



# *Bases de Datos*

## Tema 3: El modelo de datos Entidad-Relación

*Dpto. de Ingeniería Informática*

# Contenidos

- Introducción
- Conceptos del modelo E/R
- El modelo EE/R
- Referencias

# Introducción

- El modelo Entidad-Relación (E/R):
  - Es un modelo conceptual de alto nivel
  - Representa el mundo a modelar mediante estructuras lógica y abstractas de los datos
  - Se basa en entidades y relaciones
- Propuesto por P. Chen en los años 70
- Modelo con mayor difusión entre las BD
- Propiedades:
  - Estática: estructuras
  - Dinámica: operaciones
- Independiente del SGBD y del hardware

# Conceptos del modelo E/R

- Todo modelo dispone de un conjunto de herramientas que ayuda a conseguir el esquema correspondiente
  - El E/R es un modelo abstracto: ofrece un conjunto de símbolos para representar los conceptos del mundo
- Conceptos a representar:
  - Entidades
  - Relaciones
  - Conexiones
  - Atributos
  - Dominios
  - Restricciones

# Entidad

- Objeto del mundo a modelar con existencia propia y distinguible del resto de objetos
- Pueden ser:
  - Físico: objeto tangible
    - Un libro
    - Una persona
    - Una película
    - Una casa
  - Abstracto: objeto no tangible
    - Un curso universitario
    - Una provincia
    - Una transacción bancaria
    - Un puesto de trabajo
- Se representan igual sean físicos o abstractos

# Atributos

- Los atributos indican propiedades específicas de las entidades
- Por ejemplo:
  - Entidad coche, atributos: matricula, nº chasis, marca, modelo, ...
  - Entidad asignatura, atributos: código, nombre, créditoaje, curso, ...
- Cada instancia de una entidad tiene un valor para cada uno de sus atributos
- Tipo de atributos:
  - Simples o compuestos
  - Monovaluados o multivaluados
  - Almacenados o derivados

# Atributos

- Atributos simples o compuestos:
  - Compuesto: se pueden dividir en componentes más pequeños con significado propio. El valor es la concatenación de los valores de los atributos
    - Útil para modelar situaciones en las que se hace referencia a la propiedad como una unidad, pero también a los componentes por separados
    - Ejemplo: fecha=(día,mes,año)
  - Simples: no son divisibles
    - Útiles cuando siempre nos referimos al atributos sin necesidad de dividirlo

# Atributos

- Atributos monovaluados o multivaluados:
  - Monovaluados: tienen un solo valor para cada entidad
    - Una persona: DNI, fecha de nacimiento, DNI\_padre, DNI\_madre, . . .
  - Multivaluados: tienen más de un valor para la misma entidad
    - Ej: el libro [Elma02] tiene dos autores.
  - Pueden tener límites superior y/o inferior del nº de valores por entidad
    - Ej: Teléfono de un profesor: 0 a 2, despacho y/o dpto.



# Atributos

- Atributos almacenados o derivados:
  - Derivados: su valor se calcula a partir de otra información ya existente (atributos, interrelaciones, ...)
    - Es una información redundante (evitarlo):
      - Edad de una persona: se calcula de su fecha de nacimiento (un atributo)
        - Aunque a veces sí puedo guardar edad (de jubilación) o la edad con la que alguien hizo algo (si no quiero/puedo saber su fecha de nacimiento)
      - Nº de artículos suministrados por un proveedor: se calcula por interrelación
  - Almacenados: su valor no se deriva de otra información
    - Nombre de una persona, apellidos, etc

# Atributos

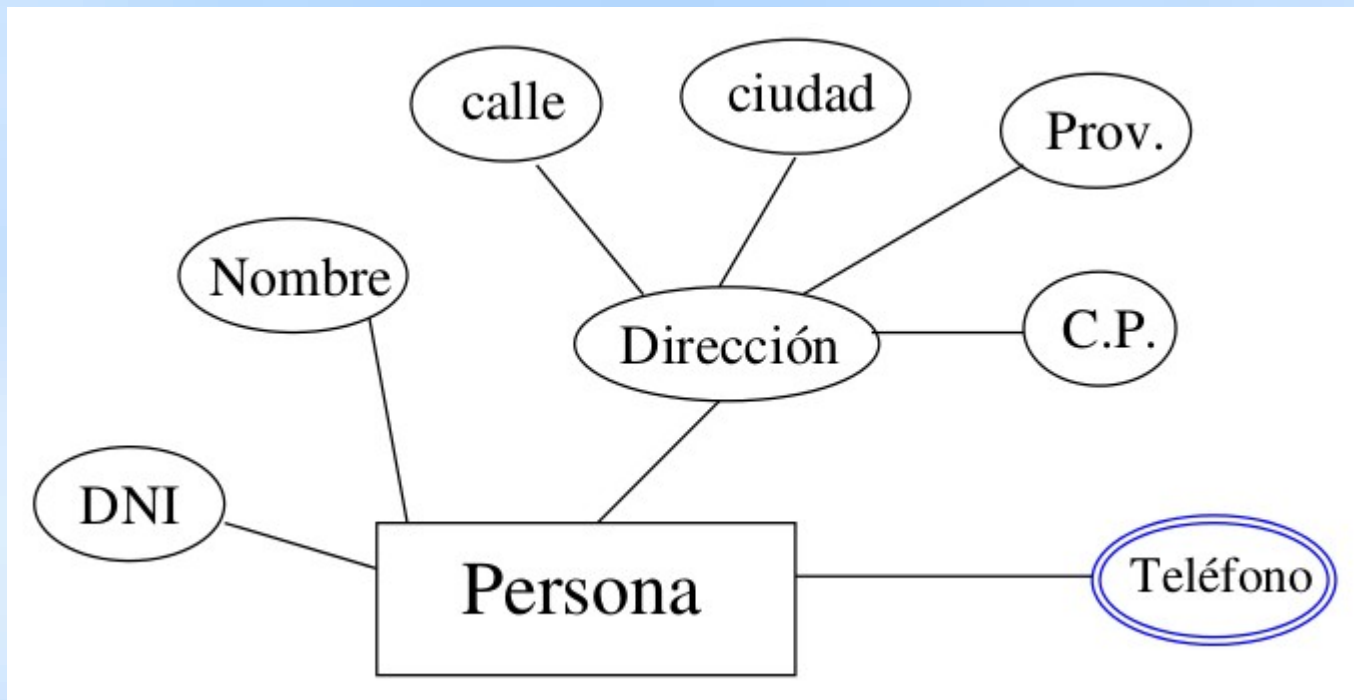
- Valor nulo (*null*) se crea cuando se dan situaciones especiales para un atributo concreto. Por ejemplo:
  - El valor para el atributo existe pero se desconoce. Ej: tengo un cliente pero no sé su dirección
  - No se sabe si el valor del atributo existe o no. Ej: no sabemos si un cliente tiene teléfono fijo
  - La entidad no tiene ningún valor aplicable para el atributo (puede ser temporal o permanentemente)
    - Ej: fecha de la última ITV de un coche recién comprado
    - Ej2: fecha de la próxima ITV de un coche siniestrado

# Tipo de entidad

- Lo común es que muchas entidades posean los mismos atributos y se traten igual → se agrupan formando un conjunto de entidades del mismo tipo
- Un *tipo de entidad* se describe por su nombre y la lista de nombres de sus atributos (*esquema*)
  - Persona (DNI, nombre, apellidos, ...)
  - Asignatura (asg\_num, asg\_nom, ...)
- Una *ocurrencia* de un tipo de entidad es una entidad con valores para cada atributo
  - Persona (91404505, Pepe, Pérez, . . . )
  - Asignatura (24, Dibujo técnico, . . . )
- Un conjunto de ocurrencias forma la *instancia* (vs. esquema)
  - Ejemplo: conjunto de todas las personas.

# Notación

- Tipos de entidades: se representan por rectángulos
- Atributos: se representan por elipses
- Representación del tipo entidad Persona que tiene un atributo compuesto (dirección) y otro multivaluado (teléfono)



# Atributos claves

- Todos los tipos de entidades suelen tener un atributo (o conjunto de ellos) cuyo valor es distinto para cada entidad individual → atributos claves
  - Los atributos claves definen unívocamente a cada entidad dentro de un conjunto de entidades
- Puede existir más de un atributo (o conjunto de ellos) que cumpla la condición → claves candidatas
  - Persona (DNI, nomb, apell, dom, fec-nac, ...)
    - Fecha es *año-mes-día-hora-minutos-segundos*
  - Claves candidatas:
    - a) {DNI}
    - b) {nomb,apell,fec-nac,dom}

# Atributos claves

- Pregunta: ¿es el DNI un identificador único en España?

# Atributos claves

- Pregunta: ¿es el DNI un identificador único en España?
- No, no lo es.
  - Y no lo digo yo: *“Estimación de los DNI duplicados en España”, de Justino García del Vello. Revista Estadística Española, Vol. 38, Núm. 141, 1996, págs. 219 a 235.*
- Entonces, ¿vale como identificador único?
  - En clase sí, supondremos que es el DNle

# Atributos claves

- Clave primaria: es la clave candidata elegida por el diseñador de la DB para identificar cada ocurrencia de un tipo de entidad
- Casi todos los modelos la representan subrayando sus miembros:
  - Persona (DNI, nomb, apell, dom, fec-nac, . . . )
  - Asignatura (cod\_asig\_cod, nom\_asig, . . . )
- Las claves candidatas no elegidas quedan como claves alternativas
  - Persona (DNI, nomb, apell, dom, fec-nac, . . . )
  - Atributos claves:
    - {DNI} clave primaria (CP)
    - {nomb,apell,fec-nac,dom} clave alternativa (CA)



# Atributos claves

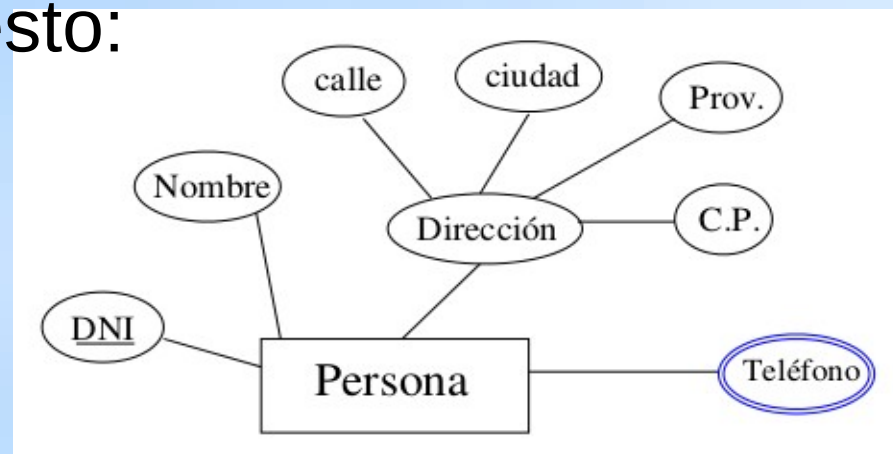
- Pregunta: ¿El nombre y apellidos de una persona puede cambiar en el tiempo?

# Atributos claves

- Pregunta: ¿El nombre y apellidos de una persona puede cambiar en el tiempo?
  - Hay países donde las mujeres al casarse adquieren el apellido de su marido
    - Y al divorciarse pueden perderlo o no
  - Puede haber autorizaciones judiciales (cambio de sexo, adopciones, ...)

# Notación

- Atributos que pertenecen a la clave primaria: se representan subrayados
- Representación del tipo entidad Persona con:
  - Atributo DNI como clave primaria
  - Atributo Dirección compuesto:
    - calle
    - ciudad
    - provincia
    - código postal
  - Atributo teléfono es multivaluado



# Dominios

- Dominio: conjunto de valores que puede tomar un atributo
- Todo atributo atómico tiene asociado un único dominio que indica sus valores válidos
- Matemáticamente: sean  $D$  un dominio y  $P$  un predicado asociado a  $D$ . Dos formas de expresarlo

$v \in D | P(v)$  es verdadero

$$D = \{v_i | P(v_i)\}$$

- Cada atributo  $a_i \in E$  toma valores en uno de los dominios válidos del conjunto de dominios del tipo de entidad

# Cardinalidad

- Cardinalidad de un atributo: números mínimo y máximo de valores que puede tomar un atributo ( $a$ ) en una ocurrencia de un tipo de entidad ( $E$ ). Ej:
  - $\text{Card\_min}(a, E) = 0$ 
    - $a$  puede no tomar valor
    - $a$  puede ser nulo (null)
  - $\text{Card\_min}(a, E) = 1$ 
    - $a$  debe tomar obligatoriamente un valor
  - $\text{Card\_max}(a, E) = 1$ 
    - $a$  tomará, como mucho, un valor individual a la vez
  - $\text{Card\_max}(a, E) > 1$ 
    - $a$  puede tomar más de un valor para la misma ocurrencia de entidad
    - $a$  es multivaluado

# Relaciones

- Una relación es una asociación entre entidades que está relacionadas en el mundo de estudio
- Un tipo de relación es un conjunto de relaciones que comparten las mismas características
  - Un tipo de relación  $R$  entre  $n$  tipos de entidades  $E_1, E_2, \dots, E_n$  define un conjunto de asociaciones entre entidades de esos tipos
- Matemáticamente: un tipo de relación es una relación matemática sobre  $E_1, E_2, \dots, E_n$  que puede definirse también como un subconjunto del producto cartesiano

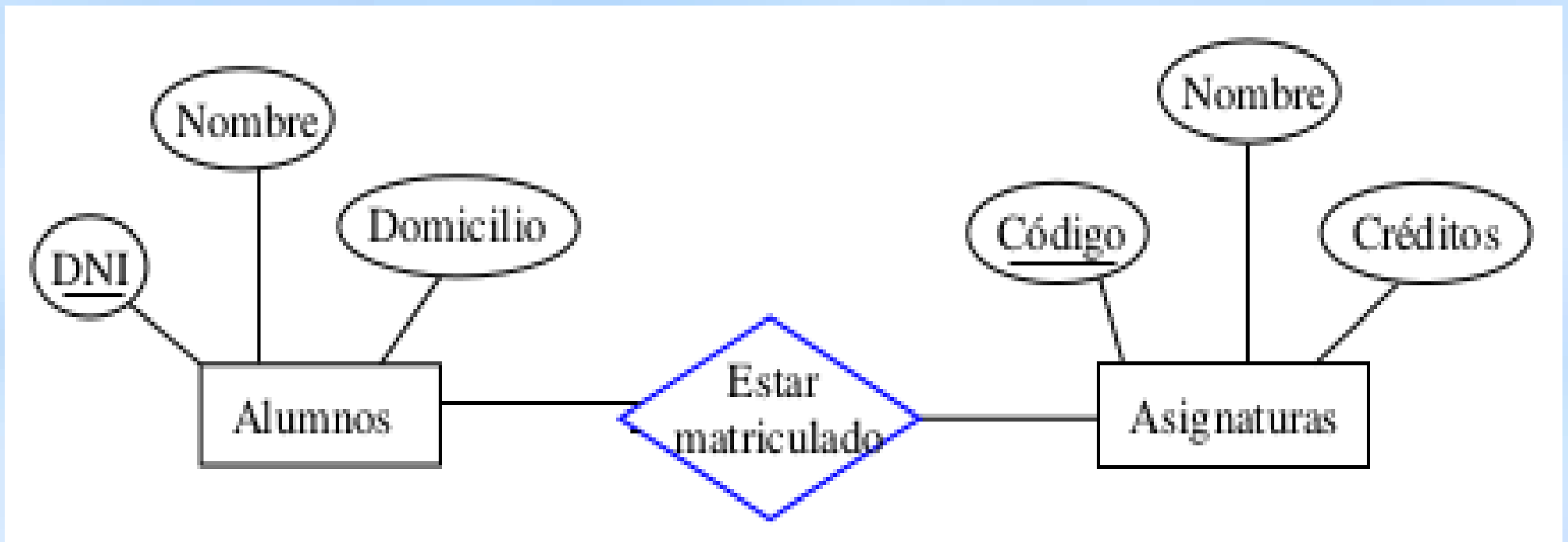
$$E_1 \times E_2 \times \dots \times E_n$$

$$\{(e_1, e_2, \dots, e_n) | e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

donde  $(e_1, e_2, \dots, e_n)$  es una relación

# Relaciones

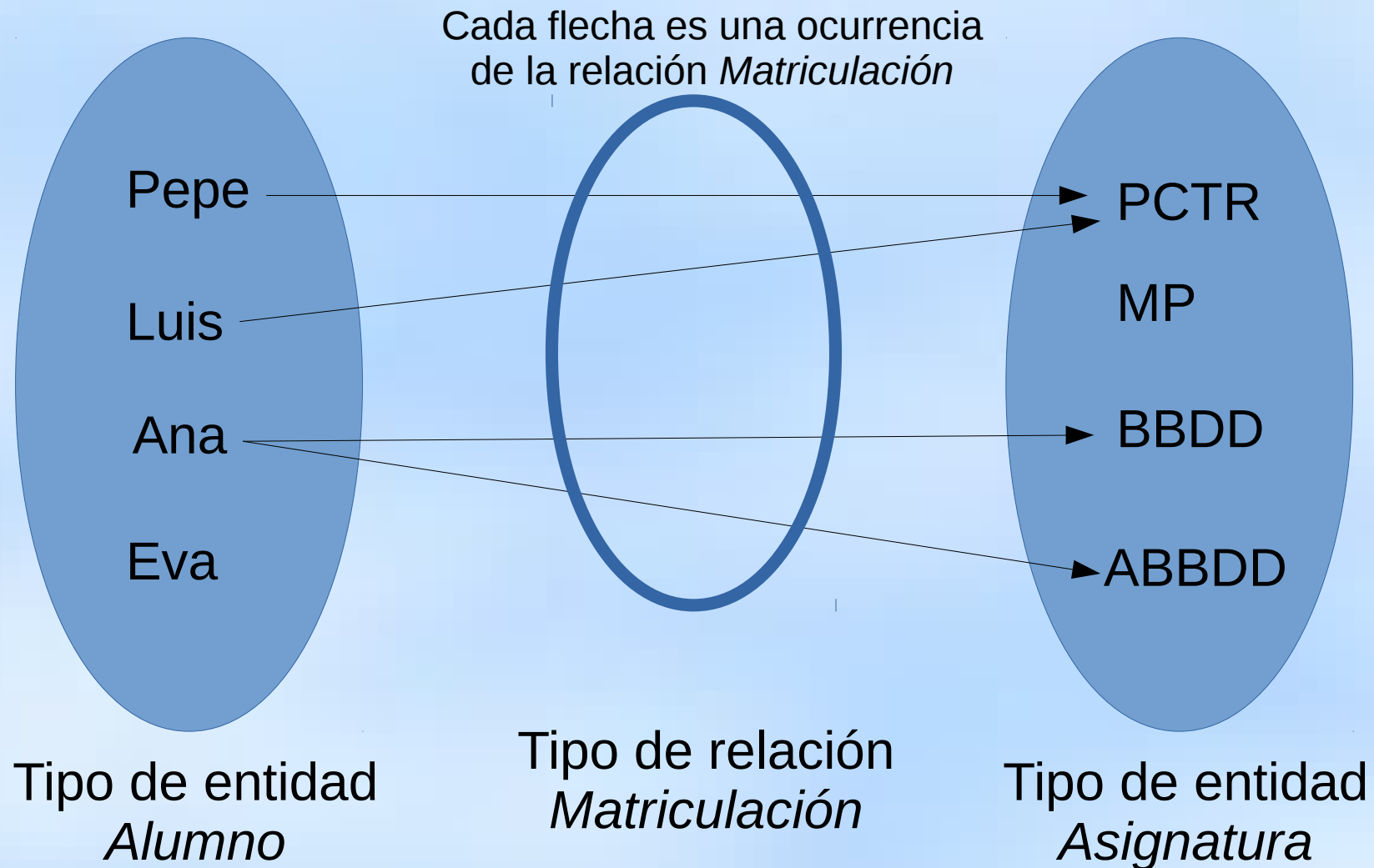
- Representación del tipo de relación *Estar matriculado* entre tipos entidades *alumno* y *asignatura*



- Este es el diagrama E/R de tipos

# Relaciones

- Ejemplo: alumnos matriculados en asignaturas

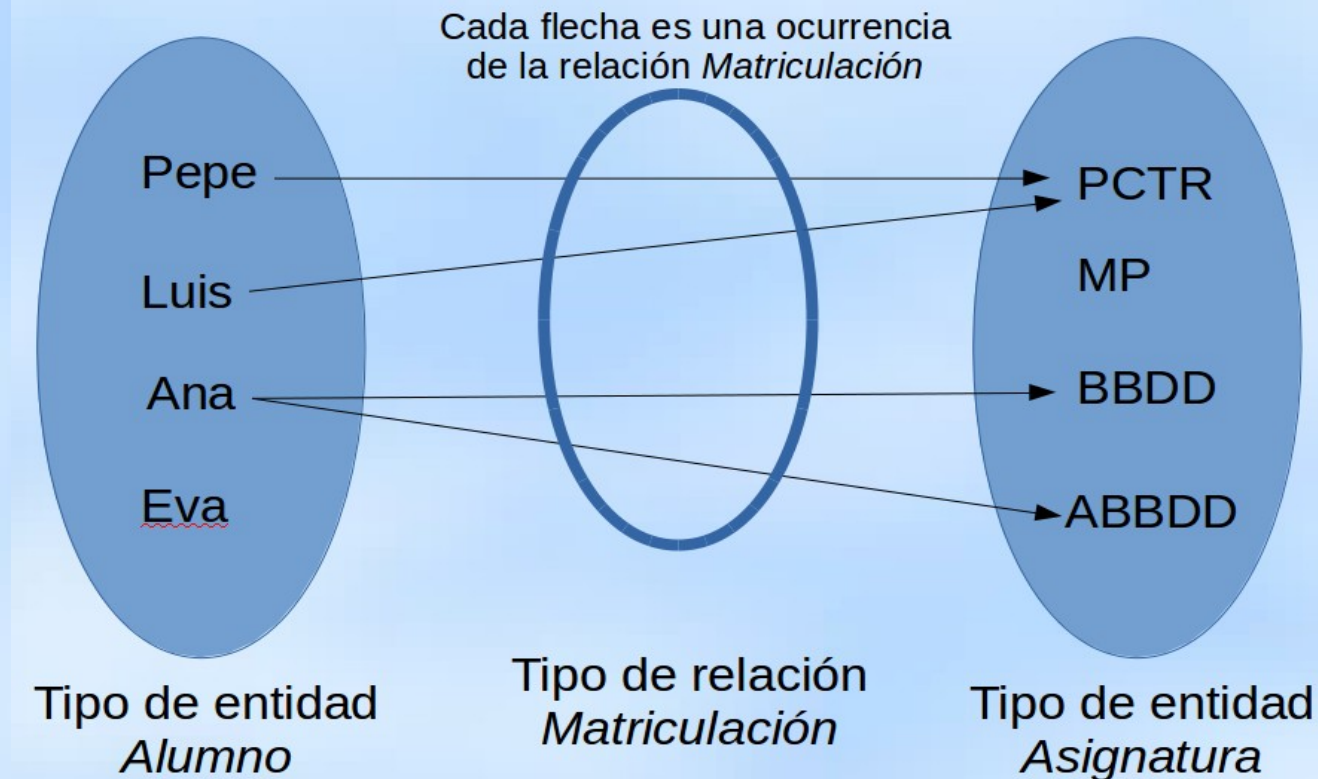




# Relaciones

- Algunas instancias para el diagrama concreto:
  - Alumnos: {3, Pepe, "Calle Sol 37"} {4, Ana, "Av. Pez 4"} ...
  - Asignaturas: {1, PCTR, 6} {8, BBDD, 6} {9, ABBDD, 6} ...
  - Estar matriculado: {3, 1} {4, 8} {4, 9} ...

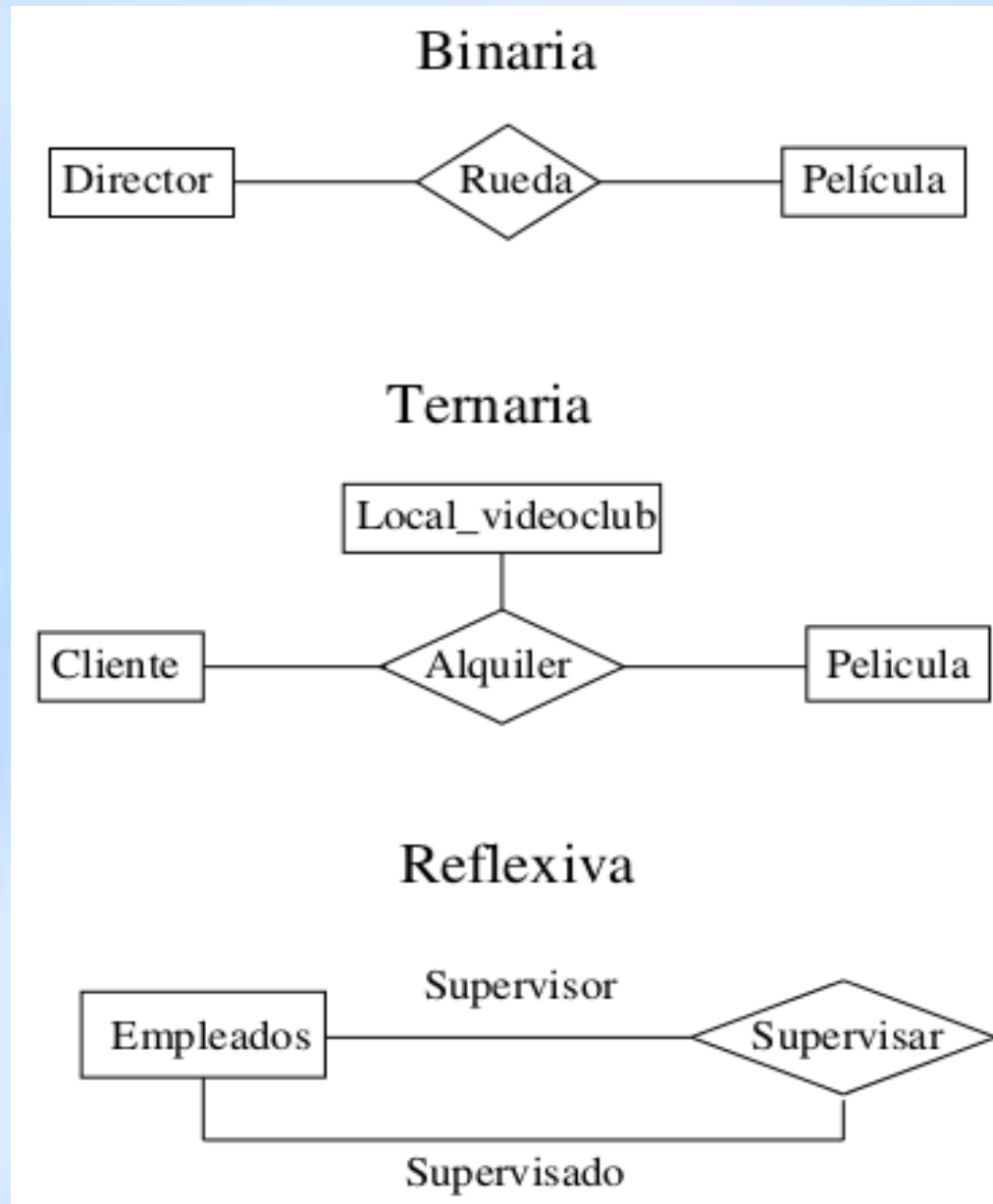
OJO, cuidado con esta representación de la relación con diagramas de Venn y flechas, porque entre dos entidades pueden producirse N relaciones distintas (habría que "colorear" o etiquetar las flechas para distinguirlas)



# Tipos de relaciones

- Grado: número de tipos de entidades que participan en el tipo de relación
  - *Binario*: intervienen dos tipos de entidades (el más frecuente)
  - *Ternario*: intervienen tres tipos de entidades
  - N-ario: menos común (evitar si es posible)
  - *Reflexivo*: interviene un único tipo de entidad
- Rol: papel que un tipo de entidad juega en un tipo de relación
  - *Alumno*: persona que se matricula
  - *Asignatura*: materia que se imparte
  - Los papeles son especialmente importantes en los tipos de relación reflexiva

# Tipo de relaciones



# Tipo de relaciones

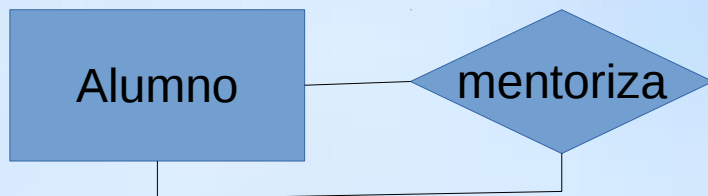
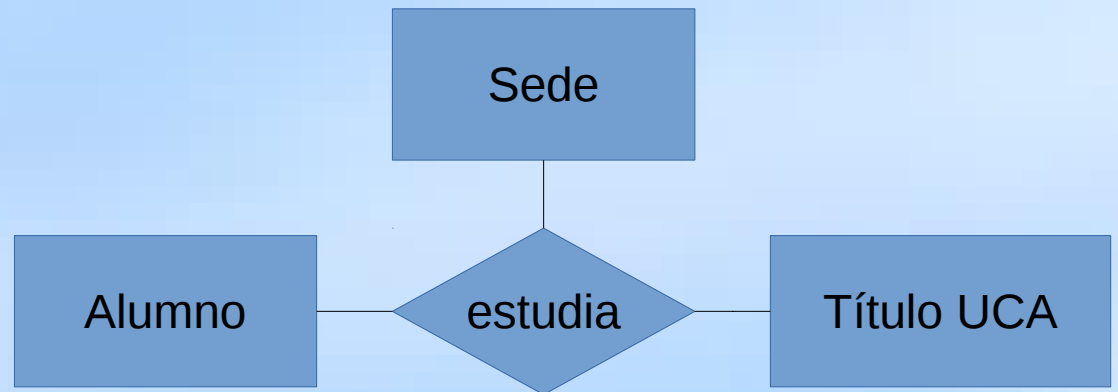
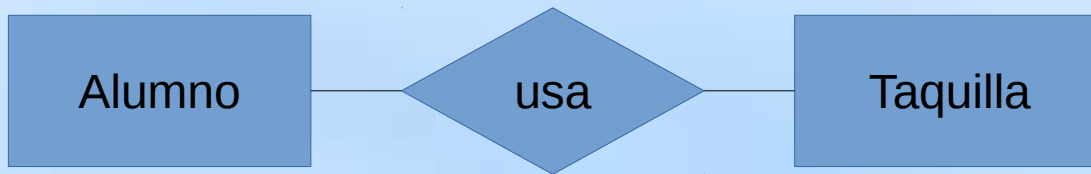
- ¿Qué grados tiene las siguientes relaciones?
  - Los alumnos estudian títulos en la UCA
    -
  - Los alumnos asisten a clases de asignaturas un edificio concretos
    -
  - Los alumnos se mentorizan entre sí
    -

# Tipo de relaciones

- ¿Qué grados tiene las siguientes relaciones?
  - Los alumnos estudian títulos en la UCA
    - Binario
  - Los alumnos asisten a clases de asignaturas un edificio concretos
    - Ternario
  - Los alumnos se mentorizan entre sí
    - Reflexivo
- Dependiendo del problema habrá cosas que se modelen como entidad o atributo

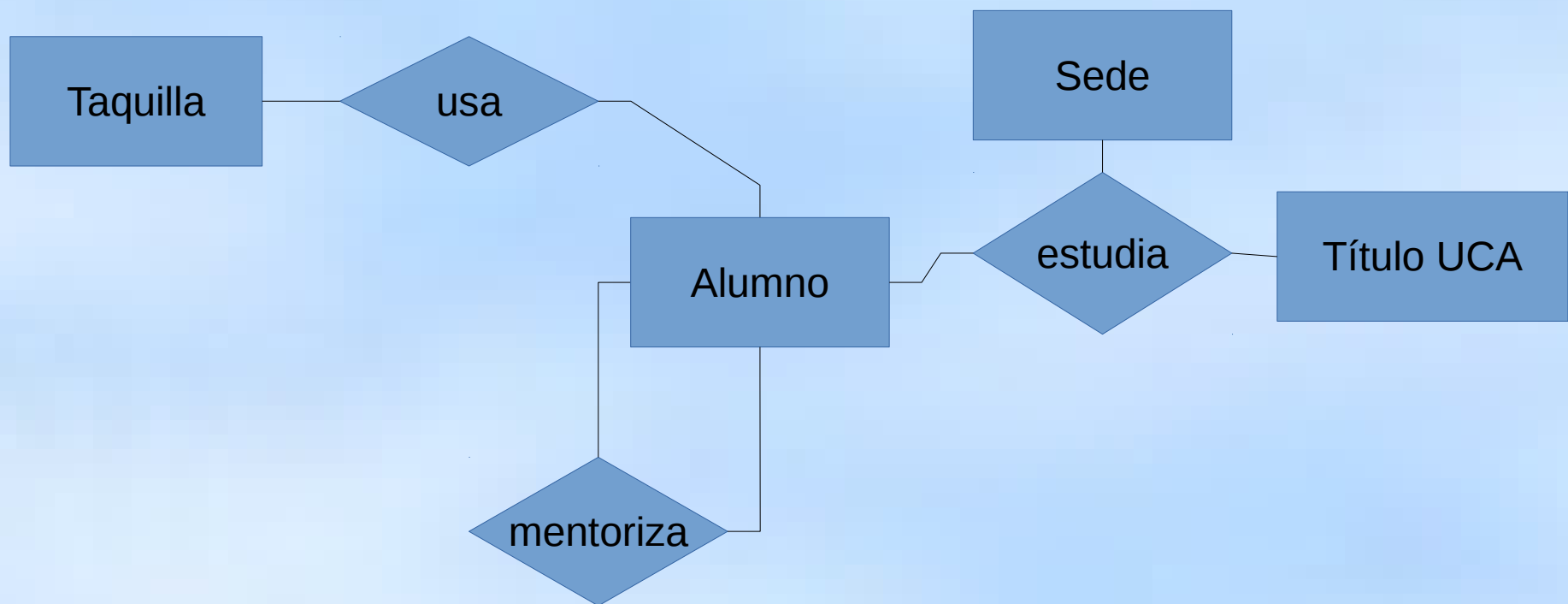
# Tipo de relaciones

- ¿Qué grados tiene las siguientes relaciones?



# Tipo de relaciones

- ¿Qué grados tiene las siguientes relaciones?



# Tipo de relaciones

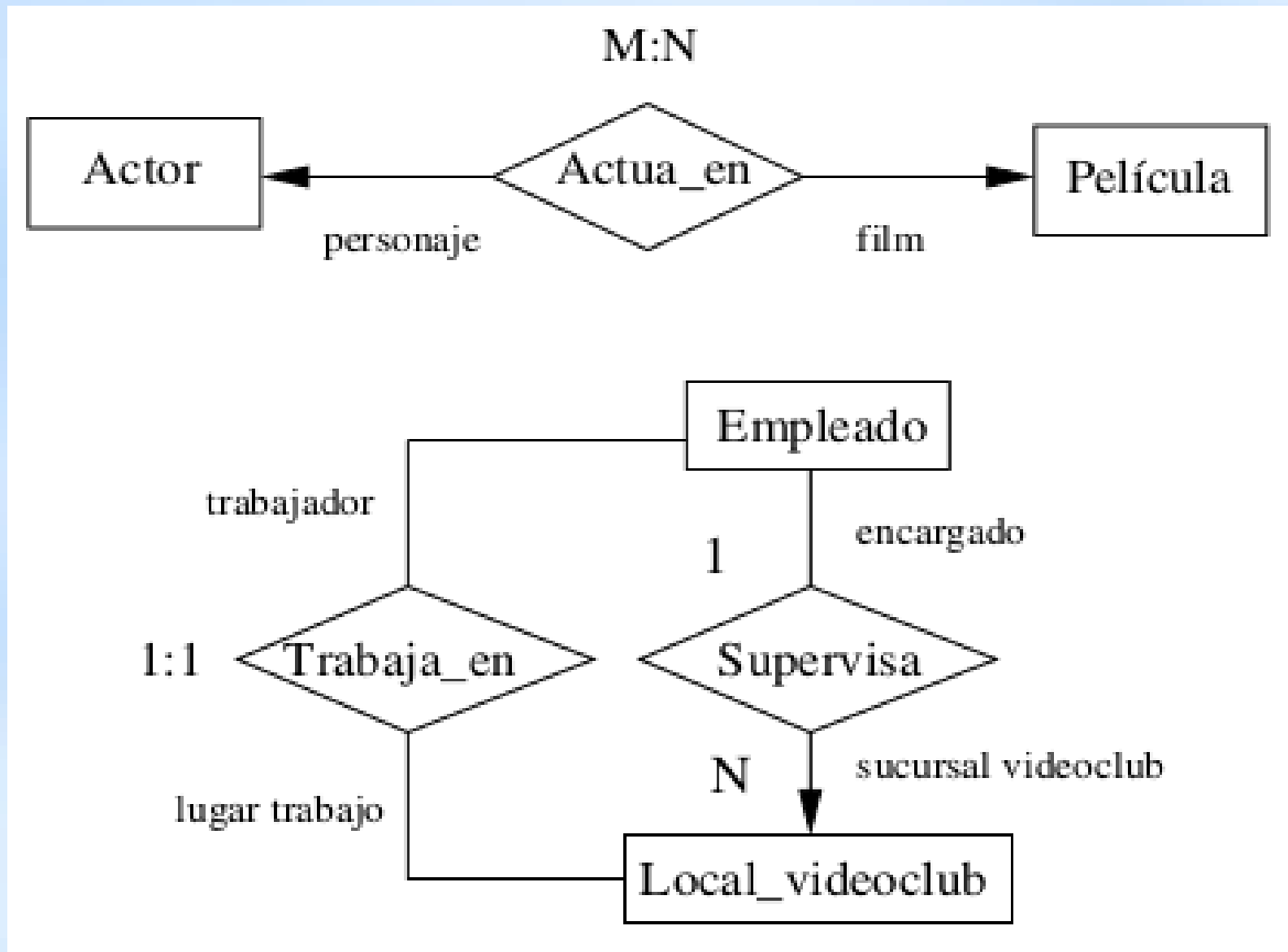
- Restricciones: limitaciones en las combinaciones de entidades que pueden participar en las relaciones
  - Cardinalidad: sobre el máximo
  - Participación: sobre el mínimo



# Tipo de relaciones

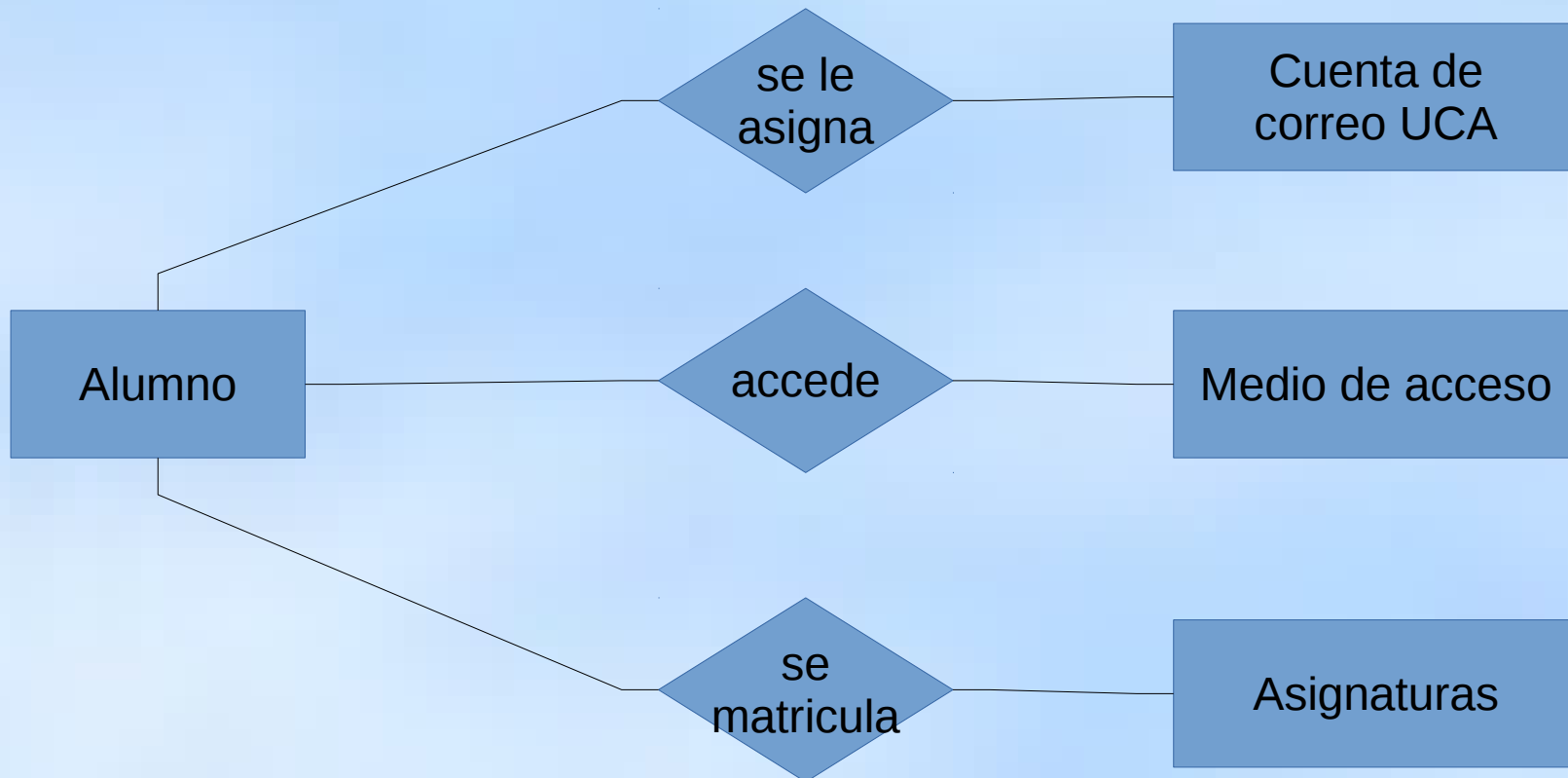
- Cardinalidad de un tipo de relación binaria (o reflexiva):
  - Número máximo de relaciones en las que puede participar una misma entidad
  - En E/R se definen tres 1:1, 1:N (o N:1), M:N
    - Se puede encontrar 2:4, 1:10, ... incluso (0-2):(1-M) fuera del estándar
      - En clase sólo se indica si el máximo es 1 o más en cada participante
    - A veces en E/R se pone una punta de flecha en los extremos M y N

# Tipo de relaciones



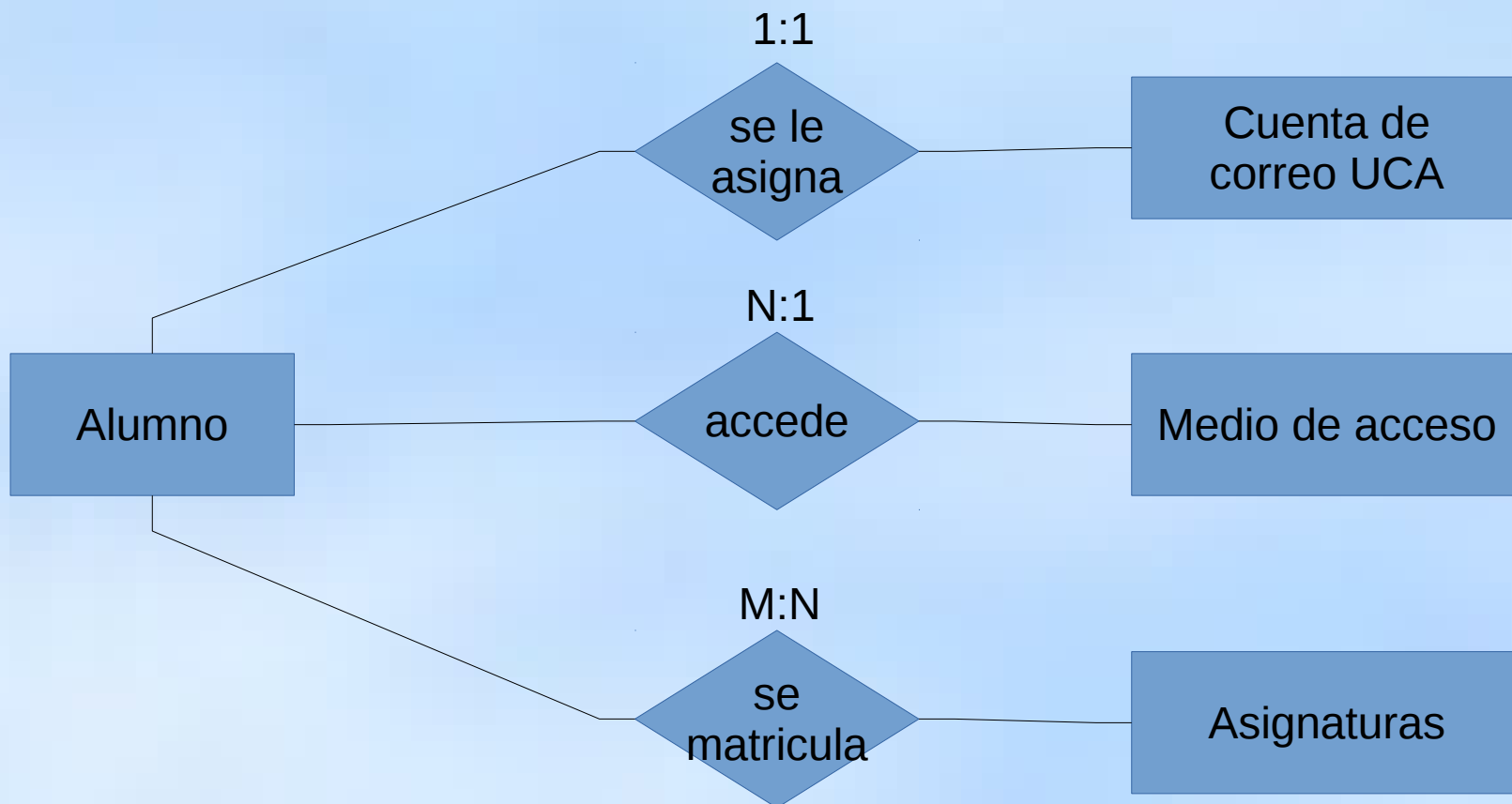
# Tipo de relaciones

- ¿Qué cardinalidades tiene las siguientes relaciones?



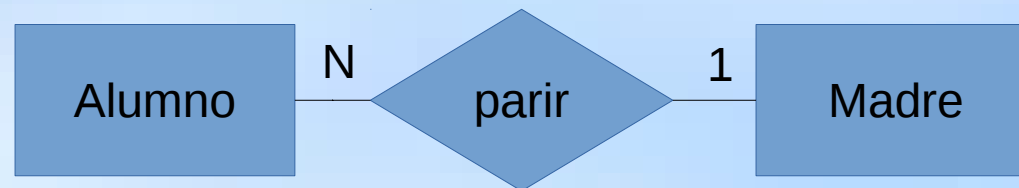
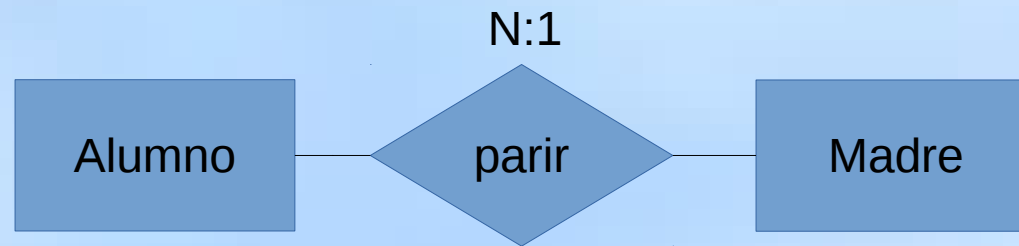
# Tipo de relaciones

- ¿Qué cardinalidades tiene las siguientes relaciones?



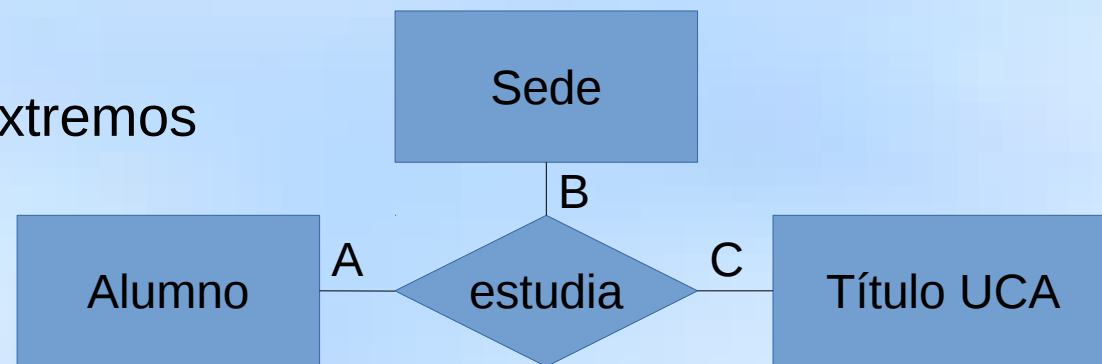
# Tipo de relaciones

- ¿Dónde va el 1 el dónde el N?
  - Ej: “Madre no hay más que una”



# Tipo de relaciones

- ¿Cómo se indica la cardinalidad en una relación ternaria (o n-aria)?
  - Se pone un valor a cada entidad participante que sería el máximo para uno de cada uno de los demás participantes
    - En el ejemplo:
      - A: máximo de alumnos que pueden estudiar un título UCA concreto en una sede concreta
      - B: máximo de sedes en las que un alumno concreto puede estudiar un título UCA concreto
      - C: máximo de títulos UCA que puede estudiar un alumno concreto en una sede concreta
  - A veces no se pone
    - Entonces es M en los n extremos



# Tipos de relaciones

- En una relación binaria  $R$  entre un tipo de entidad  $A$  con  $N_1$  ocurrencias y otro tipo  $B$  con  $N_2$  ocurrencias
  - ¿Cuántas ocurrencias de  $R$  se pueden dar como mínimo entre ellas si  $R$  tiene una cardinalidad de ...?
    - 1:1
    - 1:N
    - M:N

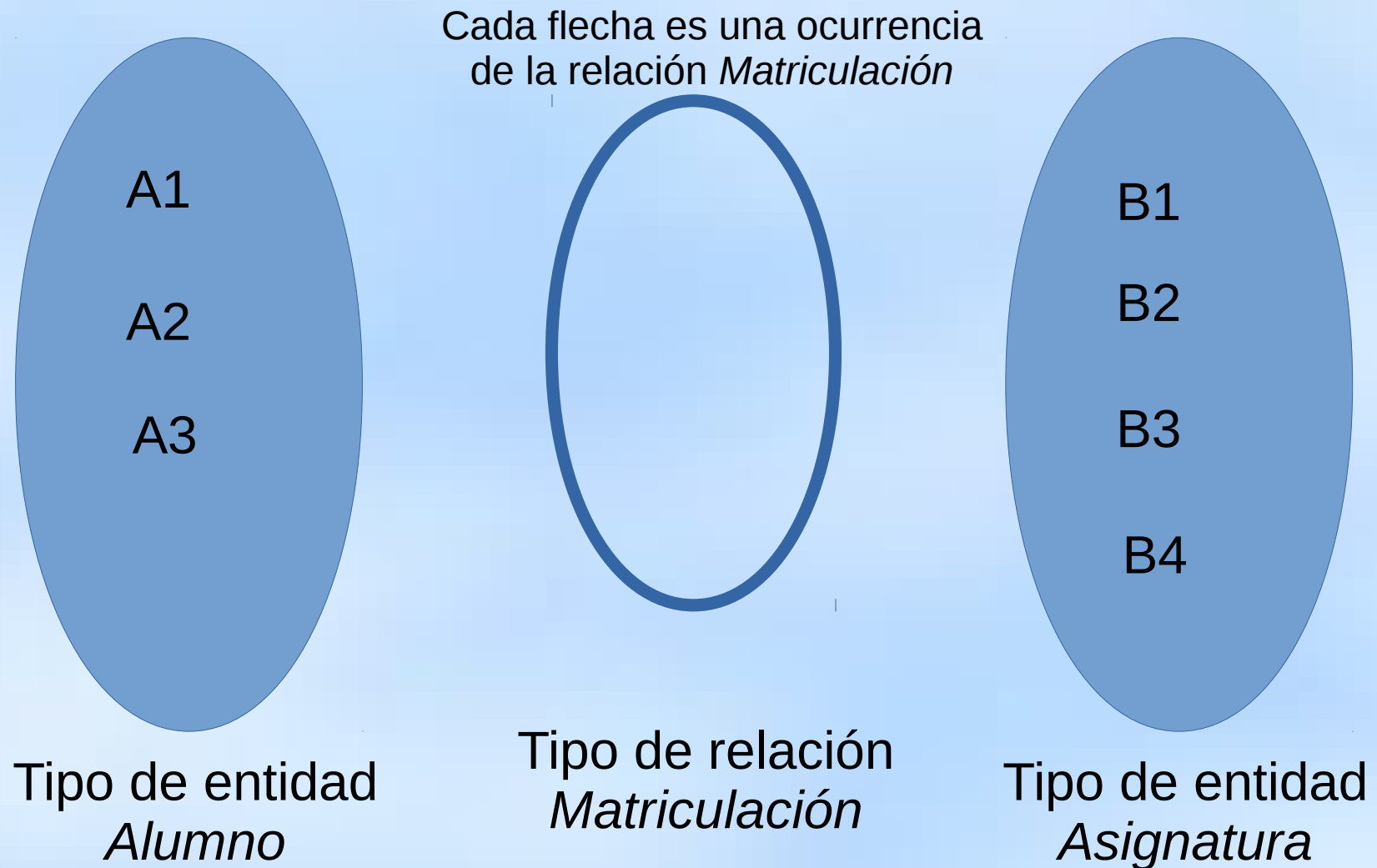
# Tipos de relaciones

- En una relación binaria  $R$  entre un tipo de entidad  $A$  con  $N_1$  ocurrencias y otro tipo  $B$  con  $N_2$  ocurrencias
  - ¿Cuántas ocurrencias de  $R$  se pueden dar como mínimo entre ellas si  $R$  tiene una cardinalidad de ...?
    - 1:1
    - 1:N
    - M:N
  - El mínimo siempre es 0
    - Se daría cuando en el modelo se pudieran relacionar instancias de  $A$  y  $B$  pero momentáneamente no lo hacen
  - ¿Y máximo?



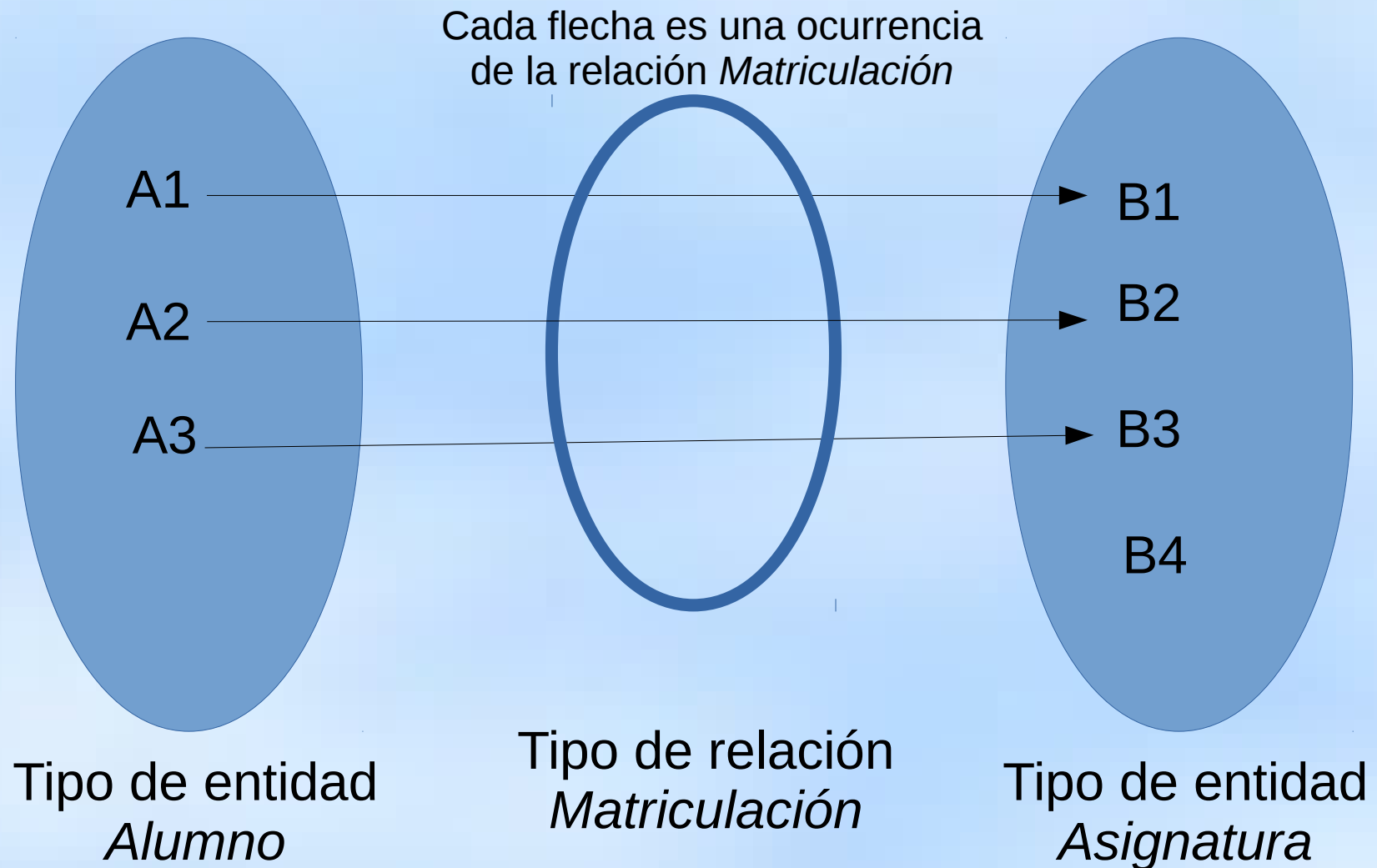
# Relaciones

- Relación 1:1, máximo = ?



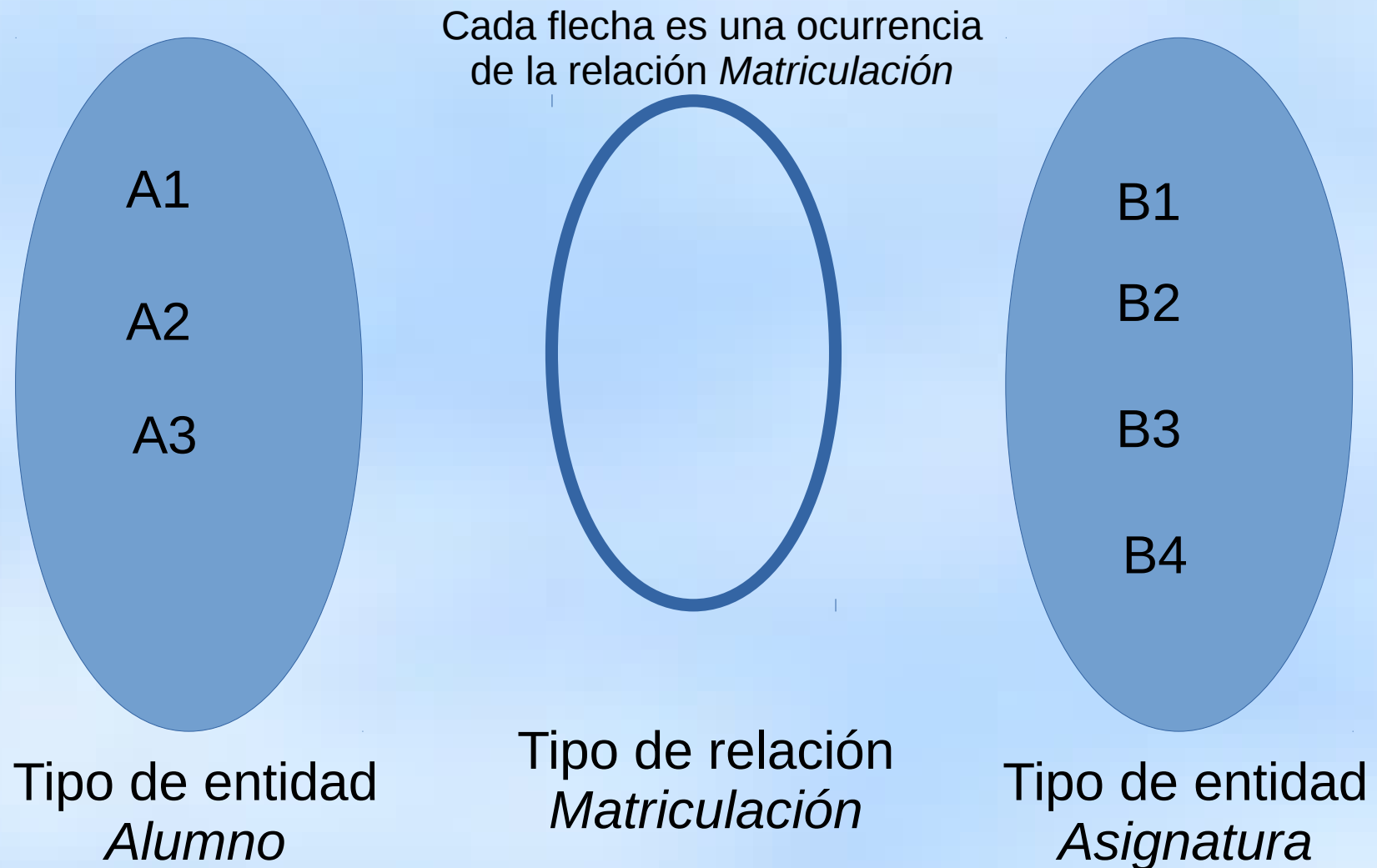
# Relaciones

- Relación 1:1, máximo =  $\min(N_1, N_2)$



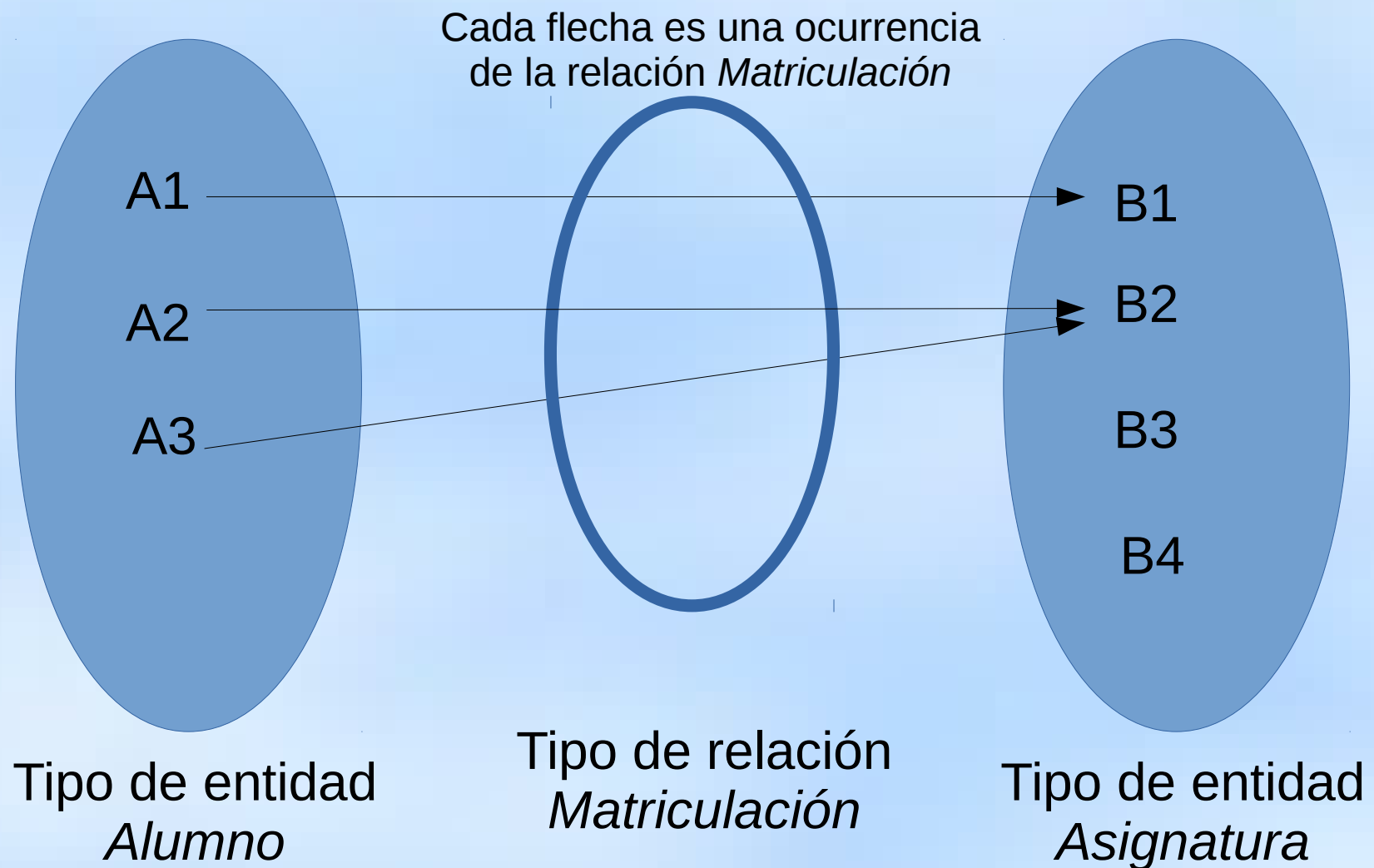
# Relaciones

- Relación N:1, máximo = ?



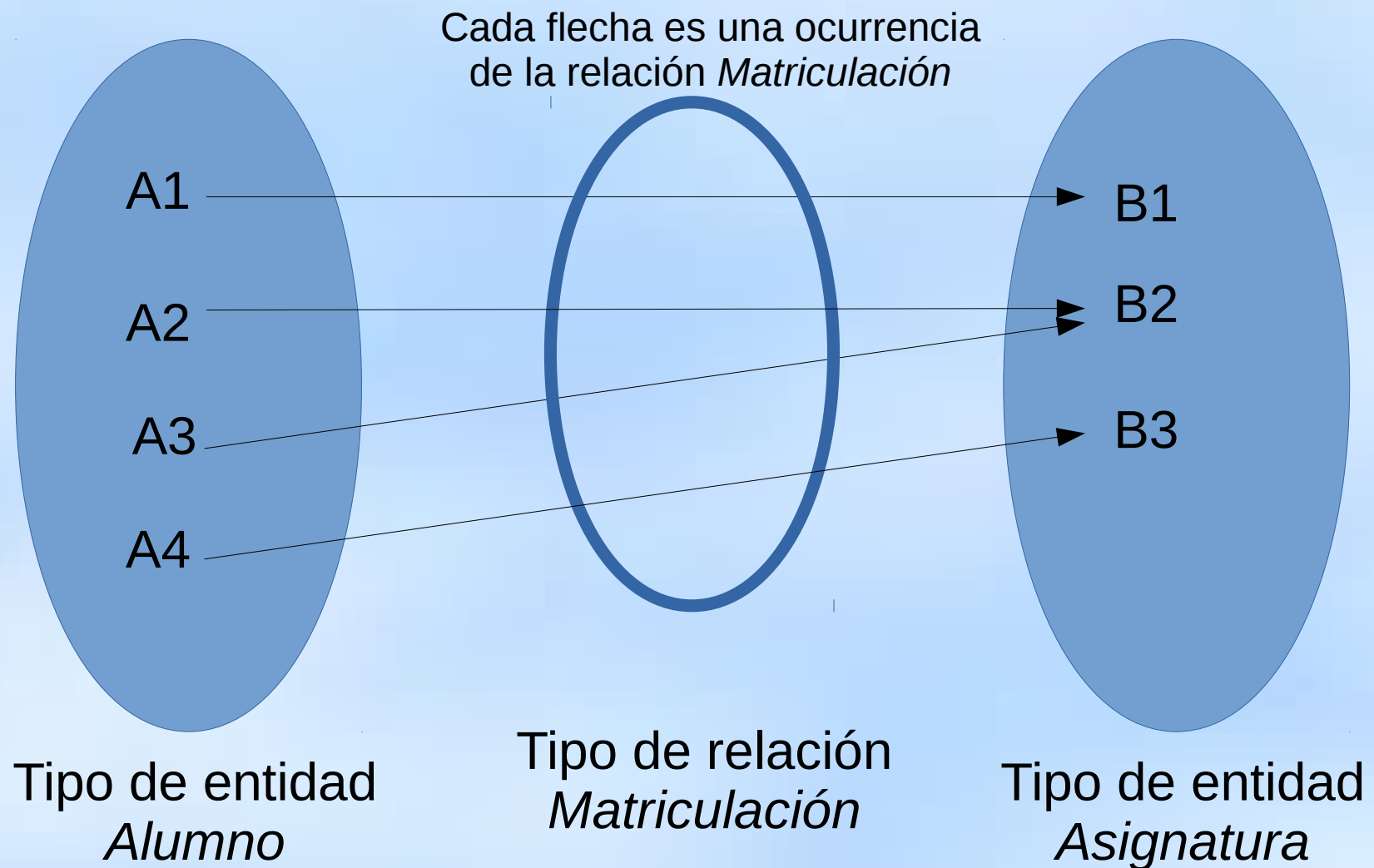
# Relaciones

- Relación N:1, máximo =  $N_1$



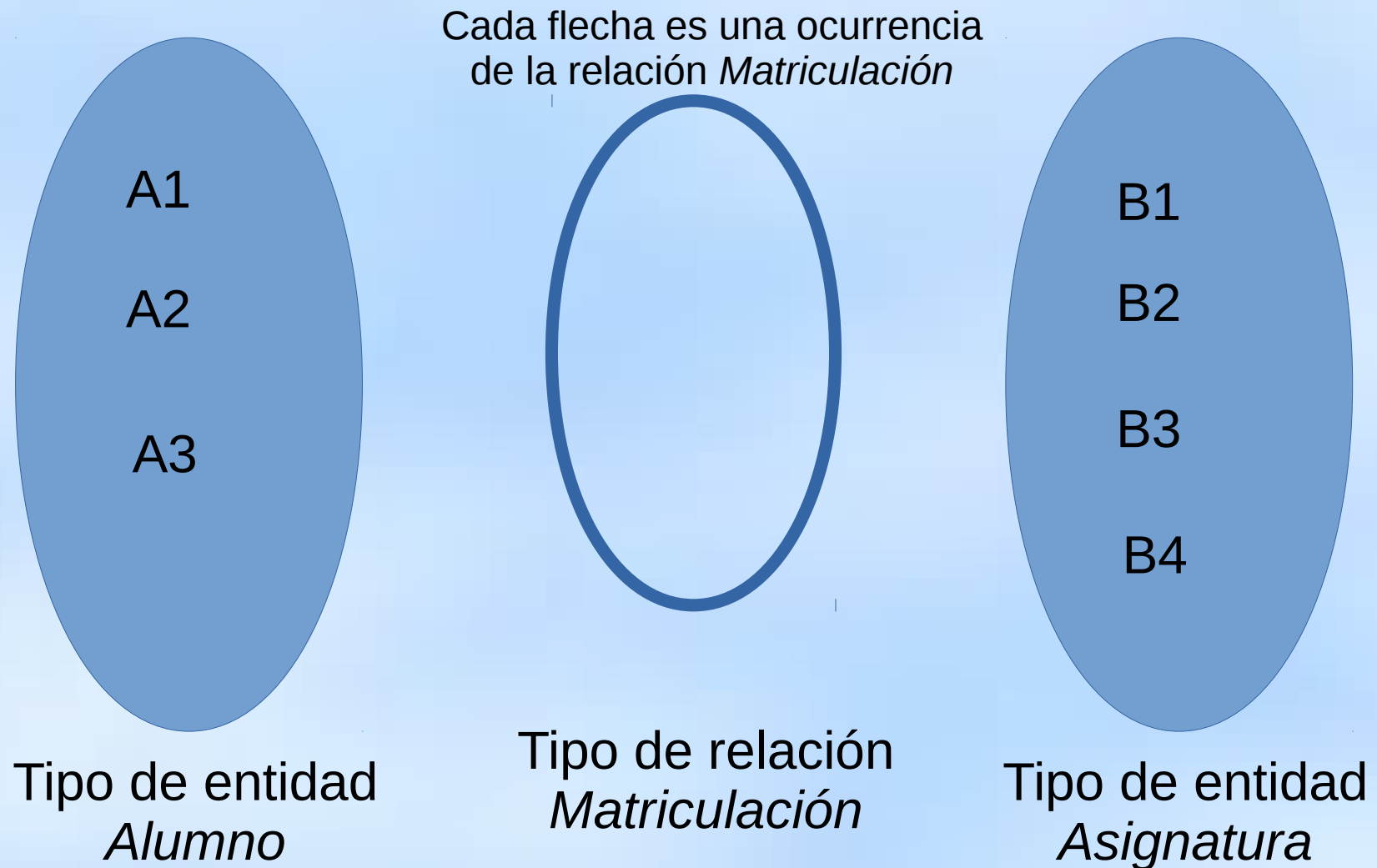
# Relaciones

- Relación N:1, máximo =  $N_1$



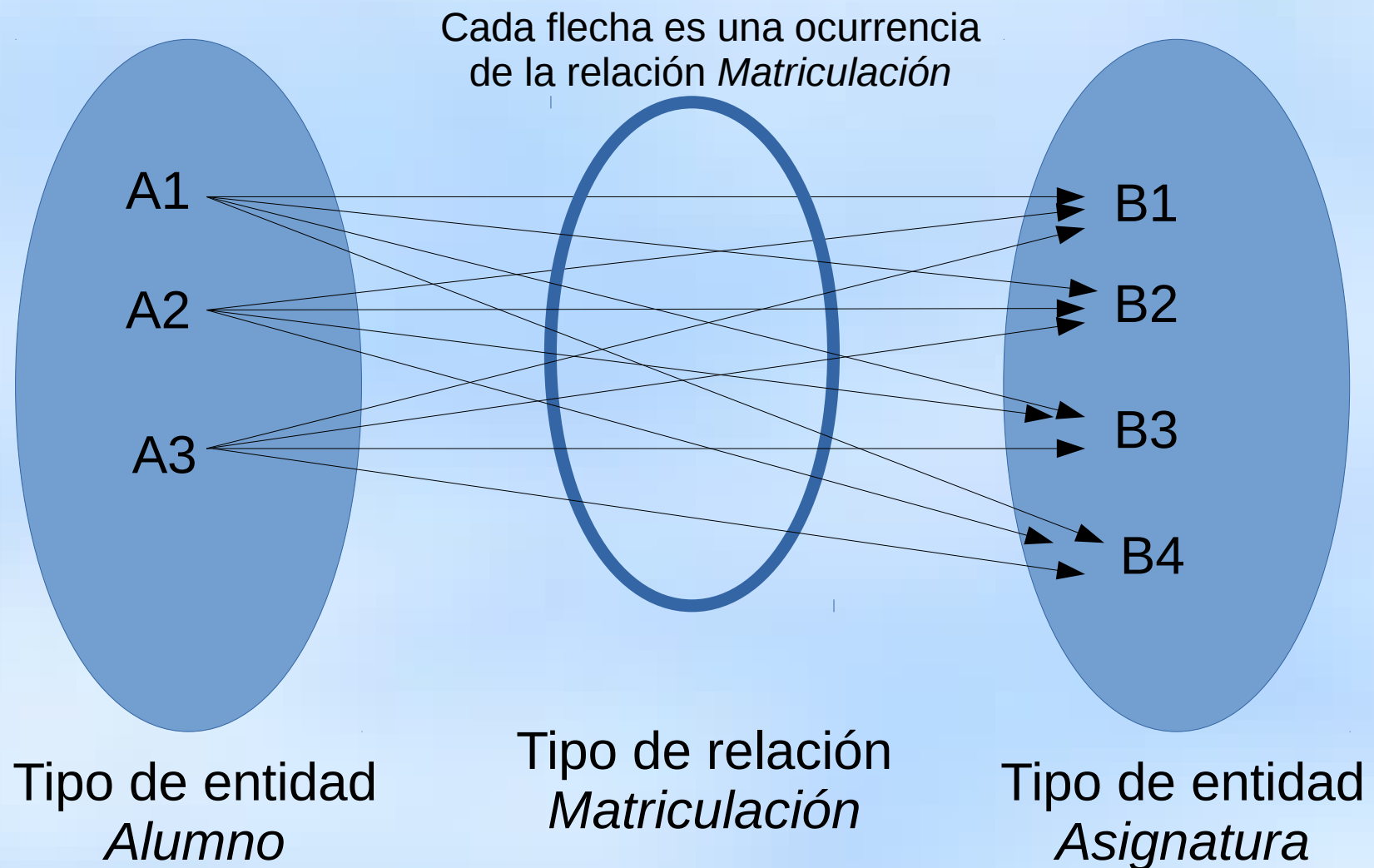
# Relaciones

- Relación M:N, máximo = ?



# Relaciones

- Relación M:N, máximo =  $N_1 * N_2$



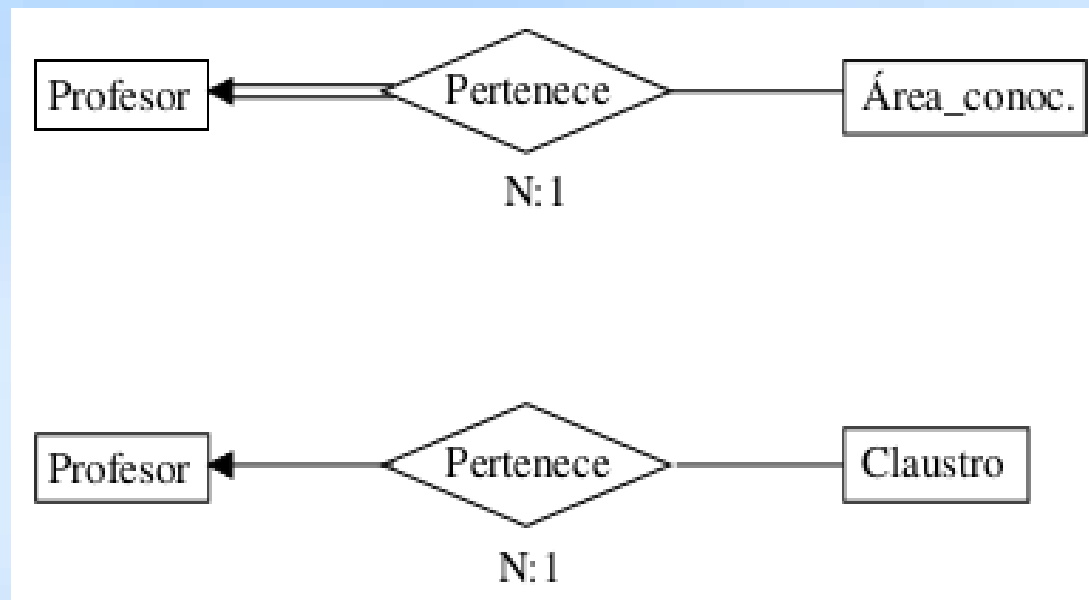
# Tipo de relaciones

- Participación:
  - Indica si la existencia de una entidad depende de que esté relacionada con otra mediante una determinada relación
  - Clases de restricciones de participación:
    - *Partición parcial*: no necesariamente toda ocurrencia de un tipo de entidades tienen que estar relacionada con ocurrencia(s) del otro tipo de entidad
    - *Partición total o de existencia*: toda entidad del conjunto de entidades debe estar relacionada con otra entidad



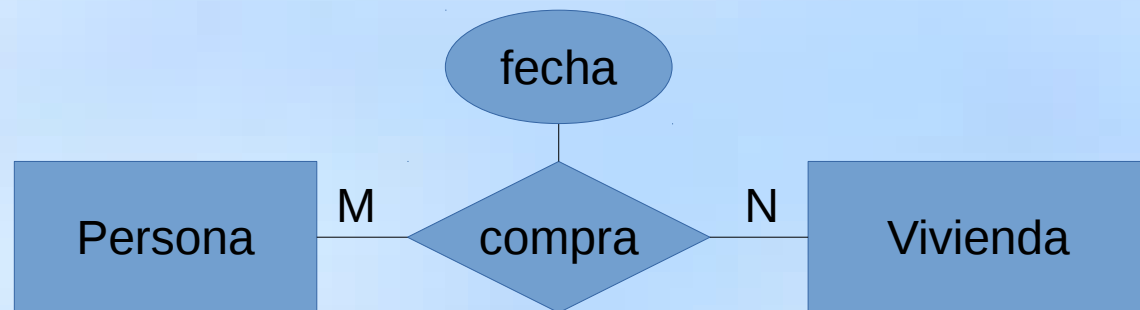
# Tipo de relaciones

- Ejemplo:
  - *Participación total o existencial*: todo profesor de universidad debe pertenecer a un área de conocimiento
  - *Participación parcial*: algunos profesores de universidad pertenecen al claustro universitario



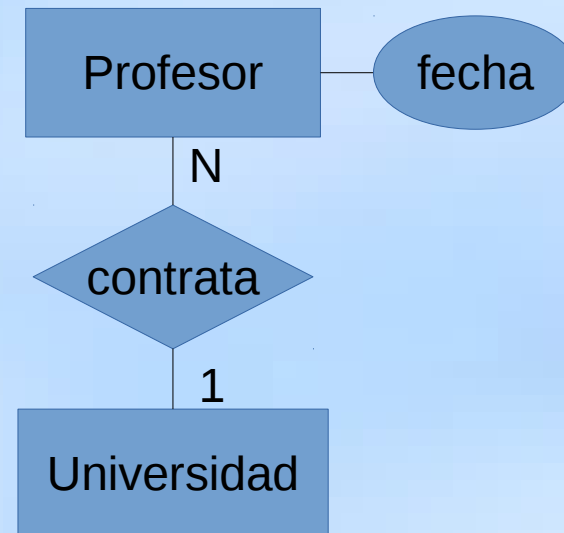
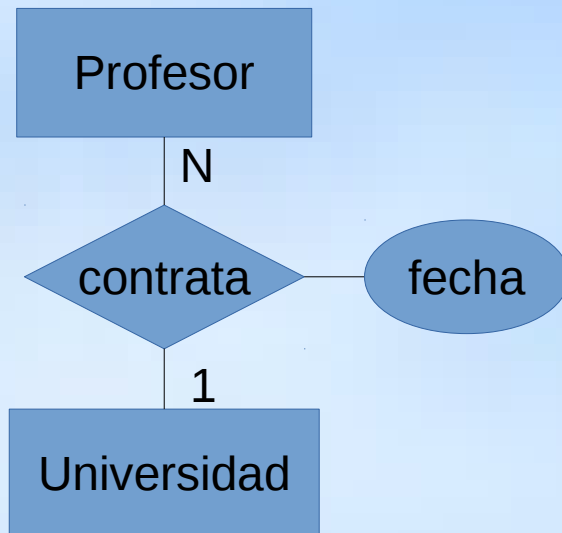
# Atributos de tipos de relación

- Los tipos de relaciones pueden tener atributos, con el mismo significado que en los tipos de entidades
- Tratamiento distinto según cardinalidad:
  - En los tipos de relaciones con cardinalidad N:M es muy probable que sus atributos estén determinados por la combinación de atributos de los tipos de entidades participantes (*no incorporables*)



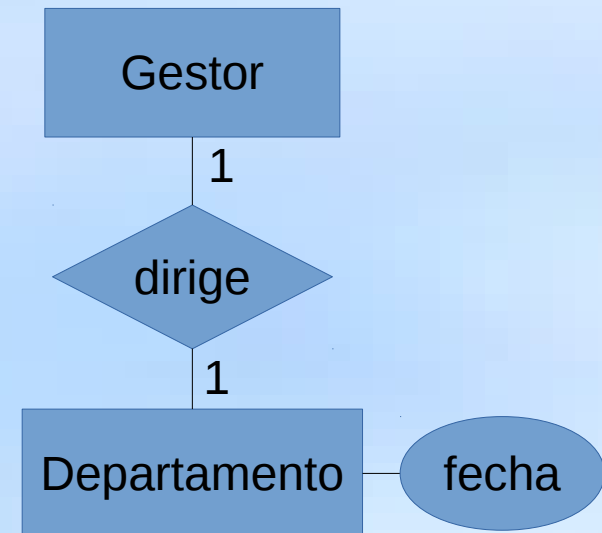
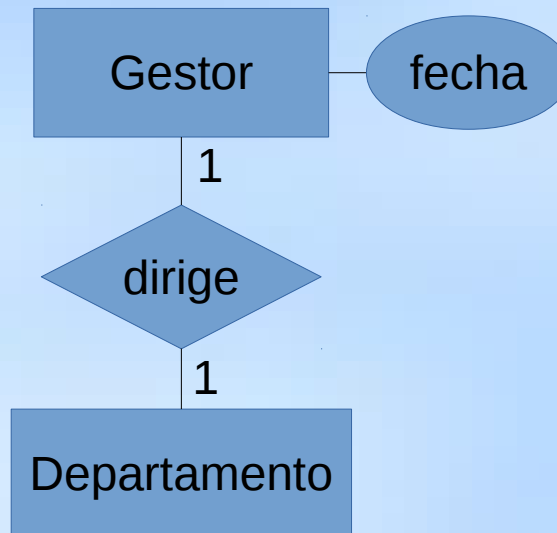
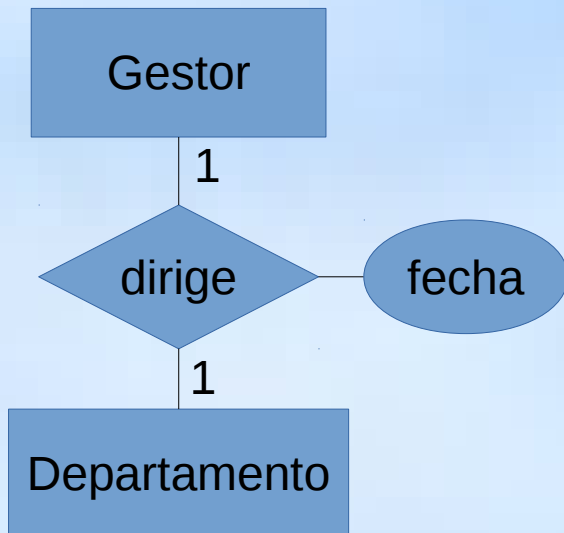
# Atributos de tipos de relación

- Tratamiento distinto según cardinalidad:
  - Los atributos de los tipos de relación 1:1 y 1:N pueden considerarse atributos de los tipos de entidades participantes:
    - En relaciones 1:N, los atributos pueden incorporarse al tipo de entidad con cardinalidad 1



# Atributos de tipos de relación

- Tratamiento distinto según cardinalidad:
  - Los atributos de los tipos de relación 1:1 y 1:N pueden considerarse atributos de los tipos de entidades participantes:
    - En relaciones 1:1 puede incorporarse a cualquier entidad



# Atributos de tipos de relación

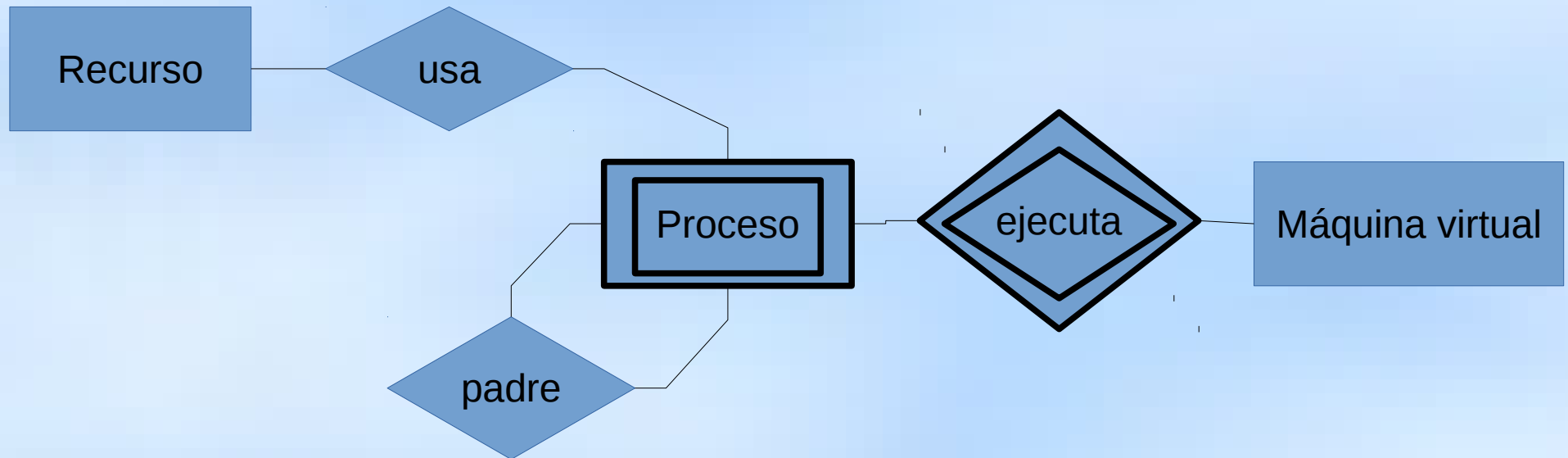
- Tratamiento distinto según cardinalidad:
  - Los atributos de los tipos de relación 1:1 y 1:N pueden considerarse atributos de los tipos de entidades participantes:
    - En relaciones 1:1 puede incorporarse a cualquier entidad

# Tipo de entidades débiles

- Tipo de entidad que no tiene suficientes atributos para formar una clave primaria (CP)
- Una ocurrencia no existe por sí misma, sino por su relación con una ocurrencia de un tipo de entidad fuerte (con CP)
- Se da en relaciones de 1:N
  - ¿Se podría dar en una 1:1?
- El tipo de relación no posee atributos descriptivos, si los hubiere formarían parte del conjunto de entidades débiles
- Se representa por un rectángulo de borde doble

# Tipo de entidades débiles

- Una entidad débil siempre tiene una restricción de participación total en la relación que le une a su entidad dominante o fuerte
  - Si no tiene clave necesita que exista “quien le dé la clave”



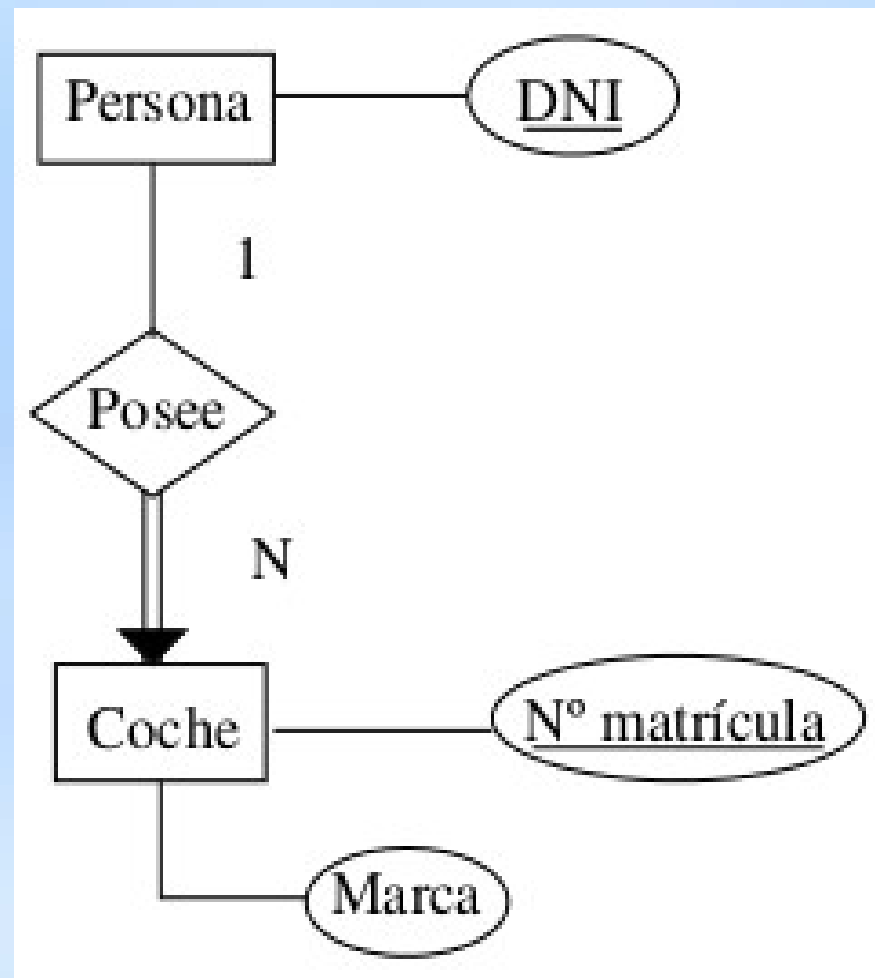
# Tipo de entidades débiles

- Ahora bien, no toda dependencia de existencia (participación total) implica que sea un tipo de entidad débil. Dependencia:
  - En *existencia*: la ocurrencia del tipo de entidad débil no puede existir sin la ocurrencia de la entidad fuerte de la que depende. Pero tiene CP
  - En *identificación*: además de la dependencia en existencia, las ocurrencias del tipo de entidad débil no se pueden identificar por si mismas
    - Es necesaria la unión de la CP del tipo de Entidad fuerte y el discriminador que tenga la débil {CP fuerte + discriminador débil} para identificarla



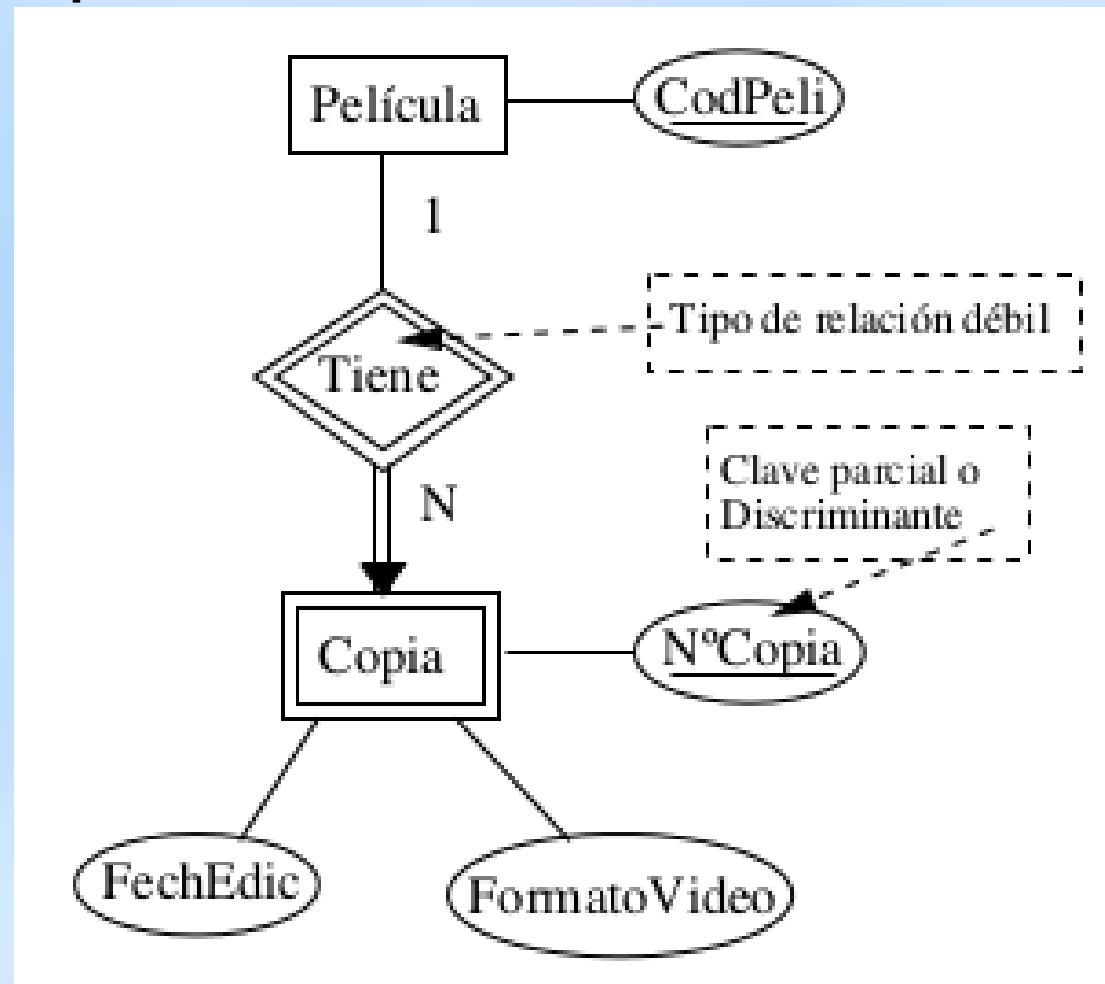
# Tipo de entidades débiles

- Ejemplo: Coche depende en existencia de Persona, pero dispone de su propia CP, luego no es débil



# Tipo de entidades débiles

- Copia es débil, pues depende en existencia de Película, y además depende en identificación
  - CP de Copia es {CodPeli, N° Copia}



# Modelo E-E/R

- El modelo E/R fue muy aceptado
- Pero la evolución de la informática (y la ingeniería del software en particular) hizo que se hicieran necesario constructores adicionales
  - Modelo Extended E/R
- El modelo E-E/R incluye
  - Herencia entre subclases/superclases
  - Generalización y especialización
  - Categoría

# Herencia

- A veces un tipo de entidad engloba a subgrupos de entidades con determinadas diferencias entre si
  - Ej: empleado puede ser jefe, administrativo, ...
- La clase que engloba es superclase, y cada clase englobada es subclase
- Importante: todo miembro de una subclase lo es también de la superclase, pero con un *rol*
  - Pero no todo miembro de la superclase lo tiene que ser de alguna subclase (puede ser de 0, 1 o más)

# Herencia

- De aquí sale el concepto de herencia de tipos
  - Recordamos que el tipo de una entidad está definido por sus atributos y relaciones
- Las subclases *heredan* todos los atributos y relaciones de la superclase
  - Y sus miembros pueden tener o no valores y participar o no, como los miembros de la superclase
  - Además, las subclases pueden tener atributos y relaciones específicas (*locales*)

# Generalización y especialización

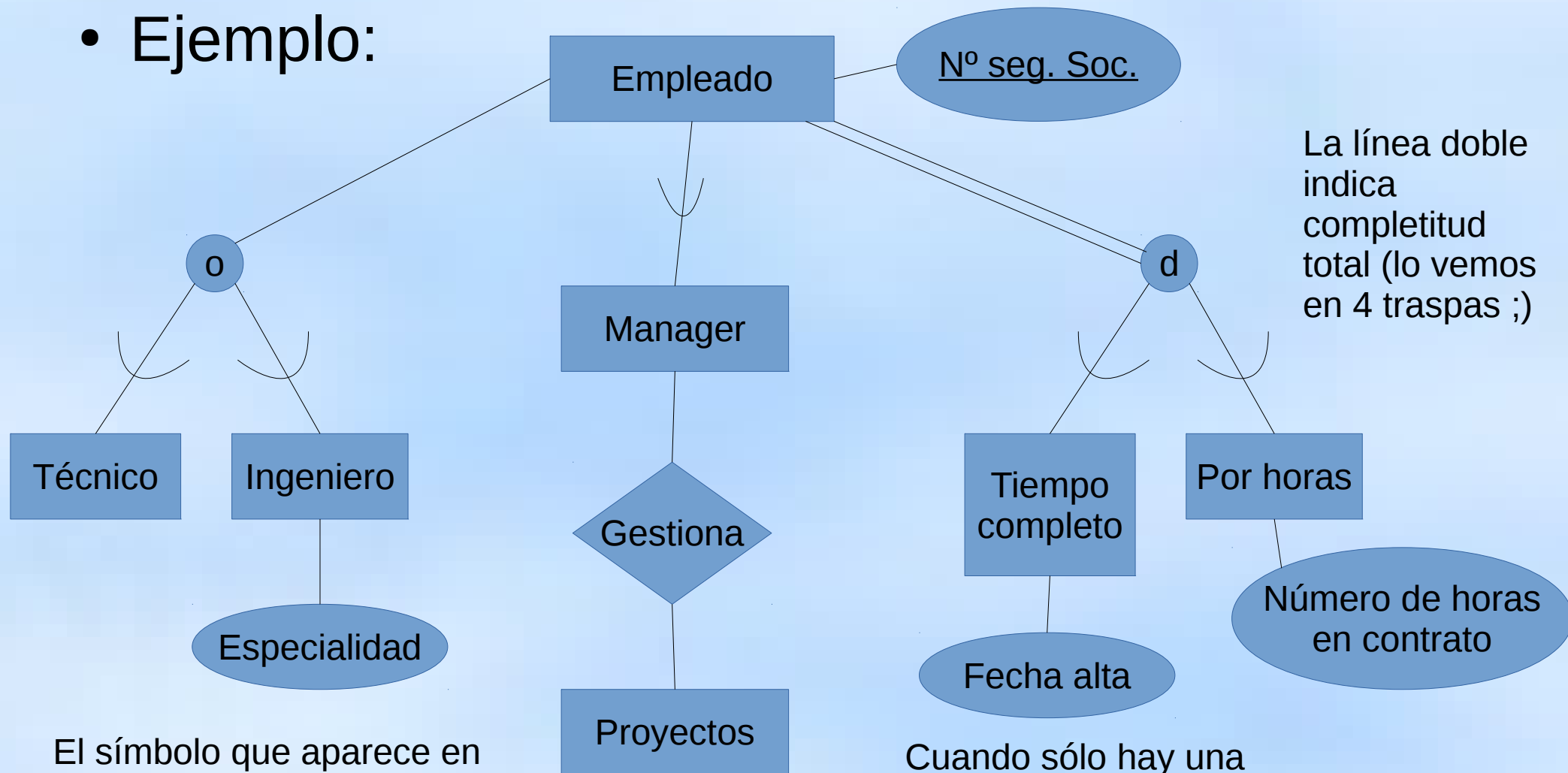
- Son tipos de relaciones que articulan el mecanismo de herencia mediante una características diferenciadora entre subclases
  - Una misma superclase puede tener varias especializaciones con distintas características
- El conjunto de entidades A es una especialización del conjunto de entidades B si

$$\forall a \in A \Rightarrow a \in B$$

El conjunto de entidades A está incluido en el conjunto de entidades B

# Generalización y especialización

- Ejemplo:



El símbolo que aparece en las conectoras es el del "subconjunto"

Cuando sólo hay una subclase de un tipo no hace falta poner el círculo

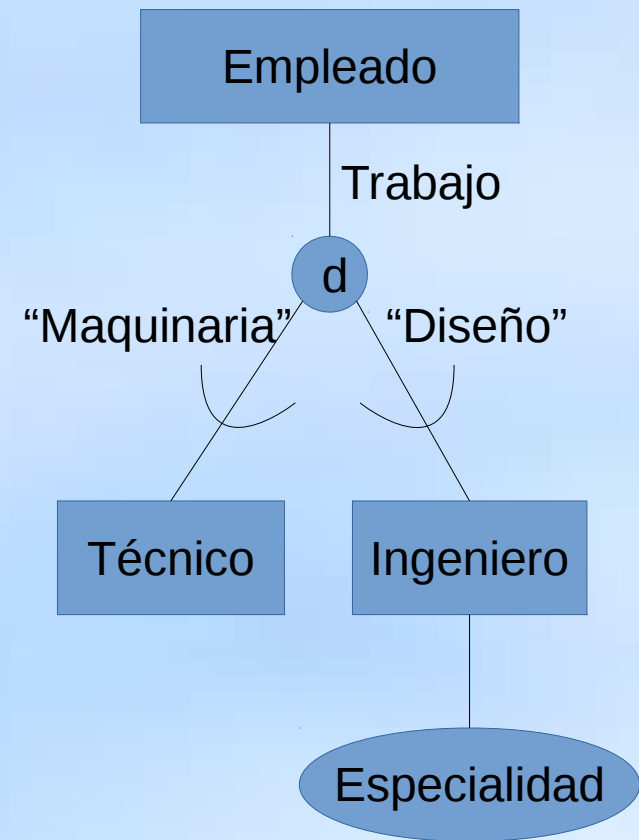
# Generalización y especialización

- La generalización es relación la inversa de la especialización
  - Según el aspecto a destacar se usa una o otra
- La generalización/especialización admiten varias restricciones:
  - Subclases definidas por atributos
  - Pertenencia a varias subclases
  - Completitud
  - Unión / categoría



# Generalización y especialización

- Subclases definidas por atributos
  - Definen a qué subclase pertenece cada instancia mediante el valor de un atributo
  - Algunas notaciones también admiten predicados
  - Si todas subclases tienen valor para dicho atributo se llama:
    - Atributo definidor
    - Especialización por atributo
  - Caso contrario: *user-defined*



# Generalización y especialización

- Pertenencia a varias subclases
  - *Restricción de disjuntas*: una entidad sólo puede pertenecer a una subclase como máximo
    - Se indica con una “**d**” en el círculo
    - Las subclases definidas por atributos siempre son disjuntas
      - ¿Y las definidas por predicado?
  - *Restricción de posible solapamiento*: una entidad puede pertenecer a varias subclases
    - Se indica con una “**o**” en el círculo
    - Es el comportamiento por defecto si no se indica nada

# Generalización y especialización

- Completitud: indica si toda entidad de la superclase tiene que ser de una subclase
  - *Total*: toda entidad de la superclase tiene que pertenecer necesariamente a una subclase
    - Se indica con una línea doble
  - *Parcial*: pueden existir entidades de la superclase que no pertenezcan a una subclase
- La completitud y la pertenencia a varias subclases son independientes, pudiendo darse las cuatro combinaciones posibles

# Generalización y especialización

- Consideraciones de insertado/borrado:
  - Insertar una entidad en una superclase de una especialización total implica ...

# Generalización y especialización

- Consideraciones de insertado/borrado:
  - Insertar una entidad en una superclase de una especialización total implica ...
    - Insertarla en al menos una subclase
  - Insertar una entidad en una superclase de una generalización definida por atributo/predicado implica

# Generalización y especialización

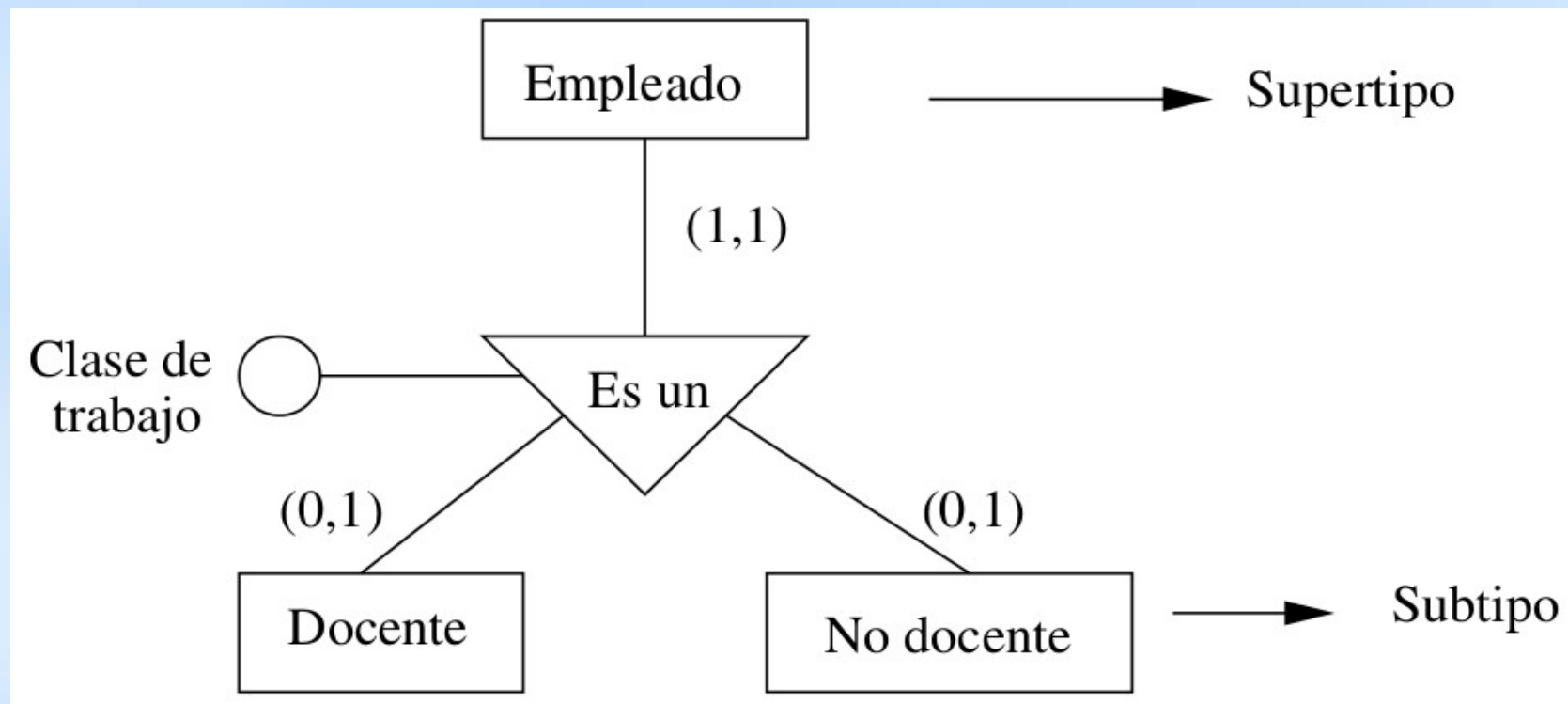
- Consideraciones de insertado/borrado:
  - Insertar una entidad en una superclase de una especialización total implica ...
    - Insertarla en al menos una subclase
  - Insertar una entidad en una superclase de una generalización definida por atributo/predicado implica
    - Insertarla en todas las subclases definidas por atributo/predicado que se cumpla
  - Borrar una entidad de una superclase implica

# Generalización y especialización

- Consideraciones de insertado/borrado:
  - Insertar una entidad en una superclase de una especialización total implica ...
    - Insertarla en al menos una subclase
  - Insertar una entidad en una superclase de una generalización definida por atributo/predicado implica
    - Insertarla en todas las subclases definidas por atributo/predicado que se cumpla
  - Borrar una entidad de una superclase implica
    - Borrarla de todas las subclases a las que pertenezca
  - ...

# Generalización y especialización

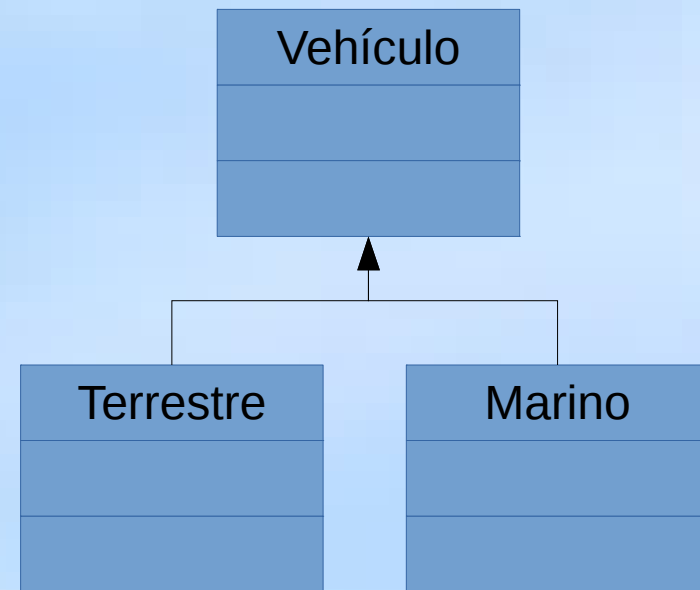
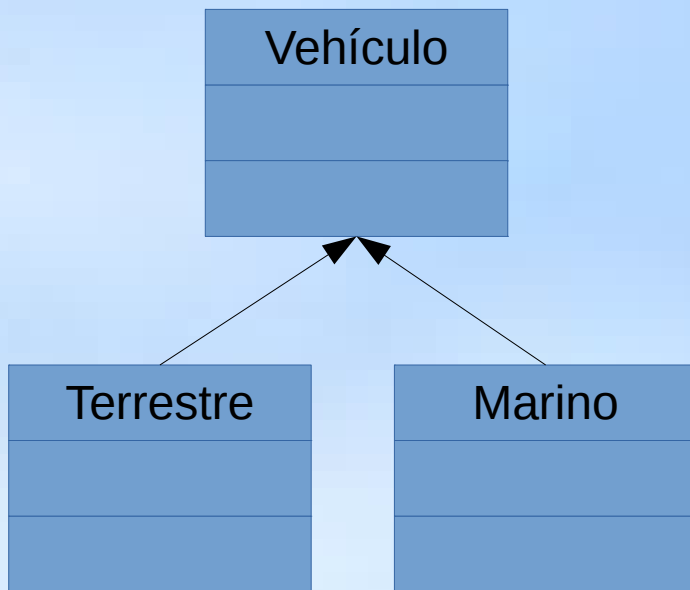
- La notación que hemos visto son las que usaremos en clase y en los exámenes
- La notación que hemos visto no es estándar, hay otras notaciones alternativas





# Generalización y especialización

- La notación que hemos visto no es estándar, hay otras notaciones alternativas: UML (x2)

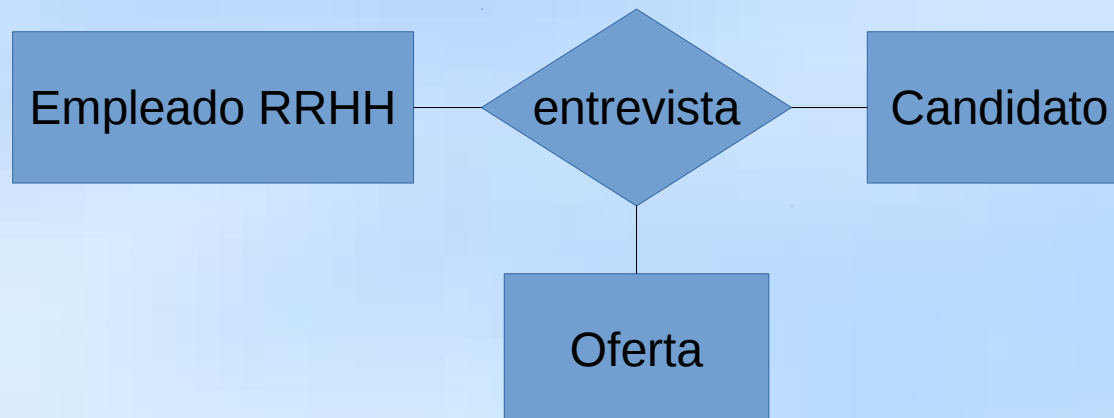


# Generalización y especialización

- Las subclases de una especialización pueden ser superclases de otras especializaciones
  - Se da una herencia de atributos y relaciones de todos sus ancestros
- Restricciones:
  - *Jerarquía de especialización*: cada clase sólo puede ser subclase de una especificación
  - *Cuadrícula de especialización: (lattice)* cada clase puede ser subclase de varias especificaciones (*shared subclass*)
    - Recibe herencia múltiple

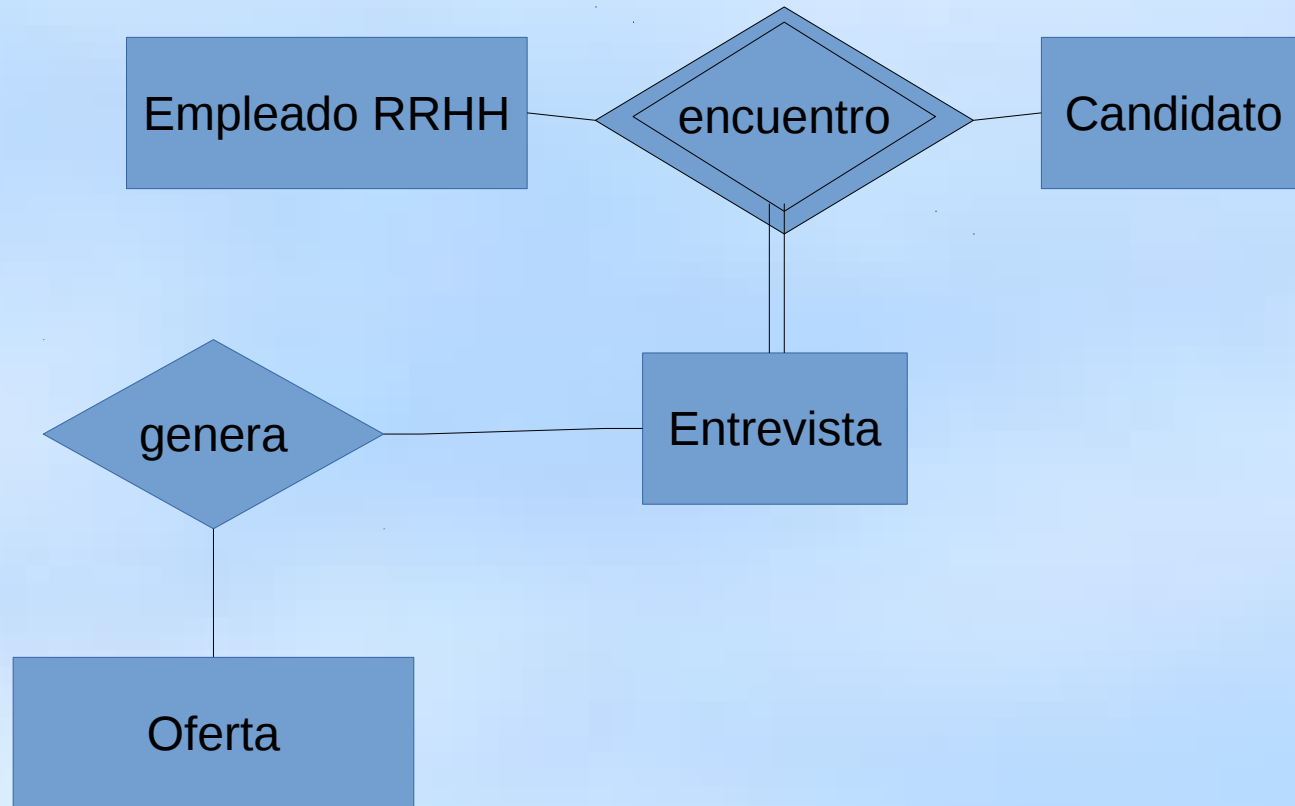
# Categoría: agregación

- La *categoría* simplifica determinadas problemáticas del modelo E/R
  - Ej: en mi empresas hacemos entrevistas de trabajo, que pueden terminar en una oferta formal
    - La solución siguiente no me vale, porque implicaría que toda entrevista termina en oferta :(



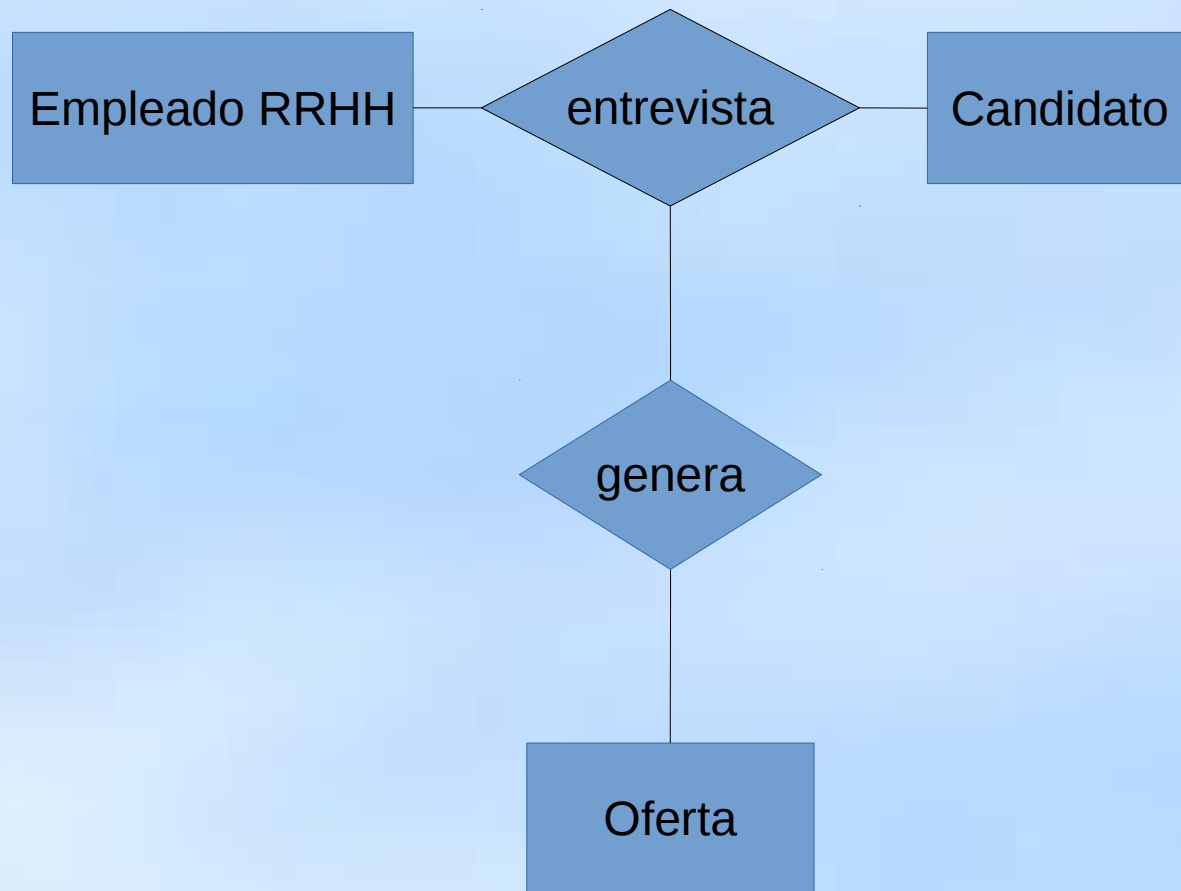
# Agregación

- Se podría hacer así:



# Agregación

- Lo ideal sería lo siguiente, pero no es válido:

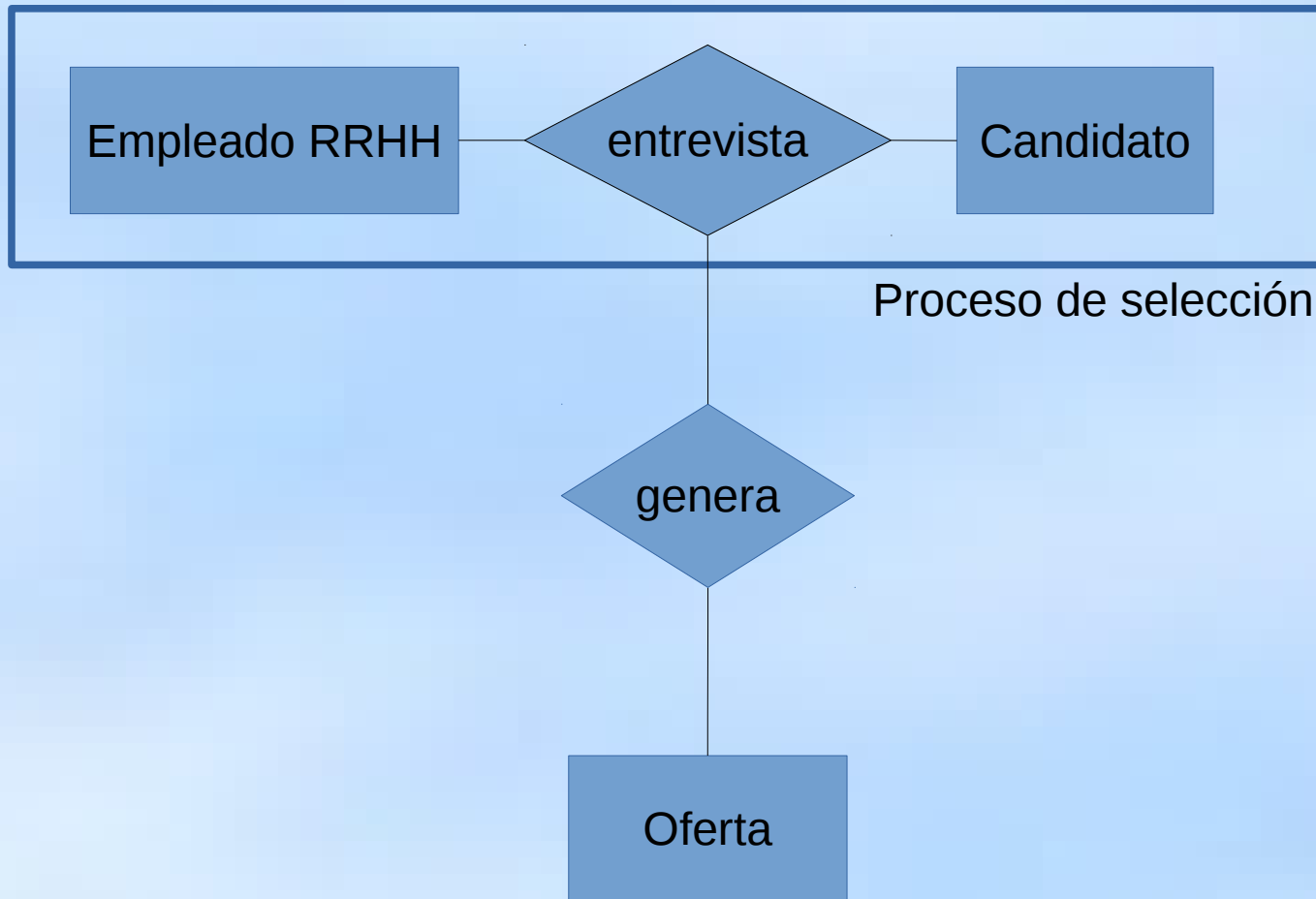


# Agregación

- Expresar relaciones entre relaciones o relaciones entre relaciones y conjuntos de entidades
- Se puede considerar la agregación como un tipo de entidad genérica sin especificar su estructura interna; como una «caja negra» donde sólo debemos conocer las CP de los tipos de entidades a los que integra
  - O una abstracción a través de la cual las relaciones se tratan como tipos de entidades de un nivel más alto
- Permite combinar varios tipos de entidades para formar un tipo agregado de nivel superior. Útil cuando el tipo de entidad agregado debe relacionarse con otros tipos de entidades.

# Agregación

- La agregación nos permite hacer lo siguiente:



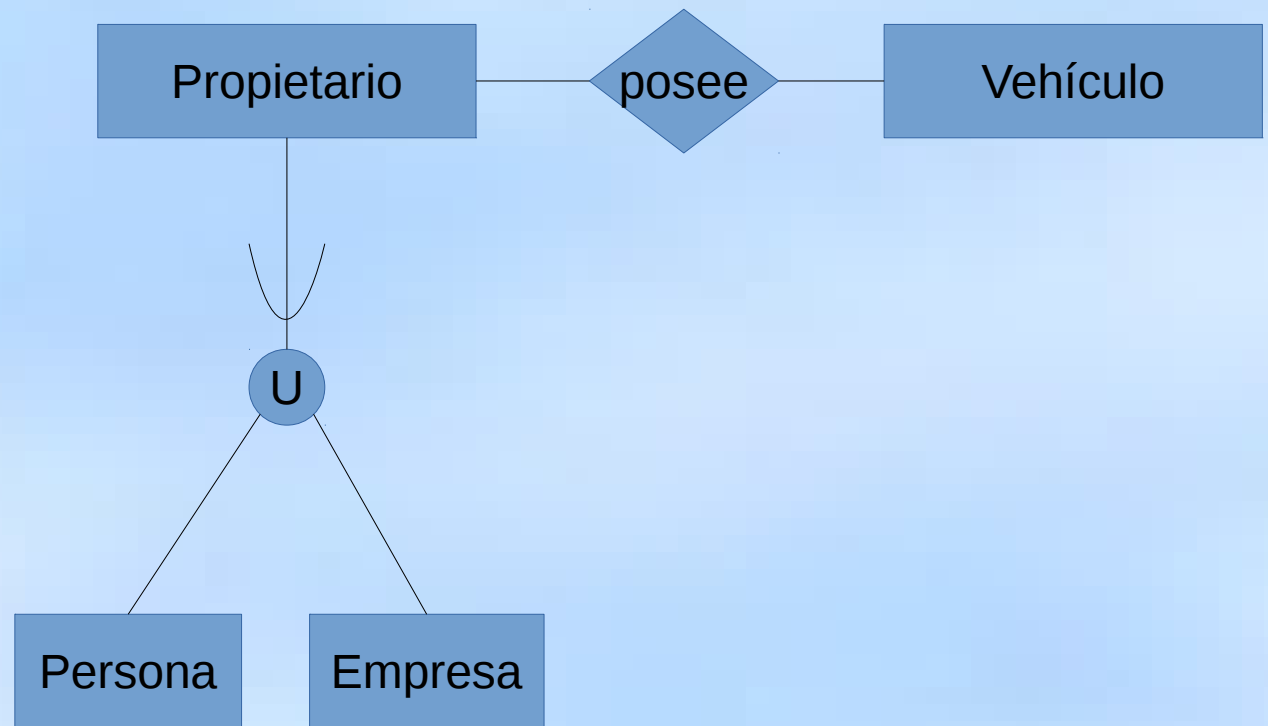
# Agregación vs asociación

- A veces se admite el modo de asociación
- Cuando en la agregación un miembro se borra el conjunto deja de tener sentido
  - Por ejemplo: una agregación de las piezas de un coche
- En una asociación que un miembro se borre no hace que deje de tener sentido (se usa en UML)



# Unión

- Si deseamos que una empresa o persona puedan poseer un vehículo:



# Problemas modelo E-E/R

- El modelo EE/R no está normalizado
  - La herencia es la extensión más comúnmente aceptada
- Diferentes metodologías usan diferentes constructores
  - O los mismos con distinto significado :-P
- Algunos tienen limitaciones
  - Si no permite herencia múltiple puedo crear a mano las combinaciones de entidades

# Diseño conceptual

- Las etapas del diseño conceptual son:
  - Análisis de requisitos
  - Diseño del esquema conceptual (EC)
- *Análisis de requisitos*: se analizan las especificaciones de requisitos para identificar aquellos relacionados con la información manipulada por el sistema
  - Necesitamos eliminar:
    - Ambigüedades de los requisitos recabados → de vuelta al cliente
    - Completar los requisitos
    - Dotarlos de una estructura: párrafo → viñetas
    - Entender realmente el significado de todos los términos → puede requerir formación

# Diseño conceptual

- Diseño del esquema conceptual: realiza un refinamiento y estructuración sucesivos del esquema percibido para obtener el EC
- Utilización en un MD de alto nivel independiente de la implementación
  - Entendimiento completo de la estructura, semántica (significado), interrelaciones y restricciones de la BD
  - Descripción del contenido (estructural) de la BD, invariante en el tiempo: se puede cambiar el SGBD pero permanece el EC
  - Es general y expresivo, y facilita el entendimiento entre usuarios, diseñadores y analistas. Conceptos más fáciles de entender que los MD de bajo nivel

# Diseño conceptual

- El paso de esquema descriptivo a un primer Esquema Conceptual (EC) se puede realizar mediante dos enfoques complementarios:
  - Lingüístico: atendiendo al texto que describe los requisitos
  - Categorización de los objetos: atendiendo a los objetos que se describen en los requisitos

# Diseño conceptual

- Enfoque lingüístico
  - Por lo general (no siempre):
    - Sustantivo que actúa como sujeto o complemento directo en una frase → tipo de entidad
    - Nombre propio → ocurrencia de un tipo de entidad
    - Verbo transitivo o una frase verbal → tipo de relación
    - Preposición o frase preposicional entre dos nombres → tipo de relación o asociación entre una entidad y alguno de sus atributos

# Diseño conceptual

- Enfoque de categorización de los objetos:
  - Si un concepto tiene una estructura simple sin propiedades y/o describe a un objeto al que sólo se le asigna un valor
    - Suele ser atributo del concepto al cual se refiere
  - Si un concepto tiene más propiedades que su nombre, y/o describe un tipo de datos con existencia autónoma
    - Suele ser un tipo de entidad
  - Si un concepto relaciona (con lógica) varias entidades, y/o hace posible la selección de una entidad a través de una referencia a un atributo de otra entidad
    - Suele ser una relación

# Diseño conceptual

- Ejemplo sobre una editorial
  - Libro es una entidad pues posee propiedades y su existencia es independiente de otros conceptos
  - Si un libro puede encontrarse en varios idiomas nos indica que idioma es un atributo multivaluado. Sería conveniente considerar el tipo de entidad Idioma interrelacionado con Libro
  - Una editorial puede tener propiedades por lo que es mejor considerar que los libros son editados por las editoriales.
  - Si queremos conocer los libros que ha escrito un autor es mejor representar Autor como un tipo de entidad y una relación entre los libros y los autores que los escriben en vez de que sean una propiedad de libro

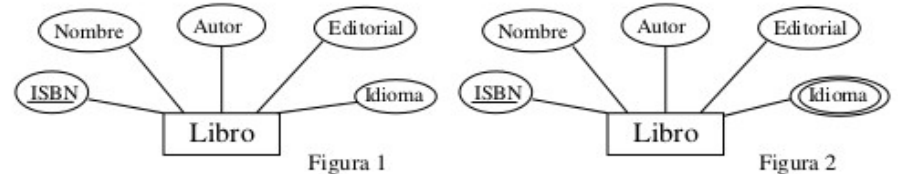


# Diseño conceptual

- Ejemplo sobre una editorial
  - Libro posee propiedades y su existencia es independiente de otros conceptos
  - Un libro puede encontrarse en varios idiomas
  - Una editorial puede tener propiedades
  - Si queremos conocer los libros que ha escrito un autor se puede replantear el rol del autor

# Diseño conceptual

- Libro es una entidad pues posee propiedades y su existencia es independiente de otros conceptos
- Si un libro puede encontrarse en varios idiomas nos indica que idioma es un atributo multivaluado. Sería conveniente considerar el tipo de entidad Idioma interrelacionado con Libro
- Una editorial puede tener propiedades por lo que es mejor considerar que los libros son editados por las editoriales
- Si queremos conocer los libros que ha escrito un autor es mejor representar Autor como un tipo de entidad y una relación entre los libros y los autores que los escriben en vez de que sean una propiedad de libro



# Documentación

- Es ABSOLUTAMENTE necesario adjuntar documentación para la interpretación del EC
  - A veces estamos tentados a decir “esto es de perogrullo” y no documentar
    - A lo mejor es de perogrullo, pero si después de X días lo retomas ¿te acordarás?
    - ¿Todas las personas que lean el documento lo entenderán igual?
    - A veces recibirá procesamiento automático
    - En muchas ocasiones lo que para un cliente es de una forma, para otro será de otra

# Documentación

- La documentación se puede organizar como un DD. Se requieren dos tablas:
  - Entidades:
    - Nombres y descripciones
    - Atributos
    - Identificadores (CP)
  - Relaciones:
    - Nombres y descripciones
    - Entidades involucradas y sus cardinalidades
    - Atributos
- Las restricciones del Universo del Discurso también se agrupan en una tabla

# Problemas en el diseño conceptual

- Un esquema es correcto cuando se usan adecuadamente los elementos del modelo E-R
  - *Sintácticamente correcto*: los elementos se usan de acuerdo a sus definiciones
    - Usar un atributo en vez de una entidad
    - Usar una relación con un nº erróneo de entidades participantes
    - Usar una entidad en lugar de una relación
    - Omitir alguna especificación de cardinalidad
  - *Semánticamente correcto*: los conceptos se expresan correctamente en el esquema
    - Ej: “El programa suma dos números a veces”

# Problemas en el diseño conceptual

- Redundancia: un esquema es redundante si representa el mismo concepto con dos elementos. Hay que evitarla:
  - *Elementos redundantes*
  - *Atributos derivados*
  - *Ciclos redundantes*
- Vemos cada caso en detalle

# Problemas en el diseño conceptual

- *Elementos redundantes*: son elementos (entidades, atributos o relaciones) que representen lo mismo
  - Se pueden dar porque se usen sinónimos
  - Pueden resolverse con generalización / unión: dos entidades que tengan una misma relación con otra tercera (relación que evolucionaría en el tiempo igual para ambas)

# Problemas en el diseño conceptual

- *Atributos derivados*: la redundancia puede deberse a la existencia de un atributo cuyo valor puede calcularse a partir de los valores de otros atributos
  - El cliente puede estar muy preocupado por saber el total de cada factura
    - Si el total de cada factura se puede calcular a partir de las cantidades y los precios de artículos no lo guardo en el E/R
    - Si el total es algo introducido por el usuario (porque puede hacer descuentos o poner el precio que le venga en gana) sí que iría

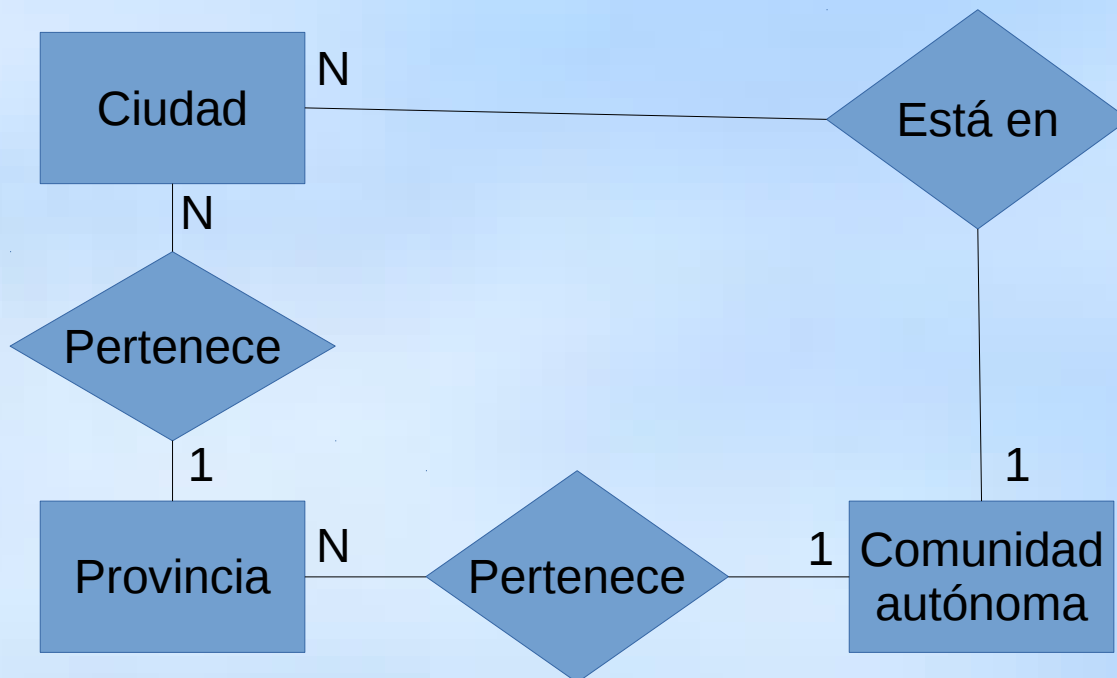


# Problemas en el diseño conceptual

- *Ciclos redundantes:*
  - Un ciclo se da cuando una relación  $R_1$  entre dos entidades posee el mismo contenido de información que una ruta de relaciones ( $R_2, R_3, \dots, R_n$ ) que conecta a los mismos pares de ocurrencias de entidades que  $R_1$
  - Hay ciclos redundantes y ciclos no redundantes

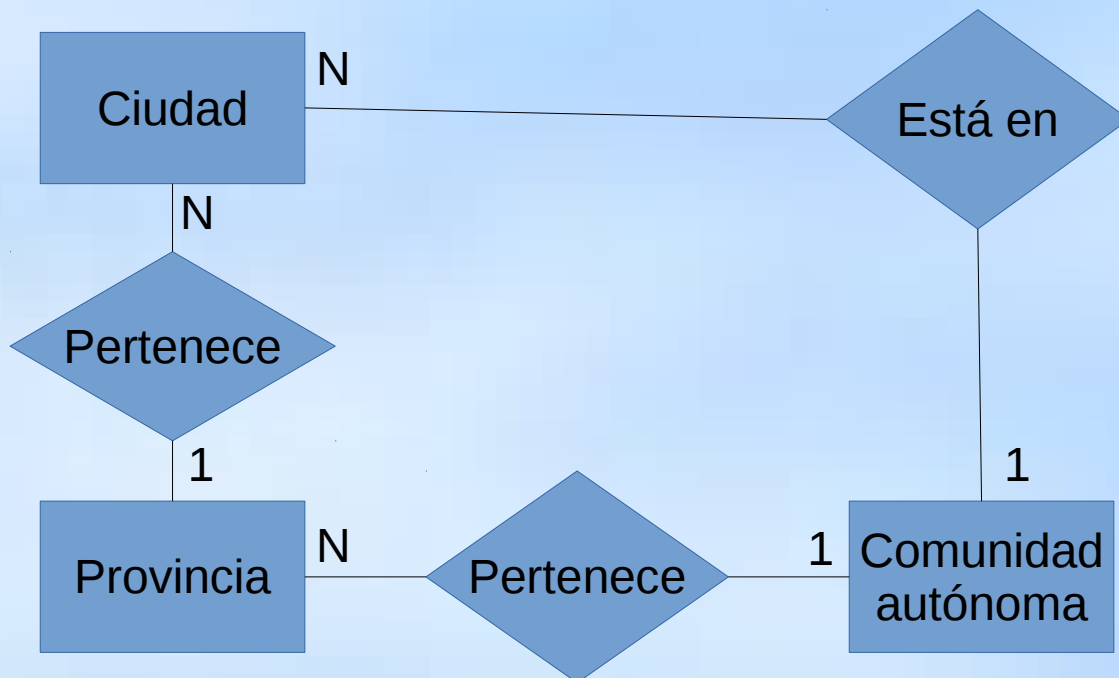
# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Está correcto?



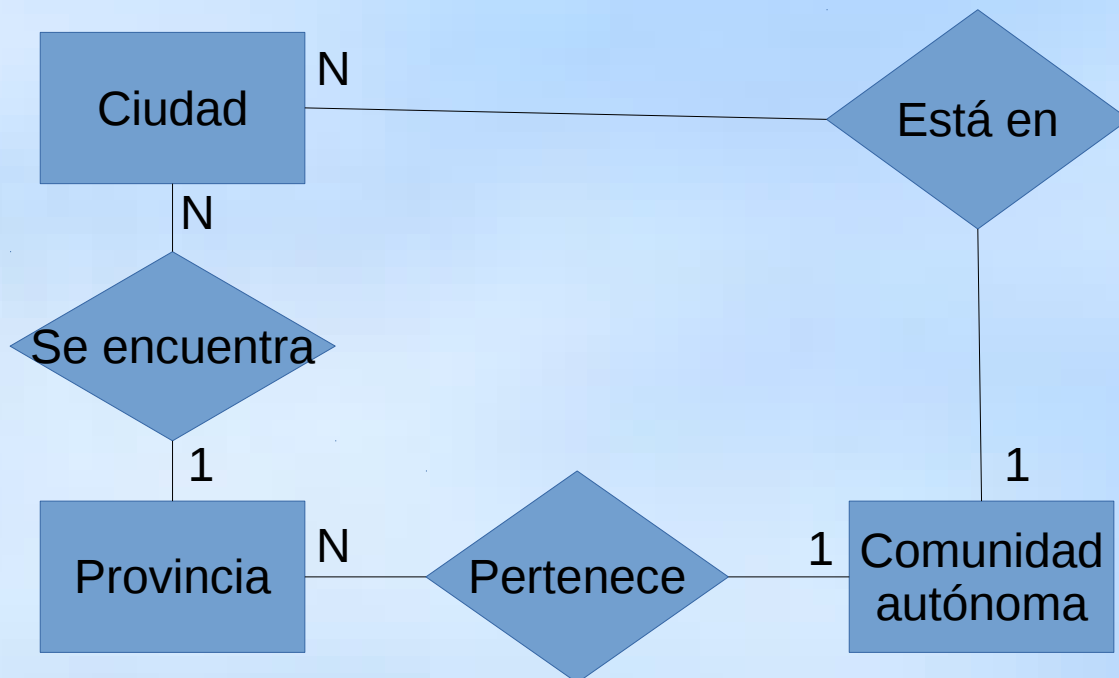
# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Está correcto? No, error sintáctico de *Pertenece*



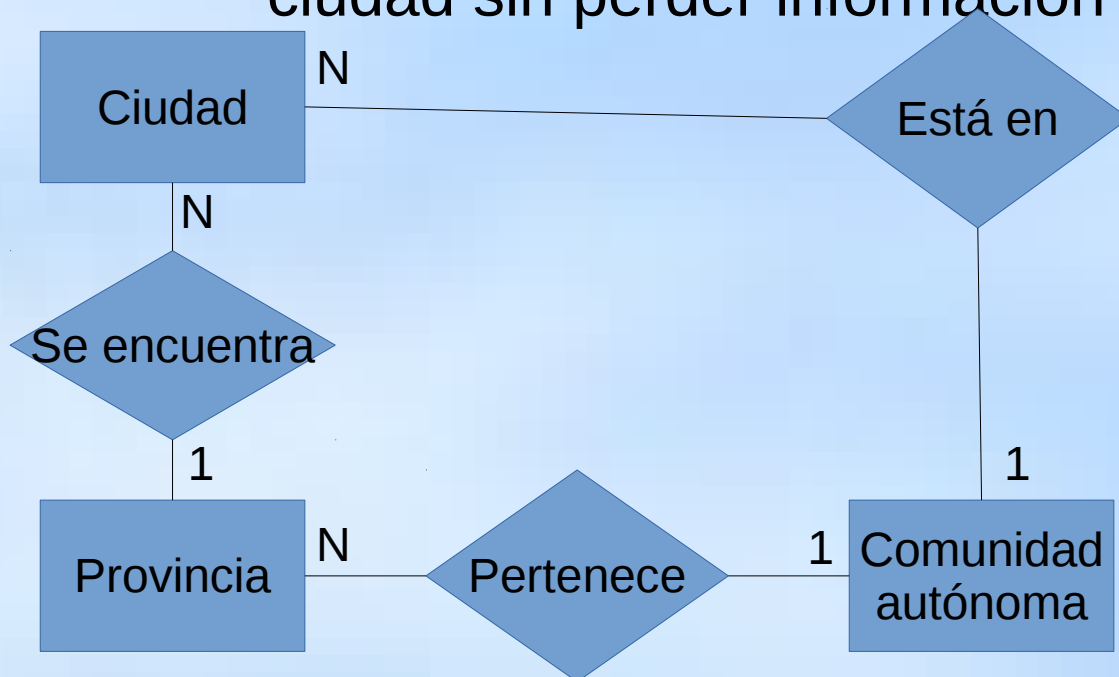
# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Está correcto?



# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Está correcto? No porque CA es navegable desde ciudad sin perder información



# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Y si en un caso en que haya que navegar por 8 relaciones para sacar un dato el cliente me dice que es un dato crítico y necesita accederse rápidamente?

# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Y si en un caso en que haya que navegar por 8 relaciones para sacar un dato el cliente me dice que es un dato crítico y necesita accederse rápidamente?
      - Eso no es problema del E/R, es un *requisito de rendimiento*

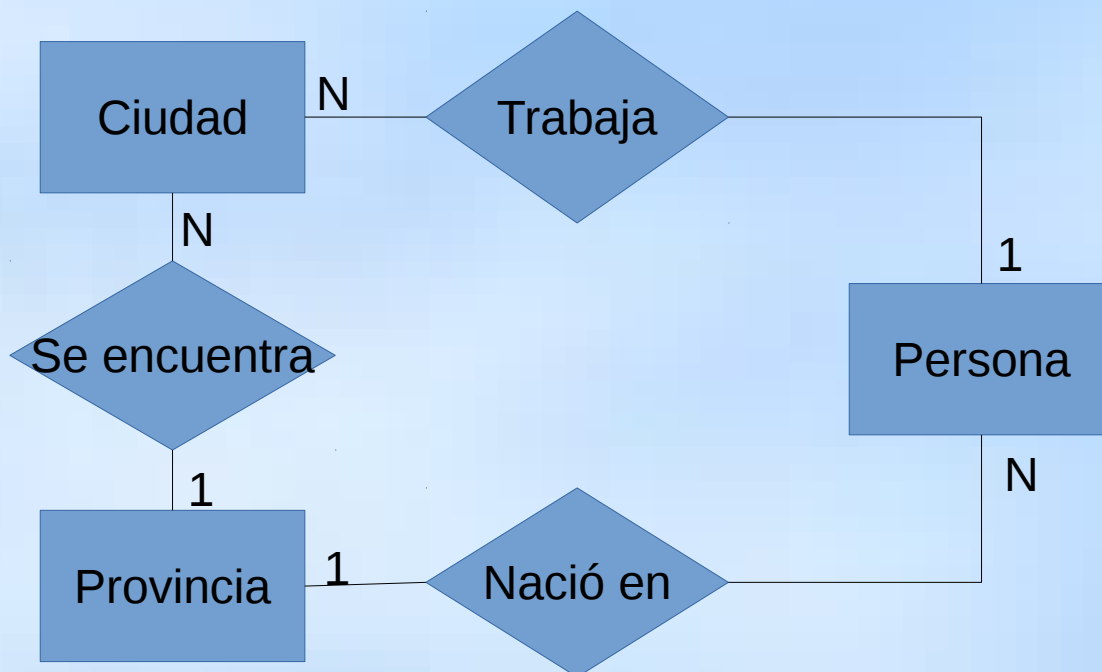
# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 1*: información sobre ciudades, provincias y comunidades autónomas (CA). Y necesitamos conocer en qué CA está cada ciudad
    - ¿Y si en un caso en que haya que navegar por 8 relaciones para sacar un dato el cliente me dice que es un dato crítico y necesita accederse rápidamente?
      - Eso no es problema del E/R, es un *requisito de rendimiento*
      - Es posible que una solución sea pase por crear una tabla que relacione dichos elementos (o no)
        - Pero habría que tener cuidado con las actualizaciones
      - Y puede ser que no sea suficiente (¿cuánto de rápido necesito?)
        - Organización interna de ficheros, índices, caché, ...



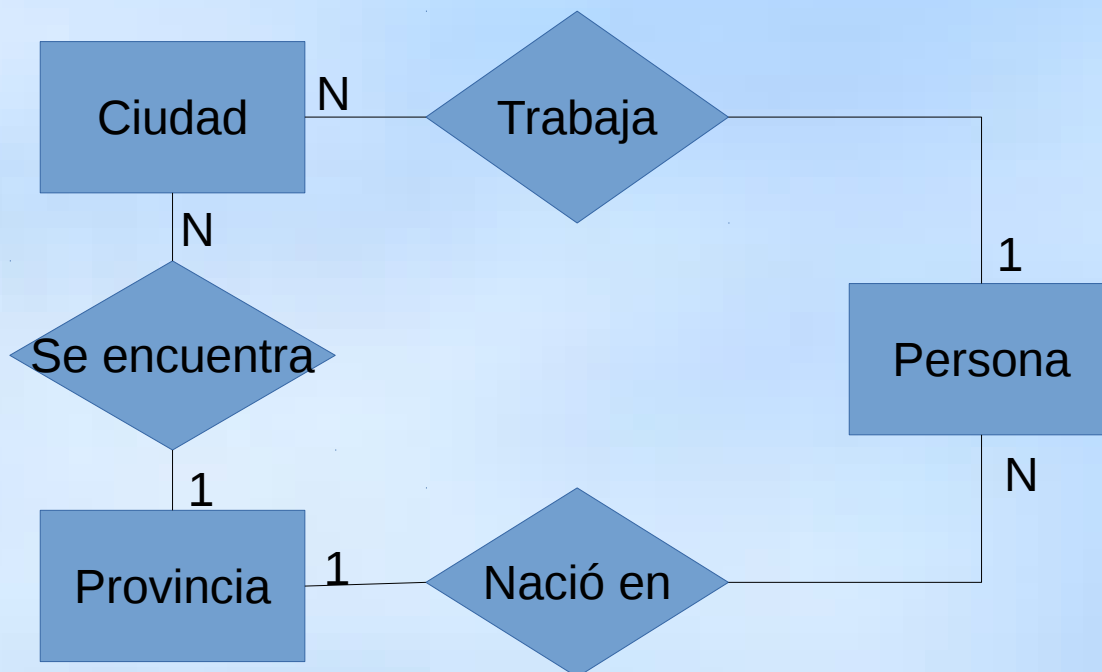
# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 2*: información sobre ciudades y provincias. Y necesitamos conocer en qué provincia nació y en qué ciudad trabaja una persona
    - ¿Es redundante trabaja?



# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 2*: información sobre ciudades y provincias. Y necesitamos conocer en qué provincia nació y en qué ciudad trabaja una persona
    - ¿Es redundante trabaja? No



# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 2*: información sobre ciudades y provincias. Y necesitamos conocer en qué provincia nació y en qué ciudad trabaja una persona
    - ¿Es redundante trabaja? No. Pero ¿por el 1:N que ha cambiado respecto al ejemplo 1?



# Problemas en el diseño conceptual

- Dos ejemplos de ciclos
  - *Ejemplo 2*: información sobre ciudades y provincias. Y necesitamos conocer en qué provincia nació y en qué ciudad trabaja una persona
    - ¿Es redundante trabaja? No. No, por la *semántica de la relación*



# Problemas en el diseño conceptual

- No todos los ciclos de relaciones son fuentes de redundancia: lo serán o no dependiendo de su significado
  - En la figura anterior aunque existe un ciclo no hay redundancia, pues una persona no tiene por qué haber nacido en la misma provincia en la que está la ciudad donde trabaja

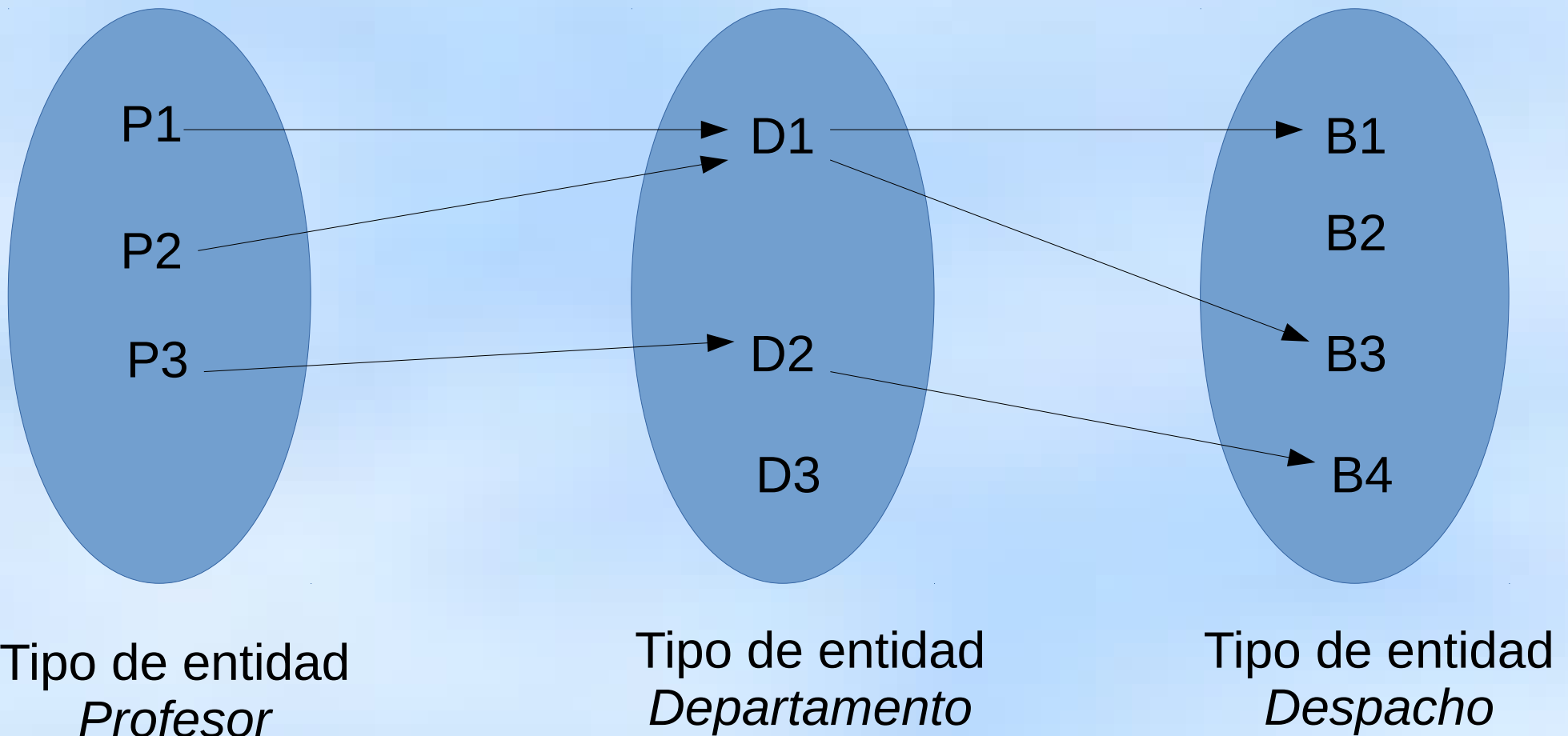
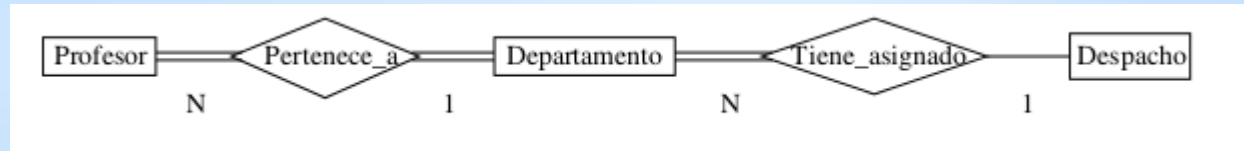
# Problemas en el diseño conceptual

- Existen dos tipos de problemas que pueden presentarse en un esquema E-R:
  - *Trampa del abanico*: ocurre cuando un modelo representa una relación entre tipos de entidades, pero el camino entre ciertas ocurrencia de las entidades es ambiguo.
  - *Trampa de la grieta*: ocurre cuando un modelo sugiere la existencia de una relación entre tipos de entidades, pero no existe camino entre ciertas ocurrencia de entidades.

# Problemas en el diseño conceptual

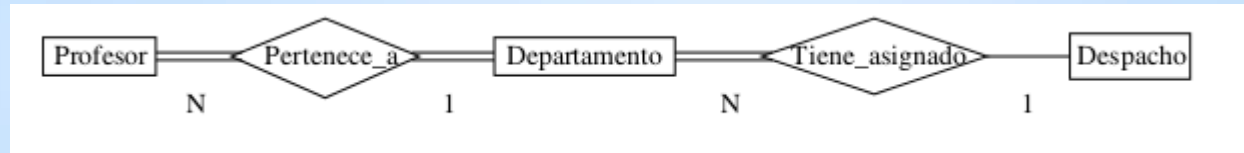
- Ejemplo abanico: deseamos almacenar información sobre los despachos, Departamentos y profesores que los ocupan
  - En la Universidad los despachos se ceden a los Departamentos y estos los asignan a sus profesores

# Problemas en el diseño conceptual

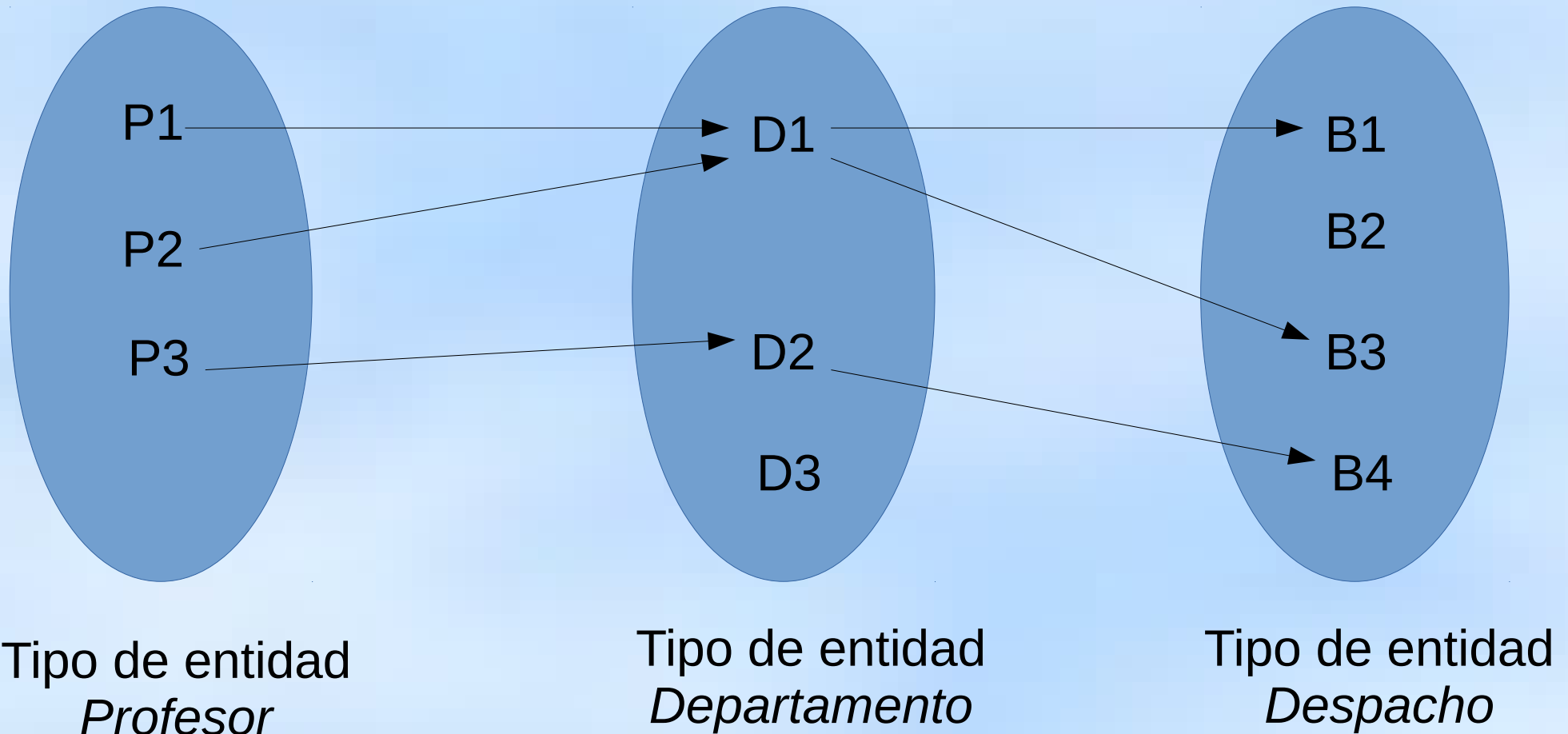




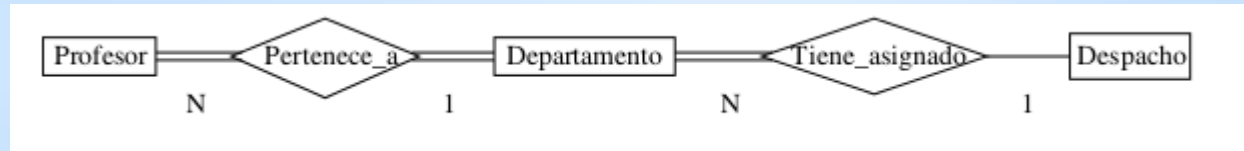
# Problemas en el diseño conceptual



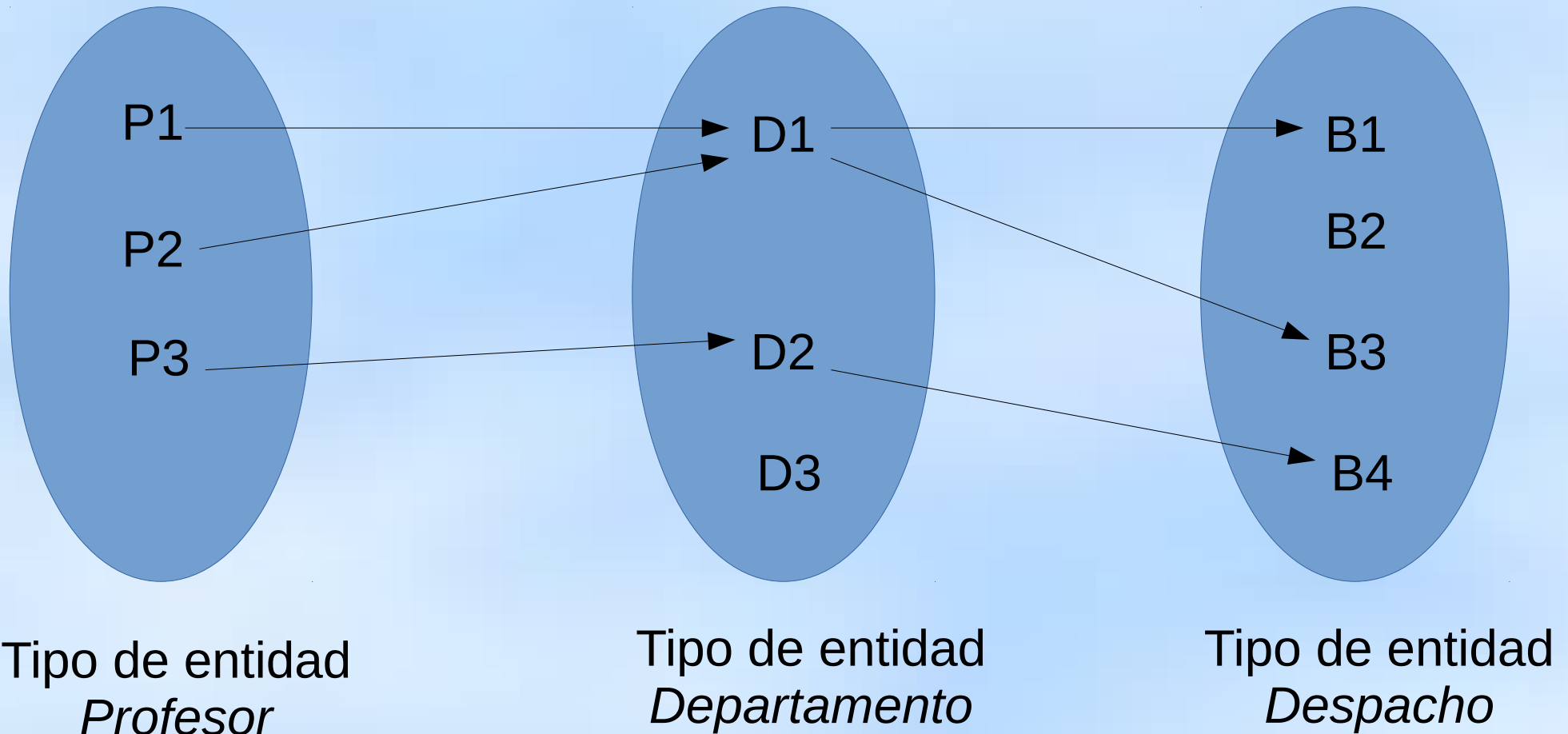
¿Poner una relación entre profesor y Despacho es buena idea?



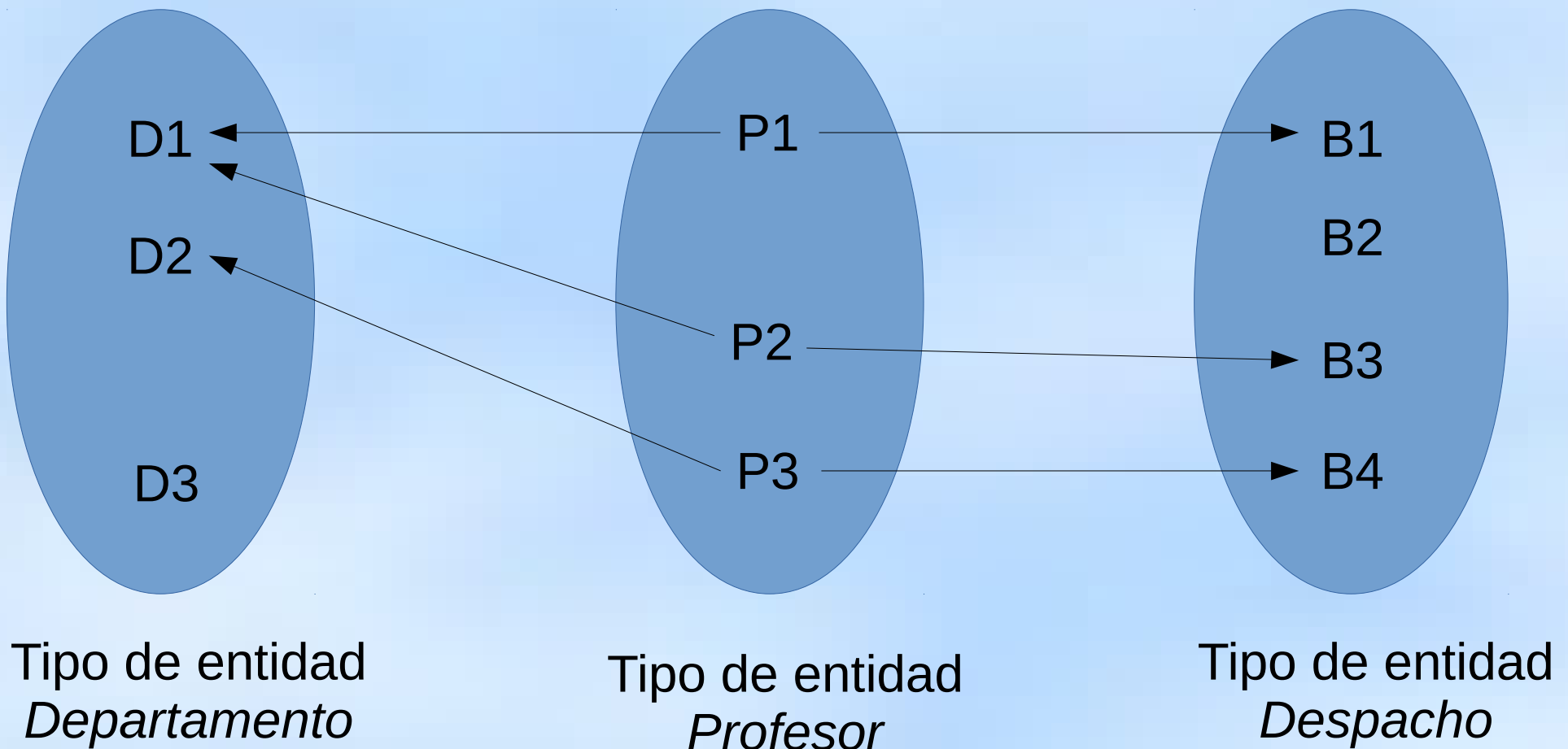
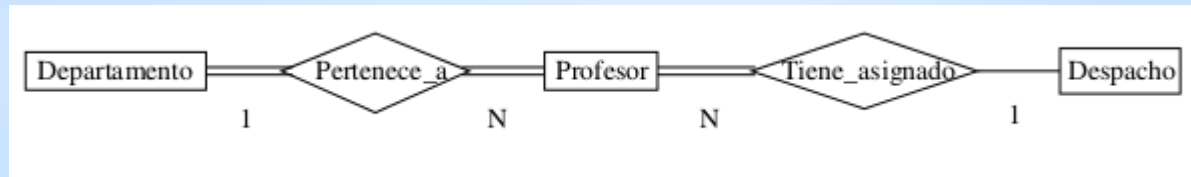
# Problemas en el diseño conceptual



Una relación entre Profesor y Despacho tendría dependencias de las otras (si se desasigna un Despacho al Departamento se le tendría que quitar al profesor)



# Problemas en el diseño conceptual



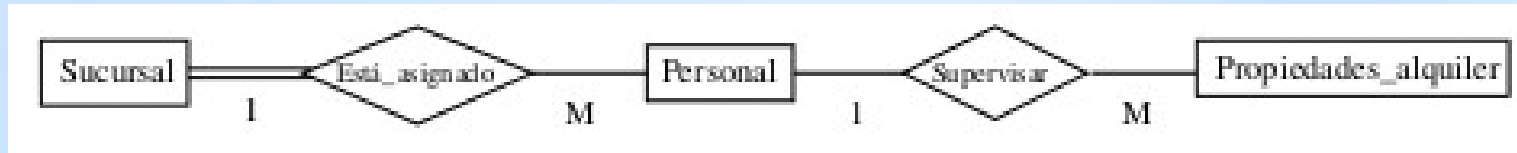
# Problemas en el diseño conceptual

- Abanico: en el primer esquema conceptual no sabemos cuál es el despacho de un profesor en concreto
  - Sólo sabemos los despachos que están asignados a un departamento en concreto
- En cambio, en el segundo esquema sí sabemos el despacho que ocupa un profesor concreto

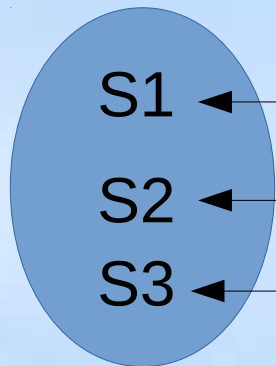
# Problemas en el diseño conceptual

- Ejemplo grieta: deseamos almacenar información sobre el personal de las sucursales y los alquileres que supervisan

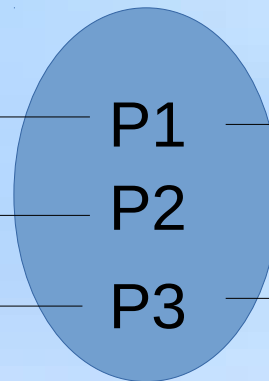
# Problemas en el diseño conceptual



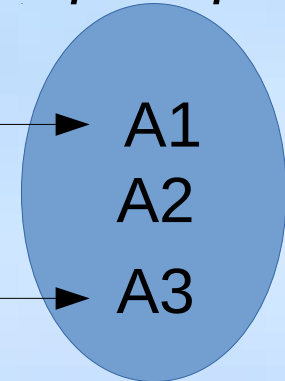
Tipo de entidad  
*Sucursal*



Tipo de entidad  
*Personal*

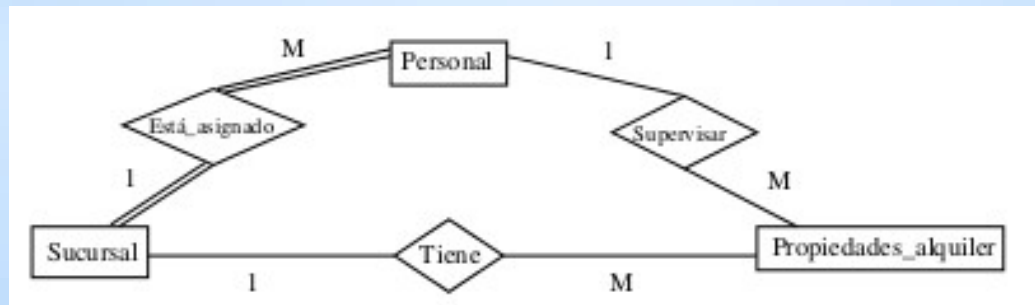


Tipo de entidad  
*Prop\_alquilable*



Con este modelo una propiedad no alquilada no sabríamos a qué sucursal pertenece

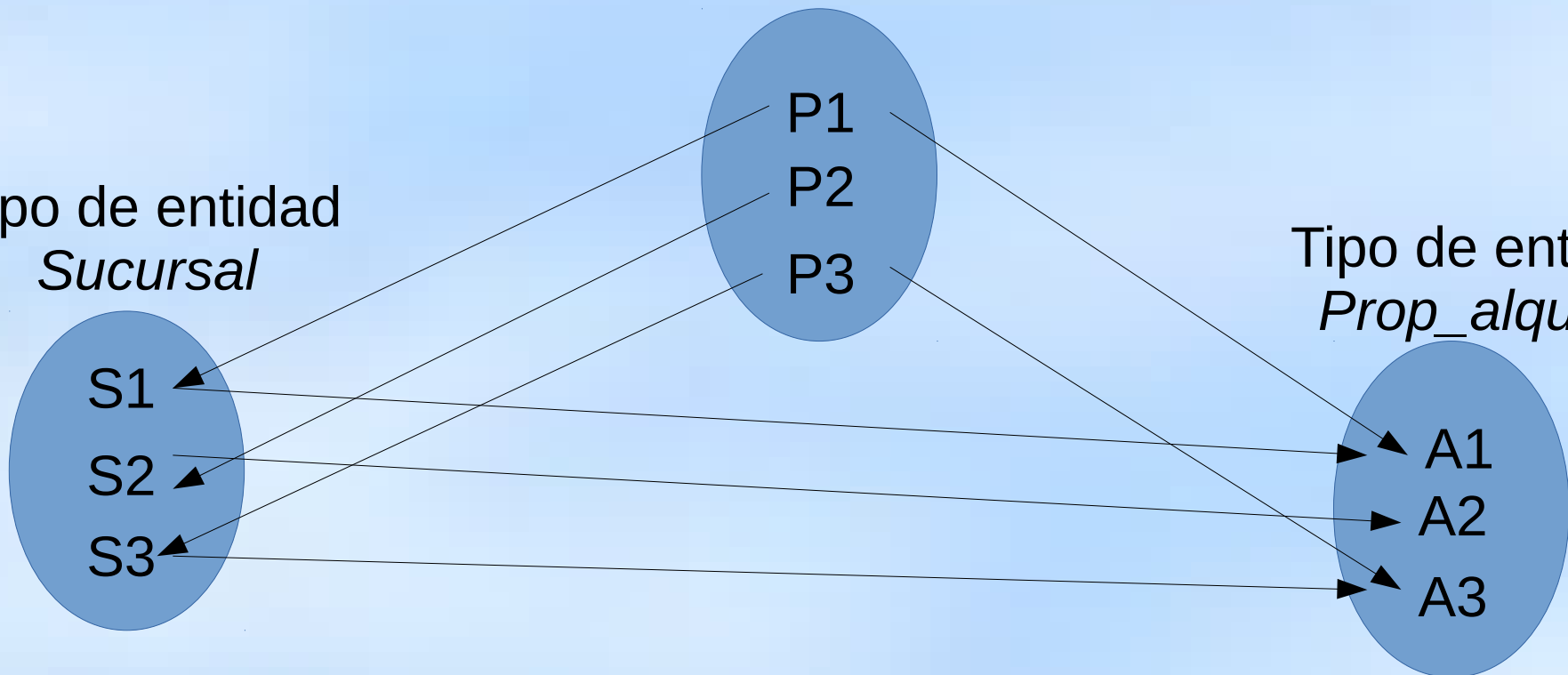
# Problemas en el diseño conceptual



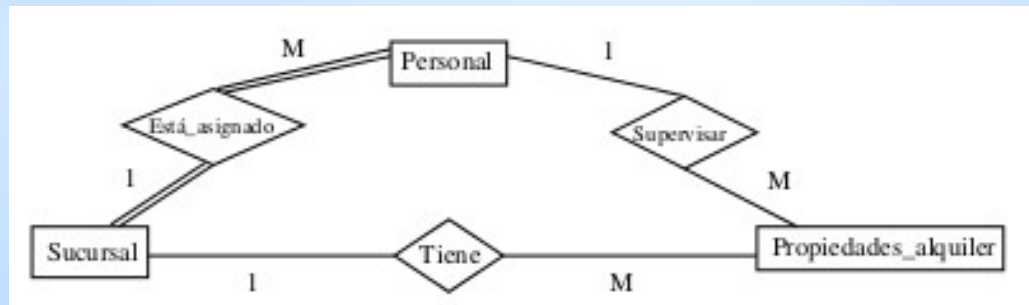
Tipo de entidad  
*Personal*

Tipo de entidad  
*Sucursal*

Tipo de entidad  
*Prop\_alquiler*



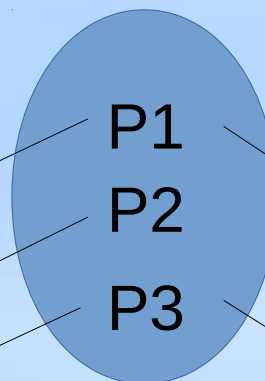
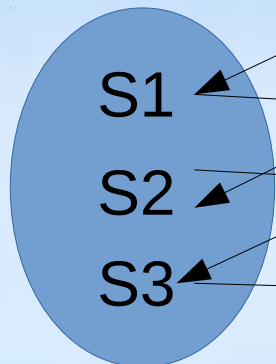
# Problemas en el diseño conceptual



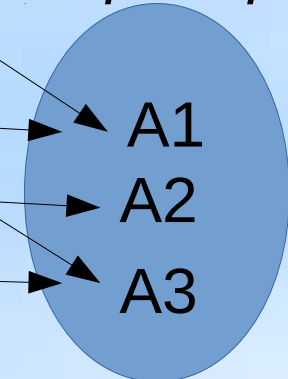
Tipo de entidad  
*Personal*

Esta nueva relación “tiene”  
tendría dependencia con  
“supervisa”, pero no hay  
otra alternativa

Tipo de entidad  
*Sucursal*

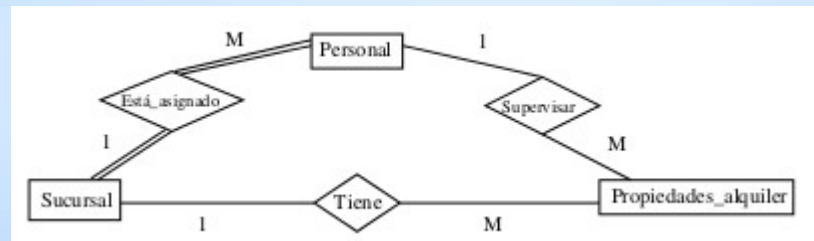


Tipo de entidad  
*Prop\_alquiler*





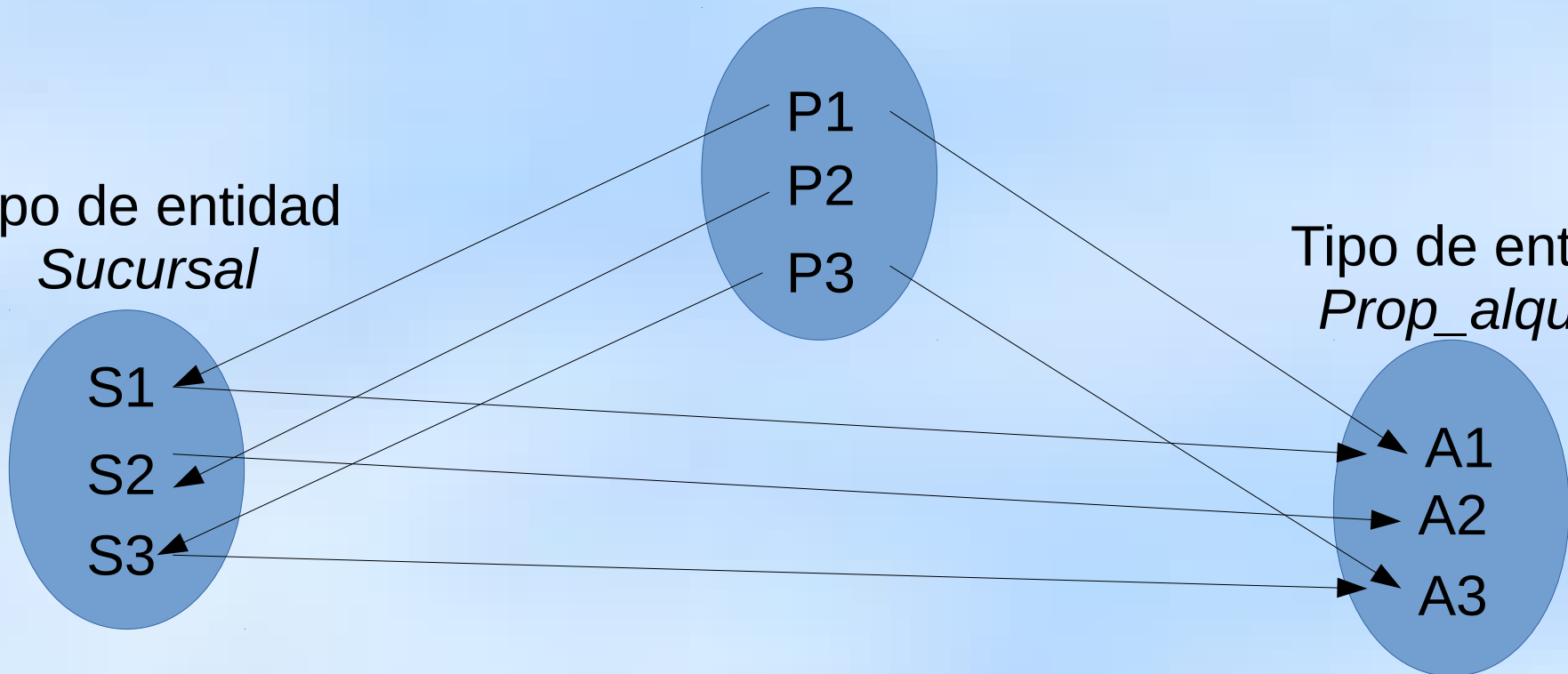
# Problemas en el diseño conceptual



Tipo de entidad  
*Personal*

Tipo de entidad  
*Sucursal*

Tipo de entidad  
*Prop\_alquiler*



Esta relación “tiene” tendría dependencia con “supervisa”, pero no hay otra alternativa

# Problemas en el diseño conceptual

- En el primer esquema conceptual comprobamos que no todo el personal supervisa propiedades ni todas las propiedades están supervisadas por algún personal
- Grieta: el problema se presenta cuando queremos saber qué propiedades están disponible en cada sucursal
- Para ello se crea una nueva relación *Tiene*

# Problemas en el diseño conceptual

- Existen otras situaciones que se dan en modelados similares y que se pueden refinar
  - No son recetas mágicas ni para todo caso
- Por ejemplo, si varias entidades tienen un mismo atributo (que representa lo mismo), es probable que sea mejor pasarlo a entidad
  - Ej: los profesores, los alumnos y el PAS tienen como atributo un centro universitario

# Problemas en el diseño conceptual

- Otro problema: *Entidades “vacías” innecesarias*
  - Las observo muy a menudo en exámenes
  - Ejemplo:
    - Los clientes llevan sus coches a revisiones a un taller concreto, donde varios mecánicos trabajan con ellos



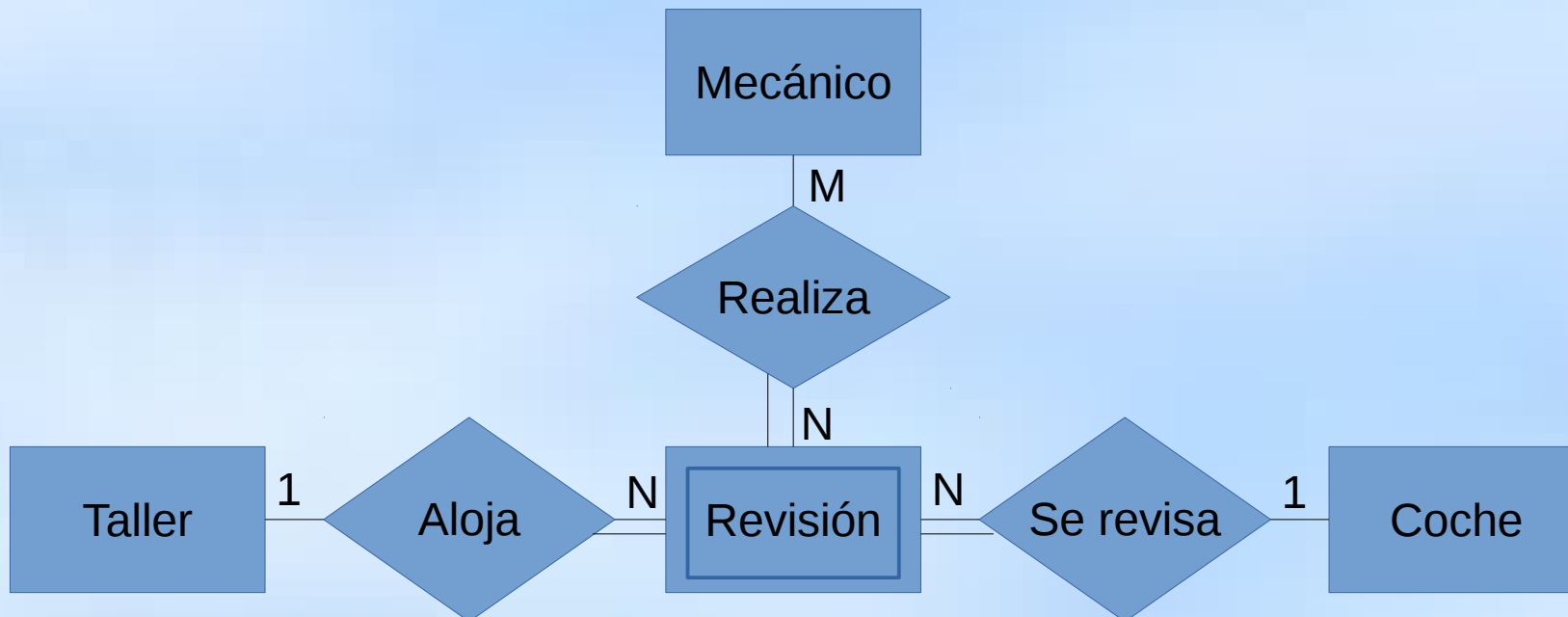
# Problemas en el diseño conceptual

- Otro problema: *Entidades “vacías” innecesarias*
  - Ejemplo:
    - Los clientes llevan sus coches a revisiones a un taller concreto, donde varios mecánicos trabajan con ellos
    - Pero ¿qué clave tiene la Revisión? Me puedo inventar un código, pero realmente no tiene ...



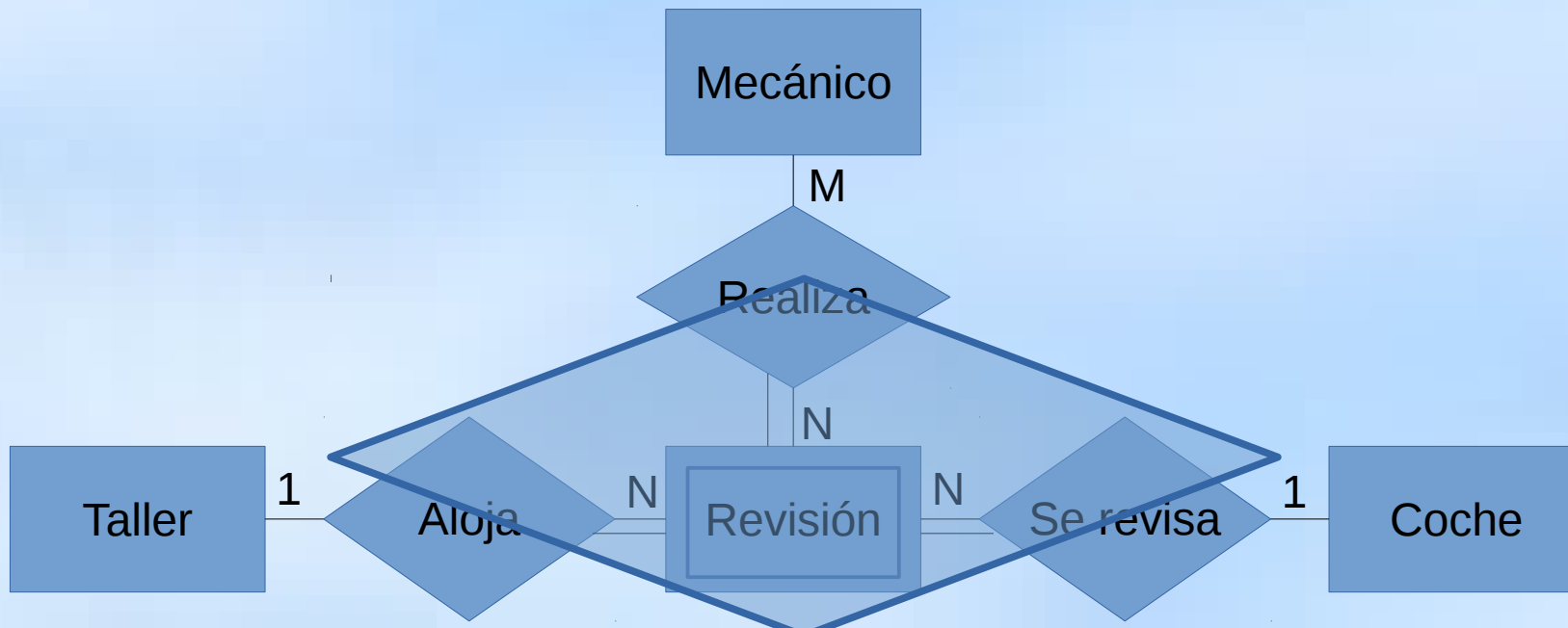
# Problemas en el diseño conceptual

- Otro problema: *Entidades “vacías” innecesarias*
  - Ejemplo:
    - Los clientes llevan sus coches a revisiones a un taller concreto, donde varios mecánicos trabajan con ellos
    - Es más, ¿puede hacerse una Revisión sin coche, taller o mecánicos?



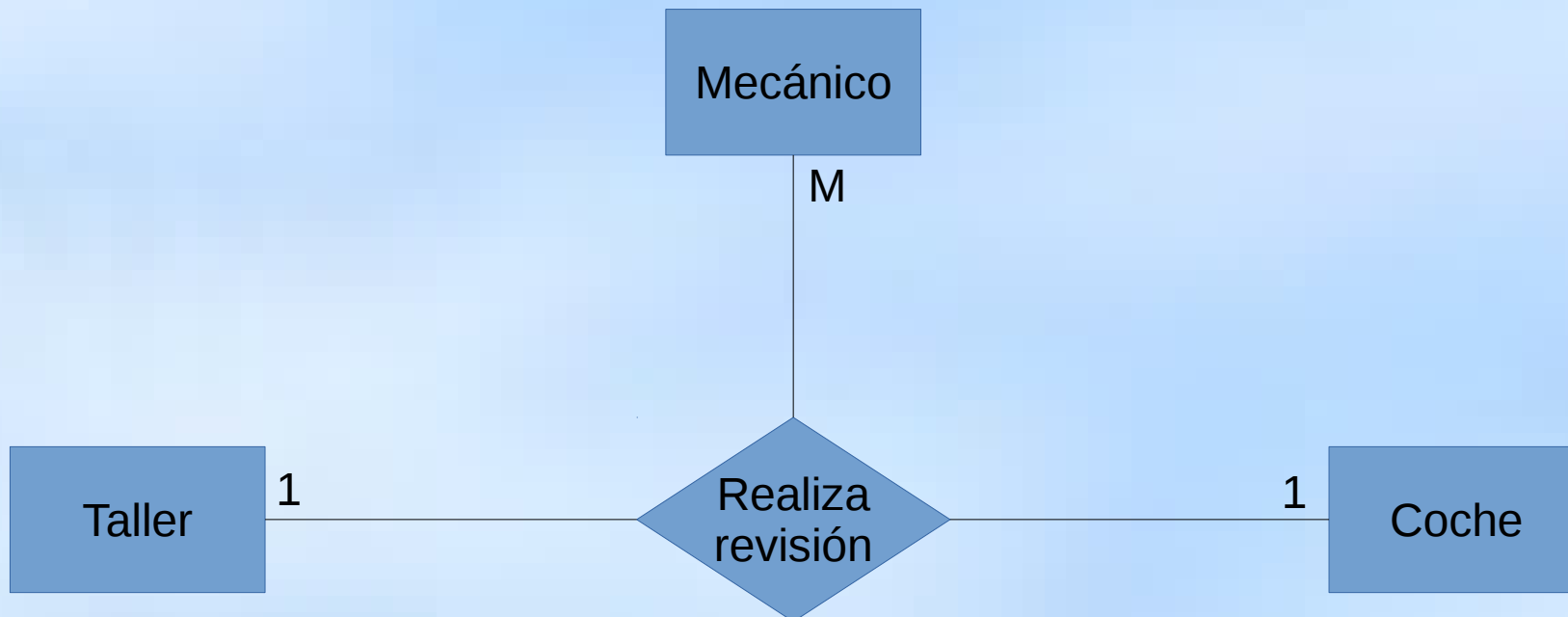
# Problemas en el diseño conceptual

- Otro problema: *Entidades “vacías” innecesarias*
  - Ejemplo:
    - Los clientes llevan sus coches a revisiones a un taller concreto, donde varios mecánicos trabajan con ellos
    - Además esos nombres de relaciones tan genéricos suelen dar que pensar ...



# Problemas en el diseño conceptual

- Otro problema: *Entidades “vacías” innecesarias*
  - Ejemplo:
    - Los clientes llevan sus coches a revisiones a un taller concreto, donde varios mecánicos trabajan con ellos
    - Normalmente suele ser mejor dejarla como una relación





# Problemas en el diseño conceptual

- Existen autores (y herramientas) que no consideran relaciones entre más de dos entidades
  - Y podemos encontrar otras extensiones. Que a veces pueden ayudarnos mucho en el trabajo con determinadas herramientas
  - No obstante, en clase usaremos relaciones entre todas las entidades que hagan falta
    - Ejemplo: guardo datos sobre gimnastas, jueces y polideportivos. Si creo una relación “Competir” puede ser que haya que incluir en ella un polideportivo, una serie de jueces y varios gimnastas ... incluso patrocinadores ;)

# Referencias

- Apuntes de bases de datos de la profesora Esther Gadeschi 2003/4
- Libro “Fundamentals of Database Systems” Elmasri y Navathe (3a Ed.)

Gracias por la atención  
*¿Preguntas?*