

Desenvolupament J2EE

Francisco Laboria Vallés
ETIG

Consultor: Javier Ferró Garcia

18/06/2007

Agraïments

És difícil personalitzar moltes de les fonts d'informació disponibles en Internet. Si hagués d'escollir una seria sens dubte www.programacion.net per la seva estructura, amplitud de continguts útils i per tenir-los traduïts al castellà.

També agrair als autors de la Pràctica Tutoritzada de J2EE per la seva bona feina que m'ha ajudat a fer mes entenedor aquest mon tan complex.

Gràcies també al consultor per no deixar cap missatge sense resposta.

I sobretot, gràcies a la meva dona per haver suportat les meves absències durant el desenvolupament d'aquest treball final i durant tota la carrera.

Conceptes de desenvolupament J2EE

La definició de Java 2 Enterprise Edition més estesa és la que diu que:
“J2EE és un conjunt d’especificacions de desenvolupament d’aplicacions dissenyada per Sun Microsystems. Aquestes especificacions son regulades per les empreses membres del Comitè Executiu JSP.”

Al ser un conjunt d’especificacions permet la creació de productes als fabricats que desitgin adaptar-se a dites especificacions, això ofereix un ventall ampli de possibilitats que té com a característiques la disponibilitat de gran nombre de productes de desenvolupament i suport, això es converteix en avantatges i inconvenients aprofitats pels defensors i detractors del J2EE que citarem més endavant.

Altres característiques de J2EE son les possibilitats de disseny modular, l’existència d’elements integradors entre els components que automatitzen les funcionalitats, redueixen el codi i les característiques pròpies dels llenguatges orientats a objectes com son el polimorfisme i l’herència que faciliten la reutilització.

Els arguments principals dels defensors de J2EE respecte altres plataformes son la seva experiència i maduresa, els casos d’èxit, la portabilitat, el codi lliure i el no estar monopolitzat per una única empresa. Els detractors veuen com a inconvenients la poca modernitat respecte altres plataformes com Visual Studio .NET, la complexitat per integrar els components i ser únicament compatible pel llenguatge Java.

El que sembla estar prou clar és que J2EE no pot faltar mai, encara que sigui només com a alternativa, en cap projecte seriós de desenvolupament d’aplicacions.

Índex

Desenvolupament J2EE	1
Agraïments	2
Conceptes de desenvolupament J2EE	3
Índex.....	4
Índex de figures	5
CAPITOL 1 - Especificacions	6
Introducció.....	6
Justificació.....	8
Objectius TFC	9
Enfocament i mètode seguit.....	10
Planificació del projecte.....	11
Productes obtinguts.....	13
Desenvolupament de l'aplicació.	14
CAPITOL 2 – Entorn J2EE	15
Introducció.....	15
Programari necessari	15
Paràmetres de configuració.....	15
CAPITOL 3 - Disseny	17
Introducció.....	17
Model del concepte del negoci	19
Model de casos d'ús.....	20
Diagrames de classes de disseny	25
Diagrames de components.....	27
Diagrames de desplegament.....	28
Diagrames de seqüència.....	29
Interfície gràfica.....	30
Taules de la base de dades	34
CAPITOL 4 - Implementació	36
Introducció.....	36
Desplegament	36
Implementació dels EJBs d'entitat.....	37
Implementació del EJB de sessió.....	41
Implementació del Servlet	43
Implementació dels JSPs	46
CAPITOL 5: Manual d'instal·lació de la botiga virtual	47
Introducció.....	47
Instal·lació	47
Execució.....	47
CAPITOL 6 – Valoració Econòmica	48
Introducció.....	48
Valoració econòmica, el termini de lliurament i detalls planificació d'aplicacions J2EE.....	48
Supervisió i valoració de desenvolupadors d'aplicacions J2EE	52

CAPITOL 7 - Conclusions	53
Disseny	55
GLOSSARI.....	57
BIBLIOGRAFIA	58

Índex de figures

Figura 1: Arquitectura tres capes	16
Figura 2: Patró MVC-2	17
Figura 3: Classes de negoci	18
Figura 4: Casos d'ús	19
Figura 5: EJB de sessió	24
Figura 6: Classes de disseny	25
Figura 7: Diagrama de components	26
Figura 8: Diagrama de desplegament	28
Figura 9: Diagrama de seqüència – Efectuar comanda	29
Figura 10: Model entitat-relació	34
Figura 11: Esquema Servlet	44
Figura 12: Gràfica relació experiència-cost	50
Figura 13: Quadre comparatiu J2EE vs VisualStudio .NET	54
Figura 14: Aplicació Struts.....	55

CAPITOL 1 - Especificacions

Introducció

Desenvolupament aplicació J2EE

El ritme d'evolució de les tecnologies de la informació és realment imparable. Els professionals informàtics no tenen altre remei que especialitzar-se en algun dels camins que ofereix aquesta professió perquè és impensable voler abraçar un coneixement adequat per tots els desenvolupaments, programari, aplicacions, llenguatges,... existents en el mercat.

L'opció d'especialitzar-se pot ser de vegades una opció dolenta si les aspiracions professionals són més altes que les de ser alguna cosa més que pica-tecles, programador o inclús supervisor, almenys en el món industrial que és el que jo conec.

Excepte en contades ocasions, els professionals informàtics tenen devaluada la seva feina en el món industrial, perquè no són un servei directe a la producció sinó un servei de altres serveis com projectes industrials, manteniment d'instal·lacions o com compres, finances o recursos humans.

Avui en dia poques empreses poden ignorar els avantatges de tenir implantada una bona arquitectura informàtica i encara així molts dels càrrecs de gerència, directius, caps de departament i de secció inclús d'entorn informàtic de les empreses les formen membres amb especialitats econòmiques o financeres o pròpies del tipus de producció.

Les grans decisions informàtiques en empreses del món industrial són preses de forma top-down i quan arriben a l'usuari final, que són qui les pateixen, semblen ser preses amb criteris econòmics i sense un assessorament informàtic adient. Les aportacions informàtiques preses des de l'usuari requereixen justificacions extenses i acurades per ser implementades.

Un professional informàtic en el món industrial que conviu diàriament amb els problemes dels usuaris, necessita complir dos requeriments principals: conèixer els processos interns (know-how) i conèixer les possibilitats que ofereix el mercat.

Pel primer dels requeriments es necessita experiència i formació interna dins de l'empresa, i pel segon alguna cosa més que mirar catàlegs en Internet o revistes d'informàtica.

Existeixen àrees millors per obtenir bons criteris per l'elecció de productes i per la planificació de projectes informàtics, però soc de l'opinió que entrar en el detall, el desenvolupament d'una aplicació J2EE pot aportar a un professional informàtic

bons coneixements que li poden ser necessaris. Aquests coneixements estan descrits en l'apartat d'objectius.

El producte desenvolupat pel TFC no tracta de ser més que una aproximació a la realitat. És per això que des d'un inici, la idea ha estat ampliar una aplicació existent per entrar al detall del desenvolupament i obtenir els coneixements necessaris esmentats anteriorment. És clar, que res en aquesta vida és gratuït i per això es requereix la meva aportació per a la millora del producte que correspon a una botiga virtual.

Botiga virtual

El projecte TFC Botiga Virtual dissenyarà i implementarà una aplicació per la venda online de llibres mitjançant la plataforma J2EE.

Disposarà d'un catàleg de llibres, àrea pública, on hi accedeixen els clients i poden visualitzar la informació dels diferents llibres classificats per categories. Es podran realitzar comandes i la compra es gestionarà mitjançant el carret de la compra.

Els administradors podran efectuar les accions de manteniment de l'aplicació amb les opcions de gestionar categories, llibres, clients i comandes.

Justificació

No cal dir la repercussió que avui en dia té Internet, podria buscar dades del nombre d'usuaris actuals, el nombre d'empreses, planes web... però quan aquesta memòria sigui llegida, el número ja estarà desfasat.

Existeix una gran extensió de URLs desenvolupades amb J2EE. Per treballs anteriors, tinc la certesa del gran impacte que té aquesta plataforma en el sector que més requereix de les tecnologies de la informació per ser competitiu, el sector bancari.

Jo soc membre de una empresa multinacional del món industrial. La implantació d'aplicacions J2EE és mínima en aquest sector.

Una de les meves primeres aspiracions pel TFC va ser el desenvolupament d'una petita aplicació amb Java que fos útil per alguns dels processos de la meva empresa. Un cop desenvolupada podria tenir una base per justificar nous projectes de mica en mica.

El primer impediment va ser la propietat intel·lectual del TFC. El segon, la poca informació existent respecte a la utilització de programari lliure o lliure excepte per usos comercials, això va provocar que em tirés enrera amb la meva idea perquè intentar obrir un procés d'aquestos en una empresa extremadament burocratitzada requereix temps i recursos dels que no dispo. També és cert que els meus coneixements en la matèria no estaven prou madurs.

La situació inicial de coneixements per un projecte J2EE m'ha sorprès, no era conscient de la poca idea que tenia. La Pràctica Tutoritzada em va fer començar a veure llum i vaig proposar-me ampliar les seves funcionalitats. D'aquesta manera començaria a adquirir coneixements sobre una arquitectura sòlida i acceptada.

Sembla raonable des d'un punt de vista acadèmic que la correcta ampliació de funcionalitats d'un programari existent, tingui suficient pes per aprovar el TFC i d'aquesta manera obtenir suficients fonaments per tirar endavant la meva idea inicial. El desenvolupament d'alguna petita aplicació per automatitzar alguns dels processos interns de la meva empresa, com he dit abans, requerirà un llarg procés burocràtic.

Objectius TFC

Tal com s'ha esmentat en la Introducció, els objectius principals son d'àmbit personal i es poden resumir en els següents punts:

- Obtenció d'un criteri suficient per considerar J2EE com la plataforma de desenvolupament adequada per un projecte determinat.
- Coneixement inicial de quin és el programari necessari, els seus requeriments i de l'entorn on ha de funcionar
- Punts de suport suficients per a la planificació d'un projecte de desenvolupament en referència al terme de lliurament, valoració econòmica i altres detalls de la planificació.
- Millors criteris per escollir els perfils de desenvolupadors adients.
- Coneixement de detalls per la supervisió i valoració de programadors.
- Aportació crítica de decisions de disseny.
- Aportació crítica de mètodes de desenvolupament.
- Imatge mental o visió suficient per tenir la capacitat de justificar o vendre el producte final.

Tot això mitjançant l'enteniment, ampliant i millorant si és possible l'aplicació existent de botiga virtual.

Botiga virtual

El projecte TFC Botiga Virtual té com objectiu tenir al final del semestre una aplicació plenament funcional, aprofitant el disseny i implementació de la Pràctica Tutoritzada per l'àrea pública de l'aplicació per dissenyar i implementar l'àrea privada de la mateixa.

Paral·lelament es tractarà d'aportar millores a la arquitectura i funcionalitats de l'aplicació.

Enfocament i mètode seguit

S'ha dividit el projecte en les etapes típiques del desenvolupament d'una aplicació:

- Configuració de l'entorn
- Especificació
- Disseny
- Prototip
- Implementació
- Tractament d'excepcions
- Test
- Estudi qualitat

Per raons de temps només s'han elaborat les cinc primeres etapes i s'han adaptat a les tasques específiques d'un desenvolupament J2EE amb un patró MVC-2.

- Configuració de l'entorn
 - o Programari operatiu
 - o Test aplicacions de prova
- Especificació
 - o Descripció del projecte
 - o Tasques
 - o Planificació
- Disseny
 - o Model de negoci
 - o Casos d'ús
 - o Diagrames CASE
- Prototip
 - o Interfície gràfica
- Implementació
 - o Base de dades
 - o EJB's d'entitat
 - o EJB de sessió
 - o Servlet
 - o JSP's

Paral·lelament es guardarà el material lliurat en les PACs per la realització d'aquesta memòria i de la presentació final.

Planificació del projecte

Les etapes esmentades en l'apartat anterior es poden concretar en les tasques següents:

Tasca 0: (<10/03/2007)

Programari instal·lat i funcionant correctament.
Execució dels exemples amb el JBoss.

Tasca 1: Definició del model del concepte del negoci.
(05/03/2007 – 10/03/2007)

Tasca 2: Definició del model de casos d'ús
(10/03/2007 – 12/03/2007)

Tasca 3: Creació de diagrames de classes de disseny
(12/03/2007 – 18/03/2007)

Tasca 4: Creació dels diagrames de components
(19/03/2007 – 21/03/2007)

Tasca 5: Diagrama de desplegament
(22/03/2007 – 24/03/2007)

Tasca 6: Definició de la interfície gràfica
(25/03/2007 – 13/04/2007)

Tasca 7: Creació de les taules a la base de dades
(13/04/2007 – 16/04/2007)

Tasca 8: Disseny, implementació, desplegament i prova dels EJBs d'entitat
(17/04/2007 – 5/05/2007)

Tasca 9: Disseny, implementació, desplegament i prova de l'EJB de sessió
(06/03/2007 – 21/05/2007)

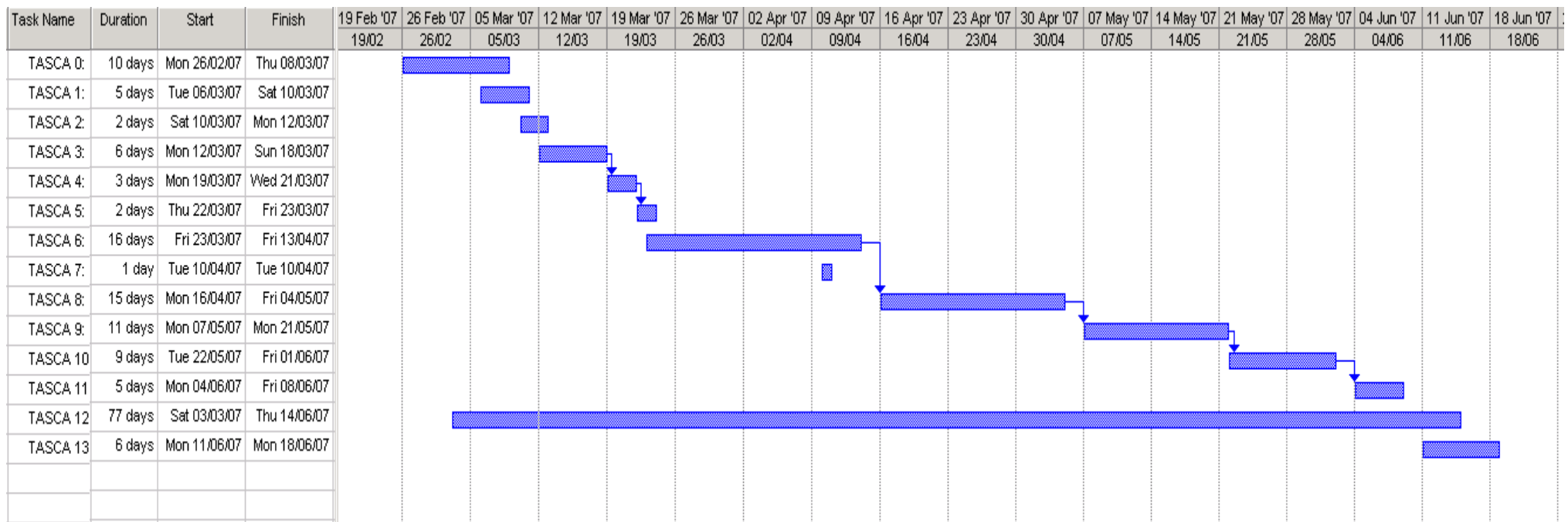
Tasca 10: Creació de les JSPs que faran de clients
(22/05/2007 – 02/06/2007)

Tasca 11: Introducció d'un servlet a la capa de presentació
(03/06/2007 – 10/06/2007)

Tasca 12: Memòria (08/03/2007 – 15/06/2007)

Tasca 13: Presentació (15/06/2007 – 18/06/2007)

Diagrama de Gantt



Productes obtinguts

La consecució final d'un desenvolupament en J2EE és l'obtenció d'una aplicació plenament funcional.

En aquest cas es tracta d'una botiga virtual per la venda de llibres online.

Per restriccions de temps aquesta aplicació no te automatitzats tots els mètodes possibles, però si amplia les funcionalitats existents i els seus mètodes son plenament operatius.

La botiga virtual implementa una arquitectura MVC-2. Incorpora els mètodes de negoci i persistència necessàries per l'accés i manteniment a les dades de categories, productes, clients i comandes. També ofereix la interfície gràfica per les operacions dels administradors i dels clients.

Aquest producte s'anirà explicant en els apartats següents d'aquesta memòria. Paral·lelament es lliura l'aplicació i el manual d'instal·lació.

Desenvolupament de l'aplicació.

Els següents quatre apartats mostren la feina realitzada al llarg del quadrimestre.

Degut a la poca experiència en la plataforma J2EE, s'han comès alguns errors de disseny plasmat en les PACs lliurades i que es rectifiquen en aquesta memòria.

Capítol 2: Configuració de l'entorn

Breu explicació del programari i els paràmetres de configuració establerts per l'òptim funcionament de l'aplicació.

Capítol 3: Disseny de la botiga virtual

En aquest capítol es mostren i justifiquen les decisions de disseny de la botiga virtual.

Conté els casos d'ús, la lògica del negoci, els diagrames de classes, seqüència, component i desplegament, i la maqueta de la interfície gràfica.

Capítol 4: Implementació de la botiga virtual

Breu explicació del mètodes d'implementació dels EJBs locals i remots, del EJB de sessió, del servlet i dels JSPs.

Capítol 5: Manual d'instal·lació de la botiga virtual

Manual d'instal·lació de la botiga virtual.

CAPITOL 2 – Entorn J2EE

Introducció

Degut a la poca trivialitat de tenir l'entorn preparat per l'execució d'una aplicació J2EE, en aquest apartat s'explica el programari utilitzat i els paràmetres de configuració requerits pel correcte funcionament de la botiga virtual.

Programari necessari

JDK 1.5.0_11
Servidor d'aplicacions JBoss 4.0.5.GA
Base de dades MySQL 5.0.27
MySQL GUI Tools Bundle for 5.0
Connector Java JDBC de MySQL 5.0.4
ANT 1.7.0

Paràmetres de configuració

Variables d'entorn:

Crear les següents variables:

JAVA_HOME			
[Directori instal·lació JDK]	~	...	jdk1.5.0_11
JBOSS_HOME			
[Directori instal·lació JBoss]	~	...	jboss4.0.5.GA
ANT_HOME			
[Directori instal·lació Ant]	~	...	apache-ant-1.7.0

Afegir els següents valors, separats per ; als valors existents de les variables:

PATH:
; %JAVA_HOME%\bin
; %JBOSS_HOME%\bin
; %ANT_HOME%\bin

CLASSPATH:
;
; %JBOSS_HOME%\client\jboss-all-client.jar
; %JBOSS_HOME%\client\log4j.jar

```
; %JAVA_HOME%\client\jboss-all-client.jar  
; [ Directori instal·lació connector Java JDBC de MySQL]\mysql-  
connector-java-5.0.4-bin.jar ~ c:\mysql-connector-java-  
5.0.4mysql-connector-java-5.0.4-bin.jar
```

MySQL:

Tenir un usuari amb permisos per connectar-se a la BD i permisos adients per accedir al Schemata on penjaran les taules de la BD.

Configurar SQL per treballar amb InnoDB.

Copiar el fitxer mysql-connector-java-5.0.4-bin.jar que està en el directori d'instal·lació del connector Java JDBC de MySQL a %JBOSS_HOME%\server\default\lib i a %JBOSS_HOME%\lib

Fitxers de configuració:

Copiar els següents fitxers continguts a la carpeta ...fitx del arxiu EPCSD.zip lliurat i reemplaçar-los per els existents.

Copiar ...fitx\ear-deployer.xml a

%JBOSS_HOME%\server\default\deploy\

Copiar ...fitx\jboss-service.xml a %JBOSS_HOME%\server\default\conf\

Copiar ...fitx\mysql-ds.xml a %JBOSS_HOME%\server\default\deploy\

Copiar ...fitx\login-config.xml a %JBOSS_HOME%\server\default\conf\

* (Dins d'aquest fitxer s'han de substituir els valors [nom d'usuari = MONOMAIL] i [paraula de pas = monomail] corresponents a l'usuari amb permisos per efectuar les operacions en la BD)

Copiar ...fitx\standardjbosscomp-jdbc.xml a

%JBOSS_HOME%\server\default\conf\

Esborrar el fitxer %JBOSS_HOME%\server\default\deploy\gangster-cmp2.jar si existeix.

CAPITOL 3 - Disseny

Introducció

En aquest capítol s'assenten les bases de disseny de l'aplicació J2EE botiga virtual.

L'aplicació botiga virtual tindrà una arquitectura en tres capes:

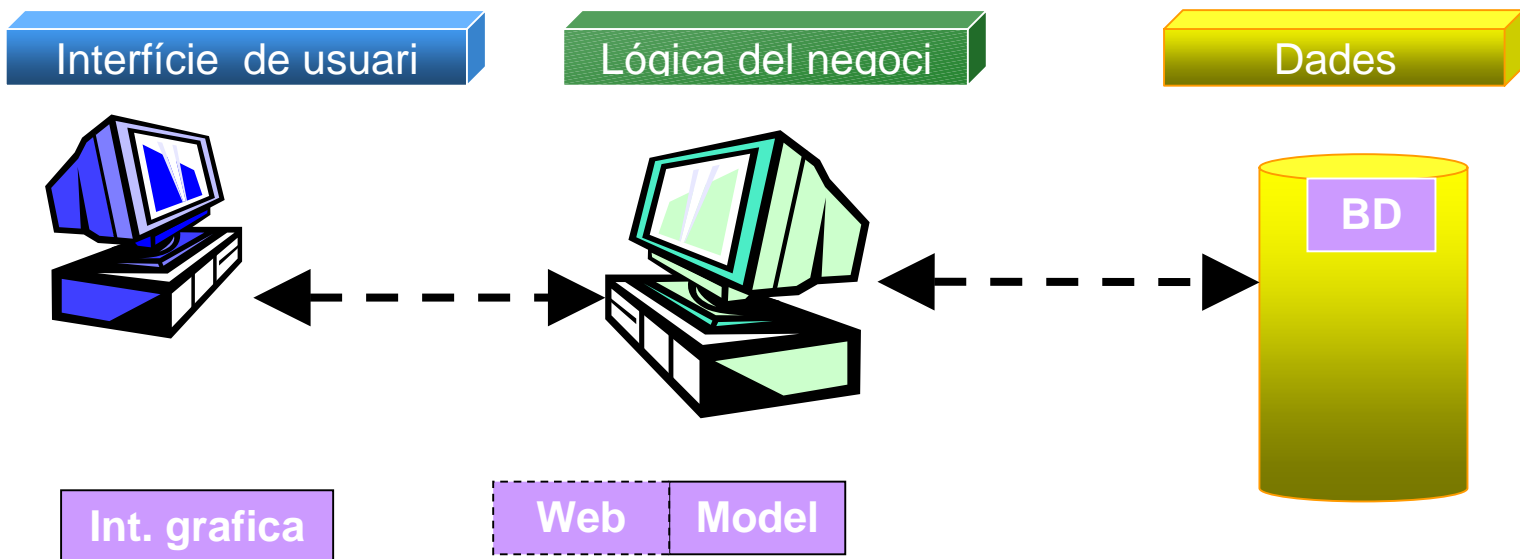


Figura 1: Arquitectura tres capes

Justificació:

Una arquitectura en tres capes ofereix una sèrie d'avantatges:

- Manté separat la lògica del negoci dels accessos dels usuaris, minimitzant l'estructura, mètodes i codi que és visible pels usuaris. D'aquesta manera fomenta la seguretat de l'aplicació.
- Una bona arquitectura amb un acoblament entre capes mínim, ofereix avantatges de portabilitat, permet transportar les capes per separat i efectuar el mínim nombre de canvis al passar d'una a altra plataforma.
- Rendibilitat de la reusabilitat de l'aplicació. Permet l'aprofitament de capes senceres per noves aplicacions o per una re-enginyeria de l'existent.
- Optimització del manteniment de cara a possibles canvis, actualitzacions o ampliacions de l'aplicació.

S'aplicarà un patró arquitectònic MVC-2 per ser àmpliament acceptat i tenir una sèrie d'avantatges respecte altres patrons:

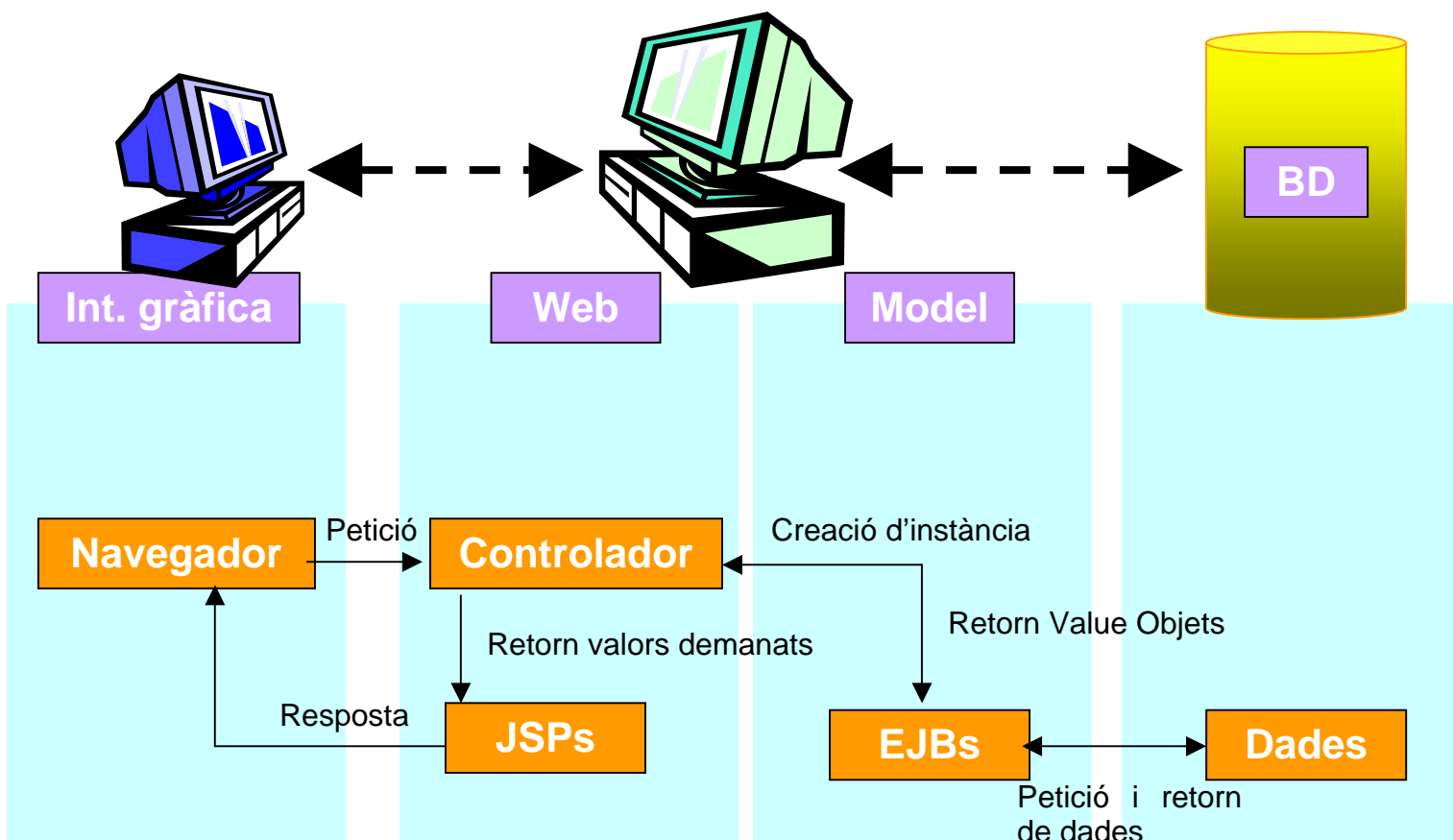


Figura 2: Patró MVC-2

Justificació:

Un patró arquitectònic MVC-2 permet gestionar i filtrar totes les peticions d'usuari, les comprovacions de seguretat i els logs de les accions realitzades en un únic punt de la capa vista de l'aplicació. Obtenim els següents beneficis:

- Desacobla la capa vista de la capa presentació augmentant la seguretat, fent visible a l'usuari el mínim codi possible.
- Augmenta la facilitat de manteniment, reusabilitat i portabilitat al tenir els mètodes de la capa presentació en un únic punt centralitzat.
- Millora el rendiment al reduir el nombre d'invocacions a mètodes de l'aplicació.

Model del concepte del negoci

Un cop introduït l'arquitectura de l'aplicació entrarem al detall de la botiga virtual. A partir de l'especificació de la Practica Tutoritzada farem les modificacions per afegir les classes de negoci requerides de l'àrea privada de l'aplicació:

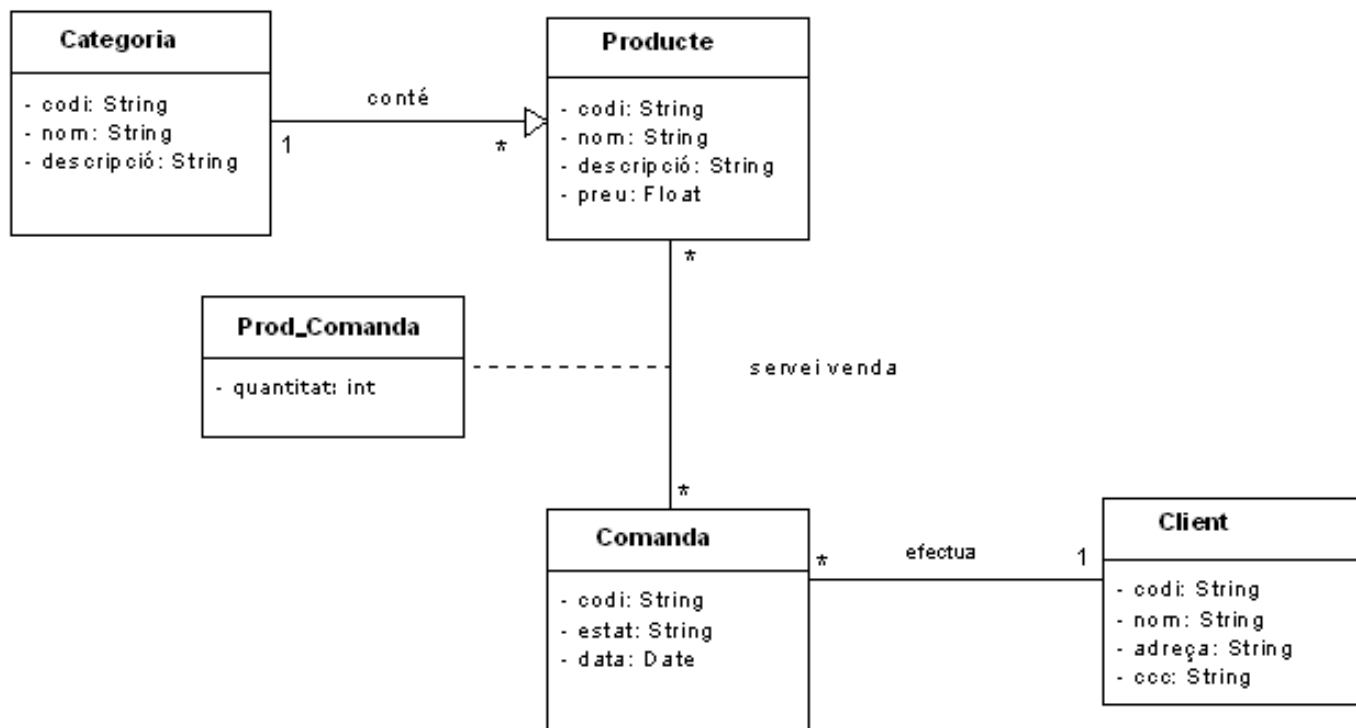


Figura 3: Classes de negoci

Model de casos d'ús

Els casos d'ús marcats en vermell no està previst que siguin desenvolupats en aquest aplicatiu.

Actors:

Al sistema poden actuar els administradors o gestors i els clients. Per futures versions es podrien incloure altres tipus d'actors com els "auxiliars" o "gestors" amb permisos diferents que s'encarregarien del manteniment i gestió dels productes, clients i comandes.

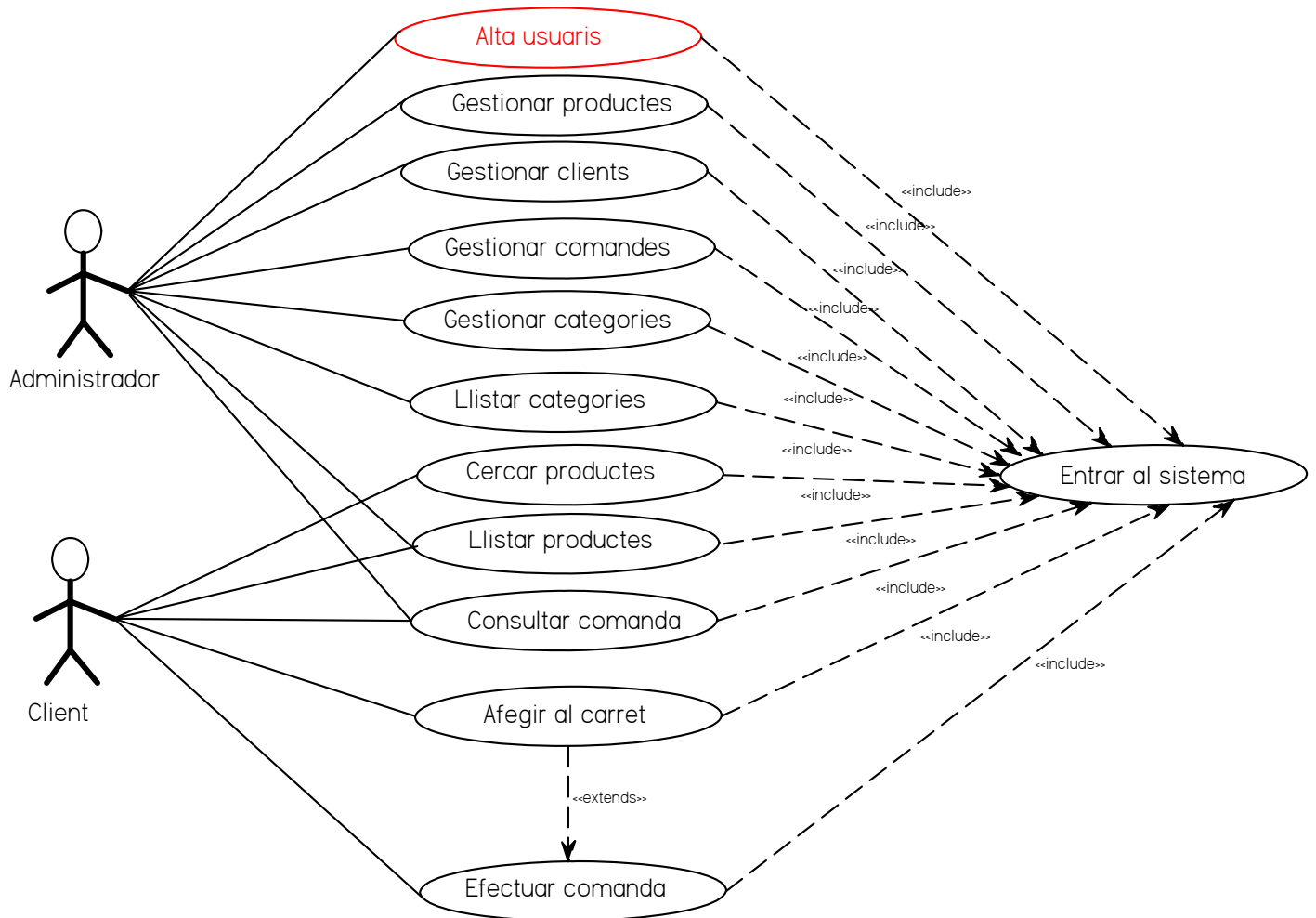


Figura 4: Casos d'ús

Casos d'us:

En aquesta versió de l'aplicació es parlarà d'administradors o gestors indistintament, essent el mateix tipus d'usuari.

Alta Usuaris

No està previst que un usuari es pugui donar d'alta al sistema com a client. Seran els administradors qui efectuaran aquesta operació fora d'aquest aplicatiu.

Entrar al sistema

Permet als usuaris de l'aplicatiu (administradors i clients en aquesta versió) l'entrada al sistema.

Llistar Categories

Permet als usuaris de l'aplicatiu llistar les categories que hi ha al catàleg de productes..

Llistar Productes

Permet als usuaris de l'aplicatiu llistar els productes d'una categoria determinada. *No es té previst diferents tipus de llistat de productes per aquesta versió de l'aplicatiu.*

Cercar Producte

Permet als usuaris trobar productes pel nom.

Gestionar Categories

Permet als administradors o gestors fer les tasques de manteniment de les categories (CRUD).

Gestionar Productes

Permet als administradors fer les tasques de manteniment dels productes (CRUD). *No s'implementarà el control d'stocks de l'aplicació. Aquesta tasca podria ser desenvolupada per Gestors o Auxiliars en futurs desenvolupaments de l'aplicatiu.*

Gestionar Clients

Permet als administradors o gestors fer les tasques de manteniment dels clients (CRUD).

Gestionar Comandes

Permet als administradors canviar els estats de les comandes.

Afegir Productes al Carret

Permet als clients afegir un producte al carret de compra. Els productes del catàleg tenen stock o està previst que en tinguin. Els administradors s'encarregaran, fora de l'aplicatiu, de gestionar els conflictes que es produeixen per la "rotura de stocks".

Efectuar Comanda

Permet als clients validar la seva selecció de compra i efectuar la comanda.

Consultar Comanda

Permet als clients consultar l'estat de les seves comandes. Als gestors els permetrà consultar totes les comandes existents al sistema.

Especificació textual

Es descriuen els cas d'ús més interessant de l'aplicatiu, els altres corresponents a alta, baixa, modificació i visualització que no es descriuen per ser prou típics.

Cas d'ús Login

Resum de la funcionalitat: Permet a un usuari identificar-se com a usuari de vàlid de l'aplicació

Actors: Client, Administrador

Casos d'us relacionats: Tots

Precondició: L'usuari està donant d'alta com a usuari de l'aplicació i no s'ha identificat prèviament.

Postcondició: L'usuari està identificat al sistema.

Passos:

Flux d'events principal

1. L'usuari introdueix les dades d'identificació:
 - a. Nom d'usuari
 - b. Mot de pas
2. El sistema valida que el nom d'usuari i el mot de pas siguin correctes
3. El sistema redirigeix a l'usuari a la pantalla llistat categories.

Flux d'events alternatiu:

3. .b. Les dades són incorrectes i el sistema mostra a l'usuari una pantalla d'error

Cas d'ús Afegir al Carret

Resum de la funcionalitat: El client selecciona un producte que el sistema emmagatzema

Actors: Client

Casos d'us relacionats: Entrar al sistema, afegir al carret.

Precondició: El client està identificat al sistema.

Postcondició: El sistema conté una llista de productes.

Passos:

Flux d'events principal

1. El sistema ofereix un llistat dels productes disponibles.
2. El client tria l'opció "afegir al carret".
3. El sistema guarda el codi del producte seleccionat pel client i es torna al pas 1 tants cops com el client vulgui.

Flux d'events alternatiu

2. b. En cas que el producte ja hagi estat seleccionat pel client el sistema no fa res.

Cas d'ús Efectuar Comanda

Resum de la funcionalitat: El client valida els productes que ha seleccionat per efectuar la comanda

Actors: Client

Casos d'ús relacionats: Entrar al sistema, afegir al carret

Precondició: El client està identificat al sistema i té un o més productes afegits al carret.

Postcondició: Comanda registrada.

Passos:

Flux d'events principal

1. El sistema ofereix les dades de la comanda:
 - El codi client
 - Data comanda
 - Llista dels productes afegits pel client
 - Quantitat de cada producte
2. El client modifica els productes i quantitats si ho creu adient i tria Acceptar.
3. El sistema emmagatzema la comanda, la posa en estat pendent i es redirigeix a la pantalla llistatcomandes.

Flux d'events alternatiu:

2. b. L'usuari tria l'opció Cancel·lar.
3. El sistema esborra els productes de la comanda i es redirigeix a la pantalla llistatproductes

Diagrames de classes de disseny

Seguint amb el patró arquitectònic MCV-2 es detallen a continuació les classes dels EJBs d'entitat i de sessió.

L'ús de CMP implica la no existència de relació entre els entity beans.

L'ús de CMP es justifica per l'estalvi de codi escrit per la connexió JDBC a la base de dades.

Punt oberts a valorar en futurs desenvolupaments:

1. Implicacions de la falta de transparència per l'ús de CMP en la connexió a la BD.
2. Càlculs d'eficiència temporal i espacial en lectura i escriptura segons el tipus de contenidor

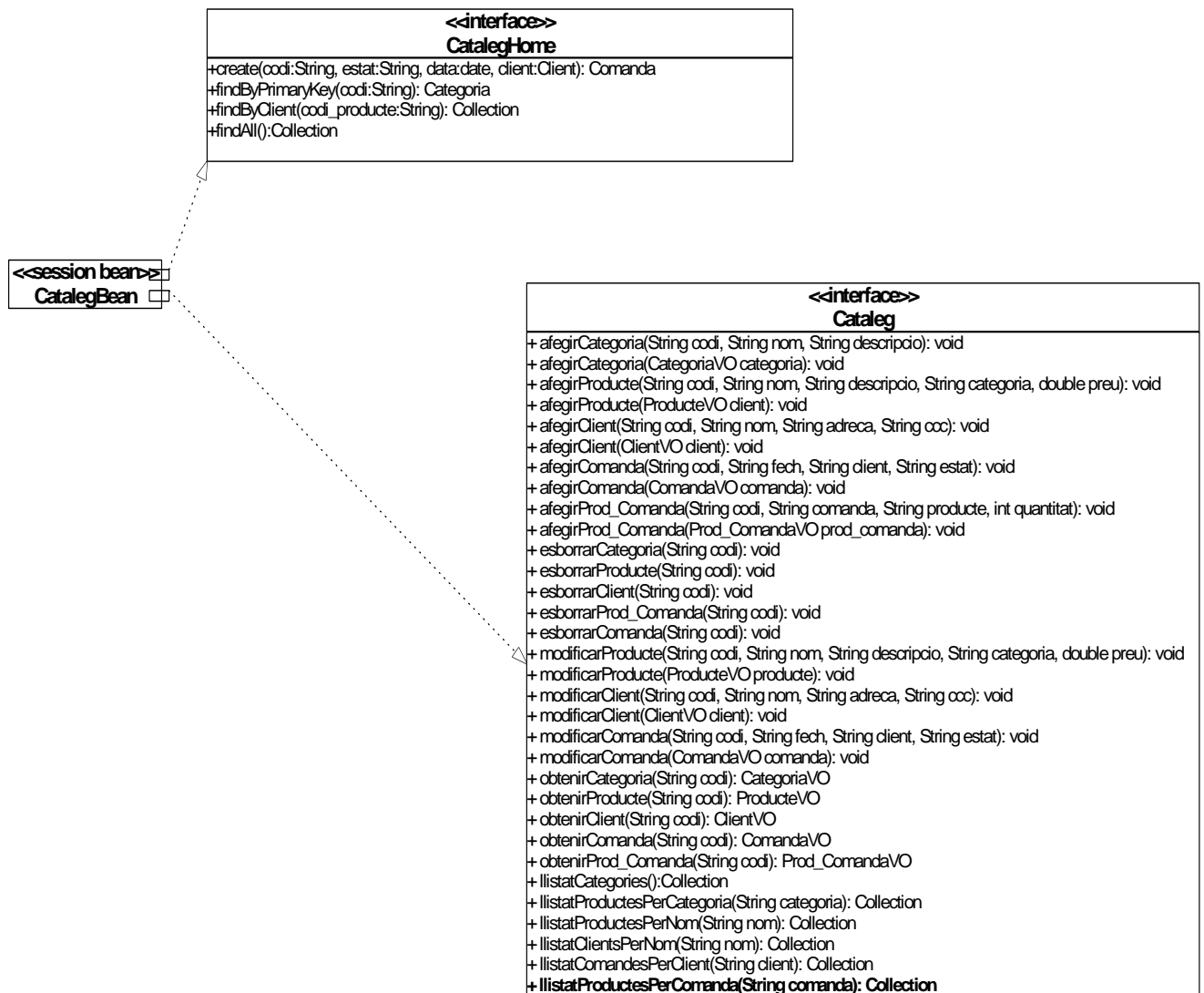


Figura 5: EJB de sessió

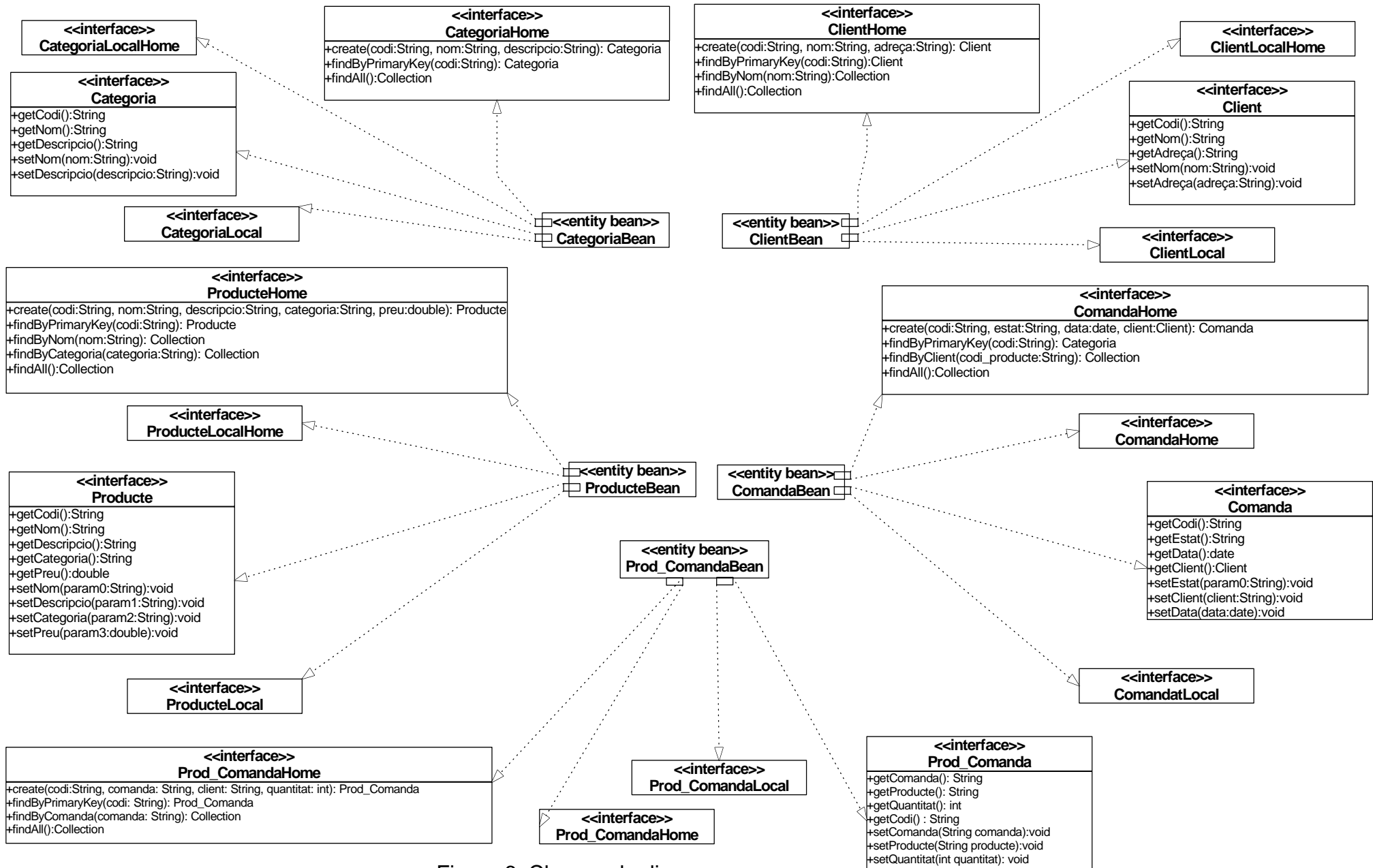


Figura 6: Classes de disseny

Diagrames de components

Es mostra en el diagrama els components J2EE que es faran servir en l'aplicació i la forma en que es connecten entre ells.

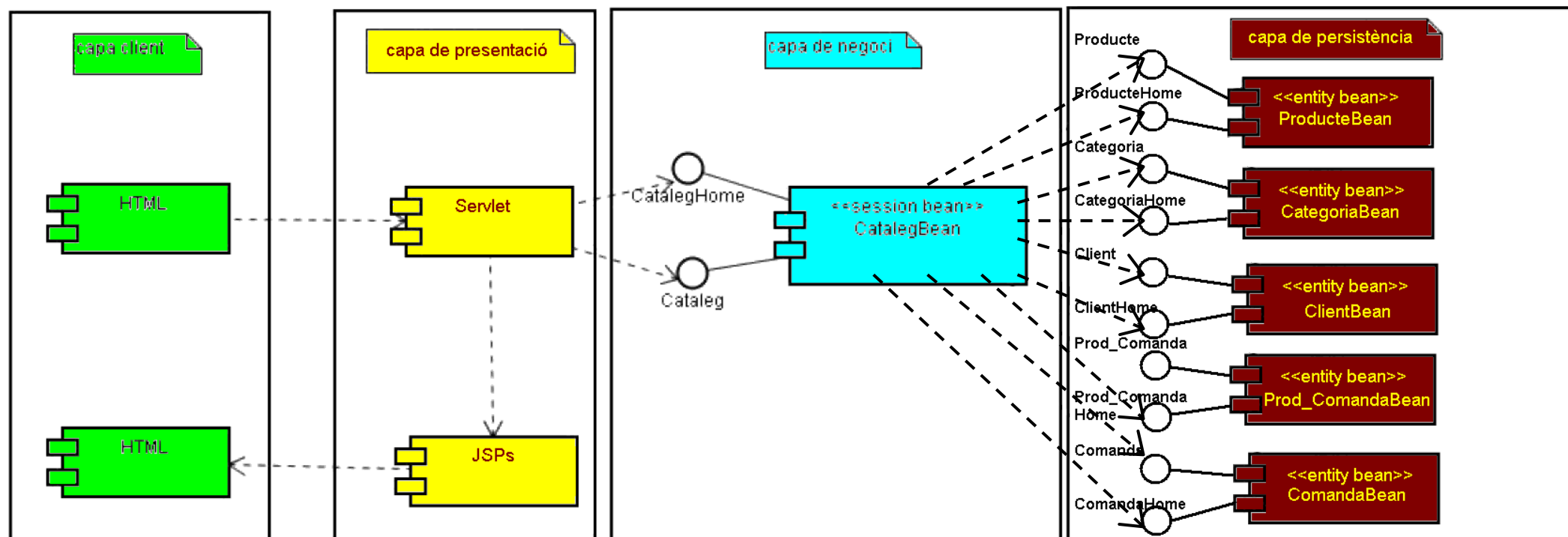


Figura 7: Diagrama de components

Diagrames de desplegament

Es suposen tres màquines:

Client: On els usuaris es connecten al servidor d'aplicacions a través del protocol HTTP del seu navegador.

Servidor d'aplicacions: Màquina on està instal·lat el servidor d'aplicacions JBoss amb els contenidors web i EJB. Es connecta al servidor de base de dades.

Servidor de base de dades: màquina que emmagatzema les dades persistents dins una base de dades relacional.

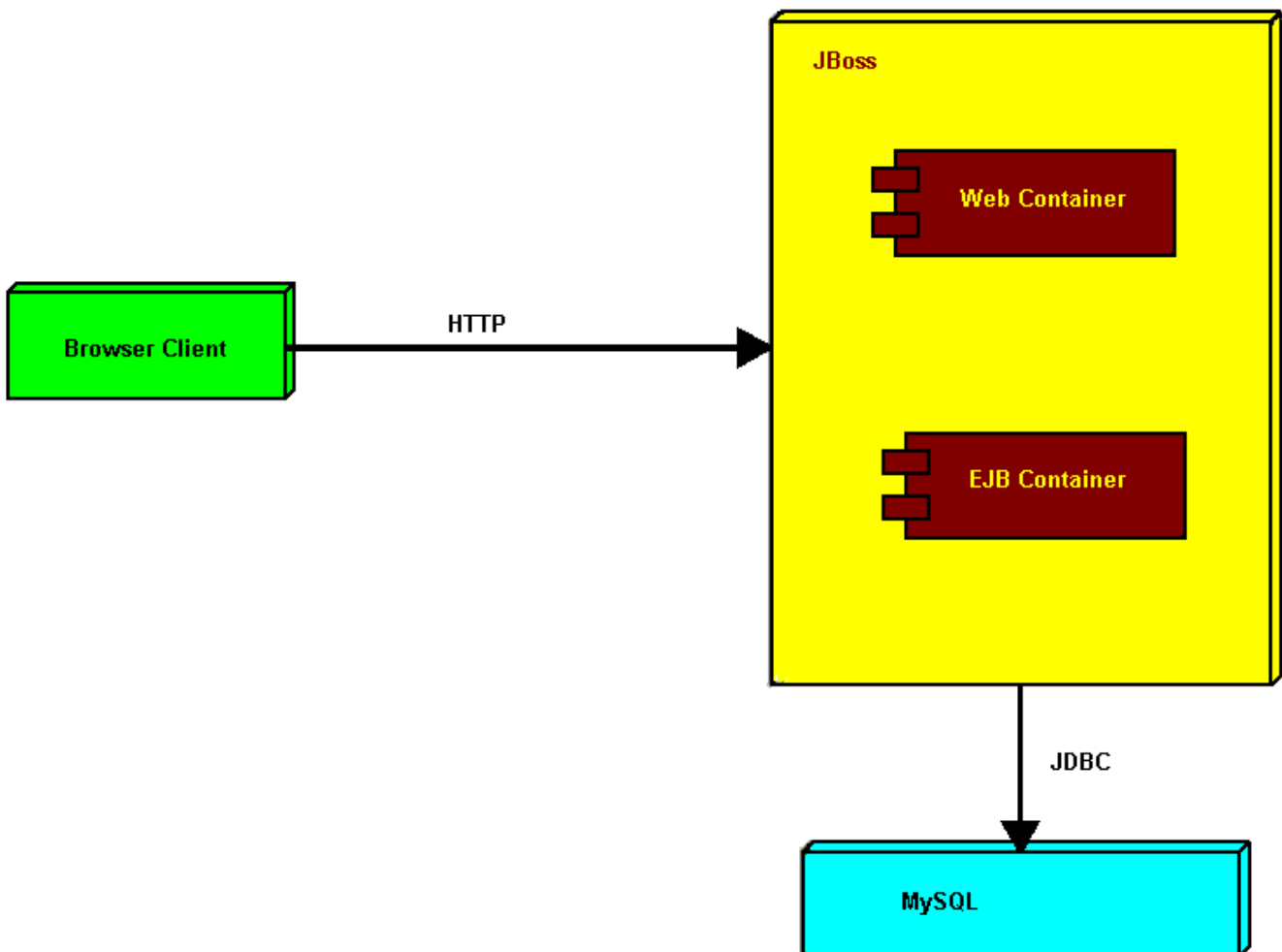


Figura 8: Diagrama de desplegament

Diagrames de seqüència

S'ha seleccionat els casos d'ús Efectuar Comanda per la seva major complexitat ja que inclou Entrar al Sistema i Afegir al carret.

Efectuar Comanda:

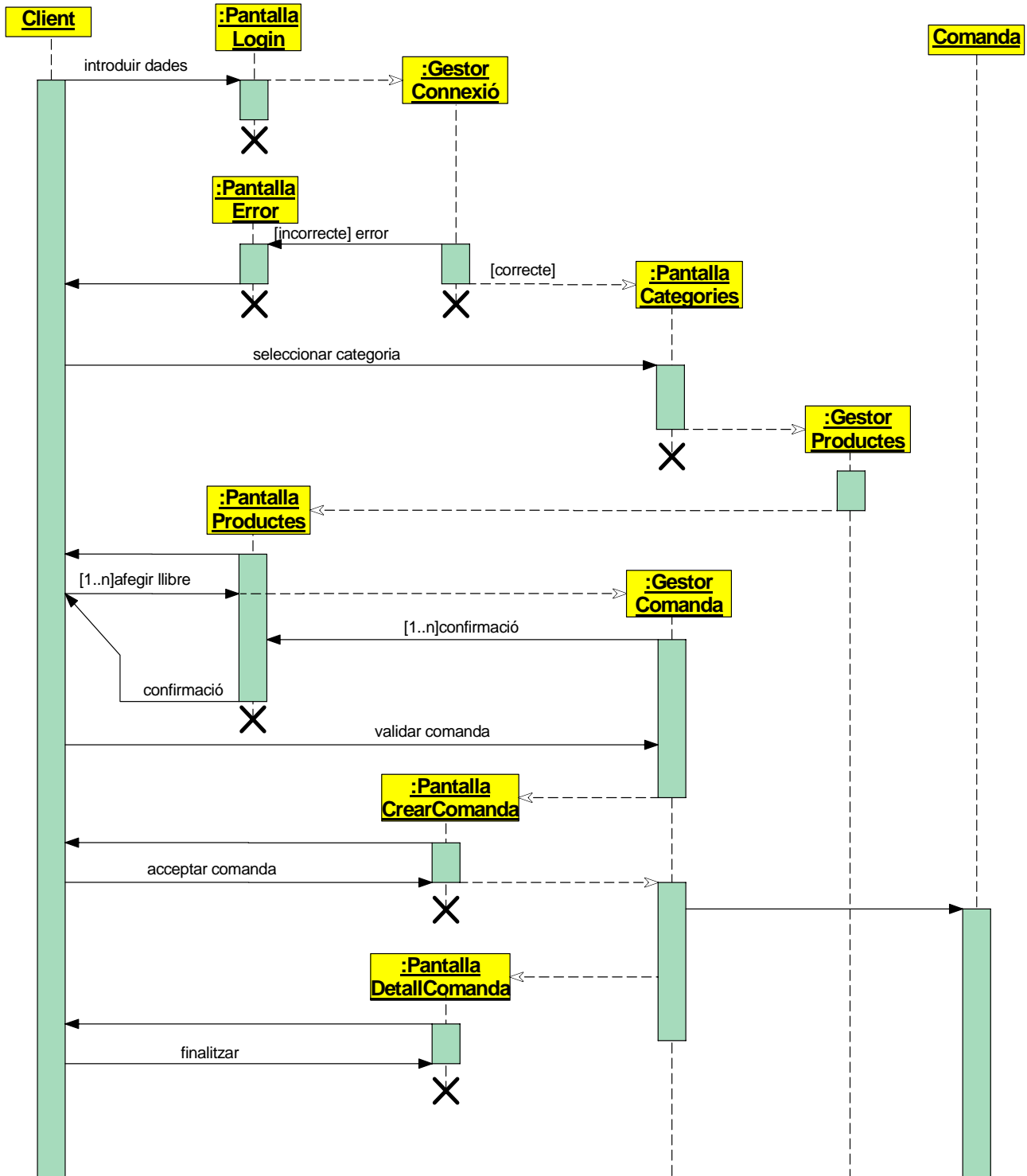


Figura 9: Diagrama de seqüència – Efectuar comanda

Interfície gràfica

Es presenta la maqueta de l'aplicació a partir dels casos d'ús de l'apartat 2. Aquesta maqueta correspon a sèrie de pantalles estàtiques amb HTML amb dades inventades.

Aquestes pàgines HTML es convertiran posteriorment a JSPs per a formar part de la capa de presentació de l'aplicació.

Les pantalles es troben al directori maqueta, respectant i continuant el mateix format que les pantalles del TutorialEPCSD.

Pantalles

Es fa distinció entre les pantalles a les que tenen accés els gestors i els clients.

Des de qualsevol pantalla de gestors es pot accedir a la pantalla **login.htm** mitjançant el botó **Inici** i mitjançant l'índex de l'esquerra al llistat o creació de categories, productes o comandes.

Des de qualsevol pantalla dels clients es pot accedir a la pantalla **crearcomandaU.htm** mitjançant el carret en la part superior i mitjançant l'índex de l'esquerra al llistat de categories, productes o comandes.

Login ([login.htm](#))

Pantalla que permet als usuaris accedir a l'aplicació. En aquesta versió de l'aplicació es fa la distinció entre els usuaris Gestors i els Clients.

En la maqueta es presenten dos botons per cadascun del tipus d'usuaris.

Error ([error.htm](#))

Pantalla genèrica que representa un error de l'aplicació.

Llistat de categories usuaris ([llistarcategoriesU.htm](#))

Ofereix als usuaris un llistat de les categories existents al catàleg.

Ofereix la possibilitat de mostrar el llistat de productes de cada categoria clicant sobre el nom de la categoria.

Dona accés al detall de cada categoria existent.

Permet l'accés al cercador de productes per nom.

Llistat de categories gestors ([llistarcategories.htm](#))

Ofereix als gestors un llistat de les categories existents al catàleg.

Dona accés al detall de cada categoria existent.

Dona accés a les pantalles de gestió de les categories:

- Creació de categoria
- Modificació de categoria
- Esborrat de categoria

Ofereix la possibilitat de mostrar el llistat de productes de cada categoria clicant sobre el nom de la categoria.

Permet l'accés al cercador de productes per nom.

Alta d'una categoria ([crearcategoria.htm](#))

Ofereix als gestors la possibilitat de donar d'alta categories al catàleg.

Per cada categoria s'ha d'especificar el codi, el nom i la descripció.

Modificació d'una categoria ([modificarcategoria.htm](#))

Ofereix als gestors la possibilitat de modificar categories del catàleg.

Per cada categoria es pot modificar el nom o la descripció (el codi no es pot modificar).

Detall d'una categoria ([detallcategoria.htm](#) - [detallcategoriaU.htm](#))

Ofereix als gestors i als clients la possibilitat de visualitzar la informació d'una categoria del catàleg.

En la pantalla es mostren el codi, el nom i la descripció de la categoria.

Llistat de llibres usuaris ([llistatproductesU.htm](#))

Ofereix als usuaris un llistat dels llibres d'una categoria existents al catàleg.

Permet afegir llibres al carret mitjançant el dibuix a la dreta de cada llibre.

Dona accés al detall de cada llibre existent.

Permet l'accés al cercador de productes per nom.

Llibre afegit ([afegit.htm](#))

En la versió final de l'aplicació al afegir un llibre al carret es visualitzarà una indicació dels llibres afegits sota el carret dalt a la dreta.

Per la maqueta es presenta una pantalla de confirmació quan s'afegeix un llibre al carret.

Llistat de llibres gestors ([llistatproductes.htm](#))

Ofereix als usuaris un llistat dels llibres d'una categoria existents al catàleg.

Dona accés al detall de cada llibre existent.

Dona accés a les pantalles de gestió dels llibres:

- Creació de llibres
- Modificació de llibres
- Esborrat de llibres

Permet l'accés al cercador de productes per nom.

Alta d'un llibre ([crearproducte.htm](#))

Ofereix als gestors la possibilitat de donar d'alta llibres al catàleg.

Per cada llibre s'ha d'especificar el codi, el nom, la descripció, la categoria i el preu.

Modificació d'un llibre ([modificarproducte.htm](#))

Ofereix als gestors la possibilitat de modificar llibres del catàleg.

Per cada llibre es pot modificar el nom, la descripció, la categoria o el preu (el codi no es pot modificar).

Detall d'un llibre ([detallproducte.htm](#) - [detallproducteU.htm](#))

Ofereix als gestors i als clients la possibilitat de visualitzar la informació d'un llibre del catàleg.

Pels clients ofereix la possibilitat d'afegir el producte al carret de compra.

En la pantalla es mostren el codi, el nom, la descripció, la categoria i el preu.

Llistat de comandes usuaris ([llistatcomandesU.htm](#))

Ofereix als usuaris un llistat de les comandes efectuades a la botiga virtual (només a les comandes del client).

En la pantalla es mostren el codi de la comanda, el codi client i l'estat de la comanda.

Permet l'accés al detall de la comanda.

Llistat de comandes gestors ([llistatcomandes.htm](#))

Ofereix als gestors un llistat de totes les comandes efectuades a la botiga virtual.

En la pantalla es mostren el codi de la comanda, el codi client i l'estat de la comanda.

Permet l'accés al detall i a la modificació de l'estat de la comanda.

Creació d'una comanda ([crearcomandaU.htm](#))

Ofereix als clients la possibilitat de donar de creació d'una comanda amb els llibres afegits al carret.

En la pantalla es mostren el codi del client, la data actual de la comanda i la llista de llibres amb nom i quantitat afegits a la comanda.

Permet canviar les quantitats dels productes demanats i esborrar productes.

Modificació d'una comanda ([modificarcomanda.htm](#))

Ofereix als gestors la possibilitat de canviar l'estat de les comandes .

En la pantalla es mostren el codi de la comanda i del client, la data actual de la comanda, l'estat i la llista de llibres amb el codi del llibre i la quantitat afegits a la comanda.

Detall comanda ([detallcomanda.htm](#) - [detallcomanda.htm](#))

Ofereix als gestors i als clients la possibilitat de visualitzar la informació de una comanda.

En la pantalla es mostren el codi de la comanda i del client, la data actual de la comanda i l'estat

Pels clients ofereix el nom del llibre, la quantitat i el preu dels productes de la comanda.

Pels gestors ofereix el codi del llibre i la quantitat dels productes de la comanda.

Taules de la base de dades

Les dades persistents que s'han d'emmagatzemar a la BD de la versió que es pretén en aquest pràctica, són les que corresponen a categories, productes, clients, comandes i productes_comandes.

En vermell les taules que farien que l'aplicació estigués completa.

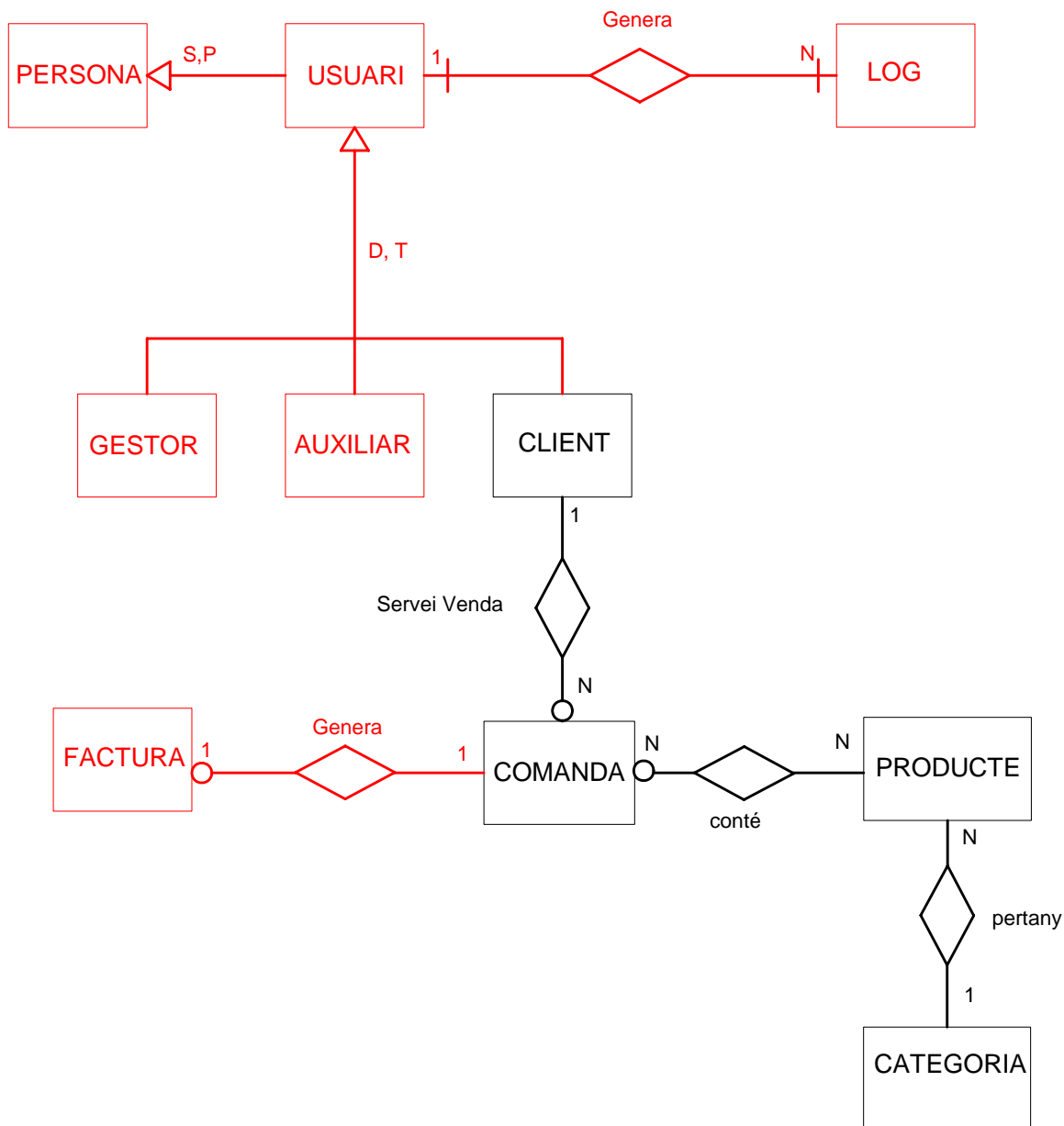


Figura 10: Model entitat-relació

Transformacions del disseny:

El disseny de la BD relacional ens obliga a transformacions per les relacions 1 a N i ternària.

- Per un costat, es crea un camp en la taula PRODUCTE referent al codi-categoria a la que pertany el producte.
- Per un altre costat, en la taula COMANDA tindrem un camp referent al codi-client.
- I una altra taula PROD_COMANDA que relaciona la comanda amb els N productes que pot tenir.

Script:

L'script de creació de les taules de la BD per MySQL està disponible al fitxer EPCSD.zip, en la carpeta ...\\EPCSD\\script

CAPITOL 4 - Implementació

Introducció

En aquest capítol es fa referència a la implementació de l'aplicació botiga virtual.

No es detalla el codi de cadascuna de les classes ja que està disponible dins del fitxer lliurat EPCSD.zip en les carpetes ...\\EPCSD\\cat\\uoc\\epcsd\\tutorial\\...

La botiga virtual és una evolució de la Pràctica Tutoritzada, és per això que s'ha aprofitat molt del codi existent inclús l'estructura de directoris, els fitxers de desplegament i els executables. Reaprofitar codi és molta de la gràcia que té el desenvolupament en capes d'una aplicació J2EE i la programació OO.

S'ha eliminat el codi de la Pràctica Tutoritzada que no compleix amb el patró MCV-2.

M'hagués agradat desenvolupar la capa presentació amb l'Struts, sembla ser que s'està convertint en un estàndard de facto. El problema ha estat de temps.

A més de l'ampliació de les funcionalitats, separació còmoda dels mètodes d'administrador i client, test i resolució de codi erroni en algun dels components, s'han aportat millores en la implementació del Servlet que es podran veure en detall en el capítol corresponent.

Desplegament

En el fitxer **application.xml** dins la carpeta ...\\EPCSD\\scr\\META-INF del arxiu EPCSD.zip lliurat es descriu el nom dels fitxers .jar i .war que contindran les classes java, els paràmetres de configuració de l'aplicació i permetran la seva execució.

Mitjançant aquest fitxer és generen els executables de l'aplicació:

EPCSD.ear: Fitxer comprimit que conté els moduls de l'aplicació i els descriptors de desplegament.

Negoci.jar : Fitxer que conté les classes java (EJBs) que modelen la lògica del negoci.

Presentacio.war: Fitxer que conté els elements (JSPs i Servlet) que modelen la capa presentació de l'aplicació.

En el fitxer **ejb-jar.xml** dins la carpeta ...\\EPCSD\\scr\\META-INF del arxiu EPCSD.zip lliurat es descriu les característiques de desplegament dels EJBs.

En el fitxer **web.xml** dins la carpeta ...\\EPCSD\\scr\\WEB-INF del arxiu EPCSD.zip lliurat es descriuen els paràmetres de configuració i seguretat per l'accés a l'aplicació.

Com a punt destacat, shan creat permissos pels clients, tot plegat lligat amb els fitxers **roles.properties** i **user.properties**.

Dins el fitxer **build.xml** dins la carpeta ...\\EPCSD\\scr del arxiu EPCSD.zip lliurat s'estableixen els paràmetres de l'aplicació per a que puguin ser compilants mitjançant l'**ant**.

Implementació dels EJBs d'entitat

Segons el disseny en l'apartat diagrama de classes del capítol 3 (Figura 6) s'implementaran els EJBs d'entitat de les classes Categoria, Client, Producte, Comanda i Prod_Comanda.

Es fa un breu resum de les interfícies i classes utilitzades. No es fan grans explicacions ni s'annexa el codi perquè està disponible dins del fitxer EPCSD.zip lliurat.

Pels EJBs s'han desenvolupat mètodes main de proves. La sortida obtinguda està disponible a la carpeta ...EPCSD\out dins del fitxer EPCSD.zip lliurat.

Per la implementació dels EJBs d'entitat s'han seguit els següents paràmetres:

- Utilització d'Entity Beans
- EJB que faran mapeig a les taules CATEGORIA, PRODUCTE, CLIENT, COMANDA i PROD_COMANDA
- Els EJBs segueixen l'especificació EJB 2.0
- Ús de CMP, els EJBs tindran persistència gestionada pel contenidor
- No es farà servir CMR
- Els EJBs tindran interfícies locals i remotes
- No es faran servir Value Objects entre la capa de persistència i negoci.
- Es fan servir les opcions per defecte del JBoss excepte en l'ús de MySql en comptes de Hypersonic per emmagatzemar les dades persistents.

Classe Producte

Conté la següent informació:

codi
nom
descripció
categoria
preu

Codi del Entity Bean disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/ejb/ProducteBean.java](#)

Codi interfície d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ProducteHome.java](#)

Codi interfície local d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ProducteLocalHome.java](#)

Codi interfície remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Producte.java](#)

Codi interfície local remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ProducteLocal.java](#)

Classe Categoria

Conté la següent informació:

codi
nom
descripció

Codi del Entity Bean disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/ejb/CategoriaBean.java](#)

Codi interfície d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/CategoriaHome.java](#)

Codi interfície local d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/CategoriaLocalHome.java](#)

Codi interfície remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Categoria.java](#)

Codi interfície local remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/CategoriaLocal.java](#)

Classe Client

Conté la següent informació:

codi
nom
adreça
codi compte corrent

Codi del Entity Bean disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/ejb/ClientBean.java](#)

Codi interfície d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ClientHome.java](#)

Codi interfície local d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ClientLocalHome.java](#)

Codi interfície remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Client.java](#)

Codi interfície local remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ClientLocal.java](#)

Classe Comanda

Conté la següent informació:

codi
data
client
estat

Codi del Entity Bean disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/ejb/ComandaBean.java](#)

Codi interfície d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ComandaHome.java](#)

Codi interfície local d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ComandaLocalHome.java](#)

Codi interfície remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Comanda.java](#)

Codi interfície local remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/ComandaLocal.java](#)

Classe Prod Comanda

Conté la següent informació:

codi
comanda
producte
quantitat

Codi del Entity Bean disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/ejb/Prod ComandaBean.java](#)

Codi interfície d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Prod ComandaHome.java](#)

Codi interfície local d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Prod ComandaLocalHome.java](#)

Codi interfície remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Prod Comanda.java](#)

Codi interfície local remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Prod ComandaLocal.java](#)

Classes client per proves dels EJBs

S'han implementat classes client de prova per cada EJB.
El codi per cada classe està disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/client/CategoriaClient.java](#)
[...EPCSD/cat/uoc/epcsd/tutorial/client/ProducteClient.java](#)
[...EPCSD/cat/uoc/epcsd/tutorial/client/ClientClient.java](#)
[...EPCSD/cat/uoc/epcsd/tutorial/client/ComandaClient.java](#)
[...EPCSD/cat/uoc/epcsd/tutorial/client/Prod_ComandaClient.java](#)

I les sortides de cada classe de prova a:

[...EPCSD\out\Categoria.out](#)
[...EPCSD\out\Producte.out](#)
[...EPCSD\out\Client.out](#)
[...EPCSD\out\Comanda.out](#)
[...EPCSD\out\Prod_Comanda.out](#)

Implementació del EJB de sessió

Segons el disseny en l'apartat diagrama de classes del capítol 3 (Figura 5) s'implementà un EJB d'entitat que farà de façana cap els EJBs d'entitat de les classes Categoria, Client, Producte, Comanda i Prod_Comanda.

Es fa un breu resum de les interfícies i classes utilitzades. No es fan grans explicacions ni s'annexa el codi perquè està disponible dins del fitxer EPCSD.zip lliurat.

Pel EJB de sessió s'ha desenvolupat mètode main de proves. La sortida obtinguda està disponible a la carpeta ...\\EPCSD\\out dins del fitxer EPCSD.zip lliurat.

Per la implementació dels EJBs de sessió s'han seguit els següents paràmetres:

- Utilització d'un EJB de sessió sense estat
- Implementarà tots els mètodes que els clients poden accedir, segons diagrama de classes mostrat posteriorment
- Utilització d'interfícies remotes i no ús d'interfícies locals.
- Especificació EJB 2.0
- Es faran servir Value Objects.
- Es desplega el EJB amb el nom JNDI per defecte.
- Els mètodes de negoci no faran servir CMR.

Classe CatalogBean

Conté els mètodes descrits al diagrama de classes:

Codi del Entity Bean disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/ejb/CatalogBean.java](#)

Codi interfície d'inici disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/CatalogHome.java](#)

Codi interfície remota disponible a:

[...EPCSD/cat/uoc/epcsd/tutorial/interfaces/Catalog.java](#)

Classes Value Object

[...EPCSD/cat/uoc/epcsd/tutorial/vo/ProducteVO.java](#)

[...EPCSD/cat/uoc/epcsd/tutorial/vo/CategoriaVO.java](#)

[...EPCSD/cat/uoc/epcsd/tutorial/vo/ClientVO.java](#)

[...EPCSD/cat/uoc/epcsd/tutorial/vo/ComandaVO.java](#)

[...EPCSD/cat/uoc/epcsd/tutorial/vo/Prod_ComandaVO.java](#)

Classes client per proves del EJB

El codi de la classe està disponible a:

<...EPCSD/cat/uoc/epcsd/tutorial/client/CatalegClient.java>

I les sortides de cada classe de prova a:

<...EPCSD\out\Cataleg.out>

Implementació del Servlet

Seguint el patró MVC-2 s'implementa un ControladorServlet que farà de receptor de les peticions dels clients, gestionarà els accessos a la lògica de negoci i dirigirà els JSPs.

El ControladorServlet de la Pràctica Tutoritzada fa servir una estructura if-then-else per gestionar les accions i dades necessàries per l'usuari durant l'accés a la botiga virtual, que també es va impementar nicialment i és funcional

[../EPCSD\deprecated\ControladorServlet.java](#)

però s'ha rebutjat per les millores actuals.

El contolador implementat no té una estratègia òptima com la que implementa l'Struts mitjantzant action-mapping, però redueix les línies de codi i igualment entendor que la estructura if-then-else.

No s'annexa el codi perquè està disponible dins del fitxer EPCSD.zip lliurat a la carpeta:

[../EPCSD\cat\uoc\epcsd\tutorial\web\ControladorServlet.java](#)

Milllores Servlet

La idea d'implementació d'un nou servlet te com a punt de partida la reducció de codi.

Mirant atentament les accions que s'efectuen en una aplicació com la botiga virtual ens podem adonar que tenen un patró que és pot generalitzar en aquest algorisme d'alt nivell.

Obtenir Request

```
Efectuar_accions_segons_operació
  [opcional] llegir_paràmetres
  [opcional] accions_des_de_negoci - Value Objects
  [opcional] accions_cap_a_negoci - CRUD
  [opcional] escriure_atributs
  carregar_següent_JSP
```

Enviar Response

A partir de l'algorisme anterior hauriem de poder treure alguna estratègia millor.

La idea és plasma amb aquest esquema:

S'obté la request, es compara el paràmetre operacio amb les operacions disponibles:

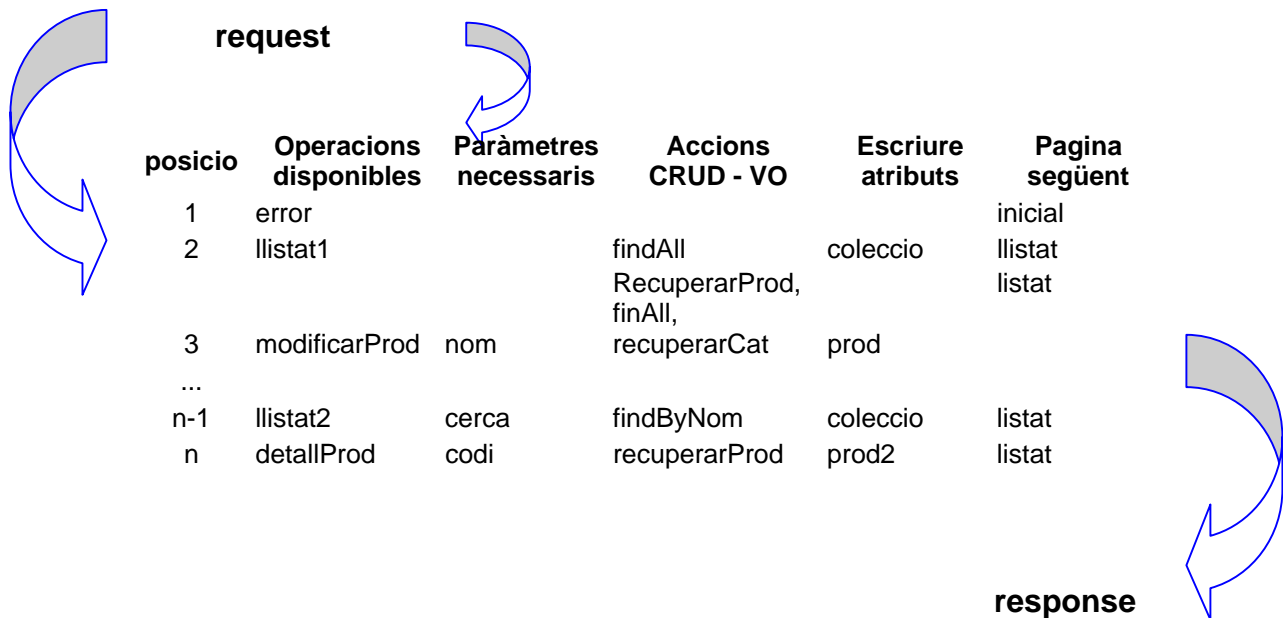


Figura 11: Esquema Servlet

L'algorisme determina si es requereixem paràmetres a partir de la posició de la taula i en cas afirmatiu es guarden en un String (podria també ser una llista).

El següent pas és determinar quines son les accions CRUD (Crear, llegir, actualitzar o esborrar) i VO (retorn de Value Object) y realitzar-les.

Després d'aquest pas serà necessari escriure els atributs a la request.

Per últim, i abans de fer el forward i enviar la response, l'algorisme determina la pagina següent a carregar.

Aquest seria el codi algorítmic aproximat:

```
String operacio = request.getParameter("operacio");
Int Posicio = 1;
Mentre no trobat fer
    Si Operacio == taula[1, posicio] llavors trobat == cert; fsi
    Posicio ++;
fmentre

Si taula[2,posicio] != "" llavors
    String parametre = request.getParameter(taula[2,posicio]);
fsi

Mentre mes_accions fer
    //Cal extreure les accions per separat
    Extreure_nom_accio()
    //Requereix factoria d'accions
    Objecte = Efectuar_accio (taula[3,posicio]);
fmentre

request.setAttribute(taula[4,posicio], Objecte);

String SeguentPag = taula[5,posicio];

RequestDispatcher rd =
getServletContext().getRequestDispatcher(seguentPag);
rd.forward(request, response);
```

El servlet implementat fa una primera aproximació a aquest disseny amb vectors.

Encara queden algunes qüestions per resoldre com la millorar font per generar la taula (llistes, arrays, fitxers, piles...).

D'altres com la de generar accions a partir d'un text, una possibilitat seria un codi semblant a aquest:

```
public class AccionFactory {
    public static Accion getAccion(String accion){
        Accion action = null;
        try {
            action = (Accion)Class.forName(accion).newInstance();
        } catch (InstantiationException e) {
            System.out.println("Error al instanciar la classe");
            e.printStackTrace();
        } catch (IllegalAccessException e) {
            System.out.println("Acceso ilegal a no se que ¿?");
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            System.out.println("Tas tonto esa clase no existe");
            e.printStackTrace();
        } return action;
    }
}
```

Implementació dels JSPs

Aprofitant la maqueta dissenyada en la fase anterior de desenvolupament de la botiga virtual, s'afegirà el codi java necessari per automatitzar les accions seleccionades per l'usuari en la botiga virtual.

Per l'accés d'administrador de l'aplicació és desenvolupen 15 JSPs disponibles a la carpeta:

[...../EPCSD/src/docroot/model2/admin](#)

Per l'accés d'administrador de l'aplicació és desenvolupen 7 JSPs disponibles a la carpeta:

[...../EPCSD/src/docroot/model2/client](#)

I tres JSPs comunes, corresponents a login i errors disponibles a la carpeta:

[...../EPCSD/src/docroot/model2/](#)

CAPITOL 5: Manual d'instal·lació de la botiga virtual

Introducció

Per instal·lar la botiga virtual s'ha tenir correctament configurat el programari detalla en el capítol 2 d'aquesta memòria.

Instal·lació

1. Extreure en el directori desitjat els fitxers i les carpetes del document EPCSD.zip.
2. Executar l'script EPCSD.sql per la base de dades disponible a la carpeta ...\\EPCSD\\sql\\
3. Compilar amb l'ant el projecte, des de una finestra de DOS accedir a la carpeta ...\\EPCSD\\src i escriure **ant**.
4. Posar en marxa el JBoss, accedir des de DOS al directori bin de la instal·lació de JBoos i escriure **run**.

Execució

Obrir el navegador:

- a. Si es vol accedir a les opcions d'administrador accedir a <http://localhost:8080/epcsd/model2/admin/benviguda.jsp>
- b. Si es vol accedir a les opcions d'administrador accedir a <http://localhost:8080/epcsd/model2/admin/benvigudaU.jsp>

Per accedir com administrador cal posar com a usuari: admin i paraula de pas: admin.
Per accedir com client cal posar com a usuari: client i paraula de pas: client.

CAPITOL 6 – Valoració Econòmica

Introducció

Encara que no sigui l'àmbit d'aquest tipus de TFC m'agradaria remarcar alguns aspectes a tenir en compte que he après durant el desenvolupament de l'aplicació J2EE.

1. Importància de l'experiència
2. Perfils dels desenvolupadors

Aquests dos aspectes afecten sobre dos punts que qualsevol enginyer informàtic ha d'aspirar a saber controlar:

Detalls de la planificació i supervisió dels programadors.

Valoració econòmica, el termini de lliurament i detalls planificació d'aplicacions J2EE

Els dos conceptes principals pel desenvolupament d'una aplicació, valoració econòmica i termini de lliurament, depenen de factors propis de l'empresa desenvolupadora i de la situació del mercat.

Aquests factors són principalment el preu/hora dels desenvolupadors i els recursos disponibles, segons aquests paràmetres es poden engegar els detalls de la planificació del projecte.

Existeix un altre factor que personalment considero especialment crític per aquest tipus de desenvolupament, l'experiència.

No puc entrar en dades detallades de la diferència que hi hauria entre els pressupostos d'una empresa amb llarga experiència en desenvolupaments J2EE i un altra amb poca o cap per una mateixa aplicació.

El que sembla segur és que les característiques pròpies de les aplicacions J2EE proporcionen molts beneficis a les empreses que disposen de llibreries de classes, components integrats i codi extensament provat que poden reutilitzar per nous projectes.

La programació OO ja ofereix aquests beneficis però l'arquitectura en capes multiplica la reusabilitat i la re-enginyeria. El ventall de possibilitat de programari i el no tenir una plataforma integrada sinó diferents components, augmenta la complexitat també a favor de les empreses amb experiència.

Aquest concepte que sembla evident en qualsevol tipus de negoci, ha de ser estudiat en detall en un món tant competitiu i canviant com és el de les tecnologies de la informació.

Des del meu punt de vista, sembla evident i alhora preocupant que per ser competitiu, els primers desenvolupaments d'aplicacions J2EE hagin de ser per força deficitaris i el retorn de la inversió s'espera a no curt termini.

Sense anar més enllà, plasmo gràficament els conceptes explicats en aquest capítol:

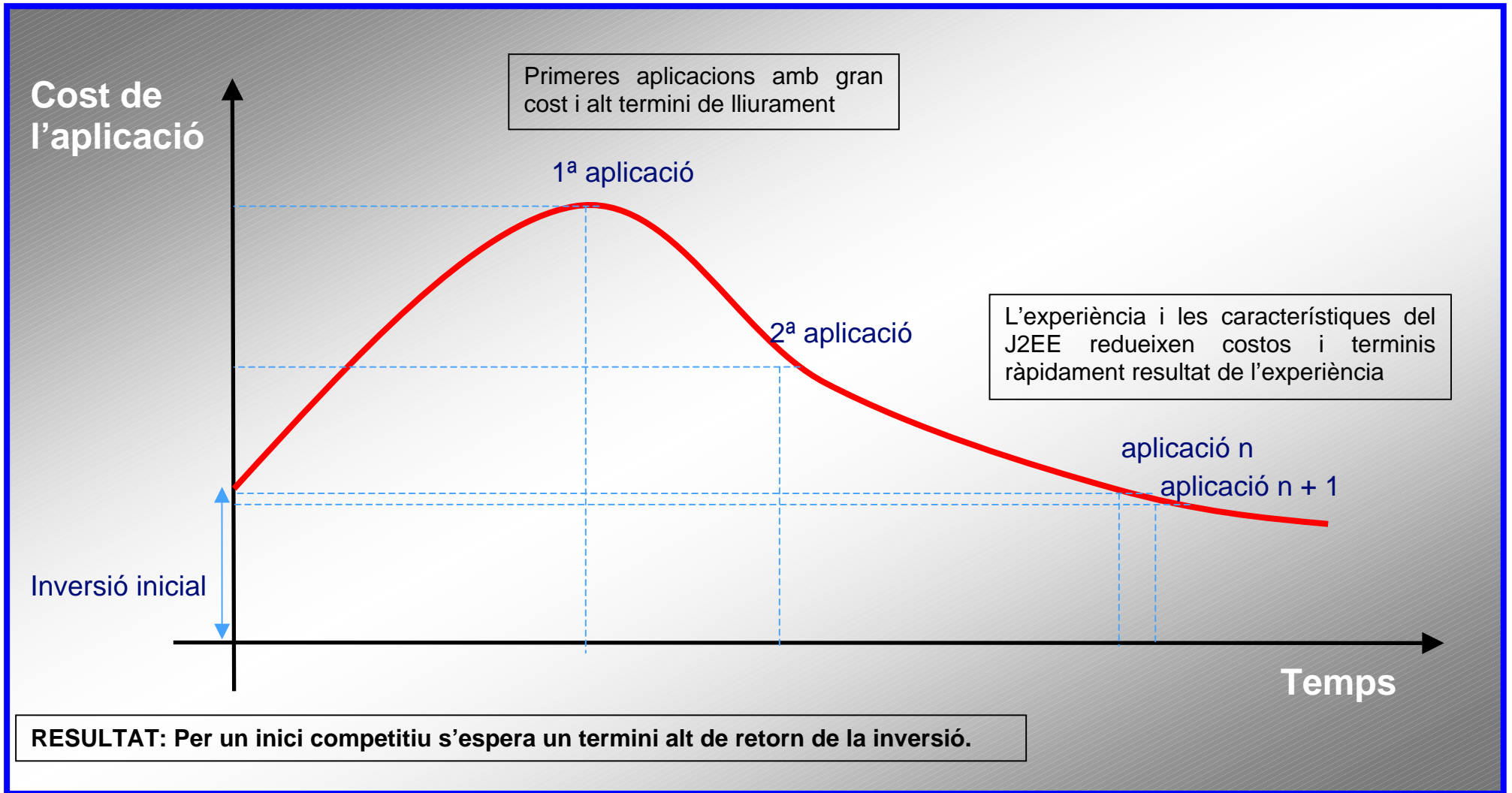


Figura 12: Gràfica relació experiència-cost

Supervisió i valoració de desenvolupadors d'aplicacions J2EE

El concepte de desenvolupament en capes introdueix un nou concepte respecte la programació clàssica: la divisió de feines i per tant la importància del treball en equip.

Sembla evident que el programador ja no és un element aïllat al qui s'encarrega una feina i obté un producte que tradicionalment només ell coneix.

Aquest concepte va ajudar a que moltes empreses es decantessin a adaptar solucions de programari existent en el mercat, en comptes de seguir amb desenvolupament propis.

La programació OO i la utilització de les eines CASE va millorar la situació per als desenvolupaments a mida, però les característiques de J2EE fan que s'ampliïn les possibilitats.

S'obren camins per perfils d'empleats que per les seves actituds i aptituds no eren els més adequats per la programació clàssica. L'arquitectura en capes permet la divisió de les feines de manera que existeixin diferents perfils de persones que puguin adaptar-se a alguna de les capes que componen l'arquitectura J2EE en el desenvolupament d'aplicacions i alhora s'hagi de donar molta rellevància al treball en equip.

Dins d'aquest nou concepte, pren rellevància la supervisió i valoració dels programadors perquè ara més que mai formen part d'un equip.

CAPITOL 7 - Conclusions

Per començar voldria reflectir les diferències més rellevant que he trobat de J2EE respecte Visual Studio .NET

	J2EE	VisualStudio .NET
Extensió	Més de 10 anys amb gran nombre d'aplicacions	Modernitat i gran evolució
Maduresa	Alt nombre d'eines disponibles i evolucionades	Menys eines i menys evolucionades
Competitivitat	Organisme de control format per empreses amb renom	Propietari únic
Portabilitat	Alta	Windows majoritàriament
Simplicitat	Elitista i heterogeni	Facilitats i homegeni

Figura 13: Quadre comparatiu J2EE vs VisualStudio .NET

Aquestes característiques atorguen a les dos plataformes avantatges o inconvenients segons l'entorn on s'hagi de desenvolupar l'aplicació i segons la seva mida.

- Elecció de J2EE per desenvolupaments multiplataforma.
- Elecció de J2EE per projectes d'envergadura elevada.
- Elecció de .NET per projectes des de zero o projectes petits sobre Microsoft Windows.

En tot cas l'elecció d'una o altra plataforma dependrà d'un estudi econòmic acurat, prenent com a referència altres paràmetres com l'experiència de l'empresa desenvolupadora, el mercat de programadors, cost de llicències...

Disseny

En quan al disseny d'aplicacions J2EE queda palès que una arquitectura MCV-2 té suficients arguments per ser plenament acceptada.

En el desenvolupament de la botiga virtual s'ha implementat aquest disseny però s'ha perdut eficiència en l'estratègia d'implementació del ControladorServlet.

La comunitat Java està fent estàndard de facto l'estratègia de disseny que implementa Struts. En termes generals Struts utilitza un patró MVC amb un controlador frontal qui dirigeix el flux de peticions i respostes mitjançant una estratègia action-mapping que és força eficient respecte altres com la implementada en la botiga virtual:

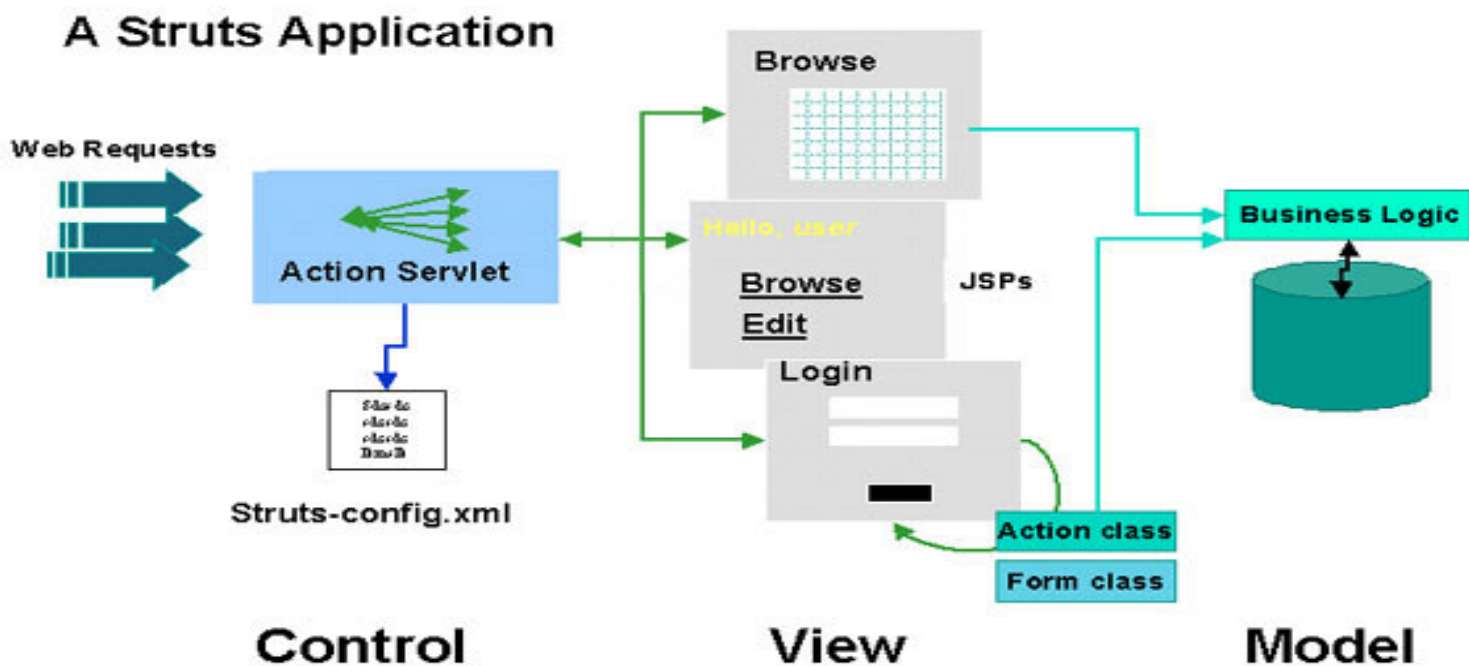


Figura 14: Aplicació Struts

Per finalitzar voldria fer una reflexió de la repercussió d'aquest treball final de carrera en la meva persona:

Vaig iniciar aquest treball final de carrera amb la idea que J2EE pertany a un món elitista.

L'extensió actual i les constants campanyes de marketing de la multinacional més gran de software (tots sabem el nom), la facilitat de pirateig de productes que requereixen llicència i els requeriments de coneixements complexos per tenir una plataforma de desenvolupament adequada a les necessitats, no ajuden a una plataforma amb la filosofia de codi lliure com és J2EE.

Encara són poques les ànimes inquietes que tinguin la necessitat d'iniciar-se en aquest món. Esperem que Internet i tot el que envolta als beneficis dels serveis web, vagi resolent aquesta situació.

Acabo aquest projecte amb la mateixa idea reforçada, és clar que els desenvolupadors en són d'elitistes, però el que s'hauria de resoldre és la poca incursió d'aplicacions J2EE en el món industrial (no així en el financer i educatiu). Haurien d'haver més empreses que desenvolupessin el seu software sobre plataformes J2EE, més claredat en el ús comercial de les aplicacions de l'entorn J2EE i més enginyers informàtics en la presa de decisions.

Finalitzo amb l'idea clara que he d'iniciar un projecte de desenvolupament J2EE dins de la meva empresa. Potser encara em queda molt de camí a recórrer, però aquest TFC m'ha aclarit molts conceptes que a partir d'ara seré capaç de defensar.

GLOSSARI

ant – Apache Ant, és un eina que permet automatitzar els mecanismes de construcció i compilació de manera sencilla

application.xml – Explicació pag.36

BMP – Bean Managed Persistence. Estratègia de desenvolupament que té com a característica la gestió de la persistència per la classe i no pel contenidor.

build.xml – Explicació pag.36

CASE - Computer Aided Software Engineering – eines de suport que ajuden al desenvolupament, manteniment i qualitat en el desenvolupament d'aplicacions.

CMP – Container Managed Persistence. Estratègia de desenvolupament que permet que la persistència sigui gestionada pel contenidor.

CMR - Container-Managed Relationships. Estratègia de desenvolupament que permet que el contenidor gestioni les relacions entre les taules de la base de dades.

CRUD - Create, read, update and delete – creació, lectura, actualització i esborrat que són les tasques habituals del manteniment de les entitats de les aplicacions.

EJB – Enterprise JavaBeans – Estratègia i model que permet encapsular els objectes del negoci sense haver d'implementar detalls de concurrència, transaccions, persistència, o seguretat

ejb-jar.xml – Explicació pag.36

EPCSD.ear – Explicació pag.36

interfície gràfica – manera gràfica que conté els punts de contacte de l'usuari per que aquest pugui interactuar amb l'aplicació

J2EE – Explicació pag.3

JDBC - Java Database Connectivity – Col·lecció de mètodes que permeten l'execució d'operacions des de el codi cap a la base de dades.

JNDI – Java Naming and Directory Interface – Extensió de la plataforma Java que proveix a les aplicacions amb una interfície unificada per que puguin ser cridades independentment dels detalls d'implementació.

Know-how – Saber-com, saber com adaptar tasques genèriques als processos interns propis de l'entorn on es desenvolupen.

MVC-2 – Patró arquitectònic, explicació pàg 18

negoci.jar – Explicació pag.36

OO – Orientat a objectes – Es defineix com la definició d'una sèrie d'objectes i les relacions entre ells com a mètode de programació.

presentacio.war – Explicació pag.36

propietat intel·lectual – Propietari de la idea o producte generat

TFC – Treball final de carrera

Top-down – Decisió presa des de els nivells més alts de direcció cap a la resta d'empleats, normalment no són discutibles.

web.xml – Explicació pag.36

BIBLIOGRAFIA

General

<http://www.programacionfacil.com>
<http://www.programacion.net/>
<http://es.wikipedia.org/wiki/Portada>
<http://www.adictosaltrabajo.com/>
<http://www.codigofuente.net/>
<http://webexperto.com/tutoriales/>

J2EE

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>
<http://www.programacion.net/java/tutoriales/J2EE/>
<http://tomcat.apache.org/tomcat-3.3-doc/tomcat-ug.html>
<http://www.adrformacion.com/cursos/j2ee/leccion3/tutorial4.html>

Patrons

<http://java.sun.com/blueprints/corej2eepatterns/Patterns/>
<http://javaweb.osmosislatina.com/curso/mvc.htm#java>
<http://www.codejava.org/?idxpagina=9&idxnota=32795&destacada=1>
<http://www.programacion.net/java/tutorial/patrones/>

EJBs

<http://java.sun.com/products/ejb/>
<http://www.osmosislatina.com/java/ejb.htm>
http://www.programacion.net/java/tutorial/jap_aplic_iboss/

Servlet I JSPs

<http://java.sun.com/products/jsp/>
<http://geneura.ugr.es/~jmerelo/JSP/>
http://www.programacionfacil.com/java_jsp:java_server_page
<http://webexperto.com/tutoriales/listado.php?cod=15>