Disseny d’un sistema de gestió per a una cambra de congelació robotitzada de masses congelades.

Nom Estudiant Francesca Tomás Artigues.
ETIS

Nom Consultor José Antonio Raya Martos.

10 / 01 /2005

Disseny d’un sistema de gestió per a una cambra de congelació robotitzada de masses congelades.

Algun dels grans fabricants del sector de les masses congelades, fan servir robots per l’emmagatzematge, servei i gestió dels productes de les cambres de congelació. Aquests robots només emmagatzemen i serveixen palets de caixes d’un mateix tipus de producte, així doncs, només són útils quan el client és un gran consumidor com un distribuïdor o una delegació, però no un petit comerç, com un restaurant o una fleca, els quals també són clients potencials del fabricant. Per tant, el problema esdevé en el moment de servir a aquells clients que tenen problemes d’emmagatzematge i aleshores es veuen obligats a fer comandes de varis productes i que, a més a més, en la seva totalitat no fan ni molt menys un palet sencer. A partir d’aquí la idea és dissenyar un sistema, utilitzant la tècnica de l’orientació a objectes (O-O), que gestioni un conjunt de robots que a més d’emmagatzemar i gestionar una cambra de congelació, siguin capaços de servir les comandes fent una selecció de caixes de productes concrets, és a dir, aquelles que sol·licita el client que ha fet la comanda per caixes i no per palets. Per qüestions d’organització i d’stockatge, aquests robots han de ser independents dels altres, és a dir, han de treballar en una cambra independent a la dels robots que treballen per palets (actualment aquesta feina la fan treballadors que ja treballen dins d’una cambra independent a la dels robots).

ÍNDEX

1. Introducció..... 3
1.1. Justificació del TFC i context en el qual es desenvolupa:..... 4
1.1.1. Punt de partida..... 4
1.1.2. Aportació del TFC..... 4
1.2. Objectius..... 5
1.3. Enfocament i mètode seguit..... 5
1.4. Planificació..... 6
1.5. Productes obtinguts..... 7
1.6. Breu descripció..... 8
2. Recollida i documentació de requisits..... 10
2.1. Descripció del sistema..... 10
2.1.1. Organització de la cambra de congelació..... 10
2.1.2. Rutes dels repartiments..... 10
2.1.3. Divisió de les rutes dels repartiments..... 11
2.1.4. Comandes i repartiments..... 11
2.1.5. Devolucions..... 12
2.1.6. Productes..... 12
2.2. Descripció detallada dels principals objectes del sistema..... 12
2.2.1. Palets..... 12
2.2.2. Caixes..... 12
2.2.3. Passadissos..... 13
2.2.4. Prestatgeria..... 13
2.2.5. Columnes..... 13
2.2.6. Clients..... 13
2.2.7. Comandes..... 13
2.2.8. Línia de comanda..... 13
2.2.9. Devolucions..... 13
2.2.10. Repartiments..... 14
2.2.11. Rutes..... 14
2.2.12. Subrutes..... 14
2.3. Guions..... 14
2.3.1. Guió del Robot..... 14
2.3.2. Guió de l'operador logístic..... 15
2.3.2.1. Els clients..... 15
2.3.2.2. Les rutes..... 15
2.3.2.3. Els repartiments..... 15
2.3.2.4. Les devolucions..... 15
2.3.2.5. Les noves comandes..... 15
2.4. Actors..... 16
2.5. Casos d'ús..... 17
2.5.1. Relacions entre actors i casos d'ús..... 18
2.6. Documentació textual dels casos d'ús..... 19
2.7. Especificació dels requisits de la interfície d'usuari..... 34
3. Anàlisi..... 36
3.1. Identificació de les classes d'entitat..... 36
3.2. Identificació de les classes de frontera..... 38
3.3. Identificació de les classes de control..... 44
3.4. Modelat d'interaccions entre classes: diagrames de seqüència..... 44
3.5. Identificació d'associacions entre classes d'entitat..... 50
3.6. Identificació dels atributs de les classes d'entitat..... 51
3.7. Paquets d'anàlisi i de servei..... 53
4. Disseny del sistema..... 56
4.1. Reutilització de classes..... 56

4.2. Identificació de les operacions i revisió dels atributs 58
4.3. Diagrama estàtic de disseny..... 72
4.4. Disseny de la persistència..... 73
4.4.1. Supressió de l'herència..... 72
4.4.2. Transformació del model estàtic al model entitat_relació..... 72
4.4.3. Transformació del model entitat-relació en taules relacionals..... 75
4.4.4. Gestors de disc..... 76
4.5. Disseny de la interfície d'usuari..... 82
4.6. Disseny dels subsistemes..... 84
5. Conclusions..... 86
Glossari..... 87
Bibliografia..... 89
Annexos..... 90

1. INTRODUCCIÓ

1.1. Justificació del TFC i context en el qual es desenvolupa

1.1.1. Punt de partida

Per fer el disseny partirem de la descripció que tenim del sistema[1]. Aquesta descripció, juntament amb la descripció detallada dels principals objectes que formaran part del sistema[2], no està extreta d'una empresa de masses congelades real, i per tant, és totalment fictícia, encara que per crear-la, s'ha seguit el model d'una empresa real on es pot donar la necessitat de crear una cambra de congelació robotitzada de les característiques proposades. S'ha intentat que en aquesta documentació no falti cap detall per poder desenvolupar el TFC amb èxit. Per aconseguir-ho, a partir del model de l'empresa real de masses congelades, s'han intentat reflectir quins serien els punts de vista i necessitat dels diferents tipus d'empleats que participarien, d'una forma directa o indirecta, en la manipulació del sistema.

1.1.2. Aportació del TFC

Si l'empresa té un gran volum de clients minoritaris, encara que la inversió inicia que s'ha de fer per implementar el sistema dissenyat sigui elevada, els avantatges en un futur immediat són considerables, assumint la inversió inicial i augmentant els beneficis de l'empresa:

- El fet que la manipulació dels productes de la cambra estigui feta per robots i no per persones, fa que el servei als clients minoritaris sigui molt més ràpid i eficient, ja que la rapidesa en la manipulació dels productes que pot aconseguir un robot és molt més elevada que la que pot aconseguir una persona, i per un altre costat, la probabilitat d'errors en la manipulació dels productes es redueix considerablement perquè un robot farà el que li ordeni el sistema sense cap tipus de variació, mentre que si la manipulació és feta per persones, entra en joc el factor equivocació i per tant hi ha un risc afegit d'errors. Això donarà com a resultat un millor servei i un augment de la capacitat de servei, podent efectuar més serveis dels que abans de la implementació del sistema s'estaven fent.

- Pel que fa als riscos laborals, l'implementació del sistema també aporta beneficis per l'empresa i el treballador, ja que treballar a 20° graus sota zero (què és la temperatura de les cambres de congelació), per moltes prevencions que es prenguin, comporta moltes malalties (moltes d'elles cròniques) que fan que el treballador tingui que deixar la feina amb les conseqüències negatives que això comporta tant pel treballador com per l'empresa. Per un altre costat, una persona no pot estar més d'un temps determinat treballant en aquestes condicions tèrmiques, amb la qual cosa l'empresa és veu obligada a augmentar el personal de la cambra per fer torns, o bé, a aturar el treball dins la cambra fins que el treballador pugui tornar a entra dins d'ella. Amb els robots no existeixen cap d'aquests inconvenients: poden treballar de forma continua sense risc a agafar malalties, menys quan s'espatllen, però el risc d'espatllar-se és molt més inferior al risc que té un treballador en posar-se malalt i el seu temps de recuperació és molt més curt que el d'un treballador.

[1] Mirar l'apartat 2.1 "Descripció del sistema" del capítol 2.
[2] Mirar l'apartat 2.2 "Descripció detallada dels principals objectes del sistema" en el capítol 2.

1.2. Objectius.

L'objectiu general i principal és dissenyar un sistema que gestioni una cambra de congelació robotitzada formada per un conjunt de robots que a més d'emmagatzemar i gestionar la cambra, siguin capaços de servir les *comandes* fent una selecció de caixes de productes concrets, és a dir, aquelles que sol·licita el client que ha fet la comanda per caixes i no per palets..

El sistema ha de ser capaç de[3]:

- Gestionar les rutes que fan els transportistes per lliurar les comandes als clients:
  - Assignar una ruta a un nou client.
  - Eliminar d'una ruta a un client que s'ha donat de baixa.
  - Crear una nova ruta.
  - Dividir una ruta en dues rutes noves.
  - Modificar una ruta.

- Gestionar les comandes fetes pels clients en realció al seu lliurament:
  - Organitzar les comandes fetes pels clients en repartiments.
  - Crear subrutes temporals de les rutes existents.
  - Crear nous repartiments.

- Gestionar les accions i respostes dels robots:
  - Enviar als robots les comandes que s'han de servir.
  - Gestionar el servei de les caixes de les comandes enviades als robots.
  - Gestionar el canvi d'ubicació dels palets dins la cambra.
  - Gestionar l'entrada d'un nou palet dins la cambra.
  - Gestionar la caducitat d'un palet de dins la cambra.

- Gestionar les devolucions de les caixes fetes pels clients.

Els objectius metodològics del TFC són:

- Utilitzar la tècnica de l'orientació a objectes (O-O) per analitzar i dissenyar el sistema.
- Seguir les dues primeres etapes del cicle de vida del *Rational Unified Process*, és a dir, l'etapa d'inici que correspon a la recollida de requisits i l'etapa d'elaboració que correspon a l'anàlisi i disseny.
- Utilitzar el model UML en la construcció dels diagrames que ens descriuran diversos aspectes de l'estructura i la dinàmica del sistema.
- Utilitzar el mètode de disseny de la persistència basat en les bases de dades relacionals.
- Utilitzar el patró *Template Method* per dissenyar els gestors de disc.

1.3. Enfocament i mètode seguit.

Com hem vist en l'apartat 1.2. d'aquest capítol, un dels objectius és seguir el mètode orientat a objectes seguint el cicle de vida del *Rational Unifed Process* en les seves dues primeres etapes, i per tant, a partir de la descripció del sistema[4] i de la descripció detallada dels seus principals objectes[5], en el capítol 2, farem la recollida i documentació dels requisits delimitant l'abast del sistema, per després, en el

[3] Mirar la descripció textual del casos d'ús al capítol 2 apartat 2.6 "Descripció textual dels casos d'ús".
[4] Mirar l'apartat 2.1 "Descripció del sistema" del capítol 2.
[5] Mirar l'apartat 2.2 "Descripció detallada dels principals objectes del sistema" del capítol 2.

capítol 3, passar a fer el seu anàlisis, i finalment, en el capítol 4, realitzarem el disseny, a partir de l'anàlisi de la recollida i documentació de requisits fet en el capítol 3, establint l'arquitectura general del sistema.

En la realització dels diferents diagrames que ens descriuen aspectes de l'estructura i dinàmica del sistema (com el diagrama de casos d'ús, el diagrama estàtic de classes,....) es farà servir el model UML usant la eina *Visio Professional 2002* de Microsoft.

Pel que fa al disseny de la persistència, es farà servir el mètode basat en les bases de dades relacionals ja que la base de dades de l'empresa és una base de dades relacional i per dissenyar els gestors de disc de la base de dades farem servir el patró *Template Method*, en el qual hi ha una superclasse abstracta (*GestorGeneric*) i dues subclasses intermèdies, *GestorRelacional* i *GestorFitxers*, que són superclasse de tots els gestors relacionals i de tots els gestors de fitxers respectivament. En el nostre cas els gestors seran relacionals, i per tan, farem servir *GestorRelacional*.

1.4. Planificació.

En el següent diagrama de Grantt es pot veure quina planificació s'ha seguit per realitzar el TFC:

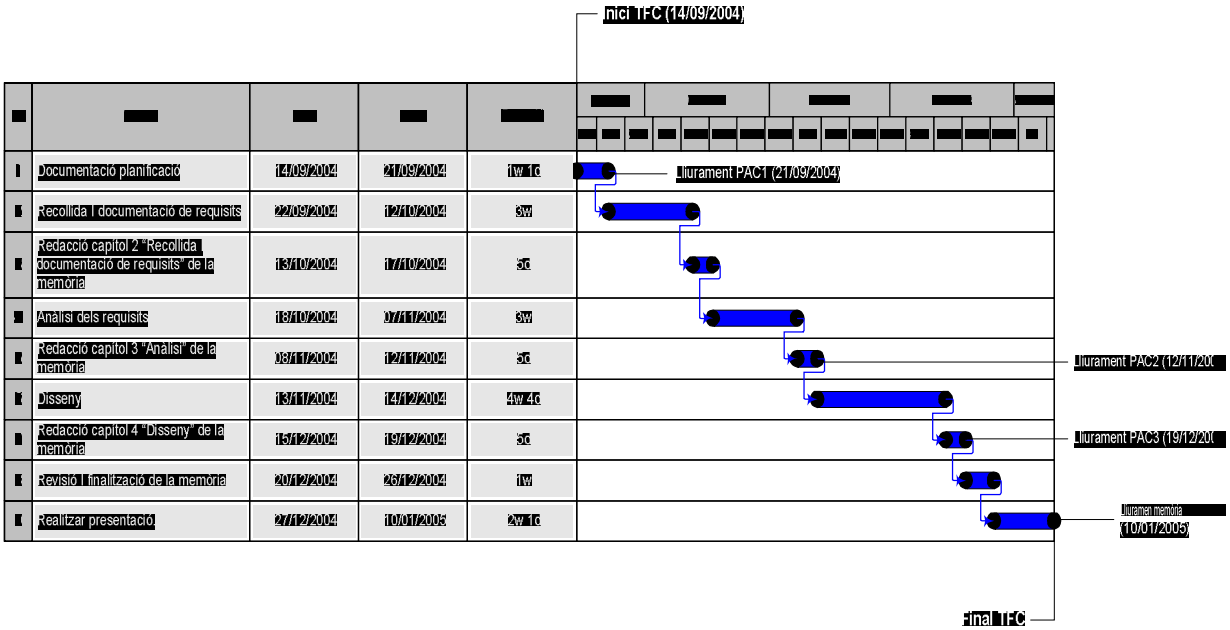


Diagrama de Grantt per a la planificació del TFC

- La columna "Tasca", indica el nom de cadascuna de les tasques que s'han de realitzar.
- La columna "Inici" indica la data en que s'ha de començar la tasca.
- La columna "Final" indica en quina data s'ha d'acabar la tasca.
- La columna "Duració" el temps total que hi ha per realitzar la tasca en setmanes ("w") i dies ("d").
- Els indicadors que veiem en el diagrama de Grantt ens indiquen quina posició ocupen les següents dates rellevants:

  - La data d'inici i de final del TFC.

- Les dates de lliurament de les tres PAC's. On la PAC1 correspon a la documentació de la planificació, la PAC2 a la documentació de la memòria referent al capítol 2 "Recollida i documentació de requisits" i al capítol 3 "Anàlisi", i finalment, la PAC3 correspon a la documentació de la memòria referent al capítol 4 "Disseny".
- La data de lliurament de la memòria i la presentació.

Pel que fa a la "Revisió i finalització de la memòria", dir que fa referència a la redacció dels capítols 1 i 5 [6], a la redacció de la bibliografia, i a la revisió del glossari, annexos i nucli de la memòria, que s'hauran fet al llarg de la recollida i documentació de requisits, de l'anàlisi i del disseny, quan en el diagrama de Grantt ens referim a "Redacció capítol...." [7]. Per aquest fet, he allargat el període d'anàlisi i disseny, demorant el lliurament de les PAC's 2 i 3 a les dates indicades en el diagrama de Grantt.

1.5. Productes obtinguts.

En el capítol 2 "Recollida i documentació de requisits", s'han obtingut els següents productes:

- Guions: Obtenim dos guions, el del robot i el de l'operador logístic. Aquests dos guions ens expliquen les operacions més freqüents que faran l'operador logístic i el robot en relació al sistema de gestió..
- Actors: Obtenim cinc actors, Operador logístic**,** Robot**,** Sistema de fàbrica i Instant temporal, els quals interactuen amb el sistema donant-li o rebent informació.
- Casos d'ús: Obtenim 22 casos d'ús[8]. Els quals ens indiquen els processos autònoms iniciats per un actor o per un altre cas d'ús del sistema. Representen les funcions ofertes pel sistema, identificant les entrades i les sortides i descriuen el què d'aquestes funcions.
- Diagrama de casos d'ús: Aquest diagrama ens indica les relacions d'extensió o d'inclusió que hi ha entre els casos d'ús del sistema, alhora que ens indica quins actors interactuen amb quins casos d'ús per activar-los, o bé, per rebre informació generada en el cas d'ús que li envia la informació.
- Requeriments de la interfície d'usuari: Són tots aquells requeriments que s'han de tenir en compte en el moment de dissenyar les interfícies d'usuari, que en aquest cas només seran dissenyades per a la figura de l'operador logístic.

En el capítol 3 "Anàlisi", s'han obtingut els següents productes:

- Classes d'entitat: En principi obtenim 14 classes d'entitat a partir de la descripció textual dels casos d'ús[9], però en la construcció del diagrama estàtic de les classes d'entitat[10] afegirem una nova classe, per tant, obtenim 15 classes d'entitat. Aquestes classes corresponen als objectes del domini del sistema, dels quals el sistema ha de fer servir informació.
- Classes de frontera: En el nivell d'anàlisis representen la interfície d'usuari per pantalla.
- Classes de control: En el nostre cas, hi ha una per cada cas d'ús i seran les responsables de crear o esborrar les classes de frontera.

[6] Mirar l'index del TFC.
[7] Mirar tasca 3, 5 i 7 del diagrama de Grantt.
[8] Mirar taula de l'apartat 2.5 del capítol 2.
[9] Mirar apartat 3.1 "Identificació de les classes d'entitat" del capítol 3.
[10] Mirar apartat 3.5 "Identificació d'associacions entre classes d'entitat" del capítol 3.

- Diagrames de seqüència: Els diagrames de seqüències que hem obtingut ens han servit per comprovar si falta algun objecte o algun comportament en l'especificació del sistema[11].
- Diagrama estàtic d'anàlisi: Aquest diagrama ens mostra les diferents relacions que hi ha entre les diferents classes d'entitat, sense incloure ni els atributs, ni les operacions de les classes.
- Atributs de les classes d'entitat: Cadascuna de les classes d'entitat té un conjunt d'atributs d'instància que en serveixen per definir els objectes de les diferents classes.
- Paquets d'anàlisi i de servei: Hem obtingut tres paquets d'anàlisi, GestióRutes, GestióCambra i GestióDevolucions. Aquest tres paquets en que s'ha dividit el sistema ens facilitaran la feina en el moment d'implementar-lo, dividint la feina d'implementació en diferents grups, un per cada un dels paquets. Cadascun d'aquests paquets té un conjunt de paquets de servei cadascun dels quals reuneixen un conjunt de casos d'ús relacionats entre si.

En el capítol 4 "Disseny", s'han obtingut els següents productes:

- Classes agregades: Hem obtingut classes agregades a les classes del sistema administratiu i de fàbrica que fem servir per poder agregar-hi nous atributs i operacions.
- Operacions: Són les operacions de les classes d'entitat i de les classes de control obtingudes en l'etapa d'anàlisi.
- Diagrames estàtic de disseny: Diagrama estàtic de les classes d'entitat amb els seus atributs, operacions i visibilitat de ambdues.
- Model entitat-relació: Obtingut a partir del model estàtic ens serveix per dissenyar les taules relacionals de la base de dades.
- Taules relacionals: Obtingudes a partir del mode entitat-relació, són les taules relacionals del nostre sistema que han d'implementar-se a la base de dades de l'empresa. Hi ha una per cada una de les classes d'entitat.
- Gestors de disc: Proporcionen les operacions bàsiques sobre les files de les taules relacionals del nostre sistema. Hi ha un per taula.
- Disseny de la interfice d'usuari a partir de les diferents classes i dels requisits obtinguts al capítol 2.
- Subsistemes: Els subsistemes obtinguts corresponen amb els paquets d'anàlisi obtinguts a l'etapa d'anàlisi.

1.6. Breu descripció.

El capítol 2 "Recollida i documentació de requisits", comença amb la descripció del sistema per després passar a identificar els guions, actors i casos d'ús extrets de la descripció. Continua fent una descripció textual dels casos d'ús. Finalment és fa una especificació dels requisits de la interfície d'usuari.

El capítol 3 "Anàlisis", comença identificant les classes d'entitat, frontera i control a partir de la documentació textual dels casos d'ús feta al segon capítol, per després, a través de diagrames de seqüències, comprovar si hi ha alguna classe més. Continua identificant les associacions i atributs de les classes d'entitat. Finalment identifica els paquets d'anàlisi i de servei en que és pot dividir el sistema.

El capítol 4 "Disseny", comença dissenyant la reutilització de les classes d'entitat del sistema administratiu i de fàbrica que es veuen implicades en el nostre sistema, per continuar identificant les operacions de les classes d'entitat i control, i re-

[11] Mirar apartat 3.4 "Modelat d'interaccions entre classes: diagrames de seqüències" del capítol 3.

visant si a les classes d’entitat els manca algun atribut o s’ha de modificar. Com a resultat d’aquests primers passos es crea el diagrama estàtic de disseny. A continuació és fa el disseny de la persistència de les dades del sistema, per acabar fent el disseny de la interfície d’usuari i dels subsistemes.

2. RECOLLIDA I DOCUMENTACIÓ DE REQUISITS

2.1. Descripció del sistema

La descripció del sistema està dividida en 5 apartats: **Organització cambra de congelació, Rutes de repartiment, Comandes, Devolucions** i **Productes.**
Aquesta és tota la informació de que es disposa per crear el sistema, i per tant, d'on es treuen totes les dades necessària per l'obtenció dels requisits del sistema (objectius, classes, operacions, requeriments no funcionals,...).

2.1.1. Organització cambra de congelació

La cambra estarà dividida per prestatgeries formant passadissos. A cada un d'aquests passadissos trobarem un robot que s'encarregarà de la recepció, organització i servei dels productes que tindrà a les prestatgeries d'esquerra i dreta. Cada un dels passadissos es comunicarà a l'exterior per una porta per on els robots recolliran i serviran els productes pertinents[12].

Dins la cambra hi haurà emmagatzemats dos palets de caixes de cada un dels tipus de productes que es fabriquen. El sistema ha de controlarà el stockatge de la cambra. Cada palet pot contenir 60, 30 o 15 caixes d'un determinat producte, depèn de la mida de la caixa (gran, mitjana o petita) que ve determinat pel producte que conté. Pel que fa a l'organització dels productes dins la cambra, cada palet té una ubicació determinada identificada per un codi. Les ubicacions dels palets es distribueixen en dues files al llarg de cadascuna de les prestatgeries, formant *columnes* de dos ubicacions. A cadascuna de les *columnes* es col·loquen els dos palets d'un mateix producte, a la part de sobre el palet que ha entrat l'últim i a la part de sota el que ha entrat primer [13]. El robot per servir les caixes sempre agafarà les caixes del palet de sota (què són les que caducaran abans). Un cop s'acabi el palet de sota, el sistema avisarà al robot que col·loqui el palet de sobre a sota i demanarà a fàbrica la fabricació d'un nou palet que el robot col·locarà a sobre quan aquest li sigui servit. Cal tenir en compte que pot passar que alguna o algunes caixes caduquin dins la cambra perquè no tinguin demanda, aquest fet també s'ha de controlar per treure el palet caducat de dins la cambra i reposar-ho per un de nou. Quan un palet caduqui el sistema avisarà al robot pertinent, demanarà a fàbrica la fabricació d'un nou palet de les mateixes característiques i el robot retornarà el palet a l'exterior de la cambra.

2.1.2. Rutes de repartiment

Els robots en el moment de servir les caixes han de seguir un ordre determinat. Aquest ordre està determinat per les rutes que l'operador logístic ha assignat a cadascun dels transportistes, és a dir, quan es carrega un camió encarregat de fer el repartiment d'una determinada ruta, hem de tenir en compte que la comanda que es lliura primer és l'última en carregar-se. Cada un dels clients estarà assignat a una única ruta i cadascuna de les rutes tindrà assignada una quantitat de clients variable, és a dir, el nombre de clients en les diferents rutes no té perquè ser igual i aquest nombre pot variar segons els clients es donin d'alta o de baixa. Quan es fa un client nou l'operador logístic s'encarrega d'assignar-li una ruta determinada, donar-li un lloc determinat dins la ruta (depenent de la seva adreça) i canviar el lloc dels clients posteriors a ell (tots els clients posterior a ell, han de desplaçar-se un lloc cap a baix). De la mateixa manera, si un client es dona de baixa l'operador lo-

[12] Mirar Annex 1
[13] Mirar Annex 2

gístic l'eliminarà de la seva ruta i canviarà el lloc assignat dins la ruta d'aquest client als clients que són visitats posteriorment a aquest (tots els clients posteriors a ell, han de desplaçar-se un lloc cap a dalt dins la ruta). El que si cal tenir en compte és que un codi postal no pot pertànyer a dues rutes alhora, ja que es calcula que un codi postal pot arribar a tenir un màxim de 15 clients, amb la qual cosa no té sentit que dues rutes comparteixin un mateix codi postal. Un transportista només té una ruta assignada la qual fa cada dia. Fer una ruta no implica visitar a tots els clients de la ruta, sinó només a aquells clients que han fet una comanda per un dia determinat.

2.1.3. Divisió de rutes de repartiment

Pot donar-se el cas que un dia determinat les comandes d'una ruta superin la capacitat del camió assignat a aquesta ruta, llavors el sistema dividirà la ruta. Si la capacitat del camió és superada en 60 caixes o més de mida gran, l'operador logístic assignarà a un transportista eventual per a que faci aquests repartiments, si la capacitat del camió és superada en menys de 60 caixes de mida gran, llogar a un transportista eventual no surt a compte i per tant el repartiment d'aquestes caixes el farà el mateix transportista que fa la ruta inicial. Aquests transportistes eventuals no pertanyen a la plantilla de l'empresa. Si aquest fet es converteix en un continuo, l'operador logístic haurà de dividir la ruta de forma definitiva i cadascuna de les divisions es convertirà en una nova ruta a la que li assignarà un nou transportista fix (passarà a formar part de la plantilla de l'empresa), menys una d'elles que es considerarà una modificació de la ruta inicial ja que l'únic canvi que patirà és una reducció de clients, els quals passaran a formar part de les noves rutes. Per a que una ruta surti a compte el transportista de la ruta com a mitja ha de fer 30 repartiments diaris (180 setmanals) d'una mitja de 4 caixes de mida gran cadascun, és a dir ha de repartir una mitja de 120 caixes de mida gran diàries (720 setmanals). Per tant, sortirà a compte dividir la ruta de forma definitiva si la mitja de 720 caixes de mida gran setmanals, passa a convertir-se en una mitja de 1440 caixes setmanals. En tot cas l'última paraula en la divisió definitiva de rutes la té l'operador logístic.

2.1.4. Comandes i Repartiments.

Una comanda és un conjunt de caixes d'un tipus o més de productes servides per un o més robots i demanades per un únic client. Els robots no poden començar a servir una nova comanda d'un mateix repartiment fins que no acaben de servir la comanda actual, i no poden començar a servir les comandes d'un nou repartiment fins que no acaben de servir totes les comandes del repartiment actual. Un repartiment és un conjunt de comandes fetes pels clients d'una mateixa ruta i repartides pel mateix transportista al llarg de la seva ruta diària. Les comandes són introduïdes en el sistema administratiu per l'administratiu i el sistema s'encarregarà de crear els repartiments segons les rutes assignades a cada client. En el moment de fer el repartiment s'ha de tenir en compte la mida i la quantitat de caixes de les comandes per no sobrepassar la capacitat màxima de carrega del camió. Els repartiments són d'un dia per l'altre com a mínim, per tan l'operador logístic ha de deixar enllestit els repartiments el dia anterior al repartiment per evitar qualsevol problema (Contractar més camions si calen, dividir rutes, introduir nous clients si es dona el cas, etc..). L'administratiu li passa les comandes al sistema a través del sistema administratiu. Quan hi ha un camió apunt per ser carregat l'operador logístic consulta els repartiments que s'han de lliurar aquest dia, selecciona el repartiment que ha de fer el camió i el sistema passa la primera comanda del repartiment que s'ha de servir a tots els robots, quan un robot ha servit totes les seves caixes de la comanda avisa al sistema (si el robot no té caixes per servir en la

comanda avisa al sistema com si ja hagués servit totes les caixes) i el sistema descomptà les caixes servides dels palets corresponents. Quan el sistema rep l'avís de tots els robots, aquest els hi passa la segona comanda del repartiment, i així fins que acaben de servir totes les comandes del repartiment.

2.1.5. Devolucions

Poden existir devolucions de producte perquè hi ha hagut algun error en la comanda/es d'un o més clients, o bé perquè les caixes estan en mal estat (trencades, gènere dolent,...). Les devolucions són introduïdes en el sistema per l'operador logístic. L'administratiu tindrà que donar de baixa les caixes de la comanda pertinent i per tant, l'operador logístic té que passar-li les devolucions a través del sistema.

2.1.6. Productes

Els productes es divideixen en dos grans grups: PAN i BOLLERIA. Cadascun d'aquest dos grups es divideixen en famílies, que alhora és poden dividir en subfamílies, etc. i cadascun dels productes pertany a una d'aquestes famílies[14].
Tant els dos grans grups, com les famílies, les subfamílies i els productes poden variar en un futur (afegir més o treure algun o alguns). El fitxer mestre d'articles es troba a la base de dades de l'empresa[15] i l'encarregat de modificar aquest fitxer és el sistema administratiu.

2.2. Descripció detallada dels principals objectes del sistema

A partir de la descripció del sistema detallem les característiques dels principals objectes que el composen.

2.2.1. Palets

Dels palets ens cal saber quantes caixes porten, quina és la data de caducitat de les caixes que porta, a quina columna pertanyen i quina és la seva ubicació dins la columna. Per identificar-los farem servir un codi (codi_palet) que serà el mateix per tots els palets que el contingut de la seves caixes sigui el mateix producte, és a dir, quan reposem un palet per un altre aquest ha de tenir el mateix codi_palet que el reposat, i de la mateixa manera, dos palets situats a la mateixa columna, també han de tenir el mateix codi_palet. La seva distinció vindrà donada per la ubicació que ocupen dins la columna i la seva data de caducitat.

2.2.2. Caixes

Cadascuna de les caixes té un número d'identificació (codi_caixa). D'elles ens cal saber quina és la mida de la caixa (gran, mitjana o petita) i el palet al que pertany.

2.2.3. Passadissos

Dels passadissos només ens interessa saber a quin Robot pertanyen. El robot ve identificat per un número d'identificació (codi_robot), que pot ser el codi que identifica al passadís. Si pensem ve, en el nostre cas, passadís equival a robot.

[14] Mirar Annex 3.
[15] La base de dades de l'empresa la comparteixen els diferents sistemes de l'empresa i és en aquesta on es guarda tota la informació creada pels diferents sistemes. L'accés a la base de dades és restringit per als diferents sistemes, és a dir, només tenen accés a les dades que necessiten.

2.2.4. Prestatgeries

Cadascuna de les prestatgeries ve identificada per un número d'identificació (codi_prestatgeria). D'elles només ens cal saber a quin passadís pertanyen.

2.2.5. Columnes

Les columnes tenen un número d'identificació (codi_columna) què és igual a la referència del producte que contenen i a més dos codis "SOB" i "SOT", que diferencia les dues ubicacions que composen les columnes, "SOB" per la ubicació de sobre i "SOT" per la ubicació de sota. De les dues ubicacions també ens interessa saber si estan buides (no tenen palet) o no estan buides (tenen caixes o palet sencer). D'aquestes ens interessa saber a quina prestatgeria pertanyen.

2.2.6. Clients

Dels clients ens interessa saber el seu NIF, nom, domicili social, ciutat del domicili social, codi postal i amb quina ruta és troben relacionats.

2.2.7. Comandes

Cadascuna de les comandes té un número d'identificació (*numero_comanda*). D'elles ens interessa saber la data en que el client ha fet la comanda, la data en que s'ha de lliurar la comanda, quin client ha fet la comanda, quines són les línies de la comanda, si la comanda ja està servida o no, el total de caixes per mida gran, és a dir quatre caixes petites és consideraran una caixa i dues caixes mitjanes també es consideraran una caixa, i si la comanda està assignada a un repartiment, a quin repartiment està assignada[16].

2.2.8. Línia de comanda

Una línia d'una comanda s'identifica per la referència del tipus de producte que demana el client, i d'ella ens interessa saber el nombre de caixes demanades pel client del producte al que pertany la referència i la mida de les caixes (serà únic per totes ja que totes contenen el mateix producte). Una comanda ha de tenir com a mínim una línia de comanda (es demanen caixes d'un únic producte) i com a màxim, tantes com diferents productes demani el client[17].

2.2.9. Devolucions

Cadascuna de les devolucions té un número d'identificació (*numero_devolució*). D'elles ens interessa saber la data en que el client ha fet la devolució, a quina comanda pertany, quines són les caixes que s'han retornat, els motius de la devolució (poden ser tres: caixa trencada, malvestat producte i equivocació producte) i amb quin repartiment és va enviar la comanda.

2.2.10. Repartiments

Cadascun dels repartiments porta un número d'identificació (codi_repartiment). D'ell ens interessa saber amb quina ruta està relacionat, la data de creació i de lliurament del repartiment, el nombre total de caixes de mida gran que el compo-

[16] Mirar Annex 6.
[17] Mirar Annex 6.

sen, si està tancat (no accepta més caixes) o està obert (accepta més caixes) i si ja s'ha servit o no.

2.2.11. Rutes

Cadascuna de les rutes porta un número d'identificació (codi_ruta), els codis postals als que pertany la zona de la ruta, el nom del transportista que fa la ruta, la capacitat del camió en caixes de mida gran, ja que sabent el que ocupa una caixa de mida gran podem saber el que ocupa una caixa de mida mitja i petit (dues caixes mitjanes ocupen el lloc d'una gran, dues caixes petites el lloc d'una mitjana i per tan, quatre caixes petites el lloc d'una gran)[18] i les destinacions que formen l'itinerari de la ruta ordenats d'última destinació a primera destinació. Una destinació porta un número que indica quina posició ocupa dins la ruta i es defineixen com el domicili social del client associat a aquesta destinació.

2.2.12. Subrutes

Cadascuna de les subrutes porta un número d'identificació (codi_subruta), el nom del transportista que fa la subruta, la capacitat del camió en caixes de mida gran i les destinacions que formen l'itinerari de la subruta ordenats d'última destinació a primera destinació.

2.3. Guions

A partir de la descripció del sistema, identifiquem dos guions. El guió de l'operador logístic l'hem dividit en cinc apartats, ja que per la seva extensió és molt més clarificador dividir-lo en apartats. Cadascun dels apartats fa referència a una de les tasques que activa l'operador logístic.

2.3.1. Guió del Robot

El sistema passa la referència de la primera línia no enviada de la primera comanda no servida del repartiment al robot del passadís on es troba ubicada la columna amb el seus codi igual a la referència de la primera línia no enviada de la primera comanda no servida. El robot busca la columna amb el seu codi igual a la referència rebuda, es situa davant d'ella i quan la troba avisa al sistema que l'ha trobat indicant el codi de columna trobada. Un cop el sistema rep l'avís, cerca la referència en la primera línia de la comanda que s'està tractant i llegeix la quantitat de caixes que s'han de servir del producte. Un cop llegida la quantitat de caixes el sistema envia la quantitat llegida al robot que li ha enviat el missatge d'avís. El robot agafa la quantitat de caixes que li ha enviat el sistema de la part de sota de la columna que té al davant i les serveix. Quan el robot serveix les caixes envia al sistema un avis de caixes servides i aquest senyala la línia de la comanda com enviada. Quan el sistema té marcades totes les línies de la comanda com enviades (arriba al final de la comanda), el sistema marca la comanda com servida i ja pot començar a enviar les línies de la següent comanda. Quan el sistema té totes les comandes d'un repartiment marcades com servides (arriba al final del repartiment), marca el repartiment d'aquestes comandes com a servit. Cada cop que el robot serveixi una caixa el sistema ha de descomptar la caixa del palet al que pertany. El robot per servir les caixes sempre agafarà les caixes del palet de sota (què són les que ca-

[18] A l'hora d'indicar la capacitat del camió pot ser que aquesta no s'ajusti a un total de caixes de mida gran, es a dir, si per exemple el camió pot transportar un total de 120 caixes de mida gran i una de mitjana, nosaltres direm que la seva capacitat en caixes de mida gran és de 120 caixes i mitja, ja que dues caixes de mida mitjana fan una de mida gran, el mateix passa si és tractes d'una caixa de mida petita, aleshores direm que la capacitat del camió en caixes de mida gran és de 120 caixes i un quart, ja que 4 caixes de mida petita fan una de mida gran.

ducaran abans). Un cop s'acabi el palet de sota, el sistema avisarà al robot que col·loqui el palet de sobre a sota i demanarà a fàbrica la fabricació d'un nou palet que el robot col·locarà a sobre quan aquest li sigui servit. Quan un palet caduqui (la data de caducitat del palet coincideix amb la data del sistema) el sistema avisarà al robot pertinent de quin palet es tracta, demanarà a fàbrica la fabricació d'un nou palet de les mateixes característiques, el robot retornarà el palet a l'exterior de la cambra i col·locarà el nou palet al seu lloc quan fàbrica li serveixi.

2.3.2. Guió de l'Operador logístic.

2.3.2.1. Els Clients

Quan es fa un client nou el sistema administratiu envia al sistema el NIF del nou client. L'operador logístic consulta el llistat de nous clients, si hi ha clients nous tria un d'ells i mira quin codi postal i domicili social té. Cerca la ruta associada al seu codi postal, selecciona la ruta i finalment s'encarrega d'assignar-li una destinació determinada introduint el número que indicarà quina posició ocupa la destinació dins la ruta. Un cop afegit el nou client, el sistema canvia el número que indica quina posició ocupa la destinació dins la ruta de tots els clients que tenen un número més alt o igual que ell, incrementant-los en una unitat. De la mateixa manera, si un client es dona de baixa l'operador logístic l'eliminarà de la seva ruta i l'aplicació canviarà el número que indica quina posició ocupa la destinació dins la ruta de tots els clients que tenen un número més alt que ell, disminuint-los en una unitat.

2.3.2.2. Les rutes

Si el sistema divideixi una ruta de forma eventual, vol dir que el sistema crea tantes subrutes de la ruta mare com divisions ha fet (las divisions les fa per ordre de ruta). Quan el sistema crea subrutes, està dividint la ruta mare en dos subrutes: la primera subruta tindrà el mateix nom de transportista que la ruta mare, la mateixa capacitat del camió i els mateixos destins amb el mateix ordre que tenen associada una comanda en el mateix repartiment. La segona subruta ha de tenir totes les destinacions del repartiment que encara no tenen assignada cap comanda i el mateix nom del transportista que fa la ruta i la mateixa capacitat del camió en caixes de mida gran. Totes dues han de tenir un codi de subruta. Un cop creada la subruta, el sistema crearà un repartiment associat a la subruta. Un cop creat el repartiment, el sistema contarà el número de caixes que té el repartiment, si aquest número és igual o major que 60, avisarà a l'operador logístic i aquest introduirà el nom del transportista eventual i la capacitat del camió.[19] Si aquest fet es converteix en un continuo (a criteri de l'operador logístic), serà l'operador logístic el que divideixi la ruta de forma definitiva, tenint en compte que les dues rutes no poden compartir un mateix codi postal. Cadascuna de les divisions es convertirà en una nova ruta menys una d'elles que es considerarà una modificació de la ruta inicial. Per tant l'operador logístic crearà una nova ruta introduint els codis postals als que pertany la zona de la ruta, el nom del transportista que fa la ruta, la capacitat del camió en caixes de mida gran i les destinacions que formen l'itinerari de la ruta ordenats d'última destinació a primera destinació. I modificarà la ruta mare esborrant els codis postals i destinacions que ara són de la nova ruta.

[19] Aquí es va pensar que, com les subrutes creades queden guardades a la base de dades, abans de crear una subruta calia consultar si alguna de les subrutes creades ens podia servir. Això no surt a compte, ja que és molt difícil que una subruta creada coincideixi amb la que ens fa falta (mateixos clients amb les mateixes comandes, mateix transportista,…), amb la qual cosa l'aplicatiu perderia temps en la cerca de coincidencies obtenint un resultat nul. Per tant el millor és crear-les de nou sempre que les necessitem.

L'operador logístic també pot crear una nova ruta que no té res a veure amb una ruta anterior, perquè l'empresa a decidit ampliar les zones de repartiment. L'operador logístic també ha de poder modificar les dades d'una ruta, per si un transportista és dona de baixa i cal introduir un altre nom de transportista i una nova capacitat del camió, o perquè un transportista canvia de camió, aleshores només caldrà canviar, si cal, la capacitat del camió.

2.3.2.3. Els repartiments.

Quan hi ha un camió apunt per ser carregat l'operador logístic consulta els repartiments que s'han de lliurar aquest dia, selecciona el repartiment que ha de fer el camió i li demana al sistema que li passi als robots.

2.3.2.4. Les devolucions.

Si hi ha una devolució l'operador logístic crea una devolució introduint la data en que el client ha fet la devolució, *numero_comanda* de la comanda a la que pertany la devolució, quines són les caixes de la comanda que s'han retornat, els motius de la devolució i amb quin repartiment és va enviar la comanda. Un cop creada la devolució l'envia al sistema administratiu.

2.3.2.5. Les noves comandes

L'operador logístic consulta la llista de noves comandes i selecciona una. El sistema cerca la ruta assignada al client de la comanda seleccionada. Si la ruta no té associat cap repartiment pel dia en que s'ha de lliurar la comanda, els sistema avisa a l'operador logístic i aquest crea un nou repartiment assignant-li un codi_repartiment, la *data actual* (data de creació), la data de lliurament i a continuació afegeix la comanda. El sistema suma el número total de caixes al número de caixes del repartiment. Si la ruta ja té associat un repartiment, el sistema selecciona el repartiment i afegeix la comanda de forma ordenada. Quan el número de caixes d'un repartiment coincideix o sobrepassa la capacitat màxima del camió, l'operador logístic a de crear una subruta. Si el total de caixes de l'última comanda sobrepassa la capacitat màxima del camió, el sistema no l'afegeix al repartiment i la guarda com a comanda pendent. El sistema avisa a l'operador logístic i aquest crea un nou repartiment associat a la segona subruta i afegeix les comandes pertinents, on la primera comanda pot ser la primera comanda no afegida al repartiment anterior. Si torna a igualar la capacitat màxima del camió, torna a crear subrutes i repartiments associats a la subruta, així fins acabar amb totes les comandes.

2.4. Actors

A partir dels guions podem distingir 4 actors: **Operador logístic**, **Robot**, **Sistema de fàbrica i Instant temporal**. L'actor **Operador logístic** és la persona encarregada de fer un ús directe del sistema. L'actor **Robot** és qualsevol robot de la cambra encarregat de gestionar-la segons les ordres que li dona el sistema. Tots els **robots** de la cambra tenen les mateixes responsabilitats i per tant, actuen en ubicacions diferents dins la cambra, però desenvolupen les mateixes funcions. L'actor **Sistema de fàbrica** és el sistema de l'àrea de fabricació de l'empresa. L'actor **Instant temporal** és la *data actual* que ve marcada pel rellotge intern del sistema i que s'actualitza cada 24 hores.

2.5. Casos d'ús

A partir dels guions podem distingir 22 casos d'ús. D'aquests, com es pot observar en la taula següent, 16 són activats per una actor. Els 6 restants són activats per altres casos d'ús, és a dir mantenen una relació d'inclusió amb els casos d'ús que els activen. Aquest 6 casos d'ús s'han creat, o bé perquè algun cas d'ús tenia la mateixa seqüència d'accions dins del seus procés què un altre cas d'ús, i per tant, per simplicitat, era millor crear un cas d'ús on el seu procés fos la seqüència d'accions comuns a aquests casos d'ús, com en el cas d'ús 15.Modificar buit ubicació , o bé, perquè algun cas d'ús era massa llarg i s'ha pogut dividir en diferents casos d'ús, com en els casos d'ús 6.Seleccionar comanda i 7.Enviar línia de comanda, respecte al cas d'ús 5.Seleccionar repartiment, o bé, com en els casos d'ús 10.Descomptar caixes palet i 11.Guardar línia de comanda, respecte al cas d'ús 9.Servir caixes.

En la següent taula podem veure quins actors activen als diferents casos d'ús, i amb quins actors que no els activen i altres casos d'ús estan relacionats:

Casos d'ús	Actors activen	Casos d'ús relacionats	Actors relacionats
1.Afegir comanda	**Operador logístic**	2.Crear subrutes 3.Crear repartiment 17.Assignar ruta client	
2.Crear subruta		1.Afegir comanda	**Operador logístic**
3.Crear repartiment	**Operador logístic**	1.Afegir comanda	
4.Seleccionar repartiment	**Operador logístic**	5.Seleccionar comanda	
5.Seleccionar comanda		4.Seleccionar repartiment 6.Enviar línia de comanda	
6.Enviar línia de comanda		5.Seleccionar comanda	**Robot**
7.Cercar columna	**Robot**	9.Descomptar caixes palet 12.Tractar nou palet 15.Tractar caducitat	
8.Servir caixes	**Robot**	9.Descomptar caixes palet 10.Guardar línia de comandes	
9.Descomptar caixes palet		8.Servir caixes 7.Cercar columna	**Robot**
10.Guardar línia de comanda		8.Servir caixes	
11.Canviar palet de lloc	**Robot**	14.Modificar buit ubicació	
12.Tractar nou palet	**Sistema de fàbrica**	7.Cercar columna	**Robot**
13.Ubicar palet nou	**Robot**	14.Modificar buit ubicació	

14.Modificar buit ubicació		11.Canviar palet de lloc 13.Ubicar palet nou 16.Treure palet	
15.Tractar caducitat	**Instant temporal**	7.Cercar columna	**Robot**
16.Treure palet	**Robot**	14.Modificar buit ubicació	
17.Assignar ruta client	**Operador logístic**	1.Afegir comanda	
18.Eliminar client ruta	**Operador logístic**		
19.Crear ruta	**Operador logístic**		
20.Dividir ruta	**Operador logístic**		
21.Modificar ruta	**Operador logístic**		
22.Crear devolució	**Operador logístic**		

2.5.1. Relacions entre Actors i Casos d'ús

Pel que fa a les relacions entre els casos d'ús:

- Els casos d'ús 3.Crear repartiment i 17.Assignar ruta client, que es poden executar independentment del cas d'ús 1.Afegir comanda i tenen el mateix actor primari (**Operador logístic**) que el cas d'ús 1.Afegir comanda, extenen aquest cas d'ús, ja que quan es compleixen certes condicions en el cas d'ús 1.Afegir comanda[20], aquest executa els casos d'ús que l'extenen.

- Els casos d'ús 11.Canviar palet de lloc, 13.Ubicar palet nou i 16.Treure palet, inclouen el cas d'ús 14.Modificar buit ubicació, ja que aquest és un procés comú als tres casos d'ús que l'inclouen, tal i com s'havia explicat en l'apartat anterior.

En els següents dos apartats les relacions que s'expliquen també són d'inclusió, però, tal i com s'ha dit a l'apartat anterior, en aquest cas s'ha fet servir la relació d'inclusió per reduir la complexitat en el model dels casos d'ús següents:

- El cas d'ús 4.Seleccionar repartiment inclou els casos d'ús 5.Seleccionar comanda i 6.Enviar línia de comanda. En aquest cas, els dos casos d'ús que estan inclosos en el primer, s'activen de forma seqüencial, és a dir, primer s'activa el cas d'ús 5.Seleccionar comanda i després aquest activa el cas d'ús 6.Enviar línia de comanda.
- El cas d'ús 8.Servir caixes inclou els casos d'ús 9.Descomptar caixes palet i 10.Guardar línia de comanda. En aquest cas, els dos casos d'ús que estan inclosos en el primer, s'activen de forma paral·lela, és a dir no cal activar un dels dos casos d'ús per activar l'altre.

[20] Mirar l'apartat "Alternatives del procés i excepcions" del cas d'ús 1.Afegir comanda en l'apartat 1.5 "Documentació textual dels Casos d'ús" d'aquest capítol.

- El cas d'ús 1.Afegir comanda inclou al cas d'ús 2.Crear subrutes. En aquest cas el segon cas d'ús està inclòs en el primer, i només s'activarà quan l'activi el primer, no pot executar-se de forma independent. L'**Operador logístic** rebrà i retornarà informació al cas d'ús 2.Crear subrutes, però en cap moment serà el responsable de la seva activació.

Pel que far a les relacions entre els actors i els casos d'ús:

- A la taula de l'apartat anterior hem pogut veure quins actors activen a cadascun dels casos d'ús.

- L'actor **Robot** rep informació dels casos d'ús 6.Enviar línia de comanda, 9.Descomptar caixes palet, 12.Tractar nou palet i 15.Tractar caducitat

- Com hem comentat abans l'actor **Operador logístic** rep i envia informació al cas d'ús 2.Crear subrutes.

El diagrama de casos d'ús de l'annex 4 ens mostra les relacions que hi ha entre els casos d'ús i els casos d'ús amb els actors.

2.6. Documentació textual dels Casos d'ús

En aquest apartat es presenta la documentació textual de cadascun dels 23 casos d'ús mencionats en l'apartat anterior:

Cas d'ús: 1.Afegir comanda

Resum de la funcionalitat: Serveix per afegir una nova *comanda* a un *repartiment*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic.
Casos d'ús relacionats: 2.Crear subrutes, 3.Crear repartiment , 17.Assignar ruta client
Precondició: Hi ha una petició d'afegir *comanda*.
Postcondició: S'ha afegit una nova *comanda* a un *repartiment*.

Procés normal principal:

1. El sistema cerca a la base de dades les *comandes* que no tenen assignat cap *repartiment*.
2. El sistema mostra a l'**operador logístic** la llista de *comandes* no assignades a cap *repartiment*.
3. L'**operador logístic** selecciona una *comanda* i accepta.
4. El sistema cerca en la base de dades el *repartiment* assignat a la *ruta* a la que pertany el *client* que a fet la *comanda* amb *data de lliurament* igual a la *data de lliurament* de la *comanda* i amb l'estat obert.
5. El sistema suma el total de *caixes* del *repartiment* trobat al nombre de *caixes* de la *comanda* seleccionada.
6. El sistema comprova que el total de *caixes* del *repartiment* seleccionat no sobrepassa la *capacitat màxima del camió* encarregat de fer el *repartiment* seleccionat.
7. El sistema afegeix de forma ordenada la *comanda* seleccionada al *repartiment* trobat.
8. El sistema mostra a l'**operador logístic** el *repartiment* amb la *comanda* afegida.

9. L'**operador logístic** accepta la *comanda* afegida.
10. El sistema guarda el *repartiment* amb la *comanda* afegida a la base de dades.
11. El sistema acaba el procés

Alternatives del procés i excepcions:
3a. L'**administratiu** cancel·la el procés:
3a1. El sistema acaba el procés.
4a. El sistema troba un *repartiment* de la *ruta* a la que pertany el client de la *comanda* seleccionada amb la *data de lliurament* de la *comanda* seleccionada i amb l'estat tancat:
4a1. El sistema cerca a la base de dades la *subruta* de la *ruta* a la que pertany el *client*:
4a1a. El sistema no troba cap *subruta*:
4a1a1. El sistema crida al cas d'ús 2.Crear subrutes.
4a2. El sistema cerca a la base de dades el *repartiment* assignat a la *subruta* trobada.
4b. El sistema no troba cap *repartiment* amb la *data de lliurament* de la nova *comanda* assignat a la *ruta* a la que pertany el *client* que ha fet la *comanda*:
4b1. El sistema crida al cas d'ús 3.Crear repartiment.
4c. El sistema no troba cap *ruta* assignada al *client*:
4c1. El sistema crida al cas d'ús 17.Assignar ruta client.
6a. El nombre de *caixes* del *repartiment* sobrepassa la *capacitat màxima del camió*:
6a1. El sistema canvia el estat obert del *repartiment* pel de tancat.
6a2. El sistema guarda el *repartiment* a la base de dades amb l'estat de tancat.
6a3. El sistema crida al cas d'ús 2.Crear subrutes.
6b. El *repartiment* pertany a una *subruta* i el total de *caixes* del *repartiment* és més gran que 60:
6b1. El sistema presenta a **l'operador logístic** les dades de la *subruta*.
6b2. L'**operador logístic** modifica el nom del transportista i la capacitat del camió (si cal) i confirma les dades modificades.
6b3. El sistema modifica les dades de la *subruta*.
6b4. El sistema guarda la *subruta* modificada.
9a. L'**Operador logístic** no accepta la *comanda* afegida:
9a1. El sistema acaba el procés.

Qüestions que cal aclarir: L'excepció **4a1a** es dona quan el *repartiment* assignat a una *ruta* està tancat, però el nombre de *caixes* que el formen és igual a la *capacitat màxima del camió* i per tant fins aquest moment no ha calgut crear una *subruta*.
En l'excepció **4a** punt **4a2**, si trobem un *repartiment* que compleix les condicions, aquest no pot tenir mai l'estat de tancat, ja que si fos així voldria dir que la *comanda* del client ja ha estat introduïda[21] i per un altre costat, no pot tenir mai una altre *data de lliurament* en l'únic *repartiment* assignat que té que no sigui la del *repartiment* assignat a la *ruta* mare.
En el punt **6b2** es absurd que l'**Operador logístic** cancel·li el procés ja que algú ha de encarregar-se de repartir la *comanda*. Per això l'**Operador logístic** només té l'opció d'acceptar les dades un cop las ha modificat.

[21] Mirar Annex 4.

Cas d'ús: 2.Crear subrutes

Resum de la funcionalitat: Crea dues *subrutes* a partir de la *ruta* a la que pertany el client en el cas d'ús 1.Afegir comanda.
Paper dins el treball de l'usuari: bàsic.
Actors: Operador logístic.
Casos d'ús relacionats: 1.Afegir comanda
Precondició: Hi ha un *ruta* seleccionada i el cas d'ús 1.Afegir comanda ha activa el cas d'ús 2.Crear subrutes
Postcondició: Se han creat dues noves *subrutes.*

Procés normal principal:

1. El sistema crea la primera *subruta* assignant-li un codi, el nom del transportista i la *capacitat màxima del camió* iguals als de la *ruta* o *subruta* seleccionada al cas d'ús 1.Afegir comanda.
2. El sistema busca totes les *destinacions* del *repartiment* tancat que pertanyen a la *ruta* o *subruta* seleccionada.
3. El sistema assigna a la primera *subruta* creada totes les *destinacions* trobades.
4. El sistema crea un *repartiment* de la primera *subruta* creada assignant-li un codi, la *data actual*, la *data de lliurament* i el total de *caixes* del *repartiment* tancat de la *ruta* o *subruta* seleccionada, l'estat de tancat i el de no servit.
5. El sistema assigna les *comandes* assignades al *repartiment* tancat de la *ruta* o *subruta* seleccionada al *repartiment* creat.
6. El sistema crea la segona *subruta* assignant-li un codi, el nom del transportista i la *capacitat màxima del camió* iguals als de la *ruta* o *subruta* seleccionades.
7. El sistema busca totes les *destinacions* de la *ruta* o *subruta* seleccionada que no pertanyen al *repartiment* tancat.
8. El sistema assigna a la segona *subruta* creada totes les *destinacions* trobades.
9. El sistema crea un *repartiment* de la segona *subruta* creada assignant-li un codi, la *data actual*, la *data de lliurament* del *repartiment* tancat de la *ruta* o *subruta* seleccionada i el total de *caixes* igual a 0, l'estat d'obert i el de no servit.
10. El sistema mostra a l'**operador logístic** les dues *subrutes* creades juntament amb el *repartiment* assignat a la primera.
11. L'**operador logístic** accepta les dues *subrutes.*
12. El sistema guarda el *repartiment* creat per a la primera subruta a la base de dades com a *repartiment* de la primera *subruta*.
13. El sistema guarda la primera *subruta* creada a la base de dades com a *subruta* de la *ruta* seleccionada.
14. El sistema guarda el *repartiment* creat per a la segona subruta a la base de dades com a *repartiment* de la segona *subruta*.
15. El sistema guarda la segona *subruta* creada a la base de dades com a *subruta* de la *ruta* seleccionada.
16. El sistema guarda a la base de dades el *repartiment* tancat de la *ruta* seleccionada amb l'estat servit.
17. El sistema guarda a la base de dades la *ruta* seleccionada.
18. El sistema finalitza el procés.

Alternatives del procés i excepcions:

13a. La *ruta* seleccionada no és una *ruta* sinó una *subruta*:
- **12a1.** El sistema cerca a la base de dades la *ruta* a la que pertany la *subruta* seleccionada.
- **12a2.** El sistema guarda la primera *subruta* creada a la base de dades com a *subruta* de la *ruta* trobada.

15a. La *ruta* seleccionada no és una *ruta* sinó una *subruta*:

13a1. El sistema cerca a la base de dades la *ruta* a la que pertany la *subruta* seleccionada.
13a2. El sistema guarda la segona *subruta* creada a la base de dades com a *subruta* de la *ruta* trobada.
16a. La *ruta* seleccionada no és una *ruta* sinó una *subruta*:
16a1. El sistema esborra de la base de dades el *repartiment* de la *subruta* seleccionada.
17a. La *ruta* seleccionada no és una *ruta* sinó una *subruta*:
17a1. El sistema esborra de la base de dades la *subruta* seleccionada.

Qüestions que cal aclarir: En l'alternativa **16a** del procés principal, el sistema esborra la *subruta* selecciona a l'inici del procés principal ja que aquesta mai es portarà a terme. Aquesta *subruta* passa a equivaler a les dues *subrutes* que hem creat i per tant les que de moment si es portaran a terme. Així doncs la *subruta* inicialment seleccionada no cal guardar-la, estalviant espai en la base de dades, guanyant en claredat i evitant que en el moment de carregar el camió es carreguin les mateixes *comandes* més d'un cop. Per un altre costat les noves *subrutes* creades passaran a ser *subrutes* de la *ruta* a la que estava associada la *subruta* inicialment seleccionada, tal i com és pot veure a les alternatives **13a** i **15a** del procés principal[22].
En el pas **9** del procés normal principal, el sistema crea un *repartiment* buit de la segona *subruta*, ja que aquesta si en un futur té algun *repartiment* assignat ha de ser amb la mateixa *data de lliurament* que el de la ruta mare.
En el pas **10** del procés normal principal, l'**Operador logístic** accepta les rutes per avisar al sistema que ja las ha vist, però no amb l'opció de rebutjar-les, ja que no hi ha una altra opció en el moment de convertir una ruta o subruta en dues subrutes.
En el pas **16** del procés normal principal, el sistema canvia l'estat de no servit del *repartiment* de la *ruta* inicialment seleccionada pel de servit ja que quan el cas d'ús 4.Seleccionar repartiment mostri a l'**Operador logístics** els *repartiments* pendents de ser servits pels **robots,** només mostrarà els *repartiments* de les *subrutes* (què si tindran l'estat de no servit) de la *ruta* seleccionada inicialment. D'aquesta manera el sistema evitarà la duplicació de *repartiments* en el moment de ser servits.

Cas d'ús: 3.Crear repartiment

Resum de la funcionalitat: Serveix per crear un *repartiment*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic.
Casos d'ús relacionats: 1.Afegir comanda.
Precondició: Hi ha una petició de crear *repartiment*.
Postcondició: El *repartiment* està creat.

Procés normal principal:
1. El sistema demana a l'**operador logístic** que introdueixi la *data de lliurament* del *repartiment* que és vol crear.
2. L'**operador logístic** introdueix la data i la confirma.
3. El sistema cerca les *rutes* que no tenen un *repartiment* amb la *data de lliurament* igual a la data introduïda per l'**operador logístic.**
4. El sistema mostra a l'**operador logístic** la llista de *rutes* trobades.
5. L'**operador logístic** selecciona una de les *rutes* i confirma la selecció.

[22] Mirar Annex 4.

6. El sistema crea un nou *repartiment* de la *ruta* seleccionada assignant-li un codi, la *data actual*, la *data de lliurament* introduïda per l'**operador logístic**, l'estat d'obert i l'estat de no servit.
7. El sistema mostra a l'**operador logístic** el nou *repartiment*.
8. L'**operador logístic** accepta el nou *repartiment*.
9. El sistema guarda el nou *repartiment* de la *ruta* seleccionada a la base de dades.
10. El sistema acaba el procés.

Alternatives del procés i excepcions:

2a. L'**operador logístic** cancel·la el procés:
2a1. El sistema acaba el procés.

5a. L'**operador logístic** cancel·la el procés:
5a1. El sistema acaba el procés.

8a. L'**operador logístic** cancel·la el procés:
8a1. El sistema acaba el procés.

Cas d'ús: 4.Seleccionar repartiment

Resum de la funcionalitat: selecciona un *repartiment* no servit.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic
Casos d'ús relacionats: 5.Seleccionar comanda
Precondició: Hi ha una petició de seleccionar repartiment.
Postcondició: El *repartiment* no servit està seleccionat.

Procés normal principal:

1. El sistema llegeix de la base de dades tots els *repartiments* amb el camp servit amb valor no i amb *data de lliurament* igual a la *data actual*.
2. El sistema presenta la llista de *repartiments* no servits amb *data de lliurament* igual a la *data actual*.
3. L'**operador logístic** selecciona el *repartiment* que ha de fer el camió i l'accepta.
4. El sistema crida al cas d'ús 5.Seleccionar comanda

Alternatives del procés i excepcions:

1a. El sistema no ha trobat cap *repartiment* amb el camp servit amb valor no i amb *data de lliurament* igual a la *data actual*:
1a1. El sistema acaba el procés.

3a. L'**operador logístic** cancel·la el procés:
3a1. El sistema acaba el procés.

Cas d'ús: 5.Seleccionar comanda

Resum de la funcionalitat: Selecciona la primera *comanda* no servida del *repartiment* seleccionat al cas d'ús 4.Seleccionar repartiment.
Paper dins el treball de l'usuari: bàsic
Actors: Cap
Casos d'ús relacionats: 4.Seleccionar repartiment, 6.Enviar línia de comanda.
Precondició: La *comanda* no servida no està seleccionada, hi ha un *repartiment* seleccionat i el cas d'ús 4.Seleccionar repartiment ha activa el cas d'ús 5.Seleccionar comanda.
Postcondició: La *comanda* no servida està seleccionada.

Procés normal principal:

1. El sistema cerca la primera *comanda* del *repartiment* seleccionat en el cas d'ús 4.Seleccionar repartiment amb el camp servida amb valor no.
2. El sistema selecciona la *comanda* trobada.
3. El sistema crida al cas d'ús 6.Enviar línia de comanda

Alternatives del procés i excepcions:

1a. El sistema no troba cap *comanda* no servida:

- **1a1.** El sistema canvia el valor no pel valor si al camp servit del *repartiment* seleccionat al cas d'ús 4.Seleccionar repartiment.
- **1a3.** El sistema guarda el *repartiment* modificat a la base de dades.
- **1a4.** El sistema acaba el procés.

Cas d'ús: 6.Enviar línia de comanda

Resum de la funcionalitat: serveix per enviar una *línia de comanda* no enviada a un **robot**.
Paper dins el treball de l'usuari: bàsic
Actors: Robot
Casos d'ús relacionats: 5.Seleccionar comanda.
Precondició: La *línia de comanda* no està enviada, hi ha una *comanda* seleccionada i el cas d'ús 5.Seleccionar comanda ha activa el cas d'ús 6.Enviar línia de comanda.
Postcondició: La *línia de comanda* està enviada al **robot** corresponent.

Procés normal principal:

1. El sistema cerca la primera *línia de comanda* de la *comanda* seleccionada al cas d'ús 5.Seleccionar comanda amb el camp enviada amb valor no.
2. El sistema llegeix la referència de la *línia de comanda* trobada.
3. El sistema cerca a la base de dades la *prestatgeria* a la que pertany la *columna* amb el mateix codi que la referència enviada.
4. El sistema llegeix el codi de la *prestatgeria* trobada.
5. El sistema cerca el *passadís* al que pertany la *columna*.
6. El sistema llegeix el codi del *passadís* trobat.
7. El sistema envia al **robot** que té el mateix codi que el del *passadís* llegit, la referència i el codi de la *prestatgeria* llegits.

Alternatives del procés i excepcions:

1a. El sistema no troba cap *línia de comanda* no enviada:

- **1a1.** El sistema esborra el valor no del camp servida de la *comanda* seleccionada al cas d'ús 5.Seleccionar comanda.
- **1a2.** El sistema escriu el valor si en el camp servida de la *comanda* seleccionada al cas d'ús 5.Seleccionar comanda.
- **1a3.** El sistema guarda la *comanda* modificada a la base de dades.
- **1a4.** El sistema crida al cas d'ús 5.Seleccionar comanda.

Qüestions que cal aclarir: Al **robot** juntament amb la referència se li envia el codi de *prestatgeria*, per a que el procés de cerca del **robot** sigui més curt allargant la vida del robot, ja que aquest només cercarà en una de les dues *prestatgeries* que controla, és a dir, en la que es troba la *columna* amb el codi igual a la referència.

Cas d'ús: 7.Cercar columna

Resum de la funcionalitat: Serveix per a que un **Robot** és col·loqui davant d'una de les seves *columnes*.
Paper dins el treball de l'usuari: bàsic

Actors: Robot
Casos d'ús relacionats: cap
Precondició: El **robot** ha rebut les dades del sistema per realitzar la cerca de la *columna* i un missatge.
Postcondició: El **robot** està situat davant la *columna* trobada i ha llegit el missatge.

Procés normal principal:
1. El **robot** llegeix el codi de la *prestatgeria* que li ha enviat el sistema.
2. El **robot** busca la *prestatgeria* amb el mateix codi que ha llegit.
3. El **robot** llegeix la referència enviada pel sistema.
4. El **robot** busca la *columna* amb el codi igual a la referència en la *prestatgeria* trobada.
5. El **robot** es para davant de la *columna* trobada i envia al sistema un missatge de "columna trobada".
6. El sistema envia dades al **robot** segons la tasca que tingui que realitzar.

Alternatives del procés i excepcions:
6a. El **robot** ha de servir *caixes*:
- **6a1.** El sistema llegeix de la *línia de comanda* trobada al cas d'ús 6.Enviar línia de comanda el nombre de *caixes* que el robot ha de servir.
- **6a2.** El sistema envia al **robot** el nombre de *caixes* que ha de servir.
- **6a3.** El sistema acaba el procés.

6b. El **robot** ha de canviar un *palet* d'*ubicació*:
- **6b1.** El sistema envia al **robot** el codi d'*ubicació* "SOB" al **robot**.
- **6b2.** El sistema acaba el procés.

6c. El **robot** ha de ubicar un *palet* nou:
- **6c1.** El sistema cerca en la *columna* enviada al **robot** la *ubicació* de la *columna* amb el camp buit igual a si.
- **6c2.** El sistema llegeix la *ubicació* trobada.
- **6c3.** El sistema envia al **robot** la *ubicació* trobada i un missatge de "ubicar palet nou".
- **6c4.** El sistema acaba el procés.

6d. El **robot** ha de treure un *palet* de la cambra de congelació:
- **6d1.** El sistema envia la *data actual* al **robot**.
- **6d2.** El sistema acaba el procés.

Qüestions que cal aclarir: Després d'haver-se executat el cas d'ús 8, els **robots** saben el que han de fer pel tipus de dades que els envia el sistema. En el cas de l'excepció **6c** el sistema, a part de les dades, envia al **robot** un missatge de "ubicar *palet* nou". Això és així, perquè en el cas què la *ubicació* enviada pel sistema al **robot** sigui "SOB", el **robot** sinó rep alguna distinció per part del sistema, no sabrà si executar el cas d'ús 12 o el 14. El missatge de "ubicar palet nou", és la distinció que envia el sistema al **robot** per saber quin dels dos casos d'ús ha d'executar, en aquest cas el cas d'ús 14.
L'excepció **6b**, es produeix quan un *palet* s'ha acabat i s'ha de fer el canvi de lloc del *palet* de dalt al lloc de baix. Com el **robot**, el primer que farà amb aquest llocs és fer el canvi de lloc del *palet* de dalt, no cal modificar el valor del camp buit de la *ubicació* de baix, ja què tornarà a no estar buit en el moment que el **robot** faci el canvi de lloc del *palet*.
En la excepció **6d**, s'ha de tenir en compte que les dates de caducitat dels dos *palets* que es troben en una *columna*, mai coincidiran, ja què fàbrica no serveix un nou *palet* fins al dia següent del dia en que és fa una petició de fabricació de *palet*. L'únic moment en que podrien coincidir les dates de caducitat de dos *palets* d'una mateixa *columna*, és la vegada en que s'ompli la cambra per primer cop. Això

s'evitarà fent la carrega de la cambra en dos dies, el primer dia s'omple una *ubicació* i al segon l'altre, o bé, posant-se d'acord amb fàbrica perquè no serveixi els dos *palets* d'un mateix producte amb la mateixa data de caducitat. Per un altre costat, la mitjana de consum de *caixes* d'un producte determinat al llarg d'un dia és d'un *palet*, per tant és difícil que s'acabin el dos el mateix dia. En tot cas, si es dones aquesta situació, fàbrica no serveix més d'un *palet* del mateix producte en el mateix dia i per tant no hi pot haver cap problema.

Cas d'ús: 8.Servir caixes

Resum de la funcionalitat: Serveix per a que el **Robot** serveixi les *caixes* a partir de les dades enviades pel sistema en cas d'ús 7.Cercar columna.
Paper dins el treball de l'usuari: bàsic
Actors: Robot
Casos d'ús relacionats: 9.Descomptar caixes palet, 10.Guardar línia de comandes.
Precondició: El sistema ha enviat al **robot** el nombre de *caixes* que ha de servir.
Postcondició: Les *caixes* han estat servides i el robot ha enviat un missatge de "caixes servides" al sistema.

Procés normal principal:
1. El **robot** llegeix el nombre de *caixes* que li ha enviat el sistema.
2. El **robot** cerca en la *columna* que té al davant la *ubicació* "SOT".
3. El **robot** agafa el nombre de *caixes* llegides de la *ubicació* "SOT" de la *columna* i es dirigeix a la porta del seu *passadís*, on serveix les *caixes*.
4. El **robot** envia al sistema un missatge de "caixes servides".
5. El sistema crida al cas d'ús 9.Descomptar caixes palet i al cas d'ús 10.Guardar línia de comanda.

Alternatives del procés i excepcions:
Cap.

Cas d'ús: 9.Descomptar caixes palet

Resum de la funcionalitat: Serveix per a descomptar del *palet* al que pertanyen les *caixes* servides pel **Robot,** el nombre de *caixes* servides.
Paper dins el treball de l'usuari: bàsic
Actors:
Casos d'ús relacionats: 7.Cercar columna, 8.Servir caixes .
Precondició: El sistema ha rebut un missatge de "caixes servides" enviat pel **robot** en el cas d'ús 8.Servir caixes i el cas d'ús 8.Servir caixes ha activat el cas d'ús 9.Descomptar caixes palet.
Postcondició: Les *caixes* han estat descomptades del seu *palet*.

Procés normal principal:
1. El sistema busca a la base de dades el *palet* associat a la *ubicació* "SOT" de la *columna* amb el codi igual a la referència de la línia de *comandes* què ha trobat en el cas d'ús 6.Enviar línia de comanda.
2. El sistema llegeix el nombre total de *caixes* que té el *palet* trobat.
3. El sistema resta el nombre de *caixes* llegides de la *línia de comanda* al total de *caixes* llegides del *palet* trobat.
4. El sistema comprova que el resultat de la resta no sigui 0.
5. El sistema esborra el total de *caixes* del *palet* trobat.
6. El sistema escriu el resultat de la resta calculada en el total de *caixes* del *palet* trobat.

7. El sistema guarda en la base de dades el *palet* modificat.
8. El sistema acaba el procés.

Alternatives del procés i excepcions:
6a. El resultat de la resta és 0:
- **6a1.** El sistema llegeix el codi del *palet* trobat.
- **6a2.** El sistema envia un missatge al sistema de fàbrica amb el codi del *palet* que s'ha acabat perquè fabriqui un de nou.
- **6a3.** El sistema envia al **robot** que ha enviat el missatge de "caixes servides" en el cas d'ús 8.Servir caixes, el codi de la *prestatgeria* llegit al cas d'ús 6.Enviar línia de comanda, el codi de la *columna* de la *línia de comanda* trobada al cas d'ús 6.Enviar línia de comanda.
- **6a4.** El sistema crida al cas d'ús 7.Cercar columna, per cercar la *columna* on s'ha de fer el canvi d'*ubicació*.

Cas d'ús: 10. Guardar línia de comanda

Resum de la funcionalitat: Serveix per a guardar una *línia de comanda* enviada.
Paper dins el treball de l'usuari: bàsic
Actors:
Casos d'ús relacionats: 8.Servir caixes.
Precondició: La *línia de comanda* té el valor no en el camp enviada i el cas d'ús 8.Servir caixes ha activat el cas d'ús 10. Guardar línia de comanda.
Postcondició: La línia de *comanda* té el valor si en el camp enviada i ha estat guardada.

Procés normal principal:
1. El sistema esborra el valor no del camp enviada de la *línia de comanda* trobada al cas d'ús 6.Enviar línia de comanda.
2. El sistema escriu el valor si en el camp enviada de la *línia de comanda*.
3. El sistema guarda la *línia de comandes* modificada.
4. El sistema acaba el procés.

Alternatives del procés i excepcions:
Cap

Cas d'ús: 11.Canviar palet de lloc

Resum de la funcionalitat: serveix per canviar el *palet* de la *ubicació* de dalt d'una *columna* al lloc de baix, quan el *palet* de baix s'ha acabat.
Paper dins el treball de l'usuari: bàsic
Actors: Robot
Casos d'ús relacionats: 14.Modificar buit ubicació.
Precondició: El sistema ha enviat al **robot** la *ubicació* "SOB".
Postcondició: El *palet* està canviat de lloc i el sistema ha avisat ha fàbrica perquè fabriqui un nou *palet*.

Procés normal principal:
1. El **robot** llegeix la *ubicació* "SOB" que li ha enviat el sistema.
2. El **robot** busca la *ubicació* amb codi "SOB" de la *columna*.
3. El **robot** agafa el *palet* situat en la *ubicació* de dalt de la *columna* trobada.
4. El **robot** busca la *ubicació* amb codi "SOT" de la *columna*.
5. El **robot** col·loca el *palet* agafat en la *ubicació* de sota de la *columna* trobada.
6. El **robot** envia al sistema el codi de la *columna*, la *ubicació* "SOB" amb valor buit igual a si.

7. El sistema crida al cas d'ús 14.Modificar buit ubicació.

Alternatives del procés i excepcions:
Cap

Qüestions que cal aclarir: Quan un *palet* s'ha acabat i s'ha de fer el canvi d'*ubicació* del *palet* de dalt a la *ubicació* de baix. Com el **robot** no farà res, amb aquestes *ubicacions*, fins què no hagi fet el canvi d'*ubicació* del *palet* de dalt, no cal modificar el valor del camp buit de la *ubicació* de baix, ja què tornarà a no estar buit en el moment que el **robot** faci el canvi d'*ubicació* del *palet*. Per això el **robot** només envia la *ubicació* amb el camp buit igual a si al sistema.

Cas d'ús: 12.Tractar nou palet

Resum de la funcionalitat: serveix perquè el sistema tracti un *palet* nou enviat pel **sistema de fàbrica** a un **robot** determinat.
Paper dins el treball de l'usuari: bàsic
Actors: Sistema de fàbrica, robot.
Casos d'ús relacionats: 7.Cercar columna
Precondició: El **sistema de fàbrica** ha enviat el *palet* al **robot**.
Postcondició: El *palet* està tractat pel sistema.

Procés normal principal:
1. El **sistema de fàbrica** envia un missatge al sistema per avisar-lo de que ja ha lliurat el *palet* al **robot** amb el codi del *palet*, el número de *caixes* que conte i la data de caducitat.
2. El sistema llegeix el codi del *palet*.
3. El sistema cerca a la base de dades el codi de *palet* igual al codi enviat pel **sistema de fàbrica**.
4. El sistema llegeix el nombre total de *caixes* i la data de caducitat enviats pel **sistema de fàbrica**.
5. El sistema escriu el nombre de *caixes* i la data de caducitat enviades pel robot en el camp nombre de *caixes* del *palet* trobat.
6. El sistema cerca a la base de dades la *columna* on ha de ser col·locat el nou *palet*.
7. El sistema llegeix el codi de la *columna* trobada.
8. El sistema cerca a la base de dades la *prestatgeria* a la que pertany la *columna* trobada.
9. El sistema llegeix el codi de la *prestatgeria* trobada.
10. El sistema cerca el *passadís* al que pertany la *columna*.
11. El sistema llegeix el codi del *passadís* trobat.
12. El sistema envia al **robot** que té el mateix codi que el del *passadís* llegit, el codi de la *prestatgeria* i de la *columna* llegits.
13. El **robot** recull el *palet* de la seva porta.
14. El sistema crida al cas d'ús 7.Cercar columna per cercar la *columna* on s'ha d'ubicar el nou *palet*.

Alternatives del procés i excepcions:
Cap

Cas d'ús: 13.Ubicar palet nou

Resum de la funcionalitat: serveix per ubicar un nou *palet* enviat pel sistema de fàbrica en la seva *ubicació* dins la cambra.
Paper dins el treball de l'usuari: bàsic

Actors: Robot
Casos d'ús relacionats: 14.Modificar buit ubicació.
Precondició: El sistema ha enviat al **robot** el missatge "ubicar nou palet" i la *ubicació* del nou *palet* .
Postcondició: El *palet* està situat en la seva *ubicació* dins la cambra.

Procés normal principal:
1. El **robot** llegeix la *ubicació* que li ha enviat el sistema.
2. El **robot** busca la *ubicació* llegida de la *columna* on està situat.
3. El **robot** col·loca el *palet* a la *ubicació* trobada.
4. El **robot** envia al sistema el codi de la *columna*, la *ubicació* enviada pel sistema amb el camp buit igual a no.
5. El sistema crida al cas d'ús 14.Modificar buit ubicació.

Alternatives del procés i excepcions:
Cap

Cas d'ús: 14.Modificar buit ubicació

Resum de la funcionalitat: Modifica el camp buit de la *ubicació* d'una *columna* un cop el **robot** ha col·locat el *palet* que faltava.
Paper dins el treball de l'usuari: bàsic
Actors:
Casos d'ús relacionats: 11.Canviar palet de lloc, 13.Ubicar palet nou.
Precondició: El **robot** ha enviat el codi de la *columna*, la *ubicació* i el valor què ha de prendre el camp buit de la *ubicació* al sistema i el cas d'ús 11.Canviar palet de lloc o el cas d'ús 13.Ubicar palet nou han activat al cas d'ús 14.Modificar buit ubicació.
Postcondició: El camp buit de la *ubicació* de la *columna* enviat pel **robot** ha estat modificat i el lloc guardat.

Procés normal principal:
1. El sistema llegeix el codi de la *columna*, la *ubicació* enviada pel **robot** i el valor del camp buit.
2. El sistema cerca la *ubicació* enviada pel **robot** en la *columna* amb codi igual al enviat pel **robot**.
3. El sistema esborra el valor del camp buit de la *ubicació* trobat.
4. El sistema escriu el valor enviat pel **robot** en el camp buit de la *ubicació* trobada.
5. El sistema guarda la *ubicació* modificada en la base de dades.
6. El sistema acaba el procés.

Alternatives del procés i excepcions:
Cap

Cas d'ús: 15.Tractar caducitat

Resum de la funcionalitat: serveix perquè el sistema localitzi la *columna* on es troba un *palet* caducat dins la cambra.
Paper dins el treball de l'usuari: bàsic
Actors: Instant temporal.
Casos d'ús relacionats: 7.Cercar columna
Precondició: La *data actual* coincideix amb la data de caducitat d'un *palet*.
Postcondició: La *columna* del *palet* ha estat localitzada.

Procés normal principal:

1. El sistema cerca el primer *palet* amb data de caducitat igual a la *data actual*.
2. El sistema cerca la *ubicació* de la *columna* on es troba el *palet* caducat.
3. El sistema llegeix el codi de la *columna* i el codi de la *ubicació* on es troba el *palet*.
4. El sistema cerca la *prestatgeria* a la que pertany la *columna* trobada.
5. El sistema llegeix el codi de la *prestatgeria* trobada.
6. El sistema cerca el *passadís* al que pertany la *prestatgeria*.
7. El sistema llegeix el codi del *passadís* trobat.
8. El sistema envia un missatge al sistema de fàbrica amb el codi del *palet* caducat perquè fabriqui un de nou.
9. El sistema envia al **robot** que té el mateix codi que el del *passadís* llegit, el codi de la *prestatgeria* i de la *columna* llegits
10. El sistema crida al cas d'ús 7.Cercar columna per cercar la *columna* on es troba el *palet* caducat.

Alternatives del procés i excepcions:
Cap

Cas d'ús: 16.Treure palet

Resum de la funcionalitat: serveix per treure un *palet* caducat de dins de la cambra.
Paper dins el treball de l'usuari: bàsic
Actors: Robot
Casos d'ús relacionats: 14.Modificar buit ubicació.
Precondició: El sistema ha enviat al **robot** una data .
Postcondició: El *palet* esta fora de la cambra.

Procés normal principal:

1. El **robot** llegeix la data que li ha enviat el sistema.
2. El **robot** busca el *palet* amb la mateixa data de caducitat que la data que li ha enviat el sistema.
3. El **robot** treu el *palet* trobat de la seva *ubicació*.
4. El **robot** llegeix el codi de la ubicació i el codi de la *columna* d'on a tret el *palet*.
5. El **robot** envia al sistema el codi de la *columna* i el codi de la *ubicació* llegits amb el valor del camp buit igual a si.
6. El **robot** es dirigeix a la seva porta per servir el *palet*.
7. El sistema crida al cas d'ús 14.Modificar buit ubicació.

Alternatives del procés i excepcions:
Cap

Cas d'ús: 17.Assignar ruta client

Resum de la funcionalitat: serveix per assignar una *ruta* a un *nou client*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic.
Casos d'ús relacionats: Cap
Precondició: Hi ha una petició d'assignar *ruta* a un *nou client*.
Postcondició: El *nou client* té assignada una *ruta*.

Procés normal principal:

1. El sistema cerca a la base de dades els *clients* que no estan donats de baix i que no tenen assignada una *ruta*.

2. El sistema presenta la llista de *clients nous* a l'**operador logístic.**
3. L'**operador logístic** selecciona a un dels *clients nous* i accepta la selecció.
4. El sistema cerca les *destinacions* de la *ruta* amb un dels seus codis postals igual al codi postal del client.
5. El sistema presenta a l'**operador logístic** la llista de les *destinacions* trobades i li demana que introdueixi el domicili social i el número que vol que ocupi el domicili social del client dins la *ruta*.
6. L'**operador logístic** introdueix les dades demanades pel sistema i les confirma.
7. El sistema ordena les *destinacions* de la *ruta*.
8. El sistema guarda a la base de dades les *destinacions* de la *ruta* amb les modificacions fetes.
9. El sistema acaba el procés.

Alternatives del procés i excepcions:

2a. El sistema no ha trobat cap *client* nou:
  2a1. El sistema acaba el procés.

3a. L'**operador logístic** cancel·la el procés:
  3a1. El sistema acaba el procés.

6a. L'**operador logístic** cancel·la el procés:
  6a1. El sistema acaba el procés.

Qüestions que cal aclarir: Al punt **5** del procés normal principal a l'**Operador logístic** se li mostra el llistat de les *destinacions* ordenades perquè pugui escollir quina posició vol que ocupi la *destinació* del nou client.

Cas d'ús: 18.Eliminar client ruta

Resum de la funcionalitat: serveix per donar de baixa a un *client* de la seva *ruta*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic.
Casos d'ús relacionats: Cap
Precondició: Hi ha un petició d'eliminar a un *client* de la *ruta*.
Postcondició: El *client* no té cap *ruta* assignada.

Procés normal principal:

1. El sistema cerca a la base de dades els *clients* que s'han donat de baixa i què tenen una *destinació* assignada.
2. El sistema presenta la llista dels *clients* que s'han donat de baixa a l'**operador logístic.**
3. L'**operador logístic** selecciona a un dels *clients* i confirma la selecció.
4. El sistema esborra la *destinació* del *client* de la *ruta* que el sistema ha trobat a la base de dades.
5. El sistema reordena les *destinacions* de la *ruta*.
6. El sistema guarda a la base de dades les *destinacions* de la *ruta* amb les modificacions fetes.
7. El sistema acaba el procés.

Alternatives del procés i excepcions:

2a. El sistema no ha trobat cap *client* donat de baixa:
  2a1. El sistema acaba el procés.

3a. L'**operador logístic** cancel·la el procés:
  3a1. El sistema acaba el procés.

Cas d'ús: 19.Crear ruta

Resum de la funcionalitat: serveix per crear una nova *ruta*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic
Casos d'ús relacionats:
Precondició: Hi ha una petició de crear una *ruta* nova.
Postcondició: La *ruta* està creada.

Procés normal principal:
1. El sistema demana a **l'operador logístic** els codis postals als que pertany la zona de la *ruta*, el nom del transportista que fa la *ruta*, la capacitat del camió en *caixes* de mida gran.
2. L'**operador logístic** introdueix les dades que li ha demanat els sistema i les confirma.
3. El sistema crea la nova *ruta* amb les dades introduïdes per l'**operador logístic**.
4. El sistema mostra a l'**operador logístic** la nova *ruta* creada.
5. L'**operador logístic** accepta la nova *ruta*.
6. El sistema guarda la *ruta* creada a la base de dades.
7. El sistema acaba el procés.

Alternatives del procés i excepcions:
2a. L'**operador logístic** cancel·la el procés:
2a1. El sistema acaba el procés.
5a. L'**operador logístic** cancel·la el procés:
5a1. El sistema acaba el procés.

Cas d'ús: 20.Dividir ruta

Resum de la funcionalitat: serveix per dividir una *ruta* creant una nova *ruta* i modificant la primera.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic
Casos d'ús relacionats:
Precondició: Hi ha una petició de dividir *ruta*.
Postcondició: Se ha creat una nova *ruta* i una modificació de la primera.

Procés normal principal:
1. El sistema demana a **l'operador logístic** el nom del transportista que farà la nova *ruta* i la capacitat del camió en *caixes* de mida gran.
2. L'**operador logístic** introdueix les dades que li ha demanat els sistema i confirma les dades.
3. El sistema crea la nova *ruta* amb les dades introduïdes per l'**operador logístic** assignant-li un codi de *ruta*.
4. El sistema presenta la llista de *rutes* trobades a la base de dades.
5. L'**operador logístic** selecciona la *ruta* que vol dividir i confirma la selecció.
6. El sistema presenta la llista de codis postal de la *ruta* seleccionada, que el sistema a trobat a la base de dades.
7. L'**operador logístic** selecciona els codis postals que han de formar part de la nova *ruta* i confirma la selecció.
8. El sistema assigna els codis postals seleccionats a la nova *ruta* i els esborra de la *ruta* principal.
9. El sistema assigna a la nova *ruta* les *destinacions* que ha trobat a la base de dades que pertanyen als codis postals seleccionats.

10. El sistema ordena les *destinacions* de la nova *ruta* i de la *ruta* principal modificada.
11. El sistema presenta per pantalla la nova *ruta* i la *ruta* principal modificada amb les seves *destinacions* ordenades.
12. L' **operador logístic** confirma la nova *ruta* i la *ruta* principal modificada.
13. El sistema guarda a la base de dades la nova *ruta* i la *ruta* principal modificada.
14. El sistema acaba el procés.

Alternatives del procés i excepcions:
2a. L'**operador logístic** cancel·la el procés:
2a1. El sistema acaba el procés.
5a. L'**operador logístic** cancel·la el procés:
5a1. El sistema acaba el procés.
7a. L'**operador logístic** cancel·la el procés:
7a1. El sistema acaba el procés.
12a. L'**operador logístic** cancel·la el procés:
12a1. El sistema acaba el procés.

Cas d'ús: 21.Modificar ruta

Resum de la funcionalitat: Modifica les dades d'una *ruta*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic.
Casos d'ús relacionats: Cap
Precondició: Hi ha una petició de modificació de *ruta*.
Postcondició: La *ruta* està modificada.

Procés normal principal:
1. El sistema presenta a l'**operador logístic** un llistat amb totes les *rutes* que hi ha guardades a la base de dades.
2. L'**operador logístic** selecciona la *ruta* que vol modificar i confirma la selecció.
3. El sistema presenta a **l'operador logístic** les dades de la *ruta* seleccionada: número d'identificació de la *ruta* (codi_ruta), els codis postals als que pertany la zona de la *ruta*, el nom del transportista que fa la *ruta*, la capacitat del camió en *caixes* de mida gran i les *destinacions* que formen l'itinerari de la *ruta* ordenats d'última *destinació* a primera *destinació*.
4. L'**operador logístic** modifica les dades pertinents i les confirma.
5. El sistema busca totes les *subrutes* de la *ruta* seleccionada que tinguin *reparti-ments* assignats amb l'estat no servit.
6. El sistema substitueix les dades de les *subrutes* trobades que siguin iguals a les dades de la *ruta* seleccionada i les quals han estat modificades per l'**Operador logístic**.
7. El sistema substitueix les dades de la *ruta* seleccionada per les dades que ha modificat l'**Operador logístic**.
8. El sistema presenta per pantalla la *ruta* modificada.
9. L' **operador logístic** confirma la modificació.
10. El sistema guarda la *ruta* modificada juntament amb les seves *subrutes*.
11. El sistema acaba el procés.

Alternatives del procés i excepcions:
2a. L'**operador logístic** cancel·la el procés:
2a1. El sistema acaba el procés.
4a. L'**operador logístic** cancel·la el procés:
4a1. El sistema acaba el procés.

5a. El sistema no troba cap *subruta* de la *ruta* seleccionada que tingui *repartiments* assignat amb l'estat no servit:
- **5a1.** El sistema substitueix les dades de la *ruta* seleccionada per les dades que ha modificat l'**Operador logístic**.
- **5a2.** El sistema guarda la *ruta* modificada.
- **5a3.** El sistema acaba el procés.

9a. L'**operador logístic** cancel·la el procés:
- **9a1.** El sistema acaba el procés.

Qüestions que cal aclarir: Una *ruta* només podrà ser modificada quan no tingui assignat cap *repartiment* amb estat no servit.

Cas d'ús: 22.Crear devolució

Resum de la funcionalitat: serveix per crear una *devolució*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic
Casos d'ús relacionats:
Precondició: Hi ha una petició de crear *devolució*.
Postcondició: La *devolució* està creada.

Procés normal principal:
1. El sistema demana a **l'operador logístic** el NIF del *client* i la *data de lliurament* del *repartiment* o de la *comanda* a la que pertanyen les *caixes* que s'han retornat.
2. L' **Operador logístic** introdueix les dades que li ha demanat els sistema i les confirma.
3. El sistema busca a la base de dades les *línies de comanda* de la *comanda* del *client* amb NIF i *data de lliurament* iguals als introduïts per l'**operador logístic.**
4. El sistema mostra a l'**operador logístic** les dades de la *comanda*: número d'identificació de la *comanda* (*numero_comanda*), la data en que el *client* ha fet la *comanda*, la data en que s'ha de lliurar la *comanda*, el NIF, nom, domicili social, ciutat del domicili social i codi posta del *client* que ha fet la *comanda* i el total de *caixes* per mida gran amb la llista de les *línies de comanda* trobades.
5. L'**Operador logístic** selecciona les *línies de comanda* de les que s'ha retornat alguna *caixa* indicant quantes *caixes* s'han retornat, quins són els motius de la *devolució* i el codi de les *caixes* retornades per cadascuna de les *línies de comanda* seleccionades i confirma la selecció.
6. El sistema crea una *devolució* amb el codi de la *devolució*, la *data actual*, el total de *caixes* retornades i el número de *comanda* al qual pertanyen les *línies de comanda*; la referència, els motius de la *devolució*, el total de *caixes* retornades i els codis de les *caixes* per cadascuna de les *línies de comanda* seleccionades
7. El sistema li mostra a l'**operador logístic** la *devolució* creada.
8. L'**operador logístic** accepta la *devolució*.
9. El sistema guarda a la base de dades la *devolució* creada.
10. El sistema acaba el procés.

Alternatives del procés i excepcions:

3a. L'**operador logístic** cancel·la el procés:
- **3a1.** El sistema acaba el procés.

7a. L'**operador logístic** cancel·la el procés:
- **7a1.** El sistema acaba el procés.

10a. L'**operador logístic** cancel·la el procés:
- **10a1.** El sistema acaba el procés.

2.7. Especificació dels requisits de la interfície d’usuari:

Els únics usuaris que necessitaran una interfície d’usuari són el operadors logístics. Tots els usuaris coneixen a fons la seva feina, però no estan acostumats a fer servir ordinadors, ja que fins ara la feina a estat totalment manual. Per una altre banda cal tenir en compte que l’aplicació no s’assembla a cap aplicació de les que fins ara s’han fet servir dins l’empresa. La rotació del personal és molt baixa: un operador logístic pel torn de matí i un altre pel torn de tarda, amb la qual cosa no hi haurà masses usuaris diferents manipulant l’aplicació. No totes les funcions del sistema és faran molt sovint, com és el cas de crear rutes, afegir clients,..., aquestes funcions són més bé esporàdiques. Cal destacar que hi ha funcions que normalment sempre farà l’operari del matí i d’altres el de la tarda, encara que esporàdicament pot donar-se el cas que algun dels dos faci funcions que normalment fa l’altre. Això és degut a que el repartiment de comandes normalment és fa pel matí i la preparació de repartiments per la tarda. Un altre fet important que cal recalcar, és que la informació que crea l’operari de la tarda, la fa servir l’operari del mati.

Tot això ens porta a dues conclusions a l’hora de dissenyar les interfícies:

- Les interfícies han de ser el més intuïtives i simples possibles.

- Per iniciar el sistema cal una contrasenya per evitar intrusions d’altre personal de l’empresa que no té perquè utilitzar el sistema i pot malversar informació crítica, però no cal que aquesta contrasenya sigui diferent pels operadors logístics que fan servir l’aplicació, ja que són pocs usuaris i comparteixen la informació creada amb el nostre sistema

3. ANÀLISIS

3.1. Identificació de les classes d'entitat:

A partir de la documentació textual dels casos d'ús[23] identifiquem les possibles classes d'entitat següents[24]:

●**Cas d'ús:** 1.Afegir comanda.

Classes: *Client, Comanda, Repartiment, Ruta, Subruta,*

●**Cas d'ús:** 2.Crear subruta.

Classes: *Comanda, Destinacio, Repartiment*, Ruta*, Subruta*,.*

●**Cas d'ús:** 3.Crear repartiment.

Classes: *Repartiment*, Ruta*, Subruta*.*

●**Cas d'ús**: 4.Seleccionar repartiment.

Classes: *Repartiment*.*

●**Cas d'ús:** 5.Seleccionar comanda.

Classes: *Comanda*, Repartiment*.*

●**Cas d'ús:** 6.Enviar línia de comanda.

Classes: *Columna, Comanda*, Linia_comanda, Passadis, Prestatgeria.*

●**Cas d'ús:** 7.Cercar columna.

Classes: *Columna*, Linia_comanda*, Palet*, Prestatgeria*, Ubicacio .*

●**Cas d'ús:** 8.Servir caixes.

Classes: *Caixa?, Columna*, Ubicacio*.*

●**Cas d'ús:** 9.Descomptar caixes palet.

Classes:, *Columna*, Linia_comanda*, Palet*, Prestatgeria*, Ubicacio*.*

●**Cas d'ús:** 10.Guardar línia de comanda.

Classes: *Linia_comanda*.*

●**Cas d'ús:** 11.Canviar palet de lloc.
Classes: *Columna*, Palet*, Ubicacio*.*

[23] Mirar apartat 2.6 "Documentació textual dels casos d'ús" del capítol 2.
[24] Quan la possible classe d'entita està seguida per " ***** " vol dir que aquesta es repeteix en més d'un cas d'ús i quan està seguida per " **?** " vol dir que no és clar que sigui realment una classe d'entitat. Si les possibles classes d'entitat no estan seguides per res vol dir que realment són identificades com a classes d'entitat.

●**Cas d'ús:** 12.Tractar nou palet.

Classes: *Columna*, Palet*, Passadis*, Prestatgeria*.*

●**Cas d'ús:** 13.Ubicar palet nou.

Classes: *Columna*, Palet*, Ubicacio*.*

●**Cas d'ús:** 14.Modificar buit ubicació.

Classes: *Columna*, Ubicacio*.*

●**Cas d'ús:** 15.Tractar caducitat.

Classes: *Columna*, Palet*, Passadis*, Prestatgeria*, Ubicacio*.*

●**Cas d'ús:** 16.Treure palet.

Classes: *Columna*, Palet*, Ubicacio*.*

●**Cas d'ús:** 17.Assignar ruta client.

Classes: *Client*, Destinacio*, Ruta*.*

●**Cas d'ús:** 18.Eliminar client ruta.

Classes: *Client*, Destinacio*, Ruta*,*

●**Cas d'ús:** 19.Crear ruta.

Classes: *Ruta*.*

●**Cas d'ús:** 20.Dividir ruta.

Classes: *Destinacio*, Ruta*.*

●**Cas d'ús:** 21.Modificar ruta.

Classes: *Ruta*.*

●**Cas d'ús:** 22.Crear devolució.

Classes: *Caixa?*, Comanda*, Devolucio, Repartiment*.*

Totes les classes que hem identificat tenen atributs tal i com es pot veure a la documentació textual dels casos d'ús[25]. Pel que fa a la possible classe d'entitat *Caixa* amb tots els seus atributs, ens adonem que és important pel sistema administratiu i de fàbrica de l'empresa, però per al nostre sistema l'únic que ens interessa és la quantitat de *caixes* dels *palets*, les *comandes* i els *repartiments*, els codis de les *caixes* per crear les *devolucions* i les mides de les *caixes* per poder comptabilitzar-les. Pel que fa a la quantitat de *caixes* serà un atribut de cada una de les classes d'entitat citades abans, i pel que fa al codi de les *caixes* i la seva mida considerarem que són els dos atributs que formen la classe d'entitat *Caixa*.

[25] Mirar Capítol 2 apartat 2.2 "Descripció detallada dels principals objectes del sistema" per més claredat.

Per tant la primera llista de classes d'entitats és: *Caixa, Client, Columna, Comanda, Destinacio, Devolucio, Linia_comanda, Palet, Passadís, Prestatgeria, Repartiment, Ruta, Subruta i Ubicacio.*

3.2. Identificació de les classes de frontera:

Tots els casos d'ús que són activats per l'**Operador logístic** tenen una classe de frontera en comú: *Menu*. Aquesta classe és la que fa servir l' **Operador logístic** per fer les peticions de les diferents opcions del sistema. Correspon al menú de l'aplicació.
A partir de la documentació textual dels casos d'ús[26] identifiquem les possibles classes de frontera següents:

●**Cas d'ús:** 1.Afegir comanda.

Classes:

FinestraComandesNoAssignadesRepartiment: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat de les *comandes* que no tenen assignat cap *repartiment* quan **l'Operador logístic** fa una petició d'afegir *comanda*. La *FinestraComandesNoAssignadesRepartiment* conté camps per especificar el número d'identificació de la *comanda*, la data en que el client ha fet la *comanda*, la data en que s'ha de lliurar la *comanda*, quin client ha fet la *comanda* i el total de caixes per mida gran de cadascuna de les *comandes* que apareixen al llistat, un boto (o un altre control) per acceptar la *comanda* seleccionada prèviament i un altre per cancel·lar el procés.

FinestraRepartiment: Finestra que mostra en l'*EstacioOperadorLogistic* el *repartiment* on s'ha afegit la comanda seleccionada mitjançant la *FinestraComandesNoAssignadesRepartiment*. La *FinestraRepartiment* conté camps per especificar tots els atributs d'un *repartiment*, els atributs de la *comanda* afegida, el número d'identificació de la *comanda*, la data en que el *client* ha fet la *comanda*, un boto (o un altre control) per acceptar el *repartiment* creat i un altre per cancel·lar el procés.

FormulariDadesSubruta: Aquesta classe fa referència a l'excepció **6a** del procés normal principal. Formulari que es fa servir per poder modificar el nom del transportista i la capacitat del camió (si cal) de la *subruta* seleccionada. Aquest formulari se li presenta a l'**Operador logístic** en l'*EstacioOperadorLogistic* quan aquest ha de modificar una *subruta*. El *FormulariDadesSubruta* conté un camp modificable amb el nom del transportista per poder introduir el nou nom del transportista, un camp modificable amb la capacitat màxima del camió per poder introduir una nova capacitat i un boto (o un altre control) per confirmar la data introduïda.

●**Cas d'ús:** 2.Crear subruta.

Classes:
FinestraSubrutesCreades: Finestra que mostra en l'*EstacioOperadorLogistic* les dues *subrutes* creades juntament amb el nou *repartiment* tancat de la primera *subruta*. La *FinestraSubrutesCreades* conté camps per especificar tots els atributs del nou *repartiment* tancat, de la primera *subruta* i de la segona, i un boto (o un altre control) per acceptar.

[26] Mirar apartat 2.6 "Documentació textual dels casos d'ús" del capítol 2.

●**Cas d'ús:** 3.Crear repartiment.

Classes:

FormulariDataLliurament: Formulari que es fa servir per donar la *data de lliurament* d'un *repartiment*. Aquest formulari se li presenta a l'**Operador logístic** en l'*EstacioOperadorLogistic* quan fa una petició de crear *repartiment*. El *FormulariDataLliurament* conté un camp per introduir la *data de lliurament*, un boto (o un altre control) per confirmar la data introduïda i un altre per cancel·lar el procés.

FinestraRutesRepartimentNoCreat: Finestra que mostra en l'*EstacioOperadorLogistic* a l'**Operador logístic** un llistat de les *rutes* que no tenen cap repartiment assignat amb la mateixa data de lliurament introduïda al *FormulariDataLliurament*. La *FinestraRutesRepartimentNoCreat* conté camps per especificar el número d'identificació de la *ruta* i els codis postals als que pertany la zona de la *ruta* de cadascuna de les *rutes* que apareixen al llistat, un boto (o un altre control) per acceptar la *ruta* seleccionada prèviament i un altre per cancel·lar el procés.

FinestraNouRepartiment: Finestra que mostra en l'*EstacioOperadorLogistic* el *repartiment* creat pel sistema a partir de la selecció feta mitjançant la *FinestraRutesRepartimentNoCreat*. La *FinestraNouRepartiment* conté camps per especificar el codi del nou repartiment, la *data actual*, la data de lliurament introduïda per l'operador logístic mitjançant el *FormulariDataLliurament*, l'estat obert i l'estat de no servit, un boto (o un altre control) per acceptar el *repartiment* creat i un altre per cancel·lar el procés.

●**Cas d'ús**: 4.Seleccionar repartiment.

Classes:

FinestraLlistatRepartiments: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat dels *repartiments* no servits amb *data de lliurament* igual a la *data actual* quan **l'Operador logístic** fa una petició de seleccionar *repartiment*. La *FinestraLlistatRepartiments* conté camps per especificar el número d'identificació del *repartiment*, , el número d'identificació de la *ruta* o *subruta* a la que esta assignat el *repartiment* i els codis postals als que pertany la zona de la *ruta* o *subruta* de cadascuna dels repartiments que apareixen al llistat, la data de lliurament dels repartiments, un boto (o un altre control) per acceptar el *repartiment* seleccionat prèviament i un altre per cancel·lar el procés.

●**Cas d'ús:** 6.Enviar línia de comanda.

Classes:

MissatgeLiniaDeComanda: Missatge que el sistema envia a un determinat **Robot**. El contingut del missatge consta de:

- La referència d'una *línia de comanda* que anteriorment ha estat llegida pel mateix sistema d'una *línia de comanda* pertanyent a una *comanda* assignada al *repartiment* seleccionat per l'**Operador logístic** mitjançant la *FinestraLlistatRepartiments.*

- I el codi de la *prestatgeria* a la que pertany la *columna* amb el mateix codi que la referència enviada.

●**Cas d'ús:** 7.Cercar columna

Classes:

MissatgeColumnaTrobada: Missatge que un determinat **Robot** envia al sistema per avisar-lo que ja ha trobat la columna que tenia que cercar. El contingut del missatge és "Columna trobada".

MissatgeNombreCaixes: Missatge que el sistema envia a un determinat **Robot** per notificar-li quantes *caixes* a de servir. El contingut del missatge és el nombre de *caixes* que ha de servir el **Robot.**

MissatgeCanviUbicacio: Missatge que el sistema envia a un determinat **Robot** per avisar-lo que ha de canviar un *palet* d'*ubicació* (el de dalt a sota). El contingut del missatge és "SOB" què és la *ubicació* on es troba el *palet* que ha de canviar a la *ubicació* de sota ("SOT")**.**

MissatgeUbicarPaletNou: Missatge que el sistema envia a un determinat **Robot** per avisar-lo que ha d'ubicar un nou *palet*. El contingut del missatge és la *ubicació* on ha de anar el nou *palet* i "Ubicar palet nou".

MissatgeTreurePalet: Missatge que el sistema envia a un determinat **Robot** per notificar-li que ha de treure els *palets* caducats de dins la cambra. El contingut del missatge és la *data actual***.**

●**Cas d'ús:** 8.Servir caixes.

Classes:

MissatgeCaixesServides: Missatge que un determinat **Robot** envia al sistema per avisar-lo que ja ha servit el nombre de *caixes* que el sistema li va enviar mitjançant el *MissatgeNombreCaixes*. El contingut del missatge és "Caixes servides".

●**Cas d'ús:** 11.Canviar palet de lloc.

Classes:

MissatgeCanviUbicacióFet: Missatge que un determinat **Robot** envia al sistema per avisar-lo que ja ha canviat d'ubicació el *palet* que el sistema li va notificar mitjançant el *MissatgeCanviUbicacio*. El contingut del missatge és el codi de la *columna* on es troba la *ubicació* i "SOB" amb valor buit igual a si.

●**Cas d'ús:** 12.Tractar nou palet.

Classes:

MissatgePaletLLiurat: Missatge que envia el **Sistema de Fàbrica** al sistema per avisar-lo de que ja ha lliurat el *palet* al **robot**. El contingut del missatge és el codi del *palet*, el número de *caixes* que conte i la data de caducitat.

MissatgePrestageriaColumna: Missatge que el sistema envia al **Robot** que té el mateix codi que el *passadís* llegit per notificar-li en quina *columna* ha d'ubicar el *palet* nou. El contingut del missatge és el codi de la *prestatgeria* i de la *columna* on ha d'anar ubicat el nou *palet*.

●**Cas d'ús:** 13.Ubicar palet nou.

Classes:

MissatgePaletNouUbicat: Missatge que un determinat **Robot** envia al sistema per avisar-lo que ja ha ubicat el nou *palet* a la *ubicació* que li va notificar el sistema mitjançant el *MissatgeUbicarPaletNou*. El contingut del missatge és el codi de la *columna* on es troba la *ubicació*, la *ubicació* que li va enviar el sistema amb el camp buit igual a no.

●**Cas d'ús:** 15.Tractar caducitat.

MissatgePrestatgeriaColumna: Missatge que el sistema envia al **Robot** que té el mateix codi que el *passadís* llegit per notificar-li de quina *columna* ha de treure el *palet* caducat. El contingut del missatge és el codi de la *prestatgeria* i de la *columna* on ha d'anar ubicat el nou *palet*. Aquesta classe de frontera és la mateixa que hi ha a la classe d'ús 12.Tractar nou palet.

●**Cas d'ús:** 16.Treure palet.

Classes:

MissatgePaletTret: Missatge que un determinat **Robot** envia al sistema per avisar-lo que ja ha tret el *palet* caducat amb la data de caducitat que li va notificar el sistema mitjançant el *MissatgeTreurePalet*. El contingut del missatge és el codi de la *columna* on es troba la *ubicació*, la *ubicació* que li va enviar el sistema amb el camp buit igual a si.

●**Cas d'ús:** 17.Assignar ruta client.

Classes:

FinestraLlistatClientsNous: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat dels clients que no estan donats de baixa i que no tenen assignada una *ruta* a **l'Operador logístic** quan fa una petició d'assignar *ruta* a nou client. La *FinestraLlistatClientsNous* conté camps per especificar el NIF, el nom, el domicili social, la població i els codi postal de la població de cadascun dels clients que apareixen al llistat, un boto (o un altre control) per acceptar el client seleccionat prèviament i un altre per cancel·lar el procés.

FormulariLlistatDestinacions: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat de las *destinacions* trobades a l'**Operador logístic** quan fa una petició d'assignar *ruta* a nou client, i alhora serveix de formulari per a que l'**Operador logístic** assigni el nou client a la *ruta*. La *FinestraFormulariLlistatDestinacions* conté camps per especificar el domicili social i el número que indica quina posició ocupa dins la *ruta* cadascuna de les *destinacions* que surten al llistat, un camp per introduir el domicili social del nou client i un altre per introduir el número que ha d'ocupar el domicili social del nou client dins la *ruta*, un boto (o un altre control) per acceptar les dades introduïdes i un altre per cancel·lar el procés.

●**Cas d'ús:** 18.Eliminar client ruta.

Classes:

FinestraLlistatClientsBaixa: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat dels clients que estan donats de baixa i tenen assignada una *ruta* a **l'Operador logístic** quan fa una petició d'eliminar a un client d'una *ruta*. La *FinestraLlistatClientsBaixa* conté camps per especificar el NIF, el nom, el domicili social, la població i els codi postal de la població de cadascun dels clients que apareixen al llistat, un boto (o un altre control) per acceptar el client seleccionat prèviament i un altre per cancel·lar el procés.

●**Cas d'ús:** 19.Crear ruta.

Classes:

FormulariDadesRuta: Formulari que es fa servir per poder crear una nova *ruta*. Aquest formulari es presenta a l'**Operador logístic** en l'*EstacioOperadorLogistic* quan aquest fa una petició de crear *ruta*. El *FormulariDadesRuta* conté diferents camps per introduir els codis postals als que pertany la zona de la *ruta*, el nom del transportista que farà la *ruta* i la *capacitat màxima* del camió *en caixes de mida gran*, un boto (o un altre control) per confirmar les dades introduïdes i un altre per cancel·lar el procés.

FinestraNovaRuta: Finestra que mostra en l'*EstacioOperadorLogistic* la nova *ruta* creada pel sistema a partir de les dades introduïdes mitjançant el *FormulariDadesRuta*. La *FinestraNovaRuta* conté camps per especificar el codi de la nova *ruta*, el nom del transportista, la *carrega màxima* del camió en *caixes de mida gran* i els codis postals associats a la *ruta*, un botó (o un altre control) per acceptar la nova *ruta* creada i un altre per cancel·lar el procés.

●**Cas d'ús:** 20.Dividir ruta.

Classes:

FormulariNovaRuta: Formulari que es fa servir per poder crear una nova *ruta*. Aquest formulari es presenta a l'**Operador logístic** en l'*EstacioOperadorLogistic* quan aquest fa una petició de dividir *ruta*. El *FormulariNovaRuta* conté diferents camps per introduir el nom del transportista que farà la *ruta* i la *capacitat màxima* del camió *en caixes de mida gran*, un boto (o un altre control) per confirmar les dades introduïdes i un altre per cancel·lar el procés.

FinestraLlistatRutes: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat de les *rutes* a **l'Operador logístic** després de introduir les dades al *FormulariNovaRuta*. La *FinestraLlistatRutes* conté camps per especificar el codi, el nom del transportista, la *carrega màxima* del camió en *caixes de mida gran* de cadascuna de les *rutes* que apareixen al llistat, un boto (o un altre control) per acceptar la *ruta* seleccionada prèviament i un altre per cancel·lar el procés.

FinestraLlistatCodisPostals: Finestra que mostra en l'*EstacioOperadorLogistic* un llistat dels codis postals associats a la *ruta* que anteriorment **l'Operador logístic** ha seleccionat mitjançant la *FinestraLlistatRutes*. La *FinestraLlistatCodisPostals* conté un camp per especificar la numeració de cadascun dels codis postals que apareixen al llistat, un boto (o un altre control) per acceptar els codis postals seleccionats i un altre per cancel·lar el procés.

FinestraDivisioRuta: Finestra que mostra en l'*EstacioOperadorLogistic* la nova *ruta* creada pel sistema i la modificació de la *ruta* seleccionada. La *FinestraDivisoRuta* conté camps per especificar el codi de la nova *ruta*, el nom del transportista, la *carrega màxima* del camió en *caixes de mida gran* i els codis postals associats a les dues *rutes*, un botó (o un altre control) per acceptar la divisió de la *ruta* seleccionada i un altre per cancel·lar el procés.

●**Cas d'ús:** 21.Modificar ruta.

Classes:
FinestraLlistatRutes: Aquesta classe de frontera, és la mateixa que apareix al cas d'ús 20.Dividir ruta, amb la diferència que s'activa quan **l'Operador logístic** fa una petició de modificar *ruta*.

FormulariModificarRuta: Formulari que es fa servir per poder modificar les dades d'una *ruta*. Aquest formulari es presenta a l'**Operador logístic** en l'*EstacioOperadorLogistic* quan aquest ha seleccionat una ruta mitjançant la classe *FinestraLlistatRutes*. El *FormulariModificarRuta* conté diferents camps per modificar el nom del transportista, la *capacitat màxima* del camió *en caixes de mida gran*, els codis postals i destinacions assignades a la ruta, un boto (o un altre control) per confirmar les dades introduïdes i un altre per cancel·lar el procés.

FinestraRutaModificada: Finestra que mostra en l'*EstacioOperadorLogistic* la *ruta* modificada pel sistema a partir de les dades introduïdes mitjançant el *FormulariModificarRuta*. La *FinestraRutaModificada* conté camps per especificar el codi de la *ruta*, el nom del transportista, la *carrega màxima* del camió en *caixes de mida gran*, els codis postals i destinacions associades a la *ruta*, un botó (o un altre control) per acceptar la nova *ruta* modificada i un altre per cancel·lar el procés.

●**Cas d'ús:** 22.Crear devolució.

Classes:

FormulariDevolucio: Formulari que es fa servir per poder introduir el NIF del client i la *data de lliurament* de les *caixes* retornades pel client. Aquest formulari es presenta a l'**Operador logístic** en l'*EstacioOperadorLogistic* quan aquest ha fa una petició de crear *devolució*. El *FormulariDevolucio* conté diferents camps per introduir el NIF del client i la *data de lliurament* de la *comanda* a la que pertanyen les *caixes* retornades, un boto (o un altre control) per confirmar les dades introduïdes i un altre per cancel·lar el procés.

FinestraFormulariComanda: Finestra que es fa servir per escollir les *línies de comanda* de la *comanda* trobada pel sistema a partir de les dades introduïdes per l'**Operador logístic** mitjançant la *FinestraFormulariDevolucio* i a les que pertanyen les *caixes* retornades pel client. Un cop seleccionades les *línies de comanda*, aquesta classe de frontera permet introduir a l'**Operador logístic** els codis de les *caixes* retornades, els motius de la *devolució* i quantes *caixes* s'han retornat per cadascuna de les *línies de comanda* seleccionades. A part, també conté un boto (o un altre control) per confirmar les dades introduïdes i un altre per cancel·lar el procés.

FinestraDevolucióCreada: Finestra que mostra en l'*EstacioOperadorLogistic* la *devolució* creada pel sistema a partir de les dades introduïdes mitjançant la *FinestraFormulariComanda*. La *FinestraDevolucióCreada* conté camps per especi-

ficar el codi de la *devolució*, la *data actual*, el total de *caixes* retornades i el *numero de comanda* al qual pertanyen les *línies de comanda*, la referència, els motius de la *devolució*, el total de *caixes* retornades i els codis de les *caixes* per cadascuna de les *línies de comanda* seleccionades, un botó (o un altre control) per acceptar *devolució* creada i un altre per cancel·lar el procés.

3.3. Identificació de les classes de control:

Tots els casos d'ús que són activats per l'**Operador logístic** tenen una classe de control en comú: *GestorMenu*. La classe de frontera *Menu* passa l'opció que selecciona l'**Operador logístic** a la classe GestorMenu, que crida la classe de control principal de cadascun del casos d'ús.
A partir de la documentació textual dels casos d'ús[27] identifiquem una classe de control per cadascun dels casos d'ús. Totes aquestes classes tenen la mateixa funció dins del seu cas d'ús, són les classes de control principal, responsables de cridar a les classes de frontera pertinents i a les classes d'entitat. Així doncs, en la següent taula és mostra quines són les classes de control per cadascun dels casos d'ús:

Casos d'ús	Classes de control	Casos d'ús	Classes de control
1.Afegir comanda	*AdiccioComanda*	12.Tractar nou palet	*TractamentNouPalet*
2.Crear subruta	*CreacioSubruta*	13.Ubicar palet nou	*UbicacioPaletNou*
3.Crear repartiment	*CreacioRepartiment*	14.Modificar buit ubicació	*ModificacioBuitUbicació*
4.Seleccionar repartiment	*SeleccioRepartiment*	15.Tractar caducitat	*TractamentCaducitat*
5.Seleccionar comanda	*SeleccioComanda*	16.Treure palet	*ExtraccioPalet*
6.Enviar línia de comanda	*EnviamentLiniaDeComanda*	17.Assignar ruta client	*AssignacioRutaClient*
7.Cercar columna	*SeleccioColumna*	18.Eliminar client ruta	*EliminacioClientRuta*
8.Servir caixes	*ServeiCaixes*	19.Crear ruta	*CreacioRuta*
9.Descomptar caixes palet	*SubstraccioCaixesPalet*	20.Dividir ruta	*DivisioRuta*
10.Guardar línia de comanda	*GuardantLiniaComanda*	21.Modificar ruta	*ModificacioRuta*
11.Canviar palet de lloc	*CanviPaletDeLloc*	22.Crear devolució	*CreacioDevolucio*

3.4. Modelat d'interaccions entre classes: diagrames de seqüències:

Mitjançant els diagrames de seqüències comprovem si falta algun objecte o algun comportament en l'especificació del sistema. Els refinaments que hem fet en el casos d'ús 1.Afegir comanda estan indicats en negreta i cursiva.

●**Cas d'ús:** 1.Afegir comanda.

En aquest cas d'ús, fent el diagrama de seqüències del procés normal principal ens adonem que hem de millorar el comportament, així doncs les variacions fetes seran:

[27] Mirar apartat 2.6 "Documentació textual dels casos d'ús" del capítol 2.

- El pas **7** s'ha de produir un cop l'**Operador logístic** ha acceptat que la *comanda* seleccionada sigui afegida al seu *repartiment*, per tant el pas **7** es produirà després que l'**Operador logístic** hagi acceptat que la *comanda* seleccionada pell sigui afegida al *repartiment* seleccionat pel sistema (*FinestraRepartiment*).
- Els passos **9** i **10** (**7** i **8** en el cas d'ús refinat), s'han rescrit per donar una major claredat.
- S'ha afegit una nova alternativa o excepció al procés normal principal (**2a**), ja que no s'havia tingut en compte que pot passar que quan el sistema cerca *comandes* que no tenen assignat cap *repartiment* no en trobi cap. Si això passes, tal i com s'indica en **2a**, el sistema acabaria el procés.

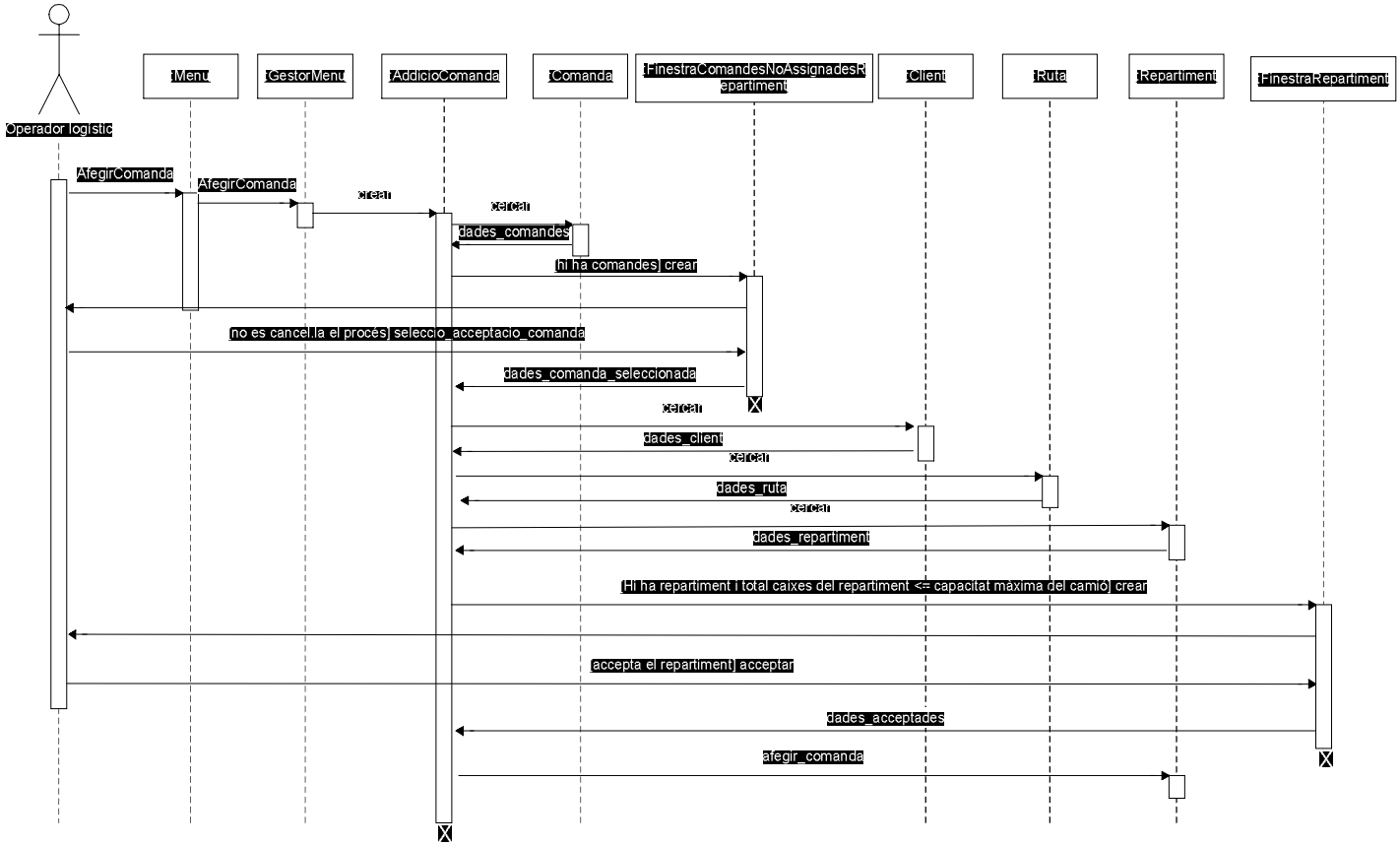


Diagrama de seqüències del procés normal principal del cas d'ús 1

Així doncs el cas d'ús 1 refinat seria el següent:

Cas d'ús: 1.Afegir comanda

Resum de la funcionalitat: Serveix per afegir una nova *comanda* a un *repartiment*.
Paper dins el treball de l'usuari: bàsic
Actors: Operador logístic.
Casos d'ús relacionats: 2.Crear subrutes, 3.Crear repartiment , 17.Assignar ruta client
Precondició: Hi ha una petició d'afegir *comanda*.
Postcondició: S'ha afegit una nova *comanda* a un *repartiment*.

Procés normal principal:

1. El sistema cerca a la base de dades les *comandes* que no tenen assignat cap *repartiment*.
2. El sistema mostra a l'**operador logístic** la llista de *comandes* no assignades a cap *repartiment*.
3. L'**operador logístic** selecciona una *comanda* i accepta.
4. El sistema cerca en la base de dades el *repartiment* assignat a la *ruta* a la que pertany el *client* que a fet la *comanda* amb *data de lliurament* igual a la *data de lliurament* de la *comanda* i amb l'estat obert.
5. El sistema suma el total de *caixes* del *repartiment* trobat al nombre de *caixes* de la *comanda* seleccionada.
6. El sistema comprova que el total de *caixes* del *repartiment* seleccionat no sobrepassa la *capacitat màxima del camió* encarregat de fer el *repartiment* seleccionat.
7. El sistema mostra a l'**operador logístic** el *repartiment* ***trobat*** amb la *comanda* ***seleccionada***
8. L'**operador logístic** accepta ***el repartiment trobat amb la comanda seleccionada.***
9. ***El sistema afegeix de forma ordenada la comanda seleccionada al repartiment trobat.***
10. El sistema guarda el *repartiment* amb la *comanda* afegida a la base de dades.
11. El sistema acaba el procés

Alternatives del procés i excepcions:

2a. ***No hi ha comandes sense cap repartiment assignat***:
- **2a1.** ***El sistema acaba el procés.***

3a. L'**administratiu** cancel·la el procés:
- **3a1.** El sistema acaba el procés.

4a. El sistema troba un *repartiment* de la *ruta* a la que pertany el client de la *comanda* seleccionada amb la *data de lliurament* de la *comanda* seleccionada i amb l'estat tancat:
- **4a1.** El sistema cerca a la base de dades la *subruta* de la *ruta* a la que pertany el *client*:
  - **4a1a.** El sistema no troba cap *subruta*:
    - **4a1a1.** El sistema crida al cas d'ús 2.Crear subrutes.
- **4a2.** El sistema cerca a la base de dades el *repartiment* assignat a la *subruta* trobada.

4b. El sistema no troba cap *repartiment* amb la *data de lliurament* de la nova *comanda* assignat a la *ruta* a la que pertany el *client* que ha fet la *comanda*:
- **4b1.** El sistema crida al cas d'ús 3.Crear repartiment.

4c. El sistema no troba cap *ruta* assignada al *client*:
- **4c1.** El sistema crida al cas d'ús 17.Assignar ruta client.

6a. El nombre de *caixes* del *repartiment* sobrepassa la *capacitat màxima del camió*:
- **6a1.** El sistema canvia el estat obert del *repartiment* pel de tancat.
- **6a2.** El sistema guarda el *repartiment* a la base de dades amb l'estat de tancat.
- **6a3.** El sistema crida al cas d'ús 2.Crear subrutes.

6b. El *repartiment* pertany a una *subruta* i el total de *caixes* del *repartiment* és més gran que 60:
- **6b1.** El sistema presenta a **l'operador logístic** les dades de la *subruta*.
- **6b2.** L'**operador logístic** modifica el nom del transportista i la capacitat del camió (si cal) i confirma les dades modificades.
- **6b3.** El sistema modifica les dades de la *subruta*.
- **6b4.** El sistema guarda la *subruta* modificada.

8a. L'**Operador logístic** no accepta la *comanda* afegida:

8a1. El sistema acaba el procés.

Qüestions que cal aclarir: L'excepció **4a1a** es dona quan el *repartiment* assignat a una *ruta* està tancat, però el nombre de *caixes* que el formen és igual a la *capacitat màxima del camió* i per tant fins aquest moment no ha calgut crear una *subruta*.
En l'excepció **4a** punt **4a2**, si trobem un *repartiment* que compleix les condicions, aquest no pot tenir mai l'estat de tancat, ja que si fos així voldria dir que la *comanda* del client ja ha estat introduïda[28] i per un altre costat, no pot tenir mai una altre *data de lliurament* en l'únic *repartiment* assignat que té que no sigui la del *repartiment* assignat a la *ruta* mare.
En el punt **6b2** es absurd que l'**Operador logístic** cancel·li el procés ja que algú ha de encarregar-se de repartir la *comanda*. Per això l'**Operador logístic** només té l'opció d'acceptar les dades un cop las ha modificat.

Fent el diagrama de seqüències de la nova excepció **2a**, descobrim una nova classe de frontera: *FinestraComandaNoTrobada*. Aquesta finestra avisa a l'**Operador logístic** que el sistema no ha trobat cap comanda.

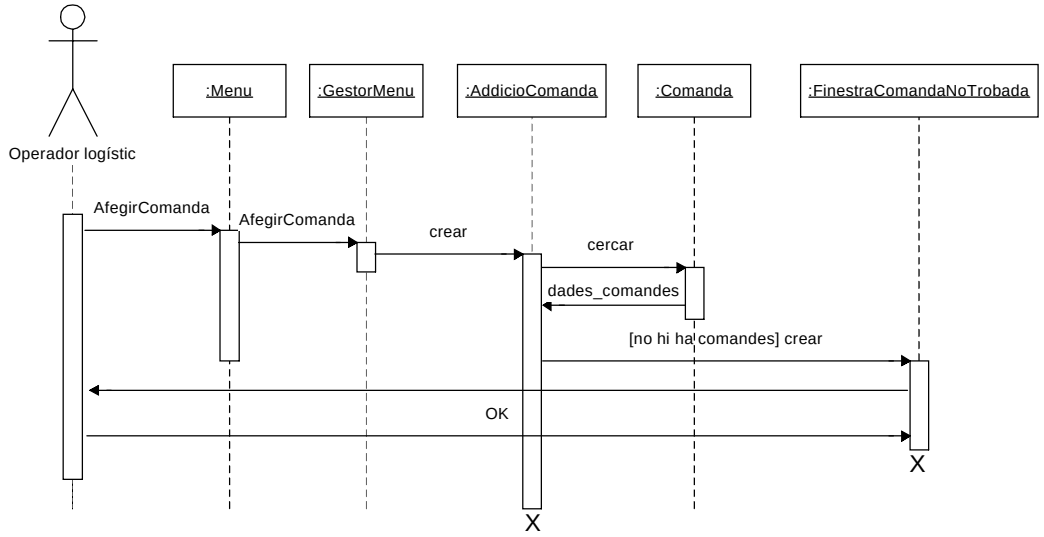


Diagrama de seqüències de l'excepció 2a del cas d'ús 1

La classe de frontera *FinestraComandaNoTrobada* també la trobarem en el cas d'ús 5.Seleccionar comanda en la seva excepció **1a**, ja que el sistema no ha trobat la *comanda* i en conseqüència aquest cancel·la el procés, com passa en l'excepció del cas d'ús 1.Afegir comanda. El seu desenvolupament serà el mateix que el que es mostra en el diagrama de seqüències de l'excepció **2a** del cas d'ús 1.Afegir comanda.

Fent el diagrama de seqüències de l'excepció **3a** o bé de la **8a**, descobrim una nova classe de frontera: *FinestraProcesCancellat*. Aquesta finestra avisa a l'**Operador logístic** que s'ha cancel·lat el procés.

[28] Mirar Annex 5.

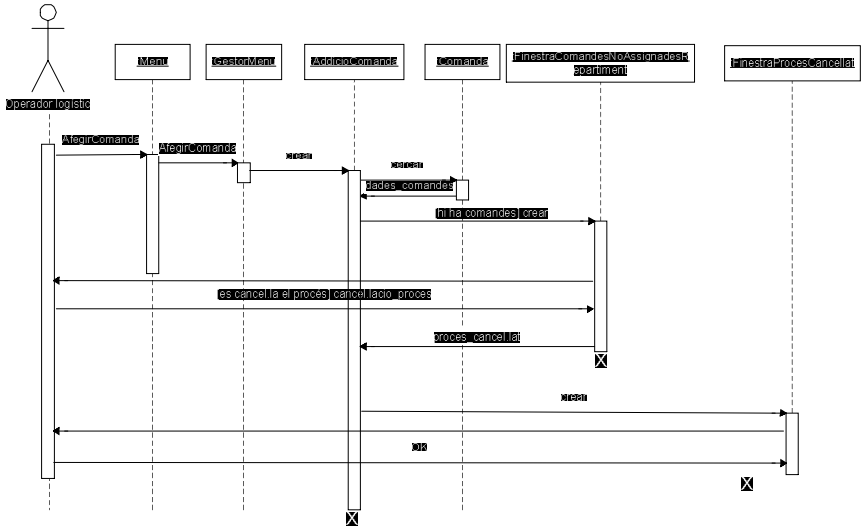
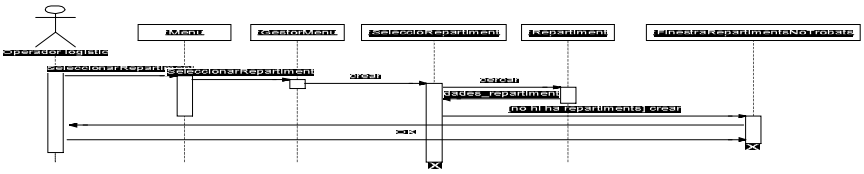


Diagrama de seqüències de l'excepció 3a del cas d'ús 1

La classe de frontera *FinestraProcesCancellat* serà comú a tots els casos d'ús en que en alguna de les seves excepcions l'**Operador logístic** cancel·la el procés. El seu desenvolupament serà el mateix que el que es mostra en el diagrama de seqüències de l'excepció **3a** del cas d'ús **Cas d'ús:** 1.Afegir comanda.

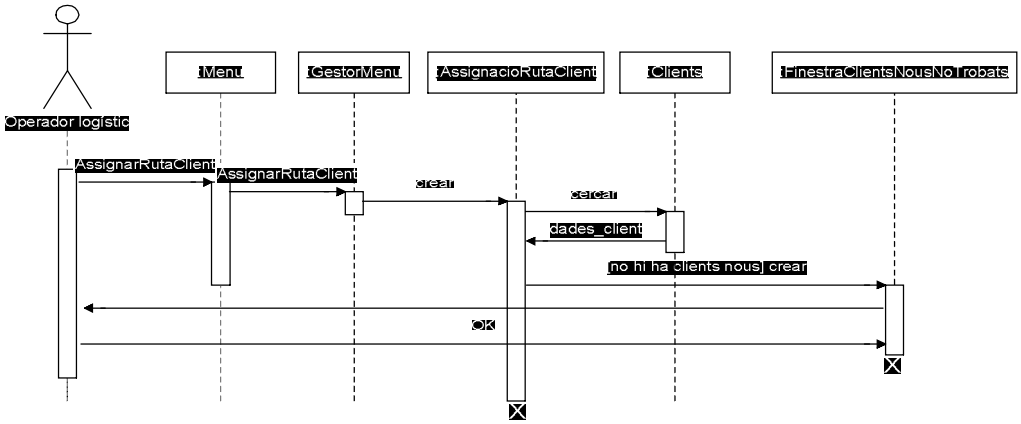
●**Cas d'ús**: 4.Seleccionar repartiment.

Fent el diagrama de seqüències de l'excepció **1a**, descobrim una nova classe de frontera: *FinestraComandaNoTrobada*. Aquesta finestra avisa a l'**Operador logístic** que el sistema no ha trobat cap comanda.



●**Cas d'ús:** 17.Assignar ruta client.

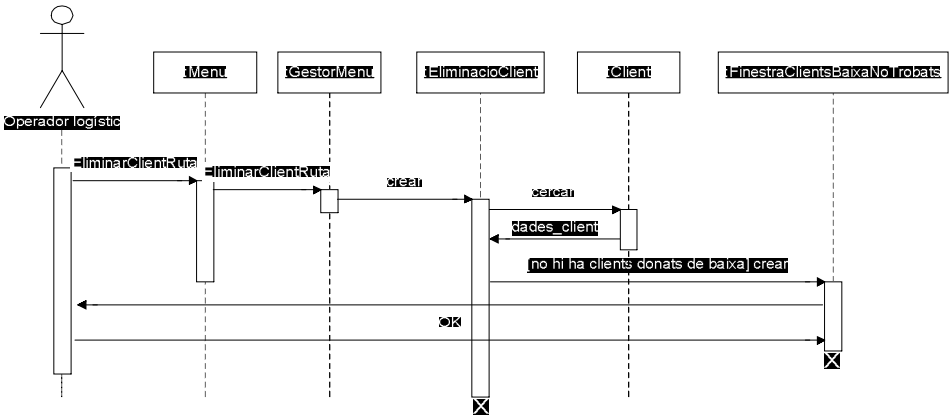
Fent el diagrama de seqüències de l'excepció **2a**, descobrim una nova classe de frontera: *FinestraClientsNousNoTrobats*. Aquesta finestra avisa a l'**Operador logístic** que no hi ha clients nous.



- **Cas d'ús:** 18.Eliminar client ruta.

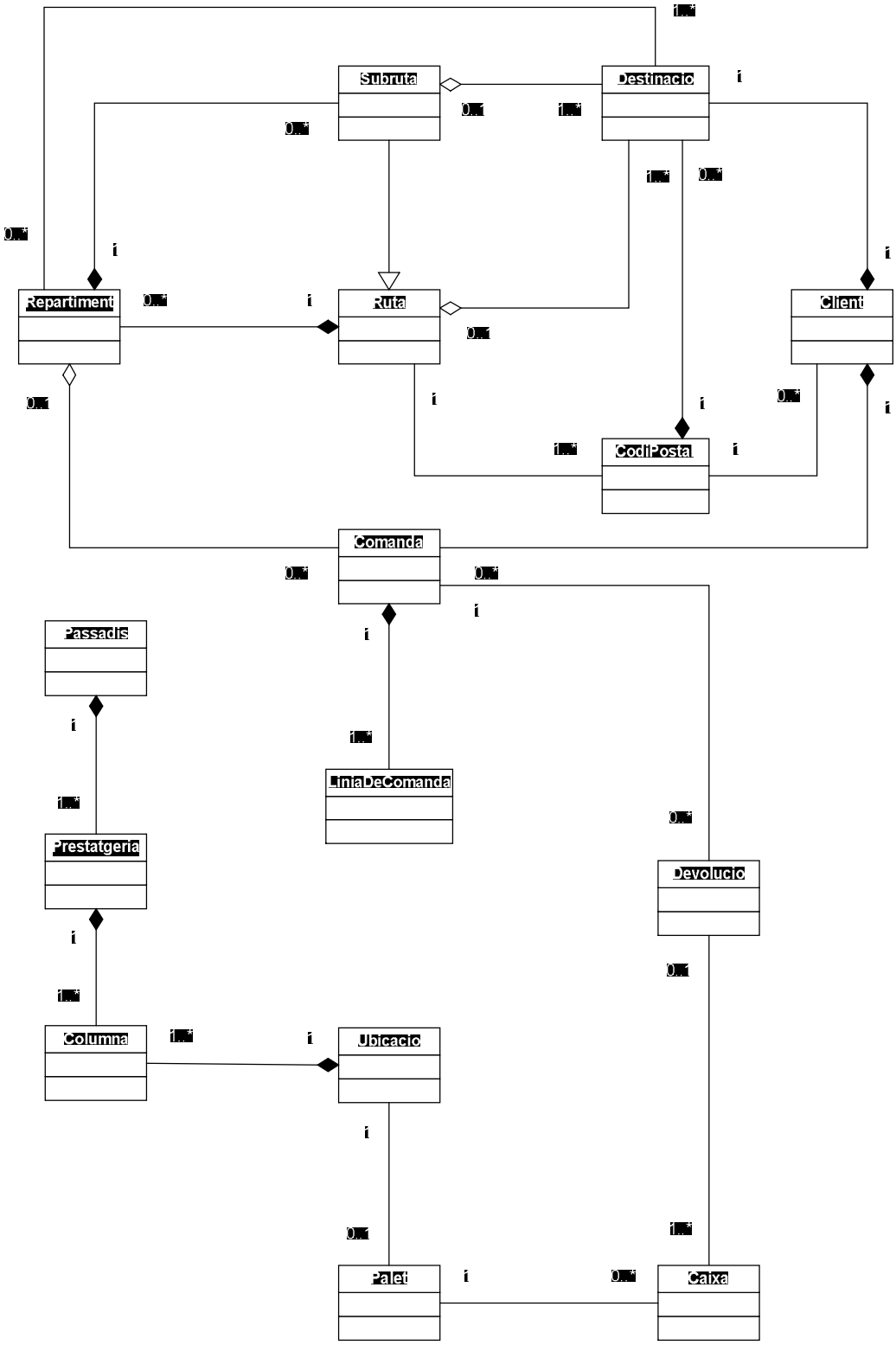
Classes:

Fent el diagrama de seqüències de l'excepció **2a**, descobrim una nova classe de frontera: *FinestraClientsBaixaNoTrobats*. Aquesta finestra avisa a l'**Operador logístic** que no hi ha clients donats de baixa.



3.5. Identificació d’associacions entre classes d’entitat:

Mitjançant el següent diagrama estàtic de les classes d’entitat podem veure quines són les associacions entre les diferents classes d’entitat:



En el diagrama estàtic podem veure que hem creat una nova classe d’entitat: *CodiPostal*. Aquesta classe té un únic atribut, el número que representa el codi postal

d'una zona determinada, però l'hem convertir en classe ja que moltes de les classes d'entitat accedeixen a ella i sobretot perquè qualsevol *ruta* té assignat un conjunt de codis postals.

3.6. Identificació dels atributs de les classes d'entitat:

●*Classe Caixa*: codi_caixa (integer), mida_caixa (string).

- *codi_caixa*: número que identifica a cadascuna de les caixes.
- *mida_caixa*: mida de la caixa. Aquest atribut només pot prendre un d'aquests tres valors: gran, mitjana o petita.

●*Classe Client*: NIF (string), nom (string), rao_social (string), poblacio (string), donat_de_baix (bolea).

-*NIF*: NIF del client.
-*nom*: nom del client.
-*rao_social*: adreça on es troba el negoci del client.
-*poblacio*: població on es troba el negoci del client.
-*donat_de_baixa*: atribut que indica si el client s'ha donat de baixa o no.

●*Classe Codi_postal: codi_postal(integer).*

- *codi_postal: codi postal de la zona on està situat el negoci del client*

●*Classe Columna*: codi_columna (integer).

- *codi_columna:* número que identifica a cadascuna de les *columnes*.

●*Classe Comanda*: codi (integer), data_comanda (data), data_lliurament (data), total_caixes (integer), servida(bolea).

-*codi*: número que identifica a cadascuna de les *comandes*.
-*data_comanda*: data en la que el client fa la *comanda*.
-*data_lliurament*: data en que han de ser lliurades les *caixes* de la *comanda* al client.
-*total_caixes*: total de *caixes* de la *comanda* en *caixes* de mida gran.
-*servida*: atribut que indica si la *comanda* ja ha estat lliurada al transportista o no.

●*Classe Destinacio*: posició (integer), rao_social (string).

-*posicio*: posició que ocupa la raó social del client dins la *ruta* que té assignada.
-*rao_social*: raó social del client.

●*Classe Devolució*: numero_devolucio (integer), data_devolucio (data), motius_devolucio (string).

-*numero_devolucio*: número que identifica a cadascuna de les *devolucions*.
-*data_devolucio*: data en la que el client fa la *devolució*.
-*motius_devolucio*: motius pels quals el client retorna les caixes. Aquest atribut només pot prendre un d'aquests tres valors: caixa trencada, malvestat producte o equivocació producte.

●*Classe Linia_comanda*: referencia_producte (integer), mida_caixa (string), total_caixes_linia_comanda (integer), enviada (bolea).

-*referencia_producte*: número que identifica a cadascuna de les *línies de comanda* d'una mateixa *comanda*. Aquest número coincideix amb la referència del producte que contenen les *caixes*.
-*mida_caixa*: mida de les *caixes* que pertanyen a la *línia de comanda*. Aquesta mida només pot ser gran, mitjana o petita.
-*total_caixes_linia_comanda*: total de *caixes* de la línia de *comanda.*
-*enviada*: atribut que indica si la *línia de comanda* ja ha estat lliurada al transportista o no.

●*Classe Palet*: codi_palet (integer), total_caixes_palet (integer), data_caducitat (data).

-*codi_palet*: número que identifica a cadascun dels *palets*.
-*data_caducitat*: data en la que les *caixes* que conte el *palet* caduquen.
-*total_caixes_palet*: total de *caixes* que conte el *palet.*

●*Classe Passadis*: codi_passadis (integer).

-*codi_passadis*: número que identifica a cadascun dels *passadissos*. Aquesta identificació és la mateixa que identifica als **robots** de cadascun dels *passadissos*.

●*Classe Prestatgeria*: codi_prestatgeria (integer).

-*codi_prestatgeria*: número que identifica a cadascuna de les *prestatgeries*.

●*Classe Repartiment*: codi_repartiment (integer), data_creacio (data), data_lliurament (data), total_caixes (integer), obert (bolea), servit (bolea).

-*codi_repartiment*: número que identifica a cadascun dels *repartiments*.
-*data_creacio*: data en la que es crea el repartiment.
-*data_lliurament*: data en que han de ser lliurades les *comandes* que formen el *repartiment*. Aquesta data coincideix amb la *data de lliurament* de les comandes del *repartiment*.
-*total_caixes*: total de *caixes* del *repartiment* en *caixes* de mida gran.
-*obert*: estat en el que al *repartiment* encara se li poden assignar més *comandes*.
-*servit*: atribut que indica si el *repartiment* ja ha estat lliurada al transportista o no.

●*Classe Ruta*: codi_ruta (integer), nom_transportista (string), capacitat_camio (integer).

-*codi_ruta*: número que identifica a cadascun de les *rutes*.
-*nom_transportista:* nom del transportista encarregat de fer la *ruta*.
-*capacitat_camio*: *capacitat màxima del camió* del transportista.

●*Classe Subruta*: codi_subruta (integer), nom_transportista (integer), capacitat_camio (integer).

-*codi_subruta*: número que identifica a cadascun de les *subrutes*.
-*nom_transportista:* nom del transportista encarregat de fer la *subruta*.

-*capacitat_camio*: *capacitat màxima del camió* del transportista.

●*Classe Ubicacio*: codi_ubicacio (string), buida (bolea).

-*codi_ubicacio*: paraula que identifica a cadascun de les dues *ubicacions* possibles. Aquestes només poden ser SOT o SOB.
-*buida:* atribut que indica si la *ubicació* conté algun palet (buida) o no.

3.7. Paquets d'anàlisi i de serveis:

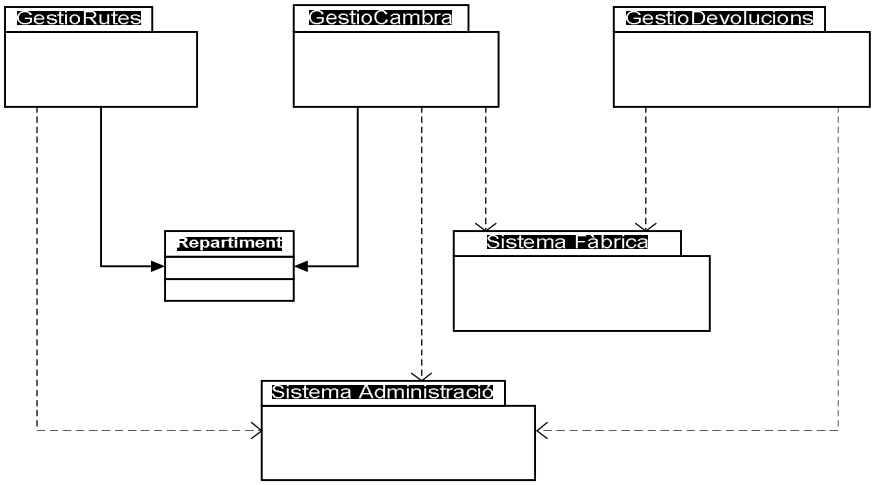
3.7.1. Paquets d'anàlisi:

Dividim el sistema en paquets d'anàlisi per a que en un futur puguin ser implementats per equips diferents, aconseguint que l'implementació del sistema sigui més efectiva i més segura.

A l'hora de crear els paquets d'anàlisi hem tingut en compte què:

- El nostre sistema participa de dos sistemes que ja estan creats dins de l'empresa: el **sistema administratiu** i el **sistema de fàbrica**. D'aquests dos sistemes utilitzem classes pròpies que, a demés, ampliarem afegint-hi nous atributs i noves operacions[29]. A aquests sistemes pertanyen les classes *Client, Comanda, LiniaComanda i CodiPostal* per al que fa al sistema administratiu, i la classe *Palet i Caixa* per al que fa al sistema de fàbrica.
- Entre ells hi hagi les mínimes relacions de dependència possibles.
- Les relacions que hi ha entre els casos d'ús[30] i les classes d'entitat que en formen part[31].

Així doncs hem creat tres paquets d'anàlisis: *GestioRutes*, *GestioCambra* i *GestioDevolucions*. La classe *Repartiment* la hem deixat fora dels paquets ja que es compartida pels paquets *GestioRutes* i *GestioCambra*. Podem veure els paquets d'anàlisi creats i les seves relacions en el següent diagrama estàtic:



[29] Mirar apartat 4.1 "Reutilització de classes" del capítol 4.
[30] Mirar apartat 2.5.1 "Relacions entre actors i casos d'ús" del capítol 2.
[31] Mirar apartat 3.6 "Identificació dels atributs de les classes d'entitat" d'aques mateix capítol.

- *GestioRutes*: Comprèn les classes *Ruta, Subruta* i *Destinacio*. Pel que fa als casos d'ús, inclou l'1. Afegir comanda, el 2. Crear subruta, el 3.Crear repartiment, , el 17. Assignar ruta client, el 18. Eliminar client ruta, el 19.Crear ruta, 20. Dividir ruta i 21 Modificar ruta.

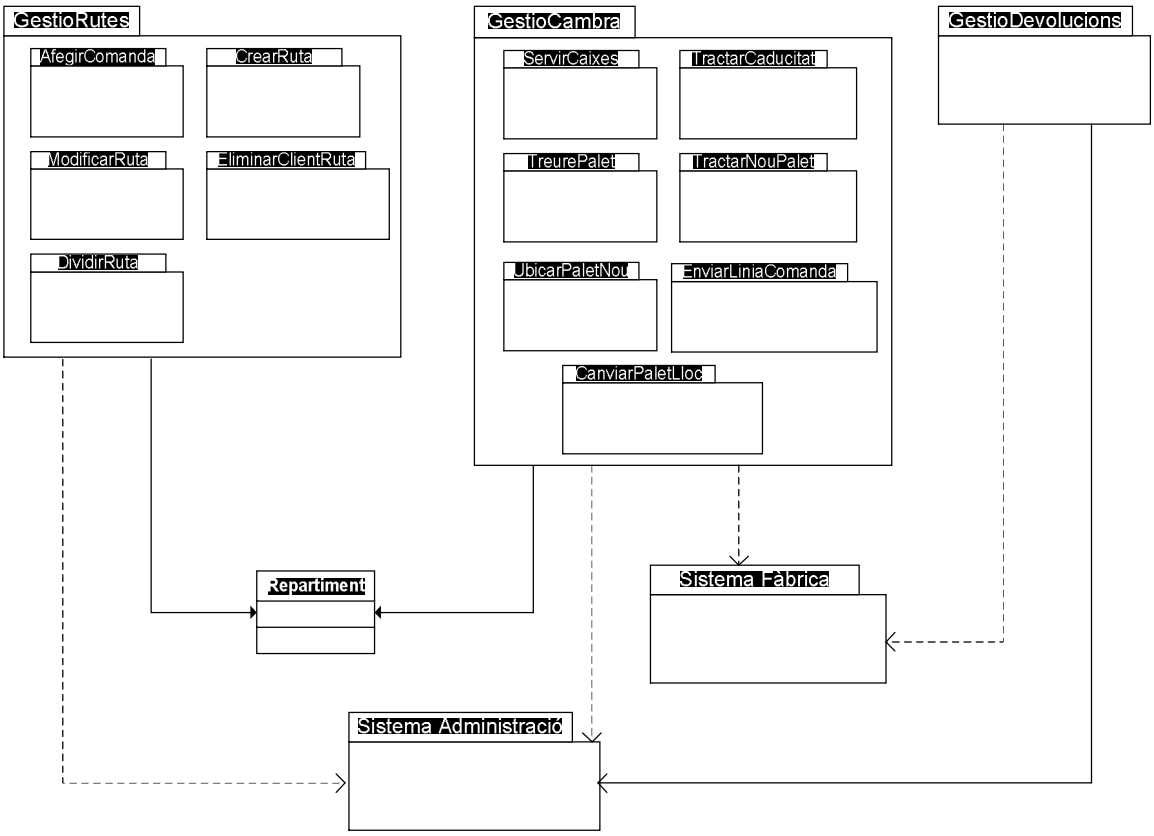
- *GestioCambra*: Comprèn la classes *Passadis, Prestatgeria, Columna* i *Ubicacio*. Pel que fa als casos d'ús, inclou el 4.Seleccionar repartiment, el 5.Seleccionar Comanda, el 6.Enviar línia comanda, el 7.Cercar columna, el 8.Servir caixes, el 9.Descomptar caixes palet, el 10.Guardar línia de comanda, el 11.Canviar palet de lloc, el 12.Tractar nou palet, el 13. Ubicar palet nou, el 14. Modificar buit ubicació, el 15.Tractar caducitat i el 16. Treure palet.

- *GestioDevolucio:* Comprèn una única classe, la classe *Devolucio* i només inclou un cas d'ús, el cas d'ús 22. Crear devolució. La classe devolució només es troba implicada en el cas d'ús 22.

3.7.2. Paquets de servei:

Dins de cadascun dels paquets d'anàlisis hem creat paquets de servei. Cadascun dels paquets creats està format per casos d'ús relacionats. Els paquets de servei són les opcions possibles que té cadascun dels paquets d'anàlisis.

Podem veure els paquets de servei creats en el següent diagrama estàtic:



Paquets de servei del paquet d'anàlisi *GestioRutes*:

- *AfegirComanda*: Comprèn els casos d'ús 1. Afegir comanda, el 2. Crear subruta, el 3.Crear repartiment i el 17. Assignar ruta client. En aquest cas existeixen relacions d'inclusió i d'extensió entre els casos d'ús. Tots quatre intervenen dins el procés d'afegir una comanda a un repartiment, per tant, tots quatre formen una opció dins del paquet.

- Pel que fa als altres paquets de servei del paquet d'anàlisi *GestioRutes*, estan formats per un únic cas d'ús. Aquests casos d'ús són totalment autònoms, no tenen relació amb cap altre cas d'ús. Tots ells són opcions possibles dins del paquet *GestioRutes*. *ModificarRuta* comprèn el cas d'ús 21 Modificar ruta, *CrearRuta* comprèn el cas d'ús 19.Crear ruta, *EliminarClientRuta* comprèn el cas d'ús 18. Eliminar client ruta i *DividirRuta* comprèn el cas d'ús 20. Dividir ruta.

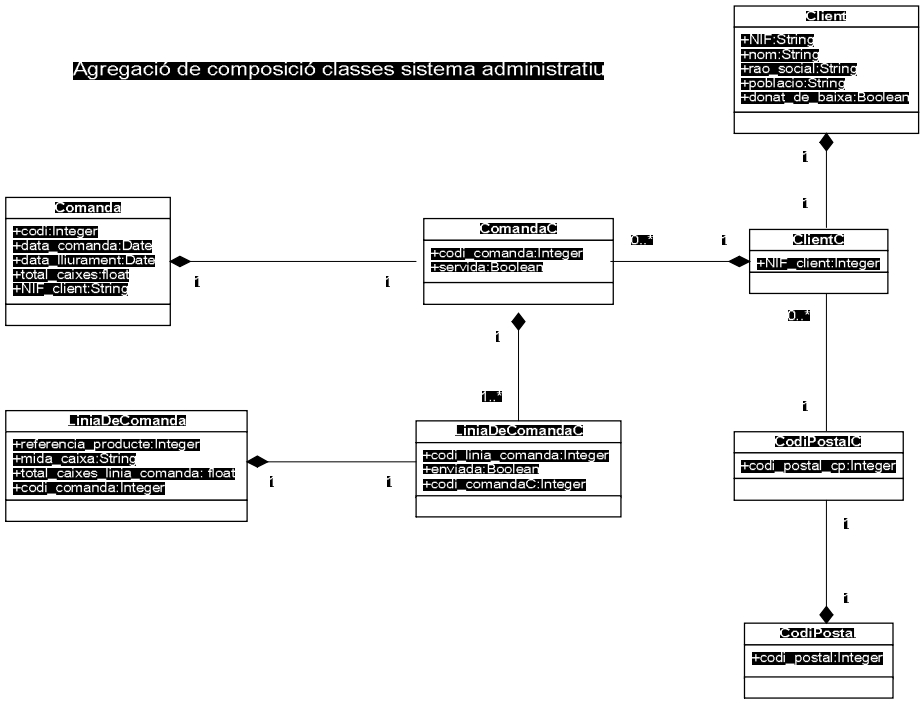
Paquets de servei del paquet d'anàlisi *GestioCambra*:

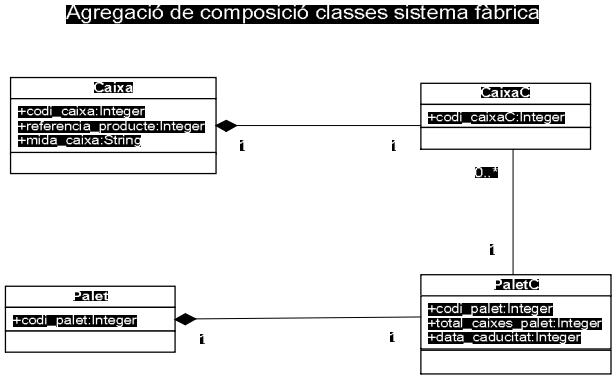
- *EnviarLiniaComanda*: Comprèn els casos d'ús 4.Seleccionar repartiment, el 5.Seleccionar Comanda i 6.Enviar línia comanda. En aquest cas existeixen relacions d'inclusió entre els casos d'ús, per tant, tots tres formen una opció dins del paquet. Tots tres intervenen dins el procés d'enviar una línia de comanda a un robot.
- *ServirCaixes*: Comprèn els casos d'ús 8.Servir caixes, 9.Descomptar caixes palet i 10.Guardar línia de comanda. En aquest cas existeixen relacions d'inclusió entre els casos d'ús, per tant, tots tres formen una opció dins del paquet. Tots tres intervenen dins el procés de servir caixes.
- *TractarNouPalet*: Comprèn els casos d'ús 7.Cercar columna i 12.Tractar nou palet. Aquest dos casos d'ús estan relacionats intervenint en el procés de tractar un palet que entra nou a la cambra.
- *TractarCaducitat*: Comprèn els casos d'ús 7.Cercar columna i 15.Tractar caducitat. Aquest dos casos d'ús estan relacionats intervenint en el procés de tractar un palet de dins la cambra que ha caducat.
- *UbicarPaletNou*: Comprèn els casos d'ús 13. Ubicar palet nou i 14. Modificar buit ubicació. En aquest cas existeix una relació d'inclusió entre els casos d'ús, per tant, tots dos formen una sola opció dins del paquet. Tots dos intervenen en el procés d'ubicar un nou palet dins de la cambra.
- *TreurePalet*: Comprèn els casos d'ús 16. Treure palet i 14. Modificar buit ubicació. En aquest cas existeix una relació d'inclusió entre els casos d'ús, per tant, tots dos formen una sola opció dins del paquet. Tots dos intervenen en el procés de treure un palet de dins de la cambra.
- *CanviarPaletLloc*: Comprèn els casos d'ús 11.Canviar palet de lloc i 14. Modificar buit ubicació. En aquest cas existeix una relació d'inclusió entre els casos d'ús, per tant, tots dos formen una sola opció dins del paquet. Tots dos intervenen en el procés de canviar d'ubicació un palet de dins de la cambra.

4. DISSENY DEL SISTEMA

4.1. Reutilització de classes.

Com hem vist abans, les classes *Comanda*, *LiniaComanda*, *Client* i *CodiPostal* pertanyen al sistema administratiu de l'empresa, i les classes *Palet* i *Caixa* al sistema de fàbrica. D'aquestes classes no ens interessen les seves operacions, ni tampoc part dels seus atributs, pel desenvolupament del nostre sistema. El que ens interessa és afegir-hi noves operacions, nous atributs (en algun cas) i aprofitar alguns dels seus atributs, que tots plegats són els que farem servir en el desenvolupament del nostre sistema. El que farem és complementar aquestes classes dels sistemes d'administració i de fàbrica, mitjançant agregació, ja que al no ser abstractes no ens veiem obligats a fer-ho mitjançant subclasses, la qual cosa ens interessa, perquè si ho féssim mitjançant subclasses finalment tindríem que tornar a desfer l'herència per poder dissenyar la base de dades relacional i segons quin llenguatge orientat a objectes féssim servir per la implementació, també ens veuríem obligats a desfer-la. En els següents diagrames estàtics de classes d'entitats, es mostren les noves classes agregades a les ja existents i la relació que hi ha entre ells:





Com es pot veure en els diagrames:

- les classes agregades a les ja existents són *ClientC, ComandaC, LiniaDeComandaC, CodiPostalC, CaixaC i PaletC*.
- La agregació que s'ha fet és de composició ja que si desapareixen les classes dels sistemes d'administració o de fàbrica també desapareixen les seves classes agregades.
- Les relacions d'agregació tenen una cardinalitat d'1 a 1 ja que a cada objecte de les classes agregades només els hi correspon un objecte de la classe a la que s'agreguen, i al inrevés, a cada objecte de les classe a les que s'agreguen les noves classes, només els hi correspon un objecte de la classe agregada.
- De les classes que ja existien només mostrem els atributs que ens interessen.
- Les noves operacions a afegir s'indiquen en el següent apartat "Identificació de les operacions i revisió dels atributs".
- Per utilitzar els atributs de les classes que ja existien, reutilitzarem les classes en format font o bé compilades.
- Els atributs de totes les classes dels diagrames estan explicats a l'apartat 3.6 "Identificació d'atributs de classes d'entitat" del capítol 3, menys:

Classe CaixaC:

-*codi_caixaC* (integer): número que indica el codi de la caixa de la que forma part un objecte de la classe *CaixaC*. El valor que pren aquest atribut és el mateix valor de l'atribut *codi_caixa* de la caixa de la que forma part l'objecte de la classe *CaixaC*.

Classe ClientC:

-*NIF_client* (integer): número que indica el NIF del client del que forma part un objecte de la classe *ClientC*. El valor que pren aquest atribut és el mateix valor que pren l'atribut *NIF* del client del que forma part l'objecte de la classe *ClientC*.

Classe CodiPostalC:

-*codi_postal_cp* (integer): número que indica el número de codi postal del codi postal del forma part un objecte de la classe *CodiPostalC*. El valor que pren aquest atribut és el mateix valor de l'atribut *codi_postal* del codi postal del que forma part l'objecte de la classe *CodiPostalC*.

Classe ComandaC:

-*codi_comanda* (integer): número que indica el codi de la comanda de la que forma part un objecte de la classe *ComndaC*. El valor que pren aquest atribut és el mateix valor que pren l'atribut *codi* de la comanda de la que forma part l'objecte de la classe *ComndaC*.

Classe LiniaDeComandaC:

-*codi_comandaC* (integer): número que indica el codi de la comanda d'un objecte de la classe *ComandaC* a la que pertany l'objecte de la classe *LiniaDeComandaC*. El valor que pren aquest atribut és el mateix valor de l'atribut *codi_comanda* de l'objecte de la classe *ComandaC* a la que pertany l'objecte de la classe *LiniaDeComandaC*.

-*codi_linia_comanda*(integer): número que indica la referència del producte de la línia de comanda de la que forma part un objecte de la *classe LiniaDeComndaC*. El valor que pren aquest atribut és el mateix valor que pren l'atribut *referencia_producte* de la línia de comanda de la que forma part l'objecte de la classe *LiniaDeComndaC*.

Classe PaletC:

-*codi_palet* (integer): número que indica el codi del palet del que forma part un objecte de la classe *PaletC.* El valor que pren aquest atribut és el mateix valor que pren l'atribut *codi_palet* del palet del que forma part l'objecte de la classe *PaletC.*

Els següents atributs són atributs de les classes *Comanda* i *LiniaDeComanda* que no estan explicat en l'apartat 3.6 "Identificació d'atributs de classes d'entitat" del capítol 3:

Classe Comanda:

NIF_client(string): codi que identifica al client que ha fet la comanda. El valor d'aquest atribut coincideix amb el valor de l'atribut NIF del client que ha fet la comanda.

Classe LiniaDeComanda:

-*codi_comanda* (integer): número que indica el codi de la comanda a la que pertany la línia de comanda. El valor que pren aquest atribut és el mateix valor del codi de la comanda a la que pertany la línia de comanda.

-*referencia_producte(integer)***:** codi que identifica el producte que conte la caixa. Aquest codi és igual a la referència del producte que identifica les línies de comandes.

4.2. Identificació de les operacions i revisió dels atributs

A partir de la descripció textual dels casos d'ús i de l'anàlisi fet a partir dels mateixos, identifiquem les operacions per cadascuna de les diferents classes d'entitat i de control descrites a l'anàlisi. Al final de la descripció textual de les operacions hi ha uns números entre <> que fan referència als casos d'ús en que es fan servir les operacions.

En aquest procés d'identificació també s'ha fet una revisió d'alguns dels atributs d'algunes de les classes d'entitat, per a que les operacions identificades puguin portar-se a terme sense cap problema. De la mateixa manera, s'ha afegit algun atribut nou, necessari pel bon desenvolupament de les operacions.

El fet de modificar algun dels atributs dona claredat al disseny del sistema i a la posterior implementació, mentre que l'afegiment de nous atributs li dona simplicitat i rapidesa, ja que aquests nous atributs fan que es redueixi el nombre d'operacions, alhora que es fa més curt el procés d'algunes de les operacions creades.

Finalment cal aclarir que la identificació de les operacions està feta pensant que la base de dades de l'empresa és una base de dades relacional i que la creació i eliminació dels objectes de les classes d'entitat *ComandaC*, *LiniaDeComandaC* i

ClientsC són responsabilitat del sistema administratiu, com en el cas de les classes d'entitat *PaletC* i *CaixaC*, que ho són del sistema de fàbrica (operacions que es troben en les classes a les que estan agregades).

Pel que fa a les modificacions dels noms dels atributs, s'han canviat el nom d'alguns dels atributs que fan referència als codis de la classe d'entitat a la que pertanyen pel nom de *codi*. Això s'ha fet així ja que, com veurem, s'han afegit nous atributs a algunes de les classes d'entitat que fan referència al codi d'altres classes d'entitat a les quals si se'ls ha donat el nom de *codi_<nom de la classe d'entitat a la que fa referència>*. Amb aquesta modificació es fa més clar el disseny del sistema evitant confusions.

Les operacions *crear* de les classes *Passadis, Prestatgeria, Columna i UbicacioPalet*, només es faran servir quan es posi en funcionament el sistema per primer cop amb l'objectiu d'introduir les dades de l'estructura de la cambra, o bé, si en un futur és necessari fer un canvi en l'estructura de la cambra. En tot cas, per poder realitzar aquestes accions és convenient fer servir una classe de control. Aquesta nova classe de control s'anomena *CreacioEstructuraCambra*.

A continuació s'indiquen les operacions identificades, els nous atributs i els noms modificats dels atributs per les classes d'entitat i de frontera[32].

Classes d'entitat[33]

•*Classe CaixaC*

Operacions:

-*assignarDevolucio (codi_caixa:integer, codi_devolucio:integer):* Aquesta operació assigna una devolució amb el codi de devolució passat com a paràmetre, l'objecte de la classe *CaixaC* amb el codi igual al codi de la caixa passat com a paràmetre. <22>

Nous atributs:

-*codi_devolucio(integer)*: codi que identifica la devolució a la que pertany la caixa. Si la caixa no ha estat retornada el valor d'aquest atribut és nul. El valor d'aquest atribut és el mateix que el valor de l'atribut número de devolució de la devolució a la que pertany la caixa.

Noms modificats: Cap.

[32] Mirar capítol 2.
[33] Mirar l'apartat 2.1. "Identificació de les classes d'entitat" i l'apartat 2.6 "Identificació d'atributs de classes d'entitat" del capítol 2.

●***Classe ClientC***

Operacions:

-*cercar (NIF:string):* Aquesta operació cerca l'objecte de la classe *ClientC* amb el seu NIF igual al NIF passat com a paràmetre. <1>
-*cercarClientsNous():* Aquesta operació cerca els clients que no estan donats de baixa i que no tenen assignada una ruta. <17>
-*cercarClientsBaixa():* Aquesta operació cerca els clients que estan donats de baixa i que tenen assignada una ruta. <18>
-*modificarRutaClient(NIF:string, codi_ruta*:integer): Aquesta operació modifica el codi de la ruta del client amb NIF igual al passat com a paràmetre, pel codi de la ruta passat com a paràmetre.<21>

Nous atributs: Cap

-*codi_ruta(integer):* Codi de la ruta a la que pertany el client. El valor que pot prendre aquest codi és el mateix que el valor que pren l'atribut *codi* de la ruta a la que pertany.

Noms modificats: Cap

●***Classe CodiPostalC***

Operacions:

-*cercarRuta(codi_postal:integer):* Aquesta operació cerca la ruta amb un dels seus codis postals iguals al codi postal passat com a paràmetre. <17>
-*modificarCodiRuta(codi_postal:integer, codi_ruta)*: Aquesta operació modifica el codi de la ruta del codi postal passat com a paràmetre, pel codi de la ruta passat com a paràmetre. <20,21>

Nous atributs:

-*codi_ruta(integer)*: codi que identifica la ruta a la que pertany el codi postal. El valor d'aquest atribut és el mateix que el valor de l'atribut codi de la ruta a la que pertany el codi postal.

Noms modificats: Cap

●***Classe Columna***

Operacions:

-*crear (codi_prestatgeria:integer):* Aquesta operació crea un objecte columna passant-li com a paràmetre el codi de la prestatgeria a la que pertany.
-*cercarPrestatgeria (codi:integer):* Aquesta operació cerca la prestatgeria a la que pertany la columna amb el codi igual al passat com a paràmetre. <6,12,15>

Nous atributs:

-*codi_prestatgeria(integer)*: codi que identifica la prestatgeria a la que pertany la columna. El valor d'aquest atribut és el mateix que el valor de l'atribut codi de la prestatgeria al que pertany la columna.

Noms modificats:

-*codi_columna* es modifica pel nom *codi*.

●***Classe ComandaC***

Operacions:

-*cercarComandesSenseRepartiment*(): Aquesta operació cerca totes les comandes que no tenen cap repartiment assignat. <1>

-*assignarComanda(codi:integer, codi_repartiment:integer):* Aquesta operació assigna a la comanda que té el codi igual al codi passat com a paràmetre, el repartiment el codi del qual és igual al codi del repartiment passat com a paràmetre. Aquesta operació porta implícita una condició que és que el valor de l'atribut codi_repartiment de la comanda ha de ser nul, sinó no es pot efectuar l'operació. <1, 2>

-*esborrarCodiRepartiment(codi:integer):* Aquesta operació serveix per esborrar el codi del repartiment de la comanda el codi de la qual és igual al codi passat com a paràmetre. Això donarà com a resultat que l'atribut *codi_repartiment* prendrà el valor nul i per tant la comanda podrà ser assignada a un altre repartiment. <1>

-*cercarComandesNoServides*(codi_repartiment:integer): Aquesta operació cerca totes les comandes assignades al repartiment amb codi igual al codi passat com a paràmetre, què no han estat servides. <5>

-*modificarServida(codi:integer):* Aquesta operació serveix per canviar el valor de servida de la comanda amb el codi igual al passat com a paràmetre. Si l'estat és servida canviarà a no servida i si és no servida canviarà a servida. Encarà que sigui possible fer el canvi en els dos sentits, el normal és que el canvi només es faci de no servida a servida. <6>

-*cercar*(NIF_client:string, data_lliurament:data): Aquesta operació cerca totes les comandes amb NIF del client i data de lliurament iguals als valors passats com a paràmetre. <22>

Nous atributs:

-*codi_repartiment(integer)*: codi que identifica al repartiment al que pertany la comanda. El valor d'aquest atribut és el mateix que el valor de l'atribut codi del repartiment al que pertany la comanda. Si el valor d'aquest atribut és nul, això indica que la comanda encara no està assignada a cap repartiment.

Noms modificats: Cap

●***Classe Destinacio***

Operacions:

-*crear(codi_ruta:integer, codi_repartiment:integer, codi_subruta:integer, posicio:integer, rao_social:string,)*: Aquesta operació crea un objecte *Destinacio* passant-li com a paràmetre el codi de la ruta a la que pertany, el codi del repartiment al que està assignada (sinó està assignada a cap repartiment el valor passat per paràmetre serà nul), el codi de la subruta al que està assignada (sinó està assignada a cap subruta el valor passat com a paràmetre serà nul), la posició que ocuparà dins la ruta i la raó social del client al que fa referència la destinació. <2>

-*cercarDestinacionsRepartiment(codi_repartiment:integer):* Aquesta operació cerca les destinacions que estan assignades al repartiment amb el codi passat com a paràmetre. <2>

-*assignarDestinacioSubruta(codi_repartiment:integer, codi_subruta:integer)*: Aquesta operació assigna una destinació a la subruta el codi de la qual és igual al codi passat com a paràmetre. El que fa aquesta operació és cridar a l'operació *cercarDestinacions(codi_repartiment:integer)* on el valor del paràmetre serà valor del paràmetre *codi_repartiment* de la pròpia operació. Un cop *cercaDestinacions(codi_repartiment:integer)* trobi una destinació assignada al repartiment el codi del qual és igual al valor del paràmetre, l'operació *assignarDestinacioSubruta* cridarà a l'operació *crear(codi_ruta:integer, codi_repartiment:integer, codi_subruta:integer, posicio:integer, rao_social:string,)*, on el valor dels paràmetres seran els valors dels atributs de la destinació trobada, menys el valor de *codi_subruta* que serà el valor de *codi_subruta* passat com a paràmetre a la pròpia operació i el valor de *codi_repartiment* serà nul. <2>

-*cercarDestinacionsRuta(codi_ruta:integer):* Aquesta operació cerca les destinacions amb posició diferent entre ells que estan assignades a la ruta amb el codi passat com a paràmetre. <17,20>

-*ordenarDestinacionsRuta(codi_ruta:*integer): Aquesta operació ordena les destinacions de la ruta amb el codi igual al codi passat com a paràmetre. <17,18,20>

-*modificarCodiRuta(codi_ruta:integer, codi:integer):* Aquesta operació modifica el codi de la ruta de les destinacions amb codi igual al passat com a paràmetre, pel codi de la ruta passat com a paràmetre. <20,21>

Nous atributs:

-*codi_ruta* (integer): número que indica el codi de la ruta a la que està assignada la destinació. El valor que pren aquest atribut és el mateix valor del codi de la ruta a la que pertany la destinació.

-*codi_repartiment* (integer): número que indica el codi del repartiment al que està assignada la destinació. El valor que pren aquest atribut és el mateix valor del codi del repartiment al que pertany la destinació. Quan la destinació no ha estat mai assignada a cap repartiment aquest atribut pren el valor nul.

-*codi_subruta* (integer): número que indica el codi de la subruta a la que pertany la destinació. El valor que pren aquest atribut és el mateix valor del codi de la subruta al que pertany la destinació. Quan la destinació no esta assignada a cap subruta aquest atribut pren el valor nul.

Noms modificats: Cap

●***Classe Devolució***

Operacions:

-*crear (data_devolucio:data, motius_devolucio:string, total_caixes_retornades:integert, codi_comanda:integer):* Aquesta operació crea un objecte devolució passant-li com a paràmetre la data en que s'ha fet la devolució, els motius de la devolució, el total de caixes retornades i el codi de la comanda a la que pertanyen les caixes retornades. <22>

Nous atributs:

-*codi_comandaC(integer)*: codi que identifica l'objecte de la classe *ComandaC* relacionat amb la comanda a la que pertany la devolució. El valor d'aquest atribut és el mateix que el valor de l'atribut *codi_comanda* de l'objecte de la classe *ComandaC* amb la que esta relacionada la devolució.

Noms modificats: Cap

●***Classe LiniaDeComandaC***

Operacions:

-*cercarLiniaComandaNoEnviada (codi_comanda:integer):* Aquesta operació cerca la primera línia de comanda no enviada que pertany a la comanda amb el codi igual al passat com a paràmetre. <6>

-*modificarServida(referencia_producte:integer):* Aquesta operació modifica el valor de servida de la línia de comanda amb la referència del producte igual a la referència del producte passat com a paràmetre. Si l'estat és servida canviarà a no servida i si és no servida canviarà a servida. Encarà que sigui possible fer el canvi en els dos sentits, el normal és que el canvi només es faci de no servida a servida. <10>

-*cercarLiniesComanda(codi_comanda:*integer): Aquesta operació cerca totes les línies de comanda que pertanyen a la comanda amb codi igual al passat com a paràmetre. <22>

Nous atributs: Cap

Noms modificats: Cap

●***Classe PaletC***

Operacions:

-*descomptarCaixes(codi:intger, total_caixes:integer):* Aquesta operació descompta el total de caixes passades com a paràmetre al total de caixes del palet amb codi igual al codi passat com a paràmetre. <9>

-*controlarTotalCaixes(codi:integer):* Aquesta operació comprova que el total de caixes del palet amb codi igual al passat com a paràmetre, no sigui igual a 0. <9>

-*modificarTotalCaixes(codi:integer, total_caixes:integer)*: Aquesta operació modifica el valor del total de caixes del palet amb codi igual al codi passat com a paràmetre, pel valor del total de caixes passat com a paràmetre. <9>

-*modificarDades(codi*:integer, total_caixes:integer, data_caducitat:integer): Aquesta operació modifica el valor del total de caixes i la data de caducitat del palet amb codi igual al codi passat com a paràmetre, pels valors del total de caixes i la data de caducitat passats com a paràmetres. <12>

-*cercarPaletCaducat(data_actual:data)*: Aquesta operació cerca un palet amb data de caducitat igual a la data passada com a paràmetre. <15>

Nous atributs: Cap

Noms modificats: Cap

●***Classe Passadis***

Operacions:

-*crear ():* Aquesta operació crea un objecte passadís.

Nous atributs: Cap

Noms modificats:

-*codi_passadis* es modifica pel nom *codi*.

●***Classe Prestatgeria***

Operacions:

-*crear (codi_passadis:integer):* Aquesta operació crea un objecte prestatgeria passant-li com a paràmetre el codi del passadís al que pertany.
-*cercarPassadis (codi:integer):* Aquesta operació cerca el passadís al que pertany la prestatgeria amb el codi igual al passat com a paràmetre. <6,12,15>

Nous atributs:

-*codi_passadis(integer)*: codi que identifica el passadís a la que pertany la prestatgeria. El valor d'aquest atribut és el mateix que el valor de l'atribut codi del passadís al que pertany la prestatgeria.

Noms modificats:

-*codi_prestatgeria* es modifica pel nom *codi*.

●***Classe Repartiment***

Operacions:

-*crear(data_creacio:data, data_lliurament:data, total_caixes:float, codi_ruta:integer)*: Aquesta operació crea un objecte repartiment passant-li com a paràmetre la data de creació i de lliurament del repartiment, el total de caixes del repartiment (inicialment serà 0) i el codi de la ruta o subruta a la que pertany. <2>
-*cercarRepartimentObert(codi_ruta:integer, data_lliurament:data)*: Aquesta operació cerca el repartiment amb estat obert que té la mateixa data de lliurament que la data de lliurament passada com a paràmetre i el mateix codi de ruta que el codi de ruta passat com a paràmetre. <1>
-*modificarEstat(codi:integer)*: Aquesta operació serveix per canviar el valor de l'estat del repartiment amb el codi igual al passat com a paràmetre. Si l'estat és obert canviarà a tancat i si és tancat canviarà a obert. <1,2>
-*modificarServit(codi:integer)*: Aquesta operació serveix per canviar el valor de servit del repartiment amb el codi igual al passat com a paràmetre. Si l'estat és servit canviarà a no servit i si és no servit canviarà a servit. <5>
-*descomptarCaixes(codi:integer, total_caixes:float):* Aquesta operació serveix per descomptar el total de caixes passades com a paràmetre al total de caixes del repartiment amb el codi igual al passat com a paràmetre. <1>

-*sumarCaixes(codi:integer, total_caixes:float)*: Aquesta operació suma el total de caixes passades com a paràmetre de al total de caixes del repartiment amb el codi igual al passat com a paràmetre. <1>

-*compararCaixes(codi:integer, capacitat_maxima_camio:float)*: Aquesta operació serveix per comparar el total de caixes del repartiment amb el codi igual al passat com a paràmetre, amb la capacitat màxima d'un camió igual a la capacitat màxima passada com a paràmetre. Aquesta operació es fa servir per comparar que el total de caixes d'un repartiment no sobrepassi la capacitat màxima del camió de la ruta o subruta a la que pertany. <1>

-*cercar(codi_ruta:integer*): Aquesta operació cerca el repartiment que pertany a la ruta o subruta amb el codi igual al passat com a paràmetre. <1>

-*cercarRepartimentsDataLliurament(data_lliurament:data)*: Aquesta operació cerca els repartiments la data de lliurament dels quals és diferent a la data de lliurament passada com a paràmetre. <3>

-*cercarRepartimentsNoServits(data_lliurament:data)*: Aquesta operació cerca els repartiments no servits amb data de lliurament igual a la data de lliurament passada com a paràmetre. <4>

-*cercaRepartimentsRutaNoServits(codi_ruta*:integer): Aquesta operació cerca el repartiments no servits que pertanyen a la ruta amb el codi igual al passat com a paràmetre. <21>

-*modificarRuta(codi_ruta*:*integer, codi:integer)*: Aquesta operació modifica el codi de la ruta dels repartiments amb el codi igual al codi passat com a paràmetre, pel codi de la ruta passat com a paràmetre. <21>

Nous atributs:

-*codi_ruta (integer):* codi que identifica la ruta o subruta a la que pertany el repartiment. El valor que pren aquest atribut és el mateix valor del codi de la ruta o subruta a la que pertany el repartiment.

Noms modificats:

-*codi_repartiment* es modifica pel nom *codi*.

- ***Classe Ruta***

Operacions:

-*crear (nom_transportista:string, capacitat_camio:float)*: Aquesta operació crea un objecte ruta passant-li com a paràmetre el nom del transportista que farà la ruta i la capacitat màxima del camió que conduirà el transportista per fer la ruta. <19,20>

-*cercar (codi_ruta:integer):* Aquesta operació cerca la ruta que pertany a la ruta amb el codi igual al passat com a paràmetre. <3>

-*modificarDades(codi:integer, nom_transportista:string, capacitat_camio:float):* Aquesta operació modifica les dades de la ruta amb el codi igual al passat com a paràmetre per les dades passades com a paràmetre, que poden ser el nom del transportista o la capacitat del camió. <21>

Nous atributs: Cap

Noms modificats:

-*codi_ruta* es modifica pel nom *codi*.

●***Classe Subruta***

Operacions:

-*crear (nom_transportista:string, capacitat_camio:float, codi_ruta:integer)*: Aquesta operació crea un objecte subruta passant-li com a paràmetre el nom del transportista que farà la subruta, la capacitat màxima del camió que conduirà el transportista per fer la subruta i el codi de la ruta al que pertany la subruta. El nom del transportista i la capacitat màxima del camió seran les mateixes que les de la ruta a la que pertany la subruta. <2>

-*cercar (codi_ruta:integer):* Aquesta operació cerca la subruta que pertany a la ruta amb el codi igual al passat com a paràmetre. <1,21>

-*modificarDades(codi:integer, nom_transportista:string, capacitat_camio:float):* Aquesta operació modifica les dades de la subruta amb el codi igual al passat com a paràmetre per les dades passades com a paràmetre, que poden ser el nom del transportista o la capacitat del camió. <21>

Nous atributs:

-*codi_ruta (integer):* codi que identifica la ruta a la que pertany la subruta. El valor que pren aquest atribut és el mateix valor del codi de la ruta al que pertany la subruta.

Noms modificats:

-*codi_subrutat* es modifica pel nom *codi*.

●***Classe Ubicacio***

Operacions:

-*crear(codi_columna:integer, codi:string):* Aquesta operació crea un objecte ubicació buit passant-li com a paràmetre el codi de la columna a la que pertany i el seu codi amb valor "SOT" o "SOB" depenent de la posició que ocupi dins la columna.

-*cercarUbicacioBuida(codi_columna:integer):* Aquesta operació cerca la ubicació buida de la columna amb el codi igual al passat com a paràmetre. <7>

-*cercarPalet(codi_columna:*integer, codi:string): Aquesta operació cerca el codi del palet que es troba a la ubicació passada com a paràmetre de la columna passada com a paràmetre. <9>

-*modificarBuida(codi_columna:*integer, codi:string): Aquesta operació serveix per canviar l'estat buit a ple o l'estat ple a buit de la ubicació amb codi igual al passat com a paràmetre de la columna amb codi igual al passat com a paràmetre. <14>

-*cercarColumnaUbicacio(codi_palet:*integer): Aquesta operació cerca la ubicació i la columna on es troba ubicat el palet amb codi igual al passat com a paràmetre. <15>

Nous atributs:

-*codi_columna(integer)*: codi que identifica la prestatgeria a la que pertany la ubicació. El valor d'aquest atribut és el mateix que el valor de l'atribut codi de la columna al que pertany la ubicació.
-*codi_palet(integer)***:** codi que identifica el palet que es troba en una ubicació determinada.

Noms modificats:

-*codi_ubicacio* es modifica pel nom *codi*.

Classes de control[34]

●**Classe *AdiccioComanda***:

Operacions:

-*crear* ():Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació obre una finestra i instància el format de pantalla *FinestraComandesNoAssignadesRepartiment o FormulariDadesSubruta,* o bé, obre la finestra modal *FinestraProcesCancellat*, *FinestraComandaNoTrobada* o *FinestraRepartiment.* <1>

●**Classe *CreacioSubruta***:

Operacions:

-*crear():* Operació cridada des del cas d'ús 1 quan el sistema troba un repartiment amb la data de lliurament de la comanda, amb l'estat tancat i no assignat a cap subruta o bé quan el nombre de caixes d'un repartiment sobrepassa la capacitat màxima del camió. Aquesta operació instància les classe subruta i destinacions, determina els valors dels números correlatius de les noves subrutes o destinacions, i obre la finestra modal *FinestraSubrutesCreades*. <2>
-*esborrar():* Operació que esborra un objecte de la classe *Subruta*. <2>

●**Classe *CreacioRepartiment***:

Operacions:

-*crear():* Operació cridada des del cas d'ús 1 quan el sistema no troba cap repartiment amb la data de lliurament igual a la data de lliurament d'una comanda determinada o bé cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació instància la classe repartiment i determina el valor (*codi_repartiment*) del número correlatiu del repartiment creat. En el cas d'ús 3, a demés, obre una finestra i instància el format de pantalla *FormulariDataLliurament*, o bé, obre la finestra modal *FinestraRutesRepartimentNoCreat, FinestraNouRepartiment* o *FinestraProcesCancel.lat*. <2, 3>
-*esborrar():* Operació que esborra un objecte de la classe *Repartiment*. <2>

[34] Mirar l'apartat 3.2. "Identificació de les classes de frontera" del capítol 3.

●**Classe *SeleccioRepartiment***:

Operacions:

-*crear* (): Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació obre una finestra i instància el format de pantalla *FinestraLlistatRepartiments,* o bé, obre la finestra modal *FinestraProcesCancel.lat* o *FinestraRepartimentNoTrobat.* <4>

●**Classe *SeleccioComanda***:

Operacions:

-*crear* ():Operació cridada des del cas d'ús 4, quan aquest arriba al final del seu procés, o bé, des del cas d'ús 6 quan no es troba cap línia de comanda no enviada. Aquesta operació obre la finestra modal *FinestraComandaNo-Trobada*. <5>

●**Classe *EnviamentLiniaDeComanda***:

Operacions:

-*crear* ():Operació cridada des del cas d'ús 5, quan aquest arriba al final del seu procés. Aquesta operació envia el *MissatgeLiniaDeComanda* al **Robot** i obre la finestra modal FinestraLiniaDeComandaNoTrobada. <6>

●**Classe *SeleccioColumna*:**

Operacions:

-*crear* ():Operació cridada quan el **Robot** rep el codi de la columna i de la prestatgeria, o bé, quan es cridada pel cas d'ús 7 quan aquest arriba al final del seu procés. Aquesta operació activa la interfície del **Robot** per a que aquest llegeixi i busqui el codi de la prestatgeria i la columna enviades pel sistema, envia el *MissatgeNombreCaixes, MissatgeCanviUbicacio, MissatgeUbicarPaletNou o MissatgeTreurePalet* al **Robot** i rep el missatge *MissatgeColumnaTrobada* que envia el **Robot**.<7>

●**Classe *ServeiCaixes*:**

Operacions:

-*crear* ():Operació cridada quan el **Robot** rep el missatge *MissatgeNombre-Caixes*. Aquesta operació activa la interfície del **Robot** per a que aquest busqui la ubicació "SOT" de la columna que té al davant i per a que llegeixi i serveixi el nombre de caixes que li ha enviat el sistema. Aquesta operació també rep el missatge *MissatgeCaixesServides* que envia el **Robot**.<8>

●**Classe *SubstraccioCaixesPalet***:

Operacions:

-*crear* ():Operació cridada des del cas d'ús 8, quan aquest arriba al final del seu procés. Aquesta operació, quan s'acaba un palet, envia al **Sistema de Fàbrica** un missatge amb el codi del palet que s'ha acabat i al **Robot** el codi de la prestatgeria i el de la columna de la línia de comanda on es troba la ubicació del palet acabat. <9>

●**Classe *GuardantLiniaComanda***:

Operacions:

-*crear* ():Operació cridada des del cas d'ús 8, quan aquest arriba al final del seu procés. <10>

●**Classe *CanviPaletDeLloc***:

Operacions:

-*crear* ():Operació cridada quan el **Robot** rep el missatge *MissatgeCanviUbicacio*. Aquesta operació activa la interfície del **Robot** per a que aquest llegeixi i busqui la ubicació "SOB" i "SOT" de la columna que té al davant, i per a que canviï el palet de la ubicació "SOB" a la ubicació "SOT". Aquesta operació també rep el missatge *MissatgeCanviUbicacioFet* que envia el **Robot**.<11>

●**Classe *TractamentNouPalet***:

Operacions:

-*crear* ():Operació cridada quan el sistema rep del **Sistema de Fàbrica** el missatge *MissatgePaletLliurat*. Aquesta operació envia el missatge *MissatgePrestatgeriaColumna* al **Robot.** <12>

●**Classe *UbicacioPaletNou***:

Operacions:

-*crear* ():Operació cridada quan el **Robot** rep el missatge *MissatgeUbicarPaletNou*. Aquesta operació activa la interfície del **Robot** per a que aquest llegeixi i busqui la ubicació que li ha enviat el sistema, i ubiqui el palet nou al seu lloc. Aquesta operació també rep, del **Robot,** el codi de la columna i la ubicació on s'ha col·locat el nou palet. <13>

●**Classe *ModificacioBuitUbicacio***:

Operacions:

-*crear* ():Operació cridada des del cas d'ús 11 o 13, quan aquests arriben al final del seu procés. <14>

●**Classe*TractamentCaducitat*:**

Operacions:

-*crear* ():Operació cridada quan la data de caducitat d'un palet coincideix amb la data actual del sistema. Aquesta operació envia el *missatge MissatgePrestatgeriaColumna* al **Robot** i un missatge al sistema de fàbrica per a notificar-li quin palet caducat ha de servir. <15>

●**Classe *ExtraccioPalet*:**

Operacions:

- *crear* ():Operació cridada quan el **Robot** rep el missatge *MissatgeTreurePalet*. Aquesta operació activa la interfície del **Robot** per a que aquest llegeixi la data de caducitat que li ha enviat el sistema, busqui el palet amb aquesta data de caducitat i llegeixi els codis d'ubicació i columna d'on a tret el palet. Aquesta operació també rep, del **Robot,** el missatge *MissatgePaletTret*. <16>

●**Classe *AssignacioRutaClient*:**

Operacions:

-*crear():* Operació cridada des del cas d'ús 1 quan el sistema no troba cap ruta assignada al client que ha fet la comanda, o bé, l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació obre una finestra i instància el format de pantalla *FinestraLlistatClientsNous o FinestraListatDestinacions,* o bé, obre la finestra modal *FinestraProcesCancellat* o *FinestraClientNouNoTrobada*. <17>

●**Classe *EliminacioClientRuta*:**

Operacions:

-*crear():* Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació obre una finestra i instància el format de pantalla *FinestraLlistatClientsBaixa,* o bé, obre la finestra modal *FinestraProcesCancellat* o *FinestraClientBaixaNoTrobada*. <18>
-*esborrar():* Operació que esborra un objecte de la classe *Destinacio* quan un client s'ha donat de baixa. <18>

●**Classe *CreacioRuta*:**

Operacions:

-*crear():* Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació instància la classe ruta i determina el valor (codi) del número correlatiu de la ruta creada, obre una finestra i instància el format de pantalla *FormulariDadesRuta,* o bé, obre la finestra modal *FinestraNovaRuta* o *FinestraProcesCancellat*. <19>

Classe *DivisioRuta*:

Operacions:

-*crear():* Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació instància la classe ruta i determina el valor (codi) del número correlatiu de la nova ruta creada a partir de la divisió, obre una finestra i instància el format de pantalla *FormulariNovaRuta, FinestraLlistatRutes* o *FinestraLlistatCodisPostals,* o bé, obre la finestra modal *FinestraDivisioRuta* o *FinestraProcesCancellat*. <20>

●**Classe *ModificacioRuta*:**
Operacions:

-*crear():* Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació obre una finestra i instància el format de pantalla *FormulariModificarRuta o FinestraLlistatRutes,* o bé, obre la finestra modal *FinestraRutaModificada* o *FinestraProcesCancel.lat*. <21>

●**Classe *CreacioDevolucio*:**

Operacions:

-*crear():* Operació cridada quan l'**Operador logístic** selecciona l'operació corresponent al *Menú*. Aquesta operació instància la classe *Devolucio* i determina el valor (codi) del número correlatiu de la devolució creada, obre una finestra i instància el format de pantalla *FormulariDevolucio* o *FinestraFormulariComanda,* o bé, obre la finestra modal *FinestraDevolucioCreada* o *FinestraProcesCancellat*. <22>

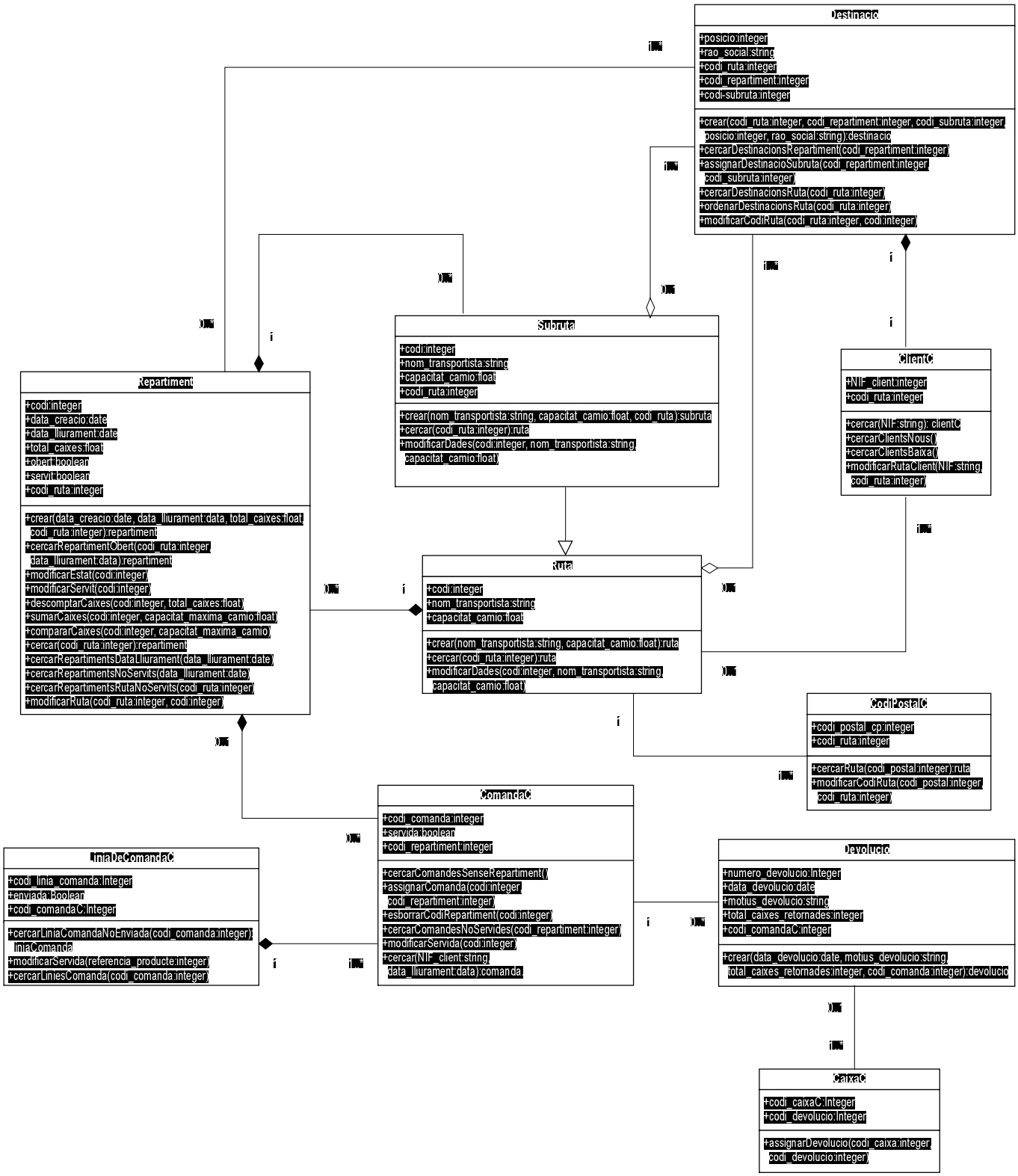
Nova classes de control

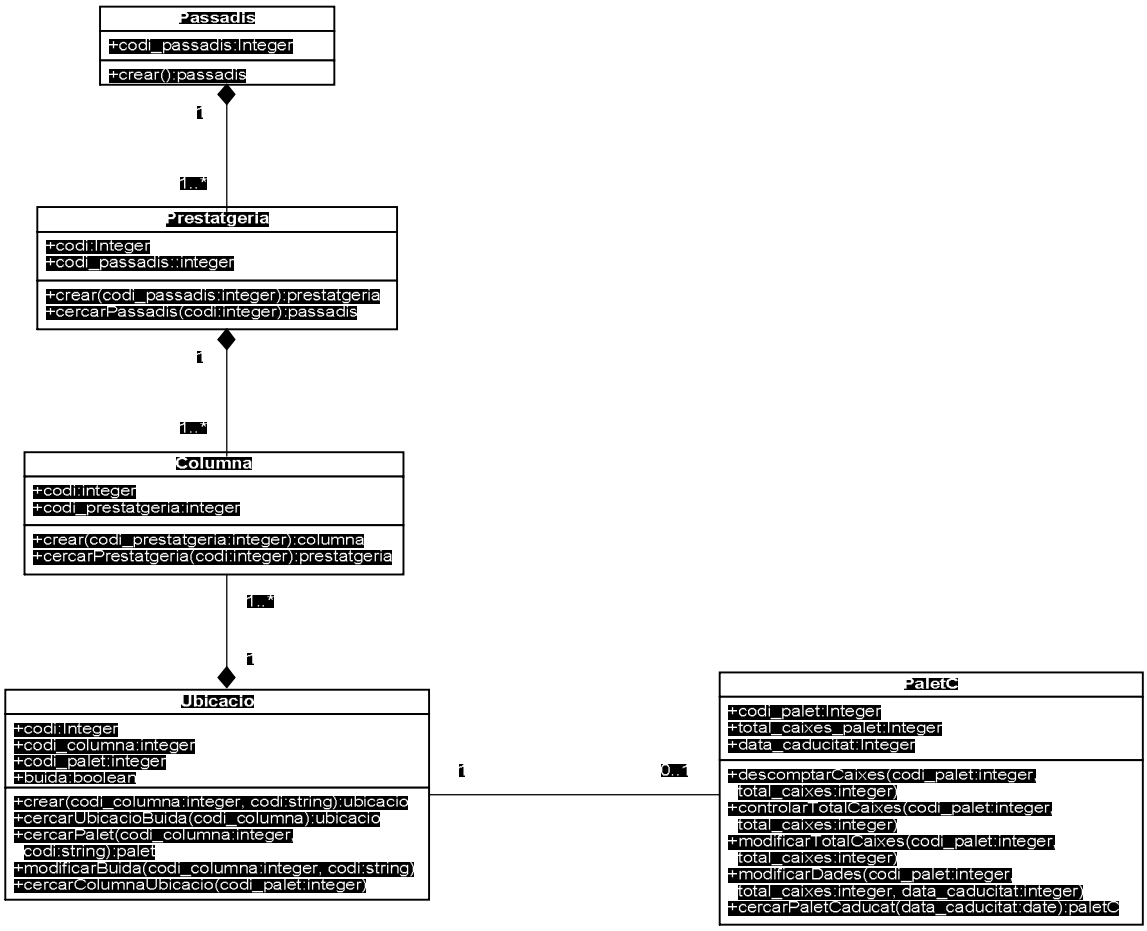
●**Classe *CreacioEstructuraCambra***:

Operacions:

-*crear():* Aquesta operació instància la classe *Passadis, Prestatgeria, Columna* o *Ubicacio* i determina el valor (*codis*) del número correlatiu dels objectes creats de les diferents classes, menys en el cas de les ubicacions que el seu codi només pot ser "SOT" o "SOB" i aquest serà passat com a paràmetre en l'operació crear de la classe Ubicació.
-*esborrar():* Operació que esborra un objecte de la classe *Passadis, Prestatgeria, Columna* o *UbicacioPalet.* Aquesta operació es farà servir quan es vulgui reestructurar la cambra, abans de crear els nous objectes.

4.3. Diagrama estàtic de disseny.





Com podem veure, el diagrama estàtic de disseny, té alguna variació respecte al diagrama estàtic d'anàlisis obtingut al capítol 2. Un cop dissenyades les operacions i la reutilització de les classes d'entitat, hem pogut simplificar el diagrama:

- Eliminant algunes de les relacions què, o bé no tenien sentit en el nostre sistema (com la relació entre la classe CaixaC i PaletC, on les caixes, com a objecte de la classe CaixaC, només tenen sentit per les devolucions i els palets, com a objecte de la classe PaletC, per l'estructura de la cambra), o bé eren redundants dins del sistema.

- Substituint les classes originals del sistema administratiu per les noves classes agregades a les primeres. Aquestes classes incorporen menys atributs.

4.4. Disseny de la persistència.

Totes les classes d'entitat del sistema són persistents i la base de dades de l'empresa és relacional, per tant, farem servir el model de la persistència per a bases de dades relacionals. De les tres maneres que hi ha de gestionar aquest model de persistència, fem servir el mètode que consisteix a crear una classe gestora de disc per cadascuna de les classes d'entitat (ja hem dit abans que totes són persistents) i fer que algunes de les seves operacions, a part de la seva funció específica, cridin a les operacions del gestor de disc pertinents. Utilitzant aquest mètode evitem limitar la portabilitat de l'aplicació i augmentem l'eficiència.
Per fer el disseny de la persistència seguirem quatre passos:

- Supressió de l'herència, ja que el model relacional no la suporta.
- Transformació del model estàtic en un model entitat-relació.

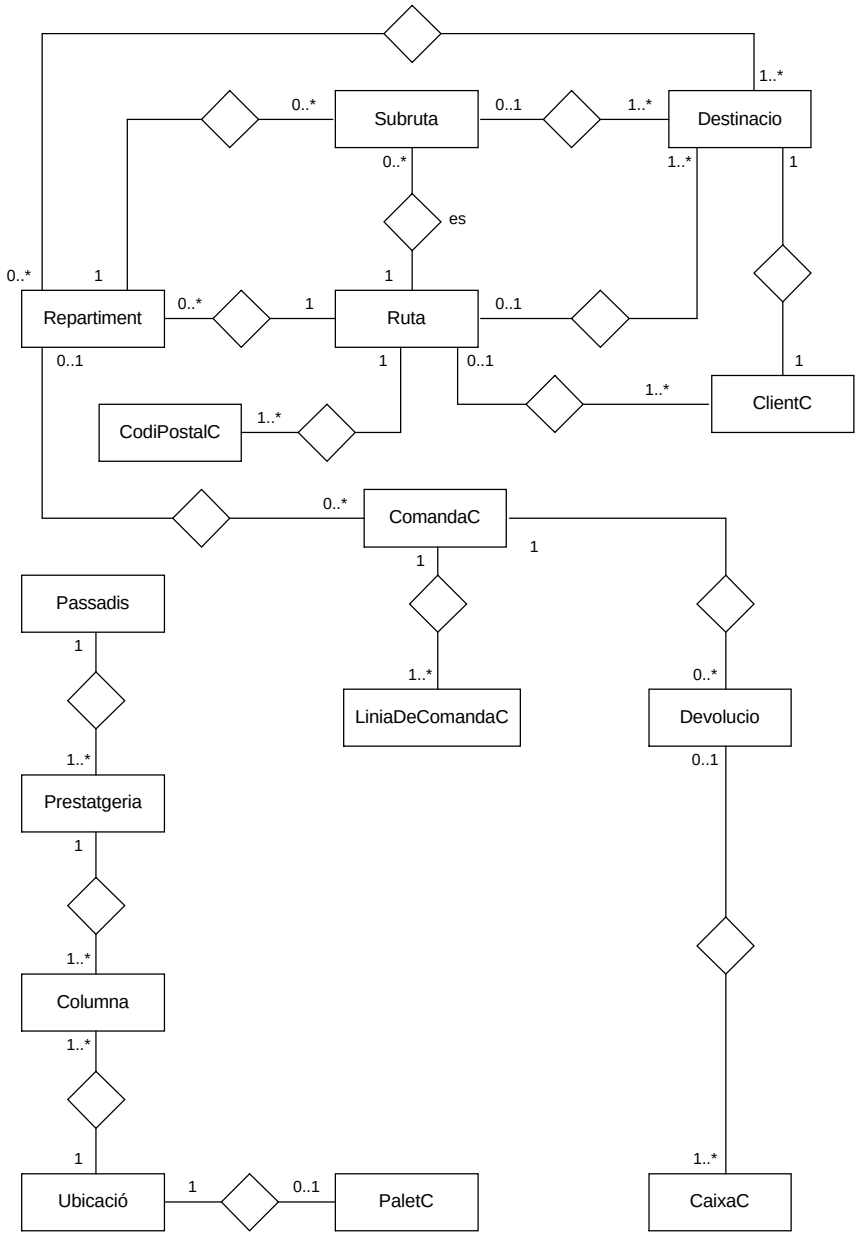
- Transformar el model entitat-relació en un model relacional segons les regles habituals del disseny de base de dades relacionals.
- Crear un gestor de disc per cadascuna de les classes d'entitat, els quals implementaran els accessos a les taules dissenyades en el pas anterior.

4.4.1. Supressió de l'herència

L'única relació d'herència que tenim és la que hi ha entre les classes Ruta i Subruta. S'ha fet una supressió mitjançant l'herència múltiple per duplicació, ja que no són gaires els atributs a duplicar, i és una forma senzilla i clara de suprimir l'herència. Així doncs hem duplicat els atributs comuns a les dos classes en la classe Subruta.

4.4.2. Transformació del model estàtic al model entitat-relació.

A partir del model estàtic del apartat 4.3 d'aquest mateix capítol, obtenim el diagrama entitat-relació a partir del qual obtindrem les taules del nostre sistema que s'implementaran en la base de dades relacional de l'empresa.



4.4.3. Transformació del model entitat-relació en taules relacionals.

Per fer la descripció de la base de dades farem servir la terminologia general del model relacional, ja que no sabem quin és el sistema de gestió de la base de dades de l'empresa[35]. Farem referència a les taules *taula_comandes, taula_linies_de_comandes, taula_clients* i *taula_codis_postals* de la base de dades de l'empresa, dissenyades des del sistema administratiu. De la mateixa manera, farem referència a les taules *taula_caixes* i *taula_palets* de la base de dades de l'empresa, dissenyades des del sistema de fàbrica. D'aquestes taules només ens interessa saber el nom de la columna que les relaciona amb les taules que obtindrem nosaltres. A partir del diagrama entitat-relació de l'apartat anterior i dels atributs de les classes d'entitat[36] obtenim les taules del nostre sistema per a que siguin implementades en la base de dades de l'empresa:

- ***taula_caixes_c,*** amb les columnes *codi_caixaC* i *codi_devolucio*. La clau primària és *codi_caixaC* i alhora és clau forana corresponent a *codi_caixa* de *taula_caixes* del sistema administratiu. Hi ha també una altre clau forana *codi_devolucio* corresponent a *numero_devolucio* de *taula_devolucions.*

- ***taula_clients_c,*** amb les columnes *NIF_client* i *codi_ruta.* La clau primària és *NIF_client* i alhora és clau forana corresponent a *NIF* de *taula_clients* del sistema administratiu. Hi ha també una altre clau forana *codi_ruta* corresponent a *codi* de *taula_rutes.*

- ***taula_codis_postals_c,*** amb les columnes *codi_postal_cp* i *codi_ruta*. La clau primària és *codi_postal_cp* i alhora és clau forana corresponent a *codi_postal* de *taula_codis_postals* del sistema administratiu. Hi ha també una altre clau forana *codi_ruta* corresponent a *codi* de *taula_rutes.*

- ***taula_columnes,*** amb les columnes *codi* i *codi_prestatgeria.* La clau primària és *codi* i *codi_prestatgeria* és clau forana corresponent a *codi* de *taula_prestatgeries.*

- ***taula_comandes_c,*** amb les columnes *codi_comanda, servida* i *codi_repartiment*. La clau primària és *codi_comanda* i alhora és clau forana corresponent a *codi* de *taula_comandes* del sistema administratiu. Hi ha també una altre clau forana *codi_repartiment* corresponent a *codi* de *taula_repartiments.*

- ***taula_destinacions,*** amb les columnes *posicio, rao_social, codi_ruta, codi_repartiment* i *codi_subruta*. La clau primària és compon dels atributs *posicio, rao_social* i *codi_ruta*. Hi ha tres claus foranes, *codi_ruta* corresponent a *codi* de *taula_rutes, codi_repartiment* corresponent a *codi* de *taula_repartiments* i *codi_subruta* corresponent a *codi* de *taula_subrutes.*

- ***taula_devolucions,*** amb les columnes *numero_devolucio, data_devolucio, motius_devolucio, total_caixes_retornades* i *codi_comandaC*. La clau primària és *numero_devolucio* i *codi_comandaC* és clau forana corresponent a *codi_comanda* de *taula_comandesC.*

- ***taula_linies_de_comandes_c,*** amb les columnes *codi_linia_comanda, enviada* i *codi_comandaC.* La clau primària és *codi_linia_comanda* i alhora és clau fora-

[35] Hem de pensar que l'empresa no és real, i per tant, crear un sistema de gestió de la base de dades, no entra en els objectius del TFC.
[36] Mirar l'apartat 4.3 "Diagrama estàtic de disseny" d'aquest mateix apartat.

na corresponent a *referencia_producte* de *taula_linia_comandes* del sistema administratiu. Hi ha també una altre clau forana *codi_comanda* corresponent a *codi* de *taula_comandes_c.*

- ***taula_palets_c,*** amb les columnes *codi_palet* i *total_caixes_palet.* La clau primària és *codi_palet* i alhora és clau forana corresponent a *codi_palet* de *taula_palets* del sistema administratiu.

- ***taula_passadissos,*** amb la columna *codi_passadis.* La clau primària és *codi_passadis.*

- ***taula_prestatgeries,*** amb les columnes *codi* i *codi_passadis.* La clau primària és *codi* i *codi_passadis* és clau forana corresponent a *codi* de *taula_passadisos.*

- ***taula_repartiments,*** amb les columnes *codi, data_cracio, data_lliurament, total_caixes, obert, servit* i *codi_ruta.* La clau primària és *codi* i *codi_ruta* és clau forana corresponent a *codi* de *taula_rutes.*

- ***taula_rutes,*** amb les columnes *codi, nom_transportista* i *capacitat_camio*. La clau primària és *codi*.

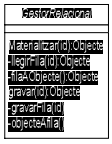
- ***taula_subrutes***, amb les columnes *codi, nom_transportista, capacitat_camio* i *codi_ruta*. La clau primària és *codi* i *codi_ruta* és clau forana corresponent a *codi* de *taula_rutes.*

- ***taula_ubicacions,*** amb les columnes *codi, codi_columna, codi_palet* i *buida.* La clau primària és compon dels atributs *codi* i *codi_columna*. Hi ha dues claus foranes, *codi_columna* corresponent a *codi* de *taula_columnes* i *codi_paletC* corresponent a *codi_palet* de *taula_palets_c.*

Hi hauria d'haver un índex per cada clau primària. Pel que fa a les claus foranes, en aquest nivell no fan falta, però es tindrà que mirar si calen a l'hora d'implementar les taules.

4.4.4. Gestors de disc.

Els gestors de disc ens proporcionaran les operacions bàsiques sobre les files de les taules del nostre sistema[37]. El gestors de disc, tal i com hem indicat en l'apartat 1.3 "Enfocament i mètode seguit" del capítol 1 , seran subclasses de *GestorRelacional* que alhora és subclasse de *GestorGeneric* dins del patró *Template Method.* La subclasse *GestorRelacional* és la següent:



Per dissenyar els gestors de disc hem tingut en compte les operacions de les classes d'entitats descrites a l'apartat 4.2. d'aquest mateix capítol.

[37] Mirar apartat anterior.

Els gestors de disc del nostre sistema són:

- ***GestorCaixaC***:

  Operacions:

  -*llegirFila():* És cridada per l'operació *assignarDevolucio* de la classe *CaixaC*. Llegeix una fila de *taula_caixes_c* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *objecteAFila()*.
  -*objecteAFila():* Mou un a un els atributs de l'objecte de *CaixaC* als valors de les columnes de la fila llegida per *llegirFila* de *taula_caixes_c* i crida a l'operació *gravarFila()*.
  -*gravarFila():* grava la fila modificada per *objecteAfila()* a *taula_caixes_c.*

- ***GestorClientC***:

  Operacions:

  -*llegirFila():* És cridada per l'operació *cerca* o *modificarRutaClient* de la classe *ClientC*. Llegeix una fila de *taula_clients_c* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte()* si l'operació que l'ha cridat és *cercar,* o crida a l'operació *objecteAFila()* si l'operació que l'ha cridat és *modificarRutaClient*.
  -*filaAObjecte():*Crea un objecte de *ClientC* i mou un a un els valors llegits per l'operació que l'ha cridat als atributs de l'objecte creat.
  -*consultar():* És cridada per l'operació *cercarClientsNous* o *cercarClientsBaixa* de la classe *ClientC*. Llegeix una a una les files de *taula_clients_c* i les de *taula_clients* identificades per la mateixa clau primària amb els valors de totes les columnes i que compleixin una condició que dependrà dels paràmetres de l'operació que ha cridat *consultar(),* i crida *filaAObjecte* per a cada una de les files obtingudes.
  -*objecteAFila():* Mou un a un els atributs de l'objecte de *ClientC* als valors de les columnes d'una fila de *taula_clients_c* i crida a l'operació *gravarFila()*.
  -*gravarFila():* grava la fila modificada per *objecteAfila()* a *taula_clients_c.*

- ***GestorCodiPostalC***:

  Operacions:

  -*llegirFila():* És cridada per l'operació *cercaRuta* o *modificarCodiRuta* de la classe *CodiPostalC*. Llegeix una fila de *taula_codis_postals_c* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte()* si l'operació que l'ha cridat és *cercarRuta,* o crida a l'operació *objecteAFila()* si l'operació que l'ha cridat és *modificarCodiRuta*.
  -*filaAObjecte():*Crea un objecte de *CodiPostalC* i mou un a un els valors llegits per *llegirFila()* als seus atributs.
  -*objecteAFila():* Mou un a un els atributs de l'objecte de *CodiPostalC* als valors de les columnes d'una fila de *taula_codis_postals_c* i crida a l'operació *gravarFila()*.
  -*gravarFila():* grava la fila modificada per *objecteAfila()* a *taula_codis_postals_c.*

- ***GestorColumna***:

Operacions:

-*llegirFila():* És cridada per l'operació *cercaPrestatgeria* de la classe *Columna*. Llegeix una fila de *taula_columnes* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte().*
-*filaAObjecte():*Crea un objecte de *Columna* i mou un a un els valors llegits per *llegirFila()* als seus atributs.
-*objecteAFila():* És cridada per l'operació *crear* de la classe *Columna*. Mou un a un els atributs de l'objecte de *Columna* als valors de les columnes d'una fila de *taula_columnes* i crida a l'operació *gravarFila()*.
-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_columnes.*

- ***GestorComandaC***:

Operacions:

-*llegirFila():* És cridada per l'operació *assignarComanda, esborrarCodiRepartiment* o *modificarServida* de la classe *ComandaC*. Llegeix una fila de *taula_comandes_c* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *objecteAFila()* si l'operació que l'ha cridat és *assignarComanda* o *modificarServida*, o crida a l'operació *esborrar()* si l'operació que l'ha cridat és *esborrarCodiRepartiment.*
-*objecteAFila():* Mou un a un els atributs de l'objecte de *ComandaC* als valors de les columnes d'una fila de *taula_comandes_c* i crida a l'operació *gravarFila()*.
-*gravarFila():* Grava la fila modificada per *objecteAfila()* a *taula_comandes_c.*
-*esborrar():* Substitueix el valor de la columna *codi_repartiment* pel valor nul d'una fila de *taula_comandes_c* i crida a l'operació *gravarFila()*.
-*consultar():* És cridada per l'operació *cercar, cercarComandesSenseRepartiment* o *cercarComandesNoServides* de la classe *ComandaC*. Llegeix una a una les files de *taula_comandes_c* i les de *taula_comandes* identificades per la mateixa clau primària amb els valors de totes les columnes i que compleixin una condició que dependrà dels paràmetres de l'operació que ha cridat *consultar(),* i crida *filaAObjecte* per a cada una de les files obtingudes.
-*filaAObjecte():*Crea un objecte de *ComandaC* i mou un a un els valors llegits per l'operació que l'ha cridat als atributs de l'objecte creat.

- ***GestorDestinacions***:

Operacions:

-*llegirFila():* És cridada per l'operació *modificarCodiRuta* de la classe *Destinacio*. Llegeix una fila de *taula_destinacions* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *objecteAFila().*
-*consultar():* És cridada per l'operació *cercarDestinacionsRepartiment* o *cercarDestinacionsRuta* de la classe *Destinacio*. Llegeix una a una les files de *taula_destinacions* identificades per la clau primària amb els valors de totes les columnes i que compleixin una condició que dependrà dels paràmetres de l'operació que ha cridat *consultar(),* i crida *filaAObjecte* per a cada una de les files obtingudes.

-*filaAObjecte():*Crea un objecte de *Destinacio* i mou un a un els valors llegits per *llegirFila()* als seus atributs.
-*objecteAFila():* És cridada per l'operació *crear o ordenarDestinacionsRuta*[38] de la classe *Destinacio,* o bé, per l'operació *llegirFila()* de *GestorDestinacions*. Mou un a un els atributs de l'objecte de *Destinacio* als valors de les columnes d'una fila de *taula_destinacions* i crida a l'operació *gravarFila()*.
-*gravarFila():* grava la fila afegida o modificada per *objecteAfila()* a *taula_destinacions.*

- ***GestorDevolucions***:

Operacions:

-*objecteAFila():* És cridada per l'operació *crear* de la classe *Devolucio*. Mou un a un els atributs de l'objecte de *Devolucio* als valors de les columnes d'una fila de *taula_devolucions* i crida a l'operació *gravarFila()*.
-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_devolucions.*

- ***GestorLiniaComandaC***:

Operacions:

-*llegirFila():* És cridada per l'operació *modificarServida* de la classe *LiniaComandaC*. Llegeix una fila de *taula_linies_comandes_c* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *objecteAFila().*
-*objecteAFila():* Mou un a un els atributs de l'objecte de *LiniaComandaC* als valors de les columnes d'una fila de *taula_linies_comandes_c* i crida a l'operació *gravarFila()*.
-*gravarFila():* Grava la fila modificada per *objecteAfila()* a *taula_linies_comandes_c.*
-*consultar():* És cridada per l'operació *cercarLiniesComanda* o *cercarLiniaComandaNoEnviada* de la classe *LiniaDeComandaC*. Llegeix una a una les files de *taula_linies_comandes_c* i les de *taula_linies_comandes* identificades per la mateixa clau primària amb els valors de totes les columnes i que compleixin una condició que dependrà dels paràmetres de l'operació que ha cridat *consultar(),* i crida *filaAObjecte* per a cada una de les files obtingudes.
-*filaAObjecte():*Crea un objecte de *LiniaDeComandaC* i mou un a un els valors llegits per l'operació que l'ha cridat als atributs de l'objecte creat.

- ***GestorPaletC***:

Operacions:

-*llegirFila():* És cridada per l'operació *descomptarCaixes, modificarTotalCaixes, modificarDades* o *controlarTotalCaixes* de la classe *PaletC*. Llegeix una fila de *taula_palets_c* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte()* si l'operació que l'ha cridat és *descomptarCaixes* o *controlarTotalCaixes*, si l'operació que l'ha cridat és *modificarTotalCaixes* o *modificarDades* aleshores crida a l'operació *objecteAFila().*

[38] L'operació *ordenarDestinacionsRuta*, crida a l'operació *objecteAfila* per cada una de les files que ha ordenat.

-*objecteAFila():* Mou un a un els atributs de l'objecte de *PaletC* als valors de les columnes d'una fila de *taula_palets_c* i crida a l'operació *gravarFila()*.
-*gravarFila():* Grava la fila modificada per *objecteAfila()* a *taula_palets_c.*
-*consultar():* És cridada per l'operació *cercarPaletCaducat* de la classe *PaletC*. Llegeix una a una les files de *taula_palets_c* que tinguin la mateixa data de caducitat que la data passada per paràmetre de l'operació *cercarPaletCaducat* i crida *filaAObjecte* per a cada una de les files obtingudes.
-*filaAObjecte():*Crea un objecte de *PaletC* i mou un a un els valors llegits per l'operació que l'ha cridat als atributs de l'objecte creat.

- ***GestorPassadis***:

Operacions:

-*objecteAFila():* És cridada per l'operació *crear* de la classe *Passadis*. Mou un a un els atributs de l'objecte de *Passadis* als valors de les columnes d'una fila de *taula_passadissos* i crida a l'operació *gravarFila()*.
-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_passadissos.*

- ***GestorPrestatgeria***:

Operacions:

-*llegirFila():* És cridada per l'operació *cercaPassadis* de la classe *Prestatgeria*. Llegeix una fila de *taula_prestatgeries* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte().*
-*filaAObjecte():*Crea un objecte de *Prestatgeries* i mou un a un els valors llegits per *llegirFila()* als seus atributs.
-*objecteAFila():* És cridada per l'operació *crear* de la classe *Prestatgeria*. Mou un a un els atributs de l'objecte de *Prestatgeria* als valors de les columnes d'una fila de *taula_prestatgeries* i crida a l'operació *gravarFila()*.
-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_prestatgeries.*

- ***GestorRepartiment***:

Operacions:

-*llegirFila():* És cridada per l'operació *modificarEstat, modificarServit, modificarRuta, descomptarCaixes, sumarCaixes* o *compararCaixes* de la classe *Repartiment*. Llegeix una fila de *taula_repartiments* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte()* si l'operació que l'ha cridat és *descomptarCaixes, sumarCaixes* o *compararCaixes,* o crida a l'operació *objecteAFila()* si l'operació que l'ha cridat és *modificarEstat, modificarServi o modificarRuta.*
-*filaAObjecte():*Crea un objecte de *Repartiment* i mou un a un els valors llegits per l'operació que l'ha cridat als atributs de l'objecte creat.
-*consultar():* És cridada per l'operació *cercarRepartimentObert, cercar, cercarRepartimentsDataLliurament, cercarRepartimentsNoServits,* o *cercarRepartimentsRutaNoServits* de la classe *Repartiment*. Llegeix una a una les files de *taula_repartiments* identificades per la clau primària amb els valors de totes les columnes i que compleixin una condició que dependrà dels paràmetres de l'operació que ha cridat *consultar(),* i crida *filaAObjecte* per a cada una de les files obtingudes.
-*objecteAFila():* És cridada per l'operació *crear* de la classe *Repartiment*, o bé per l'operació *llegirFila()* de *GestorRepartimen*. Mou un a un els atributs

de l'objecte de *Repartiment* als valors de les columnes d'una fila de *taula_repartiments* i crida a l'operació *gravarFila()*.

-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_repartiments.*

- ***GestorRuta***:

Operacions:

-*llegirFila():* És cridada per l'operació *cercar* o *modificarDades* de la classe *Ruta*. Llegeix una fila de *taula_rutes* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte()* si l'operació que l'ha cridat és *cercar,* o crida a l'operació *objecteAFila()* si l'operació que l'ha cridat és *modificarDades.*

-*filaAObjecte():*Crea un objecte de *Rutes* i mou un a un els valors llegits per *llegirFila()* als seus atributs.

-*objecteAFila():* És cridada per l'operació *crear* de la classe *Ruta,* o bé per l'operació *llegirFila()* de *GestorRuta*. Mou un a un els atributs de l'objecte de *Ruta* als valors de les columnes d'una fila de *taula_rutes* i crida a l'operació *gravarFila()*.

-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_rutes.*

- ***GestorSubruta***:

Operacions:

-*llegirFila():* És cridada per l'operació *cercar* o *modificarDades* de la classe *Subruta*. Llegeix una fila de *taula_subrutes* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *filaAObjecte()* si l'operació que l'ha cridat és *cercar,* o crida a l'operació *objecteAFila()* si l'operació que l'ha cridat és *modificarDades.*

-*filaAObjecte():*Crea un objecte de *Subrutes* i mou un a un els valors llegits per *llegirFila()* als seus atributs.

-*objecteAFila():* És cridada per l'operació *crear* de la classe *Subruta,* o bé per l'operació *llegirFila()* de *GestorSubruta*. Mou un a un els atributs de l'objecte de *Subruta* als valors de les columnes d'una fila de *taula_subrutes* i crida a l'operació *gravarFila()*.

-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_subrutes.*

- ***GestorUbicacions***:

Operacions:

-*llegirFila():* És cridada per l'operació *modificarBuida* de la classe *Ubicacio*. Llegeix una fila de *taula_ubicacions* identificada per la clau primària amb els valors de totes les columnes i crida a l'operació *objecteAFila().*

-*consultar():* És cridada per l'operació *cercarUbicacioBuida, cercarPalet* o *cercarColumnaUbicacio* de la classe *Ubicacio*. Llegeix una a una les files de *taula_ubicacions* identificades per la clau primària amb els valors de totes les columnes i que compleixin una condició que dependrà dels paràmetres de l'operació que ha cridat *consultar(),* i crida *filaAObjecte* per a cada una de les files obtingudes.

-*filaAObjecte():*Crea un objecte de *Ubicacio* i mou un a un els valors llegits per *consultar()* als seus atributs.

-*objecteAFila():* És cridada per l'operació *crear* de la classe *Ubicacio,* o bé per l'operació *llegirFila()* de *GestorUbicacio*. Mou un a un els atributs de

l'objecte de *Ubicacio* als valors de les columnes d'una fila de *taula_ubicacions* i crida a l'operació *gravarFila()*.

-*gravarFila():* grava la fila afegida per *objecteAfila()* a *taula_ubicacions.*

Tots els Gestor de disc s'instancien en engegar el sistema, per mitjà de l'operació *crear*.

4.5. Disseny de la interfície d'usuari

Per fer el disseny de la interfície d'usuari hem tingut en compte l'especificació dels requisits de la interfície d'usuari fetes en el capítol 1 apartat 1.7 i ens hem basat en la descripció textual de les classes de frontera descrites en el capítol 2 apartat 2.2.

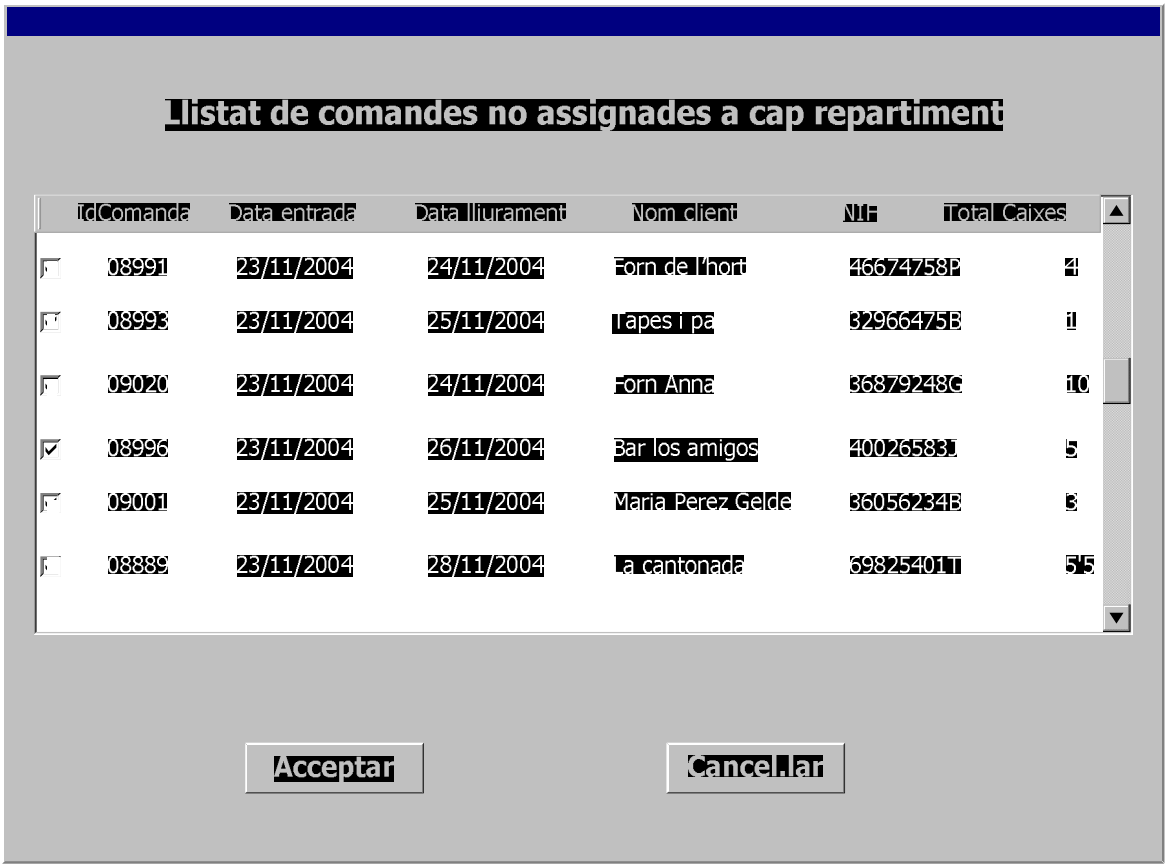
Totes les classes de frontera que són finestres modals que ensenyen un missatge[39] per pantalla, avisant a l'operador logístic d'algun fet, tenen el següent format:

Fins que l'usuari no pitgi el botó d'acceptar, la finestra modal no es tancarà, i per tant, el sistema no podrà executar cap altre procés. Això s'ha fet així pensant que el avisos (missatges, resultats d'operacions,...) són importants per a informar a l'usuari dels resultats obtinguts pel sistema. D'aquesta manera fins que l'usuari no llegeixi l'avís, invertint el temps que li calgui, el missatge romandrà a la pantalla.

Un altre tipus d'interfície que hi ha al sistema, és la que es representa amb un format de pantalla que llista un conjunt d'objectes on l'usuari a de seleccionar un. Aquest és el cas que correspon, per exemple, a la classe de frontera *FinestraComandesNoAssignadesRepartiment*[40]*.*

[39] Mirar aprtat 2.2 "Identificació de les classes de frontera", capítol 2.

[40] Mirar la descripció textual de la classe de frontera en el cas d'ús 1 a l'apartat 2.2 "Identificació de les classes de frontera" del capítol 2.



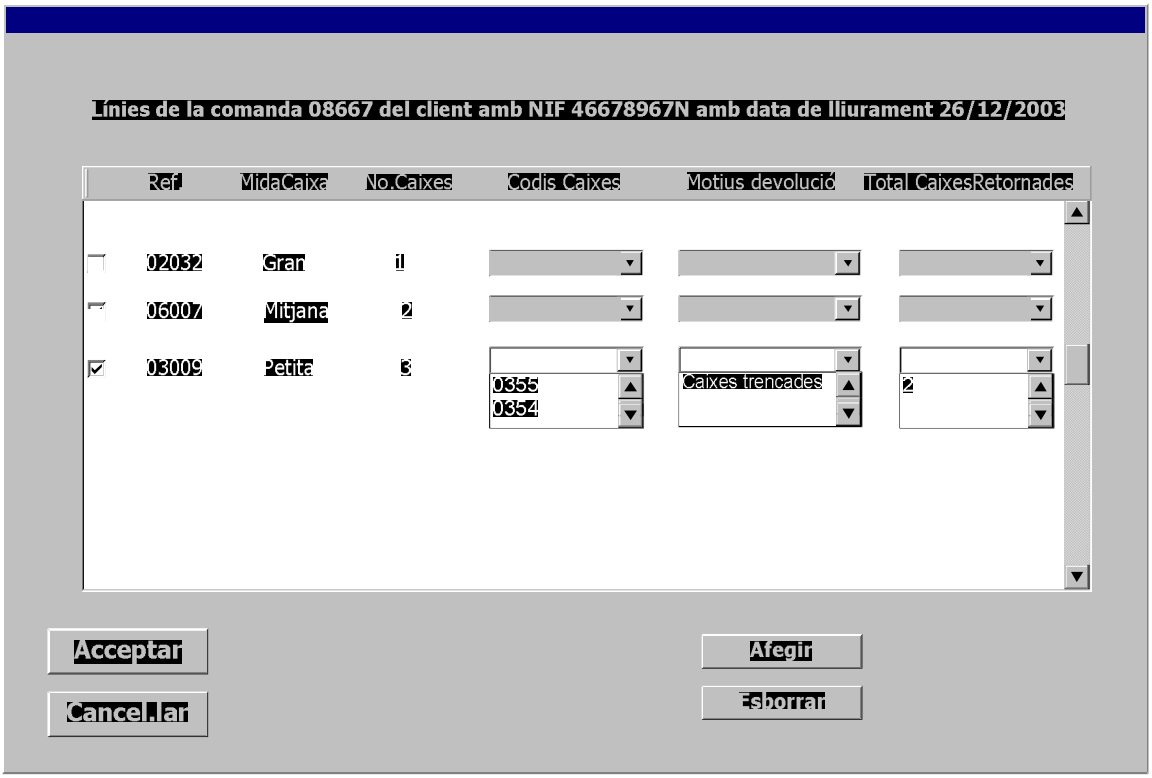
Com es pot apreciar en aquesta interfície, les característiques de les interfícies del sistema que són llistats d'objectes per seleccionar un, són:

- Botó d'acceptar per a que l'usuari, com en el cas anterior, pugui escollir amb tranquil·litat la comanda que més li convingui.
- També hi ha un botó de cancel·lar per si en un moment determinat l'usuari decideix no continuar amb aquesta tasca.
- Las caselles de selecció, estaran dissenyades per a que només pugui estar una de sola activa ja que sols es pot escollir un objecte. D'aquesta forma evitarem que es produeixin conflictes en el sistema.
- La barra de desplaçament vertical facilitarà la feina als usuaris ja que podran desplaçar-se pel llistat amb molta més rapidesa.
- La interfície està dissenyada per a que sigui molt intuïtiva i fàcil d'entendre, pensem que els usuaris que faran servir les diferents interfícies no estan gens habituats als sistemes informàtics[41]. Així doncs, la capçalera porta un títol indicatiu del que mostra la finestra amb lletres destacades per a que no passi desapercebut. El mateix passa amb els botons d'acceptar i cancel·lar. I finalment, cadascun dels camps que formen els objectes estan indicat amb un títol significatiu per saber quins valors contenen.

L'últim model d'interfície que hi ha en el sistema és la que es representa amb un sèrie de camps actius per a que l'usuari pugui introduir les dades que se li demanen. Aquest model d'interfície també pot contenir algun llistat de objectes per seleccionar, però bàsicament la seva importància rau en el fet que serveix de formulari per a que l'usuari pugui introduir les dades que el sistema necessita per a rea-

[41] Mirar apartat 1.7 "Especificació dels requisits de la interfície d'usuari" del capítol 1.

litzar segons quins processos. Aquest és, per exemple, el cas de la interfície que representa a la classe de frontera *FinestraFormulariComanda*[42]



En aquest cas les característiques de la interfície anterior també es compleixen, però cal afegir que:

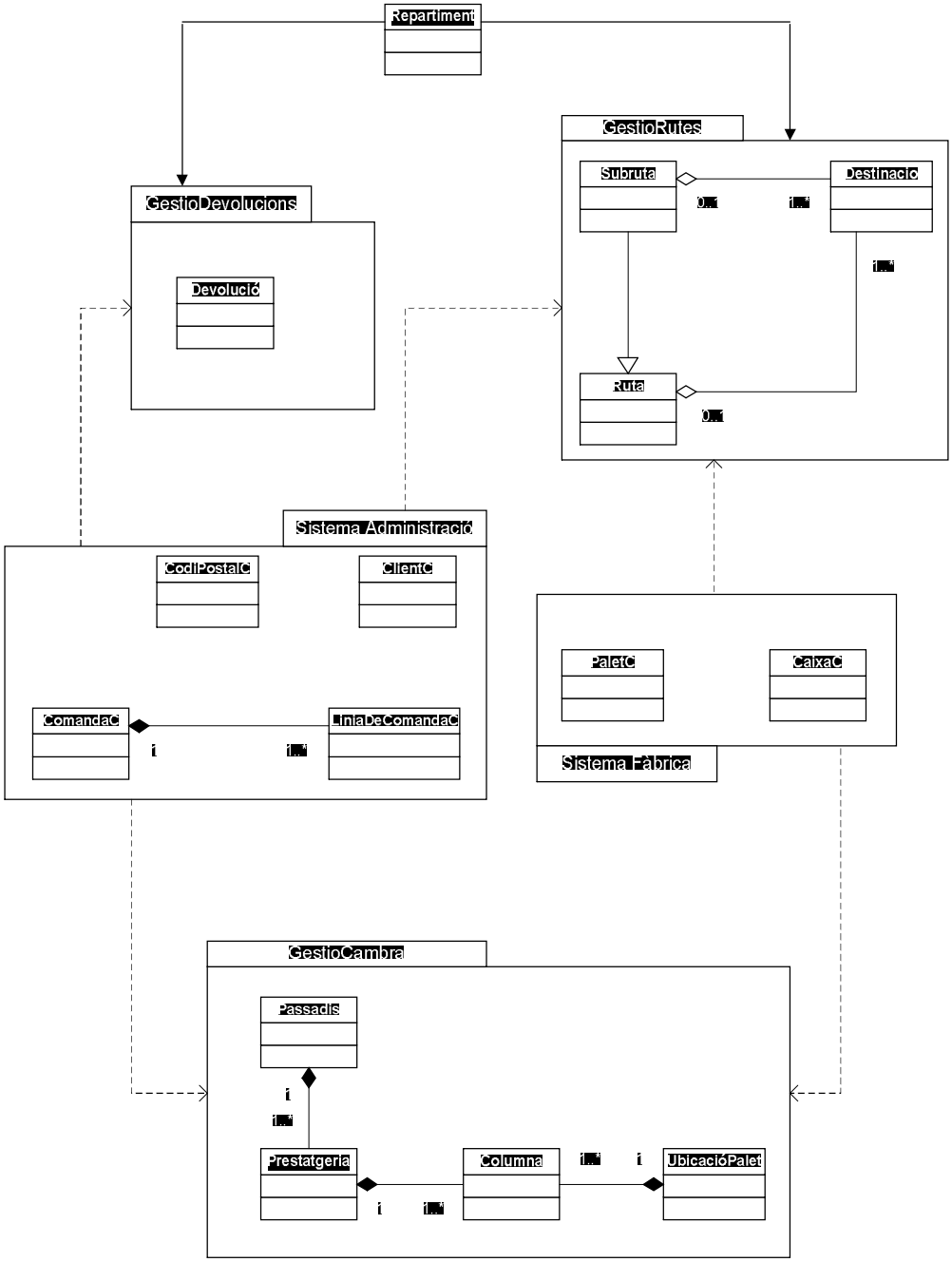
- En aquest cas en particular, la finestra s'activa a partir d'un altre finestra on l'usuari introdueix les dades que apareixen en el títol. Això és molt aclaridor per l'usuari, ja que li dona la sensació de continuar amb la feina que havia començat en la finestra anterior, i no té la sensació que al canviar de finestra també fa un canvi de feina.
- Pel que fa als menús emergents on l'usuari ha d'introduir les dades, aquests s'activen un cop l'usuari ha seleccionat l'objecte desitjat, mentre que els demés menús emergents de la resta de objectes continuen desactivats. Això ens assegura que l'usuari tingui poques possibilitats de equivocar-se al introduir les dades, evitant conflictes en el sistema. Si l'usuari desactiva l'objecte activat el menú emergent torna a passar al estat inactiu.
- També es facilita la tasca si al introduir les dades hem d'acceptar-les (boto afegir) i tenim l'opció d'esborrar les dades introduïdes abans d'acceptar-ho tot (botó acceptar).
- Un cop introduïdes i afegides les dades en un menú emergent, aquestes passen a la part de sota del menú per tenir-les a la vista fins que no acabem la feina. D'aquesta manera tenim la feina controlada en tot moment.

4.6. Disseny dels subsistemes:

Al llarg de l'etapa de disseny no s'ha obtingut cap subsistema nou que no coincideixi amb els paquets d'anàlisi, a més a més, s'ha pogut confirmar que els paquets d'anàlisi coincideixen amb els subsistemes, com per exemple amb el diagrama es-

[42] Mirar la descripció textual de la classe de frontera en el cas d'ús 22 a l'apartat 2.2 "Identificació de les classes de frontera" del capítol 2.

tàtic de disseny on es fa molt evident què els possibles subsistemes coincideixen amb el paquet d'anàlisi.
Per a cada subsistema obtenim els diagrames estàtics de classes d'entitat, tal i com és pot veure al següent diagrama. Aquests diagrames estan simplificats sense mostrar ni les operacions, ni els atributs per tal de fer més comprensible el diagrama dels subsistemes.



Com podem veure les classes d'entitat que pertanyen a cada un dels subsistemes, són les mateixes que als paquets d'anàlisi. Els casos d'ús també seran els mateixos que als paquets d'anàlisi.

5. CONCLUSIONS

Com a primera conclusió podem dir que els objectius del TFC s'han assolit amb força èxit, tenint en compte que el TFC està basat en un cas no real i que segurament falten molts detalls a tenir en compte en el moment de dissenyar la documentació inicial. Això ho he constatat al llarg del desenvolupament del TFC, ja que en moltes ocasions he tingut que introduir noves dades per poder continuar. Tot i això, encara quedaria tota la part de implementació i proves per dir que el disseny ha sigut un èxit. En tot cas, segurament es tindrien que modificar alguns elements, però aquesta és la gràcia del disseny orientat a objectes.

La part de recollida de requisit la destacaria com la part fonamental a l'hora de dissenyar un sistema. Fent un bon treball en aquesta etapa, ens assegurem gran part de l'èxit del disseny i ens clarifica molt el treball de la resta d'etapes. Per a que aquesta etapa sigui el més profitosa possible, crec que hi ha d'haver una bona recopilació d'informació pel que fa al sistema que es vol dissenyar. Pot ser no és tant la quantitat d'informació, sinó saber recollir aquella informació realment important. Massa informació ens pot arribar a confondre, amagant-nos el que és realment important pel disseny. En aquesta recollida d'informació també crec que és molt important tenir en compte tots els punts de vista d'aquells usuaris finals, que d'una forma o altre, manipularan el sistema.

El sistema dissenyat només és aconsellable per empreses grans amb un gran volum de clients minoritaris, ja que la inversió inicial i manteniment del sistema són elevats. No estem parlant només de programari, sinó també de tota la inversió i manteniment del maquinari de la cambra de congelació com és el cas dels robots.

Finalment comentar que la utilització de l'eina Visió per generar els diagrames UML et limita molt la forma de treballar, ja que, entre d'altres coses, només té en compte la generació de diagrames a partir d'uns llenguatges de programació concrets. Pot ser he trobat a faltar que no et deixi crear diagrames de forma genèrica sense tenir que limitar-te a algun dels llenguatges de programació que contempla. De totes maneres, si ens limitem als llenguatges de programació, hem sembla una barbaritat que no inclogui un llenguatge com Java, però tractant-se d'un producte de Microsoft té la seva lògica.

GLOSARI

***Caixa**:* Conjunt d'unitats d'un mateix tipus de producte. La mida de la caixa pot ser gran, mitjana o petita, depèn del producte que conté.

***Capacitat màxima del camió**:* Capacitat màxima d'un camió en *caixes* de mida gran, tenint en compte que dues *caixes* mitjanes ocupen el lloc d'una gran, dues *caixes* petites el lloc d'una mitjana i per tan, quatre *caixes* petites el lloc d'una gran.

***Columna**:* Conjunt de dues *ubicacions* disposades una sobre l'altre[43].

***Comanda**:* Conjunt de *línies de comanda* d'un mateix client amb la mateixa *data de lliurament*.

***Data actual**:* Data del rellotge intern del sistema.

***Data de lliurament**:* Data en que han de ser servides les *caixes* que pertanyen a una mateixa *comanda*. Aquesta data ha de coincidir amb la *data de lliurament* del *repartiment* al que pertany la *comanda*.

***Destinació**:* Domicili social d'un client associat a una *ruta* determinada.

***Devolució**: Caixes* retornades pel client d'una *comanda* determinada.

***EstacioOperadorLogistic**:* Computadora que fa servir l'**Operador logístic**.

***Estat del repartiment**:* Quan un *repartiment* està complert (no accepta més caixes) diem que el seu estat és *tancat* i quan no està complert (accepta més caixes) diem que el seu estat és obert.

Línia de comanda: Línia que especifica el número de caixes d'un mateix producte que un client demana, el tipus de producte que contenen en una comanda determinada[44].

Nou client: Client que no està donat de baixa com a client i no té assignada cap *ruta*.

Palet: Conjunt de *caixes* d'un mateix producte.

Passadís: Zona per on es mou un robot determinat, la qual està delimitada per dos *prestatgeries* (dreta i esquerra) de les quals és responsable el mateix robot.

Prestatgeria: Conjunt de *columnes* que delimiten un *passadís* pels costats dret i esquerra.

Repartiment: Conjunt de *comandes* amb la mateixa *data de lliurament* i que pertanyen a la mateixa *ruta*.

Ruta: Conjunt de *destinacions* amb codis postals iguals als codis postals del recorregut que té assignat un transportista determinat.

Subruta: Subconjunt d'una *ruta* al que li pertanyen part de les *destinacions* de la *ruta* que la inclou. Una *subruta* només pot tenir assignat un únic *repartiment* que tindrà la mateixa *data de lliurament* que un dels *repartiments* assignats a la seva *ruta* mare.

[43] Mirar Annex 2.
[44] Mirar Annex 6.

***Ubicació**:* Els dos buits de cadascuna de les *columnes* que contenen els *palets*. Cada *ubicació* conté un únic *palet* i sempre li correspon un *palet* del mateix producte[45]. Les *ubicacions* són de dos tipus "SOT" i "SOB". La *ubicació* "SOT" és la que es troba a la part de sota de les *columnes* i la *ubicació* "SOB" és la que es troba a la part de dalt de les *columnes*.

[45] Mirar Annex 2.

BIBLIOGRAFIA

Bruegge, B.; H. Dutoit, A. *Ingeniería del software orientado a objetos.* México, Ed. Pearson educación, 2002.

Campderrich, B. *Enginyeria del programari I .* Barcelona, UOC (XP00/05007/00298).

Jacobson, I.; Booch, G.; Rumbaugh, J. *El proceso unificado de desarrollo de software.* Madrid, Ed. Perason Educación, S.A., 2000.

Jacobson, I.; Booch, G.; Rumbaugh, J. *El lenguaje unificado de modelado.* Madrid, Ed. Addison Wesley Iberoamérica, 1999.

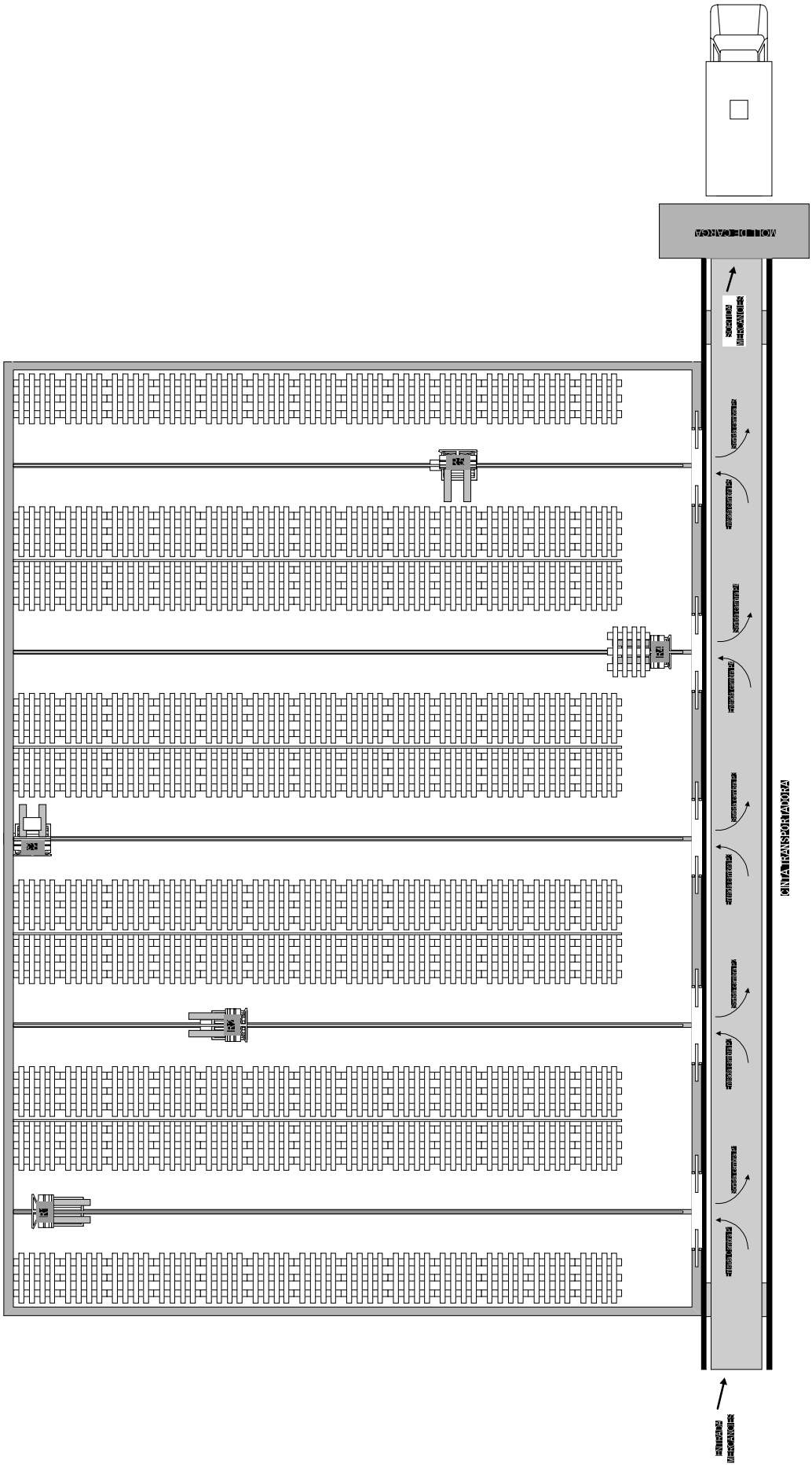
Larman, C. *UML Y PATRONES: una introducción al anàlisis y diseño orientado a objetos y al proceso unificado.* Madrid, Ed. Pearson Educación, S.A., 2003.

Marco, Ma. J. [et al.] **.** *Enginyeria del programari I I.* Barcelona, UOC (XP01/05008/00271), febrer 2002, segona edició.

Pressman, R. *Ingeniería del software: un enfoque práctico.* Madrid, Ed. McGraw Hill, 2002, (5ª edición).

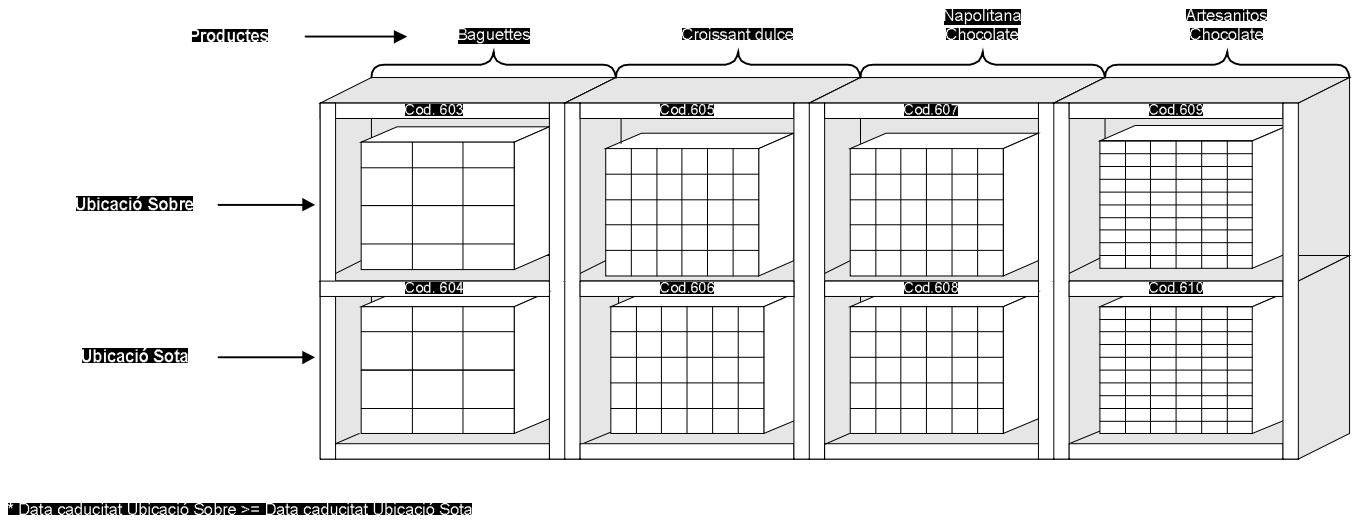
Sistac, J. [et al.]**.** *Bases de dades I.* Barcelona, UOC (P02/05002/00092), septembre 2002, segona edició.

ANNEX 1



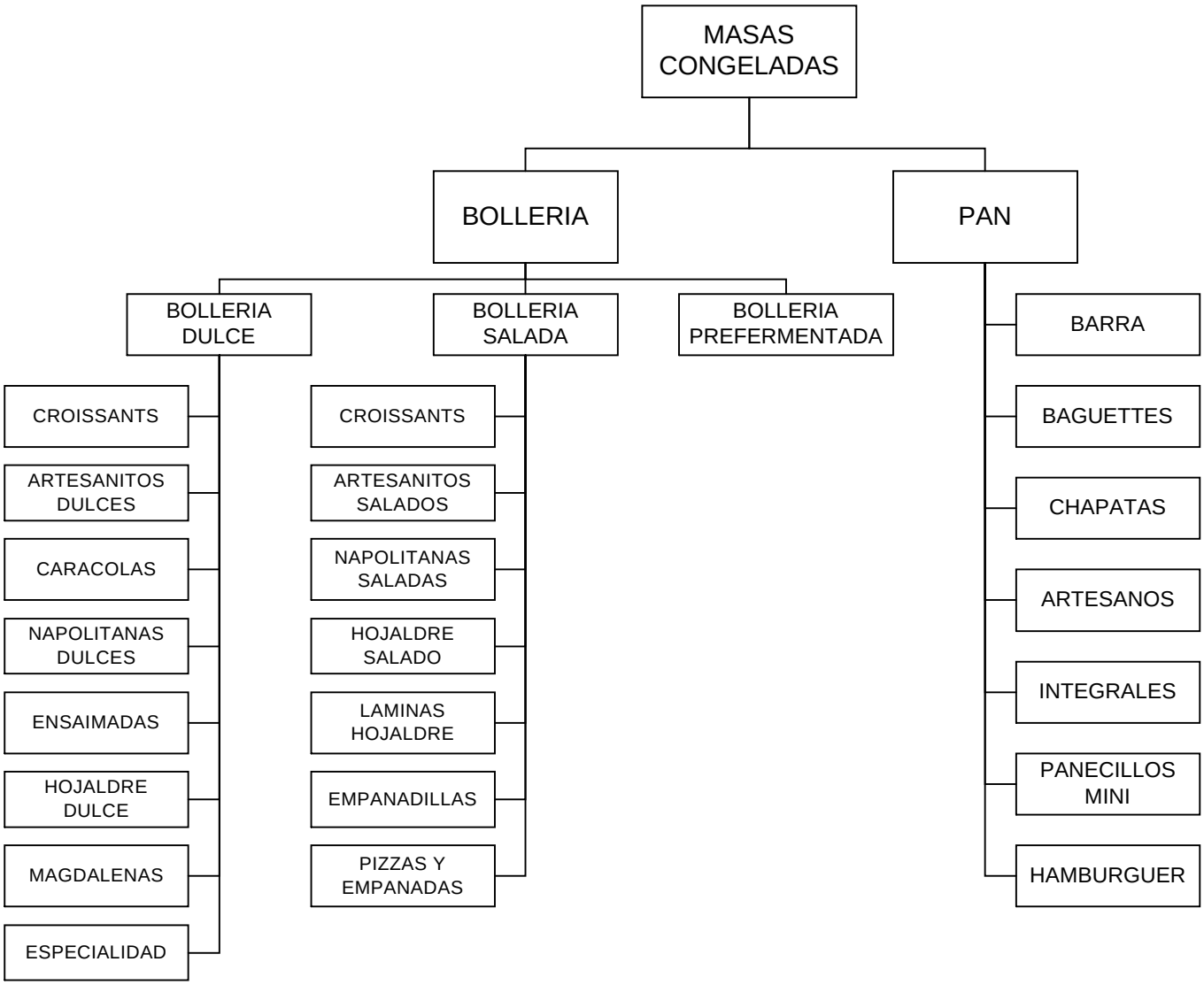
Esquema cambra de congelació

ANNEX 2



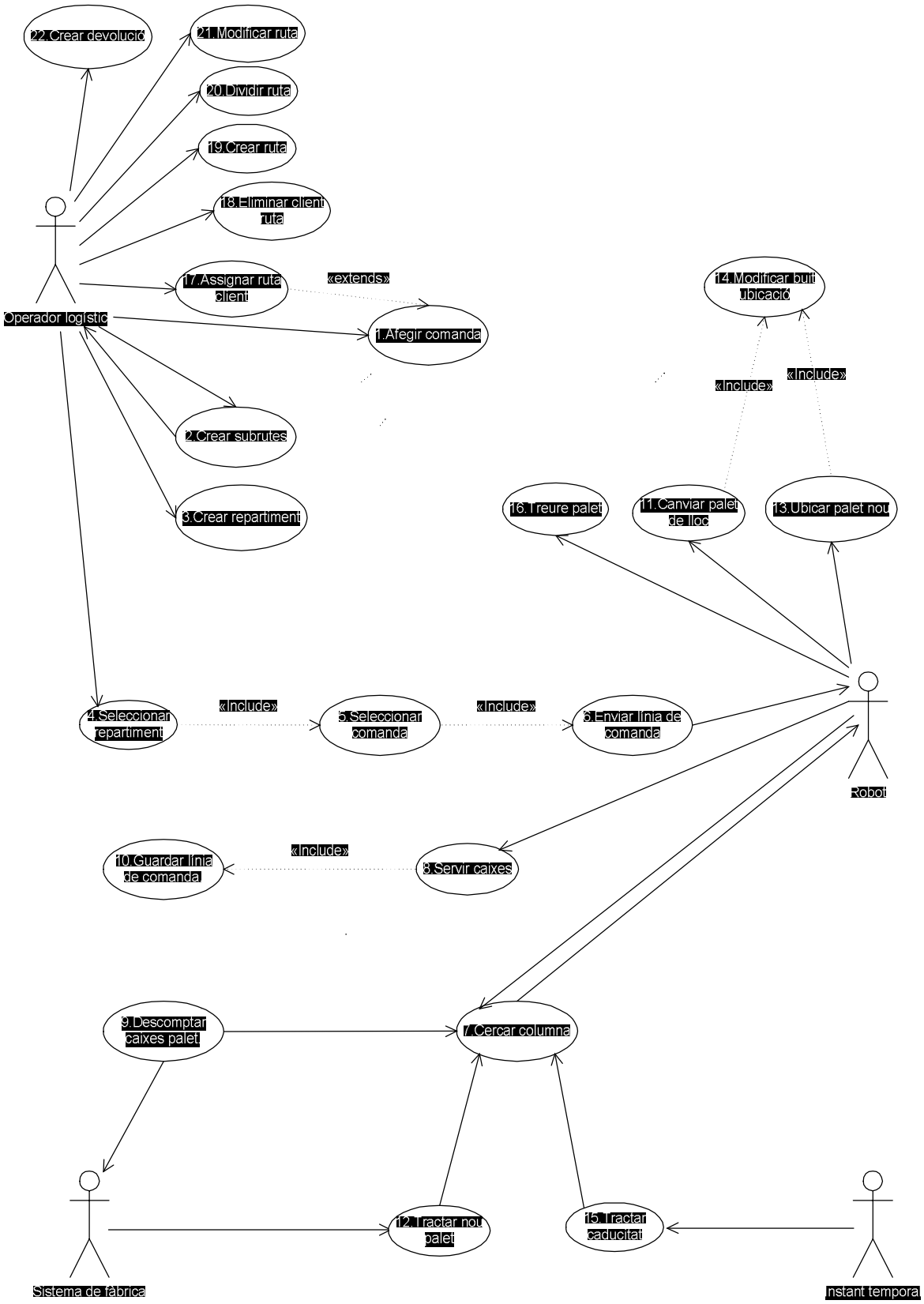
Esquema exemple prestatgeria

ANNEX 3



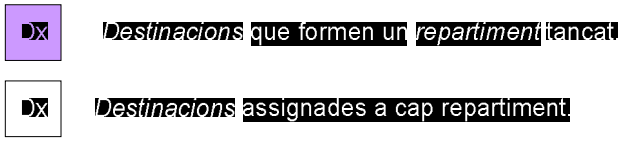
Organigrama de grups, famílies i subfamílies masses congelades

ANNEX 4


22.Crear devolució
21.Modificar ruta
20.Dividir ruta
19.Crear ruta
18.Eliminar client ruta
17.Assignar ruta client
«extends»
Operador logístic
1.Afegir comanda
14.Modificar buit ubicació
«Include»
«Include»
2.Crear subrutes
3.Crear repartiment
16.Treure palet
11.Canviar palet de lloc
13.Ubicar palet nou
4.Seleccionar repartiment
«Include»
5.Seleccionar comanda
«Include»
6.Enviar línia de comanda
Robot
10.Guardar línia de comanda
«Include»
8.Servir caixes
9.Descomptar caixes palet
7.Cercar columna
12.Tractar nou palet
15.Tractar caducitat
Sistema de fàbrica
Instant temporal

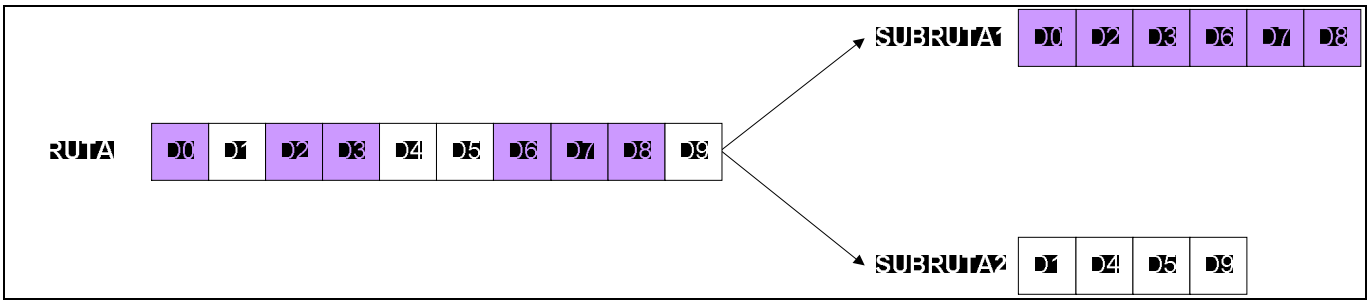
ANNEX 5

En el següent exemple es mostra com es desenvolupa el cas d'ús 2.Crear subrutes amb les seves alternatives al procés principal **12a**, **19a** i **21a**:



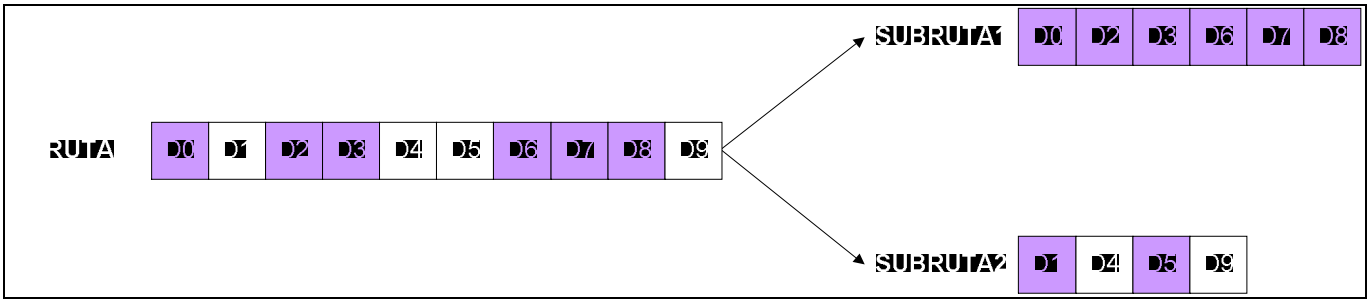
Tenim una *ruta* inicial amb un *repartiment* tancat:

PAS1: Creem dues *subrutes* a partir de la *ruta* inicial:

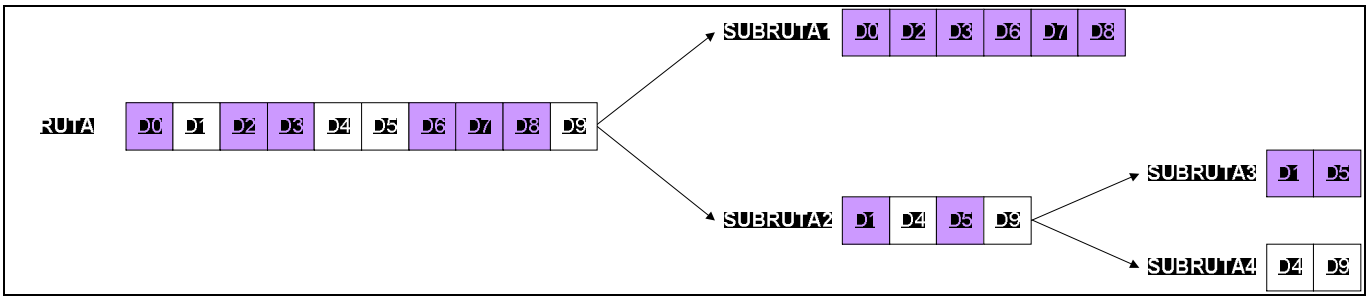


La **SUBRUTA1** té un únic *repartiment* i aquest està tancat.
La **SUBRUTA2** no té cap *repartiment* assignat.

PAS2: A la **SUBRUTA2** s'ha tancat un *repartiment*:

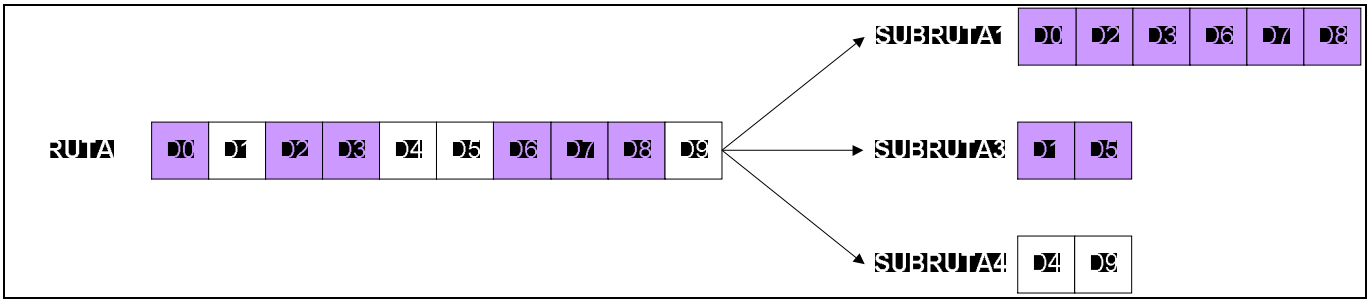


PAS3: Creem dues *subrutes* a partir de la **SUBRUTA2**:



La **SUBRUTA3** té un únic *repartiment* i aquest està tancat.
La **SUBRUTA4** no té cap *repartiment* assignat.

PAS4: Esborrem la **SUBRUTA2** ja que les *destinacions* que formen el *repartiment* tancat i les que encara no tenen cap *repartiment* assignat, es troben en la **SUBRUTA3** i **SUBRUTA4**:



Aquestes són les *subrutes* de la *ruta* inicial amb els seus *repartiments* les quals guardarem a la base de dades.

ANNEX 6

Nom i Logo de l'empresa

Id. Comanda: 23456

DATA:	27/01/2004
DATA LLIURAMENT:	29/01/2004

CLIENT

NIF:	38906921V
NOM:	Bon Forn
RAÓ SOCIAL:	C/ Creu Coberta, 39.
POBLACIÓ:	Barcelona
C.P:	08014
PROVINCIA:	Barcelona

Ref	No. Caixes	Mida Caixa
02032	1	Gran
06007	2	Mitjana
03009	1	Petita

Línia de comanda

Total de caixes del producte al que fa referència la línia de comanda

Referència del producte que identifica a la línia de comanda

Mida de les caixes del producte al que fa referència la línia de comanda

TOTAL: **3'25**

COMANDA NO SERVIDA

Exemple de comanda