

Creació d'una aplicació web per a un videoclub

Enrique Gómez Pelechano

ETIG

Albert Grau Perisé

18/06/2006

Resum

El present projecte final de carrera vol assolir un entorn web per a administrar un videoclub i que els usuaris puguin fer reserves on-line. Així doncs l'aplicació es pot dividir en dos mòduls, un d'administració i un altre de gestió de reserves.

La implementació s'ha fet sobre l'arquitectura J2EE, la qual proporciona totes les eines per fer una aplicació robusta i escalable.

Per a ajudar-nos a aconseguir aquest objectiu ens hem recolzat en el framework Struts per seguir el patró MVC i en l'ús de Tiles per tal de separar la capa de presentació de la capa de model.

Per a la persistència s'ha fet servir Hibernate que ens aïlla totalment de la base de dades i ens permet treballar amb un codi orientat a objectes.

Com a eines complementaries s'han fet servir llibreries de javascript per la validació de formularis i la llibreria quartz per a la programació de tasques.

INDEX

1	Introducció.....	4
1.1	Justificació del TFC: punt de partida i aportació del TFC.....	4
1.2	Objectius del TFC	5
1.3	Enfocament i mètode seguit.....	7
1.4	Planificació del projecte	7
1.5	Productes obtinguts	8
1.6	Descripció de la resta de capítols de la memòria.....	8
2	Arquitectura de l'aplicació	9
2.1	Introducció	9
2.2	Arquitectura J2EE	9
2.3	Patró MVC-Struts.....	12
2.3.1	Us de Tiles	15
2.4	Validacions.....	16
2.5	Persistència	16
2.6	Programació de tasques, utilització de Quartz	17
3	Especificació i Anàlisi de requeriments	18
3.1	Introducció	18
3.2	Descripció del projecte.....	18
3.3	Requeriments funcionals.....	21
3.4	Especificacions de les funcionalitats per subsistema.....	21
3.4.1	Subsistema de gestió d'usuaris	21
3.4.2	Subsistema de gestió d'articles.....	25
3.4.3	Subsistema de gestió de reserves	29
3.5	Interfície gràfica.....	33
3.5.1	Flux de pantalles Administració	34
3.5.2	Flux de pantalles Usuaris.....	41
3.6	Requeriments de maquinari i programari	44
3.7	Seguretat	45
3.8	Possibles funcionalitats addicionals	46
4	Disseny de l'aplicació	48
4.1	Introducció	48
4.2	Disseny de la base de dades	48
4.2.1	Taules de la base de dades.....	50
4.3	Disseny de les classes.....	54
4.3.1	Introducció	54
4.3.2	Diagrama de classes de model.....	56
5	Implementació.....	60
5.1	Eines de desenvolupament.....	60
5.2	Estructura de l'aplicació en paquets.....	61
6	Bibliografia	63

1 Introducció

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

La idea d'aquest projecte es fer un treball fi de carrera que sigui funcional fent servir una tecnologia moderna i actual. Tot i que el temps es limitat es pretén fer un producte que pogués sinó ser la base per fer un producte real.

Les característiques principals que es volen assolir son que l'aplicació sigui:

- Escalable
- Robusta
- Reutilitzable

Seria interessant que permetés una arquitectura distribuïda tot i que l'aplicació esta pensada per que s'utilitzi en un sol ordinador.

Per tal d'assolir aquests objectius cal una arquitectura orientada a objectes com es l'arquitectura J2EE. Dins d'aquest marc de treball hi ha moltes possibilitats (Struts, Spring, EJB..) sent una decissió difícil escollir entre tantes tecnologies.

Es va decidir utilitzar Struts com a marc de treball (Model MVC, model-vista-controlador) doncs ja porta uns quants anys al mercat i esta completament testejat i provada la seva eficacia.

Per a la persistencia es va escollir Hibernate 3.2 front l'us de EJB 3.0 ja que també es tracta d'una tecnologia molt ben posicionada en el mercat.

La motivació personal de fer aquest treball es posar en pràctica tots els coneixments obtinguts durant la carrera per fer un producte complert i totalment

funcional. Cal dir però, que amb aquests coneixements no n'hi prou per fer aquest projecte, així doncs la corba d'aprenentatge cal tenir-la molt present. Hi ha molts coneixements nous que poden ser difícils de digerir si no es té prou temps.

1.2 Objectius del TFC

La voluntat d'aquest projecte es fer una petita aplicació que permeti ampliar les funcionalitats típiques que es poden trobar a molts videoclubs (almenys els que jo conec). Partim d'un usuari típic que s'ha donat d'alta en un videoclub d'una zona que li interessa (per proximitat, tipus de pel·lícules ..). Aquest usuari disposa d'una targeta que l'identifica, amb un codi d'usuari únic i un password que pot canviar. Amb la targeta va al seu videoclub i pot recarregar la targeta, tornar un préstec, o agafar-ne un préstec nou (dvd o videojoc).

Amb aquest model tenim la desventaja que l'usuari s'ha de desplaçar físicament al videoclub per veure si hi ha alguna pel·lícula que li agrada i a més que estigui disponible, amb la consegüent pèrdua de temps en el cas que no es doni alguna de les dos condicions.

La idea d'aquest projecte es afegir una sèrie de funcionalitats web que permetin veure a l'usuari les pel·lícules disponibles i fer la reserva per assegurar-se que no es faci el viatge per res. Un cop feta la reserva, si es possible tenint en compte el saldo de que disposa (altra informació útil), tindria un temps determinat (p.ex 1h) per agafar el préstec. En cas d e que no anés a buscar-lo es penalitzaria amb 1€ (per exemple) i l'estat de l'article passaria a disponible.

No s'observa la possibilitat de que amb una targeta es puguin fer préstecs amb diferents videoclubs, es a dir una targeta representa un nom d'usuari i una contrasenya per a un videoclub determinat.

S'inclou també la possibilitat de que l'usuari pugui canviar les seves dades de contacte i la seva contrasenya, però no el seu nom d'usuari (que serà únic per al videoclub).

També es preveu un accés remot per al manteniment de les pel·lícules i dels usuaris. A aquest perfil només hi pot accedir l'administrador del videoclub. Des d'aquest perfil podem modificar les pel·lícules (sinopsi, caràtula), afegir-ne de noves o eliminar-ne (si no es troben reservades i/o alquilades). També podem consultar i modificar les dades dels usuaris, afegir i suprimir usuaris, consultar el seu saldo i modificar-lo i consultar l'històric de préstecs i reserves .

No es preveu implementar la part pròpiament de préstec (la que es trobaria en la tenda i que està en tots els videoclubs) doncs hauríem de tenir en compte els dispositius electrònics que permeten l'accés amb targeta, la entrada de diners i la entrada/sortida de articles. Encara que es suposa que l'usuari hauria de poder treure els articles reservats.

L'aplicació estarà configurada en un servidor d'aplicacions que es troba físicament al videoclub juntament amb la base de dades. Així doncs cada videoclub tindrà una base de dades única. A aquesta aplicació s'accedirà des de la plana web de la cadena de videoclubs seleccionant el videoclub en qüestió.

Aquest servidor d'aplicacions seguirà l'estàndard J2EE, utilitzant per a tal efecte el servidor Tomcat (Web Container), per ser tots dos de codi obert i per tant gratuïts. Per a la base de dades s'utilitzarà MySQL 5.0 (per les mateixes raons). Per implementar l'aplicació web es seguirà el patró MVC Struts i per el mapeig amb la base de dades, Hibernate, per ser dos de les tecnologies Java amb més projecció i suport en la actualitat i que permeten una reutilització i manteniment òptims.

1.3 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 es 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.

1.4 Planificació del projecte

A l'hora de plantejar la planificació es va tindre en compte el factor d'incertesa que implica la gran quantitat de coneixements necessaris per la consecució del projecte. Això fa molt difícil una bona planificació doncs no se sap com s'assimilaran el nous coneixements.

Es per això que es va fer una planificació molt generalista sense entrar molt en detall.

La següent taula il·lustra la planificació:

<i>Temporització</i>			<i>Corba Aprenentatge J2EE</i>
1ª Setmana (19/03)	Especificació	Funcionalitats, Concretar subsistemes	Instal·lació programari
2ª Setmana (26/03)		Casos d'us	
3ª Setmana (2/04)	Disseny	Diagrames de classes	JBoss
4ª Setmana (9/04)		Diagrames de col·laboració	
5ª Setmana (16/04)		Interfície gràfica	JSP
6ª Setmana (23/04)	Implementació		Struts Hiberna te
7ª Setmana (30/04)			
8ª Setmana (7/05)			
9ª Setmana (14/05)			
10ª Setmana (21/05)			
11ª Setmana (28/05)			
12ª Setmana (4/06)	Revisió i presentació	Redacció de la memòria i presentació.	
13ª Setmana (11/06)			

1.5 Productes obtinguts

El producte final d'aquest projecte consisteix en:

- Fitxer war amb l'aplicació mes els fonts. Aquest fitxer esta llest per instal·lar-se en un servidor d'aplicacions compatible amb J2EE.
- Codi font de totes les classes utilitzades
- Fitxers de creació de les taules de la BD mes les dades necessàries

1.6 Descripció de la resta de capítols de la memòria

En els següents capítols es comentaran aspectes sobre l'arquitectura utilitzada, anàlisi de requeriments i disseny.

2 Arquitectura de l'aplicació

2.1 Introducció

Com s'ha comentat en la introducció l'aplicació utilitza J2EE , dins aquesta arquitectura utilitzem el patró MVC mitjançant el framework Struts. Dins d'aquest patró utilitzarem Tiles per gestionar les vistes JSP. Per la persistència farem servir Hibernate. Per últim farem servir Quartz per a realitzar tasques que impliquin una temporització. També cal mencionar la important labor de Javascript en la validació dels formularis.

2.2 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 responant 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.

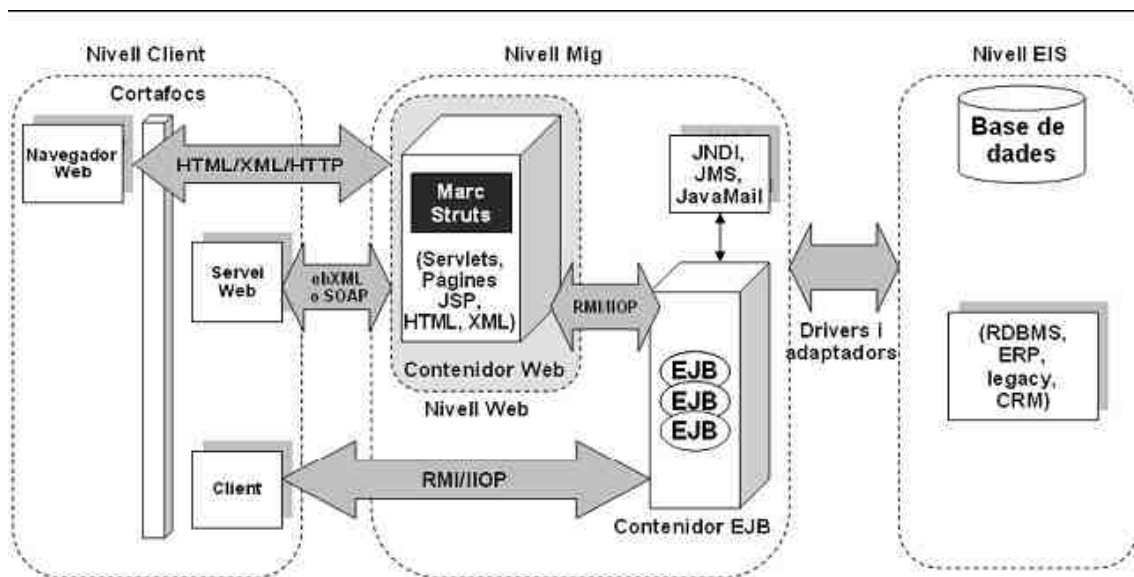
Per poder donar tota aquesta funcionalitat, la plataforma J2EE dona un conjunt d'APIs de java i serveis necessaris per al suport d'aplicacions per empreses.

La

plataforma complerta 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 complert.

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 interactua 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 interactui 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.

2.3 Patró MVC-Struts

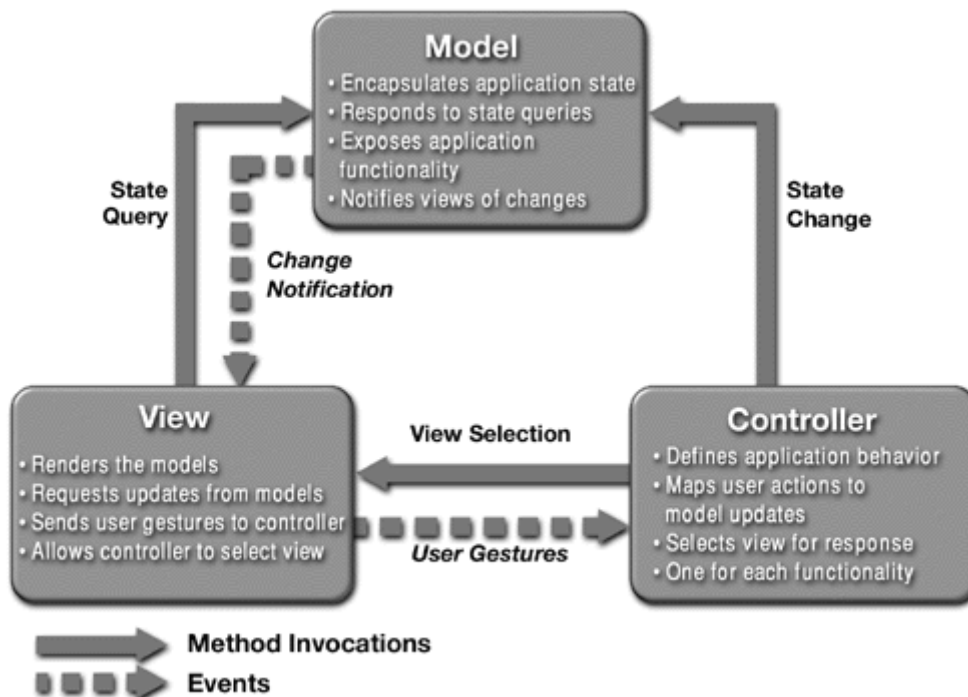
L'arquitectura de la nostra aplicació es basa en l'estructura MVC ("Model-View-Controller"). És un paradigma utilitzat en diversos desenvolupament de software. Gràcies a aquest "Framework" s'aconsegueix una divisió de les diferents parts que conformen una aplicació amb la una de les seves principals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

La següent figura mostra la interacció entre les diferents parts.



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 diseccionar 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 fitxers 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.

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'estan desenvolupant noves versions.

2.3.1 Us de Tiles

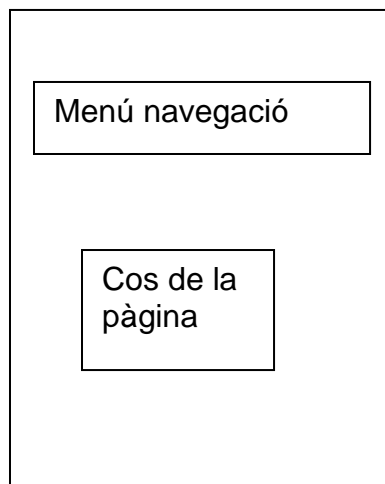
Dintre del marc de treball de Struts (tot i que actualment existeixen com a projectes separats) tenim la opció d'utilitzar tiles. La seva principal funció es la reutilització de codi jsp

Tot i que la tecnologia JSP disposa d'una eina per la reutilització a través dels *includes* tiles va un pas més lluny seguint el patró "Composite View design pattern".

Tiles permet definir layouts ("plantilles") per aprofitar tot el que es manté constant en una pàgina jsp. A través d'un fitxer (tiles-defs.xml) es munta la plantilla a partir de JSPs.

En el nostre projecte s'han fet servir dos plantilles pràcticament idèntiques per utilitzar en la part de administració i usuari respectivament.

A continuació es mostra un esquema de la plantilla:



L'ús de tiles ha sigut molt profitós, doncs a permet que el menú de navegació es pugui generar ficant els atributs necessaris en les definicions del fitxer tiles-defs.xml.

2.4 Validacions

El tema de las validacions es te molt en compte en Struts de manera que de forma senzilla es pot sobreesciure el mètode validate de ActionForm perque validi el formulari i retorni un ActionError que vagi a parar a l'etiqueta <html:errors /> del JSP.

Struts va un pas mes enllà i inclou el framework Validator per optimitzar encara mes el procés de validació. No ha fet falta en l'aplicació utilitzar aquest framework i s'ha optat per fer la gran majoria de validacions en el costat del client utilitzant funcions javascript. Només en uns pocs casos (com en la validació de imatges) s'ha fet servir el mètode validate de ActionForm.

Personalment crec que es millor fer les validacions de format dels formularis en el costat del client fent servir Javascript. En el nostre cas s'ha utilitzat una llibreria de Javascript creada específicament per solventar aquests problemes tant comuns de forma fàcil i sencilla. Aquesta llibreria es fvalidate i inclou un nombre de funcionalitats mot gran per tot tipus de validacions, p.ex:

- Longitud mínima i màxima d'un camp
- Valida si els formats numèrics s'han entrat correctament
- Valida que dos camps del formulari siguin iguals (molt útil pel tema dels passwords).
- Valida que no et deixis camps buits
- Etc..

2.5 Persistència

Avui en dia la programació orientada a objectes a calat en tot el mon de la programació de forma que qualsevol projecte mitjanament gran es fa seguint aquest paradigma.

No obstant les bases de dades continuen, com fa 20 anys, seguint el model relacional. Aquesta diferència tant gran de concepte fa que normalment la part d'accés a base de dades sigui una mica tediós i hagi que fer un esforç continu de traducció de objectes a taules i de taules a objectes.

Una solució a aquest problema es utilitzar un ORM (Object Relational Mapping) com és el cas de Hibernate, que es l'ORM que s'ha utilitzat en aquest projecte.

L'objectiu d'un ORM es pogué interactuar amb la base de dades com si fos orientada a objectes de forma que en lloc de fer `INSERT INTO USUARIS VALUES` poguem fer `session.save(usuari)`.

Evidentment aquesta comoditat te un preu, hi ha que fer el mapeig de taules (i les seves relacions) a objectes. En hibernate aixó es fa en els fixers `hbm.xml`.

Entendre els conceptes de mapejos no es trivial i requereix un temps d'aprenentatge. Un cop fet els mappings cal configurar un fitxer central de configuració (`hibernate-cfg.xml`) on entre d'altre questions s'especifica la connexió a la base de dades (driver JDBC) i el dialecte a utilitzar amb la base de dades.

Hibernate suporta pràcticament la totalitat de Bases de dades del mercat. En el nostre cas hem utilitzat el dialecte MySQL per taules MyISAM (`MySQLMyISAMDialect`) doncs els altres dialectes de MySQL donaven problemes a l'hora d'utilitzar camps de tipus TEXT.

2.6 Programació de tasques, utilització de Quartz

De bon principi ens vam trobar amb un problema que consistia en que calia realitzar una tasca de forma periòdica. En un principi es va plantejar la

utilització de triggers de la base de dades, però de seguida ens vam decantar per l'us d'una llibreria molt potent per programar tasques.

Aquesta tasca consisteix en l'execució del mètode `GestorReserves.modificaTempsReserves(int)` que el que fa es disminuir el temps d'una reserva de 5 en 5 minuts i actuar en conseqüència si el temps s'esgota.

Aquesta api es de lliure distribució i es àmpliament utilitzada per la comunitat java. Es pot trobar a <http://www.opensymphony.com/quartz/> i respon al nom de quartz.

Per fer que la tasca s'executes al arrancar el Tomcat es va crear un sevlet `VideoclubSchedulingServlet.java` i es va configurar el fixer `web.xml` perquè s'executes en segon lloc (en primer s'executa l'`ActionServlet`).

3 Especificació i Anàlisi de requeriments

3.1 *Introducció*

En aquest apartat s'especifiquen i analitzen tots els requisits que necessitem per realitzar l'aplicació videoclub

3.2 *Descripció del projecte*

El projecte consisteix en una aplicació web perquè l'usuari pugui realitzar la reserva de articles (Dvd's i jocs de Play Station 2 però fàcilment ampliable a altres). També hi ha un apartat d'administració només accessible per l'administrador del videoclub des d'on es podran gestionar els usuaris i articles.

Així doncs tenim 2 parts clarament diferenciades:

Visió de l'usuari

L'usuari es validarà i tindrà accés a un menú senzill des d'on podrà veure les seves dades personals i navegar per veure els diferents articles del videoclub. Aquesta validació es farà ficant el seu DNI i un password secret proporcionat per l'administrador del videoclub.

Aquesta navegació ha de ser lo mes intuïtiva i senzilla possible, de manera que s'ha optat per que en primera instància l'usuari seleccioni el tipus d'article que vol reservar perquè en una posterior etapa pugui seleccionar els articles per genere, categoria o fer una cerca per títol. Un cop fet això l'usuari es trobarà amb els articles que compleixin els criteris de selecció.

Els articles es troben ordenats alfabèticament per facilitar la localització dels mateixos, a més clicant en la imatge s'obrirà una finestra amb informació del article (nom, sinopsi, actors,.....,preu de préstec).

La visualització de l'estat de l'article ha de se ràpida amb l'ajuda de colors vius. Si l'estat de un article es DISPONIBLE es podrà procedir a la seva reserva. L'aplicació només ens deixarà afegir al nostre carrito de reserves un màxim de 5 articles per sessió doncs es norma del videoclub que només es puguin tindre en reserva i/o préstec 5 articles com a màxim.

Durant la sessió, l'usuari pot veure en tot moment com esta el seu carrito de reserves i pot eliminar-ne articles si vol. Un cop es decideix a formalitzar la reserva pulsarà un boto de confirmació i se li informarà de l'èxit de l'operació.

L'usuari en tot moment visualitzarà el seu saldo en la part inferior dreta de la barra de títol de l'aplicació. També tindrà disponible un link per desconnectar-se de la sessió.

També es possible que l'usuari pugui visualitzar les seves dades personals de contacte.

Visio de l'administrador

L'administrador es validarà en la mateixa finestra que l'usuari, però en el seu entorn tindrà una sèrie de funcionalitats administratives que bàsicament es poden agrupar en:

- Administració d'usuaris
 - Afegir usuari nou
 - Cercar usuaris per posteriorment modificar-los si escau

- Administració d'articles
 - Afegir articles nou
 - Cercar articles per posteriorment modificar-los si escau

A part d'aquestes funcionalitats administratives l'administrador podrà canviar el seu password d'accés.

A l'hora de insertar un usuari o un article (o modificar-los) hi hauran una sèrie de dades que seran obligatòries i a més tindran que ser introduïdes en un format concret (p.ex un telèfon ha de ser tots números). En cas d'un error s'informarà a l'administrador d'aquest fet mitjançant l'ús de funcions javascript.

Per l'eliminació d'articles o usuaris s'ha optat per l'opció de donar-los de baixa, doncs sempre es factible que es trobin a l'aplicació per motius estadístics.

Les cerques d'usuari es podran fer per dni o per nom d'usuari, mentre que les cerques d'articles només es podran portar a terme pel títol del mateix.

3.3 *Requeriments funcionals*

L'aplicació es pot dividir en diferents subsistemes:

- **Subsistema de gestió d'usuaris:** Aquest subsistema s'ocupa de les altes i baixes d'usuaris així com la modificació de les dades dels usuaris. També s'ocupa de validar als usuaris dins de l'aplicació
- **Subsistema de gestió d'articles:** Aquest subsistema s'ocupa de les altes i baixes d'articles així com la modificació de les dades dels articles.
- **Subsistema de gestió de reserves:** Aquest subsistema s'ocupa de la gestió de les reserves dels usuaris.

3.4 *Especificacions de les funcionalitats per subsistema*

3.4.1 Subsistema de gestió d'usuaris

A aquest subsistema podrà accedir l'administrador un cop s'hagi validat

Cas d'us: Login

Resum de la funcionalitat: L'administrador o usuari ha d'omplir un camp USUARI i un altre CONTRASENYA per tal que l'aplicació el validi i el deixi entrar

Actors: Administrador i Usuari

Casos d'us relacionats: Cap

Precondició: L'administrador o usuari esta donat d'alta en la base de dades de l'aplicació

Postcondicio: L'aplicació l'identifica com a administrador o com a usuari de l'aplicació

Descripció:

- L'administrador o usuari tecleja directament una url que li ha proporcionat l'empresa per accedir a una pantalla on troba un formulari amb un camp USUARI i un altre PASSWORD. L'administrador o usuari introdueix el seu codi d'usuari i el seu password:
 - Si el sistema no troba l'administrador amb el password a la base de dades informa del fet a l'usuari.
 - En cas que el sistema validi a l'usuari li dona pas a l'aplicació amb la sessió corresponent com a administrador o usuari (segon el cas).

Cas d'us: Logout

Resum de la funcionalitat: L'adminstrador o usuari surt de la sessió

Actors: Administrador i Usuari

Casos d'us relacionats: login

Precondicio L'administrador o usuari s'ha validat correctament i esta en una sessió

Postcondicio: L'usuari surt de la sessió

Descripció:

- L'usuari es troba en una sessió
- Prem el botó desconnecta i surt de la sessió
- Apareix en la pantalla inicial d'accés.

Cas d'us: Cerca usuari

Resum de la funcionalitat: L'administrador cerca un usuari a l'aplicació

Actors: Administrador

Casos d'us relacionats: Login

Precondicio: L'administrador s'ha validat correctament i esta en una sessió

Postcondicio: Trobem l'usuari del qual fem la recerca

Descripció:

- L'administrador es troba en una sessió
- L'administrador selecciona l'opció Cercar Usuari
- Li apareix un formulari on pot cercar per nom i cognoms, o per codi (dni) d'usuari
- Un cop introduïdes les dades prem el boto de cerca:
 - En cas de no trobar cap resultat se li informará del fet
 - En cas de trobar coincidències se li mostrarà una llista amb els resultats i continuem.
- L'administrador selecciona un usuari de la llista.
- Li apareix una finestra amb les dades de l'usuari en qüestió.

Cas d'us: Modifica usuari

Resum de la funcionalitat: L'administrador modifica les dades d'un usuari a l'aplicació

Actors: Administrador

Casos d'us relacionats: Cerca usuari

Precondició: L'administrador ha fet una cerca amb resultat i ha seleccionat un usuari per modificar. Ens trobem en la pantalla amb un formulari amb les dades de l'usuari

Postcondició: Hem fet la modificació correctament de les dades de l'usuari

Descripció:

- L'administrador modifica les dades d'un usuari (l'identificador no es editable):
 - En cas que falti alguna dada li sortirà un missatge d'error.
 - En cas que la modificació hagi sigut correcta se li informará del fet

Nota: Dintre de la modificació cal la possibilitat de donar d'alta i/o baixa a l'usuari

Cas d'us: Alta usuari

Resum de la funcionalitat: L'administrador entra un nou usuari a l'aplicació

Actors: Administrador

Casos d'us relacionats: Login

Precondició: L'administrador s'ha validat correctament i esta en una sessió

Postcondició: Hem fet l'alta correctament de l'usuari

Descripció:

- L'administrador es troba en una sessió
- L'administrador selecciona l'opció Alta Usuari

- Li apareix un formulari on te que omplir les dades de l'usuari. Tots els camps son obligatoris
- Un cop introduïdes les dades prem el boto de guardar:
 - En cas de que falti algun camp o estigui mal formatejat (p.ex el telèfon no es numèric, el saldo ha de ser mes gran de 2€, el password mes gran de 6 caracters repetint-lo dos vegades..) se li mostrarà un missatge d'error. Altrament continuem
 - En cas de que el DNI (clau primària) estigui duplicat es mostrarà un missatge d'error. Altrament continuem.
 - Es mostra un missatge informant de l'èxit de l'operació.

3.4.2 Subsistema de gestió d'articles

Abans de continuar cal deixar clar la diferència entre un exemplar d'un article i un article pròpiament dit, doncs son dos conceptes que es solen barrejar en funció del context. Un article pot ser un dvd o un jocPS2 però pot tindre molts exemplars (es normal veure en el videoclub que hi han 5 exemplars d'una mateixa pel·lícula, normalment un "block-buster". Posteriorment veurem en el diagrama de classes mes clara aquesta diferència.

De moment ens interessa que al modificar un exemplar d'un article estem modificant característiques intrínseques d'aquell exemplar (com poden ser l'estat: DISPONIBLE, RESERVA, o PRESTEC; o si esta o no de baixa). En canvi al modificar un article (recordem dvd o joc de ps2) estem modificant característiques comunes a un conjunt d'exemplars, com poden ser el nom o el preu de l'article.

Cas d'us: Cerca exemplar article

Resum de la funcionalitat: L'administrador cerca un exemplar d'un article a l'aplicació

Actors: Administrador

Casos d'us relacionats: Login

Precondicio: L'administrador s'ha validat correctament i esta en una sessió

Postcondicio: Trobem l'exemplar de l'article del qual fem la recerca

Descripció:

- L'administrador es troba en una sessió
- L'administrador selecciona l'opció Cercar Article
- Li apareix un formulari on pot cercar per títol d'article
- Un cop introduïdes les dades prem el boto de cerca:
 - En cas de no trobar cap resultat se li informarà del fet
 - En cas de trobar coincidències se li mostrarà una llista amb els resultats i continuem.
- L'administrador selecciona un exemplar de la llista.
- Li apareix una finestra amb les dades de l'exemplar de l'article en qüestió entre les quals es troba un link a l'article en qüestió.

Cas d'us: Cerca article

Resum de la funcionalitat: L'administrador cerca un article a l'aplicació

Actors: Administrador

Casos d'us relacionats: Cercar exemplar_article

Precondicio: L'administrador ha a cercat satisfactòriament un exemplar d'un article

Postcondicio: Trobem l'article del qual fem la recerca

Descripció:

- L'administrador prem el link de l'exemplar de l'article que el porta a les dades de l'article en qüestió

Cas d'us: Alta article

Resum de la funcionalitat: L'administrador entra un nou article a l'aplicació amb un número d'exemplars (com a mínim 1 i com a màxim 8)

Actors: Administrador

Casos d'us relacionats: Login

Precondicio: L'administrador s'ha validat correctament i esta en una sessió

Postcondicio: Hem fet l'alta correctament de l'article

Descripció:

- L'administrador es troba en una sessió
- L'administrador selecciona l'opció Alta Article
- Li apareix un formulari on te que omplir les dades de l'article. No tots els camps son obligatoris. Es important ficar el número d'items de l'article, tipus (dvd, joc), genere, ..., les imatges
- Un cop introduïdes les dades prem el boto de guardar:
 - En cas de que falti algun camp o estigui mal formatejat (p.ex la imatge no es jpg o esta buida) se li mostrarà un missatge d'error. Altrament continuem
 - Es mostra un missatge informant de l'èxit de l'operació.

Cas d'us: Modifica exemplar article

Resum de la funcionalitat: L'administrador modifica les dades d'un article a l'aplicació

Actors: Administrador

Casos d'us relacionats: Cerca exemplar article

Precondicio: L'administrador ha fet una cerca amb resultat i ha seleccionat un exemplar d'un article per modificar. Ens trobem en la pantalla amb un formulari amb les dades de l'exemplar de l'article

Postcondicio: Hem fet la modificació correctament de les dades de l'article

Descripció:

- L'administrador modifica les dades d'un exemplar d'un article (l'identificador no es editable ni l'estat tampoc).
 - Només es pot modificar la baixa d'un article. Es pot donar d'alta o de baixa. L'estat (DISPONIBLE, RESERVAT, PRESTEC) no es editable.

[Cas d'us: Modifica article](#)

Resum de la funcionalitat: L'administrador modifica les dades d'un article a l'aplicació

Actors: Administrador

Casos d'us relacionats: Cerca article

Precondicio: L'administrador ha fet una cerca amb resultat i ha seleccionat un exemplar d'article. Un cop en la pantalla de modificació de l'exemplar, tenim un link que ens porta a una pantalla amb un formulari amb les dades de l'article

Postcondició: Hem fet la modificació correctament de les dades de l'article

Descripció:

- L'administrador modifica les dades d'un article (l'identificador no es editable).
 - En cas que falti alguna dada o estigui mal formatejada li sortirà un missatge d'error.
 - En cas que la modificació hagi sigut correcta se li informará del fet

3.4.3 Subsistema de gestió de reserves

Aquest es un subsistema molt important doncs es on el client farà les reserves i les confirmarà. Com hem dit anteriorment el client disposa d'un carrito que pot fer servir per acumular reserves (5 com a molt) durant la sessió. Un cop s'ha decidit a formalitzar les reserves aquests exemplars d'article passaran a estat RESERVAT a l'espera que el client vagi al videoclub físicament a buscar els articles, moment en el que l'estat de l'article passarà a PRESTEC.

Aclarim aquí que el carrito es un objecte que es guarda en sessió que guarda els articles que l'usuari va afegint.

Cas d'us: Cerca exemplar article

Resum de la funcionalitat: L'usuari cerca un article a l'aplicació

Actors: Usuari

Casos d'us relacionats: Login

Precondició: L'usuari s'ha validat correctament a l'aplicació

Postcondició: Trobem l'article del qual fem la recerca

Descripció:

- L'usuari navega pel menú i decideix fer la cerca per a un tipus d'article (DVD o JOC de PS2) segons un dels següents criteris: PER GÈNERE, PER CATEGORIA o PER TÍTOL.
- Si es decideix per GÈNERE segons sigui un DVD o JOC de PS2 tenim que :
 - DVD. Podem tenir com a gènere:
 - Acció
 - Drama
 - Suspens
 - Ficció
 - Comèdia
 - Terror
 - Infantil
 - Porno
 - JOC PS2. Podem tenir com a gènere:
 - Acció
 - Aventura
 - Esports
 - Lluita
 - Rol
 - Estratègia
- Si es decideix per CATEGORIA tant per DVD com per JOC de PS2 tenim com a categoria:
 - Novetats
 - Supernovetats
- Si es decideix per Títol tant per DVD com per JOC de PS2 tenim que apareixerà un formulari de cerca per introduir la paraula de cerca.
- En qualsevol dels tres casos anteriors l'usuari es trobarà en una pantalla amb els resultats de la cerca on podrà visualitzar l'estat

ràpidament dels exemplars d'article. A més clickant a damunt de les imatges se li obrirà una finestra amb informació de l'article.

Cas d'us: Reserva exemplar article

Resum de la funcionalitat: L'usuari reserva un exemplar d'un article i l'afegueix al carrito

Actors: Usuari

Casos d'us relacionats: Cerca exemplar article

Precondicio: L'usuari ha fet una cerca i disposa d'un llistat amb tots els exemplars d'article que compleixen els criteris de cerca

Postcondicio: Hem reservat un dels articles del llistat

Descripció:

- L'usuari selecciona un dels articles i prem el botó de Reservar.
 - En cas de que la suma dels articles que estan al carrito mes els que esta afegint sumi mes de 5 o que s'afegeixi el mateix exemplar d'article (no article) mes d'un cop sortirà un missatge d'error.
 - Altrament sortirà un missatge informatiu de l'operació a més d'un link per veure els articles al carrito.

Cas d'us: Confirma reserva exemplar article

Resum de la funcionalitat: L'usuari confirma les reserves que te al carrito

Actors: Usuari

Casos d'us relacionats: Reserva exemplar article

Precondició: L'usuari ha fet una o mes reserves d'articles

Postcondició: Hem confirmat les reserves a la base de dades. De forma que podem passar a recollir els articles al nostre videoclub.

Descripció:

- L'usuari, un cop a fet la reserva d'un article prem el link per anar a veure les reserves, o fa servir el link que es troba en el títol de la pàgina.
- Apareix un llistat on es pot veure informació relativa als articles pendents de reserva. D'aquest llistat es poden treure articles o veure informació relativa al article
- L'usuari prem el botó Confirma per confirmar les reserves:
 - Hi han varies possibilitats d'error, en qualsevol cas se li informa a l'usuari d'aquest fet:
 - Pot ser que el carrito estigui buit
 - L'usuari s'hagi donat de baixa en el transcurs de la sessio
 - L'usuari tingui un saldo negatiu
 - Els articles que l'usuari te en préstec i/o reserva mes els del carrito sumen mes de 5.
 - Que durant la sessio de l'usuari un article hagi canviat d'estat (passi de DISPONIBLE a RESERVAT o PRESTEC) o fins i tot s'hagi donat de baixa
 - En cas de que tot hagi anat be se li informa a l'usuari que te 1 hora per anar a buscar els articles i en cas de no fer-ho serà multat amb un euro per article.

Cas d'us: Veure dades personals

Resum de la funcionalitat: L'usuari veu les seves dades personals però o pot editarles

Actors: Usuari

Casos d'us relacionats: Login

Precondició L'usuari s'ha validat correctament a l'aplicació

Postcondició: L'usuari ha visualitzat les seves dades personals

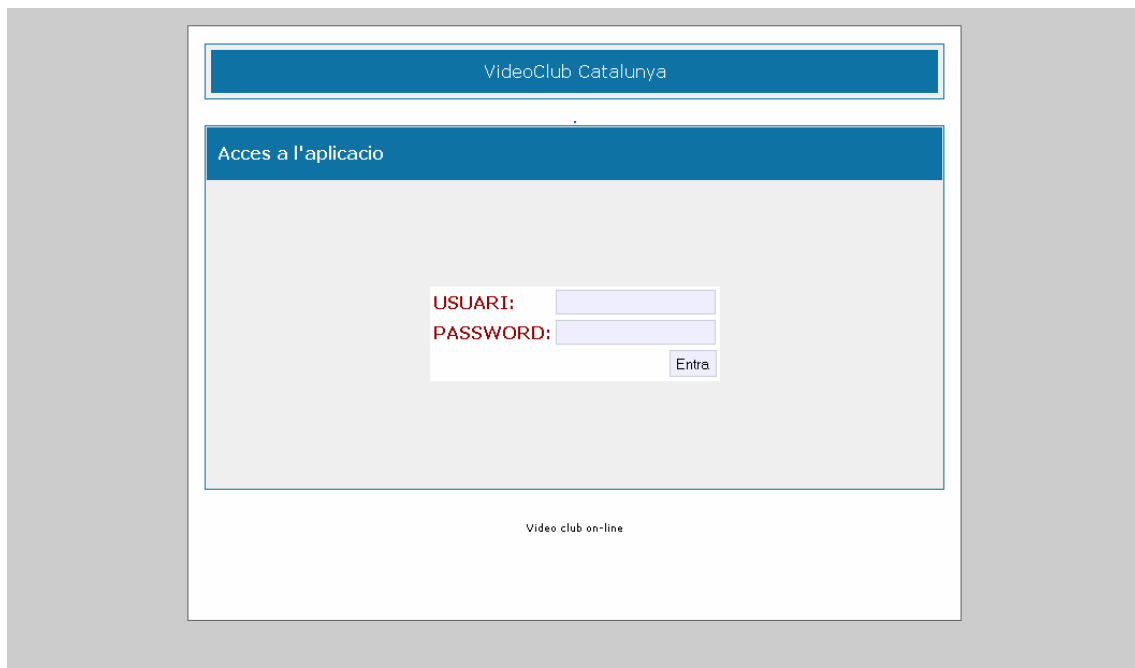
Descripció:

- L'usuari es troba a la pantalla inicial un cop s'ha autenticat
- L'usuari prem el boto veure dades
- Es mostren les seves dades de contacte.

3.5 Interfície gràfica

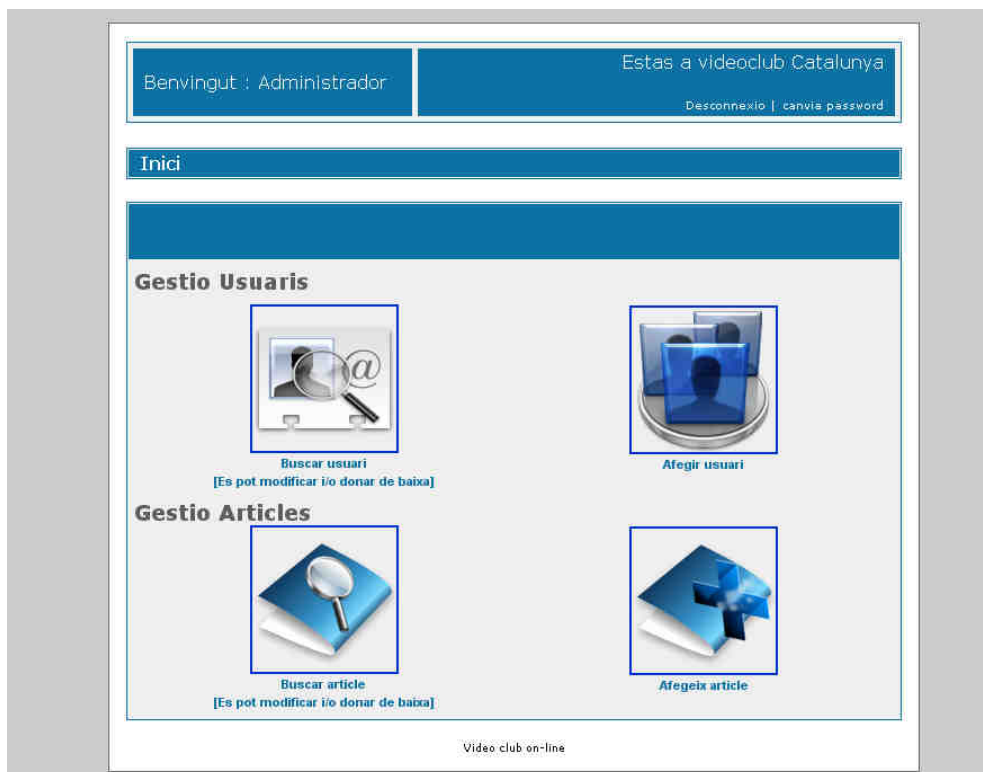
La interfície gràfica es una part molt important de la nostra aplicació, doncs es el que els usuaris veuran. S'han utilitzat com a colors principals uns tons blavosos que ajuden a la visibilitat. S'ha procurat que la navegació fos lo mes intuïtiva possible, per a tal efecte s'ha utilitzat un menú vertical sota del títol de manera que l'usuari sap en tot moment on es troba.

La pantalla de login es igual tant per als usuaris com per a l'administrador:



3.5.1 Flux de pantalles Administració

Un cop validat l'administrador es troba amb un menú general:



Gestio Usuaris

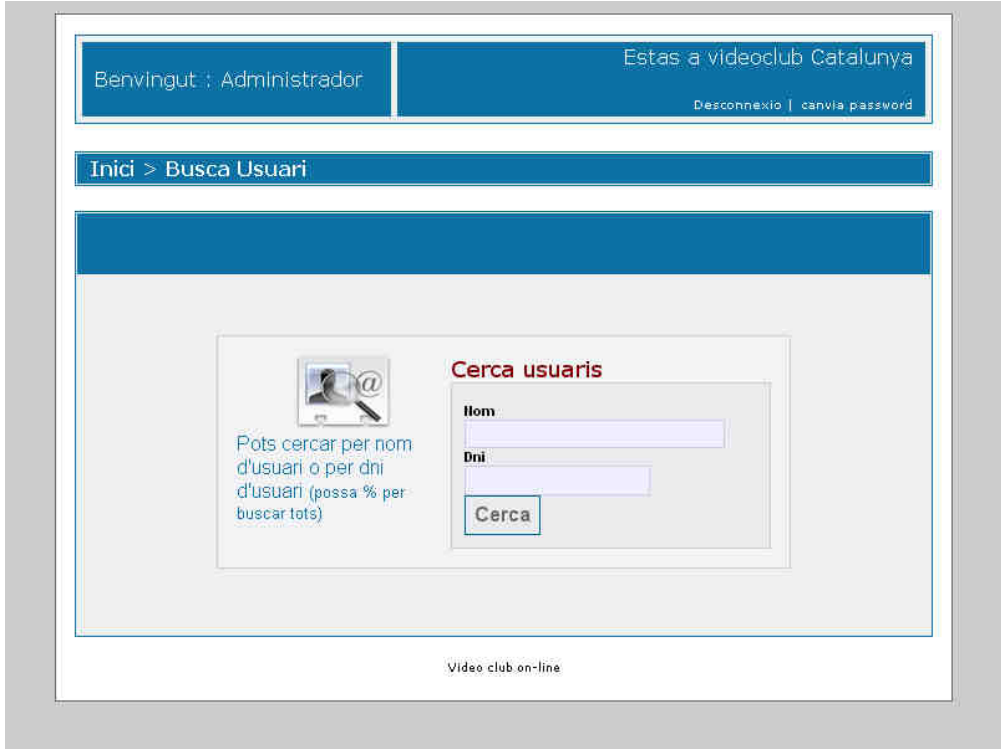
Si ens decidim per afegir un usuari:

The screenshot shows the 'Afegeix Usuari' form. The breadcrumb is 'Inici > Afegeix Usuari'. The form fields are as follows:

NOM:	<input type="text"/>
DNI:	<input type="text"/>
CIUTAT:	<input type="text"/>
DIRECCIO:	<input type="text"/>
TELEFON:	<input type="text"/>
EMAIL:	<input type="text"/>
PASSWORD: (minim 6 caracters)	<input type="password"/>
CONFIRMA PASSWORD:	<input type="password"/>
SALDO INICIAL :	<input type="text" value="0.0"/> <small>Format XX.XX Valors de 10 a 120</small>

At the bottom of the form are two buttons: 'Guarda' and 'Borra'. At the bottom center of the page, it says 'Video club on-line'.

Si volem fer una cerca d'usuari:




The screenshot shows a web application interface for user search. At the top, there is a blue header bar with two sections: on the left, it says "Benvingut : Administrador" (Welcome: Administrator); on the right, it says "Estas a videoclub Catalunya" (You are in videoclub Catalunya) with links for "Desconnexio" (Logout) and "canvia password" (Change password). Below the header is a blue navigation bar with the text "Inici > Busca Usuari" (Home > Search User). The main content area has a light gray background and contains a search form titled "Cerca usuaris" (Search users). To the left of the form is an icon of a person with a magnifying glass and an '@' symbol, with the text: "Pots cercar per nom d'usuari o per dni d'usuari (possa % per buscar tots)" (You can search by user name or user dni (use % to search all)). The search form itself has two input fields: "Nom" (Name) and "Dni" (DNI), both with light blue text and a light blue border. Below the fields is a "Cerca" (Search) button. At the bottom of the page, there is a small footer that says "Video club on-line".

On els resultats es mostren en forma de llista:

Benvingut : Administrador
Estas a videoclub Catalunya
Desconnexio | canvia password

[Inici](#) > [Busca Usuari](#)



Cerca usuaris

Pots cercar per nom d'usuari o per dni d'usuari (possa % per buscar tots).

#	Nom	DNI	Estat	Saldo
1	Enrique	11111111	ALTA	27
2	Ernesto Cabrera	11111118	ALTA	26
3	Godofredo Garcia	11111113	ALTA	25

Video club on-line

Clicant damunt del nom arribem a la pantalla de modificacio:

Benvingut : Administrador
Estas a videoclub Catalunya
Desconnexio | canvia password

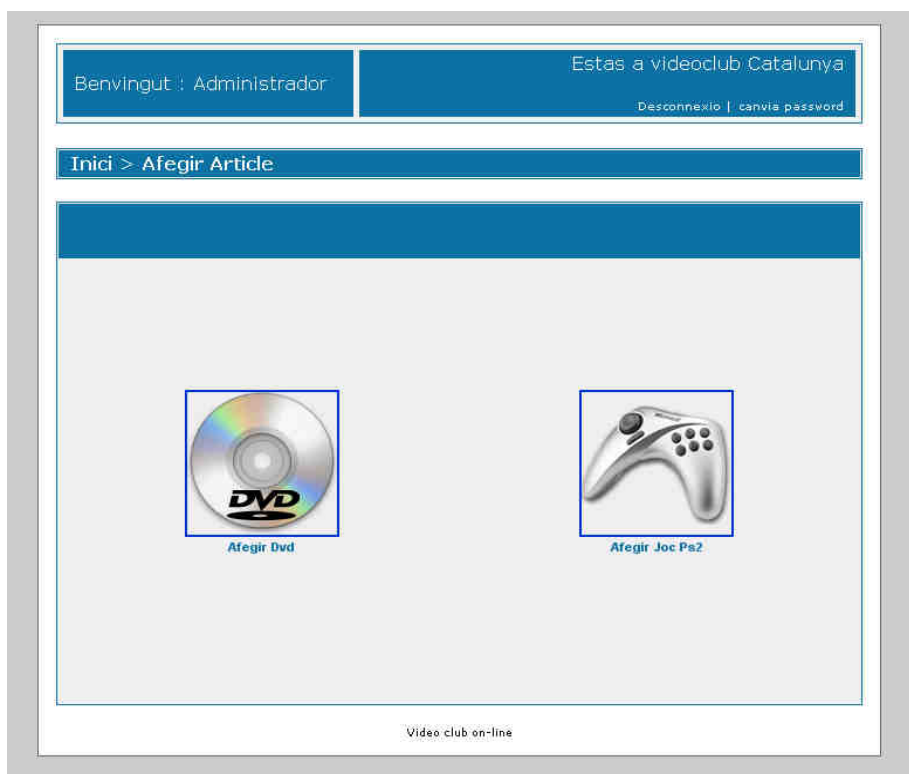
[Inici](#) > [Busca Usuari](#)

Identificador	<input type="text" value="1"/>
NOM:	<input type="text" value="Enrique"/>
DNI	<input type="text" value="11111111"/>
CIUTAT:	<input type="text" value="Tarragona"/>
DIRECCIO:	<input type="text" value="av diagonal bla bla"/>
TELEFON:	<input type="text" value="977822587"/>
EMAIL:	<input type="text"/>
PASSWORD: <small>(minim 6 caracters)</small>	<input type="password"/> <small>Si no el vols canviar no els plenis!!!!</small>
CONFIRMA PASSWORD:	<input type="password"/>
ESTAT	Alta <input type="button" value="v"/>
SALDO ACTUAL :	<input type="text" value="27.0"/> <small>Format XX.XX Com a maxim 500 €</small>

Video club on-line

Gestió Articles

Si ens decidim per afegir un article hem d'escollir entre dvd i joc :



L'operativa es similar en els dos casos. Es mostra la pantalla d'afegir dvd:

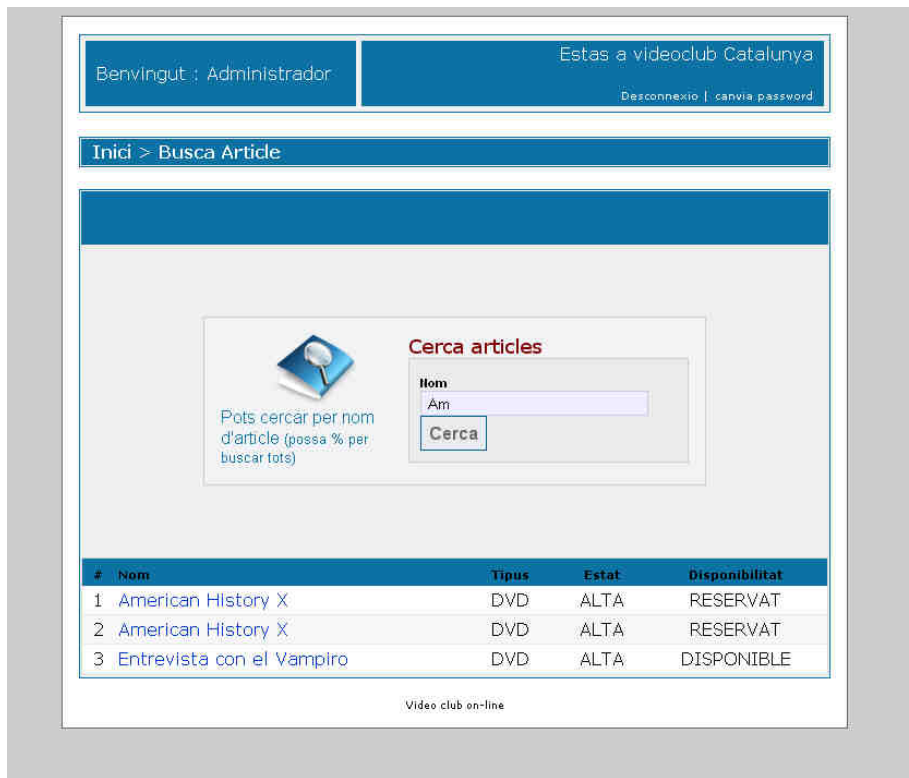
Inici > Afegir Article > Afegir Dvd

NOM:	<input type="text"/>
SINOPSI:	<input type="text"/>
DIRECTOR:	<input type="text"/>
ACTORS:	<input type="text"/>
IDIOMES:	<input type="text"/>
SUBTITOLS:	<input type="text"/>
EDAT:	<input type="text"/>
ANY:	<input type="text"/>
GENERE:	accio <input type="button" value="v"/>
CATEGORIA:	supernovetats <input type="button" value="v"/>
PREU 6 H :	1.6 <small>Format X.XX Valors de 1 a 3</small>
FRACC 1 H :	0.1 <small>Format X.XX Valors de 0 a 1</small>
NUM ITMS:	1 <input type="button" value="v"/>
IMATGE:	<input type="text"/> <input type="button" value="Examinar..."/> <small>Format jpg.</small>
<input type="button" value="Guarda"/>	<input type="button" value="Borra"/>

Video club on-line

Es important el fet de pujar una imatge jpg de qualitat acceptable. Aquesta hauria de tindre una estructura vertical de forma que la relació amplada:alçada fos aproximadament 7:10.

Podem cercar articles per nom:




Com es veu si un exemplar esta mes d'un cop es trobarà el seu nom repetit, tot i que es tracta d'exemplars diferents.

Al clicar en el nom de l'article anem a una pantalla on es troben les dades de l'exemplar:

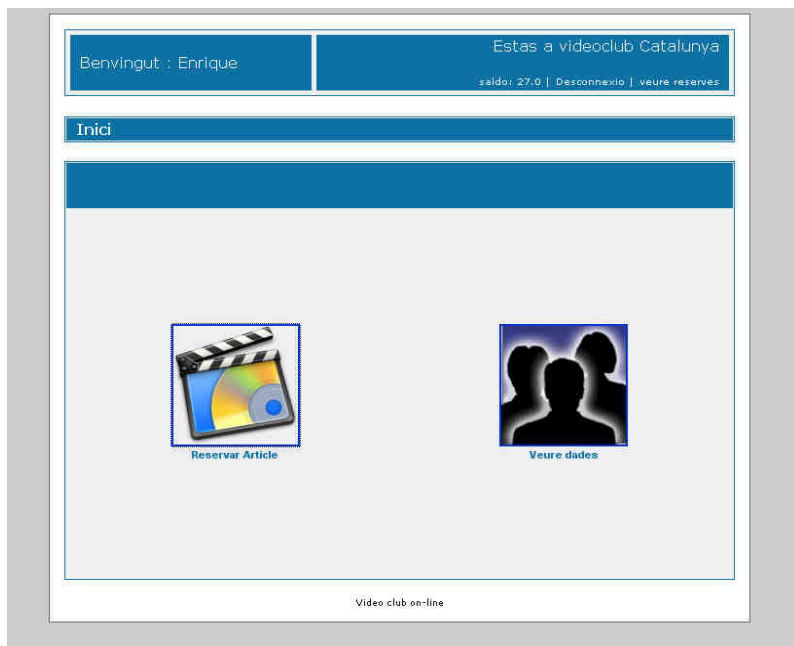


Com s'ha explicat en els casos d'us només es pot modificar la baixa de l'exemplar. Clicant damunt del nom de l'article anem a parar a una pantalla on podem modificar l'article:

ID ARTICLE:	1
NUM ITEMS:	2
NOM:	American History X
SINOPSI:	Tras cumplir condena por el asesinato de dos hombres negros, un joven logra vencer el fanatismo que le había convertido en un ser indeseable, pero no se dará por satisfecho hasta que no domine a ese demonio que también se ha apoderado de su hermano, quien le ve como un héroe e intenta seguir sus pasos.
DIRECTOR:	Tony Kaye
ACTORS:	Edward Norton, Edward Furlong, Beverly D'Angelo
IDIOMES:	Espanyol i Angles
SUBTITOLS:	Espanyol i Angles
EDAT:	> 13 años
ANY:	1998
GENERE:	accio
CATEGORIA:	supernovelats
PREU 6 H :	1.6 <small>Format: 6,XX Valor de 5 a 9</small>
FRACC 1 H :	0.1 <small>Format: 6,XX Valor de 0 a 3</small>
IMATGE:	
NOVA IMATGE:	<input type="text"/> <input type="button" value="Examinar..."/> <small>Format: jpg</small>
	Si no té vols canviar no té pleus!!!!
<input type="button" value="Guarda"/>	<input type="button" value="Tanca"/>

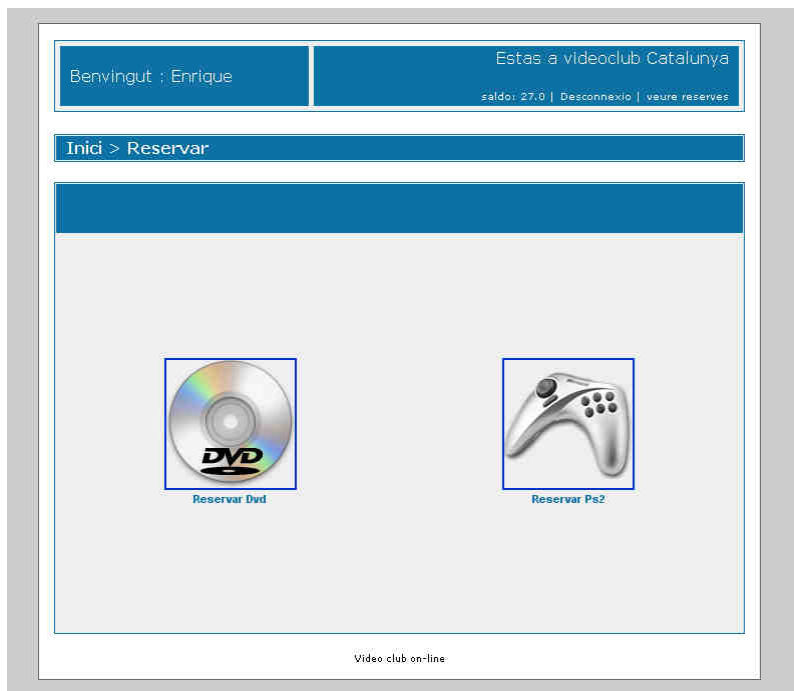
3.5.2 Flux de pantalles Usuaris

Un cop validat l'usuari es troba amb la següent pantalla:

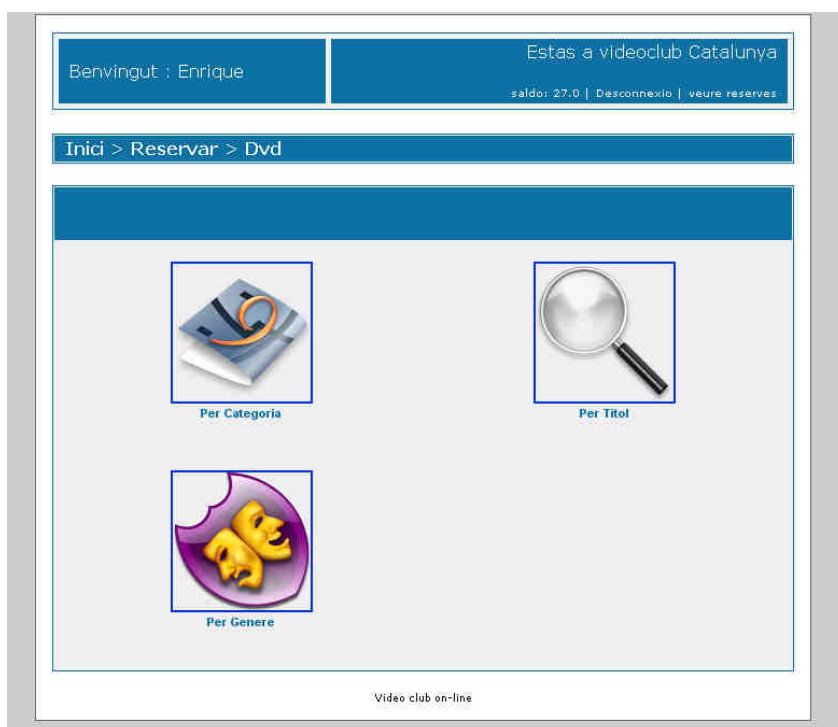


Des d'aquí podem veure les dades, però la opció més interessant es la de reservar article.

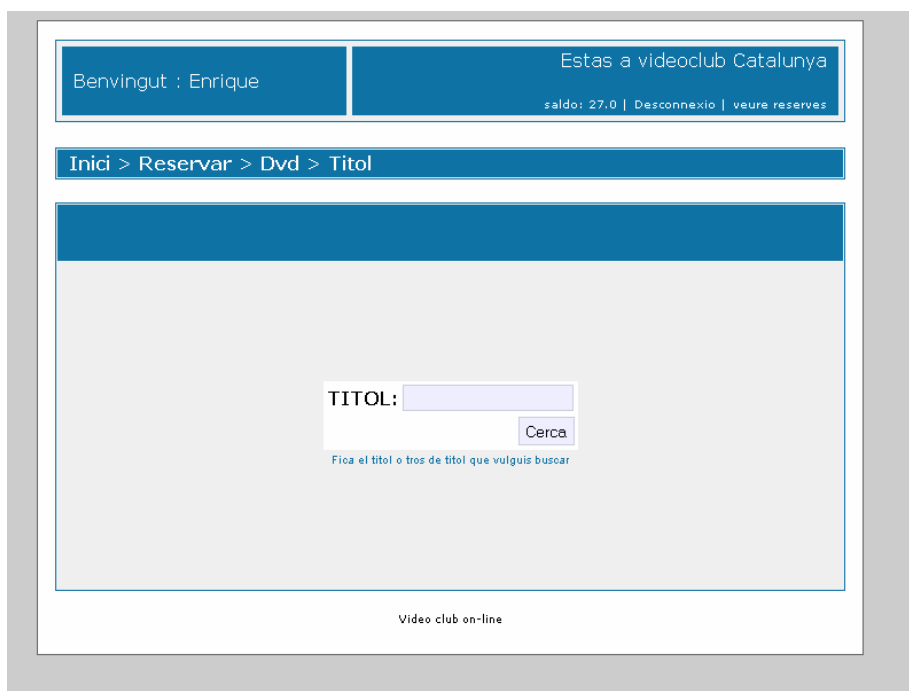
Si optem per aquesta opció, tenim en primera instància la opció d'escollir entre Dvd i Joc de PlayStation:



Seguidament pots escollir per categoria, gènere o cerca per títol:



Tant per categoria com gènere porten a una pàgina per seleccionar una subcategoria o un subgènere per concretar la cerca, mentre que la cerca per títol ens porta a la següent pantalla:

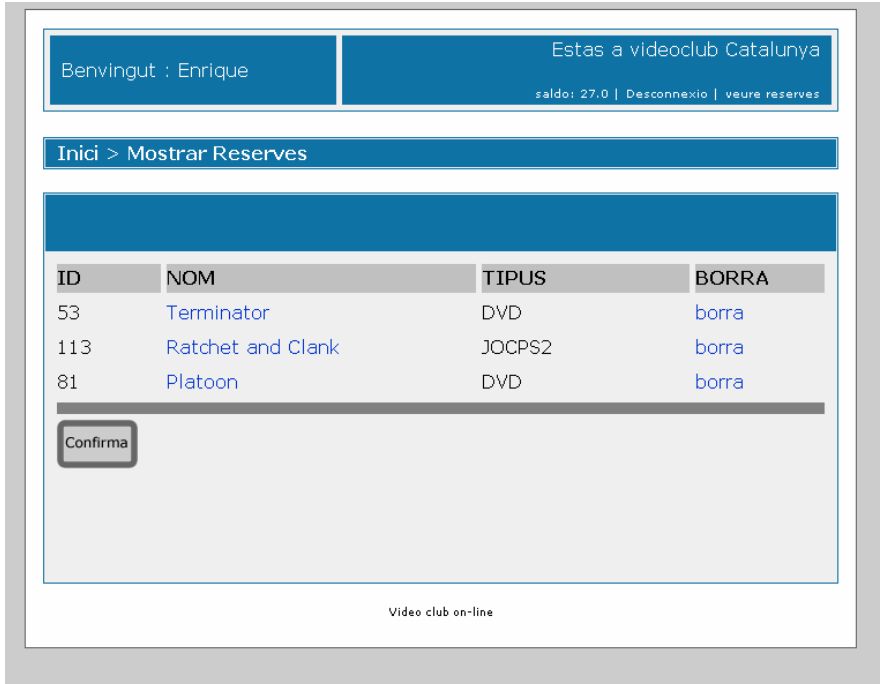


En tots els casos s'arriba al mateix lloc, una pàgina on es mostren llistats els articles ordenats alfabèticament:

The screenshot shows a web interface for a video club. At the top, a blue header bar contains the text "Benvingut : Enrique" on the left and "Estas a videoclub Catalunya" on the right, with a sub-menu "saldo: 27.0 | Desconnexio | veure reserves" below it. Below the header is a navigation bar with "Inici > Reservar". A search bar labeled "Cerca per títol" is positioned above a grid of movie posters. The grid consists of two rows of four posters each. The first row contains: 1. "THE BOURNE IDENTITY EL CASO BOURNE" (RESERVAT), 2. "EL DE MEJOR PELÍCULA" (RESERVAT), 3. "EL DE MEJOR PELÍCULA" (RESERVAT), 4. "EL DE MEJOR PELÍCULA" (RESERVAT). The second row contains: 1. "ENTREVISTA CON EL VAMPIRO" (DISPONIBLE), 2. "HANNIBAL" (DISPONIBLE), 3. "HEAT" (RESERVAT), 4. "HEAT" (RESERVAT). Each poster has a red bar at the bottom with the status "RESERVAT" or "DISPONIBLE" and a yellow bar with a "Reservar" button. At the bottom of the grid is a pagination bar with numbers 1 through 11, where number 3 is highlighted. Below the pagination bar is the text "Video club on-line".

Al clicar damunt de un article surt informació detallada per a l'usuari:

En qualsevol moment l'usuari pot veure el seu carrito fent click a el link veure reserves:



The screenshot shows a web application interface for a video club. At the top, there is a navigation bar with the user's name 'Benvingut : Enrique' and the current page 'Estas a videoclub Catalunya'. Below this, there is a breadcrumb trail 'Inici > Mostrar Reserves'. The main content area features a table with the following data:

ID	NOM	TIPUS	BORRA
53	Terminator	DVD	borra
113	Ratchet and Clank	JOCPS2	borra
81	Platoon	DVD	borra

Below the table, there is a 'Confirma' button. At the bottom of the page, it says 'Video club on-line'.

3.6 *Requeriments de maquinari i programari*

Els requeriments de programari i maquinari es divideixen en dos, segons estiguem en la part del client o del servidor:

- **Requeriments del servidor**

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

- **Servidor d'aplicacions:** En principi qualsevol servidor d'aplicacions es podria utilitzar. El desenvolupament s'ha fet amb Tomcat 5.5 que es de lliure distribució, tot i que qualsevol servidor web que segueixi les especificacions J2EE ens podria servir.
- **Base de dades:** Es fa servir la base de dades de lliure distribució MySQL 5.0. Aquesta base de dades esta perfectament suportada per hibernate

Els requeriments de maquinari per servidor no han de ser massa exigents, doncs ni Tomcat ni MySql son gaire consumidors de recursos, tot i així es recomanable tindre bastant memòria ram (1 Gb o més).

- **Requeriments dels clients**

El client només ha de tindre un navegador web, preferiblement d'última generació (IE 7.0, Mozilla Firefox 2.0), doncs l'aplicació no ha estat testada amb navegadors antics i podria donar problemes

3.7 Seguretat

L'aspecte de la seguretat s'ha tingut en compte a l'hora de controlar l'accés a l'aplicació. Per fer una aplicació mínimament segura s'han utilitzat funcions d'enciptació Hash SHA-1, per tal de guardar els passwords enciptats, de forma que si fins i tot algun hacker te accés a la base de dades, aquest no podria esbrinar el pasword.

Per evitar l'accés indegut a les pagines web s'ha fet un chequeig a nivell de sessió en les planes JSP per veure si la sessió esta correctament inicialitzada. Aprofitant l'ús de TILES es pot aprofitar la plantilla per realitzar aquesta altrament tediosa feina. En cas de que la sessió no estigui correctament inicialitzada es fa un redireccionament a la plana de login.

Es important separar els accessos per als usuaris (clients) del Videoclub del administrador, doncs en cas de que un usuari pogués accedir a l'administració els resultats podrien ser desastrosos...

Encara que aquest nivell de seguretat es fort, encara queda l'opció de protegir les ACTIONS (sobretot les que escriuen en la base de dades) perquè ningú pugui executar-les. Això es fa de la mateixa manera que en els JSP, fent un chequeig de la sessio.

3.8 Possibles funcionalitats addicionals

Les possibilitats d'afegir funcionalitats son molt grans. En part, per falta de temps, no he pogut implementar totes les funcionalitats que m'agradaria, però principalment perquè l'aplicació fos totalment operativa faria falta la part pròpiament de préstec que es trobaria físicament al videoclub (en un terminal tàctil).

Òbviament no es pot implementar aquesta part encara que es podria fer un simulador que implementés totes les funcionalitats del terminal tàctil simulant els fets físics de hardware que queden fora de l'àmbit d'aquest projecte (Ficar diners, Tornar article, Treure article..).

Aquest simulador no comportaria un excés de feina massa gran (però si el suficient per que s'hagi descartat) doncs l'estructura de l'aplicació es la mateixa (mateixa base de dades) i esta preparada perfectament.

A part d'aquest fet les possibilitats de ampliació son molt grans però perfectament afrontables degut a que l'aplicació es perfectament escalable. Algunes serien:

- Fer un mòdul de configuració: Es poden parametritzar molts aspectes de l'aplicació. A continuació es citen alguns:
 - Categories i gèneres dels dvds i jocs. Es podria afegir i modificar categories i gèneres de forma que no sigues un fet fix com ho es ara. Per això caldria un altra taula a la base de dades.
 - La presentació dels resultats de les cerques de l'usuari es presenten en forma de 8 articles per pàgina. Això es podria modificar amb certa facilitat (com sempre aquesta certa facilitat representa moltes hores de feina ...) de forma que l'usuari pogués configurar el número d'articles que vol veure per pàgina.

- Ficar en la pantalla inicial de l'usuari informació d'interès, com els últims articles agafats, novetats
- Afegir la possibilitat de recarregar telemàticament el compte de saldo de l'usuari (aquesta part seria una mica complicada)
- Afegir un mòdul d'estadístiques per treure informació de tot tipus. Aquí les possibilitats també son immenses:
 - Estadístiques de préstec d'un determinat article. Número, duració de les mateixes
 - Estadístiques de consum d'un determinat usuari. Tipus de dvd escollit, preferències....

Per últim cal dir que en un principi es va plantejar fer l'aplicació multiidioma, doncs Struts facilita molt la feina en aquest aspecte, però es va descartar aquesta opció ja que si be els botons i labels son fàcilment parametrizables en el fitxer properties que fa servir struts, els camps de la base de dades no ho son tant i hagués fet falta un rediseny de la base de dades per incloure camps multiidioma. No te sentit fer una aplicació multiidioma si després un usuari no pot veure la sinopsi d'un dvd en el seu idioma. Així doncs fer aquesta aplicació multiidioma hagués complicat molt tot el projecte i es va descartar de bon principi.

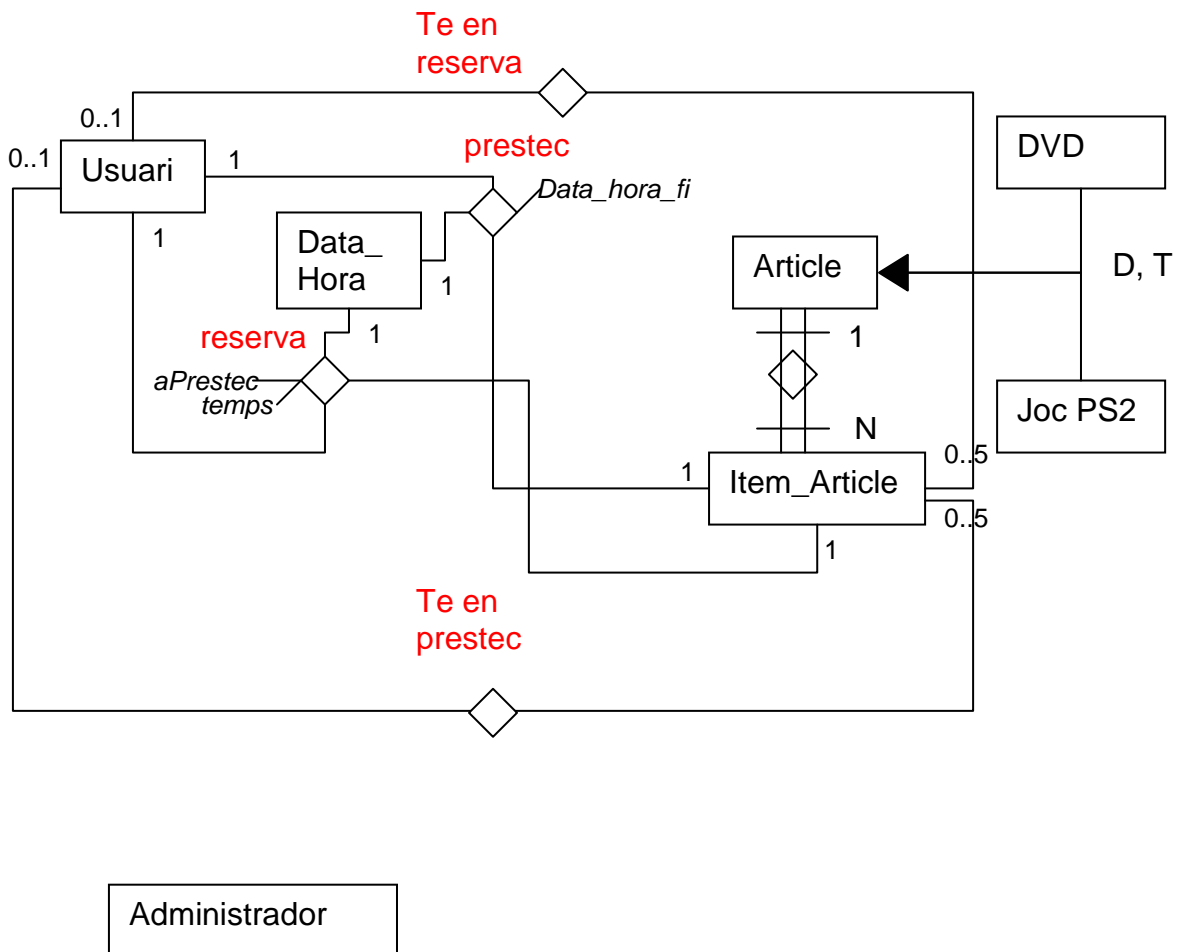
4 Disseny de l'aplicació

4.1 Introducció

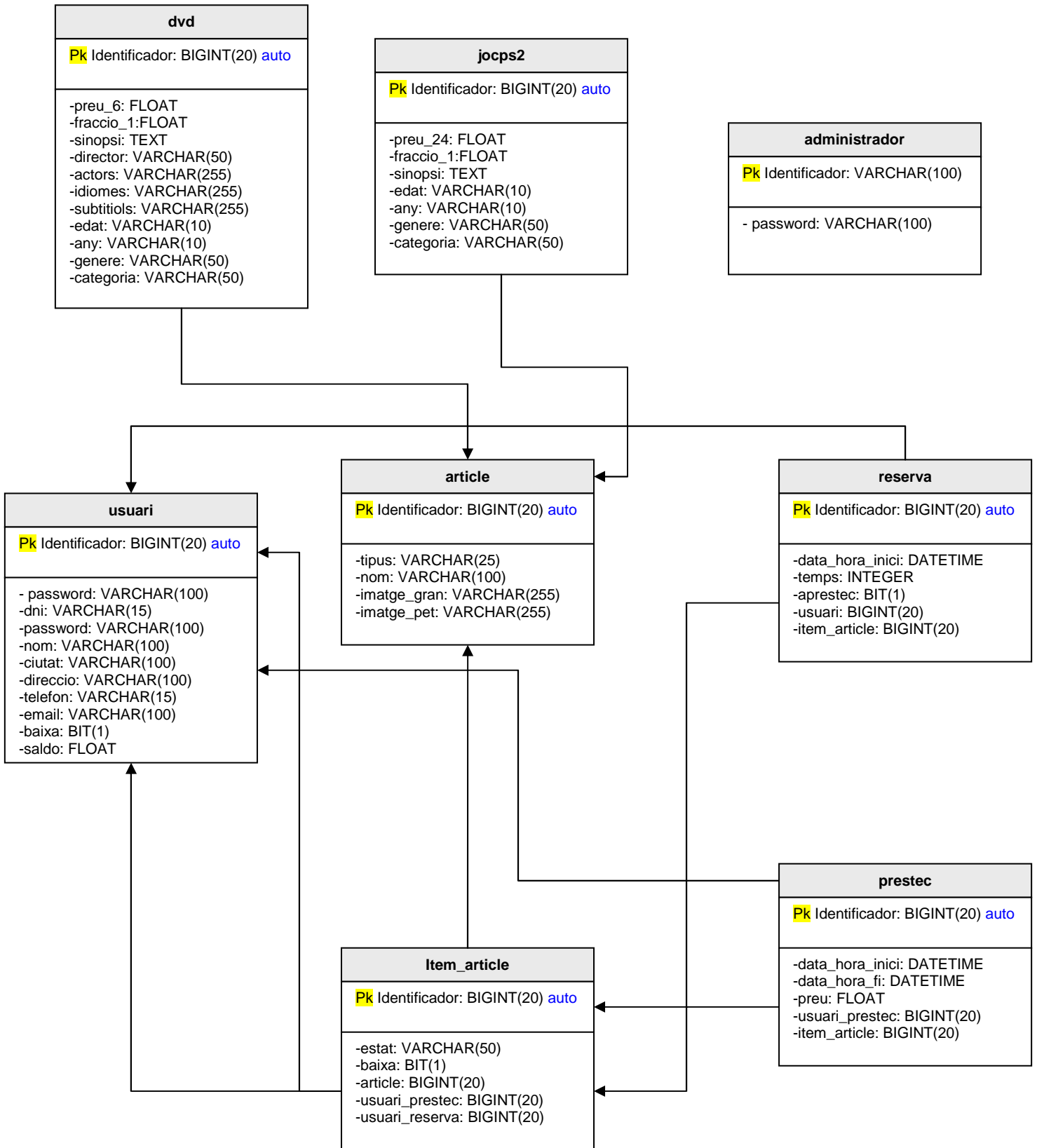
En aquest apartat s'especifica com es va fer el disseny de l'aplicació

4.2 Disseny de la base de dades

A partir de les especificacions del projecte es va decidir pel següent diagrama ER (Entity-Relationship):



Del model conceptual passem al model lògic de la base de dades que el podem representar de la següent forma:



on les fletxes representen les claus foranes.

4.2.1 Taules 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 MySQL
- Clau: PK indica que aquesta columna forma part de la clau primària. FK indica que aquesta columna es clau forana a un altra columna d'un altra taula. U indica que la columna es única (clau alternativa). I indica que s'ha fet un índex sobre la columna
- Req: indica si la columna es requerida, això es, que no accepti valors nulls
- Descripció: Breu descripció del camp

Taula usuari

Taula usuari				
Columna	Tipus	Clau	Req	Descripció
Identificador	BIGINT(20)	PK	Si	Identificador únic de usuari
Dni	VARCHAR(15)	U	Si	Dni de l'usuari
Password	VARCHAR(15)		Si	Password de l'usuari
Nom	VARCHAR(100)		Si	Nom i cognoms de l'usuari
Direccio	VARCHAR(100)		Si	Direccio de l'usuari
Telefon	VARCHAR(15)		Si	Telefon de l'usuari
Email	VARCHAR(100)		No	E-mail de l'usuari

Baixa	BIT(1)		Si	Indica si l'usuari esta de baixa
Saldo	FLOAT		Si	Indica el saldo de l'usuari

Taula administrador

Taula administrador				
Columna	Tipus	Clau	Req	Descripcio
Identificador	VARCHAR(100)	PK	Si	Identificador únic de administrador
Password	VARCHAR(100)		Si	Dni de l'usuari

Taula article

Taula article				
Columna	Tipus	Clau	Req	Descripcio
Identificador	BIGINT(20)	PK	Si	Identificador únic de article
tipus	VARCHAR(25)	U	Si	Tipus d'article
nom	VARCHAR(100)	U	Si	Nom de l'article
imatge_gran	VARCHAR(255)		Si	url relativa on es troba la imatge gran
imatge_pet	VARCHAR(255)		Si	url relativa on es troba la imatge petita

Taula item article

Taula item_article				
Columna	Tipus	Clau	Req	Descripció
Identificador	BIGINT(20)	PK	Si	Identificador únic de exemplar
estat	VARCHAR(50)		Si	Estat en que es troba l'exemplar
baixa	BIT(1)		Si	Si l'exemplar està de baixa
article	BIGINT(20)	FK1	Si	Identificador de l'article al qual aquest exemplar pertany
Usuari_prestec	BIGINT(20)	FK2	No	Identificador de l'usuari el qual te aquest exemplar en prestec
Usuari_reserva	BIGINT(20)	FK3	No	Identificador de l'usuari el qual te aquest exemplar en reserva

Taula dvd

Taula dvd				
Columna	Tipus	Clau	Req	Descripció
Identificador	BIGINT(20)	PK, FK	Si	Identificador únic de dvd
Preu_6	FLOAT		Si	Preu de les 6 primeres hores de préstec del dvd
Fraccio_1	FLOAT		Si	Preu de la fraccio d'un hora de préstec passades les 6 h.
sinopsi	TEXT		No	Sinopsi del dvd
Director	VARCHAR(50)		No	Director de la pel·lícula
actors	VARCHAR(255)		No	Actors de la pel·lícula
idiomes	VARCHAR(255)		No	Idiomes de la pel·lícula
subtitols	VARCHAR(255)		No	Subtítols de la pel·lícula
any	VARCHAR(10)		No	Any de producció
edat	VARCHAR(10)		No	Si esta recomanada a un cert sector
Genere	VARCHAR(50)		Si	Gènere al qual pertany el film
Categoria	VARCHAR(50)		Si	Categoria a la qual pertany

Taula jocps2

Taula jocps2				
Columna	Tipus	Clau	Req	Descripció
Identificador	BIGINT(20)	PK, FK	Si	Identificador únic del joc
Preu_24	FLOAT		Si	Preu de les 24 primeres hores de préstec del dvd
Fraccio_1	FLOAT		Si	Preu de la fraccio d'un hora de préstec passades les 24 h.
any	VARCHAR(10)		No	Any de producció
edat	VARCHAR(10)		No	Si esta recomanada a un cert sector
Genere	VARCHAR(50)		Si	Gènere al qual pertany el joc
Categoria	VARCHAR(50)		Si	Categoria a la qual pertany

Taula reserva

Taula reserva				
Columna	Tipus	Clau	Req	Descripció
Identificador	BIGINT(20)	PK	Si	Identificador únic de la reserva
Data_hora_inici	DATETIME		Si	Data amb precisió de minuts en la qual s'ha efectuat la reserva
Temps	INTEGER		Si	Temps en minuts que queda perquè venci la reserva
Aprestec	BIT(1)	I	Si	Si aquesta reserva a passat a préstec (o ha vençut el temps de reserva)
Usuari	BIGINT(20)	FK1	Si	Identificador de l'usuari que ha fet la reserva.
Item_article	BIGINT(20)	FK2	Si	Identificador de l'exemplar sobre el que s'ha fet la reserva

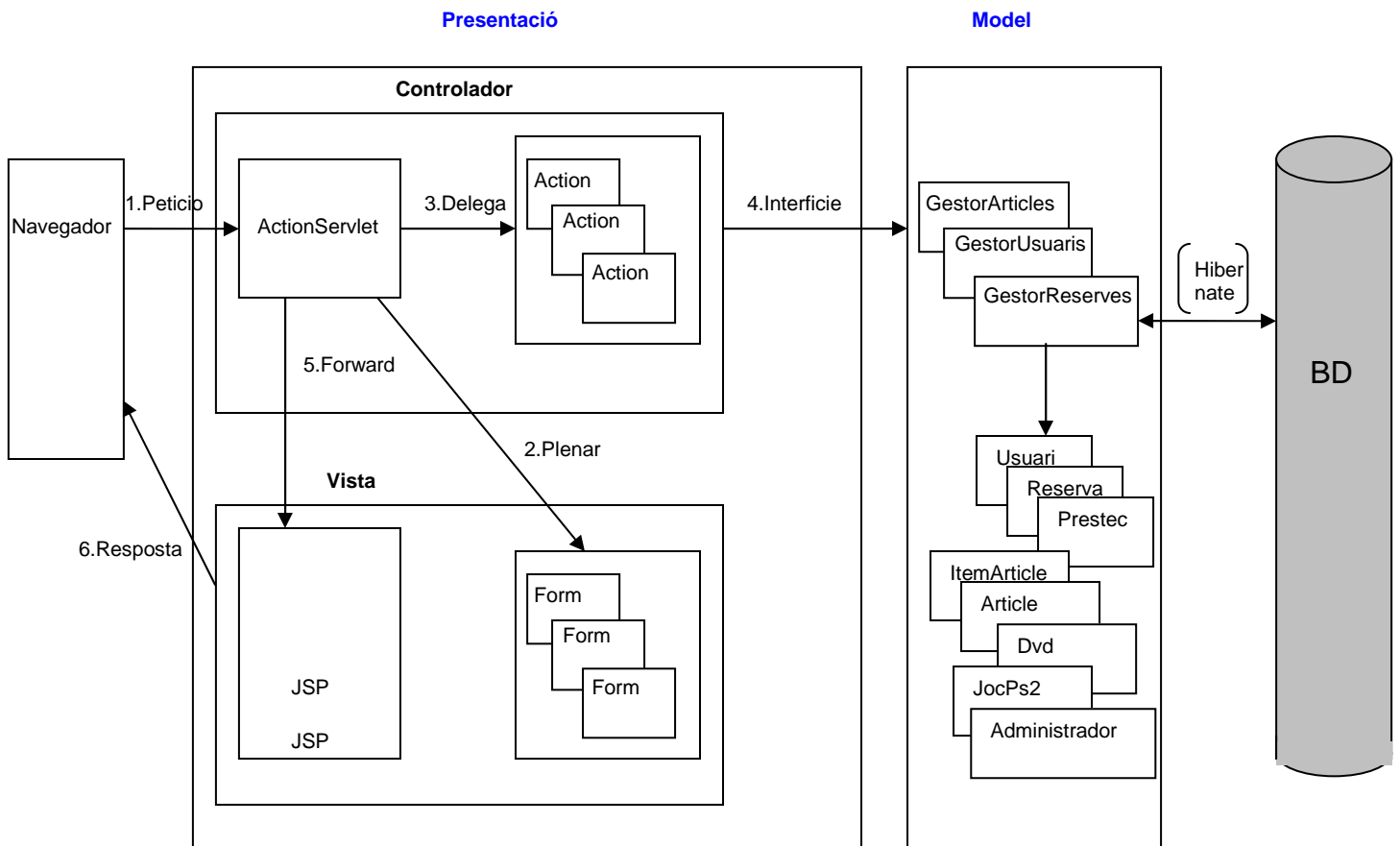
Nota: L'índex es realitza sobre aquesta columna donat el cas que cada 5 minuts un procés mira les reserves que tenen aquest camp a false.

Taula préstec

Taula préstec				
Columna	Tipus	Clau	Req	Descripció
Identificador	BIGINT(20)	PK	Si	Identificador únic del préstec
Data_hora_inici	DATETIME		Si	Data amb precisió de minuts en la qual s'ha efectuat la reserva
Data_hora_fi	DATETIME		Si	Data amb precisió de minuts en la qual s'ha efectuat el retorn
Preu	FLOAT		Si	Preu del préstec un cop finalitzat
Usuari	BIGINT(20)	FK1	Si	Identificador de l'usuari que ha fet el préstec.
Item_article	BIGINT(20)	FK2	Si	Identificador de l'exemplar sobre el que s'ha fet el préstec

4.3 Disseny de les classes**4.3.1 Introducció**

A continuació es mostra un esquema de com funciona l'aplicació



A grans trets l'aplicació es pot dividir en Presentació i Model. Presentació s'ocupa de tot el referent amb la interacció amb l'usuari mentre que Model s'ocupa de la interacció amb les dades.



Imaginem un exemple, un usuari fa una petició de un llistat de pel·lícules que continguin en el títol la paraula "america". L'usuari té prèviament carregada la pàgina i prem el boto envia en el formulari. Que es el que passa?

1. La petició pot ser p.ex una acció com `visualitzaArticles.do`, es a dir vol que es visualitzin els articles.
2. `ActionServlet` es qui controla el trafic, podriem dir que s'ocupa de llegir el fitxer `struts-config.xml` per saber com controlar el flux d'execució. El mes normal es que utilitzi un `Form` associat a l'acció que s'ocupa de plenar amb les dades que l'usuari a introduït.

3. Evidentment ActionServlet es la màxima expressió de Controlador però no ho sap fer tot, així que delega en una altra Action específica de l'acció que sap que fer amb el Form. En aquest cas del form pot treure el nom de l'article a cercar
4. En la Action es crida a negoci. Des de el seu punt de vista no sap res de la base de dades ni de la lògica de negoci, fins i tot es normal que Presentació i Model estiguin en màquines separades. Només sap la interfície que se li proporciona, que en el nostre cas serà una classe Gestora amb un mètode per llistar articles a partir d'un nom de títol.
5. Si tot va be la Action fa un forward (vol dir "tirar cap a") cap a un jsp que s'ocuparà de llistar els articles (presentar-los convenientment)

Aquest esquema pot tindre variacions, però a grans trets mostra la dinàmica de l'aplicació.

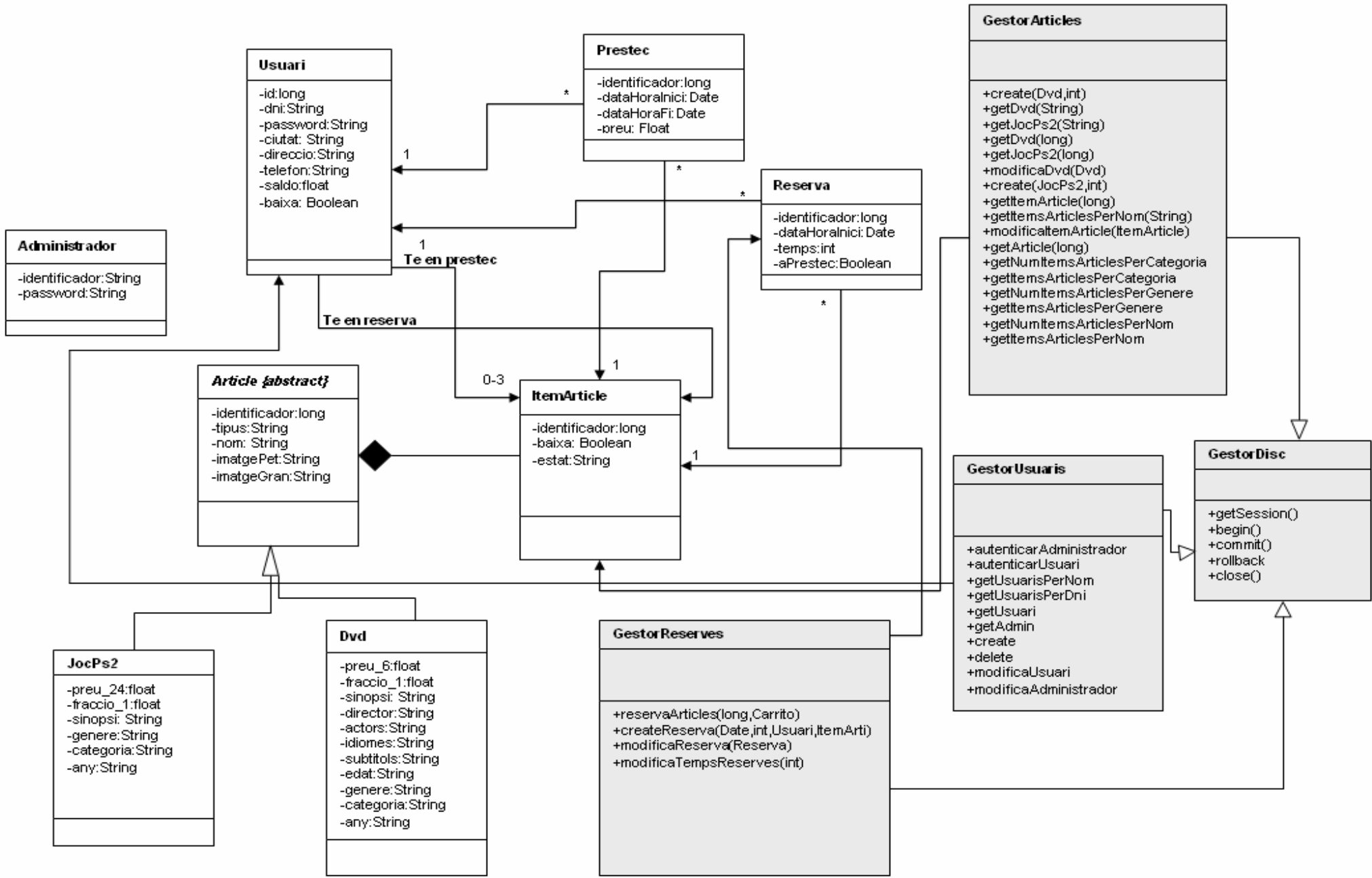
El disseny de la presentació ens ve marcat pel framework MVC de Struts, així que ens ocupem del disseny de la capa de model. Tot i així cal mencionar que per implementar el concepte de carrito es va implementar una classe Carrito que viu a presentació (només durant la sessió). Per moure dades de forma eficient en presentació es van crear també dos

classes DTO (Data Transfer Object)  ReservaArticleDTO.java  ViewArticleDTO.java

Les classes de model són totes aquelles que tenen que veure amb les dades i el tipus de problema a resoldre.

4.3.2 Diagrama de classes de model

Es mostra un diagrama amb les classes de model



Dins d'aquest diagrama es poden distingir dos grups diferents:

- Classes de negoci que corresponen a entitats del problema. Aquestes son directament mapejables en taules mitjançant els fitxers hbm d'hibernate.
- Classes gestores. Aquí es on s'implementa la lògica de negoci. En aquestes classes es on es fa l'accés a Base de dades, a mes de ser cridades en les classes Action de presentació

Justificació del diagrama

Articles

Definim Article com a classe abstracta perquè no es instanciable, només la utilitzem per encapsular uns atributs comuns a qualsevol tipus d'article. En aquest cas tenim JocPs2 i DVD, però en un futur es podrien incorporar mes sense problemes.

Cal tenir en compte que un article com pot ser un dvd de la pel·lícula que sigui, pot estar disponible amb 1 o més unitats per al seu préstec (exemplars). Aquest fet el reflectim al Diagrama fent servir una agregació, això és, els ítems (1,2,3,4,5) formen part de l'article i sense ell no tindrien sentit. Aquestes 5 unitats son diferents físicament (p.ex la unitat nº 1 de l'article 2587 pot estar alquilada mentre que la unitat 5 pot estar reservada), però formen part del mateix article.

Això fa que es parli indistintament d'un article en particular (la unitat 4 del DVD Y) com d'un article en general (el DVD Y). P.ex quan donem d'alta un article, estem donant d'alta automàticament tots els seus ítems especificant el número d'unitats.

Car remarcar la importància de l'atribut estat d'un article (DISPONIBLE, RESERVAT, PRESTEC)

Reserves/Prestecs

Segons el diagrama E-R un Préstec o Reserva queda identificat per un Usuari un ItemArticle i una data (afinant al minut). De totes formes farem servir un identificador numèric (que s'autoincrementi) doncs ens facilitarà la feina. Les taules Préstec i Reserva serveixen tant per l'històric com per la gestió.

Una Reserva pot passar a Prestec, això ho reflectim amb el boolea aPrestec que ens diu si la Reserva esta pendent o ja ha passat a Prestec.

S'ha pensat en utilitzar el temps per fer un compte enrere. El temps comença amb 70 i cada 5 minuts un procés automatitzat gestionat per Quartz farà que les reserves amb aPrestec fals i temps > 0 disminueixin el temps en 5 unitats.

Quan una Reserva es transformada a Prestec, el boolea aPrestec passa a cert i el temps para de baixar (el disparador no s'executarà).

Que una Reserva tingui el temps=0 vol dir que no ha sigut agafada (amb la consegüent multa a l'usuari de 1€)

Usuari

Un usuari te en reserva i/o préstec uns articles. Per això tindrem uns atributs de tipus array amb articles en Prestec i Reserva. Sabent-ne el tamany d'aquests arrays tindrem el numero d'articles en préstec i en reserva en un moment con

5 Implementació

5.1 *Eines de desenvolupament*

A continuació es llisten totes les eines i llibreries que s'han fet servir per desenvolupar aquest projecte:

- Com a IDE s'ha fet servir Eclipse 3.2 amb les WebTools 1.5.4 (un munt de plugins per desenvolupament web).
- S'ha fet servir el JDK 1.5.0.3
- Com a servidor d'aplicacions (millor dit contenidor de servlets) s'ha utilitzat Apache Tomcat 5.5
- Per a la base de dades s'ha escollit MySQL 5.0 i per la seva administració de forma gràfica MySql Administrator 1.2.9 i MySQL Query Browser 1.2.9 rc.
- La base de dades es va generar a partir del fitxers de configuració d'hibernate. Es va fer servir l'eina Hibernate Console 3.2.0 beta 8 per fer l'exportació. Després per trobar l'script sql de creació de la base de dades es va fer servir phpMyAdmin 2.8.0.2 per exportar l'esquema de la base de dades.
- Com a driver JDBC s'ha fet servir mysql-connector-java-5.0.3-bin.jar
- Llibreries del Framework Apache Struts, la versio 1.3.8

- Llibreries de Hibernate. S'ha fet servir la versió 3.2.4
- Llibreries de Javascript per la validació de formularis. S'ha fet servir fvalidate 5.01b
- Per a la programació de disparadors s'ha fet servir la llibreria quartz 1.6.0.

5.2 Estructura de l'aplicació en paquets

A mesura que una aplicació creix es necessita una estructura de treball més complexa, convé perdre cert temps estructurant les classes en paquets.

Per la part de fonts de Java es va estructurar l'aplicació en els següents paquets:

- **net.pelechano.administracion**
Aqui es troben totes les accions i forms que s'utilitzen en la part de presentació de l'administració de l'aplicació
- **net.pelechano.constantes**
Aqui es troba una classe per fer servir de repositori de constants de l'aplicació
- **net.pelechano.exceptions**
Aqui van les Excepcions de l'aplicació
- **net.pelechano.modelo**
Aqui van totes les classes de model així com els fitxers hbm i properties
- **net.pelechano.scheduling**
Aqui van les classes per fer la temporització de tasques
- **net.pelechano.usuarios**
Aqui es troben totes les accions i forms que s'utilitzen en la part de presentació de l'administració de l'aplicació

- **net.pelechano.util**

Classes d'utilitat per tractament d'imatges i funcions d'enciptació.

La part web esta estructurada de la següent forma:

- administracion
 - css
 - html
 - img
 - jsp
 - scripts
- usuarios
 - css
 - html
 - img
 - jsp
 - scripts
- imagenesjocsps2
 - imagenesJocsPs2Grandes
 - imagenesJocsPs2Peq
 - temp
- imagenespelis
 - imagenesPelisGrandes
 - imagenesPelisPeq
 - temp

6 Bibliografia

Respecte a la bibliografia diré que la quantitat de pàgines i documents consultats a sigut tant gran que si les fiques totes possiblement ompliria unes quantes pàgines.

Principalment he consultat dos llibres:

Struts: The Complete Reference (Osborne Complete Reference Series)

by James Holmes

Beginning Hibernate From novice to Professional (Apress)

by Dave Minter and Jeff Linkwood

També he fet servir

Core Servlets and JavaServer Pages Volume 1: Core Technologies

by Marty Hall i Larry Brown.

Com a pàgines web les mes importants:

La d'hibernate

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

La de struts

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

La llibreria quartz

<http://www.opensymphony.com/quartz/>

Llibreria fvalidate de javascript

<http://web.archive.org/web/20041209061906/www.peterbailey.net/fValidate/types/atleast>

