

# **Portal de B2C i gestió de comandes per Internet**

*Jordi Forns i Vilarrasa*

*ETIG*

*Albert Grau Perisé*

*18/06/2004*

## 1. Resum

Aquest treball de fi de carrera té la voluntat de realitzar un petit projecte informàtic seguint el cicle de vida de la forma en què s'ha vist a les assignatures d'Enginyeria del Programari, així com explorar la tecnologia Java 2 Enterprise Edition, que serà l'escollida per a la implementació del projecte.

No hi ha dubte que els nous llenguatges i entorns de programació (sobretot Java i .Net), basats pràcticament al 100% en l'orientació a objectes són un marc ideal per un projecte que té aquest model com a base tecnològica en totes les seves fases. He volgut basar-me en el model MVC durant tot el projecte per organitzar les diferents capes, classes... i també he volgut posar-lo a prova: si el model diu que els seus tres components (model, view, controller) són independents entre sí i intercanviables, en cap moment el fet de realitzar una aplicació basada en Internet ens ha de condicionar el disseny de l'aplicació, és a dir, no ha de ser diferent del que fariem per una aplicació d'escriptori o de consola. Com veurem, això s'ha complert sobradament.

Un altre objectiu d'aquest treball és, òbviament, conèixer les tecnologies més destacades de Java 2 Enterprise Edition. Si bé no s'han pogut estudiar totes, s'han cobert Java Server Pages (per la vista), Servlets i Filtres (pel controlador), Enterprise Java Beans (pel model), Java Mail. Per motius de temps, no s'han pogut cobrir Struts, Java Server Faces ni interfícies basades en XML; aquestes dues últimes tecnologies les considero especialment interessants i molt probablement s'incorporin en la versió comercial del projecte, el destinatari del qual, com s'explica més endavant, és un negoci real de venda i reparació d'ordinadors.

Finalment, dir que tot el programari d'aquest projecte s'ha realitzat mitjançant i corre sobre programari lliure: Eclipse com a IDE, JBoss com a servidor d'aplicacions, Tomcat com a servidor web i MySQL com a SGBD.

## 2. Índex de continguts

<b>1.</b>	<b>RESUM</b> .....	<b>2</b>
<b>2.</b>	<b>ÍNDIX DE CONTINGUTS</b> .....	<b>3</b>
<b>3.</b>	<b>ÍNDIX DE FIGURES</b> .....	<b>5</b>
<b>4.</b>	<b>INTRODUCCIÓ</b> .....	<b>6</b>
4.1.	CONTEXT .....	6
4.2.	OBJECTIUS .....	8
4.3.	ENFOCAMENT I MÈTODE SEGUIT .....	10
4.3.1.	<i>Fase 1</i> .....	10
4.3.2.	<i>Fase 2</i> .....	10
4.3.3.	<i>Fase 3</i> .....	11
4.4.	PLANIFICACIÓ .....	13
4.4.1.	<i>Prevista</i> .....	13
4.4.2.	<i>Real</i> .....	13
4.5.	PRODUCTES OBTINGUTS .....	14
4.6.	SOBRE AQUESTA MEMÒRIA .....	15
<b>5.</b>	<b>ANÀLISI ORGÀNIC</b> .....	<b>16</b>
5.1.	SUBSISTEMA DE MANTENIMENT DE L'ESTRUCTURA .....	16
5.2.	SUBSISTEMA D'APARADOR VIRTUAL .....	16
5.3.	SUBSISTEMA DE PETICIONS .....	17
5.4.	SUBSISTEMA DE SEGURETAT .....	18
<b>6.</b>	<b>ANÀLISI FUNCIONAL</b> .....	<b>19</b>
6.1.	SUBSISTEMA DE MANTENIMENT DE L'ESTRUCTURA .....	19
6.1.1.	<i>Diagrama estàtic</i> .....	19
6.1.2.	<i>Casos d'ús</i> .....	20
6.1.3.	<i>Descripció textual dels casos d'ús</i> .....	20
6.2.	SUBSISTEMA D'APARADOR VIRTUAL .....	29
6.2.1.	<i>Diagrama estàtic</i> .....	29
6.2.2.	<i>Casos d'ús</i> .....	29
6.2.3.	<i>Descripció textual dels casos d'ús</i> .....	30
6.3.	SUBSISTEMA DE PETICIONS .....	35
6.3.1.	<i>Diagrames d'estats</i> .....	35
6.3.2.	<i>Casos d'ús</i> .....	36
6.3.3.	<i>Descripció textual dels casos d'ús</i> .....	36

---

6.4.	SUBSISTEMA DE SEURETAT .....	41
6.4.1.	<i>Diagrama estàtic</i> .....	41
6.4.2.	<i>Casos d'ús</i> .....	41
6.4.3.	<i>Descripció textual dels casos d'ús</i> .....	41
6.5.	PAQUETS DEL SISTEMA.....	42
6.6.	DIAGRAMA ESTÀTIC .....	43
6.7.	PERSISTÈNCIA .....	44
6.7.1.	<i>FAMILIES</i> .....	44
6.7.2.	<i>ATRIBUTS</i> .....	44
6.7.3.	<i>ARTICLES</i> .....	44
6.7.4.	<i>ATRIBUTSARTICLE</i> .....	45
6.7.5.	<i>EQUIPS</i> .....	45
6.7.6.	<i>COMPONENTS</i> .....	45
6.7.7.	<i>COMANDES</i> .....	45
6.7.8.	<i>ARTICLESCOMANDA</i> .....	45
6.7.9.	<i>EQUIPSCOMANDA</i> .....	46
6.7.10.	<i>CLIENTS</i> .....	46
6.7.11.	<i>USUARIS</i> .....	46
<b>7.</b>	<b>INTERFÍCIE D'USUARI .....</b>	<b>47</b>
<b>8.</b>	<b>DISSENY ARQUITECTÒNIC.....</b>	<b>49</b>
8.1.	REQUERIMENTS DEL CLIENT.....	49
8.2.	ARQUITECTURA.....	50
8.2.1.	<i>Arquitectura del sistema</i> .....	50
8.2.2.	<i>Arquitectura de la vista</i> .....	51
8.2.3.	<i>Arquitectura del controlador</i> .....	51
8.2.4.	<i>Arquitectura del model</i> .....	52
<b>9.</b>	<b>CONCLUSIONS.....</b>	<b>53</b>
<b>10.</b>	<b>GLOSSARI .....</b>	<b>54</b>
<b>11.</b>	<b>BIBLIOGRAFIA .....</b>	<b>55</b>
11.1.	JAVA I J2EE .....	55
11.2.	JBoss .....	55
<b>12.</b>	<b>ANNEXOS.....</b>	<b>56</b>

### 3. Índex de figures

Figura 1 Vista del subsistema de manteniment de l'estructura .....	16
Figura 2 Vista del subsistema d'aparador virtual .....	17
Figura 3 Vista del subsistema de peticions .....	18
Figura 4 Diagrama estàtic del subsistema de manteniment de l'estructura .....	19
Figura 5 Casos d'ús del subsistema de manteniment de l'estructura .....	20
Figura 6 Diagrama estàtic del subsistema d'aparador virtual .....	29
Figura 7 Casos d'ús del subsistema d'aparador virtual .....	29
Figura 8 Diagrames d'estats del subsistema de peticions .....	35
Figura 9 Casos d'ús del subsistema de peticions .....	36
Figura 10 Diagrama estàtic del subsistema de seguretat .....	41
Figura 11 Casos d'ús del subsistema de seguretat .....	41
Figura 12 Diagrama de paquets del sistema .....	42
Figura 13 Diagrama estàtic del sistema .....	43
Figura 14 Diagrama E-R del sistema .....	44
Figura 15 Captura de pàgines de l'aparador .....	47
Figura 16 Captura de pàgines de peticions, manteniments i seguretat .....	48
Figura 17 Mostra del resultat de FiltreUI .....	48
Figura 18 Representació de l'arquitectura del sistema .....	50

## 4. Introducció

### 4.1. Context

Els propietaris d'una botiga de maquinari informàtic volen que creem un sistema informàtic per tal que els seus clients puguin consultar les característiques i els preus dels seus articles i equips per Internet.

El sistema ha d'oferir un mecanisme senzill i potent per tal que els clients trobin ràpidament allò que busquen, ja sigui a partir de les seves característiques, a partir de la marca, pel tipus de producte, etc. Tot i que no totes les característiques de recerca han d'estar implementades en una primera versió, el sistema ha de facilitar que més endavant se n'implementin de noves.

Quan un client trobi un article o equip que li interessi en podrà consultar les característiques tècniques i afegir-lo al seu carretó. Es podrà consultar el carretó en qualsevol moment i, quan l'usuari hagi triat tot el que li interessa, confirmarà la comanda.

Els diferents responsables i tècnics de la botiga han de poder consultar les noves comandes i les tasques que se'n deriven, així com la seva evolució. Segons la tasca que cada component de l'equip hagi de desenvolupar, s'anirà actualitzant l'estat de la comanda a mesura que es disposi de stock, s'ensambla l'equip, etc.

El sistema es divideix en quatre subsistemes:

- Manteniment de l'estructura  
Els responsables de la botiga configuraran les famílies, característiques, articles, equips, etc. que s'ofereixin a la botiga virtual.
- Aparador virtual  
Els clients de la botiga consultaran els productes disponibles i realitzaran les seves comandes, ja siguin simulades o reals.
- Peticions  
Els encarregats i tècnics corresponents rebran les comandes i les serviran i ensamblaran.
- Seguretat  
Cada component de l'equip tindrà accés a determinades funcionalitats (subsistemes), depenent del seu càrrec.

Actualment, el nostre client no disposa de personal qualificat en Java per a poder realitzar aquest projecte ni tampoc per mantenir-lo, així que ens ha demanat que li fem un prototip que serveixi de *plantilla* per a la creació d'un projecte real. El client està familiaritzat amb UML i RUP i serà l'encarregat de la realització del projecte comercial.

## 4.2. Objectius

L'objectiu final d'aquest treball no és pas la realització d'un projecte comercial, sinó explorar les possibilitats de J2EE per tal que el nostre client conegui les diferents tecnologies de les quals es pot servir per la realització del seu projecte. Més concretament, podríem dir que aquest TFC és un prototip o una plantilla que ha de servir al nostre client per enfocar i desenvolupar el seu projecte.

He volgut concentrar-me únicament en les tecnologies pertanyents a J2EE per tal que sigui molt senzill revisar la feina feta i treure'n les conclusions pertinents. Per exemple, he evitat deliberadament fer servir JavaScript o CSS per enriquir la interfície per tal que quan hom revisi el disseny i el codi del projecte no perdi el temps intentant desxifrar un JavaScript que faci alguna tasca totalment prescindible; donant, per tant, tot el protagonisme a les capacitats de J2EE.

Per tal de tenir una visió mínimament realista de què significa treballar amb J2EE he volgut fer un *enunciat* més aviat extens, que fos representatiu d'un subconjunt de les diverses tasques que hauríem d'afrontar en un projecte real. Per això cada un dels quatre subsistemes representen una situació típica en els projectes comercials:

- Els manteniments són fonamentals en qualsevol aplicació de gestió. Acostumen a ser repetitius i gens interessants per l'equip, però poden complicar-se molt depenent de les exigències del client.
- Les peticions representen les tasques de control i de transformació de dades que acostumen a fer-se portes endins de les organitzacions.
- La botiga virtual és el cas típic de gestió prolongada amb dades volàtils que poden o no esdevenir definitives i que es recolzen en dades que ja són persistents en el nostre sistema. A més, és interessant perquè ha d'oferir un punt de vista del magatzem de dades diferent al que tindria un treballador de l'empresa, concentrant-se més en facilitar que l'usuari faci una determinada cosa, que no pas en donar una visió global i organitzada del magatzem.
- La seguretat també és un cas típic; és interessant proporcionar mecanismes de control d'accessos de propòsit genèric, més que no pas haver fer que el programador faci tasques rutinàries pel control de la seguretat (com ara includes o, encara pitjor, copiar i enganxar el mateix codi aquí i allà), cosa que és propensa a errors.

Com ja he dit, he volgut posar a prova el MVC: vinc de desenvolupar aplicacions Win32 client/servidor amb clients força pesats i amb interfícies molt riques. Amb el temps es fa molt difícil no ser sistemàtic o evitar de dissenyar pensant en la plataforma, perquè hom ja coneix els



problemes amb què es trobarà i això pot acabar influint el disseny, per exemple, dels objectes de negoci. Ara bé, en aquest projecte em trobo en un entorn completament diferent: n-capes i clients molt lleugers basats en web. Si MVC *funciona*, dissenyar la lògica de negoci per una aplicació com aquesta no ha de ser diferent de com ho faria per una aplicació amb una interfície rica o amb uns controladors que funcionin molt diferent dels que estan dissenyats pel web.

Per assegurar que tot això es compleixi, he cregut molt convenient utilitzar EJB amb CMP per evitar els problemes de mantenir la persistència dels objectes de negoci, concentrant-me pràcticament en exclusiva a dissenyar i implementar la lògica de la millor forma que he cregut convenient, cosa que, d'altra banda, és un dels beneficis i objectius fonamentals de la utilització d'aquesta tecnologia.

## 4.3. Enfocament i mètode seguit

### 4.3.1. *Fase 1*

Inicialment es fa la proposta del projecte al consultor presentant-li els requeriments del client en forma d'anàlisi orgànic. Un cop el consultor hagi revisat aquest anàlisi, aconsella els canvis que creu adequats realitzar, que s'incorporen a la proposta inicial i queda tancada.

D'aquesta fase n'ha de resultar els paquets / subsistemes que composaran l'aplicació així com les seves funcionalitats, tot i que no d'una forma detallada.

### 4.3.2. *Fase 2*

La segona fase consisteix en l'anàlisi funcional del projecte. En aquest, cal detallar totes les classes de negoci que participaran en l'aplicació, així com tots els casos d'ús per especificar de forma detallada la funcionalitat del sistema. Per cada cas d'ús, es farà una representació diagramàtica i una descripció textual detallant la funcionalitat de cada cas d'ús i les seves alternatives o excepcions que puguin sorgir durant la seva execució. En cap moment s'ha vist la necessitat de realitzar cap altre diagrama per enriquir l'anàlisi perquè no hi ha cap punt que sigui especialment complicat d'entendre amb els casos d'ús.

Quan s'han decidit les classes que conformaran definitivament el sistema, es farà la transformació d'aquelles que siguin persistents al model Entitat-Relació, en el qual ens hem de basar perquè treballarem amb un SGBDR.

Com en la fase anterior, es va informant periòdicament al consultor del progrés de l'anàlisi, mentre que aquest aconsella els canvis que cregui oportuns, o demana que es justifiquin les decisions que es van prenent.

Finalment, es presenta un prototip del sistema per il·lustrar l'organització i el funcionament dels diferents subsistemes.

El resultat d'aquesta fase són les classes i funcionalitats detallades que componen cada subsistema, el model ER i un prototip del sistema.

### **4.3.3. Fase 3**

En la fase d'implementació sorgeix un problema, que és que difícilment es pugui implementar tots quatre sistemes a temps. Per això, es decideix no implementar el subsistema de manteniment de l'estructura, que és el menys interessant de tots quatre.

El primer pas d'aquesta fase és familiaritzar-se amb EJB, per tant, el primer que es fa és implementar tots els EJB, homes i interfícies locals/remotes, desplegar-los en el servidor d'aplicacions i provar-los mínimament.

El següent pas és implementar les funcionalitats. Primerament, s'implementa el subsistema de l'aparador virtual, que és el més interessant, seguidament el subsistema de peticions i, finalment, el de seguretat, que està estretament vinculat al de manteniment de l'estructura i al de peticions, que són els dos que s'executen portes endins del negoci.

Les funcionalitats es van implementant seguint el prototip del sistema presentat al final de la fase anterior.

S'escull entre utilitzar JSP o Servlets amb un criteri molt senzill:

- Tots aquells casos d'ús que no modifiquin l'estat del sistema, ja sigui volàtil (el carretó de l'aparador virtual) o persistent (qualssevol dada de la base de dades), s'implementarà amb JSP per una qüestió de simplicitat. Un document JSP és fàcil de crear i de mantenir, mentre que una unitat de codi no ho és tant. A més a més, com que no fem interfícies complexes, les JSP seran encara més senzilles.  
D'altra banda, es permet que les JSP consultin el model perquè, si bé això trenca amb el contracte de MVC, no suposa una violació general del model: que la vista accedeixi al model només per llegir, no implica incorporar funcionalitat del controlador en la vista, ni tampoc que la vista modifiqui el model, és a dir, no s'altera la funcionalitat de cap capa del model.
- En els casos en què s'hagi de modificar l'estat del sistema, es farà sempre a través del controlador, ja que només aquest és responsable de modificar l'estat del model del sistema, ja siguin dades volàtils o persistents.

Un avantatge afegit a fer-ho d'aquesta manera és que encapsulem les petites càrregues de codi inherents a les modificacions de dades en un lloc comú: els servlets. Un exemple molt clar de com de pràctic és encapsular el comportament d'una determinada funcionalitat en servlets són aquells que eliminen un objecte del model (un registre de la base de dades): primer fem la petició al servlet; aquest ens redirigeix a una JSP que











ens demana la informació i que rep del propi servlet què ha de fer tant en cas que responguem afirmativament com negativament; finalment, la JSP ens redirigeix al servlet informant-lo que ja hem confirmat l'esborrament.

A mesura que es va progressant en la implementació, es presenta al consultor l'estat del projecte, que aquest revisa i comenta amb l'alumne.











El resultat d'aquesta fase és el codi font del projecte, els descriptors de desplegament i els binaris del sistema.

## 4.4. Planificació

### 4.4.1. *Prevista*

Id		Nombre de tarea	Duración	Comienzo	Fin
1		Presentació del projecte	2 días	lun 22/03/04	mar 23/03/04
2		Casos d'ús	3 días	mié 24/03/04	vie 26/03/04
3		Anàlisi estàtic	2 días	sáb 27/03/04	lun 29/03/04
4		Disseny de la BD	2 días	mar 30/03/04	mié 31/03/04
5		Disseny arquitectònic	8 días	jue 01/04/04	lun 12/04/04
6		Implementació de la segureta	13 días	mar 13/04/04	jue 29/04/04
7		Implementació de l'aparador	8 días	vie 30/04/04	mar 11/05/04
8		Implementació de les peticion	9 días	mié 12/05/04	lun 24/05/04
9		Lliurament de l'aplicació	21 días	mar 25/05/04	mar 22/06/04

### 4.4.2. *Real*

Id		Nombre de tarea	Duración	Comienzo	Fin
1		Presentació del projecte	2 días	lun 22/03/04	mar 23/03/04
2		Casos d'ús	6 días	mié 24/03/04	mar 30/03/04
3		Anàlisi estàtic	4 días	mié 31/03/04	lun 05/04/04
4		Disseny de la BD	1 día	mar 06/04/04	mar 06/04/04
5		Disseny arquitectònic	5 días	mié 07/04/04	mar 13/04/04
6		Implementació de la segureta	9 días	mié 14/04/04	lun 26/04/04
7		Implementació de l'aparador	20 días	mar 27/04/04	lun 24/05/04
8		Implementació de les peticion	7 días	mar 25/05/04	mié 02/06/04
9		Lliurament de l'aplicació	14 días	jue 03/06/04	mar 22/06/04

## 4.5. Productes obtinguts

Podem dir que aquest projecte té dos productes diferents, basant-nos en els seus destinataris:

- Aparador virtual: aquest és l'únic subsistema amb el qual interactuaran els clients de la botiga i el seu objectiu és que aquests consultin els productes de la botiga, puguin simular les seves comandes, fer-les efectives si així ho desitgen i, posteriorment, seguir-ne l'evolució.

Hi podem accedir a través d'una URL del tipus:

[http://\\_un\\_servidor\\_:8080/uocBotiga/aparador/homeAparador.jsp](http://_un_servidor_:8080/uocBotiga/aparador/homeAparador.jsp)

- Intranet: es compon per la resta de subsistemes i ofereix les diferents funcionalitats de control i gestió que utilitzaran els diferents components de l'equip de la botiga.

Els subsistemes que actualment la componen són el seguretat, peticions i manteniment de l'estructura (aquest últim no s'ha implementat).

Hi podem accedir a través d'unes URLs del tipus:

[http://\\_un\\_servidor\\_:8080/uocBotiga/seguretat/homeSeguretat.jsp](http://_un_servidor_:8080/uocBotiga/seguretat/homeSeguretat.jsp)

[http://\\_un\\_servidor\\_:8080/uocBotiga/peticions/homePeticions.jsp](http://_un_servidor_:8080/uocBotiga/peticions/homePeticions.jsp)

[http://\\_un\\_servidor\\_:8080/uocBotiga/manteniment/homeManteniment.htm](http://_un_servidor_:8080/uocBotiga/manteniment/homeManteniment.htm)

## **4.6. Sobre aquesta memòria**

Després d'aquesta breu presentació del projecte, algunes de les seves problemàtiques, la metodologia i algunes de les decisions que s'han pres sobre el camí, en els següents capítols es mostren els diferents resultats obtinguts de les tres fases del projecte: l'anàlisi orgànic en la primera, el funcional en la segona i algunes decisions preses durant la implementació.

Segueixen les conclusions que s'han extret de la realització del projecte, el glossari que utilitzarem per comunicar-nos amb el client, la bibliografia consultada per a la realització del projecte i, finalment, els annexos a aquesta memòria.

## 5. Anàlisi orgànic

### 5.1. Subsistema de manteniment de l'estructura

Aquest subsistema serà únicament accessible pel rol d'usuari VENEDOR. Permetrà la introducció d'articles i equips, juntament amb tota la informació relacionada necessària pel seu funcionament, els nostres clients, etc., com es veu en el següent diagrama.

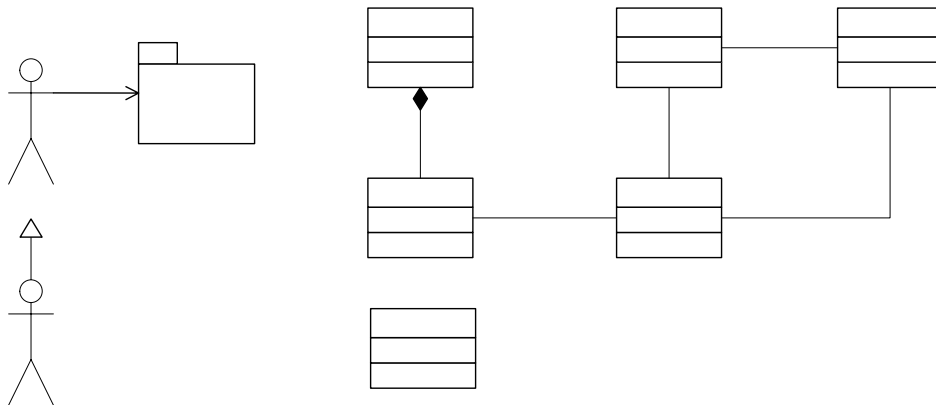


Figura 1 Vista del subsistema de manteniment de l'estructura

### 5.2. Subsistema d'aparador virtual

Els usuaris podran consultar els articles i equips disponibles i realitzar una comanda. La comanda no serà efectiva fins que l'usuari no indiqui que és definitiva i un cop introduïdes les dades de pagament que el subsistema li demanarà.

Per tal de poder fer una compra real, l'usuari ha d'existir en el sistema. Tindrem un link a tal efecte que li demanarà les dades que calgui, que seran validades en les oficines de la botiga. El client podrà realitzar la compra en el mateix moment, però abans de servir-se s'hauran de validar les seves dades (subsistema de peticions).

Només el rol d'usuari CLIENT tindrà accés al subsistema.



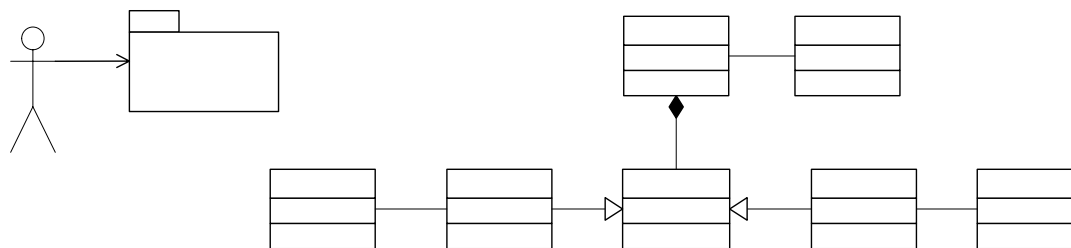


Figura 2 Vista del subsistema d'aparador virtual

### 5.3. Subsistema de peticions

## AparadorVirtual

El RESPONSABLE DE VENDES s'encarrega de validar les dades dels clients que es donin d'alta en l'aparador virtual i de validar les compres que hi realitzin.

Un cop validada una comanda, cada un dels seus ítems passa de forma individual pel RESPONSABLE D'EXISTÈNCIES, que o bé bloquejarà un ítem determinat fins que hi hagi stock suficient, o permetrà que se serveixi o ensamblí depenent de si és un article o un equip, respectivament.

Si es tracta d'un equip, l'ítem passa a l'estat d'ensamblatge i, un cop finalitzat, passa a l'estat de servit; únicament el RESPONSABLE DE MUNTATGE podrà realitzar el canvi d'estat. Els articles passen directament a aquest estat, sense passar per ensamblatge.

Una comanda també passa per diversos estats. Inicialment, pendent, fins que no és validada pel RESPONSABLE DE VENDES, passant a estar en procés. Quan tots els seus ítems són servits, passa a l'estat servida, moment en què enviarem una notificació per e-mail al client perquè vingui a recollir la seva comanda. Òbviament, cap ítem de la comanda pot canviar d'estat fins que la comanda no passi a estar en procés.

Aquests són els diagrames d'estat corresponents.

Client

A

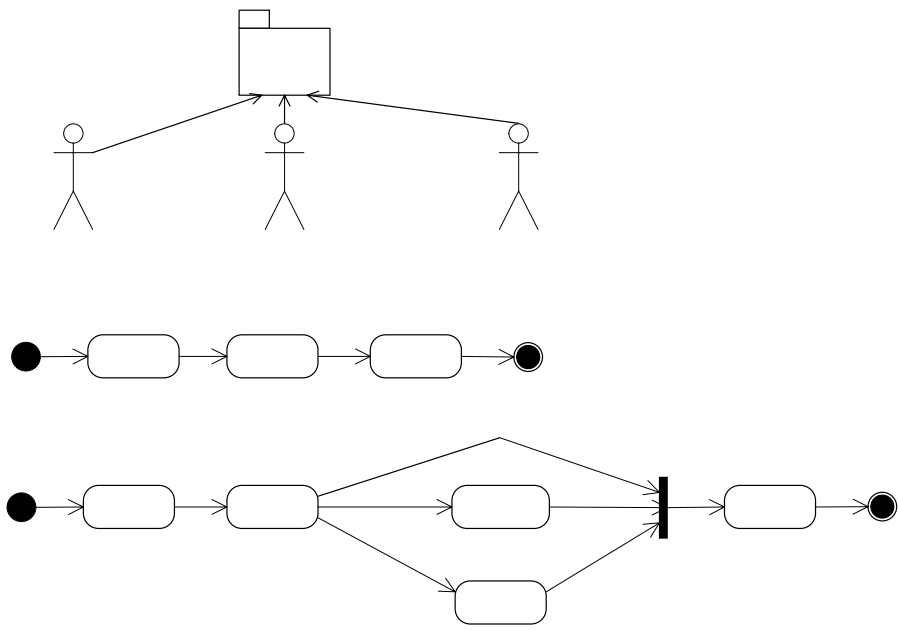


Figura 3 Vista del subsistema de peticions

### 5.4. Subsistema de seguretat

Permet configurar els usuaris que tenen accés al sistema i configurar els seus privilegis a través del seu perfil d'usuari, que pot ser TREBALLADOR / RESPONSABLE / ADMINISTRADOR.

Únicament els usuaris amb privilegi d'Administrador poden accedir a aquest subsistema.

El subsistema ha de garantir que sempre hi hagi almenys un usuari amb privilegi d'Administrador.

Responsible Existències

ResponsibleM

## Estats d'una comanda

Pendent

EnProcés

## Estats d'un ítem

Pendent

Acceptat

## 6. Anàlisi funcional<sup>1</sup>

### 6.1. Subsistema de manteniment de l'estructura

En aquest subsistema els encarregats de la botiga virtual podran introduir les dades bàsiques de l'aplicació, els articles, els equips, mantenir els clients, etc.

Poden accedir a aquest subsistema els usuaris amb perfil Treballador i Responsable.

#### 6.1.1. Diagrama estàtic

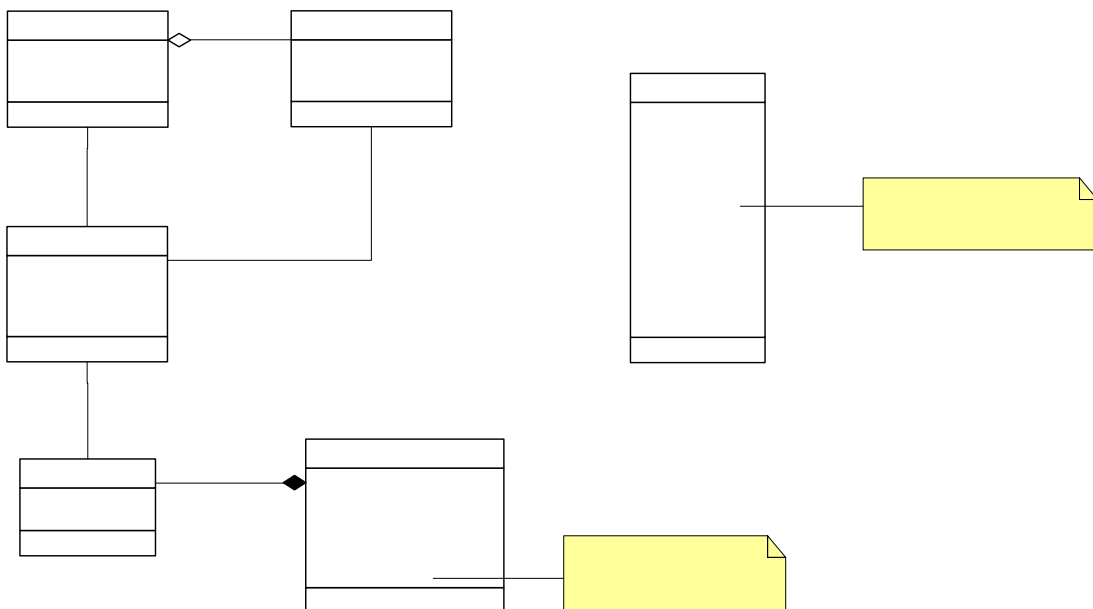


Figura 4 Diagrama estàtic del subsistema de manteniment de l'estructura

---

<sup>1</sup> Les classes de control i presentació no es detallen per tractar-se tansols d'intermediaris i per la seva funcionalitat limitada. Cada cas d'ús tindrà associada una classe de presentació i una classe de control en cas de ser necessari, generalment quan s'hagin de manipular dades.

### 6.1.2. Casos d'ús

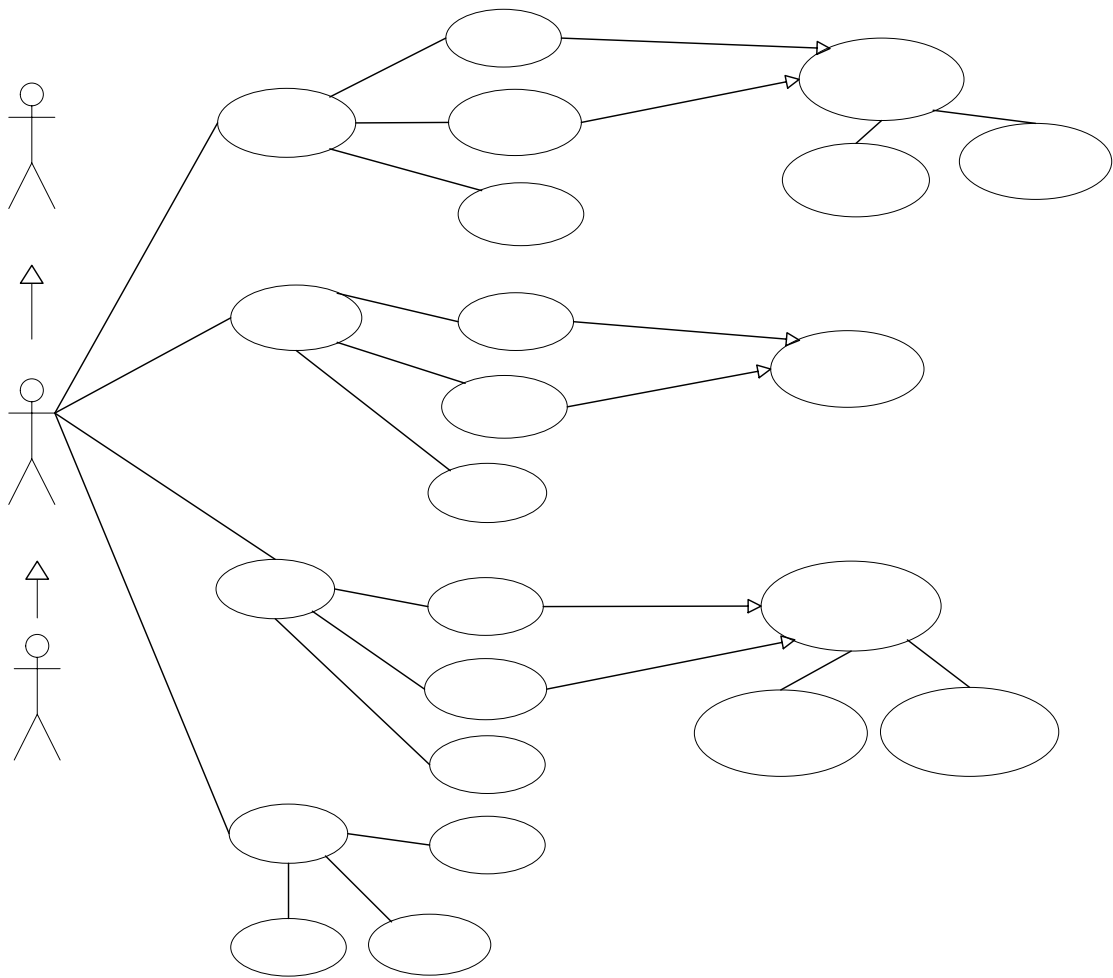


Figura 5 Casos d'ús del subsistema de manteniment de l'estructura

### 6.1.3. Descripció textual dels Casos d'ús

#### CrearFamilia

Funcionalitat: permet donar d'alta una família en el sistema.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: MantenirAtributsFamilia

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: la família s'ha creat i és única en el sistema.

Descripció:

1. es presenta el formulari perquè l'usuari introdueixi les dades de la família
2. l'usuari proporciona les dades de la família
3. el sistema emmagatema les dades

Alternatives i excepcions:

- 3.1. si la família ja existia en el sistema, es notifica a l'usuari i es torna al punt 1
- 3.2. l'usuari accedeix al manteniment dels atributs de la família

#### ModificarFamília

Funcionalitat: permet modificar les dades d'una família existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: MantenirAtributsFamília

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: la família s'ha modificat en el sistema.

Descripció:

1. el sistema demana el codi de la família que es vol modificar
2. l'usuari proporciona el codi de la família
3. es presenten les dades de la família a l'usuari
4. l'usuari modifica les dades
5. el sistema emmagatzema les noves dades de la família

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba la família, es notifica a l'usuari i es torna al punt 1
- 5.1 l'usuari accedeix al manteniment dels atributs de la família

#### EliminarFamília

Funcionalitat: permet esborrar una família existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: la família s'ha eliminat del sistema.

Descripció:

1. el sistema demana el codi de la família que es vol eliminar
2. l'usuari proporciona el codi de la família
3. es presenten les dades de la família i es demana confirmació a l'usuari
4. l'usuari verifica que és la família que vol i confirma
5. s'elimina la família del sistema

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba la família, es notifica a l'usuari i es torna al punt 1

## 5.1 la família té atributs o articles relacionats i no es pot eliminar

### AfegirAtributFamilia

Funcionalitat: permet afegir un atribut a una família ja existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: l'atribut s'ha creat i és única en el sistema.

Descripció:

1. es presenta el formulari perquè l'usuari introdueixi les dades de l'atribut
2. l'usuari proporciona les dades de l'atribut
3. el sistema emmagatema les dades

Alternatives i excepcions:

- 3.1. si l'atribut ja existia en el sistema, es notifica a l'usuari i es torna al punt 1

### TreureAtributFamilia

Funcionalitat: permet treure un atribut a una família ja existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: l'atribut s'ha eliminat del sistema.

Descripció:

1. el sistema demana el codi de l'atribut que es vol eliminar
2. l'usuari proporciona el codi de l'atribut
3. es presenten les dades de l'atribut i es demana confirmació a l'usuari
4. l'usuari verifica que és l'atribut que vol i confirma
5. s'elimina l'atribut del sistema

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba l'atribut, es notifica a l'usuari i es torna al punt 1
- 5.1 l'atribut té articles relacionats i no es pot eliminar

### CrearArticle

Funcionalitat: permet donar d'alta un article en el sistema.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: MantenirAtributsArticle

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: l'article s'ha creat i és únic en el sistema.

Descripció:

1. es presenta el formulari perquè l'usuari introdueixi les dades de l'article
2. l'usuari proporciona les dades de l'article
3. el sistema emmagatzema les dades

Alternatives i excepcions:

- 3.1. si l'article ja existia en el sistema, es notifica a l'usuari i es torna al punt 1
- 3.2 l'usuari accedeix al manteniment dels atributs de l'article

### ModificarArticle

Funcionalitat: permet modificar les dades d'un article existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: MantenirAtributsArticle

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: l'article s'ha modificat en el sistema.

Descripció:

1. el sistema demana el codi de l'article que es vol modificar
2. l'usuari proporciona el codi de l'article
3. es presenten les dades de l'article a l'usuari
4. l'usuari modifica les dades
5. el sistema emmagatzema les noves dades de l'article

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba l'article, es notifica a l'usuari i es torna al punt 1
- 5.1 l'usuari accedeix al manteniment dels atributs de l'article

### EliminarArticle

Funcionalitat: permet esborrar un article existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: l'article s'ha eliminat del sistema.

Descripció:

1. el sistema demana el codi de l'article que es vol eliminar

2. l'usuari proporciona el codi de l'article
3. es presenten les dades de l'article i es demana confirmació a l'usuari
4. l'usuari verifica que és l'article que vol i confirma
5. s'elimina l'article del sistema

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba l'article, es notifica a l'usuari i es torna al punt 1
- 5.1 l'article té atributs relacionats i no es pot eliminar

#### MantenirAtributsArticle

Funcionalitat: permet configurar els atributs que compleix un article dels disponibles per la família a la qual pertany.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; el sistema disposa de l'article amb què estem treballant i aquest té una família.

Postcondició: s'han afegit / eliminat els atributs d'un article.

Descripció:

1. el sistema presenta els atributs disponibles per la família de l'article
2. l'usuari marca / desmarca els atributs apropiats
3. el sistema configura els atributs de l'article deixant en la BD només aquells que l'usuari hagi deixat marcats.

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment

#### CrearEquip

Funcionalitat: permet donar d'alta un equip en el sistema.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: MantenirComponentsEquip

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: l'equip s'ha creat i és únic en el sistema.

Descripció:

1. es presenta el formulari perquè l'usuari introdueixi les dades de l'equip
2. l'usuari proporciona les dades de l'equip
3. el sistema emmagatema les dades

Alternatives i excepcions:



- 3.1. si l'equip ja existia en el sistema, es notifica a l'usuari i es torna al punt 1
- 3.2. l'usuari accedeix al manteniment dels components de l'equip

#### ModificarEquip

Funcionalitat: permet modificar les dades d'un equip existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: ManténirComponentsEquip

Precondició: l'usuari s'ha autènticat en el sistema i té els drets d'accés requerits.

Postcondició: l'equip s'ha modificat en el sistema.

Descripció:

1. el sistema demana el codi de l'equip que es vol modificar
2. l'usuari proporciona el codi de l'equip
3. es presenten les dades de l'equip a l'usuari
4. l'usuari modifica les dades
5. el sistema emmagatzema les noves dades de l'equip

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba l'equip, es notifica a l'usuari i es torna al punt 1
- 5.1 l'usuari accedeix al manteniment dels components de l'equip

#### EliminarEquip

Funcionalitat: permet esborrar un equip existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autènticat en el sistema i té els drets d'accés requerits.

Postcondició: l'equip s'ha marcat com a no-actiu.

Descripció:

1. el sistema demana el codi de l'equip que es vol eliminar
2. l'usuari proporciona el codi de l'equip
3. es presenten les dades de l'equip i es demana confirmació a l'usuari
4. l'usuari verifica que és l'equip que vol i confirma
5. el sistema marca l'equip com a no-actiu.

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba l'equip, es notifica a l'usuari i es torna al punt 1

### AfegirComponentEquip

Funcionalitat: permet afegir un component a una equip ja existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: s'ha afegit el component a l'equip.

Descripció:

1. es presenta el formulari perquè l'usuari introdueixi el codi del component.
2. l'usuari proporciona el codi de l'article
3. es presenten les dades de l'article i es demanen les dades del component
4. l'usuari introdueix les dades del component
5. el sistema vincula el component a l'equip.

Alternatives i excepcions:

- 3.1. si el component ja pertanyia a l'equip, es notifica a l'usuari i es torna al punt 1

### TreureComponentEquip

Funcionalitat: permet treure un article com a component d'un equip ja existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: s'ha eliminat el component.

Descripció:

1. el sistema mostra els components que pertanyen a l'equip
2. l'usuari selecciona el que vol eliminar
3. s'elimina el component del sistema

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment

### CrearClient

Funcionalitat: permet donar d'alta un client en el sistema.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: el client s'ha creat i és únic en el sistema.

Descripció:

1. es presenta el formulari perquè l'usuari introdueixi les dades del client
2. l'usuari proporciona les dades del client
3. el sistema emmagatema les dades

Alternatives i excepcions:

- 3.1. si el client ja existia en el sistema, es notifica a l'usuari i es torna al punt 1

#### ModificarClient

Funcionalitat: permet modificar les dades d'un client existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: el client s'ha modificat en el sistema.

Descripció:

1. el sistema demana el codi del client que es vol modificar
2. l'usuari proporciona el codi del client
3. es presenten les dades del client a l'usuari
4. l'usuari modifica les dades
5. el sistema emmagatzema les noves dades del client

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment
- 1.2 el sistema no troba el client, es notifica a l'usuari i es torna al punt 1

#### EliminarClient

Funcionalitat: permet esborrar un client existent.

Paper dins el treball de l'usuari: empleat/responsable

Actors: Venedor / ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: el client s'ha marcat com a no-actiu.

Descripció:

1. el sistema demana el codi del client que es vol eliminar
2. l'usuari proporciona el codi del client
3. es presenten les dades del client i es demana confirmació a l'usuari
4. l'usuari verifica que és el client que vol i confirma
5. el sistema marca el client com a no-actiu.

Alternatives i excepcions:

- 1.1 l'usuari pot cancel·lar en qualsevol moment

1.2 el sistema no troba el client, es notifica a l'usuari i es torna al punt 1

## 6.2. Subsistema d'aparador virtual

Aquest subsistema és el que els clients de la botiga utilitzaran per consultar els articles i equips, simular comandes, realitzar-les i revisar el seu carretó de la compra.

Cada cop que un usuari entra en el subsistema, se li assigna de forma automàtica una sessió de compra.

### 6.2.1. Diagrama estàtic

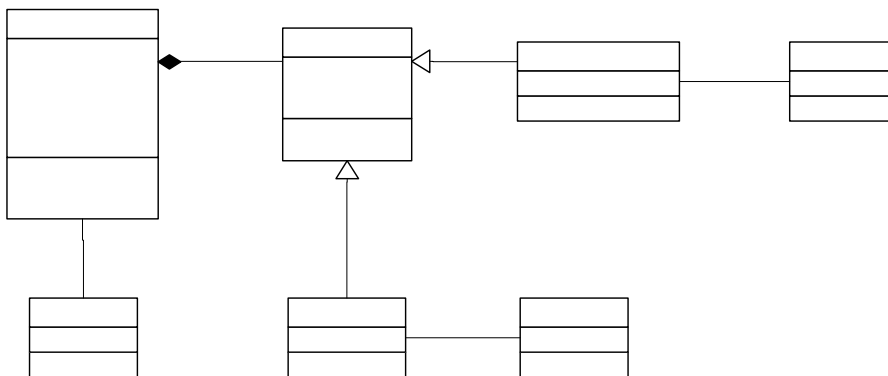
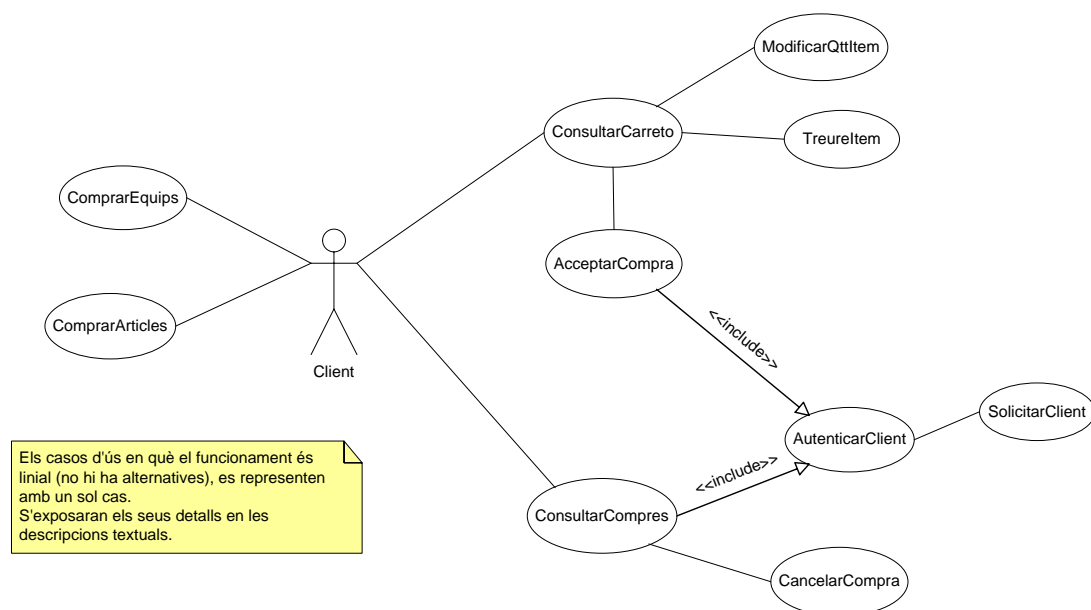


Figura 6 Diagrama estàtic del subsistema d'aparador virtual

### 6.2.2. Casos d'ús



Els casos d'ús en què el funcionament és linial (no hi ha alternatives), es representen amb un sol cas. S'exposaran els seus detalls en les descripcions textuales.

Figura 7 Casos d'ús del subsistema d'aparador virtual

## Comanda

+codi : int

+dataAlta : Date

+preu : float

+preuCalculat : float

1

1..\*

It

+ur

+pr

Lo

### **6.2.3. Descripció textual dels casos d'ús**

#### ComprarArticles

Funcionalitat: permet que el client compri un article dels que ofereix la botiga.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: --

Precondició: el client té una sessió de compra activa.

Postcondició: el client ha comprat un article (la compra encara no és definitiva).

Descripció:

1. es mostren totes les famílies disponibles en el sistema
2. l'usuari selecciona una família
3. es mostren els articles que pertanyen a la família seleccionada
4. l'usuari selecciona un article
5. es mostren les dades de l'article
6. l'usuari clica el link per comprar aquell article
7. es presenta la informació reduïda de l'article i es permet a l'usuari indicar quantes unitats desitja comprar
8. l'usuari introdueix les unitats
9. es presenta el preu
10. l'usuari accepta la compra
11. l'article i les unitats indicades s'afageixen a la sessió de compra

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 9.1. el nombre d'unitats no és acceptable<sup>2</sup>, es torna al punt 8
- 11.1. l'usuari ja havia afegit aquell article a la sessió de compra, es torna al punt 3

#### ComprarEquips

Funcionalitat: permet que el client compri un equip dels que ofereix la botiga.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: --

Precondició: el client té una sessió de compra activa.

Postcondició: el client ha comprat un equip (la compra encara no és definitiva).

---

<sup>2</sup> P.ex., un nombre d'unitats menor que 1, amb decimals, molt grans, etc.

Descripció:

1. es mostren els equips que ofereix la botiga
2. l'usuari selecciona un equip
3. es mostren les dades de l'equip
4. l'usuari clica el link per comprar aquell equip
5. es presenta la informació reduïda de l'equip i es permet a l'usuari indicar quantes unitats desitja comprar
6. l'usuari introdueix les unitats
7. es presenta el preu
8. l'usuari accepta la compra
9. l'equip i les unitats indicades s'afageixen a la sessió de compra

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 7.1. el nombre d'unitats no és acceptable, es torna al punt 6
- 9.1. l'usuari ja havia afegit aquell equip a la sessió de compra, es torna al punt 1

#### ConsultarCarreto

Funcionalitat: mostra al client un resum de la sessió de compra i li permet modificar-la.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: ModificarQtItem, TreureItem, AcceptarCompra

Precondició: el client té una sessió de compra activa.

Postcondició: s'ha actualitzat la sessió de compra (la compra encara no és definitiva).

Descripció:

1. es mostra a l'usuari un resum dels articles i equips que ha adquirit durant la sessió de compra

Alternatives i excepcions:

- 1.1. l'usuari clica a canviar la quantitat d'un article (ModificarQtItem)
- 1.2. l'usuari clica a treure un article o equip (TreureItem)
- 1.3. l'usuari clica a acceptar la compra (AcceptarCompra)

#### ModificarQtItem

Funcionalitat: permet que l'usuari canviï el número d'unitats d'un article o equip.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: --

Precondició: el client té una sessió de compra activa.

Postcondició: s'ha actualitzat el nombre d'unitats de l'article o equip (la compra encara no és definitiva).

Descripció:

1. es presenta la informació abreujada de l'ítem, la quantitat i preu actual
2. l'usuari introdueix la nova quantitat
3. es mostra el nou preu de l'ítem
4. l'usuari confirma el canvi
5. el canvi queda enregistrat en la sessió de compra

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 3.1. el nombre d'unitats no és acceptable, es torna al punt 1

#### TreureItem

Funcionalitat: permet que l'usuari tregui un ítem de la sessió de compra.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: --

Precondició: el client té una sessió de compra activa.

Postcondició: s'ha tret l'ítem de la sessió de compra (la compra encara no és definitiva).

Descripció:

1. es presenta la informació abreujada de l'ítem, la quantitat i preu actual, demanant confirmació
2. l'usuari confirma l'acció
3. l'ítem es treu de la sessió de compra

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment

#### AcceptarCompra

Funcionalitat: dona la compra per definitiva i finalitza la sessió de compra.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: AutenticarClient

Precondició: el client té una sessió de compra activa.

Postcondició: s'ha realitzat la comanda i s'ha finalitzat la sessió de compra.

Descripció:

1. es presenta un resum de la sessió de compra
2. l'usuari accepta la compra
3. la comanda queda enregistrada en el sistema.



Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 2.1. és necessari que el client s'autentiqui (AutenticarClient)

#### AutenticarClient

Funcionalitat: demana el nom d'usuari i contrassenya de client

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: SollicitarClient

Precondició: el client té una sessió de compra activa.

Postcondició: el client s'ha autenticat en la sessió de compra.

Descripció:

1. es presenta un formulari demanant nom d'usuari i contrassenya
2. l'usuari introdueix les dades
3. el sistema valida que les dades siguin correctes

Alternatives i excepcions

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 3.1. el client no existeix en el sistema o les dades no són correctes, es torna al punt 1.

#### SollicitarClient

Funcionalitat: permet l'alta d'un usuari com a client del sistema.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: --

Precondició: el client té una sessió de compra activa.

Postcondició: el client s'ha creat en el sistema, però encara no s'ha validat.

Descripció:

1. es demanen les dades necessàries per l'alta del client
2. l'usuari introdueix les seves dades
3. el sistema enregistra el nou client

Alternatives i excepcions

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 3.1. el client ja existeix en el sistema, es torna al punt 1

#### ConsultarCompres

Funcionalitat: permet que un client existent revisi les seves compres.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: AutenticarClient, CancelarCompra

Precondició: el client té una sessió de compra activa i s'ha autenticat en el sistema

Postcondició: --

Descripció:

1. es presenten les comandes realitzades pel client amb el seu estat

Alternatives i excepcions

- 1.1. l'usuari clica a anular una compra pendent (CancelarCompra)

### CancelarCompra

Funcionalitat: permet que un client cancel·li una compra que encara està pendent.

Paper dins el treball de l'usuari: client

Actors: Client

Casos d'ús relacionats: --

Precondició: la compra existeix i encara està pendent

Postcondició: s'ha eliminat la compra del sistema.

Descripció:

1. es presenta un resum de la compra, demanant confirmació
2. l'usuari confirma l'acció
3. s'elimina la comanda del sistema.

Alternatives i excepcions

- 1.1. l'usuari clica a anular una compra pendent (CancelarCompra)
- 3.1. la comanda està en un estat diferent de Pendent, es cancel·la el procés.

### 6.3. Subsistema de peticions

En aquest subsistema els responsables de la botiga validaran i serviran les comandes realitzades pels seus clients en l'aparador virtual.

En aquest subsistema no apareix cap classe nova, per això no es fa diagrama estàtic. Poden accedir a aquest subsistema els usuaris amb perfil Responsable.

#### 6.3.1. Diagrames d'estats

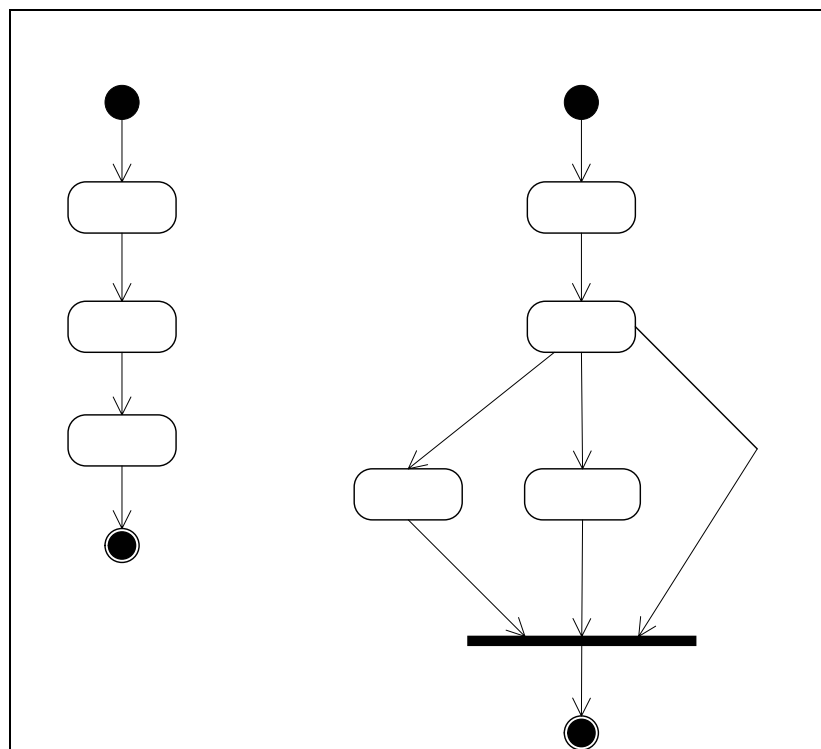


Figura 8 Diagrames d'estats del subsistema de peticions

### 6.3.2. Casos d'ús

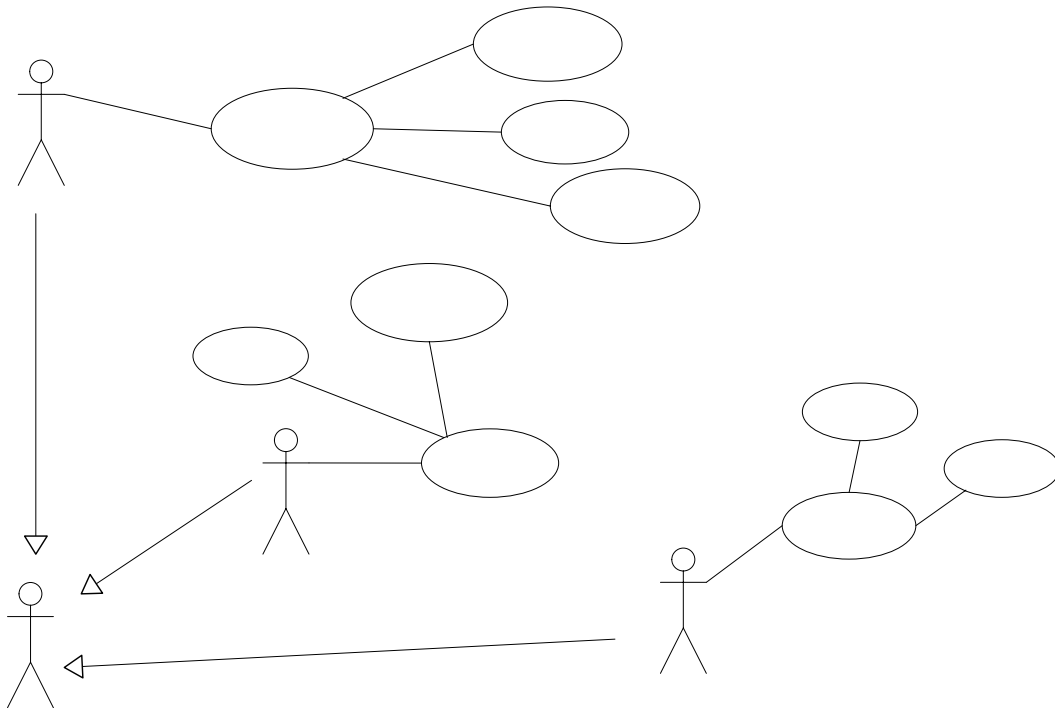


Figura 9 Casos d'ús del subsistema de peticions

### 6.3.3. Descripció textual dels casos d'ús

#### ConsultarComandes

Funcionalitat: presenta a l'usuari les comandes del sistema per tal que en pugui realitzar el manteniment.

#### **ResponsableVendes**

Paper dins el treball de l'usuari: responsable

Actors: ResponsableVendes

Casos d'ús relacionats: AcceptarComanda, ServirComanda, CancelarComanda

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: --

Descripció:

1. el sistema presenta les comandes del sistema amb el seu estat, i el detall dels ítems amb els seus estats corresponents.

Alternatives i excepcions:

- 1.1. l'usuari clica a acceptar una comanda
- 1.2. l'usuari clica a servir una comanda
- 1.3. l'usuari clica a cancel·lar una comanda

Consu

ServirAr

### AcceptarComanda

Funcionalitat: permet acceptar una comanda existent, que passarà a estar EnProces.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; la comanda existeix i està Pendent.

Postcondició: la comanda s'actualitza com a EnProces

Descripció:

1. es presenta les dades de la comanda a mode de confirmació
2. l'usuari confirma l'acció
3. s'actualitza l'estat de la comanda i dels seus ítems.

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 3.1. la comanda no compleix els requisits i no es pot acceptar (client no validat)

### ServirComanda

Funcionalitat: permet servir una comanda existent.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; la comanda existeix i està EnProcés; tots els ítems de la comanda estan servits.

Postcondició: la comanda s'actualitza com a Servida.

Descripció:

1. es presenta les dades de la comanda a mode de confirmació
2. l'usuari accepta l'acció
3. la comanda s'actualitza a Servida i s'envia una notificació via e-mail al client.

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 3.1. la comanda no compleix els requisits i no es pot servir (ítems no servits)

### CancelarComanda

Funcionalitat: permet eliminar una comanda pendent del sistema.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableVendes

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; la comanda existeix i està Pendent.

Postcondició: la comanda s'elimina del sistema.

Descripció:

1. el sistema presenta les dades de la comanda a mode de confirmació
2. l'usuari accepta l'acció
3. la comanda s'elimina del sistema.

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment
- 3.1. la comanda no compleix els requisits i no es pot cancel·lar (estat no pendent).

### ConsultarArticles

Funcionalitat: presenta a l'usuari els ítems de tipus article del sistema per tal que en pugui realitzar el manteniment.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableExistencies

Casos d'ús relacionats: ServirArticle, EsperarStockArticle

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: --

Descripció:

1. el sistema presenta els ítems de comanda de tipus article amb el seu estat.

Alternatives i excepcions:

- 1.1. l'usuari clica a servir un article
- 1.2. l'usuari clica a esperar stock d'un article

### ServirArticle

Funcionalitat: canvia l'estat d'un article a Servit.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableExistencies

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; l'ítem article està en un estat compatible amb Servir.

Postcondició: l'article es marca com a Servit.

Descripció:

1. es presenta el resum de l'ítem article
2. l'usuari confirma l'acció
3. s'actualitza l'ítem com a Servit.

Alternatives i excepcions:

1.1. l'usuari pot cancel·lar en qualsevol moment

EsperarStockArticle

Funcionalitat: canvia l'estat d'un article a PendentStock.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableExistencies

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; l'ítem article està en estat Acceptat.

Postcondició: l'article es marca com a PendentStock.

Descripció:

1. es presenta el resum de l'ítem article
2. l'usuari confirma l'acció
3. s'actualitza l'ítem com a PendentStock.

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment

ConsultarEquips

Funcionalitat: presenta a l'usuari els ítems de tipus equip del sistema per tal que en pugui realitzar el manteniment.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableMuntatge

Casos d'ús relacionats: MuntarEquip, ServirEquip

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits.

Postcondició: --

Descripció:

1. el sistema presenta els ítems de comanda de tipus equip amb el seu estat.

Alternatives i excepcions:

- 1.1. l'usuari clica a muntar un equip
- 1.2. l'usuari clica a servir un equip

MuntarEquip

Funcionalitat: canvia l'estat d'un equip a EnMuntatge.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableMuntatge

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; l'ítem equip està en estat Acceptat.

Postcondició: l'equip es marca com a EnMuntatge.

Descripció:

1. es presenta el resum de l'ítem equip
2. l'usuari confirma l'acció
3. s'actualitza l'ítem com a EnMuntatge.

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment

#### ServirEquip

Funcionalitat: canvia l'estat d'un equip a Servit.

Paper dins el treball de l'usuari: responsable

Actors: ResponsableMuntatge

Casos d'ús relacionats: --

Precondició: l'usuari s'ha autenticat en el sistema i té els drets d'accés requerits; l'ítem equip està en estat EnMuntatge.

Postcondició: l'equip es marca com a Servit.

Descripció:

4. es presenta el resum de l'ítem equip
5. l'usuari confirma l'acció
6. s'actualitza l'ítem com a Servit.

Alternatives i excepcions:

- 1.1. l'usuari pot cancel·lar en qualsevol moment



## 6.4. Subsistema de seguretat

Permet configurar els usuaris que tenen accés al sistema i configurar els seus privilegis a través del seu perfil d'usuari, que pot ser TREBALLADOR / RESPONSABLE / ADMINISTRADOR.

Poden accedir a aquest subsistema els usuaris amb perfil Administrador.

### 6.4.1. Diagrama estàtic

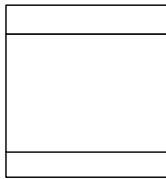


Figura 10 Diagrama estàtic del subsistema de seguretat

### 6.4.2. Casos d'ús

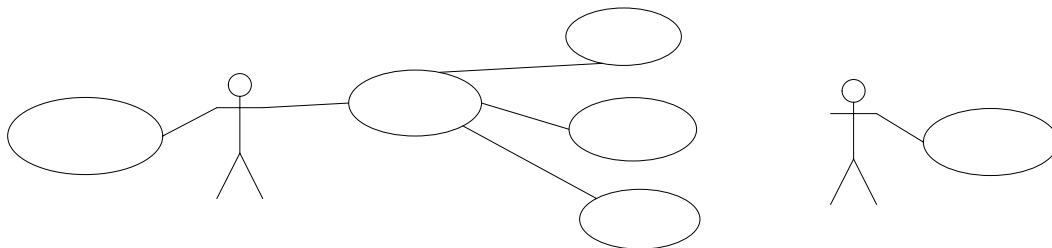


Figura 11 Casos d'ús del subsistema de seguretat

### 6.4.3. Descripció textual dels casos d'ús

Els casos d'ús del manteniment d'usuaris són equivalents als de qualsevol del subsistema de manteniment de l'estructura. L'única particularitat rau en EliminarUsuari, que marca els usuaris com a *bloquejats* enlloc d'esborrar-los del sistema.

El cas d'ús CrearAdministrador únicament verifica que hi hagi algun usuari amb el perfil Administrador i, de no ser així, el crea. Aquest cas d'ús s'executarà cada cop que s'entri en el sistema.

El cas d'ús AutenticarUsuari demanarà login i password per poder entrar a qualsevol part del sistema, excepte l'aparador virtual.

## Usuari

+codi : int  
+login : string  
+password : string  
+nomCompleert : string  
+perfil : int  
+bloquejat : bool

## 6.5. Paquets del sistema

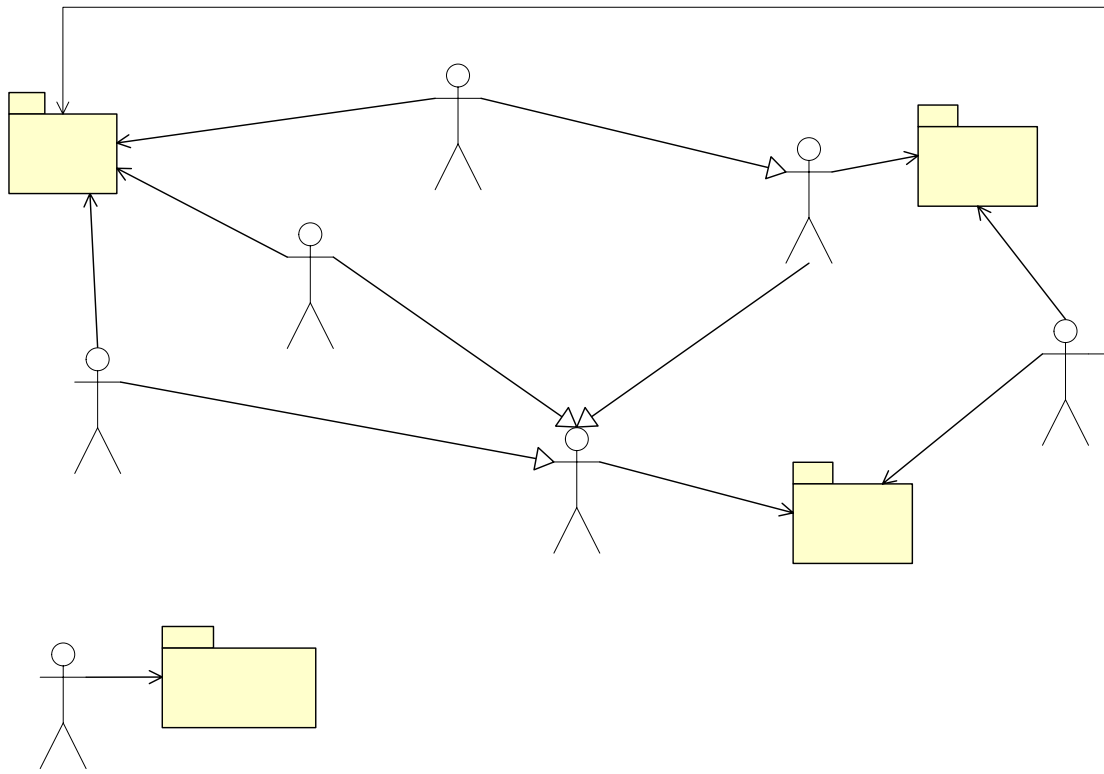


Figura 12 Diagrama de paquets del sistema

## Peticions

## 6.6. Diagrama estàtic

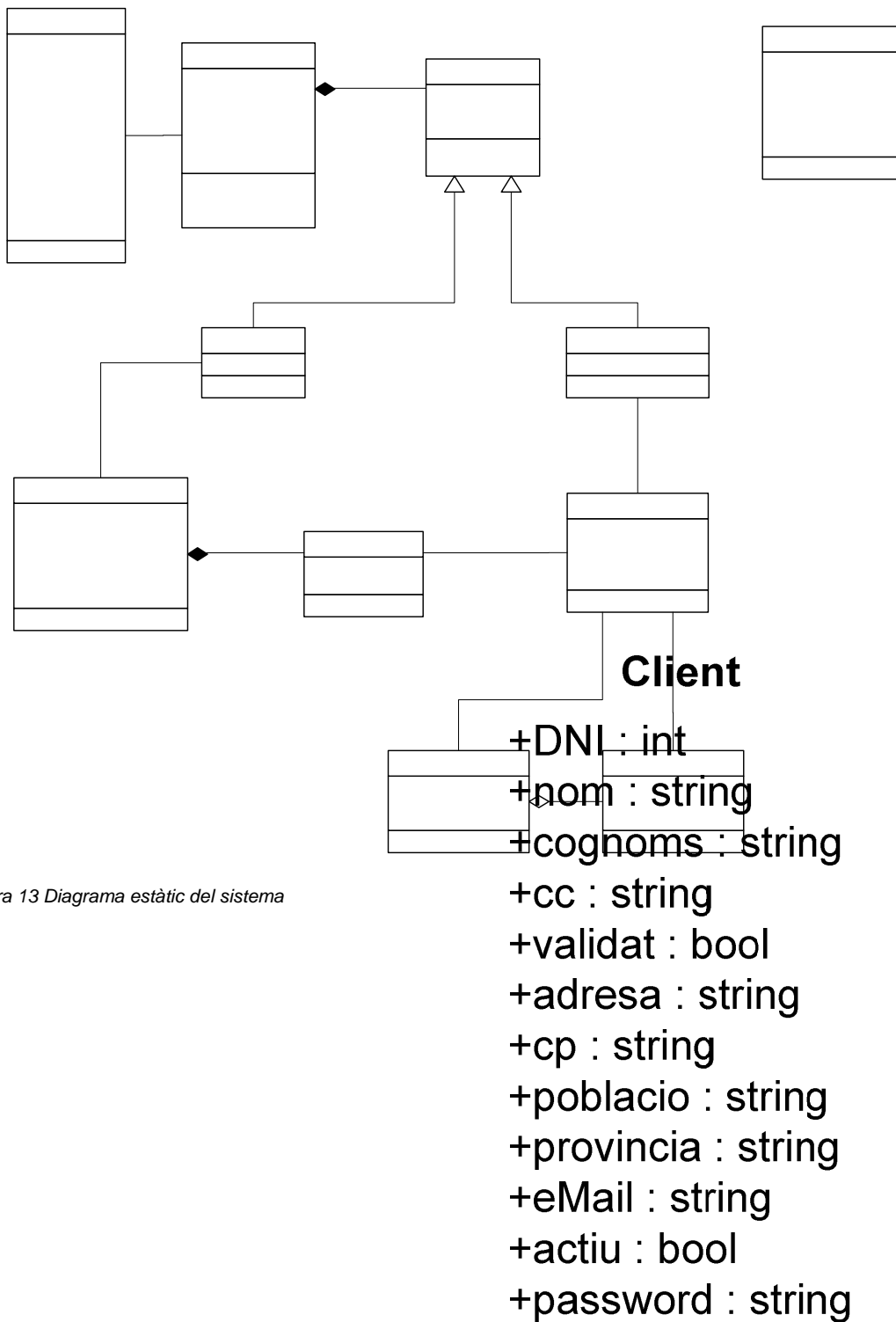


Figura 13 Diagrama estàtic del sistema

## 6.7. Persistència

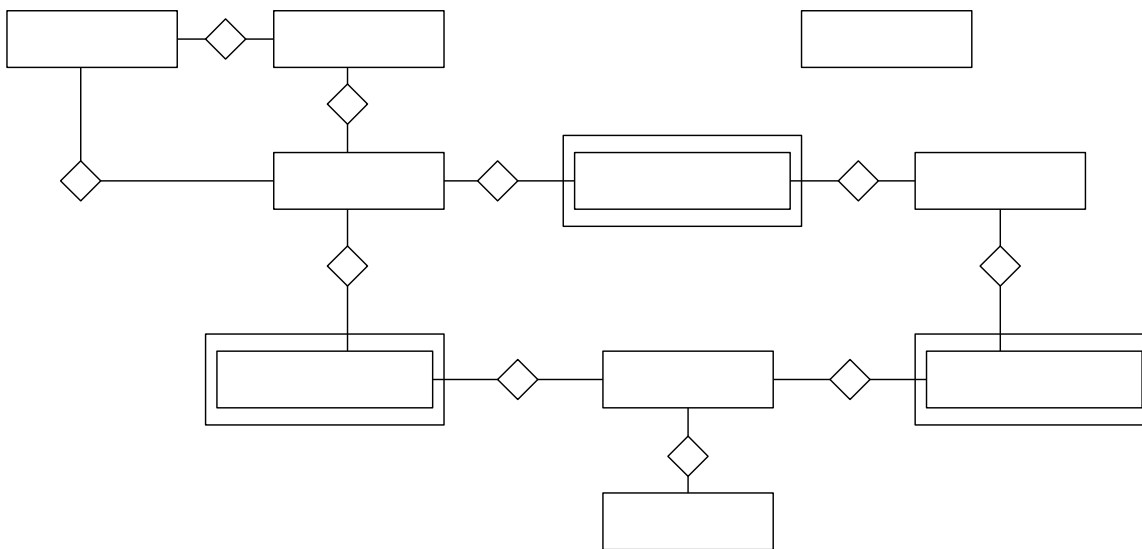


Figura 14 Diagrama E-R del sistema

### 6.7.1. **FAMILIES**

Codi: int

Descripció: str

Observacions: str

# FAMILIES

1

\*

1

### 6.7.2. **ARTICLES**

Codi: int

Descripció: str

Observacions: str

CodFamilia: int (FK → FAMILIES)

\*

### 6.7.3. **COMANDES**

Codi: int

Descripció: str

PreuVenda: currency

Observacions: str

CodFamilia: int (FK → FAMILIES)

#### **6.7.4.      *ATRIBUTSARTICLE***

CodiArticle: int (FK → ARTICLES)

CodiAtribut: int (FK → ATRIBUTS)

#### **6.7.5.      *EQUIPS***

Codi: int

Descripcio: str

Observacions: str

PreuVenda: currency

Actiu: bool

#### **6.7.6.      *COMPONENTS***

CodiEquip: int (FK → EQUIPS)

CodiArticle: int (FK → ARTICLES)

Unitats: int

PreuVenda: currency

#### **6.7.7.      *COMANDES***

Codi: int

DataAlta: date

Preu: currency

Estat: int

DataServida: date

DNIClient: int (FK → CLIENTS)

#### **6.7.8.      *ARTICLESCOMANDA***

CodComanda: int (FK → COMANDES)

CodArticle: int (FK → ARTICLES)

Unitats: int

Preu: currency

Estat: int

### **6.7.9. EQUIPSCOMANDA**

CodComanda: int (FK → COMANDES)

CodEquip: int (FK → EQUIPS)

Unitats: int

Preu: currency

Estat: int

### **6.7.10. CLIENTS**

DNI: int

Nom: str

Cognoms: str

CC: str

Validat: bool

Adresa: str

CP: str

Poblacio: str

Provincia: str

eMail: str

Actiu: bool

Password: str

### **6.7.11. USUARIS**

Codj: int

Login: str

Password: str

NomComplert: str

Perfil: int

Bloquejat: bool

## 7. Interfície d'usuari

Per les característiques d'aquest projecte, l'aspecte visual de la interfície d'usuari no és en absolut important. No es necessari decorar-la amb colors ni imatges, únicament ha de mostrar la informació suficient per poder utilitzar el sistema.

En la fase d'anàlisi es va confeccionar un prototip dels subsistemes, l'objectiu del qual era il·lustrar com s'organitzarien i també per testejar que es podien executar tots els casos d'ús. Això no és una qüestió trivial perquè la forma d'interactuar amb una aplicació basada en interfície rica, com s'ha fet fins ara, és molt diferent a com l'usuari es mou per un sistema basat en navegador web.

A continuació es mostren algunes captures del prototip. Com es pot comprovar, el disseny de les pàgines estàtiques del prototip és molt semblant a les pàgines dinàmiques generades en el producte final.

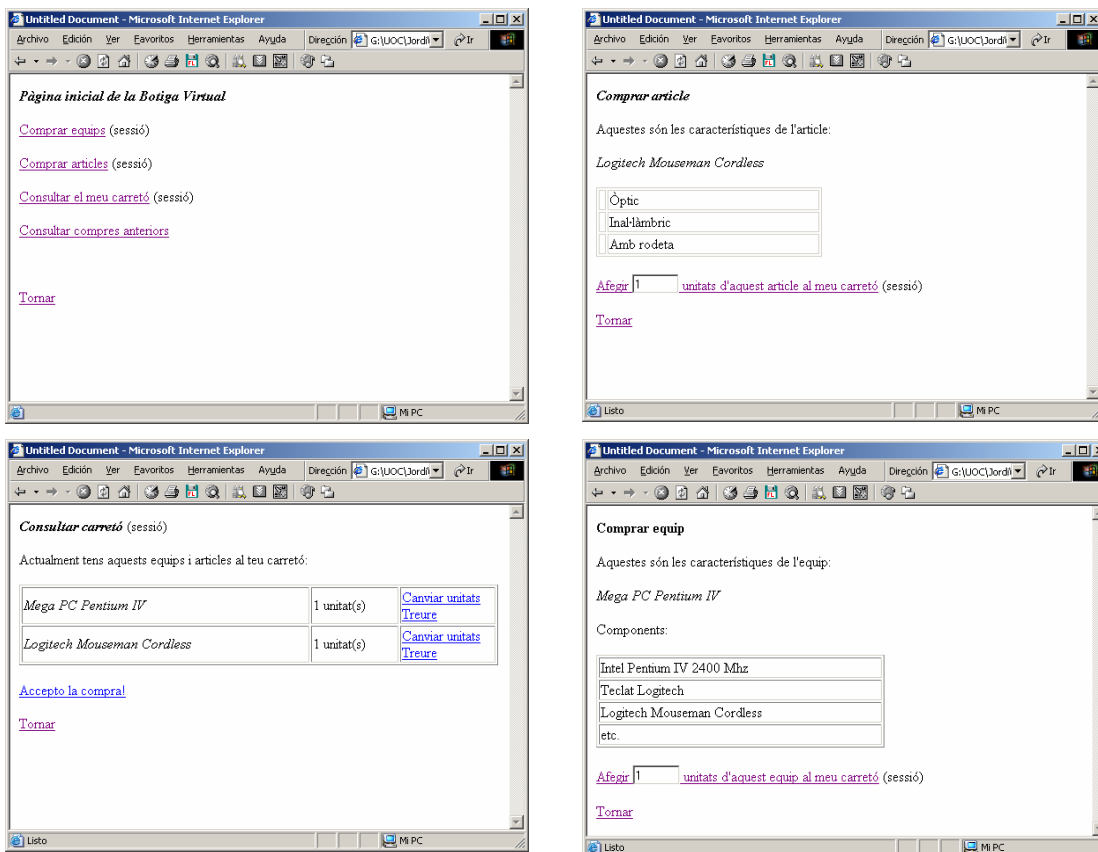


Figura 15 Captura de pàgines de l'aparador

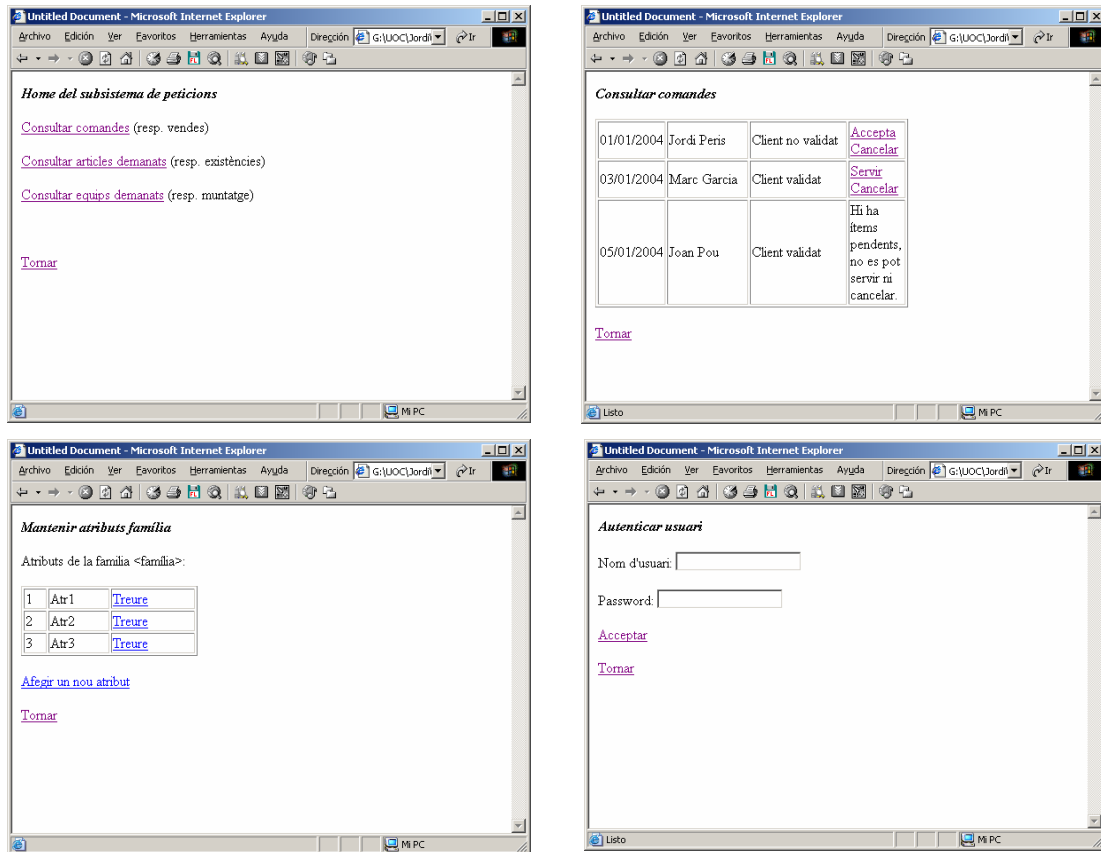


Figura 16 Captura de pàgines de peticions, manteniments i seguretat

Amb l'objectiu de demostrar com podem modificar la interfície d'usuari de forma dinàmica amb els filtres de J2EE, s'ha creat `FiltreUI` del paquet `systema`. Es tracta d'un filtre de post-processament que proporciona com a resposta a la cadena d'execució un `CharResponseWrapper`, que ell mateix manega, per tal de poder interceptar la resposta que es generi en cada cadena i, abans de transportar-la al costat client, modificar-la segons convingui.

En aquest filtre tansols s'afegeix el text `<br><p>TFC 2004 - Jordi Forn i Vilarrasa</p>` al final del body de la resposta, però serveix per il·lustrar com podríem manipular-la com ens plagui, ja sigui per combinar XML+XSLT, aplicar un CSS personalitzat segons les preferències de l'usuari, inserir informació de copyright o de contacte, modificar la codificació de caràcters, etc.

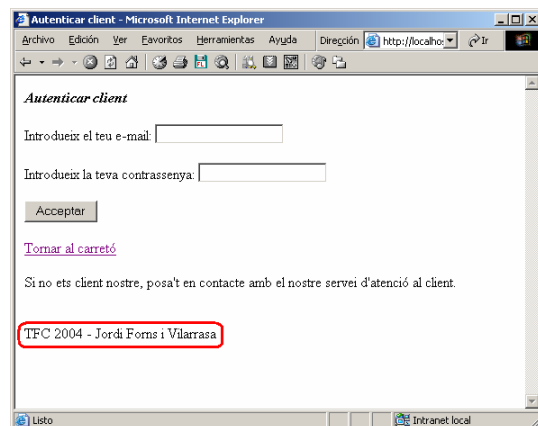


Figura 17 Mostra del resultat de FiltreUI



## 8. Disseny arquitectònic

### 8.1. Requeriments del client

El disseny arquitectònic del sistema ens el marca en bona part el client que ens ha demanat l'aplicació: es tracta d'una botiga informàtica de tamany petit/mitjà que desitja posar a disposició dels seus clients un aparador virtual on puguin navegar pels seus productes i realitzar comandes. Posteriorment, l'equip de l'empresa ha de poder satisfer les comandes i fer-ne el seguiment en les seves instal·lacions i, finalment, mantenir informat el client de l'evolució de les seves comandes.

Disposen d'una base de dades Access molt senzilla, però força extensa, que fan servir per mantenir el seu stock i que ens servirà més endavant per incorporar la seva informació en la base de dades del sistema. Encara no han estudiat quin SGBD utilitzaran, sobretot perquè no disposen de temps d'estudiar els SGBD existents en el mercat ni tampoc tenen un gran coneixement de l'oferta ni de com avaluar-la. Això és un motiu suficient per decantar-nos per EJB amb persistència manegada pel contenidor, que ens abstraïu tota la problemàtica de la persistència i ens permetrà canviar el SGBD, tant durant el desenvolupament del sistema, com quan estigui finalitzat.

La idea del client no és fer una aplicació definitiva, sinó crear una infraestructura que els serveixi de base per desenvolupar la seva solució. Actualment no disposen de personal qualificat que pugui fer el manteniment del sistema, per això els interessa explorar les capacitats de J2EE i incorporar aquelles que més els interessin.

Com es pot veure, les característiques del client condicionen força les eleccions que fem durant el projecte, cosa que, d'altra banda, acostuma a ocórrer en els projectes comercials. Per exemple, s'ha triat Java i no qualssevol altra tecnologia perquè el client disposa de tres equips que fan de servidors web en altres projectes que ja estan en marxa. Dos d'ells corren sobre Windows i un sobre Linux. Com que no tenen clar quin SO utilitzaran ni si canviaran de SO quan el sistema estigui finalitzat, desenvolupar-lo amb Java és la millor alternativa.

La situació del client ha condicionat fins i tot la forma d'estructurar l'aplicació i d'escriure el codi: enlloc d'escriure codi molt especialitzat en un aspecte, és més adequat estructurar-lo de forma molt genèrica, que sigui especialment fàcil modificar-lo i estendre'l.

## 8.2. Arquitectura

### 8.2.1. *Arquitectura del sistema*

L'enfocament més adequat per les característiques d'aquest projecte és utilitzar el model MVC (Model-View-Controller). L'objectiu d'aquest model és desacoblar les dades de l'aplicació (*model*) de la seva representació (*view*) per tal de produir un sistema altament modular. El controlador (*controller*) és l'encarregat de relacionar la vista amb el model. Ens interessa especialment aquest enfocament perquè ens ha de resultar molt fàcil reemplaçar la vista que tenim actualment per una altra de més elaborada i atractiva, cosa que és vital en l'aparador virtual.

Implementat correctament, MVC permet que els diferents components de les tres capes siguin independents els uns dels altres. Per tal d'aconseguir aquest objectiu cal separar marcadament el model de la vista del controlador. En el nostre cas utilitzarem el següent enfocament:

- La vista s'implementarà mitjançant JSPs (Java Server Pages) i, en alguns casos amb HTML.
- El controlador s'implementarà sempre en forma de Servlets o Filters.
- El model s'implementarà amb EJBs amb persistència manglejada pel contenidor (CMP).

L'arquitectura de l'aplicació es pot representar de la següent forma:

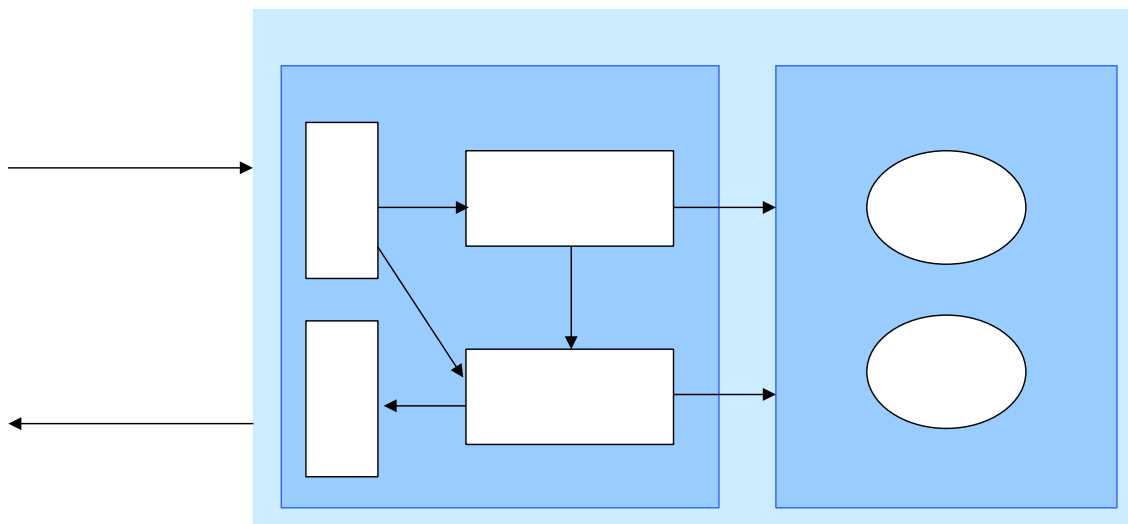


Figura 18 Representació de l'arquitectura del sistema

### **8.2.2.      *Arquitectura de la vista***

La vista s'implementarà mitjançant JSPs (Java Server Pages) i, en alguns casos amb HTML.

Per la aplicació necessitem una forma de generar contingut web de forma dinàmica, ja que ha de representar els continguts del model, que són canviants. La tecnologia JSP (Java Server Pages) ens permet d'aconseguir-ho fàcilment, alhora que facilita que separem la presentació de la lògica del sistema.

Tot i això, és fàcil caure en l'error d'incorporar codi de control o de lògica de negoci en pàgines JSP, cosa que trencaria amb els principis de MVC. Per això, és necessari imposar certes regles en el disseny de les nostres JSP:

- En cap cas la vista modificarà cap component del model, sinó que sempre li haurà de demanar al controlador adequat que ho faci, proporcionant-li la informació necessària per cada petició.

Per exemple: tenim una JSP destinada a modificar les dades d'un article. Quan l'usuari accepta la informació que ha introduït, la JSP no ha de realitzar la modificació pròpiament dita, sinó que ha de sol·licitar els serveis d'un controlador que ho sàpiga fer, facilitant-li les dades que ha introduït el client i tot allò que sigui necessari per fer la modificació.

- Es permet que les JSP realitzin tasques de consulta senzilles, per exemple, per omplir una taula amb els diversos articles del sistema. Això és així perquè no té massa sentit crear un Servlet per a que fagi la recerca i ens retorni un conjunt d'EJBs o de Data Objects, ja que és una operació molt senzilla.

### **8.2.3.      *Arquitectura del controlador***

El controlador s'implementarà sempre en forma de Servlets o Filters.

Els Servlets rebran les peticions dels components de la vista, realitzaran algun processament lleuger, interactuaran amb el model i retornaran una vista nova corresponent a la petició que li hagi arribat.

És molt important que el processament que fan els filtres no tingui res a veure amb la lògica de negoci, ja que d'això se n'ha d'encarregar exclusivament els components del model.

La tasca dels Filters és realitzar tasques de control genèriques o que afecten a qualssevol petició. Pel primer cas tenim el filtre de post-processament FiltreUI, la finalitat del qual és

únicament demostrar com amb un Filter podem modificar la vista generada per un Servlet. Pel segon cas tenim el filtre de pre-processament FiltreSeguretat, que s'encarrega de validar que l'usuari de l'aplicació té els drets suficients per a que es pugui resoldre la seva petició i per a controlar l'autenticació d'usuaris.

#### **8.2.4. *Arquitectura del model***

El model s'implementarà amb EJBs amb persistència manglejada pel contenidor (CMP).

S'ha escollit CMP enlloc de BMP per un requeriment del client, que és que la implementació del sistema ha de ser independent del SGBD utilitzat, ja que encara no s'ha decidit quin s'utilitzarà. A més, és possible que canviem de SGBD diverses vegades en diversos moments (desenvolupament o explotació) i per diversos motius (cost inviable d'un SGBD, canvi de sistema operatiu, proves de rendiment, etc.), cosa que fa que escollim d'abstraure'ns de la base de dades tant com sigui possible.

La missió d'EJB és oferir els objectes de negoci de l'aplicació i implementar la lògica de negoci del sistema de forma que siguin reutilitzables. Aquesta lògica en la seva totalitat s'implementa en el model, cosa que és absolutament imprescindible per a satisfer els objectius de MVC: qualsevol component del sistema ha de poder-se intercanviar per un de nou sense que això afecti a la resta del sistema. Per tant, si tenim tota la lògica encapsulada en components del model, fàcilment podrem substituir un component controlador o una vista, per un de nou que satisfagi els requeriments de l'aplicació i sense que el comportament general es vegi alterat.

Alguns components del model (EJBs) són composicions d'altres components, com és el cas d'un equip i els seus components. La classe contenidora s'ha de veure efectivament com un tot, és a dir, qualssevol operació que es vulgui realitzar sobre un objecte contingut ha de passar inevitablement pel seu contenidor. Per això, les classes contingudes no han de ser accessibles de forma remota, ja que no hi podem interactuar directament. Aquestes classes només tindran una interfície local, que serà la que utilitzin les classes contenidores per accedir-hi a través de les peticions que ofereixen a l'exterior.

Un altre motiu per què això és necessari és que aquestes operacions poden modificar l'estat de la classe contenidora. Per exemple, quan afegim un ítem a una comanda aquesta ha d'actualitzar el seu import total sumant-li l'import d'aquell ítem. Oferint un mètode en la comanda per afegir un ítem aconseguim fàcilment controlar el seu estat.

## 9. Conclusions

Durant la realització d'aquest Treball de Fi de Carrera, s'ha aconseguit realitzar totes les fases del cicle de vida del programari, excepte la de proves. Tot i que la implementació s'ha hagut de retallar per limitació de temps, els timings previstos s'han complert sense grans enderraments ni imprevistos.

Gràcies a la claredat dels requeriments i a l'elevat nivell de detall que es va assolir en la fase de disseny, s'ha aconseguit realitzar un sistema que permet al destinatari del projecte el seu anàlisi en profunditat, així com modificar allò que cregui convenient, gràcies a la cohesió i mínim acoblament entre les classes que el componen.

En quant als productes utilitzats durant la realització del projecte (totes gratuïtes), cal destacar l'espectacular qualitat de l'IDE Eclipse, a part, òbviament de J2EE. MySQL, JBoss i Ant han complert les expectatives que se n'espera sobradament.

En quant a les tecnologies de J2EE que s'han utilitzat en el projecte (JSP, Servlets i EJB) i comparant-les amb el seu competidor més directe, Microsoft .Net, crec que JSP+Servlets+Filtres superen de molt les capacitats que ofereix ASP.Net, mentre que no hi ha cap equivalent a EJB, la potència del qual m'ha sorprès extraordinàriament; tot i que en un primer moment treballar amb tantes classes i interfícies es fa ferragós, quan hom implementa la lògica de negoci amb la facilitat que ofereix EJB, es veu altament recompensat.

Com a desenvolupador, no obstant, vull esmentar un punt en el qual Microsoft supera a Sun de molt i, sobretot, amb l'aparició de Visual Studio .Net. Es tracta, és clar, de la potència de l'IDE de Microsoft i el fet de disposar d'un sol proveïdor, contra la marabunta de proveïdors d'eines Java. Tot i la poca complexitat d'aquest projecte, s'han fet servir cinc eines (Java, Eclipse, JBoss, Ant, MySql) de cinc proveïdors diferents, cosa que no gens atractiva a l'hora d'afrontar un projecte comercial. Si haguéssim adoptat l'aproximació de Microsoft, podríem haver-ho fet tot amb els seus productes; els equivalents als enumerats anteriorment serien .Net, Visual Studio.Net, IIS, Visual Studio.Net, SQL Server.

No es que sigui un defensor de Microsoft però, tot i que està clar que tota la varietat que existeix en el món de Java ens permet escollir entre moltes opcions, això també existeix en el món Microsoft, amb la diferència que podem optar per fer servir els productes d'un sol proveïdor.

## 10. Glossari

**Aparador virtual:** es refereix al subsistema destinat a que els clients consultin els productes de la botiga, facin les seves compres, etc. Es coneix també com la part pública (accessible publicament) del sistema, mentre que la resta es coneix com la part privada perquè és accessible únicament des de la intranet de l'organització.

**Article:** qualsevol objecte que la botiga posi a la venda i que no sigui un equip complet. Els articles s'agrupen per la família a la qual pertanyen per tal de manejar-los i localitzar-los fàcilment. En un article es configuren també els atributs disponibles en la seva família.

**Atribut:** característica que té o pot tenir un article pertanyent a una família. Els atributs d'una família són totes aquelles característiques que pot tenir un article que hi pertany; els atributs d'un article són les característiques que té un article d'una família d'entre tots els que existeixen en aquella família.

**Botiga virtual:** es refereix al sistema complet, tant als subsistemes públics com els privats.

**Carretó:** es tracta d'una comanda no definitiva. S'utilitza en l'aparador virtual per a que els clients puguin simular una comanda. Quan el client decideix portar endavant la compra, passa tots els ítems del carretó a una nova comanda real, quedant el carretó buit.

**Client:** tota persona registrada i autoritzada en la botiga per a comprar a través de l'aparador virtual.

**Comanda:** conjunt d'equips i articles que adquireix un client en un moment determinat.

**Component:** s'anomena component a qualsevol article quan forma part d'un equip.

**Equip:** es tracta d'una configuració d'un equip informàtic complet, generalment format per un ordinador, un monitor, teclat, ratolí, etc.; tots ells en seran components.

**Família:** abstracció utilitzada per agrupar articles d'un mateix tipus (ratolins, teclats...).

**Ítem d'una comanda:** es refereix indistintament a un equip o un article que pertany a una comanda.

**Petició:** cacaescuna de les tasques del conjunt necessari per servir una comanda d'un client. Cada ítem d'una comanda representa una petició per a la botiga.

**Usuari:** com a usuari del sistema, es refereix a qualsevol persona autoritzada a utilitzar els subsistemes interns (seguretat, manteniment, peticions). Generalment es tracta de treballadors de la botiga; se'ls poden assignar diferents nivells d'accés en funció del rol que ocupin en l'organització.


## 11. Bibliografia

### 11.1. Java i J2EE

F. Javier Moldes *Java 2*. Anaya, 2000.


Ed Roman, Scott Ambler, Tyler Jewell *Mastering Enterprise JavaBeans Second Edition*. Wiley Computer Publishing.

Jack Shirazi *Java Performance Tuning*. O'Reilly.

 Javier García de Jalón, José Ignacio Rodríguez, Aitor Imaz *Aprenda Servlets de Java*. Universidad de Navarra, 1999.

 DeveloperWorks *Introduction to JSP Technology*. [ibm.com/developerWorks](http://ibm.com/developerWorks).

 DeveloperWorks *Building Java HTTP Servlets*. [ibm.com/developerWorks](http://ibm.com/developerWorks).

 Sun Microsystems, Inc.

[\*Sending Email From a Servlet\*](#)

[\*Improving Designs with the MVC Design Pattern\*](#)

[\*Using Enterprise Beans with JSP Pages\*](#)

[\*EJB QL Select Methods\*](#)

[\*Advanced EJB QL\*](#)

[\*Finder Methods and EJB-QL\*](#)

[\*Filtering Responses\*](#)

[\*Filtering Servlet Requests\*](#)

[\*The Essentials of Filters\*](#)

[\*Choosing Whether To Use Enterprise Beans\*](#)

### 11.2. JBoss

Andreas Schaefer *JBoss 3.0 Quick Start Guide*. JBoss Group, LLC.

## **12. Annexos**

Presentació del projecte en format Power Point; fitxer TFC\_JForns\_Presentació.ppt.