Entorns de Desenvolupament

Introducció als programes informàtics

Fases de desenvolupament: Mètrica 3.0 i Gantt

Fases de desenvolupament: SCRUM

Control de versions

Documentació

Proves unitàries

Refacció

UML

institut sa palomera

Entorns de Desenvolupament – UF3

Introducció als programes informàtics

Fases de desenvolupament: Mètrica 3.0 i Gantt

Fases de desenvolupament: SCRUM

Control de versions

Documentació

Proves unitàries

Refacció

UML

institut sa palomera

Entorns de Desenvolupament

Introducció als programes informàtics

Fases de desenvolupament: Mètrica 3.0 i Gantt

Fases de desenvolupament: SCRUM

Control de versions

Documentació

Proves unitàries

Refacció

UML

institut sa palomera

Què és enginyeria del software?

institut sa palomera

Un *sistema de software*, també anomenat *aplicació*, *programari* o simplement *software*, és un conjunt integrat de programes, bases de dades, documentació, etc i que permeten un *sistema informàtic* realitzar les tasques per les quals ha estat dissenyat.

Què és desitja d'un software?

1. Fàcil manteniment, ben dissenyat i ben documentat.
2. Fiable i robust
3. Eficient >>> no ha de malgastar recursos del sistema.
4. Bona interfície >> Cal tenir presents els usuaris que l'utilitzaran

Què és enginyeria del software?

institut sa palomera

L'Enginyeria del Software és una disciplina que integra **processos**, **mètodes** i **eines** amb l'objectiu de desenvolupar i mantenir sistemes de software que siguin econòmics, fiables i eficients.

· Els **processos** defineixen un marc de treball que permeten un desenvolupament racional de la enginyeria del software. Formen la base del control de gestió dels projectes de software i estableixen el context en el qual s'apliquen els mètodes i es produeixen els resultats del treball: models, documents, dades, informes, formularis, etc.

· Els **mètodes** indiquem com construir el software. Ofereixen tècniques de modelatge per a les diferents etapes del procés: definició de requeriments, anàlisi, disseny, implementació, proves, explotació i manteniment.

· Les **eines** proporcionen suport automàtic o semiautomàtic per al procés i els mètodes.

Les eines CASE (Computer Assisted System Enginery) ajuden a l'automatització de tot el procés de construcció del software, ajudant a l'obtenció de resultats d'alta qualitat.

L'enginyeria del software és una activitat de *modelat* per a la solució dels problemes plantejats pel client del sistema.

Per a més informació veure document del moodle

Llenguatge de modelització unificat

institut sa palomera

- Llenguatge de modelat de sistemes de programari.
- Llenguatge gràfic per visualitzar, especificar, construir i documentar un sistema.
- Ens ofereix visualitzar els problemes mitjançant diagrames.
- Està enfocat a la resolució/representació de problemes destinats a la construcció de software.

StarUML™ The Open Source UML/MDA Platform

ENTERPRISE ARCHITECT Think it Through

Llenguatge de modelització unificat

institut sa palomera

- UML permet descriure un model d'anàlisi i disseny d'un sistema mitjançant diagrames construïts utilitzant símbols que tenen regles semàntiques, sintàctiques i pràctiques.
- Les regles semàntiques ens diuen que significa cada símbol i com interpretar-lo, tant quan es troba aïllat com quan es combina amb altres símbols en un diagrama.
- La sintaxis ens diu com mostrar i combinar els símbols per a obtenir els diagrames del model.
- Les regles pràctiques defineixen com utilitzar els símbols per a obtenir els diagrames del model de manera que siguin comprensible per altres persones.

Quins diagrames fa servir UML?

institut sa palomera

- Diagrames de Casos d'Us
- Diagrames de Classe
- Diagrames d'Objectes
- Diagrames d'Estats
- Diagrames d'Activitat
- Diagrames de Seqüència
- Diagrames de Col·laboració
- Diagrames de Components
- Diagrames de Desplegament

Factors a tenir en compte en un UML

institut sa palomera

- Anàlisi
- Disseny
- Disseny d'interfícies d'usuari
- Disseny de la persistència
- Patrons de disseny

Per a més informació veure document del moodle

Vistes

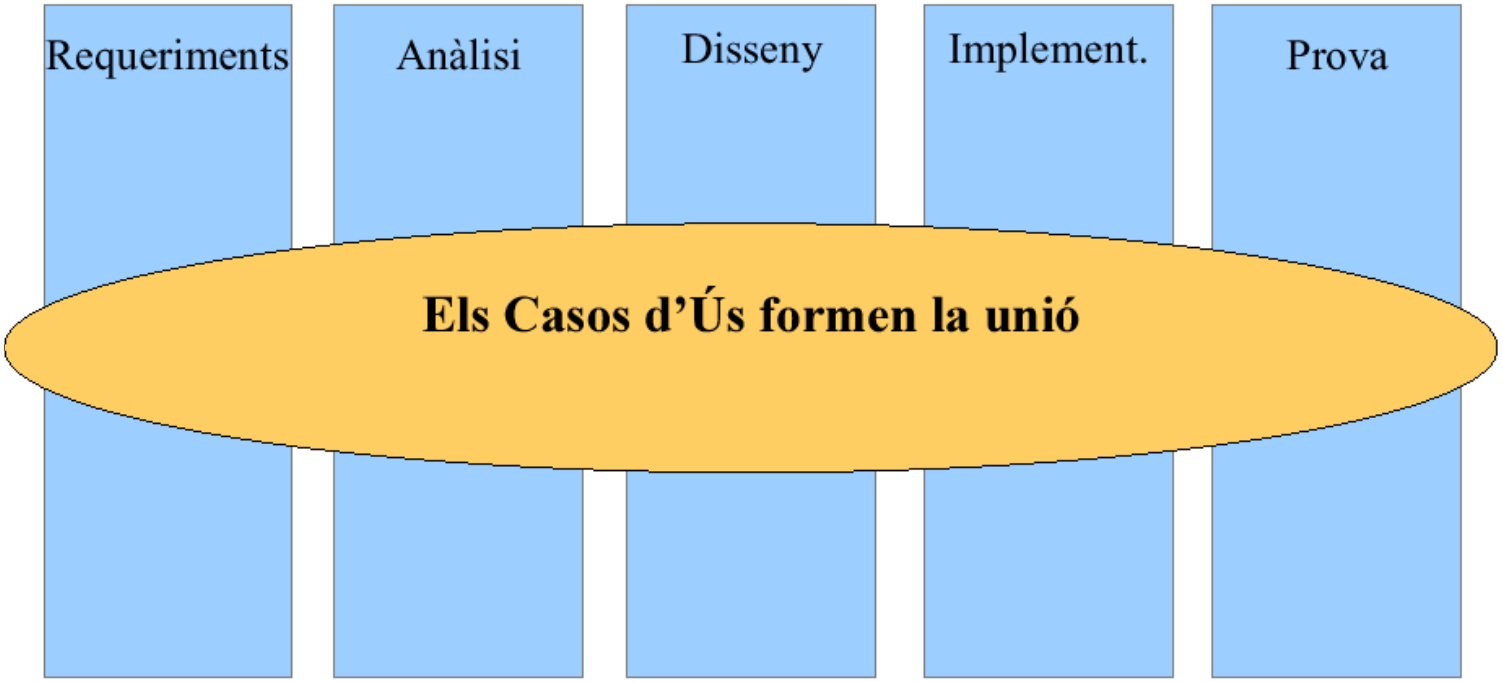
- Vista de casos d'ús
  - Punt de vista dels actors externs
- Vista lògica
  - Requeriments funcionals pel que fa a classes i objectes
- Vista de processos
  - Processos de sincronització i concurrència del sistema
- Vista d'implementació
  - Repartiment de classes en components i subsistemes
- Vista de desplegament
  - Recursos hardware, desplegament i implementació del software en aquests recursos

Casos d'ús

institut sa palomera

- Descriu el comportament (funcionalitats) d'un sistema quan interactua amb un usuari extern (actor).
- Representa un conjunt d'interaccions entre un o més actors i el sistema.
- Defineixen el comportament d'un sistema des del punt de vista dels actors.
- Representen els requeriments funcionals d'un sistema;
- Descriuen què fa un sistema, no com ho fa;
- Dirigeixen tot el procés de desenvolupament d'un sistema.

Casos d'ús



Capturar, clarificar i validar els casos d'ús

Realitzar els casos d'ús

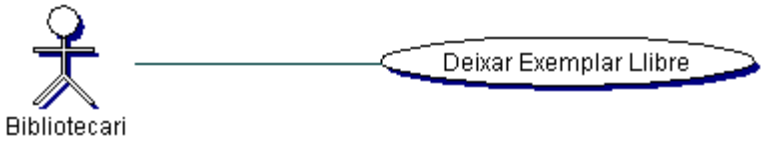
Verificar si se satisfan els casos d'ús

Diagrames de Casos d'ús

institut sa palomera

A un diagrama de casos d'ús:

- els actors es representen per figures;
- els casos d'ús es representen amb òvals;
- les comunicacions es representen per línies que uneixen actors i casos d'ús



La interrelació pot comportar diversos intercanvis d'informació en les dues direccions. Per tant no implica cap flux d'informació només en un determinat sentit.

Pels esdeveniments que passen regularment (pagament d'un salari), s'acostuma a fer servir l'actor Temps.

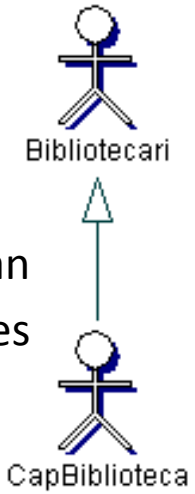
Diagrames de Casos d'ús

institut sa palomera

Característiques addicionals que fan que la informació es reculli millor:

- fronteres (dibuix d'un rectangle que delimita les accions dels actors).
- especialització i generalització d'actors (els actors descendent hereten els rols i comunicacions de l'actor pare).

L'actor *CapBiblioteca* pot fer totes les funcions que fan els actors *Bibliotecari* més altres de pròpies



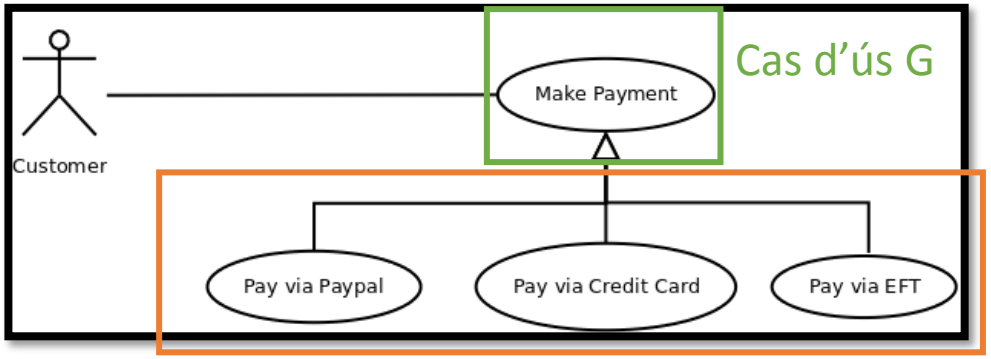
- generalització de casos d'ús
- Includes i extensions

Diagrames de Casos d'ús

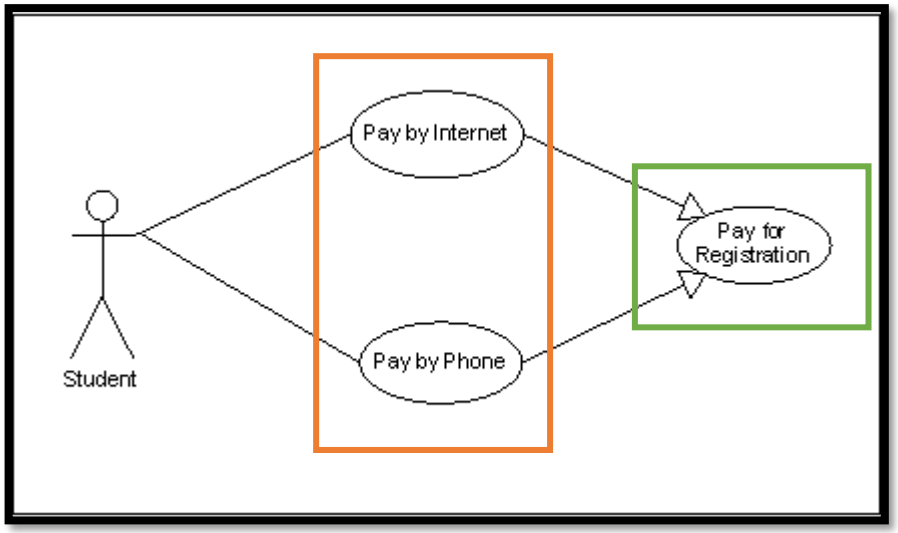
institut sa palomera

Generalització.

Una *generalització* d'un cas d'ús mostra que un cas d'ús E és un tipus especial d'un altre cas d'ús G. El cas d'ús E fa tots els processos del cas d'ús G més algun procés específic. Una generalització es simbolitza per mitjà d'una fletxa que apunta cap el cas d'ús G etiquetada amb **<<generalize>>.**



Casos d'ús E

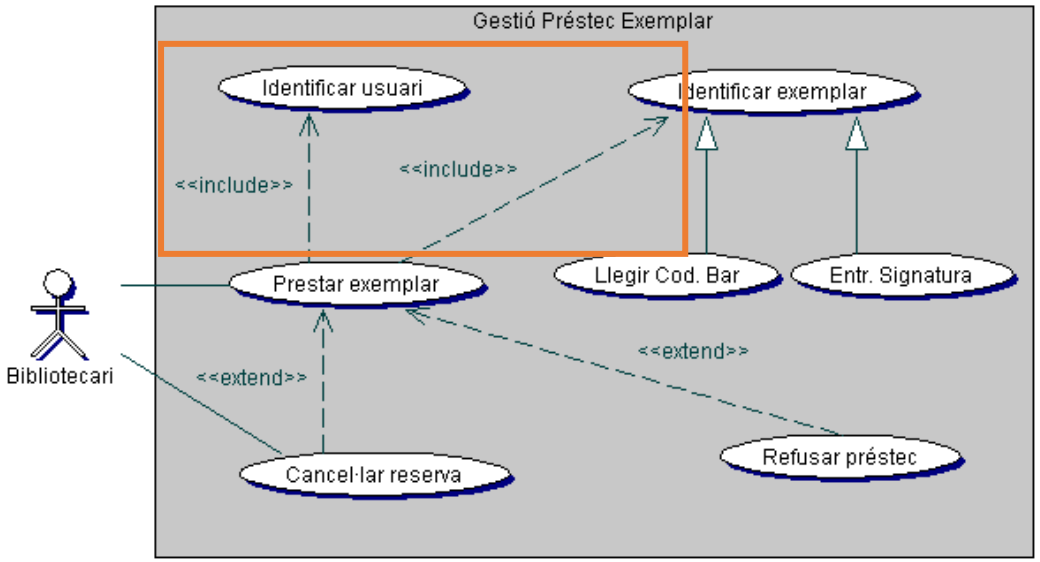


Diagrames de Casos d'ús

institut sa palomera

Inclusió.

Un cas d'ús pot incorporar explícitament el comportament d'altres casos d'ús com a fragments del seu propi comportament. Aquesta relació s'anomena *d'inclusió*. El nou cas d'ús és activat pels casos d'ús que l'inclouen. Per tant, no és un cas especial del cas d'ús original. Es fa servir una inclusió quan se sap exactament quan invocar un cas d'ús. En el diagrama es representa per una fletxa puntejada amb una etiqueta **<<include>>**.



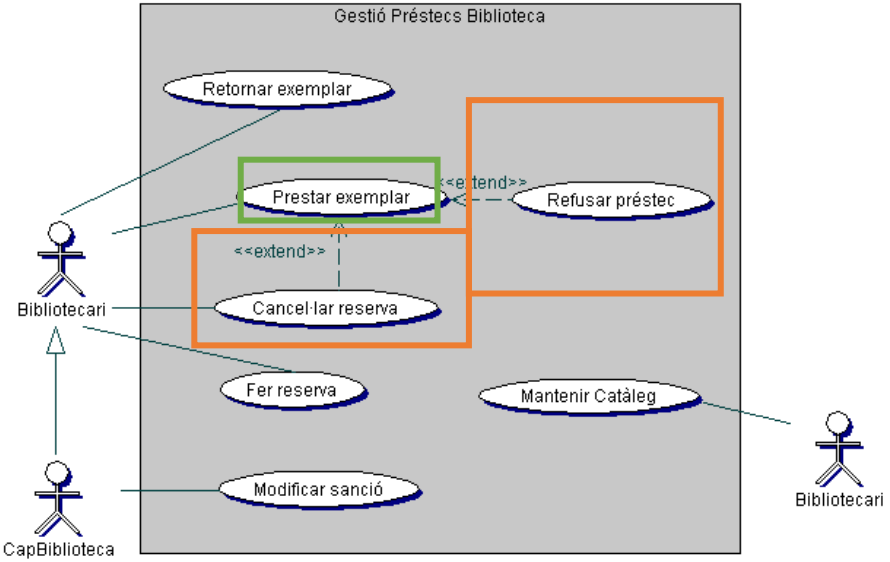
Diagrames de Casos d'ús

institut sa palomera

Extensió.

Un cas d'ús E es pot definir com una extensió opcional d'un cas d'ús base B.

Dins de B s'executa E quan es compleix una condició determinada. Aquesta relació s'anomena *d'extensió*. I pot haver vàries extensions d'un mateix cas d'ús. En el diagrama es representa per una fletxa puntejada en direcció cap el cas d'ús B, amb una etiqueta **<<extend>>**. El cas d'ús E no acostuma a ser un cas d'ús complet activable per un determinat actor.



Diagrames de Casos d'ús

institut sa palomera

Escenaris

Un *escenari* és un recorregut específic d'un cas d'ús. Cada cas d'ús té un escenari *principal*. A l'escenari principal d'un cas d'ús se suposa que tot funciona idealment: no hi ha ni errors, ni alternatives, ni interrupcions.

Els casos d'ús complexos poden tenir diversos escenaris *secundaris* que representen alternatives a l'escenari principal (errors, alternatives i interrupcions).

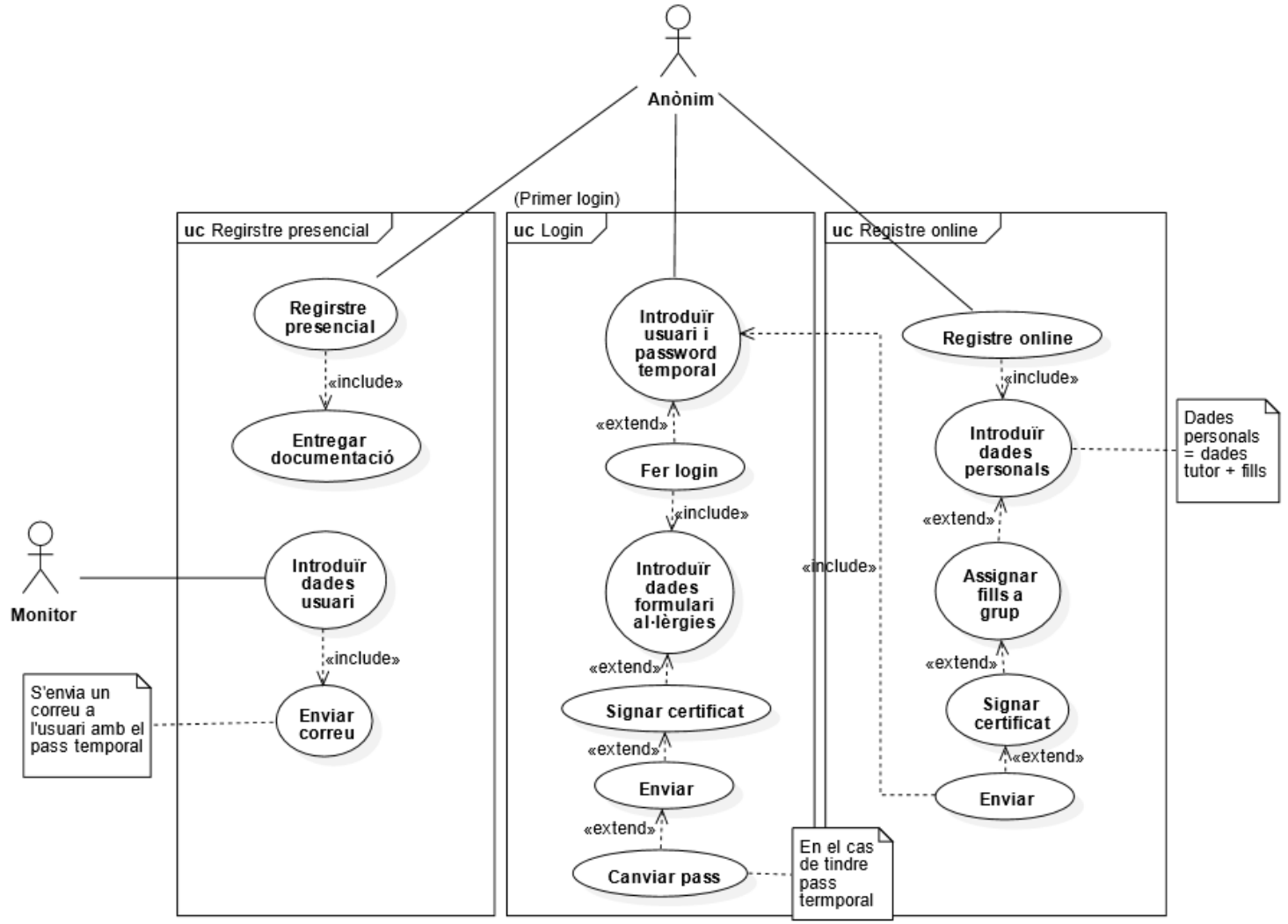
L'escenari principal i els escenaris secundaris es poden representar en un mateix diagrama de casos d'ús. No obstant, l'escenari principal i els escenaris secundaris s'especifiquen separadament.

Veiem un exemple.....

Diagrames de Casos d'ús

institut sa palomera

Login



Diagrames de Casos d'ús

institut sa palomera

Passos per a la creació del conjunt dels casos d'ús

1. Identificar tots els actors del sistema.
2. Per a cada actor trobar totes les formes d'interactuar amb el sistema.
3. Crear un cas d'ús per a cada forma d'interactuar (opcional).
4. Estructurar els casos d'ús.
5. Revisar i validar els casos d'ús amb l'usuari.

Diagrames de Casos d'ús

institut sa palomera

Avantatges de la utilització dels diagrames de casos d'ús

• Proporcionen un llenguatge de comunicació entre usuaris i desenvolupadors.

• Faciliten la determinació de requeriments i la comprensió detallada de les funcionalitats del sistema.

• Ajuden a gestionar el risc dels projectes (complexos).

• Faciliten l'estimació del temps, dels recursos i de les prioritats dels projectes.

• Faciliten la verificació de la traducció de requeriments a codi executable.

• Permeten un major control del manteniment de les successives revisions dels programes.

• Ajuden a la generació de documentació d'usuari.

• Permeten la generació de casos de prova per a diversos escenaris.

Exemple

institut sa palomera

Gestió d'una biblioteca

Guió d'un préstec

Quan un usuari vulgui retirar un llibre en préstec el bibliotecari:

- identificarà a l'usuari;
- verificarà que l'usuari no està sancionat;
- comprovarà que l'usuari no té ja en préstec el màxim de llibres autoritzats;
- identificarà el llibre;
- verificarà que un exemplar del llibre es pot deixar durant el període sol·licitat per l'usuari;
- si l'usuari té el llibre reservat, en cancel·larà la reserva;
- farà el préstec.

Exemple

institut sa palomera

Glossari

Usuari: pot ser un professor o un estudiant.

Bibliotecari: s'encarrega de mantenir el catàleg de llibres, de fer préstecs i reserves.

Cap biblioteca: fa totes les tasques d'un bibliotecari, excepte catalogar llibres. És l'únic amb la capacitat de modificar sancions.

Llibre: s'identifica amb la signatura.

Exemplar llibre: s'identifica amb signatura i número de còpia.

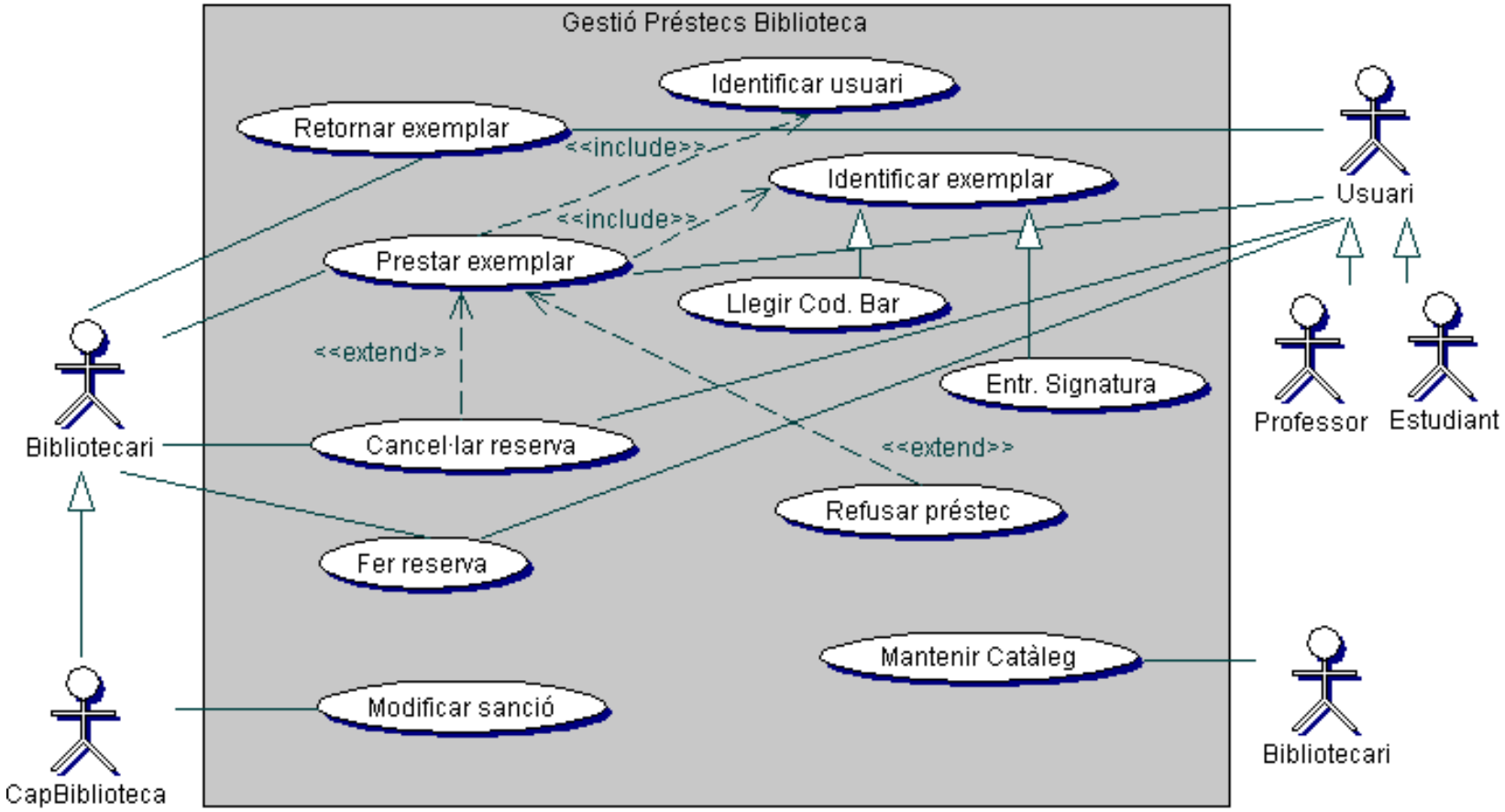
Catàleg: conjunt de tots els llibres de la biblioteca.

Préstec: es deixa un exemplar d'un llibre a un usuari per a un període de temps determinat.

Exemple

institut sa palomera

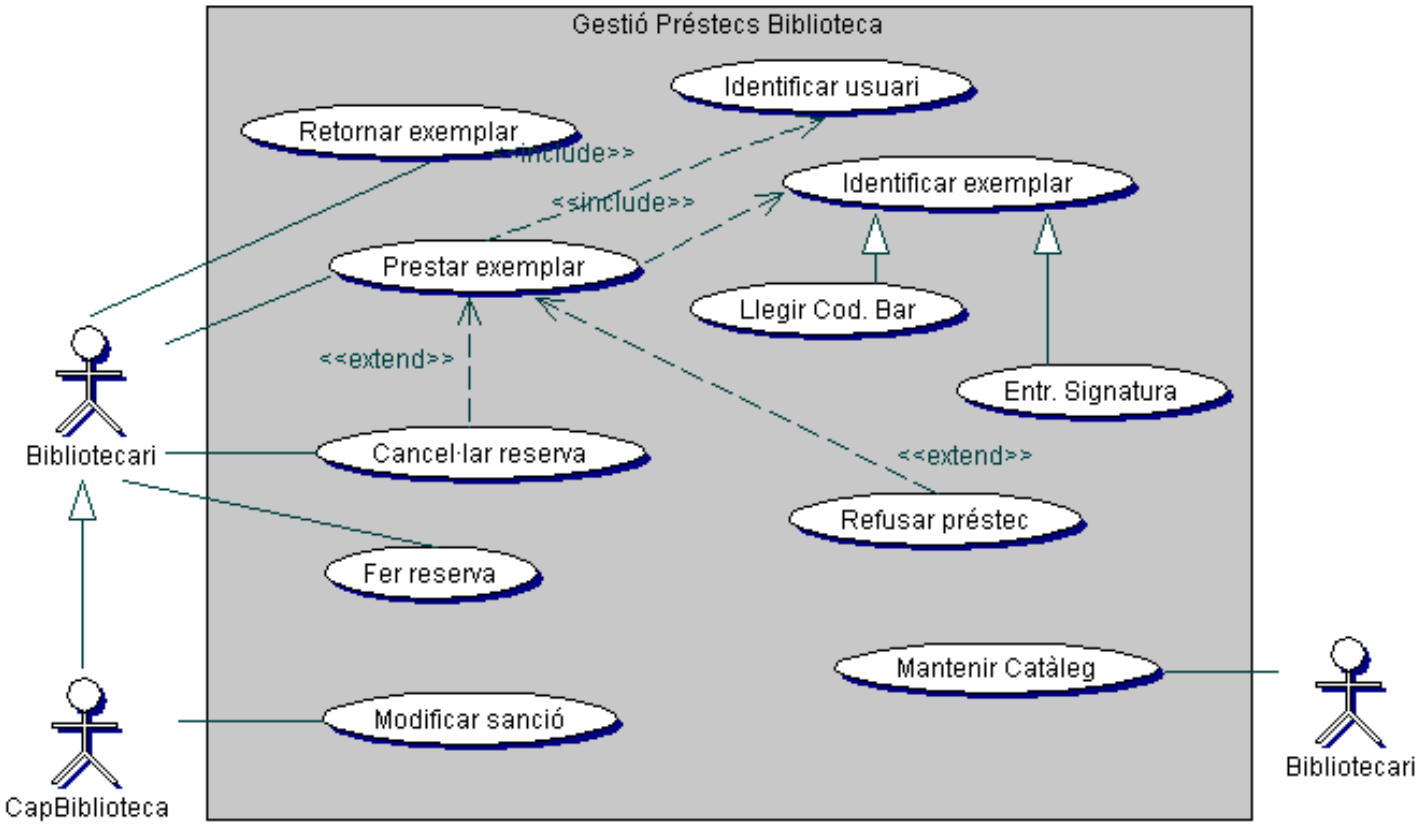
Reserva d'un llibre a nom d'un usuari.



Exemple

institut sa palomera

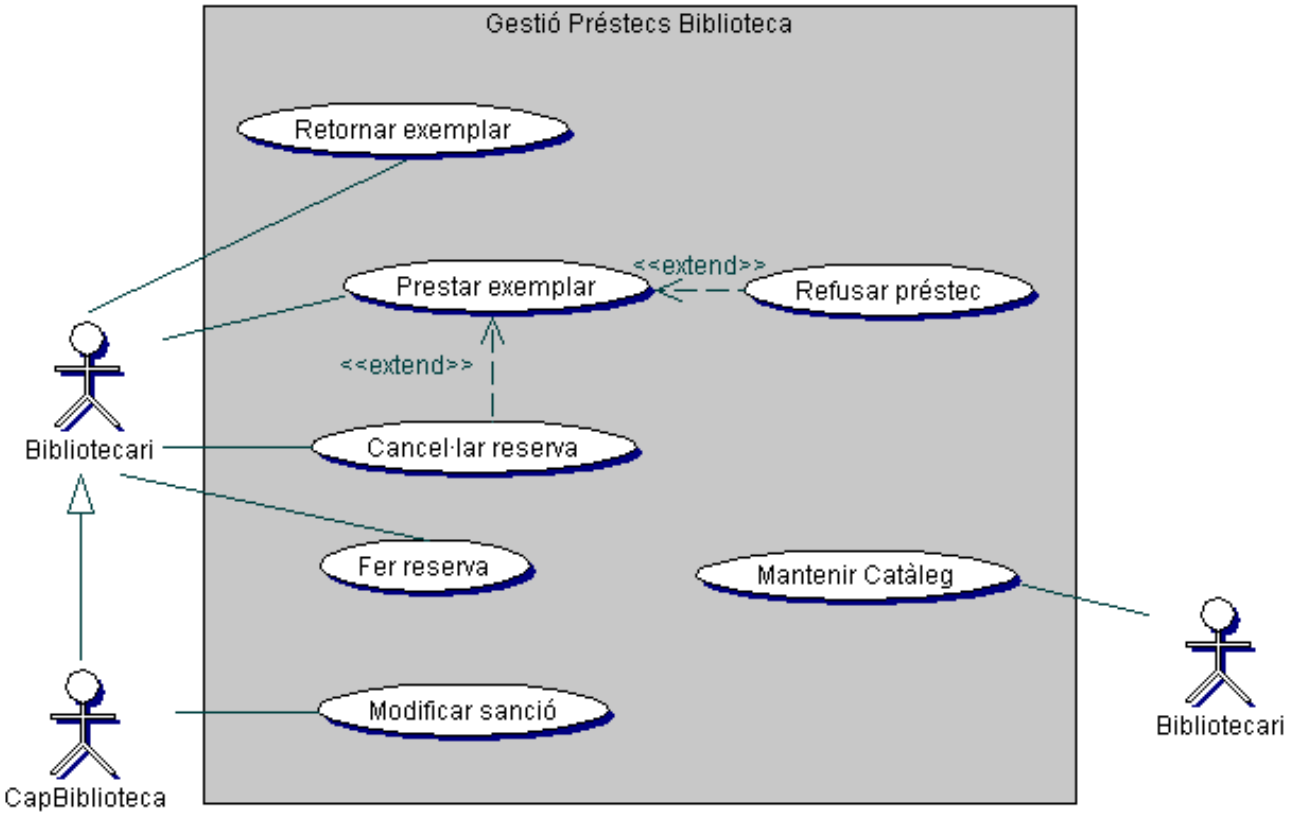
Observis que l'Usuari interactua amb el sistema per mitjà de l'actor Bibliotecari (actor facilitador). Per tant no és pròpiament un actor, i no hi ha acord sobre si és millor posar-lo no a un diagrama. El criteri que farem servir a partir d'ara, en general, serà de no posar-lo, tal com fem a l'exemple modificat que presentem a continuació:



Exemple

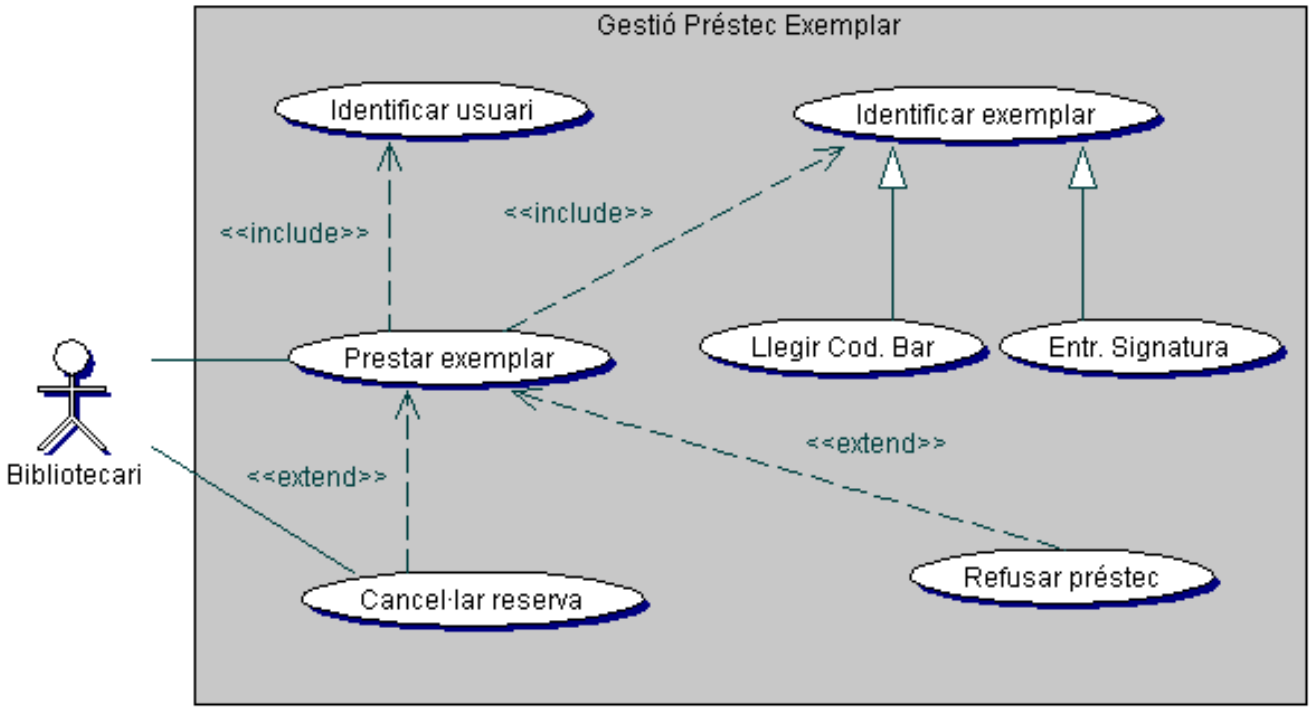
institut sa palomera

A l'anterior diagrama hi podem observar casos d'ús de diferent nivell.
Generalment dibuixarem un diagrama de casos d'ús de context que complementarem amb una fitxa per a cada cas d'us.
Diagrama de casos d'ús de context per l'exemple de gestió de la biblioteca:



Exemple

refinem el diagrama...



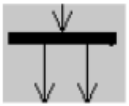
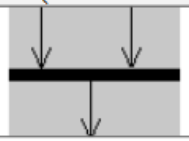
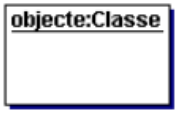
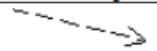
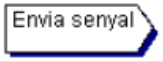
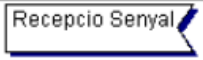
Diagrames d'Activitats

institut sa palomera

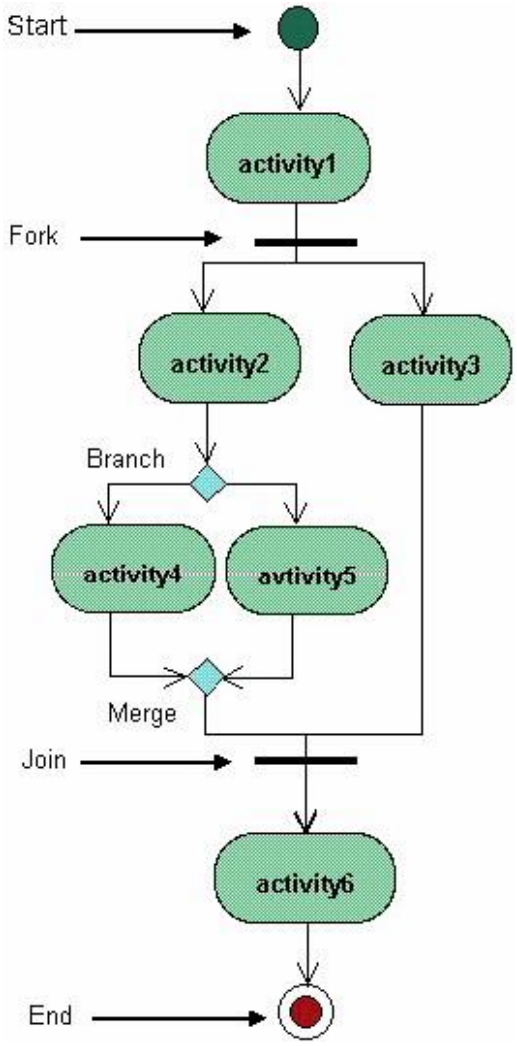
i. En un diagrama d'activitat, el procés comença a partir del cercle negra d'inici situat a la part superior o esquerra del diagrama i acaba al cercle blanc/negre de final situat a la part inferior o dreta del diagrama.
ii. Les activitats s'indiquen amb rectangles arrodonits.
iii. Els diagrames d'activitat es poden subdividir en *carrers* (swimlanes) per a mostrar el responsable (actor, objecte, unitat organitzacional, cas d´ús, ...) encarregat de l'activitat.
iv. De cada activitat se'n deriva una transició que connecta amb la següent activitat.
v. L'inici i final de branca s'indiquen amb un rombe.
vi. Una transició pot derivar cap a vàries noves transicions (branques) mútuament excloents. Les expressions que controlen quina ha de ser la nova transició es posen dins de [ ]. L'inici i final de branca s'indiquen amb un rombe.
vii. Una transició pot derivar cap a vàries activitats que es desenvolupen en paral·lel.
viii. L'inici d'activitats en paral·lel i la unió de retrobament final d'aquestes activitats (sincronització) es simbolitzen amb barres sòlides.

Components

institut sa palomera

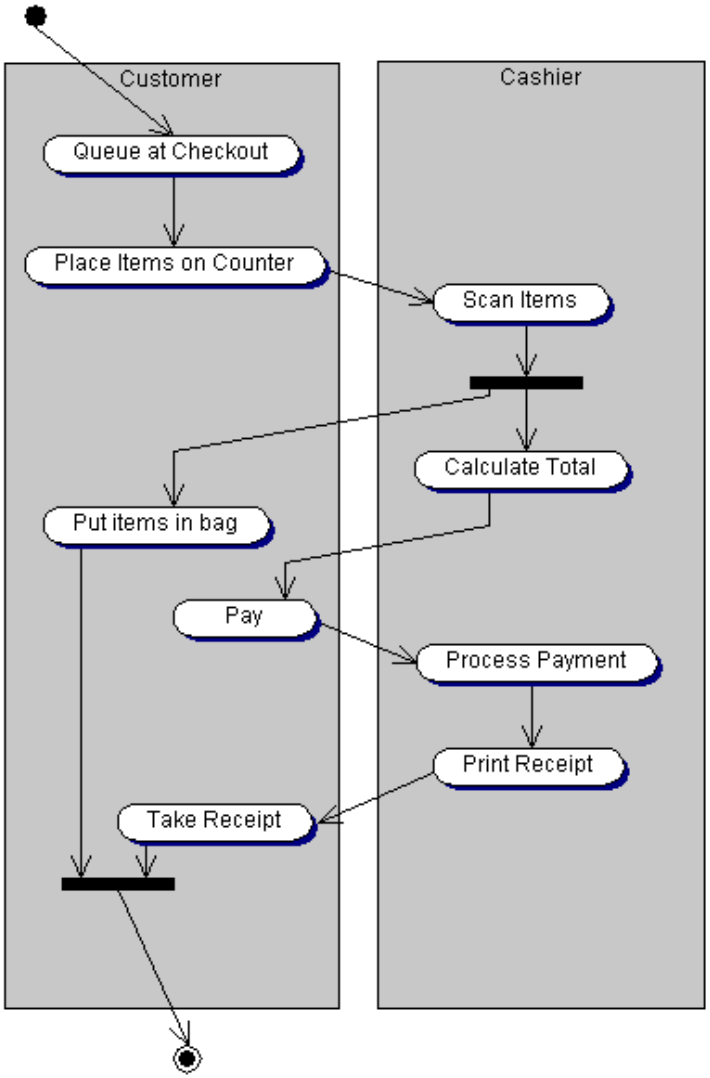
Inici	Fi	Activitat
		
Punt d'inici del diagrama	Punt de fi del diagrama	Representa un procés desenvolupat pel sistema.
Bifurcació/trobada	**Fork (concurrència)**	**Join (concurrència)**
		
Bifurcació del flux. Representa una presa de decisió. Permet també unir els fluxos divergents posteriorment.	El fork permet llançar diverses línies d'execució **simultànies**.	Uneix fluxos concurrents. Fins que tots els fluxos no han arribat a aquest punt, l'execució no continua.
Objecte	**Línia de seqüència**	**Linia d'Objecte**
		
És un objecte (informació) que es genera o consumeix en una activitat.	Indica quina és la següent activitat/bifurcació/fork...	Uneix una activitat a un objecte
Swimlane o Carril	**Enviar senyal**	**Rebre senyal**
		
Permet agrupar visualment totes les activitats realitzades per un mateix actor.	Envia una senyal degut a l'ocurrència d'un event.	Activa una línia d'execució al rebre un senyal o estímul extern. Quan el fluxe arriba a un receptor de senyal , ens quedem en espera fins a rebre la senyal esperada.

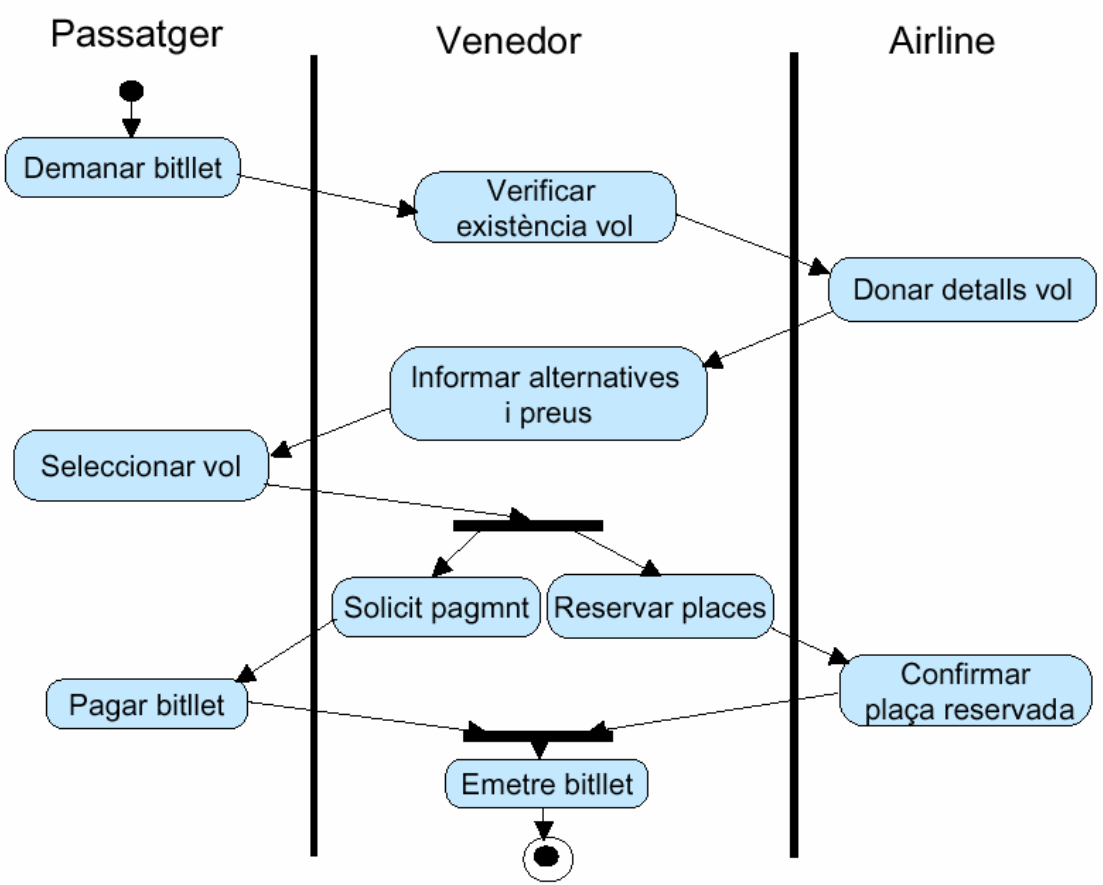
Exemple


Start
activity1
Fork
activity2
activity3
Branch
activity4
avtivity5
Merge
Join
activity6
End

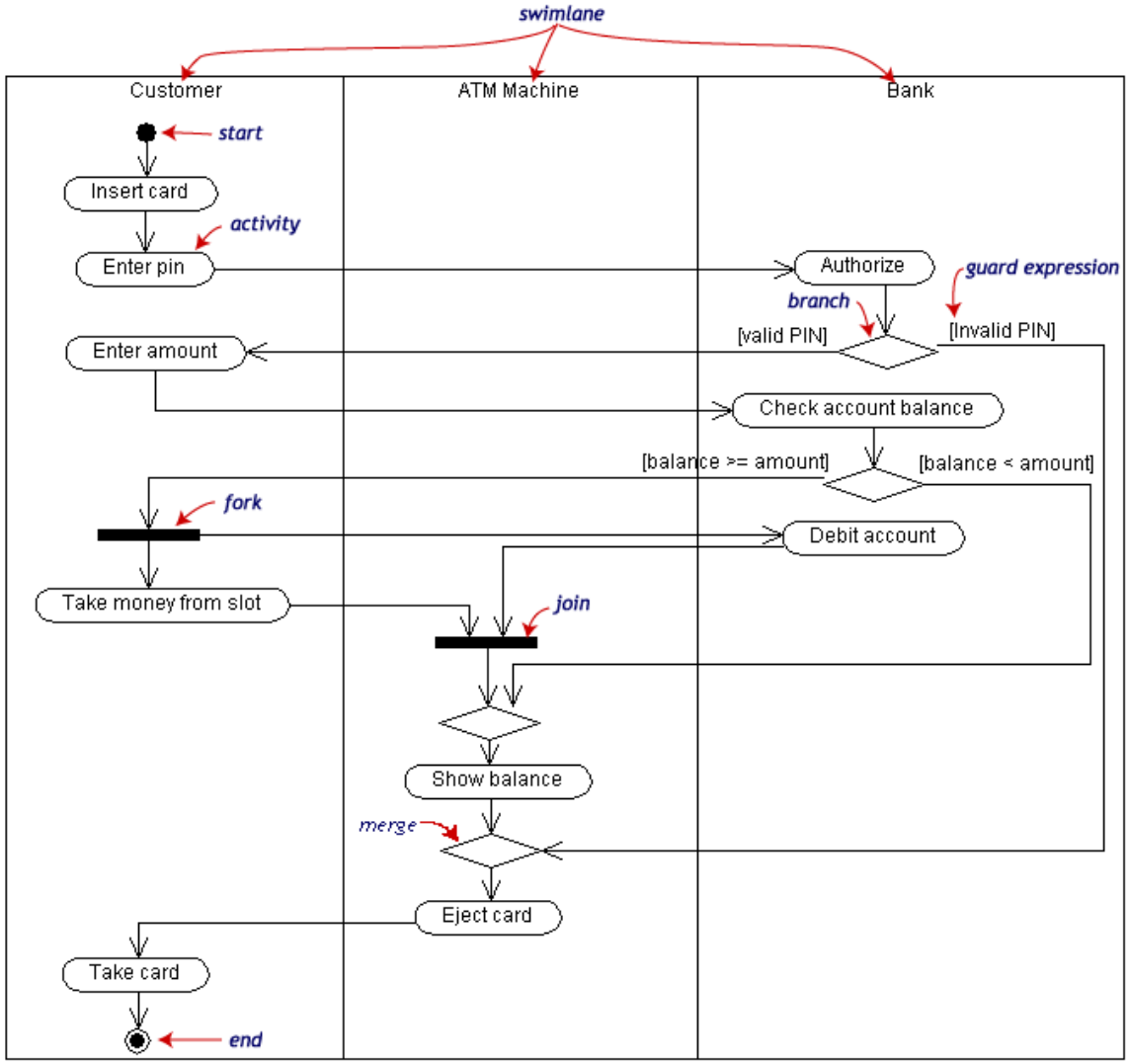
Exemple

institut sa palomera

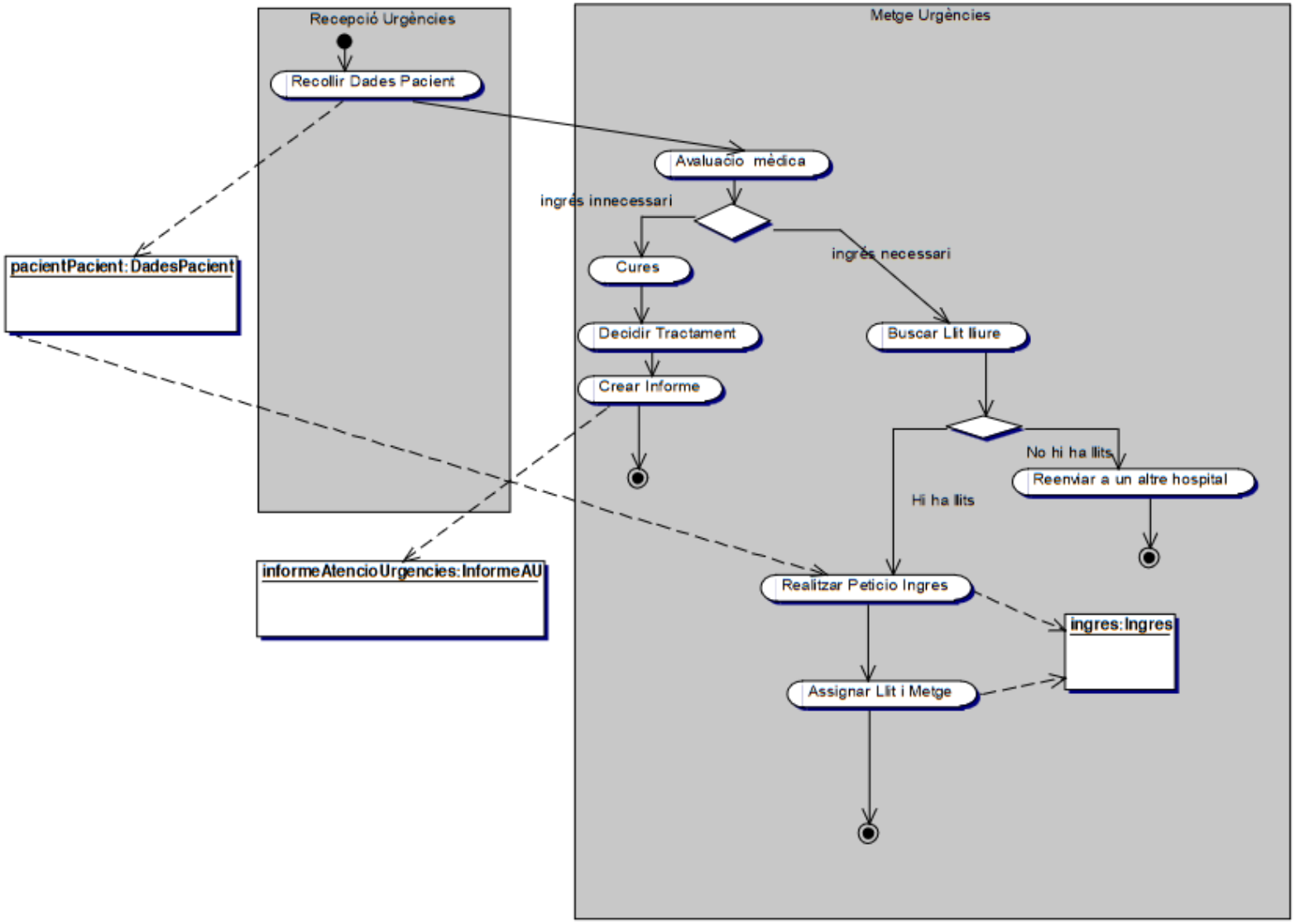




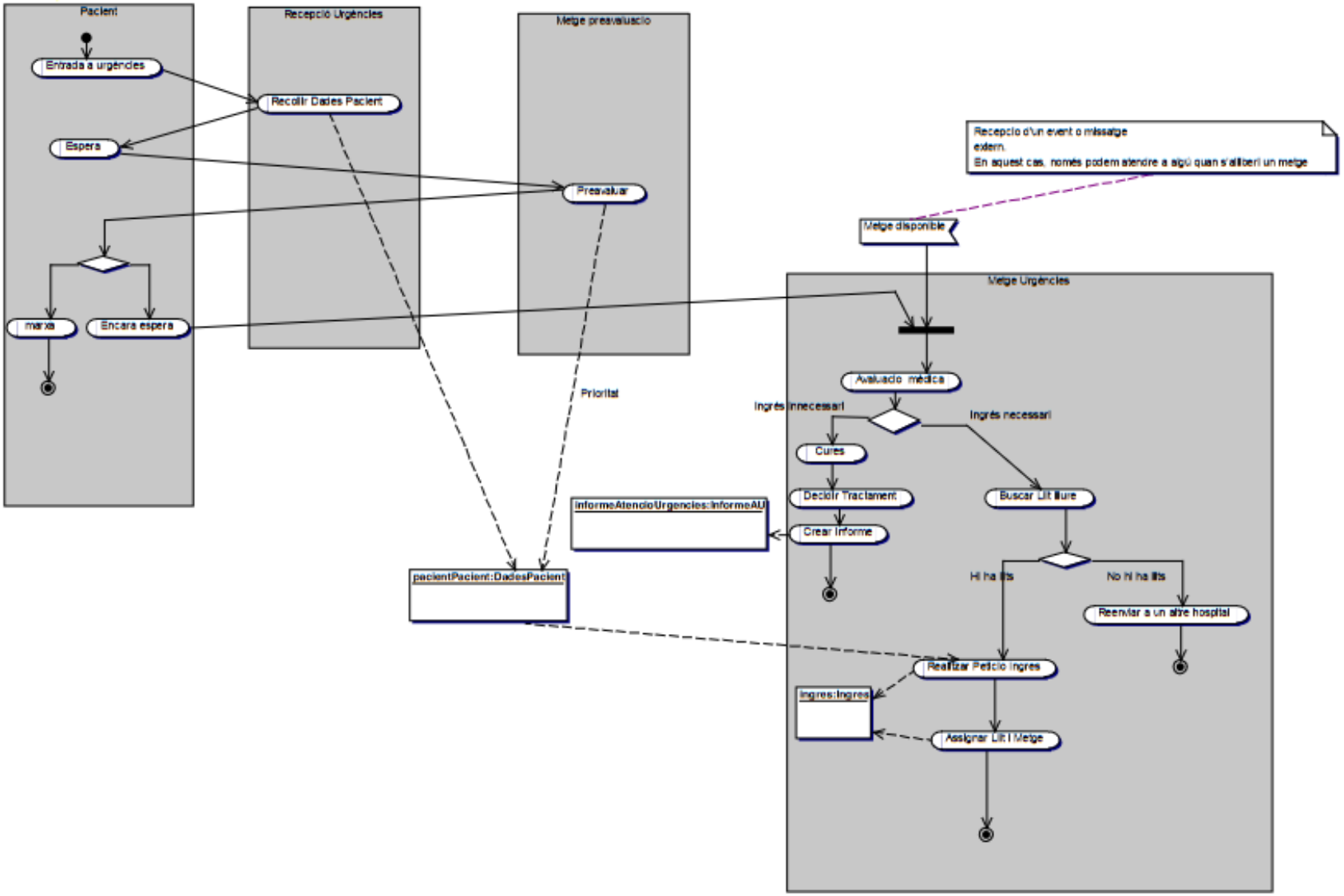
Exemple


swimlane
Customer
ATM Machine
Bank
start
Insert card
activity
Enter pin
Authorize
guard expression
branch
[valid PIN]
[Invalid PIN]
Enter amount
Check account balance
[balance >= amount]
[balance < amount]
fork
Debit account
Take money from slot
join
Show balance
merge
Eject card
Take card
end

Exemple (Nivell de detall 1)

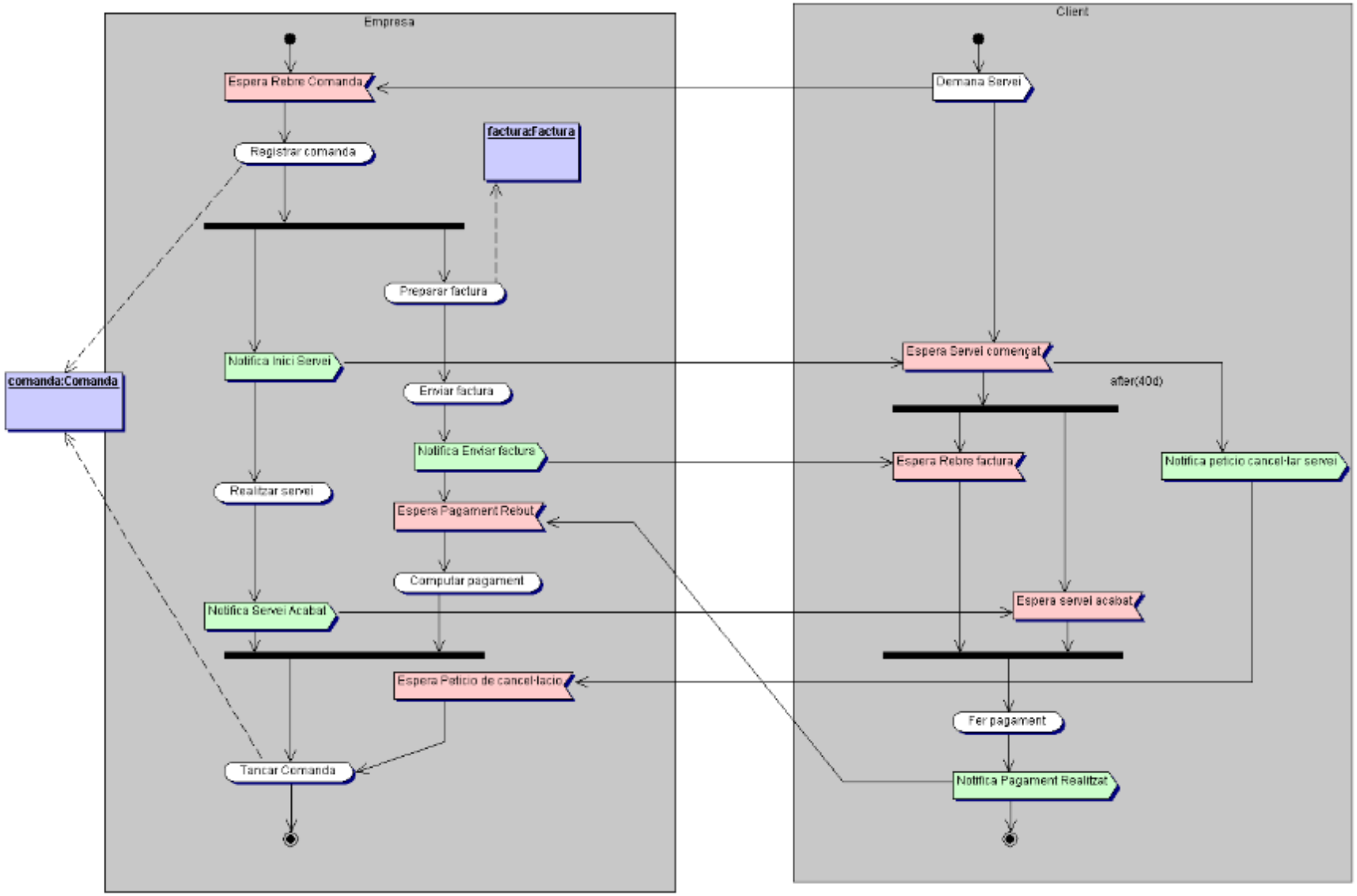


Exemple (Nivell de detall 2)



Exemple (Gestió de concurrència)

institut sa palomera



Què és l'orientació a objectes? (I)

Amb l'orientació a objectes, la informàtica sempre ha intentat que el pensament com a programador sigui el més semblant possible al pensament sobre el món real. Per tant si ens preguntem que és un objecte en el món informàtic, cal preguntar-se què és un objecte al món real.

Què és l'orientació a objectes? (II)

institut sa palomera

Aquesta poma és un objecte a la vida real? Segur. Aquest escriptori? Bé, és clar, absolutament!
Tots són objectes, tot això són coses.

Entenem que els objectes estan separats els uns dels altres. Tenen la seva pròpia existència, la seva pròpia identitat que és independent d'altres objectes.

Què és l'orientació a objectes? (III)

institut sa palomera

Aquí en tenim dos tasses, són la mateixa tassa? NO!!!!
Són objectes diferents, tenen la seva pròpia identitat.

Sabem que ser objecte no té res a veure amb la complexitat.
Una poma és un objecte, però també ho és un portaavions, un telèfon, i sabem que un objecte pot contenir altres objectes.

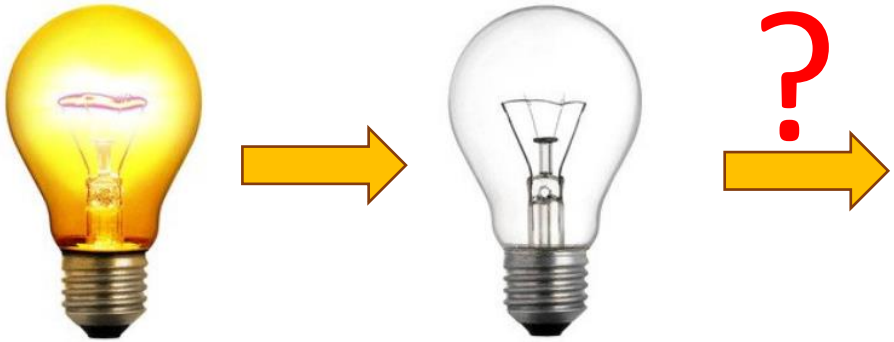
Què és l'orientació a objectes? (IV)

institut sa palomera

Sabem que els objectes tenen característiques i propietats inherents que els descriuen.

són els atributs de qualsevol objecte → el color, el pes, la mida...

Apagar un llum, apaga tots els llums del món????



Descriuen l'estat actual d'un objecte i l'estat d'un objecte és independent de la resta.

Què és l'orientació a objectes? (V)

institut sa palomera

La majoria dels objectes tenen **múltiples atributs.**

La tassa pot estar:
- Plena
- Mig plena
- Mig buida
- Buida

i al mateix temps, la tassa pot ser:
- Verda
- Blanca
- ...

I al món real tenen un **comportament.**

- Un telèfon pot sonar
- Un avió pot volar
- Una poma no sona
- Un telèfon no vola

Aquests 3 conceptes **IDENTITAT, ATRIBUTS** i **COMPORTAMENTS** són les mateixes 3 coses que descriuen un objecte en un llenguatge de programació orientat a objectes

- **Identitat:** l'objecte és autònom, per tant te una identitat separada de la resta d'objectes.
- **Atributs:** l'objecte te els seus propis i descriuen el seu estat actual
- **Comportament:** allò que l'objecte pot fer

Què és l'orientació a objectes? (VI)

institut sa palomera

A la informàtica anem més enllà i també considerem els objectes no tangibles com a objectes amb la seva identitat, atributs i comportaments, per exemple:

Un compte bancari...

- ✓ te identitat ja que és diferent a altres comptes bancaris
- ✓ te atributs o dades que descriuen el seu estat actual
- ✓ te un comportament, podem ingressar, retirar-ne, obrir, tancar, ...

Com esbrinem si alguna cosa de la nostra aplicació és un objecte potencial nou?

P.ex, la nostra aplicació és per a gestió d'esdeveniments... Seria un objecte?

Una forma de saber-ho és fent servir substantiu i determinant:

La tassa, la poma, el cotxe, el compte bancari, l'hora, l'esdeveniment, ...

Però mai diríem:

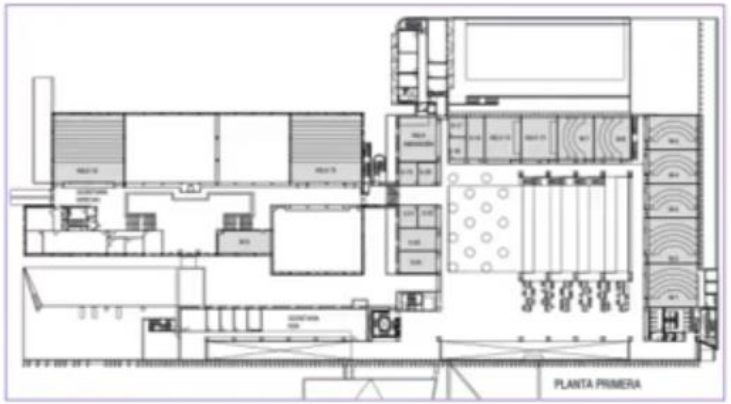
L'estalviar, el imprimir, l'explotar, ...

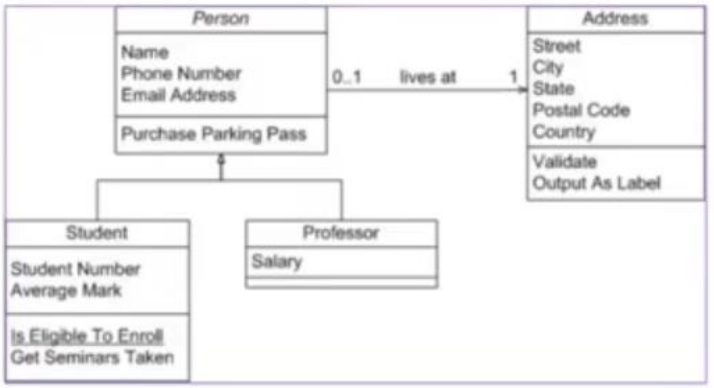
I aquests objectes que podem crear al nostre programa, d'on provenen? De la Classe!!!

Què és una Classe? (I)

institut sa palomera

Objectes i classes van de la mà. No podem parlar d'un sense parlar de l'altre. I tot l'objectiu del disseny orientat a objectes no es tracta d'objectes, sinó de classes, perquè fem servir classes per crear objectes. Així, una classe descriu què serà un objecte, però no és l'objecte en si.
Per exemple:





Què és una Classe? (II)

institut sa palomera

Una Classe és un objecte que **no conté** una identitat pròpia, però si te uns atributs i un comportament.

La identitat pròpia apareix quan de la classe creem un objecte i per tant aquest objecte tindrà **una identitat pròpia**, uns atributs i un comportament.

En termes informàtics vindria a ser:

Nom de la classe faria referència a Tipus
Atributs faria referència a Propietats
Comportament faria referència a Operacions

Què és una Classe? (III)

institut sa palomera

Ens preguntem:
Quines operacions pot fer una Classe?
És a dir, quin és el comportament d'una classe?
Escriure aquest comportament en codi vindria a ser crear mètodes

4 conceptes més...

institut sa palomera

Abstraction

Polymorphism

Inheritance

Encapsulation

Tots 4 els haurem de tenir en compte a l'hora de crear una classe

Què és Abstracció? (I)

institut sa palomera

Si dic Taula, sabeu a que em refereixo?
Cadascú segur que pensa en una taula diferent, però la idea de taula la entenem, l'abstracció de taula la tenim a la ment ...

Dit això, Abstracció vindria a ser que ens centrem en les qualitats essencials d'una cosa i per tant descartem el que no té importància o és irrellevant.

Què és Abstracció? (II)

institut sa palomera

- Abstracció significa que podem tenir una idea o un concepte completament separat de qualsevol instància específica.
- Abstracció és al cor de la programació orientada a objectes perquè és el que estem fent quan fem una classe.
- Abstracció significa que no creem una classe per al compte bancari d'en Pere Pi i una classe separada per al compte bancari de la Marta Mas. Ens centrarem en les qualitats essencials de la idea, i escriurem una classe de compte bancari.

Què és Encapsulament? (I)

institut sa palomera

Penseu en una càpsula espacial, una càpsula de medicament o un recipient d'aliments...

És la idea d'envoltar alguna cosa per a mantenir els continguts junts i protegir-los.

Què és Encapsulament? (II)

institut sa palomera

Restringir l'accés al funcionament intern d'aquesta classe o a qualsevol objecte basat en aquesta classe, és a dir, ocultació d'informació o ocultació de dades.

El principi és que un objecte no ha de revelar res sobre si mateix, excepte el que és absolutament necessari perquè altres parts de l'aplicació funcionin

Què és Encapsulament? (III)

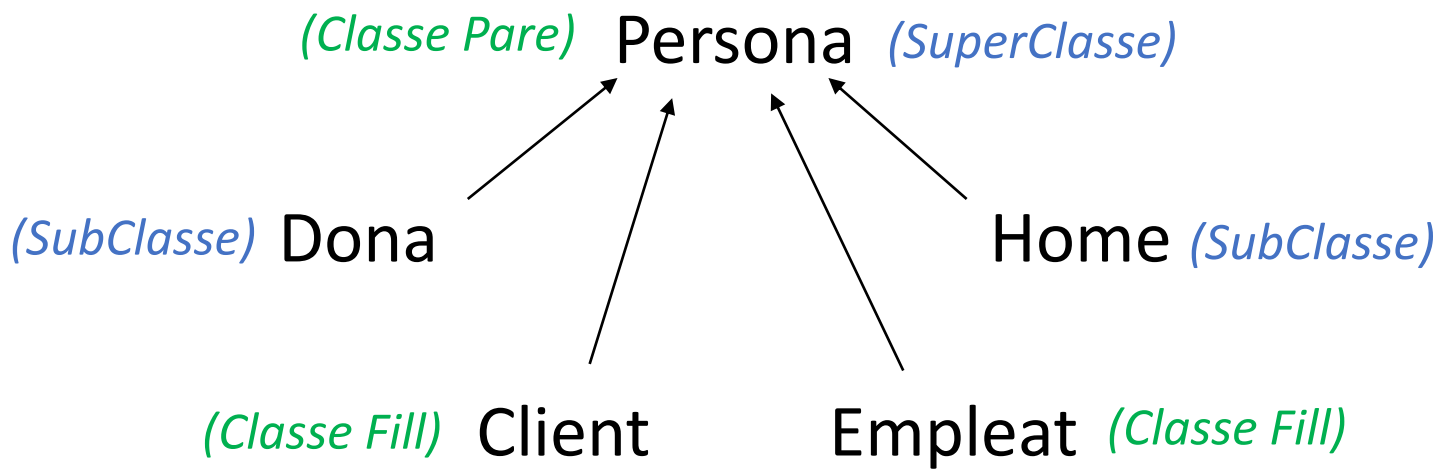
institut sa palomera

Exemple:
Penseu en el funcionament d'un telèfon.
Volem que la interacció sigui el més senzilla possible, per tant fem servir el teclat per fer una trucada,
ens importaria si canviés el funcionament intern del telèfon si continuéssim amb el mateix teclat?

La idea d'encapsulació és que tanqueu, encapsuleu els atributs i mètodes del vostre objecte i, a continuació, amagueu tot sobre aquest objecte excepte el que és absolutament necessari exposar.

Què és Herència?

Forma de reutilització de codi. Podem crear una nova classe, però en lloc d'escriure-la des de zero, podem basar-la en una classe existent.



L'herència ens permet:

- Estalviar temps ja que podem reutilitzar codi
- Utilitzar l'últim dels nostres quatre termes clau, polimorfisme

Què és Polimorfisme? (I)

institut sa palomera

A+B

Quin comportament tindrà això?

Doncs depèn del valor de les variables A i B...

Són enters, són strings...??

Què és Polimorfisme? (II)

institut sa palomera

```
class Operacions_Concatenar{
    //mètode que suma dos enters
    void suma (int num1,int num2)
    {
        System.out.println('el resultat:' + (num1+num2));
        }
    //sobrecàrrega del mètode per a sumar dos cadenes
    void suma(String str1, String str2)
    {
        String result = str1 + ' ' + str2;
        System.out.println('el resultat:' + result);
        }
}

public class Main {
    public static void main(String[] args)
    {
        //crear un objecte de la classe
        Operacions_Concatenar abc = new Operacions_Concatenar();
        //crida del 1er mètode
        abc.suma (3,4);
        //crida del 2on mètode
        abc.suma ('Hello' , 'World!');
    }
}
```

Presentació elaborada per Xavier Martin per al mòdul 5
Versió 3.3
Data: 31/01/2023