



# **Tècniques de desenvolupament de programari**

---

PAC3: Tractament d'excepcions i RMI

Luis Belmonte Sain  
Raúl Peña

**12/11/2014**

## ÍNDEX

1.	Introducció .....	3
1.1.	Requisits .....	3
1.2.	Presentació de subsistemes .....	3
2.	Assignació de membres per subsistema .....	4
3.	Subsistema de manteniment .....	6
3.1.	Descripció textual de les funcionalitats .....	6
3.2.	Patró Model–View–Controller . Diagrama de classes i jerarquies UML. ....	8
3.2.1.	Diagrama de classes i jerarquies UML.....	9
3.2.2.	Classes Controller .....	12
3.2.3.	Classes Gestores .....	13
3.3.	Diagrama de seqüència .....	17
3.3.1.	Diagrama de seqüència alta taller.....	18
3.3.2.	Diagrama de seqüència alta producte. ....	19
3.4.	Variacions respecte el prototip inicial.....	20
3.4.1.	Pantalla CRUD (genèrica) .....	20
3.4.2.	Exemple pantalla de selecció .....	22
3.4.3.	Alguns detalls a afegir com millores en els prototips inicials.....	22
4.	Subsistema de control de flux .....	23
4.1.	Funcionalitat general.....	23
4.2.	Diagrama de classes i jerarquia UML .....	27
4.3.	Diagrames d'interacció.....	28
4.4.	Canvis en els prototip d'interfície gràfica inicials.....	32
5.	Subsistema de llistats i estadístiques .....	34
5.1.	Descripció general del subsistema .....	34
5.2.	Diagrama de classes i jerarquia UML .....	35
5.3.	Diagrames d'interacció.....	38
6.	Subsistema de connexió.....	41
6.1	Descripció textual de les funcionalitats .....	41

6.2	Diagrama de classes i jerarquies UML.....	43
6.3	Diagrama de seqüència .....	45
6.4	Variacions respecte el prototip inicial.....	46
7.	Reutilització .....	48
7.1.	Interfície gràfica .....	48
7.2.	Accés a disc/dades .....	48
7.3.	Llibreries de l'aplicació .....	49
7.4.	Llibreries de Java .....	49
8.	Relació de classes .....	453
9.	Disseny de la persistència.....	53
9.1	Descripció general del model de persistència.....	5453
9.2	Descripció del diagrama ER versió 1 .....	54
9.3	Descripció del diagrama ER versió 2 simplificada .....	56
9.4	Scripts SQL (DDL amb DML) sobre la versió 1 del ER .....	57
10.	Annex.....	74
10.1	Fitxes CRC .....	74
10.2	Planificació detallada .....	77

# 1. Introducció

## 1.1. Requisits

El projecte que s'inicia ara pretén gestionar el material en estoc del magatzem central que serveix als tallers que formen part d'aquesta cadena de tallers en tot el territori català ja que aquesta empresa només treballarà en aquest territori. Aquest projecte es denominarà 'eTallerStock' i l'empresa que es dedicarà la importació i la venda de aquests materials es 'TotCatCars'.

D'aquesta manera, l'objectiu principal del projecte és habilitar un sistema únic que sigui fiable i que pugui garantir i faciliti el seguiment de l'estoc a tots els tallers de la cadena.

Per tal de fer aquesta part del projecte, ens basarem en dos apartats:

- Les especificacions inicials que s'han establert pel Departament d'Informàtica i Multimèdia en el document "Enunciat del projecte 'eTallerStock'".
- El document corregit com a solució oficial i que es va posar juntament a l'enunciat d'aquesta PAC a l'espai comú de treball BSCW. En aquest apartat, cal dir que la solució oficial proposada va ser la nostra, per tant aquest document es pot considerar continuació de l'anterior.

## 1.2. Presentació de subsistemes

La descripció dels subsistemes proposada a l'enunciat de partida d'aquesta fase no ha sofert cap variació amb respecte a la nostra primera proposta ja que el prototip escollit va ser el del nostre propi grup 'The JavaClasses Team'.

## 2. Assignació de membres per subsistema

Per aquesta part del projecte, com que l'organització i divisió dels membres per cada subsistema ens va donar bon resultat, s'ha decidit continuar amb la mateixa divisió, que serà la següent:

	Coordinador	S. Manteniment	S. Control de fluxe	S. Llistat i estadístiques	S. Connexió
Fase 1	Luis Belmonte	Raúl Casado	José Vicente	Ramón Giribert	Raúl Peña
Fase 2	Raúl Casado	José Vicente	Ramón Giribert	Raúl Peña	Luis Belmonte
Fase 3	José Vicente	Ramón Giribert	Raúl Peña	Luis Belmonte	Raúl Casado
Fase 4	Ramón Giribert	Raúl Peña	Luis Belmonte	Raúl Casado	José Vicente
Fase 5	Raúl Peña	Luis Belmonte	Raúl Casado	José Vicente	Ramón Giribert
Fase 6	Luis Belmonte	Raúl Casado	José Vicente	Ramón Giribert	Raúl Peña

**Nota:** La part del quadre que està en color groc, és la divisió que tindrem per fer aquesta part del projecte



### 3. Subsistema de manteniment

#### 3.1. Descripció textual de les funcionalitats

Tal vegada caldria explicar la relació entre Rols, Menús i Usuaris per tal d'aclarir el sentit de les dades del manteniment. Una explicació ràpida és que els Rols es poden veure simplement com etiquetes que li posem tant a una opció de menú en concret com a un usuari, de tal manera que aquest usuari tindrà accés a tots el menús etiquetats amb el mateix rol que ell. Amb el manteniment de Rols donem de alta aquestes etiquetes i amb els manteniments de menús i usuaris les assignem a cadascú d'ells, després el subsistema de connexió s'encarrega de fer el procés corresponent i mostrar a cada usuari només els menús al que té accés.

D'altra banda en la especificació del subsistema manteniment ja va quedar clar el seu abast i objectiu funcional, que per resumir podríem dir que es donar servei al usuari Administrador per fer les operacions CRUD sobre Usuaris, Menús, Tallers, i Productes.

Des de el punt de vista del disseny s'han de proporcionar les estructures i la descripció de les seves funcionalitats associades, suficients per garantir aquesta funcionalitat, procurant que sigui àgil, consistent, eficaç i escalable.

Des de el punt de vista del disseny orientat a objectes, el nostre programari necessita per el seu funcionament un conjunt de classes i de les seves instàncies els objectes, que siguin el que podríem anomenar fortament persistents i que implementin les principals estructures de dades. El abast del subsistema de manteniment es just les operacions CRUD sobre aquestes objectes. Pera això es considera essencial la identificació correcta d'aquestes classes.

Segons el nostre criteri, aquestes classes i les seves instàncies són les que compleixen el següent:

- És imprescindible garantir la seva persistència abans de poder engegar per primera vegada la nostra aplicació realment en productiu. (el cas del usuari 'admin' i les dades de configuració del programari queden fora del abast d'aquest disseny ja que pertanyen al moment de la instal·lació)
- Són les classes que implementen totes les dades que necessitem introduir en lo que habitualment s'anomena: parametrització i càrrega inicial de dades
- Són les classes tal que els seus objectes no es poden crear o esborrar com a resultat de algun procés automàtic del sistema

En el cas de 'TotCarsCat' hem identificat les següents:

Usuari, Taller, Menú, Producte

Una vegada s'ha aprofundit més en el anàlisi dels cas d'us i en el moment del disseny del diagrama de classes, s'han identificat noves classes que aconsegueixen aquests criteris i que per tant s'han d'afegir al subsistema manteniment, i que són les següents:

Rol, Proveïdor, Grup, Subgrup, Població i Província

D'aquesta manera podem redefinir la funcionalitat del subsistema com:

*Donar servei al usuari Administrador per fer les operacions CRUD sobre Usuaris, Menús, Tallers, Productes, Rols, Proveïdors, Grups, Subgrups, Poblacions i Províncies.*



### 3.2. **Patró Model-View-Controller . Diagrama de classes i jerarquies UML.**

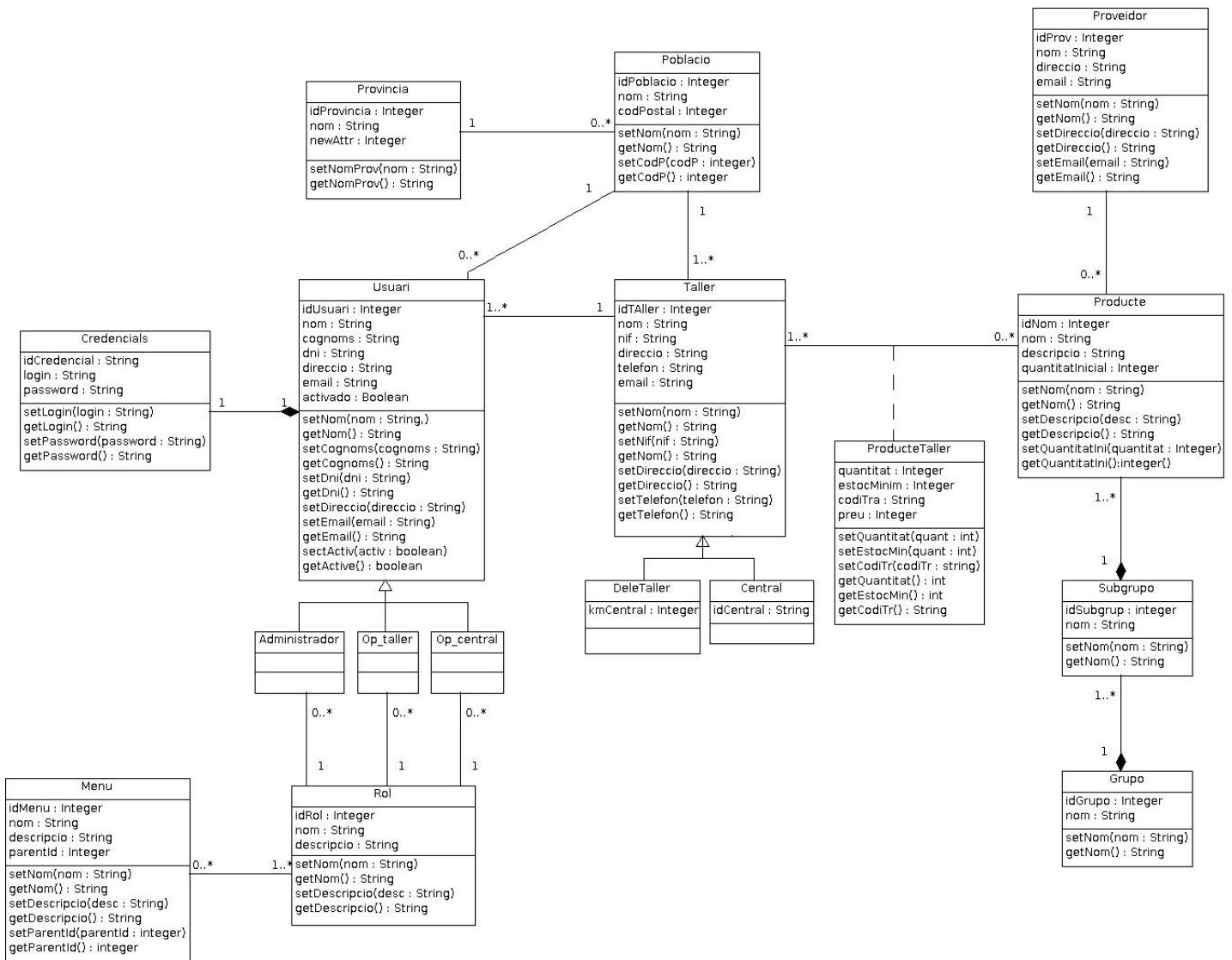
Per tal garantir el desacoblament entre la interfície gràfica i la lògica de aplicació i de negoci s'ha optat per utilitzar el patró MVC, també amb aquesta elecció procurem la senzillesa del manteniment posterior del programari i la seva flexibilitat a l'hora de escalar principalment en la capa de presentació.

Per això caldria dir que en el diagrama de classes detallat només s'han afegit les classes directament implicades en les operacions de CRUD del subsistema. Després en el diagrama sencer s'han afegit les classes controladores i gestores que implementaran el gros del codi (aplicació=events&flux, negoci=regles&procés&SQL) del nostre programari.

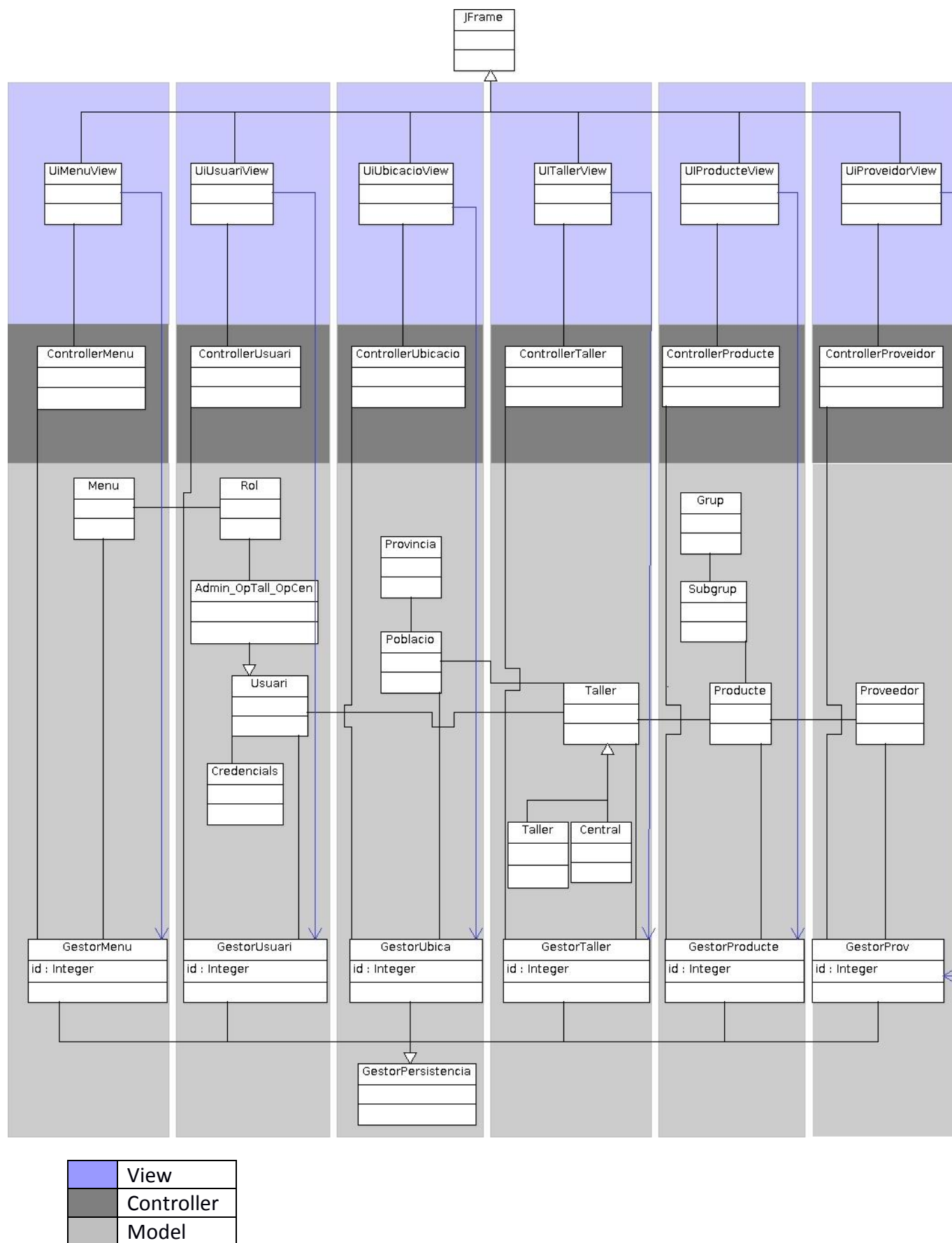
### 3.2.1. Diagrama de classes i jerarquies UML

En primer lloc mostrarem el diagrama de classes , sobre el qual caldria explicar algunes coses:

1. La classe associativa ProducteTaller es podria representar alternativament amb dues relacions 1..\* amb Taller i producte
2. Província podria ser un << Enumeration >> però pensant en facilitar una possible expansió s'ha decidit implementar com classe
3. Credencials existeix com a classe independent, per poder encriptar si fos necessari aquestes dades
4. Les herències tant de Usuari com de Taller podrien ser qüestionables



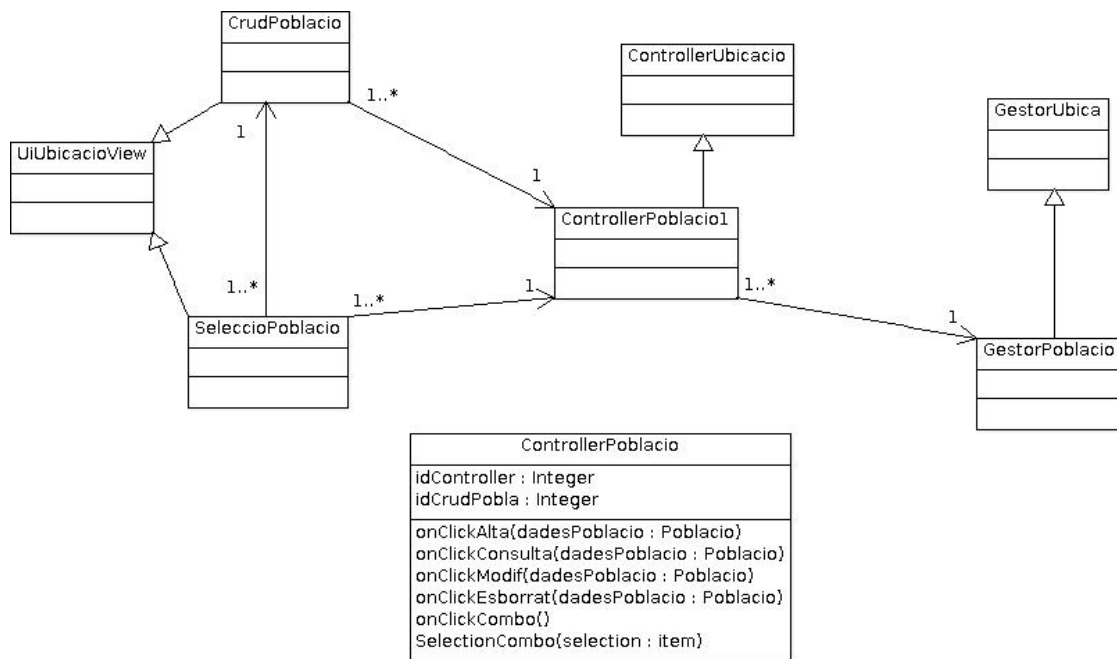
I a continuació, tot el diagrama sencer de jerarquies UML del subsistema de manteniment.



### 3.2.2. Classes Controller

Les classes Controller implementen la lògica de la aplicació que ja hem definit com gestió d'esdeveniments i flux de programa. Encara que no es poden considerar lògica de negoci pròpiament dita, s'ha considerat interessant detallar al menys una d'aquestes classes així com la seva relació amb les seves vistes i gestors.

#### Diagrama de classes per ControllerPoblacio



Com veiem **ControllerPoblacio** gestiona els esdeveniments produïts per la interacció del usuari amb la interfície gràfica de la aplicació però també gestiona el flux de la aplicació en aquest punt.

Per exemple podem de manera informal simular el diagrama de seqüència corresponent al esdeveniment `onClickAlta()` per tenir una idea de com gestionarà la nostra aplicació aquest control del flux de programa: (ens basem en la interfície gràfica [CRUD002](#) = CrudPoblació)

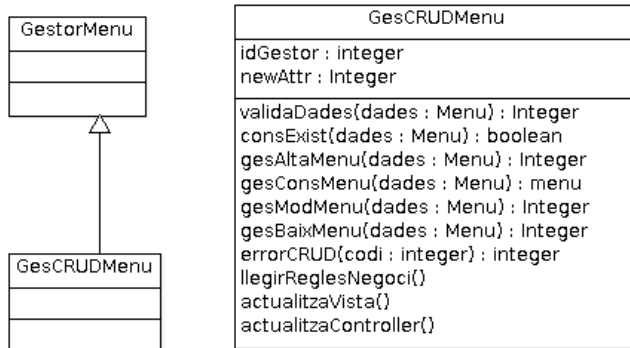
1. El usuari clica al botó Alta i l'esdeveniment és rebut pel Controller (el usuari només ha emplenat el camp nom)
2. Amb les dades de Població existents actualment en el formulari el Controller crida al GestorPoblació sense fer cap validació

3. El gestor aplica regles de negoci i detecta que no hi ha suficients dades per donar de alta una població i genera el missatge corresponent per el controller i per la vista
4. El controller deriva el flux de programa , perquè així esta definit en la nostra lògica d'aplicació, cap a onClickConsulta()
5. El Gestor aplica les regles de negoci i determina que es pot fer una consulta amb les dades introduïdes i com a resultat troba aquesta població i envia missatge a controller i vista
6. La vista mostra el missatge "població ja existent " i les dades de la població

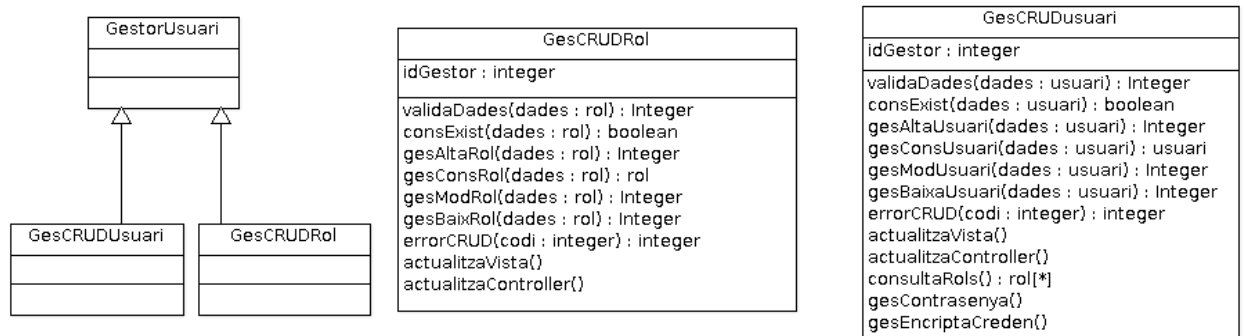
### 3.2.3. Classes Gestores

Donat que les classes gestores encapsulen la majoria de la lògica de negoci s'ha cregut interesant el seu detall :

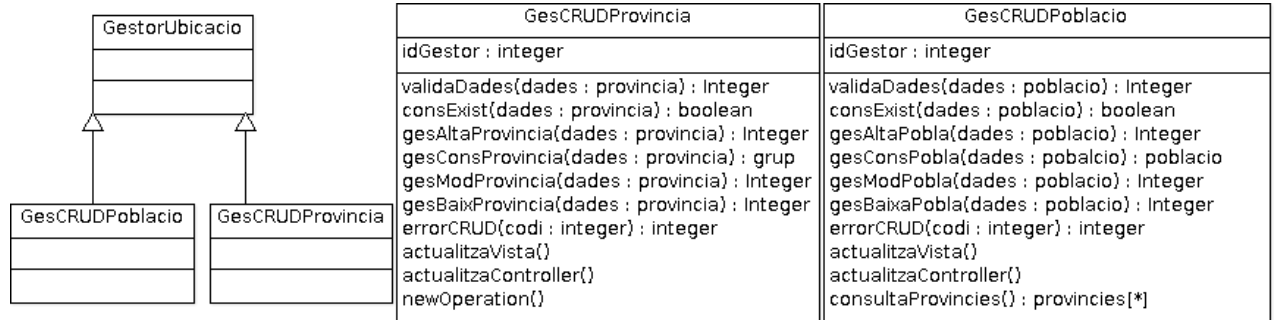
#### Gestor Menús



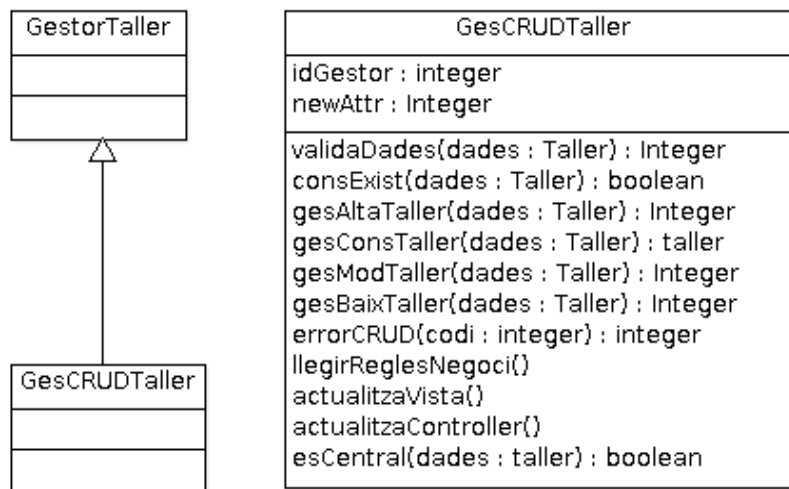
#### Gestors Usuaris i Rols



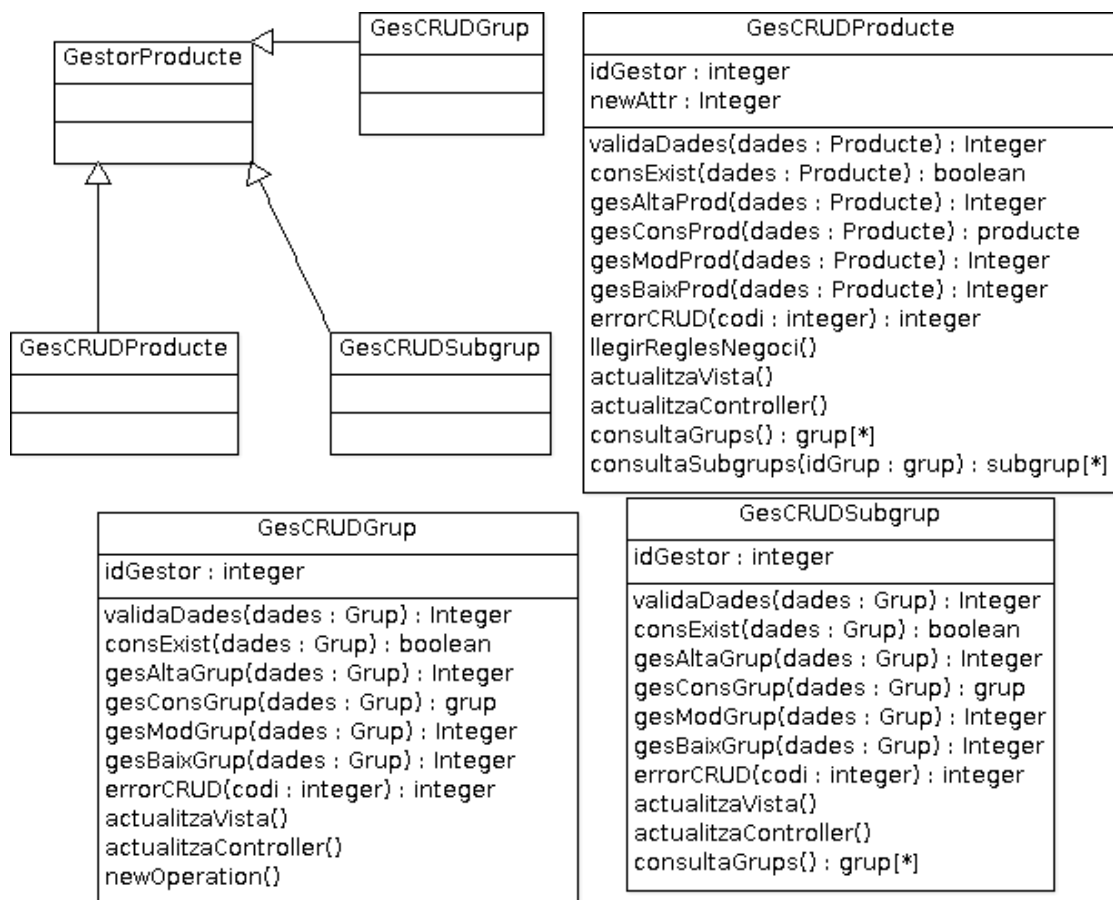
### Gestors Ubicació (Poblacions i Productes)



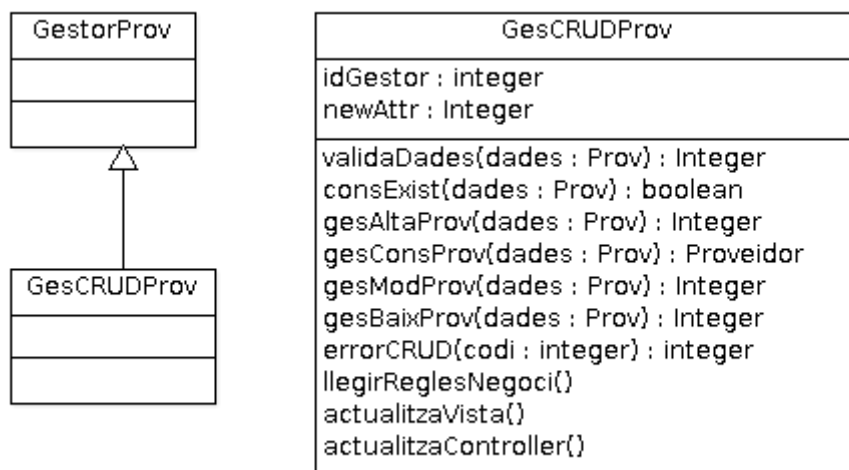
### Gestor Tallers



### Gestor Productes (Producte, Grup i Subgrup)



### Gestor Proveïdors

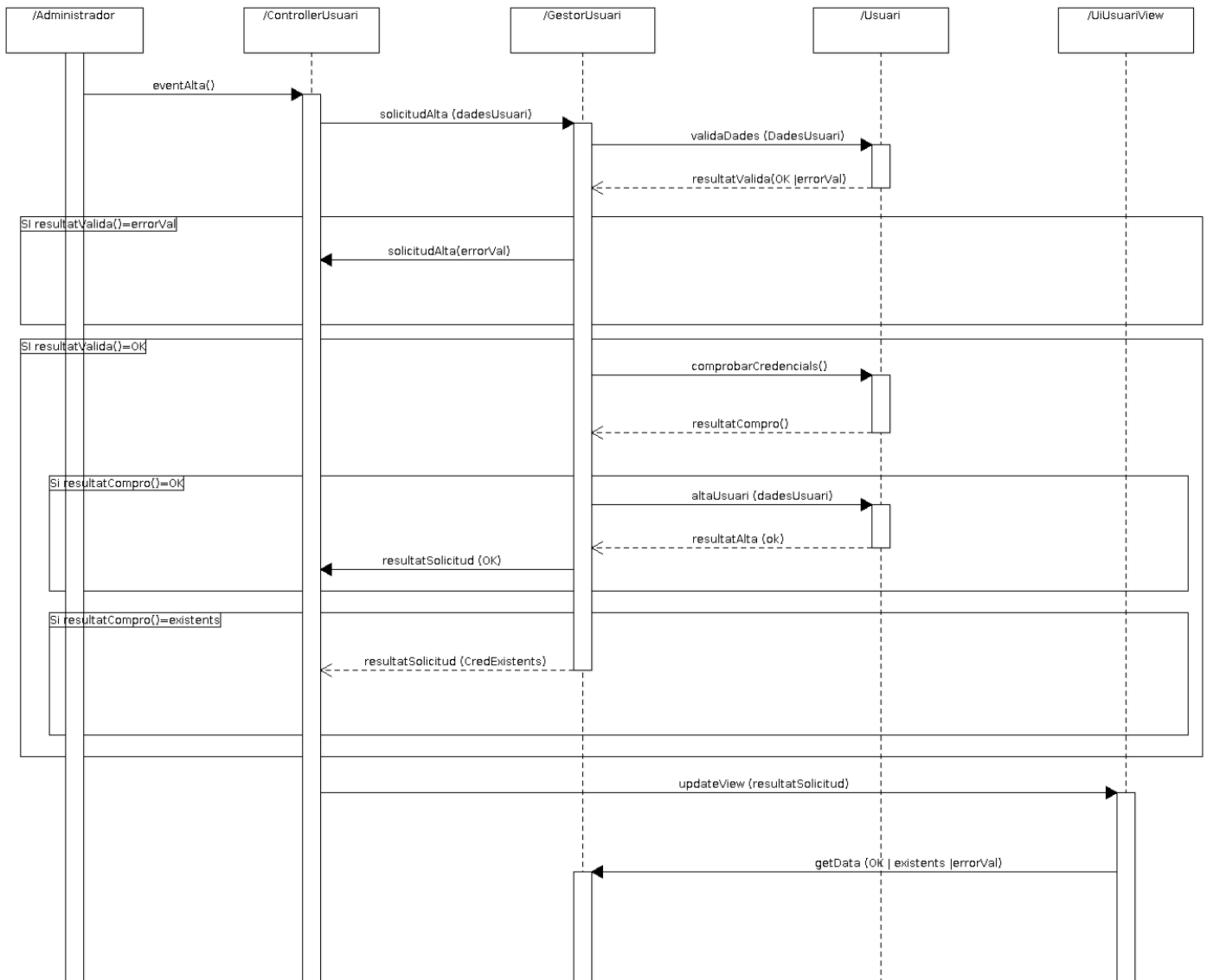




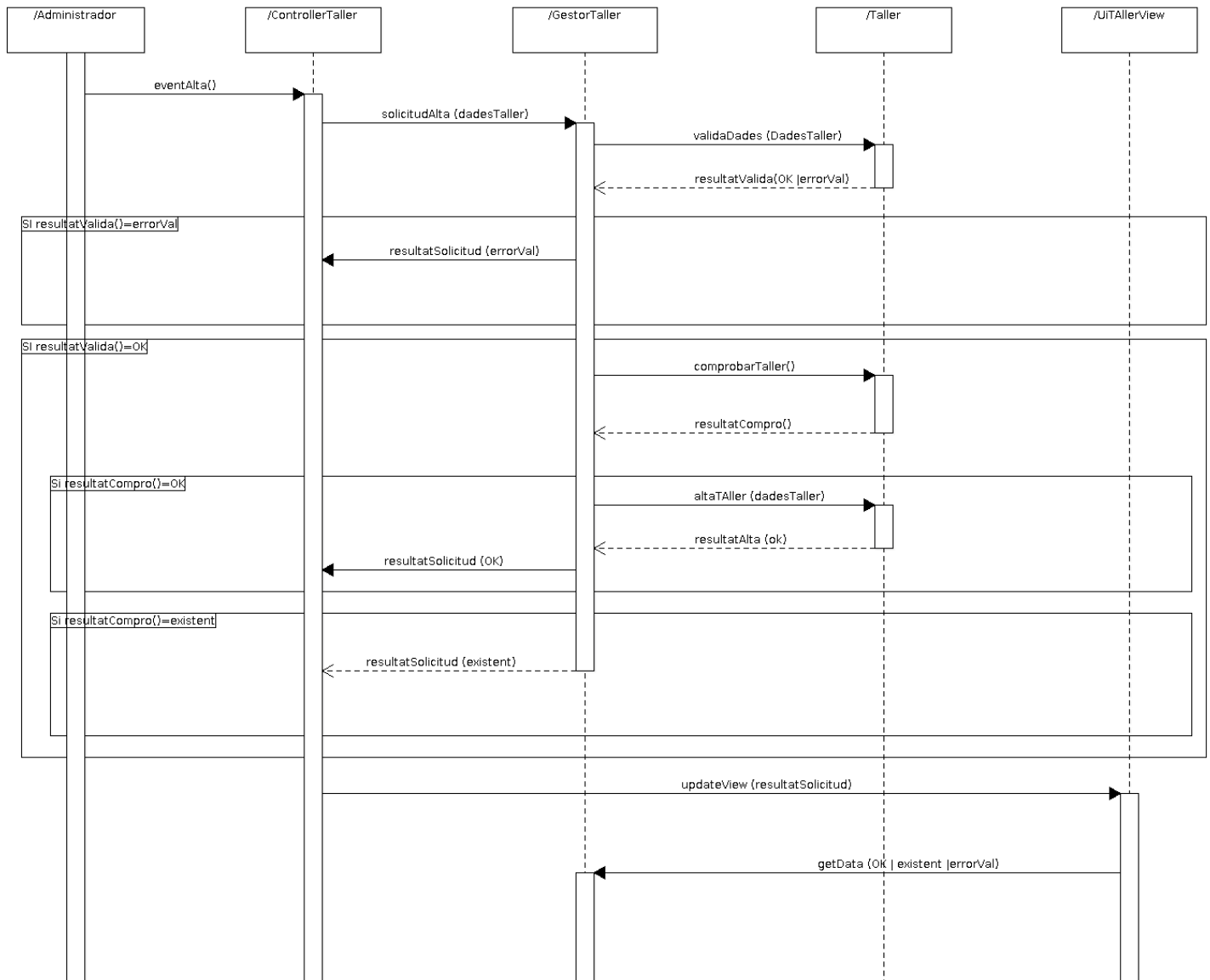


### 3.3. Diagrama de seqüència

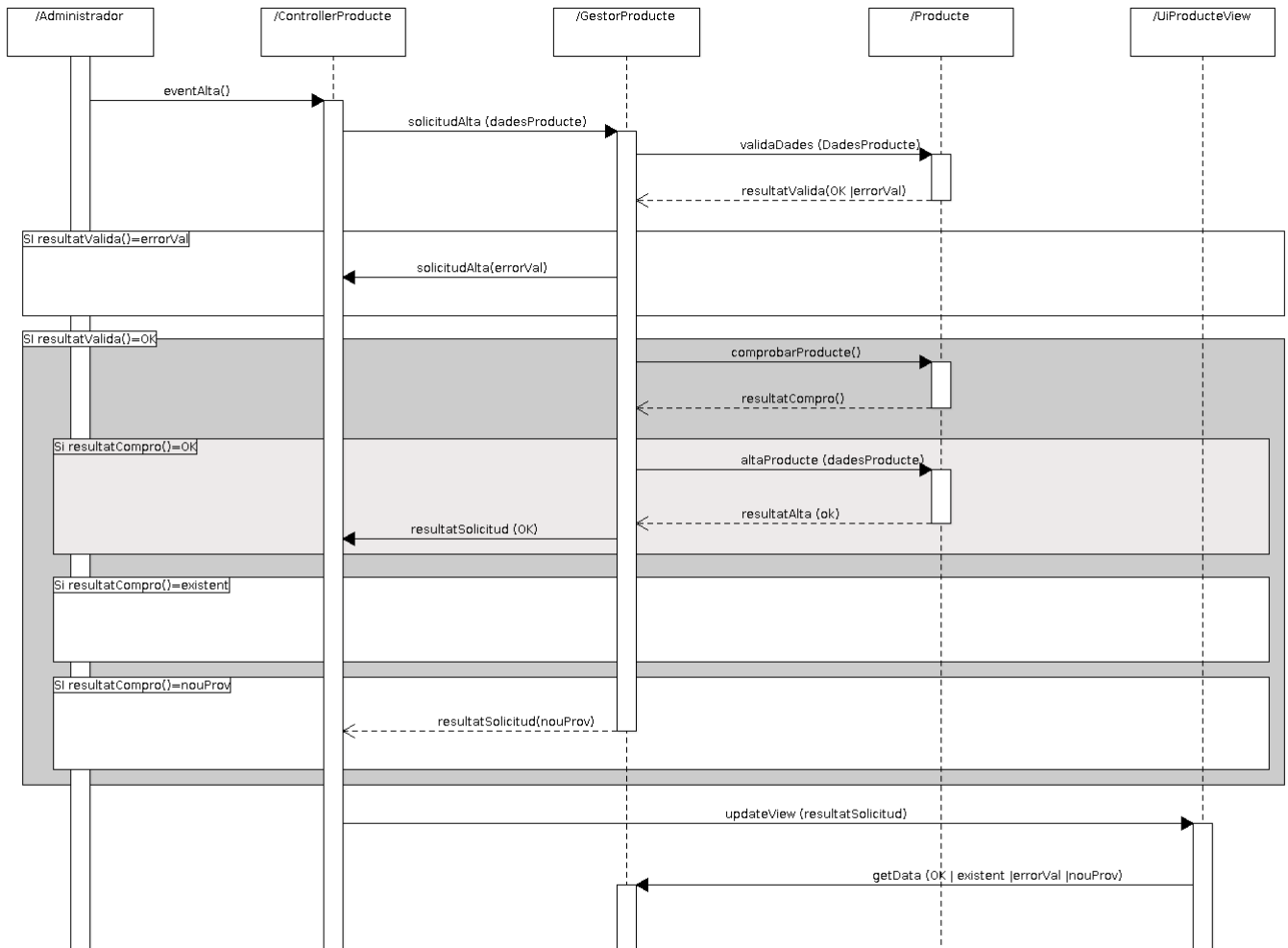
Diagrama de seqüència per a realitzar alta usuari.



### 3.3.1. Diagrama de seqüència alta taller



### 3.3.2. Diagrama de seqüència alta producte.



### 3.4. Variacions respecte el prototip inicial

Donat que es planteja afegir els manteniments de:

Grup i Subgrup, Província i Població, Proveïdor

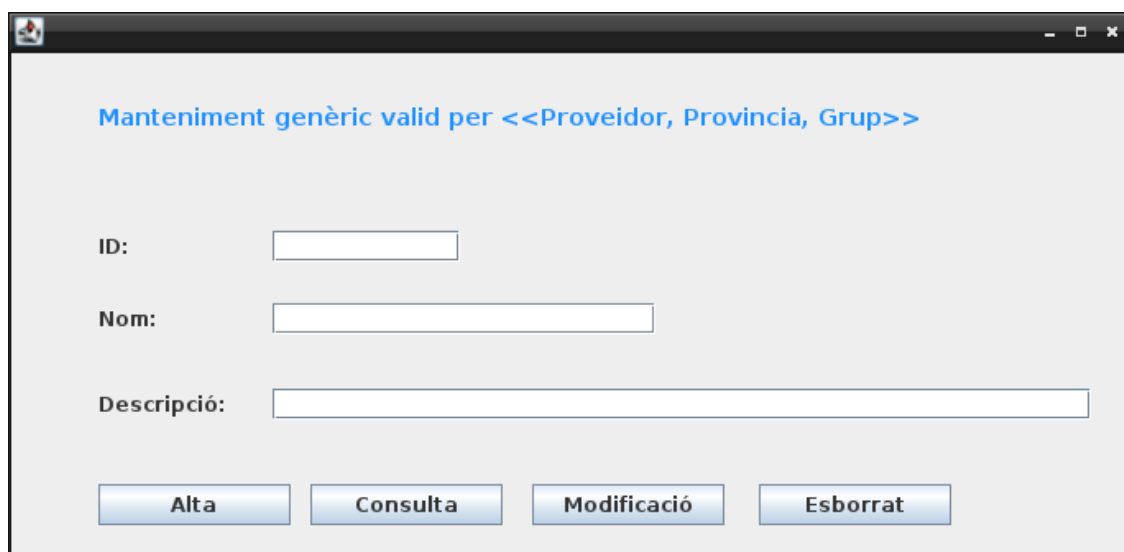
S'han de dissenyar les interfícies gràfiques corresponents. La estratègia es fer servir només dues pantalles per classe:

- Una per tota la funcionalitat del CRUD, de manera que el formulari de dades, que serà molt senzill (com es veu a les imatges) , servirà com a formulari de filtre de dades per la operació de 'Read' en si mateixa, o prèvia al Update o 'Delete', i a la vegada com a formulari per la operació 'Create'.
- Després un altra per triar una selecció concreta quan com a conseqüència d'una operació 'Read' tinguem com a resultat múltiples opcions

#### 3.4.1. Pantalla CRUD (genèrica)

CRUD001

Aquesta pantalla serà vàlida per Proveïdor, Província, Grup i també es proposa per Rol per donar la possibilitat de crear rols diferents que podrien ser necessaris en un futur.



Manteniment genèric vàlid per <<Proveïdor, Província, Grup>>

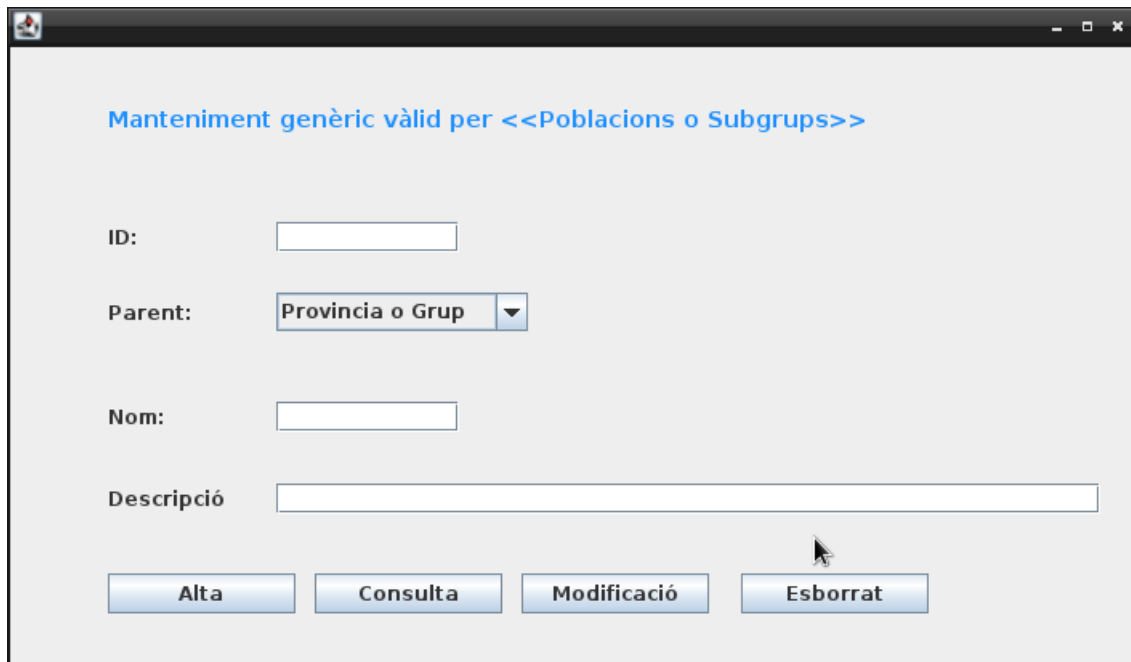
ID:

Nom:

Descripció:

Alta Consulta Modificació Esborrat

## CRUD002

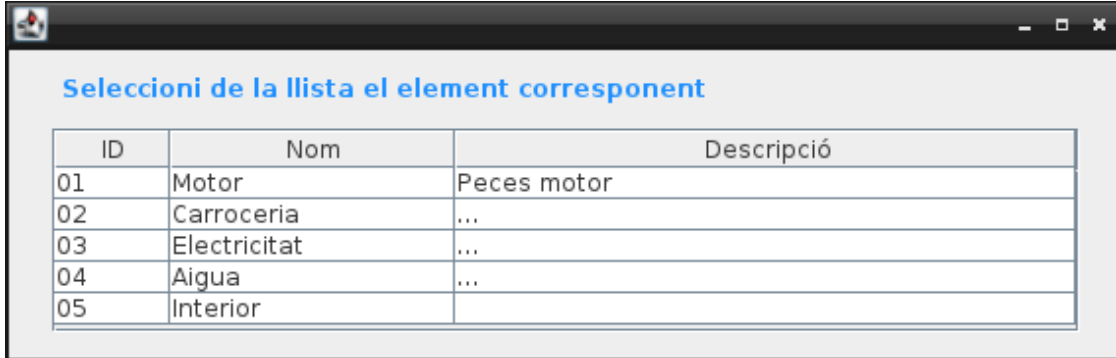


The screenshot shows a web application window with a title bar. The main content area has a light gray background and contains the following elements:

- Title:** "Manteniment genèric vàlid per <<Poblacions o Subgrups>>" in blue text.
- Form Fields:**
  - ID:** A text input field.
  - Parent:** A dropdown menu with the text "Província o Grup" and a downward arrow.
  - Nom:** A text input field.
  - Descripció:** A long text input field.
- Buttons:** Four buttons are located at the bottom of the form: "Alta", "Consulta", "Modificació", and "Esborrat".

### 3.4.2. Exemple pantalla de selecció

CRUD SELECCIO



ID	Nom	Descripció
01	Motor	Peces motor
02	Carroceria	...
03	Electricitat	...
04	Aigua	...
05	Interior	

### 3.4.3. Alguns detalls a afegir com millores en els prototips inicials

En la pantalla de assignació de rol (SM-AU005) i on donem de alta també la contrasenya, es planteja afegir un camp *login* d'aquesta manera el usuari de accés al sistema pot ser un altre diferent al nom de usuari.

En tot el CRUD de Menús fer el camp *Menú parent:* com a desplegable dels mateixos menús.

Afegir a tot el CRUD de Taller les poblacions i províncies com a desplegables

Afegir a tot el CRUD de Producte els camps Grup i Subgrup com a desplegables.

## 4. Subsistema de control de flux

### 4.1. Funcionalitat general

El subsistema continua mantenint els mateixos objectius i gestiona les mateixes tasques ja descrites en l'apartat d'especificació i planificació del projecte. Vol mantenir el control i la gestió del moviment del producte que genera el negoci de 'TotCatCars', les peces de recanvi.

Aquest moviment es un flux que alimenta la cadena de negoci i que comença per l'entrada de les peces a la central, que fa un paper de distribuïdor/centralitzador de les peticions que fan els terminals finals de venda, els tallers. Aquests tallers a la vegada procedeixen a realitzar el flux de sortida de les peces.

Per a que l'anterior circuit de negoci no s'interrompi, cal evitar que en cap moment un taller es quedi sense estoc d'una peça i per tant deixi de servir un client. Per a evitar aquesta desagradable situació s'estableixen dos mecanismes bàsics de control, la consulta de l'estoc a petició dels usuaris i la localització i/o traçabilitat d'una peça. La diferència és que l'usuari central pot accedir a la consulta de qualsevol peça i taller, mentre que l'usuari taller només pot consultar l'estat de les peces que ha demanat a la central i el seu propi estoc.

Com a mecanisme de compensació en cas de no poder evitar una ruptura d'estoc, el taller té l'opció de demanar a la central que li proporcioni un taller alternatiu al client per tal de poder atendre la seva petició, la central rep la petició i autoritza el re adreçament segons la informació de l'estoc de la resta de tallers sobre la peça en qüestió.



- **Agents implicats**

El subsistema implica els mateixos usuaris ja identificats en el pas d'especificació i planificació del projecte.

1. **Operador de taller.** Usuaris dels diferents tallers de la xarxa.
2. **Operador de central.** Usuari de la central.

A continuació es fa una nova classificació de les funcionalitats del sistema segons l'àmbit de comunicació, aquesta novetat només té efectes a nivell de presentació del subsistema per tal d'entendre millor la seva operativitat. Dins de la nova classificació es descriuen funcionalitats de l'especificació i algunes noves relacionades amb la identificació de noves entitats.

- **Funcions i operacions de control del flux de comunicació amb l'exterior**

Aquestes operacions només les pot efectuar l'usuari central.

1. L'operador central tractarà en exclusiva amb els proveïdors per tant centralitzarà totes les peticions dels tallers.
2. S'ha creat una entitat anomenada comanda que agrupa les peces demanades per la xarxa de tallers en un interval de temps per tal que, la central les demani al proveïdor en cas de no tenir prou estoc. El motiu d'aquesta agrupació és l'estalvi de costos de transport i gestió que representa respecte a les peticions unitàries o aleatòries als proveïdors.
3. L'usuari central s'encarregarà de fer la recepció de les comandes, emetre l'albarà i gestionar la comanda, que inclou la seva distribució. La central també tractarà com una comanda la devolució de peces per part dels tallers i n'haurà de fer la gestió corresponent.

- **Funcions i operacions de control de flux de comunicació amb l'interior**

Operacions i funcionalitats que s'encarreguen del control i de la supervisió del flux de negoci de l'empresa i per tant tenen un àmbit interior. Els tallers només han de poder veure la part que ells controlen de la part del flux de les peces de recanvi, mentre que la central ha de veure la totalitat de tallers. De fet en el document d'especificació es proveeix a la central d'una centraleta d'avisos on pot veure en temps real l'estat de la circulació de les peces, així com de les peticions realitzades pels tallers.

a) Operador de la central

1. Consulta de l'estoc de tots els tallers
2. Consulta de la traçabilitat de cadascuna de les peces que circulen dins l'empresa
3. Rebre les peces retornades pels tallers
4. Consulta de la centraleta d'alertes

La classe controladora d'aquestes operacions està separada de la del taller precisament per l'abast de la informació que rep cada usuari. La centraleta d'alertes funciona a banda de la classe controladora de peces de la central i de la de comandes perquè rep informació dels dos àmbits i per tant ha d'estar a nivell de la gestió directa de l'usuari central.

b) Operador de taller

1. Consulta de l'estoc de peces propi del taller.
2. Consulta de la traçabilitat de les peces que ha demanat el taller o que té en estoc.
3. Efectuar la devolució de peces a la central.
4. Efectuar la sortida de peces.
5. Fer la petició de peces a la central per abastir l'estoc.

- **Funcions i operacions mixtes**

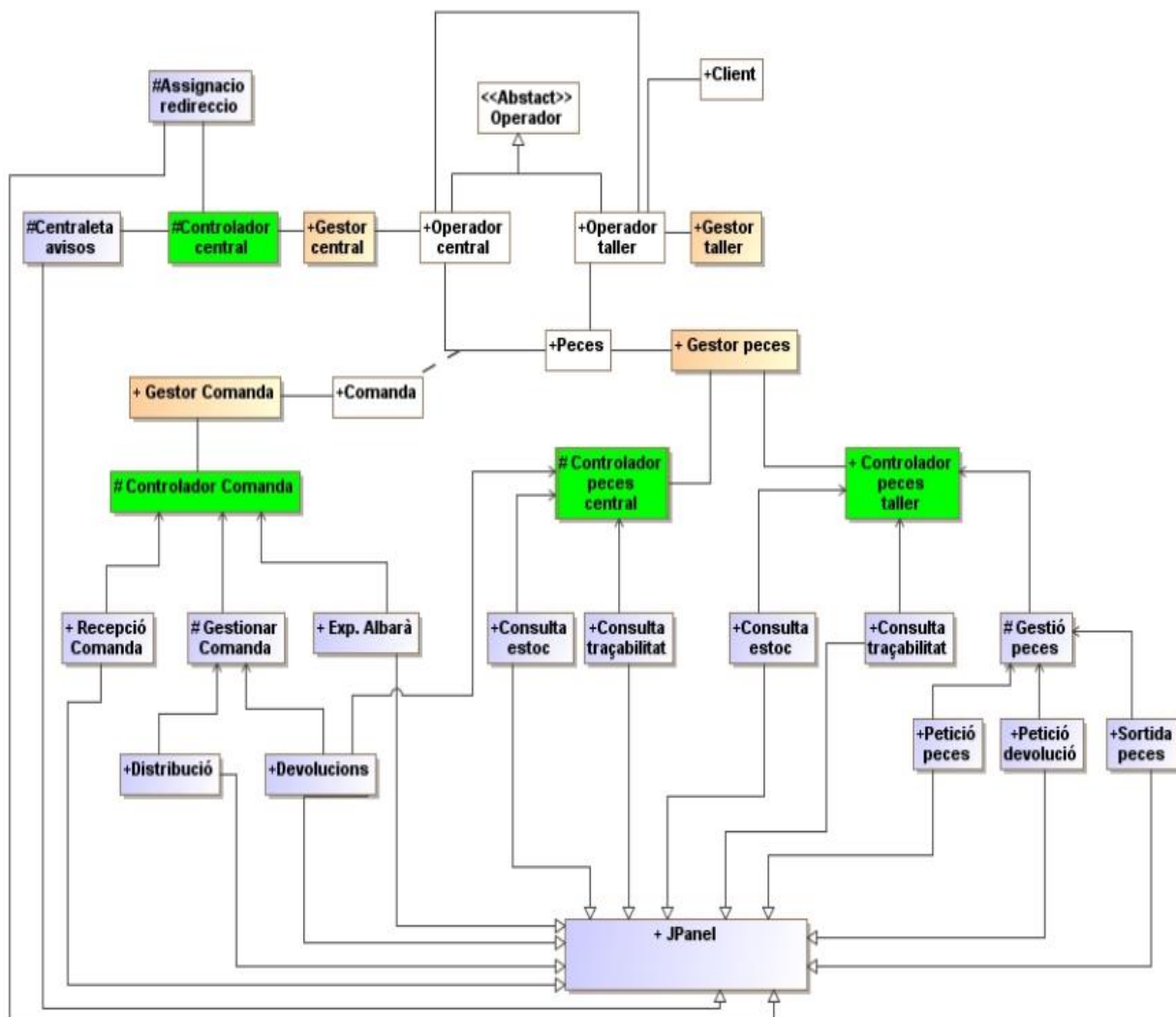
Es fa un anàlisi a banda d'una funcionalitat específica del subsistema que fa referència a la redirecció dels clients per part d'un taller en cas de ruptura d'estoc. La situació que es donaria i la seqüència de resposta seria la següent:

1. El taller informa mitjançant el procés corresponent, d'una fallada en l'estoc i demana que s'envii informació sobre el taller més pròxim on el client pot aconseguir la peça que desitja.
2. La central rep la petició i la informació i consulta la localització de la peça a tot el sistema.
3. En cas que hi hagi disponibilitat de la peça en la xarxa de tallers, es busca el centre més pròxim a la incidència i s'envia la informació al taller que ha fet la petició

S'ha optat per no posar aquesta funcionalitat dins del món de les operacions de comunicació interna donat que inclouen part de comunicació externa, la que s'estableix amb el client que malgrat que és una extensió del model original cal tenir en compte i per tant preveure'n la possibilitat.

## 4.2. Diagrama de classes i jerarquia UML

Com en la resta de subsistemes del projecte, s'usa com a patró el model vista-controlador, que separa la gestió del sistema i les dades de l'aplicació de usuari (UI) per tal de confeccionar el diagrama UML i la modelització del projecte.



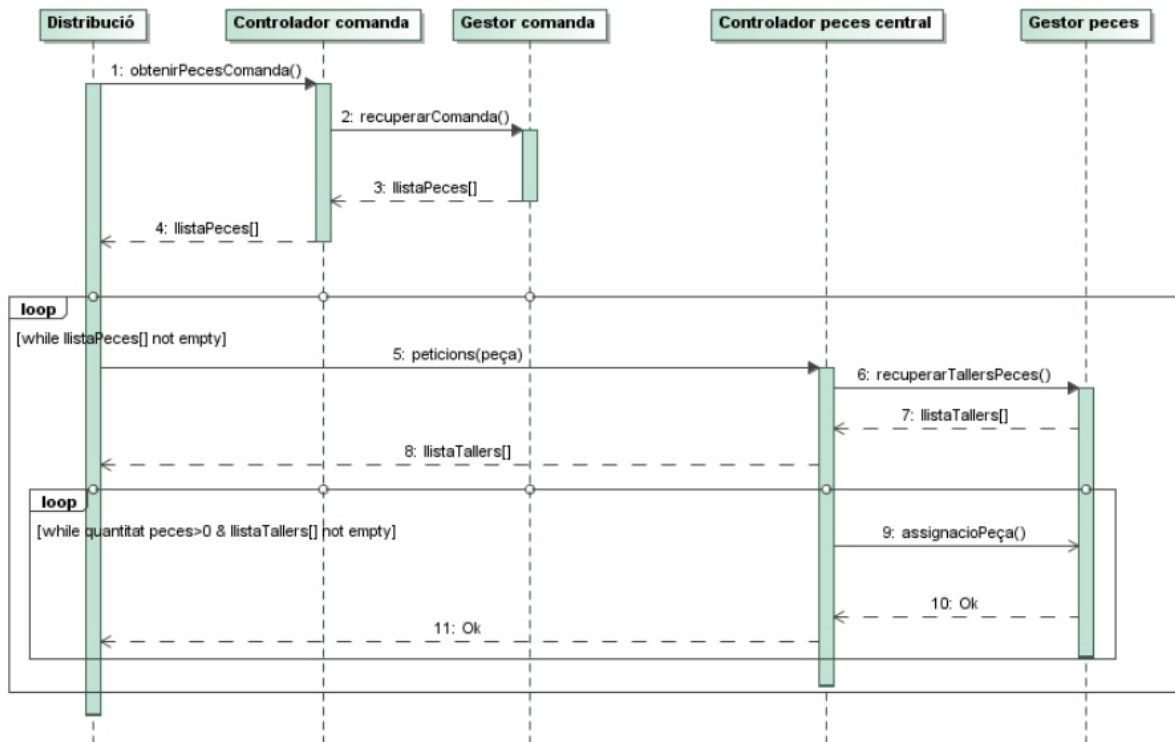
### 4.3. Diagrames d'interacció

A continuació es descriuen tres diagrames d'interacció que descriuen la dinàmica dels casos d'ús més representatius del subsistema.

#### 1. Distribució d'una comanda

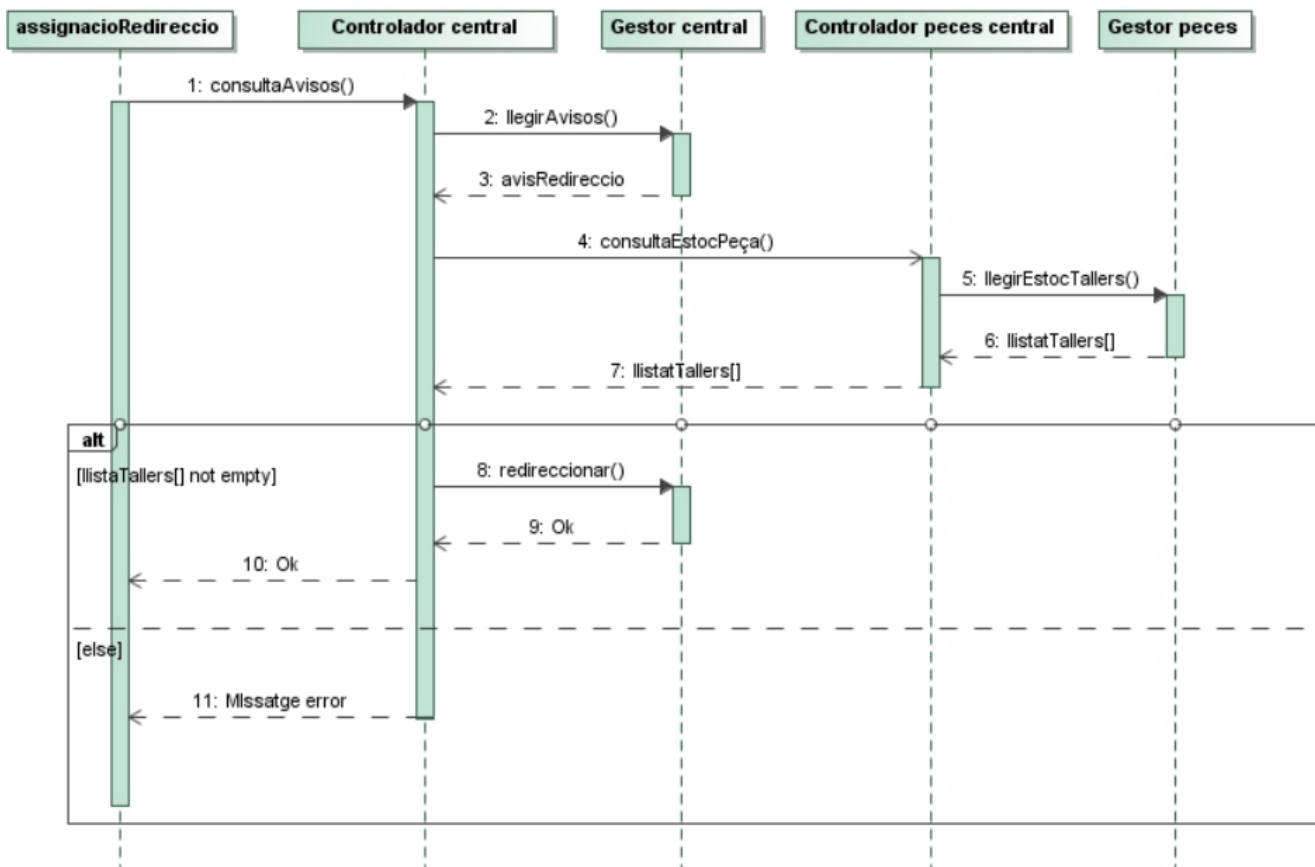
Aquesta funcionalitat consisteix en la mecànica usada pel sistema per tal de distribuir les peces que integren una comanda entre els diferents tallers que han efectuat peticions.

En aquest cas la pantalla de la interfície d'usuari Distribució es posa en contacte amb el controlador de comandes per obtenir les peces de comanda i aquesta recupera la comanda del gestor de comandes, el controlador de comandes llavors recupera les peces de la comanda. A continuació per cada lot de peces del mateix tipus el controlador de peces de la central consultarà l'estoc dels tallers i anirà assignant les peces fins que s'esgoti aquest tipus de peça entre els tallers segons els criteris marcats. Es repeteix el procés amb el següent lot de peces i grup de tallers, fins que s'esgota la llista de peces de la comanda.



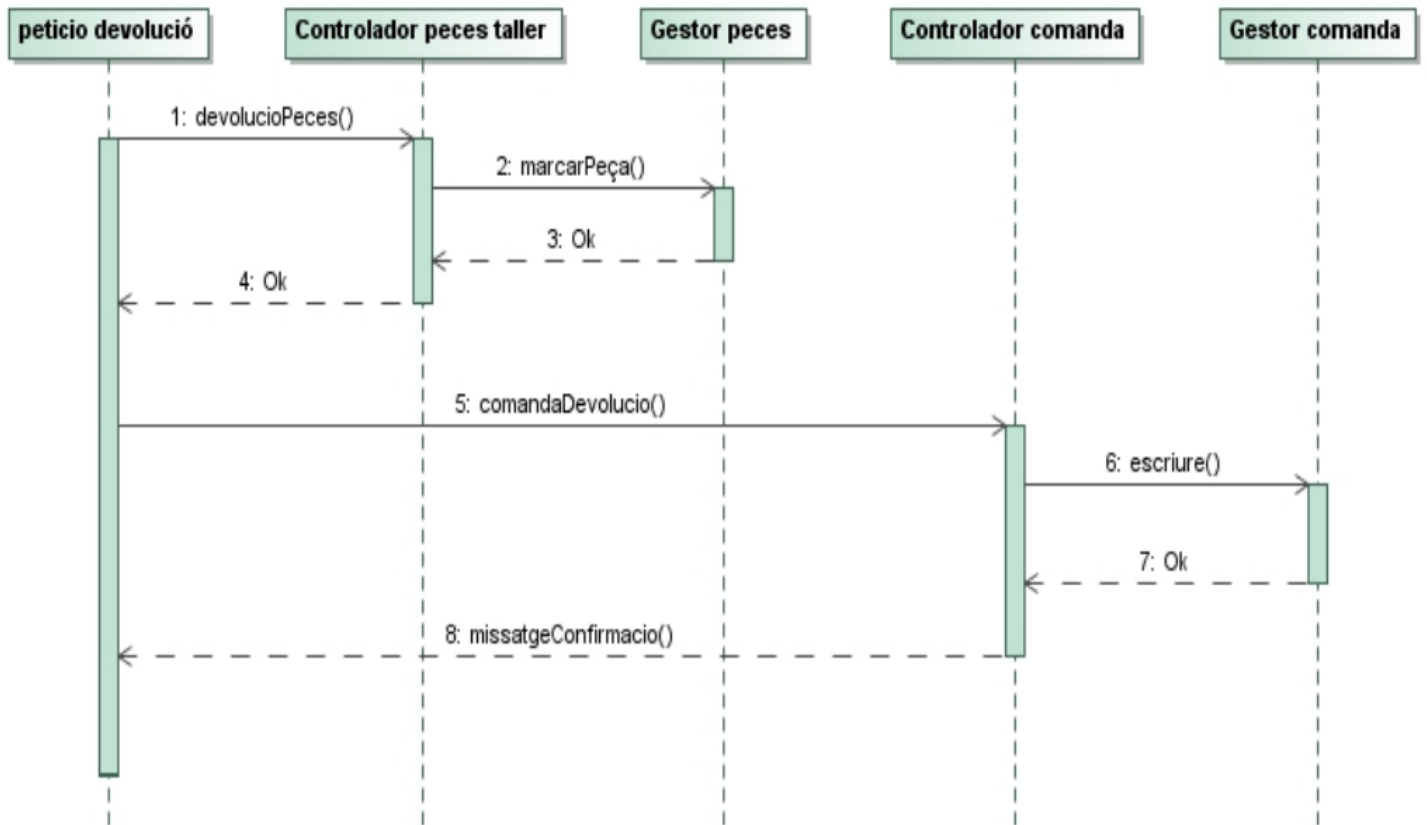
## 2. Redirecció d'un client a un altre taller

Aquesta funcionalitat descriu el mecanisme pel qual un taller fa una petició de derivació d'un client que ha d'atendre l'operador central per tal de trobar el taller amb estoc de la peça demanada més proper. La mecànica del cas començaria amb la interfície de l'usuari operador taller on pot efectuar una petició de derivació. Aquesta es posa en contacte amb el controlador central que recupera del gestor els avisos, a continuació si s'escau demana els estocs de la peça en qüestió a tots els tallers al controlador de peces central que a través del gestor proporciona al controlador central els tallers alternatius, aquest controlador escull el taller idoni i l'envia al gestor i la confirmació a l'usuari taller que fa la petició. Ara bé si cap taller té estoc suficient per atendre la derivació el controlador central envia a l'usuari un missatge d'error.



### 3. Devolució d'una peça per part d'un taller

En aquesta funcionalitat es descriu com un taller emet una devolució que recull la central i com la tracta com una comanda més. A través de la interfície de devolució un operador de taller demana la devolució d'una peça, el controlador de peces del taller s'encarrega de marcar la peça com a retornada en el gestor de peces. A continuació a través de la petició al controlador de comanda es transmet a la central la informació necessària per a preparar la comanda de devolució que es grava al gestor de comanda i finalment envia un missatge de confirmació a la interfície de devolució del taller.





#### 4.4. Canvis en els prototip d'interfície gràfica inicials

S'han plantejat canvis en les interfícies d'usuari que fan referència a la traçabilitat d'una peça, tant en el cas de l'usuari operador central com en el cas de l'usuari operador de taller.

Aquest canvi es deu al fet que en la fase de disseny es crea una referència a la comanda dins d'una peça, aquest fet fa que es pugui recuperar directament l'historial d'una peça només consultant la referència de la comanda on va arribar a la central. Després es rescata del gestor de comandes la comanda triada i ja es pot disposar de les dades de traçabilitat de la peça.

També es considera adequat proporcionar una caixa de text on poder triar la data d'extracció de les dades per tal que es pugui acotar l'abast temporal de la consulta de comandes.

D'altra banda s'ha de remarcar que entre les interfícies de l'operador de la central i l'operador de taller l'única diferència es troba en el fet que per al primer es dona l'opció d'escollir el taller per al que es vol fer la consulta de l'històric de moviments d'una peça.

Operador de la central:

**Operador central: Gestió magatzem - Consulta traçabilitat**

**Consulta traçabilitat**      Data In:       Data fi:

**Dades peça a consultar**

Referència       Motor       Grup       Descripció       3 estoc

Taller      

Ref. comanda	Data comanda	Data recepcio

Origen	Data Entrada	Dies en	Data sortida	Destinacio	Click /

Operador de taller:

**Operador taller: Gestió magatzem - Consulta estoc i traçabilitat**

**Consulta traçabilitat**      Data In:       Data fi:

**Dades peça a consultar**

Referència       Motor       Grup       Descripció       3 estoc

Ref. comanda	Data comanda	Data recepcio

Origen	Data Entrada	Dies en	Data sortida	Destinacio	Click /

## 5. Subsistema de llistats i estadístiques

### 5.1. Descripció general del subsistema

Aquest subsistema realitza la gestió de les tasques ja detallades a la fase d'especificació i planificació i, per tant, continua amb els mateixos objectius. Ho recordem, a continuació, per tal de fer-ho més clarificador. Es vol realitzar un control sobre la gestió d'estocs implementada amb la nova aplicació, per poder valorar-ne l'eficàcia, i avaluar i detectar punts a millorar així com possibles defectes. Per a implementar aquests mecanismes de control s'estableixen cinc informes. Tot cinc tenen en comú una característica d'ús, en tots els casos la informació que extregui el subsistema ha de venir referenciada a un interval concret de temps (una data d'inici i una data final).

Per altra banda, no tots els informes requereixen del mateix grau d'entrada d'informació per part de l'usuari, per exemple l'informe de vendes es pot generar directament un cop decidit el període d'extracció de les dades, mentre que la resta poden requerir entrar més dades com, el taller o el producte.

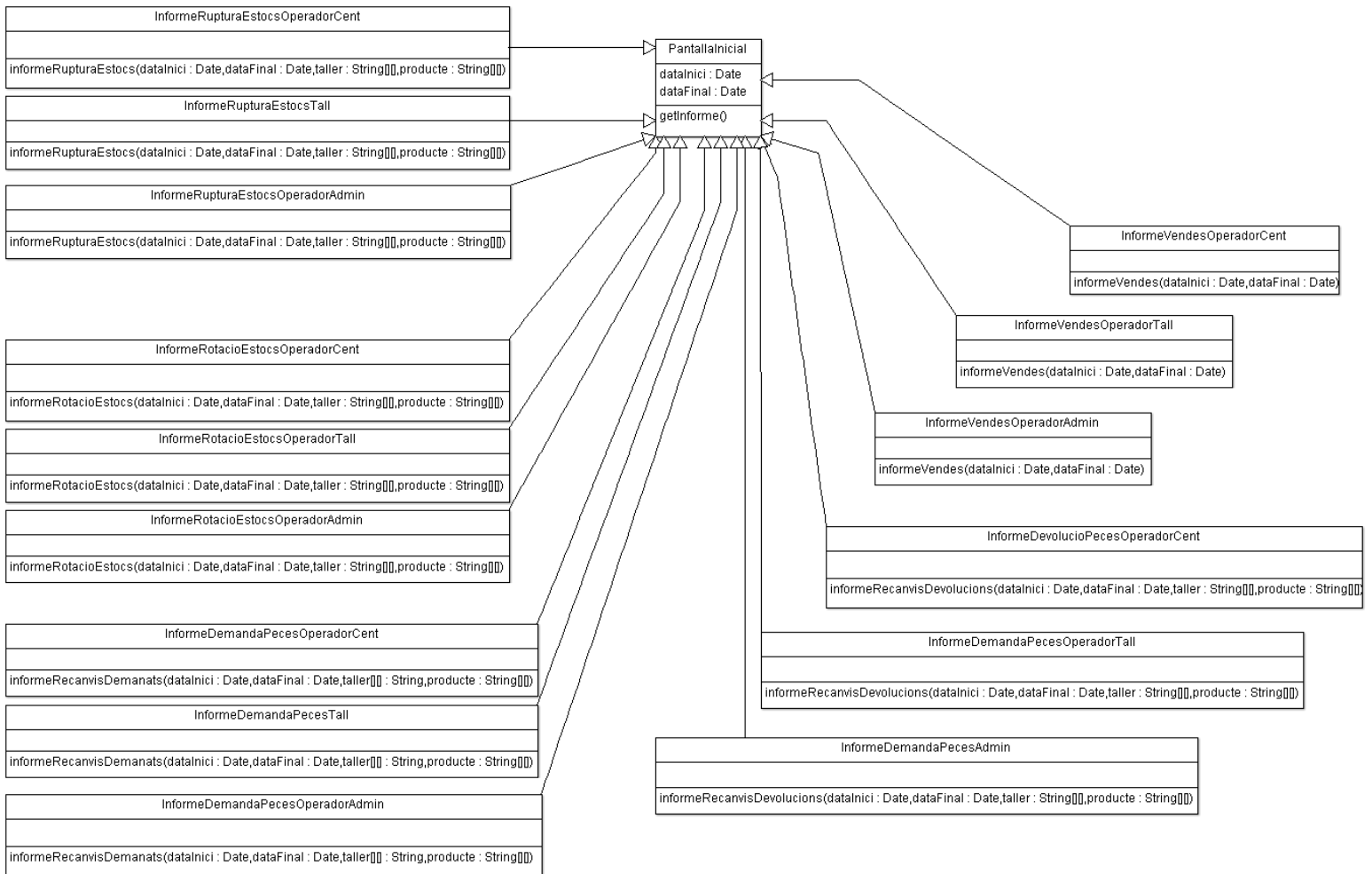
Llistem, a continuació, els diferents informes:

- 1) Informe de ruptura d'estocs
- 2) Informe de rotació d'estocs
- 3) Informe de recanvis més/menys demandats
- 4) Informe de recanvis amb més/menys percentatge de devolucions
- 5) Informe de vendes

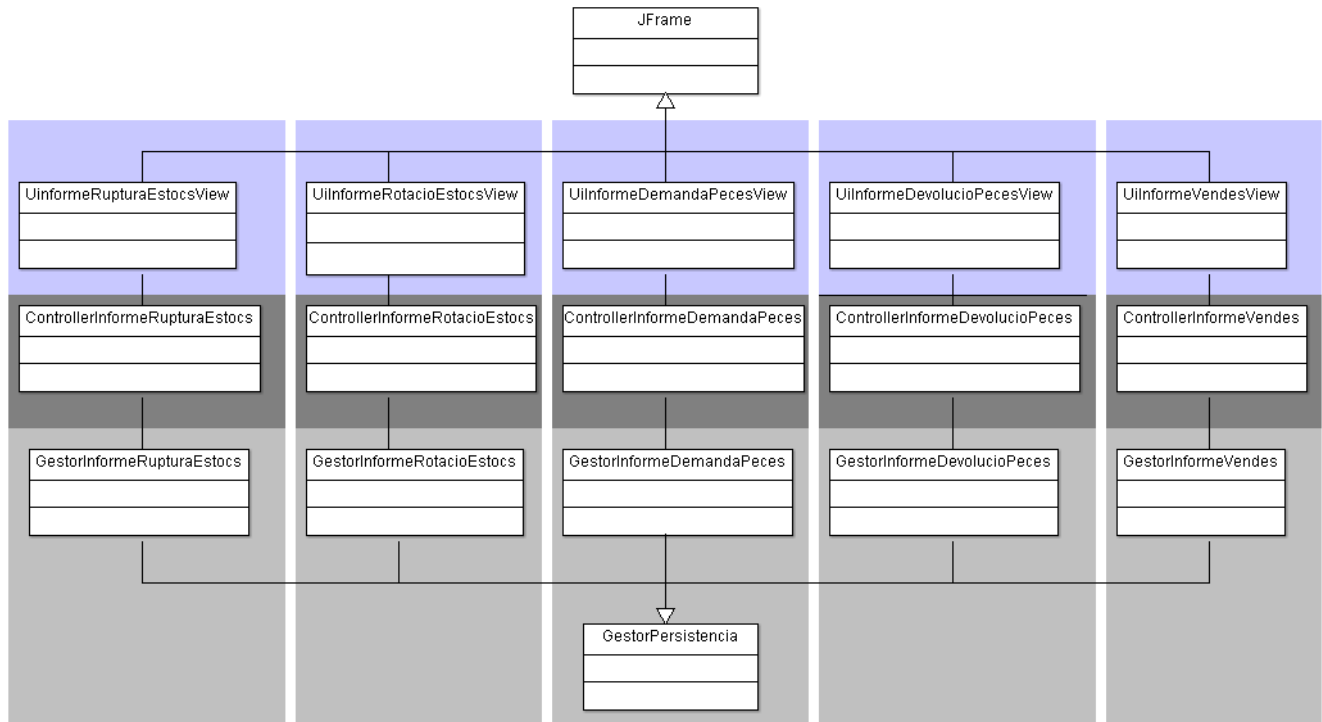
## 5.2. Diagrama de classes i jerarquia UML

Primerament, es representa els diagrames de classes i jerarquia del UML del nostre subsistema de llistat i estadístiques seguint, com sempre, el patró MVC.

## Diagrama de classes:



### Diagrama de jerarquies:

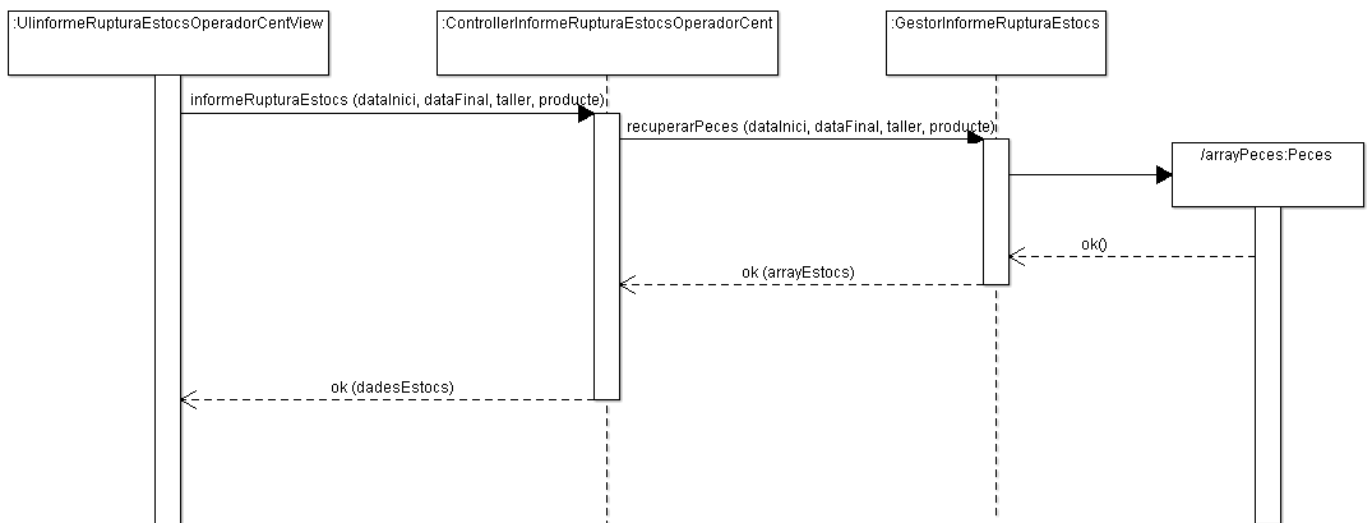


### 5.3. Diagrames d'interacció

A continuació i per tal de fer els diagrames de seqüència o interacció, s'han seleccionat els tres casos d'ús més representatius d'aquest subsistema:

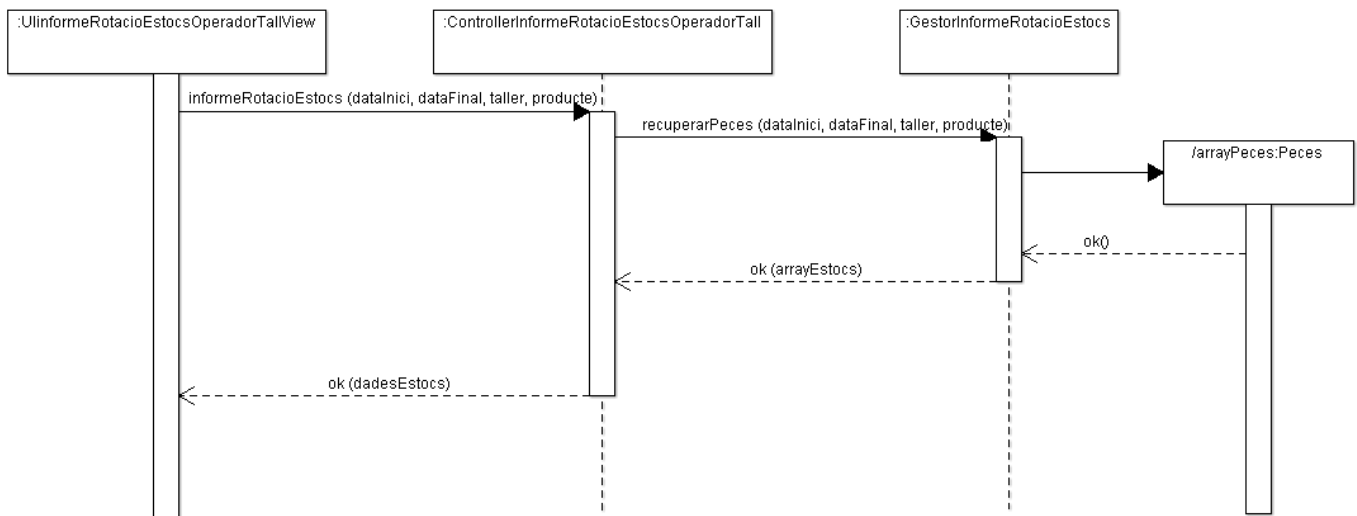
#### 1. Informe de ruptura d'estocs

En primer lloc, es representa la interacció de l'Operari de la Central amb el sistema per treure un informe del número de vedades que no s'ha pogut servir una peça degut a la manca d'estoc als taller i/o la central:



## 2. Informe de rotació d'estocs

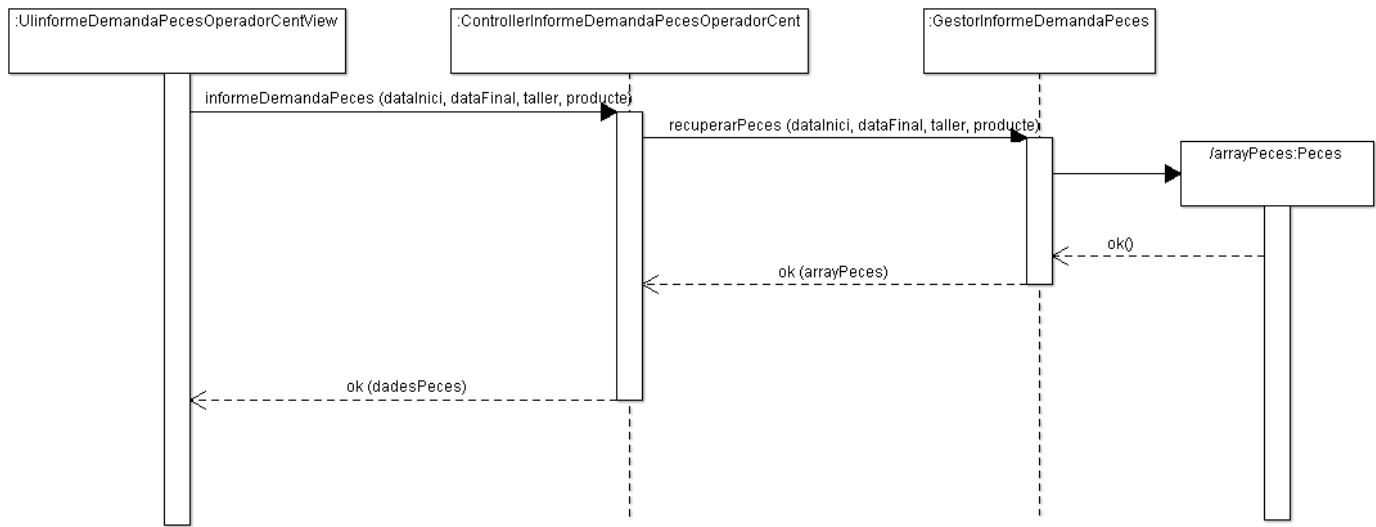
En segon lloc, aquesta interacció entre l'Operari de Taller i el sistema, ens permet, donat un període de temps que correspondrà a la data inici i final de l'informe la rotació d'estocs d'un tipus de peça a escollir:





### 3. Informe de demanda de peces

Finalment, en la darrera interacció, podem veure el número de peces venudes en quant a unitats de peça i els seus imports un cop l'Operari de la Central ho demana al nostre sistema:



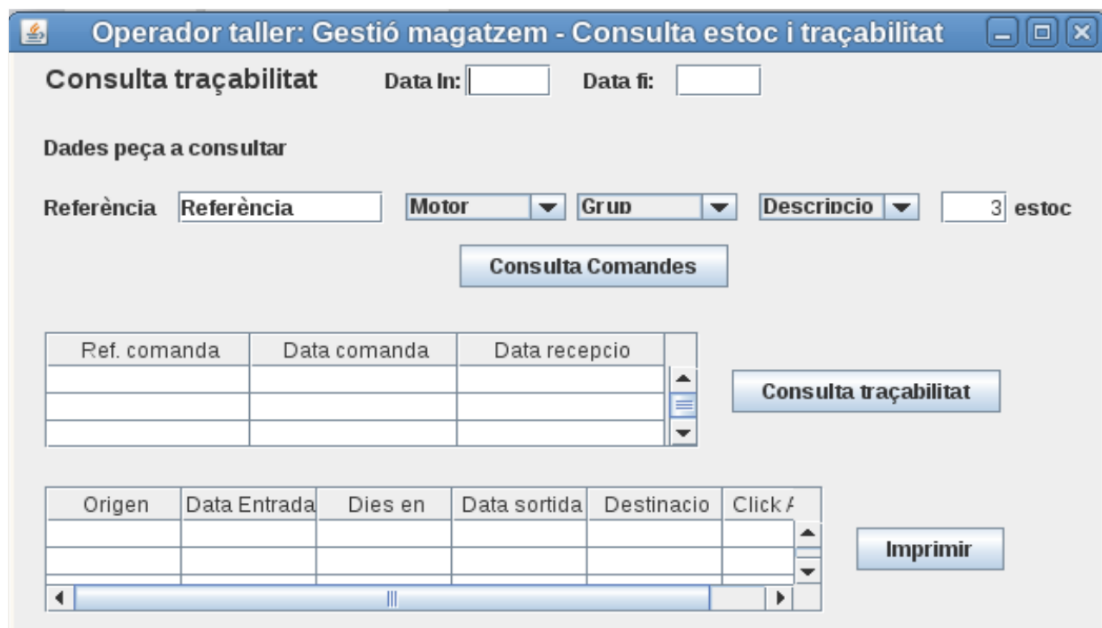
## 6. Subsistema de connexió

### 6.1 Descripció textual de les funcionalitats

En primer lloc, indicar que no hem realitzat cap modificació en les primeres funcionalitats que vam proposar a l'anterior fase, per tant a continuació farem una breu explicació del que ja es va explicar a mode de recordatori.

El subsistema de connexió s'executa quan l'usuari ha accedit a l'aplicació. Un cop hem fet el "login" el sistema ha de detectar quin rol tenim a l'aplicació i una vegada ja sap quin es, podem fer unes operacions o unes altres.

- 1) Autenticar-se a l'aplicació.
- 2) Un cop dintre, podem realitzar la modificació de la contrasenya o la modificació de l'idioma de l'aplicació.



- 3) Funcionalitats segons ROL.

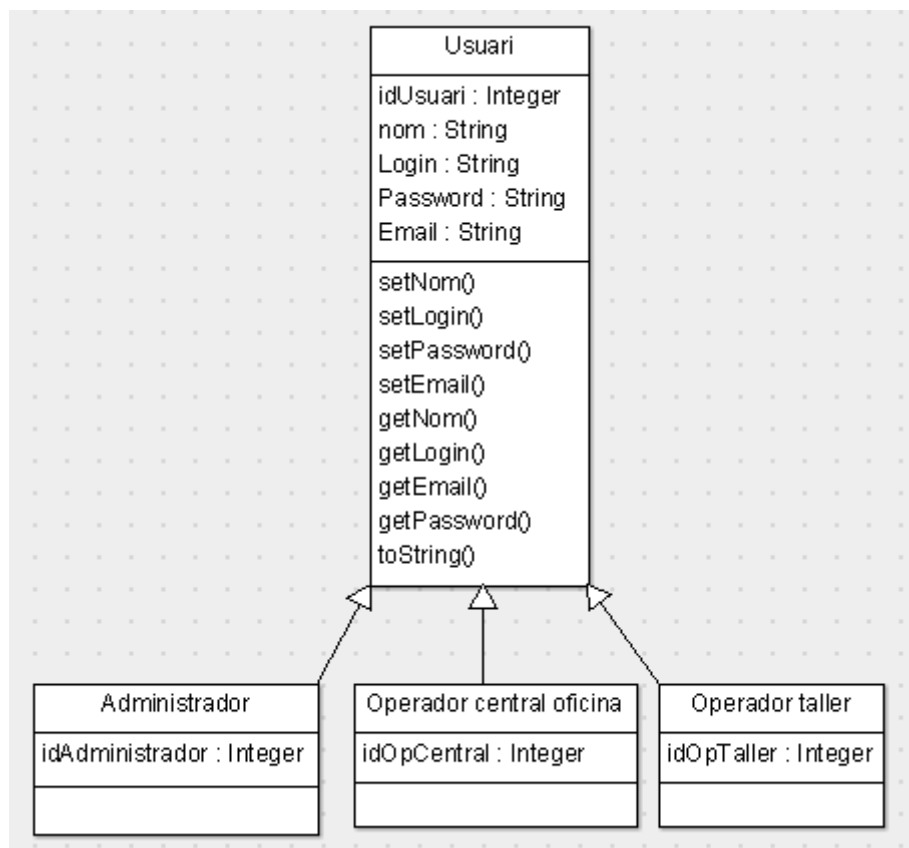
Recordem que els rols son els següents:

- Operador d'oficina central

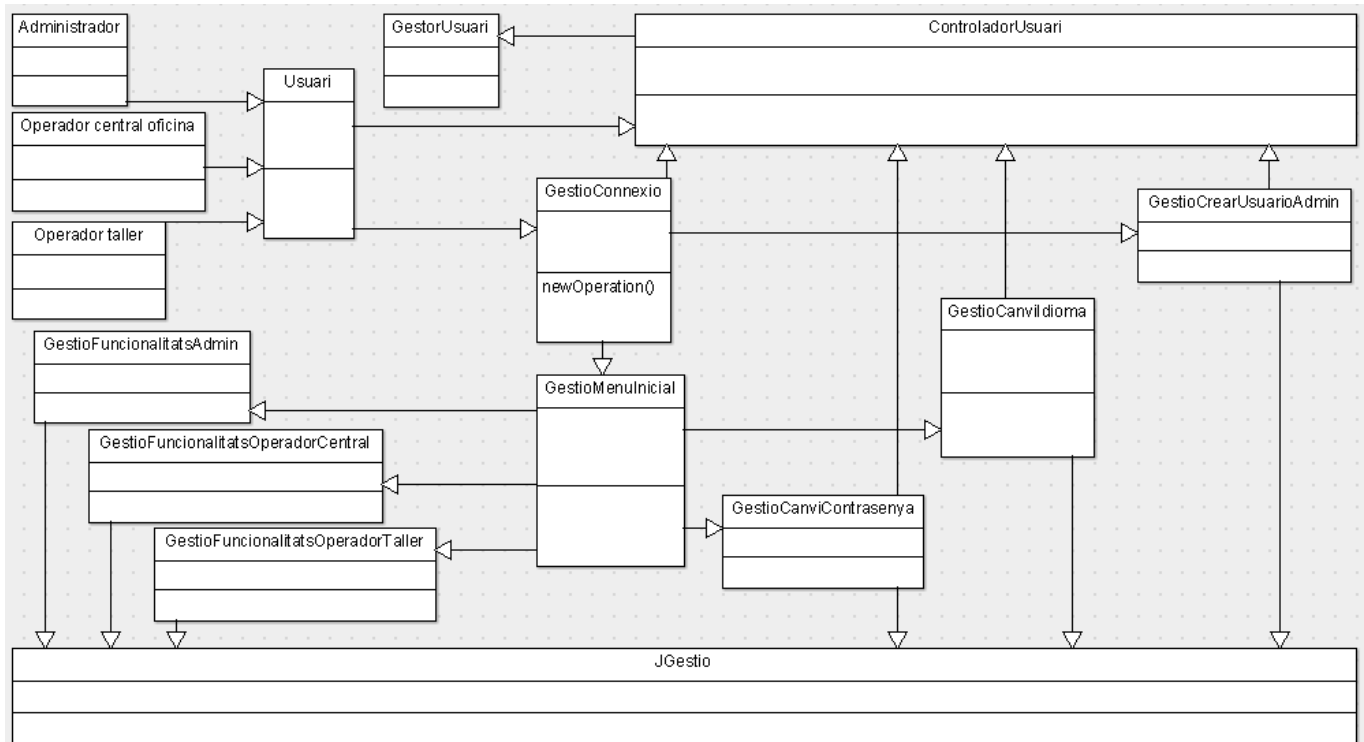
- Operador de taller
  - Administrador
- a) Les operacions que pot fer l'operador d'oficina central són:
- consultar la traçabilitat i localització de les peces de recanvi de qualsevol producte a tots els tallers
  - fer la recepció del material i expedir l'albarà pertinent
  - consultar totes les estadístiques
  - rebutjar el material rebut al magatzem central
- b) Pel que respecte a l'operador de taller, haurà de fer:
- consultar les peces al taller juntament amb la seva traçabilitat des del seu origen amb informació del proveïdor, dies al magatzem central, dies 'in itinere', data d'arribada al taller i links a tots els albarans
  - demanar una peça de recanvi a les oficines centrals
  - recepció de tot el material i expedició de l'albarà
  - consultar les estadístiques del taller
- c) L'administrador podrà fer les operacions de CRUD, és a dir, 'create', 'read', 'update' i 'delete' (crear, obtenir, actualitzar i esborrar)

## 6.2 Diagrama de classes i jerarquies UML

En primer lloc mostrarem el diagrama de classes únicament de l'usuari i els diferents rols.



I a continuació, tot el diagrama sencer de jerarquies UML del subsistema de connexió.



### 6.3 Diagrama de seqüència

Diagrama de seqüència per a realitzar login.

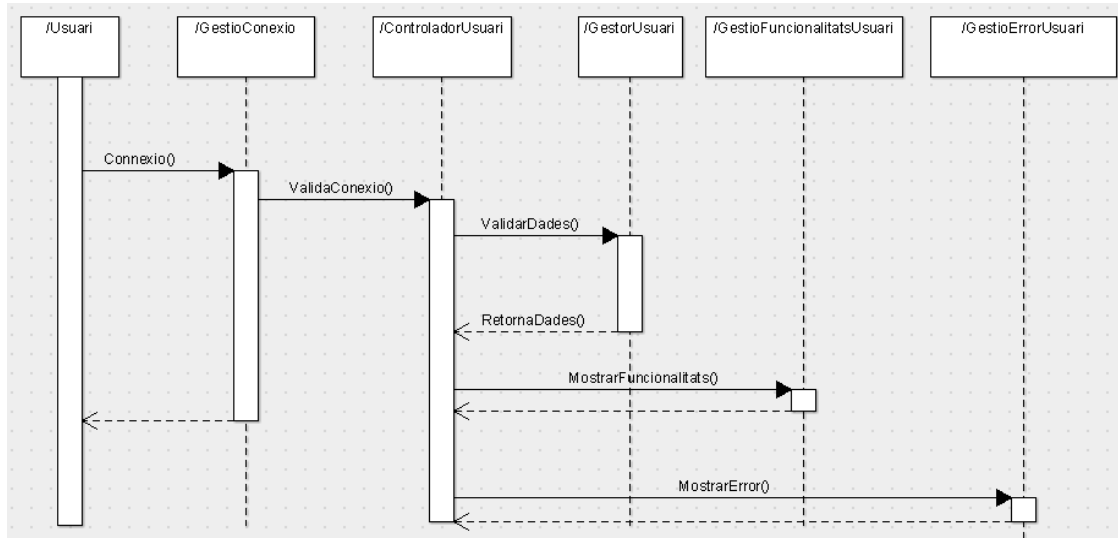


Diagrama de seqüència per a realitzar la gestió del canvi d'idioma.

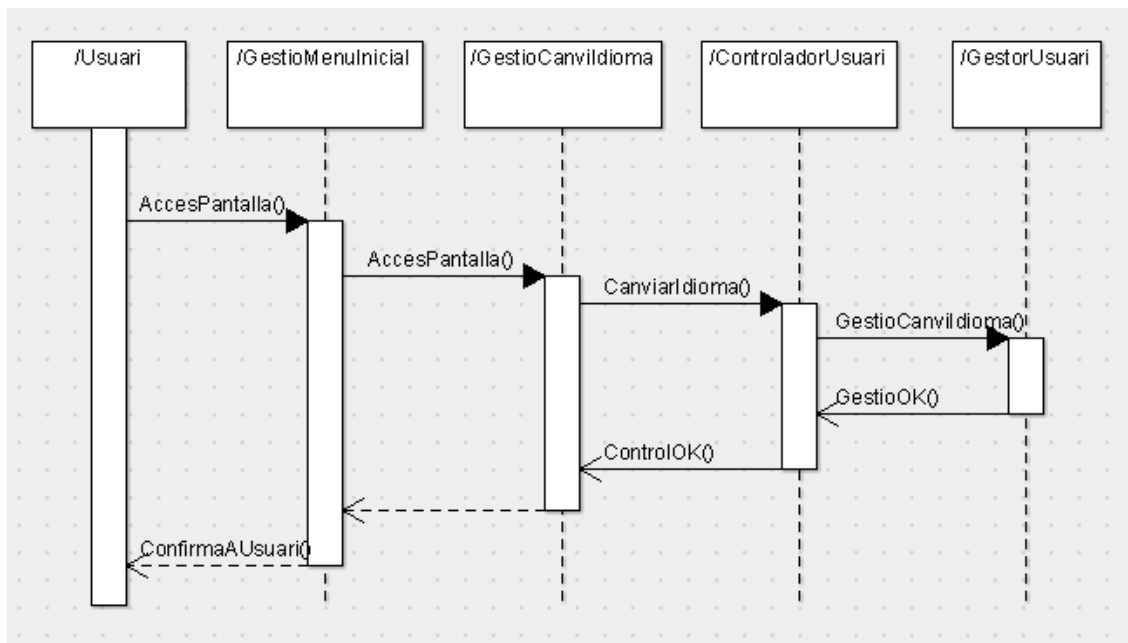
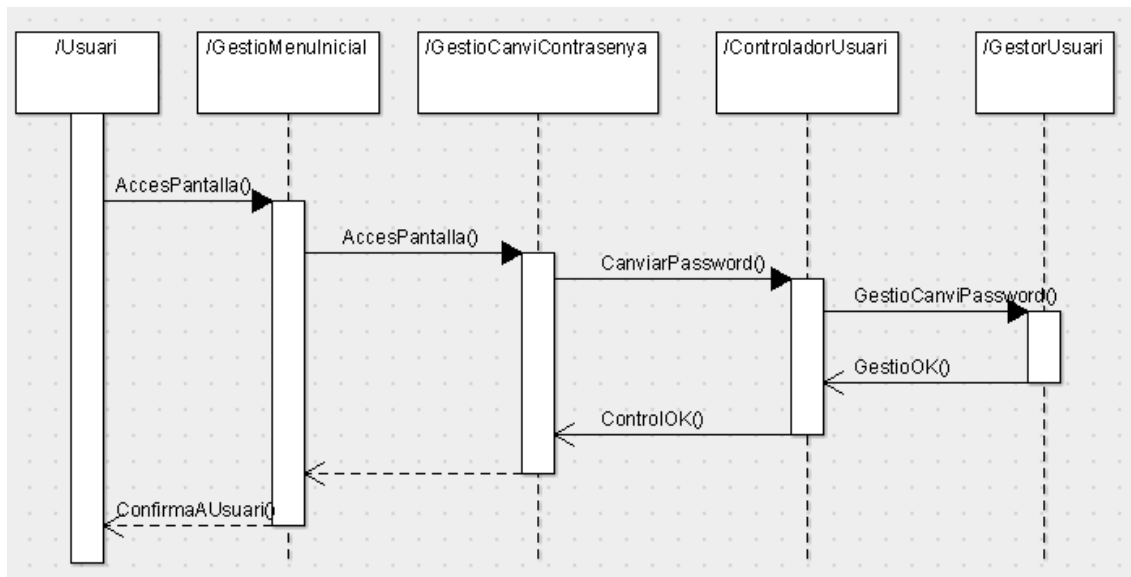


Diagrama de seqüència per al canvi de contrasenya.



## 6.4 Variacions respecte el prototipus inicial

Gràficament no hem introduït cap canvi, però si que hem decidit fer una millora en quant al xifrat de la contrasenya. Per motius de seguretat, es important tenir la contrasenya ben protegida i es per això que utilitzarem el següent algorisme per tal de realitzar un xifrat SHA-1.

El SHA-1 (algorisme de 'hash' segur), es una família 'hash' de xifrat publicades per l'Institut Nacional d'estàndards i tecnologia (NIST).

SHA-1, produeix una sortida de 160 bits (20 bytes) d'un missatge que pot tenir una mida màxima de  $2^{64}$  bits.

A continuació, posaré el codi Java que utilitzarem:

```
private static String prConvertirToHEX(byte[] data) throws
java.io.IOException
{
    /*
     * Funció per a convertir a Hexadecimal, que farem servir per
codificar la contrasenya al terminal.
     */
    StringBuffer sb = new StringBuffer();
    String hex=null;

    hex=Base64.encodeToString(data, 0, data.length, 0);

    sb.append(hex);

    return sb.toString();
}

public static String prPassToHash(String password)
{
    1. /** Funció que fem servir per a xifrar la contrasenya mitjançant SHA-
    1. */
    MessageDigest mdSha1 = null;
    try{
        mdSha1 = MessageDigest.getInstance("SHA-1");
    }
    catch (NoSuchAlgorithmException e1){
        e1.printStackTrace();
    }
    try{
        mdSha1.update(password.getBytes("ASCII"));
    }
    catch (UnsupportedEncodingException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    byte[] data = mdSha1.digest();
    try{
        SHAHash=prConvertirToHEX(data);
    }
    catch (IOException e){
        e.printStackTrace();
    }

    return SHAHash;
}
```



## 7. Reutilització

Analitzats els elements de la nostra proposta es relacionen els que seran objecte de reutilització a la construcció del projecte eTallerStock. S'ha d'indicar que hem intentat coordinar-nos per tal d'utilitzar les mateixes classes en tots els subsistemes intentant fer l'adaptació a complir les necessitats en lloc de crear models *ad hoc* particulars

### 7.1. Interfície gràfica

A l'aplicació s'utilitzaran classes i controls derivats de Java swing. Per aquesta manera, els components utilitzats seran JFrame, JDialog i JWindow com a classes pare de les pantalles d'aplicació. També es preveu l'ús de Swing Application Framework com a causa de major complexitat

### 7.2. Accés a disc/dades

Com s'ha descrit en aquest mateix document, s'utilitzarà la classe GestorPersistencia per tal de poder accedir a les dades.

Tot i això, a mesura que es vagi avançant a la implementació serà necessari tenir en compte que el seu funcionament s'haurà d'ajustar en relació a la definició final

### 7.3. Llibreries de l'aplicació

Tenim previst utilitzar la llibreria i18n, que s'ha ofert a l'espai BSCW, per a la gestió multi-idioma de la presentació tant a textos en pantalles com a missatges de l'aplicació.

### 7.4. Llibreries de Java

A més d'utilitzar les llibreries indicades per fer ús de swing (javax.swing), la gestió del nostre projecte també es realitzarà mitjançant l'ús de les llibreries d'ús comú com serien 'Date' per a dates, 'Hour' per a hores, etc.

El nostre propòsit seria utilitzar al màxim el potencial de les llibreries estàndard de Java.

## 8. Relació de classes

En aquest punt es presenten les classes detectades durant el treball de disseny del projecte. Només es detallen les classes que impacten directament en el cicle de negoci de cada subsistema, és a dir les classes que en cada subsistema suporten la funcionalitat que el caracteritza i per al que ha estat dissenyat.

Classes	Subsistemes
Administrador	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> <li>• Control Flux</li> <li>• Connexió</li> </ul>
Comanda	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
ControladorCentral	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
ControladorComanda	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
ControladorPecesCentral	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
ControladorPecesTaller	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
ControladorUsuari	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>

Credencials	<ul style="list-style-type: none"> <li>• Connexió</li> </ul> Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
Distribucio	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
GestorInformeRotacióEstocs	Subsistemes: <ul style="list-style-type: none"> <li>• Llistats i Estadístiques</li> </ul>
GestorInformeRupturaEstocs	Subsistemes: <ul style="list-style-type: none"> <li>• Llistats i Estadístiques</li> </ul>
GestorInformeDevolucióPeces	Subsistemes: <ul style="list-style-type: none"> <li>• Llistats i Estadístiques</li> </ul>
GestorInformeVendes	Subsistemes: <ul style="list-style-type: none"> <li>• Llistats i Estadístiques</li> </ul>
Grup	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
Menu	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
OperariCentral	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> <li>• Control Flux</li> <li>• Connexió</li> </ul>
OperariTaller	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>

	<ul style="list-style-type: none"> <li>• Control Flux</li> <li>• Connexió</li> </ul>
PeçaTaller	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
Producte/Peça	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> <li>• Control Flux</li> </ul>
Proveidor	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
Recepcio	Subsistemes: <ul style="list-style-type: none"> <li>• Control de Flux</li> </ul>
Subgrup	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
Taller	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> <li>• Control Flux</li> </ul>
Usuari	Subsistemes: <ul style="list-style-type: none"> <li>• Manteniment</li> <li>• Control Flux</li> <li>• Connexió</li> </ul>

## 9. Disseny de la persistència

### 9.1 Descripció general del model de persistència

Donat el possible creixement de l'empresa 'TotCatCars' hem optat per donar persistència a algunes entitats que en principi podrien haver segut representades per altres estructures de dades perquè tenen poques ocurrencies

En concret Província 1->\*Població

I també Rol 1->\* Usuaris ,

D'altra banda **hem dissenyat dos models ER dels que podríem dir que la versió 1 hereta de la versió 2 que es mes general i simplificada.**

Hem preferit donar les dues opcions perquè de fet generant la BD de la versió 1, es podria implementar el model 2 , si calgués, en les següents fases ja que com hem dit no son models disjunts sinó que versió 1 inclou versió 2 (únicament hi hauria que fer uns petits canvis en les FK )

La diferencia entre els dos models resideix en el disseny de taules independents -o no- per la gestió del magatzem de la oficina central y també el tractament independent dels diferents tipus de usuari creant taules o no.

**Per fer una elecció o una altra del model definitiu se tindran en compte els següents factors:**

Necessitat de independitzar a nivell de taula les dades del magatzem central per: numero d' accés concurrents de tallers i central a la mateixa taula, existència de dades de preu sensibles o d'un altre tipus en les peces de la central.

Nous comportaments o relacions diferenciades de cada tipus de usuari con respecte als sistemes del projecte o altres sistemes de l'empresa. Necessitat de nous atributs per cada tipus de usuari .

D'una altra banda s'ha de tenir en compte el nivell de complexitat afegit en el codi font per el fet de tenir diferents taules per magatzem central i magatzem dels tallers.

Des de el nostre punt de vista a priori sense mes dades pensem que el model versió 2 podria ser una primera opció valida, però preferim tenir mes dades del client per prendre una decisió.

## 9.2 Descripció del diagrama ER versió 1

En aquesta versió hem creat taules independents per cada tipus de usuari i també per la central i el seu magatzem, pressuposant el escenari mes restrictiu en quant accés concurrent, confidencialitat de les dades i noves atribucions dels diferents usuaris.

Les taules **Peticio** i **Redireccio** son les que implementen la comunicació entre taller i central , i en concret sobre la taula Peticio es on la central fa la seva distribució de peces.

La traçabilitat de totes les peces es garanteix amb el camp **codiTrac**, que al formar part de la clau primària de les taules implicades ens permet tenir traçabilitat per diferents peces de una mateixa referencia.

El us de **codiTrac** suposa que hi ha un subsistema comercial de comandes fora del abast d'aquest subsistema que alimenta la taula **Peça** de manera que quan es fa un comanda al proveïdor de una Peça existent o no, ja se li assigna un **codiTrac** el qual es utilitzat després una vegada es fa la recepció de la peça, la devolució o qualsevol tractament de la mateixa. En el nostre cas i donat que no tenim aquesta connexió amb el subsistema comercial **ho implementarem mitjançant el alta de producte en el manteniment per mig del usuari administrador.**

Les devolucions i les sortides comparteixen taules **SortidaDevolucio** i **SortidaDevolucioPeça**, perquè en realitat suposen una sortida de material del magatzem, es diferencien per el camp Tipus.

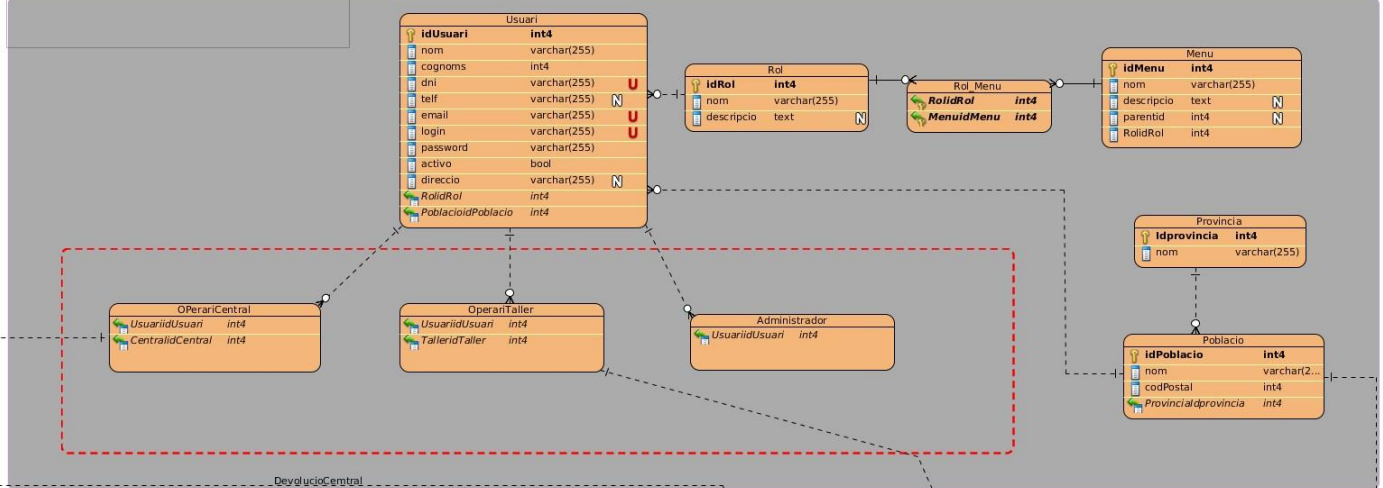
Tant la taula **SortidaDevolucioPeça** com **PeçaRecepcio** tenen el camp **snapEstoc** en aquest camp s'emmagatzema el estoc actual en el moment de realitzar la operació, d'aquesta manera es garanteix la possibilitat de consultar després la dada de **rotació d'estocs**.

Les taules **Peçataller** i **PeçaCentral** donant persistència als magatzems dels diferents tallers i del magatzem central.

I la resta de taules son pràcticament auto-explicatives.

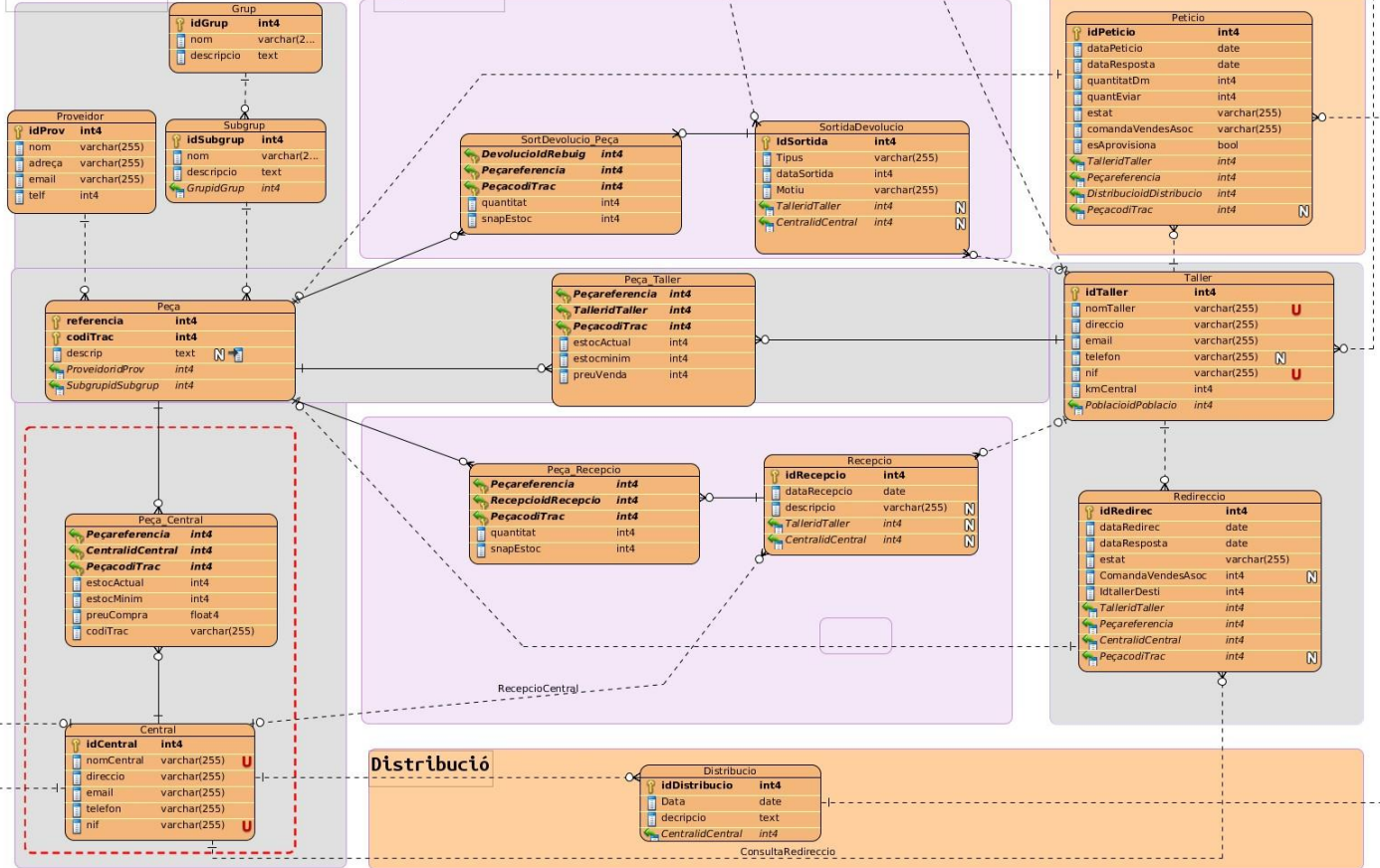
Visual Paradigm Community Edition 100 for commercial use

## Gestió Usuaris i Connexió



## Gestió estocs

## Expedicions



VERSÍO 1 DEL ER

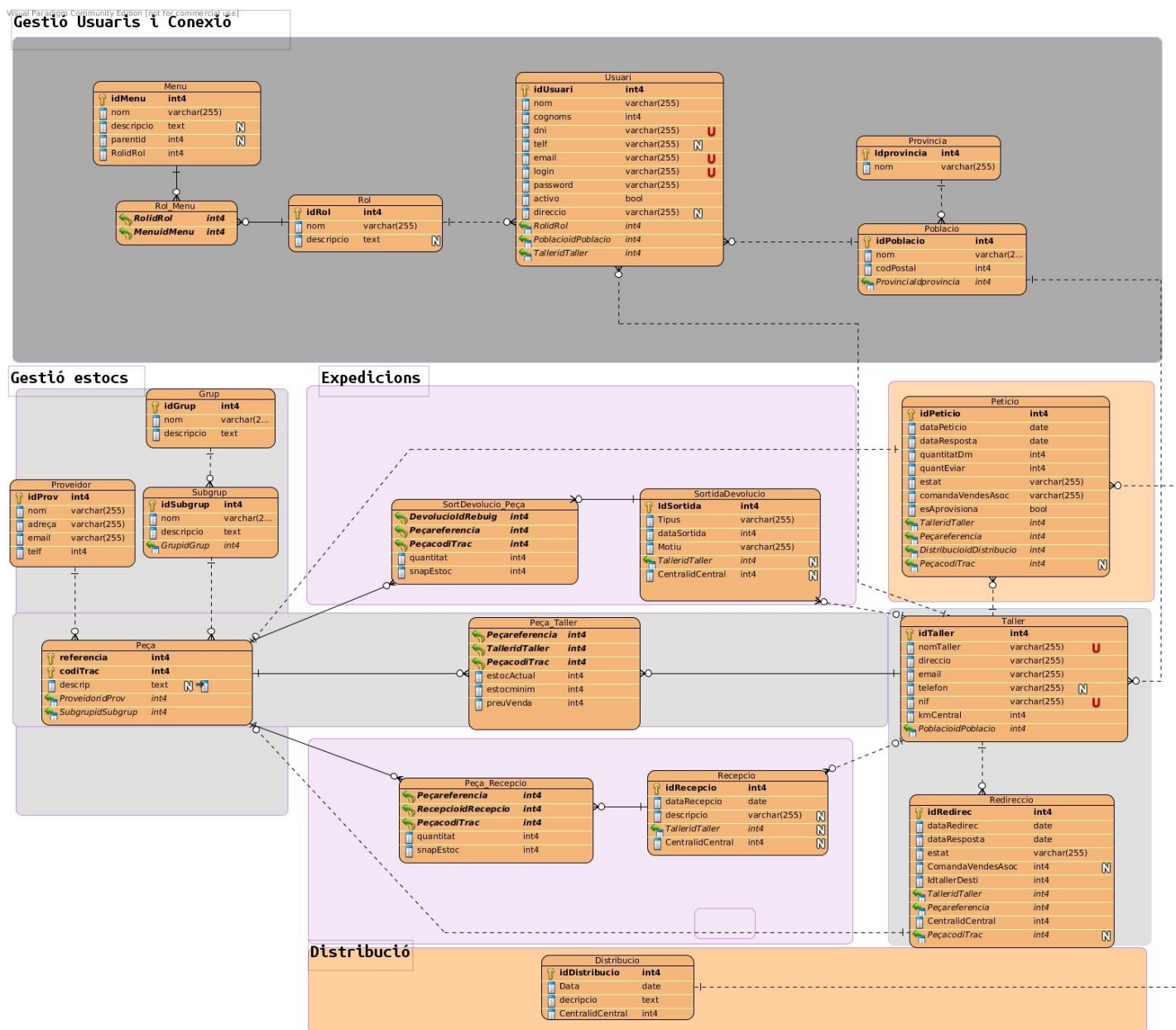


### 9.3 Descripció del diagrama ER versió 2 simplificada

En aquest model la oficina central i el seu magatzem son tractats a tots els efectes com un taller mes, només hi haurà un procés diferent que pot fer la central que es la distribució que utilitza una taula diferent dels tallers que es la taula distribució que es només que la agrupació de les diferents peticions i els seus enviaments corresponents.

D'altra banda les taules especificues per els usuaris Operador Central, Operador Taller i Administrador, desapareixen ja que les diferents usuaris s'implementen mitjançant la relació amb el seu rol que seran com a mínim d'aquest tres tipus. Operador central, Operador taller i Administrador però que està obert a la creació de qualsevol altre rol.

La resta es compartit en la versió 1.



VERSÍO 2 SIMPLIFICADA ER

## 9.4 Scripts SQL (DDL amb DML) sobre la versió 1 del ER

### DDL

#### *Creates*

```
CREATE TABLE Provincia (
  Idprovincia SERIAL NOT NULL,
  nom          varchar(255) NOT NULL,
  PRIMARY KEY (Idprovincia));
CREATE TABLE Poblacio (
  idPoblacio    SERIAL NOT NULL,
  nom           varchar(255) NOT NULL,
  codPostal     int4 NOT NULL,
  ProvinciaIdprovincia int4 NOT NULL,
  PRIMARY KEY (idPoblacio));
CREATE TABLE Rol_Menu (
  RolIdRol      int4 NOT NULL,
  MenuIdMenu    int4 NOT NULL,
  PRIMARY KEY (RolIdRol,
  MenuIdMenu));
CREATE TABLE Rol (
  idRol         SERIAL NOT NULL,
  nom           varchar(255) NOT NULL,
  descripcio    text,
  PRIMARY KEY (idRol));
CREATE TABLE Menu (
  idMenu        SERIAL NOT NULL,
  nom           varchar(255) NOT NULL,
  descripcio    text,
  parentid      int4,
  RolIdRol      int4 NOT NULL,
  PRIMARY KEY (idMenu));
CREATE TABLE Distribucio (
  idDistribucio SERIAL NOT NULL,
  Data          date NOT NULL,
  decripcio     text NOT NULL,
  CentralidCentral int4 NOT NULL,
  PRIMARY KEY (idDistribucio));
CREATE TABLE SortDevolucio_Peça (
  DevolucioIdRebuig int4 NOT NULL,
  Peçareferencia    int4 NOT NULL,
  PeçaacodiTrac     int4 NOT NULL,
  quantitat         int4 NOT NULL,
  snapEstoc         int4 NOT NULL,
  PRIMARY KEY (DevolucioIdRebuig,
  Peçareferencia,
  PeçaacodiTrac));
CREATE TABLE Peça_Recepcio (
  Peçareferencia    int4 NOT NULL,
  RecepcioIdRecepcio int4 NOT NULL,
  PeçaacodiTrac     int4 NOT NULL,
  quantitat         int4 NOT NULL,
  snapEstoc         int4 NOT NULL,
  PRIMARY KEY (Peçareferencia,
```

```

    RecepcioidRecepcio,
    PeçacodiTrac));
CREATE TABLE SortidaDevolucio (
    IdSortida          SERIAL NOT NULL,
    Tipus              varchar(255) NOT NULL,
    dataSortida        int4 NOT NULL,
    Motiu              varchar(255) NOT NULL,
    TalleridTaller     int4,
    CentralidCentral   int4,
    PRIMARY KEY (IdSortida));
CREATE TABLE Recepcio (
    idRecepcio         SERIAL NOT NULL,
    dataRecepcio        date NOT NULL,
    descripcio          varchar(255),
    TalleridTaller     int4,
    CentralidCentral   int4,
    PRIMARY KEY (idRecepcio));
CREATE TABLE Peça_Central (
    Peçareferencia     int4 NOT NULL,
    CentralidCentral   int4 NOT NULL,
    PeçacodiTrac       int4 NOT NULL,
    estocActual        int4 NOT NULL,
    estocMinim         int4 NOT NULL,
    preuCompra         float4 NOT NULL,
    codiTrac           varchar(255) NOT NULL,
    PRIMARY KEY (Peçareferencia,
    CentralidCentral,
    PeçacodiTrac));
CREATE TABLE Central (
    idCentral          SERIAL NOT NULL,
    nomCentral         varchar(255) NOT NULL UNIQUE,
    direccio           varchar(255) NOT NULL,
    email              varchar(255) NOT NULL,
    telefon            varchar(255) NOT NULL,
    nif                varchar(255) NOT NULL UNIQUE,
    PRIMARY KEY (idCentral));
CREATE TABLE Redireccio (
    idRedirec          SERIAL NOT NULL,
    dataRedirec        date NOT NULL,
    dataResposta       date NOT NULL,
    estat              varchar(255) NOT NULL,
    ComandaVendesAsoc int4,
    IdtallerDesti      int4 NOT NULL,
    TalleridTaller     int4 NOT NULL,
    Peçareferencia     int4 NOT NULL,
    CentralidCentral   int4 NOT NULL,
    PeçacodiTrac       int4,
    PRIMARY KEY (idRedirec));
CREATE TABLE Peticio (
    idPeticio          SERIAL NOT NULL,
    dataPeticio        date NOT NULL,
    dataResposta       date NOT NULL,
    quantitatDm        int4 NOT NULL,
    quantEviar         int4 NOT NULL,
    estat              varchar(255) NOT NULL,
    comandaVendesAsoc varchar(255) NOT NULL,
    esAprovisiona      bool NOT NULL,
    TalleridTaller     int4 NOT NULL,

```

```

    Peçareferencia          int4 NOT NULL,
    DistribucioidDistribucio int4 NOT NULL,
    PeçacodiTrac            int4,
    PRIMARY KEY (idPeticio));
CREATE TABLE Peça_Taller (
    Peçareferencia int4 NOT NULL,
    TalleridTaller int4 NOT NULL,
    PeçacodiTrac   int4 NOT NULL,
    estocActual    int4 NOT NULL,
    estocminim     int4 NOT NULL,
    preuVenda      int4 NOT NULL,
    PRIMARY KEY (Peçareferencia,
    TalleridTaller,
    PeçacodiTrac));
CREATE TABLE Proveidor (
    idProv SERIAL NOT NULL,
    nom     varchar(255) NOT NULL,
    adreça  varchar(255) NOT NULL,
    email   varchar(255) NOT NULL,
    telf    int4,
    PRIMARY KEY (idProv));
CREATE TABLE Grup (
    idGrup SERIAL NOT NULL,
    nom     varchar(255) NOT NULL,
    descripcio text,
    PRIMARY KEY (idGrup));
CREATE TABLE Subgrup (
    idSubgrup SERIAL NOT NULL,
    nom        varchar(255) NOT NULL,
    descripcio text,
    GrupidGrup int4 NOT NULL,
    PRIMARY KEY (idSubgrup));
CREATE TABLE Peça (
    referencia          int4 NOT NULL,
    codiTrac            int4 NOT NULL,
    descripcio          text,
    ProveidoridProv    int4 NOT NULL,
    SubgrupidSubgrup   int4 NOT NULL,
    PRIMARY KEY (referencia,
    codiTrac));
CREATE TABLE Taller (
    idTaller SERIAL NOT NULL,
    nomTaller varchar(255) NOT NULL UNIQUE,
    direccio  varchar(255) NOT NULL,
    email     varchar(255) NOT NULL,
    telefon   varchar(255),
    nif       varchar(255) NOT NULL UNIQUE,
    kmCentral int4 NOT NULL,
    PoblacioidPoblacio int4 NOT NULL,
    PRIMARY KEY (idTaller));
CREATE TABLE Administrador (
    UsuariidUsuari int4 NOT NULL);
CREATE TABLE OperariTaller (
    UsuariidUsuari int4 NOT NULL,
    TalleridTaller int4 NOT NULL);
CREATE TABLE OperariCentral (
    UsuariidUsuari int4 NOT NULL,
    CentralidCentral int4 NOT NULL);

```

```
CREATE TABLE Usuari (
    idUsuari SERIAL NOT NULL,
    nom varchar(255) NOT NULL,
    cognoms int4 NOT NULL,
    dni varchar(255) NOT NULL UNIQUE,
    telf varchar(255),
    email varchar(255) NOT NULL UNIQUE,
    login varchar(255) NOT NULL UNIQUE,
    password varchar(255) NOT NULL,
    activo bool NOT NULL,
    direccio varchar(255),
    RolidRol int4 NOT NULL,
    PoblacioidPoblacio int4 NOT NULL,
    PRIMARY KEY (idUsuari));
CREATE INDEX Peça_descrip
    ON Peça (descrip);
ALTER TABLE Poblacio ADD CONSTRAINT FKPoblacio263547 FOREIGN KEY
    (ProvinciaIdprovincia) REFERENCES Provincia (Idprovincia);
ALTER TABLE Taller ADD CONSTRAINT FKTaller55004 FOREIGN KEY
    (PoblacioidPoblacio) REFERENCES Poblacio (idPoblacio);
ALTER TABLE Usuari ADD CONSTRAINT FKUsuari565520 FOREIGN KEY
    (PoblacioidPoblacio) REFERENCES Poblacio (idPoblacio);
ALTER TABLE Rol_Menu ADD CONSTRAINT FKRol_Menu389658 FOREIGN KEY
    (RolidRol) REFERENCES Rol (idRol);
ALTER TABLE Rol_Menu ADD CONSTRAINT FKRol_Menu719436 FOREIGN KEY
    (MenuidMenu) REFERENCES Menu (idMenu);
ALTER TABLE Peticio ADD CONSTRAINT FKPeticio382777 FOREIGN KEY
    (DistribucioidDistribucio) REFERENCES Distribucio (idDistribucio);
ALTER TABLE OperariCentral ADD CONSTRAINT FKOperariCen439429
    FOREIGN KEY (UsuariidUsuari) REFERENCES Usuari (idUsuari);
ALTER TABLE OperariTaller ADD CONSTRAINT FKOperariTal814070 FOREIGN
    KEY (UsuariidUsuari) REFERENCES Usuari (idUsuari);
ALTER TABLE Administrador ADD CONSTRAINT FKAdministra512494 FOREIGN
    KEY (UsuariidUsuari) REFERENCES Usuari (idUsuari);
ALTER TABLE Peça_Taller ADD CONSTRAINT FKPeça_Talle256090 FOREIGN
    KEY (Peçareferencia, PeçacodiTrac) REFERENCES Peça (referencia,
    codiTrac);
ALTER TABLE Peça_Taller ADD CONSTRAINT FKPeça_Talle977252 FOREIGN
    KEY (TalleridTaller) REFERENCES Taller (idTaller);
ALTER TABLE Peça ADD CONSTRAINT FKPeça701925 FOREIGN KEY
    (ProveidoridProv) REFERENCES Proveidor (idProv);
ALTER TABLE Peça ADD CONSTRAINT FKPeça702489 FOREIGN KEY
    (SubgrupidSubgrup) REFERENCES Subgrup (idSubgrup);
ALTER TABLE Subgrup ADD CONSTRAINT FKSubgrup157315 FOREIGN KEY
    (GrupidGrup) REFERENCES Grup (idGrup);
ALTER TABLE Peticio ADD CONSTRAINT FKPeticio976175 FOREIGN KEY
    (TalleridTaller) REFERENCES Taller (idTaller);
ALTER TABLE Peticio ADD CONSTRAINT FKPeticio237928 FOREIGN KEY
    (Peçareferencia, PeçacodiTrac) REFERENCES Peça (referencia,
    codiTrac);
ALTER TABLE Redireccio ADD CONSTRAINT FKRedireccio739089 FOREIGN
    KEY (TalleridTaller) REFERENCES Taller (idTaller);
ALTER TABLE Redireccio ADD CONSTRAINT FKRedireccio999157 FOREIGN
    KEY (Peçareferencia, PeçacodiTrac) REFERENCES Peça (referencia,
    codiTrac);
ALTER TABLE Peça_Central ADD CONSTRAINT FKPeça_Centr500021 FOREIGN
    KEY (Peçareferencia, PeçacodiTrac) REFERENCES Peça (referencia,
    codiTrac);
```

```
ALTER TABLE Peça_Central ADD CONSTRAINT FKPeça_Centr550384 FOREIGN
KEY (CentralidCentral) REFERENCES Central (idCentral);
ALTER TABLE Redireccio ADD CONSTRAINT ConsultaRedireccio FOREIGN
KEY (CentralidCentral) REFERENCES Central (idCentral);
ALTER TABLE Recepcio ADD CONSTRAINT FKRecepcio339387 FOREIGN KEY
(TalleridTaller) REFERENCES Taller (idTaller);
ALTER TABLE Peça_Recepcio ADD CONSTRAINT FKPeça_Recep260098 FOREIGN
KEY (Peçareferencia, PeçacodiTrac) REFERENCES Peça (referencia,
codiTrac);
ALTER TABLE Peça_Recepcio ADD CONSTRAINT FKPeça_Recep247290 FOREIGN
KEY (RecepcioidRecepcio) REFERENCES Recepcio (idRecepcio);
ALTER TABLE SortidaDevolucio ADD CONSTRAINT FKSortidaDev47973
FOREIGN KEY (TalleridTaller) REFERENCES Taller (idTaller);
ALTER TABLE SortDevolucio_Peça ADD CONSTRAINT FKSortDevolu322642
FOREIGN KEY (DevolucioIdRebuig) REFERENCES SortidaDevolucio
(IdSortida);
ALTER TABLE SortDevolucio_Peça ADD CONSTRAINT FKSortDevolu219682
FOREIGN KEY (Peçareferencia, PeçacodiTrac) REFERENCES Peça
(referencia, codiTrac);
ALTER TABLE Distribucio ADD CONSTRAINT FKDistribucio196673 FOREIGN
KEY (CentralidCentral) REFERENCES Central (idCentral);
ALTER TABLE Recepcio ADD CONSTRAINT RecepcioCentral FOREIGN KEY
(CentralidCentral) REFERENCES Central (idCentral);
ALTER TABLE SortidaDevolucio ADD CONSTRAINT DevolucioCemtral
FOREIGN KEY (CentralidCentral) REFERENCES Central (idCentral);
ALTER TABLE Usuari ADD CONSTRAINT FKUsuari390832 FOREIGN KEY
(RolidRol) REFERENCES Rol (idRol);
ALTER TABLE OperariCentral ADD CONSTRAINT FKOperariCen80281 FOREIGN
KEY (CentralidCentral) REFERENCES Central (idCentral);
ALTER TABLE OperariTaller ADD CONSTRAINT FKOperariTal243533 FOREIGN
KEY (TalleridTaller) REFERENCES Taller (idTaller);
```

### Drop

```
ALTER TABLE Poblacio DROP CONSTRAINT FKPoblacio263547;
ALTER TABLE Taller DROP CONSTRAINT FKTaller55004;
ALTER TABLE Usuari DROP CONSTRAINT FKUsuari565520;
ALTER TABLE Rol_Menu DROP CONSTRAINT FKRol_Menu389658;
ALTER TABLE Rol_Menu DROP CONSTRAINT FKRol_Menu719436;
ALTER TABLE Peticio DROP CONSTRAINT FKPeticio382777;
ALTER TABLE OperariCentral DROP CONSTRAINT FKOperariCen439429;
ALTER TABLE OperariTaller DROP CONSTRAINT FKOperariTal814070;
ALTER TABLE Administrador DROP CONSTRAINT FKAdministra512494;
ALTER TABLE Peça_Taller DROP CONSTRAINT FKPeça_Talle256090;
ALTER TABLE Peça_Taller DROP CONSTRAINT FKPeça_Talle977252;
ALTER TABLE Peça DROP CONSTRAINT FKPeça701925;
ALTER TABLE Peça DROP CONSTRAINT FKPeça702489;
ALTER TABLE Subgrup DROP CONSTRAINT FKSubgrup157315;
ALTER TABLE Peticio DROP CONSTRAINT FKPeticio976175;
ALTER TABLE Peticio DROP CONSTRAINT FKPeticio237928;
ALTER TABLE Redireccio DROP CONSTRAINT FKRedireccio739089;
ALTER TABLE Redireccio DROP CONSTRAINT FKRedireccio999157;
ALTER TABLE Peça_Central DROP CONSTRAINT FKPeça_Centr500021;
ALTER TABLE Peça_Central DROP CONSTRAINT FKPeça_Centr550384;
ALTER TABLE Redireccio DROP CONSTRAINT ConsultaRedireccio;
ALTER TABLE Recepcio DROP CONSTRAINT FKRecepcio339387;
ALTER TABLE Peça_Recepcio DROP CONSTRAINT FKPeça_Recep260098;
ALTER TABLE Peça_Recepcio DROP CONSTRAINT FKPeça_Recep247290;
```



```

ALTER TABLE SortidaDevolucio DROP CONSTRAINT FKSortidaDev47973;
ALTER TABLE SortDevolucio_Peça DROP CONSTRAINT FKSortDevolu322642;
ALTER TABLE SortDevolucio_Peça DROP CONSTRAINT FKSortDevolu219682;
ALTER TABLE Distribucio DROP CONSTRAINT FKDistribucil96673;
ALTER TABLE Recepcio DROP CONSTRAINT RecepcioCentral;
ALTER TABLE SortidaDevolucio DROP CONSTRAINT DevolucioCentral;
ALTER TABLE Usuari DROP CONSTRAINT FKUsuari390832;
ALTER TABLE OperariCentral DROP CONSTRAINT FKOperariCen80281;
ALTER TABLE OperariTaller DROP CONSTRAINT FKOperariTal243533;
DROP TABLE IF EXISTS Provincia CASCADE;
DROP TABLE IF EXISTS Poblacio CASCADE;
DROP TABLE IF EXISTS Rol_Menu CASCADE;
DROP TABLE IF EXISTS Rol CASCADE;
DROP TABLE IF EXISTS Menu CASCADE;
DROP TABLE IF EXISTS Distribucio CASCADE;
DROP TABLE IF EXISTS SortDevolucio_Peça CASCADE;
DROP TABLE IF EXISTS Peça_Recepcio CASCADE;
DROP TABLE IF EXISTS SortidaDevolucio CASCADE;
DROP TABLE IF EXISTS Recepcio CASCADE;
DROP TABLE IF EXISTS Peça_Central CASCADE;
DROP TABLE IF EXISTS Central CASCADE;
DROP TABLE IF EXISTS Redireccio CASCADE;
DROP TABLE IF EXISTS Peticio CASCADE;
DROP TABLE IF EXISTS Peça_Taller CASCADE;
DROP TABLE IF EXISTS Proveidor CASCADE;
DROP TABLE IF EXISTS Grup CASCADE;
DROP TABLE IF EXISTS Subgrup CASCADE;
DROP TABLE IF EXISTS Peça CASCADE;
DROP TABLE IF EXISTS Taller CASCADE;
DROP TABLE IF EXISTS Administrador CASCADE;
DROP TABLE IF EXISTS OperariTaller CASCADE;
DROP TABLE IF EXISTS OperariCentral CASCADE;
DROP TABLE IF EXISTS Usuari CASCADE;

```

## DML

### Selects

```

SELECT Idprovincia, nom
  FROM Provincia;
SELECT idPoblacio, nom, codPostal, ProvinciaIdprovincia
  FROM Poblacio;
SELECT RolidRol, MenuidMenu
  FROM Rol_Menu;
SELECT idRol, nom, descripcio
  FROM Rol;
SELECT idMenu, nom, descripcio, parentid, RolidRol
  FROM Menu;
SELECT idDistribucio, Data, decricao, CentralidCentral
  FROM Distribucio;
SELECT DevolucioIdRebuig, Peçareferencia, PeçacodiTrac, quantitat,
snapEstoc
  FROM SortDevolucio_Peça;
SELECT Peçareferencia, RecepcioIdRecepcio, PeçacodiTrac, quantitat,
snapEstoc
  FROM Peça_Recepcio;
SELECT IdSortida, Tipus, dataSortida, Motiu, TalleridTaller,
CentralidCentral
  FROM SortidaDevolucio;
SELECT idRecepcio, dataRecepcio, descripcio, TalleridTaller,
CentralidCentral
  FROM Recepcio;
SELECT Peçareferencia, CentralidCentral, PeçacodiTrac, estocActual,
estocMinim, preuCompra, codiTrac
  FROM Peça_Central;
SELECT idCentral, nomCentral, direccio, email, telefon, nif
  FROM Central;
SELECT idRedirec, dataRedirec, dataResposta, estat,
ComandaVendesAsoc, IdtallerDesti, TalleridTaller, Peçareferencia,
CentralidCentral, PeçacodiTrac
  FROM Redireccio;
SELECT idPeticio, dataPeticio, dataResposta, quantitatDm,
quantEviar, estat, comandaVendesAsoc, esAprovisiona,
TalleridTaller, Peçareferencia, DistribucioidDistribucio,
PeçacodiTrac
  FROM Peticio;
SELECT Peçareferencia, TalleridTaller, PeçacodiTrac, estocActual,
estocminim, preuVenda
  FROM Peça_Taller;
SELECT idProv, nom, adreça, email, telf
  FROM Proveidor;
SELECT idGrup, nom, descripcio
  FROM Grup;
SELECT idSubgrup, nom, descripcio, GrupidGrup
  FROM Subgrup;
SELECT referencia, codiTrac, descrip, ProveidoridProv,
SubgrupidSubgrup
  FROM Peça;
SELECT idTaller, nomTaller, direccio, email, telefon, nif,
kmCentral, PoblacioidPoblacio
  FROM Taller;

```



```
SELECT UsuariidUsuari
  FROM Administrador;
SELECT UsuariidUsuari, TalleridTaller
  FROM OperariTaller;
SELECT UsuariidUsuari, CentralidCentral
  FROM OperariCentral;
SELECT idUsuari, nom, cognoms, dni, telf, email, login, password,
activo, direccio, RolidRol, PoblacioidPoblacio
  FROM Usuari;
```

### *Inserts*

```
INSERT INTO Provincia
  (Idprovincia,
   nom)
VALUES
  (?,
   ?);
INSERT INTO Poblacio
  (idPoblacio,
   nom,
   codPostal,
   ProvinciaIdprovincia)
VALUES
  (?,
   ?,
   ?,
   ?);
INSERT INTO Rol_Menu
  (RolidRol,
   MenuidMenu)
VALUES
  (?,
   ?);
INSERT INTO Rol
  (idRol,
   nom,
   descripcio)
VALUES
  (?,
   ?,
   ?);
INSERT INTO Menu
  (idMenu,
   nom,
   descripcio,
   parentid,
   RolidRol)
VALUES
  (?,
   ?,
   ?,
   ?,
   ?);
```

```
INSERT INTO Distribucio
(idDistribucio,
Data,
decripcio,
CentralidCentral)
VALUES
(?,
?,
?,
?);
INSERT INTO SortDevolucio_Peça
(DevolucioIdRebuig,
Peçareferencia,
PeçacodiTrac,
quantitat,
snapEstoc)
VALUES
(?,
?,
?,
?,
?);
INSERT INTO Peça_Recepcio
(Peçareferencia,
RecepcioIdRecepcio,
PeçacodiTrac,
quantitat,
snapEstoc)
VALUES
(?,
?,
?,
?,
?);
INSERT INTO SortidaDevolucio
(IdSortida,
Tipus,
dataSortida,
Motiu,
TalleridTaller,
CentralidCentral)
VALUES
(?,
?,
?,
?,
?,
?);
INSERT INTO Recepcio
(idRecepcio,
dataRecepcio,
descripcio,
TalleridTaller,
CentralidCentral)
VALUES
(?,
?,
?,
```

```

?,
?);
INSERT INTO Peça_Central
(Peçareferencia,
CentralidCentral,
PeçacodiTrac,
estocActual,
estocMinim,
preuCompria,
codiTrac)
VALUES
(?,
?,
?,
?,
?,
?,
?,
?);
INSERT INTO Central
(idCentral,
nomCentral,
direccio,
email,
telefon,
nif)
VALUES
(?,
?,
?,
?,
?,
?);
INSERT INTO Redireccio
(idRedirec,
dataRedirec,
dataResposta,
estat,
ComandaVendesAsoc,
IdtallerDesti,
TalleridTaller,
Peçareferencia,
CentralidCentral,
PeçacodiTrac)
VALUES
(?,
?,
?,
?,
?,
?,
?,
?,
?,
?,
?);
INSERT INTO Peticio
(idPeticio,
dataPeticio,
dataResposta,

```

```
quantitatDm,  
quantEviar,  
estat,  
comandaVendesAsoc,  
esAprovisiona,  
TalleridTaller,  
Peçareferencia,  
DistribucioidDistribucio,  
PeçacodiTrac)  
VALUES  
  (?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?);  
INSERT INTO Peça_Taller  
  (Peçareferencia,  
  TalleridTaller,  
  PeçacodiTrac,  
  estocActual,  
  estocminim,  
  preuVenda)  
VALUES  
  (?,  
  ?,  
  ?,  
  ?,  
  ?,  
  ?);  
INSERT INTO Proveidor  
  (idProv,  
  nom,  
  adreça,  
  email,  
  telf)  
VALUES  
  (?,  
  ?,  
  ?,  
  ?,  
  ?);  
INSERT INTO Grup  
  (idGrup,  
  nom,  
  descripcio)  
VALUES  
  (?,  
  ?,  
  ?);  
INSERT INTO Subgrup  
  (idSubgrup,
```

```

        nom,
        descripcio,
        GrupidGrup)
VALUES
    (? ,
    ? ,
    ? ,
    ? );
INSERT INTO Peça
    (referencia,
    codiTrac,
    descrip,
    ProveidoridProv,
    SubgrupidSubgrup)
VALUES
    (? ,
    ? ,
    ? ,
    ? ,
    ? );
INSERT INTO Taller
    (idTaller,
    nomTaller,
    direccio,
    email,
    telefon,
    nif,
    kmCentral,
    PoblacioidPoblacio)
VALUES
    (? ,
    ? ,
    ? ,
    ? ,
    ? ,
    ? ,
    ? ,
    ? ,
    ? );
INSERT INTO Administrador
    (UsuariidUsuari)
VALUES
    (? );
INSERT INTO OperariTaller
    (UsuariidUsuari,
    TalleridTaller)
VALUES
    (? ,
    ? );
INSERT INTO OperariCentral
    (UsuariidUsuari,
    CentralidCentral)
VALUES
    (? ,
    ? );
INSERT INTO Usuari
    (idUsuari,
    nom,
    cognoms,
```

```

dni,
telf,
email,
login,
password,
activo,
direccio,
RolidRol,
PoblacioidPoblacio)
VALUES
(?,
?,
?,
?,
?,
?,
?,
?,
?,
?,
?,
?,
?,
?) ;

```

### *Updates*

```

UPDATE Provincia SET
    nom = ?
WHERE
    Idprovincia = ?;
UPDATE Poblacio SET
    nom = ?,
    codPostal = ?,
    ProvinciaIdprovincia = ?
WHERE
    idPoblacio = ?;
UPDATE Rol_Menu SET

WHERE
    RolidRol = ? AND MenuidMenu = ?;
UPDATE Rol SET
    nom = ?,
    descripcio = ?
WHERE
    idRol = ?;
UPDATE Menu SET
    nom = ?,
    descripcio = ?,
    parentid = ?,
    RolidRol = ?
WHERE
    idMenu = ?;
UPDATE Distribucio SET
    Data = ?,
    decripcio = ?,
    CentralidCentral = ?

```

```

WHERE
    idDistribucio = ?;
UPDATE SortDevolucio_Peça SET
    quantitat = ?,
    snapEstoc = ?
WHERE
    DevolucioIdRebuig = ? AND Peçareferencia = ? AND PeçacodiTrac =
    ?;
UPDATE Peça_Recepcio SET
    quantitat = ?,
    snapEstoc = ?
WHERE
    Peçareferencia = ? AND RecepcioidRecepcio = ? AND PeçacodiTrac =
    ?;
UPDATE SortidaDevolucio SET
    Tipus = ?,
    dataSortida = ?,
    Motiu = ?,
    TalleridTaller = ?,
    CentralidCentral = ?
WHERE
    IdSortida = ?;
UPDATE Recepcio SET
    dataRecepcio = ?,
    descripcio = ?,
    TalleridTaller = ?,
    CentralidCentral = ?
WHERE
    idRecepcio = ?;
UPDATE Peça_Central SET
    estocActual = ?,
    estocMinim = ?,
    preuCompra = ?,
    codiTrac = ?
WHERE
    Peçareferencia = ? AND CentralidCentral = ? AND PeçacodiTrac = ?;
UPDATE Central SET
    nomCentral = ?,
    direccio = ?,
    email = ?,
    telefon = ?,
    nif = ?
WHERE
    idCentral = ?;
UPDATE Redireccio SET
    dataRedirec = ?,
    dataResposta = ?,
    estat = ?,
    ComandaVendesAsoc = ?,
    IdtallerDesti = ?,
    TalleridTaller = ?,
    Peçareferencia = ?,
    CentralidCentral = ?,
    PeçacodiTrac = ?
WHERE
    idRedirec = ?;
UPDATE Peticio SET
    dataPeticio = ?,

```

```

dataResposta = ?,
quantitatDm = ?,
quantEviar = ?,
estat = ?,
comandaVendesAsoc = ?,
esAprovisiona = ?,
TalleridTaller = ?,
Peçareferencia = ?,
DistribucioidDistribucio = ?,
PeçacodiTrac = ?
WHERE
    idPeticio = ?;
UPDATE Peça_Taller SET
    estocActual = ?,
    estocminim = ?,
    preuVenda = ?
WHERE
    Peçareferencia = ? AND TalleridTaller = ? AND PeçacodiTrac = ?;
UPDATE Proveidor SET
    nom = ?,
    adreça = ?,
    email = ?,
    telf = ?
WHERE
    idProv = ?;
UPDATE Grup SET
    nom = ?,
    descripcio = ?
WHERE
    idGrup = ?;
UPDATE Subgrup SET
    nom = ?,
    descripcio = ?,
    GrupidGrup = ?
WHERE
    idSubgrup = ?;
UPDATE Peça SET
    descrip = ?,
    ProveidoridProv = ?,
    SubgrupidSubgrup = ?
WHERE
    referencia = ? AND codiTrac = ?;
UPDATE Taller SET
    nomTaller = ?,
    direccio = ?,
    email = ?,
    telefon = ?,
    nif = ?,
    kmCentral = ?,
    PoblacioidPoblacio = ?
WHERE
    idTaller = ?;
UPDATE Administrador SET
    UsuariidUsuari = ?
WHERE
    ;
UPDATE OperariTaller SET
    UsuariidUsuari = ?,

```



```

    TalleridTaller = ?
WHERE
;
UPDATE OPerariCentral SET
    UsuariidUsuari = ?,
    CentralidCentral = ?
WHERE
;
UPDATE Usuari SET
    nom = ?,
    cognoms = ?,
    dni = ?,
    telf = ?,
    email = ?,
    login = ?,
    password = ?,
    activo = ?,
    direccio = ?,
    RolidRol = ?,
    PoblacioidPoblacio = ?
WHERE
    idUsuari = ?;

```

### Deletes

```

DELETE FROM Provincia
    WHERE Idprovincia = ?;
DELETE FROM Poblacio
    WHERE idPoblacio = ?;
DELETE FROM Rol_Menu
    WHERE RolidRol = ? AND MenuidMenu = ?;
DELETE FROM Rol
    WHERE idRol = ?;
DELETE FROM Menu
    WHERE idMenu = ?;
DELETE FROM Distribucio
    WHERE idDistribucio = ?;
DELETE FROM SortDevolucio_Peça
    WHERE DevolucioIdRebuig = ? AND Peçareferencia = ? AND
PeçacodiTrac = ?;
DELETE FROM Peça_Recepcio
    WHERE Peçareferencia = ? AND RecepcioidRecepcio = ? AND
PeçacodiTrac = ?;
DELETE FROM SortidaDevolucio
    WHERE IdSortida = ?;
DELETE FROM Recepcio
    WHERE idRecepcio = ?;
DELETE FROM Peça_Central
    WHERE Peçareferencia = ? AND CentralidCentral = ? AND
PeçacodiTrac = ?;
DELETE FROM Central
    WHERE idCentral = ?;
DELETE FROM Redireccio
    WHERE idRedirec = ?;
DELETE FROM Peticio

```

```
WHERE idPeticio = ?;  
DELETE FROM Peça_Taller  
WHERE Peçareferencia = ? AND TalleridTaller = ? AND PeçacodiTrac  
= ?;  
DELETE FROM Proveidor  
WHERE idProv = ?;  
DELETE FROM Grup  
WHERE idGrup = ?;  
DELETE FROM Subgrup  
WHERE idSubgrup = ?;  
DELETE FROM Peça  
WHERE referencia = ? AND codiTrac = ?;  
DELETE FROM Taller  
WHERE idTaller = ?;  
DELETE FROM Administrador  
WHERE ;  
DELETE FROM OperariTaller  
WHERE ;  
DELETE FROM OperariCentral  
WHERE ;  
DELETE FROM Usuari  
WHERE idUsuari = ?;
```

## 10. Annex

A continuació presentem un parell de punts que, una vegada consensuats amb tots els membres del grup, em acordat que seria necessari posar-los dintre d'aquest apartat.

D'aquesta manera s'ha pensat que es tindria un projecte més complet

### 10.1 Fitxes CRC

Tot i que no es demana, hem vist que seria útil fer un petit exemple de com es farien les fitxes CRC i que són.

A continuació, es posaran alguns exemples de fitxes CRC del subsistema de flux:

Classe	Controlador comanda
Descripció:	Classe que s'encarrega de la gestió de les comandes
Tipus:	Controlador
Característiques:	Concreta
Responsabilitats:	Poder modificar, obtenir i usar instàncies de comanda
Col·laboracions:	Col·labora amb l'entitat Comanda
Constructor:	+controladorComanda()
Atributs:	
Mètodes:	+generarAlbara(idComanda:String, dataRecepcio:Date): idAlbara:String +rebrecomanda(idComanda:String, dataRecepcio:Date) +afegirPeça(idComanda:String,peça:String) +pecesComanda(idCom:String): peces[Peces]

Classe	Controlador comanda
Descripció:	Classe que s'encarrega de la gestió de les comandes
Tipus:	Controlador
Característiques:	Concreta
Responsabilitats:	Poder modificar, obtenir i usar instàncies de comanda
Col·laboracions:	Col·labora amb l'entitat Comanda
Constructor:	+controladorComanda()
Atributs:	
Mètodes:	+generarAlbara(idComanda:String, dataRecepcio:Date): idAlbara:String +rebrecomanda(idComanda:String, dataRecepcio:Date)

	+afegirPeça(idComanda:String,peça:String) +pecesComanda(idCom:String): peces[Peces]
--	--

Classe	Controlador peces central
Descripció:	Classe que s'encarrega d'implementar els mètodes de control i supervisió del flux de peces a tot el sistema
Tipus:	Controlador
Característiques:	Concreta
Responsabilitats:	Poder modificar, obtenir i usar instàncies de peces, exercir el control del flux de peces a tot el sistema
Col·laboracions:	Col·labora amb l'entitat Comanda, operador central i operador taller.
Constructor:	+controladorPecesCentral()
Atributs:	
Mètodes:	+consultaTallers(idPeça:String): tallers[idTaller:String,nomTaller:String,quantitat:int] +consultaPeces(idTaller:String): peces[idPeça:String,quantitat:int] +consultaHistoric(idPeça:String,dataIn:Date,dataFi:Date) comandes[com:Comanda]

















Classe	Controlador peces taller
Descripció:	Classe que s'encarrega d'implementar els mètodes de control i supervisió del flux de peces, així com les operacions que calguin per a fer devolucions i mantenir l'estoc en un taller en concret,
Tipus:	Controlador
Característiques:	Concreta
Responsabilitats:	Poder modificar, obtenir i usar instàncies de peces, exercir el control del flux de peces al taller, fer operacions de devolució, de petició de peces a la central i de sortida de peces
Col·laboracions:	Col·labora amb les entitats operador central, operador taller i peces .
Constructor:	+controladorPecesTaller()
Atributs:	
Mètodes:	+consultaPeça(idPeça:String): quantitat:int +consultaPeces(): peces[ iPeça:Peces,quantitat:int] +consultaHistoric(idPeça:String,dataIn:Date,dataFi:Date): comandes[com:Comanda]

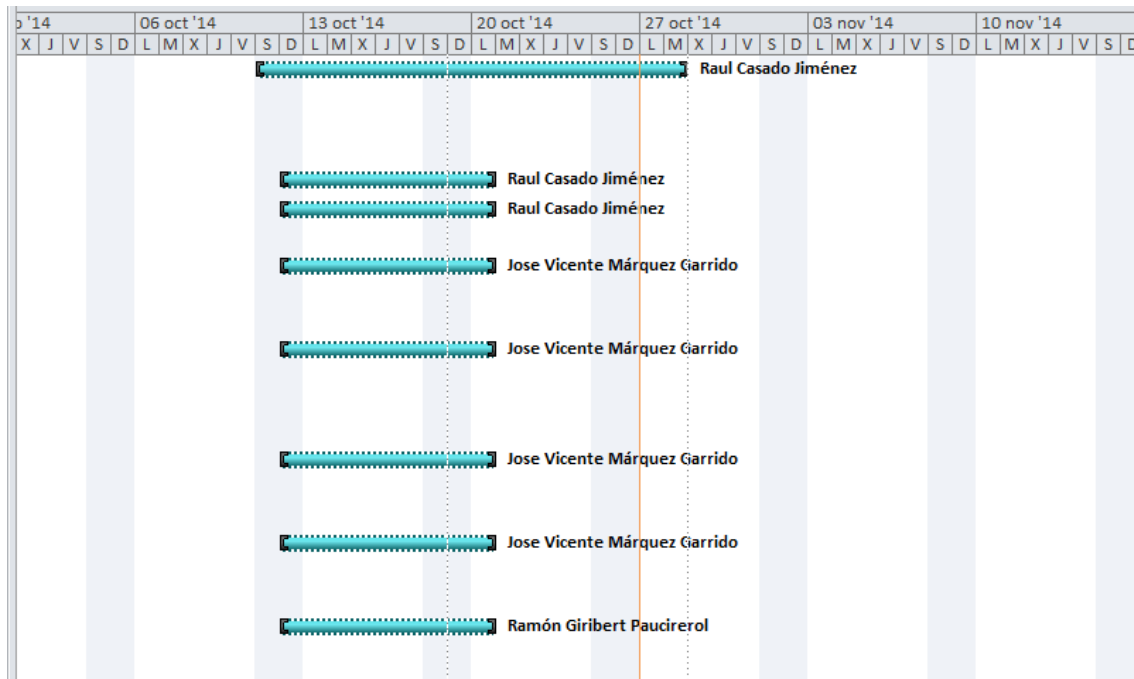
Classe	Controlador operador central
Descripció:	Classe que s'encarrega d'implementar els mètodes de control i supervisió de que disposa específicament l'operador de la central
Tipus:	Controlador
Característiques:	Concreta
Responsabilitats:	Donar accés a l'operador de la central a les funcionalitats de consulta d'avisos de la pantalla de control central i a la redirecció de clients a altres centres.
Col·laboracions:	Col·labora amb les entitats operador taller.
Constructor:	+controladorOperadorCentral()
Atributs:	
Mètodes:	+consultaAvisos() +redireccionament(idTallerIn:String,idTallerDest:String)

## 10.2 Planificació detallada

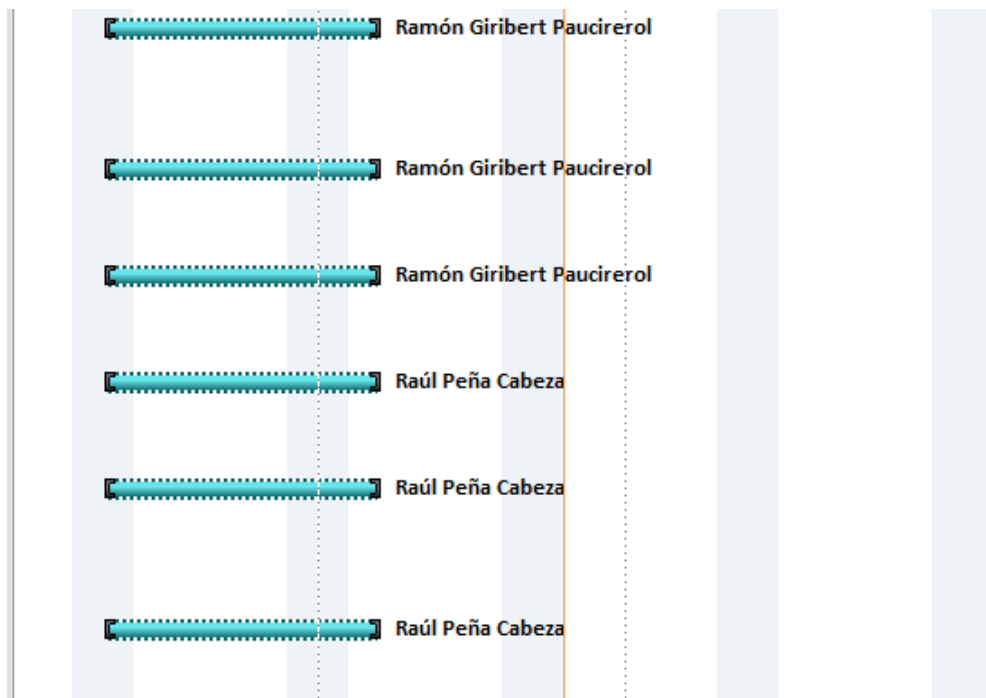
També s'ha establert entre tots els membres del grup que a l'annex es podria posar la planificació detallada de cada una de les parts que ha fet cada membre del grup i quants dies s'ha trigat en fer cada una de les parts del projecte.

Aquesta planificació també s'adjunta amb el seu diagrama de 'Gantt' corresponent:

1			Log assignació de tasques i creacions actes reunions i situacions actuals	13 días	sáb 11/10/14	mar 28/10/14	Raul Casado Jiménez
2			Introducció	7 días	dom 12/10/14	lun 20/10/14	Raul Casado Jiménez
3			Breu descripció de subsistemes	7 días	dom 12/10/14	lun 20/10/14	Raul Casado Jiménez
4			Breu descripció textual subsistema de manteniment	7 días	dom 12/10/14	lun 20/10/14	Jose Vicente Márquez Garrido
5			Diagrama de classes i jerarquia UML subsistema de manteniment	7 días	dom 12/10/14	lun 20/10/14	Jose Vicente Márquez Garrido
6			Diagrama global subsistema de manteniment	7 días	dom 12/10/14	lun 20/10/14	Jose Vicente Márquez Garrido
7			Diagrama de seqüència subsistema de manteniment	7 días	dom 12/10/14	lun 20/10/14	Jose Vicente Márquez Garrido
8			Breu descripció textual subsistema de control de fluxe	7 días	dom 12/10/14	lun 20/10/14	Ramón Giribert Paucirerol



9			Diagrama de classes i jerarquia UML subsistema de control de fluxe	7 días	dom 12/10/14	lun 20/10/14	Ramón Giribert Paucirerol
10			Diagrama global subsistema de control de fluxe	7 días	dom 12/10/14	lun 20/10/14	Ramón Giribert Paucirerol
11			Diagrama de seqüència subsistema de control de fluxe	7 días	dom 12/10/14	lun 20/10/14	Ramón Giribert Paucirerol
12			Breu descripció textual subsistema de llistat i estadístiques	7 días	dom 12/10/14	lun 20/10/14	Raúl Peña Cabeza
13			Diagrama de classes i jerarquia UML subsistema de control i estadístiques	7 días	dom 12/10/14	lun 20/10/14	Raúl Peña Cabeza
14			Diagrama global subsistema de llistat i estadístiques	7 días	dom 12/10/14	lun 20/10/14	Raúl Peña Cabeza



15			Diagrama de seqüència subsistema de llistat i estadístiques	7 dies	dom 12/10/14	lun 20/10/14	Raúl Peña Cabeza
16			Breu descripció textual subsistema de connexió	7 dies	dom 12/10/14	lun 20/10/14	Luis Belmonte Sain
17			Diagrama de classes i jerarquia UML subsistema de connexió	7 dies	dom 12/10/14	lun 20/10/14	Luis Belmonte Sain
18			Diagrama global subsistema de connexió	7 dies	dom 12/10/14	lun 20/10/14	Luis Belmonte Sain
19			Diagrama de seqüència subsistema de connexió	7 dies	dom 12/10/14	lun 20/10/14	Luis Belmonte Sain
20			Possible reutilització de classes entre subsistemes	2 dies	lun 20/10/14	mar 21/10/14	Raul Casado Jiménez
21			Relació de classes a implementar	2 dies	mié 22/10/14	jue 23/10/14	Ramón Giribert Paucirerol; Luis Belmonte Sain
22			Disseny de la persistència	3 dies	jue 23/10/14	dom 26/10/14	Jose Vicente Márquez Garrido; Ramón Giribert Paucirerol
23			Passar projecte a net i informes	2 dies	dom 26/10/14	lun 27/10/14	Raul Casado Jiménez
24			Revisió del projecte	2 dies	lun 27/10/14	mar 28/10/14	Jose Vicente Márquez Garrido; Luis Belmonte Sain; Ramón Giribert Paucirerol; Raúl Peña Cabeza

