

Creació d'un entorn web de treball per a una biblioteca
Fent ús de Struts, Tiles i Hibernate

Rafael Alonso Jurado
ETIG

Abel Almazan Fernandez

16 de Juny de 2005

Resum del Treball de Final de Carrera

El següent projecte final de carrera vol assolir la creació d'un entorn web de treball per a una Biblioteca. L'aplicació al món real es basa en un mòdul administratiu, que podrà ser utilitzat pels treballadors de la pròpia Biblioteca i un mòdul que servirà de consulta als socis.

L'anàlisi i disseny es basa en el paradigma de l'orientació a objectes dins d'una arquitectura distribuïda.

La implementació s'ha fet amb llenguatge Java dins de l'arquitectura J2EE que proporciona tot el necessari per poder construir una aplicació robusta, escalable i altament reutilitzable, característiques que avui dia són molt importants i imprescindibles dins de qualsevol aplicació empresarial a Internet.

El llenguatge que s'ha fet servir per a la implementació de l'aplicació ha estat Java, dins de l'arquitectura J2EE. S'ha triat aquesta tecnologia per tal de poder assolir les característiques fonamentals d'una bona aplicació empresarial a Internet: ésser una aplicació robusta, escalable i altament reutilitzable.

Afinant més, ens agradaria comentar que en aquest TFC hem utilitzat diversos patrons de disseny i frameworks. En aquest cas, s'ha utilitzat el patró MVC a partir de la utilització del bastiment Struts, el que s'ha integrat amb el framework Tiles. Gràcies al seu ús, hem pogut separar les diferents funcionalitats entre components, amb la finalitat de desacoblar el codi d'accés a les dades, el codi de la lògica de negoci i el codi de presentació. Tot això ens dona un grau molt més alt de reutilització i reducció de codi duplicat. D'aquesta manera és molt més senzilla de mantenir, reutilitzar o modificar.

Per la persistència de dades remarcar que hem fet ús d'Hibernate que té el gran avantatge d'aïllar totalment el motor de base de dades ja que tracta directament amb objectes. D'aquesta manera es té un codi molt més net i molt més orientat al paradigma de l'orientació a objectes.

1 INDEX

2	INTRODUCCIÓ	6
2.1	Justificació del TFC: punt de partida i aportació	6
2.2	Objectius del TFC	7
2.3	Enfocament i mètode seguit	8
2.4	Planificació del projecte	9
2.5	Productes obtinguts	12
2.6	Descripció de la resta de capítols de la memòria	12
3	ESPECIFICACIÓ I ANÀLISI DELS REQUERIMENTS	13
3.1	Introducció	13
3.2	Descripció del projecte	13
3.3	Requeriments funcionals	15
3.4	Especificacions de les funcionalitats per subsistema	16
3.4.1	Subsistema principal	16
3.4.2	Subsistema de gestió d'usuaris	18
3.4.3	Subsistema de gestió de documents	22
3.4.4	Subsistema de gestió dels préstecs	24
3.4.5	Subsistema de configuració	27
3.5	Interfície gràfica	28
3.5.1	Flux de pantalles	28
3.5.1.1	Inici de l'aplicació	28
3.5.1.2	Subsistema gestió d'usuaris	29
3.5.1.2.1	Alta de socis	29
3.5.1.2.2	Baixa i modificació de socis	30
3.5.1.2.3	Alta, Baixa i Modificació d'empleats	31
3.5.1.3	Subsistema de gestió de documents	32
3.5.1.3.1	Alta de documents	32
3.5.1.3.2	Baixa i modificacions de documents	32
3.5.1.4	Subsistema de gestió de préstecs	33
3.5.1.4.1	Renovacions de préstecs per part del soci	33
3.5.1.4.2	Reserves de documents per part del soci	34
3.5.1.4.3	Préstecs i Reserves de documents	35
3.5.1.4.4	Tornar préstecs i renovacions de documents	35
3.5.1.5	Subsistema de configuració	36
3.6	Requeriments de maquinari i de programari	37
3.7	Seguretat	37
3.8	Futures funcionalitats en properes versions	38
4	Disseny de l'arquitectura de l'aplicació	39
4.1	Introducció	39
4.2	Arquitectura J2EE	39
4.3	Paradigma "MVC" - Struts	41
4.4	El Component Controlador de Struts	44
4.5	Components de Vista de Struts	46
4.5.1	JavaScript i fulls d'estil	46
4.5.2	Ús de Tiles	46

4.5.3	Validació de formularis	49
4.6	Components de Model de Struts	51
4.6.1	Disseny de la persistència	51
4.6.1.1	Disseny de la base de dades	53
4.6.1.1.1	Diagrama ER	54
4.6.1.1.2	Disseny lògic de la base de dades	55
4.6.1.1.3	Taula Biblioteques	56
4.6.1.1.4	Taula Usuaris	56
4.6.1.1.5	Taula Administradors	57
4.6.1.1.6	Taula Empleats	57
4.6.1.1.7	Taula Socis	57
4.6.1.1.8	Taula Documents	58
4.6.1.1.9	Taula Video	58
4.6.1.1.10	Taula Audio	59
4.6.1.1.11	Taula Revistes	59
4.6.1.1.12	Taula Llibres	60
4.6.1.1.13	Taula Reserves	60
4.6.1.1.14	Taula Préstecs	60
4.6.1.2	Disseny de classes	61
4.6.1.2.1	Introducció	61
4.6.1.2.2	Diagrama estàtic de classes	61
4.6.1.2.3	GestorDisc	63
4.6.1.2.4	GestorUsuaris	64
4.6.1.2.5	GestorDocuments	65
4.6.1.2.6	Gestor Préstecs	66
4.6.1.2.7	Gestor Biblioteques	67
4.6.1.2.8	Biblioteca	68
4.6.1.2.9	Document	68
4.6.1.2.10	Llibre	68
4.6.1.2.11	Audio	69
4.6.1.2.12	Video	69
4.6.1.2.13	Revistes	69
4.6.1.2.14	Usuari	70
4.6.1.2.15	Administrador	70
4.6.1.2.16	Empleat	70
4.6.1.2.17	Soci	71
4.6.1.2.18	Reserva	71
4.6.1.2.19	Préstec	71
5	Implementació	72
5.1	Requeriments del programari	72
5.2	Eines de desenvolupament	72
6	Valoració econòmica	73
7	Conclusions	74
8	Glossari	75

9	Bibliografia	76
10	Annex 1. Manual d'usuari	77
10.1	Introducció	77
10.2	Manual del soci	77
10.3	Manual de l'empleat	80
10.3.1	Gestor d'usuaris	81
10.3.2	Gestor de documents	83
10.3.3	Gestor de préstecs	85
10.4	Manual de l'administrador	87
11	Annex 2. Sentències SQL	88

2 INTRODUCCIÓ

2.1 *Justificació del TFC: punt de partida i aportació*

La idea original es fer un treball de fi de carrera totalment funcional i de gestió amb un tecnologia actual i amb futur. Es basaven en unes característiques principals que l'aplicació tindria que tenir:

- Escalable
- Robusta
- Reutilitzable
- Programada orientat a objectes
- Que permeti una estructura distribuïda

Es per això que s'han tingut diferents opcions per triar i al final ens vam decidir per l'arquitectura J2EE ja que dona moltes facilitats pel desenvolupament d'aplicacions empresarials. Dins d'aquest marc de treball i veient el gran ventall de possibilitats que dona aquesta arquitectura, vam decidir utilitzar el framework Struts pel desenvolupament de l'aplicació que està dins del paradigma MVC (model-vista-controlador). Tot això ho vam completar amb el framework de persistència Hibernate que ens dona la guinda a una futura aplicació amb les característiques que parlàvem anteriorment.

Tota una tecnologia al nostre abast per desenvolupar una aplicació web que permeti controlar tota una biblioteca. Tots els coneixements adquirits durant la carrera els necessitarem per afrontar aquest TFC d'Enginyeria Tècnica en Informàtica Gestió. Hem de comentar, que tota la temàtica de programació distribuïda, components, patrons, etc., no s'imparteixen durant la nostra carrera i, per tant, han sigut tots coneixements nous que hem hagut d'aprendre.

Es per això, que volem després de realitzar el TFC arribar a tenir un bons coneixements de programació distribuïda i del desenvolupament d'aplicacions web de gestió, que sumem a tots els coneixements adquirits durant l'Enginyeria com són la programació Orienta a l'Objecte, Enginyeria del programari, o Tècniques de desenvolupament del programari.

Per tant, l'aportació d'aquest TFC s'aplica a la tecnologia J2EE i a la combinació de tots els coneixements teòrics dels que partim i implementar-los en el patró MVC utilitzant el framework Struts.

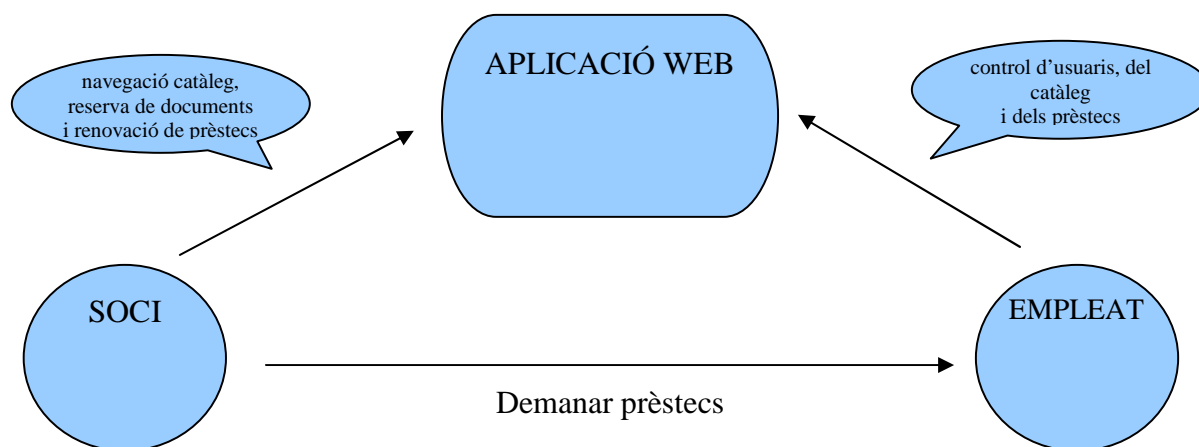
2.2 Objectius del TFC

L'objectiu del projecte que es vol realitzar es crear una aplicació web que permeti als treballadors d'una biblioteca administrar informàticament tots els documents (llibres, vídeos, àudios, etc.). A més, es voldria tenir un control total sobre el registre d'usuaris així com de tot el que comprèn els préstecs de documents.

Per posar un exemple, un treballador de la biblioteca podria enregistrar per primer cop a una persona que vol ser sòcia. En aquest moment, aquesta persona podria agafar qualsevol document i agafar-ho en préstec. El treballador, gràcies a l'aplicació creada, enregistraria aquest préstec.

A part de ser una eina d'administració, voldríem també que fos una eina per l'usuari. Voldríem afegir la funcionalitat de que l'usuari pogués consultar des de casa el catàleg de la biblioteca, així com fer diverses operacions després d'una autenticació. Aquestes operacions podrien ser, per exemple, les de renovació de préstecs i reserva de documents.

A continuació s'exposa un gràfic on es pot veure tot més clar:



Ja exposada la funcionalitat de l'aplicació web, hem de dir que com objectius generals volem que el nostre projecte tingui com a característica principal la senzillesa ja que serà una aplicació bàsicament d'administració. Això vol dir que els treballadors han de tenir un entorn fàcil i clar. A més, ens interessa molt que aquesta aplicació sigui robusta i sigui fàcilment reutilitzable per altres tipus de projectes que tinguin característiques similars.

A nivell més tècnic, comentar que volem implementar un patró MVC (modelo-vista-controlador) basat en Jakarta Struts. Struts és una estructura d'aplicació Web de codi obert dissenyat utilitzant el servlet i las API JSP. El motiu d'utilitzar un patró MVC és dissecionar la lògica de negoci, la lògica de control i el codi de presentació de

l'aplicació, permetent millorar la seva posterior utilització i el seu manteniment.

Volem integrar també Hibernate en el nostre projecte que ens proporcionarà altra eina de codi obert de bones prestacions, documentació i estabilitat. D'aquesta manera volem tenir un motor entremig entre la base de dades i el nostre procés de negoci.

Organització de l'equip

L'equip que desenvoluparà aquest projecte està format per Rafael Alonso i Abel Almazan. La organització de l'equip és molt senzilla i es detalla a continuació.

Rafael Alonso, 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'Abel.

Recursos de programari

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

- El campus de la UOC permetrà la comunicació entre estudiant i consultor.
- La suite informàtica Microsoft Office permetrà redactar els diferents documents del projecte.
- L'entorn integrat de desenvolupament Eclipse permetrà dur a terme les fases d'implementació i proves de l'aplicació web.
- La base de dades Oracle serà la que s'utilitzarà per desar totes les dades de l'aplicació.

2.3 Enfocament i mètode seguit

L'enfocament principal que es volia tenir per aquest TFC és seguir amb tot l'aprenentatge del desenvolupament de programari fent ús de totes aquelles tècniques de les que teníem coneixements i ampliar-les amb tot allò que tenim que aprendre per poder aconseguir els objectius proposats. Utilitzar la tecnologia J2EE amb Struts, Tiles, Hibernate, etc., és una dels punts forts del TFC i poder aconseguir una aplicació escalable, robusta i reutilitzable.

La gran quantitat de coneixements a adquirir és una de les dificultats principals del projecte, i la seva aplicació a un projecte real. L'estratègia que s'ha seguit és clara i es mou per adquirir tots els coneixements que ens falta a partir de nous estudis ajudats pels coneixements que ja teníem. A partir d'aquest moment, aplicar-ho tot a l'aplicació tenint en compte els objectius finals.

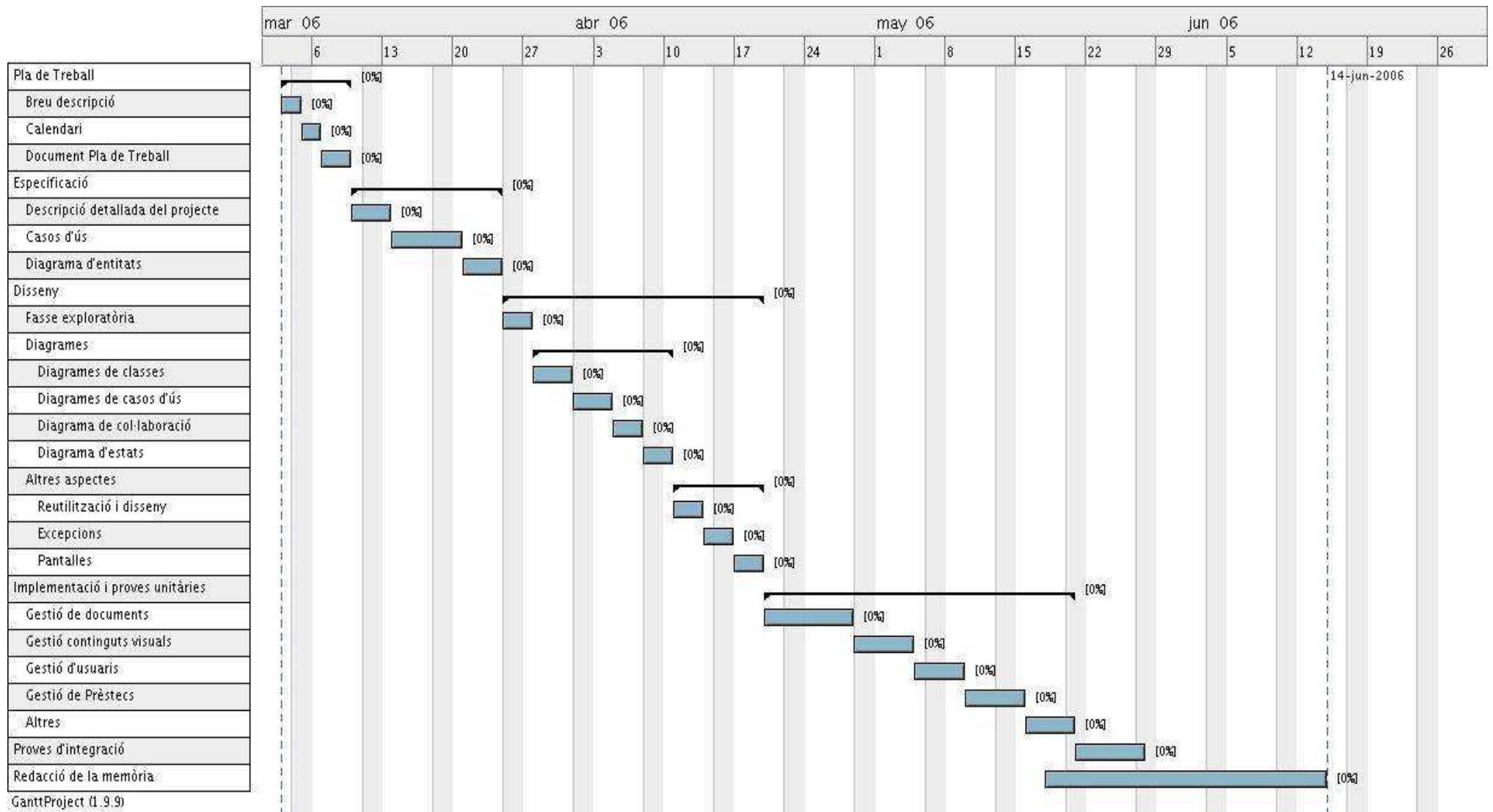
2.4 Planificació del projecte

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 Treballador
- 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
 - Pantalles
- Implementació i proves unitàries
 - Gestió de documents
 - Gestió continguts visuals
 - Gestió d'usuaris
 - Gestió de Préstecs
 - Altres
- Proves d'integració
- Redacció de la memòria
- Formació en J2EE

<i>Id</i>	<i>Tasca</i>	<i>Duració</i>	<i>Inici</i>	<i>Final</i>	<i>Predec.</i>
1	Pla de Treball	7	03/03/2006	09/03/2006	
2	Breu descripció	2	03/03/2006	04/03/2006	
3	Calendari	2	05/03/2006	06/03/2006	2
4	Document Pla de Treball	3	07/03/2006	09/03/2006	3
5	Especificació	15	10/03/2006	24/03/2006	1
6	Descripció detallada del projecte	4	10/03/2006	13/03/2006	
7	Casos d'ús	7	14/03/2006	20/03/2006	
8	Diagrama d'entitats	4	21/03/2006	24/03/2006	
9	Disseny	22	25/03/2006	24/04/2006	5
10	Fase exploratòria	3	25/03/2006	27/03/2006	
11	Diagrames	13	28/03/2006	11/04/2006	
12	Diagrames de classes	4	28/03/2006	01/04/2006	
13	Diagrames de casos d'ús	4	01/04/2006	05/04/2006	
14	Diagrama de col·laboració	3	05/04/2006	08/04/2006	
15	Diagrama d'estats	3	08/04/2006	11/04/2006	
16	Altres aspectes	9	11/04/2006	20/04/2006	
17	Reutilització i disseny	3	11/04/2006	14/04/2006	
18	Excepcions	3	14/04/2006	17/04/2006	
19	Pantalles	3	17/04/2006	20/04/2006	
20	Implementació i proves unitàries	31	20/04/2006	23/05/2005	9
21	Gestió de documents	9	20/04/2006	29/04/2006	
22	Gestió continguts visuals	6	29/04/2006	05/05/2006	
23	Gestió d'usuaris	5	05/05/2006	10/05/2006	
24	Gestió de Préstecs	6	10/05/2006	16/05/2006	
25	Altres	5	16/05/2006	21/05/2006	
26	Proves d'integració	7	21/05/2006	28/05/2006	20
27	Redacció de la memòria	28	18/05/2006	15/06/2006	9
28	Formació J2EE	30			



2.5 Productes obtinguts

El producte que neix de la implementació d'aquest projecte conté una web d'arquitectura J2EE que es compon de diferents fitxers, documents, etc. Aquests són:

- Fitxer de distribució .war que conté tota la web de l'aplicació més tot el model compilat. Totalment instal·lable per fer servir en un servidor d'aplicacions o contenidor web compatible amb J2EE.
- Codi font de totes les classes utilitzades
- Els fitxers de desplegament i de configuració utilitzats, conjuntament amb les llibreries necessàries pel seu funcionament.
- Els fitxers de creació de taules i inserció de dades necessàries per poder utilitzar a la Base de dades
- Un Manual que s'adjunta amb aquest document com apèndix.

2.6 Descripció de la resta de capítols de la memòria

En els següents capítols comentarem tot l'aspecte teòric d'especificació i anàlisi dels requeriments així com tot el disseny que hem confeccionat per poder donar solució als objectius proposats al principi.

3 ESPECIFICACIÓ I ANÀLISI DELS REQUERIMENTS

3.1 *Introducció*

En aquest capítol especificuem i documentem tots els requisits que necessitem per construir el programari que hem definit anteriorment. S'inclou tot l'anàlisi necessària per poder traduir els requisits en especificacions molt més formals (diagrames d'UML). Això facilitarà tot el posterior desenvolupament de disseny, així com la identificació de les classes fonamentals i l'expressió dels casos d'ús.

Comencem un estudi molt més global per anar després analitzant els diferents subsistemes del que es compon l'aplicació.

3.2 *Descripció del projecte*

El nostre projecte és una aplicació d'administració de Biblioteques que doni suport a la gestió de documents (llibres, revistes, àudio, vídeo, etc.), gestió de socis i a la gestió de préstecs, renovacions i reserves. A més de la part administrativa, volem també que doni accés als usuaris i els serveixi per interactuar amb els catàlegs de la biblioteca, així com fer funcions que no necessitin del personal propi de la biblioteca.

L'aplicació necessitarà d'un servidor web J2EE que executarà l'aplicació. A més, necessitem una base de dades per poder tenir persistència. En aquest cas, farem ús de Oracle 10g ja que és una de les més potent del mercat actual de base de dades. I per que l'usuari pugui fer servir l'aplicació faran ús de qualsevol navegador per poder interactuar amb ella. Necessitaran una direcció URL per poder accedir a la plana web principal. El fet de tenir l'aplicació web en un servidor J2EE i fer ús de navegadors com clients ens dona la facilitat de que des de qualsevol lloc una persona es pugui connectar. D'aquesta manera simplement tindriem que fer una instal·lació del servidor i no de tots els clients que vulguin accedir.

Haurà tres tipus d'usuaris: El personal de l'empresa que són aquells que portaran tota la gestió de la biblioteca (gestió de préstecs i gestió de socis). Els socis que seran aquells que rebin els préstecs. A més, podran fer algunes funcions bàsiques com fer renovacions dels documents que tinguin en préstec i fer reserves d'altres documents que tinguin interès en agafar. I, per últim, l'administrador que serà qui podrà gestionar qualsevol usuari i podrà configurar l'aplicació.

L'administrador haurà d'instal·lar un servidor web J2EE i la base de dades d'Oracle com a únics requisits perquè funcioni. Tindrà que crear una estructura de base de

dades que es donarà amb l'aplicació. Una vegada fet això, s'haurà de configurar l'aplicació perquè funcioni al servidor i perquè connecti a Oracle. En aquest moment haurà de donar d'alta els diferents empleats de la Biblioteca. Llavors, ja tindrem el sistema operatiu per a que els empleats puguin començar a inserir dades sobre documents i sobre socis, així com a fer les diferents funcions dels préstecs.

La aplicació tractarà de manera diferent als usuaris. Principalment perquè té diferents funcions per cadascú. En el cas dels socis, podran entrar per la plana principal. A més, tindran un formulari apart per poder autenticar la seva sessió i poder renovar documents ja prestats, reservar documents o veure/modificar les seves dades personals (algunes no pot canviar-les).

En canvi, l'empleat accedirà des d'una plana diferent creada expressament per a ells on tindran que autenticar-se per poder entrar a l'aplicació. Allà és on tindrà totes les funcionalitats disponibles per poder gestionar la biblioteca. L'entrada per l'administrador serà la mateixa que la de l'empleat, però se li afegiran dos funcionalitats més ja que podrà gestionar empleats i configurar l'aplicació.

La recerca de documents es farà mitjançant un formulari. Aquí es podrà fer una recerca per ISBN, Títol, Autor, Matèria o donant una Paraula Clau (que farà la recerca en tots els camps possibles). A més, es podrà fer una recerca avançada combinant dos recerques simples amb la lògica I, O o NO. Una vegada trobat el document en qüestió es podria, a part de consultar-ho, donar de baixa o modificar les seves dades.

Per donar d'alta documents tindrem un altre formulari on l'empleat tindrà que inserir les dades del document. Per cada tipus de llibre hi ha uns camps a omplir (molts d'ells són comuns). Hem definit quatre tipus diferents de documents: llibres, revistes, àudio i vídeo. L'empleat és l'únic que pot gestionar els documents.

Dins de la gestió d'usuaris, l'empleat i l'administrador són els únics que poden fer altes, baixes i modificacions. L'empleat trobarà un formulari on tindrà que emplenar totes les dades del soci (algunes de les quals són obligatòries). Una vegada complert el formulari, el soci ja estarà enregistrat a la base de dades i l'empleat ja li podrà prestar documents. En el cas de donar de baixa o modificar dades d'un soci, primer s'haurà de trobar el soci en qüestió i després fer les diferents accions possibles. La recerca es realitzarà mitjançant el NIF ja que és únic per a cada persona. Una vegada trobat l'usuari, ja es podria donar de baixa o modificar les seves dades. L'administrador pot gestionar en la seva totalitat els usuaris. En canvi, l'empleat podria sols gestionar als socis.

En el cas dels préstecs, l'empleat ha de poder donar en préstec qualsevol document de la biblioteca a un soci i que s'enregistren totes aquestes dades. El que es vol es que una vegada fem un préstec, es sàpiga en qualsevol moment qui ha donat el préstec, quin document a donat i a qui se li ha donat aquest préstec. Aquesta funcionalitat és clau per poder portar a bon terme la biblioteca. La única persona que pot donar préstecs és el propi empleat de la biblioteca.

L'empleat, a més dels préstecs, també pot fer renovacions i reserves de documents. Aquestes funcionalitats les comparteix amb el propi soci, ja que aquest també les pot

fer a través de la web i després d'autenticar-se. L'empleat podrà renovar un préstec d'un soci si fa una recerca del soci i veu els préstecs que té. D'aquesta manera podria fer la renovació ràpidament. Per les reserves tindria que fer una recerca del document a reserva i assignar-lo a un soci. Totes aquestes funcionalitats les podria fer el soci al autenticar-se ja que li sortirien aquells documents que té en préstec. Des d'allà podria renovar qualsevol document que tingués. Per poder fer una reserva tindria que fer el mateix que l'empleat ja que primer hauria de fer una recerca del document a reservar. En aquest moment fer la reserva.

Aquestes són les funcionalitats bàsiques que es poden portar a terme dins d'una biblioteca. L'aplicació està pensada per a que no solament faci ús una sola biblioteca sinó que ho faci tota una cadena de biblioteques. Totes aquestes biblioteques accedirien al mateix servidor on estaria instal·lada l'aplicació. La idea és que tots els usuaris pertanyen a tota la cadena de biblioteques i no a una sola biblioteca. Per exemple, un soci podria agafar un o varis documents de qualsevol biblioteca. Això si, els documents si que són de cada biblioteca. Cada document si que pertany a una biblioteca en qüestió i no es pot traspasar d'un lloc a un altre. Un soci que agafi un préstec en una biblioteca no pot entregar aquests documents en una altra.

Això implica que els usuaris es tindran que autenticar de la manera següent:

- Administradors: s'autentiquen com administradors. Es relacionen amb tota la cadena.
- Empleats: s'autentiquen en relació a una de les biblioteques. Això és important ja que després, al donar préstecs, ho donarà en relació a aquesta biblioteca. Igualment, podrà canviar la relació dins de l'aplicació
- Soci: simplement s'autenticarà. El sistema no necessita cap relació amb cap biblioteca.

3.3 *Requeriments funcionals*

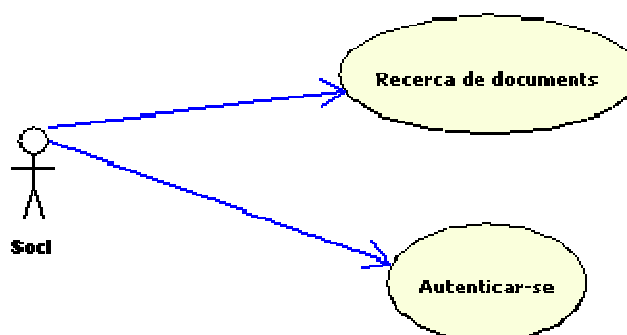
L'aplicació la tenim dividida en diferents subsistemes:

- **Subsistema principal:** compren la plana inicial de l'aplicació i les diferents accions que poden fer els usuaris.
- **Subsistema de gestió d'usuaris:** l'aplicació té que tenir un control de les altes, baixes i modificacions de dades dels usuaris. Aquest sistema és qui controla tot això.
- **Subsistema de gestió de documents:** s'encarrega de tractar les altes, baixes i modificacions de dades dels documents.
- **Subsistema de gestió de préstecs:** és l'encarregat de fer totes les gestions referents als préstecs, renovacions i reserves de documents.
- **Subsistema de configuració:** Permet fer diferents configuracions en l'aplicació.

3.4 Especificacions de les funcionalitats per subsistema

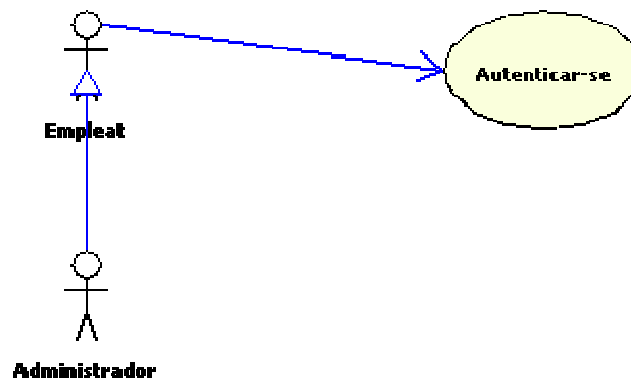
3.4.1 Subsistema principal

Compren la plana inicial de l'aplicació i les diferents accions que poden fer els usuaris al començament. El casos d'ús que posem a continuació fan referència als tres usuaris que poden entrar a l'aplicació. El soci, com ja sabem, entra a la plana principal i pot fer una recerca de documents o, pot autenticar-se per fer reserves o renovacions. En canvi, els empleats i l'administrador han d'entrar des d'una plana reservats a ells. Allà no podem més que autenticar-se. L'explicació dels casos d'ús és la següent:



Cas d'ús: Recerca de documents	
Resum	Permet fer recerques sobre els documents de la biblioteca.
Paper	Plana principal dels usuaris i, també, es pot accedir des del menú.
Actors	<u>Soci</u> (com a plana principal). Els <u>Empleats</u> i <u>Administradors</u> també poden accedir des del menú.
Cas d'ús relacionat	
Precondició	És la primera plana que veu el soci.
Postcondició	El soci pot fer la recerca de documents o, mitjançant el menú, pot autenticar-se
Descripció	El soci, quan entra a la plana principal de la web, el primer que veurà serà la pantalla de recerca de documents. Aquesta es la plana d'entrada a la web ja que és l'acció més important de l'aplicació.

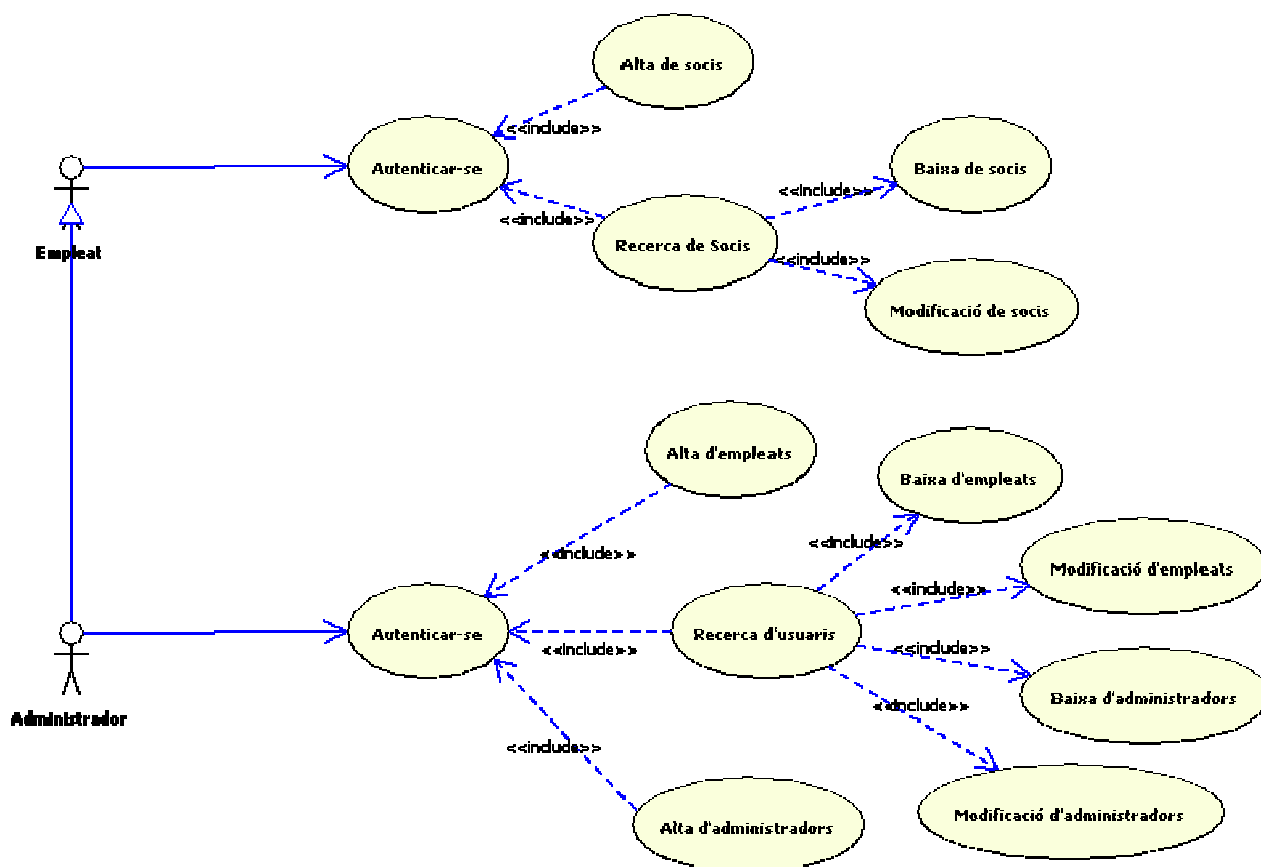
Cas d'ús: Autenticar-se	
Resum	Permet autenticar-se al soci per fer altres operacions (reserves i renovacions).
Paper	Autenticació del soci
Actors	<u>Soci</u>
Cas d'ús relacionat	
Precondició	Ha de ser un soci per accedir a aquesta autenticació.
Postcondició	S'identifica el soci. Apareixen més funcionalitats (reserves i renovacions).
Descripció	El soci pot autenticar-se. D'aquesta manera li apareixeran dues funcionalitats més que li donaran els privilegis corresponents per fer renovacions i reserves de documents.



Cas d'ús: Autenticar-se	
Resum	Permet autenticar-se als empleats i als administradors.
Paper	Autenticació d'empleats i d'administradors.
Actors	Empleat, Administrador
Cas d'ús relacionat	
Precondició	Ha de ser un empleat o un administrador per accedir a aquesta autenticació.
Postcondició	S'identifica l'empleat o l'administrador. Apareixen més funcionalitats depenen si és un o l'altre.
Descripció	Els empleats o administradors poden autenticar-se des d'una plana que no és la principal. D'aquesta manera i una vegada identificats, podran fer el treball que té cadascú. En el cas dels empleats accedirà a la gestió de préstecs, gestió d'usuaris i gestió de documents. L'administrador podrà gestionar usuaris i gestionar la configuració de l'aplicació.

3.4.2 Subsistema de gestió d'usuaris

Aquest subsistema està planificat per poder gestionar tots els usuaris de l'aplicació. Tenim tres tipus d'usuaris: socis, empleats i administradors. Cadascú pot fer diferents funcions. Els únics que podem gestionar usuaris són els empleats (que solament poden gestionar socis) i els administradors, que poden gestionar a qualsevol. El cas d'ús del subsistema de gestió d'usuaris és el següent:



Cas d'ús: Alta de socis	
Resum	Permet donar d'alta als socis
Paper	Donar d'alta als socis
Actors	<u>Empleat</u> , <u>Administrador</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'empleat o administrador ha d'estar autenticat
Postcondició	Es dona d'alta un soci
Descripció	Els empleats o administradors després d'autenticar-se podran donar d'alta un soci després d'haver inserit totes els seves dades.

Cas d'ús: Recerca de socis	
Resum	Permet trobar un soci
Paper	Trobar un soci
Actors	<u>Empleat, Administrador</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'empleat o administrador ha d'estar autenticat
Postcondició	Trobem al soci del qual hem fet la recerca
Descripció	Els empleats o administradors després d'autenticar-se podran fer una recerca d'un soci per consultar, donar de baixa o modificar.

Cas d'ús: Baixa de socis	
Resum	Dona de baixa a un soci
Paper	Donar de baixa a un soci
Actors	<u>Empleat, Administrador</u>
Cas d'ús relacionat	<u>Recerca de socis</u>
Precondició	L'empleat o administrador han d'haver trobat el soci en qüestió.
Postcondició	S'haurà donat de baixa a un soci
Descripció	Els empleats o administradors hauran fet una recerca d'un soci per poder donar-ho de baixa.

Cas d'ús: Modificació de socis	
Resum	Modificar dades del soci
Paper	Modificar dades del soci
Actors	<u>Empleat, Administrador</u>
Cas d'ús relacionat	<u>Recerca de socis</u>
Precondició	L'empleat o administrador han d'haver trobat el soci en qüestió.
Postcondició	S'hauran modificat algunes dades del soci
Descripció	Els empleats o administradors hauran fet una recerca d'un soci per poder modificar qualsevol dada del soci en qüestió.

Cas d'ús: Alta d'empleats	
Resum	Permet donar d'alta empleats
Paper	Donar d'alta a empleats
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'administrador ha d'estar autenticat
Postcondició	Es dona d'alta un empleat
Descripció	Els administradors després d'autenticar-se podran donar d'alta un empleat després d'haver inserit totes els seves dades.

Cas d'ús: Alta d'administradors	
Resum	Permet donar d'alta administradors
Paper	Donar d'alta a administradors
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	Administrador ha d'estar autenticat
Postcondició	Es dona d'alta un administrador
Descripció	Els administrador després d'autenticar-se podran donar d'alta un altre administrador després d'haver inserit totes els seves dades.

Cas d'ús: Recerca d'usuaris	
Resum	Permet trobar qualsevol usuari
Paper	Trobar qualsevol usuari
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'administrador ha d'estar autenticat
Postcondició	Troben l'usuari del qual hem fet la recerca
Descripció	Els administradors, després d'autenticar-se, podran fer una recerca d'un usuari per consultar, donar de baixa o modificar.

Cas d'ús: Baixa d'empleats	
Resum	Dona de baixa a un empleat
Paper	Donar de baixa a un empleat
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Recerca d'usuaris</u>
Precondició	L'administrador ha d'haver trobat l'empleat en qüestió.
Postcondició	S'haurà donat de baixa a un empleat
Descripció	Els administradors hauran fet una recerca d'un usuari per poder donar-ho de baixa.

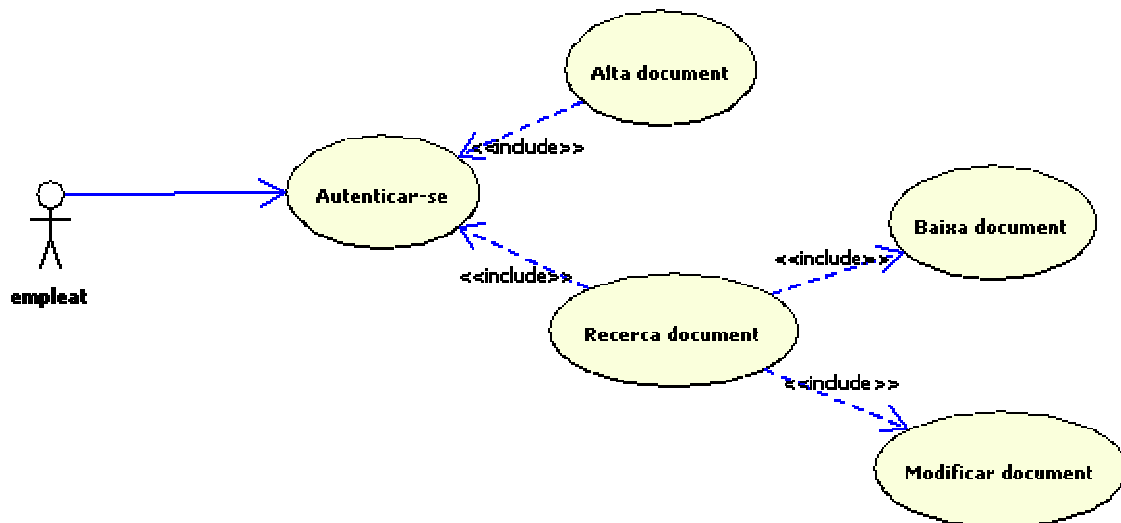
Cas d'ús: Baixa d'administradors	
Resum	Dona de baixa a un administrador
Paper	Donar de baixa a un administrador
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Recerca d'usuaris</u>
Precondició	L'administrador ha d'haver trobat l'administrador en qüestió.
Postcondició	S'haurà donat de baixa a un administrador
Descripció	Els administradors hauran fet una recerca d'un administrador per poder donar-ho de baixa.

Cas d'ús: Modificació d'empleats	
Resum	Modificar dades d'un empleat
Paper	Modificar dades d'un empleat
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Recerca d'usuaris</u>
Precondició	L'administrador ha d'haver trobat l'empleat en qüestió.
Postcondició	S'hauran modificat algunes dades de l'empleat
Descripció	Els administradors hauran fet una recerca d'un usuari per poder modificar qualsevol dada d'aquest empleat en qüestió.

Cas d'ús: Modificació d'administradors	
Resum	Modificar dades d'un administrador
Paper	Modificar dades d'un administrador
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Recerca d'usuaris</u>
Precondició	L'administrador ha d'haver trobat l'administrador en qüestió.
Postcondició	S'hauran modificat algunes dades de l'administrador
Descripció	Els administradors hauran fet una recerca d'un usuari per poder modificar qualsevol dada de l'administrador en qüestió.

3.4.3 Subsistema de gestió de documents

Aquest subsistema el que fa es controlar la gestió dels documents. Volem tenir enregistrats els documents per que puguin ser utilitzats per als préstecs. Ho farà únicament l'empleat. Ell serà l'únic que podrà donar d'alta, baixa o modificar documents. El cas d'ús és el següent:



Cas d'ús: Alta document	
Resum	Permet donar d'alta al sistema documents
Paper	Donar d'alta documents
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'empleat ha d'estar autenticat
Postcondició	Es dona d'alta un document
Descripció	L'empleat donarà d'alta un document en el sistema sempre i quan s'hagi autenticat anteriorment. Per poder donar d'alta tindrà que haver inserit les dades del document.

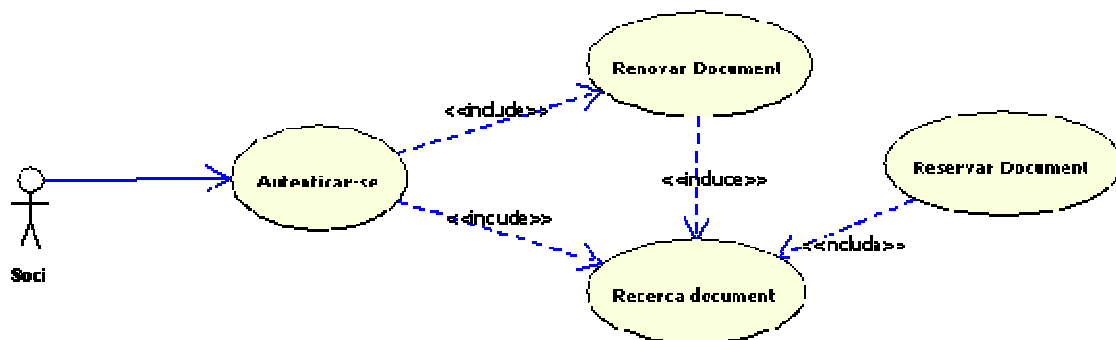
Cas d'ús: Baixa document	
Resum	Permet donar de baixa un document
Paper	Donar de baixa documents
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Recerca document</u>
Precondició	L'empleat ha d'haver trobat el document en qüestió
Postcondició	Es dona de baixa un document
Descripció	L'empleat donarà de baixa un document en el sistema sempre i quan s'hagi autenticat anteriorment. Per poder donar de baixa, abans haurà d'haver fer una recerca sobre el document.

Cas d'ús: Modificar document	
Resum	Permet donar modificar un document
Paper	Modificar un document
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Recerca document</u>
Precondició	L'empleat ha d'haver trobat el document en qüestió
Postcondició	Es modifiquen dades del document
Descripció	L'empleat modificarà dades d'un document que ha trobat anteriorment.

Cas d'ús: Recerca de documents	
Resum	Permet trobar qualsevol document
Paper	Trobar qualsevol document
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'empleat ha d'estar autenticat
Postcondició	Trobem el document del qual hem fet la recerca
Descripció	Els empleats, després d'autenticar-se, podran fer una recerca d'un document per consultar, donar de baixa o modificar.

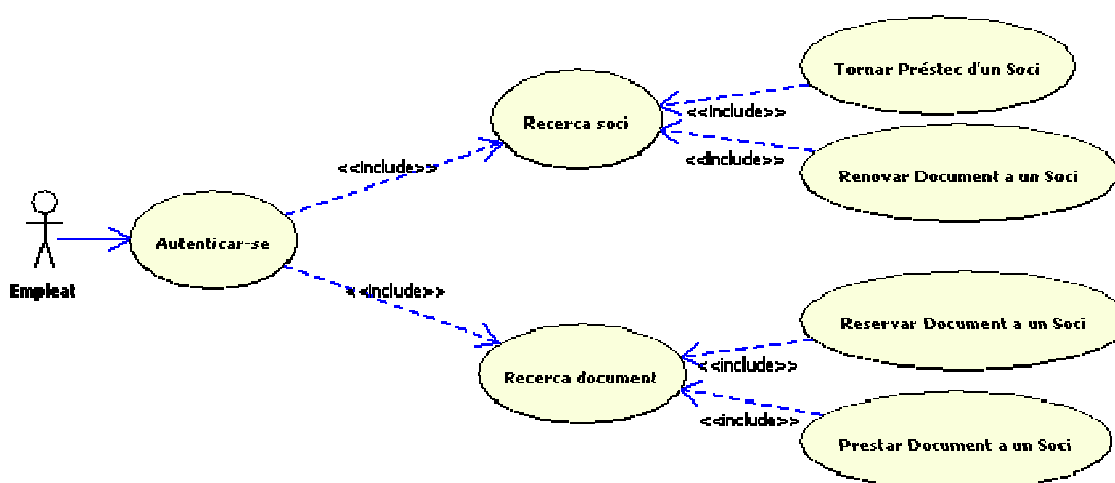
3.4.4 Subsistema de gestió dels préstecs

Aquest és el subsistema principal ja que és el que farà les funcions més importants. La gestió dels préstecs compren els préstecs, renovacions i reserves. Hem d'entendre que els préstecs els donaran els empleats. En canvi, els renovacions i reserves les podran fer els empleats als socis però els socis també podran renovar-se i reservar documents. Primer tindrem els cas d'ús dels socis:



Cas d'ús: Renovar document	
Resum	Permet fer renovacions dels documents que té en préstec
Paper	Renovar documents
Actors	<u>Socis</u>
Cas d'ús relacionat	<u>Autenticar-se</u> , <u>Recerca document</u>
Precondició	El soci ha d'estar autenticat i ha de tenir un document en préstec.
Postcondició	Renovem el document sempre i quan no sigui la tercera vegada que faci la renovació.
Descripció	Els socis poden renovar un document que ja havien agafat en préstec. Aquesta renovació es farà efectiva sempre i quan tingui aquest document en préstec. Per poder renovar s'haurà d'autenticar abans.

Cas d'ús: Reserva document	
Resum	Permet reservar qualsevol document
Paper	Reserva document
Actors	<u>Socis</u>
Cas d'ús relacionat	<u>Recerca document</u>
Precondició	L'empleat ha d'haver trobat el document en qüestió
Postcondició	Després de trobar el document, fem la reserva
Descripció	Els socis, després de fer una recerca sobre documents, trobaran un document en qüestió del qual podran fer una reserva.



Cas d'ús: Tornar Préstec d'un soci	
Resum	Permet retornar un préstec d'un soci
Paper	Tornar préstec d'un soci
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Recerca soci</u>
Precondició	L'empleat ha de trobar el soci en qüestió. El soci ha de tenir el document a tornar
Postcondició	Després de trobar el soci, fa la funció de retornar el document del llistat de documents en préstec
Descripció	L'empleat podrà, després de fer la recerca del soci, retornar els préstecs que aquest soci tingui en aquest moment. No fa falta una recerca de document ja que té que sortir un llistat automàticament del que té.

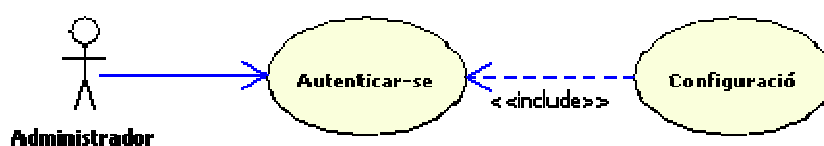
Cas d'ús: Renovar Document a un soci	
Resum	Permet renovar un document a un soci
Paper	Renovar préstec a un soci
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Recerca soci</u>
Precondició	L'empleat ha de trobar el soci en qüestió. El soci ha de tenir el document a renovar
Postcondició	Després de trobar el soci, renova un document del llistat de documents en préstec que té el soci.
Descripció	L'empleat podrà, després de fer la recerca del soci, renovar un document en préstec que tingui el soci. No fa falta una recerca de document ja que té que sortir un llistat automàticament del que té.

Cas d'ús: Reservar Document a un soci	
Resum	Permet reservar un document a un soci
Paper	Reservar un document a un soci
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Recerca document</u>
Precondició	L'empleat ha de trobar el document a reservar. Ha d'estar el document en préstec.
Postcondició	Després de trobar el document, es reserva a un soci en qüestió.
Descripció	L'empleat podrà, després de fer la recerca del document, fer la reserva d'aquest a un soci en qüestió.

Cas d'ús: Prestar Document a un soci	
Resum	Permet donar en préstec un document a un soci
Paper	Prestar document a un soci
Actors	<u>Empleat</u>
Cas d'ús relacionat	<u>Recerca document</u>
Precondició	L'empleat ha de trobar el document en qüestió. No ha d'estar actualment ni en préstec ni en reserva.
Postcondició	Després de trobar el document, donaria en préstec el aquest a un soci
Descripció	L'empleat podrà, després de fer la recerca del document, fer el préstec d'aquest a un soci. Això si, no ha d'estar el document actualment en préstec o en reserva per altre soci.

3.4.5 Subsistema de configuració

Aquest subsistema és únicament per a l'administrador. Gràcies a ell podrà tenir opcions de canviar certes característiques de l'aplicació.



Cas d'ús: Configuració	
Resum	Permet configurar l'aplicació
Paper	Configurar l'aplicació
Actors	<u>Administrador</u>
Cas d'ús relacionat	<u>Autenticar-se</u>
Precondició	L'administrador ha d'estar autenticat per a poder configurar l'aplicació.
Postcondició	Després d'autenticar-se, l'administrador podrà canviar certes característiques de l'aplicació.
Descripció	L'administrador podrà canviar certes característiques de l'aplicació gràcies tenir diferents opcions per configurar-la.

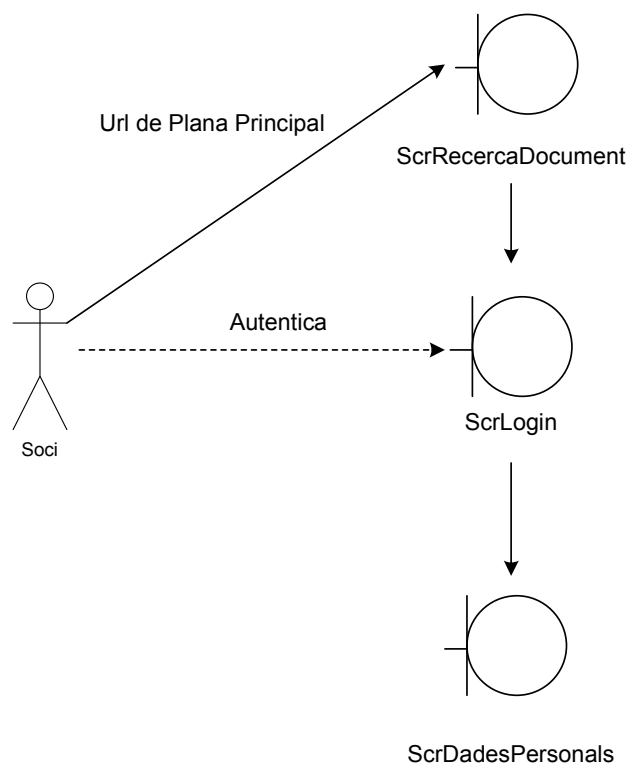
3.5 Interfície gràfica

La interfície gràfica d'aquesta aplicació és clara i està relacionada amb els navegadors web. L'usuari veurà planes webs on podrà interactuar amb els diferents events, els quals el aniran guiant per les diferents planes. Aquestes planes li aniran donant tota la informació que vagi demanant. Comentar que encara que els socis i els empleats/administradors no entren per la mateixa url, la interfície gràfica es comú i és igual.

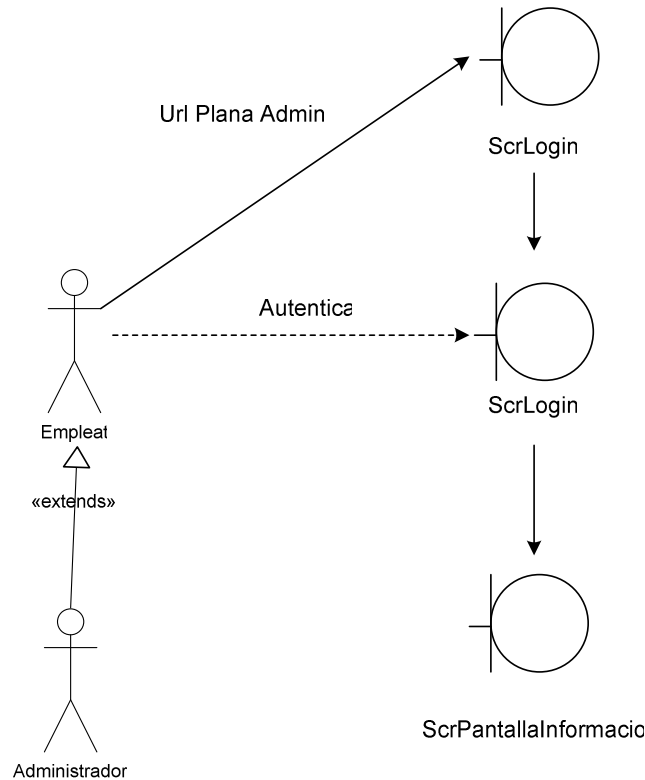
3.5.1 Flux de pantalles

3.5.1.1 Inici de l'aplicació

Depenent de l'usuari que entra a l'aplicació podem tenir un tipus d'entrada o un altre. Per diferenciar això, direm que un soci tindrà la url principal que li portarà a la plana de recerques de documents. A més, des del menú tindrà la possibilitat de autenticar-se. Una vegada autenticat, l'aplicació el portarà a la pantalla de les dades personals on podrà veure les seves dades i els documents que te en préstec en el moment d'autenticar-se.



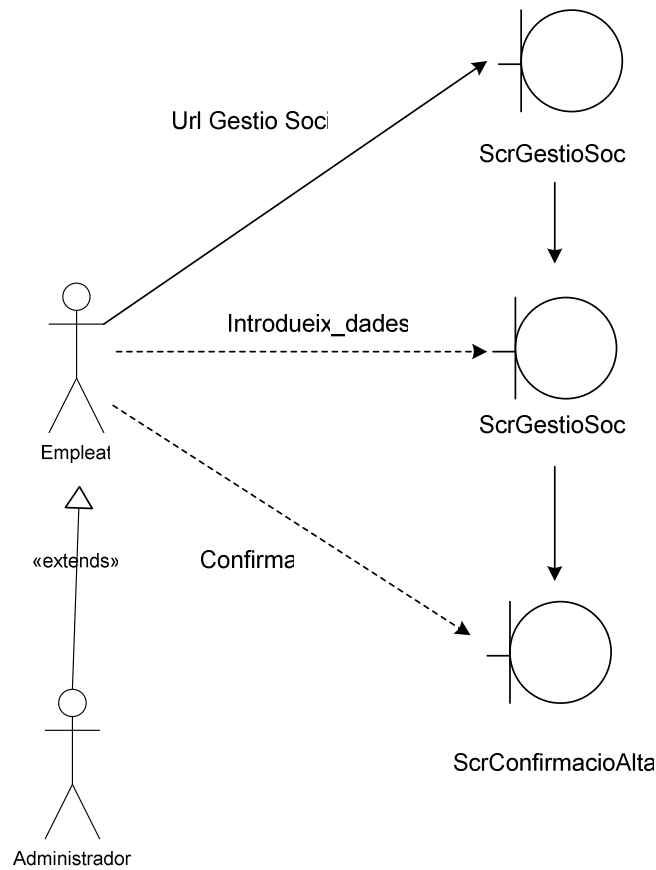
En canvi, els empleats i administradors, per poder entrar a l'aplicació per treballar tindran una url diferent (per seguretat) i simplement es podran autenticar. Una vegada autenticats, l'aplicació els portarà a una plana de informació/ajuda on s'indicarà el que poden fer amb el menú (recerca documents, gestió d'usuaris, gestió de préstecs, etc.).



3.5.1.2 Subsistema gestió d'usuaris

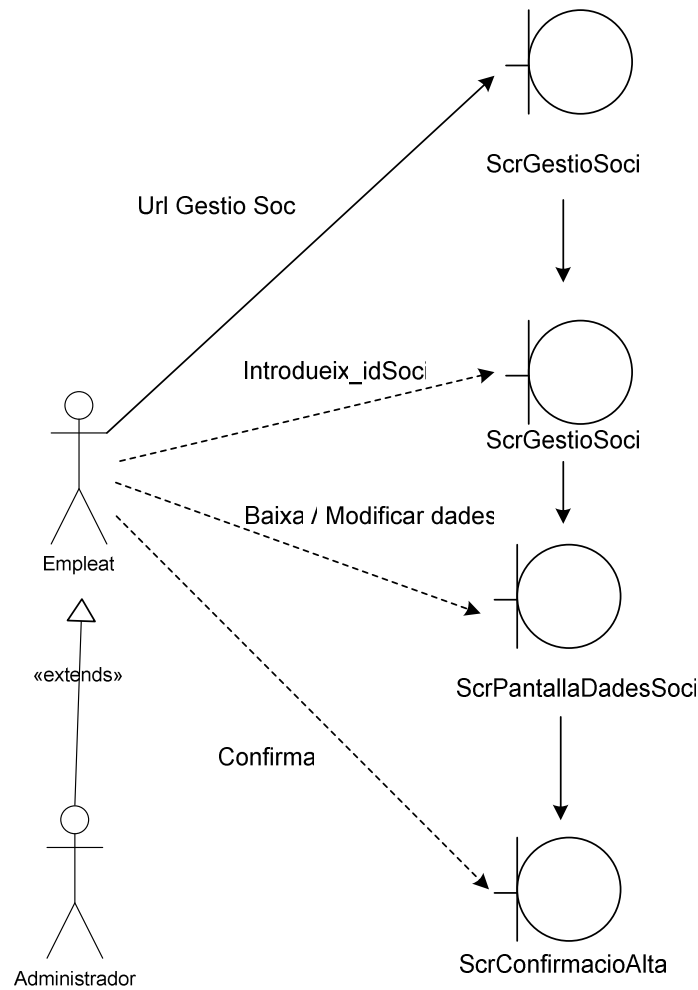
3.5.1.2.1 Alta de socis

L'alta de socis ho poden fer tant l'empleat com l'administrador. Per poder fer-la han de tenir la url de la pantalla de gestió de soci (pel menú es pot anar). Allà tindran dos formularis, un de recerca de soci i una altre per donar d'alta un soci. Per fer l'alta, s'introduiran les dades en aquest últim. Per confirmar arribarem a una pantalla amb les dades introduïdes on tindrem que confirmar.



3.5.1.2.2 Baixa i modificació de socis

La baixa i la modificació de socis segueixen les mateixes pantalles. Per tant, hem decidit d'agrupar-les en el mateix diagrama. En aquests casos, abans de poder fer una baixa o una modificació de dades, primer s'haurà de trobar el soci en qüestió per després fer les accions oportunes i, per últim, confirmar. El diagrama seria el següent:



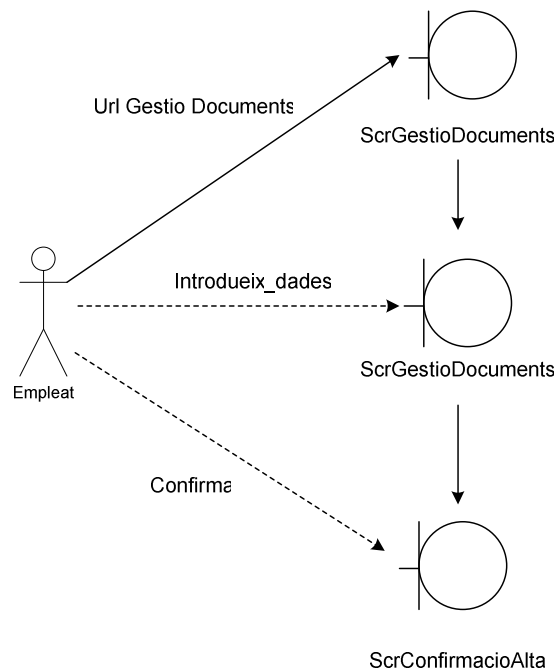
3.5.1.2.3 Alta, Baixa i Modificació d'empleats

Aquestes pantalles tenen una similitud a les pantalles tractades anteriorment. Les diferències que poden existir es que l'únic que pot fer aquestes accions és l'administrador. Les pantalles seran les mateixes però anomenades més generalment, ja que no seran socis sinó empleats. El cas és que quan l'administrador tingui que donar d'alta un usuari tindrà l'opció de donar-ho d'alta com a soci o com a empleat. El mateix passarà amb les baixes i modificacions de dades.

3.5.1.3 Subsistema de gestió de documents

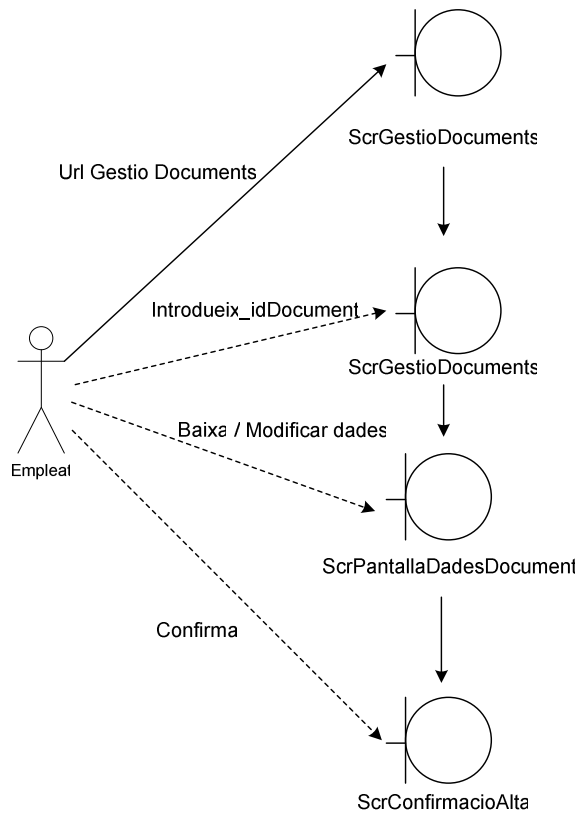
3.5.1.3.1 Alta de documents

Com ja s'ha dit anteriorment, l'única persona que pot gestionar documents és l'empleat. En el cas de l'alta, l'empleat haurà d'introduir les dades del document i confirmar que vol fer l'alta. El diagrama seria el següent:



3.5.1.3.2 Baixa i modificacions de documents

Per donar de baixa o modificar documents s'haurà de fer una recerca abans del document en qüestió. Es per això que en la pantalla de Gestió de documents hauran d'introduir el identificador del document. Això portarà a l'empleat a la pantalla de dades del document on podrà realitzar l'acció corresponent (podrà modificar dades o donar de baixa). I, per últim, confirmar la baixa o la modificació de dades.

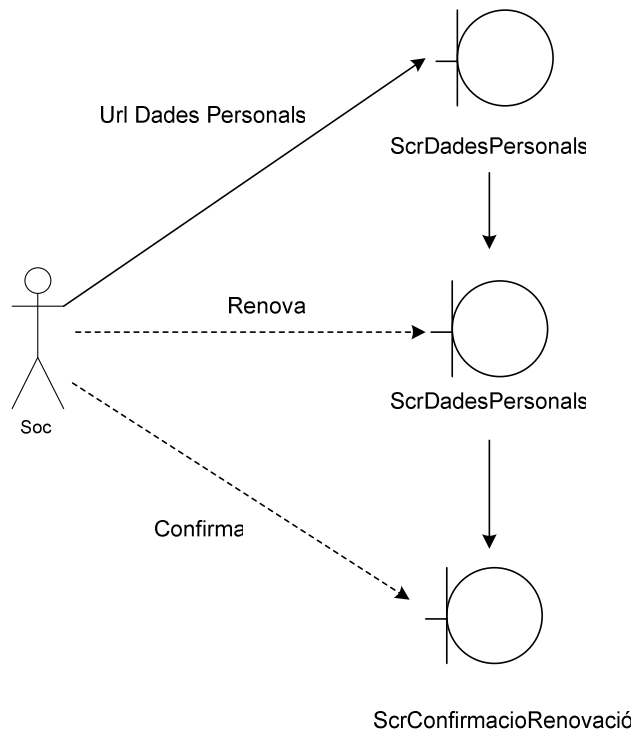


3.5.1.4 Subsistema de gestió de préstecs

Aquest subsistema es pot dividir en dos. Un amb les pantalles que veurà el soci, ja que pot fer reserves de documents i renovacions dels seus préstecs, i un altre per a l'empleat, que pot fer préstecs, reserves i renovacions als socis.

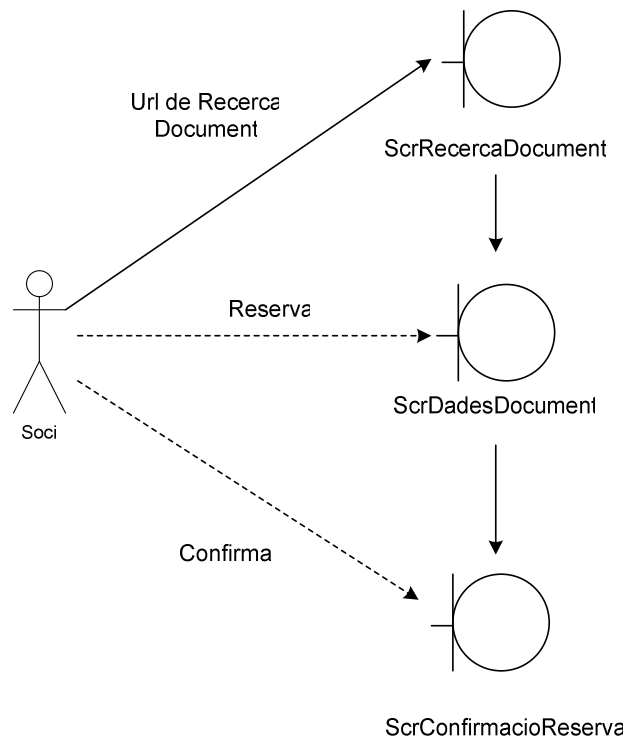
3.5.1.4.1 Renovacions de préstecs per part del soci

En aquest cas, el soci pot fer renovacions dels préstecs que té. Quan està a la pantalla on es veu les seves dades personals i els préstecs que té actualment, podrà renovar qualsevol d'aquests documents. El diagrama seria el següent:



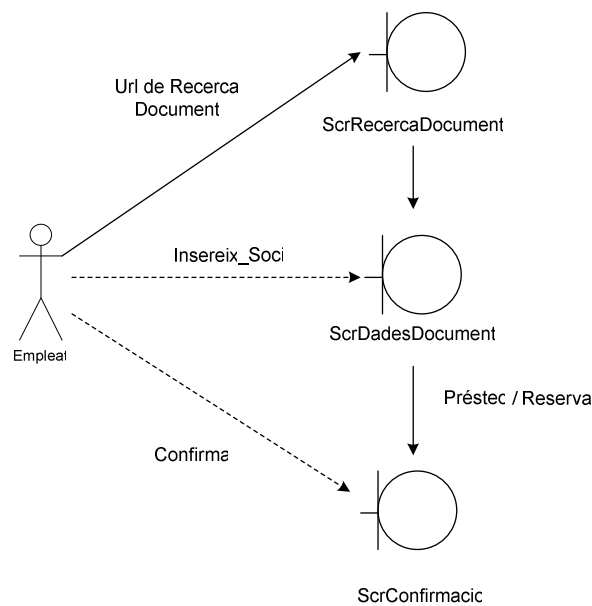
3.5.1.4.2 Reserves de documents per part del soci

Per fer les reserves de documents, el soci haurà, primer de tot, trobar el document el qüestió per a fer la reserva. Després ho confirmarà.



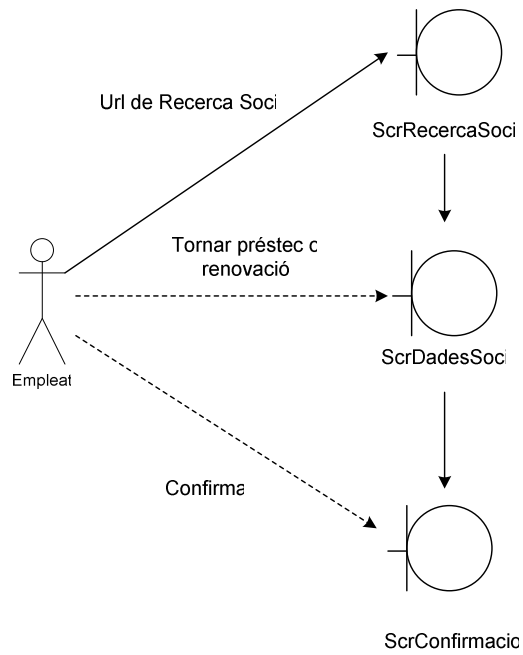
3.5.1.4.3 Préstecs i Reserves de documents

Ens trobem davant d'uns dels casos més importants, ja que és quan l'empleat donarà en préstec un document a un soci. En aquest cas, l'empleat haurà de fer la recerca del document inserint l'identificatiu del document. Una vegada té les dades principals del document tindrà un petit formulari on podrà inserir l'identificatiu del soci. D'aquesta manera podrà fer l'acció pertinent, ja sigui donar préstec o reservar. En tots dos casos, portarà a l'empleat a una pantalla de confirmació.



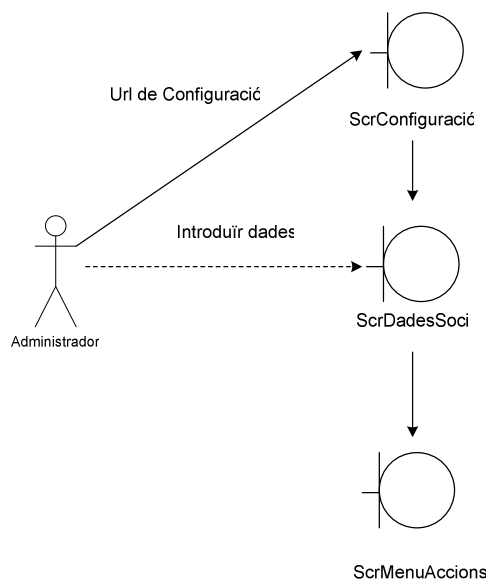
3.5.1.4.4 Tornar préstecs i renovacions de documents

Aquests dos casos són diferents a l'anterior en l'aspecte que del primer que hem de fer la recerca és del soci. Com ja hem dit, l'empleat fa primer la recerca del soci. En aquest cas sortirien les dades del soci més un llistat dels documents que té en préstec. És en aquesta pantalla on es podrà tornar el préstec o fer les renovacions pertinents.



3.5.1.5 Subsistema de configuració

El subsistema de configuració és únicament per a l'administrador. Tindrà diferents funcionalitats per poder configurar l'aplicació.



3.6 *Requeriments de maquinari i de programari*

Els requeriments bàsics de maquinari i de programari els definim a continuació. Hem de dividir aquests requeriments en dos parts (clients i servidors) ja que es basa en una arquitectura de programari distribuïda.

- Requeriments dels clients

Un client, per accedir a l'aplicació simplement necessitarà un navegador web. Amb les urls d'entrada podrà iniciar l'aplicació sense problemes. Aquest navegador serà el que enviarà peticions HTTP al servidor que conté l'aplicació i d'aquesta manera es podran fer les diferents accions que es permeten.

- Requeriments del servidor

El servidor ha de tenir instal·lats el programari adient per poder executar una aplicació del tipus de J2EE. Per l'aplicació de la biblioteca es necessita el següent programari:

- Servidor d'aplicacions: Qualsevol servidor d'aplicacions es podria utilitzar. Les proves s'han fet amb JBoss que és de lliure distribució i amb un ampla utilització en el mercat. Aquest utilitza el servidor web Apache Tomcat, un dels més importants i més utilitzats.
- Base de dades: S'ha elegit la base de dades Oracle 10g, una de les més potents del mercat. Les dades s'han de persistir i, per tant, necessitem una base de dades per poder fer-ho. A més suporta JDBC i l'Hibernate el suporta perfectament.

Com a requeriments de maquinari pel servidor necessitem una màquina potent amb molta memòria RAM per poder anar lliure i més ràpid. Personalment, pensem que la base de dades tindria que estar en altra màquina ja que Oracle consumeix molts recursos.

3.7 *Seguretat*

En l'aspecte de seguretat d'accés hem de fer referències a les mesures que s'han d'aplicar per controlar i garantir el correcte accés al sistema. L'accés a les accions administratives han d'estar totalment controlades ja que no tothom pot entrar i poder fer ús d'aquestes funcionalitats. L'aplicació ha de distingir entre el tipus d'usuari que entra i amb això donar unes funcionalitats o altres.

És molt important separar els accessos per als socis a la web dels d'administració (empleats i administradors). Es farà amb distintes urls per no tenir més seguretat de que una persona externa mai podrà tenir accés al sistema.

Per tant, parlem de dos entrades diferents al sistema, una pel socis (molt limitat d'accions i de funcionalitats) i l'altra entrada pels empleats i administradors on es tindrà que definir quin tipus d'usuari està entrant al sistema.

Per la seguretat de les dades, es tindria que fer uns backups d'aquestes per poder tenir segures totes les dades que tenim introduïdes. Hem de pensar que aquestes dades es la principal vàlua de l'aplicació. Es per això, que l'administrador del sistema haurà de definir un mecanisme de còpies de seguretat per poder tenir les dades segures.

També s'aconsella la instal·lació d'un RAID, així com de disposar d'assistència tècnica professional que s'encarregui del manteniment i reparació del maquinari per possibles problemes.

3.8 Futures funcionalitats en properes versions

- Incorporar un nivell de Seguretat SSL (Secure Sockets Layer) al sistema d'administració.
- Incorporar un log d'accions en tota l'aplicació
- Tenir més control sobre els usuaris (nombre d'intent d'entrades, temps màxim de les connexions, etc)
- Limitar l'accés des de l'exterior a la part administrativa.
- Afegir més opcions en la part de configuració de l'aplicació per a l'administrador.

4 Disseny de l'arquitectura de l'aplicació

4.1 Introducció

El disseny de l'aplicació s'ha basat en J2EE intentant donar una solució a l'anàlisi que hem fet a l'apartat anterior. Farem una descripció teòrica del que s'ha emprat per poder construir una aplicació que tingui totes les característiques ja esmentades anteriorment. Parlarem de l'arquitectura J2EE, de l'estructura MVC que implementem gràcies al framework Struts, a la utilització de Tiles per gestionar les vistes JSP i a la persistència mitjançant Hibernate.

A més, es presenta un disseny UML del model de classes i un disseny Entitat-Relació per l'estructura de dades que acabarà tancant el disseny de l'arquitectura de l'aplicació.

4.2 Arquitectura J2EE

J2EE és l'edició empresarial del paquet Java creada i distribuïda per Sun Microsystems. És un conjunt d'especificacions i funcionalitats orientades al desenvolupament d'aplicacions empresarials. Algunes de les funcionalitats més importants són:

- Accés a base de dades
- Utilitzat per BEA, IBM, Oracle, Sun i Apache Tomcat entre d'altres.
- Utilització de directoris distribuïts (JNDI)
- Accés a mètodes remots (RMI/CORBA)
- Funcions de correu electrònic (JavaMail)
- Aplicacions Web (JSP i Servlet)
- Utilització de Beans, etc.

La plataforma J2EE és fruit de la col·laboració de SUN amb els líders del sector de software empresarial (IBM, Apple, Bea Systems, Oracle, Inprise, Hewlett-Packard, Novell, etc.) per definir una plataforma robusta i flexible orientada a cobrir les necessitats empresarials en e-business i business-to-business.

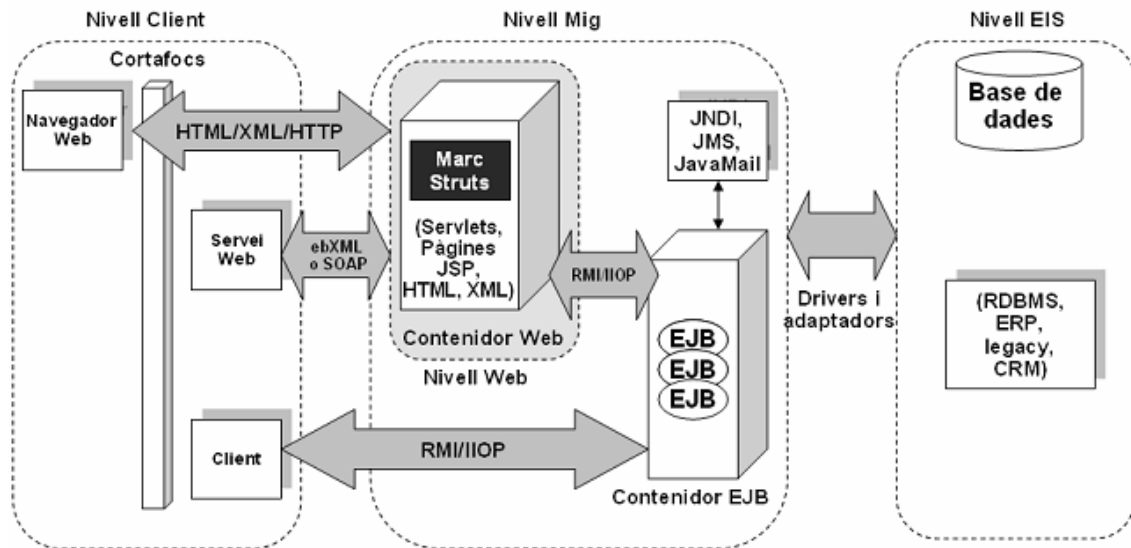
Una de les característiques més importants que té la plataforma J2EE és que redueix el cost i la complexitat de desenvolupaments i dona com a resultats serveis que poden ser creats ràpidament i fàcilment millorats responen a les pressions competitives d'una empresa.

J2EE està basat en una arquitectura del costat del servidor (Served-based). Aquest tipus d'arquitectura concentra la majoria de tots els processos en l'aplicació en el servidor o en una part d'aquest. Aquest tipus d'arquitectura té dos avantatges en comparació amb d'altres tipus. Aquests són:

- Múltiples clients: Una arquitectura basada en el servidor requereix una clara separació entre la capa client (interfície) i la capa servidor, en la qual es realitzen els processos de l'aplicació. Això permet que una simple aplicació suporti simultàniament clients amb diferents tipus d'interfícies, incloent interfícies poderoses per equips corporatius, interfícies multimèdia interactives per usuaris amb connexions d'alta velocitat, interfícies eficients basades en text per usuaris amb connexions de baixa velocitat, etc.
- Operacions robustes: Una arquitectura basada en el servidor suporta escalabilitat, confiabilitat, disponibilitat i recuperabilitat. Aplicacions basades en el servidor poden ser dividides i distribuïdes en múltiples processadors. Components de l'aplicació poden ser replicats per donar suport a caigudes instantàniament.

Per poder donar tota aquesta funcionalitat, la plataforma J2EE dona un conjunt d'APIs de java i serveis necessaris per al suport d'aplicacions per empreses. La plataforma completa pot ser implementada en un sol sistema, o la plataforma de serveis pot ser distribuïda a través de diferents sistemes. Però totes les APIs especificades tenen que ser incloses en alguna part del sistema complet.

A continuació podem veure una il·lustració de l'arquitectura:



No totes les aplicacions han de tenir els nivells il·lustrats. Per moltes aplicacions més petites, el nivell mig pot constar d'un contenidor Web que interactuï directament amb una base de dades en el nivell de Sistema de Informació Empresarial (EIS).

- **Nivell client:** El nivell client proporciona un medi per a que els usuaris interactuïn amb l'aplicació. Aquesta interacció pot ser mitjançant un navegador Web, o mitjançant de programació a través d'una interfície de serveis Web. Amb independència del tipus de client, la interacció pot incloure enviar una petició i rebre algun tipus de resposta del nivell mig.
- **Nivell mig:** Aquest nivell el podem conèixer com a "nivell d'aplicació" o "servidor". Això és així perquè normalment existeix un servidor d'aplicacions dins d'aquest nivell. El que existeix també molt sovint és un nivell web dins d'aquest més algun tipus de component de servidor d'aplicacions (per exemple, EJB). Aquests dos nivells es poden combinar i molts servidors d'aplicacions donen la funcionalitat de nivell Web. El nivell Web permet que el nivell client es comuniqui i interactuï con la lògica d'aplicació que resideix en altres nivells. En aplicacions Web més tradicionals, no és estrany que part o tota la lògica d'aplicació resideixi en aquest nivell. En aplicacions empresarials més grans, el nivell Web actua com un traductor i mapeja peticions HTTP en invocacions de servei en el nivell mig. També és responsable de gestionar el flux de pantalla basant-se en l'estat de l'aplicació i de l'usuari. El nivell Web es comunica amb una base de dades o, en el cas d'una aplicació empresarial, amb un servidor d'aplicació.
- **Nivell EIS:** El nivell de Sistema de Informació Empresarial o EIS conté dades i serveis que s'utilitzen en tota l'empresa. Proporciona accés als recursos de l'empresa tals com base de dades, mainframes, aplicacions CRM i sistemes d'aplicacions de recursos. El nivell mig es comunica amb els components en el nivell EIS utilitzant protocols específics de recursos.

4.3 Paradigma "MVC" - Struts

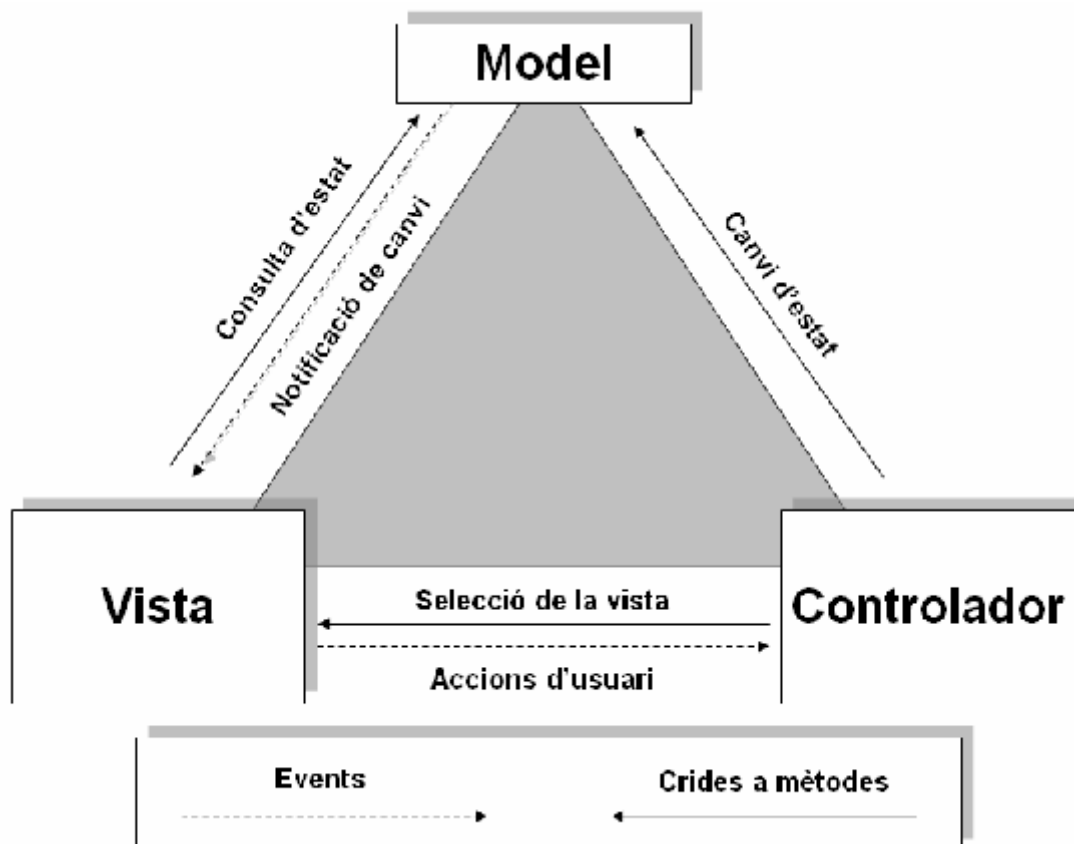
L'arquitectura de la nostra aplicació es basa en l'estructura MVC ("Model-View-Controller"). És un paradigma utilitzat en diversos desenvolupament de software. Gràcies a aquest "Framework" s'aconsegueix una divisió de les diferents parts que conformen una aplicació amb la una de les seves principals raons: manteniment del codi font.

Conforme s'incrementen les necessitats de qualsevol aplicació, la modificació del codi existent es fa imminent i si no existeix una clara divisió d'ús, el codi no sols es torna indesxifrable sinó que en ocasions es torna impredecible degut a la barreja de funcionalitats que poden sorgir.

A través de MVC es realitza la següent divisió:

- **Model:** Concentra les funcionalitats relacionades amb el Model de dades. Això és l'accés i la manipulació de depòsits informatius com Base de dades i Arxius.
- **View:** Es basa en l'aspecte visual/gràfic que serà empleat per l'aplicació en qüestió.
- **Controller:** Parlaríem d'un mediador entre el medi gràfic ("View") i el model ("Model") coordinant les accions entre ambdós.

En la figura següent, veiem l'arquitectura MVC en la seva forma més general. Hi ha un Model, múltiples Controladors que manipulen aquest Model i hi han diferents Vistes de les dades del Model, canvien quan canvia l'estat d'aquest Model.



Pensem les avantatges que pot presentar aquesta arquitectura:

- Hi ha una clara separació entre els components d'un programa; això ens ajuda a implementar-los per separat.
- La connexió entre el Model i les seves Vistes es dinàmica; això s'aconsegueix en temps d'execució, no en temps de compilació.

Al incorporar el model d'arquitectura MVC en un disseny, les peces d'un programa es poden construir per separat i després fer la unió en temps d'execució. Si un dels Components, posteriorment, s'observa que funciona malament, es pot reemplaçar sense que les altres peces es vegin afectades.

La nostra aplicació utilitza arquitectura JSP/Servlet/J2EE i hi han diverses implementacions per utilitzar un "Framework MVC". Entre totes hem decidit agafar Struts.

Struts (org.apache.struts) és una estructura d'aplicació Web de codi obert, basada en el patró de disseny MVC i dissenyat utilitzant el servlet i les API JSP per construir complexes aplicacions Web. Ens deixa dissecionar la lògica de negoci, la lògica de control i el codi de presentació d'una aplicació, i ens ajuda a millorar la seva posterior utilització i el seu manteniment. L'estructura Struts forma part del projecte Jakarta, gestionat per la Fundació Apache Software (<http://jakarta.apache.org/struts>).

Entre les característiques de Struts es poden mencionar les següents:

- Configuració de control centralitzat
- Interrelacions entre Accions i pàgina o altres accions s'especifiquen per taules XML en lloc de codificar-les en els programes o pàgines.
- Components d'aplicació, que són el mecanisme per compartir informació bidireccionalment entre l'usuari de l'aplicació i les accions de model.
- Llibreries d'entitats per facilitar la gran part de les operacions que generalment realitzen les pàgines JSP.
- Struts conté eines per validació de camps de plantilles sobre diferents esquemes que van des de validacions locals en la pàgina (JavaScript) fins les validacions de fons fetes a nivell d'accions.

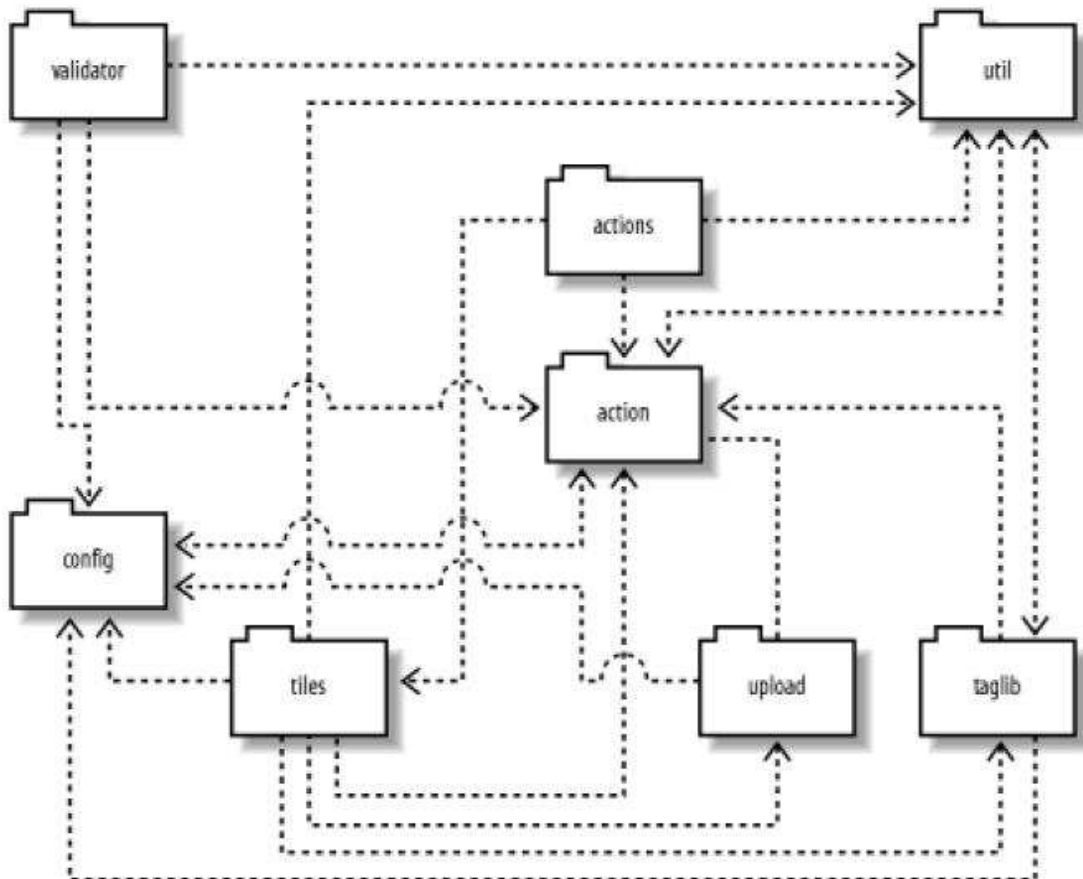
En resum, Struts permet que el que construeix l'aplicació es pugui concentrar en el disseny d'aplicacions complexes com una sèrie simple de components del Model i de la vista intercomunicats per un control centralitzat. Dissenyant d'aquesta manera es té que obtenir una aplicació més consistent i més fàcil de mantenir.

El marc de treball de Struts es un conjunt d'aproximadament 300 classes Java, que es divideixen en 8 paquets principals. I això està creixent ja s'està desenvolupant noves versions.

Descripció dels paquets:

- **Action:** Conté les classes del controlador, ActionForm, ActionMessages i alguns altres components de marc obligatoris.
- **Actions:** Conté les classes Action, tals com DispatchAction, que els desenvolupadors poden utilitzar o ampliar.
- **Config:** Conté les classes de configuració, que són representacions en memòria de l'arxiu de configuracions de Struts.
- **Taglib:** Conté les classes que controlen les etiquetes per les llibreries d'etiquetes Struts.

- **Tiles:** Conté les classes utilitzades pel marc de treball Tiles.
- **Upload:** Aquest paquet dona les classes necessàries per enviar o descarregar arxius des del sistema d'arxius local utilitzant un navegador.
- **Util:** Conté classes d'utilitats genèriques utilitzades per tot el marc.
- **Validator:** Són les classes d'ampliació específiques de Struts utilitzades per Struts quan es desplega el validator. Les classes i interfícies de Validator es troben en el paquet commons, separats de Struts.



4.4 El Component Controlador de Struts

Els components del controlador de Struts són responsables de detectar l'entrada de dades de l'usuari, actualitzant possiblement el model de domini i seleccionant la següent vista per al client. El controlador ajuda a separar la presentació del model en sí. Aquesta separació li ofereix molt més llibertat per desenvolupar una varietat de presentacions basades en un sol model de domini. Utilitzar un controlador proporciona un punt de control centralitzat on es processen inicialment totes les peticions dels clients. Centralitzar el control d'aquesta forma realitza dos requisits del disseny MVC.

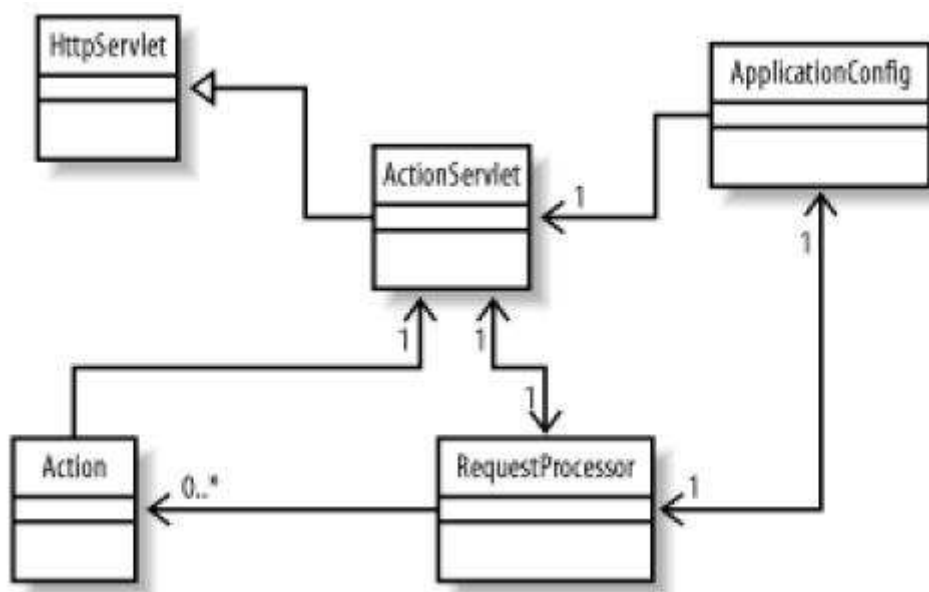
- El controlador actua com el mediador/traductor entre l'entrada de dades del client i el model, proporcionant una funcionalitat comú tal com seguretat, connexió i altres serveis importants en nom de cada petició del client.
- La vista es desacobla tant de la lògica de negoci com d'altres components de vista. La vista que es torna al client depèn per complet del controlador. Això fa que les aplicacions siguin molt més flexibles.

El patró de disseny Front Controller J2EE utilitza un sol controlador per canalitzar Totes les peticions dels clients a través d'un punt central. Entre les moltes avantatges que aquest patró dona a la funcionalitat de l'aplicació, es que els serveis com, per exemple, el de seguretat, internacionalització i connexió estan concentrats dins del controlador. Això permet l'aplicació coherent d'aquestes funcions en totes les peticions. Quan el comportament d'aquests serveis necessita modificació, els canvis que afecten potencialment a tota l'aplicació es tenen que realitzar solament en una petita àrea del programa.

Les responsabilitats més importants del controlador són:

- Interceptar peticions del client.
- Mapejar cada petició a una operació específica de negoci.
- Recopilar els resultats de l'operació de negoci i posar-los disponibles per al client.
- Determinar la vista a mostrar al client basada en l'estat actual i resultat de l'operació de negoci.

Els components de Struts que es reparteixen les responsabilitats del controlador es veuen clarament dins del següent esquema:



4.5 Components de Vista de Struts

En sentit general, una vista representa una mostra del model de domini en una interfície. Poden existir moltes vistes diferents del mateix model. El model de domini conté les entitats de negoci (estat de l'aplicació). Parlant en sentit metafòric, una vista és una finestra que pot ser diferent depenen de la ventana per la que mira el client.

Generalment, les vistes en el marc Struts es creen utilitzant pàgines JSP. Hi ha altres tipus d'enfocaments i marcs que es troben disponibles per fer les mateixes funcions, però JSP és la tecnologia de presentació més àmpliament utilitzada en la comunitat Struts. Components addicionals que es poden utilitzar juntament amb pàgines JSP per mostrar les vistes porten:

- Documents HTML
- Llibreries d'etiquetes personalitzades JSP
- JavaScript i fulls d'estil
- Arxius multimèdia
- Paquets de recurs de missatge
- Classes ActionForm

4.5.1 JavaScript i fulls d'estil

Struts proporciona funcionalitat dins de les llibreries d'etiquetes per facilitar la utilització de JavaScript. També es pot utilitzar fulls d'estil en una pàgina JSP per poder tenir molt més control sobre l'apariència de la Web. D'aquesta manera podrem controlar característiques com tamany de caràcter, color, font i altres característiques d'aquest tipus es poden canviar en una ubicació central i aconseguir un efecte ràpid en tota la web.

En l'aplicació que nos pertoca hem fet ús de JavaScript i dels fulls d'estil intentant donar-li dinamisme i un aspecte visual molt més treballat.

4.5.2 Ús de Tiles

Quan s'organitzen i s'assemblen contingut i disseny de pàgines JSP surt la típica problemàtica d'una gran quantitat de codi redundat que impossibilita després actualitzacions senzilles. Sempre s'ha de pensar que el disseny d'una web pot sofrir molts canvis durant el seu cicle de vida. El que ens hem trobat és que la tecnologia JSP desafortunadament no proporciona cap suport directe per dissenys o gestors de disseny. Poder utilitzar dissenys i gestors de dissenys podria haver ajudat a separar els diferents apartats de la pàgina del disseny de manera que es podria alterar amb mínim impacte en la resta de l'aplicació.

Per això es va crear l'enfocament basat en plantilles. Una plantilla és una pàgina JSP que utilitza una llibreria d'etiquetes personalitzada JSP per descriure el disseny d'una pàgina. Podem dir que la plantilla actua com una definició de l'aspecte que tindrà les pàgines d'una aplicació, sense especificar el contingut. El contingut s'insereix en la pàgina de plantilla en el moment d'execució. Una o més pàgines poden utilitzar la mateixa plantilla.

La finalitat d'una plantilla es tenir un aspecte consistent dins d'una aplicació sense tenir que codificar manualment aquest aspecte en cada una de les pàgines. Té sentit que la majoria de les pàgines utilitzin la mateixa plantilla. El que no té sentit i res comú és tenir un aspecte diferent per algunes pàgines dins d'una aplicació i, per tant, necessitar més d'una plantilla.

L'entorn de treball Tiles proporciona un mecanisme de plantilles que permet separar les responsabilitats de disseny de les de contingut. Té la possibilitat d'establir un disseny i insereix dinàmicament els continguts de les seves pàgines en aquest disseny en el moment d'execució. És important pensar que aquestes funcions són molt potents i en el futur de l'aplicació es pot arribar a personalitzar basant-se en mecanismes com la internacionalització, preferències d'usuari o els típics canvis d'aspecte que ocorren en tota aplicació Web en algun moment futur.

Les característiques de Tiles són:

- Possibilitats d'ús de plantilles
- Construcció i carga de pàgina dinàmica
- Definicions de pantalla
- Suport per reutilització de pàgina i disseny
- Suport de internacionalització
- Suport de múltiples canals

Amb Tiles, el contingut de les aplicacions Web segueix gestionat per pàgines JSP i JavaBeans. Això sí, el disseny s'especifica dins d'una pàgina JSP apart o d'un arxiu XML. Per l'aplicació de la biblioteca hem utilitzat un arxiu XML per configurar Tiles. La raó és seguir el patró de tota l'aplicació de tenir tots els arxius de configuració en format XML. Simplement, aquest arxiu XML ha de seguir la sintaxis de l'arxiu tiles-config.dtd. Per tant, es molt més senzill de consultar o modificar qualsevol canvi en el disseny visual.

Un tile és un àrea o regió dins d'una pàgina Web. Una pàgina pot constar de una sola regió o estar dividida en vàries regions:



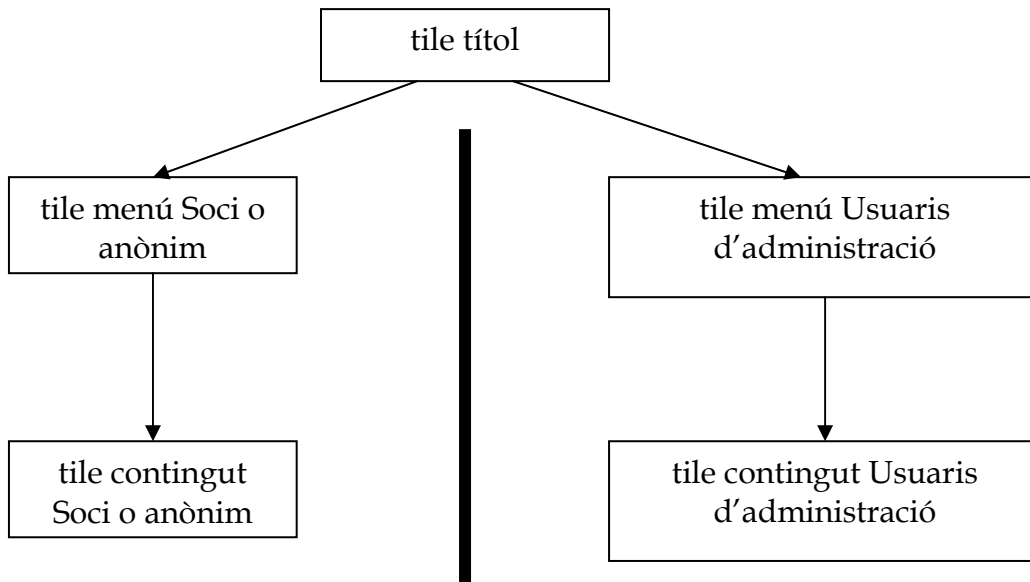
En aquest cas, la il·lustració ens mostra la divisió feta en l'aplicació de la biblioteca. Està dividida entres parts:

- **tile títol:** consta d'una imatge que correspon al títol de l'aplicació.
- **tile menú:** consta d'un menú horitzontal que dona accés a les diferents opcions que pot fer el usuari.
- **tile contingut:** consta d'una imatge que correspon al títol de l'aplicació.

L'aspecte més important de un tile és que és reutilitzable. Això és cert per dissenys al igual que per continguts. A diferència de la majoria de pàgines JSP, els components tile se reutilitzen dins d'una aplicació i possiblement a través de diferents aplicacions.

En aquest projecte s'ha fet un disseny amb Tiles que separa la part administrativa de la part dels socis. D'aquesta manera, per exemple, es pot tenir una barra de menú d'opcions totalment diferent depenent del usuari que estigui examinant la web.

El següent esquema il·lustra el que diem:



Com veiem a la il·lustració anterior, totes les pàgines de la web comparteixen una regió. Aquesta és la de la capçalera on tenim un títol. A partir d'aquesta divisió, totes les regions són diferents si estan o no en administració.

4.5.3 Validació de formularis

Quasi tota aplicació Web té que acceptar entrada de dades dels usuaris. Alguns exemples d'entrades de dades dels usuaris són nombres d'usuari i contrasenyes, informació de targetes de crèdit, informació de cobro i adreça d'enviament. L'aplicació de servidor pot recuperar tots aquests valors, realitzar una validació de dades i després passar les dades a altre component en l'aplicació on es portin a terme el procés real d'autenticació. Si les dades d'entrada no passen les regles de validació d'entrada de dades, l'aplicació tindria que regressar a la ubicació anterior, tornar a mostrar alguns o tots els valors incorporats i mostrar un missatge d'error.

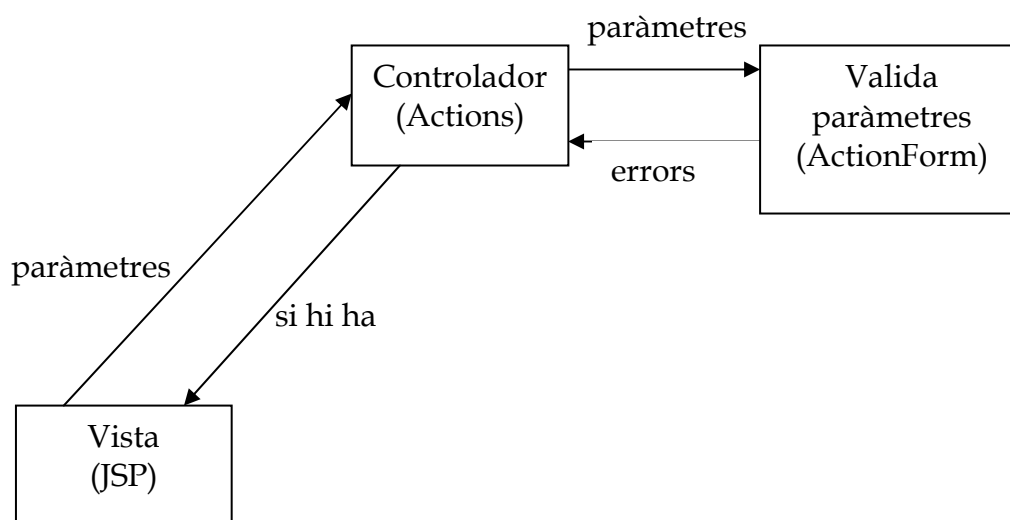
El marc de treball Struts proporciona aquesta funcionalitat i gestiona aquestes tasques en nombre de l'aplicació. Es basa en la classe ActionForm com el component clau per gestionar aquestes tasques. Per l'aplicació biblioteca la hem utilitzat ja que els usuaris incorporen a vegades dades no vàlides. Per això necessiten disposar d'un mitjà per enregistrar les dades d'entrada temporalment de manera que es puguin tornar a

mostrar quan ocorre un error. Les funcionalitats d'un ActionForm:

- Actua com un buffer que conté l'estat de les dades que ha facilitat l'usuari
- Actua com un firewall per l'aplicació en el sentit de que ajuda a mantenir l'entrada de dades vàlides o sospitoses fora del nivell de negoci fins que estiguin analitzades per les regles de validació.
- Quan el nivell de negoci retorna les dades, es pot completar una determinada ActionForm i utilitzar per una pàgina JSP per mostrar els camps d'entrada de dades per un formulari HTML. Això permet una major consistència en els formularis HTML, ja que sempre extrauen dades de l'ActionForm, no de diferents JavaBeans.

En l'aplicació biblioteca fem ús de les ActionForm per validar tots els formularis que utilitzem de tal manera que ens aprofitem de les seves funcionalitats que no són més que avantatges. D'aquesta manera podem retornar errors ràpidament a l'usuari per mostrar-li que ha introduït malament.

És el controlador de Struts el que fa ús dels ActionForms per validar els formularis. En la il·lustració següent es mostra l'ús dels ActionForms:



Per poder utilitzar la validació mitjançant els ActionForm, hem d'utilitzar un conjunt d'etiquetes especials en les vistes. L'entorn de treball Struts s'aprofita de la característica de llibreria d'etiquetes de JSP per incloure varies categories diferents d'etiquetes que ajuden a fer la capa de presentació més manejable i reutilitzable. Utilitzant les llibreries d'etiquetes personalitzades de Struts, els desenvolupadors poden interactuar amb la resta del marc sense incloure codi Java en les pàgines JSP.

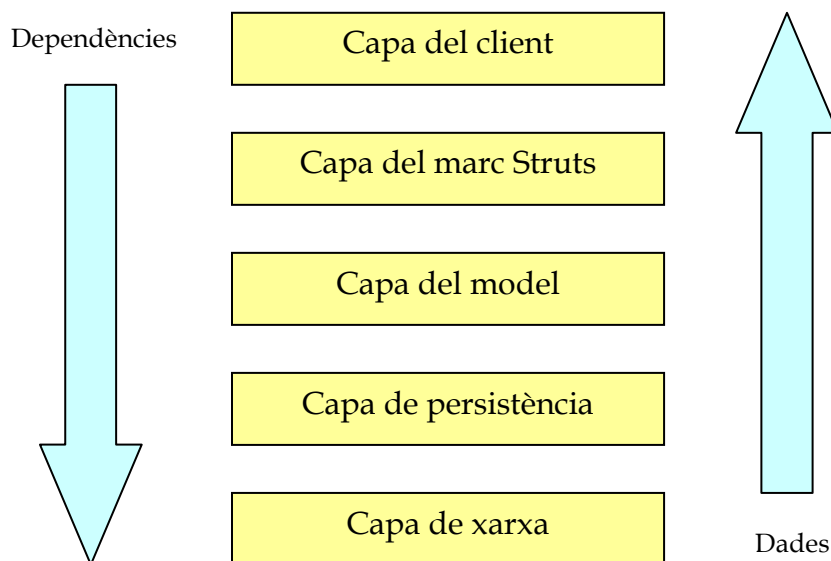
En aquest cas, l'aplicació utilitza les etiquetes HTML de Struts. Aquestes contenen etiquetes utilitzades per crear formularis d'entrada de dades HTML. Per exemple, en

lloc d'utilitzar un camp d'entrada de text HTML normal, es pot utilitzar l'etiqueta de text d'aquesta llibreria. S'han utilitzat en aquesta aplicació perquè estan dissenyades per funcionar amb els altres components del marc Struts (com ja hem dit, ens interessa que cooperin amb els ActionForms).

4.6 Components de Model de Struts

El model inclou les entitats de negoci i les regles que governen l'accés i la modificació de les dades. És de vital importància que aquest es mantingui en una única ubicació per mantenir la integritat vàlida de dades, reduir redundància i augmentar la reutilització.

El model ha de ser independent del tipus del client que s'utilitza per accedir als objectes de negoci i les seves regles associades. De fet, els components dins del model no tindrien saber què tipus de client o marc els utilitza. La idea és que quan s'utilitza una arquitectura per capes, la capa superior pot tenir dependències en capes inferiors, però les capes inferiors no tindrien que dependre de les capes per damunt d'elles.



4.6.1 Disseny de la persistència

En general, persistència significa que les dades que s'incorporen en l'aplicació, tant per un usuari com per altres mitjans, existiran més allà de la vida de l'aplicació. Encara que l'aplicació pot existir o l'ordinador es pot apagar, la informació sobreviurà. Això és molt important dins d'una organització ja que té

la necessitat de que les dades persistís.

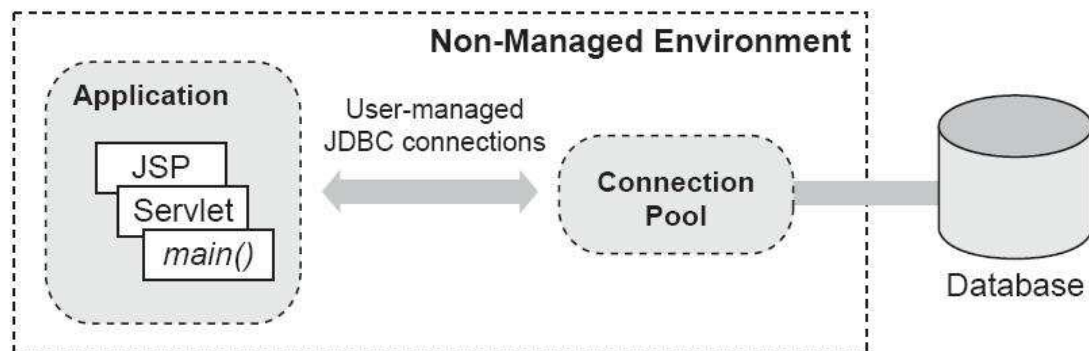
Quan arribem al moment de connectar o mapejar els objectes de negoci a la base de dades, existeix una varietat d'enfocaments entre els que podem triar. Les opcions depenen de diferents factors que poden canviar d'una aplicació a altra i d'una situació a altre. Alguns d'aquests enfocaments són:

- Utilitzar crides JDBC
- Utilitzar un enfocament ORM propi
- Utilitzar un marc ORM propietari
- Utilitzar un marc ORM no propietari
- Utilitzar una base de dades d'objecte

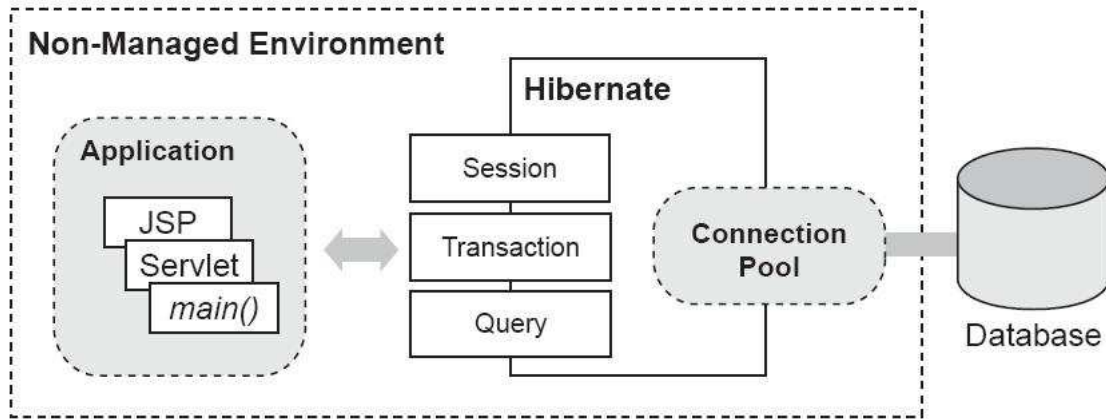
Per l'aplicació de la biblioteca es va decidir per un enfocament sobre productes ORM (propietari o no propietari). El que es volia, independentment de que sigui comercial o de codi lliure, és que la implementació del marc de mapeig no s'esmunyi en l'aplicació.

El ORM que vam decidir agafar va ser Hibernate. Deixa desenvolupar les classes persistents que segueixen el llenguatge usual de Java - incloent l'associació, l'herència, el polimorfisme, la composició, i el marc de col·leccions de Java. Hibernate permet també que les consultes es puguin fer en una pròpia extensió portable del SQL (HQL), així com en el SQL natiu, o con els objectes basats en Java.

En la següent il·lustració veiem com es treballa amb connexions JDBC:



Hibernate controla les connexions i es col·loca entre l'aplicació i la connexió de la següent manera:



El mapejat dels objectes de negoci a les taules de la base de dades tenen lloc dins d'arxius XML. Els arxius són analitzats pel marc de mapeig en temps d'execució i s'utilitzen per executar SQL en la base de dades. La configuració de la base de dades permet que el driver JDBC es connecti a la base de dades es fa en l'arxiu hibernate.cfg.xml. Es tindrà que configurar els paràmetres corresponents en aquest arxiu per l'entorn específic i tots el mapejos per l'aplicació.

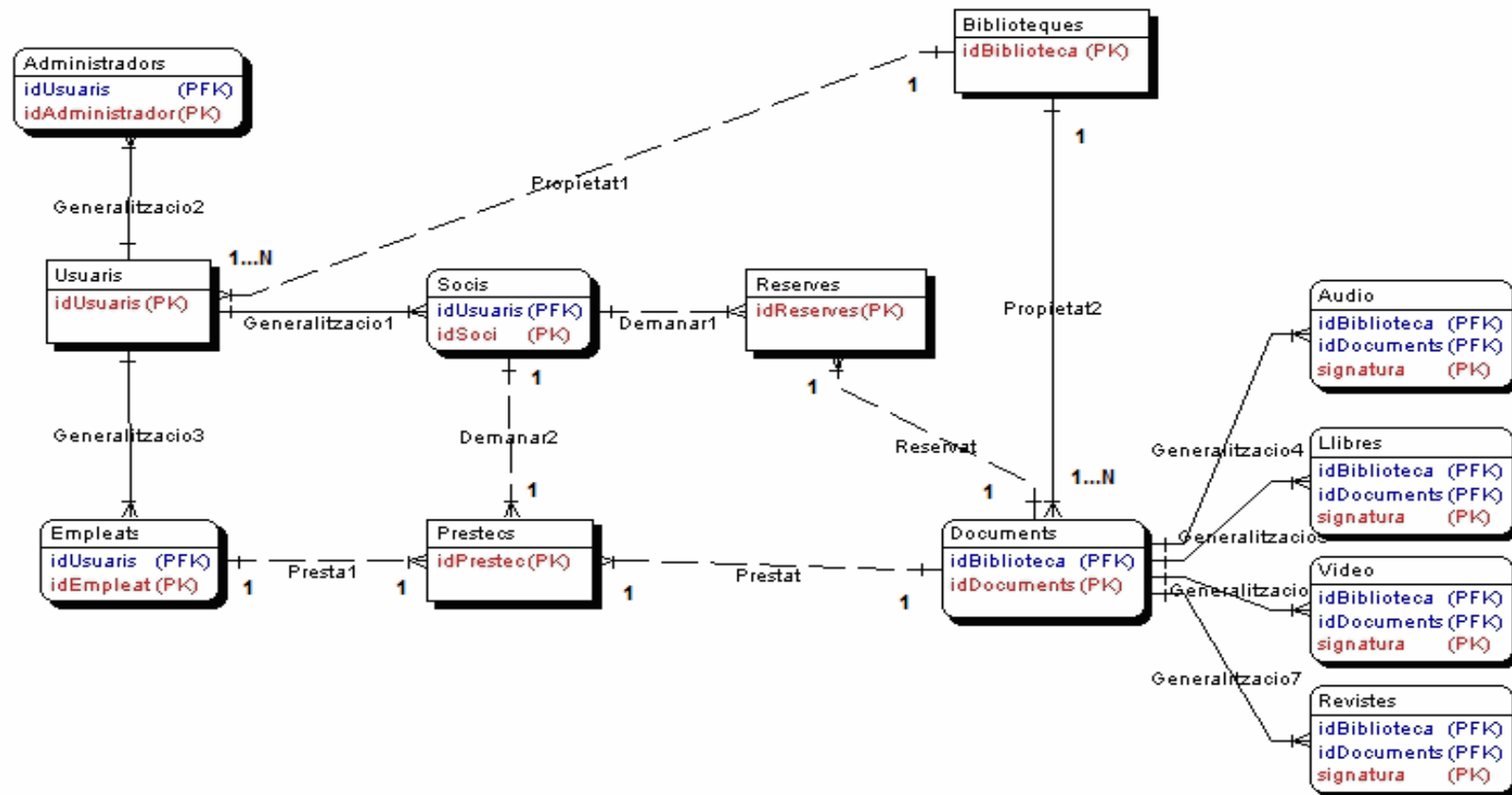
Això és tot el que s'ha tingut que fer a l'aplicació de la biblioteca per poder donar-li persistència. Anem al disseny propi de la base de dades.

4.6.1.1 Disseny de la base de dades

Arrel de l'anàlisi fet en capítols anteriors i el que hem anat definint en el disseny podem ja especificar el disseny conceptual de la base de dades, també conegut com a diagrama ER (Entity-Relationship).

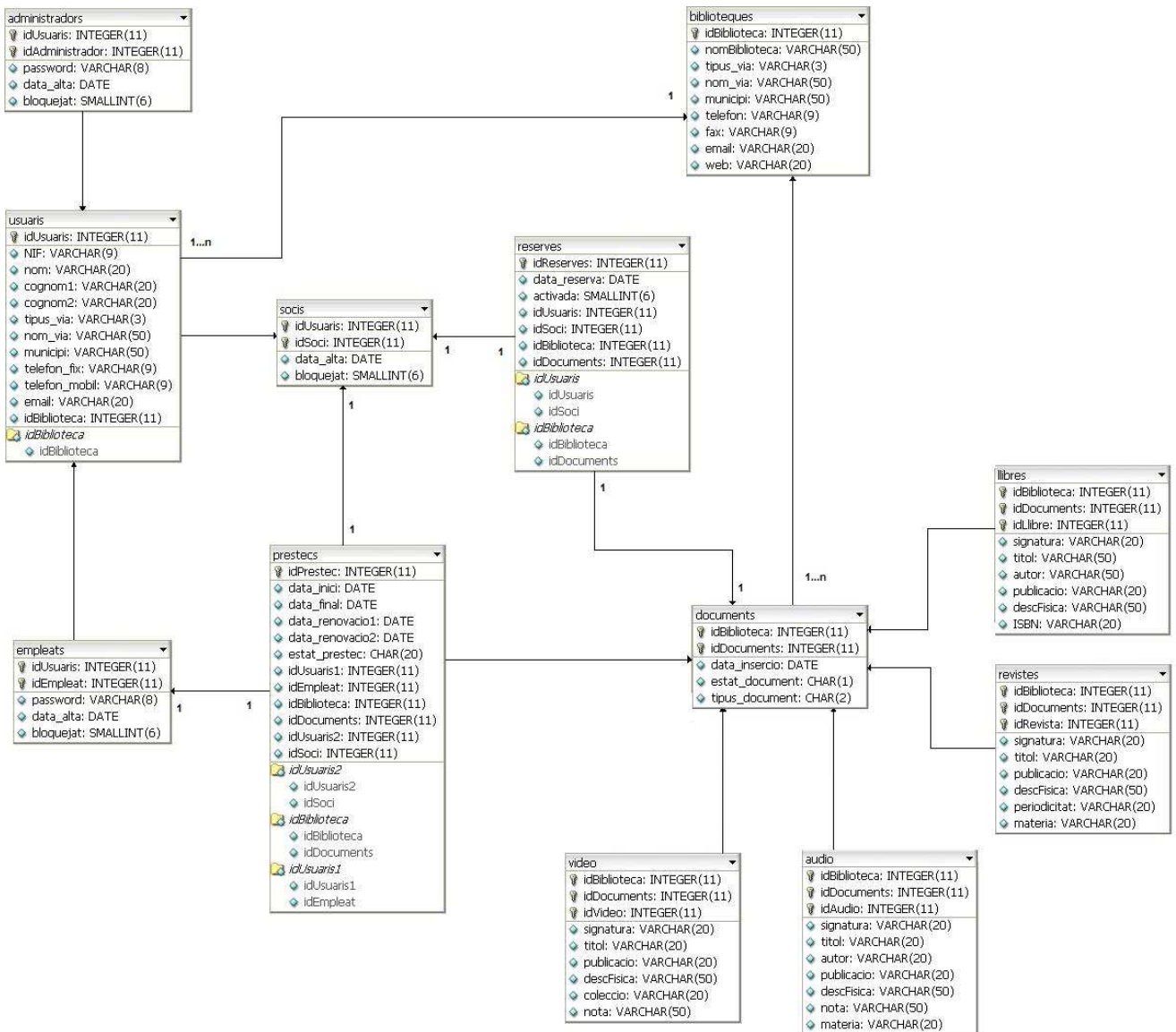
4.6.1.1.1 Diagrama ER

[2.1]



D'aquest model conceptual podem deduir també el disseny lògic per simples transformacions a partir de les entitats i relacions aquí especificades.

4.6.1.1.2 Disseny lògic de la base de dades



La descripció de les taules de la base de dades es compon de la següent informació:

- **Columna:** el nom de la columna de la taula
- **Tipus:** el tipus de dada segons Oracle

- **Clau:** PK *-primary key-* indica que aquesta columna forma part de la clau primària. FK *-foreign key-* indica que aquesta columna forma part d'una clau forana a una columna o columnes d'una altra taula.
- **Req.:** indica si la columna és requerida, és a dir que no accepta valors nuls.
- **Descripció:** conté una breu descripció de la columna.

4.6.1.1.3 Taula Biblioteques

Aquesta taula conté les dades de les diferents biblioteques que pot portar l'aplicació. Són bàsicament són l'adreça i dades de contacte.

Taula Biblioteques				
Columna	Tipus	Clau	Req.	Descripció
idBiblioteca	Integer	PK	Si	Identificador de la biblioteca
nomBiblioteca	Varchar2(50)		Si	Nom de la biblioteca
tipus_via	Varchar2(3)		Si	Carrer, Avinguda, Plaça, etc
nom_via	Varchar2(50)		Si	Nom de la via
municipi	Varchar2(50)		Si	Nom del municipi
telefon	Varchar2(9)		Si	Telefon de contacte
Fax	Varchar2(9)		Si	Fax de la biblioteca
email	Varchar2(20)		No	Email de contacte
web	Varchar2(20)		No	Adreça web de la biblioteca

4.6.1.1.4 Taula Usuaris

La taula que tenim a continuació mostra les dades comunes entre els diferents usuaris de l'aplicació, ja siguin administradors, empleats o socis.

Taula Usuaris				
Columna	Tipus	Clau	Req.	Descripció
idUsuari	Integer	PK	Si	Identificador d'usuari
NIF	Varchar2(9)		Si	Document d'identitat
Nom	Varchar2(20)		Si	Nom de l'usuari
cognom1	Varchar2(20)		Si	Primer cognom de l'usuari
cognom2	Varchar2(20)		Si	Segon cognom de l'usuari
Tipus_via	Varchar2(3)		No	Carrer, Avinguda, Plaça, etc
Nom_via	Varchar2(50)		No	Nom de la via
municipi	Varchar2(50)		No	Nom del municipi
telefon_fix	Varchar2(9)		No	Telefon fix de contacte
telefon_mobil	Varchar2(9)		No	Telefon mobil de contacte
email	Varchar2(9)		No	Email de contacte
idBiblioteca	Integer		Si	Biblioteca on es fa l'alta d'usuari

4.6.1.1.5 Taula Administradors

La següent taula conté la resta de dades de l'administrador.

Taula Administradors				
Columna	Tipus	Clau	Req.	Descripció
idUsuaris	Integer	PFK	Si	Identificador d'usuari
idAdministrador	Integer	PK	Si	Identificador d'administrador
password	Varchar2(8)		Si	Contrasenya
data_alta	Date		Si	Data de l'alta
bloquejat	Smallint		Si	Estat bloquejat

4.6.1.1.6 Taula Empleats

La següent taula conté la resta de dades d'un empleat

Taula Empleats				
Columna	Tipus	Clau	Req.	Descripció
idUsuaris	Integer	PFK	Si	Identificador d'usuari
idEmpleat	Integer	PK	Si	Identificador de l'empleat
password	Varchar2(8)		Si	Contrasenya
data_alta	Date		Si	Data de l'alta
bloquejat	Smallint		Si	Estat bloquejat

4.6.1.1.7 Taula Socis

La següent taula conté la resta de dades d'un soci.

Taula Socis				
Columna	Tipus	Clau	Req.	Descripció
idUsuaris	Integer	PFK	Si	Identificador d'usuari
idSoci	Integer	PK	Si	Identificador del soci
data_alta	Date		Si	Data de l'alta
bloquejat	Smallint		Si	Estat bloquejat

4.6.1.1.8 Taula Documents

Aquesta taula conté les dades comunes d'un document. Un document pertany a una biblioteca. Per això definim l'identificador de la biblioteca com a PFK

Taula Documents				
Columna	Tipus	Clau	Req.	Descripció
idBiblioteca	Integer	PFK	Si	Identificador de biblioteca
idDocuments	Integer	PK	Si	Identificador del document
data_insercio	Date		Si	Data d'inserció del document
estat_document	Char(1)		Si	Estat del document
tipus_document	Char(2)		Si	Tipus de document

4.6.1.1.9 Taula Video

Taula dels documents del tipus Video.

Taula Video				
Columna	Tipus	Clau	Req.	Descripció
idBiblioteca	Integer	PFK	Si	Identificador de biblioteca
idDocuments	Integer	PFK	Si	Identificador del document
idVideo	Integer	PK	Si	Identificador de Video
Signatura	Varchar2(20)		Si	Signatura del document
Titol	Varchar2(20)		Si	Titol de Video
Publicacio	Varchar2(20)		No	Nom de la publicació
descFisica	Varchar2(50)		No	Descripció física del Video
Coleccio	Varchar2(20)		No	Nom de la col·lecció
Nota	Varchar2(50)		No	Informació addicional

4.6.1.1.10 *Taula Audio*

Taula dels documents del tipus Audio.

Taula Audio				
Columna	Tipus	Clau	Req.	Descripció
idBiblioteca	Integer	PFK	Si	Identificador de biblioteca
idDocuments	Integer	PFK	Si	Identificador del document
idAudio	Integer	PK	Si	Identificador d'Audio
Signatura	Varchar2(20)		Si	Signatura del document
Titol	Varchar2(20)		Si	Titol de Audio
Autor	Varchar2(20)		Si	Nom de l'autor
Publicacio	Varchar2(20)		No	Nom de la publicacio
descFisica	Varchar2(50)		No	Descripció física del Video
Coleccio	Varchar2(20)		No	Nom de la col·lecció
Nota	Varchar2(50)		No	Informació addicional
Materia	Varchar2(20)		No	Temàtica de l'Audio

4.6.1.1.11 *Taula Revistes*

Taula dels documents del tipus Revista.

Taula Revistes				
Columna	Tipus	Clau	Req.	Descripció
idBiblioteca	Integer	PFK	Si	Identificador de biblioteca
idDocuments	Integer	PFK	Si	Identificador del document
idRevista	Integer	PK	Si	Identificador de Revista
Signatura	Varchar2(20)		Si	Signatura del document
Titol	Varchar2(20)		Si	Titol de Revista
Publicacio	Varchar2(20)		No	Nom de la publicacio
descFisica	Varchar2(50)		No	Descripció física de la Revista
Periodicitat	Varchar2(20)		No	Nom de la col·lecció
Materia	Varchar2(20)		No	Temàtica de la revista

4.6.1.1.12 *Taula Llibres*

Taula dels documents del tipus Llibre.

Taula Llibres				
Columna	Tipus	Clau	Req.	Descripció
idBiblioteca	Integer	PFK	Si	Identificador de biblioteca
idDocuments	Integer	PFK	Si	Identificador del document
idLlibre	Integer	PK	Si	Identificador del Llibre
Signatura	Varchar2(20)		Si	Signatura del document
Titol	Varchar2(20)		Si	Titol del Llibre
Autor	Varchar2(50)		Si	Autor del Llibre
Publicacio	Varchar2(20)		No	Nom de la publicacio
descFisica	Varchar2(50)		No	Descripció física del Llibre
ISBN	Varchar2(20)		Si	ISBN del Llibre

4.6.1.1.13 *Taula Reserves*

Taula on es van afegint les dades de les reserves de documents que es van fent.

Taula Reserves				
Columna	Tipus	Clau	Req.	Descripció
idReserves	Integer	PK	Si	Identificador de la Reserva
data_reserva	Date		Si	Data de quan es fa la reserva
Activada	Smallint		Si	Activat és encara està reservat
idUsuaris	Integer		Si	Identificador de l'usuari
idSoci	Integer		Si	Identificador del soci
idBiblioteca	Integer		Si	Identificador de la biblioteca
idDocument	Integer		Si	Identificador del document

4.6.1.1.14 Taula Préstecs

Aquesta taula va desant els préstecs que es van realitzant i en serveix per a controlar les diferents dades de com va el préstec.

Taula Préstecs				
Columna	Tipus	Clau	Req.	Descripció
idPrestec	Integer	PK	Si	Identificador del Préstec
data_inici	Date		Si	Data de quan es fa el Préstec
data_final	Date		Si	Data de retorn del Préstec
data_renovacio1	Date		No	Data de quan es fa la 1 ^a renovació
data_renovacio2	Date		No	Data de quan es fa la 2 ^a renovació
estat_prestec	Char(2)		Si	Estat del Préstec
idUsuaris1	Integer		Si	Identificador de l'usuari (Empleat)
idEmpleat	Integer		Si	Identificador de l'empleat
idBiblioteca	Integer		Si	Identificador de la biblioteca
idDocuments	Integer		Si	Identificador del document
idUsuaris2	Integer		Si	Identificador de l'usuari (Soci)
idSoci	Integer		Si	Identificador del soci

4.6.1.2 Disseny de classes

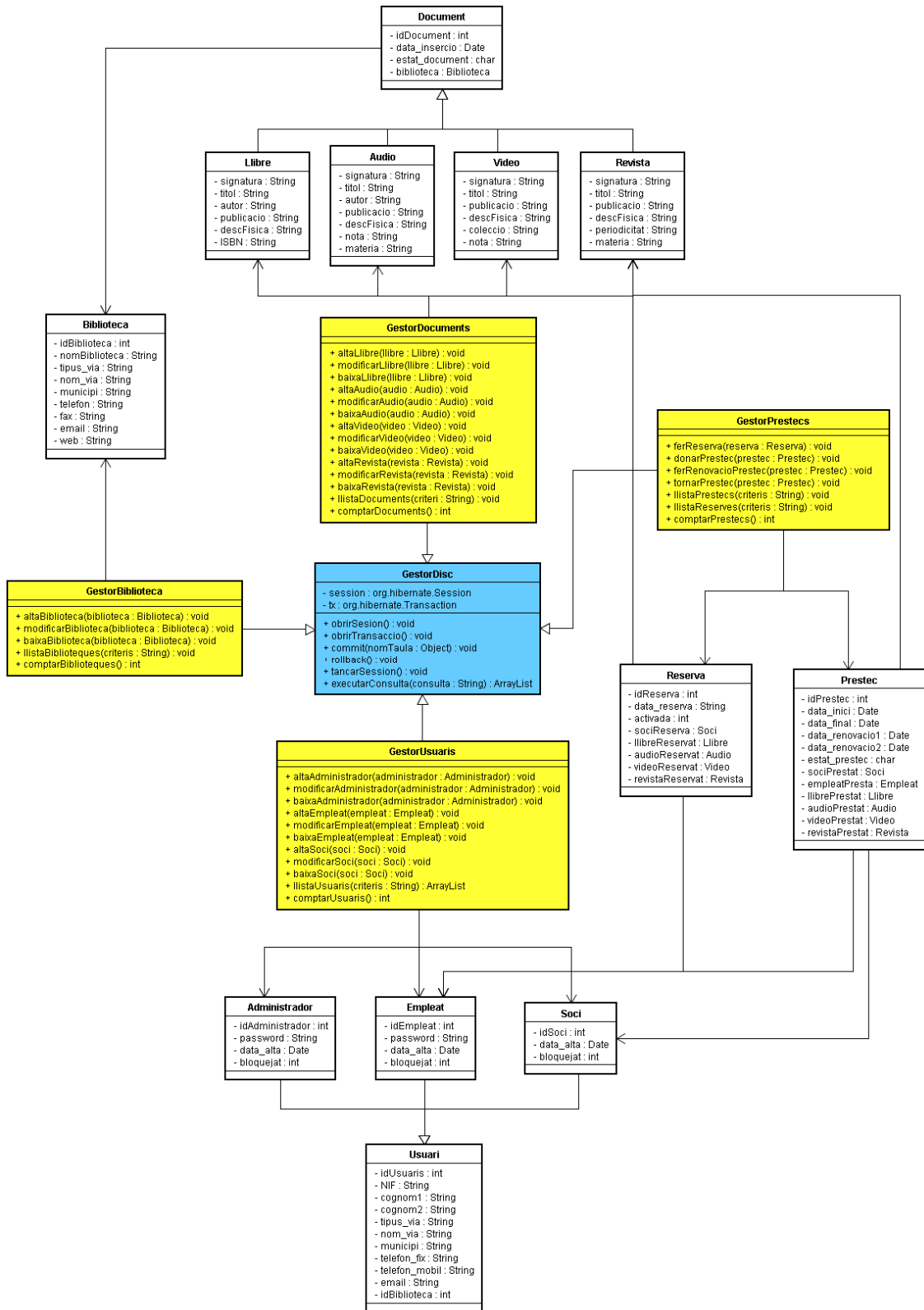
4.6.1.2.1 Introducció

Les classes del Model són totes aquelles que tenen a veure amb les dades i el tipus de problema resolt. Això ens porta al següent diagrama estàtic

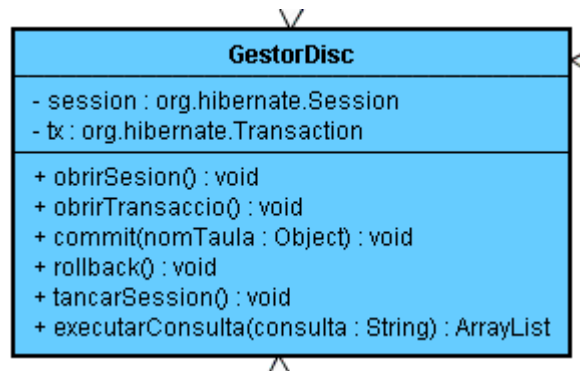
4.6.1.2.2 Diagrama estàtic de classes

Dins d'aquest diagrama estàtic podem distingir dos grups diferents:

- Les classes que corresponen a entitats del problema
- Les classes gestores corresponents a les entitats que requereixen operacions especials de persistència a la SGBD. Aquestes s'anomenen GestorX, sent X la classe entitat a la qual donen suport.



4.6.1.2.3 GestorDisc



Aquesta és la classe que s'encarrega de les interaccions amb la SGBD. Comentar que l'aplicació que estem construint fa ús de les llibreries Hibernate. Per tant, és una mica diferent a qualsevol sistema JDBC o ODBC. En aquest cas necessitarem obrir una sessió amb l'hibernate i comunicar-li que anem a fer una transacció.

obrirSesion()

Obre una sessió amb l'hibernate

obrirTransaccio()

Es comença una transacció amb la base de dades a través de l'hibernate

commit()

Fem un commit per a que totes les accions es traspassin a la DB.

rollback()

Retrocedeix el fet fins ara de la transacció en curs

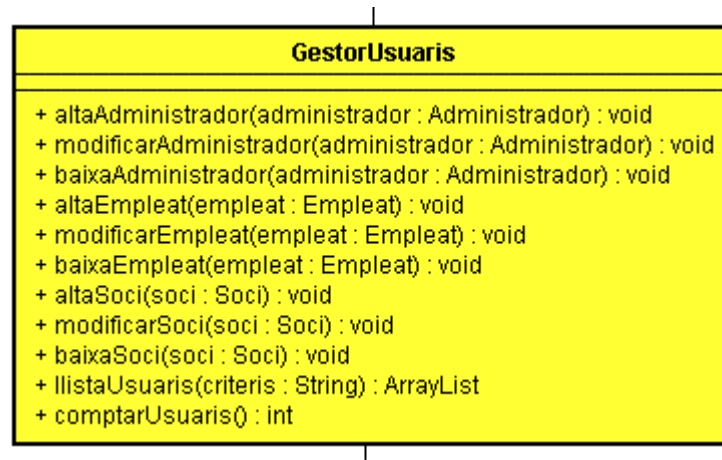
tancarSession()

Tanquem la sessió amb l'hibernate

executarConsulta()

Executem una consulta

4.6.1.2.4 GestorUsuaris



Aquesta classe interactua amb la taula d'usuaris. Els mètodes que ofereix són:

`altaAdministrador()`

Insereix un administrador a la taula Administrador de la base de dades

`modificarAdministrador()`

Modificarà dades de l'administrador a la base de dades

`baixaAdministrador()`

Posarà en estat bloquejat a l'administrador a la base de dades

`altaEmpleat()`

Insereix un empleat a la taula Empleat de la base de dades

`modificarEmpleat()`

Modificarà dades de l'empleat a la base de dades

`baixaEmpleat()`

Posarà en estat bloquejat a l'empleat a la base de dades

`altaSoci()`

Insereix un soci a la taula Soci de la base de dades

`modificarSoci()`

Modificarà dades del soci a la base de dades

`baixaSoci()`

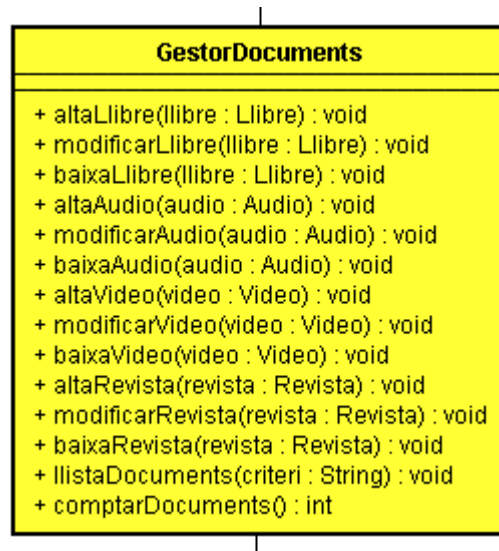
Posarà en estat bloquejat al soci a la base de dades

`llistaUsuaris()`

Torna una llista d'usuaris a partir dels criteris de selecció indicats.

comptarUsuaris()
 Compta el nombre d'usuaris actius a la taula

4.6.1.2.5 GestorDocuments



altaLlibre()
 Dona d'alta un llibre a la base de dades

modificarLlibre()
 Modifica dades d'un llibre a la base de dades

baixaLlibre()
 Canviarà l'estat del llibre per a que no es pugui donar en préstec

altaAudio()
 Dona d'alta un àudio a la base de dades

modificarAudio()
 Modifica dades d'un àudio a la base de dades

baixaAudio()
 Canviarà l'estat del àudio per a que no es pugui donar en préstec

altaVideo()
 Dona d'alta un vídeo a la base de dades

modificarVideo()

Modifica dades d'un vídeo a la base de dades

baixaVideo()

Canviarà l'estat del vídeo per a que no es pugui donar en préstec

altaRevista()

Dona d'alta una revista a la base de dades

modificarRevista()

Modifica dades d'un revista a la base de dades

baixaRevista()

Canviarà l'estat del revista per a que no es pugui donar en préstec

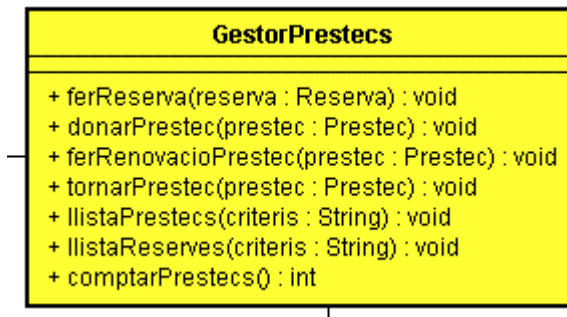
llistaDocuments()

Torna una llista de documents a partir dels criteris de selecció indicats.

comptarDocuments()

Compta el nombre de documents actius a la taula

4.6.1.2.6 Gestor Préstecs



ferReserva()

Insereix a la BD una reserva que faci el Soci o l'empleat

donarPrestec()

Insereix un préstec donat a un Soci

ferRenovacio()

Fa la renovació d'un préstec a un Soci.

tornarPrestec()

Canvia l'estat del préstec indicant que el document ha estat tornat

llistaPrestecs()

Dona una llista de préstecs en relació a uns criteris de recerca

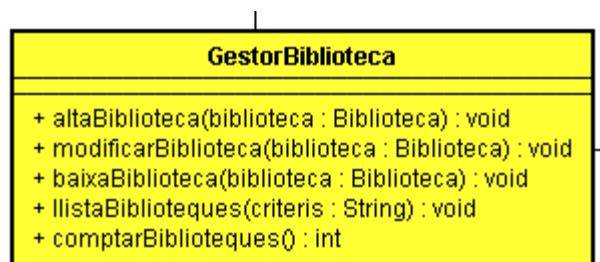
llistaReserves()

Dona una llista de reserves en relació a uns criteris de recerca

comptarPrestecs()

Retorna el número de préstecs actius en l'actualitat

4.6.1.2.7 Gestor Biblioteques



altaBiblioteca()

Dona d'alta una biblioteca

modificarBiblioteca()

Modifica dades de la biblioteca

baixaBiblioteca()

Bloqueja la biblioteca

llistaBiblioteques()

A partir d'uns criteris de recerca, ens dona una llista de biblioteques

comptarBiblioteques()

Ens dona el número de biblioteques actuals

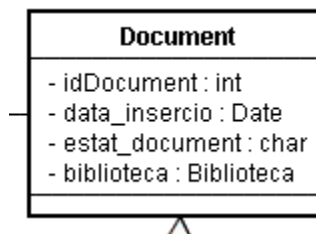
4.6.1.2.8 Biblioteca



La classe biblioteca conté les dades de la biblioteca a la que pertanyen els documents.

4.6.1.2.9 Document

És una classe pare que conté dades comunes a altres classes: Llibre, Audio, Video, Revista.



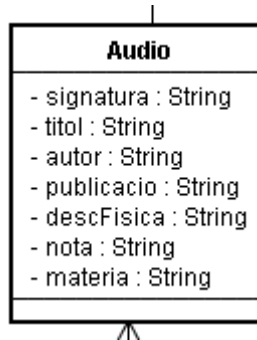
4.6.1.2.10 Llibre

És la classe que representa a un Llibre.



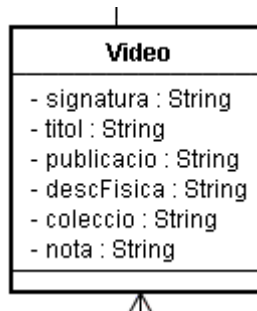
4.6.1.2.11 *Audio*

Aquesta classe representa aquells documents en format Audio, per exemple, cd's.



4.6.1.2.12 *Video*

Classe que conté els documents en format de Video, com per exemple DVD's o VHS.

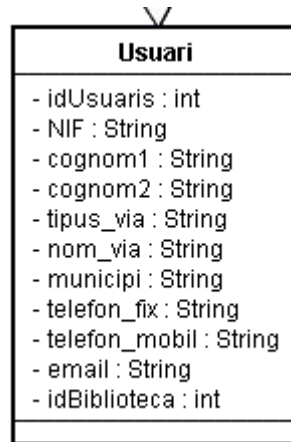


4.6.1.2.13 *Revistes*

És la classe que conté documents del tipus de Revistes.



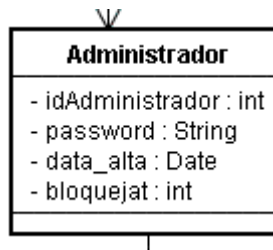
4.6.1.2.14 *Usuari*



La classe Usuari conté les dades comunes entre els tres tipus de usuaris: Administrador, Empleat i Soci.

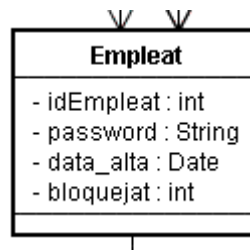
4.6.1.2.15 *Administrador*

La classe que conté un administrador.



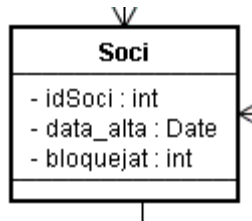
4.6.1.2.16 *Empleat*

La classe que conté un administrador.



4.6.1.2.17 *Soci*

La classe que conté un administrador.



4.6.1.2.18 *Reserva*

La classe Reserva ens dona informació de una reserva d'un document per part d'un soci.



4.6.1.2.19 *Préstec*



La classe Prestec conté les dades d'un préstec d'un document fet per un empleat a un soci.

5 Implementació

5.1 *Requeriments del programari*

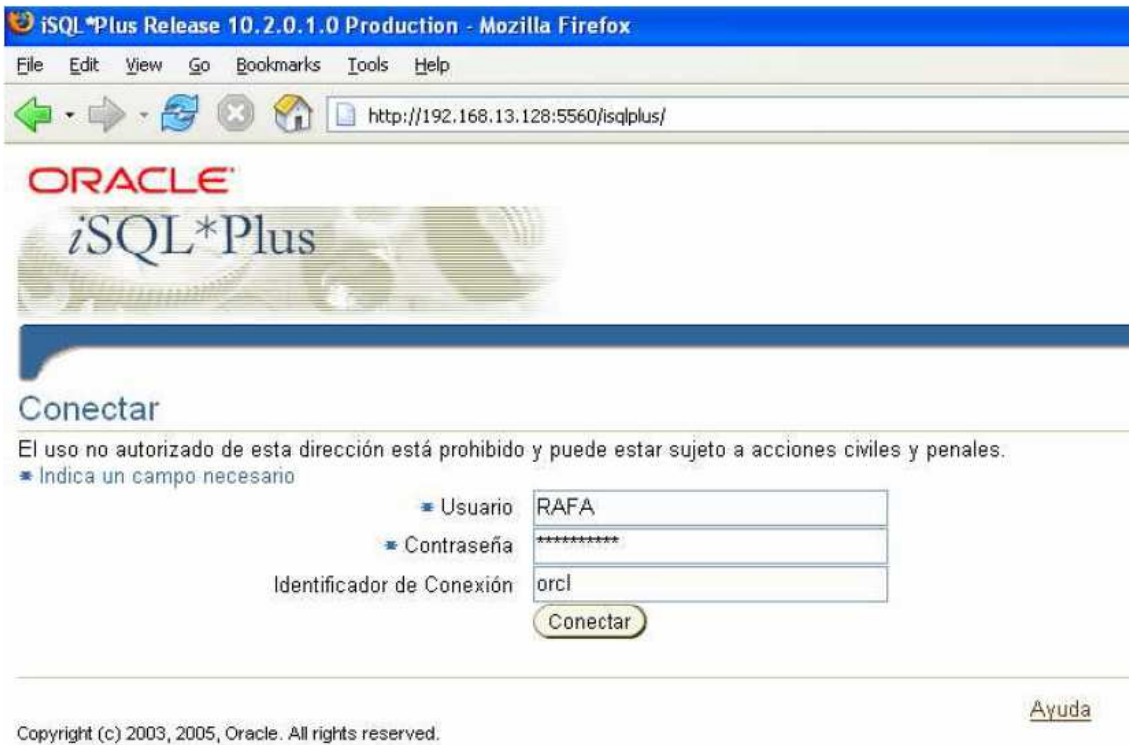
El que es necessita per implementar aquest sistema és el que destaquem a continuació entre programari i configuracions:

- Navegador de pàgines web que suportin HTML i que permetin l'execució de codi JavaScript. Aquest últim es necessita per determinats aspectes visuals de la interfície.
- J2SE 1.5 update 3 (SDK)
- Servidor d'aplicacions. Nosaltres utilitzem el JBoss 4.0.4 GA
- Servidor web. Nosaltres utilitzem el Apache Tomcat. JBoss conté aquest servidor web.
- SGBD. Utilitzem Oracle 10g per ser una base de dades potent i amb molta seguretat.
- Llibreries de connexió JDBC per poder connectar el SGBD amb l'aplicació final.
- Llibreries del Framework Apache Struts (MVC) pel desenvolupament de la capa web. Struts inclou Tiles.
- Llibreries d'Hibernate per la persistència. Amb això connectem l'aplicació amb l'ajuda dels drivers de connexió JDBC amb la base de dades d'Oracle.

5.2 *Eines de desenvolupament*

Com a eines de desenvolupament hem de pensar en totes aquelles eines que tenim que utilitzar per poder portar a terme aquest projecte.

- Utilitzem com a IDE l'Eclipse 3.0.2 (programari lliure i open source)
- Plugin JBoss JBossIDE-1.6.0.GA-ALL
- Eines que proporciona Oracle 10g per poder gestionar la base de dades (iSQLPlus).



6 Valoració econòmica

Tots els components que s'han utilitzat al projecte són tots gratuïts. Això sí, les llicències per cada component s'han de consultar dins de la web de cada fabricant. La plataforma que s'ha utilitzat per desenvolupar i provar l'aplicació és el sistema operatiu Windows. Aquest no es gratuït. Podem dir que tot el programari que s'utilitza té versions per a altres sistemes operatius com, per exemple, qualsevol distribució de LINUX.

Ara bé, no és objectiu del TFC donar una valoració molt estudiada de lo que pot suposar una implantació de l'aplicació. Bàsicament per la gran quantitat d'hores que s'han utilitzat en l'aprenentatge de l'arquitectura J2EE i dels frameworks Struts, Tiles i Hibernate.

7 Conclusions

Voldria dividir les conclusions d'aquest TFC en dos aspectes que van molt relacionats en la meua persona i que aspiren moltes vegades al mateix objectiu però que voldria comentar per separat.

- Aspecte personal

El meu interès per la programació sempre ha estat molt gran i volia aprofitar tots els meus coneixements que he anat aprenent al llarg de la carrera en el desenvolupament d'un projecte molt més real i molt més complex dels que he anat fent fins ara.

Vaig decidir-me ràpidament per la tecnologia J2EE ja que penso que actualment és una de les tecnologies punteres en el desenvolupament de programari. A més, volia començar a rebre coneixements relacionats amb tot el que es la programació distribuïda de programari ja que no he fet cap cosa important a l'enginyeria tècnica.

Referent als frameworks que hem aplicat al TFC comentar que tenia molt clar que ho volia fer amb Struts ja que tenia molta curiositat d'aprendre com funcionava i assolir els coneixements necessaris per poder aplicar-ho en qualsevol altre projecte que tingui que fer en el futur. En el cas d'Hibernate, al principi no tenia clar si utilitzar JDBC o un ORM i al final vaig triar Hibernate per voler ampliar coneixements als que ja tenia ja que JDBC si que havia fet durant la carrera.

També volia remarcar l'ús de la base de dades Oracle 10g, eina que no havia utilitzat en cap moment durant la carrera i que, personalment, tenia moltes de ganes de treballar amb ella i trobar diferències (avantatges si tingués) amb altres com Informix que si que havia utilitzat.

- Aspecte professional

El fet de desenvolupar un TFC en J2EE hem dona tots els coneixements necessaris per poder afrontar molts reptes professionals que hem siguin assignats en aquest marc de treball.

En aquest TFC s'utilitzen uns frameworks que són molt coneguts i molt utilitzats com són Struts, Tiles i Hibernate. Moltes empreses utilitzen aquests frameworks en moltes implementacions ja que donen solucions a molts problemes que apareixen quan es desenvolupa programari. El paradigma MVC s'ha implementat donant unes característiques al TFC que s'intentaven

trobar des del principi i crec que s'ha aconseguit:

- Escalable
- Robusta
- Reutilitzable

8 Glossari

- Administrador: persona encarregada de la gestió d'usuaris i configuració de l'aplicació de la biblioteca.
- Empleat: usuari que treballa a la biblioteca i que pot gestionar documents, socis i préstecs.
- Socis: usuari clau dins d'una biblioteca ja que es a qui es presta els documents d'aquesta.
- Préstecs: préstec de documents a un soci per part d'una biblioteca.
- Reserves: reserves de documents en un biblioteca per part d'un soci.
- Documents: documents d'una biblioteca que poden ser un llibre, de format d'àudio o de vídeo , o revistes

9 Bibliografia

Wiki Struts

<http://wiki.apache.org/struts/>

JavaDoc Struts

<http://struts.apache.org/struts-action/apidocs/index.html>

HIBERNATE - Relational Persistence for Idiomatic Java

http://www.hibernate.org/hib_docs/v3/reference/en/html/index.html

Diseño de aplicaciones Internet usando los Patrones de diseño J2EE

http://www.programacion.com/java/articulo/corej2ee_patterns/

Aplicación de Patrones J2EE en un Caso de Estudio

<http://www.programacion.com/java/articulo/inukisoft/>

Manual de CSS

<http://www.desarrolloweb.com/manuales/2/>

MVC aplicat a Java

<http://java.sun.com/blueprints/patterns/MVC-detailed.html>

MVC avantatges i inconvenients

<http://wact.sourceforge.net/index.php/ModelViewController>

MVC definició a Wikipedia

http://en.wikipedia.org/wiki/Model_view_controller

JavaServerPages

<http://java.sun.com/products/jsp/technical.html>

Jboss Doc

<http://labs.jboss.com/portal/jbossas/docs>

10 Annex 1. Manual d'usuari

10.1 Introducció

El següent manual explica les principals funcions d'usuari que es pot trobar a l'aplicació. Dividirem el manual en tres parts:

- Manual del soci
- Manual de l'empleat
- Manual de l'administrador

La divisió la hem fet d'aquesta manera perquè considerem que aquests usuaris són els que tenen diferents formes d'actuar dins l'aplicació. No fa les mateixes funcionalitats un soci que un empleat. Això sí, tenen pantalles que són molt semblants i que fan el mateix però amb menys o més limitacions.

10.2 Manual del soci

El soci podrà entrar a l'aplicació des de l'adreça d'Internet principal:

http://ip_de_la_biblioteca/LibraryWeb/

Aquesta url porta a la següent pantalla:



El soci podrà fer consultes en qualsevol Biblioteca que estigui donada d'alta a l'aplicació. També podrà fer una recerca pel tipus de document especificat (Llibre, Audio, Video o Revista). La recerca es pot fer per diferents tipus de paràmetres (Títol, Autor, etc). Depenent del tipus de document.

La recerca està preparada per retornar resultats de paraules o frases. Si no les troba iguals, donarà aquells resultats que comencin amb el que s'ha entrat. Un exemple és la següent il·lustració on entrem "Shake". Això donarà tots aquells Autors que comencin amb "Shake". Es per això que esperem que ens retorni documents de Shakespeare.



El resultat que dona aquesta recerca són els següents:

The screenshot shows a web browser window with the URL `http://localhost:8080/LibraryWeb/resultatsDocumentsSoci.do;jsessionid=A47A435A158106B6E8116FECD3AA51A`. The page header includes the logo 'blioteca' and navigation links: 'Accions', 'Catàleg', 'Reserva Documents', and 'Login'. The main content area is titled 'resultat recerca' and displays 'Resultats de la recerca: 2 elements amb la recerca "Shake"'. Below this is a table with two rows of search results, each with a 'Reservar' button.

Entrada	Biblioteca	Codi	Títol	Autor	Publicació	ISBN	
1	1	2	Comedias	Shakespeare, William	RBA	24681434	Reservar
2	1	1	Cimbelf	Shakespeare, William	Institut del Teatre	24681434	Reservar

Com veiem a la il·lustració, l'aplicació dona uns resultats i li dona l'opció al soci de poder fer una reserva del document que vulgui. Això sí, per poder fer la reserva d'algun tipus de document, haurà d'estar autenticat. Es per això que si el soci prem el botó de Reservar i no estava autenticat prèviament, l'aplicació li portarà a la pantalla de Login. La pantalla és la següent:

The screenshot shows a web browser window with the URL `http://192.168.1.2:8080/LibraryWeb/loginSoci.do`. The page header includes the logo 'Biblioteca' and navigation links: 'Menu d'acions', 'Catàleg', 'Reserva Documents', and 'Login'. The main content area is titled 'login' and features a form for user authentication with fields for 'NIF' and 'Codi d'usuari', and a 'Login' button.

Autenticació

Inserta un nif i un password correctes per autenticar-te.

NIF

Codi d'usuari

Login

Com podem veure a la pantalla, per autenticar-se es necessita el número de NIF i el codi d'usuari que la biblioteca li dona una vegada el dona d'alta. Normalment, aquest codi d'usuari anirà en els carnets que la biblioteca dona després de l'alta.

Una vegada s'ha autenticat tindrà accés a la pantalla de Reserva i podrà acceptar la reserva feta.

10.3 Manual de l'empleat

L'empleat sols podrà entra a l'aplicació autenticant-se. Així que la primera pantalla que veurà l'empleat serà la d'un formulari on tindrà que introduir el seu NIF i la seva contrasenya. La pantalla és la següent:



Autenticació
 Inserta un NIF i un password correctes per autenticar-te.

login

Biblioteca1

NIF
53079837T

Password

Tipus d'usuari
Empleat

Login

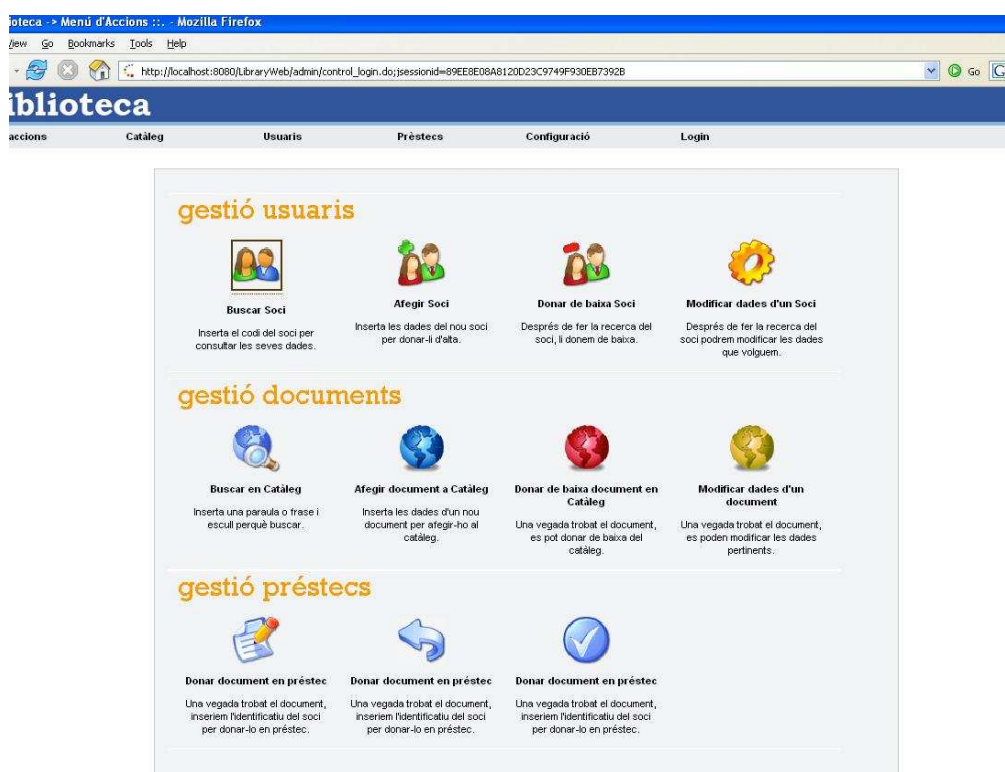
L'empleat haurà de dir en quina biblioteca està. No s'ha d'oblidar de marcar que entra com a Empleat ja que aquesta pantalla la comparteixen Administrador i Empleat. En el cas de que intentés autenticar-se com Administrador, l'aplicació ho comunicaria com error.

Quan l'empleat finalment s'autentifica tindrà accés a les diferents funcionalitats

que dona l'aplicació: Gestor d'usuaris, Gestor de documents i Gestor de Préstecs. Abans es mostrarà una pantalla d'ajuda per l'empleat on podrà veure les possibles accions a fer.

Comentar que aquesta pantalla és simplement d'ajuda. D'aquesta manera l'empleat pot saber ràpidament que funcionalitats li pot oferir l'aplicació i si ho suporta. Com veiem a la pantalla, es divideix en les funcionalitats abans esmentades:

- Gestor d'usuaris
- Gestor de documents
- Gestor de préstecs



A partir de la barra de menú es poden anar a les diferents accions.



10.3.1 Gestor d'usuaris

A la pantalla del gestor d'usuaris s'arriba a través de "Usuaris" de la barra de l'Empleat. Aquesta pantalla s'ha dividit en dos parts:

- Un formulari destinat a fer diferents funcions donant un codi d'usuari (pensem que el codi del soci vindrà informat als carnets de soci i es podria, per exemple, llegir amb un lector de codi de barres).
- Un formulari destinat a introduir dades d'usuari i donar d'Alta a aquests.

La podem veure en la següent captura:

The image shows two screenshots of a web application interface. The top screenshot is titled 'socs' and features a search form for existing members. It includes a navigation bar with links for 'Catàleg', 'Usuaris', 'Prèstecs', 'Configuració', and 'Login'. The search form has a text input for 'Codi del soci', a dropdown menu for 'Llista d'accions' with 'Consultar dades' selected, and a 'Buscar' button. The bottom screenshot is titled 'alta de soci' and shows a registration form. It is divided into sections: 'Dades personals:' with fields for 'NIF', 'Nom', 'Primer Cognom', and 'Segon Cognom'; 'Adreça:' with fields for 'Tipus de Via' (set to 'CL'), 'Nom de la Via', 'Municipi', and 'Provincia'; and 'Contacte:' with fields for 'Telèfon Fixe', 'Telèfon Mòbil', and 'E-m@il'. At the bottom are buttons for 'Donar d'alta' and 'Limpiar formulari'.

En el formulari de socis es poden fer les següents funcions:

- Consultar dades → Si es consulta les dades surt un formulari amb les dades del soci.
- Modificar dades → Si es vol modificar les dades d'un soci, sortirà un formulari amb les dades del soci i es podrà modificar qualsevol de les dades.
- Bloquejar usuari → Es bloqueja l'usuari i no podrà agafar cap préstec ni fer reserves.
- Desbloquejar usuari → Desbloqueja un usuari anteriorment bloquejat.
- Donar Préstec → Dona un document en préstec al soci. Aquesta funció se redirigeix cap al gestor de préstecs.
- Fer Reserva → Reserva un document per al soci per a un posterior préstec. Aquesta funció es redirigeix cap al gestor de préstecs.

10.3.2 Gestor de documents

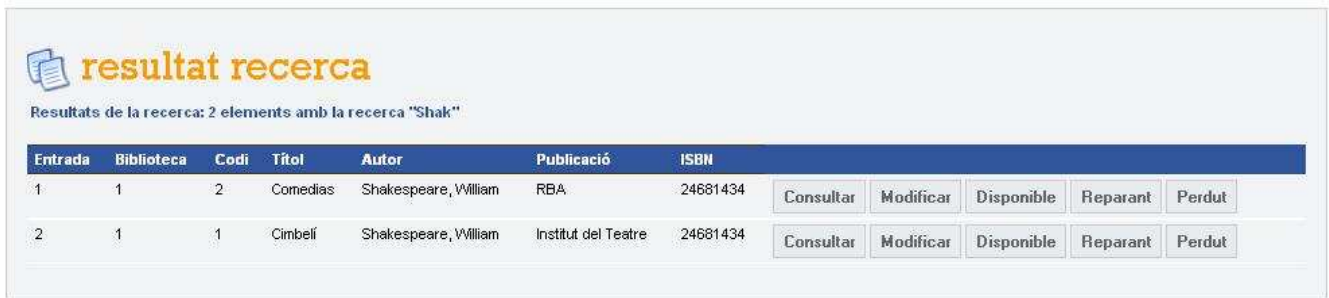
A la pantalla del gestor de documents s'arriba per "Catàleg" del menú de l'Empleat. Aquesta pantalla està dividida en dos parts:

- Un formulari que serveix per fer recerques d'un document en qüestió. Es pot fer consulta pel tipus de document especificat (Llibre, Audio, Video o Revista). La recerca es pot fer per diferents tipus de paràmetres (Títol, Autor, etc). Depenent del tipus de document. Comentar per últim, que dona la funció de poder fer la recerca per codi de document. D'aquesta manera es podria inserir un codi de document i poder fer la recerca per aquest.
- Un formulari per fer l'Alta d'un document en qüestió. S'insereix les dades del nou document i es procedeix a fer l'alta.

El formulari de recerca de documents es molt semblant a la pantalla de recerca de documents del soci. El funcionament és idèntic amb el mateix tipus de recerca avançada.

En la gestió de documents comentar que tenim una llista desplegable que ens dona el quatre tipus de documents que té la biblioteca. Depenent de tipus que agafem, els camps s'activen o no per poder inserir dades. Una vegada ja la tenim, simplement falta donar d'alta.

Això era en el cas de fer altes de documents. En el cas d'altres tipus d'accions sobre els documents, abans de tot haurem de fer una recerca amb el formulari de recerca del document en qüestió. Per exemple, en la següent imatge tenim una recerca que hem fet:



resultat recerca

Resultats de la recerca: 2 elements amb la recerca "Shak"

Entrada	Biblioteca	Codi	Títol	Autor	Publicació	ISBN	Consultar	Modificar	Disponible	Reparant	Perdut
1	1	2	Comedias	Shakespeare, William	RBA	24681434	Consultar	Modificar	Disponible	Reparant	Perdut
2	1	1	Cimbelí	Shakespeare, William	Institut del Teatre	24681434	Consultar	Modificar	Disponible	Reparant	Perdut

Com es pot veure, hi han uns botons que ens deixen fer més accions sobre els documents que vulguem. Aquestes són:

- Consultar → Ens mostra més dades del document.
- Modificar → Ens mostra més dades del document i ens deixa modificar-les.
- Disponible → Canvia l'estat del document a disponible / no disponible.
- Reparant → Canvia l'estat del document a reparant / no reparant.
- Perdut → Canvia l'estat del document a perdut / no perdut.

Com veiem, els tres últims botons canvia serveixen per canviar l'estat del document en qüestió. S'ha de dir que en les tres opcions sortirà una pantalla de confirmació per fer l'acció.

10.3.3 Gestor de préstecs

A la pantalla del gestor de préstecs s'arriba a partir de "Préstecs" de la barra de l'empleat. Aquesta ens portarà a una pantalla amb un formulari on posant el codi de soci i el codi del document, podrem fer diferents accions. Aquestes accions són:

- Fer Préstec
- Fer Renovació
- Fer Reserva
- Retornar Préstec
- Llistar Préstec → En aquest simplement necessitem el codi de soci

La pantalla és la següent:



En les accions de préstec, el resultat de les accions ens portarà a altra pantalla on ens mostrarà el llistat de préstecs que té aquest soci. La pantalla es la següent:

N° Entrada	Títol Document	Tipus	Estat Préstec	Data d'entrega
1	Comedias	Llibre	A	2006-07-18 15:47:54.0
2	Cimbelfí	Llibre	C	2006-07-03 15:42:46.0

Com veiem, aquesta pantalla ens dona informació dels préstecs que té agafats el soci en aquesta biblioteca. La dada d'Estat de Préstec és molt important ja que ens diu el moment en el que es troba el préstec en aquests moments:

- Estat A: Acabat de fer el préstec
- Estat B: S'ha renovat una vegada
- Estat C: S'ha renovat dos vegades

A partir de l'estat C, ja no es podrà renovar més i s'haurà d'entregar el document a la biblioteca en la data d'entrega.

10.4 Manual de l'administrador

Per poder autenticar-se com a administrador ho haurà de fer com a tal a la pantalla de login. Una vegada autenticat, es mostrarà una pantalla d'ajuda amb accions que pot arribar a fer l'administrador.

Comentar que pot gestionar tots els usuaris (empleats, socis i administradors) i que les úniques diferències que es veuen a simple vista són unes llista desplegable on haurà de triar el tipus d'usuari a donar d'alta, modificar, etc.

Una de les funcionalitats que té l'administrador diferent als altres usuaris és la de configuració. En aquest cas podrà configurar les biblioteques i afegir més a la base de dades de l'aplicació. Simplement es tenen que afegir les dades de la nova biblioteca. A partir de l'alta, es podrà començar a afegir documents al catàleg d'aquesta biblioteca (responsabilitat de l'empleat).

La pantalla d'alta de biblioteques és la següent:

Catàleg	Usuaris	Prèstecs	Configuració	Login
---------	---------	----------	--------------	-------



Afegir una Biblioteca
Inserta les dades d'una nova biblioteca.

configuració

Alta ▼

Dades de la biblioteca:

Nom	Tipus Via	Nom Via
<input type="text"/>	<input type="text"/>	<input type="text"/>
Municipi	Telefon	Fax
<input type="text"/>	<input type="text"/>	<input type="text"/>
Email	Web	
<input type="text"/>	<input type="text"/>	

11 Annex 2. Sentències SQL

```

/*
Created          28/03/2006
Modified         29/05/2006
Project         Treball Fi de Carrera
Model           Base de dades de @Biblioteca
Company         UOC
Author          Rafael Alonso Jurado
Version         1.1
Database        Oracle 10g
*/

```

```
ROLLBACK;
```

```

Drop table "Prestecs";
Drop table "Reserves";
Drop table "Llibres";
Drop table "Revistes";
Drop table "Audio";
Drop table "Video";
Drop table "Documents";
Drop table "Administradors";
Drop table "Socis";
Drop table "Empleats";
Drop table "Usuaris";
Drop table "Biblioteques";

```

```
-- Creació de Taules
```

```

Create table "Biblioteques" (
    "idBiblioteca" Number(5) NOT NULL ,
    "nomBiblioteca" Varchar2 (50) NOT NULL ,
    "tipus_via" Varchar2 (3) NOT NULL ,
    "nom_via" Varchar2 (50) NOT NULL ,
    "municipi" Varchar2 (50) NOT NULL ,
    "telefon" Varchar2 (9) NOT NULL ,
    "fax" Varchar2 (9) NOT NULL ,
    "email" Varchar2 (20),
    "web" Varchar2 (20),
    primary key ("idBiblioteca")
);

```

```

Create table "Usuaris" (
    "idUsuaris" Number NOT NULL ,
    "NIF" Varchar2 (9) NOT NULL ,
    "nom" Varchar2 (20) NOT NULL ,

```



```

    "cognom1" Varchar2 (20) NOT NULL ,
    "cognom2" Varchar2 (20) NOT NULL ,
    "tipus_via" Varchar2 (3),
    "nom_via" Varchar2 (50),
    "municipi" Varchar2 (50),
    "provincia" Varchar2 (50),
    "telefon_fix" Varchar2 (9),
    "telefon_mobil" Varchar2 (9),
    "email" Varchar2 (20),
    "idBiblioteca" Number NOT NULL ,
primary key ("idUsuaris")
);

```

```

Create table "Empleats" (
    "idUsuaris" Number NOT NULL ,
    "idEmpleat" Number NOT NULL ,
    "password" Varchar2 (8) NOT NULL ,
    "data_alta" Date NOT NULL ,
    "bloquejat" Smallint NOT NULL ,
primary key ("idUsuaris","idEmpleat")
);

```

```

Create table "Socis" (
    "idUsuaris" Number NOT NULL ,
    "idSoci" Number NOT NULL ,
    "data_alta" Date NOT NULL ,
    "bloquejat" Smallint NOT NULL ,
primary key ("idUsuaris","idSoci")
);

```

```

Create table "Administradors" (
    "idUsuaris" Number NOT NULL ,
    "idAdministrador" Number NOT NULL ,
    "password" Varchar2 (8) NOT NULL ,
    "data_alta" Date NOT NULL ,
    "bloquejat" Smallint NOT NULL ,
primary key ("idUsuaris","idAdministrador")
);

```

```

Create table "Documents" (
    "idBiblioteca" Number NOT NULL ,
    "idDocuments" Number NOT NULL ,
    "data_insercio" Date NOT NULL ,
    "estat_document" Char (1) NOT NULL ,
    "tipus_document" Char (2) NOT NULL,
primary key ("idBiblioteca","idDocuments")
);

```

```

Create table "Video" (

```

```

    "idBiblioteca" Number NOT NULL ,
    "idDocuments" Number NOT NULL ,
    "idVideo" Number NOT NULL,
    "signatura" Varchar2 (20) NOT NULL ,
    "titol" Varchar2 (20) NOT NULL ,
    "publicacio" Varchar2 (20),
    "format" Varchar2 (10),
    "descFisica" Varchar2 (50),
    "coleccio" Varchar2 (20),
    "nota" Varchar2 (50),
primary key ("idBiblioteca","idDocuments","idVideo")
);

```

```

Create table "Audio" (
    "idBiblioteca" Number NOT NULL ,
    "idDocuments" Number NOT NULL ,
    "idAudio" Number NOT NULL,
    "signatura" Varchar2 (20) NOT NULL ,
    "titol" Varchar2 (20) NOT NULL ,
    "autor" Varchar2 (20) NOT NULL ,
    "publicacio" Varchar2 (20),
    "format" Varchar2 (10),
    "descFisica" Varchar2 (50),
    "nota" Varchar2 (50),
    "materia" Varchar2 (20),
primary key ("idBiblioteca","idDocuments","idAudio")
);

```

```

Create table "Revistes" (
    "idBiblioteca" Number NOT NULL ,
    "idDocuments" Number NOT NULL ,
    "idRevista" Number NOT NULL,
    "signatura" Varchar2 (20) NOT NULL ,
    "titol" Varchar2 (20) NOT NULL ,
    "publicacio" Varchar2 (20),
    "descFisica" Varchar2 (50),
    "periodicitat" Varchar2 (20),
    "materia" Varchar2 (20),
primary key ("idBiblioteca","idDocuments","idRevista")
);

```

```

Create table "Llibres" (
    "idBiblioteca" Number(5) NOT NULL ,
    "idDocuments" Number NOT NULL ,
    "idLibre" Number NOT NULL,
    "signatura" Varchar2 (20) NOT NULL ,
    "titol" Varchar2 (50) NOT NULL ,
    "autor" Varchar2 (50) NOT NULL ,
    "publicacio" Varchar2 (20),

```

```

        "descFisica" Varchar2 (50),
        "ISBN" Varchar2 (20) NOT NULL ,
primary key ("idBiblioteca","idDocuments","idLlibre")
);
Create table "Reserves" (
        "idReserves" Number NOT NULL ,
        "data_reserva" Date NOT NULL ,
        "activada" Smallint NOT NULL ,
        "idUsuaris" Number NOT NULL ,
        "idSoci" Number NOT NULL ,
        "idBiblioteca" Number NOT NULL ,
        "idDocuments" Number NOT NULL ,
primary key ("idReserves")
);

Create table "Prestecs" (
        "idPrestec" Number NOT NULL ,
        "data_inici" Date NOT NULL ,
        "data_final" Date NOT NULL ,
        "data_renovacio1" Date,
        "data_renovacio2" Date,
        "estat_prestec" Char (20) NOT NULL ,
        "idUsuaris1" Number NOT NULL ,
        "idEmpleat" Number NOT NULL ,
        "idBiblioteca" Number NOT NULL ,
        "idDocuments" Number NOT NULL ,
        "idUsuaris2" Number NOT NULL ,
        "idSoci" Number NOT NULL ,
primary key ("idPrestec")
);

-- Creació de les Foreign keys

Alter table "Usuaris" add foreign key ("idBiblioteca") references "Biblioteques"
("idBiblioteca") ;
Alter table "Documents" add foreign key ("idBiblioteca") references "Biblioteques"
("idBiblioteca");
Alter table "Empleats" add foreign key ("idUsuaris") references "Usuaris" ("idUsuaris");
Alter table "Socis" add foreign key ("idUsuaris") references "Usuaris" ("idUsuaris");
Alter table "Administradors" add foreign key ("idUsuaris") references "Usuaris"
("idUsuaris");
Alter table "Prestecs" add foreign key ("idUsuaris1","idEmpleat") references "Empleats"
("idUsuaris","idEmpleat");
Alter table "Reserves" add foreign key ("idUsuaris","idSoci") references "Socis"
("idUsuaris","idSoci");
Alter table "Prestecs" add foreign key ("idUsuaris2","idSoci") references "Socis"
("idUsuaris","idSoci");
Alter table "Video" add foreign key ("idBiblioteca","idDocuments") references

```

```
"Documents" ("idBiblioteca","idDocuments");
Alter table "Audio" add foreign key ("idBiblioteca","idDocuments") references
"Documents" ("idBiblioteca","idDocuments");
Alter table "Llibres" add foreign key ("idBiblioteca","idDocuments") references
"Documents" ("idBiblioteca","idDocuments");
Alter table "Revistes" add foreign key ("idBiblioteca","idDocuments") references
"Documents" ("idBiblioteca","idDocuments");
Alter table "Prestecs" add foreign key ("idBiblioteca","idDocuments") references
"Documents" ("idBiblioteca","idDocuments");
Alter table "Reserves" add foreign key ("idBiblioteca","idDocuments") references
"Documents" ("idBiblioteca","idDocuments");
```

-- Creació d'unes Sequences

```
Drop sequence incremento_idBiblioteca;
Drop sequence incremento_idUsuaris;
Drop sequence incremento_idPrestec;
Drop sequence incremento_idReserva;
```

```
Create sequence incremento_idBiblioteca increment by 1;
Create sequence incremento_idUsuaris increment by 1;
Create sequence incremento_idPrestec increment by 1;
Create sequence incremento_idReserva increment by 1;
```

-- Inserts

```
INSERT INTO "Biblioteques" VALUES
(incremento_idBiblioteca.nextVal,'Biblioteca1','CL','Sagarra','Santa Coloma de
Gramenet','677769888','933856719','ralonsoj@gmail.com',null);
INSERT INTO "Biblioteques" VALUES
(incremento_idBiblioteca.nextVal,'Biblioteca2','CL','Osona','Sabadell','677769888','93385
6719','ralonsoj@gmail.com',null);
```

```
INSERT INTO "Documents" VALUES (1,1,SYSDATE,'D','L');
INSERT INTO "Llibres" VALUES (1,1,1,'NOVELA','Cimbelí','Shakespeare,
William','Institut del Teatre','descripcio','24681434');
INSERT INTO "Documents" VALUES (1,2,SYSDATE,'D','L');
INSERT INTO "Llibres" VALUES (1,2,2,'NOVELA','Comedias','Shakespeare,
William','RBA','descripcio','24681434');
INSERT INTO "Documents" VALUES (1,3,SYSDATE,'D','A');
INSERT INTO "Audio" VALUES (1,3,1,'AUDIO','Tu te
vas','Chayane','Sony','format','descripcio','24681434','latino');
INSERT INTO "Documents" VALUES (1,4,SYSDATE,'D','V');
INSERT INTO "Video" VALUES (1,4,1,'VIDEO','Regreso al
futuro','pub','Sony','descripcio','24681434','nota');
INSERT INTO "Documents" VALUES (1,5,SYSDATE,'D','L');
INSERT INTO "Llibres" VALUES (1,5,4,'NOVELA','Jonathan Strange','Clarke,
Susanna','Salamandra','descripcio','24681434');
INSERT INTO "Documents" VALUES (2,1,SYSDATE,'D','L');
```

```
INSERT INTO "Llibres" VALUES (2,1,1,'NOVELA','Cementerio de Animales','King, Stephen','PJ','descripcio','24681434');
```

```
INSERT INTO "Usuaris" VALUES  
(incremento_idUsuaris.nextVal,'Admin','Admin','Admin','Admin',null,null,null,null,  
null,null,1);  
INSERT INTO "Administradors" VALUES (1,1,'Admin',SYSDATE,0);
```

```
COMMIT;
```