

# Treball final de carrera - J2EE

## Cooperativa Agrícola Virtual

Memòria del projecte

---

**Nom Estudiant: David Álvarez Arrarás**  
ETIS

**Nom Consultor: Jose Juan Rodríguez**

20/06/2011

Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons. Podeu copiar-lo, distribuir-lo i transmetre'ls públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada. La llicència completa es pot consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.

### **Dedicatòria i agraïments**

Dedico aquest treball final de carrera als meus pares, pel seu constant recolzament i per tots els ànims que m'han donat al llarg d'aquest temps.

També vull dedicar-se'l a la meva novia, pel gran recolzament que m'ha donat i agrair-li la seva comprensió per tots els caps de setmana i festius que hem tingut que passar a casa per poder estudiar.

A tots ells, moltes gràcies.

### **Resum**

En aquest treball final de carrera de l'àrea J2EE, s'ha realitzat l'anàlisi, disseny i implementació d'una aplicació per una cooperativa agrícola, a partir de la qual la cooperativa podrà mostrar i vendre els seus productes, administrar els productes, clients i treballadors. Per una altra banda el treballadors podran introduir les collites que facin i els clients podran enregistrar-se i realitzar les seves compres. Tota l'aplicació esta basada en l'arquitectura J2EE.

Actualment, les empreses es veuen amb la necessitat d'utilitzar Internet com a mitjà de comunicació, per tal de poder ser més competitius. Per aquest motiu, van augmentar les demandes d'aquest tipus d'aplicacions i han anat apareixent diferents productes per a la creació d'aplicacions distribuïdes de cara al món empresarial, com a estàndard de Java, Sun Microsystems va dissenyar J2EE. L'arquitectura J2EE ha permès un augment en la productivitat d'aquestes aplicacions amb una forma senzilla i eficient de programar. J2EE especifica un model estàndard d'aplicacions, basades en capes, contenidors, components i serveis. J2EE parteix de la plataforma Java i per tant el seu llenguatge de programació es Java, disposa d'unes especificacions API com JDBC, XML, JMS, RMI, etc. I defineix els seus components com poden ser JSP, sevlets, enterprise Java Beans(EJB), portlets, etc.

## Índex de continguts

1. Capítol 1: Introducció.....	pàg.6
1.1. Justificació del Treball final de carrera i context en el qual es desenvolupa.....	pàg.6
1.2. Objectius del Treball final de carrera.....	pàg.6
1.3. Enfocament del projecte.....	pàg.7
1.4. Planificació del projecte.....	pàg.7
1.4.1. Recursos.....	pàg.7
1.4.2. Requeriments de software.....	pàg.7
1.4.3. Temporització.....	pàg.8
1.5. Productes obtinguts.....	pàg.9
1.6. Breu descripció de la resta de capítols....	pàg.9
2. Capítol 2: Anàlisi.....	pàg.10
2.1. Descripció funcional del projecte.....	pàg.10
2.2. Actors.....	pàg.11
2.3. Diagrama de casos d'us.....	pàg.12
2.4. Descripció de casos d'us.....	pàg.13
3. Capítol 3: Disseny.....	pàg.21
3.1. Diagrama de classes.....	pàg.21
3.2. Diagrames de seqüència.....	pàg.22
3.3. Diagrama d'estats.....	pàg.33
3.4. Disseny Base de dades.....	pàg.33
3.5. Arquitectura del sistema.....	pàg.36
3.5.1. Struts 2.....	pàg.37
3.5.2. Hibernate.....	pàg.38
3.6. Patrons de disseny que s'utilitzaran.....	pàg.39
3.7. Interfície gràfica.....	pàg.41
4. Capítol 4: Implementació.....	pàg.42
4.1. Eines de desenvolupament.....	pàg.42
4.1.1. Base de dades.....	pàg.42
4.1.2. Utilitats BBDD.....	pàg.42
4.1.3. Entorn de desenvolupament.....	pàg.42
4.1.4. Contenedor d'aplicacions.....	pàg.42
4.1.5. Struts2.....	pàg.42
4.1.6. Persistència de dades.....	pàg.43
4.2. Manual de configuració de les eines emprades.....	pàg.43
4.2.1. Configuració base de dades.....	pàg.43
4.2.2. Configuració d'hibernate.....	pàg.44
4.2.3. Configuració Struts2.....	pàg.46
4.3. Estructura del projecte.....	pàg.47
4.4. Manual d'usuari.....	pàg.50
4.4.1. Zona administrador.....	pàg.50
4.4.2. Zona treballadors.....	pàg.54
4.4.3. Zona clients.....	pàg.55
4.5. Proves realitzades.....	pàg.57
4.6. Alguns consells per a provar l'aplicació..	pàg.58
5. Capítol 5: Conclusions.....	pàg.59
6. Glossari.....	pàg.59
7. Bibliografia.....	pàg.60

## Índex de figures

Figura 1. Diagrama de Gantt de la temporització.....	pàg.8
Figura 2. Taula temporització.....	pàg.8
Figura 3. Diagrama general casos d'us.....	pàg.12
Figura 4. Diagrama de classes.....	pàg.21
Figura 5. Diagrama de seqüència d'autenticació d'usuaris.....	pàg.22
Figura 6. Diagrama de seqüència consultar productes per categoria.....	pàg.22
Figura 7. Diagrama de seqüència de l'alta d'un client(Enregistrament).....	pàg.23
Figura 8. Diagrama de seqüència consultar carro compra.....	pàg.23
Figura 9. Diagrama de seqüència modificació d'una línia del carro.....	pàg.23
Figura 10. Diagrama de seqüència eliminació d'una línia del carro.....	pàg.24
Figura 11. Diagrama de seqüència realitza comanda carro.....	pàg.24
Figura 12. Diagrama de seqüència consulta i modificació dades personals.....	pàg.25
Figura 13. Diagrama de seqüència tancament de sessió.....	pàg.25
Figura 14. Diagrama de seqüència afegir collita.....	pàg.25
Figura 15. Diagrama de seqüència afegir producte.....	pàg.26
Figura 16. Diagrama de seqüència consultar producte.....	pàg.26
Figura 17. Diagrama de seqüència eliminar producte.....	pàg.27
Figura 18. Diagrama de seqüència modificar producte.....	pàg.27
Figura 19. Diagrama de seqüència alta treballador.....	pàg.28
Figura 20. Diagrama de seqüència consultar treballador.....	pàg.28
Figura 21. Diagrama de seqüència baixa treballador.....	pàg.29
Figura 22. Diagrama de seqüència modificació dades d'un treballador.....	pàg.30
Figura 23. Diagrama de seqüència consultar client.....	pàg.31
Figura 24. Diagrama de seqüència baixa d'un client.....	pàg.32
Figura 25. Diagrama d'estats.....	pàg.33
Figura 26. Disseny base de dades.....	pàg.33
Figura 27. Arquitectura J2EE.....	pàg.36
Figura 28. Model-Vista-Controlador.....	pàg.38
Figura 29. Patró MVC.....	pàg.39
Figura 30. Zona pública.....	pàg.41
Figura 31. Zona privada.....	pàg.41
Figura 32. Llibreries bàsiques Struts2.....	pàg.43
Figura 33. Llibreries Hibernate.....	pàg.43
Figura 34. Estructura principal del projecte.....	pàg.47
Figura 35. Estructura carpeta JSP.....	pàg.47
Figura 36. Estructura paquet predeterminat.....	pàg.48
Figura 37. Estructura Actions.....	pàg.48
Figura 38. Estructura Beans.....	pàg.49
Figura 39. Estructura Interceptor.....	pàg.49

Figura 40.	Estructura Mapeig.....	pàg.49
Figura 41.	Pàgina principal.....	pàg.50
Figura 42.	Pàgina administració.....	pàg.50
Figura 43.	Pàgina gestor productes.....	pàg.51
Figura 44.	Imatges adjuntes.....	pàg.51
Figura 45.	Pàgina afegir producte.....	pàg.52
Figura 46.	Pàgina cerca de productes.....	pàg.52
Figura 47.	Pàgina edició de productes.....	pàg.53
Figura 48.	Pàgina cerca de clients.....	pàg.53
Figura 49.	Pàgina eliminació de clients.....	pàg.54
Figura 50.	Pàgina principal area treballadors.....	pàg.54
Figura 51.	Pàgina afegir collita.....	pàg.55
Figura 52.	Pàgina principal area clients.....	pàg.55
Figura 53.	Pàgina catàleg productes.....	pàg.56
Figura 54.	Pàgina carro compra.....	pàg.56
Figura 55.	Pàgina dades personals.....	pàg.56
Figura 56.	Test 1.....	pàg.57
Figura 57.	Test 2.....	pàg.57
Figura 58.	Test 3.....	pàg.58
Figura 59.	Test 3.1.....	pàg.58

**MEMÒRIA****1. Capítol 1: Introducció****1.1. Justificació del Treball final de carrera i context en el qual es desenvolupa**

En aquest projecte es crearà una aplicació web i botiga virtual per a una cooperativa d'agricultors, que es dediquen a la collita d'hortalisses (verdures i llegums), fruites i cereals.

En aquesta aplicació cada treballador de la cooperativa disposarà d'un usuari i contrasenya per tal d'accedir com a treballador i poder introduir les seves collites. D'aquesta manera la cooperativa tindria contínuament actualitzat el seu estoc de productes.

Per altre banda en aquesta aplicació web els clients o possibles clients (visitants) podran veure tots els productes que disposa la cooperativa i es podran enregistrar com a clients per a poder realitzar les compres dels productes.

Per tant, tindrem 4 actors: visitant, client, treballador i administrador.

L'administrador serà l'encarregat d'introduir, eliminar o modificar els productes. També de donar d'alta, baixa o modificació de les dades dels treballadors. I la possible eliminació d'algun client, si es necessari. L'alta de client la farà el propi client en enregistrar-se i si necessita modificar alguna dada seva, posteriorment també podrà fer-ho entrant a la seva compte.

**1.2. Objectius del Treball final de carrera**

L'objectiu principal es l'anàlisi, disseny e implementació d'una aplicació per una cooperativa agrícola amb l'arquitectura J2EE. D'aquesta manera s'adquiriran els coneixements necessaris sobre el Model Vista-Controlador (MVC), el framework Struts2 que també segueix el MVC, Hibernate, JSP, EJB i d'altres.

Un altre objectiu es proporcionar una nova eina per a les cooperatives agrícoles, que els permeti de disposar en temps real de les dades de les collites, disposar de diferents magatzems (descentralització) que els permetrà estalviar en transport i millorarà coordinació del transport, afavoreix l'unió de cooperatives de diferents zones geogràfiques, ja que amb la venda per Internet entrem en un mercat més global. Tot això podrà ser administrat per una o diverses persones sense gaires coneixements informàtics (no tindran cap necessitat de modificar el codi de la web), ja que l'aplicació ha d'oferir un entorn amigable e intuïtiu.

### **1.3. Enfocament i mètode seguit**

La metodologia que es seguirà es un cicle de vida clàssic(en cascada)sense tenir en compte l'etapa de manteniment. L'aprenentatge de la tecnologia J2EE es realitzarà en paral·lel durant el període d'anàlisi i disseny.

Les etapes definides son:

- o *Pla de treball + estudi inicial tecnologia J2EE(cerca informació dels diferents frameworks i eines J2EE per poder definir més correctament el disseny e implementació que es realitzaran)*
- o *Estudi tecnologia J2EE*
- o *Anàlisi*
- o *Disseny*
- o *Implementació*
- o *Proves*

### **1.4. Planificació del projecte**

#### **1.4.1. Recursos**

Es disposarà d'una persona entre dues i tres hores diàries de dilluns a divendres i entre 6 i 8 hores dissabtes i diumenges.

#### **1.4.2. Requeriments de software**

Les necessitats de software detectades inicialment van ser:

- o *Openproj*
- o *MagicDraw UML Personal Edition*
- o *Netbeans 6.9.1*
- o *Oracle 10g Express Edition*
- o *Apache Tomcat 6.0.2.6*

Posteriorment amb l'estudi inicial de la tecnologia J2EE s'han pogut definir la resta de requeriments de software, que son Strut2 i Hibernate.

### 1.4.3. Temporització

La temporització l'he definit basant-me en els terminis establerts per l'entrega de les diferents PACs. Tota la temporització es troba definida al següent diagrama de Gantt.

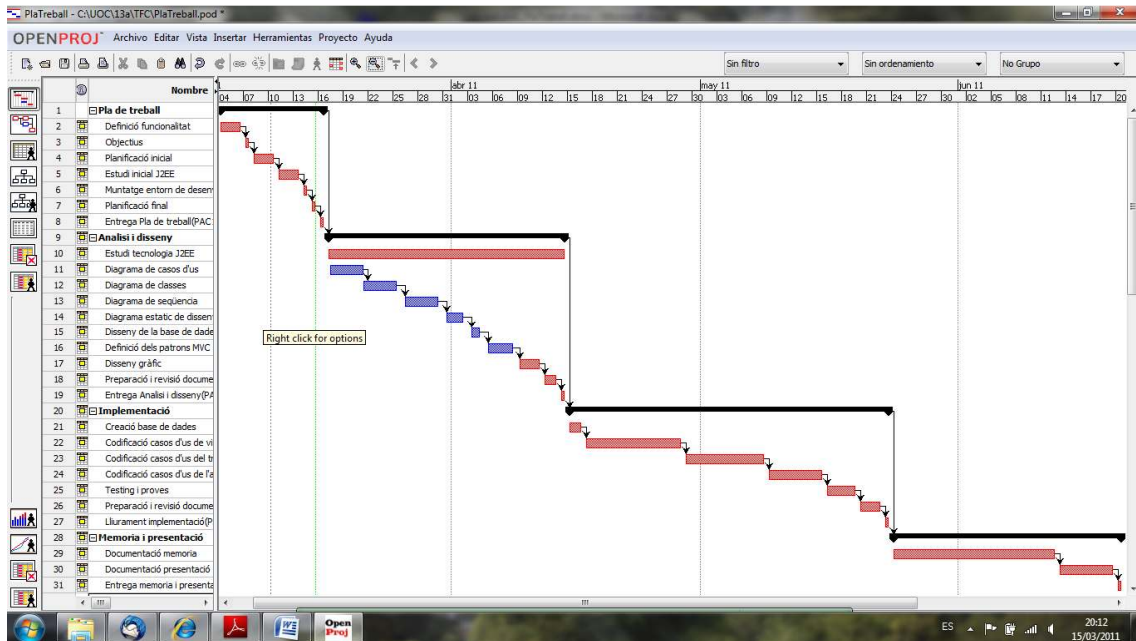


Figura 1. Diagrama de Gantt de la temporització

Per a la creació del diagrama de Gantt anterior s'ha utilitzat la següent taula on podem veure els terminis establerts per a cada tasca i la seva durada total.

	Nombre	Duració	Inicio	Terminado
1	<b>Pla de treball</b>	<b>13 days?</b>	<b>4/03/11 8:00</b>	<b>16/03/11 17:00</b>
2	Definició funcionalitat	3 days?	4/03/11 8:00	6/03/11 17:00
3	Objectius	1 day?	7/03/11 8:00	7/03/11 17:00
4	Planificació inicial	3 days?	8/03/11 8:00	10/03/11 17:00
5	Estudi inicial J2EE	3 days?	11/03/11 8:00	13/03/11 17:00
6	Muntatge entorn de desenvolupament	1 day?	14/03/11 8:00	14/03/11 17:00
7	Planificació final	1 day?	15/03/11 8:00	15/03/11 17:00
8	Entrega Pla de treball(PAC 1)	1 day?	16/03/11 8:00	16/03/11 17:00
9	<b>Anàlisi i disseny</b>	<b>29 days?</b>	<b>17/03/11 8:00</b>	<b>14/04/11 17:00</b>
10	Estudi tecnologia J2EE	29 days?	17/03/11 8:00	14/04/11 17:00
11	Diagrama de casos d'us	4 days?	17/03/11 11:00	21/03/11 11:00
12	Diagrama de classes	4 days?	21/03/11 11:00	25/03/11 11:00
13	Diagrama de seqüència	4 days?	26/03/11 12:00	30/03/11 13:00
14	Diagrama estatic de disseny	2 days?	31/03/11 11:00	2/04/11 11:00
15	Disseny de la base de dades	1 day?	3/04/11 11:00	4/04/11 11:00
16	Definició dels patrons MVC	3 days?	5/04/11 11:00	8/04/11 11:00
17	Disseny gràfic	3 days?	9/04/11 7:00	11/04/11 17:00
18	Preparació i revisió documentació	2 days?	12/04/11 8:00	13/04/11 17:00
19	Entrega Anàlisi i disseny(PAC 2)	1 day?	14/04/11 8:00	14/04/11 17:00
20	<b>Implementació</b>	<b>39 days?</b>	<b>15/04/11 8:00</b>	<b>23/05/11 17:00</b>
21	Creació base de dades	2 days?	15/04/11 8:00	16/04/11 17:00
22	Codificació casos d'us de visitant i client	12 days?	17/04/11 8:00	28/04/11 17:00
23	Codificació casos d'us del treballador	10 days?	29/04/11 8:00	8/05/11 17:00
24	Codificació casos d'us de l'administrador	7 days?	9/05/11 8:00	15/05/11 17:00
25	Testing i proves	4 days?	16/05/11 8:00	19/05/11 17:00
26	Preparació i revisió documentació	3 days?	20/05/11 8:00	22/05/11 17:00
27	Lliurament implementació(PAC 3)	1 day?	23/05/11 8:00	23/05/11 17:00
28	<b>Memòria i presentació</b>	<b>28 days?</b>	<b>24/05/11 8:00</b>	<b>20/06/11 17:00</b>
29	Documentació memòria	20 days?	24/05/11 8:00	12/06/11 17:00
30	Documentació presentació	7 days?	13/06/11 8:00	19/06/11 17:00
31	Entrega memòria i presentació(PAC 4)	1 day?	20/06/11 8:00	20/06/11 17:00

Figura 2. Taula temporització



### **1.5. Productes obtinguts**

S'ha obtingut aquesta memòria com a descripció i justificació de les solucions adoptades per a dur a terme el projecte.

També s'ha obtingut l'aplicació web per a la cooperativa agrícola que ha sigut empaquetada en un fitxer war i que pot ser desplegada directament en un servidor d'aplicacions Tomcat.

Per últim, també s'entrega una presentació del treball final de carrera, que es una síntesi del projecte.

### **1.6. Breu descripció de la resta de capítols**

Als següents capítols trobarem tot el desenvolupament del projecte. Al capítol 2 trobarem la fase d'anàlisi, on s'explica quines son les funcionalitats del projecte, explicant els actors del sistema, quin son els casos d'us de cada actor i la seva descripció.

Al capítol 3 trobarem la fase del disseny, on s'explica com actuarà el sistema. Aquí trobarem el diagrama de classes, diagrames de seqüència i estats, l'explicació de l'arquitectura del sistema, quins patrons de disseny s'han utilitzat i una breu descripció del format de l'interfície gràfica.

Al capítol 4 s'explica la fase d'implementació, quines eines s'han utilitzats per a realitzar l'implementació, com instal·lar-les, com està estructurat, les proves de rendiment realitzades, un manual d'usuari i uns consells pràctics per tal d'inicialitzar fàcilment l'aplicació.

Finalment al capítol 5, trobarem les conclusions a les que s'ha arribat a partir de la realització del projecte.

## 2. Capítol 2: Anàlisi

### 2.1. Descripció funcional del projecte

La pàgina principal de l'aplicació ha de donar una imatge general del funcionament de la cooperativa i ha de permetre a qualsevol usuari accedir a l'informació de l'empresa referent a com contactar, quin es el seu negoci, quins son els seus productes. Per tant, a partir de la pàgina principal es tindrà accés a una pàgina on estar explicat com contactar amb la cooperativa, a una pàgina que explicarà amb detall el seu negoci i al catàleg de productes (aquest catàleg per tal de donar una imatge ben estructurada, estarà diferenciat per categories). A més, ha de permetre als usuaris enregistrar-se per tal de fer-se clients de la cooperativa i autenticar-se per tal d'accedir a la zona de clients.

Existiran 3 zones privades, una per als clients, una per als treballadors i una per a l'administrador. Per tal d'accedir a les zones privades, s'haurà d'estar registrat al sistema amb el rol d'usuari corresponent.

A partir de la pàgina principal introduint el nom d'usuari i la seva contrasenya, el sistema buscarà si l'usuari existeix, si es així, verificarà les dades introduïdes i si son correctes, li donarà els permisos corresponents al seu rol, enviant-lo cap a la zona privada corresponent al seu rol.

Per tant, si l'usuari autenticat es un client, es un treballador o un administrador llavors accedirà a la zona clients, treballadors o administrador respectivament.

Els clients a la seva zona privada, podran visualitzar i modificar les seves dades personals, navegar pel catàleg de productes, visualitzar i modificar el carro de la compra i confirmar la compra dels productes del carro. Per a realitzar les compres els clients aniran navegant pel catàleg de productes diferenciat per categories i podran afegir al carro tots els productes que vulguin, cada cop que afegeixin un producte al carro, s'afegirà un quilo del producte seleccionat (la comanda mínima per producte es un quilo), el catàleg de productes també a de permetre visualitzar una descripció detallada del producte al seleccionar un producte. Sempre que vulguin podran visitar el carro de la compra, on podran modificar o eliminar les línies del carro, a cada línia tindrem un producte diferent amb la quantitat demanada i el seu preu. Si el client esta d'acord amb els productes del carro només a d'indicar comprar i es posa en marxa la comanda. Per tal de mantenir informat al client del cost del carro de la compra, a totes les pàgines en un lateral es trobarà la informació del cost total del seu carro.

Els treballadors a la seva zona podran afegir totes les collites que facin. A cada collita indicaran el producte, la categoria del producte i la quantitat recol·lectada. Els treballadors també podran visualitzar i modificar les seves dades personals.

L'administrador a la seva zona privada podrà gestionar els productes (altes, baixes, consultes i modificacions), gestionar els clients (baixes i consultes) i els treballadors (altes, baixes i consultes).

A totes les zones privades existirà un menú al lateral per tal de permetre a l'usuari moure's per l'aplicació amb més facilitat i poder anar a l'apartat que vulgui només prement un botó.

## 2.2. Actors

A partir de l'especificació funcional de projecte es poden diferenciar 2 tipus d'actors, els que no es troben registrats al sistema i els que si estan registrats, per tant, tindrem l'actor Anònim i l'actor Usuari respectivament.

L'actor Anònim podrà visitar la zona que anomenarem pública (accessible per qualsevol usuari) i per tant podrà visitar la pàgina principal, el catàleg de productes, pàgina contactar i qui som.

L'actor Usuari com es troba registrat al sistema podrà autenticar-se com a tal (iniciar sessió), consultar/modificar les seves dades personals i tancar sessió. A més existeixen 3 rols diferents d'Usuari, per tant, cada rol correspon a un actor diferent, aquests actors son Client, Treballador i Administrador, on tots hereten d'Usuari però a més cadascú té les funcionalitats específiques del seu rol.

Per una altra banda també tenim l'actor Sistema, que realitzarà l'actualització de les existències dels productes quan es realitzin comandes o s'afegeixin collites, aquest actor també s'encarregarà de buidar el carro un cop s'hagi confirmat una comanda.

Per tant tindrem 6 actors:

- Anònim.
- Usuari.
- Client.
- Treballador.
- Administrador.
- Sistema.

### 2.3. Diagrama general de casos d'us

En el següent diagrama podem veure la funcionalitat necessària per a cada actor.

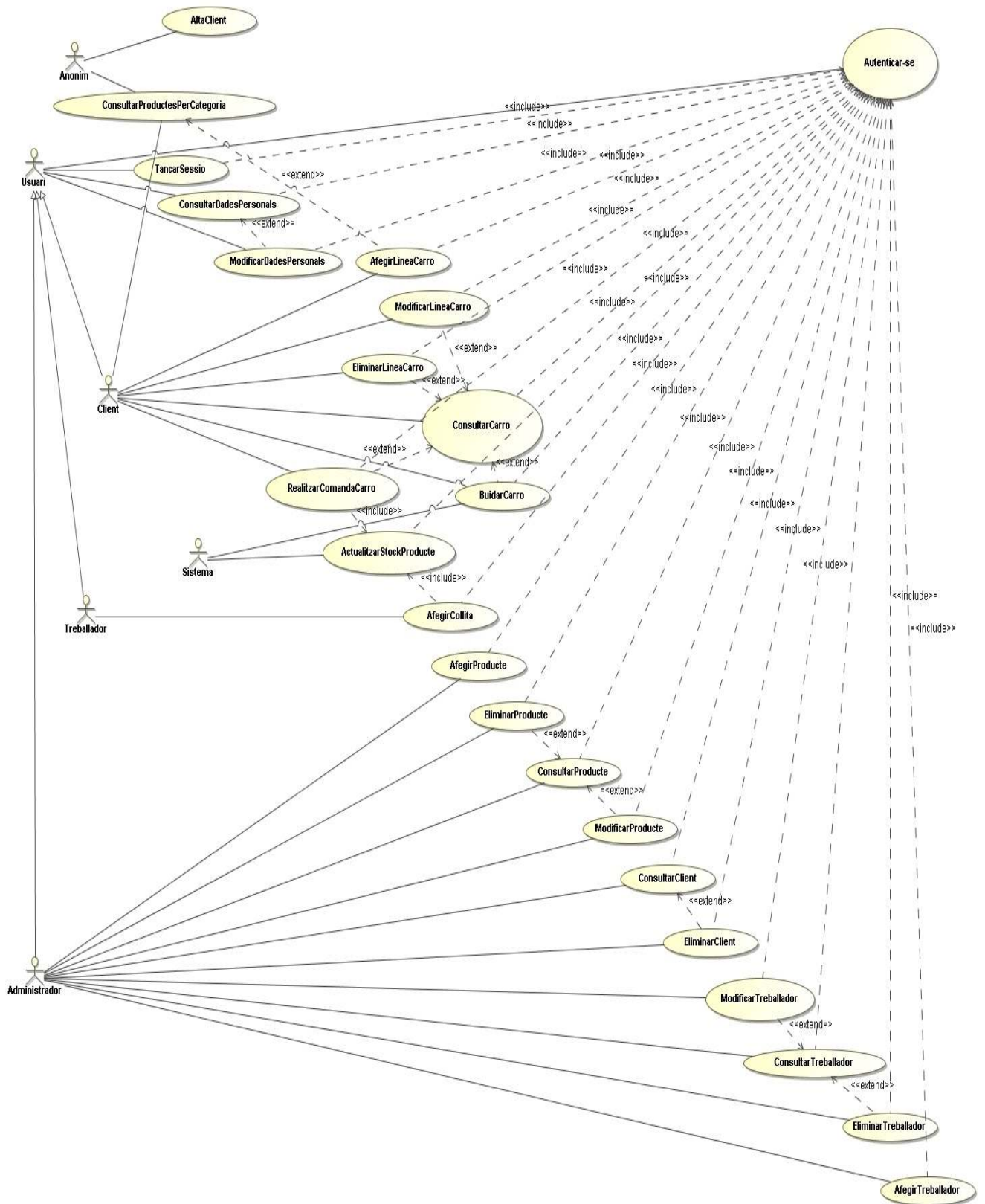


Figura 3. Diagrama general casos d'us

## 2.4. Descripció de casos d'ús

### Cas d'ús "AltaClient"

Resum funcionalitat general: Qualsevol visitant podrà donar-se d'alta com a client de la cooperativa.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de anònim

Actors: Anònim.

Casos d'ús relacionats: Cap.

Precondició: Cap.

Postcondició: El Client queda donat d'alta al sistema.

Descripció: El futur client accedeix al formulari d'enregistrament. Emplenarà les dades, el sistema comprovarà les dades i les enregistrarà al sistema si son correctes, sinó enviarà un missatge indicant el problema.

### Cas d'ús "ConsultarProductesPerCategoria"

Resum funcionalitat general: Qualsevol visitant o client podrà consultar el catàleg de productes.

Paper dins el treball de l'usuari: És un dels casos d'ús principals d'anònim i client.

Actors: Anònim, Client.

Casos d'ús relacionats: Cap.

Precondició: Cap.

Postcondició: El Client visualitzarà els productes de la categoria seleccionada.

Descripció: El visitant o client accedeix al catàleg, seleccionant la categoria dels productes que esta interessat, els sistema mostrarà aquest productes.

### Cas d'ús "Autenticar-se"

Resum funcionalitat general: Qualsevol usuari podrà autenticar-se davant el sistema.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels usuaris.

Actors: Usuari.

Casos d'ús relacionats: Cap.

Precondició: Cap.

Postcondició: L'usuari serà reconegut pel sistema i obtindrà els privilegis que li pertocuen al seu rol (Client, Treballador, Administrador).

Descripció: L'usuari introdueix el seu nom d'usuari i contrasenya, si son correctes, el sistema el redireccionarà cap a la zona privada que li pertoca segons el seu rol. Si les dades introduïdes no es troben al sistema o no s'han introduït, el sistema enviarà un missatge comunicant a l'usuari el problema.

### Cas d'ús "TancarSessio"

Resum funcionalitat general: Qualsevol usuari podrà tancar la sessió per tal de sortir de l'aplicació.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels usuaris.

Actors: Usuari.

Casos d'ús relacionats: Autenticar-se.

Precondició: S'ha d'estar registrat e identificat al sistema.

Postcondició: L'actual sessió es tancarà i totes les dades de la sessió s'esborraran.

Descripció: L'usuari pitjarà el botó tancar sessió i el sistema tancarà l'actual sessió, esborrant abans totes les dades de la sessió, finalment l'usuari es reenviat cap a la pàgina principal(zona pública) i si vol tornar a accedir a l'aplicació s'haurà de tornar a autenticar.

### **Cas d'ús "ConsultarDadesPersonals"**

Resum funcionalitat general: Qualsevol usuari podrà consultar les seves dades personals.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels usuaris.

Actors: Usuari.

Casos d'ús relacionats: Autenticar-se.

Precondició: S'ha d'estar registrat e identificat al sistema.

Postcondició: Es visualitzaran les dades de l'usuari.

Descripció: L'usuari pitjarà el botó consultar dades personals i el sistema mostrarà les seves dades.

### **Cas d'ús "ModificarDadesPersonals"**

Resum funcionalitat general: Qualsevol usuari podrà modificar les seves dades personals.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels usuaris.

Actors: Usuari.

Casos d'ús relacionats: Autenticar-se,  
ConsultarDadesPersonals.

Precondició: S'ha d'estar registrat e identificat al sistema i estar consultant les dades personals.

Postcondició: Els noves dades personals de l'usuari quedarà emmagatzemades al sistema.

Descripció: L'usuari pitjarà el botó consultar dades personals i el sistema mostrarà les seves dades, l'usuari modificarà les dades que vulgui i pitjarà el botó modificar, el sistema comprovarà les dades i si son correctes les emmagatzemarà enviant un missatge de dades modificades correctament, si les dades introduïdes tenen algun error o estan en blanc, llavors el sistema retornarà un missatge indicant el problema, sense modificar res.

### **Cas d'ús "AfegirLineaCarro"**

Resum funcionalitat general: Els clients podran afegir al carro de la compra tants productes del catàleg com vulguin.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels clients.

Actors: Client.

Casos d'ús relacionats: Autenticar-se,  
ConsultarProductesPerCategoria.

Precondició: S'ha d'estar registrat e identificat al sistema com a client i estar consultant el catàleg de productes.

Postcondició: El producte seleccionat s'afegirà al carro de la compra.

Descripció: El client pitjarà el boto afegir corresponent a un producte, el producte s'afegirà al carro, si el producte no existeix al carro, es generarà una nova línia per al producte, si el producte ja es present al carro, s'incrementarà la quantitat del producte en 1kg.

#### **Cas d'ús "ConsultarCarro"**

Resum funcionalitat general: Els clients podran visualitzar els productes del seu carro de la compra en qualsevol moment.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels clients.

Actors: Client.

Casos d'ús relacionats: Autenticar-se.

Precondició: S'ha d'estar registrat e identificat al sistema com a client.

Postcondició: Es visualitzaran els producte del carro.

Descripció: El client pitjarà el botó consultar carro, el sistema mostrarà els productes del carro i el cost total del carro per pantalla.

#### **Cas d'ús "ModificarLineaCarro"**

Resum funcionalitat general: Els clients podran modificar la quantitat de cada producte que tenen al carro, la quantitat mínima per a qualsevol producte sempre serà un quilo i la quantitat màxima serà el nombre d'existències del producte.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels clients.

Actors: Client.

Casos d'ús relacionats: Autenticar-se, ConsultarCarro.

Precondició: S'ha d'estar registrat e identificat al sistema com a client i estar consultant el carro.

Postcondició: La quantitat del producte serà modificada.

Descripció: El client pitjarà la fletxa amunt per augmentar la quantitat del producte que vulgui en 1kg o pitjarà la fletxa avall per disminuir la quantitat en un 1kg.

#### **Cas d'ús "EliminarLineaCarro"**

Resum funcionalitat general: Els clients podran eliminar tantes línies del carro com vulguin.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels clients.

Actors: Client.

Casos d'ús relacionats: Autenticar-se, ConsultarCarro.

Precondició: S'ha d'estar registrat e identificat al sistema com a client i estar consultant el carro.

Postcondició: El producte corresponent a la línia s'elimina del carro.

Descripció: El client pitjarà el botó esborrar de la línia corresponent al producte que l'interessi treure del carro i el sistema eliminarà la línia.

### **Cas d'ús "ActualitzarStockProducte"**

Resum funcionalitat general: El sistema actualitza automàticament les existències d'un producte.

Paper dins el treball de l'usuari: És un dels casos d'ús principals del sistema.

Actors: Sistema.

Casos d'ús relacionats: RealitzarComandaCarro, AfegirCollita.

Precondició: El sistema ha afegit una nova línia a una comanda o ha afegit una nova collita.

Postcondició: La quantitat del producte serà modificada.

Descripció: El sistema quant es afegeix una nova línia a una comanda, resta la quantitat comprada d'aquell producte a les existències del producte i si el que s'ha realitzat es una collita, llavors suma la quantitat recol·lectada a les existències del producte.

### **Cas d'ús "BuidarCarro"**

Resum funcionalitat general: El sistema buidarà el carro d'un client cada vegada que es realitza la comanda d'un carro, un client pot buidar el seu carro en qualsevol moment.

Paper dins el treball de l'usuari: És un dels casos d'ús principals del sistema i del client.

Actors: Sistema, Client.

Casos d'ús relacionats: RealitzarComandaCarro.

Precondició: Si es tracta d'un client cap, si es tracta del sistema la precondició es que s'ha realitzar una comanda d'un carro.

Postcondició: El carro s'ha buidat.

Descripció: El client a pitjat el botó buidar carro o ha realitzat una comanda, llavors el sistema eliminarà del carro tots els productes.

### **Cas d'ús "RealitzarComandaCarro"**

Resum funcionalitat general: Els clients podran confirmar la compra del carro quan vulguin, sempre i quan tinguin algun producte dins del carro.



Paper dins el treball de l'usuari: És un dels casos d'ús principals dels clients.

Actors: Client.

Casos d'ús relacionats: Autenticar-se, ConsultarCarro, ActualitzarStockProducte, BuidarCarro.

Precondició: S'ha d'estar registrat e identificat al sistema com a client i estar consultant el carro.

Postcondició: Tots els productes del carro son passats a una nova comanda, el carro es buida i les existències de tots els productes comprats son actualitzades.

Descripció: El client pitjarà el botó Comprar i si el sistema verifica que el carro conté almenys una línia, llavors es realitzarà la comanda i finalment es buidarà el carro.

### **Cas d'ús "AfegirCollita"**

Resum funcionalitat general: Els treballadors podran afegir tantes collites com facin.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels treballadors.

Actors: Treballador.

Casos d'ús relacionats: Autenticar-se, ActualitzarStockProducte.

Precondició: S'ha d'estar registrat e identificat al sistema com a treballador.

Postcondició: La collita es afegida al sistema i la quantitat del producte collit serà actualitzada.

Descripció: El treballador pitjarà el botó afegir collita i introduirà les dades de la collita, si les dades son correctes, el sistema afegirà la collita i actualitzarà les existències del producte.

### **Cas d'ús "AfegirTreballador"**

Resum funcionalitat general: L'administrador podrà donar d'alta als treballadors dins la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se.

Precondició: L'administrador ha d'estar registrat e identificat.

Postcondició: El treballador queda donat d'alta al sistema.

Descripció: L'administrador accedeix al Gestor de treballadors i entra a afegir treballador. Emplenarà les dades, el sistema comprovarà les dades i les enregistrarà al sistema si son correctes, sinó enviarà un missatge indicant el problema.

### **Cas d'ús "ConsultarTreballador"**

Resum funcionalitat general: L'administrador podrà consultar les dades dels treballadors.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se.

Precondició: L'administrador ha d'estar registrat e identificat.

Postcondició: Es visualitzarà un treballador, alguns treballadors o tots els treballadors segons el criteri de cerca utilitzat.

Descripció: L'administrador accedeix al Gestor de treballadors i entra a consultar treballadors, aquí introduirà els criteris de la cerca i pitjarà el botó cercar, apareixeran els treballadors que compleixin els criteris indicats a la cerca.

### **Cas d'ús "ModificarTreballador"**

Resum funcionalitat general: L'administrador podrà modificar les dades d'un treballador.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se,  
ConsultarTreballador.

Precondició: El treballador s'ha de trobar en la consulta realitzada i l'administrador ha d'estar registrat e identificat.

Postcondició: El treballador queda modificat al sistema amb les noves dades.

Descripció: L'administrador accedeix al Gestor de treballadors i entra a consultar treballadors, un cop ha trobat el treballador que vol modificar, seleccionarà modificar treballador, modificarà les dades que necessiti i guardarà els canvis, el sistema comprovarà les dades i les guardarà al sistema si son correctes, sinó enviarà un missatge indicant el problema.

### **Cas d'ús "EliminarTreballador"**

Resum funcionalitat general: L'administrador podrà donar de baixa als treballadors dins la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se,  
ConsultarTreballador.

Precondició: El treballador s'ha de trobar en la consulta realitzada i l'administrador ha d'estar registrat e identificat.

Postcondició: El treballador queda eliminat del sistema.

Descripció: L'administrador accedeix al Gestor de treballadors i entra a consultar treballadors, un cop ha

trobat el treballador que vol eliminar, seleccionarà eliminar treballador i el treballador quedarà eliminat.

#### **Cas d'ús "AfegirProducte"**

Resum funcionalitat general: L'administrador podrà donar d'alta els productes dins la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se.

Precondició: L'administrador ha d'estar registrat e identificat.

Postcondició: El producte queda donat d'alta al sistema.

Descripció: L'administrador accedeix al Gestor de productes i entra a afegir producte. Emplenarà les dades, el sistema comprovarà les dades i les enregistrarà al sistema si son correctes, sinó enviarà un missatge indicant el problema.

#### **Cas d'ús "ConsultarProducte"**

Resum funcionalitat general: L'administrador podrà consultar les dades dels productes.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se.

Precondició: L'administrador ha d'estar registrat e identificat.

Postcondició: Es visualitzarà un producte, alguns productes o tots els productes segons el criteri de cerca utilitzat.

Descripció: L'administrador accedeix al Gestor de productes i entra a consultar productes, aquí introduirà els criteris de la cerca i pitjarà el botó cercar, apareixeran els productes que compleixin els criteris indicats a la cerca.

#### **Cas d'ús "ModificarProducte"**

Resum funcionalitat general: L'administrador podrà modificar les dades d'un producte.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se, ConsultarProducte.

Precondició: El producte s'ha de trobar en la consulta realitzada i l'administrador ha d'estar registrat e identificat.

Postcondició: El producte queda modificat al sistema amb les noves dades.

Descripció: L'administrador accedeix al Gestor de productes i entra a consultar productes, un cop ha trobat el producte que vol modificar, seleccionarà modificar producte, modificarà les dades que necessiti i guardarà els canvis, el sistema comprovarà les dades i les guardarà al sistema

si son correctes, sinó enviarà un missatge indicant el problema.

#### **Cas d'ús "EliminarProducte"**

Resum funcionalitat general: L'administrador podrà donar de baixa els productes dins la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se, ConsultarProducte.

Precondició: El producte s'ha de trobar en la consulta realitzada i l'administrador ha d'estar registrat e identificat.

Postcondició: El producte queda eliminat del sistema.

Descripció: L'administrador accedeix al Gestor de productes i entra a consultar producte, un cop ha trobat el producte que vol eliminar, seleccionarà eliminar producte i el producte quedarà eliminat.

#### **Cas d'ús "ConsultarClient"**

Resum funcionalitat general: L'administrador podrà consultar les dades dels clients.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se.

Precondició: L'administrador ha d'estar registrat e identificat.

Postcondició: Es visualitzarà un client, alguns clients o tots els clients segons el criteri de cerca utilitzat.

Descripció: L'administrador accedeix al Gestor de clients i entra a consultar clients, aquí introduirà els criteris de la cerca i pitjarà el botó cercar, apareixeran els clients que compleixin els criteris indicats a la cerca.

#### **Cas d'ús "EliminarClient"**

Resum funcionalitat general: L'administrador podrà donar de baixa als clients dins la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administrador.

Actors: Administrador.

Casos d'ús relacionats: Autenticar-se, ConsultarClient.

Precondició: El client s'ha de trobar en la consulta realitzada i l'administrador ha d'estar registrat e identificat.

Postcondició: El client queda eliminat del sistema.

Descripció: L'administrador accedeix al Gestor de clients i entra a consultar client, un cop ha trobat el client que vol eliminar, seleccionarà eliminar client i el client quedarà eliminat.

### 3. Capítol 3: Disseny

Amb les dades obtingudes en l'anàlisi, ja estem en disposició de començar el disseny de l'aplicació.

Per començar trobarem el diagrama de classes on trobarem totes les classes necessàries per l'aplicació amb els seus atributs i les relacions que hi ha entre elles. Com podem veure no s'han introduït els mètodes setters i getters per qüestions d'espai, però tots els atributs de totes les classes tindran els getters i setters.

Els diagrames de seqüència ens permetran entendre més exactament el funcionament que ha de tenir l'aplicació.

Després trobarem el diagrama estàtic per tal de poder definir més tard les taules necessàries de la base de dades. Veurem que aquí s'han canviat les associacions per atributs dins de les classes corresponents.

A partir del diagrama estàtic s'ha definit totes les taules i s'han creat els scripts necessaris per a la creació de les taules.

#### 3.1. Diagrama de classes

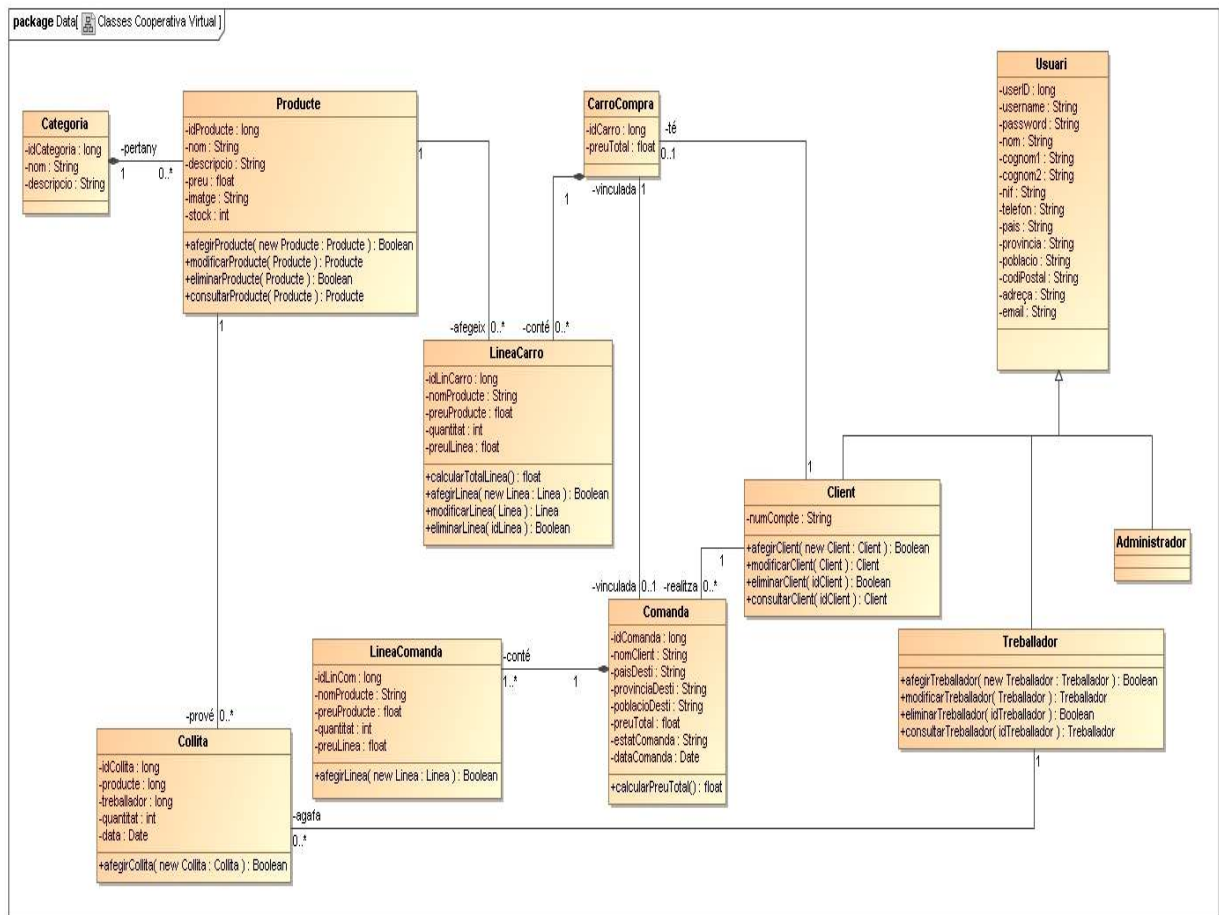


Figura 4. Diagrama de classes

### 3.2. Diagrames de seqüència

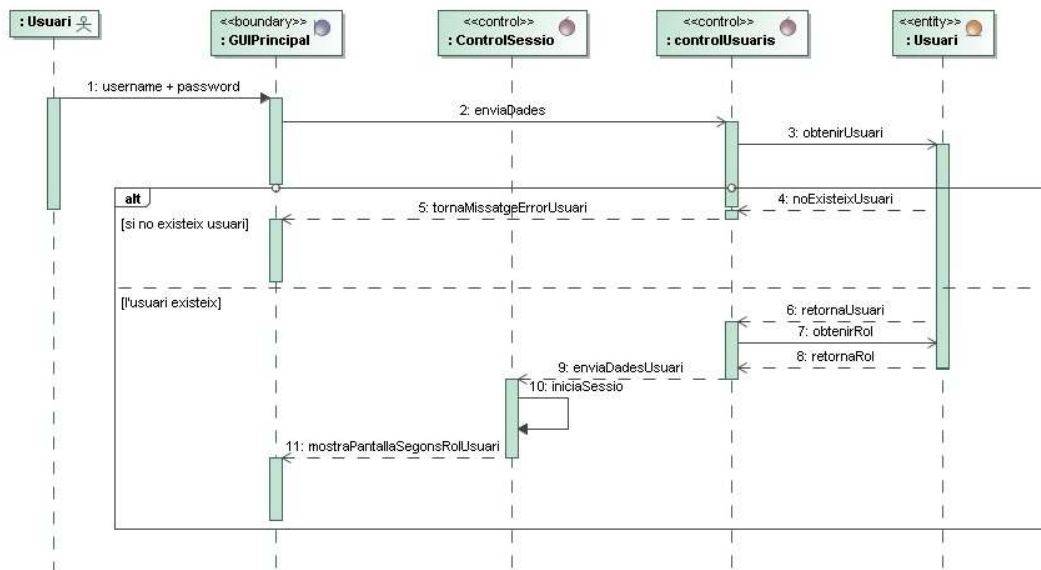


Figura 5. Diagrama de seqüència d'autenticació d'usuaris

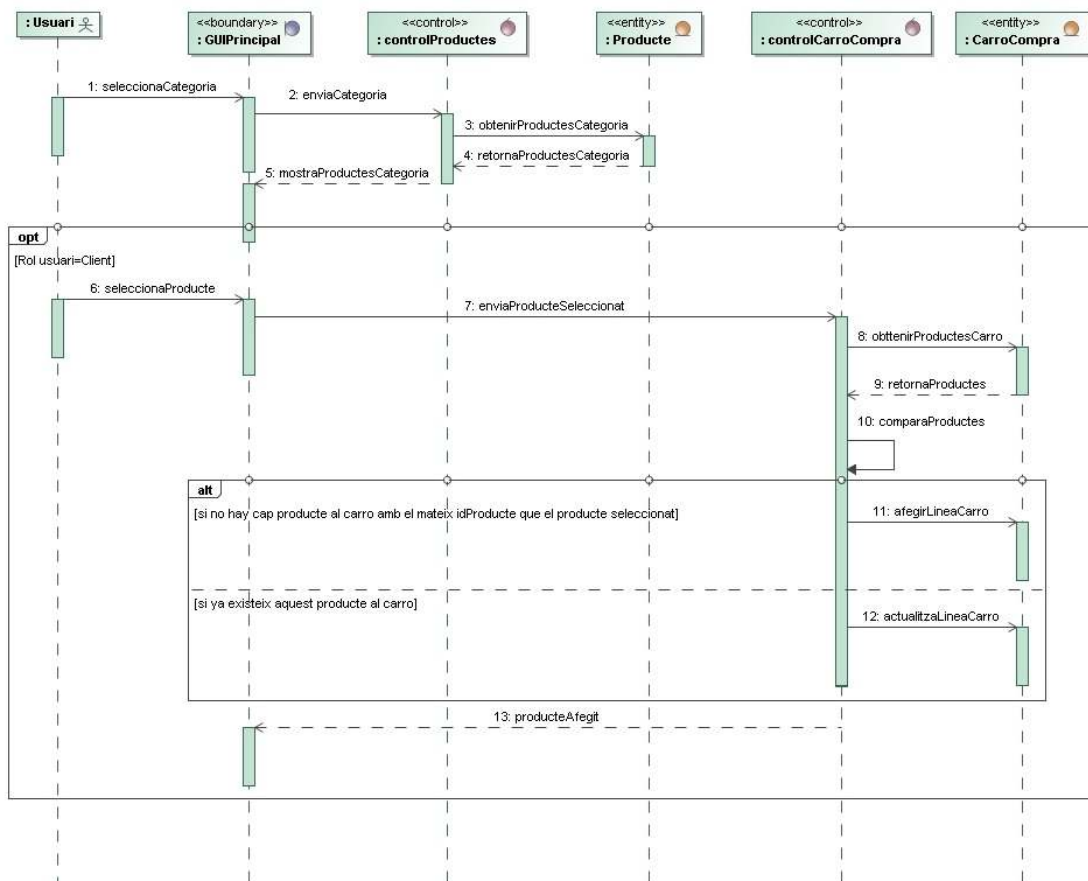


Figura 6. Diagrama de seqüència consultar productes per categoria

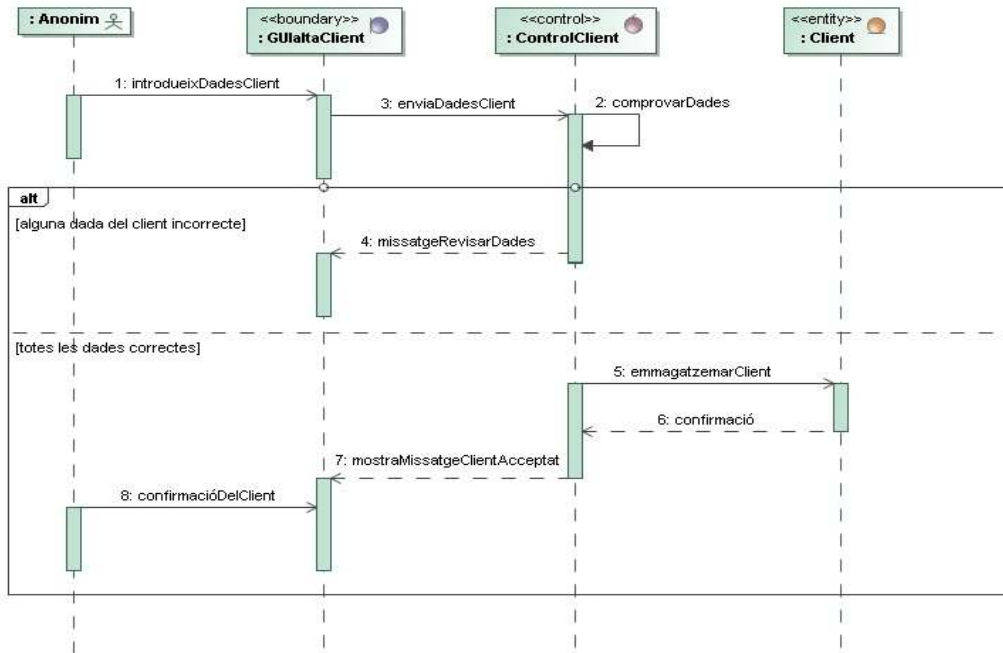


Figura 7. Diagrama de l'alta d'un client (Enregistrament)

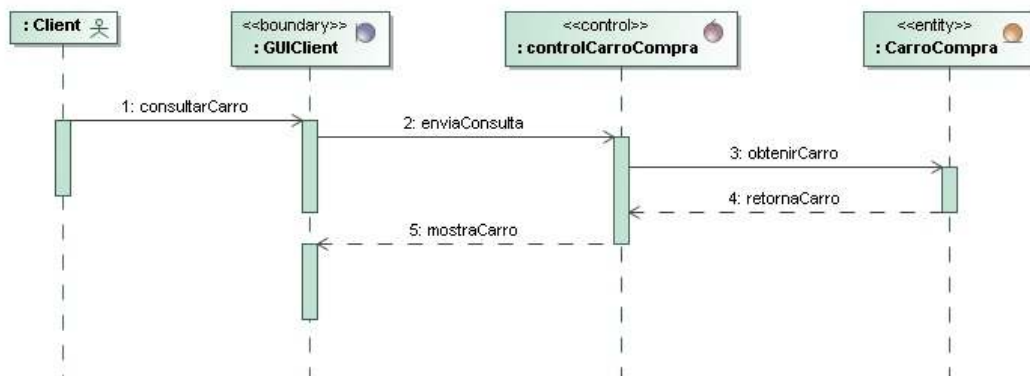


Figura 8. Diagrama de seqüència consultar carro compra

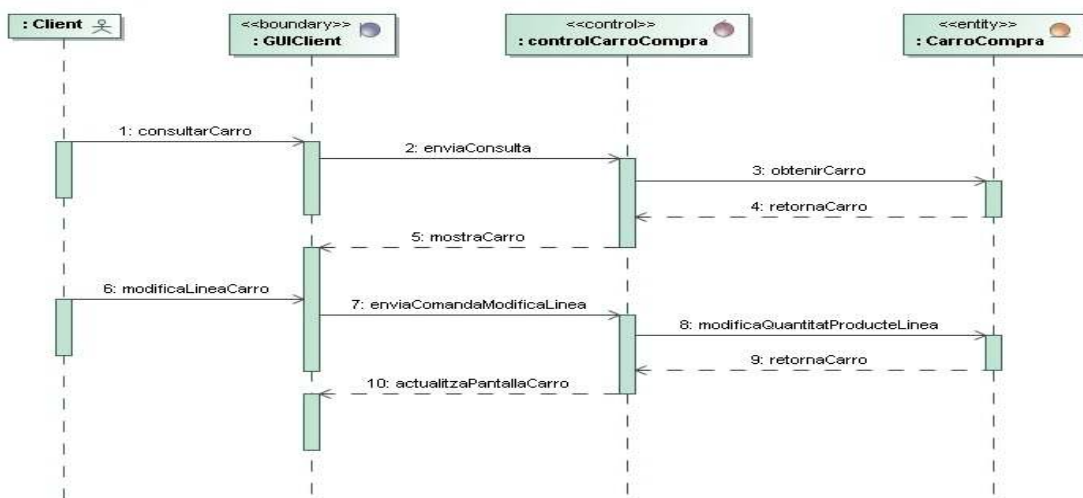


Figura 9. Diagrama de seqüència modificació d'una línia del carro

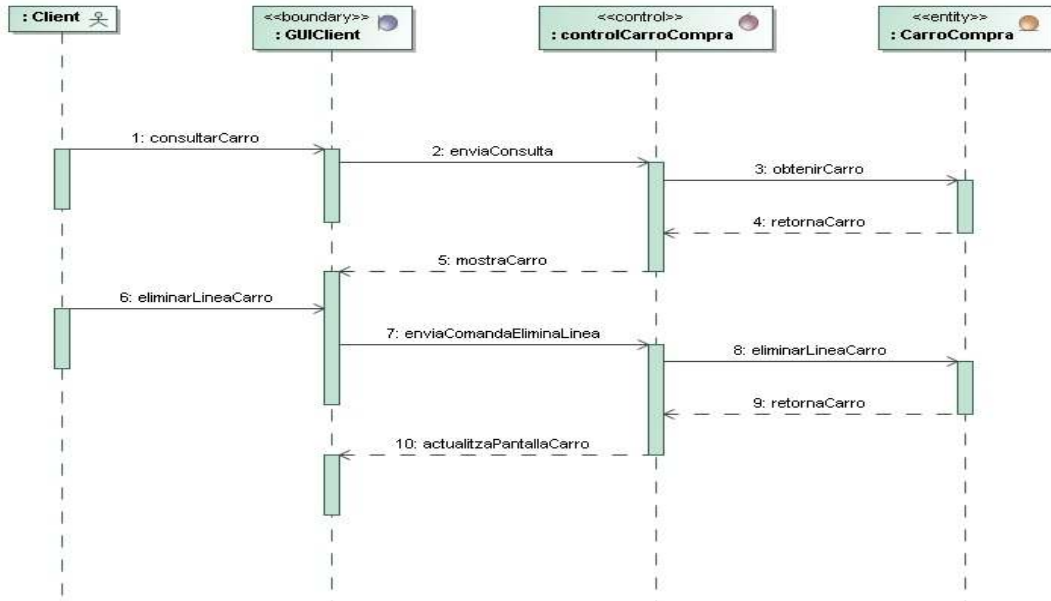


Figura 10. Diagrama de seqüència eliminació d'una línia del carro

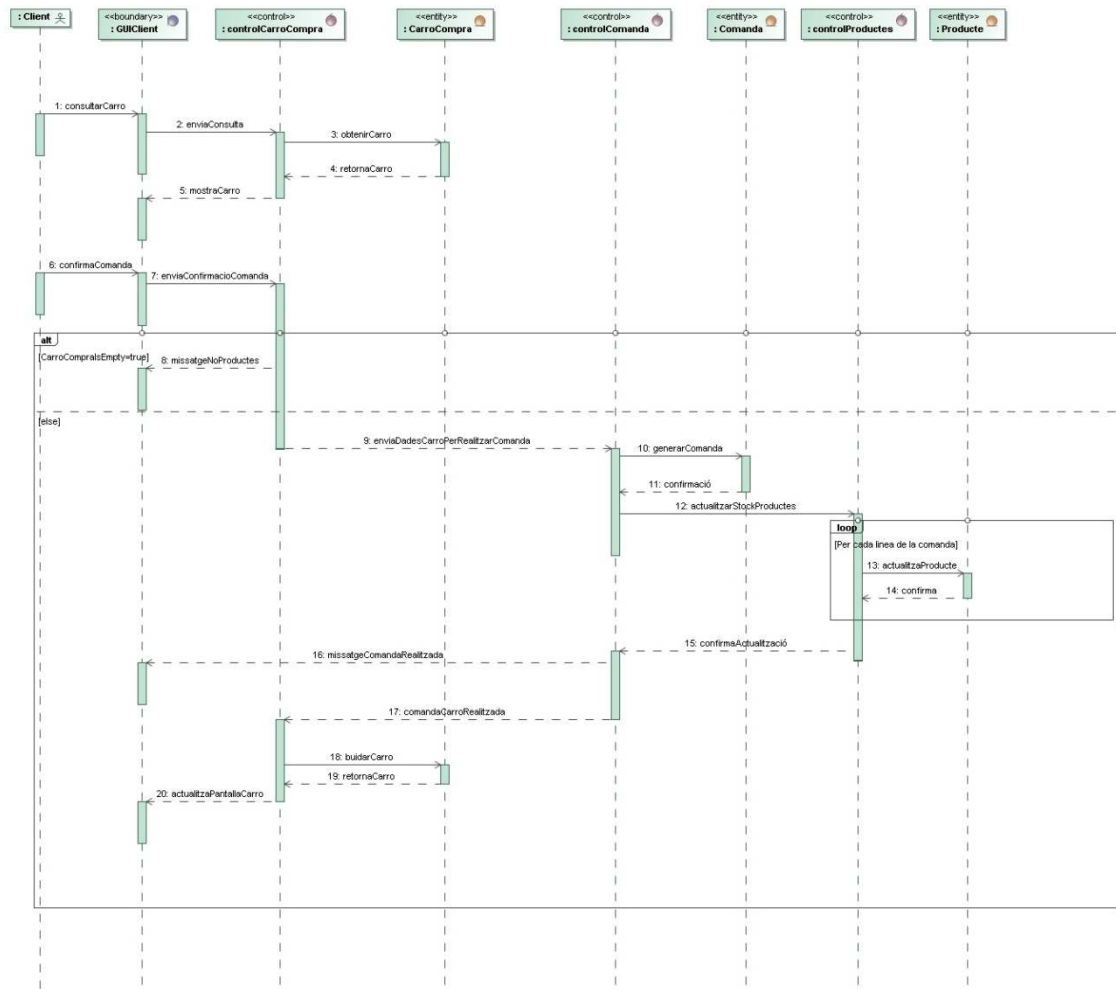


Figura 11. Diagrama de seqüència realitza comanda carro



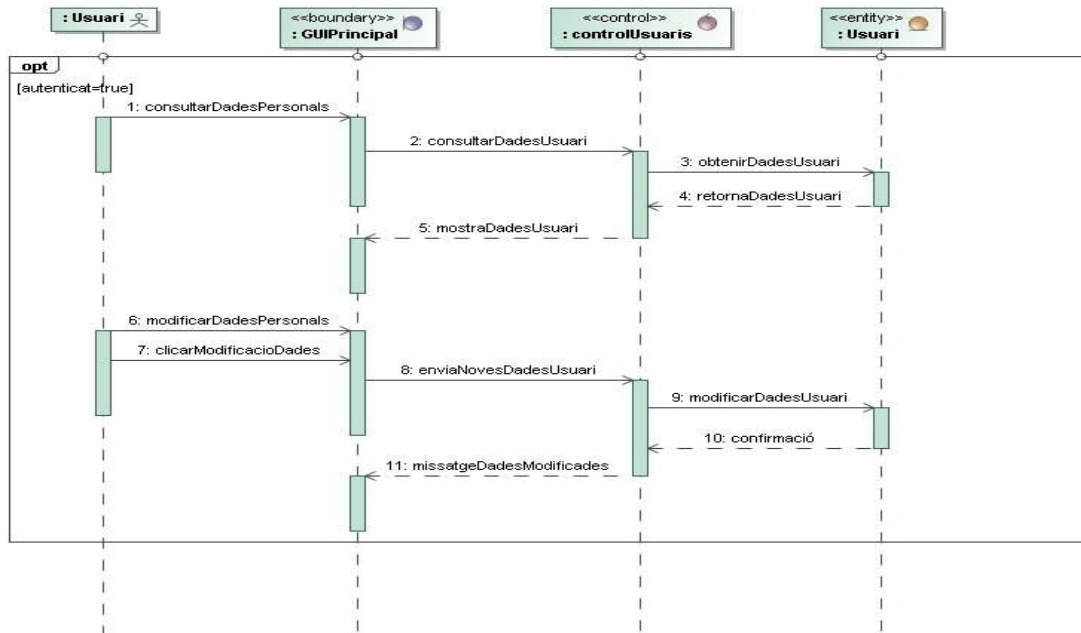


Figura 12. Diagrama de seqüència consulta i modificació dades personals

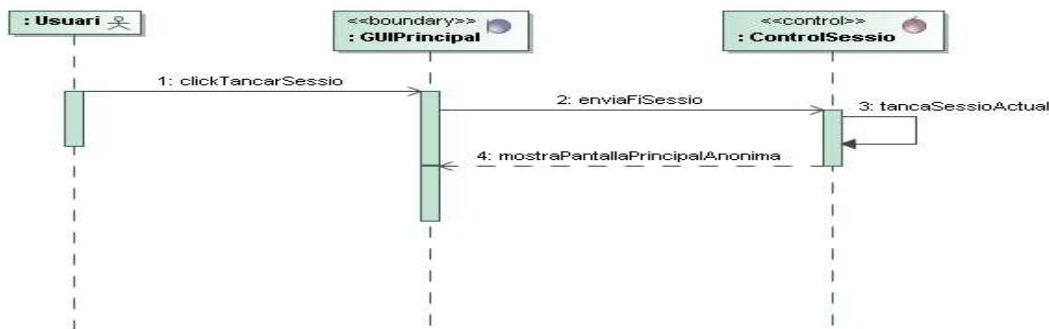


Figura 13. Diagrama de seqüència tancament de sessió

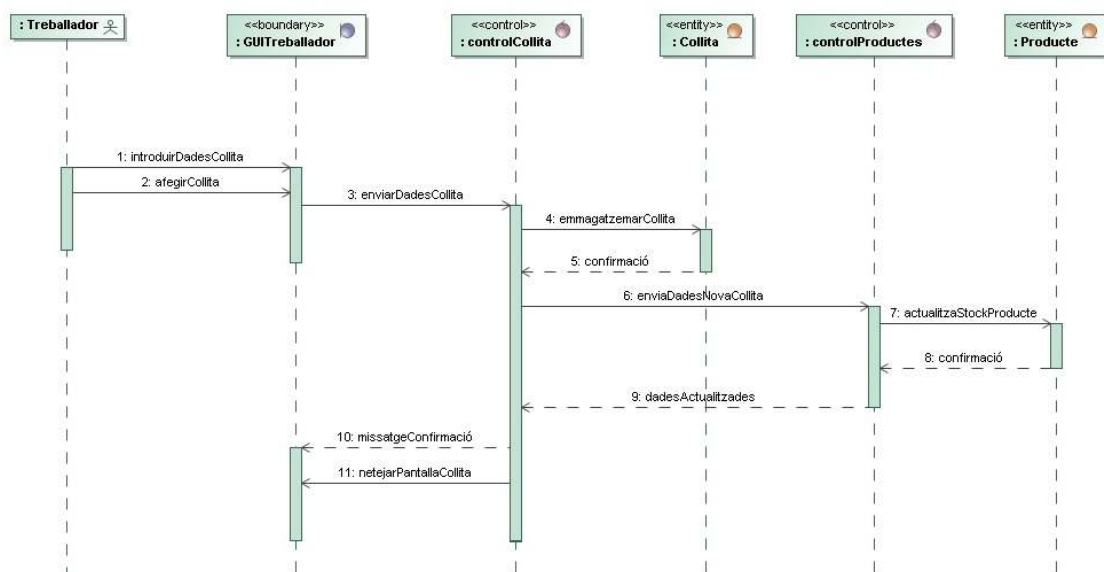


Figura 14. Diagrama de seqüència afegir collita

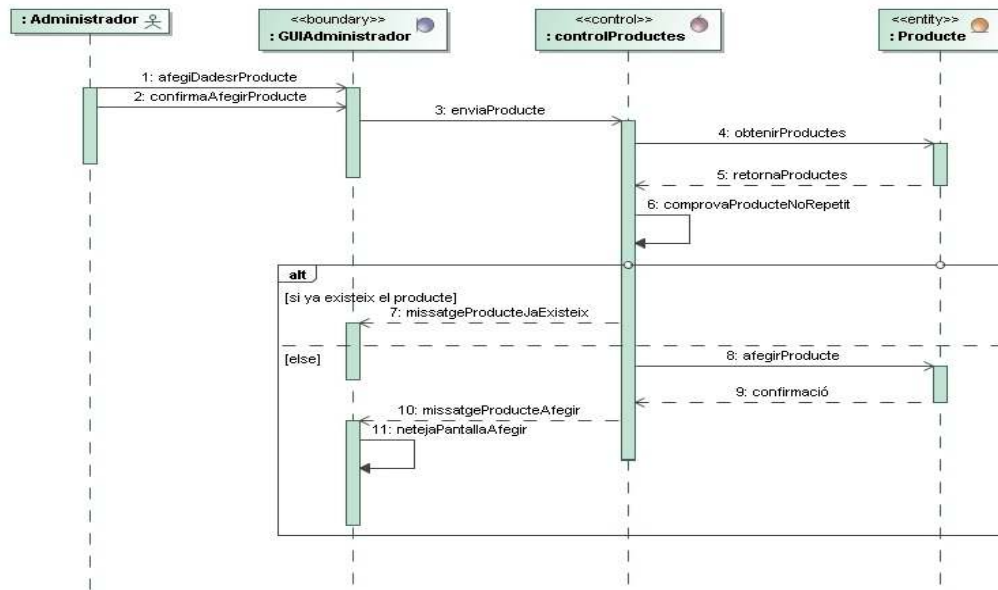


Figura 15. Diagrama de seqüència afegir producte

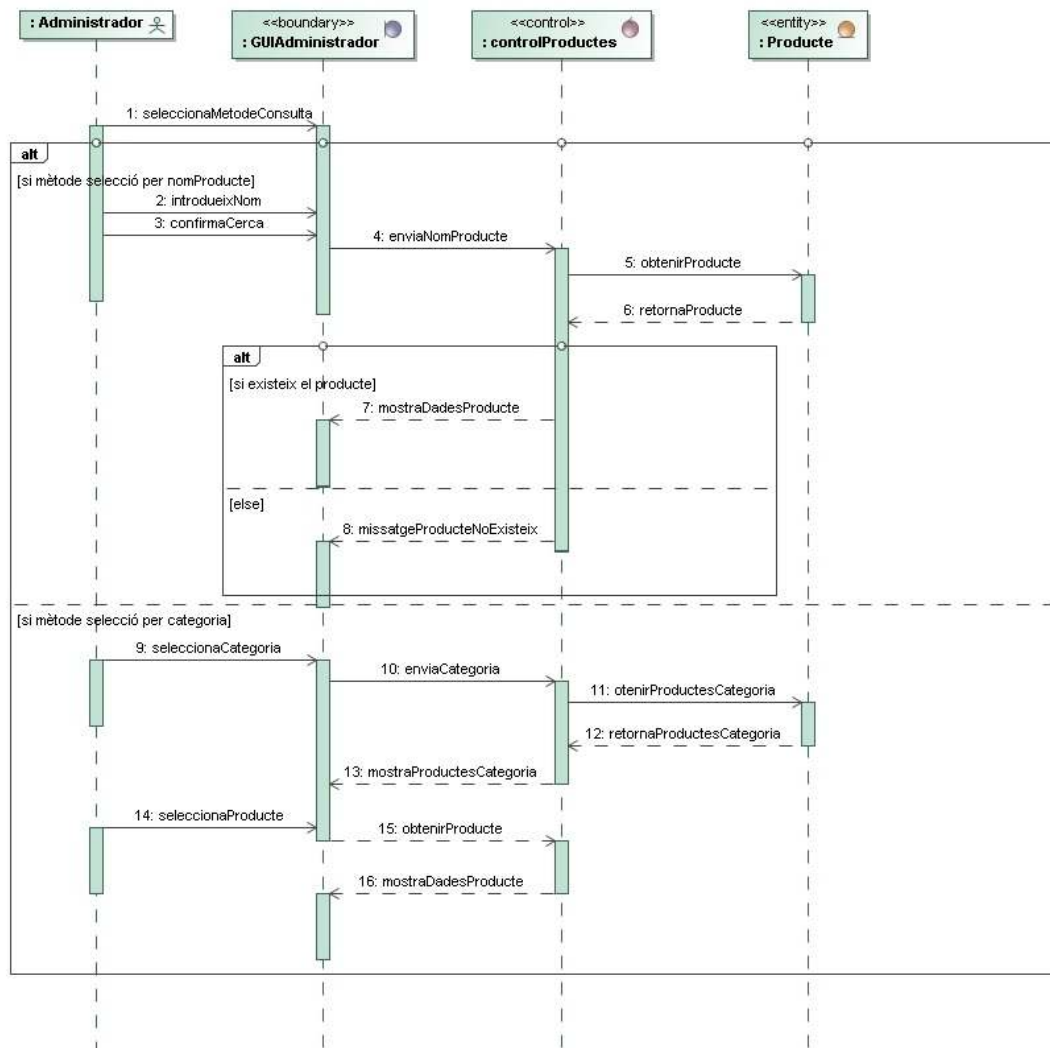


Figura 16. Diagrama de seqüència consultar producte

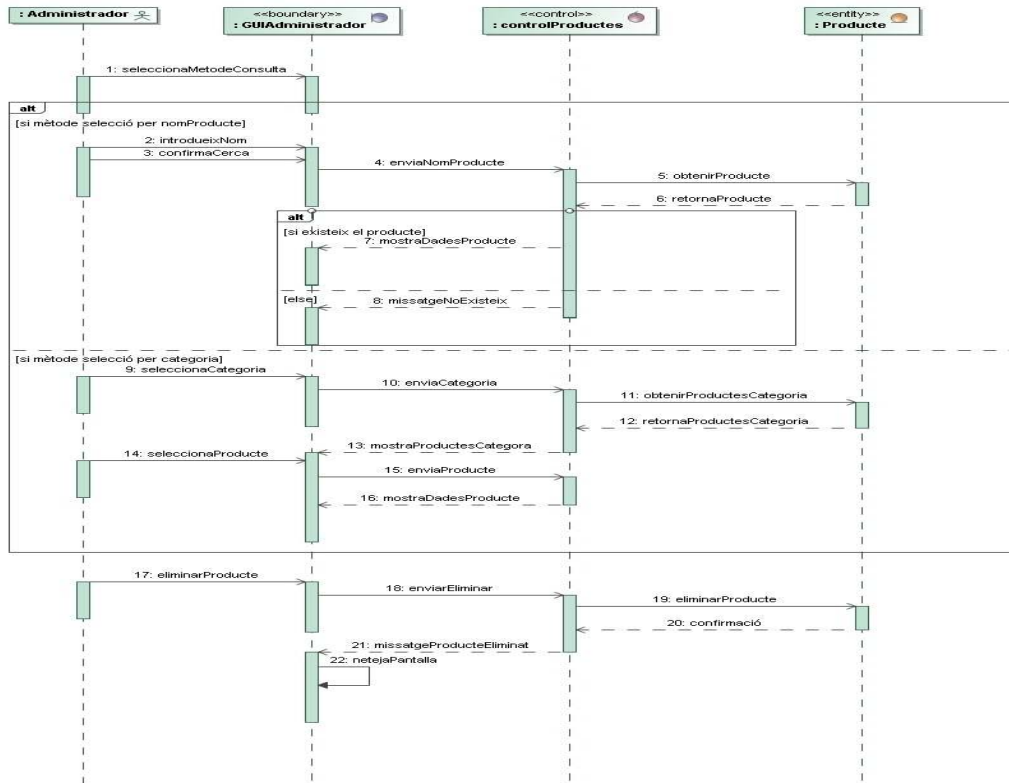


Figura 17. Diagrama de seqüència eliminar producte

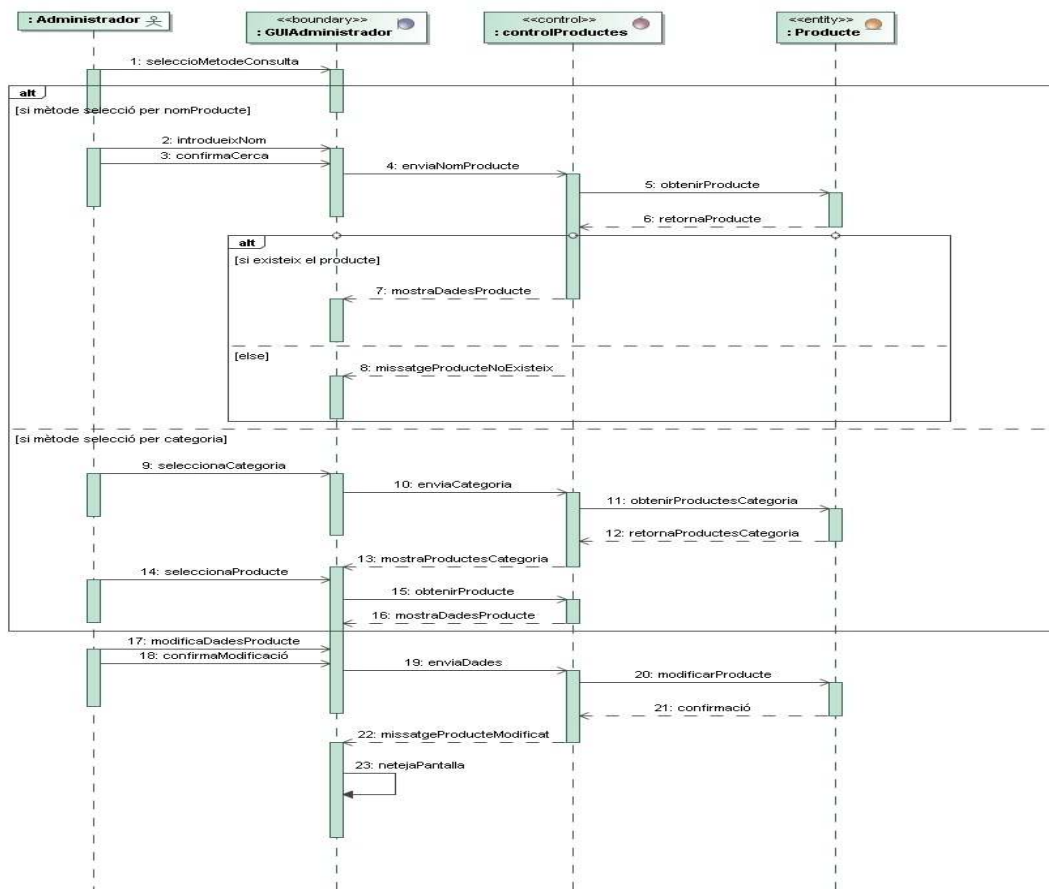


Figura 18. Diagrama de seqüència modificar producte

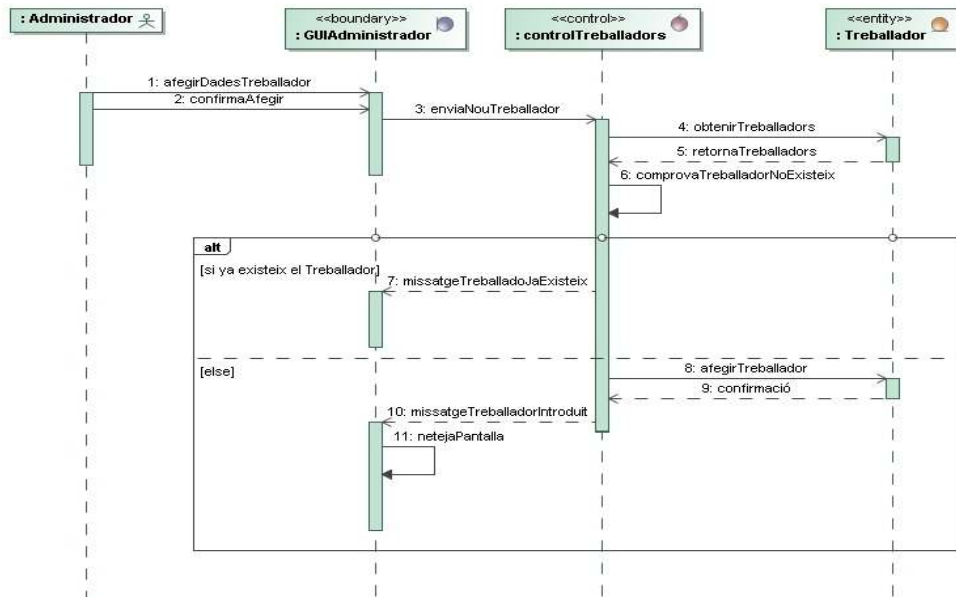


Figura 19. Diagrama de seqüència alta treballador

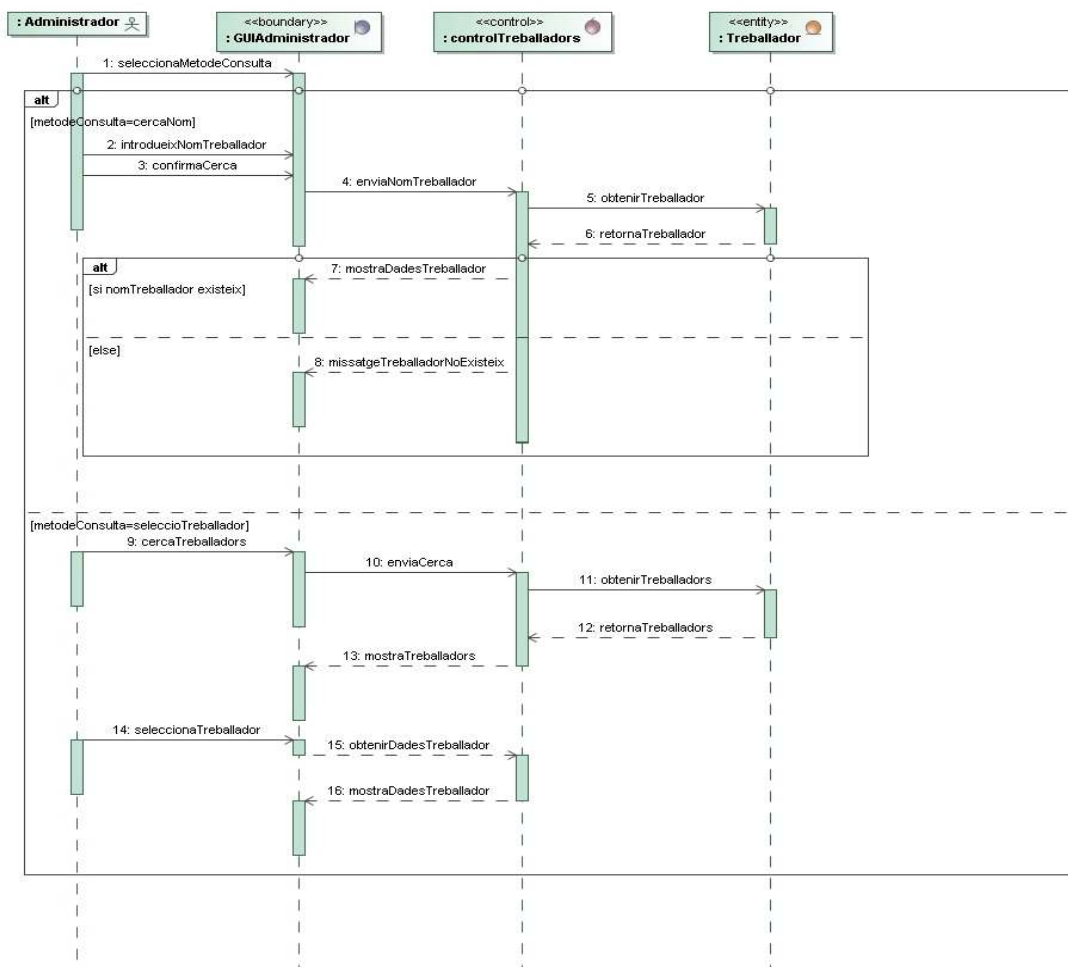


Figura 20. Diagrama de seqüència consultar treballador

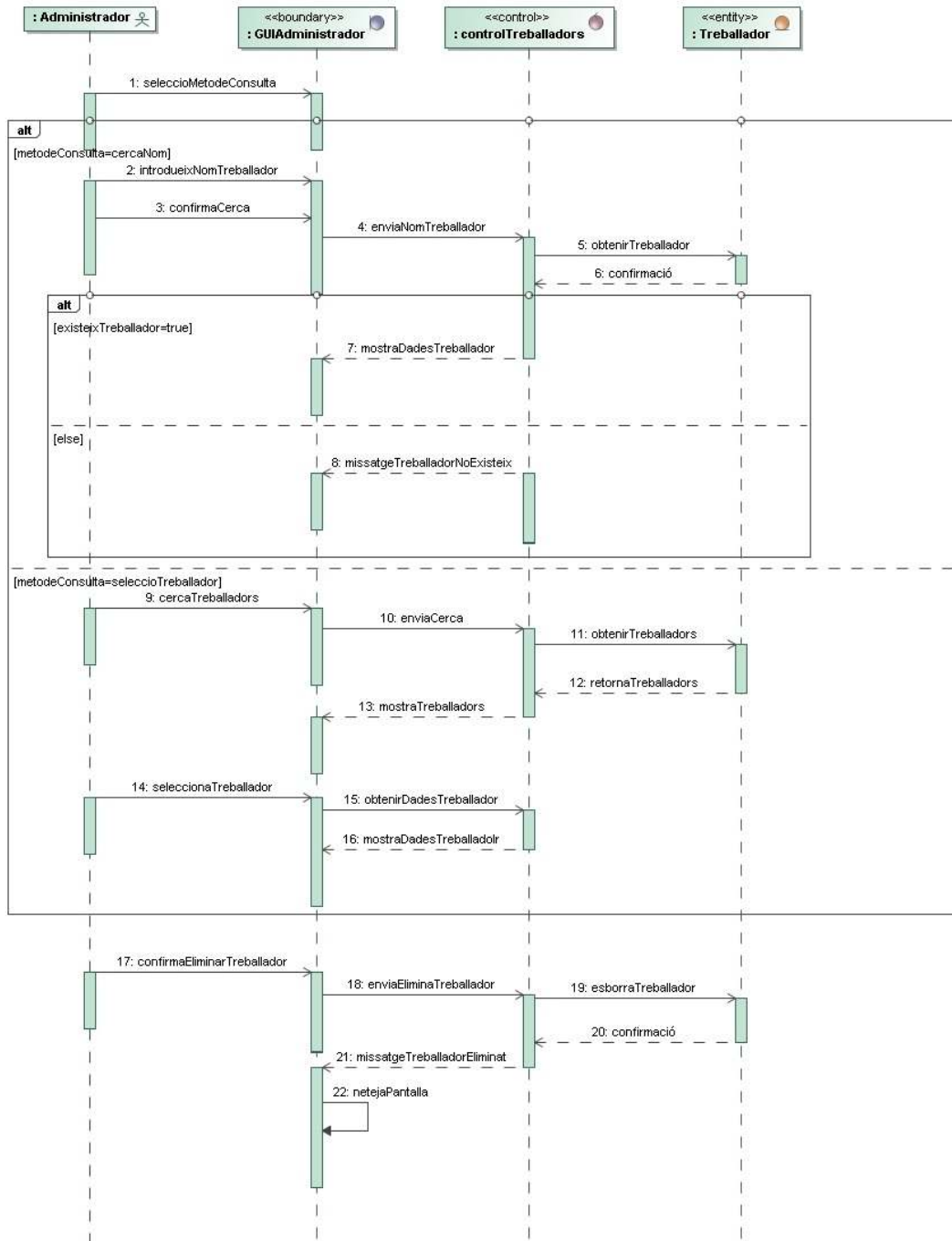


Figura 21. Diagrama de seqüència baixa treballador

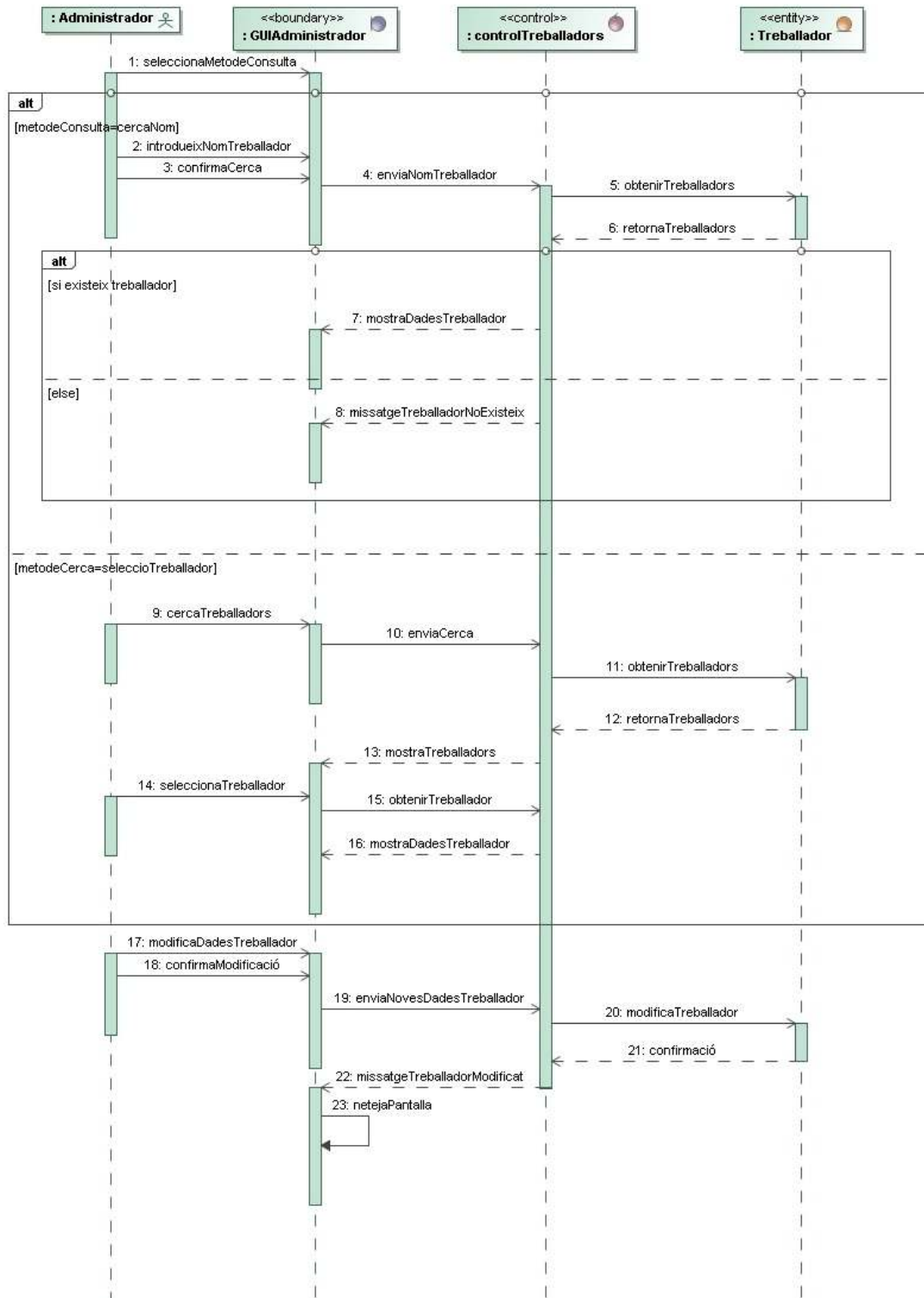


Figura 22. Diagrama de seqüència modificació dades d'un treballador

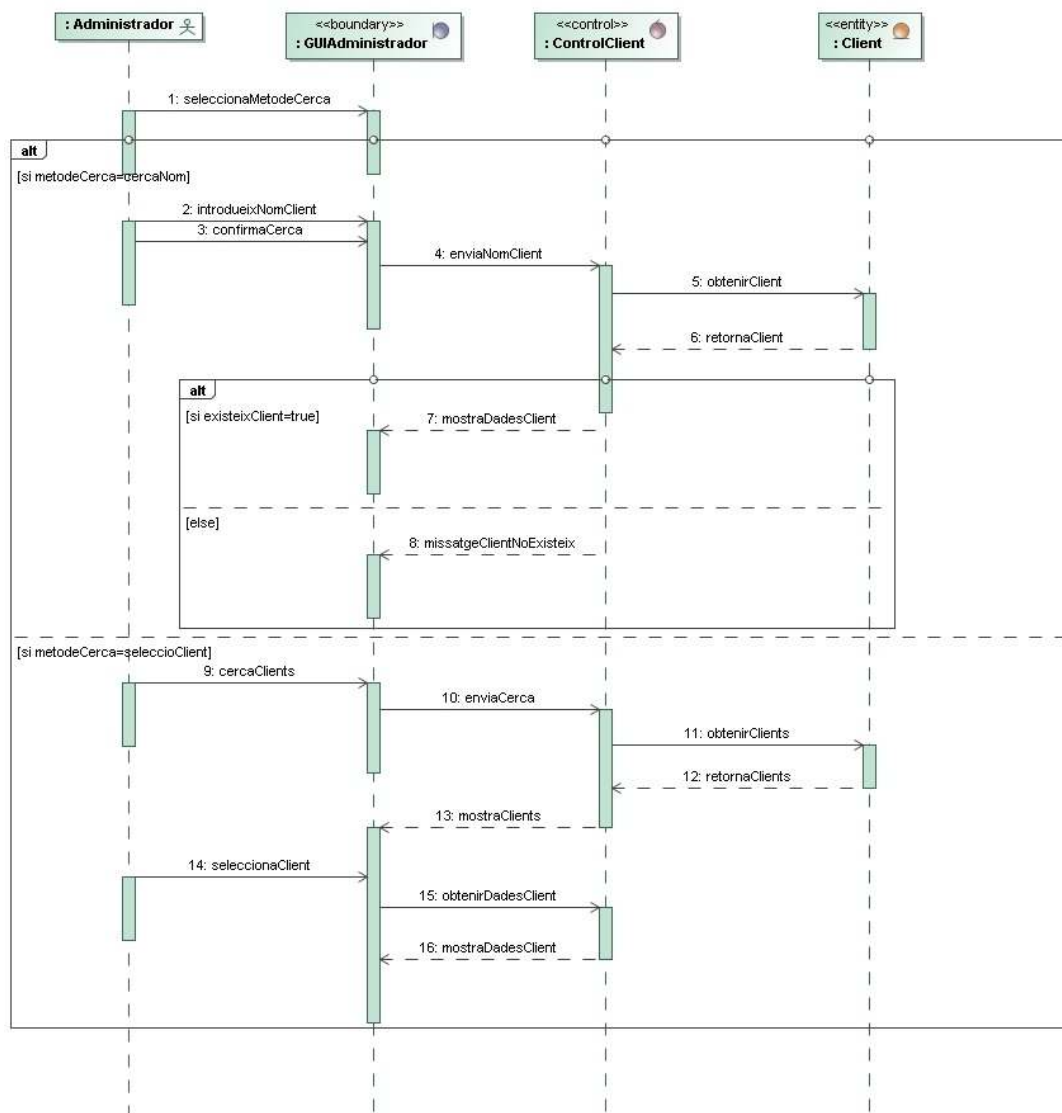


Figura 23. Diagrama de seqüència consultar client

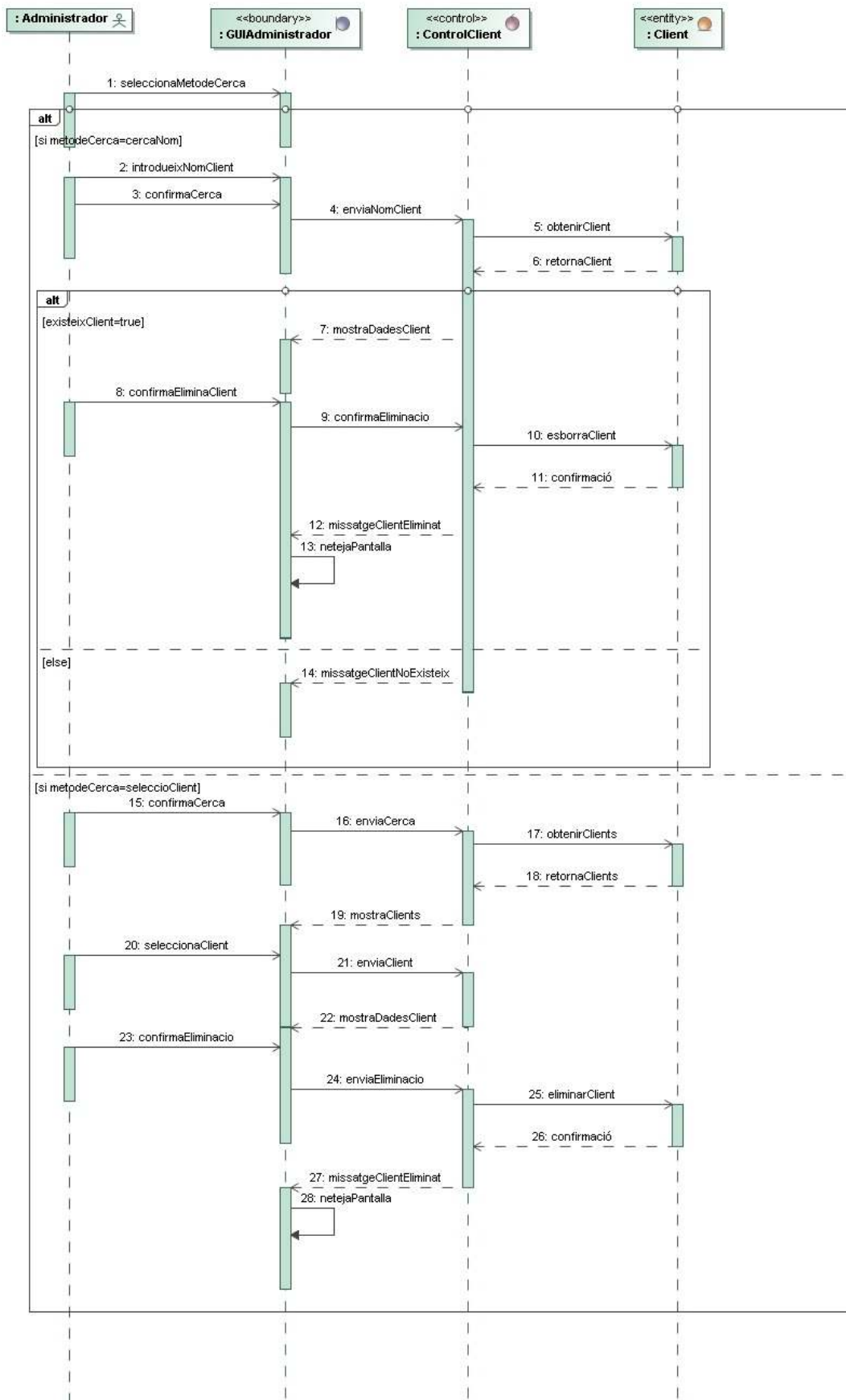


Figura 24. Diagrama de seqüència baixa d'un client



### 3.3. Diagrama d'estats

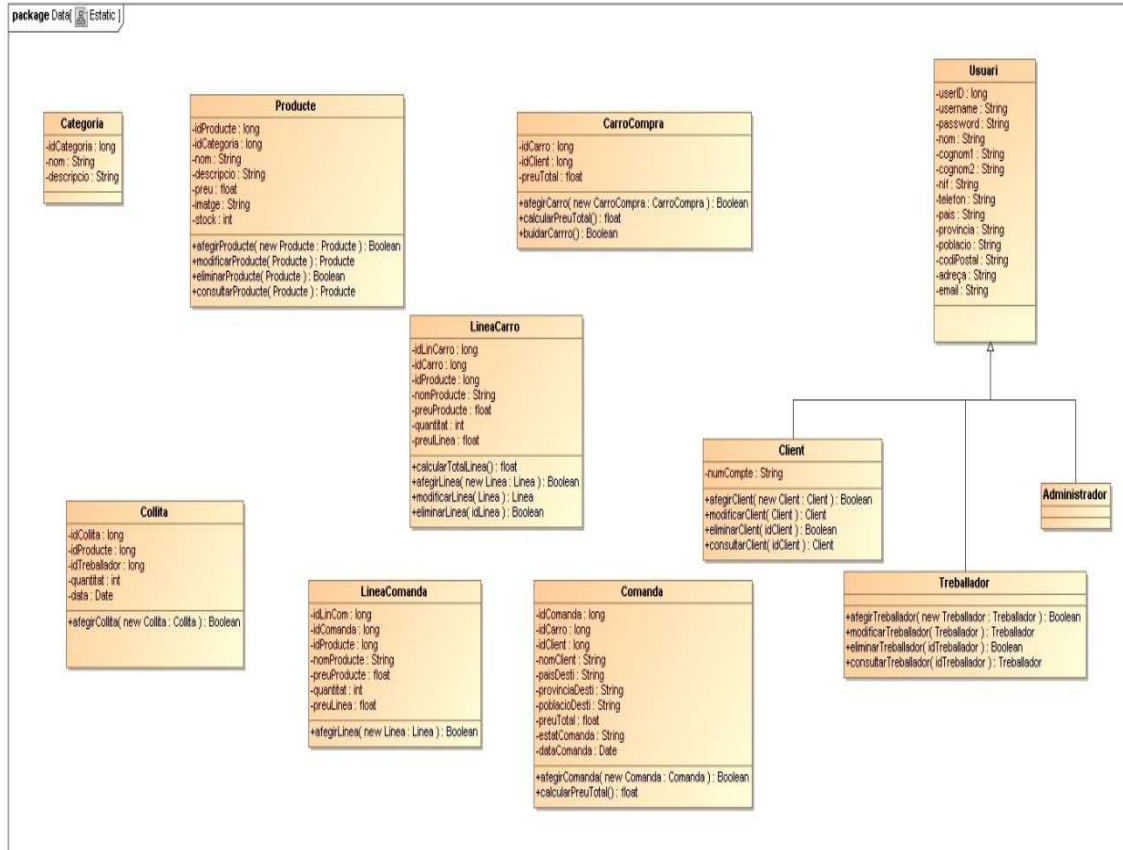


Figura 25. Diagrama d'estats

### 3.4. Disseny Base de dades

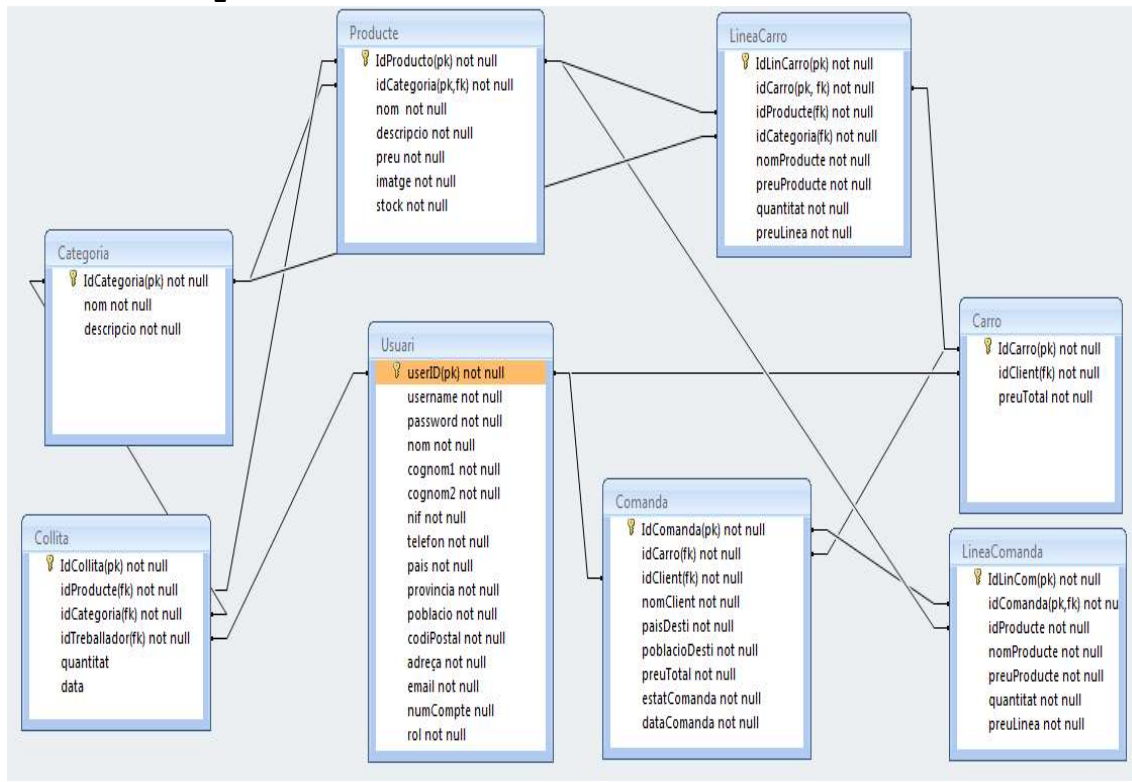


Figura 26. Disseny base de dades

Script per a la creació de les taules

```
CREATE TABLE CATEGORIA(  
  IdCategoria NUMBER CONSTRAINT PK_CATEGORIA PRIMARY KEY,  
  nom VARCHAR2(20 CHAR) CONSTRAINT NN_Categoria_nom NOT NULL  
    CONSTRAINT UN_Categoria UNIQUE,  
  descripcio VARCHAR2(100 CHAR) DEFAULT 'Sense descripció'  
);  
  
CREATE TABLE PRODUCTE(  
  IdProducte NUMBER CONSTRAINT PK_PRODUCTE PRIMARY KEY,  
  IdCategoria CONSTRAINT FK_PRODUCTE REFERENCES CATEGORIA(IdCategoria),  
  nom VARCHAR2(20 CHAR) CONSTRAINT NN_Producte_nom NOT NULL  
    CONSTRAINT UN_Producte UNIQUE,  
  descripcio VARCHAR2(100 CHAR) DEFAULT 'Sense descripció',  
  preu NUMBER(11,2) CONSTRAINT NN_Producte_preu NOT NULL,  
  imatge VARCHAR2(50 CHAR) CONSTRAINT NN_Producte_imatge NOT NULL,  
  stock NUMBER(11,2) DEFAULT 0  
);  
  
CREATE TABLE USUARI(  
  userID NUMBER CONSTRAINT PK_USUARI PRIMARY KEY,  
  userName VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_userName NOT NULL  
    CONSTRAINT UN_Usuari UNIQUE,  
  pass VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_password NOT NULL,  
  nom VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_nom NOT NULL,  
  cognom1 VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_cognom1 NOT NULL,  
  cognom2 VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_cognom2 NOT NULL,  
  nif CHAR(9) CONSTRAINT NN_Usuari_nif NOT NULL,  
  telefon VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_telefon NOT NULL,  
  pais VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_pais NOT NULL,  
  provincia VARCHAR2(20 CHAR) CONSTRAINT NN_Usuari_provincia NOT NULL,  
  poblacio VARCHAR2(30 CHAR) CONSTRAINT NN_Usuari_poblacio NOT NULL,  
  codiPostal VARCHAR2(10 CHAR) CONSTRAINT NN_Usuari_codiPostal NOT NULL,  
  adreca VARCHAR2(50 CHAR) CONSTRAINT NN_Usuari_userAdreca NOT NULL,  
  email VARCHAR2(30 CHAR),  
  numCompte VARCHAR2(25 CHAR),  
  rol CHAR(1) CONSTRAINT NN_Usuari_rol NOT NULL,  
  type VARCHAR2(1 CHAR)  
);
```

```
CREATE TABLE CARRO (
  IdCarro NUMBER CONSTRAINT PK_CARRO PRIMARY KEY,
  IdClient CONSTRAINT FK_CARRO REFERENCES USUARI(userID),
  preuTotal NUMBER(11,2) DEFAULT 0
);

CREATE TABLE COMANDA (
  IdComanda NUMBER CONSTRAINT PK_COMANDA PRIMARY KEY,
  IdCarro CONSTRAINT FK1_COMANDA REFERENCES CARRO(IdCarro),
  IdClient CONSTRAINT FK2_COMANDA REFERENCES USUARI(userID),
  nomClient VARCHAR2(20 CHAR) CONSTRAINT NN_Comanda_nomClient NOT NULL,
  paisDesti VARCHAR2(20 CHAR) CONSTRAINT NN_Comanda_paisDesti NOT NULL,
  poblacioDesti VARCHAR2(20 CHAR) CONSTRAINT NN_Comanda_poblacioDesti
  NOT NULL,
  preuTotal NUMBER(11,2) CONSTRAINT NN_Comanda_preuTotal NOT NULL,
  estatComanda VARCHAR(20 CHAR) DEFAULT 'Rebuda',
  dataComanda DATE CONSTRAINT NN_Comanda_data NOT NULL
);

CREATE TABLE LINEACARRO (
  IdLinCarro NUMBER CONSTRAINT PK_LINEACARRO PRIMARY KEY,
  IdCarro CONSTRAINT FK1_LINEACARRO REFERENCES CARRO(IdCarro),
  IdProducte CONSTRAINT FK2_LINEACARRO REFERENCES PRODUCTE(IdProducte),
  IdCategoria          CONSTRAINT          FK3_LINEACARRO          REFERENCES
  CATEGORIA(IdCategoria),
  nomProducte VARCHAR2(20 CHAR) CONSTRAINT NN_Lineacarro_nomProd NOT
  NULL,
  preuProducte NUMBER(11,2) CONSTRAINT NN_Lineacarro_preuProd NOT NULL,
  quantitat NUMBER CONSTRAINT NN_Lineacarro_quantitat NOT NULL,
  preuLinea NUMBER(11,2) CONSTRAINT NN_Lineacarro_preuLinea NOT NULL
);

CREATE TABLE LINEACOMANDA (
  IdLinCom NUMBER CONSTRAINT PK_LINEACOMANDA PRIMARY KEY,
  IdComanda CONSTRAINT FK_LINEACOMANDA REFERENCES COMANDA(IdComanda),
  idProducte NUMBER CONSTRAINT NN_Lineacomanda_idProducte NOT NULL,
  nomProducte VARCHAR2(20 CHAR) CONSTRAINT NN_Lineacomanda_nomProducte
  NOT NULL,
  preuProducte NUMBER(11,2) CONSTRAINT NN_Lineacomanda_preuProducte NOT
  NULL,
  quantitat NUMBER CONSTRAINT NN_Lineacomanda_quantitat NOT NULL,
  preuLinea NUMBER(11,2) CONSTRAINT NN_Lineacomanda_preuLinea NOT NULL
```

```

);

CREATE TABLE COLLITA (
  IdCollita NUMBER CONSTRAINT PK_COLLITA PRIMARY KEY,
  IdProducte CONSTRAINT FK1_COLLITA REFERENCES PRODUCTE(IdProducte),
  IdCategoria CONSTRAINT FK2_COLLITA REFERENCES CATEGORIA(IdCategoria),
  IdTreballador CONSTRAINT FK3_COLLITA REFERENCES USUARI(userId),
  quantitat NUMBER CONSTRAINT NN_collita_quantitat NOT NULL,
  data DATE CONSTRAINT NN_collita_data NOT NULL
);

```

### 3.5. Arquitectura del sistema

L'arquitectura del sistema es J2EE. Java EE es una plataforma de programació que parteix de la plataforma Java. J2EE esta definida en un especificació, que consisteix en un conjunt de serveis, API i protocols per al desenvolupament d'aplicacions de múltiples capes, que es recolza en components de software modulars.

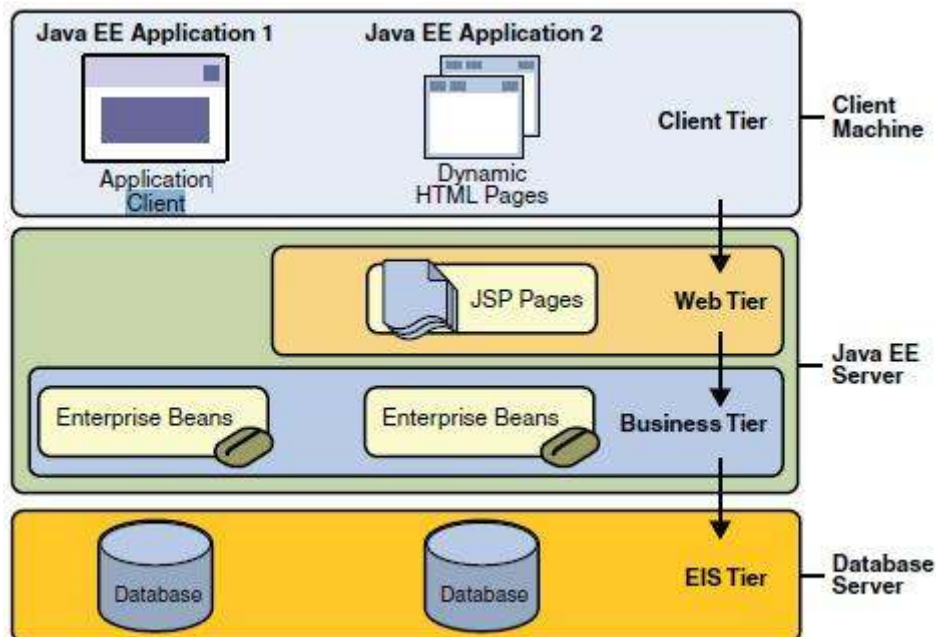


Figura 27. Arquitectura J2EE

J2EE permet el desenvolupament d'aplicacions d'empresa d'una manera senzilla i eficient, redueix el cost i complexitat en el desenvolupament d'aplicacions de múltiples capes. Al dividir l'aplicació en n capes, obtenim una aplicació més escalable i mantenible. Una altre avantatge es la seva portabilitat, ja que al ser una aplicació desenvolupada en Java pot funcionar en qualsevol plataforma que tingui instal·lat Java Virtual Machine. Les capes d'una aplicació que segueix l'especificació J2EE tal com hem pogut veure a l'anterior gràfic, son la capa

client, la capa web, la capa de negoci i la capa del sistema d'informació.

Tota aplicació J2EE s'executa en un servidor d'aplicacions o contenidor, que es on tindrem instal·lats els components.

Cada capa del servidor té els seus components:

- Els components de la capa web son els servlets i JSP.
- Els components de la capa de negoci son la lògica de negoci, aquest components son els EJB(Enterprise Java Bean).

De contenidors hi ha de 2 tipus:

- Contenedors Web o contenidors de Servlet/JSP, que gestionen l'execució dels Servlets i pàgines JSP.
- Contenedors Enterprise Java Beans, que gestionen l'execució dels EJB.

Els contenidors inclouen descriptors de desplegament, que son arxius XML que serveixen per a configurar l'entorn d'execució(paràmetres d'inicialització, rutes d'accés, etc.).

Per tal de desenvolupar aplicacions J2EE existeix diferents frameworks, que s'adapten perfectament a la majoria d'IDEs. En els següents apartats explicarem els frameworks utilitzats dins del projecte.

### **3.5.1. Struts 2**

Struts 2 es la nova versió d'Struts, encara que existeixen grans diferències entre les versions. Això es degut a que Struts 2 esta basat en WebWork, un altre framework de desenvolupament web en Java. Podem dir que Struts 2 ha sigut creat a partir de WebWork però mantenint les avantatges que aportava Struts.

El framework ens permet reduir temps de desenvolupament d'aplicacions web. Aquest framework es de software lliure i es compatible amb qualsevol plataforma que tingui disponible Java.

Struts 2 té una alta capacitat de configuració i extensibilitat i permet l'ús de plugings de components e integració amb altres frameworks. Com a grans avantatges d'aquesta nova versió d'Struts podem remarcar:

- S'ha simplificat el disseny, ja que ara s'utilitzen Interfases en comptes de classes abstractes.
- Els actions son POJOs, per tant qualsevol classe Java amb un mètode execute pot actuar com a action.
- Es poden testejar unitàriament els actions.
- S'han reemplaçar els ActionForms per JavaBeans.

- S'han canviat els Action Forward per results, obtenint més flexibilitat.
- Han aparegut les anotacions.
- Té una fàcil integració amb Spring.
- Té suport per AJAX, poden fer aplicacions interactives d'una forma més simple.

Struts 2 es basa en el patró MVC.

En Struts2 el model es l'Action, la vista es el Result i el controlador es el FilterDispatcher.

El FilterDispatcher es defineix a web.xml i es l'encarregat de gestionar les accions i de seleccionar els interceptor a invocar.

La configuració principal de Struts2 esta a l'arxiu struts.xml. Aquest arxiu es pot segmentar en tants arxius com es necessiti per tenir ben estructurades les accions. Existeixen interceptors predefinits a Struts2 com login, logger, etc., però també podem definir de nous estenent de AbstractInterceptor i després declarar-los en struts.xml. Al següent gràfic podem veure com Struts2 implementa el patró MVC.

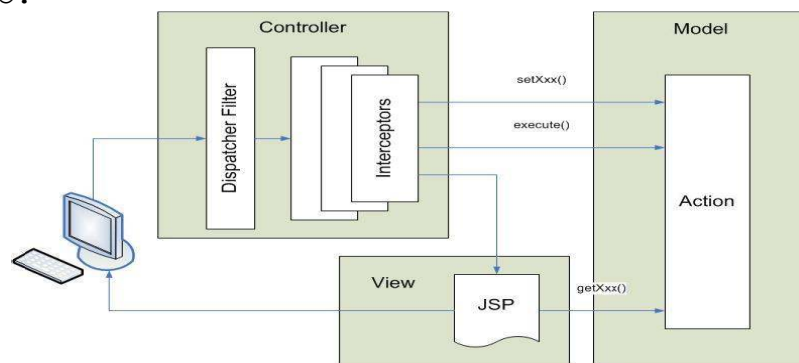


Figura 28. Model-Vista-Controlador

### 3.5.2. Hibernate

Es una eina de mapeig objecte-relacional(ORM) per a la plataforma Java.

Es a dir, la funció d'Hibernate es solucionar el problema que hi ha en les aplicacions en les que coexisteixen dades del model orientat a objectes(com pot ser Java) i dades relacionals(com son la majoria de SGBD comercialitzats). Aquesta eina facilita el mapeig entre atributs d'una BBDD relacional i el model orientat a objectes d'una aplicació. Aquest mapeig es pot fer mitjançant fitxers xml o mitjançant anotacions al propi codi de l'entitat. Aquests fitxer hbm.xml per al mapeig, el que fan es associar un POJO a la seva corresponent taula de la base de dades, en aquest fitxer també s'especifiquen les relacions entre els objectes, les restriccions de les dades(not null, unique),

la clau primària, si s'utilitzen seqüències per a generar la clau primària i el seu tipus, etc. Gràcies a tot això, es possible canviar de base de dades sense tenir que modificar el codi de l'aplicació, només es necessari modificar l'arxiu de configuració d'Hibernate, per tal d'indicar-li com a d'accedir a la base de dades i el nou driver a utilitzar per a la connexió. El fitxer de configuració d'Hibernate es hibernate.cfg.xml. Una característica molt important d'Hibernate genera les sentències SQL i d'aquesta manera els desenvolupadors es poden oblidar de la base de dades i dedicar-se exclusivament en l'aplicació. Hibernate també ofereix un llenguatge de consulta anomenat HQL que es molt semblant a l'SQL, però es tracta d'un llenguatge molt potent, orientat a objectes que compren nocions d'herència, polimorfisme i associació.

### 3.6. Patrons de disseny que s'utilitzaran

#### - Patró MVC

Per al desenvolupament d'aquest tipus d'aplicacions es molt recomanable l'ús del patró MVC.

El model-vista-controlador(MVC) es compon de 3 capes clarament diferenciades. Aquest patró proporciona una separació entre el model(dades aplicació), la vista(interfície gràfica usuari) i el controlador(lògica de control).

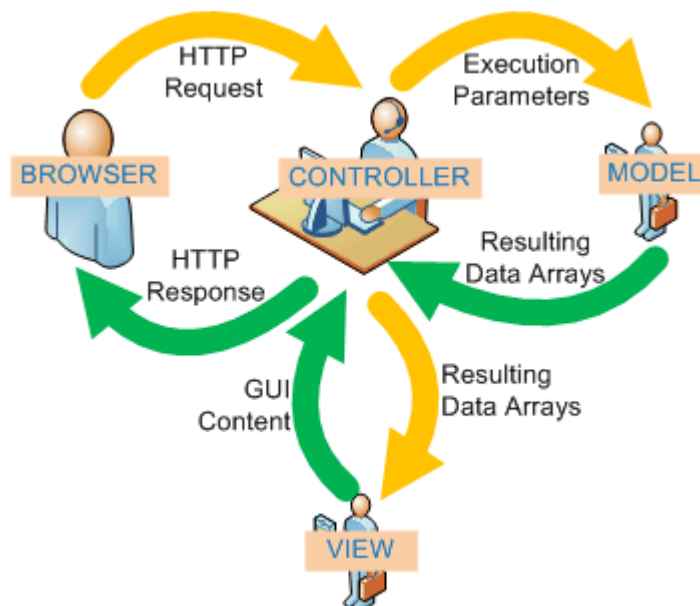


Figura 29. Patró MVC

Per tant, el SGBD i la lògica de negoci formen el model. La vista es l'encarregada de genera d'interfície gràfica que es mostrarà a l'usuari. I el controlador es qui rep les

peticions corresponents al model, modificar el model i en funció dels paràmetres rebuts, redirigeix la vista. D'aquesta manera es poden programar les diferents capes de manera independent, proporciona un manteniment de l'aplicació més senzill, on es poden localitzar de forma àgil els errors. Suposa un disseny modular i molt poc acoblat, que afavoreix a la reutilització.

- **Patró Front Controller**

L'ús d'aquest patró obliga a totes les peticions a passar per un servlet amb funcions de controlador. El controlador proporciona un punt d'entrada únic per totes les peticions. Això evita de tenir la mateixa lògica de control a totes les JSP. Generalment, s'utilitza junt amb un Dispatcher, que serà l'encarregat de redirigir el flux d'execució.

- **Patró Intercepting Filter**

Aquest patró tal com el seu nom indica s'utilitza tant per a interceptar les peticions d'entrada com les respostes de sortida i d'aquesta manera poder analitzar o inclús modificar l'informació. Això se'n diu un pre post processat. D'aquesta manera, es pot controlar si estem rebent codi maliciós, podem crear un log, també s'encarregarà de redirigir cada petició al servlet adequat.

Per cada acció podem tenir tots els filtres que necessitem, els filtres van encadenats y s'executen un darrera de l'altre en l'ordre en que es trobin.



### 3.7. Interfície gràfica

En aquest apartat veurem com es composaran tant les pàgines externes o públiques, a les que tindrà accés tothom i la composició de les pàgines de les zones privades on només tindran accés els usuaris registrats.



Figura 30. Zona pública



Figura 31. Zona privada

## **4. Capítol 4: Implementació**

### **4.1. Eines de desenvolupament**

#### **4.1.1. Base de dades**

Com a Base de dades utilitzada s'ha escollit Oracle 10g Express Edition, ja que es el SGBD que tenia instal·lat per altres assignatures cursades.

Per tal de poder comunicar amb la base de dades des del projecte necessitarem tenir instal·lat el driver de comunicació.

Concretament ojdbc14.jar (incorporar aquest arxiu a la carpeta Libraries del projecte).

Es pot descarregar gratuïtament des de la següent pàgina:

<http://www.findjar.com/jar/mule/dependencies/maven1/oracle-jdbc/jars/ojdbc14.jar.html>

#### **4.1.2. Utilitats BBDD**

Per tal d'administrar la base de dades s'ha utilitzar l'SQLDeveloper.

#### **4.1.3. Entorn de desenvolupament**

Com a entorn de desenvolupament s'ha utilitzat Netbeans, es tracta d'un producte lliure i gratuït, sense cap restricció d'us. La versió utilitzada es Netbeans 6.9.1. Com l'aplicació que s'ha de desenvolupar es J2EE, llavors es recomanable escollir l'opció de descarrega Java EE, amb aquesta opció ens descarregarem totes les eines necessàries per la nostre aplicació, a més aquesta opció també inclou la descarrega dels servidors GlassFish OSE (Open Source Edition) 3.0.1, i Apache Tomcat 6.0.26.

Pàgina de descarrega Netbeans:

<http://netbeans.org/downloads/>

#### **4.1.4. Contenedor d'aplicacions**

S'ha utilitzat Apache Tomcat 6.0.26, versió que ja porta incorporada Netbeans amb l'opció de descarrega escollida. Tomcat es tracta d'un contenidor d'aplicacions o contenidor Servlet/JSP i servidor web. Com Tomcat es escrit en Java, pot funcionar a qualsevol sistema que disposi de màquina virtual Java.

#### **4.1.5. Struts2**

La versió utilitzada es Struts2.2.1.8 Basic.

Per tal d'incorporar Struts2 a la nostra aplicació primer farem la descarrega <http://struts.apache.org/2.x/>

Un cop descarregat només hem d'afegir al nostre projecte a la carpeta Libraries els següents arxius .jar:

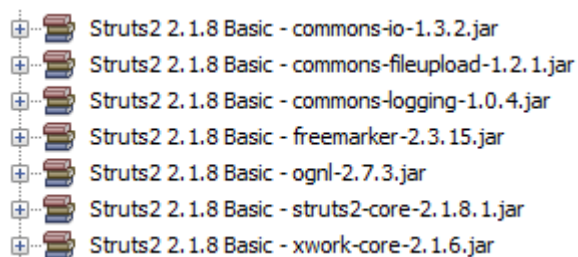


Figura 32. Llibreries bàsiques Struts2

#### 4.1.6. Persistència de les dades

La versió emprada d'hibernate es la 3.2.5, aquest framework es pot escollir a l'hora d'instal·lar netbeans al pas 4. I directament ens afegirà a la carpeta Libraries els següents arxius .jar:

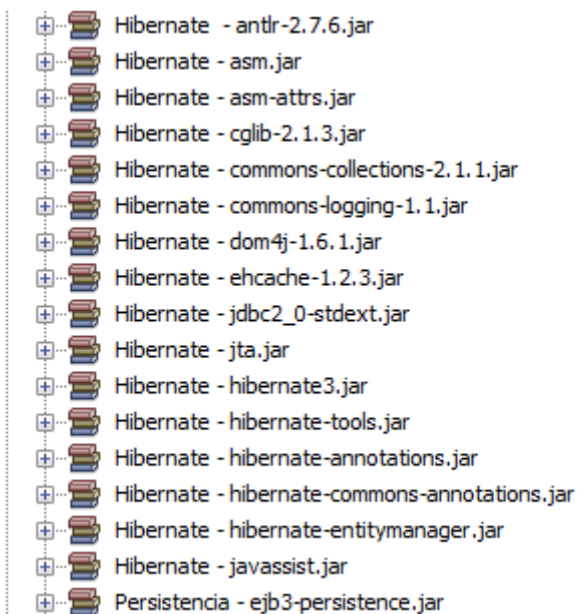


Figura 33. Llibreries Hibernate

## 4.2. Manual de configuració de les eines emprades

En aquesta secció s'explica com configurar cada part de l'aplicació per a un correcte funcionament.

### 4.2.1. Configuració de la BD

Per a la preparació de la base de dades s'han creat uns scripts per a facilitar aquesta tasca. Aquests scripts s'entreguen junt amb la resta d'arxius de l'implementació, es troben dins una carpeta anomenada SQL.

#### Primer pas

Primer de tot com a usuari SYSTEM, crearem un tablespace per a la nostre aplicació i un usuari donant-li el privilegis necessaris. Per a fer això accedirem a la base de dades com a usuari SYSTEM i executarem l'script anomenat User.sql. El nom d'usuari utilitzat es TFC i el password passTFC.

Segon pas

Un cop ja tenim creat el tablespace i l'usuari, accedirem a la base de dades com a TFC(nou usuari creat al punt anterior) i crearem totes les taules del nostre projecte. Per aquesta tasca executarem l'script taules.sql.

Tercer pas

Ara també com a usuari TFC crearem les seqüències, les quals posteriorment a partir d'hibernate cridarem per tal de crear la clau primària de cada una de les taules. Per tant, crearem una seqüència per a cada taula. Per a la creació de les seqüències, executarem l'script sequencies.sql.

Quart pas

Aquest es l'últim pas que hem de realitzar, es tracta de l'inserció de les dades inicials, necessàries pel funcionament de l'aplicació. En aquest pas insertarem les 3 categories de productes i l'administrador de l'aplicació. L'administrador que es crearà tindrà com a username admin i com a password admin. Per aquesta tasca s'ha d'executar l'script InsercionsInicials.sql.

Ara ja tenim la base de dades preparada.

**4.2.2. Configuració d'hibernate**

Primer de tot s'ha de crear l'arxiu de configuració d'hibernate, per a fer-ho ens col·loquem a la carpeta predeterminada, botó dret del ratolí, seleccionem nou i altre, aquí escollim la carpeta Hibernate i dins de les opcions assistent per a la configuració d'hibernate. Ens demanarà un nom, posem hibernate.cfg i l'últim pas es indicar-li les dades de la base de dades, en el meu cas son les següents:

Nom servidor: Oracle Thin(con Servicio ID(SID))

Servidor: localhost

Port:1521

Servicio ID(SID): xe

Nom d'usuari: TFC

Contrasenya: passTFC

I tot això ens a de generar l'url:

jdbc:oracle:thin:@localhost:1521:xe

Ara generarem un arxiu d'enginyeria inversa, on es guardarà l'informació de la nostra base de dades. Per a fer això, fem botó dret del ratolí, seleccionem nou i altre, aquí tornem a escollir la carpeta Hibernate i seleccionem l'opció Assistent d'enginyeria inversa d'hibernate. Aquí l'indiquem quin es el nostre arxiu de configuració d'hibernate(creat en l'apartat anterior) i netbeans ens

retornarà totes les taules de la nostra base de dades, les seleccionem totes i finalitzem, amb això se'ns haurà creat l'arxiu hibernate.reveng.xml.

Gracies a aquest últim arxiu ara li podem demanar a netbeans que ens generi automàticament els arxius amb els mapes d'hibernate i els POJOS. Per fer això tornem a fer botó dret del ratolí, nou, altre i escollim carpeta Hibernate i seleccionem l'opció arxius de mapes d'hibernate i POJOS de la base de dades. Aquí només hem d'indicar a els dos arxius creats anteriorment i netbeans ens crearà els mapes i POJOS.

Com els usuaris a la base de dades s'emmagatzemen tots dins de la taula Usuaris, però en canvi es vol aprofitar l'herència i tenir les classes Client, Treballador i Administrador que hereten d'Usuari, llavors haurem de modificar l'arxiu Usuari.hbm.xml per tal de corregir-lo, també s'ha hagut de corregir el tipus de generador de la clau primària indicat a cada arxiu de mapeig quina seqüència li correspon.

Als arxius de mapeig es pot veure aquestes modificacions. També s'han tingut que crear els POJOS Client, Treballador i Administrador.

Només necessitem crear un altre arxiu l'HibernateUtil.java que s'encarregarà de crear les sessions. Igual que amb la resta d'arxius, ens ho pot crear netbeans, fent botó dret del ratolí, seleccionem nou, altre, carpeta Hibernate i seleccionem HibernateUtil.java.

A partir d'ara ja no ens hem de preocupar més de la base de dades, serà hibernate qui s'encarregarà.

Per millorar el rendiment de l'aplicació s'ha creat un pool de connexions a hibernate, ja que crear una nova connexió surt car i mantenir connexions ocioses també. Per a crear un pool de connexions a hibernate, hem d'obrir l'arxiu de configuració(hibernate.cfg.xml) i afegir les següents línies:

```
<property name="hibernate.c3p0.min_size=5"></property>
<property name="hibernate.c3p0.max_size=20"></property>
<property name="hibernate.c3p0.timeout=300"></property>
<property
name="hibernate.c3p0.max_statements=50"></property>
<property
name="hibernate.c3p0.idle_test_period=300"></property>
```

El que estem indicant amb aquestes línies es:

- El nombre mínim de connexions que es mantindran preparades.
- El nombre màxim de connexions al pool.

- El període de temps en segons després del qual si una connexió no es utilitzada, s'eliminarà del pool.
- El nombre màxim de sentències preparades a la memòria intermèdia.
- El temps en segons que una connexió deu estar sense utilitzar-se perquè es validi de forma automàtica.

#### 4.2.3. Configuració d'Struts2

Com a controlador de l'aplicació s'ha definit StrutsPrepareAndExecuteFilter, controlador recomanat a partir de la versió d'Struts 2.1.3, es tracta d'un servlet filter amb funcionalitat de front-controller que s'encarregarà d'interceptar i gestionar qualsevol petició que arribi. Per a indicar a Apache Tomcat que tota petició a de passar per aquest controlador, s'ha de definir a l'arxiu web.xml.

```
<filter>
    <filter-name>struts2</filter-name>
    <filter-
class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepare
AndExecuteFilter</filter-class>
</filter>
    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
```

Per tal de definir el mapeig dels actions, s'ha utilitzat l'arxiu de configuració, struts.xml.

Dins aquest arxiu es troben 5 includes dels 5 arxius xml, que configuren l'aplicació, cada un d'aquest arxius pertany a un package.

La divisió dels package s'ha fet pels diferents tipus d'usuaris possibles, tenim l'usuari anònim, l'usuari autènticat, client, treballador i l'administrador.

D'aquesta manera al tenir-ho dividit pel tipus d'usuaris es pot configurar amb facilitat els interceptors que actuen sobre cada usuari i les condicions que ha de complir cadascun.

L'usuari anònim només tindrà dret a accessos a les pàgines sense privilegis, fora de les zones privades (zona clients, zona treballadors i zona administració).

Al package usuari autènticat es troben les actions comunes per als usuaris autènticats.

I les altres 3 packages com el seu nom indica es troben les actions per al tipus d'usuari indicat.

Cada package té definit un interceptor-stack, tots inclouen defaultStack però es diferencien del l'interceptor que

acredita a l'accés, ja que s'ha creat un interceptor per a cada tipus d'usuari.

D'aquesta manera un usuari només pot fer crides als accions que li pertoca, si intentes fer una crida a qualsevol altre acció seria reenviat cap a la pàgina principal (zona externa).

Per a cada JSP també s'han definit condicions per a que no puguin ser cridades només que pels usuaris que pertoca, sinó també es reenvia cap a la pàgina principal.

### 4.3. Estructura principal del projecte

L'estructura del projecte es pot veure a la següent imatge:

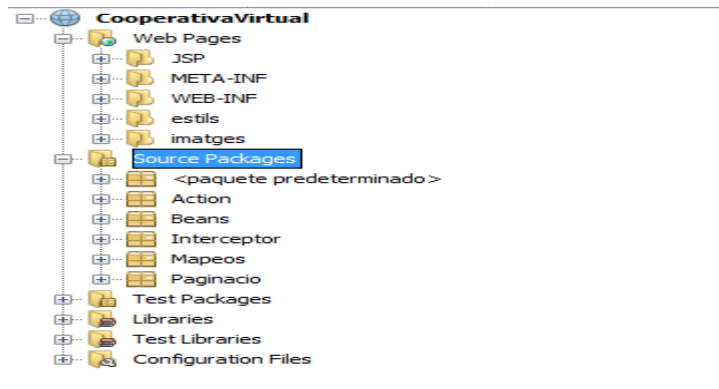


Figura 34. Estructura principal del projecte

Ara explicarem quin es el contingut de les carpetes més importants. Tal com es pot veure a l'imatge els noms de les carpetes son bastant intuïtius. Començarem per Web Pages.

A la carpeta JSP trobarem totes les pàgines de l'aplicació, podem veure a la següent imatge aquestes JSP's.

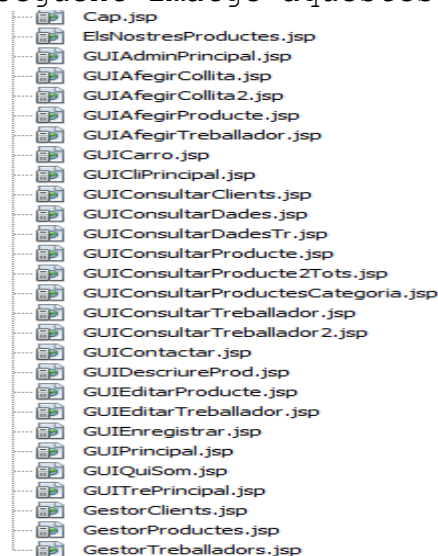


Figura 35. Estructura carpeta JSP

A la carpeta WEB-INF, trobarem el fitxer descriptor de desplegament de l'aplicació, el web.xml. Tal com ja s'ha comentat en aquest fitxer indicarem a Tomcat que tota petició passi pel controlador d'Struts2, també s'indica el time-out de la sessió i la pàgina de benvinguda.

A la carpeta estils trobarem el fitxer css que s'utilitza per donar un format homogeni a les pàgines que es mostren de l'aplicació.

I finalment trobem la carpeta imatges, on es troben emmagatzemades totes les imatges de l'aplicació i a més conté 3 subcarpetes per les 3 categories de productes existents. És molt important emmagatzemar cada imatge dels productes en la carpeta adient, es a dir, si volem emmagatzemar la imatge de taronges, s'ha d'emmagatzemar a la carpeta fruites, ja que l'aplicació anirà a buscar aquesta imatge a dins d'aquesta carpeta.

Ara passem a explicar l'estructura de la carpeta source packages que conté tots els paquets de l'aplicació.

Primer trobem <paquete predeterminado> on es troben els fitxers de configuració d'Struts2 i d'Hibernate.

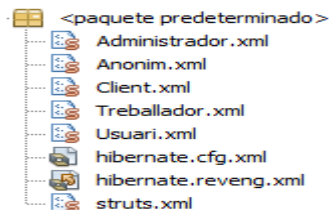


Figura 36. Estructura paquet predeterminat

Tots aquests fitxers s'han explicat als apartats de configuració d'Hibernate i Struts2.

Al paquet Action, trobarem totes les accions del projecte, com podem veure a les següents imatges es el més extens.

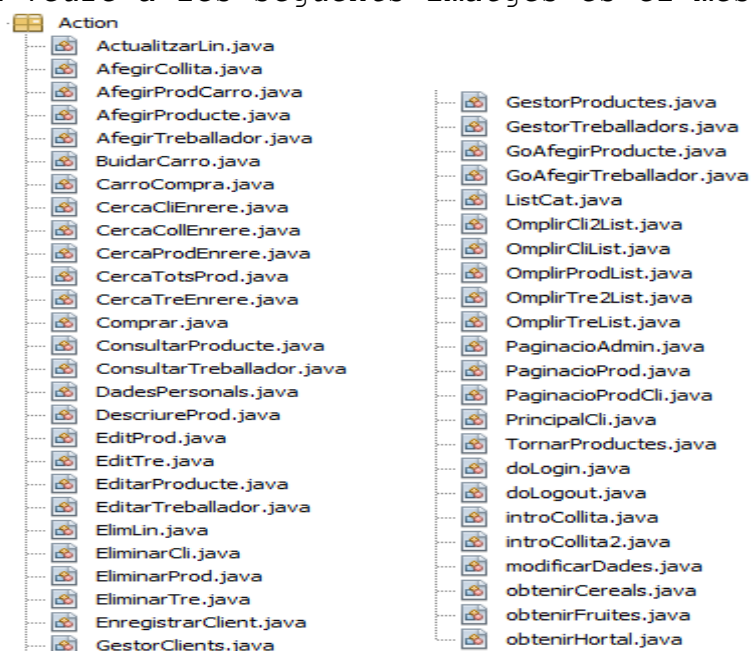


Figura 37. Estructura Actions



Al paquet Beans trobarem totes les Classes del projecte i l'arxiu HibernateUtil.java, ja explicat anteriorment.

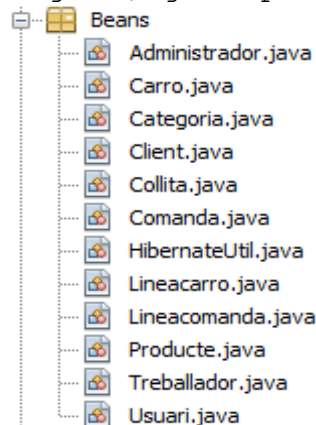


Figura 38. Estructura Beans

Al paquet Interceptor trobem els 4 interceptors creats per a verificar l'autenticitat dels usuaris.

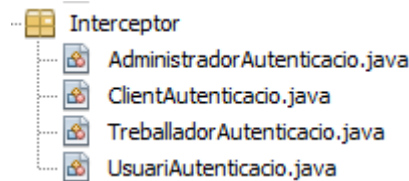


Figura 39. Estructura Interceptor

Tal com es pot veure disposem de un interceptor per a cada tipus d'usuari (Client, Treballador i Administrador) i un quart interceptor comú, que s'utilitza per a accions comunes dels usuaris.

Al paquet Mapeos trobem els fitxers de mapeig necessaris per Hibernate.

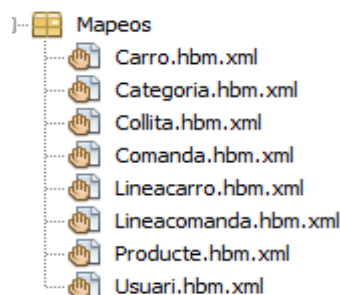


Figura 40. Estructura Mapeig

I finalment trobarem el paquet Paginacio que conté una utilitat per a la paginació dels resultats obtinguts. El fitxer es Paginacio.java.

Per donar un entorn visual més agradable s'han definit que es mostrin els productes de 5 en 5, els clients de 10 en 10 i els treballadors de 8 en 8.

Tot això es pot modificar molt fàcilment dins aquest fitxer.

#### 4.4. Manual d'usuari

En aquest apartat descriurem les principals pàgines de l'aplicació.

A partir de la pàgina principal els usuaris podran accedir a la zona privada corresponent al tipus d'usuari. També els possibles clients es podran enregistrar com a clients de la cooperativa, es té accés al catàleg de productes, a la pàgina d'informació de l'empresa i a visitar les dades de contacte.



Figura 41. Pàgina principal

##### 4.4.1. Zona administrador

Per tal d'accedir com a administrador,

nom d'usuari: admin

contrasenya: admin

Al pitjar al botó Accedir, l'aplicació ens portarà cap a la zona d'administració.



Figura 42. Pàgina administració

Com es pot observar un cop hem accedit a la zona d'administració ens dona accés a la gestió de productes, clients i treballadors.

També des de qualsevol gestor ens podem desplaçar cap a un altre gestor.

Observem cada gestor per separat.

#### Gestor de productes

Al gestor de productes podrem afegir un producte nou o consultar els productes, a la consulta de productes podrem modificar o eliminar qualsevol producte.



Figura 43. Pàgina gestor productes

Afegir producte:

Per afegir un producte necessitarem introduir totes les dades que ens demanen, s'ha de tenir en compte que el nom de l'imatge a de estar present dins de la carpeta imatges i dins de la subcarpeta corresponent a la seva categoria. Per exemple, introduïm el Producte Taronja.

Mirem el nom del fitxer que conté l'imatge de Taronja.

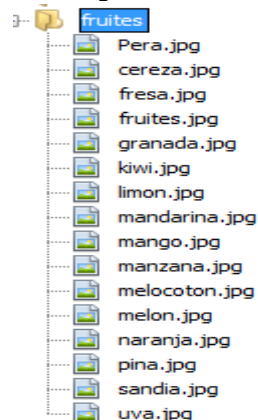


Figura 44. Imatges adjuntes

Com veiem a l'imatge anterior, triariem naranja.jpg

**Sessió iniciada per:**  
David Alvarez

Gestor de productes  
Gestor de clients  
Gestor de treballadors  
Tancar Sessió

**Dades producte**

Categoria\*: FRUITES

Nom producte\*: Taronja

Descripció del producte: Taronges de l'horta valenciana

Preu del producte\*: 2.25

Nom arxiu de l'imatge del producte\*: naranja.jpg

Afegir Producte

Figura 45. Pàgina afegir producte

Un cop hem introduït els productes que ens interessin, podem anar al gestor de productes i consultar-los.

**Sessió iniciada per:**  
David Alvarez

Gestor de productes  
Gestor de clients  
Gestor de treballadors  
Tancar Sessió

**Cerca de productes**

**Cercar productes per nom**  
Introdueixi el nom del producte i premi el botó

**Cercar productes per categoria**  
Seleccioni primer la Categoria del producte i després premi el botó

Cercar productes

Cercar productes

Figura 46. Pàgina cerca de productes

Com podem veure els productes els podem consultar per nom o per categoria. Si consultem per nom, podem introduir el nom sencer, si volem veure tots els productes que comencin per T, llavors només introduïm T i premem el botó cercar i si no introduïm res, sens mostraran tots els productes. També podem consultar per categories, només hem de seleccionar una categoria i sens mostraran tots els productes d'aquesta categoria.

Si per exemple no poséssim res en la cerca per nom, obtindríem el següent:



Figura 47. Pàgina edició de productes

Com podem veure hem obtingut tots els productes que disposem, els productes es mostren de 5 en 5 i per tant si volem veure més podem seleccionar endavant. També a la part superior tenim un botó per torna enrere i varia els criteris de la cerca.

A cada línia de producte podem escollir editar o eliminar el producte.

#### Gestor de clients

Al gestor de clients podrem cercar els clients, com podem veure a la següent imatge podrem cerca els client per nom, cognoms o DNI, també aquesta cerca permet introduir només una part i es buscaran coincidències.



Figura 48. Pàgina cerca de clients

I aquesta cerca ens portaria a la següent pantalla on podríem eliminar els clients que ens convingui.



Figura 49. Pàgina eliminació de clients

El gestor de treballadors te unes característiques semblants als gestors que hem vist, ens permetrà afegir treballadors o consultar-los, dins la consulta de treballadors podríem editar o eliminar el treballador que vulguem.

#### 4.4.2. Zona treballadors

Ara anem a veure la zona dels treballadors. Per això sortim(Tancar Sessió) i introduïm nom d'usuari i contrasenya d'un treballador que haguem creat com a administradors i accedim a la zona de treballadors.



Figura 50. Pàgina principal area treballadors

Com podem veure als treballadors se'ls permet afegir collites o consultar les seves dades personals, si consulten les seves dades tenen l'opció de modificar-les. Per afegir una collita se'ns demanarà que escollim la categoria del producte que hem collit, un cop seleccionada i acceptada, es passarà a la següent pàgina:

Figura 51. Pàgina afegir collita

El camp Nom Producte es tracta un camp autocomplete, per tant, podem seleccionar un producte directament o introduir el nom, el propi camp anirà auto completant el que escrivim. També per facilitar la feina es pot seleccionar la data de la collita a través del calendari.

#### 4.4.3. Zona clients

A partir de la pàgina principal, els clients podran veure el catàleg de productes, consultar les seves dades personals i veure el seu carro de la compra.

Figura 52. Pàgina principal area clients

Si anem al catàleg de productes visualitzarem els productes per categories i tindrem l'opció d'afegir els productes que necessitem al carro. També si pitgem a sobre del nom del producte, sens mostrarà la seva descripció.



Figura 53. Pàgina catàleg productes

Si anem al carro de la compra podrem modificar la quantitat dels productes seleccionats i si no ens interessa algun producte també el podrem eliminar-lo del carro. Finalment també tindrem les opcions de realitzar la compra o de buidar el carro.



Figura 54. Pàgina carro compra

Si el que volem es modificar alguna dada personal o modificar el nom d'usuari o contrasenya o podem fer des de Consulta dades personals i anirem a la següent pantalla.



Figura 55. Pàgina dades personals



#### 4.5. Proves realitzades

S'han realitzat proves de carrega i rendiment a l'aplicació, el programa utilitzat a sigut JMeter. He definit 30 fils d'execució amb un temps de pujada d'un segon i un bucle sens fi. A la següent imatge podem veure el resultat obtingut sobre la pàgina principal. Obtenint un bon rendiment amb un percentatge d'errors del 0%.

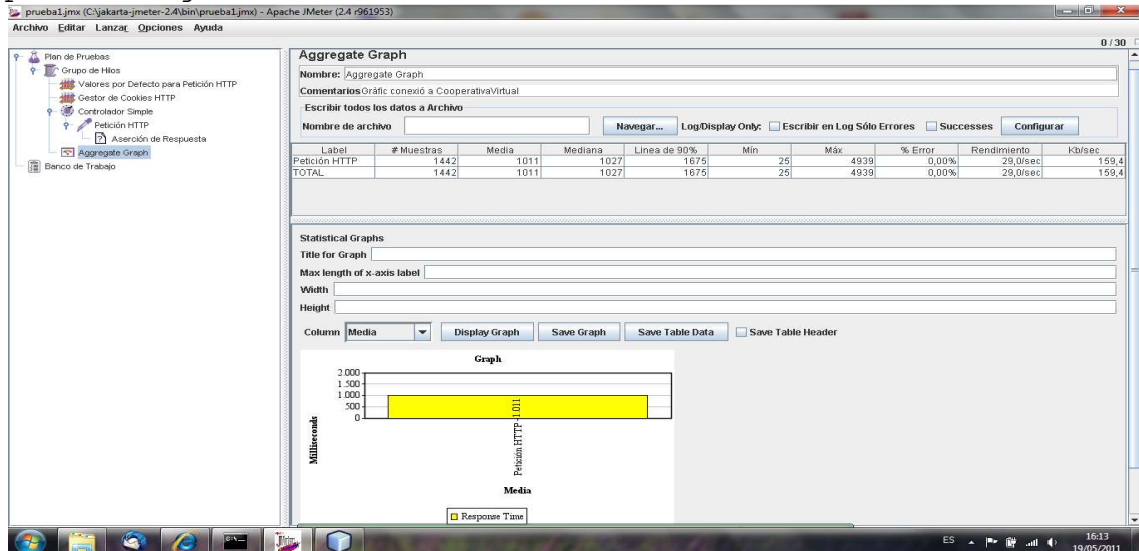


Figura 56. Test 1

A la següent imatge podem veure el resultat obtingut sobre la pàgina del catàleg de productes, amb la qual s'accedeix a la base de dades per obtenir els productes. Obtenint un bon rendiment amb un percentatge d'errors del 0%.

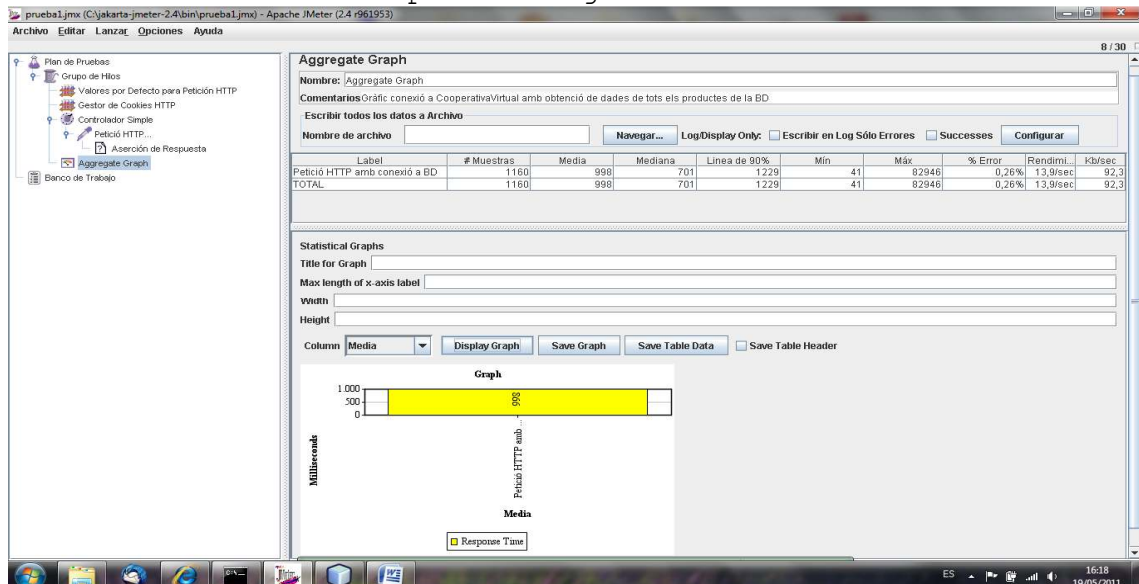


Figura 57. Test 2

Finalment s'han realitzant les dues proves anterior en conjunt, es a dir, accedint a la pàgina principal i a la pàgina del catàleg de productes. Obtenint un bon rendiment amb un percentatge d'errors del 0%.

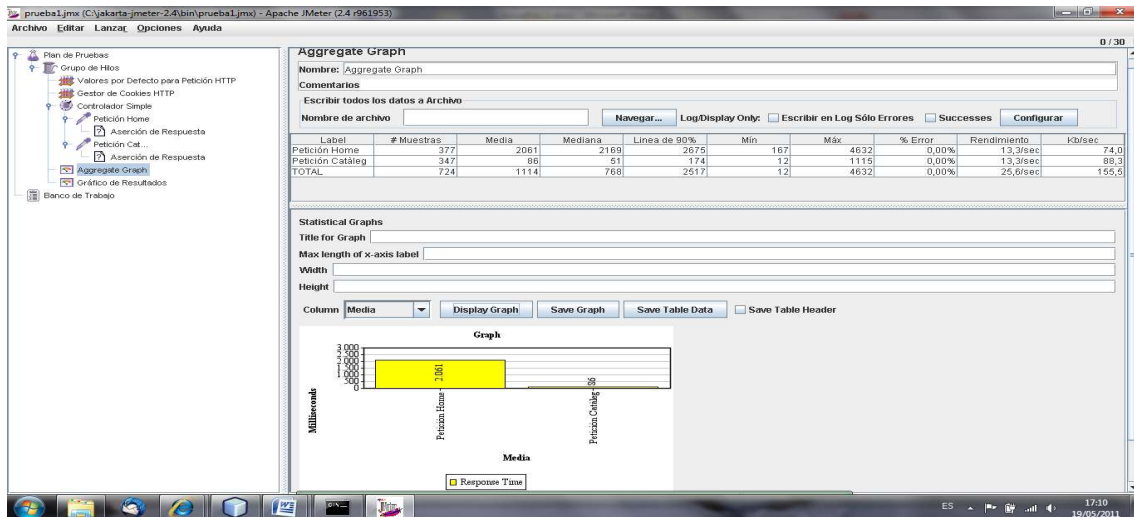


Figura 58. Test 3

En aquesta última prova també s'ha capturat un gràfic on es pot veure la mediana la mitjana, la desviació i el rendiment.

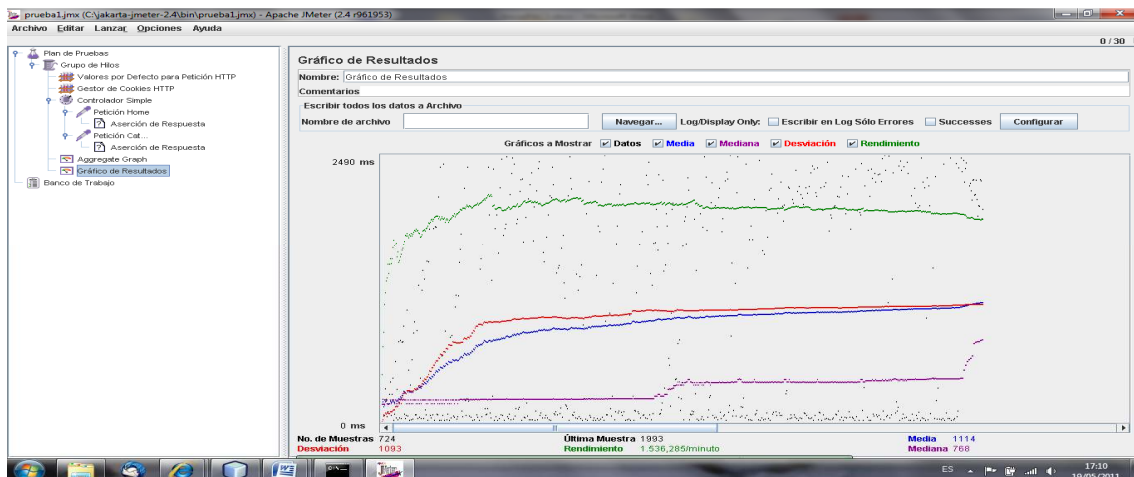


Figura 59. Test 3.1

#### 4.6. Alguns consells per a provar l'aplicació

Com inicialment l'aplicació només tindrà un administrador i les categories dels producte, ja que les haurem afegit al executar l'script InsercionsIniciais, llavors es recomanable accedir a l'aplicació com a administrador i primer de tot afegir un quants productes i al menys un treballador.

Un cop fet això s'hauria d'accedir a l'aplicació com a treballador i afegir algunes collites de productes, d'aquesta manera incrementarem les existències dels productes que hem creat, ja que de bon principi les existències son a zero.

I a partir d'aquí ja podem provar l'aplicació com a clients i podríem realitza compres.

## 5. Capítol 5: Conclusions

L'aprenentatge J2EE inicialment es lent, es necessita cercar molta informació, ja que existeixen moltes solucions per un mateix problema. Això es degut a d'inexperiència en aquest tipus d'aplicacions i el desconeixement inicial de tots els productes que hi ha al mercat, com poden ser els frameworks que es poden utilitzar. Però quan es porten uns quants dies programant i fent proves, es va veien que cada vegada tot va més ràpid i es a les hores quan es veu el gran potencial de producció d'aplicacions que dona J2EE. Personalment Struts2 m'ha agradat molt, ja que facilita molt la feina. La instal·lació e integració dels diferents frameworks a estat fàcil i ràpida. Afegir funcionalitats com Ajax o d'altres es molt senzill, només es necessari afegir les llibreries corresponents.

Es molt important la utilització de patrons de disseny, ja que et donen la millor solució a problemes que d'altres ja han tingut. Una de les coses que modificaria i no que no he pogut refer per falta de temps es la utilització del patró DAO, amb aquest patró hagués obtingut més independència a l'hora d'accedir a la base de dades.

Una altra cosa que m'hagués agradat fer i que no he pogut provar es la creació d'índexs i vistes a la base de dades per tal d'optimitzar el seu rendiment. Encara que per les proves de rendiment que he realitzat, el resultat obtingut ha sigut bo.

## 6. Glossari

**Autenticar-se:** Per tal d'accedir a la zona privada, un usuari ha d'introduir el seu nom d'usuari i contrasenya per tal d'identificar-se com a tal davant del sistema.

**Carro compra:** cistella virtual on els clients poden anar emmagatzemant els productes que desitgen compra i que poden veure i modificar en qualsevol moment.

**Client:** Usuari de l'aplicació que pot realitzar compres dels productes que es venen a la cooperativa.

**Treballador:** Usuari de l'aplicació que pot realitzar insercions de collites dels productes.

**Administrador:** Usuari de l'aplicació encarregat de l'administració de clients, treballadors i productes.

**Zona clients:** zona privada de l'aplicació on poden accedir exclusivament els clients.

**Zona treballadors:** zona privada de l'aplicació on poden accedir exclusivament els treballadors.

**Zona administració:** zona privada de l'aplicació on pot accedir exclusivament l'administrador.

## 7. Bibliografia

### J2EE

<http://download.oracle.com/javaee/>

<http://www.programacion.com/articulo/introduccion a j2ee 208>

<http://java.sun.com/j2ee/overview.html>

### Patrons de disseny J2EE

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=patronesj2ee>

<http://www.programacion.com/articulo/catalogo de patrones de diseno j2ee i capa de presentacion 240>

<http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>

### Struts 2

<http://struts.apache.org/2.x/index.html>

<http://www.javahispano.org>

<http://www.roseindia.net/struts/struts2/index.shtml>

<http://programacionconejemplos.blogspot.com/>

<http://www.struts2.net>

### Hibernate

<http://www.hibernate.org/>

<http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/index.html>