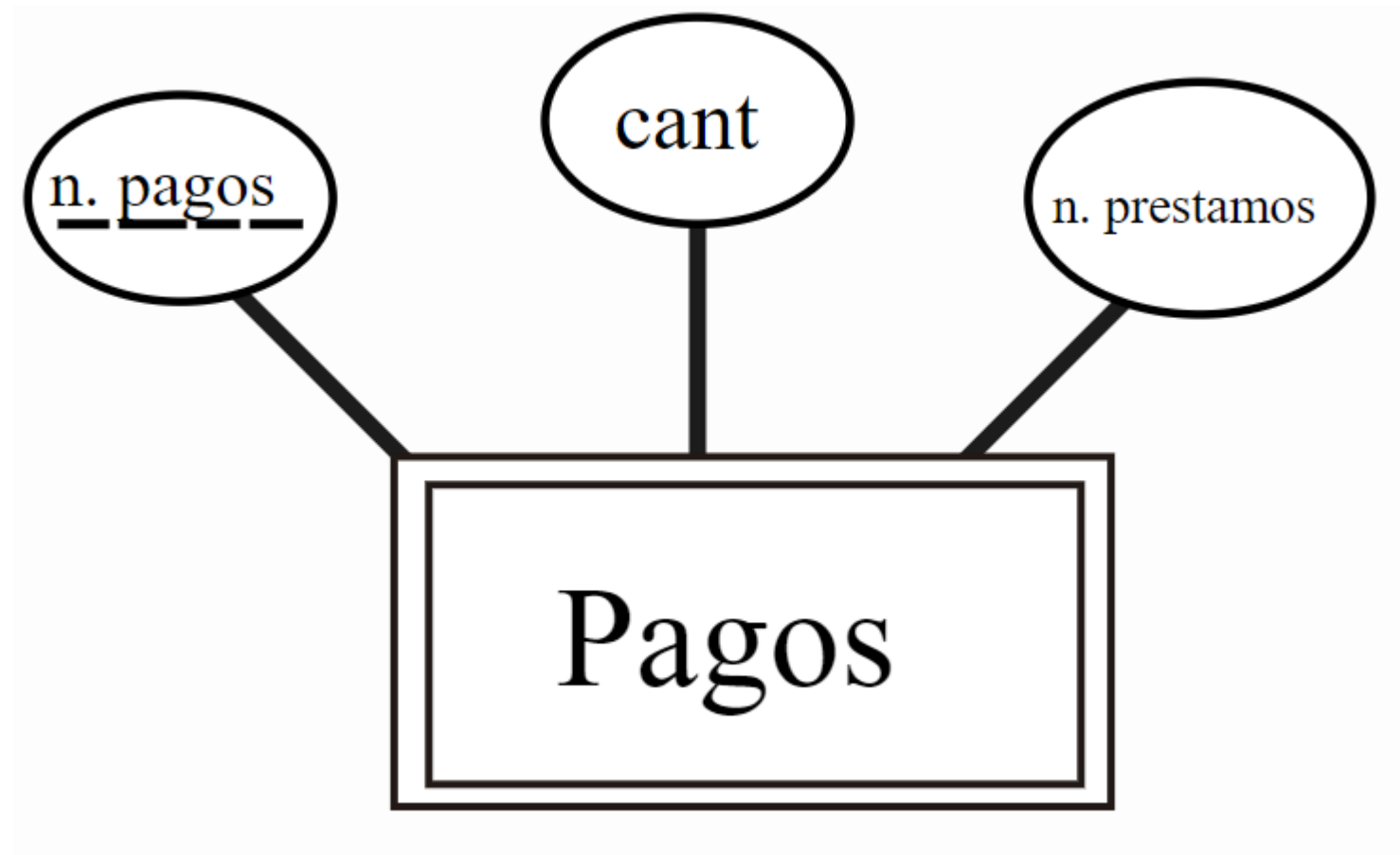
Porque sin una orden, no puede existir productos asociados a dicha orden.

**La opción B es una entidad débil**, con dependencia de identificación.

Porque los casilleros si existen en el mundo físico. Pero sin alumnos no puede existir una asociación de un casillero para cada estudiantes.

## Discriminante

Las entidades débiles dependen de las fuertes, el atributo discriminante es el atributo clave que usamos en las entidades débiles en combinación de la clave foránea proveniente de la entidad fuerte.





# Modelo Entidad-Relacion

## Valor y dominio

Una entidad se representa mediante un conjunto de **atributos**. Los atributos describen propiedades que posee cada miembro de un conjunto de entidades: En el futuro mientras las entidades se convierten en tablas, los atributos serán las columnas de dichas tablas.

Cada entidad tiene **atributos** (propiedades particulares que la describen).

Ejemplo: *La entidad EMPLEADO el tiene cuatro atributos:*

- *Nombre*
- *Dirección*
- *Edad*
- *Telefono;*

Cada entidad tiene un **valor** para cada uno de sus atributos., llamados el **dominio**, o el **conjunto de valores**, de ese atributo y tienen valores permitidos:

- *'José Pérez',*
- *'Honduras, Tegucigalpa, residencial plaza',*
- *'55'*
- *2236-0074*

**Cuando los atributos se transforman en columnas en nuestra base de datos**, requerimos especificarle que reglas tendrán esos atributos.

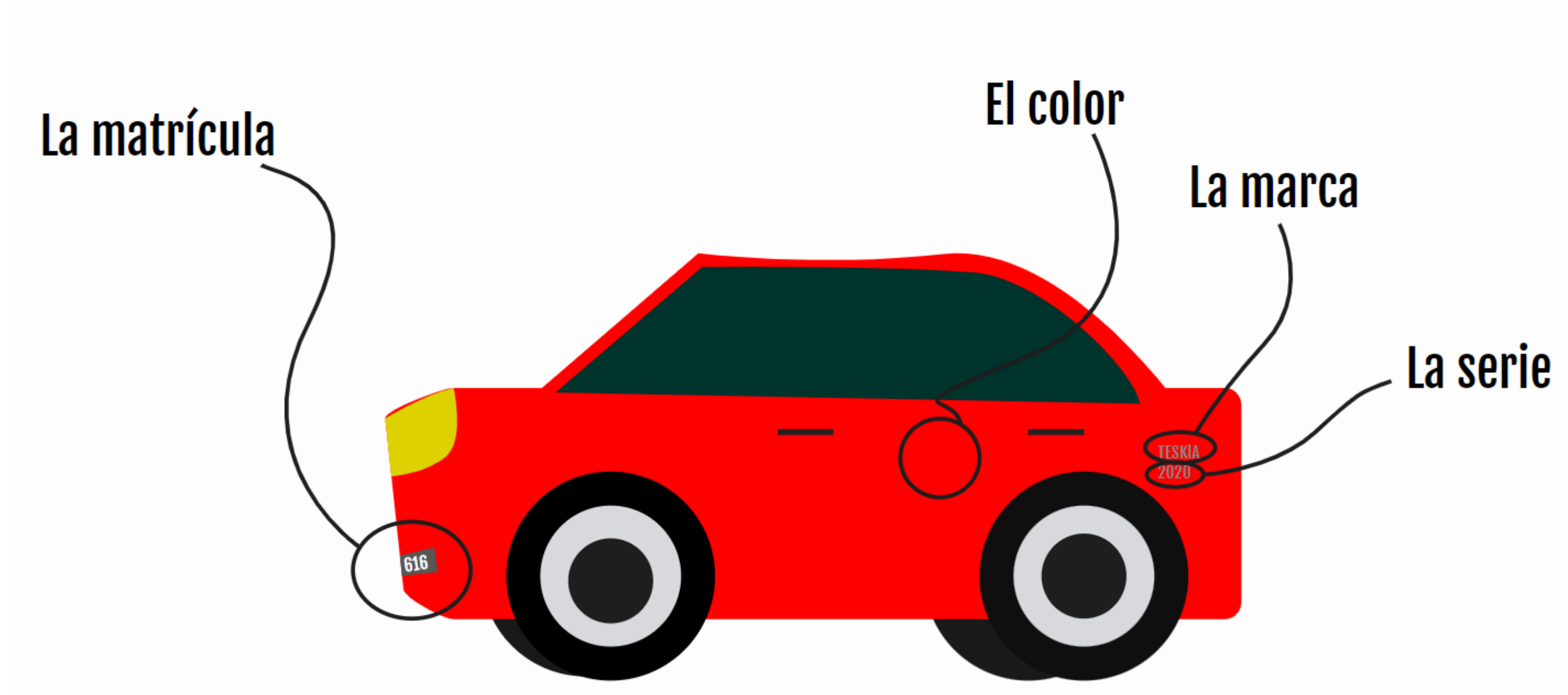
Para que tenga una coherencia lógica y una correcta integridad en los datos.

Los dominios **son esas especificaciones que nos permiten comprender que datos se deben almacenar y sus características.**

### **Propiedades de los dominios**

- **Nombre:** Un nombre correcto y descriptivos, sin caracteres especiales que permiten identificar el atributo.
- **Descripción:** Algún detalle que quieras agregar para entender este atributo.
- **Tipo de campo:** ¿Que almacenara? ¿Enteros? ¿Decimales? ¿Caracteres?
- **Tipo de dominio:** El rango de valores (cuanta información se guardara) y rango de codificado (por el momento no es necesario conocerlo)

## Modelo Entidad-Relacion Valor y dominio



# Modelo Entidad-Relacion

## Tipos de Atributos

Atributos **simples** : No pueden ser divididos

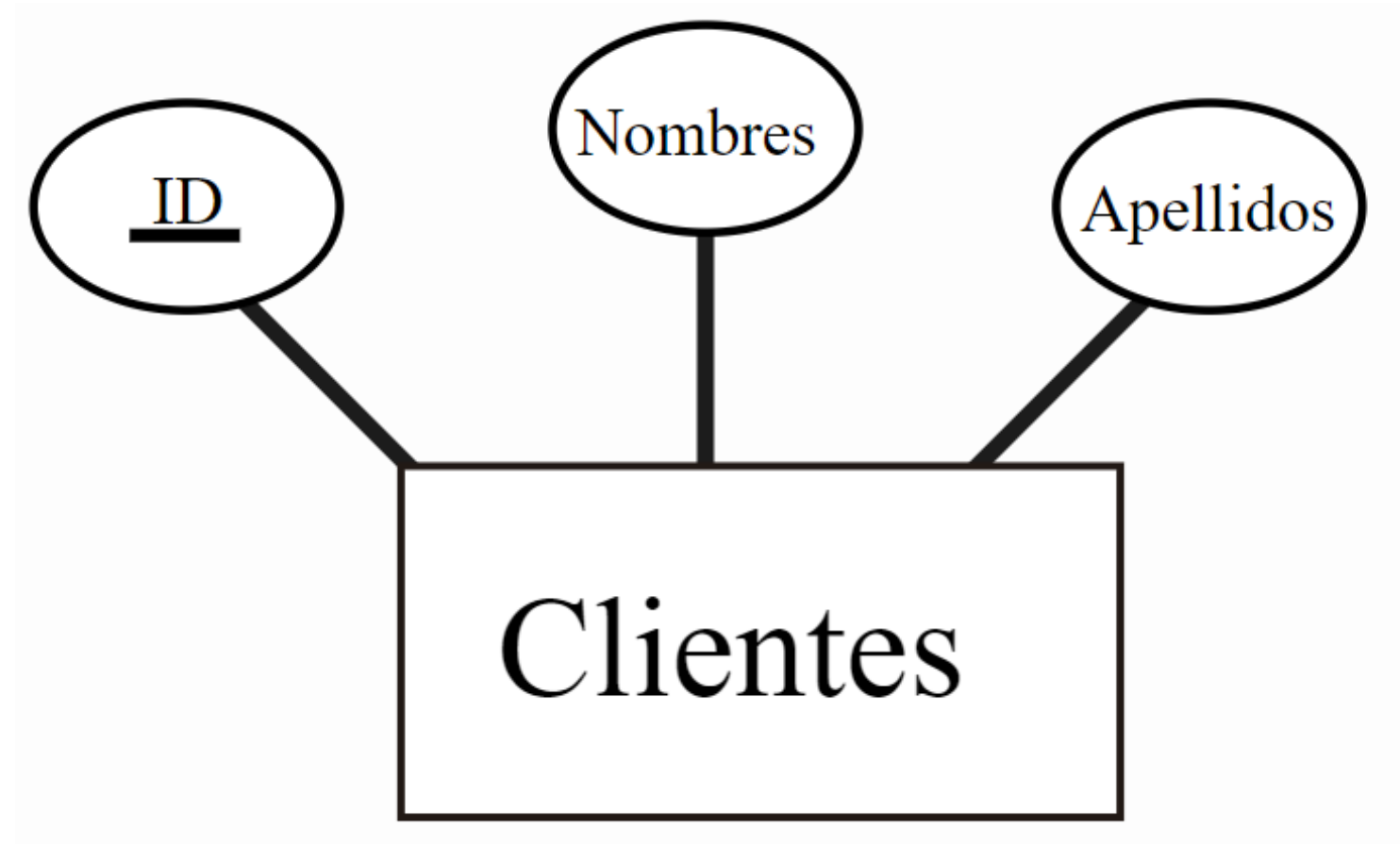
*Ejemplo:*

*Genero*

Son los mas comunes y se usan para definir las características de las entidades.

**Es el atributo mas común donde almacenaremos información.**

**pueden estar tanto en una entidad como en algunas relaciones** en caso de ser de mucho a muchos



# Modelo Entidad-Relacion

## Tipos de Atributos

Atributos **compuestos**: pueden ser divididos, también pueden tener una jerarquía.

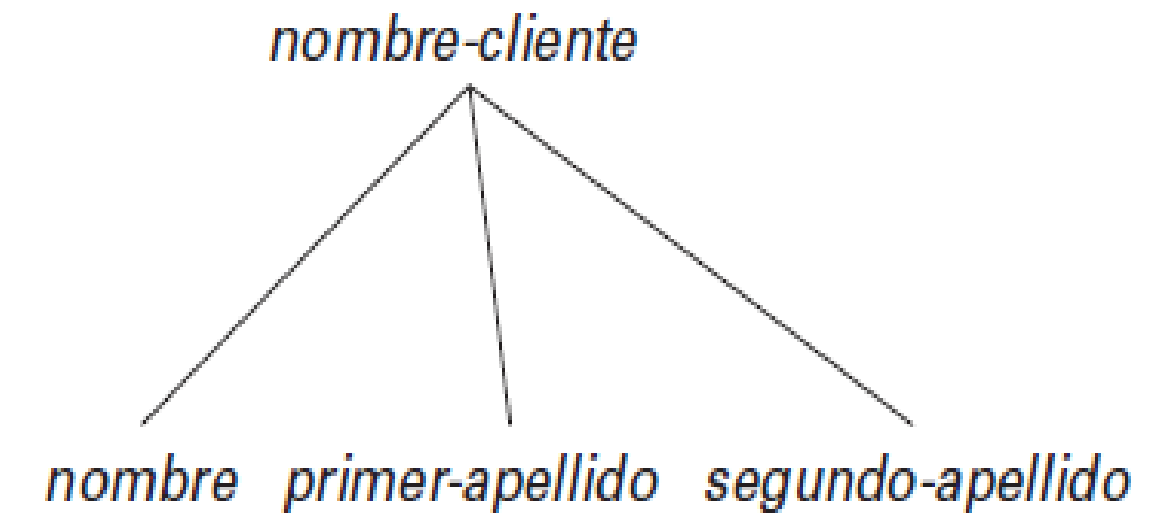
**Pueden tener diferentes dominios (características)**

*El atributo nombre, se puede descomponer en*

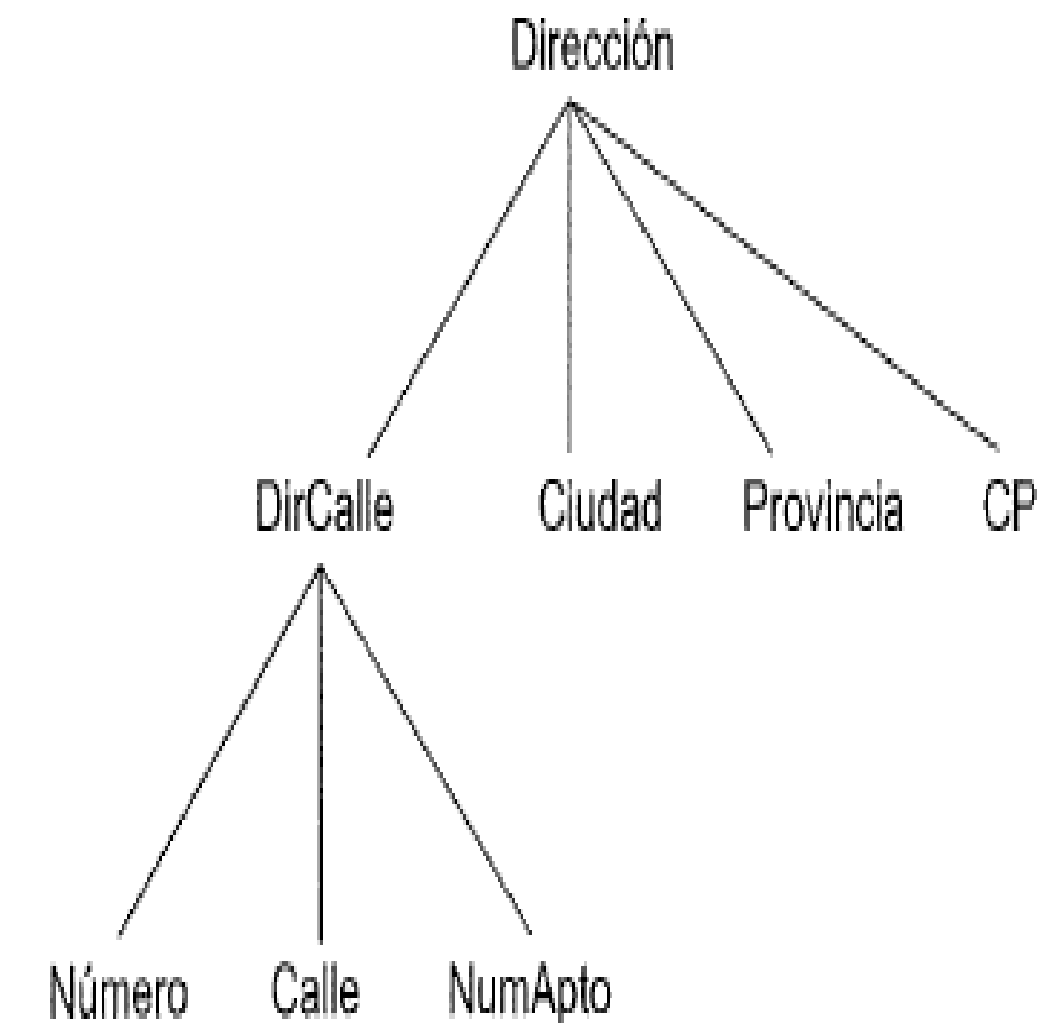
- *PrimerNombre,*
- *SegundoNombre,*
- *PrimerApellido*
- *SegundoApellido*

*El atributo Direccion, puede ser dividido en País, Departamento, Ciudad, Bloque, calle, etc*

Atributos  
compuestos



Atributos  
componentes



Si al cliente **ahora le quiero pedir la dirección.**

La dirección en una sola columna podría generar algún problema de redundancia. Entonces para evitar esto **digo en el modelo entidad relación que quiero dividir los datos que obtenga de ese atributos.**

De dirección yo **podría sacar, Estado, ciudad, nombre de la calle calle y numero de casa...** Solo por decir algunos.

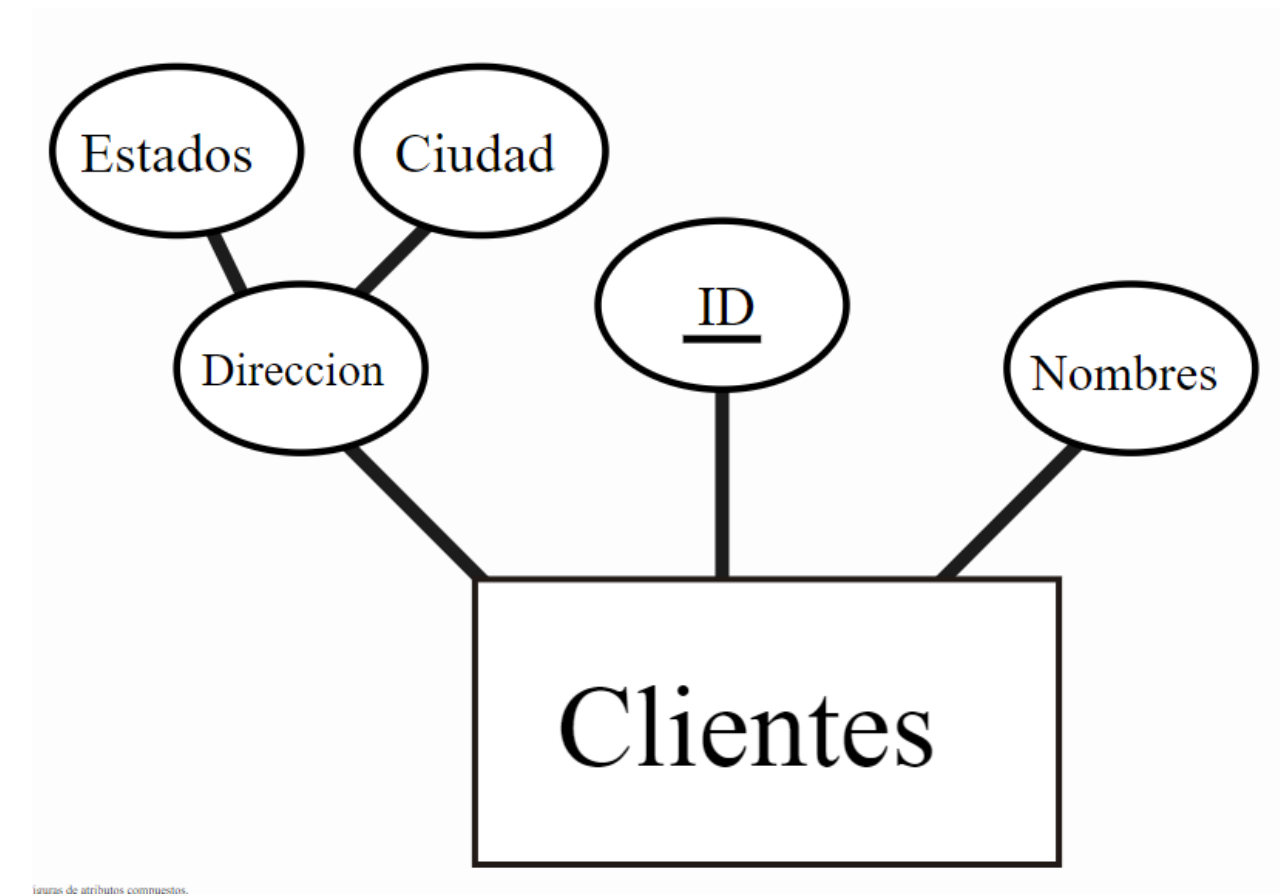
Estado, ciudad y nombre de calle pueden ser del mismo dominio. Pero numero de casa tendrá un dominio distinto.

**En este caso la única manera de solucionar es creando otra tabla. Y asociándola a la tabla clientes.**

Recuerda que **cada columna solo va a tener un solo tipo de datos.**

**No debes guardar ni separando con comas, ni en arreglos sino en columnas distintas con distintos dominios.**

De esta manera **tengo un atributo mas detallado que permitirá al analista de base de datos entender como esta estructurada la información**



# Modelo Entidad-Relacion

## Tipos de Atributos

Atributos **Monovalorados** : Solo posee un valor para una entidad en particular ejemplo *NumeroDeIdentidad*

Atributos **Multivalorados**: posee varios valores para cada entidad *Ejemplo: : NumeroTelefono*

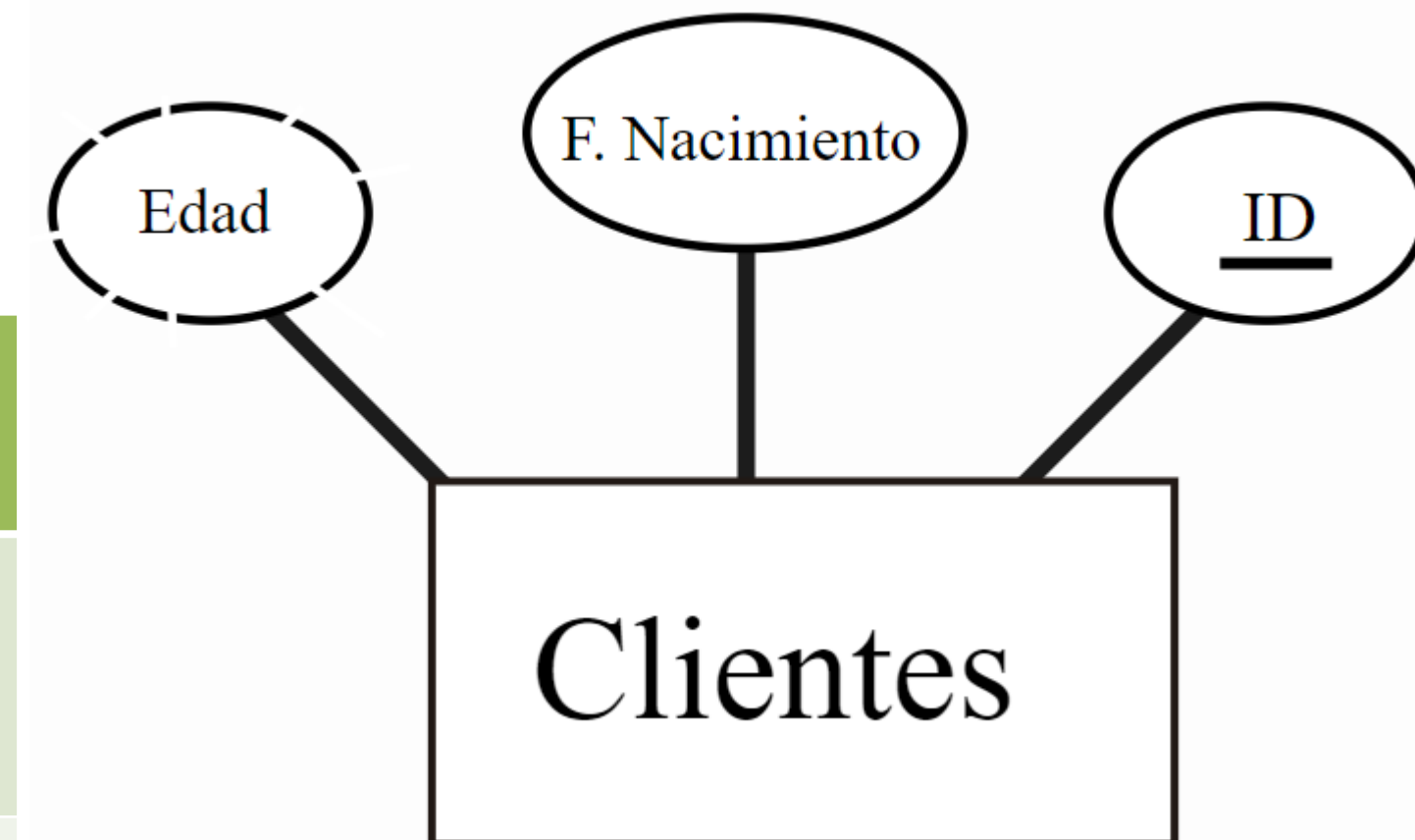
Atributos **Derivados**. El valor para este tipo de atributo se puede derivar de los valores de otros atributos o entidades relacionados.

*Ejemplo:*

*FechaDeNacimiento (Atributo base) Edad (Atributo derivado)*

*FechaComienzo (Atributo base) Antigüedad (Atributo derivado)*

| Nombre         | Numero de Teléfono               | Numero de Identidad |
|----------------|----------------------------------|---------------------|
| Santos Alfonso | 95201314<br>98721546<br>22270101 | 08011980010101      |
| Juan Fernández | 33405028                         | 08011990050505      |
| Ericka María   |                                  | 08011984232599      |





# Modelo Entidad-Relacion

## Atributos Multivaluados

Siguiendo con el ejemplo del cliente... **Imagina que quiero pedir dos números el de la casa y el personal. No puedo dibujar dos veces el mismo atributo**

Podemos insertar un atributo multivaluado que **se representa que este atributo puede replicarse.**

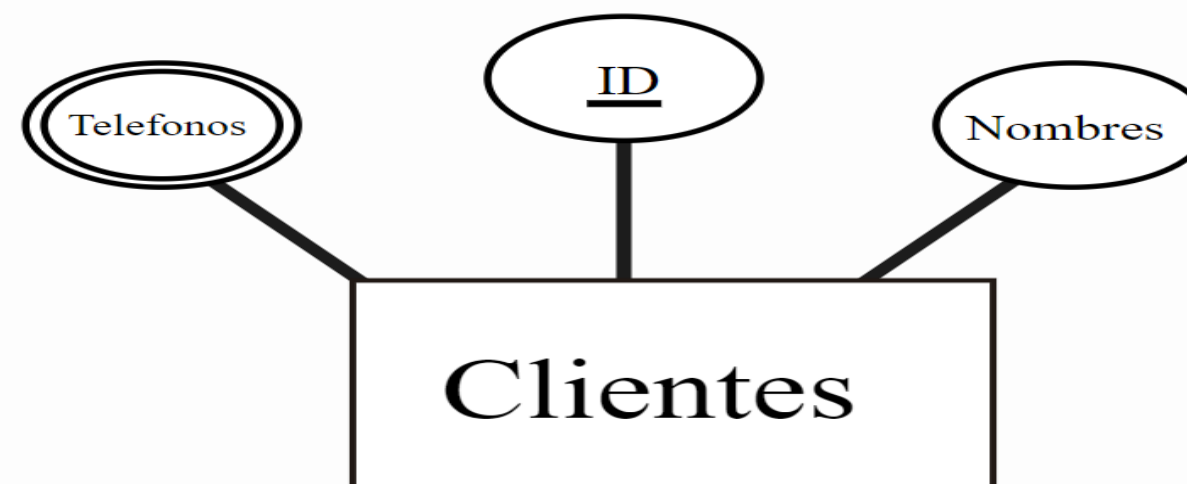
Cuando transformemos nuestra entidad en tablas, **si vemos este atributo hay dos formas de solucionar: La primera creando varias columnas del mismo dominio (que tenga las mismas características)**

Por ejemplo, **el numero de teléfono es de tipo entero.** Entonces ambas columnas que crearemos en la tabla final serán exactamente iguales.

**Y la segunda manera,** en caso de que no se sepa la cantidad de teléfonos que vas a obtener.

Puedes **crear una tabla exclusivamente para guardar el numero de teléfonos y la asocias a la tabla de clientes.**

***Alerta: nunca debes guardar dos números de teléfonos distintos en la misma columna porque esto es un error grave.***

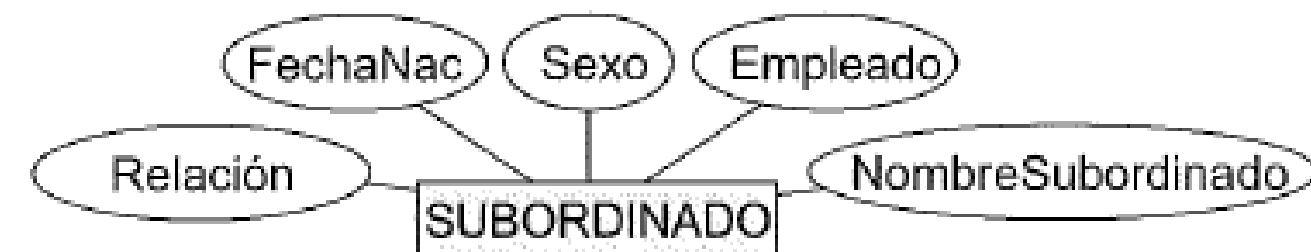
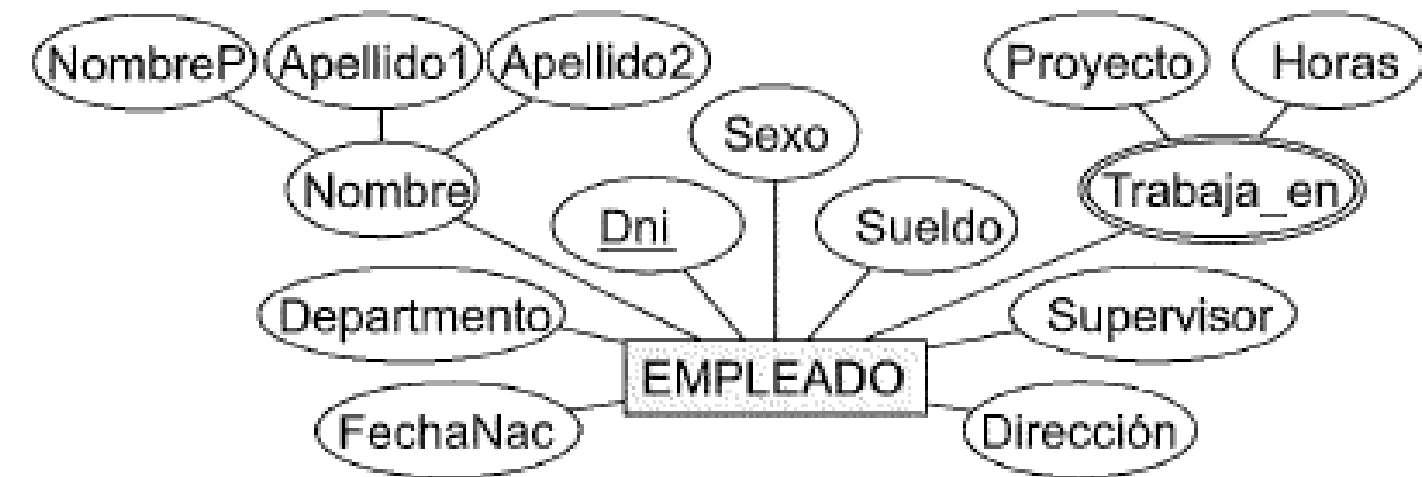
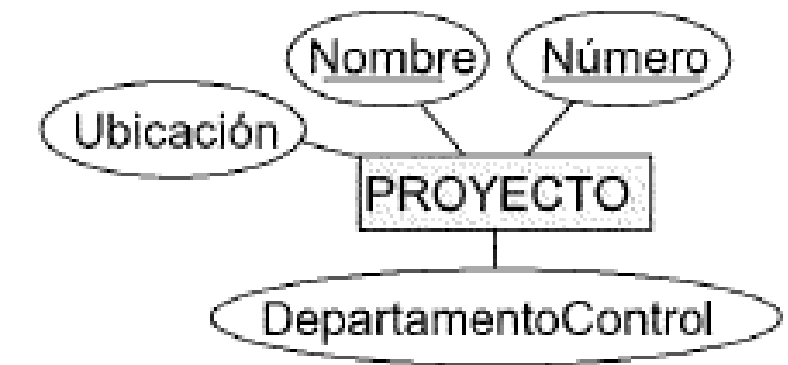
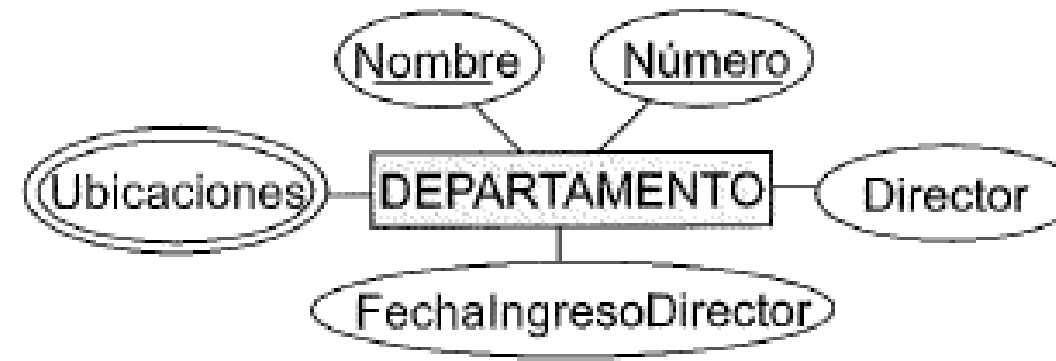


## Modelo Entidad-Relación Relaciones (Noción)

Hay varias *relaciones implícitas* entre los distintos tipos de entidades. De hecho, en cuanto un atributo de un tipo de entidad se refiere a otro tipo de entidad, decimos que existen algunas relaciones.

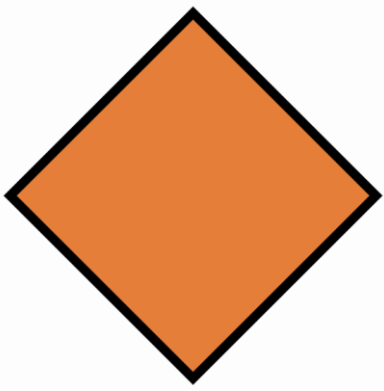
En el modelo ER, estas referencias no deben representarse como atributos, sino como **relaciones**.

Representar ***explícitamente*** las relaciones.



# Modelo Entidad-Relación

## Relaciones



Una **relación** es una asociación entre diferentes entidades

Un **tipo de relación**  $R$  entre  $n$  tipos de entidades  $E1, E2, \dots, En$  define un conjunto de asociaciones (o un **conjunto de relaciones**) entre las entidades de esos tipos de entidades.

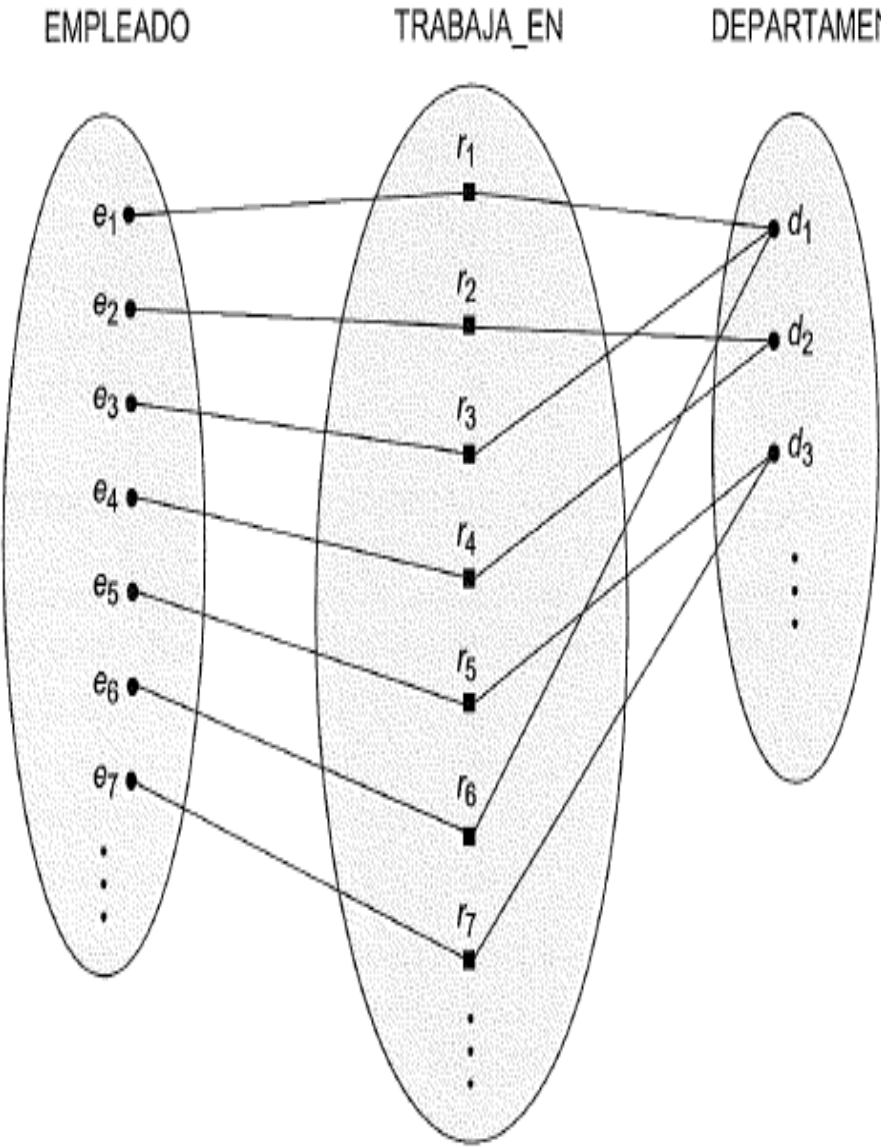
La asociación entre conjuntos de entidades se conoce como *participación*; es decir, los conjuntos de entidades **participan** en el conjunto de relaciones

La función que desempeña una entidad en una relación se llama **papel** de la entidad.

|            |            |          |            |      |       |
|------------|------------|----------|------------|------|-------|
| 32.112.312 | Santos     | Mayor    | Peguerinos | P-17 | 1.000 |
| 01.928.374 | Gómez      | Carretas | Cerceda    | P-23 | 2.000 |
| 67.789.901 | López      | Mayor    | Peguerinos | P-15 | 1.500 |
| 55.555.555 | Sotoca     | Real     | Cádiz      | P-14 | 1.500 |
| 24.466.880 | Pérez      | Carretas | Cerceda    | P-19 | 500   |
| 96.396.396 | Valdivieso | Goya     | Vigo       | P-11 | 900   |
| 33.557.799 | Fernández  | Jazmín   | León       | P-16 | 1.300 |

cliente

préstamo

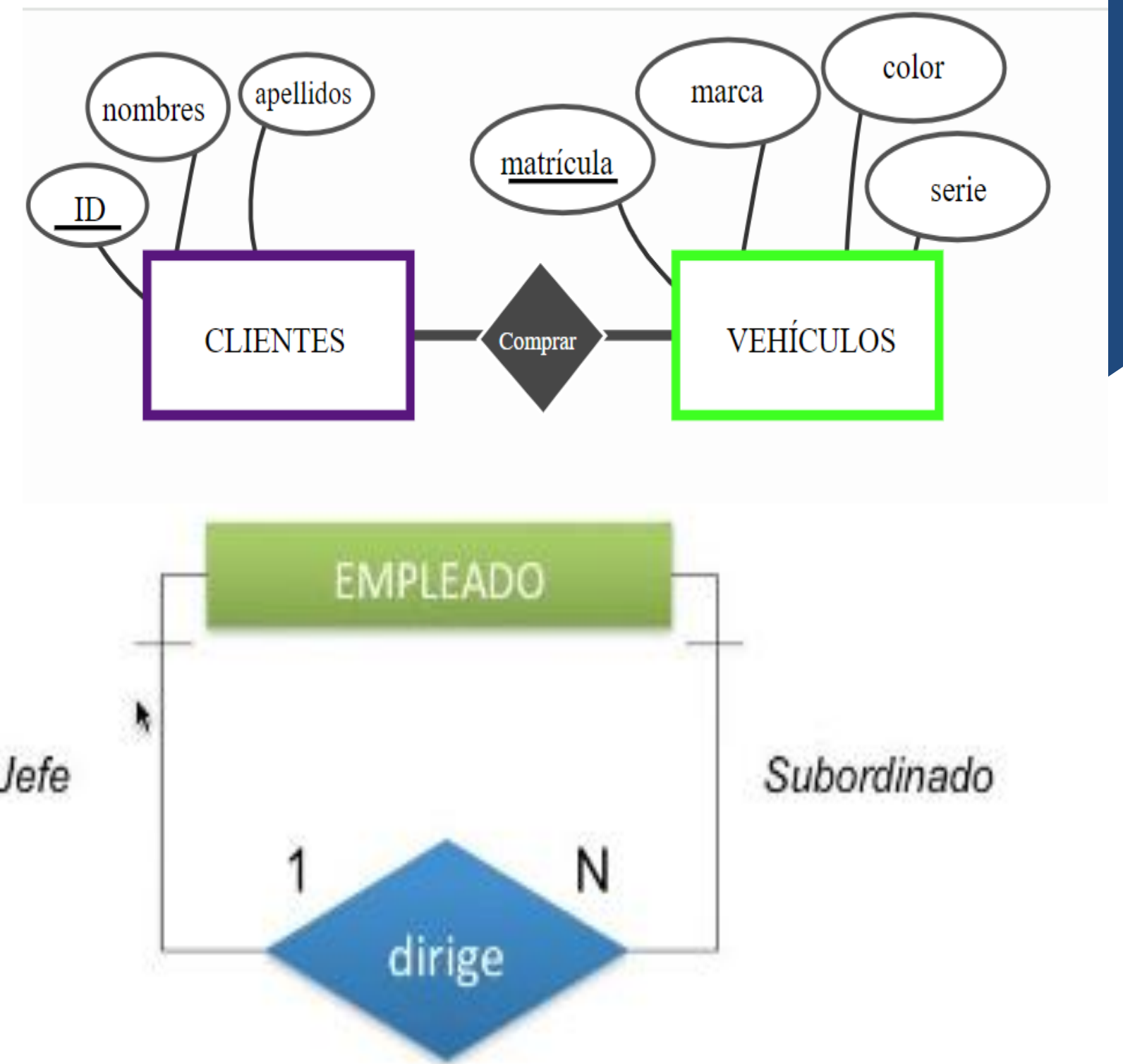


# Modelo Entidad-Relación

## Relaciones

Debido a que los conjuntos de entidades que participan en un conjunto de relaciones son generalmente distintos, los papeles están implícitos y no se especifican normalmente.

Sin embargo, son útiles cuando el significado de una relación necesita aclaración. Tal es el caso cuando los conjuntos de entidades de una relación no son distintos; es decir, el mismo conjunto de entidades participa en una relación más de una vez con diferentes papeles. En este tipo de conjunto de relaciones, que se llama algunas veces conjunto de relaciones **recursivo**, es necesario hacer explícitos los papeles para especificar cómo participa una entidad en un ejemplar de relación.





# Modelo Entidad-Relación

## RESTRICCIONES

### (Cardinalidades y Participación)

La **correspondencia de cardinalidades**, o razón de cardinalidad, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

# Modelo Entidad-Relación

## RESTRICCIONES

### (Cardinalidades y Participación)

La cardinalidad se divide en dos partes:

Cardinalidad Máxima (Grado).

Denotar es la cantidad de instancias de una entidad que están asociadas a una o más instancias de otra entidad, Dentro del Grado existen:

1. Relación de uno a muchos -
2. Relación de muchos a muchos
3. Relación de uno a uno.

Cardinalidad Mínima (opcionalidad).

Denota si las instancias de una entidad **deben** estar asociadas a una instancia de otra entidad, o no necesariamente, Dentro de la opcionalidad existen:

- Relaciones mandatorias
- Relaciones opcionales

\_\_\_\_\_

-----

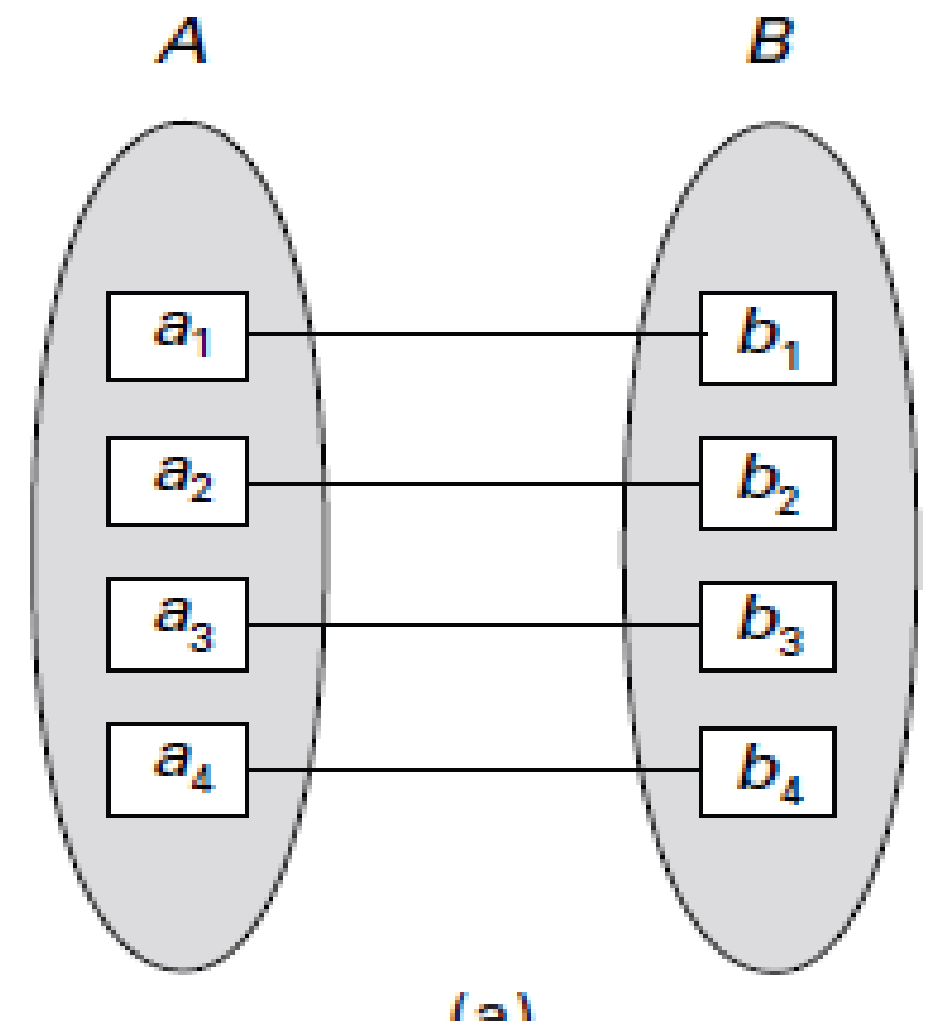
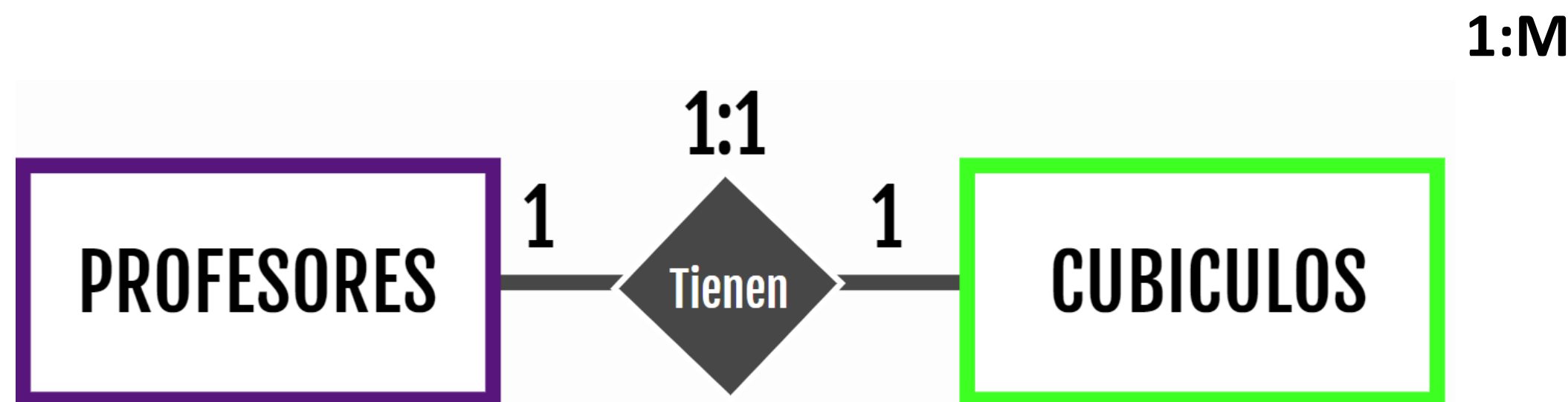


# Modelo Entidad-Relación

## RESTRICCIONES

(Cardinalidades y Participación)

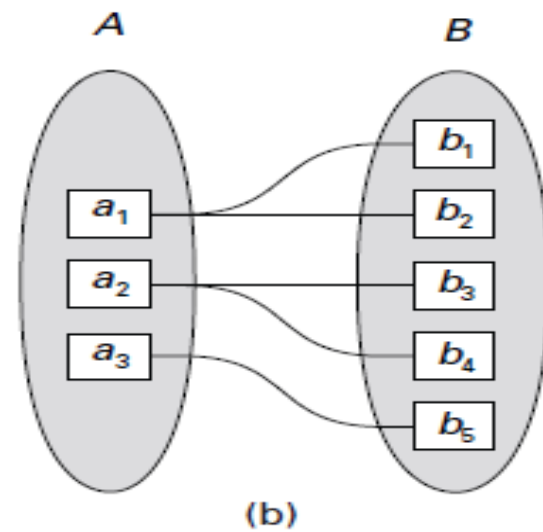
- **Uno a uno.** Una entidad en  $A$  se asocia con *a lo sumo* una entidad en  $B$ , y una entidad en  $B$  se asocia con *a lo sumo* una entidad en  $A$  (véase la Figura 2.4a).





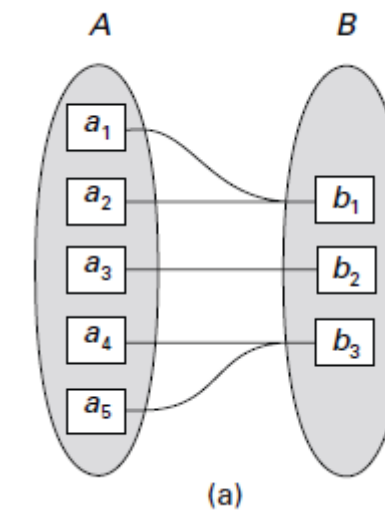
# Modelo Entidad-Relación (Cardinalidades y Participación)

- **Uno a varios.** Una entidad en  $A$  se asocia con cualquier número de entidades en  $B$  (ninguna o varias). Una entidad en  $B$ , sin embargo, se puede asociar con *a lo sumo* una entidad en  $A$  (véase la Figura 2.4b).

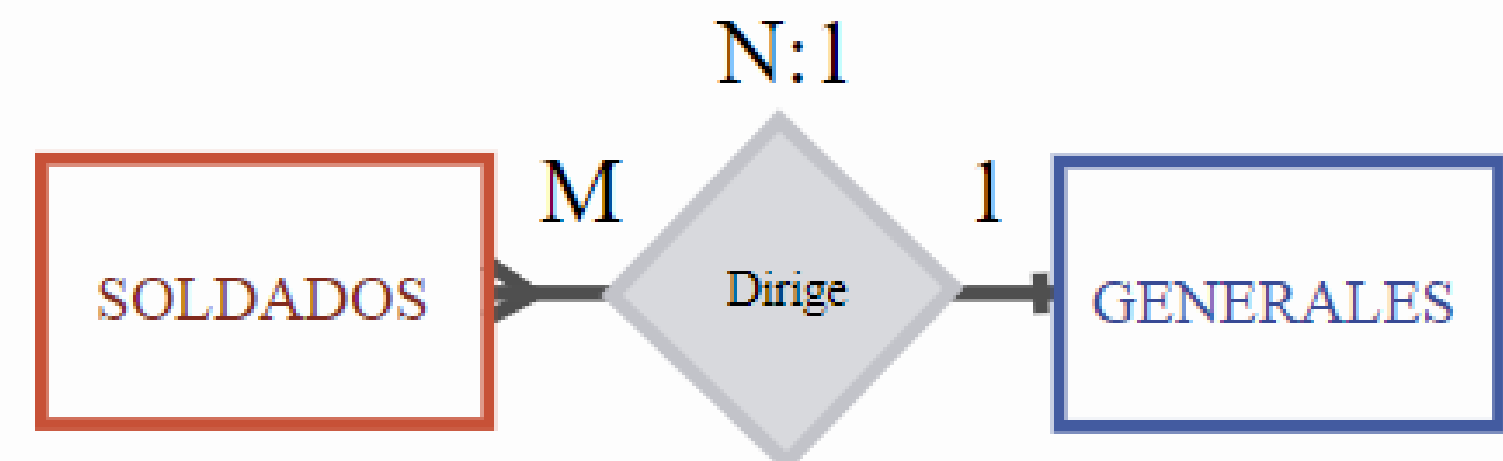
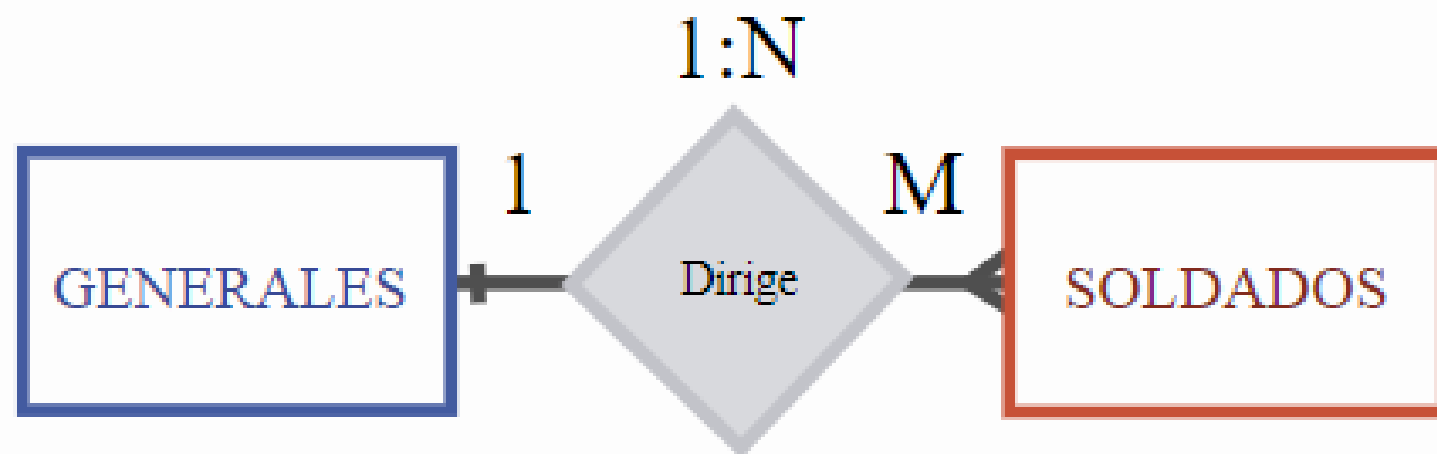


**1:M**

- **Varios a uno.** Una entidad en  $A$  se asocia con *a lo sumo* una entidad en  $B$ . Una entidad en  $B$ , sin embargo, se puede asociar con cualquier número de entidades (ninguna o varias) en  $A$  (véase la Figura 2.5a).



**M:1**

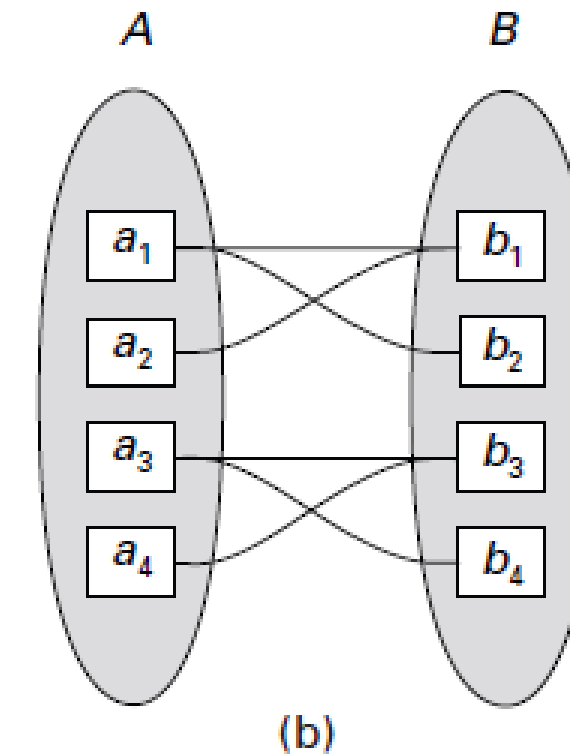


# Modelo Entidad-Relación

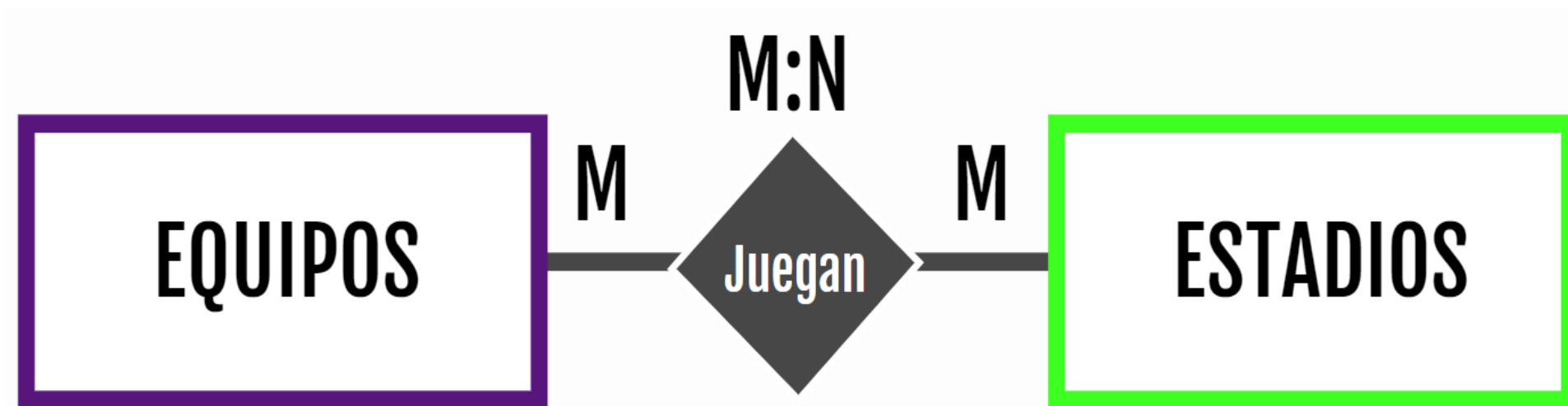
## RESTRICCIONES

(Cardinalidades y Participación)

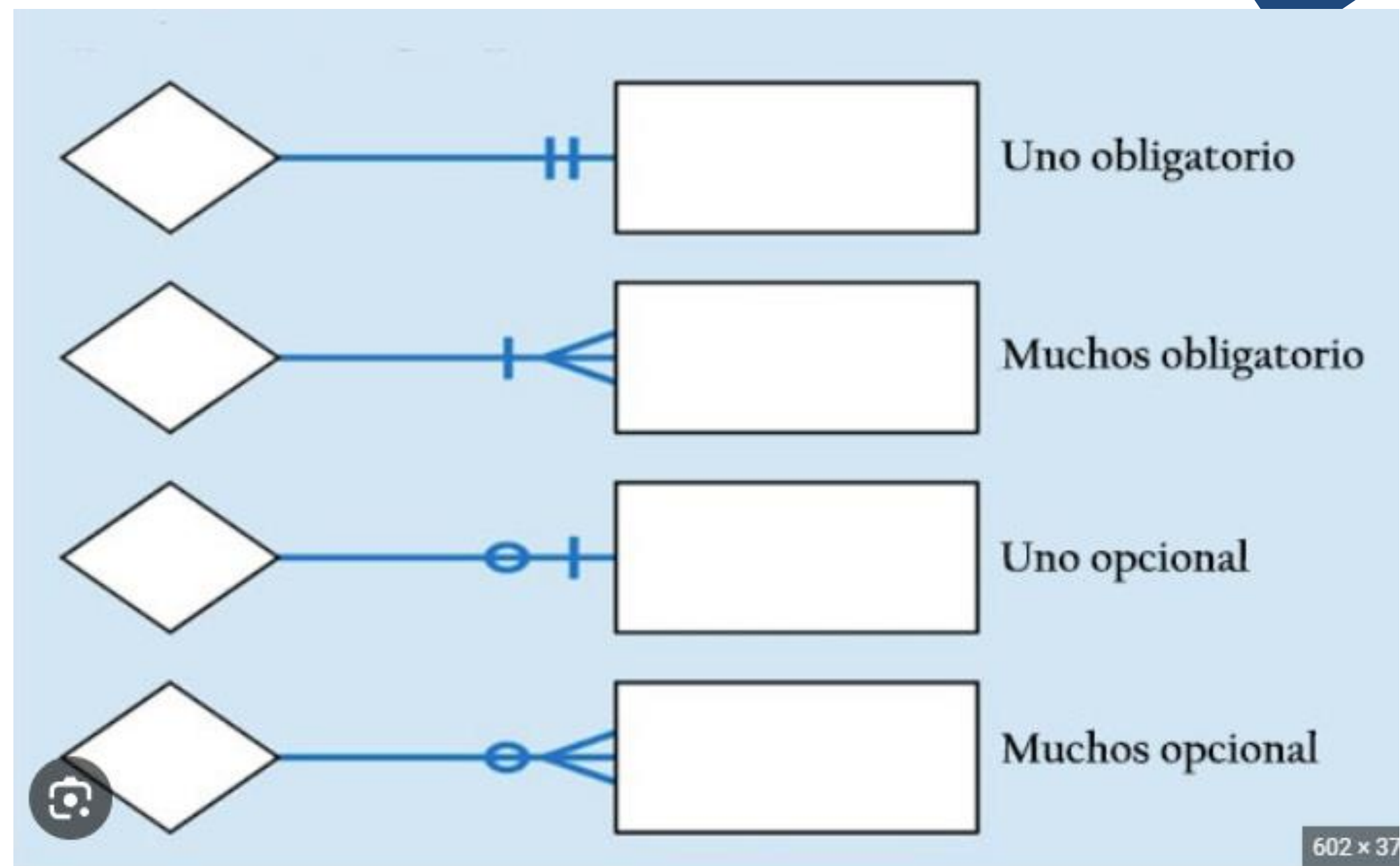
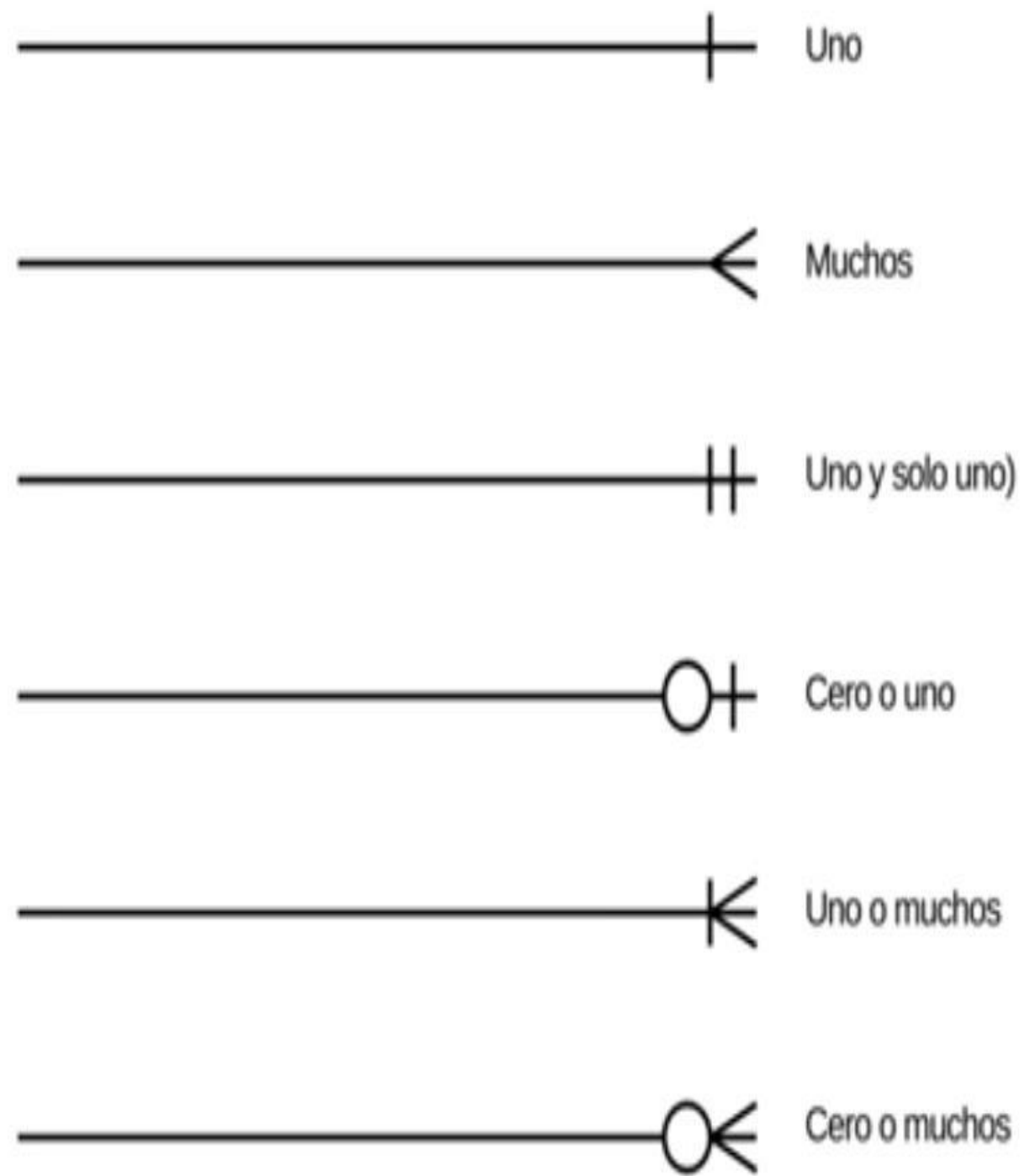
- **Varios a varios.** Una entidad en  $A$  se asocia con cualquier número de entidades (ninguna o varias) en  $B$ , y una entidad en  $B$  se asocia con cualquier número de entidades (ninguna o varias) en  $A$  (véase la Figura 2.5b).



**M:N**



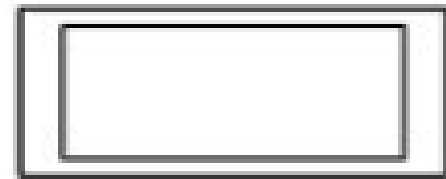
## Cardinalidad



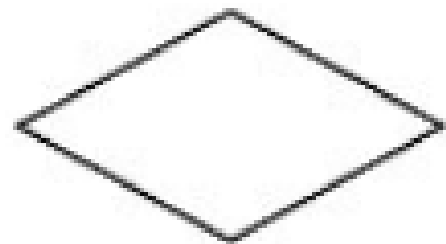
## Cardinalidad



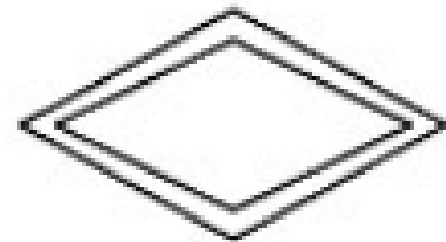
**Entidad**



**Entidad débil**



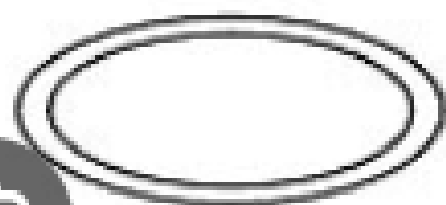
**Relación**



**Relación débil**



**Atributo**



**Atributo multivalor**



Cero o uno



Muchos



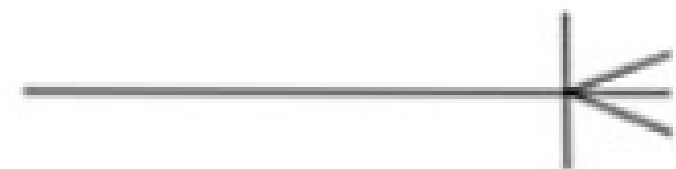
Uno



Uno y solo uno)

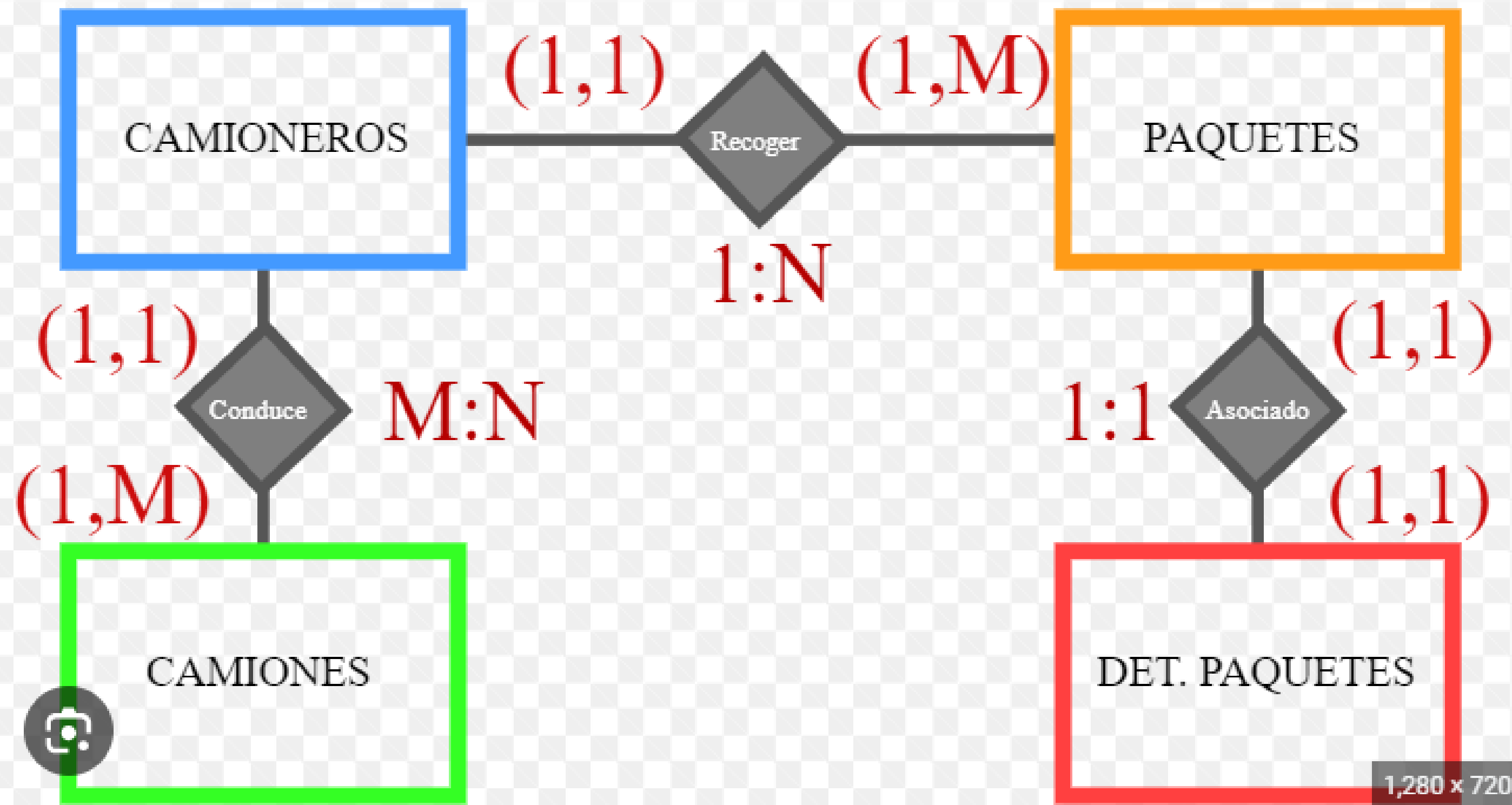


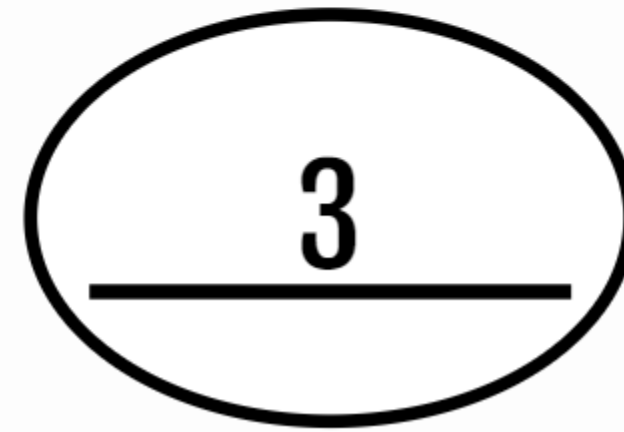
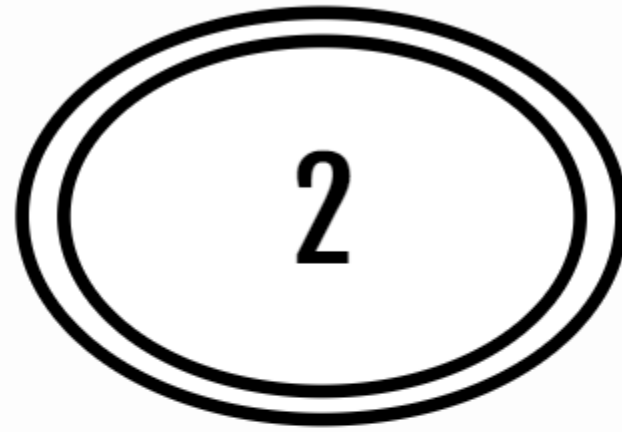
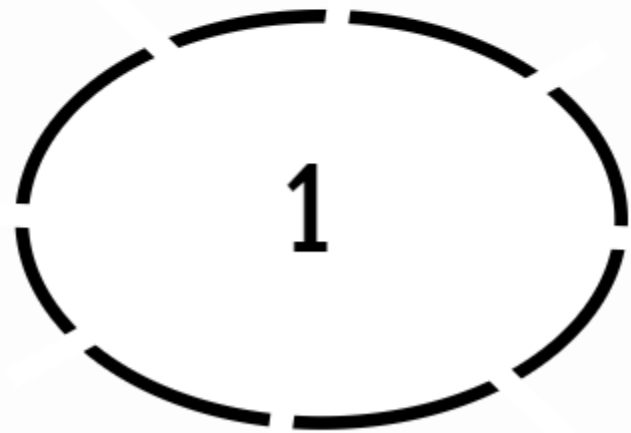
Cero o muchos



Uno o muchos







- 1.Derivados.
- 2.Multivaluados.
- 3.Clave primaria.











