

CFGS Desenvolupament d'Aplicacions Multiplataforma (DAM)
CFGS Desenvolupament d'Aplicacions Web (DAW)

Mòdul 5 – Entorns de Desenvolupament.
UF3 – Introducció al disseny orientat a objectes

Exemples UML



Índex de contingut

Diagrama de classes.....	3
Diagrames de casos d'ús.....	5
Diagrames de seqüència.....	9
Diagrames d'activitats.....	12
Diagrama d'estats.....	18

Diagrama de classes

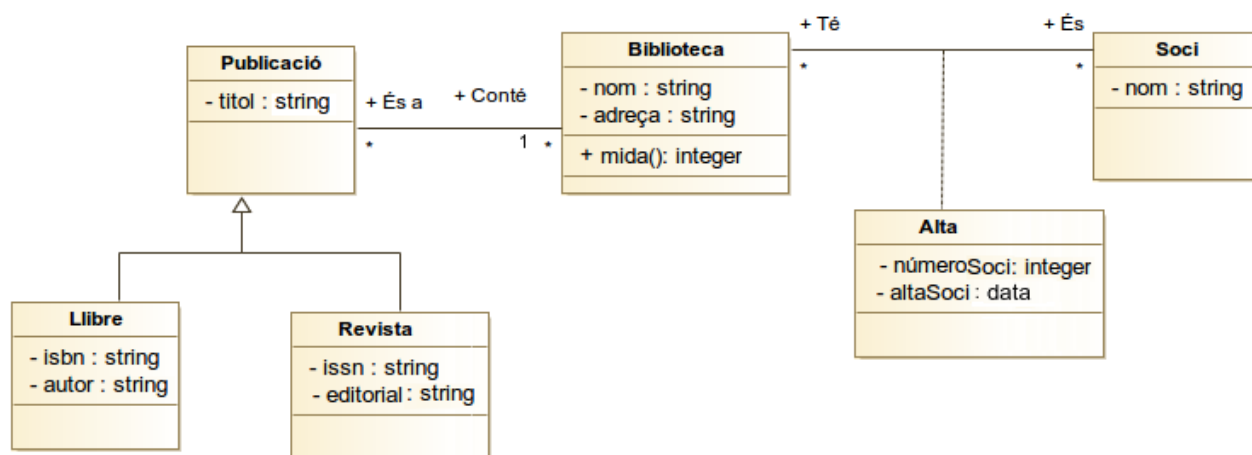
Cadascun dels següents exemples representa en forma de diagrama de classes l'enunciat corresponent. A cada enunciat s'indica les classes que hi han d'aparèixer. No obstant, pot caldre afegir alguna classe associativa o alguna subclasse. La resta de classes s'esmenten explícitament. El nom de les classes sempre ha d'aparèixer en singular. El tipus cadena sempre es refereix a una cadena de caràcters. Els atributs sempre seran privats, mentre que els mètodes sempre seran públics. S'ometran els *getters* i *setters* (es suposa que, per a cada atribut caldrà definir un *setter* i un *getter*).

Exemple 1: Biblioteca

Classe Biblioteca: representa una biblioteca. Té com a atributs el nom i l'adreça, tots dos de tipus cadena de caràcters. Té com a mètode *mida()*, que retorna un enter amb el nombre de publicacions que conté. Una biblioteca sol tenir bastants publicacions i socis.

Classe Publicació: representa una publicació d'una biblioteca. Una publicació pot estar a més d'una biblioteca (ja que parlem de títols, no d'exemplars). Una publicació té l'atribut títol, de tipus cadena. Una publicació pot ser de dos tipus: llibre i revista; si és llibre, cal enregistrar també el seu isbn i el seu autor; si és de tipus revista, cal enregistrar també el seu issn i l'editorial. L'isbn, l'issn, l'autor i l'editorial són de tipus cadena de caràcters.

Classe Soci: representa un soci d'alguna biblioteca. Un soci pot ser-ho de més d'una biblioteca. Ens interessa enregistrar el nom del soci (que és de tipus cadena) i, per cada biblioteca de la qual sigui soci, el seu número de soci, que és un enter, i la seva data d'alta (que és de tipus data).



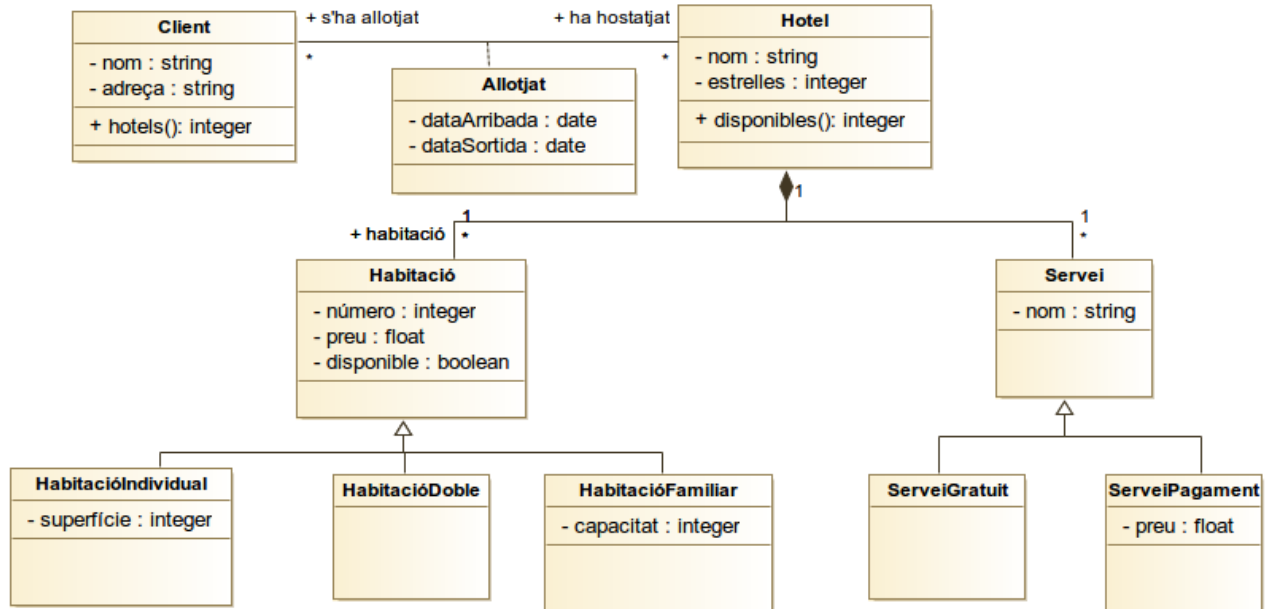
Exemple 2: Habitacions d'un hotel

Classe Hotel: representa un hotel; té els atributs nom, de tipus cadena, i estrelles (de tipus enter); també té el mètode *disponibles()*, que retorna un enter amb el nombre d'habitacions disponibles actualment a l'hotel. Un hotel està format per habitacions i serveis (recepció, cambrers, nevera, spa, piscina...).

Classe Habitació: representa una habitació de l'hotel; té els atributs número, de tipus enter, preu, de tipus float, i disponible, de tipus booleà. Hi ha habitacions individuals, dobles i familiars; cal tenir enregistrar, per les habitacions individuals, la superfície que tenen (es representarà amb un enter, que informa dels metres quadrats que té) i, per les habitacions familiars, la capacitat (també representada amb un enter).

Classe Servei: representa un servei de l'hotel; té l'atribut nom; hi ha serveis gratuïts (com la recepció, la neteja d'habitacions...) i d'altres de pagament (com la bugaderia, la sauna...). En aquest darrer cas, ens interessa enregistrar el seu preu.

Classe Client: representa un client d'un o més hotels. Té els atributs nom i adreça, de tipus cadena. També té el mètode hotels(), que retorna un enter amb el nombre d'hotels als quals s'ha allotjat el client durant els últims 12 mesos. Per a cada client cal enregistrar en quins hotels s'ha allotjat i, per a cada vegada que s'hi ha allotjat, la data d'arribada i de sortida.



Exemple 3: Comandes

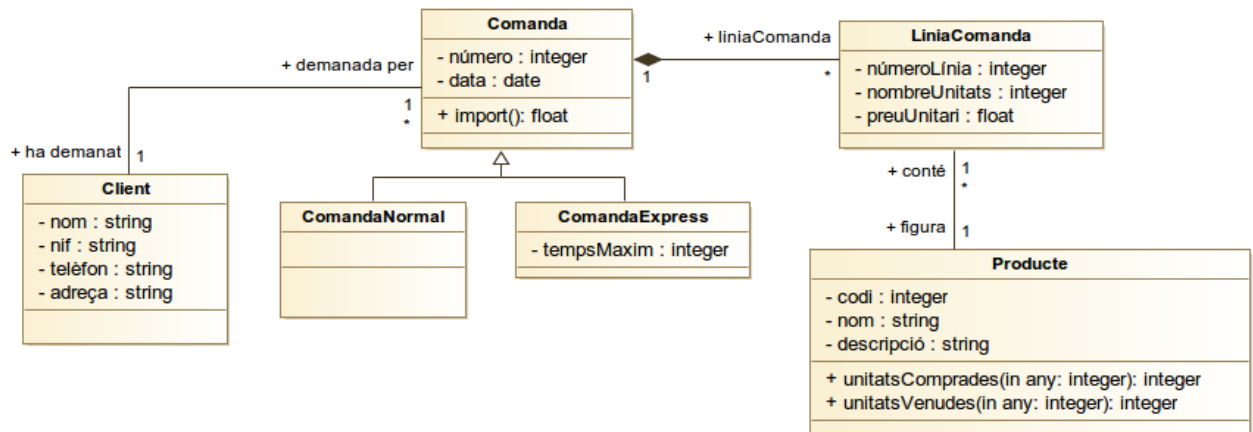
Classe Comanda: representa una comanda que ens ha fet un client. Té els atributs número (és un enter) i data (és de tipus data). A més, té el mètode import(), que retorna un float amb l'import total de la comanda. La comanda està formada per línies de comanda.

Hi ha dos tipus de comandes: normals i express; d'aquestes darreres ens interessa enregistrar el temps que trigen, com a màxim, a ser servides. És un enter i representa el nombre d'hores màxim que pot trigar-se a servir.

Classe LíniaComanda: representa una línia de la comanda. Cada línia té el seu número de línia (dins la comanda, lògicament), el nombre d'unitats i el preu unitari del producte en el moment de fer la comanda. El número de línia i el nombre d'unitats són enters, mentre que el preu unitari és de tipus float. Una línia de comanda sempre fa referència a un sol producte.

Classe Producte: representa un producte associat a alguna línia d'alguna comanda. Té com atributs el seu codi (de tipus enter), el seu nom (de tipus cadena de caràcters) i una descripció (de tipus cadena). Un producte pot aparèixer a més d'una línia de comanda. La classe Producte també té els mètodes unitatsComprades(any) i unitatsVenudes(any) que, respectivament, retornen un enter amb el nombre d'unitats comprades i venudes d'aquest producte durant l'any indicat pel paràmetre del mètode (que és de tipus enter).

Classe Client: representa un client que, almenys, ens ha demanat una comanda. Té els atributs de tipus cadena nom, NIF, telèfon i adreça.



Diagrames de casos d'ús

Conceptes principals:

- **Cas d'ús:** acció o operació del sistema desencadenada per un actor.
- **Actor:** qui desencadena un cas d'ús o en rep el resultat.
- **Extensió:** mecanisme per separar d'un cas d'ús una o més parts que són opcionals. Quan es fa cal indicar la condició que provoca la realització d'aquest cas d'ús. Es sol fer quan ens interessa destacar aquesta part opcional o bé perquè també forma part d'altres casos d'ús.
- **Inclusió:** mecanisme per separar del cas d'ús una part que és bàsica; sol fer-se quan una part del cas d'ús és comuna a diferents casos d'ús o perquè és important destacar aquesta part (sinó, no es recomana realitzar la separació).
- **Associació:** camí de comunicació entre un actor i un cas d'ús.

Com s'ha apuntat en la descripció dels conceptes anteriors, només s'ha de desdoblar un cas d'ús en altres (utilitzant l'*extensió* o la *inclusió*) per un motiu important. Per exemple, no s'han de desdoblar necessàriament totes les parts opcionals (això ens podria portar a fer una extensió per cada *if* del programa). Només es fa quan es consideri adient. Com s'ha indicat, normalment, els motius són aquests:

- És important destacar el cas d'ús inclòs o estès.
- Una part d'un cas d'ús és comuna també a altres casos d'ús. Quan això passa, es sol utilitzar la inclusió.

En aquest article es comenta més extensament la problemàtica i se'n donen criteris: <http://www.abiztar.com.mx/articulos/casos-a-incluir-casos-a-extender.html>

Als exemples que segueixen, si s'esmenta explícitament una part opcional, es considerarà que és prou important com per indicar-ho al diagrama. D'altra banda, es considerarà també que els casos d'ús utilitzats per més d'un dels altres casos d'ús cal separar-los amb una inclusió.

Exemple 1: Gestió senzilla d'una biblioteca

Un usuari pot:

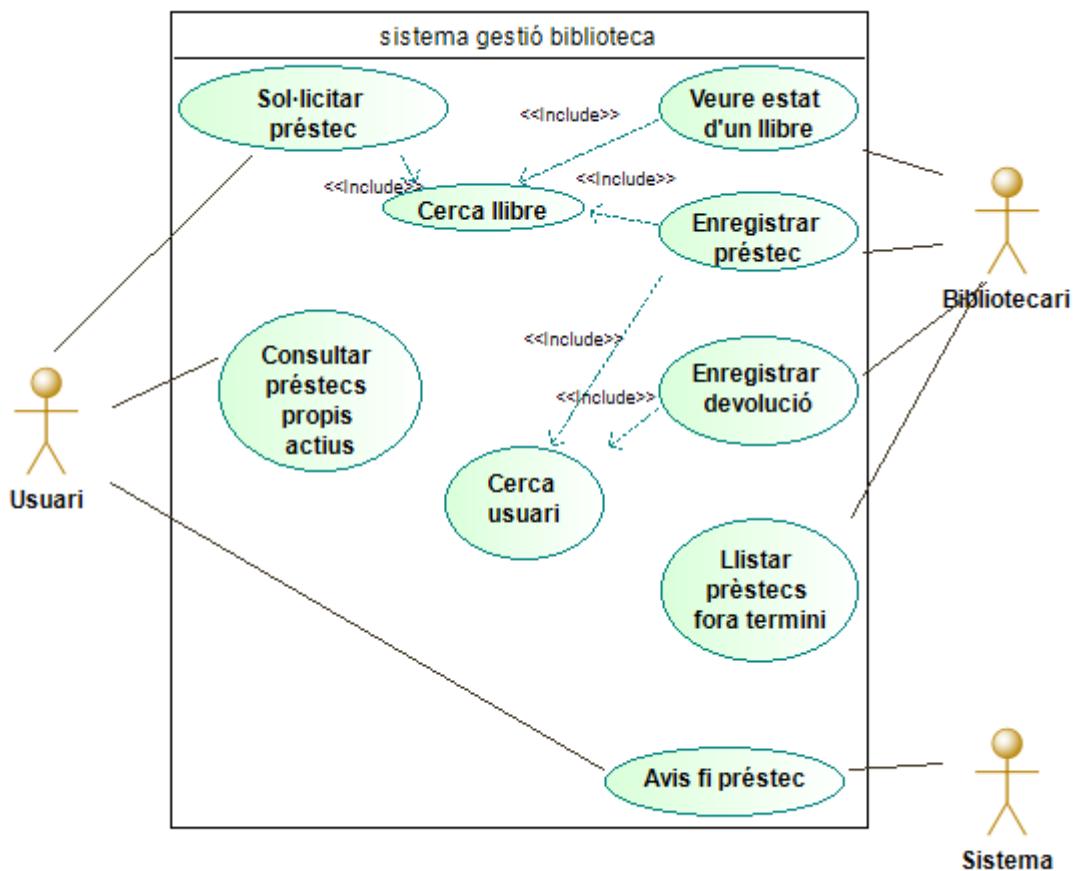
- Sol·licitar un préstec, prèvia cerca del llibre que desitja.
- Consultar els préstecs que té actius.

Un bibliotecari pot:

- Veure l'estat d'un llibre, prèvia cerca d'aquest.
- Enregistrar un préstec. Inclou la cerca del llibre i de l'usuari.
- Enregistrar la devolució d'un préstec. També inclou la cerca del llibre i de l'usuari.
- Llistar els préstecs fora de termini.

El sistema:

- Avisa a l'usuari el dia abans de la finalització d'un dels seus préstecs.



- Comentaris:

- Els actors són l'Usuari, el Bibliotecari i el Sistema, doncs són els que fan accions i/o en reben les conseqüències.
- Cada punt de l'enunciat és una acció que fa l'actor corresponent. Per tant, cadascun es representarà com, com a mínim, un cas d'ús.
- A més, les accions "cerca llibre" i "cerca usuari" són utilitzades per més d'un dels casos d'ús anteriors. Per tant, convé separar-los del cas d'ús on s'utilitzen amb una inclusió, ja que són accions que sempre es fan.

Exemple 2: Gestió senzilla de les reserves d'un hotel

Un client pot:

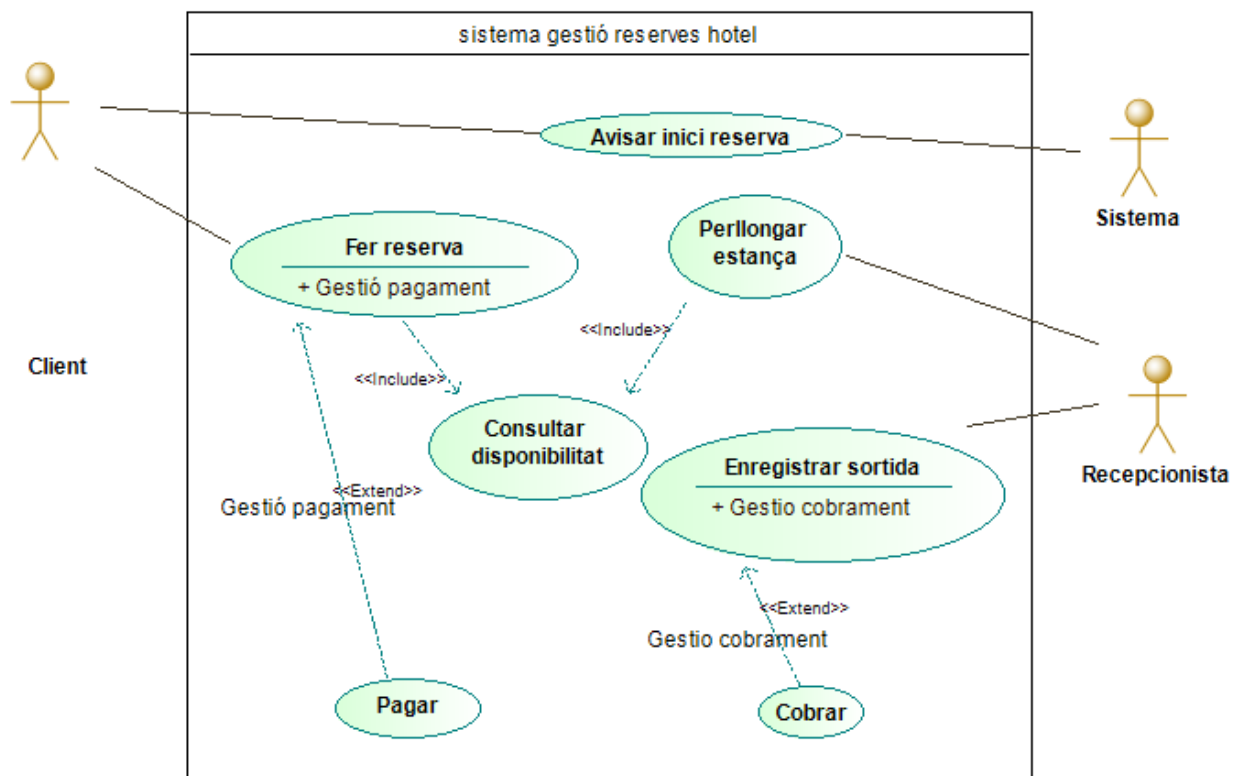
- Fer una reserva i, eventualment, un pagament, prèvia consulta de la disponibilitat d'habitacions.

El recepcionista pot:

- Perllongar una estança, prèvia consulta de la disponibilitat d'habitacions.
- Enregistrar una sortida. Pot incloure cobrar (és un procés diferent de pagar).

El sistema:

- Avisa el client el dia abans de l'inici de la seva reserva.



- Comentaris:

- Els actors són el Client, el Recepcionista i el Sistema, doncs són els que fan accions i/o en reben les conseqüències.
- Cada punt de l'enunciat és una acció que fa l'actor corresponent. Per tant, cadascun es representarà com, com a mínim, un cas d'ús.
- L'acció "consultar disponibilitat" és utilitzada per més d'un dels casos d'ús anteriors. Per tant, convé separar-la del cas d'ús on s'utilitza mitjançant una inclusió, ja que és una acció que sempre es fa.
- Les accions "pagar" i "cobrar" són opcionals. Com apareixen a l'enunciat, entendrem que són prou importants com per separar-les del cas d'ús principal.

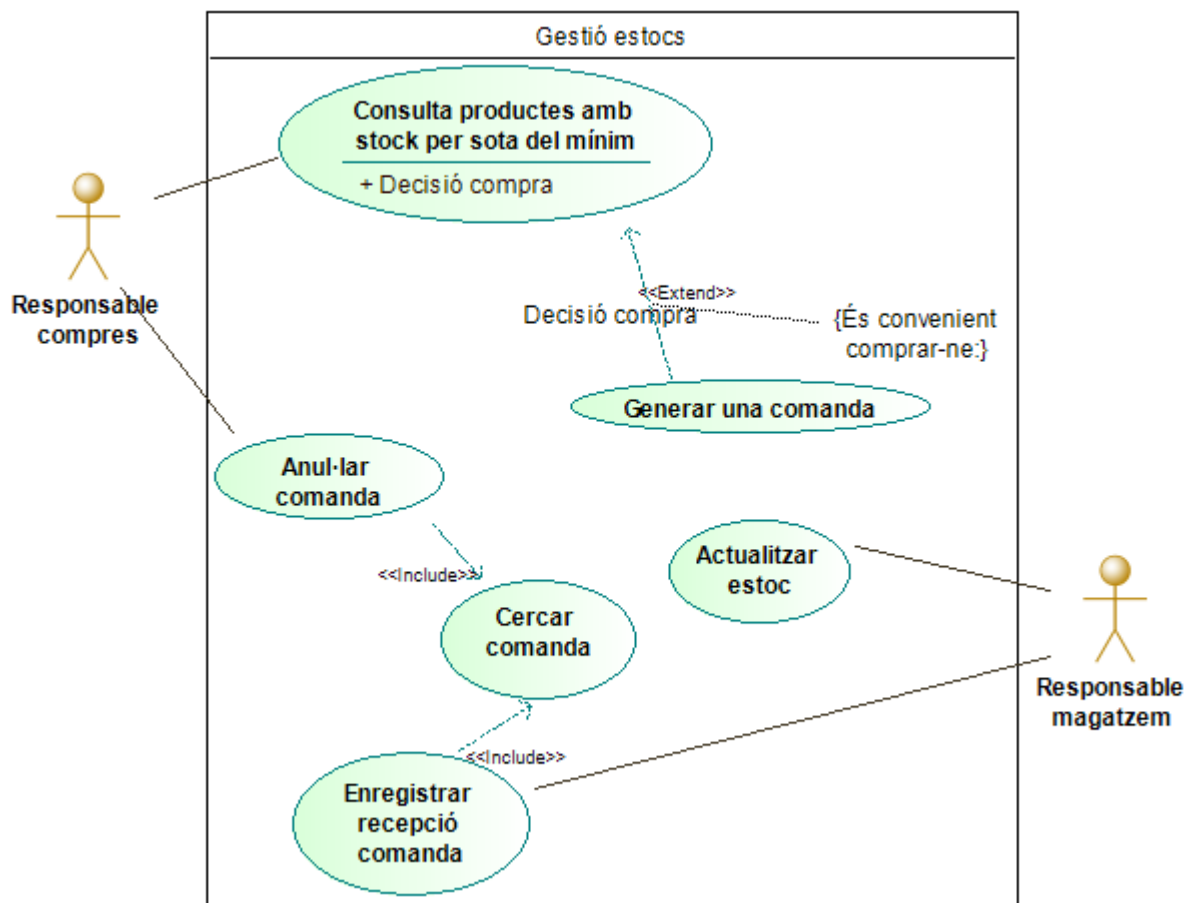
Exemple 3: Gestió senzilla d'estocs:

- El responsable compres pot:

- Consultar els productes amb estoc per sota del mínim; si creu convenient comprar-ne, generar una comanda.
- Anul·lar una comanda, després d'haver-la cercat.

El responsable del magatzem pot:

- Enregistrar una comanda rebuda, després d'haver-la cercat.
- Actualitzar l'estoc quan es rep o es ven mercaderia.



- Comentaris:

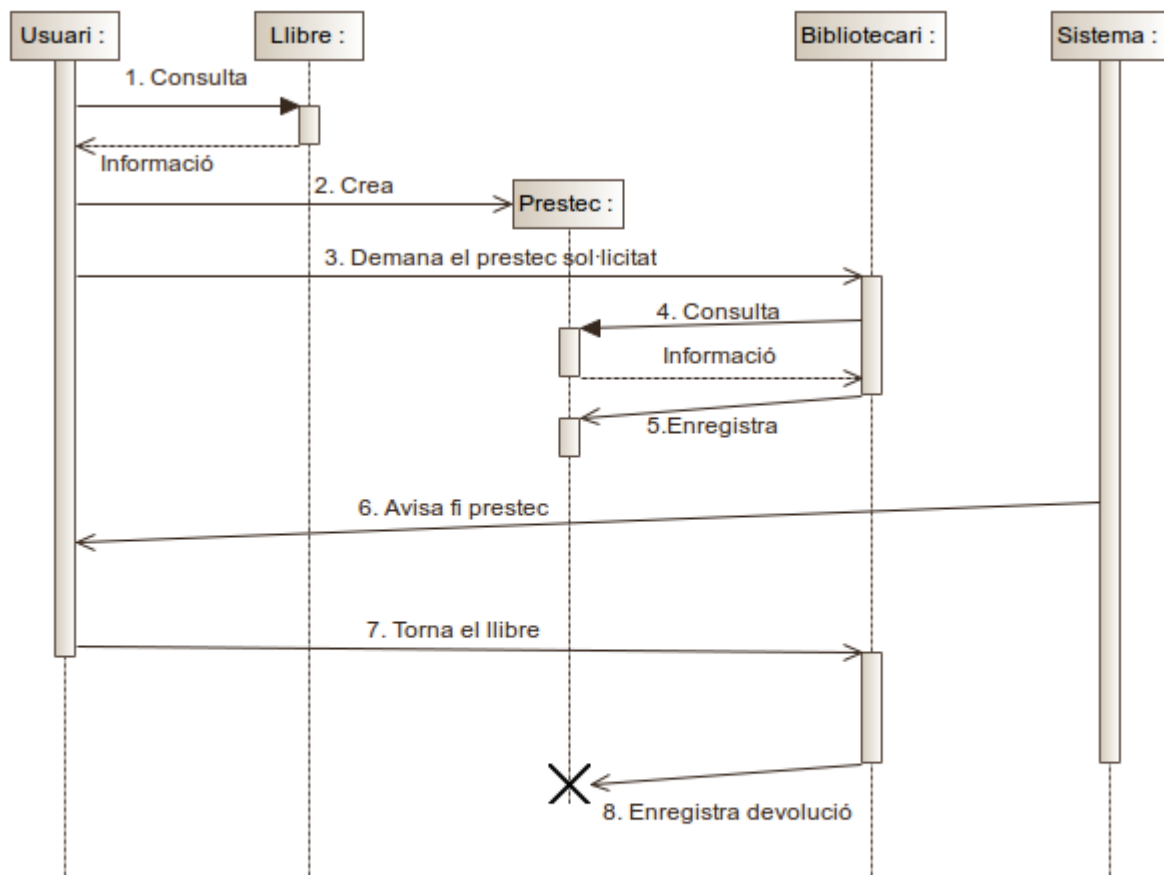
- Els actors són el Responsable de compres i el responsable del magatzem, doncs són els que fan accions i/o en reben les conseqüències.
- Cada punt de l'enunciat és una acció que fa l'actor corresponent. Per tant, cadascun es representarà com, com a mínim, un cas d'ús.
- L'acció "cercar comanda" és utilitzada per més d'un dels casos d'ús anteriors. Per tant, convé separar-la del cas d'ús on s'utilitza mitjançant una inclusió, ja que és una acció que sempre es fa.
- L'acció "generar una comanda" és opcional. Com apareix a l'enunciat, entendrem que és prou important com per separar-les del cas d'ús principal.

Diagrames de seqüència

Exemple 1: Gestió del préstec de llibres en una biblioteca

Cal representar el procés habitual que segueix un préstec:

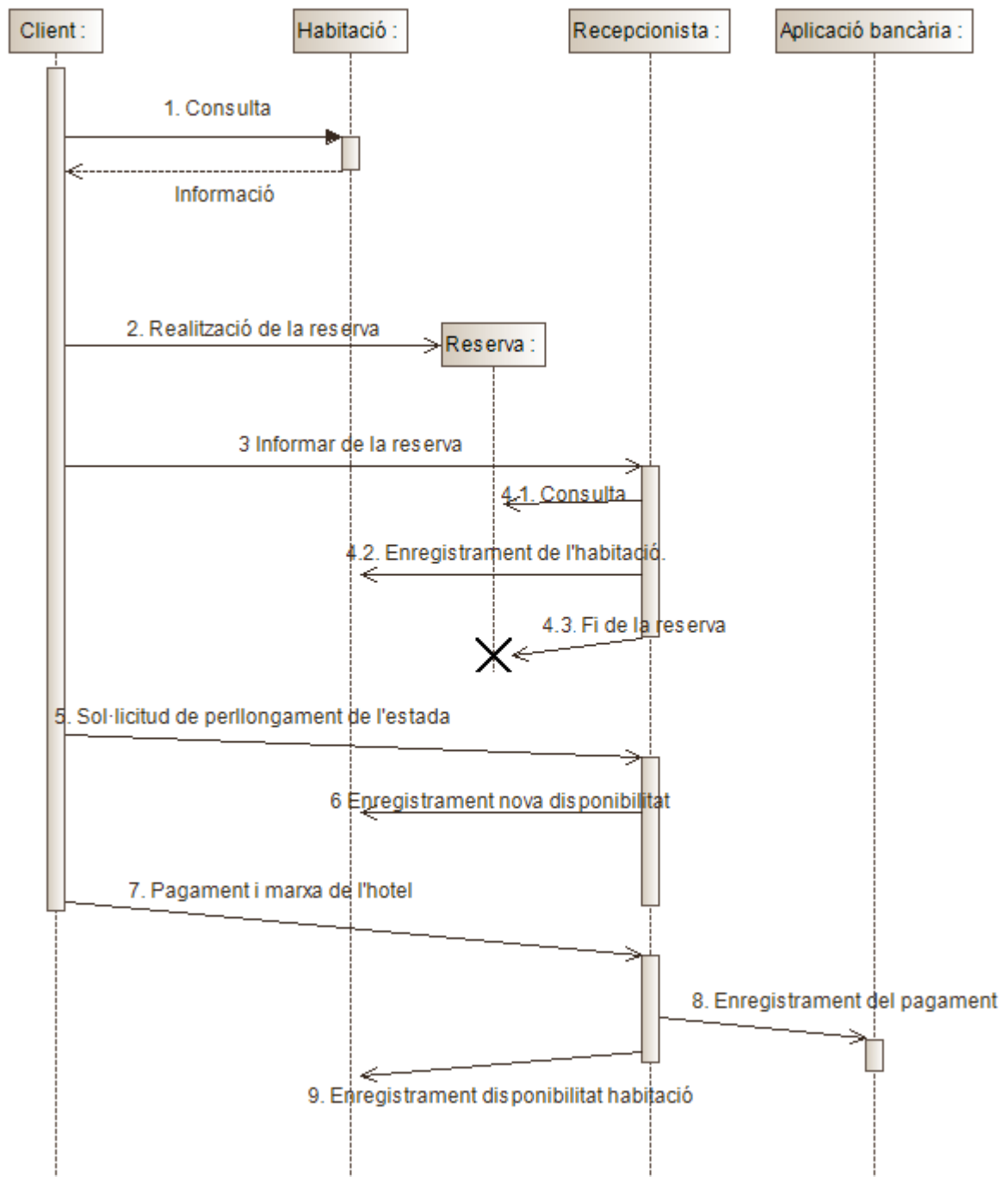
1. L'usuari consulta un llibre.
2. L'usuari fa una sol·licitud de préstec. El préstec es crea com a sol·licitat.
3. L'usuari demana el préstec sol·licitat al bibliotecari.
4. El bibliotecari consulta la informació sobre el préstec sol·licitat.
5. El bibliotecari enregistra el préstec. El préstec passa a estar actiu.
6. El dia abans de la finalització del préstec, el sistema avisa l'usuari.
7. L'usuari torna el llibre al bibliotecari.
8. El bibliotecari enregistra la devolució. El préstec finalitza.



Exemple 2: Gestió de les reserves en un hotel

Cal representar el procés que pot seguir una reserva i ús d'unes habitacions:

1. El client consulta les habitacions disponibles.
2. El client reserva una habitació.
3. El client arriba a l'hotel i indica al recepcionista la reserva que ha realitzat.
4. El recepcionista consulta la reserva i enregistra les habitacions que ocupa el client. Amb això últim acaba la reserva.
5. El client demana al recepcionista perllongar l'estada.
6. El recepcionista actualitza la informació de les habitacions.
7. Quan ha de marxar, el client paga al recepcionista i, a continuació, abandona l'hotel.
8. El recepcionista cobra al client quan li està pagant i enregistra el pagament a l'aplicació bancària.
9. El recepcionista enregistra que l'habitació ha quedat disponible.



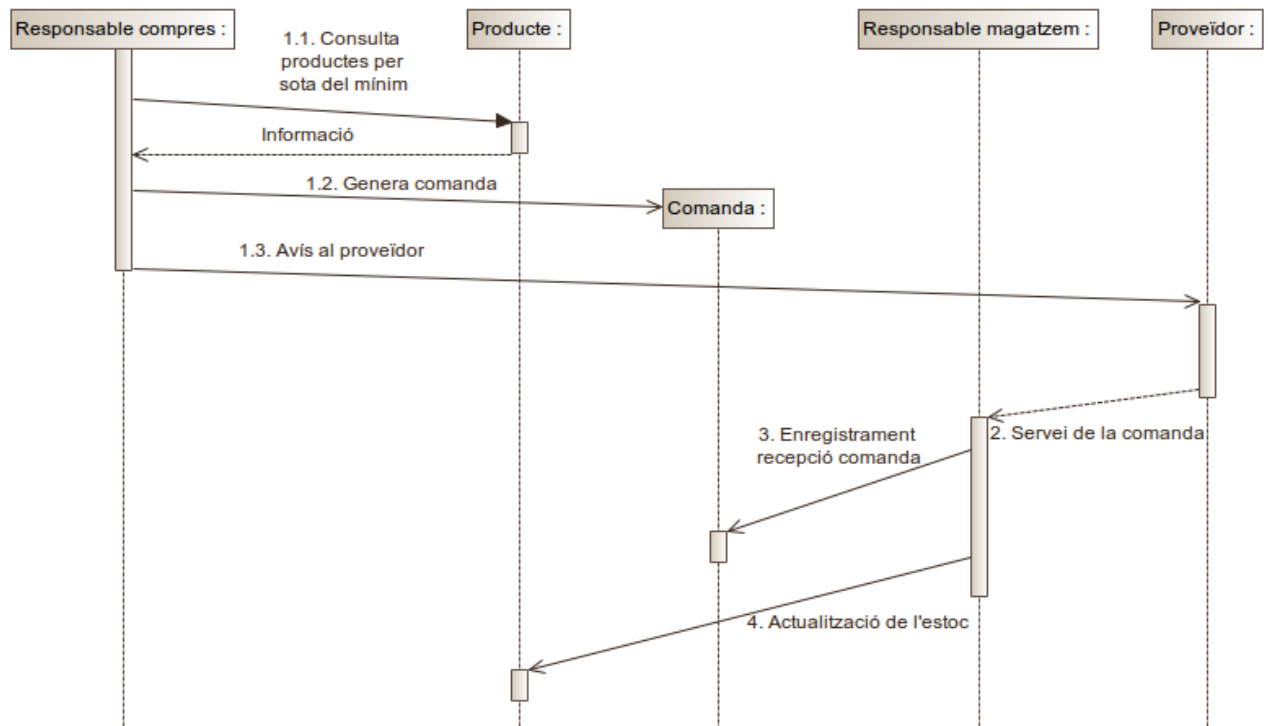
Exemple 3: Gestió d'estocs

Cal representar el procés habitual que segueix la gestió d'estocs quan s'acaba l'estoc d'un producte.

1. El responsable de compres consulta els productes per sota del mínim, en genera comandes i avisa al proveïdor.
2. El proveïdor serveix la comanda.

Com a conseqüència:

3. El responsable del magatzem enregistra la recepció de la comanda. La comanda passa a l'històric.
4. El responsable del magatzem actualitza l'estoc.



Diagrames d'activitats

Un diagrama d'activitats representa en detall les activitats que passen dins un cas d'ús o dins el comportament d'un objecte.

Si una activitat depèn d'una condició, es representa amb un node de decisió, que té forma de rombe (◆).

Si dues o més activitats poden fer-se simultàniament, es representa amb un fork i un join (—).

Són diagrames bastant semblants als tradicionals *ordinogrames* que s'utilitzen en programació per indicar els passos que segueix un algorisme.

Als dos primers exemples parlem d'un hotel que utilitza targetes de cartró foradades que serveixen de clau (targetes-clau). Cada cop que arriba un client a l'hotel, cal generar-li'n una amb una combinació pròpia, que és destruïda quan marxa.

Exemple 1: El recepcionista aten l'arribada d'un client a l'hotel

El recepcionista realitza les següents activitats:

Demana la identificació al client.

Demana al client si té una reserva realitzada.

Si el client té una reserva realitzada:

Enregistra que es farà efectiva (amb aquesta operació, l'habitació passa a figurar com a ocupada).

Si no té reserva,

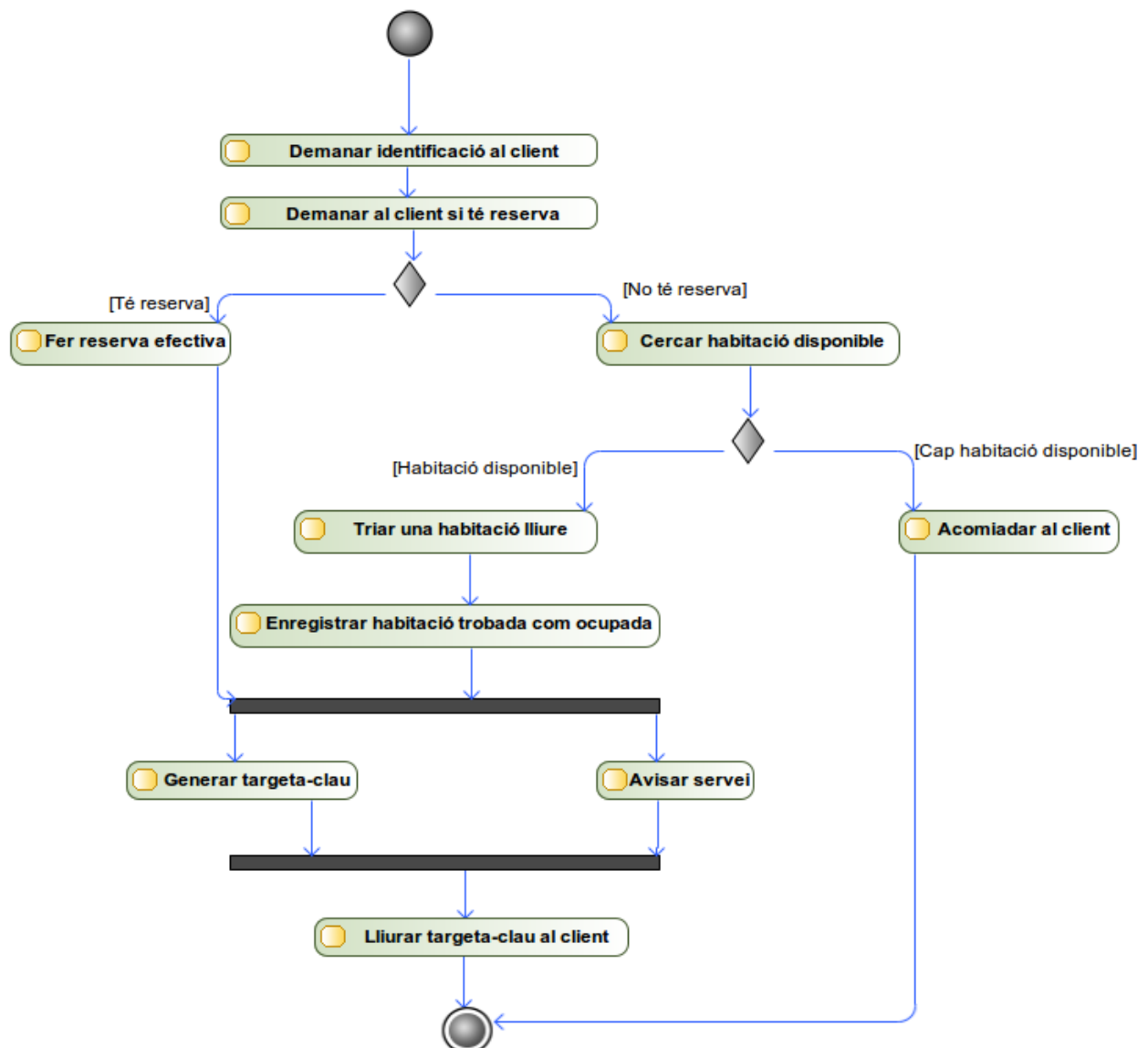
Comprova si hi ha habitacions disponibles.

Si hi ha habitacions disponibles, en tria una i enregistra que està ocupada.

Si no n'hi ha, es limita a acomiadar al client.

Si finalment s'ha pogut assignar una habitació al client, genera una targeta-clau i, a la vegada, avisa al servei que vingui per pujar les maletes a l'habitació.

Finalment, lliura la targeta-clau al client.



Exemple 2: El recepcionista d'un hotel atén la sortida d'un client.

El recepcionista realitza les següents activitats:

Rep la targeta-clau del client.

Amb el nombre d'habitació, localitza les dades de l'estada.

Calcula l'import

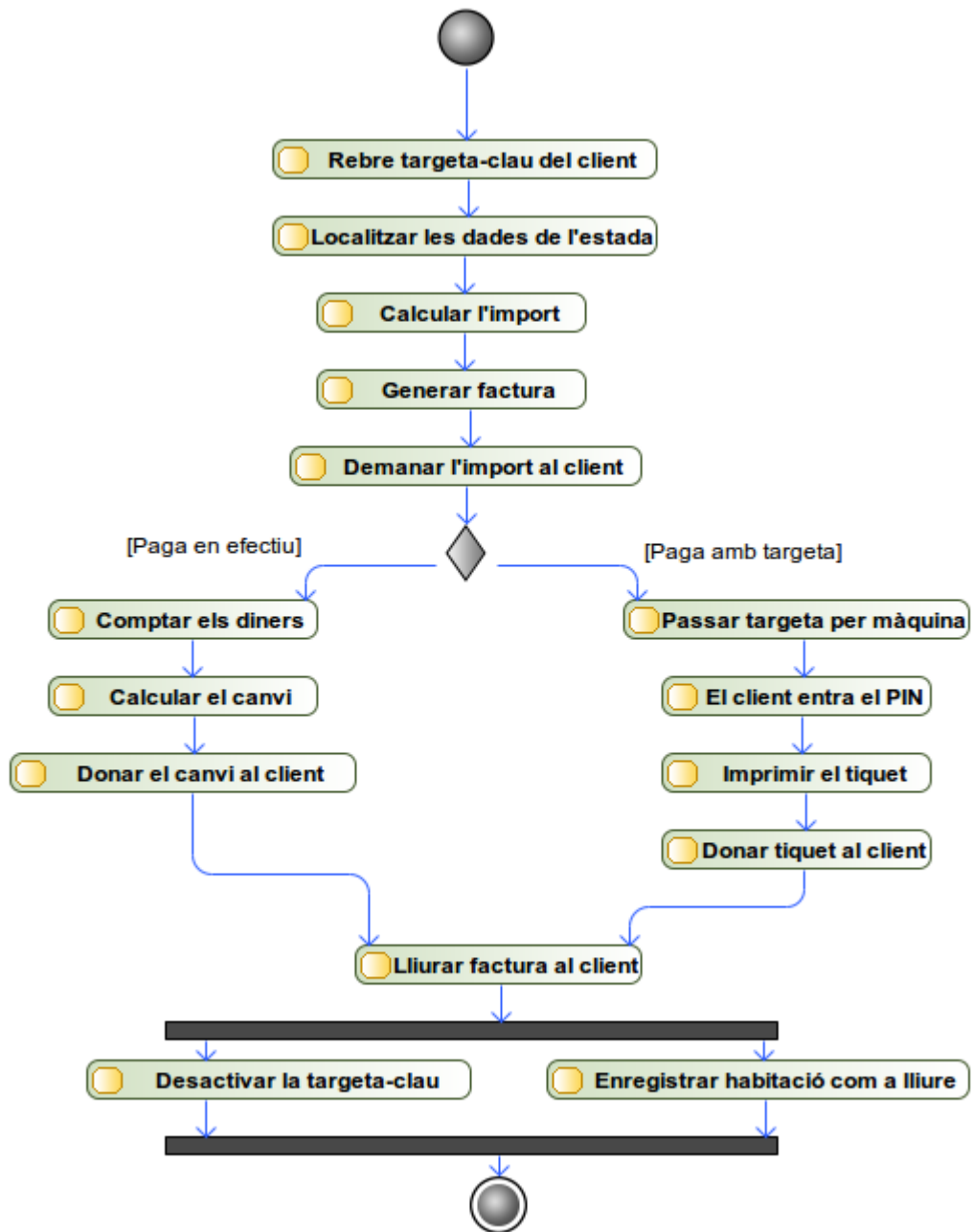
Genera la factura.

Demana l'import al client:

- Si paga en efectiu:
 - Compta els diners.
 - Calcula el canvi.
 - Dona el canvi al client.
- Si paga amb targeta:
 - Passa la targeta per la màquina
 - El client entra el PIN
 - Imprimeix el tiquet
 - Dona el tiquet al client.

Lliura la factura al client.

Desactiva la targeta-clau i, a la vegada, enregistra l'habitació com a lliure.



Exemple 3: Un caixer automàtic.

Per simplificar, suposarem que el caixer només dispensa diners.

El caixer realitza les següents activitats:

Llegeix la targeta introduïda.

Demana el PIN.

Llegeix el PIN

- Si el PIN és incorrecte:
 - Torna a demanar el PIN
 - Llegeix el PIN
 - Si el PIN és incorrecte:
 - Torna a demanar el PIN
 - Llegeix el PIN
 - Si el PIN és incorrecte:
 - Mostra un missatge d'error avisant que no retornarà la targeta i ja no fa res més
- Si l'usuari ha entrat un PIN correcte:
 - Demana a l'usuari la quantitat que vol treure.
 - Comprova si l'usuari està autoritzat a treure aquesta quantitat.
 - Si l'usuari no està autoritzat a treure la quantitat, mostra un missatge d'error
 - Si l'usuari està autoritzat a treure la quantitat:
 - Comprova si el caixer té prou bitllets.
 - Si no té prou bitllets, mostra un missatge d'error
 - Si té prou bitllets, expedeix els bitllets.
- Torna la targeta a l'usuari.

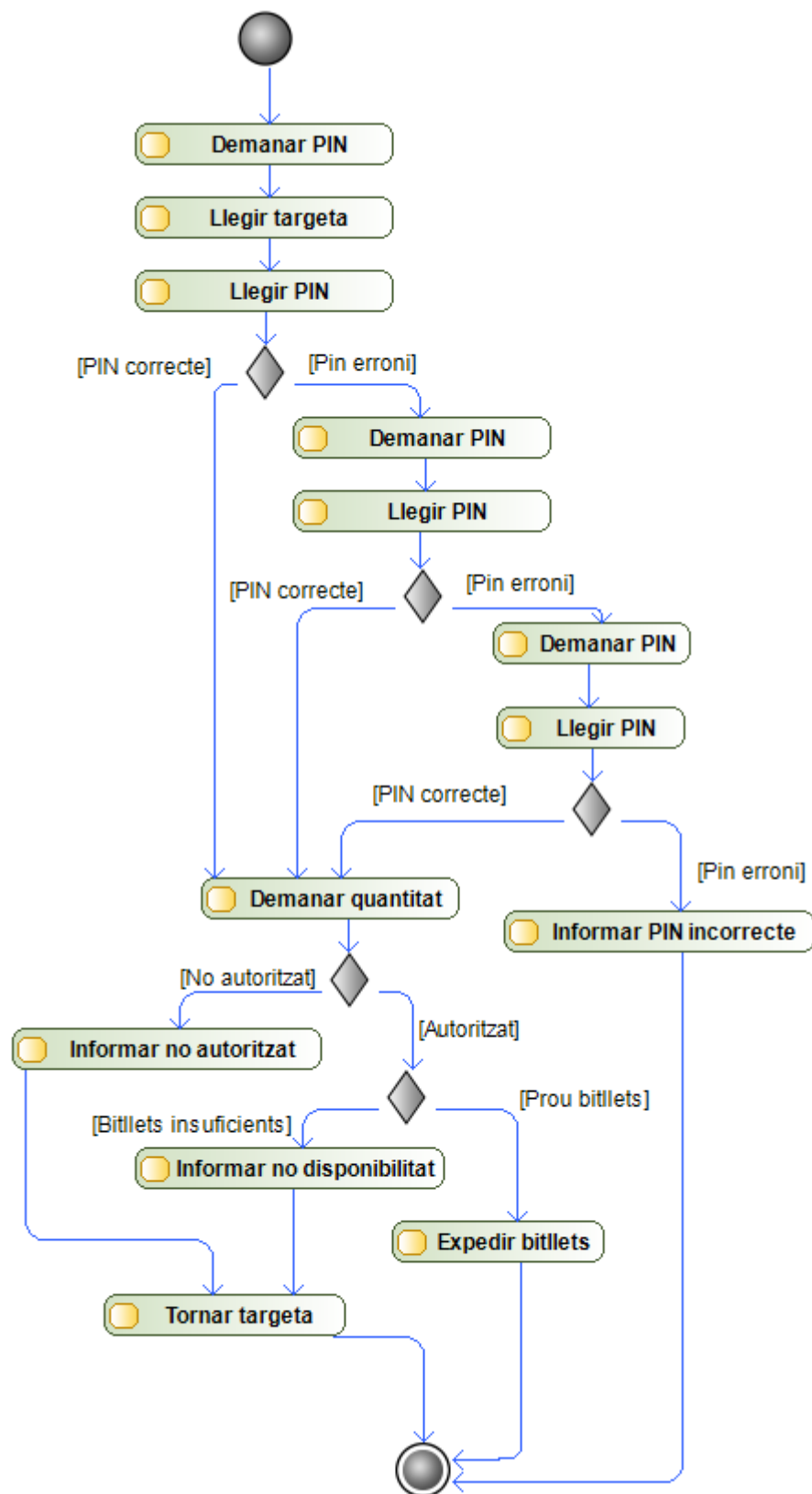


Diagrama d'estats

El **diagrama d'estat** representa els estats pels quals passa un objecte. Determinats esdeveniments fan que l'objecte canviï d'estat.

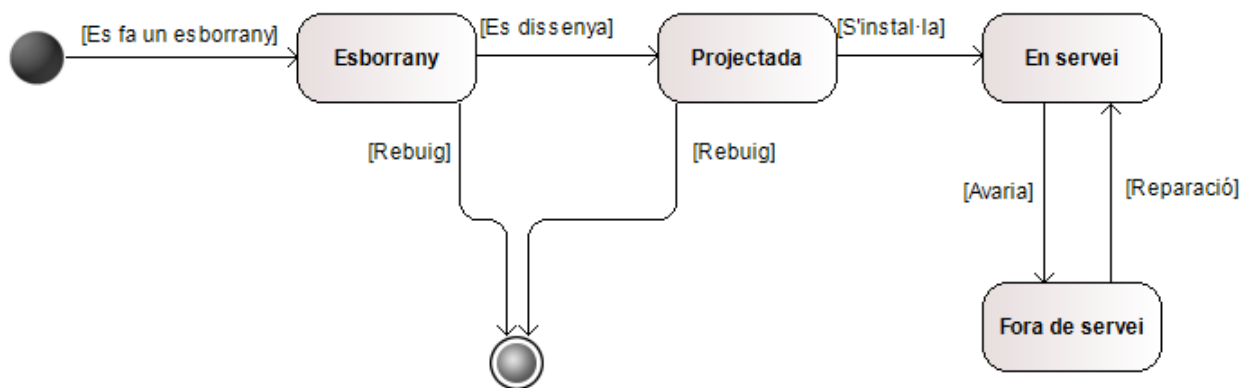
Als enunciats, els noms dels estats van entre cometes (« »). Al diagrama no es posa nom als estats finals.

Exemple 1: Instal·lació d'un ascensor.

Es fa un esborrany de la instal·lació prevista. La instal·lació està en estat «esborrany». En aquest estat, la instal·lació pot ser acceptada o rebutjada.

Si és acceptada, a partir de l'esborrany es dissenya un projecte. La instal·lació està en estat «projectada». També en aquest punt la instal·lació pot ser rebutjada o acceptada.

Si s'accepta, es realitza la instal·lació i passa a estar en estat «en servei». Quan l'ascensor és «en servei», pot avariar-se i restar «fora de servei» fins que es repara i torna a estar «en servei».



Exemple 2: Compra en línia.

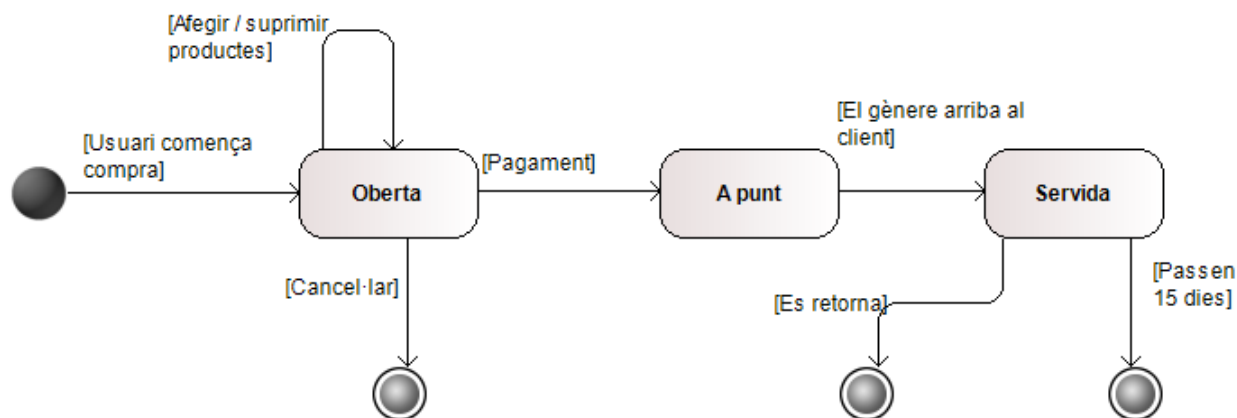
L'usuari inicia la compra.

Aquesta queda «oberta» fins que l'usuari fa el pagament o cancel·la l'operació.

Mentre la compra és «oberta», l'usuari pot afegir-hi o suprimir productes o, com s'ha dit, cancel·lar l'operació.

Quan l'usuari paga la compra, aquesta passa a estar «a punt». Ho està fins que el gènere arriba al client. En aquest punt, la compra passa a estar «servida».

Una compra «servida» pot ser retornada durant els 15 dies següents a la recepció. Passats aquests 15 dies, passa a quedar acceptada.



Exemple 3: Màquina expendedora

Per simplificar suposem que només serveix un tipus de producte i que mai es queda sense existències (l'empresa té un sistema molt eficient de reposició de producte). No hi ha estat final.

A l'inici, la màquina està «apagada». Un cop la màquina es connecta, passa a estar «disponible». Només en aquest estat es pot tornar a apagar. Quan el client introdueix una moneda, la màquina pot passar a un d'aquests estats:

- «pagat», si aquesta moneda serveix per pagar justament el que falta.
- «falta», si aquesta moneda encara és insuficient per pagar el que falta.
- «sobra», si aquesta moneda sobrepassa l'import demanat

Quan la màquina està en estat «sobra», roman en aquest estat fins que torna el canvi; en aquest moment, passa a l'estat «pagat».

Quan la màquina està en estat «falta», roman en aquest estat fins que l'usuari introdueix una nova moneda; llavors la màquina pot passar a l'estat «pagat», romandre a l'estat «falta» o passar a l'estat «sobra» en funció de si aquesta moneda completa exactament l'import restant, no hi arriba o el supera.

Quan la màquina està en estat «pagat», serveix el producte i torna a l'estat «disponible».

