



# UML

# UML

- UML (Llenguatge unificat de modelatge) son un conjunt d'eines que ens permeten generar dissenys que capturin les nostres idees d'una manera convencional i fàcil de transmetre.
- El conjunt de diagrames que representen el sistema des de diferents perspectives es coneix com a model.



# UML

- Al model podem trobar
  - Diagrama de classes
  - Diagrama de objectes
  - Diagrama de casos d'ús
  - Diagrama de estats
  - Diagrama de seqüència
  - Diagrama d'activitats
  - Diagrama de col·laboracions
  - Diagrama de components
  - Diagrama de distribució

# UML – Casos d'ús.

- Es una estructura que ajuda als analistes a treballar amb els usuaris per determinar de la manera que ha de funcionar un sistema.
- Cal saber quines necessitats es tenen (requeriments)
- Un cas d'us es una col·lecció de situacions respecte l'ús d'un sistema. Cadascun d'aquests escenaris defineix una seqüència d'events.



# UML – Casos d'ús

- Les diferents seqüències d'un cas d'us s'inicien per una persona, un altre sistema, una part de hardware o per el pas del temps.
- Les entitats que inicien una seqüència s'anomenen ACTORS
- Tota seqüència te com a resultat quelcom utilitzable, ja sigui per l'actor que la ha iniciat com per un altre actor.

# UML – Casos d'ús

Exemple. La màquina de begudes.

- El client selecciona la beguda.
- El client introdueix l'import
- La màquina entrega la beguda
- El client agafa la beguda.



# UML – Casos d'ús

Exemple. La màquina de begudes.

Hi ha consideracions addicionals al escenari.

- Un estat inicial en que el client te set
- Un estat final en el que el client obté una beguda.
- També tenim alternatives
  - Què passa si no hi ha canvi?
  - Què passa si el producte esta esgotat?

# UML – casos d'ús

De vegades pot passar que un cas d'ús pressenti el procés des del punt de vista d'un actor, però pot ser que intervinguin també altres usuaris.

Ens caldrà doncs definir casos d'ús per cadascun d'aquests diferents escenaris.



# UML – Casos d'ús

Exemple. La màquina de begudes.

- Distingim dos casos addicionals
  - Recol·lecció dels diners.
  - Restablir begudes.

# UML – Casos d'ús

Exemple. La màquina de begudes.

- En la recol·lecció dels diners tenim que
  - El recol·lector obre el sistema de tancament de la màquina.
  - El recol·lector obre la porta de la màquina
  - El recol·lector fixa la porta de la màquina
  - El recol·lector recupera els diners de la caixa.
  - El recol·lector deixa un mínim de moneda petita
  - El recol·lector tanca la màquina.



# UML – Casos d'ús

Exemple. La màquina de begudes.

- En l'abastiment de begudes tenim
  - El proveïdor obre el sistema de tancament de la màquina.
  - El proveïdor obre la porta de la màquina
  - El proveïdor fixa la porta de la màquina
  - El proveïdor col·loca les diferents begudes als seus compartiments
  - El proveïdor tanca la porta de la màquina.
  - El proveïdor tanca el sistema de tancament de la màquina.

# UML – Casos d'ús

## Relacions entre casos d'ús.

- La més senzilla de totes es la relació **d'associació**.
- Indica la invocació des de un actor o una operació (cas d'ús) a una altre cas d'us



# UML – Casos d'ús

## Relacions entre casos d'ús.

- Ens podem trobar que dos casos d'ús facin servir la mateixa seqüència de passos.
- En aquest cas podem definir un cas d'ús que agrupi aquests passos i fer que els dos casos originals facin referència a aquest nou cas d'ús.
- A aquest tipus de relació entre cas d'us s'anomena relació d'**inclusió**.
- Direm que els casos originals inclouen el cas comú.

# UML – Casos d'ús

Exemple. La màquina de begudes.

- Si repassem els processos d'abastiment i recol·lecció, veurem que hi ha subprocessos (obrir la màquina i tancar la màquina) comuns a tots dos.
- Podem definir dos casos d'ús, obrir màquina de begudes i tancar màquina de begudes.
- Als casos originals inclourem aquests nous dos casos comuns.



# UML – Casos d'ús

## Relacions entre casos d'ús.

- Pot donar-se el cas que un cas d'ús s'assembli molt a un ja definit, però que aquest segon faci alguna cosa més.
- En aquest cas podem incorporar de manera implícita dins el segon cas d'ús el comportament del primer a un punt especificat.
- Aquest tipus de relació s'anomena **d'extensió**.
- Direm que un cas d'ús extens un altre cas d'ús.

# UML – Casos d'ús

Exemple. La màquina de begudes.

- Imaginem-nos que al procés d'abastiment de begudes, hi afegim la possibilitat que el proveïdor, abans de restablir les begudes, prengui nota de quines s'han venut més i quina menys. En cas de que una beguda no es vengui gens, podrà substituir-la per la més venuda.
- Per tant, caldrà que canviï l'etiquetatge de la màquina.
- En aquest cas necessitarem un nou cas d'ús que extengui el cas d'us d'abastiment.



# UML – Casos d'ús

- Altres relacions
  - La relació de **dependència** es dona quan un cas d'ús depèn de la realització d'un altre cas.
  - La relació de **generalització** es dona quan tenim que un cas d'ús hereta d'un altre.
  - L'**agrupament** es una manera d'organitzar casos d'ús.

# UML – Casos d'ús

Exercici.

Tenim que fer un gestor personal que inclogui un directori de contactes i un calendari on emmagatzemar les nostres cites i un sistema de control de tasques.

- La primera tasca a realitzar serà una entrevista amb l'usuari final on establirem quines funcionalitats ha de incloure aquesta aplicació.



# UML – Casos d'ús

Exercici – Gestor personal

Realitzar els diagrames per els casos d'ús de la nostra aplicació gestor personal.

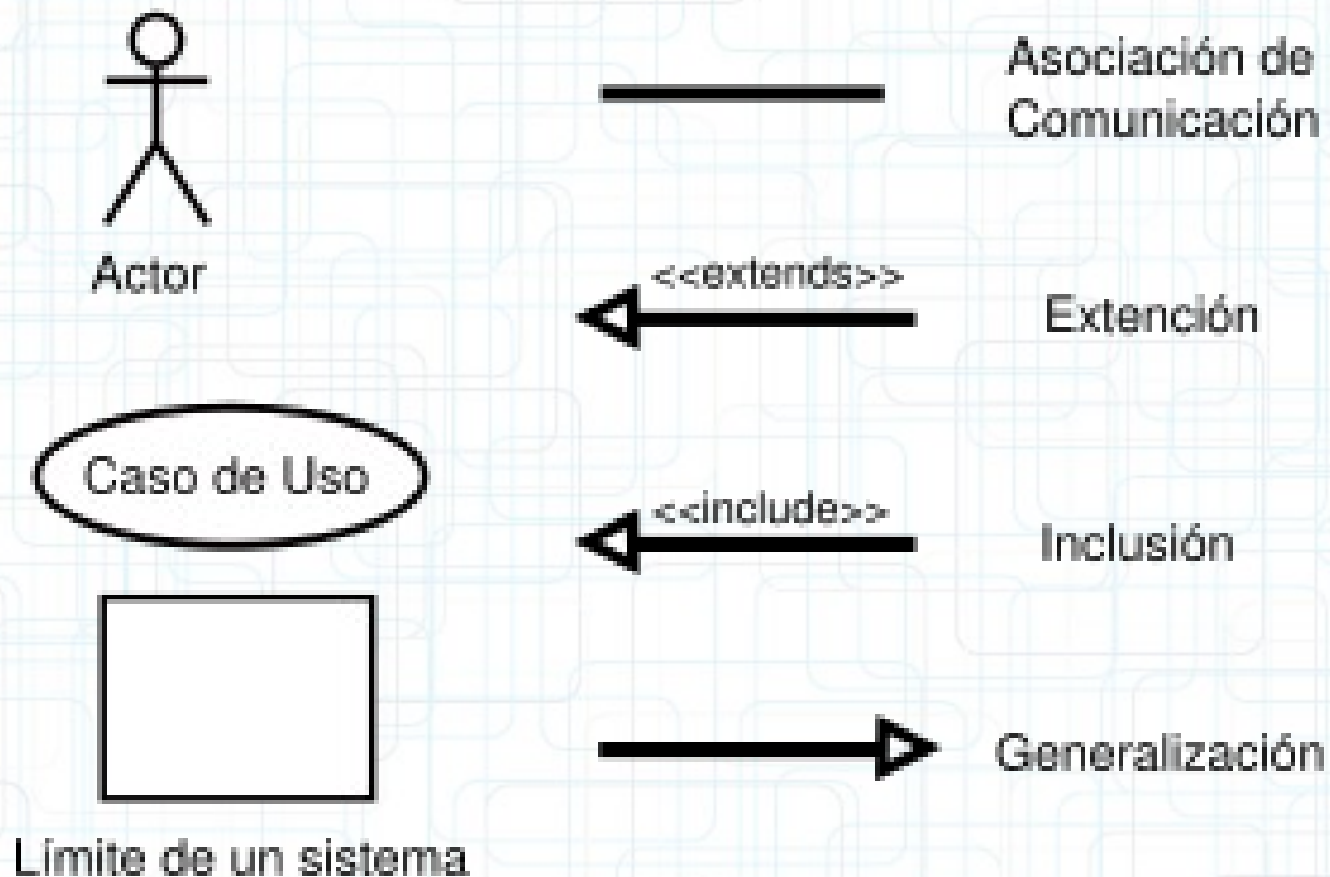
# UML - Diagrames Cas d'ús

Els diagrames de cas d'ús ens permet identificar els elements primaris (Actors) i processos (Casos d'ús) que formen el sistema.

En aquests diagrames podem identificar quins actors interactuen amb cada cas d'ús.

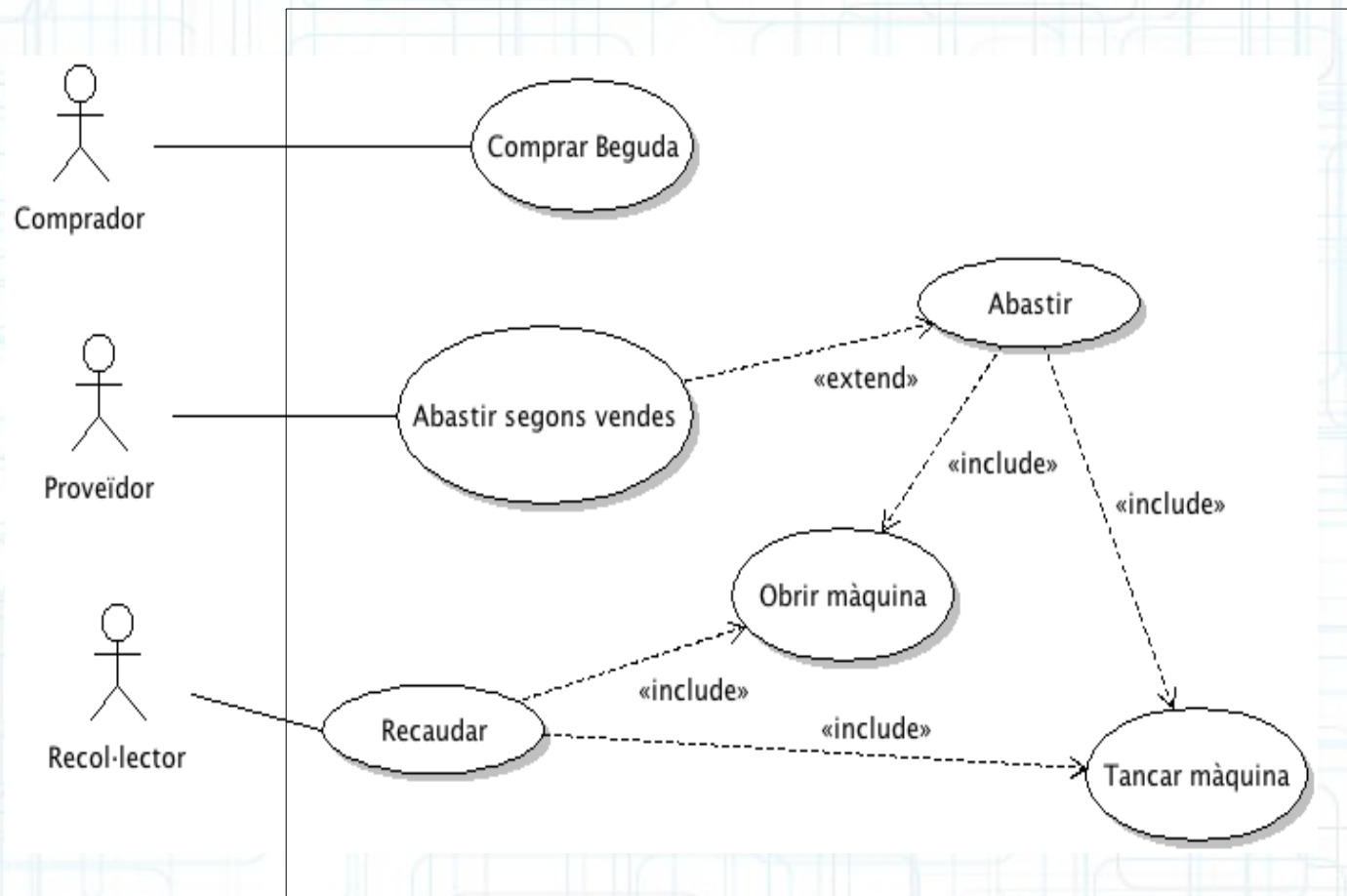


# UML - Diagrames Cas d'ús



# UML - Diagrames Cas d'ús

Exemple. La màquina de begudes..





# UML - Diagrames Cas d'ús

Els diagrames no inclouen descripció dels passos. Cal afegir-los com a documentació addicional on inclourem

- L'actor que inicia el procés.
- Condicions prèvies per el cas d'us.
- Passos dins l'escenari.
- Condicions posteriors a l'execució.
- L'actor que es beneficia en cada cas.

# UML – Diagrames cas d'ús

- Exercici.
  - Realitzar el diagrama del cas d'us per fer servir un comandament a distancia de la tele, amb les funcions principals (volum, canviar de canal, encendre, apagar)



# UML – Casos d'ús

## Exercici – Gestor Personal

- Cal realitzar els diagrames de cas d'ús per la nostra aplicació gestor personal.

# UML Diagrama de Classes

Serveix per visualitzar i identificar les classes que defineixen el sistema, així com les relacions que s'estableixen entre elles.

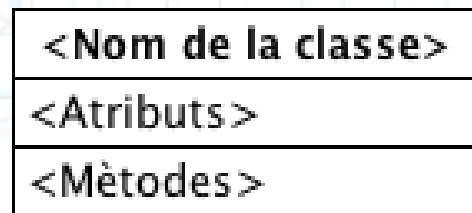
Les relacions entre les classes poden ser associatives, d'herència, d'ús i de conteniment.



# UML Diagrama de Classes

## Elements del diagrama de classes

- La classe es representa amb un rectangle dividit en tres parts
  - La superior per indicar el nom de la classe
  - La central per indicar els atributs
  - La tercera per indicar els mètodes.



# UML Diagrama de Classes

Depenent del la visibilitat amb el seu entorn, els atributs i els mètodes d'una classe poden ser

- Públics. Es representen amb el signe +
- Privats. Es representen amb el signe -
- Protected. Es representen amb el signe #

Les eines CASE poden fer servir icones específics per els diferents nivell d'access



# UML Diagrama de Classes

- El tipus de dades del atribut i el de retron d'un mètode s'indiquen a continuació del nom del mateix, precedits de :
- Als mètodes també indicarem els paràmetres llistats entre parèntesis.
  - El tipus precedit de dades de cada paràmetre s'inclou precedit de :

# UML Diagrama de Classes

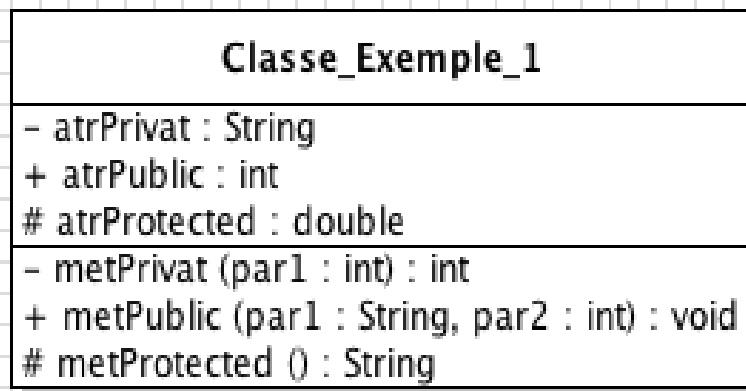


Diagrama de classe.



# UML Diagrama de Classes

- El segon element a indicar son les relacions entre les classes.
- Un punt a tenir en compte es la **Cardinalitat** de la relació que indica el grau i nivell de dependència de les classes.
- Les possibles cardinalitats a UML son
  - Un o molts (  $1..n$  /  $1..\infty$  )
  - Zero o molts (  $0..n$  /  $0..\infty$  )
  - Numero fixe (n)

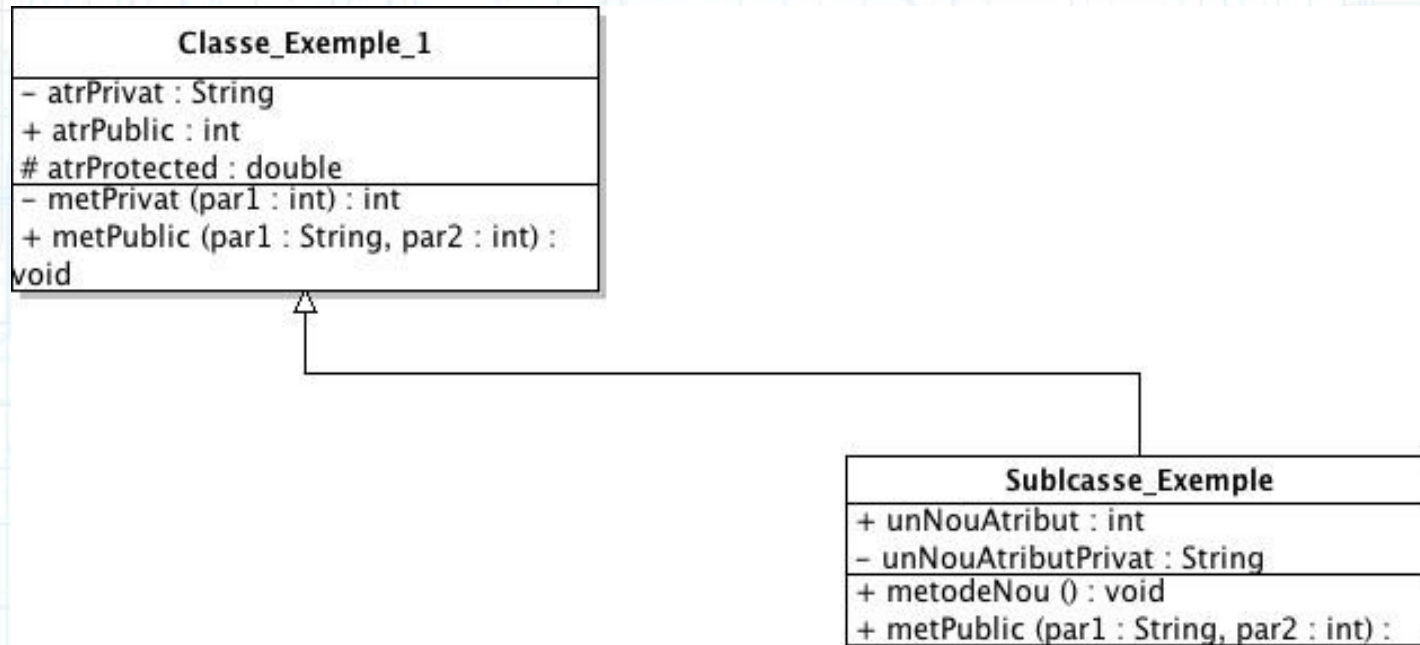
# UML Diagrama de Classes

- Relació d'herència (especialització / generalització).
  - Indica que una subclasse hereta els mètodes i atributs públics i protected d'una superclasse.
  - Adicionalment la subclasse posseeix nous mètodes i atributs propis.
  - Indicarem també els mètodes de la superclasse que es sobrescriguin.



# UML Diagrama de Classes

La relació d'herència s'indica en els diagrames amb una fletxa de punta blanca que va des de la subclasse a la superclasse



# UML Diagrama de Classes

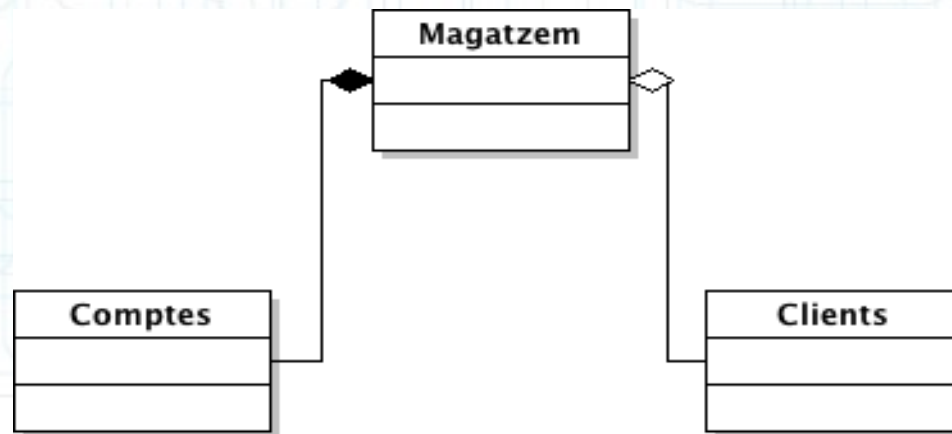
- La relació d'**Agregació** ens permet indicar que una classe fa servir com a atribut objectes d'una altra classe.
- Es parla de agregació **per valor** si el temps de vida del objecte inclòs es troba lligat al del objecte que l'inclou.
- Es parla de agregació **per referencia** si el temps de vida l'objecte inclòs es independent del temps de vida del objecte que l'inclou.



# UML Diagrama de Classes

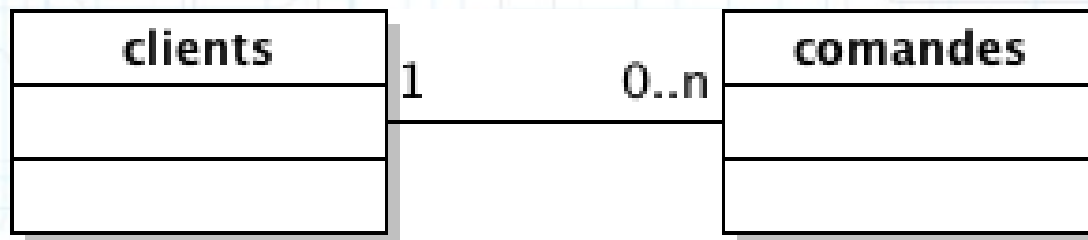
La relació d'agregació es representa amb una fletxa amb un rombe al seu inici.

- Un rombe blanc indica agregació (per referencia).
- Un rombe negre indica composició (per valor)



# UML Diagrama de Classes

- La relació d'**associació** permet indicar objectes que col·laboren entre si.
- No es una relació forta. Els temps de vida dels objectes no depenen entre ells.
- Es representa amb una línia, indicant la cardinalitat.





# UML Diagrama de Classes

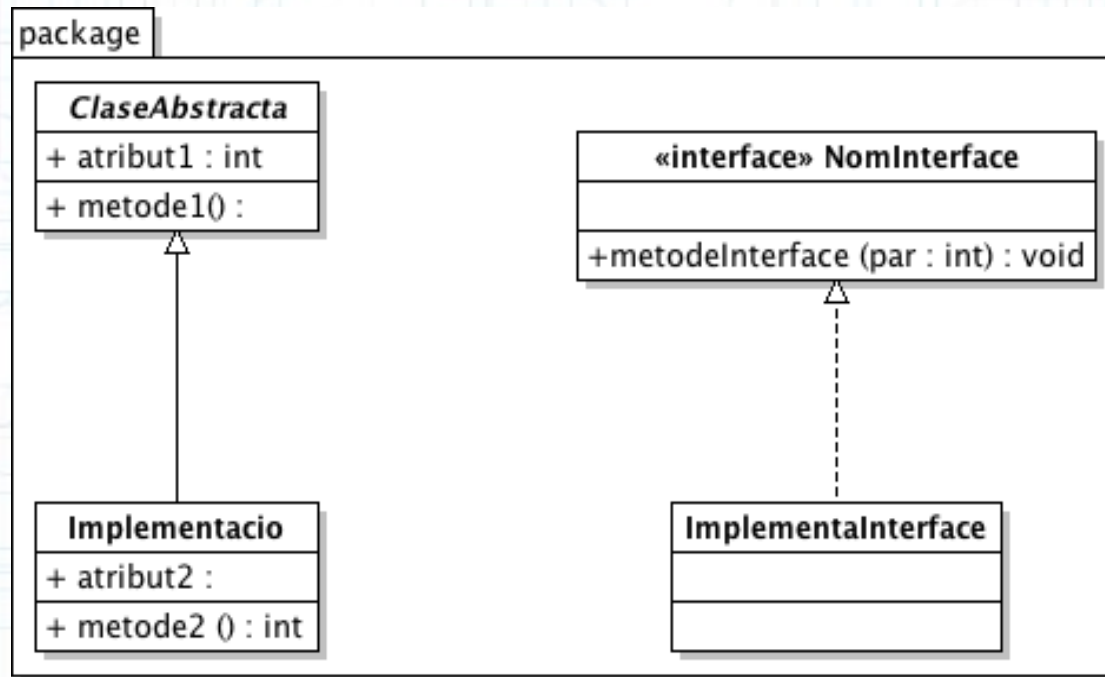
- La relació de **Dependència** o d'**Instanciació** serveix per indicar que una classe es instanciada per una altra,
- La classe instanciada no s'emmagatzema a la classe que la crida.
- Es representa amb una fletxa discontinua



# UML Diagrama de Classes

- Les classes abstractes les indicarem amb el seu nom en *cursiva*. Les classes que l'implementen es relacionaran fent servir herència.
- Els interfaces els representarem com una classe, però afegint <<interface>> al seu nom i indicant el nom en cursiva.
- Podem agrupar les classes en paquets, que es representen com una carpeta.

# UML Diagrama de Classes





# UML Diagrama de Classes

- Exercici. Gestor Personal.
  - Per la nostra aplicació de gestor personal, realitzem el diagrama de classes indicant les seves relacions.

# UML Diagrama d'estat

Els diagrames d'estat ens permeten presentar els diferents estats en els que es poden trobar els objectes del nostre sistema, així com les transicions entre els mateixos, mostrant els punts inicials i final dels mateixos.

Els canvis d'estat poden venir donats com a resposta d'un succés o per el pas del temps.

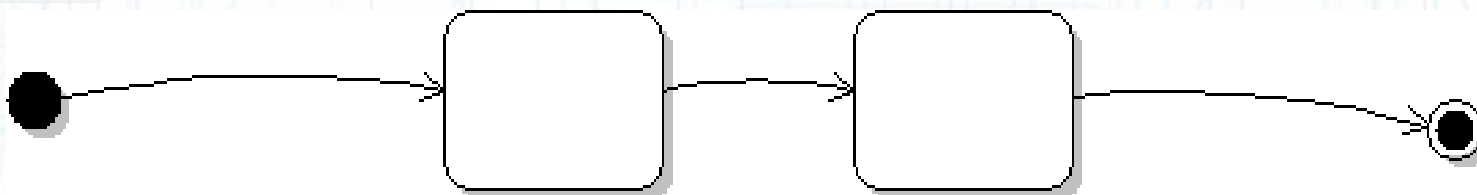


# UML Diagrama d'estat

Representarem l'estat per un rectangle de cantonades rodones amb el nom de l'estat al seu interior.

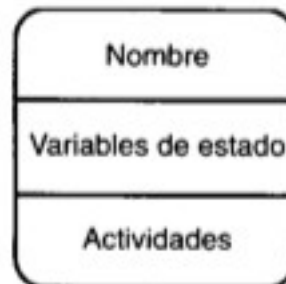
Indicarem l'inici del procés amb un cercle negre i el final amb un cercle negre inscrit en un de blanc.

Les transicions s'indiquen amb una fletxa amb el nom de la transició.



# UML Diagrama d'estat

- La transició indica l'event que causa el canvi d'estat. Si es posa entre corxets [], indiquem que s'ha de complir una condició de seguretat
- Els estats els podem dividir en tres parts, indicant el nom en la part superior, les variables d'estat al centre i les activitats o accions que pot realitzar en la part inferior.





# UML Diagrama d'estats

Exercici.

- Fer el diagrama d'estats per el procés d'una rentadora.

# UML Diagrama d'estats

- Exercici. Gestor personal.
  - Identificar quins objectes de la nostra aplicació poden variar el seu estat i representar el seu diagrama d'estats.

# UML Diagrama de seqüències

El diagrama de seqüències ens permet definir com els objectes es comuniquen entre ells al llarg del temps.

Els objectes es representen com a rectangles a la part superior del diagrama i d'ells parteix la línia de vida, que ens marca el component temporal.

Sobre la línia de vida es situen els rectangles d'activació

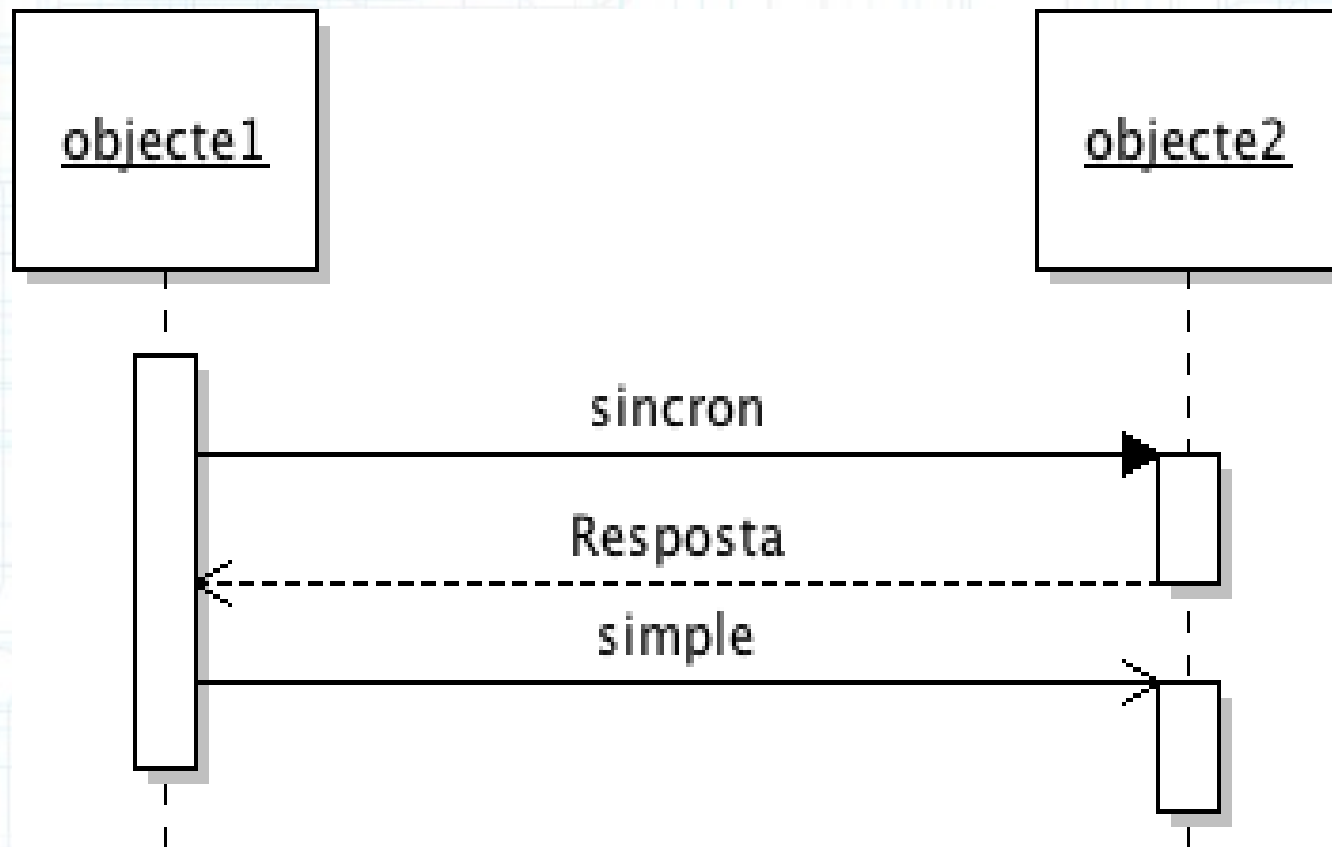


# UML Diagrama de seqüències

Els diferents objectes s'intercanvien missatges, que poden ser

- Simples. Que suposen la transferència del control de la aplicació d'un a l'altre
- Sincrons. En aquest cas l'objecte que fa la crida ha d'esperar resposta del segon objecte per tal de poder seguir el seu fluxe.
- Asincron. L'objecte fa una crida a un segon però no espera resposta abans de continuar.

# UML Diagrama de seqüències



# UML Diagrama de seqüències

- Per representar bucles o bifurcacions col·locarem les seqüències afectades dins un rectangle, indicant a la part superior esquerra la paraula loop i les condicions que el executen.





# UML Diagrama de seqüències

Exercici.

- Fer el diagrama de seqüències de l'enviament d'un fax.

# UML Diagrama de seqüències

## Exercici. Gestor personal

- Realitzar el diagrama de seqüències per els processos d'alta, baixa i edició.

# Diagrama de col·laboracions

- El diagrama de col·laboracions ens mostra com els objectes col·laboren entre ells per realitzar una tasca.
- Aquest diagrama mostra els objectes que intervenen, justament amb els missatges que s'envien entre ells.

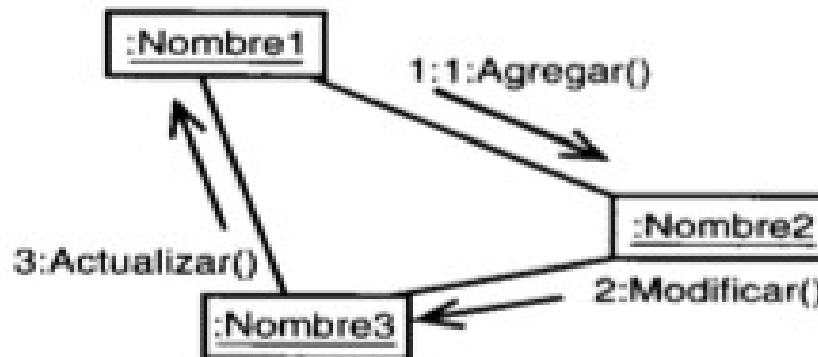


# Diagrama de col·laboracions

- A diferencia del diagrama de seqüències, que feia palesa l'ordre de les interaccions, aquest diagrama fa incís en el context i en la organització general dels objectes que interactuen.
- El diagrama de Seqüències mostra el component temporal, mentre que el de col·laboracions mostra el component espacial.

# Diagrama de col·laboracions

- Representarem dos objectes com a rectangles i afegirem una línia d'associació entre ells.
- Al costat de la línia d'associació, hi posarem una o més fletxes que indiquin els missatges que intercanvien entre ells.





# Diagrama de col·laboracions

Si volem expressar un canvi d'estat, cal fer el següent

- Representar el objecte dos cops, amb el seus estats inicial i final indicats entre parèntesis al costat del nom.
- Els unim amb una fletxa discontinua amb l'etiqueta <<es torna>>



# Diagrama de col·laboracions

Exercici. Gestor d'aplicacions.

- Fer el diagrama de col·laboracions per els processos d'alta, baixa i modificació.

# UML Diagrama de seqüències

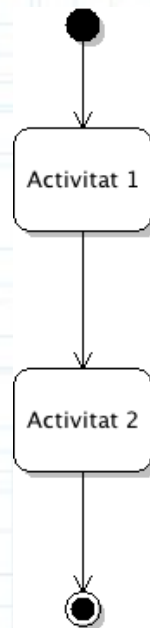
## Exercici. Gestor Personal

- Definir el diagrama de seqüències per crear un nou event, un nou contacte i una nova tasca en el nostre programari.

# UML Diagrama d'activitats

Formalment es molt similar als diagrames de flux.

Representa els fluxos de treball pas a pas dels components d'un sistema.





# UML Diagrama d'activitats

Podem indicar decisions amb un rombe del que surten dos o més camins (fletxes). En aquestes fletxes indicarem entre parèntesis quadrats la condició que cal satisfer per fer la bifurcació.

# UML Diagrama d'activitats



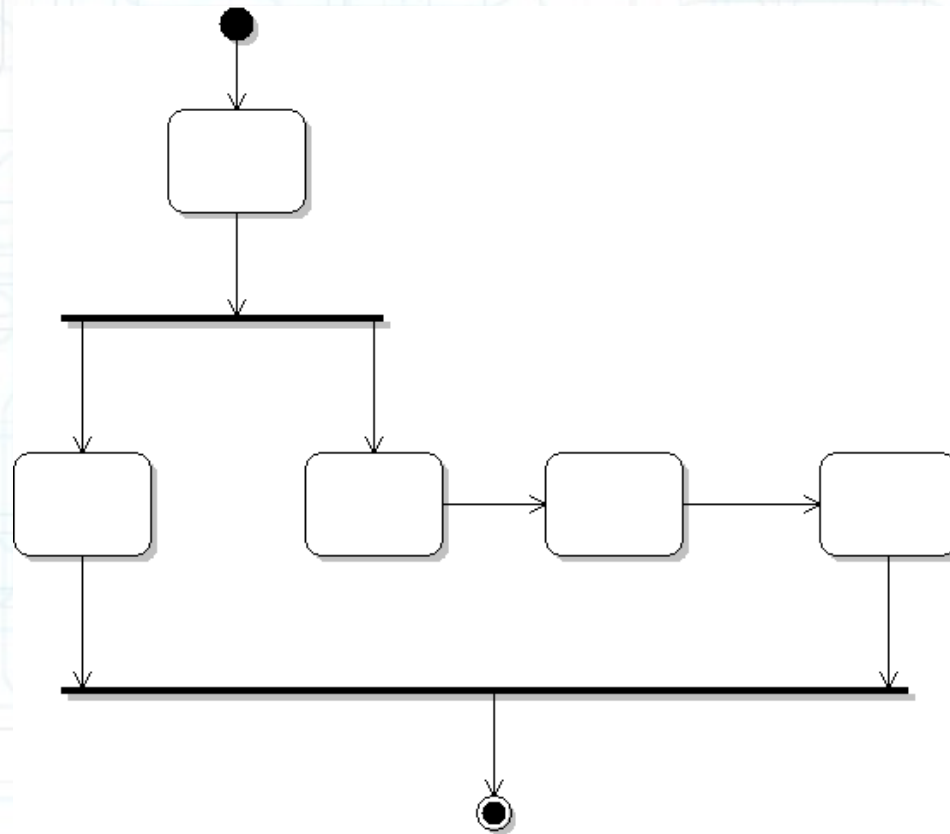
# UML Diagrama d'activitats

Podem indicar també processos de flux paral·lels. Per fer-ho cal que indiquem la bifurcació amb una barra perpendicular al flux.

Aquesta mateix element el farem servir per el punt on els dos processos paral·lels s'ajuntin



# UML Diagrama d'activitats



# UML Diagrama d'activitats

- En ocasions podem necessitar que dos activitats intercanviïn missatges durant un procés col·laboratiu. En aquest cas farem servir els símbols



# UML Diagrama d'activitats

- Aquest diagrama ens permet també establir marcs de treball, podent indicar quin actor té la responsabilitat de la activitat.
- Per fer-ho dividirem el fons en diferents rectangles, assignant un per cada actor i col·locarem els elements del flux a cada una de les regions creades.



# UML Diagrama d'activitats

