

Aplicació web per a una botiga de mobles

Estudiant

Joan Puig Guitart

Director

Antoni Oller Arcas

Primavera 2004

Memòria del Treball Final de Carrera

Eng. Tèc. en Informàtica de Gestió

Universitat Oberta de Catalunya

0. Índex

0. Índex.....	3
1. Introducció	5
1.1. Motivació	5
1.2. Estructura.....	5
1.2.1. Estructura de la memòria	6
2. Pla de treball.....	7
2.1. Introducció.....	7
2.2. Organització de l'equip.....	7
2.3. Recursos de programari.....	7
2.4. Descomposició del projecte en tasques	8
2.5. Calendari del projecte	9
2.6. Temporalització i documentació	9
3. Especificació	11
3.1. Descripció detallada del projecte	12
3.1.1. Catàleg.....	12
3.1.2. Usuari	12
3.1.3. Projecte	13
3.1.4. Gestió Continguts	13
3.1.5. Gestió Usuaris	14
3.2. Casos d'ús	14
3.2.1. Catàleg.....	14
3.2.2. Usuari	15
3.2.3. Projecte	17
3.2.4. Gestió Continguts	18
3.2.5. Gestió Usuaris	19
3.3. Diagrama d'entitats	21
4. Disseny	22
4.1. Diagrames.....	22
4.1.1. Diagrama de classes	22
4.1.2. Diagrames de casos d'ús.....	23
4.2. Disseny de la persistència.....	30

4.2.1.	Diagrama de la base de dades	31
4.2.2.	Gestors de la persistència	31
4.3.	Altres aspectes.....	32
4.3.1.	Patrons.....	32
4.3.2.	Reutilització	34
4.3.3.	Excepcions	35
4.3.4.	Interfícies gràfiques	35
5.	Implementació.....	37
5.1.1.	El paquet tfc.jar	37
5.1.2.	L'aplicació web.....	39
5.1.3.	Proves	40
6.	Conclusions	41
6.1.	Temporalització	41
6.2.	Abast.....	41
6.3.	En resum	43
7.	Manuais	44
7.1.1.	Manual d'instal·lació.....	44
7.1.2.	Manual d'usuari.....	45
8.	Recursos.....	51
8.1.	Bibliografia.....	51
8.2.	Recursos a Internet	51

1. Introducció

1.1. Motivació

Aquest document és la memòria del Treball Final de Carrera dels estudis d'enginyeria tècnica en informàtica de gestió. I aquest Treball Final de Carrera és un projecte de construcció de programari: una aplicació web.

Paral·lelament als estudis d'informàtica, la majoria d'estudiants portem a terme projectes relacionats amb els estudis: fer una pàgina web per una entitat cultural, fer una base de dades per un venedor conegut, etc. En el meu cas, els projectes que he dut a terme han estat principalment per l'empresa familiar, una botiga de mobles.

Els projectes que he dut a terme per aquesta empresa han estat la construcció del programari de gestió, i la pàgina web.

En el moment de fer el treball de fi de carrera, va semblar una bona opció aprofitar el coneixement que ja tenia del sector per a aquest treball, i el resultat del treball per a l'empresa familiar. Així doncs, el projecte escollit va ser la construcció d'una aplicació web per a una botiga de mobles, que ampliés a bastament l'anterior.

Un altre motiu per dur a terme aquest projecte ha estat per la relació amb Internet, amb la web. És un projecte diferent en la forma dels que es duen a terme durant la carrera, tot i que no en el fons: el procés de construcció de programari ha estat el clàssic cicle de vida en cascada, utilitzant la metodologia d'orientació a objectes, i aplicant patrons de disseny.

1.2. Estructura

Aquest Treball Final de Carrera consta de diferents parts que, conjuntament, recullen el treball que he dut a terme durant aquest quadrimestre.

- **Memòria.** És aquest mateix document, on es presenta el projecte de construcció de programari i inclou els principals documents elaborats (pla de treball, especificació, disseny, manuals), així com comentaris sobre la implementació, i una presentació del producte.
- **Producte.** És l'aplicació web desenvolupada, de forma empaquetada. Amb l'ajut del manual d'instal·lació es pot posar en funcionament l'aplicació web.
- **Presentació en diapositives.** És un resum del treball dut a terme durant aquest quadrimestre. Es pot considerar com un guió de lectura d'aquesta memòria.

1.2.1. Estructura de la memòria

Tal com el producte final és el resultat d'aquest Treball Final de Carrera, aquesta memòria recull l'esforç dut a terme per arribar a aquest producte final. Per això comença amb el pla de treball que s'ha seguit per dur a terme el projecte de construcció de programari, on s'hi recull l'abast del projecte, i la temporalització prevista per dur-lo a terme.

Un cop establert el pla de treball (o definició del projecte), el projecte segueix les fases habituals del cicle de vida en cascada. Es presenta el document d'anàlisi, el document de disseny, i es comenten els aspectes generals referents a la implementació duta a terme, així com les proves que s'han fet.

Un programari no es redueix a un arxiu executable en un ordinador. El programari també ha d'incloure un manual d'instal·lació i un manual d'usuari. Aquests dos manuals també s'inclouen en aquesta memòria.

2. Pla de treball

2.1. Introducció

L'empresa Mobles Poble Nou SLL (www.mpoblenou.com), dedicada al comerç al detall de mobiliari, es planteja l'ampliació de la funcionalitat de la seva pàgina web, de cara a oferir un servei més proper als seus clients, i obrir les portes a nous clients.

Un dels objectius del projecte és el desenvolupament d'una aplicació web que permeti consultar el **catàleg de productes** que l'empresa per Internet. També ens proposem que el navegant pugui elaborar-se un **pressupost** per agilitzar el procés de compra.

Per complir l'objectiu d'apropar l'empresa als seus clients, ens agradaria oferir-los alguns **serveis en línia**, com ara:

- demanar hora perquè no s'hagin d'esperar en arribar a la botiga.
- fer un seguiment dels diferents projectes que l'empresa els està portant a terme.

Com a últim objectiu del projecte, volem que la **introducció i posada al dia de continguts** de la pàgina web sigui **senzilla**, perquè el personal de l'empresa se'n pugui cuidar. De la mateixa manera, els diferents serveis que s'oferiran han de ser senzills de gestionar.

Tot i que l'aplicació es desenvolupi per a una empresa en concret, volem que sigui fàcilment reutilitzable. Per una altra banda, en el marc del creixement que està experimentant el comerç electrònic, volem que l'aplicació contempli la possibilitat d'augmentar les seves prestacions, mantenint la feina feta.

2.2. Organització de l'equip

L'equip que desenvoluparà aquest projecte està format per en Joan Puig i l'Antoni Oller. La organització de l'equip és molt senzilla i es detalla a continuació.

En Joan, l'estudiant que està cursant l'assignatura *Treball Final de Carrera* es cuidarà de fer la feina planificada, sota la supervisió i l'assessorament de l'Antoni.

2.3. Recursos de programari

Per a la realització del projecte comptem amb diversos recursos que ens ajudaran. Aquests es van detallant a continuació amb la seva funcionalitat.

- El campus de la UOC permetrà la comunicació entre estudiant i consultor.

- La suite informàtica Ms Office permetrà redactar els diferents documents del projecte. En concret, utilitzarem l'aplicació Ms Project per a la planificació i seguiment del projecte.
- L'entorn integrat de desenvolupament Netbeans permetrà dur a terme les fases d'implementació i proves de l'aplicació web.

2.4. Descomposició del projecte en tasques

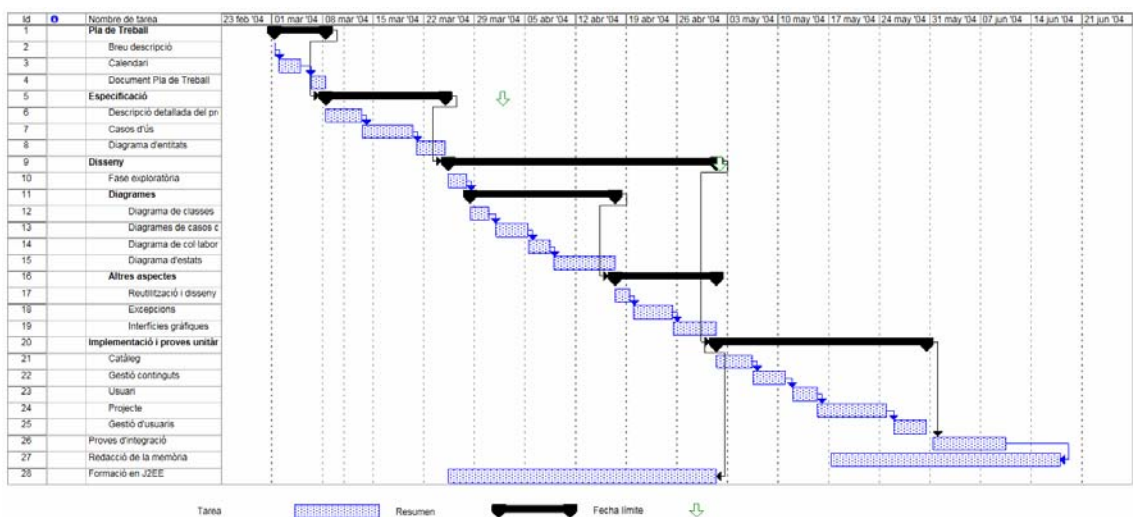
Per a establir aquesta descomposició del projecte en tasques, considerem les habituals en tot projecte de desenvolupament de programari, afegint-hi la redacció de la memòria (que contindrà també la documentació del programari) i la formació en J2EE, per aprendre a utilitzar les eines que s'utilitzaran abans de necessitar-les.

Vet aquí doncs, la descomposició del projecte en tasques:

- Pla de Treball
 - Breu descripció
 - Calendari
 - Document Pla de Treball
- Especificació
 - Descripció detallada del projecte
 - Casos d'ús
 - Diagrama d'entitats
- Disseny
 - Fase exploratòria
 - Diagrames
 - Diagrama de classes
 - Diagrames de casos d'ús
 - Diagrama de col·laboració
 - Diagrama d'estats
 - Altres aspectes
 - Reutilització i disseny
 - Excepcions
 - Interfícies gràfiques
- Implementació i proves unitàries
 - Catàleg
 - Gestió de continguts
 - Usuari
 - Projecte
 - Gestió d'usuaris
- Proves d'integració
- Redacció de la memòria
- Formació en J2EE

2.5. Calendari del projecte

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Pla de Treball	7 días	lun 01/03/04	lun 08/03/04	
2	Breu descripció	1 día	lun 01/03/04	lun 01/03/04	
3	Calendari	3 días	lun 01/03/04	jue 04/03/04	2
4	Document Pla de Treball	3 días	sáb 06/03/04	lun 08/03/04	3
5	Especificació	15 días	lun 08/03/04	mié 24/03/04	1
6	Descripció detallada del pr	4 días	lun 08/03/04	sáb 13/03/04	
7	Casos d'ús	7 días	sáb 13/03/04	sáb 20/03/04	6
8	Diagrama d'entitats	4 días	sáb 20/03/04	mié 24/03/04	7
9	Disseny	30 días	jue 25/03/04	sáb 01/05/04	5
10	Fase exploratòria	3 días	jue 25/03/04	sáb 27/03/04	
11	Diagrames	14 días	dom 28/03/04	sáb 17/04/04	10
12	Diagrama de classes	3 días	dom 28/03/04	mar 30/03/04	
13	Diagrames de casos c	5 días	mié 31/03/04	lun 05/04/04	12
14	Diagrama de col·labor	3 días	lun 05/04/04	jue 08/04/04	13
15	Diagrama d'estats	3 días	jue 08/04/04	sáb 17/04/04	14
16	Altres aspectes	13 días	sáb 17/04/04	sáb 01/05/04	11
17	Reutilització i disseny	3 días	sáb 17/04/04	lun 19/04/04	
18	Excepcions	5 días	lun 19/04/04	dom 25/04/04	17
19	Interfícies gràfiques	5 días	dom 25/04/04	sáb 01/05/04	18
20	Implementació i proves unitàr	28 días	sáb 01/05/04	dom 30/05/04	9
21	Catàleg	5 días	sáb 01/05/04	jue 06/05/04	
22	Gestió continguts	5 días	jue 06/05/04	lun 10/05/04	21
23	Usuari	3 días	mar 11/05/04	sáb 15/05/04	22
24	Projecte	10 días	sáb 15/05/04	lun 24/05/04	23
25	Gestió d'usuaris	5 días	mar 25/05/04	dom 30/05/04	24
26	Proves d'integració	10 días	lun 31/05/04	jue 10/06/04	20
27	Redacció de la memòria	30 días	lun 17/05/04	jue 17/06/04	26FF+7 días
28	Formació en J2EE	30 días	jue 25/03/04	sáb 01/05/04	20CF



2.6. Temporalització i documentació

Els lliuraments parcials que establim per al seguiment del projecte coincideixen amb la finalització de les fases d'especificació i disseny, en què caldrà lliurar els documents obtinguts com a culminació de la fase.

El lliurament final del projecte queda establert pels dies 14 al 17 de juny. El dia 14 de juny hi haurà el lliurament d'una versió *entregable*, deixant un cert marge per a últims retocs.

Lliurament	Data
Document d'anàlisi	1 d'abril
Document de disseny	1 de maig
Lliurament final	14 al 17 de juny

El lliurament final del projecte constarà del producte de programari desenvolupat, i de la memòria del projecte.

La memòria del projecte inclourà, com a mínim, tota la informació del procés de desenvolupament (especificació, disseny, manual del programador) i la informació que necessitarà el client o usuari final del producte (manual de l'administrador, manual d'usuari, ...).

Abans de tots els lliuraments, l'estudiant farà arribar esbossos al consultor per elaborar cada document amb els seus comentaris i suggeriments.

3. Especificació

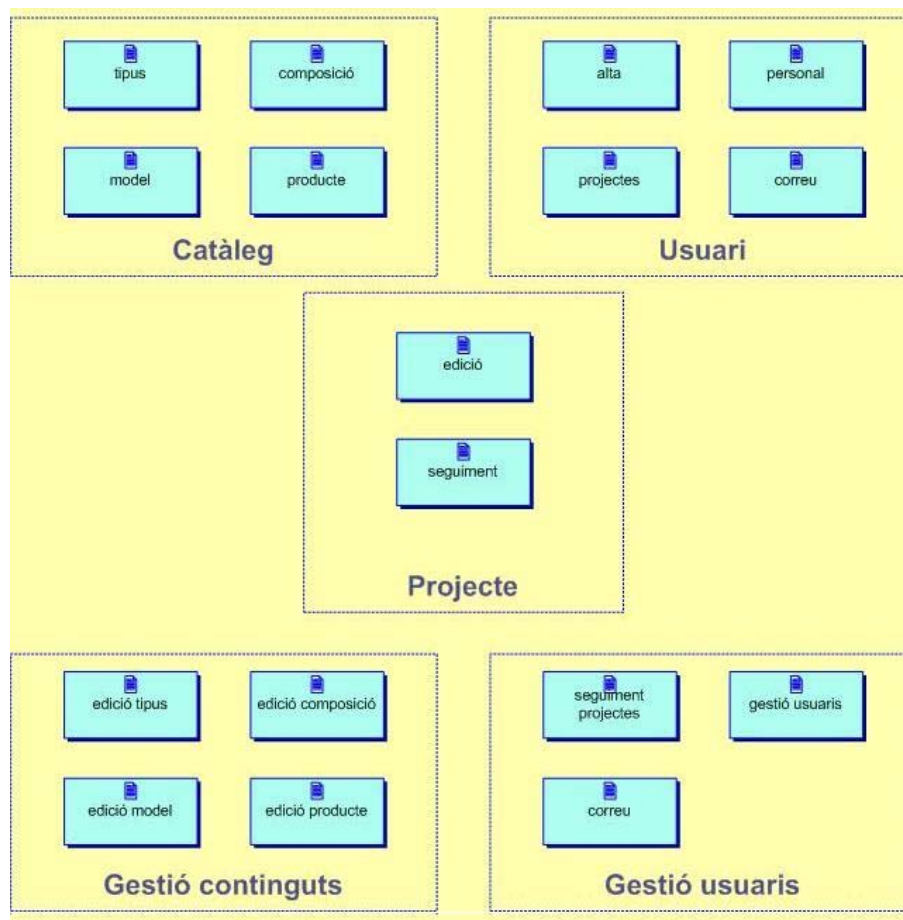
En aquest projecte ens plantegem desenvolupar una aplicació web per a una botiga dedicada al comerç al detall de mobles. En aquest document ens fem ressò de la funcionalitat que ha d'oferir aquesta aplicació, tot especificant-ne els requeriments.

Un dels objectius del projecte és que l'aplicació web permeti consultar el **catàleg de productes** de l'empresa per Internet. També ens proposem que el navegant pugui elaborar un **pressupost** per agilitzar el procés de compra.

Per una altra banda, també es pretén apropar els serveis de l'empresa als seus clients, i per això plantegem oferir-los **serveis en línia**, com ara demanar hora perquè no s'hagin d'esperar en arribar a la botiga, o fer un seguiment dels diferents projectes que l'empresa els està portant a terme. Es tracta d'una funcionalitat B2C (Business to Consumer: Empresa a Client).

Com a últim objectiu del projecte, volem que la **introducció i posada al dia de continguts** de la pàgina web sigui **senzilla**, perquè el personal de l'empresa se'n pugui cuidar. De la mateixa manera, els diferents serveis que s'oferiran han de ser senzills de gestionar.

A continuació es presenta un primer esquema de l'aplicació web que estem descrivint.



Tot i que l'aplicació es desenvolupi per a una empresa en concret, volem que sigui fàcilment reutilitzable: que pugui servir per a altres comerços propers o afins. Per una altra banda, en el marc del creixement que està experimentant el comerç electrònic, volem que l'aplicació contempli la possibilitat d'augmentar les seves prestacions, mantenint la feina feta.

Finalment, afegim com a requisit la possibilitat que l'aplicació importi o exporti dades en formats estàndard (xml, per exemple), per poder vincular de manera simple el catàleg online amb el programari de gestió emprat per l'empresa.

3.1. Descripció detallada del projecte

Per a descriure detalladament l'aplicació, especificarem per separat cada subsistema o mòdul que ha d'incloure.

3.1.1. Catàleg

L'aplicació ha de contenir un catàleg de productes, i ha de permetre

- Consultar un producte, i personalitzar-lo. Com a exemple, s'ha de poder consultar informació sobre una tauleta de nit, i veure les diferents possibilitats de personalització (número de calaixos, tipus de fusta i color, model de tiradors, ...)
- Crear composicions de productes d'un mateix tipus i model. Per tipus entenem el destí del producte: mobiliari per a menjador, per a dormitori, ... Per model, entenem una línia de productes d'un proveïdor.
- Editar les composicions creades anteriorment.
- Elaborar pressupostos per a cada composició.
- Considerar la possibilitat futura d'introduir una imatge del producte (o una descripció tridimensional).

3.1.2. Usuari

Per a oferir serveis personalitzats als clients, el subsistema d'usuaris ha de permetre

- Donar d'alta nous usuaris, automàticament.
- Editar les dades personals de cada usuari (nom, clau d'accés, compte de correu, ...)
- Gestionar els projectes de l'usuari (guardar composicions, editar projectes, ...) perquè aquest hi pugui treballar en diferents sessions d'ús de l'aplicació
- Oferir un correu intern per a la comunicació amb l'empresa. Així es podrà fer un seguiment independent del correu electrònic, tot i que també seria interessant poder sincronitzar aquest servei amb una

adreça de correu electrònic (per rebre els missatges via web, i a la bústia personal).

- Reservar hora de visita. Per evitar que els usuaris hagin d'esperar-se en venir a la botiga, volem oferir-los la possibilitat de concretar una hora de visita, en què seran rebuts al moment per un venedor, a la botiga (física) de l'empresa.
- Possibilitat de login al sistema, que doni accés a tots els apartats descrits anteriorment. També es podria oferir la possibilitat de login automàtic a l'usuari que accedeix a l'aplicació des d'un ordinador privat (mitjançant l'ús de galetes).

3.1.3. Projecte

Els usuaris donats d'alta de l'aplicació poden guardar les composicions elaborades, amb els seus pressupostos respectius, com a projectes, per a poder recuperar-los posteriorment.

A més, l'aplicació ha de permetre fer un seguiment del projecte, si aquest es duu a terme. Això inclou poder veure imatges del disseny acordat (plànol bidimensional, perspectives, ...), i obtenir informació sobre l'estat del projecte: aprovat, si les comandes respectives han estat demanades, si s'han rebut, si ja es poden anar a muntar, etc.

També és interessant poder compartir altres documents amb l'usuari, com la factura en format pdf, informació extra sobre altres serveis (per exemple, publicitat sobre matalassos per clients que compren un dormitori de matrimoni), etc.

Més endavant, aquest mòdul també podria arribar a incloure un terminal punt de venda (TPV), per a poder vendre els productes al client en línia. En aquests moments ens sembla poc apropiat degut a les característiques del negoci (basat en projectes a mida pel client).

3.1.4. Gestió Continguts

Perquè l'usuari pugui consultar el catàleg, aquest s'ha d'haver posat en línia prèviament. Aquest subsistema es cuida de la gestió dels continguts, és a dir, de l'entrada d'informació i edició del catàleg.

Per poder configurar el catàleg apropiadament, cal que ens permeti:

- Donar d'alta, editar o esborrar els tipus de mobiliari. Per tipus de mobiliari entenem l'ús o espai on es farà servir. Així, podem distingir entre rebedors, menjadors, dormitoris de matrimoni, moble juvenil, estudi o oficina, ...
- Donar d'alta, editar o esborrar els proveïdors de mobiliari i els models que proporcionen. Cada proveïdor ens ofereix diferents models de mobiliari, és a dir, grups de productes que segueixen una mateixa línia.

Ens cal tenir els productes ben classificats per proveïdors i models, perquè productes de diferents models i/o proveïdors possiblement no es

poden combinar en un mateix projecte (perquè no lliguen del tot: el color cirerer varia de proveïdor en proveïdor, i pot canviar segons el model).

- Un cop tenim la classificació de productes, hem de poder donar d'alta, editar o esborrar els productes. La seva descripció inclourà les característiques que els defineixen i les diferents variacions possibles d'un mateix producte (nombre de calaixos, tipus de tirador, ...)
- Ara que ja comptem amb productes, volem començar a agrupar-los en composicions (per exemple, una taula i quatre cadires, o un escriptori, una llibreria, i una cadira d'oficina), perquè l'usuari pugui escollir-los de cop de cara a crear les seves pròpies composicions.

3.1.5. Gestió Usuaris

Aquest subsistema és per donar suport a les funcionalitats dels subsistemes d'usuari i projecte. Vet aquí les tasques que engloba:

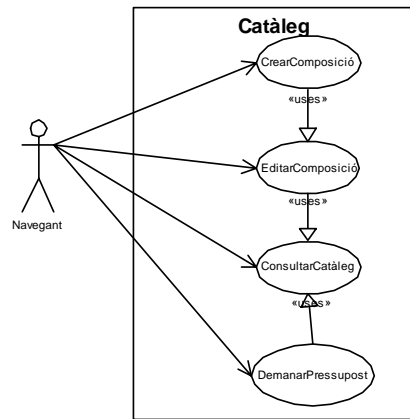
- Donar d'alta, editar o esborrar usuaris, per evitar-los feina a ells mateixos, o per poder donar de baixa els usuaris que entenem que són perjudicials pel sistema (algú que guarda molts projectes, envia molts missatges sense sentit, o fa molt temps que no es connecta, ..., en resum, ocupa els recursos del sistema sense motiu).
- Donar d'alta, editar o esborrar projectes, per evitar la feina als mateixos usuaris, i poder donar de baixa els projectes guardats històrics, o els que no tenen cap interès (buits, un sol producte, ...).
- Gestionar els missatges de correu dels usuaris, podent-los llegir i respondre.
- Afegir comentaris a un projecte perquè l'usuari en conegui l'estat: hem rebut la comanda del seu projecte el 29 de març.
- Incloure documents a un projecte, perquè l'usuari els pugui consultar: imatges, factures, ...

3.2. Casos d'ús

En aquest apartat detallem els casos d'ús de l'aplicació web, organitzats pel subsistema o mòdul del programari al qual pertanyen. En cada cas s'inclouen els diagrames UML necessaris per il·lustrar el que s'està explicant.

3.2.1. Catàleg

Vet aquí les diferents funcionalitats que ha d'oferir el catàleg de l'aplicació web, junt amb la seva descripció detallada en forma de cas d'ús.



En aquest cas, entenem per navegant la persona que utilitza l'aplicació web (navegant per les pàgines web que la conformen) per consultar-ne la seva informació.

- Consulta producte o composició

A partir de la navegació entre tipus o models de productes, el navegant ha de poder escollir una composició o producte per a consultar-la (veure'n una imatge, consultar-ne les característiques, ...)

- Personalitza producte

Un cop escollit un producte, aquest pot oferir opcions de personalització. I el navegant ha de poder-les modificar.

- Crea composició

A partir de diferents productes, o composicions prèvies, el navegant ha de poder crear noves composicions, tot agregant els diferents productes o composicions.

- Edita composició

Les composicions s'han de poder editar: tant les existents, com les creades anteriorment. I editar una composició ha d'incloure: afegir-hi més productes o composicions, eliminar-ne, modificar les opcions de personalització de cadascun dels productes, ...

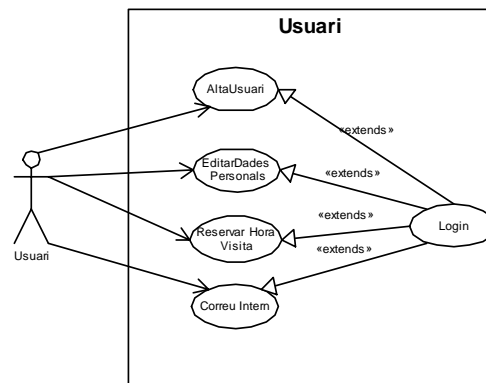
- Elabora pressupost

Un cop l'usuari ha trobat, o creat una composició que li agrada, ha de poder demanar un pressupost detallat per aquella composició.

Per al càlcul del preu s'ha de tenir en compte el preu de cada producte, segons la personalització establerta, i els possibles descomptes per composicions completes.

3.2.2. Usuari

Vet aquí les funcionalitats que ha d'oferir l'aplicació web de cara als diferents usuaris.



En aquest cas, entenem per usuari el navegant que s'ha donat d'alta (registrat) o es vol donar d'alta en el sistema. Un usuari no deixa de ser un navegant però, a l'estar registrat, pot accedir a més serveis.

- Alta / edició

Un usuari ha de poder donar-se d'alta del sistema automàticament, donant les seves dades personals.

Aquestes dades personals li serviran per identificar-se al sistema (nom d'usuari i clau d'accés) i per permetre que l'empresa es posi en contacte amb ell (adreça de correu electrònic).

En tot moment, un usuari ha de poder modificar les seves dades personals. Per això, caldrà que s'hagi identificat correctament al sistema mitjançant el procés de login.

- Gestió projectes

L'interès principal del registre d'usuaris a l'aplicació web és per poder fer un seguiment individual d'un projecte. Així doncs, un usuari ha de poder consultar els diferents projectes amb què està treballant.

Per començar, un usuari pot crear un nou projecte a partir d'una composició. Això li permetrà guardar-la per poder seguir editant-la en les següents visites a l'aplicació web. I també pot guardar els diferents pressupostos elaborats.

Per una altra banda, un projecte també ha de permetre compartir informació entre l'usuari i l'empresa: esbossos del disseny del projecte, altres pressupostos, ofertes de serveis addicionals, ... i l'usuari ha de poder consultar aquesta informació.

Finalment, el seguiment d'un projecte també porta lligada una comunicació, que es pot dur a terme mitjançant el correu intern, que s'explica en un altre cas d'ús.

- Reservar hora de visita

Aquesta és una funcionalitat que creiem que és un valor afegit a l'aplicació web: que els clients puguin reservar hora de visita a l'empresa, de manera que quan ells puguin anar a la botiga de l'empresa, no s'hagin d'esperar perquè els atenguin.

Volem que l'usuari pugui escollir una hora de visita entre les hores d'atenció al públic de la botiga, i que no pugui escollir les hores que algun altre usuari ja ha reservat.

- Correu intern

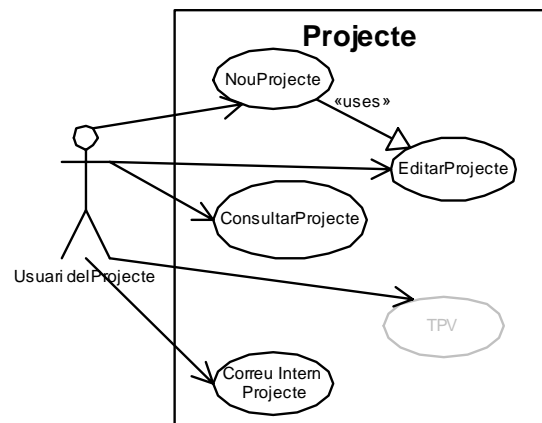
L'usuari ha de poder escriure missatges de correu a l'empresa, i rebre'n la resposta. Tot això s'ha de poder dur a terme en l'entorn web. També seria interessant poder combinar aquest correu amb una adreça de correu electrònic existent (rebre les respostes en una altra bústia de correu).

- Login

Cal que els usuaris s'identifiquin per tenir accés als seus projectes i funcionalitats. Un sistema de clau d'accés (contrasenya) permetrà evitar que algú s'identifiqui com un altre usuari. A més, cal garantir la confidencialitat de les dades i, per això, caldria guardar de manera xifrada les claus d'accés dels usuaris.

3.2.3. Projecte

En aquest apartat es detalla la funcionalitat que ha d'oferir l'aplicació web per als usuaris en relació als seus projectes. Per projecte entenem un servei concret de l'empresa com per exemple moblar un menjador o un dormitori de matrimoni.



- Nou Projecte

Un usuari ha de poder crear un nou projecte, ja sigui per guardar una composició o per crear un canal de comunicació amb l'empresa per tractar un tema en particular (un projecte com ara moblar un estudi).

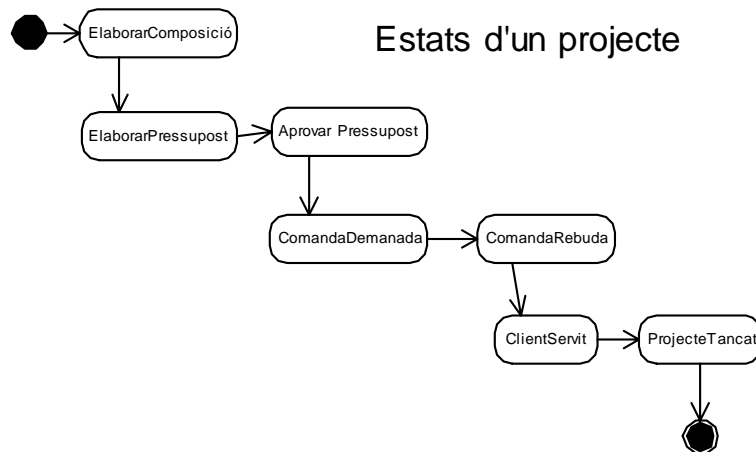
- Editar Projecte

L'usuari ha de poder editar els projectes amb què ja ha treballat, consultant la informació que conté, fent canvis a la composició (conjunt de productes) del projecte, ...

- Consultar Projecte

L'usuari ha de poder consultar tota la informació referent a un projecte: el seu estat actual, els pressupostos que hi estan relacionats, possibles imatges del disseny o plànols en planta, ...

A continuació s'hi poden veure els diferents estats en què es pot trobar un projecte de venda de mobiliari.

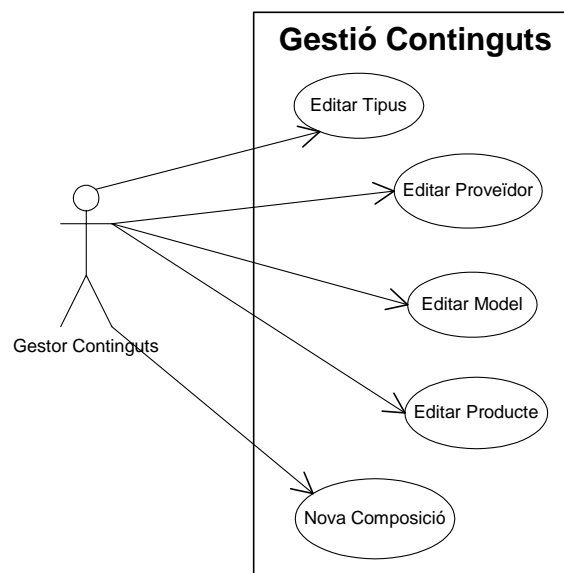


- Correu Intern del Projecte

L'usuari també ha de poder escriure missatges de correu a l'empresa fent referència a un projecte en concret, i rebre'n la resposta. Els missatges i les respostes han de quedar lligats al projecte en qüestió.

3.2.4. Gestió Continguts

En aquest apartat es detalla la funcionalitat que ha d'oferir el mòdul destinat a la gestió de continguts: a introduir els productes al catàleg, a organitzar-los convenientment, a actualitzar-ne les característiques, ...



- Editar Tipus

L'aplicació ha de permetre editar els diferents tipus de mobiliari que ven l'empresa (moble juvenil, menjadors, dormitoris de matrimoni, mobiliari d'oficina, etc.)

- Editar Proveïdor/Model

L'aplicació ha de permetre editar els diferents proveïdors de mobiliari del catàleg, així com els diferents models que ofereix cadascun d'ells. Aquesta classificació ens servirà per poder organitzar adequadament els productes que hi haurà al catàleg.

- Editar Producte

La tasca més important per la posada al dia del catàleg és la introducció de nous productes, i l'actualització dels antics. Per això cal poder editar, afegir o donar de baixa productes.

En la descripció d'un producte, hi considerarem les seves dades bàsiques (nom, referència, proveïdor, ...) i també les opcions de personalització que ofereix.

- Nova Composició

Finalment, ens interessa poder agrupar productes fent composicions, que també s'oferiran conjuntament als navegants que utilitzin l'aplicació web.

3.2.5. Gestió Usuaris

Vet aquí les diferents funcionalitats que ens permet controlar el mòdul de Gestió d'Usuaris. Aquest mòdul està orientat als serveis a l'usuari, i el següent diagrama de casos d'ús resumeix les opcions que ens ofereix:

- Editar Usuaris

Aquest mòdul de l'aplicació ha de permetre l'edició d'usuaris en general: crear-ne de nous, modificar les dades dels existents, donar-ne de baixa si ja no tenen cap relació amb l'empresa (o demanen no tenir-ne), etc.

- Editar Projectes

És des d'aquesta part de l'aplicació que es poden apropar els serveis de l'empresa als clients o usuaris registrats. A part de la funcionalitat essencial de crear nous projectes o donar-ne de baixa, també ha de permetre unes altres operacions.

L'aplicació web ha de permetre afegir informació addicional als projectes, com ara imatges (perspectiva d'un disseny, plànol en planta, foto d'una composició, ...), pressupostos o altres documents d'interès.

També ha de permetre informar de l'estat del projecte, seguint el diagrama d'estats que hem vist anteriorment; i oferir un canal de comunicació entre el client i l'empresa en relació amb el projecte en qüestió. Ho detallem en el següent cas d'ús.

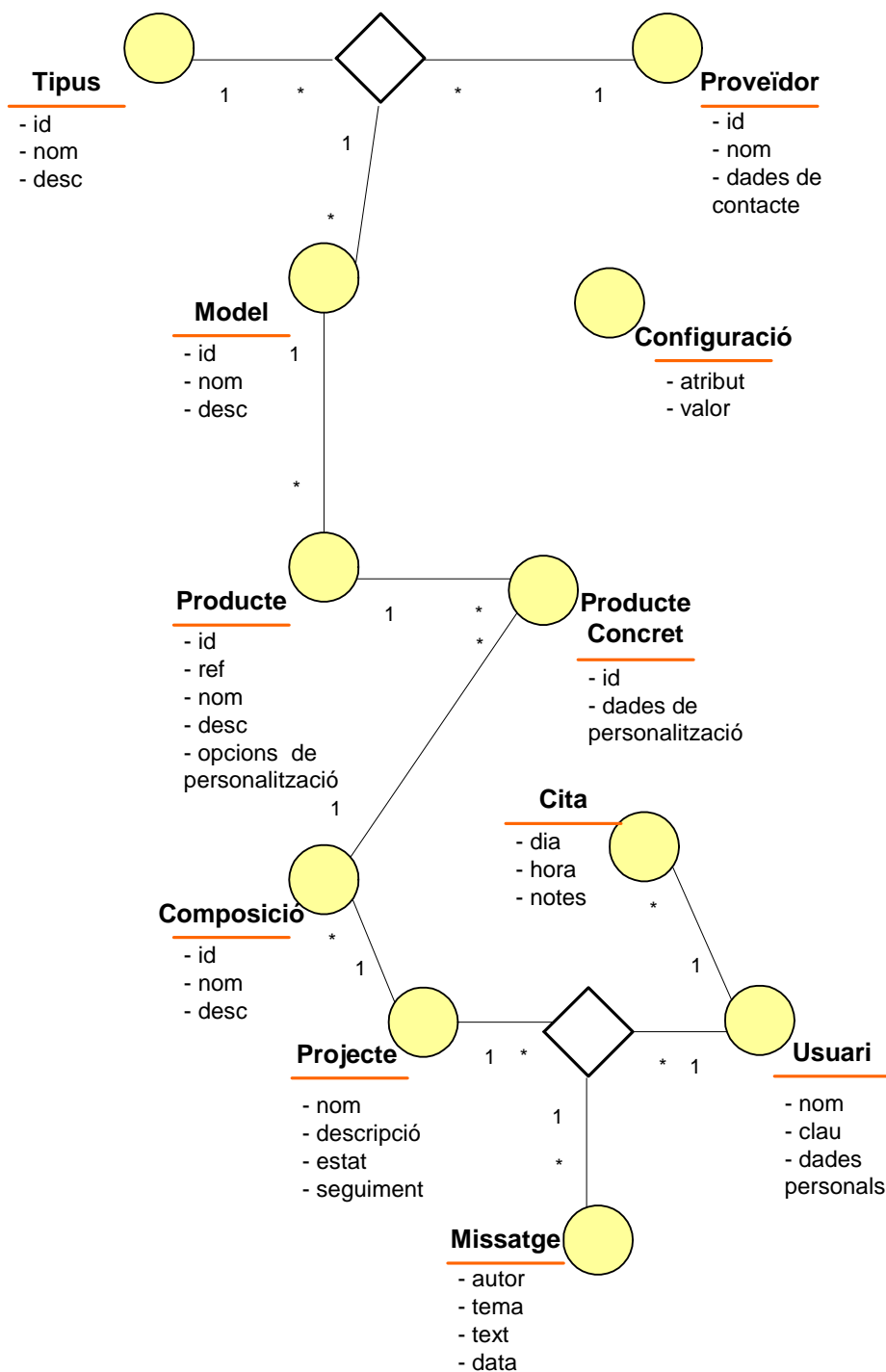
- Gestionar Correu

En la introducció d'aquest document d'especificació, ja es parlava d'establir un canal de comunicació entre els usuaris i l'empresa, en forma de missatges de correu electrònic. I ja hem descrit en els casos d'ús d'Usuari i de Projecte que l'usuari podia utilitzar aquests missatges. En aquest cas, és l'empresa que pot respondre o crear els missatges.

En la gestió dels usuaris, cal que els empleats de l'empresa puguin consultar (rebre) els missatges que han rebut per part dels usuaris, i també han de poder-ne escriure ells, ja sigui per respondre als anteriors, o per iniciar una nova comunicació.

3.3. Diagrama d'entitats

Aquest és el diagrama d'entitats que creiem que reflexa (modela) la realitat



que volem reflexar i amb què treballarà la nostra aplicació web.

Entitats

Tipus, Proveïdor, Model, Producte, Composició, ProducteConcret, Usuari, Projecte, Cita, Missatge, Configuració

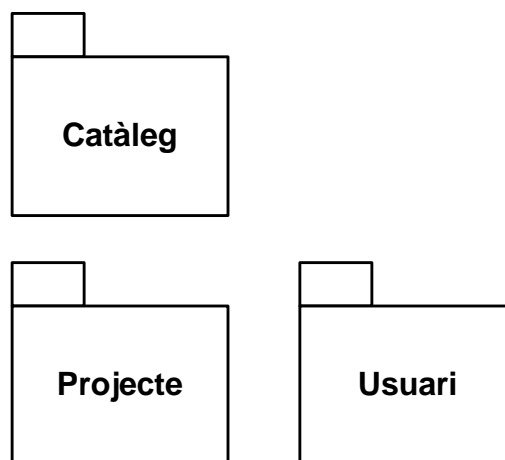
4. Disseny

En aquest document es presenta el disseny per a una aplicació web que respon a l'especificació presentada en un document anterior. Es tracta d'una aplicació web per a una botiga dedicada al comerç al detall de mobles.

En l'especificació de l'aplicació web, hem presentat les funcionalitats que ha de complir dividint-les en cinc mòduls o sistemes: Catàleg, Usuari, Projecte, Gestió Contingut, Gestió Usuari.

De l'àmbit que inclou cada mòdul, és clar que n'hi ha que estan molt lligats: Catàleg i Gestió Contingut, Usuari i Gestió Usuari. Hem preferit aquesta separació en diferents mòduls de cara a l'especificació de funcionalitats perquè, certament, estan clarament diferenciats des del punt de vista de l'actor (un navegant o un usuari per una banda, un empleat de l'empresa per l'altra).

Ara bé, en l'apartat de disseny, potser ja no ens interessa tant el punt de vista de l'actor de cada cas d'ús o funcionalitat. Estem més interessats en la informació (les entitats) amb què es treballa: la lògica del negoci. Des d'aquesta altra òptica, ens interessa més una altra separació en mòduls (que anomenarem paquets, ja que estem fent un disseny orientat a la implementació en tecnologia J2EE). Els paquets amb què treballarem seran, doncs: catàleg, usuari, projecte.

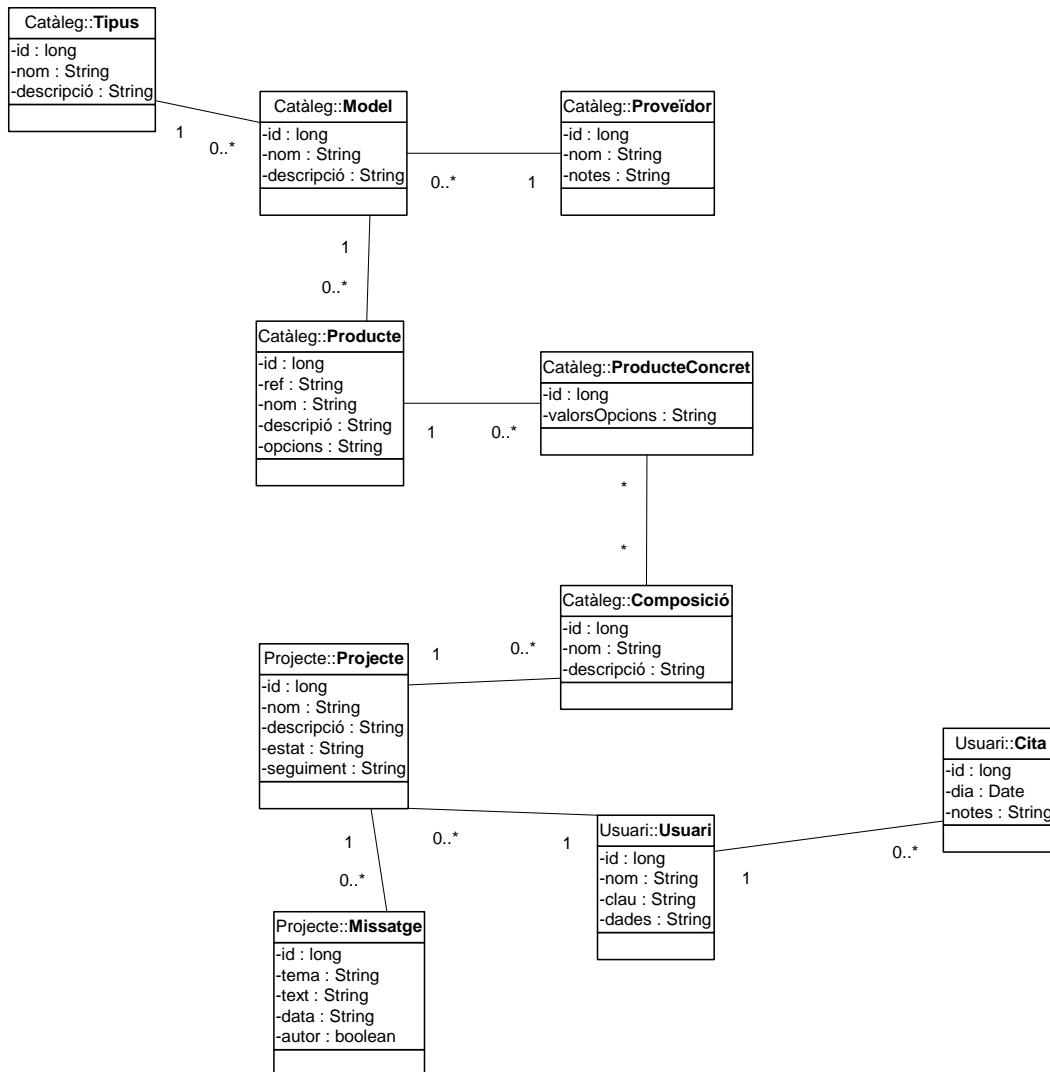


4.1. Diagrames

4.1.1. Diagrama de classes

A continuació hi podem trobar el diagrama de classes que modelitza el sistema d'informació amb què estem treballant.

En aquest cas el que estem representant és el model de dades amb què treballarem, i no totes les classes que haurem d'implementar.



A primer cop d'ull, el diagrama és una simple i directa traducció del diagrama d'entitats que apareixia en el document d'especificació.

La diferència principal que s'ha introduït és la reinterpretació de les relacions ternàries: Projecte - Missatge - Usuari i Tipus - Proveïdor - Model. Entenem que es tracta de relacions ternàries, que es poden representar també com a relacions binàries, cosa que facilitarà la implementació.

4.1.2. Diagrames de casos d'ús

En aquest apartat es descriuen tots els casos d'ús que ha de contemplar l'aplicació, especificant-los d'una manera concisa. La intenció és que aquests casos d'ús passin a ser directament els mètodes d'una classe durant la fase d'implementació.

Presentem els casos d'ús segons la informació amb què treballen, seguint la divisió per paquets esmentada anteriorment (catàleg - usuari - projecte). De tota manera, en cada cas d'ús remarcarem explícitament qui és l'actor que l'utilitza.

- **Catàleg**

Cas d'ús	Consultar Producte
Descripció	El navegant que entra a la nostra pàgina web ha de poder consultar les característiques d'un producte en concret, i consultar-ne les opcions de personalització.
Actor	Navegant
Procés	<ul style="list-style-type: none">• El navegant demana la informació referent a un producte• L'aplicació cerca la informació pertinent mitjançant les classes que implementen la lògica del negoci (que cercaran la informació en una base de dades).• Un cop obtinguda la informació, es crea un objecte de tipus ProducteConcret.• El ProducteConcret es presenta a l'usuari, que hi pot interactuar (per exemple, canviant les opcions de personalització que aquest ofereix).

Cas d'ús	Crear Composició
Descripció	El navegant ha de poder agrupar diversos productes (amb les personalitzacions que els hagi aplicat) tot formant composicions.
Actor	Navegant
Procés	<ul style="list-style-type: none">• El navegant demana crear una composició.• Es crea un objecte Composició• Es presenta l'objecte Composició a l'usuari perquè pugui editar-lo (cas d'ús Editar Composició).

Cas d'ús	Editat Composició
Descripció	El navegant ha de poder treballar amb les composicions que es crea: modificar-ne la descripció, afegir-hi o esborrar-ne productes, editar els productes que conté, ...
Actor	Navegant
Procés	<ul style="list-style-type: none">• El navegant demana editar una composició.• Es recupera l'objecte Composició, i es presenta la informació a l'usuari: el nom, la descripció, i el conjunt de productes de què forma part.• També es permet que l'usuari interactuï amb l'objecte: que hi afegixi productes, que editi els que ja hi ha, que demani un pressupost, ...

Cas d'ús	Elaborar Pressupost
Descripció	Un cop ha elaborat una composició, el navegant o l'usuari poden demanar un pressupost d'aquella composició en particular. Aquest s'elabora automàticament i es presenta al navegant o usuari.
Actor	Navegant o Usuari
Procés	<ul style="list-style-type: none">• Després d'elaborar una composició, es demana un pressupost• Es cerquen els preus de cadascuna de les parts de la composició• Si es tracta d'un usuari registrat, es poden aplicar descomptes particulars, i els empleats podran modificar (<i>arreglar</i>) posteriorment el pressupost• Es presenta la informació obtinguda al navegant

Cas d'ús	Alta Tipus
Descripció	Per poder classificar els productes segons la seva tipologia, ens cal poder definir els possibles tipus amb què tractarem, és a dir: crear-ne de nous, editar-los, ...
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor del catàleg indica que vol donar d'alta un nou tipus.• Es crea un nou tipus amb valors per defecte.• Es mostra a l'usuari la pàgina d'edició del nou tipus que s'ha creat (mitjançant el cas d'ús Edita Tipus).

Cas d'ús	Baixa Tipus
Descripció	Per poder classificar els productes segons la seva tipologia, ens cal poder definir els possibles tipus amb què tractarem, és a dir: crear-ne de nous, editar-los, ...
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol donar de baixa un tipus• Es mostren al gestor les conseqüències de donar de baixa el tipus (número de productes que seran donats de baixa, número de composicions afectades, ...)• Es demana confirmació a l'usuari• En cas de confirmació, es dona de baixa el tipus.

Cas d'ús	Editar Tipus
Descripció	Per poder classificar els productes segons la seva tipologia, ens cal poder definir els possibles tipus amb què tractarem, és a dir: crear-ne de nous, editar-los, ...
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol editar un tipus• Es cerca la informació referent al tipus, que es presenta al gestor en un formulari on pot canviar la informació.• El gestor acaba per acceptar els canvis que ha realitzat• La nova informació es guarda.

Cas d'ús	Alta Proveïdor
Descripció	Per poder classificar els productes segons el seu proveïdor, ens cal poder definir els diferents proveïdors amb què tractarem, és a dir: crear-ne de nous, editar-los, ...
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol crear un nou proveïdor• Es crea un nou proveïdor amb valors per defecte• S'envia l'usuari a la pàgina d'edició del nou proveïdor

Cas d'ús	Baixa Proveïdor
Descripció	Per poder classificar els productes segons el seu proveïdor, ens cal poder definir els diferents proveïdors amb què tractarem, és a dir: crear-ne de nous, editar-los, ...
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol donar de baixa un proveïdor• S'informa al gestor de les conseqüències de donar de baixa el proveïdor (productes i models que s'anul·laran, composicions anul·lades, ...)• Es demana confirmació al gestor i, si procedeix, es dona de baixa el proveïdor

Cas d'ús	Editar Proveïdor
Descripció	Per poder classificar els productes segons el seu proveïdor, ens cal poder definir els diferents proveïdors amb què tractarem, és a dir: crear-ne de nous, editar-los, ...
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol editar un proveïdor• Es cerca la informació pertinent i es mostra al gestor en un formulari on podrà editar-la. Dins de la informació que es pot editar, hi contemplem també els models que ofereix el proveïdor en qüestió.• En rebre l'ordre del gestor, s'actualitza la informació.

Cas d'ús	Alta Producte
Descripció	Per poder omplir el catàleg, s'han de poder donar d'alta els productes.
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol donar d'alta un producte• Es crea un nou producte amb els valors per defecte• Es mostra a l'usuari la pantalla d'edició del nou producte (cas d'ús Edita Producte)

Cas d'ús	Baixa Producte
Descripció	Per poder mantenir el catàleg al dia, cal poder donar de baixa els diferents productes que conté.
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor està editant un producte• El gestor indica que vol donar de baixa el producte• S'informa al gestor de les conseqüències de donar de baixa el producte (composicions que el contenen, ...) i es demana confirmació.• Es dóna de baixa el producte

Cas d'ús	Editar Producte
Descripció	Per poder mantenir el catàleg al dia, cal poder editar els diferents productes que conté: canviar-ne les dades bàsiques, i definir-ne les opcions de personalització que s'oferiran a l'usuari.
Actor	Gestor Catàleg
Procés	<ul style="list-style-type: none">• El gestor indica que vol editar un producte• Es cerca la informació pertinent i es mostra al gestor en un formulari on podrà editar-la. Dins de la informació que es pot editar, hi contemplem també les opcions de personalització pel producte en concret• En rebre l'ordre del gestor, s'actualitza la informació.

- **Usuari**

Cas d'ús	Alta Usuari
Descripció	Per poder treballar amb usuaris, ens cal poder-los donar d'alta. Això és el que ens permet aquest cas d'ús.
Actor	Usuari o Gestor Usuaris
Procés	<ul style="list-style-type: none">• S'indica que es vol crear un usuari• Es crea un nou usuari amb informació per defecte• Es mostra la pantalla d'edició de l'usuari (cas d'ús Editar Usuari)

Cas d'ús	Editat Usuari
Descripció	La informació referent a un usuari s'ha de poder consultar i modificar. Això és el que ens permet aquest cas d'ús.
Actor	Usuari o Gestor usuaris
Procés	<ul style="list-style-type: none">• Un usuari identificat correctament demana editar les seves dades, o un gestor d'usuaris demana editar les dades d'un usuari en particular.• Es mostra la informació pertinent, i es permet la seva modificació.• Quan s'indiqui, s'actualitza la informació amb les noves dades.

Cas d'ús	Baixa Usuari
Descripció	En diverses situacions, pot interessar donar de baixa un usuari (l'usuari vol desvincular-se de l'empresa, l'usuari no utilitza l'aplicació web, ...).
Actor	Usuari o Gestor usuaris
Procés	<ul style="list-style-type: none">• Un actor amb permís per fer-ho (el mateix usuari o un gestor usuaris), sol licita donar de baixa un usuari.• Es mostra la informació pertinent a l'usuari i es demana confirmació.• Si s'escau, es dona de baixa l'usuari.

Cas d'ús	Reservar Hora Visita
Descripció	Un dels serveis que volíem facilitar als clients és la possibilitat de reservar hora de visita per Internet.
Actor	Usuari
Procés	<ul style="list-style-type: none">• L'usuari indica que vol reservar hora de visita• Es mostra l'horari en què pot reservar hora de visita, perquè esculli una hora en concret.• Es reserva l'hora per a l'usuari, tot indicant-li que rebrà confirmació de la reserva (mitjançant el correu intern).

Cas d'ús	Login
Descripció	És un cas d'ús que utilitzen tots els altres: tots els casos d'ús que tenen per actor un usuari i no un navegant, només poden ser duts a terme per usuaris registrats, que s'han d'identificar apropiadament.
Actor	Usuari
Procés	<ul style="list-style-type: none">Es demana el nom d'usuari i la clau d'accésEs cerca l'usuari a la base de dades i s'obté la seva clau d'accés xifradaEs xifra la clau d'accés proporcionada i es compara amb la clau d'accés xifrada.Si coincideixen les claus, l'usuari passa a ser un usuari identificat.

- Projectes**

Cas d'ús	Alta Projecte
Descripció	Per poder treballar amb diferents projectes, cal poder-ne donar d'alta, editar-los, ...
Actor	Usuari o Gestor usuaris
Procés	<ul style="list-style-type: none">Un actor identificat sol licita crear un projecteEs crea un nou projecte, associat a l'usuari que ho ha demanat (o a l'usuari que hagi indicat el gestor d'usuaris) amb valors per defecteEs mostra el nou projecte per editar-lo.

Cas d'ús	Editar Projecte
Descripció	Per poder treballar amb diferents projectes, cal poder-ne donar d'alta, editar-los, ...
Actor	Usuari o Gestor usuaris
Procés	<ul style="list-style-type: none">Un actor identificat sol licita editar un projecteEs cerca la informació pertinent i es mostra a l'actor perquè pugui modificar-la. També es mostren les diferents accions possibles relacionades amb el projecte: escriure o llegir correu, editar composició,...Si l'actor així ho indica, s'actualitza la informació del projecte.

Cas d'ús	Baixa Projecte
Descripció	Per poder treballar amb diferents projectes, cal poder-ne donar d'alta, editar-los, ...
Actor	Usuari o Gestor usuaris
Procés	<ul style="list-style-type: none">• Un actor identificat indica que vol donar de baixa el projecte• Es demana confirmació tot mostrant la informació relacionada amb el projecte• Es dóna de baixa el projecte

Cas d'ús	Enviar Correu
Descripció	Un altre dels serveis que volem oferir: una via de comunicació entre clients i empresa.
Actor	Usuari
Procés	<ul style="list-style-type: none">• L'usuari indica que vol enviar un correu a l'empresa (que pot estar relacionat a un projecte en particular)• Es demana el títol del missatge i el contingut• S'envia el missatge

Cas d'ús	Llegir Correu
Descripció	Junt amb la possibilitat d'enviar missatges de correu, també es poden rebre les respostes.
Actor	Usuari
Procés	<ul style="list-style-type: none">• L'usuari indica que vol llegir el correu• Es mostra una llista dels últims missatges rebuts, indicant el títol i el projecte al que fan referència• L'usuari escull un dels missatges• Es mostra el text complet del missatge en qüestió. A més, s'ofereix a l'usuari l'opció de respondre el missatge

4.2. Disseny de la persistència

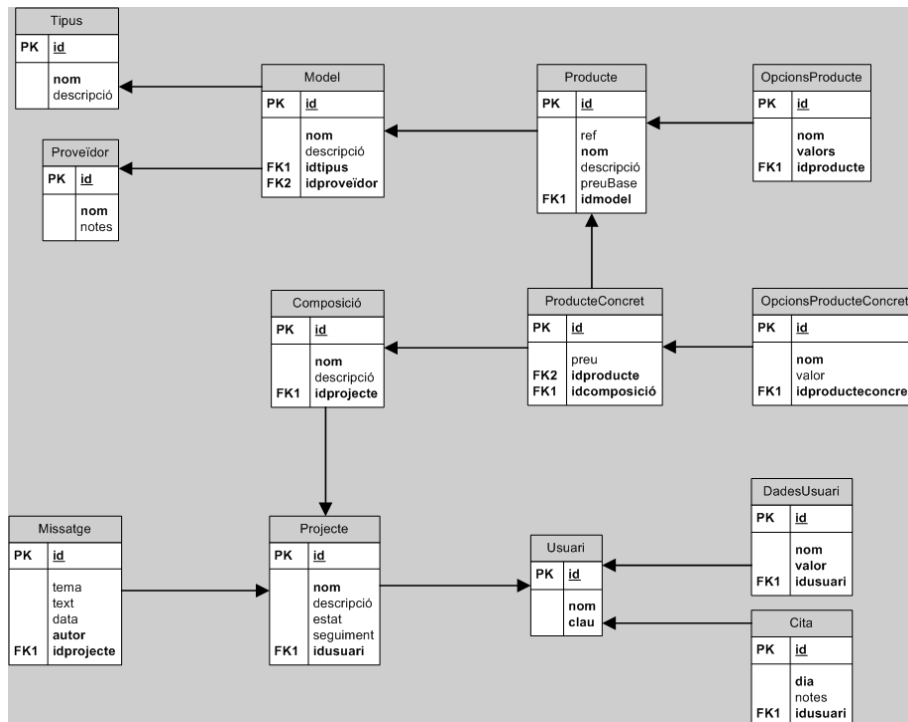
Per emmagatzemar la informació que hi ha en el model utilitzarem una base de dades, i un conjunt d'objectes per connectar a la base de dades i accedir a cadascuna de les seves taules.

En aquest apartat es presenta el disseny de la base de dades i els objectes gestors de la persistència. El fet d'utilitzar aquests objectes gestors de la persistència fa que puguem desconnectar el nostre model de les dades que conté ja que el model només utilitzarà aquests gestors, i funcionarà independentment de què aquests gestors consultin una base de dades local, remota, un fitxer de disc, o la memòria RAM.

4.2.1. Diagrama de la base de dades

Per dur a terme el disseny de la base de dades, el procés ha estat molt senzill. Partint del diagrama de classes presentat a l'inici d'aquest disseny, es converteixen totes les classes en entitats i s'afegeixen les claus externes per poder emmagatzemar les relacions.

El diagrama que es presenta a continuació està elaborat amb Ms Visio, que permet traduir-lo directament en un *script* de definició de base de dades en SQL (mitjançant la part de SQL anomenada DDL: Data Definition Language).



4.2.2. Gestors de la persistència

Per guardar les dades del model a la base de dades, tot deixant oberta la possibilitat de canviar de sistema d'emmagatzematge, hem utilitzat una sèrie d'objectes d'accés a dades.

La classe principal és *GestorBdd* que permet establir una connexió a la base de dades i crear sentències per interactuar-hi.

La resta d'objectes d'accés a dades són subclasses de *GestorBdd* que es cuiden de guardar la informació de cadascuna de les classes del model a la base de dades: *GestorCita*, *GestorComposició*, *GestorMissatge*, *GestorModel*, *GestorProducte*, *GestorProducteConcret*, *GestorProjecte*, *GestorProveedor*, *GestorTipus*, *GestorUsuari*.

Vegem, com a exemple, els mètodes que ens proporciona la classe *GestorProjecte*. La resta de classes ofereixen mètodes similars.

Llista de mètodes

`deleteProjecte(id)`, `getAllProjectes()`, `getAllProjectesByUsuari(id)`, `getProjecte(id)`, `newProjecte()`, `newProjecte(p)`, `saveProjecte(p)`

Mètodes heretats

connecta, creaStatement, preparaStatement

4.3. Altres aspectes

Per a implementar l'aplicació web, utilitzarem la tecnologia J2EE. Això vol dir que treballarem amb servlets i pàgines JSP. El fet d'escollir la tecnologia J2EE ens facilita àmpliament l'ús dels patrons que aquesta tecnologia defineix.

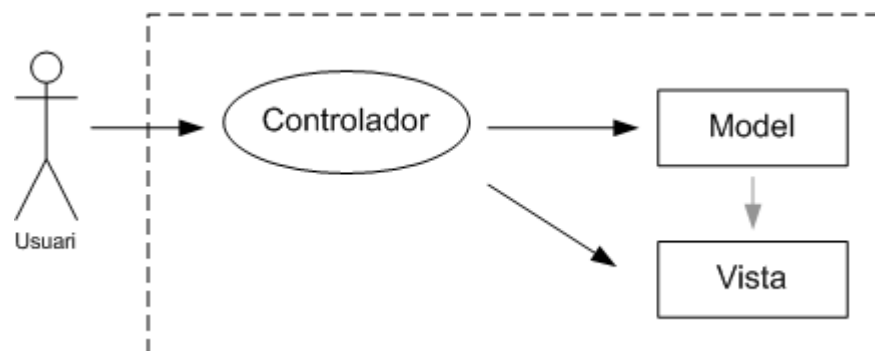
En aquest apartat detallarem quins patrons i com s'aplicaran en la implementació de l'aplicació, a part de presentar altres detalls com la reutilització que hi haurà, les excepcions que considerem, i una mostra d'interfícies gràfiques, per centrar idees sobre el que estem desenvolupant.

4.3.1. Patrons

Si entenem els patrons com un recull de millors pràctiques per fer una tasca en concret, o com una guia de resolució d'un problema determinat; l'ús de patrons fa que adoptem la millor solució, sense haver-nos d'esforçar excessivament a trobar-la.

En concret, utilitzarem una arquitectura MVC (Model-View-Controller) per implementar la nostra aplicació web. Aquest patró de disseny busca la separació entre les dades amb què treballa una aplicació (el model), i les vistes (view) que proporciona: la presentació d'aquestes dades.

El controlador és l'element que ho fa funcionar tot. Responent a les accions de l'usuari, accedeix al model per obtenir la informació requerida, i utilitza les vistes per presentar aquesta informació.



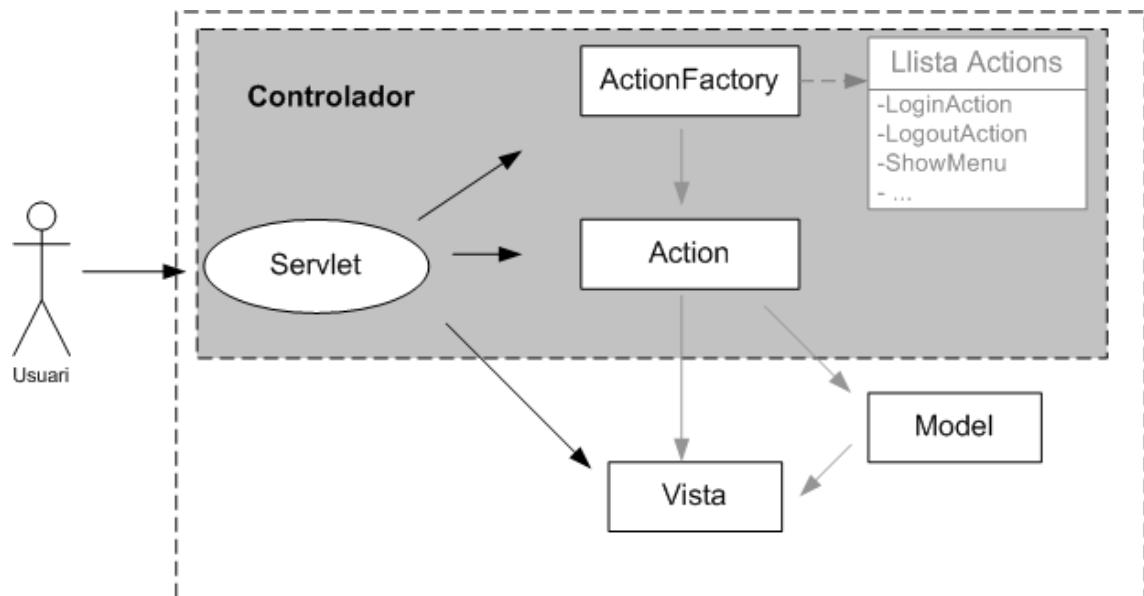
En el nostre cas, podem detallar una miqueta més aquest patró de disseny, perquè ja sabem que treballarem sobre la plataforma J2EE.

El model vindrà definit per un seguit de classes escrites en Java (veure el diagrama de classes a l'inici d'aquest document) i la seva informació es guardarà en una base de dades, les vistes (view) s'obtindran mitjançant pàgines JSP, i el controlador (controller) serà un servlet, seguint el patró de disseny *Front Controller*.

Donat el paper central que hem donat al controlador, entrem més a fons en el seu disseny. És molt important que puguem modificar fàcilment aquest controlador, ja sigui per afegir-li noves opcions, retirar-ne, modificar-ne, etc. , i això ens porta a dissenyar-lo de manera que sigui modular (separant clarament les diferents accions que ofereix a l'usuari) i escalable (que permeti afegir noves opcions).

En aquest cas, l'estratègia que seguim és definir una interfície *Action* que ens permetrà implementar per separat cadascuna de les accions que pot dur a terme el controlador, i una fàbrica d'accions (*ActionFactory*) que proporcionarà al controlador l'acció necessària en cada moment.

Vegem com queda el disseny.

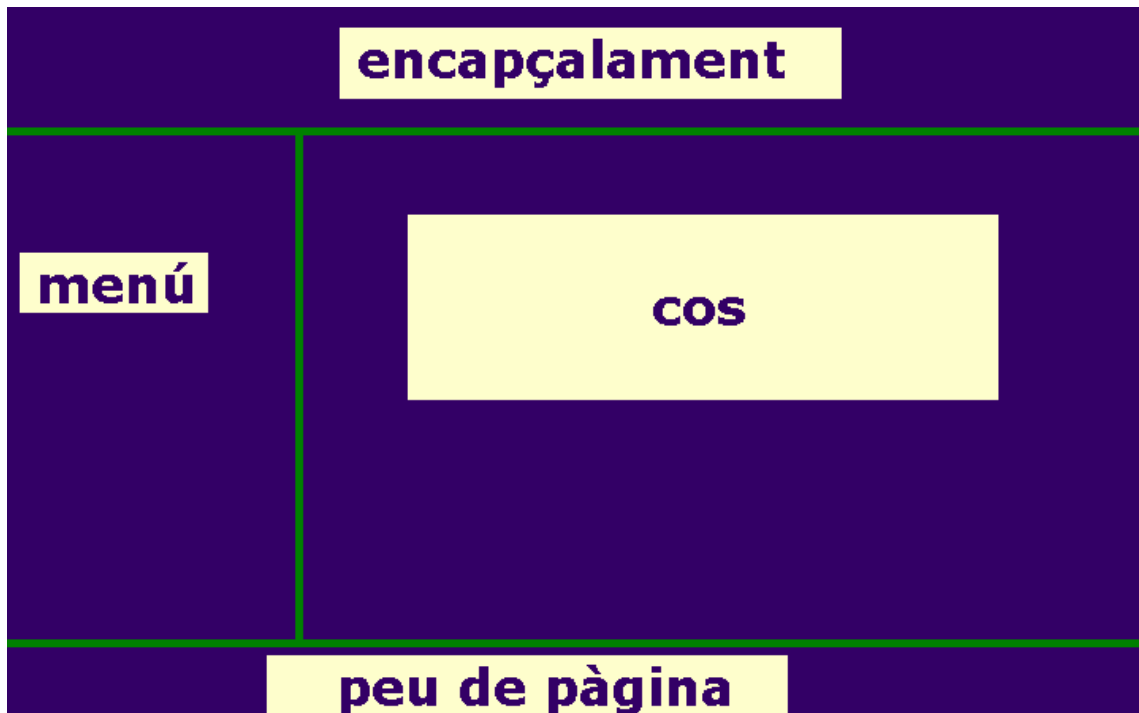


Així doncs, quan l'usuari fa una sol·licitud a l'aplicació, el servlet la recull i demana a la fabrica d'accions *ActionFactory* una acció *Action* que pugui resoldre la sol·licitud. Per a fer-ho, accedirà al model, d'on obtindrà les dades necessàries, i indicarà al servlet quina vista ha d'utilitzar per mostrar-les.

En aquest cas, el patró s'anomena *FrontController*. També s'utilitza el patró *Factory* en la seva implementació.

El patró *MVC* és el principal que s'utilitza en aquest disseny pel seu caràcter estructural o arquitectònic. De tota manera, no és l'únic. Un altre patró que s'utilitza és el patró *CompositeView* o vista composta, que ens permet generar la vista final a partir de diverses vistes. En aquest cas, ens interessem per poder mantenir una mateixa estructura en totes les vistes, i que aquesta sigui fàcilment modificable.

En aquest cas, l'estructura que utilitzarem és la següent:



4.3.2. Reutilització

Seguint la metodologia de la programació orientada a objectes, per implementar aquesta aplicació web, mirarem de fer servir la reutilització pels avantatges que suposa (proporciona trossos de l'aplicació ja implementats i provats, cosa que va en favor de la qualitat de l'aplicació; i ens estalvia força feina).

Hi ha diferents arguments que ens fan decantar-nos cap a la plataforma J2EE per implementar l'aplicació web, i una d'elles és la llibreria estàndard que proporciona el llenguatge Java. A l'hora de parlar de la reutilització, convé no oblidar que utilitzarem la llibreria estàndard de Java per diverses coses:

- Paquets habituals: les classes per al tractament de cadenes de text, per a càlculs numèrics, les classes contenidors d'objectes, ...
- Servlets i JSP, que ens permetran que la nostra aplicació s'integri en un servidor web compatible amb J2EE.
- JDBC, que ens permetrà connectar amb una base de dades qualsevol, tot oblidant les peculiaritats del sistema gestor de bases de dades utilitzat.

A més a més, també es reutilitzaran algunes classes que s'han desenvolupat (o simplement utilitzat) al llarg de la carrera d'enginyeria tècnica en informàtica de gestió, com classes de suport a la connexió a la base de dades, una classe per mantenir un log de les operacions que es duen a terme, classes de suport al xifratge (criptografia), ...

També convé destacar que utilitzarem dues eines que ja existeixen per a la fase d'implementació, i la posterior fase de desplegament de l'aplicació:

- Tomcat: un servidor web que segueix l'especificació J2EE. Com a principals avantatges en destaquem els fets de ser lliure i gratuït. Per ser gratuït, els costos econòmics queden a bastament reduïts. Per ser lliure i àmpliament utilitzat, es tracta d'un producte molt provat i, per tant, d'alta qualitat.
- MySQL: un sistema gestor de bases de dades. D'aquest sistema gestor de bases de dades també en destaquem el fet de ser lliure i gratuït, cosa que porta als dos avantatges comentats anteriorment.

4.3.3. Excepcions

Creiem que les excepcions que utilitzarem en aquesta aplicació web són de caire molt general i que, per tant, podem reutilitzar-ne diverses que ja existeixen.

Les excepcions que contemplarem són doncs, les següents:

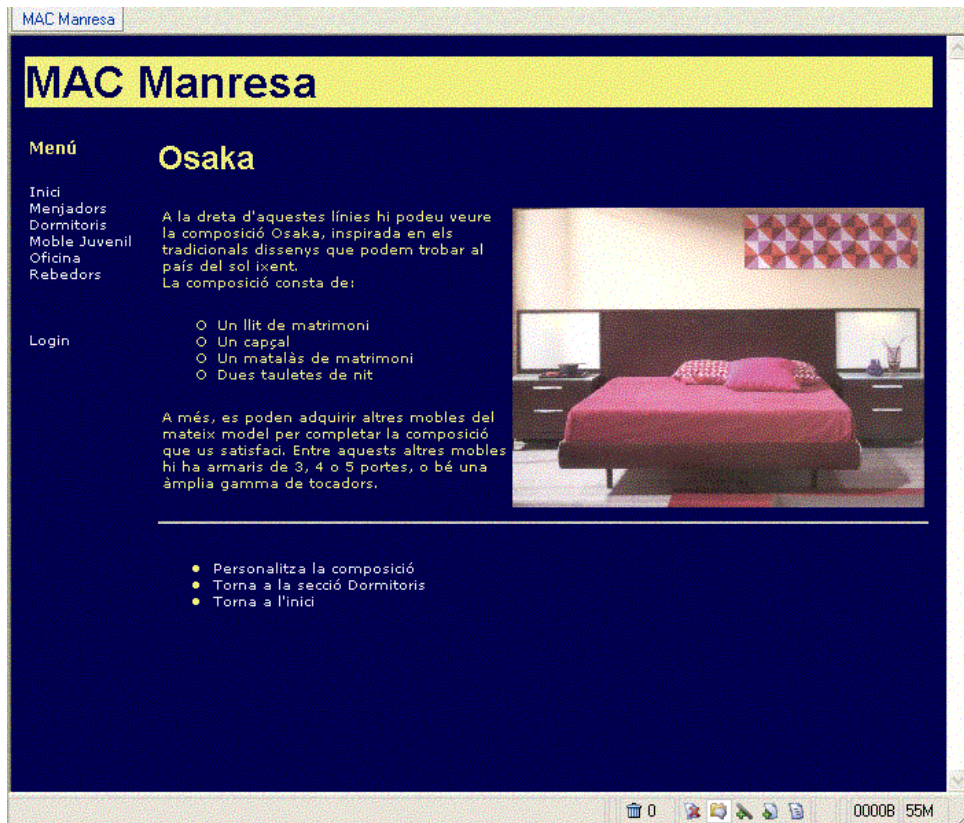
- Exception
 - IOException
 - SQLException
 - **LoginException**
 - **UserInputException**

Només les excepcions que estan en negreta són *noves*. Les altres ja pertanyen a la jerarquia d'excepcions de Java.

4.3.4. Interfícies gràfiques

A continuació hi podem trobar alguns prototipus de com volem que acabi sent l'aplicació web que hem dissenyat en aquest document.

Per començar, aquí hi tenim una mostra de com ha de ser el catàleg. En aquest cas estem veient una composició en concret.



Aquest segon exemple mostra la pàgina de projectes, on es mostren els diferents projectes i les opcions que s'ofereixen a l'usuari en referència al projecte



5. Implementació

En aquest apartat es presenta, en línies generals, la implementació que s'ha fet del disseny proposat en l'apartat anterior. En concret, explicitem com s'han aplicat els patrons de disseny presentats.

Totes les classes escrites en Java per a l'aplicació web s'han empaquetat en un arxiu *tfc.jar*, el contingut del qual detallarem a continuació. A partir d'aquí, presentarem com s'estructura realment l'aplicació web.

5.1.1. El paquet *tfc.jar*

Vet aquí l'estructura de paquets que hem desenvolupat per aquesta aplicació:

```
tfc
tfc.Catàleg
tfc.Catàleg.Actions
tfc.Projecte
tfc.Projecte.Actions
tfc.Usuari
tfc.Usuari.Actions
```

En el paquet *tfc* hi ha les classes bàsiques de l'aplicació.

- La interfície *Action* que determina com ha de ser una acció de l'aplicació web.
- La classe *ActionFactory* que implementa el patró *Factory* per a crear diverses accions (*Action*).
- La classe *GestorBdd* que implementa la funcionalitat mínima per connectar amb la base de dades i després interactuar-hi.
- La classe *SaludaAction* d'exemple: és una mostra d'implementació de la interfície *Action*.
- La classe *Servlet* que és el servlet que controla tota l'aplicació web.

Per detallar com s'ha implementat el patró MVC, a continuació es presenta un extracte del codi de les classes. El codi de l'aplicació no és el que es presenta, si no que és molt més robust, però aquí s'elimina la robustesa per transmetre més bé la idea.

La interfície *Action* es limita a definir dos mètodes:

```
public interface Action {
    public void execute(javax.servlet.http.HttpServletRequest req);
    public String getView();
}
```

La classe *ActionFactory* permet obtenir un objecte *Action* a partir del nom de l'acció que es vol dur a terme. Per a assignar cada nom d'acció a una classe en concret, s'utilitza un objecte *Properties* (una taula de dispersió), que es guarda en el fitxer *actions.prop*.

```
public class ActionFactory {
    private static java.util.Properties props=null;

    public static void init(String path) {
        ...
    }

    /** Retorna un objecte Action */
    public static Action getAction(String sAction) {
        Action action = null;
        sAction = props.getProperty(sAction);
        action = (Action) Class.forName(sAction).newInstance();
        return action;
    }
}
```

A partir de l'objecte *ActionFactory* ja podem presentar el codi del servlet, que quan rep una petició, esbrina quina és l'acció demanada, obté l'objecte *Action* apropiat gràcies a *ActionFactory*, l'executa i passa la sol·licitud a la vista apropiada.

```
public class Servlet extends HttpServlet {
    ...

    protected void processRequest(    HttpServletRequest request,
                                    HttpServletResponse response)
        throws ServletException, IOException {

        /** obté el valor del paràmetre action */
        String sAction = request.getParameter("action");
        String view;

        /** demana un objecte Action a ActionFactory */
        Action action = ActionFactory.getAction(sAction);

        /** executa l'acció desitjada */
        action.execute(request);
        view = action.getView();

        /** crea el dispatcher on enviar la sol·licitud */
        RequestDispatcher dispatcher =
            getServletContext().getRequestDispatcher(view);

        /** envia la sol·licitud al dispatcher */
        dispatcher.forward( request, response);

    }

    ...
}
```

Els paquets *tfc.Catàleg*, *tfc.Projecte* i *tfcUsuari* contenen les classes del model i els gestors de la persistència. Vegem la llista de classes:

tfc.Catàleg

Composició, GestorComposició, GestorModel, GestorProducte, GestorProducteConcret, GestorProveïdor, GestorTipus, Model, Producte, ProducteConcret, Proveïdor, Tipus

tfc.Projecte

GestorMissatge, GestorProjecte, Missatge, Projecte

tfcUsuari

Cita, GestorCita, GestorUsuari, Usuari

La resta de paquets (tfc.???.Actions) contenen totes les possibles accions, i estan classificats segons els elements del model sobre els que actuen.

5.1.2. L'aplicació web

Vet aquí el sistema d'arxius que configura l'aplicació web. Després de presentar-lo, hi ha els comentaris pertinents a diferents carpetes/arxius.

/ (directori arrel de l'aplicació web)

- **imgs**
 - (diferents imatges de l'aplicació)
- **jsps**
 - home.jsp
 - error.jsp
 - (altres vistes jsp de l'aplicació)
- **WEB-INF** (carpeta de configuració)
 - **lib**
 - mysql-connector-java-3.0.14-production-bin.jar (driver JDBC de MySQL)
 - tfc.jar (paquet comentat anteriorment)
 - web.xml (arxiu de configuració)
- actions.prop (configuració d'accions)
- bdd.prop (configuració de la base de dades)
- index.jsp (pantalla d'entrada a l'aplicació)
- style.css (fulla d'estil de l'aplicació)

L'arxiu actions.prop estableix les classes que han de respondre a cada acció demanada per l'usuari. Vet aquí un fragment de l'arxiu:

```
#Mostra usuaris  
showUsuari=tfc.Usuari.Actions.ShowUsuariAction  
showCita=tfc.Usuari.Actions.ShowCitaAction
```

```
#Edita usuaris
editUsuari=tfc.Usuari.Actions.EditUsuariAction
editDadesUsuari=tfc.Usuari.Actions.EditDadesUsuariAction
editCita=tfc.Usuari.Actions.EditCitaAction

#Crea usuaris etc
newUsuari=tfc.Usuari.Actions.NewUsuariAction
newCita=tfc.Usuari.Actions.NewCitaAction
login=tfc.Usuari.Actions.LoginAction
```

L'arxiu *bdd.prop* estableix els paràmetres de connexió a la base de dades: el nom d'usuari i la clau d'accés.

En el directori *imgs* s'hi posen les imatges de l'aplicació.

En el directori *jsp* hi ha totes les vistes de l'aplicació. Aquestes reben la informació que han de mostrar a través de beans: només interactuen amb les classes del model i alguna de Java (per exemple, *Enumeration*), però mai amb el servlet, o els gestors de la persistència.

La pàgina *home.jsp* és la pàgina mare, ja que determina l'estructura bàsica de la pàgina (capçalera, menú, cos, peu de pàgina), i mostra en el seu cos la pàgina que toca mostrar en cada moment.

5.1.3. Proves

Per a provar l'aplicació, s'han afegit mètodes executables a la majoria de classes susceptibles de dur problemes. Això inclou algunes de les classes del model i tots els gestors de la persistència.

En quan a les proves d'integració, convé destacar que l'aplicació ha estat *online* en el meu ordinador personal durant tres dies, en què diferents companys i amics l'han provat i han comunicat els errors que hi ha trobat.

Un cop passat el període de prova, crec que l'aplicació resultant és força robusta i que funciona correctament.

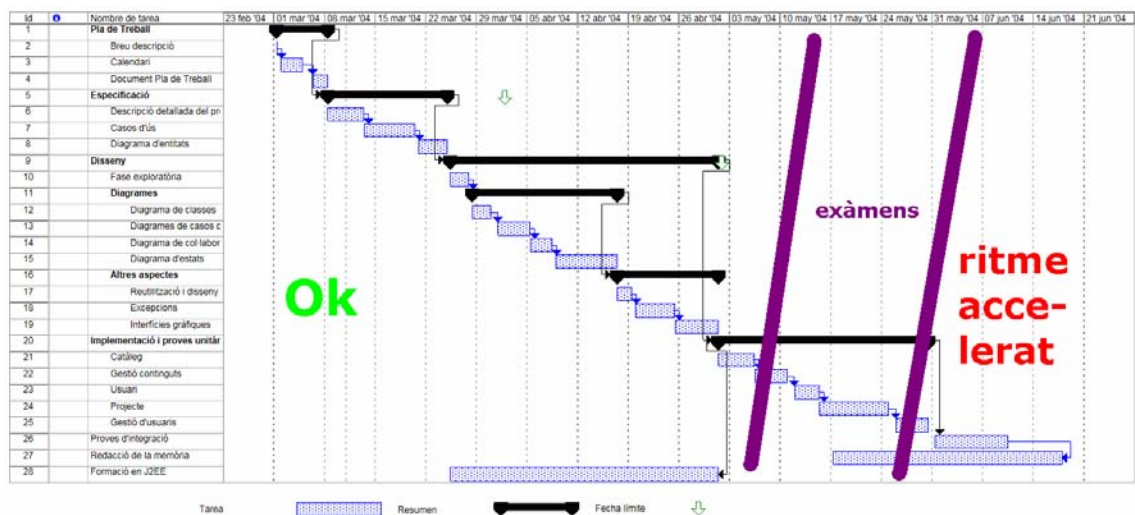
6. Conclusions

Un cop dut a terme el projecte de construcció d'una aplicació web per a una botiga de mobles, puc fer una valoració de com ha anat el projecte, comparant l'abast del projecte descrit inicialment i l'abast aconseguit al final; o bé comparant la planificació temporal establerta al principi, amb la temporalització real del projecte.

6.1. Temporalització

Si comencem parlant de la temporalització, cal dir que he seguit bastant el pla de treball establert al principi, sobretot en les primeres fases. La fase d'implementació però, s'ha barrejat amb l'època d'exàmens d'uns altres estudis que segueixo, i això ha fet que la feina s'hagi concentrat en els últims quinze dies. Tot i que, sense explicitar-ho en el pla de treball, ja havia previst aquest desajust temporal, potser hauria sigut convenient afegir alguna fita intermitja en la fase d'implementació.

Ho podem resumir en el següent diagrama comentat:



6.2. Abast

He de reconèixer que potser en la definició del projecte vaig ser més ambiciós del compte, i això ha portat a que l'abast real del projecte no fos el que s'havia previst.

De tota manera, estic satisfet d'haver-ho fet així, ja que ara puc assegurar que el programari desenvolupat està pensat per ser ampliat per acomplir els objectius plantejats al principi.

Així doncs, un cop comentat l'abast en general, podem entrar a estudiar el grau d'acompliment dels objectius i remarcar especialment les parts que s'haurien de millorar per acomplir els objectius plantejats. Per això, veiem els casos d'ús que s'han implementat.

Cas d'ús	Comentari
Consultar Producte	Implementat
Crear Composició	Implementat
Editar Composició	Implementat
Elaborar Pressupost	No implementat
Alta Tipus	Implementat
Baixa Tipus	Implementat en GestorTipus, però no com a acció.
Editar Tipus	Implementat
Alta Proveïdor	Implementat
Baixa Proveïdor	Implementat en GestorProveïdor, però no com a acció.
Editar Proveïdor	Implementat
Alta Producte	Implementat
Baixa Producte	Implementat en GestorProducte, però no com a acció.
Editar Producte	Implementat
Alta Usuari	Implementat
Baixa Usuari	Implementat en GestorUsuari, però no com a acció.
Editar Usuari	Implementat
Reservar Cita	Implementat
Login	Implementat
Alta Projecte	Implementat
Editar Projecte	Implementat
Baixa Projecte	Implementat
Enviar correu	Implementat
Llegir correu	Implementat

Així doncs, en resum, no s'ha implementat un dels 23 casos d'ús, i n'hi ha 4 que necessitarien acabar-los d'enllestir. De tota manera, el programari està preparat per afegir-los fàcilment.

El grau d'acompliment dels casos d'ús no és l'únic paràmetre a tenir en compte en l'acompliment del projecte. Per exemple, en aquest cas podem dir que els productes i les composicions es poden editar, però el procés per afegir-hi imatges és poc còmode (posar-les en un directori amb un nom en concret), i és un dels punts que caldria millorar.

Un requeriment no contemplat en la implementació és el fet de poder compartir fitxers entre l'usuari i l'empresa, ja siguin documents pdf, imatges, etc.

En resum, crec que el programari resultant d'aquest treball final de carrera és una molt bona base per fer la botiga de mobles d'una empresa, on caldria millorar la inclusió d'imatges de productes i composicions per fer la navegació més atractiva pel visitant.

La part de serveis oferts als clients o usuaris registrats, en canvi, trobo que és bastant complerta, i un molt bon espai de trobada entre l'empresa i els seus clients.

6.3. En resum

He començat aquesta memòria explicant que la principal motivació per haver escollit aquest treball de fi de carrera era el fet d'haver fet ja una pàgina web per una botiga de mobles, i voler-la millorar a bastament.

La meva valoració és que estic satisfet del treball realitzat, així com del resultat obtingut. I estic convençut que val la pena utilitzar-ho: seguiré treballant en el projecte per incorporar la major part de l'aplicació web desenvolupada a l'aplicació web actual.

Per exemple, oferir la possibilitat de muntar-se una composició i preparar un pressupost pot ser bastant interessant de cara al client, o la part de serveis oferts, que poden aportar valor afegit a la botiga de mobles.

7. Manuals

7.1.1. Manual d'instal·lació

Per poder instal·lar aquesta aplicació web en un servidor cal tenir dos programaris instal·lats prèviament. Es tracta del servidor Tomcat i del sistema gestor de bases de dades MySQL, ambdós oferts sota la fórmula de programari lliure.



Apache Tomcat és un servidor web que pot contenir servlets i es pot trobar a l'adreça <http://jakarta.apache.org/tomcat/>



MySQL és un sistema gestor de bases de dades i es pot trobar a l'adreça <http://www.mysql.com/>

Un cop tenim aquest programari en funcionament en el nostre equip servidor, la instal·lació del nostre programari és bastant senzilla. Primer cal crear la base de dades on s'emmagatzemarà la informació, i això ho aconseguim executant l'*script* de creació proporcionat: *CreaBdd.sql*.

El SGBD MySQL proporciona diferents mètodes per executar l'*script* de creació: des d'una interfície web com *phpMyAdmin*, o bé per línia de comandes mitjançant les ordres:

```
[usr]: "c:\[ruta]\mysql\bin\mysql.exe" -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20 to server version: 4.0.16-nt
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> \. CreaBdd.sql
Query OK, 1 row affected (0.00 sec)
...
Query OK, 1 row affected (0.00 sec)

mysql> \. OmpleBdd.sql
Query OK, 1 row affected (0.00 sec)
...
```

El segon *script*: *OmplBdd.sql* permet iniciar la base de dades amb uns valors d'exemple inicials.

Un cop hem creat la base de dades, només queda desplegar l'aplicació web en el servidor Tomcat (en un altre servidor J2EE també hauria de funcionar). Per això, el més fàcil és utilitzar les eines visuals que proporciona:

Upload a WAR file to install

Select WAR file to upload:

L'arxiu que hem de desplegar és el que es proporciona: *tfc.war*, que conté tota l'aplicació web.

Un cop desplegada l'aplicació, només queda configurar-la perquè pugui connectar-se a la base de dades, proporcionant els valors de nom d'usuari i clau d'accés a la base de dades en l'arxiu *bdd.prop*.

Per poder modificar la informació de l'aplicació web, l'*script* de creació es cuida de crear un usuari administrador amb nom d'usuari *admin* i clau d'accés *password*.

7.1.2. Manual d'usuari

En entrar a la pàgina web, se'ns presenten diversos tipus de mobiliari (dormitoris, menjadors, etc.) perquè escollim el tipus en què estem interessats.

Un cop escollit el tipus, se'ns presenten els diferents models disponibles d'aquell tipus, els quals podem anar consultant. I per cada model, se'ns presenten els diferents productes que en formen part.

Una altra possibilitat és consultar directament diverses composicions preparades pel personal de l'empresa. Això ho podem fer mitjançant el menú de l'empresa, amb l'opció: navega per composicions.



Un cop hem escollit una composició que ens agrada, podem escollir-la perquè passi a ser la nostra composició. A partir d'aquest moment, podem editar la composició: afegir-hi productes, esborrar-ne, personalitzar-los, etc.

No cal partir d'una composició predeterminada per muntar-nos la nostra pròpia composició a mida. Navegant entre tipus de mobiliari i models, sempre

podem afegir els diferents productes a la nostra composició. És com un carro de la compra, que sempre podrem consultar amb l'opció *La meva composició* del menú.

Mobles

Menú

- Login
- Navega per tipus de mobiliari
- Navega per composicions
- **La meva composició**

Eclipse Bàsic



Aquesta composició inclou els productes bàsics del model Eclipse per moblar el dormitori de casa vostra. La seva alta qualitat a un preu assequible el fa la tria més adequada per a la majoria de famílies.

[Afegeix sencera a la teva composició](#) (eliminant els altres productes)

Lista de productos de la composició

- Llit de matrimoni Eclipse
- Tauleta de nit Eclipse
- Tauleta de nit Eclipse
- Armari Eclipse

En afegir una composició sencera, la nostra composició anterior queda substituïda. Però si afegim els diferents productes que ens interessen, simplement es van sumant a la llista de productes de la nostra composició.

En consultar cada producte, ens mostra les seves dades bàsiques i les opcions de personalització que permet. Per exemple, el Tocador Eclipse permet escollir el número de calaixos.

The screenshot shows a web interface with a dark blue background and a yellow header bar containing the word 'Mobles'. On the left, there is a 'Menú' section with a yellow background, containing a 'Login' link and two navigation options: 'Navega per tipus de mobiliari' and 'Navega per composicions'. Below the menu is a 'La meva composició' section with a bullet point. The main content area is titled 'Tocador Eclipse' and includes the following information: 'Referència: EC023', 'Model: Eclipse', 'El tocador Eclipse que combina amb la resta de la gamma.', 'Preu base: 230.0', and 'Opcions:'. Under 'Opcions', there is a list item 'Calaixos' with a dropdown menu showing '3 calaixos'. At the bottom of the product details, there is a button labeled 'afegeix a la teva composició'.

Un cop que un producte passa a formar part de la nostra composició, ja podem personalitzar-lo. I només en aquest cas ho podem fer. En l'exemple d'aquí a sota hi veiem que l'armari Eclipse ens permet escollir el nombre de portes, el nombre de postades i el nombre de calaixos.

The screenshot shows a web interface similar to the previous one, with a dark blue background and a yellow header bar containing the word 'Mobles'. On the left, there is a 'Menú' section with a yellow background, containing a 'Login' link and two navigation options: 'Navega per tipus de mobiliari' and 'Navega per composicions'. Below the menu is a 'La meva composició' section with a bullet point. The main content area is titled 'Armari Eclipse' and includes the following information: 'Referència: EC011', 'Model: Eclipse', 'L'armari Eclipse, amb àmplies possibilitats de personalització perquè te'l facis tot teu.', 'Preu: 2400.0', and 'Opcions:'. Under 'Opcions', there is a list of three items: 'Postades' with a dropdown menu showing '4 postades', 'Portes' with a dropdown menu showing '3 portes', and 'Calaixos' with a dropdown menu showing '3 calaixos'. Below these options is a button labeled 'D'acord'. At the bottom of the product details, there are two buttons: 'esborra el producte' and 'torna a la composició'.

Per poder treballar còmodament amb les composicions, també tenim l'opció d'esborrar el producte, i així el retirem de la composició.

Així doncs, ja hem vist com podem navegar per la pàgina web, tot muntant-nos la nostra pròpia composició, i personalitzant els productes perquè satisfacin les nostres necessitats.

Però nosaltres volíem oferir un servei més proper a l'usuari. Mitjançant un procés molt senzill de registre, podem oferir altres possibilitats com:

- treballar amb diferents composicions,
- guardar-les per poder-hi seguir treballant en la propera visita a l'aplicació web
- organitzar les composicions per projectes (projecte de moblar el dormitori, de reformar la cuina, etc.)
- enviar i rebre missatges al personal de l'empresa
- demanar cita per no haver de fer cua en visitar la botiga.

En la propera pàgina hi veiem com fer el procés de login en cas de ser un usuari registrat, i un enllaç per registrar-se, junt amb la llista d'avantatges de registrar-se.

The screenshot shows a web application interface with a dark blue background and yellow highlights. At the top, a yellow banner contains the word "Mobles" in bold black text. Below this, the page is divided into two main sections. On the left, a yellow header "Menú" is followed by a list of navigation options: "Login", "Navega per tipus de mobiliari", and "Navega per composicions". On the right, a "Login" section features two input fields: "Nom d'usuari:" with the text "joan" and "Clau d'accés:" with a masked password of ten dots. A yellow "Login" button is positioned below the password field. Underneath the login form, a question asks "Vols registrar-te a la pàgina web? sí". Below this, a yellow banner reads "Avantatges de registrar-te", followed by a list of three benefits: working with multiple compositions, organizing projects, and scheduling appointments. At the bottom, a call to action says "A què estàs esperant? registra't ara!".

El procés de registre és molt senzill: només cal seguir l'enllaç (pitjant sobre sí, o ara) i escollir un nom d'usuari i una clau d'accés. La clau d'accés es guarda xifrada i, per tant, ningú pot conèixer-la.

Un cop ja ens hem registrat, ens apareix el menú d'usuari, on hi podem trobar les nostres dades, i afegir-n'hi, editar-les, etc. També hi trobem les opcions per crear noves cites, o nous projectes.

En cas que ja haguem concertat alguna cita, o creat nous projectes, ens apareix una llista exhaustiva de les cites i els nostres projectes, com podem veure en l'exemple a continuació.

The screenshot shows a web application interface for 'Mobles'. At the top, there is a yellow header with the word 'Mobles' in bold black text. Below the header, the page is divided into two main columns. The left column contains a 'Menú' section with a yellow background, listing navigation options: 'Menú d'usuari', 'Navega per tipus de mobiliari', and 'Navega per composicions'. Below this, there is a section for 'La meua composició'. The right column displays the user profile for 'joan', including contact information: 'Telèfon mòbil: 606256158', 'NIF: 39371793W', and 'Adreça: Pic Aneto, 38 de Santpedor'. Below the profile information, there are three action links: 'edita l'usuari', 'crea una nova cita', and 'crea un nou projecte'. Further down, there are two sections with yellow headers: 'Llista de cites de l'usuari' and 'Llista de projectes de l'usuari'. The 'Llista de cites de l'usuari' section contains one entry: '17/06/04 18:00 Cita per parlar del projecte Dormitori'. The 'Llista de projectes de l'usuari' section contains one entry: 'Dormitori', with a description: 'Estat: estudiant composicions. Aquest projecte és per moblar el dormitori de matrimoni de casa meua. Espero poder estudiar diferents composicions de mobles modern i escollir la que consideri més atractiva, tenint en compte també el seu preu.'

I un cop hem vist el menú d'usuari, al qual sempre podrem accedir mitjançant el menú, anem a veure com podem treballar amb els projectes.

Tota la informació d'un projecte es resumeix en el menú del projecte, on hi podem trobar el nom la descripció que li hem donat, l'estat en què es troba, i informació sobre el seguiment del projecte: si finalment decidim comprar alguna de les composicions, podrem seguir el procés que duu a terme la botiga per Internet (passar una comanda al proveïdor, rebre el gènere, establir un dia i hora de muntatge).

En un projecte hi podem tenir diverses composicions com a possibles solucions al problema que pretén resoldre el projecte: moblar algun espai en concret de l'habitatge.

A més a més, en relació a un projecte es poden enviar i rebre missatges als / dels empleats de l'empresa comentant les composicions, demanant algun ajust de preus, ...

En el menú del projecte hi trobem doncs, les opcions per crear composicions i enviar missatges, així com una llista dels missatges enviats i rebuts en relació al projecte, i les composicions que s'estan estudiant.

Menú

- Menú d'usuari
- Navega per tipus de mobiliari
- Navega per composicions
- La meua composició**

Dormitori

Usuari: joan
Aquest projecte és per moblar el dormitori de matrimoni de casa meua. Espero poder estudiar diferents composicions de mobles modern i escollir la que consideri més atractiva, tenint en compte també el seu preu.
Estat: estudiant composicions
Seguiment
Creat el dimarts 15 de juny.
Composició afegida el dimarts 15 de juny
Composició afegida el dimecres 16 de juny

edita el projecte
crea un nou missatge
crea una nova composició

Llista de missatges del projecte

- 17/06/04 02:08. Composició Eclipse enviat

Llista de composicions del projecte

- Llit senzill
Composició molt senzilla que consta d'un sol llit. És la més bàsica.
- Eclipse Bàsic
Aquesta composició inclou els productes bàsics del model Eclipse per moblar el dormitori de casa vostra.
La seva alta qualitat a un preu assequible el fa la tria més adequada

La mateixa aplicació web contempla un usuari administrador que pot modificar els elements del catàleg (tipus, models, productes), així com les composicions que trobem per defecte.

Aquest usuari administrador també pot consultar els projectes i composicions de tots els usuaris, enviar-los missatges, modificar els preus dels productes, etc.

Per permetre modificar la informació de l'aplicació web, l'*script* de creació es cuida de crear aquest usuari administrador amb nom d'usuari *admin* i clau d'accés *password*.

A més, l'usuari administrador pot afegir imatges a les composicions. Només li cal consultar el codi identificador de la composició i posar una imatge en la carpeta `imgs/` anomenada `comp[id].jpg`. Per exemple, `comp3.jpg`, per la composició amb identificador 3.

8. Recursos

8.1. Bibliografia

- Andrew Patzer. *JSP. Ejemplos prácticos*. Anaya Multimedia (El llibre presenta l'arquitectura MVC, tot detallant els diferents patrons)
- John Zukowski. *Programación en Java 2*. Anaya Multimedia
- David Tansley. *Create dynamic web pages using PHP and MySQL*. Addison Wesley (Interessant per la informació sobre MySQL)

8.2. Recursos a Internet

- <http://java.sun.com/> La web de Sun amb molta informació sobre el llenguatge Java, la plataforma J2EE, els patrons de disseny i el programari per descarregar.
- <http://jakarta.apache.org/tomcat/> La pàgina web del projecte Tomcat amb molta informació sobre l'Apache Tomcat i el programari per descarregar.
- <http://jakarta.apache.org/struts/> Un esquelet per desenvolupar aplicacions segons el patró MVC. La seva presentació del patró és molt exhaustiva i bastant didàctica.
- <http://www.w3schools.com/> Una pàgina amb un munt de tutorials sobre llenguatges utilitzats en el world wide web: html, css, javascript, ...