

# Sistemas de Información II

## Tema 4. El modelo entidad-relación (continuación)

Bibliografía:

Elmasri y Navathe: “Fundamentos de Sistemas de Bases de Datos”

3ª edición, 2002 (Capítulo 4).

Garcia-Molina, Ullman y Widom: “Database systems: the complete book”. Prentice-Hall (Capítulo 2).

**Carlos Castillo**

UPF – 2007

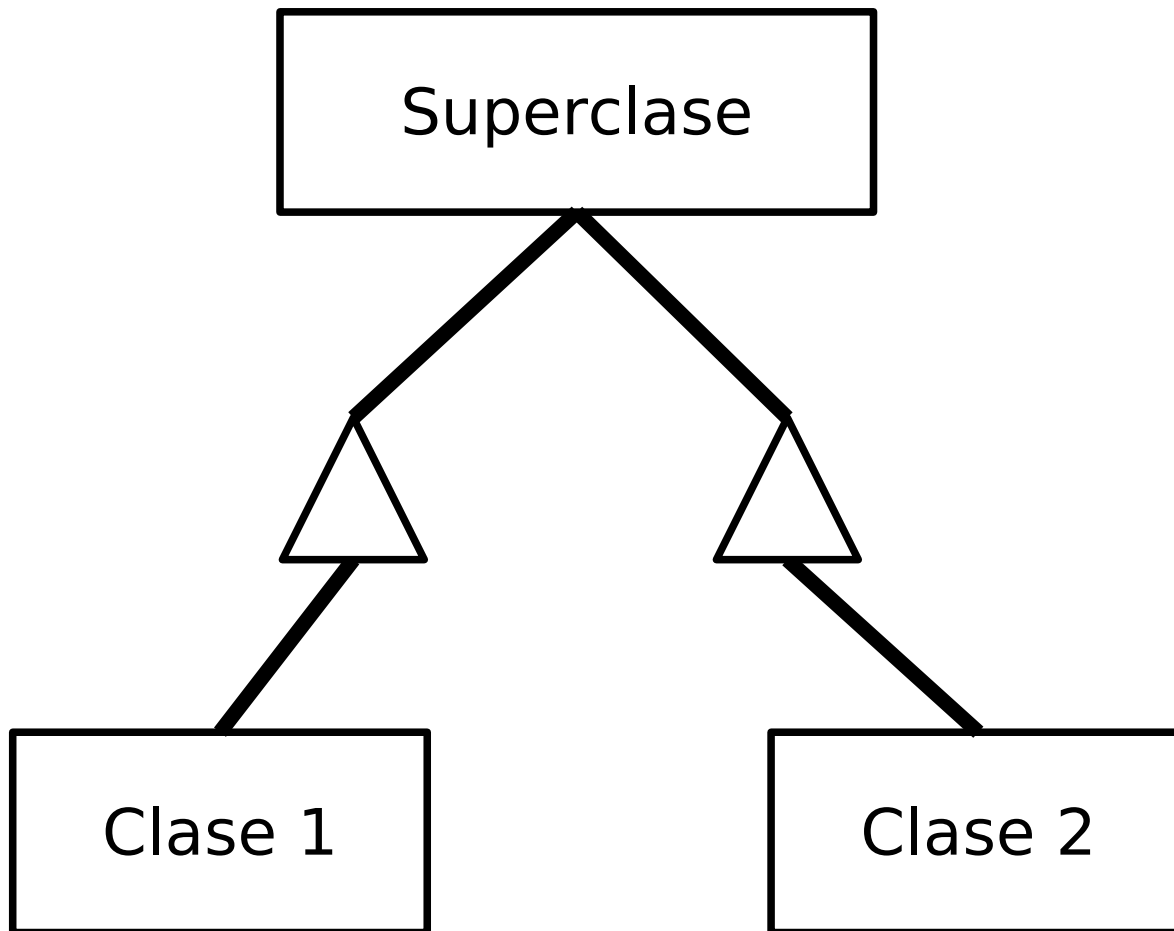
# Subclasses

# ¿Cómo surgen?

- Ejemplo 1:
  - Avión (superficie alar, peso, envergadura)
  - Avión\_de\_carga ( ... + carga máxima)
  - Avión\_de\_pasajeros ( ... + número pasajeros)
- Ejemplo 2:
  - Persona (dni, nombre, apellido)
  - Propietario ( ... + relación con sus propiedades)
  - Arrendatario ( ... + relación con lo que arrienda)

Se usan subclases cuando detectamos una entidad que tiene un grupo de instancias con ciertas particularidades

# Notación



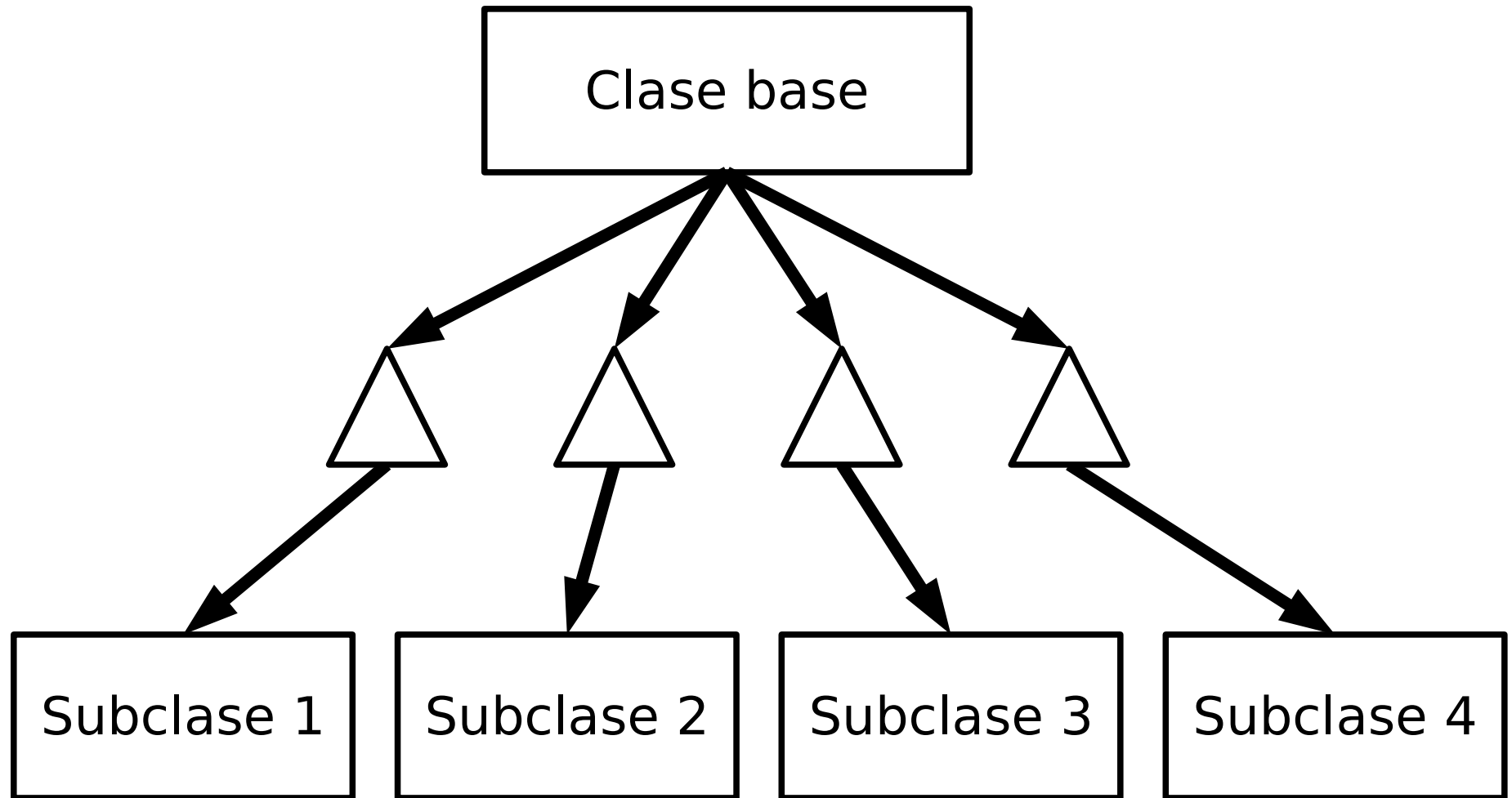
Usaremos esta notación simple en esta asignatura, más una letra para el tipo de sub-clase.

Esta notación simplificada no es estándar, es para fines pedagógicos.

En la siguiente asignatura aprenderán UML.

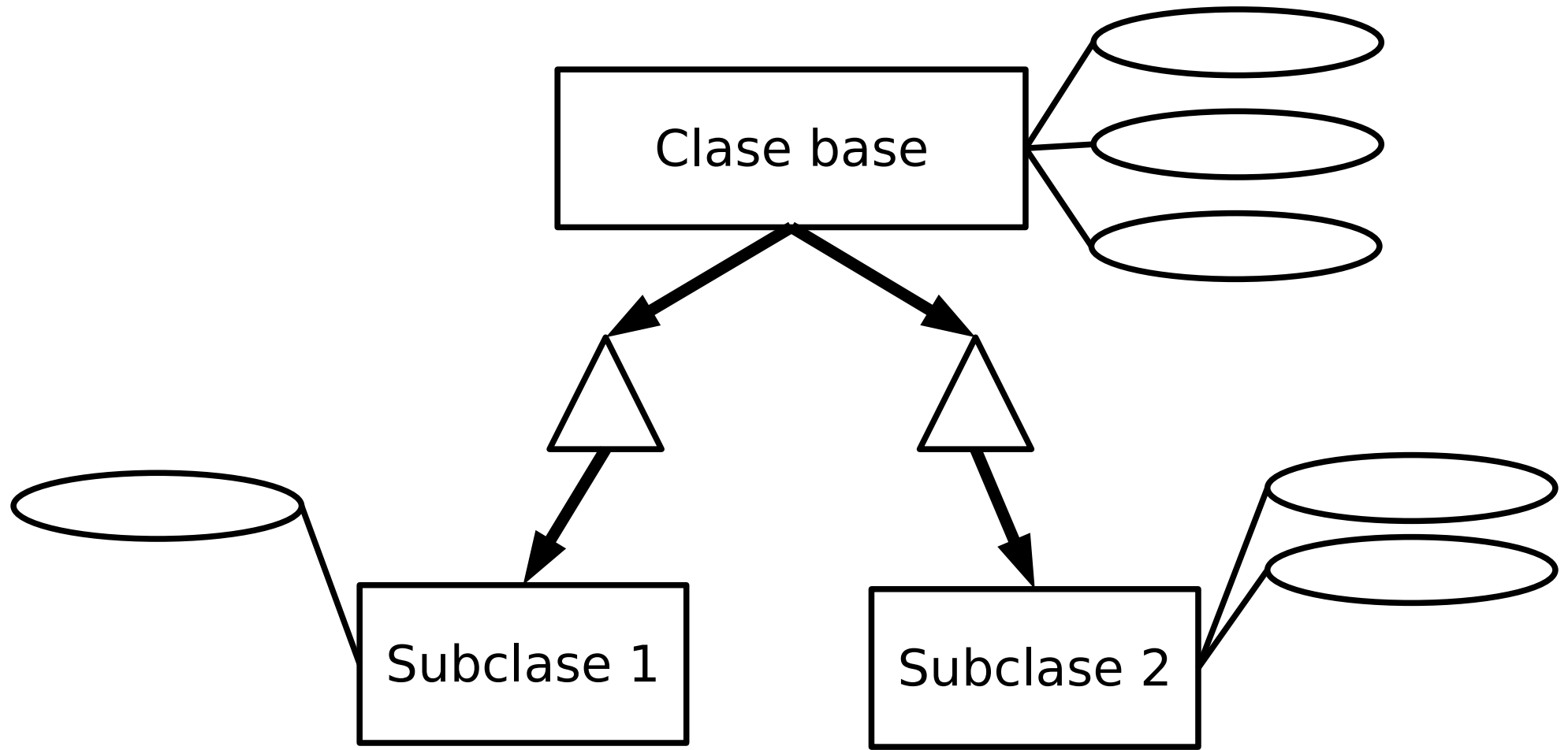
# Especialización

= proceso para definir subclases



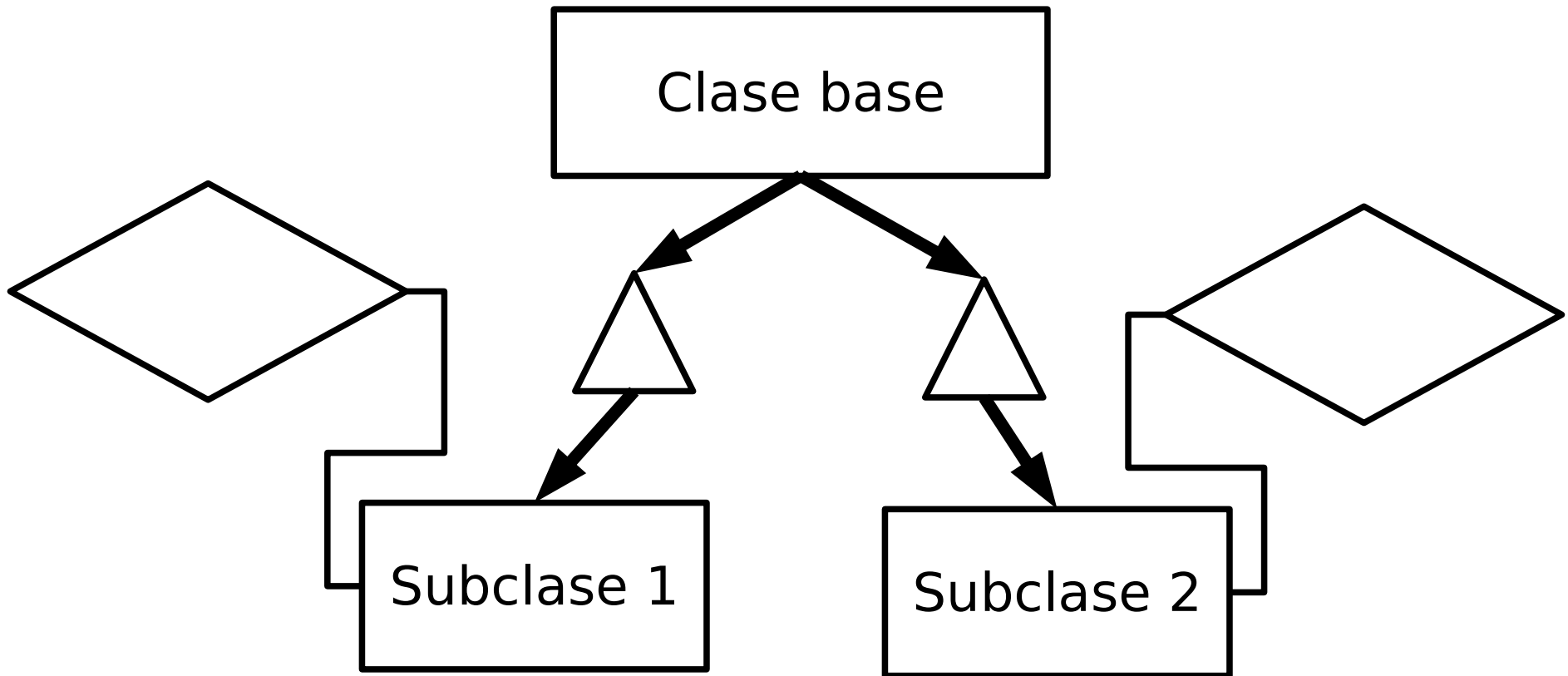
# Especialización:

## 1. atributos específicos



# Especialización:

## 2. relaciones específicas

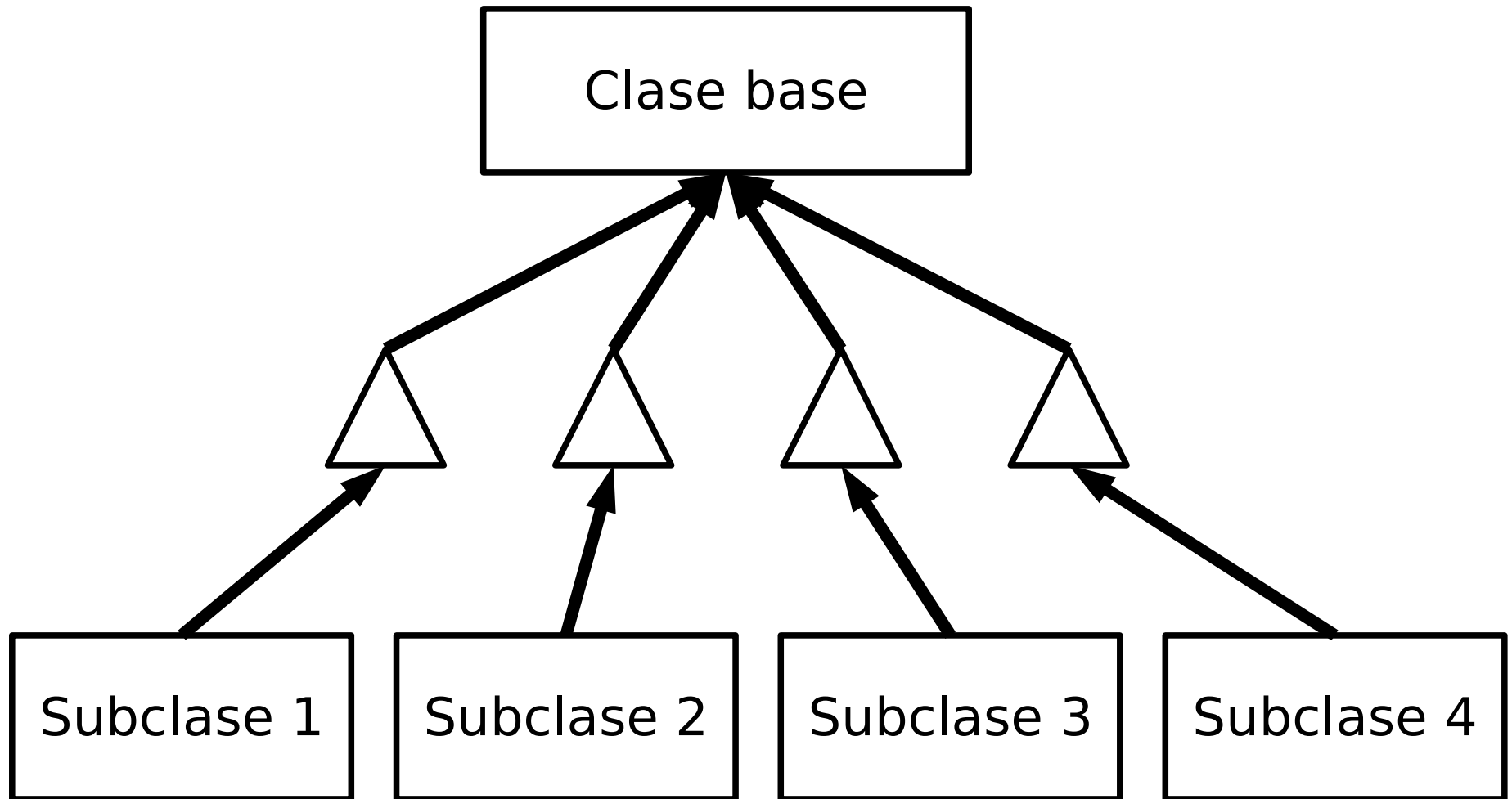


# No especializar demasiado

- Está bien que ciertos atributos sean nulos
  - No toda ausencia/presencia de atributo es relevante
- Mantener la complejidad del problema baja
- Menos entidades es mejor

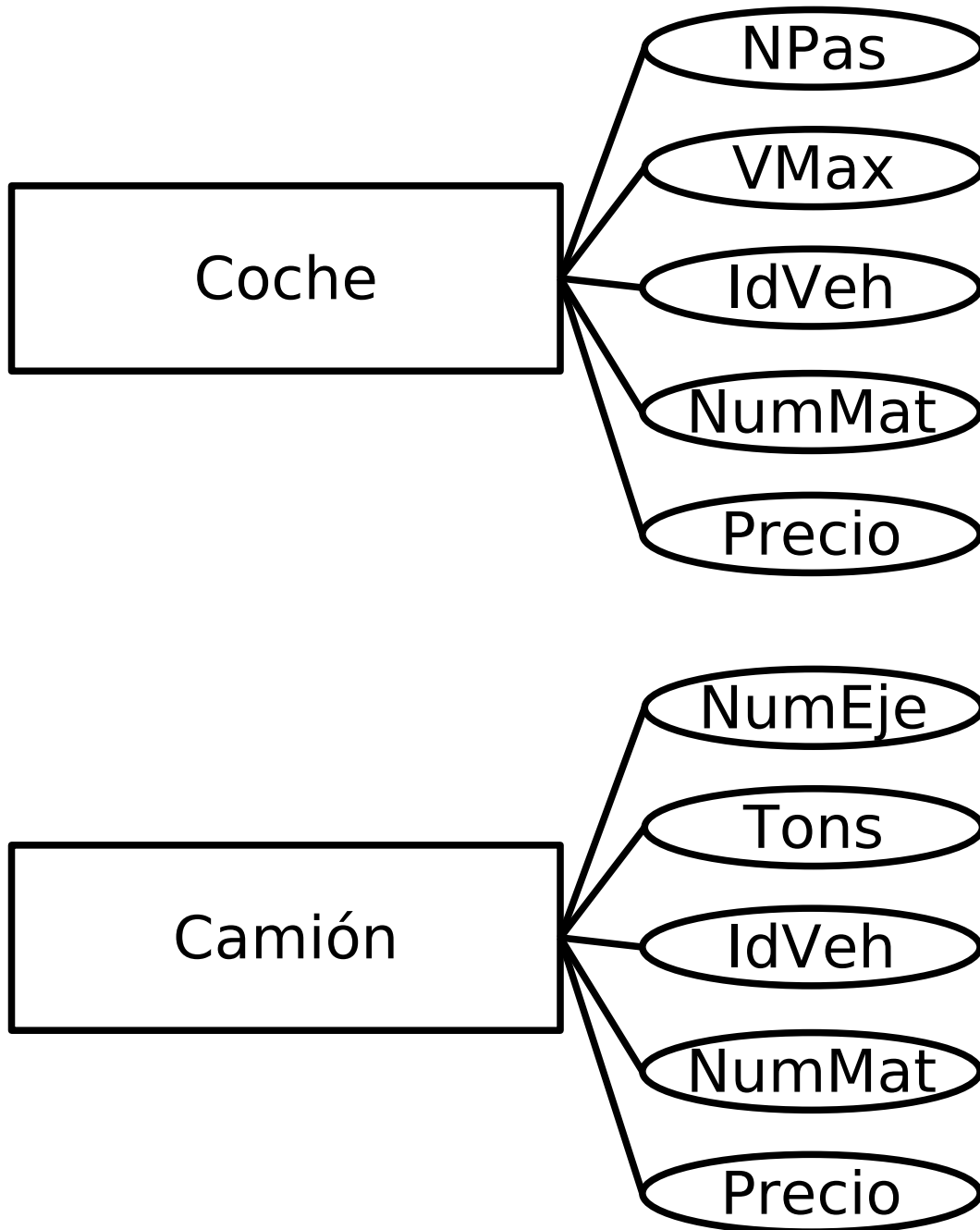
# Generalización

= proceso encontrar superclase

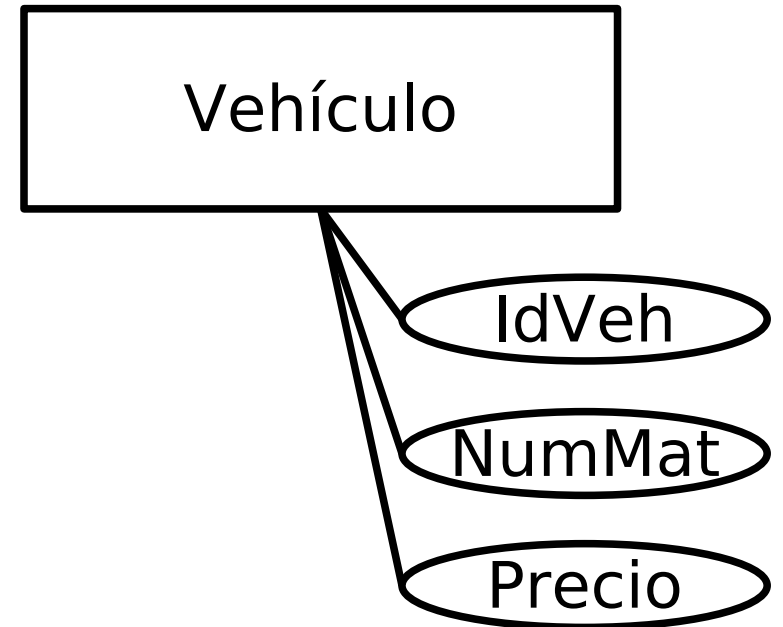


# ¿Cómo generalizar?

- Encontrar atributos en común
- Encontrar relaciones en común
- Siempre intentar generalizar



# Superclase



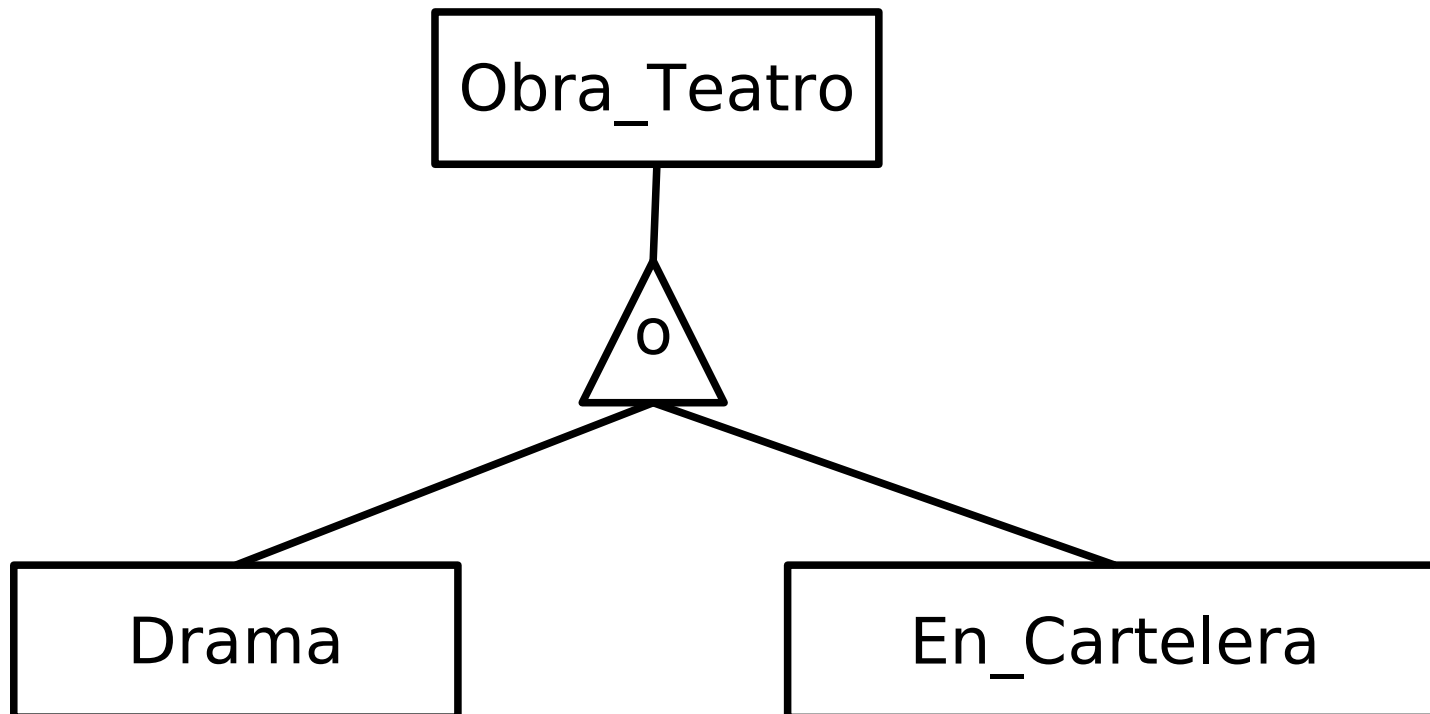
# Restricciones de especialización

# Tipos de subclase

- Definidos por un valor de atributo
  - Ej.: “TipoDeTrabajo=Secretaria”, entonces la entidad es Secretaria (sub-clase-de) Empleado
  - Se denominan: definidas por predicado
- No dependen de un valor de atributo
  - No hay una regla automática -- visible para el sistema
  - Se denominan: definidas por el usuario

# Restricción de disyunción

- ¿Puede pertenecer a dos subclases?
  - No: disjuntas [d]
  - Sí: solapadas (*overlapped*) [o]



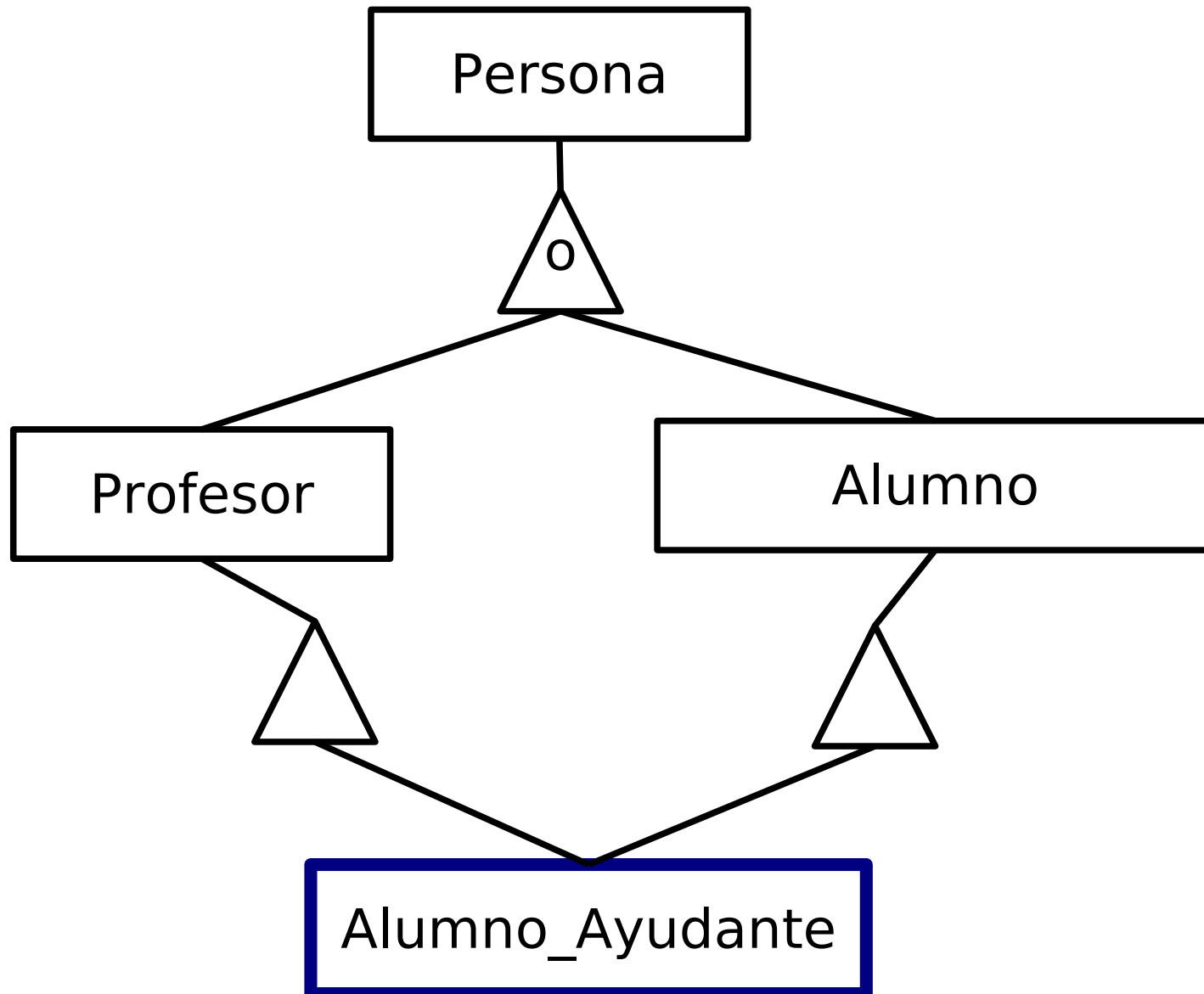
# Restricción de completitud

- Especialización total
  - Todos deben ser miembros de subclase
  - Equivalente a cuando en OO, una superclase es abstracta
- Especialización parcial
  - Pueden ser miembros de la superclase

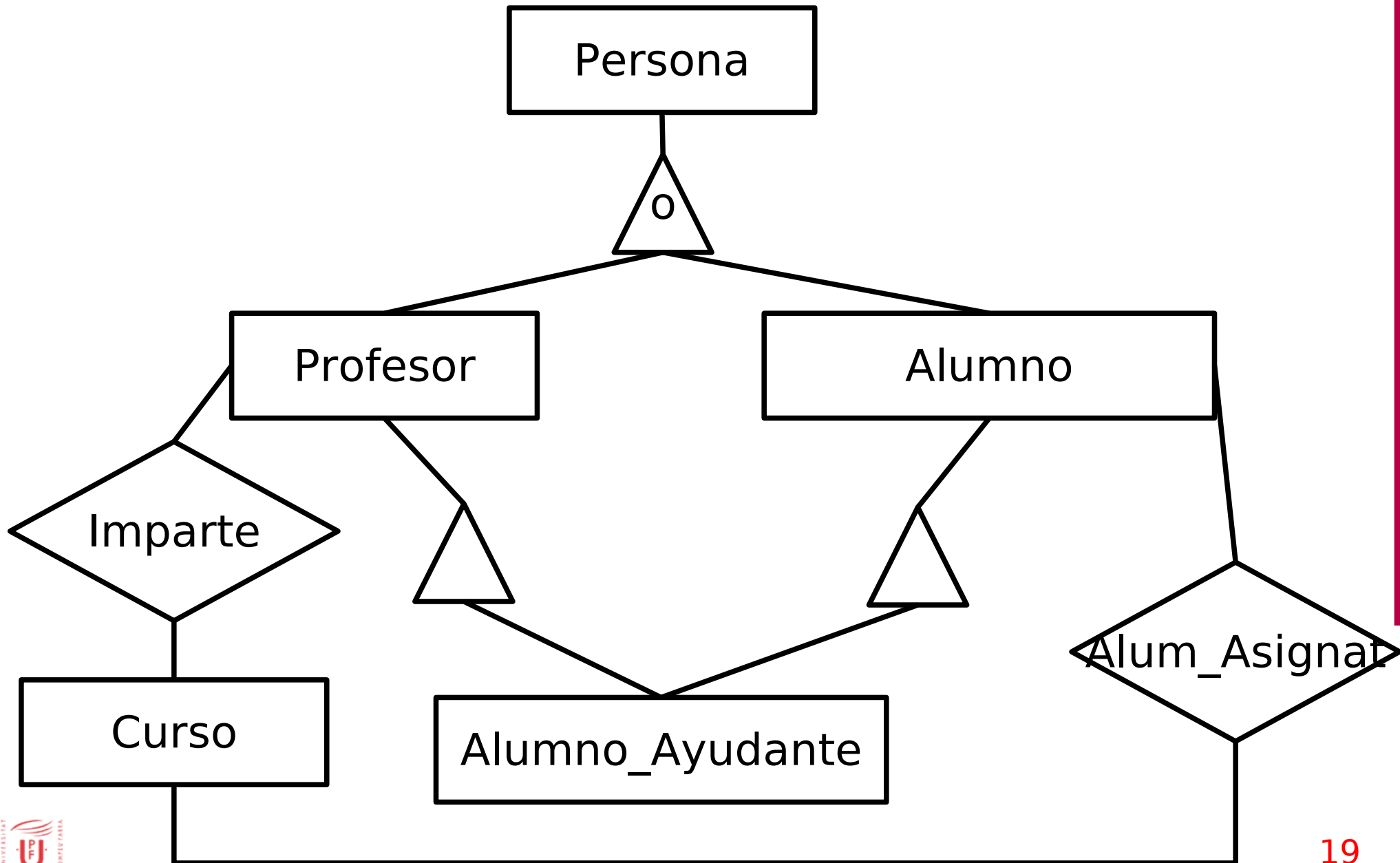
# Reglas derivadas de lo que hemos visto

1. Eliminar de una superclase significa eliminar de todas las subclases
2. Insertar en una superclase implica insertar en las subclases definidas por predicado
3. Insertar en una superclase cuando hay especialización total (una superclase abstracta) implica insertar en al menos una subclase

# Herencia múltiple



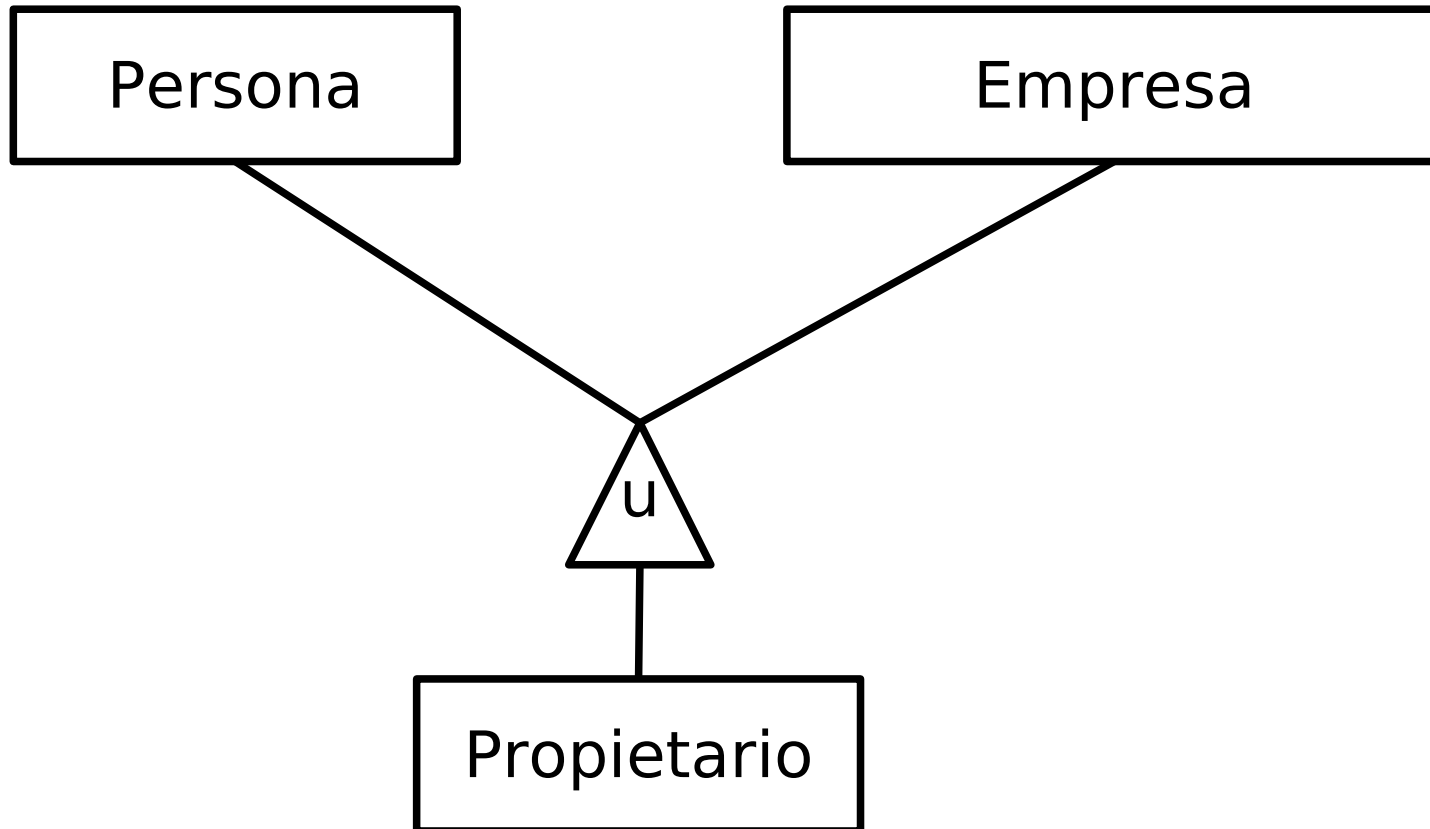
# Herencia múltiple



# Tipos unión

- Hasta el momento siempre hay una superclase
- Podría no ser el caso
- Ejemplo:
  - Un banco da créditos para compra de vehículos
  - Los créditos los da tanto a personas como a empresas

# Ejemplo tipo unión



La subclase unión no tiene atributos propios  
Tampoco hereda todos los atributos de las superclases,  
sino sólo los de una de ellas a la vez

# Ejemplo completo ...

- Universidad
- Se guardan todos los datos usuales de persona, incluyendo nombre y dirección como atributo compuesto
- Personas pueden ser profesores o alumnos. De profesores se guarda número de despacho, de alumnos año de ingreso.
- Profesores pertenecen a uno o varios departamentos, cada uno de los cuales tiene un director que es un profesor

# ... ejemplo completo ...

- Alumnos están inscritos en un departamento o varios y en uno o varios cursos
- Alumnos obtienen calificaciones en varias Pruebas, que pueden ser Práctica, Control o Exámen, con diferentes ponderaciones y promedios

# ... ejemplo completo.

- Departamentos tienen nombre, y pertenecen a una facultad que tiene nombre y decano que es un profesor
- Departamentos ofrecen asignaturas con secciones
- Alumnos pueden ser Alumnos\_Licenciados, con un asesor y varios profesores en su comité
- Profesor\_Investigador puede ser un Alumno\_Licenciado o un Profesor, dicta sección de una asignatura

# Principios diseño

- Correctitud: representa adecuadamente el modelo
- Evitar redundancia (ej.: relación estudio\_película + atributo estudio en película) ¿Por qué?
- Buscar simplicidad: entidades que no sean estrictamente necesarias
- Cuando aparece un ciclo, hay redundancia, sólo si es necesario por eficiencia (ej.1: alumno-curso-departamento, ej.2: actor-contrato-película actor-rol-película)

# Entidades eliminables

- Sólo tienen relaciones 1-1 con otra entidad
  - Ej.: direcciones de personas
  - Convertir en atributos (al menos durante el proceso de modelamiento)
- No pensar en las tablas. El modelo E-R es para conceptualizar
- Leer y re-leer el diagrama
  - Simular consultas típicas
  - Pensar en posibles extensiones
  - Estar preparado para explicar decisiones

# Resumen

- Buen diseño
  - **“Lo más simple posible pero no más simple que eso” -- A. Einstein**
  - Resiste el paso del tiempo
    - No es inmutable: es extensible
- Mal diseño
  - **Empeora con el tiempo**
  - Casos especiales, más y más entidades, etc. al final la única solución es deshacerse del diseño
    - Y del que lo diseñó :-)