

# **Aplicació e-comerç desenvolupada en l'especificació J2EE**

## **Tenda on-line de llibres. BookSector**

**Pere Prohens Galmés**  
ETIG

**Albert Grau Perisé**

18/06/2011

## 1. Dedicatòria i agraïments

Vull agrair, primer de tot a la meva parella, per recolzar-me i tenir la paciència que ha tingut durant el meu trajecte en aquest TFC i durant tota la carrera. M'ha animat quan ho he necessitat i ha fet que tot sigui més bo de dur.

També vull donar les gràcies a la meva família i als meus amics per ser comprensius amb les meves absències durant aquest aquests semestres tan durs, als quals no he pogut dedicar gaire temps.

Al nostre consultor del TFC Albert Grau per estar sempre atent a les meves andanades de correus enviats i per respondre'ls de forma tan eficient.

Al tots els consultors per gestionar les assignatures de forma tan ben estructurada i per estar allà per a donar una mà.

## 2. Presentació i continguts

Aquesta memòria de projecte pretén ser un complet document, en el qual trobarem ,primer de tot, una petita descripció del que es vol fer i el perquè. Llavors s'aprofundeix més en l'objectiu que es marca per al projecte i quina orientació i tecnologia s'escull per a ell, així com el perquè de les decisions preses i els productes obtinguts.

Seguidament aprofundirem en els documents d'especificació i anàlisi, el disseny i la fase d'implementació. Cada fase estarà totalment diferenciada de les altres i es farà ús de varis diagrames, gràfics i imatges; les quals estaran indicades en l'índex de contingut del document.

Finalment es revisaran altres aspectes complementaris com la valoració econòmica, les conclusions, així com un glossari de termes i la bibliografia. Altres continguts importants s'han recollit en varies que es poden utilitzar per separat.

Com es veurà en la descripció, el projecte tracta sobre una aplicació web distribuïda client-servidor, desenvolupada en Java i l'especificació J2EE i el recull de tecnologies que aquesta marca. De forma general es tracta d'una pàgina web de compra de llibres per a un petit negoci de llibreria que es vol orientar a les ventes on-line a petita escala. Per a tal fi, s'han utilitzat les tecnologies de la J2EE com Struts2, Hibernate i JSP, entre altres.

## Índex

<b>1. Dedicatòria i agraïments .....</b>	<b>1</b>
<b>2. Presentació i continguts.....</b>	<b>1</b>
<b>3. Introducció .....</b>	<b>6</b>
<b>3.1 Justificació: Punt de partida i aportació del TFC. ....</b>	<b>6</b>
<b>3.2 Objectius del TFC .....</b>	<b>7</b>
<b>3.3 Enfocament i mètode seguit.....</b>	<b>7</b>
<b>3.4 Planificació del projecte .....</b>	<b>8</b>
<b>3.5 Productes Obtinguts .....</b>	<b>10</b>
<b>3.6 Descripció dels altres capítols de la memòria. ....</b>	<b>10</b>
<b>4. Especificació i anàlisi del projecte. ....</b>	<b>11</b>
<b>4.1. El perquè d'aquest document.....</b>	<b>11</b>
<b>4.2. Descripció general de projecte .....</b>	<b>11</b>
<b>4.3. El projecte real.....</b>	<b>13</b>
<b>4.4. Sistema basat en n-capes .....</b>	<b>15</b>
<b>4.5. Màquines i contenidors.....</b>	<b>15</b>
<b>4.6. Subsistemes i paquets.....</b>	<b>16</b>
4.6.1. Subsistemes.....	16
4.6.2. Paquets.....	17
<b>4.7 Funcionalitats .....</b>	<b>18</b>
<b>4.8 Actors.....</b>	<b>19</b>
<b>4.9 Especificació textual dels casos d'ús .....</b>	<b>21</b>
4.9.1 Diagrama de casos d'ús .....	21
4.9.2 Descripció i detall dels casos d'ús.....	23
4.9.2.1 Subsistema de Connexió .....	23
4.9.2.2 Subsistema de Catàleg .....	25
4.9.2.3 Subsistema de Compra .....	27
4.9.2.4 Subsistema d'Administració.....	29
4.9.3 Aclariments sobre els casos d'ús .....	33
<b>4.10 Requisits per a l'execució del programa.....</b>	<b>34</b>
<b>4.11 Possibles implementacions futures.....</b>	<b>34</b>
<b>5 Disseny del Projecte.....</b>	<b>35</b>
<b>5.1 El perquè del document de disseny .....</b>	<b>35</b>
<b>5.2 Plataforma/es i tecnologia .....</b>	<b>35</b>
5.2.1 J2EE Multicapa.....	35

<b>5.3</b>	<b>Hibernate .....</b>	<b>38</b>
<b>5.4</b>	<b>Struts .....</b>	<b>38</b>
<b>5.5</b>	<b>Diagrama estàtic de classes i jerarquies .....</b>	<b>39</b>
5.5.1	Diagrama estàtic de classes de la lògia de negoci .....	39
5.5.2	Jerarquies d'herència més importants .....	43
5.5.2.1	Diferents funcionalitats segons el Rol.....	43
5.5.2.2	Superclasse Llibre .....	44
<b>5.6</b>	<b>Diagrama d'estats.....</b>	<b>45</b>
<b>5.7</b>	<b>Diagrames de seqüència.....</b>	<b>46</b>
5.7.1	Subsistema de Connexió.....	46
5.7.2	Subsistema de Catàleg.....	49
5.7.3	Subsistema de Compres .....	52
5.7.4	Subsistema de Administració .....	54
<b>5.8</b>	<b>Persistència .....</b>	<b>59</b>
5.8.1	Diagrama entitat Relació .....	59
<b>6.</b>	<b>Implementació. ....</b>	<b>61</b>
<b>6.1</b>	<b>Requisits de programari per a la implementació .....</b>	<b>61</b>
<b>6.2</b>	<b>Eines de desenvolupament utilitzades.....</b>	<b>61</b>
<b>6.3</b>	<b>Estructura Interna del projecte.....</b>	<b>62</b>
<b>7.</b>	<b>Valoració econòmica .....</b>	<b>64</b>
<b>8.</b>	<b>Conclusions.....</b>	<b>64</b>
<b>9.</b>	<b>Glossari.....</b>	<b>65</b>
<b>10.</b>	<b>Bibliografia .....</b>	<b>66</b>
<b>11.</b>	<b>Annexos.....</b>	<b>68</b>
11.1	Annex I: Interfície gràfica. Petita guia de funcionament. ....	68
11.2	Annex II: Script de creació de la BD.....	75

## ÍNDEX DE FIGURES

Taula 1:	Planificació.....	8
Taula 2:	CU Connexió.....	23
Taula 3:	CU Desconnexió .....	24
Taula 4:	CU Login.....	24
Taula 5:	CU Registre Nou Usuari.....	25
Taula 6:	CU Buscar Llibre al Catàleg .....	26
Taula 7:	CU Consultar Llibre.....	26

Taula 8: Sol·licitar donació llibre vell .....	27
Taula 9: CU Afegir Llibre al Carretó .....	27
Taula 10: CU Consultar carretó .....	28
Taula 11: CU Suprimir Llibre del Carretó .....	28
Taula 12: CU Confirmar comanda.....	29
Taula 13: CU Canviar estat de comanda .....	30
Taula 14: CU Afegir Llibre al Catàleg.....	31
Taula 15: CU Suprimir llibre Catàleg.....	31
Taula 16: CU Editar Llibre.....	31
Taula 17: CU Buscar Usuari .....	32
Taula 18: CU Editar Usuari .....	32
Taula 19: CU Afegir Usuari .....	33
Taula 20: CU Suprimir Usuari .....	33
Imatge 1: Diagrama de Gantt.....	9
Imatge 2: Diagrama de capes simple.....	15
Imatge 3: Diagrama de contenidors .....	16
Imatge 4: Diagrama de Subsistemes .....	18
Imatge 5: Diagrama d'herència Actors .....	20
Imatge 6: Diagrama casos d'ús.....	22
Imatge 7: Diagrama de Capes simple 2 .....	35
Imatge 8: Diagrama de capes complex.....	36
Imatge 9: Esquema estructura Struts2.....	39
Imatge 10: Diagrama estàtic classes 1 .....	40
Imatge 11: Diagrama estàtic classes 2 .....	41
Imatge 12: Diagrama herència classes 1 .....	43
Imatge 13: Diagrama herència classes 2 .....	44
Imatge 14: Diagrama d'estats comanda.....	45
Imatge 15: DA Login .....	47
Imatge 16: DA Registrar Nou Usuari.....	48
Imatge 17: DA Buscar Llibre a Catàleg.....	49
Imatge 18: DA Consulta Llibre .....	50
Imatge 19: DA Sol·licitar donació.....	51
Imatge 20: DA Afegir Llibre al Carretó .....	52

Imatge 21: DA Consultar Carretó .....	53
Imatge 22: DA Canviar estat Comanda.....	54
Imatge 23: DA Afegir Llibre a catàleg.....	55
Imatge 24: DA Editar Llibre .....	56
Imatge 25: DA Afegir Usuari .....	57
Imatge 26: DA Buscar Usuari.....	58
Imatge 27: Diagrama persistència 1.....	59
Imatge 28: Diagrama persistència 2.....	60
Imatge 29: Arbre de paquets 1.....	62
Imatge 30: Arbre de paquets 2.....	63
Imatge 31: GUI capçalera .....	68
Imatge 32: GUI capçalera validada .....	68
Imatge 33: GUI Inici .....	69
Imatge 34: GUI Catàleg .....	69
Imatge 35: GUI Fitxa de Llibre .....	70
Imatge 36: GUI Carretó.....	70
Imatge 37: GUI Confirmació Comanda .....	71
Imatge 38: GUI Donació .....	72
Imatge 39: GUI Gestió Catàleg, Nou llibre .....	72
Imatge 40: GUI Gestió comandes .....	73
Imatge 41: GUI Edició de comanda .....	73
Imatge 42: GUI Administració Usuaris .....	74

## 3. Introducció

### 3.1 Justificació: Punt de partida i aportació del TFC.

Com a Treball Final de Carrera, aquest TFC ha de servir per a plasmar i demostrar els coneixements adquirits durant aquests anys en l'enginyeria tècnica, tant pels que s'ha après directament de les assignatures cursades, com pels que s'han adquirit de forma autònoma i autodidàctica, cosa essencial i indispensable en el món de la programació.

Amb el TFC es cerca posar en pràctica la identificació d'un problema real, la planificació i l'estructuració d'una solució també real, partint dels coneixements adquirits durant aquests anys i la posta en marxa d'aquesta per a obtenir un producte robust i reutilitzable que satisfaci els requisits i necessitats establerts en un entorn d'aplicacions distribuïdes client-servidor i seguint uns patrons actuals i que estan en la capdavantera de la programació distribuïda.

Així la investigació, l'aprenentatge de diverses tecnologies, especificacions i tècniques de disseny i implementació seran un camí a seguir per a complementar aquests coneixements inicials amb els que s'ha iniciat el treball, i que portaran a bon port tot el treball fet durant els anys d'estudi i treball realitzats en la carrera.

El llenguatge JAVA, sens dubte, s'ha fet un lloc en el món de la programació, i no ha estat casualitat. Aquest llenguatge explota d'una forma admirable el paradigma de la programació orientada a objectes, i permet un fàcil aprenentatge i codificació, cosa que deriva a una depuració i a un manteniment més senzill de les aplicacions realitzades, comparat amb altres llenguatges de programació del mercat. També s'ha de destacar el grau de robustesa, escalabilitat i reutilització que el paradigma orientat a objectes i en concret el llenguatge Java ens ofereix.

Es tracta d'un llenguatge orientat a objectes, cosa que proporciona un gran nivell de reutilització i fiabilitat, i a més, gràcies a que els programes s'executen en la seva màquina virtual, aconseguim aplicacions independents de la plataforma i una gran portabilitat. Això últim és molt important en la programació d'aplicacions distribuïdes client-servidor en general, i en concret en les aplicacions web empresarials, on podem trobar un gran ventall de tecnologies que han de col·laborar per a complir uns objectius comuns; i per això Java és una gran elecció per a fer produir aquest tipus d'aplicacions.

L'especificació J2EE de Java (Java Platform, Enterprise Edition) ens proporciona una forma molt intel·ligent de programar aplicacions web distribuïdes, client – servidor, que s'executen en un servidor d'aplicacions, i ens determina com combinar i enllaçar diverses tecnologies (pròpies de Java i d'altres entitats), com Servlets, EJB, JSP, Apache Struts 2, que ens proporciona les tècniques necessàries per a utilitzar el patró MVC 2; o les tècniques de mapeig de classes entitats que ens ofereixen les llibreries o fitxers d'Hibernate. Tot això, amb la garantia de totes les avantatges de programar en

Java, que durant els anys que he passat en l'estudi d'aquesta enginyeria, han demostrat de sobres el seu potencial.

Per aquests motius Java i el J2EE és ideal per a la realització d'aquest projecte.

### 3.2 Objectius del TFC

De forma resumida es pretén crear una aplicació web distribuïda, client servidor, que permeti gestionar una tenda de compra-intercanvi de llibres (llibres físics) per comerços senzills i simples que desitgin entrar en el món de les ventes per Internet. Les quals gaudeixen de la seva clientela tradicional però que es veuen perjudicades per les tendències actuals de les compres on-line.

Es vol tractar el problema des de les dues cares de la moneda. Per una banda, la part amb la qual el client interactua i es comunica amb l'aplicació; per l'altra, la part amb la qual el propietari o administradors del negoci gestionen l'aplicació web per a tenir-la al dia. Tot això d'una forma fàcil i intuïtiva que no requereixi formació o coneixements de cap tipus.

L'objectiu del projecte és donar una solució eficient i eficaç per a tal problema, fent servir l'especificació dissenyada per Java J2EE. De forma més específica es farà ús de l'estructura multicapa que aquesta especificació ofereix amb el model MVC de tipus 2 ( Model – Vista – Controlador) fent ús de *Apache Struts 2* i la seva gestió de *EJB's i Servlets i planes JSP* que ens proporciona, així com altres tecnologies orientades al món Java com els connectors *JDBC* per a realitzar les connexions amb la base de dades amb la genial gestió que les llibreries *Hibernate* ens proporcionen per a definir una relació directa entre les classes Java d'entitat i el model relacional en la base de dades, mitjançant mapejos *XML* i anotacions en les pròpies classes.

### 3.3 Enfocament i mètode seguit

Des del principi del semestre ja es va veure que per a finalitzar el projecte amb èxit s'hauria de treballar en paral·lel amb dues línies principals. Per una banda, les PAC 1 i 2, requerien un temps a reflexionar i planificar un projecte que requeria de coneixements molt desconeguts per jo, però els quals havia d'adquirir per a poder continuar amb les etapes del projecte. Per altre banda, i com he dit, la corba d'aprenentatge requeria aprendre un bon grapat de tecnologies, especificacions i tècniques que necessitava per les etapes 3 i 4 del projecte.

Així doncs, durant el semestre s'ha optat per un treball a dues cares, que consistia en l'elaboració de totes les etapes del projecte mentre s'aprofundia i s'avançava per la pesada corba d'aprenentatge. El fet de començar des de zero en l'especificació J2EE (no de Java, ni MySQL, ni de JDBC...) ha fet que l'únic mètode a seguir hagi estat



aquest: seguir una corba d'aprenentatge complexa a mesura que requeria dels coneixements per poder avançar en les etapes de les PACs.

### 3.4 Planificació del projecte

Durant la realització del pla de projecte, es va realitzar una planificació adient a les dates límit marcades pel calendari docent de l'assignatura. En quasi tots els aspectes s'han respectat aquests terminis, i realment aquesta planificació ha servit per administrar el temps del que disposava i per no quedar enrere amb les diferents entregues que ens han demanat.

Per tant la planificació seguida és gairebé la mateixa que la que es va proposar en el pla de projecte, amb algun petit canvi en les entregues a realitzar.

El projecte anava lligat a una sèrie de PAC a entregar, les quals ens han servit per definir una planificació que ens ha permès dedicar el temps just i necessari a cada etapa que tindria el projecte. Cada una de les PAC va lligada a una fase de l'elaboració del projecte. S'han respectat les dates fixades al calendari, menys en la PAC 3, com s'explica a la taula següent

**Inici : 04/03/11** PAC1 Un document, que es considera el punt de partida del TFC, on hi ha d'haver descrites les funcionalitats, arquitectura, i planificació. El pla de treball.  
**Entrega:**  
**21/03/11**

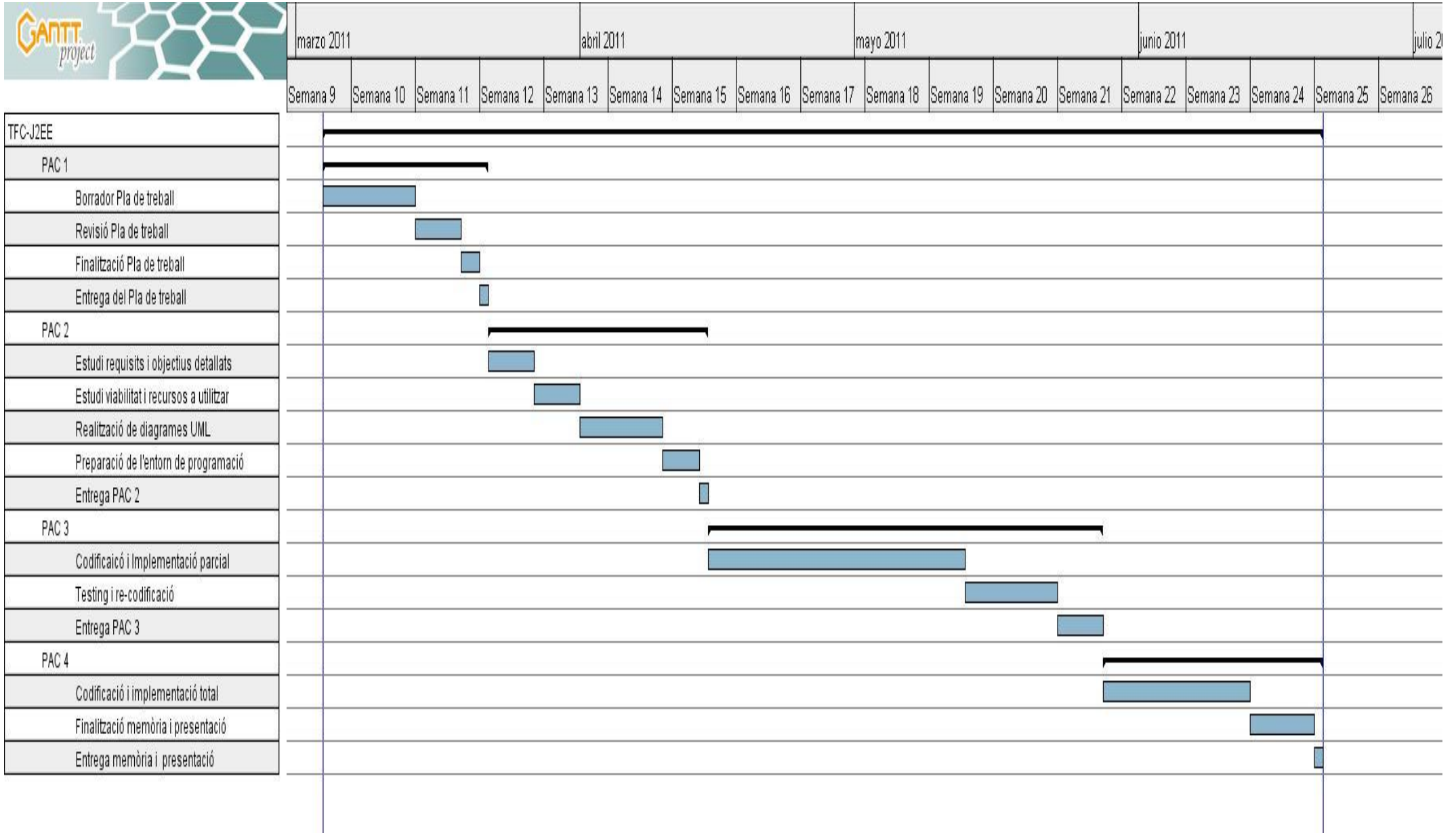
**Inici : 22/03/11** PAC2 Document complert amb tots els anàlisis, UML [diagrames de classes, d'us, algun de seqüència]  
**Entrega:**  
**14/04/11**

**Inici : 15/04/11** PAC3 Implementació de bona part del TFC. Aquesta entrega va ser aplaçada uns 4 dies. En lloc de fer-se dia 23, com estava marcat es va permetre l'entrega uns dies després. Fins dia 27, que va ser entregada.  
**Entrega:**  
**27/05/11**

**Inici : 24/05/11** PAC4 Memòria + Presentació Virtual + Codi  
**Entrega:**  
**20/06/11**

#### Taula 1: Planificació

En la següent plana hi ha un diagrama de *Gantt* on podem veure de forma gràfica la planificació i temporització del projecte.



Imatge 1: Diagrama de Gantt

### 3.5 Productes Obtinguts

Cada producte s'explica en detall en l'apartat oportú però a mode de resum els productes obtinguts fins al moment d'entregar la memòria són els següents:

- Carpeta del projecte amb els següents elements:
  - Carpeta *src* amb els fitxers Java de les classes amb tot l'arbre de *packages* utilitzats.
  - Carpeta *resources* on trobem altres fitxers de configuració com el de *Hibernate* o *Struts*.
  - Carpeta *WebContent* en la qual trobem el contingut web de l'aplicació, com planes *JSP*, el fitxer *XML web* i les imatges utilitzades. Com també, totes les llibreries utilitzades en el projecte
  - Un fitxer *build.xml* que s'executa mitjançant l'eina *Ant* i que s'encarrega de muntar el projecte i preparar-lo per desplegar-lo en el servidor Apache Tomcat 7.0
  - Carpeta *build* on es el *build.xml* genera tot el contingut, el qual és el que s'ha de lliurar al servidor per tal que es pugui executar.
  - Carpeta *dist*, en la que trobem el fitxer *war* del projecte, que facilita el desplegament. Tan senzill com dipositar aquest fitxer dins la carpeta *webapps* de Apache 7.0.
  - Carpeta *doc* amb tota la documentació generada per Java sobre el projecte.
- Petits manuals d'usuari amb les explicacions necessàries per a poder utilitzar l'aplicació.
- Fitxer SQL per a generar la base de dades i el seu contingut, de forma que l'aplicació tingui unes dades inicials.
- Fitxers *txt* amb instruccions addicionals i indicacions d'instal·lació i funcionament.

### 3.6 Descripció dels altres capítols de la memòria.

A continuació s'exposen en forma de capítols principals, els documents d'especificació i anàlisi, el de disseny, i les explicacions necessàries per a referir-se a l'etapa d'implementació. Totes aquestes amb les modificacions necessàries per adaptar-se al que finalment s'ha fet, ja que part d'aquesta documentació es va generar abans de fer la implementació.

Finalment trobarem altres capítols més complementaris com les decisions econòmiques, les conclusions finals, bibliografia, glossari i annexos.

## 4. Especificació i anàlisi del projecte.

### 4.1. El perquè d'aquest document

Un cop presentat el pla de treball del projecte, on s'oferia una visió general del camí que havia de seguir el treball, quins objectius es volien complir i com es volia arribar a ells; ha arribat el moment de presentar aquest document, on recollim l'especificació i l'anàlisi, més detallat, dels requeriments del programa, quins són, de quina forma es poden satisfer, i la solució que es donarà per tal fi.

Aquesta és una de les passes que es segueix a l'hora d'abordar un projecte de magnituds reals, i abans de començar amb la implementació d'aquest. En ell podrem trobar, d'una manera més formal, l'especificació de classes, casos d'ús identificats, etc... Tot això, sense entrar a l'etapa de disseny, que serà el pas previ a la implementació i que pretén vincular aquest document amb la implementació final.

### 4.2. Descripció general de projecte

En relativament pocs anys, els comerços de tot el món s'han vist davant d'una gran proliferació d'empreses que venen els seus productes a través d'Internet, permetent així un altre tipus de negoci, que a poc a poc s'ha convertit quasi en un estàndard en el món comercial i laboral. Tot i que moltes empreses venen tots els seus productes mitjançant tendes virtuals a través de la seva pàgina web, també ens trobem el cas de comerços més tradicionals que, durant molts d'anys han anat venent els seus productes de forma presencial al seu local, i que ara es veuen perjudicats al no poder competir amb les compres on-line que, cada cop més, la gent usa.

Amb aquest projecte pretenc crear una aplicació web distribuïda, client servidor, que permeti gestionar una tenda de compra-intercanvi de llibres (llibres físics) per comerços que desitgin entrar en el món de les ventes per Internet.

Des del punt de vista del possible client, l'aplicació web permetrà a aquest accedir al catàleg de llibres, per tal de, si així ho desitja, comprar-ne un amb un pagament on-line amb tarja de crèdit i dades bancàries (no es descarten, en un futur, acceptar altres formes de pagament). El client també podrà acumular punts amb l'opció de l'aplicació destinada a tal acció. Enviant els seus llibres usats al negoci. El propietari els revisarà i donarà punts al client, els quals podrà gastar per a obtenir altres llibres de segona mà, que per altra banda, també podrà obtenir comprant-los com si es tractés d'un llibre nou però amb un preu reduït en relació a l'estat del llibre.

A priori, no es necessitarà cap registre ni dades per entrar a l'aplicació web. Un client anònim, podrà entrar a l'aplicació i navegar pel catàleg de productes nous i usats. Serà necessari un registre amb les dades personals per tal de comprar llibres nous i usats i enviar els propis llibres al negoci per tal d'obtenir punts

Des del punt de vista del propietari del negoci, aquest podrà gestionar completament el catàleg de productes en venda (afegir, suprimir i actualitzar la informació dels llibres), els preus, les comptes d'usuari de tots els usuaris que s'hagin registrat a l'aplicació o crear-ne de noves, així com revisar les comandes que els clients han realitzats per fer-ne un seguiment i, si s'escau, canviar-ne l'estat.

L'aplicació estarà preparada pel possible cas que es volguessin afegir funcionalitats, en un futur.

Resumint doncs, l'aplicació permetrà:

- Al propietari del negoci, gestionar els productes, preus, comptes d'usuari, comandes, entre altres, per tal de tenir controlades les vendes i els productes que ven.
- Al client, registrar-se per poder comprar llibres nous pagant amb diners, o comprar llibres usats, mitjançant diners o punts obtinguts per la donació de llibres usats.

Explicava més a dalt, darrerament s'ha vist un increment significatiu de comerços que han orientat el seu negoci, en part o en la seva totalitat a la venda dels seus productes per internet. No només negocis de noves empreses, que ja han estat consolidades en plena època de les aplicacions web, sino comerços antics o que els seus productes es venen venent des de fa molts anys de la forma tradicional. Molts dels comerços tradicionals han decidit fer un canvi en la seva forma de negoci, i han implantat un sistema de comerç web, per recolzar les vendes d'una altre forma, i que per altre banda, ofereix una alternativa de negoci que permet accedir a un ventall de clients més ampli, que de la forma tradicional no seria possible.

Les tendes de llibres físics, les llibreries de tota la vida, poden ser molt bon exemple de negoci tradicional, que durant molts d'anys han venut els seus productes de forma presencial, i que es troben ara en una situació en que molts competidors venen els seus productes a través d'internet, mitjançant les pàgines web que ofereixen tendes on-line.

Abans d'entrar en més detalls, s'ha de dir que l'aplicació que es desitja desenvolupar com a projecte, va destinada a negocis minoritaris i petites empreses, i que no es tracta d'una gran aplicació industrial per a empreses multinacionals o que vinguin els seus productes arreu del món.

En concret ens referirem a una tenda en concret, encara que l'aplicació va potser adaptada fàcilment a un altre tenda de llibres físics.

### 4.3. El projecte real

L'empresa *BookSector* és una tenda que es dedica a la venda de llibres. El fundador va començar amb el negoci fa uns 20 anys i l'empresa gaudeix del seu públic, que agraeix el tracte personal que ofereix *BookSector* als seus clients. Al ser una tenda de llibres única ubicada a Palma (Mallorca), els clients van, de forma presencial a comprar els llibres que desitgin. *BookSector* és un local gran, amb gran varietat de llibres: des de llibres tècnics, els quals els estudiants universitaris en fan bon ús, a novel·la fantàstica, llibres d'assaig, poemes, literatura clàssica, etc...

L'actual propietari del negoci, és molt conscient del guany que ha tingut darrerament la competència, des de que han implantat pàgines web on vendre els seus llibres. Fins al dia d'avui, la tenda ha funcionat prou bé pel tracte personal i el bon servei que oferia als clients, però es creu que *BookSector* ha d'expandir-se a nous tipus de negoci, mantenint no obstant, el tipus de negoci clàssic i presencial que fins ara tant bé li ha funcionat.

*BookSector* necessita d'una aplicació web distribuïda que li permeti vendre els seus llibres on-line, mitjançant una pàgina web. La idea és que el client accedeixi a l'aplicació amb el seu navegador web amb una URL que disposarà *BookSector*. El client podrà explorar un catàleg de llibres ampli, disposant de diferents mètodes de recerca:

- Per nom
- Per autor
- Per categoria
- Per codi ISBN

El client, un cop ha trobat el llibre desitjat pot comprar-lo. Per tal fi, el client ha d'estar enregistrat da les BBDD de *BookSector*, cosa que pot fer fàcilment amb l'apartat web dedicat a tal fi. Un client que ja s'ha enregistrat, podar entrar amb el seu nom d'usuari i contrasenya. Si per contra, no està enregistrat, amb uns simple clic, podrà registrar-se les serves dades per tal de ser un usuari registrat de la tenda.

El sistema de compra via web de *BookSector* ofereix un mètode de compra per carretó. Es tracta d'un sistema de gestió de llibres que es volen escollir. Aquests són afegits al carretó de compra de l'usuari registrat. Aquest, a mesura que explora el catàleg, pot afegir llibres al seu carretó. Si decideix comprar els llibres afegits al carretó, només ha de triar les opcions de pagament i la forma d'entrega que ofereix el carretó. Llavors amb l'opció corresponent a la interfície gràfica, i es procedirà a enregistrar la comanda a la base de dades de l'aplicació i es mostrarà la confirmació de pagament i un resum de la comanda enregistrada.

També és possible editar el carretó. El client registrat pot entrar a consultar els llibres del seu carretó i, si així o desitja, eliminar-ne un a un, augmentar el nombre d'exemplars que vol d'un llibre en concret, buidar el carretó o afegir més llibres, cosa que el retornaria de nou al catàleg.

A l'hora de fer el pagament, aquest es farà amb les dades bancàries que el client ha enregistrat a l'hora de fer el registre. Si aquestes dades no són correctes se li notificarà per mail. Abans de confirmar la comanda, s'han de triar dues opcions:

- Gastar SectorPoints (punts que pot gastar en producte que no són nous) o no amb llibres de segona mà que hi ha al carretó.
- Recollida a la tenda o enviament al domicili.

Finalment es demanarà confirmació per a les dades escollides i es podrà confirmar o rebutjar.

Com hem dit, també existeix la possibilitat de que un client registrar acumuli punts anomenats *SectorPoints*. La forma d'aconseguir *SectorPoints* és senzilla. Un client pot enviar per correu ordinari o dur-ho personalment al negoci, un llibre usat, per tal que la *BookSector* el posi a la disposició d'altres clients que el vulguin comprar a un preu més reduït. Si es vol enviar el llibre per correu, com a pas previ, el client haurà d'entrar a l'espai dedicat a tal efecte a la pàgina web i sol·licitar el servei. L'aplicació li donarà l'opció d'imprimir un etiqueta amb totes les dades d'enviament per poder enviar el paquet amb el llibre de forma fàcil i ràpida. Quan *BookSector* rebí el llibre, serà revisat pel propietari o empleat, i se li assignaran *SectorPoints* al client, depenent de l'estat i de la qualitat del llibre.

El client podrà comprar llibres de segona mà. El catàleg indica quants d'exemplars nous hi ha d'un llibre en concret i quants de segona mà. Si es tria l'opció de comprar-ho de segona mà, el client podrà gastar *SectorPoints* per pagar-ho, en lloc de diners reals, si així ho desitja.

El projecte, també va orientat a la gestió del catàleg per part del propietari o d'algun empleat. Així doncs, es proporcionarà un sistema de gestió de la web fàcil, gràfic i molt intuïtiu, amb el que es podran fer varies coses com:

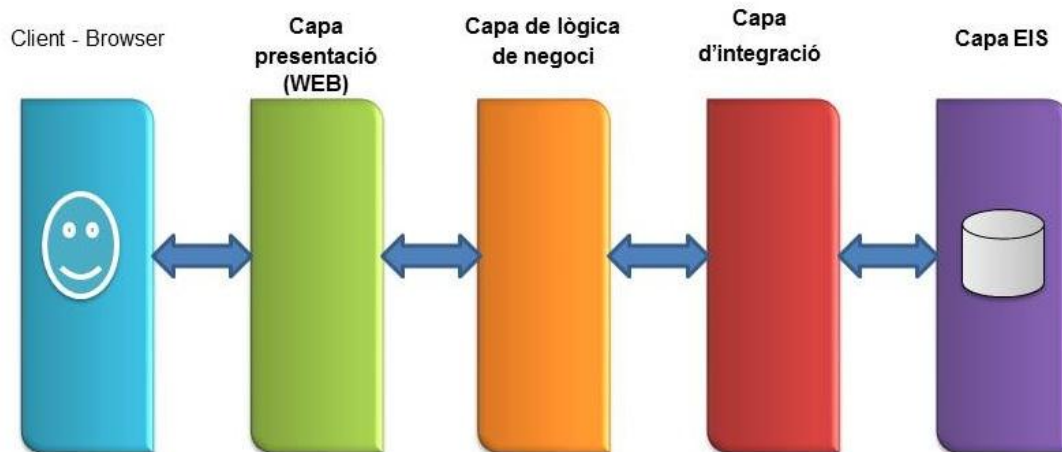
- ▶ Afegir, editar, suprimir llibres del catàleg.
- ▶ Administrar les comandes realitzades pel client.
- ▶ Editar i suprimir la informació dels clients registrats al sistema.
- ▶ Gestionar els punts que un usuari registrar té assignats.
- ▶ Afegir usuaris nous.

Aquesta és doncs, l'aplicació que ens demana el propietari de *BookSector*, i amb la que basaré el projecte.

#### 4.4. Sistema basat en n-capes

El projecte pretén obtenir una aplicació distribuïda client servidor, seguint l'especificació les tecnologies de la J2EE, seguint el paradigma de la programació orientada a objectes i seguint l'estructura en capes de n-nivells que ens ofereix una seguretat, escalabilitat i robustesa que són crucials a l'hora de crear aplicacions web d'aquest tipus.

El disseny per n-nivells està directament relacionada amb l'especificació i les tecnologies J2EE de Java. Serà interessant doncs, definir aquestes capes per tenir clar com es distribuiran les classes de les que es compondrà l'aplicació:



Imatge 2: Diagrama de capes simple

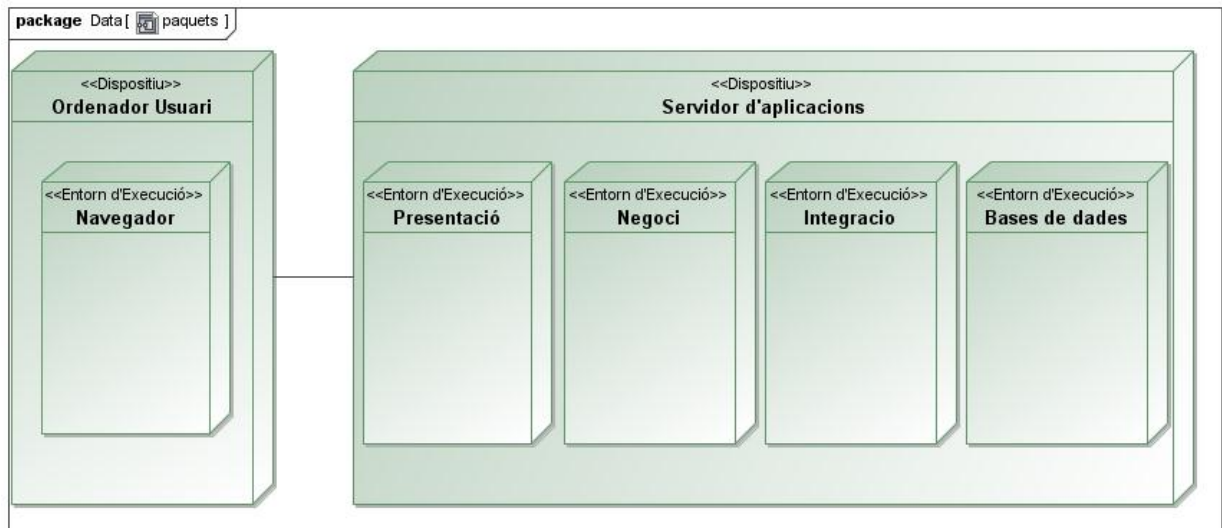
#### 4.5. Màquines i contenidors

J2EE ens proposa una estructura en capes en n-nivells, d'aquesta forma agrupem uns components, que són peces de programari semblants en contenidors, que permeten l'accés i les funcionalitats que ens ofereixen els components un com s'han desplegat en ell. Aquests contenidors ens brinden varies avantatges, ja que implementen operacions de baix nivell de seguretat, accés, entre altres i per tant ens podem dedicar a la lògica dels components. Es troben en un servidor d'aplicacions i fan que les classes de les quals es componen, col·laborin entre elles per arribar a la funcionalitat desitjada.

Aquests contenidors poden estar distribuïts de varies formes: en màquines físiques diferents, en la mateixa, en la mateixa màquina física però executant-se en diferents màquines virtuals...

En el nostre cas, seguirem una estructura com aquesta:





**Imatge 3: Diagrama de contenidors**

Com podem veure, es distingeixen dues màquines físiques diferenciables, per una banda la màquina del client o l'usuari, i per l'altre, el servidor d'aplicacions:

- ▶ L'usuari, al control del seu ordinador, com per exemple un PC, executarà el seu navegador i accedirà a les vistes dels diferents components de l'aplicació, com al servei de login, al carretó, a mirar les comandes, explorar el catàleg, etc... L'usuari accedeix als serveis de forma remota, cosa que gestiona el seu navegador presentant-li les pàgines JSP que es generaran i iniciant les peticions que el client executi mitjançant la interfície de la pàgina en la que es trobi. La tasca de la màquina client acaba en aquest punt.
- ▶ El servidor d'aplicacions pot ser una o més màquines físiques, on s'executin una o més màquines virtuals per fer córrer els serveis dels contenidors. En el nostre cas tenim una màquina física que fa de servei d'aplicacions, i a dins trobem tots els altres contenidors que serien les diferents capes de les que s'ha parlat. Presentació, lògica de negoci, integració i EIS amb les bases de dades.

## 4.6. Subsistemes i paquets

És necessari distingir les funcionalitats de l'aplicació, i detallar-les, però primer és necessari distingir els subsistemes que podem trobar i els paquets que aquests formaran.

### 4.6.1. Subsistemes

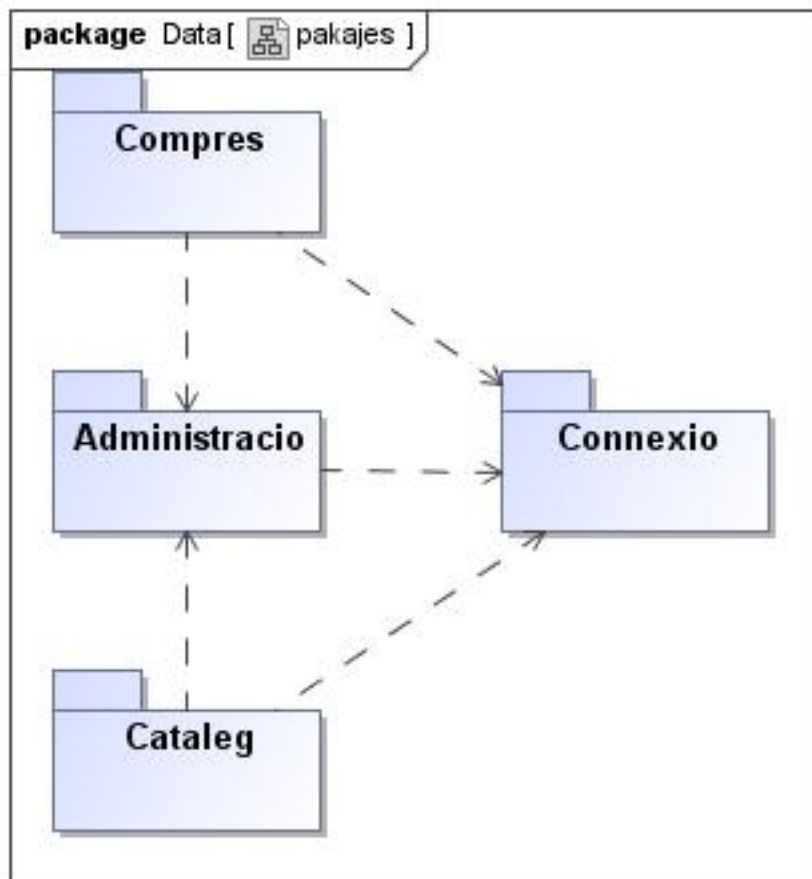
En la petició que fa el client, veiem que les funcionalitats que ha de tenir el projecte són vàries. Tanmateix les podem agrupar en unes entitats amb integritat lògica, els Subsistemes. Els subsistemes són conjunts de classes que col·laboren entre elles per complir un conjunt de responsabilitats.

Podem distingir 4 subsistemes majoritaris, amb les següents característiques:

- **Subsistema de connexió i registre:** s'encarrega de la part relacionada amb la connexió dels usuaris, tant clients com administradors. Gestiona la validació dels usuaris, controlant les sessions iniciades, la seguretat d'aquestes. També s'encarregarà dels registres d'usuaris client nous.
- **Subsistema de compra:** de fet, aquest subsistema el podríem anomenar subsistema de carretó, però queda més representada la seva funcionalitat així. És el subsistema encarregat de les gestions de compra del client. La implementació de tot el sistema de carretó de compra, del tipus de pagament entre altres, es troba dins aquest subsistema. També hi trobem el sistema de comandes, ja que al acabar el procés amb el carretó de compra, possiblement la següent passa sigui validar la comanda de llibres que es vol obtenir.
- **Subsistema de catàleg:** és el subsistema en el qual trobem les implementacions del sistema de catàleg. El catàleg de llibres de *BookSector* es genera a partir d'una base de dades relacional, on l'administrador afegeix els llibres i els gestiona mitjançant un altre subsistema que s'explicarà a continuació. En la capa de presentació, en el catàleg web, es veuen reflectides aquestes gestions que l'administrador fa. Mitjançant el catàleg, un client pot fer una cerca apurada de llibres, amb varis criteris. El catàleg generarà el filtre adient a cada tipus de cerca i mostrarà ràpidament el resultat al client.
- **Subsistema d'administració:** l'administrador del negoci, o bé el propietari o un empleat, poden voler afegir, modificar o esborrar dades del sistema, i per això, trobem un conjunt de funcionalitats dedicades a això. Aquest subsistema ofereix funcionalitats com administrar usuaris registrats, gestionar els llibres del catàleg o gestionar les comandes que s'han realitzat mitjançant el subsistema de compra.

#### 4.6.2. Paquets

Els paquets de programari podrien quedar distribuïts d'aquesta que presentem en la figura següent, seguint l'estructura en subsistemes. Malgrat això, l'estructura de fitxers no segueix un sistema per paquets per subsistema, si no més bé una estructura per tipus de capa o nivell. Igualment és bo presentar aquest diagrama on apareixen un "paquets virtuals" per veure com es relacionen aquests grups de funcionalitats descrits en el punt anterior.



Imatge 4: Diagrama de Subsistemes

#### 4.7 Funcionalitats

Anem a fer una especificació textual en format de llista per enumerar de les funcionalitats que s'agrupen dins els subsistemes que hem definit en el punt anterior:

- **Subsistema de Connexió:**
  - Validació de l'usuari.
  - Control de privilegis per entrar als diferents subsistemes.
  - Control de la sessió.
  - Enregistrament d'un nou usuari
  - Accés d'usuari anònim
- **Subsistema de Catàleg:**
  - Generar el catàleg dels llibres ubicats en la persistència de la base de dades relacional que s'utilitzarà.
  - Permetre cerques per:
    - Nom del llibre
    - Autor

- Categoria
- ISBN
- Mostrar cada llibre amb la seva fitxa tècnica i amb fotografies d'aquest si escau.
- Sol·licitar enviament de Llibre de segona mà.
- **Subsistema d'Administració:**
  - Gestionar els usuaris creats
  - Eliminar usuaris
  - Crear nous usuaris
  - Gestionar els permisos dels usuaris existents
  - Assignar o llevar *SectorPoints* a usuaris
  - Administrar els llibres que apareixen al catàleg:
    - Afegir-Editar-Eliminar llibres nous
    - Afegir-Editar-Eliminar llibres de segona mà
  - Gestionar l'estat de les comandes existents, permetent canviar l'estat d'aquestes.
- **Subsistema de Compra:**
  - Control del carretó de compra:
    - Afegir llibre
    - Suprimir llibre
    - Inspeccionar llibre
    - Buidar carretó
    - Calcular subtotal
    - Gestionar pagament per *SectorPoints*
  - Triar mètode d'entrega
  - Triar tipus de cobrament
  - Mostrar confirmació de compra

## 4.8 Actors

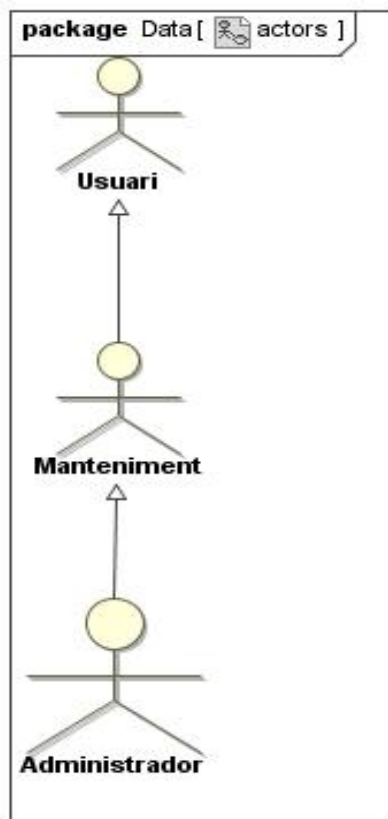
A continuació identificarem els diferents actors i les possibles relacions entre ells:

- ▶ **Administrador:** és l'entitat amb màxima responsabilitat i drets per a manipular la informació de tots els subsistemes. Serà l'encarregat de gestionar els usuaris del sistema, els permisos d'aquests, assignar *SectorPoints*, administrar en la seva totalitat el catàleg i gestionar les comandes, així com totes les accions de

qualsevol client registrat. Segurament aquest càrrec el tindria el propietari del negoci, al tractar-se d'una empresa petita.

- ▶ **Manteniment:** el personal de manteniment tenen els mateixos drets que l'administrador, però a diferència d'aquest, no poden administrar les comptes d'usuari. El personal de manteniment són actors auxiliars al personal d'Administració per tal que tota la càrrega de feina no pesi sobre l'administrador.
- ▶ **Usuari:** es tracta de l'usuari de l'aplicació com a tal. Serà el client que accedirà a l'aplicació web amb el navegador, es validarà, farà compres de llibres i enviarà llibres usats al negoci per rebre *SectorPoints*.

Existeixen relacions d'especialització entre aquestes tres actors, ja que el personal de **Manteniment** té els mateixos drets que l'**Usuari**, mentre que l'**Administrador** té els mateixos drets que el personal de **Manteniment**. Per tant partirem de la premissa d'aquest senzill diagrama.



**Imatge 5: Diagrama d'herència Actors**

Igualment, cal dir que aquest diagrama es refereix a les accions reals que es fan durant l'execució de l'aplicació, ja que, com veurem més endavant, en el model de classes, aquestes relacions d'herència són eliminades per motius que més endavant descriurem. Així que com hem dit, aquestes relacions es refereixen als drets que cada tipus d'actor té depenen del seu rol.

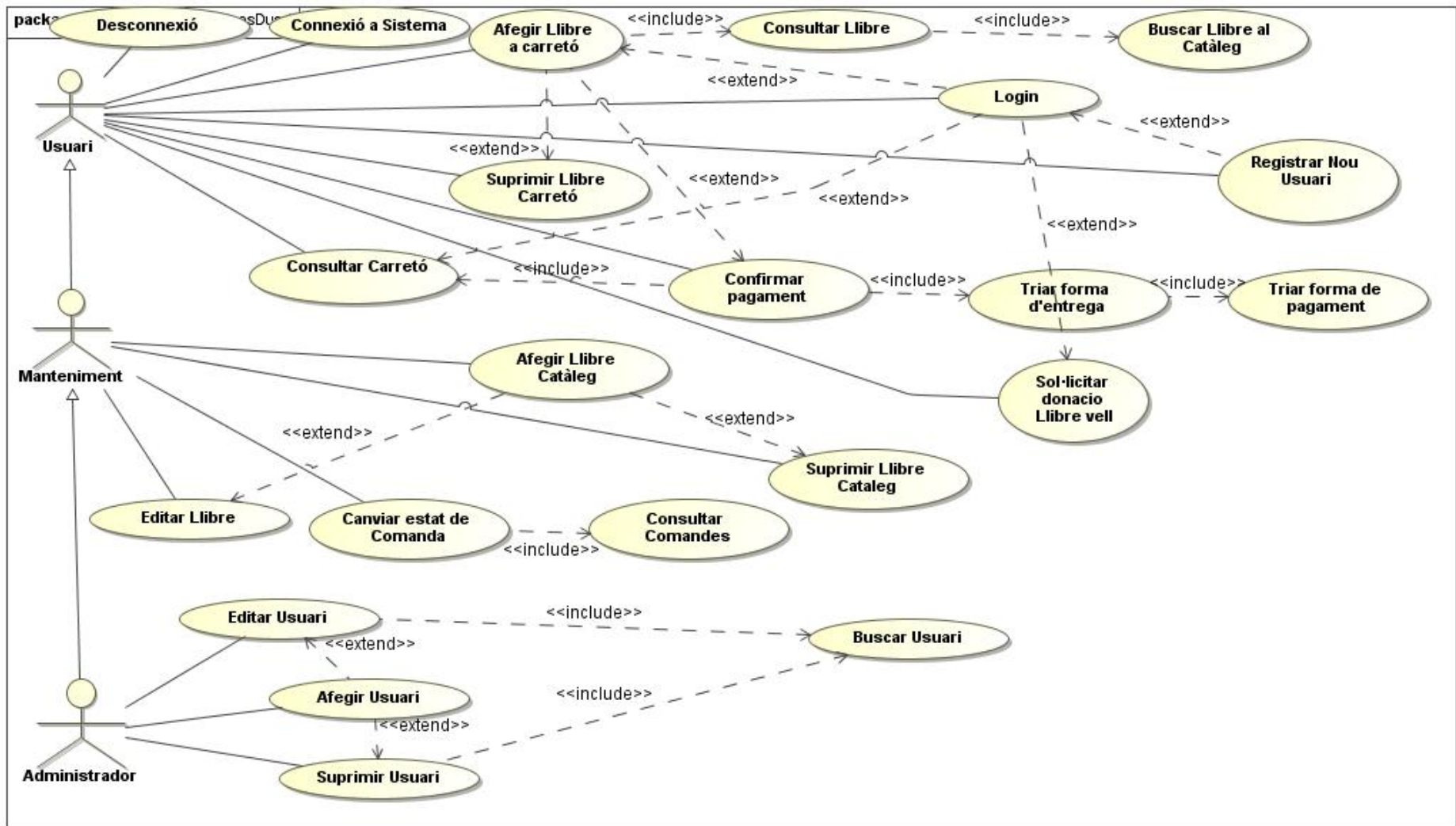
## 4.9 Especificació textual dels casos d'ús

Serà necessària identificar els diferents casos d'ús que podrem torbar durant el funcionament de l'aplicació per part dels actors. A continuació veurem un diagrama general dels actors i els casos d'ús pel que fa al funcionament de l'aplicació a nivell de lògica de negoci. Llavors veurem llistat de casos d'ús, amb les seves característiques.

### 4.9.1 Diagrama de casos d'ús

A la pàgina següent es pot veure el diagrama de casos d'ús general, amb tots els actors del sistema i els seus casos d'ús. S'ha de dir que hi ha casos d'ús de més baix nivell i que no he inclòs en el diagrama, ja que aquest sortiria excessivament gran si haguessin de sortir fins als més insignificants casos d'ús.

(El diagrama es presenta el la següent plana per qüestions d'espai)



Imatge 6: Diagrama casos d'ús

Sobre el diagrama anterior, s'ha de concretar que, tots els casos d'ús que el personal de manteniment i el administradors fan requereixen sempre del cas d'ús "Login", ja que es tracta d'accions administratives que només el personal autoritzat pot fer.

#### 4.9.2 Descripció i detall dels casos d'ús

Farem un llistat de casos d'ús en format de taula i ho farem per subsistema. El llistat de casos d'ús és el següent:

##### 4.9.2.1 Subsistema de Connexió

<b>Nom</b>	<b>Connexió</b>
<b>Resum</b>	L'usuari accedeix a l'URL especificada pel programa, en el que se li presenta la pàgina principal. Es tracta d'un accés com a usuari anònim.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	No hi ha
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ha indicat al navegador la URL apropiada per accedir a la pàgina.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• L'actor està validat i disposa de certs drets com a usuari anònim.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. L'actor accedeix a l'URL amb el seu navegador web per accedir a la pàgina</li> <li>2. La pàgina es carrega amb els privilegis mínims com a usuari no-registrat.</li> </ol>
<b>Flux alternatiu</b>	

Taula 2: CU Connexió

<b>Nom</b>	<b>Desconnexió</b>
<b>Resum</b>	L'usuari independentment del Rol que utilitzi, finalitza la sessió i es desconnecta de la pàgina web de presentació del sistema.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, amb validació o sense validació.</li> </ul>



<b>Post-condició</b>	<ul style="list-style-type: none"> <li>L'actor no està connectat amb els servidors i no disposa de cap vista generada per la pagina web</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>L'actor es desconnecta amb l'opció del menú de la pantalla principal amb el nom "Tanca sessió", finalitzant així tota sessió activa com a usuari registrat.</li> <li>L'actor tanca la finestra del navegador per tal d'acabar la connexió amb el servidor web.</li> </ol>
<b>Flux alternatiu</b>	<ol style="list-style-type: none"> <li>L'actor, si és un usuari anònim que no validat, pot tancar la finestra del navegador web directament.</li> </ol>

Taula 3: CU Desconnexió

<b>Nom</b>	<b>Login</b>
<b>Resum</b>	L'actor que ha accedit al sistema, es valida amb les seves dades d'usuari per tal de poder fer compres i altres accions, depenent del seu Rol.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema com a usuari anònim.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>L'actor s'ha d'haver connectat abans, sense validació i com a usuari anònim.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>L'actor ja no utilitza un usuari anònim i disposa de drets i privilegis que depenen del Rol utilitzat amb el seu usuari.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>L'actor accedeix al petit formulari de la dreta de la pàgina principal de presentació i emplena el camp <i>Usuari</i> i <i>Contrasenya</i> i a continuació prem el botó <i>login</i> per realitzar la validació.</li> </ol>
<b>Flux alternatiu</b>	<ol style="list-style-type: none"> <li>L'actor pot anar a l'opció <i>Login</i> del menú, per tal que la validació se li presenti si encara no s'ha registrat. També es pot donar el cas que l'usuari sense privilegis, intenti accedir a una zona restringida a altres privilegis, i en aquest cas se li presentarà una pantalla de Login per tal de validar-se amb un altre usuari que si que tingui els drets necessaris.</li> </ol>

Taula 4: CU Login

<b>Nom</b>	<b>Registrar Nou Usuari</b>
------------	-----------------------------

<b>Resum</b>	L'actor amb usuari anònim pot introduir les seves dades per obtenir un usuari amb drets per realitzar compres de llibres, entre altres accions.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema com a usuari anònim.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, sense validació i com a usuari anònim.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ja té un usuari amb el qual fer el <i>Login</i> i que li donarà privilegis per fer compres de llibres amb les noves credencials</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. L'actor anònim va a l'opció del menú <i>Registra't</i>.</li> <li>2. En la finestra nova que s'obtindrà, disposarà d'un formulari on posar les seves dades, entre les que es troben el nom d'usuari que desitja tenir, la contrasenya que utilitzarà per fer el login, el correu electrònic utilitzat, adreces d'on resideix l'actor, etc...</li> </ol>
<b>Flux alternatiu</b>	

Taula 5: CU Registre Nou Usuari

#### 4.9.2.2 Subsistema de Catàleg

<b>Nom</b>	<b>Buscar Llibre al Catàleg</b>
<b>Resum</b>	L'actor connectat al sistema, pot utilitzar diferents sistemes de recerca de llibres amb les opcions que li presenta el catàleg, per torbar un llibre o un grup d'ells
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema com a usuari anònim o registrat.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, amb validació o sense validació (com a usuari anònim).</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• L'actor veu per pantalla el llibre o el conjunts d'ells que concorden amb els paràmetres de recerca que ell ha especificat.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. A la pantalla principal, disposa de varis selectors per triar el mètode de recerca, i un camp de text per escriure caràcters relacionats amb la cerca.</li> <li>2. Un cop escollits els camps anteriors, pot prémer el botó "cercar".</li> <li>3. Per pantalla es veurà un llistat dels resultats de la cerca (cap, un o més llibres)</li> </ol>
<b>Flux alternatiu</b>	<ol style="list-style-type: none"> <li>1. Si així ho desitja, l'actor pot no triar cap mètode de i deixar aquesta</li> </ol>

	<p><i>select</i> en blanc, de forma que el criteri de recerca es farà per qualsevol coincidència amb els caràcters escrits. Per exemple, l'actor pot escriure les paraules "Tolkien" i els resultats poden ser llibres escrits per aquest autor, o bé llibres que en el seu títol tenen la paraula Tolkien, com per exemple la biografia d'aquest autor.</p>
--	--

Taula 6: CU Buscar Llibre al Catàleg

<b>Nom</b>	<b>Consultar Llibre</b>
<b>Resum</b>	L'actor pot prémer sobre la icona "Fitxa", i se li mostrarà per pantalla la fitxa amb tota la informació del llibre seleccionat.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema com a usuari anònim o registrat.</li> <li>• Buscar Llibre al catàleg.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, amb validació o sense validació (com a usuari anònim).</li> <li>• L'actor ha fet la recerca de llibre mitjançant el cas d'ús Buscar Llibre al catàleg</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• Se li ha presentat la fitxa del llibre per pantalla amb tots els detall d'aquest.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. L'actor prem sobre la icona "Fitxa" del llibre objectiu.</li> <li>2. S'obre una pantalla amb totes les dades per consultar</li> </ol>
<b>Flux alternatiu</b>	

Taula 7: CU Consultar Llibre

<b>Nom</b>	<b>Sol·licitar donació llibre vell</b>
<b>Resum</b>	L'actor pot decidir donar a l'empresa <i>BookSector</i> un llibre seu usat per tal d'obtenir <i>SectorPoints</i>
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema.</li> <li>• Fer validació com a usuari registrat.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ha obtingut una etiqueta per fer l'enviament del llibre en qüestió.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. L'actor prem sobre l'opció <i>Donació</i> del menú de la pantalla</li> </ol>

	<p>principal.</p> <p>2. Se li obre una pantalla amb les seves dades d'usuari, i les dades de la tenda BookSector.</p> <p>3. L'usuari podrà utilitzar les eines del seu navegador web per a imprimir l'etiqueta.</p>
<b>Flux alternatiu</b>	

Taula 8: Sol·licitar donació llibre vell

#### 4.9.2.3 Subsistema de Compra

<b>Nom</b>	<b>Afegir Llibre al Carretó</b>
<b>Resum</b>	Al haver triat un llibre es dona l'opció de comprar-lo nou, o de segona mà si hi ha existències de segona mà. L'actor pot afegir-ho al carretó de compra si ha iniciat sessió amb el seu usuari.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema.</li> <li>• Fer validació com a usuari registrat.</li> <li>• Buscar llibre al catàleg</li> <li>• Visualitzar Fitxa del llibre</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>. També és necessari que abans hagi cercat el llibre desitjat mitjançant el buscador del catàleg i hagi visualitzar la seva fitxa.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• El llibre s'ha afegit al carretó de compra del client.</li> </ul>
<b>Flux normal</b>	1. Triar opció "Afegir al carretó" que es mostra a la fitxa del llibre.
<b>Flux alternatiu</b>	

Taula 9: CU Afegir Llibre al Carretó

<b>Nom</b>	<b>Consultar carretó</b>
<b>Resum</b>	L'usuari registrat accedeix a veure el contingut del seu carretó on es mostra el subtotal i els llibres que ha afegit.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema.</li> <li>• Fer validació com a usuari registrat.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>

<b>Post-condició</b>	<ul style="list-style-type: none"> <li>L'usuari visualitza el contingut del carretó, que conté els llibres triats, el subtotal desglossat, entre altres.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>Triar opció "carretó" que es mostra en el menú de la pantalla principal de l'aplicació, que té forma d'una imatge de carretó.</li> </ol>
<b>Flux alternatiu</b>	

Taula 10: CU Consultar carretó

<b>Nom</b>	<b>Suprimir Llibre del Carretó</b>
<b>Resum</b>	L'usuari elimina un llibre dels que hi ha al llistat del carretó.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>Connexió: abans d'aquest cas d'ús, cal que l'actor s'hagi connectat abans al sistema.</li> <li>Fer validació com a usuari registrat.</li> <li>Afegir llibre al carretó.</li> <li>Consultar carretó.</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> <li>També ha d'haver afegit algun llibre al carretó de compra</li> <li>S'ha d'entrar abans a consultar el carretó de compra</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>El llibre en qüestió no està al carretó de compra.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>Un cop dins el carretó de compra es selecciona el llibre que es vol eliminar.</li> <li>Es prem el botó <i>Eliminar</i> que surt juntament amb el llibre dins el carretó.</li> <li>El carretó s'actualitza i mostra els llibres que hi havia menys el que s'acaba d'eliminar.</li> </ol>
<b>Flux alternatiu</b>	<ol style="list-style-type: none"> <li>Un cop dins el carretó de compra es prem el botó <i>Buidar Carretó</i> a la part dreta de la pantalla que ofereix el carretó.</li> </ol>

Taula 11: CU Suprimir Llibre del Carretó

(en els propers casos d'ús ometrem el cas d'ús relacionat estar connectat i validat, ja que es dona per suposat)

(el següent cas d'ús en té dos més d'intrínsecs i per tant no es detallaran, però es troben especificats al diagrama de casos d'ús)

<b>Nom</b>	<b>Confirmar comanda</b>
<b>Resum</b>	L'usuari confirma la compra dels llibres dins el carretó
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Usuari</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Triar forma de pagament</li> <li>• Triar forma d'entrega</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> <li>• També ha d'haver afegit algun llibre al carretó de compra</li> <li>• S'ha d'entrar abans a consultar el carretó de compra.</li> <li>• S'ha de triar una forma d'entrega i una forma de pagament.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• La comanda està realitzada i pagada i els llibres triats es troben en camí del domicili del comprador.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. Primer es trien si els possibles llibres de segona mà s'han de pagar amb els <i>SectorPoints</i> dels que disposa l'usuari o si tot el pagament es farà amb diners. Es gastaran tots els punts que sigui possible per a pagar llibres de segona mà i l'opció es tria.</li> <li>2. Es tria el tipus d'entrega que pot ser en el local o a domicili a la direcció que l'usuari té enregistrada amb les seves dades.</li> <li>3. Es prem el botó <i>Realitzar comanda</i>.</li> <li>4. Apareix el cartell de confirmació de la comanda.</li> <li>5. Apareix el resum si la comanda s'ha executat correctament.</li> </ol>
<b>Flux alternatiu</b>	

Taula 12: CU Confirmar comanda

#### 4.9.2.4 Subsistema d'Administració

(aquest cas d'ús té un altre cas d'ús intrínsec en ell: **Consultar Comandes**.)

<b>Nom</b>	<b>Canviar estat de comanda</b>
<b>Resum</b>	El personal de manteniment (i també l'administrador per herència) entren al llistat de comandes realitzades pels clients i canvien l'estat d'aquesta. Els estats poden ser: pendent d'enviar, enviada, rebuda i cancel·lada.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Manteniment</li> </ul>

<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Consultar Comandes</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ha de ser de Manteniment (o per herència Administrador)</li> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• La comanda seleccionada com a tal, es troba en un estat diferent a la del principi.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. S'entra a la pàgina web de presentació.</li> <li>2. Es va a l'opció <i>Portal Privat</i> del menú principal.</li> <li>3. Es fa un <i>Login</i> (veure cas d'ús)</li> <li>4. A la nova pantalla es tria l'opció <i>Gestió de Comandes</i>. (fins aquí arriba el cas d'ús <i>Consultar Comandes</i>)</li> <li>5. Al costat de cada comanda trobem un botó <i>Editar</i>. Amb ell s'obre el detall de la comanda, amb els productes i el resum de les dades. En el formulari es pot canviar l'estat de la comanda.</li> <li>6. Es guarda el canvi i es torna a la pantalla on surten totes les comandes.</li> </ol>
<b>Flux alternatiu</b>	

Taula 13: CU Canviar estat de comanda

(a partir d'aquí els passos de l'1 fins al 3, es donen per suposat, ja que totes les opcions requereixen entrar al portal privat, i per tant fer el *Login* (veure cas d'ús). I per tant no els tindrem en compte)

<b>Nom</b>	<b>Afegir Llibre al Catàleg</b>
<b>Resum</b>	El personal de manteniment (i també l'administrador per herència) afegeixen un llibre que estarà disponible quan un client consulti el catàleg.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Suprimir Llibre Catàleg</li> <li>• Consultar Catàleg</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ha de ser de Manteniment (o per herència Administrador)</li> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• Hi ha un llibre més en la base de dades.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. Es tria l'opció <i>Gestió de Catàleg</i>.</li> <li>2. En la pantalla nova surt un llistat de tots els llibres dins els catàlegs i es pot emplenar el formulari per a agregar un llibre al catàleg. Entre les dades del llibre hi ha l'opció de triar si es un llibre nou o no, i l'sotck inicial del llibre.</li> </ol>

	3. Es prem el botó Afegir Llibre.
<b>Flux alternatiu</b>	

Taula 14: CU Afegir Llibre al Catàleg

<b>Nom</b>	<b>Suprimir llibre Catàleg</b>
<b>Resum</b>	El personal de manteniment (i també l'administrador per herència) suprimeix un llibre del catàleg
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Afegir Llibre Catàleg</li> <li>• Consultar Catàleg</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ha de ser de Manteniment (o per herència Administrador)</li> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• Hi ha un o varis llibres menys en la base de dades de l'aplicació</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. Es tria l'opció Gestió de Catàleg del menú principal.</li> <li>2. En la pantalla nova surt un llistat de tots els llibres dins els catàlegs amb dos botons per cada un: Eliminar i Editar.</li> <li>3. Es tria l'opció <i>Eliminar</i> del llibre desitjat.</li> </ol>
<b>Flux alternatiu</b>	

Taula 15: CU Suprimir llibre Catàleg

<b>Nom</b>	<b>Editar llibre</b>
<b>Resum</b>	El personal de manteniment (i també l'administrador per herència) accedeix a la llista d'un llibre per editar-ne les seves dades
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>• Manteniment</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>• Consultar Catàleg</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>• L'actor ha de ser de Manteniment (o per herència Administrador)</li> <li>• L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>• Les dades del llibre són diferents a abans de l'acció del cas d'ús.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>1. Es tria l'opció Administrar Catàleg.</li> <li>2. En la pantalla nova surt un llistat de tots els llibres dins els catàlegs amb dos botons per cada un: Eliminar i Editar.</li> <li>3. Es tria l'opció <i>Editar</i> del llibre desitjat.</li> <li>4. Es canvien les dades del formulari que té les dades del llibre</li> <li>5. Es confirmen els canvis amb el botó del formulari.</li> </ol>
<b>Flux alternatiu</b>	

Taula 16: CU Editar llibre



<b>Nom</b>	<b>Buscar Usuari</b>
<b>Resum</b>	Els administradors poden accedir a un llistat d'usuaris i buscar-ne un
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>Administradors</li> </ul>
<b>CU relacionats</b>	
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>L'actor ha de ser de Administrador</li> <li>L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>L'administrador visualitza l'usuari desitjat.</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>Es tria l'opció <i>Gestió d'usuaris</i> del menú.</li> <li>Surt la llista completa dels usuaris actuals</li> <li>Cada usuari té dos botons: Editar i Eliminar.</li> <li>Es mostra el formulari amb les dades de l'usuari, editables.</li> </ol>
<b>Flux alternatiu</b>	

Taula 17: CU Buscar Usuari

<b>Nom</b>	<b>Editat Usuari</b>
<b>Resum</b>	Els administradors accedeixen a la informació dels usuaris i per editar-la
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>Administradors</li> </ul>
<b>CU relacionats</b>	<ul style="list-style-type: none"> <li>Buscar Usuari</li> </ul>
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>L'actor ha de ser de Administrador</li> <li>L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>Les dades de l'usuari són diferents</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>S'entra en el formulari amb les dades de l'usuari amb el cas d'ús <i>Buscar Usuari</i> i mitjançant el botó <i>Editat</i>.</li> <li>Es canvien les dades desitjades, com les dades personals, els <i>SectorPoints</i>, entre altres...</li> <li>Es confirmen els canvis amb el botó del formulari</li> </ol>
<b>Flux alternatiu</b>	

Taula 18: CU Editat Usuari

<b>Nom</b>	<b>Afegir Usuari</b>
<b>Resum</b>	L'administrador afegeix un usuari nou de forma manual
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>Administradors</li> </ul>
<b>CU relacionats</b>	
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>L'actor ha de ser de Administrador</li> </ul>

	<ul style="list-style-type: none"> <li>L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>Hi ha un usuari més en el sistema</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>En la pantalla de Gestió d'usuaris que apareix en prémer el botó Gestió d'Usuaris apareix un formulari en blanc per a guardar un nou usuari amb totes les dades necessàries.</li> <li>Entre elles s'especifica quin tipus d'usuari és.</li> <li>Es confirmen els canvis amb el botó del formulari.</li> </ol>
<b>Flux alternatiu</b>	

Taula 19: CU Afegir Usuari

<b>Nom</b>	<b>Suprimir Usuari</b>
<b>Resum</b>	L'administrador elimina un usuari de la BBDD del sistema.
<b>Actor(s)</b>	<ul style="list-style-type: none"> <li>Administradors</li> </ul>
<b>CU relacionats</b>	
<b>Pre-condició</b>	<ul style="list-style-type: none"> <li>L'actor ha de ser de Administrador</li> <li>L'actor s'ha d'haver connectat abans, i haver-se validat mitjançant el cas la funcionalitat <i>Login</i>.</li> </ul>
<b>Post-condició</b>	<ul style="list-style-type: none"> <li>Hi ha un usuari menys en el sistema</li> </ul>
<b>Flux normal</b>	<ol style="list-style-type: none"> <li>En la pantalla de Gestió d'usuaris es selecciona un usuari de la llista al qual es vol eliminar.</li> <li>Es prem el botó <i>Eliminar</i> de l'usuari seleccionat.</li> <li>L'usuari es eliminat</li> </ol>
<b>Flux alternatiu</b>	

Taula 20: CU Suprimir Usuari

#### 4.9.3 Aclariments sobre els casos d'ús

Actualment l'aplicació manca de dues funcionalitats:

- **Cancel·lar una comanda:** per cancel·lar una comanda el client s'haurà de posar en contacte per mitjà telefònic o bé per e-mail, per tal que la seva comanda sigui cancel·lada abans de que aquesta sigui enviada.
- **Consultar comandes:** No hi ha un mòdul per consultar les comandes que s'han fet i el seu estat, per part del client. Malgrat això, es preveu que els administradors i personal de manteniment facin un seguiment via e-mail de les comandes als seus clients. Quan una comanda sigui realitzada o enviada, s'enviarà un avís als clients per tal que tinguin constància que la seua comanda s'ha realitzat correctament.

#### 4.10 Requisits per a l'execució del programa.

- **La màquina Servidor:**
  - He de suportar una JVM v1.6 com a mínim per a executar les aplicacions en Java.
  - Base de dades MySQL 5.0
  - Servidor Apache Tomcat 7.0.
  - Suport de divers JDBC i de l'HIBERNATE
  - Processador Intel Pentium III o equivalent.
  - 2 GB de RAM
  - 4 GB d'espai lliure en disc.
  - Connexió internet ADSL o equivalent a velocitat mínima de 1 MB.
- **La màquina Client:**
  - Connexió internet ADSL o equivalent a velocitat mínima de 1 MB.
  - Navegador tipus NetScape, Internet Explorer, Google Chrome, Firefox, Safari, etc... que suporti la darrera versió d'HTTP.
  - És convenient que es tingui instal·lada la JVM de Java.
  - Sobre la hardware, és necessari que la màquina pugui executar la màquina virtual de Java i el navegador sense problemes.

#### 4.11 Possibles implementacions futures

Aquesta és una llista de possibles modificacions i agregacions que la versió actual podria sofrir en un futur:

- Seguiment de les comandes per part de l'usuari que l'ha realitzada.
- Possibilitat de cancel·lar comandes realitzades durant les primeres 18 hores.
- Mòdul d'edició de dades personals per part de l'usuari
- Pagament per PayPal o amb tarja de crèdit.
- Paginació per els resultats de cerca del catàleg.
- Paginació pels llistats de llibres i usuaris de l'administració.
- Sistema multi-tenda, en el cas que BookSector s'expandís a més d'una tenda.

## 5 Disseny del Projecte

### 5.1 El perquè del document de disseny

El document d'especificació i anàlisi de requeriments que s'ha confeccionat pretenia fer un recull de les funcionalitats que havia de tenir el programa i oferir una visió més àmplia del projecte en general.

El document de disseny pretén acostar aquestes especificacions de requeriments prèvies a la implementació i per tant amb ell intentarem aconseguir mostrar com podem oferir totes les funcionalitats i especificacions que s'han mostrat a l'altre document.

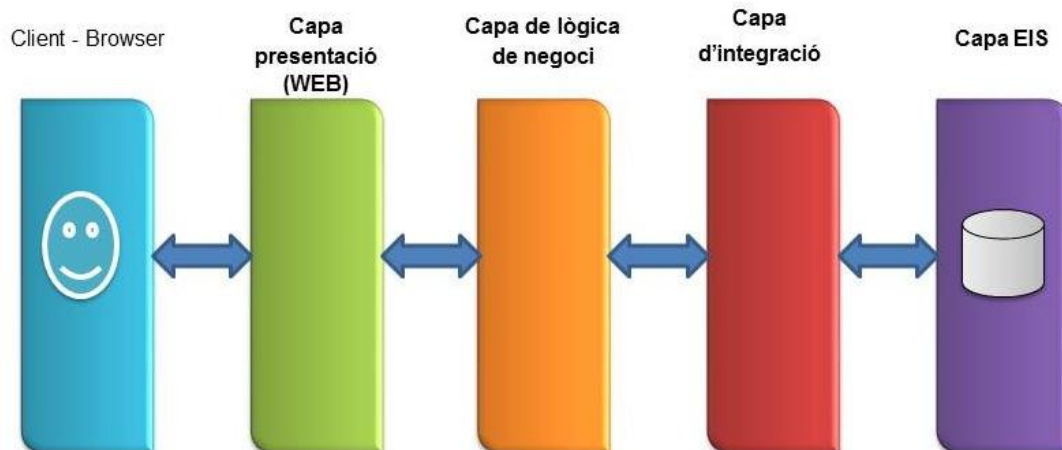
Es detallaran detalls més precisos i tècnics, les tecnologies a utilitzar i diagrames i esquemes que ja fan referència a quin camí seguirà la implementació del programa.

### 5.2 Plataforma/es i tecnologia

En la capa d'especificació ja es van fer mencions sobre la tecnologia a utilitzar per a la implementació del projecte. Ara s'aprofundirà en aquest tema d'una forma més concisa.

#### 5.2.1 J2EE Multicapa

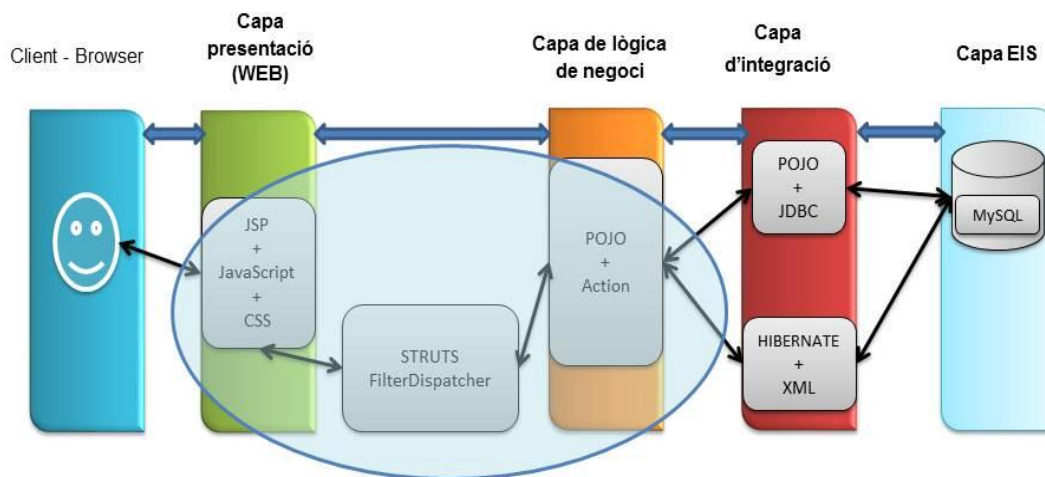
Com dèiem, la intenció és utilitzar l'arquitectura proposada per la J2EE, com a model arquitectònic basat en diverses capes, cadascuna amb uns objectius i funcions molt marcats. Ja es va fer un esquema previ sense detallar en la fase d'especificació, i era el següent:



**Imatge 7: Diagrama de Capes simple 2**

Aquesta separació per capes proposada per la J2EE, permet obtenir bons nivells d'escalabilitat, flexibilitat i també preservar les avantatges que la POO ens ofereix, com la seguretat, l'encapsulament, la reutilització, etc...

Ara es el moment d'ampliar una mica el gràfic anterior i especificar una mica més què passa en cada capa.



**Imatge 8: Diagrama de capes complex**

En el cercle de color cel, s'han encerclat les entitats que Struts2 Gestiona en les tres capes: Vista, Controlador i Model (d'esquerra a dreta)

- **Capa client:** Per client entenem qualsevol dels actors especificats en el document *Especificació i Anàlisi de Requeriment*. Així doncs, la capa client té marcat el funcionament per les accions d'aquest client amb el navegador WEB de la seva màquina client. En el document *Especificació i Anàlisi de Requeriment* es va mostrar com es distribueixen les capes en relació a les màquines que intervenen en el funcionament del sistema. Així la màquina client visualitzarà, mitjançant el seu navegador web, les interfícies gràfiques que la capa de presentació (i concretament capa web) li presenti, mitjançant JSP's i Servlets gestionat tot per Struts 2.
- **Capa de Presentació (web):** La capa de presentació presenta al client o usuari final, els resultats o peticions que es requereixen en cada moment. S'estableix una comunicació remota entre el navegador web de la capa client i la capa de presentació web. Al ser una capa de presentació, connecta amb la capa de la lògica de negoci per tal de fer-se amb les dades que al final ha de mostrar al client.

Com podem veure en el gràfic, s'utilitzen *Java Server Pages* (JSP) per a la presentació visual davant el client, juntament amb *Javascript* i *CSS* per a complementar aquestes pàgines de la capa web. Aquesta capa es correspondria amb la secció de *View* de l'estructura Struts2. Aquestes pàgines JSP són el resultat de les accions que crida Struts2 depenent de la petició del client.

Struts 2 és un *framework* de presentació basat amb el patró MVC (Model View Controller) el qual brinda una gran capacitat de configuració i extensibilitat. A més permet la utilització de plugins de components.

Per això, en la nostra aplicació, on hi ha multitud de pàgines on hi apareixen formularis que s'ha de validar, taules, etc... i que a més són dinàmics, Struts serà idoni per a la implementació d'aquests requeriments.

- **Capa de lògica de negoci:** Aquesta capa conté les classes que implementen el funcionament intern de l'aplicació. Trobem les classes que implementen realment les funcionalitats que generen resultats, es serveixen de les dades emmagatzemades en disc (utilitzant algun tipus de persistència i la capa d'integració), i en definitiva s'executen de forma transparent al client, que finalment rep els resultats de l'execució dels mètodes de les classes de la capa de la lògica de negoci. En el nostre cas per a les classes d'aquesta capa, Struts2 cridarà a unes classes que implementen les anomenades "accions" per a la implementació de les funcionalitats i com a resposta de les peticions del client. Aquestes classes hereten el comportament d'*ActionSupport* i es complementen amb l'ajuda de POJO clàssic que, en el nostre cas, i com veurem més endavant, també tenen altres funcionalitats.

En l'especificació Struts2 aquesta correspon a la capa de model.

L'especificació J2EE presenta els Struts2 com la millor opció a l'hora d'implementar les funcionalitats de capes amb el model MVC. Malgrat això, no és l'única opció per tant també podríem fer ús de POJO conjuntament amb les EJB de sessió. Però vist la bona gestió que fa Struts2 de les diferents capes i seccions, si no és necessari, no farem ús d'aquests components.

- **Capa d'integració:** en aquesta capa trobem les classe, estructures i sistemes encarregats de la connexió amb la base de dades i el control de persistència per a les dades que s'han de conservar durant un temps indeterminat, però que no es poden perdre durant les sessions de funcionament del programa.

Per a implementar la persistència utilitzarem:

- **Hibernate + XML:** Hibernate és un eina de mapeig objecte – relacional (tècnica de representació de dades d'un model d'objectes a un model de dades relacionals persistents) per a programació en Java. S'encarrega de relacionar d'una forma molt directe i intuïtiva els objectes instanciats de les classes Java amb la seva taula corresponent de la BBDD, fent possible noves insercions, eliminacions i modificacions de les dades al nivell de persistència a mesura que ho necessitem i durant el seu cicle de vida. Encarregant-se d'aquesta forma de multitud de tasques relacionades amb la persistència de dades, manejant tots els problemes relatius amb la base de dades de manera transparent per al desenvolupador.

Hibernate permet fer aquest mapeig de dues formes diferenciades. Per una banda amb fitxers XML per a cada classe que es vol mapejar amb la base de dades, i per altra mitjançant anotacions *Java* directament sobre els atributs i mètodes de les classes que es volen representar en el model relacional de la Base de Dades.

- **Capa EIS:** en aquesta capa trobem la base de dades en si mateixa. Els accessos i les accions es faran sobre una base de dades MySQL Server, fent servir, com hem dit, Hibernate per a controlar l'accés i la interacció amb les dades.

### 5.3 Hibernate

Hibernate és una eina de software lliure amb llicència GNU LGPL, una eina de mapeig objecte-relacional. Facilita el mapeig d'atributs entre la base de dades relacional i el model d'atributs dels objectes Java d'una aplicació. Aquesta relació es declara mitjançant arxius XML i anotacions en les classes, més referències a certes llibreries d'Hibernate que s'obtenen en forma de *jar*.

Hibernate convertirà les dades d'un model a l'altre mantenint la portabilitat entre els dos motors de bases de dades només amb un petit increment del temps d'execució, que per aplicacions com aquesta serà menyspreable.

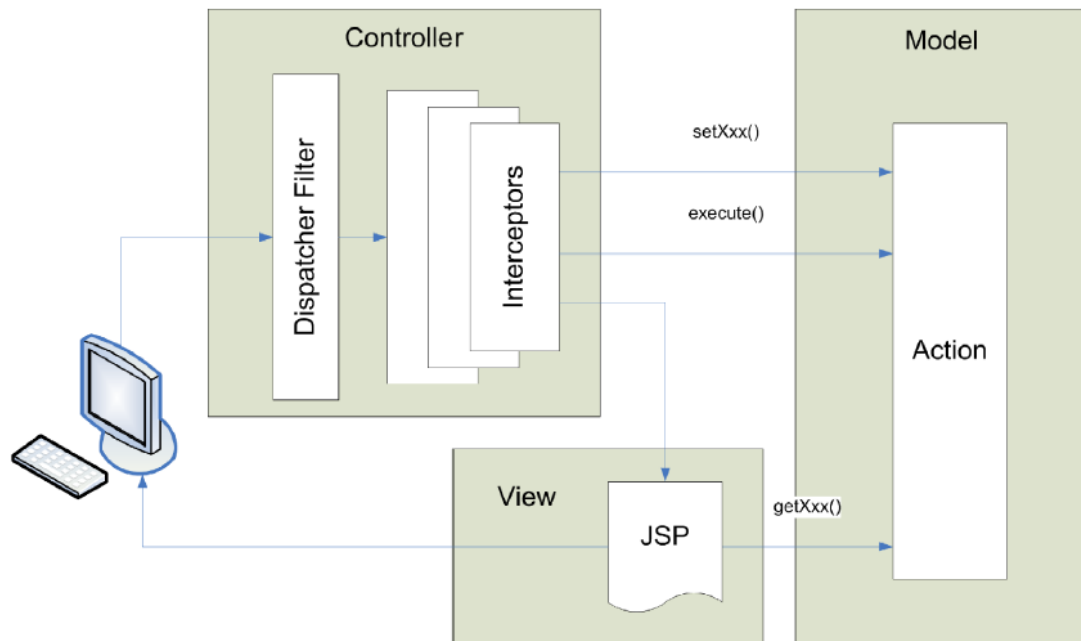
A més, un canvi de SGBD serà molt senzill, ja que tota la configuració externa al codi *Java* que el programador utilitza, es farà sobre fitxers XML. És a dir, les classes *Java* que interactuen amb la BD, ho fan de forma genèrica, i és Hibernate l'encarregat de fer la connexió amb la base de dades i generar les sentències SQL que corresponen. Seguint amb l'exemple, canviar de SGBD seria tan senzill com modificar un grapat de línies en un fitxer de configuració XML.

### 5.4 Struts

Com dèiem abans, Struts 2 està basat en el patró MVC (Model – Vista – Controlador). Això busca reduir l'acoblament dividint les responsabilitats e 3 capes:

- **El model:** fa referència a les dades que l'aplicació utilitza i les regles de negoci. Podríem dir que es tracta de les accions en si i que en el nostre projecte es troben dins un paquet (*uoc.tfc.jee.view*).
- **La vista:** és l'encarregada de generar la interfície gràfica amb la qual l'usuari interactua. Serien els resultats que es mostren a l'usuari com a *Results*. En el nostre projecte aquests resultats agafen forma de pàgines *JSP*, complementades amb *Javascript* i *CSS*.
- **El controlador:** comunica la vista i el model donant resposta a les accions de l'usuari que provoquen esdeveniments en la vista, invocant els canvis corresponents i tornant la informació a la vista. El controlador s'implementa mitjançant el filtre *FilterDispatcher*, que és l'encarregat de filtrar les peticions, i per tant examina les sol·licituds i determina l'acció apropiada en cada cas.

Aquest és un diagrama que mostra l'estructura que segueix Struts2:



Imatge 9: Esquema estructura Struts2

## 5.5 Diagrama estàtic de classes i jerarquies

Arribats a aquest punt és necessari oferir una diagrama de classes. En ell s'especificaran els atributs, si més no els més importants, ja que si es representessin tots els atributs i mètodes del projecte el diagrama estàtic de classes seria massa extens i confós.

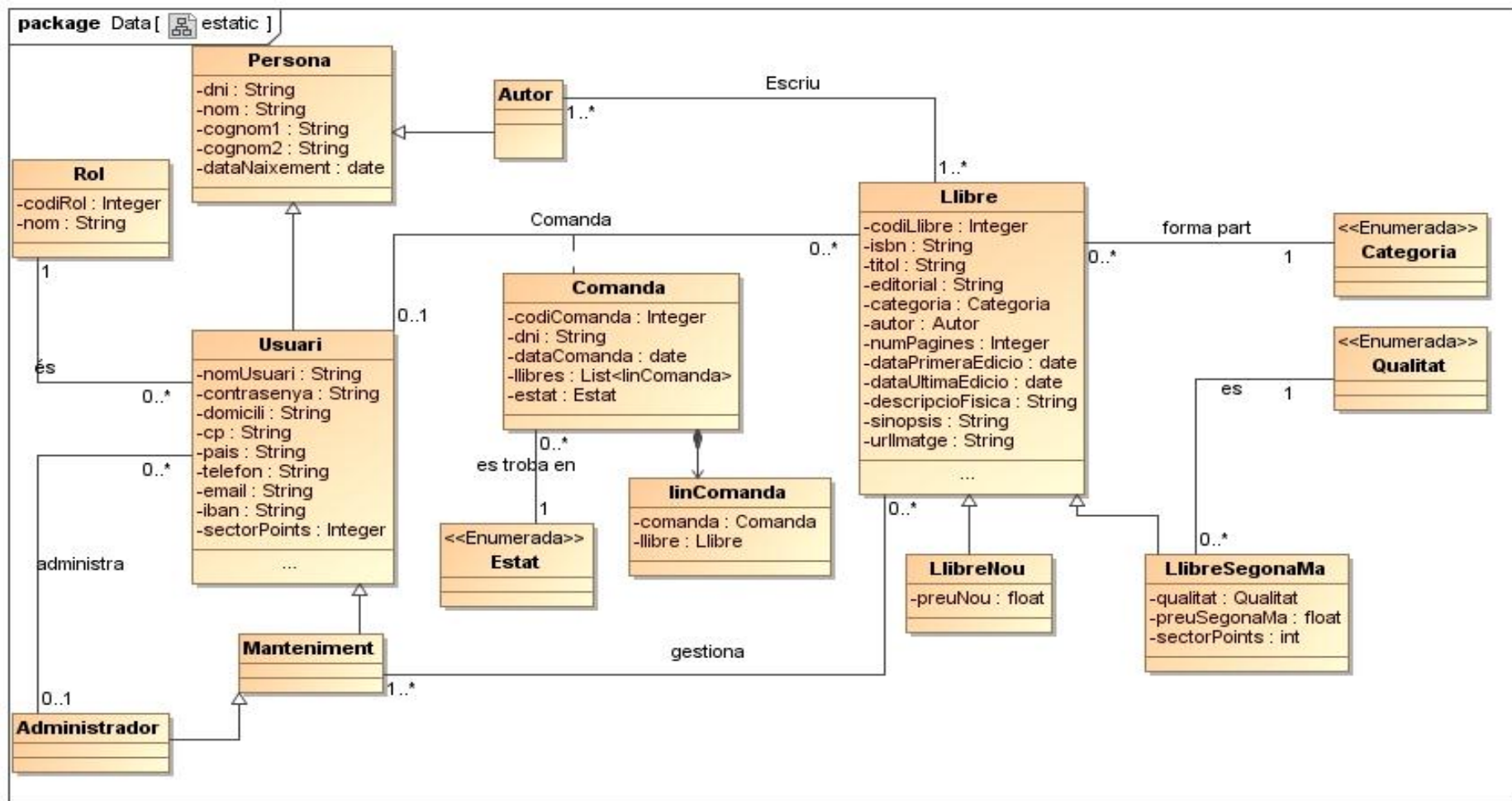
En els diagrames següent es representen les classes entitats que formen part de la lògica de negoci i que (en la seva majoria) representen les entitats del món real, són les classes que podem extreure de l'anàlisi de requeriments i que tenen un símil a la realitat.

### 5.5.1 Diagrama estàtic de classes de la lògica de negoci

Aquest és el diagrama estàtic de classes entitats que actuen a la lògica de negoci.

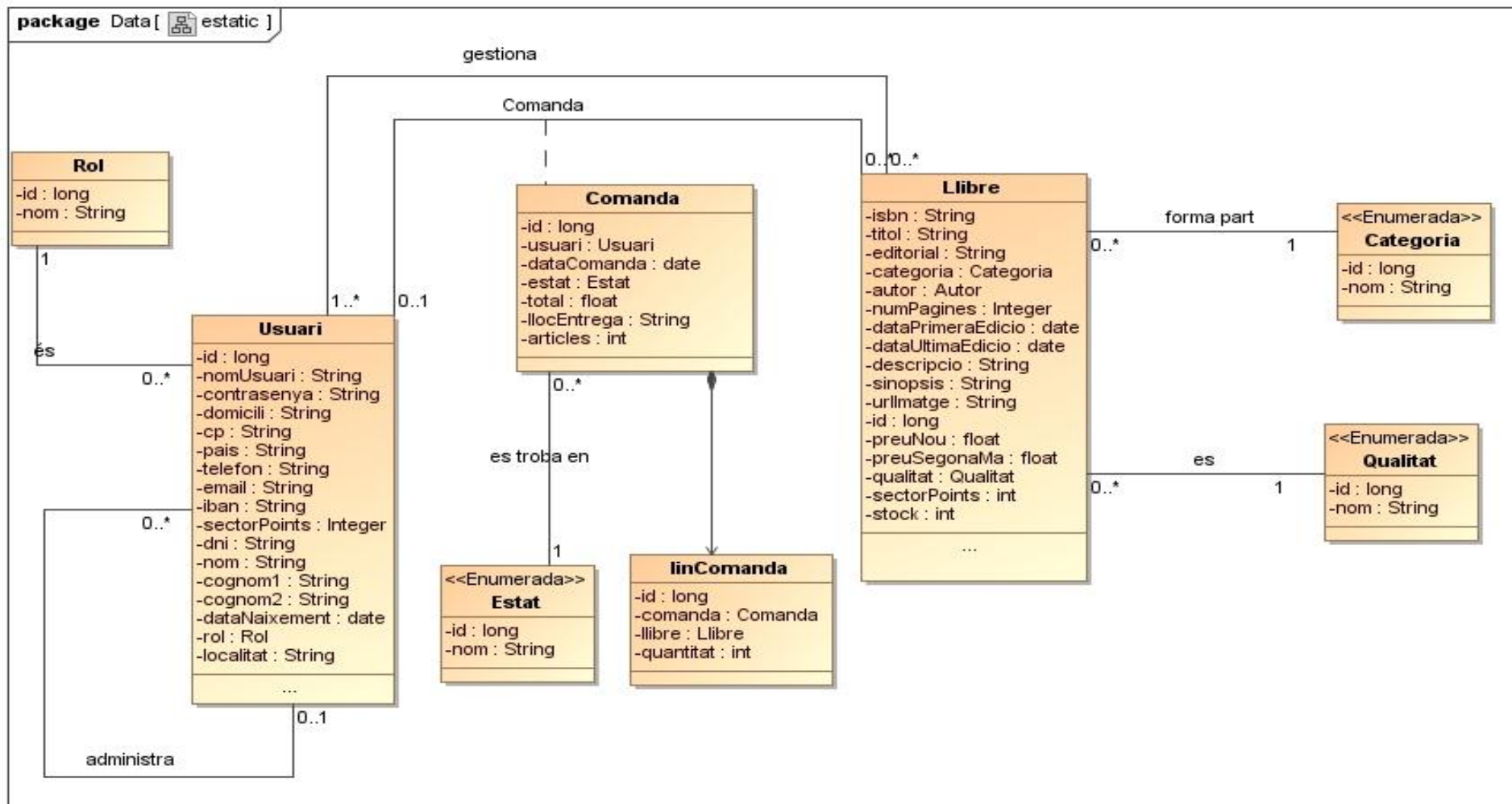
A continuació es presenta el que es va planejar per a la fase de disseny en la PAC2.





Imatge 10: Diagrama estàtic classes 1

Aquest és el diagrama de classes que finalment he utilitzat, ja que per unes dificultats que han sorgit amb algunes configuracions i mapejos d'Hibernate, he suprimit les relacions d'herència, fent una única classe amb multi-funcionalitat. Aquest és el diagrama resultant:



Imatge 11: Diagrama estàtic classes 2

Les coses més importants a destacar del diagrama són:

- **La classe Usuari:** conté les dades més comunes que pot tenir qualsevol persona, sense tenir en compte que pot formar part del grup d'usuaris d'una aplicació informàtica. Per aquest fet té aquests atributs bàsics. A més d'això també té els atributs necessàries per a clients, personal de manteniment i administradors (nom d'usuari, contrasenya, rol, sectorPoints...). El fet que hagi eliminat les relacions d'herència ha fet que hagi d'afegir més atributs.

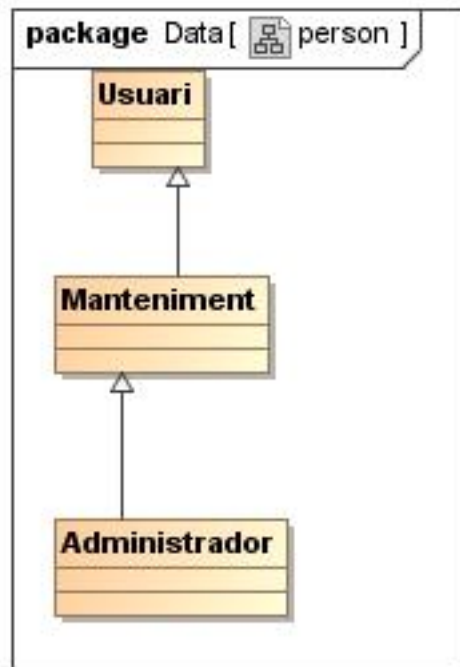
És important destacar l'atribut *IBAN* que es refereix a les dades bancàries. L'usuari insereix les seves dades perquè quedin registrades dins el sistema i es puguin fer els cobraments després de realitzar les comandes. Nos es descarta implementar altres tipus de pagaments, com per exemple *PayPal*.

- **La Llibre:** representa a qualsevol llibre de la realitat, amb les dades que tots els llibres tenen com el títol, l'editorial, l'autor, una imatge de la portada... Inicialment no es feia cap incís en els preus ni punts que costava el llibre, ja que hi havia una relació d'herència que especialitzava aquesta classe cap a dues més: *LlibreNou* i *LlibreSegonaMa*. A causa del que he explicat, he eliminat aquesta relació i ara, per tant, tenim una única classe *Llibre*, amb totes les dades necessàries. Com que es relaciona amb la classe *Qualitat*, sabrem en tot moment si el llibre és nou d'impremta o si per contra, és usat, danyat... etc... Per tant així el llibre serà nou o de segona mà depenent de la seva *Qualitat*, i tindrem l'*stock* actual de llibre.
- **L'atribut *stock*:** Sobre aquest atribut, s'ha de dir que té una doble utilitat:
  - Definir quants llibres hi ha disponibles per comprar en el catàleg.
  - Indicar quants se'n compren per part de l'usuari quan l'afegeix al seu carretó de compra.
- **Les classes enumerades *Qualitat*, *Categoria*, *Rol*, *Estat*:** són classes defineixen valors acotats, és a dir, la qualitat i la categoria d'un llibre (la primera només pels de segona mà), l'estat d'una comanda i el rol dels usuaris prenen uns valors d'una llista enumerada, i només poden prendre aquells. Per exemple la classe *Categoria* podria prendre els següents valors: nou, semi-nou, usat, danyat... Tots aquests valors es prenen de la BD.
- **La classe associativa *Comanda*:** aquesta classe sorgeix de l'associació de la classe *Usuari* i la classe *Llibre*. Representa una compra per part d'un usuari registrar que vol comprar llibres. Aquesta classe té associades una llista de línies de comanda representades per la classe *linComanda* que té com a atributs, el codi de comanda a la que pertany, i el llibre que es vol comprar i la quantitat triada d'aquell llibre.
- **Composició de *Comanda* en *linComanda*:** la relació de composició de la comanda i les línies de la comanda, representen que una comanda que fa el client, està composta per un o més productes, i que aquestes línies de la comanda no poden existir sense al comanda.

## 5.5.2 Jerarquies d'herència més importants

En el diagrama inicial, com ha hem dit, hi ha varies relacions d'herència. Ara aquestes no hi són, però podem expressar les funcionalitats de cada classe en relació al seu *Rol* si parlem de *Usuari*, i per la seva *Qualitat* si parlem de *Llibre*.

### 5.5.2.1 Diferents funcionalitats segons el Rol



Imatge 12: Diagrama herència classes 1

Encara que no existeix una relació d'herència real entre aquestes classes, l'atribut *Rol* ens permet diferenciar diferents funcionalitats que defineixen, al cap i a la fi, tres tipus de comportament.

Per una banda, i el més simple, es tracta d'un *Usuari*, que si està registrat podrà fer de potencial client de l'aplicació i de *BookSector*, consultant el catàleg, comprant llibres, i donant llibres usats per aconseguir punts.

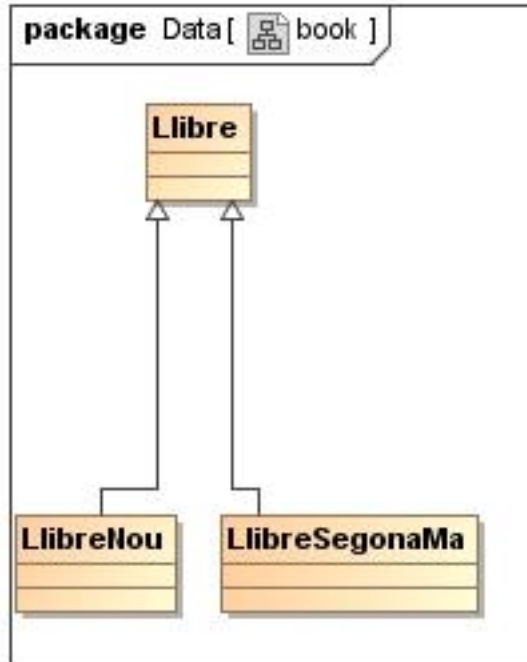
Del comportament d'*Usuari*, s'especialitza en el de *Manteniment*, que a part de fer tot el que fa l'usuari, també pot gestionar els llibres del catàleg i administrar l'estat de les comandes.

Finalment l'administrador especialitza el comportament de *Manteniment*, afegint la funcionalitat de gestió d'usuaris del sistema.

Aquest control de privilegis està controlat mitjançant *Interceptors d'Struts*. En el fitxer *struts.xml* que es troba dins la carpeta *resources* del projecte, s'estableixen 4 paquets d'accions. Un d'ells és el general, on hi van les accions que poden fer tots els usuaris, fins i tot els anònims. Llavors hi ha un paquet on hi a les accions que només els

Usuaris registrats poden fer, un on hi ha les accions que només els de Manteniment poden fer i un últim pels Administradors. D'aquesta forma es controlen els privilegis en l'aplicació.

### 5.5.2.2 Superclasse Llibre



Imatge 13: Diagrama herència classes 2

Continuant amb el que dèiem, aquesta relació d'herència s'ha eliminat, però la idea queda reflectida amb l'atribut de la classe Llibre: *Qualitat*.

Aquest atribut pot prendre els següents valors:

- Nou
- Semi-nou
- Usat
- Danyat
- Molt Danyat

Tots els llibres nous tenen l'atribut Qualitat com a Nou. Els de segona mà tenen qualsevol de les altres opcions depenent de l'estat en el que es trobi.

### 5.6 Diagrama d'estats.

Un diagrama d'estats sobre els estat pels quals passa una comanda, ens ajudarà a entendre l'evolució d'aqueta durant les execucions de l'aplicació.

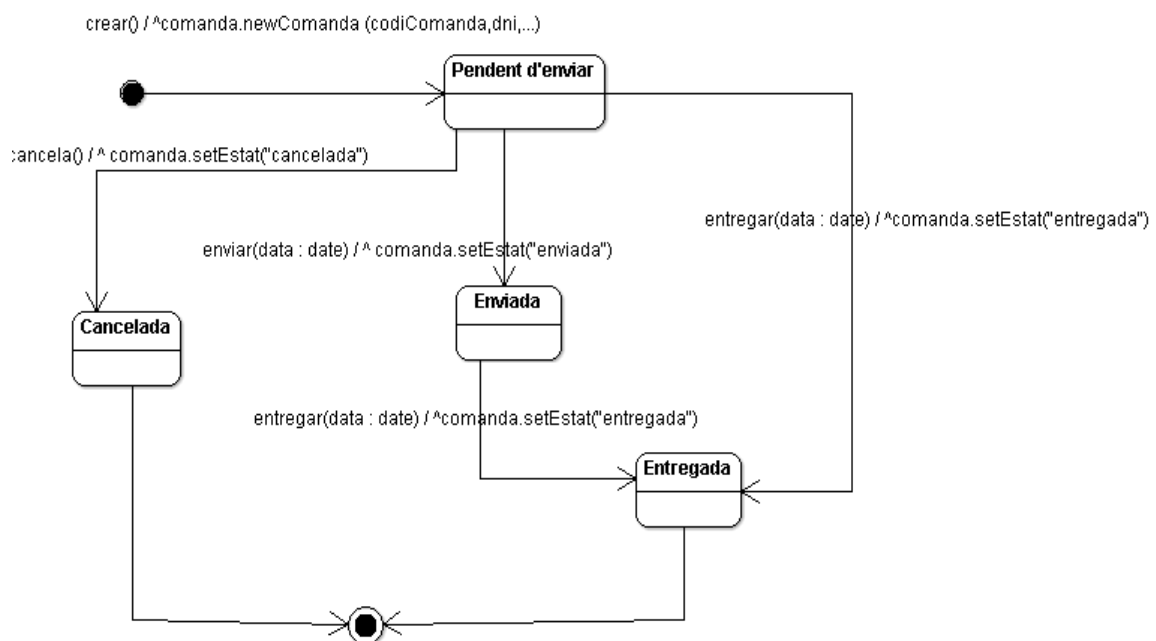
El flux normal d'una comanda és el següent: la comanda s'inicia quan l'usuari executa una compra d'un o més llibres. Llavors la comanda es troba "pendent d'enviar". Llavors només és qüestió que el personal de manteniment o els administradors entrin en la zona privada per anar al llistat de comandes. Allà poden fer que una comanda pendent entri en estat de "enviada". Això passarà quan es faci l'enviament per correu de la comanda realitzada.

De l'estat "pendent d'enviar" també es pot passar al l'estat "entregada". Aquest és el cas en que el client procedeix a buscar al comanda a la tenda, ja que a l'hora de realitzar la comanda hi ha l'opció de triar la forma d'entrega.

Finalment, una comanda "enviada" pot passar a l'estat "entregada". Es tracta del cas en que una comanda enviada per correu al client, arriba a la casa del destinatari.

S'ha de tenir en compte que una comanda es pot cancel·lar, però no mitjançant l'aplicació. En tot cas, el client es posaria en contacte per mail o telefònicament amb *BookSector* per anul·lar una comanda. Com s'ha comentat en el document d'Anàlisi i especificació, no es descarta aquesta funcionalitat en un futur.

Anem a veure això amb un diagrama d'estats.



Imatge 14: Diagrama d'estats comanda

## 5.7 Diagrames de seqüència

A continuació s'exposaran alguns diagrames de seqüència d'alguns casos d'ús. No es generaran tots els diagrames de seqüència possibles però sí els més importants. Amb aquests diagrames es veurà de forma més visual i esquemàtica, el camí que segueixen les funcionalitats.

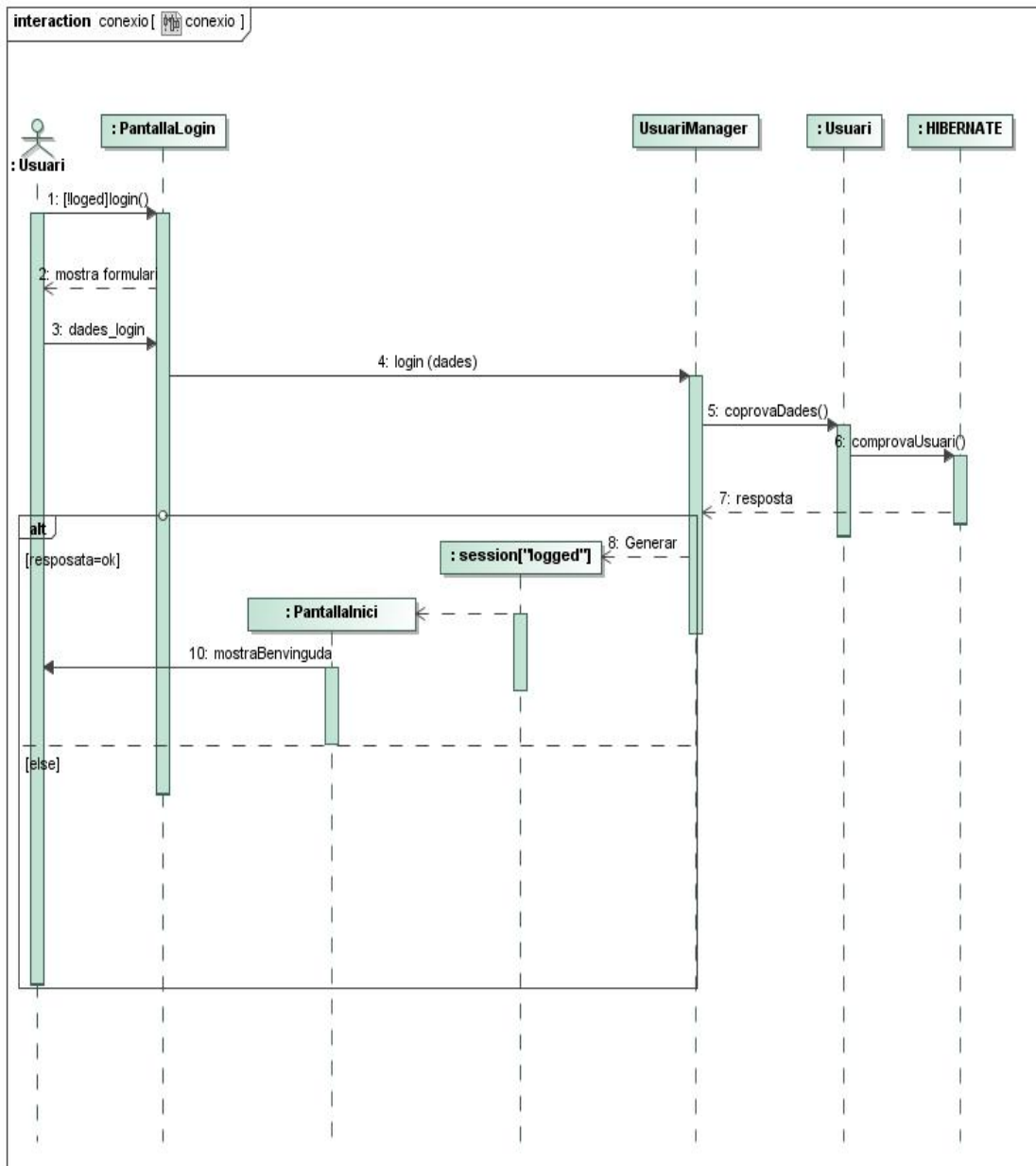
Els diagrames de seqüència mostren la interacció de diferents objectes, instàncies, actors... a través del temps i quin tipus d'informació circula entre ells durant les interaccions.

En els diagrames següents, surten pantalles, les classes que implementen les accions, beans o classes entitat, i hibernate. El que no surt, són les classes que implementen les consultes a la BD, ja que es tracta d'un pas intermedi entre les accions i les peticions a la BD. Cal dir però que la majoria de casos d'ús que es fan van des de una pantalla, passant pel controlador i el filtre de Struts2, s'executa l'acció corresponent i si es necessiten dades de la BD, es crida a les classes Manager, que executen les sentències com a hereves d'HibernateUtil. Aquestes últimes són les que no he inclòs als diagrames, per tal que siguin més simples i més llegibles.

### 5.7.1 Subsistema de Connexió

A continuació es presentaran els diagrames de seqüència d'alguns casos d'ús del subsistema de connexió.

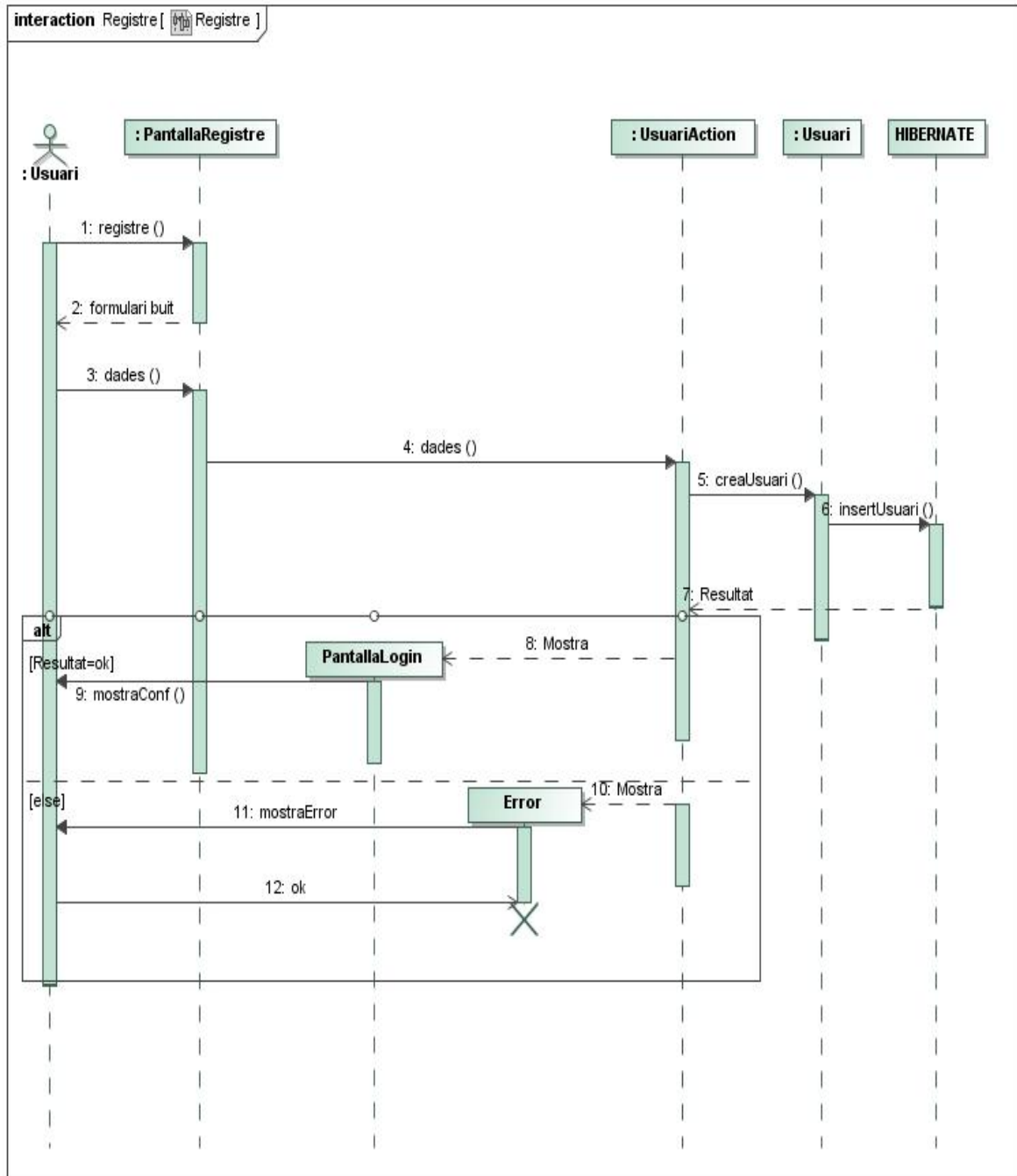
### Login



Imatge 15: DA Login



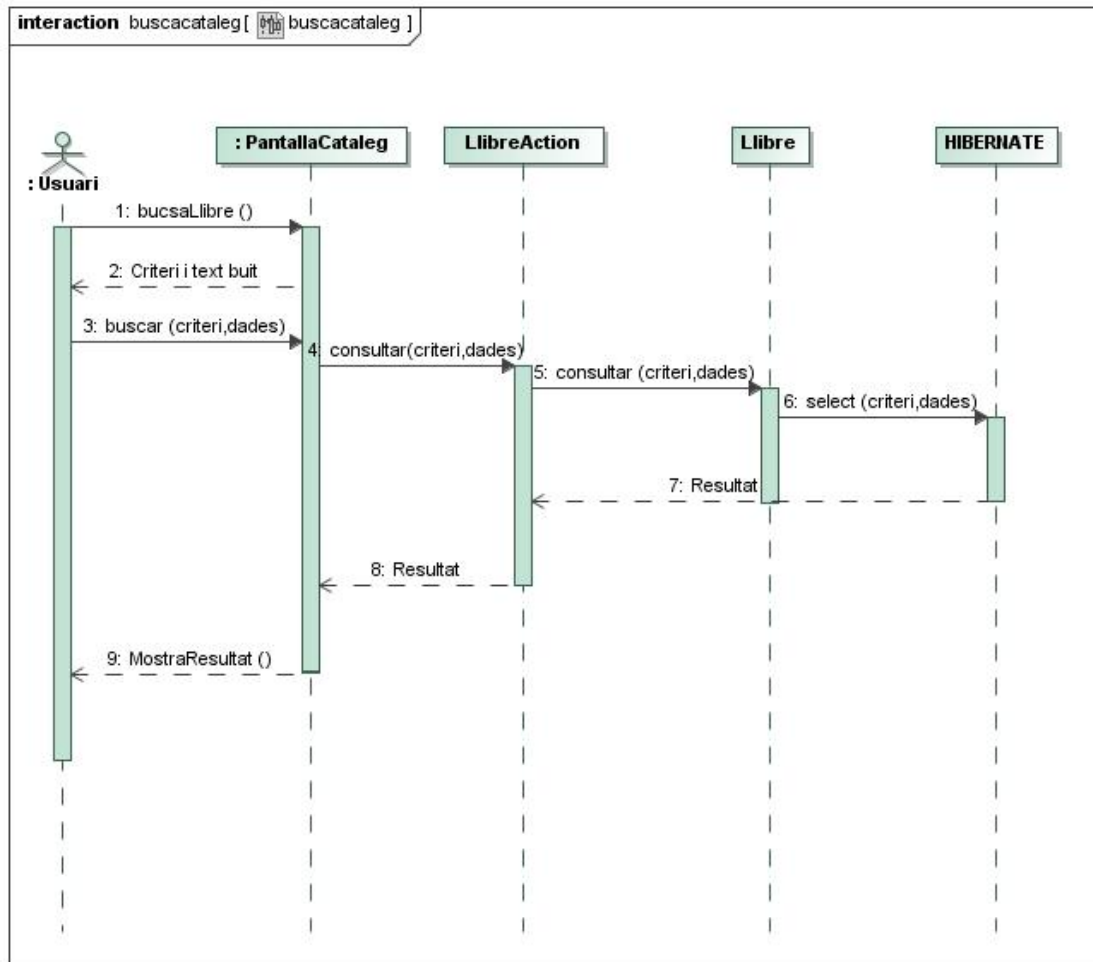
### Registrar Nou Usuari



Imatge 16: DA Registrar Nou Usuari

### 5.7.2 Subsistema de Catàleg

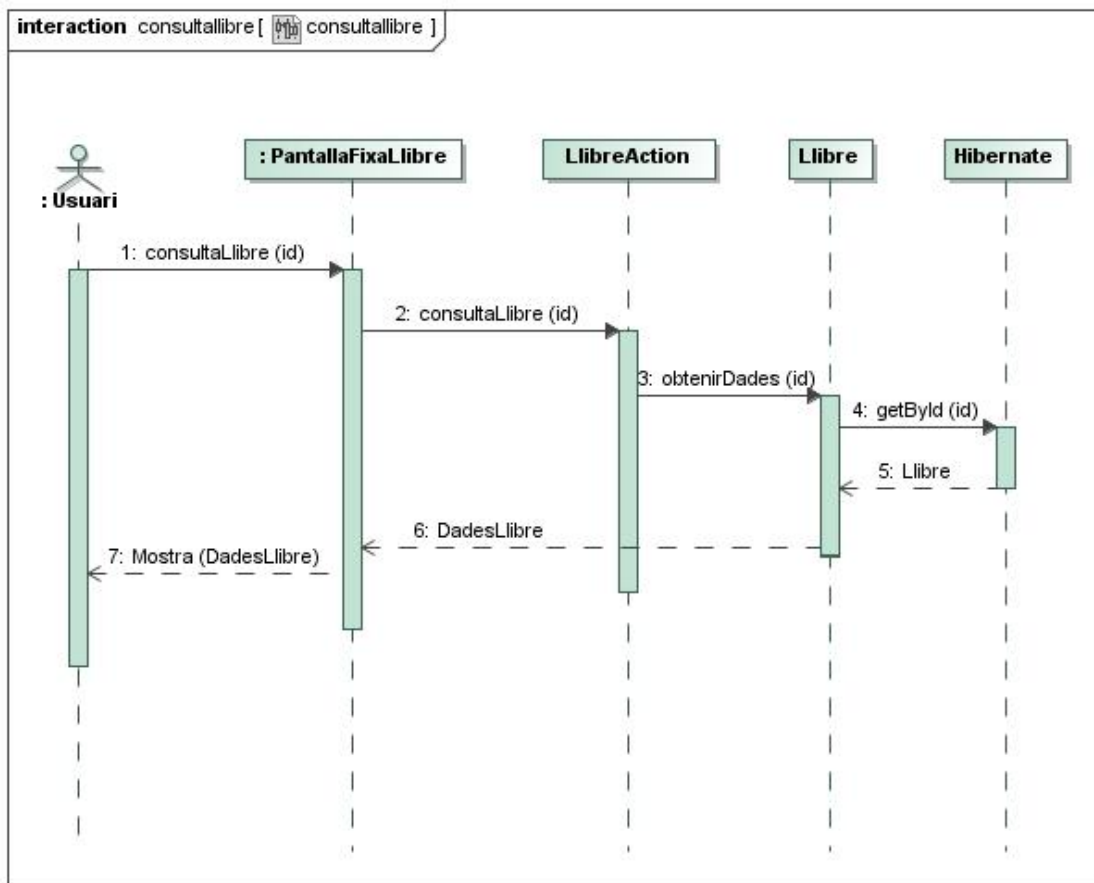
#### Buscar Llibre a Catàleg



Imatge 17: DA Buscar Llibre a Catàleg

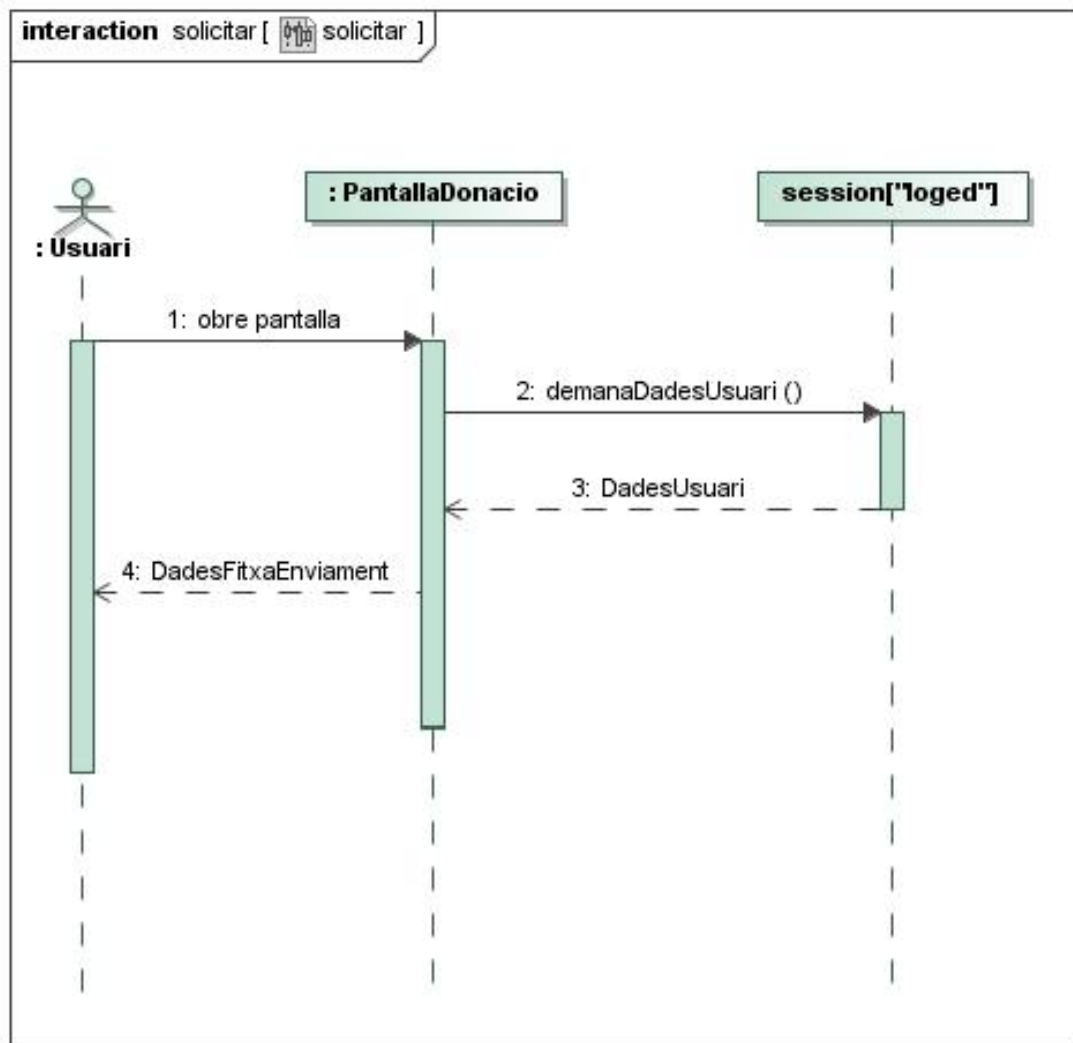
#### Consulta Llibre

Aquest cas d'ús és fa després d'haver cercat els llibres al catàleg amb el cas d'ús anterior. En haver obtingut la llista de llibres de la cerca (en el cas que hi hagi hagut coincidències) l'usuari clica sobre un dels llibres de la llista per veure la seva fitxa completa.



Imatge 18: DA Consulta Llibre

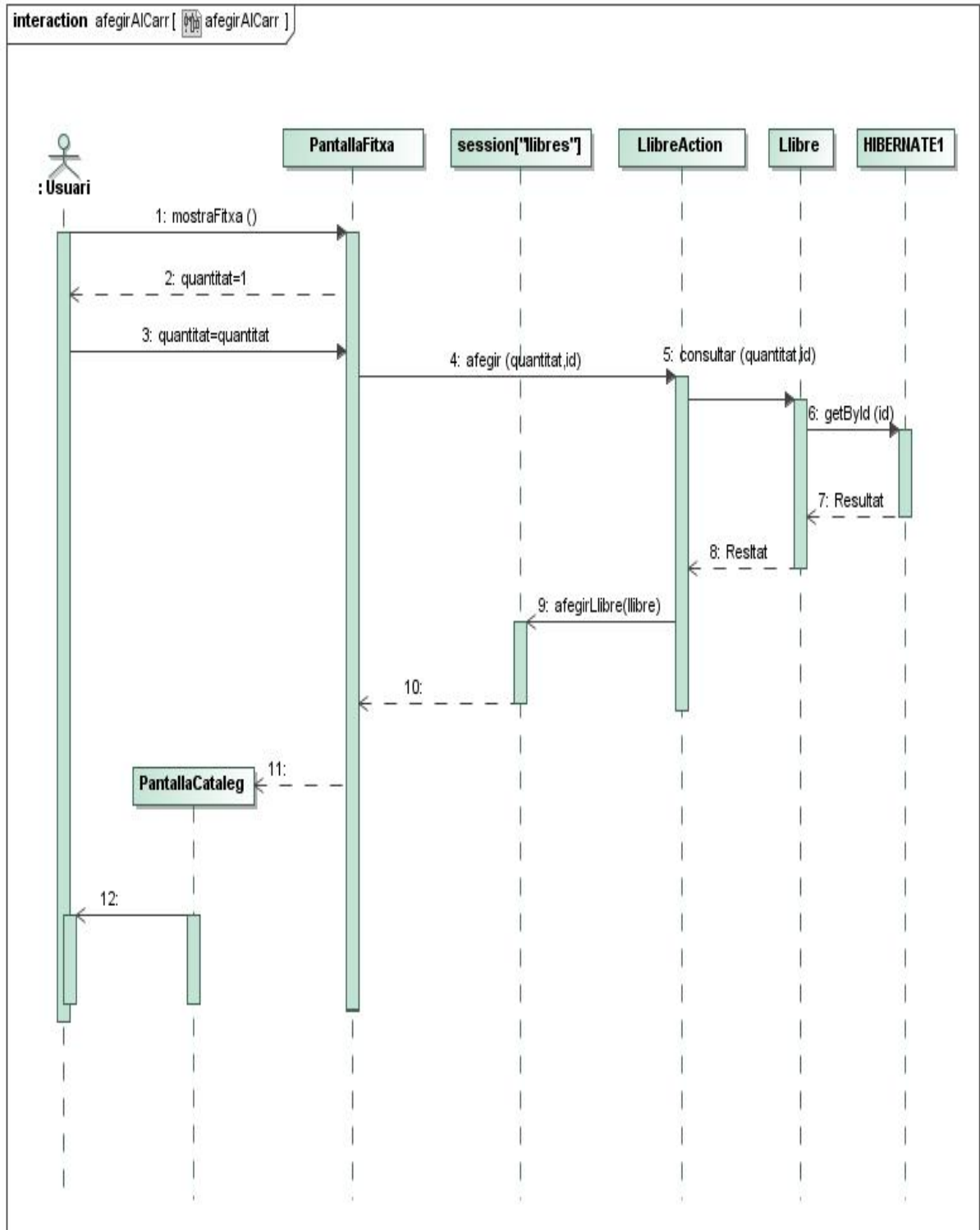
**Sol·licitar donació llibre vell**



**Imatge 19: DA Sol·licitar donació**

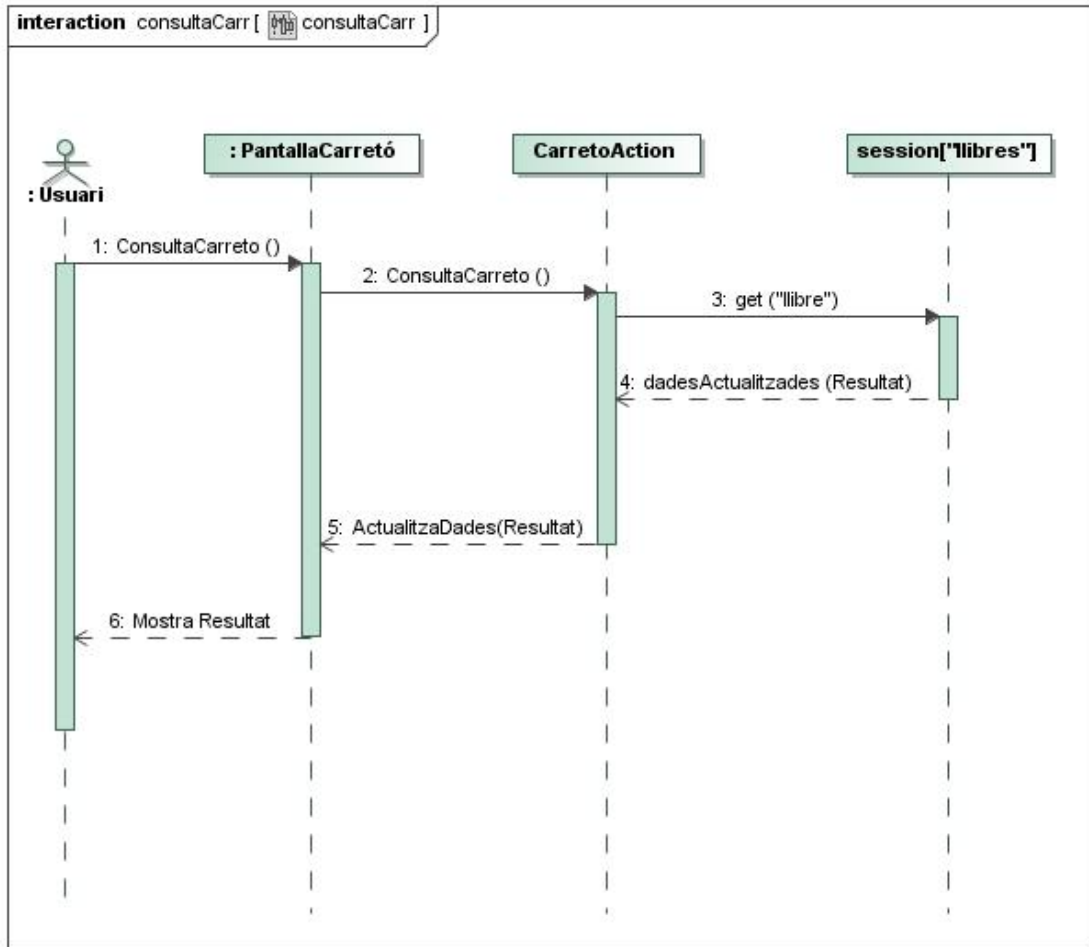
### 5.7.3 Subsistema de Compres

#### Afegir Llibre al Carretó



Imatge 20: DA Afegir Llibre al Carretó

### Consulta Carretó



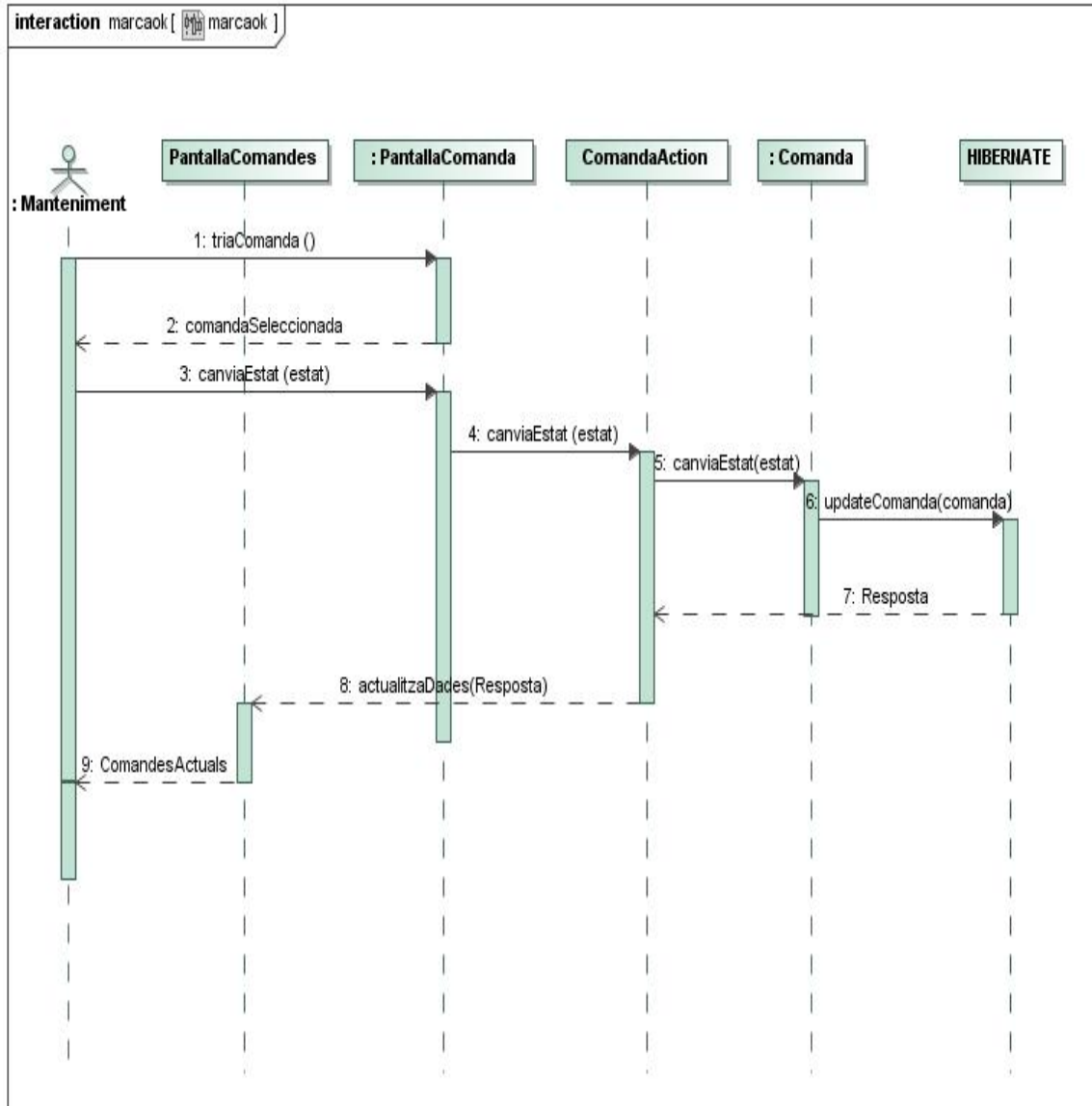
Imatge 21: DA Consultar Carretó

### Suprimir llibre del carretó

Aquest diagrama no és necessari, ja que es donaria el cas d'ús de consulta del carretó i llavors el procés seria molt paregut al d'afegir el llibre, però suprimint el llibre del carretó. Els processos d'actualització serien molt semblants.

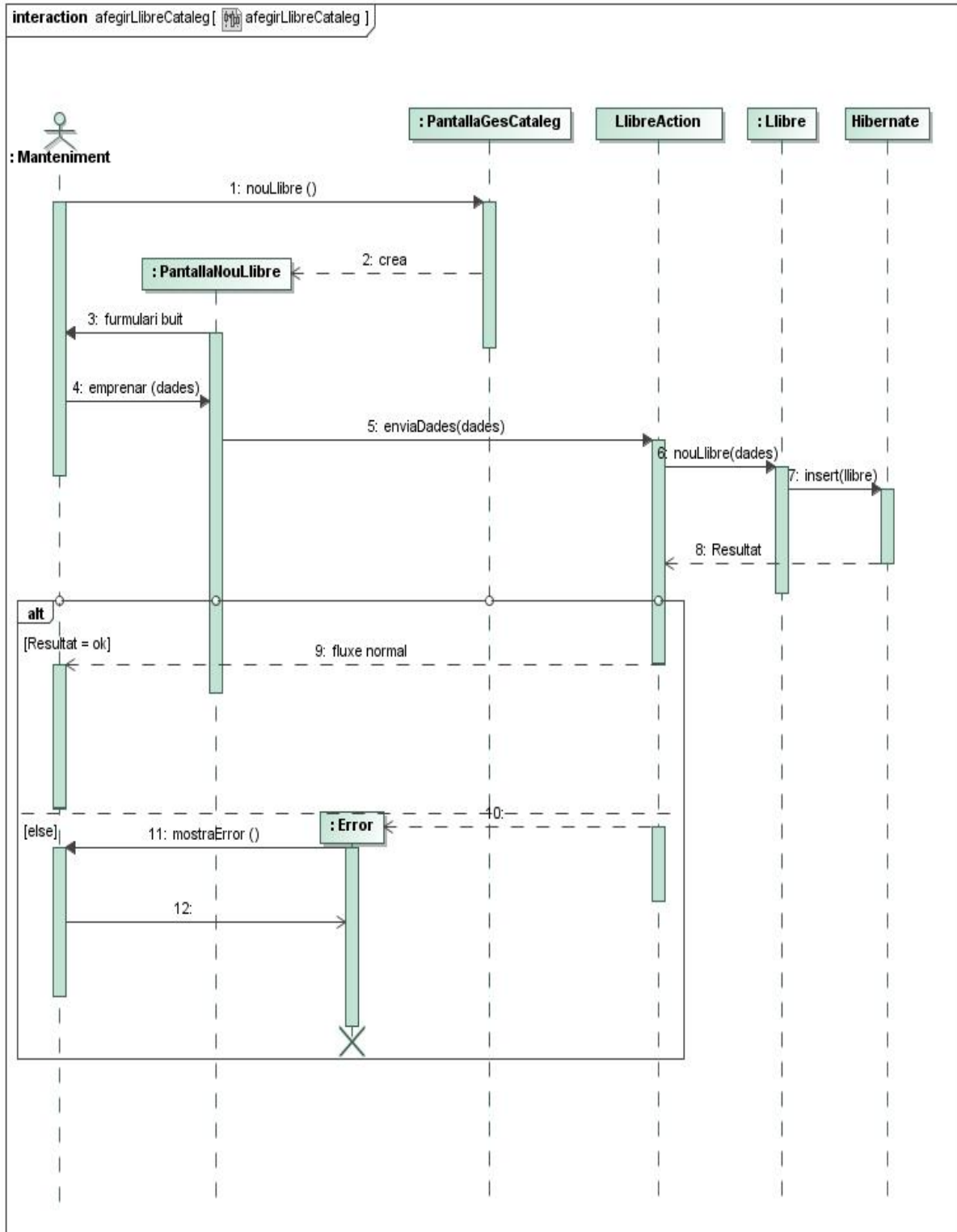
### 5.7.4 Subsistema de Administració

#### Canviar estat comanda



Imatge 22: DA Canviar estat Comanda

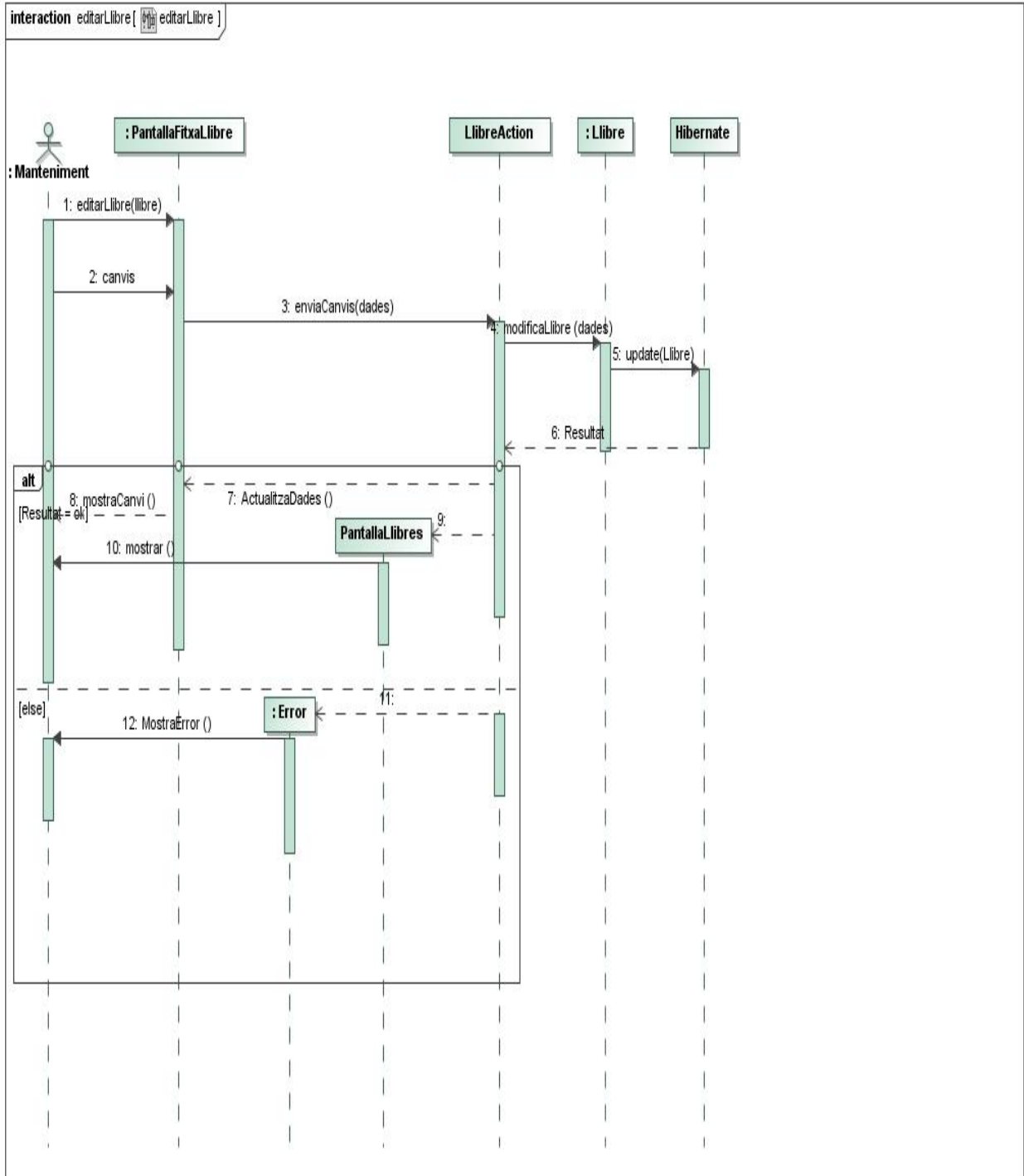
### Afegir Llibre a catàleg



Imatge 23: DA Afegir Llibre a catàleg

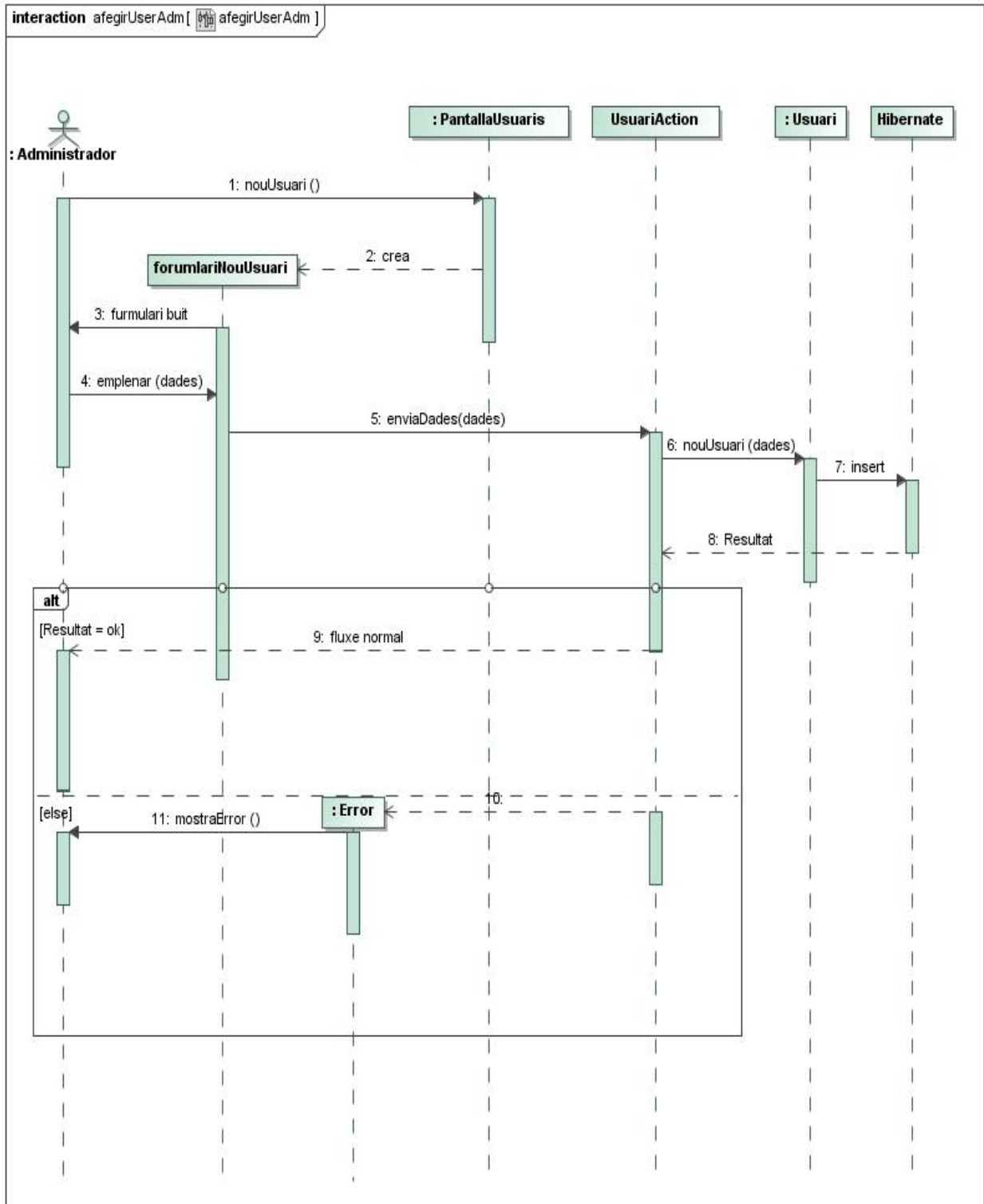


### Editar Llibre



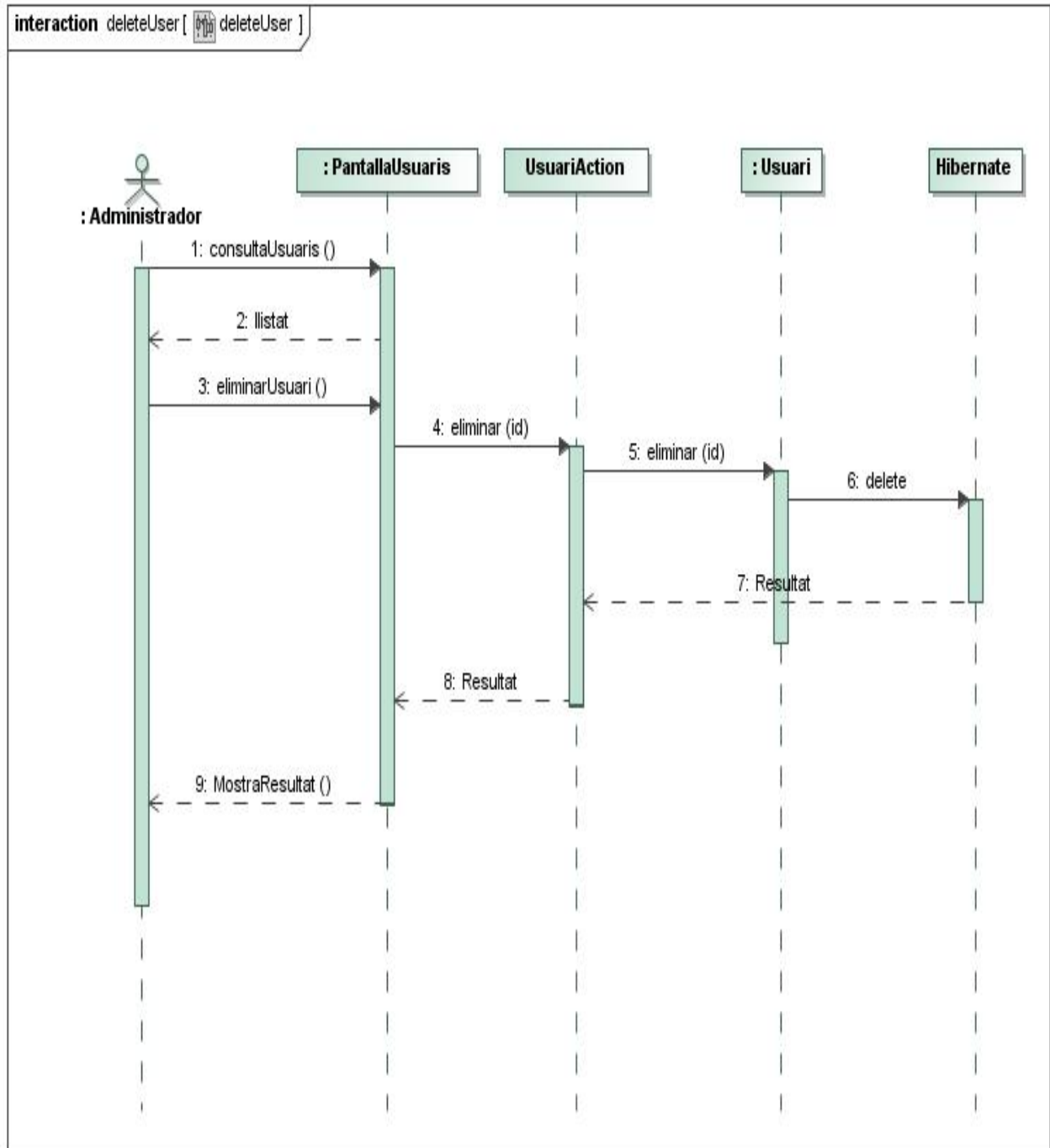
Imatge 24: DA Editar Llibre

### Afegir Usuari



Imatge 25: DA Afegir Usuari

### Buscar Usuari



Imatge 26: DA Buscar Usuari

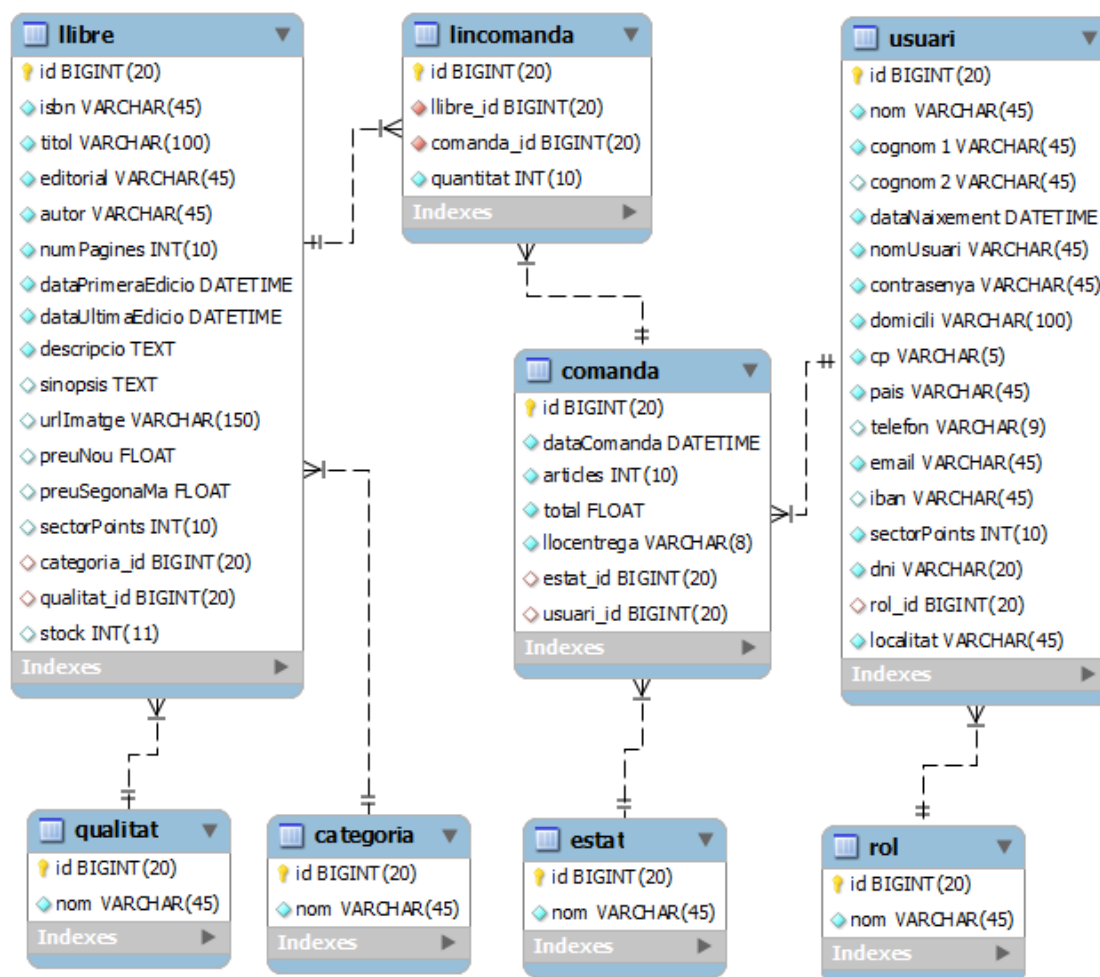
## 5.8 Persistència

### 5.8.1 Diagrama entitat Relació

A continuació dispo de varis diagrames ER on es veuen les relacions de les taules de la base de dades pel que fa al diagrama estàtic de classes UML que torbem en aquest mateix document.

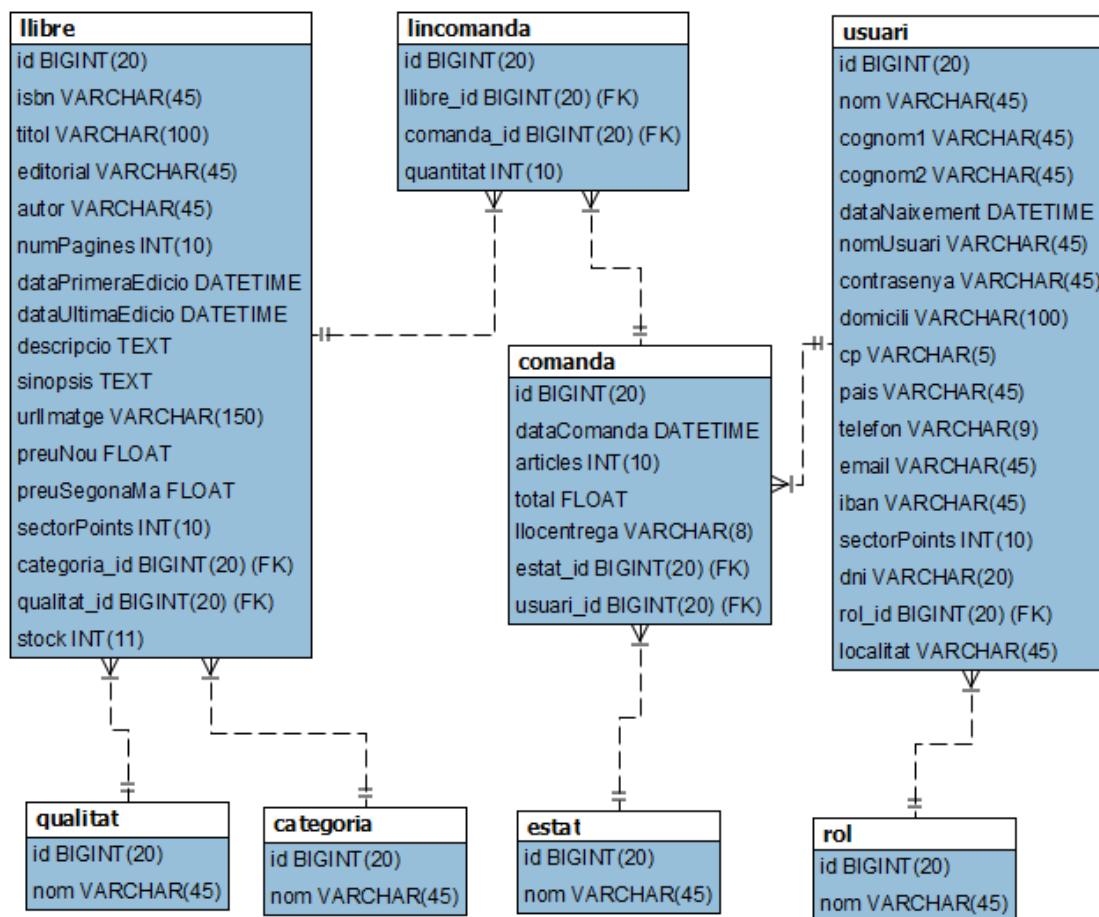
S’ha de dir que, l’herència que en un principi hi havia en el disseny UML de classes estàtic s’ha passat per alt, i per tant les relacions d’herència no surten representades en la base de dades. Per això s’ha utilitzat una mateixa taula per a representar les superclasses i les subclasses.

Hi ha altres alternatives per representar la supressió de l’herència però pel que fa a l’herència d’aquest projecte, incloure tota superclasse i subclasse dins la mateixa taula sembla lo més adient.



Imatge 27: Diagrama persistència 1

Amb el següent diagrama es veu la mateixa informació però d’una forma més textual.



Imatge 28: Diagrama persistència 2

S’ha de dir que les relacions entre les taules tal i com es representen en els dos diagrames anteriors, no hi són de forma “física”. És *Hibernate* el que gestiona totes les relacions de claus foranies i primàries. Igualment, totes les claus primàries que trobem aquí, són auto-incrementables, cosa que *Hibernate* admet. Les claus foranies no hi són com a tal si no que, per exemple, *Hibernate* sap que la columna “rol\_id” de la taula “Usuari”, fa referència a un valor de la taula “Rol”.

Totes aquestes relacions es fan amb les anotacions directament sobre les classes Java. Les classes model amb anotacions es troben dins la carpeta “uoc.tfc.jee.model” del projecte.

**Nota:** en els annexos hi ha scripts de creació de la base de dades juntament amb les dades inicials pe veure el complet funcionament del projecte.

## 6. Implementació.

### 6.1 Requisits de programari per a la implementació

Aquests són els requisits per a realitzar la implementació d'aquest projecte.

- **Software i configuracions necessàries:**
  - Java Runtime Environment 1.6 i Java Development Kit 1.6 (JRE6 i JDK6) o superior per al desenvolupament i execució del projecte.
  - Sistema Gestor de Base de dades MySQL 5.1 Server o superior.
  - Servidor d'aplicacions web Apache 7.0, que suporta l'especificació J2EE i permet el desplegament de fitxers .war del projecte.
  - ANT, per a l'execució de fitxers build.xml que compilen i despleguen el projecte de forma automatitzada. Es proporciona un fitxer build.xml i les llibreries necessàries amb la carpeta del projecte.
  - Navegador tipus NetScape, Internet Explorer, Google Chrome, Firefox, Safari, etc... que suporti la darrera versió d'HTTP.
  - Llibreries de la JDBC per a les connexions a la BD (incloses amb l'actual projecte).
  - Hibernate, llibreries i fitxers necessaris per a l'execució de consultes i realització de les estructures de la BD (incloses amb l'actual projecte).
  - Especificació Struts 2, per tal d'implementar el model MVC 2, tal com s'ha explicat en aquest document. Fitxers i llibreries necessàries (incloses amb l'actual projecte).

### 6.2 Eines de desenvolupament utilitzades.

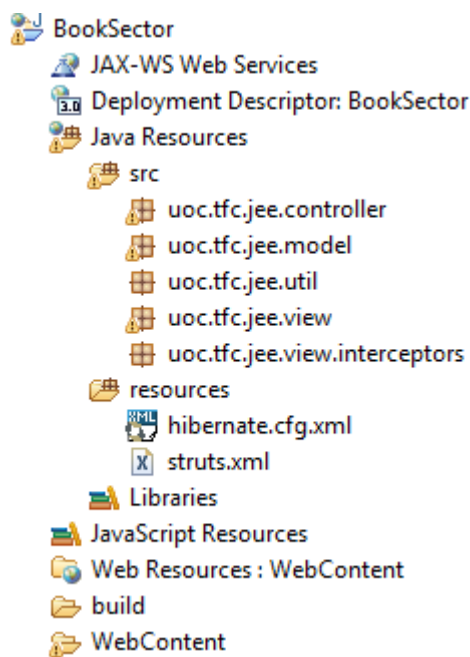
Per a realitzar el projecte he fet servir les següents eines de desenvolupament que han agilitat la feina.

- Eclipse IDE for Java EE Developers, que permet la instal·lació de diferents “plugins” per al desenvolupament d'aplicacions amb l'especificació J2EE.
  - Plugin per a servidors Apache Tomcat i JBOSS.
  - Plugin per a connexions Hibernate.
- MySQL Administrator, de les GUI TOOLS de MySQL. Facilita la gestió de la BD.
- MySQL Workbench 5.2, que ofereix diferents funcionalitats addicionals per a tractar i inspeccionar les BD.

- DreamWeaver, per facilitar certes tasques de disseny i regles CSS.
- ArgoUML i MagicDraw per la realització dels diferents diagrames UML.
- Gantt Project per la creació i edició de diagrames de Gantt.

### 6.3 Estructura Interna del projecte

L'objectiu d'aquest punt és donar una idea aproximada de l'estructura de fitxers, llibreries i classes del projecte. Simplement comentaré com es distribueix tot el material dins la carpeta del projecte, les carpetes del qual s'han anat creant i gestionant utilitzant Eclipse:

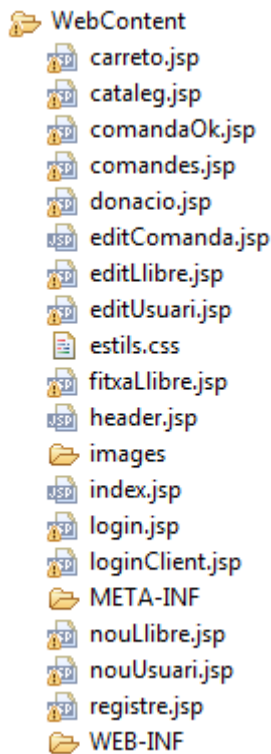


Imatge 29: Arbre de paquets 1

Hi ha dues carpetes principals de codis:

- src: amb totes les classes Java, en la que hi trobem els paquets:
  - uoc.tfc.jee.controller: Hi ha les classes Manager, que s'encarreguen de la interacció amb la base de dades.
  - uoc.tfc.jee.model: Hi ha les classes entitat que hem definit en el diagrama de classes UML. Aquestes tenen les anotacions Java d'Hibernate per a fer el mapeig que les relacionen amb les taules de la BD.
  - uoc.tfc.jee.util: La classe SessionFactory d'Hibernate.
  - uoc.tfc.jee.view: Conté les classes que implementen les accions de Struts2.

- uoc.tfc.jee.view.interceptors: Hi ha els interceptors de Struts que controlen els privilegis dels usuaris dins les diferents seccions de l'aplicació.
- resources: conté fitxers de configuració:
  - hibernate.conf.xml: és el fitxer de configuració d'Hibernate. És el que s'ha de modificar en canviar l'aplicació de Host, pel que fa a la localització usuari i contrasenya de la base de dades.
  - struts.xml: és el fitxer que fa de filtre i controlador de l'especificació Struts2, on hi ha el control de les accions i els resultats.



**Imatge 30: Arbre de paquets 2**

- WebContent: Aquesta carpeta conté:
  - Tots els fitxers jsp que componen l'aplicació i una fulla d'estils CSS.
  - Una carpeta "images", on hi ha les imatges de l'aplicació i les portades dels llibres.
  - WEB-INF
    - Carpeta "lib" on hi ha totes les llibreries utilitzades per l'aplicació
    - Fitxer "web.xml" amb la configuració de l'espai web.



## 7. Valoració econòmica

L'objectiu del TFC era afrontar un problema real, i posar a prova els coneixements obtinguts durant la carrera mentre s'avançava per una corba d'aprenentatge que s'havia d'administrar mentre es lliuraven les diferents etapes.

No s'ha pretès en cap moment crear una aplicació comercial amb ànim de lucre. A més, tot els software utilitzat en la realització del projecte és lliure i de llicència i codi obert. Per tant no farem cap més incís en aquest punt de valoració econòmica.

## 8. Conclusions

Primer de tot, dir que la realització d'aquest projecte ha estat una experiència molt enriquidora, tant a nivell personal com a nivell acadèmic. Arribar a aquest punt de la carrera d'informàtica i poder aplicar un ventall tan ampli de coneixements adquirits durant semestres, fa que un s'adoni de tot el que ha guanyat i crescut un relativament poc temps.

Ha estat tot en repte per mi, ja que, a nivell acadèmic, mai m'havia enfrontat a un problema tan complex, que requerís tan bona gestió del temps i dels coneixements adquirits i a que al mateix temps, requerís un nivell d'investigació i aprenentatge simultani a la feina realitzada.

Gràcies a la feina realitzada, he pogut comprovar la importància de les bones costums dins aquest món que és la programació i la gestió de projectes, i m'ha permès copsar una mica com enfocar els problemes de la vida real en el món laboral de la informàtica. Posar en pràctica tot el que sabia m'ha fet veure tot el que em queda per aprendre i descobrir, no només pel que fa a la informàtica, si no també pel que fa a l'afrontament de problemes, quan la falta de coneixement i temps van a la teva contra i als quals no pots dedicar tot el temps que desitjaries.

Vaig triar aquest TFC perquè durant els semestres a la UOC he quedat sorprès de les possibilitats que ofereix la programació en Java i el paradigma orientat a objectes, i a més volia introduir-me en la programació d'aplicacions distribuïdes amb tecnologies totalment desconegudes per jo.

Tot el coneixement obtingut durant aquest semestre sobre l'especificació J2EE, Struts2, Hibernate, entre altres, m'ha mostrat la magnitud del domini que Java pot arribar a exercir sobre el món de les aplicacions web distribuïdes.

Considero que em queda moltíssim per aprendre en aquest món, ja que penso que l'aplicació desenvolupada té mancances i errades que de ben segur no cometré en projectes futurs, però que com a primera experiència real i en aquest món, amb la dura corba d'aprenentatge i amb les dates d'entrega en contra, juntament amb les altres assignatures que he cursat, puc donar-me per satisfet sabent que, des de el meu punt de vista, he fet una bona feina.

Espero poder millorar i arribar a dominar molt bé l'especificació J2EE durant el meu trajecte acadèmic i laboral en la informàtica.

## 9. Glossari

**Usuari:** qualsevol persona que es connecta a la pàgina de BookSector per utilitzar el programa. Pot ser un usuari anònim o un usuari registrat.

**Manteniment:** és un Usuari que té drets per administrar el catàleg de llibres i les comandes.

**Administrador:** Com l'usuari de manteniment però que també pot administrar els usuaris enregistrats.

**Catàleg:** sistema que mostra llibres a l'usuari depenent d'uns criteris de cerca.

**Carretó:** secció implementada per una sessió que conté en tot moments els llibres que l'usuari ha afegit buscant-los pel catàleg. Permet afegir llibres, eliminar-los o realitzar la comanda dels llibres seleccionats.

**Comanda:** conjunt de llibres comprats per l'usuari i que pot tenir diferents estat.

**Línia de comanda:** cada llibre del qual es compon una comanda, on s'especifica el llibre i la quantitat d'ells que es volen comprar.

**Subtotal:** quantitat acumulada pel preu dels llibres dins el carretó.

**Total:** quantitat que s'obté restant del subtotal, els preu dels llibres que es compren amb *SectorPoints*.

**Nom d'Usuari / Codi d'Usuari:** combinació de caràcters alfanumèrics que identifiquen l'usuari juntament amb la contrasenya a l'hora de fer el *Login*.

**Contrasenya:** combinació de caràcters alfanumèrics que es visualitzen xifrats, que juntament amb el Nom d'Usuari serveix per fer el Login de l'usuari.

**Login:** procés de validació que fan els usuaris de BookSector per tal d'identificar-se com a usuaris registrats.

**Stock:** nombre d'exemplars disponibles d'un llibre en un moment concret.

**Rol:** elements de qualsevol usuari que l'identifica com a Usuari, Manteniment o Administrador.

**SectorPoints:** punts que acumules els usuaris per la donació de llibres usats al negoci.

**Categoria:** àrea temàtica a la que pertany un llibre.

## 10. Bibliografia

### Struts2

- <http://www2.elo.utfsm.cl/~elo326/Presentaciones/Ronda4/Struts.pdf>
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=miPrimeraWebStruts>
- <http://www.roseindia.net/struts/struts2/index.shtml>
- <http://www.roseindia.net/struts/struts2/struts2validation.shtml>
- <http://www.roseindia.net/struts/struts/struts2.2.1/shoppingcart/index.html>
- <http://es.debugmodeon.com/articulo/autenticacion-web-usando-interceptor-de-struts-2>
- <http://mundogeek.net/archivos/2009/02/08/struts-2/>

### Hibernate

- <http://es.wikipedia.org/wiki/Hibernate>
- <http://www.javatutoriales.com/2009/05/hibernate-parte-1-persistiendo-objetos.html>
- <http://www.hibernate.org>
- <http://viralpatel.net/blogs/2010/01/tutorial-struts2-hibernate-example-eclipse.html>
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernateTools>

### JSP – Servlets

- [http://www.programacion.com/articulo/servlets\\_y\\_jsp\\_82](http://www.programacion.com/articulo/servlets_y_jsp_82)
- <http://www.adrformacion.com/cursos/j2ee/leccion1/tutorial1.html#>
- <http://lineadecodigo.com/java/mi-primer-servlet/>
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jspbasico>

### J2EE

- j2ee.pdf (tutorial UOC – Albert Grau Perisé)
- Un cas pràctic “Tenda virtual mascotas” (tutorial UOC – Albert Grau Perisé)
- Laboratori\_EPCSD\_v1.20c (tutorial UOC – Albert Grau Perisé)

### UML i Gestió de projectes

- Mòdul 3. UML (II): el model dinàmic i d'implementació “Enginyeria del software” (Material UOC)
- Gestió de projectes informàtics de programari orientat a objectes “Tècniques de desenvolupament de programari” (Material UOC).

-Desenvolupament de programari orientat a objectes “Tècniques de desenvolupament de programari” (Material UOC).

### **Muntatge ANT**

<http://www.webproyecto.com/info/articulo.php?id=134>

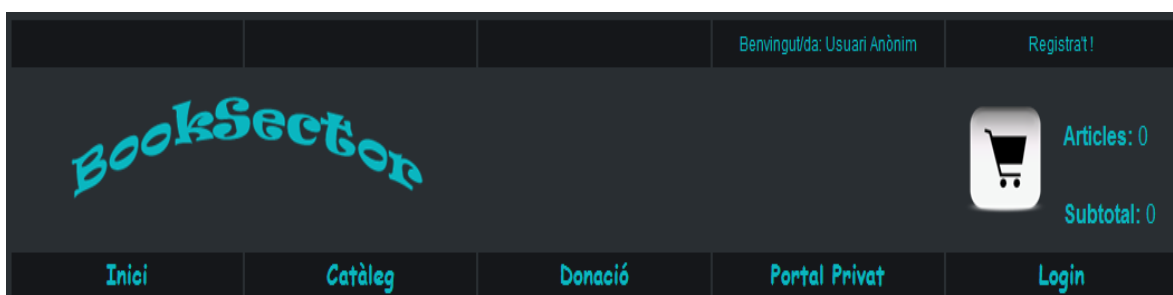
## 11. Annexos

### 11.1 Annex I: Interfície gràfica. Petita guia de funcionament.

La interfície gràfica ha estat implementada mitjançant pàgines JSP, algunes d'elles complementades amb Javascript. Per quasi tots els formularis que es van utilitzant en la interfície gràfica s'han utilitzat els "tags" o les etiquetes que ofereix Struts2. Llavors tant la plana JSP com les etiquetes de Struts2 es tradueixen a codi HTML per a ser presentades a l'usuari final.

Totes les pàgines inclouen una capçalera, i el contingut que hi ha sota canvia depenent de la pàgina JSP a la que cridi el controlador d'Struts2 com a resposta a una "action".

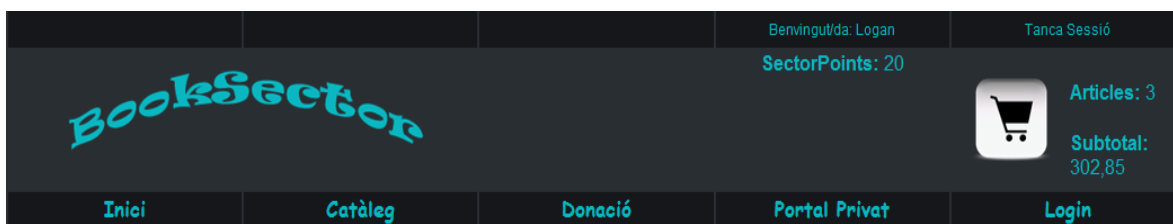
La capçalera és la següent:



Imatge 31: GUI capçalera

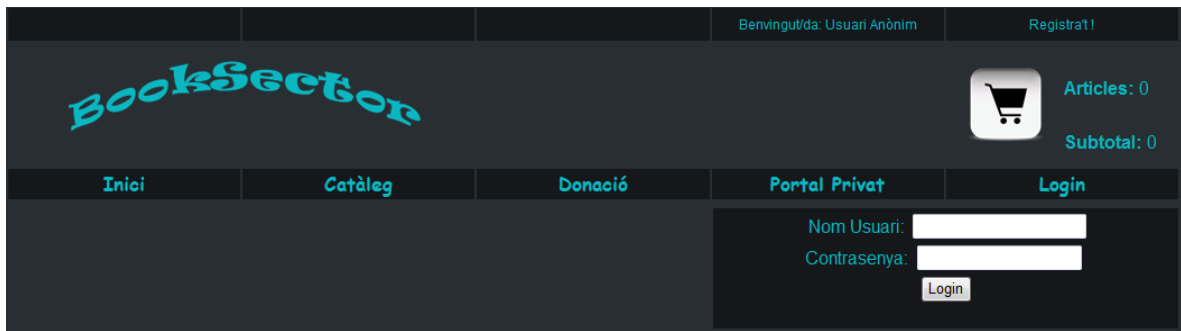
Com podem veure hi ha un botó-link per a cada secció (Inici, Catàleg, Donació, Portal Privat i Login). A la part superior dreta trobem les dades que es mostren amb un usuari anònim. Com es pot veure, dona la benvinguda a l'anònim, ofereix l'opció de registrar-se i mostra la imatge del carretó que clarament, està buit.

La següent figura mostra l'estat de la capçalera quan hi ha un usuari connectat i validat amb el seu nom d'usuari i contrasenya.



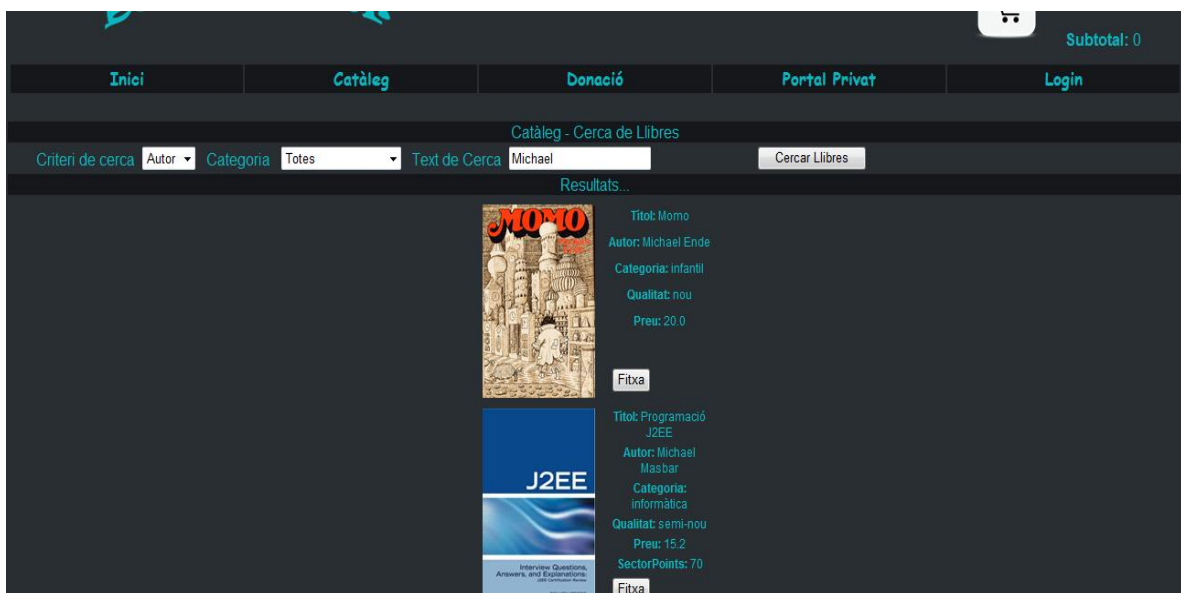
Imatge 32: GUI capçalera validada

Com es pot veure, ara les dades que mostra la capçalera han canviat. Es mostra el nom d'usuari, s'ofereix l'opció de tancar sessió, es mostren també els SectorPoints que té actualment l'usuari i les dades del carretó ara mostren que hi ha tres articles al carretó i que s'ha acumulat un subtotal de 302,85 €. A més, ara la imatge del carretó és un botó, amb el que podem entrar a consultar el carretó.



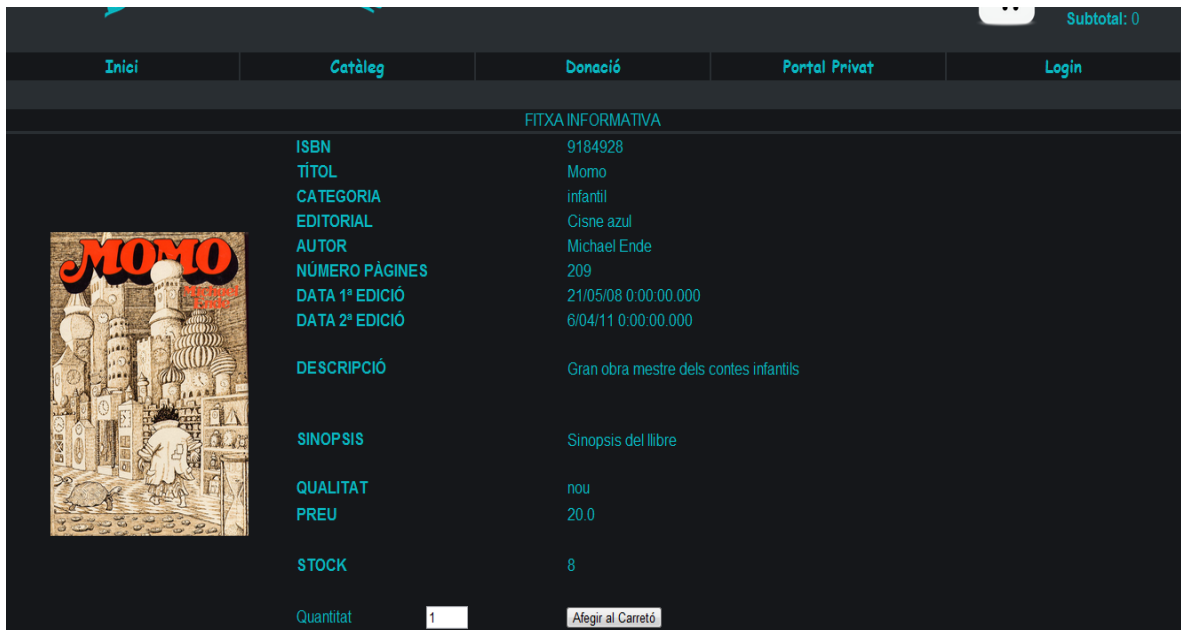
**Imatge 33: GUI Inici**

Aquest és la secció inicial i de benvinguda de l'aplicació. Inicialment, pel fet de no haver-se validat, dóna l'oportunitat de fer el "Login". Un cop fet el "Login" correctament, el formulari desapareix. Aquesta secció no necessita haver iniciat sessió.



**Imatge 34: GUI Catàleg**

L'anterior pantalla és el catàleg, on tenim diferents criteris de cerca per filtrar els llibres que en surten. Cada cop que premem el botó "Cercar Llibres" s'aplica el filtre seleccionat i es mostren els resultats a sota. Els resultats són els llibres que coincideixen amb els criteris de cerca i es mostra, per cada un d'ells, la portada, les principals dades, i un botó "Fitxa" per mostrar la fitxa completa del llibre. No es necessari estar registrat per fer Aquesta acció.



Imatge 35: GUI Fitxa de Llibre

Aquesta és la fitxa d'un llibre. Surt tota la informació del llibre, entre ella, si es nou o de segona mà i dona l'opció d'afegir-ne una certa quantitat al carretó, mitjançant el camp i el botó que hi ha a la part inferior. Per fer aquesta acció, s'han de tenir drets d'usuari registrat i haver-se validat al sistema.



Imatge 36: GUI Carretó

L'anterior imatge mostra el carretó. Com s'observa, a la part esquerra mostra els llibres que hi ha dins el carretó, amb la quantitat que s'ha tirat a la fitxa de llibre. A la part dreta surten les opcions de pagar els possibles llibres amb punts o només amb diners, i l'opció de recollir els llibres a la tenda o de rebre'ls a casa.

Els dos botons grans de la dreta són per realitzar la comanda amb els llibres i les opcions triades i per buidar el carretó respectivament.



**Imatge 37: GUI Confirmació Comanda**

Aquesta pàgina surt quan la comanda s'ha acceptat i realitzat correctament. És un resum de la comanda realitzada i la idea és que l'usuari ho imprimeixi a forma de comprovant.

La següent imatge mostra la plana per a imprimir l'etiqueta per l'enviament d'un llibre usat per part d'un usuari registrat. A ella s'hi accedeix amb l'opció "Donació" del menú principal i es necessari estar registrat per accedir-hi.



DONACIÓ LLIBRE - Dades d'Etiqueta

Etiqueta d'enviament de llibre. Usuari Logan

Dades d'usuari	
ID	1
DNI/Doc. Idtf.	41231544H
Nom	Salvi
Llinatges	Planes Martorell
Carrer i N <sup>a</sup>	c/ estany 34
Localitat	Manacor
CP	12312
País	Espanya

Adreça de destinació del Llibre

Carrer i n <sup>o</sup>	c/ sombrereria 1
Localitat	Palma
CP	07005
País	Espanya

Imatge 38: GUI Donació

Les següents imatges necessiten dret de “Manteniment” i per tant l’usuari que es validi, ha de disposar d’aquests drets.

GESTIÓ DE CATÀLEG

ISBN:	<input type="text"/>
Títol:	<input type="text"/>
Editorial:	<input type="text"/>
Categoria:	<input type="text" value="literatura clàssica"/>
Autor:	<input type="text"/>
Nombre de pàgines:	<input type="text"/>
Data Primera Edició:	<input type="text"/>
Data Última Edició:	<input type="text"/>
Descripció:	<input type="text"/>
Sinopsis:	<input type="text"/>
URL de la imatge:	<input type="text"/>
Preu llibre nou:	<input type="text"/>
Preu segona mà:	<input type="text"/>
Qualitat:	<input type="text" value="nou"/>
SectorPoints:	<input type="text"/>
Stock:	<input type="text"/>

Títol	Editorial	Autor	Categoria	Qualitat	Preu	SectorPoints		
Economia	Mc Grow Hill	Cristian Larroulet	economia	nou	100.95	-	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Economia	Mc Grow Hill	Cristian Larroulet	economia	molt danyat	20.0	40	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Imatge 39: GUI Gestió Catàleg, Nou llibre

En el “Portal privat” i en el subapartat “Gestió Catàleg” es pot entrar en la secció de la imatge anterior, en la que es pot veure un formulari per afegir un nou llibre, i els llibres

existents a la llista de la part inferior, cadascun dels quals té un botó per eliminar-ho i un per editar-lo. Per l'edició s'utilitza el mateix formulari que per afegir llibres.

ADMINISTRACIÓ							
Gestió de Catàleg		Gestió de Comandes			Administració d'Usuaris		
GESTIÓ DE COMANDES							
ID	Data	Nº Articles	Total Pagat	Entrega	Estat	Usuari	
1	13/06/11 15:10:56.000	9	650.7	Domicili	entregada	Logan	Detalls/Editar
2	13/06/11 15:32:39.000	9	605.7	Domicili	pendent d'enviar	Logan	Detalls/Editar
3	13/06/11 16:30:44.000	1	0.0	Domicili	pendent d'enviar	Logan	Detalls/Editar
4	13/06/11 16:41:04.000	1	15.0	Domicili	pendent d'enviar	Logan	Detalls/Editar
5	13/06/11 16:43:07.000	9	605.7	Domicili	pendent d'enviar	Logan	Detalls/Editar

**Imatge 40: GUI Gestió comandes**

En el subapartat “Gestió de Comandes” es pot veure un llistat de les comandes fetes pels usuaris. Cada una disposa d'un botó per editar la comanda o veure'n els detalls.

EDICIÓ DE COMANDES						
ID	8					
Data	13/06/11 17:03:04.000					
Articles	6					
Total	432.85					
Lloc Entrega	Domicili					
Estat	enviada					
Usuari	Logan					
Guarda Comanda						
Llibre	Categoria	Qualitat	Quantitat	Preu	SectorPoints	
Economia - Cristian Larroulet	economia	nou	3	100.95	-	
Juego de Tronos - George R.R. Martin	literatura moderna	usat	2	15.0	30	
La Célula - Alberts	ciències	nou	1	100.0	-	

**Imatge 41: GUI Edició de comanda**

En l'anterior imatge es mostra el formulari d'edició d'una comanda. Es pot canviar només l'estat d'aquesta i també es mostra un llistat dels productes que té aquesta comanda.

Les següents imatges corresponen a pàgines de l'aplicació a les que només l'administrador pot accedir.

La primera és l'administració d'usuaris. Com en el cas dels llibres, disposem d'un formulari amb les dades de l'usuari nou que volem afegir i a sota un llistat del usuaris que hi ha actualment. Cada un té un botó per eliminar-ho i l'altre per editar-ho. La pantalla d'edició és igual que la següent amb la diferència que l'usuari es modifica i no s'afegeix.

ADMINISTRACIÓ D'USUARIS

*DNI:*   
*Nom:*   
*Cognom 1:*   
*Cognom 2:*   
*Data Naixement:*    
*Nom Usuari:*   
*Contrasenya:*   
*Domicili:*   
*Localitat:*   
*CP:*   
*Pais:*   
*Telefon:*   
*Email:*   
*IBAN:*   
*SectorPoints:*   
*Rol:*

Nom	Llinatges	DNI	Email	Telefon	SectorPoints		
Salvi	Planes Martorell	41231544H	mail@es	2301203	20	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Selbi	Dondarrion Ferly	12312312R	noLanis@setmago.com	675867832	0	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

**Imatge 42: GUI Administració Usuaris**

El registre que fa un usuari anònim amb l'opció de la capçalera és quasi idèntic a aquesta pantalla. La diferència és que no surt cap llista d'usuaris i no es pot triar el rol de l'usuari, que per defecte es posa a usuari.

## 11.2 Annex II: Script de creació de la BD

Aquest és l'script de creació de la BD, també adjuntat com a fitxer complementari a la carpeta del projecte.

```
-- MySQL Administrator dump 1.4
--
-- -----
-- Server version      5.1.56-community

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101                                     SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8 */;

/*!40014      SET          @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101      SET          @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

--
-- Create schema booksector
--

CREATE DATABASE IF NOT EXISTS booksector;
USE booksector;

--
-- Definition of table `categoria`
--
```

```
DROP TABLE IF EXISTS `categoria`;  
CREATE TABLE `categoria` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `nom` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `categoria`
```

```
--
```

```
/*!40000 ALTER TABLE `categoria` DISABLE KEYS */;  
INSERT INTO `categoria` (`id`,`nom`) VALUES  
(1,'Literatura Clàssica'),  
(2,'Literatura Moderna'),  
(3,'Humanitats'),  
(4,'Ciències'),  
(5,'Informàtica'),  
(6,'Economia'),  
(7,'Viatges i Oci'),  
(8,'Gastronomia'),  
(9,'Infantil'),  
(10,'Juvenil'),  
(11,'Arts'),  
(12,'Filologia i Formacio');  
/*!40000 ALTER TABLE `categoria` ENABLE KEYS */;
```

```
--
```

```
-- Definition of table `comanda`
```

```
--
```

```
DROP TABLE IF EXISTS `comanda`;  
CREATE TABLE `comanda` (  

```

```
`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
`dataComanda` datetime NOT NULL,  
`articles` int(10) unsigned NOT NULL,  
`total` float NOT NULL,  
`llocentrega` varchar(8) NOT NULL,  
`estat_id` bigint(20) DEFAULT NULL,  
`usuari_id` bigint(20) DEFAULT NULL,  
PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;  
  
--  
-- Dumping data for table `comanda`  
--  
  
/*!40000 ALTER TABLE `comanda` DISABLE KEYS */;  
INSERT INTO `comanda`  
(`id`,`dataComanda`,`articles`,`total`,`llocentrega`,`estat_id`,`usuari_id`) VALUES  
(1,'2011-06-18 17:29:18',5,302.85,'Tenda',4,1),  
(2,'2011-06-18 18:09:54',5,190,'Domicili',3,1);  
/*!40000 ALTER TABLE `comanda` ENABLE KEYS */;  
  
--  
-- Definition of table `estat`  
--  
  
DROP TABLE IF EXISTS `estat`;  
CREATE TABLE `estat` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `nom` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;  
  
--
```

```
-- Dumping data for table `estat`
--

/*!40000 ALTER TABLE `estat` DISABLE KEYS */;
INSERT INTO `estat` (`id`,`nom`) VALUES
(1,'enviada'),
(2,'cancelada'),
(3,'pendent d\'enviar'),
(4,'entregada');
/*!40000 ALTER TABLE `estat` ENABLE KEYS */;

--
-- Definition of table `lincomanda`
--

DROP TABLE IF EXISTS `lincomanda`;
CREATE TABLE `lincomanda` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `llibre_id` bigint(20) NOT NULL,
  `comanda_id` bigint(20) NOT NULL,
  `quantitat` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `lincomanda`
--

/*!40000 ALTER TABLE `lincomanda` DISABLE KEYS */;
INSERT INTO `lincomanda` (`id`,`llibre_id`,`comanda_id`,`quantitat`) VALUES
(1,4,1,3),
(2,2,1,2),
(3,22,2,5);
```

```
/*!40000 ALTER TABLE `lincomanda` ENABLE KEYS */;

--
-- Definition of table `llibre`
--

DROP TABLE IF EXISTS `llibre`;
CREATE TABLE `llibre` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `isbn` varchar(45) NOT NULL,
  `titol` varchar(100) NOT NULL,
  `editorial` varchar(45) NOT NULL,
  `autor` varchar(45) NOT NULL,
  `numPagines` int(10) unsigned NOT NULL,
  `dataPrimeraEdicio` datetime NOT NULL,
  `dataUltimaEdicio` datetime NOT NULL,
  `descripcio` text,
  `sinopsis` text,
  `urllmatge` varchar(150) DEFAULT NULL,
  `preuNou` float DEFAULT NULL,
  `preuSegonaMa` float DEFAULT NULL,
  `sectorPoints` int(10) unsigned DEFAULT NULL,
  `categoria_id` bigint(20) NOT NULL,
  `qualitat_id` bigint(20) NOT NULL,
  `stock` int(11) NOT NULL,
  PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `llibre`
--

/*!40000 ALTER TABLE `llibre` DISABLE KEYS */;
```



```

INSERT INTO `llibre`
(id`,`isbn`,`titol`,`editorial`,`autor`,`numPagines`,`dataPrimeraEdicio`,`dataUltimaEdicio`,`descripcio`,`sinopsis`,`urllmatge`,`preuNou`,`preuSegonaMa`,`sectorPoints`,`categoria_id`,`qualitat_id`,`stock`) VALUES
(2,'111111','Programació J2EE','Altaya','Michael Masbar',344,'2000-03-07 00:00:00','2010-04-15 00:00:00','Complet llibre sobre programació Java J2EE','Aprèn l'especificació j2ee','/images/portades/j2ee.jpg',0,15.2,70,5,2,6),
(4,'333333','Economía','Mc Grow Hill','Cristian Larroulet',708,'2009-07-24 00:00:00','2010-08-26 00:00:00','Prestigiós llibre d'economia aplicada','Larroulet ens intrudueix en el meravellós món de l'economia','/images/portades/economia.jpg',100.95,0,0,6,1,6),
(6,'17361237','Juego de Tronos','Gigamesh','George R.R. Martin',232,'2003-10-15 00:00:00','2007-02-24 00:00:00','Primera entrega de la saga de fantàstica `Cancion de Hielo i Fuego`','Tras el largo verano, el invierno se acerca a los Siete Reinos.','/images/portades/juegoTronos.jpg',22.95,15,30,2,3,5),
(13,'9184928','Momo','Cisne azul','Michael Ende',209,'2008-05-21 00:00:00','2011-04-06 00:00:00','Gran obra mestre dels contes infantils','Sinopsis del llibre','/images/portades/momo.jpg',20,0,0,9,1,12),
(14,'3333333','Economía','Mc Grow Hill','Cristian Larroulet',708,'2009-07-24 00:00:00','2010-08-26 00:00:00','Prestigiós llibre d'economia aplicada','Larroulet ens intrudueix en el meravellós món de l'economia','/images/portades/economia.jpg',0,20,40,6,5,3),
(15,'254231','La Célula','OMEGA','Alberts',1200,'1979-07-19 00:00:00','2010-03-01 00:00:00','La bíblia de la biologia','Alberts presenta aquest recull formidable de biologia.','/images/portades/celula.jpg',100,0,0,4,1,10),
(16,'08022005','Fonaments de la Bioquímica','ANAYA','Lehninger',1200,'1965-04-16 00:00:00','2011-06-12 00:00:00','Bíblia de la Bioquímica','the greatest knowledge compilation of the sector ever.','/images/portades/leninguer.jpg',200,0,0,4,1,20),
(22,'999777888','Cocina Fresca','Centaurio','Karlos Arguiñano',113,'1990-01-01 00:00:00','2010-01-01 00:00:00','Llibre de cuina del famós cuiner espanyol.','/images/portades/cuina.jpg',38,0,0,8,1,10);
/*!40000 ALTER TABLE `llibre` ENABLE KEYS */;

--
-- Definition of table `qualitat`
--

DROP TABLE IF EXISTS `qualitat`;
CREATE TABLE `qualitat` (

```

```
`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
`nom` varchar(45) NOT NULL,  
PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `qualitat`  
--
```

```
/*!40000 ALTER TABLE `qualitat` DISABLE KEYS */;  
INSERT INTO `qualitat` (`id`,`nom`) VALUES  
(1,'nou'),  
(2,'semi-nou'),  
(3,'usat'),  
(4,'danyat'),  
(5,'molt danyat');  
/*!40000 ALTER TABLE `qualitat` ENABLE KEYS */;
```

```
--  
-- Definition of table `rol`  
--
```

```
DROP TABLE IF EXISTS `rol`;  
CREATE TABLE `rol` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `nom` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `rol`  
--
```

```
/*!40000 ALTER TABLE `rol` DISABLE KEYS */;
INSERT INTO `rol` (`id`,`nom`) VALUES
(1,'usuari'),
(2,'manteniment'),
(3,'administrador');
/*!40000 ALTER TABLE `rol` ENABLE KEYS */;

--
-- Definition of table `usuari`
--

DROP TABLE IF EXISTS `usuari`;
CREATE TABLE `usuari` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `nom` varchar(45) NOT NULL,
  `cognom1` varchar(45) NOT NULL,
  `cognom2` varchar(45) DEFAULT NULL,
  `dataNaixement` datetime DEFAULT NULL,
  `nomUsuari` varchar(45) NOT NULL,
  `contrasenya` varchar(45) NOT NULL,
  `domicili` varchar(100) NOT NULL,
  `cp` varchar(5) NOT NULL,
  `pais` varchar(45) NOT NULL,
  `telefon` varchar(9) DEFAULT NULL,
  `email` varchar(45) NOT NULL,
  `iban` varchar(45) NOT NULL,
  `sectorPoints` int(10) unsigned NOT NULL DEFAULT '0',
  `dni` varchar(20) NOT NULL,
  `rol_id` bigint(20) DEFAULT NULL,
  `localitat` varchar(45) NOT NULL,
  PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

```

--
-- Dumping data for table `usuari`
--

/*!40000 ALTER TABLE `usuari` DISABLE KEYS */;
INSERT INTO `usuari`
(id`,`nom`,`cognom1`,`cognom2`,`dataNaixement`,`nomUsuari`,`contrasenya`,`domicili`,`cp`,`pais`,`telefon`,`email`,`iban`,`sectorPoints`,`dni`,`rol_id`,`localitat`) VALUES
(1,'Jordi','Planes','Martorell','1987-04-18 00:00:00','admin','1234','c/ estany 34','12312','Espanya','2301203','mail@es','213123',10,'41231544H',3,'Manacor'),
(2,'Selbi','Dondarrion','Ferly','1987-04-18 00:00:00','manteniment','1234','av. trident 3','12312','Set Regnes','675867832','noLanis@setmago.com','1231239124124',0,'12312312R',2,'Rius grans'),
(4,'Miquel','Fabregues','Cambrius','1986-09-22 00:00:00','usuari','1234','carrer peixet 4','07525','Espanya','876876876','god4lv@gmail.com','132120940129',100,'12312312 G',1,'Cala Anguila'),
(6,'Martí','Grupau','Calaped','1990-01-01 00:00:00','luter','king','c/ mosquell 13','93428','Espanya','999999','marti@marti.es','7897987',200,'11122233U',1,'Barcelona');
/*!40000 ALTER TABLE `usuari` ENABLE KEYS */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

```