

Enginyeria Tècnica en Informàtica de Sistemes

Treball Fi de Carrera:

Anàlisi, disseny i implementació d'una aplicació  
de comerç electrònic usant J2EE

## Memòria del projecte

Albert Grau Aymerich

[agraua@uoc.edu](mailto:agraua@uoc.edu)

15/01/2007

## RESUM

Aquest treball de fi de carrera consisteix en el disseny i implementació d'una botiga web bàsica. Aquesta botiga ha de permetre mantenir un inventari d'articles a la venda agrupats en categories, i la seva posada disposició dels clients que hi accedeixen mitjançant un navegador web. Tècnicament., la implementació ha de basar-se en l'especificació J2EE de Sun Microsystems, amb una clara separació de les capes de presentació, negoci i dades. S'ha optat per l'utilització de patrons de disseny que ha permès aconseguir un reduït acoblament entre la capa de presentació i negoci alhora que segurament ha evitat part dels errors i complicacions innecessàries que podien haver sorgit per falta d'experiència.

Cal tenir present en tot moment que aquest desenvolupament és un treball acadèmic. L'objectiu a desenvolupar-lo no ha estat tant construir un producte exportable comercialment, sinó arribar a aprendre el funcionament de la tecnologia J2EE.

Finalment, cal tenir present que aquesta implementació no és, ni pretén ser un implementació complerta de les funcions que ha de tenir un lloc web de comerç electrònic. La complexitat i el nombre de característiques que ha de tenir una aplicació d'aquest tipus fa inviable el seu desenvolupament per part d'una única persona amb poques hores per dedicar-s'hi.

## Taula de continguts

### 1. Introducció

- 1.1. Justificació del TFC
- 1.2. Descripció i objectius
- 1.3. Planificació del projecte
  - 1.3.1. Abast del projecte
  - 1.3.2. Recursos
  - 1.3.3. Descripció
  - 1.3.4. Planificació del projecte

### 2. Anàlisi

- 2.1. Part pública
  - 2.1.1. Accés a la botiga
  - 2.1.2. Registre d'un nou usuari
  - 2.1.3. Pàgina principal de navegació
  - 2.1.4. Informació del llibre
  - 2.1.5. Carret de la compra
- 2.2. Part privada
  - 2.2.1. Accés a l'àrea privada
  - 2.2.2. Pantalla d'administració
- 2.3. Classes del domini
  - 2.3.1. Diagrama de classes
  - 2.3.2. Descripció de les classes
- 2.4. Casos d'ús
  - 2.4.1. Diagrama de classes d'ús
  - 2.4.2. Descripció de les classes d'ús
- 2.5. Anàlisi dinàmica

### 3. Disseny

- 3.1. Patrons
  - 3.1.1. Arquitectura de presentació i control
  - 3.1.2. Arquitectura del model
- 3.2. Diagrama de classes de disseny
- 3.3. Diagrames de col·laboració
  - 3.3.1. Cas d'ús de cerca de productes
  - 3.3.2. Cas d'ús d'editar el carret
  - 3.3.3. Cas d'ús de tramitar una comanda
- 3.4. Disseny de la persistència
  - 3.4.1. Diagrama de persistència
  - 3.4.2. Hibernate
- 3.5 Disseny de les interfícies d'usuari
  - 3.5.1. Pantalla de login
  - 3.5.2. Formulari de registre
  - 3.5.3. Botiga
  - 3.5.4. Pantalla de descripció de l'article
  - 3.5.5. Carret de la compra

### 4. Implementació

- 4.1. Estructura general

4.2. Característiques de la implementació

4.3. Estructura de la implementació

4.3.1. com.albert.libreria.dades

4.3.1. com.albert.libreria

4.3.1. com.albert.libreria.struts

4.3.1. com.albert.libreria.struts.form

4.3.1. com.albert.libreria.struts.action

4.4. Action Forms i Action Beans

4.5. Objectes valor

4.6. Session Façade

4.7. Internalització

5. Conclusions

5.1. Objectius assolits

5.2. Objectius no assolits

6. Bibliografia

---

# 1. Introducció

---

## 1.1 Justificació del TFC

L'erupció d'Internet ha suposat, entre d'altres coses, un augment espectacular de l'activitat en la indústria del programari. Tot plegat ha fet que en els darrers anys hagin sorgit tota una sèrie de tecnologies que competeixen a l'hora de captar l'atenció dels desenvolupadors que treballen en aquest entorn. Alguns exemples d'aquestes tecnologies són : desenvolupaments de l'àmbit del programari lliure, com ara PHP, infraestructures purament propietàries, com a .NET, i finalment, l'entorn J2EE, proposat per Sun Microsystems i recolzat pels pesos pesants de la indústria, com són Oracle, BEA Systems i IBM. No entrarem a valorar quines són millor o pitjors i ens centrarem amb l'entorn J2EE que serà el tema central d'aquesta memòria.

D'entrada direm que l'especificació J2EE té una molt bona acceptació dins de la indústria del programari gràcies a la seva portabilitat, robustesa, escalabilitat i relativa rapidesa de desenvolupament. A la vegada disposa d'una comunitat de desenvolupadors molt activa que fa que no sigui una especificació congelada, si no que està en constant evolució.

No obstant estem parlant d'aplicacions que poden ser força complexes donat el nombre de tecnologies que inclouen. L'ús de patrons de disseny pretén fomentar solucions que prèviament han s'ha mostrat exitoses en resoldre problemes semblants. En aquest sentit, un patró documenta i explica un problema corrent que qualsevol desenvolupador es pot trobar quan dissenya o implementa una aplicació. La documentació del patró proposa i discuteix la millor solució pràctica per aquest problema.

En aquest TFC, un estudiant amb coneixements de Java, però sense experiència prèvia en aplicacions distribuïdes, dissenya i implementa un exemple típic d'aplicació distribuïda consistent en una aplicació de comerç electrònic utilitzant la tecnologia J2EE i amb el seguiment de patrons de disseny.

## 1.2 Descripció i Objectius

L'objectiu principal d'aquest projecte es l'aprenentatge i familiarització de la tecnologia J2EE i els seus entorns de desenvolupament i producció. I en cap moment perdrem de vista aquest enfocament. O dit en altre paraules, l'objectiu al desenvolupar-lo no ha estat tant construir un producte explotable comercialment, sinó arribar a aprendre i entendre el funcionament de la tecnologia J2EE .

L'altre objectiu, més específic, és el de la implementació d'una aplicació de botiga virtual, amb dues àrees ben diferenciades:

- Una àrea privada on els administradors podran dur a terme les seves taques administratives de la botiga com ara veure els usuaris, gestionar els llibres, gestionar les comandes , etc..

- 
- Una àrea pública on els clients hi accedeixin. Dins d'aquesta àrea els clients podran consultar la informació dels diferents llibres classificats per categories a la vegada que també podran realitzar comandes. La compra es gestionarà mitjançant un carret de compra virtual i els clients hi accediran mitjançant un navegador estàndard.

Tècnicament, la implementació ha de basar-se en [l'especificació J2EE de Sun Microsystems](#) , com ara :

- Desenvolupament de servlets, ús de Tomat (contenedor de Servlets).
- Tecnologia JSP (Java Server Pages).
- Tecnologia JNI (Java Native Interface).
- Desenvolupament amb components: Enterprise Java Beans (EJBs). Ús d'un servidor d'aplicacions J2EE, per exemple JBoss.
- Estudi i ús de patrons de disseny, aplicant-los en l'arquitectura J2EE.
- Comunicació amb bases de dades relacionals (per exemple MySQL).
- Ús de sistemes d'autenticació amb aplicacions Java (JAAS).
- Struts
- Temes involucrats amb l'arquitectura J2EE en general.

I tot plegat, encaixant-ho dins d'una arquitectura de desenvolupament MVC (Model-Vista-Controlador).

## **1.3 Planificació del projecte.**

### **1.3.1. Abast del projecte**

Funcionalitat general d'una botiga virtual, tenint en consideració la limitació de temps imposada en el seu desenvolupament. Es considera prioritari la qualitat del disseny enfront als detalls d'implementació o l'estètica de la interfície d'usuari.

### **1.3.2. Recursos**

Aquest es durà a terme per una sola persona. Tenint en compte que disposem de menys de quatre mesos , es calcula un temps aproximat de 150h invertides en aquest projecte, sense comptar amb el temps destinat amb la formació.

Requeriments de maquinari:

-Pentium IV 2.40 GHz, 768 MB de RAM

Requeriments de programari:

- Sistema Operatiu Linux (distribució Ubuntu)

- JDK 1.4

- JBoss com a implementació de referència J2EE 1.3.

- mySQL com a base de dades.

- Eclipse o NetBeans (està per decidir) com a eina de desenvolupament en Java.

### 1.3.3. Riscos del projecte

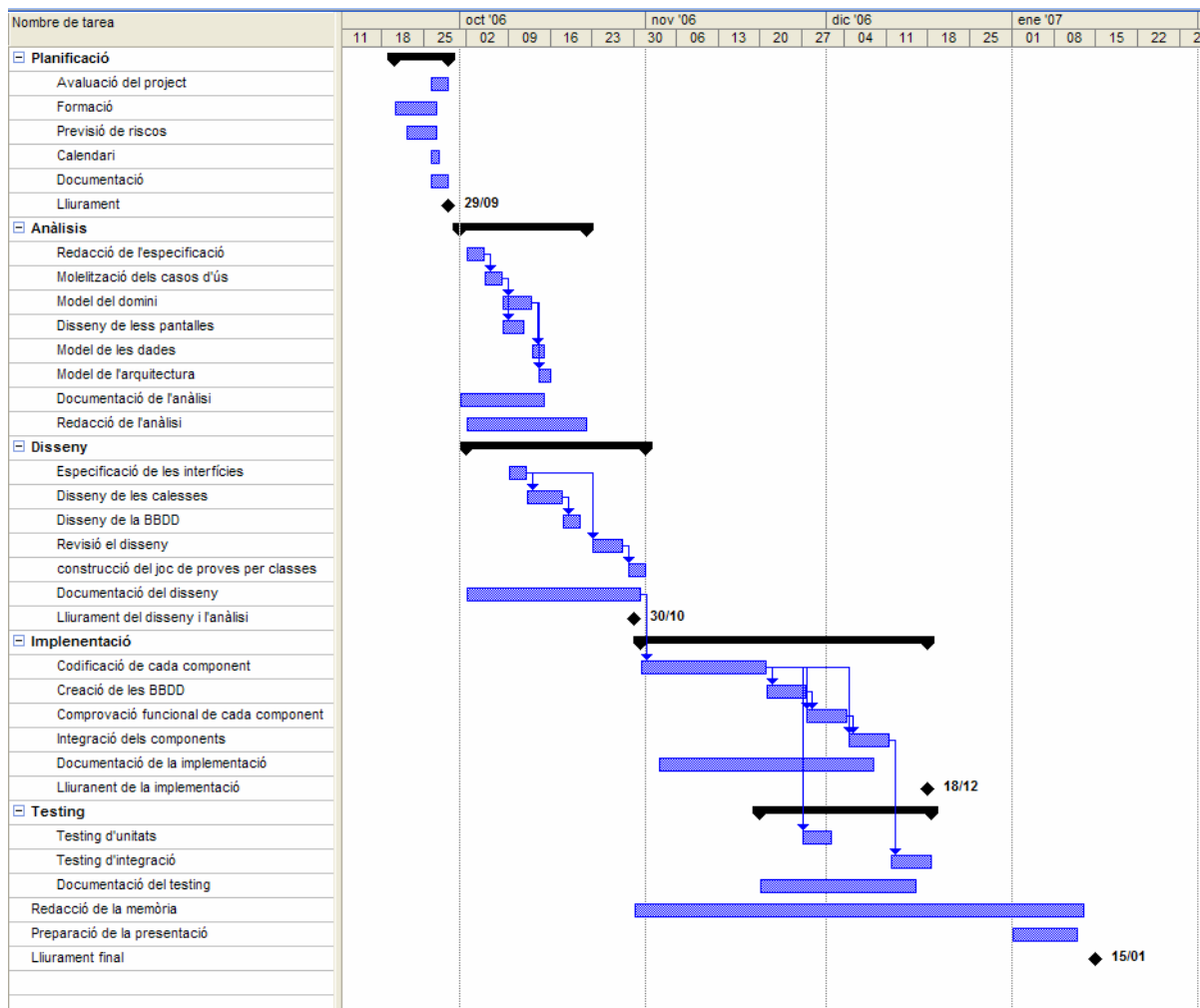
El risc principal a tenir molt en compte es la inexperiència en implementació d'aplicacions distribuïdes unit a la poca coneixença de l'arquitectura J2EE. I per tant, ens pot conduir a dissenys incorrectes, ineficaços o difícils d'implementar, així com a alentir la implementació. Per a mitigar aquests riscos serà necessari des del principi del projecte dedicar un temps addicional a la formació, mitjançant manuals i llibres online. També s'espera compensar la inexperiència amb el seguiment de solucions proposades per altres, en forma de patrons.

### 1.3.4. Calendari

El calendari anirà marcat per les dates dels lliurables:

- 29 de setembre: pla del projecte.
- 30 d'Octubre: document d'anàlisi i disseny.
- 18 de Desembre : lliurament final, consistent en una memòria (màxim 60 planes) i unapresentació (màxim 20 diapositives).

El calendari de les fases i tasques es detalla seguidament en un diagrama de Gantt.



Com es pot comprovar, s'ha seguit aproximadament el mètode RUB. Els temps estimats de cada tasca, en aquestes alçades del projecte, es poden considerar provisionals .

---

## **2. Anàlisi**

---

### **2.1. Part Pública:**

#### **2.1.1. Accés a la botiga**

Quan un usuari es connecti al nostre lloc web, es mostrarà un pantalla d'entrada amb un missatge de benvinguda. També es mostrarà un quadre de diàleg on es demanarà que el client s'identifiqui amb un nom d'usuari i una contrasenya.

En el cas que el client encara no estigui donat d'alta, el sistema li oferirà la possibilitat de registrar-me.

Un cop introduïts el nom i la contrasenya, el sistema validarà si es tracta d'un usuari registrat i en cas afirmatiu, se li mostrarà la pàgina d'inici de la botiga.

#### **2.1.2. Registre d'un nou usuari**

El sistema mostrarà una pantalla que constarà d'un formulari on es demanarà a l'usuari les dades necessàries par a donar-se d'alta a la botiga. Algunes d'aquestes dades seran el nom, l'adreça de correu electrònic, adreça postal, contrasenya etc.

Seguidament el sistema validarà les dades introduïdes. Si tot es correcte el sistema li mostrarà la pantalla d'accés a la botiga.

#### **2.1.3. Pàgina principal de navegació**

La pàgina principal de la botiga constarà d'un menú amb les següents opcions:

- Llistar categories
- Llistar el catàleg de llibres
- Veure el Carret
- Sortir de l'aplicació

També constarà d'un cercador, on es podran fer cerques per autor, títol del llibre o codi ISBN.

Tant si fem un cerca, com si clickem sobre una categoria, a la part central de la pàgina apareixeran els llibres que compleixen els criteris de cera. Per a cada llibre ens apareixeran dos enllaços , que ens serviran o per afegir el llibre al carret de la compra o per a consultar la informació complerta del llibre.

Si clickem al catàleg, ens apareixeran tots els llibres que tenim donats d'alta.



#### **2.1.4. Informació del llibre**

Aquesta pantalla constarà d'una imatge del llibre ( si es disposa) i tota aquella informació referent a l'autor, ISBN, preu, descripció llarga. Si es el llibre buscat, es podrà afegir al carret de la compra. Si no, es podrà continuar buscant per la web.

Si s'afegeix al carret, automàticament s'actualitzaran les dades del preu i nombre de ítems del carret.

#### **2.1.5. Carret de la compra**

El carret de la compra contindrà en tot moment la llista de productes que en un moment donat el client vol comprar i es podrà accedir des de totes les pantalles de la web.

Aquesta plana constarà d'una taula amb tantes files com productes s'ha afegit. I cada fila contindrà

- Títol del llibre
- Preu
- Quantitat
- Total

El client podrà modificar la quantitat . A sota de la taula es mostrarà un una sèrie de botons que permetran a l'usuari:

- Actualitzar totals
- Seguir comprant
- Tancar la compra i buidar el carret
- Buidar el carret

Un cop s'hagi modificat la quantitat i es premi actualitzar totals, el sistema recalculerà els nous imports. En el cas que la nova quantitat sigui 0, el sistema eliminarà del carret l'article en qüestió.

### **2.2. Part privada:**

#### **2.2.1. Accés a l'àrea privada**

Per accedir a l'àrea privada l'usuari administrador s'haurà d'autenticar. Per tant, en aquesta pantalla l'administrador entrarà un nom d'usuari i una contrasenya.

#### **2.2.2. Pantalla d'administració**

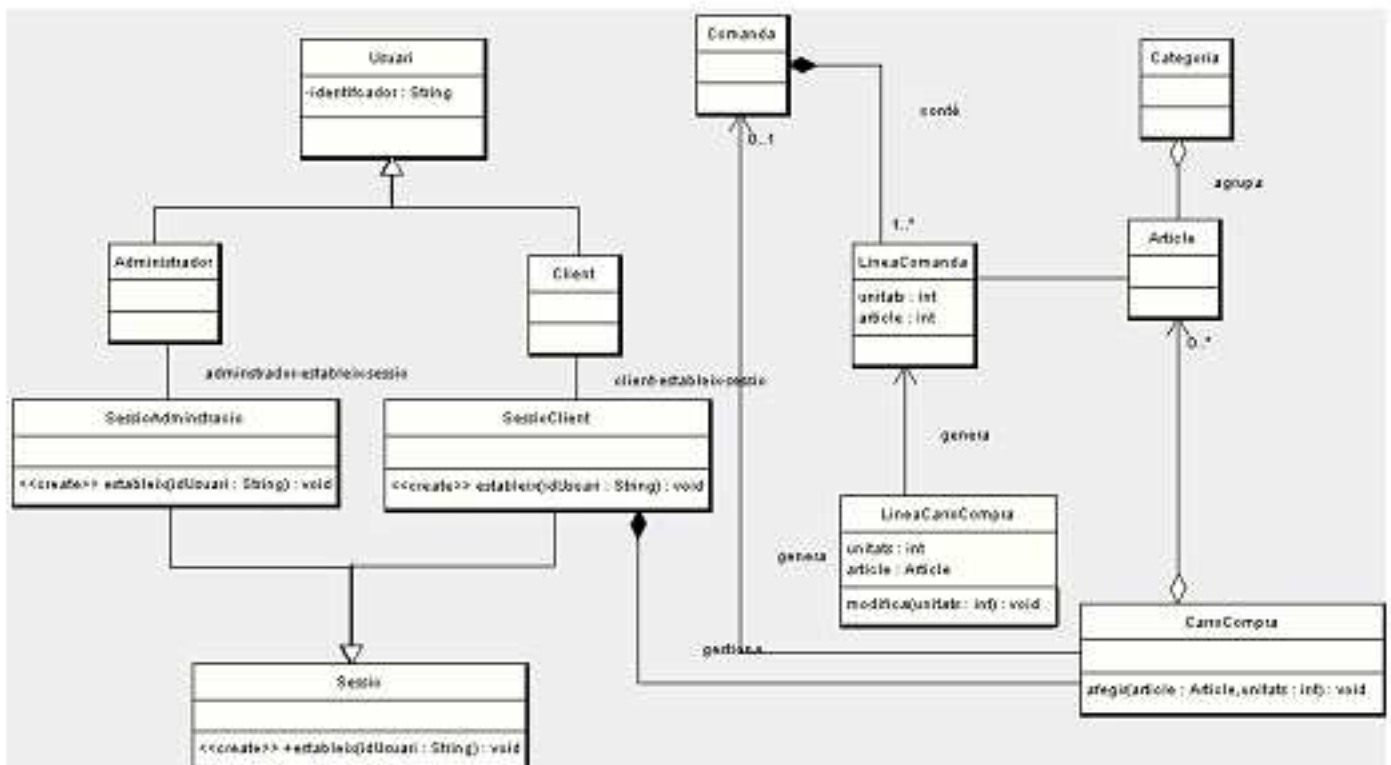
Des de aquesta pantalla l'administrador podrà realitzar les següents tasques:

- Navegar per la botiga (com administrador)
- Afegir, esborrar i editar una nova categoria
- Afegir , esborrar i editar un nou llibre
- Consultar les comandes

L'administrador podrà navegar per les mateixes pantalles (menys el carret de la compra) que els clients i a més podrà afegir,eliminar o editar en cada moment categories i llibres.

## 2.3. Classes del Domini

### 2.3.1. Diagrama de Classes



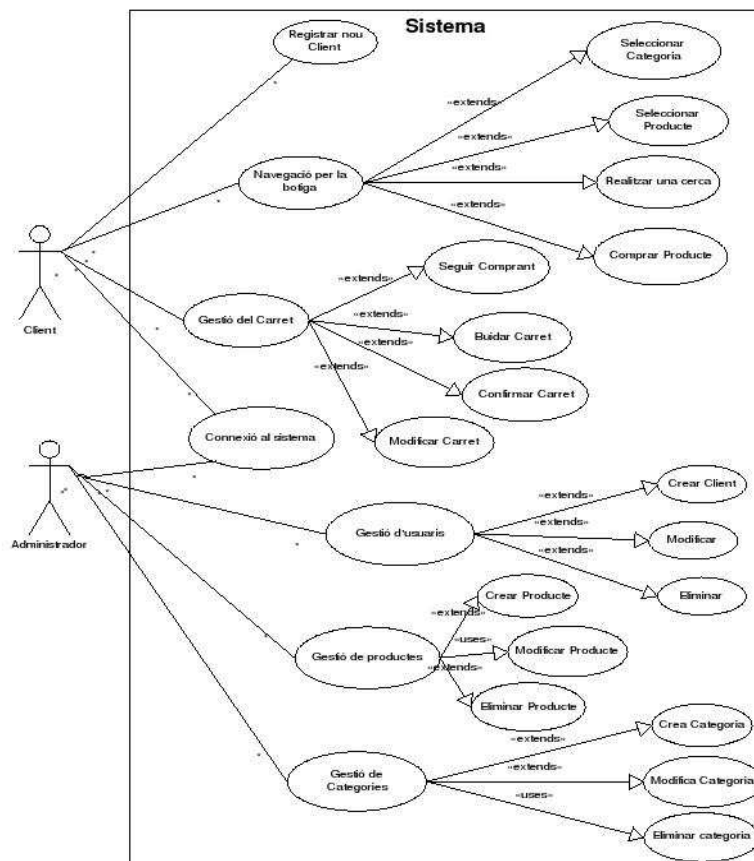
### 2.3.2. Descripció de les Classes

- **Usuari.** Classe abstracta que contindrà tots aquells atributs i mètodes comuns entre la classe Administrador de sistema i la classe Client de la botiga.
- **Administrador.** Usuari de la botiga amb uns certs privilegis. Pot accedir a l'àrea privada.
- **Client.** Usuari dins de la botiga. Està registrat i té permís per a fer compres per la botiga.
- **Sessió.** Classe abstracta. Contindrà aquelles funcionalitats bàsiques per establir un sessió al sistema.

- **SessioAdminstracio.** Classe derivada de l'anterior i que conté funcionalitats pròpies d'una sessió al sistema amb certs pri vilegis.
- **SessioClient.** Classe derivada de sessió i que conté funcionalitats pròpies d'una sessió relacionada amb una compra.
- **Categoria.** Classe que classifica i agrupa els articles de la botiga
- **Article.** Cada un dels productes disponibles per a la venda.
- **CarroCompra.** Classe conté l'estat de la compra del client.
- **LineaCarroCompra.** Classe que fa referència a cada un dels articles que conté la classe Carro compra.
- **Comanda.** Classe que representa una comanda en ferm feta i confirmada pel client.
- **LineaComanda.** Cada un dels ítems que formen part de la comanda.

## 2.4. Casos d'ús

### 2.4.1. Diagrama dels casos d'ús



### 2.4.2. Descripció dels casos d'ús

Els casos d'ús més importants i que poden merèixer un tractament més detallat en aquesta etapa del projecte són:

- Navegació per la botiga
- Gestió del carret de la compra

La resta de casos són sempre repeticions del mateix model: actualitzacions d'un conjunt de dades (altes, baixes, modificacions i consultes) que no han de presentar en principi dificultats de plantejament ni de desenvolupament.

#### Gestió del carret de la compra

<b>Nom del Cas d'Ús</b> : <i>Gestió del Carret de la compra</i>		<b>Codi:</b>
<b>Autor</b>	Albert Grau Aymerich	
<b>Descripció</b>	Cas d'ús que descriu la interacció del client amb el carret de la compra	
<b>Prioritat (1.. 10) 1 com a màxim. 10 com a mínim.</b>	2	
<b>Actor/s</b>	Client	
<b>Precondicions:</b>	<b>Casos d'Ús</b>	El client s'ha d'autenticar al sistema El client està dins d'una sessió de compres
	<b>D'altres</b>	
<b>Postcondicions</b>		
<b>Pseudocodi (flux normal)</b>	<p>1. El sistema mostrarà al usuari el contingut del carret de la compra.. L'usuari pot escollir entre confirmar la compra, descartar la compra, modificar la compra o seguir comprant.</p> <p>2. Un cop confirmada la compra el sistema li mostrarà un resum de tots els articles que ha comprat</p>	
<b>Extensions</b>	<p>1.a Si l'usuari selecciona tornar a la navegació el sistema li retorna de nou el formulari de cerca i catàleg.</p> <p>1.b. Si l'usuari selecciona buidar el carret, el sistema elimina tot el seu contingut.</p> <p>2.c Si l'usuari modifica les unitats comprades, i les noves unitats són superiors a zero s'actualitzen els imports de la compra.</p> <p>2.d Si l'usuari modifica les unitats i les noves unitats són zero el sistema elimina aquest ítem del carret i actualitza els imports.</p>	

**Navegació per la botiga**

<b>Nom del Cas d'Ús</b> : Navegació per la botiga		<b>Codi:</b>
<b>Autor</b>	Albert Grau Aymerich	
<b>Descripció</b>	Cas d'us que descriu la navegació del client per la botiga	
<b>Prioritat (1.. 10) 1 com a màxim. 10 com a mínim.</b>	2	
<b>Actor/s</b>	Client	
<b>Precondicions:</b>	<b>Casos d'Ús</b>	El client s'ha d'autenticar al sistema
	<b>D'altres</b>	
<b>Postcondicions</b>		
<b>Pseudocodi (flux normal)</b>	<p>1. El sistema mostrarà al usuari el contingut del catàleg de llibres del que disposa amb les novetats del mes.</p> <p>2. L'usuari procedeix a navegar per la botiga. Les opcions són:</p> <ul style="list-style-type: none"> <li>• Consultar una categoria</li> <li>• Consultar el catàleg general de llibres</li> <li>• Realitzar una cerca</li> <li>• Seleccionar un llibre</li> <li>• Afegir un llibre al carret de la compra</li> <li>• Seguir navegant per la botiga</li> </ul>	
<b>Extensions</b>	<p>2.a Si l'usuari decideix triar una categoria, el sistema mostrarà tots els llibres inclosos en dins d'aquesta categoria.</p> <p>2.b Si l'usuari realitza una cerca, el sistema mostrarà tots els llibres que compleixin els criteris de cerca.</p> <p>2.c Si l'usuari selecciona un llibre del catàleg pot:</p> <ul style="list-style-type: none"> <li>• Consultar la informació que hi ha disponible d'aquest llibre.</li> <li>• Afegir el llibre al carret de la compra.</li> </ul> <p>2.d L'usuari pot continuar navegant</p>	

**Connexió al sistema**

<b>Nom del Cas d'Ús</b> : Connexió al sistema		<b>Codi:</b>
<b>Autor</b>	Albert Grau Aymerich	
<b>Descripció</b>	Cas d'us que descriu com un usuari es connecta al sistema	
<b>Prioritat (1.. 10) 1 com a màxim. 10 com a mínim.</b>	2	
<b>Actor/s</b>	Client i administrador	
<b>Precondicions:</b>	<b>Casos d'Ús</b>	El usuari ha d'estar donat d'alta al sistema

	<b>D'altres</b>	
<b>Postcondicions</b>		
<b>Pseudocodi (flux normal)</b>	<ol style="list-style-type: none"> <li>1. L'usuari introdueix una direcció de correu i una contrasenya. Seguidament en prem el botó de connectar.</li> <li>2. El sistema comprova que els camps contenen un valor correcte.</li> <li>3. El sistema valida a l'usuari.</li> </ol>	
<b>Extensions</b>	<p>3.a Si les dades són correctes el sistema li presenta la pantalla principal de navegació de la botiga.</p> <p>3.b Si les dades no corresponen a cap usuari del sistema, el sistema li mostra altre cop la pantalla de login i li informa que les dades que ha introduït no són correctes.</p>	

### Registre d'un nou usuari

<b>Nom del Cas d'Ús</b> : Registre si un nou usuari		<b>Codi:</b>
<b>Autor</b>	Albert Grau Aymerich	
<b>Descripció</b>	Cas d'us que descriu el mecanisme per a registrar-se com a usuari de la botiga	
<b>Prioritat (1.. 10) 1 com a màxim. 10 com a mínim.</b>	1	2
<b>Actor/s</b>	Client i administrador	
<b>Precondicions:</b>	<b>Casos d'Ús</b>	Cap
	<b>D'altres</b>	
<b>Postcondicions</b>	S'ha donat d'alta un usuari nou	
<b>Pseudocodi (flux normal)</b>	<ol style="list-style-type: none"> <li>1. L'usuari omple el formulari de registre que consta de :  Nom i cognoms  Adreça de correu electrònic  Adreça postal  Contrasenya</li> <li>2. Els sistema verifica si totes les dades són correctes.</li> <li>3. El sistema crea un nou usuari usant l'adreça de correu electrònic com a identificador d'usuari.</li> <li>4. El sistema mostra la pantalla de login.</li> </ol>	

### Gestió de Categories, Gestió de Productes.

Aquests dos casos d'ús són molt semblants. L'actor és l'administrador. Un cop l'usuari Administrador està degudament autenticat, el sistema li mostra la possibilitat de crear productes o categories noves, modificar-ne o eliminar-ne.

En el tres casos el sistema mostrarà un formulari que servirà per a introduir les dades dels nous i també servirà per a mostrar les dades dels que ja estan creats .

En el cas que es vulgui modificar o eliminar un registre, primer s'haurà de fer una cerca i presentar un llistat dels registres que compleixen amb els criteris.

Si es tracta de categories la cerca es farà per nom de categoria

Si es tracta de productes (llibres es aquest cas) la cerca es farà per

Autor

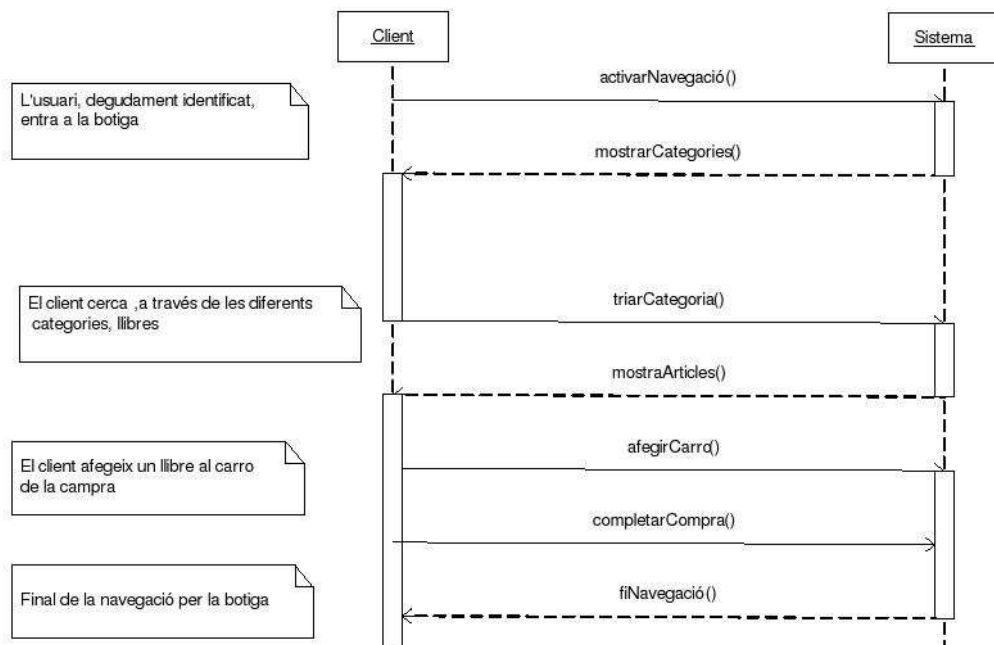
Títol

ISBN

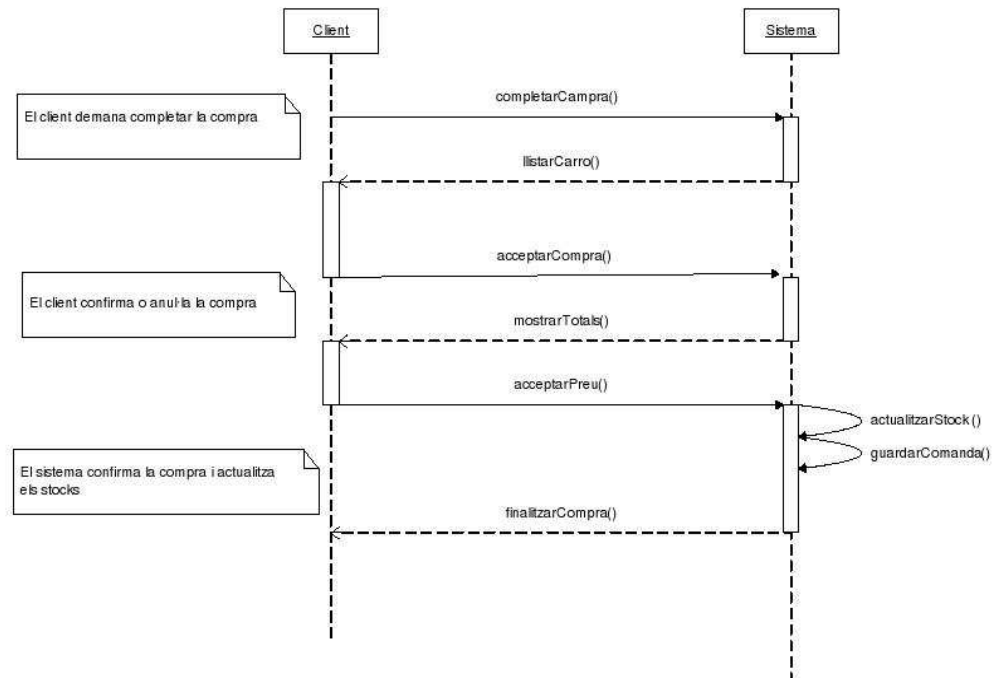
## 2.5. Anàlisi Dinàmica

A partir dels dos casos d'ús més importants que hem identificat, com ara són la navegació per la botiga i la gestió del carret, presentem el DSS ( Diagrames de Seqüència del Sistema).

### •Navegació per la botiga



- **Gestió del Carret de la Compra**





## 3 Disseny

### 3.1. Patrons

Sun Microsystems proporciona un conjunt de patrons que poden ser usats en el context del disseny d'aplicacions Java 2 Enterprise Edition, J2E. Els patrons de disseny J2EE consisteixen en solucions recurrents i documentades de problemes comuns de disseny d'aplicacions J2EE. Tenint en compte això, podem escollir els patrons de disseny que seguirem, tant en el que respecta a la presentació i control, persistència de dades o capa de negoci.

En quant a la capa de negoci és important utilitzar un patró que minimitzi el nombre de crides entre components i que desacobli capes. Per tant, és recomanable utilitzar el patró Session Façade.

En quant a la capa de dades és important aplicar algun patró que ocult i les consultes a la base de dades com ara el patró DAO.

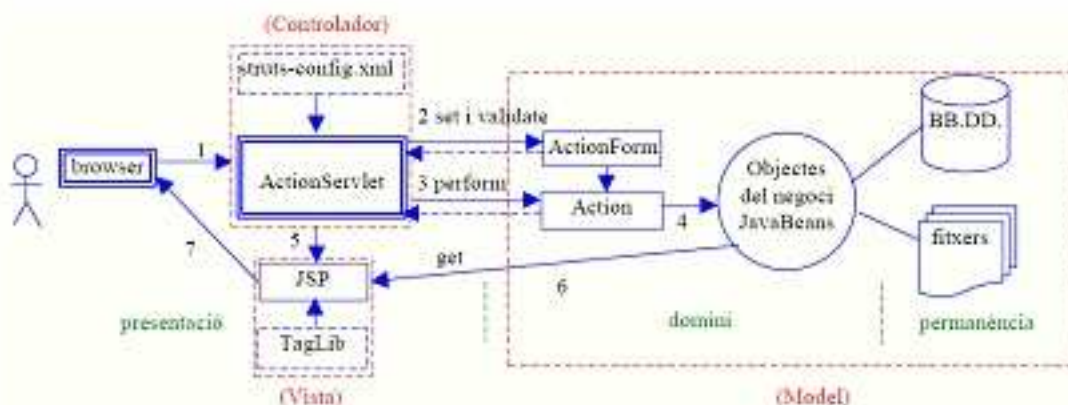
A continuació es descriurà el funcionament d'aquests patrons.

#### 3.1.1. Arquitectura de presentació i control

Basarem la presentació de les pantalles i el control de flux en el conegut model **MCV (Model-vista-controlador)**. En el patró de disseny MVC, el flux de l'aplicació està dirigit per un controlador central. El controlador delega sol·licituds a un manejador apropiat. Els manejadors estan units a un Model i cada manejador actua com un adaptador entre la sol·licitud i el Model.

Ens basarem en la infraestructura Struts, que ens proporcionarà la part de controlador i eines per a construir la presentació (Vistes). Totes les vistes seran generades usant components JSP.

El marc de treball Struts implica definir una classe ActionForm per a cada formulari d'entrada necessari en la nostra aplicació. Llavors, i mitjançant arxius (ActionMapping), el servlet controlador realitza automàticament els serveis que li indiquem.



L'estructura general és la següent:

- L'usuari fa una petició d'acció amb paràmetres a través del browser.
- La petició és processada per un servlet amb l'ajut d'un fitxer de configuració escrit en xml. El servlet construeix un ActionForm amb tots els paràmetres de la petició de l'usuari i crida a el mètode validate(). El sevlet és únic.
- Si la validació es positiva crida al mètode perform(), passant-li el ActionForm.
- L'Action realitza l'acció demanda actualitzant l'estat dels objectes (JavaBeans) del negoci i informant al sevlet de quin és el forwar a realitzar.
- El servlet realitza el forward demanat pel Action.
- S'executa la JSP i aquesta consultarà l'estat dels objectes de negoci per representar la part dinàmica.
- L'usuari rep la resposta en HTML+CSS+javascript

Seguin aquest model no ens caldrà preocupar de controlar nosaltres mateixos el flux del programa per codi. Per exemple en la validació d'un usuari, tindrem un formulari amb dos camps (email i password) i un botó. Aquests dos camps estaran mapejats dintre d'una classe ActionForm. Al enviar el formulari de login, el servlet controlador de l'Struts carregarà a la classe ActionForm el valor dels camps dins de les seves propietats respectives mitjançant els mètodes "get" i "set" que ha d'implementar tota ActionForm. Seguidament mirarà dins de l'arxiu struts-config, que és on es guarden totes les accions mapejades amb el seu corresponent ActionForm. A partir d'aquí es la classe action l'encarregada de cridar a un bean i fer la consulta a la base de dades, i en funció de si existeix l'usuari donat d'alta o no, mostrar la pantalla d'entrada a l'aplicació o altra cop la pantalla de login indicant que no està donat d'alta.

### **3.1.2. Arquitectura del model (lògica de negoci )**

La part del Model dins del patró MVC es basarà en l'ús d'EJB d'entitat, unint persistència manegada per contenidor sempre que sigui possible. Per definir la resta de lògica de negoci ens basarem en el patró **façana de sessió (sessinn façade)**. Els controladors de façana l'implementaran en forma de EJB de sessió sense estat que encapsulen la complexitat de les interaccions dintre els objectes de negoci participants en el flux de treball. El session façade manega els objectes de negoci i

proporciona un servei d'accés uniforme als clients. És a dir, els clients ( en aquest les pàgines JSP) no tractaran amb els EJB ni amb la complexitat dels accessos remots , ni amb JDBC , si no que tractaran amb col·leccions i Objectes-Valor .

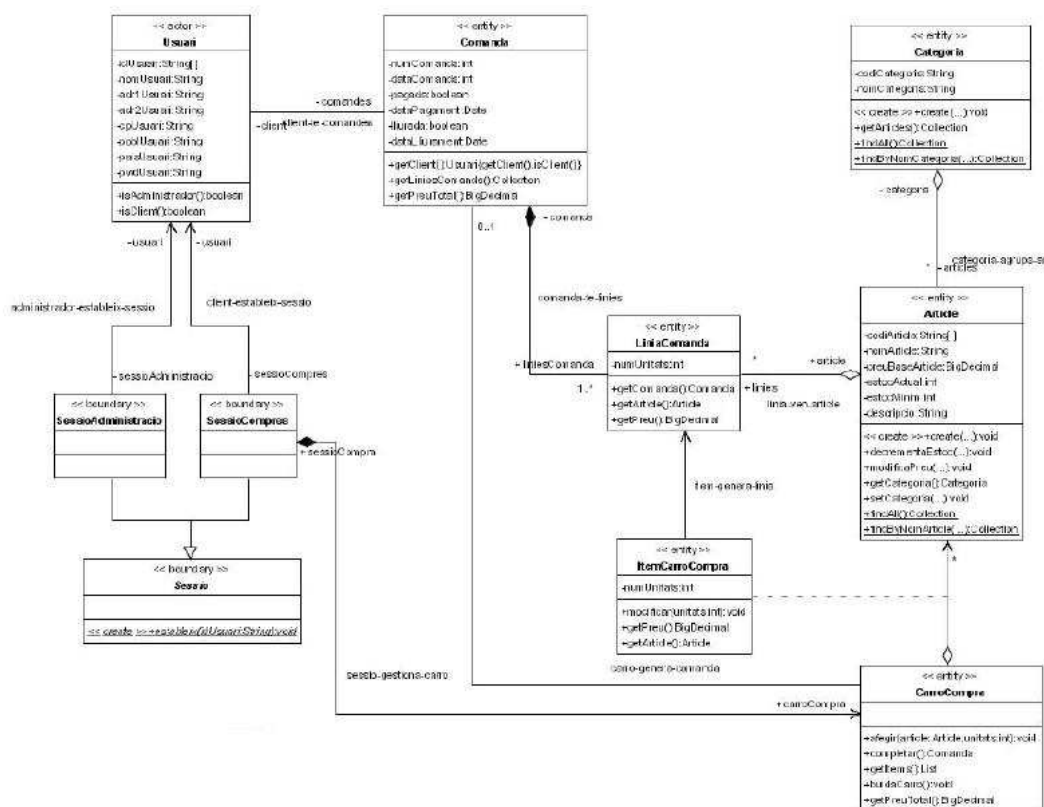
En el nostre cas utilitzarem i degut a que la nostra aplicació tindrà poc negoci, crearem un bean que anomenarem FacanaAplica i en ell encapsularem tots els accessos a dades persistents. Seguint amb l'exemple anterior, la classe action encarregada del login, preguntarà al nostre bean de façana si l'usuari existeix o no i d'aquesta manera poder executar una acció o una altra. Per altra banda no li caldrà preocupar de res més.

### 3.2. Diagrama de classe de disseny

El diagrama de disseny es pràcticament el mateix que es va presentar a la fase d'anàlisi amb una diferència:

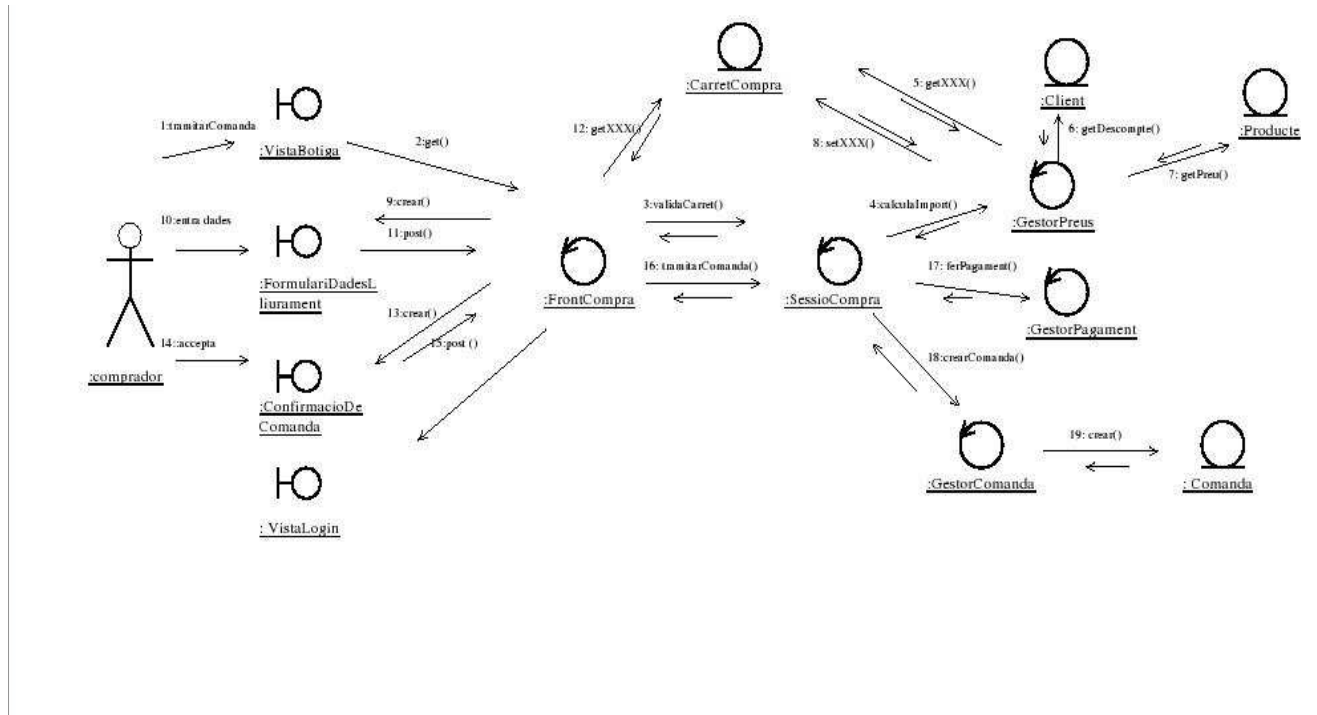
En la fase d'anàlisi es va crear una classe Administrador i una classe Client (que heretaven de la classe Usuari). En aquesta fase només tindrem la classe Usuari amb el mètode EsAdminstrador .

El diagrama queda així:

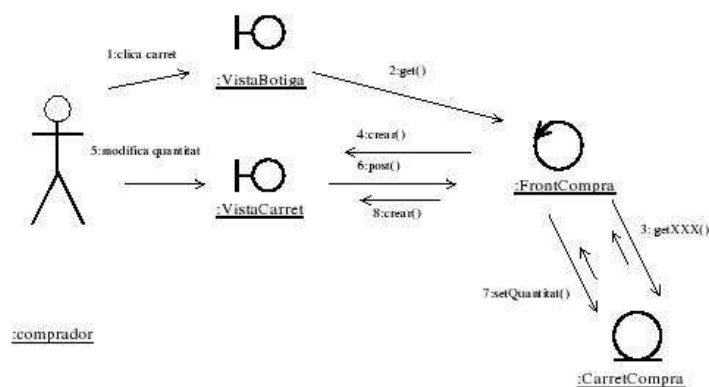


### 3.3. Diagrames de col·laboració

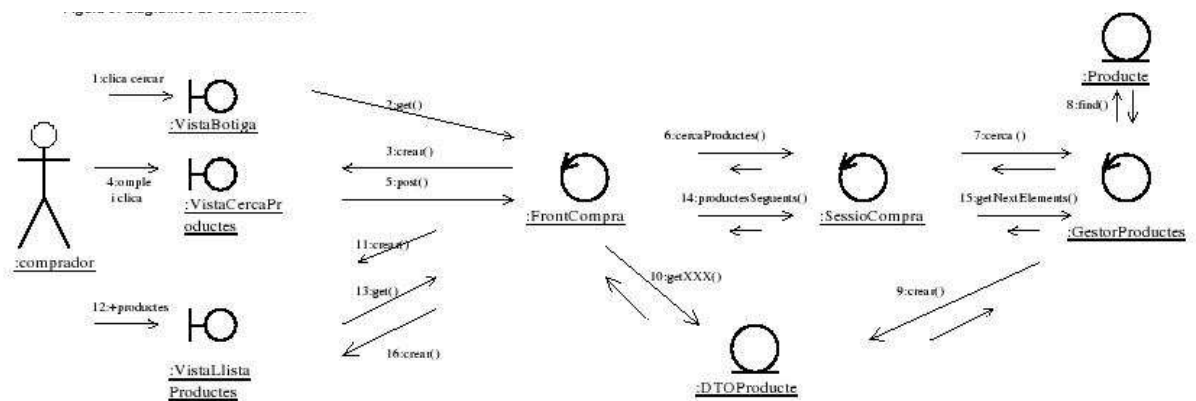
#### 3.3.1. Cas d'ús de cerca de productes per la botiga



#### 3.3.2. Cas d'ús d'editar el carret



### 3.3.3. Cas d'ús de tramitar una comanda

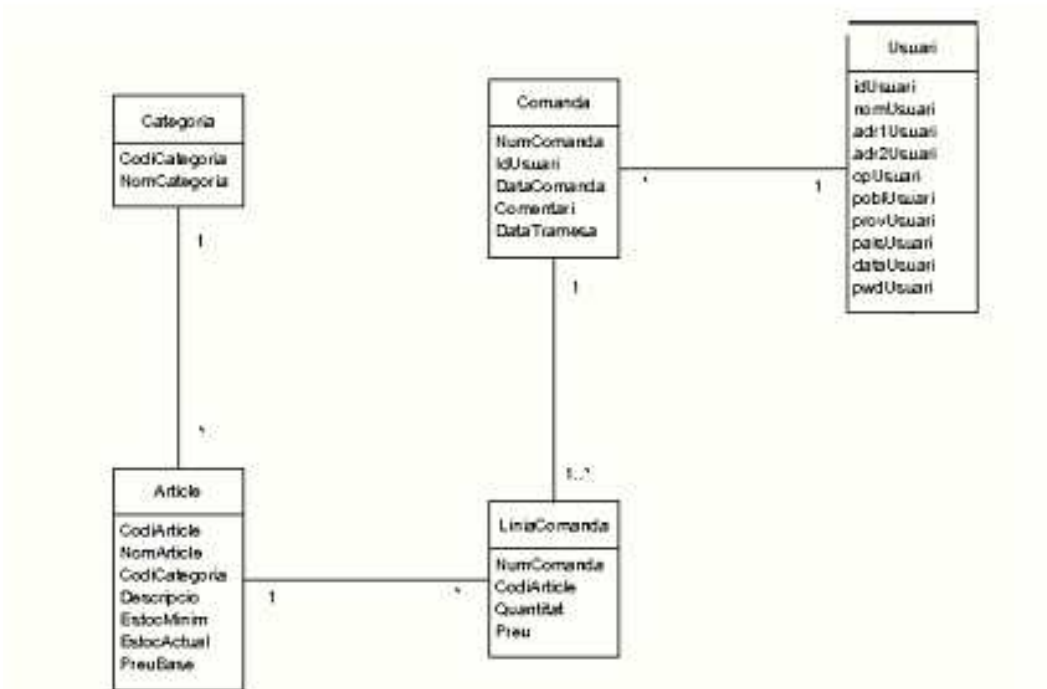


## 3.4. Disseny de la persistència

### 3.4.1 Diagrama de la persistència

Si ens fixem en el diagrama de classes d'entitats observem que només és necessària la persistència per a les següents classes:

- Categoria
- Articles
- Comandes
- Línies de comandes
- Usuaris



Per tal d’implementar la persistència ens decantarem per una tecnologia de mapeig objecte-relacional. Concretament utilitzarem Hibernate.

### 3.4.2. Hibernate

Hibernate es un marc de treball que proporciona mecanismes de mapeig objecte/relacional per definir com s’emmagatzemen, eliminen, actualitzen i recuperen els objectes persistents.

En le disseny d’una aplicació, una part molt important és la manera com accedim a les nostres dades que es troben a la base de dades. Determinar aquesta part es converteix en un punt crític per al futur del desenvolupament.

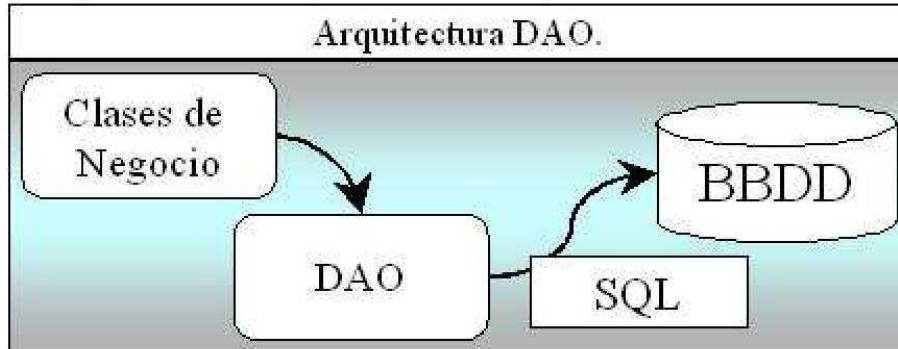
La manera tradicional d’accedir seria a través de JDBC directament connectat la bases de dades mitjançant execucions de sentències SQL



Aquesta primera aproximació podria ser útil per a projectes o arquitectures sense gairebé classes de negoci, atès que el manteniment del codi està altament lligat als canvis en el model de dades relacional

de la BBDD, i un mínim canvi implica la revisió de quasi tot el codi així com la seva compilació i nova instal·lació al client.

Una aproximació més avançada seria la creació de classes d'accés a dades (DAO Data Access Object). D'aquesta manera la nostra capa de negoci interactua amb la capa DAO i aquest s'encarregaria de realitzar les operacions sobre la BBDD

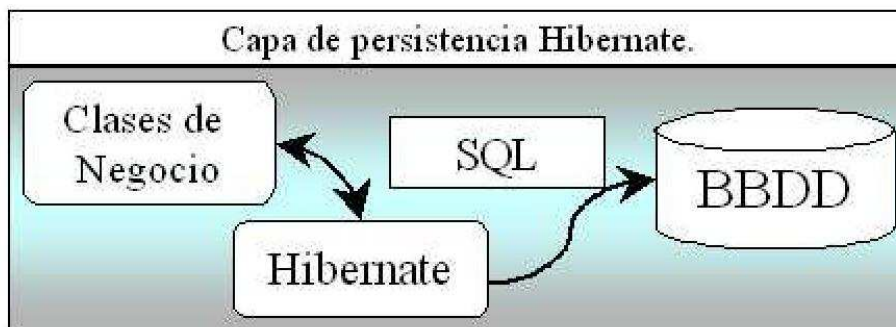


2. Ejemplo de DAO (Data Access Object)

El problema d'aquesta implementació continua sent el manteniment de la mateixa així com la seva portabilitat. L'únic que podem dir es que tenim el codi encapsulat dins de les classes DAO.

En aquest projecte i donat que es tracta d'un projecte purament acadèmic, s'ha optat per Hibernate.

Hibernate resolt en part els inconvenients anteriors. Sembla clar que s'ha de separar el codi de les nostres classes de negoci de la realització de les nostres sentències SQL contra la nostra BBDD. Per tant Hibernate és el pont entre els nostra aplicació i la BBDD.



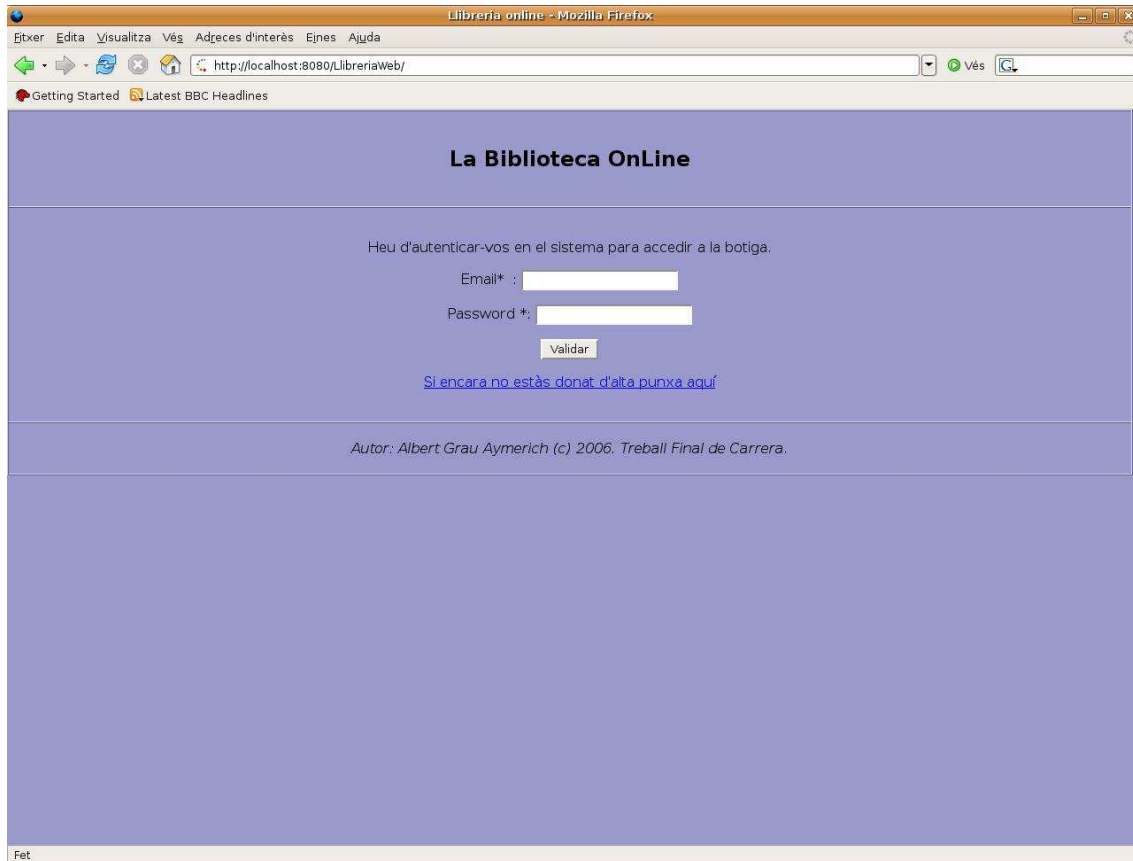
3. Persistencia con Hibernate

Hibernate ens permetrà, una vegada tinguem dissenyada la persistència, mapejar-la a fitxers XML seguint la DTO de mapeig de Hibernate. I des d'aquests generar el codi de les nostres objectes persistents en classes Java independent de l'entorn escollit.

### 3.5. Disseny de les interfícies d'usuari

Es presenten a continuació, a tall d'exemple, algunes pantalles de l'aplicació. Concretament són les pantalles que veurà l'usuari Client.

#### 3.5.1. Pantalla de login





### 3.5.2. Formulari de registre:


The screenshot shows a Mozilla Firefox browser window titled 'Llibreria online - Mozilla Firefox'. The address bar contains the URL 'http://localhost:8080/LlibreriaWeb/jsp/altausuari.jsp;jsessionid=F981C0F34A2135CE186AC668CE1D89E0'. The page content is on a purple background with the title 'La Biblioteca OnLine' at the top. Below the title, a message states 'Els camps marcats amb un \* són obligatoris.' (Fields marked with an asterisk are mandatory). The registration form includes the following fields: 'Nom\*', 'Email\*', 'Adreça 1\*', 'Adreça 2', 'Codi Postal\*', 'Població\*', 'Província\*', 'Pais\*', 'Password \*', and 'Confirmació del Password \*'. Each field is represented by a white text input box. An 'Enviar' button is located below the form. At the bottom of the page, the text 'Autor: Albert Grau Aymerich (c) 2006. Treball Final de Carrera.' is displayed. The browser's status bar at the bottom left shows the word 'Fet'.

### 3.5.3. Botiga

**La Biblioteca OnLine**

**MENÚ**  
[Veure categories](#)  
[Veure llibres](#)  
[Veure carret](#)  
[Sortir](#)

**Cercador**  
 Per Autor  Per Títol  Per ISBN

	<ul style="list-style-type: none"> <li>• <b>Autor</b> : Maverich</li> <li>• <b>Títol</b> : La vella i la bèstia</li> <li>• <b>ISBN</b> : 9799977999</li> <li>• <b>El nostre preu</b> : 13,0 EUR</li> </ul>	<a href="#">Consultar</a> <a href="#">Afegir al carret de la compra</a>
	<ul style="list-style-type: none"> <li>• <b>Autor</b> : Autor dos</li> <li>• <b>Títol</b> : Llibre dos</li> <li>• <b>ISBN</b> : 1112333</li> <li>• <b>El nostre preu</b> : 12,0 EUR</li> </ul>	<a href="#">Consultar</a> <a href="#">Afegir al carret de la compra</a>

Autor: Albert Grau Aymerich (c) 2006. Treball Final de Carrera.

Fet

### 3.5.4. Pantalla de descripció de l'article

Libreria online - Mozilla Firefox

Fitxer Edita Visualitza Vés Adreces d'interès Eines Ajuda

http://localhost:8080/LibreriaWeb/libreDetail.do?sidLibre=1

Getting Started Latest BBC Headlines

## La Biblioteca OnLine

### MENÚ

- [Veure categories](#)
- [Veure llibres](#)
- [Veure carret](#)
- [Sortir](#)

### Cercador

Per Autor  Per Títol  Per ISBN

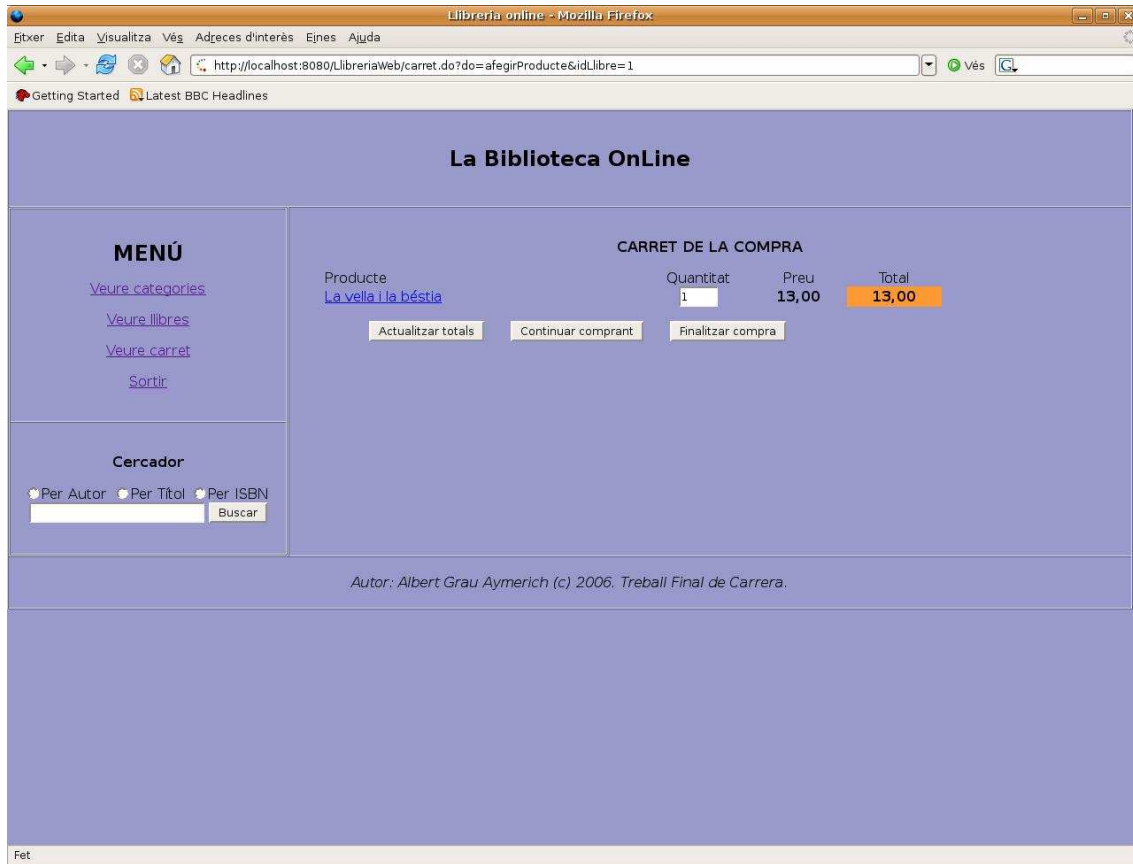
### Corsarios de Levante

- **Autor\*** : Arturo Pérez Reverte
- **ISBN\*** : 9799977999
- **El nostre preu** : 13.0
- **Descripció llarga** : Durante casi dos años serví con el capitán Alariste en las galeras de Nápoles. Por eso hablaré ahora de escaramuzas, corsarios, abordajes, matanzas y saqueos. Así conocerán vuestras mercedes el modo en que el nombre de mi patria era respetado, temido y odiado también en los mares de Levante. Contaré que el diablo no tiene color, ni nación, ni bandera; y cómo, para crear el infierno en el mar o en la tierra, no eran menester más que un español y el filo de una espada. En eso, como en casi todo, mejor nos habría ido haciendo lo que otros, más atentos a la prosperidad que a la reputación, abriéndonos al mundo que habíamos descubierto y ensanchado, en vez de enrocamos en las sotanas de los confesores reales, los privilegios de sangre, la poca afición al trabajo, la cruz y la espada, mientras se nos pudrían la inteligencia, la patria y el alma. Pero nadie nos permitió elegir. Al menos, para pasmo de la Historia, supimos cobrárselo caro al mundo, acuchillándolo hasta que no quedamos uno en pie.

Autor: Albert Grau Aymerich (c) 2006. Treball Final de Carrera.

Fet

### 3.5.5. Carret de la compra



---

## 4. Implementació

---

En aquest capítol s'exposa l'estructura de l'aplicació tal i com s'ha implementat: estructura dels paquets, interrelacions entre classes, etc.

### 4.1 Estructura general

Aquesta aplicació es basa en l'especificació J2EE. Això implica que la interfície d'usuari es basa en servlets i pàgines JSP. Tal i com s'ha comentat en apartats anteriors, s'usarà EJBs per a resoldre la persistència, mentre que la interfície d'usuari estarà basada en pàgines JSP gestionades per la infraestructura Struts ( que no deixa de ser un servlet universal que incorpora el patró de disseny MVC. Per a la implementació del codi java d'aquest projecte s'ha usat Eclipse 3.1 amb el plugin MyEclipse que facilita la creació, compilació, desplegament i depuració del codi .

El servidor d'aplicacions sobre el que s'ha desplegat l'aplicació ha estat JBoss amb Tomcat incorporat com a servidor web. Pel que fa a la persistència s'ha usat Mysql com a sistema de gestor de bases de dades.

Per a facilitar la codificació i per a separar lògicament els diferents tipus de components s'ha optat per una infraestructura en paquets que reflexa la classificació formal dels components.

### 4.2. Característiques de la implementació

Tal com era el requisit en aquest projecte, s'ha aplicat un model MVC que separa clarament la capa de presentació, la capa de negoci i la capa de dades.

### 4.3. Estructura de la implementació

Tots els paquets pengen d'una arrel com.albert. La relació de paquets és i classes és la que segueix:

#### 4.3.1 com.albert.llibreria.dades

Nom del paquet: com.albert.llibreria.dades		
Contingut: Implementacions de la persistència usant Hibernate		
Tipus	Nom	Funció
Classe	FacanaAplica	Controlador i gestor de la persistència .

#### 4.3.2 com.albert.llibreria

Nom del paquet: com.albert.llibreria.dades		
Contingut: Classes d'implementació pels EJB's i		
Tipus	Nom	Funció

Classe	AbstractCategoria	Implementació de l'entitat Categoria. Aquests dues classes contenen la definició del EJB.
Classe	Categoria	
Classe	AbstractUsuari	Implementació de l'entitat Usuari. Aquests dues classes contenen la definició del EJB.
Classe	Usuari	
Classe	AbstractLlibre	Implementació de l'entitat Llibre. Aquests dues classes contenen la definició del EJB
Classe	Llibre	
Classe	AbstractComanda	Implementació de l'entitat Llibre. Aquests dues classes contenen la definició del EJB
Classe	Comanda	
Classe	AbstractLineaComanda	Implementació de l'entitat Llibre. Aquests dues classes contenen la definició del EJB
Classe	LineaComanda	
Classe	AbstractLlibre	Implementació de l'entitat Llibre. Aquests dues classes contenen la definició del EJB
Classe	Llibre	
Document XML	Llibre.hbm	Arxiu xml on es realitza el mapeig de les propietats de la classe Llibre amb la taula Llibre
Document XML	Usuari.hbm	Arxiu xml on es realitza el mapeig de les propietats de la classe Usuari amb la taula Usuari
Document XML	Categoria.hbm	Arxiu xml on es realitza el mapeig de les propietats de la classe Categoria amb la taula Categoria
Document XML	Comanda.hbm	Arxiu xml on es realitza el mapeig de les propietats de la classe Comada amb la taula Comanda
Document XML	LineaComanda.hbm	Arxiu xml on es realitza el mapeig de les propietats de la classe LineaComada amb la taula LineaComanda
Classe	HibernateSessionFactory	Implementació del codi que utilitza Hibernate per a la gestió dels accessos a la BD.

#### 4.3.3. com.albert.llibreria.struts

Nom del paquet: com.albert.llibreria.struts		
Contingut: Implantacions vàries		
Tipus	Nom	Funció
Classe	ActionBase	Implementació dels mètodes que seran comuns a totes les classes que implementen les Actions.
Classe	ContenedorUsuari	Implementació de la classe que contindrà en tot moment les dades del usuari connectat.
Classe	Carret	Implantació de les funcionalitats referents a la gestió del carret de

		la compra.
Classe	DispatchActionBase	Implementació dels mètodes que seran comuns a totes les classes que implementen les DispatchActions.
Classe	CarretProducte	Classe que emmagatzema i gestiona els articles del carret.

#### 4.3.4. com.albert.libreria.struts.form

Nom del paquet: com.albert.libreria.		
Contingut: ActionForms per Struts		
Tipus	Nom	Funció
Classe	AltaUsuariForm	ActionForm per a les dades d'un usuari a crear.
Classe	LlibreEditarForm	ActionForm que gestionarà tant la creació d'un nou llibre com la edició d'un de ja existent.
Classe	CategoriaEditarForm	ActionForm que gestionarà tant la creació d'una nova categoria com la edició d'una de ja existent
Classe	LoginForm	ActionForm que gestionarà la entrada a la botiga d'un usuari
Classe	CercadorForm	ActionForm que gestionarà les cerques de llibres.
Classe	LlibreDetalForm	ActionForm que mapaja la pàgina que mostra la informació detallada d'un llibre
Classe	LlibreLlistarForm	ActionForm que mapeja el formulari del catàleg de llibres
Classe	CategoriaLlistarForm	Action que mapeja el formulari de llista les categories
Classe	ComandaLlistarForm	ActionForm que mapeja el formulari que llista les comandes
Classe	CarretForm	ActionForm que mapeja el formulari del Carret de la Compra

#### 4.3.5. com.albert.libreria.struts.action

Nom del paquet: com.albert.libreria.		
Contingut: Action per Struts		
Tipus	Nom	Funció
Classe	AltaUsuariAction	Alta i edició d'un nou usuari
Classe	LlibreEditarAction	Alta i edició d'articles.
Classe	CategoriaEditarAction	Alta i edició de categories.
Classe	LoginAction	Validació d'un usuari al sistema.
Classe	CercadorAction	Cercador de llibres
Classe	LlibreDetallAction	Consulta de tota la informació disponible per a un llibre en concret.

Classe	LlibreLlistarAction	Navegació per a la llista de Llibres
Classe	CategoriaLlistarAction	Navegació per a la llista de Categories.
Classe	ComandaLlistarAction	Navegació per a la llista de Comandes.
Classe	CarretAction	Gestió del carret de la compra : creació d'una comanda a partir d'un carret de la compra

#### 4.4. Action Forms i Action Beans

Les classes derivades de ActionForm i d'Action formen part de la infraestructura Struts. Per recorda-ho breument, la comunicació entre el controlador i la presentació es fa mitjançant especialitzacions d'ActionForm, mentre que la comunicació entre el model (dades i negoci) i el controlador es fa mitjançant especialitzacions de Action.

En la implementació realitzada, totes les ActionForm contenen:

- Tots aquells atributs necessaris per a construir un objecte a més de les dades necessàries per a gestionar el formulari.
- Llistes d'objectes a presentar (com ara llibres o categories)

S'ha seguit el model d'agafar el nom del formulari que es mapeja i afegir-li la terminació "form".

Dins d'una classe ActionForm trobarem tot un seguit d'atributs privats i els seus corresponents mètodes accessors ("get" i "set"). Per exemple, dins del formulari de login, hi ha un camp que és *email*. Aquest camp té el seu corresponent atribut dins de la classe LoginForm que el mapeja. El controlador de Struts accedirà a aquest camp gràcies als mètodes get i set que hi ha implementats.

Les classes Action criden a les classes ActionForm per a rebre o carregar els valors dins dels seus atributs procedents de les classes que implementen el negoci.

Cada Action implementa una acció determinada dins del funcionament de l'aplicació i cada Action retorna un mapping.findForward que representa un entrada dins de l'arxiu xml struts-config. D'aquesta manera, si una Action es concret arriba al final es comprova si hi ha algun error. Si no hi ha hagut cap, s'actualitzen les dades de l'ActionForm i es respon amb un èxit.



## 4.5. Objectes Valor

En poques paraules, el patró d'objectes valor, consisteix en passar la informació corresponent als atributs dels objectes persistents entre les diferents capes de l'aplicació en forma d'instàncies d'objectes creats especialment a aquest efecte, en lloc d'usar referències remotes i accedir als atributs un a un. Això permet estalviar tràfic de xarxa i separar més clarament la capa de persistència de les capes de lògica de negoci.

Els objectes valor que s'han definit en l'aplicació han estat els següents:

Llibre  
 Categoria  
 Comanda  
 LineaComanda  
 Usuari

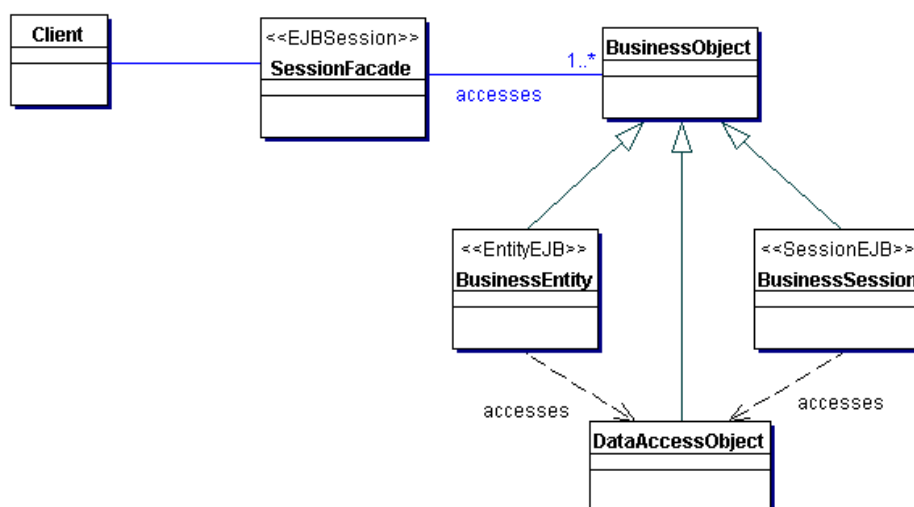
## 4.6. Session Façade

Tots els accessos a dades s'ha encapsulat dintre de la classe FacanaAplica. Aquesta classe es crida des de gairebé totes les classe Action que necessiten recuperar o guardar alguna dada a BBDD.

Podríem dir que, juntament amb les classes que ens proporciona Hibernate, formen la capa de dades.

El seu contingut no es més que una sèrie de funcions que proporcionen a l'aplicació tota la gestió de les dades persistents.

En un altre entorn, amb més volum de classes de negoci, la façana s'hauria d'haver utilitzat com a classe que gestiona i crida altres classes de negoci i accés a dades tal com indica l'esquema.



## **4.7. Internalització**

S'ha intentat que aquesta aplicació fos fàcilment internacionalitzable . Per a tal fi, l'aplicació disposa d'un `ApplicationResource.properties` que inclou gairebé tots els missatges i texts que es apareixen a l'aplicació.

Canviar d'idioma és tant fàcil com a es traduir les missatges en l'idioma que es vulgui sense haver de retocar cap línia de codi.

---

## 5. Conclusions

---

Es el moment de fer balanç sobre el que ha estat el desenvolupament d'aquest projecte. El principal inconvenient ha estat la limitació del temps a 3 mesos i mig. I tenint en compte que (en el meu cas) es va començar el projecte des de zero, aquest inconvenient ha estat determinant. Ha faltat temps i s'han deixat certs punts inacabats.

Totes les previsions de temps s'han quedat curtes i ha estat impossible seguir el calendari marcat. El principal obstacle que s'ha trobat ha estat el configurar tot l'entorn de programació. Per a un programador sense experiència amb J2EE, resulta complicat la configuració de totes les tecnologies que hi ha al darrera d'un projecte Java i més concretament d'un projecte J2EE (struts, hibernate, tomcat ..).

Si es té en compte que dels tres mesos i mig, la meitat del temps s'ha gastat amb l'anàlisi i el disseny, ens queda un total d'un mes i mig per a digerir tot el que implica el J2EE.

Es podria dir, com a principal conclusió, que s'ha fet el s'ha pogut. Tanmateix, l'experiència general ha estat bona i m'ha quedat amb ganes d'aprofundir més. Això sí, sense tantes preses.

### 5.1 Objectius Assolits

En primer lloc, i tot i que no és un objectiu qualificable, es pot dir que s'han assolit els objectius respecte a l'aprenentatge i familiarització amb la tecnologia J2EE. Tot i que la corba d'aprenentatge comença tenint una pendent considerable, arriba un moment a partir del qual la productivitat en quan al desenvolupament es dispara.

S'han posat en pràctica els coneixements adquirits al llarg del TFC sobre les diverses metodologies, tecnologies i eines implicades en el desenvolupament d'aplicacions web amb J2EE. S'ha posat en pràctica la integració de les com JSP, servlets, i accés a dades mitjançant Hibernate. Com a eines s'ha usat Poseidon, Eclipse, Tomcat, JBoss i Mysql.

En que respecta a les funcionalitats desenvolupades, podem considerar com a aconseguits els següents objectius,

Registre d'usuaris.

Connexió i desconnexió del sistema.

Gestió de les categories per part de l'administrador.

Cerques

Navegació per la botiga per part del client.

Gestió del carret de compra

Finalització de la compra i gravació de la compra.

També es poden considerar objectius aconseguits la clara separació de les tres capes que intervenen

### 5.2 Objectius no assolits

La relació dels objectius inicials que no s'ha pogut resoldre per falta de temps han estat:

- Gestió de les comandes de l'usuari. La gestió es limita a consultar i res més. Caldria un bona gestió on es pogués consultat l'estat , per exemple.
- No s'ha pogut implementar un gestió dels usuaris per part de l'administrador
- Tampoc s'ha resolt la paginació de les consultes.
- Una bona gestió dels errors. No s'ha dut a terme un aspecte tant important com la gestió dels errors.

---

## 6. Bibliografia

---

Deepak Alur, John Crupi, Dan Malks. Core J2EE™ Patterns: Best Practices and Design Strategies. Prentice Hall PTR. 2001. ISBN : 0-13-064884-1

Gail Anderson, Paul Anderson. Enterprise JavaBeans™ Component Architecture: Designing and Coding Enterprise Applications. Prentice Hall PTR. 2002. ISBN : 0-13-035571-2.

Marty Hall. Core Servlets and JavaServer Pages™. Prentice Hall PTR. 2000. ISBN : 0-13-089340-4.

Marty Hall. More Servlets and JavaServer Pages™. Prentice Hall PTR. 2001. ISBN : 0-13-067614-4.

Ivar Jacobson, Grady Booch & James Rumbaugh. El proceso unificado de desarrollo desoftware. Addison-Wesley, 1998. ISBN: 84-7829-0362.

Alan Shalloway, James R. Trott. Design Patterns Explained. A New Perspective on Object Oriented Dessign. Addison Wesley. 2002. ISBN: 0-201-71594-5.

ML y Patrones (2ª Edición) – Craig Larman – Pearson/Prentice Hall, 2003 – ISBN: 0-13- 092569.1

JDBC API Tutorial and Reference, Second Edition

Jakarta Struts Live – Rick Hightower – SourceBeat LLC, 2004 – ISBN: 097-48843-08

JBoss administration and development Third Edition – Scott Stark & The JBoss Group – JBoss Group LLC, 2003 – (eBook).

<http://www.theserverside.com>.

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

<http://www.ibm.com/developerworks/websphere>

Material UOC de l'assignatura Enginyeria del Programari.

<http://www.google.com>.