

Tema 4 **Model de disseny**

Maria Salamó Llorente

Disseny de Software

Enginyeria Informàtica

Facultat de Matemàtiques, Universitat de Barcelona

PART 1. Introducció al disseny

4.1 Introducció al model de disseny

4.2 Diagrames d'interacció

4.2.1 Diagrames de seqüència

4.2.2 Diagrames de col·laboració

PART 2. Dissenyar amb patrons

4.3 Patrons generals d'assignació de responsabilitats en software

- Responsabilitats
- Patrons:
 - 1) Expert en informació
 - 2) Creador
 - 3) Alta cohesió
 - 4) Baix acoblament
 - 5) Controlador

PART 3. Finalitzant el disseny (pas previ a la codificació)

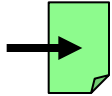
4.4 Diagrama/es de classes

4.5 Visibilitat

- Per atribut
- Per paràmetre
- Local
- Global

4.1 Introducció al model de disseny

Models anteriors



Requisits de l'usuari i del sistema

Cap a on anem?

Diagrames de casos d'ús



Especificació del UC

CASO DE USO:
Actor principal
Personal involucrado e intereses
Precondiciones
Postcondiciones
Escenario principal de éxito (Flujo básico)
Extensiones
Requisitos especiales
Lista de tecnología y variaciones de datos (requisitos no funcionales)
Frecuencia
Temas abiertos

Orientació a objectes

Models anteriors

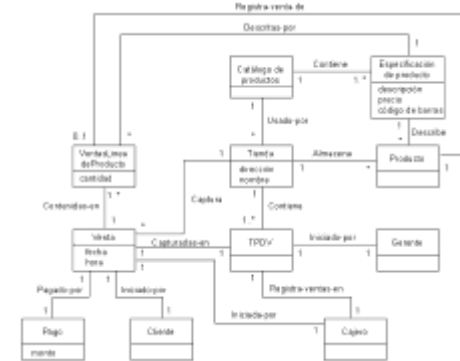
Requisits de l'usuari i del sistema

Models de sistema (Anàlisi)

Diagrames de casos d'ús



Diagrames de classes conceptuais



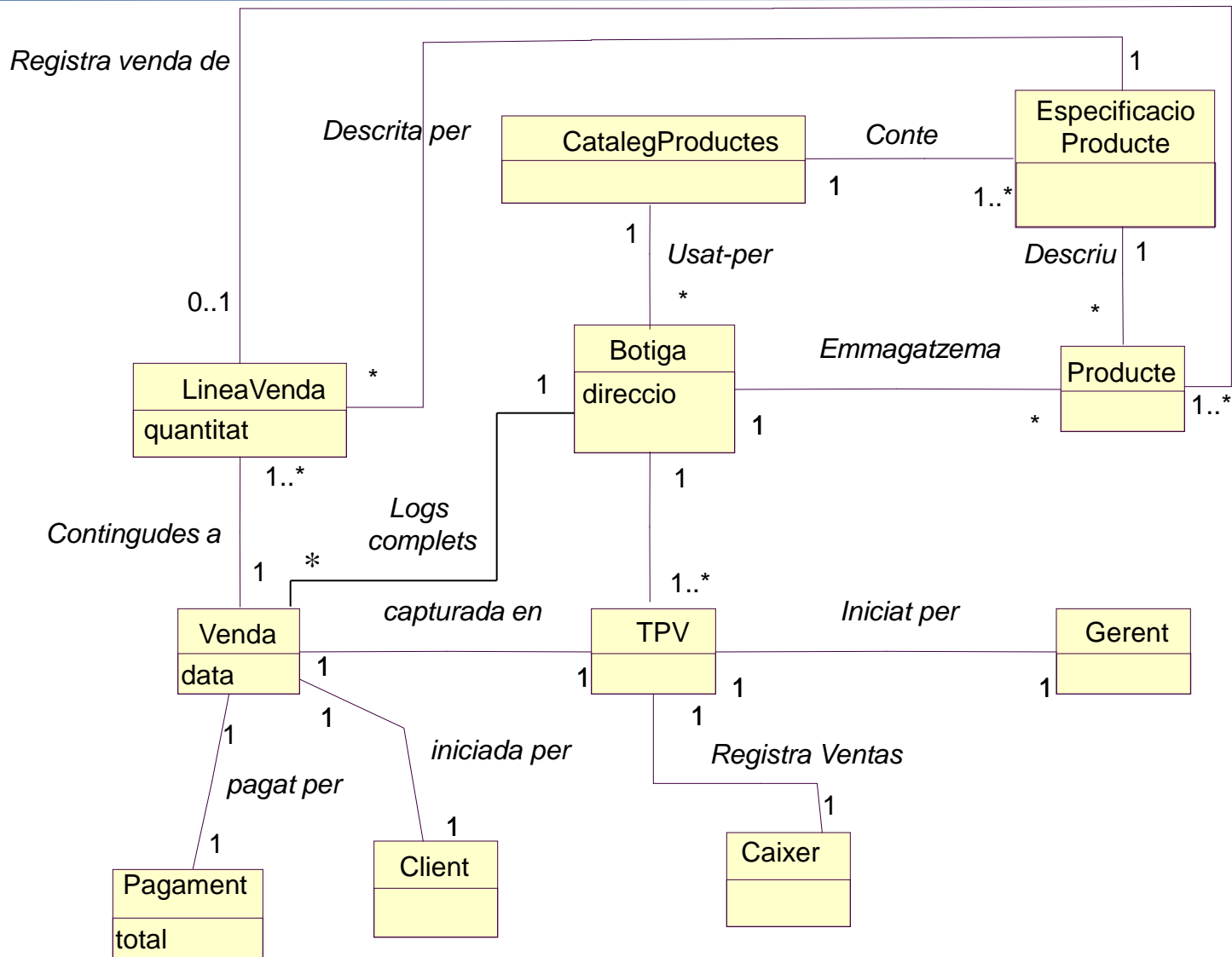
Orientació a objectes

Especificació del UC

CASO DE USO:
Actor principal
Personal involucrado e intereses
Precondiciones
Postcondiciones
Escenario principal de éxito (Flujo básico)
Extensiones
Requisitos especiales
Lista de tecnología y variaciones de datos (requisitos no funcionales)
Frecuencia
Temas abiertos

També anomenat
MODEL de DOMINI

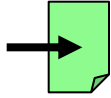
Exemple model domini TPV



Model de disseny

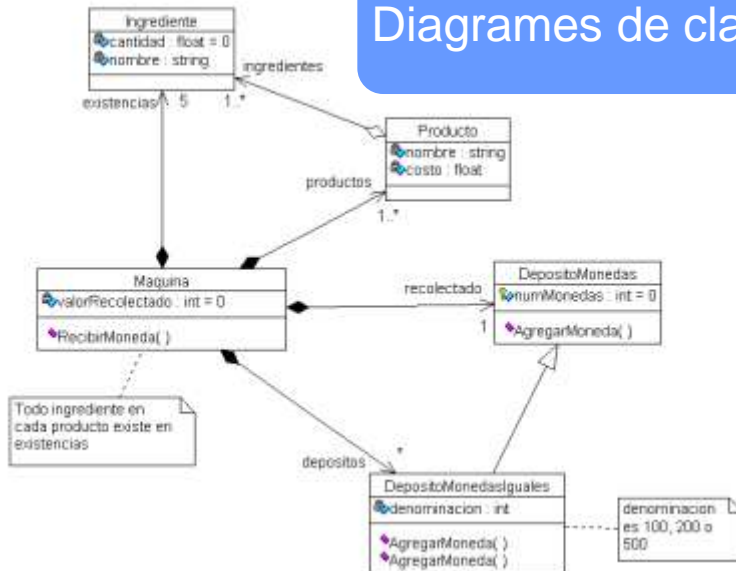
- La creació del **model de disseny** és la construcció d'una solució basada en el paradigma orientat a objecte.
 - El pas fonamental és la construcció de **diagrames d'interacció** que representen com els objectes col·laboren per satisfer els requisits.
 - Després d'aquests (o en paral·lel) es construeixen els **diagrames de classes** que resumeixen la definició de les classes de software que s'implementaran.
- Aprendrem a dibuixar aquests diagrames en UML, però ho fonamental és aprendre a usar-los com a eines de disseny. *Aprendre a dissenyar*

Model de disseny

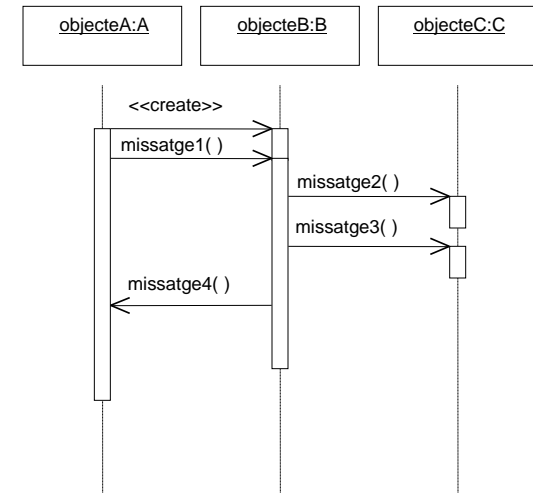
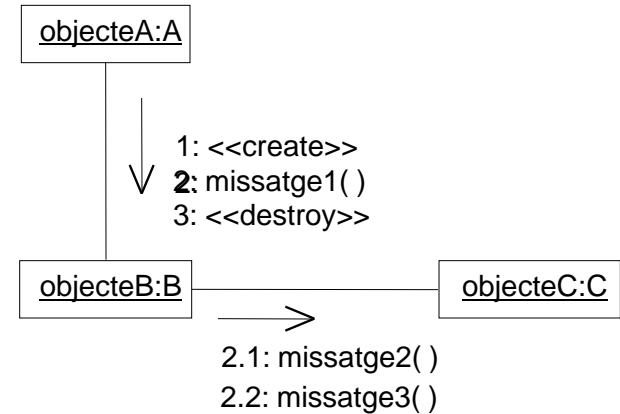


Model de disseny
orientat a objectes

Diagrames de classes



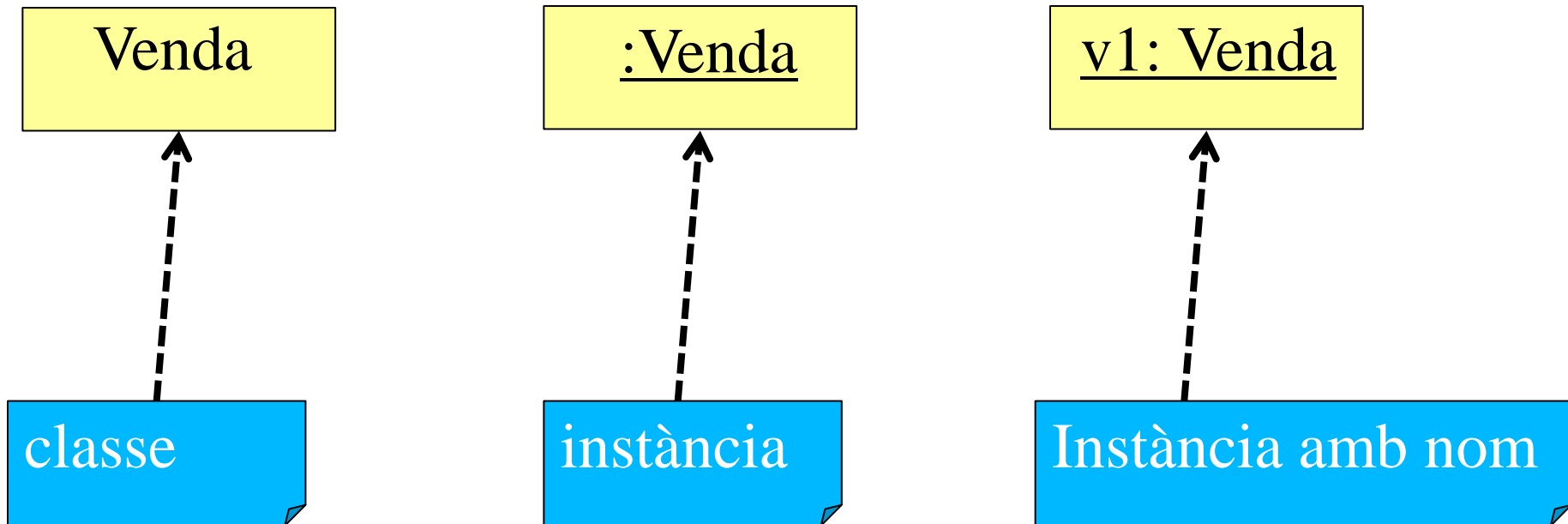
Diagrames d'interacció



Conceptes fonamentals

- Una **classe** és una especificació (com un plànol) que permet construir objecte d'un cert tipus. Defineix quins seran els atributs i mètodes d'aquests objectes
- Un **objecte** és un paquet d'informació
 - **atributs** i **mètodes** que gestionen aquesta informació.
 - És sinònim **d'instància**.
- Un objecte pot enviar **missatges** a un altre objecte (invocació d'algun dels seus mètodes)

Instàncies



Expressions de missatge

- UML segueix una sintaxi estàndard per expressar missatges.:

`retorn := missatge (parametre:TipusParam) :TipusRetorn`

- La informació de tipus es pot excloure en el cas de que es consideri obvia o no important:
 - `prod := obteProducte (id)`
 - `prod := obteProducte (id:ItemID)`
 - `prod := obteProducte (id: ItemID) :Producte`

4.2 Diagrames d'interacció

Definició i Tipus

Què són?

Un diagrama d'interacció consisteix en un **conjunt d'objectes** i les seves relacions, incloent-hi els **missatges** que es poden enviar entre ells

[modelar els aspectes dinàmics d'un sistema]

Tipus

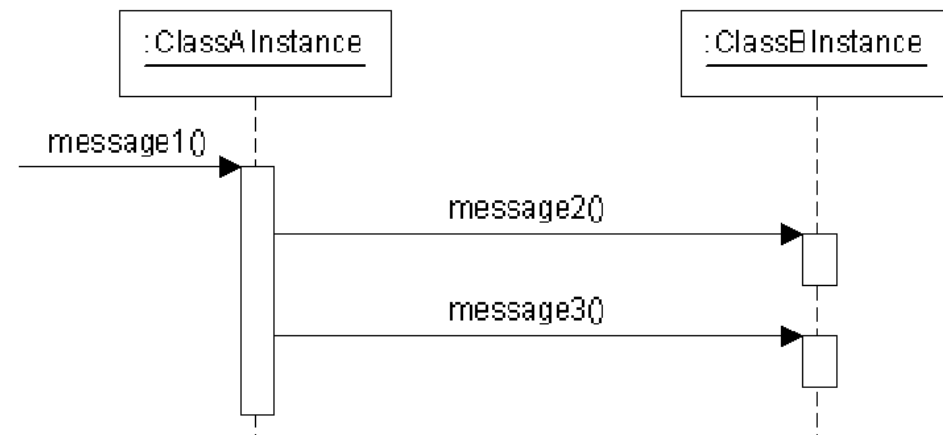
1. **Diagrames de seqüència**: destaquen l'ordre temporal dels missatges
2. **Diagrames de col·laboració**: destaquen l'organització estructural dels objectes

Ambdós diagrames (seqüència i col·laboració) són semànticament equivalents. Es pot passar d'un a l'altre sense pèrdua d'informació

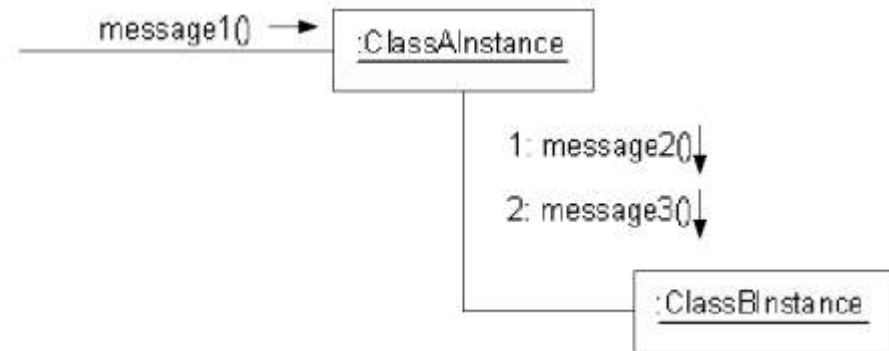
Consideracions

- Realitzant els diagrames d'interacció és quan es decideixen els **detalls d'un disseny** orientat a objectes. És un pas molt creatiu.
- És millor dissenyar els diagrames d'interacció per parelles, no sol. Dues persones veuen i aprenen més que una de sola.
- És possible aplicar principis coneguts, en forma de "**patrons**" per resoldre situacions comuns amb un disseny de millor qualitat.

Tipus de diagrames d'interacció



(a) Diagrama de seqüència



(b) Diagrama de col·laboració

Col·laboració vs. Sequència

Diagrama de col·laboració vers Diagrama de seqüència

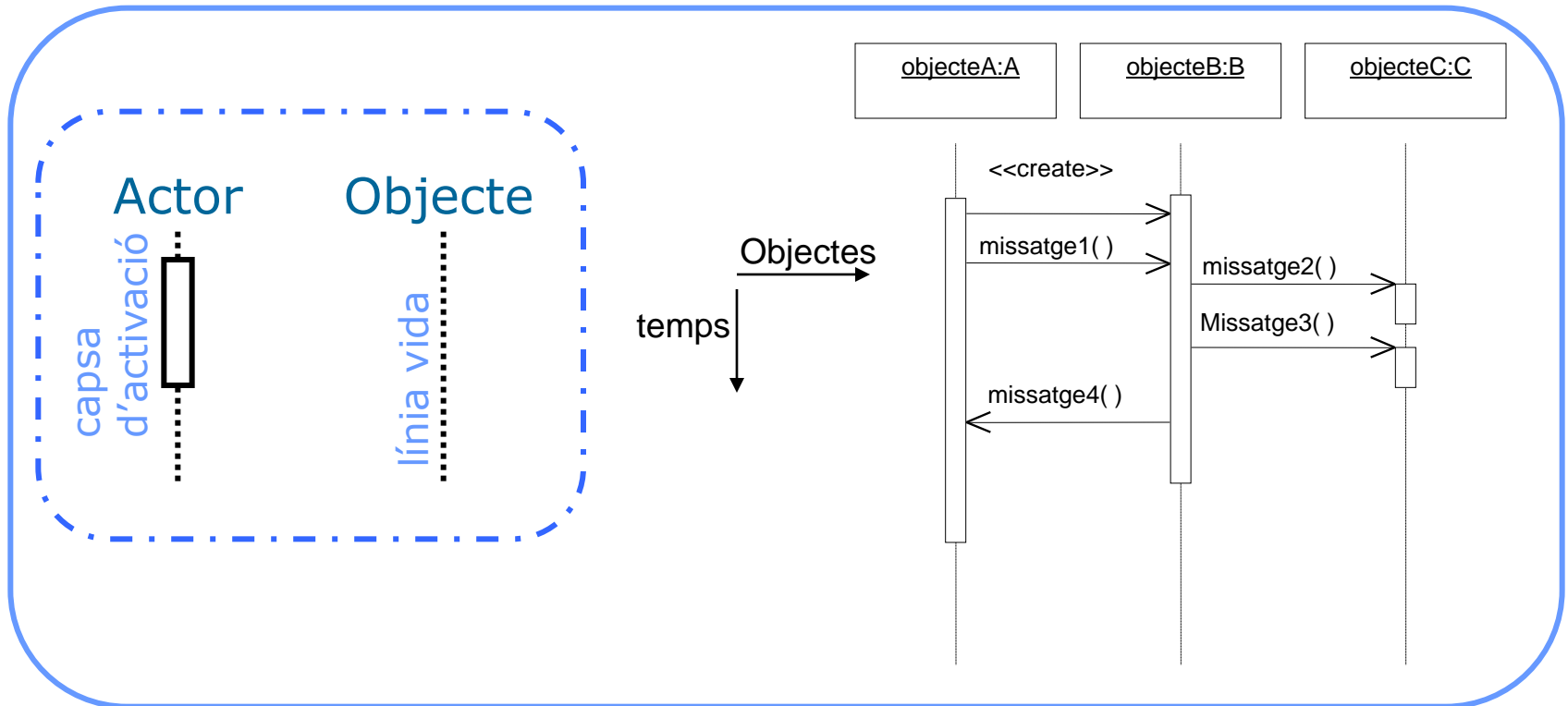
Tipus	Punts Forts	Punts Febles
Col·laboració	<ul style="list-style-type: none"> - Economia d'espai - És més adequat per il·lustrar ramificacions complexes de missatges 	<ul style="list-style-type: none"> - És més complex (no és fàcil veure la seqüència de missatges) - Notació complexa
Seqüència	<ul style="list-style-type: none"> - Mostra clarament l'ordenació temporal de missatges - Notació simple 	<ul style="list-style-type: none"> - Consumeix espai horitzontal (força a estendre's per la dreta)

4.2.1 Diagrames de seqüència

Definició

Què són?

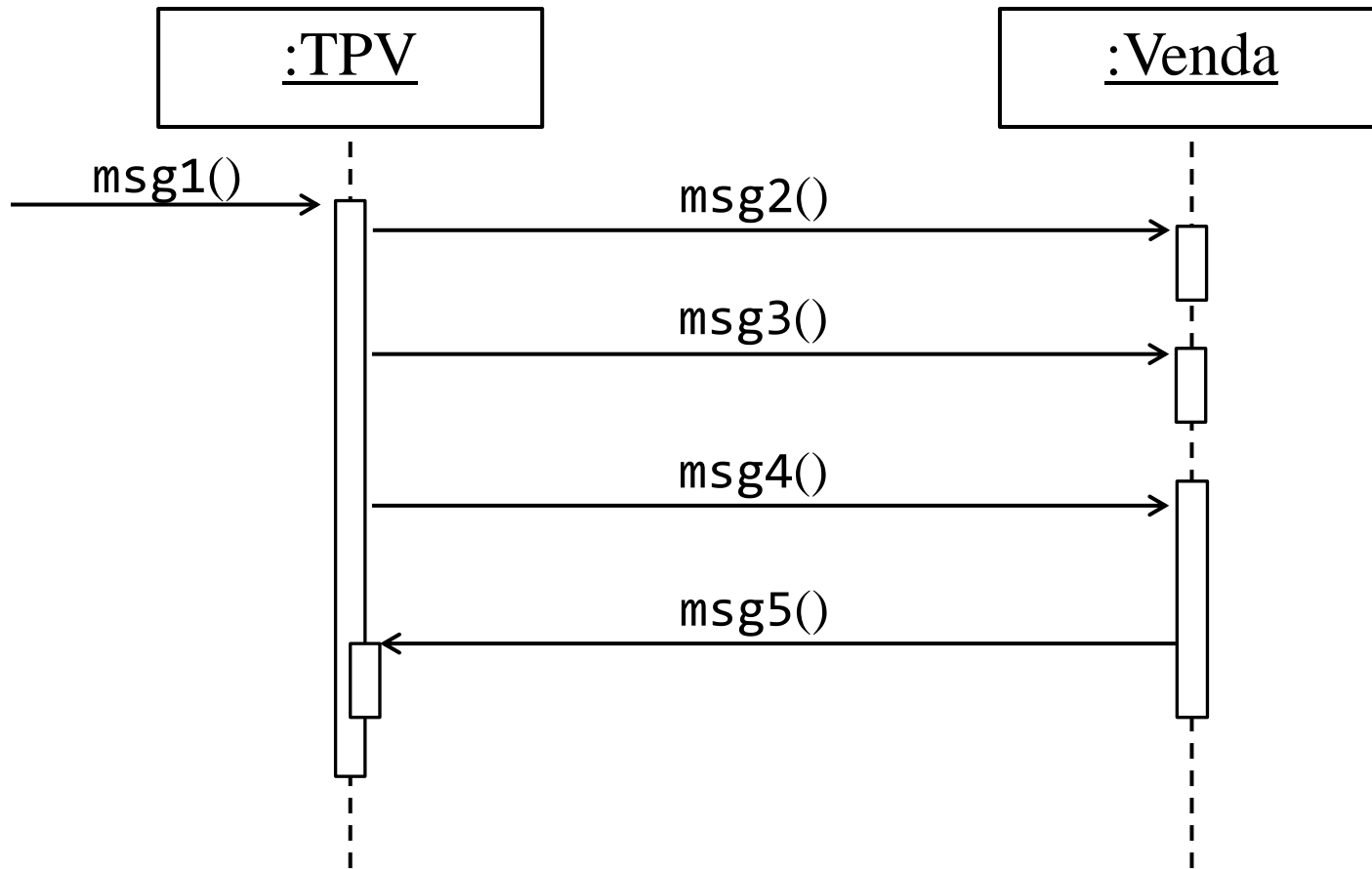
Els diagrames de seqüència destaquen l'ordenació temporal dels missatges.



Consideracions

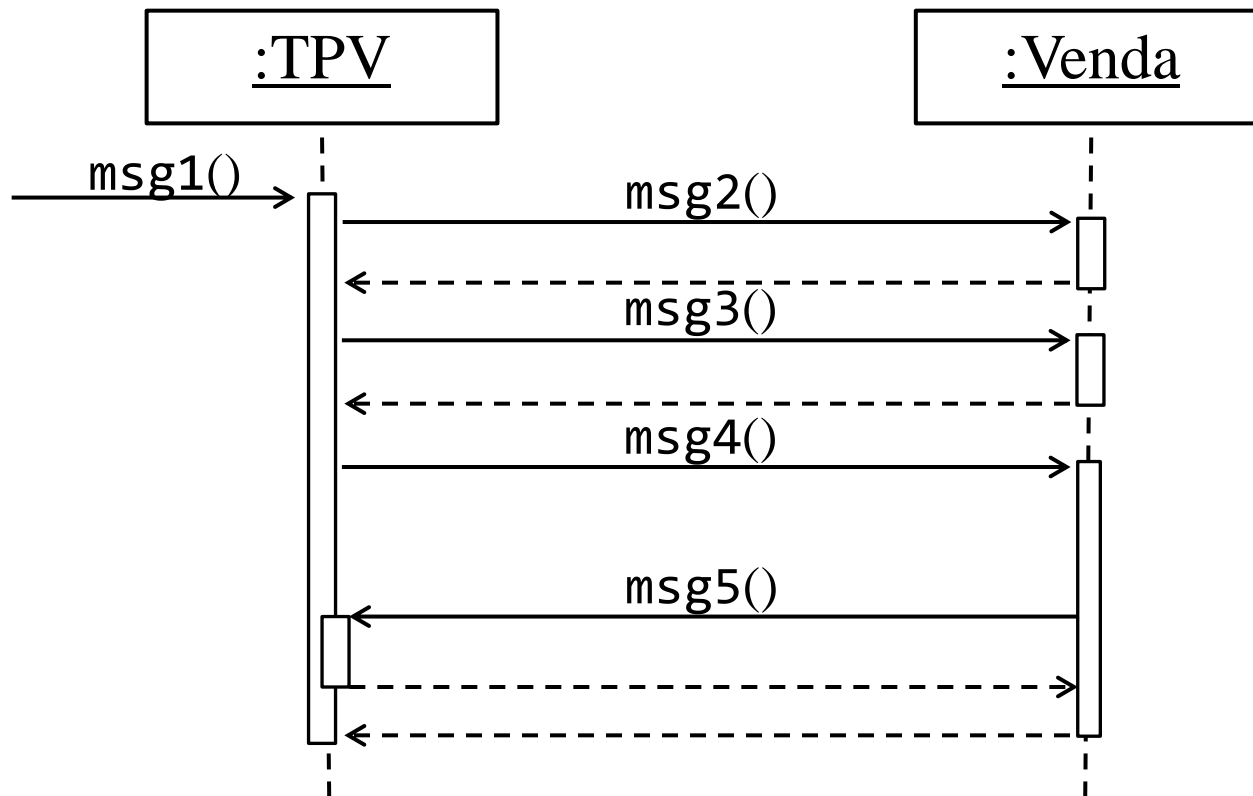
- Cada missatge entre objectes es representa com una expressió de missatge sobre una fletxa entre els objectes. L'ordenació temporal comença a dalt i acaba a baix.
- Els diagrames de seqüència permeten il·lustrar també el flux de control usant **capses d'activació**. L'ús de les capses d'activació és opcional però freqüent.

Missatges i capses d'activació



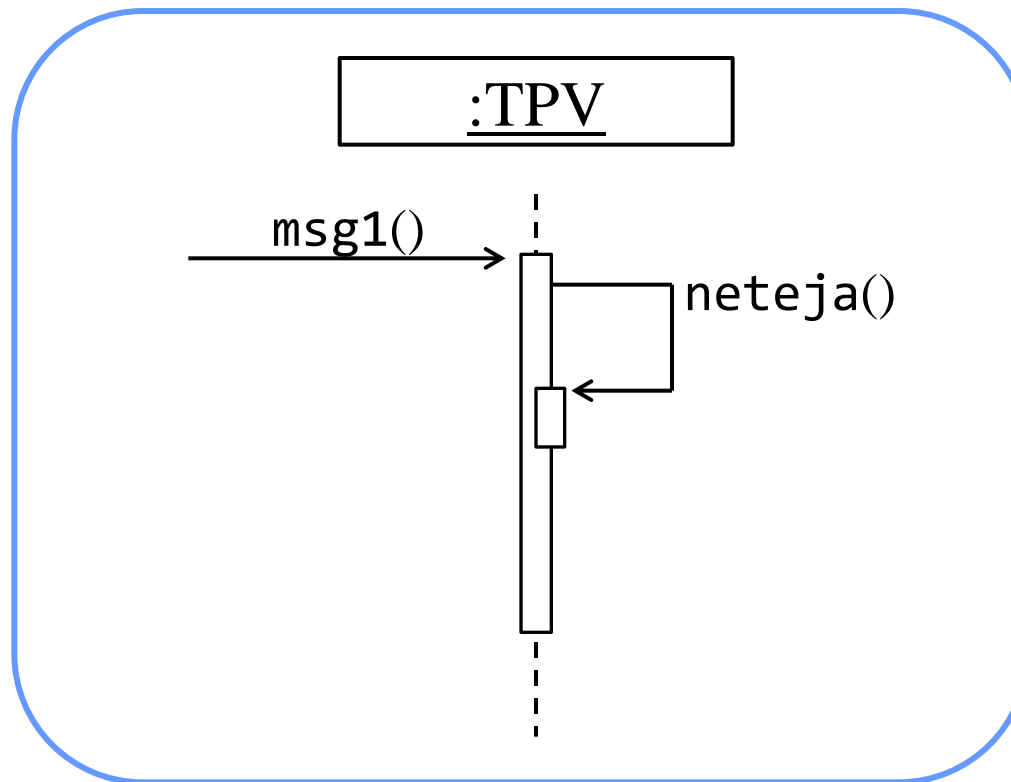
Retorns

Un diagrama de seqüència pot mostrar explícitament el retorn d'un missatge mitjançant una línia discontinua que es pot anotar amb el contingut del que es retorna

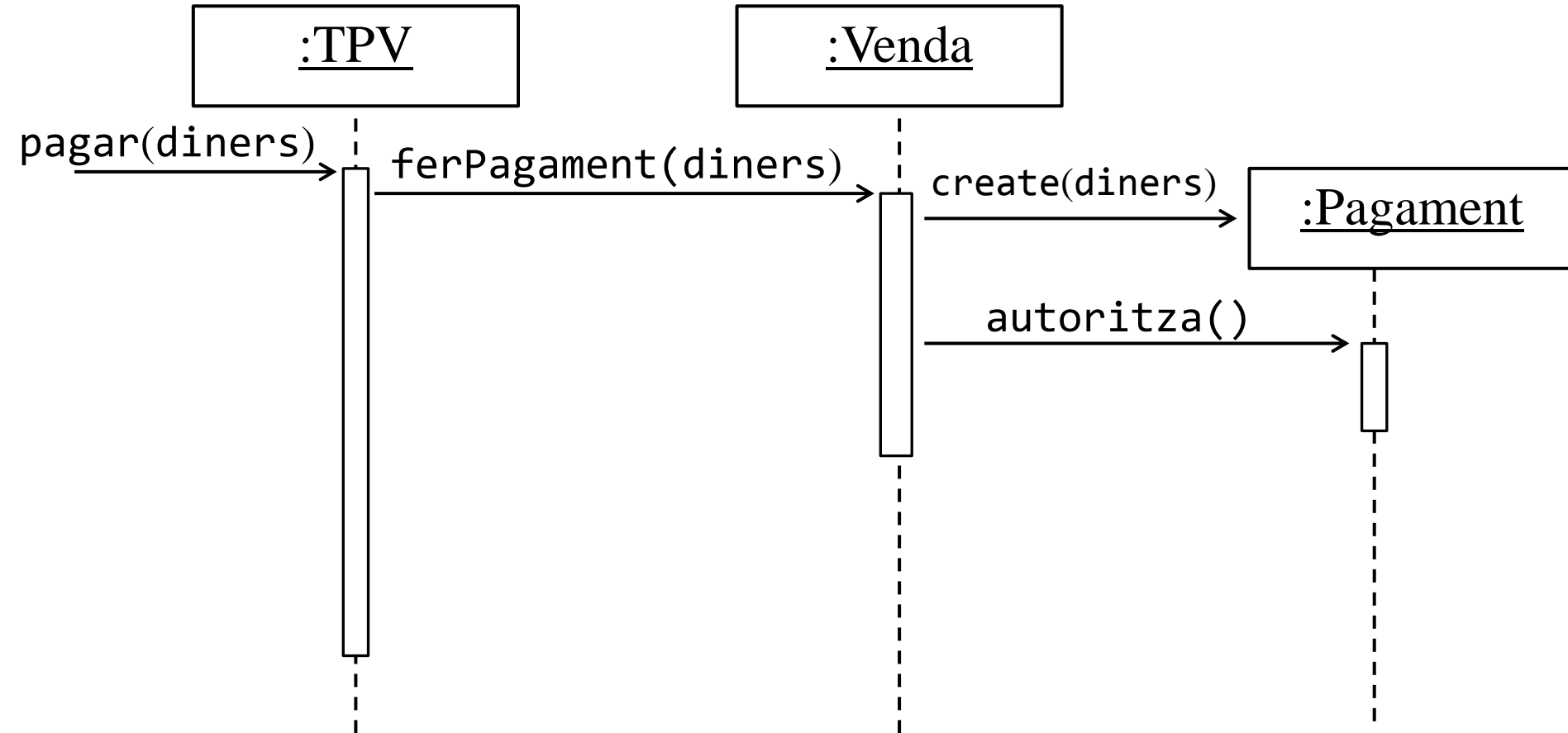


Missatges a sí mateix

- Un objecte pot enviar missatges a si mateix usant una capsula d'activació aniuada



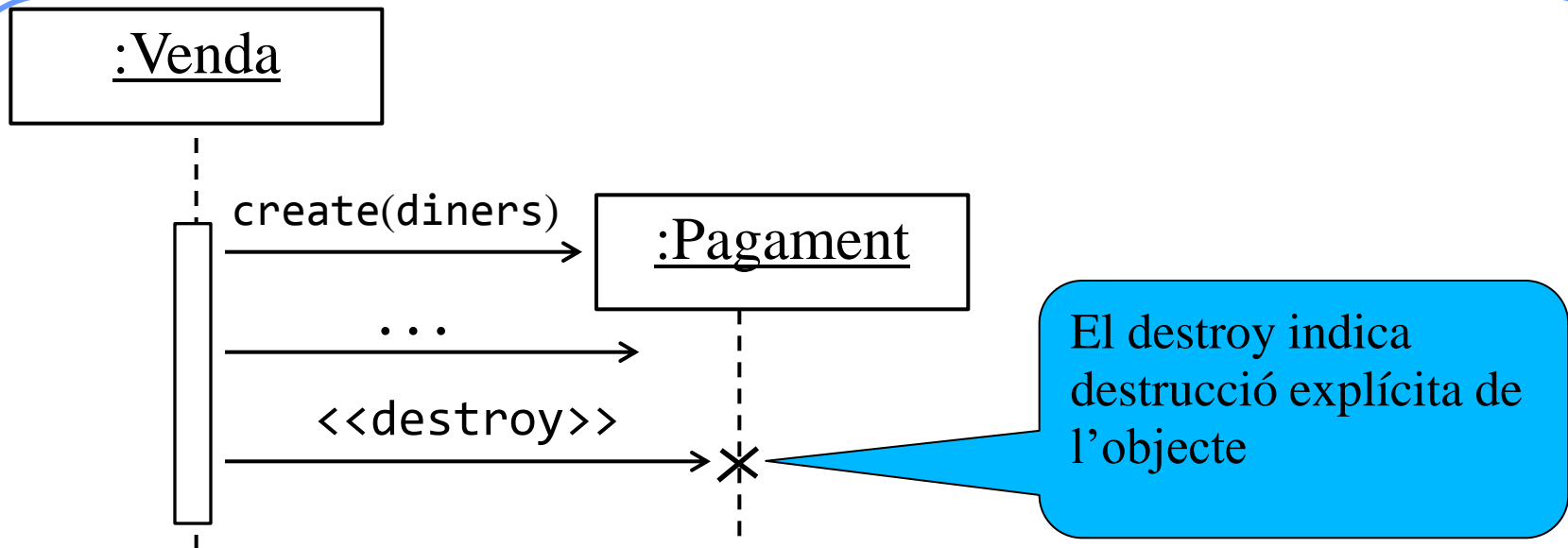
Creació d'instàncies



La línia de vida mostra la durada de la vida d'un objecte en el diagrama

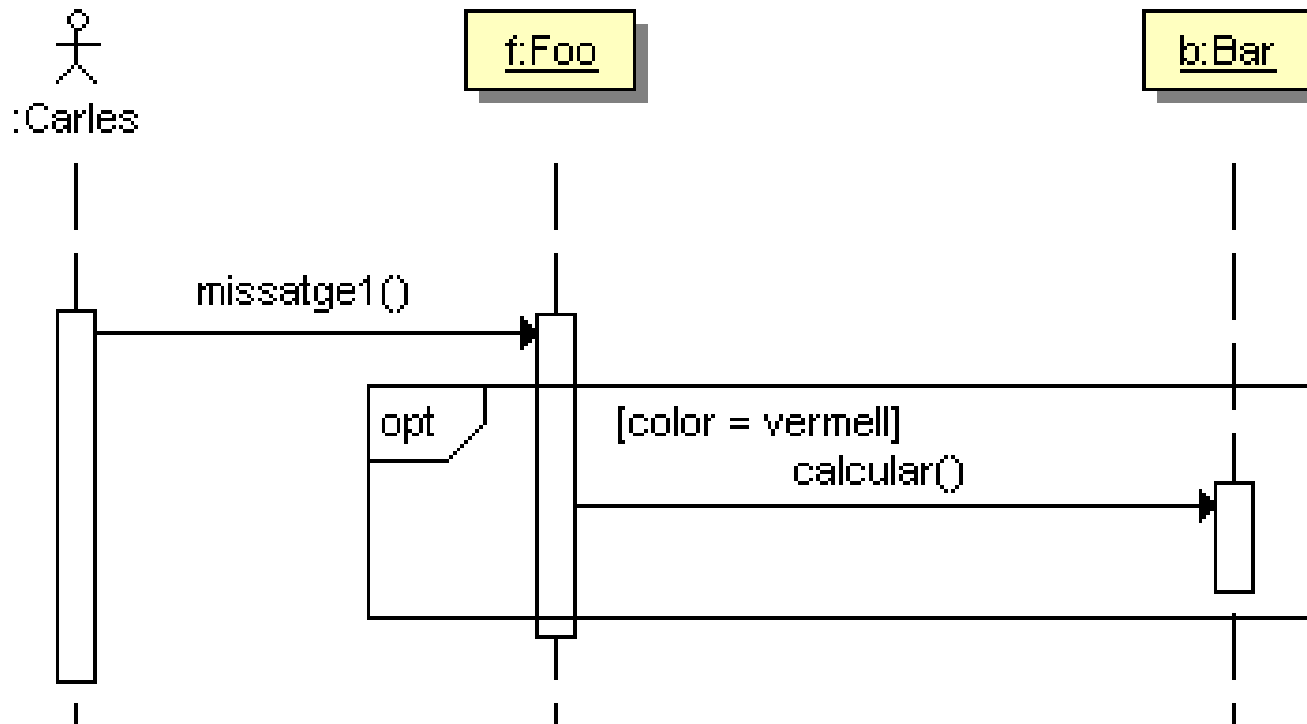
Línies de vida i destrucció d'objectes

- Les línies verticals puntejades representen en quins moments l'objecte està viu en el diagrama
- Es pot representar la destrucció d'un objecte mitjançant l'enviament d'un mètode *destroy*



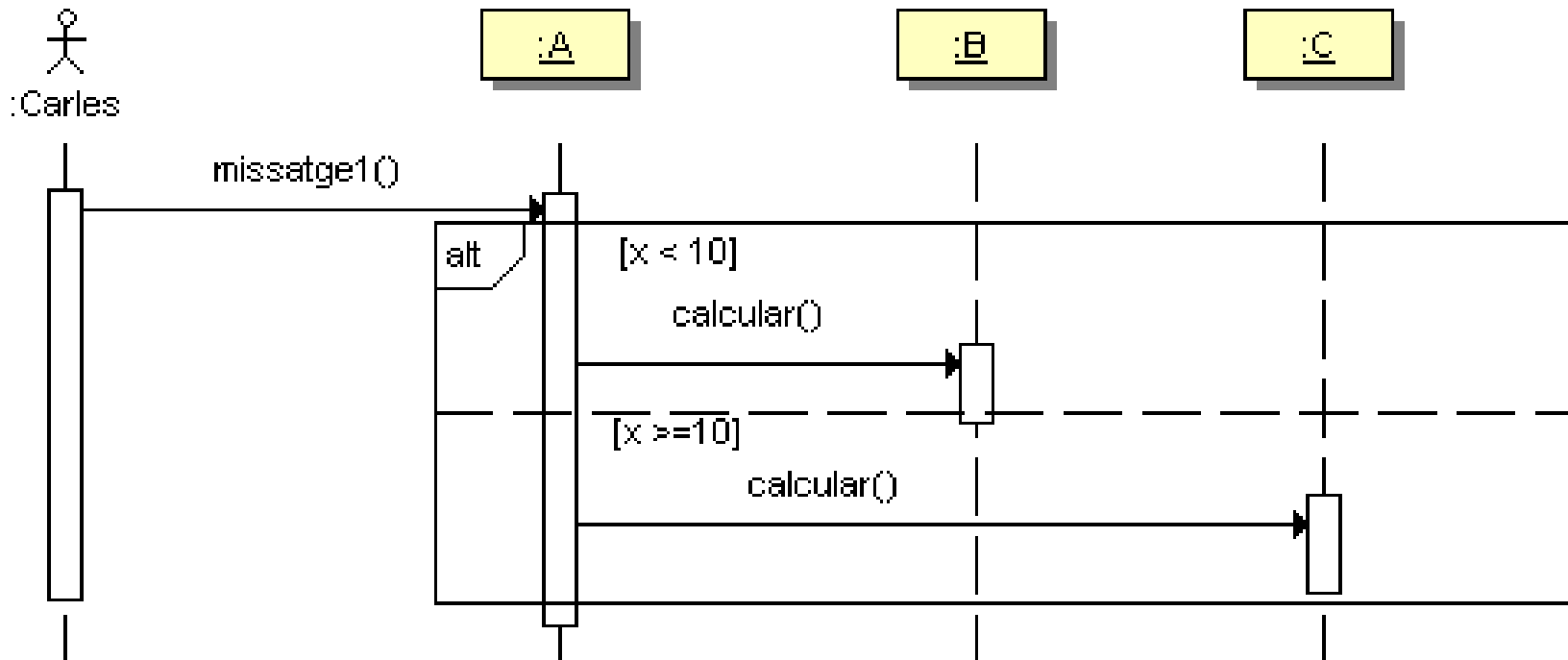
Condicionals

Missatges condicionals



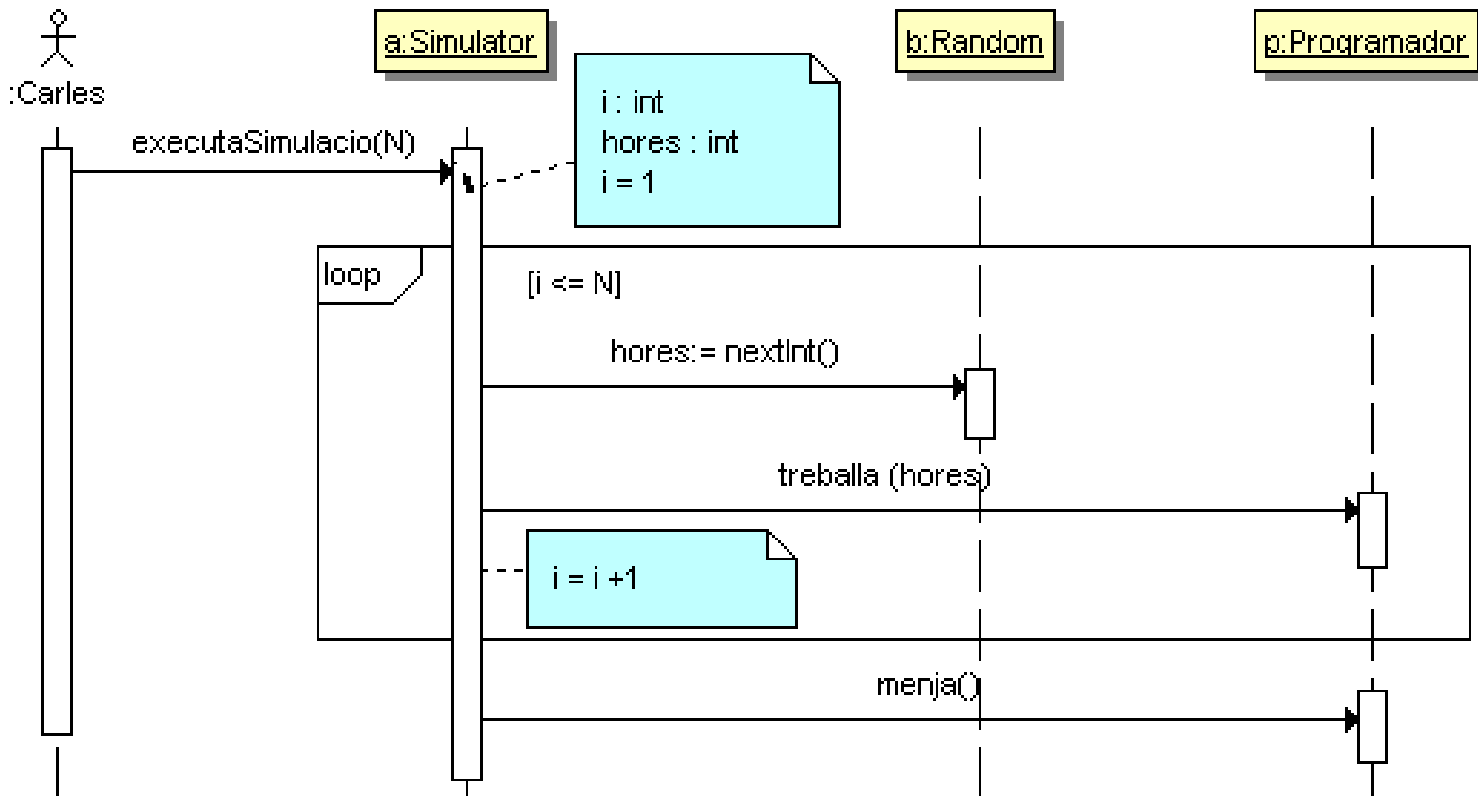
Condicional mútuament excloent

Missatges condicionals mútuament excloent



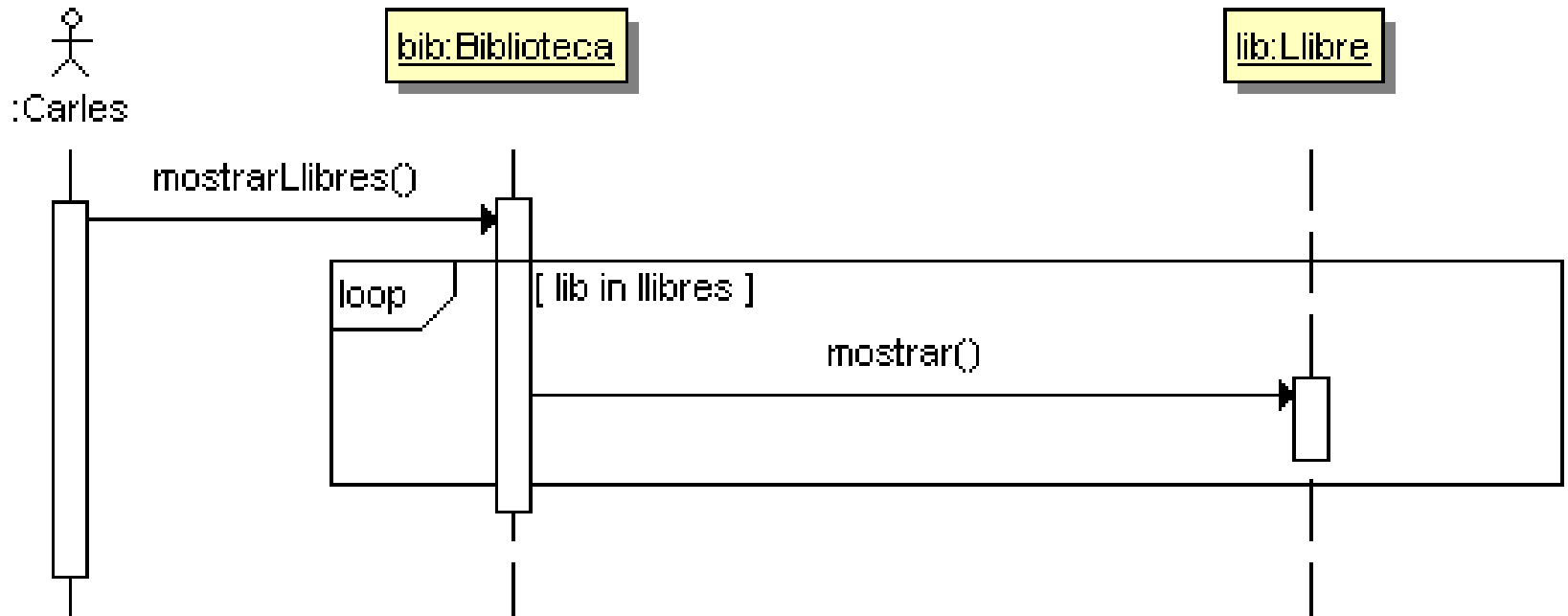
Iterar

Iteració

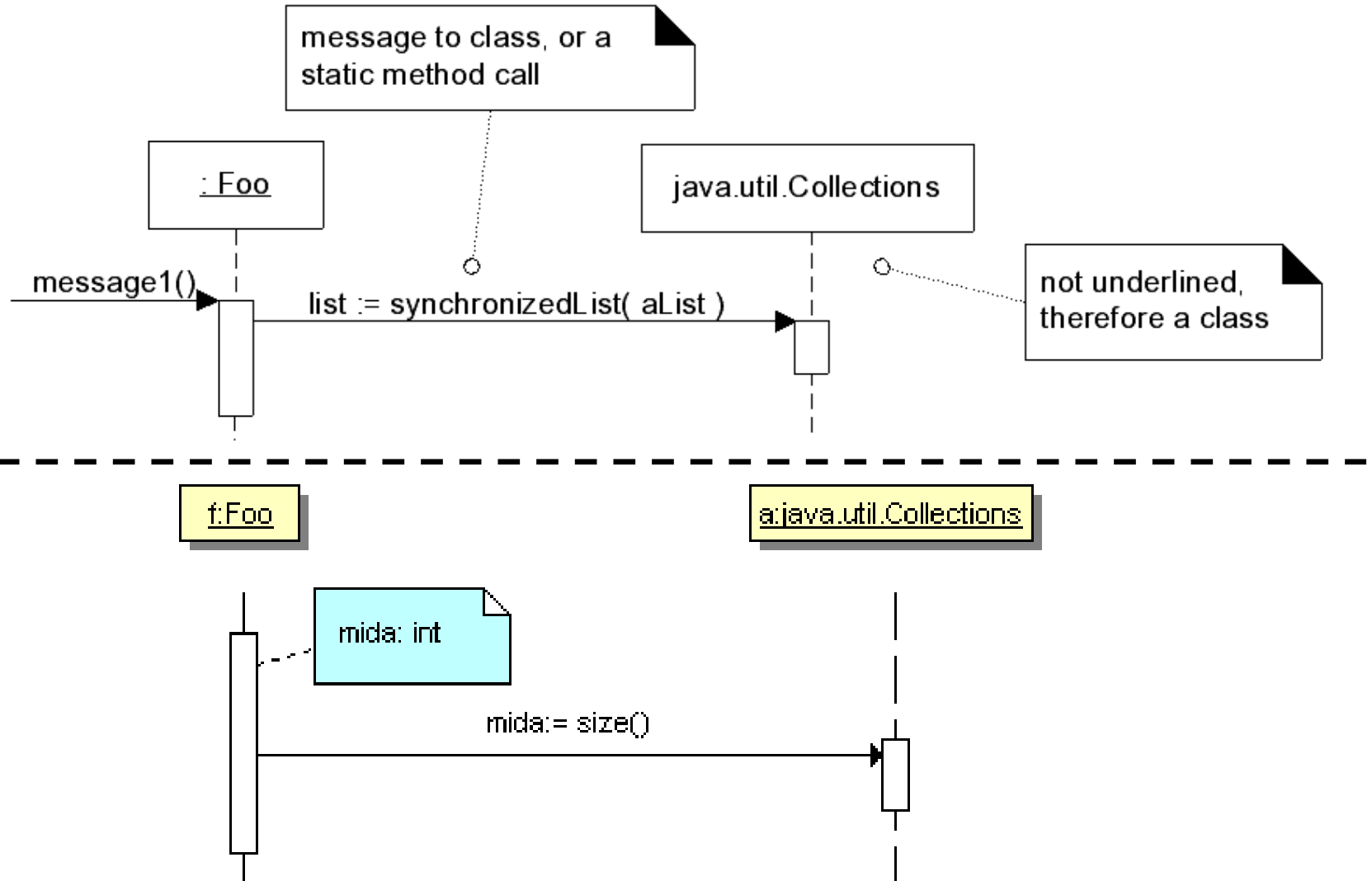


Iterar sobre una col·lecció

Iteració sobre una col·lecció



Missatges a una classe o a un objecte

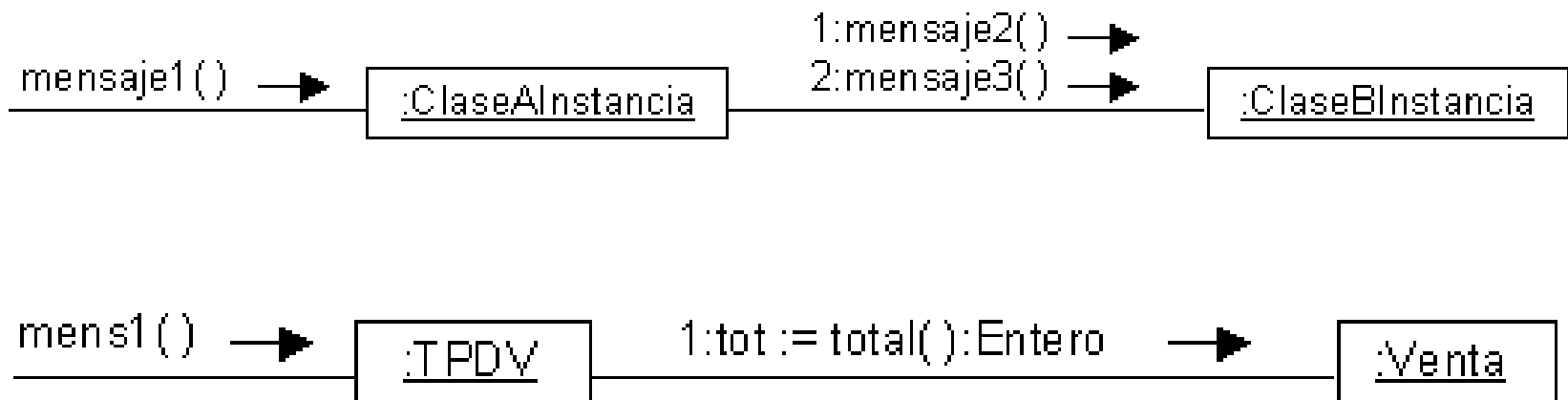


4.2.2 Diagrames de col·laboració

Definició

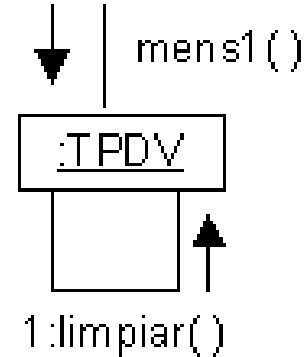
Què són?

Els diagrames de col·laboració expliquen gràficament les interaccions entre les instàncies del model (objectes).



Missatges a sí mateix i iteracions

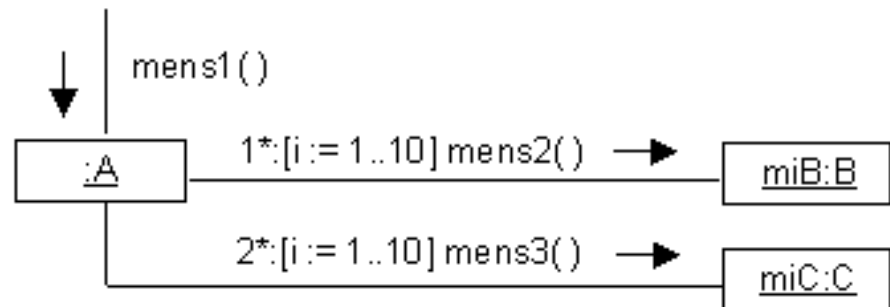
Un objecte es pot enviar un missatge a si mateix:



Es possible representar iteracions:

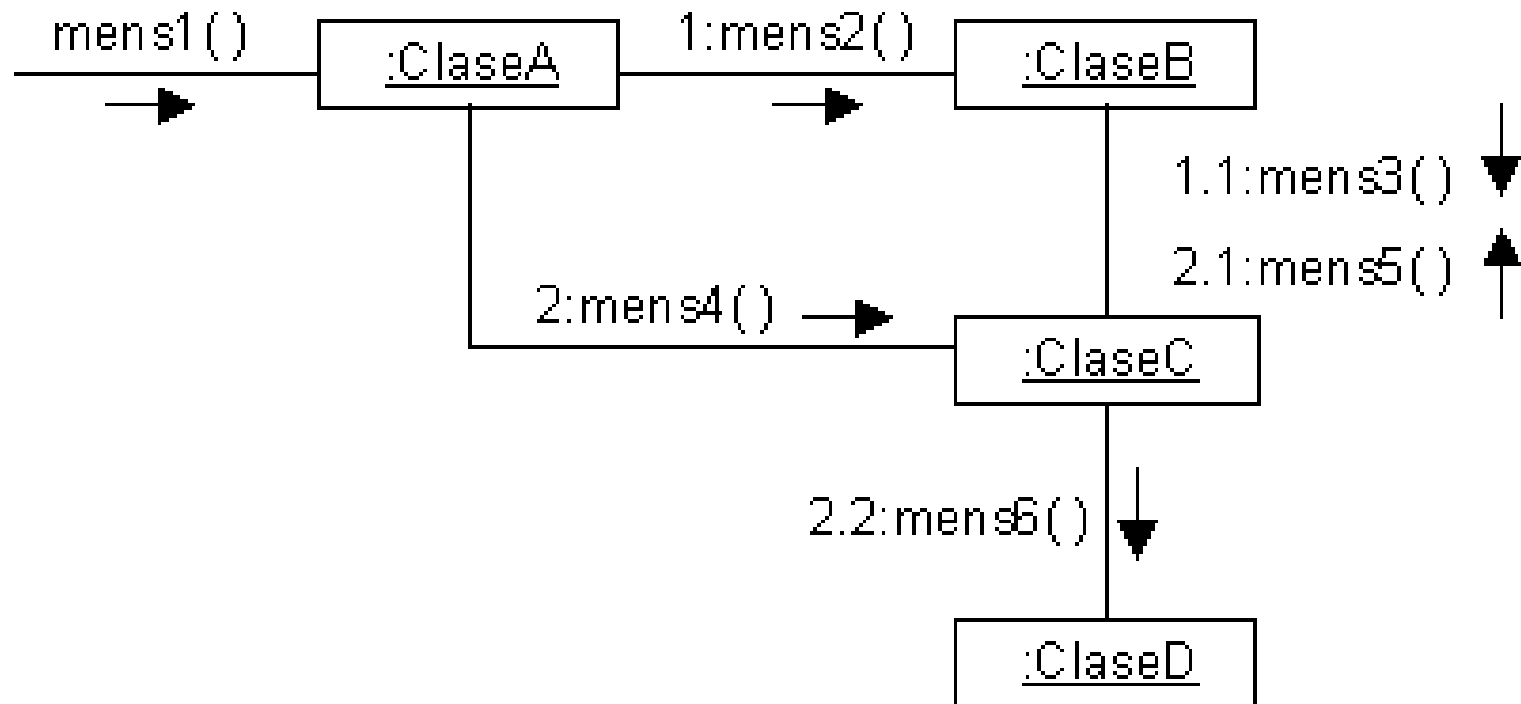
```

msg1() {
  for i := 1 to 10 {
    miB.mens2();
    miC.mens3();
  }
}
  
```



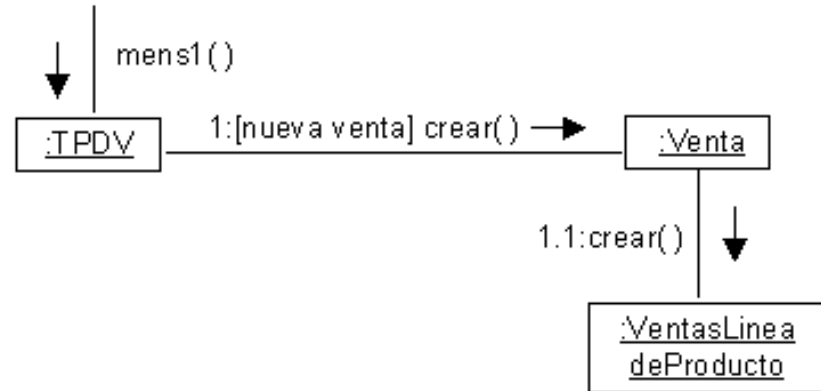
Exemple

Seqüència dels missatges en un diagrama de col·laboració:

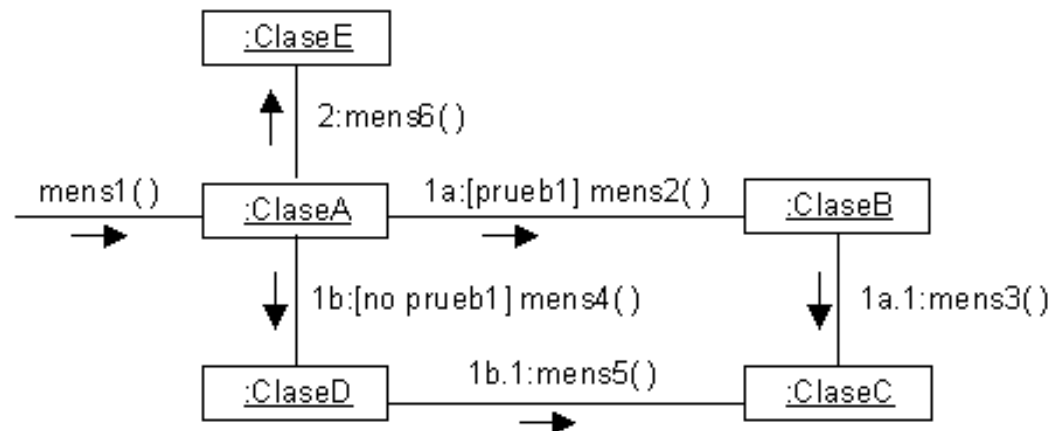


Conditionals

Es possible definir missatges condicionals:



Es possible definir trajectòries mútuament excloents:

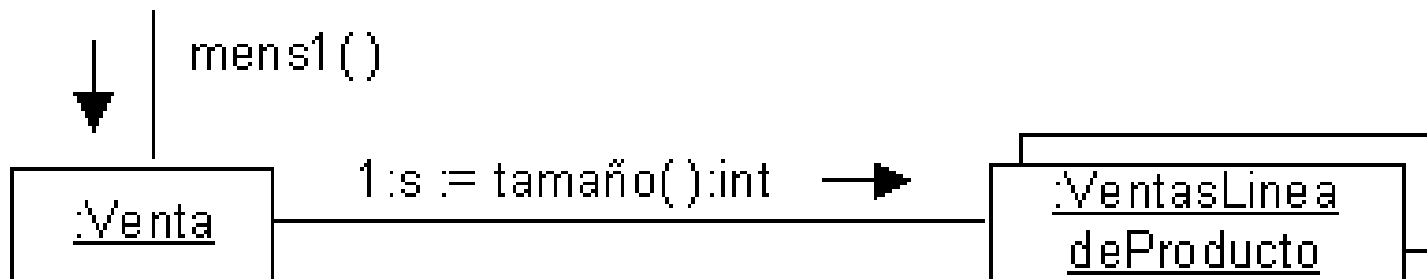


Multiobjectes

Un multiobjecte, per exemple un array, es representa com una pila d'objectes:



Es poden enviar missatges a multiobjectes:



Creació i agregació a un multiobjecte

Exemple de crear un objecte i agregar-lo a un multiobjecte:

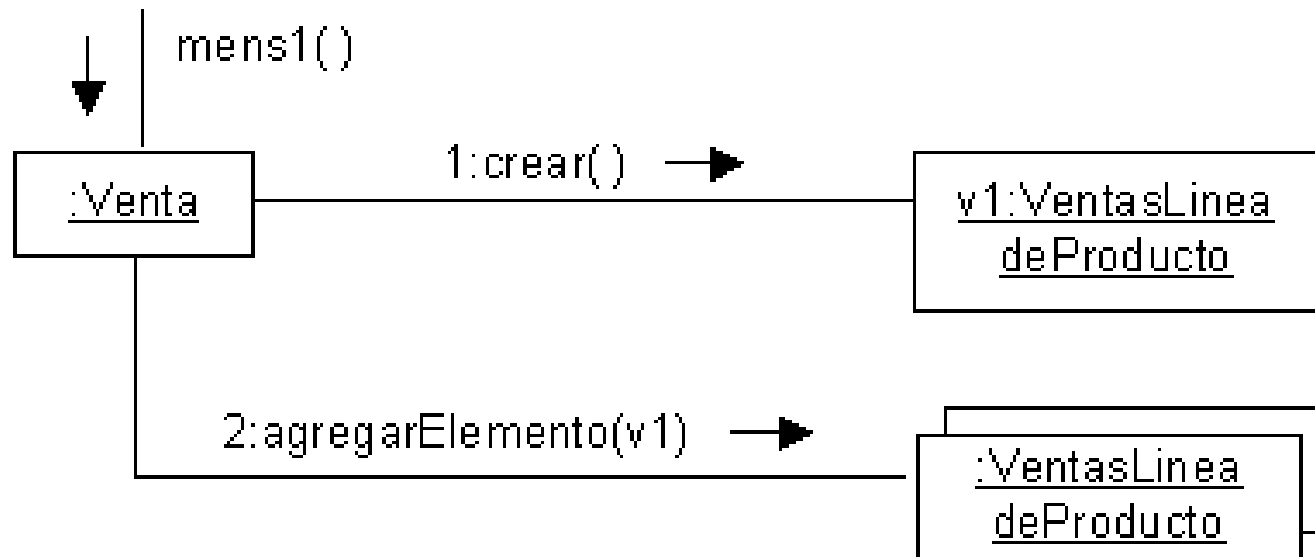


Diagrama de col·laboració TPV

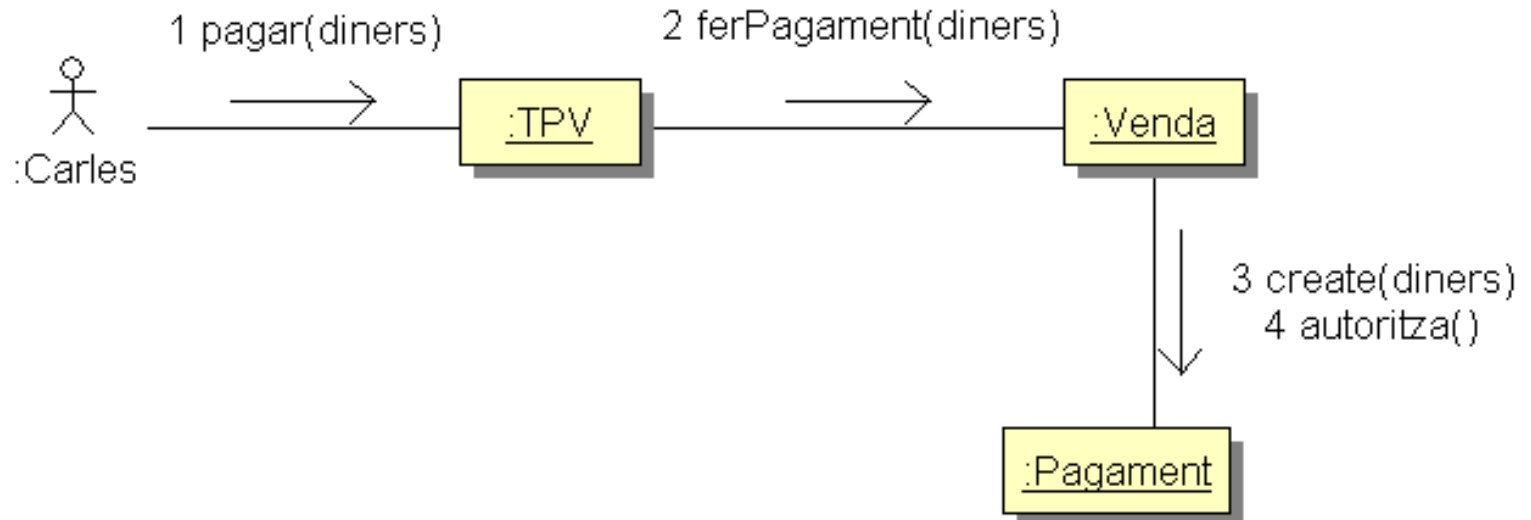


Diagrama de seqüència TPV

