

# Tema 2: Anàlisi de requisits i especificació

Anna Puig

Enginyeria Informàtica

Facultat de Matemàtiques i Informàtica,

Universitat de Barcelona

Curs 2019/2020



UNIVERSITAT DE  
BARCELONA

# Temari

1	Introducció al procés de desenvolupament del software	
2	Anàlisi de requisits i especificació	2.1 Anàlisi de requisits: Model FURPS+
3	Disseny	2.2 Especificació: Casos d'ús
4	Del disseny a la implementació	2.3 Especificació: User stories
5	Ús de frameworks de testing	2.4 Especificació: Model de Domini

# 2.4. Especificació: Model de Domini

---

2.4

Model de Domini

2.4.1

Definició i exemples

2.4.2

Conceptes o classes conceptuals

2.4.3

Associacions

2.4.4

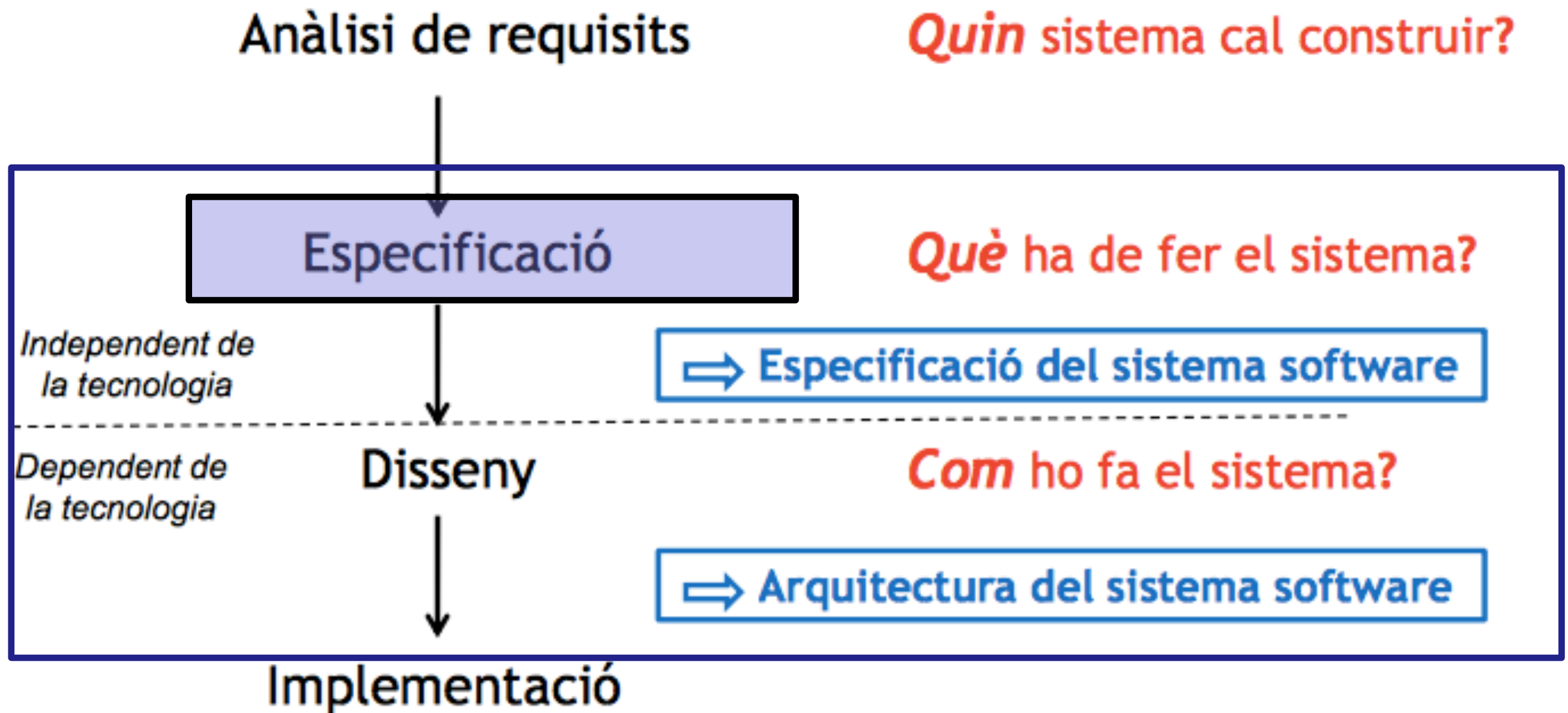
Generalització

2.4.5

Atributs

## 2.4. Especificació: Model de Domini

### Procés sistemàtic:



## 2.4.1 Model de Domini: Definició

**Model de Domini.** Representació *visual* de classes conceptuais o objectes del món real en un domini d'interès.

- També es coneix com: Model conceptual, Model d'objectes del domini o Model d'objectes d'anàlisi
- L'objectiu és entendre el domini del problema i el sistema que es vol implementar
- S'utilitzen els **casos d'ús** de la fase de captura de requisits

## 2.4.1 Model de Domini: Definició

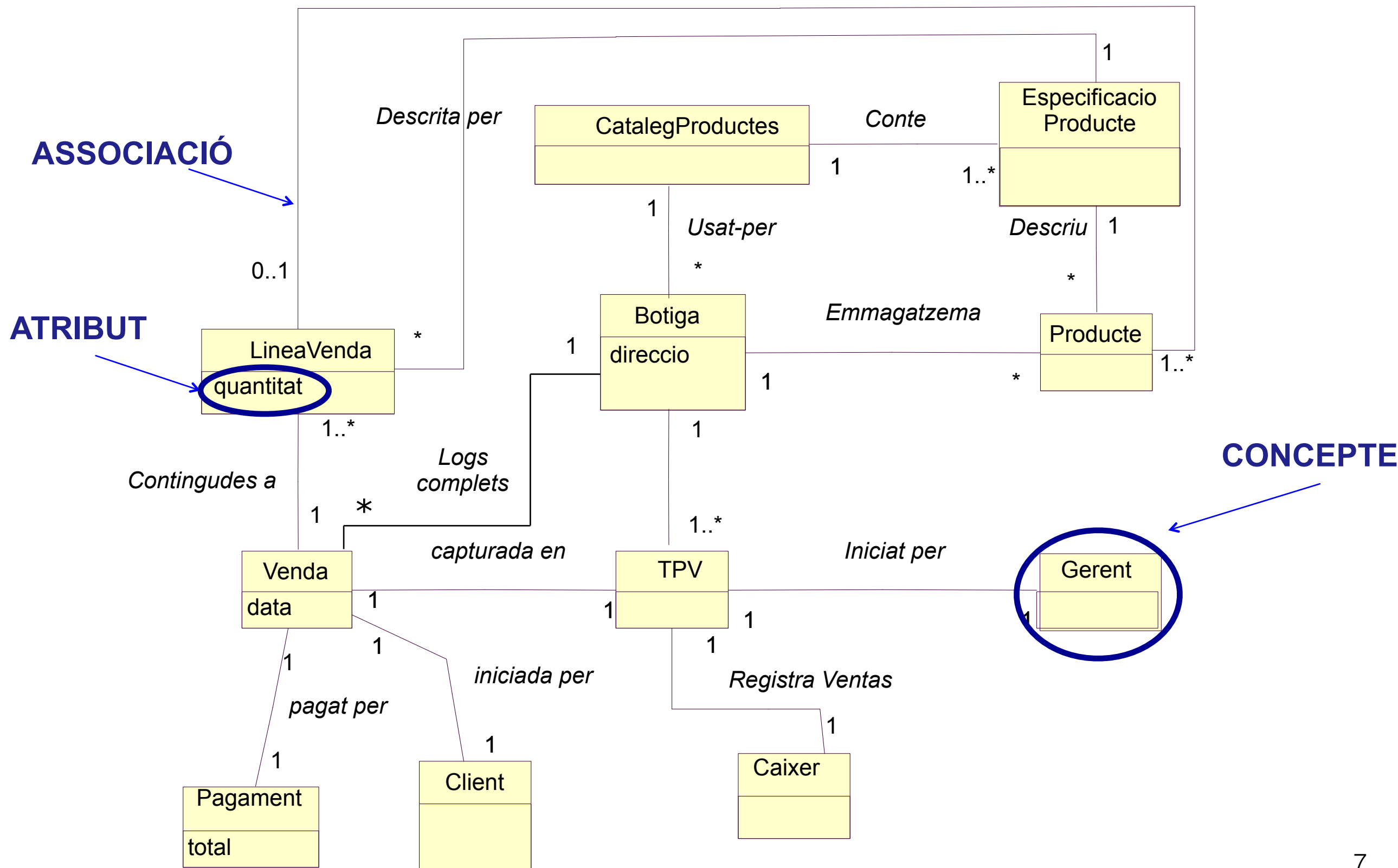
---

- El model de domini s'il·lustra mitjançant un conjunt de diagrames de classes en UML en el ***que no apareixen operacions***.

Conté:

- **Classes conceptuais** o objectes del domini
  - Relacions (**associacions**) entre classes conceptuais
  - **Atributs** de les classes conceptuais
- 
- El model de domini constitueix un **diccionari** visual de les abstraccions rellevants

# Exemple venda de productes



# Com construir un model de domini?

---

1. Elaborar una **llista de classes conceptuais candidates** usant la tècnica de la llista de categories de classes conceptuais i la identificació de sintagmes nominals en els requisits
2. Dibuixar-les en un model de domini
3. Afegir les **associacions** necessàries per emmagatzemar les relacions de les que és necessari guardar memòria
4. Afegir els **atributs** necessaris per satisfer els requisits d'informació



## 2.4.1. Consideracions

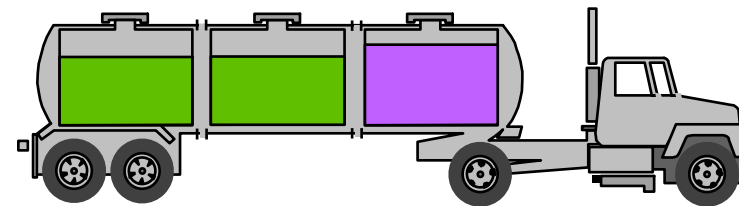
---

- S'ha d'actuar com un cartògraf
  - Usar els noms existents en el territori
  - Excloure les característiques irrelevantes
  - No afegir res que no estigui allà
- És un error comú oblidar-se classes i descobrir-ho a posteriori
  - És millor sobreespecificar-lo amb una gran quantitat de classes conceptuais de gra fi.
- No associar directament només classes amb informació
  - Poden existir classes que només continguin comportament (sense atributs)

## 2.4.2. Conceptes

**Conceptes o classes conceptuais.** Entitats que representen elements físics, conceptuais i estats del problema

- Entitat física



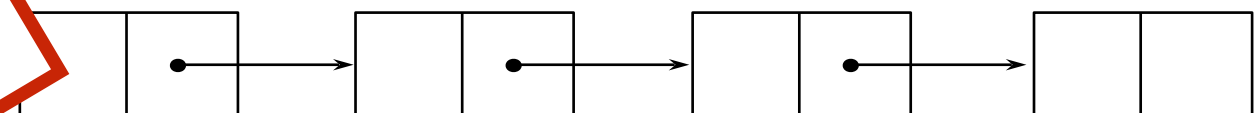
Camió

- Entitat conceptual



Procés  
químic

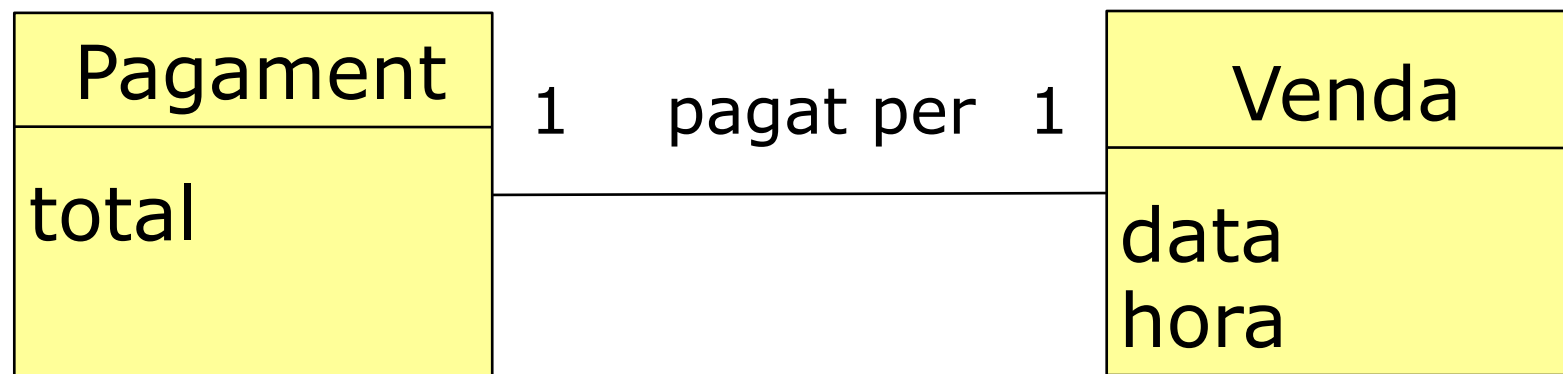
- Entitat de software



Llista Enllaçada

## 2.4.2. Conceptes

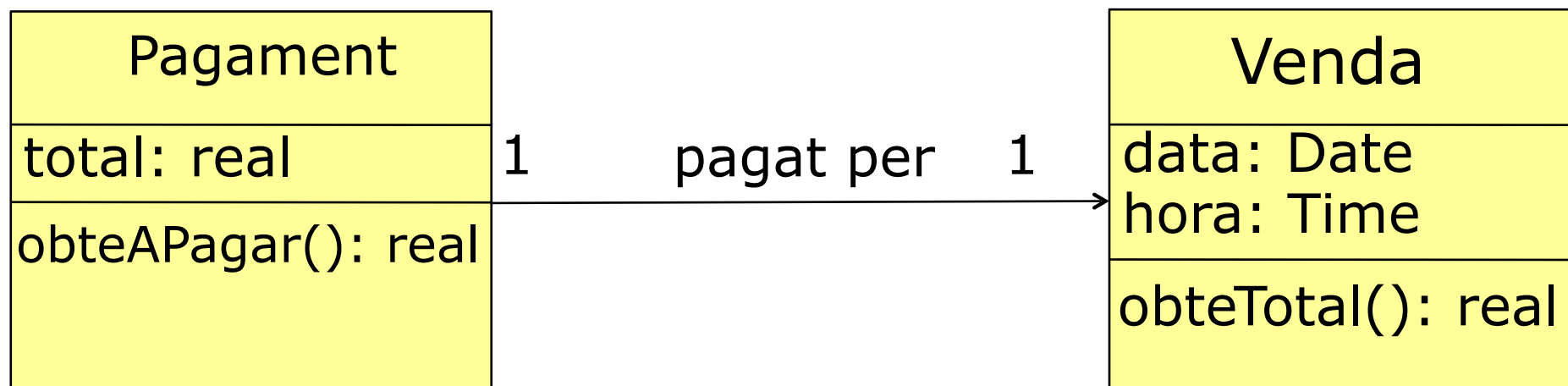
Perspectiva conceptual amb el **Model de Domini (Especificació)**



Model essencial que visualitza conceptes del mon real

---

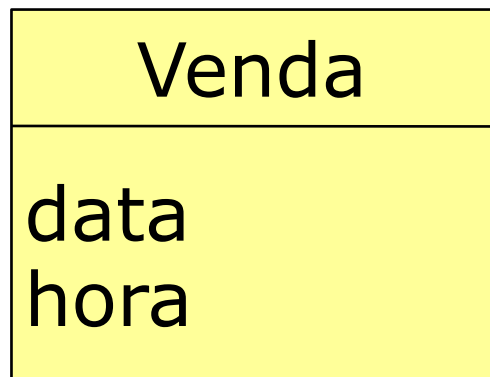
Perspectiva d'especificació amb el diagrama de classes: **Model de Disseny o Diagrama de Classes**



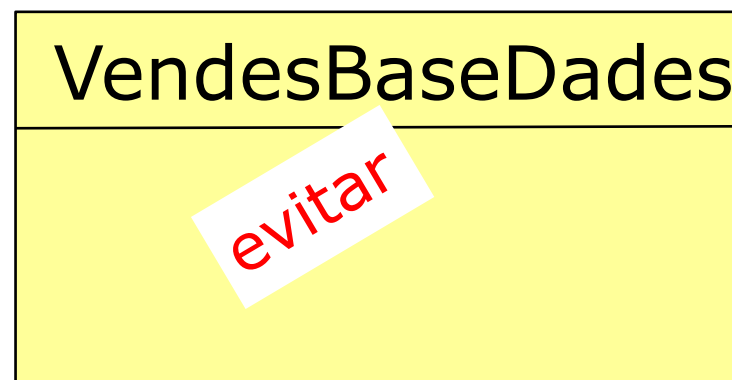
Model específic que visualitza components software

## 2.4.2. Conceptes: exemples

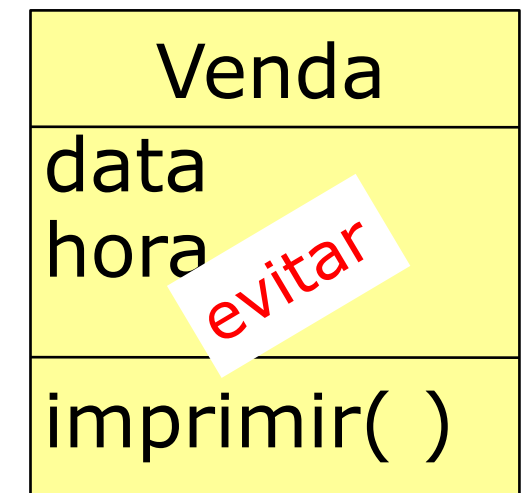
- Un model de domini **NO** representa components software
- No s'han de representar artefactes software, responsabilitats o mètodes



(a) Concepte de domini



(b) Artefacte software



(c) Classe software

## 2.4.2. Conceptes

---

- Formalment, una classe conceptual es pot considerar en termes de:
  - **Símbol**: paraules o imatges que representen una classe conceptual
  - **Intenció**: la definició de la classe conceptual. Descripció del que fa la classe conceptual
  - **Extensió**: El conjunt d'exemples al que la classe conceptual aplica
- Estratègia per identificar-les
  - Usar una llista de categories de classes conceptuais
  - Identificar Sintagmes Nominals

# Llista de categories de classes conceptuais (I)

<b>Categoria de classe conceptual</b>	<b>Exemples</b>
Objectes físics o tangibles	Capsa, avió
Especificacions, dissenys o descripcions de les coses	EspecificacioDeProducte DescripcioDeVol
Llocs	Botiga, Aeroport
Transaccions	Venda, Pagament, Reserva
Línies de transacció	LiniaDeVenda
Rols de gent	Caixer, Pilot
Contenidors d'altres coses	Magatzem, paquet, aeroplà
Coses contingudes en un contenidor	Passatger, ítem o article

# Llista de categories de classes conceptuals (II)

<b>Categoria de classe conceptual</b>	<b>Exemples</b>
Altres sistemes informàtics o electromecànics externs al sistema	SistemaDAutoritzacioDeCredits ControlDeTraficAeri
Noms de conceptes abstractes	Ànsia, Aerofòbia
Organitzacions	DepartamentDeVendes, CompanyiaAeria
Esdeveniments	Venda, Pagament, Reunió, Vol, Accident, Aterratge
Regles i polítiques	PoliticaDevolucions, PoliticaDeCancelacions

# Llista de categories de classes conceptuais (III)

<b>Categoria de classe conceptual</b>	<b>Exemples</b>
Catàlegs	CatalegDeProductes CatalegDePeces
Registres de finances, treball, contractes, assumptes legals	Rebut, ContracteDeTreball, RegistreDeManteniment
Instruments i serveis financers	LiniaDeCredit, Stock
Manuals, documents, llibres, fulls de referència	LlistaDiariaDeCanvisDePreu, ManualDeReparacions

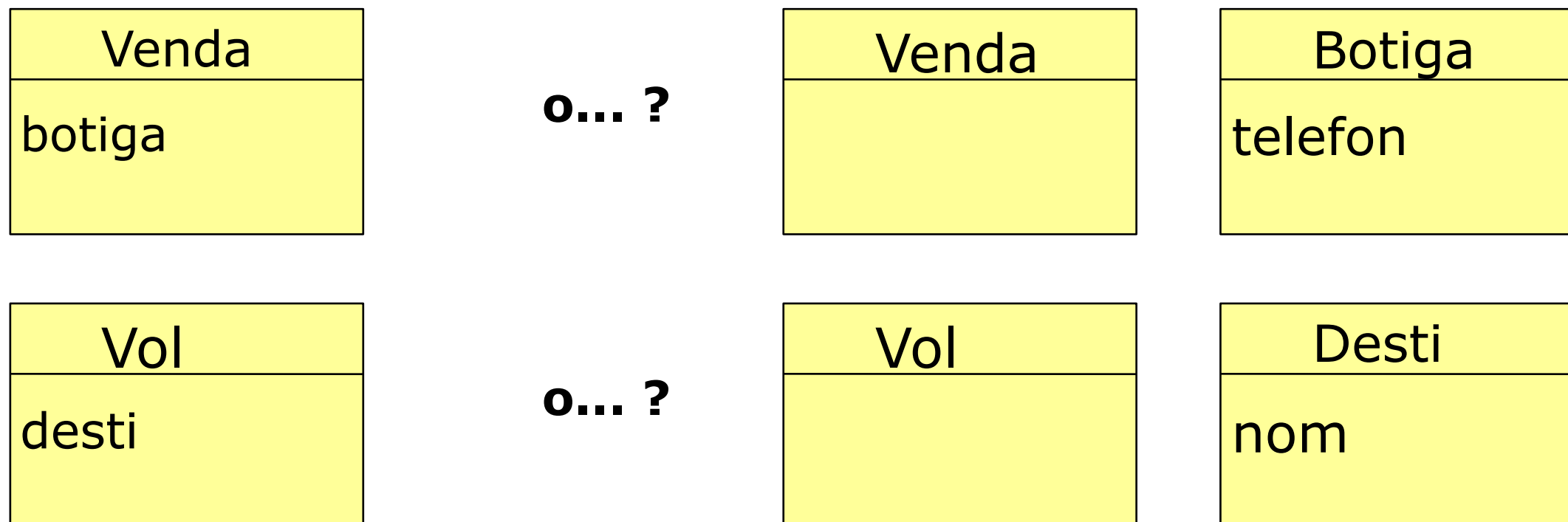


# Diferenciar entre concepte i atribut

**L'error més comú al construir un model de domini és representar alguna entitat com un atribut quan hauria de ser un concepte**

① Si per un ens X no pensem en ell com un número o un text en el món real, llavors X hauria de ser una classe conceptual i no un atribut

*En cas de dubte, considereu un concepte separat.*



# Exercici: UB Cultura

---

En el centre cultural UBCultura es fan préstecs de discs i llibres. Pels dos es guarda informació general, com el seu codi, el títol i l'autor. Quan es tracta de llibre també es guarden el número de pàgines. En el cas dels discs, es guarda el nom de la discogràfica.

En UBCultura arriben clients a qui es demana el seu DNI i el nom. Aquests clients realitzen una sèrie de peticions de préstec de discs o llibres (com a molt en poden fer 5). Per a cada petició es guarda la data d'inici i el final del préstec.

**Objectiu:** Realitzar una abstracció en detall de les **classes conceptuais** que estimes apropiades pel problema.

# Exercici: UB Cultura

En el **centre cultural UBCultura** es fan **prèstecs** de **discs** i **llibres**. Pels dos es guarda informació general, com el seu **codi**, el **títol** i **l'autor**. Quan es tracta de llibre també es guarden el **número de pàgines**. En el cas dels discs, es guarda el **nom de la discogràfica**.

En UBCultura arriben **clients** a qui es demana el seu DNI i el nom. Aquests clients realitzen una sèrie de peticions de **prèstec** de discs o llibres (com a molt en poden fer 5). Per a cada petició es guarda la **data d'inici** i el **final** del préstec.

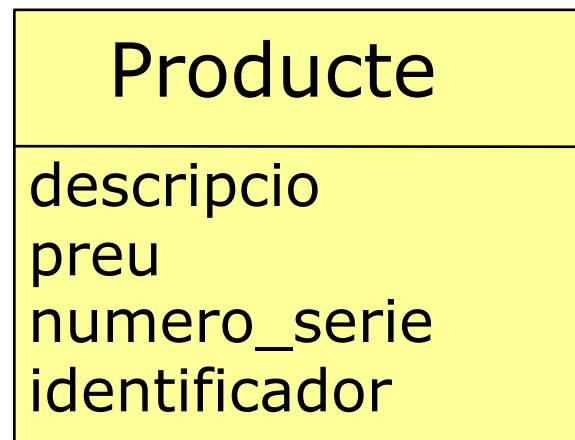
**Objectiu:** Realitzar una abstracció en detall de les **classes conceptuais** que estimes apropiades pel problema.

## 2.4.2. Classes conceptuais d'especificació o descripció

- Una classe **EspecificacióDeX** descriu un X
- S'ha d'afegir una classe d'especificació quan:
  - Es necessari tenir la **descripció d'un ítem** o servei amb independència de l'existència d'un exemplar
  - **Eliminar** instàncies d'un cert tipus resulta una **pèrdua d'informació** que ha de ser mantinguda
  - **Redueix la informació duplicada** o redundant

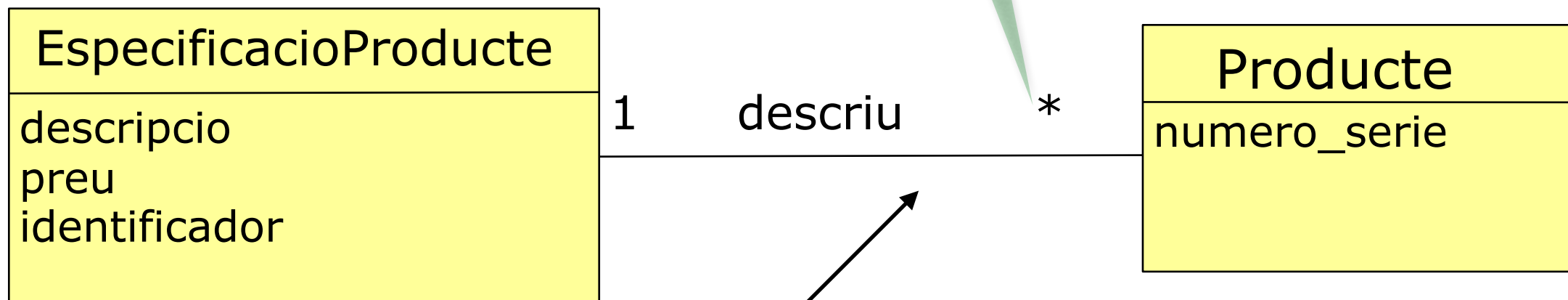
## 2.4.2. Classes conceptuais d'especificació o descripció

**Pitjor**



Noteu la multiplicitat "\*"

**Millor**

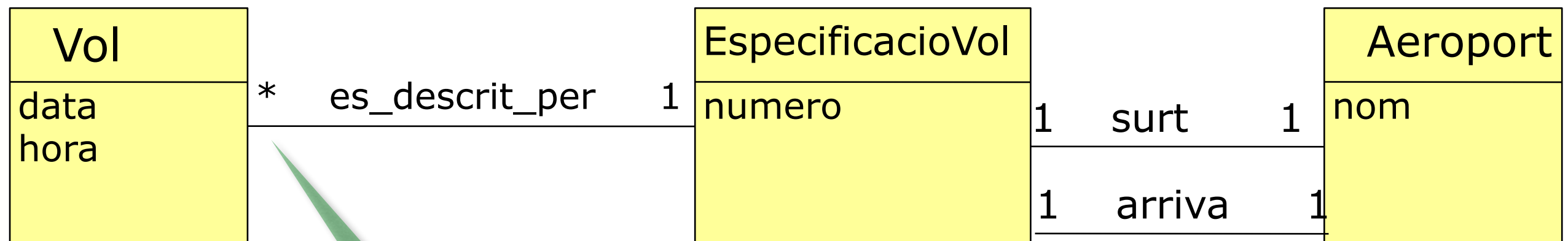


ASSOCIACIÓ

## 2.4.2. Classes conceptuais d'especificació o descripció



**Pitjor**



**Millor**

Noteu la  
multiplicitat "\*"

## 2.4.3. Associacions

Una **associació** és una relació entre classes conceptuais (instàncies d'aquests tipus) que indica alguna connexió interessant i significativa

- Existeixen 3 tipus de relacions entre classes:
  - Associació
  - Agregació
  - Composició
- Es representa per una línia entre classes amb un nom d'associació
  - Les associacions són inherentment bidireccionals (no en software o implementació)
  - Existeix un símbol per indicar la direcció de lectura
  - Els extrems contenen expressions de **multiplicitat**

## 2.4.3. Associacions



- Tota associació té un nom
- S'ha de donar nom a les associacions basant-se en el format:

NomDeClasse – PerífrasisVerbal – NomDeClasse

On la perífrasis verbal crea una seqüència que és llegible i té significat en el context del model

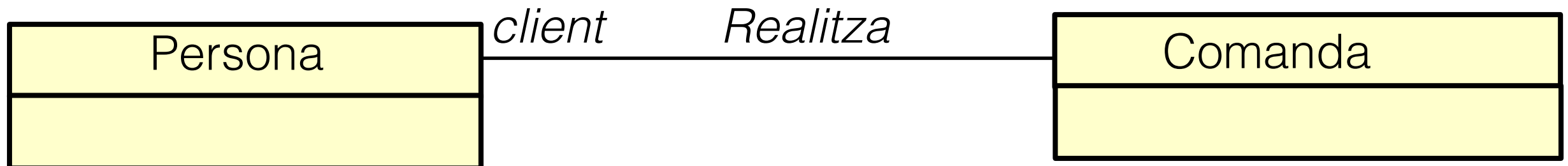
- Per defecte es llegeixen d'esquerra a dreta i de dalt a baix



## 2.4.3. Associacions: Rol

Cada extrem d'una associació s'anomena **rol**

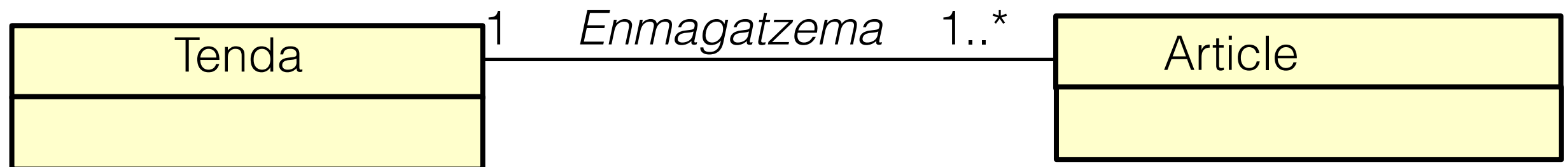
- El rol s'usa per millorar la comprensió del diagrama (opcional)
- Una o ambdues classes que connecta l'associació pot tenir un rol assignat



## 2.4.3. Associacions: Multiplicitat

La **multiplicitat** indica el número d'instàncies d'una classe relacionada a una instància d'una altra classe

- El valor de la multiplicitat comunica quantes instàncies es permeten associar amb una altra en un moment concret en el temps, no al llarg d'un interval de temps

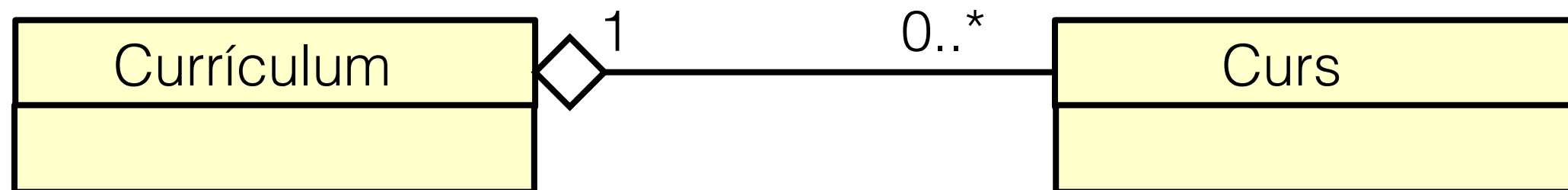


## 2.4.3. Associacions: Multiplicitat

*	T	Zero o més (molts)
1..*	T	Un o més
1.. 40	T	D'un a 40
5	T	Exactament 5
3, 5, 8	T	Exactament 3, 5 o 8

## 2.4.3. Associacions: Agregació

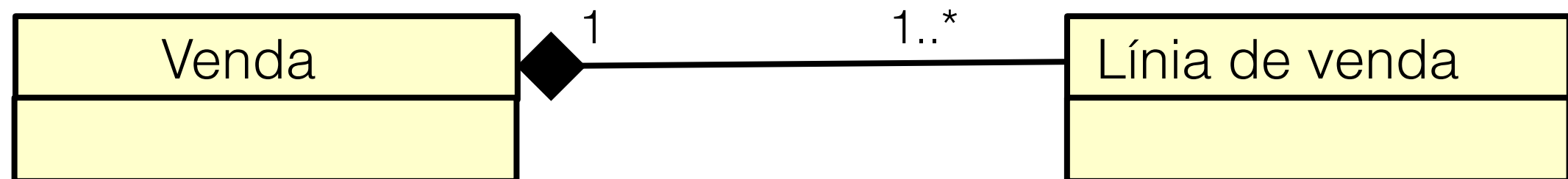
**L'agregació** és una associació especialitzada en la qual un TOT es relaciona amb les seves parts.



## 2.4.3. Associacions: Composició

La **composició** és una associació especialitzada d'una agregació

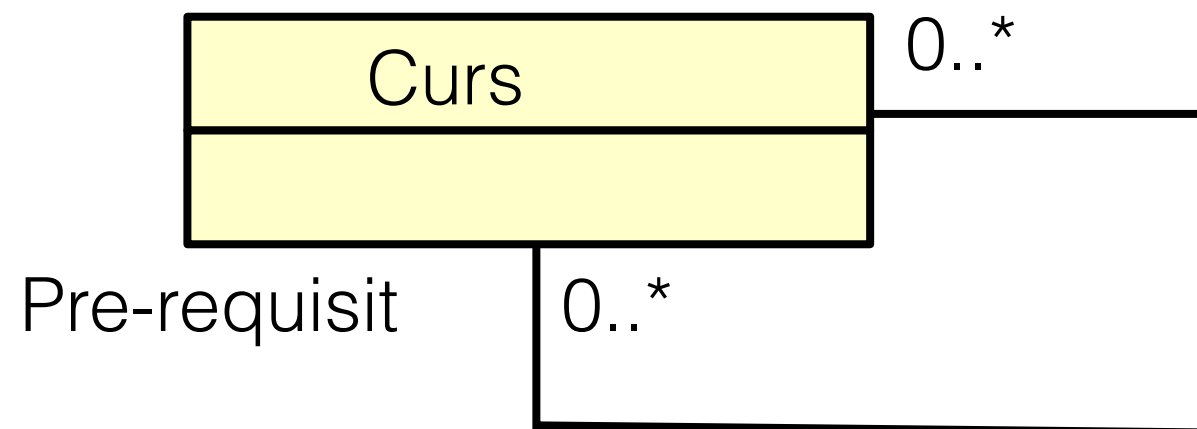
- Les parts es creen i es destrueixen amb el tot



## 2.4.3. Associacions reflexives

En una **associació reflexiva**, els objectes d'una mateixa classe estan relacionats

- Indica que els objectes d'una mateixa classe col·laboren entre ells per aconseguir una responsabilitat

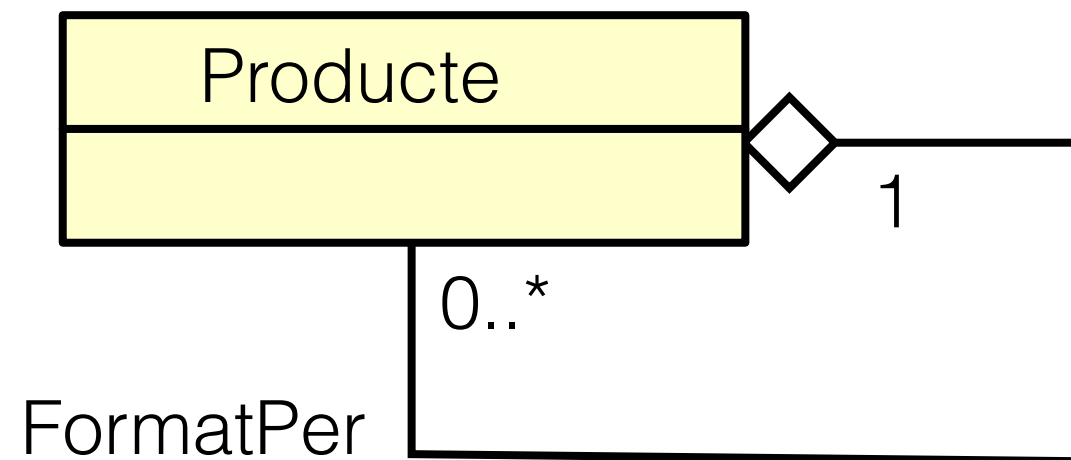


Un curs pot tenir molts pre-requisits

Un curs pot ser pre-requisit per molts altres cursos

## 2.4.3. Associacions reflexives

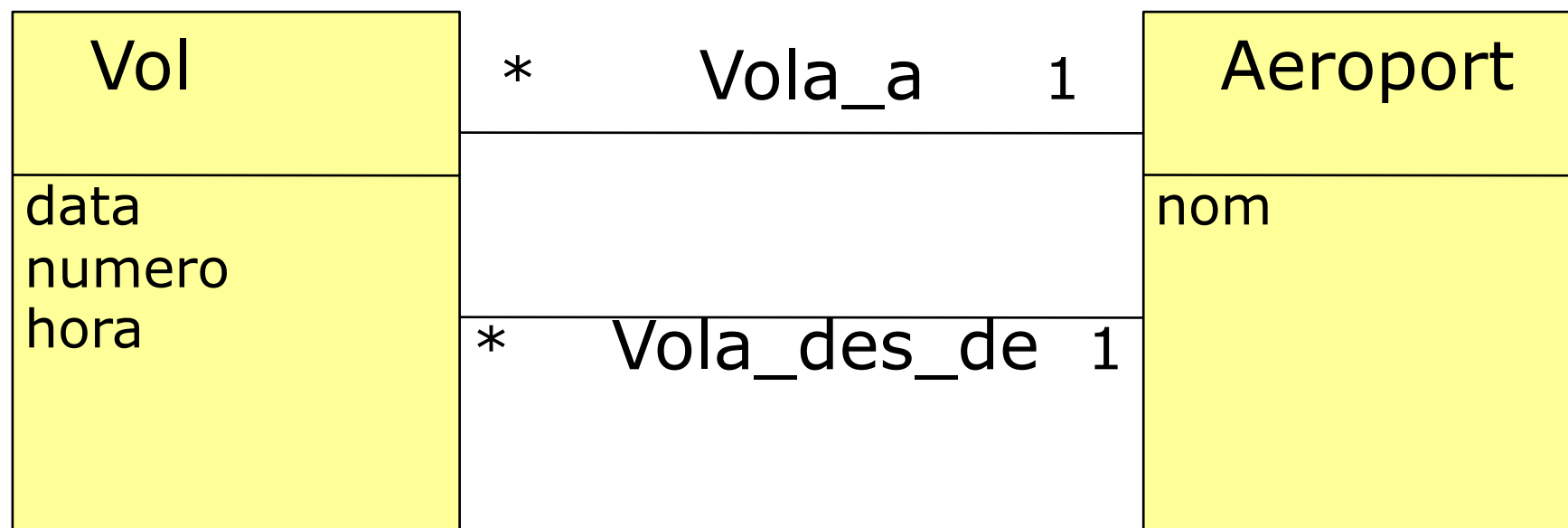
- En una agregació reflexiva els objectes d'una mateixa classe poden ser part del tot



Un producte pot estar compost de molts altres productes

## 2.4.3. Associations multiples

Dos classes poden tenir **múltiples associacions** entre elles





# Com identificar associacions

---

## Recomanacions

- Centrar-se en aquelles associacions per les que el coneixement de la relació es necessita **preservar** per algun temps
- És més important identificar classes conceptuais que identificar associacions
- Masses associacions tendeixen a confondre enlloc d'aclarir
- S'ha d'evitar mostrar associacions redundants o derivables

# Llista associacions fonamentals

<b>Categoria</b>	<b>Exemples</b>
A és físicament una part de B	Ala-Avió, Mà-Cos
A és lògicament una part de B	LíniaDeFactura-Factura, Capítol-Llibre
A està físicament continguda en B	Passatger-Avió, Producte-Embalatge
A està lògicament continguda en B	DescripcióDeProducte-Cataleg, Vol-HorariDeVols
A és conegut / registrat/ reportat/ capturat en B	Factura-LlibreDeFactures

# Llista associacions comuns (I)

<b>Categoria</b>	<b>Exemples</b>
A és una descripció de B	Producte- DescripcioDeProducte, Vol-DescripcioDeVol
A és un element de línia d'una transacció de B	Factura-LiniaDeFactura, Comanda-LiniaDeComanda
A és un membre de B	Pilot-LiniaAeria, Professor- Departament
A és una subunitat organitzacional de B	Departament-Universitat, Delegació-Empresa
A usa o gestiona B	Pilot-Avió, Professor-Aula
A es comunica amb B	Client-Caixaer, Alumne-Professor

# Llista associacions comuns (II)

<b>Categoria</b>	<b>Exemples</b>
A està relacionat amb una transacció B	Client-Pagament, Passatger-Tiquet
A és una transacció relacionada amb un altra transacció B	Pagament-Venda, Reserva-Cancelació
A és el següent de / està junt a B	LiniaDeFactura-LiniaDeFactura, Ciutat-Ciutat
A és l'amo de B	LiniaAeria-Avió
A és un esdeveniment relacionat amb B	Venda-Client, Sortida-Vol

# Associacions i implementació

---

## **Durant el modelat de domini:**

- una associació *no es refereix a flux de dades*, variables d'instàncies o connexions d'objectes en una solució software
- una associació es refereix a una relació en sentit conceptual.
- Algunes associacions del model de domini no seran necessàries per la implementació
- El model de domini només ha de mantenir-se al dia (com qualsevol altre artefacte) si això té sentit en el projecte

# Exercici: UB Cultura

---

En el centre cultural UBCultura es fan préstecs de discs i llibres. Pels dos es guarda informació general, com el seu codi, el títol i l'autor. Quan es tracta de llibre també es guarden el número de pàgines. En el cas dels discs, es guarda el nom de la discogràfica.

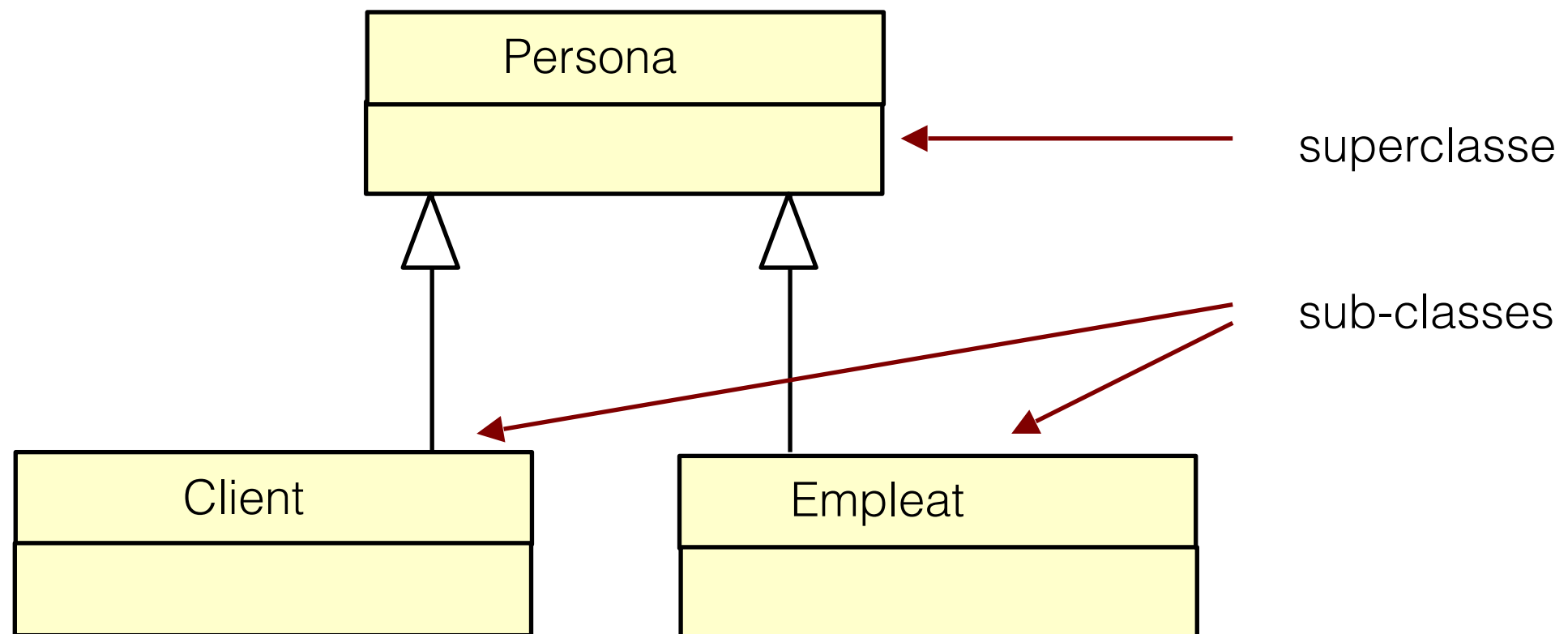
En UBCultura arriben clients a qui es demana el seu DNI i el nom. Aquests clients realitzen una sèrie de peticions de préstec de discs o llibres (com a molt en poden fer 5). Per a cada petició es guarda la data d'inici i el final del préstec.

**Objectiu:** Definir les **associacions** entre les classes conceptuais.

## 2.4.4. Generalització

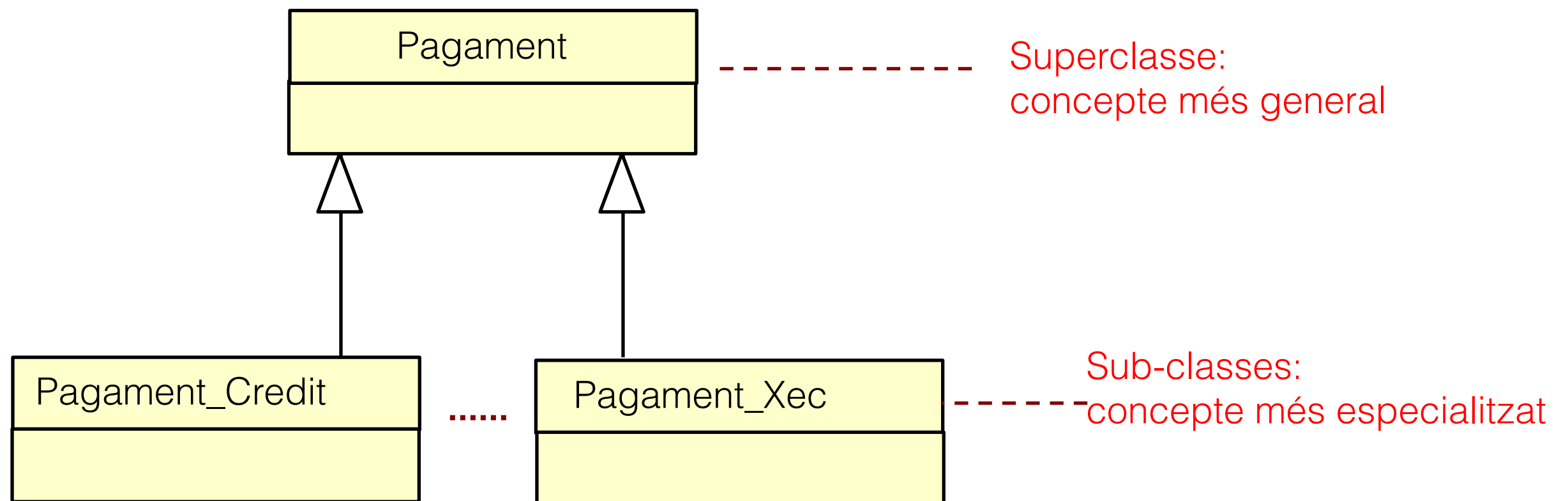
**Generalització:** És l'activitat d'identificar factors comuns entre conceptes

- L'herència es representa a través d'una associació de generalització
  - Es defineixen relacions entre superclasse i sub-classe



## 2.4.4. Generalització

- L'associació de **generalització** representa una jerarquia de classes.
  - Jerarquia de generalització /especialització





## 2.4.4. Generalització: Regles

---

Regles necessàries per la definició d'una subclasse

- **Regla del 100%:** El 100% de la definició d'una classe conceptual ha de ser aplicable a les seves sub-classes.
  - És a dir, per la subclasse tenen sentit el 100% dels atributs i les associacions de la superclasse
- **Regla És-un:** Tots els membres d'una subclasse han de ser membres de la superclasse.
  - *Sub-classe és una superclasse*

# Quan crear una subclasse

---

1. Si la subclasse té atributs addicionals interessants
2. Si la subclasse té associacions addicionals interessants
3. Si s'ha d'operar, manegar, reaccionar o manipular de forma diferent i interessant el concepte de la nova subclasse del de la superclasse o altres subclasses
4. Si el concepte de la subclasse representa un ens animat que respon de forma diferent i interessant que la superclasse o altres subclasses

# Quan crear una superclasse

---

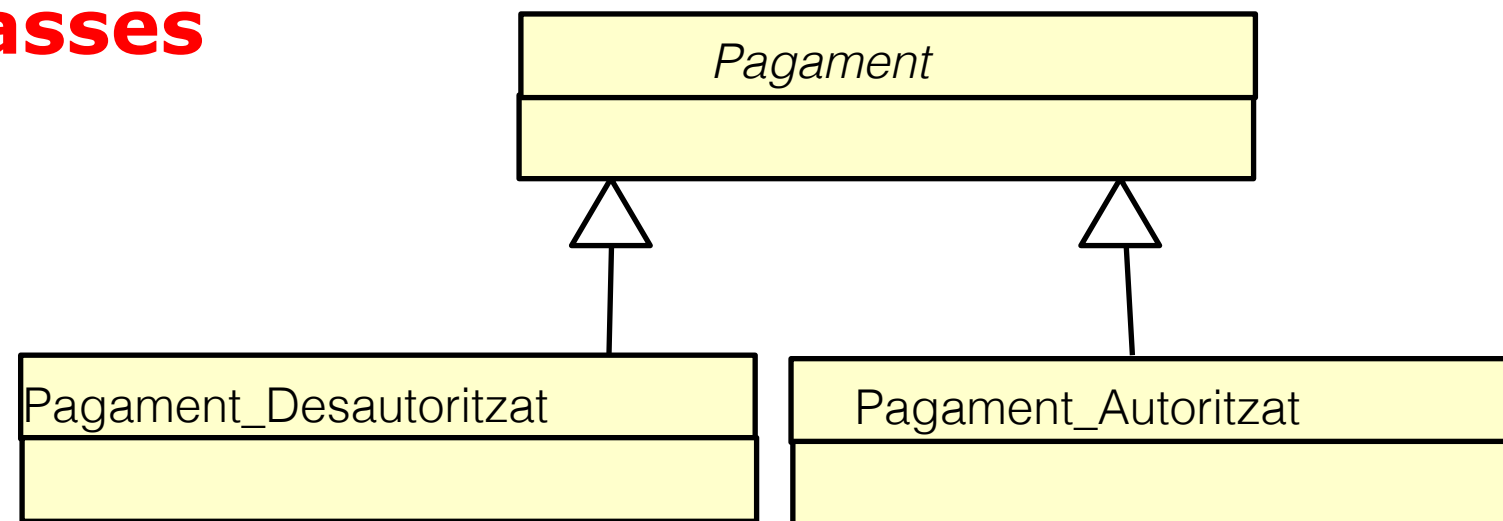
1. Les subclasses conceptuais potencials representen subconceptes d'un concepte similar
2. Les subclasses compleixen amb la regla del 100% i la regla es-un/a
3. Totes les subclasses comparteixen un **atribut** del que es pot treure “factor comú” i incloure'l en la superclasse
4. Totes les subclasses comparteixen una **associació** de la que es pot treure “factor comú” i incloure-la en la superclasse

# Classes conceptuais abstractes

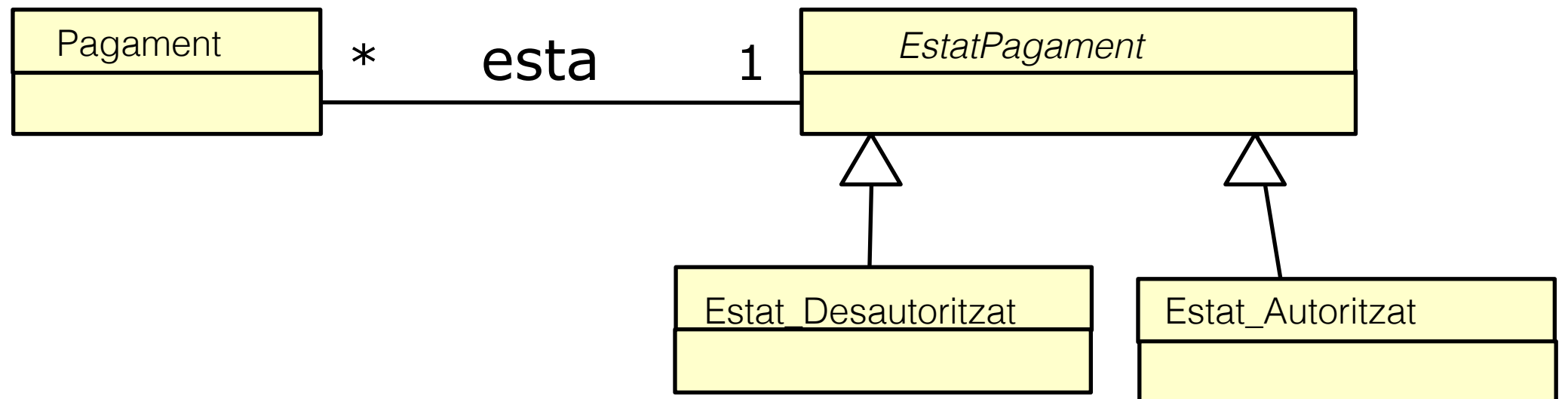
- Si tots els membres d'una classe C han de ser també membres d'una subclasse, llavors la classe C és una classe conceptual abstracta
- Exemple:
  - Superclasse abstracta:  
*Triangle* ← Escalens, Equilàter, Isòsceles
  - Superclasse no abstracta:  
Temàtica ← Misteri, Policíac
- En UML s'indica el nom d'una classe conceptual abstracta ***en cursiva***

# Modelant estats

**No és útil perquè les subclasses canvien l'estat intern de la superclasse**



**Millor**



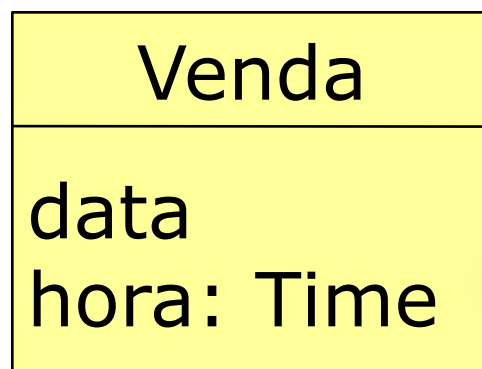
## 2.4.5. Atributs

Un **atribut** és un valor de dades lògic d'un concepte o classe conceptual

- En un model de domini s'han d'incloure:
  - Els atributs per als que els requisits suggereixen o impliquen una necessitat de recordar informació

## 2.4.5. Atributs

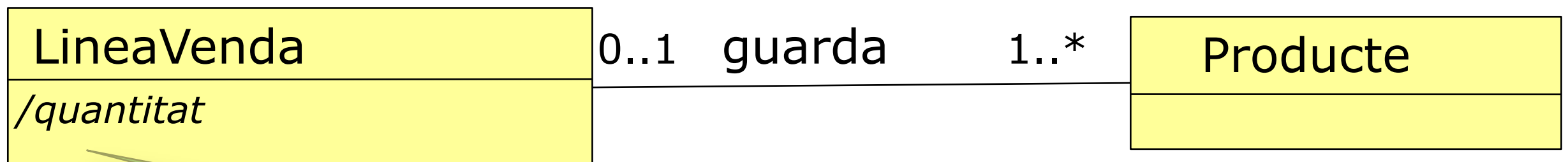
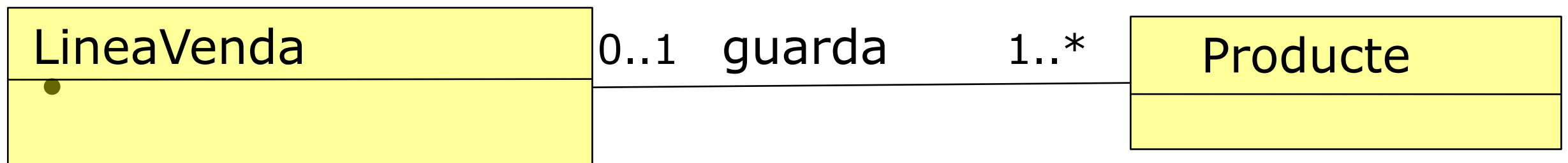
- Els **atributs** d'un model de domini han d'ésser preferiblement simples
- Opcionalment es pot mostrar el **tipus**
- Els tipus de dades més comuns són:
  - booleà, Date, numero (int o real), cadena de caràcters (string), Time
- Altres tipus de dades inclouen:
  - Adreça, color, geometries (punt, rectangle), numero de telèfon, Numero de seguretat social, Codi de producte universal, codi postal, tipus enumerats



Atributs de la classe  
venda

## 2.4.5. Atributs derivats

- Un atribut **derivat** és aquell que pot ser inferit a partir d'una altra informació del model



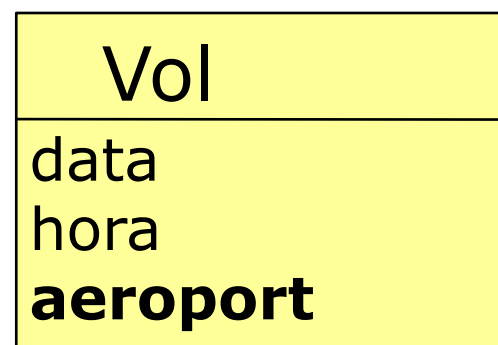
Atribut derivat a partir de la multiplicitat



## 2.4.5. Atributs

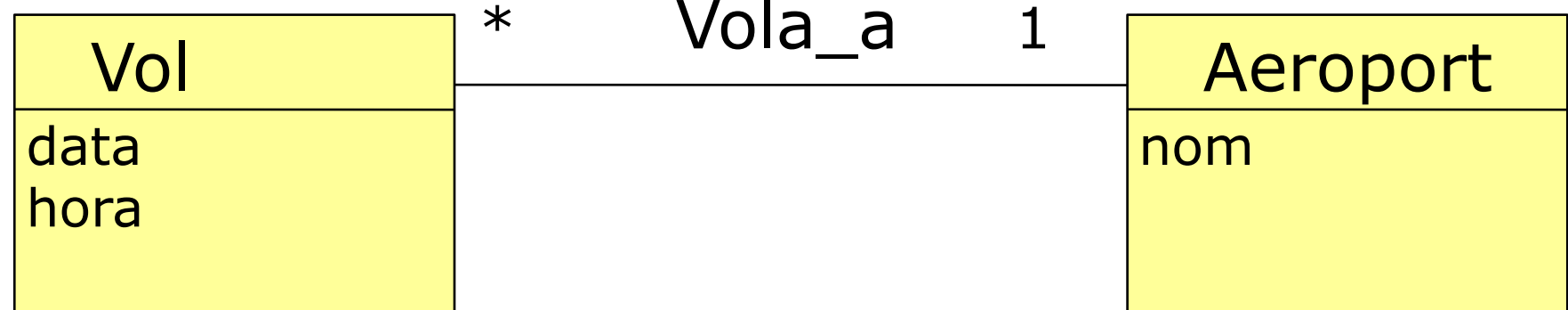
- Les classes conceptuais han de relacionar-se mitjançant associacions, no mitjançant atributs
- Si el valor de l'atribut és un concepte complex del domini s'ha de representar mitjançant una classe conceptual i no com un atribut

**Pitjor**



aeroport no és un atribut **"simple"**

**Millor**

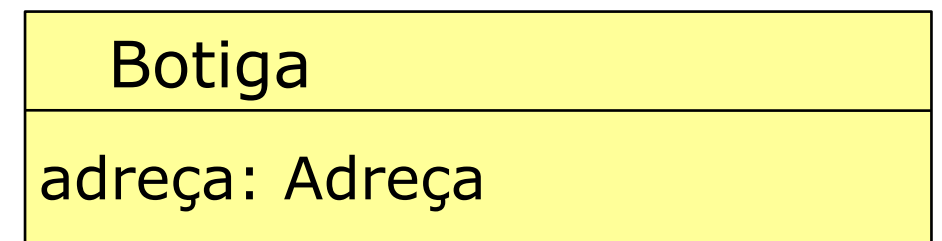
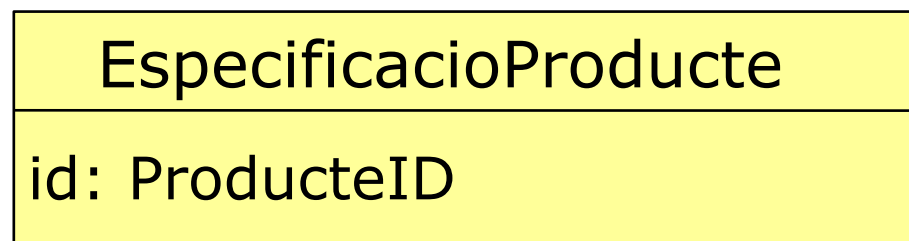
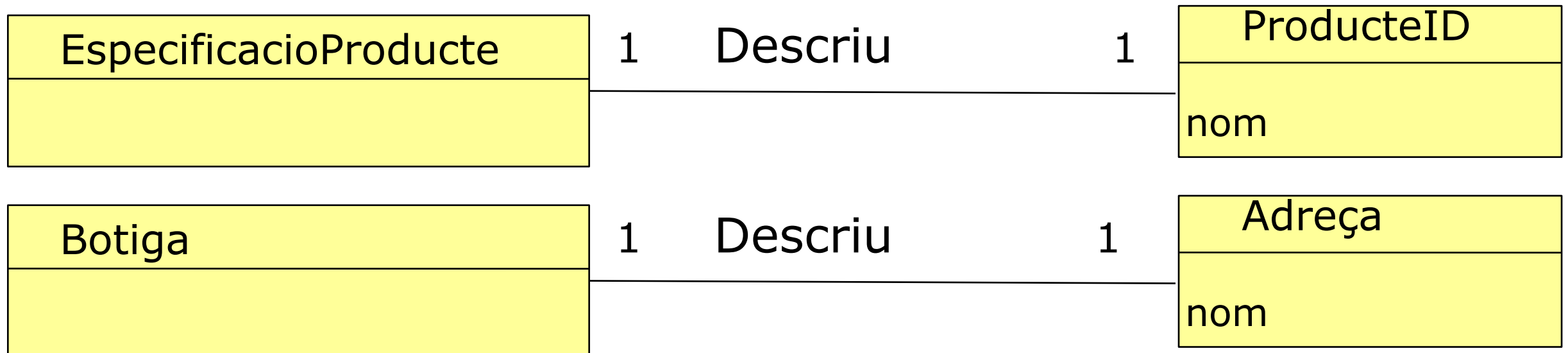


## 2.4.5. Atributs: Tipus de dades no primitius

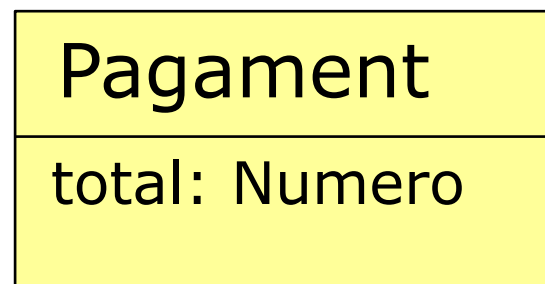
- Un tipus de dada que pot ser considerat com primitiu s'ha de representar com una **classe** si:
  - Es compon de diferents seccions
    - *Exemple: Direcció, Telèfon, NomDePersona*
  - Té operacions associades amb ell:
    - *Exemple: Validació en Numero de seguretatSocial, NumerodeTargeta*
  - Té altres atributs:
    - *Exemple: PreuPromocional té data d'inici i data final*
  - És una quantitat com una unitat:
    - *Exemple: Diners*

## 2.4.5. Atributs: Tipus de dades no primitius

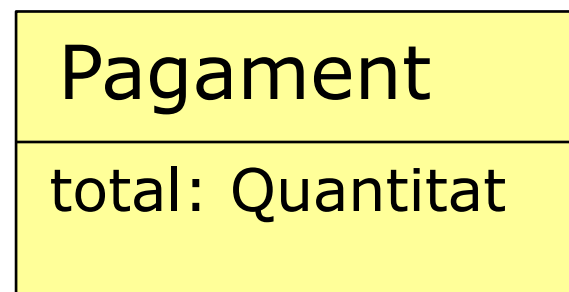
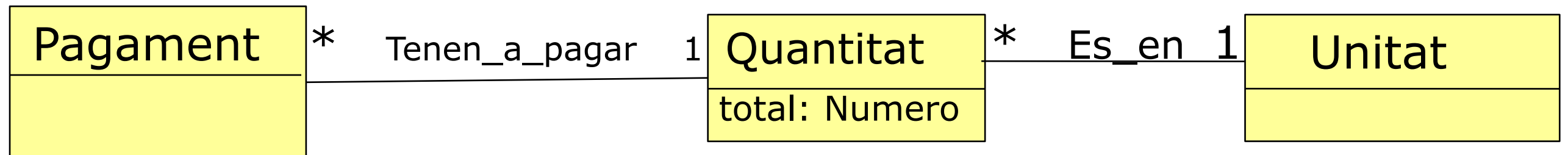
- Seleccionar l'opció que faciliti la comunicació



# Exemple: Modelant quantitats i unitats



**No és útil**



**Millor qualsevol d'aquestes dues opcions**

# Exercici: UB Cultura

---

En el centre cultural UBCultura es fan préstecs de discs i llibres. Pels dos es guarda informació general, com el seu codi, el títol i l'autor. Quan es tracta de llibre també es guarden el número de pàgines. En el cas dels discs, es guarda el nom de la discogràfica.

En UBCultura arriben clients a qui es demana el seu DNI i el nom. Aquests clients realitzen una sèrie de peticions de préstec de discs o llibres (com a molt en poden fer 5). Per a cada petició es guarda la data d'inici i el final del préstec.

**Objectiu:** Completar les classes conceptuais amb els **atributs** que pensis que són propietats pel problema plantejat.