

# UML Dinàmic

## Diagrames d'Interacció

# UF3

INTRODUCCIÓ AL DISSENY  
ORIENTAT A OBJECTES

## M05. Entorns de desenvolupament

Hèctor López  
hector.lopez@fje.edu



JESUÏTES El Clot  
Escola del Clot



# Diagrama d'interacció

## Introducció

L'execució d'un programari orientat a objectes és una **seqüència d'operacions i canvis d'estat** d'objectes.

Els diagrames d'interacció s'utilitzen per **descriure els casos d'ús en termes d'objectes i missatges entre objectes**.

Durant les transicions o les execucions de les operacions es criden operacions d'altres objectes i s'envien senyals que provoquen altres transicions.

Aquests casos són precisament els que es poden representar mitjançant **diagrames d'interacció**.



UNIFIED MODELING LANGUAGE™



# Diagrama d'interacció

## Introducció

### Interaccions

Una **interacció** és l'especificació del comportament d'un cas d'ús o d'una operació en termes de seqüències de missatges entre objectes. Aquests missatges contenen estímuls que poden ser peticions d'execució d'operacions o senyals. S'anomena fil d'execució a una seqüència de missatges.

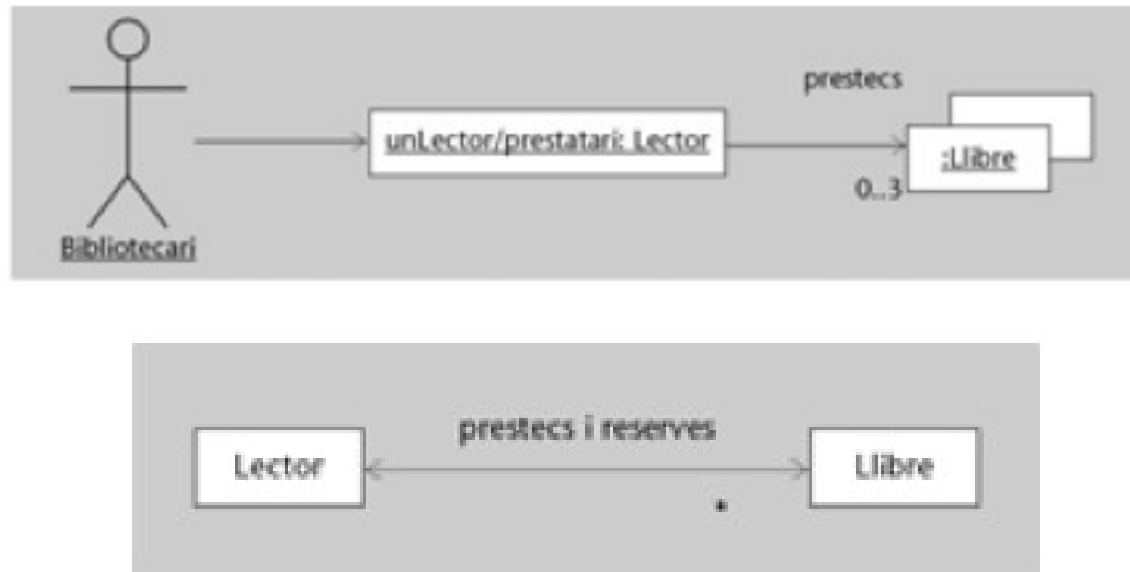
### Col·laboracions

Una **col·laboració** és un conjunt de papers, de classificadors, o d'instàncies; i de papers d'associacions entre els objectes que intervenen en una interacció. Les col·laboracions s'utilitzen per coherència entre les classes i associacions definides en el model estàtic i les associacions i els objectes definits en els casos d'ús. Les col·laboracions es poden definir a partir de classificadors o d'instàncies.

# Diagrama d'interacció

## Introducció

A vegades us interessarà representar **multiojectes** que representen un conjunt d'objectes d'un paper amb multiplicitat més gran que 1 dins d'una associació.



# Diagrama d'interacció

## Introducció

UML ofereix 3 diagrames per representar interaccions i col·laboracions:

- El diagrama de **comunicació**, posa l'èmfasi en la descripció de la **col·laboració**.
- El **diagrama** de **seqüència**, que posa l'èmfasi en la successió temporal de missatges en una interacció.
- El **diagrama** d'**activitats**, que serveix per a descriure els estats d'una activitat.

En aquest bloc de diapositives els analitzarem tots tres, parant especial atenció als diagrames de seqüència...

# Diagrama de col·laboració

## Descripció

El diagrama de col·laboració és la **representació** d'una **interacció**.

**Parteix** del **diagrama estàtic**, sobre el qual es representen els **missatges** de la interacció.

Cada missatge té la següent especificació:

***PREDECESSORS GUARDA EXPRESSIONS\_SEQÜÈNCIA VALORS\_RETORN SIGNATURA***

on:

**PREDECESSORS** = LLISTA MISSATGES PREDECESSORS '(' NUM\_SEQ, NUM\_SEQ ')'

**GUARDA** = CONDICIÓN QUE S'HA DE COMPLIR PERQUÈ S'ENVII EL MISSATGE

**EXPRESSIONS\_SEQÜÈNCIA** = NUM\_SEQ '[' RECURRÈNCIA ']', ..., ':'

**RECURRÈNCIA** = '\*' '[' CLÀUSULA\_ITERACIÓ ']'

**VALORS\_RETORN** = VALORS DE RETORN COM A CONSEQÜÈNCIA DEL PROCÉS ENGEAT  
PEL MISSATGE VALOR\_RETORN ', ' VALOR\_RETORN ... ':='

**SIGNATURA** = NOM ESTÍMUL I LLISTA D'ARGUMENTS

# Diagrama de col·laboració

## Tipus de missatges (I)

Es pot indicar també la creació d'objectes (*new*), destrucció (*destroyed*) i enllaços (*transient*).

**Missatges Simples:** corresponen a una simple execució dins d'un fil d'execució:



**Missatges Síncrons:** aquests missatges es donen quan el client envia un missatge al subministrador i aquest accepta el missatge. La classe emissora s'espera a rebre el resultat de l'operació associada al missatge.



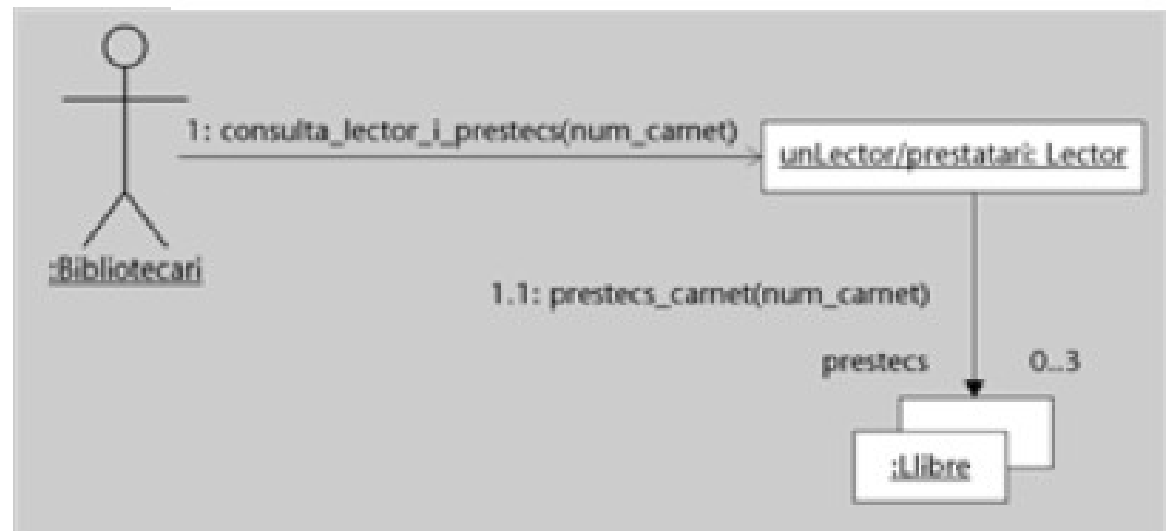
# Diagrama de col·laboració

## Tipus de missatges (II)

**Missatges Asíncrons:** en aquest cas, la classe emissora envia un missatge al subministrador i es continua executant sense esperar que arribi el resultat de la classe receptora. En aquests casos habitualment la classe receptora posarà en una cua l'execució de la petició rebuda.

### Missatges de Resposta:

← — — — — —





# Diagrama de seqüència

## Descripció (I)

A diferència dels diagrames de col·laboració, els de seqüència no representen explícitament el paper de les associacions.

En canvi, sí que **representen el pas del temps** i la **durada dels missatges**

El temps es representa cap avall, i no està necessàriament fet a escala.

En horitzontal, i en franges successives, es representen els diferents papers dels classificadors que participen en la interacció.

Els missatges circulen d'esquerra a dreta i les respostes a l'inrevés.

S'anomena **línia de vida** a un element que simbolitza l'existència del paper en un cert període de temps. Es representa per una línia vertical discontinua que va des de la creació fins a la destrucció de l'objecte.

Per representar la **destrucció** d'un objecte es posa una **X** al final de la línia de vida.

# Diagrama de seqüència

## Descripció (II)

Durant la vida d'un objecte poden haver-hi **diversos constructors** segons les condicions que emeti el missatge.

Una **activació** és una part de la línia de vida durant la qual un determinat paper executa una acció, o bé altres papers executen altres accions com a conseqüència d'una acció executada pel primer.

Les activacions coincideixen amb l'**enviament** d'un missatge i la **recepció** d'una resposta.

Si hi ha **crides reflexives**, aquestes es representen amb un altre requadre lleugerament desplaçat.

Els missatges poden ser del mateix tipus que els diagrames de col·laboració, i les fletxes es representen igual.

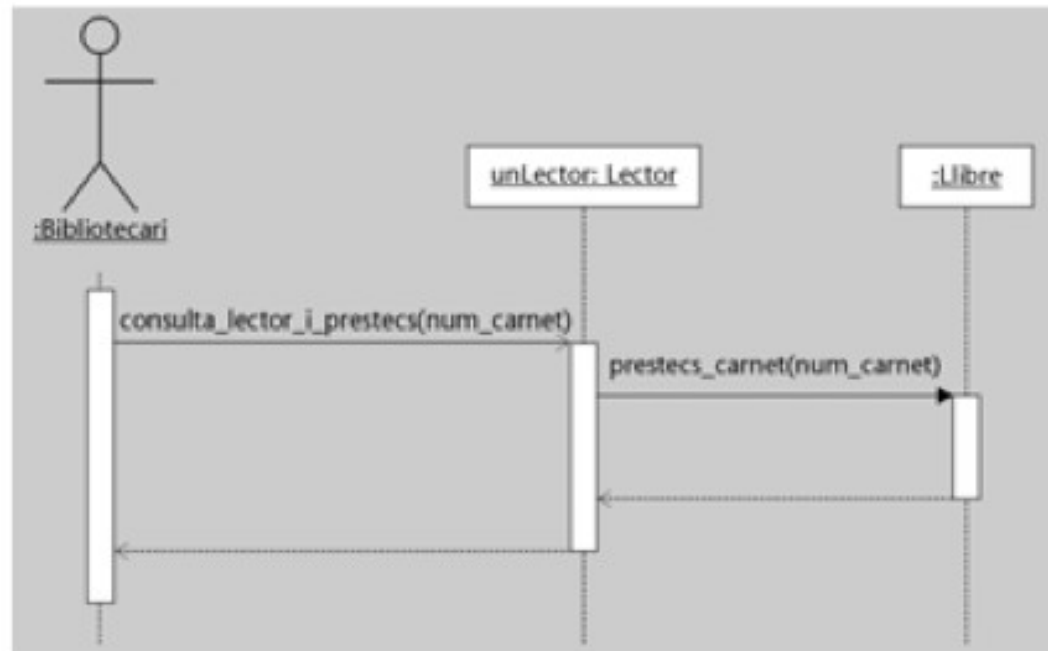
**L'ordre d'execució es representa de dalt a baix en el temps.**

# Diagrama de seqüència

## Descripció (III)

- No s'indiquen els números de seqüència perquè estan implícits.
- Poden sortir dues fletxes d'un mateix punt d'activació si s'envien dos missatges alhora.
- Els missatges de retorn es representen amb una línia discontinua.

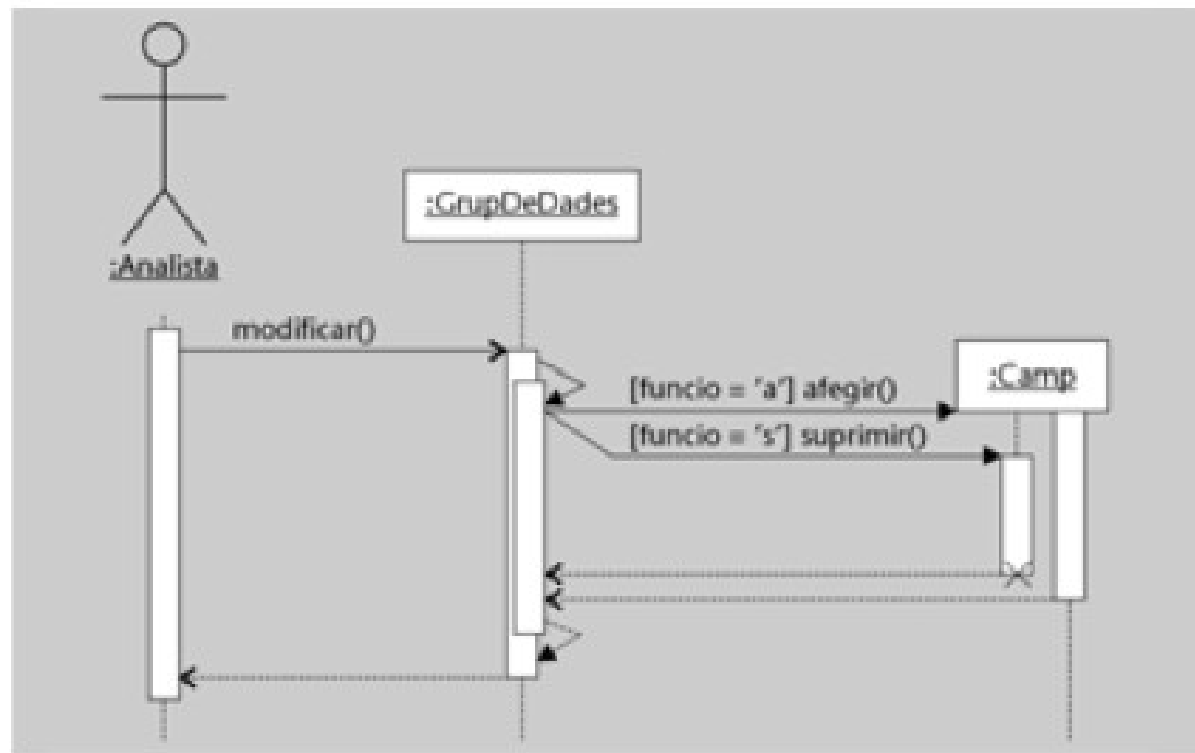
### Exemple:



# Diagrama de seqüència

## Descripció (IV)

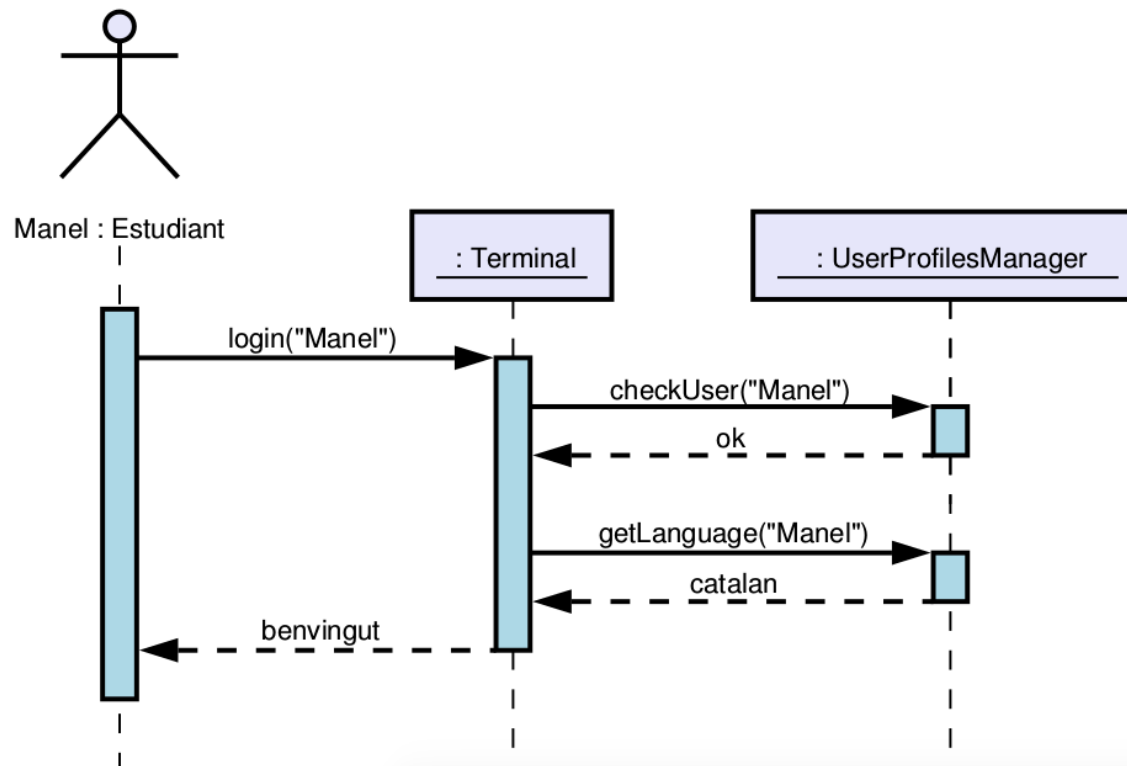
Exemple:



# Diagrama de seqüència

## Concrecions i exemples

Els diagrames de **casos d'ús**, com ja hem vist en capítols anteriors, us presenten una **vista externa** del sistema. Els diagrames d'interacció, que inclouen els de **seqüència**, descriuen com els casos d'ús es realitzen (es fan realitat) com a interacció entre conjunts d'objectes.



# Diagrama de seqüència

## Concrecions i exemples

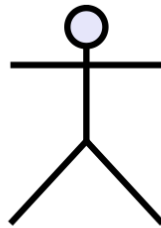
Els diagrames de seqüència mostren el fluxe de missatges (*events*) entre diverses intàncies al llarg del temps. De fet, ofereixen una manera formal d'especificar un escenari.

Es pot veure immediatament l'ordre en què s'envien els missatges perquè el **temps** és una **dimensió explícita** del diagrama (eix vertical).

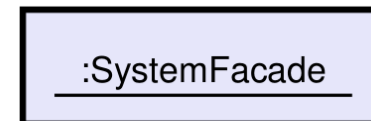
Cada instància es representa mitjançant un recorregut vertical diferent.

És una vista gràfica d'un escenari (un camí concret d'un diagrama de casos d'ús) que il·lustra les interaccions entre objectes com a seqüència temporal. De fet, es pot crear a partir del cas d'ús que descriu.

**Interlocutors:**



John Doe:Client

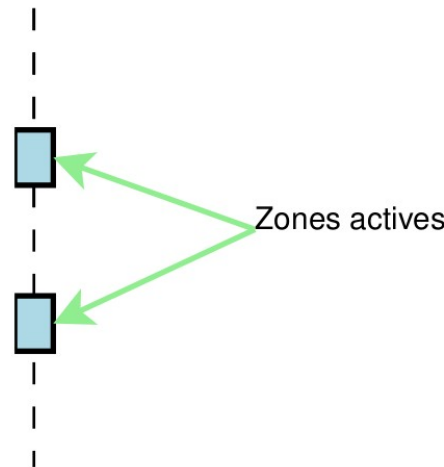


# Diagrama de seqüència

## Elements del diagrama (I)

- **Actors i Objectes:** són els elements que es comuniquen. Com que tant els objectes com els actors, són instàncies, es fa servir la nomenclatura d'objecte amb els dos punts i el nom subratllat.
- **Línies de vida:** són unes línies que representen la vida dels objectes (o actors) al llarg del temps. El temps va de dalt cap baix. La línia es fa gruixuda quan aquest objecte està executant una operació. Diem que l'objecte és actiu en aquell moment. Sovint els objectes deixen d'existir. Per indicar el punt on l'objecte es destrueix, es posa una aspa (o creu) al final de la línia.

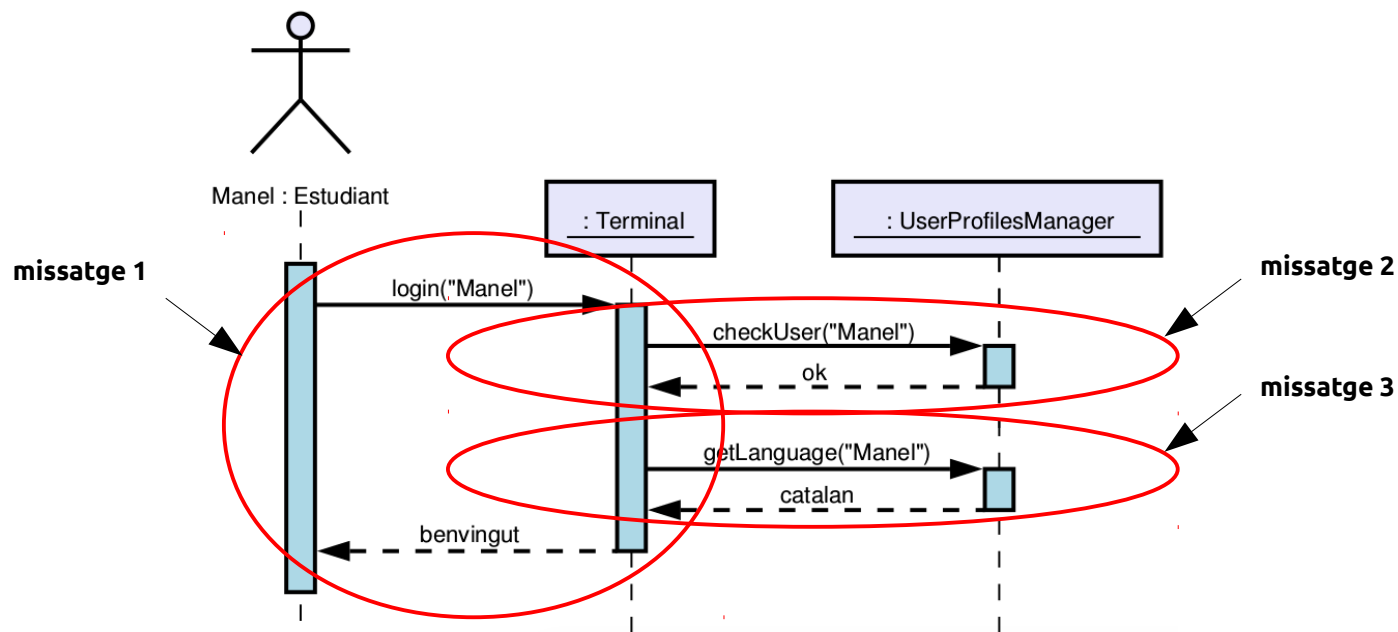
### Exemple de línia de vida:



# Diagrama de seqüència

## Elements del diagrama (II)

- **Missatges:** es representen amb fletxes que van d'emissor a receptor i estan etiquetades amb el missatge que s'envien. La recepció d'un missatge fa que l'objecte iniciï la seva activitat. El retorn de la crida s'indica amb una fletxa a traços. Només és necessari posar-ho si volem fer explícit què es retorna. Si no, es pot suposar que retorna quan la zona activa del receptor acaba.





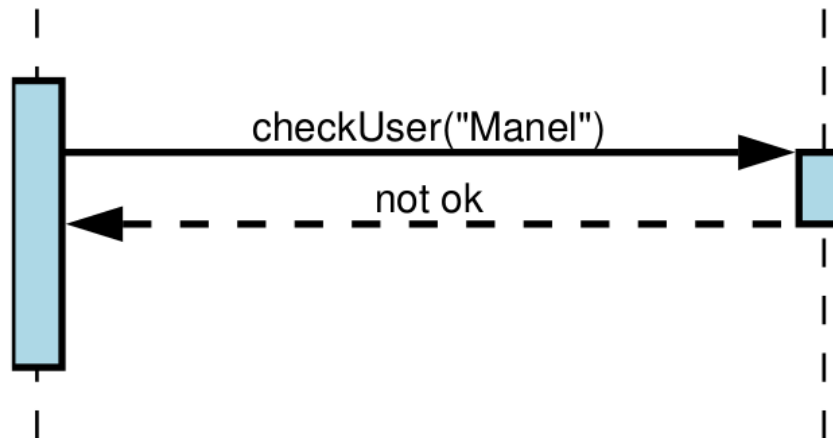
# Diagrama de seqüència

## Estructures de control (I)

Els diagrames de seqüència permeten fer servir algunes estructures de control, que van molt bé per simplificar-los.

Quan fem servir estructures de control no estem representant només un escenari concret, sinó que estem generalitzant.

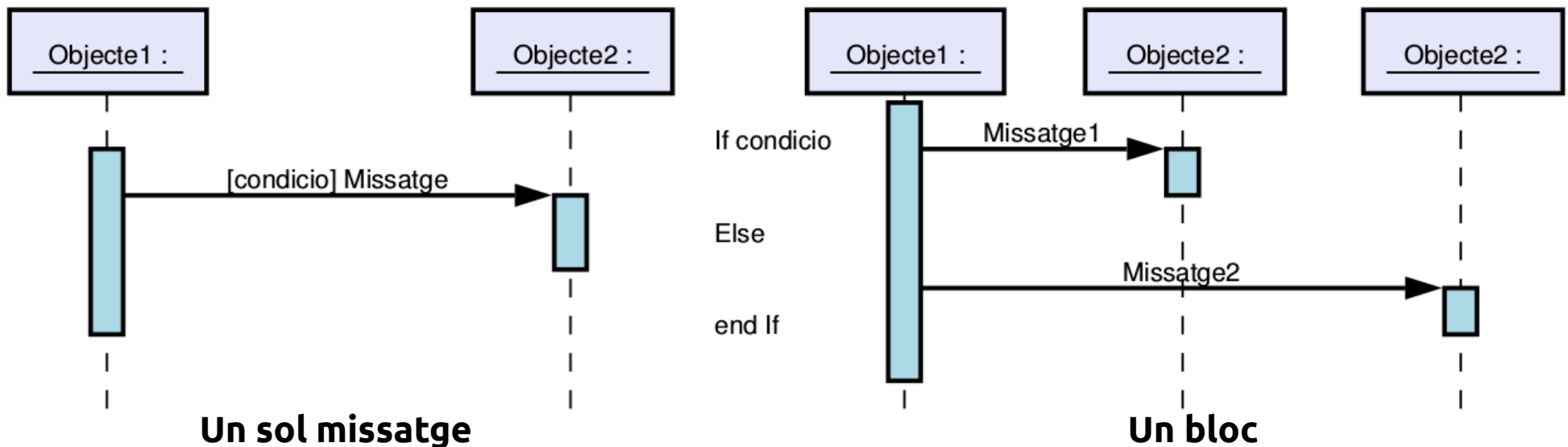
**Exemple d'un missatge entre dos objectes:**



# Diagrama de seqüència

## Estructures de control (II)

Exemple d'execució condicional al diagrama de seqüència:

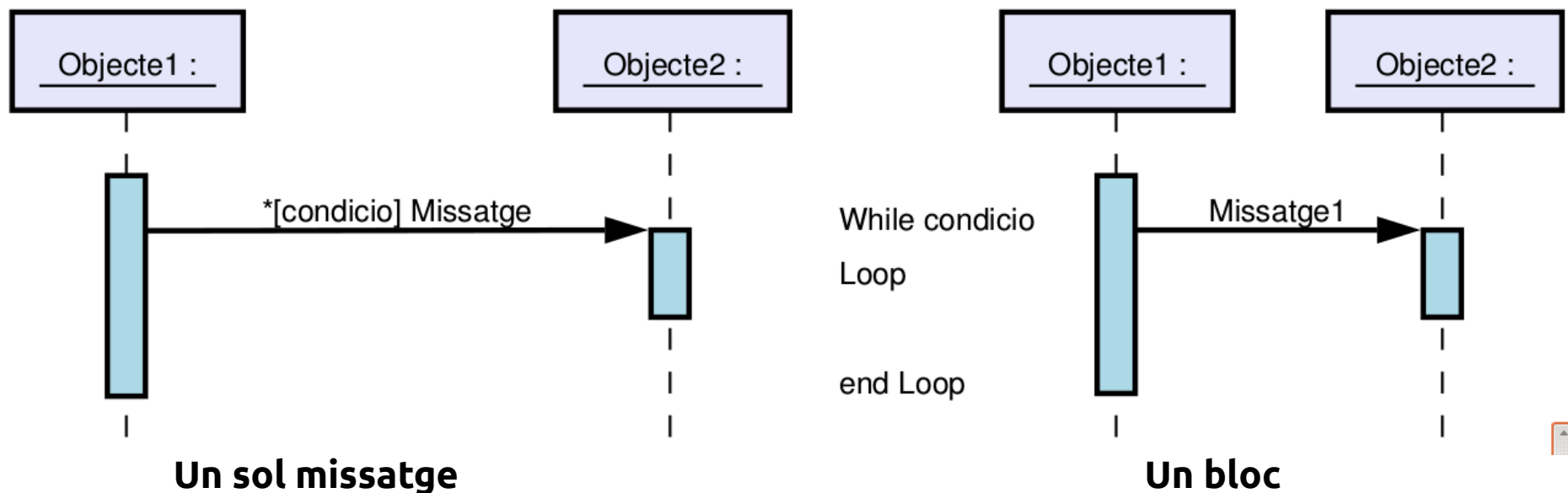


El diagrama indica dues maneres d'expressar alternatives. Una fent una part de la vida condicional i una altra condicionant l'enviament del missatge amb una guarda.

# Diagrama de seqüència

## Estructures de control (II)

Exemple d'iteració al diagrama de seqüència:

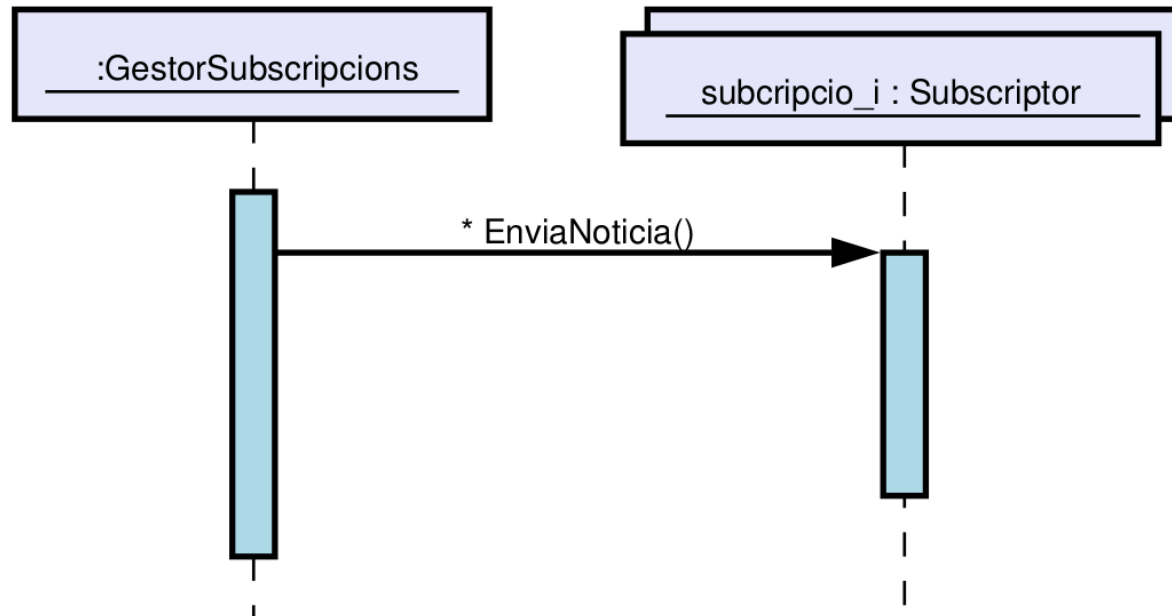


Aquest diagrama indica dues maneres d'expressar una iteració. També, una per tot un bloc de missatges i una altra per iterar l'enviament d'un sol missatge.

# Diagrama de seqüència

## Estructures de control (III)

Exemple de repetició per a cada element:



Si calgués enviar un missatge a cadascun del conjunt d'objectes, podem simplificar el diagrama representant el conjunt d'objectes amb un objecte múltiple com es mostra a la figura superior.

# Diagrama de seqüència

## Creació i destrucció (I)

A vegades un objecte pot **crear** o **eliminar** un altre objecte.

Això cal representar-ho com a **missatges especials**.

Un **missatge de creació** no apunta a una zona activa de la línia de vida, sinó que apunta a l'objecte mateix (ha d'estar més avall que els altres).

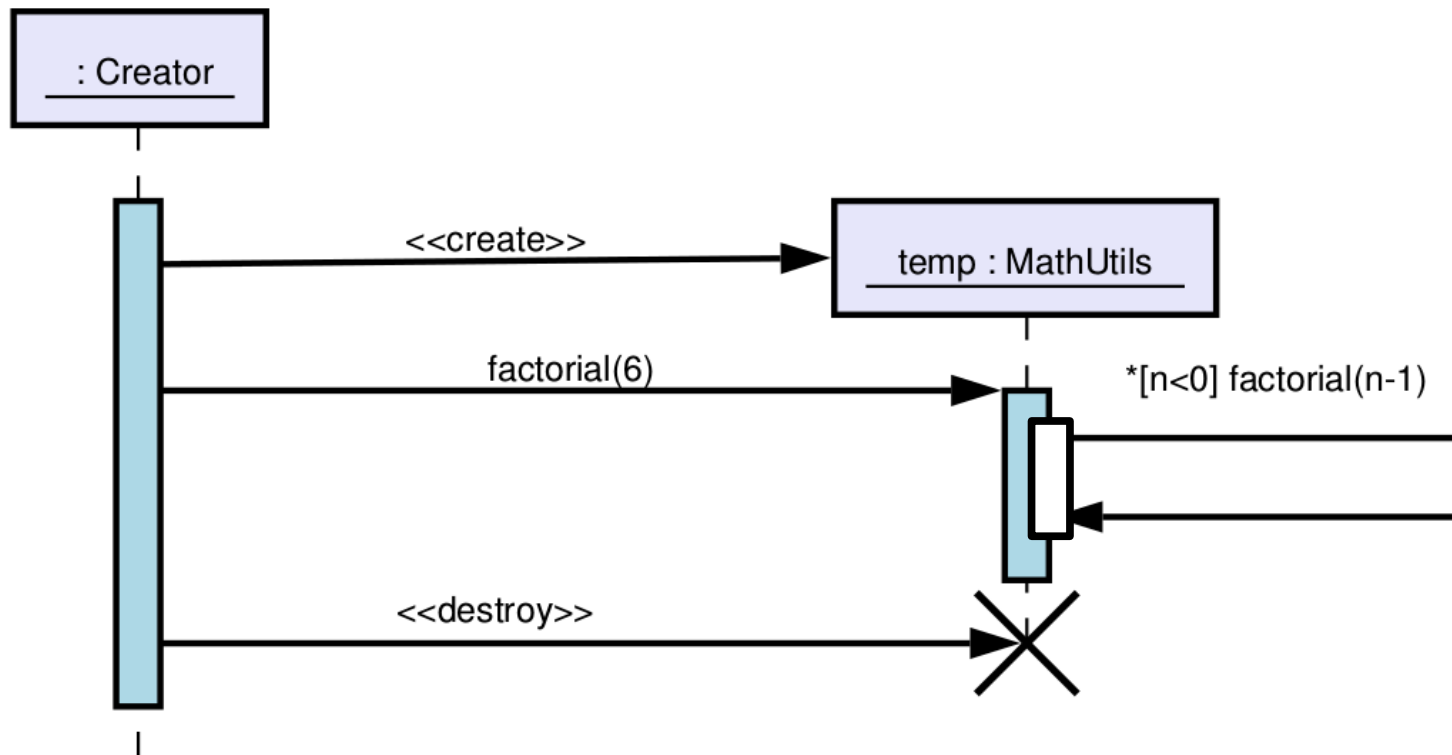
El **missatge de destrucció** apunta cap a una aspa al final de la línia de vida que representa el final de la vida de l'objecte.

Ho podeu estudiar a la diapositiva que ve a continuació...

# Diagrama de seqüència

## Creació i destrucció (II)

Exemple de creació, destrucció i crides a mètodes propis



# Diagrama de seqüència

## Diagrams de Seqüència vs. Col·laboració

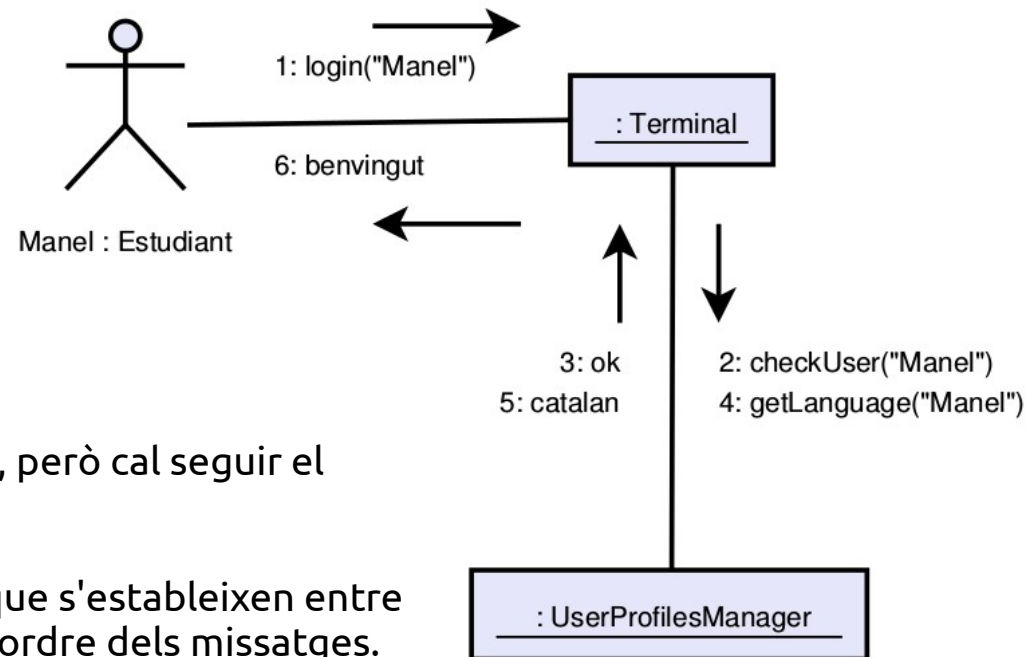
El **diagrama de col·laboració** és, com el diagrama de seqüència, una altra realització dels casos d'ús. Són una combinació dels diagrames d'objecte i els de seqüència. Mostren el fluxe d'esdeveniments entre objectes.

### Exemple de diagrama de col·laboració:

*Recordem que el diagrama de col·laboració mostra com els objectes treballen els uns amb els altres enviant missatges i intercanviant dades. Els diagrames de col·laboració i seqüència són dos vistes diferents de les interaccions entre objectes. Les dues vistes representen exactament la mateixa informació.*

El **diagrama de seqüència** reforça la vista temporal, però cal seguir el diagrama per saber qui es relaciona amb qui.

El **diagrama de col·laboració** reforça les relacions que s'estableixen entre els objectes, però cal seguir el diagrama per saber l'ordre dels missatges.



# Diagrama d'activitats

## Descripció (I)

El **diagrama d'activitats** és una variant del diagrama d'estats i del diagrama d'interacció perquè serveix per a descriure els estats d'una activitat, que és el conjunt d'accions seqüencials o concurrents en la qual hi poden intervenir diversos classificadors.

Aquests diagrames tenen molts elements en comú amb els diagrames de col·laboració i amb els d'estat. Els conceptes de transició, estat, esdeveniment, senyal, flux de control, i objectes, són els mateixos.

Els elements específics són:

- Estats d'acció
- Flux d'objectes
- Estats de flux d'objectes
- Estats de subactivitat
- *Swimlanes*
- Icones de control



# Diagrama d'activitats

## Descripció (II)

Els diagrames d'activitats s'utilitzen per capturar fluxos d'activitats i seqüències de decisions. Són ideals per representar el **comportament intern**:

- d'un mètode (la realització d'una operació),
- d'un cas d'ús, o,
- d'un procés de negoci (*Workflow*)

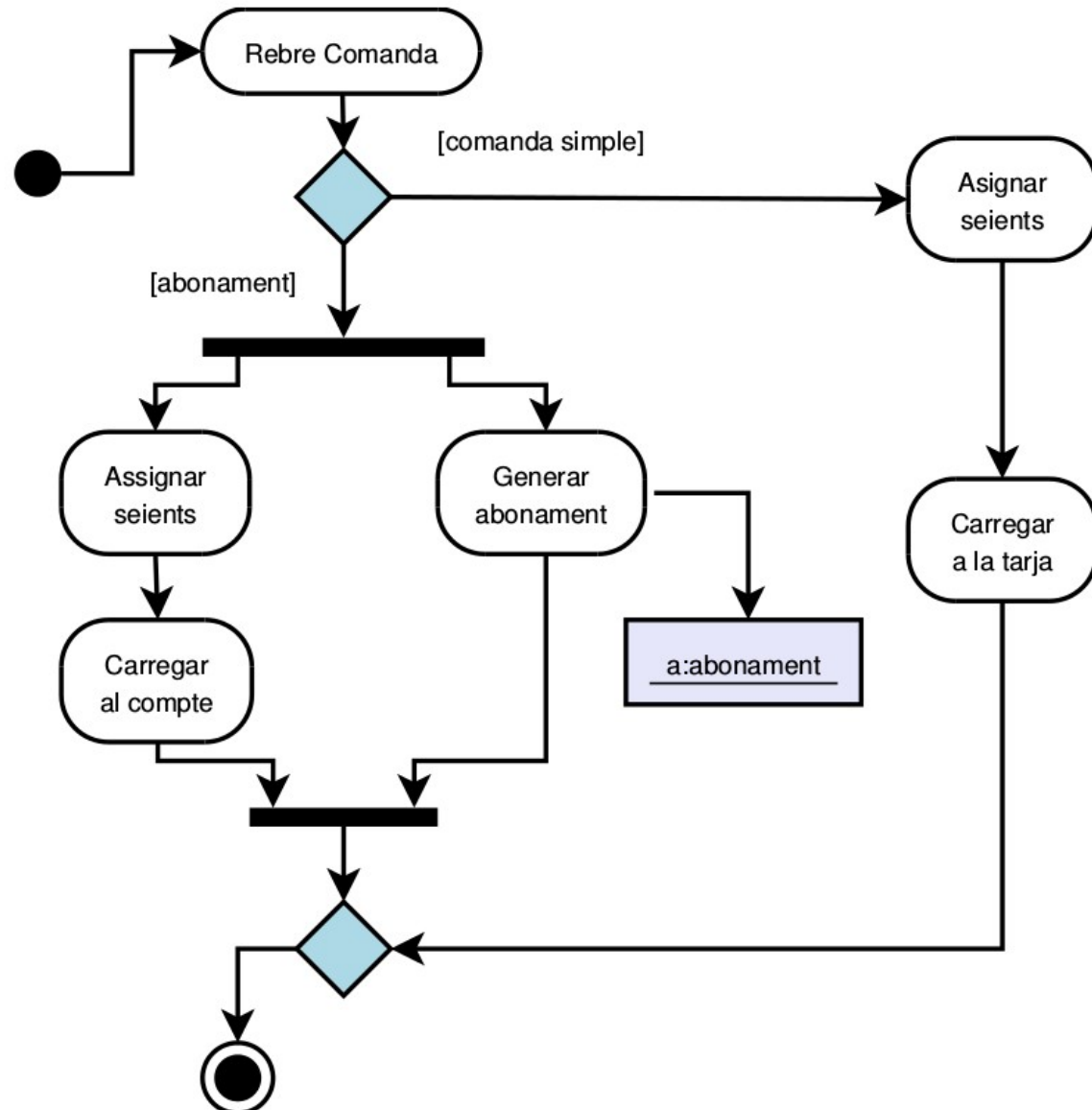
Es poden entendre com una extensió dels diagrames d'estats, on cada estat es caracteritza per estar-hi executant-se una activitat i, on totes les transicions són automàtiques tan bon punt s'acaba l'activitat. Ara bé, s'enriqueix amb altres elements:

- **Punts de decisió**: condicions que canvien el fluxe de les accions.
- **Camins d'execució concurrents**.
- **Senyals**, amb punts d'**enviament** i **recepció** explícits.
- **Objectes** afectats per les **accions**.

# Diagrama d'activitats

## Descripció (III)

Exemple:



En UML 2.0 és possible representar a més a més, condicions, sincronitzacions, senyals, tokens, etc.

# Diagrama d'activitats

## Descripció (IV)

### Activitats i transicions

Les activitats s'**enllacen** per **transicions automàtiques**, i és a través d'aquestes que es passarà d'una activitat a una altra.

**Quan una activitat acaba** es desencadena el **pas a la següent** activitat, automàticament.

Les activitats **no** tenen **transicions internes** ni **transicions desencadenades per esdeveniments**. Podríem dir que les transicions es disparen de manera automàtica, quan finalitza una activitat

És una variació d'una màquina d'estats en la qual els estats representen l'**execució d'accions o subactivitats** i les transicions es **disparen per la finalització** de les accions o subactivitats.

El **diagrama d'activitats** representa la **màquina d'estats** d'un **procediment**.

# Diagrama d'activitats

## Descripció (V)

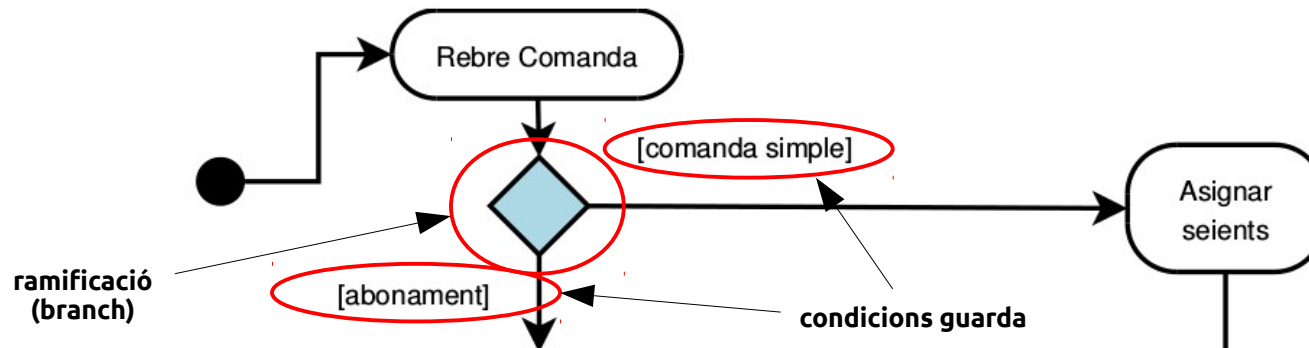
### Ramificacions i fusions

Una ramificació (*branch*) és una divisió del fluxe d'activitats en dos **fil·ls alternatius**. Cada vegada que s'executi el diagrama, s'optarà per un o per l'altre depenent d'una condició.

La **condició** està especificada com a **guarda** a cadascuna de les connexions que en surten.

Una **fusió** (*merge*) és la unió de dos fil·ls alternatius en un sol fil.

Les ramificacions i les fusions es representen amb un **rombus buit**.



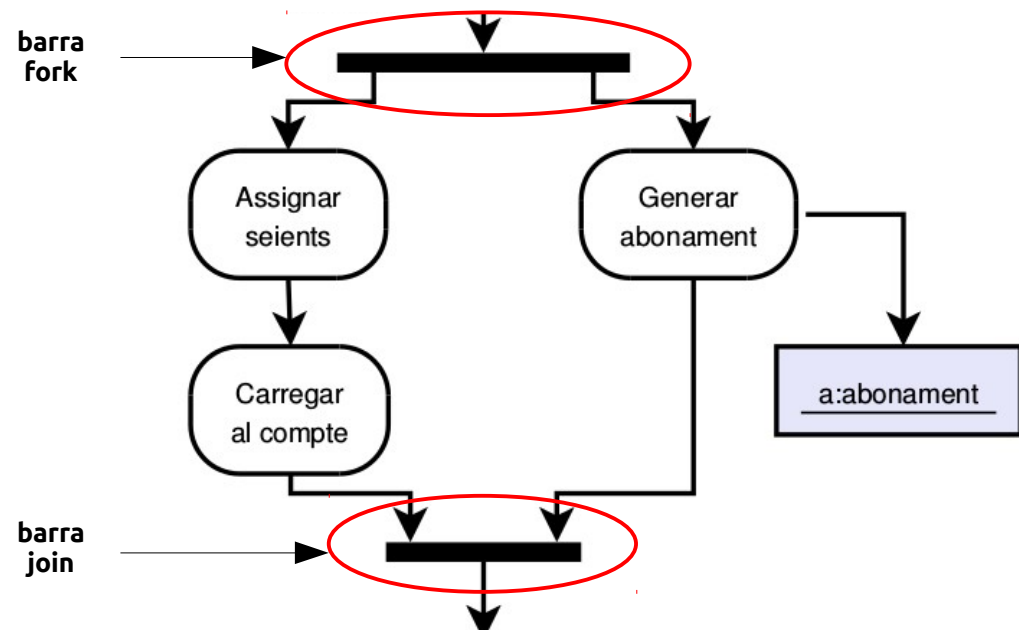
# Diagrama d'activitats

## Descripció (VI)

### Divisions i unions

Les barres de sincronització permeten, igual que les ramificacions i fusions, dividir el fluxe d'activitats en diversos fils i després ajuntar-los de nou. La diferència està en que ara no són fils alternatius, sinó fils **concurrents**. Això vol dir que sempre s'executaran els dos fils i sempre s'executaran alhora.

La barra que separa els fils l'anomenem barra de **fork**, i la barra que ajunta els fils l'anomenem barra de **join**. Cal dir que la barra de **join** aquí té una conseqüència: si un fil arriba abans que l'altre al punt de **join**, caldrà que esperi que l'altre arribi, doncs és un punt de sincronització.



# Diagrama d'activitats

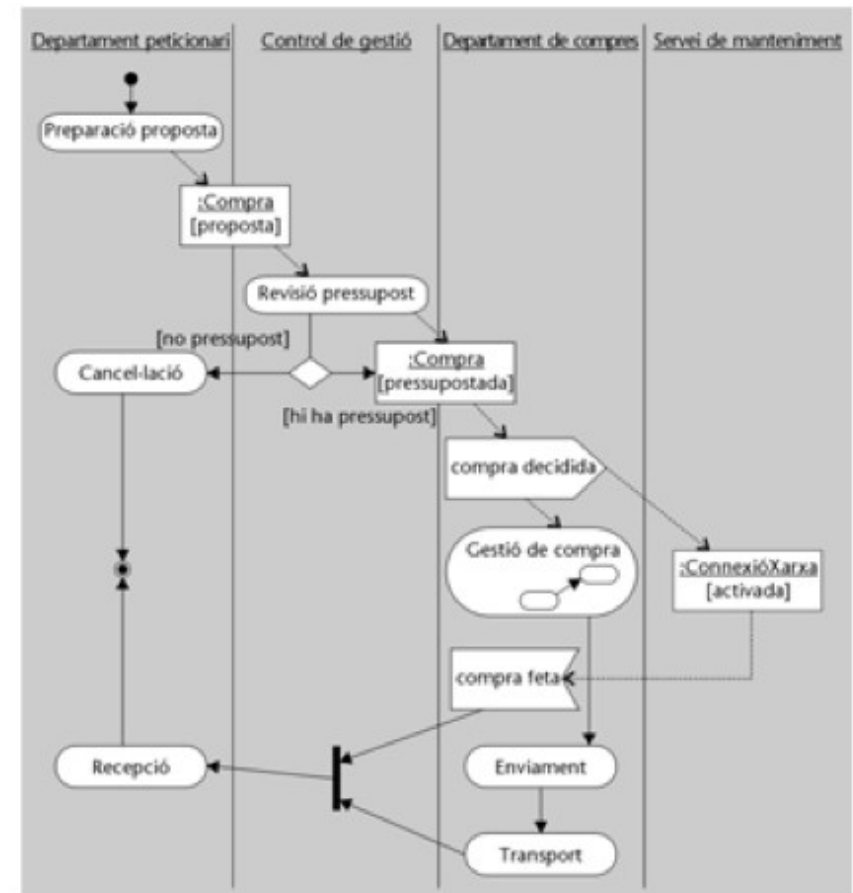
## Descripció (VII)

### Carrils o *swimlanes*

Els carrils serveixen per separar les activitats que realitzen diferents objectes o actors. Unes línies gruixudes a mode de corxera, divideixen en carrils el diagrama. A cada carril hi posarem només les activitats que farà l'actor o objecte a qui pertany el carril.

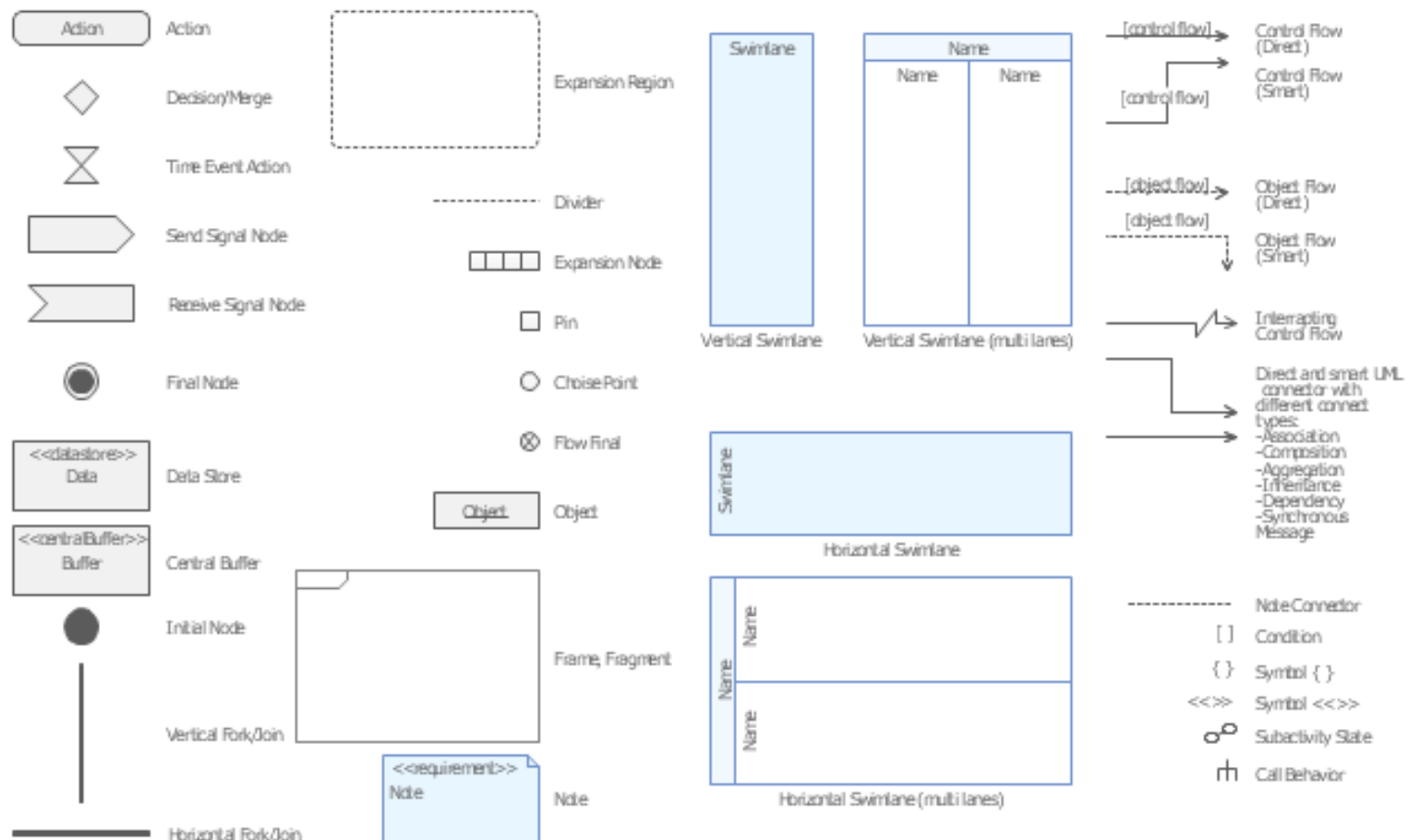
### Flux d'objectes: Entrades i productes

Sovint les activitats requereixen l'existència d'objectes o generen objectes o en canvien el seu estat. Aquests objectes es poden afegir al diagrama amb les convencions del diagrama d'objectes. Podem relacionar els objectes amb les activitats fent servir fletxes que indiquin el fluxe d'objecte.



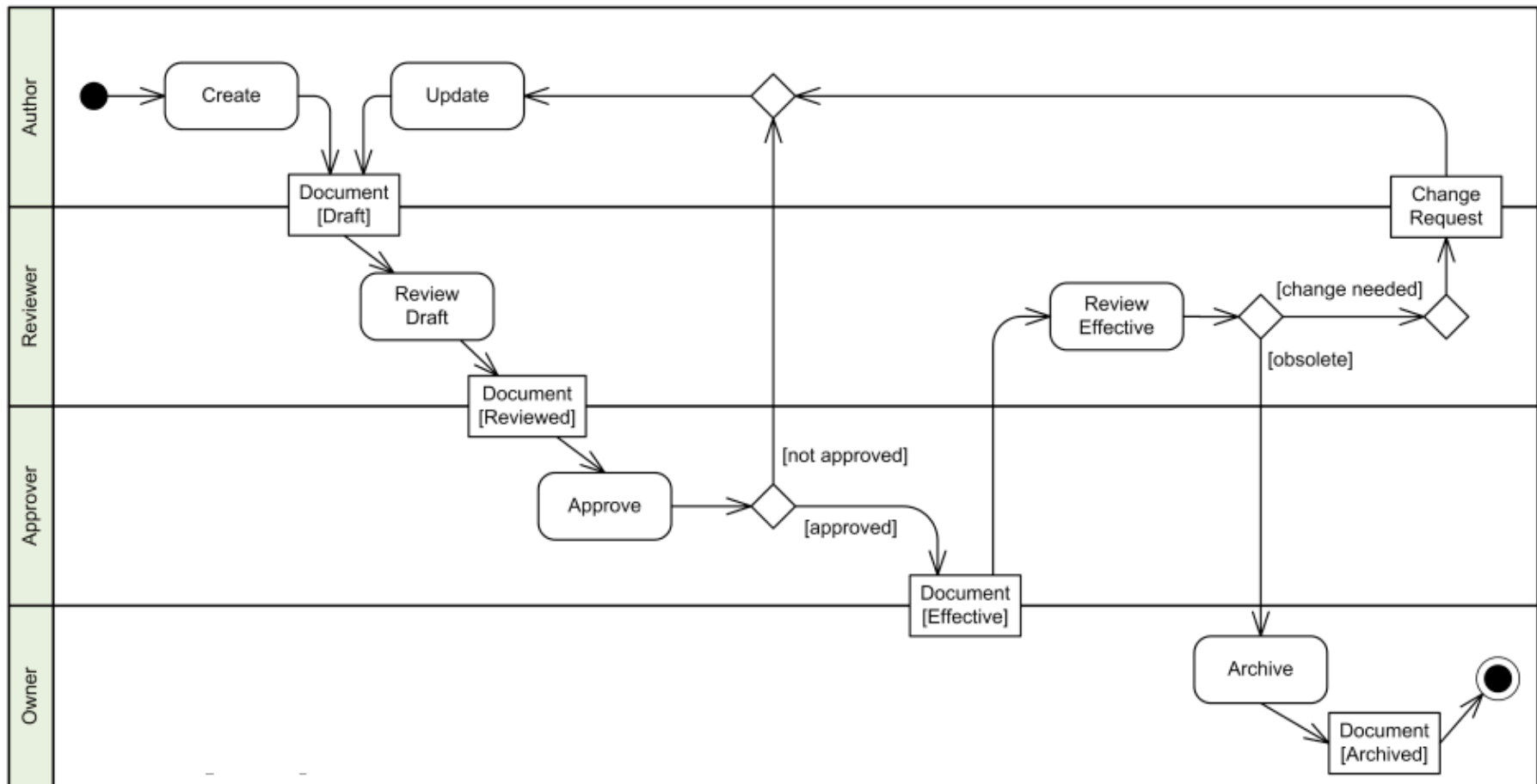
# Diagrama d'activitats

## Flowchart



# Diagrama d'activitats

## Exemple





# Bibliografia

# UF3

## M05 Entorns de Desenvolupament

**INTRODUCCIÓ AL DISSENY  
ORIENTAT A OBJECTES**

- UML. OMG. <http://www.uml.org>
- OMG Unified Modeling Language™. OMG Document Number: formal/2015-03-01.
- Enginyeria del Software. Universitat Politècnica de Catalunya. FIB. Xavier Franch i Dolors Costal.
- UML & Patterns. Ed. Prentice Hall. 2nd Edition. Craig Larman.
- El Lenguaje Unificado de Modelado. Manual de Referencia. Ed. Addison Wesley. VVAA
- Enginyeria del Software I. Apunts de teoria. Univesitat Pompeu Fabra. VVAA.
- Enginyeria del Software I. Problemes. Universitat Pompeu Fabra. VVAA.
- Enginyeria del Programari Orientada a Objectes. Universitat de Girona. Toni Sellarès.
- Enginyeria del Programari Benet Campderrich Falgueras. Universitat Oberta de Catalunya.
- Diagrama d'activitat i d'estats. Universitat de Castelló. Departament LSI. Reyes Grangel Seguer.
- Desarrollo Orientado a Objetos con UML. Programación C.E.C.yT. Juan de Dios Bátiz Paredes – IPN.
- Modelos y Herramientas UML: Modelo Estructurado. Máster en Computación. Elena Mediavilla.
- Ingeniería del Software. Facultad de Informática de San Sebastián UPV/EHU. Germán Rigau.
- Ingeniería del Software. Universidad de Sevilla. Departamento de Informática y Automatas. VVAA.
- Diagramas UML de casos de uso y requisitos. Universidad de Sevilla. Dr. Javier Jesús Gutiérrez.
- Ingeniería del Software. UNED. VVAA.
- Introduction to OCL. Universität Koblenz-Landau. Bernhard Beckert.

