

Universitat Oberta de Catalunya.
Enginyeria Tècnica en Informàtica de Sistemes.

Treball Fi de Carrera:

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

Nom de l'Estudiant

Jaume Casadesús Brugués (jcasadesusb@uoc.edu)

Nom del Consultor

Xavier Ferró Garcia

Data de Lliurament

18 de juny de 2004

Resum.

La omnipresència d'Internet ha fet créixer la tendència cap a aplicacions distribuïdes que usen una interfície web entre l'usuari i l'aplicació. Un cas típic en són les aplicacions de comerç electrònic.

Es tracta d'aplicacions amb una certa complexitat al darrera donat el nombre de tecnologies que inclouen, ja que han de cobrir des de la interacció amb l'usuari fins a la persistència de les dades, passant per la lògica del negoci i per tots els serveis de middleware que ho fan possible.

L'especificació J2EE proposa una arquitectura per a aquest tipus d'aplicacions que té una bona acceptació en la indústria gràcies a la seva portabilitat, robustesa i relativa rapidesa de desenvolupament.

En aquesta memòria s'exposa el procés de desenvolupament d'una aplicació corresponent a una botiga virtual, que s'implementa mitjançant J2EE, amb una clara separació de les capes de presentació, negoci i dades.

La capa de presentació s'ha implementat amb un servlet per a la part pública i un altre per a la part privada, que usen diversos JSPs per a crear les vistes dels diferents casos d'ús. La capa de negoci s'oculta darrera dues Session Beans, una per la part pública i l'altra per la part privada, que deleguen algunes tasques de processament a altres classes del negoci. L'accés a les dades s'encapsula dins de 3 Entity Beans amb persistència controlada pel propi bean.

En aquest projecte, la tecnologia J2EE ha permès desenvolupar una aplicació de certa complexitat sense haver de fer explícitament cap crida al middleware. Al mateix temps, el seguiment de patrons de disseny ha permès aconseguir un reduït acoblament entre la capa de presentació i la de negoci alhora que segurament ha evitat part dels errors i les complicacions innecessàries que podien haver sorgit per falta d'experiència.

Index de seccions.

1. Introducció	5
2. Planificació	6
3. Anàlisi	8
3.1. Definició del concepte operacional	8
3.2. Definició dels requeriments	8
3.3. Model d'interfície gràfica d'usuari	11
3.4. Diagrama de classes de l'anàlisi	11
3.5. Descripció dels casos d'ús	12
4. Disseny	23
4.1. Descripció general de l'arquitectura	23
4.2. Llista de classes	24
4.3. Fitxes de classes	24
4.4. Disseny de la Base de Dades	37
4.5. Diagrama de classes	38
4.6. Diagrames de col·laboració	41
4.7. Disseny de les interfícies d'usuari	44
5. Implementació i proves	49
5.1. Procés d'implementació	49
5.2. Resultat de la implementació	50
5.3. Característiques de la implementació	50
5.4. Proves	52
5.5. Algunes vistes de l'aplicació obtinguda	53
6. Conclusions	56
7. Glossari	58
8. Bibliografia	58
9. Annexos	59
9.1. Instruccions d'instal·lació	59
9.2. Material que s'adjunta	60

Índex de figures.

Figura 1: diagrames de Gantt.....	7
Figura 2: model de GUI.....	11
Figura 3: diagrama de classes de l'anàlisi.....	11
Figures 4: diagrames de casos d'ús.....	12 - 23
Figura 5: diagrama ER.....	37
Figura 6: classes comuns entre presentació i negoci.....	38
Figura 7: classes del model de negoci.....	39
Figura 8: diagrama de classes de la presentació.....	40
Figura 9: diagrames de col·laboració.....	41 - 44
Figura 10: disseny de les interfícies d'usuari.....	44 - 49
Figura 11: diagrama de components.....	50
Figura 12: separació en 3 capes.....	51
Figura 13: algunes vistes de l'aplicació obtinguda.....	53 - 56

Introducció

Enterprise JavaBeans (EJB) és l'especificació d'una arquitectura per a components Java desplegable en el servidor d'aplicacions. Aquesta especificació, que consisteix en un conjunt d'interfícies Java, fixa com han d'interactuar els components i servidors d'aplicacions per a permetre que qualsevol component que s'ajusti a l'especificació pugui executar-se en qualsevol servidor que també segueixi l'especificació.

Un dels punts forts de EJB és que els serveis de middleware els proporciona el Contenedor d'EJBs, de manera que en el codi de l'aplicació no es fa cap referència al middleware. Això comporta alguns avantatges com són la rapidesa de desenvolupament, la millora de la robustesa i mantenibilitat del codi, i també facilita la gestió de la propietat intel·lectual.

D'aquesta manera la feina dels desenvolupadors de EJB se centra a resoldre el què realment són els problemes del negoci: control del flux de treball del negoci, operacions específiques del domini de l'aplicació i manipulació de les dades.

A més, al centrar la lògica de l'aplicació en els components EJB, aquesta arquitectura és molt adient per a usar-la amb clients web. En aquest sentit, la combinació de components EJB amb servlets i Java Server Pages (JSP) resulta molt adequada en aplicacions com ara comerç electrònic.

No obstant totes aquestes virtuts, estem parlant d'aplicacions que poden ser força complexes donat el nombre de tecnologies que inclouen. Com qualsevol tecnologia nova, un desenvolupador que s'introdueix per primera vegada en aplicacions distribuïdes corre el risc de cometre errors costosos o d'invertir moltes hores per acabar reinventant la roda. En aquest context, l'ús de patrons de disseny pot ser una ajuda molt valuosa.

Els patrons de disseny pretenen fomentar l'ús de solucions que prèviament s'han 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 a aquest problema.

Els patrons de disseny s'anticipen i preveuen una sèrie de problemes que poden restar ocults fins a fases més avançades del projecte. A més, també són útils perquè aporten major claredat quan es discuteix un disseny o una implementació, ja que amb un sol terme resumeixen una estructura i funcionalitat que poden ser força difícils d'expressar d'altra manera.

En el cas concret de l'arquitectura EJB, els patrons de disseny poden ser particularment útils per a resoldre la relació entre capes del model MVC.

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. L'ús de la tecnologia J2EE li permet que no s'hagi d'ocupar de les funcionalitats del middleware, de manera que guanya temps, portabilitat i fiabilitat. El seguiment de patrons de disseny afavoreix l'ús de solucions eficaces a la primera i d'aquesta manera li permet anticipar i evitar part dels errors i complicacions innecessàries que es podria trobar per falta d'experiència.

Objectius del TFC:

L'objectiu d'aquest TFC és realitzar l'anàlisi, disseny i implementació d'una aplicació de comerç electrònic fent servir l'arquitectura J2EE, en què cal separar clarament la presentació, la lògica del negoci i l'accés a les dades. L'exemple escollit és una botiga virtual, concretament una llibreria.

Metodologia:

El desenvolupament del projecte segueix de manera simplificada el procés unificat del desenvolupament de programari.

En la fase d'anàlisi el procés ve dirigit pels casos d'ús. En la fase de disseny es posa atenció a patrons de disseny que poden ser d'ajut en el cas d'aplicacions J2EE. En la fase d'implementació es posen en pràctica la tecnologia J2EE així com l'API Servlet i els JSPs.

Producte final.

Al final del projecte s'obté una aplicació funcional, consistent en diversos components de desplegament, que correspon a la part pública i la part backend d'una llibreria que ven per Internet. Això es distribueix en 2 aplicacions client, una per a la part pública i l'altra pel backend, 2 Session Beans i 3 Entity Beans.

Al llarg de la memòria es descriuen les diverses fases d'execució del projecte.

2. Planificació del projecte.

Objectiu general:

Anàlisi, disseny i implementació d'una botiga virtual usant una arquitectura de 3 capes basada en J2EE.

Es pretén separar clarament la presentació, la lògica de negoci i l'accés a dades. Concretament, la presentació es preveu implementar amb combinació de servlets i JSPs, la lògica de negoci amb una session bean i la capa de dades amb entity beans.

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.

Recursos:

Els recursos humans es limiten a una persona, amb una dedicació d'unes 150h, sense comptar formació, que haurà de jugar els diversos rols implicats en el desenvolupament d'aplicacions distribuïdes.

Requeriments de maquinari:

Punt de treball estàndar de la UOC.

Requeriments de programari:

- JDK 1.4
- JBoss com a implementació de referència J2EE 1.3.
- MySQL com a base de dades.
- Eclipse com a eina de desenvolupament en Java.

Riscos del projecte:

Es consideren els riscos derivats de la inexperiència en implementació d'aplicacions distribuïdes. Concretament la inexperiència pot conduir a dissenys incorrectes, ineficaços o difícils d'implementar, així com a enlentir la implementació. Per a mitigar aquests riscos serà necessari des del principi del projecte dedicar un temps addicional a la formació, mitjançant tutorials i llibres online. També s'espera compensar la inexperiència amb el seguiment de solucions proposades per altres, en forma de patrons.

Calendari:

El calendari anirà marcat per les dates dels lliurables:

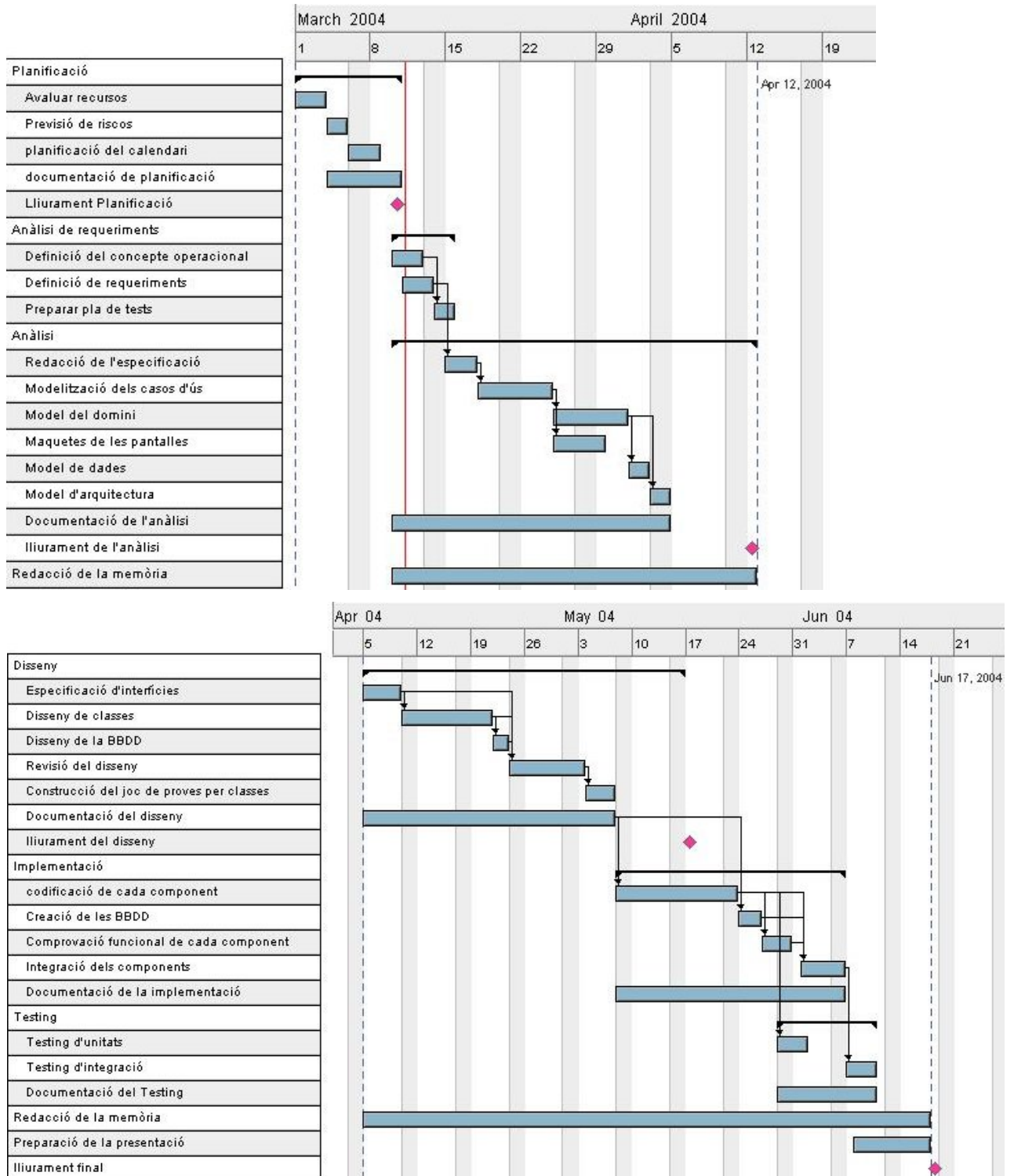
- 8 de març: pla del projecte.
- 13 d'abril: document d'anàlisi.
- 17 de maig: document del disseny.
- 18 de juny: lliurament final, consistent en una memòria (màxim 60 planes) i una presentació (màxim 20 diapositives).

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

Fases del projecte:

El desenvolupament del projecte seguirà aproximadament el mètode RUP, degudament simplificat considerant el context d'aquest projecte com a TFC. Les fases del projecte seràn Planificació, Anàlisi, Disseny, Implementació, i Proves. Els temps estimats per a cada tasca i la seva temporalització és la que es representa als diagrames de Gantt següents.

Figura 1: diagrames de Gantt.



3. Anàlisi.

3.1. Definició del concepte operacional

3.1.1. Objectius:

El programari **e-botiga** serà una aplicació amb funcionalitat de botiga electrònica, concretament una llibreria, que té com a principal objectiu servir de demostració de la tecnologia J2EE.

La funcionalitat de botiga electrònica s'oferirà a qualsevol usuari degudament identificat que es connecti a través d'Internet usant qualsevol navegador. A més, s'oferirà els serveis de backend als empleats acreditats.

3.1.2. Composició del programari:

El programari consistirà d'una part servidora implementada amb tecnologia J2EE, ubicada en una seu central, un client "lleuger" corresponent als compradors, que accediran a la botiga des d'Internet usant qualsevol navegador convencional, i un altre client (sense especificar encara si serà un servei web o una aplicació Java) a través del qual els empleats acreditats realitzaran les tasques de manteniment de la rebotiga.

3.1.3. Resum dels requeriments funcionals.

Es tracta del desenvolupament d'una aplicació de comerç electrònic consistent en una botiga virtual accessible per tots els usuaris a través d'Internet usant qualsevol navegador i una part privada, accessible només als operaris del *backend*, que no necessàriament ha de ser un servei web. El producte que es ven en aquesta botiga són llibres.

En la botiga virtual els usuaris han de poder veure els productes classificats per categories i realitzar comandes, usant un carret de compra.

En la part privada es gestionarà tota la informació relacionada amb el backend. Això inclou la realització d'altres, baixes i modificacions de productes, usuaris i comandes.

En tractar-se d'una aplicació de demostració de J2EE les prestacions en ús real de l'aplicació no són importants. Per tant no seran crítics alguns paràmetres que sí que ho són en una aplicació comercial real com són la usabilitat de les interfícies, el temps de resposta o la seguretat.

3.1.4. Requeriments tecnològics.

Es demana específicament que el disseny i implementació de la botiga virtual ha de ser en arquitectura de 3 capes, usant J2EE, a on se separi clarament la presentació, la lògica de negoci i l'accés a les dades. Es maximitzarà l'ús d'estàndards i la persistència de les dades serà en BD relacional. L'escalabilitat, fiabilitat i portabilitat seran les inherents a la tecnologia J2EE.

3.2. Definició dels requeriments

3.2.1. Part pública.

3.2.1.1. Accés a la botiga.

Al connectar al lloc web, s'oferirà una pantalla d'entrada amb el logo de l'empresa i un quadre de diàleg en què es demanarà al client que s'identifiqui amb nom i password per a entrar a la botiga.

També se li mostrarà a on ha de clicar per a registrar-se com a nou client si encara no ho és.

Si el client ja està donat d'alta introduirà el nom d'usuari i password i a continuació clicarà al lloc indicat per a enviar les dades d'identificació al servidor. En cas de resposta afirmativa per part del servidor accedirà a la botiga; en cas contrari seguirà en la mateixa pàgina per si ho vol tornar a provar.

3.2.1.2. Registre de nou client.

Jaume Casadesús

Es mostrarà una pantalla amb els següents camps que el client haurà d'introduir obligatòriament: nom d'usuari, password, adreça e-mail i adreça de correu a on es faran els lliuraments. L'adreça de correu contindrà els camps: adreça postal, codi postal, municipi i país.

Opcionalment podrà entrar 2 números de telèfon de contacte.

Un cop el client hagi omplert aquests camps haurà de clicar al lloc indicat per a enviar aquestes dades al servidor i obtenir-ne resposta. Si s'han omplert tots els camps obligatoris amb valors vàlids se li mostrarà al client un missatge de confirmació i se li indicarà a on ha de clicar per accedir a la botiga.

Si algun dels camps obligatoris s'ha omplert amb un valor que no és vàlid, es mostrarà la mateixa pàgina però amb un missatge d'error indicant que hi ha entrades incorrectes i s'assenyalarà aquests camps precedint-los per un asterisc de color vermell.

3.2.1.3. Navegació per la botiga.

Durant tot el procés de compra, en la part superior de la pantalla es mostrarà al client un menú a través del qual podrà accedir a:

- Cercador
- Productes d'interès.
- Carret de la compra.
- Tramitar comanda.
- Sortir.

La resta de la pantalla anirà variant segons la navegació del client. Inicialment, quan el client entri a la botiga, es trobarà amb un llistat de recomanacions seguit de l'apartat de novetats. Les recomanacions seran personalitzades per a cada client, basades en la llista d'interès i en compres anteriors. Després de cada operació de cerca de productes aquesta pàgina mostrarà els resultats de la cerca. De manera semblant, quan el client demani veure la llista de productes d'interès es mostrarà la llista de productes que el client ha inclòs en la llista d'interès en aquesta sessió o en alguna d'anterior.

En tots aquests llistats, de cada producte es mostrarà el nom de producte, una breu descripció de poques línies de text, el preu, la disponibilitat, així com altres possibles atributs del producte que encara estan per concretar. Al final d'aquestes línies de text es mostraran tres botons, un per a obtenir informació addicional sobre el producte, un altre per afegir-lo a la llista de productes d'interès i un per a afegir-lo al carret.

Si es clica sobre el botó d'informació addicional s'obrirà una pàgina en què només apareixerà el producte en qüestió i els camps de text de la descripció del producte sense limitació de llargada, seguit de tota la informació que es disposi sobre aquest producte, que pot consistir en un nombre indeterminat de notes de text. Al final de totes les notes de text apareixeran els botons per afegir el producte a la llista d'interès, afegir-lo al carret de la compra o retornar a pàgina anterior.

Si el client clica sobre el botó d'afegir a la llista d'interès s'afegirà el producte a la llista d'interès i a partir d'aquest moment el producte es mostrarà en la llista ressaltat amb un marc de línia puntejada. Si clica sobre el botó d'afegir a carret de compra s'afegirà el producte al carro de la compra i a partir d'aquest moment el producte es mostrarà ressaltat amb un marc de línia contínua i gruixuda.

3.2.1.4. Cercador:

El client accedirà al cercador des de la barra de menú. Se li obrirà una pàgina de formulari en què se li demanarà que ompli un o més dels camps:

- Matèria
- Autor
- Títol
- Editorial
- ISBN

El camp matèria serà una llista seleccionable de matèries predefinides. Els altres camps seran camps d'entrada de text. Una vegada omplerts els camps adients el client clicarà un botó per a que s'efectuï la cerca.

Com a resultat de la cerca es mostrarà el nombre de productes trobats i seguidament els 10 primers, en el format descrit anteriorment, acabant amb unes fletxes per a accedir als 10 següents.

3.2.1.5. Carret de la compra.

S'hi accedirà des del menú de la pàgina principal. El carret de la compra es mostrarà com una taula amb tantes files com productes diferents s'hagin seleccionat i amb les columnes

Quantitat
Codi del producte
Nom del producte
Preu unitari
Import (preu x quantitat)

En aquesta taula el client podrà modificar les cel·les corresponents a la quantitat de cada producte.

La taula acabarà amb l'import total corresponent a la suma de tots els imports.

A continuació de la taula es mostren els botons "buidar carret", "actualitzar imports" i "efectuar compra". Si es clicka a "actualitzar imports" s'actualitzarà la columna import si hi ha hagut un canvi en la quantitat d'algun dels productes.

Quan es clicka a "efectuar compra" s'actualitzen els imports, es mostra en pantalla les dades d'enviament i es demana a l'usuari que confirmi que hi està d'acord.

En cas que hi estigui d'acord es connectarà a l'aplicació de pagament, que és externa al programari *e-botiga*.

3.2.1.6. Llista de productes d'interès.

En tots els llistats de productes, ja siguin els productes recomanats i novetats que es troba a l'entrar a la botiga, així com els corresponents a resultats de cerques, el client pot seleccionar productes per a afegir-los a una llista de productes d'interès. Aquesta llista és personal per a aquest client i es conservarà entre sessions de compra.

El client podrà accedir a aquesta llista des de la barra de menú de la pàgina principal. Això li mostrarà la llista de productes seleccionats, amb el mateix format que els resultats de les cerques i amb els mateixos botons excepte el botó per afegir a llista d'interès que ara serà per a eliminar de la llista d'interès.

3.2.1.7. Tramitació de la comanda.

Un cop el client hagi seleccionat els productes a comprar i els hagi inclòs al carret de la compra seleccionarà la opció "tramitar comanda" per a procedir a efectuar la compra. Podrà entrar en aquesta opció tant des de la barra de menú de la pàgina principal com clickant al lloc indicat en el carret de la compra.

Per a tramitar la comanda se li mostrarà al client el carret de la compra, l'adreça i modalitat d'enviament, i se li demanarà que confirmi aquestes dades o que les modifiqui si cal.

Una vegada el client confirma les dades es connecta a una aplicació de pagament segur i una vegada aquesta finalitza amb èxit ja s'haurà completat el procés de compra. Finalment s'informa al client del resultat de l'èxit o fracàs de l'operació.

3.2.2. Part privada:

2.2.1. Gestió de productes.

Els empleats acreditats entraran nous productes i editaran els atributs dels productes de la base de dades.

Per a donar d'alta un nou llibre l'empleat omplirà un formulari amb els camps:

Matèria
Autor
Títol
Editorial
ISBN
Descripció breu

Jaume Casadesús

Per a editar un producte ja existent a la base de dades primer es passarà per la interfície de cerca, que serà semblant a la de cerca per als clients.

A més dels atributs esmentats es podrà guardar informació addicional sobre un producte en forma de notes de text, compostes de títol i contingut. En aquest sentit es podrà guardar qualsevol nombre de notes referents a un mateix producte.

3.2.2.2. Gestió dels clients.

Els empleats acreditats podran modificar el % de descompte que s'aplicarà a un client determinat i podran afegir o eliminar productes a la llista de productes recomanats a un client determinat, que són els que li apareixeran a la pàgina principal quan es connecti.

Per a afegir una recomanació usaran una interfície de cerca de productes i una altra de clients i associaran un o més productes a un o més clients.

Per a eliminar recomanacions podrà eliminar totes les recomanacions a un determinat client, totes les recomanacions d'un producte concret o bé un producte concret a un client determinat.

3.2.2.3. Gestió del magatzem.

Els empleats del magatzem podran modificar el nombre de productes en existència.

Per a aquesta funció primer passaran per una interfície de cerca de productes.

Com a resultat de cada procés de compra es genera un lliurament que haurà d'expedir l'empleat del magatzem. En aquest sentit, l'empleat del magatzem haurà de consultar quants lliuraments té pendents, processar-los i seguidament marcar-los com a enviats, indicant la data d'expedició.

3.3. Model d'interfície gràfica d'usuari.

La interfície gràfica d'usuari consistirà en una pàgina web amb una part comú a totes les vistes en la que es mostrarà una barra de menú, i una part específica de cada cas d'ús, que pot ser una vista del carret de la compra, o un formulari de cerca o una llista de resultats, o un formulari d'edició, etc.

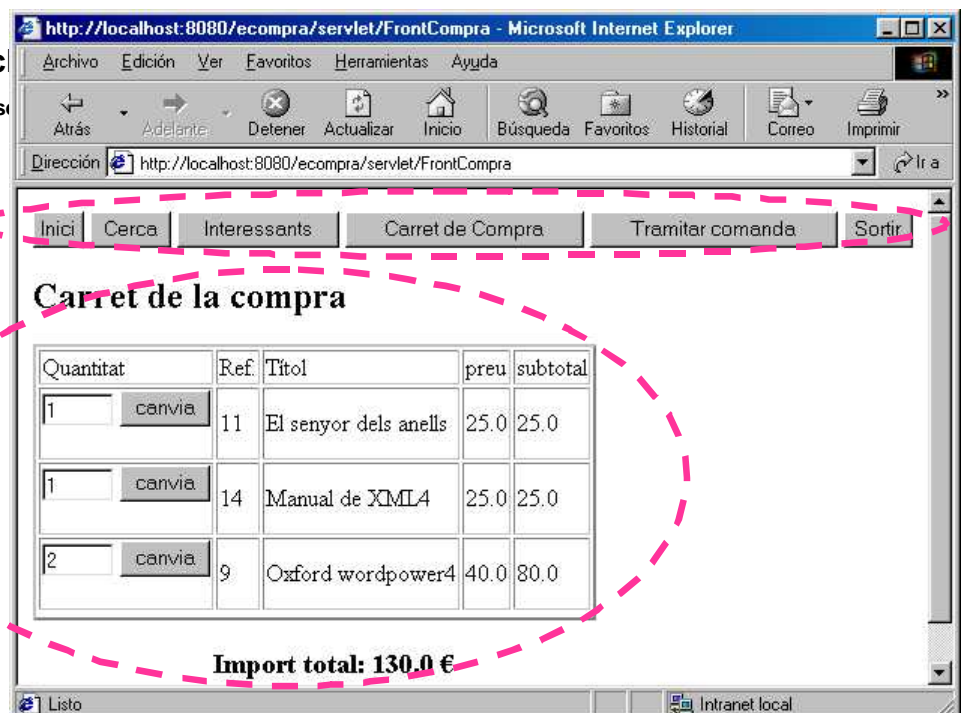
Figura 2: model de GUI.

3.4. Diagrama de classes

Figura 3: diagrama de classes

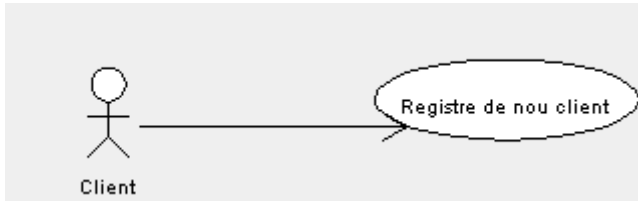
Menú en totes les vistes

Contingut dependent del cas d'ús



entrar al cas d'ús "Registre de nou client". En cas d'identificació positiva el client entra a la pàgina principal de la botiga. En cas contrari segueix en la mateixa pàgina fins que o bé s'identifica de manera positiva o bé decideix sortir de la pàgina.

3.5.1.2. Registre de nou client



Resum de la funcionalitat: el futur client introdueix les seves dades en un formulari que envia al servidor per a donar-se d'alta com a client.

Paper dins del rol del client: cal que el client passi una vegada per aquest cas d'ús abans d'accedir per primera vegada a la botiga. Normalment un client només passarà una vegada per aquest cas d'ús.

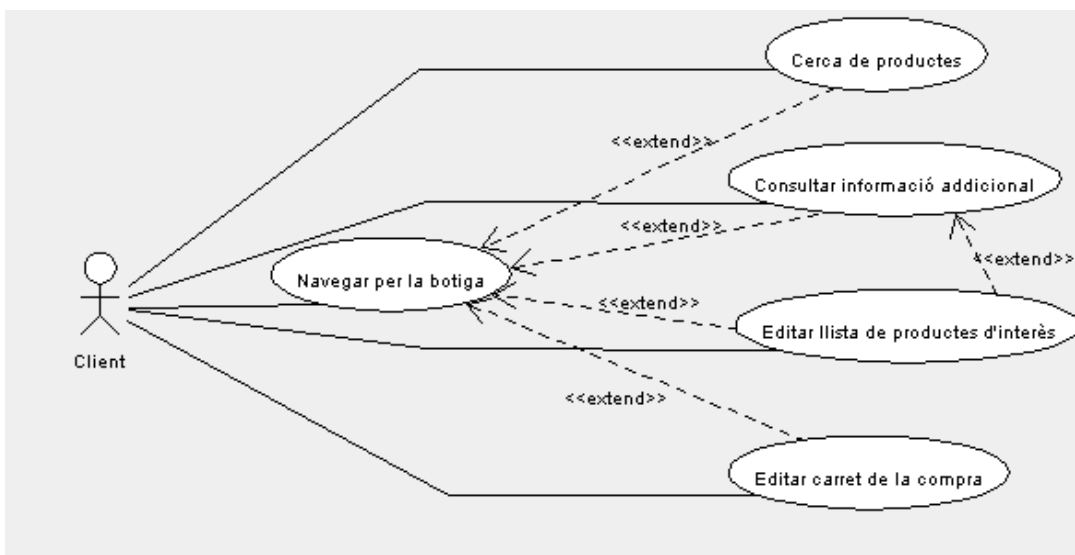
Casos d'ús relacionats: Identificació del client.

Precondicions: cal que el futur client es connecti des d'Internet usant un navegador convencional. El client encara no està donat d'alta a la base de dades.

Postcondicions: el client s'haurà donat d'alta a la base de dades i a partir d'ara es podrà identificar amb el nom de client i password que ha entrat.

Descripció: El client omplirà els camps nom, password, adreça e-mail i l'adreça de correu a on es faran els lliuraments, amb els camps nom, adreça, codi postal, població i país. Opcionalment podrà entrar 2 números de telèfon de contacte. Seguidament haurà de clicar al lloc indicat per a enviar aquestes dades al servidor. Si tots els camps s'han omplert de manera correcta, com a resposta afirmativa per part del servidor es mostrarà un missatge indicant que ja pot entrar a la botiga clickant al lloc indicat. Si hi ha algun camp incorrecte es mostrarà un missatge indicant que existeixen entrades no vàlides i es marcarà amb un asterisc vermell els camps que contenen una entrada no acceptable.

3.5.1.3. Navegar per la botiga



Resum de la funcionalitat: El client podrà moure's pels catàlegs de productes de la botiga, fent cerques, buscant més informació sobre els productes seleccionats i afegir-los al carret de la compra o a la llista de productes d'interès.

Paper dins del rol del client: És un cas d'ús fonamental per a la funcionalitat de la botiga virtual. Des d'aquí accedeix a tots els altres casos d'ús que pot fer com a client.

Casos d'ús relacionats: Cerca de productes, Editar llista de productes d'interès, Editar carret de la compra, Tramitar comanda, Desconnectar.

Precondicions: El client ha estat degudament identificat en el cas d'ús "Identificació del client", i es connecta des d'Internet usant un navegador convencional.

Postcondicions: Els productes seleccionats pel client seran al carret de la compra fins al final de la sessió a no ser que el client els elimini expressament. Els productes afegits a la llista de productes d'interès s'hi guardaran per a properes sessions.

Descripció: El client navegarà per les llistes de productes que li ofereixi directament la botiga com a recomanacions personalitzades i novetats, o pels resultat de cerques de productes que farà el mateix client en el cas d'ús "Cerca de productes". El client es mourà per la llista i podrà demanar més informació sobre un producte i afegir-lo a la llista de productes d'interès o al carret de la compra.

Per cadascun dels productes que se li mostren pot clicar les següents opcions:

- Informació addicional sobre el producte
- Afegir el producte a la llista d'interès
- Afegir el producte al carret de la compra

A la part superior de la pàgina tindrà una barra de menú amb les opcions

"Cercador" que el portarà al cas d'ús Cerca de productes,

"Productes d'interès" que el portarà al cas d'ús Editar llista dels productes d'interès.

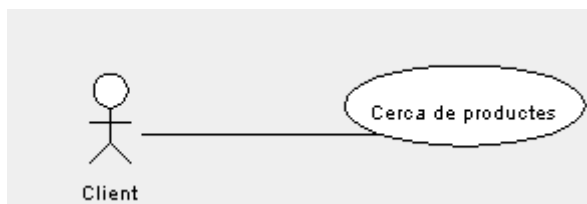
"Carret de la compra" que el portarà al cas d'ús d'Editar carret de la compra,

"Tramitar comanda" que el portarà al cas d'ús Tramitar comanda",

"Desconnectar" que el portarà al cas d'ús Desconnectar.

Observacions: Després de tots els casos d'ús excepte Desconnectar, es tornarà a Navegar per la botiga.

3.5.1.4. Cerca de productes



Resum de la funcionalitat: El client busca productes que compleixin alguna condició, com pot ser la matèria o la presència d'una paraula en algun camp de descripció del producte.

Paper dins del rol del client: És el principal camí per a que el client arribi fins al producte a comprar.

Casos d'ús relacionats: Navegar per la botiga.

Precondicions: El client es troba en el cas d'ús Navegar per la botiga.

Postcondicions: El client es troba al cas d'ús Navegar per la botiga, a on se li mostra una llista de productes que compleixen les condicions de cerca.

Descripció:

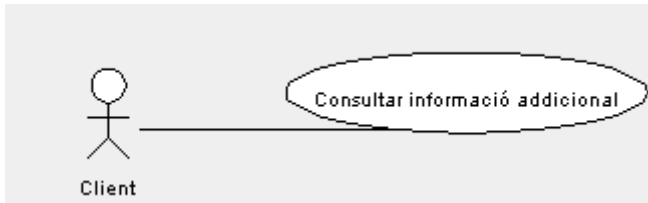
El client omple algun dels camps que defineixen la cerca, que pot ser:

- Matèria, a escollir en una llista seleccionable.
- Editorial, a escollir en una llista seleccionable.
- ISBN, entrada com a text.
- Text a localitzar en el camp Autor
- Text a localitzar en el camp títol.
- Text a localitzar en qualsevol camp de descripció del producte.

Jaume Casadesús

Com a resultat de la cerca s'actualitzarà el contingut de la pàgina principal, mostrant al principi de la pàgina el nombre de productes trobats, a continuació se li mostren els 10 primers amb el format descrit en el cas d'ús Navegar per la botiga i al final unes fletxes a on clicar per a que es mostrin els 10 següents. El client podrà interactuar amb aquesta llista a través del cas d'ús Navegar per la botiga.

3.5.1.5. Consultar informació addicional



Resum de la funcionalitat: El client demana informació addicional sobre un producte determinat.

Paper dins del rol de l'actor: el client accedeix voluntàriament a aquest cas d'ús, que no és imprescindible per a completar el procés de compra.

Casos d'ús relacionats: Navegar per la botiga, Editar llista de productes d'interès.

Precondicions: el client es troba en el cas d'ús Navegar per la botiga o en Editar llista de productes d'interès, i clicka sobre el botó "+info" d'un producte determinat.

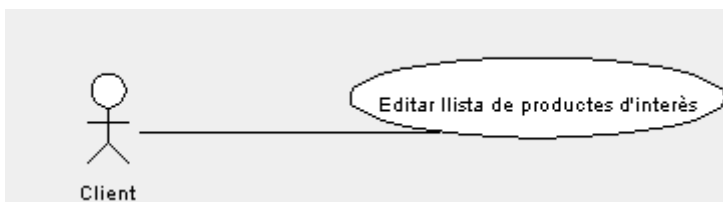
Postcondicions: es mostra al client una pàgina amb tota la informació disponible sobre aquest producte.

Descripció: el client clicka a l'opció "veure informació addicional" sobre un producte determinat i com a resposta se li mostra una pàgina amb la imatge no reduïda del producte, tots els camps de descripció del producte sense restriccions de longitud i finalment tota la informació addicional resultant de fer una cerca sobre notes que facin referència a aquest producte.

Al final se li ofereix les opcions

- Afegir/Eliminar el producte de la llista d'interès
- Afegir/Eliminar el producte del carret de la compra
- Tornar a la llista de productes

3.5.1.6. Editar llista de productes d'interès



Resum de la funcionalitat: El client consulta la llista de productes d'interès que ell mateix ha confeccionat i en pot eliminar de la llista, consultar més informació sobre un d'ells, o afegir-lo al carret de la compra.

Paper dins del rol de l'actor: el client accedeix voluntàriament a aquest cas d'ús, que no és imprescindible per a completar el procés de compra.

Casos d'ús relacionats: Navegar per la botiga, Consultar informació addicional, Editar carret de la compra.

Precondicions: el client ha afegit prèviament algun producte a la llista d'interès.

Postcondicions: es mantenen a la llista de productes d'interès tots els productes excepte els que hagi eliminat expressament el client.

Descripció: es mostra al client una llista amb els productes que prèviament ha afegit a la llista d'interès, en aquesta sessió o en qualsevol d'anterior.

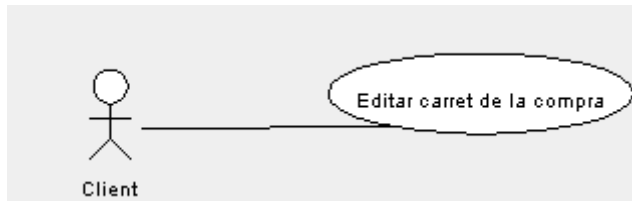
Jaume Casadesús

La llista es mostra, igual que els resultats de les cerques, en el format descrit en el cas d'ús Navegar per la botiga, amb la única diferència que ara enlloc de l'opció "Afegir a llista d'interès" s'ofereix l'opció "Eliminar de la llista d'interès".

També s'ofereix la opció de buidar la llista d'interès.

Observacions: la llista d'interès té persistència entre sessions.

3.5.1.7. Editar carret de la compra



Resum de la funcionalitat: El client podrà consultar la llista de productes afegits al carret de la compra i podrà editar-ne la quantitat de cada cadascun.

Paper dins del rol del client: el client podrà accedir opcionalment a aquest cas d'ús durant el cas d'ús Navegar per la botiga i haurà de passar-hi obligatòriament en el cas d'ús Tramitar comanda.

Casos d'ús relacionats: Navegar per la botiga, Tramitar comanda.

Precondicions: el client es troba en el cas d'ús Navegar per la botiga o en Tramitar comanda.

Postcondicions: S'han actualitzat les quantitats de cada producte seleccionat i el corresponent import.

Descripció: Es mostra una taula amb cadascun dels productes seleccionats pel client, en què apareix la quantitat, el seu codi de referència, el nom del producte, el preu unitari i l'import. Al final de la taula es mostra el total resultant de la suma de tots els imports.

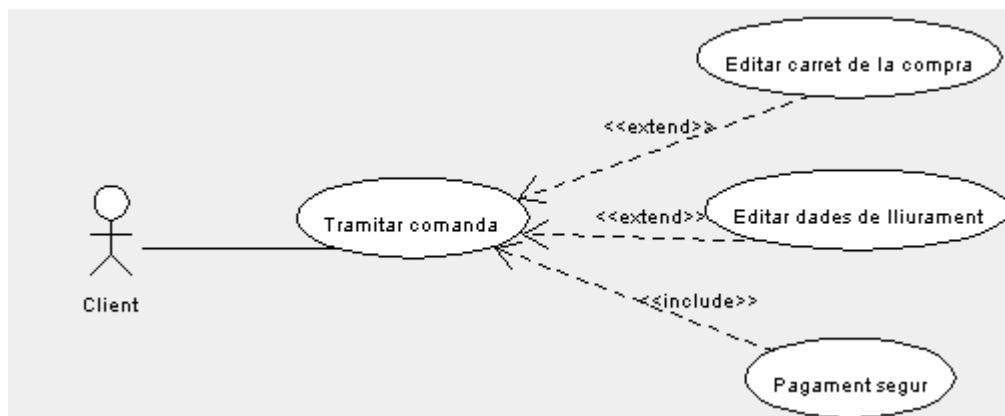
El client pot modificar la quantitat de cada producte i clicar al lloc indicat per a actualitzar els imports.

S'ofereix al client les opcions de:

- “Buidar el carret”, que elimina tots els productes del carret.
- “Tornar a la botiga”, que retorna al cas d'ús Navegar per la botiga.
- “Tramitar comanda”, que entra al cas d'ús tramitar comanda.

Observacions: Si posa un valor de zero en algun producte eliminarà aquest producte del carret.

3.5.1.8. Tramitar comanda



Resum de la funcionalitat: es completen els passos necessaris per a que s'efectuï la compra.

Paper dins del rol de l'actor: imprescindible per a que es realitzi la compra.

Casos d'ús relacionats: Editar carret de la compra, Editar dades de lliurament, Pagament segur.

Precondicions: el client ha afegit algun producte al carret de compra.

Postcondicions: el client ha acceptat les quantitats i imports especificats en el carret de la compra, ha validat les dades d'enviament i ha completat el procés de pagament electrònic. A partir del carret de la compra s'ha generat una comanda que s'ha guardat en estat "pendent de lliurament".

Descripció: Es mostra al client el carret de la compra amb les quantitats i imports actualitzats, i l'import total dels productes. A continuació se li mostra l'adreça de tramesa i modalitat d'enviament i el corresponent import.

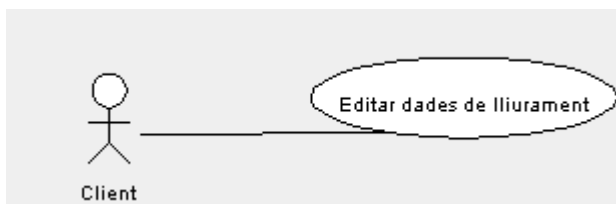
S'ofereix al client les opcions

- Tornar a Navegar per la botiga
- Editat dades i opcions de tramesa
- Acceptar i procedir al pagament

Una vegada el client accepta entra en l'aplicació de pagament electrònic segur, externa al programari "e-botiga". Un cop l'aplicació de pagament confirma que s'ha completat el pagament es mostra un missatge de confirmació al client. Quan aquest clicka per acceptar accedeix al cas d'ús desconnectar de la botiga i finalitza la sessió.

Amb les dades del carret de la compra i les dades de tramesa es crea un objecte Lliurament, en estat "pendent de processar", que es guarda a la base de dades. El carret es destrueix al tancar la sessió.

3.5.1.9. Editar dades de lliurament.



Resum de la funcionalitat: el client revisa les dades de lliurament: adreça i modalitat d'enviament.

Paper dins del rol de l'actor: cal passar-hi per a evitar errors d'enviament.

Casos d'ús relacionats: Tramitar comanda.

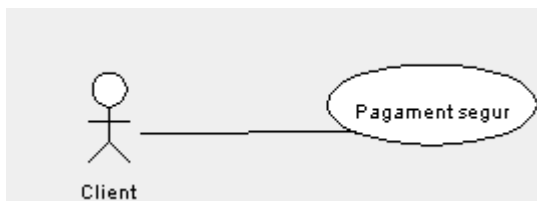
Precondicions: el client ha afegit algun producte al carret de compra.

Postcondicions: el client ha validat les dades d'enviament: adreça, modalitat i cost.

Descripció: Es mostra al client l'adreça de tramesa, que pot ser diferent a la seva, la modalitat d'enviament per defecte i el corresponent import. La modalitat d'enviament per defecte és "normal" i es pot canviar per "urgent", que té un cost més alt.

Finalment el client ha d'acceptar les dades clickant al lloc indicat.

3.5.1.10. Pagament segur.



Resum de la funcionalitat: el client efectua el pagament mitjançant tarja de crèdit.

Paper dins del rol de l'actor: imprescindible per a completar el procés de compra.

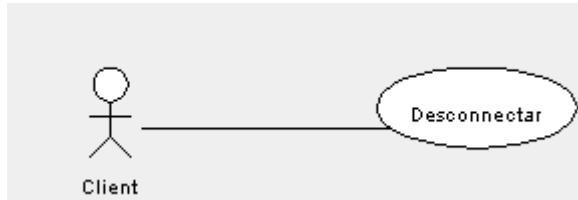
Casos d'ús relacionats: Tramitar comanda.

Precondicions: el client ha afegit algun producte al carret de compra i ha validat el contingut del carret i les dades de tramesa.

Postcondicions: el client ha fet el pagament.

Descripció: el client es connecta a una aplicació externa de pagament segur mitjançant tarja de crèdit.

3.5.1.11. Desconnectar de la botiga.



Resum de la funcionalitat: el client finalitza la sessió i surt de la botiga.

Paper dins del rol del client: necessari per a finalitzar la sessió de manera controlada.

Casos d'ús relacionats: Navegar per la botiga, Tramitar comanda.

Precondicions: el client ha clickat en l'opció desconnectar o bé ha completat el procés de tramitar comanda..

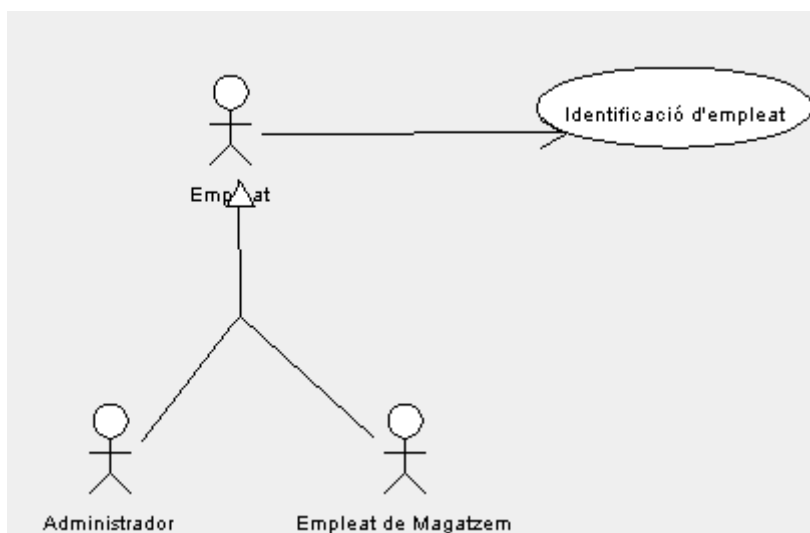
Postcondicions: ha finalitzat la sessió i el client es troba a la pantalla d'inici. S'han guardat a la base de dades les dades persistents i s'han eliminat la resta.

Descripció: El client decideix voluntàriament desconnectar de la botiga clickant a l'opció "desconnectar" o bé ha completat el procés de compra.

Finalitza la sessió, es guarden les dades necessàries i es mostra la pàgina d'inici.

3.5.2. Casos d'ús dels empleats:

3.5.2.1. Identificació d'empleat



Resum de la funcionalitat: identifica els empleats per a permetre'ls realitzar tasques de manteniment de la rebotiga.

Paper dins del rol de l'empleat: primer pas, obligatori, en cada sessió de treball de l'empleat.

Casos d'ús relacionats:

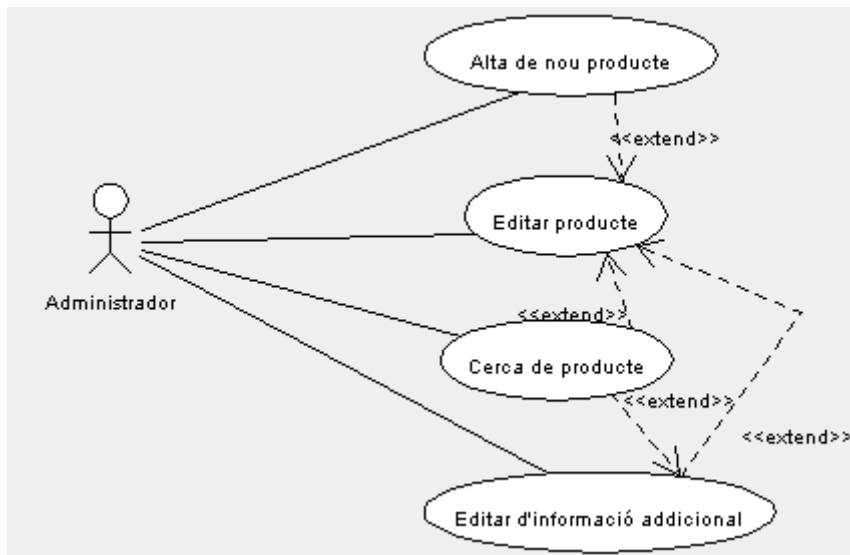
Precondicions: l'empleat ha d'estar donat d'alta en el servidor, amb un login i password que l'identifiquen.

Postcondicions: l'empleat s'ha identificat i ara podrà accedir als casos d'ús de manteniment de la botiga.

Descripció: L'empleat s'identifica amb login i password. Una vegada reconegut pel servidor se li mostra un missatge de benvinguda i un menú amb les diverses opcions, corresponents als casos d'ús en què pot intervenir.

Observacions: se suposa que els empleats es donen d'alta a través d'alguna altra aplicació o directament a la base de dades.

3.5.2.2. Editar producte



Resum de la funcionalitat: l'empleat modifica algun atribut d'algun producte.

Paper dins del rol de l'actor: necessari per a mantenir actualitzat el catàleg de productes.

Casos d'ús relacionats: Alta de nou producte, Cerca de producte, Editar informació addicional.

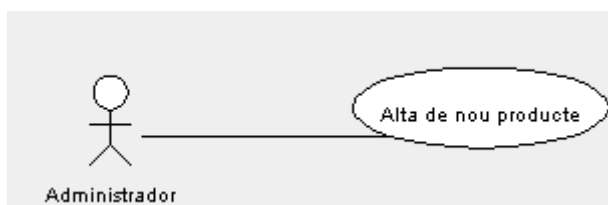
Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients i el producte existeix a la base de dades.

Postcondicions: el producte s'ha guardat a la base de dades amb les modificacions.

Descripció: l'empleat fa la cerca d'un o més productes i clicka damunt dels que vol modificar. Aleshores li apareix un formulari que mostra els valors actuals de tots els atributs persistents del producte seleccionat. Modifica els valors oportuns i clicka al lloc indicat per a actualitzar els canvis en la base de dades.

Observacions: la baixa del producte es produeix quan el valor del camp "data de baixa" és igual o inferior a la data actual. De tota manera el producte continua a la base de dades.

3.5.2.3. Alta de nou producte



Resum de la funcionalitat: l'empleat entra un nou producte.

Paper dins del rol de l'actor: necessari per a tenir productes en la botiga.

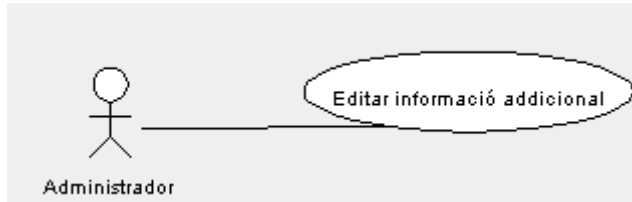
Casos d'ús relacionats: Editar producte.

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients.

Postcondicions: el producte creat s'ha guardat correctament a la base de dades.

Descripció: l'empleat omple els diversos camps d'un formulari amb els valors corresponents al producte en qüestió. Envia el formulari i com a resposta rep un missatge confirmant que s'ha guardat el nou producte o bé un missatge d'error indicant que s'ha omplert algun camp de manera incorrecta.

3.5.2.4. Editar informació adicional



Resum de la funcionalitat: l'empleat afegeix/edita informació adicional referent a un producte.

Paper dins del rol de l'actor: permet afegir/editar informació opcional a un producte.

Casos d'ús relacionats: Editar producte.

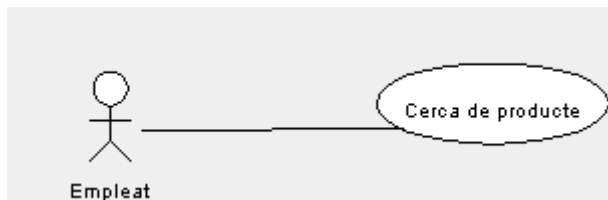
Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients i el producte existeix a la base de dades.

Postcondicions: els canvis s'han guardat a la base de dades.

Descripció: l'empleat selecciona un producte i se li mostra totes les notes d'informació adicional sobre aquest producte. Pot seleccionar una d'aquestes notes per a editar-la o eliminar-la o pot crear una nota nova.

Observacions: En un futur potser s'afegirà un atribut "tipus d'informació" que permetrà fer cerques entre les notes d'informació adicional.

3.5.2.5. Cerca de producte



Resum de la funcionalitat: l'empleat cerca un o més productes a la base de dades.

Paper dins del rol de l'actor: necessari per a gestionar els productes.

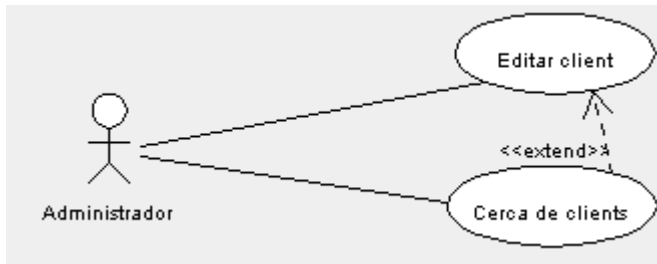
Casos d'ús relacionats: Editar productes, Editar informació adicional, Editar recomanacions.

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients.

Postcondicions: es mostra la llista de productes que compleixen les condicions de cerca.

Descripció: L'empleat omple un o més camps en un formulari de cerca i l'envia al servidor. Com a resposta se li mostra el nombre de productes que compleixen les condicions de cerca així com un llistat d'aquests productes.

3.5.2.6. Editar client



Resum de la funcionalitat: l'empleat edita un client per a modificar-ne algun valor com pot ser el descompte personalitzat que se li aplica.

Paper dins del rol de l'actor: opcional en el procés de manteniment de la botiga.

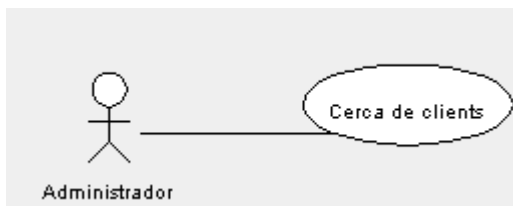
Casos d'ús relacionats:

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients i el client existeix a la base de dades.

Postcondicions: els canvis s'han guardat a la base de dades.

Descripció: l'empleat selecciona un client i se li mostra un formulari amb les dades del client. L'empleat pot modificar aquestes dades, entre elles el % de descompte que s'aplicarà a aquest client, i clicka al lloc indicat per a actualitzar els canvis en la base de dades.

3.5.2.7. Cerca de clients



Resum de la funcionalitat: l'empleat cerca un o més clients a la base de dades.

Paper dins del rol de l'actor: necessari per a gestionar els clients.

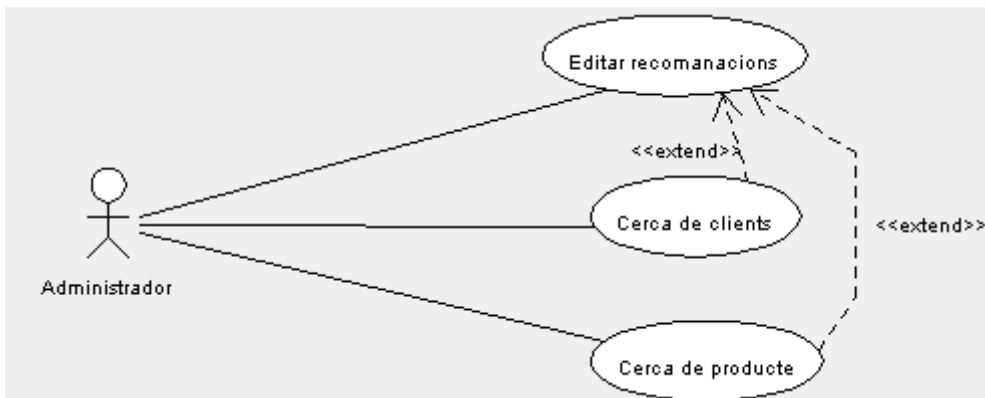
Casos d'ús relacionats: Editar client, Editar recomanacions.

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients.

Postcondicions: es mostra la llista de clients que compleixen les condicions de cerca.

Descripció: L'empleat omple un o més camps en un formulari de cerca i l'envia al servidor. Com a resposta se li mostra el nombre de clients que compleixen les condicions de cerca així com una llistat d'aquests clients.

3.5.2.8. Editar Recomanacions



Resum de la funcionalitat: l'empleat afegeix o elimina productes a una llista de recomanacions adreçada a un client determinat.

Paper dins del rol de l'actor: opcional dins del funcionament de la botiga.

Casos d'ús relacionats:

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients i tant el client com els productes existeixen en la base de dades.

Postcondicions: els canvis s'han guardat a la base de dades.

Descripció: l'empleat selecciona un client i se li mostra la llista de productes recomanats a aquest client. Pot seleccionar productes per a eliminar-los de la llista, i també pot fer cerques de productes a la base de dades per a seleccionar productes a afegir a la llista. Finalment clicka al lloc indicat per a que els canvis es facin persistents a la base de dades.

3.5.2.9. Editar Stock



Resum de la funcionalitat: l'empleat del magatzem modifica la quantitat d'un producte que hi ha en stock.

Paper dins del rol de l'actor: necessari en el manteniment de la botiga.

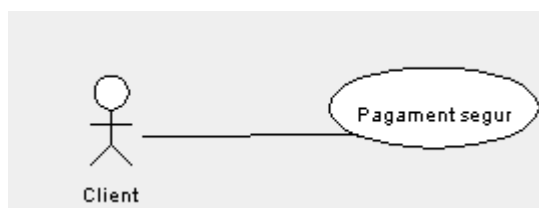
Casos d'ús relacionats:

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients i el producte existeix en la base de dades.

Postcondicions: els canvis s'han guardat a la base de dades.

Descripció: l'empleat fa una cerca i en el resultat selecciona un producte. Se li mostra un formulari en què només pot modificar la quantitat en stock. Després d'introduir el nou valor ha de clicar al lloc indicat per a que s'actualitzin els canvis en la base de dades.

3.5.2.10. Processar lliurament.



Resum de la funcionalitat: l'empleat del magatzem fa l'expedició de la comanda.

Paper dins del rol de l'actor: necessari en el procés de compra.

Casos d'ús relacionats:

Precondicions: l'empleat s'ha connectat al sistema amb els permisos adients i existeix un lliurament pendent d'expedició.

Postcondicions: el lliurament és en estat "enviat" i se li ha posat la data actual com a data d'expedició.

Descripció: l'empleat consulta els lliuraments pendents d'expedició i en canvia l'estat a "enviats". Amb això finalitza el procés de compra corresponent a una sessió del client.

4. Disseny.

4.1. Descripció general de l'arquitectura.

Tant la botiga com el backend són aplicacions amb arquitectures molt similars, cadascuna amb la seva pròpia capa de presentació basada en un client web, i que comparteixen la mateixa capa de dades i bona part de la capa de negoci.

La capa de presentació s'ha dissenyat seguint un patró de **Front Controller** amb un controlador principal per a la botiga i un altre per al *backend*, cadascun amb les seves pròpies vistes i altres classes de suport. D'aquesta manera se centralitza en un sol punt d'accés tots els processos d'autenticació, delegació del processament de la petició, creació de vistes i gestió d'errors. El seguiment d'aquest patró pot evitar la duplicació de codi, facilita mantenir la coherència entre les diverses vistes, i afavoreix el manteniment en tenir ben localitzats els punts a on es prenen decisions.

També en la capa de presentació, la navegació per la botiga segueix un patró **CompositeView**, en què la vista que s'ofereix al client està composta per diverses subvistes de manera modular, que fan de **Dispatchers**. Aquestes subvistes s'implementaran com a fitxers JSP.

La capa de negoci s'ha dissenyat seguint un patró de **Session Façade**, per tal d'obligar que totes les peticions a la part servidor passin per un sol punt d'accés, que amaga i encapsula la complexitat que hi pot haver al darrera. En aquest cas es tracta de dues **Session Façade** independents, una per a la botiga i l'altre per al *backend*. En el cas de la botiga s'apunta cap a una *stateful session bean*, per si cal mantenir la informació al llarg del procés de compra com és la identitat del client i el contingut del carret, tot i que aquesta informació també es pot mantenir a la capa de presentació, com a atributs de la sessió http. A més, en usar una *stateful session bean* es pot guardar en la sessió el resultat de les cerques que faci l'usuari i d'aquesta manera aplicar el patró **Value List Handler**, que permet fer un "cache" dels resultats de les cerques en la part servidor i enviar-ne al client porcions de llargada màxima predefinida, així que aquest ho vagi demanant. D'aquesta manera sempre s'envien al client llistes de resultats d'una mida acotada, que facilita la presentació i evita sobrecarregar la xarxa amb dades que potser el client no arribarà a consultar. En el cas del *backend* també s'ha acabat optant per una *stateful session bean* per tal de poder aplicar també el patró **Value List Handler** i permetre que l'empleat del *backend* pugui realitzar una cerca i en crides successives demanar porcions successives dels resultats de la cerca.

La major part del model del negoci és comú per a la botiga i el *backend* i consta d'un conjunt de classes la persistència de les quals es recolza en 3 *entity beans*: Producte, Comanda i Client. Cadascuna de les *entity beans* no està composta únicament per atributs de tipus senzills sinó que entre els atributs n'hi ha que són classes de llibreria (p.ex. *java.util.List*) i alguns que són classes definides en el mateix model de negoci. D'aquesta manera, seguim el patró **Composite Entity** en què els Entity Beans són "entitats de gra gros", que representen grafs d'objectes. Això ens redueix l'overhead que tindria usar entitats de gra més fi, que sobretot tindria un sobrecost important en temps de programació

Per tal que la capa de presentació pugui accedir al conjunt de dades que representen les entity beans però sense que entrin directament dins del model del negoci, s'ha aplicat un patró de **Data Transfer Object**, (DTO). Seguint aquest patró, definim unes classes *Serializable* accessibles tant per la capa de a la capa de negoci com per la capa de presentació, i omplim els atributs amb una còpia dels atributs de les entitats del model. A més, també ens interessa evitar que aquests DTO generin un acoblament massa fort entre capa de presentació i model. Per això apliquem concretament un patró de **Data Transfer HashMap**, en què els DTO són instàncies de *java.util.HashMap* en què les parelles clau-valor corresponen a "nom de l'atribut" – "valor de l'atribut". D'aquesta manera, els casos d'ús que accedeixen o modifiquen dades usen una interfície molt genèrica, reusable entre els diferents casos d'ús i que no caldrà modificar si es modifica el model. Això ens permet que moltes classes de la presentació no hagin de conèixer prèviament ni el nombre ni el nom dels atributs als quals han d'accedir, i aleshores podem modificar el nombre o nom dels atributs de les entitats sense que calgui retocar les classes de la presentació. Concretament, ens pot ser útil per a visualitzar o editar els atributs dels Productes, ja que amb el temps és molt probable que apareguin nous atributs o subclasses de productes amb diferent nombre d'atributs. Per tal de donar algunes funcionalitats

addicionals als DTO s'ha dissenyat la classe *DTOGeneric*, que hereta de *HashMap* i que usa com a claus objectes de la classe *comu.Atribut*. La classe *Atribut* encapsula un mínim d'informació per a que la capa de presentació pugui fer una primera validació de les dades al costat client, com és acceptar només entrades numèriques, acceptar o no valors nuls, etc.

En el cas de la sessió de compra, s'usa el *CarretDeCompra* com a DTO. En aquest cas se segueix un patró de **Domain Data Transfer Object**, perquè aquí la part client necessita "conèixer el significat" d'alguns dels atributs del *CarretDeCompra* o dels seus *Items* i aleshores seria contraproduent aplicar un accés genèric a aquests atributs.

4.2. Llista de Classes.

Classes de transferència de dades:

DTOGeneric, DTOProducteBotiga, CarretCompra, Item, Atribut, NotaInfo, ExcepcioNegoci, ExcepcioServei.

Capa de presentació:

Classes de control:

FrontController, FrontCompra, FrontBackend.

Vistes de la part botiga:

VistaLogin, VistaErrorLogin, VistaExcepcio, FormulariRegistre, MenuBotiga, VistaBotiga, VistaLlista, VistaFormulari, VistaLlistaProductes, FormulariCerca, VistaDetallsProduce, VistaCarretCompra, FormulariDadesLliurament, ConfirmacioComanda,

Vistes de la part backend:

MenuBackend, VistaLlistaProductesBackend, VistaLlistaComandes, VistaLlistaClients, FormulariCercaProductes, FormulariCercaClients, FormulariCercaComandes, FormulariEdicioProducte, FormulariEdicioInfoAddicional, FormulariEdicioClient, FormulariEdicioRecomanacio, FormulariEdicioComanda

Capa de negoci:

Session Beans:

SessioCompra, SessioBackend.

Altres classes d'entitats:

LlistaDInteres, Lliurament

Altres classes de control:

GestorClients, GestorComandes, GestorProductes, GestorPagament, GestorPreus, GestorLliurament.

Excepcions:

ExcepcioDTO, ExcepcioCarret, ExcepcioCompra, ExcepcioBackend, ExcepcioCerca, ExcepcioModificacio, ExcepcioPagament,

Capa de dades:

EntityBeans:

Client, Comanda, Producte.

4.3. Fitxes de classes.

Classe SessioCompra	
Responsabilitats	Col·laboracions
És la <i>session Façade</i> per a tots els casos d'ús del client que compra a la botiga.	FrontCompra GestorProductes

	GestorPagament CarretCompra Client
--	--

SessioCompra extends SessionBean	
+SessioCompra(idClient: Object): SessioCompra +getCampsCercaProductes(): List +getCategoriesProductes(): List +getOpcionsLliurament(): List +cercaProductes(HashMap): List +productesSeguents(): List +productesAnteriors(): List +validaCarret(CarretCompra):CarretCompra +getLlistaDInteres(): List +setLlistaDInteres(llista: List) +getInfoAddicional(idProducte: Object): List +getDadesLliurament(): DTOGeneric +tramitaComanda(carret: CarretCompra, dadesLliurament: DTOGeneric): int +disconnecta()	

Classe SessioBackend	
Responsabilitats	Col·laboracions
És la <i>session Façade</i> per a tots els casos d'ús de l'empleat de la botiga que realitza tasques de backend. A més, informa a la capa de presentació de quins atributs es disposa en els productes, clients i comandes.	GestorClients GestorProductes GestorComandes

SessioBackend extends SessionBean	
+getCampsCercaProductes(): List +getCampsCercaClients(): List +getCampsCercaComandes(): List +cercaProductes(HashMap): List +productesSeguents(int): List +productesAnteriors(int): List +modificaProducte(HashMap): HashMap +cercaClients(HashMap): List +clientsSeguents(int): List +clientsAnteriors(int): List +modificaClient(HashMap): HashMap +cercaComandes(HashMap): List +comandesSeguents(int): List	

+comandesAnteriors(int): List +modificaComanda(HashMap): HashMap

Classe GestorDades	
Responsabilitats	Col·laboracions
<p>Superclasse que encapsula les funcionalitats comuns a tots els gestors d'entitats de la botiga i el backend:</p> <ul style="list-style-type: none"> -accés a la base de dades per a fer altes, modificacions i cerques. -implementa ValueListIterator per tal d'aplicar el patró <i>Value List Handler</i> a l'hora de subministrar les llistes de resultats. -crea els DTOGeneric corresponents a les classes d'entitat. 	<p>EntityEJB DTOGeneric</p>

Gestor implements ValueListIterator
-llistaResultats: List -cursor: int
<p>+abstract getCampsCercables(): List +abstract cerca(condicions: HashMap): List +abstract modifica(dades: DTOGeneric): DTOGeneric // mètodes de la interfície ValueListIterator +getSize(): int +getCurrentElement():Object +getPreviousElements(count: int): List +getNextElements(count: int): List +resetIndex()</p>

Classe GestorProductes	
Responsabilitats	Col·laboracions
<p>Cerca els productes que compleixen les condicions que demana el client o l'empleat del backend.</p>	<p>Producte DadesProducte</p>

GestorProductes extends Gestor
+cerca(condicions: HashMap): List

Classe GestorClients	
Responsabilitats	Col·laboracions
<p>Cerca els clients que compleixen les condicions que demana l'administrador de la botiga. Implementa ValueListIterator per tal d'aplicar el patró <i>Value List Handler</i> a l'hora de subministrar les llistes de</p>	<p>Producte DadesProducte</p>

resultats.	
------------	--

GestorClients extends Gestor	
+cerca(condicions: HashMap): int	
+modifica(dades: DTOGeneric): DTOGeneric	

Classe GestorComandes	
Responsabilitats	Col·laboracions
Cerca els productes que compleixen les condicions que demana l'administrador de la botiga. Implementa ValueListIterator per tal d'aplicar el patró <i>Value List Handler</i> a l'hora de subministrar les llistes de resultats.	Producte DadesProducte

GestorComandes extends Gestor	
+cerca(condicions: HashMap): List	
+modifica(dades: DTOGeneric): DTOGeneric	

Classe Producte	
Responsabilitats	Col·laboracions
Persistència de les dades de cada producte que es ven a la botiga.	NotaInfo

Producte extends EntityBean	
-id: String	
-titol: String	
-autor: String	
-categoria: String	
-breuDescripcio: String	
-preu: float	
-stock: int	
-infoAddicional: List	
+getData(): HashMap	
+setData(dades: HashMap)	
+getInfoAddicional(): List	
+setInfoAddicional(List)	

Classe Client	
Responsabilitats	Col·laboracions
Persistència de les dades de cada client de la botiga.	Comanda

Client extends EntityBean	
-id: String	

-descompte: float -llistaDInteres: List
+getData(): HashMap +setData(dades: HashMap) +getLlistaDInteres(): List +setLlistaDInteres(List) -

Classe Comanda	
Responsabilitats	Col·laboracions
Persistència de les dades de cada comanda que s'efectua a la botiga. Genera una comanda a partir d'un carret de la compra i unes dades de lliurament.	Item Lliurament Client

Comanda extends EntityBean	
-id: String -estat: int -client: Client -data: Date -lliurament: Lliurament -items: List -import: float	
+Comanda(client: Client, carret: List, lliurament: Lliurament): Comanda +getData(): HashMap +setData(dades: HashMap) +getItems(): List +setItems(carret:List)	

Classe Item	
Responsabilitats	Col·laboracions
Encapsula les dades de cada línia de la comanda, com és la referència del producte i la quantitat.	Producte

Item	
-idProducte: Object -quantitat: int -nom: String -preuUnitari: float -import: float	
+Item(producte: DTOProducteBotiga, quantitat: int): Item +getIdProducte(): Object +setIdProducte(id: Object) +getQuantitat(): int +setQuantitat(quant: int) +getNom(): String	

+setNom(nom: String) +getPreuUnitari(): float +setPreuUnitari(preu: float) +getImport(): float +setImport(valor: float)

Classe CarretCompra	
Responsabilitats	Col·laboracions
Encapsula les dades referents als productes que el client vol comprar. Calcula l'import de la compra.	Item GestorPreus

CarretCompra
-items: List -importTotalProductes: float
+getItems():List +getItem(num: int): Item +addItem(Item) + getImportProductes (): float +setImportProductes(valor: float)

Classe GestorPreus	
Responsabilitats	Col·laboracions
Calcula el preu d'una determinada quantitat d'un producte per a un determinat client.	

GestorPreus
+calculaPreu(prod: Producte, quant: int, client: Client): float

Classe GestorPagament	
Responsabilitats	Col·laboracions
Encapsula tota la funcionalitat de pagament electrònic de la compra del client.	

GestorPagament
+ferPagament(client: Client, import: float)

Classe Lliurament	
Responsabilitats	Col·laboracions
Conté les dades de lliurament d'una comanda: adreça, modalitat de lliurament, opcions, etc. Controla l'estat de lliurament d'una comanda.	Comanda

Lliurament	
-adrecaEnviament: Adreça -modalitat: ModalitatLliurament -dataLliurament: Date	
+getAdrecaEnviament(): Adreça +setAdrecaEnviament(adreça: Adreça) +getModalitat(): ModalitatLliurament +setModalitat(modalitat: ModalitatLliurament) +getData(): HashMap +setData(dades: HashMap)	

Classe LlistaDInteres	
Responsabilitats	Col·laboracions
Guarda una referència als productes que el client ha marcat com a interessants per a properes sessions de compra.	Client Producte

LlistaDInteres	
+getProductes(): List +setProductes(llista: List) +afegeixProducte(idProducte: Object) +eliminaProducte(idProducte: Object)	

Classe Categoria	
Responsabilitats	Col·laboracions
Encapsula la funcionalitat de categoria de productes.	Producte

Categoria	
-nom: String	
+getDades(): HashMap +setDades(HashMap)	

Classe ModalitatLliurament	
Responsabilitats	Col·laboracions
Encapsula la funcionalitat de modalitat de lliurament de comandes.	Lliurament

ModalitatLliurament	
-nom: String	
+getDades(): HashMap +setDades(HashMap)	

Classe Adreça	
Responsabilitats	Col·laboracions
Encapsula les dades que formen una adreça	Lliurament Client

Adreça
-nom: String -cognoms: String -email: String -adrecaCarrer: String -codiPostal: String -municipi: String -pais: String
+getDades(): HashMap +setDades(HashMap)

Classe DTOGeneric	
Responsabilitats	Col·laboracions
Fa de Data Transfer Object genèric a través de la funcionalitat d'un HashMap. A més, proposa a les classes de la presentació una ordenació dels atributs i els informa de si cadascun dels atributs és editable o no. Fa una primera validació de dades al client.	Gestor GestorLlistes GestorFormularis

DTOGeneric extends HashMap
+getClausAtributs(): List +isEditable(clauAtribut: String): boolean +isAllowedNull(clauAtribut: String): boolean +isNumeric(clauAtribut: String): boolean

Classe ExcepcioDTO	
Responsabilitats	Col·laboracions
Excepcio produïda a l'intentar donar un valor no permès a un camp d'un DTOGenèric.	SessioCompra

Classe ExcepcioCompra	
Responsabilitats	Col·laboracions
Excepcio produïda en el servidor durant el procés de compra.	SessioCompra

Classe ExcepcioBackend	
Responsabilitats	Col·laboracions
Excepcio produïda en el el servidor durant procés de backend.	SessioBackend

Classe ExcepcioCerca	
Responsabilitats	Col·laboracions
Excepcio produïda en el servidor al fer una	SessioCompra

cerca.	SessioBackend GestorComandes GestorProductes GestorClients
--------	---

Classe ExcepcioModificacio	
Responsabilitats	Col·laboracions
Excepcio produïda en el servidor al fer una modificació en una entitat.	SessioCompra SessioBackend GestorComandes GestorProductes GestorClients

Classe ExcepcioCarret	
Responsabilitats	Col·laboracions
Excepcio produïda en el carret de la compra	SessioCompra

Classe ExcepcioPagament	
Responsabilitats	Col·laboracions
Excepcio produïda en el servidor al fer el pagament de la comanda.	SessioCompra GestorPagament

Capa de presentació:

Classe FrontController	
Responsabilitats	Col·laboracions
Superclasse per a encapsular les funcionalitats comuns al controlador de la botiga i el del backend.	VistaLogin VistaLoginError VistaExcepcio GestorLogin GestorLlistes GestorFormularis

FrontController extends Servlet	
<pre>+creaVistaLogin(): VistaLogin +creaVistaErrorLogin(): VistaErrorLogin() +checkLogin(login: String, password String): boolean +mostraLlista(List): VistaLlista +mostraFormulari(HashMap): VistaFormulari +processaFormulari(formulari: String): HashMap</pre>	

Classe FrontCompra	
Responsabilitats	Col·laboracions
FrontController per a la part de presentació de la botiga. Centralitza la creació de totes les vistes i és el punt d'entrada de totes les peticions.	SessioCompra FormulariRegistre MenuBotiga VistaBotiga

	FormulariCerca VistaDetallsProducte VistaCarretCompra FormulariDadesLliurament ConfirmacioDeComanda
--	---

FrontCompra extends FrontController	
<pre> +mostraFormulariRegistre() +creaVistaBotiga(idClient: String): VistaBotiga +creaVistaCarret(idClient: String): VistaCarretCompra +creaFormulariCerca (): FormulariCerca +creaVistaDetallsProducte(idProducte: String): VistaDetallsProducte +creaFormulariLliurament(): FormulariDadesLliurament +creaConfirmacioDeComanda(): ConfirmacioDeComanda +cercaProducte(textFormulari: String): VistaLlista +afegeixProducteACarret(idProducte: String, idClient: String): int +afegeixProducteALlistaDInteres(idProducte: String, idClient: String): int +modificaCarret(textFormulari t: String, idClient): VistaCarret </pre>	

Classe FrontBackend	
Responsabilitats	Col·laboracions
Front Controller per a la capa de presentació del backend. Centralitza la creació de totes les vistes i és el punt d'entrada de totes les peticions.	SessioBackend MenuBackend VistaLlistaProductesBackend VistaLlistaComandes VistaLlistaClients FormulariCercaProductes FormulariCercaClients FormulariCercaComandes FormulariEdicioProducte FormulariEdicioClient FormulariEdicioComanda FormulariEdicioRecomanacio

FrontBackend extens FrontController	
<pre> +creaVistaBackend(): VistaBackend +creaFormulariCercaProductes(): FormulariCercaProductes +creaFormulariCercaClients(): FormulariCercaClients +creaFormulariCercaComandes(): FormulariCercaComandes +creaFormulariEdicioProductes(): FormulariEdicioProducte +creaFormulariEdicioClients(): FormulariEdicioClient +creaFormulariEdicioComandes(): FormulariEdicioComanda +creaFormulariEdicioRecomanacio(): FormulariEdicioRecomanacio +cercaProducte(textCerca: String): VistaLlistaProductes +cercaClient(textCerca: String): VistaLlistaClients +cercaComandes(textFormulari: String): VistaLlistaComandes +modificaProducte(textFormulari: String): VistaLlistaProductes </pre>	

+modificaClient(textFormulari: String): VistaLlistaClient +modificaComanda(textFormulari: String): VistaLlistProductes +afegeixRecomanacio(idClient: String, idProducte: String): int

Classe VistaLlista	
Responsabilitats	Col·laboracions
Encapsula la funcionalitat comuna a les diferents vistes de llistes de productes.	GestorLlistes

Classe VistaFormulari	
Responsabilitats	Col·laboracions
Encapsula la funcionalitat comuna a les diferents vistes de formularis.	GestorFormularis

Classe VistaLogin	
Responsabilitats	Col·laboracions
Mostra instruccions per a entrar login i password o bé donar-se d'alta com a nou client.	FrontController

Classe VistaErrorLogin	
Responsabilitats	Col·laboracions
Mostra fallada en el procés d'autenticació del client	FrontController

Classe VistaExcepcio	
Responsabilitats	Col·laboracions
Mostra missatge d'error quan es produeix alguna excepció.	FrontController

Classe FormulariRegistre	
Responsabilitats	Col·laboracions
Formulari per a que l'usuari entri les dades per a enregistrar-se com a nou client	FrontCompra

Classe MenuBotiga	
Responsabilitats	Col·laboracions
Menú amb les opcions per als clients de la botiga.	FrontCompra

Classe VistaBotiga	
Responsabilitats	Col·laboracions
Vista principal de la botiga	FrontCompra

Classe VistaLlistaProductes	
Responsabilitats	Col·laboracions
Mostra un llistat de productes al client	FrontCompra

Classe FormulariCerca	
Responsabilitats	Col·laboracions
Mostra un formulari per a que el client ompli els camps adients per a cercar un producte.	FrontCompra

Classe VistaDetallsProducte	
Responsabilitats	Col·laboracions
Mostra tots els camps d'un producte, incloent totes les notes d'informació addicional.	FrontCompra

Classe VistaCarretCompra	
Responsabilitats	Col·laboracions
Mostra el carret de la compra, en què per a cada línia es pot editar la quantitat.	FrontCompra

Classe FormulariDadesLliurament	
Responsabilitats	Col·laboracions
Mostra un formulari en què el client pot entrar/modificar les dades de lliurament de la compra.	FrontCompra

Classe ConfirmacioDeComanda	
Responsabilitats	Col·laboracions
Mostra una pantalla amb tota la informació sobre el contingut del carret de la compra i les dades de lliurament, en què el client ha de confirmar la seva acceptació abans de tramitar la comanda.	FrontCompra

Classe MenuBackend	
Responsabilitats	Col·laboracions
Menu amb les diverses opcions del backend.	FrontBackend

Classe VistaLlistaProductesBackend	
Responsabilitats	Col·laboracions
Mostra una llista de productes a l'empleat om a resultat d'una cerca al backend.	FrontBackend

Classe VistaLlistaComandes	
Responsabilitats	Col·laboracions
Mostra una llista de comandes a l'empleat.	FrontBackend

Classe VistaLlistaClients	
Responsabilitats	Col·laboracions

Mostra una llista de clients a l'empleat..	FrontBackend
--	--------------

Classe FormulariCercaProductes	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat faci una cerca de productes.	FrontBackend

Classe FormulariCercaClients	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat faci una cerca de clients.	FrontBackend

Classe FormulariCercaComandes	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat faci una cerca de comandes.	FrontBackend

Classe FormulariEdicioProducte	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat modifiqui un producte.	FrontBackend

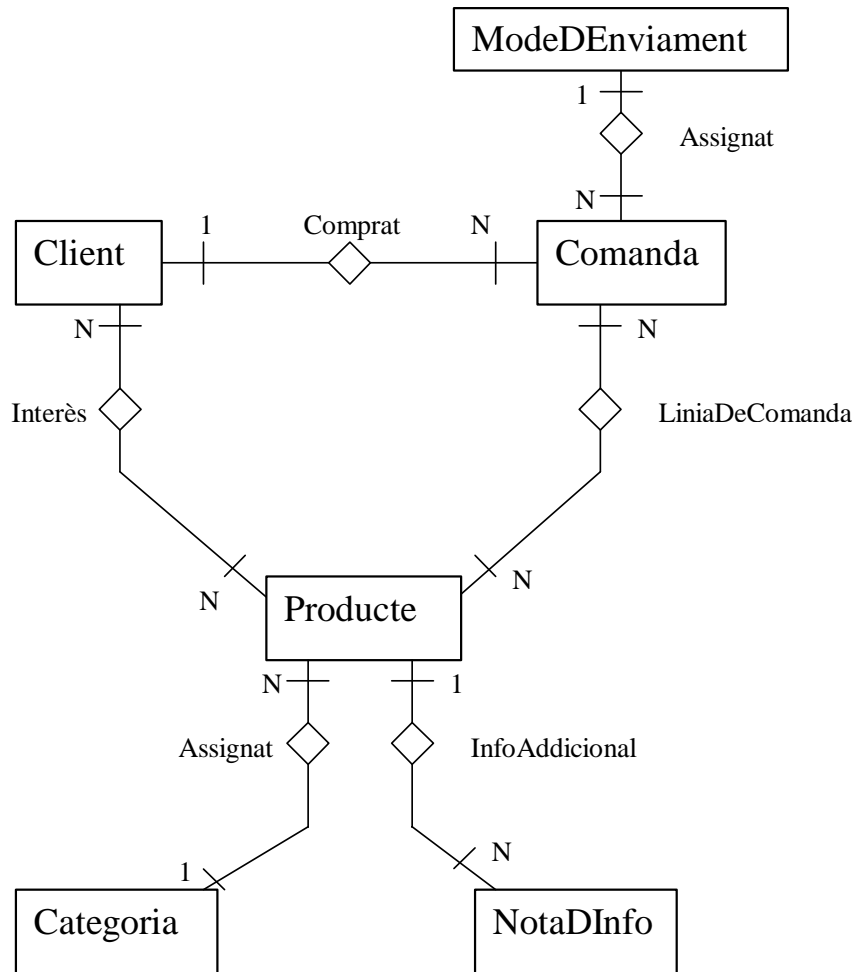
Classe FormulariEdicioInfoAdicional	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat entri informació adicional sobre un producte.	FrontBackend

Classe FormulariEdicioClient	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat faci una cerca de clients.	FrontBackend

Classe FormulariEdicioComanda	
Responsabilitats	Col·laboracions
Mostra un formulari per a que l'empleat faci una cerca de comandes.	FrontBackend

4.4. Disseny de la Base de Dades.

Figura 5: diagrama ER.



D'una banda s'han considerat les entitats representades per les EntityBeans: Producte, Client i Comanda. Després, les entitats que estan incloses en les anteriors: NotaDInfo, Categoria, ModeDEnviament. I també les relacions entre aquestes entitats, sobretot Línia de comanda, que relaciona N Productes amb M Comanda, Interès, que relaciona N Client amb M Producte i InfoAddicional, que relaciona un Producte amb N NotaDInfo.

Entitats:

Client (idClient, nom, cognoms, email, adrecaPostal, codiPostal, municipi, país, descompte)

Producte(idProducte, idCategoria, titol, autor, breuDescripcio, preu, stock)
{idCategoria} clau forana cap a Categoria

Comanda(idComanda, idClient, estat, dataComanda, dataLliurament, import, nomDestinatari, adrecaDestinatari, modalitatLliurament)
{modalitatLliurament} clau forana cap a ModalitatDEnviament
{idClient} clau forana cap a Client

Categoria(idCategoria, nom_categoria)

NotaDInfo(idNotaDInfo, idProducte, text)

{idProducte} clau forana cap a Producte

ModeDEnviament(idModeDEnviament, nom, preu)

Interrelacions:

Interès(idClient, idProducte, tipus_d_interes)

{idClient} clau forana cap a Client

{idProducte} clau forana cap a Producte

LiniaDeComanda(idComanda, idProducte, quantitat, import)

{idComanda} clau forana cap a Comanda

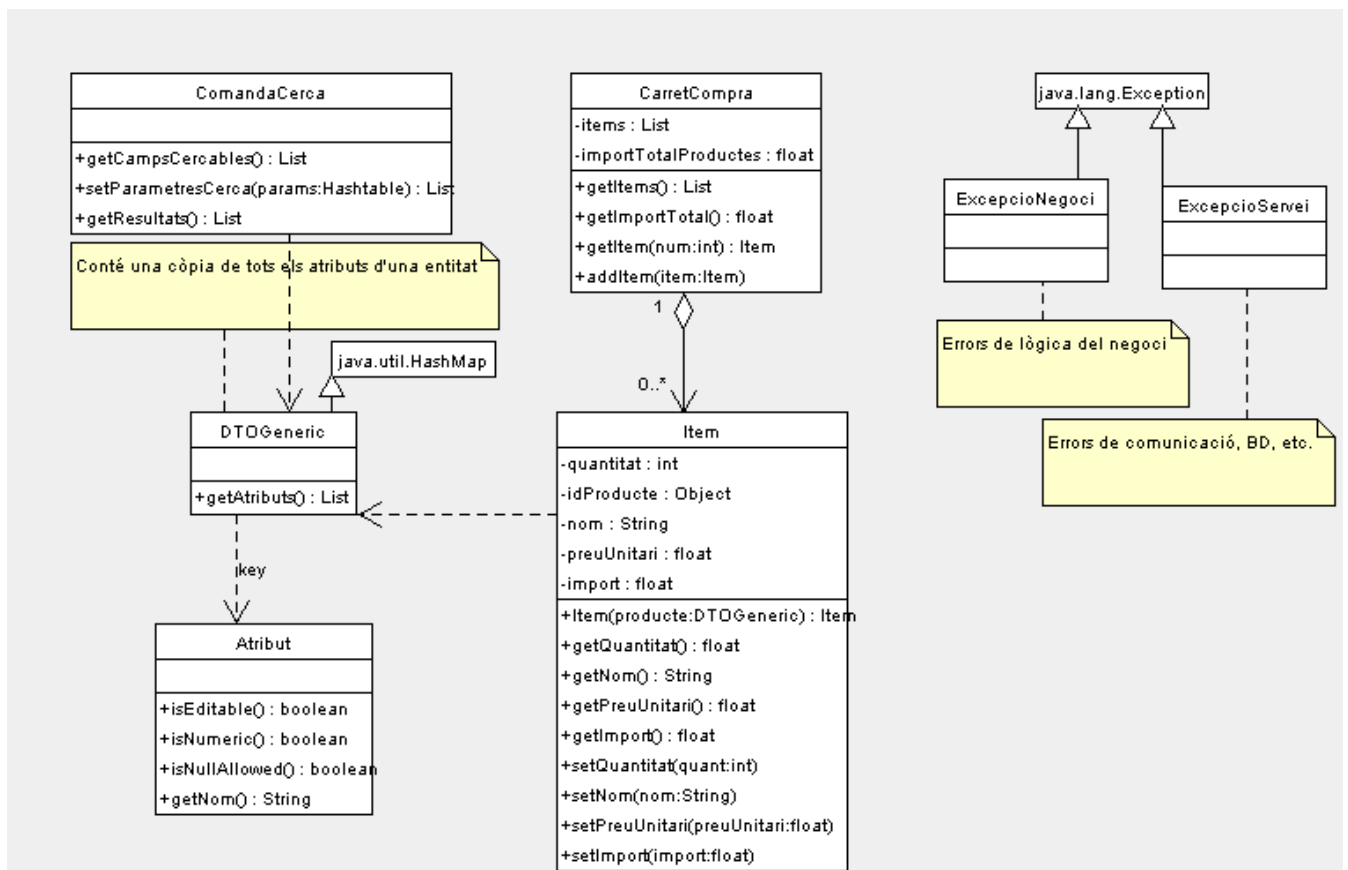
{idProducte} clau forana cap a Producte

4.5. Diagrames de classes.

4.5.1. Classes de transferència de dades.

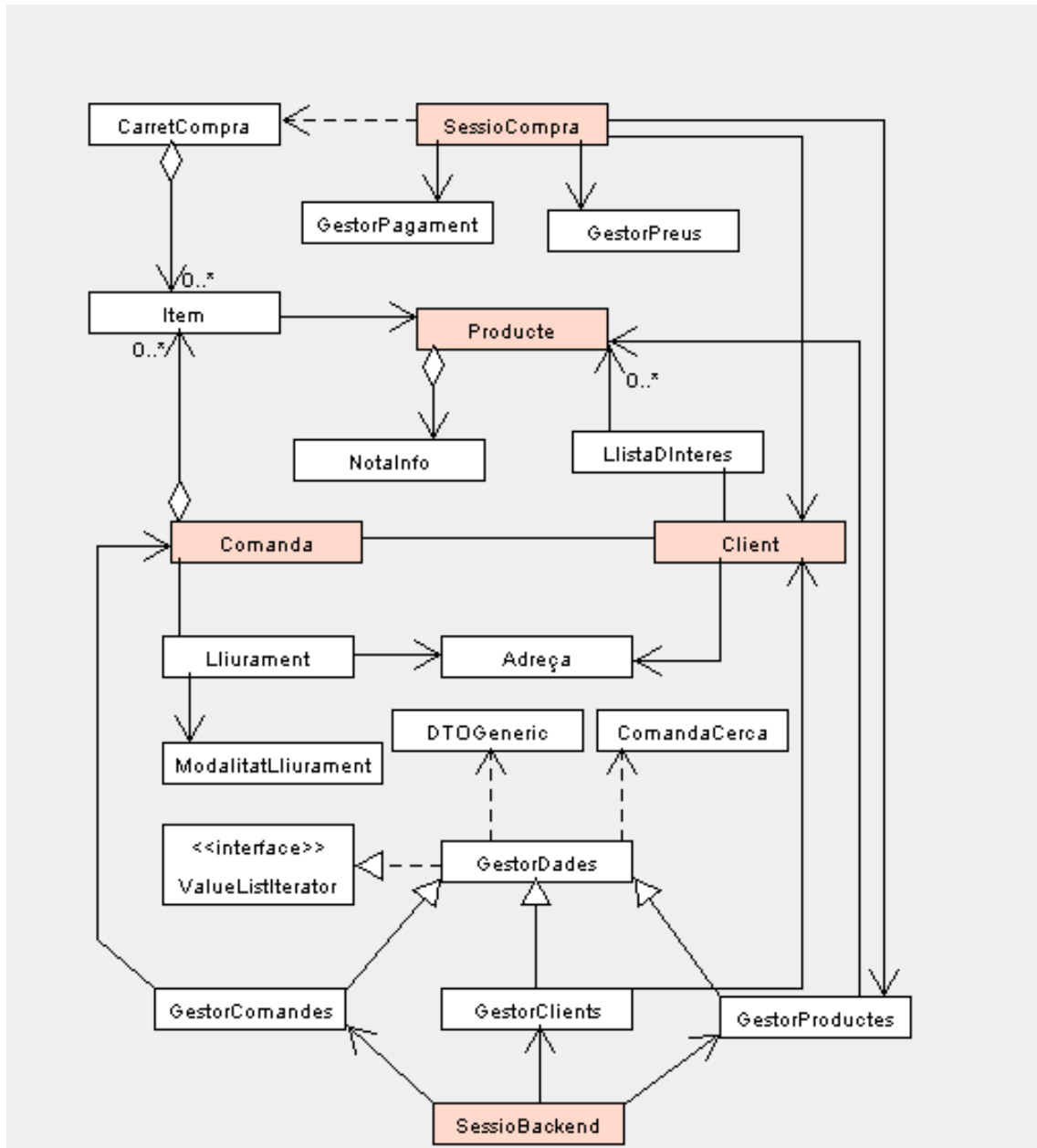
Aquestes classes són en un paquet visible tant per la capa de presentació com per la del domini i permeten aplicar alguns patrons de disseny que afavoreixen un mínim acoblament entre les dues capes. Concretament ComandaCerca s'usa per a aplicar el patró "Command" en les peticions de cerques; DTOGeneric per a aplicar el patró "HashMap Data Transfer Object" al crear o modificar entitats del domini, i CarretCompra per a aplicar "Domain Data Transfer Object" en la interacció entre les dues capes alhora de tramitar la compra.

Figura 6: classes comuns entre presentació i negoci.



4.5.2. Model

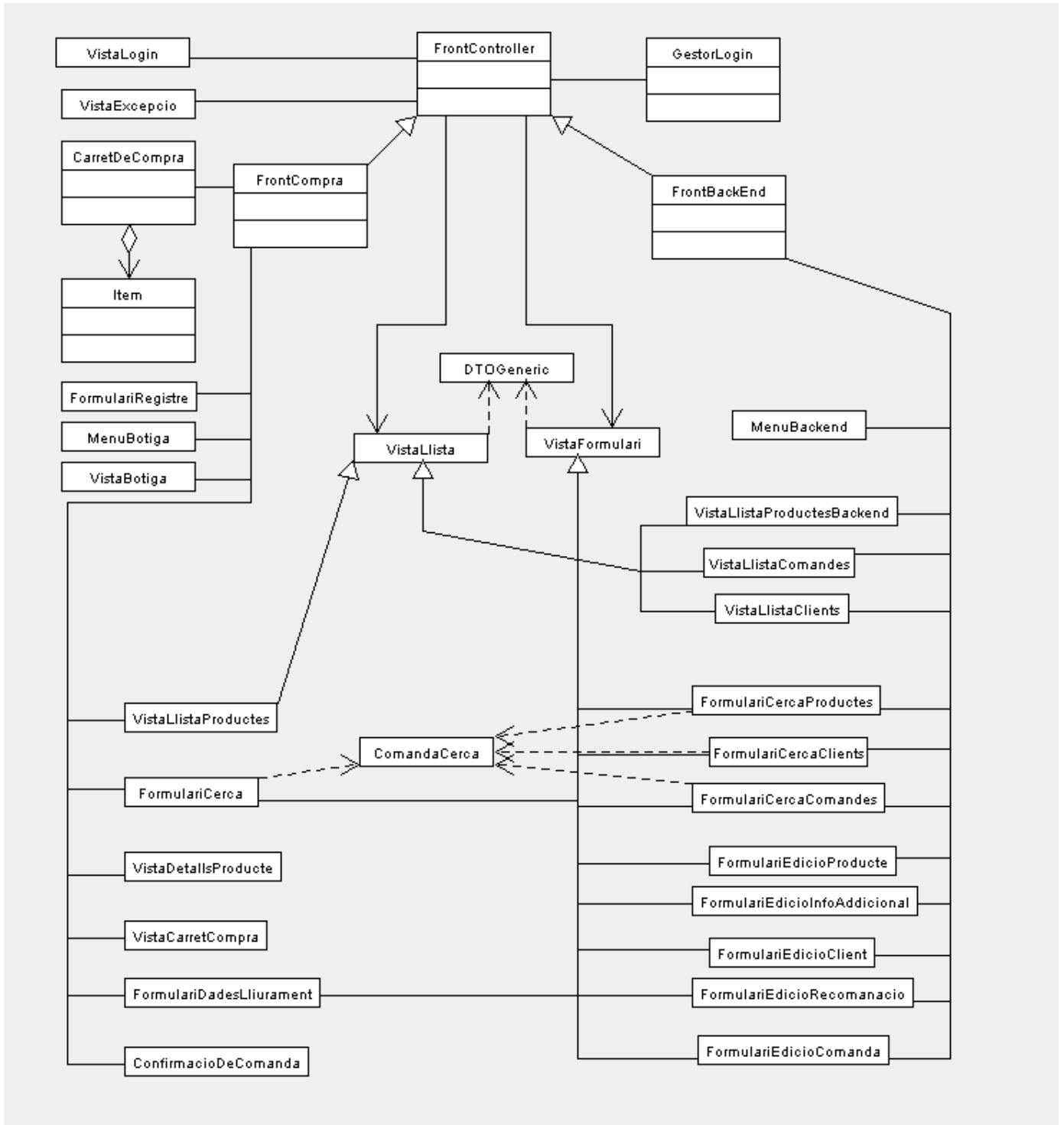
Els SessionBeans tenen associacions amb diverses classes de control a qui deleguen funcionalitats específiques del domini. Aquestes classes de control usen les classes de transferència de dades definides en el paquet comu, que poden intercanviar amb la capa de presentació. Al mateix temps, les classes de control tenen associacions amb les entitats de la capa de dades. **Figura 7: classes del model de negoci**



4.5.3. Presentació

Seguint el patró “*Front Controller*” es centralitza el control de la presentació en dos servlets, *FrontCompra* i *FrontBackEnd* que usen una colla de JSPs com a *Dispatchers* per a preparar les vistes específiques de cada cas d'ús. En la classe *FrontController* s'ha implementat la funcionalitat comú als dos servlets.

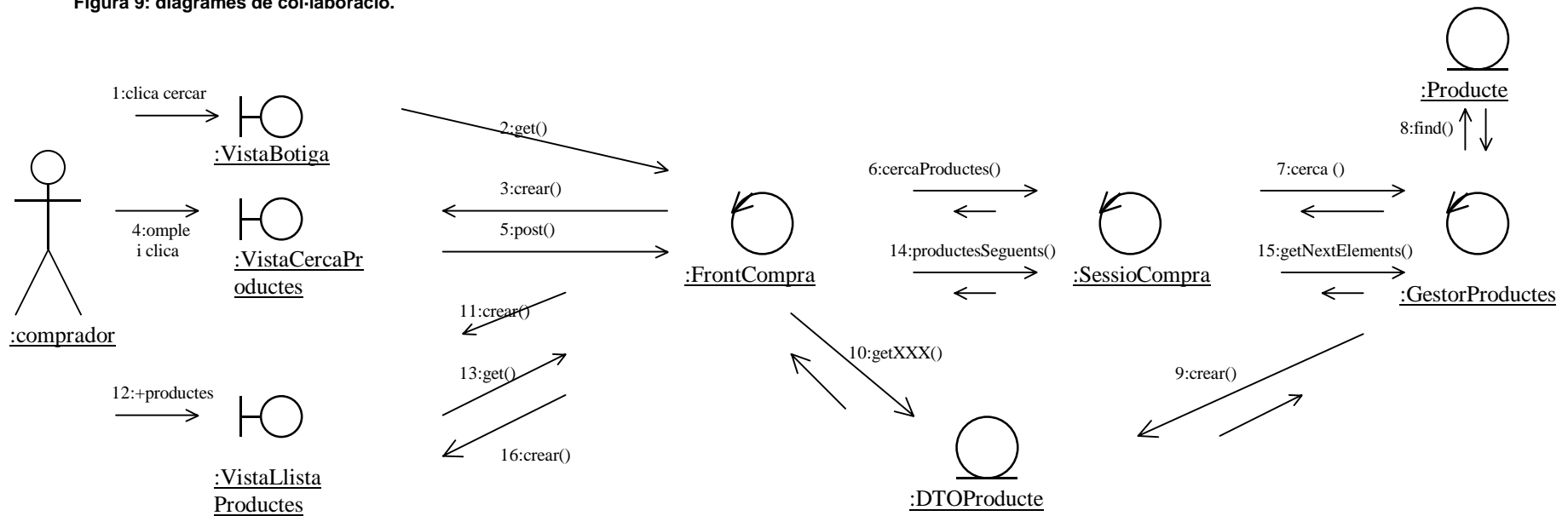
Figura 8: diagrama de classes de la presentació.



4.6. Diagrames de col·laboració.

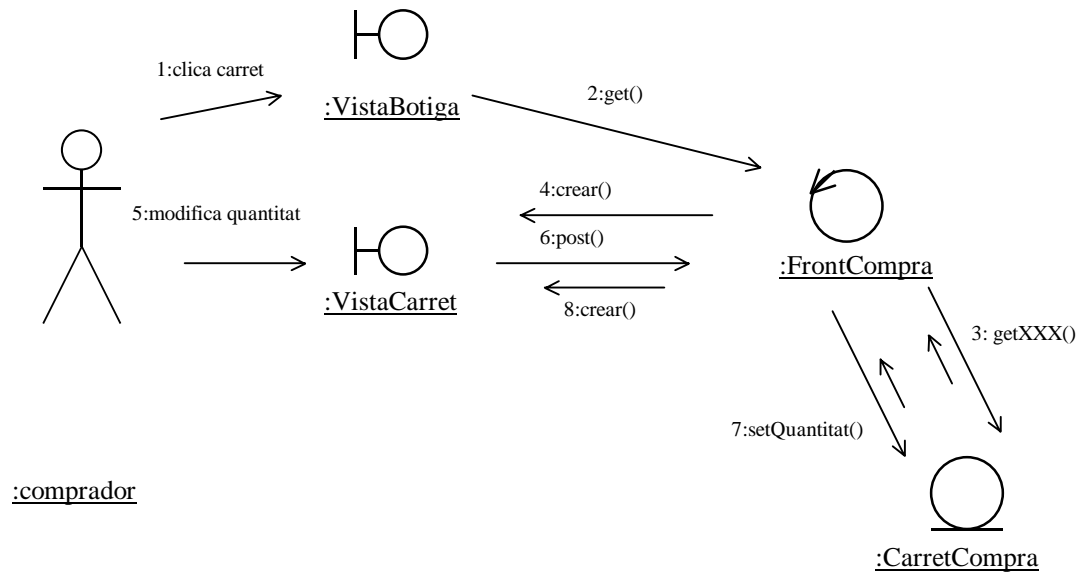
4.6.1. Cas d'ús cerca de productes a la botiga.

Figura 9: diagrames de col·laboració.

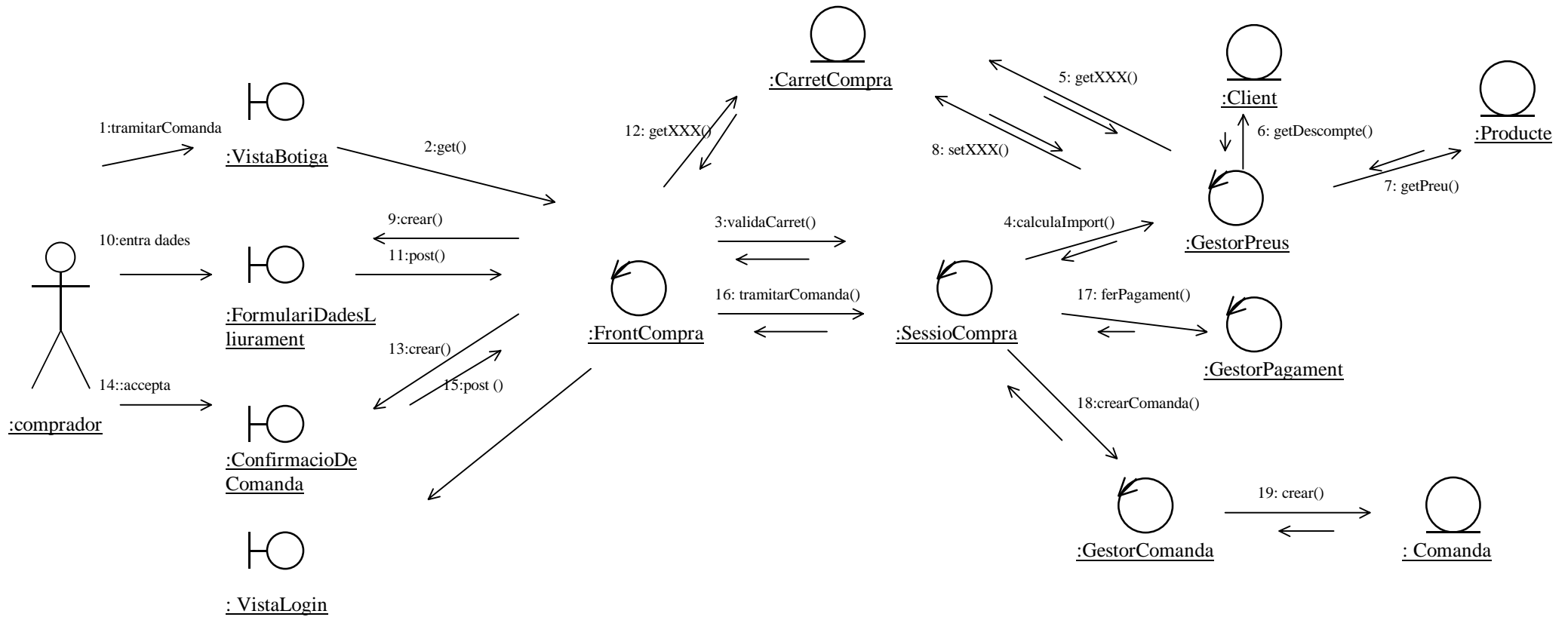


(les col·laboracions per als casos d'ús de cerca de productes, clients o comandes al backend són equivalents a aquesta).

4.6.2. Cas d'ús Editar carret:



4.6.3. Cas d'ús Tramitar comanda:



4.7. Disseny de les interfícies d'usuari.

4.7.1. VistaLogin:

Login - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

e-libreria

Per a accedir a la botiga cal que us identifiqueu amb login i password

login:

password: [enviar](#)

Si encara no ets client de **e-libreria**...
clica [aquí](#) per donar-te d'alta.

4.7.2. FormulariRegistre:

FormulariRegistre - Microsoft Internet Explorer

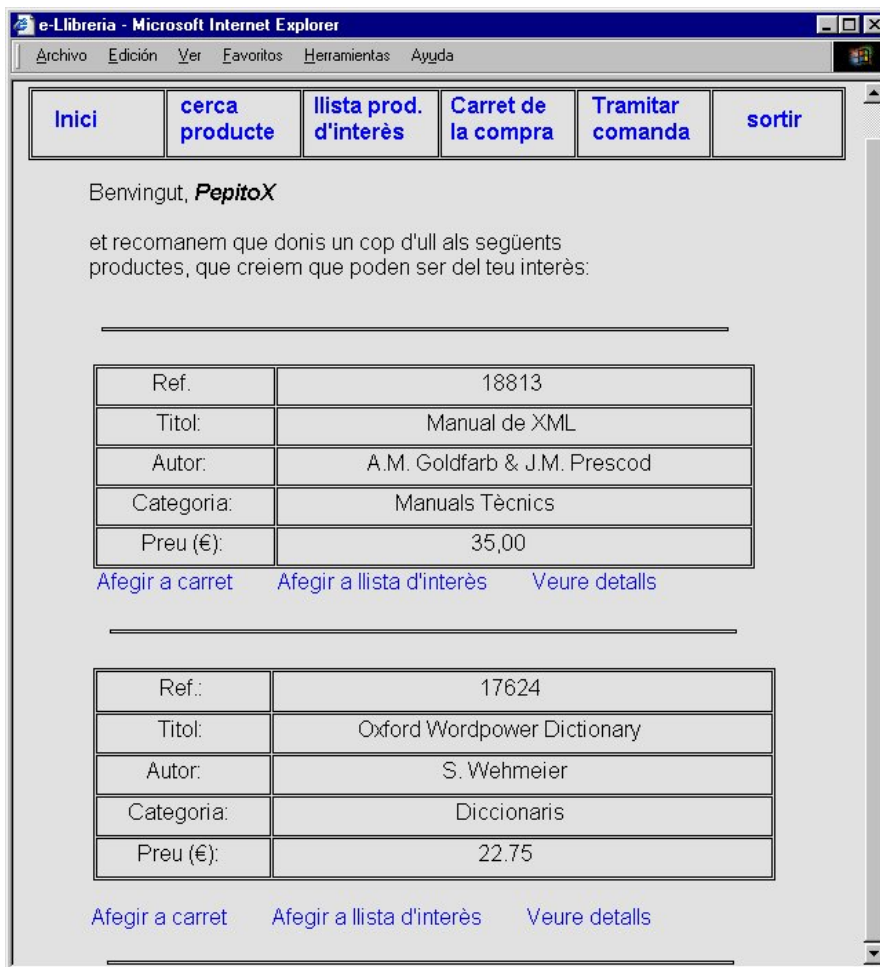
Archivo Edición Ver Favoritos Herramientas Ayuda

Registre de nou client

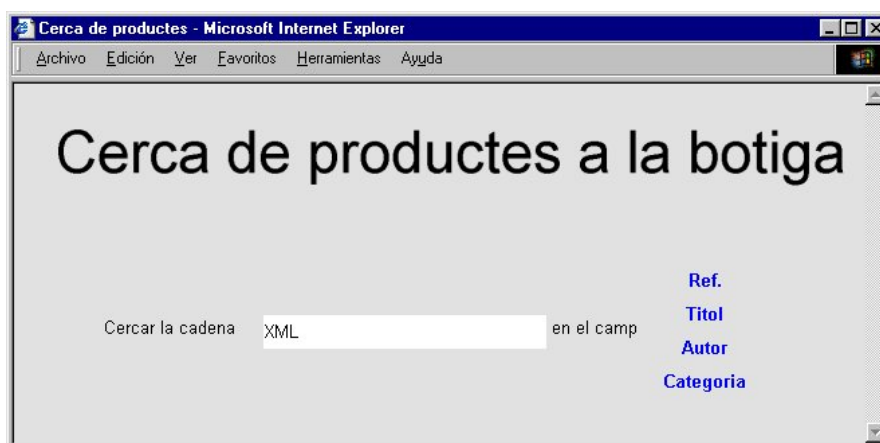
login	PepitoX
password	
repeteix password	
@ e-mail	
nom	
cognoms	
adreça postal	
codi postal	
municipi	
país	

[envia](#)

4.7.3. VistaBotiga:



4.7.4. FormulariCerca:



Els resultats de la cerca dels productes es mostren en una **VistaLlistaProductes** com la que apareix en la **VistaBotiga**.

4.7.5. VistaDetailsProductes:

Details Producte - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inici **cerca producte** llista prod. d'interès **Carret de la compra** Tramitar comanda sortir

Ref.	18813
Títol:	Manual de XML
Autor:	A.M. Goldfarb & J.M. Prescod
Categoria:	Manuais Tècnics
Preu (€):	35,00
Descripció:	Hakn asfjkn asfjkn asfjios jkodijis iahja sfh asfhjba asdfjhg asfth asfhjif afh ahjhsf oashf aoihq wr oashf asfoj asfhsouiyqbjb shj2 jhf agsk sjhahsj fujbhsdklbs hjks sdhghj asghjbs fhjsjnf asf sjisd kklk aolls gfkpm.

Informació adicional:

- Obra de referència recomanada per O. Ajnhst, cap del Dept. de Desenvolupament de adqrrfc sdgad sdgsad asfag akfkasf. Akj asfikhnan asfjkm afklkj d asfklkagl, mndg jkjalnm af lm naf.
- Mjafskn akj akbjfsk ahfopw hjia hhus agguif aghjid fgaj ghklag bhsla hkls ghajk fhulaf gyhjasf ghjkaf gafjm afghjkaf yaifkm afhjk, af huakf af asfh af jkn. lkaf kjaf kjhaf ojkbafofkahf kjhfha ljkhaf ljhaf lkahf lkahfb lkahn pñakfh lkaf lksgd ñsgj ñilkhgs oqt agljkh qtljk agljkh agljb agljkh aglkh aglkha g, lkha lk faln 2 gfkna glkbna fjkbn dkhklkma lkhnafg.
- A af gaf vbafuh afjhvb afkj b fakjgb afkj b fakj b afkn afkj b afkj b fakj galjkbaglknagf kjbagf jbbagf, nmaf kjbaf lkf lkha.

4.7.6. VistaCarretCompra:

CarretCompra - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Carret de la compra

Quant.	Ref.	Títol	preu	import
1	12387	La quarta mà	6.95	6.95
1	12462	Huye rápido, vete lejos.	7.95	7.95
2	18561	Agenda 2004	4.00	8.00
1	11051	Guia Istanbul	9.99	9.99

Import total: **32.89 €**

Clica [aquí](#) per actualitzar l'import

Clica [aquí](#) per tornar a la botiga

4.7.7. FormulariDadesLliurament:

Nom destinatari:	John Smith
Adreça destinatari:	C/ Major, 37, 2n, 2a 07043 Lleida
Mode d'enviament:	estàndard

Clica [aquí](#) per a enviar les dades

4.7.8. ConfirmacioDeComanda:

Quant.	Ref.	Títol	preu unitari	import
1	12387	La quarta m	6.95	6.95
1	12462	Huye rápido,	7.95	7.95
2	18561	Agenda	4.00	8.00
1	11051	Guia	9.99	9.99

Nom destinatari:	John Smith
Adreça d'enviament:	C/ Major 34, 2n 2a, 07128, Lleida
Modalitat d'enviament:	normal

Import productes: **32.89 €**
despeses d'enviament: **5.00 €**
Import total **37.89 €**

Clica [aquí](#) per tramitar la comanda
Clica [aquí](#) per tornar a la botiga

4.7.9. VistaBackend:



4.7.10. FormulariCercaProductes:



FormulariCercaClients i *FormulariCercaComandes* s'han dissenyat de manera semblant al *FormulariCercaProductes*.

Els resultats de les cerques es mostren de manera semblant a la *VistaLlistaProductes*.

4.7.11. FormulariEdicióProductes:

The screenshot shows a Microsoft Internet Explorer window titled 'Edició de Productes - Microsoft Internet Explorer'. The browser's menu bar includes 'Arhivo', 'Edició', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. The main content area is titled 'Edició de productes' and contains a form with the following fields:

Ref:	12645
Titol:	Practical English Usage
Autor:	Michael Swan
Any:	1980
Editorial:	Oxford University Press
Categoria:	Diccionaris
Preu (€):	22.95
Stock:	21
Descripció:	Guia de referència per a estudiants d'anglès intermitjos i avançats. Conté articles concisos sobre la majoria de problemes que causen més dificultats als estudiants.

Below the form, there is a text prompt: 'Per editar les notes d'informació addicional sobre aquest producte clica [aquí](#).' At the bottom right of the form area, there is a button labeled 'envia els canvis.'

FormulariEdicioInfoAddicional, **FormulariEdicioClient**, **FormulariEdicioRecomanació** i **FormulariEdicioComanda** tenen un disseny equivalent a **FormulariEdicioProducte**.

5. Implementació i proves.

5.1. Procés d'implementació.

Per a la implementació del codi java d'aquest projecte s'ha usat Eclipse 2.1.2 amb el plugin JBossIDE 1.0.3. Les llibreries externes que s'han usat han estat *servlet.jar*, *jbossall-client.jar* i *log4j.jar*. La compilació i desplegament s'han fet amb Ant1.5 des d'una finestra de sistema ja que la versió de sistema operatiu no permetia cridar-lo directament des d'Eclipse. Ha estat necessari crear i editar manualment els fitxers *build.xml* i *ejb-jar.xml* corresponents a cada component desplegable.

El servidor d'aplicacions sobre el que s'ha desplegat i provat l'aplicació ha estat JBoss 3.2.1 amb Tomcat 4 incorporat com a servidor web. Pel què fa a la persistència s'ha usat MySQL 4.0.13 com a sistema gestor de bases de dades. Per a la part client s'ha usat el programa AceHTML 5 Freeware com a editor HTML.

En la fase d'implementació s'han posat en pràctica els coneixements adquirits al llarg del TFC sobre el desenvolupament de Session Beans i Entity Beans, així com el desenvolupament dels clients web que usen aquests EJB.

Donada la meua inexperiència en aplicacions distribuïdes i en J2EE en particular, era previsible que la fase d'implementació requeriria un esforç molt important. Finalment aquest esforç ha quedat dins d'un rang raonable (entre 60 i 80 hores) gràcies a l'aplicació dels patrons de disseny recomanats en la bibliografia. Amb tot, l'eficiència inicial ha estat baixa degut

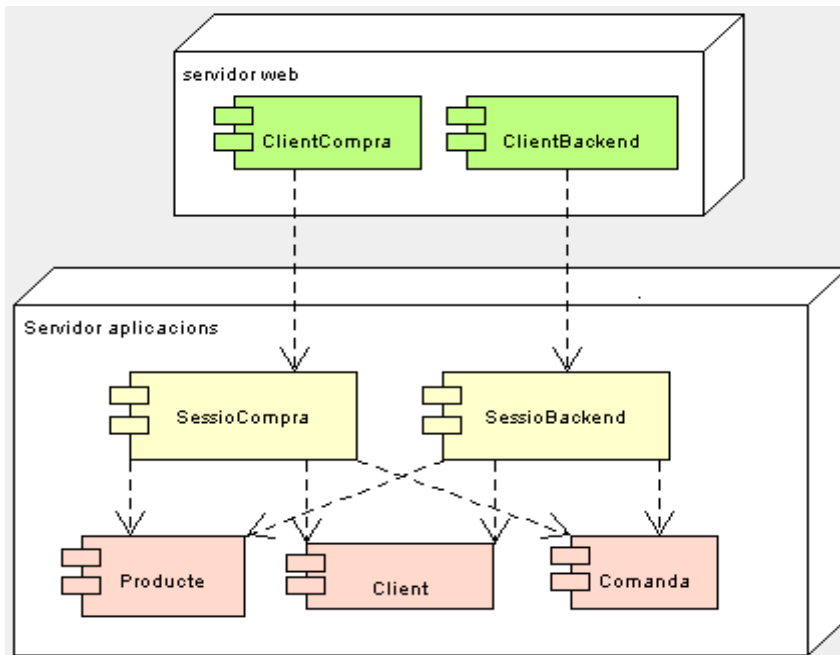
principalment al desconeixement de l'eina Ant, concretament la dificultat en adaptar els fitxers *build.xml*, així com a la dificultat inicial per entendre els ambigus missatges d'error que mostra JBoss en temps de desplegament i execució.

5.2. Resultat de la implementació:

Diagrama de components.

Com a resultat de la fase d'implementació s'han obtingut 2 clients web, 2 SessionBeans i 3 EntityBeans.

Figura 11: diagrama de components.



Pràcticament s'han implementat tots els casos d'ús tal com s'havien especificat en la fase d'anàlisi. Donades les limitacions de temps s'han simplificat alguns detalls de presentació en considerar que no eren gaire rellevants per a l'objectiu docent d'aquest treball. Alguns casos d'ús especificats a la fase d'anàlisi s'han fusionat dins d'altres com ara "editar stock" que s'ha posat dins de "editar producte". S'ha descartat el cas d'ús "afegir recomanacions a un client" perquè implicava unes hores de feina addicional i tècnicament no aportava res de nou.

En aquest projecte s'ha posat l'èmfasi en l'arquitectura més que no pas en cobrir exhaustivament totes les funcionalitats d'un cas real de comerç electrònic. Concretament s'ha enfatitzat en la separació de presentació, negoci i dades, i en l'aplicació de patrons de disseny que proporcionin solucions eficaçes en la interacció entre capes. La majoria de casos d'ús són molt simples comparats amb una aplicació real. Igualment, els atributs de les entitats no pretenen representar exhaustivament els necessaris en una aplicació real.

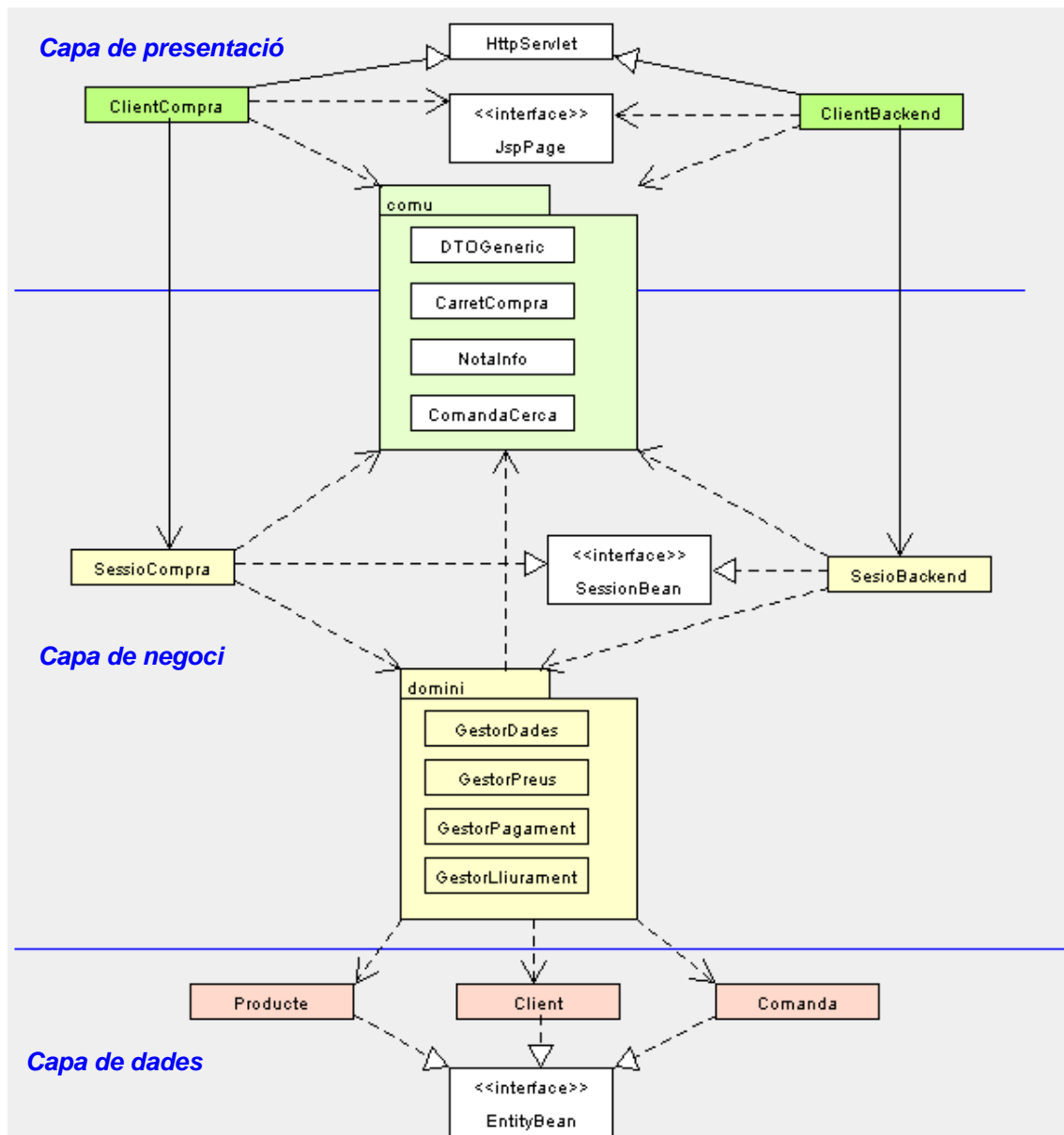
5.3. Característiques de la implementació.

Arquitectura en 3 capes:

Tal com era 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.

Tant les classes de la capa de presentació com les de la capa de negoci usen el paquet *comu*, en què s'hi ha posat les classes usades per a l'intercanvi de dades entre capes, així com les excepcions que es poden generar en la interacció. Les diverses classes de control que s'usen dins del model de negoci s'han posat dins del paquet *domini*, que només és visible per a les classes d'aquesta capa.

Figura 12: separació en 3 capes.



La capa de presentació s'ha implementat amb dos servlets que fan de *FrontController*, un per a la part pública i l'altre per al backend. Cadascun d'aquests servlets centralitza tota la interacció amb l'usuari i usa com a *Dispatcher* diversos fitxers JSP que fan de vistes per als casos d'ús. En aquesta capa s'ha usat l'*API de Servlets i Java Server Pages*, que s'ha mostrat molt assequible i efectiva per a la gestió de sessions http.

La capa de negoci s'ha amagat darrera d'una *Session Bean* per a la part pública i una altra *Session Bean* per al backend. Aquestes dues *Session Bean* implementen mètodes diferents, ja que atenen a casos d'ús ben diferents, però internament usen gairebé les mateixes classes del negoci per a respondre a les peticions. Per darrera de les *SessionBeans* hi ha les classes gestores que s'encarreguen d'executar funcionalitats molt concretes, com ara efectuar el pagament, o bé d'interaccionar amb les dades, com són els gestors de productes, clients i comandes.

La capa de dades s'ha implementat amb tres *Entity Beans*, *Producte*, *Client* i *Comanda*, amb persistència gestionada pel propi bean. S'ha optat per *Bean Managed Persistence* (BMP) enfront de *Container Managed Persistence* (CMP) per diversos motius. Un motiu és que aquestes entitats inclouen alguns atributs que són altres entitats o col·leccions d'entitats, cosa que complicava l'ús de CMP. Un altre motiu és que donada la meua inexperiència en *Abstract*

Persistence Schema i *Container Managed Relationship*, que serien necessaris en CMP, m'ha semblat que BMP era l'opció menys arriscada.

Interacció entre capes:

Per a la interacció entre la capa de presentació i la de negoci s'han definit una colla de classes comunes per a les dues capes. Entre aquestes classes cal destacar ComandaCerca, DTOGeneric i Atribut.

Tal com el seu nom suggereix, la classe ComandaCerca s'ha implementat per tal d'aplicar el patró *Command*. La capa de negoci proporciona a la presentació els objectes ComandaCerca apropiats per a que en la presentació es posi valor als paràmetres oportuns, seguidament s'envia aquest objecte a la capa de negoci a través d'algun mètode de la Session Bean i finalment es retorna l'objecte, que conté el resultat de la petició. D'aquesta manera s'aconsegueix un acoblament prou baix entre presentació i model de negoci.

Al mateix temps s'usa la classe DTOGeneric per tal d'aplicar el patró "*HashMap Data Transfer Object*". DTOGeneric és una subclasse de *HashMap* que conté una còpia dels atributs de l'entitat. D'aquesta manera, la capa de presentació no necessita conèixer exactament quins atributs té cadascuna de les entitats del model de dades. Això permetrà que qualsevol canvi posterior a les capes de negoci o de dades sigui transparent per a la capa de presentació, encara que el canvi impliqui modificar els atributs d'una entitat o bé definir noves subclasses d'aquestes entitats.

Al mateix temps, DTOGeneric també especifica alguns detalls de com cal presentar i filtrar cada atribut en la capa de presentació. Aquesta funcionalitat s'ha implementat usant objectes Atribut com a claus dels valors continguts en el DTOGeneric. La classe Atribut encapsula la informació sobre si l'atribut és editable, cercable, numèric, etc.

Gràcies a l'aplicació dels patrons de disseny esmentats, en el cas concret de la capa de presentació del *backend* s'han pogut implementar els mètodes per a crear, modificar i cercar objectes DTOGeneric, de manera que s'usa exactament el mateix codi tant si l'entitat que es treballa és un Producte com si és un Client o és una Comanda. O sigui, que l'aplicació dels patrons "*Command*" i "*HashMap Data Transfer Object*" ha aportat com a mínim aquesta possibilitat de reús.

En el cas de la capa de presentació de la part pública s'usa el CarretCompra com a *Data Transfer Object* propi del domini. La capa de presentació s'encarrega d'entrar dades en l'objecte CarretCompra, que finalment s'acaba enviant a la capa de negoci, a través del SessionBean, per a que en aquesta darrera capa s'executi la funcionalitat corresponent a validar les dades del carret, crear un objecte comanda, fer el cobrament, etc.

Com a impressió general, la inexperiència en aplicacions distribuïdes s'ha pogut compensar mitjançant l'ús de patrons de disseny de probada eficàcia en problemes similars.

5.4. Proves:

La fase de test de l'aplicació ha consistit en omplir la BD amb algunes dades d'exemple, comprovar l'execució correcta dels casos d'ús especificats a l'anàlisi tant en pantalla com en el seu efecte en la BD.

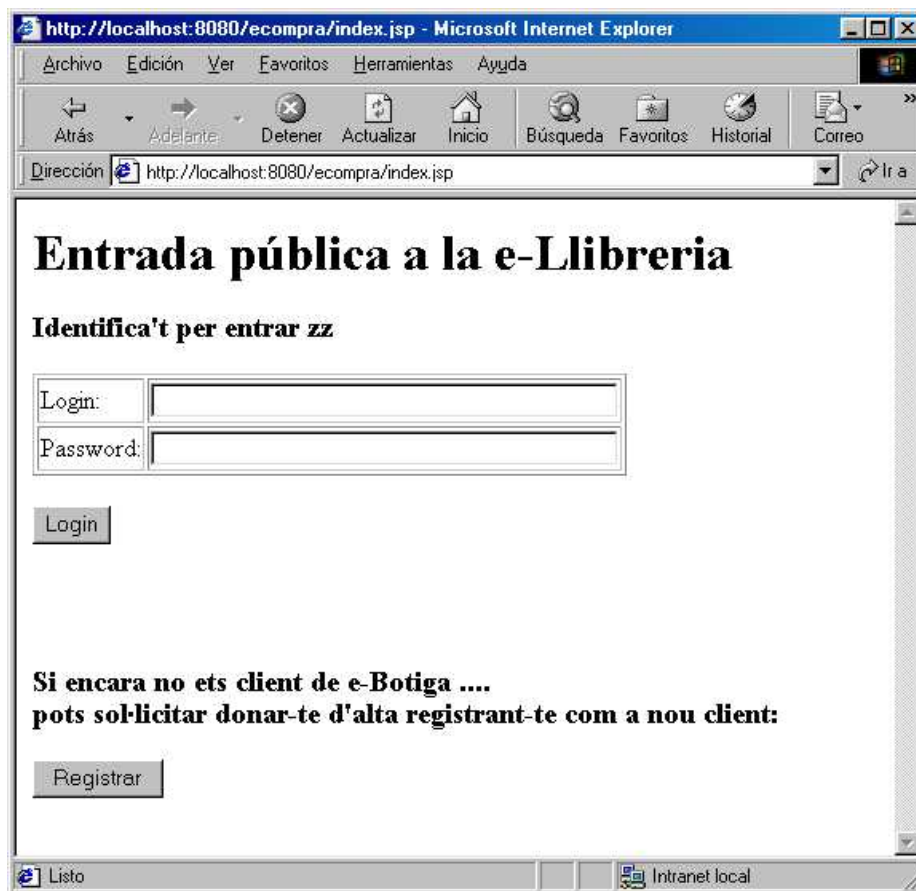
A la part backend:

- ✓ Cerca de productes,
- ✓ Edició del producte,
- ✓ Afegir nota d'informació a producte.
- ✓ Alta de nou producte
- ✓ Cerca de clients.
- ✓ Modificació de client
- ✓ Alta de nou client.
- ✓ Cerca de comanda.

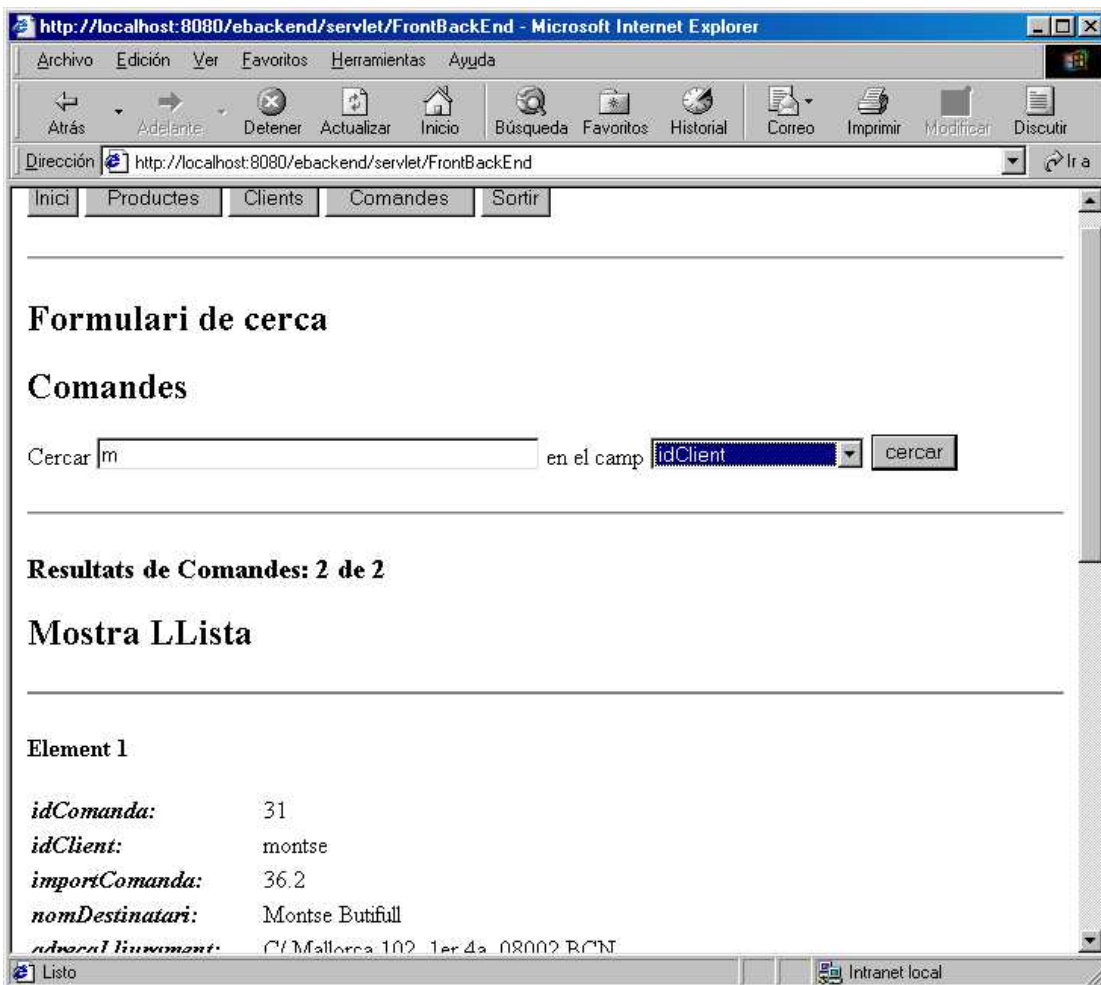
- ✓ Modificació de comanda
- ✓ Sortir

A la part pública de la botiga:

- ✓ Registre de nou client
- ✓ Comprovació de login i password
- ✓ Navegació per la botiga
 - ✓ Cerca de producte
 - ✓ Consulta d'informació adicional sobre el producte
 - ✓ Selecciona producte per afegir al carret
 - ✓ Selecciona producte per afegir a llista d'interès
- ✓ Edita carret de compra
- ✓ Tramita comanda
- ✓ Sortir

5.5. Algunes vistes de l'aplicació obtinguda.**Login a la botiga.**

cerca de comandes en el backend.



(la cerca de productes i clients al backend, i de productes a la botiga té un aspecte molt semblant).

Confirmació de dades abans de tramitar la compra.

Quantitat Ref. Títol preu subtotal

1 <input type="text"/> canvia	11	El senyor dels anells	25.0	25.0
1 <input type="text"/> canvia	14	Manual de XML4	25.0	25.0
2 <input type="text"/> canvia	9	Oxford wordpower4	40.0	80.0

Import total: 130.0 €

Formulari d'edicio

nomDestinatari :

adrecaLliurament :

modalitatLliurament :

Cal que confirmis que estàs d'acord amb el contingut del carret i amb les dades de lliurament per a completar el procés de compra.

Informació del resultat de l'operació.

Inici Cerca Interessants Carret de Compra Tramitar comanda Sortir

El procés de compra s'ha completat satisfactòriament.

Listo Intranet local

6. Conclusions.

- S'ha completat el procés d'anàlisi, disseny i implementació d'una aplicació de comerç electrònic usant J2EE. Concretament, s'han implementat i provat amb èxit gairebé la totalitat de casos d'ús plantejats a l'inici. Malgrat tot, el producte obtingut no és apte per a l'ús en una aplicació real de comerç electrònic sinó que cal situar-lo en la seva funció dins del procés d'aprenentatge.
- S'han posat en pràctica els coneixements previs o adquirits al llarg del TFC sobre la diversitat de metodologies, tecnologies i eines implicades en el desenvolupament d'aplicacions web amb J2EE. Com a metodologia s'ha seguit una simplificació del Procés Unificat de Desenvolupament de Software, en què s'ha usat diversos diagrames UML. S'ha posat en pràctica la integració de tecnologies com J2EE, JSP, servlets, i accés a bases de dades mitjançant JDBC. Com a eines s'ha usat Poseidon, Eclipse, Ant, Tomcat, JBoss i MySQL.
- S'han desenvolupat dos clients web, un per a la part pública i un per a la part backend, dues Session Bean i tres Entity Beans amb persistència gestionada pel bean.
- La implementació obtinguda presenta una separació molt clara de les tres capes. La presentació, implementada amb JSPs i servlets, interactua amb la capa de negoci a través de dues *SessionBeans* que fan de *SessionFaçade*, mentre que la capa de dades està encapsulada dins de tres Entity Beans. Per a aconseguir un menor acoblament entre la presentació i les capes de negoci i dades s'han usat els patrons "*Command*" i "*Data Transfer Object*", que a més han permès reusar codi entre diferents casos d'ús.
- S'ha pogut apreciar el benefici d'usar J2EE amb patrons de disseny. La complexitat d'una aplicació distribuïda pot desbordar fàcilment un programador que toca per primera vegada aquesta tecnologia. D'una banda l'ús de J2EE permet no haver-se de preocupar pel middleware en temps de desenvolupament. Per una altra banda, l'aplicació de patrons de disseny s'ha mostrat molt útil per a compensar la falta d'experiència. El seguiment de patrons de disseny reconeguts permet una corba d'aprenentatge molt més ràpida i fonamentada.

Possibles millores del producte.

- millora de la usabilitat i estètica de la part client
- optimització de la BD
- optimització dels deployment descriptors
- incorporació de mòduls reals per a l'autenticació de l'usuari, per al pagament, etc.

7. Glossari.

Administrador: empleat que realitza tasques administratives a la rebotiga, relacionat sobretot amb el manteniment de les dades de productes, clients i comandes.

Carret de compra: llista amb els productes que el client ha decidit comprar, amb les corresponents quantitats i preus. El carret s'inicialitza al començar la sessió de compra i es destrueix al finalitzar la sessió. Si es completa el procés de pagament, el carret de la compra es converteix en un lliurament.

Client: qualsevol persona que es connecti a la botiga des d'Internet, amb la intenció de comprar.

Dades de tramesa: dades en què s'especifica el nom i adreça de correu del destinatari, juntament amb les opcions d'enviament escollides.

Empleat de magatzem: empleat que "toca físicament" els productes, realitzant tasques de rebuda de stock dels proveïdors i enviament de les compres als clients.

Informació addicional: notes de text, de qualsevol llargada, que fan referència a un producte determinat. Per exemple poden ser retalls de premsa, opinions d'altres lectors, etc.

Llista de productes d'interès: llista que es confecciona el propi usuari amb productes que selecciona del resultat de les cerques i que encara no es decideix a comprar però vol tenir localitzats. Es guarden entre sessions.

Llista de recomanacions: llista de productes que la botiga recomana de manera personalitzada a un client.

Lliurament: selecció de productes que s'envien físicament al client com a resultat del procés de compra. Conté els productes que el client havia posat al carret de la compra, les dades de tramesa i l'estat: {pendent d'enviar, enviat}.

Nom del client: renom que usa el client per a identificar-se. No té perquè ser el que apareixerà com a destinatari del lliurament.

8. Bibliografía.

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.

Eric Armstrong, Jennifer Ball, Stephanie Bodoff, Debbie Carson, Ian Evans, Maydene Fisher, Dale Green, Kim Haase, Eric Jendrock. The J2EETM 1.4 Tutorial. Sun Microsystems, Inc. 2003.

Grady Booch. James Rumbaugh, Ivar Jacobson. El Lenguaje Unificado de Modelado. Addison Wesley. 1999. ISBN: 84-7829-028-1.

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.

Cay S. Horstmann, Gary Cornell. Java2. Características Avanzadas. Prentice Hall. 2002. ISBN: 0-13-092738-4.

Ivar Jacobson, Grady Booch & James Rumbaugh. El proceso unificado de desarrollo de software. Addison-Wesley, 1998. ISBN: 84-7829-036-2.

Budi Kurniawan. Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions. New Riders Publishing. 2002. ISBN : 0-7357-1195-X

Floyd Marinescu. EJB™ Design Patterns. Advanced Patterns, Processes and Idioms. Wiley Computer Publishing. 2002. ISBN: 0-471-20831-0.

Ed Roman, Scott Ambler & Tyler Jewell. Mastering Enterprise JavaBeans™. Second Edition. John Wiley & Sons, Inc. 2002. ISBN: 0-471-41711-4.

Doug Rosenberg, Kendall Scott. Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example. Addison Wesley. 2001. ISBN : 0-201-73039-1.

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

9. Annexos.

9.1. Instruccions d'instal·lació.

1. Desplegar els components.

Cal desplegar en el servidor d'aplicacions els fitxers:

Producte.jar	EntityBean
Client.jar	EntityBean
Comanda.jar	EntityBean
SessioCompra.jar	SessionBean per a la part pública.
SessioBackend.jar	SessionBean per a la part privada.
ecompra.jar	Client per a la part pública.
ebackend.jar	Client per a la part privada.

2. Crear la base de dades.

- 2.1. Crear l'usuari i password que s'usa en el programa.

Per a accedir a la base de dades l'aplicació usa el nom d'usuari "Mortadelo", amb password "". Per tal de poder-lo usar en una instal·lació diferent recomano crear l'usuari "Mortadelo" amb password = "" en la base de dades.

- 2.2. Crear la base de dades.

Adjunto en el directori **/fitxers/BD** el fitxer **creaBD.txt** en què hi ha les sentències SQL de creació de la base de dades.

En el cas de MySQL, cal copiar el fitxer creaBD.txt en el directori /bin de mysql, anar a aquest directori des d'una finestra de sistema i executar la comanda:

```
mysql -h nomhost -u nomusuari -p password < creaBD.txt
```

(a on cal reemplaçar nomhost, nomusuari i password pels valors oportuns).

- 2.3. Població de la base de dades.

Adjunto en el directori **/fitxers/BD** el fitxer **insertaBD.txt** en què hi ha les sentències SQL per a la inserció d'unes poques entitats a la BD.

L'entrada d'aquestes dades no és imprescindible per a l'execució de l'aplicació. Ara bé, per al funcionament de la part pública sí que és necessari haver creat alguns productes, cosa que també es pot fer des del backend.

3. Execució de la part privada.

Des de la finestra del navegador cal anar a:

<http://localhost:8080/ebackend>

Per a facilitar les proves, en aquesta versió de l'aplicació l'he deixat que accepti qualsevol login i password en la part de backend.

4. Execució de la part pública.

Des de la finestra del navegador cal anar a:

<http://localhost:8080/ecompra>

Caldrà identificar-se amb un login i password que s'hagi entrat a la base de dades, o si no, donar-se d'alta com a nou usuari.

9.2. Material que s'adjunta.

En el directori **fitxers** s'inclouen els següents subdirectoris:

fitxers/BD: conté fitxers per a crear i poblar la base de dades.

creaBD.txt conté les sentències SQL per a la creació de la base de dades que usa l'aplicació.

insertaBD.txt conté les sentències SQL per a poblar la BD amb algunes dades d'exemple per a fer proves.

fitxers/deploy: conté els components a desplegar

Producte.jar	EntityBean
Client.jar	EntityBean
Comanda.jar	EntityBean
SessioCompra.jar	SessionBean per a la part pública.
SessioBackend.jar	SessionBean per a la part privada.
ecompra.jar	Client per a la part pública.
ebackend.jar	Client per a la part privada.

fitxers/src: conté els directoris que s'han usat en la implementació d'aquesta aplicació amb Eclipse.

Els noms dels directoris corresponen als noms dels projectes que s'han treballat des d'Eclipse.

ebotiga conté els paquets:

comu classes comuns entre presentació i negoci.

domini classes de control del negoci

ejb.producte fitxers corresponents a l'entity bean Producte

ejb.client fitxers corresponents a l'entity bean Client

ejb.comanda fitxers corresponents a l'entity bean Comanda

ejb.sessioCompra fitxers corresponents a la session bean sessioCompra

ejb.sessioBackend fitxers corresponents a la session bean sessioBackend

clientBackend conté els fitxers corresponents al client de la part privada.

clientCompra conté els fitxers corresponents al client de la part pública.

fitxers/doc: conté la documentació del projecte generada amb javadoc.

fitxers/img: conté alguns pantallassos obtinguts al realitzar les proves.